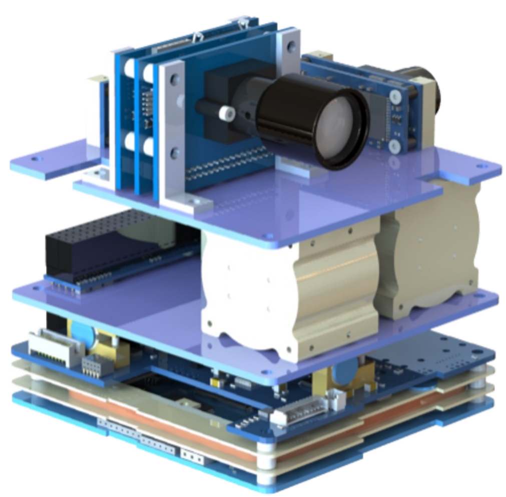
Integration and Testing of a Commercial ADCS for Naval Academy Standard Bus

MIDN 1/C Gwynn, Kaiser, McCarthy, Nordgauer, Thibault, Williams



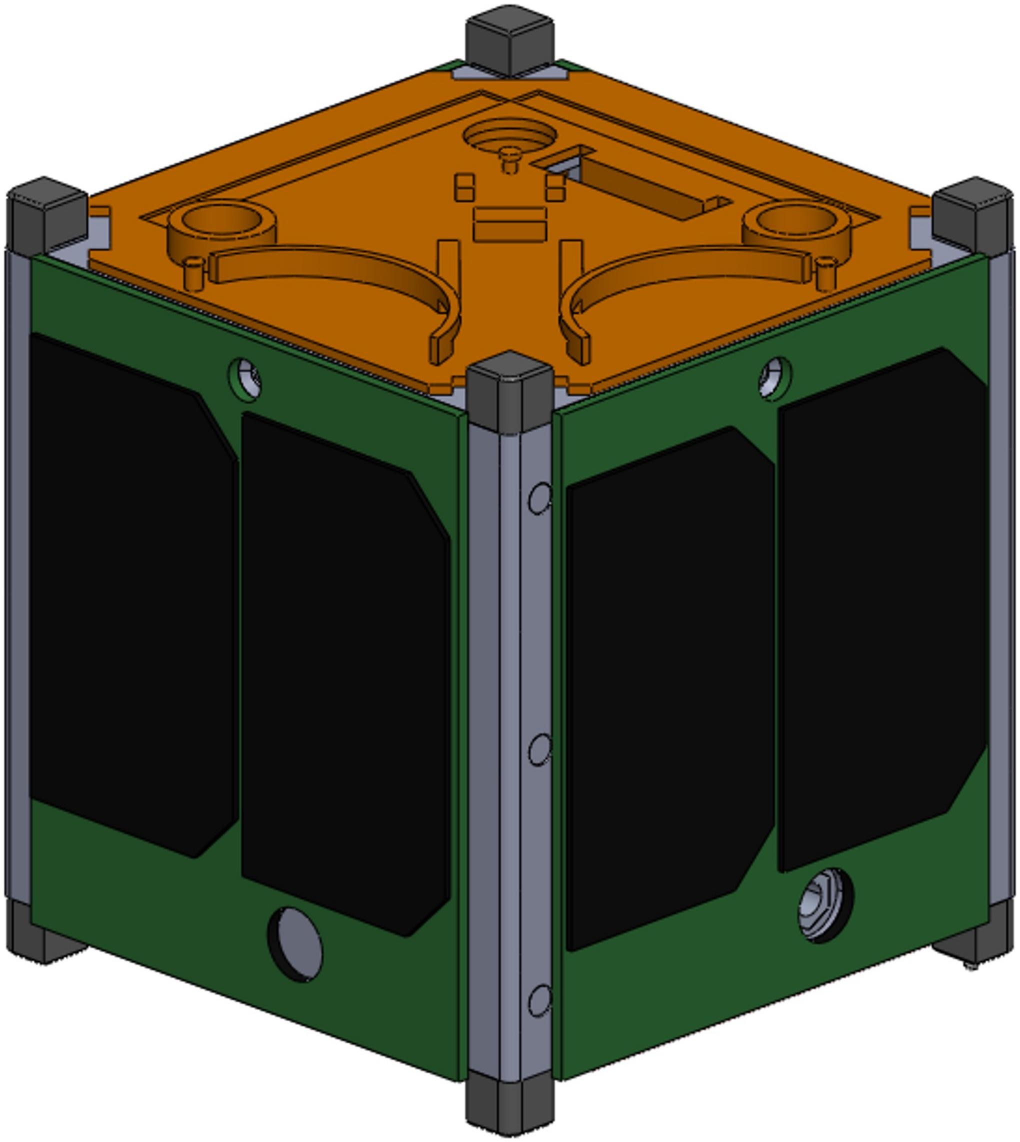
USNA Aerospace Engineering Department

Naval Academy Standard Bus

Provides:

- **Power**
- **Comms**
- **OBC**

To a variety of payloads

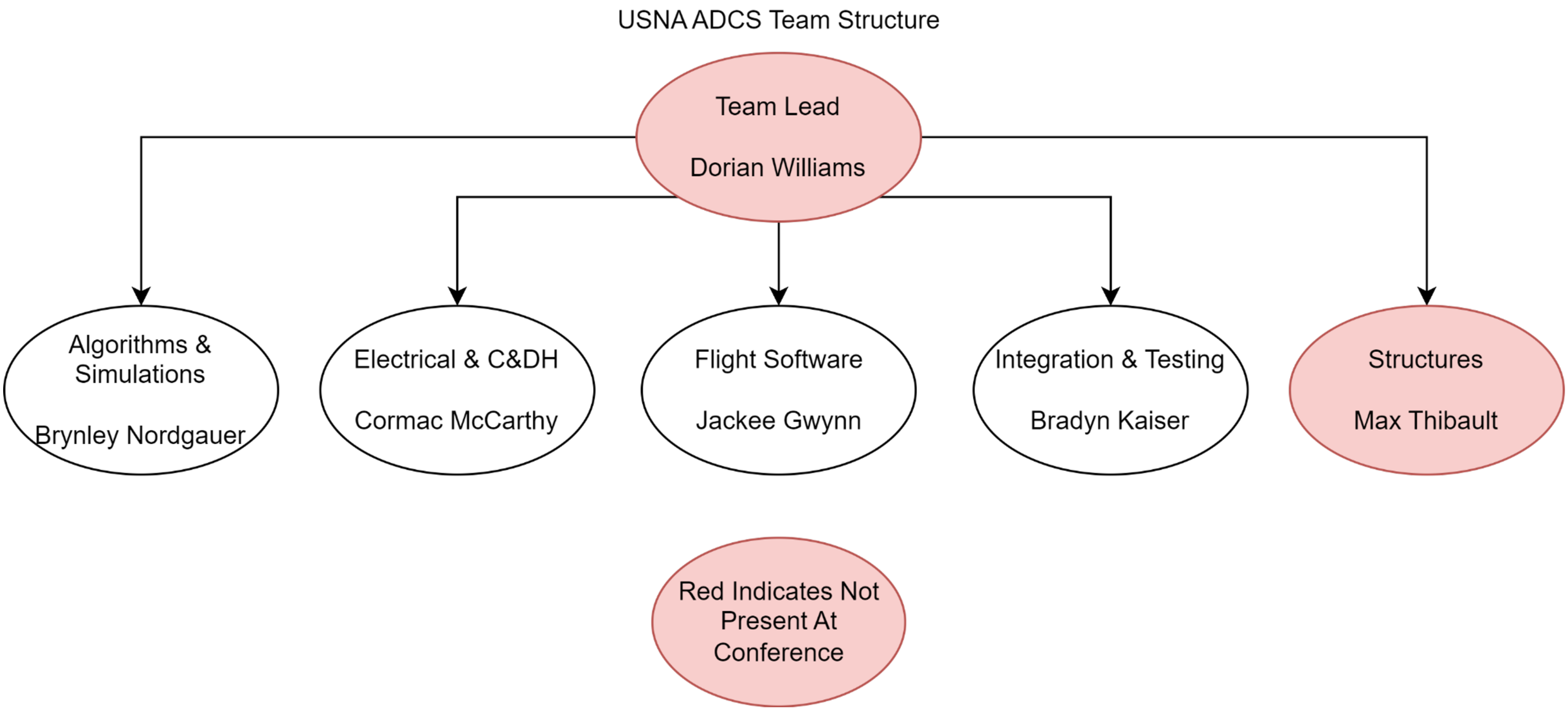


Mission Statement

The mission of the Naval Academy Standard Bus Attitude Determination and Control System (NASB-ADCS) is to:

- Integrate CubeSpace ADCS flight hardware and software with the NASB
- Develop an in-house interface between the NASB and the ADCS module
- Configure the ADCS module specifically for 3-Axis attitude control of the NASB satellite in LEO
- Design for the accommodation of payloads
- Test as a single integrated NASB-ADCS unit

