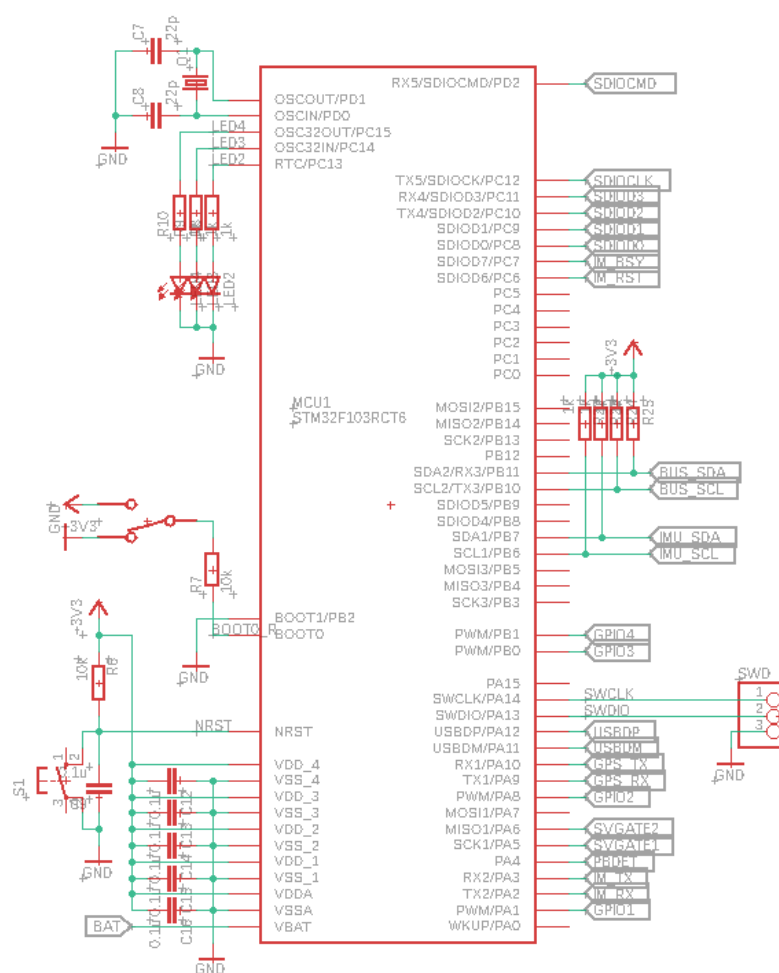


EAGLE の手引

10 期代電子班 電装担当

奥 檀



目次

1. はじめに	2
2. ライブラリの作成.....	2
使用する部品のうち、ライブラリが無いものについてライブラリを作成する。また標準で EAGLE に含まれている部品でも入手できる部品と仕様が異なる部品がないか確認する。例えば microSD カードスロットや USB ポートは各メーカから形状の異なる製品が販売されているので注意する。詳しくは別文書「EAGLE ライブラリ製作の手引」を参照して欲しい。	
.....	2
3. Schematic・Board ファイルの作成.....	2
4. 回路図の作成.....	3
4.1 電源系	3
以下に電源系の例を示す。	3
4.2 マイコンとその周辺回路.....	4

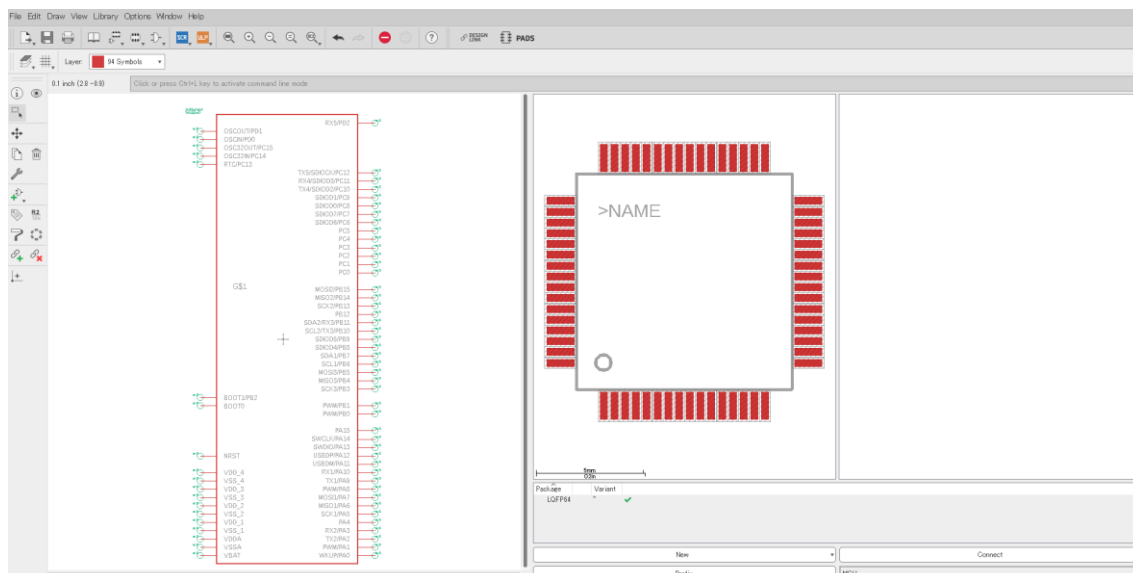
1. はじめに

この文書では EAGLE を用いた電装の製作について回路図の作成から、基板の作成、配線、ガーバーデータの出力という流れで解説する。ソフトの操作や回路のテクニックより全体の流れに重点を置いたものとなっているので、操作法などは各自で調べながら進めて欲しい。また別文書「電装製作の手引」の手順を前提としているため、使用する部品や通信方法、回路構成などは決定済みであるとする。

