# 2025 NASA Student Launch (Payload): Operation Manual

Authors: Kyle Mahoney, Nathan Hardie, Donovan Dwight, Matthew Archibald, Neil Maldonado

Senior Design II

FAMU – FSU College of Engineering

# Table of Contents

# Project Overview

The NASA Student Launch Competition is a yearly event where university teams from across the country design, build, and fly high-powered rockets. An important part of the competition is to design a payload for the rocket. The requirements for the payload change each year. This year's payload is responsible for collecting flight data. This data includes flight information like velocity, apogee, environmental temperature, and flight time. Once the payload has collected all data, it much then transmits that data via radio signals. Lastly, the payload must house four STEMnauts, or model astronauts, and these STEMnauts must be safely secured within the payload during the rocket's flight. To meet all requirements for this payload mission, the team designed a small capsule that is used to house all necessary electronics. This capsule is then mounted to the rocket's nosecone during the flight launch. This paper will discuss the assembly and operation of this payload system.

# Components and Model Description

This section will provide an overview of all individual parts/assemblies for the payload system. This includes the payload's structural components, the electrical components, and the software.

## Structure

The payload's structural design consists of four custom parts: the capsule, the divider, the data collection tray, and the transmission tray. These custom parts were all 3D printed out of Nylon-12 using Selective Laser Sintering (SLS). For those with access to the FAMU – FSU AIAA: ZENITH Program's Microsoft Teams account, the STL files for these components can be found in "24 - 25 Documentation" > "Rocket Subsystems" > "Payload" > "CAD" > "REV4" > "STL".

### *Capsule*

The capsule serves as the central housing for the payload system. It acts as a protective shell that dissipates energy away from the sensitive electronics inside and it serves as the attachment point to the rocket's main body. The capsule is shown in Figure 1.

*Figure 1: SLS printed payload capsule.*

### *Divider*

The payload divider organizes and secures all the electronic boards within the capsule. Once installed into the capsule, it acts as a flat surface to mount electronics. The divider is shown in Figure 2.

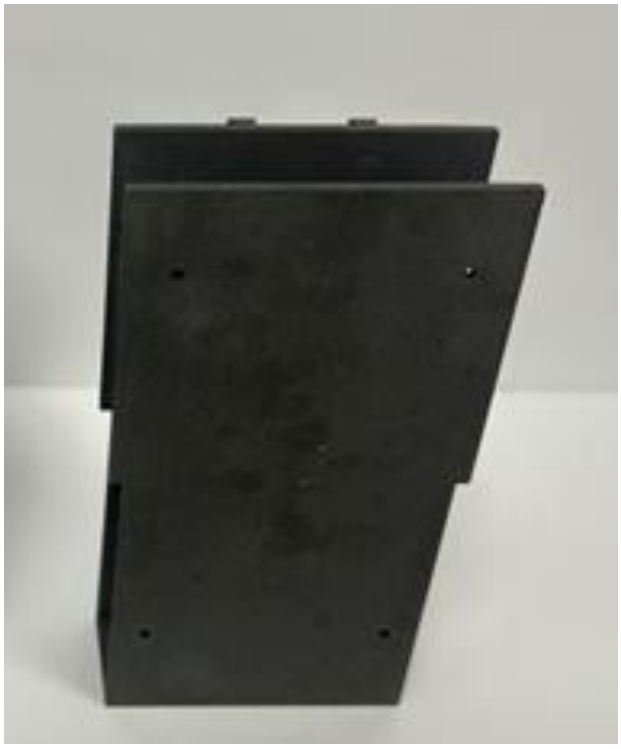

*Figure 2: SLS printed payload divider.*

### *Data Collection Tray*

The data collection tray is used to mount the data collection board. The data collection board will be discussed later in this report. This tray allows for easy attachment to the divider, streamlining the assembly process. The primary purpose of the data collection tray is to offer a secure and efficient way to install the board, making the entire setup process more manageable. Figure 3 shows the data collection board mounted to the data collection tray.

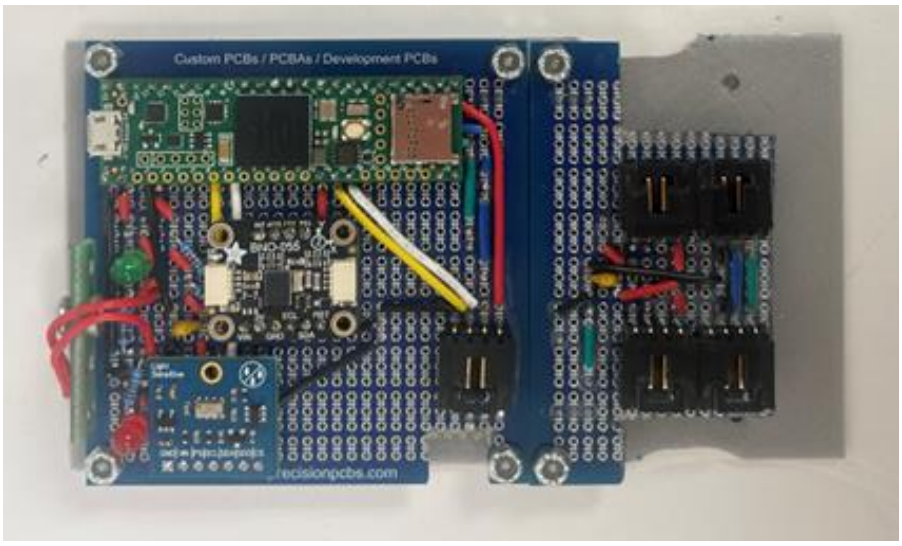


*Figure 3: 3D printed data collection tray (grey) with electronics board mounted on top.*

### *Transmission Tray*

The payload transmission tray operates in much the same way as the data collection tray, providing a dedicated mounting space for the data transmission board. The transmission board mounted to the transmission tray is shown in Figure 4.

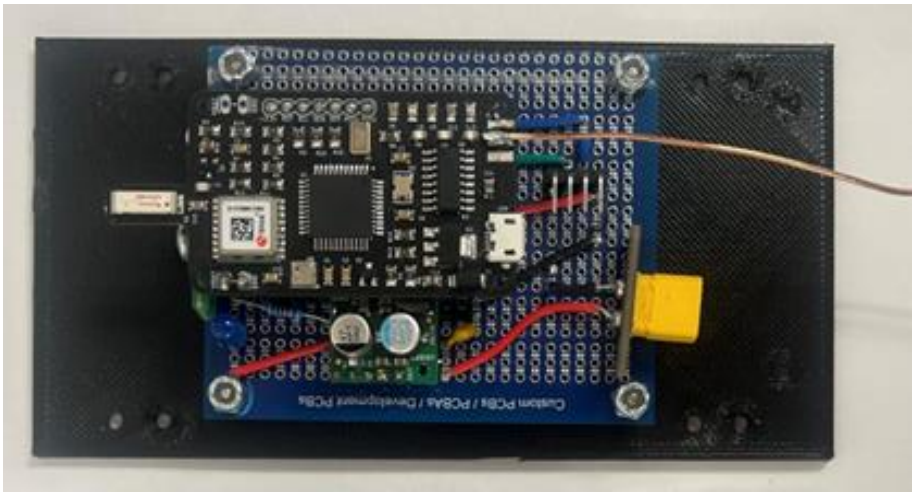

*Figure 4: 3D printed data transmission tray with electronics mounted on top.*

### *Commercial – Off – The – Self*

The payload's structural system contains several commercial – off – the – shelf (COTS) components used in the assembly process. Table 1 provides a list of all COTS components used to assemble the payload's structural system.

