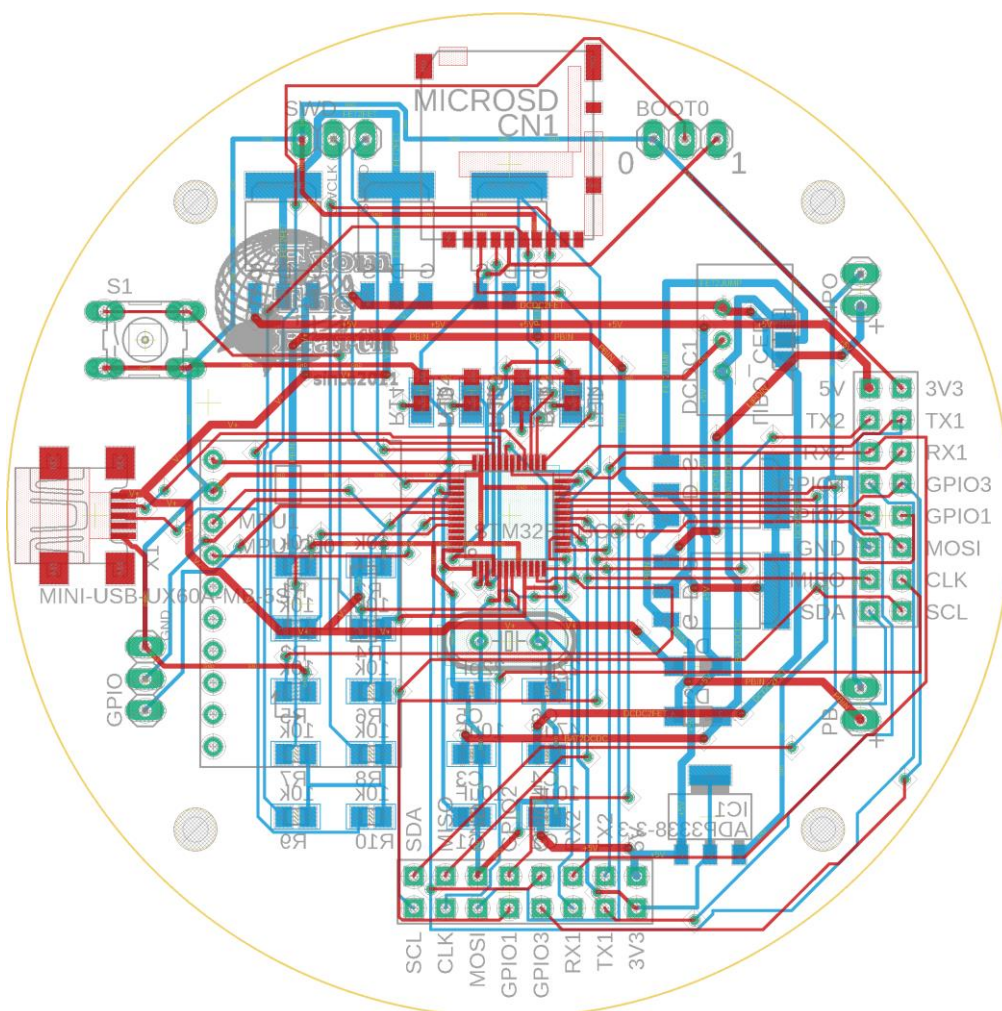


電装製作の手引き

10 期代電子班 電装担当

奥 檀



目次

1	電装とは	2
1.1	目的	2
1.2	構成	2
1.3	目標と優先順位	3
2	作成の手順	5
2.1	立案	5
2.2	選定	5
2.2.1	データシート	6
2.2.2	モジュール類	6
2.2.3	マイコン	6
2.2.4	電源	6
2.3	ハードウェアの設計	7
2.3.1	電源系統の設計	8
2.3.2	通信系統の設計	12
2.3.3	筐体の設計	15
2.4	ソフトウェアの設計	15
2.4.1	フローチャートの設計	15
2.4.2	プロトコルの設計	16
2.5	部品調達	16
2.5.1	調達方法	16
2.5.2	代表的な部品	18
2.6	作成	20
2.6.1	基板作成	20
2.6.2	プログラム作成	23
2.6.3	筐体作成	24
2.7	評価	25
2.7.1	完成した製品に対する評価	25
2.7.2	製作の過程に対する評価	26
3	今後実現したいこと	26
3.1	点火装置の改良	26
3.2	無線 GSE	26
3.3	冷却	26
3.4	アクティブ姿勢制御	27

1 電装とは

1.1 目的

電装の目的は大きく分けて記録と駆動の 2 つがある。記録は飛行中のロケットの位置や高度、速度といった情報を記録して回収することだ。これらのデータは次のロケットの製作に生かされることになる。また記録を取ること自体が打ち上げの目的になる場合もある。記録されるのは数値データだけにとどまらず、ビデオカメラによる空撮なども含まれる。また内部の記憶装置だけでなく無線通信機を利用して飛行中に外部にデータを送信するのも一般的である。

駆動はパラシュートなどの飛行に必要な動作のための動力を適切なタイミングで提供することだ。機械的機構だけでは例えば打ち上げ n 秒後にパラシュートの展開といった柔軟な動作は難しいので、ロケットの装備品の中で唯一飛行状態を感知している電装が動作のタイミングの指示（アクチュエータの制御信号）とエネルギーの供給（電流など）を行う。

1.2 構成

電装はロケットの搭載できる重さ、スペースなどの要因で構成を変化させる。ロケットの規模ごとに標準的と思われる構成を示す。単語については後述。

- ・ 最小構成（～G 型モデルロケット、ペットボトルロケット等）
マイコン、9 軸、気圧、記録装置
- ・ 陸打ち構成（J～K 型エンジン機、700m 程度）
マイコン、9 軸、気圧、記録装置、カメラ、アクチュエータ
- ・ 海打ち構成（K～L 型エンジン機、ハイブリッドエンジン機、1000m 以上）
マイコン、9 軸、気圧、GPS、記録装置、通信装置、水密、カメラ、アクチュエータ
- ・ CanSat（ランバック機）
SBC、電子コンパス、GPS、記録装置、通信装置、カメラ、アクチュエータ

規模が大きくなるほどより高度で複雑な構成となることがわかるだろう。マイコンとはマイクロコントローラの略で MCU とも呼び、CPU やメモリなどが一体となったもので電装の頭脳である。9 軸とはジャイロセンサ、加速度センサ、磁気センサ各 3 軸が 1 体となった部品で、マイコンに観測値を送信する。気圧とは気圧センサのことで同様にマイコンに情報を送信する。大気圧力の変化から現在高度が計算できるので、高度の記録やパラシュートを電装で展開する際には極めて重要な部品となる。GPS とは水平方向の位置を記録するためのセンサであり通信装置で緯度経度を地上側に送信することで落下後のロケットを回収することができる。記録装置とは観測値を記録するための装置で FLASH や EEPROM などもあるが、速度容量の面から microSD カードを使うのが普通である。通信

装置とは無線を用いて地上側にデータを送信する部品である。水密については後で詳しく述べるが防水されたケースのことで、海に落下したロケットの電装が浸水しないためのものである。カメラは飛行中の映像などを記録する装置で、電装と連携してなにかをすることは今の所ない。アクチュエータとはモータやソレノイドといった電気での物理的な動作を生じる装置である。パラシュートの展開時などに用いる。

おまけとして CanSat の構成も書いておいたがロケット電装と少々趣が異なることに注意されたい。SBC (Single Board Computer) とはラズパイのような IO ポートと OS を持った超小型のコンピュータのことである。CanSat では記録の他に GPS やカメラから自己位置を推定し、目的の場所までたどり着くためにより高い処理能力が必要となるためマイコンではなく SBC を搭載する。またアクチュエータにも歩行用の脚や、走行用のタイヤなどロケットより高度で複雑な制御が求められる。反面ロケットのようなコマ何秒の処理速度は求められないためソフトウェアの設計にも違いが出てくる。

