



ARLISS 2017

Post Flight Review

The University of Tokyo
TEAM THE-O
Comeback Competition

Project Management

- Slackでコミュニケーション
- Google Apps ScriptsとGitHub Issuesで進捗管理
- GitHubでコード共有
- Google Drive(東大ECCSアカウントだと容量無制限)でドキュメント共有



ランバック部門優勝(詳細は割愛)

➤ 1回目 0.00m

➤ 2回目 0.08m



Mission Summary

Minimum Success

- パラシュートを展開してローバーを安全に着陸させる。
- 着地を判定してケースから展開させる。

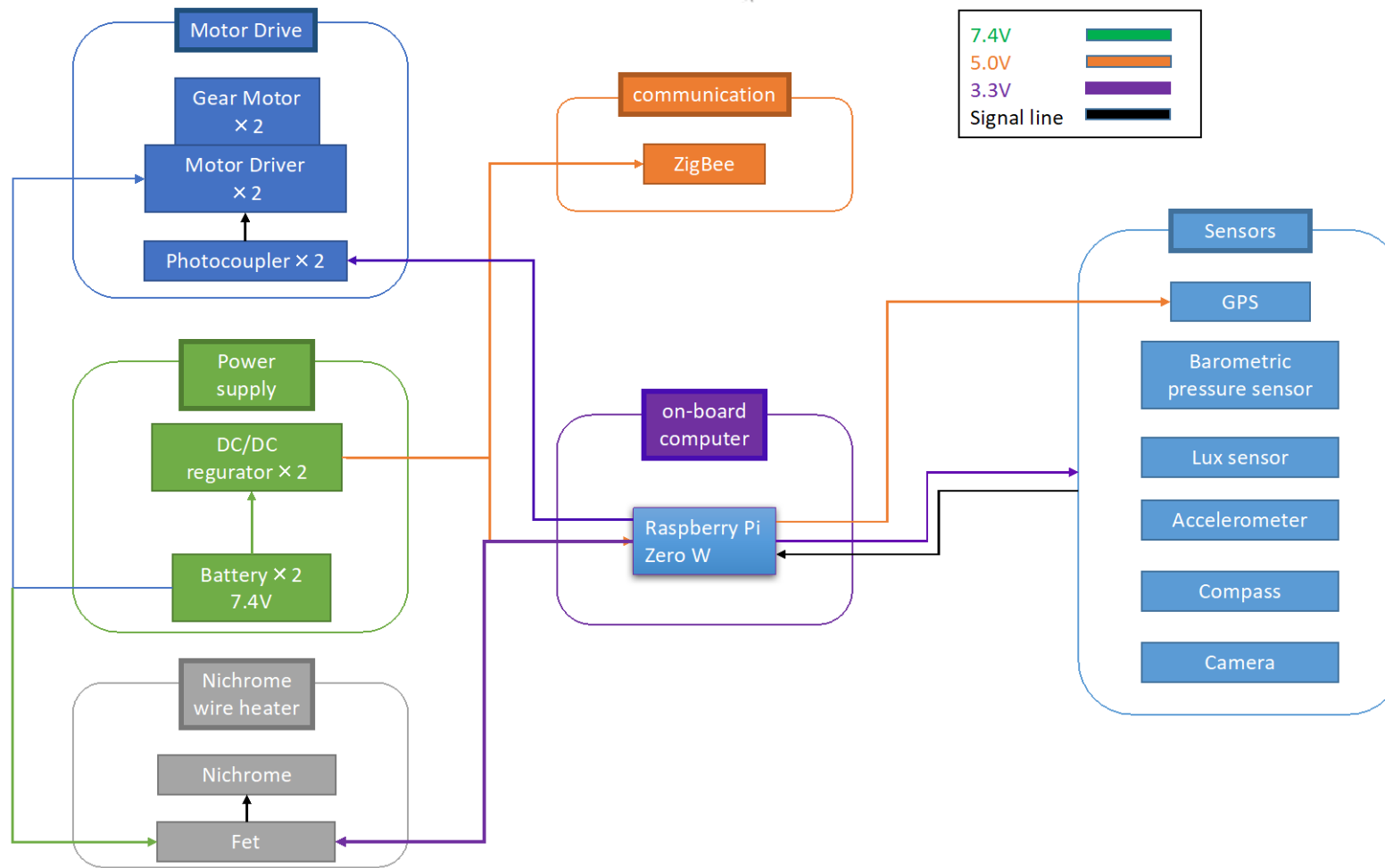
Full Success

- GPSなどを用いてゴールから50m以内まで行く

Extra Success

- カメラを用いてゴールから3m以内へ行く

Circuit Diagram

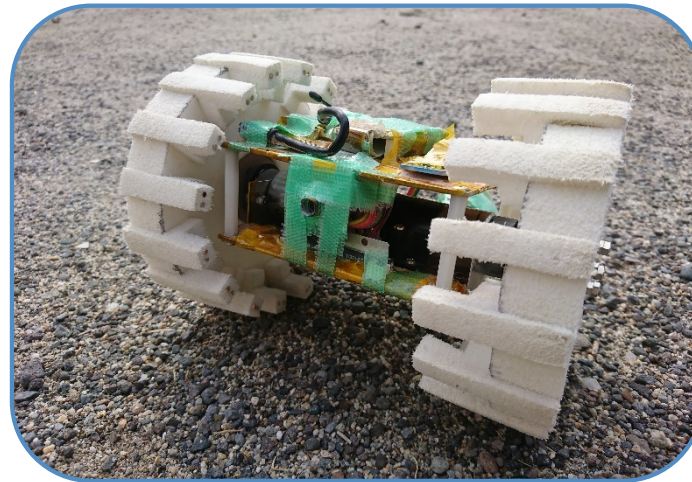