実例として STM32F103RCT6 マイコンを用いた電装基板の作成を行う。

2. ライブラリの作成

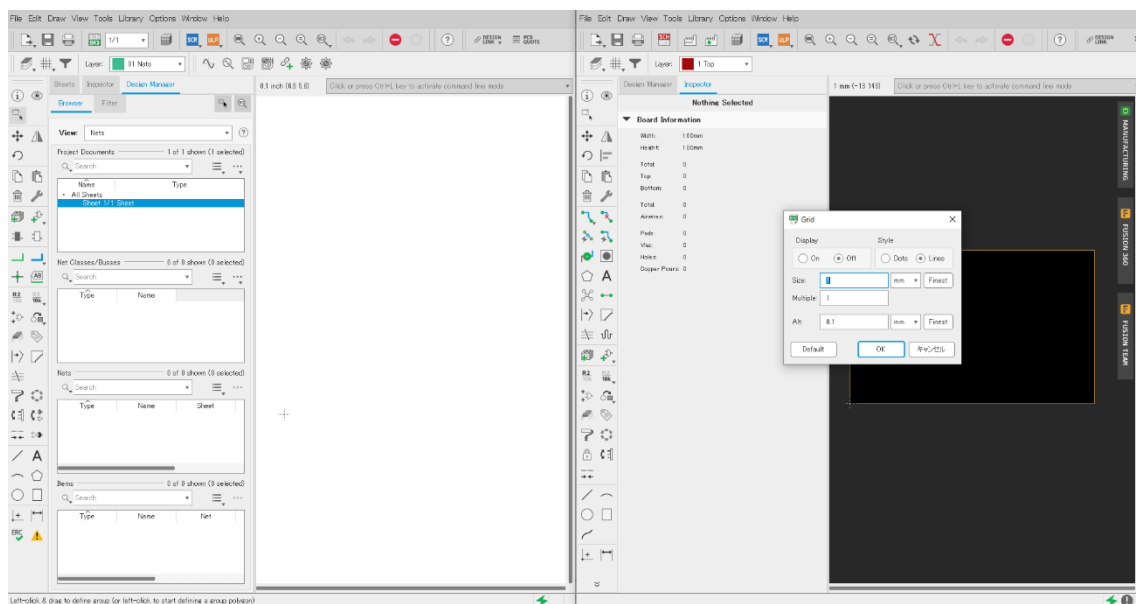
使用する部品のうち、ライブラリが無いものについてライブラリを作成する。また標準で EAGLE に含まれている部品でも入手できる部品と仕様が異なる部品がないか確認する。例えば microSD カードスロットや USB ポートは各メーカーから形状の異なる製品が販売されているので注意する。詳しくは別文書「EAGLE ライブラリ製作の手引」を参照して欲しい。



例 1：今回は STM32F103RCT6 マイコンのライブラリを作成した

3. Schematic・Board ファイルの作成

EAGLE で Schematic ファイルを新しく作成して左上の緑のボタンから対応する Board ファイルを作成する。続いて Board 画面から Grid を mil 単位から mm 単位に変更する。これは部品を配置する際の間隔で、座標や長さがそれに対応して表示される。Alt は Alt キーを押しながら動かす際の移動単位である。この設定を最初にしないと部品が 1.27mm(=0.05inch=50mil)単位で配置されてしまい手間が増える。



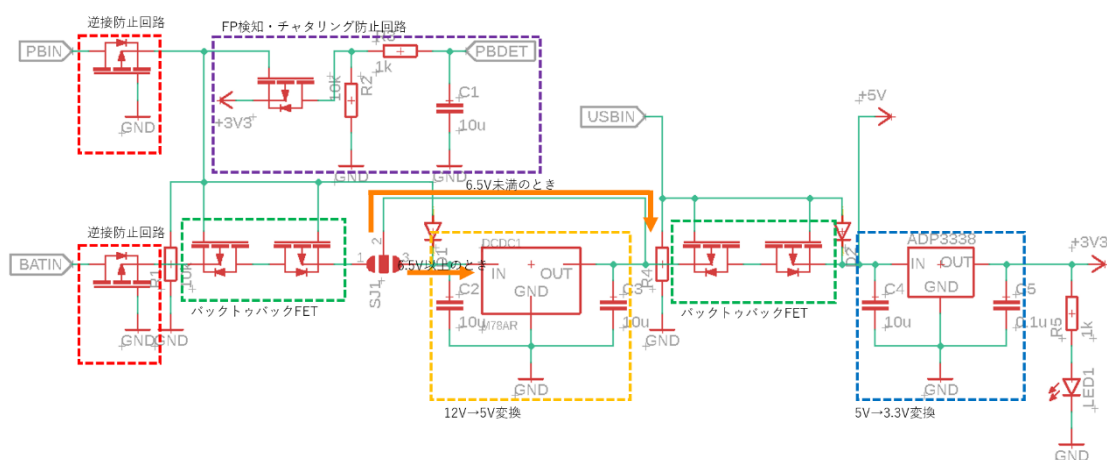
例 2 : Grid を mm 単位に変更する

4. 回路図の作成

Schematic で回路図を作成する。Schematic で部品を配置すると自動的に Board に反映されていく。片方のウィンドウを閉じたままもう一方を変更すると同期が切れて作り直しになるので両方開いたまま作業を行う。回路図上では同じシンボルでも部品形状が複数ある部品もあるので実際に使用する形状の部品と照らし合わせながら配置していく。EAGLE では同じ名前をつけた配線は Schematic 上で切れていても Board では繋がっているものとして認識される。これを利用してブロックごとに整理して描くとよい。配線 (Net) の名前がデフォルトのままだと Board を見たときに分かりづらいので全ての Net には名前をつけておく。

