

# エレクトーン演奏演技における MIDI の活用

[https://github.com/wakmin-oxo/Electone\\_and\\_MIDI\\_document](https://github.com/wakmin-oxo/Electone_and_MIDI_document)

wakmin (@wakmin\_)

2023 年 12 月

## 目次

1	はじめに	4
1.1	免責	4
2	MIDI データの基礎知識	5
2.1	General MIDI と XG、EL ボイス	6
2.2	MIDI チャンネルメッセージ	6
2.2.1	ノートオン・オフ	6
2.2.2	チャンネルプレッシャー	7
2.2.3	ピッチベンドチェンジ	7
2.2.4	プログラムチェンジ (PC)	7
2.2.5	コントロールチェンジ (CC)	8
2.3	MIDI システムメッセージ	10
2.3.1	システムコモンメッセージ	10
2.3.2	システムリアルタイムメッセージ	10
2.3.3	システムエクスクルーシブメッセージ (sysEX)	10
2.4	スタンダード MIDI ファイル (SMF)	11
3	XG サポートの作り方	12
3.1	Domino の導入	12
3.1.1	音源定義ファイル	12
3.1.2	Domino の初期設定	13
3.1.2.1	エレクトーンとパソコンの接続	13
3.1.2.2	音源定義ファイルの導入	13
3.1.2.3	MIDI-OUT の設定	14
3.1.2.4	MIDI-IN の設定	14
3.1.2.5	その他の設定	14
3.2	XG サポートを作る	14
3.2.1	ファイルの作成	15
3.2.2	XG サポートファイル製作の下準備	16
3.2.2.1	バルクダンプの削除	16

3.2.2.2	Ch.16 のメッセージの削除	17
3.2.3	XG サポートデータの記述	17
3.2.3.1	XG モードへの変更	17
3.2.3.2	楽器指定 (プログラムチェンジ)	19
3.2.3.3	その他のチャンネルセットアップ	19
3.2.3.4	ステップ録音	20
3.2.3.5	ピアノロール	20
3.2.3.6	イベントグラフ	21
3.2.4	XG サポートデータの保存とソング情報の書き換え	21
3.2.4.1	EL ON の配置と End of Track の調節	21
3.2.4.2	XG サポートデータの保存	21
3.2.4.3	USB メモリへの転送と ELS_SONG.NAM の編集	22
3.2.4.4	EL-Explorer を使った XG サポートファイルの転送	24
3.3	XG 音源特有の仕様	24
3.3.1	システムエフェクトとインサクションエフェクト*	24
3.3.1.1	リバーブエフェクト	24
3.3.1.2	コーラスエフェクト	25
3.3.1.3	バリエーションエフェクト	25
3.3.2	メガボイス	26
4	スタイルファイルの作り方 (ドラム)	33
4.1	EL Data Analyzer	33
4.2	スタイルファイルの構造	34
4.3	メイン・アドドラムの打ち込み:MIDI セクションを作る	35
4.3.1	拍子の指定と Main A マーカーの挿入	36
4.3.2	システムセットアップ*	36
4.3.3	チャンネルセットアップ	37
4.3.4	セクションパターン小節の打ち込み	37
4.3.5	1 小節目と 2 小節目以降の違い	38
4.3.6	スタイルファイルの保存とエレクトーンでの読み込み	38
4.4	ドラムセットアップの解説*	39
4.4.1	ドラムセットアップ (NRPN)*	40
4.4.2	ドラムセットアップ (sysEX)*	41
5	スタイルファイルの作り方 (アカンパニメント) *	43
5.1	CasmEdit の導入	44
5.2	Ctab で設定できるパラメータ	44
5.2.1	入力チャンネル・出力チャンネル	44
5.2.2	Source Chord Root / Type (C.R. / C.T.)	45
5.2.3	NTR	45
5.2.4	NTT	46
5.2.5	Highest Key (H.K.)	47
5.2.6	Low Limit / High Limit (L.L. / H.L.)	47

5.2.7	RTR	47
5.2.8	Auto Start	48
5.2.9	Note Mute (NMute) / Chord Mute (CMute)	48
5.2.10	Editable Bit	48
5.3	アカンパニメントを含むスタイルファイルの作り方	48
5.3.1	アカンパニメントパートの MIDI セクションの打ち込み	48
5.3.1.1	アカンパニメントパートの MIDI セクションの注意点	48
5.3.2	CASM セクションの作成と編集	49
5.3.3	CASM セクションの保存とバックアップ	50
6	Arduino を用いた MIDI デバイス製作	51
6.1	Arduino の基礎	51
6.1.1	Arduino とは	51
6.1.2	Arduino ボード	51
6.1.3	Arduino 言語	52
6.1.4	Arduino 始めの一步 L チカ	52
6.1.5	Arduino とシリアル通信	53
6.2	MIDI ソフトウェア設計 (Arduino MIDI Library)	53
6.3	MIDI ハードウェア設計	54
6.3.1	MIDI OUT	54
6.3.2	MIDI IN	55
6.4	電子電気回路部品の紹介	57
6.5	フォン端子とその他のインターフェース	58
6.5.1	フォン端子	58
6.5.2	その他のインターフェース	58

# 1 はじめに

エレクトーンは1000を超える音 [1] を有し、1 台で 3(+2) パート<sup>\*1</sup>まで演奏できるため、本来的には合奏しないと演奏できない曲でも 1 人で演奏しきることができる。エレクトーン単体の性能の自己完結ぶりは、他の電子楽器よりも優れていると言っても過言ではない。

その一方で、エレクトーン単体で完結しないようなエディットは非常に敷居が高い。例えば、自動演奏補助として XG サポートが存在するが、XG ボイスリストは公式で公開されておらず、XG サポートを作るための GUI を備えた現代的なエディタはもはや存在しない。スタイルは他の電子楽器に搭載されているスタイルクリエーターを使わないとエディットできず、それも隠しパラメータが存在する。そもそもこれらの仕様に関しては知名度が低く、エディットができることすら知られていない上に、製作しようと志すことができたとしても、体系的にまとまっているテキストがほとんど存在しない。そのため、「知る人ぞ知る」エレクトーンの隠し機能として、現在まで埋もれていた。

そこで本稿では、読者が自力でオリジナルの自動演奏補助が作れるようにすることを目的に、これらエレクトーンの「隠し機能」を初学者にも分かりやすく解説する。MIDI に関する低級な話題はあまり紹介せずに、即演奏に応用できるような情報のみをセレクトしたつもりである。ぜひ自分のエレクトーンで試してみて、演奏表現の幅を広げてほしい。ただし、節によっては内容が難しくなったり、実用的でない情報も含まれたりしてしまったため、そのような節には「\*」を付けておいた。興味がある読者や、MIDI の取り扱いに慣れた読者でなければ飛ばして構わない。

また本稿の最後には、エレクトーンの演奏演技の幅を広げる手段として、Arduino を用いたデバイス開発のすすめを書いた。自動演奏補助とは異なるが、sysEX をリアルタイム演奏に活用したり、エレクトーンのインターフェースを拡張したりできる。ぜひチャレンジしてほしい。

なお、本稿は予告なくアップデートされる。最新版は [https://github.com/wakmin-oxo/Electone\\_and\\_MIDI\\_document](https://github.com/wakmin-oxo/Electone_and_MIDI_document) で配布するため、時々確認してほしい。最新の情報をリリースするために、**この pdf ファイルの再配布はせず**に、上記 url を教える形でこのドキュメントを広めることをお願いする。また、内容の誤りを発見した場合、配布している github で Issue を submit するか、twitter([@wakmin\\_](https://twitter.com/wakmin_)) などで報告してほしい。

## 1.1 免責

できる限り正確な情報をまとめたつもりであるが、独自研究が多く含まれているため、**本稿の内容を実行してどのような問題が生じたとしても、私はいかなる責任も取らない**。本稿を読んで実行したことに対して、読者は自己責任で実行することを認めたものとする。

---

<sup>\*1</sup> 鍵盤 3 本 + ドラム 2 本。

## 2 MIDI データの基礎知識

MIDI とは Musical Instrument Digital Interface の頭字語で、異なる機種・異なるメーカー間であっても「演奏情報」をコミュニケーションするために策定された規格である [3]。実際の音色は各楽器で合成されるが、通信されるのは「何番目の音を」「どれくらいの強さで」「弾き始める」などという情報である。このような演奏情報を、**MIDI チャンネルメッセージ**と呼ぶ。また、演奏情報以外の情報を、**MIDI システムメッセージ**と呼ぶ。これら 2 つを合わせて、**MIDI メッセージ**と呼ぶ (図 1 を参照)。

本稿では、MIDI メッセージの具体的な表現 (バイナリ表現) には立ち入らず、「どのような情報が流れているか」に焦点を絞って説明する\*2。

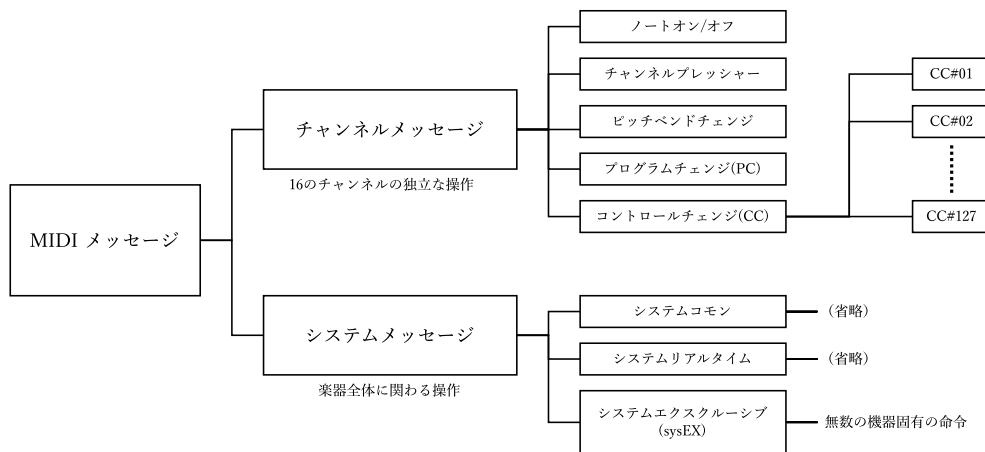


図 1: MIDI メッセージの種類。

我々がエレクトーンの 3 段の鍵盤を触ったり、パネルを操作したりすると、それらの操作 1 つ 1 つに対応する MIDI メッセージが音源部に送信されて、実際の音に反映される。単に演奏するだけであればこのことを意識する必要はないが、自動伴奏や自動演奏を製作しようとすれば、これら演奏情報を (実際のリアルタイムな演奏無しで) 音源部に送信しなければならないため、MIDI メッセージの理解が不可欠となる。そこで**この章では、この後の内容を読むのに最低限必要な知識を紹介する。**

この章の参考書

- MIDI 入門 [2]  
MIDI を知らない人は、**まずこれから読むと良い**。ヤマハの初心者向けテキストである。
- MIDI1.0 規格書 [3]  
規格書である。量が多いので全部読もうとするのは勧めない。辞書がわりに使うのが良い。
- ELS-02/ELS-02C/ELS-02X MIDI リファレンス [4]  
ELS-02 シリーズが対応している MIDI メッセージの一覧。エレクトーンに対してどのような命令を MIDI で行えるのかの参考になる。

\*2 本稿では Arduino MIDI Library や Domino を用いて MIDI の処理をするため、具体的な表現はほぼ必要にならない (それらに上手くパッケージ化されている)。ただし、MIDI システムエクスクルーシブメッセージを Arduino で制御しようとしたときは値を直接打ち込む必要がある。

## 2.1 General MIDI と XG、EL ボイス

MIDI はもともと演奏情報を取り扱う規格であって、受信側でどのような音を発生させるかはメーカーごとに異なっていた。これをある程度統一するために現れた規格が **General MIDI (GM)** であり、プログラムチェンジ（後述）やドラムマップ<sup>\*3</sup>の統一などがなされた [7][8]。ヤマハは GM を拡張して **XG**<sup>\*4</sup>という独自規格を作り、GM で作られたデータを再生できるようにしつつ、表現能力を高めた。

さて、エレクトーンは XG 規格楽器である。しかし、エレクトーン鍵盤で演奏する音色が XG によるものかという点、（大部分が）そうではない。エレクトーンには、自動演奏のためのボイス（XG ボイス）と、鍵盤で演奏するためのボイス（EL ボイス）とが存在する<sup>\*5</sup>。XG ボイスはスタイルと XG サポートでしか現れない<sup>\*6</sup>。本稿では XG ボイスを主に紹介する。

## 2.2 MIDI チャンネルメッセージ

MIDI チャンネルメッセージは、演奏に関する情報と、音色の設定の情報である。よく使用されるメッセージは、**ノートオン・オフ**、**チャンネルプレッシャー**、**ピッチベンドチェンジ**、**プログラムチェンジ**、**コントロールチェンジ**である。

MIDI チャンネルメッセージは 16 の独立した**チャンネル**を持っており、それぞれのチャンネルでバラバラに演奏できるようになっている。MIDI チャンネルメッセージは全て、1 から 16 までの<sup>\*7</sup>チャンネルを指定しながら送信する。エレクトーンにおいて、MIDI チャンネルはデフォルト<sup>\*9</sup>で図 2 のように鍵盤ごとに割り当てられている。図 2 を見ればわかるように、Ch.5 から Ch.14 までの MIDI チャンネルはエレクトーン側に対応するコントローラが存在していない。

### 2.2.1 ノートオン・オフ

**ノートオン**とは、ノート<sup>\*10</sup>を演奏開始する命令である。キーが押されたときに送信される。押されたキーの位置は**ノートナンバー**という値で指定され、ノートナンバーは中央 C=60 と定められている。また、演奏の強さは**ベロシティ**という（0 から 127 を取る）値で指定される。ベロシティは、エレクトーンではイニシャルタッチの強さに関係がある。

**ノートオフ**とは、ノートの演奏を終了する命令である。キーが離されたときに送信される。なお、ノートオ

<sup>\*3</sup> ドラムマップとは、鍵盤上にドラムパーカッションを並べるとき、どのキーがどのパーカッションに対応しているかの割り当て。C2 にキック、D2 にスネアなど、GM によって決まっている。

<sup>\*4</sup> XG は、DTM の開祖・ローランドの GS 規格に対抗するために打ち立てられた [5]DTM 音源の規格であった。1996 年にローランドが SC-88Pro という伝説的な機体を出したのもこの頃である。お互いに音源規格の覇権を争っていたが、両社とも 2001 年に GM Level 2 に合流することで戦争は収束に向かい [6]、「XG 音源」を前面に打ち出す必要がなくなった。DTM 用音源がその後の時代でプラグイン音源に置き換わったため、XG は 2021 年現在、表舞台から姿を消した。EL3 桁代の頃は DTM 用音源との互換性のために、XG の基本機体である MU50 相当の音源を搭載していたが、それより先の ELS-01、02 シリーズになると、もはや DTM 用音源のための XG とは違った独自の方向へと進化したようである [8-78]（スタイル入門講座へのリンク）

<sup>\*5</sup> XG ボイス、EL ボイスという呼び方は本稿の便宜であるため、公式的な呼称でないことに注意してほしい。

<sup>\*6</sup> キーボードパーカッションでも扱うが、この場合 XG パーカッションを例外的に鍵盤で使っているものと理解できる。

<sup>\*7</sup> 余談：チャンネルは内部的には 0 (0H<sup>\*8</sup>) から 15 (FH) の値である。sysEX などチャンネル指定するとき、Ch.9 を指定しようとしてうっかり 09H を送信してしまうことがよくある（実際には 08H を送信しなくてはならない）。

<sup>\*8</sup> 数字 H という書き方は、書かれた数字が 16 進数表現であることを示す。16 進数とは、1 桁に 0 から 15 までの 16 通りの数字を書くことができる数の表現であり、10 は AH、11 は BH、...、15 は FH と書く。16 から繰り上がりが生じ、16 は 10H となる。16 進数表現から 10 進数表現に直すには、(上の位)×16+(下の位)を計算する。例えば、4AH = 4 × 16 + 10 = 74 のようにする。MIDI メッセージは 16 進数で 2 桁の数字を用いて説明されることが多い。

<sup>\*9</sup> 受信チャンネルは図 2 で固定（EL モードの場合）だが、送信チャンネルは STAGEA の本体設定で変更できる。変更する場合、UTILITY ボタンを押して、画面上部のタブメニューから MIDI を選択すれば良い。

<sup>\*10</sup> ノートとは、音符の演奏情報のことである。鍵盤のキーひとつづつに対応する。

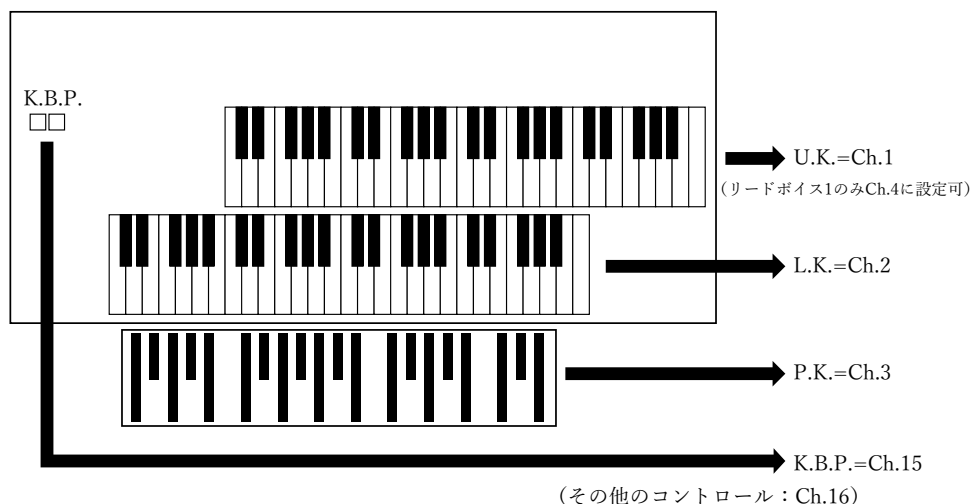


図 2: エレクトーンのデフォルト MIDI チャンネル設定。図は EL モードの場合。XG モードの場合は後述。

ンのペロシティ 0 はノートオフと同じとして解釈される<sup>\*11</sup>。

ノートオンからノートオフまでの時間のことを、**ゲートタイム**（またはデュレーション）と呼ぶ。

## 2.2.2 チャンネルプレッシャー

チャンネルプレッシャーは、そのチャンネルのアフタータッチの強さを指定する命令である<sup>\*12</sup>。0 から 127 の値をとる。EL ボイスのみ音色変化が起こり、XG ボイスでは無視される。

## 2.2.3 ピッチベンドチェンジ

ピッチベンドチェンジは、そのチャンネルのサウンドのピッチを変化させる命令である。エレクトーンの場合、**水平タッチ**の強さが送信される。16384 段階に量子化されており<sup>\*13</sup>、デフォルトで 0 で、-8192 から 8191 の値をとると表されることが多い。

## 2.2.4 プログラムチェンジ (PC)

プログラムチェンジは **PC** とも書かれ<sup>\*14</sup>、そのチャンネルの音色を変更する命令である。ほとんどの場合、後述の CC#00/32 と組み合わせて使う。ただし、**エレクトーンの場合、Ch.16 の PC=1 から PC=16 はそれぞれレジストレーションメモリーに対応しており**<sup>\*15</sup>、メモリーしたレジストを呼び出すことに使える。XG ボイスの変更に使わず、EL ボイスでは使用しない。プログラムチェンジ自体は 1 から 128 までの値<sup>\*16</sup>をとるが、CC#00/32 と組み合わせて使うため、膨大な種類がある。単に「プログラムチェンジ」と言った場合、

<sup>\*11</sup> MIDI の通信はデータの最初に命令の種類を書くのだが、命令の種類を省いた場合は前の命令と同じ命令であると解釈される（ランニングステータス）。発音を止めたいときは、ノートオンとして出力した方がランニングステータスのおかげでデータが圧縮できるため、通常はノートオフを出力せずに、ノートオンのペロシティ 0 を出力する。なお、一応ノートオフにもペロシティが存在し、離す強さを表せるが、私は対応しているハードウェアを見たことがない。

<sup>\*12</sup> ELS-02C において、アフタータッチは各鍵盤に 1 つずつ、計 3 つの感圧センサーで鍵盤への力を測定して処理している。このような、チャンネルごとのアフタータッチをチャンネルプレッシャーと呼ぶ。なお、キーごとにアフタータッチを認識する場合、ポリフォニックキープレッシャーという別の命令で処理されるが、ELS-02 シリーズには未搭載である。シンセサイザー MONTAGE M8x(2023) などにはポリフォニックアフタータッチという名前で搭載されている。

<sup>\*13</sup> 128 通りを表現できるデータバイトを 2 個組み合わせて使っているため、 $128 \times 128 = 16384$  通りを表現できる。

<sup>\*14</sup> PC とだけ書くと Personal Computer と混同するため、本稿ではなるべくプログラムチェンジと書く。

<sup>\*15</sup> レジストレーションメモリーは sysEX（後述）でも操作できる。

<sup>\*16</sup> プログラムチェンジの番号は、内部的には 0(0H) から 127(7FH) の値。



CC#00/32 と組み合わせられて表現される 1 つのボイスの番地を表すことが多い。

### 2.2.5 コントロールチェンジ (CC)

コントロールチェンジは CC とも書かれ、音量や音の性質を変化させる命令である。2 つの値がセットになっており、1 つ目の値を**コントロールチェンジ番号 (CC#)**と呼ぶ。2 つ目の値がデータ値 (0 から 127 をとる) である。CC#は 0 から 127 まで存在するが、よく使われるのは十数種類程度しかない。CC についての詳細な説明は[コントロールチェンジー一覧表\\*17](#)などを参照してほしい。多くの CC が XG ボイス (Ch.5-Ch.14) にしか対応していないが、CC#04 フットコントローラーと CC#11 エクスプレッションはエレクトーン本体にコントローラーがついており、EL ボイスの操作で使うことができる [4]。それ以外は基本的に XG ボイスの調節で使うと考えて構わない。以下、エレクトーン奏者が使うと思われる代表的な CC を列挙する。

#### CC#00/32 バンクセレクト

プログラムチェンジで大まかな音色を指定し、CC#00 でそのサブカテゴリを指定、CC#32 でさらにそのサブカテゴリを指定する。XG ボイスの音色指定で使う。CC#00 → CC#32 → プログラムチェンジの順に送信する。エレクトーンの XG ボイスリストは、公式には発表されておらず、[株式会社コムコムのデータライブラリ\\*18](#)などで入手する。

#### CC#01 モジュレーション

多くの場合、ビブラートの強さを指定する。

#### CC#04 フットコントローラー

**EL ボイスで使う。**セカンドエクスプレッションペダルの値。エレクトーンの場合、Ch.4(リード分離の場合) または Ch.16 の命令のみボイスに作用する。

#### CC#06 データエントリー

CC#98/99/100/101 と一緒に使う。説明はそちらを参照してほしい。

#### CC#07 ボリューム

チャンネルのボリュームを指定する。CC#11 エクスプレッションと挙動はほとんど同じだが、CC#07 はパート間の音量バランス調整で使い、ミキサー的な役割を担う。通常はセットアップ小節でのみ設定し、曲中では動かさない。デフォルトは CC#07=100。

#### CC#10 パン

チャンネルのパンを指定する。CC#10=0 でパンを左に全振りした状態になる。中央は CC#10=64。

#### CC#11 エクスプレッション

**EL ボイス、XG ボイス両方で使う。**エクスプレッションペダルの値。音量の抑揚表現やアクセントのために用いる。CC#07 と異なり、曲中で連続的に動かす用途を想定されている。デフォルトは CC#11=127。XG ボイス (Ch.5 から Ch.14) および EL ボイス (Ch.16) で使う。

#### CC#64 サスティン

サスティンペダル。64 以上でオン、未満でオフ。通常は 0 または 127 の値のみ使う。

#### CC#65 ポルタメント

ポルタメントとは、2 つの異なるノートを演奏したときに、その間の音程を滑らかにつなぐ奏法。64 以上でオン、未満でオフ。通常は 0 または 127 の値のみ使う。

