

ARLISS 大会報告・技術詳細報告書

提出日：2019年 10月 15日

文責：山本悠介

● チーム情報

CanSat チーム名	鹿児島大学 100kinSAT
CanSat チーム 代表者	山本 悠介
UNISEC 団体名	東京工科大学 鹿児島分室 チーム名「100kinSAT」
UNISEC 団体 学生代表	山本 悠介
責任教員	服部 聖彦
CanSat クラス	350 ml Class

● メンバー

役割	名前（学年）
回路・制御	山本悠介（M2）
設計・加工	須藤順平（M2）
試験設備設計（ペットボトルロケット）	入江将崇（B3）
回路・制御	梶浦梨央（B3）
回路・制御	濱田祐介（B3）

● CanSat の製作目的・大会参加理由

私たちのチームは小型・軽量化、複数機投下を目指して CanSat を開発した。大会参加理由は、大型のローバが多い ARLISS において、私たちの小型 CanSat がミッション（ランバック）をどれだけ達成できるか挑戦したかったからである。

目次

第1章	ミッションについて	6
1.	ミッションの意義と目的	6
2.	ミッションシーケンス	7
第2章	サクセスクライテリア	13
第3章	要求項目の設定（各チームごとに項目を追加してください）	14
1.	システム要求（安全確保のために満たすべき要求）	14
2.	ミッション要求（ミッションを実現するためのシステム要求）	15
第4章	システム仕様	16
1.	CanSat 設計図（CAD 図面や回路レイアウト図や写真など、公開可能であれば図面をここに示し、データも添付をお願いします。）	16
2.	CanSat 外観/質量/サイズ	17
3.	CanSat 内観・機構/電力	19
4.	使用部品	23
➤	電子系	23
➤	動力系	24
➤	構造系	24
5.	製作時に使用した機材・サービス	24
6.	プログラム・アルゴリズム	24
7.	会計	25
第5章	試験項目設定（項目別試験、結合試験、EndtoEnd 試験）	27
第6章	実施試験内容	29
1.	質量試験 【済】	29
➤	目的	29
➤	試験/解析内容	29
➤	結果	29
2.	キャリア収納試験 【済】	30
➤	目的	30
➤	試験/解析内容	30
➤	結果	30
3.	ロスト対策試験 【済】	32
➤	目的	32
➤	試験/解析内容	32
➤	結果	32
4.	落下試験 【済】	35
➤	目的	35
➤	試験/解析内容	35
➤	結果	35
➤	結論	36

5.	分離・展開試験 【済】	37
➤	目的	37
➤	試験/解析内容	37
➤	結果	37
➤	結論	37
6.	静荷重試験 【済】	38
➤	目的	38
➤	試験/解析内容	38
➤	結果	38
7.	振動試験 【済】	41
➤	目的	41
➤	試験/解析内容	41
➤	結果	41
8.	放出時衝撃試験 【済】	43
➤	目的	43
➤	試験/解析内容（1）	43
➤	結果（1）	43
➤	試験/解析内容（2）	43
➤	結果（2）	44
9.	通信試験 【済】	46
➤	目的	46
➤	試験/解析内容	46
➤	結果	46
10.	通信チャンネル変更試験 【済】	48
➤	目的	48
➤	試験/解析内容	48
➤	結果	48
11.	End-to-End 試験 【済】	49
➤	目的	49
➤	試験/解析内容	49
➤	結果	49
12.	走行試験 【済】	51
➤	目的	51
➤	試験/解析内容	51
➤	結果	51
13.	位置情報試験 【済】	52
➤	目的	52
➤	試験/解析内容	52
➤	結果	52

14.	自律制御走行試験 【済】	55
➤	目的	55
➤	試験/解析内容	55
➤	結果	55
15.	横転復帰試験 【済】	56
➤	目的	56
➤	試験/解析内容	56
➤	結果	56
16.	制御履歴記録試験 【済】	57
➤	目的	57
➤	試験/解析内容	57
➤	結果	57
17.	制御履歴提出試験 【済】	58
➤	目的	58
➤	試験/解析内容	58
➤	結果	58
18.	通信機器電源 ON/OFF 試験 【済】	62
➤	目的	62
➤	試験/解析内容	62
➤	結果	62
19.	ゴール判定試験 【済】	63
➤	目的	63
➤	試験/解析内容	63
➤	結果	63
20.	電力試験 【済】	64
➤	目的	64
➤	試験/解析内容	64
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）	67
1.	チーム内・審査会等	67
2.	各担当（ハード・ソフト・全体などの進行状況・予定を記入）	67
第8章	大会結果	68
1.	能代宇宙イベント	68
2.	ARLISS（上記能代と同様に記載してください）	68
➤	目的	68
➤	結果	68
➤	取得データ	69
➤	故障原因解析・解決手段等	80
第9章	まとめ	83
1.	工夫・努力した点（ハード、ソフト、マネジメント面すべて）	83

2. 良かった点・課題点	84
3. チームのマネジメント等、プロジェクト全体での良かった点、反省点	85

第1章 ミッションについて

1. ミッションの意義と目的

私たちのチームは 350ml クラスにおいて、「超小型 CanSat を 2 機搭載してランバックミッション」を行う。

近年の CanSat 競技は、0m ゴールを達成するチームが増えてきており、ただゴールするだけでなく、複雑なミッションをこなす機体が多くなってきている。それに伴って CanSat も大型かつ構造が複雑で高重量な機体が増えてきたため、ARLISS では 350ml クラスでの参加自体が少ないといえる。

しかしながら、CanSat は宇宙機を目指している以上、システムの簡素化や軽量化といった事項も極めて重要な要素となる。そこで私たちは、100 円ショップで購入できるような低コストな（簡素なシステム・軽量・低価格など）CanSat をコンセプトとした「100 均 SAT」を開発する。

100 均 SAT の最も特徴的な部分は、メジャーを使用した展開車輪である。超小型 CanSat は小型であるが故に車輪径が小さくなり、大型の CanSat に比べて走破性能が劣る。しかし、100 均 SAT はメジャーの展開車輪を備えることで収納時の小型化、軽量化を実現しており、展開時は Open Class にも引けを取らない車輪径を確保することで走破性能を獲得している。そのような展開車輪に使われている素材は 100 円ショップで購入したメジャー である。

また、低コストである利点を活かして複数機製作・搭載する。そうすることで、複数機体によるランバックミッションを行うことができ、ミッション成功確率の向上が期待できる。

2. ミッションシーケンス

私たちは 350ml クラスにおいて、同じシステムを有した 2 機の CanSat を搭載し、ランバックミッションを行う。図 1-1 にローバ 1 機のシステム構成図、図 1-2 に競技の流れを示す。

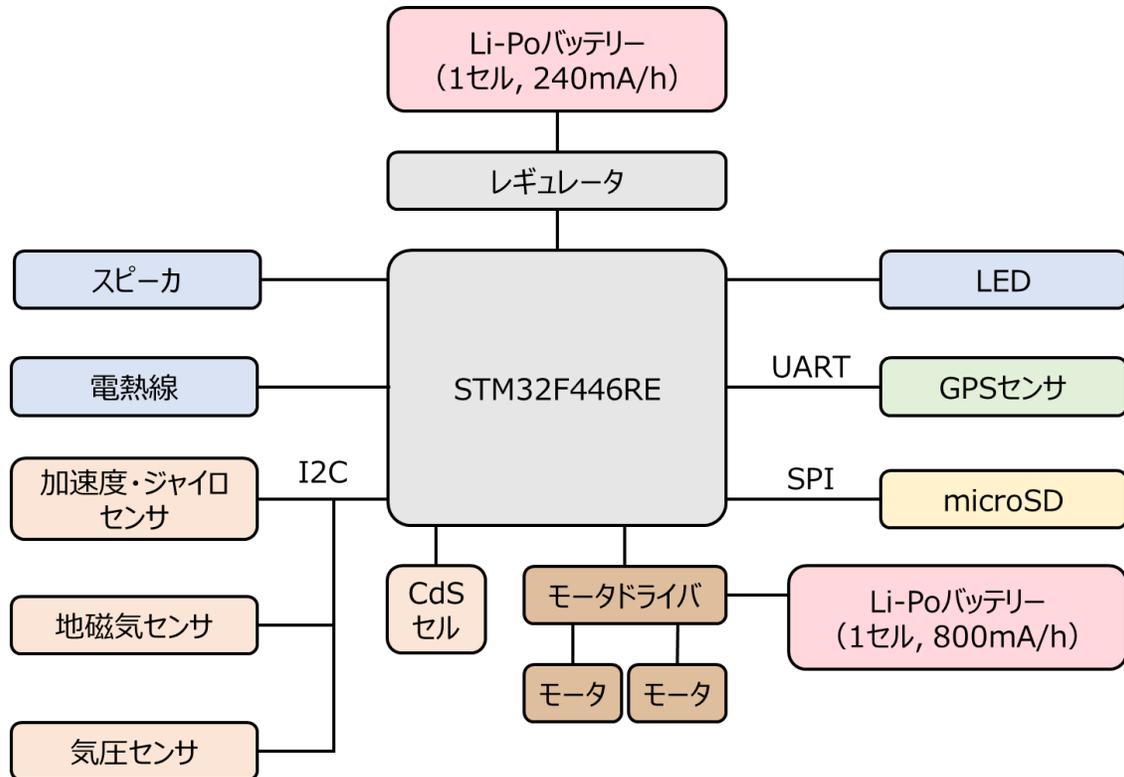


図 1-1 ローバのシステム構成

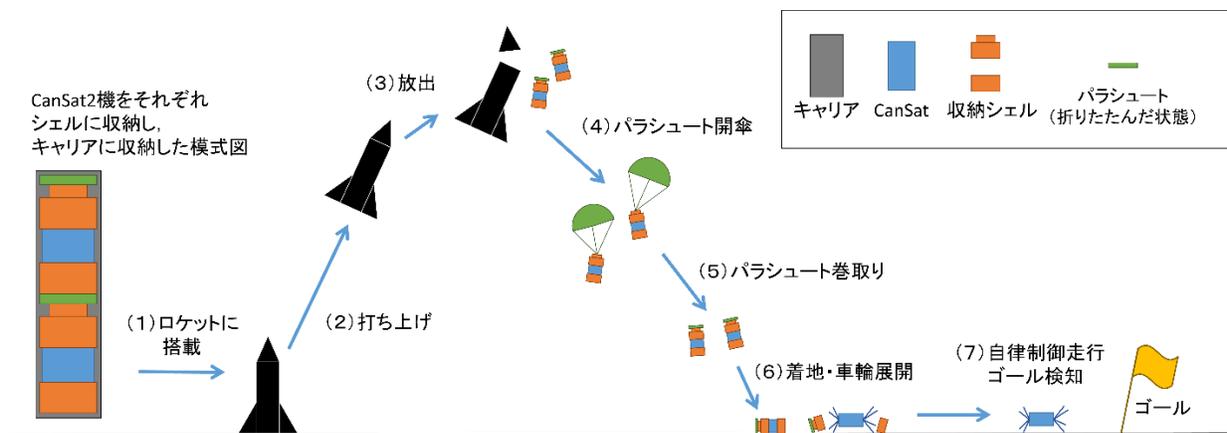


図 1-2 競技の流れ

図 1-3 にミッションシーケンス図を示す。

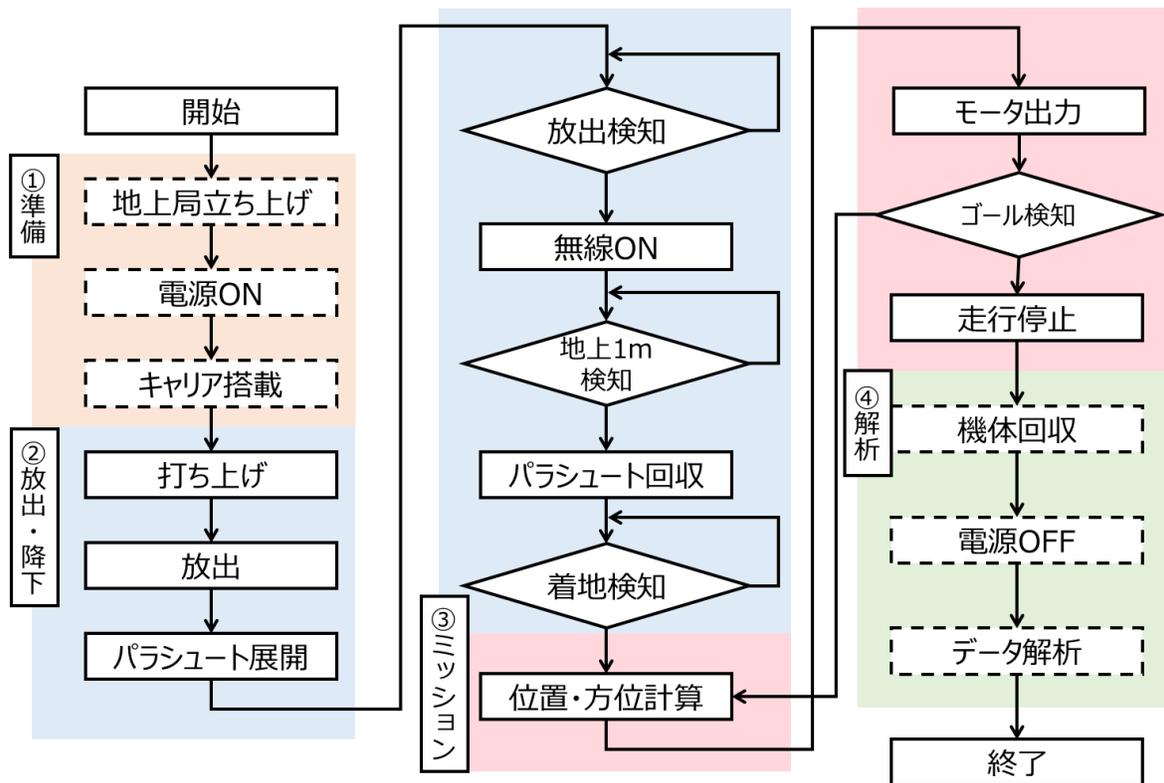


図 1-3 ミッションシーケンス

ミッションシーケンスの各段階について説明する。

1. 準備

準備段階では地上局を立ち上げ、CanSat の電源を ON にした後、キャリアに収納する。キャリア収納後は照度センサによる放出判定に移行する。電源 ON 直後（キャリア収納前）に照度センサによる判定が行われてしまわないように、CanSat の電源を ON にしてからキャリアに収納するまでに必要な待機時間を適切に設定する。

2. 放出・降下

キャリアに収納されている CanSat は、放出された際の周囲の明暗変化を認識することで放出を検知する。明暗変化の検知には照度センサ（CdS セル、MI527）を用いる。放出後、パラシュートが開傘することで減速する。

降下時には気圧センサ（LPS25H）から所得した値をもとに高度を算出する。地上から 1m の高さに到達した時点でパラシュート巻取り機構（図 1-4）によってパラシュートの巻き取りを開始する。巻取り機構はゼンマイを使った機構となっており、メジャーの巻き取りをイメージしていただくとわかりやすい。詳しくは第 4 章で説明する。この機構によって

パラシュートを回収することで、走行開始時に CanSat がパラシュートに絡まって走行不能になる事態を回避する。

パラシュート巻取り機構によるパラシュートの巻取りは、風の状況や落下時の姿勢によって機能しないこともあるが、巻取り機構がない場合と比較して、ローバがパラシュートに絡まってしまい走行不能になるという事態は減少した。

着地の検知は加速度センサ (LSM6DS33) によって行う。降下時は機体が風に吹かれて大きく揺れるため、加速度センサの変化量が大きい。一方で、着地後は CanSat が地面で静止に近い状態となるため、加速度センサの変化量が小さい。この違いを利用して着地を検知する (図 1-5, 種子島ロケットコンテスト 2019 におけるログ)。

着地検知後はニクロム線によって固定テグスを焼き切り、CanSat 本体を収納しているシェル (図 1-6) から脱出する。固定テグスは打ち上げや放出の衝撃で切れてしまうことのないように、固定テグスにそれらの衝撃が直接伝わらないように工夫する。



図 1-4 パラシュート巻取り機構

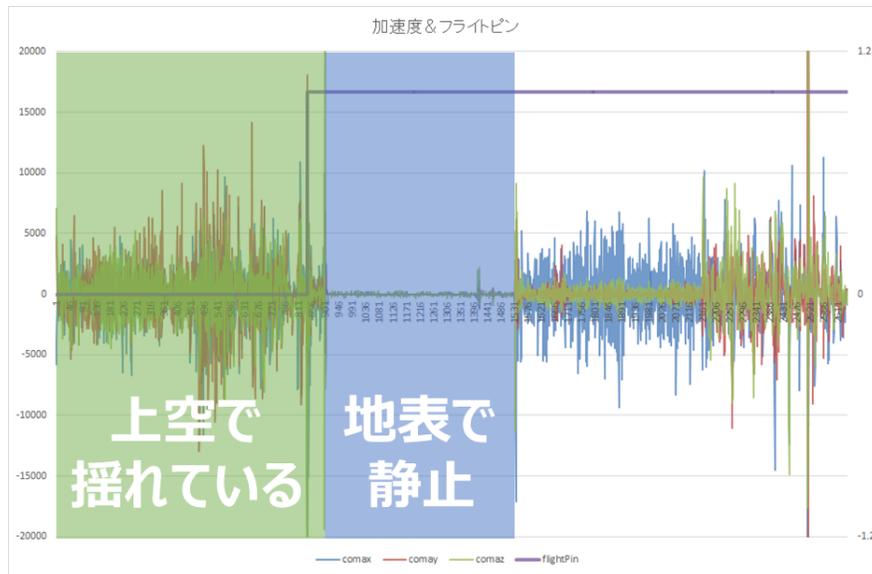


図 1-5 加速度センサから所得した値



図 1-6 CanSat 収納シェルとパラシュート

3. ミッション

CanSat 本体が収納シェルから脱出した後、ランバックミッションを開始する。CanSat は、GPS センサ (GYSFDMAXB) から所得した緯度・経度の情報をもとに現在位置を求める。また、競技開始前にあらかじめインプットしていたゴールの緯度・経度と合わせることで進むべき方位と距離を算出する。次に地磁気センサ (LIS3MDL) から得られた値をもとにして、CanSat がどの方位を向いているか求めて、進むべき方位に向くように旋回を行う。旋回を行った後、前進しゴールへ向かう。

走行路面の状況によってはわだちなどの影響で正確に前進できないため、適切に地磁気センサによる現在方位をフィードバックして旋回行動をとり、CanSat の向きを修正する。

ゴールとの距離が 10m 以内となったらゴール判定を行い停止する。

4. 解析

CanSat は、競技時に所得したデータ類を microSD カードに適宜記録する。データの保存形式は csv ファイルとし、Excel で表やグラフを製作しやすいフォーマットにする (図 1-7、種子島ロケットコンテスト 2019 におけるログ)。

解析段階では、CanSat から microSD カードを回収して記録されたデータを解析する。解析する項目を以下に示す。

- ① CanSat の電源を ON にしてから放出検知モードに遷移したか
- ② 照度センサによって放出を検知したか
- ③ 気圧センサによって高度を記録できたか
- ④ 気圧センサの値で判定を行い、パラシュート巻取り機構の固定テグスを焼き切るために、ニクロム線に出力信号を送れたか
- ⑤ 加速度センサの値で着地判定ができたか
- ⑥ 収納シェルの固定テグスを焼き切るために、ニクロム線に出力信号を送れたか
- ⑦ GPS センサから緯度・経度の値を所得できたか
- ⑧ 地磁気センサから CanSat の向きを求めることができたか
- ⑨ GPS センサと地磁気センサから求めた値をもとにモータへ出力できたか
- ⑩ CanSat がどのような経路をたどってゴールへ向かったか