4.1 電源系

以下に電源系の例を示す。



例 3 : 3 種類の電源を 2 つの電圧に変換する回路

入力フライトピン (FP) の 12V、内蔵電池の 6V、USB の 5V の 3 種類。出力はアクチュエータの 5V、マイコンの 3.3V の 2 種類である。チャタリング防止回路と表記した部分は CR 回路でローパスフィルターになっており、ピンが擦れたときなどの瞬断を誤検知しないようにするためのものである。途中にジャンパが存在するのは内蔵バッテリーが 6.5V より高いときは DC-DC コンバータへ、低いときは迂回して直接レギュレータへの選択ができるようにである。これは使うバッテリーの種類によってパターンをブリッジさせて選択する。

Type	Name	Sheet	Row	Column
Net	+3V3	1	0	0
Net	+5V	1	0	0
Net	BAT_ARV2FET	1	0	0
Net	BAT_FET2FET	1	0	0
Net	BAT_FET2JMP	1	0	0
Net	BATIN	1	0	0
Net	DCDCIN	1	0	0
Net	GND	1	0	0
Net	PB_ARV2GATES	1	0	0
Net	PB_FET2R	1	0	0
Net	PBDET	1	0	0
Net	PBIN	1	0	0
Net	REG_DCDC2FET	1	0	0
Net	REG_FET2FET	1	0	0
Net	REG_R2LED	1	0	0
Net	USBIN	1	0	0

例 4：例 3 の Net リスト

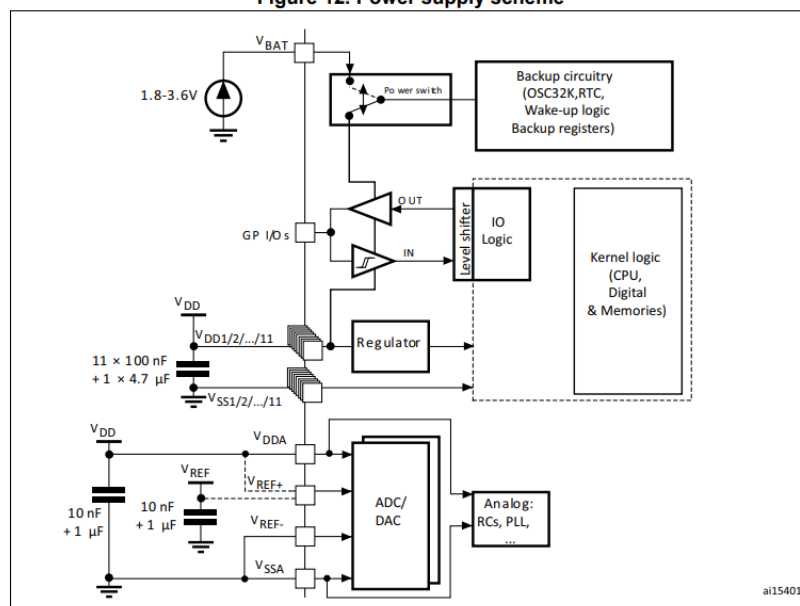
Net につける名前の例を示す。PB はフライトピンの鉛蓄電池、ARV は Anti ReVerse (逆接防止)、DET は Detect (検知)、JMP は Jumper、R は Resistor (抵抗) の意味である。このへんは設計者が覚えられる範囲で略して良い。また 2 は to の略でソフトウェア開発ではしばしば用いられる略語である (似たものに for → 4 がある)。REG_R2LED なら 3 端子レギュレータ周辺にある抵抗から LED へ向かう配線を表す。

4.2 マイコンとその周辺回路

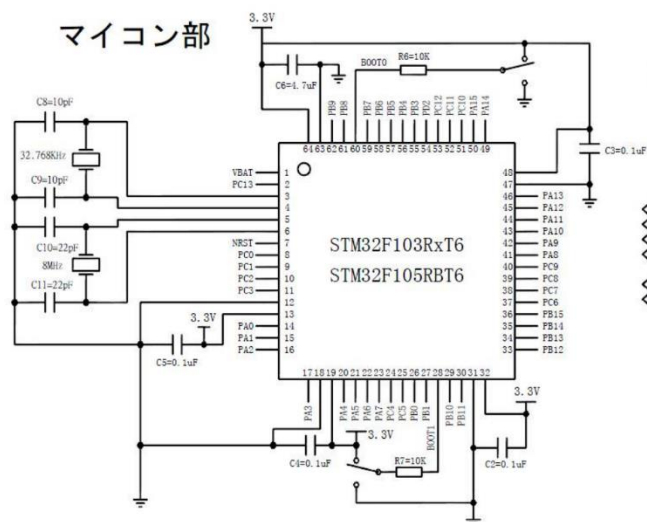
続いてマイコンとその周辺回路を配置する。周辺回路とはマイコンを動作させるために必要な水晶発振回路やリセット回路などである。本来であればデータシートの PowerSupply の項目を参考にして設計すべきだが、面倒だしパーツの入手などで非効率になるかもしれないので開発ボードなどの回路図を参考にすると良い。たとえば ATmega328P なら Arduino、STM32 なら Nucleo などである。ただしどこまでが動作させるのに必要な部品なのかを見極める必要はある。