1.3 目標と優先順位

ここでは最も機能の多い海打ち機体を念頭に説明する。ロケット電装において最も重要な目標は打ち上げ前、飛行中、落下中、着水後を通して確実に動作することである。これは電装が途中で停止してしまったりするとパラシュート開傘などに問題が発生し、ロケット本体だけでなく地上側にも影響する重大な事故が発生する可能性があるためである。この確実に動作することを「信頼性が高い」という。このためには様々な条件でテストを繰り返して不具合の発生原因を特定し、それを解決していくことが必要である。また、なにか問題（たとえば通信の切断など）が起こったとしても動作を継続できることが求められる。このような問題が発生しても動作を継続できる性能を「冗長性が高い」という。冗長性を高めるためソフトウェア、ハードウェアが連携して“自身が正常に動作していない”ことを検出できる仕組みを備える必要がある。この一つの手段としてタイムアウトについて後述する。実際のロケットや航空機、サーバなどではメイン系・サブ系、プライマリ・セカンダリとして非常時に切り替わる仕組みを持っていたり、同一のシステムを複数備える二重化、三重化を行ったりしている。電装が不安定になる要因としては一般的な順に電池切れ、衝撃による断線・接触不良、水密への浸水、ソフトウェアのエラー、瞬間的な電力不足、電気系統のノイズなどがあげられる。通信の切断を含めなかったのは、通信は切れるのが当たり前なのでそれで問題が発生するとしたら設計の不良だからである。

続いて重要な要素は小型軽量さである。エンジンパワーを 20%アップするのは大変なことだが機体重量を（必要なエネルギーが同等となるように）17%削減するのは比較的容易である。使用したい部品などの重量が機能に対してあま

りに大きい場合には軽量な代替品を検討したり搭載を諦めたりすることも必要である。しかし電子部品は多くが軽量なので重量が問題となることは少ない。むしろ小型化のほうが重要である。これは一見関係がなさそうに思われるかもしれないが実際の水密を含めた電装を手にとって見るとわかるだろう。電装部の重量の多くを占めているのは基板や部品ではなく水密の構造（カプラ、アクリルチューブ）や電池類である。縦 4 段基板の電装を 2 枚にすればアクリルチューブの長さが減って大幅な軽量化となる。また基板の小径化によって機体直径を 100mm から 90mm にできれば前方投影面積は約 20% の減となる。また機体の構造にもよるが、アクリルチューブを短くすることでロケット全体の剛性向上にも効果がある。小型軽量化の手段としては、より小型の部品を用いる（リード部品→チップ部品）、配線密度を上げる（ユニバーサル基板裏面に手配線→PCB 基板 4 層）、部品配置の工夫（パズルをイメージしてほしい）などがある。この他に電源系統の高効率化も挙げられる。たとえばバッテリーの 11V をマイコンの 3.3V に変換する際、変換効率 30% のレギュレータではなく 98% の DC-DC コンバータを使えば（回路が複雑になるのでその分も考慮する必要があるが）バッテリー容量を半分に減らすことができる。勘違いしてはいけないのは、信頼性の優先順位の方が上であるということである。すなわち軽量化のために安全のための部品を削減したり、バッテリーが不十分になったりしては本末転倒である。むしろ小型軽量化によって生じた余裕を信頼性に活かすくらいで良い（バックアップの搭載やバッテリー容量増加など）。

続いて機能性である。これはどれだけ多くの機能を持っているかということで、搭載しているセンサの種類やその精度、観測できるデータの量などである。これは上記の小型軽量性とトレードオフであり、そのロケットの目的によっては（高度は目指さずデータ収集を主目標とするなど）こちらが優先される場合もある。これはハードウェア的にどれだけ多くのセンサを搭載しているかだけでなく、ソフトウェアの高速化や表示のわかりやすさなども関係してくる。ソフトウェアの高速化や高度化は重量への影響がないので、打ち上げ直前以外でも通年して行うべき改良である。エンジンパワーを 20% アップするのは大変なことだがソフトウェアは工夫次第で 10 倍、100 倍の高速化、高度化が可能なので、プログラムは目立たない、地味などという印象を捨てて取り組んでほしい。

最後にコストパフォーマンスである。ロケット開発にはお金を惜しむべきでない、ケチくさいことをするなという人があるがこれは大きな誤りである。本当に開発費が湯水のように湧いてくるならそれに越したことはないが実際はそうではない。高価な電装が数機用意できるだけでは水没試験や落下試験など過激な試験をやる気が起きないし、もしかしたら試験をした（ダメージを負った）電装で打ち上げに挑むことになりかねない。できることなら 10 機ほど作成して水

没、衝撃、振動などあらゆる試験を実施し、本番もメイン+バックアップ2機から選べるくらいだと最高である。また値段を下げることで試作回数を増やすことができる。試作→評価→改善→試作のサイクルを何度も繰り返して信頼性を向上させることが重要である。肝心の方法についてだが、後述する調達の工夫（まとめて大量に購入する、購入する先を変更する）、設計の工夫（基板枚数や部品点数を減らす）、電子班全体での戦略的取り組み（基板の規格化共通化、ストックの管理）などが挙げられる。

高性能な部品はほぼ例外なく重く大きく高価になるので、用途に合った過不足のない部品の選定が小型軽量化と低コスト化の鍵となる。

2 作成の手順

2.1 立案

まずロケットを作るにあたっての大まかな計画が立てられる。打ち上げの目的（ミッション）、目標高度、使用するエンジン、打ち上げ時期・場所などが定められると、それらを達成するために必要な重量、サイズなどの要素が決定されていく。この時に電装班として注意すべき事項は

- ・ 電装のサイズ・重量の制限
- ・ パラシュートなど電装が関わる機構
- ・ 準備から打ち上げ、回収までにかかる時間

などがある。これ以外にもミッションの達成や新しい機能の実現には電装班とその他の班の連携が不可欠なので、緊密に情報共有し必要があれば分かりやすく説明する。

続いて電装班内部で上記に加えて以下のことを協議する。

- ・ 搭載するセンサの種類
- ・ 役割分担
- ・ タイムスケジュール

この他にも同時期に別のプロジェクトが進行していたら共有できる部品や作業があるかもしれないので、自分が参加しているプロジェクト以外についても把握しておくが良い。

2.2 選定

立案で大まかな概要が見えてきたら部品の選定に入る。どうしても部品種類が多くなりがちなので複数人で分担したり電源類、センサ類などで分担したりしてもよい。部品の選定に熟達するのは経験が必要なので、必ず詳しい人にその選定が妥当かどうかを確認する。説明の都合上選定を先に書いたが、特に電源系などはその他の部品の消費電流が定まってからでないと部品の選定ができないのでこの順番通りに進める必要はない。設計や評価の結果、性能が不足していることが判明したり、より小型軽量の部品で置き換えられることがわかったりし

た場合柔軟に部品を変更する。

2.2.1 データシート

全ての電子部品にはデータシートと呼ばれるその部品の特性などを記したデータシートという文書がメーカーから公開されている。部品を選定する場合には必ずこのデータシートを確認する癖をつける。英語の場合が多いので読みたくなくなるかもしれないがデータシートにはその部品の電気的性能や寸法だけでなく、製品を使った回路の例や避けるべき事項など有用な情報が載っていることもある。

最低限以下の事項については確認する。

マイコン	動作電圧、消費電流
抵抗	抵抗値、定格電流
コンデンサ	電気容量、定格電圧
MOSFET	定格電圧・電流、チャンネル、しきい値電圧
ダイオード	定格電圧・電流、順方向電圧降下
三端子レギュレータ	入力電圧、出力電圧、最大出力電流、ドロップアウト電圧
センサ類	動作電圧、消費電流、通信方式

2.2.2 モジュール類

立案で定めた種類のセンサについて実際にどのメーカーのどの製品を使うのかを決定する。たとえば 9 軸だけでも Adafruit の MPU9250 や Bosch の BNO055、ST マイクロの LSM9DS1 など様々な製品がある。それぞれ性能や通信方式、サイズや値段に違いがあるので適切なものを選定する。また人気のあるセンサのプログラムはインターネット上で提供されていることもあるので考慮する。機能が多いものは魅力的だが初期設定などプログラミングが難しくなることもあるので担当者の技量を考慮して選定する。

実際には出回っている製品で性能が足りないということは稀なので、サイズと値段、マイコンが扱える通信方式をみて選ぶことになる。後述の代表的な部品の項も参照すること。

2.2.3 マイコン

センサやアクチュエータを操作するのに十分な IO ポートや性能を持ったマイコンを選定する。開発ボードと呼ばれる周辺回路とマイコンが一体となった部品は単体で動作するのでテストしやすい、電源回路などが内蔵されているなどの利点があるが、規模の大きい電装ではスペースの問題が出てくる事がある。開発ボードではなくマイコン単体で使用する場合、最適化して全ての性能を引き出すことができるが回路の設計から始めなければならないため大変面倒になる。