Team Member Roles & Responsibilities



Concept of Operations

Brynley Nordgauer

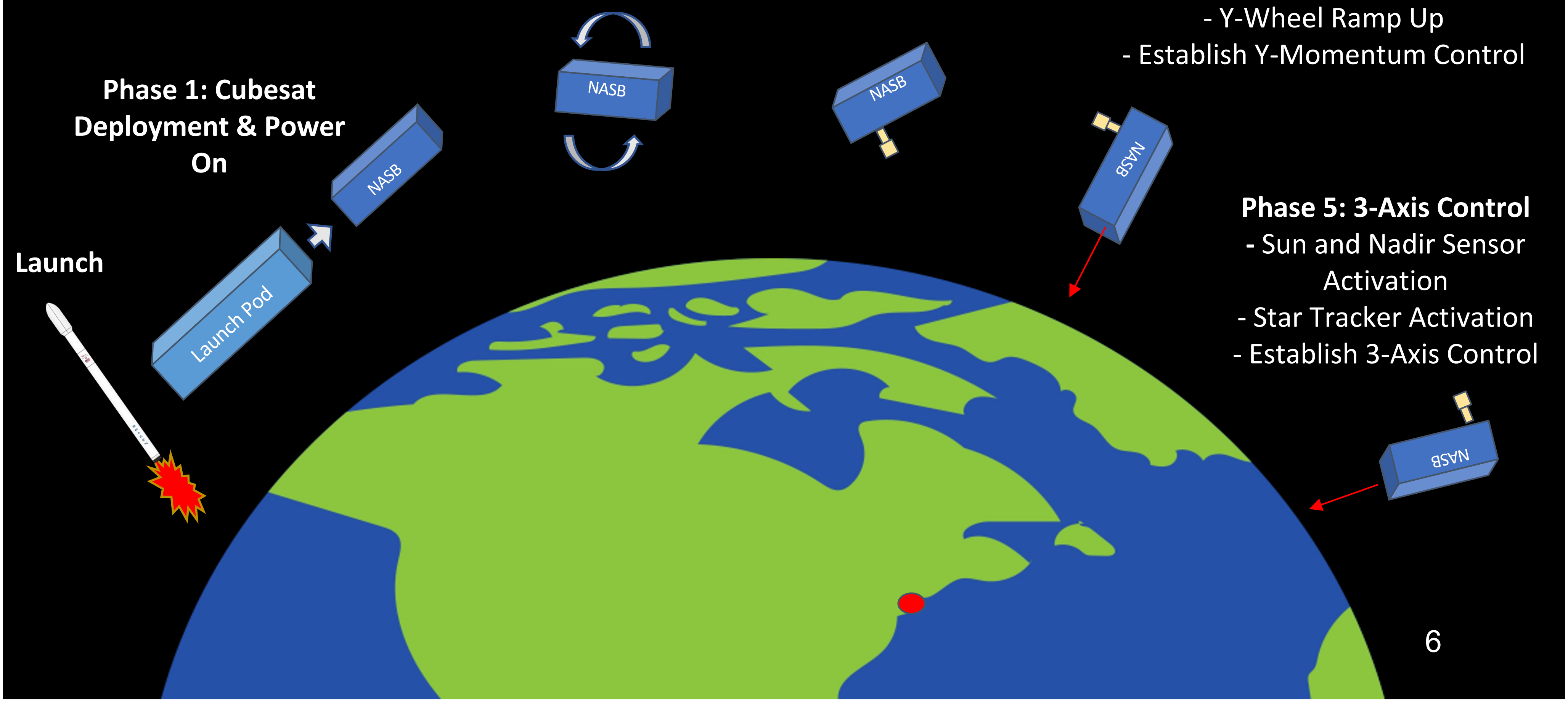
Concept of Operations

Note: Phase 3 can go to Phase 5 directly, but Phase 4 is passed through, since Phase 4 could be a terminal mode for some missions.

Phase 2: Detumble
- Initial Angular Rate Estimation
- Magnetic Detumble

Phase 3: Magnetometer Deployment
- Boom Deployment
- Calibration

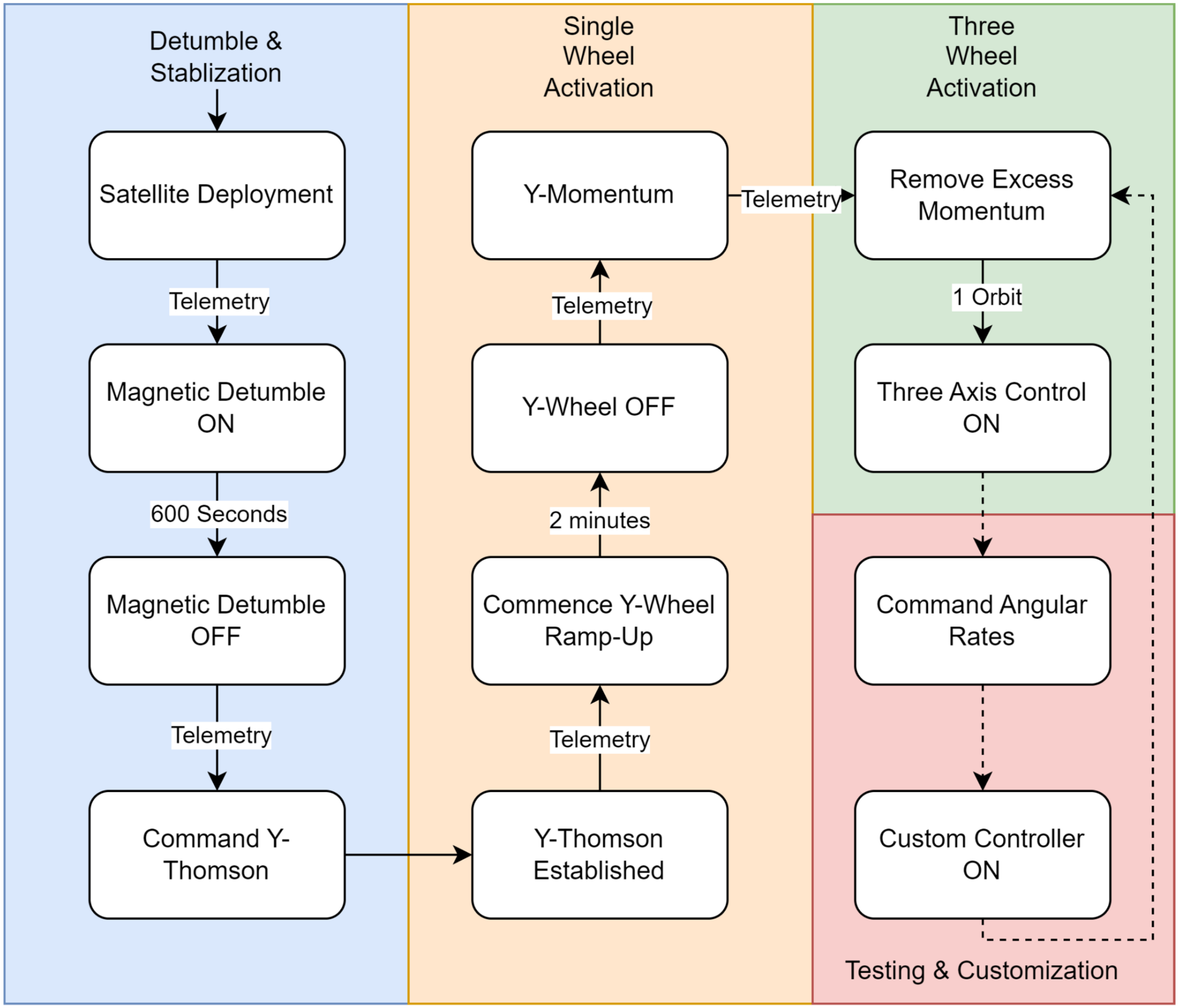
Phase 4: Y-Momentum
- Angular Rate/Pitch Estimation
- Y-Wheel Ramp Up
- Establish Y-Momentum Control



Algorithms & Simulations

Brynley Nordgauer

Switching Logic Flowchart



Simulation Capabilities

- Earth Models
  - Magnetic - Dipole Model
  - Orbit Propagator - Two Body
- CubeSat Models
  - 4 Inertia Test Cases Developed
- Disturbance Torques
  - Gravity Gradient
- Commissioning
  - Simulated through conditional switching logic
  - Magnetometer deployment assumed to have negligible effects
- Run Time
  - 30,000 seconds (8 hours and 20 minutes)

