

# ARLISS2025大会報告書

提出日:2025年11月21日

## チーム情報

CanSatチーム名	東京大学 中須賀・船瀬・五十里研究室チーム名 SEIKOKU
CanSatチーム 代表者情報	溝口礼 mizoguchi-rei222@g.ecc.u-tokyo.ac.jp, 08044129625
UNISEC団体名	東京大学中須賀船瀬五十里研究室Avispa
UNISEC団体 学生代表	西川諒
責任教員	中須賀真一 nakasuka@space.t.u-tokyo.ac.jp, 03-5841-6590
CanSatクラス	Open Class

# 目次

## 目次

### 第1章 ミッション定義

#### 第1.1節 ミッションステートメント

##### 第1.1.1項 ミッション背景

##### 第1.1.2項 ミッションステートメント

#### 第1.2節 ミッション内容

#### 第1.3節 サクセスクライテリア

### 第2章 システム要求

#### 第2.1節 レギュレーションを満たすためのシステム要求

#### 第2.2節 ミッションを達成するためのシステム要求

### 第3章 システム試験項目の設定

#### 第3.2節 ミッションを達成するためのシステム試験項目

### 第4章 システム仕様

#### 第4.1節 機体概観

##### 第4.1.1項 機体概観

##### 第4.1.2項 機体諸元

###### 第4.1.1.1目 電力

###### 第4.1.1.2目 質量

###### 第4.1.1.3目 寸法

#### 第4.2節 機体機構

##### 第4.2.1項 パラシュート・エンベロープ

##### 第4.2.2項 起立機構

##### 第4.2.3項 フライト機構

#### 第4.3節 搭載機器

##### 第4.3.1項 システム図と使用コンポーネント

###### 第4.3.1.1目 システム図

###### 第4.3.1.1目 電子系 / 動力系

###### 第4.3.1.3目 構造系

##### 第4.3.2項 バッテリー

#### 第4.4節 アルゴリズム

#### 第4.5節 製作時に使用した機材・サービス・ツール

##### 第4.5.1項 製作時に使用した機材・サービス

##### 第4.5.2項 その他設計制作で利用したツール・アプリケーション

###### 第4.5.2.1目 コミュニケーション

###### 第4.5.2.2目 ドキュメンテーション

###### 第4.5.2.3目 構造

###### 第4.5.2.4目 電装

###### 第4.5.2.5目 制御

### 第5章 システム試験

#### 第5.1節 レギュレーションを満たすためのシステム試験

##### (RV1) 質量試験

##### (RV2) 機体収納試験

[\(RV3\)GPSデータダウンリンク試験](#)

[\(RV4\)準静的荷重試験](#)

[\(RV5\)振動試験](#)

[\(RV6\)分離衝撃試験](#)

[\(RV7\)パラシュート性能試験](#)

[\(RV8\)開傘衝撃試験](#)

[\(RV9\)無線性能試験](#)

[\(CRV1\)End-to-end試験](#)

[\(CRV2\)制御履歴レポート作成](#)

[\(FRV1\)FAA機体申請書](#)

#### [第5.2節 ミッションを達成するためのシステム試験](#)

[\(MV1\)機体放出試験](#)

[\(MV2\)放出判定試験](#)

[\(MV3\)着地衝撃試験](#)

[\(MV4\)着地判定試験](#)

[\(MV5\)起立展開試験](#)

[\(MV6\)腕展開試験](#)

[\(MV7\)機体離陸試験](#)

[\(MV8\)飛行性能確認試験](#)

[\(MV9\)ログ試験](#)

[\(MV10\)GPS航法飛行試験](#)

[\(MV11\)機体着陸試験](#)

[\(MV12\)ゴール検知試験](#)

[\(MV13\)画像航法試験](#)

[\(MV14\)航続距離検証試験](#)

#### [第6章 工程管理](#)

#### [第7章 大会結果報告](#)

##### [第7.1節 目的](#)

##### [第7.2節 結果](#)

[第7.2.1項 投下1回目](#)

[第7.2.3項 投下2回目](#)

[第7.2.3項 投下3回目](#)

[第7.2.4項 閉会式後飛行試験](#)

##### [第7.3節 考察](#)

##### [第7.3.1項 取得データ](#)

[第7.3.1.1目 投下1回目](#)

[第7.3.1.2目 投下2回目](#)

[第7.3.1.3目 投下3回目](#)

[第7.3.1.4目 閉会式後飛行試験](#)

##### [第7.3.2項 故障個所・原因解析](#)

[第7.3.2.1目 投下1回目](#)

[第7.3.2.2目 投下2回目](#)

[第7.3.2.3目 投下3回目](#)

#### [第8章 まとめ](#)

[第8.1節 工夫・努力した点](#)

[第8.2節 成果](#)

[第8.2.1項 構造班](#)

[第8.2.2項 電装班](#)

[第8.2.3項 制御班](#)

[第8.2.4項 マネジメント](#)

[第8.3節 課題点](#)

[第8.3.1項 構造班](#)

[第8.3.2項 電装班](#)

[第8.3.3項 制御班](#)

[第8.3.4項 マネジメント](#)

[第8.3節 今後の展望](#)

# 第1章 ミッション定義

## 第1.1節 ミッションステートメント

### ドローンによるピンポイント誘導

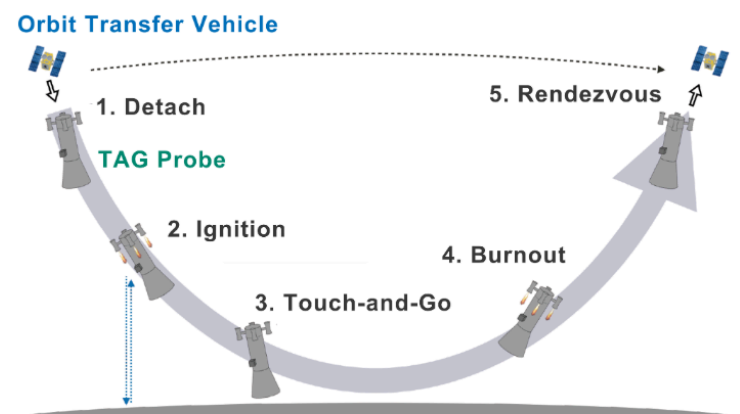
#### 第1.1.1項 ミッション背景

小惑星探査機「はやぶさ」および「はやぶさ2」では、小惑星表面からのサンプルリターンを目的として、探査機本体によるタッチダウンが実施された。これにより、貴重な科学的成果が得られた一方で、タッチダウンには熱的損傷や衝撃による機体損傷のリスクが伴い、探査機の安全性に大きな課題があることも明らかとなった。

さらに、小惑星表面は凹凸が激しく、十分に安全な着陸地点は限られている。このような制約の中で、高精度な着陸を行うには、より小型で機動性の高い機体の方が有利であると考えられる。

これらの背景を踏まえ、近年では探査機本体（親機）がタッチダウンを行うのではなく、小型探査機（子機）を親機から分離し、子機がタッチダウンを行うという新たな探査方式が検討されている。この方式により、探査機本体へのリスクを最小限に抑えたサンプル採取が可能になることが期待されている。

この方式の実現にあたって、小型かつ軽量でありながら、高精度な自律飛行位置制御を行える技術が必要不可欠である。



出典:JAXA宇宙科学研究所 Astrodynamics and Space Systems Laboratory

<https://assl.isas.jaxa.jp/vision.html>

## 第1.1.2項 ミッションステートメント

上記の背景を踏まえて、本ミッションのミッションステートメントを以下のように定義する。

「ドローンによるピンポイント誘導」

本ミッションでは、小惑星探査を念頭に置き、ソフトランディングと、ドローンによる目標地点へのピンポイント誘導を目指す。ピンポイント誘導の定義は、フルサクセスとして目標地点から「5m以内」、エクストラサクセスとして「目標地点に接触した状態での着陸」と設定した。フルサクセスの定義はGPSの誤差を参考にした。

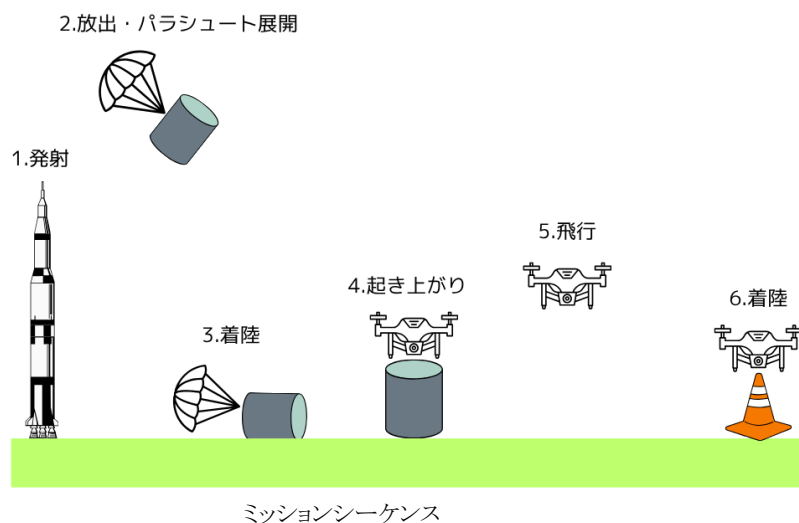
超小型機体による、ソフトランディングからの離陸姿勢への移行と自律飛行制御による目標地点への移動を実証することで、今後の惑星探査ミッションの可能性を広げることが期待される。これにより、超小型探査機の着陸技術と機動性が向上し、よりチャレンジングかつ低リスクの探査の実現につながる。推進手段としては固体スラスターと空気力スラスターで異なるものの、いずれも限られたリソース下において高度な位置制御を行うという点で、共通の技術的課題に取り組んでいると考えている。

## 第1.2節 ミッション内容

ミッション内容は大きく分けて以下のようになると想定している

1. 起動
2. キャリアへの収納
3. キャリアからの放出・着陸
4. 展開
5. 離陸
6. 移動
7. ゴール探索
8. 画像航法
9. 着陸・終了

各状態でのミッション内容を以下に詳しくまとめる。



### 1. 起動

- 1.1 機体セッティング
- 1.2 電源投入

- 1.3 フライトコントローラパラメータ確認
- 1.4 センサーキャリブレーション
- 1.5 メインプログラム開始

## 2. キャリアへの収納

- 2.1 キャリアへ収納
- 2.2 収納判定
- 2.3 収納シーケンス終了を地上局へ送信
- 2.4 LoRaを切る

## 3. キャリアからの放出・着陸

- 3.1 パラシュートが風により展開する
- 3.2 落下を検知
- 3.3 LoRaを起動
- 3.4 放出を地上局へ送信
- 3.5 着陸を検知
- 3.6 着陸を地上局へ送信

## 4. 展開

- 4.1 ニクロム線によりテグスを断線し、ケーシングを展開
- 4.2 機体が直立(4.1と同時)
- 4.3 姿勢を判定
- 4.4 ニクロム線によりテグスを断線し、アームを展開
- 4.5 姿勢を判定
- 4.6 展開シーケンス終了を地上局へ送信

## 5. 離陸

- 5.1 Take-off モードにて離陸する
- 5.2 展開機構と機体が分離(5.1と同時)
- 5.3 ホバリング
- 5.4 安定判定
- 5.5 離陸シーケンス終了を地上局へ送信

## 6. 移動

- 6.1 ゴール座標まで移動
- 6.2 6.1中にLiDARの値を元に高度を調整
- 6.3 移動終了を地上局へ送信

## 7. ゴール探索

- 7.1 画像認識をする
- 7.2 物体が検知されれば画像航法に移る
- 7.3 検出されなければ機体を回転させ7.1に戻る

## 8. 画像航法

- 8.1 画像を元に再度ゴール座標を割り出す
- 8.2 目標物上空まで移動
- 8.3 再度画像を元にゴール座標を割り出す
- 8.4 画像航法シーケンス終了を地上局へ送信

## 9. 着陸・終了

- 9.1 着陸
- 9.2 着陸を検知
- 9.3 全シーケンス終了を地上局に送信

### 第1.3節 サクセスクライテリア

ミニマムサクセス	放出後パラシュートにより減速して落下し、機体に損傷なく離陸姿勢を整えて機体が離陸できる。ミッション内容5.2までの達成を指す。
フルサクセス	安定したGPS航法により機体を見失うことなく目標物5m以内に着陸し、自律飛行による絶対位置制御技術を実証する。ミッション内容6までの達成を指す。
エクストラサクセス	画像航法によりゴール座標に着陸し、高精度な自律飛行による相対位置制御技術を実証する。ミッション内容全ての達成を指す。

またさらに詳細な各シーケンスごとの達成項目とその評価方法を以下の表にまとめる。

	内容	評価方法
Minimum success	m1.放出されたことをログに残す	ログで確認
	m2.機体の座標・高度をログに残す	ログで確認
	m3.9軸データ(加速度, 地磁気, 角速度)のログを取る	ログで確認
	m4.パラシュートが展開する	目視で確認
	m5.着地判定をログに残す	ログで確認
	m6.着地後に機体がパラシュートを切り離せる	目視で確認
	m7.展開機構の展開により機体が離陸姿勢を取れる	目視で確認
	m8.機体の腕が展開できる	目視で確認
	m9.機体と展開機構を切り離す	目視で確認
	m10.放出・着地・展開シーケンスの完了及び機体状態(座標・高度)を地上局に送信する	目視で確認
	m11.機体が離陸する	目視で確認
Full success	f1.機体が一定の高度を保つことができる	目視で確認
	f2.機体が目標物に向かって飛行する	目視で確認
	f3.飛行中の座標・高度を地上局に送信する	目視で確認
	f4.目標物から5m以内に着陸する	目視で確認
Extra success	e1.目標物を認識してゴール座標を割り出す	ログで確認

	内容	評価方法
Minimum success	m1.放出されたことをログに残す	ログで確認
	m2.機体の座標・高度をログに残す	ログで確認
	m3.9軸データ(加速度, 地磁気, 角速度)のログを取る	ログで確認
	m4.パラシュートが展開する	目視で確認
	m5.着地判定をログに残す	ログで確認
	m6.着地後に機体がパラシュートを切り離せる	目視で確認
	m7.展開機構の展開により機体が離陸姿勢を取れる	目視で確認
	m8.機体の腕が展開できる	目視で確認
	m9.機体と展開機構を切り離す	目視で確認
	m10.放出・着地・展開シーケンスの完了及び機体状態(座標・高度)を地上局に送信する	目視で確認
	m11.機体が離陸する	目視で確認
Full success	f1.機体が一定の高度を保つことができる	目視で確認
	f2.機体が目標物に向かって飛行する	目視で確認
	e2.ゴール座標に着陸する	目視で確認

## 第2章 システム要求

### 第2.1節 レギュレーションを満たすためのシステム要求

番号	レギュレーションを満たすためのシステム要求
R1	CanSatは規定のサイズと質量以内であることが必要
R2	打ち上げ時の準静的荷重によって、安全基準を充足するための機能が損なわれていないことが必要
R3	打ち上げ時の振動荷重によって、安全基準を充足するための機能が損なわれていないことが必要
R4	ロケット分離時の衝撃荷重によって、安全基準を充足するための機能が損なわれていないことが必要
R5	CanSatはロケットから放出後位置が特定できることが必要
R6	CanSatは規定の終端速度の範囲内で降下できることが必要
R7	CanSatは落下時、パラシュート開傘時の衝撃を受けた後もその機能を維持しなくてはならない
R8	CanSatはロケット搭載時に無線の送出手を停止できることが必要。ただし、許可された無線機は除く。
R9	使用する全ての無線機はチャンネルを変更できることが必要
R10	CanSatはロケットに搭載後メンテナンスなしにミッションを維持できることが必要
R11	CanSatはロケットに損傷を負わせる可能性のある機構や物質を搭載してないことが必要
R12	CanSatは制御されることなく着地できることが必要

番号	レギュレーションを満たすためのシステム要求 (Comeback Competition)
CR1	CanSatは完全に自律的に制御されることが必要
CR2	チームは2種類のコントロールレコードを提出できることが必要

番号	レギュレーションを満たすためのシステム要求 (UAS)
UR1	CanSatはFAA認証を取得することが必要

UR2	CanSatは対地高度400ftを超えて飛行しないことが必要
-----	--------------------------------

## 第2.2節 ミッションを達成するためのシステム要求

1.3節、シーケンスごとの達成項目とその評価方法から、以下のシステム要求を設定した。

番号	ミッションを達成するためのシステム要求
M1	放出されたことをログに記録できることが試験で確認できている
M2	機体の座標・高度をログに記録できることが試験で確認できている
M3	9軸センサ(加速度, 地磁気, 角速度)のデータをログに記録できることが試験で確認できている
M4	着地時の衝撃荷重によって, ミッションを実現するための機能が損なわれていないことが試験で確認できている
M5	着地を検知し, ログに記録できることが試験で確認できている
M6	着地後に機体がパラシュートを切り離せることが試験で確認できている
M7	起立機構により機体が離陸姿勢を取れることが試験で確認できている
M8	機体の腕が展開できることが試験で確認できている
M9	放出・着地・展開シーケンスの完了及び機体状態(座標・高度)を地上局に送信できることが試験で確認できている
M10	機体が起立機構から離陸できることが試験で確認できている
M11	機体が一定の高度を維持して飛行できることが試験で確認できている
M12	機体が目標地点に向かって飛行できることが試験で確認できている
M13	機体が飛行中の座標・高度を地上局に送信できることが試験で確認できている
M14	機体が目標物から5m以内に着陸できることが試験で確認できている
M15	機体が目標物を認識し, 目標座標を割り出せることが試験で確認できている
M16	機体が目標座標に着陸できることが試験で確認できている
M17	機体が危険を検知した際, 安全に着陸できることが試験で確認できている

## 第3章 システム試験項目の設定

### 第3.1節 レギュレーションを満たすためのシステム試験項目

番号	検証項目名	対応するシステム要求番号	実施日
RV1	質量試験	R1	7月31日
RV2	機体の収納試験	R1	7月30日
RV3	GPSデータダウンリンク試験	R5	7月30日
RV4	準静的荷重試験	R2	8月5日
RV5	振動試験	R3	7月15日
RV6	分離衝撃試験	R4	7月15日
RV7	パラシュート性能試験	R6	5月22日
RV8	開傘衝撃試験	R7	7月4日
RV9	無線性能試験	R2 R8 R9	7月31日

番号	検証項目名	対応するシステム要求番号	実施日
CRV1	End-to-end試験	R10, R11, CR1	7月29日
CRV2	制御履歴レポート作成試験	CR2	7月29日

番号	検証項目名	対応するシステム要求番号	実施予定日
FRV1	FAA機体申請書	FR1	7月5日

### 第3.2節 ミッションを達成するためのシステム試験項目

番号	検証項目名	対応するシステム要求番号	実施日
MV1	機体放出試験	M1	7月8日
MV2	放出判定試験	M1	7月17日

MV3	着地衝撃試験	M4, M6	7月7日
MV4	着地判定試験	M7	7月29日
MV5	起立展開試験	M6, M7	7月7日
MV6	腕展開試験	M8	7月7日
MV7	機体離陸試験	M10	7月7日
MV8	飛行性能確認試験	M11	7月4日
MV9	ログ試験	M1, M2, M3, M9, M13	7月18日
MV10	GPS航法飛行試験	M12, M14	7月29日
MV11	機体着陸試験	M17	7月5日
MV12	ゴール検知試験	M15	7月29日
MV13	画像航法試験	M15, M16	7月29日
MV14	航続距離検証試験	M12	8月30日

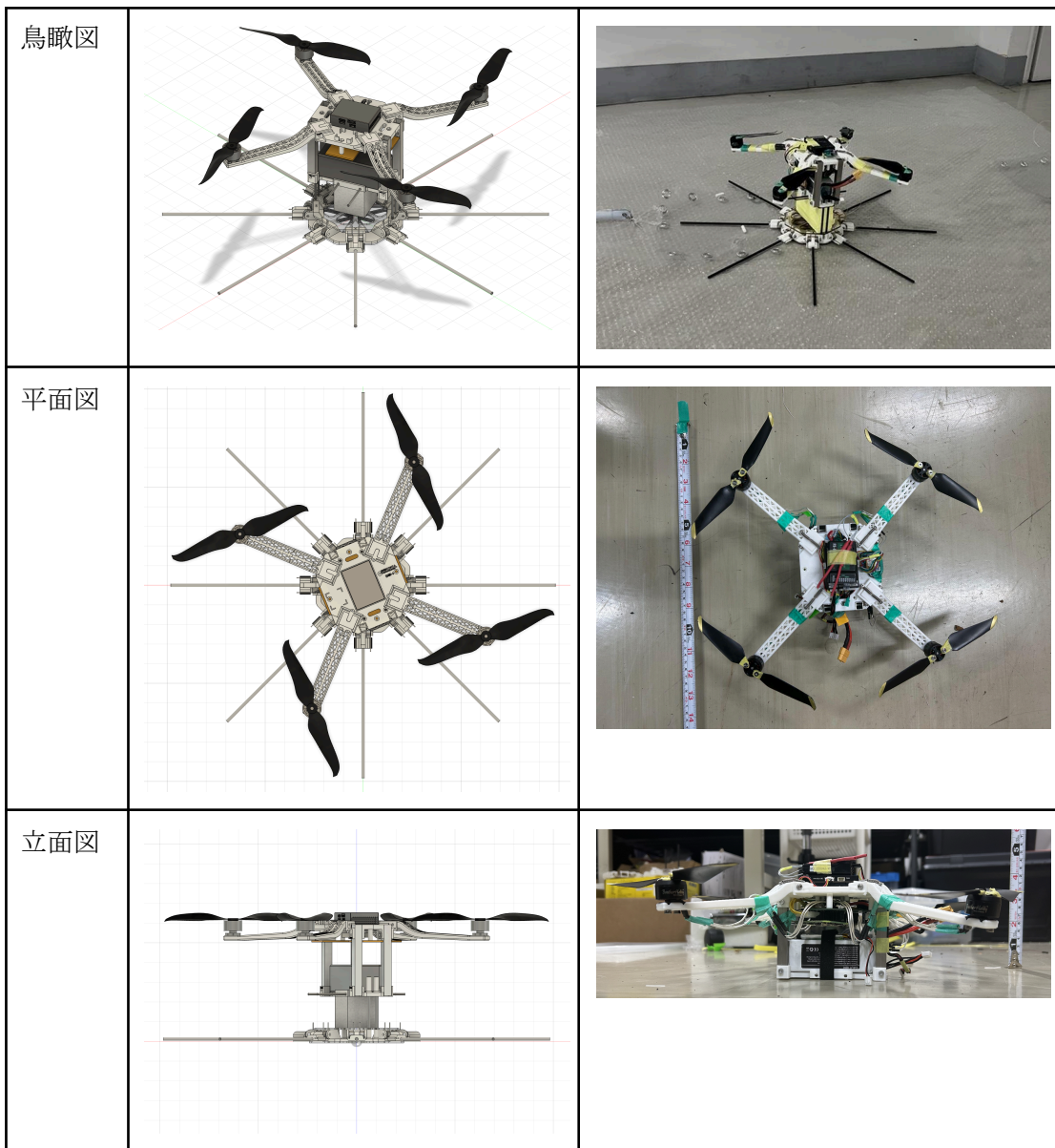
## 第4章 システム仕様

### 第4.1節 機体概観

#### 第4.1.1項 機体概観

以下の図はフライト機構が起立機構に載っている図である。  
 フライト機構とは、その名の通り飛翔する機構であり、これがゴール座標へ向かう。起立機構とは、フライト機構がロケットからの放出・落下・着地後に飛行姿勢をとるための機構である。ロケット内部ではフライト機構が起立機構に収納されるため、フライト機構の腕は折り畳まれる仕様となっている。

図	CAD	実機
---	-----	----



## 第4.1.2項 機体諸元

### 第4.1.1.1目 電力

厳密な計算は行っておらず、開発初期段階においてeCalcに機体諸元やモーター等を入力した上でバッテリーのセル数及び容量を変化させ、当初目標としていた航続距離6kmを達成できるバッテリーとして4S 3300mAhのものを選択した。

その後構造の変更は多少あったものの、バッテリーの変更は大幅な設計変更に繋がるものと考えたため、また長距離飛行試験においてバッテリーの残量の変化を確認するのが遅れたためバッテリーは変更しなかった。

### 第4.1.1.2目 質量

機体質量について、以下の表にまとめる。

機体質量

構成要素	質量 (g)
機体親機	735
機体子機	165
エンベロープ・パラシュート	146
合計	1046

#### 第4.1.1.3目 寸法

機体寸法について、以下の表にまとめる。

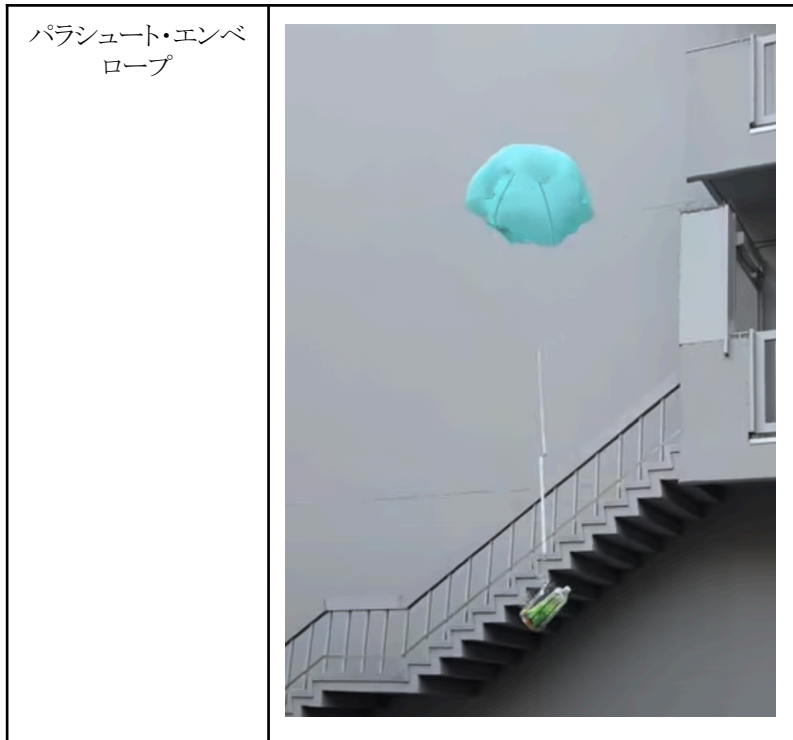
状態	展開	収納	飛行
直径 [mm]	491	139	359.8
高さ [mm]	165.05	165.05(パラシュート除く)	106.05

## 第4.2節 機体機構

### 第4.2.1項 パラシュート・エンベロープ

開傘時半径380mmの半球で、天頂部に半径40mmの円形の穴が空いている(1kgをぶら下げたときに終端速度が5m/sになる大きさ)。素材はリップストップナイロン。根本に金属メジャー2本を使って作った長さ1.2mの棒があり、着地時にパラシュートが被らないようにしてある。

図	画像
---	----



エンベロープはPVCの台形2枚で、畳まれた起立機構とフライト機構の側面を囲む。エンベロープのパラシュートと繋がっている部分の周りにT字状に養生テープを貼ることによって周辺の強度を増し、ロケットからの分離時の開傘衝撃に耐えられるようにした。エンベロープを閉じておくためにテグスが通っており、ニクロム線を使ってテグスを溶断後、起立機構のダブルトーションバネの力で開く。

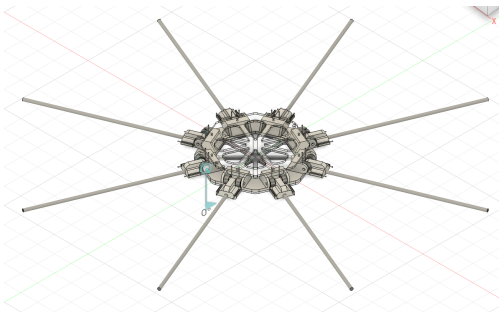
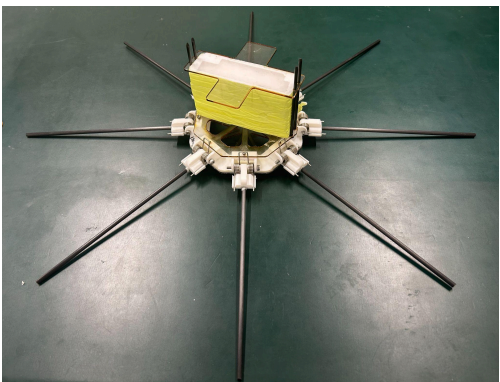
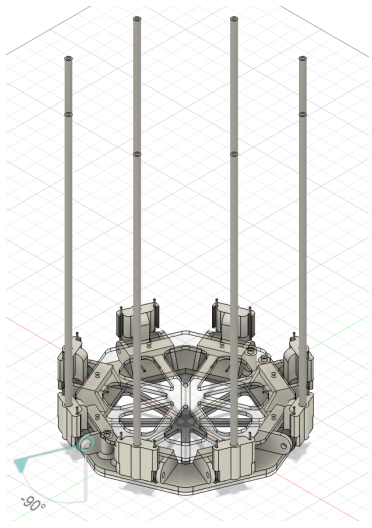
	図	画像
エンベロープ		