2.2.4 電源

膨大な選択肢があるため最も選定が難しい部分である。まず電池としてなにが使えるのかを確認する。多くのマイコンは 5V か 3.3V、アクチュエータは 5V 程度で動作するので電圧が足りない場合、必要な電圧を確保するために直列にする。以下によく用いられる電池の電圧、特徴を記す。

乾電池	1.3V	入手性が良い。比較的重い。一次電池。
角電池	7V	入手性が良い。出力できる電流が少ない。一次電池。
リチウム電池	3V	一次電池。比較的軽い。1A 程度まで出力できる。
鉛蓄電池	12V	重いので地上側電源用。二次電池。
Li イオンバッテリー	5V	モバイルバッテリー等。二次電池。
LiPo・LiFe バッテリー	3.6~24V	高電圧高出力。軽い。二次電池。

このうち、Li イオン・LiPo・LiFe については 2020 年現在、能代、伊豆での搭載が一部を除いて（カメラの内蔵バッテリー等）禁止されているのでそれらには使用できない。重量あたりエネルギーなどから見ると Li 系が最強なので大変残念だ。代替として一次リチウム電池があるが使い捨てである。

モジュールやアクチュエータに必要な電圧、電流を確保でき、準備～回収まで動作を続けられるだけの容量を備えた電池を選定する。またフライトピンという打ち上げ時に外れる電線を使って打ち上げまでの間外部から電源を供給すれば機体に搭載する電池はより少なくて済む。モータを動作させる電源とマイコンに供給する電源と別にして信頼性を高めることや、レギュレータと DC-DC コンバータの入力電圧と効率の関係など非常に多くのことを考えて選定する必要がある。

電池から供給される電圧は多くの場合マイコンやセンサに必要な電圧と異なっているので電圧を変換する部品を選定する。多くの場合は三端子レギュレータか DC-DC コンバータを用いて電圧を下げることになるはずだ。この際、モジュールやマイコンが要求する電流に余裕を持って出力できる部品を選定する。また三端子レギュレータは下げた分の電圧を熱として放出するのであまりに損失が大きいと電装内部の熱が原因で不具合を起こす。詳しくは冷却の項を参照されたい。

2.3 ハードウェアの設計

部品の選定が完了したら設計に入る。ここで言う設計とは簡単に言えばそれぞれの部品の端子をどのように接続するかを決めることである。量が膨大になるためここには詳しくは記載しないが、回路を考えて回路図を描いていく。この作業は後述する基板作成で使うソフトでやってもいいが、修正回数が多いので最初は手書きのほうがイメージをつかみやすい。

基本的には電源系統からマイコンとモジュールへの電源供給ライン、マイコンからモジュールへの通信ラインを配置していくことになる。

どの部品をどう配線するか決まったら回路作成ソフトで回路図を作成していく。Autodesk 社の回路作成ソフト EAGLE は学生であれば無料で使えて強力な自動配線を備えている。標準のライブラリに使いたい部品がない場合は（といってもよほど基本的な抵抗やコンデンサでないかぎり入っていないのだが）自作することになる。

2.3.1 電源系統の設計

電源系統の設計は流れる電流が大きいのでミスをすると基板が燃えだすなどの事故が発生するから注意して行う。多くの場合、電池以外にも電源を供給する線があるのが普通だからその順番にも留意する。例えば電池、フライトピン、USB 端子を搭載した電装では、なるべく電池を温存するため USB 端子、フライトピン、電池の順で優先して流れる設計とする。特に開発ボードを使用する場合、USB からの電源と基板に供給する電池の電流が衝突して発煙する事故がよく発生している。ダイオードや MOSFET を用いて、ある電源から電流が供給されているときは別の電源経路が遮断される設計とする。

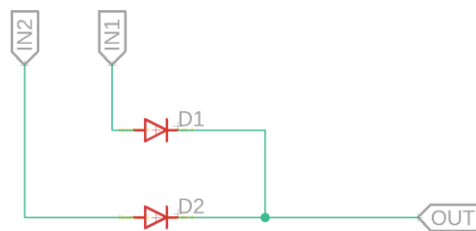


図 1

図 1 は IN1 と IN2 に同時に電圧を加えた場合の逆流を防止する回路である。OUT に出力される電圧は高い方の電圧になり電流もそちらから供給される。そのため例えば 6V の電池と USB の 5V を IN1、IN2 にそれぞれ繋いだ場合、電池の方ばかり消費されてしまう。また OUT に出力される電圧はダイオードの順方向降下電圧分（1V 程度）下がるのでその分高い電圧を供給する必要があり、流れた電流に応じて損失も大きくなってしまう。

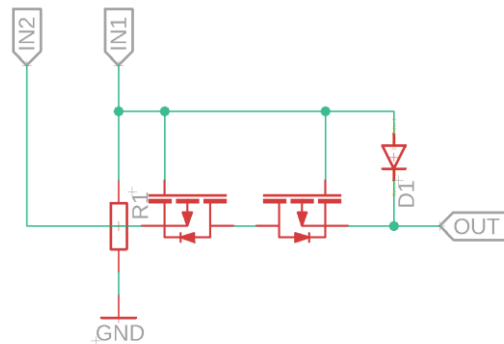


図 2

図 2 は IN1 を IN2 に優先して OUT から出力する回路である。IN1 に電圧が供給されていない時、Pch MOSFET のゲートは抵抗 R1 (10kΩ程度) によってプルダウン (GND に接続) されドレイン・ソース間を電流が流れるようになり OUT から IN2 の電圧が供給される。この状態で電流が流れたとき抵抗は FET の ON 抵抗 (0.1Ω程度) 2 つ分だけなので損失が極めて少なくなる。IN1 に電圧が供給されている時、FET のゲート電圧が高くなり FET は遮断され、ダイオード D1 を通して OUT に IN1 からの電流が流れる。ただしこのときダイオードの順方向降下電圧だけ IN1 の電圧が降下して OUT に出力される。この場合の損失は順方向降下電圧×電流である。この回路は IN1 に外部からの電源、IN2 に機体のバッテリーをつなぐことを想定しているので IN1 の損失は気にせず IN2 の損失を極力減らすようにできている。当然ながらこれらの FET やダイオードは流れる電流やかかる電圧に十分対応できる製品を選ぶ必要がある。

続いて電圧の変換をする部分を設計する。基本的に DC-DC コンバータを使うのは 1A 超えるような大電流を変換する場合や 12V から 3.3V のような目的とする電圧が大幅に離れている場合のみである。マイコンとモジュールのたかだか 200mA 程度であれば三端子レギュレータがサイズ、重量、供給される電圧の安定性などの面で勝る。

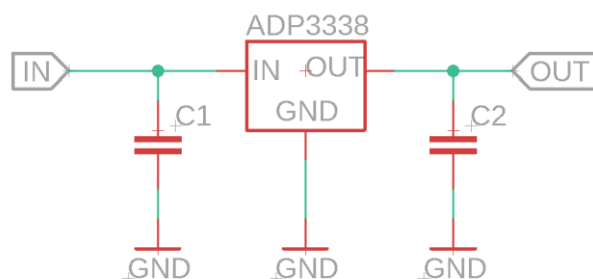


図 3

図 3 は三端子レギュレータ ADP3338 を用いて 8V~3.6V の入力電圧を 3.3V

に変換する回路である。三端子レギュレータはその名の通り 3 つの端子のうち入力端子に電源を、出力端子に出力を、GND 端子に GND を接続するだけで希望の電圧が得られる便利な部品である。C1 は入力電圧を安定させるコンデンサ、C2 は変換した電圧を安定させるコンデンサである。特に C2 のコンデンサについてはレギュレータの仕組み上、容量や応答速度が定められている場合があるのでデータシートのアプリケーション欄を確認する。ADP3338 の場合 1 μ F のコンデンサが指定されている。レギュレータには内部の損失のため目的の電圧よりある程度高い電圧を供給してやる必要がある。この電圧のことを内部で降下する電圧という意味でドロップアウト電圧という。このドロップアウト電圧が低いレギュレータを使うとより低い電源電圧を用いることができる。つまり直列する電池の数を減らすことができるかもしれないということである。ここには記載しないが近年では DC-DC コンバータでもレギュレータサイズのもので登場してきている。



