

エレクトーン演奏演技における MIDI の活用

wakmin ([@wakmin_](#))

2022 年 8 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	MIDI とは	5
2.2	General MIDI と XG、EL ボイス	5
2.3	MIDI チャンネルメッセージ	6
2.3.1	ノートオン・オフ	7
2.3.2	チャンネルプレッシャー	7
2.3.3	ピッチベンドチェンジ	8
2.3.4	プログラムチェンジ (PC)	8
2.3.5	コントロールチェンジ (CC)	8
2.4	MIDI システムメッセージ	10
2.4.1	システムコモンメッセージ	10
2.4.2	システムリアルタイムメッセージ	10
2.4.3	システムエクスクルーシブメッセージ (sysEX)	11
2.5	スタンダード MIDI ファイル (SMF)	11
3	XG サポート	12
3.1	XG サポートとは	12
3.2	Domino	12
3.2.1	音源定義ファイル	13
3.2.2	Domino の初期設定	13
3.2.2.1	エレクトーンとパソコンの接続	13
3.2.2.2	音源定義ファイルの導入	13
3.2.2.3	MIDI-OUT の設定	13
3.2.2.4	MIDI-IN の設定	14
3.2.2.5	その他の設定	14
3.3	XG サポートの作り方	15
3.3.1	ファイルの作成	15
3.3.2	XG サポートファイル製作の下準備	16

3.3.2.1	バルクダンプの削除	16
3.3.2.2	Ch.16 のメッセージの削除	17
3.3.3	XG サポートデータの記述	17
3.3.3.1	XG モードへの変更	17
3.3.3.2	楽器指定 (プログラムチェンジ)	19
3.3.3.3	その他のチャンネルセットアップ	19
3.3.3.4	ステップ録音	20
3.3.3.5	ピアノロール	20
3.3.3.6	イベントグラフ	21
3.3.4	XG サポートデータの保存とエレクトーンでの読み込み	22
3.3.4.1	EL ON の配置と End of Track の調節	22
3.3.4.2	XG サポートデータの保存	22
3.3.4.3	USB メモリへの転送と ELS.SONG.NAM の編集	22
3.4	XG 音源特有の仕様	24
3.4.1	システムエフェクトとインサクションエフェクト*	24
3.4.1.1	リバーブエフェクト	24
3.4.1.2	コーラスエフェクト	24
3.4.1.3	バリエーションエフェクト	24
3.4.2	メガボイス	25
4	スタイルファイルの作り方 (メイン/アドドラムのみ)	32
4.1	スタイル	32
4.2	スタイルファイルの構造	33
4.3	アドドラム・メインドラムの作り方	34
4.3.1	拍子の指定と Main A マーカーの挿入	34
4.3.2	システムセットアップ*	34
4.3.3	チャンネルセットアップ	35
4.3.4	ノートの打ち込みと Main A 以降	36
4.3.5	スタイルファイルの保存とエレクトーンでの読み込み	37
4.4	ドラムセットアップの解説	37
4.4.1	ドラムセットアップ (NRPN)*	38
4.4.2	ドラムセットアップ (sysEX)*	39
5	メモ	41
6	アカンパニメントを含めたスタイルファイルの作り方	42
6.1	自動伴奏の種類	42
6.1.1	アカンパニメント	42
6.1.2	オートベースコード (A.B.C.)	42
6.1.3	ローリズミック	42
6.2	EL Data Analyzer について	43
6.3	MIDI セクション	43
6.3.1	MIDI セクションにおける MIDI チャンネルの役割	43

6.3.2	2 小節目以降	44
6.4	CASM セクション (Ctab)	46
6.4.1	NTT	46
6.4.2	NTR	47
6.4.3	Source Root / Source Chord	47
6.4.4	High Key	47
6.4.5	Note Low Limit / Note High Limit	48
6.4.6	On Bass	48
6.4.7	RTR	49
6.4.8	Auto Start	49
6.4.9	In Channel / Out Channel	49
6.4.10	Channel Switch Chord Root / Channel Switch Chord Type	50
6.4.11	Editable Bit	50
7	Arduino を用いた MIDI デバイス製作	51
7.1	Arduino の基礎	51
7.1.1	Arduino とは	51
7.1.2	Arduino ボード	51
7.1.3	Arduino 言語	52
7.1.4	Arduino 始めの一步 L チカ	52
7.1.5	Arduino とシリアル通信	52
7.2	MIDI ソフトウェア設計 (Arduino MIDI Library)	53
7.3	MIDI ハードウェア設計	53
7.3.1	MIDI OUT	54
7.3.2	MIDI IN	55
7.3.3	その他のインターフェース	55
7.3.3.1	フットスイッチ	55
7.3.3.2	エクスプレッションペダル	56
7.3.3.3	ブレスコントローラ	56

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^{*1}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ーが存在する。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。

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。**

^{*1} 鍵盤 3 本 + ドラム 2 本。

2 MIDI データの基礎知識

この節では、MIDI のデータ部の基礎知識を解説する。とはいえ、MIDI についての解説はネットで検索すればいくらでも質の良いものがあるため、エレクトーンに应用する知識のみを紹介する。早く XG サポートを作りたい読者は**読み飛ばして構わない**。

この節の参考書

- MIDI 入門 [2]
MIDI に関するヤマハの初心者向けテキストである。電子楽器に明るくない人が一番最初に読む文献としてふさわしいと思う。
- MIDI1.0 規格書 [3]
MIDI のほとんど全ての情報を得ることができる。ただし、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。Domino (後述) を使うなら MIDI メッセージのバイナリ表現を気にする必要はほとんどないのだが、MIDI が本来的には数情報の羅列であることは心に留めておいてほしい。

2.1 MIDI とは

MIDI とは Musical Instrument Digital Interface の略で、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。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。

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。

2.2 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*3の統一などがなされた [7][8]。ヤマハは GM を拡張して **XG***4という独自規格を作

*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。

*3 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。

*4 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

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、エレクトーンの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。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。XG ボイスはスタイルと XG サポートでしか現れない^{*5}。本稿では XG ボイスを主に紹介する。

2.3 MIDI チャンネルメッセージ

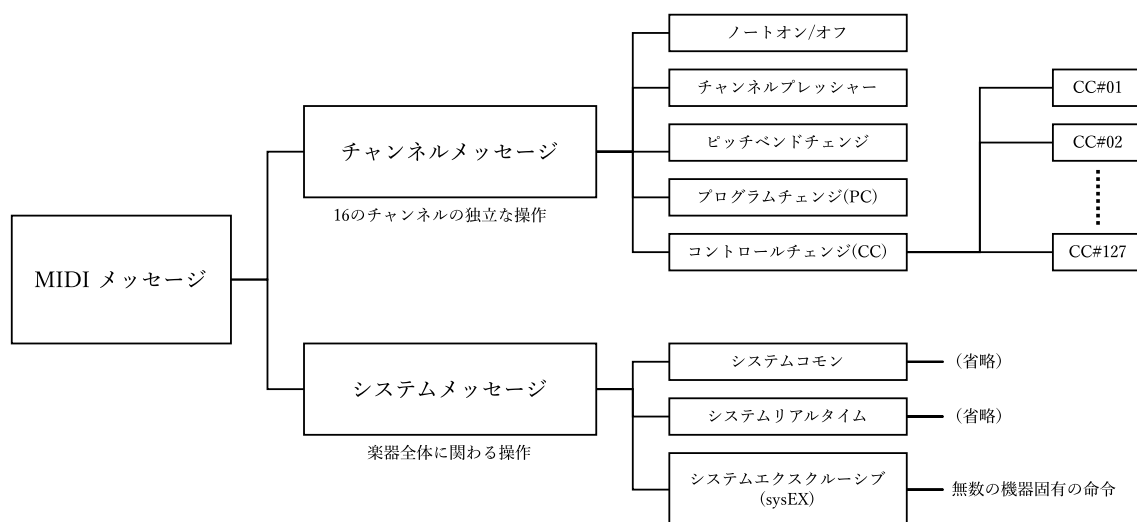


図 1: MIDI メッセージの種類。

図 1 に MIDI メッセージの種類を示す。図の上半分にある MIDI チャンネルメッセージについて、この節では解説する。

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をすることが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^{*6}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^{*8}で図 2 のように鍵盤

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)

^{*5} 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。

^{*6} 余談：チャンネルは内部的には 0 (0H^{*7}) から 15 (FH) の値である。sysEX などで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)。

^{*7} 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。

^{*8} 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 の本体設定で変更することができる。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。

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。

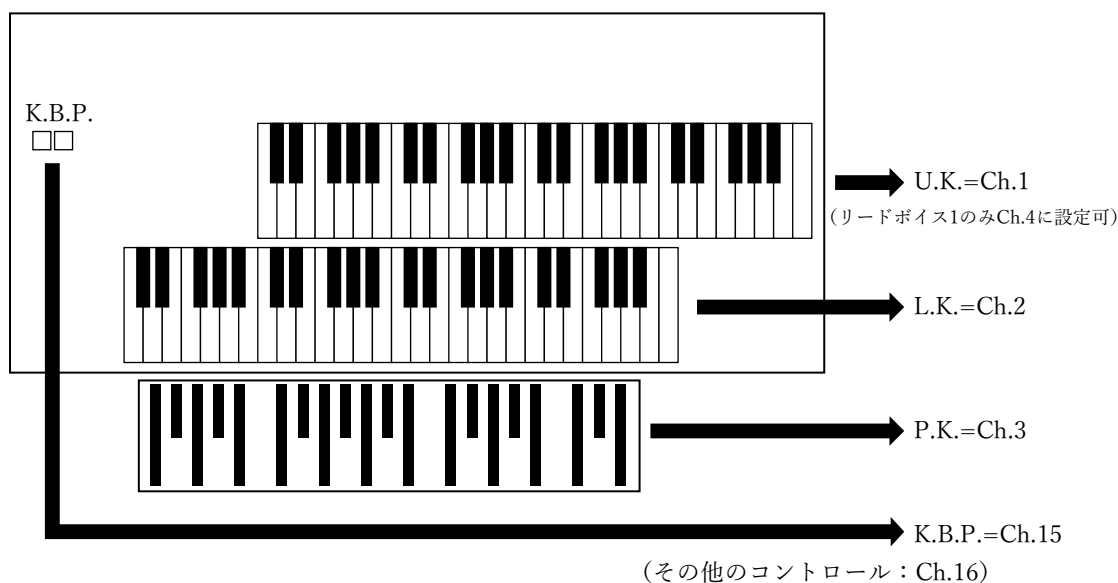


図 2: エレクトーンのデフォルト MIDI チャンネル設定。図は EL モードの場合。XG モードの場合は後述。

2.3.1 ノートオン・オフ

ノートオンとは、ノート^{*9}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ンisial 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、ノートオンのベロシティ 0 はノートオフと同じとして解釈される^{*10}。

