

# ACTS報告書

提出日: 2021年 12月 17日

- チーム情報

CanSatチーム名	東京大学 DHIO
CanSatチーム 代表者	瀬戸翔一 sho1seto@gmail.com
UNISEC団体名	東京大学中須賀船瀬研究室
UNISEC団体 学生代表	大橋奈央
責任教員	中須賀真一 nakasuka@space.t.u-tokyo.ac.jp

- メンバー

役割	名前(学年)
PM, 制御	瀬戸翔一(B3)
サブPM, 制御	岡本岳(B3)
サブPM, 構造	安福亮(B3)
会計, 構造	船場陸央(B3)
構造	粟木早恵(B3)
回路	森裕一郎(B3)
回路	草次優樹(B3)
制御	木原開(B3)

- CanSatの製作目的・大会参加理由

例年東京大学航空宇宙工学科のB3はARLISSに出場しており今年も出場を目指して機体を製作した。日本ではマルチコプターでのフライバックの先例はないが、我々はフライバックでの0m達成を目指すべくマルチコプターのCanSatの開発を行った。

# 目次

第1章	ミッションについて	4
1.	ミッションステートメント(ミッションの意義と目的)	4
2.	ミッションシーケンス	4
第2章	サクセスクライテリア	5
第3章	要求項目の設定	6
1.	システム要求(安全確保, レギュレーションのための要求)	6
2.	ミッション要求(ミッションを実現するにあたり要求される性能)	7
第4章	システム仕様	8
1.	CanSat外観	8
2.	CanSat内観・機構	8
3.	システム図(CanSat搭載計器仕様一覧)	8
4.	アルゴリズム	8
第5章	試験項目設定	9
第6章	実施試験内容	10
1.	システム要求を満たすための試験内容	10
2.	ミッション要求を満たすための試験内容	10
第7章	工程管理, ガントチャート(スプレッドシートを推奨) 各担当(ハード・ソフト・全体などの進行状況・予定を記入)	11 11
第8章	大会結果報告	
1.	目的	
2.	結果	
3.	報告	10
第9章	まとめ	12
1.	工夫・努力した点(ハード, ソフト, マネジメント面すべて)	
2.	課題点	
3.	今後の展望	13

# 第1章 ミッションについて

## 1. ミッションステートメント(ミッションの意義と目的)

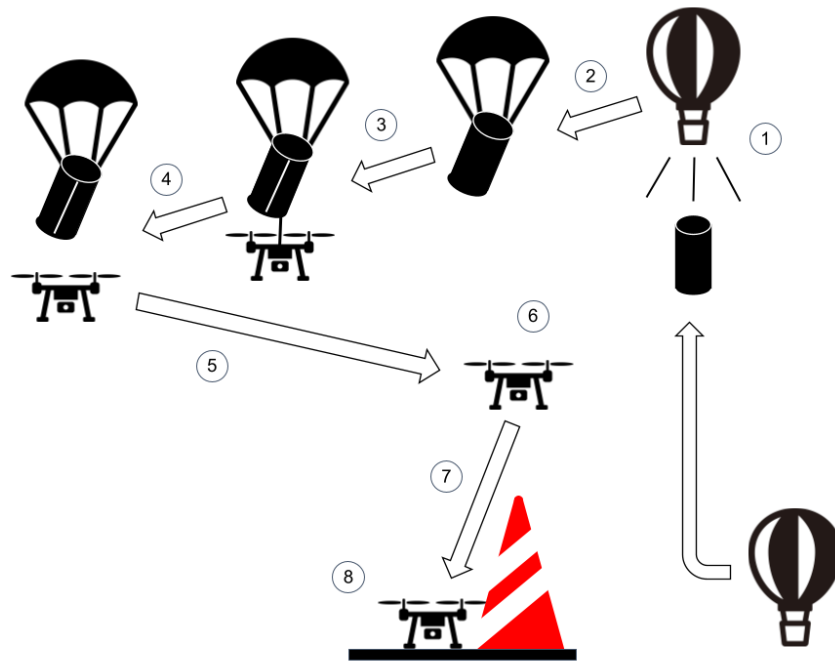
マルチコプターによる飛行中観測及び目標領域内への着陸

※目標領域: 目標物体との距離が1m以下となる領域

本学では数年前からARLISSにおいてフライバックに挑戦しており、我々もフライバックに挑戦する。フライバックは過酷な地表状況においても探査が可能な点で有意義だといえる。実際、多くの惑星や小惑星は地表温度が高い、障害物が多いなどの理由から着陸可能な範囲が狭く、着陸探査が難しい。それ故、目標地点まで着陸せずに到達することは重要である。我々はフライバックの手段としてホバリングが可能なマルチコプターを用い、飛行中の観測と高精度な着陸の実現を目指す。

## 2. ミッションシーケンス

大まかな流れは以下の通り。詳細は「第4章4. アルゴリズム」を参照されたい。



1. キャリアから機体の入ったケースが放出される
2. パラシュートが展開する
3. ケースがパラシュートの力で展開し、機体が放出される
4. ケースから機体が分離され、機体が飛行を始める
5. 目標領域内まで飛行する
6. 撮影画像から目標位置を補正する
7. 徐々に下降しながら位置を調整し、着地する
8. 終了処理を行う

## 第2章 サクセスクライテリア

ミニマムサクセス	<ol style="list-style-type: none"><li>1. パラシュートが展開/機能する</li><li>2. ケースが展開する</li><li>3. 機体とケースを分離できる</li><li>4. 機体のアームが展開できる</li></ol>
フルサクセス	<ol style="list-style-type: none"><li>1. 機体地上局間通信ができる</li><li>2. 機体が自律飛行できる</li><li>3. 機体が飛行中に地上を撮影できる</li><li>4. 機体が地上との距離を測定できる</li><li>5. 目標物体から5m以内に着陸できる</li><li>6. 適切な終了判定, 終了処理ができる</li><li>7. 機体が損傷せずに終了できる</li></ol>
エクストラサクセス	<ol style="list-style-type: none"><li>1. 撮影画像から機体に対する目標物体の位置が推測できる</li><li>2. 目標物体から1m以内に着陸できる</li></ol>

### 第3章 要求項目の設定

#### 1. システム要求(安全確保, レギュレーションのための要求)

要求番号	自己審査項目
	<a href="#">ACTS安全基準</a>
R1	<a href="#">質量と容積</a> がレギュレーションを満たすことが確認できている
R2	<a href="#">ロスト対策</a> を実施しており, 有効性が試験で確認できている (例: 地上局にダウンリンクする場合, ACTSで十分な通信距離が実現できるだろうと推測できる根拠が明確に示されていること.)
R3	地表近くで危険な速度で落下させないための <a href="#">減速機構</a> を有し, その性能が試験で確認できている
R4	打ち上げ時の <a href="#">準静的荷重</a> によって, 安全基準を充足するための機能が損なわれないことが試験で確認できている
R5	打ち上げ時の <a href="#">振動荷重</a> によって, 安全基準を充足するための機能が損なわれないことが試験で確認できている
R6	分離時の <a href="#">衝撃荷重</a> によって, 安全基準を充足するための機能が損なわれないことが試験で確認できている
R7	打ち上げ時の <a href="#">無線機の電源OFF</a> の規定を遵守できることが確認できている (FCC認証かつ100mW以下の機器はOFFしなくて良い. <b>また, スマートフォンを用いる場合はFCC認証かつソフトウェアまたはハードウェアスイッチでoffにできること(2017年追加)</b> )
R8	無線のチャンネル調整に応じる意思があり, また実際に調整ができることを確認できている
R9	R1-R8の充足を確認した設計の機体によって, ロケットへの装填から打ち上げ後の回収までを模擬したEnd-to-end試験を実施できている. 今後, 安全性に関わる大幅な設計変更はない
R10	CanSatの収納・投下準備が5分以内できている
	<a href="#">カムバックコンペティションルールの充足</a>
R11	ミッション時に人間が介在しない <a href="#">自律制御</a> を実施することが確認できている (注: 2014年のレギュレーション改定以降, 地上局設備に計算機能を持たせてアップリンクしても良い)
R12	ミッション後, 規定された <a href="#">制御履歴レポートを運営者へ提出</a> する準備ができている ( <b>以下の根拠の項に制御履歴レポートの例を添付すること. ダミーデータを使用しても良い</b> )

#### 2. ミッション要求(ミッションを実現するにあたり要求される性能)

要求番号	自己審査項目 ( <a href="#">ミッション実現要求項目</a> )

M1	機体地上局間通信ができること
M2	機体が4つのモータによる振動衝撃に耐えられること
M3	適切なタイミングでケースの展開ができること
M4	適切なタイミングで機体とケースが分離できること
M5	機体のアームが展開できること
M6	機体が自律飛行できること
M7	飛行中に地上の撮影ができること
M8	撮影画像から機体に対する目標物体の位置が推測できること
M9	機体が目標物体近くで着地できること
M10	適切な終了判定及び終了処理が行えること

