



ARLISS 2011

最終報告書

チーム名	チーム”みるみる”
所属名	東京工業大学松永研究室
Project Manager 氏名	森井 翔太
メールアドレス	morii@lss.mes.titech.ac.jp

目次

1. ARLISS2011 参加の目的	3
2. ミッション定義	4
3. システム構成	5
3.1 システム概要	5
3.2 ミッションシーケンス.....	6
3.3 衛星システム	6
3.4 サブシステム	8
3.4.1 構造系	8
3.4.2 電源系	20
3.4.3 通信系	21
3.4.4 C&DH 系.....	22
3.4.5 センサ系	33
3.4.6 メモリ系	38
3.4.7 カメラ系	42
3.4.8 子機系	44
4. サクセスレベル	46
4.1 サクセスレベルの定義.....	46
4.2 サクセスレベルの評価方法.....	46
5. ARLISS2011 試験結果.....	48
5.1 1 st Flight 結果.....	48
5.2 1 st Flight デバッグ	50
5.3 2 nd Flight 結果	51
5.4 2 nd Flight デバッグ	54
6. ミッション達成度の評価.....	56
7. 総括	57

1. ARLISS2011 参加の目的

OPEN-CLASS カンサットの製作, および ARLISS2011 への参加を通じて, 将来小型衛星開発に必要な要素技術を新規取得すると同時に, 松永研究室に蓄積された衛星開発に関する技術的な知識・ノウハウを新入生に伝承する. さらに他団体との交流を通じて衛星開発・宇宙開発の視野を広げるとともに, UNISEC の活動に貢献することを目的とする. 今年度は, 修士 1 年および学部 4 年生が計 7 人参加し, OPEN-CLASS カンサットを制作する過程において, 衛星開発に必要な不可欠な技術を習得すると共に, 概念設計から打ち上げ・運用までの衛星開発の全工程を短期間に経験することで, 衛星への知識を深める. また, 短期間での開発を行うことで, プロジェクト管理能力を養う.

今年度の松永研究室のカンサットは主衛星と分離可能なターゲット衛星の 2 機体から構成され, 主衛星に搭載したカメラにより, 分離したターゲット衛星のパラシュートを画像認識し, 追従制御をかけることで常にターゲット衛星を視野の中心に捕捉することをミッションに掲げている. 完全に分離してターゲット衛星を見失った後は, 周辺環境(地上など)の撮影を実施する. 画像処理速度の目標値は, 1 次分離時の振り子運動の周期から概算した 2frame/sec 以上を目標にしている.

松永研究室は今回の ARLISS を通じて, 新たにオペレーション・システム(OS)の要素技術を取得したいと考えている. 将来の人工衛星に課される課題として, ミッションの高度化に伴う衛星バスシステムの巨大化, 演算処理の高速化・複雑化が挙げられる. 本課題において, OS の搭載は複雑化した衛星バスを効率よく統率する方法の一つである. そこで, 本年度の ARLISS を OS 技術実証の機会と考え, OS を中心としたカンサットバスシステムの構築を目標とした. 高速画像処理・カメラ指向撮影・衛星分離など複雑な機能をもつ OPEN-CLASS カンサットバスを, OS を用いて構築することで, 複数機能のタスクスケジューリングを容易にすると同時に, OS に関する開発技術を取得する. OS を組み込みシステムで稼働させる上での技術的課題として, 各インターフェースのデバイスドライバの作成がある. 本カンサットでは, UART, I2C, SPI, I/O の 4 種類のインターフェースを使用するため, 対応するデバイスドライバを全て開発する必要がある. また, 開発と同時に技術の要点を盛り込んだ報告書を作成し, 後代までの技術伝達を図る.

さらに, 本年度は ARLISS 運営側としての活動も行っており, 大会前のレビュー会の企画・運営を担当した. これは, ARLISS 本番でのミッションが成功するために必要な実機試験レビュー項目を検討し, 今後の ARLISS レビューの基礎を提案するものであり, 本活動を通じて ARLISS の運営に貢献することも参加意義の一つである.

2. ミッション定義

カンサットは親機(主衛星)と子機(ターゲット衛星)の2機体から構成され、親機に搭載したカメラにより、分離した子機のパラシュートを画像認識し、追従制御をかけることで常に子機を視野の中心に捕捉するミッションを提案している。打ち上げ時、親機と子機はテザーにより接続されており、カンサットが所定高度(3000m)に到達すると、テザーを約1.5m伸展させて振り子モードに移行し(1次分離)、その後所定時間後にテザーを切断することで、親機・子機を完全に分離する。この間、親機搭載のカメラは常に子機を追従しており、完全に分離して子機を見失った後は、周辺環境(地上など)の撮影を実施する。画像処理速度の目標値は、1次分離時の振り子運動の周期から概算した2frame/sec以上を目標にしている。以上、高速画像処理・カメラ指向撮影・衛星分離など複雑な機能をもつOPEN-CLASSカンサットバスを、Linux-OSを用いて構築することが本ARLISSの目的であり、複数機能のタスクスケジューリングを容易に行える高機能・高性能なカンサットバスの構築を目指す。

動作模式図を図1に示す。各シーケンスにおけるカンサットの動作は以下の通りである。

- ◆ シーケンスⅠ...①Linux-OS ブート
 - ②各センサのイニシャライズ
 - ③House Keeping (センサデータの保存&データダウンリンク)
- ◆ シーケンスⅡ...①テザー伸展機構作動(1次分離)
 - ②カメラによる子機指向撮影
- ◆ シーケンスⅢ...①テザー溶断機構作動(2次分離)
 - ②カメラによる子機指向撮影
 - ③(子機ロスト後) 周辺景色撮影

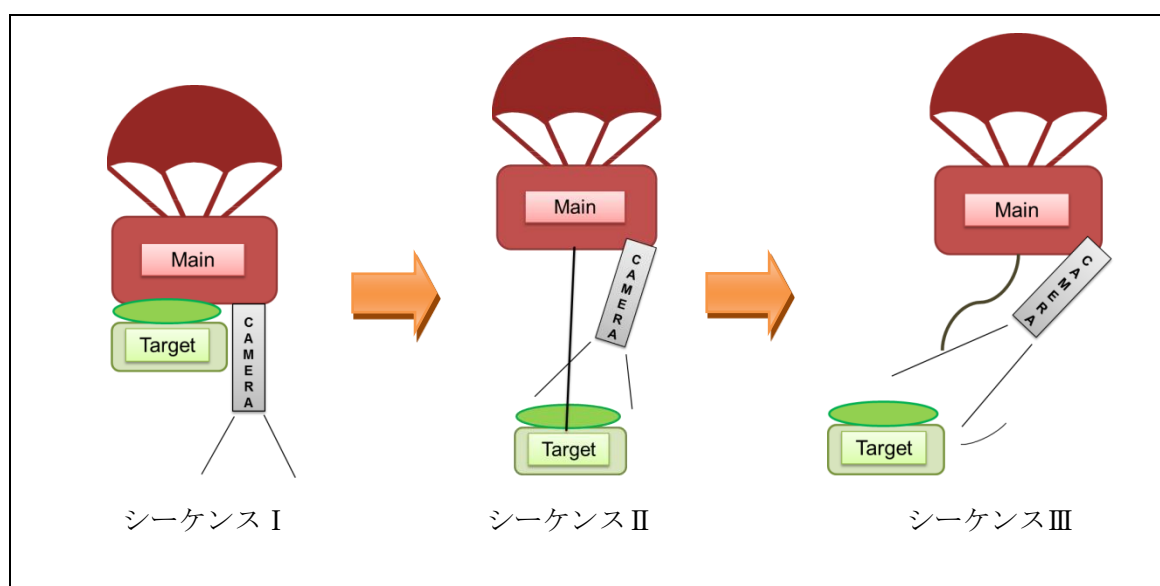


図 1 カンサットの動作模式図

3. システム構成

3.1 システム概要

図2にシステム・ダイアグラムを示す。システムは、大きく分けて親機・子機・地上局の3つにより構成されている。親機のMPUにはOSとしてLinuxを搭載したFPGA(SUZAKU)を用いており、各種センサ、メモリ、カメラ、通信機等のデータハンドリングを行う。一方、子機はMPUとしてPICを搭載した最小限のシステムで構成されており、親機とは分離機構を介して接続されている。親機、子機のテレメトリデータの取得にはそれぞれ専用の地上局(無線機+PC)を用意する。

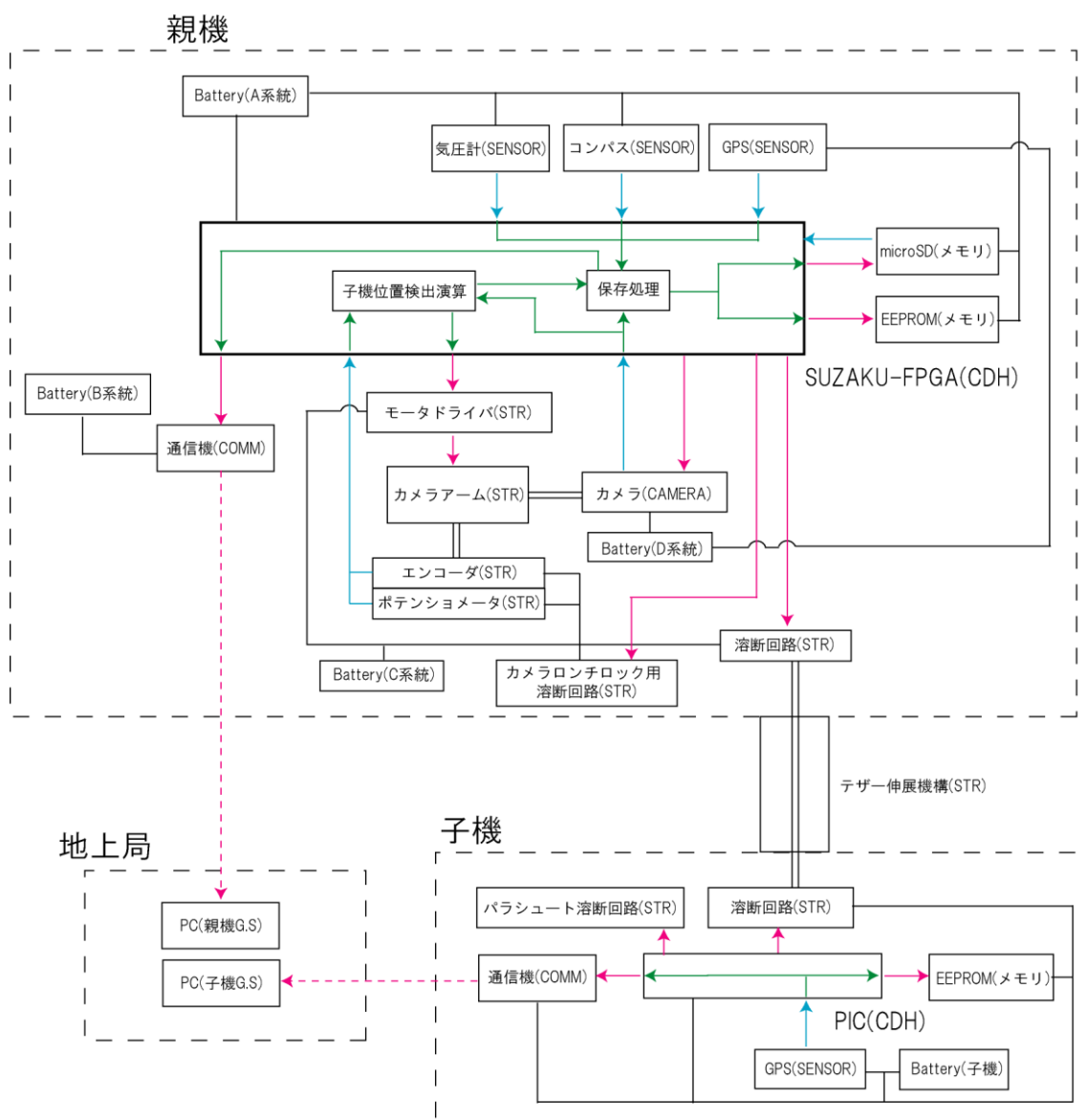


図2 システム・ダイアグラム

3.2 ミッションシーケンス

図 3 にミッションシーケンス図を示す。フェーズの移行は、気圧計による高度データをメインのトリガとして行い、OR として、タイマを用いたトリガを別途設けている(図の黒塗四角)。初期運用フェーズは、親機の電源投入から Linux の起動やセンサのイニシャライズ、HK 確認を行う。高度が 3000m に到達すると、次のテザー伸展フェーズに移行し、子機のテザー伸展、および子機の指向撮影を行うミッションモードに入る。テザー伸展後、500s が経過すると完全分離フェーズに移行し、子機側でテザーを溶断し、親機と完全に分離する。親機は、子機を追従できている間は指向撮影ミッションを実施し、子機をロスト後は、周辺の景色を撮影するモードに移行する。

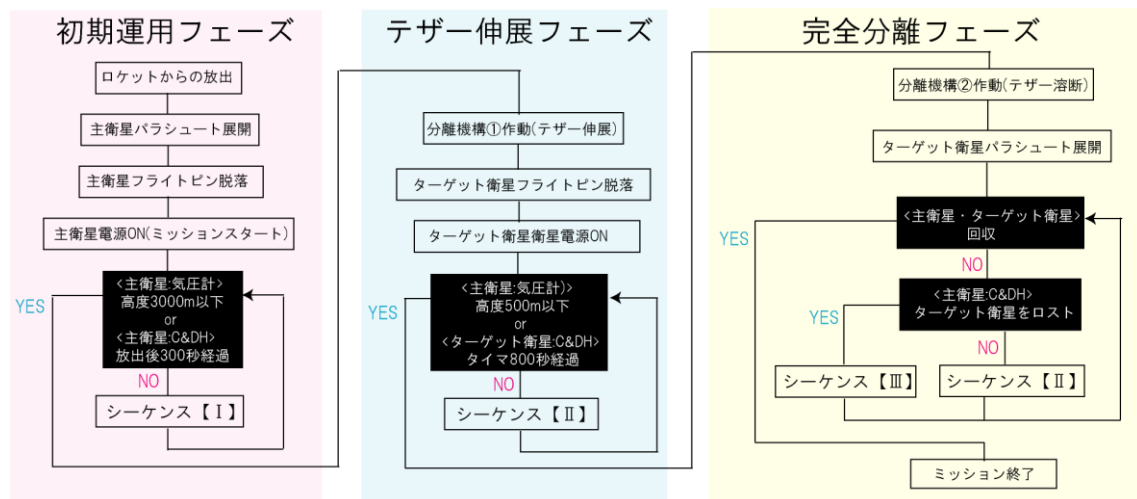


図 3 ミッションシーケンス図

3.3 衛星システム

図 4 にカンサット親機・子機概観(CAD)を示す。親機システムを構成するサブシステムは、構造系、電源系、通信系、C&DH 系、センサ系、メモリ系、カメラ系である。子機は、分離機構により親機に取り付けられており、シーケンス進行に伴い、テザー伸展、及び分離を行う。各サブシステムの詳細は、3.4 節に記載されている。

図 5 にカンサット親機・子機の FM 実機概観を示す。親機・子機質量は、それぞれ 847g / 177g であり、衛星包絡域含めロケットコンフィギュレーションを満足している。

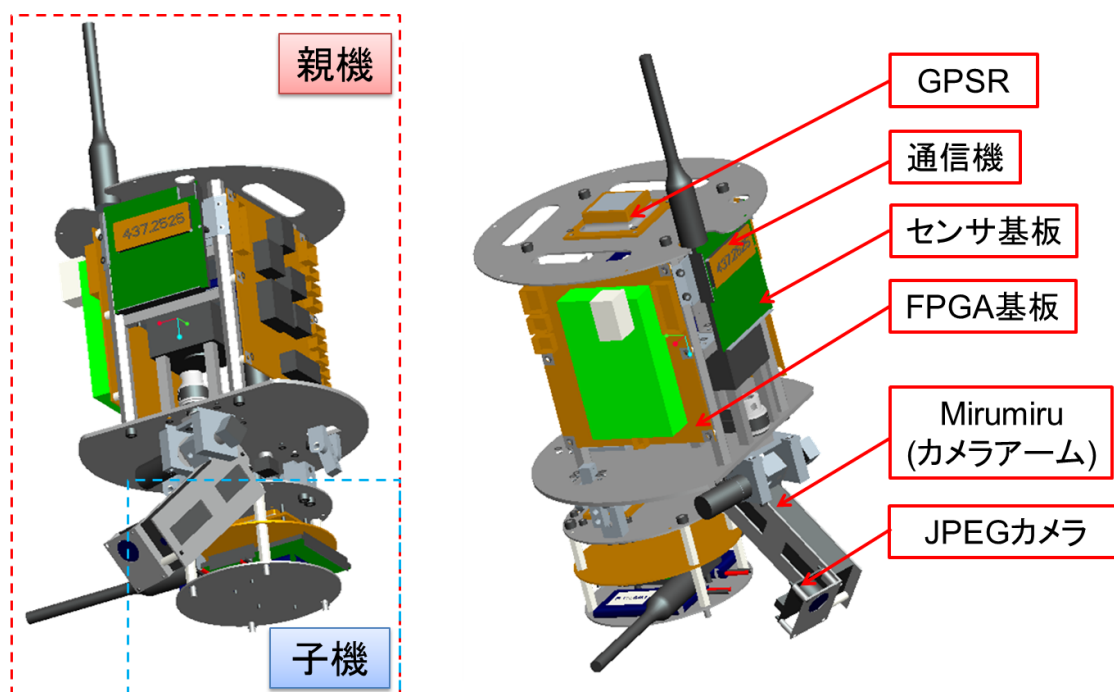
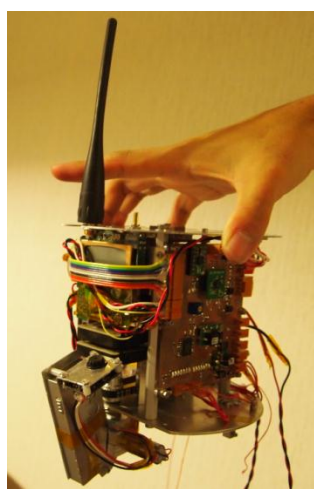


