

# Arduinoによるハモンドオルガンの実現

長嶋洋一<sup>1</sup>

**概要**：1999年に筆者が出版した「コンピュータサウンドの世界」(CQ出版)には、同時進行で執筆した「作るサウンドエレクトロニクス」という上級編もあったが、「幻の本」(未出版)となり全てをインターネットに公開した。ここでは世界の楽器メーカーが開発してきた古典的な電子楽器群を網羅的に解説しつつ、まだArduinoもGainerも存在しない20世紀に秋月電子のAKI-H8やディスクリート回路によって実際にポリフォニックシンセサイザを実現した。2015年以降はArduino用の各種音源shield(MIDI・mp3)が豊富に提供されるようになったが、パイプオルガン/ハモンドオルガンの古典的な音色生成を完全(同時発音数=鍵盤数)ポリフォニックで実現するのはいまだ困難である。GitHubで公開されているArduinoのサウンド生成ライブラリ「Mozzi」は、ビーブ音だったArduinoの生成サウンド(Shieldなど外部ハードウェアを使わない)を画期的に改善させた優れたアルゴリズムであるが、その限界はモノフォニック(単音)にある。Mozziを活用したオープンソース・プロジェクトとしてGitHubで紹介されている「Arduino Organ」は、60年代のHammondサウンドを実現するためにドローバーとトーンホイールの概念にこだわった特殊なシステム(ペーパーマシン)であるが、新しい楽器や音具をデザインする上での有益な多くのポイントを提供している。本稿では、Mozziの内部処理とともにこのシステムのアイデアを再検討し、「幻の本」と共に温故知新として音楽情報科学コミュニティに問題提起してみたい。

## Realizing a Hammond Organ with Arduino

YOICHI NAGASHIMA<sup>†1</sup>

**Abstract.** In 1999, I published a book entitled "The World of Computer Sound" (CQ Publishing), which included an advanced book entitled "Creating Sound Electronics" that I wrote at the same time but not published. Since 2015, we can get various sound Shields for Arduino. The Arduino sound generation library "Mozzi" is available on GitHub. The library "Mozzi" is an excellent algorithm that has revolutionized Arduino generated sounds (without external hardware such as Shield) from cheap beeps, but its limitation is monophonic (single note). The Arduino Organ, an open source project on GitHub, is a special system (paper machine) that sticks to the concept of drawbars and tone wheels to achieve the Hammond sound of the 60s, but offers many useful points for designing new instruments and sound tools. In this paper, we will discuss Mozzi's paper machine. In this paper, I would like to revisit the idea of this system along with the internal processing of Mozzi, and raise the issue to the music information science community.

### 1. はじめに

まず最初にお断りしておかないといけないのは、本稿のタイトル「Arduinoによるハモンドオルガンの実現」についてである。このタイトルから「Arduinoを使ってハモンドオルガンを作った?」という風に誤解されたくないの明記しておくが、正しく書くとタイトルとしては不適切に長くなるので避けたものの、「Arduinoを使ってハモンドオルガンを作ろう、というユニークなアイデアに触発されてポリフォニック電子楽器に関する歴史を概観した、楽器デザインに関する考察としての音楽的/工学的な(温故知新的)お話」というのが真のタイトルである。

次にお断りしておかないといけないのは、本稿に登場するハモンドオルガン(Hammond Organ)というのは、現在、鈴木楽器グループの(株)ハモンド・スズキが販売している「電子楽器のハモンドオルガン」[1]ではなくて、1934年にローレンス・ハモンドによって発明され、1986年までハモンド・オルガンカンパニーによって提供されたオリジナ

ル[2]のみを指している点である。スズキハモンドの電子オルガンB-3[3]が物凄い情熱とこだわりの結晶であることに敬意を表しつつも、本稿においては実際にトーンホイールを回転させるオリジナルのみを対象としている点に注意されたい。なお本稿では素晴らしいHammondサウンドの重要な要素であるレスリー・スピーカーについては(ググッと堪えて)敢えて触れないでおく点も付記しておきたい。

かつて筆者が出版した「コンピュータサウンドの世界」[4](絶版)には、同時進行で執筆した「作るサウンドエレクトロニクス」[5]という上級編もあったが、「音楽方面に弱いCQ出版」編集長の泣きが入って未出版となった。本稿の第2節ではこの2連作をベースとして(可能であれば共立出版「bit」別冊「コンピュータと音楽の世界」第2章を読むことも推奨)、全体の議論の基盤として世界の楽器メーカーが開発してきた古典的な電子楽器のシステムの考え方と歴史を概説する。ソフトウェアばかりでハードウェアに疎い理工系/情報系の学生は、これを機会として是非、この部分をきちんと理解した上で音楽情報科学の道に進むこと