Physical Layout



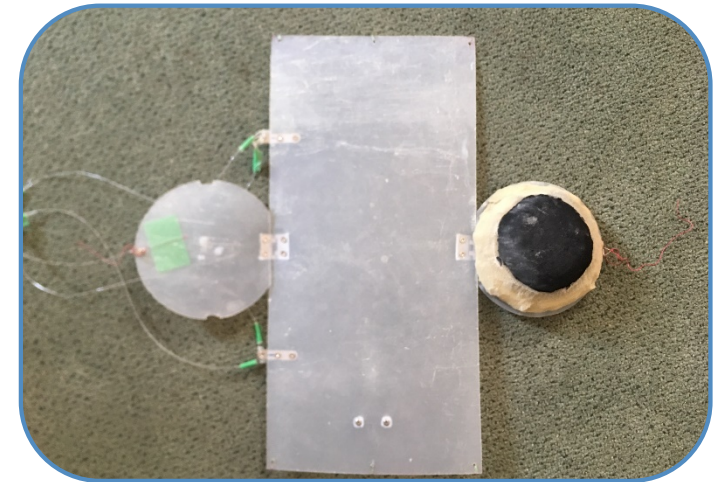
Parachute

- Weight 65g
- Rip Stop Nylon Fabric
- Nylon Line



Rover

- Weight 770g
- Copper Plate
- Expanded Polyethylene Tire
- Steel Flange



Container

- Weight 175g
- Polypropylene Plate
- Polyethylene Line

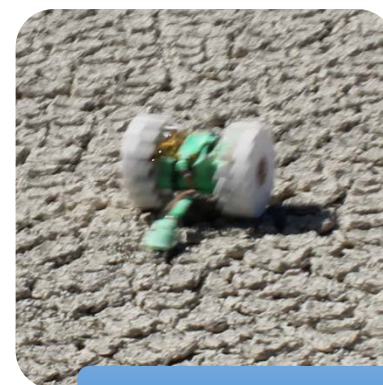
PLANNED Sequence of Events



Launch



Land and deploy a parachute



GPS and compass navigation

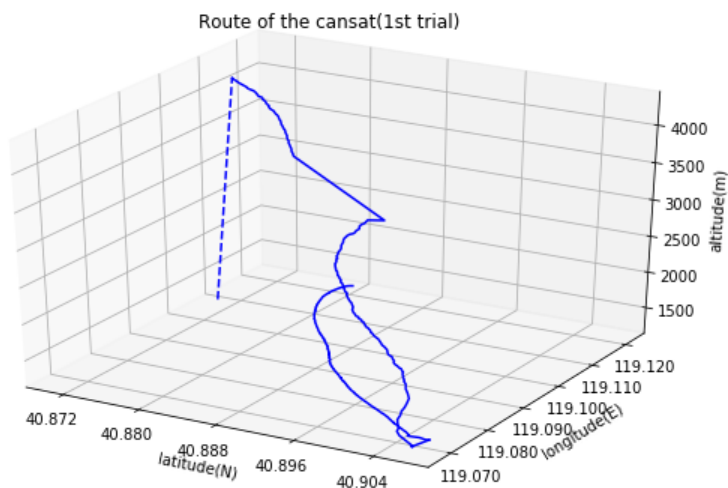


Navigation using camera

Results



1st Launch (1.34m)