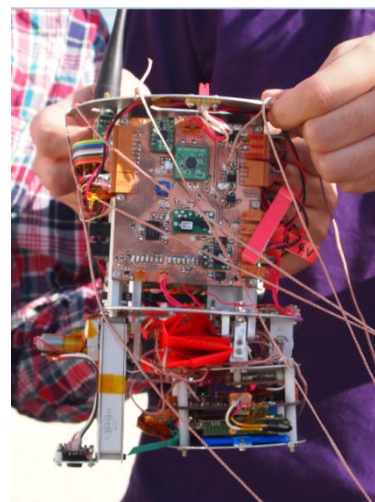
図 4 カンサットシステムの構成(CAD)



(a) 親機



(b) 子機



(c) 親機+子機

図 5 カンサット FM 実機概観

3.4 サブシステム

3.4.1 構造系

3.4.1.1 構造システム構成

図 6 に構造系内のシステム構成図を示す。主構造は、各機器の取り付けの役割の他に、ロケット振動やパラシュート展開衝撃、落下衝撃のロードパスの役割を担う。親機、子機ともに、減速機構としてパラシュートを用いている。また、主構造には、各サブシステムの図に示すコンポーネントが搭載される。親機・子機間のインターフェースとして、親機側に把持分離機構、テザー伸展機構を有し、子機側には親機把持機構の受けとなる把持爪、およびテザーの溶断回路が搭載される。さらに、親機の大きな特徴は 2 軸回転自由度を持ち、2 種類の撮影デバイスを有するカメラユニットである。アームの無限回転を実現するため、制御回路とはスリップリングを介して接続されている。

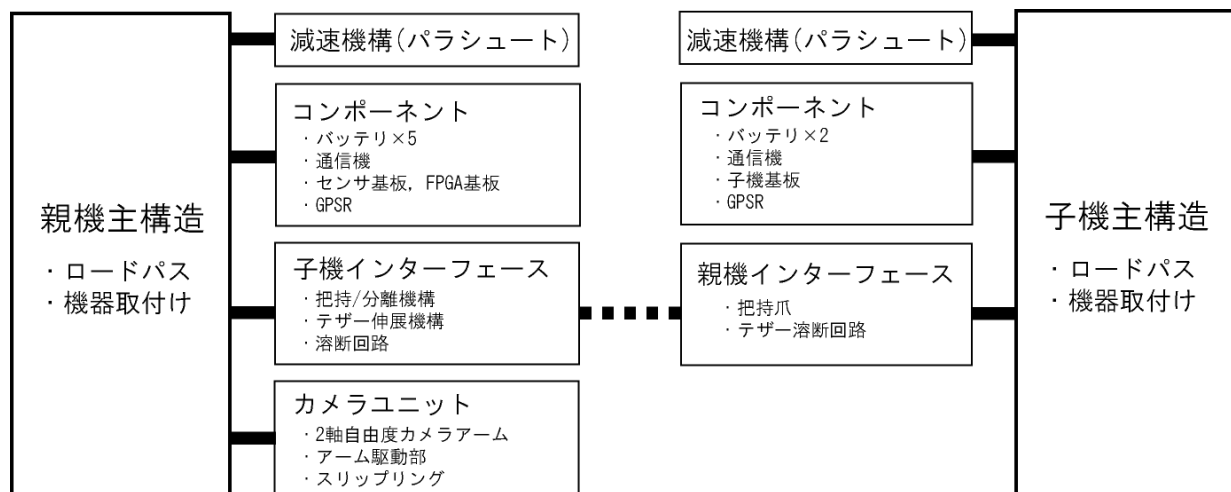


図 6 構造サブシステム構成図

構造設計を行うに当たり、考慮した構造系へのシステム要求項は以下の通りである。

- ・ ロケットの搭載コンフィギュレーションを満足する
 - ◆ 直径 140mm 以下、高さ 240mm 以下、質量 1050g 以下
 - ◆ ロケット振動により破損しないこと
- ・ ミッション期間を長くするため、滞空時間をできるだけ長くすること
- ・ 子機をテザー伸展でき、さらに分離できること
- ・ できるだけ広範囲を撮影可能なカメラシステムを有すること
- ・ FPGA をはじめとした電装機器を全て搭載できること
- ・ リチウムポリマーバッテリーを安全に、かつ必要個数搭載できること
- ・ 動画撮影デバイス”mirumiru”を搭載できること

これら要求をもとに、概念設計を行った結果、現在のサブシステムを採用した。特に要求が厳しかったのは、質量の制約、子機との干渉の制約を受けながら、サイズの大きな mirumiru を駆動するスペースを確保することであり、これが今回の構体設計の肝となった。

3.4.1.2 FM 設計結果

図 7 に、FM の三面図を示す。開発コストを削減するため、ほぼ全ての部品を研究室のアルミ廃材を用いて自作した。加工は石川台 3 号館の創造工房にて、技官の山本さんの協力の下行った。次節以降では、詳細な設計結果について示す。

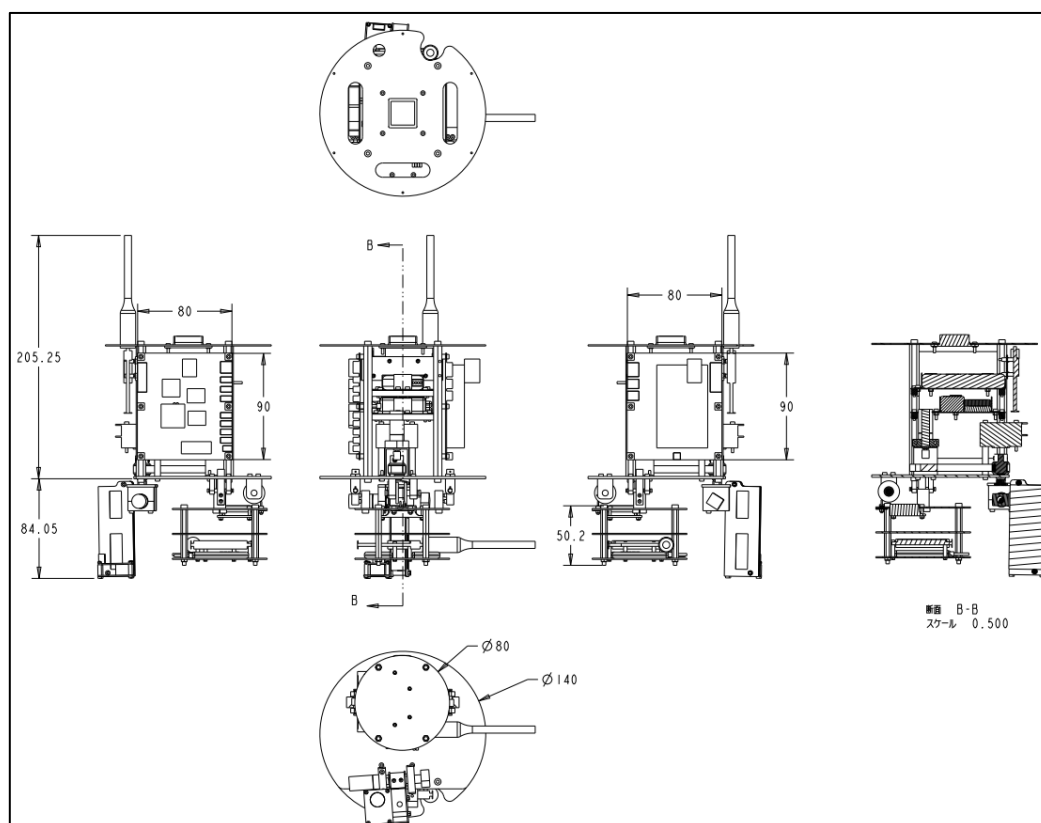


図 7 FM 三面図

3.4.1.3 筐体主構造設計(親機)

図 8 に、親機主構造の CAD モデルを示す。親機はアルミ合金製の六角支柱 4 本(MISUMI で購入)と、四角支柱 4 本を組み合わせた剛性の高い桁構造を有しており、この主構造がロケット振動荷重のロードパスとなるよう設計を行なった。桁構造の中央板部には、バッテリーセルが計 5 つ搭載されているが、主構造の中央部のため直接大きな振動レベルを受けることはなく、振動耐性は十分高いと考えられる。また、桁構造を上下面から支えるのは、厚さ 2mm の底板と、厚さ 1mm の天板である。底板には、子機の把持・分離機構が取りつき、落下の衝撃が最初に加わる部分であるため、強度に主眼を置き、厚さを 2mm とした。

一方、天板は軽量化に主眼を置き、肉抜きによる軽量化処置を行った。

また、高剛性であることに加え、搭載コンポーネントの着脱が容易に行えるのが、親機主構造の大きな特徴である。6 角支柱は表面積が大きいいため、各面を機器取付け面として使用することができるため、狭いスペースを有効活用できる利点がある。図 9 は、主構造に通信機(DJC7), センサ基板, FPGA 基板を取り付けた場合の概観図である。これらの機器は、全て独立に取り外すことができ、さらに天板と底板も独立に取り外せるため、バッテリー以外の機器は全て独立して取り外しが可能であり、組立て、およびデバッグが容易である。バッテリーに関しては、主構造を取り外さなければアクセスできないが、バッテリー交換の頻度が他機器に比べて小さいことを考慮したためである。

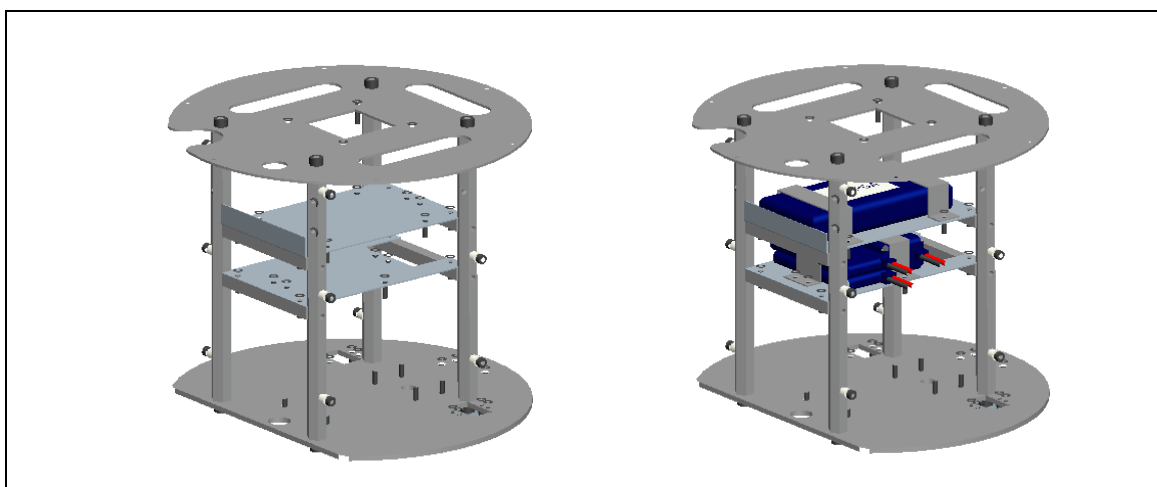


図 8 親機の桁主構造(右図はバッテリーを搭載した場合)

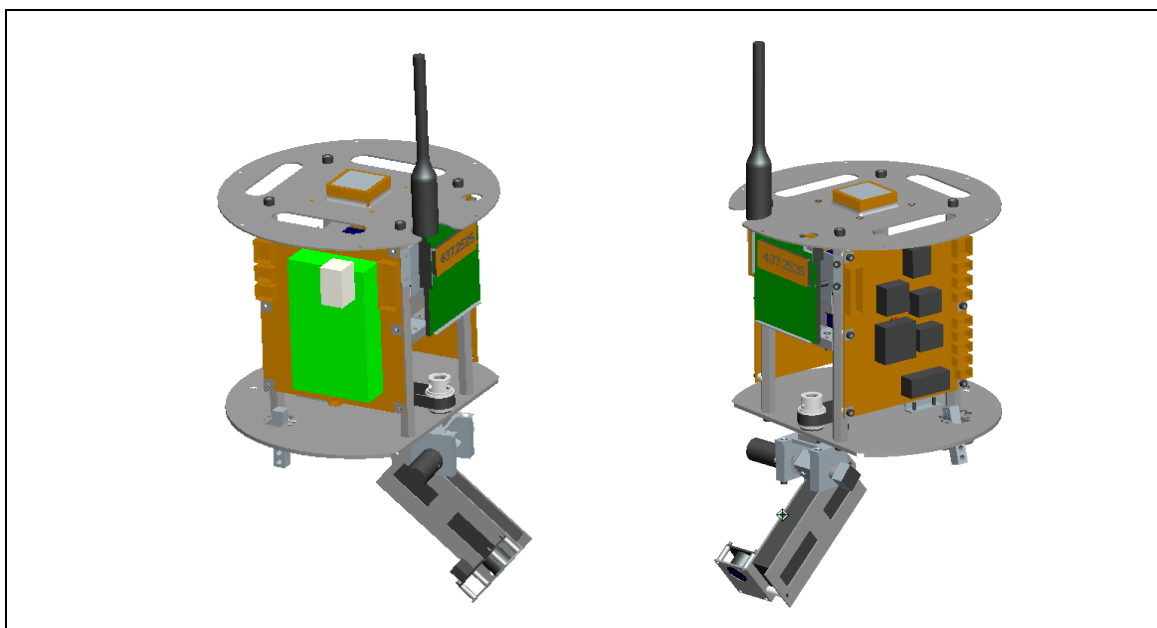


図 9 主構造に機器を取り付けた場合の概観図

3.4.1.4 子機インターフェース(親機)

図 10 に親機, 子機の分離機構を示す. 本分離機構は, 当研究室で開発中の衛星 TSUBAME で使用される分離機構を参考にしたものであり, テザー張力を利用した把持システム原理の確認を行うことを目的として設計されている. 機構は, アルミ合金製の直方体リンク, 把持爪(ホーローネジを流用), 及びねじりばねにより構成され, リンクの一端にテザーの張力がかかることで子機を外側から挟んで把持する. テザーのもう一端は, 底板にねじ止めされた溶断回路に繋がっており, テザーを溶断して張力を解放すると, ねじりばねによりリンクは逆方向に回転して, 子機の把持が解除される仕組みになっている(図 11). テザーは子機の振動により生じる荷重に対して十分強く, 打ち上げ中に子機が分離することはない. なお, 張力をトルクにより管理することができなかったのが本分離機構の欠点であり, 張力がかかる際には十分な注意を払う必要があった.



図 10 親機・子機分離機構の CAD(左)と試作品(右)

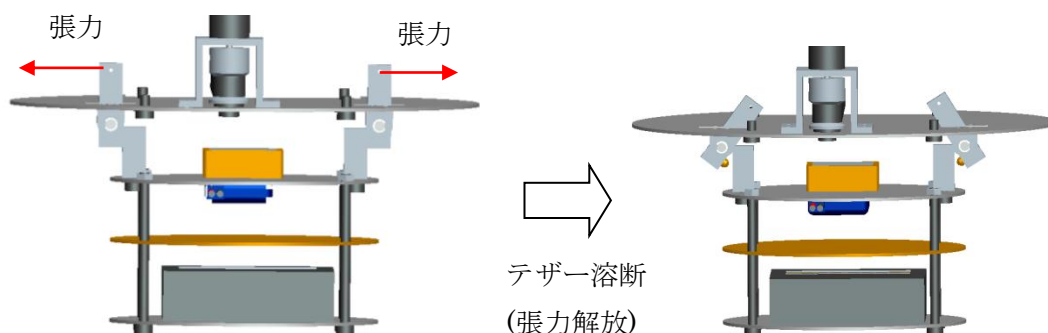


図 11 分離機構の動作メカニズム

子機指向撮影ミッションでは、指向撮影の時間を長時間確保するために、カメラの被写体である子機をある一定距離にとどめておく工夫が必要であった。そこで、子機と親機をテザーにより繋ぎ、振り子運動をする子機を追従制御する方針で設計を行った。図 12 に、テザー伸展機構概観を示す。分離機構の動作後、子機はテザーを伸展させながら落下していく、約 1.5m 伸展後に停止する。テザーは一端をプーリに結び、もう一端を残してあらかじめプーリに巻きつけておく。ロケット振動によるプーリの回転、及びテザー弛緩を防止するために、打ち上げ時はナイロン線を用いてプーリを固定しておく。シーケンス移行時にナイロン線は溶断され、フリー状態になったプーリは、分離機構の動作後、子機の自重により回転し、テザーが自動伸展する(図 13)。テザーが完全に伸展後、子機側の溶断回路を作動させることで、親機とつながるテザーを切断して完全に分離するとともに、子機パラシュートを縛っているナイロン線を溶断してパラシュートを展開させる。

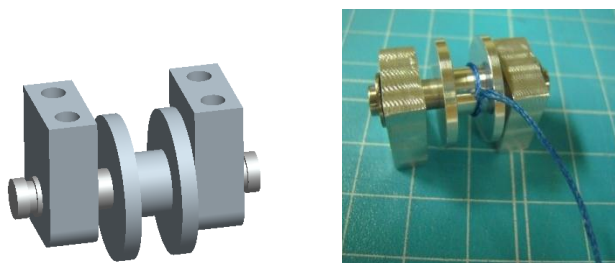


図 12 テザー伸展機構の CAD(左)と試作品(右)