#### CC#71 レゾナンス

CC#74 ブライツネスで指定した周波数の周りの周波数特性を変えて、特徴的な音色を出す。64 で効果

---

\*17 [http://quelque.sakura.ne.jp/midi\\_cc.html](http://quelque.sakura.ne.jp/midi_cc.html)

\*18 [http://www.comcom2.com/lib/els\\_ext\\_xg\\_voice\\_list.html](http://www.comcom2.com/lib/els_ext_xg_voice_list.html)



なし。

#### CC#74 ブライトネス

ローパスフィルターをかけて高音を削ることで音の明るさを調整する。64 で効果なし。それより上げるとカット周波数が高くなり高音が多くなり、下げるとカット周波数が低くなり高音が少なくなる。

#### CC#72 リリースタイム

#### CC#73 アタックタイム

#### CC#75 ディケイタイム

エンベロープの調整をする。64 で効果なし。上げるとタイムが長くなり、下げるとタイムが短くなる。エンベロープについての解説は [Wikipedia](https://ja.wikipedia.org/wiki/ADSR)<sup>\*19</sup>などにある。古典的なシンセサイザーはアタックタイム・ディケイタイム・サスティンレベル・リリースタイムの4つのパラメータを使って、波形の時間的な包絡線（エンベロープ）を設計する。このそれぞれのパラメータの頭文字を取って、エンベロープをADSRと呼ぶことがある。これらADSRはシンセサイザーの基本であるため、電子楽器使いなら知っておくべきである。また、[ドラムキットにかけることで面白い効果](#)を得られる [10]。

#### CC#91-95 エフェクトセンドレベル

エフェクトをどれくらいかけるかを設定する。エレクトーンで使うのは、CC#91 リバーブ、CC#93 コーラス、CC#94 バリエーションである。これらは sysEX（後述）でリバーブ・コーラス・バリエーションの種類を決めておいて、CC#91-95 でかかり具合を調節する形で使う。リバーブの種類はレジスト全体設定になるため、XG ボイスそれぞれで別々のリバーブタイプを設定することはできない [6-5]（スタイル入門講座へのリンク）。リバーブのデフォルトは 40 で、他は 0。

#### CC#98/99 NRPN（ノンレジスタードパラメータナンバー）

NRPN は、GM 規格によって動作が定義されていない、機器固有の命令を送信するために使われる。CC#98 と CC#99 の組み合わせで命令の種類を決定し、CC#06 でその値を送信する。命令の種類を指定した後は、CC#06 だけでその値を弄ることができる。値の調整が終わった後は、後述の RPN ヌルを送信して、誤作動を防止する。エレクトーンにおいて NRPN は XG ボイスのみ使われ、その動作の定義は XG 仕様書に書いてある [11]。

#### CC#100/101 RPN（レジスタードパラメータナンバー）

RPN は NRPN とは違い、GM 規格で動作が定義されているようなチャンネル設定を行う。CC#100 と CC#101 の組み合わせで命令の種類を決定し、CC#06 でその値を送信する。命令の種類を指定した後は、CC#06 だけでその値を弄ることができる。値の調整が終わった後は、RPN ヌルを送信して、誤作動を防止する。RPN の動作定義は表 1 の通り。

表 1: RPN の定義

CC#101	CC#100	動作	説明
00	00	ピッチベンド感度	ピッチベンドの効きの強さを指定する。
00	01	チャンネルファインチューン	ピッチを微量調節する。
00	02	チャンネルコースチューン	ピッチを調節する。
00	05	モジュレーションデプスレンジ	モジュレーションのビブラートの幅を調節する。
127	127	RPN ヌル	RPN のターゲットを指定していない状態にする。

\*19 <https://ja.wikipedia.org/wiki/ADSR>

## 2.3 MIDI システムメッセージ

MIDI システムメッセージは、演奏情報以外の楽器全体に関わる MIDI メッセージである。(チャンネルメッセージではないため、当然ながら) チャンネルの概念はない。**システムコモンメッセージ**と**システムリアルタイムメッセージ**、**システムエクスクルーシブメッセージ**に分けられる。

### 2.3.1 システムコモンメッセージ

sysEX (後述) の開始コードと終了コード、および複数の MIDI 機器を同期させて演奏するためのメッセージ群である。少なくとも私はそこまで重要ではないと思うため、割愛する。興味のある読者は MIDI 規格書を参照してほしい。

### 2.3.2 システムリアルタイムメッセージ

短いバイト数で、優先度も高く設定されており、リアルタイム性を保証するメッセージ群である。このメッセージが来ると、受信機器は現在の処理に割り込んで処理しなければならないとされている。こちらについても全ては解説せず、重要だと思う 3 つについて説明する。

#### スタート

エレクトーンではリズムスタートとして扱われる。

#### ストップ

同様に、リズムストップとして扱われる。

#### アクティブセンシング

接続ができているかの確認用の信号。演奏の途中で MIDI ケーブルが抜けてしまった場合、ノートオフ命令が届かずに音が鳴り続けてしまうなどの問題が起こりうる。これを防止するために、まともな MIDI 機器は接続中に常に一定の間隔でアクティブセンシングを送り続けており、受信側はアクティブセンシングが消えた段階でノートをオフしたり、リズムをストップしたりする<sup>\*20</sup>。

### 2.3.3 システムエクスクルーシブメッセージ (sysEX)

システムエクスクルーシブメッセージは **sysEX** とも書かれ、メーカーや機器ごとに特有の命令を書くためのメッセージである。GM では F0H から始まり F7H で終わることしか定義されていない。

ELS-02 シリーズの MIDI リファレンス [4] を参照すれば、**エレクトーンのほぼ<sup>\*21</sup>全ての命令を sysEX で制御できる**ことがわかる。しかもエレクトーンが対応している sysEX はこれだけではなく、**XG 規格の sysEX はここに載っていないし、公表されている XG 規格に載っていない XG パラメータも存在する<sup>\*22\*23</sup>**。この膨大な種類の命令が存在するおかげで、MIDI を駆使することによってしかできないパフォーマンスが可能になる。例えば、レジストレーションメモリーを消費せずに、8 つのボイス全てのエフェクトパラメータを同時に操作できたり、セカンドエクスプレッションペダルで EL ボイスのローパスフィルターのカット周波数を操

<sup>\*20</sup> 外部 MIDI 鍵盤と ELS-02C とを繋いで演奏していて、外部鍵盤を接続した EL のリズムが勝手に止まるのはアクティブセンシング (と、勝手に抜ける MIDI 端子) のせいである。

<sup>\*21</sup> ほぼ全ての、「ほぼ」について、例えばライトフットスイッチの命令が MIDI リファレンスに見当たらない。他にも操作できない操作子が存在するかもしれない。

<sup>\*22</sup> ヤマハ公式サイトから入手できる XG 規格書 [11] は 1999 年のもので、ELS-01 (2004 年) よりも昔である。当然、01 シリーズで追加された XG の sysEX は XG 規格に載っていない。しかも、ELS-01 シリーズの取扱説明書にも載っていない。このため、ELS シリーズの XG 規格周りは完全にブラックボックス化している。本稿を書くモチベーションは、このブラックボックスと化した XG やスタイルの周りの仕様をまとめ、分かりやすく伝えることにある。

<sup>\*23</sup> 機器専用命令で、取扱説明書 [22] や MIDI リファレンス [4] に掲載されていない sysEX が存在する。たとえば、sysEX を送信することでエレクトーン本体パネルのスクリーンショットが可能である。

作できたりする。

このように、sysEX をうまく活用することで表現の幅がグッと広がる。MIDI リファレンスを眺めながら、どのような工夫ができそうか考えてみてほしい。

## 2.4 スタンダード MIDI ファイル (SMF)

これまでの MIDI メッセージの説明は、全てリアルタイムな演奏情報のやり取りのための約束である。これとは対照的に、**スタンダード MIDI ファイル (SMF)** は、あらかじめ MIDI メッセージを保存しておいて、順次送信や再生をするための規格である。

エレクトーンで MDR 録音をしたファイルをパソコンで覗くと、(バルクファイル「.B00」の他に) MIDI ファイル「.MID」が確認できる。この「.MID」ファイルが、SMF である。世の中に出す「きちんとした」SMF には様々な取り決めがあるが、XG サポートやスタイルファイルの製作に用いる程度にしか使わないのであれば、次の点を理解しておけば良い。

- SMF にはフォーマット 0、1、2 の 3 種類があり<sup>\*24</sup>、XG サポートやスタイルファイルは**フォーマット 0**で製作する。
- SMF が処理できる最小の時間単位を**分解能**と呼び、この分解能の最小単位のことを **Tick** と呼ぶ。Domino を初めとする多くの MIDI シーケンサーにおいて、一般的に分解能は 1/480 四分音符がよく用いられる。エレクトーンの場合、MDR 録音したデータの分解能は 1/480 四分音符であり、リズムパターンプログラムやスタイルの分解能は 1/1920 四分音符である (ただし、エレクトーンがリズムパターンプログラムで表示する見掛け上の分解能は 1/480 四分音符である)<sup>\*25</sup>。
- できるだけ MIDI メッセージが同じタイミングに重複しないように、メッセージ間は最低 1 Tick ずつインターバルを空けるのが望ましい。

---

<sup>\*24</sup> フォーマット 0 は 1トラックで 16 チャンネル分のデータを格納する。フォーマット 1 はトラック毎に情報を独立して持っており、マルチトラックとして編集ができるほか、1 つのチャンネルに複数のトラックを対応させられる。フォーマット 2 は複数の曲を 1 つのデータに格納できるらしいが、普及しておらず、私も全くわからない。

<sup>\*25</sup> スタイルにおけるエレクトーンの分解能が 1/1920 四分音符であることについて、スタイル入門講座が解説をしている【10-116】。また、後述の EL Data Analyzer で切り出したスタイルを開いても、分解能が 1/1920 四分音符であることが確認できる。たとえ 480 でスタイルを自作しても、エレクトーン側で 1920 に自動的に再計算して読み込まれるため、この仕様の差について注意する必要はない。

## 3 XG サポートの作り方

既に図 2 で示したように、エレクトーンの MDR 録音機能を使って記録できる MIDI チャンネルメッセージは、上鍵盤 (Ch.1)、下鍵盤 (Ch.2)、足鍵盤 (Ch.3)、上鍵盤リードボイス (Ch.4)、キーボードパーカッション (Ch.15)、コントロール (Ch.16) である。これら以外のチャンネル (Ch.5-14) は、エレクトーン単体では演奏に使うことはできないが、エレクトーンの MDR 機能を使って、XG サポートデータが記録されている SMF を再生することで使うことができる。このようにして、Ch.5-14 も使って自動演奏や演奏補助をすることを、本稿では **XG サポート** と呼ぶ。

この章では、エレクトーンの自動演奏・演奏支援である **XG サポート** を作る方法について説明する。加えて、Windows で MIDI を扱う上で非常に頼りになるフリーソフトである Domino の使い方も、XG サポートを作りながら解説する。

### Domino の参考書

- Domino マニュアル  
Domino に付属しているマニュアルである。最も信頼できる参考書である。
- XG 楽曲データ製作の指針 [13]  
XG データを作成する際に気をつけるべきことがまとまっている YAMAHA 製のテキスト。XG サポートを作るなら一度は目を通しておくべきであろう。
- Dr. 青山の XG 解体新書 [12]  
YAMAHA の読み物。主に DTM 用 XG 音源のフラグシップモデルであった MU シリーズの仕様について解説してある。ELS-02 は MU50 相当の XG 音源再生能力・拡張 XG 音源を搭載しており、XG の基本的な使用は共通していると思われる。これを読むことができれば、XG についてさらに深い知識を得られる。

### 3.1 Domino の導入

Domino はたかば一氏によって製作された Windows 用 MIDI シーケンサである。公式サイト<sup>\*26</sup>から入手できる。本稿執筆時点での最新バージョンは 1.44 であるため、このバージョンの Domino を用いて解説を進める。他の MIDI シーケンサーでも同様のことはできるが、エレクトーン用に製作された音源定義ファイル (後述) があるため、Domino を使うことを勧める。

#### 3.1.1 音源定義ファイル

**音源定義ファイル**とは、楽器固有のプログラムチェンジや sysEX などを手軽に扱えるようにするための、Domino 専用のモジュールである。MIDI の低級な部分<sup>\*27</sup>を意識することなく MIDI 編集を行うことができるようになる。すずとも氏 (@SuzuTomo2001) と私が製作した。公式サイト<sup>\*28</sup>からダウンロードできる。本稿ではこの音源定義ファイルを導入する。

<sup>\*26</sup> <https://takabosoft.com/domino>

<sup>\*27</sup> ここでの「低級」は「機械語に近い」という意味である。MIDI リファレンス [4] を見ればわかるように、本来的には MIDI の制御は数字の羅列を取り扱う必要があるが、音源定義ファイルを使うことで、そのような面倒な部分を意識せずとも良くなる。

<sup>\*28</sup> <https://www.kamekyame.com/el/domino-define>

### 3.1.2 Domino の初期設定

#### 3.1.2.1 エレクトーンとパソコンの接続

まず、パソコンとエレクトーンを USB ケーブルで接続する。ELS-02/02C/02X, ELC-02, ELB-02 の場合<sup>\*29</sup>は、USB Type A-B ケーブル（俗にプリンタケーブルと呼ばれる）でエレクトーンの「TO HOST」端子に接続し、パソコンにエレクトーンを認識させる（図 3）。

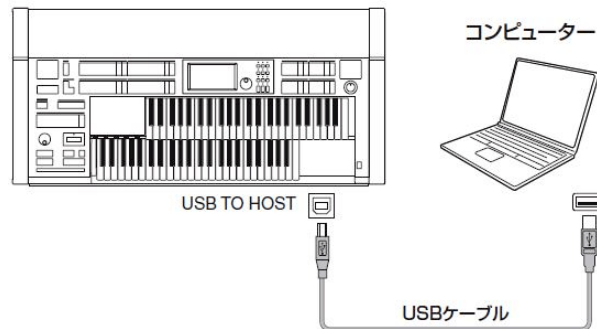


図 3: エレクトーンの TO HOST 端子とパソコンを接続する。エレクトーンとは USB で MIDI をやりとりできる。画像は[ヤマハ Q&A サイト](#)より引用。

#### 3.1.2.2 音源定義ファイルの導入

エレクトーンの接続が終わったら、以下の通りに音源定義ファイルを導入する。[音源定義ファイルの公式サイト](#)からダウンロードした electone.xml を Domino の Module フォルダの中に移動させ、Domino.exe をダブルクリックして起動する。Domino のメニューから「ファイル (F)」→「環境設定 (E)...」を開く（もしくは F12 キーを押す）。現れた環境設定ウィンドウの左側のカテゴリから「MIDI-OUT」を選択し、ポート A の音源定義ファイルを「YAMAHA」フォルダの中の「Electone」に設定すれば、導入が完了する（図 4）。

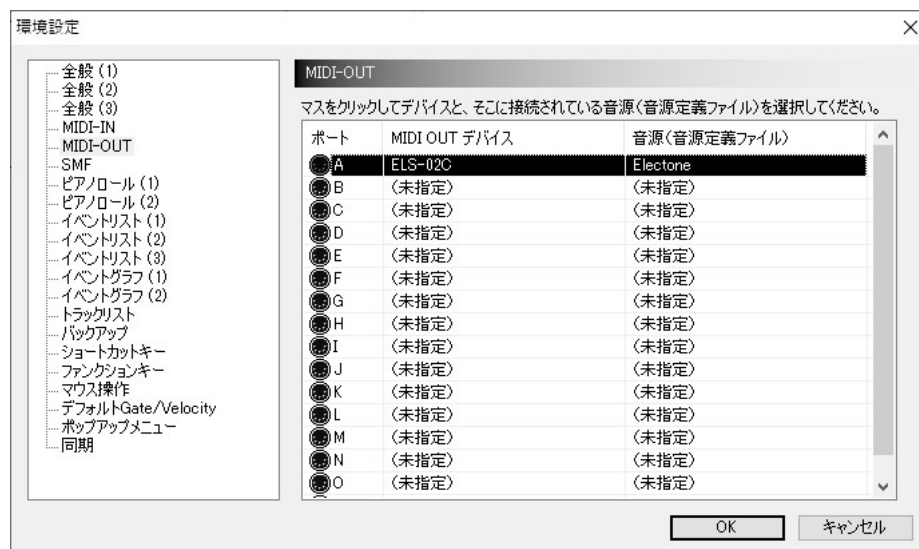


図 4: Domino の MIDI-OUT 環境設定の例。初期設定が終われば以降は設定する必要はない。

<sup>\*29</sup> その他のエレクトーンの場合は、ヤマハ公式から USB-MIDI Driver をインストールする必要がある。詳しくは、[ヤマハ公式の Q&A](#)<sup>\*30</sup>を参照せよ。

### 3.1.2.3 MIDI-OUT の設定

同じ画面で MIDI-OUT の設定を行う。MIDI-OUT デバイスとして「ELS-02C」のようなエレクトーンが選択できるようになっているから、ポート A の MIDI-OUT デバイスをエレクトーンにしよう (図 4)。これにより、Domino で入力した (ポート A の) MIDI データはエレクトーンに送信されるようになった。もしもエレクトーンが表示されていなければ、Domino を一旦終了して、エレクトーンとパソコンを USB で接続してから Domino を再起動する。エレクトーンをパソコンから操作しようとするときは、**エレクトーンとパソコンを USB で接続してから Domino を起動**しなければならないことを覚えておこう。

### 3.1.2.4 MIDI-IN の設定

必要であれば MIDI-IN の設定を行う<sup>\*31</sup>。MIDI-IN の設定を行うと、エレクトーンを含む MIDI キーボードによる演奏を Domino で認識できるようになり、打ち込みの速度が飛躍的に向上する。ただし、エレクトーンはオクターブシフトができず、特にドラムの打ち込みに難があるため、エレクトーン以外の MIDI キーボードを新しく準備することを勧める。

### 3.1.2.5 その他の設定

以上の他に、(必須ではないが) 設定しておくとして作業が捗る設定が複数存在する。本稿では**以下に示す設定を設定済みとして解説を進める**。設定しておくことを強く勧める。

- メニューバーの「表示 (V)」→「トラックセレクトペイン (A)」を押し、「トラックセレクトペイン」を表示する。これを表示することにより複数のトラックの行き来が GUI で簡単に行えるようになる。
- 全般 (1) で「時間の表し方」を「Measure:Beat:Tick」にする。イベントリストで拍が表示されるようになる。
- 同じく全般 (1) で「オクターブ」を「Note#60=C3」にする。Note#60 は国際式とヤマハ式でキーが違っており、このように設定するとヤマハ式にできる。
- 全般 (2) で「マクロ番号の表示」を「表示する」に設定する。**コントロールチェンジマクロ (CCM)**<sup>\*32</sup>の番号が表示されるようになる。
- イベントリスト (1) で「コントロールチェンジイベントの番号」を「表示する」に設定する。
- 同じくイベントリスト (1) で「リズムノートイベントのノート番号」を「表示する」に設定する。ドラムトラックは打ち込んだノートが楽器名で表示されるが、ここにノートナンバーを付記できる。

## 3.2 XG サポートを作る

いよいよ XG サポートデータを作っていく。この節では詳細な仕様の解説はせずに、XG サポートを作るための最小の手順を解説する。

なお、前提として **USB メモリの中にプロテクトソングは無い方が良い**<sup>\*33</sup>。また、**Windows の設定で拡**

<sup>\*31</sup> 私はノートの打ち込みは全て Cubase で行っているため、MIDI-IN の設定はしていない。Cubase で打ち込みをするには次のようにすれば良い：Cubase で打ち込んだデータを SMF として書き出す。Domino を 2 台立ち上げ、片方はスタイルファイルや XG サポートのフォーマットを整え、もう片方で Cubase の MIDI を開く。Domino はウィンドウ間でのコピーアンドペーストができるため、Cubase の MIDI を全選択してコピーし、もう片方の Domino に貼り付ける。

<sup>\*32</sup> コントロールチェンジマクロ (CCM) とは、RPN/NRPN や sysEX などの複雑な命令をコントロールチェンジ (CC) のように簡単に取り扱えるように設計された Domino の機能である。CC は 1 番から 127 番までしか存在しないが、CCM は 1300 番まで使用できるようになっている。本稿では CCM#として CCM の番号を指定する。なお、CCM は音源定義ファイルによって定義されているため、音源定義ファイルが異なれば挙動が全く違うものになる。したがって、**本稿で紹介したすずとも氏の音源定義ファイル以外を使う場合は、本稿の CCM#の記述は全く役に立たないことに注意してほしい。**

<sup>\*33</sup> プロテクトソングは鍵付きのアイコンで表される。ヤマハ公式が配布しているソングや有料で購入したソングなどがこれに含まれる。なお、**プロテクトソングをパソコンから適当に触ってはいけない**。移動するには「ヤマハミュージックソフトダウンローダー」



張子を表示するように変更しておかなければならない。また、本稿の内容を実行する場合は、**本稿の内容を実行して生じる結果について、私が一切の責任を負わないことを認めたものとする。**

### 3.2.1 ファイルの作成

XG サポートデータは SMF であるため、Domino などで白紙から作ることもできるのだが、エレクトーンの MDR 機能を用いることで自動で生成される SMF から作り始めた方が簡単である。そこで本稿では、XG サポートデータの元となるファイルをエレクトーンにて MDR 録音することで作成する<sup>\*34</sup>。

曲の始めから終わりまでのソングデータ（シーケンスを含む）を完成させた後、このソングだけが入っているようなフォルダをエレクトーン上で用意する。以下ではこのフォルダを「XGSupportFolder」と名付けたことにする。

図 5 のように、このソングに演奏を付け足す MDR 録音をする。録音待機の画面で録音するチャンネルを選択できるので、「コントロール」だけを録音するように設定し<sup>\*35</sup>、録音を開始する。録音中にシーケンスを ON にしてリズムスタートさせ、楽曲が終わるまで（演奏せずに）待つ。シーケンスが終わったら録音を停止する。こうすることにより、曲の開始時刻（リズムスタート）と終了時刻、および拍子の情報が SMF に書き込まれる。なお、シーケンス関連で書き込まれるのは、シーケンスの ON/OFF およびリズムの Start/Stop のみであり、リズムパターンのノートやレジストチェンジが書き込まれることはない<sup>\*36</sup>。

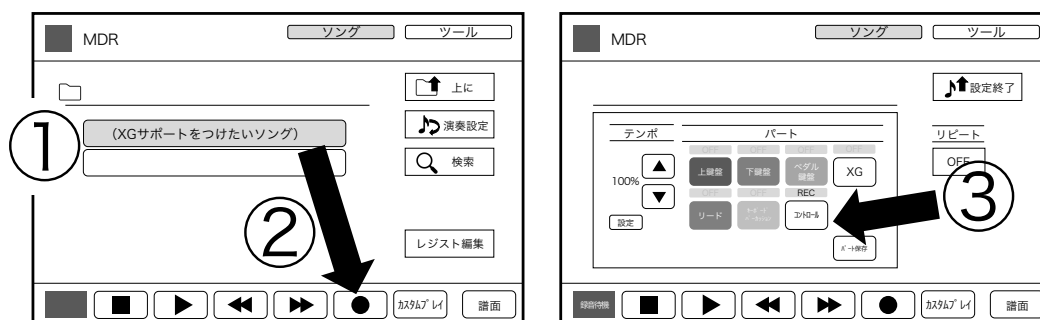


図 5: XG サポートの付け方。① 1 つのソングだけが入ったフォルダを用意し、②そのソングに録音を付け足す。③コントロールだけを録音するよう設定し（他については OFF または PLAY にする）、シーケンスを曲の終わりまで流す。

というソフトが必要である。今からソングをエクスプローラで細工するため、事故防止のために、プロテクトソングが 1 つも入っていない USB メモリを準備した方がよい。

<sup>\*34</sup> MDR 録音によって得られるデータのうち、XG サポートデータ作成において欲しいデータは、**リセット系、拍子、テンポ、シーケンス ON、リズムスタート、リズムストップ、End of Track** である。