The rover ran 5.2km

Time [s]	Event
0	Tue Sep 12 12:17:42 2017 プログラム起動
1019	ロケットからの放出判定
1935	着地判定&ケーシング展開
1939	地上走行開始
7935	ゴールから3m以内で停止 ->画像処理シーケンス
8078	ゴール判定->制御終了

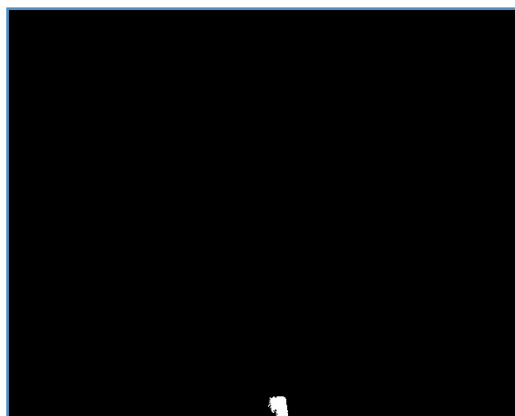
Analysis

- 放出判定->着地判定->ケーシング展開は問題なく成功
- ログを解析するとケーシング展開後すぐにスタック判定
 - ケーシングか何かにスタックしただろうと判断(後述)
しかしすぐ復帰して,その後はゴールへ制御開始
- 1.34m→ゴール判定が早かった
 - カメラの取り付け角度が水平でなく上向き
 - 太陽が写真の中に入る
 - 逆光の時のコーンも認識するようにHSVパラメーターを調整すると背景の山の色もとる
 - 山の方に行かないようにゴール判定を早めに設定してた