Power supply scheme

Figure 12. Power supply scheme

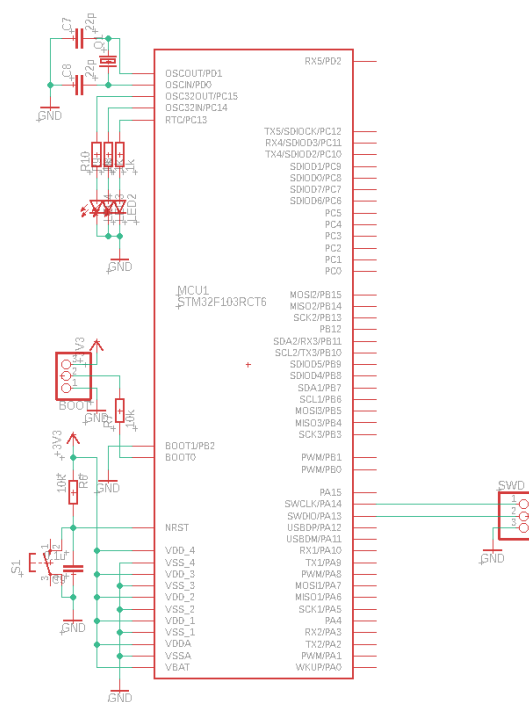


例 5：STM32 のデータシートの電源系統図の一部



例 6：秋月の開発ボードのマイコン周辺回路

たとえばSTM32で必要となるのは、電源を安定させるためのコンデンサ、書き込みモードを切り替える端子、書き込みの端子、リセット回路、水晶発振回路である。

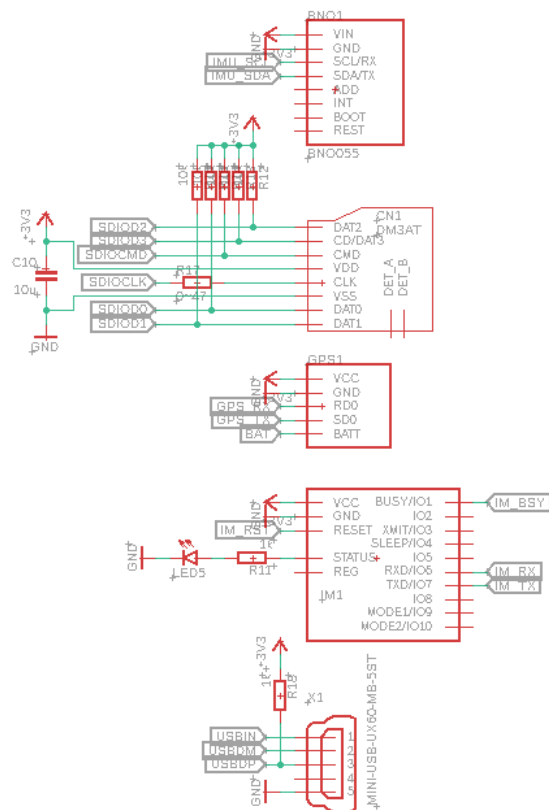


例 7：マイコンと周辺回路

左上から水晶発振回路、LED、ブートモード選択、リセット回路、電源、右側は書き込み端子

4.3 基板上モジュールの配置

基板上に実装する部品を配置していく。ここでも各部品のデータシートや実装例を参考に周辺回路を構成する。USB をどちらに入れるか迷ったが打ち上げ時は接続しないので内部扱いとした。

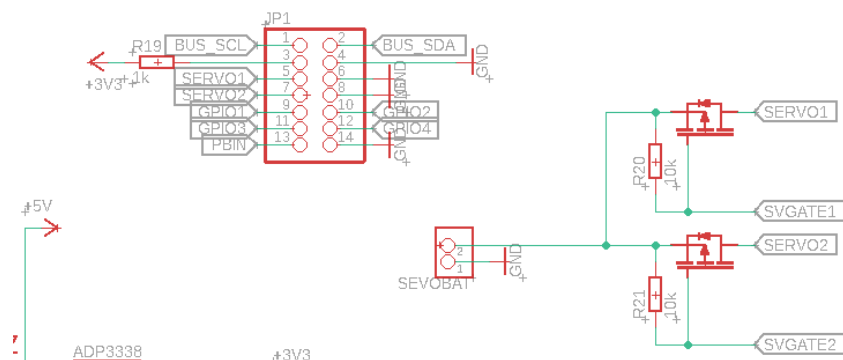


例 8：基板上のモジュールと周辺回路、配線名称

例えば USB 端子の DP は子機側を指定の抵抗値でプルアップする。また SD カードのように数 MHz 以上の信号が走る回路ではインピーダンスによるノイズが発生しやすいので、調整用の抵抗や場合によってはインダクタンスを取り付けられるようにしておく。といってもノイズに関しては配線パターン等にも起因し、発生するかもしれないししないかもしれない、極稀に発生する、といった不安定なもので問題の解決にはアナログ回路の知識とかなりの経験を要する（少なくとも私は無理）。問題が発生したら通信速度を落とすか、基板を別のパターンで作り直したほうがいいのかもわからない。