	A	B	C	D	E	F	G	H	I	J	K	L
1	millis	year	month	day	hour	minute	second	state	lat	lng	alt	ax
5277	680235	2019	3	8	0	32	24	1	30.37417	130.9597	-6.5	-5972
5278	680362	2019	3	8	0	32	24	1	30.37417	130.9597	-6.5	-5919
5279	680488	2019	3	8	0	32	24	1	30.37417	130.9597	-6.5	-5945
5280	680620	2019	3	8	0	32	24	1	30.37417	130.9597	-6.5	-6014
5281	680754	2019	3	8	0	32	25	1	30.37417	130.9597	-6.5	-5972
5282	680883	2019	3	8	0	32	25	1	30.37417	130.9597	-6.5	-5870
5283	681012	2019	3	8	0	32	25	1	30.37417	130.9597	-6.5	-5958
5284	681138	2019	3	8	0	32	25	1	30.37417	130.9597	-6.5	-5980
5285	681269	2019	3	8	0	32	25	1	30.37417	130.9597	-6.5	-5936
5286	681396	2019	3	8	0	32	25	1	30.37417	130.9597	-6.5	-5991
5287	681523	2019	3	8	0	32	25	1	30.37417	130.9597	-6.5	-5968
5288	681654	2019	3	8	0	32	25	1	30.37417	130.9597	-6.5	-5932
5289	681781	2019	3	8	0	32	26	1	30.37417	130.9596	-6.5	-6011
5290	681910	2019	3	8	0	32	26	1	30.37417	130.9596	-6.5	-5936
5291	682040	2019	3	8	0	32	26	1	30.37417	130.9596	-6.5	-5956
5292	682167	2019	3	8	0	32	26	1	30.37417	130.9596	-6.5	-5984
5293	682295	2019	3	8	0	32	26	1	30.37417	130.9596	-6.5	-5944
5294	682424	2019	3	8	0	32	26	1	30.37417	130.9596	-6.5	-5948
5295	682552	2019	3	8	0	32	26	1	30.37417	130.9596	-6.5	-5932
5296	682690	2019	3	8	0	32	26	1	30.37417	130.9596	-6.5	-5974
5297	682821	2019	3	8	0	32	27	1	30.37417	130.9596	-6.6	-5967
5298	682947	2019	3	8	0	32	27	1	30.37417	130.9596	-6.6	-5975
5299	683078	2019	3	8	0	32	27	1	30.37417	130.9596	-6.6	-5855

図 1-7 ログの記録例

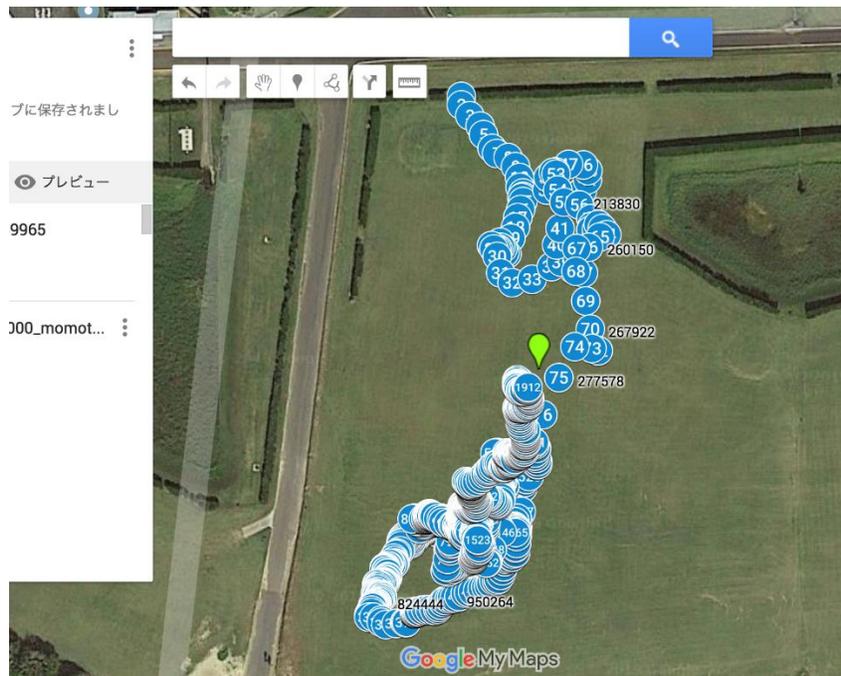


図 1-8 走行経路のプロット

第2章 サクセスクライテリア

ミニマムサクセス	2機のローバがシステムを正常に作動しながら軟着陸する →ロケットから放出後のダウンリンクしたログで判断
ミドルサクセス	2機のローバがパラシュートと絡まることなく分離し、走行開始する →目視で確認
フルサクセス	2機のローバが2 km以上走行する これによって、350mlにおける2機ローバによるランバックシステムの構築ができたことを評価する
アドバンスドサクセス	どちらか一方の機体がゴールの10m以内にたどり着く

第3章 要求項目の設定（各チームごとに項目を追加してください）

1. システム要求（安全確保のために満たすべき要求）

要求番号	自己審査項目（ ARLISS 打ち上げ安全基準 ）
M1	質量と容積がレギュレーションを満たすことが確認できている
M2	ロスト対策を実施しており、有効性を試験していること
M3	減速機構を有しており、危険のない速度で落下することが試験できていること
M4	打ち上げ時の準静的荷重で、安全基準を損なわれないことが試験で確認できる
M5	打ち上げ時の振動荷重で、安全基準を損なわれないことが試験で確認できる
M6	分離衝撃荷重によって、安全基準を損なわれないことが試験で確認できる
M7	打ち上げ時の無線機の基準を満たすことが確認できること
M8	無線のチャンネル調整に応じることができること
M9	M1-M8 の基準を満たした設計の機体で End-to-End 試験を実施し、実施後に安全に関わる大幅な設計変更を行わないこと

2. ミッション要求（ミッションを実現するためのシステム要求）

要求 番号	自己審査項目（ミッション実現要求項目）
S1	通信機器の電源を ON/OFF にできること、また、その確認ができること
S2	打ち上げから着地までの衝撃や走行などの衝撃で部品が外れないこと
S3	落下時や展開時に 2 機が絡まらないこと
S4	放出・着陸判定ができること
S5	ARLISS 会場の環境（風など）でも電熱線で固定テグスを切り、機体の展開ができる
S6	パラシュートが機体から不備なく分離できること
S7	展開時や走行時に機体が横転しても復帰できること
S8	任意の方向に前進・旋回などの走行が可能であること
S9	走破性を持ち、ARLISS 会場の環境においてスタックしない
S10	位置座標データが所得できること
S11	ミッション時に自律制御を実施することが確認できる
S12	制御履歴を記録・提出する準備ができている
S13	ゴール地点まで走行可能なバッテリー容量を有する
S14	誤差半径 10m 以内の位置情報が所得できる
S15	CanSat がゴールを検知でき、ゴールしたことを外部から確認できること

第4章 システム仕様

1. CanSat 設計図 (CAD 図面や回路レイアウト図や写真など、公開可能であれば図面をここに示し、データも添付をお願いします。)

本体とシェルの設計は Fusion360 で行った。本体とシェルの設計図をそれぞれ図 4-1 と図 4-2 に示す。また、Fusion360 のデータ共有機能で CAD 設計データを公開している (設計者のライセンス切れが起こるとみられなくなる可能性あり?)。ただし、完成版のアセンブリデータは製作していないため、最も完成状態に近いアセンブリデータとなる。

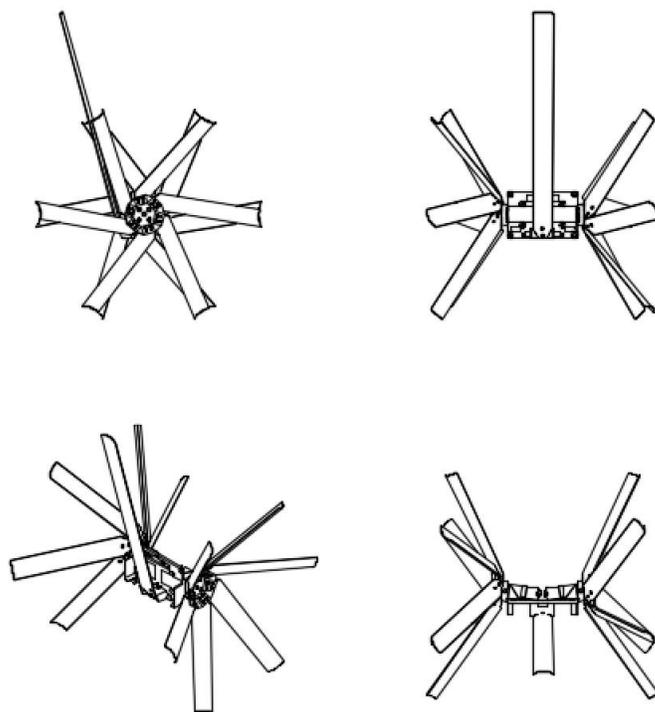


図 4-1 本体の設計図

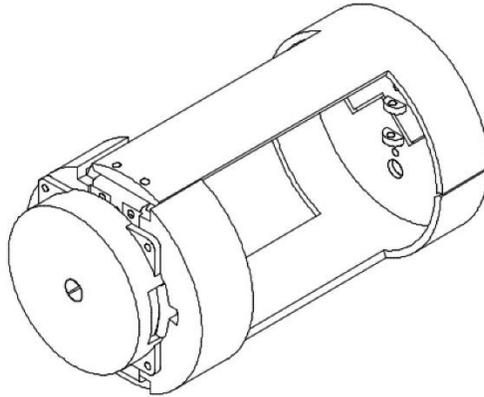


図 4-2 シェルの設計図

項目	リンク先
本体	https://a360.co/2O3faqo
シェル	https://a360.co/2YEQ1TR

CanSat の回路・プログラムを GitHub 上で公開している。

項目	リンク先
回路図	https://github.com/ymt117/ARLISS2019/tree/master/circuit
ガーバデータ	https://github.com/ymt117/ARLISS2019/tree/master/Gerver
プログラム	https://github.com/ymt117/ARLISS2019/tree/master/src

2. CanSat 外観/質量/サイズ

(図あるいは写真を用いる。大まかな質量と、本体のほかに、パラシュートや分離機構などを含めて、サイズ内にどのように収納しているのか、断面図などのイメージ図を掲載)

今回のミッションでは 350ml クラスにおいて 2 機の CanSat を搭載するが、2 機の機能に差異はないため 1 機について説明する。

図 4-3 に CanSat 外観図を示す。CanSat は車輪とスタビライザーにメジャーを用いた。スタビライザーは根元部分がメジャーを 2 本重ねた構造で折れ曲がりづらくなっており、走行時スタビライザーにかかる負荷に耐えられるようになっている。モータハブと本体構造は 3D プリンタで出力した。3D プリンタの素材には PLA を用いた。モータハブとメジャー（車輪）の固定は M2 のプラねじとプラナット、スプリングワッシャーで固定している。車輪 1 組にねじ、ナット、スプリングワッシャーを 24 セット用いるため、軽量化を目的に予備審査の段階で金

属だったねじとナットの素材をプラスチックのものに変更した（振動試験等で強度に問題のないことを確認済み）。本体構造とスタビライザーの固定はより強固に固定するため金属ねじを使用している。

車輪は展開時に直径およそ 165 mm となる（図 4-3, D）。一方で、収納時には直径およそ 60 mm となる（図 4-4, B）。収納時の直径は、350ml クラスのレギュレーションを満たしている。また収納時の高さは 1 機分がおよそ 110 mm となり（図 4-4, A）、2 機搭載してもレギュレーションを満たす。

また、機体の大きさに対して車輪径が大きいいため、予備審査の段階では横転すると復帰することが困難であった。これを解決するためにバッテリーの固定位置を変更し、車輪の回転軸から重心を遠ざけ、偏心して回転することを利用して復帰できるようになっている（詳しくは横転復帰試験を参照）。

リンク先 (https://youtu.be/m3gTD_DoJeQ) に走行テストの様子を投稿している。

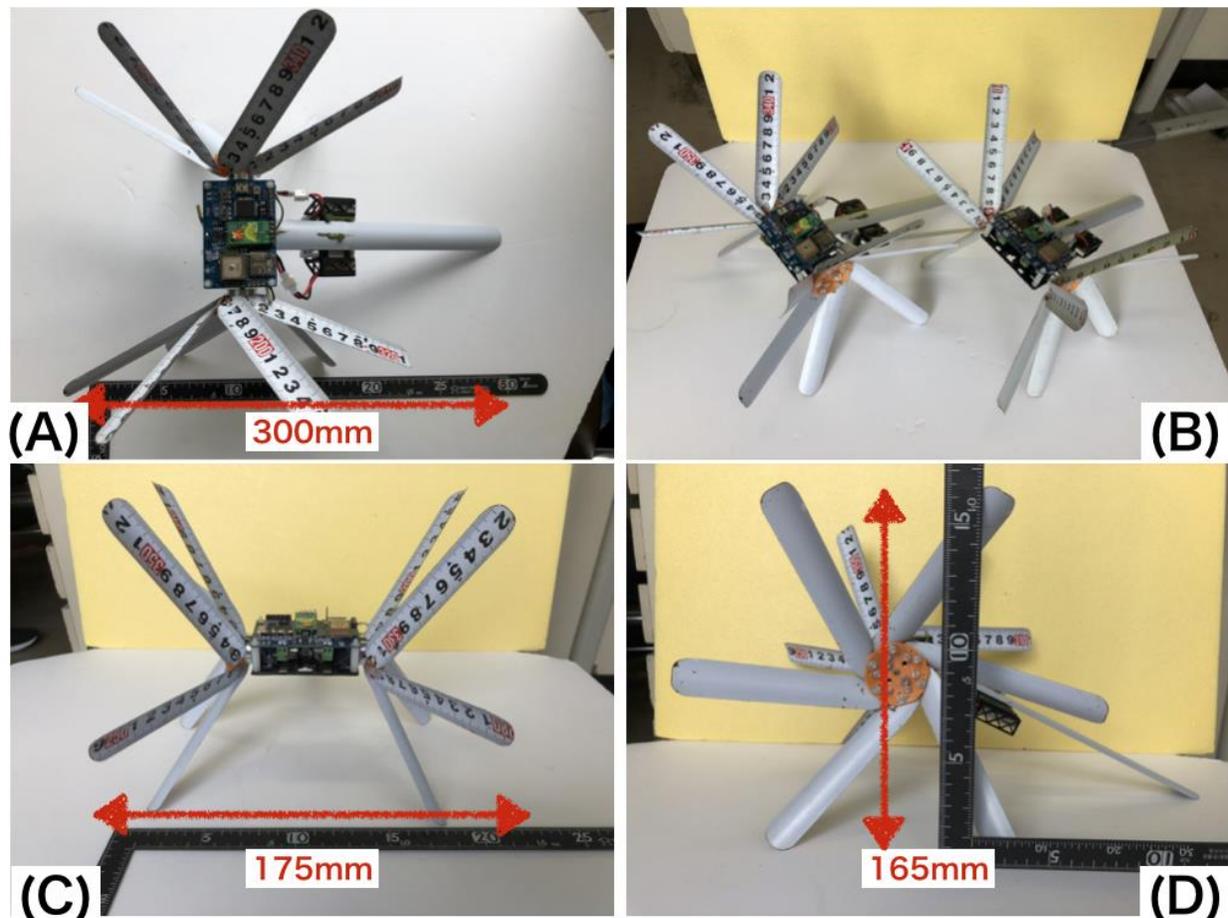


図 4-3 CanSat 外観図（展開時）



図 4-4 CanSat 外観図 (収納時)

3. CanSat 内観・機構/電力

(図や動画を用いて機体の基本的な機能や仕組み、パラシュートの展開・分離/翼の展開などの仕組みを説明、一回の打ち上げで使用する電力とバッテリー要領の計算結果などを記載)

A) 機構

➤ 展開車輪

コンベックスメジャーの復元力を生かした展開車輪となっている。

メジャーには折れ曲がりやすい方向と折れ曲がりづらい方向があるため、ハブの形状を工夫し CanSat が前進する方向に車輪が回転する時、メジャーが折れ曲がりづらい方向に力を受けるようになっている (図 4-5)。

収納時には折り曲がりやすい方向に折りたたむことで、350 mmクラスのレギュレーションを満たす直径まで収納することができる。

➤ パラシュート巻取り機構

展開車輪の弱点はパラシュートの紐を巻き取りやすいことだった、

それを改善するためパラシュート巻取り機構を制作した。図 4-6 はパラシュート巻取り機構の内部構造を示した写真である。

メジャーの巻き取られる機構に使われているゼンマイを取り出し、3Dプリンタで制作した容器に入れることでパラシュート巻取り機構とした。

ロケットでの上昇中及びパラシュートでの降下中、パラシュート巻取り機構はテグスで巻取り部を固定されており、電熱線でテグスを切ることでパラシュートが巻き取られる。

➤ 収納シェル

CanSat 本体は収納シェル（図 1-6）に収納してキャリアに納める．収納シェルの模式図を図 4-7 に示す．収納シェルはパラシュート巻取り機構と一体になっている．

CanSat の制御基板には電熱線を 2 か所設けており，それぞれが巻取り機構固定テグスと収納シェル固定テグスと接した状態で CanSat 本体を収納する．パラシュート巻取り機構を動作させるときや収納シェルから CanSat 本体が脱出するときはそれぞれの電熱線を熱してテグスを焼き切ることで動作させる．