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。

2.3.2 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^{*11}。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。

^{*9} **ノート**とは、音符の演奏情報のことである。鍵盤のキーひとつ分に対応する。

^{*10} 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、ノートオンのベロシティ 0 を出力する。なお、一応ノートオフにもベロシティが存在し、離す強さを表すことができるが、私は対応しているハードウェアを見たことがない。

^{*11} 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、ELS02 シリーズには未搭載である。

2.3.3 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、**水平タルタッチ**の強さが送信される。16384 段階に量子化されており^{*12}、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。

2.3.4 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^{*13}、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**^{*14}、メモリーしたレジストを呼び出すことに使える。XG ボイスの変更に使用し、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^{*15}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、CC#00/32 と組み合わされて表現される 1 つのボイスの番地を表すことが多い。

2.3.5 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**^{*16}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 of 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**^{*17}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

^{*12} 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。

^{*13} 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。

^{*14} レジストレーションメモリーは sysEX (後述) でも操作できる。

^{*15} プログラムチェンジの番号は、内部的には 0(OH) から 127(7FH) の値。

^{*16} http://quelque.sakura.ne.jp/midi_cc.html

^{*17} http://www.comcom2.com/lib/els_ext_xg_voice_list.html

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 ブライト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)^{*18}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）

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

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 の動作定義は以下の表の通り。

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。

2.4 MIDI システムメッセージ

この節では、図 1 の下半分にある MIDI システムメッセージについて解説する。

MIDI システムメッセージは、チャンネル指定はなく、演奏情報以外の楽器全体に関わる MIDI メッセージである。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。

2.4.1 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。

2.4.2 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

接続ができていないかの確認用の信号。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^{*19}。

^{*19} 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。

2.4.3 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^{*20}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 パラメーターも存在する^{*21}**。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。

2.5 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^{*22}、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）^{*23}。
- 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。

^{*20} 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。

^{*21} 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。

^{*22} 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ることができる。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。

^{*23} 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

この節では、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ることができる。

3.1 XG サポートとは

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することができる 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。

なお、一般的には XG サポートデータは**リズム付きの楽曲の出だしから終わりまで**を一つの SMF で作る。これは、XG サポートデータはあくまでシーケンスであり、人間の演奏の具合で動的に変更できるものではないからである。リズム無しの曲にもサポートを付けられないということはないが、製作難易度も演奏難易度も上がる。また、曲の途中から XG サポートを再生する事もできるが、MIDI の読み込みにラグがあり、MIDI 再生と同時にリズムがストップするため、現実的ではない。

3.2 Domino

Domino はたかぼ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](https://takabosoft.com/domino)^{*24}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。

^{*24} <https://takabosoft.com/domino>

3.2.1 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^{*25}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](#)^{*26}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。

3.2.2 Domino の初期設定

3.2.2.1 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^{*27}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）。

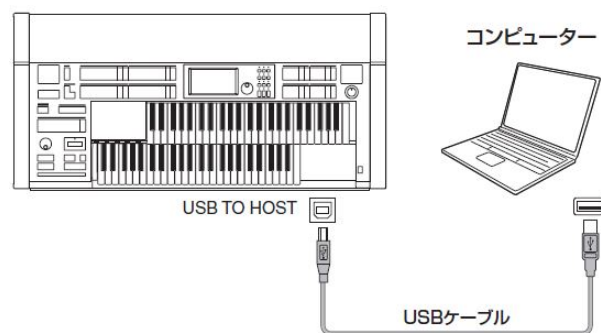


図 3: エレクトーンの TO HOST 端子とパソコンを接続する。エレクトーンとは USB で MIDI をやりとりできる。画像は[ヤマハ Q&A サイト](#)より引用。

3.2.2.2 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）。

3.2.2.3 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。

^{*25} 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。

^{*26} <https://www.kamekyame.com/el/domino-define>

^{*27} 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](#)^{*28}を参照せよ。

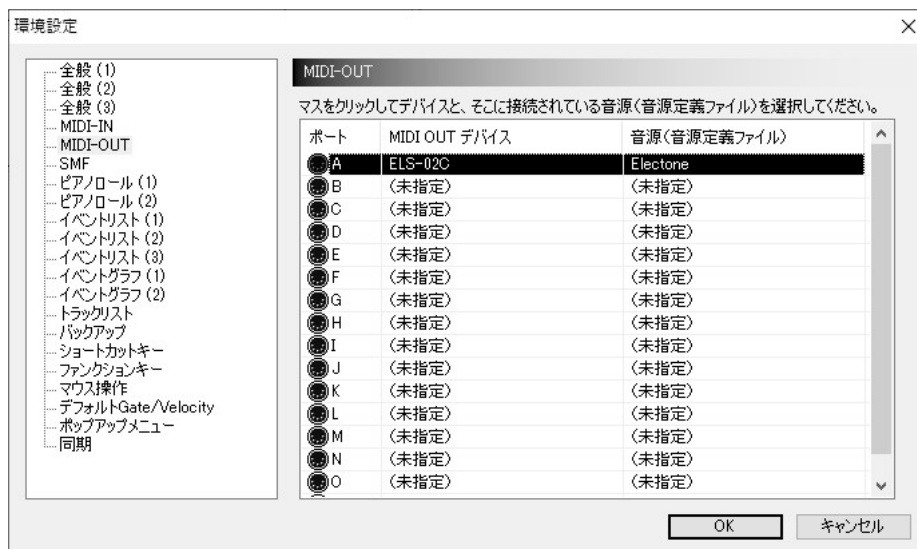


図 4: Domino の MIDI-OUT 環境設定の例。初期設定が終われば以降は設定する必要はない。

3.2.2.4 MIDI-IN の設定

必要であれば MIDI-IN の設定を行う^{*29}。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。

3.2.2.5 その他の設定

以上の他に、(必須ではないが) 設定しておくことで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)**^{*30}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

^{*29} 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。

^{*30} 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。**

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することができる。

3.3 XG サポートの作り方

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**^{*31}。また、**Windows の設定で拡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**。

3.3.1 ファイルの作成

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、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。

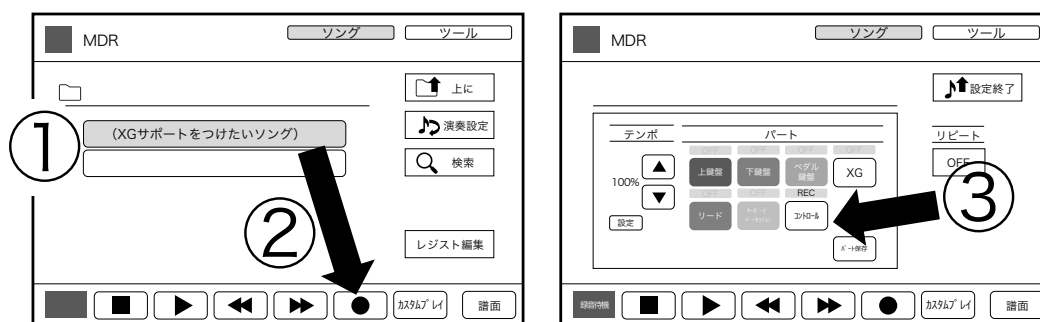


図 5: XG サポートの付け方。① 1 つのソングだけが入ったフォルダを用意し、②そのソングに録音を付け足す。③コントロールだけを録音するように設定し（他については OFF または PLAY にする）、シーケンスを曲の終わりまで流す。

^{*31} 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」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良い。

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^{*32}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。

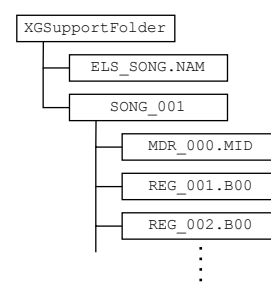


図 6: ソング内の構造。

3.3.2 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^{*33}。

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。

3.3.2.1 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^{*34}「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^{*35}。

[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^{*36}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。

^{*32} .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 [19]。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。