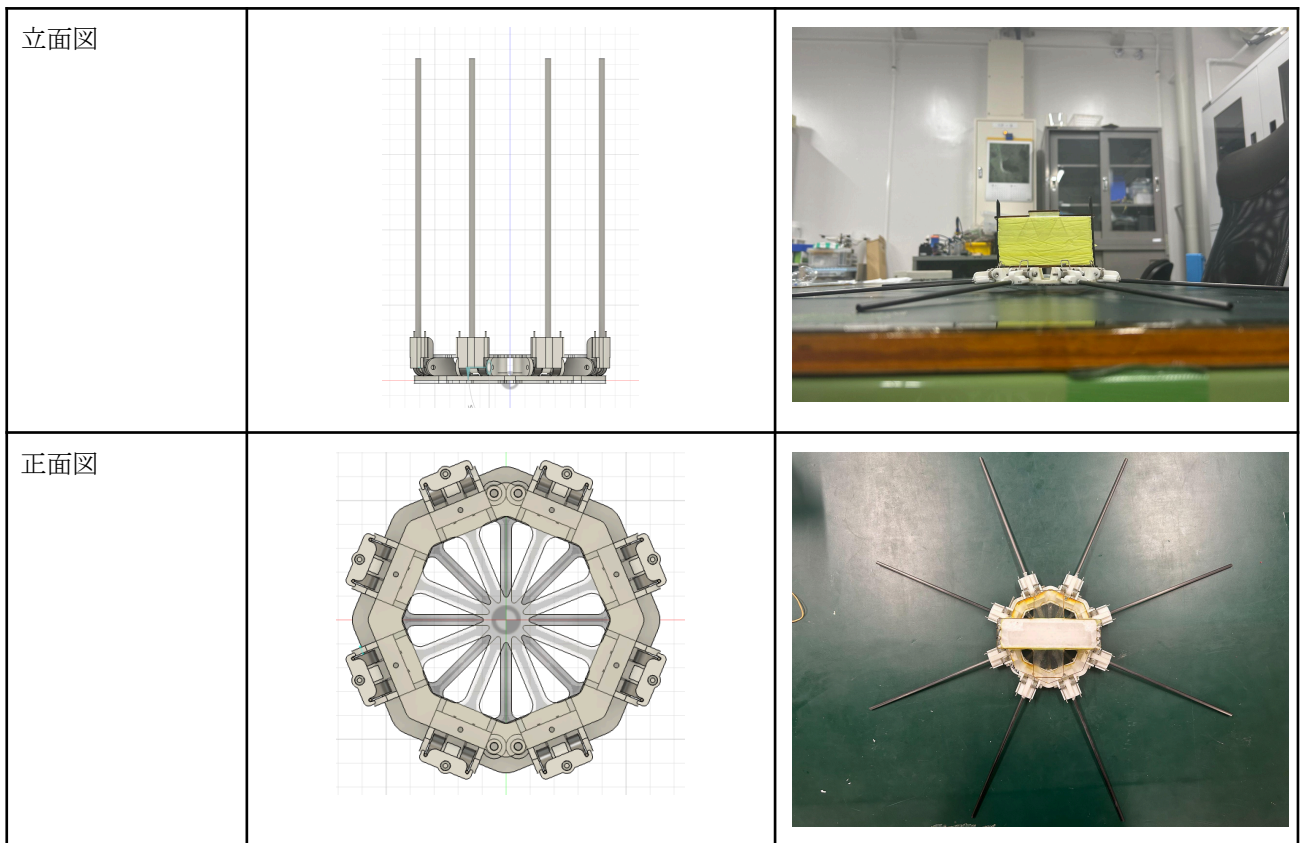
パラシュートはエンベロープの片方に繋がっており、テグス溶断によってエンベロープが開いた際にパラシュートもフライト機構から分離される。

## 第4.2.2項 起立機構

ダブルトーションバネを用いた起立機構を採用。テグスによって土台の腕部分が固定されており、着地を判定するとニクロム線加熱により溶断する。土台の腕と土台の接続に用いられるダブルトーションバネにより土台の腕が開き、頭を上にした状態で起き上がる。他のアクチュエーターを用いる起き上がり機構であると機体の重量が大幅に増加するため、軽量化を狙いニクロム線溶断を採用した。

ダブルトーションバネにより横たわった状態から機体が離陸できるように起立する。腕部はPLAで作成されたバネ受け部にダブルトーションバネとカーボンロッドが接続されている。起立機構下部と上部にはポリカーボネート板が取り付けられており、落下の衝撃やトーションバネのトルクを分散する役割を果たしている。実際落下試験などに耐えることが確認されたため、基本をこの形状とし、今後インフィルを下げるなどなるべく重量の削減に取り組む。

図	CAD	実機
鳥瞰図		
収納時の鳥瞰図		

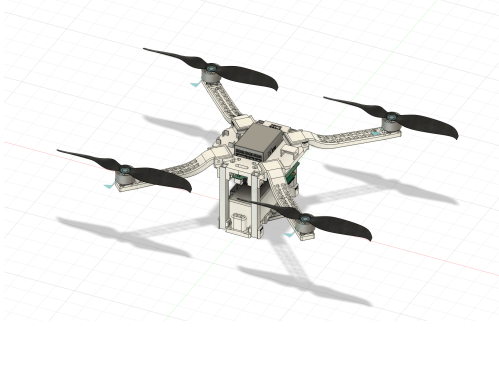

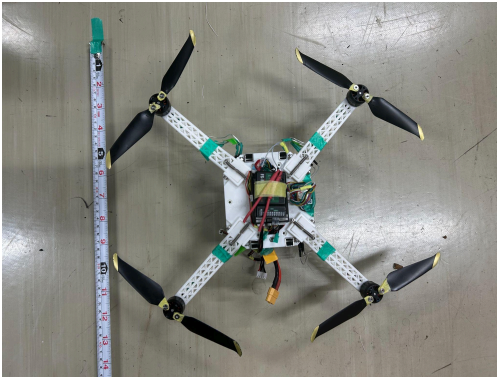
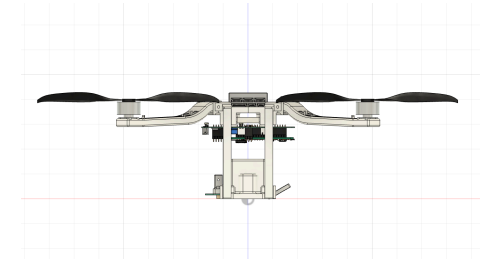
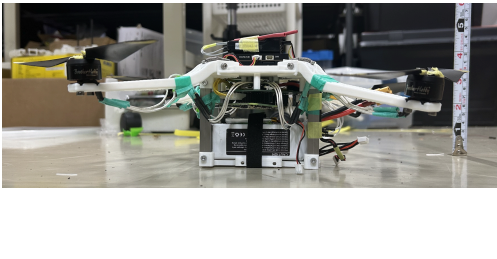


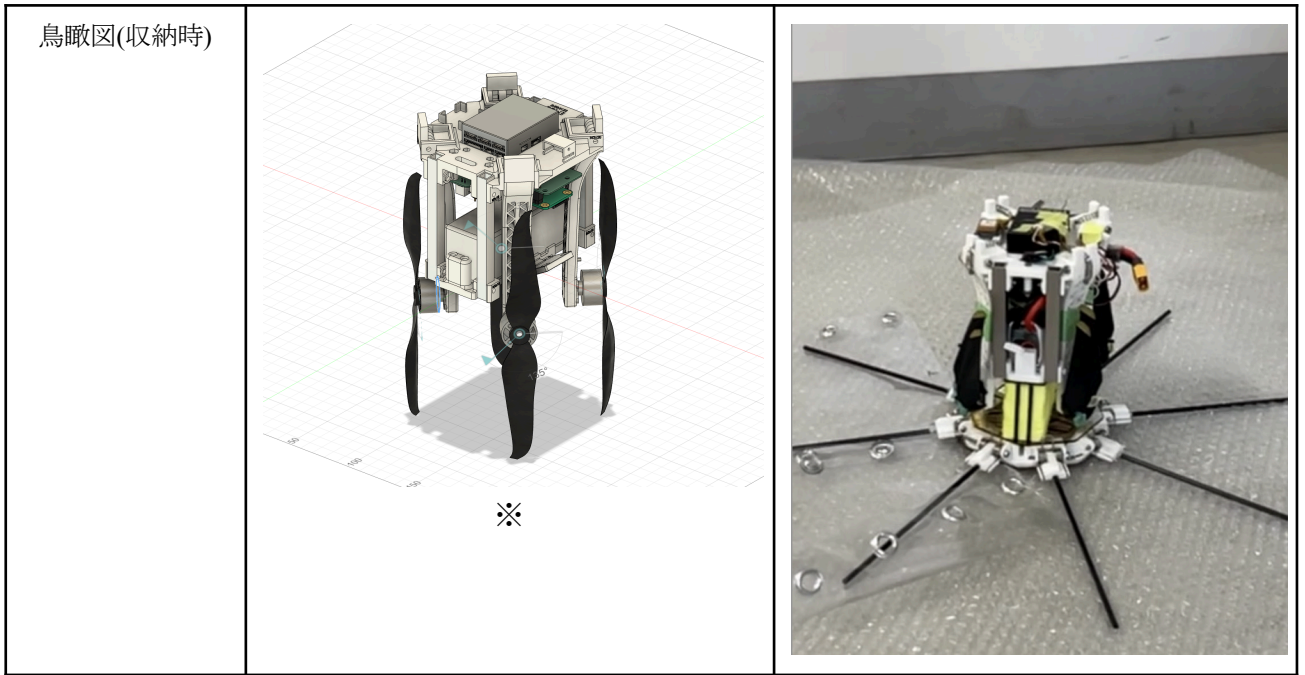
### 第4.2.3項 フライト機構

フライト機構は、主に、3Dプリンタで製作する3種類のPLA素材部品（底板、天板、アーム）と、4本のアルミ支柱、そして各電子部品から構成される。本機の特徴は、アームの根本が回転することによる高い収納性を有しながらも、柱にアルミ角パイプを採用することで機体全体として高い剛性を備えていることである。

まず、底板には起立機構と接続するためにカーボン棒を通す筒が4箇所ある。底板の四隅はアルミ支柱とねじ止めされている。底板に搭載される部品は、モーター駆動用リポバッテリー、カメラ、LiDARである。また、アームを収納状態から展開状態にするためのニクロム線溶断機構が設置される。

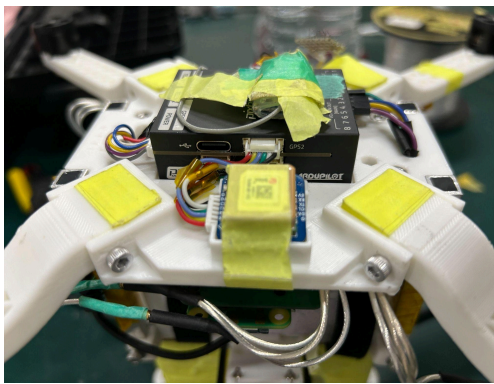
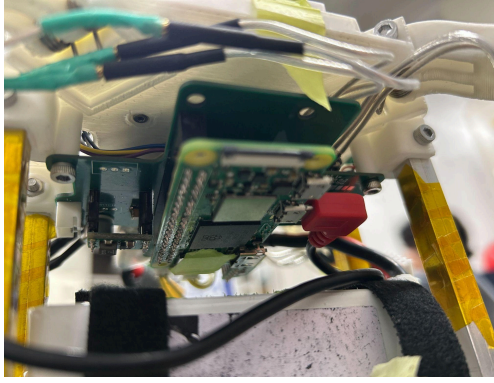
アルミ支柱の上端には天板が接続される。天板に搭載される電子部品は、上側にフライトコントローラー、GPS、ニクロム線溶断機構、下側にESC、プリント基板である。このニクロム線溶断機構はエンベロープを繋ぎ止めるテグスを切断し、起立機構を展開させるためのものである。天板の四隅からは等方に4本のアームが接続される。アームは接続軸を中心に下向きに90度回転し、収納が可能である。従来はアームと天板を上側で結ぶ引きバネによる弾性力で展開していたが、弾性力の強化のため、ダブルトーションバネによる展開方式に変更予定である（パーツ単位で実験済みであり、現在実装途中）。アームの先にはモーターとプロペラが装着されている。アームの展開機構の構想について説明する。起立機構と同じく、ダブルトーションバネを4本のアームと胴体との接続部分それぞれに使用する。もともとはアームを根本から90度折って収納した状態でテグスを巻いて固定しており、起き上がりが完了したのちに、テグスをニクロム線加熱により溶断し、アームが90度開く機構となる。ニクロム線溶断は、電気制御を最小にすることで、重いモーターやバッテリーを削減でき、電気系統の故障リスクも根本から回避できる。よって機体の軽量化と再現性向上につながるため採用した。

図	CAD	実機
鳥瞰図		
立面図		
正面図		

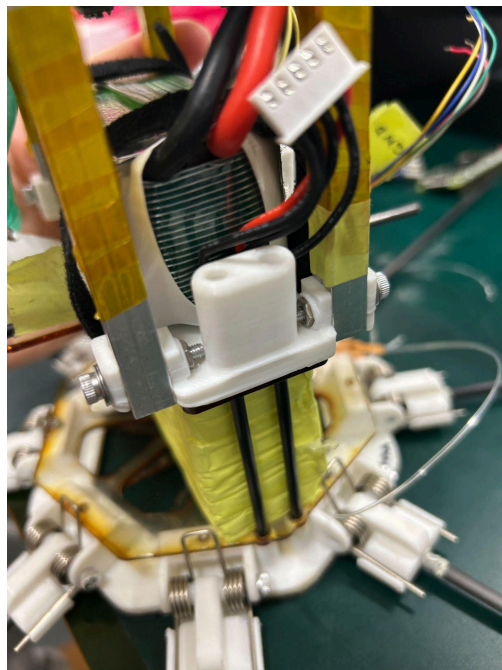


※CADの都合上、収納時の鳥瞰図ではプロペラが180度開いているが、実際には2枚のプロペラ同士のなす角が約0度になるように回転させて収納する。実機画像参照。

以下に機体のより詳細部分の写真とその説明をまとめる。

	写真	説明
天板部		<p>天板にはGPS(1枚目手前の四角状のもの)とフライトコントローラー(写真中央の黒い直方体)とRC受信機(1枚目フライトコントローラー上部に取り付けているもの。灰色の線画アンテナである。)、テグス溶断部(フライトコントローラーの奥)が搭載されている。</p>
天板裏		<p>十字形の基板が搭載されている。写真では基板の裏に隠れてしまっているが、4in1ESCとPowerModuleも搭載されている。</p>

起立展開機構接合部



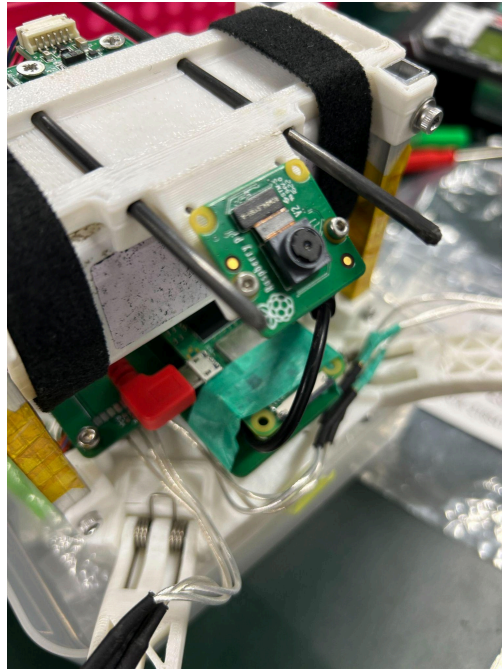
起き上がり時に写真の通り機体前方と後方に2つずつある穴に起立機構から伸びているカーボンロッドが刺さり、これが支えとなってフライト機構の離陸姿勢を整える。

アーム折り畳み部



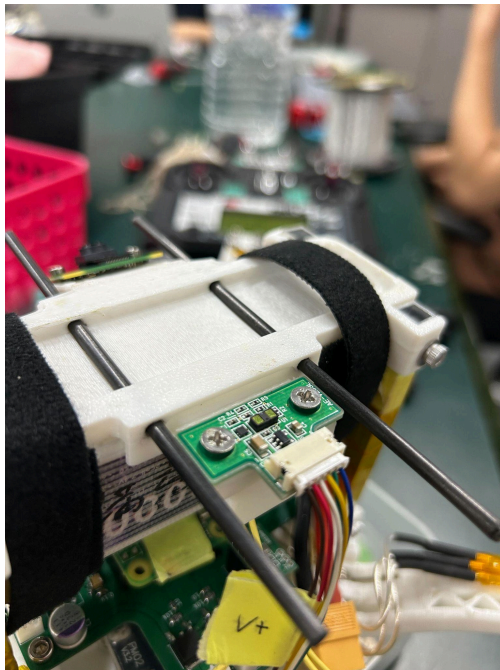
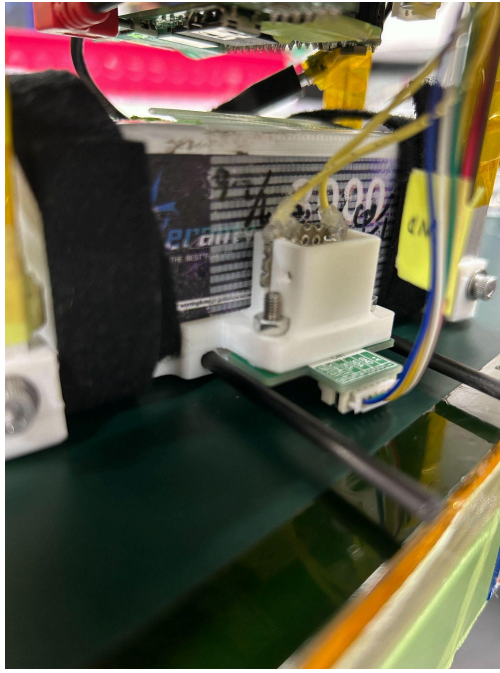
腕の根本にあるダブルトーションバネにより開いた状態を維持し、折り畳み時はこれをテグスにより縛る。

底板カメラ取り付け部



水平面に対して22.5°の角度で機体進行方向左向きにカメラを取り付けている。写真では取り付け位置が想定とずれており2箇所  
のねじ止めとなっているが、4箇所のプラスチックネジによる固定を予定している。

底板測距センサ取  
り付け部



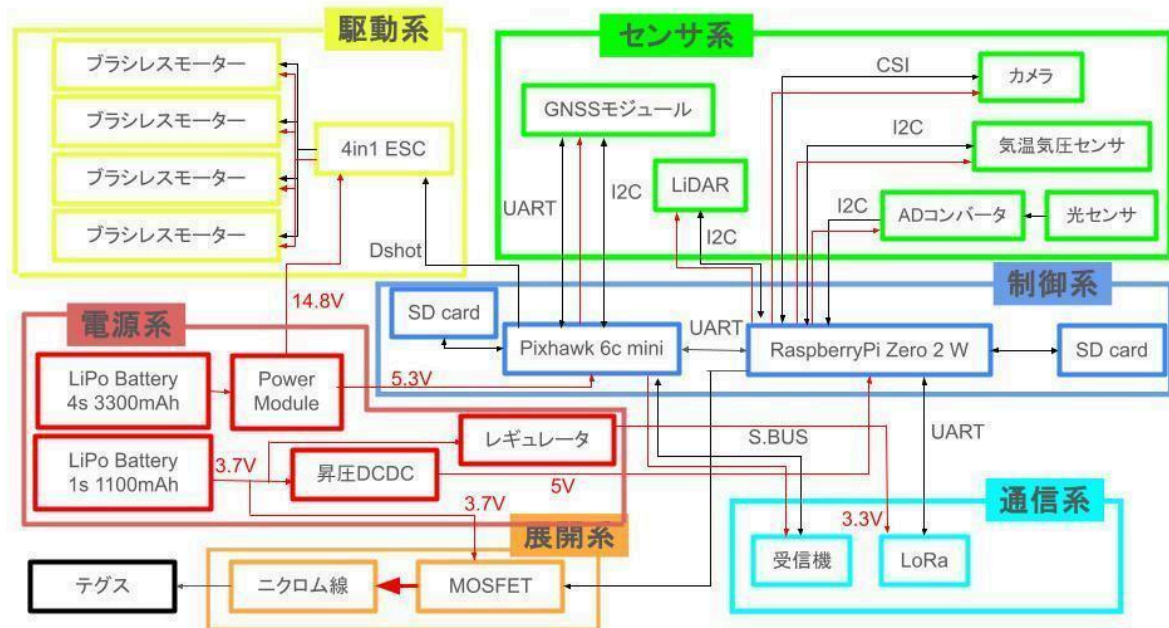
機体進行方向右側に、飛行時の機体下向きに測距センサを取り付ける。固定方法は2箇所のおねじ止めである。2本のカーボンロッドにより機体が傾いた際の測距センサの地面との衝突を回避している。

## 第4.3節 搭載機器

### 第4.3.1項 システム図と使用コンポーネント

#### 第4.3.1.1目 システム図

システム図を以下に示す。



各コンポーネント間の給電システムを赤矢印で示し、通信を黒線で示した。給電線の電圧の値は矢印の横に示してある。通信線の通信方式についても同様に矢印の横に併記している。 nichrome線とテグスの間の矢印に関しては通信ではないものの nichrome線から繋がるコンポーネントということで黒矢印により繋いだ。

#### 第4.3.1.1目 電子系 / 動力系

以下の表に各コンポーネントの詳細についてまとめた。

表: 使用コンポーネント詳細

搭載計器	型番	使用理由	データシート	購入先
フライトコントローラー	Pixhawk 6c mini model A	姿勢判定、モーター制御に用いる。	<a href="https://docs.holybro.com/autopilot/pixhawk-6c-mini">https://docs.holybro.com/autopilot/pixhawk-6c-mini</a>	<a href="https://x.gd/vyjdi">https://x.gd/vyjdi</a>
マイコン	Raspberry Pi Zero 2 W	センサからの情報の処理や画像認識を行う。	<a href="https://d2air1d4eqhwg2.cloudfront.net/media/files/051aac55-4763-4418-8d26-58889bb34b54.pdf">https://d2air1d4eqhwg2.cloudfront.net/media/files/051aac55-4763-4418-8d26-58889bb34b54.pdf</a>	<a href="https://raspberrypi.ksyic.com/">https://raspberrypi.ksyic.com/</a>
ブラシレスモーター	BrotherHobby F2008 1000kv	CanSatを飛行させる。	<a href="https://www.brotherhobbystore.com/products/f-2008-motor-208">https://www.brotherhobbystore.com/products/f-2008-motor-208</a>	<a href="https://www.brotherhobbystore.com/products/f-2008-motor-208">https://www.brotherhobbystore.com/products/f-2008-motor-208</a>
ESC	Diatone-Drone mamba Mini f40_bls	モーターの回転を制御するために使用する。	<a href="https://www.diatone.us/products/mamba-f40_bls-dshot600-4in1-esc-40a-6s?srsId=AfmBOoqmKudJYlhBs3pi2_snRHlRqitSZG_IAV1IA26FNsBInG0p2oQ">https://www.diatone.us/products/mamba-f40_bls-dshot600-4in1-esc-40a-6s?srsId=AfmBOoqmKudJYlhBs3pi2_snRHlRqitSZG_IAV1IA26FNsBInG0p2oQ</a>	<a href="https://ja.aliexpress.com/item/1005005902099570.html">https://ja.aliexpress.com/item/1005005902099570.html</a>
パワーモジュール	Holybro PM02 V3	4SのLiPoバッテリーからフライトコントローラーへの給電を行う。	<a href="https://holybro.com/products/pm02-v3-12s-power-module?srsId=AfmBOop7LoZtRO1SFspqCpulG4iz3IK17omgDr4HsUuo">https://holybro.com/products/pm02-v3-12s-power-module?srsId=AfmBOop7LoZtRO1SFspqCpulG4iz3IK17omgDr4HsUuo</a>	<a href="https://rcdrone.top/ja/products/holybro-pm02-v3-power-module-12s?srsId=AfmBOop">https://rcdrone.top/ja/products/holybro-pm02-v3-power-module-12s?srsId=AfmBOop</a>

			<a href="#">OmRF02aJc8Ub</a>	<a href="#">p-V8-rdcCKRdwpVoshzoCMa1tuw14WfwxUDxoROMfFHyutZp5</a>
昇圧DCDC	TPS61230A	1SのLiPoバッテリーからマイコンへ給電を行う。	<a href="https://strawberry-linux.com/pub/tps61230a-manual.pdf">https://strawberry-linux.com/pub/tps61230a-manual.pdf</a>	<a href="https://strawberry-linux.com/catalog/items?code=16123">https://strawberry-linux.com/catalog/items?code=16123</a>
LoRaモジュール	TLM922S-P01A	地上局にデータを送信するために使用する。	<a href="https://akizukidenshi.com/goodsuffix/tlm922s.pdf">https://akizukidenshi.com/goodsuffix/tlm922s.pdf</a>	<a href="https://akizukidenshi.com/catalog/g/g114076/">https://akizukidenshi.com/catalog/g/g114076/</a>
レギュレータ	ADP151AUJZ-3.3	LoRaモジュールへの電源供給	<a href="https://akizukidenshi.com/goodsuffix/ADP151_jp.pdf">https://akizukidenshi.com/goodsuffix/ADP151_jp.pdf</a>	<a href="https://akizukidenshi.com/catalog/g/g114180/">https://akizukidenshi.com/catalog/g/g114180/</a>
受信機	FS-A8S	緊急時の停止信号受信用に搭載する。	<a href="https://www.flyskyttech.com/upload/photo/20210426/FS-A8S%E8%AF%B4%E6%98%8E%E9%A1%B520210426.pdf">https://www.flyskyttech.com/upload/photo/20210426/FS-A8S%E8%AF%B4%E6%98%8E%E9%A1%B520210426.pdf</a>	<a href="https://japanese.alibaba.com/product-detail/Flysky-FS-A8S-2-4G-8CH-1600210752117.html">https://japanese.alibaba.com/product-detail/Flysky-FS-A8S-2-4G-8CH-1600210752117.html</a>
MOSFET	MTB7D0N6RJ3	機体展開のためのニクロム線によるテグス切断の際に使用する。	<a href="https://akizukidenshi.com/goodsuffix/MTB7D0N06RJ3_220718.pdf">https://akizukidenshi.com/goodsuffix/MTB7D0N06RJ3_220718.pdf</a>	<a href="https://akizukidenshi.com/catalog/g/g117191/">https://akizukidenshi.com/catalog/g/g117191/</a>
GNSSモジュール	M10Q-5883	測位衛星からの情報をもとに目的地の位置を得るために用いる。	<a href="https://www.mateksys.com/?portfolio=m10q-5883">https://www.mateksys.com/?portfolio=m10q-5883</a>	<a href="https://dayscape.jp/products/mateksys-gnss-compass-m10q-5883?srltid=AfmBOoocooUr5TZNRS8-zkOYq_BOQ-zPW3p1niQF66K3QBvB2zkz_aYj">https://dayscape.jp/products/mateksys-gnss-compass-m10q-5883?srltid=AfmBOoocooUr5TZNRS8-zkOYq_BOQ-zPW3p1niQF66K3QBvB2zkz_aYj</a>
LiDAR	TF mini-s	地面との距離を計測する。	<a href="https://doc.switch-science.com/media/files/5e6511ef-aabb-4144-977b-a97d7a7ac853.pdf">https://doc.switch-science.com/media/files/5e6511ef-aabb-4144-977b-a97d7a7ac853.pdf</a>	<a href="https://www.switch-science.com/products/8152">https://www.switch-science.com/products/8152</a>
ADコンバータ	MCP3425A0T-E/CH	光センサのアナログ信号をデジタル信号に変換する。	<a href="https://akizukidenshi.com/goodsuffix/MCP3425.pdf">https://akizukidenshi.com/goodsuffix/MCP3425.pdf</a>	<a href="https://akizukidenshi.com/catalog/g/g107638">akizukidenshi.com/catalog/g/g107638</a>
光センサ	MI527/MI5527	キャリア放出時の判定に使用する。	<a href="https://akizukidenshi.com/goodsuffix/MI5_series_jp.pdf">https://akizukidenshi.com/goodsuffix/MI5_series_jp.pdf</a>	<a href="https://akizukidenshi.com/catalog/g/g100110/">https://akizukidenshi.com/catalog/g/g100110/</a>
気温気圧センサ	AE-BME280	高度判定に用いる。	<a href="https://akizukidenshi.com/goodsuffix/AE-BME280_manu_v1.1.pdf">https://akizukidenshi.com/goodsuffix/AE-BME280_manu_v1.1.pdf</a> <a href="https://akizukidenshi.com/goodsuffix/BST-BME280_DS001-10.pdf">https://akizukidenshi.com/goodsuffix/BST-BME280_DS001-10.pdf</a>	<a href="https://akizukidenshi.com/catalog/g/g109421/">https://akizukidenshi.com/catalog/g/g109421/</a>
カメラ	Pi カメラ公式v2 913-2664	ゴール認識に用いる。	<a href="https://www.raspberrypi.com/documentation/accessories/camera.html">https://www.raspberrypi.com/documentation/accessories/camera.html</a>	<a href="https://raspberrypi.ksvic.com/main/index/pdp.id/144/pdp.open/144">https://raspberrypi.ksvic.com/main/index/pdp.id/144/pdp.open/144</a>