<sup>1</sup> 静岡文化芸術大学  
Shizuoka University of Art and Culture

を切望する。続く第3節では「楽器の王様」パイプオルガンと対比させて、電気楽器から電子楽器に移行する歴史的意義を持つHammondオルガンの特長を考察し、現在でも電子楽器ではその古典的な音色生成を完全ポリフォニック(同時発音数=鍵盤数)で実現できない事実を確認する。続く第4節ではArduinoでのサウンド生成に関して、各種音源shield(MIDI・mp3)とオープンソース・サウンド生成ライブラリ「Mozzi」について整理する。そして第5節で、Mozziを活用したプロジェクトとしてGitHubで紹介されている「Arduino Organ」、すなわち60年代のHammondサウンドを実現するためにドローパーとトーンホイールの概念にこだわった特殊なシステム(ペーパーマシン)を分析・検討する。ここには新しい楽器や音具をデザインする上での有益な多くのポイントがあり、第6節ではそのアイデアの拡張と発展の可能性について議論していく。

## 2. 「コンピュータサウンドの世界」と「作るサウンドエレクトロニクス」

1999年に筆者が出版した「コンピュータサウンドの世界」[4]は、出版社在庫の最後10数冊を筆者が購入し絶版となった。それ以降は筆者が教育の場で使う教材としてのコピー配布を編集長から認めてもらっていたが、2011年3月、筆者のところに研修に来た某県の工業技術試験場・研究者の好意によりPDF化された[4]。まず最初に、エレキギター・フェンダーローズ・ハモンドオルガン等の電気楽器とテルミン・オンドマルトノ・シンセサイザ・電子オルガン等の電子楽器の違いを解説した。そして電子楽器におけるモノフォニック(単音)・ポリフォニック(複音)の構成的な違いを明確にして、ミニムグのような自由な楽音合成は単音楽器で、トップオクターブ方式の電子ピアノや電子オルガンは「鍵盤の全ての音が同時に鳴る」複音楽器として登場した歴史を紹介した。その後、CPUの登場により「発音割り当て」アルゴリズムとして生まれた「アサイナ(assigner)」という新しい概念から、多様なサウンドをポリフォニック生成する電子楽器に「同時発音数」という本質的限界が出現した事情を解説した。さらに同時期に発展したデジタル信号処理(DSP)技術によって登場したのが、時分多重化処理によって1チップでポリフォニック楽音生成を実現する「音源チップ」であり、この流れは現在の電子楽器まで続いている。

この本と同時進行で筆者が執筆した「作るサウンドエレクトロニクス」[5]は「実際に工学的に解説/製作することを目指した上級編である。まず第1章では「サウンドエレクトロニクス・システム」全体を概観して、「通信カラオケ装置と電子ピアノがシステムとしてほぼ同じ」という基本的な理解を確認し、続く第2章ではアナログ的なサウンドエレクトロニクスについて具体的な回路や信号と共に解説した。そして第3章では、1980年代に登場したMIDI規格、秋月電子のカードマイコンAKI-H8などのデジタル技術の紹介から、実際にAKI-H8を使ったデジタルオシレータやピッチトラッカーのシステム実現まで紹介した。続く第4章は類書の無い高度でユニークな内容となっているが、

楽器業界に特有の技術的課題や解決方法、3台のAKI-H8によって実際にVCO/VCF/VCAブロックを構成して、モノフォニックな古典的アナログシンセサイザ機能をデジタル的に実装するという具体例までを、回路図やアセンブラ・ソースコードの完全公開と共に提供した。乗算命令を持つAKI-H8アセンブラのデジタルフィルタによりVCFを実装する事例まで公開したが、GitHubが世の中に登場するより10年以上も前のことである。