➤ **バッテリーの取付部・スタビライザーの展開**

コンベックスメジャーの復元力で展開する構造となっている．また，スタビライザーの中間にバッテリーを取り付けることで，回転軸から重心を離す機構を兼ねている．

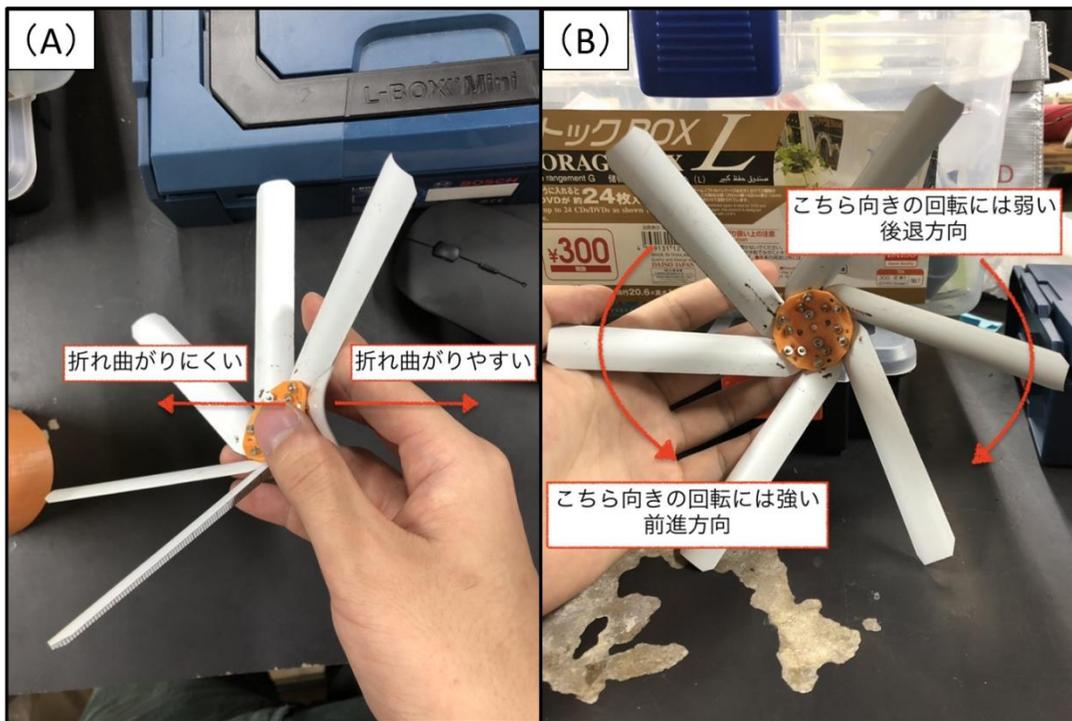


図 4-5 展開車輪

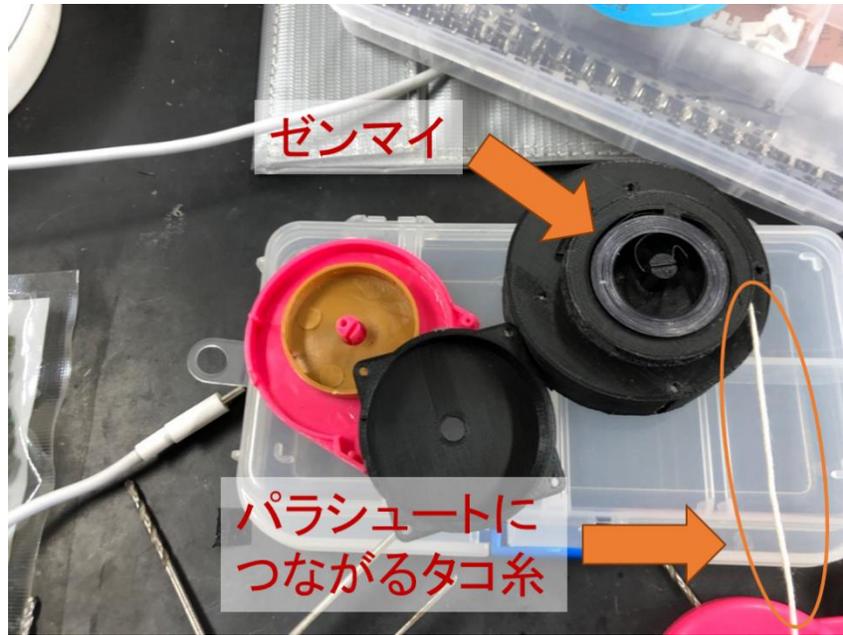


図 4-6 パラシュート巻取り機構

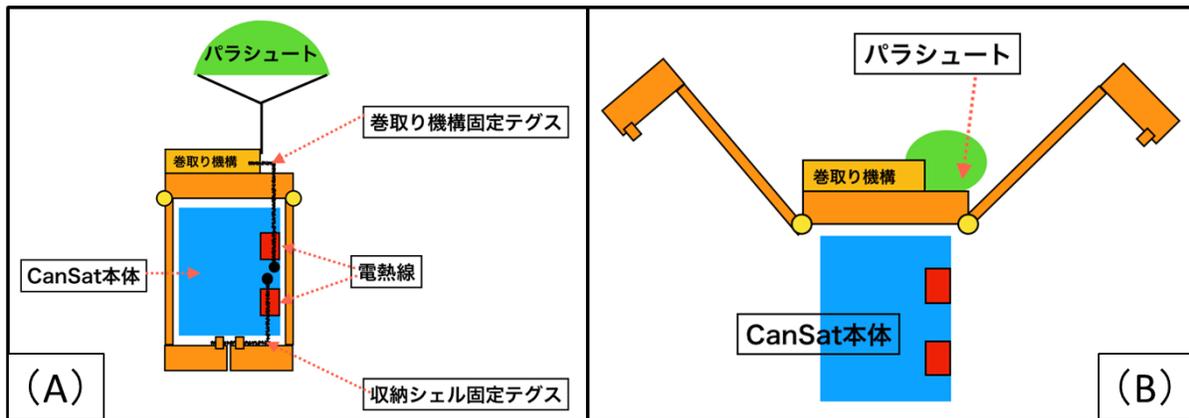


図 4-7 シェルの模式図

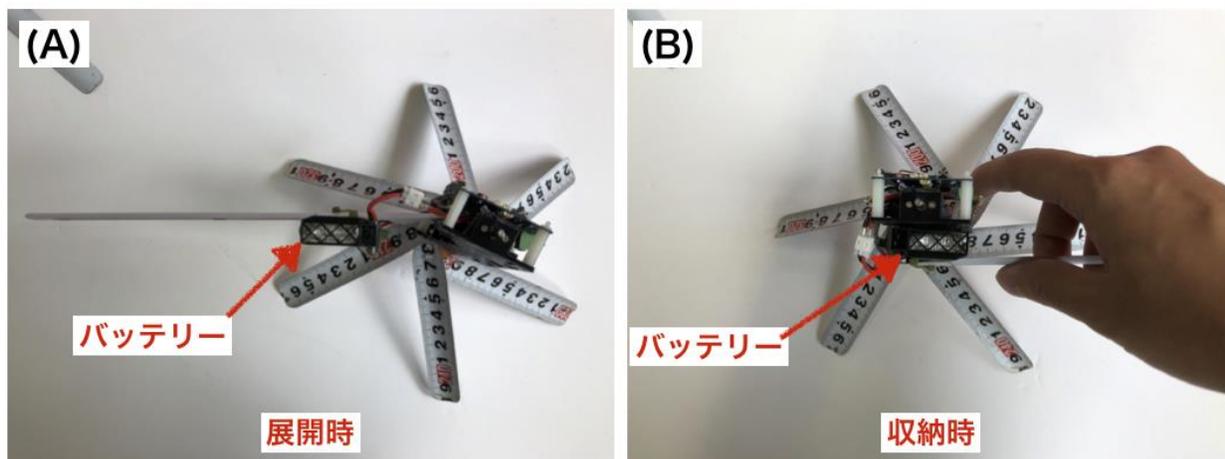


図 4-8 スタビライザー・バッテリーの展開

B) 回路

CanSat の制御回路について説明する． CanSat の制御回路は KiCAD を用いてプリント基板を設計し，中国の企業（ELECROW）に発注した． 製作したプリント基板を図 4-9 に示す． 図 4-9 (A) は電子部品を実装する前の基板である． 図 4-9 (B) (C) は電子部品を実装した基板である（写真では TWE-Lite と 9 軸センサが未実装）． 図 1-1 のシステム図で示した機能を満たす回路部品を実装している．

基板のサイズは 40×60 mm である． 使用している電子部品について表 4-1 に示す． また，動作確認の動画をリンク先 (<https://youtu.be/-eqpR2kEMmg>) に投稿している．

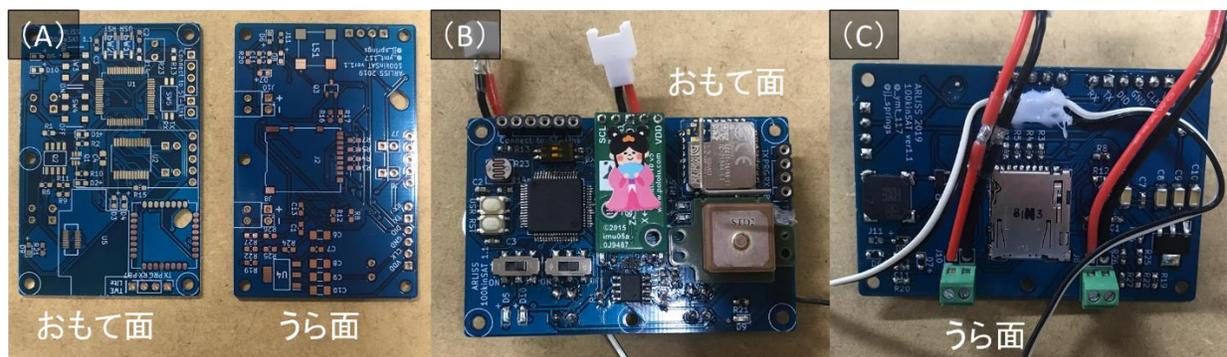


図 4-9 CanSat の制御回路

表 4-1 基板の詳細

マイコン	STM32F446RE
GPS センサ	GYSFDMAXB
加速度・ジャイロセンサ	LSM6DS33
地磁気センサ	LIS3MDL

気圧センサ	LPS25H
照度センサ	MI527
スピーカ	UGCT7525AN4
モータドライバ	TB6612FNG
レギュレータ	ADP3338AKCZ-3.3

4. 使用部品

➤ 電子系

分類	名称・型番	入手先・参考情報等	備考
マイコン	STM32F446RE	https://www.digikey.jp/product-detail/en/stmicroelectronics/STM32F446RET6/497-15376-ND/5175962	
GPS	GYSFDMAXB	http://akizukidenshi.com/catalog/g/gM-09990/	
IMU	AltIMU-10 v5	https://www.switch-science.com/catalog/2659/	
CdS	MI527/MI5527	http://akizukidenshi.com/catalog/g/gI-00110/	
スピーカ	UGCT7525AN4	http://akizukidenshi.com/catalog/g/gP-09801/	
モータドライバ	TB6612FNG	http://akizukidenshi.com/catalog/g/gI-11317/	
レギュレータ	ADP3338AKCZ-3.3	http://akizukidenshi.com/catalog/g/gI-12635/	
LED	OSHR1608C1A	http://akizukidenshi.com/catalog/g/gI-03978/	その他カラーも使用
ダイオード	RB521G-30T2R	http://akizukidenshi.com/catalog/g/gI-13956/	GPS 用
タクトスイッチ	SKRSPACE010	http://akizukidenshi.com/catalog/g/gP-06185/	リセットボタン等
スライドスイッチ	SSSS213202	http://akizukidenshi.com/catalog/g/gP-05043/	電源用
DIP スイッチ	KHS22E	http://akizukidenshi.com/catalog/g/gP-08928/	TWE-Lite 用
コンデンサ	GRM31CB31E226KE15	http://akizukidenshi.com/catalog/g/gP-06038/	各 IC のパスコン等 その他容量も使用
FET	FDS4935A	http://akizukidenshi.com/catalog/g/gI-01246/	電熱線用
FET	2N7002DW	http://akizukidenshi.com/catalog/g/gI-05999/	電熱線用
SD コネクタ	DM3AT-SF-PEJM5	http://akizukidenshi.com/catalog/g/gC-02395/	
抵抗	RC0603J1K	http://akizukidenshi.com/catalog/g/gR-06102/	1k, 10k 等
TWE-Lite	TWE-L-W3	http://akizukidenshi.com/catalog/g/gM-13309/	
コネクタ	WJEK254-2.54-02P-140-00A	http://akizukidenshi.com/catalog/g/gP-14217/	電熱線用
コネクタ	B2B-PH-K-S	http://akizukidenshi.com/catalog/g/gC-12802/	マイコンバッテリー用

➤ 動力系

分類	名称・型番	入手先・参考情報等	備考
モータ	Pololu 型番: 1097	https://store.shopping.yahoo.co.jp/suzakulab/pololu-1097.html	

➤ 構造系

分類	名称・型番	入手先・参考情報等	備考
プラねじ			サイズは M2
プラナット			サイズは M2
金属ねじ		ハンズマン	サイズは M2
金属ナット		ハンズマン	サイズは M2
スプリングワッシャー		ハンズマン	サイズは M2
メジャー		100円ショップ	
3D プリントフィラメント	PLA	https://www.amazon.co.jp/dp/B072BBVFJB/ref=cm_sw_r_tw_dp_U_x_NEtPDbNWBPW2R	

5. 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
プログラム	Arduino IDE 1.8.9		
プログラム	Arduino Core STM32	https://github.com/stm32duino	
回路	KiCAD 5.1.2		
3D プリンタ	ALUNAR 【工場直販】 3D プリンター Reprap Prusa i3 DIY 未組立キット	https://www.amazon.co.jp/gp/product/B01JZ4GSGG/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1	中華性格安個人向け 3D プリンタ、購入時価格 3 万円程度 今回使用した 3D プリンタ部品はすべてこれで出力したもの

※3D プリンタや CNC などの近年発達の著しい機材や外部サービス、特殊な機材を使用した場合は記入してください。一般的な工具類は記入不要です。

6. プログラム・アルゴリズム

使用言語:C 言語

アルゴリズムは図 1-3 を参照。

7. 会計

プロジェクト全体で使用した金額と大まかな内訳

プロジェクト全体で使用した金額と内訳を表に示す。

項目	購入元等	単価	数量	合計購入価格	備考
本番用 CanSat 材料費		14921	5	74605	
振動試験諸経費	鹿児島県工業技術センター	1040	2	2080	(1)
振動試験交通費	レンタカー	5000	2	10000	(2)
EtoE 試験用ドローン 飛行費用	エルム・スカイアクション	32400	1	32400	(3)
EtoE 試験交通費	レンタカー	5000	1	5000	(2)
東京-アメリカ交通費	ユナイテッド航空	106160	4	424640	
鹿児島-東京交通費	ジェットスター・スカイマー ク	15000	4	60000	(4)
海外旅行保険	損害保険ジャパン日本興亜	5210	4	20840	
ESTA		14 ドル	4	6000	
ARLISS 参加費		15000	4	60000	
打ち上げ費用		135 ドル	2	30000	
薄板用ドリル	ハンズマン	500	2	1000	(5)
グルーガン (コード レス)	アマゾン	2600	1	2600	
ジャンパワイヤ	アマゾン	750	1	750	
3D プリント消耗品類	アマゾン	2000	1	2000	(6)
3D プリントフィラメ ント		3000	2	6000	
リポバッテリー類	アマゾン	10000	1	10000	(7)
電子部品関係	主に秋月電子, アマゾン	30000	1	30000	(8)
プリント基板	Elecrow	2000	3	6000	(9)
試作費用	主にダイソー, ハンズマン	10000	1	10000	(10)
			合計	793915	

※単価は単位の無いものは「円」、ドルで支払ったものは「ドル」

※合計購入価格のところでドルをおおよそで円に換算した

- (1) 振動試験を2回行ったため
- (2) 大まかな値段（ガソリン代含む）
- (3) ドローンのオペレーション費用，実験場手配等も含む，半日程度&学生ということで割引していただいた
- (4) 各自異なるため大体の価格，鹿児島空港へのバス代も含む
- (5) 車輪メジャーの穴開け用
- (6) 細かい消耗品，プラットフォームシート，ノズル等（フィラメントは含まず）
- (7) 結果使わなかったものも含む，今回のプロジェクトのために買ったもの
- (8) 試作段階で買ってみた部品や予備で買ったが使わなかった電子部品
- (9) 試作の基板
- (10) 試作段階の材料などにかかった費用

第5章 試験項目設定（項目別試験、結合試験、EndtoEnd 試験）

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）	実施日
V1	質量試験	M1	7/5
V2	キャリア収納試験	M1	7/5
V3	ロスト対策試験	M2	7/4
V4	落下試験	M3, S2, S3, S4	8/16
V5	分離・展開試験	S2, S3, S5, S6	8/18
V6	静荷重試験	M4, S2	7/5(1回目) 8/18(2回目)
V7	振動試験	M5, S2	8/6(1回目) 8/19(2回目)
V8	放出時衝撃試験	M6	8/22
V9	通信試験	M7	6/25
V10	通信チャンネル変更試験	M8	6/25
V11	End-to-End 試験	M9	8/21
V12	走行試験	S8, S9	7/7
V13	位置情報試験	S10, S14	7/4
V14	自律制御走行試験	S11	8/18
V15	横転復帰試験	S7	7/29
V16	制御履歴記録試験	S12	8/2

V17	制御履歴提出試験	S12	8/24
V18	通信機器電源 ON/OFF 試験	S1	7/24
V19	ゴール判定試験	S15	8/21
V20	電力試験	S13	8/18

第6章 実施試験内容

(上記試験項目リストの番号・検証項目名を記し、それぞれについて「試験／解析の内容」「結果」「結論」を記述すること(適宜、図表・写真を挿入し、取得したデータを添付すること)

1. 質量試験 【済】

➤ 目的

質量がレギュレーションを満たしていることを確認する。

➤ 試験/解析内容

全モジュールを計りで計測し、350g以下であることを確認する。

➤ 結果

図6-1にCanSatの質量試験の様子を示す。キャリアを載せた状態で0g表示にセットした計りにCanSat(本体、収納シェル、パラシュート、バッテリー)を載せて計測した。計測の結果、質量は315gとなった。これは350mlクラスのレギュレーションを満たしている。



図 6-1 質量試験の様子

2. キャリア収納試験 【済】

➤ 目的

ARLISS2019 規定のロケットに収納可能であることを確認する。

➤ 試験/解析内容

全モジュールを ARLISS2019 規定のキャリアと同サイズのケースに収納し、収納可能であることを確認する。収納方法（パラシュートのたたみ方、ローバの形状など）が、大会当日と同じになるようにする。

➤ 結果

3D プリンタを用いて ARLISS2019 規定サイズのキャリアを制作した(図 6-2)。2 機の CanSat をそれぞれシェルに収納した状態でキャリアに収納する。

シェル収納の様子をリンク先 (<https://youtu.be/SZ1ObeJPzgA>) に投稿している。

CanSat2 機およびパラシュートを上記の容器に収納することができた (図 6-3)。



図 6-2 収納容器



図 6-3 CanSat とパラシュートを 2 機分，収納容器に収納した様子

3. ロスト対策試験 【済】

➤ 目的

- ① パラシュートを含む CanSat 本体が視認できる最大視認距離を確認する.
- ② CanSat 搭載無線機の通信可能距離を測定する.

・①と②の結果がローバのロスト対策として有効であることを確認する.

➤ 試験/解析内容

(ア) 試験方法 1 視認性試験

パラシュートがどれくらいの距離まで視認できるか試験する.

視認できる最大距離を GPS 座標から求め, 最大視認距離とする.

(イ) 試験方法 2 通信距離試験

2台の TWE-Lite の通信距離を遠ざけ, 接続が切れたときの距離を通信可能距離として距離を調べる. 距離はそれぞれの地点の GPS 座標から求める.

➤ 結果

(ア) 試験結果 1 視認性試験

大学構内で最も長い直線距離をとれる北辰通り (図 6-4) 約 550m で視認性試験を行った. 試験の結果, 図 6-5 に示したピンク色のパラシュートを目視でかろうじて確認できた. 双眼鏡を使った場合, しっかりと視認できた.