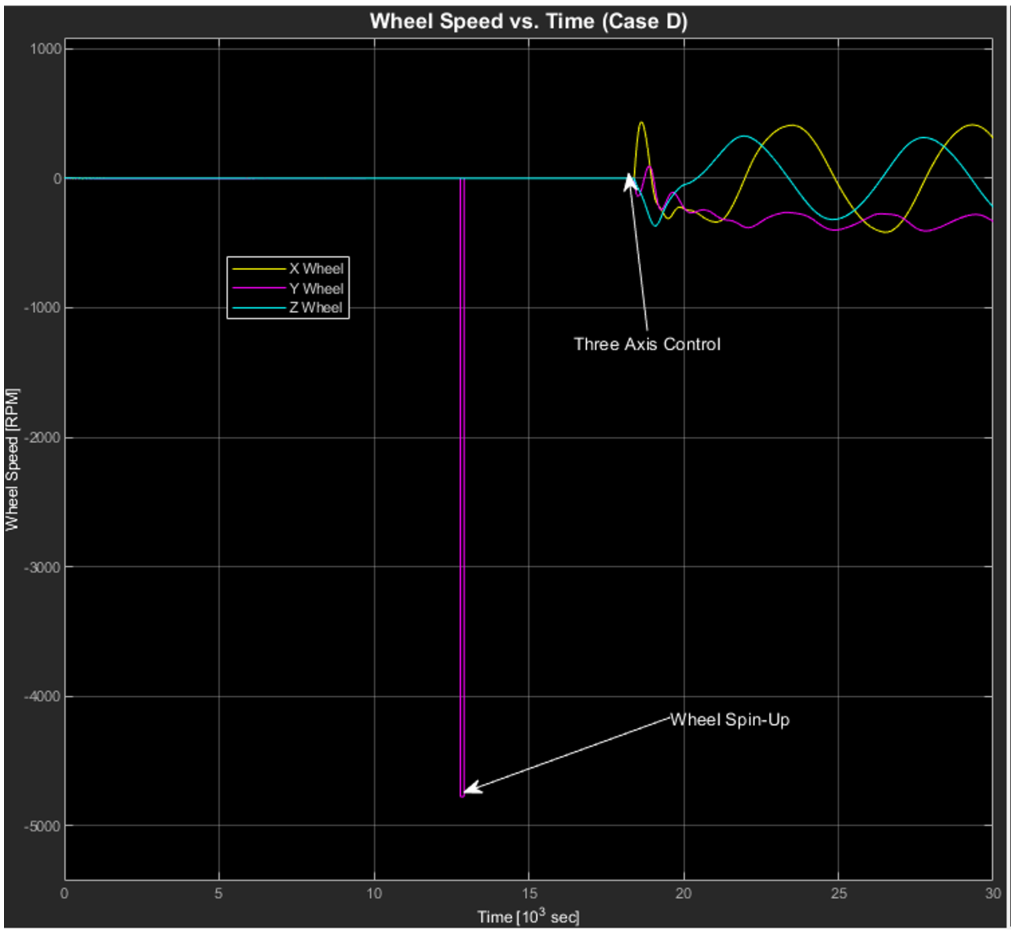
Simulation Initial Conditions

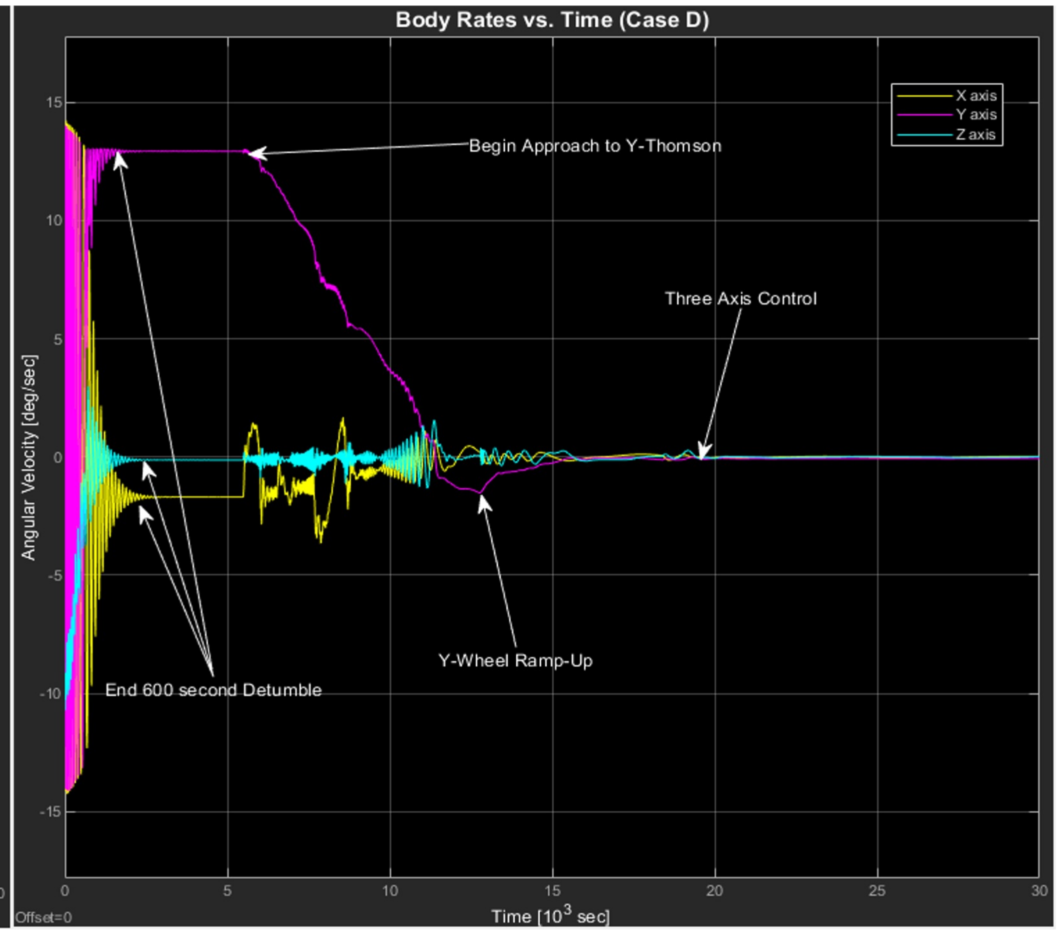
- Initial Conditions
  - $\underline{\omega}_0$ = [10, 10, -10] deg/sec
- Inertia Test Cases

Case	Inertia Tensor [g-m^2]	Dummy Mass
A	$I_A = \begin{bmatrix} 93.44 & -0.09724 & 0.5380 \\ -0.09724 & 95.39 & -0.2500 \\ 0.5380 & -0.2500 & 13.11 \end{bmatrix}$	No
B	$I_B = \begin{bmatrix} 69.04 & -0.1064 & 0.5522 \\ -0.1064 & 69.15 & -0.1982 \\ 0.5522 & -0.1982 & 9.781 \end{bmatrix}$	Yes
C	$I_C = \begin{bmatrix} 71.91 & -0.1055 & 0.5495 \\ -0.1055 & 72.21 & -0.2058 \\ 0.5495 & -0.2058 & 10.14 \end{bmatrix}$	Yes
D	$I_D = \begin{bmatrix} 78.91 & -0.1020 & 0.3104 \\ -0.1020 & 79.67 & -0.6101 \\ 0.3104 & -0.6101 & 11.05 \end{bmatrix}$	Yes

Table 3.2: Inertia Tensor Test Cases.