最後の第5章では1万ゲート以上の「音源チップ」に相当する高度なシステムを実際にディスクリート回路として実現した。時分割パイプライン処理で同時に16チャンネル・ポリフォニック楽音生成を実現するDSPシステムであり、正確なピッチを生み出す位相累算ダイレクトシンセサイザ、単なる波形メモリ読出し(PCM)方式でない時間的音色変化(タイムバリエーション)、デジタルエンベロープジェネレータを実装し、実際にAKI-H8でMIDI入力に対して動くアサイナにより発音する。半田付けして実際に製作すると共にシステム設計の全貌や全回路[6-8]を詳細に解説し、いくつものMIDIシーケンス情報で自動演奏させたサンプル音響mp3まで全てを公開している[5]。ユニバーサル基板にAKI-H8と共にICやメモリやD/Aコンバータを計73個搭載したこの基板[9-10](図1)は現存しており、1999年から公開しているこのサイトでは未知の読者(学生など)からの質問も届いていたが、22年以上が経過してさすがに詳細は忘却の彼方なので、今後の新たな質問は勘弁していただきたい。実際に作って動いたので、よく読めば分かる筈であり、応援できないものの再現/追試は歓迎する。

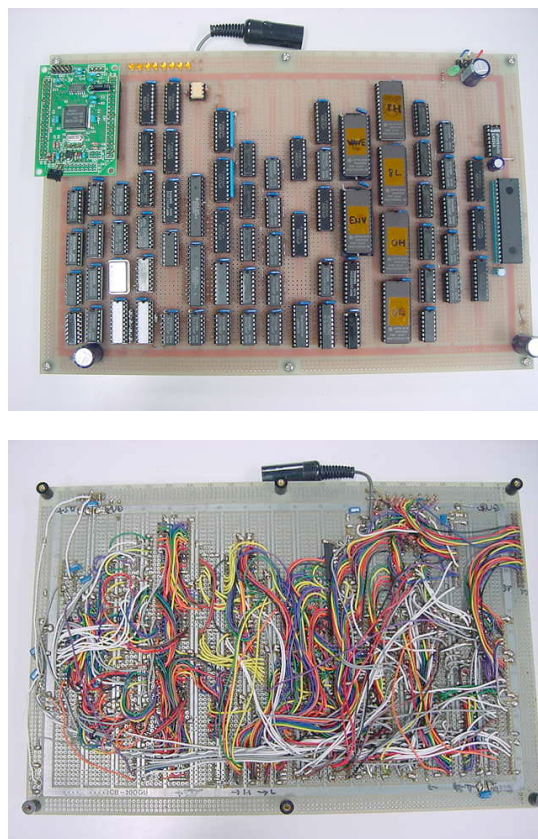


図1 製作した基板。

### 3. Hammondオルガンの特長

音楽音響において最も重要な原理の一つは「協和」であり、「楽器の王様」と言われるパイプオルガン[11]の音色作成の手法、すなわち各種feet率ごとの多数のストップ[12]によって対応した多数のパイプを1つの鍵盤で鳴らすという概念は、音響学的にも王道である。なお本稿ではオクターブを12音のピッチに分割する原理である「音律」[13]については敢えて深入りしない。人間の聴覚特性に親近性のある、鍵盤に対応した音の音色を高次倍音構成として設定するというパイプオルガンのストップ(ON/OFF)を、多段階のハーモニックドロワーによって可変ミックスすると共に、個々の音源波形はトーンホイールからのサイン波のみに限定しているのがHammondオルガンの特長である。なお本稿では、その個性的な音響の要素である「パーカッション」・「ビブラート&コーラス」、および音域の折り返しである「フォールドバック」についても敢えて深入りしない。

鍵盤に対応したドロワーは8feetであり、16feetはそのオクターブ下、4feetはオクターブ上、2feetは2オクターブ上、1feetは3オクターブ上の倍音に相当するが、Hammondサウンドの魅力は純正完全5度(3倍音)および純正長3度(5倍音)の系列に対応するドロワーによって最大限に発揮される。横文字テキストで表現しにくいので「16/3feet」(5と1/3feet)・「8/3feet」(2と2/3feet)・「8/5feet」(1と3/5feet)・「4/3feet」(1と1/3feet)と書くが、分母が3のドロワー3本は鍵盤に対して完全5度上のオクターブ系列、「8/5feet」は鍵盤に対して2オクターブと純正長3度上のサウンドである(図2)。つまり全ての鍵盤について見れば12等分平均律に従った楽器でありながら、メロディー演奏など同時演奏鍵盤が少ない場合には古典的な純正サウンドを生み出すところがHammondオルガンの真骨頂である。マイコン(CPU)どころか鍵盤信号でサウンド系の電気信号をON/OFFするキーイング(ゲート)回路すら無かった時代なので、鍵盤ごとに多数の信号を一気にON/OFFする「多列接点鍵盤」が用いられており、本質的にキークリック音までがその個性としてHammondサウンドを構成している[14]。