## 第4章 システム仕様

### 0. はじめに

本章で説明する機体はACTS一ヶ月前のものである。諸事情により、それ以降のハード及びソフトの変更についての解説は「第9章 まとめ」にて付記する。

### 1. CanSat外観

機体の外観を図1に、機体の諸元を表1に示す。

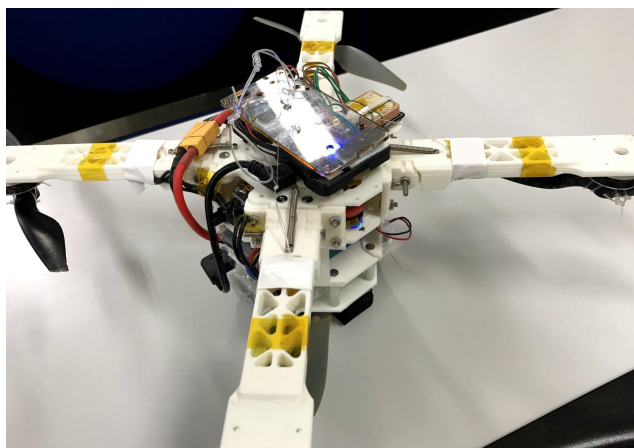


図1. 機体外観

	機体	機体(ケース収納時)	機体+ケース
全長(幅)[mm]	550mm	132mm	145mm
高さ[mm]	130mm	235mm	240mm
重量[g]	803g	-	1040g

表1. 機体諸元

### 2. CanSat内観・機構

#### 2.1 機体

- 概要

機体としてマルチコプターを採用した。条件に即した構造と高い再現性を実現するため3Dプリントを中心とした機体作製を行う。

マルチコプターのコアとなる部分は四段で構成され、それぞれの段にセンサーや電池などの搭載物を載せている。

マルチコプターのアームはケース収納時に下に折りたたみ、機体放出後、折りたたまれたアームを展開しアームの位置をロックすることにより飛行する。アームを上向き

に展開することで展開において揚力を有効に利用できることが利点としてあげられる。

- 各部位の説明  
マルチコプターの各部位について図2を用いて説明する。

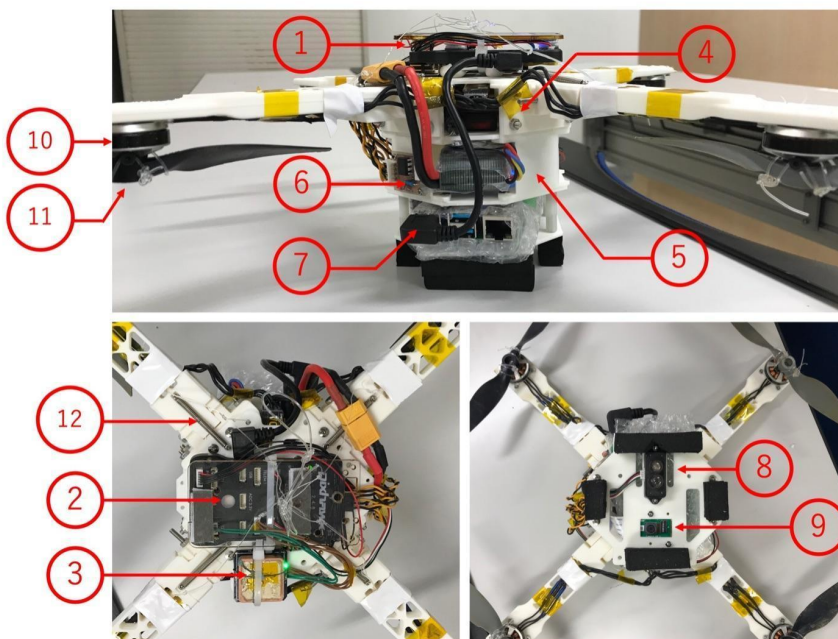


図2:ドローン各部位の説明

### <コア部>

#### 一段目(①)

Pixhawk(②)	Pixhawkの正面とドローンの正面が一致する向きで固定. 機体の振動がセンサーに与える影響を小さくするためにPixhawkと機体の間にクッションを挟んで固定している. また, ケーブルの劣化を防ぎ, 軽い衝撃に耐えられるようにカバーを取り付けている.
GPS(③)	Pixhawkの磁気センサなどがGPS内の磁気の影響を受けないよう, Pixhawkやブラシレスモーターからできるだけ離れていて, かつ上に障害物がない場所に配置.

#### 二段目(④)

分電盤/ パワーモジュール	ケーブルがLipoに繋ぎやすく, 収納しやすく, 問題がないかの確認できるように2つのみを中段中央部に配置.
テレメトリ	地上局との通信の為に配置.



### 三段目(⑤)

Lipoバッテリー	なるべく重心を下にするために三段目に配置. 取り出しやすいように十分な高さを確保している. ケーブルの収納を考慮して, 写真の向きに電池を配置することとした. なお, モーター駆動用の3セルと制御用電源の1セルを搭載している.
光センサ(CdSセル)(⑥)	アームの展開を検知するための光センサを, 機体側面に外側を向くようにビスで固定している.

### 四段目(⑦)

Raspberry pi 4	飛行指示の為に搭載. 落下や振動などから防護するために緩衝材で包んで四段目中央部に配置.
距離センサ(下面)(⑧)	地形に追従して正確な自己位置を推定するために距離センサを下面に配置.
ラズパイカメラ(下面)(⑨)	飛行中に地上を観測しつつ目標(カラーコーン)を画像認識するためのカメラを下面に配置.

### <アーム部>

ブラシレスモータ(⑩)	折りたたみ時にケースに収まるよう, 写真のように展開時に地面側に来るように配置. ビスによりアームに固定.
プロペラ(⑪)	7インチのものを使用. モーターが下向きに固定されていることに留意してプロペラを取り付け.

### <展開機構(⑫)>

- 展開時には引きばねを用いて腕を持ち上げる形で展開する. 飛行中はプロペラの揚力も加わることでロック機構がなくとも展開位置で固定される.
- アームの展開の様子は以下の動画の通り. 尚, 動画内のアームをロックする機構はなくても問題がないことから現在は取り付けられていない.

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

#### ● その他

- 一部壊れたときの修復性を考えて機体を一体で設計せずにアーム, コア部それぞれを別パーツとしている.
- アームには軽量で十分な剛性があるトラス構造を採用した.
- コア部は軽量化及び配線の柔軟性確保のため, 剛性に問題ない程度の肉抜きをしている.

- 固定はボルトナット締結を基本としている. ネジロックにより動作時にナットが外れないようにする.
- 試験機4号時点での概形及びケース収納時の概形は図3, 図4の通り.

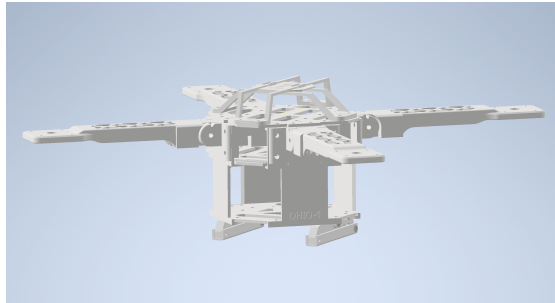


図3. 試験機概形

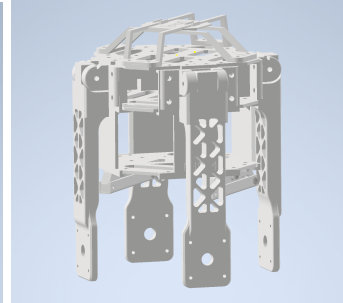


図4. ケース収納時の概形

## 2.2. ケース

- 概要  
ケースの概形は以下の図5のようにになっている.  
図5のように面A, 面Bを定め以下の説明に用いる.

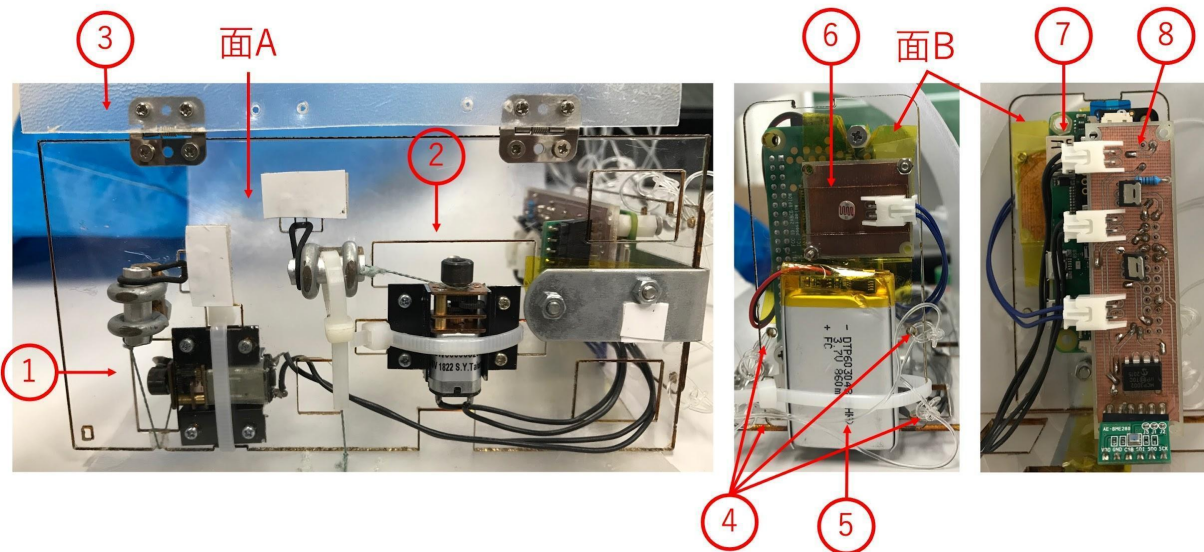


図5: ケース概形

- 素材

部品取付部 (面A, B)	高弾性をもつポリカーボネート材を使用. また, 大きな撃力がケースにかかった際に塑性変形を起こす危険があるため, 剛性を上げるために縦通材で補強している.
カバー(③)	柔軟性のあるポリプロピレンシートを使用.