Analysis



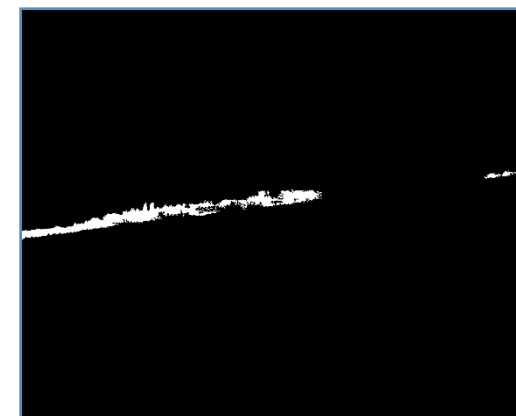
逆光の時のコーン



二値化画像

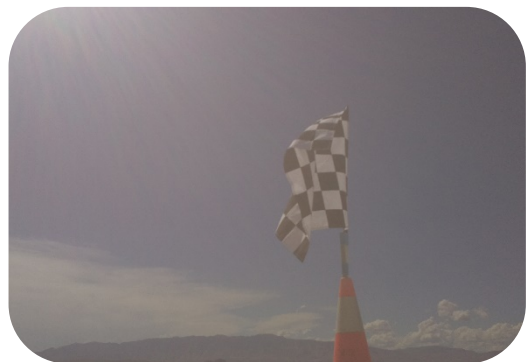


山

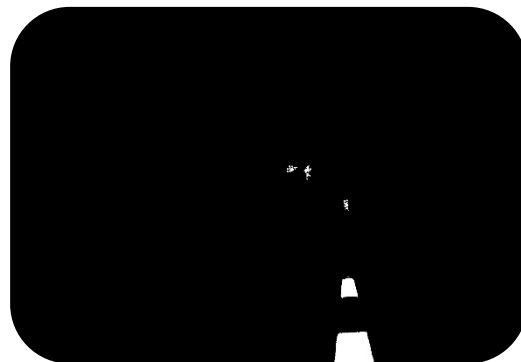


二値画像

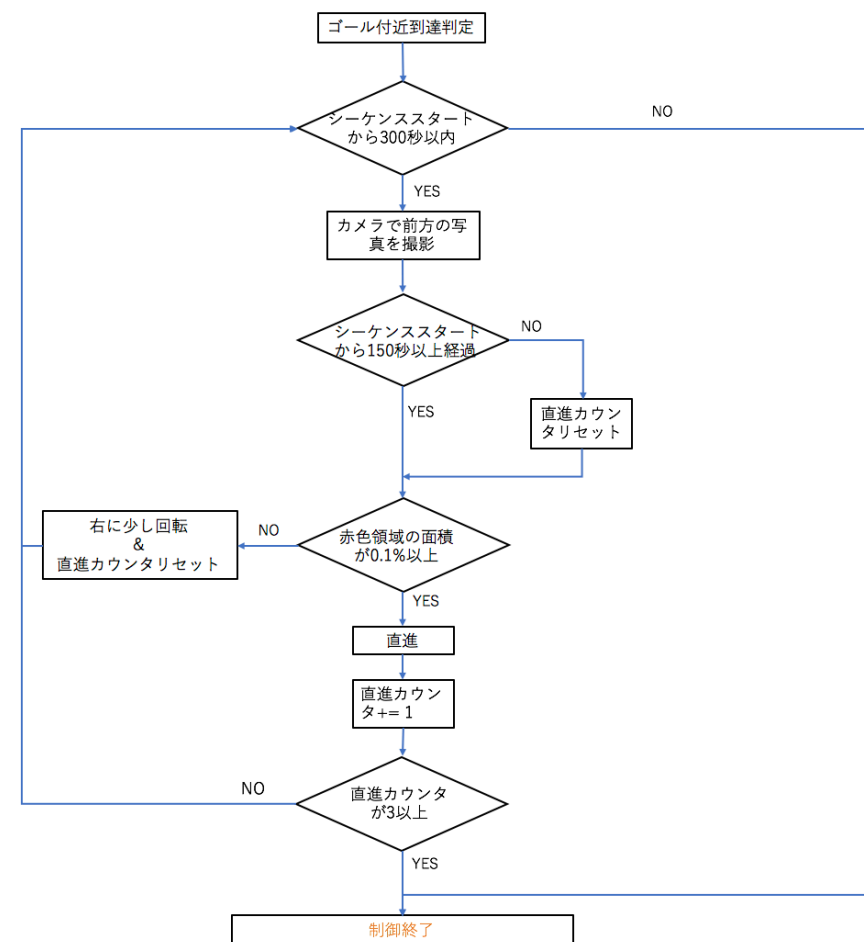
画像処理



Raw Image

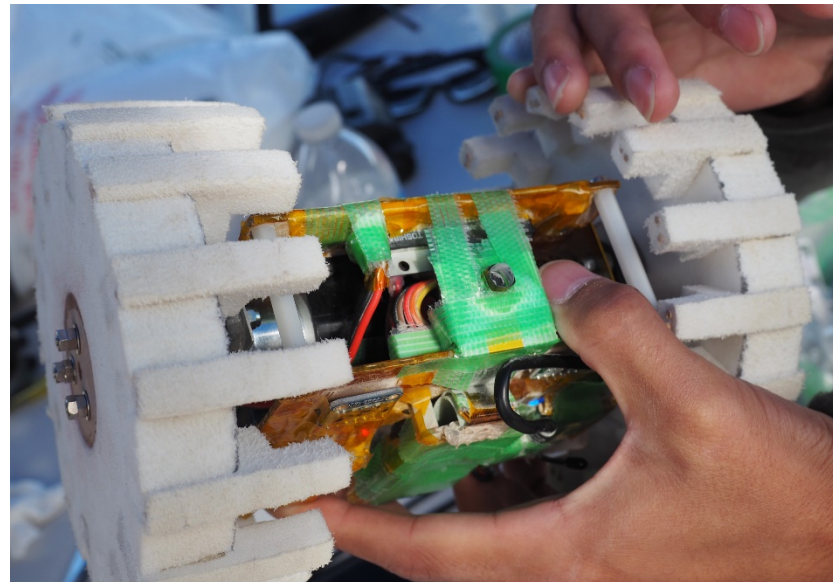


