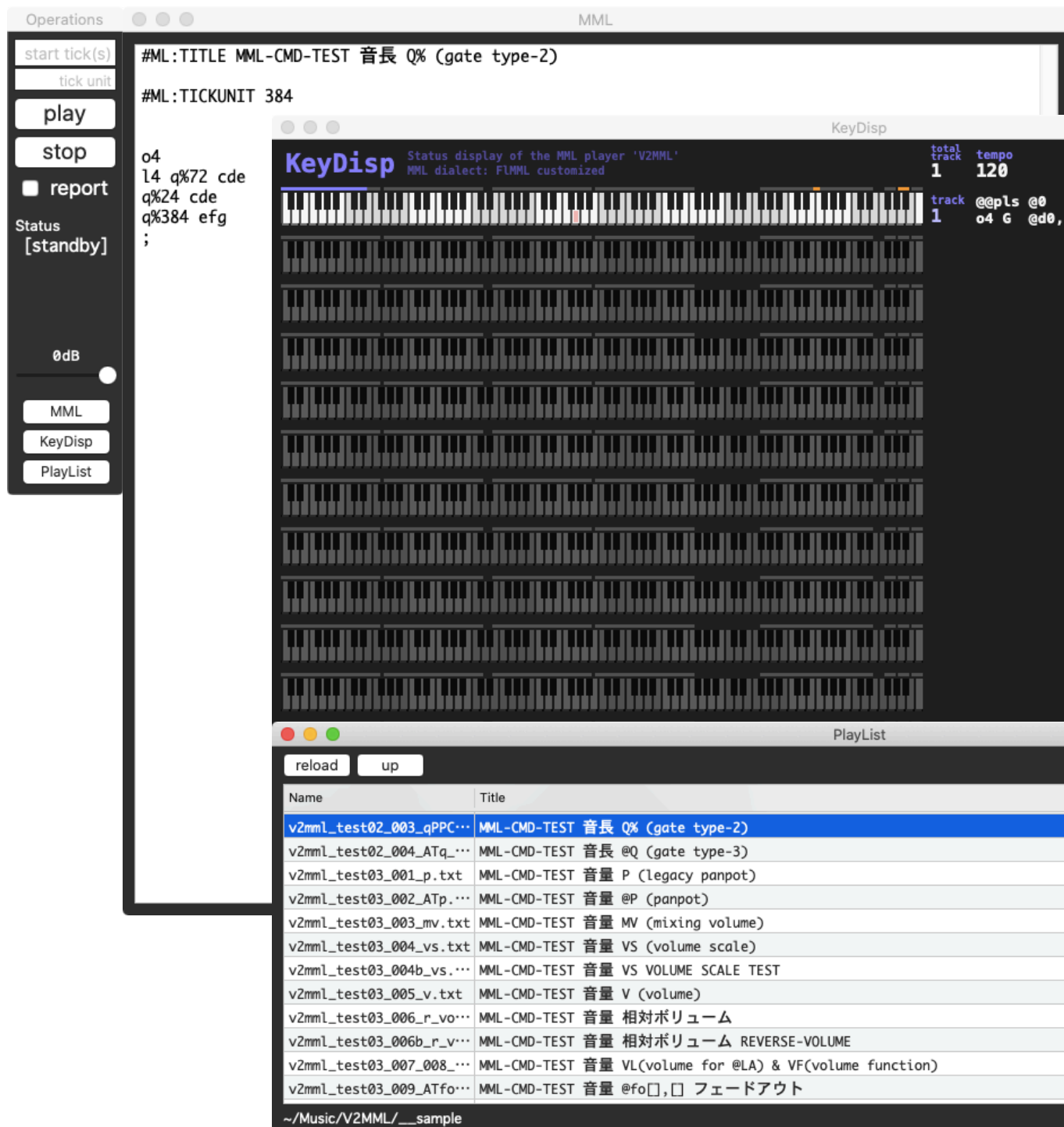


V 2 M M L 説明書

for V2MML Version 1.0.34
 最終更新日：2024-02-02 [A]
 著者：Akira SUZUKI



1. 基本情報	7
1.1. 特徴	8
1.2. 基本的な使い方	9
1.3. トラック定義	10
1.4. MML テキストの記述	11
・ MML 定義について	11
・ マクロ定義について	11
・ メタデータ定義について	12
2. 音程関連	13
2.1. CDE... : 音符の音名指定 (ドレミファソラシ)	13
2.2. 音程番号表	15
2.3. N : 音符の音程番号指定	16
2.4. & : タイまたはスラー	19
2.5. ^ : 音長短縮記号 (タイ拡張機能)	20
2.6. R : 休符	21
2.7. 0 : オクターブ	22
2.8. > または < : 相対オクターブ	23
2.9. PS : ピッチスケール	24
2.10. NS : ノートシフト	26
2.11. @NS : 相対ノートシフト	27
2.12. @D : デチューン	28
2.13. _ : ポルタメント	29
2.14. @U : MIDI風ポルタメント	30
2.15. #MB:PITCHSCALE : ピッチスケール定義	32
・ TypeID=0 のとき (1 2 平均律)	33
・ TypeID=10 のとき (F-Number)	34
・ TypeID=11 のとき (PSG分周比)	37
・ TypeID=12 のとき (PSG分周比Full Keyモード)	40
・ TypeID=20 のとき (WSGステップレート)	43
2.16. #ML:REVERSE-OCTAVE : 相対オクターブ記号反転	46
3. 音長関連	47
3.1. L : デフォルト音長	47
3.2. Q : ゲートタイム TYPE-1	48
3.3. Q% : ゲートタイム TYPE-2	50
3.4. @Q : ゲートタイム TYPE-3	51
3.5. #ML:TICKUNIT : TICKカウント単位設定	52

4. 音量関連	54
4.1. 音量設定の前提事項	54
4.2. デシベル計算	56
4.3. MV：ミキシングボリューム	58
4.4. @P：パンポット	59
4.5. P：レガシーパンポット	60
4.6. VS：ボリュームスケール	61
4.7. V：ボリューム (for @EA)	63
4.8.) または (：相対ボリューム (for @EA)	64
4.9. VL：ボリュームL (for @LA)	65
4.10. VF：ボリュームファンクション	66
4.11. @F0：フェードアウト	67
4.12. @FI：フェードイン	69
4.13. #MB:VOLUMESCALE：ボリュームスケール定義	71
・TypeID=0 のとき (配列定義モード)	72
・TypeID=1 のとき (線形モード)	75
・TypeID=2 のとき (デシベルモード)	77
4.14. #ML:REVERSE-VOLUME：相対ボリューム記号反転	80
5. 制御関連	81
5.1. プリプロセッサの処理順序	81
5.2. T：テンポ	82
5.3. ; (セミコロン)：トラック区切り	85
5.4. /* ... */：コメントブロック	86
5.5. //：コメントライン	87
5.6. : ... : ：繰り返し	88
5.7. @RP：無限リピートエントリ	90
5.8. !：MML ステータス表示	93