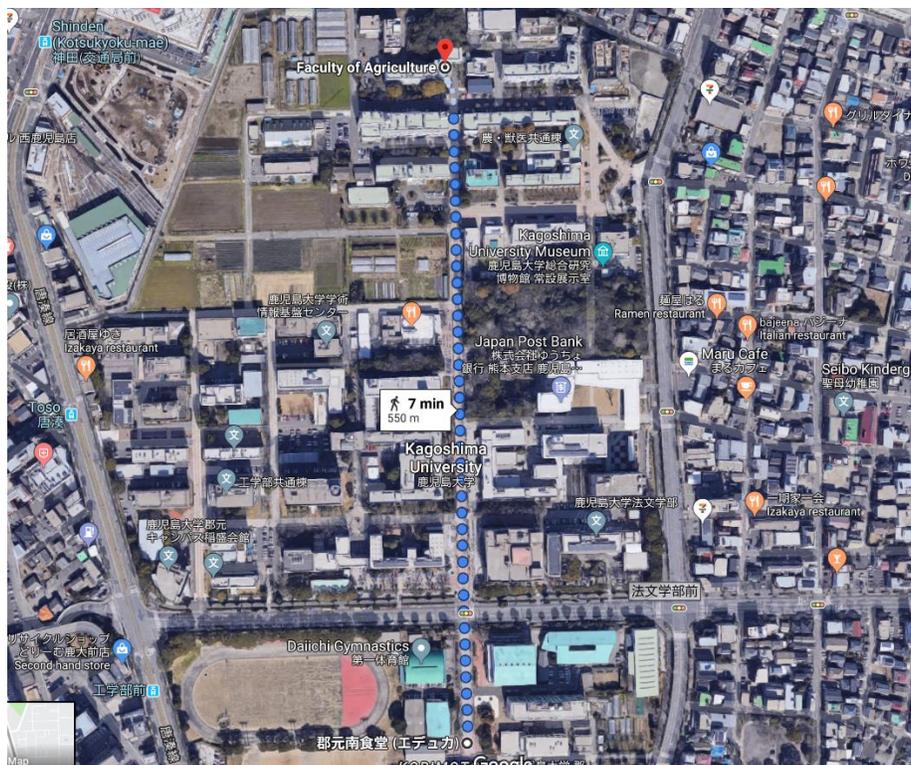


図 6-4 視認性試験の場所（北辰通り）



図 6-5 視認性試験に使用したパラシュート

(イ) 試験結果 2 通信距離試験

機体に搭載予定の通信機（TWE-Lite BLUE）2 台の距離を徐々に遠ざけながら通信を行い、通信可能距離を直線距離で何 m になるか調べた。

最大で 200m 程度の距離で通信速度が非常に遅くなったが、通信できた。このことからロスト対策として、通信機を用いた CanSat 捜索は 200m 程度の通信距離で可能なことが分かった。

また、実験時は雨が降っており、周囲に建物が並んでいる環境であったため、ARLISS 会場の環境では 200m より長い距離で通信可能になると予想される。今後、ARLISS 会場により近い環境下（天気晴れ、見渡しの良い場所）で通信状況や通信距離を調べる。

まとめ

①の試験結果より、500m 程度の距離であれば目視でパラシュートを確認できることがわかった。また、②の試験結果より、200m 程度の距離であれば TWE-Lite の通信により CanSat の位置を補足できることがわかった。

4. 落下試験 【済】

➤ 目的

パラシュートを用いた落下試験を行い、パラシュートが確実に開くこと、危険のない速度で落下することを確認する。また、着地時の衝撃でローバに損傷や異常がないか確認する。

➤ 試験/解析内容

1. ドローンからの落下試験

ドローンからの落下試験は End-to-End 試験と同時に行った。ドローンで高さ 50m まで機体を持ち上げ放出させた。機体が放出・落下状態を検知できていること、着地時の衝撃で損傷・異常がないことを確認する。

➤ 結果

落下時の気圧センサから求めた高度を図 6-6 に示す。グラフが直線になっているため 50m から落下させた場合は終端速度に達していると考えられる。機体が 6.84m から 3.6m までの落下には 600ms かかったため終端速度はおよそ 5.4m/s であった。

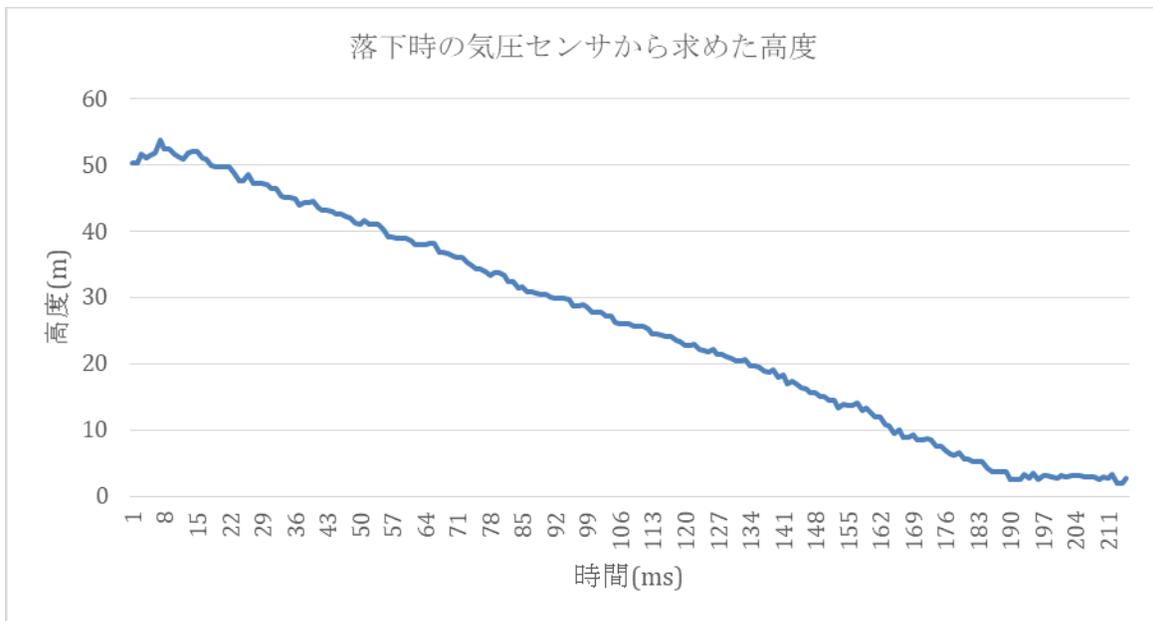


図 6-6 落下時の気圧センサから求めた高度

4回試験を行い、そのうち4回成功した。結果を表6-1に示す。

表 6-1 落下試験の結果

	結果	リンク先
1回目	放出・落下の判定ができた	https://youtu.be/uXblsWbEIjQ
2回目	放出・落下の判定ができた	https://youtu.be/xwsRu8c5rbQ
3回目	放出・落下の判定ができた	https://youtu.be/hnQP6ZhDfD0
4回目	放出・落下の判定ができた	https://youtu.be/wszQjJhTrp0

➤ **結論**

実験から放出・落下を検知できること、着地時の衝撃で損傷・異常がないことが確認できた。

5. 分離・展開試験 【済】

➤ 目的

地上落下後に電熱線でテグスを焼き切り、パラシュートを巻き取ること、ローバの展開タイヤが展開できることを確認する。

➤ 試験/解析内容

ドローンで高さ 50m 地点まで機体を持ち上げた後に放出し落下させる。放出・落下検知後、1つ目の電熱線でパラシュート巻き取り用のテグスを焼き切った後、2つ目の電熱線で機体を展開するためのテグスを焼き切り、機体が展開することを確認する。

➤ 結果

分離・展開試験は4回行い、そのうち2回成功した。

	結果	リンク先
1回目	パラシュートが機体に絡まっていたため分離・展開できなかった	https://youtu.be/uXblsWbEljQ
2回目	分離・展開に成功した	https://youtu.be/xwsRu8c5rbQ
3回目	パラシュートが機体に絡まっていたため分離・展開できなかった	https://youtu.be/hnQP6ZhDfD0
4回目	分離・展開に成功した	https://youtu.be/wszQjJhTrp0

➤ 結論

4回中2回パラシュートが機体に絡まってしまう、分離・展開できなかった。2機投下するので、成功確率が5割であればどちらかは分離・展開に成功すると考える。

6. 静荷重試験 【済】

➤ 目的

ロケットの打ち上げにかかる想定される大きさの静荷重をローバに与え、ローバがそれに耐えられることを実証する。

➤ 試験/解析内容

機体を入れた容器に紐をつなぎ、10Gの静的荷重がかかるまで回転させる。荷重がかかっているかどうかは加速度センサの値を無線で送信してPCで確認する。10Gがかかったところから10秒間経過したところで回転を止める。その後機体の損傷がないかを調べる。

図6-9は上記の試験方法において、人間がCanSatを地面に対して平行に振り回すことが厳密には不可能であるため、地面に対してある程度斜めに（図中では θ_0 ）ずれた状態で振り回したとしても十分な静的荷重を加えることが可能であることを求めたものである。

➤ 結果

試験用キャリアにシェルに収納したCanSat2機を入れて紐付きの蓋を取り付けた（図6-7）。

ハンマー投げの要領で試験用容器を振り回してCanSatに静的荷重を加えた。

CanSatに搭載された加速度センサの値を無線機でPCに送り、10Gを超えた状態を10秒以上保つ、これを振り回し方や振り回す人を変えて数回行った。

また、その間のデータをSDカードに記録し、後にデータをグラフ化し再確認を行った（図6-8）。

その後、CanSatに故障や部品の外れ等が無いか目視で確認し、基板が正常に動くか試した結果、故障や部品の外れ等は無いことが確認できた。

実験の様子をリンク先（<https://youtu.be/RYrplDcayuw>）に投稿している。



図 6-7 静的荷重試験用の容器

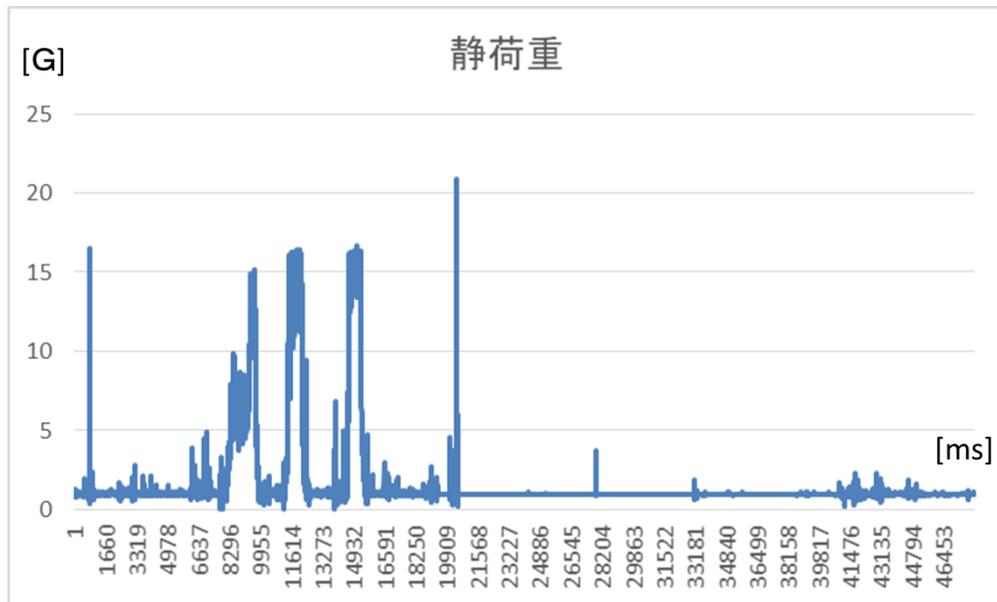


図 6-8 静的荷重試験で所得した値

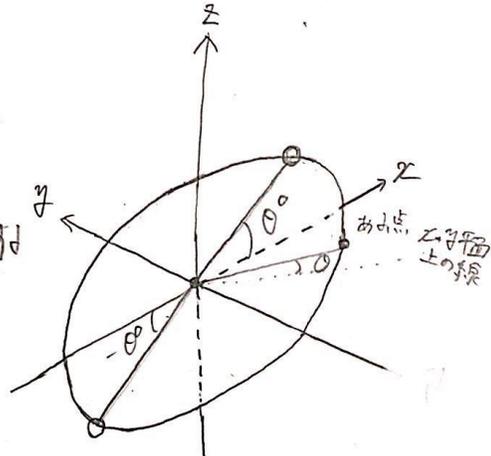
z軸を中心に、x-y平面に對して、

右図のように θ_0 傾いた平面上に回転する質点を考える。

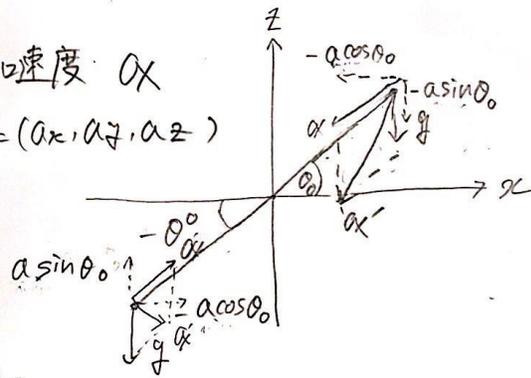
このとき、円周上のあつ点の x-y 平面に對する傾きを θ とすると、

$$-\theta_0 < \theta < \theta_0$$

$$\left(\theta_0 < \theta_0 < \frac{\pi}{2} \right)$$



円運動による中心方向の加速度 α
 実際に見いだせる加速度 $\alpha' = (\alpha_x, \alpha_y, \alpha_z)$
 重力加速度 g



とすると、 $(|\alpha| = a)$

$$\alpha = -a \cos \theta e_x - a \sin \theta e_z$$

$$g = g e_z$$

$$\alpha' = \alpha + g$$

$$= -a \cos \theta e_x + (g - a \sin \theta) e_z$$

$$|\alpha'| = \sqrt{a^2 \cos^2 \theta + (g - a \sin \theta)^2}$$

$$= \sqrt{a^2 \cos^2 \theta + g^2 - 2ag \sin \theta + a^2 \sin^2 \theta}$$

$$= \sqrt{a^2 - 2ag \sin \theta + g^2}$$

$$(|\alpha'| = \sqrt{\alpha_x^2 + \alpha_y^2 + \alpha_z^2})$$

$$|\alpha'| = A \text{ とおくと}$$

$$(a=10, g=1 \text{ とする})$$

(x-y平面に平行な単位ベクトル e_x, e_y
 z軸方向の単位ベクトル e_z)

$\theta_0 = 0$ のとき

$$A_{\min} = A_{\max} = \sqrt{100 + 1}$$

$$= \sqrt{101}$$

$$\theta_0 = \frac{\pi}{2} \text{ のとき}$$

$$A_{\max} = \sqrt{100 + 20 + 1}$$

$$= 11$$

$$A_{\min} = \sqrt{100 - 20 + 1}$$

$$= 9$$

図 6-9 静荷重試験の計算式

7. 振動試験 【済】

➤ 目的

ロケットの打ち上げ時の振動にローバが耐えられることを確認する。

➤ 試験/解析内容

CanSat とパラシュートを振動試験用容器に収納（レギュレーション規定を守るサイズのもの）、加振機を用いてロケット上昇時に生じる振動および衝撃を模擬した状況を作り CanSat に与える。その後パラシュート巻取り・収納シェルの展開・CanSat の走行開始が正常できることを確認する。

➤ 結果

鹿児島県工業技術センターにて加振機を使用した試験を行った。試験条件は正弦波掃引 30Hz～2000Hz で行った。試験結果と動画のリンク先を表 6-2 に示す。また、ローバが記録した加速度の大きさのグラフを図 6-10 に示す。

8/9 に行った試験では microSD カードが外れてしまった。これは microSD カードを取り出しやすいように基板から少しはみ出す設計にしていたことが原因であった。そこで、8/19 に microSD カードの位置を工夫して再設計した基板（図 4-9）で再試験を行った。その結果、microSD カードが外れることなく記録できた。

表 6-2 振動試験の結果

日付	結果	リンク先
8/9(1回目)	振動で SD カードが外れてしまい、記録できていなかった(8/19 に再設計した基板で再試験して解決) 回路の動作に異常はなかった 一部ねじが外れていた(2回目の試験にてホットボンドで固定することで解決)	https://youtu.be/drk72MZECQk
8/9(2回目)	振動で SD カードが外れてしまい、記録できていなかった 回路の動作に異常はなかった	https://youtu.be/1tawXE0LOU4
8/19	SD カードに記録できていた 回路の動作に異常はなかった	https://youtu.be/TucpjEy6ufg

図 6-11 に実験に使用した容器・治具を示す。

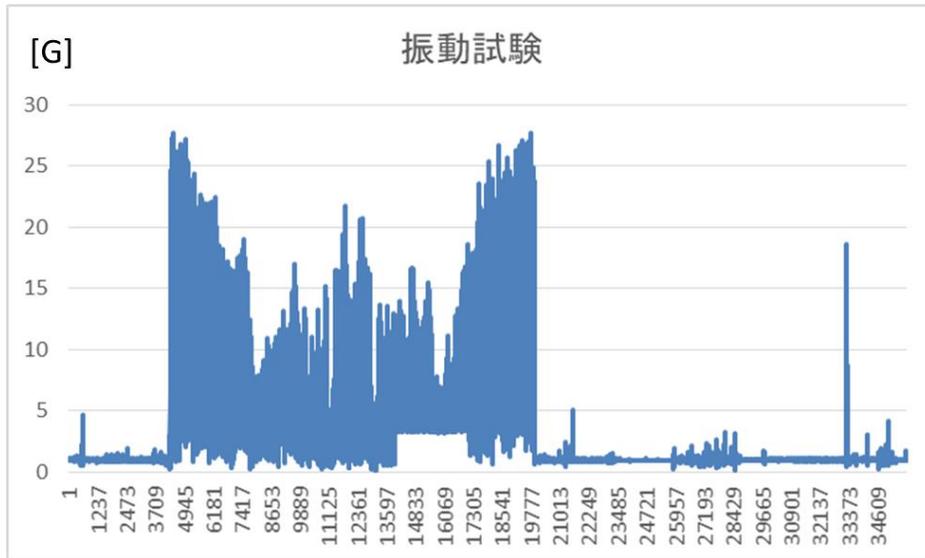


図 6-10 振動試験で CanSat が記録した加速度の大きさ

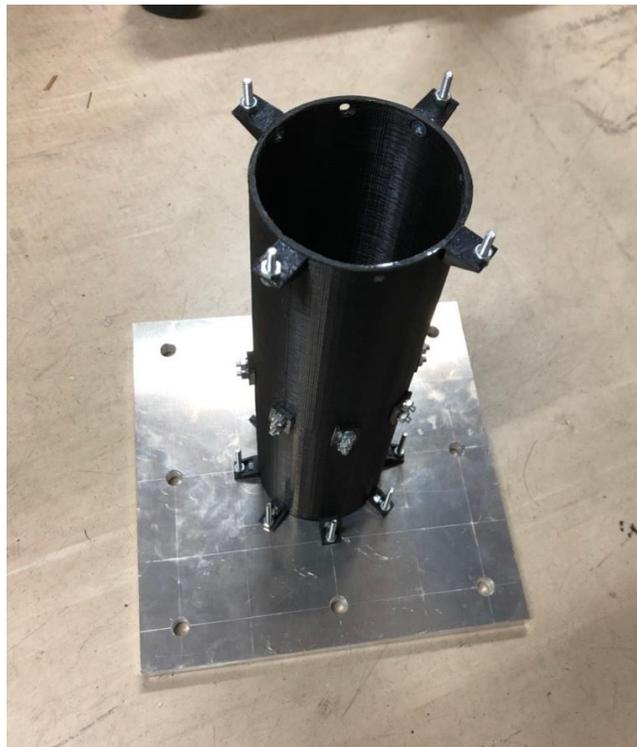


図 6-11 容器・治具

8. 放出時衝撃試験 【済】

➤ 目的

ローバがロケットから放出されるときの衝撃に機体が耐えうるかを衝撃試験により確認する。また、ロケットから放出され、パラシュートが開傘する際の衝撃に結合部が耐えられ、パラシュートや機体に損傷がないことを確認する。

➤ 試験/解析内容（1）

CanSat のパラシュート部を持ち、CanSat を落下させて紐の長さ分の自由落下の衝撃を CanSat とパラシュートに与える。これを数回繰り返し、CanSat に損傷等がないことを確認する。また、この実験中の加速度センサの値を CanSat の SD カードに記録しておき、実際にどのくらいの衝撃が加わったのか確かめる。