- ケース展開方式

図6の番号と併せて説明する. 門方式(ギヤードモータ(①)と糸で繋がったアルミ棒(②)とケース用の糸(③)をアイボルト2つ(④)に通し, 展開時にギヤードモータを回してアイボルトからアルミ棒を引き抜くことで, ケース用の糸が外れて展開する方式)を取る. なお, 展開する前に門が抜けないう, 門にゴム(⑤)をかけて横から固定している. ケースの展開の様子は以下の通り.

<https://youtu.be/zN7195tRz-4>

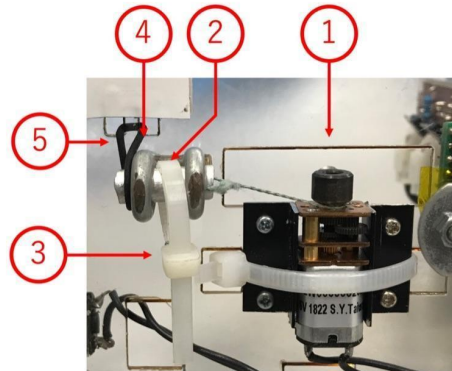


図6: 門構造

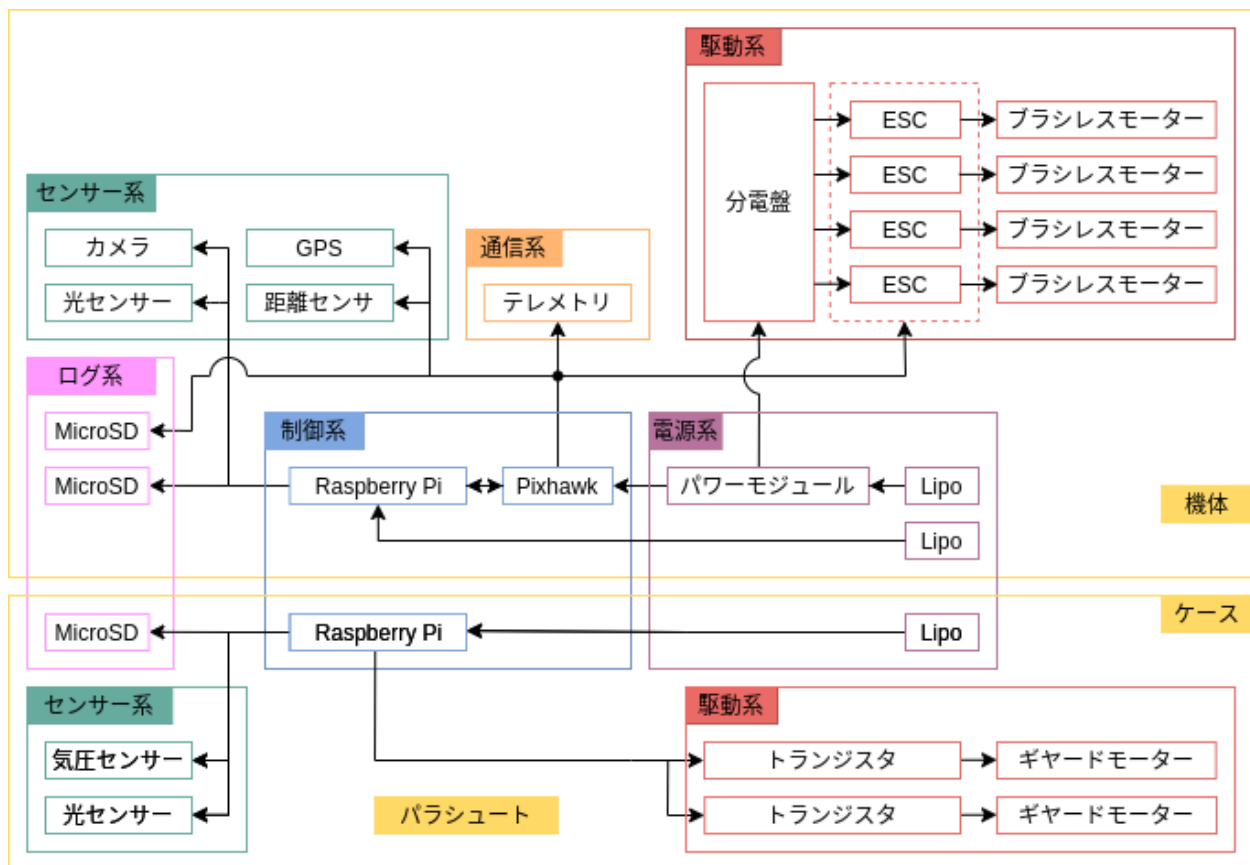
- 搭載物

門構造(①, ②)	面Aに配置. ①はケースと機体の分離に, ②はケースの展開に用いる.
回路基板/Raspberry Pi(⑦, ⑧)	面Bの内側に配置. パラシュートの分離やケースの展開を制御する.
光センサモジュール(⑥)	キャリア放出の判定に用いる. 面B外側に配置.
パラシュート(④)	面Bに配置. 糸の絡まりを防ぐため, 中央にスイベルを挟んで, ケースと繋いでいる.
Lipoバッテリー(⑤)	ケース側のラズパイやモーターに電源を供給する.

- パラシュート

- 現在終端速度5m/s以上で設計しているが(試験済), 50~100mという低高度から投下する今大会の特性上, 機体安定のために十分な時間が取れないことから終端速度を小さくする予定である. その際, 投下可能な最大横風風速を考慮し, 分離後のケース及びパラシュートがフィールド外に出ないように注意して設計する.

### 3. システム図 (CanSat搭載計器仕様一覧)



ケースの制御系としてマイコンのRaspberry Pi(以下ラズパイ)を, その電源には1SのLipoを用いている. ラズパイは気圧センサや光センサのセンサ値を取得し, ギヤードモータを駆動している.

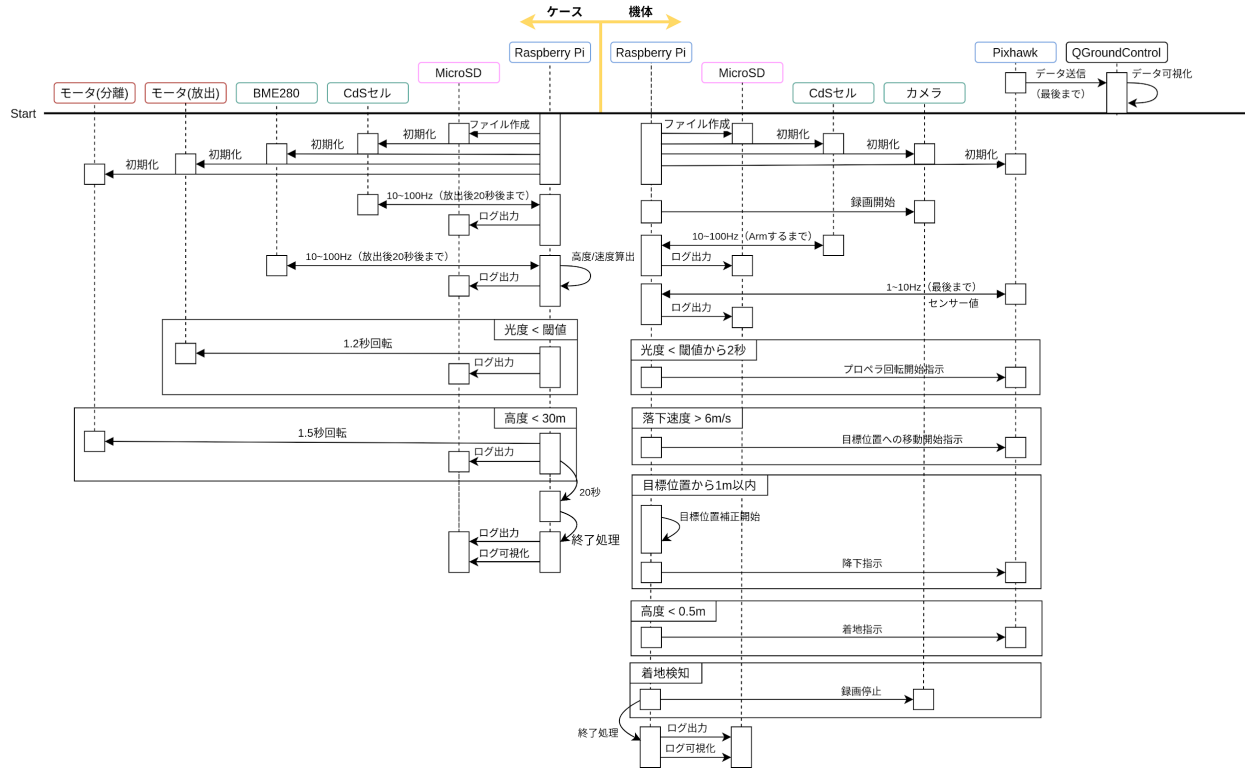
機体の制御系としてはマイコンのラズパイとフライトコントローラーのPixhawkを用いている. 電源はモータとPixhawkには3S, ラズパイには1SのLipoを用いている. Pixhawkは内部のIMUとジャイロの他, GPSと距離センサを備えており, モータを駆動して飛行制御を行う. ラズパイはPixhawkから各センサの値を取得しつつPixhawkに飛行指示をして出しており, カメラによる撮影と撮影した画像を用いた認識も行っている.