Simulation Results Case D

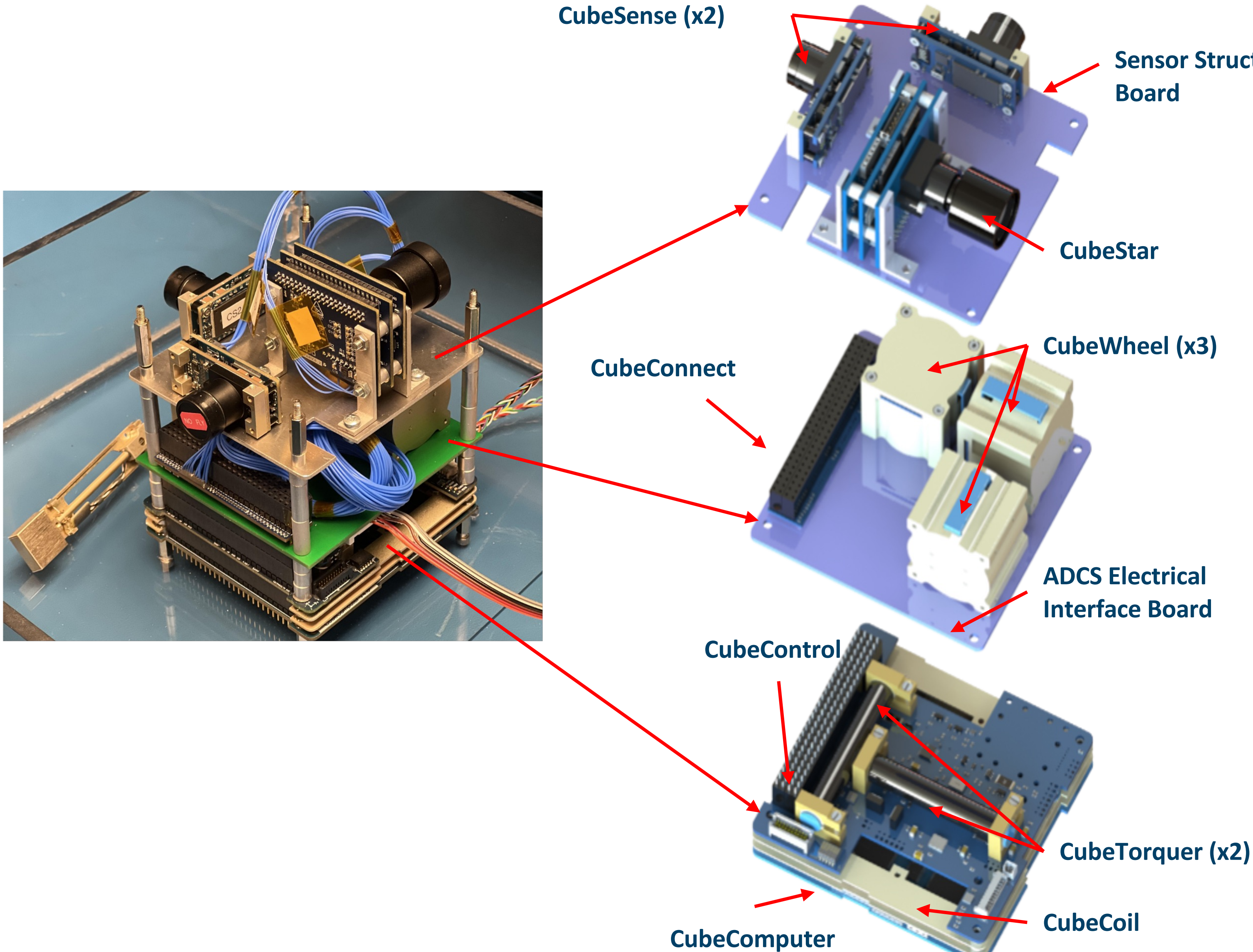




ADCS CubeSpace Hardware

Bradyn Kaiser

Hardware Overview - Stack



ADCS Processor Board

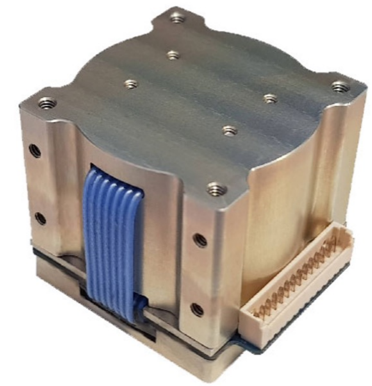
- CubeComputer
  - Control unit of ADCS stack
  - Houses estimator and control algorithms
  - TLM logging
  - Manages communication of ADCS modules

- Functions
  - Integrated RTC and internal and external watchdog
  - 4 MB Flash for code and in-flight reprogramming
  - FPGA for EDAC and SEU protection
  - Current monitoring for latchup protection and power cycle ability
  - I2C, CAN, and UART interfaces

ADCS Actuators

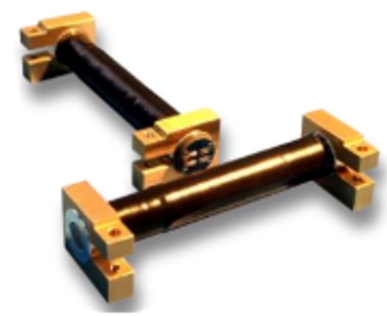
Small+ CubeWheel x3

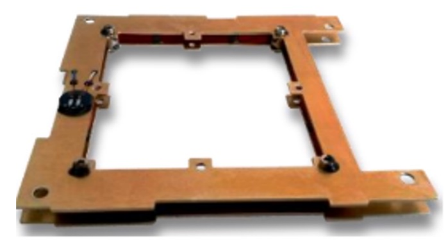
- Performance
  - Max momentum: 3.6 mNms
  - Max wheel speed: +/- 6000 rpm
  - Max torque: 2.3 mNm
  - Dynamic imbalance < 0.014 g-cm^2



CubeRod x2 and CubeCoil x1

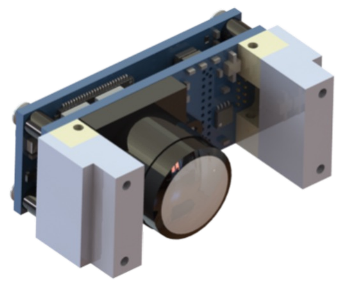
- Directly interfaced with CubeControl
- Performance
  - Max magnetic moment: +/- 0.48 Am^2



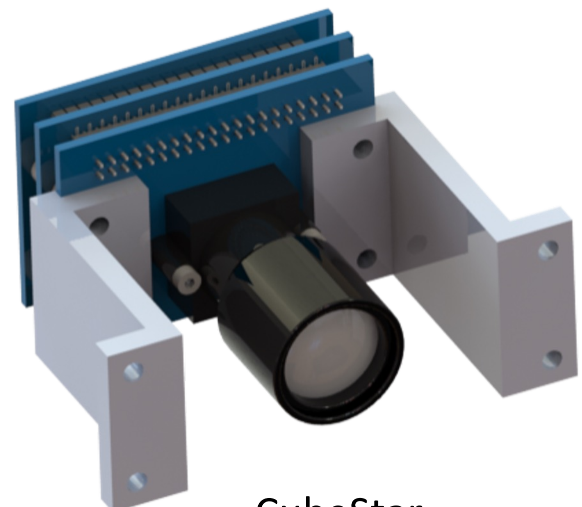


Hardware Overview - Sensors

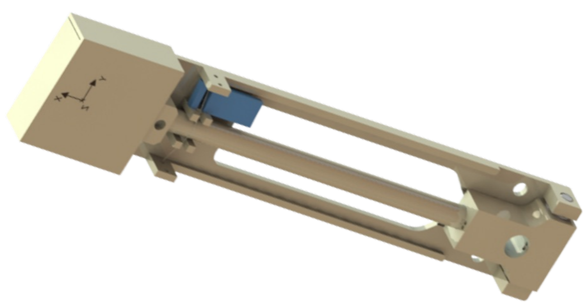
Sensor Suite



CubeSense



CubeStar

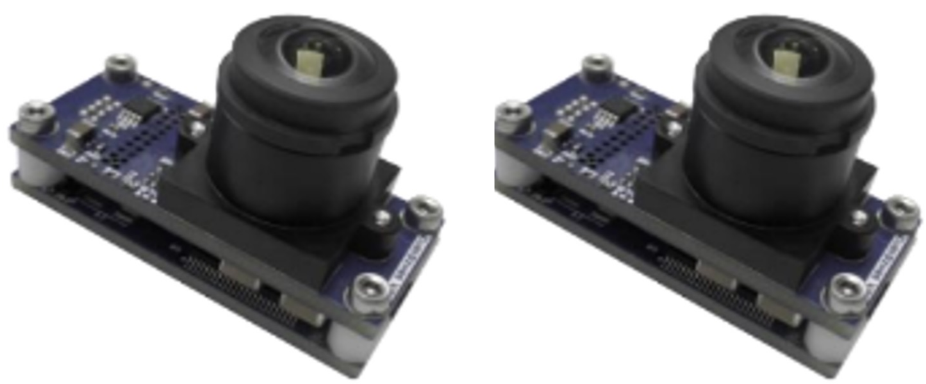


CubeMag

ADCS Sensors

CubeSense

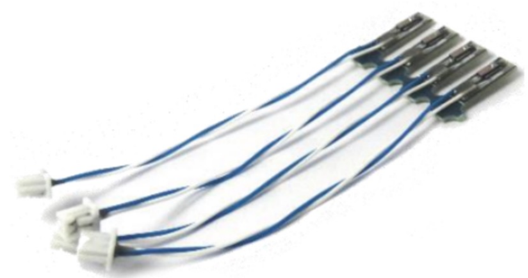
- Configurable to Nadir or Sun sensors during manufacturing
  - NASB ADCS has one of each configuration
- Performance
  - Sensor Accuracy: <0.2°
  - FOV: 180°



ADCS Sensors

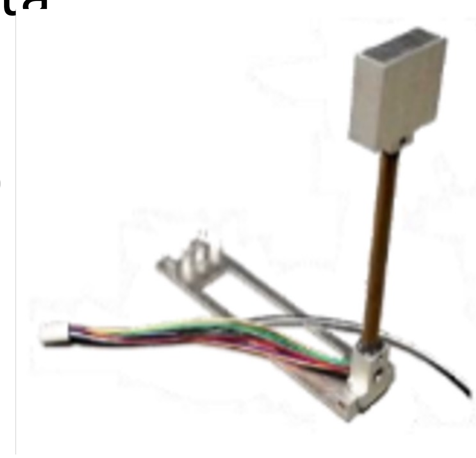
Coarse Sun Sensors (CSS) x10

- Spread across different faces of satellite
- Accuracy: < 10°



CubeMag

- External deployable magnetometer
- Provides entirety of magnetic field data
    - Deployed/Undeployed
- Measurement noise < 50 nT (per axis)