➤ 結果（1）

使用している加速度センサの測定上限が 16G なため、16G 以上の放出衝撃は測定することができなかった（図 6-12）。また、センサの時間分解能が足りないため、記録された値が加えた衝撃のピーク値なのかわからなかった。

しかし、CanSat に 16G 以上の衝撃を複数回加えても、損傷等がないことが確認できた。

➤ 試験/解析内容（2）

加速度センサの測定上限が 16G なため、加速度センサの範囲内で 40G 相当の衝撃を加える方法を考えた。

収納時の CanSat 下部に重りを取り付けた状態で、実験 1 と同様な実験をすることでセンサから測定される衝撃荷重が小さくとも 40G 相当の衝撃を加えることができる。

運動方程式 $F=ma$ より

CanSat 質量 $m:0.2[\text{kg}]$ (2 機合わせた重量が 315g なため、1 機 157.5g なので、計算しやすく 200g として設定)

レギュレーションより、40G の衝撃を加えるので

$$0.2 \times 40 = F \quad \dots (1)$$

重りに、近くにあったお茶 500ml を利用することとしたので、重りの重さが 0.5kg のため、CanSat+重りの重さは

$$0.2+0.5=0.7\text{kg}$$

これに対して運動方程式を作ると以下ようになる。

$$0.7 \times a = F \quad \dots (2)$$

式(1)(2)の F が等しい時の(2)式の a が CanSat 単体で 40G の衝撃を加えた時と等しくなる。計算すると $a=11\text{G}$ となるので重りをつけた状態で、センサの値が 11G を超えていれば、CanSat に 40G 相当の衝撃を加えたとみなすことができる。

➤ 結果 (2)

実験後に SD カードに記録された値を見ると、16G 以上の衝撃が加わっていることがわかった (図 6-13)。また、CanSat に損傷等は見られなかった。よって、CanSat に加わった衝撃は 40G 相当よりも大きく、またその衝撃でも損傷がなかったため 40G の放出時衝撃には耐えられる。

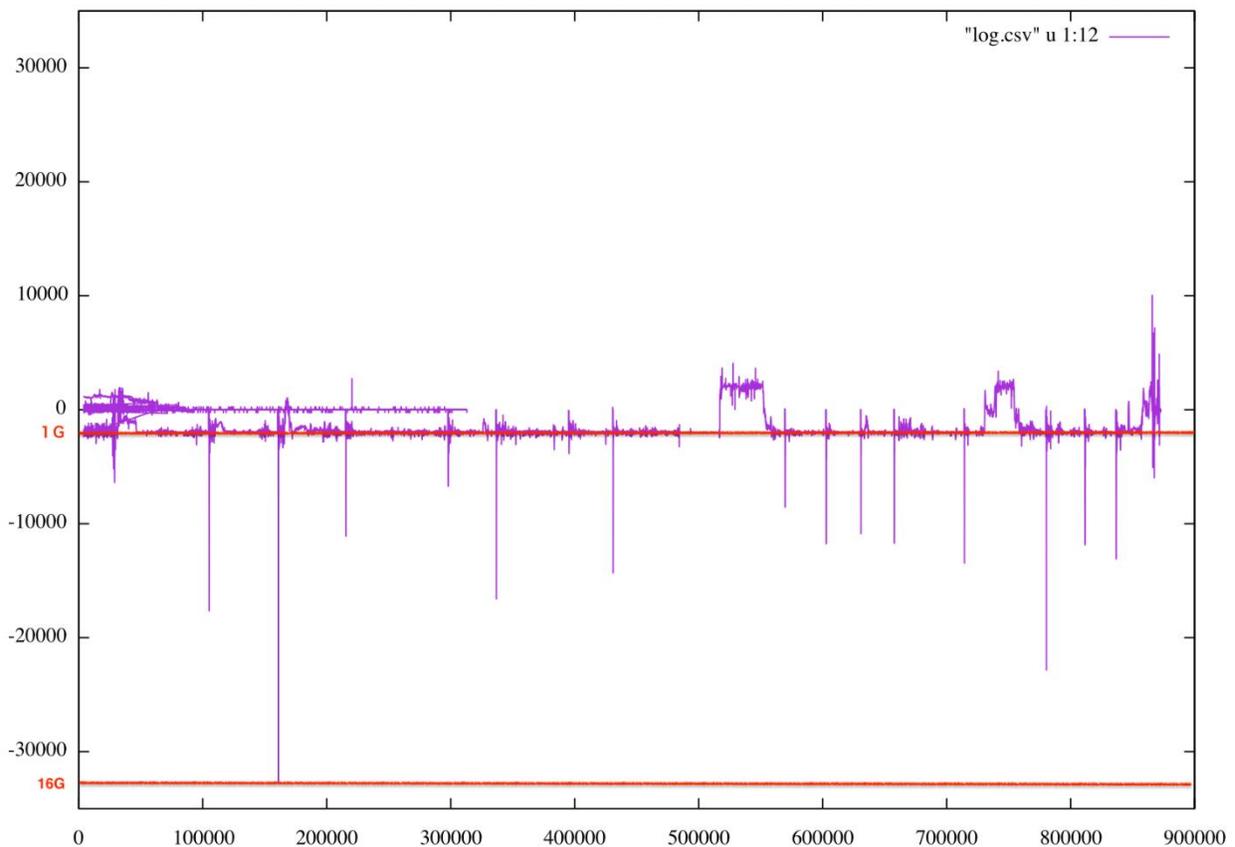


図 6-12 放出時衝撃試験 (1) の測定結果

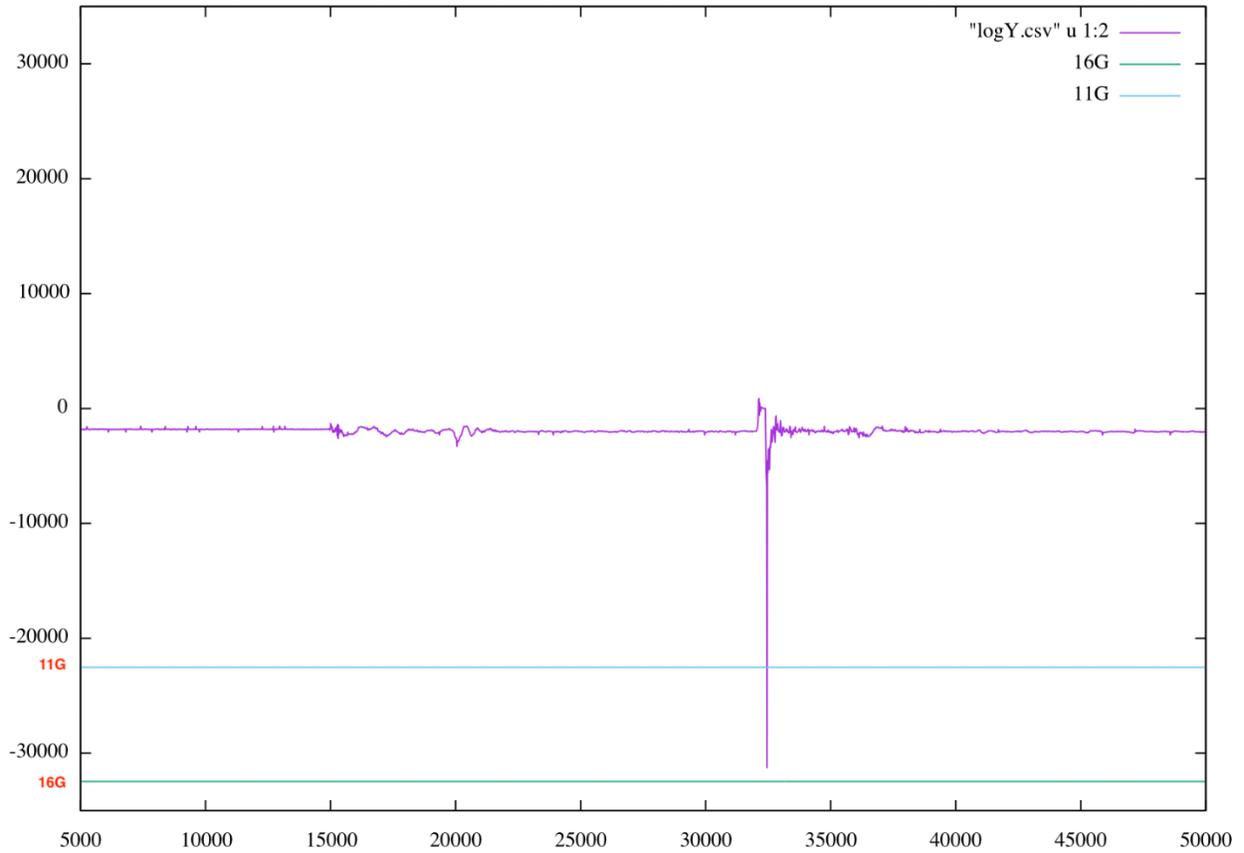


図 6-13 放出時衝撃試験（2）の測定結果

表 6-3 試験の動画

	リンク先
試験（1）	https://youtu.be/CZKTZUHPC_o
試験（2）	https://youtu.be/KZHjJuWlxOw

9. 通信試験 【済】

➤ 目的

長距離通信が行えることを確認する。また、打ち上げ時の無線機の基準を満たすことを確認する。

➤ 試験/解析内容

CanSat に搭載されている無線機からの電波を、PC に接続された無線機で受信できることを確認する。受信データを接続した PC で確認することができるかどうか確認する。

また、打ち上げ時の無線機の基準「打ち上げ時の無線機の電源 OFF 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。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））」を満たすことを確認する。

➤ 結果

CanSat の照度センサの値を無線機で送信しながら、照度センサを指で隠したり開いたりして無線から送られて来る値が変化することが確認できた。そのときの実験の様子をリンク先 (https://youtu.be/vO2rdoTU_3Q) に投稿している。

また、今回搭載予定の無線モジュールは TWE-Lite BLUE であり、このモジュールは FCC 認証かつ 100mW 以下の機器のため、規定上は機器の電源を OFF する必要は無いはずであるが、無線出力を ON, OFF できるか試した。

TWE-Lite BLUE は 27 番ピンが LOW になるとスリープモードとなり、無線出力を停止する、マイコンと 27 番ピンをつなぎ、H/L を行い通信が途切れることが確認されたので、無線出力が ON,OFF できた。

次に大学のグラウンドにて無線通信が長距離で可能であることを確認した。試験の様子を図 6-14 に示す。試験の結果から、約 170m 離れた地点で通信できることが確認できた（図 6-15）。

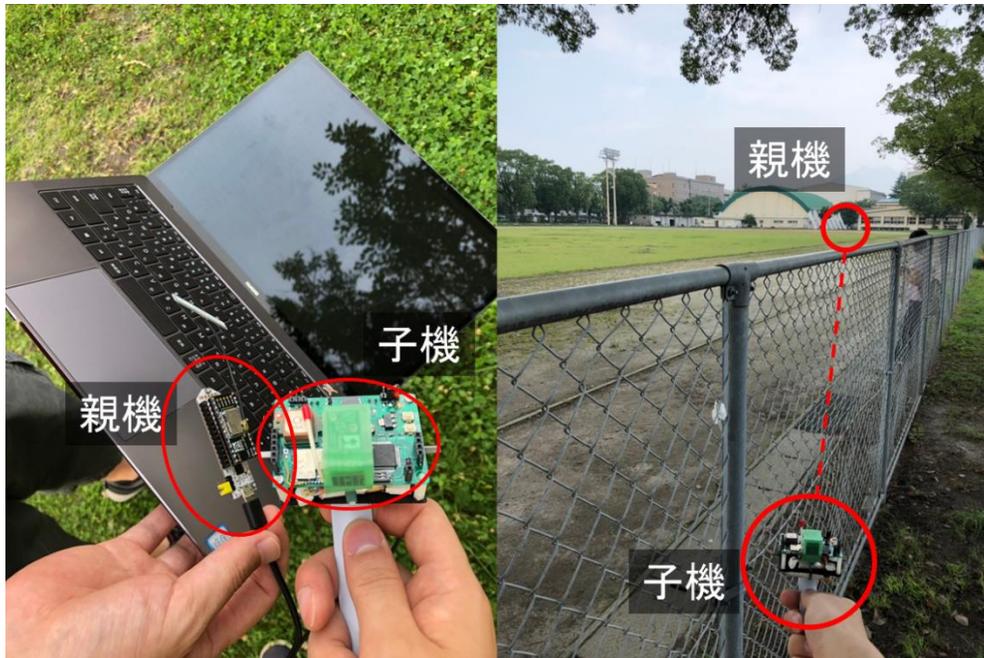


図 6-14 通信試験の様子



図 6-15 親機と子機の通信距離

10. 通信チャンネル変更試験 【済】

➤ 目的

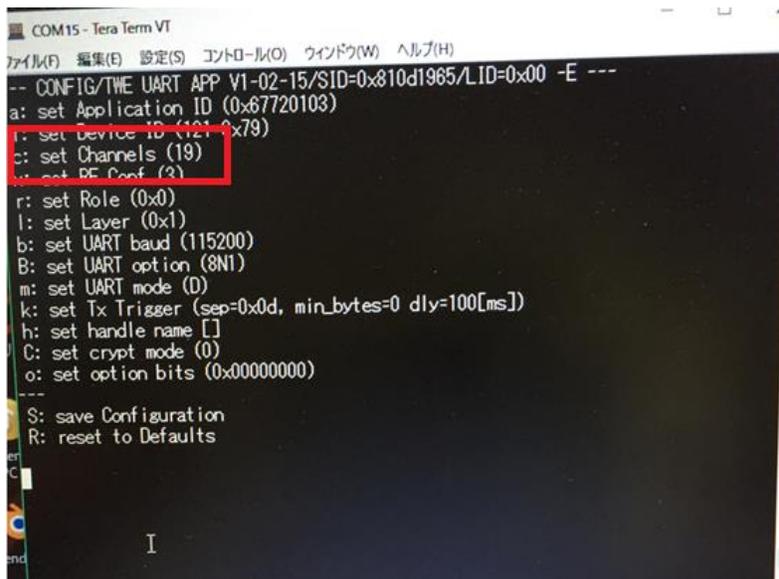
無線の周波数変更が行えることを確認する。

➤ 試験/解析内容

無線のチャンネルを変更すると周波数が変更できる。まず、TWE-Lite BLUE の受信機のチャンネルを 19ch に設定する。そして、送信機から送ったデータが受信機に送られているか受信機に接続したパソコンから確認する。その後、受信機のチャンネルを 18ch にプログラム内で設定し直し、同様に送信機からデータが送られているか確認したところ届かないことを確認する。

➤ 結果

プログラム中の以下の部分でチャンネルを設定する。チャンネルを 19ch に設定した場合、送信機から受信機へデータが送られてくることが確認できたが、18ch に変更したところデータは送信されて来なくなった。



```
COM15 - Tera Term VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
-- CONFIG/TWE UART APP V1-02-15/SID=0x810d1965/LID=0x00 -E ---
a: set Application ID (0x67720103)
r: set Device ID (1210x79)
c: set Channels (19)
r: set RF Conf (3)
r: set Role (0x0)
l: set Layer (0x1)
b: set UART baud (115200)
B: set UART option (8N1)
m: set UART mode (D)
k: set Tx Trigger (sep=0x0d, min_bytes=0 dly=100[ms])
h: set handle name []
C: set crypt mode (0)
o: set option bits (0x00000000)
---
S: save Configuration
R: reset to Defaults
I
```

図 6-16 チャンネル変更試験の様子

11. End-to-End 試験 【済】

➤ 目的

M1-M8 の基準を満たした設計の機体によって、ロケットの装填から打ち上げ後の回収までを模擬した End-to-End 試験を実施し、今後、安全性に関わる大幅な設計変更がないことを確認する。

➤ 試験/解析内容

キャリアからローバが放出⇒着地⇒パラシュートからの離脱⇒目的地までの自律走行⇒ゴール判定という一連の流れを通してできるか確認する。キャリアを取り付けたドローンに CanSat を 2 機収納し、ドローンから投下することで実験を行った。

➤ 結果

実験の結果を表 6-4 に示す。実験の様子はリンク先 (<https://youtu.be/QOnq1QBYIsY>) に投稿している。表の各項目について説明する。

(1)～(4)は無線機で受信した情報によって動作確認した。(5)(6)は無線機で受信した情報と目視、音で動作確認した。(7)は目視で確認した。(8)はゴール到達までに時間がかかると判断したため中断した。(10)は GPS センサから所得した値のみ記録できていなかった。そのほかのデータは記録できていることを確認した。

表 6-4 End-to-End 試験の動作結果

	項目	結果
(1)	気圧センサによる上昇の検知	○
(2)	照度センサによる放出の検知	△
(3)	気圧センサによる着地の検知	○
(4)	ジャイロセンサによる着地の検知	○
(5)	電熱線によるパラシュートの巻取り	△
(6)	電熱線によるシェルの展開	○
(7)	走行	○
(8)	ゴールへの自律誘導	×
(9)	無線による CanSat の制御履歴確認	○
(10)	SD カードに保存したログの確認	△

※「○」は動作した。「△」は人の手を加えた結果動作した。「×」は動作しなかった。

【追加実験】

上記結果から達成できなかった(2)照度センサによる放出の検知と(8)ゴールへの自律誘導に関して追加実験を行い、動作することを確認した。実験の様子をリンク先 (<https://youtu.be/jcRyvMCOXgs>) に投稿している。

12. 走行試験 【済】

➤ 目的

ローバが走行するための機能を有していることを確認する。

➤ 試験/解析内容

ローバがモータを回転させることで前進，旋回，後退を平坦な地面で行うことができることを確認する。また，段差をどのくらいの高さを乗り越えられるかを確認する。

➤ 結果

製作した CanSat に前進，旋回，後退を繰り返すプログラムを書き込み，大学構内の芝生で走行試験を行った。試験の結果，前進，旋回，後退を行うことが確認できた。走行の様子をリンク先 (https://youtu.be/m3gTD_DoJeQ) に投稿している。

また，砂場で約 100 mm の段差を作り，CanSat が段差を超えることができるか試験した（図 6-17）。試験の様子をリンク先 (<https://youtu.be/rSS7zguT8CQ>) に投稿している。



図 6-17 走行試験の様子

13. 位置情報試験 【済】

➤ 目的

ローバがセンサから緯度・経度の位置情報を所得できることを確認する。

➤ 試験/解析内容

GPS センサで緯度と経度の情報を得ることができるかどうか調べる。また、GPS で得られた値と iPhone の「コンパス」アプリに表示された値を比較する。

➤ 結果

見通しの良い場所で図 6-18 に示すようにセンサ，マイコン，PC を接続して GPS の値を取得した。図 6-19 は GPS センサから所得した値をシリアルコンソールに表示したときのスクリーンショットである。5 つの値の平均をとったところ経度は 13032.61922° ，緯度は 3134.0967° であった。一方で，iPhone に搭載されているコンパスアプリで求めた経度は 13032.5° ，緯度は 3134.3° であった（図 6-20）。