ログはマイコンとフライトコントローラーそれぞれで保存している.

搭載計器	用途	データシート等
Raspberry Pi 4	マイコン	<a href="https://akizukidenshi.com/catalog/g/gM-14778/">https://akizukidenshi.com/catalog/g/gM-14778/</a>
Raspberry pi zero wh	マイコン	<a href="https://akizukidenshi.com/catalog/g/gM-12961/">https://akizukidenshi.com/catalog/g/gM-12961/</a>
Pixhawk	フライトコントローラ (機体の姿勢や速度などを制御するための装置)	<a href="http://www.holybro.com/manual/Pixhawk4-DataSheet.pdf">http://www.holybro.com/manual/Pixhawk4-DataSheet.pdf</a> <a href="https://docs.px4.io/en/flight_controller/pixhawk.html">https://docs.px4.io/en/flight_controller/pixhawk.html</a>

Lipo SHIM	マイコンへの電源供給	<a href="https://shop.pimoroni.com/products/lipo-shim">https://shop.pimoroni.com/products/lipo-shim</a>
3S Lipo(機体Power)	Pixhawk及びモーターの電源供給	<a href="https://www.amazon.co.jp/gp/product/B07H7GSOMQ/ref=ppx_yo_dt_b_asin_title_o03_s00?ie=UTF8&amp;psc=1">https://www.amazon.co.jp/gp/product/B07H7GSOMQ/ref=ppx_yo_dt_b_asin_title_o03_s00?ie=UTF8&amp;psc=1</a>
1S Lipo(機体Logic)	マイコンへの電源供給	<a href="https://www.marutsu.co.jp/contents/shop/marutsu/data-sheet/836419.pdf">https://www.marutsu.co.jp/contents/shop/marutsu/data-sheet/836419.pdf</a>
1S Lipo(ケース側)	ケース側の電源供給	<a href="https://www.marutsu.co.jp/contents/shop/marutsu/data-sheet/Spec-01-603048(PHR)-860mah-1C-en-1.0ver.pdf">https://www.marutsu.co.jp/contents/shop/marutsu/data-sheet/Spec-01-603048(PHR)-860mah-1C-en-1.0ver.pdf</a>
パワーモジュール	機体側の電源供給	<a href="https://www.amazon.co.jp/dp/B00HC5XGC0/ref=cm_sw_r_cp_apa_glt_i_GB10JBKTJ8263DSGNVG5?_encoding=UTF8&amp;psc=1">https://www.amazon.co.jp/dp/B00HC5XGC0/ref=cm_sw_r_cp_apa_glt_i_GB10JBKTJ8263DSGNVG5?_encoding=UTF8&amp;psc=1</a>
光センサ	キャリアからの放出検知	<a href="http://akizukidenshi.com/download/ds/senba/GI5528_1_M.pdf">http://akizukidenshi.com/download/ds/senba/GI5528_1_M.pdf</a>
大気圧センサ	高度推定	<a href="https://akizukidenshi.com/download/ds/bosch/BST-BME280_DS001-10.pdf">https://akizukidenshi.com/download/ds/bosch/BST-BME280_DS001-10.pdf</a>
トランジスタ		<a href="http://akizukidenshi.com/download/ds/toshiba/2SK4017.pdf">http://akizukidenshi.com/download/ds/toshiba/2SK4017.pdf</a>
ADコンバータ		<a href="https://akizukidenshi.com/download/ds/microchip/mcp3002.pdf">https://akizukidenshi.com/download/ds/microchip/mcp3002.pdf</a>
ブラシレスモータ	機体の動力	<a href="https://store.shopping.yahoo.co.jp/littlebellanca/21363.html">https://store.shopping.yahoo.co.jp/littlebellanca/21363.html</a>
ESC (DR-20A)		<a href="http://www.gforce-hobby.jp/products/G0211.html">http://www.gforce-hobby.jp/products/G0211.html</a>
ギヤードモータ	門の抜き取り	<a href="http://akizukidenshi.com/download/ds/shayangye/RA12W006000202R.pdf">http://akizukidenshi.com/download/ds/shayangye/RA12W006000202R.pdf</a>
Micro SD	ログの記録	<a href="https://akizukidenshi.com/catalog/g/gS-15844/">https://akizukidenshi.com/catalog/g/gS-15844/</a>
カメラ	飛行中の観測	<a href="https://www.amazon.co.jp/サインスマート-魚眼レンズ-カメラモジュール-Camera-Module-Raspberry/dp/B00N1YJKFS">https://www.amazon.co.jp/サインスマート-魚眼レンズ-カメラモジュール-Camera-Module-Raspberry/dp/B00N1YJKFS</a>
距離センサ	自己位置推定における高度補正	<a href="https://www.amazon.co.jp/Lidar範囲センサーモジュール-Benewake-single-point-RangingモジュールLidar検出器-for-raspberry/dp/B08G52Z9KL">https://www.amazon.co.jp/Lidar範囲センサーモジュール-Benewake-single-point-RangingモジュールLidar検出器-for-raspberry/dp/B08G52Z9KL</a>
GPS	自己位置推定	<a href="https://docs.px4.io/master/en/gps_compass/">https://docs.px4.io/master/en/gps_compass/</a>
テレメトリ	機体地上局間通信	<a href="http://www.holybro.com/product/transceiver-telemetry-radio-v3-915mhz/">http://www.holybro.com/product/transceiver-telemetry-radio-v3-915mhz/</a>

## 4. アルゴリズム



ケースは初期化の後、気圧センサや光センサの値を10~100Hzで取得し続けており、光センサがある閾値を下回ったときに展開を検知して展開用のモータを回してケースが展開する。次に高度が30mを下回ってから分離用のモータを回すことで機体が分離する。その後十分に時間が経過した後、各センサの終了処理を行う。ここで各センサとも観測誤差が十分に大きいとして強めにカルマンフィルタをかけており、飛び値によって誤作動することがないようにしている。これは機体側の光センサでも同様である。

機体は初期化の後、光センサの値を10~100Hzで、Pixhawkのセンサ値を1~10Hzで取得している。光センサが閾値を下回ったときに展開を検知し、そこから2秒後にプロペラの回転指示を出す。その後、機体が分離後自由落下して落下速度が6m/s以上になったときに、目標位置への移動指示を出す。目標地点に近付いたら認識を用いた目標位置補正を出し、50cm以下になったら着地指示を出す。その後各センサの終了処理を行う。

尚、これまで200回以上時と場所を変えて光センサと気圧センサの動作実験をしたが誤作動を起こしたことはない。ケーブルなどの不具合があった場合には各センサとも値が0となるが、その際は特に何も実行されず、そのままパラシュートで降下する。その他、降下指示を出してからある一定時間以上着地しなかった場合やバッテリー残量が少なくなった場合、地上局との通信が途切れた場合に、着地処理に移るようにしている。

※以下のリンクからも参照できる。

[https://drive.google.com/file/d/1iaxNybHT5zfvO76w-HwJHVY-\\_sX\\_K\\_B-/view?usp=sharing](https://drive.google.com/file/d/1iaxNybHT5zfvO76w-HwJHVY-_sX_K_B-/view?usp=sharing)

## 第5章 試験項目設定

番号	検証項目名	対応する自己審査項目の 要求番号(複数可)	実施予定日
V1	質量試験	R1	7/20
V2	機体の収納・放出試験	R1	8/12
V3	準静的荷重試験	R4	7/22
V4	振動試験	R5	8/20
V5	分離衝撃試験	R6	8/20
V6	パラシュート投下試験	R3	6/26
V7	展開試験	M3, M5	8/12
V8	分離試験	M4	8/12
V9	手動飛行試験	M2	7/16
V10	自律飛行試験	M6, M10, R11	8/12
V11	目標物体検知試験	M8	9/20
V12	撮影試験	M7	8/23
V13	GPSデータダウンリンク試験	M1, R2	8/13
V14	End-to-end試験	M6, M7, M9, R9	10/23
V15	制御履歴レポート作成試験	R12	10/23
V16	収納試験	R10	8/12
V17	無線チャンネル変更試験	R8	8/23
V18	無線機電源オフ試験	R7	8/13

