

ハンズオン電子楽器ソフトウェアの開発

熊谷 武洋

Development of The Hands-on Electronic Musical Instrument Software

KUMAGAI Takehiro

(Received September 25, 2015)

Key words: MIDI, Computer Music, DTM, Hands-on

1. はじめに

電子楽器は、アコースティック楽器とは異なり物理的な制約を受けない。音の発振や音量については電子的に制御したり電氣的に増幅が可能であるために、発音原理と形態とが物理的な対応をしていなくともよい(図1)。

電子楽器のインターフェイスは基本的にスイッチである。ボタンを押すことによってピアノ音を出すことが可能であり、電子ギターを爪弾いてオーケストラヒットを奏することも可能である。仕様によっては物理的なスイッチすら必要ではなく、スイッチが押された信号をプログラムによって生成すれば、人間がスイッチに触れることなく電子楽器から発音され、演奏が可能である。このように技術的には、あらゆる音色、演奏形態を採りうることができる(図2)。

しかし、目的に応じた合理的かつ適切な技術の援用や適用というものがある。電子楽器は演奏することなく手ぶらで楽器として機能できる、ということだけが利点や特徴として挙げられるべきではない。むしろ、新しい演奏スタイルや楽曲様式を見出すことこそが、電子楽器とその制御技術が発達した今日における新たな課題と言えよう。

よって、本研究室では、既存楽器を用いた従来型演奏形態の代替手段としてではなく、新しい可能性として電子楽器とその制御技術の特性を活かしたシステムを考案、試作した。

スマートフォンやタブレットPCをはじめとする携帯情報端末の普及にともない、画面上で制御する音楽アプリが増加してきた。しかしこれらのインターフェイスは、画面上に現れる鍵盤や弦を模した画面や、アイコンやシンボルによる抽象的な画面によってデザインされている。いずれも仮想画面であることに変わりはないので、楽器としての体感性は得られない。

そこで、本論では、物理的なフィードバックのあるインターフェイスをデザインし、ハンズオン電子楽器として、体感的要素をインターフェイスに繰り込んでみた。そして、想定利用者を従来楽器の初学者ではなく、奏楽に関心はあるものの、従来楽器演奏に不慣れな者、奏楽スタイルに関して先入観の少ない児童・生徒を対象と定め、要件と仕様を策定、本研究室にて試作システムを開発した。

2. 着想と仕様

そもそも楽曲の基本は音の高低(音高)とその長短(音価)である。当然のことながら、それ以外の音響的要素や表現上の複雑な要素も多く含まれているが、最低限この二つの要素があ

れば、音楽的完成度は別として曲を奏でることはできる。そして、音高と音価を身体動作で直接的に操作する人、つまり演奏者の思惑通りに制御するための入力装置として完成されているのは、長い歴史を持つ、いわゆるアコースティック楽器である。

このことから、試作システム的设计においては、音高と音価をどのように制御するか、あるいは、その信号をどのように生成するかという点に着目して着想した。そこでまず、これらの要件を間接的な体感動作と直接的な体感動作の二種に区分し、それぞれのアプローチを探ってみた。

前者については、その体感動作が演奏情報に直結しないが、その行為は反映するものとして捉え、そのインターフェイスとして、玩具のレゴブロックを選定した。

レゴブロックというなじみ深いアイテムを用いて、音高と音価の情報を生成、取得する仕様である。レゴブロックの盤面に、各色のレゴブロックを上下に配置、その配置の位置と色から、情報を取得、リマップするのである。

後者については、体感動作自体が、演奏情報に直結するが、反映効果を反転させるものとして捉え、そのインターフェイスとしてダイナミックレンジの広いセンサーを搭載するドラムマシンを選定した。

ダイナミックレンジの広いセンサーを搭載するドラムマシンの音量を標本化し、それを離散化し、音量が大きいと音高が高くなり、小さくなると下がるという仕様である。感情の起伏と音階の高低が対応・呼応していることは経験的にも既知のことであるが、この現象を積極的に活用するという意図である。

高揚し、感情が高ぶり、強く打つと音が高くなり弱いと低いというこの仕組みは言葉で表現すると単純ではあるが、従来楽器では物理的に困難な発音方法である。電子的に制御が可能であるからこそ実現できる方法と言えよう。厳密には、高次倍音の発生などの現象により、強く打つと音が高く聞こえるという近似した現象は音響学的にはあり得るが、電子的に制御し、情報をリマップ、エディット可能なので、強弱のレンジは随意に調整可能であり、またそのタイミングや発音時間も打楽器のそれとは異なり、持続時間は物理的な力に依存しないので、楽器としての可能性は幅広いと言える。