4.4 外部端子の配置

外部と接続する部品を配置していく。ここで注意するのはどういった端子を使うのかということである。ピンヘッダやピンソケット、XH コネクタ、ターミナル、DC-DC ジャックなどからサイズや流れる電流の大きさ、抜けにくさなどを考慮して選ぶ。また基板を電装に組み込んだときにどうなるかも考えて端子を選ぶ。例えば上方向に余裕がない部分には垂直方向の端子ではなく水平方向の端子を選ぶ。外部に出す端子は場所を取るし防水も難しくなるので基本的には必要最小限にする。

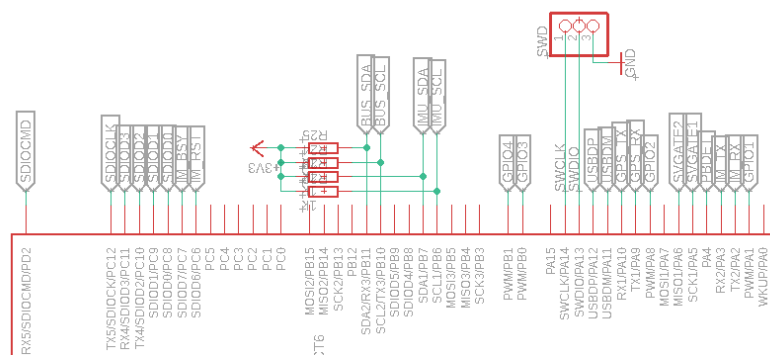


例 9：外部端子とショート対策回路

外部に出す端子は（特に海打ち機体で）ショートしたときの対策をしておく必要がある。またモータなどのアクチュエータに接続する端子はサージやノイズが乗るので適切に対策する。ここでは SVGATE1,2 をオープンドレインにしてハイサイドスイッチを駆動することで駆動時のみ電源と外部端子が接続されるようにした。またノイズ対策として電源を分離した。このあたりに関しては実験不足であり今回の電装の試験結果によってはフォトカプラによる GND 分離などより強力な対策を要するかもしれない。外部 3.3V 端子は気圧センサ用であり直列に 1kΩが入っているのは浸水時に電流を制限するためである。センサ類は消費電流が少ないので、BMP180 等についてはこの方法で正常に動作することを確認している。

4.5 マイコンと部品を接続

4.3 と 4.4 で配置した部品とマイコンを接続していく。マイコンのピンは複数の機能を持っているピンがあるので、数が少ない機能から繋いでいく。例えばあるピンが I2C の SDA と USB の DP の機能を持っている、マイコンが I2C を 3 チャンネル、USB を 1 チャンネル持っている場合、当然 USB はそこにしか繋げないので USB を優先する。



例 10：接続した端子

名前をつける機能を使わずに回路図を描くと煩雑になって理解しづらくなる。

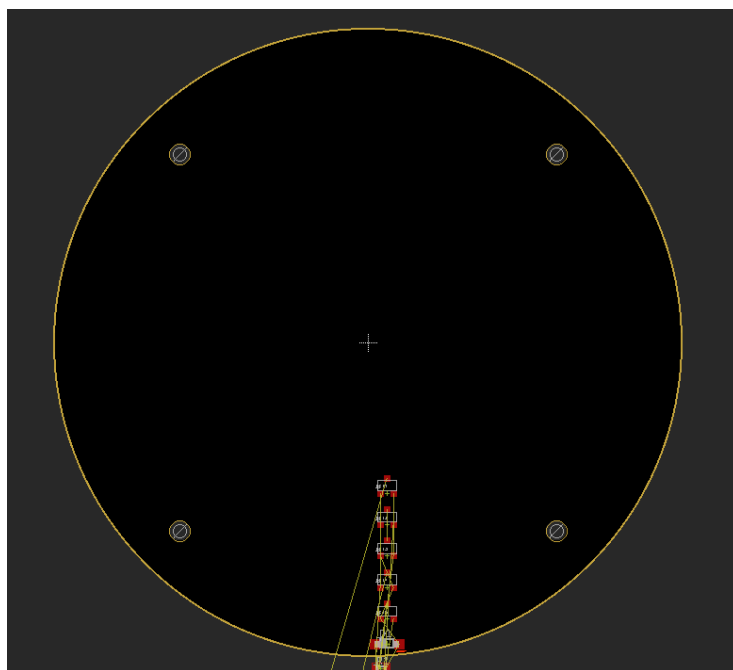
例のように名前をラベル表示すると良い。注意点として仮想的繋がった配線でも、名称を変更する際 Change Name Of で全てのセグメントを選択しないと一箇所だけ名称が変更されて接続が切れてしまう。

5. 基板の作成

回路図ができたら基板の作成に移る。Schematic を閉じずに画面を Board にすると回路図に配置した部品が置かれていて黄色い線で繋がっているのを確認する。

5.1 基板外形の設定

まず基板の外形を設定する。Layer を Dimension に切り替えて Delete を選択し、もとからある線をクリックして削除する。続いて Line、Circle などから描くのに使うツールを選択して基板外形を描く。左上にマウスの座標が表示されるので参考にする。このときにネジ穴なども開けておく。大きい穴なら Circle で描けるが、小さい穴は Hole で指定した方がいいかもしれない。