ADCS Sensors

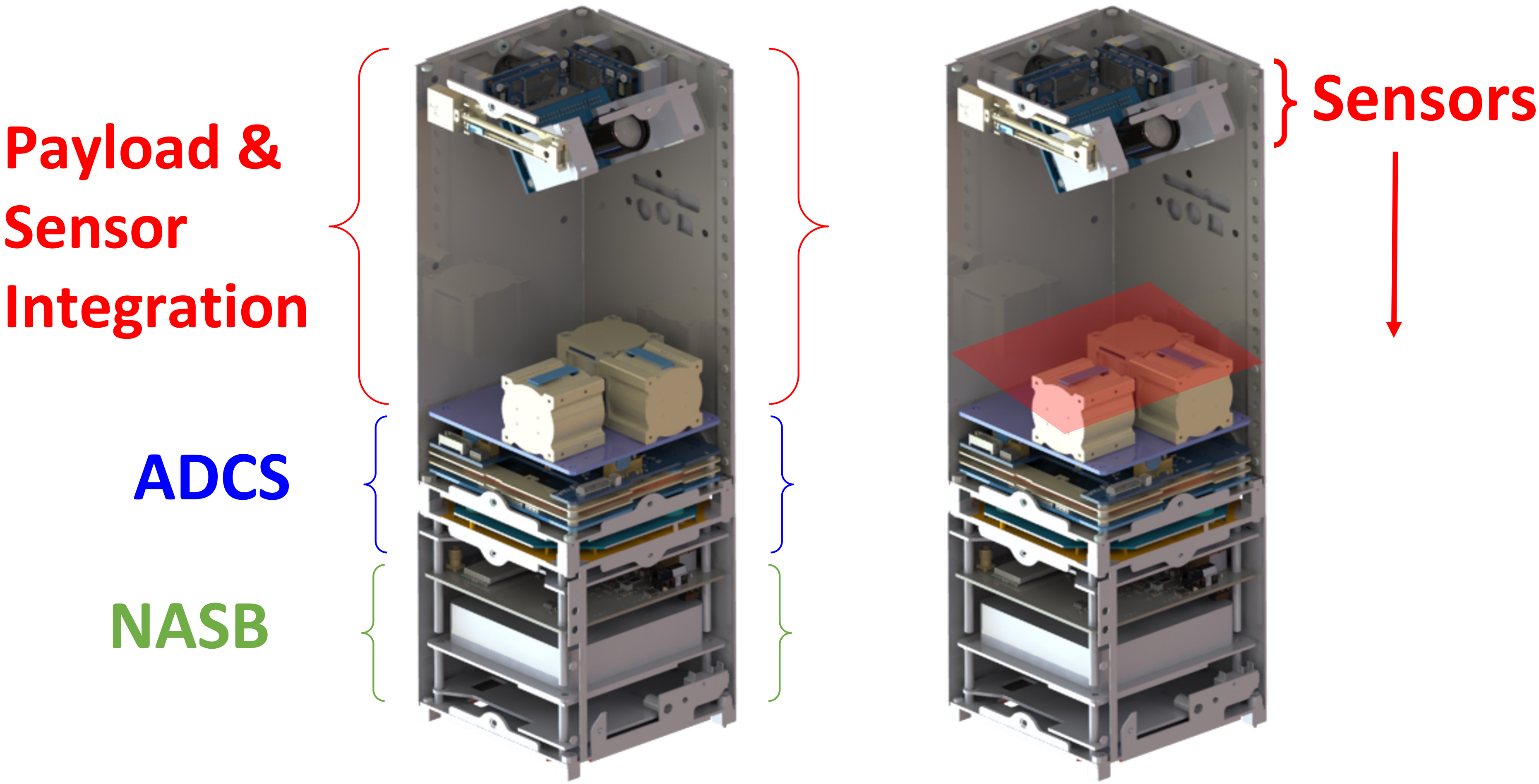
CubeStar

- Intended for usage in low-power, high performance
- Performance
  - FOV: 58° x 47°
    - Designed Sun Exclusion Angle - 35° off-boresight
  - Hipparcos Star Catalogue - 410 Stars
    - Max: 38 \ Min: 2
    - Sensitivity: < 3.8 Star Magnitude
  - Sky cover: 99.71%
  - Accuracy: 0.02° (across boresight, 3σ)
  - Max acquisition rate: 0.3°/s
  - Up to 1 Hz update rate

Structures

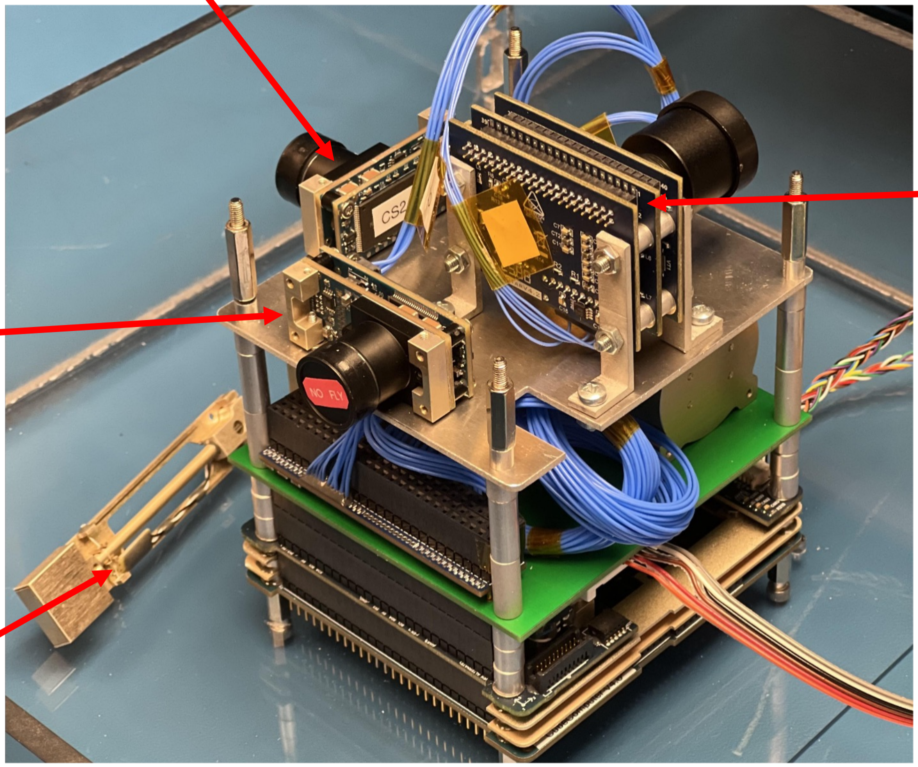
Bradyn Kaiser

Overall 3U Integration



CAD Models: Sensor Configuration

CubeSense
(Earth Horizon)



CubeSense
(Earth Horizon)

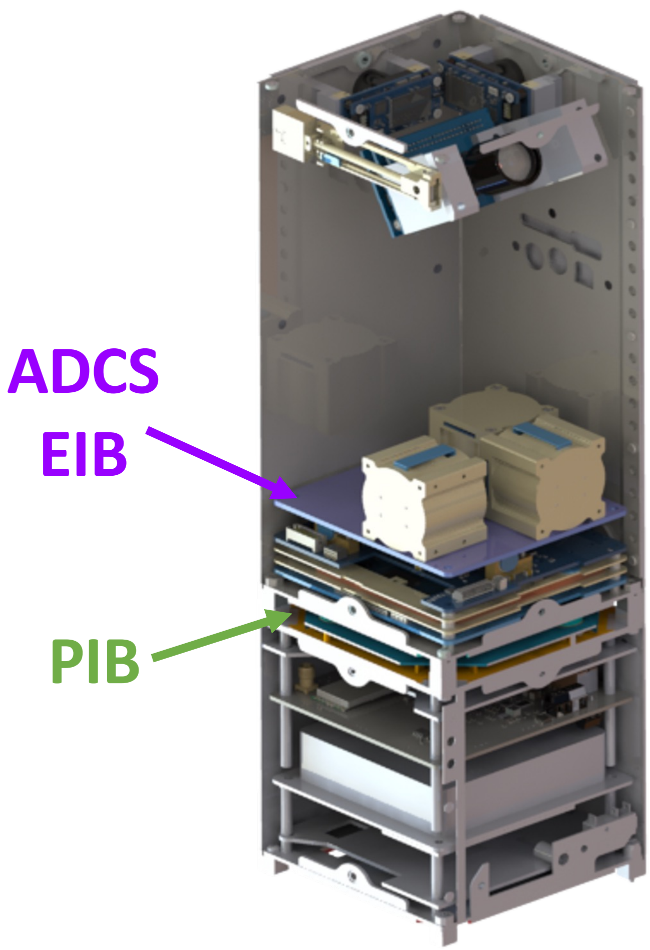
CubeStar
(Star Tracker)