前者をブロック玩具システム、後者を太鼓システムと呼称し、仕様内容と処理の仕組みついて次に詳述する。



図1 古典的電子楽器テルミンの復刻器

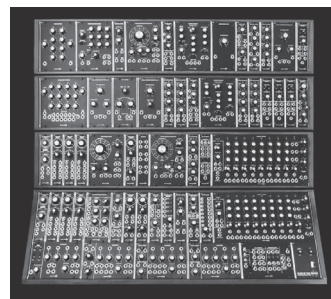


図2 Club of the kno社SYSTEM35
拡張モジュール群によるセンセ

3-2. 画像認識処理系統

入力画像信号から特定色を検出するために、jit.findboundsオブジェクトを用いた。jit.findboundsオブジェクトでは色の数値情報を入力することで特定対象とする色の指定を行うことが出来る。特定対象色の指定については、色の各要素について最大値と最小値を設定する事で指定を行える。そして最大と最小の差分を加減するによって、誤差を最小限に抑えて色の検出を行うことが出来る。

このようにして特定対象色を指定して、入力画像信号をjit.findboundsオブジェクトに入力すると、指定した色を含む部分が矩形領域で抽出される。次いでjit.findboundsオブジェクトから矩形領域の左上の座標がleftとtopの2つの値で出力され、右下の座標の情報rightとbottomの2つの値で出力される。また、入力画像信号内のブロックの中心点を求めることでブロック自体の移動を認識することになるため、数式の値を求めるexprオブジェクトを使用してjit.findboundsオブジェクトが出力する矩形領域の中心点を求めている。また、jit.brcosaオブジェクトを使用してbrightnessとContrastの値を上げることにより、レゴブロックを配置する白い盤面の色情報を飽和させることができる。いわゆる白飛びである。同様にSaturationの値を上げることによってレゴブロックの色の彩度を高くできる。これらのポスト処理を加えることで白い盤面とレゴブロックの境界領域を明確にし、色と形の認識の精度向上を実現できた。(図6、図7)

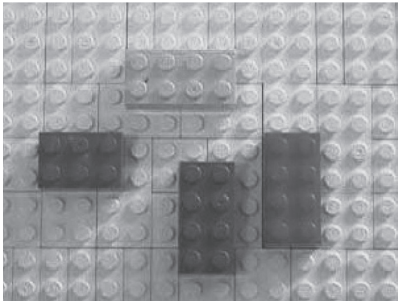


図6 brcosaによるポスト処理前

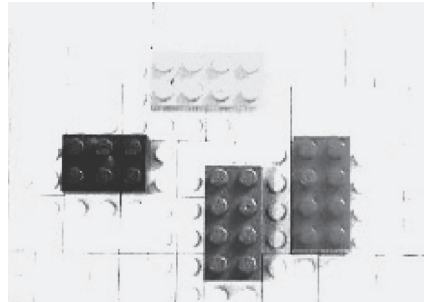


図7 brcosaによるポスト処理後

3-2. 音声信号処理系統

次に音声信号処理過程について述べる。まず、前段処理として、Max/MSP側で、フェーダやノブの情報をこれらのコントロール・チェンジ・メッセージとして取得する。コントロール・チェンジ・メッセージを生成、送信することによってさらに音楽的な表現を高めるためである。

Max/MSP内にコントロール・チェンジ・メッセージを取得するために、コントロール・チェンジ・メッセージの取得のための専用オブジェクトであるctlinオブジェクトを使用した。ctlinオブジェクトでは、コントロール・チェンジ・メッセージのコントロール番号とコントロール値の2つの値を独立して取得が可能であり、フェーダならびにノブの固有値であるコントロール番号をソフトウェア音源側の操作したい部位のパラメータの固有値に、コントロール値もパラメータの値に変換し直し、これらの値をともにソフトウェア音源側に送信することで、MIDIコントローラのフェーダ、ノブを用いてソフトウェア音源を操作することが可能となる。