図 4

図 4 はレギュレータに近い大きさを持つ超小型 DC-DC コンバータの例である。ピン配置が三端子レギュレータと共通としてあるのでコイルやコントローラについて知らなくとも容易に置き換えが可能である。DC-DC の特徴でもあるが入力範囲は 6.5~32V と幅広く、1A の出力が可能で変換効率は 90~94%に達する。レギュレータはその仕組み上電源電圧が高いほど損失が増える特性があるので、特に電池容量に気を使う用途や大電流を扱いたい場合は DC-DC コンバータの採用を検討してみてもいいかもしれない。

このようにして変換された電流はマイコンやアクチュエータに供給されるわけだが、そのままでは瞬間的に消費電流が増えた時に電圧が降下して最悪マイコンの再起動などを引き起こす。またモータのようなコイルを持つ動作機械は動くときに電氣的ノイズを発生する。これらの問題の対策としてパスコンデンサと呼ばれるコンデンサ (0.1 μ F 程度) をマイコン近くに配置したり、必要があればインダクタンスでノイズフィルタを作る。

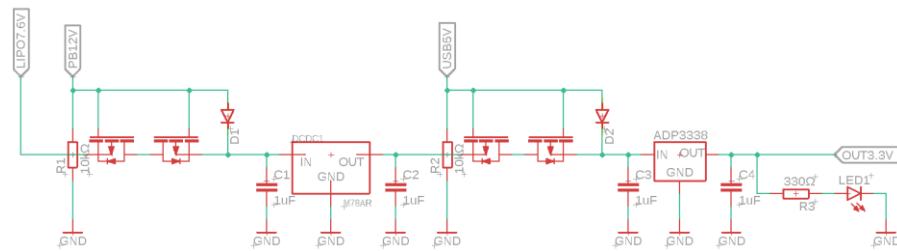


図 5

これらを組み合わせて最終的に図 5 のようになる。2020 年度能代打上のために製作した電源系統を簡略化したものである。USB5V、フライトピンの 12V (Pb バッテリー)、LiPo バッテリーの順に優先して電装に供給される。実際は前述の理由で LiPo が使用できなくなったため機体バッテリー電圧が低下しバッテリーからの電流は DC-DC コンバータを通さずレギュレータに直結される。またフライトピンを用いて発射検知をするために PB12V で FET をスイッチングし、マイコンから PB12V の接続の有無を検知する回路が追加される。加えてサーボモータを駆動するための 5V をレギュレータの直前から取り出すことになる。出力の最終段 (右端) の LED は電源の有無を確認するためのものである。

より小規模な電装ではマイコンの開発ボード上に実装されているレギュレータを用いることで小型軽量かつ容易に開発を進めることができる。実験や小規模な打上のための電装であれば Arduino の VIN 端子に角電池を接続し、基板の 5V、3.3V 端子から取り出してモジュールに供給するのが手軽である。ただし必要な電流が内蔵レギュレータの供給電流を超えないようにする。

電源系統で注意すべきこととして、レギュレータや DC-DC コンバータの供給電流が足りないからと言って並列に接続してはいけない。出力電圧が微妙に異なっているので逆流してレギュレータを破壊する可能性があるからである。同様に USB5V とレギュレータで発生した 5V を接続してはいけない。また、電池についても製造誤差で微妙に電圧が異なるので特に並列可能なものを除いて並列接続してはならない。2.9V と 3.1V のリチウム電池を並列接続した場合、内部抵抗が 0.1Ω でも 2A が流れることになる。実際に電装製作中にリチウム電池を並列接続し、強塩基性の電解液が吹き出す事例が発生している。

歴史的にみると以前は集中型の電源で使用する数種類の電圧を発生させて各部へ供給する例が多かった。例えばデスクトップ PC 用 ATX 電源ではコンセントの AC100V を 12V、5V、3.3V に変換してメモリ、冷却装置、CPU

などへ供給する。これだと配線が増えて回路が複雑になったり、一部の負荷が全体に影響しやすくなったりするため、近年では使用する部品の近くで必要な電圧に変換する例が増えてきた。これを電源の分散化（POL）という。例えば携帯端末などではカメラに必要な電力はカメラの近くで変換している。表面実装のレギュレータの小型高性能化などによって部品のモジュール化が進んだためである。現在のスマートフォンなどでは稼働時間を伸ばすためにこのレギュレータをより高効率の DC-DC コンバータで置き換えようという取り組みもある。

2.3.2 通信系統の設計

続いて通信系統の設計を行っていく。マイコンとセンサ、記録装置、コンピュータなどの間で用いられる通信方式には以下のようなものがある。

名称（呼び方）	用途
I ² C（アイツーシー、アイスクエアドシー）	9 軸、気圧センサ等
SPI（エスピーアイ）	SD カード
UART（ユーエーアールティ、ユアート）	GPS、通信装置
USB（ユーエスビー、Universal Serial Bus）	コンピュータとの通信
シリアル通信	コンピュータとの通信

一般的に呼ばれている名称で記載したが、UART は定まった規格でなく部品の名称であるとか、USB は通信方式の規格ではないとか面倒な話があるので専門家や詳しい人、細かいことを気にする人と話すときは注意する。通信方式については説明事項が多いため重要なものについて個別に解説を加える。

2.3.2.1 I²C

主にセンサで用いられている通信方式である。正式名称は I²C（アイスクエアドシー）だが簡便に I2C（アイツーシー）と表記呼称されている。通信の主である 1 つのマスターとそれに応答する複数のスレーブという用語を用いて通信方向を表す。特徴はスレーブを数珠つなぎに配線できるため特にモジュール数が多い場合に配線本数が少なく済むこと、マスターが通信タイミングを指示できるためプログラミングが容易なことなどである。逆に他の通信方式に比べて通信速度はそこまで高速ではない（読み出すデータ量は少ない場合が多いので問題にならないが）。また数珠つなぎにしたセンサのどれかが故障するとそれ以外のセンサとも通信ができなくなるといったデメリットがある。

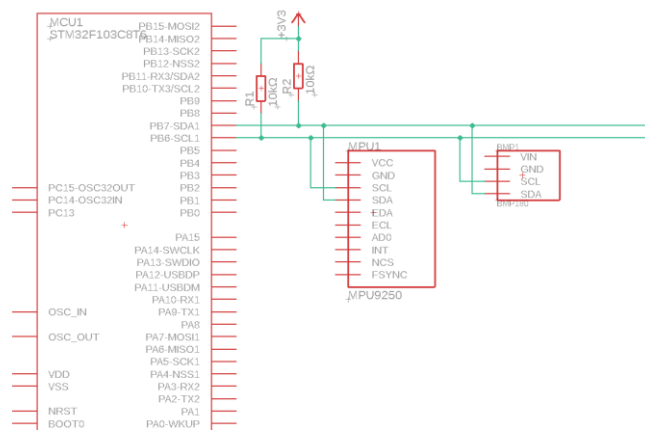


図 6

図 6 は STM32 マイコンと 9 軸 MPU9250、気圧センサ BMP180 を I2C で接続した例である。詳しい仕様は Wikipedia 等を参照していただきたいが、I2C で必要な配線は信号のタイミングを伝える SCL（クロック）とデータを伝える SDA（データ）の 2 本のみである。ただし図のように 2 本とも 1k~47kΩ 程度の抵抗でプルアップしておく必要がある。また当然だがセンサの電源端子と GND 端子に適切な電圧の電源を供給する必要がある。I2C ではモジュールごとにアドレスと呼ばれる番号が決まっている。また各モジュール内にレジスタという読み書きできる場所がある。モジュールのアドレスとどのレジスタがどういう役割を持っているかはデータシートもしくはレジスタマップに記載してあるのでそれを参考にする。