Binary Image



Improvement

- 背景の山の赤を取らないように逆光の時にコーンの認識を諦める
- カメラにサンシェードを追加する



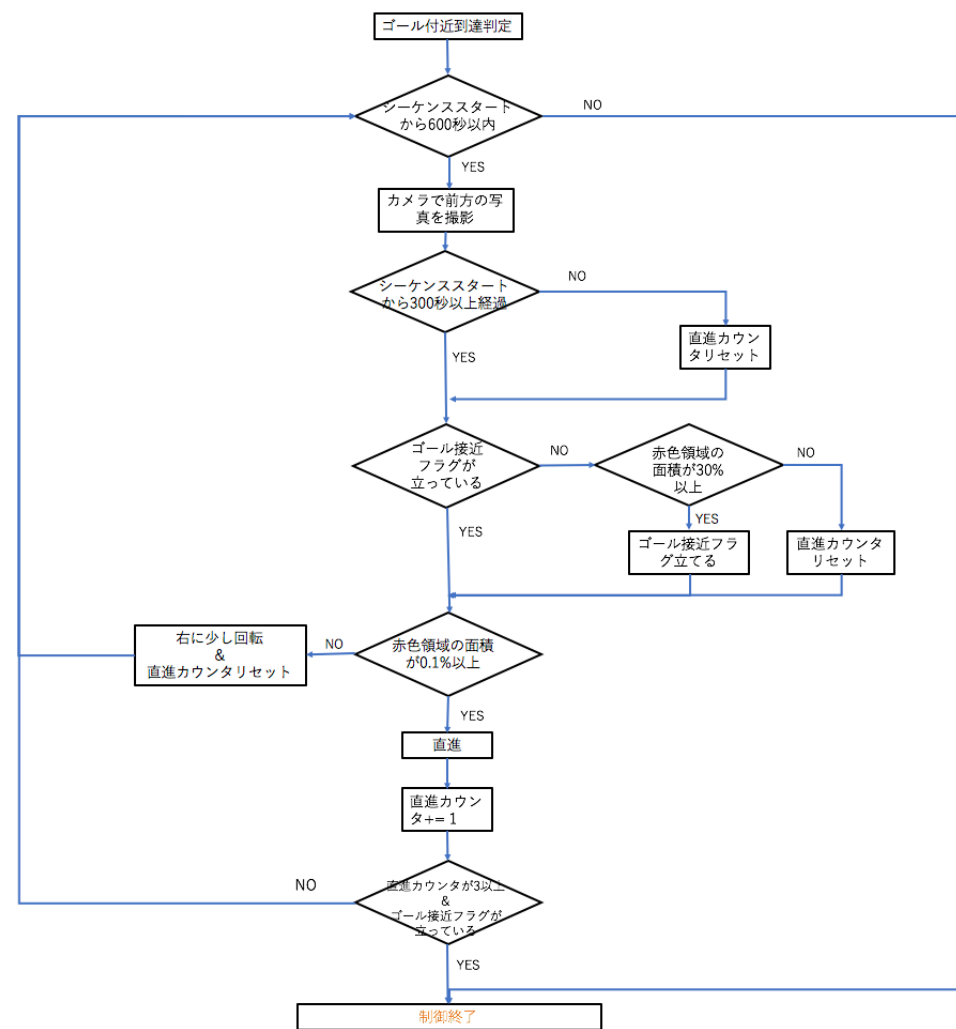
サンシェード付きのカメラの様子

Improvement

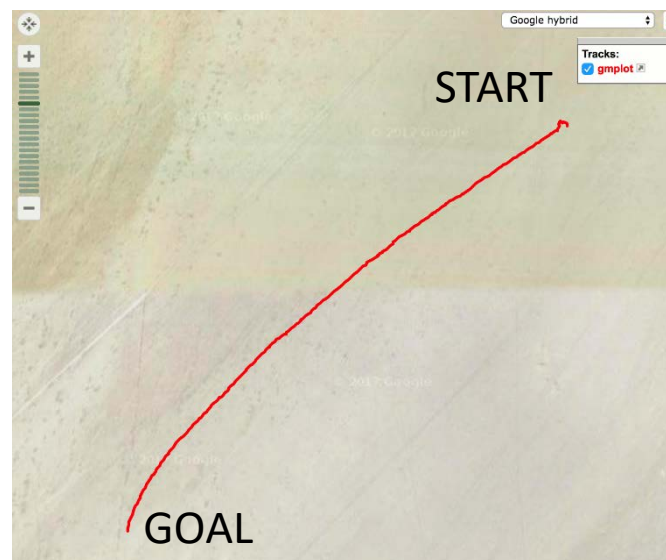
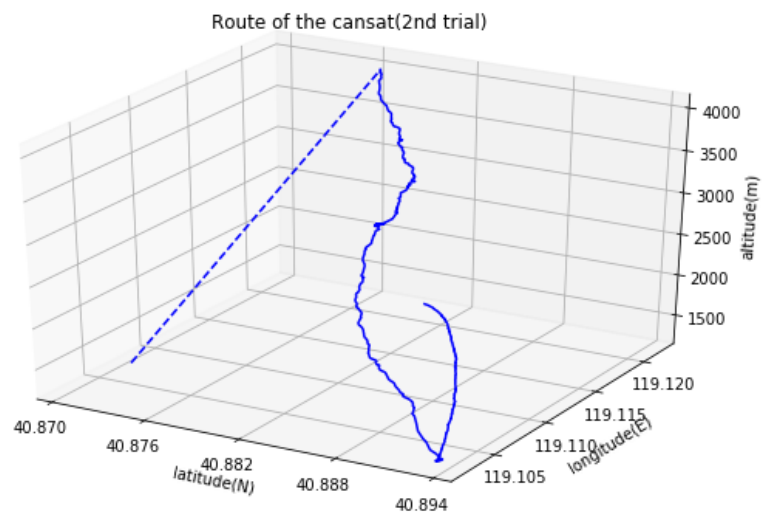
ゴール判定基準
能代宇宙イベント
300秒経過後制御終了