これらの値から測地線長を求めたところ 10.406m であった（図 6-21）。緯度・経度からの測地線長には以下のサイトを利用した。

測量計算（距離と方位角の計算） - 国土地理院

<https://vldb.gsi.go.jp/sokuchi/surveycalc/surveycalc/bl2stf.html>

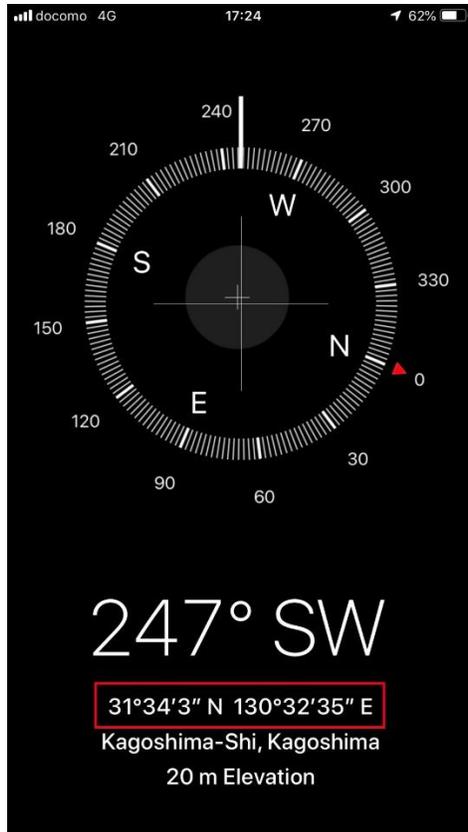


図 6-20 iPhone から所得した値
距離と方位角の計算

[トップページ](#)
[操作方法](#)
[計算式](#)
[お問い合わせ](#)

入力値

1点毎の計算 一括計算

座標値の入力方法 数値入力 地図上で選択

楕円体 GRS80

地図上で確認 地図クリア

出発点	緯度	3134.3
	経度	13032.35
到着点	緯度	3134.0967
	経度	13032.61922

入力単位選択 度分秒 十進法度単位

【緯度・経度の値の入力例(度分秒)】
緯度 36° 6'13.5892" → 360613.5892
経度 140° 5'16.2781" → 1400516.2781
ddd mm ss.s → dddmss.s

計算実行

計算結果

入力値

楕円体	GRS80	
出発点	緯度	北緯 0°31'34.3000"
	経度	東経 1°30'32.3500"
到着点	緯度	北緯 0°31'34.0967"
	経度	東経 1°30'32.6192"

出力値

測地線長	10.406(m)	
方位角	出発点→到着点	126°52'27.64"
	到着点→出発点	306°52'27.65"

印刷

図 6-21 国土地理院のサイトで計算した結果

14. 自律制御走行試験 【済】

➤ 目的

ローバが自律制御でスタート地点からゴール地点まで移動することを確認する。

➤ 試験/解析内容

GPS から取得した値をもとにゴールまで自律的に近づくことができるか確認する。実験環境は周囲で実験することのできるグラウンドを検討している。

➤ 結果

ローバが GPS センサから所得した緯度・経度の値によってゴールへ近づき、ゴール判定することを確認した。実験の様子は End-to-End 試験の追加実験動画を参照。動画のリンク先 (<https://youtu.be/jcRyvMCOXgs>) 。

15. 横転復帰試験 【済】

➤ 目的

ローバが横転し走行不能な姿勢になった場合に、走行可能な状態に復帰できるか確認する。

➤ 試験/解析内容

反転・横転状態になるように地面に置き、プログラムで復帰できることを調べる。また、走行途中で手で反転・横転状態にした場合も復帰できることを確認する。

➤ 結果

CanSat を横転した状態で地面に置き、CanSat が復帰できるか試験を行った。

~~結果として、現状では復帰ができないことがわかった。プログラムの改良も行ってみたが、復帰はできなかつたため、今後、機体本体の改良が必要となることがわかった。~~

バッテリーの取り付け位置を変更することで横転復帰できることを確認した。横転復帰の様子をリンク先 (<https://youtu.be/1rvnXIm059I>) に投稿している。

16. 制御履歴記録試験 【済】

➤ 目的

ローバの制御履歴を記録できることを確認する。

➤ 試験/解析内容

位置軌跡が自律制御の結果によるものだと示すための記録を microSD カードに記録する。

➤ 結果

各センサ（GPS センサ除く）から所得した値，制御モードの遷移状況を microSD カードに記録できることが確認できた。しかし、GPS から所得した値に関してのみ記録できていない現状である。自律誘導の際に無線機から受信した情報によると、GPS の値は所得できているため、プログラムのミスが考えられる。本番までに記録できるように対応する。

GPS センサから所得した値も問題なく記録できることを確認した。

17. 制御履歴提出試験 【済】

➤ 目的

記録した制御履歴から、ローバがゴールに向かって走行したことを証明する。また、規定に従った制御履歴を運営に提出できることを確認する。

➤ 試験/解析内容

ローバの動作試験のデータから実際にレポートを作成する。

➤ 結果

ローバに搭載しているセンサから所得した生のデータを時系列で csv ファイル (図 6-22) に記録した。また、生データだけでは人間にわかりづらいため、テキストベースでローバが制御アルゴリズムに則った制御を行えたか txt ファイル (図 6-23) に記録した。

csv ファイルには以下に示す項目を記録している。

- ・プログラム起動後の経過時間
- ・GPS センサから所得した緯度, 経度, 日付
- ・加速度, ジャイロ, 地磁気センサの値 (それぞれ 3 軸)
- ・気圧センサの値
- ・照度センサの値
- ・バッテリーの電圧 (分圧したものを IO ピンで読み込み)

次に txt ファイルに記録している情報について説明する。txt ファイルにはローバがどのモード (打ち上げ検知モードや走行モードなど) を実行しているか, ゴールとの距離や方位の値, 位置情報などの計算結果による IO ピンへの出力量 (モータや電熱線の制御量) はどれくらいかといった情報が時系列順で記録している。

また, 矢印付きで走行履歴をプロットできるように新規に csv ファイルを追加した。csv ファイルは緯度, 経度, 制御量 (角度) の 3 つの列から構成される。Python で作成したスクリプトでこの csv ファイルを読み込むことで矢印付きの図をプロットできる。実際の走行試験データから作成した図を図 6-24 に示す。

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	
1	millis	state	year	month	day	hour	minute	second	lat	lng	ax	ay	az	gx	gy	gz	
2	5240	6	2000		0	0	0	0	0	0	0	498	-16	-1969	28	28	-10
3	49342	6	2019	8	30	7	28	42	31.5681	130.5449	472	-14	-1969	15	27	-9	
4	71833	6	2019	8	30	7	29	4	31.56806	130.5449	454	10	-1965	16	33	-9	
5	80014	6	2019	8	30	7	29	12	31.56794	130.5449	493	3	-1944	21	25	-9	
6	102493	6	2019	8	30	7	29	34	31.56787	130.5448	470	-12	-1986	25	20	-10	
7	110674	6	2019	8	30	7	29	42	31.56794	130.5448	490	-14	-2000	28	20	-9	
8	133187	6	2019	8	30	7	30	4	31.56802	130.5448	472	10	-1964	30	18	-9	
9	141344	6	2019	8	30	7	30	12	31.56802	130.5448	486	-32	-1963	33	11	-10	
10	163855	6	2019	8	30	7	30	34	31.56803	130.5448	475	5	-1987	24	22	-11	
11	175765	6	2019	8	30	7	30	46	31.56803	130.5448	480	-32	-1986	16	37	-10	
12	198249	6	2019	8	30	7	31	8	31.56808	130.5449	477	11	-1971	28	26	-10	
13	206427	6	2019	8	30	7	31	16	31.56811	130.5449	471	3	-1974	24	8	-9	
14	228916	6	2019	8	30	7	31	38	31.56814	130.5449	478	8	-1953	24	6	-10	
15	237088	6	2019	8	30	7	31	46	31.56813	130.5449	481	-9	-1967	32	10	-21	
16	259593	6	2019	8	30	7	32	8	31.56813	130.5449	475	-3	-1944	15	34	-11	
17	271215	6	2019	8	30	7	32	19	31.56814	130.5449	495	-17	-1981	-175	28	-1	
18	293720	6	2019	8	30	7	32	41	31.56809	130.545	468	-23	-1991	1	20	-9	
19	301892	6	2019	8	30	7	32	49	31.56809	130.545	462	-8	-1959	27	5	-10	
20	324375	6	2019	8	30	7	33	11	31.56808	130.545	488	-26	-1986	14	29	-10	
21	335854	6	2019	8	30	7	33	22	31.56808	130.545	478	-21	-1964	40	18	-10	
22	358378	6	2019	8	30	7	33	44	31.5681	130.545	462	-12	-1990	26	12	-21	
23	366541	6	2019	8	30	7	33	52	31.56808	130.5449	484	-22	-1968	24	22	-10	
24	389046	6	2019	8	30	7	34	14	31.5681	130.5449	483	0	-1962	52	16	-10	
25	400298	6	2019	8	30	7	34	25	31.5681	130.5448	490	242	-1952	28	15	-10	
26	422801	6	2019	8	30	7	34	47	31.56811	130.5447	487	-31	-1959	21	24	-10	
27	430980	6	2019	8	30	7	34	55	31.56811	130.5447	464	-12	-1956	56	14	-10	
28	453452	6	2019	8	30	7	35	17	31.56811	130.5448	495	247	-1981	25	23	-10	
29	465355	6	2019	8	30	7	35	29	31.56811	130.5447	483	-2	-1979	88	45	-9	
30	487902	6	2019	8	30	7	35	51	31.56813	130.5449	487	-13	-1997	24	13	-10	
31	496430	6	2019	8	30	7	35	59	31.56814	130.5449	501	1	-1951	21	20	-10	
32	518945	6	2019	8	30	7	36	21	31.56817	130.5449	500	-2	-1964	24	25	-9	
33	527124	6	2019	8	30	7	36	29	31.56816	130.545	493	-13	-1959	19	24	-10	
34	549644	6	2019	8	30	7	36	51	31.56813	130.545	480	-18	-1968	26	28	-9	
35	557825	6	2019	8	30	7	36	59	31.56813	130.5449	494	0	-1952	31	15	-10	
36	580303	6	2019	8	30	7	37	21	31.56816	130.545	478	-5	-1987	27	16	-10	
37	588447	6	2019	8	30	7	37	29	31.56815	130.545	483	-9	-1986	22	12	-10	

図 6-22 csv ファイルのログ

system_log.txt - メモ帳

ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)

```
***** start program *****  
[State: Initialization] ... Start CanSat Program!!!  
Altitude OFFSET value: -52.38
```

```
[State: Run to GOAL]  
old Lat: 31.568688  
old Lng: 130.542595  
old Distance to GOAL: 140.89  
Moving forward 15000 [ms]  
Motor stop  
new Lat: 31.568648  
new Lng: 130.542468  
new Distance to GOAL: 144.09  
direction_tehta: 143.07  
direction_phi: 70.02  
diff_direction: 73.05  
[!] Adjust the direction of the rover [Turn LEFT] 106 [ms]  
old Lat: 31.568591  
old Lng: 130.542472  
old Distance to GOAL: 138.78  
Moving forward 15000 [ms]  
Motor stop  
new Lat: 31.568565  
new Lng: 130.542467  
new Distance to GOAL: 136.85  
direction_tehta: 140.71  
direction_phi: 9.42  
diff_direction: 131.28  
[!] Adjust the direction of the rover [Turn LEFT] 48 [ms]  
[!] Judgment Stack  
diff_distance: 2.90  
old Lat: 31.568565
```

図 6-23 txt ファイルのログ

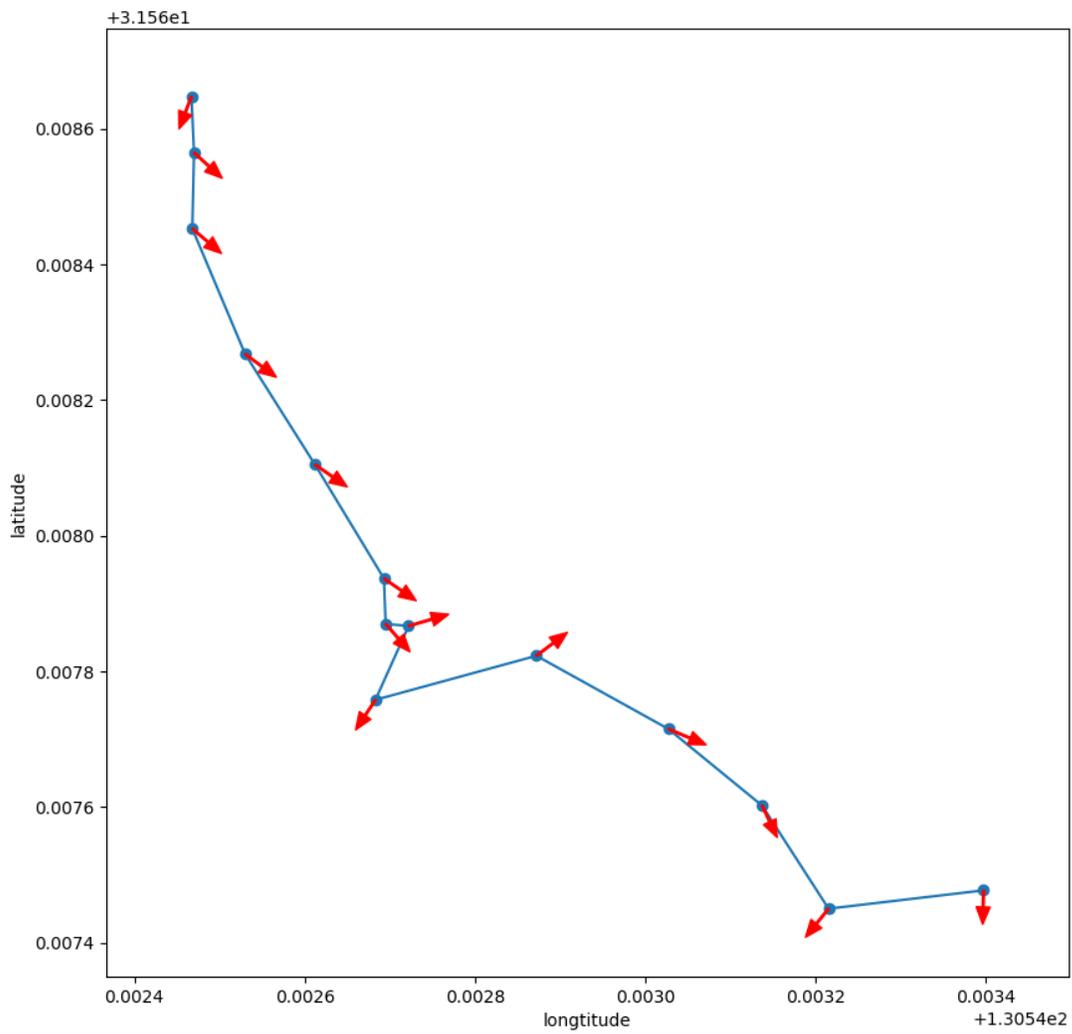


図 6-24 走行履歴 (矢印付きの図)

18. 通信機器電源 ON/OFF 試験 【済】

➤ 目的

通信機器の電源を ON/OFF できることを確認する。

➤ 試験/解析内容

マイコンから TWE-Lite にスリープ信号を送り，通信機器の電源を ON できることを確認する。また，マイコンから復帰信号を送り，通信機器の電源を OFF にできることを確認する。

➤ 結果

TWE-Lite の M3 ピン (27 番ピン) を HIGH/LOW 切り替えることで，通信機器の ON/OFF を切り替えることができる。実際に HIGH/LOW を切り替えて，動作を確認できた。実験の様子をリンク先 (<https://youtu.be/vyJ1r4e5laU>) に投稿している。

19. ゴール判定試験 【済】

➤ 目的

機体を走らせ、GPS 座標をもとにゴールから 10m 以内に近づいたときに機体がゴール判定できることを確認する。

➤ 試験/解析内容

フルサクセス・アドバンスドサクセスでの目標がゴールから 10m 以内に入ることであるため、GPS センサから所得した値をもとにゴールへ向かうプログラムを実装し、機体を走らせた。ゴール判定の確認はブザーを鳴らすことで確認した。

➤ 結果

結果を表 6-5 に示す。2 回試験を行い、ゴール判定できることを確認した。

表 6-5 ゴール判定試験の結果

	結果	リンク先
1 回目	GPS センサの値をもとにゴールに近づき（無線機に送られてきた情報で確認）、ゴール付近でゴール判定した（ブザー音で確認）	https://youtu.be/jcRyvMCOXgs
2 回目	GPS センサの値をもとにゴールに近づき（無線機に送られてきた情報で確認）、ゴール付近でゴール判定した（ブザー音で確認）	https://youtu.be/ulOzGMYbNwo

20. 電力試験 【済】

➤ 目的

ローバが競技に必要な時間動作できることを確認する。

➤ 試験/解析内容

本番で想定されるモータや通信機器に電力を供給して想定される駆動時間の間、電力が持つかどうか実際に動かして確認する。CanSat を走らせた状態で3時間耐久できることを確認する。

マイコン用のバッテリー、モータ・電熱線用のバッテリーをそれぞれ満充電した後、マイコン側では本番と同様の初期化处理、シリアル出力（無線出力）と microSD カードへの書き込みを行い、モータは PWM 出力値最大で回転させ続ける（前進の動作のみ）。この状態で過放電状態となる 3.5V 以下になる前までの時間を測定する。バッテリーの電圧は分圧してマイコンのアナログ入力ピンに接続して測定した。

図 6-25 は電力試験の様子である。

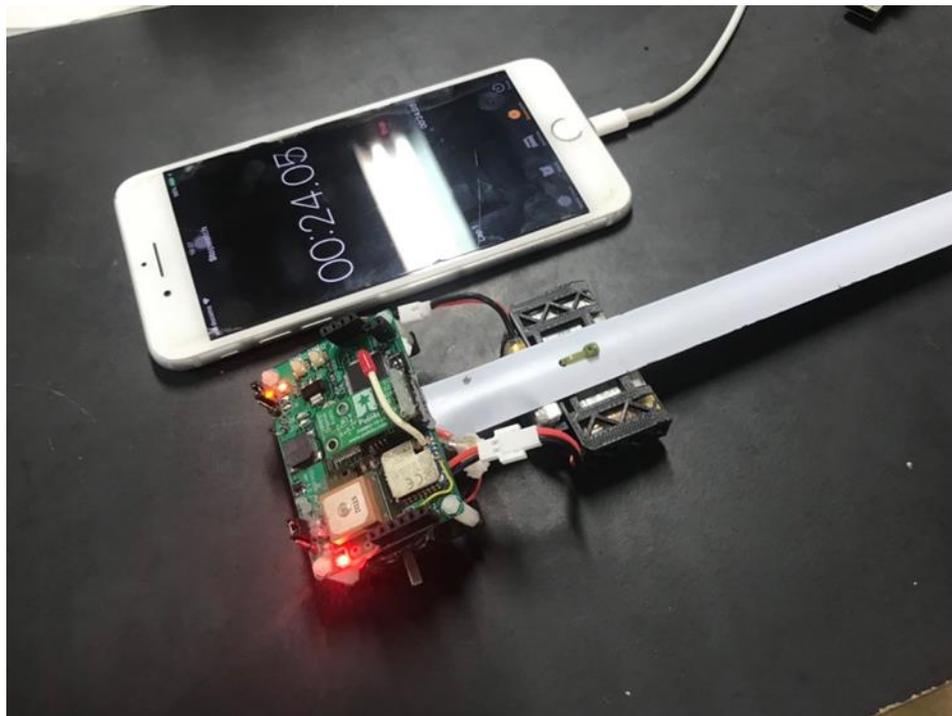


図 6-25 電力試験

【試験 1 回目】

条件：

マイコン側 (Li-Po, 1セル, 150mAh)