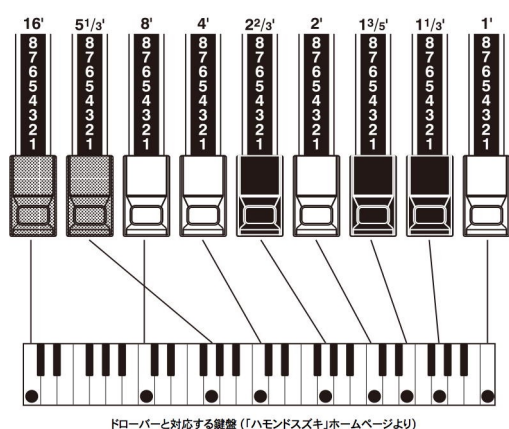


図2 Hammondオルガンのハーモニックドロワー

モーターによって回転するトーンホイールごとにピックアップでその「歯車」の波形を多数のサウンド電気信号と

して取り出し、ドロワーごとの強度でミックスして音色を形成した信号の全てを鍵盤ごとの多列接点で通過させて出力する、というHammondオルガンは「電気楽器」の頂点であると言える。スズキハモンドの電子オルガンB-3[3]では、オリジナルのトーンホイール波形をサンプリングしてD/A変換した後に、敢えて愚直に多列接点鍵盤でミックスしているので完全ポリフォニック(同時発音数=鍵盤数)である。しかし現在の「電気楽器」でハモンド風のサウンドを生成しているシンセサイザや電子オルガンでは、デジタル音源システムに同時発音数の限界があって、アサインによって発音を割り当てているために、MIDI入力でも鍵盤演奏でも、その全ての音が発音されるわけではない。これは楽器店の試奏コーナーに行くと、黒鍵と白鍵の鍵盤全体に2本の長い定規を当てて一気に全部を押してみればすぐに判明する。この一気押しを毎秒数回繰り返すというのが楽器業界で定番の「いじわるテスト」であり、そこからMIDI出力された情報を受けた電子楽器が鳴り続けるなど暴走すれば商品にならないので、各社とも必ず行っている筈である。

### 4. ArduinoとサウンドシールドとMozzi

Arduino[15]の世界では、もはや世界的な共通規格となったShield基板としてMusic/Soundというキーワードで検索してみると、SparkFun[16]、Adafruit[17]、SeeedStudio[18]、などの優れたサウンドシールドが廉価で提供されている。いずれも関連ライブラリを充実させつつ、MIDI音源としてだけでなくmp3プレイヤーとしても活用できるため、Arduino電子工作でとりあえずLEDを点灯させたデザイン学生は、次に「音を鳴らす」というステップでこれらのサウンドシールドに飛び付く。オープンソース(ブラックボックス)の効能として、プログラミングの初心者でも、自分で録音した音声をmp3化してマイクロSDカードに格納したり、MIDI音源のシーケンス自動再生演奏を実装し、センサやスイッチに反応してそのサウンドを鳴らすようなシステムが簡単にデザインできる時代となった。

MIDI自動演奏であれば問題ないが、Arduinoとサウンドシールドを用いたシステムを「MIDIを受信する音源」化するには注意点があり、普通にプログラミングするC言語ベースで31.25kbpsのシリアル入力をloop()でポーリングしたのでは、前節の「いじわるテスト」を持ち出すまでもなく、ArduinoはMIDI入力を受けきれずに簡単に暴走する[19]。MIDI送信というのはCPUが暇な時にすればいいので何でもないが、MIDI受信情報を1バイトでも受け損ねれば、例えばフルートやオルガンのような持続音系の楽器音のノートオフを失ったら「永遠に音が鳴り止まない」という楽器として致命的なトラブルが起きる。「作るサウンドエレクトロニクス」[5]のAKI-H8のMIDI受信では割り込みを使用してFIFOバッファに積む、という定番テクニック[20]を用いているので、絶対に受け損なわないのと対照的である。一部のArduino(例えばArduino Micro)は、ホストPCとのUSB通信に関する「MIDIUSB」というライブラリに対応しており、このライブラリでシステムを構築した