^{*33} 基本的に、不要な 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。

^{*34} 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。

^{*35} 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。

^{*36} 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>

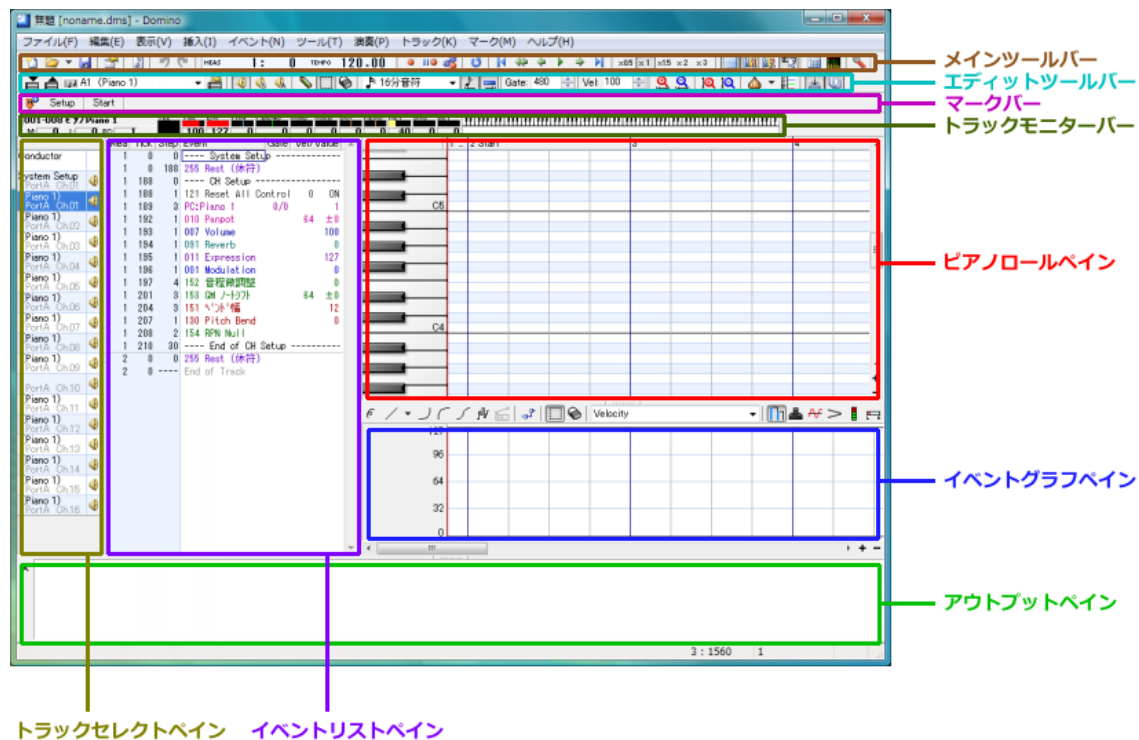


図 7: Domino の画面名称。図は Domino マニュアルから引用。

その他、[1206 Bar Signal^{*37}] や、[509 Rhythm Stop^{*38}] なども削除することができる。必要に応じて削除しよう。

3.3.2.2 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^{*39}。

3.3.3 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。

3.3.3.1 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

^{*37} 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。

^{*38} [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。

^{*39} 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。

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^{*40}で 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^{*41*42}。

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 ^{*43}	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**^{*44}。

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。

^{*40} 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。

^{*41} 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にはならない。

^{*42} 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。

^{*43} 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。

^{*44} 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 はノータイム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。

3.3.3.2 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^{*45}、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^{*46}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。

3.3.3.3 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^{*47}。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^{*48}。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 ^{*49}	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^{*50}。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) などを入力することができる。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

^{*45} 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。

^{*46} 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。

^{*47} 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。

^{*48} 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 などに設定することができる。

^{*49} 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。

^{*50} 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。

3.3.3.4 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することができる機能である。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^{*51}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）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^{*52}。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することができるようになる。

3.3.3.5 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**として使用することができる。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**^{*53}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。

^{*51} 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。

^{*52} 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。

^{*53} 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。

3.3.3.6 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、次はベ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 にベ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 とベロシティの中から、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^{*54}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。

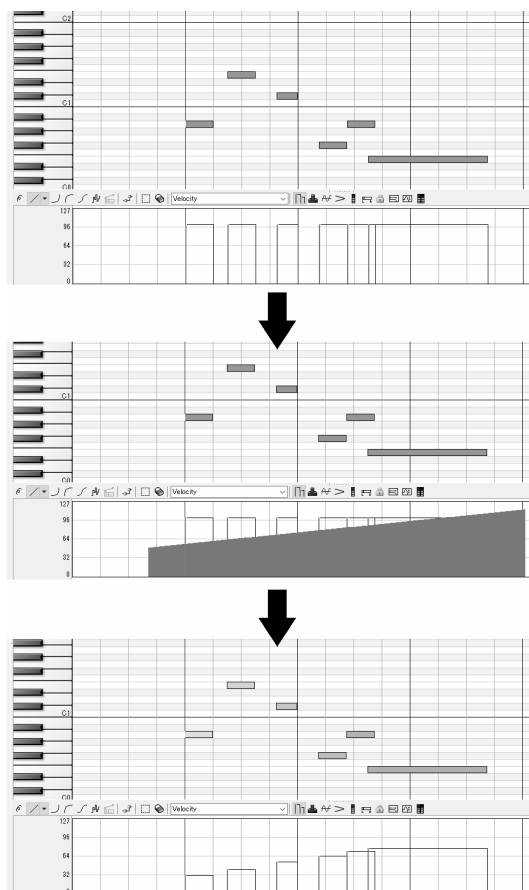


図 8: イベントグラフの使い方。ベロシティを編集するモードで直線を描くと、配置されているノートのベロシティが描いた直線に沿って指定される。

^{*54} 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。

3.3.4 XG サポートデータの保存とエレクトーンでの読み込み

3.3.4.1 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^{*55}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。

3.3.4.2 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。

3.3.4.3 USB メモリへの転送と ELS_SONG.NAM の編集

作成した SMF(XGS.MID) を、エレクトーン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^{*56}。

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。

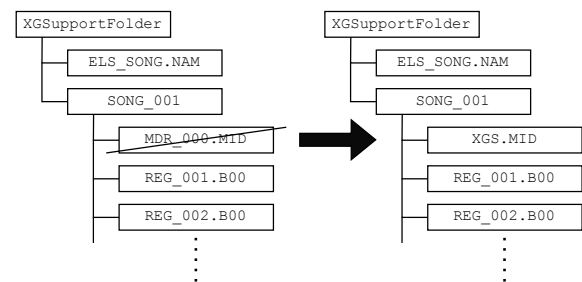


図 9: もともと存在した MDR_000.MID を消去し、作成した XG サポートデータを入れる。

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
```

^{*55} 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。

^{*56} 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。

```
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。エレクトーン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 [23]。

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。

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。

3.4 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。

3.4.1 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 10 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。

3.4.1.1 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^{*57}。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。

3.4.1.2 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 を用いてコーラスタ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。

3.4.1.3 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することができる**。図 10 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^{*58}。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

^{*57} 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り量 (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。

^{*58} 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。

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ク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。

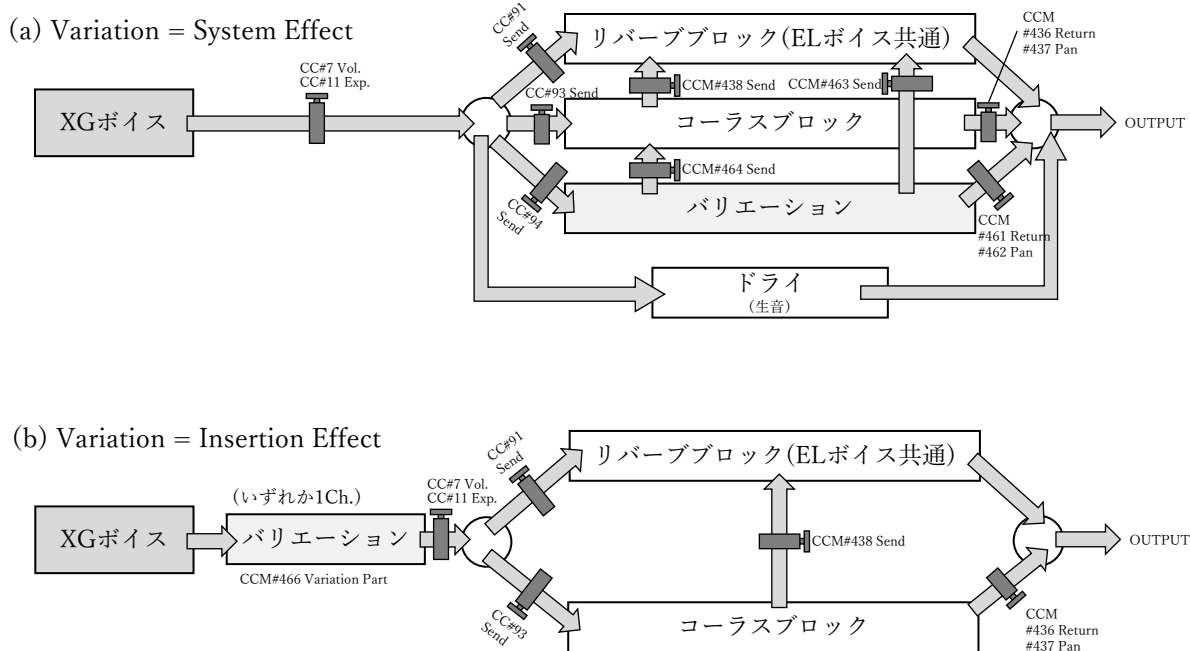


図 10: 2 種類のバリエーションエフェクトのかけ方。バリエーションを (a) システムエフェクトとして使う場合、(b) インサクションエフェクトとして使う場合。インサクションの場合は CC#94, CCM#461, 462, 463, 464 は無視される。

3.4.2 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 11 から図 16 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*59。