各コンポーネントの選定理由については次の表にまとめた。

表：使用コンポーネント選定理由

搭載計器	型番	選定理由
------	----	------

フライトコントローラー	Pixhawk 6c mini model A	CanSatの大きさの制約、飛行性能などを考慮して選定した。
マイコン	Raspberry Pi Zero 2 W	並列処理ができ、小ささと新しさを考慮して選定した。過去の作動実績もある。
ブラシレスモーター	BrotherHobby F2008 1000kv	対応電圧、重量などから選定した。昨年の作動実績もある。
ESC	Diatone-Drone mambba Mini f40_bls	4s対応、小型、軽量な4in1ESCを選定した。
パワーモジュール	Holybro PM02 V3	比較的小型であり、信頼性の高いものを選定した。
昇圧DCDC	TPS61230A	スイッチングが速く、定格電流が大きい。
LoRaモジュール	TLM922S-P01A	伝達距離と法律に違反しないことを考慮し選定した。
レギュレータ	ADP151AUJZ-3.3	通信への影響を考慮し低ノイズ・低損失のものを選定した。
受信機	FS-A8S	使用経験があり、十分小さく軽量である。
MOSFET	MTB7D0N6RJ3	対応電圧から選定した。価格と入手性を考慮した。
GNSSモジュール	M10Q-5883	精度と価格と入手性を考慮した。
LiDAR	AE-VL53L1X	必要な計測性能を有している上、使用経験があり、価格と入手性を考慮した。
ADコンバータ	MCP3425A0T-E/CH	十分小型であること、昨年使用した実績があることから使用している。
光センサ	MI527/MI5527	必要な計測性能を有している上、使用経験があり、価格と入手性を考慮した。
気温気圧センサ	AE-BME280	精度と価格と入手性を考慮した。
カメラ	Piカメラ公式v2 913-2664	Raspberry Piに公式で対応しており、入手性もよく価格も手頃であった。

#### 第4.3.1.3目 構造系

以下の表に構造系の使用部品についてまとめた。

表: 構造系使用部品一覧

分類	材質・型番	入手先・参考情報等	備考
パラシュート	リップストップナイロン	<a href="https://www.amazon.jp/dp/B085T5WF3T?ref=ppx_pop_mob_ap_share">https://www.amazon.jp/dp/B085T5WF3T?ref=ppx_pop_mob_ap_share</a>	
スイベル	ステンレススイベル 3mm 0.29kN	<a href="https://www.monotaro.com/p/3713/9444/">https://www.monotaro.com/p/3713/9444/</a>	
テグス	ナイロンテグス50号		パラシュート用
テグス	ナイロンテグス30号		包む用
ダブルトーションバネ	NoF7555	<a href="https://www.samini.co.jp/order/stock-springs.php">https://www.samini.co.jp/order/stock-springs.php</a>	
ダブルトーションバネ	NoF7557	<a href="https://www.samini.co.jp/order/stock-springs.php">https://www.samini.co.jp/order/stock-springs.php</a>	
支柱	アルミ角パイプ 8mm*8mm*1mm		
起立機構補強板	ポリカーボネート		下記の起立機構上の台座

			を挟むように2枚使用
3Dプリンタ造形パーツ	PLA, PLA-CF	東京大学大学院工学系研究科 国際工学教育推進機構 ものづくり部門	使用機種 Bambu Lab X-1 carbon
ハトメ	スチール(真鍮メッキ) NO1 4.2mm	<a href="https://www.monotaro.com/g/02687850/?t.q=%E3%83%8F%E3%83%88%E3%83%A1%E3%83%97%E3%83%A9%E3%82%A4%E3%83%A4%E3%83%BC%E3%82%BB%E3%83%83%E3%83%88">https://www.monotaro.com/g/02687850/?t.q=%E3%83%8F%E3%83%88%E3%83%A1%E3%83%97%E3%83%A9%E3%82%A4%E3%83%A4%E3%83%BC%E3%82%BB%E3%83%83%E3%83%88</a>	
ネジ	アルミニウム		
ネジ	ポリカーボネート		
起立機構上のフライト機構を乗せる台座	発泡スチロール	ハンズ	
起立展開・機体離陸用棒	カーボンロッド		
エンベロープ	軟質塩ビ CR-SCA2N-T	<a href="https://www.monotaro.com/p/7395/9262/">https://www.monotaro.com/p/7395/9262/</a>	
パラシュートにつけていたメジャー棒	金属メジャー	100均	

### 第4.3.2項 バッテリー

電源系統はモーター駆動用と制御回路用で分けている。

モーター駆動用リポバッテリー

リチウムポリマーバッテリー Teranty 4s 3300mAh

購入サイト:<https://x.gd/7Z5b2>

制御回路駆動用リポバッテリー

リチウムポリマーバッテリー EEMB 1s 1100mAh

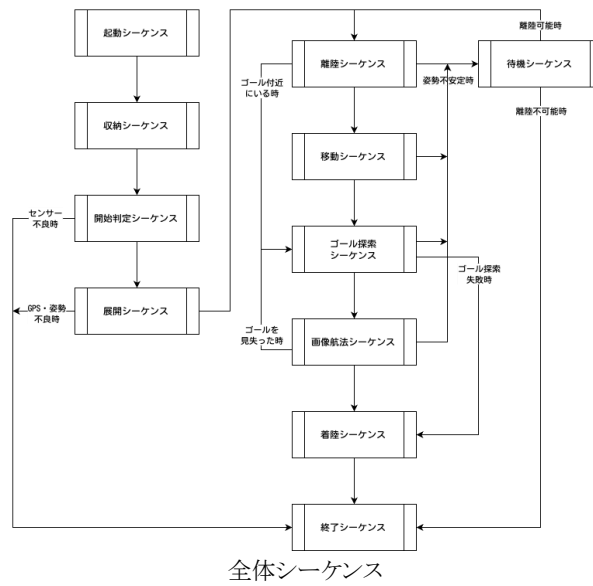
購入サイト:<https://x.gd/71Is2R>

安全対策

- 使用時または充電時は必ず1人以上の見張りをつけ、異常を感じた場合は使用を中止する。
- 使用前には毎回リポチェッカーで状態を確認する。
- 運送・保管時はコネクタの金属部分を絶縁テープで覆った上で難燃性素材で作られたリポバッテリーセーフティバッグに入れる。
- 長時間使用しない場合はストレージモードで充電しておく。

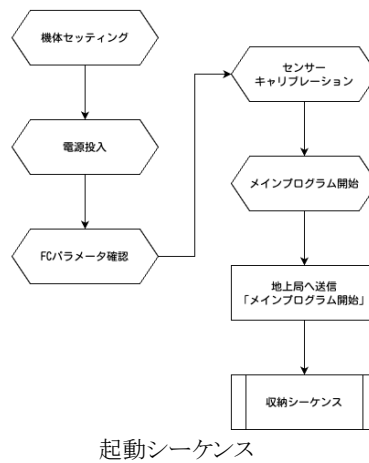
## 第4.4節 アルゴリズム

- 全体シーケンス



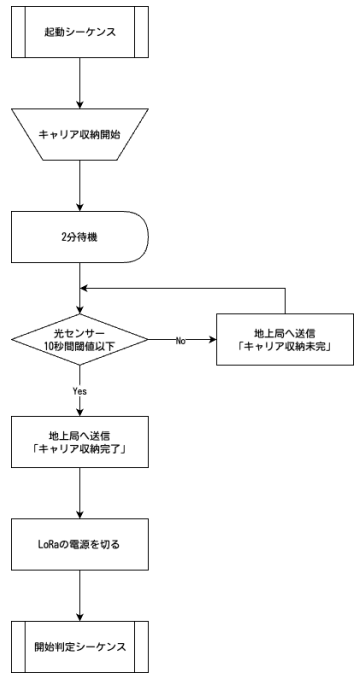
全体のシーケンスとしては左の2列の「起動」「収納」「開始判定」「展開」「離陸」「移動」「ゴール探索」「画像航法」「着陸」「終了」のシーケンスを順に行うことを想定している。またそれらの中で不具合があった時に「待機」「終了」シーケンスに移行するような例外処理も用意している。以下では全体シーケンスに含まれる個々のシーケンスについて述べる。なお、センサーの値の取得時間や待機時間などについてはまだ実験を行えていないため仮の値を書いている。

- 起動シーケンス



起動シーケンスではキャリア収納前の最終的なセッティングをするシーケンスである。ここでミッション達成のための最終チェックをする。

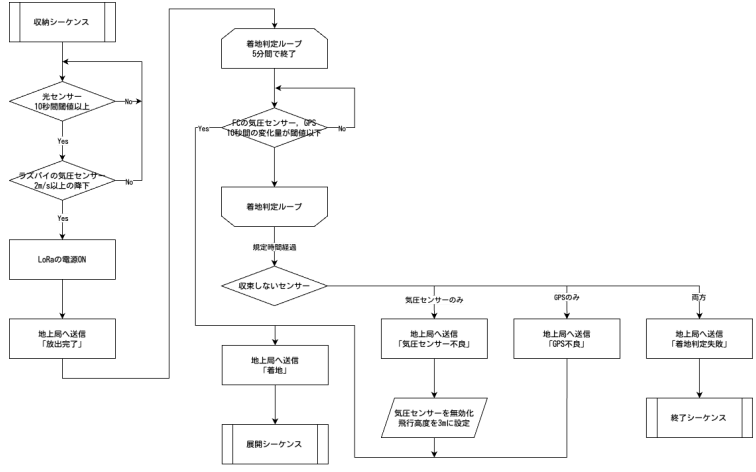
- 収納シーケンス



収納シーケンス

収納シーケンスではキャリアへの収納と開始判定シーケンスに移行する準備を行う。光センサーの値をみて一定時間暗くなった時にキャリアへの収納を検知し、開始判定シーケンスへ移行する。ここでLoRaの電源を切るのは、実際のロケット内では衛星の通信を行わないという要件が科される場合があることを見越しての処置である。

● 開始判定シーケンス

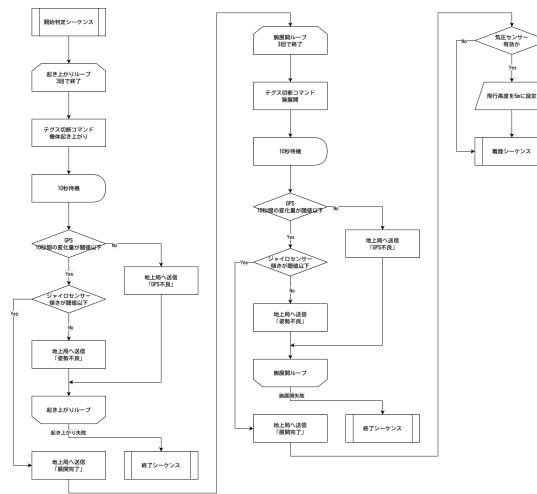


開始判定シーケンス

開始判定シーケンスでは、この後機体の展開や飛行を行って良いかを判定する。今回パラシュートで降下して着地した状態から飛行することを想定するため、着地を検知することと、そのために降下を検知することが必要となる。まず、キャリアからの放出・降下は光センサーによる明るさの判定、そして気圧センサーによる降下速度の判定により行う。判定できたらLoRaの通信を再開する。次に、着地は気圧センサーとGPSから得られる機体位置変化量を元に検出する。ここで、気圧センサーとGPSの値の不良時の例外処理を用意した。気圧センサーの値がなくても飛べる(要実験)ので、気圧センサー不良時には気圧センサーを無効化する。その場合機体下部に設置したLiDARを元に地面との相対位置を把握することを考え、飛行高度をLiDARが届く4mより下に設定することを想定している。GPSセンサー不良時には、展開機構がGPSの受信に影響している可能性を考え、この段階ではエラー通知のみすることにした。両方

のセンサーが不良である場合は機体に何かしらの問題がある可能性があるので全シーケンスを終了することにした。

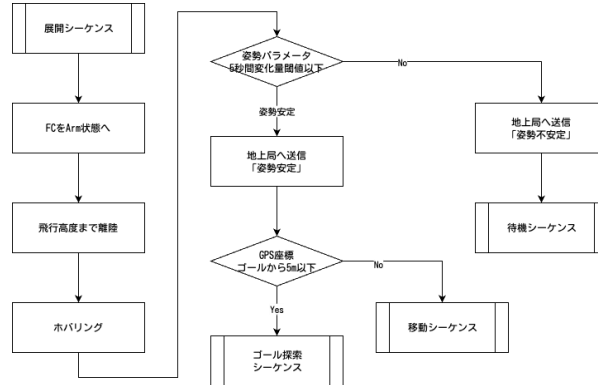
- 展開シーケンス



展開シーケンス

展開シーケンスでは起き上がり機構と腕の展開を行う。それぞれの展開後にGPSの座標が一定の範囲に収束していることを確認し、問題がある場合は終了する。また、機構の展開に問題がある場合を想定して、問題発生時には動作を何度か繰り返すループを導入する。

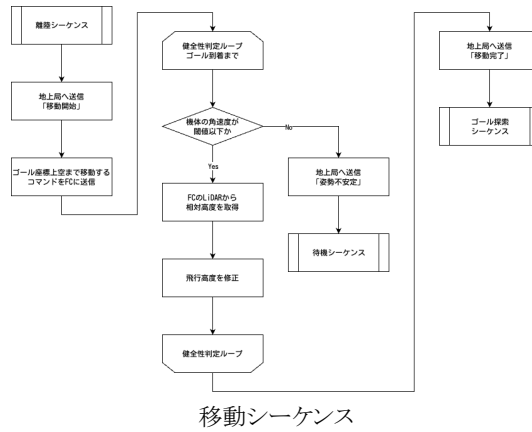
- 離陸シーケンス



離陸シーケンス

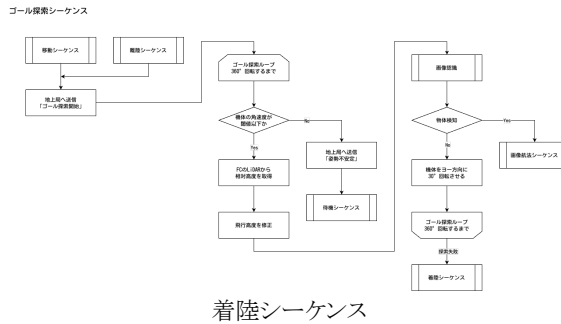
離陸シーケンスでは、その前のシーケンスで機体の姿勢が正しいことを確認しているため、そのまま離陸する。離陸後は機体の安定性を確認するために数秒ホバリングする。その後、ゴール付近にいるか否かでゴール探索シーケンスと移動シーケンスに移行する。

- 移動シーケンス



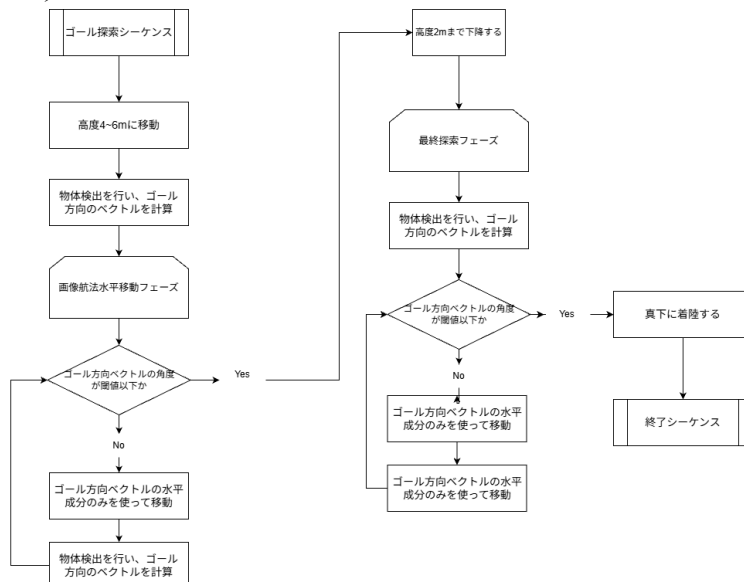
移動シーケンスは目標物上空まで飛行して移動するシーケンスである。LiDARの値を判断材料に、機体が想定よりも降下してしまった場合には機体を上昇させることにする。機体が不安定になった場合は待機シーケンスに移行する。

- ゴール探索シーケンス



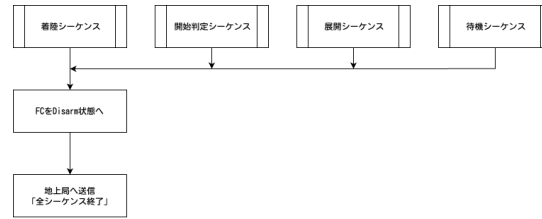
移動シーケンス後、ゴールを見つけ、画像航法に移行するためのシーケンスである。

- 画像航法シーケンス



GPS誤差や風などの外乱対策として、ゴールに接近していくシーケンスを複数回に分けた。方向ベクトルを用いて移動させる際に、画像方向奥行き情報が足りないことが大きな問題となった。LiDARの値を用いることで解決することができ、再現性のある画像航法シーケンスとなった。

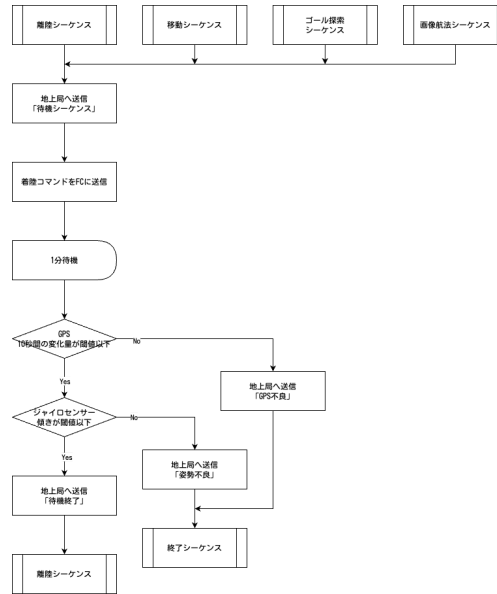
- 終了シーケンス



終了シーケンス

終了シーケンスではフライトコントローラーをDisarm状態にして全シーケンスを終了する。

- 待機シーケンス



待機シーケンス

待機シーケンスは機体が不安定であることを検知した時に一度着陸させるセーフティーモードである。姿勢の安定性を判定してもう一度離陸できるか終了する必要があるか判定する。

- githubリンク

本プロジェクトでは、githubを活用し、Ver管理や実験・本番でのコーディングログに役立てることができた。以下にリンクを示す。

githubリンク：<https://github.com/ARLISS2025/SEIKOKU>

## 第4.5節 製作時に使用した機材・サービス・ツール

### 第4.5.1項 製作時に使用した機材・サービス

分類	名称・型番	入手先・参考情報等	備考
3Dプリンタ	Bambu Lab X-1 carbon	東京大学大学院工学系研究科 国際工学教育推進機構 ものづくり部門	
レーザーカッター	Epilog FusionPro32	同上	
3Dプリンタ	BambuLab A1 mini	東京大学航空宇宙技術研究会(UTAT)	
EDAツール	KiCAD 9.0		

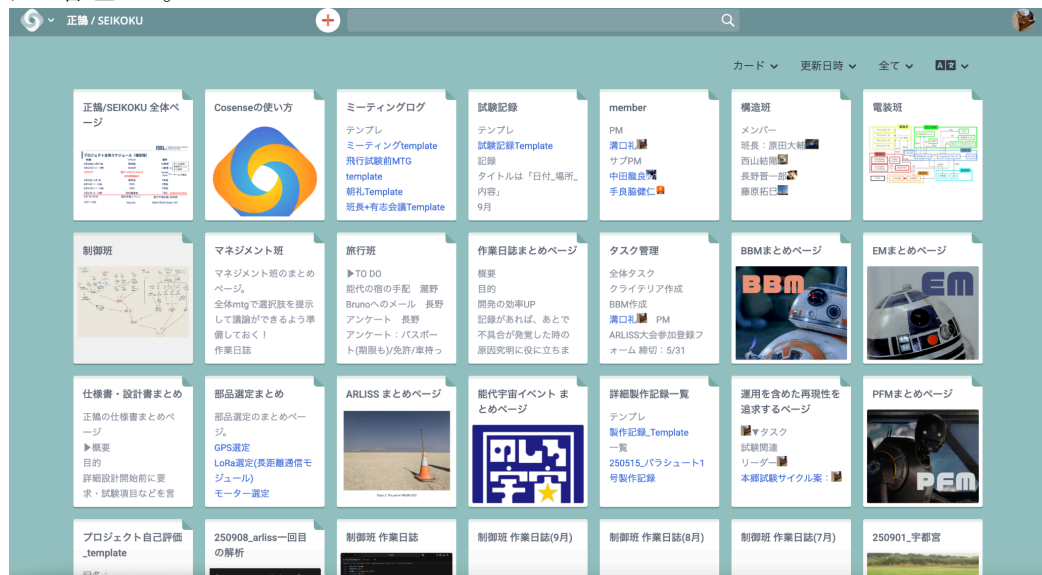
## 第4.5.2項 その他設計制作で利用したツール・アプリケーション

### 第4.5.2.1目 コミュニケーション

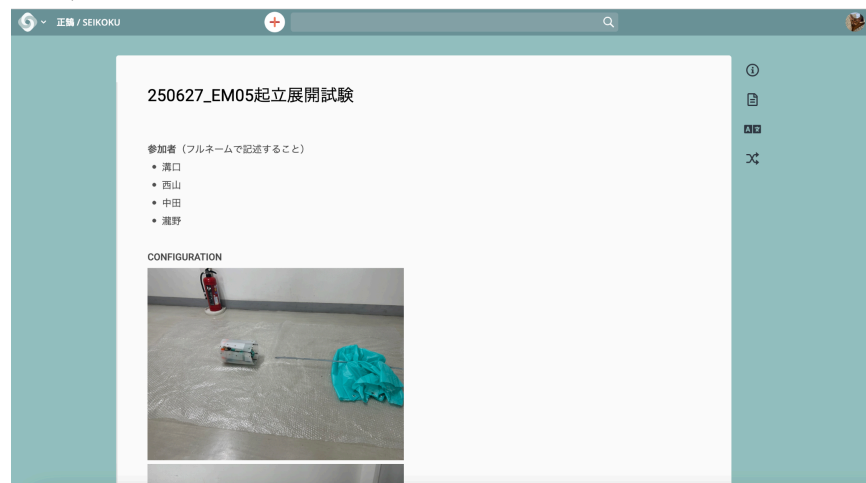
- Slack
  - コミュニケーションツールとして利用した。

### 第4.5.2.2目 ドキュメンテーション

- Scrapbox
  - ミーティング議事録、試験記録、作業日誌、技術的なドキュメンテーションなどはすべてこれで管理した。



- 特に、試験開始前に試験記録のページを作成し、以下の手順を徹底することでドキュメンテーションの習慣化をはかった。
  - 事前に記入するもの
    - 参加者



- 試験機体、サクセスクライテリア

**試験機体**

- EM05 起立機構
- EM04 飛行機体

**確認事項**

- [試験持ち物リスト](#) (—実施日のシートを原本からコピーして使用)
- [充電リスト](#)
- [飛行前チェックリスト](#)

**サクセスライテリア**

- MINIMUM**
  - EM05起立機構が組み立て終わる
  - 起立機構・飛行機体・エンベロープを組み立てた状態で、ラズパイでテグス溶断ができる
- FULL**
  - エンベロープが開く
  - 部々継ぎが固まる

● 試験前タスクの整理、試験手順

- 起立機構が展開する
- 腕が展開する
- EXTRA**
  - 飛行機体が垂直に起立する 
    - 機体と起立機構の土台をくっつけていなかったため

**試験までのタスク**

- EM05起立機構を完成させる
  - ①設計
    - ポリカでの補強
    - EM05フライト機構に合わせたネジ穴位置などの修正
  - ②材料準備
    - ポリカ
    - カーボン棒
  - ③製作・組み立て
    - ポリカのレーザーカット
    - 土台の3Dプリント
    - カーボン棒カット
  - ④試験準備
    - テグスを巻く
    - ラズパイの準備

**試験の流れ**

- 溶断機構にテグスを通す

■ 試験中に記入するもの

● 試験結果を時刻・動画・画像とともに記入

- 溶断機構をマステで機体のアルミ柱に固定する
- .

**試験結果**

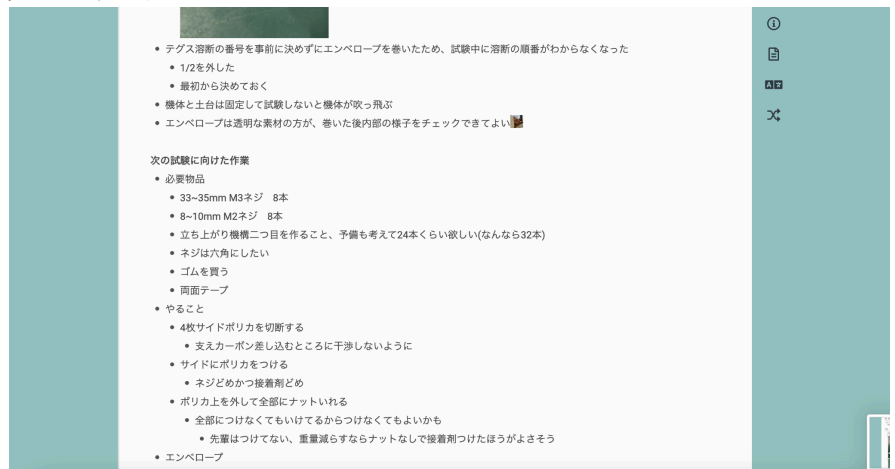
- ラズパイからのニクロム線溶断に成功
- 250627\_23:05
- 4.2V→3.9V

■ 試験後記入するもの

● 事故・故障とその原因、反省



- 次回試験に向けたタスクの整理



#### 第4.5.2.3目 構造

- **Fusion360**
  - 機体設計に用いた。

#### 第4.5.2.4目 電装

- **KiCAD**
  - 基板設計に用いた。

#### 第4.5.2.5目 制御

- 箱庭ドローンシミュレータ
  - 制御ロジックの検証や画像認識・画像航法の試験を行った。詳しくは、第8.3.2項に記した。
  - リンク: <https://hakoniwa-lab.net/products/drone/>

## 第5章 システム試験

### 第5.1節 レギュレーションを満たすためのシステム試験

#### (RV1) 質量試験

- 目的  
CanSatがキャリア格納時にレギュレーションにある規定重量を下回ることを確認する。(R1)
- 試験内容  
CanSat本体とパラシュートの合計質量を計測する。
- 結果  
下の画像に示すように、はかりにCanSatとパラシュートの全てを載せて総重量を測ったところ、1035gであり、レギュレーションの1050g以下であることを確認した。 図に質量測定結果の結果の様子を示す。



図:パラシュートとCanSatの質量



#### (RV2) 機体収納試験

- 目的  
CanSatとパラシュートを合わせて直径146mm高さ240mmの円筒キャリア内に収まることを確認する。(R1)

- 試験内容  
直径146mm高さ240mmの円筒キャリアを3Dプリンターで作成し、そこにCanSatとパラシュートが収まることを確認する。
- 結果  
以下の写真の通り、キャリアに収まることが確認できた。



	写真	測定値
--	----	-----

直径		145mm
高さ		239mm

- 結果  
以下の表に示すように、CanSatをキャリアに収納できることが確認できた。なお、収納の際にはパラシュートを押し込む必要がある。
- 

日付	実験動画	収納
7/30	<a href="https://youtube.com/watch?v=2FZwuVNbz04">https://youtube.com/watch?v=2FZwuVNbz04</a>	押し込むことでキャリア内に収まることが確認できた。

### (RV3) GPSデータダウンリンク試験

- 目的  
地上局がドローンからGPSを取得できていることを確認する。(R5)
- 試験内容  
機体を手で持って移動させながら機体のGPS座標を地上局に送信し、受信したものをgooglemapに表示した。
- 結果  
無線の送付、終了ともに成功している。  
地上局(実験中同じ場所)と受信した機体のGPS座標はそれぞれ以下であり、両者の距離は555mであった。  
35° 41' 25.5926" N 139° 45' 27.4583" E  
35° 41' 23.7998" N 139° 45' 5.5001" E  
  
試験の解説動画は以下である。(動画内では東大内と説明したが、実際は校外で行っている。)  
<https://youtu.be/m39TbmfB4Y>  
  
試験での受信側ログ全文は以下である。  
<https://drive.google.com/file/d/1-ikMt2dizZ1bvITAJeYyuKTzRjJZYsqx/view?usp=sharing>
- 考察  
以上の結果から、無線機を用いて地上局がドローンのGPS座標を取得できることを確認できた。

### (RV4) 準静的荷重試験

- 目的  
打ち上げ時に鉛直方向にかかる準静的荷重10Gに10秒間CanSatが耐えることを確認する。(R2)
- 試験内容  
紐の先につけた機体を10秒間回転させたのち、機体が正常に動作するかを確かめる。半径0.65mの円周上で1秒間に2回転させることで遠心加速度を与えた。この条件下で、 $a=r\omega^2 \approx 10G$ である。
- 結果  
以下の動画に示すように、10秒間の回転試験後、収納 → 放出 → 着地 → 起立展開 → 離陸 → GPS 航法 → 画像航法 → 着陸のシーケンスを問題なく実施できた。

日付	実験動画
9/1	<a href="https://youtu.be/Es_E_vBujww">https://youtu.be/Es_E_vBujww</a>