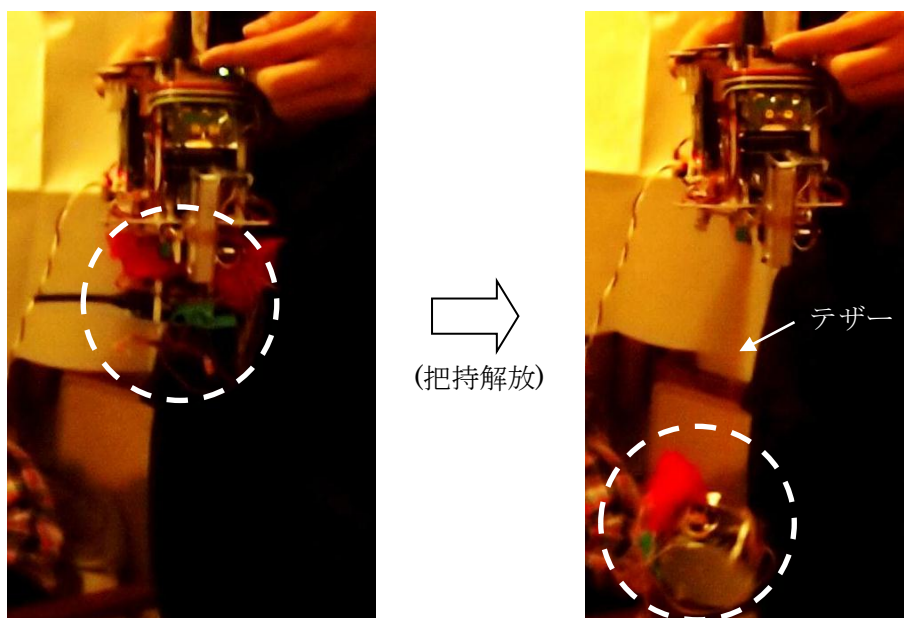


図 13 把持機構動作と連動したテザーの伸展動作

3.4.1.5 カメラユニット(親機)

本カンサットの大きな特徴は、図 14 に示す回転駆動カメラアームユニットであり、DC ギアモータを用いたサーボ系を併せて 2 つ使用することで、カンサット下半球のほぼ全視野を撮影することができる。カメラアームには、図 14(a)に示すように 2 種類のカメラを搭載しており、一つは画像処理追尾に使用する JPEG カメラ、もう一つは打ち上げから回収まで一連の動作を確認・評価するためのポータブルビデオデバイス”mirumiru”である。

カメラアームの駆動には小型 DC モータとエンコーダ、及びポテンショメータを組み合わせたサーボシステムを使用する。サーボモータは、回転範囲が狭いものがほとんどであり、要求仕様を満足するものがなかったため、DC モータを回転センサと併せて使用する方式を採用した。アブソリュートタイプでないエンコーダによる制御は、パルスカウントを正確にモニタできれば、原理的には無限回転可能なサーボ系を組むことができるが、絶対角速度を測定できないため、FPGA 等に予期せぬリセットがかかった場合に問題が生じる可能性がある。図 14(b)において、モータ①に関しては、回転角度領域が 180deg あれば十分だったため、ポテンショメータの AD 変換によるサーボ方式を採用した。一方、モータ②は、要求仕様より無限回転する必要があったため、ロータリーエンコーダによるサーボ方式を採用した。概念図を下図に示す。図に示した通り、DC モータの軸に取り付けられたポテンショメータの電圧値から MPU がモータ回転角を算出し、目標値に近づくようにモータードライバを制御するといった一種のサーボ系を構築している。このシステムによりカメラアームの垂直動作軸を指定角度で停止させるような制御をしている。

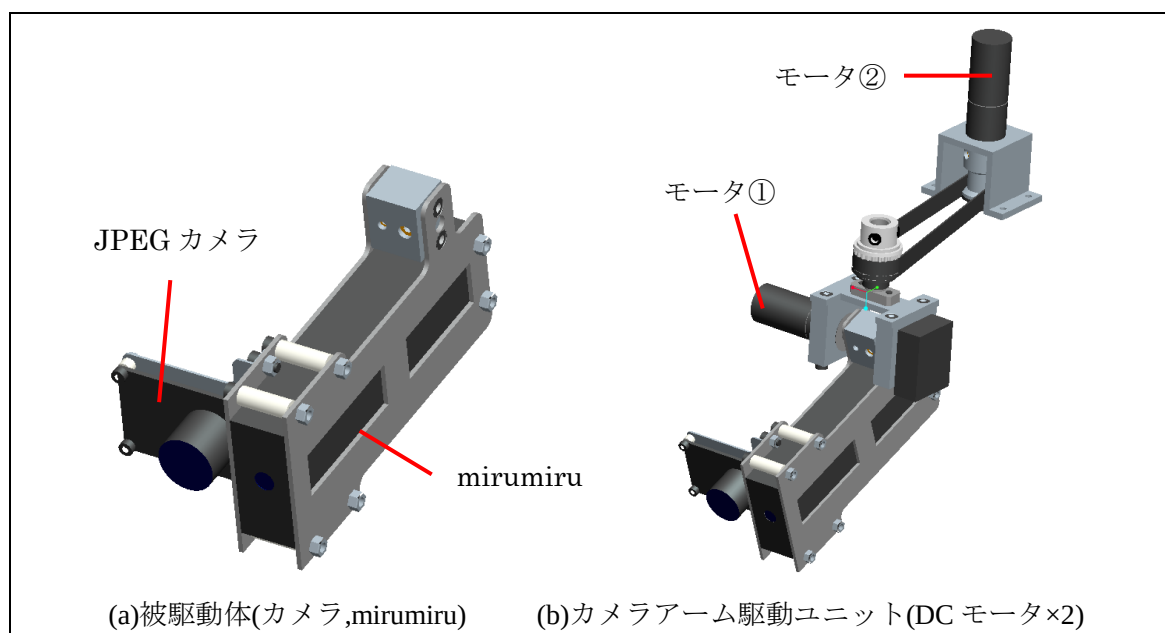


図 14 カメラアームの機構

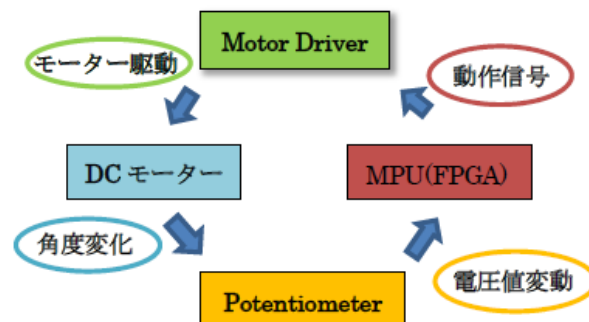


図 15 ポテンショ-モータ系の概念図

モータ諸言を表 1 に示す．本モータは必要なカメラアーム追従速度(2.5rad/s)を満足し，かつ必要なトルク要求を上回るよう選定を行った．アーム回転に必要な最大所要トルク 20mNm に対し，24.6mNm(ノミナル駆動時)を確保している．なお，シチズンマイクロ社のデータシートは，HP よりダウンロードすることができる．また，エンコーダは秋月電子，ポテンショメータは東京ラジオデパートにより購入した．

表 1 モータ諸言

型番	SCR12B4-1804-4.5
減速比	1/208
価格	約¥8,000
販売元	シチズンマイクロ

モータ②でアームを無限回転させるため，カメラおよびモータ①のハーネスがねじれないように図 16 に示す薄型 11 連スリップリング(ロータリーコネクタ)を使用した．スリップリングは秋葉原のロボット部品取扱店において約 4,000 円で入手した．スリップリングによる信号ノイズの増加は確認されなかったものの，回転摩擦トルクが予想以上に大きく，モータに高負荷がかかるのが課題点である．

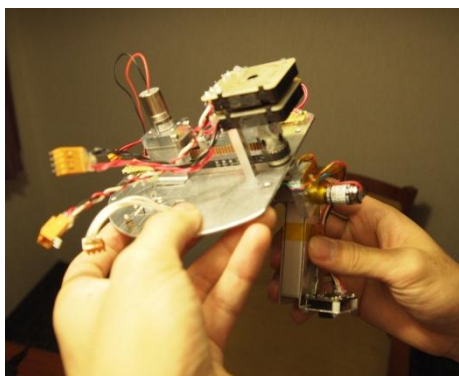
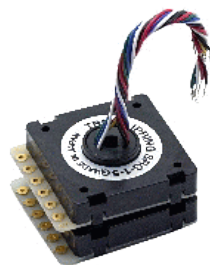


図 16 11 連スリップリング(左)と構体への組み込み(右)

3.4.1.6 子機設計

図 17 に、子機主構造の CAD モデルを示す．親機システムが複雑なため、子機はなるべくシンプルで軽量になるよう設計を行った．子機主構造は 2 枚のアルミ円板と、ナイロン製のねじ付きスペーサにより構成されており、板間には子機基板が取り付けられる．子機はサイズが小さいため、ナイロン製のねじスペーサを用いても比較的高剛性を示し、かつ軽量化も可能である．通信機、及びバッテリーは、底板に取り付けられている．

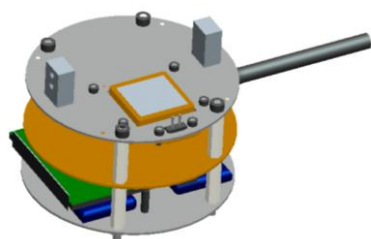


図 17 子機概観(左：CAD，右：FM)

天板には、親機分離機構に関連する把持爪、パラシュート、GPS、および 2 つの溶断回路が搭載されている．図 18 に示すように、子機パラシュートは折りたたまれた状態でナイロン線に縛られて固定されている．2 つの溶断機構は、それぞれ親機と繋がるテザーを溶断するためのものと、子機パラシュートを縛るナイロン線を溶断するためのものである．親機に繋がるテザーを溶断後、数秒でパラシュートを縛るナイロン線を溶断し、パラシュート展開を行う．親機分離機構との干渉やパラシュートひもの絡まりを防ぐため、大きなベタグラウンドを持つ GPSR を、スペーサを用いて天板裏面から取付け、子機天板表面の突起をできるだけ減らす工夫がなされている．

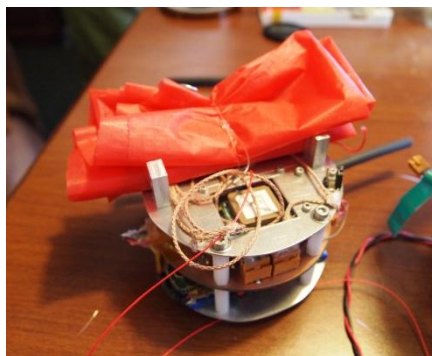


図 18 ナイロン線によるパラシュートの固定

3.4.1.7 減速機構

パラシュート径（直径）は落下速度を数値計算で見積もり，そこから適切な値を選んだ．図 19 に親機と子機の速度見積りを示す．縦軸がパラシュート径であり，横軸が径によって変化する落下速度の値である．ただしここでは親機の質量を 800[g]，子機の質量を 200[g]として計算してある．

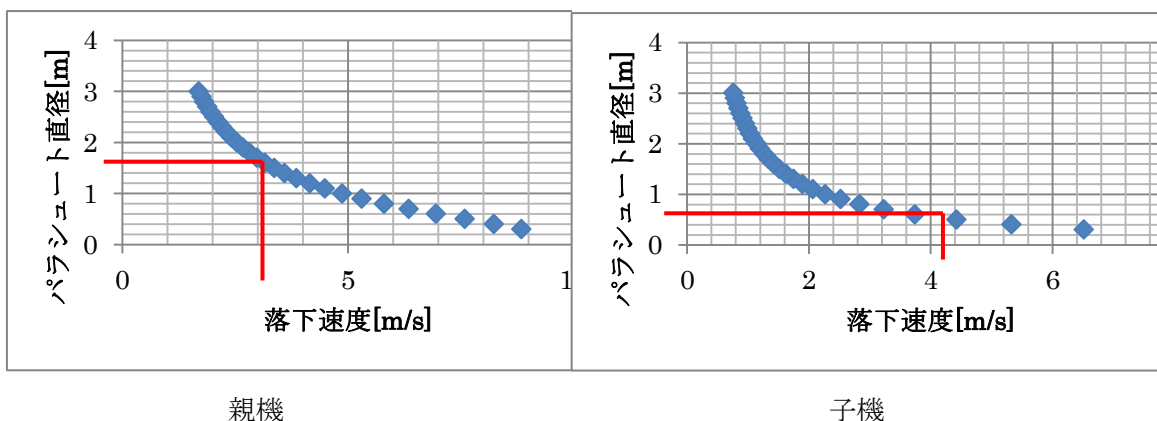


図 19 落下速度見積り

親機が子機をカメラで追従するためには親機の落下速度が子機の落下速度より遅くなくてはならないことを考慮し，図 19 の見積もりから親機パラシュート径を 160[cm]，子機パラシュート径を 60[cm]とした(図 19 中赤線の位置)．この際の落下速度見積りは親機 3.16[m/s]，子機 3.74[m/s]である．また，パラシュートの挙動安定のために中心に空気穴を設けた．この処置により落下時のパラシュートの振動を軽減できるが，落下速度は若干増えてしまう．穴の径は親機が 30[cm]，子機 10[cm]である．パラシュートを設計にはこれらのことを考慮し作成した．ここで，質量管理に余裕をもたせるために，パラシュートの素材には非常に軽量であり強度があるリップストップナイロン生地を選んだ．図 20 に作成したパラシュートを示す．EM 設計時には子機パラシュートは親機の物と同様に六角形状で製作したが，FM

では四角形状に変更した。これは子機を親機に取り付ける際、パラシュートの紐が多く、干渉してしまっていたための処置である。また、四角形状は幾何的な理由から非常に畳みやすいため、収納をコンパクトにすることも可能である。

作成したパラシュートの質量は親機が 82[g]、子機が 19[g]と非常に軽量である。丸子橋試験や気球試験といった投擲試験においても、パラシュートは見積もり速度程度の値を確認している。



図 20 作成したパラシュート（左：親機、右：子機）

3.4.1.8 その他機器の設計

FM で用いたフライトピンを図 21 に示す。フライトピンは、ピンヘッドをユニバーサル基板に半田付けしたものを用い、ピン的一端をパラシュートに結び付け、パラシュートの展開力を利用してピンを抜く。本方式は引き抜きの負荷が小さく、ピンが抜けない等の失敗もなく正常に動作したため、今後のカンサットのフライトピンシステムとして有効であると考えている(注：ピンには抜けにくさが異なるものが数種類ある)。ピンが並列に配置されているのは、誤操作防止のため、及び GPSR の電力消費を抑えるためである、



図 21 フライトピン

FM で用いた溶断回路を図 22 に示す。溶断回路は、ニクロム線をユニバーサル基板に巻

きつめたものであり、ナイロン線、あるいはテザーをニクロム線にひっかけることにより、溶断を行う。本回路の問題点は、ナイロン線(テザー)に張力がかかるとニクロム線の輪が変形してしまうこと、端部がニクロム線でむき出しの為、筐体との不意の導通の危険性があることである。特に、後者についてはデバッグ中に何回か発生しており、注意が必要である。

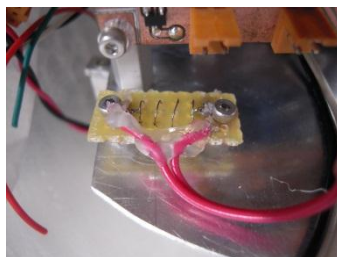


図 22 溶断回路

その他、電装系については、コネクタ周りをホットボンドにより固めることで、振動によるコネクタの脱落を防ぐ工夫を行なっている。また、打ち上げの際は、砂埃を防ぐための”ぷちぷち”を側面基板上に取り付けた(図 23)。



図 23 “ぷちぷち”をまとった FM

3.4.1.9 地上で実施した評価試験

①高所からの構体落下試験@丸子橋(計 30 投程度)

高度約 12m の橋の上からカンサット(親機・子機)を投下し、親機・子機単体でのパラシュートの展開確認、および子機分離からのパラシュート展開確認を行なった。ほとんどの場合でパラシュートは完全展開し、カンサット終端速度が約 2.5～3.2m/s と十分に減速できていることを確認した。また、分離機構の動作を絡めた一連の動作シーケンス試験も実施し、

親機・子機の分離から子機のパラシュート展開までの正常動作を確認した。なお、何回かパラシュートの展開力を利用して抜けるフライトピンが変形してしまい、フライトピンが抜けず自由落下したケースが見られたため、フライトピンを従来のものより抜けやすいものに変更して対応した。



図 24 パラシュート展開確認試験@丸子橋

②気球試験@東工大グラウンド(計 5 投)

気球高度 50m からカンサットを放出し、親機のパラシュート正常展開動作を確認した。一方、分離機構を用いた子機の分離・パラシュート展開については、分離は確認できたものの、展開機構の動作不良により確認できなかった。



図 25 気球試験@東工大

③フライレビュー@東工大グラウンド(計 3 投)

②同様、気球高度 50m からカンサットを放出し、親機のパラシュート正常展開動作を確認した。一方、分離機構を用いた子機の分離・パラシュート展開については、分離機構の動

作不良により確認できなかった。ここで発生した動作不良とは、親機と子機を結合するナイロンテザーが子機のフライトピンと絡まり子機の展開が不能になる、というものである。この失敗を受けて、子機側との機構干渉をなるべく小さくするよう分離機構部の設計変更を行い、結果を FM 構体に反映させた。



図 26 フライレビュー@東工大

3.4.2 電源系

本カンサットにおける電源系への要求は、ロケット放出から着地までの間に各機器へ安定した電力を供給することにある。

表 2 に各機器の予想最大消費電力と選定した電池の容量を示す。本カンサットは電力消費が大きい機器が多く、また電源ラインにノイズを生む機器も多いので、確実な動作を保証するために電源システムを下記の通り 5 つに分類している。ロケット放出から着地までの時間は最大 30 分と考えられるが、マージンを取ってさらに 1 時間以上電力が供給可能なように電池を選定した。また、D 系統に含まれる GPS に関しては、ロケット放出後に電源を供給すると位置データ取得に時間がかかるという問題があるため、ロケット放出前から電源を入れておく。そのため、GPS に電力を供給する D 系統には最長 1 時間 GPS の待機電力 0.29[W]を見込むとする。

表 2 消費電力一覧

電源系統	最大消費電力[W]	電池容量[Wh]	設計電池寿命[min]
A 系統(CPU・センサ)	3.33	6.66	120
B 系統(通信機)	0.60	1.18	118
C 系統(サーボ・溶断)	0.78	1.78	137
D 系統(カメラ・GPS)	0.46 (打上待機時 0.29)	1.18	137
E 系統(SD カード)	0.15	0.67	268
子機	1.08	2.36	131