Arduino MicroをMacのUSBポートに差し込むと、Mac OSXはシステムが自動で「MIDI周辺機器」と認識して、「Audio MIDI Setup」の「MIDI Studio」にデバイスのアイコンが出現する。この環境ではMIDI受信はOS側できちんに行われていて受け損ねないというメリットがある[21]。最近の新しいArduinoでは、mbedと同様にMIDI受信割り込みのイベントハンドラを記述することで、同じようにMIDIを受け損なわれないようになっている。

このように「Arduinoでサウンド」は追加ハードウェアであるサウンドシールド任せだったのに対して、オープンソース・サウンド生成ライブラリ「Mozzi」[22]はArduino単体でサウンドを生成してしまう優れたものである。Mozziは最近の高性能Arduinoに限らずUnoやNanoにも対応しており、標準では16384Hzサンプリング・8ビット精度でアルゴリズムミックにオーディオ生成処理しつつ、PWM出力ピンを直接スピーカに接続するだけでモノラル音響を出力してしまう。オーディオrateは内部的に割り込みで生成しつつ、PureDataやSuperColliderでお馴染みのより低速なControl rateは可変になっている。「basic audio toolkit: oscillators, samples, lines, envelopes, scheduling, filtering」という品揃えもPureDataを彷彿とさせ、wavetableも活用できる。全てはオープンソースであり、解説ページ[24]を見ると、図3のようにきちんとしたデジタルオーディオ処理をArduinoに実装していることがわかる。解説ページ[24]では、「浮動小数点変数を使わず符合なし整数を推奨」とか「乗算・除算を行わずにビットシフト演算を活用せよ」など、非力なArduinoで最大限のパフォーマンスを得るための涙ぐましい努力が滲み出てきており、まさに筆者が前世紀にAKI-H8とディスクリート回路で実現したシステム[9]と通ずる世界となっている。

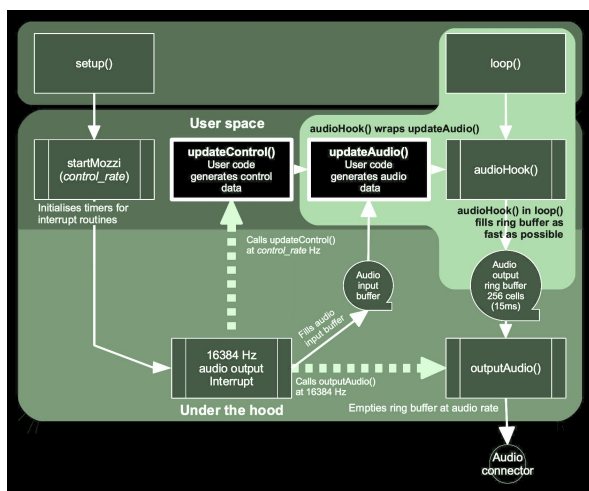


図3 Mozziの内部動作.

YAHOO.COMで「Mozzi」を検索したところ、実際にMozziを活用したシステムの製作記事など6件[25-30]が見つかったが、その最初の「Using Mozzi Library With 5 Potentiometers」[25]を作ってみたところ、何の調整もなく簡単に素晴らしいFMサウンドをぐりぐりと生成できるシステムが1時間ほどで完成してしまった[31]。サウンドシールドを用いた場合にはブラックボックスの彼方で音響生成

している印象があるが、追加ハードウェアなしにArduinoプログラミングだけでこのサウンド生成を実現できるMozziには、音楽情報科学の教材としての大きな可能性も感じている。そして6件の最後の「Arduino Organ」[30]は実際に製作されたシステムではなくて構想/サンプルスケッチのみのペーパーマシンのらしいのだが、これが本稿を執筆する契機となった。

## 5. 「Arduino Organ」プロジェクト

Mozziを活用したプロジェクトとしてGitHubに紹介されている「Arduino Organ」[30]は、60年代のHammondサウンドを実現するためにドローバーとトーンホイールの概念にこだわった特殊なシステムである。Mozziは柔軟に多様なサウンドを生成できる音源であるもののモノフォニック(単音)であり、簡単にはとてもオルガンにはならない。このシステム(Combo organ with 12 arduinos)はその壁を超えるために、Mozziの走るArduinoをなんと12個使うという荒技に出た。12個とは黒鍵と白鍵のクロマであり、最高音域の12音を次々に分周して全てのピッチを生成するという、遠い昔のトップオクターブ方式の電子楽器[4-5]にインスパイアされたい。Mozziが内部的に定義できる波形テーブルにはサイン波形を定義し、4オクターブの鍵盤を想定して7オクターブの波形をそれぞれのクロマに対応したものだけ用意することで、12個のArduinoによって全体としては84枚のトーンホイールによるサウンド生成を実現しようとしている。