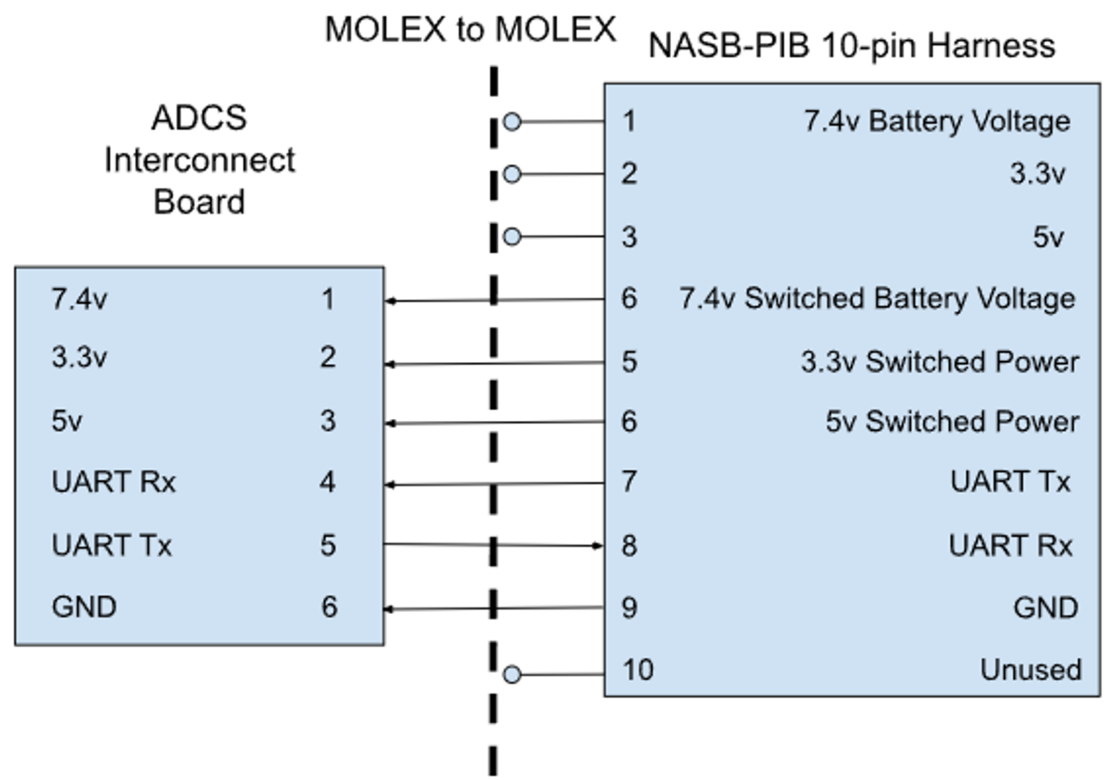
Deployable
Magnetometer

Electrical Interface

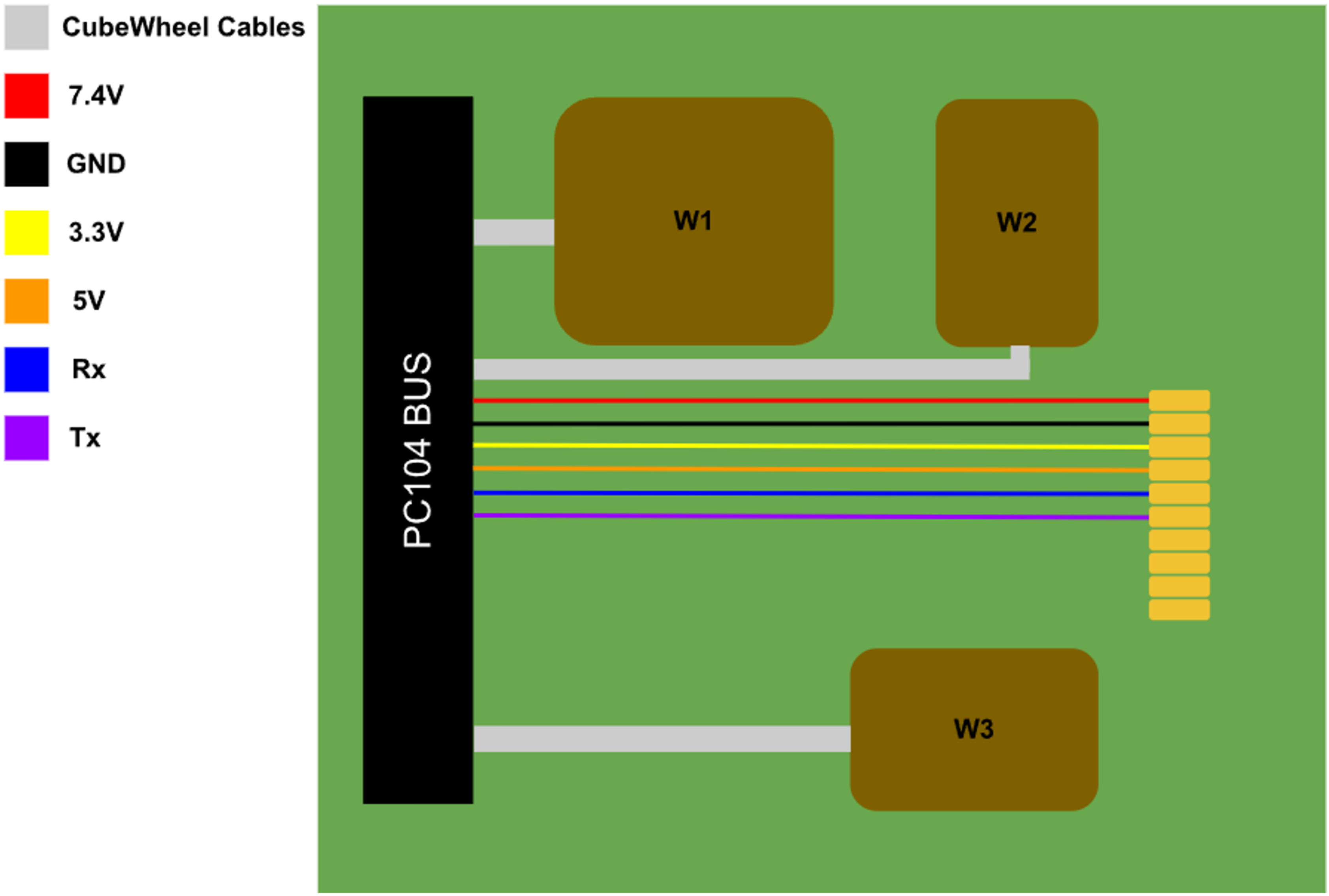
Cormac McCarthy

Electrical Integration: Block Diagram



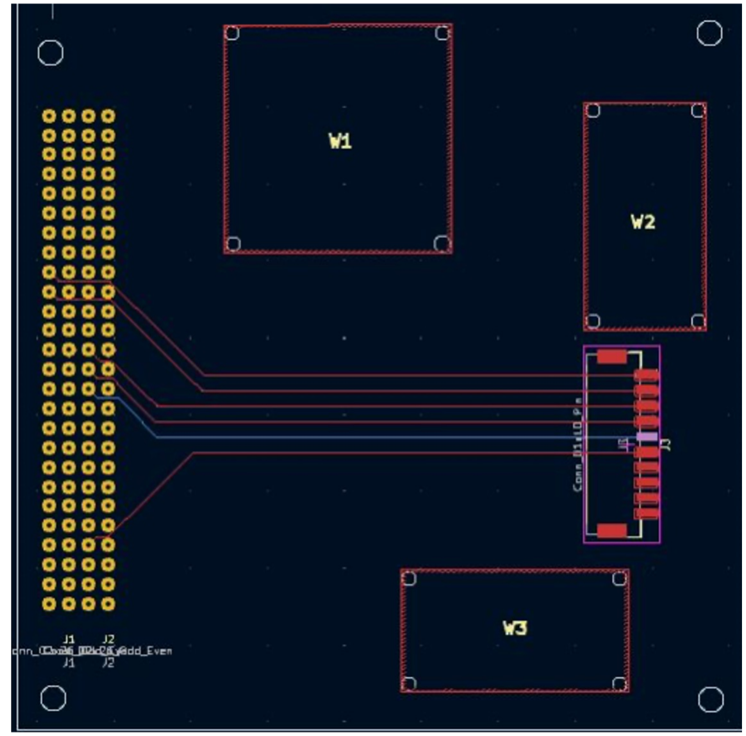


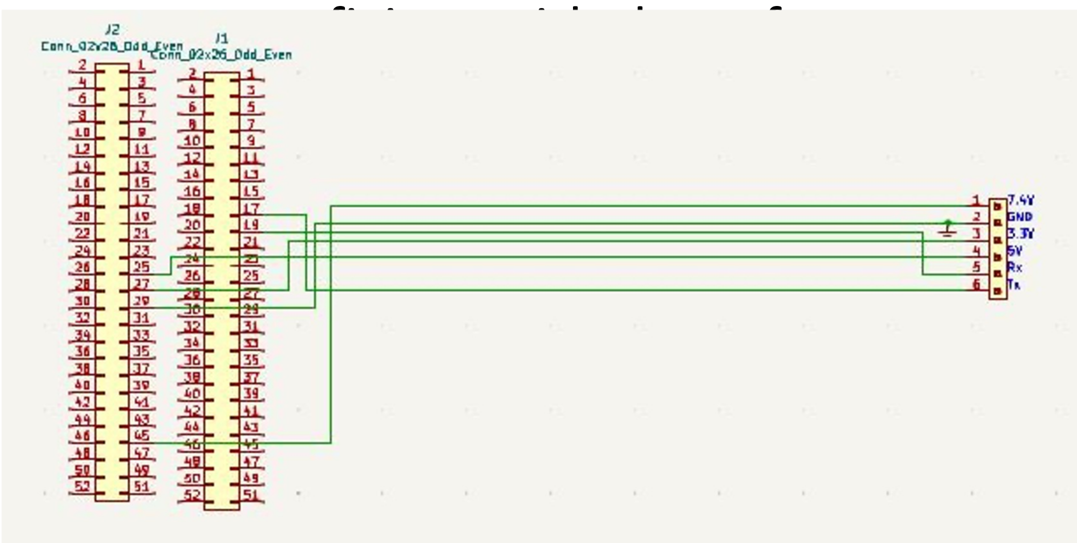
Electrical Integration: ADCS Interconnect Board