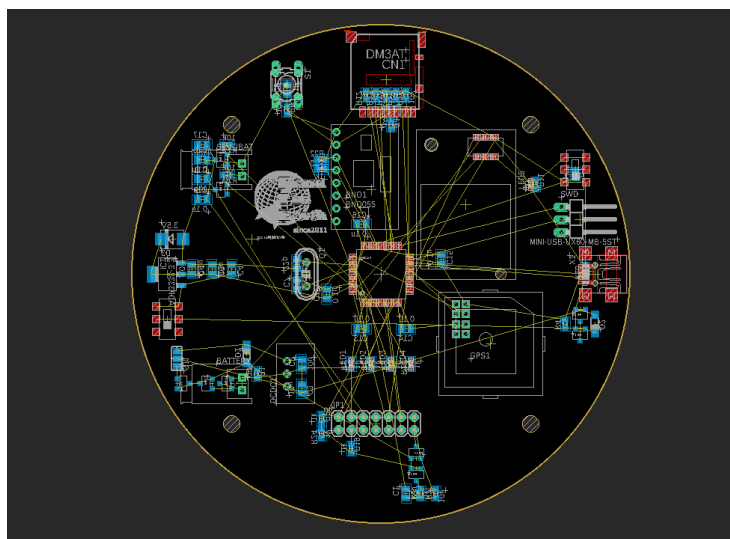


例 11：円形を指定 四隅に Hole でネジ穴

5.2 部品配置

基板に収まるように部品を配置していく。両面基板の場合裏面に配置する部品は Mirror で裏返して配置する。当たり前だが裏面においた部品は表面の部品とは干渉しない。microSD カードソケットや USB コネクタなどはきちんと差し込める位置に配置する。microSD カードソケットはカードの分の寸法を忘れがちなので、基板を入れる容器と干渉しないよう注意する。基板が複数層ある場合、ボタンなどは作業者が操作しやすいように考えて配置する。不安な場合は基板と部品の CAD を作成して確認しても良い。またあまり部品を近づけすぎると配線するスペースがなくなってしまう。回路が複雑になると余裕がなくなっていくので配置を工夫しないと配

線できなくなっていくが、パズルだと思って試行錯誤してほしい。

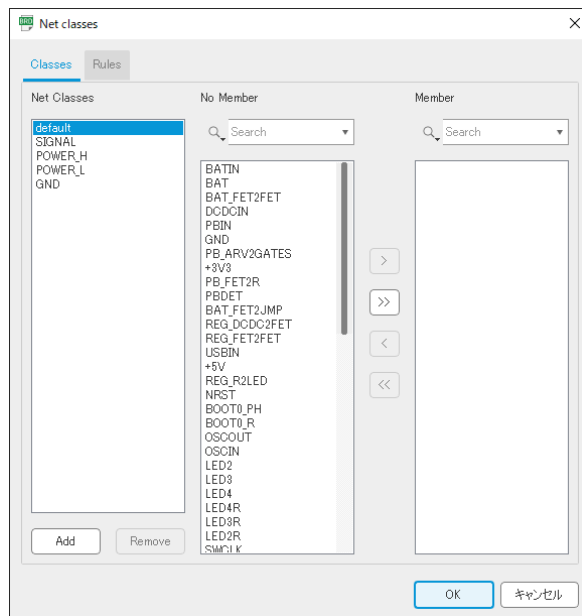


例 12：部品を配置（赤が表、青が裏）

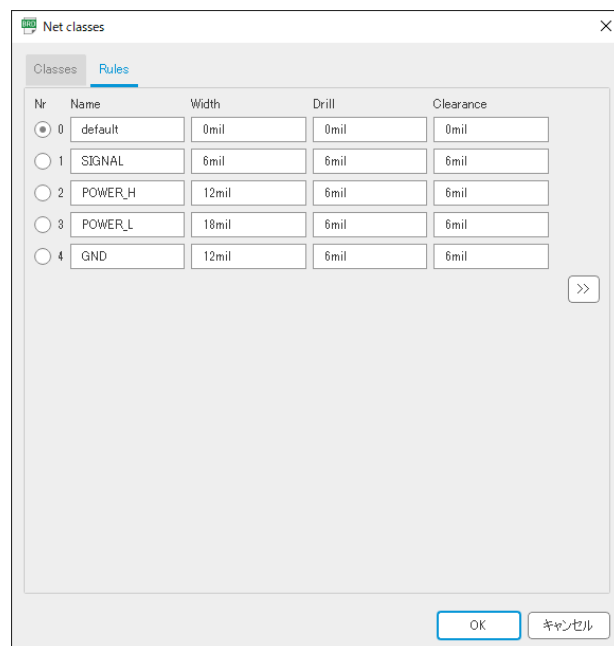
5.3 Net Class の設定

続いて Net Class の設定を行う。導線は太さによって流せる電流量が決まっているので大電流が流れる導線は太く、小さな信号しか流れない導線は細くしたい。そこで配線を数種類に分けてそれぞれの太さを設定する。さらに基板の裏面と表面を繋ぐ縦穴（ビア：Via）の大きさは電流量だけでなく寄生容量などにも影響する。また電流を流すと周囲に磁界が発生して周囲の導線に誘導電圧が生ずる。そのため大電流が流れる導線からは間隔（クリアランス：Clearance）を長く取る。