5.9. !#0：音符ログ	94
5.10. !#1：休符ログ	95
5.11. @DSP：KeyDispゲージへの表示指示	96
5.12. \$名前=内容;：マクロ定義	98
・引数つきマクロ定義	101
5.13. #ML:REPORT-TICKS：総TICKカウント数の表示	102
6. 音色関連	103
6.1. @@：使用音源モジュール指定	103
6.2. @：音源サブモジュール番号指定	105
6.3. @@SIN：サイン波音源	106

6.4.	@@TRI : 三角波音源	109
6.5.	@@SAW : ノコギリ波音源	112
6.6.	@@PLS : パルス波音源	115
6.7.	@@WVM : 波形メモリ音源	118
6.8.	@@FMS : F M音源	122
	・ Yコントロール (1) 音源モジュール設定	123
	・ Yコントロール (2) 音色レジスタアクセス体系	127
	・ Yコントロール (3) 特殊動作対応 : Y768(Y0x300)以降	131
6.9.	@@NZW : ホワイトノイズ音源	133
6.10.	@@NZP : P S Gノイズ音源	134
6.11.	@@NZF : F Cノイズ音源	136
6.12.	@@NZG : G Bノイズ音源	138
6.13.	@@NZC : 波形メモリノイズ音源	140
6.14.	@@DPC : D P C M音源	142
6.15.	@@PCM : P C M音源	143
6.16.	@@ADP : A D P C M音源	145
6.17.	@MH : @@FMS用 : O P M互換H L F O	147
6.18.	@MHA : @@FMS用 : O P N A互換H L F O	149
6.19.	@MHZ : @@FMS用 : H L F O無効	151
6.20.	@MR : @@FMS用 : A R初期レベルモード	152
6.21.	@MZ : @@FMS用 : カレントENV-LVの即時減衰機能	153
6.22.	@MZE : @@FMS用 : @MZ機能の継続適用	154
6.23.	@PH : ノートオン時の位相リセット	155
6.24.	@0 : オペレートモード	158
6.25.	@W : @@PLS用 : デューティ比	161
6.26.	@N : ノイズ音源用のサイクル	162
6.27.	@NR : ノイズ音源用のシフトレジスタ	165
6.28.	@NC : @@NZP用 : ノイズサイクル設定	168
6.29.	Y : Yコントロール	169
	・ L F O波形モジュールへの適用	171
6.30.	@DLY : ディレイエフェクト	172
6.31.	@F : フィルタ	174
6.32.	@FZ : フィルタ停止	175
6.33.	@~ : フォルマントフィルタ	176
6.34.	#ML:FILTER : フィルタデータ定義	177
6.35.	#ML:NZCSTEPM : @@NZC用 : 更新ステップ倍率設定	179
6.36.	#ML:OVSWAVEM : @@WVM用 : オーバーサンプリング設定	180
6.37.	#ML:WAVEM : @@WVM用 : 波形データ定義	181
6.38.	#ML:FMDLOCK : @@FMS用 : マスタークロック設定	183
6.39.	#ML:FMFBBIAS : @@FMS用 : F Bバイアス設定	185
6.40.	#ML:FMHLFOS1 : @@FMS用 : H L F O速度調整 1	186
6.41.	#ML:FMHLFOS2 : @@FMS用 : H L F O速度調整 2	187