*59 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。

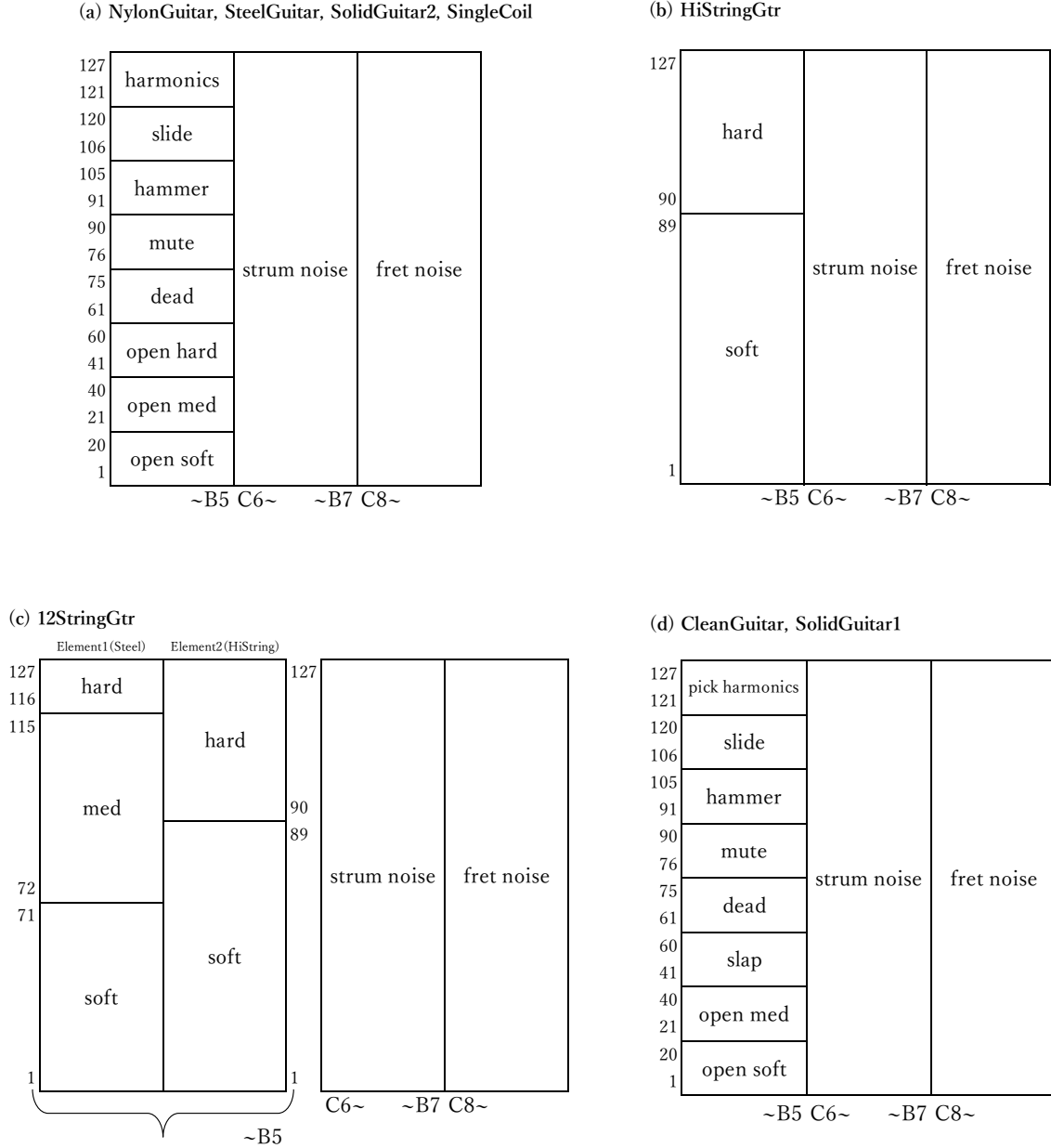


図 11: メガボイス、ギター類のボイスマッピング (1)。ボイス名の「Mega」は省略した。縦軸がペロシティで、横軸がノートナンバー。たとえば、Mega NylonGuitar の C4 をペロシティ 80 で演奏すると、mute 奏法が発音する。なお、(c) 12StringGtr の ~B5 は 2 つの元素で構成されており、それぞれが異なるペロシティマッピングになっている。

(f) SlapAmpGtr

127	pick harmonics	strum noise	fret noise	127
121				
120	slide			
106				
105	hammer			
91				
90	mute mf			
83				
82	mute mp			
76				
75	dead mf			
68				
67	dead mp			
61				
60	open slap line			
51				
50	open slap amp			
41				
40	open hard			
31				
30	open med			
21				
20	open soft		EFX	20
1				1

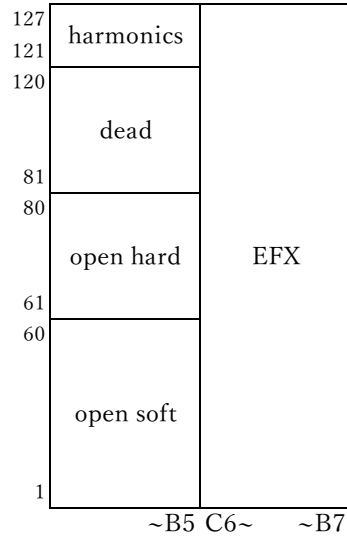
~B5 C6~

~B7 C8~

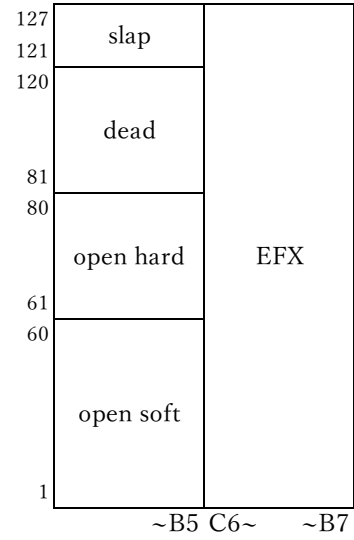
(h) JazzGuitar

127	pick harmonics	strum noise	fret noise
121			
120	slide		
106			
105	hammer		
91			
90	dead hard		
76			
75	dead soft		
61			
60	open hard		
41			
40	open med		
21			
20	open soft		
1			
	~B5 C6~	~B7 C8~	

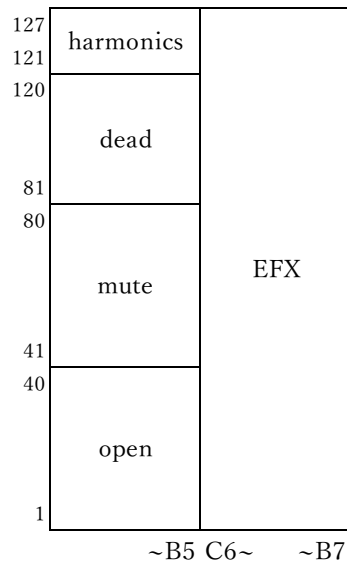
(a) AcousticBass, VintageRound, VintageFlat



(b) FingerBass



(c) PickBass, VintagePick



(d) FretlessBass

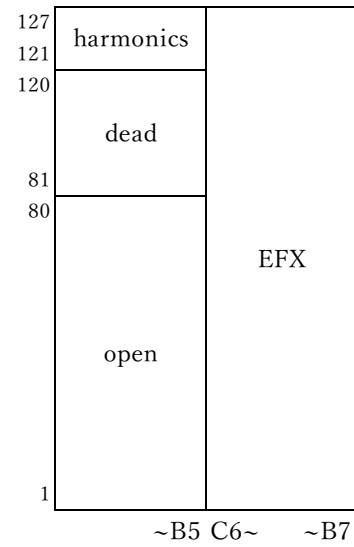


図 13: メガボイス、ベース類のボイスマッピング。

(a) PopHaa, PopHaa2, PopDaa

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

~B5 C6~ ~D#6 E6~ B6

(b) PopHoo, PopHoo2, PopShoo, PopDoo

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

~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	Ee vib f legato		
91	<i>Voice</i> vib f	breath noise	breath noise
90	Ee f legato		
81	<i>Voice</i> f		
80	Ee vib p legato		
71	<i>Voice</i> vib p		
70	Ee p legato	breath noise	breath noise
61	<i>Voice</i> p		
60			
51			
50			
41			
40			
31			
30			
21			
20			
11			
10			
1			

~B5 C6~ ~D#6 E6~ B6

(d) PopBaa

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

~B5 C6~ ~D#6 E6~ B6

図 14: 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 15: 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 16: 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（メイン/アドドラムのみ）

—— 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 セクションを編集することが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。
- ヤマハ PSR-SX600 Refernce Manual [18]
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トーン取扱説明書。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 スタイル

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ギン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 [20]。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る【1-1】。

エレクトーンの場合、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するが、メインドラム・アドドラムの 2 チャンネル分の編集と、「ローリズムミック」（後述）のみの編集となっており、スタイルの自由な編集は行うことができない。しかし、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。実際にポータトーンや 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（アカンパニメント付きの）スタイルをエレクトーンに読み込むことができる。本稿ではこの仕様を利用し、スタイルクリエイターがない環境において、パソコンで自作したスタイルファイルをエレクトーンに読み込ませる。

スタイルを自作するといっても、エレクトーン本体にリズムパターンプログラムは搭載されているため、パソコンで製作することに何の意味があるのか疑問に思うかもしれない。しかし、エレクトーン本体で弄ることができるスタイルの設定はごくごく限られており、スタイルの仕様をうまく利用することで、非常に強力な表現の手段を得ることができる。以下に例を示す。

- パソコン上のピアノロールでドラムを編集でき、細かい打ち込みを快適に行える。
- アカンパニメントの自作ができるようになる。
- 変拍子を製作できる。
- メインドラム・アドドラムに XG サポートのような自動演奏を仕込むことができる。

アカンパニメントがないユーザーリズムを作ることににおいてもスタイルファイルの活用は大いに役立つ。ぜひ自分の手でスタイルファイルを編集して、スタイルの知識を身につけて欲しい。

なお、スタイルファイルの仕様は公式には未公開であり、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。

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^{*60}。

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**^{*61}CASM セクションの記述の必要がない。そこで**本章では、スタイルファイルの作り方の流れを理解するために、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・アドドラムのみ作成方法）を紹介する。**

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**^{*62}によってさらに各セクションパターンに分割される。図 17 にセクションパターンの分割の一例を示す。

