

Arduino を用いた電子ドラムの作製と評価

2015SC024 今井栄希 2015SC040 金井抄悟

指導教員：奥村康行

1 研究背景

はじめに、ドラムは広いスペースを使用するため場所の問題や、大きな音が出るため騒音の問題が発生する。従って、ドラムの演奏を行う際はスタジオを借りる必要がある。そこで、時間的な制約や予算面などを鑑みて、そのような手間をとらず、気軽に自宅で練習を行うために電子ドラムが普及してきた。しかし、現在の電子ドラムはパッドを使用するため、場所の問題は解決しておらず、スペースに余裕がない場合は従来のドラムも電子ドラムすら使用できない状況である。さらに、電子ドラムは音は鳴らないがパッドを叩くため、叩いた際の音や振動は床に伝わるため静音効果は得られない。本研究では、場所を制限されず騒音の心配がない電子ドラムの作製を行う。

2 Arduino を用いた電子ドラムの構成と課題

Arduino と呼ばれるハードウェアや MIDI について、さらに使用するセンサについての説明を述べる。さらに、Arduino を用いた電子ドラムの概要と技術課題を述べる。

2.1 電子ドラムの概要 [1]

今回、作製する電子ドラムの概要を説明する。手の動きを加速度センサで検出し、Arduino にアナログ信号として送信する。Arduino からリズムトラックに送信する際は MIDI ケーブルを通じてデジタル信号を送信する。

4 つの加速度センサから受けた信号を、Arduino で処理して MIDI 信号をリズムトラックに送ることで電子ドラムの完成とする。このような電子ドラムの構成にすることで、今回作製する電子ドラムはパッドを叩くわけではないのでスペースに余裕がなくとも演奏することが可能で騒音や振動の問題も解決することが出来る電子ドラムとなる。

加速度センサ、Arduino と DIN コネクタの接続に関する回路を作製した [1]。回路構成を図 2 に示す。

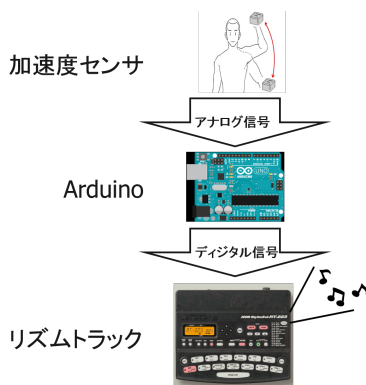


図 1 電子ドラムの全体構成

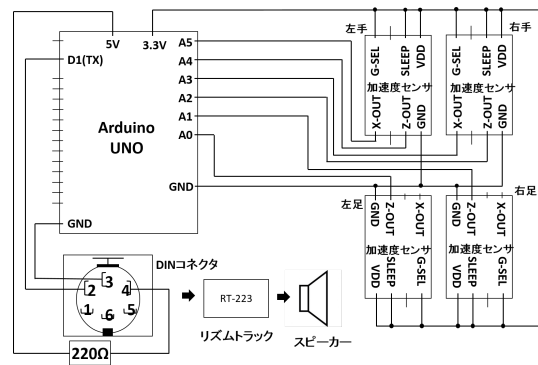


図 2 回路構成 [1]

上記の図 2 で、DIN コネクタの 2 ピンがシリアル通信のための TX ピンであり、3 ピンがグラウンドの役割を果たし、4 ピンが電源の役割を果たす VCC ピンである。以上が加速度センサ、Arduino と DIN コネクタの接続の回路図である。

2.2 MIDI について

MIDI(Musical Instrument Digital Interface) とは、デジタル信号によって電子楽器とコンピュータまたは電子楽器同士を相互に接続し、演奏情報をやりとりする世界統一規格である [2]。リズムトラックと Arduino を接続するために私たちは DIN(Deutsche Institute Norm) の 5 ピンタイプである MIDI ケーブルを使用する。

2.3 Arduino について

Arduino は、シンプルな入出力ポートやスケッチと呼ばれるプログラムの開発環境が用意されているものである。プログラミング言語は C 言語をベースとした、オープンソースハードウェアなので誰でも自由に使用できる利点がある。