ここから先は仕組みの話なので知らなくても使うことはできるがデバッグで役立つかもしれないので読んでおくことをすすめる。I2C では初めにマイコン（以後マスター）が送信開始を伝える。すると接続されている全モジュール（以後スレーブ）が受信状態に入る。続いてマスターがアドレスを送信する（名前を呼ぶ）と、そのアドレスを持っているスレーブが ACK と呼ばれる応答を返す。ACK を受け取ったマスターはレジスタの場所と、その場所を読み取ろうとしているのか書き込もうとしているのかを送信する。読み取りの場合、スレーブはそのレジスタの値を送信する。書き込みの場合、スレーブは続けてマスターから送られてきた値を指定のレジスタに書き込む。これによってアドレスが重複しない限り多数のモジュールと 2 本の線で通信ができる。

2.3.2.2 SPI

SPI は I2C の欠点である速度の遅さを改善するためにできた規格であり I2C、SPI 両対応となっているモジュールも多い。相違点としてアドレ

スを送信してスレーブを呼び出すのではなく SS (スレーブセレクト) という端子に信号を送って呼び出すようになったのと、データの送信と受信を別の線で行うようになった。そのため合計で CLK (クロック)、MOSI (マスターアウト・スレーブイン)、MISO (マスターイン・スレーブアウト)、SS で合計 3+モジュール个数分の配線が必要になる。ちなみに速度は最大で I2C のおよそ 3000 倍程度である。電装における主な用途としては microSD カードへの書き込みに SPI が使われている。

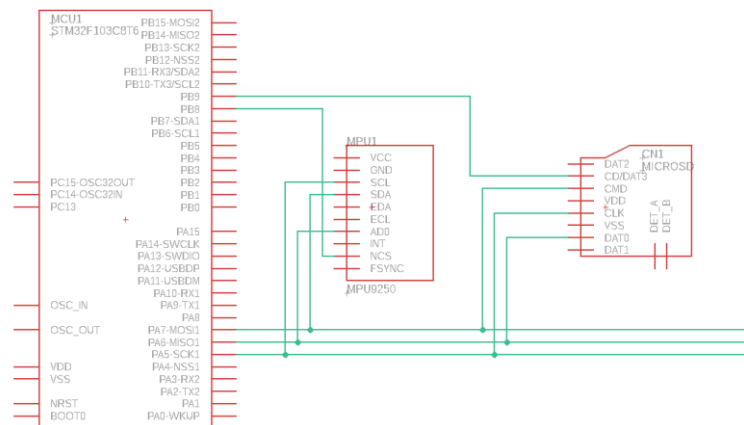


図 7

図 7 は MPU9250 と microSD カードを SPI モードで接続した例である。プルアップは必要ない。SS はモジュールによって名称が異なるので注意する必要がある。また SS はマイコン側に専用の端子がある場合もあるが、電圧の HIGH、LOW 出力ができればよいのでソフトウェアから制御する場合は空いている GPIO などを使う。

2.3.2.3 UART

非同期シリアル通信と呼ばれる方式で I2C や SPI と異なりモジュール一つに対して受信 (Rx) と送信 (Tx) の 2 本ずつを配線する必要がある。またマスターとスレーブという関係ではないので、データが突然送られてくる。そのためソフトウェアではデータを受信するたびに割り込みをかけ次のデータが送られてくる前に割り込みを完了させてまた待機、という極めてハードウェアプログラミング特有の処理をすることになる。UART は他の通信方式と違ってお互いの通信速度 (baud rate ボーレート) を合わせてやる必要がある。データシートに 9,600bps、19,200bps、115,200bps などと記載してあるのでデータが文字化けしていたら確認して正しい速度を設定する。

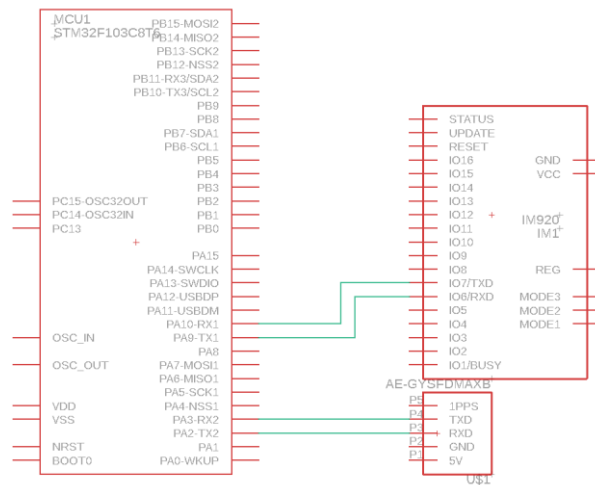


図 8

図 8 は STM32 マイコンに GPS/GNSS モジュール AE-GYSFDMAXB と 920MHz 帯通信モジュール IM920 を UART で接続した例である。マイコンの送信線はモジュールでは受信線となるので Tx は Rx に、Rx は Tx にクロスして配線する。

2.3.3 筐体の設計

筐体は少しイメージしづらいかもしれないが、基板や電池ボックス、配線を固定し外部からの衝撃や接触から保護するためのものである。基板はそのままの状態では配線が露出しているので、金属片を基板上に落としたり金属製のテーブルの上に置いたりするとショートしてしまう。また配線が固定されていないとどこかに引っ掛けた時に力がかかって断線する恐れがある。また力をかけなくてもケーブルが揺れ動くとき接合部分が劣化していく。ロケット電装の筐体はロケットの構造に安定して固定され、発射や着地の衝撃に耐える強度を持つ必要がある。また基板で荷重を支える構造にならないように注意する。筐体は基板の枚数や機体のサイズなどに制約を受けるのである程度製作の見通しがたってから設計しても良い。途中で設計を変更する必要があるかもしれないので 3D プリンタで製作するのが手軽である。

2.4 ソフトウェアの設計

2.4.1 フローチャートの設計

フローチャートとは打ち上げのどのフェーズ、どの条件でどのような動作

をするのかを定めたものである。

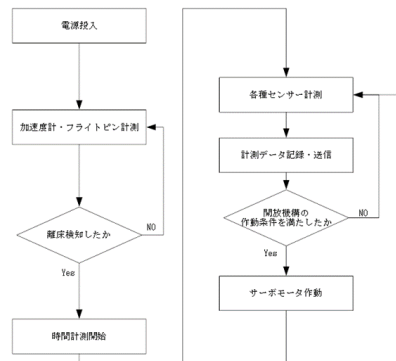


図 9

図 9 は海打ち機体のフローチャートである。条件分岐のひし形、動作の長方形などを使い分けて設計していく。処理の途中で問題が生じたときにも無限ループや未定義な動作をさせないようにする。上の図はかなりおおまかなフローチャートだが、さらに詳細に具体的な処理について記載したフローチャートを作成しても良い（多くの場合最も詳細なフローチャートはプログラムの頭の中にある）。また電装には様々な処理があるがそれらの優先順位を確定させておく。最も重要なタスクはパラシュートの開放機構の駆動である。次は機体回収のための GPS 位置送信である。逆にセンサの計測値の記録の優先度はそこまで高くない。環境によっては現在実行中のタスクより優先度の高いタスクが発生した場合に現在のタスクを中断して優先してタスクを実行する機能を持つものがある。例えば RTOS や NVIC などである。さらに何度も繰り返されるセンサのデータ取得や記録などは、その処理をどのくらいの頻度で（なんミリ秒ごとに）行うのかも決めておく。

2.4.2 プロトコルの設計

プロトコルとは通信やデータを記録するときの規則のことである。通信や記録をする際に必要以上に情報を入れようとすると時間がかかってしまう。かといって数字の羅列だけでは人間が見た時に読みづらい。また PC でファイルを開けるようにするには PC が対応したファイルシステムに従う必要がある。そのため SD カードの保存には Fat (File Allocation Table) のライブラリを用いる。拡張子としては、カンマ (,) 区切りで数値表を表す CSV 形式が簡便である。

2.5 部品調達

2.5.1 調達方法

電子部品の調達・会計は部品点数が多いなどの理由で他班と趣の異なるものになる。いちいち型番を報告し許可をもらって、という手順を踏んでいると開発に遅れがでるので部品代としてまとまった額をプールし、会計担当者が

購入した物品を取りまとめた方がスムーズに進む。また一度にまとめて大量に購入すると単価が下がるうえ送料の節約になるので、毎回使うような部品は電子班として購入して備品としたほうが良いかもしれない。