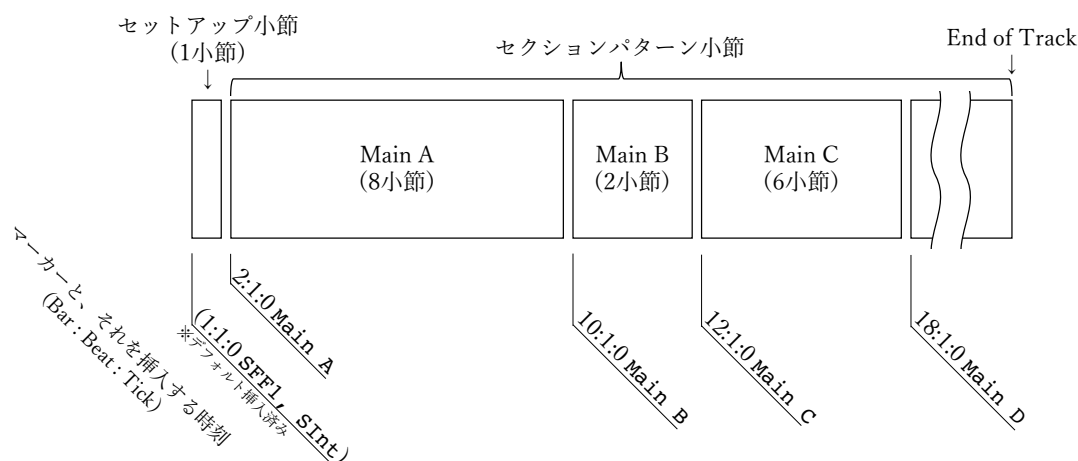


図 17: MIDI セクションの構造の例。各セクションパターンはマーカーによって分割される。例えば、Main A を MIDI セクションの 2 小節目から 8 小節目間作り、その後 Main B に移る場合は、2 小節目の頭に「Main A」と書き、10 小節目の頭に「Main B」と書く。これにより、2 小節目から 10 小節目が Main A として指定される。

各セクションパターンがエレクトーンで呼び出されるたびに、チャンネルセットアップがセットアップ小節

^{*60} 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。

^{*61} 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。

^{*62} マーカーは、任意の Tick 数に任意の文字列を挿入することができる SMF の機能である。Domino であればメニューバーの「マーカー (M)」→「マーカーの挿入 (A)...」を選ぶか、Ctrl + M でマーカーを挿入できる。

の命令で初期化される。セットアップ小節にはそのスタイルファイルで一貫している命令を書くのが良い。

なお、MIDI セクションの時刻 1:1:0 には SFF1 と SInt という 2 つのマーカーが記述されている^{*63}。これらはスタイルファイルの仕様上必要である。

4.3 アドドラム・メインドラムの作り方

いよいよ、スタイルファイルの作り方を紹介していく。まずはザッと一読して流れを掴んでから、その後にゆっくりと手を動かしながら実践するのが良い。以降、[エレクトーン用音源定義ファイル v1.6.0](#)を導入しているものとして解説する。

4.3.1 拍子の指定と Main A マーカーの挿入

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
				-----End of System Setup-----		
2	1	0	0	End of Track		

図 18: 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**をすることができる。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することができる。**

デフォルトでは図 18 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]

^{*63} 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]。

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、インサー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 小節目でのみ行うことができる**。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 19: デフォルトのドラムトラックの中身。

次に、メインドラム・アドドラムの楽器指定などを行おう。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]^{*64}を選ぶ。そうすると、図 19 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。この時、「ドラムセットリスト」から選べば、通常のエレクトーンのドラムセットを選択できる。さらに、図 20 のように「音色リスト」から選べば**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、メインドラムやアドドラムを **XG サポートのような完全自動演奏として活用**できる。

また、他のチャンネルセットアップも変更できる。XG サポートを製作する時と同様に、チャンネルセットアップを書き足したり、値を変更したりできる。なお、1 小節目に書いたチャンネルセットアップは、全てのセクションパターンで共通になる。

^{*64} [Main Drums] トラックは Ch.10 であり、基本的には^{*65}メインドラムとして使用できる。

^{*65} CASM セクションを編集する場合は、Ch.10 であってもメインドラム以外に指定できる。詳細は CASM セクションの In/Out Channel で解説する。



図 20: プログラムチェンジを変更するウィンドウ。ドラムトラックであっても、全ての XG 音源音色リストから音色を選択することができる。

4.3.4 ノートの打ち込みと Main A 以降

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、メインドラムとアドドラムに Main A のドラムを打ち込もう。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^{*66}ため、CC を使って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^{*67}。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^{*68}。

表 4: セクションパターンを指定するマーカー一覧。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			

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^{*69}。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

^{*66} セクションパターン小節では、sysEX（や NRPN, RPN）による制御はできない。

^{*67} End of Track についての解説は、XG サポートの作り方にある。

^{*68} 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。

^{*69} 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。

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。

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 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 も設定し終えたら、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 [21]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よう。

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。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**。ドラムセットアップに関する説明をエレクトーン奏者向けにしている資料が見当たらなかったため紹介するが、エレクトーンで操作できるセットアップはエレクトーンで操作した方が、バランスをとりながら編集できて良いと考えているため、必要ないと思ったら飛ばして構わない。しかし、エレクトーンから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^{*70}。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。

ドラム音源のエディットは NRPN によるものと sysEX によるものがある^{*71}。設定できる項目が多いため、節を分けて NRPN と sysEX を別々に解説する。なお、**どちらのドラムセットアップを使うにせよ、ドラムセットアップの命令は 1 小節目に書く必要がある (2 小節目以降は無視される)**。

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 21 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。

^{*70} 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。

^{*71} MIDI の一般的な約束として、sysEX と NRPN のどちらでも同じ効果が得られる場合、できるだけ NRPN を使用することになっている。そのため本来は、NRPN で設定しきれない場合のみ sysEX を用いるのが良いと思うのだが、エレクトーンで作成したスタイルでは、NRPN で操作できるはずのドラムセットアップパラメータであっても、全て sysEX で記述されている。理由は不明。

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 sysEX 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。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^{*72}。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 Rcv 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 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することが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。

^{*72} 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。

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 メモ

以下草稿です。情報の寄せ集めに過ぎません。

6 アカンパニメントを含めたスタイルファイルの作り方

6.1 自動伴奏の種類

スタイルの自動伴奏はアカンパニメントとオートベースコードという 2 つの仕組みで構成されている。スタイルファイルフォーマットではないが、エレクトーンにはロワーリズムックという自動伴奏も存在する。

6.1.1 アカンパニメント

アカンパニメントはスタイルの自動伴奏のうち、判定されたコードに対し自動で演奏される「コード 1/2」、「パッド」、「フレーズ 1/2」の 5 つを指す。

6.1.2 オートベースコード (A.B.C.)

オートベースコードはアカンパニメントにおけるコードの判定方法を指定し、加えてベースの自動伴奏の有無も指定する。

エレクトーンのオートベースコードには、以下の 4 種類のモードがある【16-95】。

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。

エレクトーン演奏に慣れているプレイヤーであれば、普通はカスタム A.B.C. を採用する。

6.1.3 ロワーリズムック

ロワーリズムックはリズムパターンプログラムで作ることができるアカンパニメントの 1 つで、下鍵盤で演奏しているノートをオリジナルのリズムで連打させることができる機能である。ロワーリズムックもアカンパニメントではあるのだが、コードの判定をしてフレーズを構成するなどの自動演奏は行われない。コード 1/2 のチャンネルを利用して発音するが、スタイルファイルフォーマットにはこれを制御する命令は存在せず^{*73}、エレクトーン内でのみロワーリズムックの指定が可能である^{*74}【19-9】。また、エレクトーンに初めから入っているリズムにロワーリズムックを使ったものは存在しない【19-2】。本稿ではロワーリズムックについてこれ以上の解説はせず、他の自動伴奏について解説する。

^{*73} ロワーリズムック用のチャンネルパラメータは Ctab ではなく Crmc という別の枠組みで処理する【19-9】。

^{*74} Domino などの MIDI シーケンサーを使ってロワーリズム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ワーリズムックの音価で打ち込んでおいて、エレクトーンに読み込ませる。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。これをロワーリズムックにするため、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。そうしてから保存すれば、ロワーリズムックとして保存される。

6.2 EL Data Analyzer について

EL Data Analyzer とは、東北大学エレクトーンサークル MUSICA の OB である岩田駿人氏によって作成された、エレクトーンで作成されたスタイルファイルをパソコン上に取り出すソフトウェアである。<https://dummy> からダウンロードできる。以下のようにして使う。

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。

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。

6.3 MIDI セクション

MIDI セクションは伴奏の MIDI メッセージを記述するセクションである。基本的にはフォーマット 0 の SMF であるため、これを編集するには前節で紹介した Domino などを用いる。

スタイルはコードを判定して MIDI セクションの自動伴奏を適切に読み換えるのだが、それでは MIDI セクションの元々の MIDI パターンでは何を書いておけば良いのだろうか？ MIDI セクションで置いたノートがどのコードを基準にしているかは CASM セクションで設定するのだが、面倒な設定抜きで、とりあえず矛盾なく演奏できるものを作りたいのなら**自動伴奏には C_{M7} の構成音を書く**ことになる^{*75*76}。これに関する詳しい説明は CASM セクションの Chord Root / Chord Type の節で行う。

推奨されるアカンパニメントのパートの使い分け [21] と、 C_{M7} を基準にしたときに無理なく使うことのできるノートを表 5 に記す。これ以外のノートを使用した場合、演奏するコードによっては意図しない移調^{*77}をされる恐れがある [16]。ただし、これらの使用可能ノートは CASM セクション内で NTT により指定されているため、CASM セクションをこだわって製作する場合は^{*78}表 5 の通りに作る必要はない。