Arduino を構成する要素として主に 2 つあり、ハードウェアとソフトウェアに分けられる。ハードウェアである Arduino ボードは入出力として使用する 14 本のデジタル I/O ピンやセンサからのアナログピンなどを搭載している。また、ソフトウェアは開発を行うために必要な ArduinoIDE(Integrated Development Environment) という専用ソフトウェアを使用する [3][4]。

マイコンの例として、Arduino や RaspberryPi, PIC などが挙げられる。RaspberryPi は Arduino よりも特性の関係より、リアルタイム性を保証するのが難しくなる。PIC と Arduino では同じ周波数のクロックの場合では、Arduino のほうが処理が速くなる。さらに、初期設定が非常にシンプルで、入出力端子に番号や記号が書いてある初

心者にも扱いやすいという点から、マイコンは Arduino を選択する。

2.4 加速度センサについて

ドラム演奏時の手の動きを感知するため、3軸の加速度センサを使用することで手足の動きを感知させる [5]。本研究では3軸加速度センサモジュール MMA7361LC を使用する。

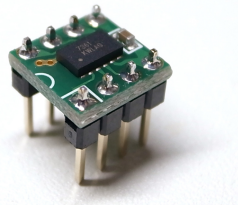


図3 MMA7361LC[6]

表1 加速度センサのスペック [6]

軸数	3軸 (X, Y, Z)
センサ	Freescale MMA7361L
検出範囲	$\pm 1.5G$ または $\pm 6G$
感度	800mV/G ($\pm 1.5G$ レンジ 3.3V 動作時) 206mV/G ($\pm 6G$ レンジ 3.3V 動作時)
電源電圧	DC2.2V ~ 3.6V
消費電力	400 μ A 標準
モジュールサイズ	約 10 × 10mm
応答周波数	DC ~ 1500Hz
ノイズ	350 μ G/ $\sqrt{\text{Hz}}$
重力加速度検出	可能
0g 検出	可能
低消費電力モード	有

2.5 技術課題

マイコンを使用した電子ドラムであるため、遅延時間が発生するものと考えられる。先行技術ではマイコンを用いた電子楽器のシステム構成のみの提案だが、本研究で作製する、マイコンを用いた電子ドラムでは、その遅延時間の特性を明らかにする。

3 Arduino と加速度センサを用いた動作確認

今回、Arduino を用いた電子ドラムを作成するにあたって、Arduino と加速度センサについての動作確認並びに、理解を深めることを目的とし、簡単なプログラムを作成し加速度センサと Arduino を用いた実験を行った。

3.1 Arduino を用いた LED 点滅

まず、Arduino を用いて LED を点滅させる実験を行う。Arduino のプログラムを作成し、Arduino の制御によって LED を点滅させることで Arduino の動作確認を行った。LED を点滅させるための回路構成として、今回の実験では Arduino の GND と 270 Ω の抵抗を接続し、LED と DIGITAL_1 ピンをつなげた状態で LED が点滅するかどうかの確認を行う [7]。同様に DIGITAL_13 ピンまで実験を行い、正常に動作した。これにより、Arduino の DIGITAL ピンに関しては動作確認を完了した。

抵抗を使用しなくとも実験は行うことが可能だが、LED を長時間点灯させる場合は抵抗を使用するため、今回は抵抗を使用する [4]。

3.2 Arduino と加速度センサを接続しての確認

次に Arduino と加速度センサを接続し、加速度センサの傾きを計測することで動作確認を行う。センサの中で構成される部品にかかる重力を検知することで、傾きを検出することが可能となる [8]。

Arduino と加速度センサを接続しているので Arduino のシリアルモニタを確認することが可能になる [9][10]。よって傾きを Arduino のシリアルモニタに表示することで Arduino と加速度センサが正しく接続され、正しく動作することが確認できる。この Arduino と加速度センサを使用した動作確認の回路構成は、図2の加速度センサの VDD と Arduino の 3.3V を接続し、加速度センサの GND と Arduino の GND を接続する。また、加速度センサの X-OUT, Y-OUT, Z-OUT をそれぞれ Arduino の Analog_0, Analog_1, Analog_2 に接続する。これにより、加速度センサと Arduino の接続がシリアルモニタにて確認することが可能となった。これを用いて動作確認を行った。

