**3 制作**

グラフィックデザインをベースに基板を制作する際、通常の基板制作とは違った点に配慮をしなければならない。

**3.1 新手法において考慮すべき点**

**3.1.1 電子パーツの実装方法**

電子パーツを基板に実装する際、スルーホール実装と表面実装の二つの実装方法を選ぶことができる。スルーホール実装の場合、穴にパーツを差し込み、それを裏側で固定するのが通常の実装方法である。差し込んだ方向と逆側の面にハンダを流し込み固定するため、銅板は逆側の面に存在しなければならない。

この二つの実装方法を踏まえて、グラフィックデザインを用いた基板制作で最終的な外見を考慮する際、削った銅板の面を見せるか見せないかを必ず考慮する必要がある。両面銅板でない場合、銅板の裏側はおもて面のブロンズの光沢のある銅に対して、特に加工のされていない板であるため、削った回路をあえて見せたいのか、銅板の裏側が表になる形でも大丈夫なのか、最終的なビジュアルの予想図によって判断しなければならない。

**3.1.2 電子パーツの形状・サイズ**

今回の基板制作手法では機能よりも外見が重視されるため、通常の基板制作に対して、電子パーツの形状が大きく作品に関わってくる。

スルーホール実装で使われる電子部品はDIP(Dual-in-line package)と表記され、表面実装で使われる電子部品はSMD(Surface Mouting Device)と表記される。SMDはDIPと比べサイズが小さい。

表面実装においては、穴は存在せず、パーツは全て削った面の上にそのまま半田で固定をする。また、表面実装とスルーホール実装のパーツは機能的には同じだが、表面実装のパーツのほうが格段に小さい場合が多い。そのため、スルーホール実装よりはるかに細かい操作を要求される。

**3.2 制作手順**

**⑴ モデルにする図面の選択**

グラフィックデザインベースの基板を制作するために、まずはモデルを選ばなければならない。適切なモデルの条件としては、

・あまり連続的ではないこと

・小さいパーツを並べることで構成できること

・実際の機能部分と意味づけのしやすいパーツが連想できること

などをあげることができる。

**⑵ 機能の選択**

モデルが決まったあとは、基板に搭載したい機能を決める。「この部分を光らせたい」「この部分をスイッチにしたい」など。

今回制作する基板はArduinoで書き込み可能な基板を対象にしているため、AVRマイコンを使用する。

**⑶ 回路図設計**

フリーの電子回路図設計ソフト「Eagle」を使って、回路図を設計する。AVRマイコンを使った回路図を作成することが難しい場合は、オープンソースのデータとして、「Fabkit」などのEagleデータをそのまま利用する。回路設計は、最終的に自分の目的としている外見に応じて、片側基板で行うのか、両面基板で行うのか、線の伸び方はどうするか、などを考慮しながら行う。

**⑷ ミリングマシンで基板を削る**

Eagleデータを画像に変換する。もしつながっている場所や、細部で変更したい箇所などあれば、イラストツールでその箇所を修正する。その画像をMITがウェブ上に公開している「Fab Modules」というサイトで、ミリングマシンに合わせたGコードに変換する。トレース用のデータ、カット用のデータ、穴を開けるのデータの３種類に分け、基板を削る。

**⑸ パーツを半田付け**

パーツを基板に半田付けする。表面実装の場合、銅板のめっきが剥げてしまう問題が多く生じてしまうため、半田付けは慎重に行う。電気がうまく通っているかをテスターで確認し、

**⑹ ブートローダー、プログラムの書き込み**

**3.3 Geikoduinoの制作**

まず、グラフィックデザインをベースに実験的に制作した作品が「Geikoduino」だ。モデルにするグラフィックデザインは九州大学大橋キャンパスを上空から俯瞰した図である。

このモデルを選んだ理由としては、建造物の形が直線的でシンプルであるため、電子パーツを想起しやすいこと、噴水・木・円形の広場などの特徴的な設備や空間があるためモデルが何であるか知っているものにとってはわかりやすく発見があること、実際の建造物であるため３次元的にも考えやすいことなどが挙げられる。一作目として、芸術工学部のキャンパスをした基板にArduinoのような機能を搭載したことから、「Geikoduino」と名付けた。

機能的にはArduinoのようにパソコンからプログラムが書き込むことができ、ピンソケットから自由に繋げることのできる基板を作成したかったため、MITの公開しているPCBミリングで作成可能な簡易的なArduinoのシステムを持った基板「Fabkit」のデータを使用した。Fabkitはeagleデータパーツの詳細等がインターネットでオープンになっている。今回はFabkitの回路図を使用し、最終的な視覚的イメージにあったパーツを変更した。片側銅板を２枚使い、表側と裏側を両方とも使いながら配線をし、表面実装のパーツとスルーホールのパーツを混在させた。配線も表側の面では、実際に通ることのできるルートにのみ配線を行うことで、視覚的・コンセプト的にキャンパスを表す基板を目指した。

まず、一つ目のプロトタイプを制作した。

Johan von Knowがオープンソースとして公開している、簡易的なPCBミリングで作成可能なarduinoと同様の仕組みをもった「Nanino」のデータを利用し、そのパーツの配置を変え、理想の形の基板へ持って行こうと試みた。しかし、結果としては失敗に終わった。失敗した理由としては、Naninoがスルーホール実装を前提としたパーツを使っていたため、回路図のグラフィックにデータを合わせて基板を削ったところ、基板の向きが逆になってしまうという問題があり、求めていた基板デザインが左右反転してしまった。また、スルーホール型のパーツを使うと

銅板が剥がれてしまうといった問題がよく発生した。この問題は銅板にパーツを取り付けようとした際、小さい面積に固定しようとした時に多く見られた。よって、回路設計の段階から、パーツが固定される場所の面積に余裕をもって設計した方がいいことがわかった。

手間をかけて作成したにもかかわらず基板がエラーになってしまうことは、作成者のモチベーションにつながってしまう。機能が多く、配線も複雑になる場合は、この手法は適切ではないことがわかった。