

Arduino を用いた 2 オクターブ電子ピアノの製作

上西 綾華 三木 楓華

1. 研究概要

私たちは、Arduino の制御を通じて電子回路の知識を深めるとともに、3D プリンタやレーザー加工機を活用して材料加工の技術を習得することを目的に、押すと光る 2 オクターブの電子ピアノの製作に取り組んだ。

2. 研究の具体的内容

今回の研究で使用した部品とソフトウェアを下に示す。

使用部品

・ Arduino Mega	1 個
・ 抵抗 (10k Ω)	52 個
・ タクトスイッチ	26 個
・ LED	26 個
・ トランジスタ	26 個
・ 圧電ブザー PT08-Z185R	1 個
・ ブレッドボード	9 枚
・ アルミ棒	50cm

使用ソフトウェア

- ・ Arduino IDE
- ・ Tinkercad
- ・ Blender
- ・ Fusion360

(1) 音の生成

音の生成には Arduino IDE を使用した。
2 から 26 ピンを入力ポートとして設定するとともに内部プルアップに設定した (図 1)。

プルアップに設定することで、プルアップ用の抵抗を用意する必要がなくなり、部品数の削減と回路の簡略化ができる。

```
void setup() {  
  
    for(int i=2; i<=26; i=i+1) {  
  
        pinMode(i, INPUT); //2~26 ピンを入力ピン  
        に設定  
  
        digitalWrite(i, HIGH); //3~10 ピンを内部  
        プルアップ  
  
    }  
}  
  
void loop() {  
  
    {  
        int t=0;  
        if digitalRead(2)==LOW{  
            t=262          //ド  
        }  
        if digitalRead(3)==LOW{  
            t=277          //ド#  
        }  
        if digitalRead(4)==LOW{  
            t=294          //レ  
        }  
        if digitalRead(5)==LOW{  
            t=311          //レ#  
        }  
        if digitalRead(6)==LOW{  
            t=330          //ミ  
        }  
    }  
}
```

図 1 音の生成 (抜粋)

if digitalRead 関数で、入力があったピンを検出し、それに応じた周波数の方形波を指定したピンに生成する (図 1)。

ボタンが押されていないときは noTone 関数を使用してパルス信号を停止させ、音が出ないようにした(図 2)。

音階と周波数の値は以下の通りである(図 3)。

```
if(t!=0){
  Tone(26,t);
}else{
  noTone(26);
}
}
```

図 2 noTone 関数

オクターブ	4 (Hz)	5 (Hz)
ド	262	523
ド#	277	554
レ	294	587
レ#	311	622
ミ	330	659
ファ	349	698
ソ	370	740
ソ#	392	784
ラ	415	831
ラ#	440	880
シ	466	932
シ#	494	988

図 3 音階と周波数の関係

※ある音を基準として周波数比が 2 倍になる音が 1 オクターブ上の音となる。

(2)回路の作成

インターネットで調べた回路を参考にしながら、スイッチを押すと LED が点灯し、ボタンに対応した音が出る 2 オクターブ分の電子ピアノの回路図を作成し(図 4)、動作確認を Tinkercad で行った(図 5)。Tinkercad とは 3D デザインや電子工作、プログラミングなどをウェブブラウザ上で行える 3D モデリング

ソフトウェアである。図 4、図 5 は、2 音分の、スイッチを押すと LED が点灯し、ボタンに対応した音が出る回路である。繰り返しであるため 3 音目以降を省略している。