また、選定電池・本番同等回路で電力耐久試験を行っており、各系統いずれも 120[min]以上の動作を確認することができている。

3.4.3 通信系

通信系のタスクとしては、ミニマムサクセスに挙げられる GPS データの正常ダウンリンクが挙げられる。以下の図 27 に、上目的を達成するために構築するために構成した通信系(衛星側)のシステム概略図を示す。

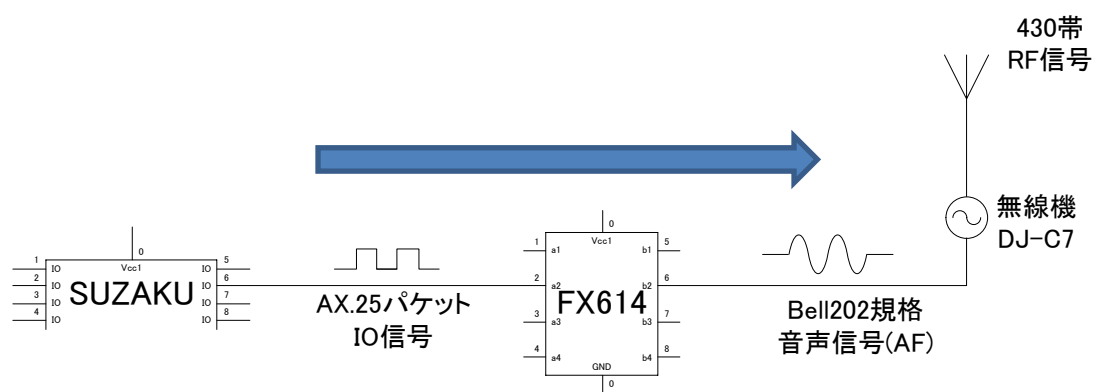


図 27 通信系概略

通信系では AFSK 変調で 1200bps の AX.25 といった通信プロトコルを採用した。MPU である SUZAKU からの信号は半導体素子 FX614 により Bell202 規格にのっとった音声信号へと変換される。この音声信号は無線機 DJ-C7 のマイクラインへと入力され、FM 変調されて伝達される。

AX.25 プロトコルの採用理由としては、松永研究室で過去に十分な仕様実績を積んでいること、無線機に比較的軽量なものが準備できること、そして松永研究室の有する小型衛星運用のための GSE とプロトコルが同一であるため単機能の確認が容易であるといったことがあげられる。このことにより無線通信に関しては、東工大アマチュア無線局を用いることにより地上局・衛星局とを別々に開発、動作確認及びデバッグできることとなり、開発の向上を図れると考えた。

GPS データは 20 バイト強のデータ量があり、AX.25 のヘッダー及びフッターを含めると伝送データは 40 バイト弱ある。これに加え検波のためにデータの前後にスタートフラグとエンドフラグを付与する。結局モデムから送信されるパケットは 100 バイト強の量となる。これを考慮すると一回の GPS データダウンリンクには 1 秒弱の時間を要する。これでは CPU のリソースを占有してしまうために、GPS データは数秒に 1 回程度行うものとした。OS によりマルチタスクが実現された暁には、このリソース占有も緩和され FM ダウンリンク中に他のタスクをこなすことが出来るといったことが期待されている。

RF の送受信にはアルインコ社製の DJ-C7 を用いる。DJ-C7 は改造されて衛星に搭載される(図 28 写真参照)。この DJ-C7 は現在松永研究室が開発している小型人工衛星 TSUBAME にも搭載予定であり、ARLISS を通して無線機改造技術を習得することが出来た。

搭載アンテナは図 29 に示すホイップアンテナを用いる。このアンテナは比較的構造的に柔軟にできており、衛星がロケットに収納されている際には折り曲げて搭載できることから、スペースを占有するといった問題も生じない。



図 28 衛星載用改造済み無線機(DJ-C7)

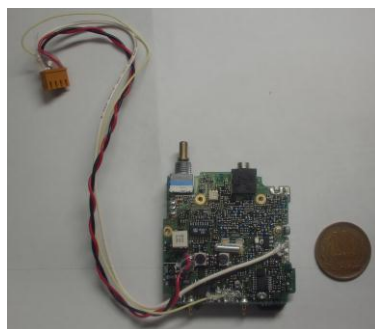


図 29 衛星搭載ホイップアンテナ

3.4.4 C&DH 系

3.4.4.1 C&DH 系概要

C&DH 系のタスクとしてはミニマムサクセスにあたる OS を用いた衛星バスシステムの構築であり、今 ARLISS におけるフライトプログラムを全て開発する。具体的には各種センサ、通信機器、メモリ、カメラ、モータを操作するためのデスドライバ開発、そしてそれらを利用した統合アプリケーションの開発がある。(図 30)

OS を組み込み開発で用いる利点としては、メモリ等のリソースを効率的に行うことが出来る、マルチスレッド・マルチプロセスといったタスクの並列処理が可能である、既存のアプリケーションやライブラリを用いることが出来る、開発の効率化が図れるといったことがあげられる。本年度の ARLISS では松永研究室としては初めて OS を利用した組み込み開発を行う。ARLISS を通して OS を用いた組み込み開発の技術習得が目標の一つである。

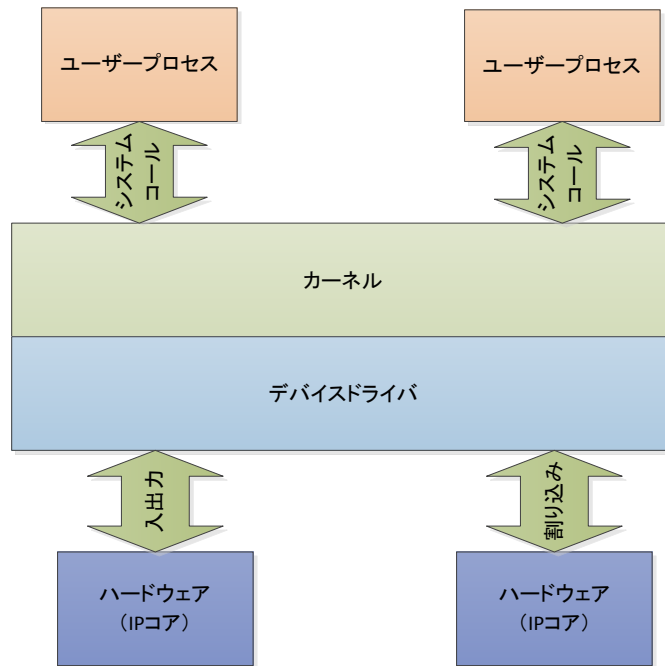


図 30 C&DH(ソフトウェア)概略図

3.4.4.2 OS 及びマイコンボード選定

開発が始まるに際してまず OBC 及び OS の調査を行い、トレードオフスタディを実施することでこれらを決定した。

OS 選定の際に候補として挙げられたのは、Linux, VxWORKS, ITRON である(図 31)。フリーで入手できるものもあり、実際に使用するにはマイコンからブートローダを起動、ブートローダからの OS 起動を行わなければならない。

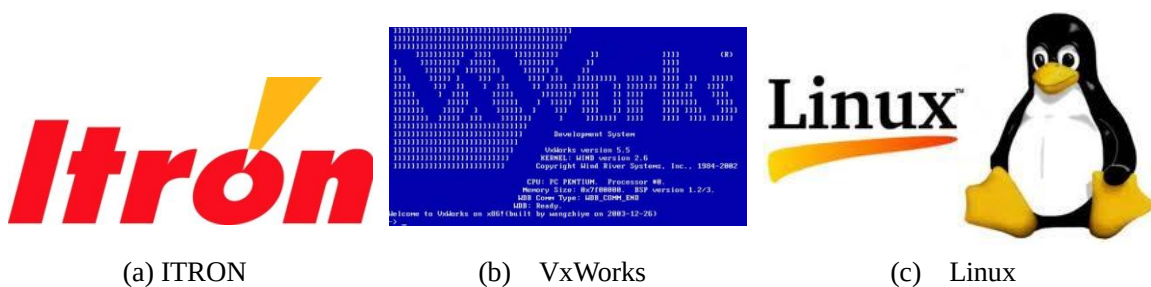


図 31 OS 選定

トレードオフスタディの際には入手性、使用実績、リファレンスの充実度をパラメータとした。これらの結果を受けて OBC としては PowerPC, FPGA を搭載した Atmark Techno 社の SUZAKU sz-410(図 32.)を使用することに決定した。SUZAKU の特徴としては工場出荷時の状態で Linux がすでにマウントされていること、FPGA による IP コアを利用できることからペリフェラルを柔軟に構築できること、Atmark Techno 社より豊富なリファレンスガ

イドや充実した開発環境が提供されていることが挙げられる。また松永研究室では近年十分に利用してきたボードであり、新規技術獲得要素が少なく、すぐにでも開発がスタートできる状況であった。Linux に関しても開発メンバーに多少の経験があること、Web 等の開発の参考となるものが多数ある等の事柄が他の OS を利用することに比べ新規開発の敷居が低いと考えた。

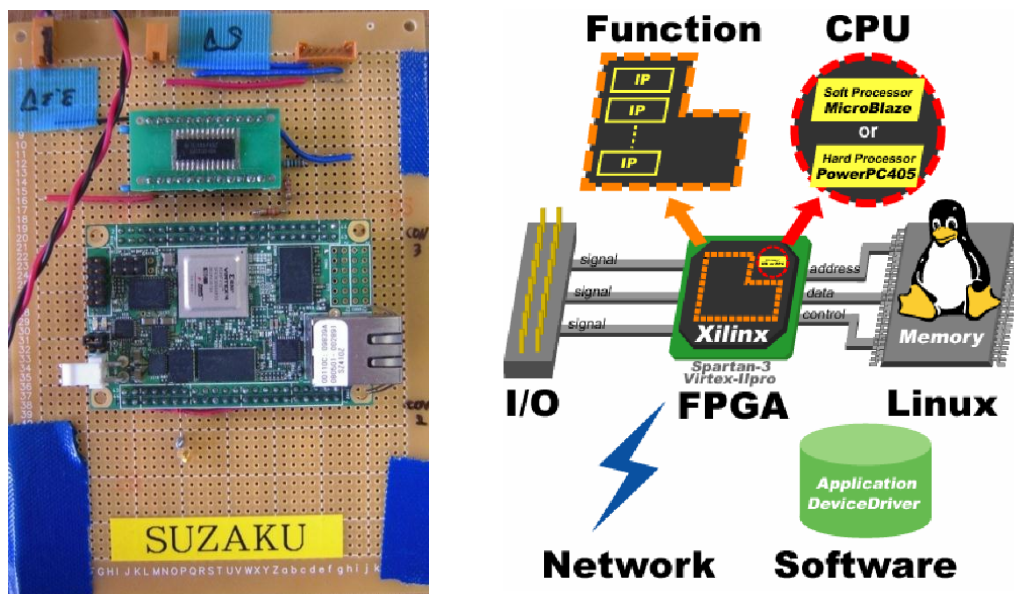


図 32 SUZAKU sz-410

3.4.4.2 開発環境構築

ソフトウェアの開発には Linux のディストリビューションである Debian に AtmarkTechno 社が開発に必要なツールを既にインストールしたディストリビューションである ATDE2 を用いた。これは Windows 上で他の OS をエミュレートするソフトウェアである VMWare Player 上で動かすことができ、AtmarkTechno 社のウェブサイトよりフリーで入手することが出来る。また、開発言語は C 言語であり、コンパイラには gcc コンパイラを用いる。

SUZAKU 上で走らせる Linux はこちらも AtmarkTechno 社の開発したディストリビューションである「atmark-dist」を用いる。

ATDE2 には開発に必要なクロスコンパイル環境が揃っており、開発者は作業 PC 上でプログラムをコンパイルし、SUZAKU 上へ FTP 転送もしくは Linux イメージの書き換えといった作業ができる。



図 33 ATDE2

ソフトウェアのデバッグにはアウトオブツリーコンパイルが便利である。

通常では作業 PC 上でアプリケーションもしくはデバイスドライバを作成後、Linux のイメージを丸ごとコンパイルし、SUZAKU へと書き込まねばならない。しかしながら、Linux カーネルを構成するプログラムだけでも膨大な量があり、コンパイルには数分の時間を要する。また、コンパイルされたイメージを SUZAKU に焼く作業にも同様に数分を要する。これでは、デバッグ性が悪い。

そこで用いられるのがアウトオブツリーコンパイルである。アウトオブツリーコンパイルとはその名の通り、Linux ディストリビューションを含むファイルシステムの外でターゲットとなるプログラムのみをコンパイルすることである。コンパイルにはアウトオブツリーコンパイル専用の Makefile を作成し、make を実行しなければならない。しかしながら、コンパイルに要する時間は数秒であり、デバッグ性は格段にアップする。コンパイルの結果出来上がった実行ファイルなどは、イーサネットケーブルによる LAN で SUZAKU 上のファイルシステムにダウンロードでき、「chmod」コマンド等でユーザー権限の変更後、SUZAKU のデバッグコンソールから実行することができる。LAN のビットレートは 100Mbps を銘打っており、多少大きなファイルでもただちに SUZAKU へとダウンロードできる。

3.4.4.3 デバイスドライバ開発

3.4.4.3.1 デバイスドライバ概要

図 34 にデバイスドライバを含む Linux システムの概要を示す。OS を用いる場合、ユーザ

が作成するアプリケーションプログラムは直接的にはハードウェア(I/O)を操作できない。そこで、アプリケーションからはデバイスドライバに対してシステムコールを行う。デバイスドライバはシステムコール通りにハードウェアを操作する。

デバイスドライバの種類は3種に大別される。

- ✓ キャラクタデバイスドライバ
- ✓ ブロックデバイスドライバ
- ✓ ネットワークインターフェースドライバ

キャラクタデバイスドライバはプリンタやスキャナといったようなキャラクタデータのストリームを扱うデバイスのドライバ、ブロックデバイスドライバはハードディスクドライブやCD-ROMといったようなランダムアクセスが可能なデバイスに区別される。ネットワークインターフェースのドライバはキャラクタ型・ブロック型どちらにも分類されず、独自にネットワークデバイスドライバとして扱われる。

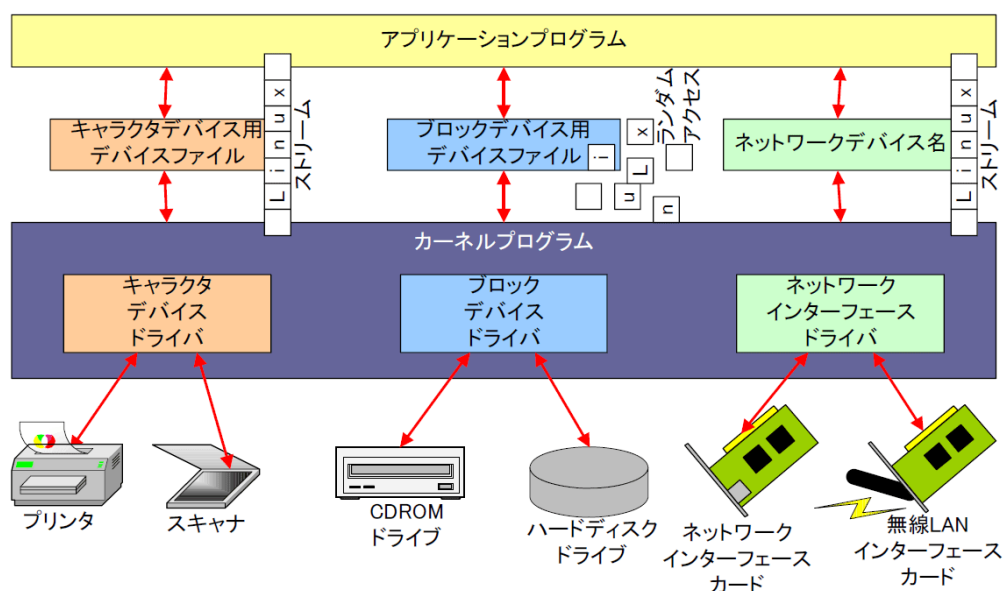


図 34 デバイスドライバの種類、役割

3.4.4.3.2 デバイスファイル

Linux ではアプリケーションがデバイスドライバを扱うインターフェースとして、デバイスファイルという仕組みを利用している。デバイスファイルとデバイスドライバはメジャー番号とマイナーはメジャー番号とマイナー番号によって結び付けられる。番号は共に 0 から 255 まで振り分けることが出来、それぞれ各デバイスへと割り当てられる。

デバイスファイルはユーザランド側(ファイルシステム上)に用意する必要がある。新たにデバイスファイルを作成する際に「mknod」コマンドによりデバイスファイル名、メジャー番号、マイナー番号を与えて実行する。

アプリケーションは作成したデバイスファイルに対して **Open** 等の関数を呼び出す。呼び出されるのはデバイスドライバ内に作成された対応する関数であり、デバイスドライバ開発者はこの関数の中身を開発することとなる。

特に **Write** 関数及び **Read** 関数はアプリケーション内から呼び出すだけでなく、**Write** 関数の場合は「**echo**」コマンドで、**Read** 関数の場合は「**cat**」コマンドで呼び出すことが出来る。この手法はわざわざアプリケーションを作成しなくても利用できることから、デバッグの際、バグ箇所の切り分けに便利である。

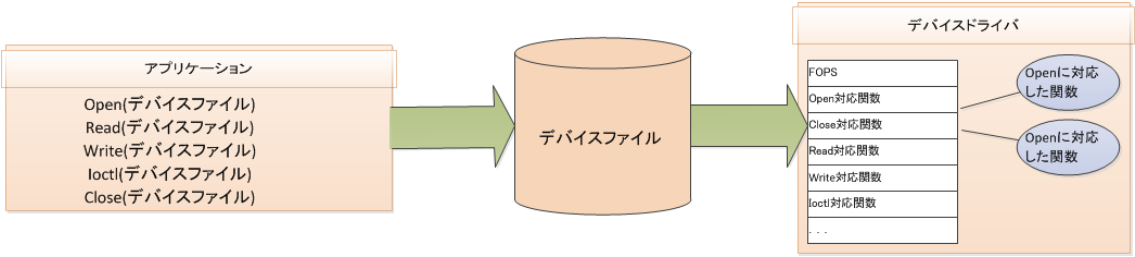


図 35 デバイスファイル

3.4.4.3.3 デバイスドライバ

本 ARLISSd では表 3 に掲げられたデバイスドライバを開発した。開発した順番は表の上の方から開発していき、逐次その知識・技術について蓄えていった。

表 3 開発デバイスドライバ

IP コア	機能
GPIO	子機分離溶断回路
	有限回転モータ駆動
GPIO+TIMER	FM 通信
UART	GPS
	SD モジュール
	CAMERA 高速撮影
SPI	気圧計
	ポテンショ用 AD 変換機
I2C	コンパス
モータエンコーダ用 自作 IP	無限回転モータ駆動, エンコーダ値読み取り
カメラ初期化用自作 IP	CAMERA 初期化

開発の際の注意点として、デバイスドライバ内で無限ループがある場合、CPU の処理がアプリケーション側に戻ってこなくなり、事実上フリーズすることがある。そのため今回は

```
while(1){
    処理
    if(条件) break;
}
```

といった表記はせず、

```
i=0;
while(i<1000){
    処理
    i++;
    if(条件) break;
}
```

と書き、確実にタイムアウトでループを抜けるような仕組みをとった。より詳細なデバイスドライバ開発ノウハウは今後資料を充実させていく予定である。

3.4.4.4 アプリケーション開発

アプリケーション開発を行うに当たっては以下のようなマルチタスクの利点が最大限生かすことを目標とした。

- 他のアプリケーションが停止しても動作可能
- 機能ごとに分けたアプリケーションを作れる

一点目の実装例として GPS データ取得とダウンリンクを行うアプリケーションを独立させることで、その他のミッション系アプリケーションが停止した場合でも(また main プログラムの場合でも、かなりクリティカルな部分を除き)、GPS データのダウンリンクが行われるようにした。また 2 点目の利点により単機能開発時のプログラムをほぼそのまま使うことができ、ソフトウェア統合時に起きるバグを減らすことができた。

今回製作するうえで導入した OS 上で動作するアプリケーション固有の動作は

- 子プロセス生成
- タスク間通信 (メッセージキュー方式)

である。このうち子プロセス生成は主に main プログラムがその他のアプリケーションを実行するために使われた。またタスク間通信は複数のアプリケーション間でデータをやり取りする際に利用した。加えて Linux 自体にシェルスクリプトや起動時間取得などの機能が備わっていたため柔軟な開発を行うことができた。

以下に各シーケンスにおけるタスク図を図 36(a)(b)示す. また各シーケンスにおける main プログラムのフローチャートを図 36(c)~(h)に示す.