*Table 1: Commercial-off-the-shelf components for payload structure.*

| COTS Payload Components | | | | |
|---|---|---|---|---|
| Name | Part Number | Qty | Description | Manufacturer |
| Flat Head Screw | 90471A432 | 4 | #6-32 x 5/8” | McMaster |
| Pan Head Screw with lock washer | 91251A197 | 4 | #6-32 x 1/2” | McMaster |
| Nylon-Insert Locknut | 90631A007 | 4 | #6-32 | McMaster |
| Pan Head Screw | 91772A108 | 4 | #4-40 x 3/8” | McMaster |
| Nylon-Insert Locknut | 90631A005 | 4 | #4-40 | McMaster |
| Heat-set Inserts | 94459A427 | 4 | #6-32 | McMaster |
| Hex Standoffs | 91075A211 | 6 | #2-56 x 1/4” | McMaster |
| Hex Standoffs | 91075A512 | 4 | #2-56 x 3/16” | McMaster |
| Flat Head Screw | 91099A104 | 10 | #2-56 x 1/4” | McMaster |
| Self-Tapping Pan Head Screw | 90087A837 | 2 | #2-56 x 1/8” | McMaster |

# Electronics

The payload's electronical design consists of three main functions: data collection, data transmission, and STEMnauts security. Each of these functions requires various sensors and electronics to complete its task. This section will discuss the components used to complete the three main functions of the payload electronics system.

## *Data Collection*

The responsibility of collecting all flight data is consolidated into a single electronics board. The base of the board is may of a 3” x 3” three-hole perfboard. This perfboard contains a BNO055 IMU, MS5607 Altimeter, Teensy 4.1 Microcontroller, Molex KK Cables, and a Screw Switch. The BNO055 IMU is used to calculate the orientation and acceleration of the payload. The MS5607 Altimeter is used as the primary altitude, or apogee, detector. This altimeter also functions as our temperature sensor keeping us updated about any temperature changes within our payload. The Teensy 4.1 Microcontroller functions as the brain of the board, allowing for communication between all sensors. The Molex KK Cables are used to allow for inter-tray connections from one electronics board to another. This will be discussed more later in this report. Finally, there is the Arming Screw Switch that arms the board before launch. Additionally, there for two LED lights

used to indicate that the payload is powered, and armed. The data collection board is shown in Figure 6.

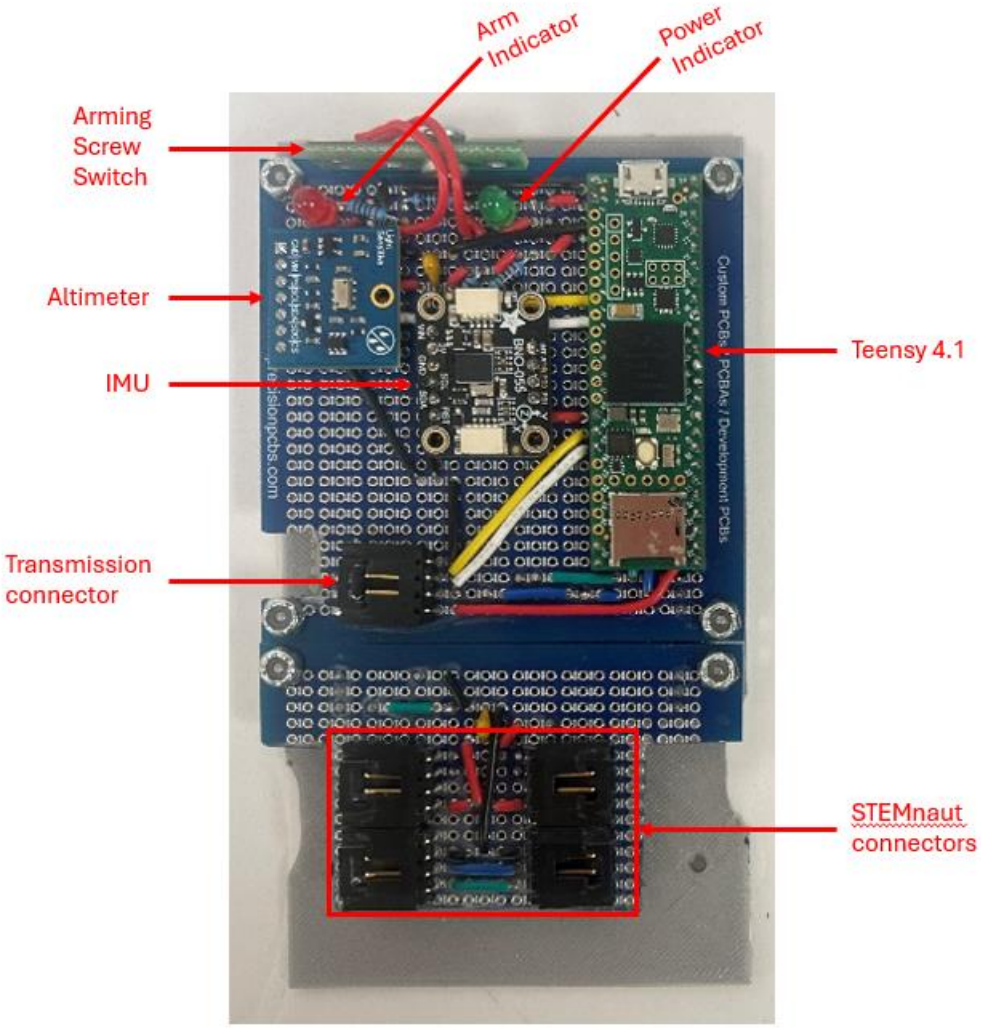


*Figure 5: Data collection board with labeled components.*

### *Data Transmission*

Once all data is collected, the data transmission board is responsible for transmitting the flight data back to the NASA ground station. The components of this board consist of a LightAPRS, Antenna, Power Screw Switch, and a Battery Connector. The LightAPRS is the Radio Frequency Module used to transmit flight data from the payload to the ground transceiver. The antenna functions as a "bridge" for the Radio Frequency Module to extend the connection from our RF module to the ground station transceiver. A Power Screw Switch is used to power the entire electronics system of the payload. Lastly, the battery connector is used to easily install a 2S Lithium Polymer Battery used to power our system. Additionally, this board contains a voltage regulator, and an LED light used to indicate the board is powered. The data transmission board is shown in Figure 6.