購入先はいくつかあるが一番わかり易いのは実店舗で購入することだろう。確実かつ即座に送料無しで部品が手に入るし、詳しい店員さんなら相談に乗ってもらえることもできる。しかし欲しい部品を取り扱っていない場合が多いし単価もインターネットで購入するより割高なので、すぐに部品がほしいときのみ利用する。なお購入の際はレシートと領収書を忘れずに保管する。仙台市内で電子部品が購入できる店はマルツと梅沢無線くらいである。

インターネットで部品を購入する方法としては電子部品専門サイトから購入する方法と一般のネットショッピングサイトから購入する2つがある。

専門サイトで代表的なのは秋月電子である。品揃えもよくかなり短納期で届く。単価も実店舗と比べると安い場合が多い。送料が 500 円かかるのであるべくまとめて注文する。よく使われる部品のほとんどは取り扱っているのでどの型番の部品を使うか検討する段階でも役に立つ。その他の専門サイトとしてマルツオンラインと Digi-key がある。この2サイトは提携していて UI も似ているのでまとめて紹介するが、非常に高度な絞り込み検索機能を持っている。例えば FET なら定格電流だけでなくパッケージやしきい値電圧での並び替えも可能である。またグローバル部品サプライヤーである Digi-key と提携したことで秋月より豊富な在庫を保有している。マルツオンラインから Digi-key の在庫を取り寄せて店舗受取りにすると送料無料で購入できるので便利である。

上記の店舗、専門サイトは最も間違いのない方法なので失敗が許されない短納期の開発ではそちらを利用したほうがいい。ここから紹介するのは本当に費用を安く抑えたい、かつ時間が十分にある場合に使える方法である。なぜこのような書き方をするのかというと発注から納品まで 1 ヶ月は当たり前、途中で荷物が行方不明になったり偽物が送られてきたりといった事例があるからである。中国版 Amazon ともいわれる AliExpress を利用すると電子部品の街深センから原価で部品を仕入れることができる。部品によっては単価が専門サイトの十分の一以下になることも珍しくない。一般の通販サイトなので当然絞り込みなどはないから型番を決めてから検索をかけると良い。それか「FET SOT-23 50V」などと検索しても見つけることができる。おそらく同じ部品を複数の業者が出品しているのでなるべく多く注文されている業者を選んで発注すると比較的安全である。それでも国内価格と比べて異様に安い場合はレビューを注意深くチェックしたほうがいい。実際に電子班で注文したレギュレータが通電すると即座に発煙する偽物だった事例がある。それで

も当たりの業者を引けば激安価格で部品を購入できるので覚悟できる人にはぜひ利用してもらいたい。購入する場合は 100pcs (piece : 100 個入り) などの単位でまとめて購入すると安くなる。チップ部品などは失敗などで案外使うのでかなり多めに買って問題ないと思われる。

2.5.2 代表的な部品

以下に各部品で代表的なものを記載しておくので検索の助けとして欲しい。

9 軸 :

- ・ MPU9250 :

6 軸センサの MPU6050 に磁気センサが追加されたもの。うまくすると Quaternion が出力できる内蔵処理チップ (DMP) が使えたりするが前述の構造のせいで扱いが難しい。それでも安いのでライブラリがあればお得。

- ・ LSM9DS1 :

FIFO やフィルタなど機能が多いが DMP はない。作りは素直なので動かすだけなら簡単だが使いこなそうとすると長大なデータシートと格闘することになる。とても高いがマルツで売っている。

- ・ BNO055 :

DMP 内蔵でそこまで高くない。

気圧センサ :

- ・ BMP085、BMP180、BMP280

同じメーカーの同じシリーズで同じプログラムが動く。

GPS :

- ・ AE-GYSFDMAXB、NEO-6M (-7M・-8M)

9600bps の UART で NMEA フォーマットなので精度とか細かい仕様以外はピン配置が違うだけ。

通信モジュール :

- ・ IM920

協賛でもらっているのでものをこれを使おう。データシートは日本語で読みやすい。応答時間のフォーマットがソフト屋泣かせ。

- ・ Xbee、Zigbee

かなり以前からある通信モジュール。

- ・ TWE-Lite

トワイライトと読む。低電力通信モジュールのルーキーだが使ったことないので性能はよくわからん。

コネクタ :

- ・ XH コネクタシリーズ

レギュレータ：

- ・ AMS1117 (1117 系)
低電力の LM1117 とか色々ある。3.3V0.8A、5V1.2A などバリエーション豊か。普通の性能で色んな製品に使われている。安い。
- ・ ADP3338-AKCZ
超 LDO でドロップは驚異の 190mV。3.3V 1A が供給できる高性能で国内価格は 1 個 300 円もする。Aliexpress ではほとんど偽物。

MOSFET：

- ・ 2SJ681
Pch MOSFET で一番一般的なものでマルツでも売っている。リードなので表面実装には足を曲げる手間がある。60V8A もあるのでもっと小さいもので置き換えたい。
- ・ 2SK4017
Nch MOSFET で一番一般的なもの。

開発ボード & マイコン：

- ・ Arduino
Atmega328P というマイコンを積んだボードと開発環境 ArduinoIDE のセット。Aliexpress なら最安で 200 円から手に入る。電源系が内蔵されているので実験や小規模電装には便利。通信 IO や RAM、FLASH の少ないので大規模な電装には使えない。
- ・ Mbed
積んでいる CPU やメモリの違いでいくつかのバリエーションがある。開発環境がブラウザなのが特徴。6 千円もする。
- ・ STM32
パッケージやメモリの違いで種類が多い。STM32F103C8T6 を積んだボードが 160 円、チップ単体が 100 円で出回っている。ArduinoIDE に追加パックを入れれば Arduino と同じプログラムが動く。公式には対応してないが実は Mbed も動く。チップ単体で購入して周辺回路を自作してやれば小型軽量安価。機能が多い分使いこなすにはかなり知識が必要。
- ・ ESP32
CPU が 2 つ載っていてメモリも速度も（マイコンとしては）超高性能機。開発は ArduinoIDE を使うことが多い。余計な Wi-Fi までついてくる。CPU2 つで通信と処理を同時にといった用途が考えられるがそこまでの要求が今の所ない。

表面実装（SMD）用パッケージで代表的なもの

MOSFET : SOT-23

レギュレータ : SOT-223

ダイオード : SOD-123

チップ部品 (海外サイトのインチ表記) : 2012 (=0805)、1608 (=0603)

2.6 作成

設計が完了したら実際に電装を作成していく。定められた製作期限がある場合は基板、ソフト、筐体の優先順位で作成していく。基板が完成しないとソフトウェアのテストができない、回路寸法が確定しないと筐体寸法が定まらないためである。

2.6.1 基板作成

基板にはユニバーサル基板と PCB 基板の 2 種類がある。PCB 基板は配線密度が高く小型軽量化しやすい、自分で配線をする必要が無いなどの利点があるが、基板作成ソフトの勉強が必要とか配線を途中で変更できない等の理由で初めはユニバーサル基板を利用することを勧める。

2.6.1.1 ユニバーサル基板

ユニバーサル基板とはガラスエポキシ製の板に多数の穴が開けてあって、穴の周りにランドという銅箔をつけたものである。穴と穴の間隔は 2.54mm (0.1 インチ) となっていて、表側からリード部品の端子を差し込んで裏側からハンダづけして固定し、スズメッキ線やポリウレタン線 (UEW 線) で配線していく。

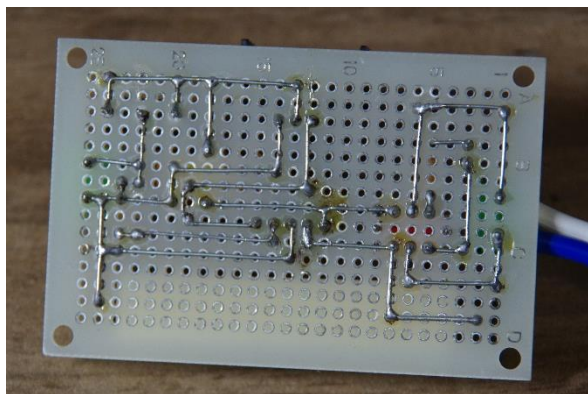


図 10

図 10 はユニバーサル基板を用いて作成した基板の裏面である。銀色の導線がスズメッキ線である。スズメッキ線は表面が被覆されていないので別の導線や金属に触れていると意図しないところに電気が流れてしまう。方眼紙などに書き込んで、どの部品をどこに配置してどう配線すれば交差しないかを考えてから配線を始める必要がある。