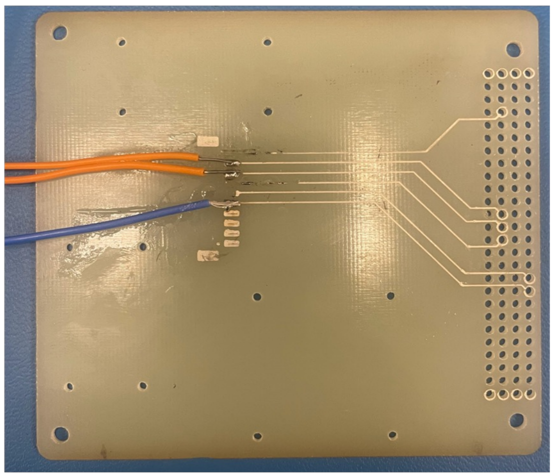
PCB Design: KiCad

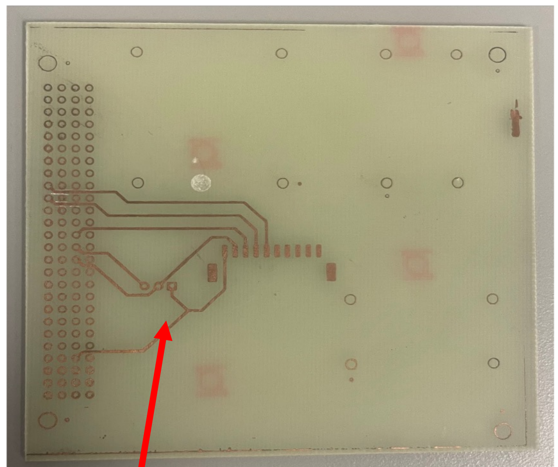
- Used the open source software KiCad to design the PCB
- Also used to format the fabrication files
- Took time to become

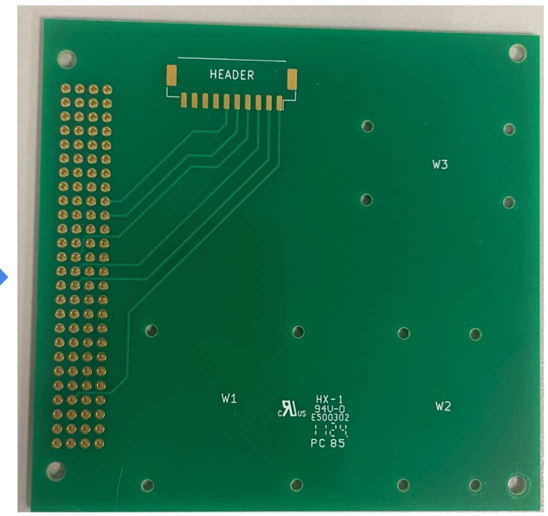




Iterations



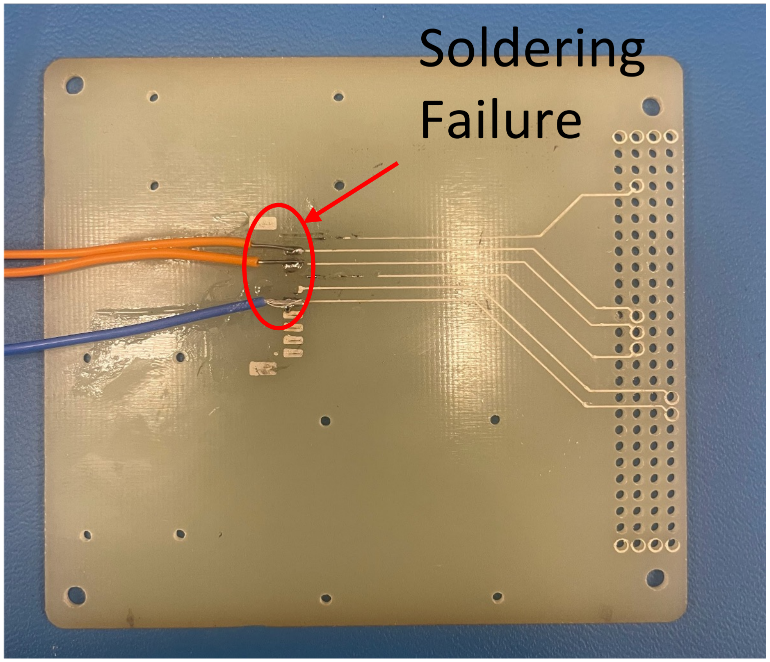




3.3V Regulator

Prototyping: Voltera

- V-One PCB Printer
- Learning curve to rapidly produce high quality prototypes
- Special temperature considerations for soldering to conductive ink
- Once perfected, a prototype could be produced in ~3 hrs



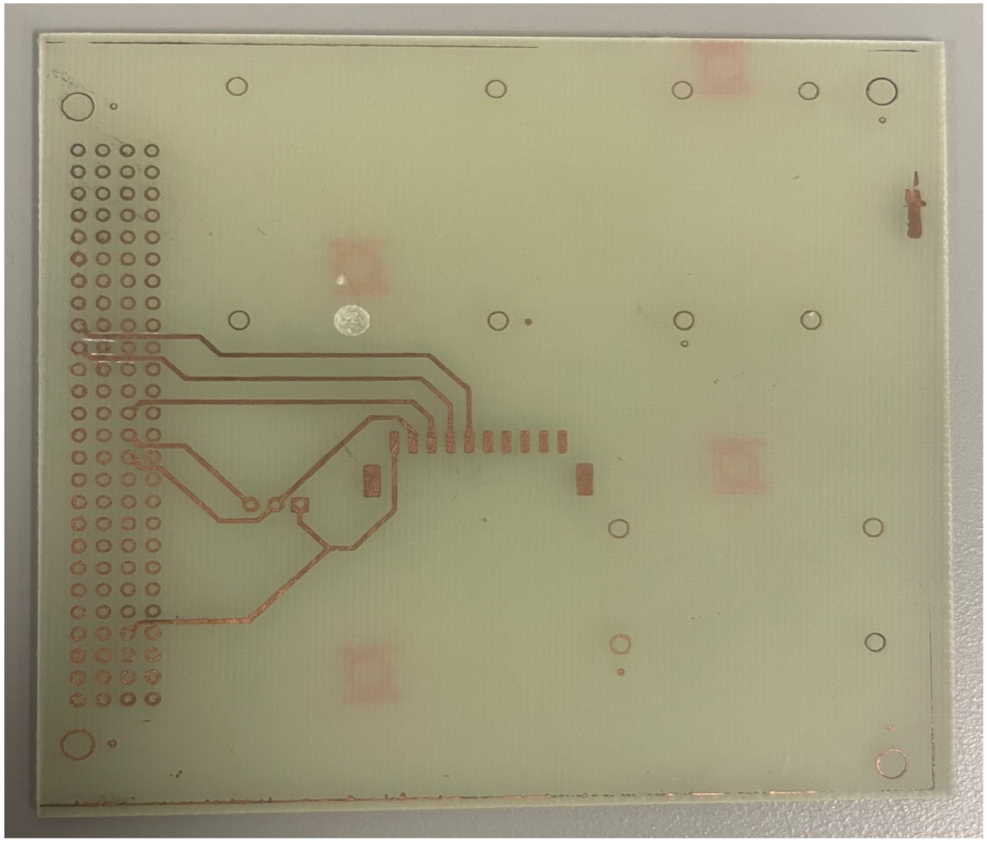
Voltera prototype

Prototyping: Etching

- Another prototyping method used involved chemically etching a copper plated sheet of FR4.

- This was outsourced to another academic department and as a result took much longer