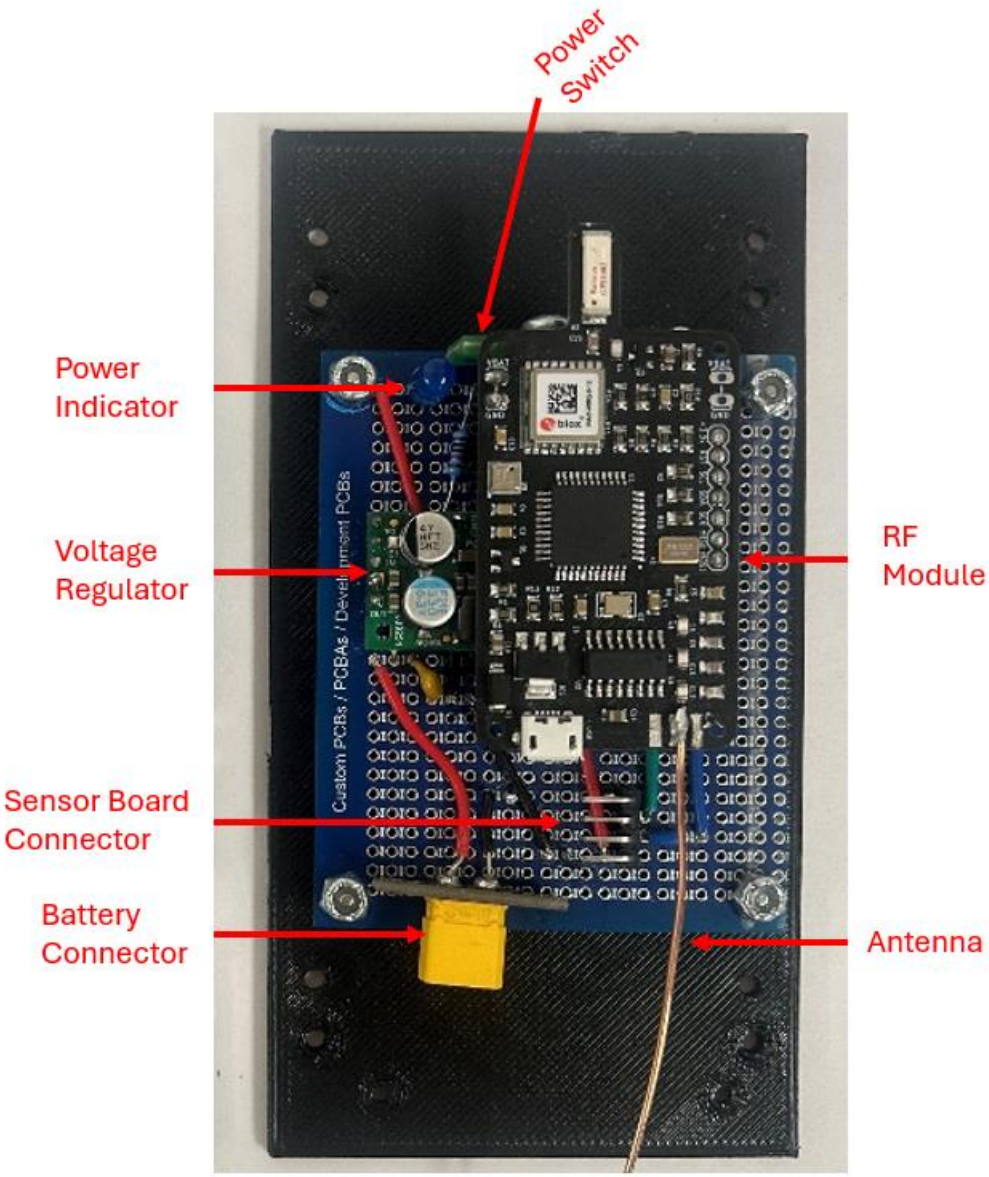


*Figure 6: Data transmission board with labeled components.*

### *STEMnauts*

The electronics behind the STEMnauts inside the payload are four FSM3000 IMU's. The purpose of the IMU is to track the orientation and acceleration of the STEMnauts throughout the flight. Each of our STEMnauts will be connected to our Data Collection board via Molex KK cables attached to each IMU shown in Figure 7.

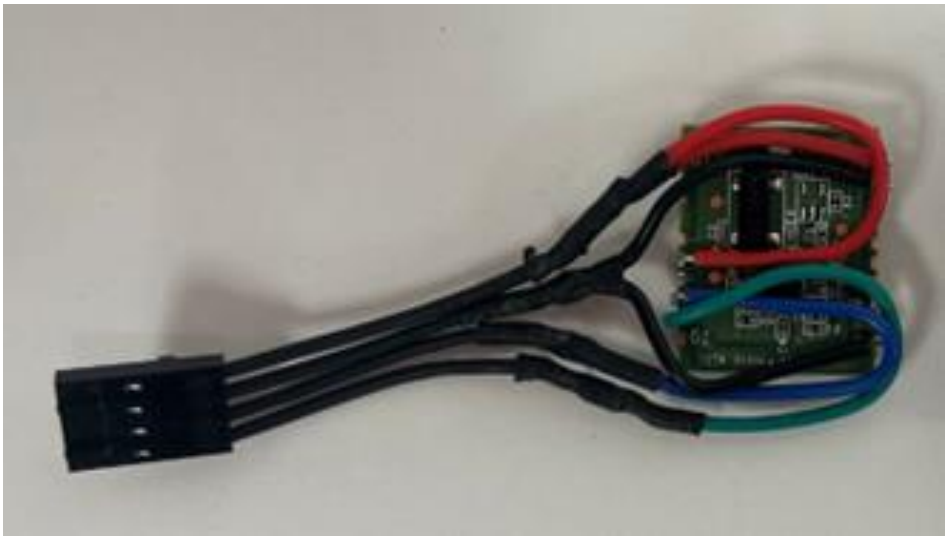

*Figure 7: IMU used for STEMnaut data collection.*

# Software

The payload software, PayloadOS, automatically performs the tasks of collecting, storing, processing, and transmitting the NASA required flight parameters. Additionally, it offers a command line interface (CLI), and hardware in the loop simulation (HIL).

## *Flight Operation*

The Software of our payload can be split into two parts The LightAPRS program and the Teensy 4.1 program which is our main program. The LightAPRS program contains the payload's transmitter, GPS, Secondary altimeter, and the analog-to-digital converter. All these programs were able to be configured through code provided by the LightAPRS manufacturers, QRP Labs. We also had to develop a way to communicate between the LightAPRS program and the Teensy 4.1 program to have a seamless interaction for Data Collection and Data Transmission. The LightAPRS also has a USB port that allows for debugging and testing efforts. The Teensy 4.1 Program, or how we know it as the PayloadOS, is the primary microcontroller used during flight. The PayloadOS is mainly used in gathering the data from all the payload sensors, storing the data onto the microSD car, analyzing post-flight data, and creating the transmission packets that will be sent back to the ground station transceiver to be reviewed. It is also important that the program can recognize important flight milestones such as flight launch, apogee, and landing for the data required by the competition and to ensure the data transmission occurs at the correct time. The PayloadOS functions as a finite state machine that runs on an 8Hz clock whereas it updates its state it will perform its associated task once per cycle. It starts by initializing itself and all the sensors. It then enters a Startup state that is followed immediately by a Standby state where it stays idle until commanded to change states. During Standby state, the program runs a series of checks and diagnostics to confirm the payload is ready to fly. This will then lead the payload into an Armed state where the payload starts logging telemetry and check the altimeters to check if the

rocket has launched. Once a launch is detected the payload enters a Flight state where it continues to collect and log telemetry while also checking for a landing event. Once the landing event occurs the payload moves into a Processing state where the log files created are then analyzed and are computed to the eight parameters required by the handbook to be transmitted. When this is done the payload enters the Transmit state where it transmits the eight parameters into packets and transmits them back to the ground station transceiver. Once this is done the payload returns to the Standby state waiting for recovery. Figure 8 shows the state diagram.