<sup>\*35</sup> コントロール以外の録音を ON にした場合、Ch.1 には上鍵盤、Ch.2 には下鍵盤、Ch.3 にはペダル鍵盤、Ch.4 にはリード 1、Ch.15 にはキーボードパーカッションが録音される。

<sup>\*36</sup> シーケンスの情報はエレクトーン内部やバルクファイルで持っており、SMF には書き込まれない。SMF には、シーケンスの開始・終了のみが書き込まれる。そのため、作成した SMF の Ch.9 や 10（通常アドラム・メインドラムとして使用される）にリズムパターンは打ち込まれることはない。エレクトーンでこの SMF を再生した時のみ、紐づけられたレジストデータに入っているシーケンスが再生されることになり、リズムパターンやレジストチェンジが再生される。



次に先ほどの録音データが入っている USB メモリをパソコンで開き、ソングデータを確認してみよう。図 6 のように、ソングが 1 つだけ入っているフォルダ (XGSupportFolder) の中に「ELS\_SONG.NAM」と「SONG\_001」という名前のフォルダがある。SONG\_001 の中にある「MDR\_000.MID」が、演奏情報が入っている SMF である。なお、「REG\_00X.B00」(X は数字が入る) は**バルクファイル**といって、エレクトーンのレジストデータ<sup>\*37</sup>が入っている。ネクストレジストを使う場合は 2 つ以上存在する。

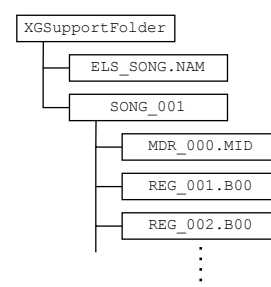


図 6: ソング内の構造。

### 3.2.2 XG サポートファイル製作の下準備

MDR\_000.MID に XG サポートの情報を書き加えていく前に、Domino の基本的な使い方を確認しつつ、必要なデータを削除していこう<sup>\*38</sup>。

まず、先ほどエレクトーンで作成した演奏データ「MDR\_000.MID」をパソコンにコピーした後、Domino にドラッグアンドドロップすることで開く。開くときに、「MIDI データを解析してコントロールチェンジマクロを復元しますか?」と聞かれるので、「はい」を選択する。これにより、SMF(.mid ファイル) から Domino のプロジェクトファイル(.dms ファイル) に変換される。Domino で編集している途中のファイルや、プロジェクトのバックアップは.dms ファイルで保存する。

読み込みが完了したら、まずメニューバーの「ファイル (F)」→「名前をつけて保存 (A)...」(または「Ctrl+Shift+S」) から、プロジェクトファイルの保存をしよう。保存された.dms ファイルを用いて、以降は編集や保存を行う。

#### 3.2.2.1 バルクダンプの削除

最初に [System Setup] の内容を見ていく。図 7 画面一番左にある<sup>\*39</sup>「トラックセレクトペイン」から、[System Setup] をクリックしよう。すると、表示・編集するトラック (カレントトラックという) が [System Setup] になる。この [System Setup] トラックは、MIDI システムメッセージの記述に使う。主に sysEX の記述に用いる<sup>\*40</sup>。

[System Setup] のイベントリストを見ると、2 つめのイベントとして「Ex:f0h 43h 70h 78h ...」が入っているだろう。このイベントをダブルクリックすると、1 画面に表示しきれないほどの大量のメッセージで構成されていることが確認できる。これは**バルクダンプ**といって、この演奏データ「MDR\_000.MID」がエレクトーンで呼び出された際に設定される一番最初のレジストの情報が入っている。しかし、脚注<sup>\*41</sup>で示すように、このデータを残すとエレクトーンが誤作動を起こす恐れがある。イベントリストでクリックして黒く選択した上で、キーボードの delete キーを押して**必ず削除**しよう。

<sup>\*37</sup> .B00 ファイルにはレジストデータ、レジストシフトの順番、ユーザーボイス、オルガンフルート、シーケンス、ユーザーリズムが連結して入っている [20]。余談だが、ユーザーリズムの手前までは固定長データで、ユーザーリズムからは可変長になるため、01 シリーズのユーザーリズムデータ (スタイルという) の切り出しは容易であった (スタイルは基本的には SMF であり、SMF データは 4DH 54H 68H 64H から始まるため、これを探せば良い)。02 シリーズになると、拡張フォーマットとしてユーザーリズムデータの後ろに固定長のデータが付く。

<sup>\*38</sup> 基本的に、不必要な MIDI メッセージは書くべきではない。受信楽器に負担を強いることになるし、編集上邪魔になる。

<sup>\*39</sup> トラックセレクトペインが表示されていない人は、「Domino の初期設定」の節を読み直してほしい。

<sup>\*40</sup> sysEX は動作に時間がかかるため、同一 Tick に複数存在してはいけない。そのため、sysEX 専用のトラックを用意して各チャンネルトラックに sysEX がばらけないようになっている。

<sup>\*41</sup> バルクダンプデータは非常に大きいメッセージで、MIDI 受信ソフトウェアによってはバッファ容量を超過するようである。実際、私がテストデータとしてエレクトーンで作った SMF を Domino で開いたところ、バルクダンプデータが F7H で終わっていないように見えた。sysEX は必ず F7H で終わる必要があるため、Domino 側で処理できていないと思われる。よって、このバルクダンプを放置するとエレクトーン側で誤作動する恐れがあるため、必ず削除しよう。参考: <https://wikiwiki.jp/tkbsoft/domino/%E8%A6%81%E6%9C%9B003/207>

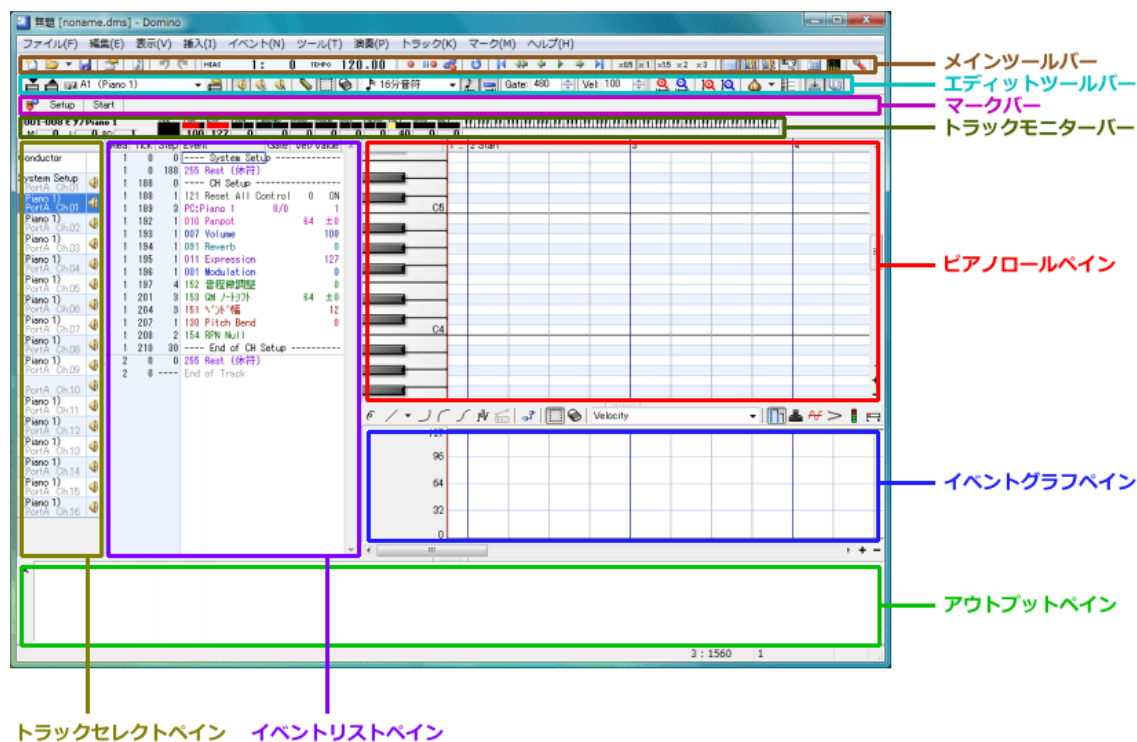


図 7: Domino の画面名称。図は Domino マニュアルから引用。

その他、[1206 Bar Signal<sup>\*42</sup>] や、[509 Rhythm Stop<sup>\*43</sup>] など削除できる。必要に応じて削除しよう。

### 3.2.2.2 Ch.16 のメッセージの削除

必要であれば Ch.16 のメッセージを削除する。画面左のトラックセレクトペインから、[Ch.16] を選択すると、[011 Expression] や [004 2nd Expression] が表示されているだろう。これらはエクスプレッションペダルやセカンドエクスプレッションペダルの入力の記録である。録音中にエレクトーンを操作していれば、それ以外にも様々なコントロールが記録される。XG サポートにおいて再生して欲しくないコントロールがあれば、ここを削除すると良いだろう<sup>\*44</sup>。

## 3.2.3 XG サポートデータの記述

ここまでで下準備が終わったので、サポートデータを打ち込んでいくことができる。

### 3.2.3.1 XG モードへの変更

起動した時点ではエレクトーンは **EL モード** である。これに対し、取扱説明書には載っていないが、**XG モード** というモードもエレクトーンにはある。XG モードは DTM 用音源と互換性のあるモードで、エレクトーンを XG 音源装置として用いることができるようになる。外部からの XG 命令を受け付けるようにするために、エレクトーンを XG モードに変更する。なお、現在のモードがどちらなのかをエレクトーン上で確認す

<sup>\*42</sup> パーシグナルは、エレクトーン本体左手のテンポ液晶の上にある BAR/BEAT 表示の赤い LED が点灯すると送信される。

<sup>\*43</sup> [509 Rhythm Stop] はリズムが止まったときに送信される。シーケンスを組まないなら実用的（手でリズムを止めるのと同等のメッセージ）だが、シーケンスを組んでいればシーケンスの終わりで自動的にリズムが止まるため、この場合は無くても構わない。

<sup>\*44</sup> なお、XG サポート再生中はエレクトーンのデフォルト設定ではエクスプレッションペダルが操作できない（Ch.16 に追従する）。これを操作可能にするには、エレクトーン本体液晶右側の UTILITY ボタンを押し、画面上部のタブメニューより MIDI を選択し、エクスプレッションペダルを「インターナル」に設定すれば良い。

ることはできない。

[System Setup]トラックを開いて、一番最初のメッセージである [1205 EL ON] をダブルクリックする。これを [205 XG System ON] に変更して、ウィンドウ左下の「テスト送信」を押してから OK を押す。これにより、エレクトーンが XG モードになった。XG モードにしておかないと、外部から XG 拡張音源のプログラムチェンジを送信したときに XG ボイスに変更されない。**XG 音源の選択の際には XG モードにしておく必要がある。**

XG サポートの打ち込みが終わった後は、冒頭のリセット命令を [1205 EL ON] に戻すことができる。戻した場合は、EL モードの状態<sup>\*45</sup> XG サポートが再生される。ただし、エレクトーンが EL モードのとき、受信したチャンネルメッセージは表 2 のように、EL ボイスと XG ボイスに分かれて作用する。一方、XG モードのとき、受信したチャンネルメッセージは全て XG ボイスに作用する<sup>\*46</sup>。XG サポートで XG ボイスだけを打ち込むならば XG モードで十分だが、たとえば「Ch.1 から 4 も MIDI を打ち込んでおいて、EL ボイスを同時に鳴らしたい」という場合に EL モードが有効である。

表 2: モードによって受信したチャンネルメッセージが作用するボイスが異なる。

受信チャンネル	EL モードの場合	XG モードの場合
Ch.1	上鍵盤ボイス	XG ボイス
Ch.2	下鍵盤ボイス	XG ボイス
Ch.3	足鍵盤ボイス	XG ボイス
Ch.4	リードボイス <sup>*47</sup>	XG ボイス
Ch.5, ..., 14	XG ボイス	XG ボイス
Ch.15	キーボードパーカッション	XG ボイス
Ch.16	コントロール	XG ボイス

ところで、[205 XG System ON] や、[1205 EL ON] を送信するとエレクトーンのモードが変わり、**XG ボイスの設定がリセット**される。これらのモード指定がない SMF を再生すると、チャンネル設定などがその前に再生したソングのままになってしまい、意図した効果が出せない恐れがある。事故を防ぐために、**SMF の最初には必ず、モードを指定する命令を書いて、機器の初期化をしなければならない**<sup>\*48</sup>。

また、演奏会などで XG モードにしたまま他の人が演奏すると、例えば外部 MIDI キーボードをエレクトーンに接続して EL ボイスを鳴らそうとしても、XG ボイスが再生されてしまう。エレクトーンは基本的に EL モードで使うべきであるから、XG サポートデータが終わったら [1205 EL ON] をして EL モードに戻しておこう。EL モードで XG サポートデータを再生する場合は [1205 EL ON] は必要ないが、そのような場合分けは事故の原因となりうるため、再生するモードにかかわらず、**XG サポートデータの一番最後には [1205 EL ON] を入れることを強く要請する。**

<sup>\*45</sup> EL モードであっても、MDR 機能を使って再生している場合は、XG モードと同様に XG 拡張音源が再生される。

<sup>\*46</sup> XG モードのときも、上鍵盤の演奏は上鍵盤ボイスで再生される。2つのモードは、あくまでも外部から受信したチャンネルメッセージをどのボイスで再生するかという違いであって、たとえ XG モードだとしても XG ボイスをエレクトーンの鍵盤で演奏できるようにはならない。

<sup>\*47</sup> エレクトーン本体の設定画面でリードボイス発音設定をエクスターナルにしたときに限り、受信した Ch.4 がリードボイスで再生される。

<sup>\*48</sup> この初期化には時間がかかるため、本来は 30 Tick ほど（XG 規格では 50 ms ほどの実行時間であると説明されている [11]）空白を設けて次のメッセージを書くのが良いとされる。ところがエレクトーンで生成した SMF はノータイトで次の命令が書いてあり、あるべき空白が存在しない。エレクトーンは処理能力が高いのか、もしくはセットアップ小節の内容は順次実行されるようになっており、インターバルが必要ないのかもしれない。

### 3.2.3.2 楽器指定（プログラムチェンジ）

次は XG サポートに用いる楽器を指定しよう。

トラックを [Ch.5] に合わせ<sup>\*49</sup>、ピアノロール上で 1 小節目の 3 拍目をクリックし、演奏線を 1:3:0<sup>\*50</sup>に合わせる。そうしたら、メニューバーから「挿入 (I)」→「プログラムチェンジ (P)」を選択する。イベントリストに [Program Change] が挿入されるので、その文字列をダブルクリックしよう。するとプログラムチェンジを指定するウィンドウが表示される。(エレクトーンを繋いでいれば) このウィンドウで音色を選択するとエレクトーンから XG ボイスが鳴るはずなので、それを聞きながら好きな音色を選択しよう。

### 3.2.3.3 その他のチャンネルセットアップ

プログラムチェンジの次は XG ボイス用のチャンネルセットアップを行おう。

まず、ピアノロールを適当にクリックしよう。このときペンツールになっていた場合は、ノートが打ち込まれてしまうが、そのノートはダブルクリックで消そう。ピアノロールをクリックしたことにより、ピアノロールの演奏線上でコントロールチェンジの挿入が行われるようになった。次に画面上部の「MEAS」をクリックして演奏位置を 1:3:10 に指定する<sup>\*51</sup>。そうしたらメニューバーから「挿入 (I)」→「コントロールチェンジ (複数) (G)」を選ぶ。挿入するコントロールチェンジを選択できるので、CC#7, 11, 91, 93 のチェックボックスにチェックを付けよう<sup>\*52</sup>。OK を押すとイベントリストに先ほど選択した CC が追加されるので、1:3:10 から順番に CC が追加されていることを確認したら、追加された CC をダブルクリックして、表 3 に示されるデフォルトの値を打ち込んでいく。

表 3: 記述が義務付けられているチャンネルセットアップのコントロールチェンジ [13]。

CC#	説明	デフォルトの値 [11]
#7	Volume	100
#11	Expression	127
#91	Reverb Send	40
#93	Chorus Send	0
#94	Variation Send <sup>*53</sup>	0

指定しなかったデータはデフォルトの値で補完されるため、MIDI データが重くならないように必要のないデータは記述しない方が良い。ただし、表 3 に示すデータは、デフォルトであっても記述することになっている<sup>\*54</sup>。記述することが指示されている MIDI メッセージ (表 3) 以外にも、CC#10 Pan(デフォルト 64)、CC#74 Brightness(デフォルト 64) などを入力できる。これらの CC を使って音作りをしたい場合は、同様にセットアップで入力すると良い。デフォルト値は XG の仕様 [11] などで確認する。

このように、SMF において 1 小節目は**セットアップ小節**として用いられる。原則としてノートは打ち込ま

<sup>\*49</sup> XG サポートを Ch.5 から打ち込むのは、Ch.1-4 がエレクトーンの鍵盤に対応しており、EL モードの場合に EL ボイスが鳴るためである。XG モードで再生することが確定していれば、Ch.1 から打ち込み始めても構わない。

<sup>\*50</sup> 本稿では SMF における時刻を「Measure:Beat:Tick」と表記することにする。1:3:0 であれば、1 小節目 3 拍目 0Tick である。

<sup>\*51</sup> 今回はプログラムチェンジを 1:3:0 で行ったので、それより後でチャンネルセットアップを行う必要がある。そのため、今回は 1:3:10 でチャンネルセットアップを開始するように解説した。プログラムチェンジよりも後にチャンネルセットアップを行っていればどこで入力しても良いが、最初の発音 (ノートオン) はセットアップの終わりから余裕を持ってインターバルを取るのが良い。

<sup>\*52</sup> コントロールチェンジ (複数) のウィンドウ左下にある Step はイベント間の Tick 数を指定できる。デフォルトでは 10 Step となっており、これで問題ないが、1 小節目 (セットアップ小節) の中に収まりきらないようであれば、間隔を詰めるために 1 Step などに設定できる。

<sup>\*53</sup> チャンネルセットアップで CC#94 Variation を入力する必要があるのはバリエーションエフェクトがシステムエフェクトの場合のみである。システムエフェクトに関する解説は次の節で行う。

<sup>\*54</sup> XG データ製作指針 [13] で示されているから説明しているが、CC#7, 11, 91, 93 のイベントの入力をサボった場合はデフォルトの値で補完されるようなので、自分で使うだけならこの段取りは必要ないかもしれない。

ず、システムセットアップやチャンネルセットアップを入力する。今回は [Ch.5] トラックだけ解説しているが、複数のチャンネルを使って打ち込みをするなら、全てのチャンネルでチャンネルセットアップを行わなくてはならない。

### 3.2.3.4 ステップ録音

いよいよノートを打ち込んでいこう。ピアノロール上でマウスでカチカチと打ち込んでもいいが、ここでは MIDI キーボードを用いて行う**ステップ録音**を解説する。ステップ録音とは、一音（一和音）ずつ時間を止めながら演奏を録音できる機能である。ゆっくりと自分のペースで演奏できるため、無駄なく正確にノートを打ち込むことができる。MIDI-IN の設定をしていればエレクトーンを初めとした MIDI キーボードを使うことができるし、そうでなくともパソコンの qwerty キーボード<sup>\*55</sup>を用いて打ち込むこともできる。

ステップ録音ウィンドウを表示するには、メニューバーの「編集 (E)」→「ステップ録音 (S)」を選択するか、もしくは画面上部「MEAS」の右横にある「|| ●」アイコンを押す。現れたステップ録音ウィンドウがアクティブなときに MIDI キーボード（もしくは、「PC キーボード」を押すと出てくる鍵盤に対応するキー）を演奏すると、ピアノロールの演奏線の位置に、ステップ録音ウィンドウで選択されている音価（Step）でノートが打ち込まれる。「PC キーボード」で打ち込んでいる場合は「Velocity」を変更することで打ち込まれるノートのペロシティを変更できるほか、「オクターブ」を変更することでオクターブシフトができる。「Gate(G)」を変更すると、指定した音価に対するゲートタイムの割合を指定できる。PC キーボードで「Ctrl + Z」を入力することで Undo できるほか（Redo は「Ctrl + Y」）、矢印キーで Step を移動させることもできる。

[System Setup] トラックを見て、[508 Rhythm Start] のイベントの時刻（おそらく 2:1:0 であろう<sup>\*56</sup>）を確認しよう。そこから計算することで自分が XG サポートを打ち込みたい時刻がわかるだろう。その時刻に演奏線を移動させて、[Ch.5] にステップ録音でフレーズを打ち込もう。

なお、音価の変更にキーボードショートカットを設定しておくことを強く推奨する<sup>\*57</sup>。ステップ録音ウィンドウの「設定 (C)...」からキーボードショートカットを設定できる。慣れてくればマウスを使わずに、リアルタイムで演奏するよりも早く打ち込みできるようになる。

### 3.2.3.5 ピアノロール

**ピアノロール**は縦軸にキー、横軸に拍をとった楽譜の表現の一つである。Domino で打ち込むノートは、基本的にピアノロールかイベントリスト（後述）で編集することになる。

Domino 画面左上部のペンのアイコンをクリックするか、またはメニューバーの「ツール (T)」→「ペン (P)」からピアノロール上のマウスポインタを**ペンツール**として使用できる。この状態でピアノロール上の空白をクリックするとノートを配置できる。配置されたノートの左端か右端をドラッグアンドドロップすることでノートを伸ばすことができる。ダブルクリックでノートを消すこともできる。ノートの配置が**クオンタイズ**<sup>\*58</sup>されているときは、Domino 画面左上のドロップダウンメニューからクオンタイズの拍を選ぶことができるほか、そのすぐ右のアイコンでクオンタイズをしないようにすることもできる。

ペンアイコンの右は**選択ツール**である。ピアノロール上でドラッグアンドドロップすると、複数のノートを範囲選択できる。選択した状態でドラッグアンドドロップすれば、移動・縮小・拡大などの操作ができる。ま

<sup>\*55</sup> ここで qwerty キーボードと呼んでいるものは、パソコンで文字などを入力するための装置である。単に「キーボード」と書くと演奏用の装置と区別がつかないため、「qwerty」を語頭に付けた。

<sup>\*56</sup> MDR 録音のときの操作のタイミングによっては、Rhythm Start が 2:1:0 ではないことがある。MDR 録音によって記録される Rhythm Start は（時刻の Beat, Tick が 0 になるように）かなり強力なクオンタイズがかかっているだけで、2:1:0 に必ず配置されるわけではない。Rhythm Start が 2:1:0 になかったとしても（本文の方法と同様に）フレーズを打ち込む時刻を Rhythm Start の時刻を基準に計算することで、特に問題はなく XG サポートを打ち込むことができる。

<sup>\*57</sup> 私は楽譜製作ソフトウェア Musecore のステップ入力の一貫性を保つように、全音符を数字「7」キーに、2 分音符を数字「6」キーに、…、32 分音符を数字「2」キーに設定している。

<sup>\*58</sup> クオンタイズとは、MIDI イベントのタイミングを拍のキリのいい場所に揃える機能のことである。



た範囲内を右クリックすれば様々な操作が行える。

選択アイコンの右は消しゴムツールである。ピアノロール上で範囲選択し、その中のノートを消すことができる。

### 3.2.3.6 イベントグラフ

ステップ録音やペンツールを使ってノートの情報を打ち込んだら、次はペロシティの調節やエクスプレッションの書き込みなど、ノートではない情報の仕上げをしよう。このときに便利なのが**イベントグラフ**である。図 8 にペロシティの変更の例を示す。イベントグラフ上で曲線を描くと、その曲線に沿って様々なデータが指定される。

イベントグラフペイン左上で曲線の種類を選択できる。図 8 の例では直線を選択した。他に、フリーハンドやプリセットの曲線を選択できる。また、関数を自分で作り、オリジナルの曲線を設定することもできる。

イベントグラフペイン中央上のプルダウンメニューからは、音源定義ファイルで設定されている CCM とペロシティの中から、イベントグラフで編集するパラメータを指定できる。また、イベントグラフペインの右上には使用頻度の高いパラメータ\*59がアイコンで指定できるようになっている。

