

16 入力デバイスに水を用いて音を出力・制御する楽器の製作

廣藤 美緒

指導教員 菅野研一

1. 目的

この研究では、フィジカルコンピューティングを使用したアート作品を製作すると共に、フィジカルコンピューティングについての理解を深め、それに応じた様々な技術を新たに学び、身につけていくことを目的としている。

2. 研究概要

2.1 作品概要

この研究では昨年度の卒業研究を基に作品の製作を進めている[1]。昨年度は光センサによる水の揺らぎの検出であったが、接続に不安定な部分もあり別の方法で検出することはできないかと考え本研究では電極による抵抗値の変化によって水の揺らぎを検出する方法を用いた。

2つの電極を使用して水の揺らぎ（抵抗値）を利用し、その値によって音を出力・制御する作品を製作する。作品は、容器に入っている水の揺らぎから測定される波形（抵抗値）を入力装置として、情報を入力する制御装置 Arduino Mega2560、検出された抵抗値によって音の出力・制御を行う PC によって構成される。

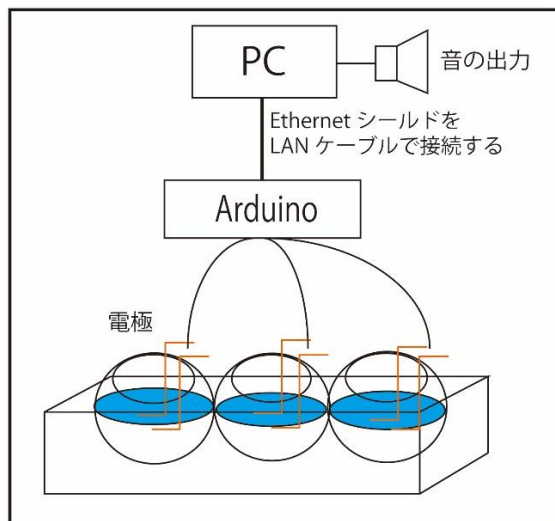


図1 完成イメージ

2.2 開発環境

2.2.1 ソフトウェア

Arduino-IDE

Processing ベースでソフトウェア開発に不慣れなアーティストでも容易にプログラミングができるように設計されている[3]。

Pure Data

音声、ビデオ、映像処理のためのリアルタイムなグラフィカルプログラミング環境。ネットワークを絡めたプロジェクトや、Arduino と組み合わせた自作楽器の製作も可能である[4]。

2.2.2 ハードウェア

Arduino Mega2560

今回の研究ではアナログ入力が9本必要になるため、水の揺らぎの情報を入力する制御装置にアナログ入力が16本あるArduino Mega2560を使用する。

また、制御装置とPCはネットワークで接続するためArduino MegaにEthernetシールドを接続する。EthernetシールドのLANポートとPCをLANケーブルで接続する。EthernetシールドとArduino Megaの間にはバニラシールドを使い可変抵抗器等を設置している。バニラシールドに取り付けている部品などについては3.2 バニラシールドで説明をする。



図 2 Ethernet シールド(上), Arduino Mega(下)
バニラシールド (中)

3. 作品について

3.1 完成品

今回の研究で製作した作品は、容器に固定した 2 つの電極から水の揺らぎを検出し、検出した値によって音の出力・制御を行うものである。

使用者は「水をかき混ぜる」等の方法によって、水に変化を与える。その時、水の揺らぎによって抵抗値が変化する。検出した抵抗値は Arduino を介して Pure Data に送られる。Pure Data では送られてきた値によって音の出力・制御を行う。様々な音の出力を行うために容器は 9 つ使用している。

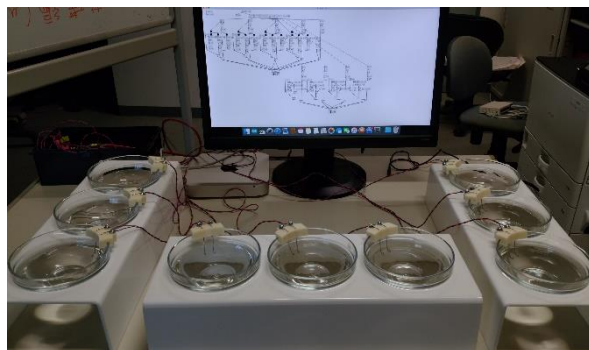


図 3 作品の全体図

3.2 バニラシールド

バニラシールドには可変抵抗器、電極に接続するためのピンヘッダを取り付けている。使用している可変抵抗器は最大で $10K\Omega$ まで抵抗値を変えることができる。水の揺らぎがはっきり検出できた時の可変抵抗器の値の約 $6K\Omega$ に設定してある。可変抵抗器の値を $0K\Omega$ にした時に電極同士が触れ合う