モータ・バッテリー側 (Li-Po, 1セル 800mAh)

試験の結果、マイコン用の電圧の方がモータ・電熱線用の電圧と比べて早く低下し、3.5V になるまでの時間は 29.9 分であった。マイコン用のバッテリーの電圧の変位、モータ・電熱線用のバッテリーの電圧の変位を図 6-26 に示す。

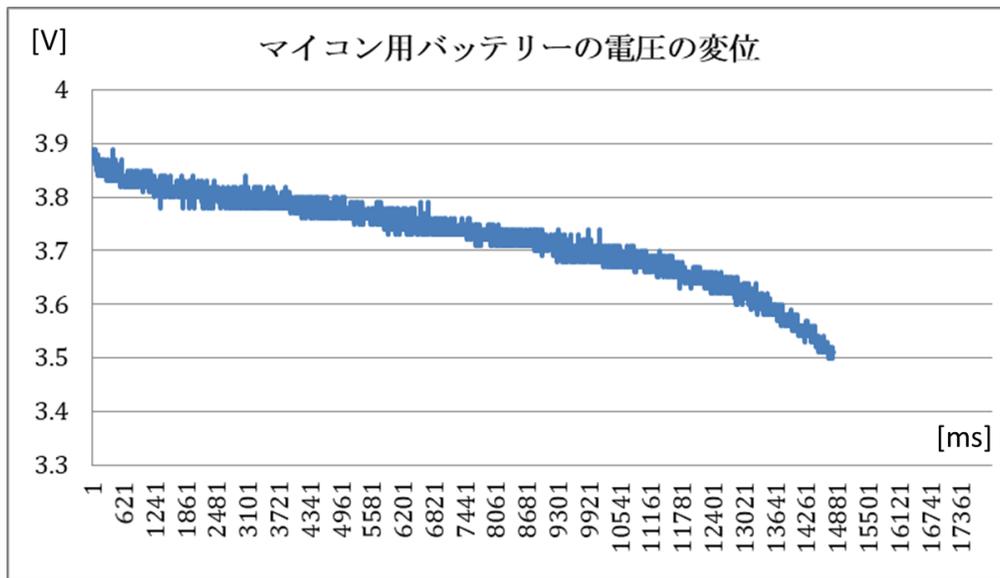


図 6-26 バッテリーの変位 (1)

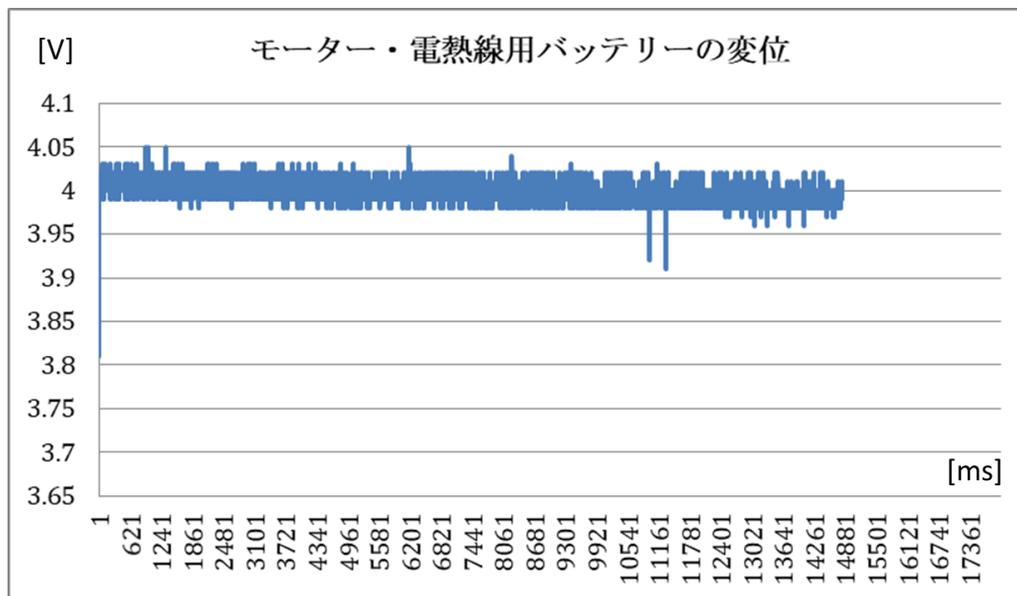


図 6-27 バッテリーの変位 (2)

【試験 2回目】

条件：

マイコン側 (Li-Po, 1セル, 240mAh)

モータ・バッテリー側 (Li-Po, 1セル 800mAh)

1回目の試験を受けて、マイコン用のバッテリーを容量 150mAh から 240mAh に変更して同様の実験を行ったところ、3.5V 以下になるまでの時間は 48 分になった。

【試験 その他】

ローバが陸上競技場の 100mトラックを何秒で走行できるか測定してローバの走行速度を求めた。走行には約 2 分 30 秒かかったため、 $100[m] \div 150[s] \doteq 2.4[km/h]$ となる。

2回目の電力試験の結果から 50 分程度走行できると考えると約 2 km走行できることが分かった。実験の様子をリンク先 (<https://youtu.be/Zv3jsgyfj-c>) に投稿している。

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. チーム内・審査会等

（チーム全体、班、各部署ごとの会議の頻度，開発メンバー以外の人を含めた審査会の有無や時期、回数を記載してください）

構想開始・・・2019年3月

設計開始・・・2019年4月

BBM(機能モデル)製作開始・・・2019年4月～

BBM(機能モデル)検証完了・・・～2019年6月

6月後半にプリント基板完成

8月中旬に改良基板完成

EFM(実物大モデル、負荷検証)完了・・・～2019年8月22日

2. 各担当（ハード・ソフト・全体などの進行状況・予定を記入）

メンバーの担当や進行状況，予定は Excel で管理している．共有リンクを以下に示す．

共有リンク↓

<https://drive.google.com/open?id=1CpDVepoVnz5MMswh4eWX49qUPx5QEDUW>

第8章 大会結果

1. 能代宇宙イベント

参加していないため、記載なし

2. ARLISS（上記能代と同様に記載してください）

➤ 目的

350ml クラスにおいて、「超小型 CanSat を 2 機搭載してランバックミッション」を行う。

➤ 結果

■ 打ち上げ1回目（2019年9月11日）

CanSat2 機のうち 1 機は上空でパラシュートが外れてしまい、自由落下。GPS センサが外れていたり、メジャーの拡張タイヤが折れていたりなど破損していた。もう一方の機体はパラシュートで軟着陸後、走行を開始したが、モータへの出力が止まらずループ状態に陥ってしまいリタイア。

破損した機体は microSD カードも破損していたため制御ログが残っていない。走行を開始した機体は制御ログが残っていたため、そちらの機体の制御ログを以下で説明する。

制御内容	成功/不成功	備考
打ち上げ検知	○	気圧センサによる検知
放出検知	○	照度センサによる検知
着地検知	○	気圧センサとジャイロセンサによる検知
パラシュート用電熱線への出力	○	
シェル展開用電熱線への出力	○	
ゴールまでの走行	△	ループ状態になり、モータが止まらなくなった

■ 打ち上げ2回目（2019年9月12日）

CanSat2 機ともに上空でパラシュートが外れ、自由落下。バッテリーが外れたり、メジャーの拡張タイヤが折れたりなど破損していた。

1機は microSD カードが折れ曲がってしまい、制御ログが残っていない。もう1機は microSD カードに制御ログが残っていた。制御ログから確認できた内容を以下に示す。

制御内容	成功/不成功	備考
打ち上げ検知	○	気圧センサによる検知
放出検知	○	照度センサによる検知
着地検知	○	気圧センサとジャイロセンサによる検知
パラシュート用電熱線への出力	×	マイコンから制御信号は出ていたが、モータ・電熱線用のバッテリーが外れてしまい出力されず
シェル展開用電熱線への出力	×	同上
ゴールまでの走行	×	同上

➤ 取得データ

■ 打ち上げ1回目 (2019年9月11日)

【システムログ】

青色の文字が microSD カードに記録されたログデータ

赤色の文字が提出用に書き足した説明文

<<<制御履歴ここから>>>

***** start program *****

[State: Initialization] ... Start CanSat Program!!!

Altitude OFFSET value: 1076.45 気圧センサのオフセット値を記録する (電源投入時の高度を基準 0m とするため)

Sleep ... Wake up!!! 30分のスリープ後起動した

[State: Launch detection] 打ち上げ検出モード (高度で判定, 予備でタイマー)

Altitude: 0.82 [m]

Altitude: 1.31 [m]

Altitude: 1.10 [m]

Altitude: 0.97 [m]

Altitude: 1.08 [m]

<<<長いので省略>>>

Altitude: 2.58 [m]

Altitude: 5.34 [m]
Altitude: 9.84 [m]
Altitude: 14.15 [m]
Altitude: 17.80 [m]
Altitude: 26.36 [m]
Altitude: 33.40 [m]
Altitude: 44.94 [m]
Altitude: 41.85 [m]
Altitude: 43.87 [m]
Altitude: 48.84 [m]
Altitude: 50.07 [m]
Altitude: 53.77 [m]
Altitude: 63.77 [m]
Altitude: 63.48 [m]
Altitude: 70.81 [m]
Altitude: 74.18 [m]
Altitude: 81.30 [m]
Altitude: 89.68 [m]
Altitude: 97.73 [m]
Altitude: 111.71 [m]
Altitude: 122.51 [m]
Altitude: 129.94 [m]
Altitude: 140.99 [m]
Altitude: 154.49 [m]
Altitude: 163.72 [m]
Altitude: 180.61 [m]
Altitude: 199.47 [m]
Altitude: 222.10 [m]
Altitude: 251.38 [m]
Altitude: 296.82 [m]
Altitude: 326.88 [m]
Altitude: 354.65 [m]
Altitude: 387.53 [m]
Altitude: 412.13 [m]
Altitude: 436.55 [m]
Altitude: 459.93 [m]
Altitude: 482.66 [m]
Altitude: 512.36 [m]
Altitude: 534.29 [m]
Altitude: 556.54 [m]
Altitude: 585.63 [m]
Altitude: 606.29 [m]
Altitude: 627.57 [m]
Altitude: 655.59 [m]

Altitude: 696.99 [m]
Altitude: 717.82 [m]
Altitude: 737.06 [m]
Altitude: 755.86 [m]
Altitude: 775.59 [m]
Altitude: 793.47 [m]
Altitude: 819.07 [m]
Altitude: 837.96 [m]
Altitude: 862.31 [m]
Altitude: 881.03 [m]
Altitude: 899.53 [m]
Altitude: 924.58 [m]
Altitude: 942.41 [m]
Altitude: 960.60 [m]
Altitude: 978.44 [m]
Altitude: 995.87 [m]
Altitude: 1018.94 [m]
Altitude: 1035.91 [m]
Altitude: 1053.16 [m]
Altitude: 1075.61 [m]
Altitude: 1092.37 [m]
Altitude: 1114.59 [m]
Altitude: 1130.61 [m]
Altitude: 1147.06 [m]
Altitude: 1163.46 [m]
Altitude: 1178.77 [m]

[!] launch detection judged by altitude 高度が 1000m を超えたあたりで打ち上げを検出した

[State: Release detection] 放出判定 (照度センサ) のモードに入る

***** start program ***** ここで放出時の衝撃によりマイコンがリブートした

[State: Initialization] ... Start CanSat Program!!!

***** start program *****

[State: Initialization] ... Start CanSat Program!!!

Altitude OFFSET value: 3587.19 リブートしたため高度 3500m 付近をオフセットとして記録してしまっている

[State: Release detection] しかし、SD カードに以前のモード (ここでは放出検知モード) を記録しているため、そこから処理を再開している

[!] release detection judged by cds 照度センサの値により放出を検知した

[State: Drop detection] 落下判定モード (気圧センサ, ジャイロセンサ) に入った

Altitude: -15.56 [m]

Altitude: -15.81 [m]

Altitude: -16.29 [m]
Altitude: -16.73 [m]
Altitude: -17.23 [m]

<<<長いので省略>>>

Altitude: -1025.74 [m]
Altitude: -1026.40 [m]
Altitude: -1027.24 [m]
Altitude: -1027.12 [m]
Altitude: -1027.75 [m]
Altitude: -1028.79 [m]
Altitude: -1029.26 [m]
Altitude: -1030.06 [m] リポートした関係で上空の高度をオフセット値として記録してしまったため、高度がマイナスの値となっている（高度 3500m 付近を基準 0m としている）
Altitude: -1030.12 [m]
Altitude: -1031.21 [m]
Altitude: -1032.04 [m]
Altitude: -1032.05 [m]

<<<長いので省略>>>

Altitude: -2503.12 [m]
Altitude: -2503.29 [m]
Altitude: -2503.76 [m]
Altitude: -2504.45 [m]
Altitude: -2505.13 [m]
Altitude: -2505.49 [m]
***** start program ***** おそらく着地の衝撃で再びマイコンがリポートしてしまった
[State: Initialization] ... Start CanSat Program!!!
Altitude OFFSET value: 1077.70

[State: Drop detection] 以前のモード（落下判定のモード）から処理を再開した
Altitude: 0.16 [m]
Altitude: 0.52 [m]
Altitude: 0.12 [m]
Altitude: 0.20 [m]
Altitude: 0.08 [m]
Altitude: 0.42 [m]
Altitude: 0.22 [m]
Altitude: 0.19 [m]
Altitude: 0.34 [m]

Altitude: 0.08 [m]

Altitude: 0.54 [m]

[!] drop detection judged by gyro and altitude 高度とジャイロセンサの値から着地を検知した

[State: First fire] パラシュート巻取り機構を動作させるために電熱線に出力した

[State: Second fire] シェルを展開するために電熱線に出力した

シェル展開後、パラシュートから離れるために前進するが、前進した状態で処理がループしてしまいモータが回転し続けた（CanSat 発見時の様子から推定）

<<<制御履歴ここまで>>>

【生データログ】

データそのものは「log.csv」参照。ここでは、csv ファイルから作成したグラフについて説明する。

図 8-1 は照度センサから所得した値と気圧センサの値をもとに計算した高度の値のグラフである。グラフから高度 2500m 付近で CanSat がキャリアから放出され、照度センサの値が低くなったことがわかる。

図 8-2 は加速度センサから所得した値（xyz 軸成分）をもとに計算して求めた G と気圧センサの値をもとに計算した高度の値のグラフである。グラフから打ち上げの瞬間に約 16G の力が CanSat に加わったことがわかる。

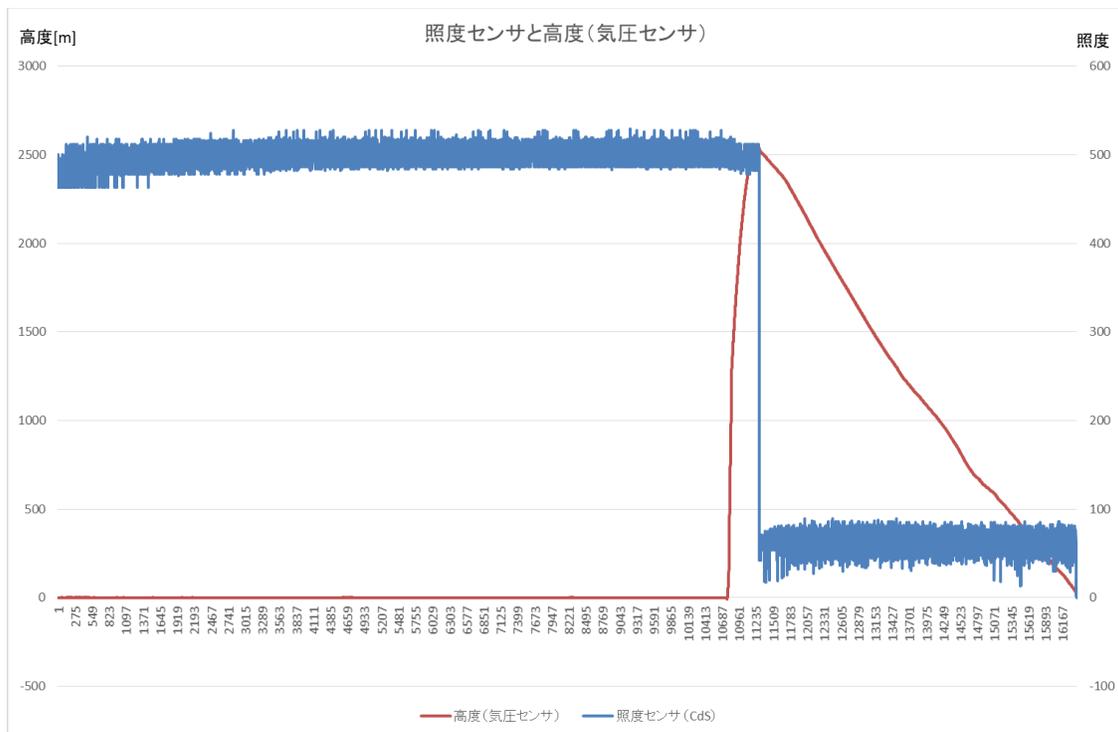


図 8-1 照度センサと高度 (気圧センサ)

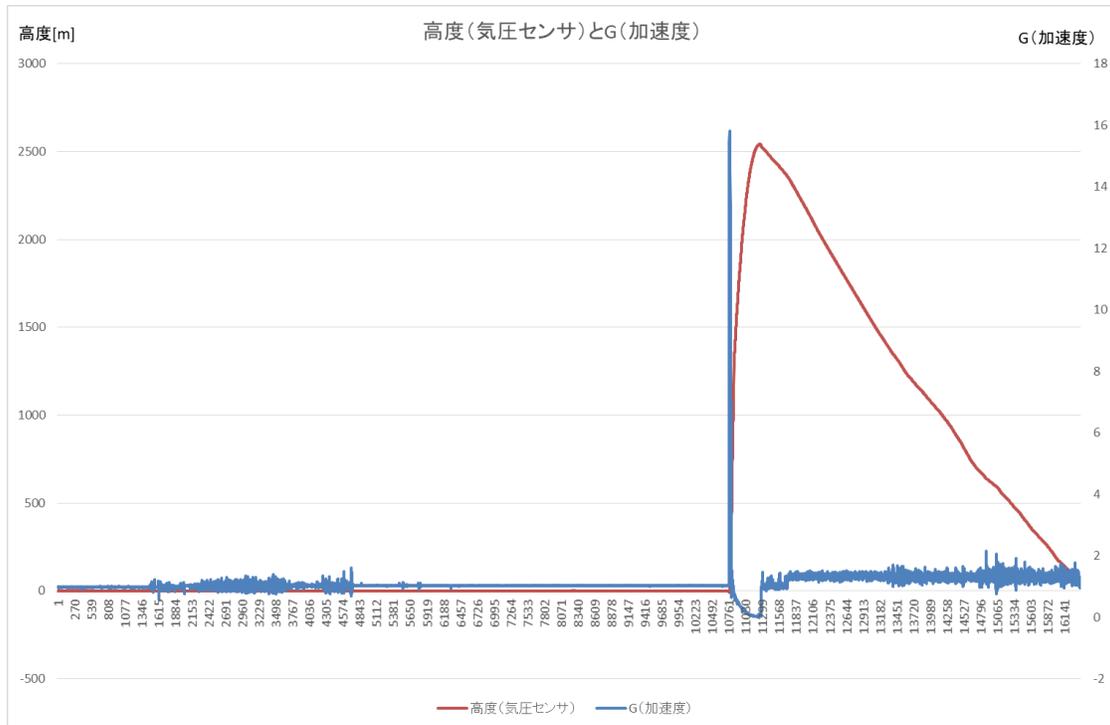


図 8-2 高度 (気圧センサ) と G (加速度)

■ 打ち上げ 2 回目 (2019 年 9 月 12 日)

【システムログ】ファイル名 : system_log.txt

青色の文字が microSD カードに記録されたログデータ

赤色の文字が提出用に書き足した説明文

<<<制御履歴ここから>>>

***** start program *****

[State: Initialization] ... Start CanSat Program!!!