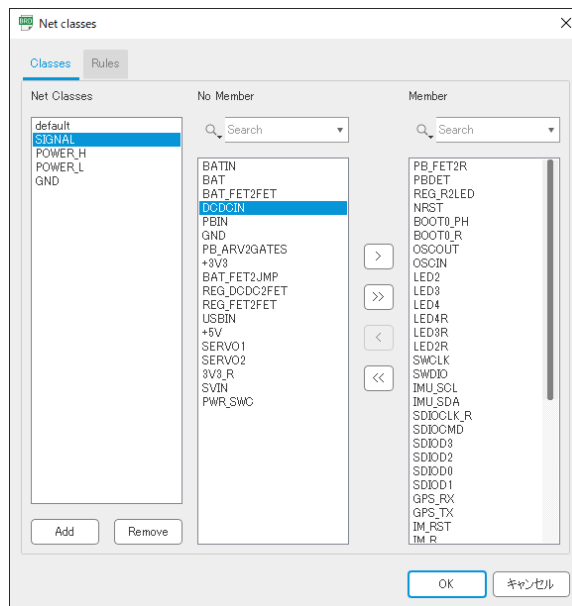
PCB 基板の場合、1A で 1mm が幅の目安と言われている。信号が流れる配線は PCB を発注する業者の最細幅（多くは 6mil）として良い。電装で最も大きい電流が流れるのはアクチュエータ用の回路だが、500mA 程度で 1 秒以内が多いので数十 mil とする。ビア径とクリアランスは今の所 6mil で問題はない。



例 13：信号線 SIGNAL、大小電源線 POWER_H,L、GND を追加



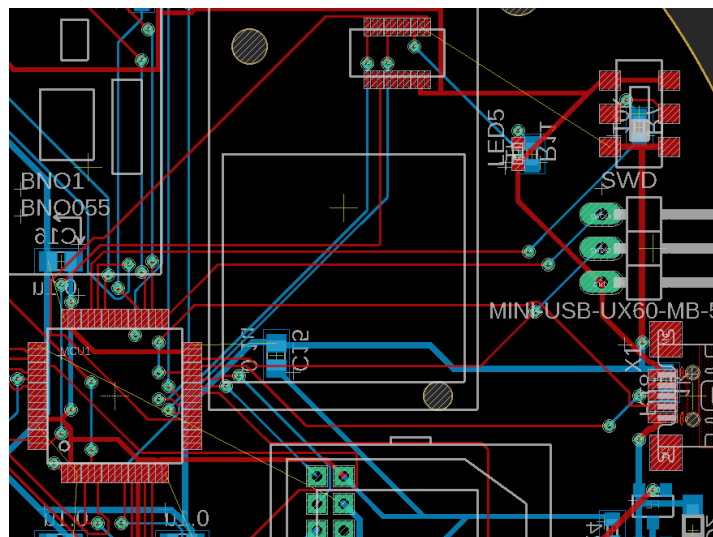
例 14：幅、ドリル（ビア）径、クリアランスを設定



例 15：各配線を Net Class に分類する

5.4 自動配線

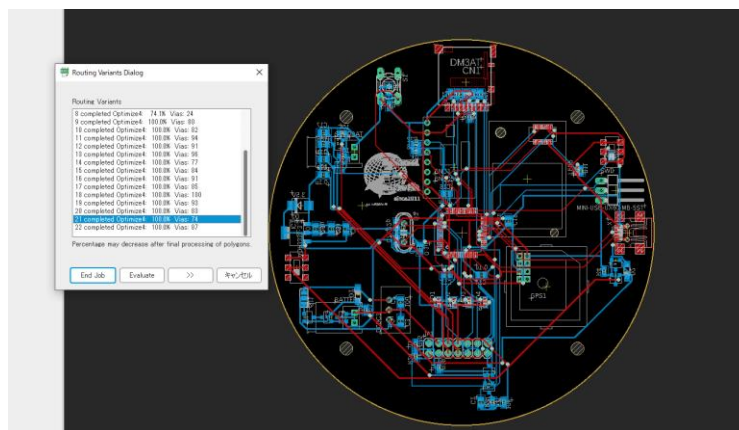
EAGLE の目玉機能、自動配線を行う。Autorouter を選択して Effort を High に切り替える。Continue、Start を押すと自動配線が始まる。パーセンテージは進行状況で全ての配線がつながると 100%になる。Vias の数字はビアの数であり配線の長さとおおよそ相関するので、あまりに多いときは部品配置を見直したほうがいいかもしれない。密度の高い基板では 1 度で 100%に届かないことが多い。どこかにたどり着けていない配線があるので、Evaluate を押して黄色い線のままのところに無理な配置がないか見直す



例 16：配線に失敗して黄色いままの線

この例だと左下の BAT 線と右上の 3.3V 線が失敗している。3.3V 線は LED5 の位

置を下に下げて対処してみる。BAT 線は POWER_H の幅を 15mil まで下げてみる。再度自動配線を実行するには Ripup から All Signals を選択して全ての配線を消去する。Autorouter を選択すると前回の結果を利用する設定になっているので Continue existing job のチェックを外して実行する。