6.3.1 MIDI セクションにおける MIDI チャンネルの役割

エレクトーンで見ることができ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 つだが、スタイルで使用するこのできる MIDI チャンネルは 16 本あり、スタイルはこの 16 チャンネル分の MIDI データを 8 パートに合流させて自動演奏させることができる。このため、スタイルには「入力チャンネル」と「出力チャンネル」という概念が存在する [17-6] ^{*79}。MIDI セクションに記述する MIDI メッセージが「入力」であり、エレクトーンの 8 つのセクションパートが「出力」である。

^{*75} C_{M7} はコードネームであり、簡単に言えばド・ミ・ソ・シのことである。

^{*76} C_{M7} の構成音を書けば良いのであって、ドミソシの順に積まないといけないわけではない。ボイシングを変えても、エレクトーンの側で適宜読み替えをしてくれる。自分が C_{M7} を演奏したときに、再生されてほしいフレーズを書こう。ただし、実際のパフォーマンスに使う場合は、必ず実験とリハーサルをすること。

^{*77} 音楽用語の「移調」の意味ではなく、本稿では自動伴奏におけるフレーズの読みかえのことを移調 (transpose) と呼ぶことにする。

^{*78} こだわって製作する場合は、CASM セクションの説明の図 23 を参照してほしい。

^{*79} 例えば、アンブラグド 1 はコード 1 に 12 チャンネル分のデータが送られる [17-8]。

表 5: アカンパニメントのパートと使用可能なノート

パート	説明	使用可能ノート (C _{M7} 基準)
ベース	-	C, D, E, G, A, B
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。	C, E, G, B
パッド	ストリングスやオルガンなどの演奏を再生する。	C, E, G, B
フレーズ 1/2	アルペジオや、パンチの効いた ブ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きわ華やかなパート。	C, D, E, G, A, B

表 6 にスタイルの出力チャンネルとパートの対応を示す。ただし、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、ということではできない【17-9】。

CASM セクションがスタイルファイルに存在しない場合、エレクトーン
のデフォルトの CASM セクションがスタイルファイルに結合されて読み
込まれ、出力チャンネル設定は入力チャンネル設定と同じになる。よって、
**メインドラムとアドドラムしか MIDI セクションを記述しなかった場合、
これらをそれぞれ Ch.10 と Ch.9 で書いていれば、CASM セクション
を記述する必要はない。**CASM セクションを記述する場合ほどのチャンネ
ルにどのパートの演奏を書いても構わないが、CASM セクションが壊れる
ような操作は多い*80 ため、**特別な理由がない場合は、MIDI メッセージの
チャンネルは表 6 になるべく合わせるべきであろう。**

この仕様は、CASM セクションを設定することで、特定の押鍵のスタイル
だけ違うパターン（違う楽器にもできる*81）で自動演奏させたり、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、
再生は 1 つのパートでしたりするときなどに活用する。具体的な方法は後
述の CASM セクションの In / Out Channel の説明を参照してほしい。

表 6: スタイルファイルの出力チャンネル設定。

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

6.3.2 2 小節目以降

イントロとエンディングは、鍵盤で押さえているコードが再生中に変更されない前提で作る。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点に注意しなくてはならない。

オートベースコードも基本的にアカンパニメントと同じように編集する。ただし、以下のようにオートベースコード特有の注意点が存在する。

- エレクトーンはベースパートのプログラムチェンジを使うことができない。書いたフレーズは足鍵盤ボイスで再生される。
- 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【16-12】。

*80 例えば、.sty ファイルの拡張子を.mid ファイルに書き換えて Domino で開いたあと、.mid ファイルを上書き保存することで、CASM セクションを消すことができる。Domino は CASM セクションを読み込むことができないので、開いた段階で CASM セクションを捨てている。

*81 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。

- CASH セクションで Note Limit を設定しないと、意図しないオクターブ移動をされる恐れがあるため、オクターブをまたぐフレーズを書く場合は CASH セクション編集の際に Note Limit を設定することを忘れないこと。

ベースにメロディアスなフレーズをつけることもできるが、カスタム A.B.C. の場合は L.K. とセットでコード判定が行われるため、意図したフレーズを再生できない場合も多い。オートベースコードには同音連打程度の簡単な伴奏を書くことを勧める。

1 小節目と 2 小節目以降とで扱うことのできる MIDI メッセージが異なる [16]。sysEX は 2 小節目以降に記述できない。また、スタイルは 2 小節目以降の拍子変更ができない^{*82}。しかし、スタイル内の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^{*83}。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。他の MIDI メッセージについては表 7 に示す^{*84}。

表 7: 扱うことができる MIDI メッセージ。[16] より引用。

MIDI イベント	1 小節目	2 小節目以降
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(ブライトネス)	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	-

^{*82} 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目単位で定義されているためである [5-34]

^{*83} 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（セットアップ小節目）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 小節目で 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。ただし、変拍子を普通に作る場合は 1 小節目の拍子を調節すれば良い。この方法は、スタイルのセクションパターンの一部だけを変拍子にしたい場合などに活用できる。

^{*84} この表については [16] から引用したものであるから、エレクトーンにおいては事情が異なる可能性がある。

6.4 CASM セクション (Ctab)

CASM セクションは、MIDI セクションの各セクションパターン (CSEG) に対する **Channel table (Ctab)** 設定の集合体である。Ctab とは MIDI セクションの各チャンネルをどのように解釈して自動伴奏を構成するかを命令のことである。CASM セクションは自動伴奏の設定であるため、**自動伴奏を設計しない場合、CASM セクションを編集しないでエレクトーンに読み込んでも問題ない**^{*85}。この節はアカンパニメントやオートベースコードの編集をする人だけ読めば良い。Ctab の全体像をざっくりと把握するために、**【17-1】**を見ると良い。本稿では、**【17-1】**の通し番号の順に Ctab の命令を説明する。エレクトーン以外のヤマハ電子楽器の取扱説明書に Ctab の説明が書いてあるものがあるため、本稿で物足りないと感じた読者は、例えばポータトーンのリファレンスマニュアル **[18]** の 44 ページ以降などを参照してほしい。

CASM セクションを編集するには、**CasmEdit**^{*86}などのソフトを用いる。本稿では CasmEdit の使い方も含めて解説する。

6.4.1 NTT

NTT とは Note Transposition Table のことで、実際に演奏しているときの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移調するかを指定する^{*87}。以下に各 NTT の特徴を説明する **【16-26】**。

Bypass

移調を行わない。メインドラム・アドドラムなどはコードタイプに合わせて移調するわけにはいかないため、NTT を Bypass に指定しなくてはならない^{*88}。ほかに、効果音的に使いたい楽器に対して Bypass を指定することがある^{*89}。

Melody

メロディライン向けの移調を行う。フレーズ 1/2 に出力されるようなパートで使う。エレクトーン^{*90}の場合はベースパートも (NTT=Bass を使用しないで) Melody に指定する。ただしこの時、On Bass というパラメータを有効にする。詳しくは後述。

Chord

コードプレイ向けの移調を行う。コード 1/2 やパッドに出力されるようなパートで使う。

Bass

ベース用の NTT であるが、現在は使われない古い仕様。NTT=Melody とほとんど同じだが、オンコード^{*91}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。現在はベースパートには NTT=Bass は使用せずに、NTT=Melody とした上で On Bass を有効にする。

その他^{*92}

^{*85} Domino で MIDI セクションまでを書き終えたら、「SMF として書き出し」からフォーマット 0 を指定して書き出し、拡張子を .mid から .sty にするだけでエレクトーンで読み込むことができる。

^{*86} https://www.mnppsaturn.ru/osenenko/Main_eng.htm

^{*87} 新しい電子楽器では Cntt というセクションが追加され、Cntt に対応している楽器では NTT の命令を Cntt で上書きするようになった。エレクトーンでは Cntt セクションを用いている。そのため、CasmEdit で編集する際は NTT ではなく Cntt を編集することになる。なお、Bypass, Melody, Chord, Bass と Melodic Minor, Harmonic Minor が NTT に存在する設定であり、その他の NTT は Cntt で記述する。後述の On Bass も Cntt の内容。詳しくは **[16]** を見よ。

^{*88} エレクトーンではメインドラム・アドドラムは NTT/NTR の設定に関わらず移調をスルーする **【7-1】**。

^{*89} 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移調されると意図した効果を出せない場合がある。

^{*90} エレクトーンを含む 2002 年以降の機種。

^{*91} オンコードとは、ベースが和音のルートを演奏しないようなコードのことである。

^{*92} Melodic Minor, Melodic Minor 5th, Harmonic Minor, Harmonic Minor 5th, Natural Minor, Natural Minor 5th, Dorian, Dorian 5th。

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移調を行う。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 [18] を参照してほしい。

6.4.2 NTR

NTR とは Note Transposition Rule のことで、移調の際に展回形を用いるかを指定する。エレクトーンで使う場合は Root Trans と Root Fixed から選ぶ。

Root Trans

図 22 左のように、移調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。例えば、ハ長調の C3, E3, G3 は、ヘ長調のときに F3, A3, C4 に移調される。普通、メロディラインがあるパートに Root Trans を設定する。

Root Fixed

図 22 右のように、移調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。例えば、ハ長調の C3, E3, G3 は、ヘ長調のときに C3, F3, A3 に移調される。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 (NTT=Bypass を指定した上でさらに) NTR=Root Fixed に指定する。