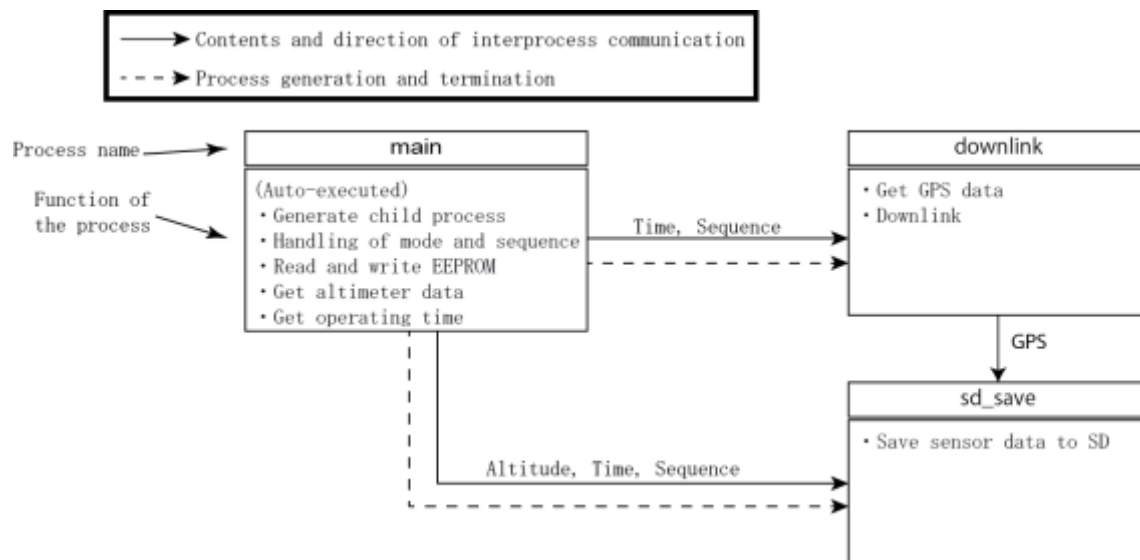


図 36 (a) Task diagram in sequence 1 and 4.

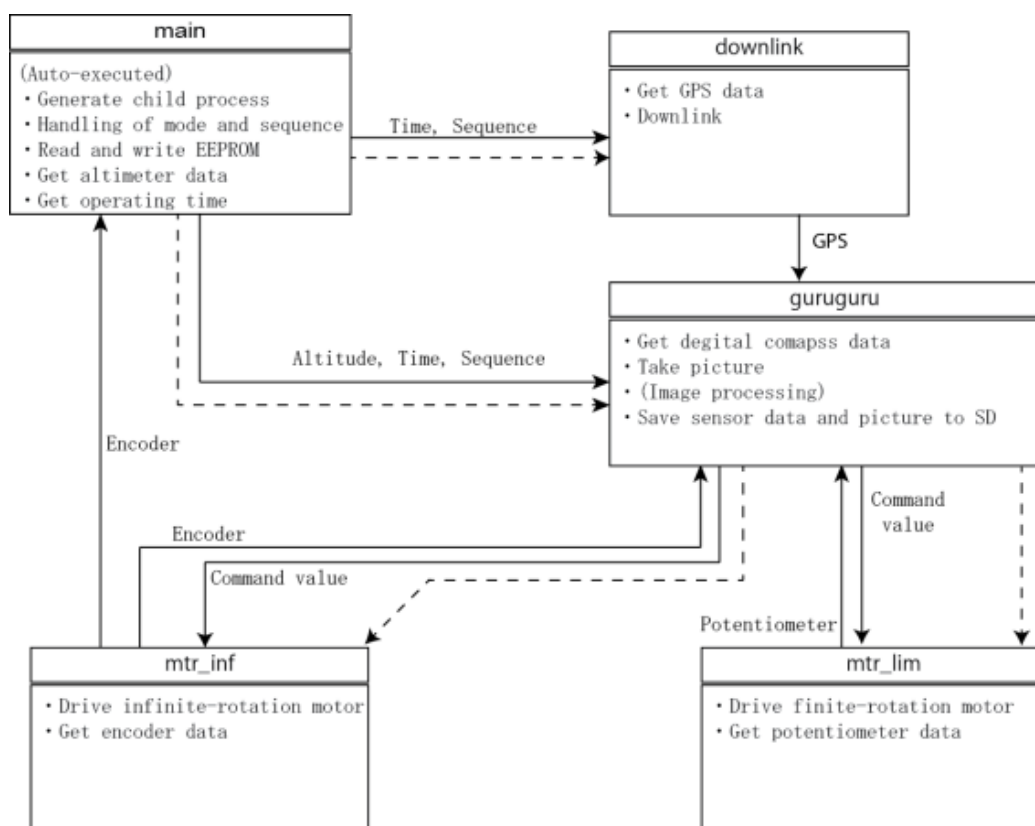


図 36 (b). Task diagram in sequence 2 and 3.

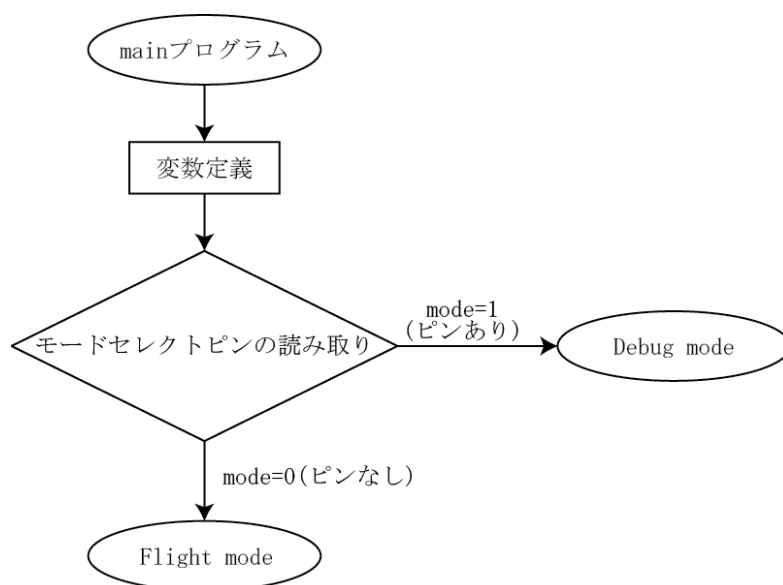


図 36 (c) Flow chart of startup routine

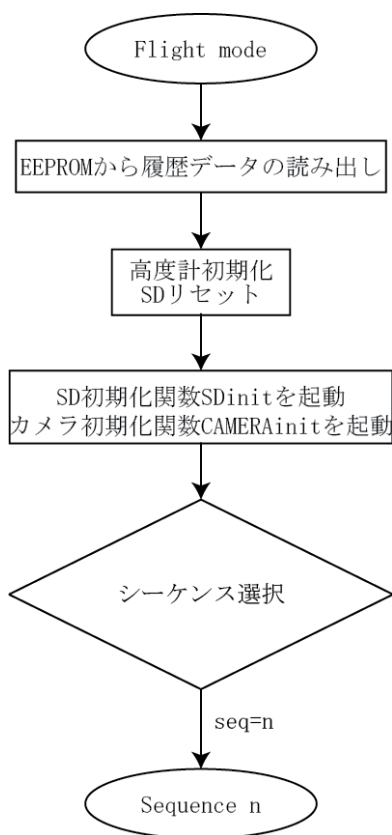


図 36 (d) Flow chart of sequence select routine

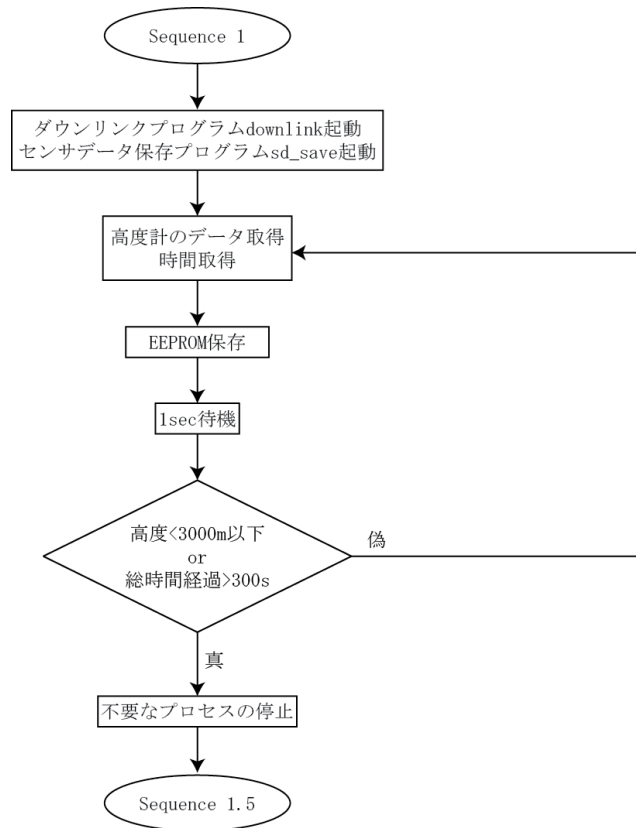


図 36(e) Flow chart of sequence 1 routine

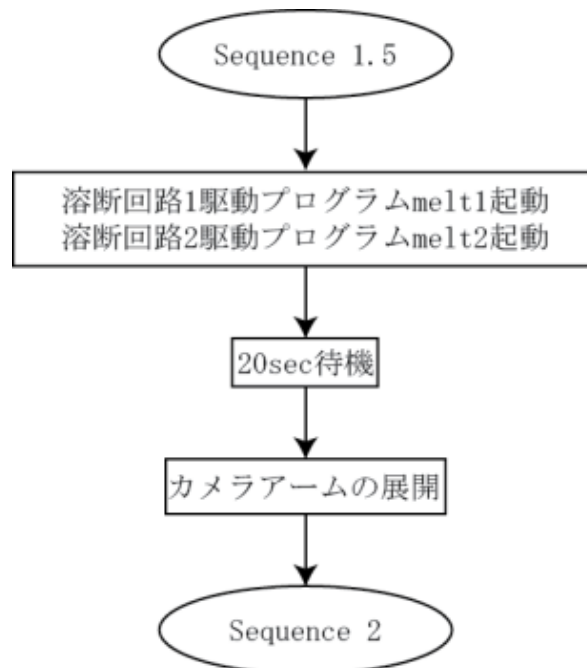


図 36 (f) Flow chart of sequence 1.5 routine

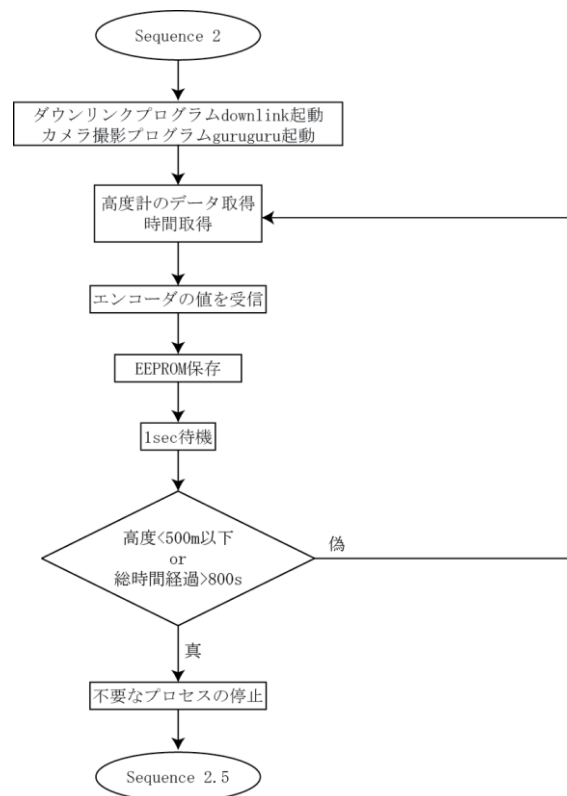


図 36 (g) Flow chart of sequence 2 routine. And in Sequence 3, main program work in almost similar fashion.

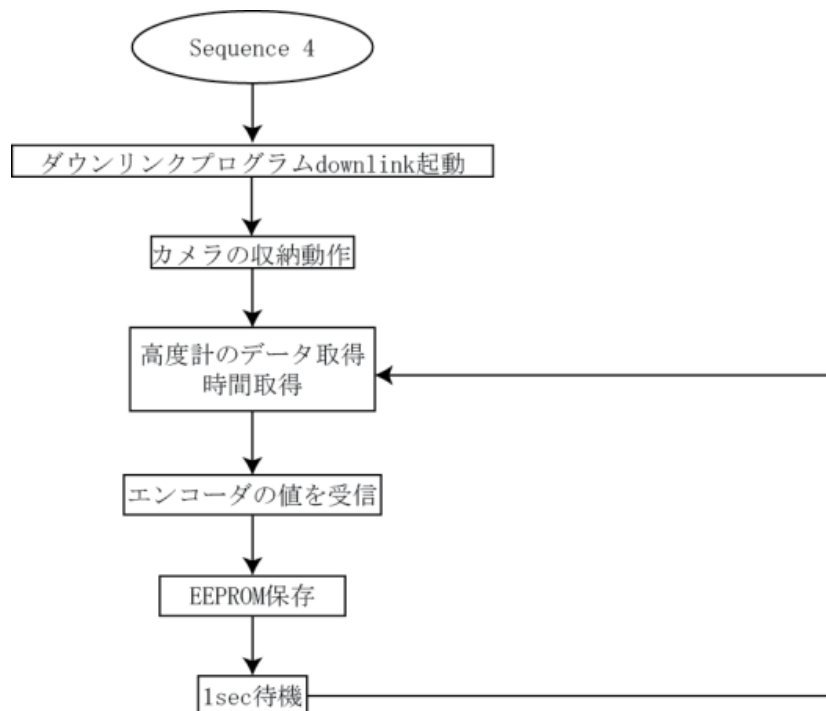


図 36 (h) Flow chart of sequence 4 routine.

3.4.5 センサ系

3.4.5.1 センサ系概要

本カンサットには下表に示した 3 種類のセンサ(子機は GPSR のみ)を搭載した.

表 4 搭載センサー一覧

種類	製品名	用途
GPSR	GH-84LB1F-C-N-B (古野電気)	緯度、経度取得
気圧センサ	SCP1000-D01 モジュール (秋月電子)	高度取得
地磁気センサ	LSM303DLH (ストロベリー・リナックス)	機体の方位角取得

それぞれのセンサの搭載理由は, GPSR はロスト対策や親機と子機の相対位置履歴を取得するため, 気圧センサはミッションシーケンス遂行において重要な高度データを高精度に取得するため, 地磁気センサは, 画像認識ミッションにおいて, 取得した画像データ以外にミッションの成功度合を評価できるようなデータを取得するためである.

以下 3.4.5.2~3.4.5.4 では各センサの詳細, 利用法について述べる.

3.4.5.2 GH-84

GH-84(古野電気)の仕様を以下の表に示す. この素子の選定理由としては, 1) 松永研究室で長年使用されてきた実績やノウハウがあること 2) 電源を投入するだけで毎秒 GPS データが送られてくるため, 扱いやすいことなどが挙げられる.