次段階として、盤面のX軸とY軸から音高、音価、発音のタイミングなどをコントロールする機構を加えた。(図8)。

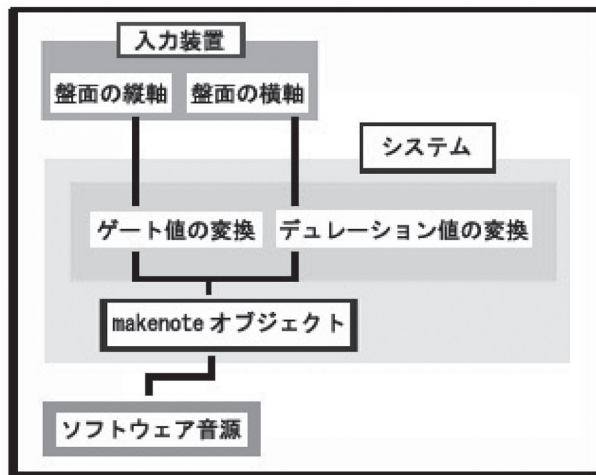


図8 音声信号処理フロー

●発音タイミング手動化による音高と音価値の1系統制御

発音タイミングについては、電子楽器であるため、自在に制御できるが、楽器であるという性格を考慮し、複数のフェーダとノブにコントローラ上に並んでいる順で左から右へコントロール番号にドレミファソラシと音高を当てはめ、次いでコントロール値に音量のパラメータを当てはめ、個々の操作子から発音し操作を行う処理過程を設計した。なおスケールは変更可能である。

●発音タイミング自動化による音高と音価値の独立2系統制御

電子楽器としての特性をさらに高めるために、発音タイミングを操作子で行わず自動で処理する機能を加えた。これは、Max/MSP内で独自に信号を生成して、この信号を発音の命令に使用するものである。信号は連続的に発生させることが可能であり、今回は、BPMのタイミングに合わせて信号生成が可能なtempoという専用オブジェクトを発音の制御に用いた。

音高に関しては、低い音から高い音に「上下する」ものとして捉えることが理解しやすいと考えられるためレゴブロックを配置する盤面の縦軸をマッピングし、レゴブロックを上下させて操作し、音価は、盤面の横軸にマッピングした。

●ゲートのパラメータの値の変換

入力されたコントロール値をゲートと呼ばれる発音のタイミングは、初期値を8分音符に設定してあるが、随意に変更可能である。

●デュレーションのパラメータの変換

発音タイミングの以外に、発音のコントロールを担うパラメータとして、デュレーションとタイムを加えた。このパラメータは音の長さを調節するためのパラメータであり、指定した時間の長さ分、音を発音させる命令を音源に対して行う。音源を発音させるMIDIメッセージであるノート・オンメッセージが与えられてから、音を止めるMIDIメッセージであるノート・オフメッセージが与えられるまでの時間の長さを指す。

●スケールマッピング

音高操作のパラメータとしてノートナンバーの値を利用している。このノートナンバーのパラメータの操作にはレゴブロックを置く盤面の縦軸を使用しており、レゴブロックの配置される位置によって一定間隔で発音される音の音高を可変させる内容になっている。

●ソフトウェア音源への接続過程

前述した入力装置からのパラメータ処理過程の全てを統合し、ソフトウェア音源に入力するMIDIデータを生成するためにmakenoteオブジェクトを用いた。makenoteオブジェクトは、ノート・ナンバー、ベロシティ、デュレーションの3種のMIDI信号を受け取り、ノート・オン、ノート・オフを生成するオブジェクトである。この処理の後、ソフトウェア音源へのMIDIデータの受け渡しに必要なヘッダー情報を付加し、MIDI音源で演奏することが可能となる。

3-3. 演奏スタイル

レゴ盤面に任意の各色レゴブロックを配置すると、MIDI音源の音が奏でられる。

感覚をつかむための多少の試行は必要であるが、操作自体は簡便であるため、特に事前練習などは必要がない

レゴブロックを上下させる動作と、レゴブロックを左右に動かす動作を数回行うことにより、音源から持続音が発音されるので、基本テンポを会得できれば何らかの演奏を展開することが可能である（図9）。

演奏フレーズの制御手段として、iOS上で稼働するMIDIモニタアプリを使用した。（図10）MIDI信号を生成、送出できれば、フィジカルサーフェスコントローラの方がハンズオンという観点からは操作上の親和性が高いが、インターフェイスを簡便にカスタマイズできるので、MIDI信号制御という二次処理には当該アプリを用いた。仕様上、意図することよりのレゴを置きながら、予想外の音楽的展開を楽しむことになるが、ある程度、処理の規則性を会得したら、意図的なフレージングは可能となる。