## 第6章 実施試験内容

### 1. システム要求を満たすための試験内容

#### (V1) 質量試験

- 目的  
CanSatとパラシュートを合わせた質量が規定質量である1050g以下を満たすことを確認する
- 試験内容  
CanSatとパラシュートを質量計で計測をし、レギュレーションに記載されている質量(1050g)以下であることを確認する。
- 結果  
機体とケースとパラシュートの総重量は1040g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7.1.1にCansat機体質量測定結果の様子を示す。尚、内訳は機体803g、ケース196g、パラシュート41gである。

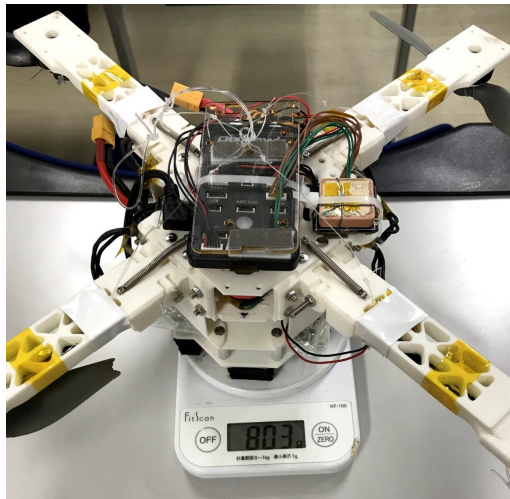


図7.1.1 CanSat機体の質量

- 考察  
パラシュートの質量を含めたCanSatの総重量が、レギュレーションを満たしていることがわかった。

#### (V2) 機体の収納・放出試験

- 目的  
機体がキャリア(内径146mm, 高さ240mm)に収納でき, 自重で放出できることを確認する。
- 試験内容  
CanSatをキャリアに収納後, 自重で放出されることを確認する
- 結果  
機体の収納・放出が問題なく行えることを確認した。                      動画は以下。

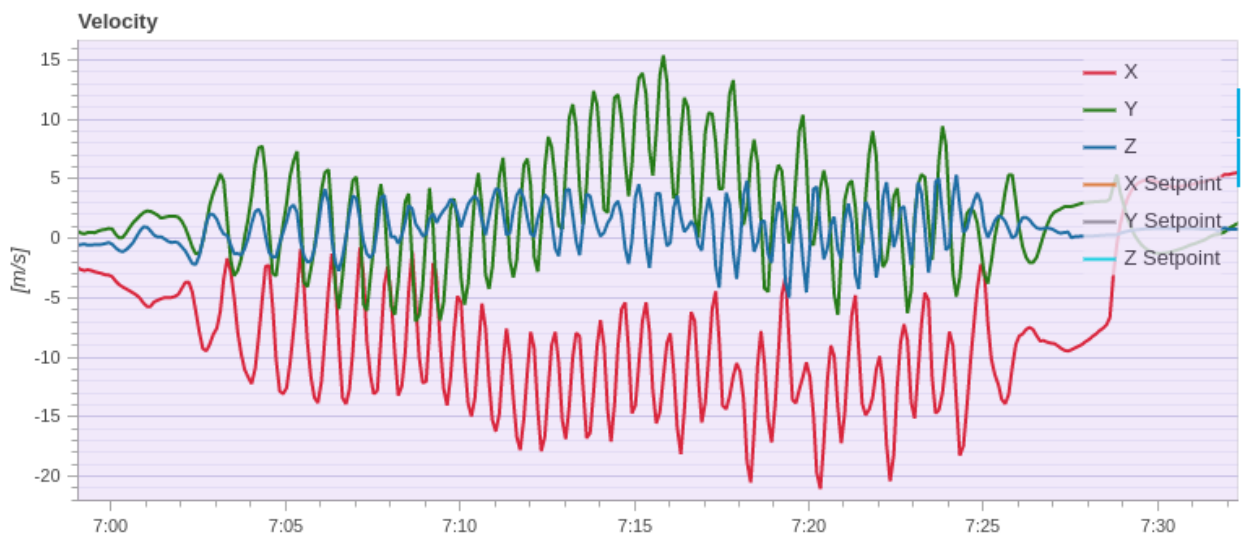


[https://www.youtube.com/watch?v=xmIS\\_8\\_7nmc](https://www.youtube.com/watch?v=xmIS_8_7nmc)

- 考察  
特に問題ないとする。

### (V3) 準静的荷重試験

- 目的  
ロケットによる打ち上げを想定した際に打ち出しにかかる準静的荷重に機体が耐えられることを確認する。
- 試験内容  
10Gの荷重に機体が耐えられることを確認する。
- 結果  
10秒超10G以上の荷重に、機体が耐えられることを確認した。その後も正常に飛行できることを確認した。結果は以下の動画の通りである。  
<https://youtu.be/xjn72TVMW8o>  
この時のセンサ値は以下の通り。



- 考察  
x, y, z方向の速度はそれぞれ平均を取ると約4m/s, 1/s, 8.5m/sである。これらより速度vは約9.4m/sと求まる。重量mは約1.04kg,半径rは約0.8mであることから、荷重は $m \cdot v^2 / r \div 116N \div 11.8G$ となる。ここで、センサ値には誤差があることと速度方向が必ずしも接線方向ではないことを考えると実際は10%程度の誤差があると考えられる。だが、誤差を考慮に入れても10Gの荷重には耐えているので問題ないと結論付ける。

### (V4) 振動試験

- 目的  
ロケットによる打ち上げを想定した際に打ち出しにかかる振動に機体が耐えられることを確

- 認する.
- 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.
- 結果  
[ランダム振動試験](#)を実施した. 振幅は15G以上である. 試験実施後においても機体に損傷は見られなかった. 試験前から懸念していた振動によるねじのゆるみ・はずれも見られなかった. また, その後の動作確認においても正常に飛行できる事が確認できた.
- 考察  
試験の結果を考えると本番においても十分安全な飛行ができると考えられる. 今後は振動によって緩む可能性の高いねじなどを中心に対策を練っていききたい.

## (V5) 分離衝撃試験

- 目的  
ロケットによる打ち上げを想定した際にロケットからの分離時にかかる撃荷重に機体が耐えられることを確認する.
- 試験内容  
40Gの撃荷重に機体が耐えられることを確認する.
- 結果  
[撃荷重試験](#)を実施した. 荷重の大きさは40G以上である. 試験実施後においても機体に損傷は見られなかった. また, その後の動作確認においても正常に飛行ができることが確認できた.
- 考察  
試験の結果を考えると本番においても十分安全な飛行ができると考えられる. 今後は撃荷重に弱いと考えられる各接続部の強化を中心に対策を練っていききたい.

## (V6) パラシュート投下試験

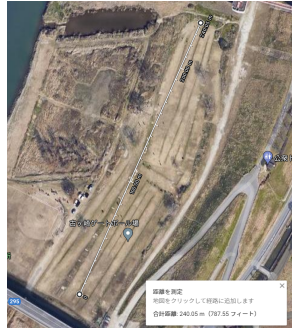
- 目的  
パラシュートの開傘と減速が可能であることを確認する.
- 試験内容  
パラシュートの投下試験を行い, 開傘と減速を確認すると共に終端速度を計測する.
- 結果  
<https://youtu.be/VlSyNm9FZE8>  
終端速度は気圧センサより求まる高度を微分した値から6.2m/sと求まった
- 考察  
パラシュートの開傘と減速が確認できた. また終端速度は5m/sより大きくなっている. 今後パラシュートを変更した場合は都度試験を行うこととする.

## (V13) GPSデータダウンリンク試験

- 目的

機体と地上局(QGCというソフトウェアを利用(<http://qgroundcontrol.com/>))の間でどれくらいの通信距離が可能か試験する。

- 試験内容  
河川敷などの開けた場所において距離ごとに通信が可能かどうか確かめる
- 結果  
GPSデータに加え、6軸センサなどのセンサ値が以下の動画のように240m程度(下図参照)移動しても取得できている。これは投下高度の50mを十分に満たしている。  
<https://www.youtube.com/watch?v=Q6KcneaDueQ>



- 考察  
投下高度でのダウンリンクは十分に可能である。

## (V14) End-to-End試験

- 目的  
CanSat投下からパラシュート展開, ケース展開, 機体分離, 自律飛行, 終了判定, 終了処理, データ取り出し, データ可視化の一連動作ができることを確認する。
- 試験内容  
試験飛行にて上記の一連動作を確認することは難しいため, 機体分離前(1)と機体分離後(2)それぞれで動作を確認する。尚, 分離動作については(V8)分離試験にて確認済みである。
  1. キャリアから放出->ケース展開->機体放出
  2. (機体分離後は機体を自由落下させることから)上空から機体を自由落下させる->目的地に向かう->着地
- 結果  
それぞれ以下の動画の通り
  1. <https://youtu.be/GwgTufveM10>
  2. [https://youtu.be/\\_NWqdDQPqBQ](https://youtu.be/_NWqdDQPqBQ)
- 考察  
機体分離前はキャリア放出後問題なくシーケンスが進行していることが確認できる。機体分離後は不安定な挙動のように見えるが, (V15)で示すように目標の動きが達成されている。それぞれ高度12m地点から落としており, ACTSの投下で想定される最低高度30mから落としても問題ないと考えられる。