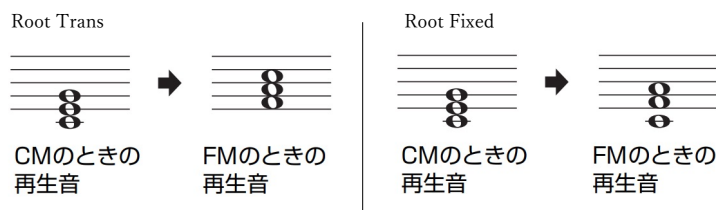


図 22: Root Trans と Root Fixed の違い。[18] より引用。

6.4.3 Source Root / Source Chord

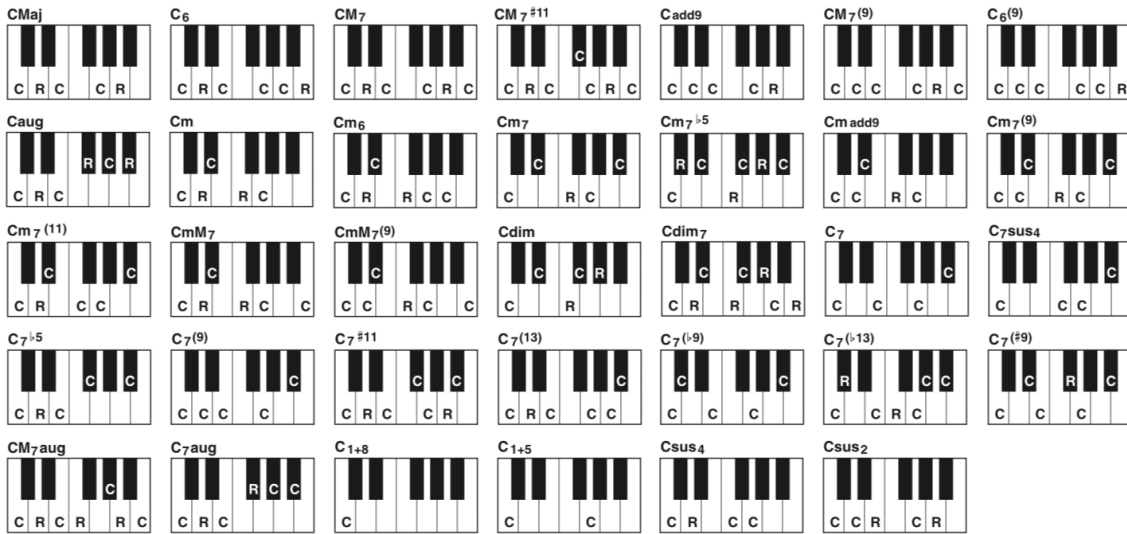
MIDI セクションで記述したパターンが、何のコードのときを基準にしているかを指定する。Source Root でコードのルートを指定し、Source Chord でコードタイプを指定する。デフォルトの CASM セクションの場合は C_{M7} になっている。指定できるコードタイプと、推奨ノートを図 23 に示す [18]。

スタイルファイルを製作する際に、MIDI セクションには C_{M7} の構成音を書くように解説することが多いのは、Source Root / Chord のデフォルトが C_{M7} だからである。これを知っていれば、 C_{M7} 準拠で MIDI セクションを記述する必要はない。CasmEdit では Source Chord Root(C.R.) と Source Chord Type(C.T.) と呼ばれている。

6.4.4 High Key

どのノートを境にパターンをオクターブ下に下げるかを指定する。High Key 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 24)。NTR=Root Trans のときのみ設定が有効になる。CasmEdit では Highest Key(H.K.) と呼ばれている。

ソースルートCの場合



C = 本来のコードノート

R = 使用推奨ノート

*ソースパターンは、CとRで構成することをおすすめします。

図 23: Source Root=C のときのコードタイプと推奨ノート一覧。[18] より引用。

【例】ハイキーがFのとき

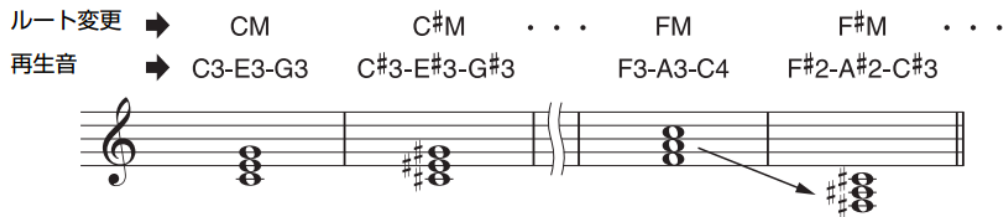


図 24: High Key がどのように効くかの説明の図。[18] より引用。

6.4.5 Note Low Limit / Note High Limit

音高変換された後のノートに Note Limit よりも外側の音が存在した場合、Note Limit の内側^{*93}にオクターブを変える。High Key が判定コード単位の変換なのに対し、Note Limit は最終的な発音に対して下限/上限の制限を設けて発音音域を整える【16-72】。CasmEdit では Low Limit(L.L.)、High Limit(H.L.) と呼ばれている。エレクトーンのプリセットのスタイルの場合、ベースパートの Low Limit は E0 になっていることが多い。これは 4 弦ベースの通常のチューニングの最低音が E0 であるからだと思われる。

6.4.6 On Bass

On Bass^{*94}が有効のとき、オンコードを演奏した場合に、ベースルートで再生される【15-31】【15-32】。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。通常の場合ベースパートは NTT=Melody にして、On Bass を有効にする。

^{*93} Note Limit のノートを含む内側。つまり、Note Low Limit=E0, Note High Limit=G8 のときに E0 を演奏するとそのまま発音されるが、Eb0 を演奏するとオクターブが上がる。

^{*94} On Bass は Cntt セクションの一部であるため、Ctab 命令ではない。CasmEdit で編集する際は、Cntt のパラメータのうち On Bass が有効のものを選択する形になる。

6.4.7 RTR

RTR とは ReTrigger Rule のことで、スタイル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】。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、新しいコードに移調する。

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移調する。

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 にすることをお勧めする。

6.4.8 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、この設定は無視され、メイン・アドドラムは Auto Start で再生され、それ以外に Auto Start は設定できないと思われる^{*95}。ドラムが強制的に Auto Start になるおかげで、**メインドラム・アドドラムのパートにドラム以外のプログラムチェンジを設定し、リズム再生に同期して（演奏に合わせて読み替えをしない、ドラム同様に固定の^{*96}）自動演奏を再生させる**ことができ、簡易的な XG サポートの代わりとして利用できる。

6.4.9 In Channel / Out Channel

「MIDI セクションにおける MIDI チャンネルの役割」の節で述べたように、スタイルファイルは 1 つのパートに対して複数のチャンネルを設定できる。これを行うのが In / Out Channel である。MIDI セクションで書いたチャンネルを In Channel にし、表 6 で示されるパートで Out Channel を選ぶ。例えば、MIDI セクションで音域別にチャンネルを分けて書いておいて、CASM セクションでチャンネルごとに異なる音高変換をさせたり、次に紹介する Channel Switch Chord Root / Type で特定コードだけ別のチャンネルを参照するようにして、パターンを弾き分けたりできる。【11-20】も参照すると良い。

^{*95} 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。

^{*96} ドラムパートが移調をスルーすることを利用している。メインドラム・アドドラムのフレーズは、どんな音色を使っているかに関係なく、そのまま Auto Start で再生されるため、メロディなども再生できる。ただし、Auto Start を設定できるのはメイン・アドドラムの 2 系統までであるため、それ以上にトラックを必要とする場合は XG サポートを書くことになる。

6.4.10 Channel Switch Chord Root / Channel Switch Chord Type

Channel Switch Chord Root / Channel Switch Chord Type^{*97}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。In / Out Channel と組み合わせて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。普通は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が、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。なお、デフォルトでは全てのコードが演奏されるようになっている【17-3】。この仕様の勉強のために、【17-2】でセブンスコードのときだけ有効になるチャンネルが存在するスタイルファイルを配布している。エレクトーンで読み込んだり、CasmEdit で読み込んだりして勉強に使うと良い。

6.4.11 Editable Bit

スタイルクリエーター搭載電子楽器での編集可能フラグ。

^{*97} Channel Switch Chord Root / Channel Switch Chord Type は、[15] ではこの名称であるが、公式名称が明らかになっていないため、ソフトや記事によって様々な呼ばれ方が存在する。他に Note Mute / Chord Mute[16]、Active Keys / Active Chords[17]、Note Mute(NMute) / Chord Mute(CMute) (CasmEdit) などの名称がある。

7 Arduino を用いた MIDI デバイス製作

この節では 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。

この節の参考書

- 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 [24]
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 [25]
Arduino 言語についての説明。
- Arduino MIDI Library の使い方 [26]
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。

7.1 Arduino の基礎

7.1.1 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)^{*98}から無料でダウンロードできる。

7.1.2 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が、私がおすすめするのは以下である。

- 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 [27]。
- 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 [28] でハードウェアを設計することになる。

^{*98} <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。

7.1.3 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。

7.1.4 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。

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`^{*99} 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^{*100}。

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。

7.1.5 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。

^{*99} 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。

^{*100} 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。

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することができる。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** ライブラリをインストールすることで、他のピンをシリアル通信のために代用することができる（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 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^{*101}。`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。

7.2 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 [26] で丁寧に解説されているため、それを熟読してほしい。

7.3 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。

^{*101} `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。

Arduino UNO の場合、[MIDI シールド](#)^{*102}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][28] 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 は送信側 MIDI OUT と受信側 MIDI IN とで回路が電氣的に絶縁されている。信号はフォトカップラ（別名オプトアイソレーター）というものでやり取りされており、これは LED と感光素子がセットになっているものである。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。