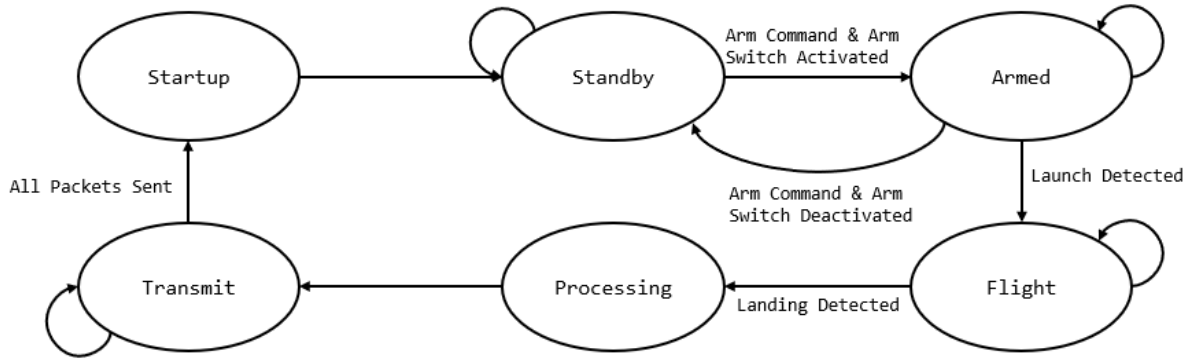


*Figure 8: State diagram for payload software.*

### *Command Line Interface*

The PayloadOS CLI is used to configure the payload, run preflight checks, control HIL simulations, and perform troubleshooting. To open the CLI, a micro USB cable is inserted into the port in the sensor compartment. Any program which provides an interface for serial communication can then be used to communicate with the CLI. The serial program should be configured for 115200 baud. A few examples of serial communication programs are PuTTy, Arduino Serial Monitor, & Tera Term.

Upon startup, the startup message sequence will be displayed in the CLI. This sequence will be printed if the payload resets or is power cycled.


*Figure 9 Display after startup*

Commands are indicated by the > character which must be present at the start of each command. After the > character, the name of the command is written, followed by the arguments for the command separated by spaces.

>commandName [arg1 arg2 …]

There are five types of supported command arguments: words, strings, signed integers, unsigned integers, and floating-point numbers. A word argument can be interpreted from any string of characters not containing a space. String arguments are enclosed by “” marks and can contain spaces. Signed integer arguments can be interpreted from numerical characters with an optional – sign. Unsigned integers may not contain a – sign. Additionally, binary and hexadecimal format is also supported for unsigned arguments. Floating point arguments can be interpreted from numerical characters with a . and – sign. Each argument type has an identification character associated with it. This is used by PayloadOS to communicate argument requirements.

| **Argument Type** | **Identification Character** | **Valid Examples** | **Invalid Examples** |
|---|---|---|---|
| Word | w | >command foo bar -1.25 | >command “foo” |
| String | s | >command “foo bar” “foo” | >command foo bar |
| Signed | i | >command -2 5 | >command 2.0 0xFF |
| Unsigned | u | >command 4 0b010 0xA2 | >command 3.6 -4 |
| Float | f | >command 2.5 5 -3.0 | >command 0xA foo |

*Figure 10 Table of valid arguments*

A set of commands are available for each of the payload’s states. Additionally, a set of general commands are available in all states, as well as the HIL control commands. The list of

availablecommands in the current state can be accessed with the ‘commands’ command. The arguments for each command are listed in brackets after the name of the command.

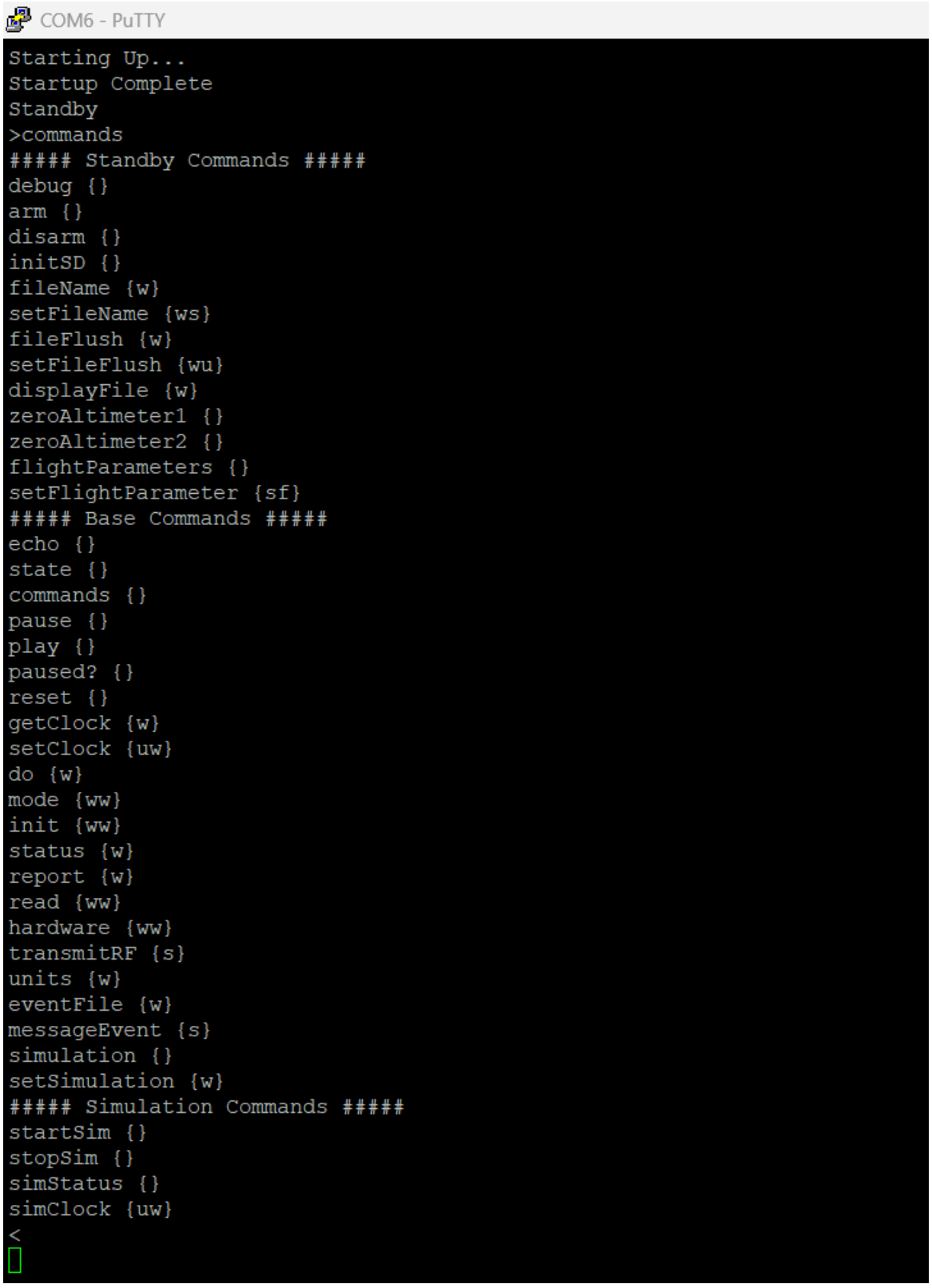

*Figure 11 List of commands*

### *Hardware in the Loop*

The payload supports hardware in the loop simulation using its SD card. Telemetry files from past flights can be read from and then used for sensor values.