表 5 GH-84 仕様

インターフェイス	UART(9600bps)
測定誤差	緯度, 経度 5~10m 以内 高度 10m 程度
受信感度	-157dBm
電源電圧	2.9V~3.6V
消費電流	76mA~100mA
測位方式	Standard Positioning Service
測位更新周期	1 秒
データフォーマット	FEC バイナリ(23Byte 長)



GH-84

この GPSR は独自のデータフォーマット(図 37 参照)に則って経度、緯度、海拔高度のデータを含んだ毎秒 23byte のデータを出力する. この 23byte のデータのうち必要なデータを適宜抜き出して利用することになるが, 例えば緯度データの場合, データを構成する各 byte デ

ータの先頭 1bit を除いた 7bit を 4 つ繋げて一つの 28bit データとし、指定された計算式で変換することで緯度データとなる。

これは経度や海拔高度データにおいても同様であり、同じく複数 byte でデータが構成される速度や方位データにおいても同様な操作が必要である。

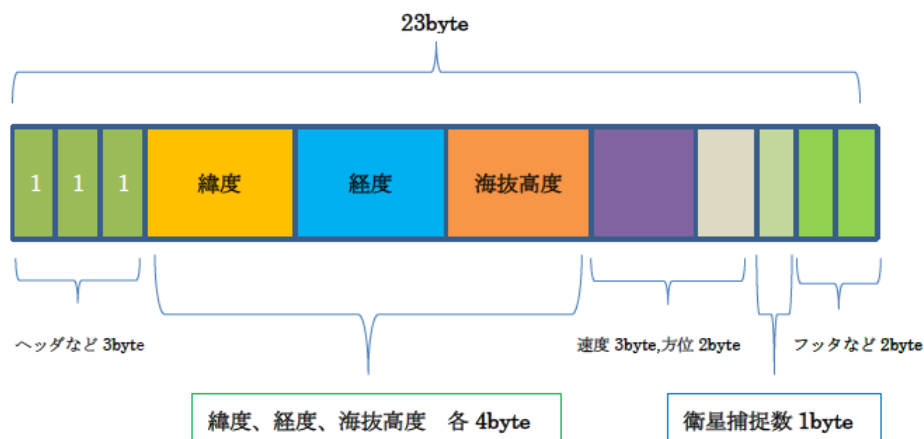


図 37 GH-84 データ構成

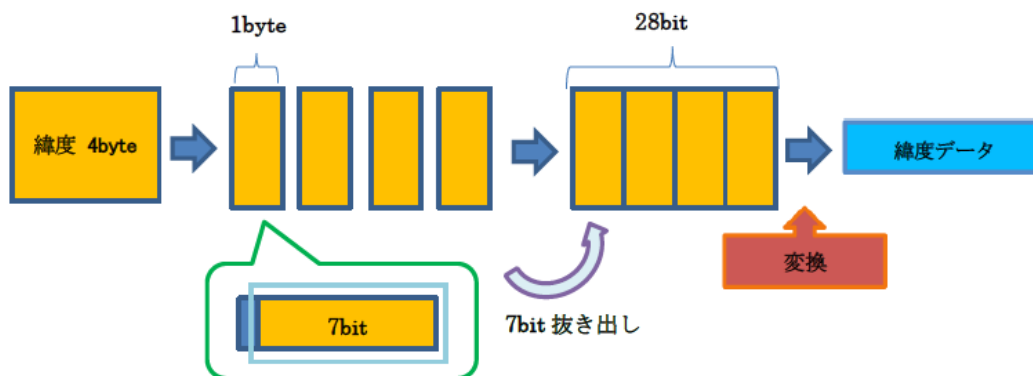


図 38 各種データの利用法(図は緯度データの場合)

なお、緯度や経度が南緯、西経になる際には各データは負の値となり、変換には 2 の補数を取る必要がある。この際の注意点として、例えば long 型の変数に緯度や経度の値を格納した場合、単純にこの変数に対して 2 の補数を取るのではなく、あくまで 28bit データに対して 2 の補数を取る必要があり、先頭の 4bit は無視しなければならない点が挙げられる。これを考慮しないと値が予想だにしないものになるので注意しなければならない。

3.4.5.2 SCP1000-D01 モジュール

SCP1000-D01 モジュール(秋月電子)の仕様を以下に示す。この素子の選定理由としては、1)高分解能であること、2)松永研究室で使用された実績があること、などが挙げられる。

表 6 SCP-1000-D01 モジュール仕様

インターフェイス	SPI
測定範囲	30kPa~120kPa
測定分解能	1.5Pa
電源電圧	2.4V~3.3V
消費電流	25μA



SCP-1000-D01

このセンサとの通信には SPI 通信が使用されるが、その際特に注意する必要があるのは、SPI 信号線の回路構成である。SPI 信号線は、MISO ,MOSI ,CS ,SCK の 4 種類で構成されるが、この内 MOSI ラインはプルアップが必須である。また MPU として FPGA を使用する場合には、FPGA の信号レベルが強いことによるオーバーシュート現象が顕著に見られるため、そのまま信号ライン(MOSI や SCK)を接続するとセンサ側が信号を読み取ることが出来ない。そのため、信号線にコンデンサ(40pF 程)を入れるなどして信号を鈍らせる必要がある。以下参考までに回路図を示す。

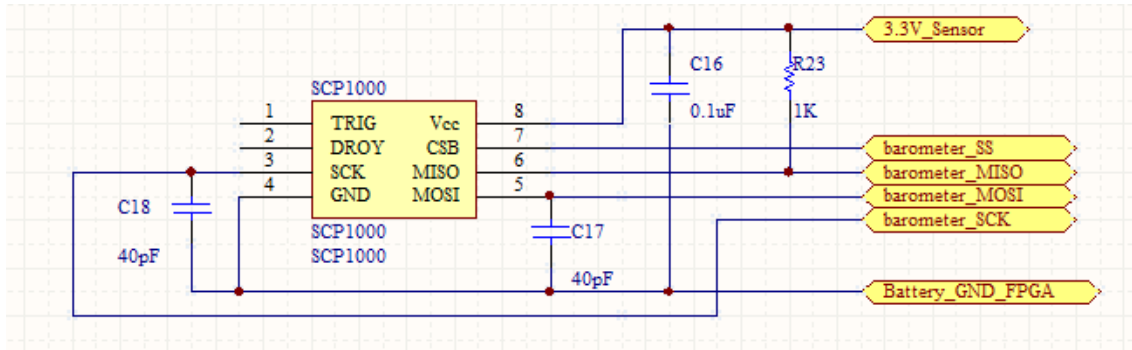


図 39 参考回路図(MPU として FPGA(SUZAKU)を用いた場合)

次にソフトウェア構成について説明する。

1.初期設定

始めにセンサの初期設定を行う。この初期設定では、測定モード(高速モード、高精度モード、低消費電力モード、測定停止モードなど)の設定を行う。具体的にはセンサの OPERATION レジスタに選択する測定モードに対応する値(例えば高精度モードならば 0x09)を書き込むことによってモードが設定される。

2.測定データ取得

初期設定が終わると、センサはデータ値を出力するようになる。実はこのセンサ、温度データも計測できる優れもの(しかも高精度)であり、データ出力要求の際に温度データが

格納してあるレジスタのアドレスを指定することで温度データも取得することができる。

基本的なデータ取得のフローチャートは図 40 に示した通りである。このセンサは測定値が更新される度にセンサの **DRDY** ピンが立ち上がり, **STATUS** レジスタの **DRDY** ビットも連動して 1 となる。データ取得シーケンスのスタートはこのデータ更新を待つことから始まり, その後温度データ, 気圧データの値が格納されているレジスタの値を読みに行くことになる。その際センサの仕様上温度データ→気圧データの順に読み出す必要があるので注意する。

温度データおよび気圧データはそれぞれ 14bit 長(符号込み,0.05℃/bit),19bit 長(符号なし,0.25Pa/bit)のデータとなっており, 温度データが一つのレジスタ(**TEMPOUT** レジスタ,16bit)に格納されているのに対して,気圧データは二つのレジスタ(**DATARD8** レジスタ 8bit,**DATARD16** レジスタ 16bit)に分割されて格納されている。そのため, 温度データは 16bit データのうち下位 14bit を抜き出せばいいのに対して, 気圧データに関してはデータの上位 bit 情報を含む 8bit データのうち下位 3bit を抜き出し,下位 bit 情報を含む 16bit データを合わせて 19bit のデータにするといった操作が必要となる。

なお詳細はセンサのデータシートの方を参考にされたい。

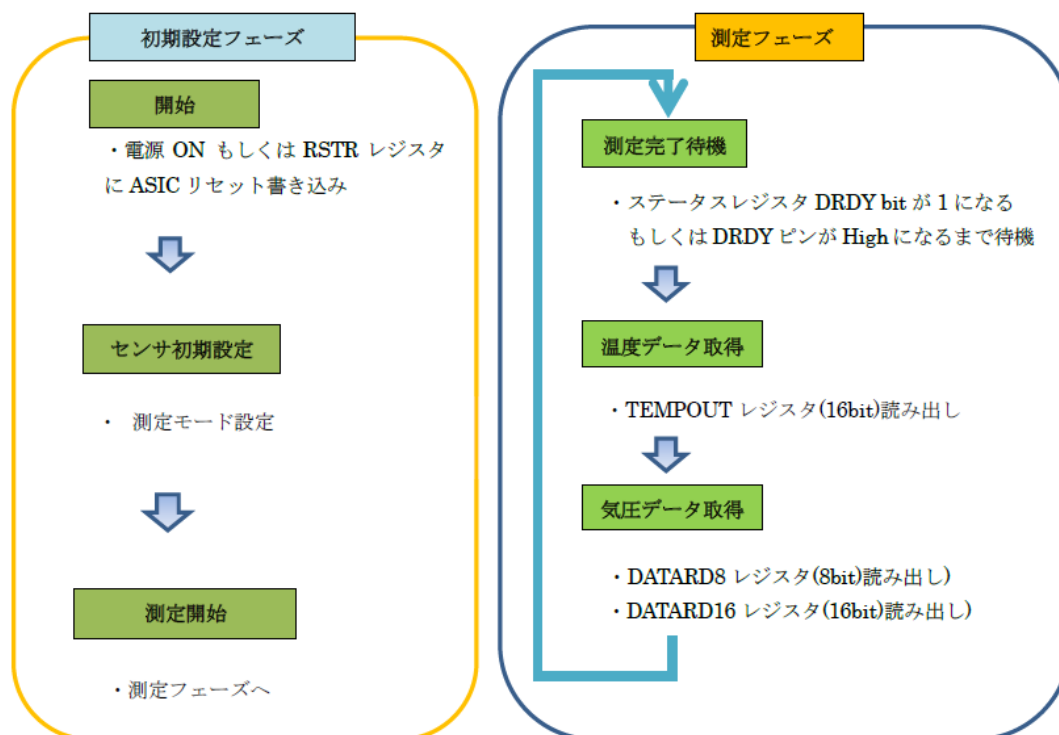


図 40 ソフトウェア構成の概略図

また, 気圧値を高度値に変換する式は, 打ち上げ高度が 4000m 以下と対流圏内にあることから, 大気的气温減率は一定とみなせるので, 大気を気温減率 6.5K/km の多方大気と仮定して状態方程式と静力学方程式から求められる次式

$$P = P_0 \{1 - 0.0065h / (t_0 + 273.15)\}^5.257$$

を用いた.

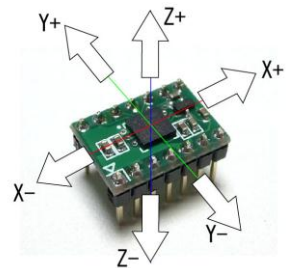
3.4.5.3 LSM303DLH

このセンサの搭載理由としては前述した通り,画像認識ミッションにおいて, 取得した画像データ以外にミッションの成功度合を評価できるようなデータを取得するためである.

より具体的にはこのセンサで取得される方位角と親機および子機の GPS データから算出される相対位置履歴とを合わせて解析することで,飛行中のカンサットの姿勢状態やカメラアームの制御状態をシミュレートするために搭載した. 以下にこのセンサの仕様を示す.

表 7 LSM303DLH 仕様

インターフェイス	I2C
測定軸	3 軸(地磁気,加速度)
測定範囲	$\pm 1.3/\pm 1.9/\pm 2.5/\pm 4.0/\pm 4.7/\pm 5.7/\pm 8.1$ (Gauss) ソフトウェア選択(地磁気) $\pm 2g/\pm 4g/\pm 8g$ ソフトウェア選択(加速度)
測定分解能	12bit
電源電圧	2.5V~3.3V
消費電流	1mA 以下



このセンサは地磁気センサではあるが, 加速度センサも搭載しているため, 地磁気とともに加速度も計測することができる. しかし, 今回は加速度を取得する必要がなかったもので地磁気の測定用のみに使用した.

通信方式には I2C を用いる. 周辺回路で注意すべきは, 信号ライン(SCL, SDA)をプルアップすることである. モジュール内ではプルアップしていないので必ず外部プルアップが必要である.

以下ではソフトウェアの概要について述べる. このセンサも気圧センサと同様, まず初めに測定モードや測定範囲,出力レートの設定といった初期設定を行う. なお,デフォルトの測定範囲は ± 1.3 (Gauss),出力レートは 15(Hz)となっている(今回は変更しなかった). これらの設定は必要に応じて適宜変更すればよいが, 一方でセンサを起動させるために必ず **operating mode** の設定をする必要がある. **operating mode** としては以下の 3 種類があり, デフォルトでは **sleep mode** となっているので動作させるには上の二つのどちらかに設定する必要がある.

表 8 operating mode

mode	
Continuous-conversion mode	連続してデータが更新されるモード.
Single-sonversion mode	単一のデータを読み出すモード. 読み出した後は自動的に Sleep mode となる
Sleep mode	スリープモード、待機モード

以上のような初期設定を行った後は，実際にデータ値を読み出す段階に入る．読み出しの際には，データが格納されているレジスタのアドレス値を指定し，読み出すといった作業を 3 軸すべてに行う．なお，各軸のデータ値は上下 8bit に分かれて格納されており，計 6 つのレジスタの値を読み出し,各軸のデータを上下合わせて 16bit のデータとすることで 3 軸の地磁気データ値を得る．

以上をまとめたのが以下の図である．詳細はデータシートを参考にされたい．

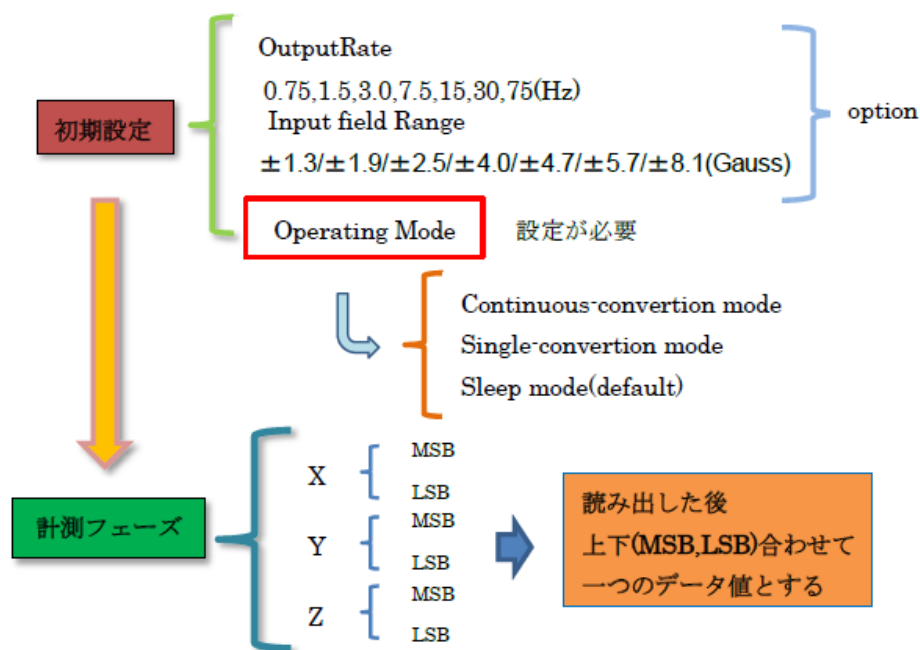


図 41 ソフトウェアの概要

3.4.6 メモリ系

今回のカンサットではメモリとして,親機は microSD カード(以下では簡便のため SD カードと記す)と EEPROM,子機は EEPROM のみを使用した．それぞれの使用用途は以下の表の通

りである。

表 9 メモリ使用用途

親機	microSD カード	写真画像データ保存, 各種データ保存,制御履歴保存,シーケンス番号保存
	EEPROM (24LC512)	バックアップ用：直前のエンコーダ値,シーケンス番号等保存
子機	EEPROM (24LC512)	GPS データ保存など

親機のメモリの一つとしてSDカードを選定した理由としては1)画像認識ミッションの要求から高通信速度かつ高容量なメモリが必要なこと 2)パソコンで容易にデータの確認が行えるけとなどが挙げられる。

一方で子機のメモリや親機のバックアップ用メモリとしてEEPROMを選定した理由としては, 1)保存するデータがGPSデータや直前のエンコーダ値などそれ程容量を必要としないデータであること 2)小型で比較的簡単に扱える記憶媒体であることなどが挙げられる。

以下にそれぞれのメモリ仕様を示す。

表 10 メモリ仕様

	microSD カード(Transcend)	EEPROM(24LC512)
インターフェイス	SPI	I2C
容量	4GB	64kB
通信速度	20MB/s	10KB/s 程度

なお、SD カードとの通信には Windows ファイルシステム(FAT)に準じた形式のデータのやりとりが行える SD カードモジュール MSC-MOD10(マイクロテクニカ)を用いた。データの保存はこのモジュールと UART 通信を行うことで行い，画像データはJPEG形式として，その他センサデータ値等は TEXT データとして保存する。以下に通信方式の概略図およびモジュールの仕様を示す。



図 42 microSD カードとの通信方式

表 11 MSC-MOD10 仕様

インターフェイス	UART(I2C も可)
通信速度	9.6kbps(初期)~921.6kbps(最大)
電源電圧	3.3V
消費電流	38mA(フルパワーモード時)