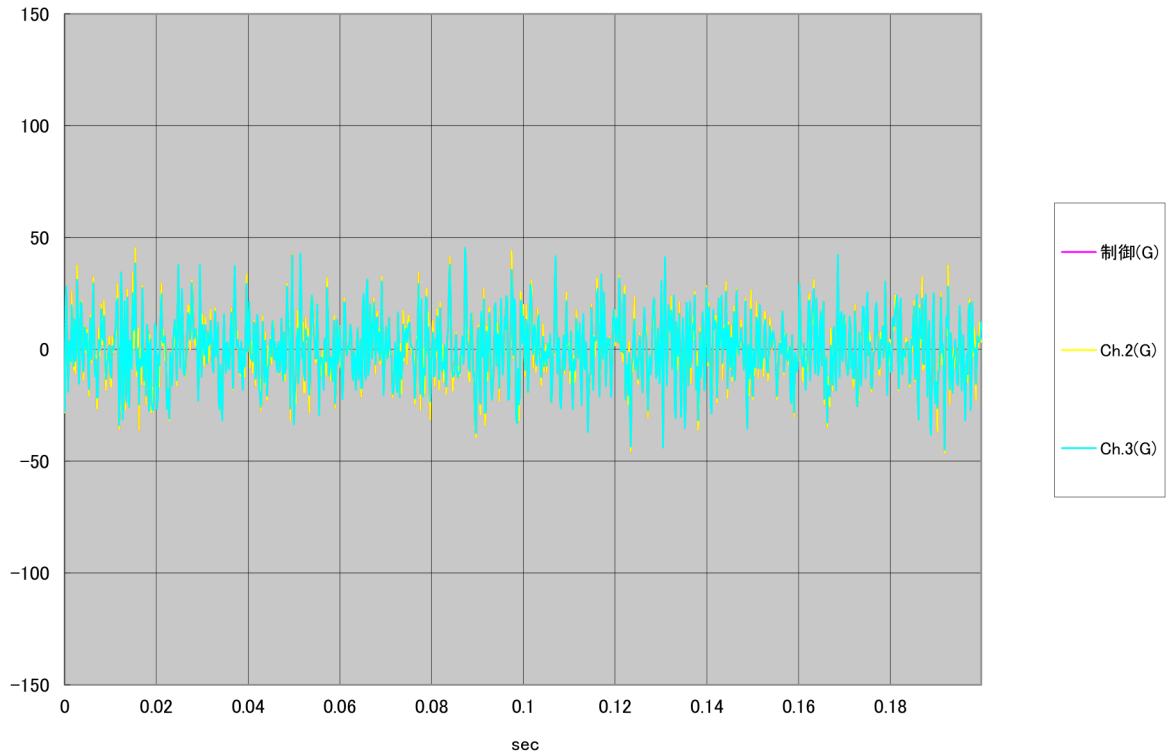
なお、本試験ではパラシュート投下は行っていないが、これは飛行場の規制上、落下試験を実施できなかったためである。  
また、安全率に相当する余裕として、実際には0.66mの紐を用いており、想定値0.65 m より大きな荷重が加わっている。

<p>紐の長さ</p>		<p>紐の長さは、0.66mであった。</p>
<p>回転試験の様子</p>		<p>1秒間に2回のペースで回転させ、10Gを10秒間与えた。</p>

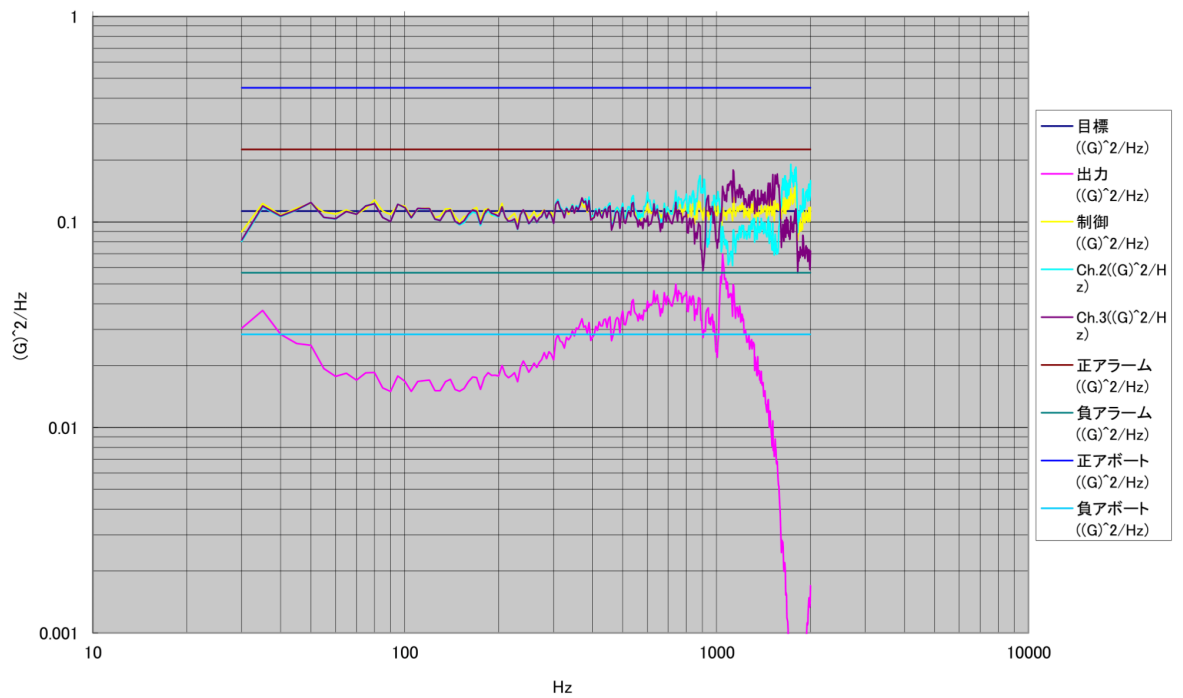
- 考察  
以上の試験により、CanSatは準静的荷重10Gに耐え、ミッションを遂行できることが確認された。

## (RV5) 振動試験

- 目的  
CanSatが打上げ中の衝撃および振動を受けた後もその機能を維持することを確認する。(R3)
- 試験内容  
正弦波振動30Hz～2000Hzまで15G、または同等のランダム振動に機体が耐えられることを確認する。後述の分離衝撃試験の後、機体に損傷やねじの緩みなどがないか、また搭載したセンサ類が機能するかを確認する。
- 結果  
ランダム振動試験の結果、機体は構造・回路ともにネジの緩みやPLAの破壊などの損傷なく、変化はなかった。加えた振動の様子は以下のグラフで示す。



応答PSDは以下のグラフのようであった。



動画は以下のリンク。

<https://youtube.com/watch?v=D3428GBU-V8?feature=share>

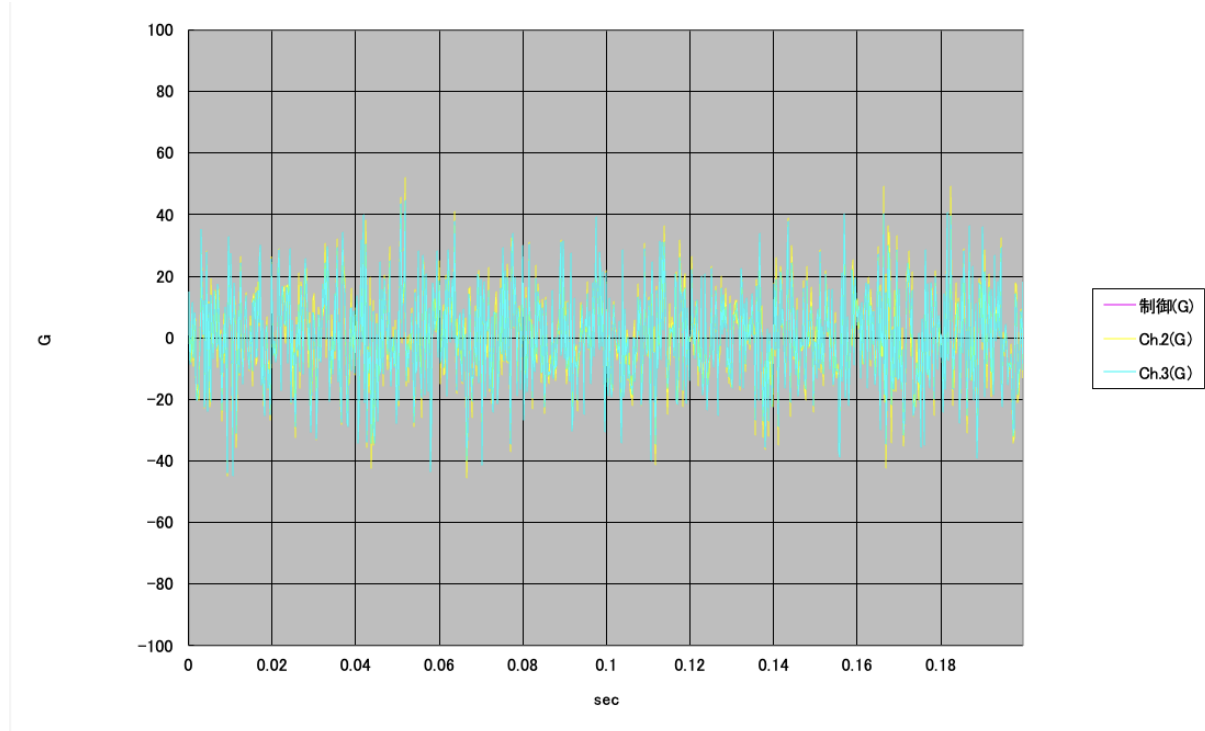
また、後述の分離衝撃試験を実施後に、ニクロム線、モーターの動作確認を行ったところ、すべて正常に作動した。GPSについてもQGCを通して位置情報が取れていることが確認できた。

- 考察  
本番で使用する機体は、ネジ止め部を外れないように接着剤で固定するため、さらに振動・衝撃が強くなると考えられる。接着剤にはアロンアルファまたはプラスチック

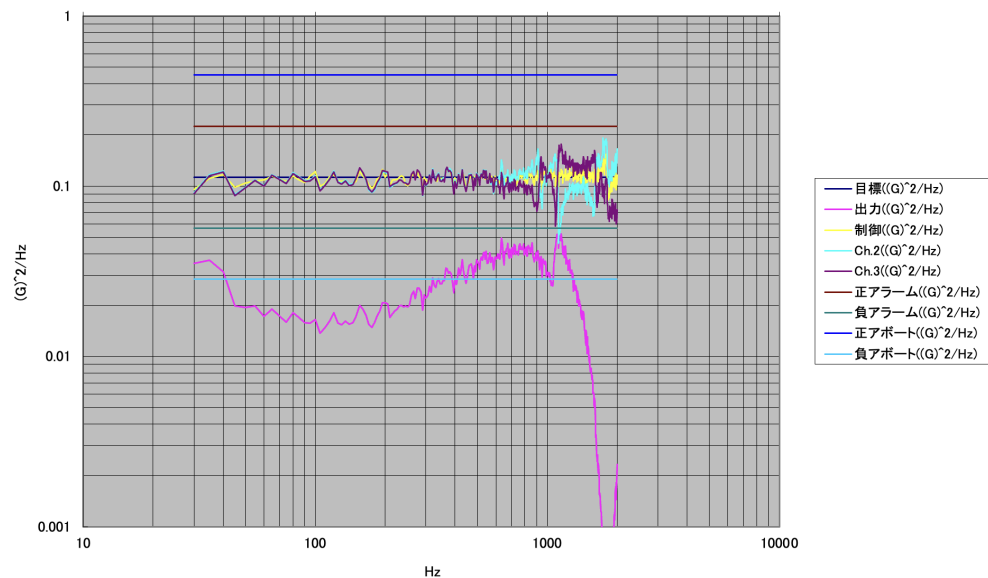
用のロックタイトを使用する。またナットについても場合によってはナイロンナットに交換する可能性がある。

機体が振動試験装置に入っていることを示すため、試験装置へセットする様子を動画で提出する必要がある。以下に再実験の結果を載せる。

- 再試験内容  
正弦波振動30Hz～2000Hzまで15G、または同等のランダム振動に機体が耐えられることを確認する。後述の分離衝撃試験の後、機体に損傷やねじの緩みなどがないか、また搭載したセンサ類が機能するかを確認する。
- 結果  
ランダム振動試験の結果、機体には構造・回路共にねじの緩みや損傷は一切なかった。加えた振動の様子はグラフで以下に示す。



応答PSDは以下の通り。



動画は以下のリンク。

<https://youtu.be/kD8HDB1Mak>

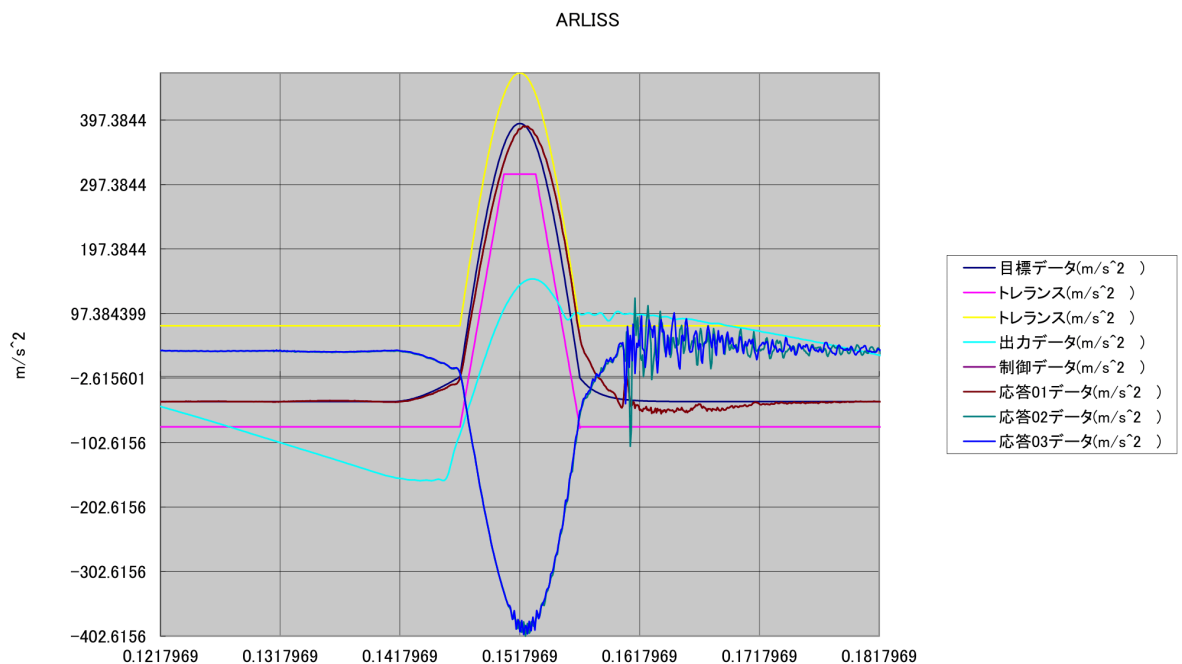
また、後述の分離衝撃試験を実施後に、放出判定から起立展開(光センサ・気圧セ

ンサ・ニクロム線の動作確認を含む)、LiDAR・カメラ・モーターの動作確認を行ったところ、全て正常に作動した。実験の様子は全て上記動画に含まれている。GPSについても、実験後に開けた場所に移動したところ取得できることが確認できた。出力値が1800Hz付近で0となっているのは、加振器の治具の共振点が1800Hz付近にあり、共振による発散を抑え制御値を一定に保つためである。

- 考察  
考察については初回実験と同様。

## (RV6)分離衝撃試験

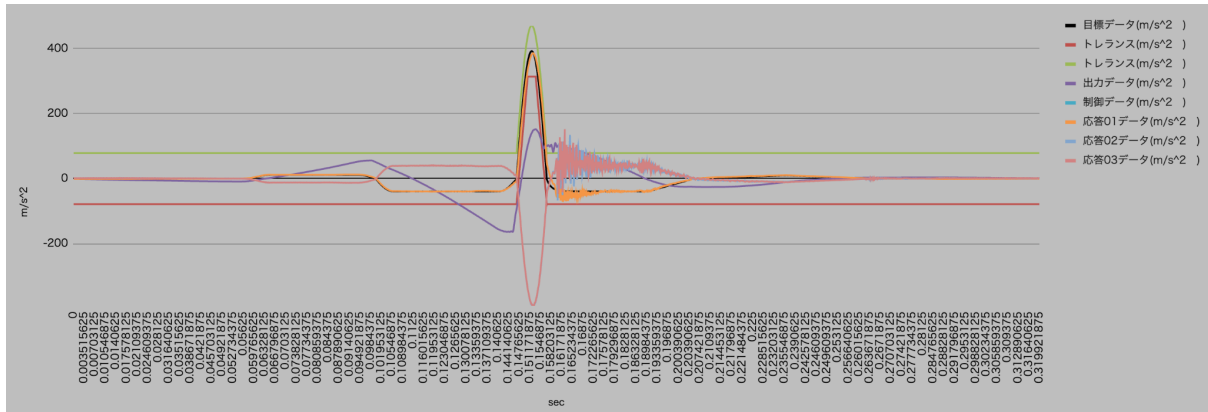
- 目的  
CanSatがロケットから分離される時の衝撃を受けた後もその機能を維持することを確認する。(R4)
- 試験内容  
ロケットから分離するときに鉛直方向にかかる撃荷重40Gに耐えることを確認する。
- 結果  
機体は構造・回路ともに損傷せず、振動試験で前述の通り正常に動作した。加えた振動の様子は以下のグラフで示す。横軸の単位は秒である。



<https://youtube.com/watch?v=TDv6GqAPQXo>

動画は以上のリンク

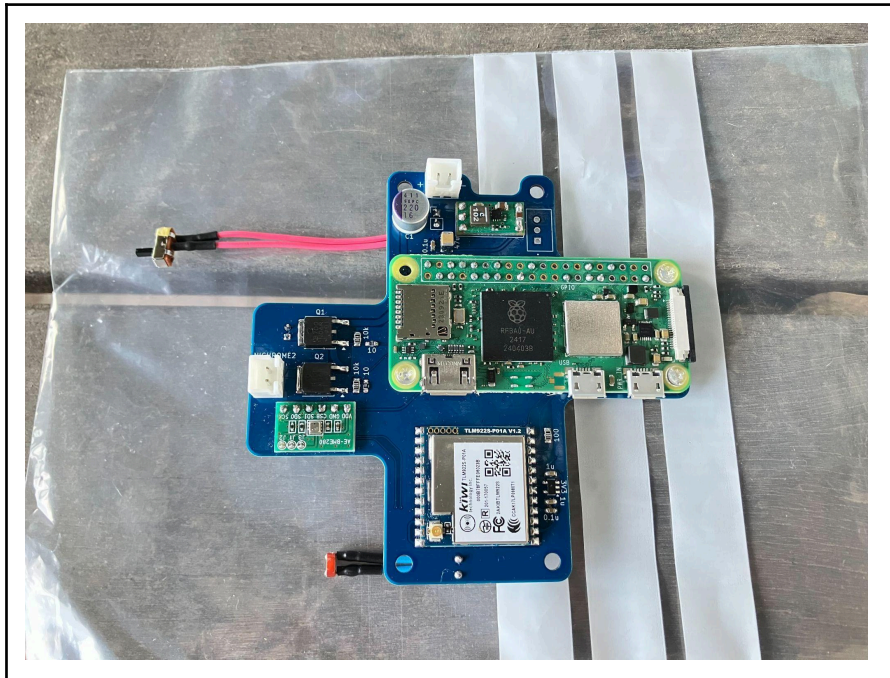
また、8月に振動試験とともに再試験を行った。再試験の様子は以下のグラフの通り。



動画については振動試験の結果に添付している。振動試験で前述の通り、機体には一切の損傷や変形がみられなかった。

- 考察

Raspberry Piのピンヘッダが基板から少しだけ抜けているように見えたが、動作に問題はなかった。基板に直接はんだ付けすることで対処する。以下が基板に直接はんだ付けした様子である。



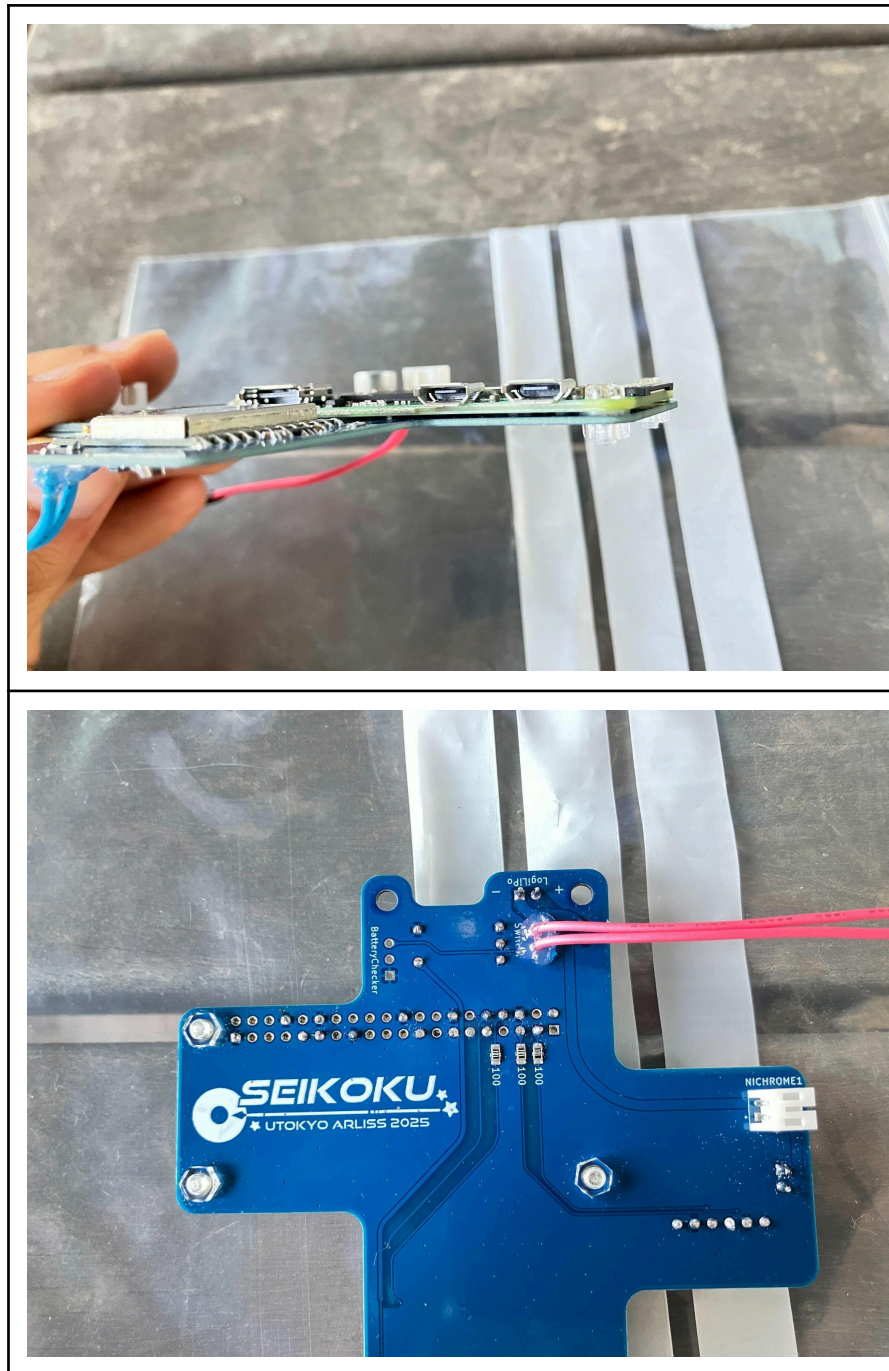
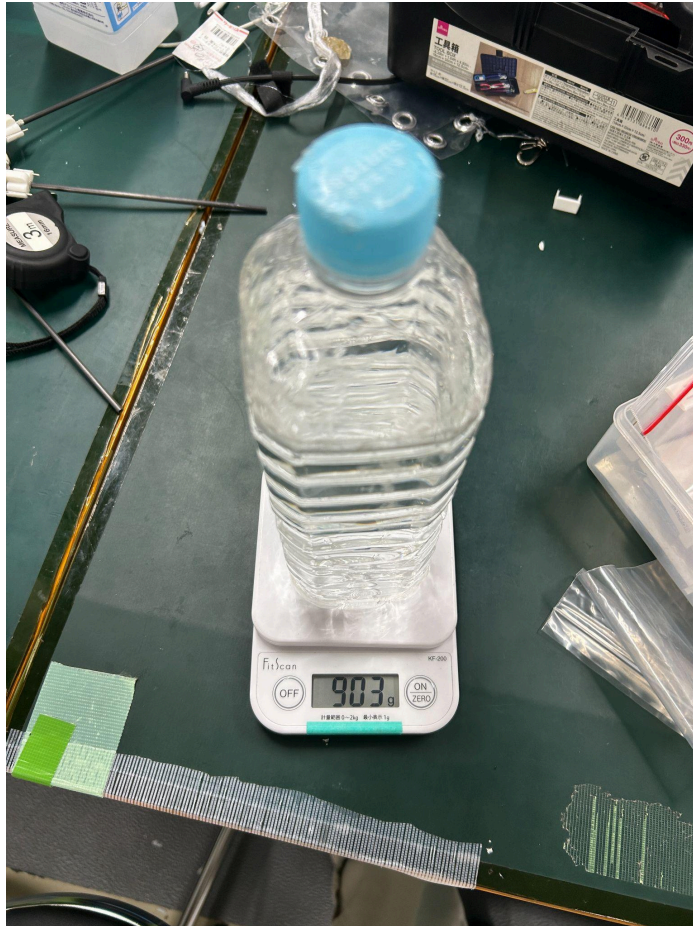


図: Raspberry Piを直接基板にはんだ付けした様子

### (RV7) パラシュート性能試験

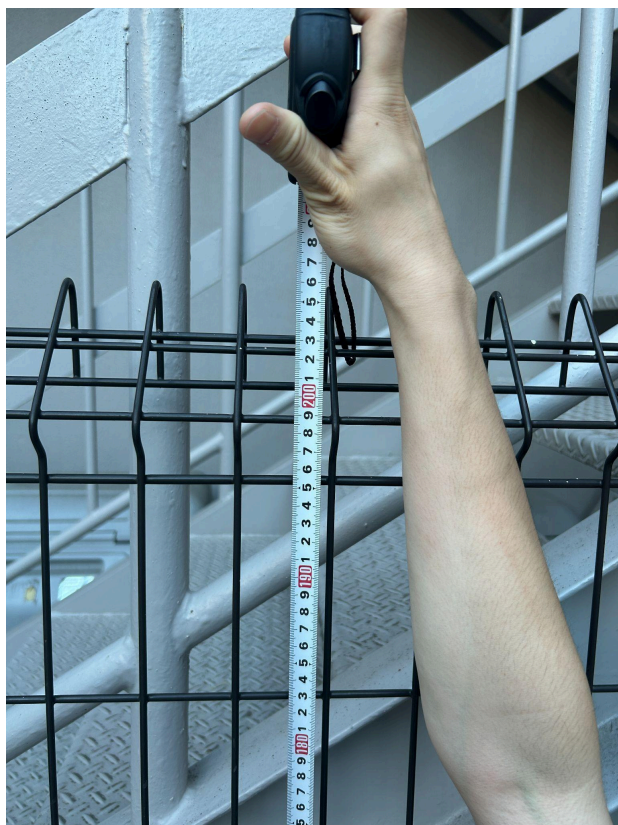
- 目的  
CanSatがパラシュートで降下するときの終端速度が4~6m/sとなっていることを確認する。(R6)
- 試験内容  
約12mの高さからパラシュートメジャー部分を除いた機体重量900gとほぼ同じ重さ903gのペットボトルを接続させて投下した。重量は以下の写真の通りである。

後日計測であり、一連の動画に収めたものはないが後述の開傘衝撃試験によりメジャーが破損したためこれを期限内に提出することが不可能となった。



その後動画内で奥にある塀の高さを通過した時刻から落下直前の時刻までを計測し、塀の高さ/計測時間により終端速度を計算する。

- 試験結果  
塀の高さは以下の画像の通り計測を行い、2.06mとした。



動画では5.00s~5.40sを切り出して計算を行った。

実験の動画は以下のリンク上である。  
<https://youtube.com/watch?v=8WXZ4BqC0cg?feature=share>  
計測方法を見直し再度計測を行った。

結果として終端速度 $v_f$ は以下の計算の通り5.15 m/sとなった。

$$v_f = 2.06 / (5.40 - 5.00) = 5.15 \text{ m/s}$$

- 結論  
パラシュートの終端速度はレギュレーションの範囲内であることが確認された。

## (RV8)開傘衝撃試験

- 目的  
パラシュート開傘時に想定される衝撃加速度50Gに対し、機体構造、起立展開機構、各種センサ、およびパラシュート系統(エンベロープ、メジャー、テグス)が十分に耐え得ることを確認する。
- (R7)
- 試験内容  
本試験は以下の2つに分けて実施した。

試験①(機体・起立展開機構・センサ):

開傘時の衝撃50Gを機体構造、起立展開機構、各種センサに与えるため、機体を自由落下させた。

スマートフォン(240fps撮影)で動画を取得し、接地時間を解析した結果、接地時間 $\Delta t=0.005 \text{ s}$ (1/200 s)であった。(後述)