ここで重要なのが、本稿第3節で紹介した、パイプオルガンと通じるHammondオルガンの音色生成のためのドローバーであり、5本のドローバーに相当するポテンションメーターの電圧は12個のArduinoの全てに共通に供給され、その他にビブラートとチューニングのポテンションメーターも持っている。このシステムがペーパーマシンであるという理由は、12個のArduinoは音源だけで精一杯であり、演奏情報(MIDI入力あるいは鍵盤演奏入力)に相当する「note-on/note-off information」は、それぞれ共通の12ポートのデジタル入力として供給される、という部分にある。音楽的なポイントは「ドローバーシステムは異なったクロマを積み付けて加算する」ためのものであるという点で、例えばある1個のMozziが走るArduinoの担当クロマを「C」とした場合、このArduinoは16feet・8feet・4feet・2feet・1feetの「C」音に対して反応するだけでなく、演奏情報として他のクロマに対応して以下のようなドローバーの積み付け演算も行う。Cが完全5度上である「F」音の16/3feet・8/3feet・4/3feet、そしてCが長3度上である「Ab」音の8/5feetについても、クロマ「C」のArduinoがその成分を担当する、というのが「Arduino Organ」プロジェクトのアイデアの「肝」である。正しいピッチを生み出すための周波数計算には浮動小数点による高精度の計算が必要となるが、個々のArduino(Mozzi)はそのクロマに対応するピッチを固定で持っているので、演奏情報の入力やドローバーが操作されて音色変化する時には、低速のControl rateで整数倍の単なる積み付け計算だけしておけば、あとはMozzi

が裏で高速オーディオrateによってサウンド出力してくれる。

このシステムを本当に実現するためには、外部MIDI入力、あるいは鍵盤演奏に対して、12個のMozzi音源Arduinoに共通の12ポートのデジタル信号を生成供給するための「コントローラ」として、さらにもう1個のArduinoないし他のマイコンが必要になる。これは筆者が2008年に製作し発表公開したインスタレーション作品「電子十二影坊 (Dodeca Propeller)」[32]と同じ発想である。こちらはArduinoではなくParallax社の8CPU内蔵・並列処理プロセッサ「Propeller」を使って、27インチテレビを12画面並べた大型モニタのそれぞれにリアルタイム生成CGを表示するというシステム[33]であり、来場者からの働きかけをセンシングしてMIDI化した情報を受けるマスターPropellerは、12画面をそれぞれ自分のプログラムによってライブCG生成している12個のスレーブPropellerに対して、共通のイベント情報としてパラメータの変化を伝達した。たまたま同じ「12マイコン共同作業」であるが、1個で無理ならたくさん並べる、というのはいつの時代にも共通するのかもしれない。

## 6. アイデアの拡張と発展の可能性

本格的な「楽器」でなくても、電子工作の応用としてArduinoを活用してインタラクティブに何かサウンドを出す「音具」をデザインしたい、という要請は大きい。ここに、MIDI音源など画一的なArduinoサウンドシールドを使うのではなく、柔軟な音響を生成できるシンプルなMozziを活用するという新しい選択肢の意義がある。安価なオルゴールは白鍵に相当するDiatonic Scaleの7音しか持たない、あるいは世界共通で「どんなにランダムに選んでも音楽的に妥当に聞こえる」Pentatonic Scaleの5音だけでいい、という発想をここでの「Arduino Organ」に適用すると、Mozziの走るArduinoを5個だけ使った「音具」という新しいアイデアが浮上する。それぞれのArduinoはPentatonic Scaleに相当した、ヨナ抜きCDEGA音、あるいは「猫ふんじゃった」の黒鍵5音、だけに限定してピッチを生成するが、Hammondサウンドのドローバーを意識すると、全てではないものの純正な完全5度/長3度に対応する成分まで鳴らすことで、全体として豊かな響きの実現を可能にするかもしれない。