と装置が壊れるので $1K\Omega$ の抵抗を直列に入れた。図 4 は作品の回路図の一部である。容器は 9 つあるので片方の電極は A7～A15 に接続されている。

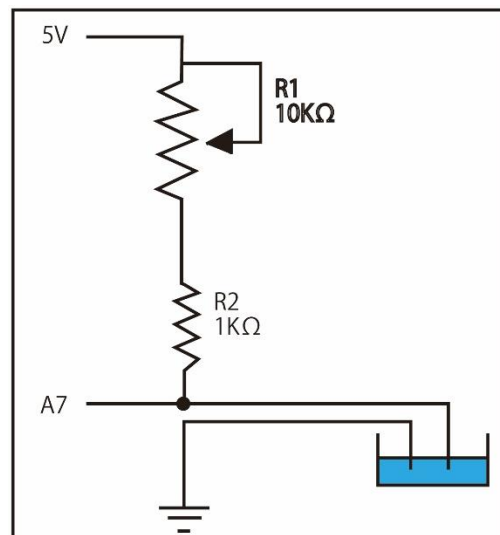


図 4 回路図

3.3 電極と固定部品について

2 つの電極を固定するためにテープを使用していたが、不安定な部分や見栄えが悪いこともあり 3D プリンタで固定する部品の作成を行った。モデリングには Tinkercad を用いた。Tinkercad とはウェブベースの 3DCAD アプリケーションである[5]。モデリングした部品と実際に 3D プリンタで作成した部品は図 5 の通りである。

部品には電極やネジを通す穴、容器に嵌めるための窪みがある。電極はスズメッキ線を使用していたが錆びやすいことが分かったため、錆びにくいステンレス製の電極を使用している。

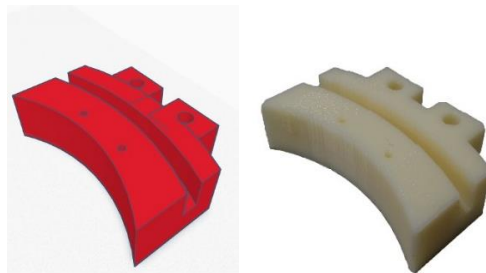


図 5 モデリングした部品(左)と 3D プリンタで作成した部品(右)

3.4 検出する値について

今回の研究では周囲の変化を取り除き、細かい揺れを検出したいため移動平均とハイパスフィルタを使用した[6][7].

$$y[n] = (x[n] + x[n-1] + \dots + x[n-(N-1)]) / N$$

$x[n]$ // センサで取得した値
N // データの範囲

図 6 移動平均の式

$$y[i] = x[i] - \text{移動平均}$$

図 7 ハイパスフィルタの式

Arduino で作成したプログラムでは、移動平均で 5 つ分の平均を取り、ハイパスフィルタで現在の値から移動平均で求めた値を引いた。また値は Arduino で絶対値を求めてから Pure Data に送信している。

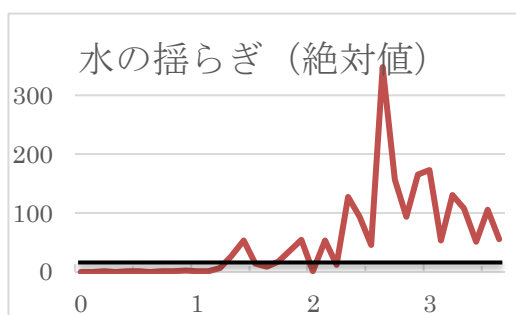


図 8 揺らぎの検出

絶対値が閾値（グラフでは黒い線の部分）以上になった時に音ができるように Pure Data のパッチを作成した。

3.5 出力する音について

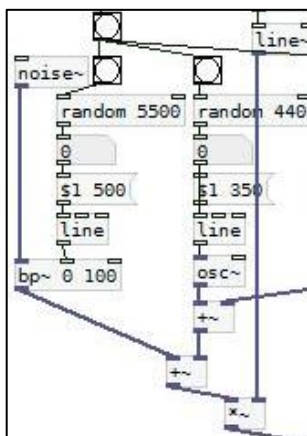


図 9 ランダムな正弦波+ノイズ

図 9 のパッチは抵抗値が検出されると random で記述している数値の範囲で値が変化していく。2 つの値を加算することでランダムな正弦波とノイズの音を作成することができる。