このとき50Gを与えるために必要な加速度は $500 \text{ m/s}^2$ ( $\approx 50\text{G}$ )、速度変化 $\Delta v=2.5 \text{ m/s}$ に相当する。

これより自由落下高さは $h=0.318 \text{ m}$ と算出される。

実際には安全側を見込んで0.5 mの高さから自由落下させた。

試験②(パラシュート系統):

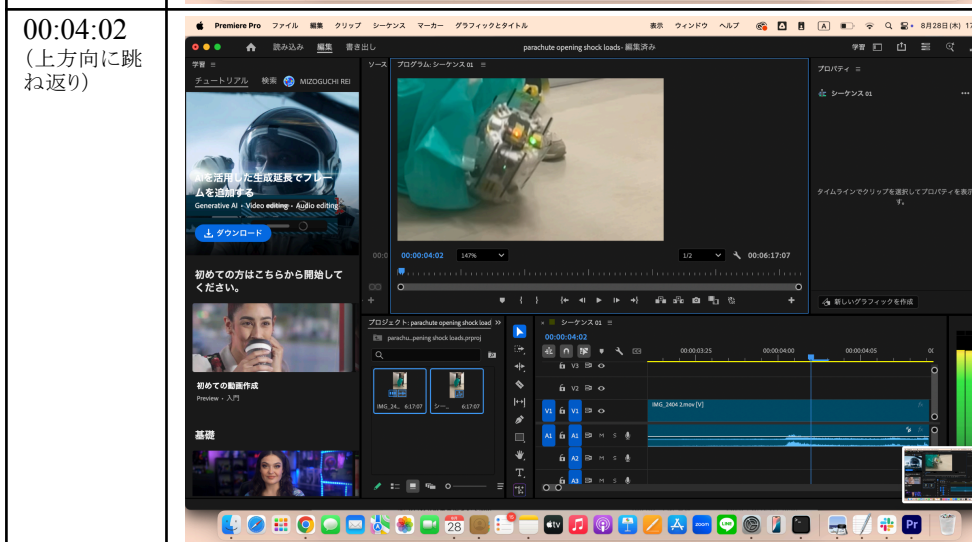
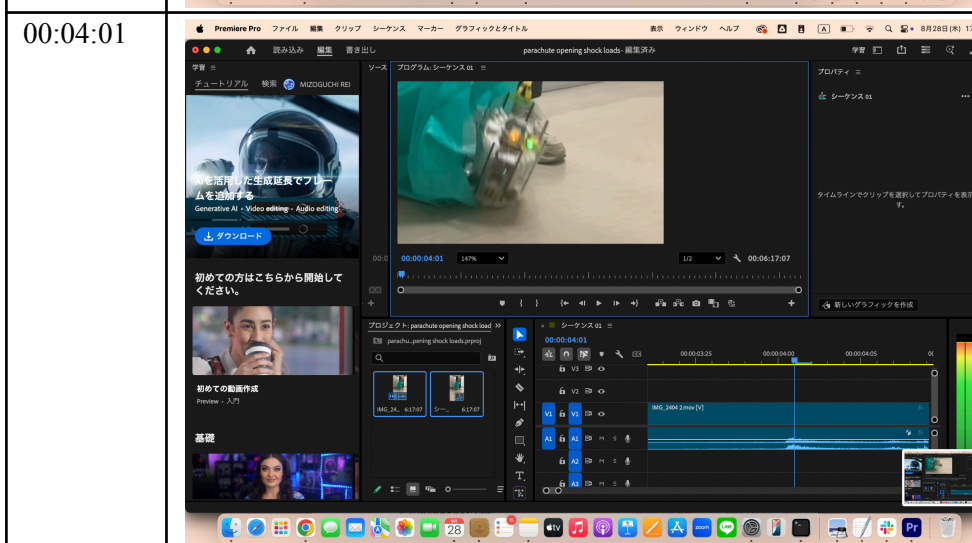
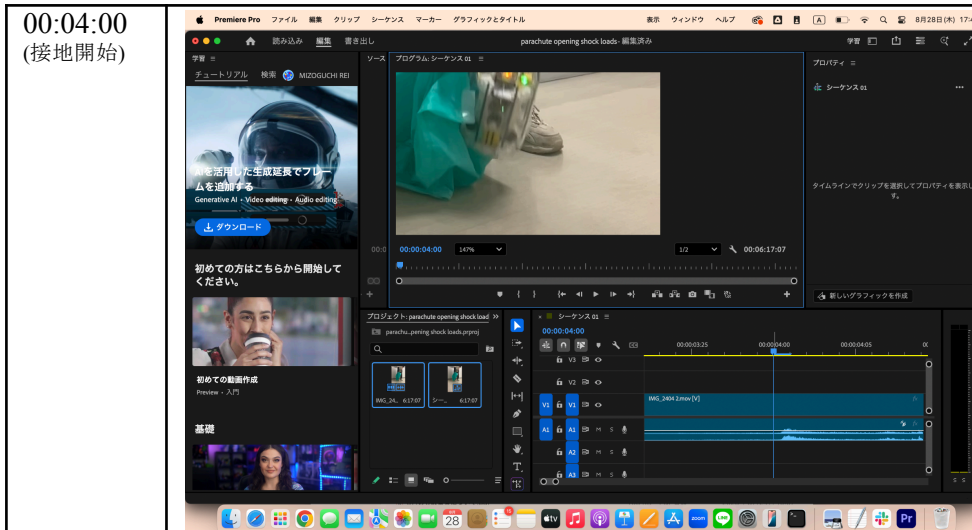
開傘時に50Gが作用すると想定されるパラシュート、エンベロープ、メジャー、テグスに対し、50 kg相当の荷重を静的に与えて検証した。

パラシュートの開傘時間は一般に約0.1s以下であり(飯島、1956)、動画解析より開傘時間は約0.09sであった。

そのうち衝撃の立ち上がり時間を0.05 sと見積もった。

よって「50kgの荷重を0.05s保持できれば耐久性あり」と判定基準を設定した。

- 結果  
試験①  
以下のように、接地時間は0.05秒であり、また上方向に跳ね返っていることから、50G以上の衝撃を機体にもたらされたと考えられる。



機体構造、起立展開機構、各種センサに損傷や機能不良は見られなかった。  
動画は以下。

8/25 | <https://youtu.be/12dPsBFVKUo>

試験②  
50kgの静荷重に0.12秒耐えた。動画は以下。

8/28 | <https://youtube.com/shorts/RECDqwO3TZk>

<p>00:04:16 6 (荷重かかり始め)</p>	
<p>00:04:28 8</p>	
<p>00:04:29 9 (テグス切断)</p>	

- 考察  
実験①および②の結果より、50Gの開傘衝撃に十分耐えると考えられる。
- 参考文献  
[パラシュート\(1956, 飯島\).pdf](#)

## (RV9) 無線性能試験

- 目的  
CanSatに実装された無線機が無線の送出をプログラムにより制御できることを確認する。また使用する無線機の周波数が自由に変更できることを確認する。(R2, R8, R9)
- 試験内容  
機体と地上局を離していき、最大通信距離を測定する。また、チャンネル変更が可能かどうか確かめる。  
1つめの実験では、無線モジュールをRaspberryPiから制御し、取得したGPS座標を地上局に送信する試験を行った。  
2つめの実験では、最大通信距離の測定のため、RV3GPSデータダウンリンク試験と同時に行った。詳しくはそちらをご確認いただきたい。
- 結果  
1つめの実験では、無線の送出、終了ともに成功した。  
無線モジュールはloraを用いて通信し、ATコマンドを用いて周波数を設定できる。  
以下の画像の通り、受信機との間で周波数を自由に変更することを確認しており、

```
> p2p get_freq
>> 926500000
> p2p set_freq 926000000
>> Ok
> p2p get_freq
>> 926000000
> p2p set_freq 926500000
>> Ok
```

実験ではこのコマンドを用いて周波数を操作するプログラムが内部に組み込まれている。

## (CRV1) End-to-end試験

- 目的  
フルサクセスまでのシーケンスがすべて自動で達成されることを確認する。(R10, R11, CR1)
- 試験内容  
我々が利用しているドローンの飛行場には高所から落下させる場所がなく、パラシュート落下などの試験を行うことが難しいので、試験を前後半で2分割し、それぞれ試験可能な場所で行う。前半は高所から落下させる場所がある東京大学で、後半はドローン飛行場で行う。  
前半: 放出・落下→開傘→着地→展開→離陸姿勢  
想定しているシーケンスは、  
1. キャリアからの放出

2. 放出判定
3. パラシュート展開
4. 着地
5. 着地判定
6. 起立展開(安全のため、コマンドを送り実施)
7. 腕展開(安全のため、コマンドを送り実施)

後半: 離陸姿勢→離陸→GPS航法→画像航法  
 想定しているシーケンスは、

1. 離陸機構からの離陸(指定高度4m)
2. GPSを用いて、目標のコーンに接近する。
3. 機体を回転させゴール探索を行う。(カメラで物体検出を行う)
4. 画像航法を開始し、ゴール地点の真上に移動させる。その後真下に着陸させる。

- 結果  
前半:

7/18	<a href="https://www.youtube.com/watch?v=oqrPVZVAKks">https://www.youtube.com/watch?v=oqrPVZVAKks</a>
------	---

想定していた挙動を示した。

後半:

7/29	<a href="https://youtu.be/ZxeBx0Idhfl">https://youtu.be/ZxeBx0Idhfl</a>
------	---

ログは制御履歴レポート作成試験にて添付する。  
 想定していた挙動を示し、目標物に十分接近することができた。

- 考察  
前半、後半の試験を通して十分にEnd to End試験を行うことができた。
- 追加試験  
前後半で一連の流れが分断されないよう、一部のシーケンスを重複させて追加試験を行った。追加試験のシーケンスは以下の通り。

1. 収納判定(光センサを手で隠す)
2. 放出判定(放出と同時に手を離す)
3. 着地
4. 着地判定
5. 起立展開腕展開
6. 離陸
7. GPS航法(往復300mの直線飛行を2回)
8. 画像航法(ARLISS 新ゴール形式で実施)

以下が試験結果である。なお、RV4準静的荷重試験と同時に行った。  
 CanSatの航続可能距離については、MV14航続距離検証試験を参照されたい。

9/1 (E2E追加試験)	<a href="https://youtu.be/Es_E_yBujww">https://youtu.be/Es_E_yBujww</a>
---------------	---

## (CRV2) 制御履歴レポート作成

- 目的  
制御履歴を保存する。(CR2)
- 試験内容  
End to End試験中に制御履歴ログを作成する。目標地点、現在地、制御指令値の推移を記録する。
- 結果  
ログファイルは以下の通り。takeoffの開始から終了、ゴール座標までの移動開始と終了、ゴール探索の開始と終了、画像航法の開始と終了、着陸の開始と終了について判定を行いログを残すことができる。

```
2025-07-29 17:44:54 - message.takeoff - INFO - running drone_connect()
2025-07-29 17:44:54 - message.takeoff - INFO - Waiting for drone to connect...
2025-07-29 17:44:55 - message.takeoff - INFO - Checking connection state...
2025-07-29 17:44:55 - message.takeoff - INFO - -- Connected to drone!
2025-07-29 17:44:55 - message.takeoff - INFO - running drone_arm()
2025-07-29 17:44:55 - message.takeoff - INFO - -- Waiting for drone to be ready to arm...
2025-07-29 17:44:56 - message.takeoff - INFO - Health: Health: [is_gyrometer_calibration_ok: False,
is_accelerometer_calibration_ok: False, is_magnetometer_calibration_ok: False, is_local_position_ok: True,
is_global_position_ok: False, is_home_position_ok: True, is_armable: False]
2025-07-29 17:44:57 - message.takeoff - INFO - Health: Health: [is_gyrometer_calibration_ok: True,
is_accelerometer_calibration_ok: True, is_magnetometer_calibration_ok: True, is_local_position_ok: True,
is_global_position_ok: True, is_home_position_ok: True, is_armable: True]
2025-07-29 17:44:57 - message.takeoff - INFO - ✓ Drone is ready to arm.
2025-07-29 17:44:57 - message.takeoff - INFO - -- Arming
2025-07-29 17:44:58 - message.takeoff - INFO - Drone is armed.
2025-07-29 17:44:58 - message.takeoff - INFO - running takeoff()
2025-07-29 17:44:58 - message.takeoff - INFO - Taking off...
2025-07-29 17:44:58 - message.takeoff - INFO - -- Waiting to reach 4.0m altitude with tolerance 0.2m...
2025-07-29 17:44:58 - message.takeoff - INFO - Current height: -0.10m (target: 4.0m)
2025-07-29 17:44:59 - message.takeoff - INFO - Current height: -0.05m (target: 4.0m)
2025-07-29 17:45:00 - message.takeoff - INFO - Current height: -0.14m (target: 4.0m)
2025-07-29 17:45:01 - message.takeoff - INFO - Current height: -0.05m (target: 4.0m)
2025-07-29 17:45:03 - message.takeoff - INFO - Current height: 0.91m (target: 4.0m)
2025-07-29 17:45:04 - message.takeoff - INFO - Current height: 2.31m (target: 4.0m)
2025-07-29 17:45:05 - message.takeoff - INFO - Current height: 3.31m (target: 4.0m)
2025-07-29 17:45:06 - message.takeoff - INFO - Current height: 3.77m (target: 4.0m)
2025-07-29 17:45:07 - message.takeoff - INFO - Current height: 3.90m (target: 4.0m)
2025-07-29 17:45:07 - message.takeoff - INFO - ✓ Target altitude reached: 3.90m
2025-07-29 17:45:07 - message.takeoff - INFO - Takeoff successful.
2025-07-29 17:45:10 - message.takeoff - INFO - running postural_stability_check()
2025-07-29 17:45:10 - message.takeoff - INFO - Checking postural stability...
2025-07-29 17:45:10 - message.takeoff - INFO - Posture (roll: 4.766129970550537, pitch: -0.922674298286438, yaw:
8.515966415405273)
2025-07-29 17:45:10 - message.takeoff - INFO - Angular velocity (roll: -0.06236857920885086, pitch:
0.022731196135282516, yaw: 0.0584416426718235)
2025-07-29 17:45:10 - message.takeoff - INFO - ✓ Angular velocity check passed.
2025-07-29 17:45:10 - message - INFO - ✓ Postural stability check passed.
2025-07-29 17:45:11 - message - INFO - 現在位置 (EKF2): 緯度35.7700214, 経度140.108194, 高度3.8630001544952393 m
2025-07-29 17:45:11 - message - INFO - -- Moving to goal position: [35.77019247055298, 140.1079803327809, 4] --
2025-07-29 17:45:32 - message - INFO - Goal position reached.
2025-07-29 17:45:33 - message - INFO - 現在位置 (EKF2): 緯度35.7701941, 経度140.1079799, 高度4.0420002937316895 m
2025-07-29 17:45:33 - message - INFO - -- Starting goal search
2025-07-29 17:45:36 - message - INFO - found object
2025-07-29 17:45:36 - message - INFO - approaching goal position
2025-07-29 17:45:37 - message - INFO - Yaw: 89.52947235107422
2025-07-29 17:45:54 - message - INFO - Goal position reached.
2025-07-29 17:45:57 - message - INFO - 現在位置 (EKF2): 緯度35.770226699999995, 経度140.1079809, 高度4.01200008392334 m
2025-07-29 17:45:57 - message - INFO - -- Landing
2025-07-29 17:45:57 - message - INFO - Waiting for drone to land...
2025-07-29 17:46:10 - message - INFO - Drone has landed.
```

また、現在地、速度ベクトル、ヨー角についても別ファイルにログとして保存できた。  
以下の通り。

2025-07-29 17:44:58 - Position: (35.770019999999999995, 140.1081958, -0.082000022649765)  
2025-07-29 17:44:59 - Position: (35.770020599999999995, 140.1081941, -0.04900002443790436)  
2025-07-29 17:45:01 - Position: (35.7700208, 140.1081922999999998, -0.15300001204013824)  
2025-07-29 17:45:02 - Position: (35.7700198, 140.1081932999999998, 0.19100001454353333)  
2025-07-29 17:45:03 - Position: (35.7700195, 140.1081951, 1.6500000953674316)  
2025-07-29 17:45:05 - Position: (35.7700199, 140.108195, 3.0300002098083496)  
2025-07-29 17:45:06 - Position: (35.770020599999999995, 140.1081929, 3.7220001220703125)  
2025-07-29 17:45:07 - Position: (35.770020099999999996, 140.1081931, 3.9040002822875977)  
2025-07-29 17:45:09 - Position: (35.770021199999999995, 140.1081918999999998, 3.8960001468658447)  
2025-07-29 17:45:10 - Position: (35.770020699999999996, 140.1081931, 3.8630001544952393)  
2025-07-29 17:45:11 - Position: (35.770022, 140.1081948, 3.886000156402588)  
2025-07-29 17:45:13 - Position: (35.770021799999999995, 140.1081969, 4.006999969482422)  
2025-07-29 17:45:14 - Position: (35.7700296, 140.1081841999999998, 4.154000282287598)  
2025-07-29 17:45:15 - Position: (35.7700622, 140.1081286999999998, 4.1020002365112305)  
2025-07-29 17:45:17 - Position: (35.7701079, 140.1080793999999998, 4.0420002937316895)  
2025-07-29 17:45:18 - Position: (35.7701525, 140.1080831, 4.019999980926514)  
2025-07-29 17:45:19 - Position: (35.7701785, 140.1080062, 4.0)  
2025-07-29 17:45:21 - Position: (35.770189099999999996, 140.1079931999999998, 4.0410003662109375)  
2025-07-29 17:45:22 - Position: (35.770192099999999996, 140.1079859999999998, 4.082000255584717)  
2025-07-29 17:45:23 - Position: (35.7701923, 140.1079821, 4.090000152587891)  
2025-07-29 17:45:25 - Position: (35.7701902, 140.1079802, 4.113000329213818)  
2025-07-29 17:45:26 - Position: (35.7701898, 140.1079795, 4.088000297546387)  
2025-07-29 17:45:27 - Position: (35.7701914, 140.1079805, 4.039000034332275)  
2025-07-29 17:45:29 - Position: (35.7701917, 140.1079789, 4.070000171661377)  
2025-07-29 17:45:30 - Position: (35.7701926, 140.1079789, 4.056000232696533)  
2025-07-29 17:45:31 - Position: (35.7701936, 140.1079784, 4.062000274658203)  
2025-07-29 17:45:33 - Position: (35.7701941, 140.1079797999999998, 4.034000396728516)  
2025-07-29 17:45:34 - Position: (35.7701937, 140.1079782, 4.104000091552734)  
2025-07-29 17:45:35 - Position: (35.770193899999999995, 140.1079781, 4.11400032043457)  
2025-07-29 17:45:37 - Position: (35.770194, 140.1079797999999998, 4.0470000408172607)  
2025-07-29 17:45:38 - Position: (35.7701932, 140.1079814, 4.05400037655029)  
2025-07-29 17:45:39 - Position: (35.770192699999999996, 140.1079809, 4.082000255584717)  
2025-07-29 17:45:41 - Position: (35.7701934, 140.107981, 4.05000076293945)  
2025-07-29 17:45:42 - Position: (35.770193, 140.10798, 4.031000137329182)  
2025-07-29 17:45:43 - Position: (35.770193299999999995, 140.10798, 4.010000228881836)  
2025-07-29 17:45:45 - Position: (35.7701929, 140.1079801, 4.031000137329182)  
2025-07-29 17:45:46 - Position: (35.7701929, 140.1079801, 4.009000301361084)  
2025-07-29 17:45:47 - Position: (35.770193, 140.1079801, 3.998000144958496)  
2025-07-29 17:45:49 - Position: (35.7701929, 140.1079797, 4.010000228881836)  
2025-07-29 17:45:50 - Position: (35.7701918, 140.1079796, 4.02400016784668)  
2025-07-29 17:45:52 - Position: (35.770192699999999996, 140.1079799, 4.1020002365112305)  
2025-07-29 17:45:53 - Position: (35.770213999999999996, 140.10798, 4.077000141143799)  
2025-07-29 17:45:54 - Position: (35.770224, 140.1079802, 4.044000148773193)  
2025-07-29 17:45:56 - Position: (35.770226, 140.1079815, 4.052999973297119)  
2025-07-29 17:45:57 - Position: (35.770226199999999996, 140.1079811999999998, 4.039999961853027)  
2025-07-29 17:45:58 - Position: (35.7702276, 140.1079797, 3.754000186920166)  
2025-07-29 17:46:00 - Position: (35.7702277, 140.1079791, 2.880000114440918)  
2025-07-29 17:46:01 - Position: (35.770227299999999995, 140.1079826999999998, 1.8160001039505005)  
2025-07-29 17:46:02 - Position: (35.770226199999999996, 140.107982, 0.906000018119812)  
2025-07-29 17:46:04 - Position: (35.770226799999999996, 140.107981, -0.001000000074974513)  
2025-07-29 17:46:05 - Position: (35.7702269, 140.1079806, -0.9390000700950623)  
2025-07-29 17:46:06 - Position: (35.7702266, 140.1079823, -1.8530000448226929)  
2025-07-29 17:46:08 - Position: (35.7702272, 140.1079819, -2.745000123977661)  
2025-07-29 17:46:09 - Position: (35.7702266, 140.1079815, -3.6480002043259277)  
2025-07-29 17:46:10 - Position: (35.770226799999999996, 140.1079814, -3.689000129699707)  
  
2025-07-29 17:44:58 - Speed: VelocityNed: [north\_m\_s: 0.08999999776482582, east\_m\_s: -0.02999999329447746, down\_m\_s: 0.04999999781976776]  
2025-07-29 17:44:59 - Speed: VelocityNed: [north\_m\_s: 0.0, east\_m\_s: -0.03999999910593833, down\_m\_s: 0.0700000029802322]  
2025-07-29 17:45:01 - Speed: VelocityNed: [north\_m\_s: -0.08999999612569809, east\_m\_s: 0.2399999946358197, down\_m\_s: 0.829999983186995]  
2025-07-29 17:45:02 - Speed: VelocityNed: [north\_m\_s: 0.08999999776482582, east\_m\_s: 0.02999999329447746, down\_m\_s: -1.2200000206102295]  
2025-07-29 17:45:03 - Speed: VelocityNed: [north\_m\_s: 0.11999999732179099, east\_m\_s: -0.0399999910593833, down\_m\_s: -0.72999995046082616]  
2025-07-29 17:45:05 - Speed: VelocityNed: [north\_m\_s: -0.02999999329447746, east\_m\_s: -0.1099999948395355, down\_m\_s: -0.1599999964232123]  
2025-07-29 17:45:07 - Speed: VelocityNed: [north\_m\_s: 0.02999999329447746, east\_m\_s: -0.01999999952965164, down\_m\_s: -0.0599999986889549]  
2025-07-29 17:45:09 - Speed: VelocityNed: [north\_m\_s: -0.1099999948395355, east\_m\_s: 0.02999999329447746, down\_m\_s: -0.1400000005904654]  
2025-07-29 17:45:11 - Speed: VelocityNed: [north\_m\_s: -0.1499999910593833, east\_m\_s: 0.1599999946232123, down\_m\_s: -0.1899999976158142]  
2025-07-29 17:45:13 - Speed: VelocityNed: [north\_m\_s: 0.02999999329447746, east\_m\_s: 0.17999999225139618, down\_m\_s: -0.1599999964232123]  
2025-07-29 17:45:15 - Speed: VelocityNed: [north\_m\_s: 2.5799999237068547, east\_m\_s: -3.0299999713097785, down\_m\_s: 0.0999999940395352]  
2025-07-29 17:45:17 - Speed: VelocityNed: [north\_m\_s: 3.0299999237068547, east\_m\_s: -3.0299999713097785, down\_m\_s: 0.04999999781976776]  
2025-07-29 17:45:18 - Speed: VelocityNed: [north\_m\_s: 2.4700000286102295, east\_m\_s: -2.490000009536743, down\_m\_s: 0.0]  
2025-07-29 17:45:19 - Speed: VelocityNed: [north\_m\_s: 1.069999932427979, east\_m\_s: -1.089999961853027, down\_m\_s: 0.0]  
2025-07-29 17:45:21 - Speed: VelocityNed: [north\_m\_s: 0.20999999344348907, east\_m\_s: -0.7999999523126282, down\_m\_s: 0.04999999781976776]  
2025-07-29 17:45:22 - Speed: VelocityNed: [north\_m\_s: 0.0599999986889549, east\_m\_s: -0.349999940395355, down\_m\_s: 0.0700000029802322]  
2025-07-29 17:45:23 - Speed: VelocityNed: [north\_m\_s: -0.1299999964232123, east\_m\_s: -0.1299999964232123, down\_m\_s: 0.0799999921180606]  
2025-07-29 17:45:25 - Speed: VelocityNed: [north\_m\_s: -0.17000000178813934, east\_m\_s: -0.0700000029802322, down\_m\_s: 0.08999999612569809]  
2025-07-29 17:45:26 - Speed: VelocityNed: [north\_m\_s: 0.1199999973179099, east\_m\_s: 0.0599999986889549, down\_m\_s: 0.08999999612569809]  
2025-07-29 17:45:27 - Speed: VelocityNed: [north\_m\_s: 0.0, east\_m\_s: -0.03999999910593833, down\_m\_s: 0.0]  
2025-07-29 17:45:29 - Speed: VelocityNed: [north\_m\_s: 0.0700000029802322, east\_m\_s: -0.07999999021180606, down\_m\_s: 0.0599999986889549]  
2025-07-29 17:45:30 - Speed: VelocityNed: [north\_m\_s: 0.1999999887907104, east\_m\_s: -0.0700000029802322, down\_m\_s: 0.0199999952965164]  
2025-07-29 17:45:31 - Speed: VelocityNed: [north\_m\_s: -0.08999999612569809, east\_m\_s: 0.00999999776482582, down\_m\_s: 0.0199999952965164]  
2025-07-29 17:45:33 - Speed: VelocityNed: [north\_m\_s: -0.02999999329447746, east\_m\_s: -0.08999999776482582, down\_m\_s: 0.0399999910593833]  
2025-07-29 17:45:35 - Speed: VelocityNed: [north\_m\_s: 0.0399999910593833, east\_m\_s: -0.1499999910593833, down\_m\_s: 0.0399999910593833]  
2025-07-29 17:45:37 - Speed: VelocityNed: [north\_m\_s: 0.0, east\_m\_s: 0.0799999921180606, down\_m\_s: 0.0700000029802322]  
2025-07-29 17:45:39 - Speed: VelocityNed: [north\_m\_s: -0.08999999776482582, east\_m\_s: 0.0700000029802322, down\_m\_s: 0.04999999781976776]  
2025-07-29 17:45:41 - Speed: VelocityNed: [north\_m\_s: -0.0700000029802322, east\_m\_s: 0.0, down\_m\_s: 0.0199999952965164]  
2025-07-29 17:45:43 - Speed: VelocityNed: [north\_m\_s: 0.0599999986889549, east\_m\_s: -0.02999999329447746, down\_m\_s: 0.0700000029802322]  
2025-07-29 17:45:44 - Speed: VelocityNed: [north\_m\_s: 0.0, east\_m\_s: -0.08999999612569809, down\_m\_s: 0.0599999986889549]  
2025-07-29 17:45:46 - Speed: VelocityNed: [north\_m\_s: 0.04999999781976776, east\_m\_s: -0.1099999948395355, down\_m\_s: 0.0599999986889549]  
2025-07-29 17:45:47 - Speed: VelocityNed: [north\_m\_s: 0.08999999776482582, east\_m\_s: -0.07999999021180606, down\_m\_s: -0.08999999776482582]  
2025-07-29 17:45:49 - Speed: VelocityNed: [north\_m\_s: -0.1099999948395355, east\_m\_s: -0.02999999329447746, down\_m\_s: 0.0199999952965164]  
2025-07-29 17:45:50 - Speed: VelocityNed: [north\_m\_s: -0.0399999910593833, east\_m\_s: 0.02999999329447746, down\_m\_s: 0.0199999952965164]  
2025-07-29 17:45:52 - Speed: VelocityNed: [north\_m\_s: 1.2799999713097785, east\_m\_s: 0.0700000029802322, down\_m\_s: 0.04999999781976776]  
2025-07-29 17:45:53 - Speed: VelocityNed: [north\_m\_s: 1.139999985048853, east\_m\_s: 0.0399999910593833, down\_m\_s: 0.0999999940395352]  
2025-07-29 17:45:54 - Speed: VelocityNed: [north\_m\_s: 0.3999999761581421, east\_m\_s: -0.02999999329447746, down\_m\_s: 0.0]  
2025-07-29 17:45:56 - Speed: VelocityNed: [north\_m\_s: 0.0700000029802322, east\_m\_s: -0.0599999986889549, down\_m\_s: 0.04999999781976776]  
2025-07-29 17:45:57 - Speed: VelocityNed: [north\_m\_s: 0.1999999887907104, east\_m\_s: -0.099999940395355, down\_m\_s: 0.0999999940395352]  
2025-07-29 17:45:58 - Speed: VelocityNed: [north\_m\_s: 0.0, east\_m\_s: -0.17999999225139618, down\_m\_s: 0.0599999966213989]  
2025-07-29 17:46:00 - Speed: VelocityNed: [north\_m\_s: 0.0, east\_m\_s: 0.0799999921180606, down\_m\_s: 0.069999988079071]  
2025-07-29 17:46:01 - Speed: VelocityNed: [north\_m\_s: -0.1499999910593833, east\_m\_s: -0.0199999952965164, down\_m\_s: 0.0699999978046550]  
2025-07-29 17:46:02 - Speed: VelocityNed: [north\_m\_s: 0.0199999952965164, east\_m\_s: -0.0700000029802322, down\_m\_s: 0.7099999785423279]  
2025-07-29 17:46:04 - Speed: VelocityNed: [north\_m\_s: -0.0199999952965164, east\_m\_s: -0.0199999952965164, down\_m\_s: 0.5999999642321311]  
2025-07-29 17:46:06 - Speed: VelocityNed: [north\_m\_s: -0.0399999910593833, east\_m\_s: 0.17999999225139618, down\_m\_s: 0.7799999713097785]  
2025-07-29 17:46:08 - Speed: VelocityNed: [north\_m\_s: 0.0999999940395355, east\_m\_s: 0.02999999329447746, down\_m\_s: 0.7599999984325680]  
2025-07-29 17:46:09 - Speed: VelocityNed: [north\_m\_s: -0.0700000029802322, east\_m\_s: 0.0399999910593833, down\_m\_s: 0.740000003674321]  
2025-07-29 17:46:10 - Speed: VelocityNed: [north\_m\_s: 0.0700000029802322, east\_m\_s: -0.1499999910593833, down\_m\_s: 0.0599999986889549]  
2025-07-29 17:46:10 - Speed: VelocityNed: [north\_m\_s: 0.0199999952965164, east\_m\_s: -0.0700000029802322, down\_m\_s: 0.0399999910593833]