なお、イベントグラフペインは最大 2 つまで同時に表示できる。メニューバーの「表示 (V)」→「イベントグラフペイン 1/2」から表示切替が行える。

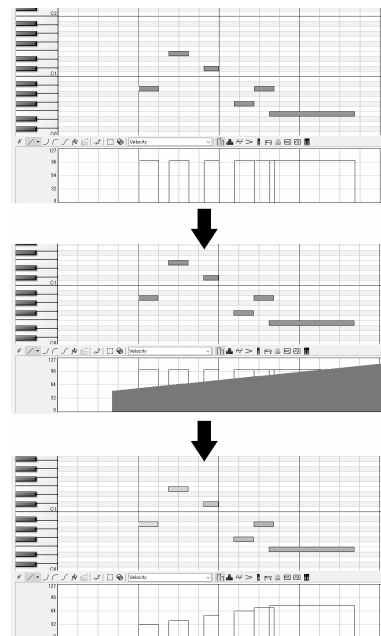


図 8: イベントグラフの使い方。ペロシティを編集するモードで直線を描くと、配置されているノートのペロシティが描いた直線に沿って指定される。

## 3.2.4 XG サポートデータの保存とソング情報の書き換え

### 3.2.4.1 EL ON の配置と End of Track の調節

全ての編集が終わったら、最後のイベントの後に [System Setup] **トラック**で [1205 EL ON] を配置する。[1205 EL ON] は全ての XG 設定がリセットされるため、曲の途中で [1205 EL ON] を配置してはならない。必ず曲の最後に [1205 EL ON] を配置する。

[1205 EL ON] を配置したら、キーボードの Tab キーを押すか、またはメニューバーの「表示 (V)」→「トラックリスト (L)」から**トラックリスト**を表示する。先程配置した [1205 EL ON] の次の小節に演奏線を合わせ、メニューバーの「イベント (N)」→「End of Track の調節 (D)」を選択して、**End of Track**\*60を設定する。これで XG サポートデータは完成である。なお、トラックリストビューから戻るには、もう一度 Tab キーを押せば良い。

### 3.2.4.2 XG サポートデータの保存

プロジェクトファイル (.dms ファイル) はエレクトーンで読み込みできないため、SMF を書き出す必要がある。メニューバーの「ファイル (F)」→「SMF 書き出し (M)...」を選び、現れたウィンドウで保存場所と

\*59 Domino Ver.1.44 時点で、アイコンから Velocity, Pitch Bend, Modulation, Expression, Volume, Panpot, Hold, Brightness, Resonance, Master Volume が指定できる。

\*60 End of Track とは MIDI データにおけるトラックの最終時刻である。End of Track の時刻で、エレクトーンは XG サポートの再生を終了する。

名前を指定する。このとき、**フォーマットを「format 0」にすることを忘れないこと**。また、名前は半角アルファベットのみにすることが望ましい。ここでは、例として「XGS.MID」と設定したとする。

### 3.2.4.3 USB メモリへの転送と ELS\_SONG.NAM の編集

作成した SMF(XGS.MID) を、エレクtoonで録音データを作成した場所に戻す。この際、もともと存在した MDR\_000.MID は消去する。

図 9 右のような構造にできたら、SONG\_001 と同じディレクトリに存在する ELS\_SONG.NAM の拡張子を .NAM から .txt に変更する。警告が出る場合があるが、そのまま拡張子を変更しよう<sup>\*61</sup>。

拡張子の変更ができたなら、ELS\_SONG.txt をダブルクリックで開く。開くと、ソース 1 のような内容が書かれている。

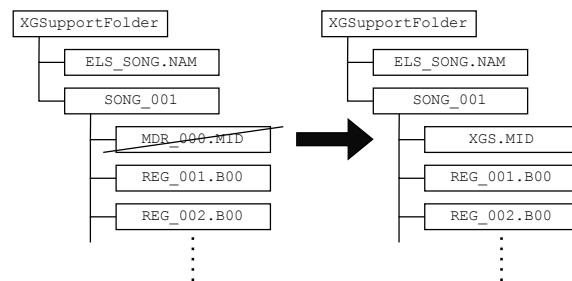


図 9: もともと存在した MDR\_000.MID を消去し、作成した XG サポートデータを入れる。

ソース 1: ELS\_SONG.NAM (変更前)

```
1  SOOX:SONGNAME = SONG_00X
2  SOOX:FOLDER = SONG_00X
3  SOOX:SECURITY = OFF
4  SOOX:MODEL =
5  SOOX:PART_UK = OFF
6  SOOX:PART_LK = OFF
7  SOOX:PART_PK = OFF
8  SOOX:PART_LEAD = OFF
9  SOOX:PART_KBP = OFF
10 SOOX:PART_CTRL = PLAY
11 SOOX:PART_XG = OFF
12 SOOX:MIDFILE = MDR_000.MID
13 SOOX:BLKFILE_001 = REG_001.B00
```

冒頭の SOOX (X には数字が入る) は、ディレクトリ内のソングのユニークナンバーになっている。エレクtoon内ではこの番号でソングの管理を行うようである。コロンの後にパラメータの種類を指定し、その後に変数を指定する構造になっている。各パラメータの内容は以下の通り [24]。

<sup>\*61</sup> ELS\_SONG.NAM の拡張子を .txt に変更しなくとも、任意のテキストエディタでこのファイルを開けば良い。ここでは、パソコンに詳しくない読者でも編集が行えるように、拡張子の変更をすることで windows 標準のテキストエディタを呼び出している。



SONGNAME	MDR で表示されるソング名。
FOLDER	ソングが入っているフォルダ。
SECURITY	レジストの保護の有無 (ON/OFF)。
MODEL	演奏を想定されたモデル。
PART_UK	上鍵盤の自動演奏の有無 (PLAY/OFF)。
PART_LK	下鍵盤の自動演奏の有無 (PLAY/OFF)。
PART_PK	足鍵盤の自動演奏の有無 (PLAY/OFF)。
PART_LEAD	リードボイス 1 の自動演奏の有無 (PLAY/OFF)。
PART_KBP	キーボードパーカッションの自動演奏の有無 (PLAY/OFF)。
PART_CTRL	コントロールの自動演奏の有無 (PLAY/OFF)。
PART_XG	XG サポートの有無 (PLAY/OFF)。
MIDFILE	FOLDER で指定したフォルダの中の XG サポートデータ。
BLKFILE_001	FOLDER で指定したフォルダの中のバルクファイル。 ネクストレジストを使う場合は 002、003... と増えていく。

したがって、現在編集中の ELS\_SONG.txt の中で、PART\_XG を OFF から PLAY に、MIDFILE を XGS.MID に書き換えれば良い。参考として、書き換えた後の ELS\_SONG の例を以下のソース 2 に示す。

ソース 2: ELS\_SONG.NAM (変更後)

```

1  SOOX:SONGNAME = SONG_00X
2  SOOX:FOLDER  = SONG_00X
3  SOOX:SECURITY = OFF
4  SOOX:MODEL   =
5  SOOX:PART_UK = OFF
6  SOOX:PART_LK = OFF
7  SOOX:PART_PK = OFF
8  SOOX:PART_LEAD = OFF
9  SOOX:PART_KBP = OFF
10 SOOX:PART_CTRL = PLAY
11 SOOX:PART_XG  = PLAY
12 SOOX:MIDFILE  = XGS.MID
13 SOOX:BLKFILE_001 = REG_001.B00

```

PART\_XG と MIDFILE のみをソース 2 のように書き換えたら保存し、拡張子を.txt から.NAM に戻そう。

もちろん、XG サポートを EL モードで保存していた場合は、Ch.1 から 4 は EL ボイスで再生できる。それを実際に再生させたいときは、同様に、PART\_UK などを PLAY にすることで再生できる。

以上で、XG サポートの製作の全工程は終了である。エレクトーンの MDR で再生して、不具合がないか確認しよう。

#### 3.2.4.4 EL-Explorer を使った XG サポートファイルの転送

上記のように、format 0 の SMF を USB メモリに保存した後、.NAM ファイルの書き換えによってエレクトーンで再生できるようにするのが、XG サポートファイルの基本的な利用方法である。さて、このような USB メモリにおけるソング構造の編集を簡易に行うことができるアプリケーションとして、すずとも氏が「EL-Explorer」を開発している。[EL-Explorer の公式サイト](https://www.kamekyame.com/el/explorer)<sup>\*62</sup>からダウンロードできる。エレクトーンのパネルのように、Windows 上で直感的に XG サポートファイルの書き込みや、.NAM ファイルの編集などを行うことができる。上述の方法のうち、format 0 の SMF を生成した後、USB メモリに保存しに行くところからは EL-Explorer に任せることができる。ぜひ一度お試しください。

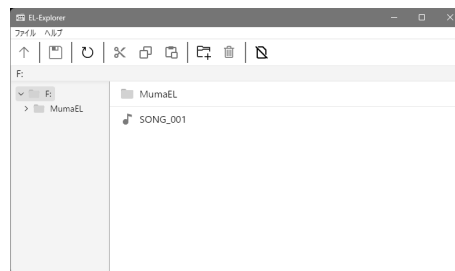


図 10: EL-Explorer の画面。EL-Explorer 公式サイトから引用。

### 3.3 XG 音源特有の仕様

以上で、XG サポートの基本的な作り方の解説は終了である。より高度なデータを作成したい読者のための情報を以下に記す。スタイルも XG 音源を使用するため、スタイルファイル編集においても以下の知識は有用である。

#### 3.3.1 システムエフェクトとインサクションエフェクト\*

EL ボイスと同じように、XG ボイスにもさまざまなエフェクトをかけることができる。

EL ボイスは、各ボイスにそれぞれ個別のエフェクトをかけることができた。このようなエフェクトのかけ方を**インサクションエフェクト**という。しかし XG においては EL ボイスの場合と異なり、ボイスそれぞれに個別のエフェクト設定をするのではなく、全てのチャンネルが 1 系統のエフェクトブロックを介してエフェクトをかけるという形をとる。このようなエフェクトのかけ方を**システムエフェクト**という。これらの違いを図 11 に示す。

XG ボイスにかけることができるエフェクトは、**リバーブ**、**コーラス**、**バリエーション**の 3 系統である。sysEX を多用してエフェクトの設定を行うため、sysEX は [System Setup] のトラックに書かなければならないことを強調しておく。

##### 3.3.1.1 リバーブエフェクト

**XG ボイスのリバーブブロックの設定は EL ボイスのリバーブ設定で行う**<sup>\*63</sup>。したがって、リバーブエフェクトはシステムエフェクトである。

各 XG ボイスは CC#91 Reverb Send Level で EL ボイスのリバーブブロックへのセンド量を指定できる。XG 仕様特有の設定は、エレクトーンでは無視されるものと思われる。

<sup>\*62</sup> <https://www.kamekyame.com/el/explorer>

<sup>\*63</sup> EL ボイスにおけるリバーブを思い出そう。本体パネル左上に存在するリバーブ音量を押すと、ディスプレイにリバーブ全体設定が現れる。これがエレクトーンにおけるリバーブブロックの設定である。各ボイスセクションはこのリバーブブロックへの音の送量（センド）を指定するという操作をするのであった。このように、EL ボイスにおけるリバーブは、もともとシステムエフェクトとして存在することがわかる。XG ボイスにおけるリバーブは、EL ボイスのリバーブブロックをそのまま使用するため、センド量を設定するだけで良い。

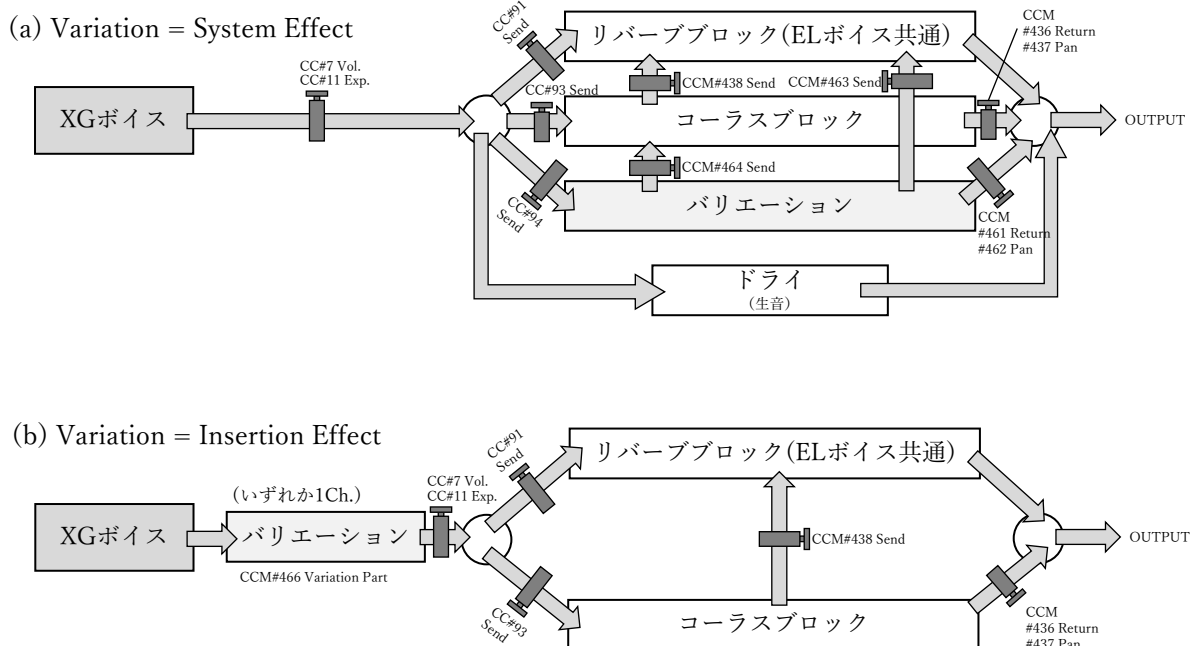


図 11: 2 種類のバリエーションエフェクトのかけ方。バリエーションを (a) システムエフェクトとして使う場合、(b) インサクションエフェクトとして使う場合。インサクションの場合は CC#94, CCM#461, 462, 463, 464 は無視される。

### 3.3.1.2 コーラスエフェクト

コーラスブロックはシステムエフェクトである。CCM#425 から CCM#444 の sysEX を用いてコーラストイプやコーラスのパラメータ設定をして、各チャンネルで CC#93 Chorus Send Level を書いてセンド量を指定する。

### 3.3.1.3 バリエーションエフェクト

バリエーションブロックはシステムエフェクトとインサクションエフェクトのどちらでも使用できる。図 11 にそれぞれの場合におけるエフェクトブロックの連結の概念図を示す。バリエーションブロックはエレクトーン全体で 1 基のみ存在し、これをシステムとして使うのか、インサクションとして使うのかを選ぶことができる。

システムエフェクトとして使う場合、エフェクトタイプの指定 (CCM#450)、システムエフェクトの設定 (CCM#465) をこの順番で行い、続いてバリエーションリターンレベルの設定 (CCM#461)、バリエーションパンの設定 (CCM#462)、バリエーションから他のシステムエフェクトへのセンド量の設定 (CCM#463, 464) をする<sup>\*64</sup>。この後、エフェクトをかけたいチャンネルの CC#94 Variation Send Level を 0 でない値に設定し、かけたくないチャンネルの CC#94 を 0 にする。システムエフェクトの場合は複数のチャンネルに渡って 1 種類のバリエーションエフェクトをかけることができる。

インサクションエフェクトとして使う場合、エフェクトタイプの指定 (CCM#450)、インサクションエフェ

<sup>\*64</sup> なお、エフェクトタイプの指定を行うと、そのエフェクトのデフォルトのパラメータ設定が読み込まれ、前のエフェクト設定は捨てられるという点に注意する。エフェクトのパラメータを設定する場合は CCM#451 から CCM#460、および CCM#474 から CCM#479 を用いる。各パラメータが何に対応する操作なのかは、まだ解析が終わっていない。この対応を明らかにするには、例えばパソコンで sysEX をリアルタイムで表示できるようにしておいて、エレクトーンで EL ボイスのエフェクトパラメータを弄ったときに送信されるメッセージを解析すれば良い。なお、一部のエフェクトに関しては海外向けエレクトーン ELA-1 やポータブルキーボード PSR-SX900 などのデータリストにパラメータの番地が書いてある [18][19]。

クトの設定 (CCM#465)、インサートするチャンネルの指定 (CCM#466) を、この順番で行う必要がある。システムとして使う場合に比べて記述する命令がシンプルで簡単である。ただし、**どれか 1 つのチャンネルにしかインサーションエフェクトを用いることができない**点に注意する。

### 3.3.2 メガボイス

メガボイスは XG ボイスの中で、**人間が演奏することを想定されていない特殊な音色マッピング**がされている高品位ボイスである。打ち込む際のノートナンバーとベロシティの両方で音色が大きく変わり、1 種類のボイスで多種多様な音色を出すことができる。図 12 から図 17 に ELS-02 シリーズの全メガボイスのマッピングを示す [14]。縦軸がベロシティ、横軸がノートナンバーに対応している。この図を見ながら実際に自分のエレクトーンでメガボイスを聞いてみると良い<sup>\*65</sup>。

---

<sup>\*65</sup> メガボイスは通常のボイスと異なり、プログラムチェンジのアドレス配置が系統分けされていない（通常のボイスは CC#00/32 でジャンル分けがされているが、メガボイスはそうではない）。そのため、音源定義ファイルにはプログラムチェンジの選択画面に「Mega Voice」というフォルダ (Map) を設け、メガボイスをこのフォルダにまとめて配置した。

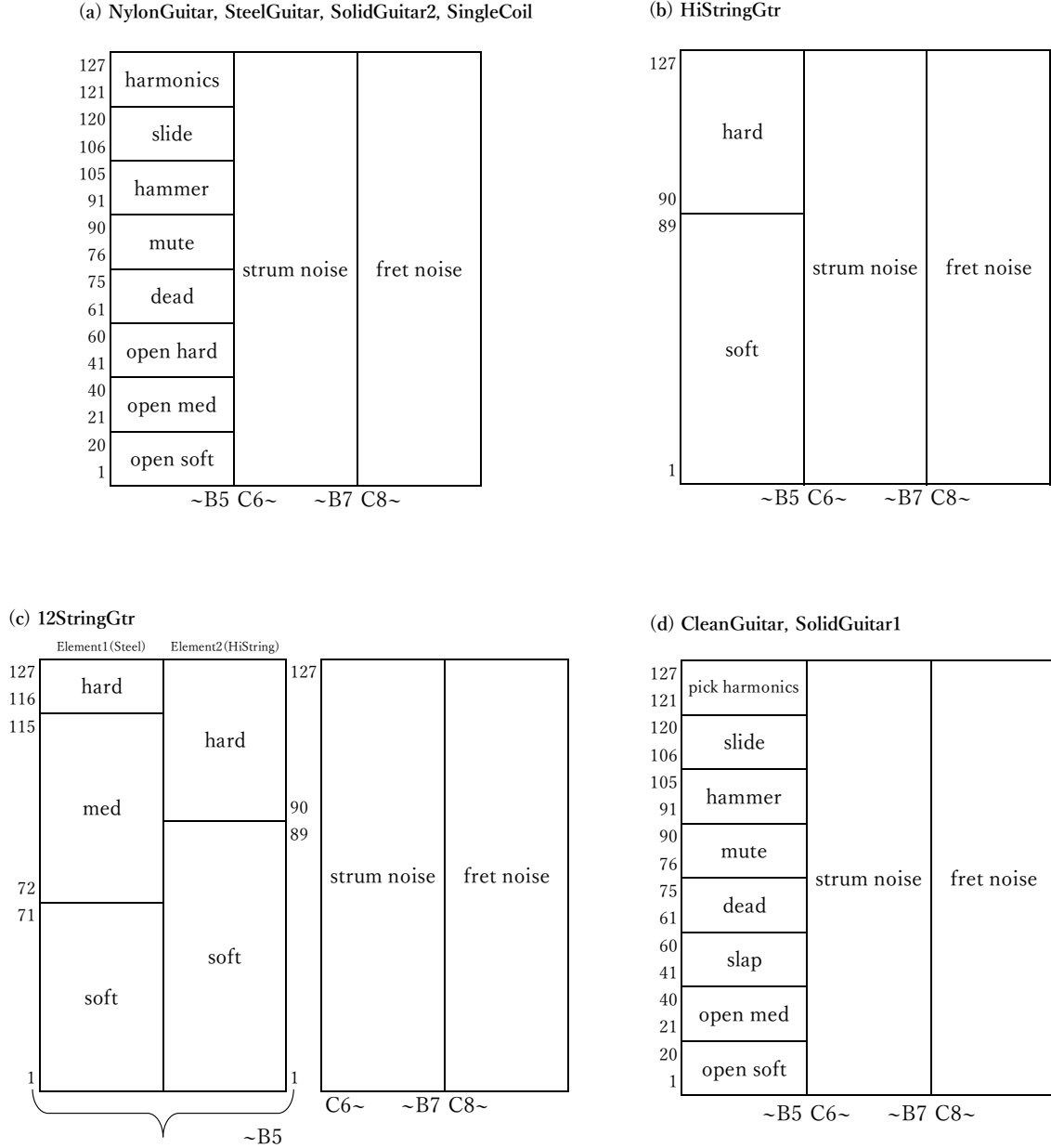
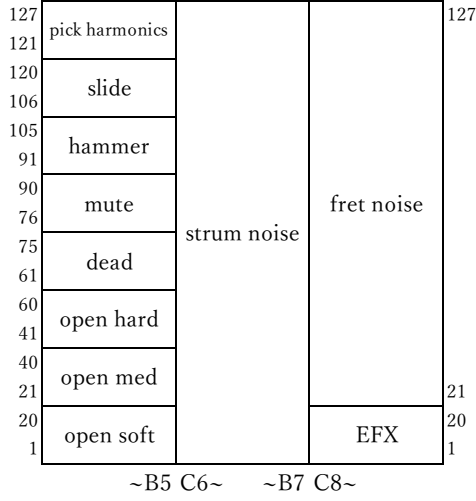
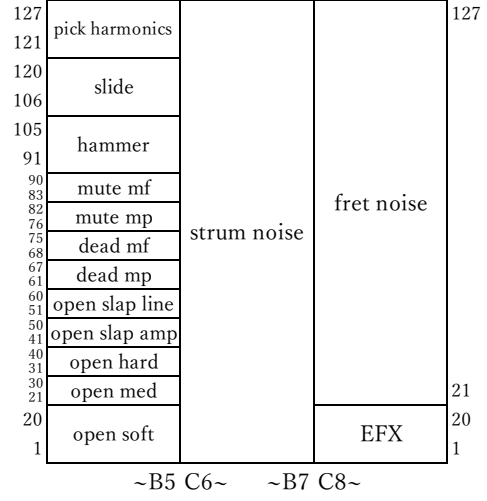


図 12: メガボイス、ギター類のボイスマッピング (1)。ボイス名の「Mega」は省略した。縦軸がペロシティで、横軸がノートナンバー。たとえば、Mega NylonGuitar の C4 をペロシティ 80 で演奏すると、mute 奏法が発音する。なお、(c) 12StringGtr の ~B5 は 2 つの元素で構成されており、それぞれが異なるペロシティマッピングになっている。

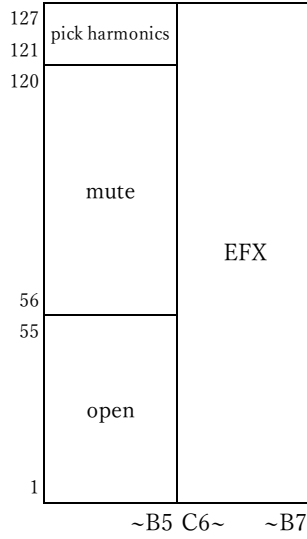
(e) FingerGtr, FingerSlapGtr, VintagePickGtr, VintageSlapGtr