ARLISS1回目
(150秒経過後直進3回)
または
300秒経過

ARLISS2回目
(300秒経過後
赤色面積30%以上かつ
直進3回)
または
600秒経過



2nd Launch (0.00m)



The rover ran 2.2km

Time [s]	Event
0	Wed Sep 13 11:15:31 2017 プログラム起動
1147	ロケットからの放出判定
2002	着地判定&ケーシング展開
2005	地上走行開始
4558	ゴールから3m以内で停止 ->画像処理シーケンス
4880	ゴール判定->制御終了

- 2回目でも着地直後にスタック判定
 - 実際にスタックしたわけではなかった
 - 着地後にしばらくの間移動してもGPSモジュールが受信する座標が変わらない
 - GPSモジュールの衛星受信状態についてログを取ってなかったなので細かい部分まで解析できず
 - 着地して急激に姿勢変化したことに起因するなんらかの不具合か？
- 修正した画像処理シーケンスは成功

Movie

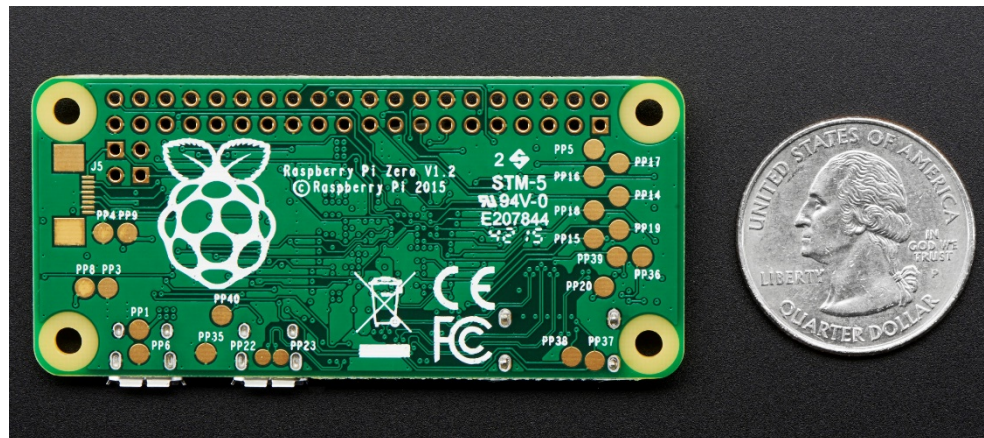
- 岩田製作所も同じ現象が起きていたらしい
- こちらの方が復帰までの時間が早かった
- 使用していたGPSモジュールの受信チップが違うせい(?)
- 岩田製作所のよりtrack channelが18個多い