図9 ブロック玩具システム演奏スタイル



図10 iOS上で稼働するMIDIモニタアプリ

4. 太鼓システム概要

本システムは、信号入力としてWAVEDRUMからの情報から振幅のみを抽出する処理、取得した情報をMIDI情報としてノートナンバーや周波数値に割り当てる処理、得られたMIDI情報を音源にマッピングする処理の3つの系統に区分されており、最終段においてこれらの3系統を多段的に組み合わせることで本システムを構築している（図11）。

次に各処理過程の機能と内容について述べる。

4-1. WAVEDRUMによる音量情報取得

ダイナミックレンジの広いセンサーを搭載するドラムマシンとしてKORG社 WAVEDRUMを使用した。（図12）

このWAVEDRUMは今回の研究で最も重要な情報ソースである音量を7bitのMIDI規格以上の精度で出力が可能である。これらの理由から、本研究ではWAVEDRUMを入力機器として選定した。

しかしながら、出力された音量情報は、離散化されておらず、そのままではデジタル情報としてMIDI信号に変換、音楽情報としてリマップができない。そこで開発環境としてcyclin'g'74社のオーディオ・ビジュアルプログラミング環境のソフトウェアMax/MSP（音響処理）を用いて、入力信号の処理を行った。

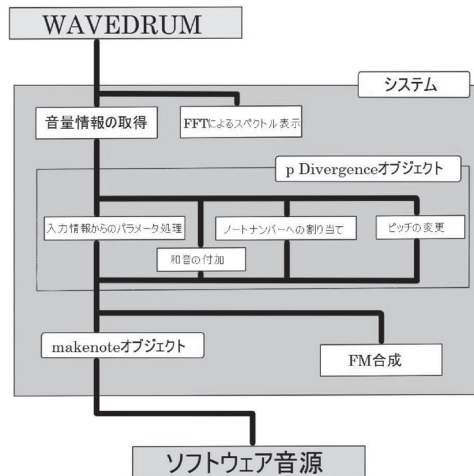


図11 太鼓システム全体の処理フロー



図12 KORG社 WAVEDRUM

4-2. 入力信号の取得処理

WAVEDRUMの音量情報をmeterオブジェクトにより、音量情報の平均を取得している。また同時にFFT、multisliderを用いて、スペクトル表示を行い、atodbオブジェクトを用いて振幅値をデシベル値に変化させてある。デシベルとは基準と比較してどの程度大きいかという表現の仕方であり、通常は量などの値やゲインなどの数値の比較のために用いる。

今回、音量情報を取得する方法として、meterオブジェクトを用いる方法と、FFTを用いる方法を試作した。その結果、取得できる情報量や幅に大差はなかったため、実際に操作感覚から、meterオブジェクトを用いる方法を採用した。そのため今回利用しているFFTampはスペクトル表示のみの利用となっておりFFTampが無くともこのソフト自体の動作は可能である。

スペクトル表示に関してはまず、WAVEDRUMからの音量情報をFFTampに送っている。このFFTampでは、fftin~ オブジェクトよりFFTを実行し、cartopor~ オブジェクトで直交座標を極座標に変換して振幅情報のみを抽出する。また、fftin~オブジェクト第3アウトレットから出力される現在の周波数ビンのインデックスも送っている。

poke~ オブジェクトは、record~オブジェクトとは異なり、第2インレットにサンプル・インデックスを受け取り、そのインデックスに第1インレットに受けとるサンプルの値を書きこんでいく。このサンプル・インデックスとして、fftin~ オブジェクトから出力される現在の周波数ビンのインデックスを利用している。

buffer~ オブジェクトの読み出しはpeek~ オブジェクトを用いて行っている。peek~ オブジェクトもサンプル・インデックスにより、そのサンプル値を読みだすが、入出力シグナルでなく実装である。uzi オブジェクトはbangメッセージを受け取ると、整数値の1から256までを出力し、これを利用してbuffer~オブジェクト最初の256個のサンプル表示にmultisliderオブジェクトを用いている。

4-3. 入力情報の加工処理

次に入力装置からのパラメータ処理過程について述べる。音高操作のパラメータとして、MIDI信号であるノートナンバーの値を利用している。このノートナンバーは振幅の大小により降順で割り当てた。これを昇順に反転することも可能であるが、音楽的な配慮から降順とした。