(f) SlapAmpGtr



(g) OverdriveGtr, DistortionGtr



(h) JazzGuitar

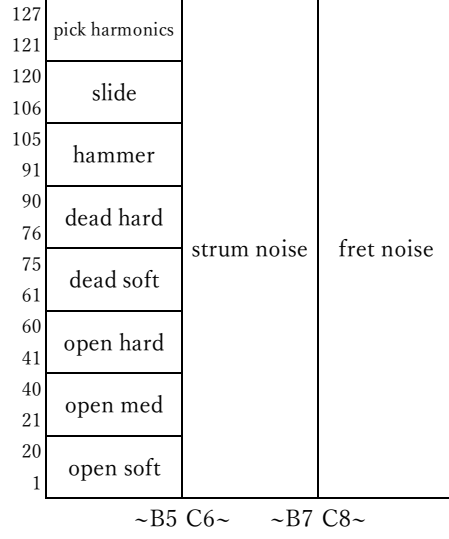
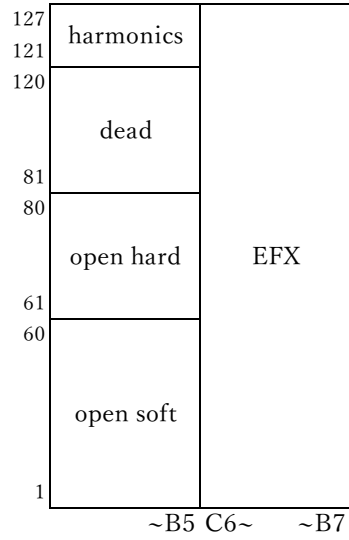
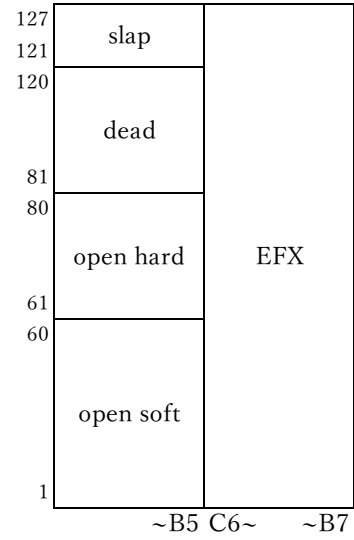


図 13: メガボイス、ギター類のボイスマッピング (2)。

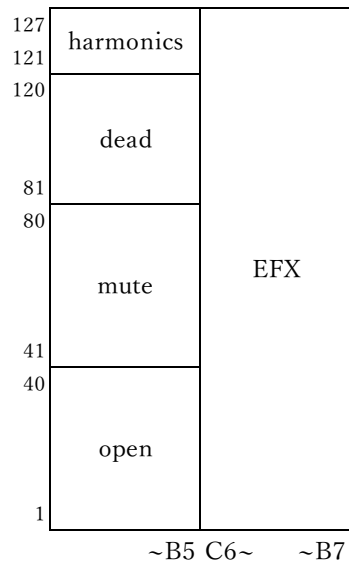
(a) AcousticBass, VintageRound, VintageFlat



(b) FingerBass



(c) PickBass, VintagePick



(d) FretlessBass

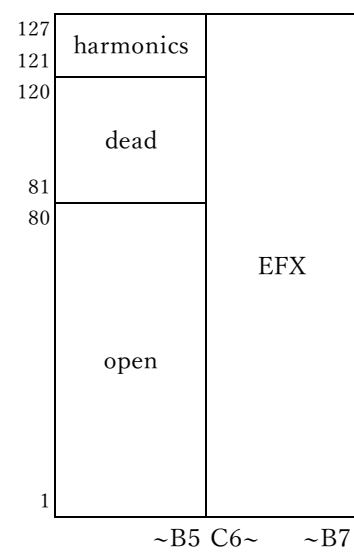


図 14: メガボイス、ベース類のボイスマッピング。



(a) PopHaa, PopHaa2, PopDaa

127	Waa vib f	breath noise	breath noise
111	Waa f		
110	Waa f		
101	Waa vib p		
100	Waa p		
91	Waa p	breath noise	breath noise
90	Waa p		
81	Waa p		
80	Aa vib f legato		
71	Aa vib f legato		
70	Voice vib f	breath noise	breath noise
61	Voice vib f		
60	Aa f legato		
51	Aa f legato		
50	Voice f		
41	Voice f	breath noise	breath noise
40	Aa vib p legato		
31	Aa vib p legato		
30	Voice vib p		
21	Voice vib p		
20	Aa p legato	breath noise	breath noise
11	Aa p legato		
10	Voice p		
1	Voice p		

~B5 C6~   ~D#6 E6~   B6

(b) PopHoo, PopHoo2, PopShoo, PopDoo

127	Yoo vib f	breath noise	breath noise
111	Yoo vib f		
110	Yoo f		
101	Yoo f		
100	Yoo vib p		
91	Yoo vib p	breath noise	breath noise
90	Yoo p		
81	Yoo p		
80	Oo vib f legato		
71	Oo vib f legato		
70	Voice vib f	breath noise	breath noise
61	Voice vib f		
60	Oo f legato		
51	Oo f legato		
50	Voice f		
41	Voice f	breath noise	breath noise
40	Oo vib p legato		
31	Oo vib p legato		
30	Voice vib p		
21	Voice vib p		
20	Oo p legato	breath noise	breath noise
11	Oo p legato		
10	Voice p		
1	Voice p		

~B5 C6~   ~D#6 E6~   B6

(c) PopHee, PopHee2, PopBee

127	Wee vib p	breath noise	breath noise
111	Wee vib p		
110	Wee vib p		
101	Wee vib p		
100	Wee vib p		
91	Wee vib p	breath noise	breath noise
90	Wee vib p		
81	Wee vib p		
80	Ee vib f legato		
71	Ee vib f legato		
70	Voice vib f	breath noise	breath noise
61	Voice vib f		
60	Ee f legato		
51	Ee f legato		
50	Voice f		
41	Voice f	breath noise	breath noise
40	Ee vib p legato		
31	Ee vib p legato		
30	Voice vib p		
21	Voice vib p		
20	Ee p legato	breath noise	breath noise
11	Ee p legato		
10	Voice p		
1	Voice p		

~B5 C6~   ~D#6 E6~   B6

(d) PopBaa

127	Yaa vib f	breath noise	breath noise
111	Yaa vib f		
110	Yaa f		
101	Yaa f		
100	Yaa vib p		
91	Yaa vib p	breath noise	breath noise
90	Yaa p		
81	Yaa p		
80	Aa vib f legato		
71	Aa vib f legato		
70	Voice vib f	breath noise	breath noise
61	Voice vib f		
60	Aa f legato		
51	Aa f legato		
50	Voice f		
41	Voice f	breath noise	breath noise
40	Aa vib p legato		
31	Aa vib p legato		
30	Voice vib p		
21	Voice vib p		
20	Aa p legato	breath noise	breath noise
11	Aa p legato		
10	Voice p		
1	Voice p		

~B5 C6~   ~D#6 E6~   B6

図 15: メガボイス、ポップボイスのボイスマッピング。Voice と斜体になっているところには、それぞれボイス名のスカットが入る。(c) のペロシティ 81 以上のマッピングに注意。

(a) Brass

127	gliss up
121	
120	falls fast f
111	
110	falls fast mf
101	
100	shake
91	
90	scoops
81	
80	attack
61	
60	f
41	
40	mf
21	
20	p
1	

~B5

(b) Trumpet

127	gliss up	valve noise	breath noise
121			
120	falls		
111			
110	shake		
101			
100	straight		
81			
80	legato		
61			
60	ff		
41			
40	f		
21			
20	mf		
1			

~B5 C6~    ~B7 C8~

(c) TenorSax

127	falls	key on noise	key off noise	breath noise
101				
100	growl			
81				
80	legato			
61				
60	f			
41				
40	mf			
21				
20	mp			
1				

~B5 C6~    ~B6 C7~    ~B7 C8~

図 16: メガボイス、ホーンズセクションのボイスマッピング。

(a) MaleChoir

127	Aah mf legato
106	
105	Aah mf
91	
90	Aah p legato
76	
75	Aah p
61	
60	Ooh mf legto
46	
45	Ooh mf
31	
30	Ooh p legato
16	
15	Ooh p
1	

~B5

(b) GospelChoir

127	Aaa legato	Ad libs FX
106		
105	Aaa	
91		
90	Hey legato	
76		
75	Hey	
61		
60	Wow legato	
46		
45	Wow	
31		
30	Hmm legato	
16		
15	Hmm	
1		

~B5 C6~

(c) SmallStrings, ClassicalStrings,  
LargeStrings, RealStrings

127	gliss down
121	
120	tremolo
111	
110	spicato ff
96	
95	spicato f
81	
80	legato
61	
60	f
41	
40	mf
21	
20	p
1	

~B5

図 17: メガボイス、クワイアとストリングスのボイスマッピング。

## 4 スタイルファイルの作り方（ドラム）

ヤマハは自動伴奏のことを**スタイル**と呼んでいる。スタイルはリズムだけを鳴らしたり、ベースやコードバックなどの伴奏パートをリズムと一緒に鳴らしたりするのに使う [21]。ELS-02 シリーズの場合、「スタイル」という名前は表に出てこないが、「リズム」というパネルで操作するものがそれにあたり【1-1】、本体画面右側中央の「リズムパターンプログラム」ボタンでスタイルを編集できる。

このように、エレクトーン本体で編集できるスタイルではあるが、**エレクトーン本体での編集の不便さの改善と、本体での編集不可能なスタイルのパラメータの制御**を目的に、本稿では**スタイルをパソコンで編集する方法**を解説する。

実は、ヤマハはスタイルを **SFF（スタイルファイルフォーマット）** という体系で製作しているため、ヤマハの電子楽器どうしはスタイルに互換性があり、エレクトーンにも他楽器のスタイルを読み込むための機能がある [22]。実際にポータートーンや Tyros などのヤマハ製キーボードにはスタイルクリエイターが内蔵されており、これで製作したスタイルをエレクトーンに読み込むことができる。そのため、パソコンで自作したスタイルファイルであっても、SFF に則っていれば、エレクトーンに読み込むことができるのである。

なお、SFF の仕様は公式では非公開であり、紹介する情報は有志の方々の先行研究によるものが多い。本稿の内容を利用するときは、あくまでも自己責任で実行すること。

### —— スタイルファイルフォーマットの参考書 ——

- スタイル入門講座 [15]  
エレクトーンに合わせてスタイルの解説を行っており、スタイルに関して日本でトップクラスの情報量を持つサイト。参考にしたページには、【章-ページ番号】でリンクを貼る。
- Style Files - Description [16]  
スタイルファイルに関して一番詳しく解説されていると思う。英語だが、非常にわかりやすい。細かい仕様に関する情報はこちらで勉強できる。
- YAMAHA Keyboard - Style CASM Section Format [17]  
CASM セクションを編集できるフリーソフトである CASM Editor の作者、Jørgen Sørensen 氏による CASM セクションの解説。端的にまとまっている。しかし、肝心の CASM Editor の方は、おそらく ELS シリーズのデータに対応しておらず、残念ながら CASM セクションの編集に使うことができないものと思われる。
- ヤマハ PSR-SX600 Refernce Manual [19]  
スタイルファイルフォーマット準拠の楽器であり、エレクトーンには存在しないスタイルクリエイターを搭載しているポータートーンの取扱説明書。第 2 章がスタイルの解説になっている。図表が多くわかりやすい。

### 4.1 EL Data Analyzer

以下、スタイルファイルの構造や、具体的な作成方法を 2 つの章に渡って解説するが、初めてスタイルファイルに触れる読者には、スタイルファイルの具体例があったほうが見通しが良いと思う。そこで、エレクトーンのデータからスタイルを切り出すソフトウェアである、**EL Data Analyzer** を紹介する（図 18）。[wakmin の github](https://github.com/wakmin-oxo/ELDataAnalyzer)<sup>\*66</sup>からダウンロードできる。以下のようにして使う。

<sup>\*66</sup> <https://github.com/wakmin-oxo/ELDataAnalyzer>

1. エレクトーンでユーザーリズムを作成する。
2. ユーザーリズムが入っているレジストデータ（.B00 ファイル）を読み込む（ドラッグアンドドロップで良い）。
3. 取り出したいユーザーリズムを選択し（もしくは空でないユーザーリズムを全て）、任意のフォルダに取り出す。
4. Domino で開くのであれば、拡張子を.sty から.mid に変更する<sup>\*67</sup>。

自作のスタイルはもちろん、エレクトーン内にもとから入っているスタイルに関しても、一度ユーザーリズムに保存することで取り出すことができ、スタイルがどのような構造を持っているかの勉強に役立つ。以降、スタイルの仕様に関する説明をするので、エレクトーン内のスタイルや、エレクトーン上で自作したスタイルなどを EL Data Analyzer を使ってパソコン上に取り込んで、拡張子を.sty から.mid に変更してから Domino で開くと良い。ただし、このようにして取り出したスタイルの配布はしてはいけない。

## 4.2 スタイルファイルの構造

スタイルファイルには MIDI 情報を書き込む **MIDI セクション**と、自動伴奏の規則のための **CASM セクション**という 2 つのセクションがある [16]。なお、セクションという用語はエレクトーンでは Main A、Main B などの再生箇所を指定するが、本稿では「セクション」が意味する語が多いため、エレクトーンにおける Main A などのセクションを「**セクションパターン**」と呼び、スタイルファイルの構造である MIDI セクション、CASM セクションと明確に区別する<sup>\*68</sup>。

スタイルの構成に CASM セクションは必要不可欠である。しかし、CASM セクションは複雑で、初学者には少々ハードルが高い。本来的には、MIDI セクションと CASM セクションの両方を記述してスタイルファイルを作る必要があるのだが、エレクトーンの場合は**条件付き**<sup>\*69</sup> CASM セクションの記述の必要がない。そこで**本章では、CASM セクションを編集せずに作ることができるスタイルファイル（メインドラム・アドドラムのみ）を紹介し、MIDI セクション・CASM セクションという 2 つのセクションをあまり意識せずとも簡単なスタイルファイルが作成できるように解説する。**アカンパニメントなどの自動伴奏を含めたスタイルファイルを作成するにはこれらの理解が必要があるため、次章で自動伴奏を作成する方法を解説するときに、



図 18: EL Data Analyzer の画面。

<sup>\*67</sup> 拡張子を.sty から.mid に変更することで、スタイルファイルのうち、MIDI セクションの部分だけを Domino で開くことができる。スタイルファイルは MIDI セクションと CASM セクションが合わさって構成されているが、Domino は CASM セクションを取り扱うことができないため、Domino でそのスタイルファイルを保存すると、CASM セクションが無くなる点に注意してはならない。切り出したスタイルファイルのうち、MIDI セクションだけを変更し、CASM セクションは元のまま使いたい場合、変更後の MIDI セクションに変更前のスタイルファイルの CASM セクションをコピーするという作業が必要になる。そのような CASM セクションの作業には、次の章で説明する CasmEdit を使用するのが良い。

<sup>\*68</sup> なお、本稿でセクションパターンと呼ぶものは、データ上は CSEG と記述されており [16]、チャンネルセグメントの略であると思われる。

<sup>\*69</sup> CASM セクションが無いスタイルをエレクトーンに読み込んだ場合、エレクトーンのデフォルトの CASM セクションが、読み込まれたスタイルファイルに結合される。従って、このデフォルト CASM セクションに沿って製作していれば、自分で CASM セクションを書く必要がない。次章にてアカンパニメントと併せて詳しく解説する。

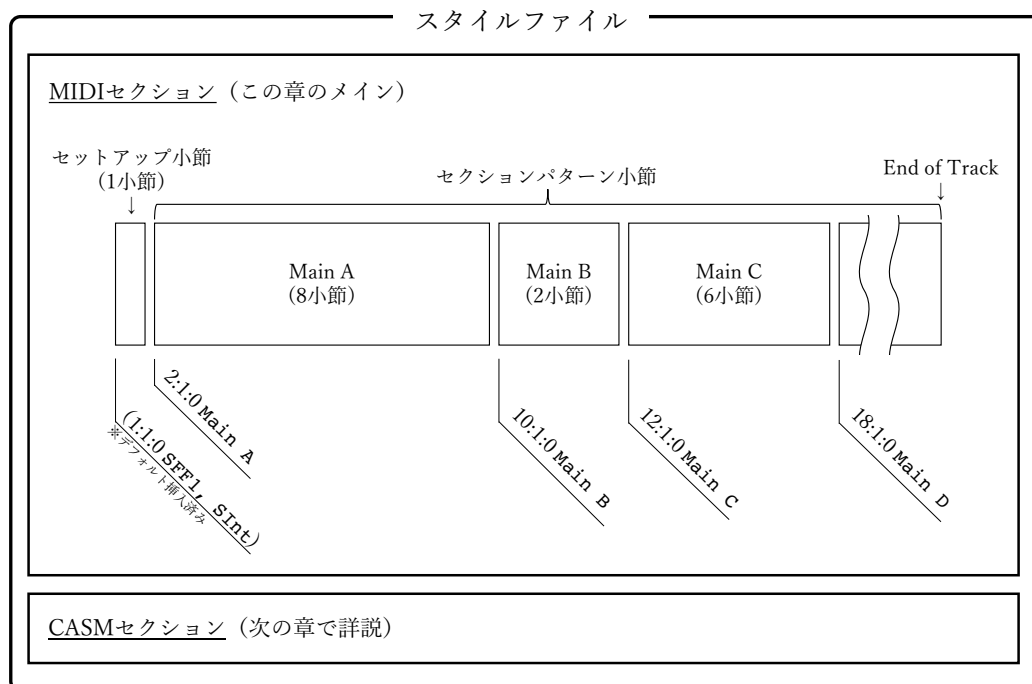


図 19: スタイルファイルの構造と、MIDI セクションの構造の例。各セクションパターンはマーカーによって分割される。例えば、Main A を MIDI セクションの 2 小節目から 8 小節目間作り、その後 Main B に移る場合は、2 小節目の頭に「Main A」と書き、10 小節目の頭に「Main B」と書く。これにより、2 小節目から 10 小節目が Main A として指定される。

改めて詳細に解説する。

MIDI セクションは、1 小節目のセットアップ小節と、2 小節目以降の Main A, B,...などのセクションパターン小節の 2 つに分かれていて、セクションパターン小節はマーカー<sup>\*70</sup>によってさらに各セクションパターンに分割される。図 19 にセクションパターンの分割の一例を示す。

MIDI セクションの時刻 1:1:0 には SFF1 と SInt という 2 つのマーカーを記述する<sup>\*71</sup>。これらはスタイルファイルの仕様上必要である。

### 4.3 メイン・アドラムの打ち込み：MIDI セクションを作る

いよいよ、スタイルファイルの MIDI セクションの作り方を、ドラムの編集を通して解説していく。以降、エレクトーン用音源定義ファイルを導入しているものとして解説する。この音源定義ファイルを導入した状態で Domino プロジェクトを新規作成すると、スタイルファイルを作成するための基本的なデータが初めから打ち込まれている。まずはプロジェクトを新規作成しよう。

<sup>\*70</sup> マーカーは、任意の Tick 数に任意の文字列を挿入できる SMF の機能である。Domino であればメニューバーの「マーク (M)」→「マークの挿入 (A)...」を選ぶか、Ctrl + M でマーカーを挿入できる。

<sup>\*71</sup> SFF1 はスタイルファイルフォーマット 1 ということを示すと思われる。Tyros などの新しい SFF 楽器においては SFF2 が採用されていることがわかっている [16]。

### 4.3.1 拍子の指定と Main A マーカーの挿入

まずは1小節目、セットアップ小節の編集である。MIDI セクションの編集で、最初に行わなくてはならないのは**拍子の指定**である。[Conductor] トラックを表示し、イベントリストの拍子指定を、作成したいスタイルの拍子に合わせて変更しよう（4/4 拍子の場合は変更の必要はない）。

4/4 拍子以外の拍子にした場合は、2 小節目の開始時刻がデフォルトデータからズレる。そのため、デフォルトで入っていた「Main A」マーカーが使用できなくなるので、このマーカーを削除し、自分で演奏線を 2:1:0 に合わせてから「Ctrl+M」で「Main A」を挿入しよう。拍子の変更をしなかった場合は、この作業は必要なく、デフォルトで入っている「Main A」マーカーをそのまま使用できる。

### 4.3.2 システムセットアップ\*

Mea	Bt	Tick	Step	Event	Gate	Vel/Value
1	1	0	0	-----System Setup-----		
1	1	0	0	[205] XG System On		
1	1	0	0	[1207] Part Mode	CH10	DRUMS1(Main Drum)
1	1	0	0	[1207] Part Mode	CH9	DRUMS2(Add Drum)
1	1	0	0	[465] Variation Connection		System
1	1	0	1920	[450] Variation Type		DISTORTION
2	1	0	0	-----End of System Setup-----		
2	1	0	---	End of Track		

図 20: デフォルトの [System Setup] トラックの中身。

次に、[System Setup] トラックを選択してシステムセットアップの設定を確認しよう。システムセットアップでは主に、**バリエーションエフェクトの設定**と**ドラムセットアップの設定**とができる。これらを設定しないのであれば、**この節をスキップして、音源定義ファイルのデフォルトのままスタイルファイルを製作できる**。

デフォルトでは図 20 のように、[205 XG System On]、[1207 Part Mode]、[465 Variation Connection]、[450 Variation Type] を書いておいた。以下でこれらの命令を一つずつ説明する。

#### [205 XG System On]

エレクトーンを XG モードに変更する命令。

#### [1207 Part Mode]

指定したチャンネルを**ドラムトラック**として使用するという命令。これを設定することにより、ドラムトラック特有の特殊な設定である**ドラムセットアップ**が使用できるようになる。音源定義ファイルのデフォルトでは、Ch.10 をメインドラム（DrumS1）に、Ch.9 をアドドラム（DrumS2）に設定してある。

#### [465 Variation Connection]

バリエーションエフェクトをシステムで使うか、インサクションで使うかを指定する命令。バリエーションエフェクトについては前章を参照してほしい。スタイルは仕様上**バリエーションエフェクトは必ずシステムエフェクトとして使用することになっている**。そのため、ここでシステムエフェクトとして使うという命令を記述した。

#### [450 Variation Type]

バリエーションエフェクトのエフェクトタイプの指定をする命令。デフォルトではディストーションを指定した。他のエフェクトを使用したい場合はここを変更すると良い。

なお、システムセットアップは1小節目でしか行うことができない。そのため、例えば Main A と Main B で異なるドラムセットアップをすることはできない。ドラムセットアップを変更する場合は別のスタイルファイルを作成し、エレクトーンに読み込むときには異なるユーザーリズムとして保存することになる。



### 4.3.3 チャンネルセットアップ

Mea	Bt	Tick	Step	Event	Gate	Vel/Value
1	1	0	0	-----Channel Setup-----		
1	1	0	0	PC:Standard Kit 1	27/0	1
1	1	0	0	[007] Volume		100
1	1	0	0	[010] Panpot		C
1	1	0	0	[011] Expression		127
1	1	0	0	[091] Reverb Send Level		40
1	1	0	0	[093] Chorus Send Level		0
1	1	0	1920	[094] Variation Send Level		0
2	1	0	0	-----End of Channel Setup-----		
2	1	0	----	End of Track		

図 21: デフォルトのドラムトラックの中身。

次に、メインドラム・アドドラムのチャンネルセットアップを行う。メインドラムもアドドラムも作業は同じであるため、ここではメインドラムの解説のみ行う。

トラックセレクトペインから [Main Drums]<sup>\*72</sup>を選ぶ。そうすると、図 21 のように、このトラックに書かれているデフォルトのチャンネルセットアップが確認できる。

デフォルトではプログラムチェンジがスタンダードキットになっているが、他のドラムキットを選びたい場合は「PC:Standard Kit 1」をダブルクリックすることで変更できる。なお、キットを変更するとピアノロールに表示されているドラムマップが更新される。

また、XG サポートを製作する時と同様に、他のチャンネルセットアップも変更できる。1 小節目に書いたチャンネルセットアップは、セクションパターンが呼ばれるたびに読み込まれる設定になるため、セクションパターン全てに共通する設定を書くのが良い。

ちなみに、プログラムチェンジを選ぶ際、「音色リスト」から選べば、エレクトーンに内蔵されている全ての XG ボイスの中からドラムの音色を選択できる。これにより、ドラムを XG サポートのような完全自動演奏として活用できる。簡単かつ非常に強力なので、一度試してみると良い。