MSC-MOD10

次にこの SD モジュールを取り扱うためのソフトウェアについて概要を述べる。以下では具体的にファイルを生成するまでの手順を簡単に説明する。

1. SD 初期化

まず,SD カードの初期化を行う。ここで言う初期化とは,SD カードの保存内容をリセットするという意味ではなく,あくまでファイルシステム(FAT)を扱えるようにするための初期設定を指す。操作自体は簡単でシリアル通信で「I」コマンドを送ればよい。

2. 通信速度設定

ここでは通信速度設定を行う。この操作はオプションで初期通信速度 9600bps のままであれば行う必要はない。今回のミッションにおいてはカメラ系からの要求速度が 460.8kbps であったため,それに対応すべく通信速度を変更させる操作を行った。この際カメラ系と同様にマルチプレクサを用いて,配線をスイッチングすることによって通信速度を変化させている。これは FPGA にはソフトウェア上で同出力ピンの通信速度を変化させる機能がないためである。

なお,コマンドとしては例えば 460.8kbps に設定する場合「B 1EF4」のように送信する。(詳細はデータシート参照)

3. ファイルオープン

ファイルを生成し,そのファイルを扱える状態にする操作を行う。具体的にはモード(書き込みか読み出し)やファイル名,モジュールがファイルを区別するために用いるファイルハンドルを設定することによってファイルが作成され,書き込める状態となる。コマンドとしては例えばファイルハンドル 0 に対して TEXT.txt という名前のファイルを生成する場合「O 0W>TEXT.txt」のように送信する。なお,読み出しモードの際は W が R となる。

4. データサイズ指定

ファイルが生成されたら実際にデータを書き込む操作に移るが,書き込むデータを送信する前に書き込むデータのデータサイズを指定する操作を行う必要がある。コマンドとしては例えば 11 バイトのデータを書き込む場合,11 を 16 進数で表すと 0x0B なので「W 0>B」のように送信する。(ASCII コードで B と送信する必要があるので注意)

5. データ送信

4 の操作が終わるとモジュールは受信待機状態となるので，実際に書き込むデータを送信する．

6. ファイルクローズ

一連の作業が終わり次第，最後にオープンしたファイルをクローズする作業を行う．データはファイルクローズ作業を終えることで正常に書き込まれ，これを怠ると書き込んだデータが全て消去されるので注意する．コマンドはファイルハンドルを指定して「C 0」のように送信する．

以上の一連の操作を行うことでデータが書き込まれたファイルが生成される．再びファイルを生成する際には，3～6 の作業を繰り返せばよい．ただし，一度モジュールの電源を切ると設定が初期設定に戻り，手順 1 から行わなければならないので注意する．

以下に一連のシーケンスを図式化した概略図を示す．

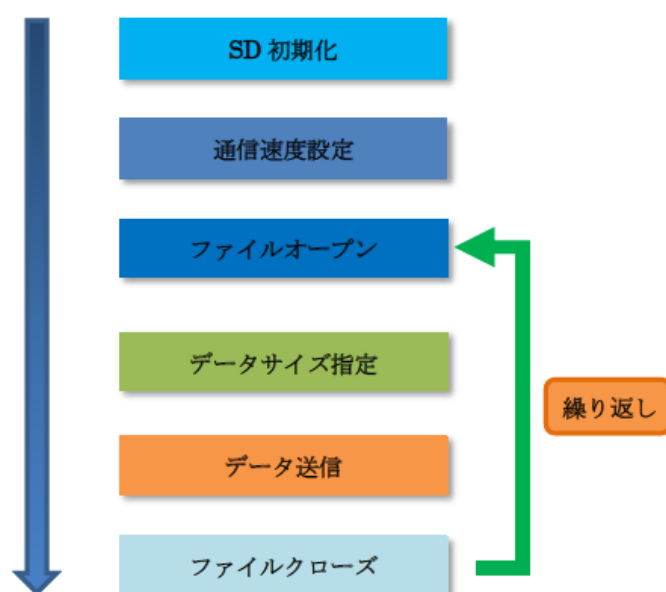


図 43 ファイル生成シーケンス

ファイル生成の手順は以上の通りで，実際にセンサ値や画像データを保存する際にも同様な手順でファイルを作成している．

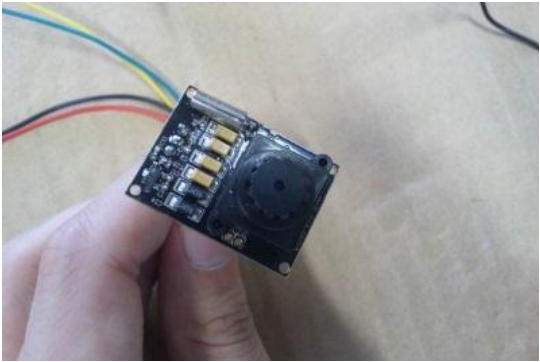
その他ソフトウェア上で工夫した点は，画像データを扱う際に一回のファイル生成シーケンスにおいて扱うデータ量を少量(200Byte 程度)に制限して，20kByte のデータを分割しながら順次ファイルに追記していくような方法を採用した点が挙げられる．この方法を採用した理由は，20kByte 程度の高容量データを 460.8kbps の高通信速度で一度に送信すると，モジュールの動作が不安定になり，データ落ちやファイルがクローズされないという現象が頻発

したためである。なお、この方式を取るとファイルオープンなどの操作が繰り返し必要になるため、通信速度が減少してしまうが、この方式を採用してもカメラ系の要求保存速度(1秒間に2枚保存できる程度の速度)は満たせることは実験で確認している。

3.4.7 カメラ系

本機体で子機の指向撮影に使用する JPEG カメラの仕様を表 12 に示す。

表 12 カメラ仕様

名称	C1098 JPEG Module
写真	
寸法	20×28[mm]
画像の種類	JPEG
画像サイズ	80×64[pixel](選択可能)
電源電圧	3.3[V]
通信方式	UART
ボーレート	最高:460800[bps](初期化時:14400[bps])

本カメラの選定理由としては、1)画像が JPEG 圧縮されるため転送するデータ量を小さくすることができる、2)通信方式が UART であり扱いやすい、3)使用実績のあるカメラの後継機であり、過去のノウハウを生かせる、等が挙げられる。

次に本カメラの駆動方法について述べる。システム図を図 44 に示す。最初にボーレート 14400[bps]の UART によるコマンド(例 0xAA,0x01,0x02,0x07,0x00,0x01)を用いてカメラの初期化を行う。このフェーズは実際に取得する画像のサイズ、転送速度といった情報を定義するものであり、開発ソフトにデフォルトで組み込まれている UART の IP コアではボーレート 14400[bps]を選択することができないため、この動作には簡易的に UART を模擬した自作 IP コアを使用する。一連の初期化コマンドを送信し終えたらマルチプレクサ(素子名:ADG774BR)によって機械的に配線をスイッチングし、高速通信による撮影フェーズに移行する。通信速度はカメラの出せる最高性能である 460800[bps]とし、この動作にはデフォルトで組み込まれている IP コアを使用する。撮影フェーズに移行したら、撮影コマンド・画像取

得コマンド・連写指示コマンド等を周期的に送信することで子機指向撮影のための連続撮影を行う。

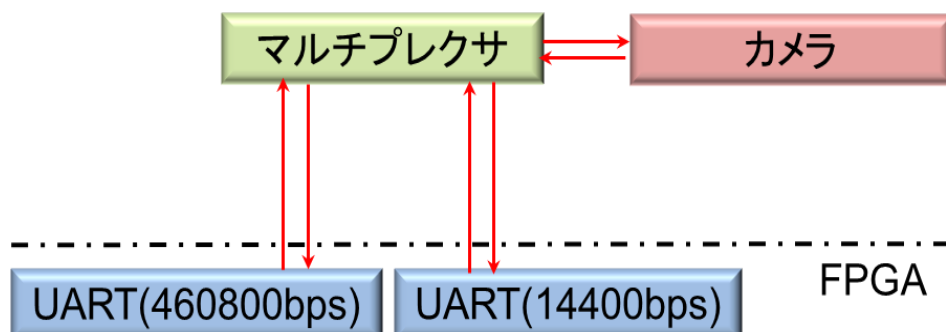


図 44 カメラ系のシステム図

撮影した画像データは、受信データの中からヘッダーとフッターを読み込むことで抽出される。

3.4.7.2 画像処理関連(FM 用プログラムには実装されていない)

子機指向撮影のためのモータ制御値を算出する画像処理には web 上に公開されている OpenCV ライブラリを用いる。具体的に開発環境を整える手順としては以下になる。まず Linux 環境のプログラム上で JPEG 画像の入出力を行うために libjpeg と呼ばれるライブラリが必要になるため、この libjpeg を powerpc-linux 向けにインストールする。続いてこの libjpeg をデフォルトで用いるように設定した上で OpenCV を powerpc-linux 向けにインストールする。こうしてインストールされた OpenCV ライブラリは、SUZAKU 上で動作させるプログラムの中で用いることが可能となり、その関数によって JPEG 画像を扱うことができるようになる。

実装された画像処理プログラムは JPEG カメラによって撮影された画像から 1).赤色部分検知, 2).二値化, 3).赤色部分の重心算出 を行う。赤色を扱う理由は砂漠での検知精度を上げるためであり、実際に検知するパラシュートは赤色のものを使用している。赤色部分の重心座標をモータ制御用アルゴリズムに入れ込み、2 軸のモータ制御を行う。以下に JPEG カメラによって撮影された画像(サンプル)からの二値化の様子を示す。

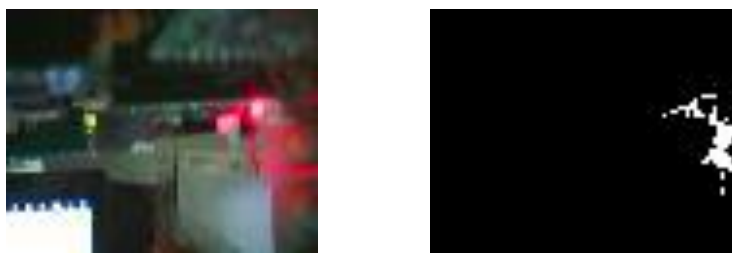


図 45 撮影された画像(左)と 二値化後の画像

この OpenCV ライブラリを使用するためにはあらかじめフラッシュメモリにライブラリを保存しておく必要があるが、実際にはライブラリは膨大な容量であり、フラッシュメモリの保存可能容量を超えてしまう。したがって OpenCV ライブラリはあらかじめ搭載する microSD カードに保存しておき、SUZAKU 起動後にこれをダウンロードして使用する。

開発状況としては、OpenCV ライブラリのダウンロードと撮影画像からの赤色部分の重心算出は完了していたが、統合試験が不十分であり、advanced success ということもあり、今回は画像処理機能の実装を断念した。

3.4.8 子機系

本カンサットにおける子機は以下の機能を有する。これはミッションからの要求によるものと、本ミッションでは子機を親機から完全に分離するため、レギュレーションからの要求によるものである。

- ・ 親機が子機をテザーで進展させた後、500 秒後にテザーを溶断して親機と完全に分離する機能(ミッション要求)
- ・ GPS を搭載し、データを取得するとともに通信機により地上局へ送信する(レギュレーション要求)

以上の機能を実現するため、子機のシステムは既述のシステム・ダイアグラムのように構成した。GPS・通信機・溶断回路・EEPROM は親機と同じものを用いている。メイン CPU は親機と異なり PIC16F886 を搭載した。以下に仕様を示す。

表 13 PIC16F886 仕様

電源電圧	2.0~5.5V	 PIC16F886
プログラムメモリ	8K ワード	
RAM	368 byte	
I/O ポート	最大 24	
A/D コンバータ	10bit×11	
周辺機能	ECCP, CCP, EUSART, MSSP, コンパレータ	

選定理由としては、当研究室で使用実績があること、電力消費量も小さく、子機として最低限の機能を有すること、小型であること、である。

子機のシーケンスを図 46 に示す。実現すべき機能は 2 つ(溶断と GPS データ取得・送信)のみであるので、極めて単純なシーケンスとなっている。

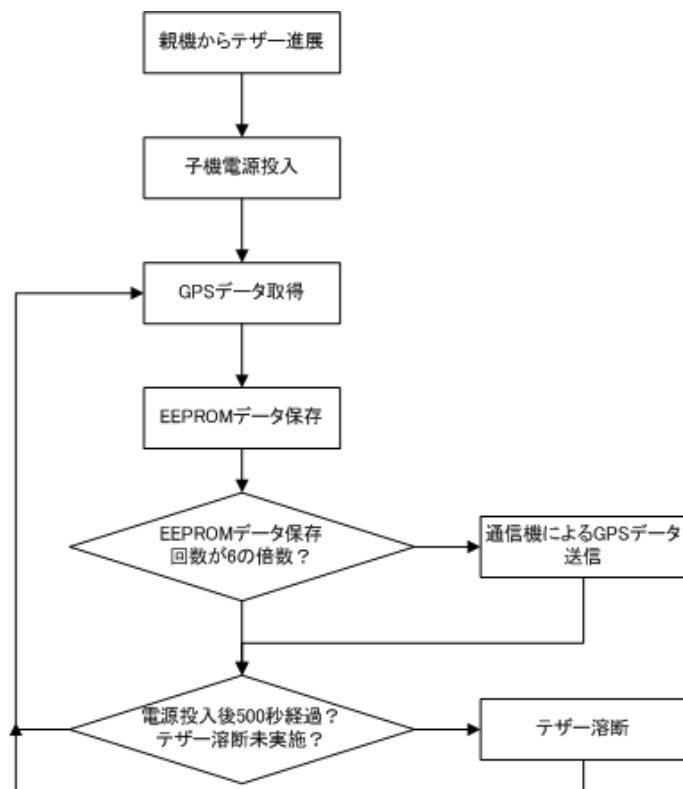


図 46 子機シーケンス図

上記シーケンスのループは 1.5[sec]程度で回り、この周期で通信機により毎回 GPS データを送信する必要はないため、およそ 10 秒に 1 回送信となるよう通信機の送信回数をループ回数の 6 分の 1 に設定している。

4. サクセスレベル

4.1 サクセスレベルの定義

表 14 にサクセスレベルを示す。なお、子機のサクセスレベルは親機との分離が成功して初めて実現できるため、すべての項目がフルサクセスとなっている。

表 14 サクセスレベル

	項目(親機)	項目(子機)
Minimum	・OSを用いた衛星バスの構築 ・パラシュート展開 ・センサデータ取得/保存 ・GPSデータダウンリンク	
Full	・テザー伸展(1次分離) ・テザー溶断(2次分離)	
	・カメラ画像取得 ・カメラアームの駆動制御	・パラシュート展開 ・センサデータ取得/保存 ・GPSデータダウンリンク
Advanced	・画像処理によるターゲット衛星位置取得 ・ターゲット衛星指向撮影	

4.2 サクセスレベルの評価方法

表 14 に示したサクセスレベルの評価方法を示す。

□ Minimum

① OS を用いた衛星バスの構築

- デバイスドライバ(UART, SPI, I2C, I/O)の開発
- 開発報告書の作成による技術蓄積

② パラシュート展開

- 衛星回収後、目視確認
- 気圧計から算出した落下速度が 5m/s(TBC)以下

③ センサデータ取得・保存

- GPS, 気圧計,コンパスの値が正常に取得できている
- 取得したデータがメモリに保存できている

④ GPS データダウンリンク

- GPS データが地上局に正常ダウンリンクできている

□ Full

- ① テザー伸展(1次分離)
 - 分離機構動作の制御履歴確認
 - 衛星回収後，目視確認
- ② テザー溶断(2次分離)
 - 溶断機構動作の制御履歴確認
 - 衛星回収後，目視確認
- ③ カメラ画像取得(主衛星)
 - 少なくとも1枚以上の撮影画像がメモリに保存できている
- ④ カメラアームの駆動制御
 - エンコーダ履歴を確認(カメラアームが指令値を追従できている)
- ⑤ ターゲット衛星
 - 主衛星の Minimum 評価方法と同様

□ Advanced

- ① 画像処理によるターゲット衛星位置取得
 - 取得/保存した画像データと画像処理演算結果を比較し，方向が 30deg 以内で特定できている
 - 主衛星/ターゲット衛星の GPS データを比較し，方向が 30deg 以内で特定できている
- ② ターゲット衛星指向撮影
 - 画像処理からの指令値通りにカメラアームが駆動できている
 - 撮影した画像の中央付近にターゲット衛星が捕捉できている

5. ARLISS2011 試験結果

5.1 1st Flight 結果

第1回試験のタイムラインを表15に示す。打ち上げ(Bob)は成功、カンサット放出高度は約3000mであり、親機、子機ともに回収に成功した。

表15 1st Flight タイムライン

-33s	打ち上げ
-3s	ロケットからの放出
0s	親機パラシュート展開 ～ミッションスタート～
1min 55s	子機1次分離(テザー伸展) 親機メインアプリケーション停止
10min 15s	子機2次分離(テザー溶断) 子機パラシュート展開
12min 56s	子機着陸
14min 27s	親機着陸



以下に詳細な試験結果を示す。

□ ◯正常動作(親機)

- 1)パラシュートは正常に展開し、回収後の目視点検でも破損は見られなかった。子機GPSデータから計算した落下速度は約4.0 [m/s]であり、およそ設計値通りの結果が得られた。
- 2)フライトピンは正常に動作し、回収後の目視点検でも破損は見られなかった。
- 3)子機分離機構は正常に動作し、回収後の目視点検でも、機構、および溶断回路に破損は見られなかった。
- 4)テザー伸展機構は正常に動作し、分離からテザー伸展までの様子を mirumiru の動画により撮影することができた。回収後の目視点検でも、機構、および溶断回路に破損は見られなかった。
- 5)ロケット振動、あるいは落下の衝撃による主構造に大きな歪み・破損箇所は確認されなかった。

□ ◯正常動作(子機)

- 6)フライトピンは正常に動作し、回収後の目視点検でも破損は見られなかった。
- 7)テザー溶断回路は正常に動作し、親機との完全分離に成功したが、回収後、ニクロム線が一部破断していることを確認した。
- 8)パラシュート用溶断回路は正常に動作し、回収後の目視点検でも破損は見られなかった。また、パラシュートも正常に展開し、回収後の目視点検でも破損は見られなかつ