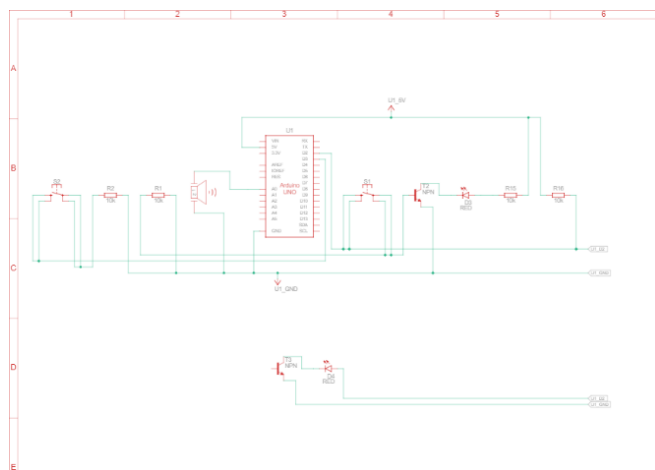


図 4 回路図

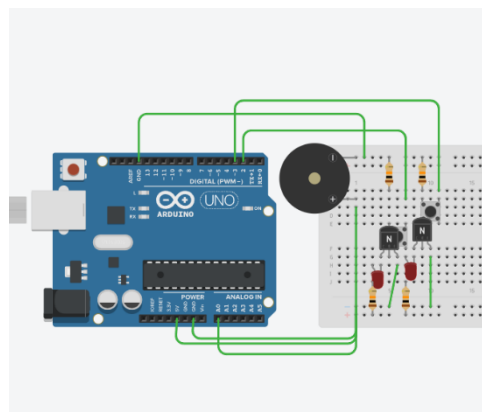


図 5 Tinkercad の配線

ブレッドボードで正常な動作を確認できたら、実際にはんだ付けを行う予定だったが、配線の数が多く回路を組むことが困難であったため、ユニバーサル基板での製作からブレッドボードでの製作に切り替えた。ブレッドボードで使用するジャンパー線は、全長が長く、配線するとアーチ状になり鍵盤が浮き上がってしまうため、短いジャンパー線を使用した。

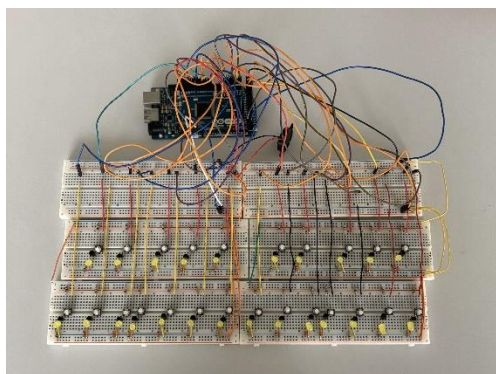


図 6 ブレッドボードの配線

(3) 追加機能について

右端の「シ」のボタンを押すと音楽が流れるよう仕様変更を行った。2 オクターブ分がびったり収まるように設計していたため、ボタンを新たに付け足すことができなかった。よって、右端のシのボタンを使用することにした。追加したプログラムは図 7、図 8、図 9 の通りである。

```
int PIINO=13; //スピーカーのピン番号
int button=2; //スイッチのピン番号
bool isPlaying=false;

#define NOTE_C3 131
//音階の定義(略)
```

図 7 変数の宣言

is Playing で音楽が再生中であることを確認し、再生中であればボタンを押されても反応しないように設定した。音階の定義では、使用する音階のみを定義している(図 7)。

```
void setup() {
  pinMode(button, INPUT_PULLUP);
  pinMode(PIINO, OUTPUT);
}

void loop() {
  if(digitalRead(button)==LOW&&!isPlaying) {
    isPlaying==true;
```

```
playMusic();
isPlaying==false;
}
}
```

図 8 ピン設定

setup 関数では、button ピンを入力モード、PIINO ピンを出力モードに設定した。loop 関数では、ボタンが押されている、かつ、音楽が再生されていないときに音楽が流れる。また、音楽が再生されているときは、isPlaying が true に設定され、音楽が再生されていないときは false になる(図 8)。

playMusic 関数では実際に再生する音楽のプログラムを行った(図 9)。

```
void playMusic() {
  Tone(PIINO, NOTE_G4, 300);
  delay(350);
  //音楽の譜面のプログラム(省略)
  delay(10000);
}
```

図 9 再生するプログラム

(4) 初期案について

当初、Arduino Uno で研究を進めていたが、1 オクターブ分の配線だけで使用可能なピンのほとんどを占有してしまい、ピン数が不足するという問題が発生した。この問題は、ピン数が 54 本ある Arduino Mega への変更や、シフトレジスタを用いて出力ピンを増やすことで解決できる。しかし、私たちは、ピン数を節約しつつ、より多くのスイッチを確保できる「キーマトリクス」という手法に挑戦することにした。

キーマトリクス回路とは、スイッチの数を効率的に増やすための配線手法であり、行列状に配線することで、少ないピン数で多くのスイッチを制御できる仕組みである。この手法は、キーボードや電子楽器など、スイッチ