### 4.3.4 セクションパターン小節の打ち込み

上記のチャンネルセットアップをアドドラムにも行ったら、セットアップ小節で行うべき設定は終了である。続いて、2 小節目からメインドラムとアドドラムに Main A のドラムノートを打ち込もう。コントロールチェンジの多くはセクションパターン小節でも使える<sup>\*73</sup>ため、CC を使っているいろいろな効果を試してみるのも面白い。

ノートの打ち込みが完了したら、最後のノートが置かれた場所の、次の小節の頭に演奏線を合わせよう。スタイルファイルに Main A しか書かないのであれば、ここに End of Track を設定する<sup>\*74</sup>。

Main A 以降も作る場合は、次に作りたいセクションパターンのマーカー（表 4 参照）を設置することになる。空白は半角で、大文字・小文字に注意して入力しよう。これらのセクションパターンはスタイルファイルにどのような順番で書いても構わないし、使わないセクションパターンについてはマーカーを設置する必要もない<sup>\*75</sup>。

なお、エレクトーン上で編集する時と同じように、Fill In は 1 小節しか扱うことができない<sup>\*76</sup>。他のセクションパターンについては小節数の制限は無いが、エレクトーンのメモリが一杯になったら読み込めなくな

<sup>\*72</sup> デフォルトデータでは [Main Drums] トラックを Ch.10 にしており、基本的にメインドラムとして使用する。Ctab の設定をすれば、他のトラックでも構わない。

<sup>\*73</sup> セクションパターン小節で使うことができない MIDI メッセージについては、後ほど説明する。

<sup>\*74</sup> End of Track についての解説は、XG サポートの作り方にある。

<sup>\*75</sup> マーカーを使わなかったセクションパターンは、ノートが何も置かれていない状態としてエレクトーン内で補完される。

<sup>\*76</sup> Fill In に 2 小節以上書いても、最初の 1 小節だけ読み込まれて以降の小節は無視される。

表 4: セクションパターンを指定するマーカー一覧 [16]。Fill In BA は BREAK。

Main A	Fill In AA	Intro A	Ending A	Fill In BA
Main B	Fill In BB	Intro B	Ending B	
Main C	Fill In CC	Intro C	Ending C	
Main D	Fill In DD			

るため【10-3】、極端に大きなサイズのセクションパターンは作るべきではない。大きくとも 64 小節程度に収まるように分割しよう\*77。

最後のセクションパターンまで打ち込みが終わったら、最後のノートの次の小節の頭に、End of Track を忘れずに配置する。これで、メインドラムとアドドラムのみのスタイル（の MIDI セクション）を打ち込むことができた。

#### 4.3.5 1 小節目と 2 小節目以降の違い

何度も述べているように、SMF の 1 小節目はセットアップ小節である。スタイルにおける 1 小節目は、各セクションパターンが呼び出されるたびに必ず呼び出される共通設定であり、この小節の設定は非常に重要である。また、表 5 のように、1 小節目でしか設定できない MIDI メッセージが存在する [16]\*78。

注意すべきなのは、NRPN/RPN や sysEX は 2 小節目以降で指定することができない点である。これにより、セクションパターンの中で特定のドラムのピッチを変更する（つまり、ドラムセットアップを変更する）ことはできない\*79。また、サスティンも無視されるため、注意してノートを打ち込む必要がある。

#### 4.3.6 スタイルファイルの保存とエレクトーンでの読み込み

メイン・アドドラムを打ち込んで、End of Track も設定し終わったら、このファイルを SMF として書き出す。XG サポートの時と同様に、メニューバーの「ファイル (F)」→「SMF 書き出し (M)...」から書き出そう。このとき、「Format 0」として SMF 書き出しすることに注意すること。

書き出しが終わったら、書き出されたファイルの拡張子を変更する。SMF の拡張子は「.MID」だが、SFF の拡張子は「.sty」であるため、拡張子を「.sty」としておこう。これで、エレクトーンに読み込ませるファイルの準備ができた。本来であればこの後に CASM セクションの編集を行うのだが、CASM セクションが無くとも自動的にエレクトーンで補完されるため、このままエレクトーンに読み込むことができる。

最後にこのファイルをエレクトーンで読み込もう。準備ができたスタイルファイルを USB メモリにコピーし、エレクトーンに挿す。あとは取扱説明書 [22]166 ページの通りに読み込めば良い。エレクトーンのリズムパターンプログラム画面の「ファイル」タブから、USB メモリに入っているパターンファイルを選択して読み

\*77 90 小節程度の大きなサイズのセクションパターンで作ったスタイルでシーケンスを組んで再生したところ、再生中にリズムの再生が異常停止した。

\*78 スタイルの途中の拍子変更ができないのはセクションパターンの格納されているデータが小節単位で定義されているためである【5-34】。しかしながら、2 小節目以降でもテンポ変更には対応しており、1 小節目で指定したテンポに対する倍率で、変更後のテンポを指定できる。これにより、レジストレーションメモリーを消費せずに *accel.* や *rit.* ができるほか、擬似的・一時的に変拍子を実装できる。なお、テンポ変更の仕様を確かめるには、次のような実験を行えば良い：MIDI セクションを作る際に、1 小節目（セットアップ小節）は 4/4 拍子でテンポ 100 としておいて、2 小節目から Main A を書き始める。Main A の 4 拍目だけをテンポ 200 とし、この 1 小節で Main A が終わるようにしてからエレクトーンに読み込ませれば、擬似的に 7/8 拍子を再現できる。この方法は、スタイルのセクションパターンの一部だけの拍子を変えたい場合などに活用できる。

\*79 メインドラム・アドドラムのどちらか一方を潰せば、ピッチを変更できる。すなわち、セットアップ小節で RPN を使用し、ピッチベンド感度を上げておいて、2 小節目以降の任意のタイミングで**ピッチベンドを書けば良い**。ただし、ピッチベンドはチャンネルメッセージであるため、それが書いてあるメインドラム・アドドラムの全てのノートのピッチが同時に上がる。特定の楽器のピッチだけを上げることはできない。

表 5: 扱することができる MIDI メッセージ [16]。

MIDI イベント	1 小節目	2 小節目以降
拍子設定	OK	-
システムメッセージ	OK	-
ノートオン・オフ	-	OK
プログラムチェンジ	OK	OK
ピッチベンド	OK	OK
CC#01(モジュレーション)	OK	OK
CC#06(データエントリー)	OK	-
CC#07(ボリューム)	OK	OK
CC#10(パン)	OK	OK
CC#11(エクスプレッション)	OK	OK
CC#64(サスティン)	OK	-
CC#71(レゾナンス)	OK	OK
CC#72(リリースタイム)	OK	-
CC#73(アタックタイム)	OK	-
CC#74(ブライツネス)	OK	OK
CC#91(リバーブセンドレベル)	OK	OK
CC#93(コーラスセンドレベル)	OK	OK
CC#94(バリエーションセンドレベル)	OK	OK
CC#98/99(NRPN)	OK	-
CC#100/101(RPN)	OK	-

込む。読み込んだだけでは保存されないため、「保存」タブから任意のユーザーリズムに保存しよう。以上でエレクトーンへの読み込みは終了である。作成したスタイルを再生し、意図通りに再生できるかチェックしたら、完成である。

## 4.4 ドラムセットアップの解説\*

ドラムは XG の仕様により**ドラムセットアップ**と呼ばれる特殊なエディットが可能である<sup>\*80</sup>。ドラムセットアップを使用することで、ドラム全体の調節のみならず、**キーに割り当てられている楽器ごとに音色をエディット**することができる。エレクトーン本体でも設定できるため、馴染み深いと思う読者も多いと思うが、エレクトーンからは触ることができないパラメータも多い。

ドラムセットアップを使うためには、特定のトラックを**ドラムトラック**として設定する必要がある。[CCM#1207 Part Mode] で指定トラックをドラムトラックにできる<sup>\*81</sup>。音源定義ファイルのデフォルトデータでは Ch.10 をメインドラム (DrumS1)、Ch.9 をアドドラム (DrumS2) として設定しておいた。

<sup>\*80</sup> ドラムセットアップの解説は、MIDI セクションの解説というよりは XG 仕様の解説となるため、本稿のどこで解説するか悩んだ。しかし、エレクトーンでは XG サポートでドラムセットアップをするのは稀だと考え、メイン・アドドラムの解説に多くを割いたこの章で解説することとした。

<sup>\*81</sup> なお、2000 年の XG 仕様書 v1.35[11] では XG 音源最高位モデルではドラムトラックは最大 4 まで使用できる。エレクトーンで 2 系統以上のドラムトラックが使用できるかは不明だが、おそらくできないと思われる。

エレクトーンでリズムパターンプログラムを使用してスタイルを製作する場合のキーの割り当てと、対応するノートナンバーを図 22 に示す。例えば、多くのキットで C1 はキックドラムに割り当てられており、これを指定するにはノートナンバーを 36 とすれば良い。

A#6	106	107	B6
G#6	104	105	A6
F#6	102	103	G6
	101	102	F6
D#6	99	100	E6
C#6	97	98	D6
	96	97	C6
A#5	94	95	B5
G#5	92	93	A5
F#5	90	91	G5
	89	90	F5
D#5	87	88	E5
C#5	85	86	D5
	84	85	C5
A#4	82	83	B4
G#4	80	81	A4
F#4	78	79	G4
	77	78	F4
D#4	75	76	E4
C#4	73	74	D4
	72	73	C4
A#3	70	71	B3
G#3	68	69	A3
F#3	66	67	G3
	65	66	F3
D#3	63	64	E3
C#3	61	62	D3
	60	61	C3
A#2	58	59	B2
G#2	56	57	A2
F#2	54	55	G2
	53	54	F2
D#2	51	52	E2
C#2	49	50	D2
	48	49	C2
A#1	46	47	B1
G#1	44	45	A1
F#1	42	43	G1
	41	42	F1
D#1	39	40	E1
C#1	37	38	D1
	36	37	C1
A#0	34	35	B0
G#0	32	33	A0
F#0	30	31	G0
	29	30	F0
D#0	27	28	E0
C#0	25	26	D0
	24	25	C0
A#-1	22	23	B-1
G#-1	20	21	A-1
F#-1	18	19	G-1
	17	18	F-1
D#-1	15	16	E-1
C#-1	13	14	D-1
	12	13	C-1

#### 4.4.1 ドラムセットアップ (NRPN)\*

例えば、メインドラムの C1 キックのボリュームを調節したい場合には、[Main Drums] (Ch.10) のチャンネルセットアップに [CCM#174 Drum Level] を追加し、Value として音量を、Gate として 36 を指定する。

ローパスフィルターのカットオフ周波数。下げれば高音成分が減る。

カットオフ周波数周りのレゾナンスをつける。

アタックタイムを増減する。減らすとアタックが遅くなり、増やすと早くなる。

ディケイタイムを増減する。減らすとディケイが遅くなり、増やすと早くなる。

ピッチを半音単位で調節できる。

ピッチを 100 分の 1 半音単位で調節できる。

40

#### CCM#174 Drum Level

音量を調節できる。

#### CCM#175 Drum Pan

パンを調節できる。XG 仕様書 [11] の 12 ページによると、最低値に設定するとパンが Random になるはずだが、エレクトーンでは再現できない。

#### CCM#176 Drum Reverb Send Level

CC#91 リバースを最大値として、そこにどの程度の量をセンドするかを決める。デフォルトはノートによって違う。

#### CCM#177 Drum Chorus Send Level

CC#93 コーラスを最大値として、そこにどの程度の量をセンドするかを決める。デフォルトはノートによって違う。

#### CCM#178 Drum Variation Send Level

CC#95 バリエーションを最大値として、そこにどの程度の量をセンドするかを決める。デフォルトは基本的に 127。

### 4.4.2 ドラムセットアップ (sysEX)\*

ドラムセットアップの命令は syxEX にも存在する。重複を除けば、NRPN よりも特殊で限定的な命令が多い。

以下に sysEX で指定できるドラムセットアップ命令を列挙する。ただし、NRPN で設定できる項目については省略する。NRPN と同様に CCM が存在するため、音源定義ファイルを使う場合に合わせて CCM# を付記する。NRPN の場合と同様に Domino で打ち込む場合は Gate がノートナンバーに対応する。

例えば、メインドラムの C#2 クラッシュシンバルでシンバルミュート奏法を再現したい場合は、[System Setup] トラックの 1 小節目に [CCM#909 Drum Rcv Note Off] を追加し、Value として 1 を、Gate として D1-49 を指定する。

#### CCM#903 ALTERNATE GROUP

ALTERNATE GROUP が同じであるような複数のノートは同時に発音できないようになる。ハイハットのオープンとクローズのように、片方が鳴ったときにもう片方をミュートしたいときなどに使う。デフォルトの ALTERNATE GROUP を知りたい場合は XG 仕様書 [11] を参考にしないといけないが、後発のエレクトーンのドラムキットが載っていないので、もし ALTERNATE GROUP を操作するのならば自分で調査するしかない<sup>\*83</sup>。通常のノートは ALTERNATE GROUP はオフになっている。

#### CCM#908 KEY ASSIGN

**この命令は、筆者の実験においては、エレクトーンで使うことができなかった<sup>\*84</sup>。**以下の説明は KEY ASSIGN の一般的な説明である。もし KEY ASSIGN を使いたいのであれば、代わりに Rev NOTE OFF を ON にして、打ち込みの段階でゲートタイムを調節して打ち込めば良い。

ドラムに限らない XG 音源の仕様として、前の音が減衰している間に次の音のノートオンが来たとき、前の音をミュートする（シングル）か減衰を続けるか（マルチ）を選ぶことができ、それは XG MULTI PART の設定の SAME NOTE NUMBER KEY ON ASSIGN (CCM#606) という命令で制御できるようになっている。XG のデフォルトでは全ての音が減衰を続ける SAME NOTE NUMBER KEY ON

<sup>\*83</sup> 海外向けエレクトーン ELA-1 やポータブルキーボード PSR-SX900 などのデータリストに、一部の ELS-02 シリーズのキットの ALTERNATE GROUP が掲載されている。完全に互換するかは不明だが、参考になるかもしれない [18][19]。

<sup>\*84</sup> XG 仕様書 [11] によると、KEY ASSIGN はオプションではないはずなので、エレクトーンでも設定変更ができると考えられる。成功した方は連絡をください。

ASSIGN=MULTI に設定されているが、ドラムパートに関してはノートごとに設定を変更できるようになっており、その場合は SAME NOTE NUMBER KEY ON ASSIGN=INST とした上で、KEY ASSIGN を使って各ノートの設定をすることになる。

#### **CCM#909 Rcv NOTE OFF**

ノートオフを無視するかどうかの設定。Rcv は Receive のこと。クラッシュシンバルなどの自然な減衰に任せて消音する楽器はノートオフを無視して鳴り続けるが、サンバホイッスルなどは鳴り続ける時間（ゲートタイム）を自分で設定でき、ノートオフで消音する。クラッシュシンバルなどを ON にすることで、自分の好きなタイミングで消音させることができる。**シンバルミュートがないエレクトーンにおいて、非常に実用的な命令。**

#### **CCM#910 Rcv NOTE ON**

ノートオンを無視するかどうかの設定。これが ON のとき、ノートは鳴らない。デフォルトでは全て OFF。

#### **CCM#914 EG DECAY1**

#### **CCM#915 EG DECAY2**

XG ボイスにはディケイタイムが 2 種類存在するものがある。NRPN による設定ではその 2 つを同時に変化させるが、この sysEX を使うことで別々に設定できる。



## 5 スタイルファイルの作り方（アカンパニメント）\*

前の章では、メイン・アドドラムを Domino で打ち込む方法を解説した。多くのアレンジでは前の章の内容で十分であると考えているが、更に「強い」アレンジを志す人は、アカンパニメントを自分で編集したいと考えたことがあるだろう。そこでこの章では「プレイヤーが演奏しているコードを検出して、伴奏パターンを変更する」機能であるアカンパニメント<sup>\*85\*86\*87</sup>をスタイルファイルに入れる方法について解説する。

アカンパニメントでは、演奏されているコードに対応して、打ち込んでおいた MIDI の発音内容が変わる。この「MIDI のリアルタイム読み替え」をするためのルールが記述されているのが **Channel Table(Ctab)** である。また、Ctab が書かれているスタイルファイルの部分を **CASM セクション**と呼ぶ。図 23 は、MIDI セクションと CASM セクションがどのように関わっているかを示した概念図である。スタイルにおいて、打ち込んだ MIDI はそのまま再生されるわけではなく、必ず CASM セクションの Ctab を通り、Ctab の命令と演奏内容に従って、音の読み替えが行われて再生される。

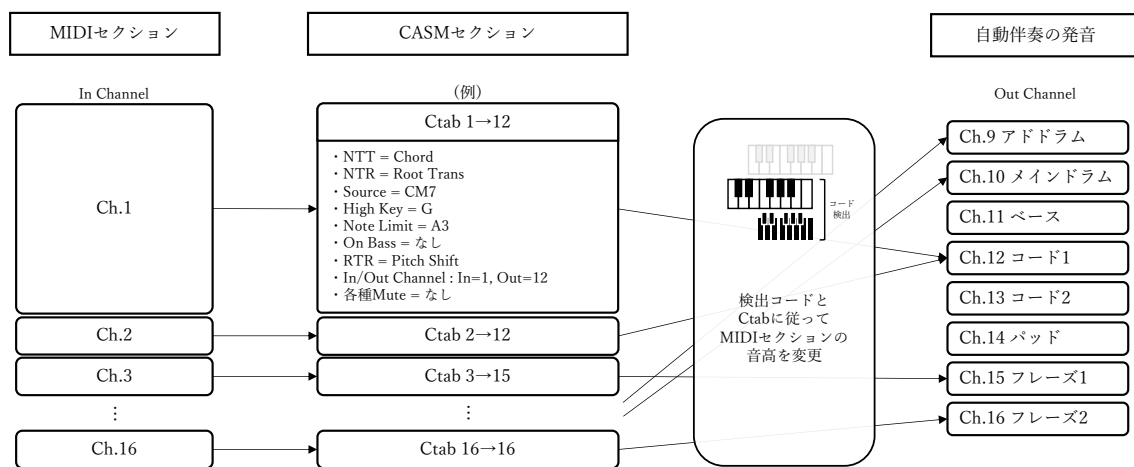


図 23: Ctab と CASM セクションの理解のための概念図。この図の Ctab に書かれている様々なルールは一例である。

実は、前の章で解説したメイン・アドドラムも例外ではない。MIDI セクションの Ch.9 と Ch.10 に打ち込まれたドラムは、エレクトーンで自動で補完されたデフォルトの CASM セクション（すなわち、「Ch.9, 10 は音高の読み替えをしない」という Ctab が書かれた CASM）によって、メイン・アドドラムとして読み込まれて再生されている。スタイルの MIDI セクションは、CASM セクションなしに再生することはできない。

さて、メイン・アドドラムは打ち込んだ内容がそのまま再生されればよかったため、デフォルトの CASM で

\*85 「アカンパニメント」という用語は混乱しており、本来的には、「コード 1/2」、「パッド」、「フレーズ 1/2」の 5 つを指す用語である [22]。しかし、プレイヤーの中ではベースの自動伴奏（オートベース）もいわゆる「アカンパニメント」として理解されていることが多い。そのため、本稿ではベースの自動演奏も、アカンパニメントと呼ぶことにする。

\*86 なお、オートベースコード (A.B.C.) はアカンパニメントにおけるコードの判定方式とオートベースの有無を指定するモードの名前であり、**シングルフィンガー**（下鍵盤を 1-3 音抑えるだけでコードを判定する。ベースの自動演奏をする）、**フィンガードコード**（下鍵盤のみでコードの判定を行う。ベースの自動演奏をする）、**カスタム A.B.C.**（下鍵盤と足鍵盤の両方でコードを判定する。ベースの自動演奏をする）、**A.B.C. なし**（下鍵盤のみでコードの判定を行う。ベースの自動演奏をしない）の 4 種類がある。

\*87 取扱説明書 [22] によると、**ローリズミック**もアカンパニメントの一種である。しかし、ローリズミックは**スタイルファイルフォーマットではない**ため、Ctab による指定ができない上に、コードの判定もしない。そのため、本稿では踏み入った解説を行わない。ちなみに、Domino などの MIDI シーケンサーを使ってローリズミックパターンを作りたい場合、コードパートにドなどの単音をローリズミックの音価で打ち込んでおいて、エレクトーンに読み込んでしまえば良い。ただしこのとき、打ち込んだ単音はスタイルとしてのコードで打ち込まれているため、エレクトーンでどのように演奏しても単音としてしか鳴らない。これをローリズミックにするため、エレクトーンのリズムパターンプログラム上でコードパートを少しだけ編集する（たとえば、あるノートの音価を 1 Tick だけ縮める）。そうしてから保存すれば、ローリズミックとして保存される [19-9]。

十分であった。しかし、それ以外の音高の読み替えを必要とするパートをスタイルに入れようとするなら、自分で CASM セクションを書かなくてはならない。そこでこの章では、読者がアカンパニメントによる自動伴奏を実現するために、Ctab を理解し、CASM セクションを自分で記述できるようになることを目標に解説する<sup>\*88</sup>。なお、メイン・アドドラムの編集方法については前の章まででほとんど解説し終えているため、この章ではアカンパニメントを主に扱う。

## 5.1 CasmEdit の導入

CASM セクションを編集するソフトウェアも複数存在するようであるが、本稿ではエレクトーンスタイルファイルを取り扱うことができる CasmEdit を使用する。CasmEdit の公式サイト<sup>\*89</sup>からダウンロードできる<sup>\*90</sup>。具体的な使い方は、後に解説する。

## 5.2 Ctab で設定できるパラメータ

Ctab の音高変換の仕組みを理解せずに、正しく再生できるアカンパニメントを作るのは難しい。そのため、アカンパニメントの MIDI セクションを打ち込む前に、まずは Ctab の命令を説明する。この節の解説を終えた後、具体的なスタイルファイルの作成方法の解説を行う。初めてアカンパニメントの自作に挑戦する人には難しいかもしれないので、その場合はこの節を一旦飛ばして、次の「作り方」から読むと良い。必要になったら適宜、この節に戻ってくれば良い。なお、スタイルクリエイター搭載楽器の取扱説明書には、公式の Ctab の解説が存在するため、そちらも合わせて参照すると良い [19]。Ctab の各パラメータには様々な呼び方があるが、基本的に本稿では見出し語として CasmEdit の呼称を採用する。

### 5.2.1 入力チャンネル・出力チャンネル

エレクトーンスタイルで使えるセクションパートは、メインドラム、アドドラム、ベース、コード 1/2、パッド、フレーズ 1/2 の 8 つである。しかし、MIDI セクションで取り扱えるチャンネルは 8 つではなく、最大 16 まで使用できる [17-6]。スタイルは、最大 16 チャンネル分の MIDI データを 8 パートに合流させて再生している。MIDI セクションで記述する最大 16 のチャンネルを入力チャンネルと呼び、スタイルとして最終的に再生するのに使用する 8 つのチャンネルを出力チャンネルと呼ぶ。出力チャンネルとして使用される 8 つのチャンネルと、それらを使用した際に再生されるセクションパートを表 6 に示す。

この仕様により、特定のコードの時に違うパターン（違う楽器にもできる<sup>\*91</sup>）を再生したり、ピアノの左手相当のノートと右手相当のノートを別々のチャンネルで編集し、再生を 1 パートでするといったことが可能になる。入力チャンネルが複雑な具体例として、エレクトーンの内蔵スタイルである「アンブラグド 1」などが

表 6: 出力チャンネル一覧。

チャンネル	パート
Ch.9	アドドラム
Ch.10	メインドラム
Ch.11	ベース
Ch.12	コード 1
Ch.13	コード 2
Ch.14	パッド
Ch.15	フレーズ 1
Ch.16	フレーズ 2

<sup>\*88</sup> [19] を読めばわかることだが、実はこの章で説明することのほとんどは公式の資料に解説がある。スタイルファイルの作り方の解説をしている章ではあるが、実際のところは「スタイル」の解説をしているに過ぎない。