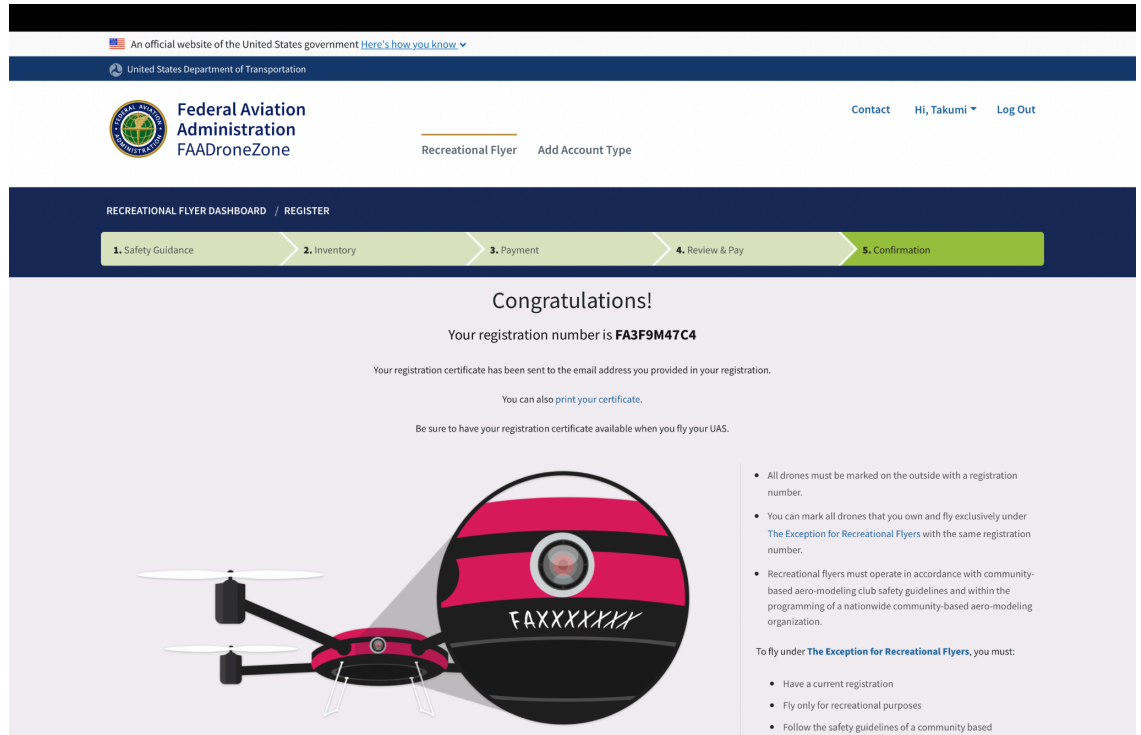
2025-07-29 17:44:58 - Yaw: 7.5358686447143555  
2025-07-29 17:44:59 - Yaw: 8.091832160949707  
2025-07-29 17:45:01 - Yaw: 9.322927474975586  
2025-07-29 17:45:02 - Yaw: 8.185508728027344  
2025-07-29 17:45:03 - Yaw: 8.368438720703125  
2025-07-29 17:45:05 - Yaw: 9.185943603515625  
2025-07-29 17:45:06 - Yaw: 9.690656661987305  
2025-07-29 17:45:07 - Yaw: 9.373856544494629  
2025-07-29 17:45:09 - Yaw: 8.79345989227295  
2025-07-29 17:45:10 - Yaw: 8.43398380279541  
2025-07-29 17:45:11 - Yaw: 8.221720695495605  
2025-07-29 17:45:13 - Yaw: 2.076481580734253  
2025-07-29 17:45:14 - Yaw: -0.696436882019043  
2025-07-29 17:45:15 - Yaw: -2.636740207672119  
2025-07-29 17:45:17 - Yaw: -1.4420303106307983  
2025-07-29 17:45:18 - Yaw: 0.5301834940910339  
2025-07-29 17:45:19 - Yaw: -0.5985799431800842  
2025-07-29 17:45:21 - Yaw: -0.5984582901000977  
2025-07-29 17:45:22 - Yaw: -0.3491703271865845  
2025-07-29 17:45:23 - Yaw: -2.0597546100616455  
2025-07-29 17:45:25 - Yaw: 0.9716392755508423  
2025-07-29 17:45:26 - Yaw: 0.9182286262512207  
2025-07-29 17:45:27 - Yaw: -0.1790928989648819  
2025-07-29 17:45:29 - Yaw: 0.7556390762329102  
2025-07-29 17:45:30 - Yaw: 0.6648014187812805  
2025-07-29 17:45:31 - Yaw: 0.5324532985687256  
2025-07-29 17:45:33 - Yaw: 54.10065460205078  
2025-07-29 17:45:34 - Yaw: 96.43439483642578  
2025-07-29 17:45:35 - Yaw: 90.70761108398438  
2025-07-29 17:45:37 - Yaw: 89.52947235107422  
2025-07-29 17:45:38 - Yaw: 90.24502563476562  
2025-07-29 17:45:39 - Yaw: 89.8501968383789  
2025-07-29 17:45:41 - Yaw: 90.00641632080078  
2025-07-29 17:45:42 - Yaw: 89.48662567138672  
2025-07-29 17:45:43 - Yaw: 89.55552673339844  
2025-07-29 17:45:45 - Yaw: 89.8853530883789  
2025-07-29 17:45:46 - Yaw: 89.73212432861328  
2025-07-29 17:45:47 - Yaw: 90.4009017944336  
2025-07-29 17:45:49 - Yaw: 90.32201385498047  
2025-07-29 17:45:50 - Yaw: 89.99810028076172  
2025-07-29 17:45:52 - Yaw: 89.24176788330078  
2025-07-29 17:45:53 - Yaw: 88.02479553222656  
2025-07-29 17:45:54 - Yaw: 91.05535125732422  
2025-07-29 17:45:56 - Yaw: 90.19969177246094  
2025-07-29 17:45:57 - Yaw: 90.53014373779297  
2025-07-29 17:45:58 - Yaw: 89.95084381103516  
2025-07-29 17:46:00 - Yaw: 90.37553405761719  
2025-07-29 17:46:01 - Yaw: 89.95513916015625  
2025-07-29 17:46:02 - Yaw: 90.25621032714844  
2025-07-29 17:46:04 - Yaw: 90.7911376953125  
2025-07-29 17:46:05 - Yaw: 88.67189025878906  
2025-07-29 17:46:06 - Yaw: 90.02135467529297  
2025-07-29 17:46:08 - Yaw: 91.8702621459961  
2025-07-29 17:46:09 - Yaw: 90.74768829345703  
2025-07-29 17:46:10 - Yaw: 91.15470886230469

- 考察

大会本番についても、今回と同様のログを生成することによって制御レポートを作成することができると考えられる。

## (FRV1) FAA機体申請書

- 目的  
FAA認証を取得する(FR1)
- 試験内容  
FAAに機体を登録する
- 結果  
以下のように登録ができた。





## 第5.2節 ミッションを達成するためのシステム試験

### (MV1)機体放出試験

- 目的  
CanSatが直径146mm高さ240mmの円筒キャリアから自重で放出できることを確認する。(M1)
- 試験内容  
ARLISSで用意される直径146mm高さ240mmの円筒キャリアより小さい直径145mm高さ239mmのキャリアを3Dプリンターで作成し、CanSatをキャリアから自重で放出する。

	写真	測定値
--	----	-----

直径		145mm
高さ		239mm

- 結果  
以下の表に示すように、CanSatを収納後、自重で放出することが確認できた。

日付	実験動画	放出
7/8	<a href="https://youtu.be/DOy1EjOveuY?si=ugqy1t5s9bmead6x">https://youtu.be/DOy1EjOveuY?si=ugqy1t5s9bmead6x</a>	自重で放出できた

- 追加試験
  - 目的  
キャリアには「パラシュート→機体」の順に収納するため、放出は「機体→パラシュート」の順で行われる必要があるにもかかわらず、従来の動画では「機体→パラシュート」の順で収納ののち、「機体→パラシュート」の順で放出されていた。これを修正し、本来のARLISSと同じ順序で収納・放出が正しく行えることを確認する。
  - 試験内容  
能代宇宙イベントにおける現地審査にて、キャリアをARLISS本番と同様に「パラ

シュート→機体」の順に収納し、落下させた。放出時には「機体→パラシュート」の順で正しく放出されることを確認する試験を実施した。

- 結果  
試験動画において、放出の瞬間に機体が一時的に画角から外れてしまった。しかし、直後の映像ではキャリアが空になっていることが確認でき、またCanSatは能代宇宙イベントでの現地審査を通過している。これらの事実から、収納・放出が正しい順序で行われたことは十分に確認できると考えられる。

8/14

[https://youtu.be/N304beQ3I4w?si=C\\_fbbPlcj-KZdKxJ](https://youtu.be/N304beQ3I4w?si=C_fbbPlcj-KZdKxJ)

- 考察  
能代宇宙イベントでの現地審査を追加試験として示すことで、収納順序「パラシュート→機体」、放出順序「機体→パラシュート」が正しく実現できていることを確認できた。放出の瞬間が動画に完全には収録されていないものの、現地審査で認められていること、ならびに空のキャリアが明確に映っていることから、要求を満たしていると判断できる。

## (MV2) 放出判定試験

- 目的  
CanSatがキャリアから放出したことを判定したことを確認する。(M1)
- 試験内容  
キャリアから機体を放出し、センサーの値から放出シーケンスが終了したことを機体が検知する。
- 試験内容  
RaspberryPiと基板のみを用いた試験を7/17に行った。内容としては、
  - 光センサーを手で覆うことで収納判定を行う
  - 基板を持った状態で頭の上から足元まで瞬時に下げて落下の状態を再現し、放出判定をする。
  - 基板を地面において着地判定をするであった。なお、上に示したように放出判定と着地判定の試験を同時に行った。また、7/18にEndtoEnd試験を行い、その一環で放出判定試験を行ったのでそちらもご参照いただきたい。CRV1EndtoEnd試験に記載している。
- 結果  
ログは以下の通りである。



#### 考察

自由落下させたとき機体は着地衝撃に耐え、その後のニクロム線溶断による起立展開、アーム展開も問題なく成功することを確認した。また自由落下による着地寸前の速度はパラシュートによる終端速度 $v_f = 5.15 \text{ m/s}$ より早いことが確認できているため本番での着地にも耐え、機能すると考えられる。また着地衝撃試験としての動画は分割したものしかなく、開発の時間的制約の中で一連の動画として提出することができなかった。しかし後述のEndtoEnd試験においてパラシュート落下から腕展開に至る様子が収められているためそちらをご参照いただきたい。

### (MV4) 着地判定試験

- 目的  
CanSatが地面に着地したことを判定したことを確認する。(M7)
- 試験内容  
機体を15mの高さからパラシュートで落下させ、着地した時に機体が着地を検知する。
- 試験内容  
EndtoEnd試験を行い、その一環で着地判定試験を行ったのでそちらをご参照いただきたい。CRV1EndtoEnd試験に記載している。

### (MV5) 起立展開試験

- 目的  
機体が着陸後に起立機構が正常に動作し、ニクロム線を用いたテグス溶断による腕展開可能状態になることを確かめる。(M6, M7)
- 試験内容  
機体を倒れた状態で配置し、起立機構を固定しているテグスをニクロム線により切断する。それにより、エンベロープを分離して離陸可能な体勢に移れるかどうかを確かめる。
- 試験結果  
以下の動画の通り、着地判定後、ニクロム線によるテグス溶断によりテグスを切断したのち機体が立ち上がり、離陸可能な姿勢で安定することが確認できた。  
動画は下記の(MV6)腕展開試験のURLと同じである。

### (MV6) 腕展開試験

- 目的  
起立展開後に、腕展開機構が正常に動作し、ニクロム線を用いたテグス溶断により離陸可能状態になることを確かめる。(M8)
- 試験内容  
起立機構が展開し、フライト機構が正常な水平姿勢となっている状態から試験を始める。腕展開機構を固定しているテグスをニクロム線加熱により溶断する。それにより、4本のアームを固定していたテグスが外れ、アームが90度展開し、離陸可能な体勢に移れるかどうかを確かめる。
- 結果

以下の動画の通り、起立後、ニクロム線によりテグスを切断をするとアームが展開し、離陸可能な姿勢で安定することが確認できた。

7/7(1回目)	<a href="https://youtu.be/5d4ywBaVj8w">https://youtu.be/5d4ywBaVj8w</a>
7/8(2回目)	<a href="https://youtu.be/c5FTrrCl1oU">https://youtu.be/c5FTrrCl1oU</a>
7/9(3回目)	<a href="https://www.youtube.com/watch?v=gj51a2vxhow">https://www.youtube.com/watch?v=gj51a2vxhow</a>

- 考察  
動画の通り、起立後、ニクロム線によってテグス切断をするとアームが展開し、離陸可能な姿勢で安定することが確認できた。

## (MV7) 機体離陸試験

- 目的  
展開機構から機体が適切に離陸できることを確認する。(M10)
- 試験内容  
まずは平坦な地面の上で起立展開機構のカーボンロッドに機体を刺し、腕などは全て展開した機体が飛行可能な状態でプロポからスロットルをあげて機体を離陸させる。それで再現性を持って離陸可能なことが確認されたら、次に様々な地面条件下(凹凸でがたつくなど)で離陸可能なことを確認する。
- 結果

7/7(1回目)	<a href="https://youtu.be/M3d28Ok4bwA?si=-sA1r9U4OoTfdMQX">https://youtu.be/M3d28Ok4bwA?si=-sA1r9U4OoTfdMQX</a>
7/7(2回目)	<a href="https://youtu.be/oLxbHAPeXcE?si=4SohNXL_q_Qs8B0q">https://youtu.be/oLxbHAPeXcE?si=4SohNXL_q_Qs8B0q</a>
7/7(3回目)	<a href="https://youtu.be/IYECdwTOBGM?si=dX49J5jOzHE95vRU">https://youtu.be/IYECdwTOBGM?si=dX49J5jOzHE95vRU</a>
7/7(4回目)	<a href="https://youtu.be/ORWXdUv7kKA?si=sOtA7GXkWI-DryFm">https://youtu.be/ORWXdUv7kKA?si=sOtA7GXkWI-DryFm</a>
7/7(5回目)	<a href="https://youtu.be/4IUsNoAIB1o?si=r1AuboG0vOKf5IE7">https://youtu.be/4IUsNoAIB1o?si=r1AuboG0vOKf5IE7</a>
7/7(6回目)	<a href="https://youtu.be/ZkVYPTdEd0I?si=w_vR5PvQ4YKoBhwd">https://youtu.be/ZkVYPTdEd0I?si=w_vR5PvQ4YKoBhwd</a>
7/7(7回目)	<a href="https://youtu.be/PVla8j7NGJ4?si=O1umW7HTT2MUyv1q">https://youtu.be/PVla8j7NGJ4?si=O1umW7HTT2MUyv1q</a>
7/7(8回目)	<a href="https://youtu.be/iIVG9T_Kg1I?si=JSemShkNVO10gHCn">https://youtu.be/iIVG9T_Kg1I?si=JSemShkNVO10gHCn</a>
7/7(9回目)	<a href="https://youtu.be/B8m3pJSMflg?si=mi5No0hkzw8Oq8Oi">https://youtu.be/B8m3pJSMflg?si=mi5No0hkzw8Oq8Oi</a>
7/7(10回目)	<a href="https://youtu.be/EvRpXCIDpY8?si=bLsfL3rnrDu6viB1">https://youtu.be/EvRpXCIDpY8?si=bLsfL3rnrDu6viB1</a>

- 考察  
平坦な場所では10回以上連続して試行して問題なく離陸できることが確認された。地面条件を変えての試験は行っていないが、条件が大きく変わるわけではないため問題なく動作すると考えている。  
また、自律しての離陸については、CR1のEnd-to-End試験の後半及び追加試験に内包されている。

## (MV8) 飛行性能確認試験

- 目的  
機体が安定してホバリングしそのまま着陸できることを確認する。(M11)
- 試験内容  
離陸後、離陸地点に着陸できることを確認する。
- 結果  
その場で離陸し、着陸できた。

- 考察  
機体が安定して飛ぶことを確認できた。

## (MV9) ログ試験

- 目的  
各シーケンスにおいてログが残せていることを確認する。(M1, M2, M3, M9, M13)
- 試験内容  
収納→放出→着地→起立→腕展開の試験の中で、各シーケンスのログが残せていることを確認する。
- 結果  
前半のログファイルは以下の通り。収納判定、放出判定、着地判定を行うことができ、また起立展開、腕展開についてもログを残すことができています。それぞれの判定に用いたセンサの値は別ファイルに保存し、判定に用いた結果をメインのログファイルに残している。  
‘...==  
2025-07-18 17:17:35 - message.storage\_check - INFO - light level: 1705.83  
2025-07-18 17:17:35 - message - INFO - Storage check passed, ready for takeoff ready check.  
2025-07-18 17:18:43 - message.TakeoffReadyCheck - INFO - light level: 686.67, speed: -1.65 m/s  
2025-07-18 17:18:43 - message - INFO - Release check passed.  
2025-07-18 17:18:54 - message.TakeoffReadyCheck - INFO - speed: -0.04 m/s  
2025-07-18 17:18:54 - message - INFO - Land check passed.  
2025-07-18 17:20:01 - message - INFO - 起立展開を開始します  
2025-07-18 17:20:01 - message.TakeoffReadyCheck - INFO - Turned ON MOSFET on pin 11  
2025-07-18 17:20:04 - message.TakeoffReadyCheck - INFO - Turned OFF MOSFET on pin 11  
2025-07-18 17:20:10 - message - INFO - 起立展開が完了しました  
2025-07-18 17:20:13 - message.TakeoffReadyCheck - INFO - Turned ON MOSFET on pin 13  
2025-07-18 17:20:16 - message.TakeoffReadyCheck - INFO - Turned OFF MOSFET on pin 13  
2025-07-18 17:20:24 - message - INFO - 腕展開が完了しました。
- 考察  
ログを残すことができた。

## (MV10)GPS航法飛行試験

- 目的  
CanSatがGPS情報を使用して飛行経路を正確にたどることを確認する。(M12, M14)
- 試験内容  
特定の緯度・経度を指定し、そこまで自律飛行させる。  
GPSモジュールを使用し、指定された飛行経路を飛行するテストを行う。

- 結果  
ゴール座標周辺に着陸した。

[https://www.youtube.com/watch?v=QWjC-tZk284&ab\\_channel=%E6%AD%A3%E9%B5%A0%2FSEIKOKU](https://www.youtube.com/watch?v=QWjC-tZk284&ab_channel=%E6%AD%A3%E9%B5%A0%2FSEIKOKU)

raspiが指示した記録について、ゴール地点での推定GPS座標は以下である

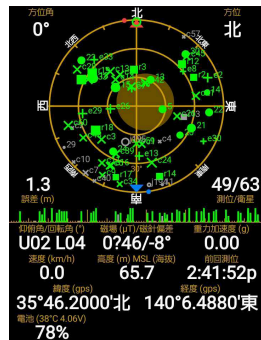
目標地点:緯度35.770019, 経度140.10817

推定位置(EKF2): 緯度35.770021899999996, 経度140.1081659

ログ全文は以下である。

<https://drive.google.com/file/d/1fzF2XTbbh9PIM35n7MONTq0eKPcIzdUv/view?usp=sharing>

動画内でのゴールスマートフォンを用いて測定した緯度経度が以下である。



十進法に変換すると、35.770000,140.108133°である。

目標地点との距離は約4mである。

- 考察  
4mのずれはGPSの誤差範囲内といえるので、ゴール座標に向かわせることに成功したといえる。

## (MV11)機体着陸試験

- 目的  
機体が着陸できることを確認する。(M17)
- 試験内容  
ホバリング後、ホバリング地点の直下に着陸できることを確認する。
- 結果  
ホバリングし、直下に着陸できた  
飛行試験動画

7/4	<a href="https://youtu.be/zBiG7UjvVGI">https://youtu.be/zBiG7UjvVGI</a>
-----	---

- 結論  
機体が安全に着陸することを確認できた。

## (MV12)ゴール検知試験

- 目的

CanSatが地上に設置された目標物を正確に検知することを確認する。(M15)

- 試験内容  
目標物を地上に配置し、CanSatに載せているRaspberry Pi Zero2W のカメラが飛行中にこれを検知する機能をテストする。
- 結果  
読み込んだ画像からGoal Markerを検知することに成功した。  
Rasberry Pi CameraでGoal Markerを撮影、検知し、ゴール座標を検出した。実際に撮影された画像は以下である



目標地点の画像内ピクセル座標は画像内でも出力されている。画像全体のピクセルが640\*445であるが、物体の座標は、320\*217と出力した。

- 考察  
正しくGoal Markerを認識して画像内のピクセル座標を出力できた。

## (MV13) 画像航法試験

- 目的  
CanSatが飛行中に撮影した画像を使用して、地上の目標物を正確に認識することを確認する。また、正常に接近できることも確認する。(M15, M16)
- 試験内容  
CanSatに搭載されたRaspberry Pi に対応するカメラでGoal Markerを捕捉し、画像処理を行って目標物を認識、付近に着陸するテストを行う。
- 試験計画  
以上の試験は7月中に行う予定である。  
結果は取得したログによって確認、提示する。
- 結果  
<https://youtu.be/ZxeBx0Idhfi>  
V25 EndtoEnd試験の一部として行った。Rasberry Pi CameraでGoal Markerである赤コーンを撮影、検知し、ゴール座標を検出、赤コーン上空まで移動して、ゴール座標付近に着陸することができた。
- 考察  
正しくGoal Markerを認識してゴール座標まで誘導することができた。
- 追加試験  
その後、ARLISS新ゴールを使用することに決定したため、ARLISS新ゴールを用いた画像航法試験を行なった。
  - 目的  
CanSatが画像航法により新ゴールを正しく認識し、着陸シーケンスを完了できるかを確認する。

- 試験内容  
離陸後にGPS航法でゴール近傍まで接近させ、その後画像航法を行った。
- 結果  
CanSatは新ゴールを正しく検出し、画像航法に基づき移動・着陸に成功した。動画は以下。

8/29	<a href="https://youtu.be/F9VDToiEC10">https://youtu.be/F9VDToiEC10</a>
------	---

- 考察  
本試験により、新ゴールにおいても画像航法が有効に機能することが確認された。

### (MV14) 航続距離検証試験

- 目的  
CanSatを用いて数km級のGPS航法試験を行い、バッテリー残量から航続可能距離を見積もる。
- 試験内容  
飛行場にて、CanSatに、あらかじめ設定した航法経路(往復300kmの直線経路)に従ったGPS自律飛行を行わせた。  
なお、安全のため、異常時は即時マニュアル介入可能とした。  
プログラム上は10kmの往復飛行を指示したが、事前にeCalcで算出した航続可能距離は約5.7kmであった。そのため、安全を考慮して6kmを越えた時点で手動着陸させる方針とした。
- 結果  
ログ解析の結果、飛行距離は6.6kmであった。この時点で駆動バッテリー残量は19%、ロジック用LiPoの残量は79%であり、いずれも十分な余裕を残していた。飛行中には大きな逸脱や通信断は発生せず、シーケンス全体を通じて安定した飛行が確認された。

8/30(航続距離検証試験)	<a href="https://youtu.be/5BvK_w3X3To">https://youtu.be/5BvK_w3X3To</a>
----------------	---

- 考察  
残量から逆算すると、推定最大航続距離は約8.1kmに達することが示された。  
ARLISS本番環境での運用に十分な余裕を持つと考えられる。

## 第6章 工程管理

本プロジェクトでは、「**ARLISS 1日目朝の打上げ**」を最重要マイルストーンとしてスケジュールを構築した。

これは、風が穏やかでかつ打上げ枠の空きが見込まれる大会初日午前を狙い、確実にミッションを遂行できる体制を整えることを目標としたためである。

特に、0mゴールを実現するためには早朝の穏やかな風条件が不可欠であり、この「1日目朝打上げ」を開発全体のゴールの1つとして位置づけた。

さらに、本プロジェクトでは、能代宇宙イベントでの**End-to-End**の成功及び課題の洗い出しを重要マイルストーンとした。

能代宇宙イベントは、大学での試験では実施できない30m以上の高さからの投下とフライバックの連続したEnd-to-End試験ができる大変貴重な機会であった。

プロジェクト全体の中での、技術検証・運用練習の場としての重要な中間マイルストーンが能代大会であった。

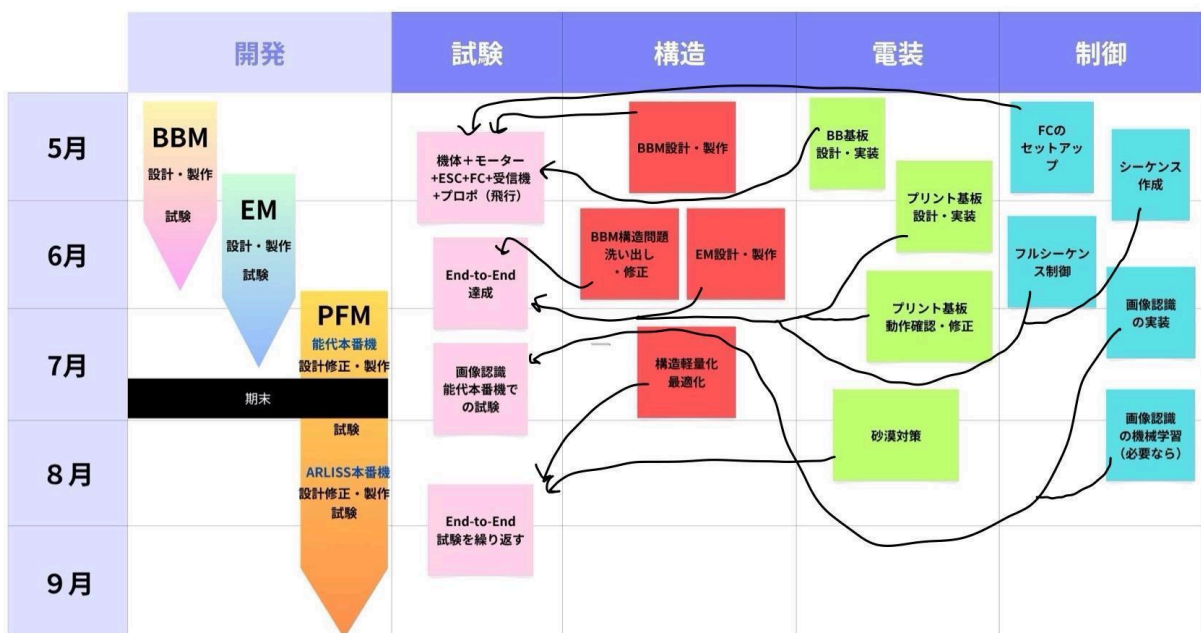
具体的な開発スケジュールとしては、再現性および信頼性の高い機体開発を目指して、目的と段階に応じて3機体(BBM、EM、PFM)を開発した。

各機体は以下のような目的を持ち、段階的な試験と改善を通じて全体性能の向上を図った。

- **BBM (Bread Board Model)**  
: 飛行機能のみ、開発ノウハウを学び、プロジェクトを軌道に載せる
- **EM (Engineering Model)**  
: End-to-End達成
- **PFM (Proto-Flight Model)**  
: End-to-End試験を繰り返す


さらに、PDR、CDR、TDRの3回の学内審査会を設け、先輩方や教員からのレビューをもらう場とした。

大まかな開発スケジュールは下図の通り。



このように、試験を通じた実証的開発手法を段階的に適用することで、信頼性の高い成果を目指した。  
スケジュール管理はScrapbox上に可視化されたガントチャートをベースとし、各段階試験をマイルストーンとして設定することで、目的指向的かつ検証可能な進捗管理を実施している。