6.42. #MB:FM@SS60P : @@FMS用 : 音色データ定義 (6 オペレータ)	188
6.43. #MB:FM@SS40P : @@FMS用 : 音色データ定義 (4 オペレータ)	190
6.44. #MB:FM@SS20P : @@FMS用 : 音色データ定義 (2 オペレータ)	192
6.45. #MB:FM@S : @@FMS用 : 旧形式音色定義 (4 オペレータ)	194
6.46. #MB:FM@OPM : @@FMS用 : 音色データ定義 (OPM互換)	196
6.47. #MB:FM@OPNA : @@FMS用 : 音色データ定義 (OPNA互換)	198
6.48. #MB:FM@OPN : @@FMS用 : 音色データ定義 (OPN互換)	200
6.49. FM音源の音色作成に関わる用語説明	202
. H L F O (hardware low frequency oscillator)	202
. エンベロープジェネレータ部	204
. フェーズジェネレータ部	206
. オペレータ同士の接続	208
6.50. FM音源6オペレータモードの接続形態	209
6.51. FM音源4オペレータモードの接続形態	226
6.52. FM音源2オペレータモードの接続形態	230
6.53. #ML:DPCM : @@DPC用 : 波形データ定義	231
6.54. #ML:U8PCM : @@PCM用 : 波形データ定義	233
6.55. #ML:ADPCM : @@ADP用 : 波形データ定義	235
6.56. #ML:EXBIN@FILE : バイナリファイル読み込み	237
6.57. 読み込むバイナリファイルのフォーマット	238
6.58. #ML:SMP@EXBIN : 一時バイナリ領域からの波形データ定義	240

7. エンベロープ関連 243

7.1. @E0 : エンベロープオプション	243
7.2. @EA : 音量エンベロープ	246
7.3. @EF : フィルタエンベロープ	247
7.4. ZS : 音量エンベロープのダンパー設定	248
7.5. @Z : エンベロープダンパー実行	250
7.6. #MB:ENV@A : 音量エンベロープデータ定義	251
7.7. #MB:ENV@F : フィルタエンベロープデータ定義	256
7.8. #ML:ENVCLOCK : エンベロープ時間単位	261
7.9. #ML:ENVRESOL : エンベロープ解像度	263
7.10. #ML:ENVDAMPD : @EA用 : ダンパーデフォルト	265
7.11. #ML:FMSDAMPD : @@FMS用 : ダンパーデフォルト	266

8. L F O 関連 267

8.1. @L0 : L F O オプション	267
8.2. @LP : ピッチ L F O	270
8.3. @LA : 音量 L F O	271
8.4. @LB : パンポットバランス L F O	272

8.5. @LF：フィルタ L F O	273
8.6. @LY：Yコントロール L F O	274
8.7. L F Oの停止	275
8.8. L F Oの再スタート	276
8.9. L F Oテーブル番号変更	277
8.10. #ML:LF0@P：ピッチ L F Oデータ定義	279
8.11. #ML:LF0@A：音量 L F Oデータ定義	282
8.12. #ML:LF0@B：パンポット L F Oデータ定義	285
8.13. #ML:LF0@F：フィルタ L F Oデータ定義	289
8.14. #ML:LF0@Y：Yコントロール L F Oデータ定義	292
8.15. #ML:LF0CLOCK：L F O時間単位	295
8.16. #ML:LF0RESOL：L F O解像度	297
8.17. #MB:LF0TBL@A：L F Oテーブル定義 (A)	299
8.18. #MB:LF0TBL@R：L F Oテーブル定義 (R)	303
8.19. #ML:LF0TBLPM：L F Oテーブル処理モード	305
8.20. L F O波形グラフ (1) (@LP/@LB/@LF)	306
8.21. L F O波形グラフ (2) (@LA)	310

9. 発音数関連 312

9.1. #ML:USING-POLY：ポリフォニックモード有効化	312
9.2. @PL：ポリフォニックモード発動	314
9.3. @PM：ポリフォニック方式の選択	316
9.4. [...]：和音記法	317