- This method also produced lower resolution prints

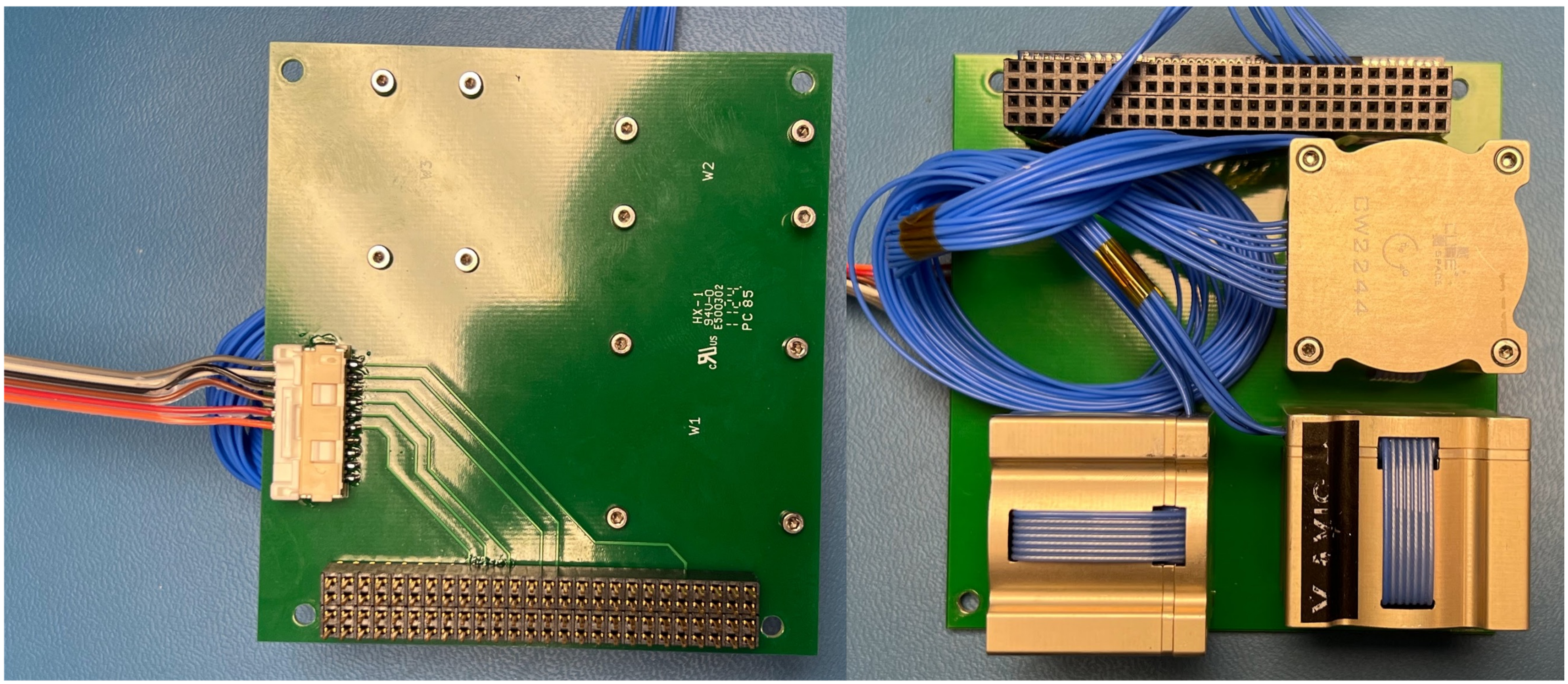


Etched prototype with voltage regulating capabilities

Final Board

- The final board was professionally printed by Precision PCBs using DigiKey's PCB Designer tool.

- A 10 circuit MOLEX header and wiring harness was constructed to interface the board with the NASB Payload Interface Board.



Final board integrated with CubeWheels

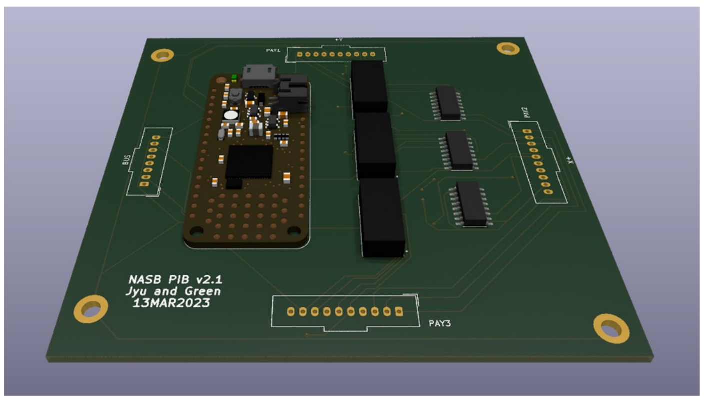
Flight Software: Telecommands and Telemetry

Jackee Gwynn

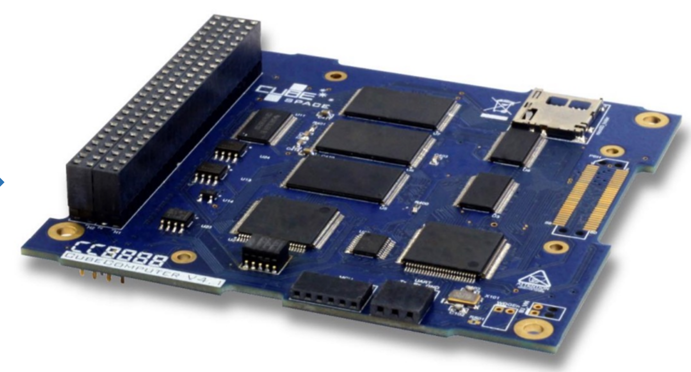
Integration Basics

NASB
Payload Interface Board
In-House

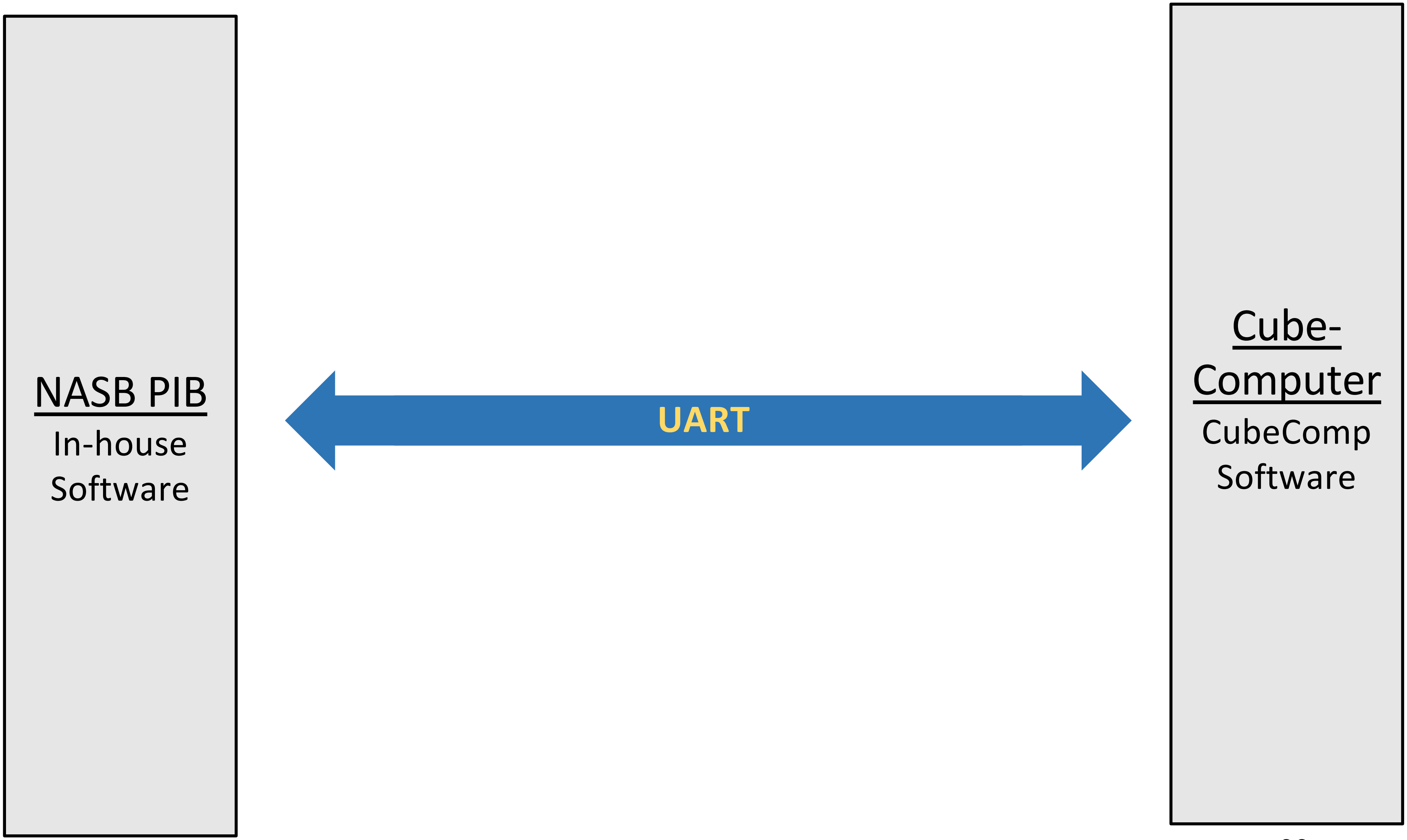
CubeComputer
Commmercial



UART



Communications



Communications

NASB PIB

In-house Software

Cube-Computer

CubeComp Software

Telemetry Request

0x1F	0x7F	TLM ID	0x1F	0xFF

Start-of-message identifier — End-of-message identifier

Telemetry Reply

0x1F	0x7F	TLM ID	TLM byte 0	...	0x1F	0xFF

Start-of-message identifier — End-of-message identifier

Telecommand

0x1F	0x7F	TC ID	TC data byte 0	...	0x1F	0xFF

Start-of-message identifier — End-of-message identifier

Telecommand Acknowledgement

0x1F	0x7F	TC ID	TC Error flag	0x1F	0xFF

Start-of-message identifier

TC Error flag:
0 = no error
1 = invalid TC ID
2 = incorrect length
3 = invalid parameters
4 = CRC error

End-of-message identifier

Software Implementation

Library: cubelib.mpy

- Commissioning Manual -> function
- Telemetry requests and telecommands
- Process all telemetry responses and telecommand acknowledgements

Main file: code.py

- Executes the commissioning sequence
- Implement algorithms in the commissioning manual using our home-built library

```python
def rwdemo():
    ADCSrunMode(1)
    setPowerControl(1,0,0,0,0,1,1,1,0,0)
    setWheelSpeed(2000,1000,3000)
    for i in range(60):
        if i==20:
            setWheelSpeed(1500,500,2500)
        getWheelSpeed()
        time.sleep(0.5)

    setWheelSpeed(0,0,0)
    setPowerControl(1,0,0,0,0,0,0,0,0,0)

def ratedemo():
    ADCSrunMode(1)
    setPowerControl(1,1,0,0,0,0,0,0,0,0)

    for i in range(10):
        print(getRateSensorRates())
```

Q&A