振幅情報はデシベル値に変化させ、-65から0までの値を取得、その情報を22分割（3オクターブ分）している。また、今回取得するデシベル値は一度の変化量が大きく、22分割以上は難しいと判断した。また、ここでは均等に22等分しておらず、大きな音（高い音）、小さい音（低い音）の幅を広く取っている。2倍大きな音を出そうと意図して、感覚的に2倍強くは叩けても、それが実際の2倍の音量にはならない。そこで感覚と実際の値に整合性を持たせるため、ノンリニアで調整を行った。

4-4. 入力情報の調整処理

WAVEDRUMから送られた情報を解析し、そのままノートナンバーを生成し、MIDI音源から発音させると、大変聞き苦しい音になる。これは、送られた情報が連続情報であり、一つの区切り音の生成ができないからである。また、WAVEDRUMの音を解析すると、高い音から徐々に低い音へと減衰することが確認できた。これを改善するために、送られた情報の音高に変化を与え、隣同士の音が発音されないように発音タイミングを分断した。さらに、送られてきた情報をGgateオブジェクトに送り、データが連続で送られない処理回路を作成した（図13）。

metoroオブジェクトとcounterオブジェクトにより、一定タイミングでGgateオブジェクトのスイッチを相互に切り替えている。これにより、連続した情報が送られないようにした(図14)。