10. 装飾関連 319

10.1. XA：パラメータへの加算	319
10.2. XM：パラメータへの乗算	321
10.3. @KT：キーイベント (トリガモード)	323
10.4. @KC：キーイベント (音程モード)	326
10.5. @KL：キーイベント (音長モード)	329
10.6. #MB:KEYEVENT@TRG：@KT用キーイベントデータ定義	332
10.7. #MB:KEYEVENT@CODE：@KC用キーイベントデータ定義	334
10.8. #MB:KEYEVENT@LEN：@KL用キーイベントデータ定義	336

11. 付加情報関連 338

11.1. #ML:TITLE：曲のタイトル	338
11.2. #ML:ARTIST：曲のアーティスト	339
11.3. #ML:COMMENT：曲のコメント	340
11.4. #ML:CODING：MML作成者名	341

1. 基本情報

V2MMLは、Mac用のMML演奏および、MML作成支援アプリです。

動作環境：

macOS 10.15 以降。

波形生成：

全てのトラックにおいて、サンプリングレート 192kHz、変位 64bit浮動小数点数にて生成。

波形再生：

macOS の AudioUnit の都合、全トラックの加算合成結果を、192kHz, 32bit浮動小数点数に変換して受け渡し、再生。

トラック数：

CPUパワーまたはメモリが許す限り（FM音源が最も処理が重いです）。

エンベロープ：

音量、フィルタ。

LFO：

ピッチ、音量、パンポット、フィルタ、音源制御CMD(Y-control)。

エフェクタ：

ディレイ。

音源モジュール：

全てのトラックにおいて、次の仮想音源を全て持っており、1つだけ選択して使用可能（選択変更は随時可能）。

- サイン波
- 三角波
- ノコギリ波
- パルス波（DUTY比可変）
- 波形メモリ音源
- FM音源（OPMベース）
- ホワイトノイズ
- PSGノイズ
- FCノイズ
- GBノイズ
- 波形メモリノイズ
- FC-DPCM音源
- PCM音源（8bit/16bit）
- ADPCM音源（4bit）

1.1. 特徴

1つのトラック定義で、複数の内部チャンネルを駆動する「ポリモード」をV1MMLから踏襲しています。これにより、1つのトラック定義で和音を鳴らしたり、リリース音が重なるような制御ができます。

波形メモリ音源、パルス波、ノコギリ波、サイン波、三角波においては、複数のオペレータを持ち、現在の仕様では7重まで定義でき、単純コーラスや、オクターブ違いユニゾン、和音を音源側で構成することができます。（音源側の複数オペレータではパンポットを別にしたり、ノートオンの時間差は作れません）

FM音源モジュール以外の仮想音源のエンベロープは、最大2048個までのマルチポイントで節を設定可能で、ループも設定できます。（ループ機能は、トレモロ効果を想定）

仮想音源の音程管理を64bit浮動小数点数化（細かいcentずれ指定が受け付け可能）。さらに、複数の音階スケールパターンにも対応。

PCM音源・ADPCM音源は、2kHz～192kHzまでのサンプリング周波数に対応。

OPMベースFM音源モジュールの機能拡張

全てのオペレータにセルフフィードバックを設定可能。

マルチプルの64bit浮動小数点数指定が可能。

フィードバックの64bit浮動小数点数指定が可能。

DETUNE3 新設（64bit浮動小数点数のcent単位の周波数ずれ指定）。

DETUNE4 新設（64bit浮動小数点数のHz単位の周波数ずれ指定）。

オペレータごとに、キーオン時のエンベロープ初期出力レベルの指定が可能。

（従来のレベル引き継ぎモードに加え、強制的に初期レベルを指定可能）

オペレータごとに、キーオン時のサイン波の初期位相を指定可能。

（従来の0スタートに加え、好きな位置からスタート可能）

サインテーブル及びエンベロープの解像度を向上。

（FM音源特有のシュワシュワいうノイズを低減）

6オペレータモード（重厚化）と、2オペレータモード（軽量化）も選択可能。

（例えば6オペは旋律、2オペは和声、4オペはリズムに振る、などが可能）

1.2. 基本的な使い方

基本的な使い方は、

- (1) MMLウィンドウに演奏内容をテキスト入力する
 - (2) OperationsウィンドウのPLAYボタンをクリックする
- 以上です。

MML (Music Macro Language)とは、プレーンテキストで書かれた音楽演奏用の言語です。いわばテキストで書かれた楽譜のようなものです。

ファイル読み込み時のプレーンテキストのエンコードは、

- ・ UTF-8
- ・ BOM無し
- ・ 改行コードはLFのみ

の形式を想定しています。

V 2 MMLのMMLコマンドは、すべて半角文字で記述します。全角文字は、コメントエリア、もしくは曲名などのメタデータ内でのみ使用できます。

【サポートサイト】

<https://mmlsound.sakura.ne.jp/>

1.3. トラック定義

V2MMLでは、セミコロン「;」がトラック定義の区切り文字としての役割を果たしています。例えば、

```
ccc;  
eee;  
ggg;
```

このように書くと、トラックが3本定義され、ドミソの和音が3回発音されます。

テキスト先頭から最初のセミコロンまでが第1トラック、さらにその次のセミコロンまでが第2トラックといった具合です。この要領で、次々とトラックを増やすことができます。