To use the HIL, first configure the peripherals into backdoor mode. This is accomplished with the mode command: >mode all backdoor. After putting the peripherals into backdoor mode, they need to be re-initialized with the init command: >init all do. To select a file to use for the simulation, use the command: >setFileName dataSim “filename.txt”. The simulation can be started with the command: >startSim. At this point the simulation will start reading from the telemetry file and will automatically stop when it reaches the end of the file. The command: >simState can be used to check if the simulation is still running or not. The command: >stopSim can be used to stop the simulation before it reaches the end of the file.

# Assembly and Integration

To begin the assembly of the payload system, the user must first assemble the two electronic boards. Two three-hole perf boards are used as the base of the board. Wires, sensors, and connectors are all secured to the perfboards via solder joints. The wiring diagram of the sensor and transmission boards are shown in Figure 12.

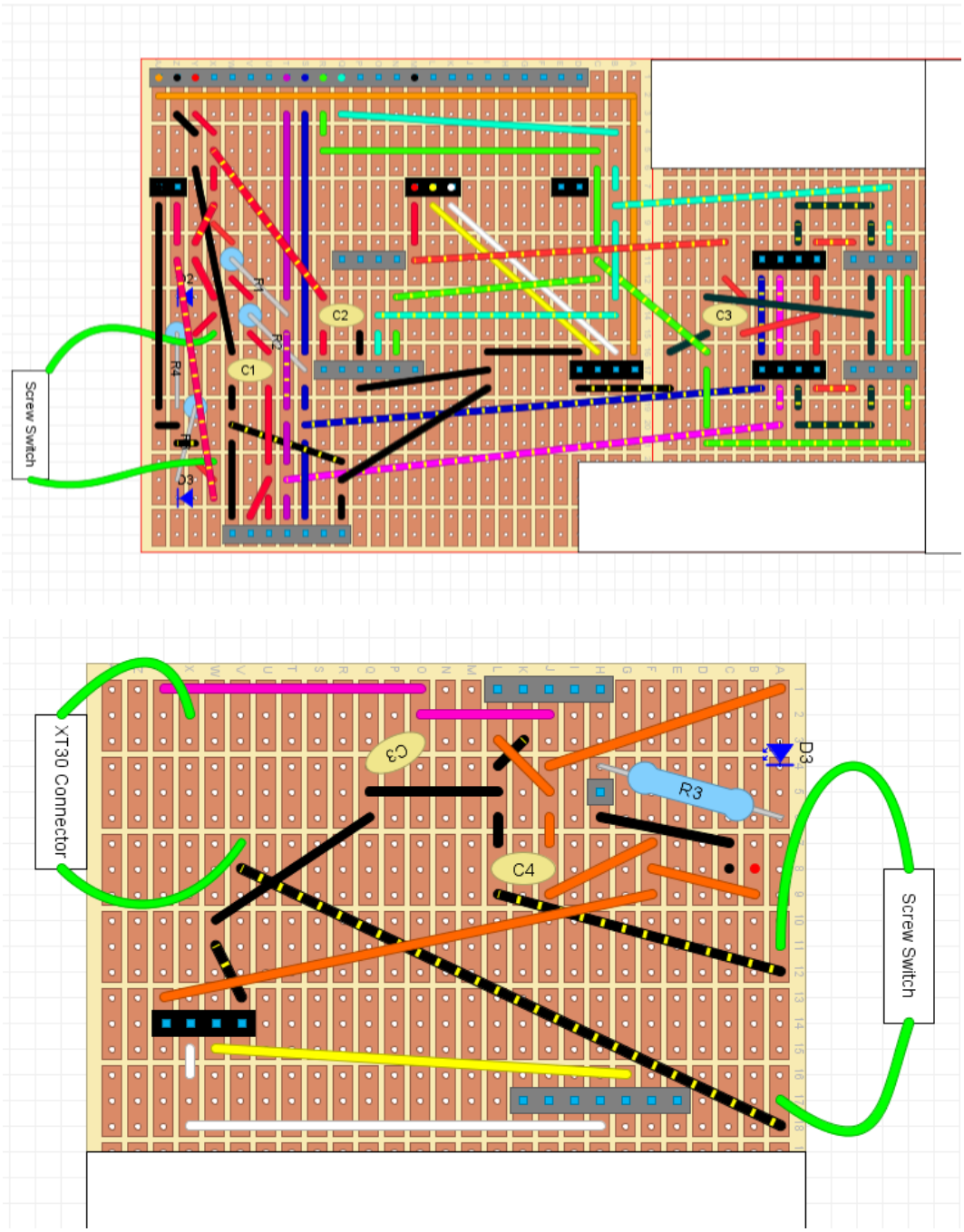


*Figure 12: Wiring diagram for data collection board (top) and data transmission board (bottom).*

The final assembly of the two electronic boards are shown in Figure 13.

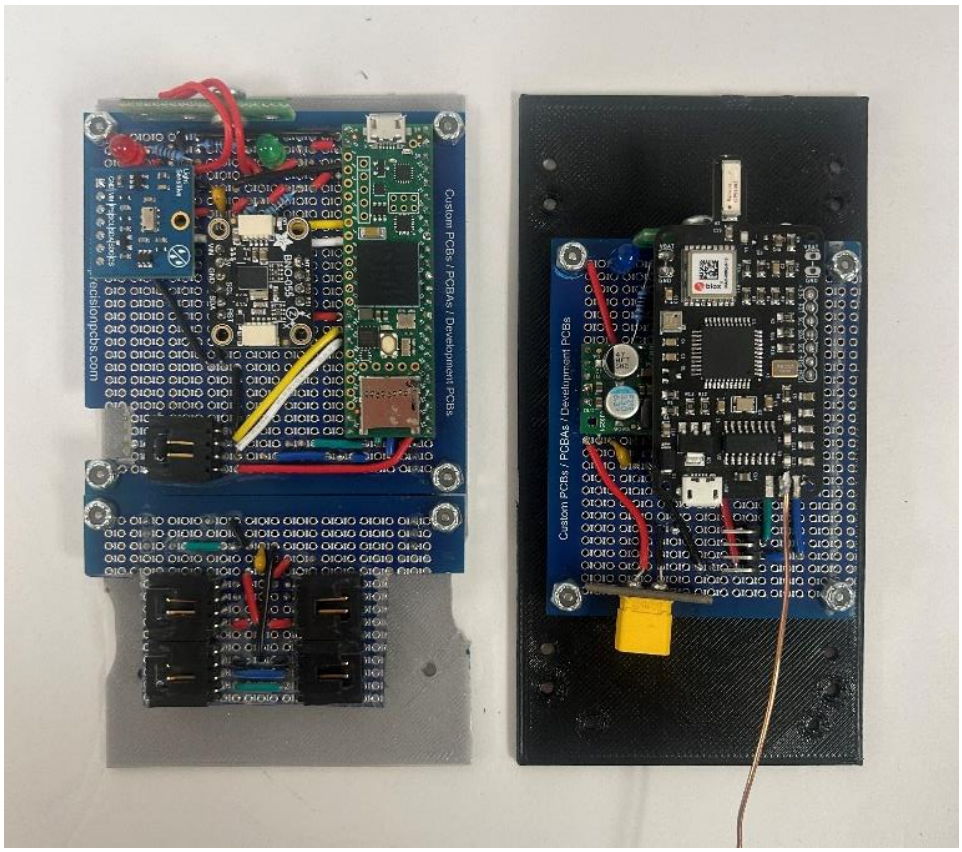

*Figure 13: Data transmission (left) and data collection (right) boards fully assembled.*

The data collection board is then mounted to its respective tray. Holes are spot drilled through the tray, aligning with the mounting holes on the data collection board. #2-56 x ¼", male-female hex standoffs are used to secure the board to the tray. Flat-head screws are used to mount the standoffs to the tray and a #2-56 lock nut is used to lock the board down on the standoffs. Figure 14 shows the data collection board mounted to the tray.