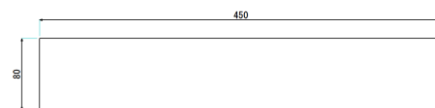
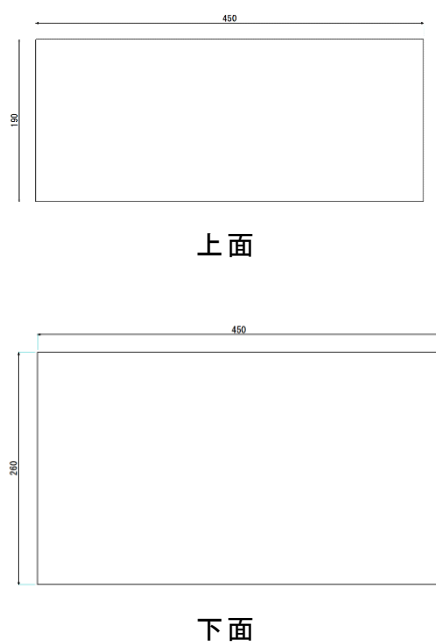
が多数必要なデバイスで広く使用されている。キーマトリクス回路には、配線の簡易化や、コストの削減といった利点がある。しかし、キーマトリクス回路を用いて研究を進めるうえで、チャタリング問題が浮き彫りになった。

ボタンが押される際、物理的な接触不良により、同じキーが数回押されたかのように信号が不安定に発生してしまうことがあった。チャタリングが起ると、音楽を演奏する際に誤作動を引き起こす可能性がある。そのため、スイッチが押されたら信号が安定するまで少し待ったり、スイッチの入力を受けた後にディレイを実行したりといった対策が考えられるが、リアルタイム処理がむずかしくなり今回の研究には適していなかった。

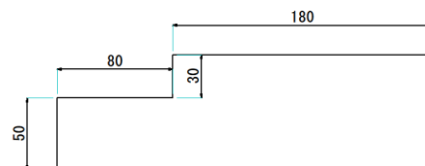
キーマトリクス回路の製作には多くの時間を費やしたが、期間内に完成させるのが難しいと判断し Arduino Mega に切り替えて研究を続行することとなった。

(5) 外装の製作

電子ピアノの内部を観察できるように、外装をアクリル板で組み立てることにした。設計は Fusion360 という 3DCAD ソフトウェアを使用して行った。



背面



側面

図 10 外装図

今回使用したユニバーサル基板は、 95×72 mm の小型のユニバーサル基板である。4 枚使用するため、横の寸法は最低でも 380mm 必要であり、横幅に余裕を持たせるため 450mm で設計を行った。縦の寸法は、Arduino を置くスペースやジャンパー線の長さを考慮して 260mm とした。高さの寸法は、Arduino とその下にジャンパー線が通る空間を確保できるように 80mm に決定した(図 10)。

次に、鍵盤の製作を行った。鍵盤は、白鍵が 14 鍵、黒鍵が 10 鍵の 24 鍵である。白鍵の縦の寸法は Arduino の縦の長さである 72mm に組み立てた際の傾斜分を加えて 80mm とし、横の寸法は Arduino 4 枚分の長さである 380mm を白鍵の数で割った 27mm とした(図 11)。

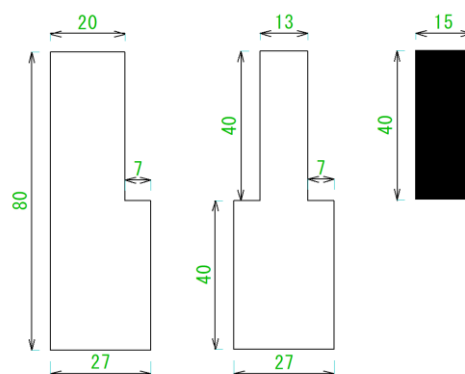


図 11 鍵盤の設計図

このままではまだ押すことができないので、3D プリンタを用いて鍵盤とアルミ棒の接続部品の製作を行い、組み合わせることで押す動作を実現することにした。接続部品は、鍵盤の上面の寸法に合わせて横幅を 20mm、13mm、15mm の三種類、上面の穴は、ねじの太さに合わせて 5mm、側面の穴はアルミ棒の太さより少し余裕を持たせて 5mm で設計した(図 12)。