セミコロンは、厳密には次のトラックを開始する命令と解釈されるので、この場合、最後のセミコロンは無くても正常に演奏されます。しかし、後からトラック追加する時に備える意味で、トラックの終端には常にセミコロンを置いておくのが無難です。演奏情報が含まれない空のトラック定義は、演奏前に自動的に破棄されます。

1.4. MML テキストの記述

MML テキストの記述で定義する内容は、大きく分けると次の3種類です。

- ・ MML 定義（トラック定義群。実際に発音させる内容）
- ・ マクロ定義
- ・ メタデータ定義

再生要求を受けると、MML テキストはコンパイルされてから演奏されます。コンパイルされる内容のうち、時間ごとにシーケンスされるのは、MML 定義の部分だけです。

MML 定義以外の部分は、コンパイル前の処理（プリプロセッサ）により、波形データに変換されたり、文字置換されたりするため、直接のシーケンスデータにはなっていない点に注意してください。

・ MML 定義について

「cdefg;」というように、発音させたい内容を書いてトラック定義します。MML 定義内では、前もって定義したマクロを「\$名前」で参照することも出来ます。MML 定義内容をマクロ参照だけで満たしても構いません。ひとつの発音トラックは、基本的に単音（和音にならない）フレーズを鳴らす内容になります。しかしながら、ひとつの発音トラックで和音を鳴らす、特別な定義方法もあります。また、MML コマンドでは大文字・小文字の区別をしないので、どちらでも書けます。

・ マクロ定義について

トラック先頭（MML コマンドを1つも記述していない状態のトラック）で、「\$名前=内容;」という書き方をすると、マクロとして定義されます。マクロの内容には改行を含んでも良いので、長いフレーズの定義も可能です。また、トラック先頭であれば、複数のマクロを定義できます。定義されたマクロは、それ以降に記述されたトラック内にて「\$名前」で参照され、「内容」に置き換えられます。この置き換えは、プリプロセッサにより行われます。すでに定義されたマクロを、以降のマクロ定義で参照するような、入れ子的なマクロ定義も可能です。マクロ定義はセミコロンで締め括りますが、トラック定義とは別物と考えてください。

トラック先頭において「\$」で始まる部分は、マクロ定義モードと解釈されません。

しかし、マクロ定義後に、1つでもMMLコマンドを記述すると、トラック定義モードとなり、「\$」は定義済みマクロの参照開始とみなされますので、MML記述中にマクロを定義することは出来ません。

また、マクロ名（\$名前=内容;の名前の部分）では、大文字・小文字が区別されるので注意してください。

・メタデータ定義について

メタデータとは、データのためのデータ、のことです。

MMLのモード設定や、波形データのユーザー定義など、付随情報全般をサポートする記述です。メタデータの定義は、行頭から「#」を付けたコマンドで行いますが、大きく2種類に分けられます。

(1) メタライン（#ML:）のグループ

(2) メタブロック（#MB:）のグループ

メタラインでは、改行を定義終端として認識しています。

メタブロックでは、中かっこ { } によってデータを括弧括弧で括弧括弧して、中かっこ閉じ「}」を定義終端として認識するため、途中で改行が入ることが許されます。

メタライン、メタブロック共に、終端にセミコロンは不要です。

メタデータを記述する位置は、MML定義の前でも後でも大丈夫です。

これはメタデータもマクロ同様、MMLが解釈される前にテキスト全体検索で情報取得後、すぐに除去されるため、MMLが解釈される時には無かったことにされているからです。

また、メタデータ定義のコマンド名では大文字・小文字が区別されるので注意してください。

メタデータ定義がグループ化されている理由は、検索回数を最小限に抑え、コンパイル時間を短くするためです。

2. 音程関連

2.1. CDE... : 音符の音名指定 (ドレミファソラシ)

記述	C D E F G A B
種別	MML コマンド

【解説】

音名をアルファベットで指定し、ノートオンします。
(オクターブは別途 0 コマンド等で指定します)

トラック先頭での、デフォルトのアルファベットと音名の対応は次の通りです。

C = ド

D = レ

E = ミ

F = ファ

G = ソ

A = ラ

B = シ

音名の直後にプラス (+) をつけると、半音上がります。

音名の直後にマイナス (-) をつけると、半音下がります。

ダブルシャープ (++)、ダブルフラット (--) の記述も受け付けます。

シャープ・フラット(+,-)の後に数字をつけると、音長を指定できます。

音長の数値を省略すると、Lコマンドによるデフォルト音長が指定されたものと解釈します。

音長の後にピリオド (.) を付けると付点音符になります。

複数の付点の記述も受け付けます。

音長の延長 (&) や、短縮 (^) も可能です。

延長や短縮する音長の数値は省略できません。

【例1】 8分音符のド

C+8

【例2】 16分音符のソのダブルシャープ

G++16

【例3】 付点16分音符のラ♭

A-16.