<sup>\*89</sup> [https://www.mnppsaturn.ru/osenenko/Main\\_eng.htm](https://www.mnppsaturn.ru/osenenko/Main_eng.htm)

<sup>\*90</sup> CasmEdit のホームページは以前はもっと内容があったと記憶しているのだが、2023 年現在はソフトウェアが置かれているだけである。無くなってしまわないか不安である。

<sup>\*91</sup> 1 つのパートで同時に鳴らすことができる楽器（プログラムチェンジコマンド）は、当然 1 種類だけであるため、例えばコード 1 でピアノとストリングスとブラスを同時に鳴らす、ということはいくつかできない [17-9]。1 つの出力チャンネルで同時に 2 以上の楽器を鳴らすことはできないが、同時にできれば複数の楽器を 1 つの出力チャンネルに通すことができる。



ある。コード 1 の 1 出力に 13 チャンネル分のデータが入力されている【17-8】。

ちなみに、コード 1/2、パッド、フレーズ 1/2 のそれぞれは、名前は違うものの、機能としては同じであると思われる\*92。この 5 つのパートにおいては、どれを使っても同じ効果が得られるため、自分のアレンジに最も適した名前のパートを使っても良いし、雑にコード 1 から順番にアカンパニメントを埋めて行っても良い。ただし、「ベース」の Ch.11 は、EL ボイスである足鍵盤ボイスで発音されるため、注意する。

CasmEdit においては、入力チャンネルは「Tr」、出力チャンネルは「Ch」とか「MIDI Chan」とかと書かれている。

### 5.2.2 Source Chord Root / Type (C.R. / C.T.)

MIDI セクションで記述したパターンの基準となるコードを指定する【16-23】。Chord Root(C.R.) でコードのルートを指定し、Chord Type(C.T.) でタイプを指定する。デフォルトの CASM セクションの場合は [C.R.=C]、[C.T.=Maj7] となっている。すなわち、デフォルトでは基準のコードが  $C_{M7}$  と指定されている。

ほとんどの場合で、C.R. / C.T. はデフォルトの  $C_{M7}$  で事足りると思うので、**C.R. / C.T. は変更せずに、MIDI セクションに打ち込むフレーズは  $C_{M7}$  のときに正しく鳴るようなフレーズとすると良いだろう。**もちろん、C.R. / C.T. を変更して、MIDI セクションのフレーズをそのコード基準に書いても構わない。

指定できるコードタイプと、推奨ノートの一覧を図 24 に示す【19】。

#### ソースルートCの場合



図 24: C.R.=C のときのコードタイプと推奨ノート一覧【19】。

### 5.2.3 NTR

NTR とは Note Transposition Rule のことで、音高変換の際に展回形を用いるかを指定する。Root Trans と Root Fixed から選ぶ。

#### Root Trans

図 25 左のように、音高変換の際に転回しないで、元のパターンの音程を保とうとする。普通、メロディ

\*92 公式【22】では、それぞれのパートは次のように説明されている。コード 1/2:「コード伴奏を担当する、ピアノやギターなどの演奏を再生する」、パッド:「ストリングスやオルガンなどの演奏を再生する」、フレーズ 1/2:「アルペジオや、パンチの効いたプラスフレーズなどを再生する。アカンパニメントの中でひととき華やかなパート」

ラインがあるパートに Root Trans を設定する。

## Root Fixed

図 25 右のように、音高変換の際にできるだけ元の音域から離れないように、転回形を用いる。普通、コードパートに Root Fixed を設定する。また、ドラムパートも Root Fixed を設定する。

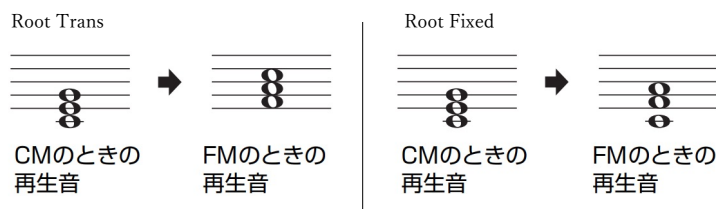


図 25: Root Trans と Root Fixed の違い。[19] より引用。

## 5.2.4 NTT

**NTT** とは Note Transposition Table のことで、演奏を検出したコードタイプに応じて、どのように MIDI セクションのデータを音高変換するかを指定する【16-26】。CasmEdit の「**NTT**」または「**Cntt**」という欄で指定する<sup>\*93\*94</sup>。通常のパートであれば「NTT」欄で指定できるが、特殊なスケールに沿って音高変換を行いたい場合と、ベースオンコード的な音高変換を行いたい場合は「Cntt」欄で指定する<sup>\*95</sup>【15-31】。以下に NTT と Cntt のオプションを解説する。

## BYPASS

[NTR=Root Fixed] ならば、一切の音高変換を行わない。メインドラム・アドドラムなどに適している<sup>\*96</sup>。効果音的に使いたい楽器に対しても Bypass を指定することがある<sup>\*97</sup>。[NTR=Root Trans] ならば、ルート音だけ音高変換を行う。

## MELODY

メロディライン向けの音高変換を行う。和音には適さないことが多い。メロディ的なフレーズや、ベースに適している。

## CHORD

和音向けの音高変換を行う。

## BASS

ベース向けの音高変換を行う。Cntt での Bass On 付きの「MELODY + Bass」と同じ【16-26】。公式では [NTT=BASS] を使わずに、[NTT=未指定] かつ [Cntt=Bass On の MELODY + Bass] とする。

<sup>\*93</sup> エレクトーンを含む新しい電子楽器では CASM セクション内の Ctab 以外に Cntt というセクションが追加され、Ctab 内の NTT よりも多彩なオプションを選ぶことができるように拡張された。なお、Cntt に対応している楽器では NTT の命令を Cntt で上書きする【16】。

<sup>\*94</sup> なお、エレクトーン内蔵のスタイルでは「NTT=未指定」かつ「Cntt=指定」または「NTT=指定」かつ「Cntt=未指定」のいずれかであるらしい。

<sup>\*95</sup> ベースパートには、Bass On 付きのものを選ぶ。Bass On 付きの物を選ぶと、オンコード（ベースが和音のルートを演奏しないようなコード）で演奏した場合にも、意図したベースを演奏するようになっている。例えば判定コードが  $C^{on E}$  のとき、On Bass が有効になっているパートでは E に集中して音高変換をする。

<sup>\*96</sup> エレクトーンではメインドラム・アドドラムは NTR/NTT の設定に関わらず音高変換をスルーする【7-1】。

<sup>\*97</sup> 例えば、メガボイスの効果音のレイヤーだけを使いたい場合。効果音のレイヤーは範囲が狭いため、普通の楽器のように安易に音高変換されると意図した効果を出せない場合がある。

### その他のスケール<sup>\*98</sup>

主にメジャーコード・マイナーコードの判定のみ行い、各スケールに合うように音高変換を行う。メジャーコード・マイナーコードのみ演奏されるセクションパターンであるイントロ・エンディングで主に用いる。詳しくは [19] を参照してほしい。

### 5.2.5 Highest Key (H.K.)

H.K. で指定されたノートよりも高い音をルートに持つ和音が演奏されたとき、和音全てを 1 オクターブ下げて再生する (図 26)。H.K. は [NTR=Root Trans] のときのみ有効になる。

#### 【例】ハイキーがFのとき

ルート変更 ➡	CM	C#M	...	FM	F#M	...
再生音 ➡	C3-E3-G3	C#3-E#3-G#3		F3-A3-C4	F#2-A#2-C#3	



図 26: Highest Key の機能説明 [19]。

### 5.2.6 Low Limit / High Limit (L.L. / H.L.)

最終的な発音に対する音高の限界を指定する。NTT、NTR、H.K. で音高変換された後のノートに Limit よりも外側の音が存在した場合、Limit 以内に入るようオクターブを変える [16-72]。

### 5.2.7 RTR

**RTR** とは ReTrigger Rule のことで、再生中に判定コードが変わった場合の処理を指定する [16-74]。演奏感に直結する重要なパラメータである。次の 5 種類がある。

#### Stop

発音中の音を止めて、次のノートオンを待つ。

#### Pitch Shift

発音中の音にピッチベンドをかけて、新しいコードにする。

#### Pitch Shift To Root

発音中の音にピッチベンドをかけて、新しいコードのルート音にする。

#### Retrigger

発音中の音を止めて、新しいコードの音で再発音する。

#### Retrigger To Root

発音中の音を止めて、新しいコードのルート音で再発音する。

エレクトーンの場合、カスタム A.B.C. でベースを弾き直すと、最初の 1 音は弾いたノートが演奏されるため、ベースが To Root か否かは関係ない。しかし、カスタム A.B.C. 以外のオートベースコードに対応するために、ベースは RTR=Pitch Shift To Root にするのが良い。

<sup>\*98</sup> MELODIC MINOR, MELODIC MINOR 5th VAR, HARMONIC MINOR, HARMONIC MINOR 5th VAR, NATURAL MINOR, NATURAL MINOR 5th VAR, DORIAN, DORIAN 5th VAR。

### 5.2.8 Auto Start

Auto Start が有効のパートはリズム再生と同時に自動で再生される。エレクトーンの場合、どのように設定してもメイン・アドドラムは Auto Start で再生され、それ以外に Auto Start は設定できない<sup>\*99</sup>ため、設定の必要はない。

### 5.2.9 Note Mute (NMute) / Chord Mute (CMute)

Mute は特定のコードが判定されたときにチャンネルをミュートしたり、逆に特定のコードのときだけ再生したりする場合に用いる。特定のコードタイプで音の濁りが生じるのを防ぐために使う。多くの場合は設定の必要はないだろうが、脚注<sup>\*100</sup>で示すように、ミュートを駆使して面白いスタイルを作ることもできる。

### 5.2.10 Editable Bit

本体で編集可能かを定めるフラグ。設定の必要はない。

## 5.3 アカンパニメントを含むスタイルファイルの作り方

いよいよ、アカンパニメントを含むスタイルファイルを作っていこう。まずは、前の章と同様に MIDI セクションの記述から始める。メイン・アドドラムは前の章で解説した方法で準備し、それ以外のセクションパートについて解説をする。

### 5.3.1 アカンパニメントパートの MIDI セクションの打ち込み

アカンパニメントパートの作り方は、基本的にドラムパートと同じである。前の章と同じように、Domino で新規プロジェクトを作成し、拍子の設定を行った後、1 小節目をセットアップ小節として使い、Main A マーカーをつけた 2 小節目からノートを打ち込んでいく。Main A の後は、次のマーカーを挿入するか、MIDI セクションの終わりであれば End of Track を挿入する。終わった後は、format 0 の SMF を書き出して、拡張子を「.mid」から「.sty」に変更する。

ドラムパートと異なるのは、**音高変換される前提でノートを打ち込む**という点である。復習であるが、Ctab の C.R. と C.T. で MIDI セクションのキーが決まり、具体的な音高変換のルールは NTR、NTT、H.K.、Limit で決まる。C.R. と C.T. では  $C_{M7}$  を指定することが多いので、MIDI セクションは  $C_{M7}$  の構成音で書くといいだろう。

#### 5.3.1.1 アカンパニメントパートの MIDI セクションの注意点

- ベースパートのアカンパニメントを書く際は、様々な点に注意する必要がある。まず、**エレクトーンで再生するベースパートは、足鍵盤ボイスで再生する。そのため、プログラムチェンジを使うことができない**。XG ボイスのベースを使いたい場合は、その他のセクションパートに入れるか、XG サポートに入れることになる。また、**アカンパニメントのベースパートは 1 オクターブ上で再生されてしまうため、オクターブを下げて保存する必要がある【16-12】**。ベースのフレーズを制御するのも簡単なことでなく、特にカスタム A.B.C. は L.K. とセットでコードを判定するため、意図したフレーズにならないことが多い。そのため、ベースパートのノートはルートと 5 度ぐらいにとどめておいて、アカンパニメ

<sup>\*99</sup> もし他のパートも Auto Start にすることができたら、ドラムを 3 以上編成できることになるが、私が実験した限りでは残念ながら Auto Start は無視される。

<sup>\*100</sup> 複数の入力チャンネルで 1 つの出力チャンネルを指定し、それぞれの入力チャンネルで異なるミュート設定にすれば、判定コードに合わせて複雑な分岐をするパートを作ることができる【11-20】。特定のコードの時だけアカンパニメントをプログラムチェンジを含めてまるっきり変更するといった積極的な使い方もできる【17-3】。

ントでは発音と休符の補助をさせ、フレーズは自分の足で演奏することをおすすめする。

- XG ボイスにバリエーションエフェクトを指定したい場合、**スタイルのバリエーションエフェクトはシステムエフェクトで固定されているため、図 11 の (a) のようにかける**。リズムでエフェクトを使用しているなら、それと同じエフェクトしかアカンパニメントにかけられない。
- セクションパターンのイントロとエンディングの再生中は、下鍵盤ボイスの発音が禁止される。また、鍵盤で押さえているコードは、イントロ・エンディングの再生中は変更されない前提で作る。イントロ・エンディングを作ろうと思っている方は、覚えておくと良いだろう。

### 5.3.2 Casm セクションの作成と編集



図 27: CasmEdit の最初のウィンドウ。スタイルファイルを開いた後の画面。メニューバーの「Create Casm Section」で Casm セクションを作ることができる。Casm セクションがある場合は、右上の「CASM」ボタンで次のウィンドウに遷移できる。

いよいよ Casm セクションの具体的な編集方法の解説に移る。まず、Domino で書き出した SMF（拡張子は「.sty」）を CasmEdit で開く（ドラッグアンドドロップで良い）。そうすると、その SMF の MIDI イベントのリストが表示される。ちなみに、この画面では Domino のイベントリストと同じように、MIDI データの直打ちを行うことでヤマハのスタイル楽器に特化したエディットが可能である。だが、本稿ではこのエディタは使用せず、Casm セクションの編集のみを行う。

Casm セクションの編集を行うには、当然ながら、ファイルに Casm セクションが存在しないとけない。しかし、Domino で作成された SMF には Casm セクションがないため、まず CasmEdit でデフォルトの Casm セクションをファイルに結合する必要がある。メニューバーの「Create」→「Create Casm Section F4」を選択し、新しく現れたウィンドウの「Create Casm」のボタンを押すと、Casm セクションが作られ、新たに Casm 編集のウィンドウ「Casm Editor...」が現れる。なお、この新しく現れたウィンドウを誤って消してしまった場合、最初のウィンドウの上部にある「CASM」ボタンを押すことでも呼び出すことができる（図 27）。

さて、Casm セクション編集ウィンドウの解説をする。Casm セクション編集ウィンドウでは、Ctab を編集できる。ウィンドウ下半分にあるリストの「Section」はセクションパターンのことで、「Tr」は入力チャン

ネルを示している。「Name」は英字8文字まででつけることができる [16] が、編集しないほうが無難である。「Section」「Tr」「Name」の3つを見て編集すべき Ctab を特定したら、その Ctab をクリックして編集を開始できる。

編集すべき Ctab をクリックして選択状態にすれば、ウィンドウ上半分の編集部分でパラメータの編集ができる。また、ウィンドウ下半分のリストの該当部分をダブルクリックすることでも編集可能である。特に Cntt の設定においては、番号で指定するのは実質的に不可能なので\*101、リストの Cntt 欄をダブルクリックして指定する (図 28)。



図 28: CASM セクション編集ウィンドウ。Ctab を選択してから図の点線囲み部分でパラメータを指定するか、Ctab のリストの該当部分を直接ダブルクリックすることでも編集できる。

### 5.3.3 CASM セクションの保存とバックアップ

全ての Ctab について適切な編集が終わったら、CASM セクションを保存しよう。CASM 編集ウィンドウである「CASM Editor...」のメニューバーの「File」→「Save (F2)」→「OK」でこの **CASM セクションが編集中のスタイルファイルに保存される**。まだこの段階では**スタイルファイル自体の保存はされていないということに十分注意せよ**。CASM セクションの保存が終わったら、「CASM Editor...」のウィンドウを閉じて、MIDI イベントが並んだ最初のウィンドウでもう一度、メニューバーから「File」→「Save as ...(F2)」→「Yamaha styles(\*.sty) で保存」で、ようやくスタイルファイルが保存される。後は、前の章と同様にエレクトーンで読み込んで、ユーザーリズムとして保存する。

ちなみに、CasmEdit は、CASM セクションをスタイルファイルから切り出して保存できる。これにより、Dominoなどで MIDI セクションを編集した後 (ここで CASM セクションは捨てられる)、バックアップしておいた編集前の CASM セクションを結合することで、Ctab を復元できる。

CASM セクションをバックアップするには、CASM セクションを編集するウィンドウのメニューバーから「File」→「Save as ...(F2)」→「保存」して、次のウィンドウで「OK」を押せば良い。

バックアップしておいた CASM セクションを結合するには、CASM セクション編集のウィンドウで「File」→「Open...(F3)」→ (csm ファイルの選択) →「開く」→「Yes 2 回」で読み込むことができる。

\*101 もちろん、[16] の通りに番号が振られているので、覚えていれば番号指定でも構わない。



## 6 Arduino を用いた MIDI デバイス製作

この章では MIDI デバイスの製作に役立つ Arduino の説明と、回路設計に関する簡単な説明を与える。ただし、Arduino の文法等に関する詳細な解説はしない。

— この章の参考書 —

- MIDI1.0 規格書 [3]  
きちんとしたものを作るなら、まず規格を参照しなくてはならない。
- Arduino のすすめ [25]  
Arduino を解説しているサイトは多いが、その中で私が最もお世話になったサイト。
- Arduino 日本語リファレンス [26]  
Arduino 言語についての説明。
- Arduino MIDI Library の使い方 [27]  
Arduino を使って MIDI を処理するなら間違いなく使うライブラリの詳細な解説。もしこの方が MIDI Library の紹介を書いていなければ、私は本稿のようなハックはしていなかった。

### 6.1 Arduino の基礎

#### 6.1.1 Arduino とは

Arduino はマイコンの一種で、Arduino ボードに Arduino 言語を書き込んで使う。Arduino ができることは単純で、せいぜい「電圧の値を読む」「電圧を出力する」「計算する」ことしかできない。しかし、それらを組み合わせれば、大いに役立つデバイスを工夫次第で作ることができる。Arduino 開発環境である Arduino IDE は、[公式サイト](https://www.arduino.cc/en/software)<sup>\*102</sup>から無料でダウンロードできる。

#### 6.1.2 Arduino ボード

Arduino 公式が出しているハードウェアにはいくつか種類がある。以下に代表的な Arduino ボードを紹介する。

- Arduino UNO  
Arduino の主要モデルである。インターネット上に情報が多いので、分からないことがあったときに解決策にたどり着く可能性が高い。初心者はまずこれを買うのが良い。
- Arduino Leonardo  
Arduino を qwerty キーボードとして Windows に認識させることができる [28]。
- Arduino MEGA  
UNO と比べて約 3 倍多くピンソケットを装備しており、メモリも大きい。MIDI 開発の観点から言うと、ハードウェアシリアルに対応しているソケットを 4 対持っていることが最大の特徴である。
- Arduino Pro Mini  
私が最もよく使っている Arduino。書き込み装置がボード上に存在しないため自分で用意する必要がある。実際の運用において書き込み装置は不要であるから、その分コストとサイズを抑えることができる。出力電圧が 3.3V のものが存在し、それを使う場合は 3.3V 用の仕様 [29] でハードウェアを設計することになる。

<sup>\*102</sup> <https://www.arduino.cc/en/software>



Arduino はオープンソースであり、ハードウェアに関しても多くの互換品が出回っている。私は最初の 1 台だけ公式の Arduino UNO を購入し、あとは全て互換品を使っている。初めて Arduino を勉強しようとする人は、Amazon などでも Arduino 初心者キットを手に入れるのが良いと思う。

### 6.1.3 Arduino 言語

Arduino 言語は C/C++ 言語をベースにしている。本稿を読みすすめるのに必要な Arduino に特有の知識を列挙する。Arduino 言語についての詳細は、本稿以外で勉強してほしい。

- Arduino におけるソースコードは「スケッチ」と呼ばれる。
- Arduino 言語に `main()` 関数は無い。その代わり、Arduino が起動したときに 1 度だけ呼ばれる `setup()` 関数と、起動している限り無限ループをする `loop()` 関数で構成されている。
- Arduino ボードには入出力用ピンが複数存在し、`setup()` の中でそれらのピンを入力で使うのか、出力で使うのかを宣言する。

### 6.1.4 Arduino 始めの一步 L チカ

流石に外部サイトに投げっぱなしでは読者が興味を持たないと思ったので、本稿でも簡単に解説をする。Arduino の雰囲気を感じ取ってもらいたい。

次は Arduino 言語における Hello World 的なスケッチであり、Blink と呼ばれる。日本では「L チカ」(LED チカチカの略?) と呼ばれる。Arduino IDE の Sample Sketch から見ることができる。

ソース 3: L チカ

```
1 void setup() {
2   pinMode(LED_BUILTIN, OUTPUT);
3 }
4
5 void loop() {
6   digitalWrite(LED_BUILTIN, HIGH);
7   delay(1000);
8   digitalWrite(LED_BUILTIN, LOW);
9   delay(1000);
10 }
```

1 行目から 3 行目までが `setup()` 関数であり、`LED_BUILTIN`<sup>\*103</sup> というピンを `OUTPUT` で使う、と初期化している。これにより、`LED_BUILTIN` に対し電圧 (オン、オフ) を出力できるようになった<sup>\*104</sup>。

5 行目から最後までが `loop()` 関数であり、`digitalWrite()` 関数と `delay()` 関数が並んでいる。`digitalWrite()` 関数は、出力に設定されているピンに対して `HIGH` または `LOW` の電圧をかける関数である。`HIGH` にした 6 行目で LED が点灯し、`LOW` にした 8 行目で LED が消灯する。`delay()` 関数は、引数 (単位はミリ秒) の間待機するという関数である。今回は 1000 ミリ秒 (=1 秒) 待機する。よって、このスケッチを実行すると、LED が 1 秒おきに点灯と消灯を繰り返すことになる。

<sup>\*103</sup> Arduino ボードには始めから LED がインストールされていることが多い。UNO の場合は 13 番ピンについている。よって UNO において `LED_BUILTIN` は 13 番ピンのことである。

<sup>\*104</sup> なお、初期化しなかった他のピンはデフォルトで `INPUT` として扱われる。

### 6.1.5 Arduino とシリアル通信

**シリアル通信**とは、HIGH または LOW の電圧レベルを時間的に連続に変化させることで、情報を送受信する通信である。MIDI がシリアル通信による送受信であるため、ここで簡単に説明する。

Arduino UNO はボードにシリアル通信用入出力ピンを装備しており、これを使うことで他の機器と通信できる。送信用のピンソケットには TX、受信用のピンソケットには RX と印刷されている。これらのシリアル通信用ピンを使ってシリアル通信する時、この通信を **HardwareSerial** と呼ぶ。なお、USB ケーブルでパソコンと接続している場合、TX/RX を使わなくても USB を介してシリアル通信を行える。

これに対して、シリアル通信用ピンが何らかの理由で使えない時、**SoftwareSerial** ライブラリをインクルードすることで、他のピンをシリアル通信のために代用できる（SoftwareSerial は明示的に宣言しないと使用されない）。しかし、SoftwareSerial はコードによって通信するため、通信が終わるまでその先の処理ができない、割り込みされるとバグる、などの不安定要素がある。MIDI の送受信においては使い物にならないため、MIDI IN/OUT を 2 セット以上搭載したい場合は、Arduino MEGA などのボードや、別なマイコンを検討する必要がある。

次のスケッチは、Arduino IDE の Sample Sketch の Graph である。

ソース 4: Graph

```
1 void setup() {  
2   Serial.begin(9600);  
3 }  
4  
5 void loop() {  
6   Serial.println(analogRead(A0));  
7   delay(2);  
8 }
```

Arduino IDE には「シリアルモニタ」「シリアルプロッタ」という、Arduino から送信されたシリアル通信のデータを表示する機能がある。このスケッチはそれらを用いて A0 ピンの電位を表示する、というチュートリアルになっている。