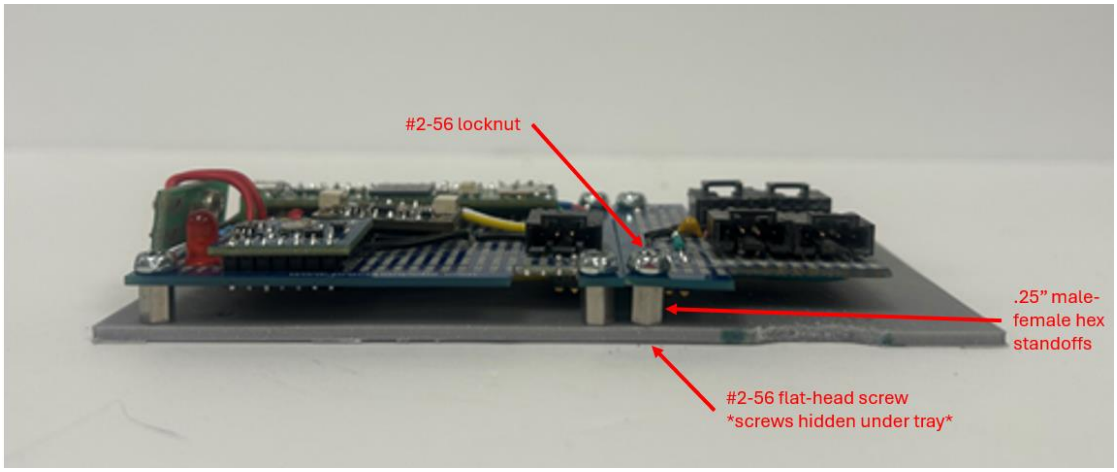


*Figure 14: Data collection board installed onto tray.*

The same process is used to mount the transmission board to the transmission tray; however, 3/16” long standoffs are used instead of the 1/4". The transmission tray assembly is shown in Figure 15.

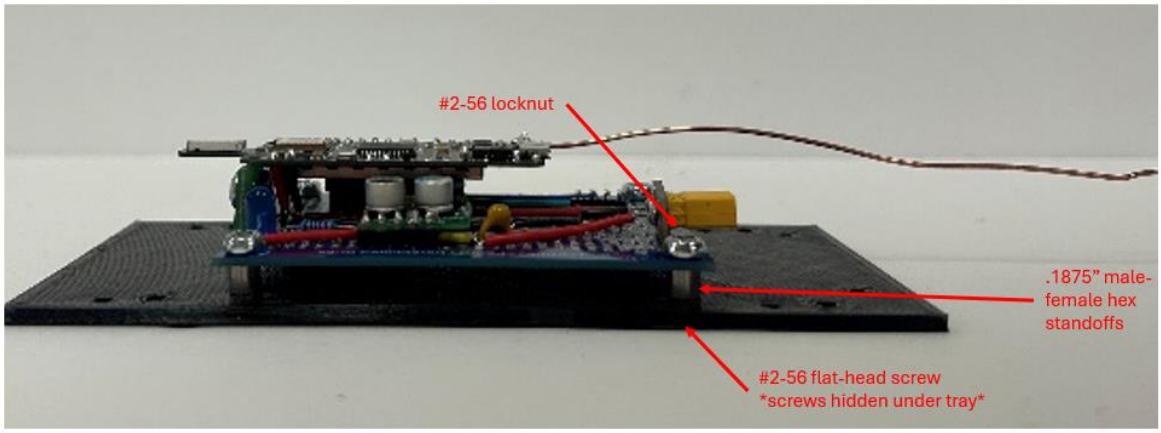


*Figure 15: Data transmission board installed onto tray.*

Once both trays are assembled, #4-40 screws with a locknut are used to secure the trays to the divider. Figure 16 shows the connection between the trays and the divider.

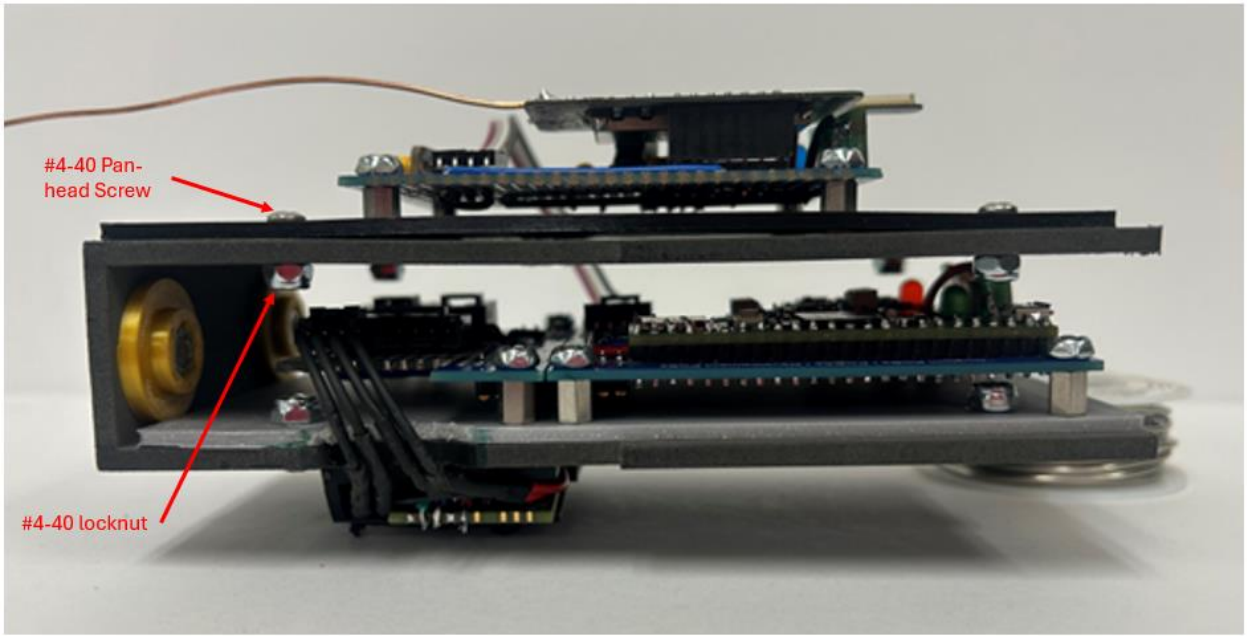


*Figure 16: Electronic trays installed onto divider.*

The STEMnaut IMU’s are secured into the backpacks of the 3D printed STEMnaut housings and the STEMnauts are slid onto the dovetail of the divider being locked into place using a #2-56 self-tapping screw shown in Figure 17.

