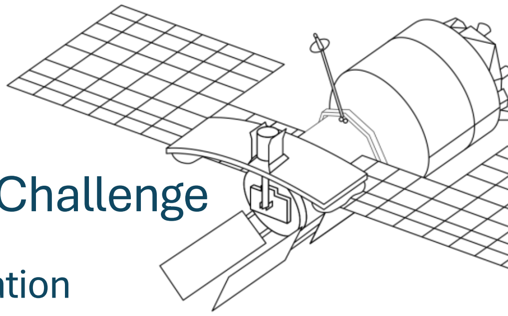


# Project ICARUS - Communications Reliability Challenge

## Candidate Technical Assessment – Dataset-Based Evaluation

### 1. Overview

Reliable communication is the lifeline of any CubeSat mission. With limited power, bandwidth, and hardware resources, small errors in modem design or coding can mean the difference between a successful downlink and complete data loss.

This challenge simulates those realities through a staged dataset of received signals. Each phase introduces controlled impairments—timing errors, noise calibration traps, coding requirements, and Doppler shifts—that mirror conditions encountered in low Earth orbit.

Candidates must design and refine receiver algorithms to recover the original bitstreams and demonstrate robust performance against defined thresholds. The process is intentionally iterative: naive solutions will fail, requiring careful analysis, algorithmic insight, and systematic debugging.

The objective is not to reproduce textbook formulas, but to think and work as an engineer—test, fail, diagnose, and converge on reliable solutions suitable for a CubeSat communications subsystem.

### 2. Learning Objectives

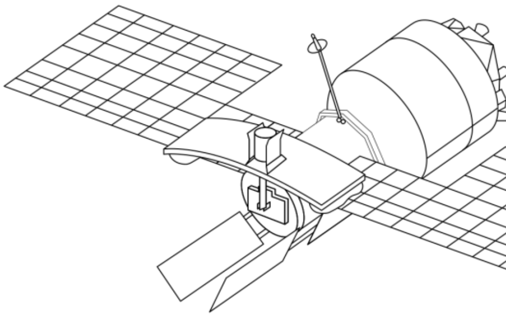
Participants will demonstrate:

- Implementation of **digital modulation and demodulation** (BPSK).
- Accurate **noise modeling and SNR calibration**.
- Practical use of **error correction coding and decoding**.
- Techniques for **synchronization and Doppler compensation**.
- **Engineering problem-solving**: diagnose, analyze, and refine.

### 3. Dataset Structure

The dataset is organized into four phases, each introducing new impairments.

```
cubesat_dataset/
  phase1_timing/      # Symbol timing offset
  phase2_snr/         # SNR calibration accuracy
  phase3_coding/      # Error correction required
  phase4_doppler/     # Doppler frequency offset
```



Each sample directory contains:

- rx.npy – received waveform (NumPy array, baseband samples).
- meta.json – metadata (SNR, impairments, ground-truth bits).

Dataset Download Link – Click here

## 4. Candidate Task

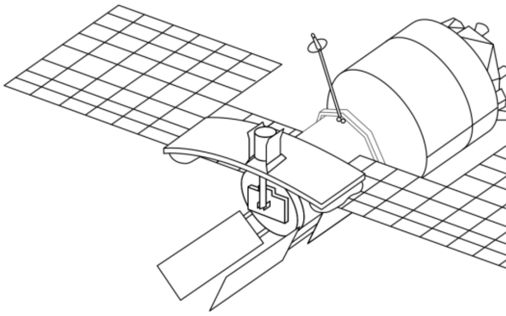
1. Implement a receiver to recover bitstreams from rx.npy.
2. Save decoded results in the same directory as:
3. decoded_bits.npy

(NumPy array of 0/1 integers).

4. Preserve the dataset folder hierarchy.
5. Run the evaluation script to generate BER/FER metrics.

## 5. Challenge Phases

| Phase | Impairment Introduced | Naive Result | Required Insight | Performance Threshold |
|---|---|---|---|---|
| 1 | Symbol timing offset | BER ≈ 20% | Matched filtering & timing recovery | BER ≤ $1\times10^{-2}$ @ 10 dB |
| 2 | SNR scaling inconsistency | BER curve off by ~3 dB | Correct signal/noise power calibration | Curve within ±2 dB of theory |
| 3 | Error correction (RS, Conv.) | High BER/FER | Implement RS(15,11) & Viterbi decoding | RS: FER ≤ $1\times10^{-3}$ @ 12 dBConv.: BER ≤ $1\times10^{-4}$ @ 8 dB |
| 4 | Doppler frequency shift | BER collapse | Frequency offset estimation & correction | BER ≤ $1\times10^{-3}$ @ 15 dB |



## 7. Deliverables

Participants must provide the following deliverables:

1. **Implementation Code**
    - A complete receiver implementation (Python, using NumPy/SciPy).
    - Code must be maintained in a **GitHub repository**.
    - The GitHub repository link must be included in the documentation.
2. **Documentation**
    - A short write-up describing:
        - **Approach**: algorithms and methods implemented.
        - **Challenges**: major difficulties encountered and how they were addressed.
        - **Implementation Notes**: details of design decisions, assumptions, and limitations.

    This documentation is **mandatory** and should be included with the code repository.

3. **Optional Technical Report**
    - A **2–3 page technical report** summarizing:
        - Design approach and justification.
        - Key challenges and lessons learned.
        - BER/FER performance results and interpretation.
    - Submission of this report is optional but encouraged.

## 7. Constraints

- Allowed: NumPy, SciPy, FFTs, filtering utilities.
- Prohibited: Communications toolboxes (MATLAB, GNURadio, IT++).
- All algorithms must be implemented by the participant.

## 8. Expected Outcomes

Candidates completing the challenge will:

- Gain hands-on experience in **CubeSat communication subsystem design**.
- Demonstrate ability to **iterate, test, and refine under constraints**.
- Show engineering maturity by bridging **theory with practical performance**.

## 9. Performance Plots (Candidate Requirement)

Final submissions must include at least:

- **BER vs SNR** curves for uncoded vs coded systems.
- **Constellation diagrams** at representative SNRs.
- **Doppler effect compensation plots** (before/after correction).