った．子機 GPS データから計算した落下速度は約 4.5[m/s]であり，およそ設計値通りの結果が得られた．

- 9) ロケット振動，あるいは落下の衝撃による主構造に大きな歪み・破損箇所は確認されなかった．
- 10) GPS データの EEPROM への保存，およびデータダウンリンクに成功した．

□ ×正常動作せず(親機)

- 1) 地上局無線機の不具合により，ダウンリンク音は聞こえるも，データがデコードできなかった．
- 2) SD カードのイニシャライズに失敗し，データ保存用のテキストファイルが生成されなかった．
- 3) シーケンス I から II への移行時，メインアプリケーションが停止した．以降，全プロセスが動作せず，カメラ撮影，及びカメラアームの駆動を行うことができなかった．
- 4) 回収後，モータが内部故障していることを確認した(目に見える破損はなし)．故障したモータは新規購入品ではなく，研究室で昔購入したものであったので，寿命の可能性もある．

図 47 に，mirumiru で撮影した子機テザー伸展の瞬間の画像を示す．

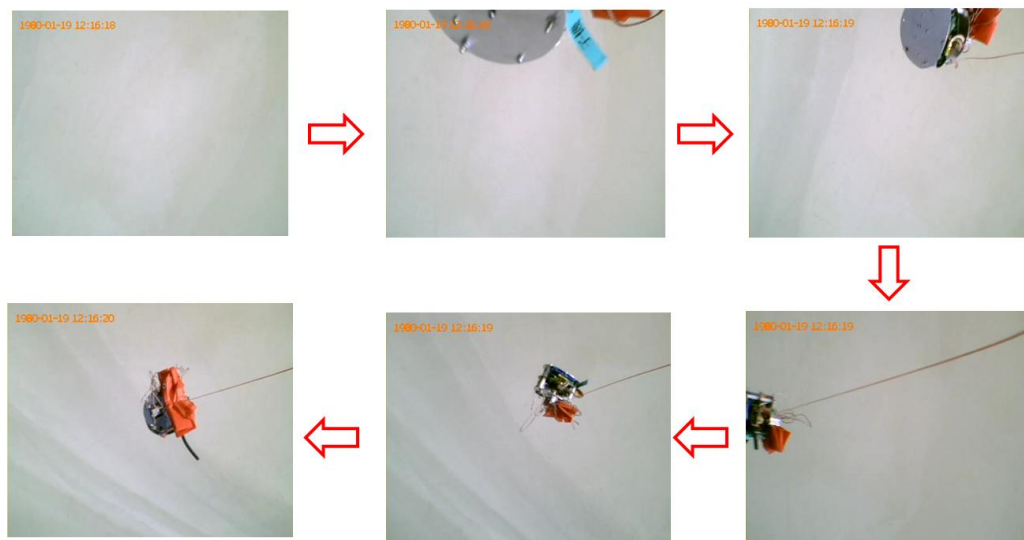


図 47 子機テザー伸展の瞬間をとらえた mirumiru のキャプチャー画像
(撮影時間は写真左上に記載)

5.2 1st Flight デバッグ

正常動作しなかった箇所について、下記のように考察・対処を行った。

- 1)地上局無線機の不具合により、ダウンリンク音は聞こえるも、データがデコードできなかった。

<考察・対処>

地上局の無線機を交換した上で、地上での通信試験を行い、ダウンリンクしたデータが正常にデコードできることを確認した。

- 2)SD カードのイニシャライズに失敗し、データ保存用のテキストファイルが生成されなかった。

<考察・対処>

まず、以下の理由により Linux は正常に起動していたことが推測される

- 子機を正常に分離している
- FM ダウンリンク音が聞こえる
- EEPROM に時間経過等が保存されている

Linux は起動していたことから、SD カードへ何もデータが保存されていなかったことの原因としては SD カードモジュールのイニシャライズに失敗したことが原因であると考え、SD へのイニシャライズコマンドに対する ack 返答の確認及びテストファイルの読み出しを行い、読み出しが成功するまでイニシャライズを繰り返すという手法を採用した。

- 3)シーケンス I から II への移行時、メインアプリケーションが停止した。以降、全子プロセスが動作せず、カメラ撮影、及びカメラアームの駆動を行うことができなかった。

<考察・対処>

Linux は起動していたことから、メインアプリケーションはシーケンス 1 終了後、子機を分離し、シーケンス 2 に入った直後に止まったことが考えられる。

- EEPROM のシーケンス番号が 2 と保存されている
- EEPROM の動作時間が 90 秒と保存されている

1st.フライト終了後、これらは SUZAKU が動作停止したことによるものだと考え、デバッグを行ったものの、再現性が見られなかった。そこで、信頼性を上げるという意味でも、フライトプログラムで動作確認を複数回行うということで、2nd.フライトへと臨んだ。

- 4)回収後、モータが内部故障していることを確認した(目に見える破損はなし)。故障したモータは新規購入品ではなく、研究室で昔購入したものであったので、寿命の可能性もある。

<考察・対処>

モータを新規購入品と取り換え，正常に動作することを確認した．

5.3 2nd Flight 結果

第 2 回試験のタイムラインを表 16 に示す．打ち上げは成功(Matt)，カンサット放出高度は約 3000m であった．風が非常に強かったため，親機，子機ともに大きく流され，1 時間以上捜索したのち，射点から 13km 離れた地点で回収した．

表 16 2nd Flight タイムライン

-32s	打ち上げ
-1s	ロケットからの放出
0s	親機パラシュート展開 ～ミッションスタート～
0s	子機1次分離(テザー伸展)
*min**s	親機メインアプリケーション停止
8min26s	子機2次分離(テザー伸展) 子機パラシュート展開
12min 30s	子機着陸
16min 15s	親機着陸(13km彼方で発見)

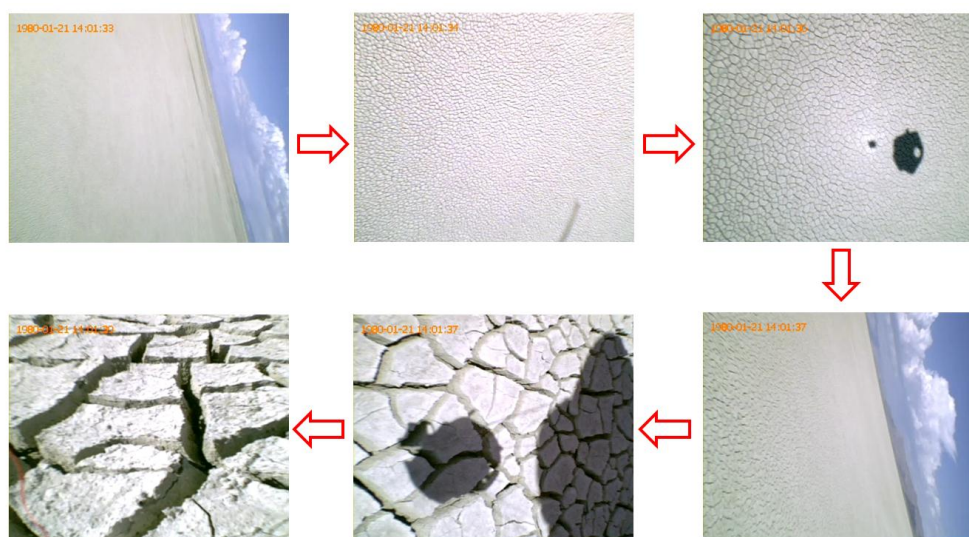


図 48 2ndFlight の mirumiru によるキャプチャー画像(風により大きく揺れている)
(撮影時間は写真左上に記載)

以下に詳細な試験結果を示す.

□ ○正常動作(親機)

- 1)パラシュートは正常に展開し、回収後の目視点検でも破損は見られなかった. 子機 GPS データから計算した落下速度は約 3.1 [m/s]であり, およそ設計値通りの結果が得られた.
- 2)フライトピンは正常に動作し、回収後の目視点検でも破損は見られなかった.

□ ○正常動作(子機)

- 6)フライトピンは正常に動作し、回収後の目視点検でも破損は見られなかった.
- 7)テザー溶断回路は正常に動作し、親機との完全分離に成功したが、回収後、ニクロム線が一部破断していることを確認した.
- 8)パラシュート用溶断回路は正常に動作し、回収後の目視点検でも破損は見られなかった. また、パラシュートも正常に展開し、回収後の目視点検でも破損は見られなかった. 子機 GPS データから計算した落下速度は約 4.5[m/s]であり, およそ設計値通りの結果が得られた.
- 9)ロケット振動,あるいは落下の衝撃による主構造に大きな歪み・破損箇所は確認されなかった.
- 10)GPS データの EEPROM への保存, およびデータダウンリンクに成功した.

□ ×正常動作せず(親機)

- 1)ロケットからの放出直後に子機分離機構が誤動作し、子機が分離していたことを mirumiru の動画により確認した.
- 2)SD カードのイニシャライズに成功したものの, 0kbyte のテキストファイルが生成されたのみで, SD カードへのデータ保存に失敗した.
- 3)GPS データダウンリンクに失敗した. 1stFlight とは異なり, 地上局ではダウンリンク音すら確認することができなかった.
- 4)1stFlight 同様, シーケンス I から II への移行時, メインアプリケーションが停止した. 以降, 全子プロセスが動作せず, カメラ撮影, 及びカメラアームの駆動を行うことができなかった.
- 5)着地時に天板が大きく変形し, アンテナが根本から折れた(図 49).

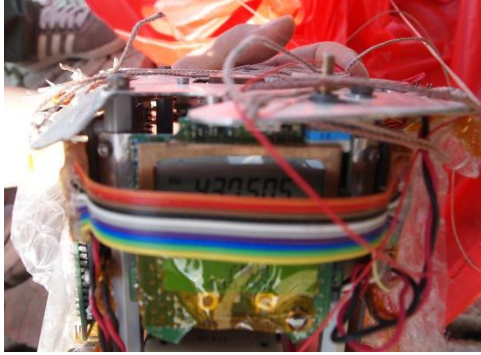


図 49 2ndFlight 直後の親機：左が曲がった天板，右が折れたアンテナ

図 50(a)(b)に，子機が取得した GPS データを解析した結果を示す．(a)は google map と照らし合わせた場合の移動軌跡，(b)は子機の高度履歴である．



図 50(a) 子機 GPS データによる子機の移動軌跡

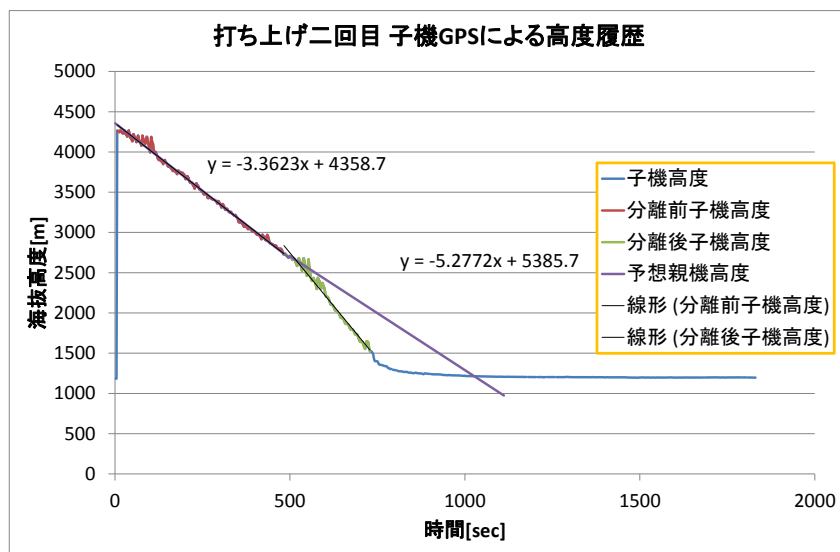


図 50(b) 子機 GPS データによる親機・子機の(予想)高度履歴

5.4 2nd Flight デバッグ

正常動作しなかった箇所について、考察を行った結果を以下に示す。

- 1) ロケットからの放出直後に子機分離機構が誤動作し、子機が分離していたことを mirumiru の動画により確認した。

<考察>

図 51 に、ロケットからの放出直後の mirumiru の動画フレームを示す。放出高度が 3000m を下回っている場合、シーケンス I を介さずに子機伸展を行う可能性がある。しかし、放出から 2 秒後には子機が確認されているため、Linux の起動時間、溶断にかかる時間を考えると、把持機構の誤動作と考えるのが妥当である。また、ロケット内でフライトピンが脱落している可能性もあるが、検証する術はない。



図 51 ロケットキャリアからの放出直後の mirumiru の動画キャプチャー
(撮影時間は写真左上に記載)

把持機構誤動作の原因の一つとして、把持機構に張力を与えるテザーが、打ち上げ中に緩み、ロケットからの解放とともに機構が動作してしまったことが考えられる。サブシステムの項でも述べた通り、今回の把持機構はテザーの張力管理が難しく、対応策としては、トルク管理を利用するなどの方法を考えるべきである。また、テザーのルーティングにも問題があり、テザーに張力が一様にかかっていなかった可能性が高い。対応策としては、ルーティングはなるべく鋭角で曲げるよう工夫し、ラウンドを大きくとるなどする必要がある。なお、2ndFlight のロケットキャリアへの搭載中、テザーの張力が緩み、把持機構が誤動作してしまうというハプニングも起きており、その場のドタバタに飲まれて上手くいかなかった、という側面もある。

2)SD カードのイニシャライズに成功したものの、0kbyte のテキストファイルが生成されたのみで、SD カードへのデータ保存に失敗した。

<考察>

原因は特定されておらず、現在デバッグ中である。デバッグの進捗に応じて、本書類をアップデートする予定である。

3)GPS データダウンリンクに失敗した。1stFlight とは異なり、地上局ではダウンリンク音すら確認することができなかった。

<考察>

回収時、無線機の電源は ON になっていたが、送信モードになっていなかった、原因は特定されておらず、現在デバッグ中である。デバッグの進捗に応じて、本書類をアップデートする予定である。

4) 1stFlight 同様、シーケンス I から II への移行時、メインアプリケーションが停止した。以降、全子プロセスが動作せず、カメラ撮影、及びカメラアームの駆動を行うことができなかった。

<考察>

原因は特定されておらず、現在デバッグ中である。デバッグの進捗に応じて、本書類をアップデートする予定である。

5)着地時に天板が大きく変形し、アンテナが根本から折れた。

<考察>

当日は風が非常に強く、mirumiru の動画によると、カンサットは大きく揺られながら地上に落下している。また、着地の際に、地面を数回転がっていることも動画により確認されたため、落下の衝撃が大きく、天板側から落ちた親機が変形した、と考えられる。アンテナが着地場所から発見されていることから、アンテナも落下衝突時に破損したものを推定される。

6. ミッション達成度の評価

5 章の結果を踏まえ、サクセスレベルの判定基準をもとに、ミッション達成度の評価を以下のように行った。青が達成、赤が未達成を示す。

□ Minimum

① OS を用いた衛星バスの構築(達成)

Linux を用いた OS 制御バスシステム構築を行い、開発報告書も作成したため、本項目は達成とする。

② パラシュート展開(未達成)

パラシュートによる正常展開は確認されたが、気圧計による親機の高度を測定することができなかったため、本項目は未達成とする。

③ センサデータ取得・保存(未達成)

SD へのデータ書き込みが行えなかったため、本項目は未達成とする。

④ GPS データダウンリンク(未達成)

地上局でのデータデコードができなかったため、本項目は未達成とする。

□ Full

① テザー伸展(1 次分離)

1stFlight では正常動作が確認できたが、2ndFlight では放出直後に誤動作を起こした可能性が高いため、本項目は 1stFlight では達成、2ndFlight では未達成とする。

② テザー溶断(2 次分離)(達成)

子機による制御履歴も取得できており、正常分離が確認されているため、本項目は達成とする。

③ カメラ画像取得(主衛星)(未達成)

メインアプリケーションが停止したため、本項目は未達成とする。

④ カメラアームの駆動制御(未達成)

メインアプリケーションが停止したため、本項目は未達成とする。

⑤ ターゲット衛星(達成)

全ての正常動作を確認したため、本項目は達成とする。

□ Advanced

① 画像処理によるターゲット衛星位置取得(未達成)

メインプログラムへの実装が間に合わなかったため、本項目は未達成とする。

② ターゲット衛星指向撮影(未達成)

メインプログラムへの実装が間に合わなかったため、本項目は未達成とする。

表 17(a)(b)に、サクセスレベルの達成度を示す。

表 17(a) 1stFlight のサクセスレベル達成度

	項目(主衛星)	項目(子機)
Minimum	・OSを用いた衛星バスの構築 ・パラシュート展開 ・センサデータ取得/保存 ・GPSデータダウンリンク	
Full	・テザー伸展(1次分離) ・テザー溶断(2次分離)	
	・カメラ画像取得 ・カメラアームの駆動制御	・パラシュート展開 ・センサデータ取得/保存 ・GPSデータダウンリンク
Advanced	・画像処理によるターゲット衛星位置取得 ・ターゲット衛星指向撮影	

表 17(b) 2ndFlight のサクセスレベル達成度

	項目(主衛星)	項目(子機)
Minimum	・OSを用いた衛星バスの構築 ・パラシュート展開 ・センサデータ取得/保存 ・GPSデータダウンリンク	
Full	・テザー伸展(1次分離) ・テザー溶断(2次分離)	
	・カメラ画像取得 ・カメラアームの駆動制御	・パラシュート展開 ・センサデータ取得/保存 ・GPSデータダウンリンク
Advanced	・画像処理によるターゲット衛星位置取得 ・ターゲット衛星指向撮影	

7. 総括

ARLISS2011 では、将来的な視野をもって、かなりチャレンジングなミッションに取り組み、本番では結果こそ残せなかったが、開発を通じて非常に多くの技術・知識を得、おそらくカンサットでは史上初の Linux の起動に成功した。最大の参加目的であった OS 利用の有効性については十分に確認できたため、今後は半年間で得た知見をドキュメント化し、技術伝承を行うことで、将来のより良いカンサット、人工衛星の製作を目指す。本ドキュメントはその一端も兼ねており、今後も内容をアップデートしていきたいと考えている。