スズメッキ線以外の配線材料としてポリウレタン線がある。これは導線の表面をポリウレタンで薄く被覆したもので、ハンダ付けをしたところだけが被覆が溶けて導通するようになるので配線を交差できる。

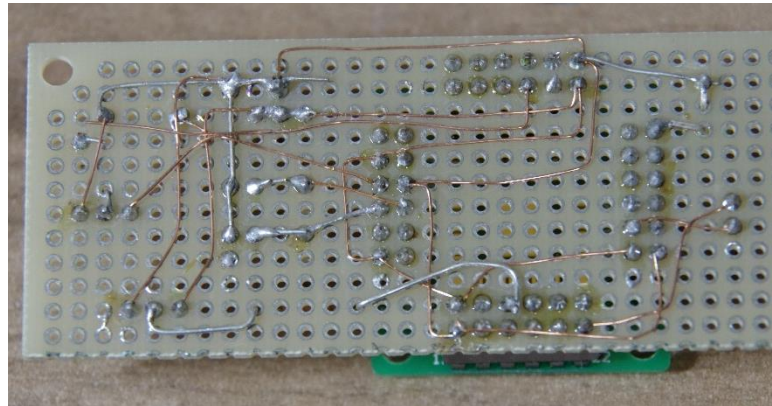


図 11

図 11 はスズメッキ線とポリウレタン線を用いて作成したユニバーサル基板の裏面である。この写真でもわかるように、ポリウレタン線はどこどこが繋がっているかが確認しづらい（目視でもテスターでも）。またハンダ付けの際の加熱が不十分だと接触不良になりやすいので、配線が混み合ってきてスズメッキ線ではどうしても配線できないときのみ用いること。

2.6.1.2 PCB 基板

PCB 基板を用いる場合には基板作成ソフトでデータを作成し、専門の業者に発注することになる。手間はかかるが一度作成してしまえば複製しやすい、ハンダ付けの際のミスが起こりにくいという利点がある。しかし最大のアドバンテージは配線密度（一定の面積にどれだけの配線が入るか）の高さと表面実装部品（SMD：Surface Mount Device）の使用による小型軽量化、コストカットである。以下では前述の EAGLE で回路図を作成したものとして一連の流れを解説する。

2.6.1.2.1 部品配置、配線

回路図から基板図を生成し、部品を適切に配置していく。使用しているライブラリが正しいものであれば実物が異なってくることは少ないが、基板図は平面なので部品の高さや幅などを想像しながら適切に配置する。配置が完了したらソフトウェアのガイドに従って配線を行っていく。大規模で複雑な基板の場合は自動配線を用いて作成する。詳しい手順に関してはここに記載するにはあまりに膨大になるため別紙と

する。

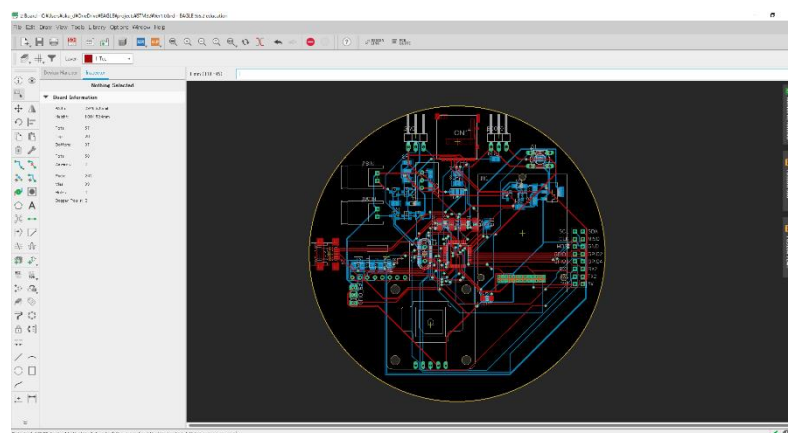


図 12：配線作業が完了した基板

赤いシンボルが表面、青いシンボルが裏面である。背の低い表面実装部品は裏面に集中して配置してある。図中では小型化のため部品の位置向きがバラバラだが、できれば同じ抵抗値の抵抗や容量のコンデンサは列にして配置するとハンダ付けの際のミスを防ぐことができる。

2.6.1.2.2 デザインルールのインポート

続いてデザインルールのインポートを行う。デザインルールとは配線の最小幅やビア（基板上の穴）の径などを指定するものである。これは業者ごとに決まっているので、注文サイトに記載してある値をもとにデザインルールを作成する。業者によっては EAGLE に対応したデザインルールファイル（.dru）を配布している場合もあるので活用する。

2.6.1.2.3 ガーバーデータ出力

指定したデザインルールにしたがってガーバーデータを出力する。ガーバーデータとは PCB 基板を作成する機械に入力するデータのことで、どこを配線するかというデータや表面に印字する文字のデータなどが入っている。EAGLE ではこの操作を CAM Processor から行う。出力したガーバーデータを個別に送信するとか zip ファイルに圧縮して送信するとか業者によって入稿方法に違いがあるので注意する。その設定をまとめたファイル(.cam)を業者が配布している場合もある。

2.6.1.2.4 発注

各業者のサイトの指示に従ってデータを入稿して発注する。PCB 基板作成業者は国内、海外を問わず様々あるが、値段の関係からほとんど中国のメーカー一択となる。国内では老舗の P 板.com やスイッチサイエンス、中国メーカーでは PCBGoGo や FusionPCB、Elecrow などが有名である。中国メーカーでは 100mm×100mm までは \$5 程度が相

場となっており、送料を入れても国内メーカーと比べるとかなり安くなる。サービス、納期の点でもかえって良かったりする（料金表にはないが両面にシルクをつけられないか聞いたところ無料に対応してくれた）のでぜひ利用してみて欲しい。ただし 100mm×100mm を超えたりオプションを指定したりすると国内メーカーと差がなくなるので、なるべくサイズ内に収める。また値段のほとんどを送料が占めるのであるべくまとめて発注する。

2.6.1.3 はんだ付け

ハンダづけの仕方などについては電子工作に関する Web サイト等を参照していただきたい。ハンダづけの技術は数をこなすしか上達の道が無いので、電装以外でも電子工作を楽しんでやってみることが必要である。

ユニバーサル基板と PCB 基板で共通する注意点として、極性のある部品は向きを間違えると危険なのでハンダ付けの前にデータシートを必ず確認する。また一見ハンダ付けされているように見えても導通していない場合があるのでハンダ付け後はテスターで導通を確認する。

2.6.2 プログラム作成

基板作成と並行してプログラムの作成を行う。ここでは具体的なソフトや言語の使用法については記載しない。プログラミング技術に関しては担当者に頑張ってもらおう他ない。

2.6.2.1 ライブラリ

ロケットの電装では様々な種類のセンサを様々なマイコンで取り扱う。新しく電装を作るたびに一から書き直していたのでは労力がかかりすぎる。そこでセンサごとに初期化、読み取り、書き込みなどの一連の処理をまとめたライブラリを作成する。同じ言語が使用できるマイコンなら種類が変わってもインターフェース（外部と関わる部分、通信など）を書き換えることで転用することができる。作成されたライブラリを使用するだけでなく、新しくライブラリを作成した場合には後述する GitHub などを用いてチームと共有する。中身を知らなくても使うことができるのがライブラリの利点だが、全くなにも知らないままでは問題が発生したときに対処できないので、最低限内容を理解しておくことは必要である。

2.6.2.2 タイムアウト

タイムアウトとは一定時間センサなどから応答がなかった場合にそれを検知したり無限ループに陥らないようにしたりする機能のことである。ハードウェアを扱うロケットの電装では様々な要因でプログラムが無限ループに陥る可能性がある。例えば「センサからの値が来るまで待つ」という処理をするとき、振動などでセンサが外れてしまっていたらそこでプログラ

ムが停止する。このような場合に備えて例えば「1 秒以内に応答が帰ってこない場合には無視して先へ進む」という動作をするのがタイムアウトである。どれくらい待つかはセンサ次第だが、パラシュート開傘などタイミングを絶対に遅らせることのできない処理が確実に実行できるようなプログラムとする。

2.6.2.3 ソースコード管理

作成したソースコードは適切な手段で管理する。例えばだれか一人の PC 上に保存されていた場合、その PC が故障、紛失した際に打ち上げができなくなってしまう。また常に最新版を書き換えて開発していると、途中で戻したくなくても戻せなくなってしまう。そこで GitHub などを用いてクラウド上でソースコードをバージョン管理する。またライブラリや EAGLE の部品などもクラウドで共有することで、次回別の人が同じ作業をする必要がなくなる。

Network graph

Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.