また、randomオブジェクトを用い、一定のテンポでスイッチが切り替わることを防ぎ、予想外の要素を与えて音楽的に面白みを持たせるような処理を加えた(図15)。

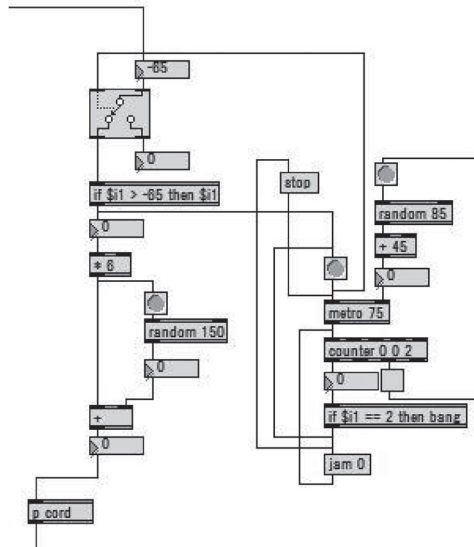


図13 連続的な音の発音を防ぐパッチ

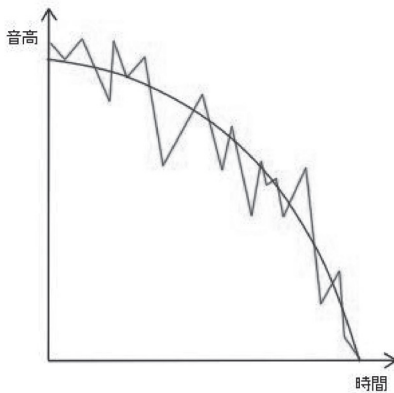


図14 スイッチによる音の切断

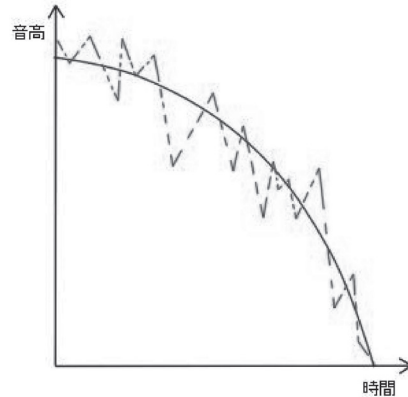


図15 randomを用いた音高の調整

これらの各処理回路を多段的に組み合わせることにより、全体の処理フローが完成した(図16)。

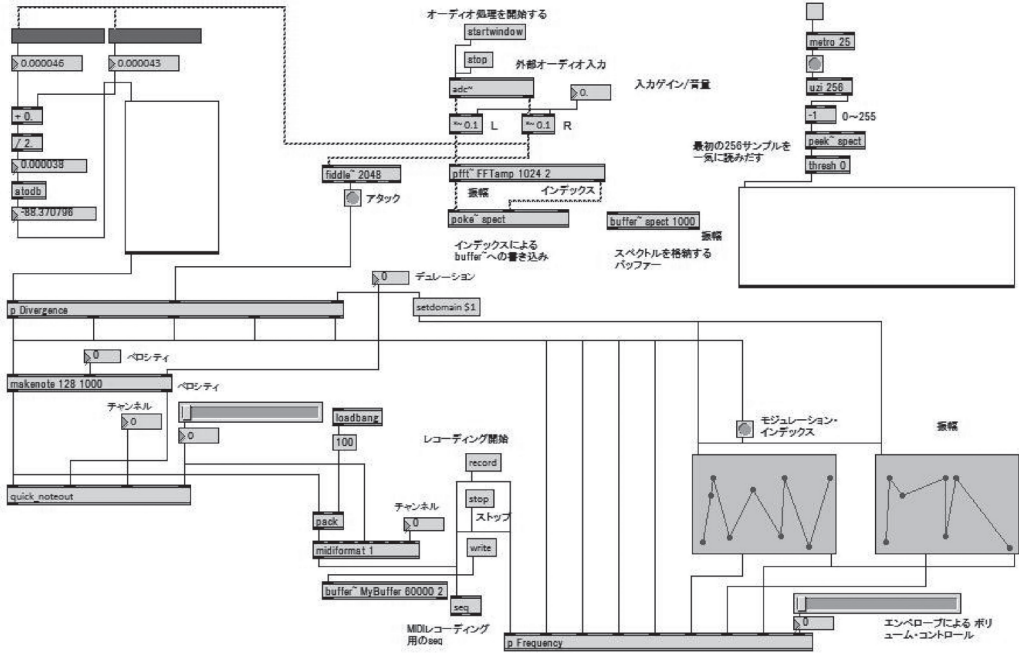


図16 太鼓システム全体処理フロー

4-5. 音声信号処理と演奏スタイル

ブロック玩具システム同様にMIDI信号に変換した後、コードやハーモニー機能を付加することに加え、太鼓システムでは、重複音の効果や、ユニゾン効果を得るためピッチの変更処理を加えた。ピッチは3オクターブ分の音階幅を設定した。このピッチの変更機能は、手動と自動で切り替えが可能となっている。自動切り替えの方が楽曲としては興味深い出力を期待できる。

このような付加機能を加えて、試作システムで実際に演奏を行ったところ、強い力で叩くと高い音、弱い力で叩くと低い音が発生され、感情の起伏に対応して音の高さの変化は実感できるという評価は得られた。しかし、いわゆる楽器演奏としての音楽的表現行為が成立するほどの体感は得られにくいという評価もあった。電子楽器ならではの表現という点では成功だが、使用者の主體的な自由感覚で演奏できる新しい電子楽器としては多くの改善の余地が残った(図17)。



図17 太鼓システム外観

5. 成果と今度の課題

今回は、新しい試みとして、ハンズオンという点に着目し、電子楽器ソフトウェアの試作を行った。

信号処理過程に未完成部分などが残るものの、通常は着想だけで終わるような仕様を一応試作として具体化させたということで次なる発展的展開が得られた。

ブロック玩具システムは、レゴブロック自体が馴染み深いアイテムということもあり、試奏の評価はおおむねよかったが、太鼓システムの方は、やり方は理解出来ても、慣れるまでに時間を要し、仮に慣れたとしても、それほど音楽的昂揚感は得られないという評価が多かった。今後の課題としては、実施検証を十分に行い、想定ユーザーのヒアリングやフィードバックを重ねてシステム自体の完成度を高めていきたいと思う。そして、当該システムから必然的に導出される音楽的表現様式を獲得し、完成度の高い即興演奏曲を創出したいと考えている。

謝辞

本研究を遂行するにあたり各システムのプログラム実装、支援いただいた後藤あゆみと井崎優の両名、および各システムの評価調査に協力いただいた表現情報処理コースの皆様には謝意を表します。

参考資料文献

- ・ノイマンピアノ（赤松正行+佐近田展康）、“2061:Maxオデッセイ”、リットーミュージック、2006
- ・ノイマンピアノ（赤松正行+佐近田展康）、“トランスMaxエクスプレス”、リットーミュージック、2001
- ・“THE Max Book”、カメオインタラクティブ
- ・高橋信之、“コンプリートMIDIブック”、リットーミュージック
- ・米本 実、“楽しい電子楽器—自作のススメ”、オーム社、2008
- ・Nicolas Collins、“Handmade Electronic Music —手作り電子回路から生まれる音と音楽”、オライリージャパン、2013
- ・中西 宣人、Arduinoではじめる手作り電子楽器—ブレッドボードを使った音の電子工作、工学社、2015