シリアル通信をする時は、2 行目のように `setup()` 内で `Serial.begin()` する必要がある。`Serial.begin()` の引数は通信速度であり、bps で指定する。ただし、Arduino MIDI Library（後述）を使って MIDI としてシリアル通信する場合は、`Serial.begin()` してはいけない。

`loop()` 内では A0 ピンの電位を `analogRead()` 関数で読んでいる<sup>\*105</sup>。`analogRead()` 関数は、アナログ入力ピンへの入力電圧を、5V=1023、GND=0 として、整数で返す関数である。これを、`Serial.println()` に渡すことで、シリアル通信でその整数値（改行コードつき）を送信している。`delay(2)` は `analogRead()` がデジタル値に変換するのにかかる時間を考慮して、安定化のために書いてある。

## 6.2 MIDI ソフトウェア設計 (Arduino MIDI Library)

MIDI の通信は、前述のシリアル通信で行われる。Arduino で MIDI を簡単に送受信するためのライブラリが Arduino MIDI Library である。インストールから基本的な使い方まで、Arduino MIDI Library の使い方 [27] で丁寧に解説されているため、それを熟読してほしい。

非常に簡単に解説すると、MIDI メッセージを読む機械を作りたければ、次のようにすれば良い。「おまじな

<sup>\*105</sup> `setup()` 内で A0 ピンを `pinMode(A0, INPUT)` しなくても良いのは、`pinMode` を指定しない場合のデフォルトが INPUT だからである。

い」を唱えたあと、`setup()` でハンドラを用意して、`loop()` で `MIDI.read()` すれば、用意したハンドラが勝手に呼び出されて、メッセージに対して応答できる。

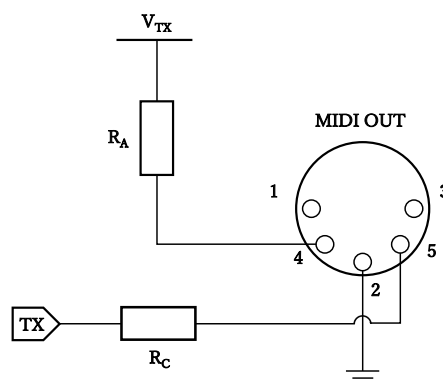
## 6.3 MIDI ハードウェア設計

MIDI データに関する処理は Arduino MIDI Library でできるようになった。次に MIDI のハードウェアのことを考える。

Arduino UNO の場合、[MIDI シールド<sup>\\*106</sup>](#) というものがあり、これを UNO の上に重ねるだけで MIDI を送受信できるようになる。ただ、MIDI シールドは値段が高いうえに大きいので、MIDI インターフェースを備えたデバイスを自分で設計できるようになっておくことは有用である。MIDI の規格書 [3][29] に設計は書いてあるが、本稿でも説明しておく。ただし、簡単にしか説明しないため、実際にハードを作るなら必ず規格を参照すること。

MIDI 端子には 5 ピンの DIN (180°) を用いる。送受信ともソケット (メス端子) を使い、送信側には「MIDI OUT」、受信側には「MIDI IN」と表記しなければならない。また、MIDI の送信側と受信側とは回路が電氣的に絶縁されており、信号はフォトカプラ (別名オプタイソレーター) というものでやり取りされている。これは LED と感光素子がセットになっているものであり、送信側の回路は受信側の LED を駆動し、受信側の回路は LED の光を信号として得る。

### 6.3.1 MIDI OUT



$V_{TX}$	$5V \pm 10\%$	$3.3V \pm 5\%$
$R_A$	$220\Omega$ 5% $\frac{1}{4}W$	$33\Omega$ 5% $\frac{1}{2}W$
$R_C$	$220\Omega$ 5% $\frac{1}{4}W$	$10\Omega$ 5% $\frac{1}{4}W$

図 29: MIDI OUT の規格 [3]。

図 29 は規格 [3] のオプションを取り外した最小構成の回路図である。送信側の電圧が 3.3V の場合と 5V の場合が規格として定められており [29]、どちらを採用するかは Arduino ボードの電圧による。Arduino UNO の場合は 5V なので、抵抗  $R_A$  と  $R_C$  はどちらも  $220\Omega$  を使う。3.3V ボードの場合は、 $R_A = 33\Omega$ 、 $R_C = 10\Omega$  とする。

<sup>\*106</sup> <https://www.switch-science.com/catalog/2492/>

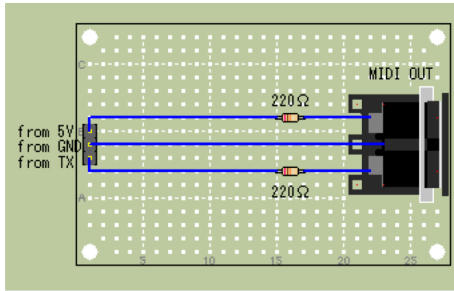


図 30: ボードが 5V の場合の MIDI OUT 回路を真上から見た場合の実体配線図。なお、ボードが 3.3V のときは、3.3V 側の抵抗を 33Ω にし、TX 側の抵抗を 10Ω にする。実体配線図は [PasS](#) で描いた。

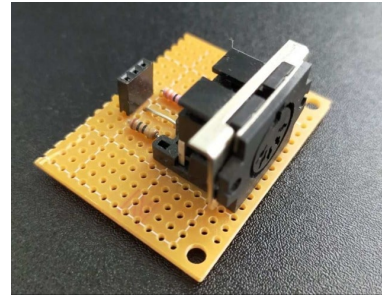


図 31: 実際の MIDI OUT 回路。Arduino とはジャンプワイヤで接続する。3.3V ボードのため、抵抗は 33Ω と 10Ω である。

また、図 30 はその実体配線図であり、図 31 はそれを実際にはんだ付けしたものである。必要なものは 2 個の抵抗だけであり、非常に簡単な回路であるということがわかると思う。

### 6.3.2 MIDI IN

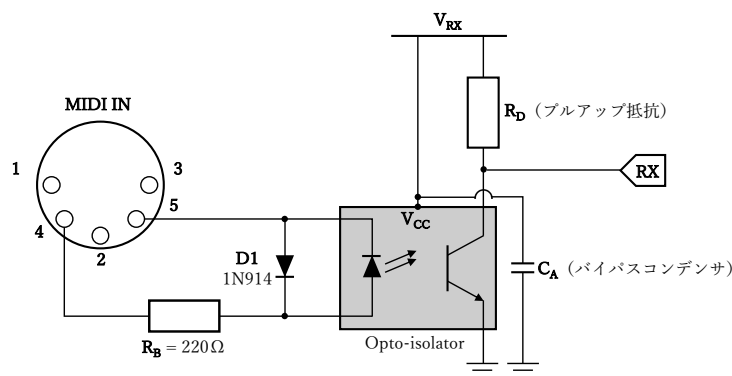


図 32: MIDI IN の規格 [3]。

図 32 は MIDI IN の規格である [3]<sup>\*107</sup>。プルアップ抵抗  $R_D$  とバイパスコンデンサ  $C_A$  は採用するフォトカプラに依存する。

フォトカプラは 5 mA 以下で駆動でき、2  $\mu$ s 以下の立ち上がり・立ち下がり時間でなくてはならない。また、MIDI 規格書 [3] に適合するフォトカプラの例が載っており、NJL-5127D、TLP513、PC-900V、PC-410(K)、PC-910(K)、HCPL-260L、HCPL-261A、HCPL-M600、QCPL-M605#500 とされている。

ここでは、電子工作を始める人のために、フォトカプラを使った実際の回路を設計していく際の流れを確認していく。まずは、選択したフォトカプラのデータシートを見よう。例として、ここでは TLP552 を採用したことにする。「TLP552 データシート」などでデータシートを閲覧できる [30]。

まずは最も重要な**絶対最大定格**から説明する。これは、この IC を使う上で**最悪のケースでも絶対に一瞬も超過してはいけない値**のことで、絶対最大定格の外側ではデバイスに損傷が出る。かならず絶対最大定格を守るように、余裕を持って設計する。

<sup>\*107</sup> 元の規格 [3] ではフォトカプラ側のパスコンが省略されていたため、図 32 では省略せずに描いた。

さて、設計のために 1 ページ目の内容を見ると、 $V_{CC}=5\text{ V}$  と書いてある。**つまりこのフォトカプラは 5 V で動作する**。また、内部回路図の注に、「8 ピンと 5 ピンの間に、バイパス用のコンデンサ  $0.1\text{ }\mu\text{F}$  をつける必要があります。」とある。このようなコンデンサを**パスコン**といい、ノイズ対策などのために付けることがある。なるべく IC の電源に近いところに、高周波の性能が良いセラミックコンデンサをつける。

新しく IC を導入する際は必ず**ピン接続図 (ピンアサイン)**を確認しよう<sup>\*108</sup>。データシートの 1 ページ目の図 (右図 33) を見ると、TLP552 のそれぞれの足に異なる機能がついていることがわかる。また、MIDI IN の回路図 (図 32) と見比べると、TLP552 の 2、3 ピンを MIDI 信号入力用の LED として使うべきであるということもわかる。よって、**TLP552 の 2 ピンを MIDI IN の 4 ピんに、TLP552 の 3 ピンを MIDI IN の 5 ピンにつなぐように配線すれば良い**。また出力側も、6 ピンが出力なので、これを Arduino の RX に**つなげば良い**。7<sup>\*109</sup>、8 ピンは 5 V につないで、5 ピンは GND につなぐので、あとはプルアップ抵抗  $R_D$  の値を決めれば、MIDI IN は実装できる。

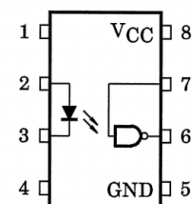
**プルアップ抵抗**とは、入力が「浮く」のを防ぐために、スイッチが切られている状態の入力を高電圧に保つための抵抗である。インターネットで調べればいくらかでも解説は出てくるのでここでの解説は省略するが、適切なプルアップ抵抗の決め方だけ簡単に解説しておく。

プルアップ抵抗の値が低すぎると、スイッチが閉じられている時に流れる電流が増える。そうすると、トランジスタで発生する電圧降下が大きくなってしまつて、スイッチが閉じられているのに Arduino が「LOW」だと判定できなくなってしまう。

逆にプルアップ抵抗が大きすぎると、Arduino のインピーダンス (大きい) を無視できなくなってしまう、スイッチが開いていても、入力端子での分圧が低すぎて「HIGH」だと判定されなくなってしまう。

したがって、プルアップ抵抗は適切な値を決める必要がある。TLP552 のデータシートではプルアップ抵抗は  $R_L$  という記号で表されているので、それに注意してデータシートのグラフ  $V_O - I_F$  を読むと、**入力電流が十分に小さいところでも俊敏に応答する  $1\text{ k}\Omega$  から  $4\text{ k}\Omega$  の抵抗値が良い**ことがわかる。

ピン接続図



- 1, 4 : N.C.
- 2 : アノード
- 3 : カソード
- 5 : GND
- 6 : 出力 (オープンコレクタ)
- 7 : イネーブル
- 8 : VCC

図 33: TLP552 のピン配線図 [30]。

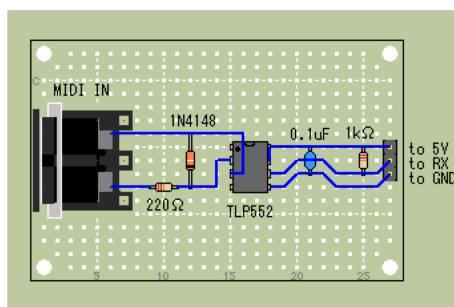


図 34: MIDI IN の実体配線図。IC は高価で熱に弱いものが多い。そのため、IC は回路に直接はんだ付けせず、ソケットだけを付ける。実体配線図においては IC がそのまま回路に乗っているが、実際にはんだ付けするのはソケットである。

以上の内容を踏まえて、図 34 で TLP552 をフォトカプラとして採用した MIDI IN の回路の実体配線図の

<sup>\*108</sup> TLP552 のような IC パッケージのピンは、丸印があるところを 1 番ピンとして、そこから反時計回りに順番に 2 番、3 番……とピンを識別する。IC によっては、丸印ではなく、IC の上方向 (12 時の方向) に切り欠きがあって、そこから反時計回りに 1 番、2 番……とすることもある。

<sup>\*109</sup> TLP552 の 7 ピンは**イネーブル**で、ここにイネーブル電圧より高い電圧を印加しないと、この IC は動かない。

例を示す。他のフォトカプラを使う場合、ピンアサインが異なったり、推奨されるパスコンの値やプルアップ抵抗の値が異なることがある。また、Arduino によってはフォトカプラを 3.3V で動かす必要も出てくる。必ずデータシートを確認して、適切な設計をしよう。

## 6.4 電子電気回路部品の紹介

MIDI デバイスを作る上で便利な電子回路部品のうち、特に使用頻度が高いものについて、ごく簡単に解説する。

### 抵抗

多くの用途があるが、例えば、流れる電流に応じて電圧降下を生んだり、電流を減らしたりするのに使う。発熱するため、種類によって決まる抵抗の容量（ワット数）を超えないように気をつける。

### 可変抵抗

抵抗の一種で、抵抗値が範囲内で自由に設定できる。Arduino では `analogRead()` すれば電圧を細かく読めるため、可変抵抗で分圧することで連続値を得られる。ボリュームノブや、フェーダーなどに使うことができるほか、圧力センサも可変抵抗であるため、プレスコントローラも作ることができる [32]。また、エクスプレッションペダルも可変抵抗である。

### ダイオード

主に、電流の流れる方向を 1 方向に制限するのに使う。順方向電圧ぶん電圧降下することに注意。

### LED

電流が流れると光る。ダイオードなので、向きがある。およそ 2V 程度の電圧降下が生じる。電流制限のための抵抗を直列に接続して、電流と光り方を制御する。

### タクトスイッチ

押している間、電流が流れる。プルアップ抵抗と組み合わせて `digitalRead()` をすれば、押されているかどうかを Arduino で検知できる。Arduino で使う場合、`pinMode()` で `INPUT_PULLUP` とすることで、プルアップ抵抗を自分で用意しなくとも、Arduino のマイコン内部のものを利用して便利である。チャタリング<sup>\*110</sup>が起きたときに誤作動しないように、シュミットトリガ回路などを組むか、スケッチの側で調整する必要がある。チャタリング防止について、例えば [31] などが参考になる。

### LCD モジュール

いわゆる 1602 系などの液晶ディスプレイ。Arduino の LiquidCrystal ライブラリを使えば、非常に簡単に文字などを出力できる。

### トランジスタ

微弱な電流を流すことで、大きな電流を制御できる。`digitalWrite(HIGH)` で流せる電流値はごくごくわずかなので、大きな電流が必要なときはトランジスタを利用する。

### リレー

電流を流すことで、別な回路のスイッチを入れたり切ったりできる。Arduino では大きすぎて出力できないような電力の制御や、回路を能動的に「切る」必要があるとき<sup>\*111</sup>などに用いる。ただし、リレーの動作にもそこそこの電流が必要なため、**リレードライブ回路**というものを組む必要がある。また、リレー動作にはコイルが用いられているため、大きな逆起電力が生じる。これを吸収するためのダイオードを入れることに注意する。

<sup>\*110</sup> チャタリングとは、物理的な衝撃によってオンとオフを高速で行き来してしまう現象のことである。

<sup>\*111</sup> 回路を切ることに直接の意味があるのは、たとえば、Arduino でフットスイッチの動作を模倣するとき。フットスイッチはただのスイッチなので、「踏む・踏まない」を回路が繋がっているか、切れているかで判断している。



## 6.5 フォン端子とその他のインターフェース

### 6.5.1 フォン端子

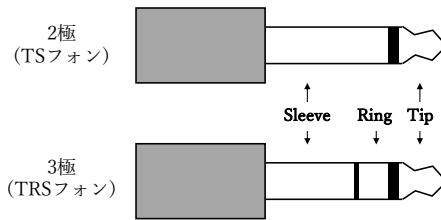


図 35: TS フォンと TRS フォンの概略図。

MIDI デバイスのインターフェースとして電子楽器用のコントローラを使うことが多いため、ここでフォン端子を紹介する。**フォン端子**は主にアナログオーディオの伝送に使われるが、電子楽器の拡張インターフェースの端子としても使われる。図 35 にフォン端子の概略図を示す。

フォン端子には絶縁体に区切られているいくつかの領域が存在している。この領域が2つのときは、それぞれチップ、スリーブと呼んで、端子を**TS フォン**と呼ぶ。領域が3つあるときは、真ん中の領域をリングと呼んで、端子を**TRS フォン**と呼ぶ。TS フォンを2極、TRS フォンを3極ということもある。イヤホンなどでオーディオを（アンバランスに）接続する場合、モノラルのものは2極、ステレオのものは3極になっていることが多い。ちなみに、マイク付きステレオイヤホンなど、コントローラを拡張すると4極以上になることもある。

### 6.5.2 その他のインターフェース

ここでは、電子楽器用のインターフェースを紹介する。

#### フットスイッチ

足で操作するスイッチで、ヤマハ FC5 や FC4A などがある。2極のものは回路的にはただのスイッチであるため、通常のタクトスイッチと同様に回路を設計すれば良い。ただし、メーカーによって極性<sup>\*112</sup>が異なるため、事前に調べておくか、メーカーを固定してしまうのが良い。フットスイッチによっては踏む強さを感知できるものもある<sup>\*113</sup>。その場合は3極になり、ただのスイッチではなく可変抵抗になる。

#### エクспRESSIONペダル

エレクトーンのものと同じように、足で操作する。普通は音量操作で用いるが、エフェクターの値をダイナミックに変更するために使うこともある。プラグは3極で、回路は可変抵抗。

<sup>\*112</sup> 踏んでいないときに ON なのか、OFF なのか、という違い。ノーマルオープン、ノーマルクローズ、と呼ばれる。製品によっては極性切り替えスイッチがついていることもある。

<sup>\*113</sup> ハーフペダル対応、と書いてあるフットスイッチ・フットペダルがそれである。



## 参考文献

- [1] ヤマハ ELS-02C 仕様<sup>\*114</sup>[https://jp.yamaha.com/products/musical\\_instruments/keyboards/electone/els-02c/specs.html#product-tabs](https://jp.yamaha.com/products/musical_instruments/keyboards/electone/els-02c/specs.html#product-tabs)
- [2] MIDI 入門[https://jp.yamaha.com/files/download/other\\_assets/6/315426/midi\\_basics\\_ja\\_v10a.pdf](https://jp.yamaha.com/files/download/other_assets/6/315426/midi_basics_ja_v10a.pdf)
- [3] MIDI1.0 規格書 <http://amei.or.jp/midistandardcommittee/MIDIspcj.html>
- [4] ELS-02/ELS-02C/ELS-02X MIDI リファレンス [https://jp.yamaha.com/files/download/other\\_assets/7/331457/els02\\_ja\\_mr\\_a0.pdf](https://jp.yamaha.com/files/download/other_assets/7/331457/els02_ja_mr_a0.pdf)
- [5] 藤本健の“DTM ステーション” 94 年にヤマハが宣戦布告。XG 規格と MU80 で DTM 全面戦争勃発 <https://www.dtmstation.com/archives/51957752.html>
- [6] ASCII.jp ローランドとヤマハ、MIDI データの互換性向上で相互協力 <https://ascii.jp/elem/000/000/319/319916/>
- [7] General MIDI -Wikipedia [https://ja.wikipedia.org/wiki/General\\_MIDI](https://ja.wikipedia.org/wiki/General_MIDI)
- [8] 偏った DTM 用語辞典 General MIDI とは <https://www.g200kg.com/jp/docs/dic/generalmidi.html>
- [9] コントロールチェンジャー一覧表 [http://quelque.sakura.ne.jp/midi\\_cc.html](http://quelque.sakura.ne.jp/midi_cc.html)
- [10] わくわくのわくみん <https://wakmin.blog.fc2.com/blog-entry-11.html>
- [11] XG 仕様書 [https://jp.yamaha.com/files/download/other\\_assets/0/321740/xg\\_v135\\_j.pdf](https://jp.yamaha.com/files/download/other_assets/0/321740/xg_v135_j.pdf)
- [12] Dr. 青山の XG 解体新書 [https://jp.yamaha.com/files/download/other\\_assets/9/321739/read\\_aoiyama.pdf](https://jp.yamaha.com/files/download/other_assets/9/321739/read_aoiyama.pdf)
- [13] XG 楽曲データ製作の指針 [https://jp.yamaha.com/files/download/other\\_assets/6/321756/xgsongdata.pdf](https://jp.yamaha.com/files/download/other_assets/6/321756/xgsongdata.pdf)
- [14] CSP-170/CSP-150 データリスト [https://jp.yamaha.com/files/download/other\\_assets/3/1101213/csp170\\_ja\\_dl\\_a0.pdf](https://jp.yamaha.com/files/download/other_assets/3/1101213/csp170_ja_dl_a0.pdf)
- [15] スタイル入門講座 <http://els01stylefile.music.coocan.jp/>
- [16] Peter Wierzba Style Files – Description <http://www.wierzba.homepage.t-online.de/stylefiles.htm>
- [17] YAMAHA Keyboard – style CASM Format [http://www.jososoftware.dk/yamaha/articles/style2\\_2.htm](http://www.jososoftware.dk/yamaha/articles/style2_2.htm)
- [18] ELA-1 Data List [https://my.yamaha.com/files/download/other\\_assets/8/1590748/ela1\\_en\\_zh\\_dl\\_a0.pdf](https://my.yamaha.com/files/download/other_assets/8/1590748/ela1_en_zh_dl_a0.pdf)
- [19] ヤマハ PSR-SX600 Reference Manual [https://jp.yamaha.com/files/download/other\\_assets/0/1346910/psrsx600\\_ja\\_rm\\_b0.pdf](https://jp.yamaha.com/files/download/other_assets/0/1346910/psrsx600_ja_rm_b0.pdf)
- [20] All that I know about Electone files <http://serge45.free.fr/electone/texte.htm#b00evt>
- [21] ヤマハ クラビノーバ CVP-109/107/105 取扱説明書 [https://jp.yamaha.com/files/download/other\\_assets/4/314934/CVP109J1.PDF](https://jp.yamaha.com/files/download/other_assets/4/314934/CVP109J1.PDF)
- [22] ELECTONE STAGEA ELS-02 ELS-02C ELS-02X 取扱説明書 [https://jp.yamaha.com/files/download/other\\_assets/0/803680/els02\\_ja\\_om\\_g0.pdf](https://jp.yamaha.com/files/download/other_assets/0/803680/els02_ja_om_g0.pdf)
- [23] 初心者になるための耳コピ MIDI 講座 <http://mimikopi.nomaki.jp/>

---

<sup>\*114</sup> ヤマハホームページや、取扱説明書のリンクは度々切れることがある。万が一リンク切れを起こしている場合は、自分で検索をかけてアクセスすること。

- [24] STAGEA と PC 編曲の連携 <https://kodack64.github.io/sound/manual/index.html>
- [25] Arduino のすすめ <https://n.mtng.org/ele/arduino/>
- [26] Arduino 日本語リファレンス <http://www.musashinodenpa.com/arduino/ref/>
- [27] Arduino MIDI Library の使い方 <https://qiita.com/yudai220/items/3bde9461f282d56d1ac2>
- [28] Electone × MIDI × Arduino <https://qiita.com/yudai220/items/b0b3dc6a8293780d5be2>
- [29] MIDI 1.0 電氣的仕様改訂 [http://amei.or.jp/midistandardcommittee/Recommended\\_Practice/ca33-j.pdf](http://amei.or.jp/midistandardcommittee/Recommended_Practice/ca33-j.pdf)
- [30] TLP552 データシート <https://toshiba.semicon-storage.com/jp/semiconductor/product/isolators-solid-state-relays/detail.TLP552.html>
- [31] jumbleat Arduino のスケッチだけでスイッチのチャタリングを回避する [https://jumbleat.com/2016/08/19/switch\\_without\\_chatter/](https://jumbleat.com/2016/08/19/switch_without_chatter/)
- [32] USB MIDI Breath Controller – Hackaday.io <https://hackaday.io/project/161678-usb-midi-breath-controller>