Altitude OFFSET value: 1046.14 気圧センサのオフセット値を記録する (電源投入時の高度を基準 0m とするため)

Sleep ... Wake up!!! 30 分のスリープ後起動した

[State: Launch detection] 打ち上げ検出モード (高度で判定, 予備でタイマー)

Altitude: 0.04 [m]

Altitude: 0.05 [m]

Altitude: 0.30 [m]

Altitude: 0.14 [m]

Altitude: 0.20 [m]

<<<長いので省略>>>

Altitude: 2.87 [m]

Altitude: 4.64 [m]
Altitude: 8.22 [m]
Altitude: 13.47 [m]
Altitude: 17.97 [m]
Altitude: 27.63 [m]
Altitude: 35.29 [m]
Altitude: 45.75 [m]
Altitude: 46.57 [m]
Altitude: 46.63 [m]
Altitude: 51.94 [m]
Altitude: 65.02 [m]
Altitude: 77.34 [m]
Altitude: 83.58 [m]
Altitude: 86.18 [m]
Altitude: 98.68 [m]
Altitude: 118.03 [m]
Altitude: 130.21 [m]
Altitude: 139.98 [m]
Altitude: 150.34 [m]
Altitude: 171.17 [m]
Altitude: 180.19 [m]
Altitude: 195.30 [m]
Altitude: 206.59 [m]
Altitude: 217.47 [m]
Altitude: 232.48 [m]
Altitude: 244.31 [m]
Altitude: 265.36 [m]
Altitude: 281.30 [m]
Altitude: 297.09 [m]
Altitude: 311.81 [m]
Altitude: 338.97 [m]
Altitude: 372.11 [m]
Altitude: 425.92 [m]
Altitude: 454.62 [m]
Altitude: 482.44 [m]
Altitude: 518.47 [m]
Altitude: 545.13 [m]
Altitude: 570.78 [m]
Altitude: 605.24 [m]
Altitude: 630.22 [m]
Altitude: 654.91 [m]
Altitude: 679.52 [m]
Altitude: 703.82 [m]
Altitude: 743.08 [m]

Altitude: 773.96 [m]
Altitude: 796.88 [m]
Altitude: 819.33 [m]
Altitude: 842.16 [m]
Altitude: 863.93 [m]
Altitude: 885.28 [m]
Altitude: 913.51 [m]
Altitude: 934.96 [m]
Altitude: 956.19 [m]
Altitude: 976.84 [m]
Altitude: 1011.40 [m]
Altitude: 1031.78 [m]
Altitude: 1051.91 [m]
Altitude: 1078.74 [m]
Altitude: 1098.59 [m]
Altitude: 1118.38 [m]
Altitude: 1144.70 [m]
Altitude: 1164.53 [m]
Altitude: 1183.73 [m]
Altitude: 1209.18 [m]

[!] launch detection judged by altitude 高度が 1000m を超えたあたりで打ち上げを検出した

[State: Release detection] 放出判定（照度センサ）のモードに入る

[!] release detection judged by cds 照度センサの値が閾値を下回ったため、放出を判定した

[State: Drop detection] 落下判定モード（気圧センサ、ジャイロセンサ）に入った

Altitude: 2709.61 [m]
Altitude: 2704.14 [m]
Altitude: 2703.76 [m]
Altitude: 2704.08 [m]
Altitude: 2700.70 [m]
Altitude: 2698.91 [m]
Altitude: 2697.55 [m]
Altitude: 2695.43 [m]
Altitude: 2694.39 [m]
Altitude: 2686.73 [m]
Altitude: 2684.23 [m]
Altitude: 2680.96 [m]
Altitude: 2685.74 [m]
Altitude: 2685.20 [m]
Altitude: 2679.23 [m]
Altitude: 2678.30 [m]
Altitude: 2670.47 [m]
Altitude: 2667.21 [m]
Altitude: 2667.34 [m]
Altitude: 2665.04 [m]

<<<長いので省略>>>

Altitude: 0.82 [m]

Altitude: 0.69 [m]

Altitude: 0.78 [m]

Altitude: 0.95 [m]

Altitude: -1.21 [m]

[!] drop detection judged by gyro and altitude 高度とジャイロセンサの値から着地を検知した

[State: First fire] パラシュート巻取り機構を動作させるために電熱線に出力した

[State: Second fire] シェルを展開するために電熱線に出力した

[State: Run to GOAL] GPS センサの値をもとにゴールへ走行するモードに入った（モータ側の電池が外れてしまっていたため、走行できず）

【生データログ】

データそのものは「log.csv」参照。ここでは、csv ファイルから作成したグラフについて説明する。

図 8-3 は気圧センサの値をもとに計算した高度と照度センサから所得した値のグラフである。グラフから高度 3000m 付近で最大高度に達したことがわかる。またその際、照度センサの値が変化していることから、CanSat がキャリアから放出されたことがわかる。

図 8-4 は気圧センサの値をもとに計算した高度と加速度センサから所得した値 (xyz 軸成分) をもとに計算した G の値のグラフである。

グラフから、高度が上昇するタイミング (打ち上げ時) に約 15G, 最高高度に達したタイミング (放出時) に約 25G, 高度が再び 0m に達したタイミング (着地時) に約 20G の衝撃が CanSat に加わったことがわかる。

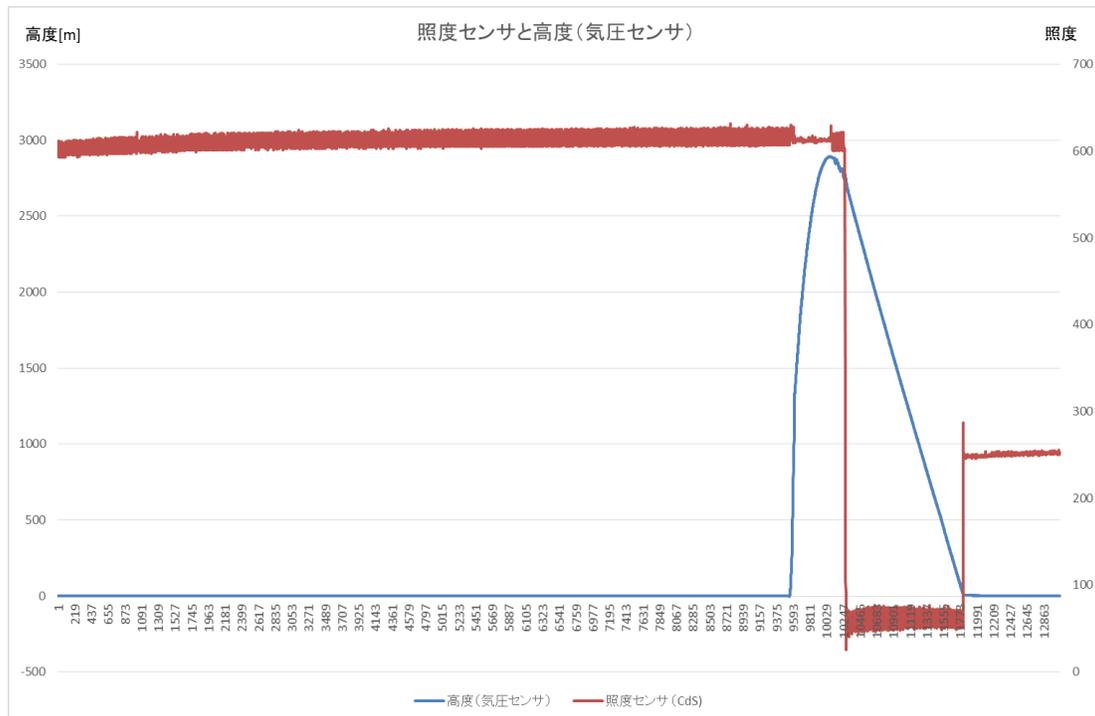


図 8-3 照度センサと高度 (気圧センサ)

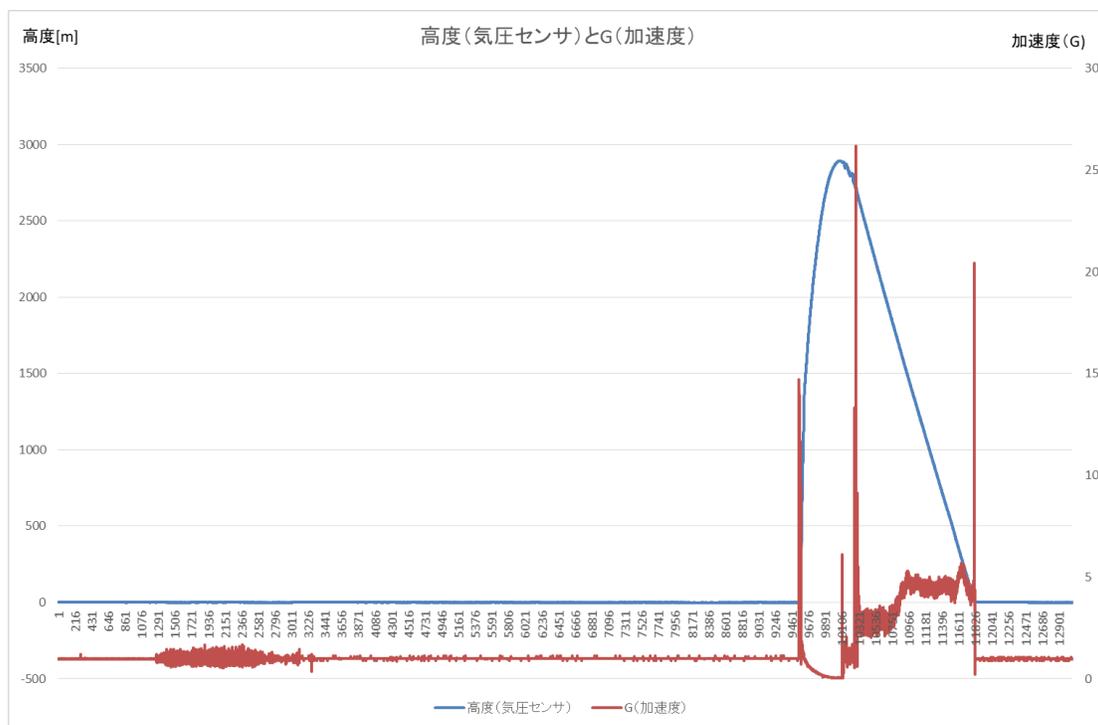


図 8-4 高度 (気圧センサ) と G (加速度)

➤ 故障原因解析・解決手段等

2回の打ち上げを行い、全部で4機のCanSatの投下を行った。4機のうち走行（ミドルサクセス）まで達成した機体は1機のみであった。しかし走行を開始したCanSatはモータが止まらなくなってしまい、リタイアとなった。

フルサクセスまで至らなかった要因として2点考えられる。

1点目は車上荒らしによって、本番用のCanSat3機とCanSatに搭載する予定だったLi-Poバッテリーすべてを失ってしまった点である（図8-5、図8-6参照）。これにより、予備で持ってきていた基板で大会に臨むこととなった。また、バッテリーに関しては、国内で実験に使用したものと同じバッテリーを現地で調達することができず、急遽購入したリチウム乾電池と他大学チームからお借りしたLi-Poバッテリーで代用することとなった。

その結果、CanSatのモータを駆動した際にモータが止まらなくなるという現象が発生した。これは電源の変更に伴いバッテリー電圧が高くなったことで、マイコンの挙動に影響を与えたためだと考えられる。

解決手段は車上荒らしに遭わないことである。そのために貴重品は常に携帯する、車を離れるときは1人以上車に残るようにするなどの対策が考えられる。

2点目は、パラシュートの強度不足である。4機打ち上げたうち、3機のパラシュートが放出時の衝撃によって外れ、自由落下してしまった。本番のログにおいて放出時におよそ25Gの衝撃がCanSatに加わっていることがわかった。それだけの衝撃に耐えられるパラシュートを作成できていなかったことが失敗の原因である。

しかし、今回国内で行った放出時衝撃試験では40Gを想定して実験していた。そのため、25Gの衝撃には耐えられるはずである。どうして衝撃に耐えることができず、パラシュートが外れてしまったのかさらに考察する。

今回本番で使用したパラシュートは、車上荒らしの被害に遭わなかった予備のパラシュートであった。放出時衝撃試験に使用したパラシュートは本番用のパラシュート（車上荒らしで盗られたパラシュート）で、予備のパラシュートはそれ以前に作成したものであった。そのため、古い製作過程で作成したパラシュートであったため25Gの衝撃に耐えられなかった可能性があると考えられる。

解決策は、本番用と予備のパラシュートの管理を適切にして、試験をクリアしたパラシュートを使用できるようにすることである。



図 8-5 車上荒らしで窓を割られたレンタカー¹

¹ サンフランシスコ アラモ・スクエアにて、公園のトイレに立ち寄ったおよそ10分間で起きた惨劇である。ここですべてを失った…。貴重品は常に携帯しないとイケない。



図 8-6 荒らされた車内

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）

■ ハード

➤ ボディ・シェル

3D プリンタを用いることで、モータの治具やスタビライザーの治具もボディと一体で製作することができ、軽量・小型化することができた。また、複雑な形状を容易に製作することができたため、細かい点の修正や肉抜きなどを実物で試すことができた点良かった。また、半自動で部品を製作してくれるため、多数の機体や予備部品を製作することに大きく寄与した。

一方、ギリギリまで軽量化を目指したため、フィラメント（3D プリンタの材料）を色付きのものに変えると強度不足になるといった軽微な変更でのトラブルに悩まされることも多かった（素材は変更していない、色などで微妙な特性の違いがある）。また、素材を3D プリンタに頼りすぎた点も良くなかった。

FDM 型の 3D プリンタで製作したものはどうしても積層方向にもろくなるため、大会を終えてから考えれば、製作部位によっては 3D プリンタ以外で製作した方が時間的にも強度的にも明らかに良かったであろう部分も思い当たる。

総合して考えると、3D プリンタを導入したことで得られたメリットは大きかったが、3D プリンタに頼りすぎずに、従来の方法も忘れずに組み合わせて製作することが必要となる。

➤ 回路

小型化を達成するためにプリント基板にして表面実装部品を多用する回路設計にした。小型化により、基板サイズは 40mm×60mm という非常に小さいサイズとなった。開発には KiCAD を使用し、基板の製造は Elecrow に発注した。発注は全部で 3 回行い、試験で判明した改良点を段階的に修正していった。

1 回目の振動試験で microSD カードが外れやすいことが分かった。そこで microSD カードのスロット位置を工夫した。また、シェル収納時にメジャーと干渉しないよう、電源スイッチの位置を基板の端から中央付近に変更した。

■ ソフト

メンバー全員が扱ったことのある言語（C 言語）と開発環境（Arduino IDE）を採用することで、メンバー全員がソースコードを理解でき、開発もできるように工夫した。

また、GitHub にソースコードをアップロードすることですべてのメンバーが参照しやすいようにした。しかし、Git を使った開発に不慣れだったため、バージョン管理機能やブランチを切って複数人で開発といった活用はできなかった。

ファイル分割（main.c や motor.c などにソースコードを分ける）を行うことでプログラムを見やすく工夫した。また、main 関数の行数が不必要に増えることのないように工夫した。その一例として、列挙型と switch-case 文を使って状態遷移していくプログラムにすることでローバの処理を分かりやすくし、各 case 文で実行する処理もほかのファイルに関数で書くことで工夫した。

2つの LED の点灯パターンとスピーカのメロディによって実行しているプログラムがどこであるか外部からわかるように工夫した。また、起動時にセンサが読み込めなかったり、microSD カードが挿さっていなかったりするとエラー音を鳴らすようにした。これにより、実験や本番で microSD カードの挿し忘れ等のミスを回避することができた。

■ マネジメント

設計・加工・回路・プログラムなど役割をメンバーで分担して行った。また、複数機体を使ったミッションであることの利点を活かして、あるメンバーは機体 A を使って落下実験、あるメンバーは機体 B を使って走行試験というように同じタイミングで異なる試験を行うことができた。

スケジュール管理の面でも、大学の課題や試験などの各自のスケジュールをある程度事前に考慮しながら、計画を立てることができた点が良かった。

また、初めての海外ではあったが、パスポート関係の申請や ESTA・国際免許・保険の申請等をメンバー全員で共有しながら出国までは問題なく進めることができた点は良かった。

振動試験設備や EtoE 試験用のドローンレンタルなどの試験設備に関しても、メンバー間で分担しながら手配できた点が良かった。

2. 良かった点・課題点

今回、種子島ロケットコンテストで以前から開発していた小型 CanSat をベースに開発を進めた。メジャーを使った展開車輪やプリント基板による小型化を継承しつつ、メジャーの種類や取り付け角度を変更して強度を高めたり、バッテリー容量の変更を行って駆動時間を長くし

たり工夫した。そのため、ARLISS という規模の大きい実験でも安全面をクリアできるような機体に仕上げることができた点が良かった。

3. チームのマネジメント等、プロジェクト全体でのよかった点、反省点

プロジェクト全体の成果として、参加チームの中で唯一 350ml クラスに出場し、2機ローバによるランバックというミッションを掲げて、現地で2回の打ち上げを行うことができた。結果は投下した機体計4機のうち1機がパラシュートを切り離し走行するまでに至った。また、microSD カードに制御ログを記録し解析することもできた。

サンフランシスコで車上荒らしの被害を受けて、本番用の CanSat や使用するすべてのバッテリーなど多くを失ったが、他大学のチームからバッテリーをお借りしたり、現地で購入できる電池で代用したりすることで実験を行うことができた。国内でしっかりと準備を行い、予備の部品を複数用意していた点や各メンバーがあきらめずホテルや砂漠で修正作業を行った点が成果にあらわれたと思う。

今回の ARLISS 参加で海外旅行保険に加入しておくことの大切さを学んだ。海外では日本と同じ感覚で過ごしてはいけない。もし実際に被害に遭った場合、盗られたクレジットカード等が悪用されないようにクレジットカード会社や保険会社にすぐさま電話で連絡する必要がある。ポリスレポート²は必ずもらっておくこと。また、海外旅行先でも日本国内に繋がる電話を所持しておくことは非常に重要である。

鹿児島大学からは初参加だったが、服部先生にリモートで多くのご指導をいただき、メンバー間で協力して開発・実験を進められた点が良かった。

スケジュール管理に甘い点が多く、安全試験を締め切り直前に行うこととなってしまったことは反省点である。

² 海外旅行先でどのような被害に遭ったかの証明書。現地の警察に被害状況を伝え発行してもらう。これがないと保険が適応されない可能性もある。

謝辞

ARLISS 参加にあたり、ご指導いただいた東京工科大学 服部聖彦教授に心から感謝いたします。

安全審査を行うにあたり、試験設備等のご協力いただきました鹿児島県工業技術センターの皆様、株式会社エルム・スカイアクションの皆様にご感謝いたします。

現地にて、盗まれてしまったバッテリー等の代わりに部品をお貸しして下さった他チームの皆様、そして、打ち上げにあたり、ご協力いただいたエアロパックの皆様にご感謝いたします。

ARLISS にて打ち上げを行うことができたのは、多くの方の支えとご指導によるものです。この場を借りて御礼を申し上げます。