付点を1個つけた場合は音長が1.5倍、2個付けた場合は1.75倍になります。
A4.. と A4&8&16 は同じ長さです。

【例4】 音長96tickカウントのド

C%96

【例5】 デフォルト音長を使った記述

L4 C D8 E8 F G

この場合、ド、ファ、ソの音長が4分音符になります。

2.2. 音程番号表

【解説】

V2MMLでは、音程に番号を振って、音程番号を管理しています。
オクターブ0の「C（ド）」の音を0番とし、半音上がるごとに番号が1増えることとしています。

デフォルトでは、オクターブ4の「A（ラ）」の音に440Hzの周波数を割り当てて、それを基準に他の音程には12平均律に従った周波数を割り当てています。

【音程番号表】（octave/key）※o-2からo11までの抜粋

	C	C+	D	D+	E	F	F+	G	G+	A	A+	B
o-2	-24	-23	-22	-21	-20	-19	-18	-17	-16	-15	-14	-13
o-1	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
o0	0	1	2	3	4	5	6	7	8	9	10	11
o1	12	13	14	15	16	17	18	19	20	21	22	23
o2	24	25	26	27	28	29	30	31	32	33	34	35
o3	36	37	38	39	40	41	42	43	44	45	46	47
o4	48	49	50	51	52	53	54	55	56	57	58	59
o5	60	61	62	63	64	65	66	67	68	69	70	71
o6	72	73	74	75	76	77	78	79	80	81	82	83
o7	84	85	86	87	88	89	90	91	92	93	94	95
o8	96	97	98	99	100	101	102	103	104	105	106	107
o9	108	109	110	111	112	113	114	115	116	117	118	119
o10	120	121	122	123	124	125	126	127	128	129	130	131
o11	132	133	134	135	136	137	138	139	140	141	142	143

- ・ KeyDispの表示では、音程番号 0 ~ 120 の範囲に応答します。
(0 未満は 0、120 より大きい場合は 120 として表示されます)

2.3. N：音符の音程番号指定

記述	N[1],[2]
種別	MML コマンド

【解説】

音程番号を指定することで、ノートオンします。

指定する引数は次の通りです。

引数[1]...音程番号 (noteNo)

引数[2]...音長 (length)

引数はカンマで区切って指定します。カンマとlengthの組は省略できます。

引数[1] (noteNo)

発音したい音程 (周波数番号) を、数値で指定します。

指定する数値のパターンは、ピッチスケール (PSコマンド) の設定によって変わります。

モード0 のとき (12平均律) :

noteNo は、音程番号×デチューン解像度 で指定します。

この場合の noteNo は、小数以下の指定も受け付けます。

例えばデフォルトの PS0 のとき、デチューン解像度が 100 ならば、オクターブ4のラの4分音符の記述は「n5700,4」となります。

モード10 のとき (Block/F-Number) :

noteNo は、14ビットの整数値で指定します。

14ビットのうち、上位3ビットはBlock (オクターブ相当) で、0 ~ 7 の範囲で指定できます。

下位11ビットは F-Number で、1~2047 の範囲で指定できます。F-Number には0は指定できないので注意してください。

したがって、PS10の場合の noteNo は、14ビットの数値である 0 ~ 16383のうち、下位11ビットが0にならない値が設定可能なものになります。

また、PS10の場合の noteNo は、通常の10進数の整数指定のほか、接頭語に「0x」を付けることで16進数で記述することもできます。

例えば「n0x2410,4」となります。

モード11 または 12 のとき (PSG分周比) :

noteNo は、0より大きい、31ビットの整数値で指定します (0は指定不可)。

「#MB:PITCHSCALE」で定義したPSG分周比のビット幅に合うよう指定してください。また、接頭語に「0x」を付けることで16進数で記述することもできます。例えば「n0x8d,4」となります。

モード20 のとき (WSGステップレート) :

noteNo は、0より大きい、31ビットの整数値で指定します (0は指定不可)。「#MB:PITCHSCALE」で定義したWSGステップレートのビット幅に合うよう指定してください。また、接頭語に「0x」を付けることで16進数で記述することもできます。

引数[2] (length)

noteNo の後にカンマで区切って、発音したい音長を整数で指定します。カンマとlengthのセットを省略した場合、Lコマンドによるデフォルト音長が指定されたものと解釈します。

音長の後にピリオド (.) を付けると付点音符になります。

複数の付点の記述も受け付けます。

音長の延長 (&) や、短縮 (^) も可能です。

延長や短縮する音長の数値は省略できません。

次の例1～例6は、ピッチスケールのモード0、デチューン解像度100の場合です

【例1】8分音符のオクターブ4のド#

N4900,8

【例2】付点16分音符のオクターブ4のラb

N5600,16.