### (V15) 制御履歴レポート作成試験

- 目的  
End-to-end試験のデータから制御履歴レポートを作成できることを確認する。
- 試験内容  
「どういうデータが取得されたのか」「どういう判断を下したのか」「どういう出力をしたのか」が明確にわかるような制御履歴レポートを提出する。
- 結果  
制御履歴を以下のように作成した。  
[本審査書 制御履歴レポート](#)
- 考察  
能代宇宙イベントにて成功した際の制御履歴を作成した。

### (V16) 収納試験

- 目的  
CanSatを素早く収納できるようにする。
- 試験/解析内容  
CanSatを5分以内で収納し、投下準備ができることを、動画に撮って示す。また、本番、CanSatを収納～投下するまでの手順を箇条書きで書き出し、各ステップの所要時間も込みで書く。
- 結果  
下記動画の通り、5分以内に収納できることが確認できた。  
<https://youtu.be/uCFFEaai30A>
- 結論  
本番で、準備時間5分以内という条件は十分満たすことができると考えられる。

### (V17) 無線チャンネル変更試験

- 目的  
無線のチャンネルを変更できることを確認する。
- 試験/解析内容  
テレメトリ通信モジュールにおいて周波数を任意に変えられることを確認する。
- 結果  
下記の動画の通り周波数を変更することが確認できた。異なる周波数帯で通信を行うテレメトリ通信モジュール2つで同時に通信を行っても干渉しないことも確認できた。  
<https://youtu.be/AmFzzEidSp0>
- 結論  
大会当日に周波数の変更を要請されたとしても対応可能である。

### (V18) 無線機電源オフ試験

- 目的  
レギュレーションの無線機の電源オフの規定を満たすことを確認する。
- 試験/解析内容

FCC認証かつ100mWの機器を用いているためレギュレーションを満たしており、試験の必要がないと判断した。

- 結果
  - 考察
- 

## 2. ミッション要求を満たすための試験内容

### (V7) 展開試験

- 目的  
キャリアから放出された後、光センサが光を検知しケースが展開されることを確認する。
- 試験内容  
キャリアからCanSatを放出する。機体と等しい重さのおもりを用いて行う。
- 結果  
下記の通りしっかりとケースが展開されることが確認できた。  
<https://www.youtube.com/watch?v=GwgTufveM10>
- 考察  
光センサの閾値を適切に設定すれば本番でもケースの展開がうまく行われると考えられる。

### (V8) 分離試験

- 目的  
ケースが機体を分離できることを確認する
- 試験内容  
上空から機体に見立てたペットボトルを入れたケースを落とし、実際にペットボトルが分離されることを目視で確認する
- 結果  
以下の動画の通り、ペットボトルが分離できた。  
[https://youtu.be/qzmqd\\_67gRE](https://youtu.be/qzmqd_67gRE)
- 考察  
分離はケースの中身によらないケース側の問題なので、機体でも同様に分離ができると考えられる

### (V9) 手動飛行試験

- 目的  
手動操縦によって機体が飛行できることを確認すると同時に、機体地上局間通信ができることとモーターの振動に機体が耐えられることを確認する。
- 試験内容

機体地上局間通信モジュールをつけた状態で手動操縦し、機体地上局間通信ができることと振動に機体が耐えられることを確認する

- 結果  
以下の動画の通り  
[https://youtu.be/aMYhKZ\\_71ak](https://youtu.be/aMYhKZ_71ak)
- 考察  
機体地上局間通信ができることとモーターの振動に機体が耐えられることが確認できた。

## (V10) 自律飛行試験

- 目的  
機体が自律飛行できることを確認し、その上で空中でホバリングできることを確認する。加えて目的地に着陸してから適切な終了処理が行われていることを確認する。
- 試験内容  
RaspberryPiからFCに指示を出して自律飛行できることを確認するとともに、空中でホバリング及び着地、終了処理ができることを確認する。
- 結果  
<https://www.youtube.com/watch?v=VtzoF1AbPKM>  
あらかじめGPSを用いて指定した点を滑らかな曲線をつないで通っていく自律飛行を行った。最初の点においてホバリングを行い、最後着陸を行っている後終了処理を行っている。
- 考察  
本番においても自動で生成する点を通して目的地まで自律飛行で帰ってきて、目的地付近でホバリング・着陸できると考えられる。

## (V11) 目標物体検知試験

- 目的  
目標物体の機体に対する位置が推定できることを確認する。
- 試験内容  
ドローンで撮影した画像からカラーコーンが検知できることを確認する。
- 結果  
カメラで撮影した動画内で検出している様子を下の動画のように確認できた。  
<https://youtu.be/dPknP-CXkUU>
- 考察  
検出可能であることが確認できた。様々な環境下で学習させているため太陽の位置などで左右されることはないと考えている。

## (V12) 撮影試験

- 目的  
飛行中に地上の動画が撮影できることを確認する。
- 試験内容

- 飛行試験の際に地上の動画を撮影し、実際に撮影されたことを確認する。
- 結果
  - 機体に取り付けたカメラにより飛行中に動画の撮影ができたことを以下の動画のように確認できた。
  - <https://youtu.be/kwytvH4ZaAk>
- 考察
  - プロペラ回転開始から着地まで問題なく撮影できている。

## 第7章 工程管理, ガントチャート(スプレッドシートを推奨)

### (1) 概要

1日で大きな動きがあるため、直近数日の予定を都度組んでいる。ミーティングの毎日開催によってタスクを各自確認すると共に、Wikiによる作業日誌及びドキュメント作成により一人一人の動きが見える体制を取っている。尚、大まかなガントチャートは9.2.に置いた。

### (2) 主な予定

7/25	ACTS予備審査書提出締切
8/4	試験期間終了/活動再開
8/12	能代宇宙イベント本審査書提出締切
8/12, 13	東大投下試験
8/23	東大投下試験
8/28	ACTS本審査書提出締切
11/5-6	能代宇宙イベント
11/20-21	ACTS

詳細なガントチャートは[スプレッドシート](#)を参照。

### (3) 担当

構造, 回路, 制御内での役割分担について述べる。

人数は構造3人, 回路2人, 制御3人となっている。

- 構造
  - ケース, パラシュート
  - コア部, ケース
  - アーム
- 回路
  - (主に)ケース側回路

- (主に)機体側回路
- 制御
  - 全体設計, 詳細設計, 物体検知, プロポ操作
  - 地上局, ファームウェア, 詳細設計
  - シミュレータ, シーケンス設計, 詳細設計

## 第8章 大会結果報告

### 1. 目的

マルチコプターのCanSatでの0m達成

### 2. 結果

#### 2.1. 1回目の投下

機体ケース間を繋ぐ紐がアーム展開の引きばねに絡まり, アームが一つ展開せずに落下.

#### 2.2. 2回目の投下

機体の姿勢推定に狂いが生じ, 機体分離まで行くも落下.

### 3. 考察

今大会の失敗は主に環境由来の原因と運由来の原因とソフト由来の原因の二つに大別される.

まず, 環境由来の失敗について述べる. 我々はACTSの投下環境についての想定が甘く「場合によっては投下高度が30m以下となること」及び「投下タイミングが不明であること(一日目)」に対応しきれなかった. 30m以下となる可能性があるということで, 試したことはないが約25m程度でも何とか飛行が可能なアルゴリズムを急遽考えて実装した. そこに時間が取られ, 突如投下順が次であると伝えられたタイミングから急いで準備をし, 計測に向かうこととなったのだが, その際投下前の展開等の試験をしていなかった. その結果, 1回目の投下では展開が問題となり失敗した, 投下環境の想定が甘く, 準備に余裕が持てずに焦ってしまったことは一つの大きな原因であるといえるだろう. 尚, 2日目は余裕をもって準備をしたことで特に問題となることはなかった.

次に, 運由来の原因について. 上で焦って試験をせずに投下したことが展開を失敗した原因の一つであると述べたが, 正直なところ運が悪かったというのが大きい. 機体ケース間を繋ぐ紐が機体やケースのどこかに絡まってしまう問題は試験飛行の段階で考えついており, 対策もしていた. 実際当日も, 機体の展開試験をせずとも, 紐が絡まることのないよう機体をしまう際にチェックをした. また, これまで, 延べ100回以上は展開の試験をしてきたが一度も失敗したことがない. そんな中絡まったのは運が悪かったとしか言えない. 一応1回目の投下後に対策はしたが, 絡まりの再現性は0に近いだろう.