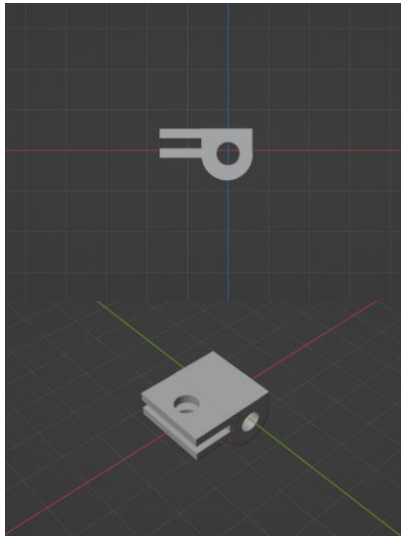


図 12 接続部品モデル

印刷した接続部品には個体差があるので、差込口のやすりがけを行いすべての鍵盤が奥まで差し込めるように調整を行った。鍵盤と接続部品は 5mm のねじとナットで固定した(図 13)。



図 13 接続部品と組み合わせた鍵盤

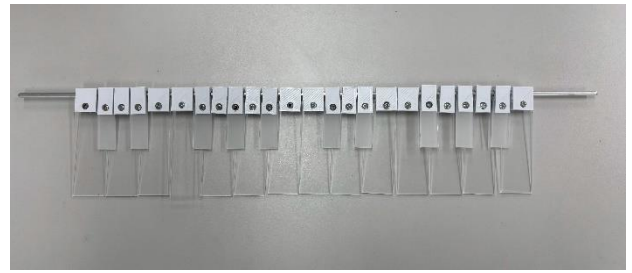


図 14 鍵盤全体図

部品が全て揃ったので、組み立てを行った。アクリル板同士の接着にはアクリル接着剤を使用し、仮留めなどを行いながら丁寧に作業した。

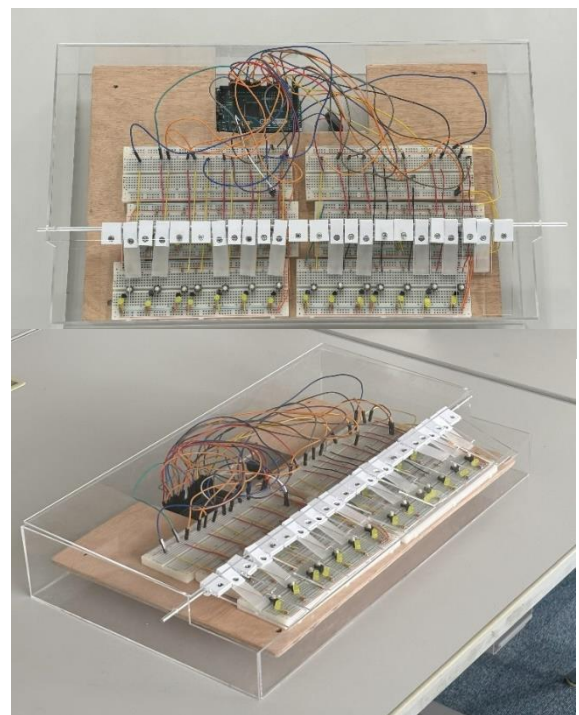


図 15 完成した電子ピアノ

アクリル板をメインとした統一感のある電子ピアノが完成した。想定していた、押すと光る 2 オクターブのほかに、音楽を再生するという新たな機能も追加することができ満足のいく電子ピアノを作ることができた。

3. 研究のまとめ

課題研究を通して、電子回路の知識を深め、材料加工の技術を身に着けるという研究目標を達成できたように思う。私たちの研究は、行き詰まることが多く完成もギリギリであったため、準備不足や計画性のなさを感じる場面が多かった。

さらに、キーマトリクス回路を完成させることができず、悔いが残る結果になってしまった。しかし、このテーマに取り組んだおかげでキーマトリクス回路の仕組みを学ぶことや、Arduino Mega というピン数の多いマイコンが存在することや、シフトレジスタを使って出力ピンを増やす方法を知ることができた。

外装面では、3D プリンタを使用した製作において、実習時は限られた機能で指示通りに作業していたが、課題研究ではビジュアルや寸法を細かく設定し、試作を繰り返すことで思い通りに設計できるようになった。

このように、課題研究が始まる前よりも知識や技術を向上させることができた。今後、機会があればキーマトリクス回路にも挑戦してみたいと考えている。

参考文献

- ・電子ピアノについて

<https://electwork.net/arduino2/>

- ・キーマトリクス回路について

<https://geekyfab.com/entry/2020/10/17/201500>

- ・Blender の使い方

<https://tonatosho.com/blender-boolean/>