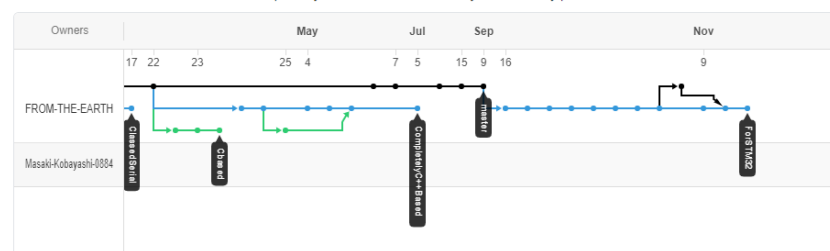


図 13：GitHub によるバージョン管理

2.6.3 筐体作成

設計をもとに筐体を作成する。実際の作業は他班に委託しても良いが、内部のパーツは基板の設計変更などに柔軟に対応するため班内で作成したほうが良い。接続部分や外部に接続されるコード類の取り回しなど、機体を作成している班との連携をとって進めていく。

2.6.3.1 CAD

3D の設計図を描くためのソフトを CAD という。基板概形が固まってきたら CAD モデルを作成し、機体のスペースにきちんと収まるか、電池などをどのように固定するかを検討する。大規模な工作の無い電子班とはいえ CAD は最低限扱えるようになっておくべきである。

2.6.3.2 水密

水密は水が侵入しないようにした容器のことである。海打ちで海面に着水する場合は水密を作成してその中に電装を格納する。これも他班と連携し

て作成していくことになるのだが、注意点として外部に出るケーブルが挙げられる。基本的に電装設計では回路に水が入ってこないものとして設計する。しかし機構駆動用のモータの電源、信号線や気圧センサ（水密内は気圧が変化しないため外部におく）、フライトピンなどどうしても外部に出さなければならない。当然着水後は電解質である海水に浸かるので、ショートを防止する回路などを備えておかなければならない。またケーブルを出す穴の部分は浸水しやすいのでケーブルグランドなどを使って適切に防水する。

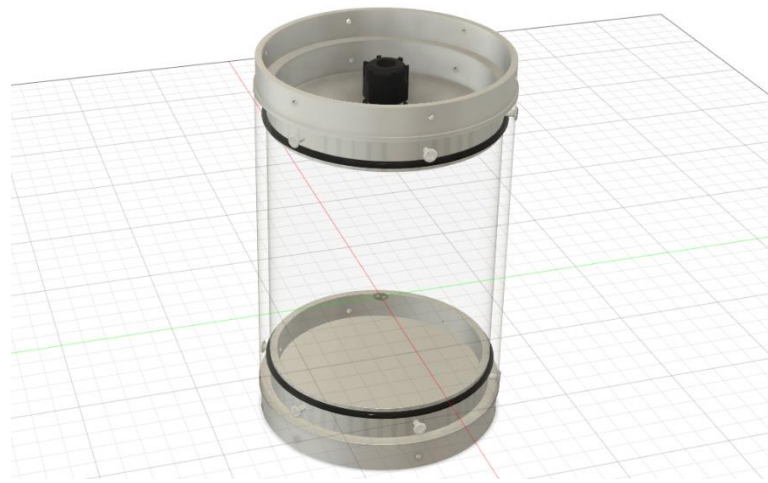


図 14：ゴムパッキンで防水された水密

2.7 評価

電装が全て出来上がってから評価を行うのではなく、回路が完成したら回路の評価、ソフトウェアが完成したらソフトウェアの評価というように細かいステップごとに行うとよい。製作と評価は歩くときの右足と左足のようなものであって、問題があるところは改善してさらに評価を行う。評価には完成した製品に対する評価と製作の過程に対する評価がある。

2.7.1 完成した製品に対する評価

実際に運用する環境に近い状態で動作させてみて、各機能が適切に動作するかを確認する。例えば電装からパラシュート进行操作するロケットであれば、正しく発射を検知して正しいタイミングで開傘するかを確認する。また 1.3 目標と優先順位で述べた観点から評価を行う。加えて可能であれば想定される環境から外れた状態での試験も行う。猛暑の日中でも排熱が十分であるか、センサが外れた、故障した場合にきちんと対処できるか等である。これらの試験を行うことで幅広い環境で運用することができるようになるだけでなく、想定される環境においてもより高い信頼性を確保できる。

これらの試験で完成した製品に問題があった場合、打ち上げまでまだ時間が

あれば適切な段階まで戻って製作手順を繰り返す。ハードウェアでは急な設計変更は困難を伴うので各ステップを慎重に実行するとともに、問題点を早急に潰していかなければならない。例えば基板作成なら基板が完成した時点で各種評価を行って製作の最終段階で問題が発生するのを避けるようにする。

2.7.2 製作の過程に対する評価

これはどちらかという下次に電装を作成するときのための作業である。製作の過程において問題はなかったかを確認する。製作の途中で失敗した、失敗しそうになったという情報を記録しておくことで次回、または別の人が作業に臨む際に事故を未然に防ぐことにつながる。また製作過程で資金、時間、人員に無駄が発生していなかったかをチェックする。

3 今後実現したいこと

3.1 点火装置の改良

現在使用されている点火装置は AC100V インバータにネオン管用 1kV 高電圧発生機を使用したものである。これを軽量でさらに信頼性の高いもので置き換える。詳しくは燃焼班の資料を参考にしたいが、イグナイタと呼ばれる端をカットした 2 軸ビニール線の端部でアークを起こす回路を作成すれば良い。GSE で使用されている鉛蓄電池を電源に使用するのが現実的であろう。昇圧回路としては DC-DC コンバータ、高周波トランス、イグニッションコイル、コッククロフト・ウォルトン回路などが考えられる。

3.2 無線 GSE

現在使用されている有線での GSE 制御を無線に置き換える。通信ケーブルの敷設、収納が不要になるので荷物が減る、展開時間の削減といったメリットがある。点火装置もそうだが燃焼班の機材は信頼性が特に重要である。無線が不調の場合に有線での使用もできるようにしておく必要がある。また作業者が間違いを起こしにくい、起こせない、間違っても事故にならない設計をする必要がある。例えばそれぞれ配線の色分け、配線を間違えると刺さらない端子、逆接防止回路などである。

3.3 冷却

電子回路に電流を流すと多かれ少なかれ熱を生じるが、小型のマイコンなどを開放空間で使用している間は問題になることは少ない。しかし酷暑の日中の打ち上げとなった 2019 年能代の電装では熱でカメラが停止する事例が生じた。筐体という密閉空間に置かれていたこと、機体外装が黒であり太陽光を吸収しやすかったこと、シーケンスが遅れたことなどが原因であると見られている。熱を抑える方法としては消費電力の小さいマイコンを使う（処理能力とトレードオフになるが）、電力伝達、電圧変換のロス減らす（三端子レギュレータでは入力と出力の差が大きいほど熱を生じる）などが挙げられる。熱を排出する方法としては機体のカプラー等をヒートシンクとして熱を逃がす、冷却水を循環させる、ペルチェ素子などが考えられる。

3.4 アクティブ姿勢制御

空中で機体の姿勢を能動的に制御する技術のことである。フィンや機体形状による安定はパッシブ姿勢制御と呼ばれる。アクティブ姿勢制御の手段としては航空機のような動翼を用いる方法とロケットのような推力を用いる方法がある。空中で能動的に姿勢を制御できるようになると、パッシブより広い範囲で安定性を獲得できるだけでなく、飛翔中に軌道を変えて効率よく高高度を目指したり、放出物の安全な射出（パラシュート、CanSat 等）、内蔵カメラでの安定した撮影をしたりすることが可能になる。しかしこの技術は失敗したときに意図しない方角への飛翔や空中で不安定化など事故が発生する可能性が高い。実験も大規模になるため実現は非常に困難であろう。

いつか私達の電装が大気圏を突破できると信じて。