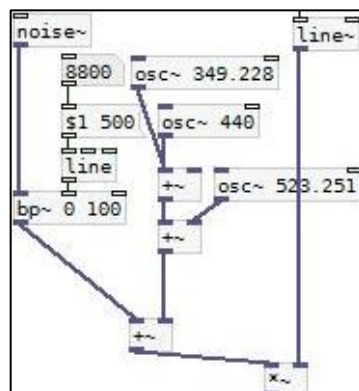


図 10 和音+ノイズ

図 10 のパッチでは osc~ で記述した正弦波の値を加算していくことで和音を作成している。ノイズと先ほど加算した正弦波をさらに加算することで和音とノイズを組み合わせた音を作成できる。このパッチでは、ノイズの値は任意で変更することができる。

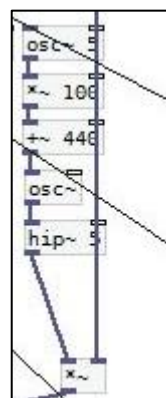


図 11 周波数が上下する音

図 11 のパッチでは上の osc~5 の出力に 100 倍し 440 を加算し下の osc~ に入力することで周波数が ±440Hz 上下する。

3.6 OSC プロトコルについて

OSC プロトコルとは、コンピュータやサウンドシンセサイザー等の機器同士でネットワークを介したデータの送受信を行うための規格のことである。特徴は、データの送受信が高速なこと、Pure Data や Processing 等のソフトウェアの

データ送受信が可能だということである。

OSC はシンプルな構成で、大きく 2 つのメッセージがあり、そのメッセージをアプリ間でやり取りしながら通信を行う [8][9]。図 13 は Arduino-IDE に記述した OSC プロトコルに関する部分の一部である。Pure Data には msg.add() に記述した部分のデータが /sensor7~/sensor15 の各メッセージに送信される。



図 12 OSC プロトコルの仕様

```
OSCMessage msg("/[sensor7,sensor8,sensor9,sensor10,sensor11,sensor12,sensor13,sensor14,sensor15]");
msg.add((int32_t)abs(aveValue[0]));
Udp.beginPacket(destIp,destPort);
msg.setAddress("/sensor7");
msg.send(Udp);
Udp.endPacket();
msg.empty();
```

図 13 OSC プロトコルに関する記述

4. 終わりに

作品を製作するにあたって、Arduino や Pure Data 等新しい技術を学ぶことができた。Pure Data に関しては今まで学んできたプログラムの書き方とは異なり、オブジェクトを線でつないでパッチを作成していくものだったためこういうプログラムの書き方があることを知ることができた。

作品に関しては、使う予定の基板が届かない等があったが無事に完成させることができた。製作の途中で Pure Data と Arduino の通信ができなくなったことがあったが、原因は機器同士がしっかりと接続できていなかったことだったためハードウェアを扱う時には接続がきちんとできているかどうかをしっかりと確認することが大切だと思った。

今回の研究では音の作成のみだったが、音に合わせてディスプレイに画像や動画を表示する機能を追加すれば体験者は音を視覚的に感じることもでき、面白いのではないかと思う。

全体的に、様々な技術を学び楽しみながら研究を進めることができたので良かった。今回学んだことは別の機会に活かしたいと思う。

5. 参考文献

[1]入力デバイスに水を用いて音を生成するアート作品

(産業技術短期大学校矢巾校卒業研究報告書
2016 上山明江, 東山真実)

[2]フィジカルコンピューティング

<http://artscape.jp/artword/index.php/%E3%83%95%E3%82%A3%E3%82%B8%E3%82%AB%E3%83%BB%E3%82%B3%E3%83%B3%E3%83%94%E3%83%A5%E3%83%BC%E3%83%86%E3%82%A3%E3%83%B3%E3%82%B0>

[3]Arduino Mega 概要

<https://trac.switchscience.com/wiki/ArduinoMega>

[4]Pure Data Japan Pure Data とは？

http://puredatajapan.info/?page_id=2

[5]Tinkercad とは

<http://nakayasu.com/lecture/tinkercad%E5%9F%BA%E7%A4%8E/4319>

[6]fugaga の備忘録ハイパスフィルタ

<http://fugaga.hatenablog.com/entry/2014/02/16/164524>

[7]移動平均フィルタ

http://bluefish.orz.hm/sdoc/fftw_digifil.html

[8]Open Sound Control (OSC) 入門

http://r-dimension.xsrv.jp/classes_j/osc/

[9]初めての電子工作超入門

<http://deviceplus.jp/hobby/entry0030/>

[10]こじ研 (音・映像メディア)

<https://www.ei.tohoku.ac.jp/xkozima/lab/avmed-pd2.html>

[11]Pd 入門 2

<http://yoppa.org/ssaw13/4567.html>