7.3.1 MIDI OUT

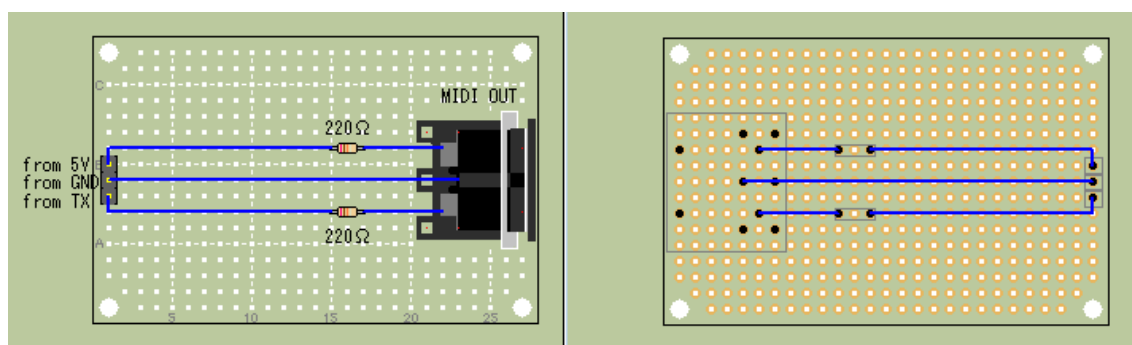


図 25: ボードが 5V の場合の MIDI OUT 回路を真上から見た場合の実体配線図。左側が表、右側が裏。なお、ボードが 3.3V のときは、3.3V 側の抵抗を 33Ω にし、TX 側の抵抗を 10Ω にする。実体配線図は [PasS](#) で描いた。

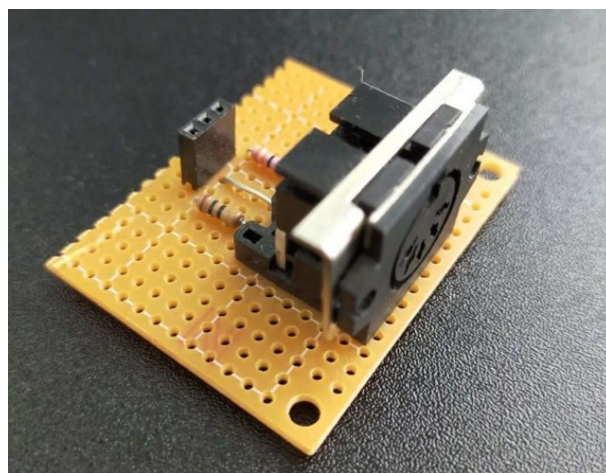


図 26: 実際の MIDI OUT 回路。Arduino とはジャンパワイヤで接続する。3.3V ボードのため、抵抗は 33Ω と 10Ω である。

図 25 は MIDI OUT の回路の（規格のオプションを全て付けない場合の）実体配線図であり、図 26 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。

^{*102} <https://www.switch-science.com/catalog/2492/>

7.3.2 MIDI IN

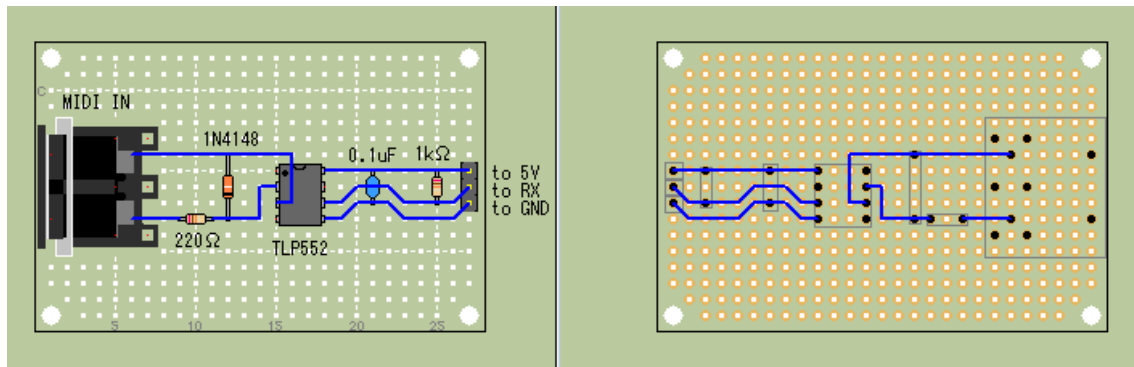


図 27: MIDI IN 回路を真上から見た場合の実体配線図。左側が表、右側が裏。IC は高速フォトカプラで、MIDI 規格書 [3] に適合するフォトカプラの例が載っている。フォトカプラは負論理である必要がある。

図 27 は 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例である^{*103}[29]。他のフォトカプラの場合、ピンアサインが異なる恐れがあるので、フォトカプラのデータシートを必ず確認すること。なお、この回路においてフォトカプラだけを HCPL-260L に交換しても動作することを確認している。

MIDI IN 端子側のダイオードはフォトカプラの保護用である。Arduino 側の 0.1μF コンデンサと 1kΩ 抵抗は、どちらもフォトカプラのための素子であり、MIDI の規格ではない。コンデンサはバイパスコンデンサ（パスコン）であり、1kΩ 抵抗はプルアップ抵抗である。プルアップ抵抗の抵抗値はフォトカプラによって異なるため、データシートを確認しなければならない^{*104}。

7.3.3 その他のインターフェース

7.3.3.1 フットスイッチ 足で操作するスイッチである。図 28 はヤマハのフットスイッチ FC5 である。ピアノのペダルを模した FC4A もある。2 極^{*105}のものは回路的にはただのタクト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^{*106}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。チャタリング^{*107}が起きたときに誤作動しないように、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 [30] などが参考になる。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り^{*108}、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。



図 28: ヤマハ FC5。画像は [ヤマハ公式サイト](#) から引用。

^{*103} IC は熱に弱いものが多く、さらに高価なことが多い。そのため、IC は回路に直接はんだ付けするのではなく、ソケットだけを付けておいて本体を着脱可能にする。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、実際にははんだ付けしたのはソケットである。

^{*104} プルアップ抵抗値はデータシートを確認しなければならないのだが、実際のところはだいたい 1kΩ ぐらいにしておけば問題ない。本当は 5V と 3.3V で使用できるフォトカプラも違って、例えば TLP552 などは 3.3V で動作することを保証していないが、3.3V ボードの MIDI IN で使用したことがある。真似をするなら自己責任で。

^{*105} 電子楽器に接続するインターフェースは、MIDI によるものとフォン端子によるものが存在し、フォン端子は絶縁体に区切られているいくつかの領域が存在する。この領域が 2 つのものを 2 極、3 つのものを 3 極と呼ぶ。オーディオを接続する場合、（乱暴に言って）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。

^{*106} 踏んでいないときに ON（ノーマルクローズ）なのか、OFF（ノーマルオープン）なのか、という違い。物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。

^{*107} 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。

^{*108} 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。

7.3.3.2 エクスプレッションペダル 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、ヤマハのシンセサイザー MONTAGE のスーパーノブのように、激しい音変化をアサインして使うことができるかもしれない。プラグは3極で、回路は可変抵抗。

7.3.3.3 ブレスコントローラ 息を吹く圧力を検知する。ウインドシンセサイザー^{*109}に搭載されているほか、圧力センサを買って自作することもできる。[\[31\]](#)などで実際にDIYしている^{*110}。

^{*109} 吹奏楽器を模したシンセサイザーのこと。しかし、ウインドシンセサイザーをただのブレスコントローラとして使用するのには、ウインドシンセサイザーの本体音源が勿体ないためお勧めしない。

^{*110} ブレスコントローラを自作する際にいくつか注意点がある。まず、ホースは着脱できるものが良い。[\[31\]](#)では鍵盤ハーモニカのホース・唄口を取り付けているが、この記事と全く同じセンサを手に入れられるとは限らないため、注文した圧力センサの穴の外径を調べ、それを覆うような太さのシリコンチューブをつけて使うことをお勧めする。その場合、唄口とシリコンチューブはシリコンシーラントなどで接着すると良い。また、タンギングが効くようにするために、唄口かホース（基本的には丈夫な唄口の方が良い。私は鍵盤ハーモニカ本体に直で挿すタイプの唄口を加工した。小径のハンドドリルなどを使用する）に小さな穴を開ける。圧力センサは6kPaまで測れるものが良いとされている[\[31\]](#)。

参考文献

- [1] ヤマハ ELS-02C 仕様 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
- [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
- [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
- [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
- [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
- [12] Dr. 青山の XG 解体新書 https://jp.yamaha.com/files/download/other_assets/9/321739/read_aoyama.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
- [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
- [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
- [18] ヤマハ PSR-SX600 Reference Manual https://jp.yamaha.com/files/download/other_assets/0/1346910/psrsx600_ja_rm_a0.pdf
- [19] All that I know about Electone files <http://serge45.free.fr/electone/texte.htm#b00evt>
- [20] 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
- [21] 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_f0.pdf
- [22] 初心者になるための耳コピ MIDI 講座 <http://mimikopi.nomaki.jp/>
- [23] STAGEA と PC 編曲の連携（東大エレクトラ資料、まさにこの名前でググれば運が良ければ見つかるかも。一時期、一瞬だけ、東大エレクトラのページが公開されていたことがあった。筆者と思われる kodack64 氏の github から東大エレクトラのサーバーへリンクが貼られている。）
- [24] Arduino のすすめ <https://n.mtng.org/ele/arduino/>

- [25] Arduino 日本語リファレンス <http://www.musashinodenpa.com/arduino/ref/>
- [26] Arduino MIDI Library の使い方 <https://qiita.com/yudai220/items/3bde9461f282d56d1ac2>
- [27] Electone × MIDI × Arduino <https://qiita.com/yudai220/items/b0b3dc6a8293780d5be2>
- [28] MIDI 1.0 電氣的仕様改訂 http://amei.or.jp/midistandardcommittee/Recommended_Practice/ca33-j.pdf
- [29] TETRASTYLE-dev-BLOG MIDI 受信回路 http://dev.tetrastyle.net/2011/07/midi_21.html
- [30] 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/
- [31] USB MIDI Breath Controller – Hackaday.io <https://hackaday.io/project/161678-usb-midi-breath-controller>