3.3 MIDI 信号が入力されているかの確認

加速度センサと Arduino の接続に関する動作確認が完了したため、図2の配線を行った後、MIDI 信号が入力されているかモニタリングを行った [11]。

MIDI 信号が入力されることを確認することで、加速度センサから Arduino に位置の変化のアナログ信号が送られ、MIDI-OX 上で MIDI 信号の入力の確認を行い動作確認を行う。Arduino からの MIDI 信号が入力されていることを確認された。

4 作製した電子ドラム

図4が本研究で作製した電子ドラムである。

実際に本研究で作製した、Arduino を用いた電子ドラムの実演動画は、南山大学理工学部機械電子制御工学科 奥村・藤井研究室のホームページ (http://www.st.nanzan-u.ac.jp/info/okumura_fujii_lab/drums.html) に掲載されている。

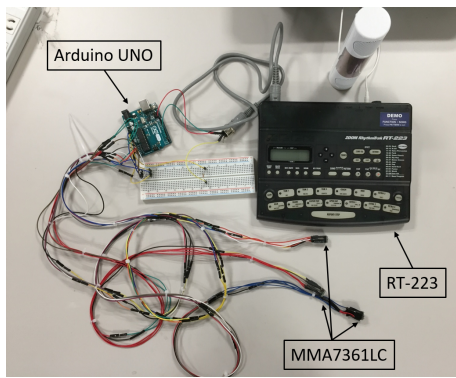


図 4 完成した電子ドラム

図 4 で、黒いものがリズムトラックであり、左上の緑色のものが Arduino であり、ジャンプワイヤの先端に加速度センサがはんだ付けしてある。

5 各装置の遅延時間

加速度センサから位置情報を検知し、Arduino で位置情報を処理しリズムトラックに MIDI 信号を送るまでに加速度センサ-Arduino 間と Arduino-リズムトラック間がスムーズに通信出来ているのか確認するため各装置間に遅延時間が生じるのか実験を行う。さらに、加速度センサを同時に動かす数を増やすことで遅延時間に違いが生じるのかを確認する。

5.1 実験方法

オシロスコープで加速度センサ-Arduino 間と Arduino-リズムトラック間を同時に観測することで生じる波形の差より遅延時間を測定する。

加速度センサを 1 つ動かす場合は、加速度センサ-Arduino 間の波形を得るためにオシロスコープの CH1 を Arduino の ANALOG.5 ピンに接続し、Arduino-リズムトラック間の波形を得るためにオシロスコープの CH2 を図 5 の DIN コネクタの 2 ピンに接続する。そして、加速度センサの動かした際の入力と出力と、通常時の入力と出力の波形を比較する。加速度センサを同時に 2 つ動かす場合も同様に実験を行い、加速度センサを同時に 2 つ動かす。加速度センサを同時に 3 つ動かす場合も同様に実験を行う。

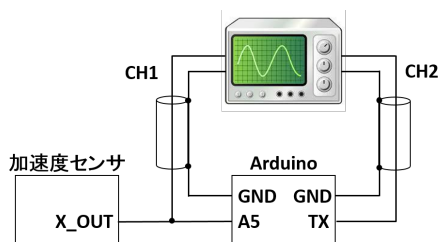


図 5 実験の略図

5.2 実験結果

加速度センサを動かし、音が出されるまでに遅延時間は感じられなかった。しかし、回路を通じるため遅延時間が発生しているものだと考える。さらに、同時に動かす加速度センサの数を増やすほど Arduino の処理に負荷がかかるため遅延時間が生じるものだと考える。実際に遅延時間について測定した結果、遅延が生じていることが判明した。さらに、同時に動かす加速度センサの数が増えるほど、遅延時間が大きくなっていくことがわかる。以下が今回の遅延時間の実験の結果である。