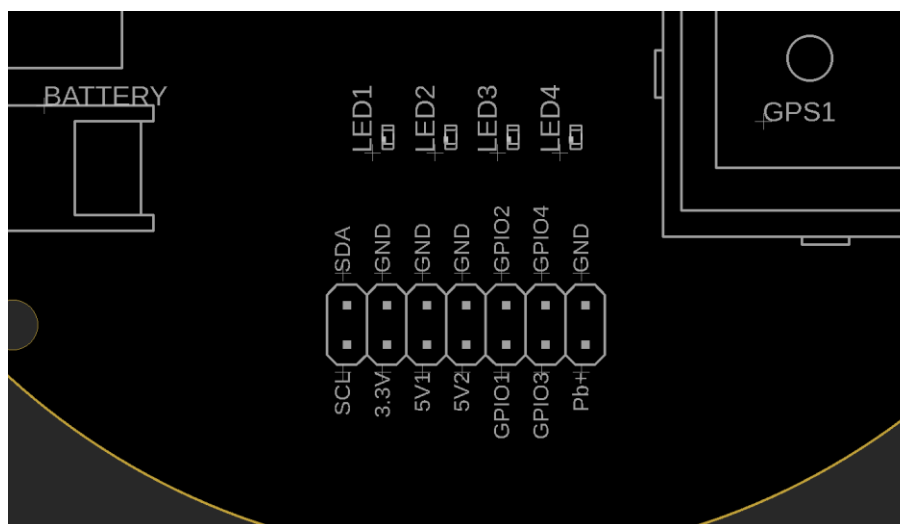


例 17：変更したら 100% になった

配線が長くなると自己インダクタンスが増えてノイズが強くなるので、ビアが少なめの結果から選ぶ。番号ごとに水平の配線が多い、斜めの配線が多いなど特性があるので、いい結果が出るまで繰り返しても良い。

5.5 シルクの書き込み

シルクとは基板表面に印字される文字や線のことである。スイッチ付近にどちらが ON なのか、外部端子のピン配置などを書き込む。Layer を tDocu に切り替えて Text を選択して書き込む。裏面は bDocu レイヤーに書き込む。



例 18：ピン配置を書き込んだ（レイヤー tDocu、tPlace、tNames のみ表示）

5.6 デザインルールのチェック

出来上がった配線が PCB 基板業者の規格に従っているかをチェックする。この規格はデザインルールと呼ばれ.dru という形式で業者が配布している場合もある。

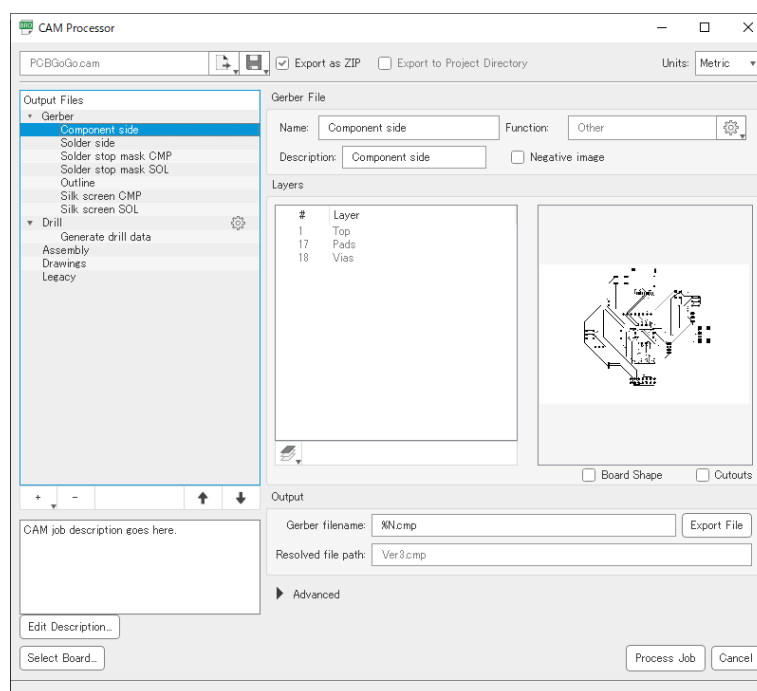
DRC から Load でデザインルールを読み込んで Check を押す。Errors が出たら対処する。

5.7 最終チェック

データ出力前の最終チェックを行う。正しく配線が繋がっているか、間違っている回路はないかなど基板を拡大して確認する。ここまで2次元で設計を進めてきたので、部品を実装したときにケーブルは想定通り通せるか、電装に組み込んだとき干渉する部分がないかなど三次元的にイメージしてチェックする。

5.8 ガーバーデータ出力

CAMProcessor でガーバーデータを出力する。ガーバーデータの形式は業者によって異なるので配布されている.cam ファイルなどを利用する。まず CAMProcessor を開いて左上のメニューから Open CAM File を選択して.cam ファイルを読み込む。細かい設定は.cam ファイルによって異なるが、だいたい表側の配線、裏側の配線、表側のランド、裏側のランド、外形、表側のシルク、裏側のシルク、穴あけ位置の8項目の場合が多い。確認したら Process Job でデータを出力する。



例 19 : PCBGogo の CAM ファイルを読み込んだ場合

6. 発注

PCB 基板業者の WEB サイトの指示に従って発注を行う。多くのメーカーでは 100×100mm までは 500 円程度に設定してある場合が多く、その範囲に収めるとかなり安く作ることができる。逆にオーバーしたりオプションをつけたたりすると値段がどんどん上がるので注意して欲しい。