以下が詳細ガントチャートとなる。

 正鶴開発スケジュール\_ARLISS提出用 - プロジェクトのタイムライン.pdf

## 第7章 大会結果報告

### 第7.1節 目的

製作したCanSatが設計通りに動作することを確認する。

### 第7.2節 結果

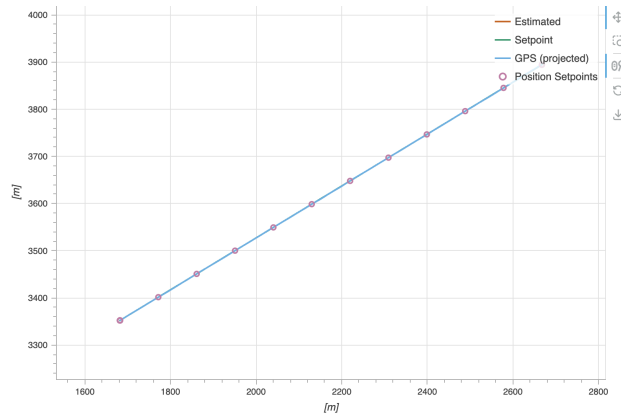
#### 第7.2.1項 投下1回目

本チームは初日朝の1チーム目として打上に成功した。  
パラシュートの開傘を目視で確認し、着地後には起立動作の成功を先生方が確認した。続いて離陸にも成功し、機体がゴール方向へ向けて飛行を開始したことも先生方により目視で確認された。  
その後、約1.25km飛行した地点(ゴールから約3.25 km地点)で、機体は約15分間(推定)ホバリングを継続した後、墜落した。  
墜落後もRaspberry Piのインジケータは点灯していたが、LoRaによる通信が途絶しており、何らかの要因によりシーケンスが停止したと判断し、リタイアとした。

9/8(離陸、ゴールへの飛行の様子)	<a href="https://youtu.be/A_WbSolHQpc">https://youtu.be/A_WbSolHQpc</a>
--------------------	---



離陸の様子



飛行の軌跡

サクセスクライテリアに照らし合わせると、以下の通り。

	内容	評価方法	結果
Minimum success	m1.放出されたことをログに残す	ログで確認	○
	m2.機体の座標・高度をログに残す	ログで確認	○
	m3.9軸データ(加速度, 地磁気, 角速度)のログを取る	ログで確認	○
	m4.パラシュートが展開する	目視で確認	○
	m5.着地判定をログに残す	ログで確認	○
	m6.着地後に機体がパラシュートを切り離せる	目視で確認	○
	m7.展開機構の展開により機体が離陸姿勢を取れる	目視で確認	○
	m8.機体の腕が展開できる	目視で確認	○
	m9.機体と展開機構を切り離す	目視で確認	○
	m10.放出・着地・展開シーケンスの完了及び機体状態(座標・高度)を地上局に送信する	目視で確認	○
	m11.機体が離陸する	目視で確認	○
Full success	f1.機体が一定の高度を保つことができる	目視で確認	○
	f2.機体が目標物に向かって飛行する	目視で確認	○
	f3.飛行中の座標・高度を地上局に送信する	目視で確認	○
	f4.目標物から5m以内に着陸する	目視で確認	×
Extra success	e1 目標物を認識してゴール座標を割り出す	ログで確認	×

ミニマムサクセスを達成したが、フルサクセス達成手前でリタイアとなった。

### 第7.2.3項 投下2回目

9時53分にコードを実行し、9時55分に収納判定を確認。10時20分20秒に打上げが行われた。

打上げ直後、車内ではLoRa通信の受信ができなかった。

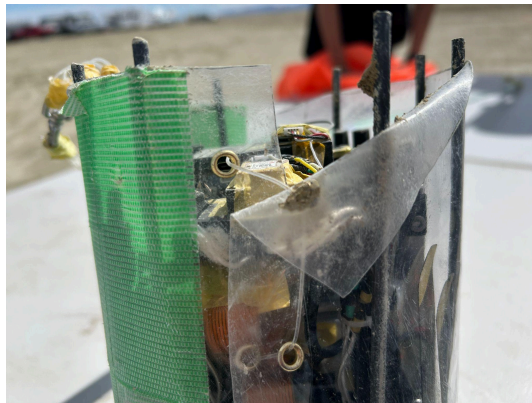
10時20分54秒の記録を最後に通信が途絶しており、この時点で機体は速度14.85 m/sで下降中、光センサ値は-1705.8であったことから、依然としてロケット内にある状態であると推定された。

その後、機体を車で追跡し、10時31分に発見。機体はパラシュートに引きずられる形で地表を移動しており、エンベロープが砂を掬い上げる様子や、機体が転がり・ひっくり返る様子が確認された。

10時32分には、Raspberry Piが再起動しWi-Fiに接続した際のSlack通知を確認。再起動によりメインプログラムが停止し、シーケンスが継続しない状態となっていたと判断し、10時55分にリタイアした。

再起動後、機体には手を触れずにSSH接続が可能であった。

また、回収時の確認により、駆動リポの固定が外れていたことも確認された。



リタイア後、ベースキャンプで撮影。パラシュートで引きずられ砂まみれになっていた。  
リタイア後起立展開試験を実施したところ、成功した。

9/11(起立展開試験)	<a href="https://youtu.be/yZ3BbwTgb_I">https://youtu.be/yZ3BbwTgb_I</a>
--------------	---

サクセスクライテリアに照らし合わせると、以下の通り。

	内容	評価方法	結果
Minimum success	m1.放出されたことをログに残す	ログで確認	○
	m2.機体の座標・高度をログに残す	ログで確認	○
	m3.9軸データ(加速度, 地磁気, 角速度)のログを取る	ログで確認	○
	m4.パラシュートが展開する	目視で確認	○
	m5.着地判定をログに残す	ログで確認	×
	m6.着地後に機体がパラシュートを切り離せる	目視で確認	×
	m7.展開機構の展開により機体が離陸姿勢を取れる	目視で確認	×
	m8.機体の腕が展開できる	目視で確認	×
	m9.機体と展開機構を切り離す	目視で確認	×
	m10.放出・着地・展開シーケンスの完了及び機体状態(座標・高度)を地上局に送信する	目視で確認	×
	m11.機体が離陸する	目視で確認	×
Full success	f1.機体が一定の高度を保つことができる	目視で確認	×
	f2.機体が目標物に向かって飛行する	目視で確認	×
	f3.飛行中の座標・高度を地上局に送信する	目視で確認	×
	f4.目標物から5m以内に着陸する	目視で確認	×
Extra success	e1 目標物を認識してゴール座標を割り出す	ログで確認	×

ミニマムサクセス未達成となった。

### 第7.2.3項 投下3回目

パラシュート開傘を目視で確認し、着地判定まではLoRa通信によって機体座標、シーケンス進行状況などを順調に確認できた。

着地後、起立動作の成功を目視で確認し、続いて離陸も目視で確認した。

しかし、離陸直後に機体が墜落(ゴールから約3.94 km地点)した。

GNSSモジュールのケーブルがアーム部の爪に巻き込まれ、コネクタが抜けていたことを目視で確認した。

また、駆動用リポバッテリーの固定が外れており、モーターケーブルの被膜も剥離していることが判明した。これらの損傷は、腕展開時に部品が干渉した可能性があると考えられる。

ミッション継続は不可能と判断し、リタイアとした。

リタイア後の確認において、ESCのショートも確認された。



墜落直後のGPSコネクタの様子



墜落直後のモーターケーブルの様子

9/11(投下3回目)	<a href="https://youtube.com/watch?v=JvIN5ZIoXil?feature=share">https://youtube.com/watch?v=JvIN5ZIoXil?feature=share</a>
-------------	---

サクセスクライテリアに照らし合わせると以下の通り。

	内容	評価方法	結果
Minimum success	m1.放出されたことをログに残す	ログで確認	○
	m2.機体の座標・高度をログに残す	ログで確認	○
	m3.9軸データ(加速度, 地磁気, 角速度)のログを取る	ログで確認	○
	m4.パラシュートが展開する	目視で確認	○
	m5.着地判定をログに残す	ログで確認	○
	m6.着地後に機体がパラシュートを切り離せる	目視で確認	○
	m7.展開機構の展開により機体が離陸姿勢を取れる	目視で確認	○
	m8.機体の腕が展開できる	目視で確認	○
	m9.機体と展開機構を切り離す	目視で確認	○
	m10.放出・着地・展開シーケンスの完了及び機体状態(座標・高度)を地上局に送信する	目視で確認	○
	m11.機体が離陸する	目視で確認	○
Full success	f1.機体が一定の高度を保つことができる	目視で確認	×
	f2.機体が目標物に向かって飛行する	目視で確認	×
	f3.飛行中の座標・高度を地上局に送信する	目視で確認	×
	f4.目標物から5m以内に着陸する	目視で確認	×
Extra success	e1 目標物を認識してゴール座標を割り出す	ログで確認	×

ミニマムサクセス未達成となった。

#### 第7.2.4項 閉会式後飛行試験

閉会式後、ARLISS運営チームの協力のもと、機体の再飛行試験を実施した。試験の目的は、「3回目の打上げで発生した不具合(GPSコネクタの抜け、モーターケーブルの固定不良)」がなかった場合、ミッションがどの程度達成できたかを検証することであった。

スタート地点は3回目の墜落地点、ゴールはそこから南に3.94km先の新設ゴール地点とした。ARLISS運営チームより新ゴールの予備をお借りし、同条件下で再試験を実施した。

その結果、再離陸シーケンスからGPS航法、さらに画像航法、着陸までを目視で全て確認することができた。シーケンス全体を通して、Loraでの通信も良好だった。機体は最終的にゴール中心から87cmの着地を達成した。さらに、冗長系の発動も確認できた。

この結果から、もし本番でGPSコネクタの抜けやモーターケーブルの固定不良がなければ、ミッションのフルサクセス達成が十分に可能であったと推察される。機体性能・制御アルゴリズム・航法シーケンスの有効性が実証され、本プロジェクトの技術的到達点を示す重要な試験となった。

9/12(実験)

<https://youtu.be/AuatWBWwLdo>



画像航法の様子



87cmゴール

## 第7.3節 考察

### 第7.3.1項 取得データ

#### 第7.3.1.1目 投下1回目

- **PX4**フライトログ

[https://review.px4.io/plot\\_app?log=0f1cbfec-6614-44c9-9078-e91b55c66dc9](https://review.px4.io/plot_app?log=0f1cbfec-6614-44c9-9078-e91b55c66dc9)

- **RaspberryPiログ**  
📄 2025\_09\_09\_0219

#### 第7.3.1.2目 投下2回目

- **RaspberryPiログ**  
📄 2025\_09\_12\_0153

#### 第7.3.1.3目 投下3回目

- **PX4フライトログ**  
[https://review.px4.io/plot\\_app?log=e50cbc2f-4dd3-4ada-8b0f-2f23378ed59f](https://review.px4.io/plot_app?log=e50cbc2f-4dd3-4ada-8b0f-2f23378ed59f)
- **RaspberryPiログ**  
📄 2025\_09\_12\_0816

#### 第7.3.1.4目 閉会式後飛行試験

- **PX4フライトログ**  
[https://review.px4.io/plot\\_app?log=ccab611b-8a17-47b3-b7e8-ad9fba18df1e](https://review.px4.io/plot_app?log=ccab611b-8a17-47b3-b7e8-ad9fba18df1e)
- **RaspberryPiログ**  
📄 2025\_09\_13\_0623

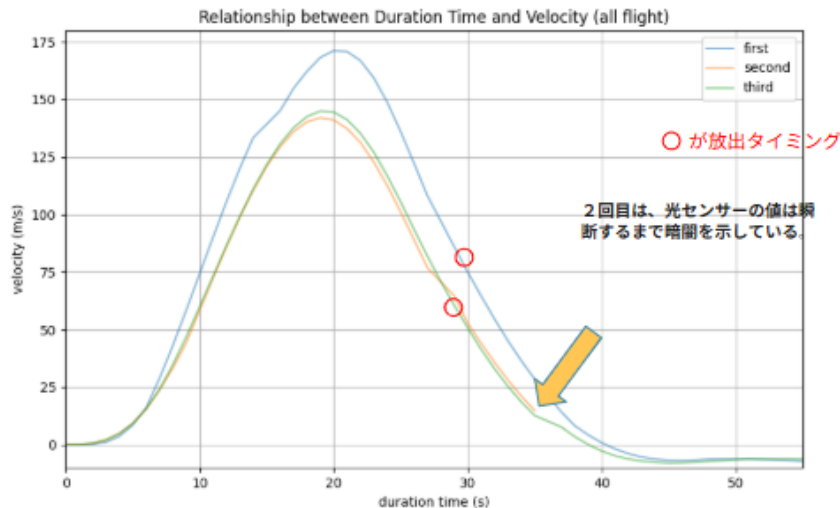
### 第7.3.2項 故障個所・原因解析

#### 第7.3.2.1目 投下1回目

- 空中での停止  
 非同期処理におけるタスクの消去ができておらず、`raspberry pi` が処理落ちしてしまい、制御信号が送られなくなり機体がフリーズした。  
 フライトコントローラからデータを取得するために非同期処理を用いていたが、スレッドの終了手続きに不備があり、未終了スレッドが大量に生成される状態となっていた。これにより、一定時間稼働後に処理が輻輳し、通信途絶・制御停止が発生したと考えられる。  
 具体的には、高周期(1秒ごと)で`async for`を呼び出し続けていた。
- 再現実験  
 翌日(9/10)、駆動系の電源を抜いた状態で本番コードを回す再現実験を行い、約15分が経過したところで、本番と同様の状態を再現することができた。
- 対策  
 最初にタスクを生成し、`async for`でEKFの位置座標やlidarの値を更新し続け、それを関数で引き出すような処理にした。問題を解消したコードでの6kmの飛行も投下2回目まで(9/10)に確認した。
- 本番まで発見できなかった原因  
 もともと、問題となる`async for`のコードは、開発初期に作られたものであり、ずっとそのコードで長距離試験を行っていた。そのため、オーバーフローの可能性を排除していた。しかし、開発が進むにつれて処理内容が増加し、本番直前に追加されたモジュールによって、ちょうどミッション中にスレッド飽和が発生する臨界点に到達したと考えられる。  
 時系列で見ると、8/30に宇都宮で6kmの長距離飛行試験を実施したが、問題の原因となったモジュールはその翌日(8/31)に書き換えられていた。  
 9/1には収納・放出・300m飛行・画像認識の試験で0mゴールを達成していたが、この時点で6km規模の長時間飛行を行っていたら、スレッド飽和を事前に検知できた可能性がある。

### 第7.3.2.2目 投下2回目

- 電源系統の瞬断  
打ち上げ後、機体を発見したとき、動いておらず、起立展開ができていなかった。機体に近づいてみると、raspberry pi がwifi接続した際の接続通知がslackに来たことを確認。  
この接続通知は、raspberry piが起動した後1回しか来ない通知である。そのため、打ち上げ～機体発見までの間にraspberry piが再起動したと判断し、リタイアした。
- 瞬断タイミングについて  
瞬断が起こったタイミングは、ロケットからの放出時ではないかと考えている。



ロケット打ち上げ後の速度と時間の関係

上図は、ロケット打ち上げ後の、速度と時間の関係のグラフである。縦軸が気圧センサーで推算した速度[m/s]、横軸がロケット打ち上げからの時間[s]となっている。黄色のグラフが瞬断が生じたグラフである。

1、3回目の打ち上げの際は、打ち上げから約30秒で放出されている。(放出タイミングは、機体の光センサーの値の変化から算出。)2回目では、センサーの値が途切れたタイミングが、約35秒とほかの2回よりも遅く、光センサーの値も収納時から変化していないため、放出タイミングで瞬断が発生したと考察している。

- 瞬断原因箇所  
状況を再現できていないため、故障場所は絞れていないが、ロジックシステムのリポバッテリーにはスイッチがついており、そのスイッチの接触の関係で瞬断が起こったと考えている。
- 瞬断対策について  
振動試験や解散衝撃試験の際には瞬断が起これなかったこと、電装設計がシンプルで信頼度が高いと考えていたことから、瞬断確率が十分に低いと考えた。  
RaspberryPiiのカーネルの設定を行うことで、対策を行うことを検討していたが、カーネルの理解にかかる学習コストや、設定ミスによる不具合、実験性の悪化、予期せぬ不具合の可能性などを鑑みて対策の導入を見送っていた。
- デーモン機能を搭載していなかったことについて  
本番では、デーモン機能を搭載していなかった。8月時点では、再起動後にシーケンスを途中から再開できるシステムの開発を進めており、8月中旬の実験でデーモンを搭載した機体をテストしたが、制御系が暴走し、原因解明に時間を要した。試験の難しさ、開発優先度の観点から、本番機ではデーモンの搭載を見送った。

再起動だけを独立して実装するという選択肢が意識から抜け落ちていたと振り返る。ARLISS本番では、打ち上げ2回目のリタイアの後、その場でデーモンを実装し、その日の内に打ち上げ3回目を実施した。

8月下旬には画像航法の冗長系を実装したが、結果として「再起動の実装を優先すべきであった」と強く反省している。シーケンス前半に関わる実装、例えば再起動の優先順位を過小評価していた。

### 第7.3.2.3目 投下3回目

- 墜落原因:GNSSモジュールのコネクタが抜け、Failsafeが発動  
GNSSモジュールのケーブルが抜けたことで、フライトコントローラーのFailsafeが発動し、着陸した。

7	1:52:18	WARNING	[mc_pos_control] Failsafe: blind land
8	1:52:18	WARNING	Failsafe activated: Autopilot disengaged, switching to Descend
9	1:52:19	CRITICAL	Accel 1 clipping, not safe to fly!
10	1:52:20	WARNING	[mc_pos_control] invalid setpoints
11	1:52:20	WARNING	[mc_pos_control] Failsafe: blind land
12	1:52:20	WARNING	[mc_pos_control] Failsafe: blind descent
13	1:52:20	INFO	GNSS data fusion stopped
14	1:52:20	WARNING	Failsafe activated: Autopilot disengaged, switching to terminate
15	1:52:22	WARNING	[health_and_arming_checks] Preflight Fail: Flight termination active
16	1:52:25	INFO	Disarmed by lockdown

フライトログより

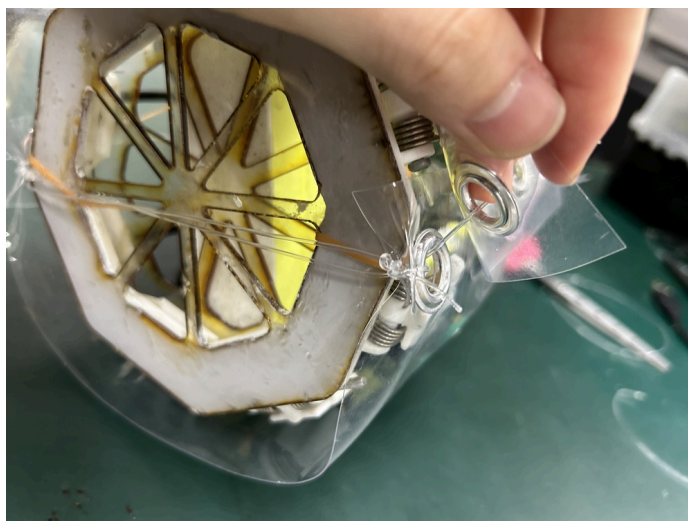
- 技術的要因:整備不良
  - GNSSモジュールケーブルの整備不良  
GNSSモジュールケーブルの一部が腕展開時にアームのツメに巻き込まれ、離陸時にモーターの推力によって強く引っ張られた結果、GNSSモジュールのコネクタが外れてしまった。  
これは、打ち上げ直前にGNSSモジュールの不調をケーブルを疑い取り替えたことに原因がある。ケーブルによる不調だったかは不確かだったが、改善したためそのまま打ち上げてしまった。  
しかし、取り替えたケーブルが8月前半に使用していたもので、8月後半から9月にかけて使用していたものより絶妙に長く、アーム根本と干渉してしまった。
  - モーターケーブルの整備不良  
打ち上げ2回目の際、機体がエンベロープに包まれた状態でパラシュートに長時間引きずられ、砂まみれになった。モーターの信頼性が疑わしかったため、予備機体を使用して打ち上げ3回目に望んだ。  
その結果、ESCのケーブル長が本番機体より微妙に長かったために固定方法に若干の誤差が生まれ、腕の開閉時にずれ、2つのアームでプロペラに干渉したと推測している。
- 背景要因の分析  
整備不良の背景には、次の2点が挙げられる。
  - ① 予備機体の採用判断の曖昧さ  
打ち上げ3回目では、電装構成などの仕様が一部異なる機体を「予備機」として使用した。しかし、翌日の飛行試験では本番機を再び使用して3.94kmの飛行に成功しており、機体差分を把握しきれないまま予備機体での打上げに踏み切った判断は適切ではなかったといえる。  
「予備」と呼べるだけの同等性を確保できていなかった点が、整備面での混乱を招き、結果としてミッション中断に繋がったと振り返る。

② 疲労による点検精度の不足と変更リスクへの意識低下  
可動部を含む全ての機構は、打上げ直前に必ず点検を行うべきであるが、限られた時間の中で変更作業を行った結果、リスク評価が不十分となってしまった。また、時間制約と疲労の蓄積(連日の4:30起床・再チャレンジによる打上げ2回目からの連続作業)により、判断力・集中力が低下していたことも要因の一つであると考えられる。(打上げ2回目のリタイア後、11:30頃から打上げ2回目の失敗原因分析と予備機体の整備、デーモンの実装を始め、16:00過ぎに現地審査を通過した。)

## 第8章 まとめ

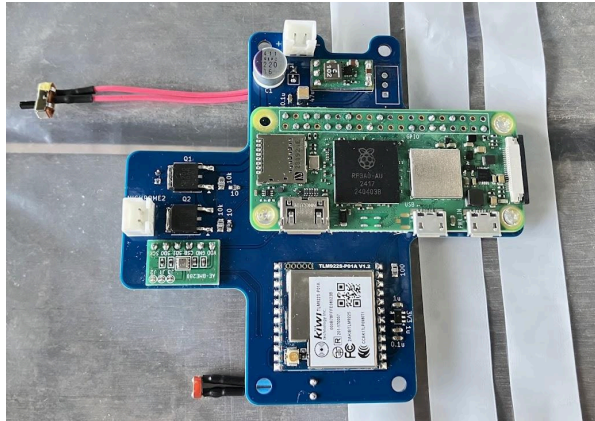
### 第8.1節 工夫・努力した点

- エンベロープの方向依存性の解消  
従来の弊団体のエンベロープには、方向依存性という課題があった。  
具体的には、展開するはずの辺が機体の下敷きになると摩擦によってエンベロープが開かなくなるというものである。  
そこで、第4.2.1項に記載した通り、エンベロープを2枚に分割し機体上部でテグスを溶断することで、180度どの辺を下敷きにしても展開が可能になる設計とした。



画像手前と画像奥に展開する辺がある

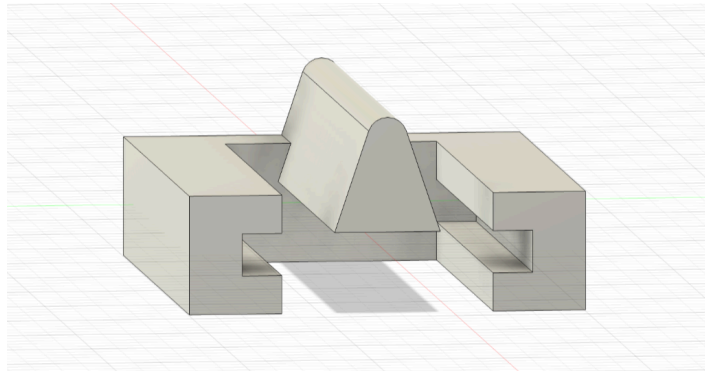
- 挙動に幅を残さない構造設計  
開発初期では、パラシュートやエンベロープ、テグスなどの柔らかいコンポーネントに多少の「あそび(可動の余裕)」を持たせることで整備性を優先していた。  
しかし、開発が進むにつれて、この「あそび」が再現性を大きく損なう原因になることがわかってきた。  
特に手を焼いたのは、テグスの挙動である。  
柔らかい部品は、収納状態等によってわずかに形状や張力が変化し、展開のタイミングや方向が毎回微妙にずれてしまう。  
そこで、設計方針を改め、なるべく「硬い」構造を採用し、柔らかい要素にも可動の余裕を残さないようにする工夫を進めた。  
具体的には、テグスを添わせるガイドを設けたり、パラシュートのシュラウドラインのテグスをメジャーに巻きつける際あまりがでない巻き付け方を手順化するなどである。さらに、テグスの溶断機構を非接触式に変更し、テグスの動きの再現性を極限まで高めた。また、同時に、テグスが引っかかる可能性のある部分を徹底的に排除した。
- 表面実装を多用した基板  
コンパクトで信頼性の高い基板を実現できた。



本番基板

- 治具を用いた再現性の確保  
製作した治具は以下の3つ。

1. ニクロム線によるテグス溶断基板製作用の治具  
溶断基板そのものの再現性が向上し、テグスとニクロム線の位置関係の再現性が担保され、テグス溶断の再現性も向上した。  
さらに、溶断基板製作の属人化が解消された。



溶断機構用治具のCADデータ

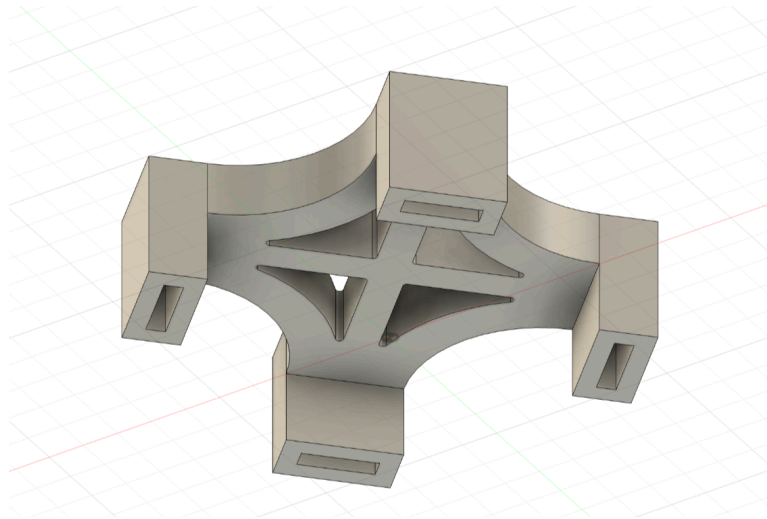


治具を用いて製作した溶断基板の再現性の高さ

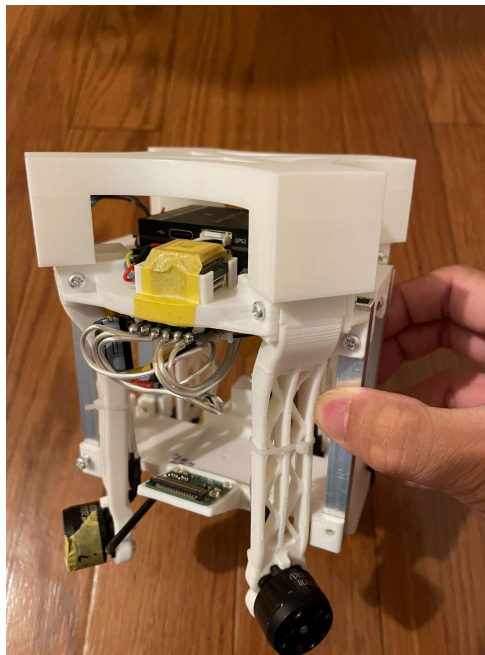
2. アームを折りたたんだ状態で保持する治具  
アームの固定作業の際、アームを手で折りたたみテグスで仮止めすると、アームの締め付け具合に差が生じ、再現性が確保できないことが課題だった。

そこで治具を製作した結果、アームの締め付け具合の再現性が担保されるようになった上に、作業負担が軽減されて作業効率が向上し、繰り返し試験の増加や本番での準備時間短縮につながった。

具体的には、2人必要だったアームの固定作業が1人でできるようになり、必要な時間が10分から3分程度に短縮された。(ただし、治具の固定自体は2人必要)



アーム固定用治具のCADデータ

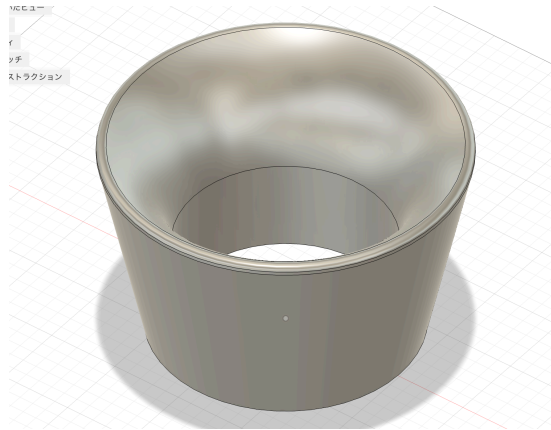


アームを固定した様子

### 3. 起立機構を折りたたむ際補助する治具

また、起立機構を折りたたむ作業も、3人がかりで行う必要がある上、締め付け具合によって再現性を確保できないという課題を抱えていた。

そこで、起立機構を折りたたむ際補助する治具を製作したところ、起立機構のたたみ具合の再現性を担保できるようになった上、1人で作業できるようになった(ただし、治具の取り付けには2~3人必要)。