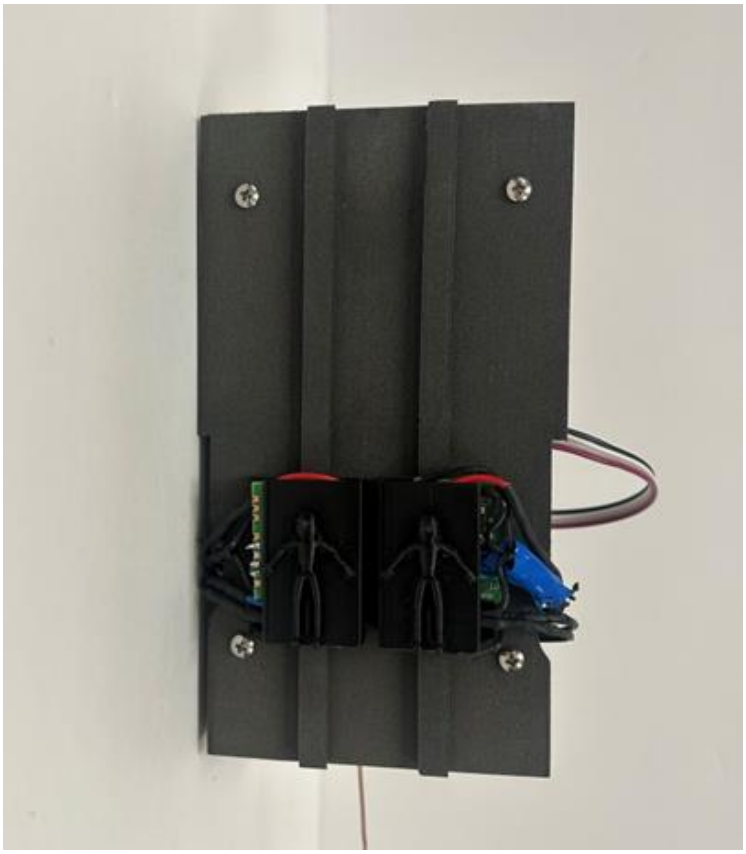

*Figure 17: STEMnauts mounted to divder.*

With all electronics mounted to the divider, inter-tray connections can be made. The male Molex KK cable connectors that are attached to the STEMnaut IMU's can wrap around the wall of the divider and connect to the four female connectors at the base of the data collection board. This is the same for the inter-tray connector that is used to connect the transmission board to the collection board. These connections are shown previously in Figure 17.

When the divider has all electronics secured to it, that completes the assembly of the payload's electronics subsystem. The complete assembly of the electronics system is shown in Figure 18.

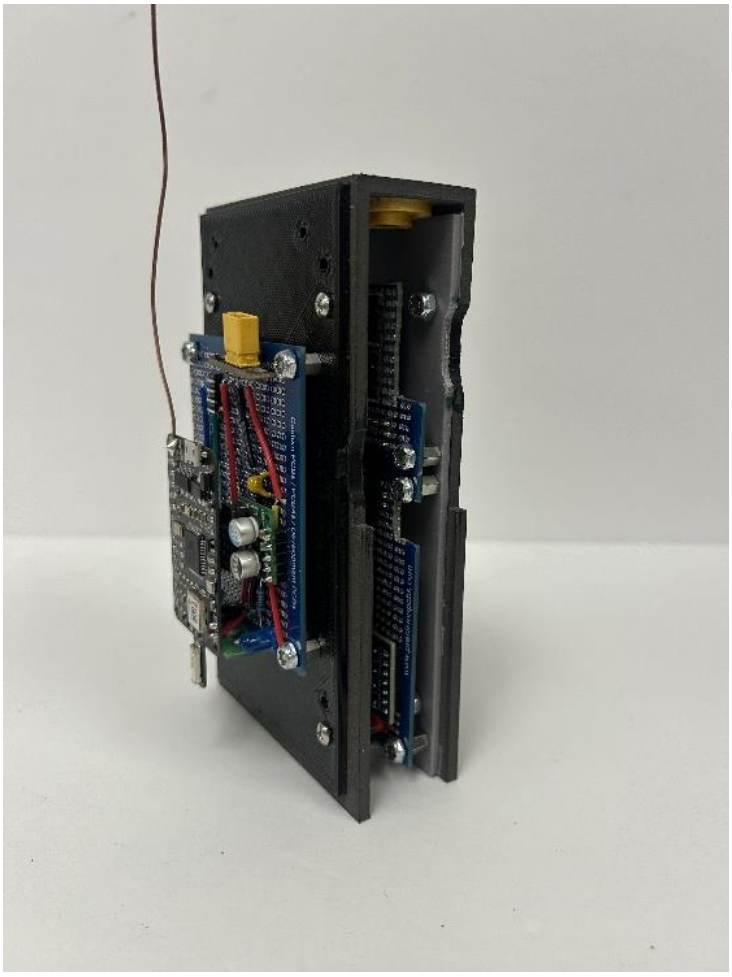

*Figure 18: Payload electronics system fully assembled.*

The entire electronics system is then slid into the capsule. When installing the electronics system, ensure that the walls of the divider align with the notches in the inner wall of the capsule. The electronics system is then secured at the bottom of the capsule using #6-32 flat head screws shown in Figure 19.

*Figure 19: Mounting screws for payload electronics system.*

When the electronics system is installed into the capsule, that completes the full assembly of the payload. The full construction of the payload is shown in Figure 20.

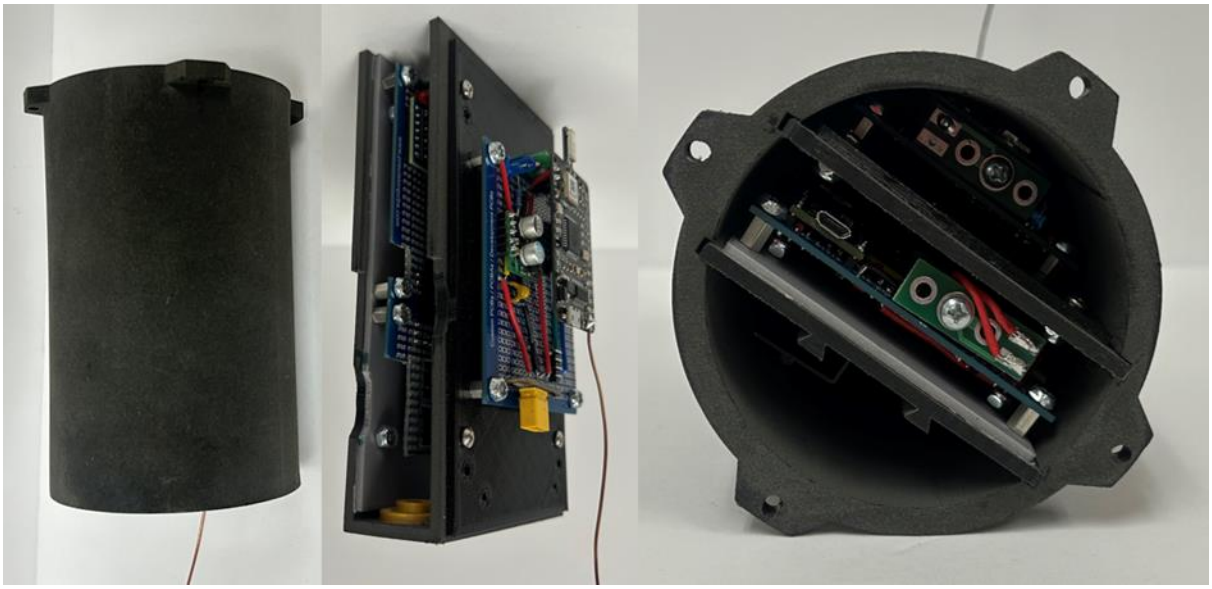

*Figure 20: Full payload construction.*

To integrate the payload assembly with the rocket, the mounting holes on the payload's flanges are aligned with #6-32 heat set inserts that are inside of the rocket's nosecone. #6-32 x 1/2" screws with built-in lock washers are installed through the flanges in the capsule and into the heat-set inserts in the nosecone. Figure 21 shows the mounting between the payload capsule and the nosecone.