付点を1個つけた場合は1.5倍、2個つけた場合は1.75倍の音長になります。

【例3】2種類の音長表現

N5600,4..

N5600,4&8&16

上記2種類は同じ長さです。

【例4】音長96tickカウンターのオクターブ4のド

N4800,%96

【例5】デフォルト音長を使った記述

```
L4 n4800 n5000,8 n5200,8 n5300 n5500
```

この場合、n4800、n5300、n5500 の音長が4分音符になります。

【例6】ポルタメント記述

```
n4800_n5700,4&n5700,1
```

この場合、o4c から o4a まで4分音符でポルタメントし、そのあと全音符で o4a を維持します。

2.4. & : タイまたはスラー

記述	&
種別	MML 擬似コマンド

【解説】

タイは、同じ音の高さで音長を伸ばします。

スラーは、異なる音の高さで、切れ目なく（ゲートタイムも一時無効になりつつ）発音します。

【例1】

A4&16&8

これはタイの記述です。この記述方法は、休符でも使えます。

4分音符のラを、16分音符分延長し、さらに8分音符分延長します。この場合、ゲートタイムの設定は合計音長に対して作用します。

仕様上、Lでデフォルト音長を指定した際に省略できる音長記述は、音名表記直後の音長だけになります。

【例2】

C4&D4&E4

これはスラーの記述です。この記述方法は、休符には使えません。

CとD、DとEの間が切れ目なく（ゲートタイム一時無効で）発音されます。

この場合、Eにのみ、指定のゲートタイムが作用します。

【例3】

A4&A16&A8

この記述はタイのように見えますが、スラーの記述と見なされます。

タイとの違いは、ゲートタイムが最後の「A8」にしか作用しない点です。

2.5. ^ : 音長短縮記号 (タイ拡張機能)

記述	^
種別	MML 擬似コマンド

【解説】

タイでは音長を延長しますが、この記号を使用して音長を繋いだ場合には、音長が短縮されます。

この機能は、MMLで装飾音を書く場合に、例えば64分音符を使って、あとから装飾の64分音符を短縮して帳尻を合わせる際に、短縮結果の音長を書く部分を簡単に書けるようにします。短縮記号は休符にも使用できます。

【例1】

```
G64 A4^64
```

64分音符で装飾音Gを発音してから、64分音符だけ短縮した4分音符のAを発音します。

【例2】

```
G%2 A4^%2
```

2 ticksで装飾音Gを発音してから、2 ticksだけ短縮した4分音符のAを発音します。

【例3】

```
G%2 R4^%2
```

2 ticksで装飾音Gを発音してから、2 ticksだけ短縮した4分休符を置きます。

2.6. R：休符

記述	R
種別	MML コマンド

【解説】

直前までノートオンされていればノートオフし、音符同様の長さ指定に従って休符が実行されます。

休符でも音符のように、長さの延長 (&) や、短縮 (^) が出来ます。

ただし、スラー (R4&R4) の記述はエラーとなります。

【例 1】

```
L16 C4 R D8 R8 R.
```

この場合、「R」は 1 6 分休符、「R.」は付点 1 6 分休符になります。

【例 2】

```
C4 R2&8 C4
```

この場合、2 分休符を 8 分休符延長しています。

【例 3】

```
C64 R8^64 C4
```

この場合、8 分休符を 6 4 分休符短縮しています。

【例 4】

```
C4 R8&R8 C4
```

この場合、エラーとなります。

2.7. 0 : オクターブ

記述 0[1]

種別 MML 擬似コマンド

【解説】

オクターブを整数値で指定します。

引数[1]には、オクターブ数を整数で指定します。
トラック先頭におけるデフォルトは、04 です。

オクターブ数そのものに制限は設けていませんが、
オクターブ指定を加味した音程に対応する、音程番号の範囲は、使用するピッチスケールに応じて次のように制限されます。（音程番号とは）

ピッチスケール	音程番号の範囲
mode:0 (12平均律)	-240 ~ +240
mode:10 (F-Number)	-24 ~ +24 のいずれかを開始音程番号として、 以降の96個の範囲
mode:11,12,20 (PSG/WSG)	0 ~ 120

(KeyDispの表示では、音程番号 0 ~ 120 の範囲に応答します)