発想を変えれば、あらかじめHammondサウンドをシミュレートした「美味しい」波形をwavetableにセットすれば、Mozziの走るArduinoを1個だけでも単音とは思えない豊かなサウンドを作れる可能性もある。固定的な波形読み出し再生では飽きるのだが、なんせMozziには時間的変化するフィルタやディレイ/リバーブまでを組み上げるaudio toolkitが提供されているのである。ちょうどKorgのLittleBitsでアナログ部品としてシンセサイザを構成したように、Arduinoプログラミングの中で楽音合成アルゴリズムを試行錯誤するというのは、システムデザイン教育としても、音楽情報科学教育としても、あるいは音楽学そのものの教育としても意義あるもののように思われる。

## 7. おわりに

分かる人だけに分かるタイトルによってちょっと刺激的な話題提供を目指した本稿は、音楽情報科学する方々に「温故知新」の機会を提供するためのものである。PCどころかスマホやタブレットにちょっとしたプログラムを走らせたならサウンドなんて何でも鳴るじゃない、という最近の人々にこそ、コンピュータがとても非力だった時代の技術的チャレンジについても知って欲しい。NASAの最新火星探査機PerseveranceのCPUはPowerPC750で、23年前のiMac G3と同じものであるという。最新の高性能CPUではとても耐えられない過酷な宇宙環境で動作する信頼性がその理由だが、リセットボタンすら押せない遠隔操作においては、システムがシンプルである事こそ重要なのである。昔の技術だからといって顧みないのでなく、しっかり理解した上で「その先」に突き進んで行ってもらいたい。

## 参考文献URL

1. [https://www.suzuki-music.co.jp/product\\_category/hammond/](https://www.suzuki-music.co.jp/product_category/hammond/)
2. <https://ja.wikipedia.org/wiki/ Hammondオルガン>
3. <https://www.suzuki-music.co.jp/products/65991/>
4. [https://nagasm.org/ASL/books/C\\_sound.pdf](https://nagasm.org/ASL/books/C_sound.pdf)
5. <https://nagasm.org/ASL/mse/>
6. <https://nagasm.org/ASL/mse/fig5-12a.gif>
7. <https://nagasm.org/ASL/mse/fig5-12b.gif>
8. <https://nagasm.org/ASL/mse/fig5-12c.gif>
9. [https://nagasm.org/ASL/Max7\\_part2\\_2/fig2/006.jpg](https://nagasm.org/ASL/Max7_part2_2/fig2/006.jpg)
10. [https://nagasm.org/ASL/Max7\\_part2\\_2/fig2/010.jpg](https://nagasm.org/ASL/Max7_part2_2/fig2/010.jpg)
11. <https://ja.wikipedia.org/wiki/オルガン>
12. [https://ja.wikipedia.org/wiki/ストップ\\_\(オルガン\)](https://ja.wikipedia.org/wiki/ストップ_(オルガン))
13. <https://nagasm.org/ASL/temper/>
14. [https://nagasm.org/ASL/Sketch12/fig1/Hammond\\_46sec.mp33](https://nagasm.org/ASL/Sketch12/fig1/Hammond_46sec.mp33)
15. <https://www.arduino.cc/>
16. <https://www.sparkfun.com/products/12660>
17. <https://www.adafruit.com/product/175>
18. [https://wiki.seeedstudio.com/Music\\_Shield\\_V2.2/](https://wiki.seeedstudio.com/Music_Shield_V2.2/)
19. <https://nagasm.org/ASL/Arduino/>
20. [https://nagasm.org/ASL/paper/SIGMUS201508\\_1.pdf](https://nagasm.org/ASL/paper/SIGMUS201508_1.pdf)
21. <https://www.youtube.com/watch?v=cM59p2CLu00>
22. <https://sensorium.github.io/Mozzi/>
23. <https://sensorium.github.io/Mozzi/learn/under-the-hood/>
24. <https://sensorium.github.io/Mozzi/learn/hints/>
25. <https://www.instructables.com/Using-Mozzi-Library-with-5-potentiometers/>
26. <https://www.instructables.com/Arduino-Based-Synth-With-Mozzi-Library/>
27. <https://awesomeopensource.com/project/tfry-git/almost.serious.synth>
28. <https://awesomeopensource.com/project/mgaldardini/nuncha>
29. <https://awesomeopensource.com/project/Tinwelint78/eBaghet>
30. <https://awesomeopensource.com/project/len/arduino-organ>
31. <https://www.youtube.com/watch?v=5wUFoPzmQh8>
32. <https://nagasm.org/ASL/12Propeller/>
33. <https://nagasm.org/ASL/12Propeller/circuit.GIF>