図 7 の画像は加速度センサを動かしていないときのものである。Arduino-リズムトラック間では動きに関係なく CH2 が信号を送信している。CH2 の波形の最大幅が 2.4ms であり、これより波形の幅が大きく、最大のものと CH1 の波形を比較することで遅延時間の計測を行う。

図 8 は加速度センサを 1 つ動かしたものであり、1 目盛 4ms の差が 3 目盛のため 12ms の遅延が発生したことになる。図 9 は加速度センサを同時に 2 つ動かしたものとなり、1 目盛 4ms の差が 9 目盛のため 36ms の遅延が発生したことになる。図 10 は加速度センサを同時に 3 つ動かしたものとなり、1 目盛 4ms の差が 5.5 目盛のため 22ms の遅延が発生したことになる。図 7 は加速度センサを動かしていない時のものである。

今回は実験を各 30 回行い、その結果の遅延時間の最小値、最大値、平均値を図 6 にまとめる。

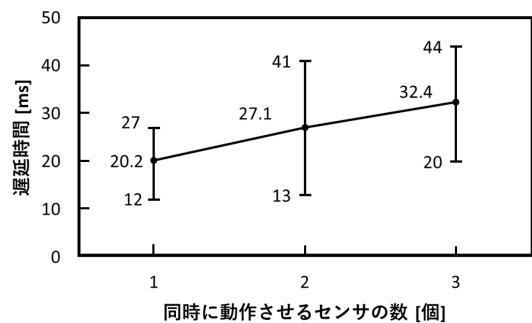


図 6 同時に動かす加速度センサの数と遅延時間について

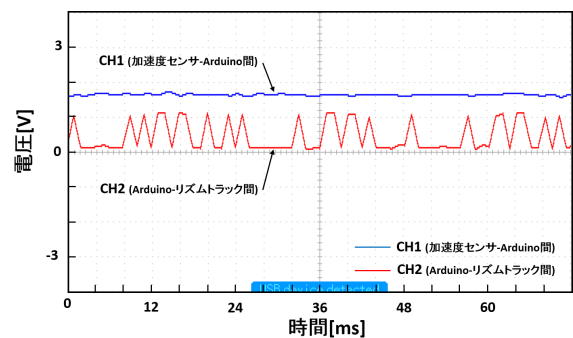


図 7 加速度センサを動かさない際の波形

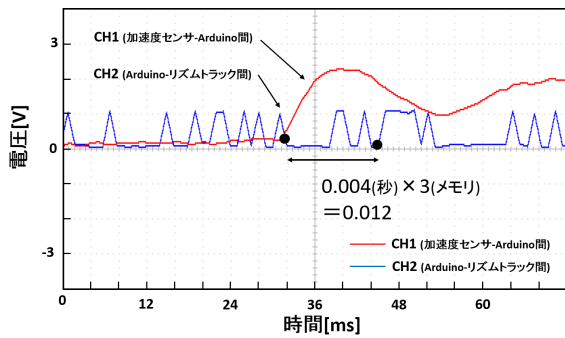


図 8 加速度センサを 1 つ動かした際の波形

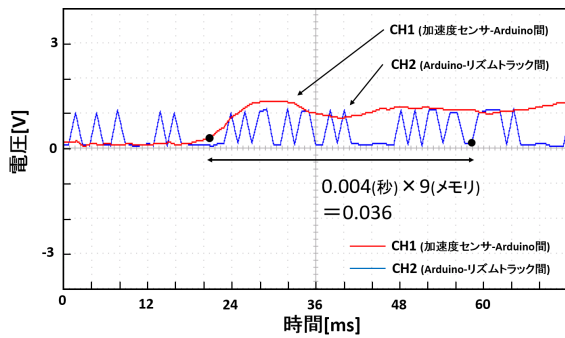


図 9 加速度センサを 2 つ動かした際の波形

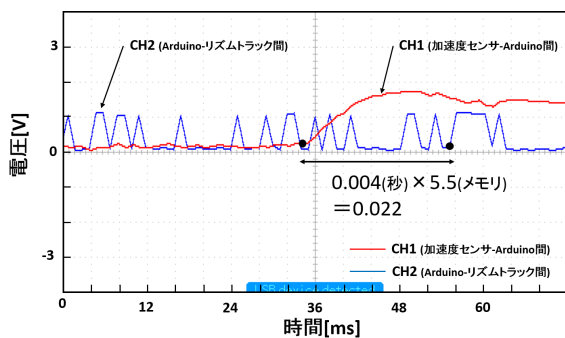


図 10 加速度センサを 3 つ動かした際の波形

6 実験の考察

図 6 の実験結果より、同時に動かす加速度センサの数が多ほど遅延時間が長くなることが確認できた。考えられる原因としては、同時に動かす加速度センサの数が増えると Arduino での処理時間が長くなるため、同時に動かす加速度センサの数が増えるほど遅延時間が発生するものだと考えられる。この技術課題を解決させるために、Arduino の処理能力を向上させる方法として、Arduino を複数個使用することで遅延時間の減少が見込めると考えられる。しかし、本研究での作成した電子ドラムは Arduino からの出力が 1 つしか使用できないため、Arduino を複数個使用して処理能力を向上させることは難しい。

7 おわりに

遅延時間の結果を見てみると、加速度センサを同時に動かす数が増えると遅延時間が大きくなっていく。しかし、電子ドラムを演奏する際に、遅延時間を感じることはなかった。これは、遅延時間が非常に小さい値のため、電子ドラムを演奏する際には感じることはできなかったためである。