最後にソフト由来の失敗について. これまで述べた考察は主に1回目の投下に関するものである. ここで2回目の失敗の原因となったソフト由来の失敗について述べる. 結果の章で姿勢推定に狂いが生じた述べたが, 正確には綺麗に180度ずれたのである. テレメトリのデータを見ている制御担当者はこのことに気付かず投下してしまい, 結果的失敗となった. だが, そもそも180度ずれているにも関わらずフェールセーフが働かなかった(フェールセーフを組み込んでいなかった)ことが問題である. 我々はオープンソースのPX4を用いているが, これはCanSatのために作られているとは到底言い難い. 今回, CanSatの環境でも使えるようにローカライズしたつもりではいたが, 一から理解しているとは言



えない状況である。一から実装するとは言わずとも、PX4の細部まで理解した上でCanSat向けに完全にローカライズした上で開発を行えば、我々が直面してきた問題のいくつかや2回目の投下の問題を回避することができただろう。ただ、マルチコプターのCanSatはソースが少なく、開発も一筋縄では行かなかったことから、そこまでのキャパシティがなかったことは事実である。仕方ないで片付けたくはないが、今回に関しては仕方なかったのかもしれない。

## 第9章 まとめ

### 1. 工夫・努力した点 (ハード, ソフト, マネジメント面すべて)

#### 1.1. ACTSでの本審査書からの変更点

ハードとソフトの2つに分けて簡潔に説明する

##### 1.1.1. ハード

ケースに関しては、ケース展開の門を引き抜く動力をモータからパラシュートの開傘衝撃に変更し、モータを一つ減らした。これによってパッシブに展開されることとなったので、安定性が増した上、モータを回す時間の分ケース展開までの時間が短くなり、ケースと基板のさらなる小型化が達成された。

機体に関しては、機体が水平なまま自由落下した際の衝撃に耐えられるよう、最も強い衝撃がかかる最下段の素材をPLAからより強度のあるポリカーボネート板に変更し、上段にかかる応力を小さくするために各段の間の接続パーツの断面積を大きくした。これにより重量は増えたものの強度が大幅に上がり、ACTSの投下二回目では自由落下したにも関わらず無傷であった。

##### 1.1.2. ソフト

投下高度が30m前後ということで、安定性を捨て落下するまでに飛行に移れるようアルゴリズムを変更した。具体的にはキャリア放出の判定に光センサのみ用いることにし、気圧計から求まる高度及び落下速度を合わせて用いることをしないようにした。また、キャリアからの放出後、パッシブなシーケンスである、パラシュート展開、ケース展開、機体放出、機体姿勢安定までが機体分離のモータを回す時間(3秒程度)+0.5秒以内に収まると仮定して分離の指示を出すようにした。これはACTSの投下一回目では判断ができなかったが、投下二回目では機体分離まで成功しており、このアルゴリズムが問題ないことを確認した。

#### 1.2. マネジメント

我々のプロジェクトの特徴的な点として、以下の5つが挙げられる。

1. メンバー同士の親交があまりない状態からのスタート
2. メンバーの電子工作やプロジェクトに対する知見があまりない
3. 半年間のプロジェクトである
4. 資金集めが必要
5. コロナ禍である
6. マルチコプターは例がない

まず1と5について。同じ学科とはいえオンライン中心の講義ばかりでメンバー同士の親交があまりない状態からのスタートだったので、コミュニケーションが大変大事であった。序盤のアイデア出しやコンセプト決めはなるべく多くの人が大学に集まれる日に一気に行い、その他の報告は基本的にslack

と毎日夜に開催していたミーティングで行っていた。PMの負担を無視すれば一日で大きく事が動くこのプロジェクトにおいて毎日のミーティングは大変有効で、忙しいメンバーも週一回は参加できる形になっており、コミュニケーションを取る場所にもなった。また、実際に喋ることで文面上には表れないメンバーの状態などを知ることができた。尚、毎日のミーティングとは別に週一度全体ミーティングを行い、一週間で行われたことと今後の予定について確認していた。

次に2について。メンバーは工学部とはいえ航空宇宙工学科の学生であり、大学のシステム上専門科目が2年後期からであることも相まって、電子工作やプロジェクトに関する知見が少ない状態からの開始であった。それ故、タスクをメンバーに振ることができずにある程度経験のあるPMにタスクが集中してしまった。具体的にはルールや制約に関すること質問をする人、各物品の管理を担当する人などの担当を決めればよかったのだが、難しかった。尚、タスクが集中することで全体の見渡しが良くなったとも言える。PMとしては忙しくてプロジェクトに時間があまり割けないメンバーでも居心地が悪くならないような環境作り(忙しさに応じたタスク振りや、開発外タスクの割り当てなど)や、メンバーのモチベーション維持など、心理的、精神的なマネジメントを重視していた。特に、試験飛行は法律の関係上東京から千葉まで移動しなければならないため、重要である一方、時間がかかり疲れがたまるものであった。大会も何度か延期されて終わりが見えない中、累計30回以上の試験飛行を行ったが、最後までメンバーのモチベーションを維持できてよかった。恐らく、試験飛行でも大会でもミーティングでも楽しく明るい雰囲気を作り続けたことが功を奏したのだろう。

次に3の半年間のプロジェクトであるということについて。半年間というのは短いようで長く、長いようで短い。今思えばマルチコプターとCanSatに関してなんの知見もなしにマルチコプターのCanSat開発を始めたのは大分無謀であったと思う。ただ、世界で他に例がない、フライバックでマルチコプターを用い、さらに認識を用いて0mを達成を目指すということは我々を動かす大きな駆動力となったと考えている。

次に4の資金について。例年このプロジェクトには大学からある程度の援助があるのだが、昨年プロジェクトがなかったことで援助が例年の半分程度であった。これを受けて学内のクラウドファンディングやその他の支援を探し、最終的には例年以上の支援を受けて活動することができた。これは序盤に予算がもともとの援助を越すことを見越すことができたことが功を奏したと考えている。

最後に6について。マルチコプターはCanSatでは国内では例がない。それ故ルール内でマルチコプターに適さない箇所も多くあったため、マルチコプターにも適するようにルールや制約の変更をお願いした。また、マルチコプターなどのフライバックに共通していることであるが、航空法などの法律の規制がある。これを満たしながら試験飛行を行うのは難しかったが、先に述べたように何度でも試験飛行を行った。

### 1.3. ハード

機体とケースに分けて述べる。

#### 1.3.1. 機体

ひたすら作って試して改善してを繰り返した。特に最初はノウハウも何もなかったので既成品をCADに起こして3Dプリンタで造形し、回路やケーブルの配置も固定できう位置に固定する形で手動飛行させた。オリジナルから始めるのではなく、既成品の造形から始めたのは工夫点と言える。フライトコントローラの動作や地上局の動作に関しては探り探りであったものの、初期機体で手動飛行が成功してからは折り畳み機構を取り入れ、手動飛行で試した。折り畳み機構に関しては上向きに畳むか、下向きに畳むかをはじめとした議論がなされていたが、最終的には飛行中の揚力によって安定すること

や、折り畳んだ時の場所の利用効率がいいことから下向きの折り畳みにし、通常アームの上モータをつけるところをアームの下につける構造を取ることとなった。

手動飛行が成功した後、自動飛行も案外うまく行った。ただそこからが大変で、安定して上空から自動飛行できるようになるために、何度も機体の微小変更を繰り返した。また、マイコンの変更や回路の変更などに応じてその都度構造が変更された。上で折り畳み機構について少し取り上げたが、我々の機体は工夫点に溢れており、どの構造をとってもその構造の必要性を説明することができる。以下工夫点を列挙する。

- 折り畳まれたアームは引きばねによって展開するが、ケースでアームを抑え込む構造を取っているため、ケースが外れた瞬間に即座にアームが展開する構造になっている。
- 機体の展開検知を行う光センサをアームと機体の中心部の側面の間に配置し、アーム側に黒いスポンジを配置した。これにより、光センサはアームが閉じている時は真っ暗になり、アームが展開した瞬間に明るくなるので、その値の差が大きく判断が容易な上、ケース展開の後、最も早く展開するアーム展開の検知を最速で行うことができる。
- 中心部は必要なものに応じて4段構成となっており、テレメトリモジュールやGPS、Lipo、マイコンなどの配置も固定していたので各段それに応じた形状をしている。無駄なスペースもなく、最適な構造と言えるだろう。
- 最下段には四辺にスポンジを配置しており、落下時の衝撃を吸収する構造を取っている
- Lipoのために1段取っており、取りやすい位置に置いておくことで、試験飛行効率を向上させた
- ACTSでの本審査書からの変更点にて触れたが、高さ2,30mから自由落下しても問題ない程度の強度を持った構造となっている
- 重量の制限は厳しいものの、モータ&フライトコントローラとマイコンで電源を分けることで電池交換の度に電源が切れることがなくなった。というのもフライトコントローラとマイコン間は接続されていて互いに給電することができるため、片方の電池の交換中は他方から給電され、試験飛行効率が上がった。
- アームには軽量で丈夫なトラス構造を用いている。アームから落ちない限り、折れたことのない、信頼度の高い構造であった。
- 先も述べたが通常のマルチコプターとは異なり、モータをアームの下に配置した。これにより半径方向のスペースを節約することができた。これに伴い、プロペラを規定の方向に固定できなくなったが、テグスで工夫して固定した。
- プロペラには十分な揚力が得られるように7インチのものを採用し、それに合わせたアームの長さを取った。
- アームの下にケーブル用の窪みをつけることで何があってもプロペラがケーブルに当たらないようにした。尚、何度かプロペラがケーブルに当たってケーブルが切れたことで試験飛行が中断された経験からこのような構造を取った。
- 造形には3Dプリンタとレーザーカッターを用いており、構造の再現性は高い。また、機体の中心部はパーツごとに分けることで、造形時間を短縮し、効率化を図った。
- 機体最上段にあるフライトコントローラには上にカバーをつけており、ひっくり返って落ちててもフライトコントローラに被害がないようにしている。実際、能代の大会では沼の中に突っ込んだものの、カバーのおかげでフライトコントローラは無事であった