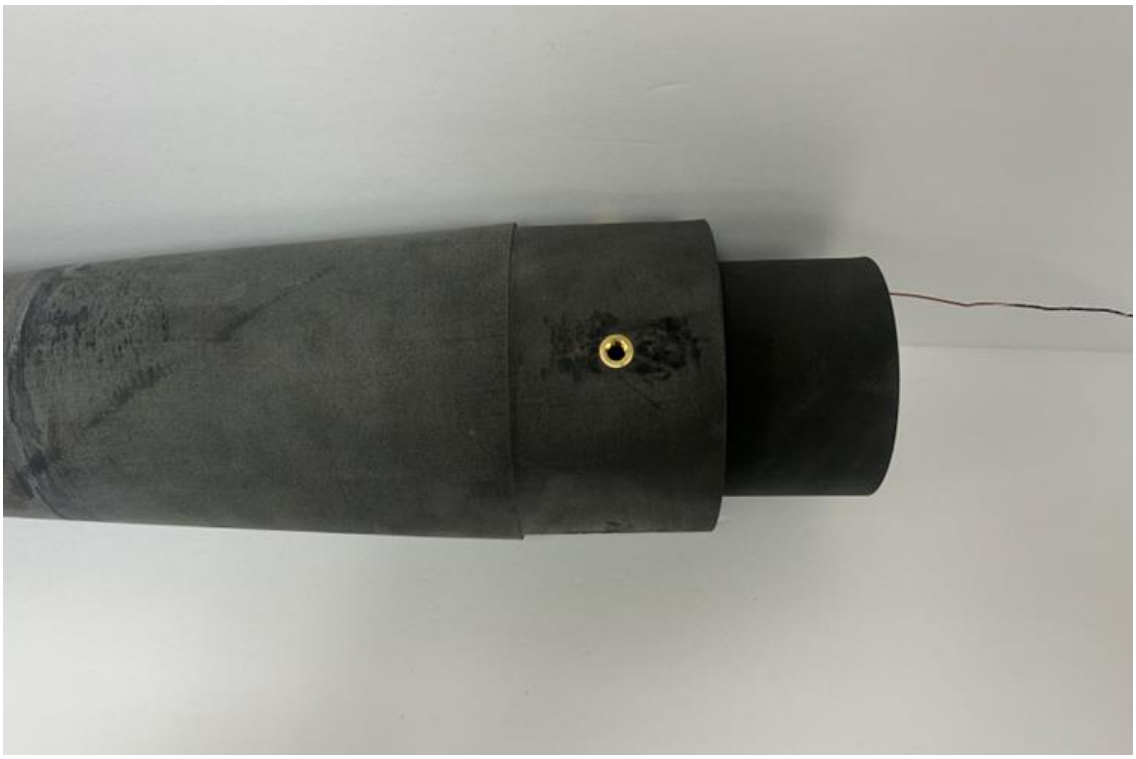

*Figure 21: Payload integration into rocket nosecone.*

# Operation

This section will discuss the operation of the payload system. Note that the steps below assume that the payload's electronics system has been fully assembled. See previous section to reference steps for assembling the electronics system.

1. Connect battery
2. Grab towards the top of the chamber divider and align the tabs with the slots in the capsule and proceed to slide it into the capsule making sure to allow the antenna to go through the hole at the bottom of the capsule first
3. Once the divider is fully slid into the capsule take 2 screws and use them to fasten the divider to the capsule and use one hand to prevent the divider still while fastening
4. Plug your computer into the micro-USB port
    4.1. If testing without battery use a regular USB cable
    4.2. If battery is installed, you shall use the modified cable with red tape on it to prevent damage to the electronics and computer
5. Arm transmitter tray board by screwing switch to the right until an LED turns on
6. On Computer open any serial communication operation software of your choice (Example: VS Code, Arduino, etc.)
7. Determine Mode: In appropriate space type the words (Mode)
8. After Proceed to type the words (Mode all get) to display all the Modes

9. When the modes have been displayed configure everything to hardware mode for flight
10. To check if the sensors are initialized type (Init all get)
11. If not, type (Init all do)
12. Then type (Arm). This arms the payload software
13. Then screw the screw switch on the data collection board to the right until you see an LED illuminate. This arms the payload hardware
14. A message will pop up on the computer stating (Payload is armed)
15. The Payload is Armed and Collecting Data
16. Fasten the payload to the nosecone using screws going through the flanges

# Troubleshooting

When assembling and operating the payload system, issues can arise that induce a detrimental effect on the system's performance. Several common problems with the payload system are misalignment or interference between parts, inability to power/arm the electronics system, or inaccurate data collection.

## Misalignment or Interference between Components

When assembling the payload structure, if components are misaligned or interfering with other components, there are several steps to take to troubleshoot this issue. Firstly, if the holes used to mount the data collection and transmission tray to the divider are not aligning, ensure that the correct tray was used to mount different electronic boards. The data transmission tray and the data collection tray are NOT the same. Each tray must be used to mount its corresponding board. Before mounting boards to the trays, it helps to align the trays with the divider.

If the electronics system is not fitting into the capsule, ensure inter-tray connection are not getting bunched up on the inner wall of the divider. Furthermore, ensure that the RF module's antenna is not getting caught at the bottom of the divider. The antenna should easily fit into the hole at the base of the capsule's body. It the electronic system is not fitting, DO NOT force it. This can damage the antenna and wiring connections. The system should fit into the capsule snug; however, easily.

If the payload system is not aligning with the rocket nosecone, ensure no debris is left inside the nosecone. Also ensure that the divider is sitting flush, or under-flush, with the top of the capsule.

## Inability To Power/Arm Electronics

If the payload's electronics are unable to be powered or armed, there are several steps to take to troubleshoot this issue. First, start by ensuring that the battery is fully charged. If the battery is charged, check each solder connection of the boards. Common areas in which solder connections can be damaged is the connection to the STEMnauts, antenna, or screw switches.

## Inability To Collect Accurate Data

Each sensor can be polled for its status using the >status sensorName command. The status command indicates if each sensor is initialized, responsive, ready, and go. If a sensor is not initialized, re-initialize it. If it is not responsive, this suggests a connectivity issue. Check that connection cables are inserted correctly. If the sensor is not ready, it may be stuck or not yet fully initialized. Wait for it to finish initializing, and if the issue persists power cycle the device. If a sensor is not go, its calibration is not complete. This only applies to the IMUs. Moving the payload around can help to calibrate the IMUs more quickly.

If the sensors are working, but data is bad, the >report sensorName command can be used to see calibrations reports, live values, and errors for each sensor. The exact contents of the report vary for each sensor but it may give clues as to why data is inaccurate. Check that the IMUs are not close to any permanent magnets, as these can disturb their magnetometers. Additionally, exposure to direct sunlight can cause the altimeters to give erroneous results.