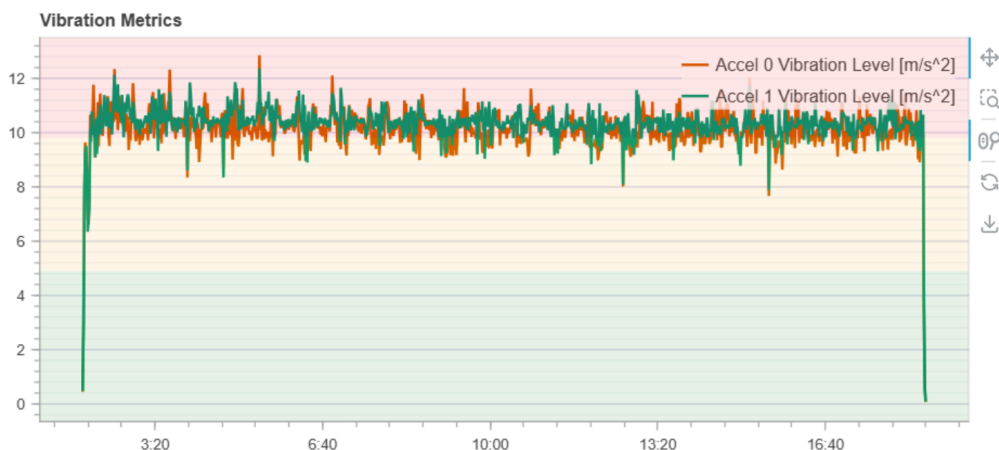
起立機構を折りたたむための治具。起立機構にセットした機体を上から押し込むことで使用する。

- シミュレーションの活用  
第8.2節成果 第8.2.3項制御班に詳細を記した。
- ソフト面での冗長設計  
各シーケンスにおいて、複数センサなどを用いた冗長系を組んだ。特に、20秒おきにセットポイントを送り直すことで飛行中にGPS情報が瞬間的に失われても飛行を続けるという冗長はARLISSでの飛行試験にて作動を確認できた(第8.2節成果 第8.2.3項制御班に詳細を記した)。
- 試験のドキュメンテーション  
第4.5.2.2目に詳細を記した。
- 運用も含めた再現性の追求  
技術面での再現に留まらず、大会当日の運用フローを含めた再現性を重視した。  
具体的には、ドキュメンテーションの徹底、治具の設計及び活用、手順書の作成などである。なお、治具や手順書は能代大会での反省を受けてARLISSで実現した。これにより、当初から目標として掲げていたARLISS初日朝の打ち上げを実現できた。  
手順書のリンク: [📄 打ち上げ前手順書\\_報告書提出用](#)  
(実際には、手順書を印刷し、紙媒体でのチェックを行った。紙で印刷することで、出国前に手順書を完成させざるを得なくなり、事前に運用面の準備ができるという利点があった。)
- **EM**ペラ振動問題の解消  
EM開発中、機体が離陸しなくなるトラブルが発生した。本チームでは、「EMペラ振動問題」と呼び、その後の開発期間を通して「開発が沼る」ことの例えとして使用された。  
この問題では、当初制御班による飛行試験でソフトウェア(特にEKF)異常を疑いデバッグに約2週間を費やした。  
しかし実際にはBBMからEMへのプロペラ変更により発生した機械的振動が原因であった。この経験から、ハード・ソフトの双方の観点から問題を切り分けることと、試験には各班から人員を配置することの重要性を痛感した。
- 構造班のタスク分配及び統合段階での構造的干渉の解消  
EM開発中、構造班内の作業分担が過度に進んだ結果、統合時に設計・実装上の干渉が多発し、スケジュールに遅れが生じた。これは、概念設計の不足及び情報共有の不足も原因だった。

## 第8.2節 成果

### 第8.2.1項 構造班

- 飛行時の振動の小ささ、飛行安定性  
本チームは緩衝材などを用いずに、単純な機体の剛性のみで下図のような振動の小ささを実現できた。



振動特性(図は8/30 宇都宮飛行試験場6.2km飛行時の振動状態)

- ミニマムサクセスまでのシーケンス(起立・離陸)の再現性の高さ  
ARLISS本番では2回の成功を達成した。繰り返しの試験により手順を確立できたと考える。
- 起立機構の非常に高い強度  
試験中パラシュートの不調により終端速度を上回る落下を複数回経験しても、起立機構は破損しなかった。これが前述の「ミニマムサクセスまでのシーケンス(起立・離陸)の再現性の高さ」に大きく寄与したと考える。
- 治具の整備  
打上前準備用の治具の整備により、作業の再現性・効率が大変向上した。能代大会では1時間以上かかっていた打ち上げ前準備が、ARLISSでは、より丁寧になったにもかかわらず、30分程度に短縮した。
- 予備機体の制作  
ほぼ同仕様の機体を2機製作したことで、試験の並行化と効率化を実現した。
- 砂漠環境における接着剤  
瞬間接着剤は空気中の水分との反応により硬化するため、砂漠での使用時に硬化時間が遅いことが問題として挙げられる。これが実験環境との差を生むため、暖かく湿った息を吹きかけることで硬化の反応速度を大きくし、確実な接着を実現した。

### 第8.2.2項 電装班

- 開発全体を通して、電装系でのトラブルは少なく抑えられた  
確実に動くものを提供するという目的は達成したと考える。  
コネクタを使い分けることでヒューマンエラーによる接続ミスを回避できた。
- 溶断機構
  - ニクロム線に対して衝撃荷重がかからない設計
    - PLAのテグスガイドによりテグスからニクロム線自体には荷重が伝わらず、放出や着地時の断線の可能性を排除した。
  - テグスをニクロム線に対して非接触で溶断
    - 加熱時のムラが少なく、溶断後の断面が綺麗な断面となりエンベロップの穴や、アームとモーターの接合部に引っかかりにくくなった。

- 接触溶断の場合に生じた、接触状態による切れ方の差異を減らすことに成功
- 治具によりニクロム線のアーチ形状の再現性を高め、ニクロム線とテグスの位置関係の再現性を高めた。(8.1節参照)
- テグス再結合対策で実験をもとに10秒間電流を流した。
- 徹底した絶縁対策
  - 能代大会での反省も踏まえて、普段の試験段階からショートによる事故の危険がないか細かく観察し、テープやソルダーレジスト補修剤により絶縁対策を施した。

### 第8.2.3項 制御班

- シミュレーションの導入

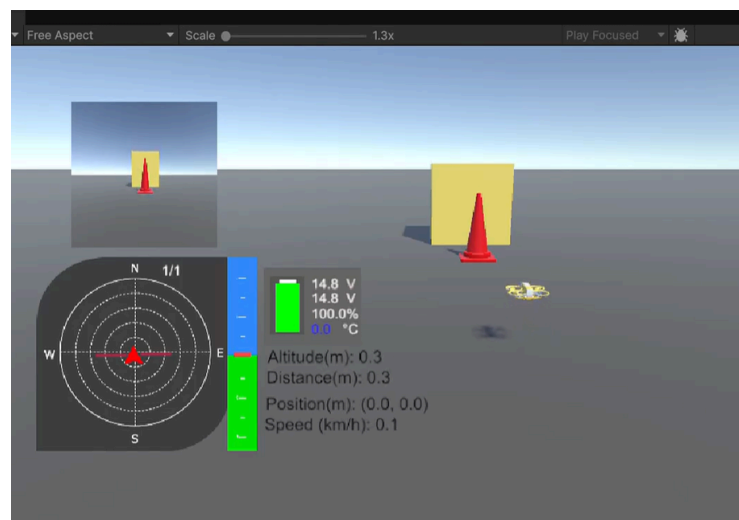
今年度は、東大チームにとって長年の課題であったPX4の「ブラックボックス」問題に対し、箱庭ドローンシミュレータを導入することで大幅な改善を実現した。

PX4やMAVSDKは、ドキュメントに記載のない挙動や仕様が多く、従来は「実際に何が起きているのか」が把握しにくい状況にあったが、シミュレーション上で事前実験を重ねることで挙動の不確実性を減らし、これまで見えなかった内部仕様を解明することができた。

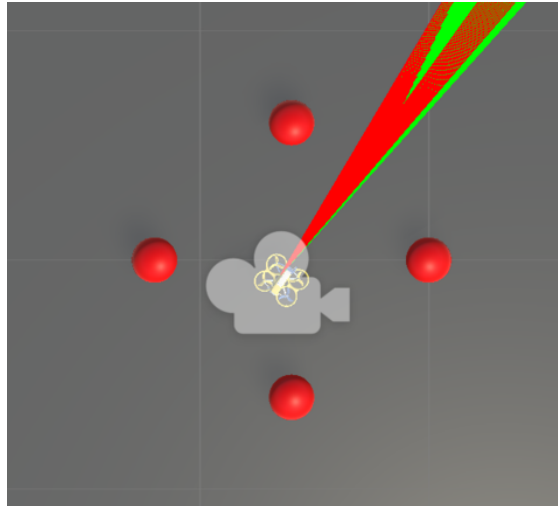
さらに、実機では再現が難しい冗長設計の検証も、仮想環境上で故障状態を模擬することでアルゴリズムの健全性を確認することが可能となった。

また、飛行試験場に限られる中で、場所を選ばず制御実験を行える環境を確立したことにより、文法エラーやアルゴリズムエラーが事前に検出され、飛行試験の効率と安全性が大幅に向上した。

シミュレーションでは、特に画像航法による着陸位置制御を再現した。Unity上で作成したゴールモデルをドローンカメラで撮影・認識し、中心座標に誘導する一連の制御を再現した(下図)。



シミュレーションの様子1



シミュレーションの様子2

その結果、実機での画像航法試験においてもゴールを正確に認識・誘導でき、シミュレーション環境が実験準備の精度を高めることを確認した。

さらに、ARLISS特有の重量制限下でプロペラガードを装着できないという安全上の制約に対しても、事前のシミュレーションにより危険挙動の発生確率を低減できた。

このように、箱庭ドローンシミュレータは試験効率化・安全性向上・知見共有の観点で極めて有効なツールとなり、チーム全体の開発プロセスを大きく前進させたと考える。

- 属人化解消の成功

LoRa、シミュレーション以外は制御班全員が理解し、デバッグできた。複数人で議論しながら開発を進められたことで、大幅な時間短縮と再現性の確保に繋がった。

- 高精度画像航法の実装

画像ごとに色相範囲を調整し、アメリカに行ってからでも物体検出のミスは1度もなかった。

- 飛行高度についての冗長設計

大きく分けて2つの冗長設計を行った。

1. LiDARによる別系統からの高度監視

前年東大チームのフライバックでは、おそらくGPS誤差により高度が低下し、墜落してしまった。この対策として、GPSとは独立したLiDARによる高度監視を追加した。高度が5m未満になったときに、機体高度を5m上げ、GPS高度も5m修正するというものである。

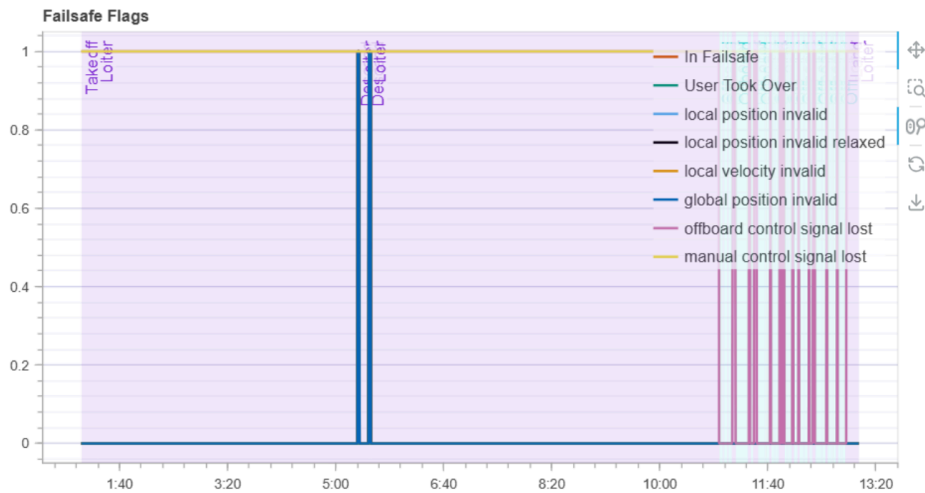
また、実験の際に、LiDARに砂がかかってしまい、値が実際の高度より小さい値をとることがあった。この対策として、LiDARが故障しているかを判定し、故障判定が出た場合は、今後のシーケンスから切り離すという処理も加えた。

2. Waypointによる飛行経路設計

従来までの飛行経路は、離陸後ゴールまで直線経路を飛行していた。この場合、長距離飛行の際、GPS誤差により高度が下がり、墜落する可能性がある。また、フライトコントローラの処理がブラックボックスなため、本番で初めて長距離をすることに不安があった。

そこで対策として、飛行経路を100 mごとに分割し、Waypoint(経路点)を設定する方式に変更した。この方法では、GPS誤差が大きくなった場合に、次のWaypointの高度を上げることで、墜落を回避できる。また、Waypointを100 m間隔で設定することで、日本国内でもARLISS本番と同様の条件を再現した飛行試験が可能となった。本番では100 mの直線飛行を複数回繰り返すため、日本でも同様の状況を模擬できると考えた。

- **ARLISS**閉会式後の飛行試験の際、冗長系が発動  
ARLISS閉会式後の実験にて、冗長系の作動を確認できた。20秒おきにセットポイントを送り直すことで飛行中にGPS情報が瞬間的に失われても飛行を続けることができるという設計である。  
飛行中にGPSの一時的な不良によりfailsafeが発動したが、復帰して87cmまで向かった。



## 第8.2.4項 マネジメント

- 1日目朝の打ち上げ及びミニマムサクセスを達成  
開発スケジュール・現地での準備がある程度機能したといえる。
- 機体を触らないMTGの継続的な実施  
学期期間中は週に1度、夏休み中は毎朝、機体を触らないMTGを継続して行えた。  
特に夏休み期間の毎朝の朝礼MTGでは、前日の試験結果・当日の試験予定・それに伴うタスクを整理・共有してから開発に取り組むことができた。  
本番直前の時期にも手を動かしたい気持ちを我慢してこのような時間をモテたことは、スムーズな開発に繋がった。
- ドキュメンテーション・文書管理の徹底  
知見や試験記録のドキュメンテーションを徹底することができ、開発全体の促進につながった。試験動画も、コツコツとYouTubeにアップロードして整理することができた。特に試験記録がデバッグに大変役立った。
- 打ち上げ手順書  
2.14.1でも述べた通り、ARLISSでは、能代大会での反省を活かし、打ち上げ手順の確立と紙の手順書の活用によって大会当日の運用フローを含めた再現性を実現できた。  
： 打ち上げ前手順書\_報告書提出用  
ただし、打ち上げ3回目には整備不良があり、手順が確立していても作業者の疲弊が再現性を狂わせてしまうということを学んだ。

## 第8.3節 課題点

### 第8.3.1項 構造班

- 配線・固定／物品管理／CAD規則の徹底  
長期的な課題であるにもかかわらず着手が遅れた。特に配線周りと物品管理はずっと明白な問題だったが、軽視し取り掛かろうとしなかった。結果、能代では離陸シーケンスに到達できず、またARLISSでもモータの予備数に不安が残る中での打ち上げとなってしまった。
- 組み立て作業の効率化に早い段階で意識的に取り掛かるべき  
能代でここが足を引っ張り、そこで初めて明確に認識された。当日会場での作業だけでなく、繰り返し行う試験の時短にも繋がる重要な部分だった。「配線・固定／物品管理／CAD規則の徹底」にも通ずる部分だが、「環境整備による作業効率の向上」は意識したい。
- 制御のログ解析に積極的に首を突っ込み、ハードで直せるバグを回収すべき  
試験で沼った時、ハードをひたすら疑う人が必要。制御班はコードの沼にいるものと思ふべき。
- アーム破損の多発  
強風に十分に強くなかった。ARLISS本番での故障はなかったが、アメリカでの画像航法試験中にアームが折れて機体が墜落するということがあった。振り返ると、アーム単体での荷重試験を行えばより良い設計フィードバックが回せたと考える。
- 設計変更の多さ  
仕様変更が重なったため、ある程度は仕方ないことだったと考えている。しかし毎回、加工組立や重量管理、ゲイン調整、試運転など膨大な作業になる設計変更回数を減らす工夫をすべきだった。例えば、開発の中終盤にセンサの変更に基づく設計変更が入ったが、序盤の選定、動作確認が機能すれば早期に防げた変更だった。

### 第8.3.2項 電装班

- 予備物品の用意不足  
動作確認もできているモーターが少なく、現地でモーター確認の手間を増やしてしまったり、そのほかの電装部品に関しても残り個数がギリギリの状態を開発を進めることが多かった。  
UART変換基板に関しては能代において実際に予備が少なく予期せぬトラブルではあったものの、最終的に3回目の実験を諦めざるをえない状況を生み出してしまった。
- 発生した不具合を完全には解明できなかった  
GNSSモジュール:アメリカに到着以降GPS座標の取得に苦戦するようになった。日本で同様の事象が起きたときは予備部品に変えることで対応できたため、問題視していなかったが、アメリカではそれだけでは対応できないものであった。UART変換基板の固定位置の変更時期とGPSの不調の時期が重なっているのではという仮説から物理的に距離を遠ざけ一定の改善は見られたが、依然として不安定なままであり、またなぜ一定の改善が見られたのかについての合理的な説明はつけられない。  
UART変換基板の故障:能代において2回故障した物品だが、故障原因については再現実験ができておらず特定できていない。どこかにあたってショートしたという仮説をもとに絶縁を強化するという対症療法により再発を防ぐことができたのはよかったが、反省すべき点である。また原因究明時に各モジュールがいつ使用されたものか、いつ購入したものかなどの個別の管理ができていなかったこともこの問題に深く関わる部分である。
- パーツ管理  
各モジュールの分類については徹底的に実施しており、細かい電装部品の管理に問題はないと思っていたが、予備部品などの動作確認が住んでいるかや、各試験においてどの

パーツを使ったか見分けられるようにするなど、トラブルが起きた際の対応までは不十分であり、大会当日に起きたトラブルに対して対応が不十分になってしまった。

- パワーモジュールの必要性に気付くのが遅れた  
バッテリー容量の見積もりの甘さ
- **EtoE**試験の遅さ
- 構造班とのコミュニケーション不足
  - 配線までを考慮した設計にできなかった
  - 構造確定までに要する工数を増やしてしまい全体の開発ペースを遅らせることにつながった。
- 個体差に対する許容誤差
  - 手作業により生まれる個体差に対する許容誤差を明確に定めていなかった。
  - 大会での打ち上げ3回目の失敗につながる

### 第8.3.3項 制御班

- プルリクエストの自己承認は禁止すべき  
自分でプルリクエストを承認(マージ)することは強く禁止するべきだった。一部行われてしまった。
- 他班との連携不足  
制御班のみで問題を抱え込み、原因をソフトウェア側に限定して考えすぎる傾向があった。実際にはハードウェアや構造的要因に起因する不具合も多く、問題解決の遅れにつながった。  
また、ログ解析結果をチーム全体に解説・周知する場が不足しており、他班が制御上の課題を把握しにくかったということも、問題解決の遅れの原因として挙げられる。

### 第8.3.4項 マネジメント

- 全体スケジュールの問題点(**BBM**の費用対効果の低さ)

能代大会までの開発は、**BBM**→**EM**→**PFM**の三段階開発を計画的に実施し、開発リズムの確立といった点では一定の成果を得た。

しかし、**BBM**(Bread Board Model)段階の費用対効果が低かったことが、後続の**EM**開発の負荷を大幅に増大させてしまったと考えている。

**BBM**では、フライトコントローラー/**QGroundControl**の習熟や、**EM**ペラ振動問題(2.14.3参照)の早期発見などの成果が得られたものの、構造的課題が少なく、より本質的な開発経験にはつながりにくかった。

その結果、**EM**(Engineering Model)開発以降に発生した設計変更や試験項目の増加を十分に予見できず、スケジュール全体が後倒しとなった。

**BBM**に費やした時間を過去機体の分析や勉強会に充てていれば、**EM**開発をより高い完成度で進められた可能性がある。

- 概念設計の不足  
4~5月まではガントチャート通りに進行し、各班の作業も順調だった。しかし、6月に入り**EM**開発が始まると、統合段階での干渉が多発し、スケジュールが大きく乱れた。これは、初期

段階での構想共有が不十分で、全員が同じ設計思想を持っていなかったことが原因である。

- 完成度  
結局、「1発で成功する機体」でないと3回打ち上げられたところでモグラ叩きになってしまう。「本番前にE2E試験で10連続成功」など明確なGO基準を設定すべきだった。
- 試験への向き合い方  
当初は「試験を重視しているつもり」であったが、実際には「自分たちが確かめたいことだけを確認する試験」にとどまっていたと振り返る。言い換えれば、試験の目的が「不具合を発見する」ことではなく、「安心を得るための確認」になっていた。

その典型例が長距離飛行試験である。先輩機体の墜落原因がバッテリー残量にあったため、本チームは電力監視を重点的に行った。しかし、モジュールを追加した9月1日以降に再試験を実施しなかった結果、1回目の打ち上げ失敗につながる重大なエラーを見逃すこととなった。

「ソフトウェアの追加はバッテリー残量に影響しないから、長距離飛行にも影響しないだろう」という思い込みが判断を誤らせた形である。

一方で、「壊す試験」を通じて不具合を発見できた成功例もあった。開傘衝撃試験では、単純な落下試験では問題が見られなかったものの、50kg荷重試験を実施した際に破断を発見し、改良につなげることができた。

本プロジェクトでは、単に「安全を確認する試験」ではなく「壊す試験」の重要性を強く実感した。今後は、試験を「安心のため」ではなく「不安を洗い出すため」に行うという原則を徹底し、目的を明確化した試験設計を行う必要がある。

- システム要求の定期的な見直しの必要性  
開発が進むにつれてシーケンスが詳細化し、自然とシステム要求が増加していったが、その都度要求全体を俯瞰的に見直し、修正するプロセスを設けていなかった。そのため、必要な試験の一部が漏れ落ちる結果となった。  
例えば、
  - 9/1のモジュール追加後に実施すべき長距離飛行試験を省略してしまったこと
  - ARLISS 3回目投下前に行うべきケーブル長および干渉確認の点検を見落とししたこと

などが挙げられる。

これらはいずれも「要求の変化に対して試験計画を更新する」というプロセスの欠如に起因している。

システム要求の定期レビューを実施し、要求の追加・削除・修正を明文化してチーム全体で共有する仕組みを導入する必要がある。

また、要求の更新に伴い試験項目を再整理するプロセスを組み込み、

開発の後半で新たな不確実要素が残らないようにすることで、より安定した開発が可能になると考える。

- ガントチャート

タスクの粒度は適切だったが、各タスクの完了基準が不明瞭であった。そのため、進捗判断が主観的になり、班や個人によって「完了」とみなす基準に差が生じた。

- **PM、サブPM、班長の役割分担**

本チームでは、サブPMが長期に渡り不在だったり、自身のタスクに追われたりしていたために、タスク管理とチームマネジメントがPM一人に集中した。サブPMが機能停止していることに早い段階で気づき、班長会議を実施するなどしてPMと班長とのコミュニケーションを密にする方向に方針転換すべきだった。

特に、PM自身が制御分野に苦手意識を持っていたこともあり、制御関連のタスクに負荷が集中した際、十分な調整が行えなかった。

結果として再起動対策やモジュール追加試験の未実施など、ミッション失敗に直接関わる重大なミスを見逃した。

- **班間のコミュニケーション不足**

開発中盤以降、班ごとの専門分化が進んだ結果、情報共有のタイミングがずれ、問題の発見・解決が遅れる場面があった。特に、構造・電装がハード側に、制御班がソフト側に閉じてしまい、お互いの連携が後手に回った。PMと班長それぞれのコミュニケーションに頼るのではなく、班長会議を実施して、班長間でのコミュニケーションを促進すべきだったと振り返る。

- **体調管理**

能代大会期間中、本チームでは熱中症による体調不良者が発生した。水分補給や休憩の声かけは互いに行っていたものの、ミッションを成功させたいという思いが強すぎたため、本人も不調に気づけず、倒れるまで作業を続けてしまった。

その結果、体調不良者が十分な回復を得られないまま能代を早期に離れることになり、本人にとっても非常に悔しい結果となった。

また、チームとしても大会進行の一時停止や対応要員の確保が必要となり、共同実験の運営全体にも影響を及ぼした。

さらに、このメンバーはARLISS本番への現地参加を断念することとなり、チーム全体にとっても大きな損失であった。

倒れたメンバー以外にも、能代から帰宅後に日射病のような疲労症状を訴える者が複数名いた。このことから、直射日光が非常に体力を奪っていたように思う。

来年度以降は、日除け対策と休息・補給のルール化を徹底する必要がある。

具体的には、

- すだれ等を用いて日陰を確保する
- 時間単位での水分・塩分補給を義務づける
- 滞在時間自体を午前・午後で区切り、過度な現地拘束を避ける

といった仕組みを導入することで、共同実験と安全確保を両立できると考える。

- **ARLISS本番でのメンバーの疲弊**

特にARLISS大会3日目あたりから、メンバーの疲弊が顕著だった。運用を含めた再現性を担保するには、作業者の健康も重要である。

CanSatは自律制御がテーマではあるものの、事前の準備にはヒューマンエラーが付き纏う。

前日に睡眠時間を確保するなど、打ち上げ本番にメンバーがそれぞれの能力を出し切れる環境作りというのも必要だと感じた。

- **心理的安全性の確保**

本プロジェクトでは、メンバーが意見を出しやすい環境づくりが不十分であり、心理的安全性が十分に確保されていなかったと考える。

その結果、本来開発に必要な技術的議論が十分に噛み合わず意見の衝突が建設的な形に転換されない場面が見られた。結果的にチーム全体の進捗を阻害してしまう一因となったと考える。

背景には、メンバー全員が同級生であり、かつARLISS未経験であるという東大チーム特有の事情が挙げられる。対等ゆえに「間違いは恥ずかしい」「場を乱したくない」といった意識が働きやすかったのではないかと考える。

具体的には、以下のような状況であった。

- 開発初期(4～6月前半):お互いに遠慮してしまい、指摘や議論が停滞した。
- 開発中盤(6月後半～8月前半):積極的にコミットするメンバー間では互いに指摘ができるようになり議論が進んだ一方、時間的コミットが少ないメンバーが議論の外に置かれがちになった。
- 開発終盤(8月後半～9月前半):初期の遠慮や議論の偏りが影響し、議論不足や衝突が顕在化した。

今後の改善策としては、以下の点が有効であると考えられる。

- ARLISS経験者の上級生等を積極的に議論に巻き込み、疑似的な階層構造を形成する  
:意見の違いが衝突ではなく議論として機能しやすくなり、開発にも反映されやすくなるだろう。
- PMを含むリーダー層が良い耳を持ち、意見を引き出す質問を意識的に実践する  
:経験やコミット量の違いに関わらず、発言機会を確保し議論の偏りを防ぐ。

一方で、メンバー全員が同級生であることは、うまく機能すれば有利に働く要素でもあると考える。開発以外での関係性を築きやすく、これが心理的安全性を高め健全な議論を促進する可能性があるからだ。

## 第8.3節 今後の展望

本年度の東大SEIKOKUの成果として、

- 振動特性・飛行安定性の優れたマルチコプター開発ノウハウの獲得
- 方向依存性のないエンベロープシステムの実証
- 治具開発・活用ノウハウの獲得
- 非接触型のニクロム線によるテグス溶断機構の実証
- PX4に対する理解の深化
- シミュレーションを活用したマルチコプター開発ノウハウの獲得
- 高精度画像航法の実装
- マルチコプター飛行中の冗長系の実証
- 初日朝打ち上げの達成
- 運用も含めた再現性の向上
- RaspberryPi再起動に関する知見の蓄積

が挙げられる。

東大チームは近年フライバックでの0mゴールを目指しており、段階的にシステムを確立してきた。

2023年度:フライバックに必要なシーケンス(着地、展開、起立、離陸、飛行、誘導、着陸)をEndtoEndで実行可能なシステムを初めて構築した。

2024年度:カーボンロッドの採用によって起立機構を中心とした構造部の再現性が向上し、特に起立シーケンスにおいて安定した挙動を確立した。

2025年度(本年度):シミュレーションの導入によってPX4の解析・動作理解が前年より進展し、飛行制御および画像航法の精度が向上した。ただし、非同期処理に関する理解が不十分であり、その影響が挙動不良として顕在化した。また、開発方針として「運用を含めた再現性」を掲げ、打上準備のマニュアル化・試験計画の整理を進め、治具の開発等の進展も見られた。その結果、初日朝打ち上げを実現することができた。

総じて、本年度は構造・起立・ニクロム線溶断の再現性改善に加え、運用手順の確立とPX4理解の深化により、墜落地点のゴールからの距離は昨年度より離れてしまったが、再現性の高いフライバック0mゴールの実現に最も近づいた年度であったと評価できる。

具体的には以下の点で課題が残っている。

- 非同期処理を含む制御系の信頼性向上
- 予備パーツの信頼性向上
- 打ち上げ準備における作業負担の軽減
- 初期設計段階における開発サイクルの高速化