What worked well and Points to be improved

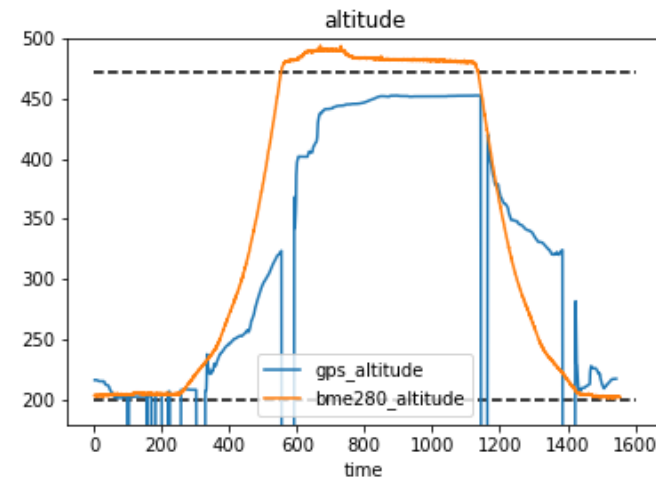
What worked well

- On-board Computer の選定
 - 2014年、2016年の東大ARLISSチームで使われていたArduinoよりも小型、低消費電力、高性能なRaspberry Pi Zero W



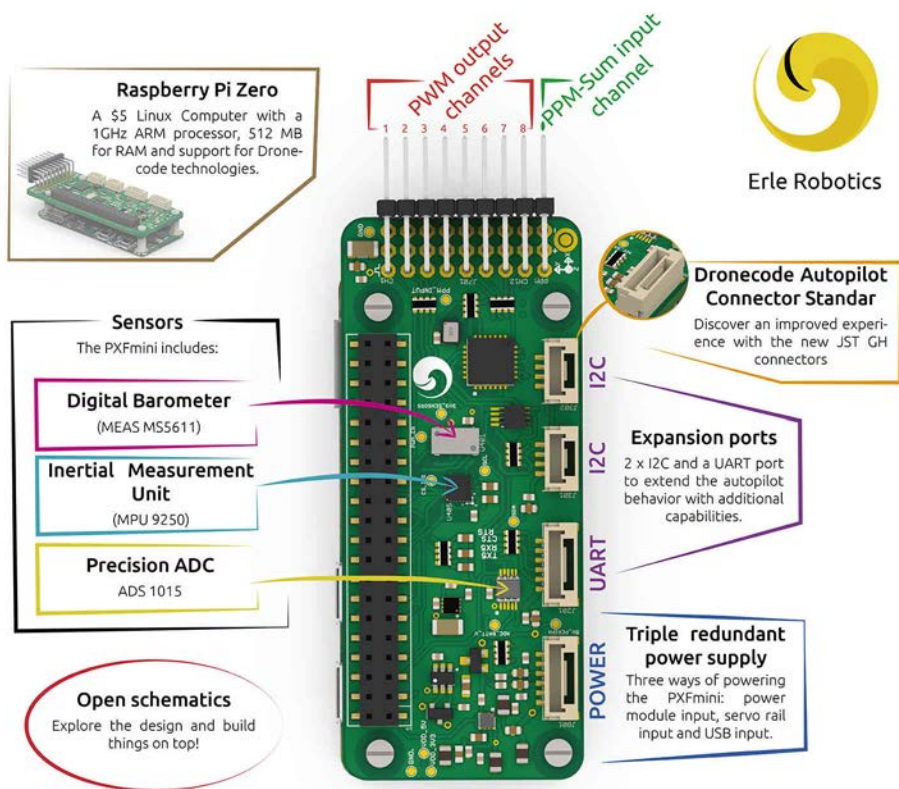
What worked well

- 能代で落下中にGPSの高度が上昇してた
- 高尾山に登りGPSの高度より気圧センサーの方が信頼性が高いと判断、気圧センサーの搭載を決断



- ARLISSでのログを見ると落下中にGPSが衛星を取れてない時があった

Points to be improved



- 既製品のモジュールを使うことによるさらなる基盤の小型化
- 基盤制作過程の短縮

Points to be improved

- モーターの選定

とりあえず 2016年と同じダイセンギアモーターを採用

- 安いが構造的に弱い
- ARLISS直前に二個故障する



- フライバックだとラジコン飛行機用モーターなので事情が異なると思うがKV値などきちんと検討して選定すべき

Points to be improved

- パラシュート分離機構を機体と別にする

東工大のパラシュート分離システム

Points to be improved

テレメトリー通信距離の拡大

- 2.4GHz帯でなく900MHzまたはそれ以下の周波数帯の通信機器を用いる(アメリカで使うはずなので技適は関係ない)
 - Digi-Key使って輸入しましょう
- LoRaを使う
- 八木アンテナの使用



Points to be improved

緻密な制御

- ✕ シングルスレッドで処理する
- ROS(Robot Operating System)という分散型フレームワークの利用



ROS



Finally!!



検討、試験を怠らないように