電子ドラムを演奏する際に、意図しない音が発生することがあった。これは、Arduino と加速度センサを接続するジャンパワイヤが短いため、数多くのジャンパワイヤの延長をした結果、接続がうまくいかない場合があり、その際に意図しない音が発生したものと考える。この改善策として、ビニル導線を使用する案が考えられる。本研究の中で実際に改善策としてのビニル導線を使用した際、ビニル導線が干切れることや、はんだ付けを行った部分から取れてしまうことがあった。この改善策として、ビニル導線を太くて丈夫なものを使用すること、はんだ付けを適切量を行うのが望ましいと考える。その他に改良ができる箇所としては、本研究はプログラムを試行錯誤的に作製したため、思うように簡単には扱えなかったりしたり、意図した音が完全に鳴らないことがあったのが今後の課題となる。

参考文献

- [1] Arduino でエアドラムのブチャラティ作成, <https://karaage.hatenadiary.jp/entry/20090912/1252782573>, Aug. 2018.
- [2] 社団法人 電子情報通信学会 編, “改訂 電子情報通信用語辞典,” 株式会社コロナ社, 東京, 1999.
- [3] Massimo Banzi (船田巧 訳), “Arduino をはじめよう,” 株式会社オライリー・ジャパン, 東京, 2009.
- [4] Massimo Banzi (船田巧 訳), “Arduino をはじめよう 第 2 版,” 株式会社オライリー・ジャパン, 東京, 2014.
- [5] 鈴木哲哉, “ボクの Arduino 工作ノート 改訂版,” 株式会社ラトルズ, 東京, 2014.
- [6] Strawberry Linux Co, <https://strawberry-linux.com/catalog/items?code=12101>, Apr. 2018.
- [7] 高橋隆雄, “たのしい電子工作 Arduino で電子工作をはじめよう,” 株式会社秀和システム, 東京, 2015.
- [8] はじめての電子工作超入門, <https://deviceplus.jp/hobby/entry016/>, May 2017.
- [9] 神崎康宏, “Arduino で計る, 測る, 量る,” CQ 出版株式会社, 東京, 2013.
- [10] 福田和宏, “これ 1 冊でできる! Arduino ではじめる電子工作超入門,” 株式会社ソーテック社, 東京, 2014.
- [11] MIDI コントローラーからの入力をモニタリングする方法, <https://support.native-instruments.com/hc/ja/articles/209544729>, Jun. 2018.