### 1.3.2. ケース

ケースは機体の収納と機体の展開及び分離タイミングの制御、パラシュートの取り付けを目的としている。ケースは可能な限り機能を減らし、サイズと重量の削減に努めた。その結果、ケースは最初に

作成したのから大幅にサイズ重量共に削減され、最終的に100g軽くなった。尚、軽くなった分はマイコンの変更や機体の強度の強化に用いられた。

ケースの展開(=機体の展開)と機体の分離には共に門構造を用いた。門構造はモーター一つで済む上、安定しているため、信頼度の高い構造である。最終的には先に述べた通り、ケース展開の門構造ではモータではなくパラシュートの開傘衝撃を用いた。これによりモーターつ分の基板及びケースサイズを削減できた上、開傘衝撃がかかるタイミングは展開にも適したタイミングであることから、最適なタイミングでの展開が可能となった。

展開の検知には光センサを用いており、アームの場合と同様ケースの光センサが機体の最下段の黒いスポンジに当たるように配置することで、ケースが展開して機体が展開するタイミングで光センサの値が大きく変化するようにしていた。機体分離の門構造では機体が分離するタイミングを制御するためにモータを用いた。尚、分離のタイミングの判定には気圧センサから求まる相対高度を用いた。

パラシュートに関しては、天井部に穴を開けることで安定化し、理論に基づいた設計を行った。最初に作成したものでは理論通りに十分に機体が減速できなかったことから、機体がしっかり減速できるよう大型化したものを作成した。その予備としてそれより少し小さいものを作成していたのだが、これのおかげでACTSの重量オーバーが解決した。

## 1.4. ソフト

フライトコントローラ、マイコン及びコード、認識の3つに分けて述べる。

### 1.4.1. フライトコントローラ

フライトコントローラとしてPixhawkを用いた。Pixhawkは高価で入手も難しいことから、我々は先代が用いていたPixhawkのv2.4.8をそのまま用い、Pixhawk4は用いなかった。ファームウェアとしてはpx4とardupilotの二つがあるが、より新しいpx4を用いた。地上局にはQGCとMission Plannerの二つがあるがテレメトリの周波数変更においてのみMission Plannerを用い、その他全ての場面でQGCを用いていた。そもそも我々はこれらの用語を知らない状態からのスタートだったので、その理解に大変苦労した。

px4には多くのパラメタがあり、基本的にはそれらを用いることを想定されているが、最終的にはCanSatでも問題なく動くよう、パラメタの範囲を広げたり、フェールセーフをCanSat向けに変えるなどのファームウェアのローカライズを行った。また、実機ができる前や試験飛行が難しいときにはGazeboなどのシミュレータを用いて試験飛行を行っていた。尚、実際の試験飛行の際には常に地上局を起動してマルチコプターと通信ができるようにしており、何かあった際には緊急停止できるようにしていた。ログはpx4で取っており、試験飛行をまとめて行った後にビジュアライズして確認するようにしていた。

躓いたところは、フライトコントローラとマイコンの通信、テレメトリの周波数の変更、px4のパラメタ理解などである。

### 1.4.2. マイコン及びコード

学科のプロジェクトでマイコンとしてArduino unoを用いた経験があったことから、最初はArduino nanoを用いていた。動作に関しても並列化を行うなどしていたが、メモリが少ないなどの理由から手動飛行ができるようになるタイミングでRaspberry Pi zeroに移行した。

コードには学科唯一のソフトの講義がPythonであったことと認識を用いたことからPythonを用いることにした。ROS2を用いることも検討したが、Raspberry Pi zeroでの環境構築がハード上厳しいことから諦め、フレームワークを自作した。その後、認識の周波数を上げるためにRaspberryPi4に移行したが、ROS2への移植が手間だったのでそのまま自作フレームワークを用いた。対話型で入力できるよう

にしていたため、試験飛行の効率が上がった。また、マイコン上の仮想端末上で実行しているため、もし途中で通信が切れたとしてもログが消えないようにしていた。

工夫した点としては、飛行ログをフライトコントローラとラズパイの二重で取り冗長化していたこと、通信の手間や問題を回避するためにケースと機体の間で通信をせずに独立して制御したこと、機体の分離指示を相対高度と光センサのみで確実に行えるようにしたこと、各値には適切にカルマンフィルタを加えて飛び値を排除したことなどがあげられる。

### 1.4.3. 認識

認識を用いて0mを目指すことになったのは大分終盤である。ケースの軽量化に伴い、RaspberryPi4が重量的に載ることがわかり、RaspberryPi4を用いた認識を行うことにした。

物体検知をする方法としてOpenCVを用いた画像処理を行う方法とAIを用いる方法があったが、環境依存性を無くすために後者を取ることにした。物体検知アルゴリズムとしては開発時にはSSD Mobilenet v3やyolov5(現在はyolox)が最新で同じ程度の性能であったが、yolov5の方が学習がうまくいきやすかったことからこちらを用いることにした。

様々な角度や環境にカラーコーン(カラーシート)をおいて動画を撮影し、それをフレームごとにアノテーションすることで学習させた。尚、少ない枚数で少し学習させたものを用いてアノテーションを自動化し、効率化を図った。この時点で使える程度には認識は高速であったが、整数量子化することで3fpsを実現した。Edge TPUの利用も考えたが、そもそも重量の問題上難しく、3fpsで十分であることからこれを用いることにした。

画像からカラーコーン(カラーシート)の位置を認識した後、その中心位置とそれを撮影した時の機体の姿勢と機体底面についているLiDARの値からカメラの画角などの情報を用いて、カラーコーン(カラーシート)の機体に対する相対座標を求め、目標位置とした。これを繰り返すことで目標座標を修正しながら降下し、0m達成が可能となった。

## 2. 課題点

ソフトについてはフライトコントローラ、特にそのファームウェアの理解が必要である。そもそもファームウェアは当たり前であるが、屋外地上飛行を想定して作られている。今回ファームウェアをCanSat向けにローカライズしたものの、フェールセーフなどCanSat独自のものを追加した方がよかったと後から感じた。我々はフライトコントローラとの通信に既存のライブラリを用いたが、これも未実装の部分があり、ファームウェアから完全に理解する必要があるのではないかと感じている。だが、半年間のプロジェクトでは厳しいため今回はベストを尽くしたとっていいだろう。

ハードについては最終的にできあがったもので強度、形状ともに申し分ないのではないと思う。強いて言えば機体とケースの穴抜きなどによるさらなる軽量化ができるだろう。また、フライトコントローラにつけたカバーをGPSにもつけて良かったのではないと思う。

試験飛行については法律上の問題があるとはいえ、移動も長く、確実な成果が得られるわけでもないのに、繰り返すと精神的に厳しくなってくる。これを解決する方法としてファームウェアを屋内でも(GPSが無くとも)使えるように修正することと、シミュレータ上でCanSatの状況を再現することが考えられる。だがこちらも半年間のプロジェクトでは厳しかった。

### 3. 今後の展望

ARLISSを見据えた時に問題となるのは消費電力とテレメトリであろう。今のままのLipoの容量では足りないことから消費電力を減らすかLipoの容量を増やすかをする必要がある。また、テレメトリは今回の大会で十分なものしか用いていないので、ARLISSで用いる場合には数kmの通信が可能であるLoRaを用いる予定であった。

以下、今後マルチコプターのCanSatを開発するチームがいるのであれば注意する必要があることについて述べる。マルチコプターの製作は案外なんとかなり、フライトコントローラであるPixhawkを用いれば手動飛行まではなんとかなる。

だが、ソフトに関しては自動飛行させるタイミングでフライトコントローラとマイコンをどのように通信させるか決める必要がある。おすすめは、我々は用いていないがRaspberryPi4でROSを用い、通信のライブラリとしてMAVROSを用いる方法である。これによりメッセージなど自作できるのである程度の柔軟性が確保できる。その後一通りpx4を理解した後に、CanSatに必要なフェールセーフをpx4の中に組み込むのがいいのではないか。また、課題点でも述べたように、px4は屋外向けであるので、これを屋内でも使えるように、正確にはGPSが無くとも制御できるようにファームウェアを修正できると試験飛行が大分楽になるのではないかと思う。また、それが難しくとも、Gazeboなどのシミュレータ上でCanSatの状況を再現できれば試験飛行に行かずとも実現可能性を試すことができるだろう。

ハードに関しては今回は素材としてPLAを用いていたが、強度計算をした上で特に荷重がかかる場所にはCFRPを用いるのも一つの手である。そうすることで壊れることがなくなり、試験飛行の効率もあがるのではないか。

その他のことに関してだが、マルチコプターは今年航空法がさらに厳しくなることで試験飛行が難しくなると思われる。法律を常に気にかけて試験飛行を行わなければいけないことに注意が必要である。また、試験飛行中は常に機体を見張っておき、いざという時には緊急停止させられるようにすべきである。我々はテレメトリ経由とマイコン経由のどちらからでも緊急停止ができるようにしており、常に2人は緊急停止ができる状態で試験飛行を行っていた。

最後に、マルチコプターのCanSatは認識を用いれば0mを達成することができる。実際に大会で示すことはできなかったが、可能であることは示せたと思うので、是非安全に注意しながら開発してみたい。