【備考】

高いオクターブ設定では、再生サンプリング周波数の限界によって波形描画品質が低下する恐れがありますのでご注意ください。

【例】

04 CDEFG 05 EDCEG

2.8. > または < : 相対オクターブ

記述	> または <
種別	MML 擬似コマンド

【解説】

当コマンド指定時の 0 コマンド の設定状態から、相対的に数値変更したオクターブに設定します。

> オクターブ値を 1 上げます。(04 だった場合 05 にする)

< オクターブ値を 1 下げます。(04 だった場合 03 にする)

2 以上変更したい場合は、複数個記述するか、数字を添えて記述します。

>3 オクターブ値を 3 上げます。

<3 オクターブ値を 3 下げます。

不等号の向きと上下の対応を反転したい場合は、「#ML:REVERSE-OCTAVE」によって反転できます。

【例 1】

```
04 CEG>DFA;
```

この場合、04 で CEG を演奏し、次に 05 で DFA を演奏します。

【例 2】

```
03 CEG >3 DFA;
```

この場合、03 で CEG を演奏し、次に 06 で DFA を演奏します。

2.9. PS : ピッチスケール

記述	PS[1],[2]
種別	MML コマンド

【解説】

ピッチスケールを設定します。ピッチスケールでは、音程名または音程番号から発音周波数を算出する方式を選択します。

トラック先頭でのデフォルトは、12平均律に設定されています。

指定する引数は次の通りです。

引数[1]...ピッチスケール種別 (psType)

引数[2]...ピッチスケール番号 (psNo)

引数はカンマで区切って指定します。カンマとpsNoの組は省略できます。

トラック先頭におけるデフォルトは、PS0 です。

引数[1] (psType)

psTypeは、0、10、11、12、20のいずれかで指定します。

psType=0 の場合

12平均律モードで音程を計算します。当コマンドを指定しなかった場合のデフォルト設定は、このモードになります。

#ML:TUNING では、このモードの初期設定に対し変更を行います。

デチューン解像度 の設定は、このモードに対してのみ作用します。

psType=10 の場合

F-Numberモードで音程を計算します。

ユーザー定義を行っていないデフォルト設定では、3.9936MHzのOPNのF-Numberシミュレートで、mucom88互換のモードになります。

psType=11 の場合

PSG分周比モードで音程を計算します。

ユーザー定義を行っていないデフォルト設定では、3.9936MHzのPSG分周比シミュレートで、mucom88互換のモードになります。

psType=12 の場合

PSG分周比Full Keyモードで音程を計算します。

ユーザー定義を行っていないデフォルト設定では、1.79MHzのPSGシミュレートモードになります。

psType = 20 の場合

WSGステップレートモードで音程を計算します。

ユーザー定義を行っていないデフォルト設定では、24000.0HzのWSGシミュレートモードになります。

引数[2] (psNo)

psNoの指定は省略することができます。

省略した場合は、システム側で持っている、psTypeに応じたデフォルトパターンに設定されます。

psNoを指定する場合は、あらかじめ「#MB:PITCHSCALE」でピッチスケールパターンをユーザー定義し、その定義番号をpsNoに指定します。

【例1】

```
ps0 ccccc;  
ps10 eeeee;  
ps11 ggggg;
```

このように書くと、1トラック目は12平均律、2トラック目はF-Numberピッチ、3トラック目はPSGピッチの計算で各キーの周波数が計算されます。

この場合、それぞれのピッチスケールパターンは、システム側で保持しているデフォルトパターンが使用されます。

【例2】

```
ps11,3 cdefg;
```

この場合、#MB:PITCHSCALE 3 {...} でユーザー定義したピッチスケールパターンで、PSG分周比モードにより周波数が計算されます。

2.10. NS : ノートシフト

記述	NS[1]
種別	MML 擬似コマンド

【解説】

ノートシフトを指定します。

引数[1]には、半音単位でキートランスポーズさせる数値を整数で指定します。トラック先頭におけるデフォルトは、NS0 です。

ノートシフト指定に範囲制限はありませんが、ノートシフト指定を加味した音程番号の範囲は、
-240 ~ +240
に制限されます。（音程番号とは）

【例】

```
NS0 CDE NS-1 CDE NS-1 CDE
```

このように書くと、二度目のドレミは半音下がります。三度目のドレミも、二度目と同様、半音下がったドレミです。

2.11. @NS : 相対ノートシフト

記述	@NS[1]
種別	MML 擬似コマンド

【解説】

ノートシフトを相対値で指定します。

引数[1]には、半音単位で相対的にキートランスポーズさせる数値を整数で指定します。（NSコマンドで指定された状態への相対加算を実行します）

ノートシフト指定に範囲制限はありませんが、
ノートシフト指定を加味した音程番号の範囲は、

-240 ~ +240

に制限されます。（音程番号とは）

【例】

```
NS0 CDE @NS-1 CDE @NS-1 CDE NS0 CDE
```

このように書くと、二度目のドレミは半音下がります。
三度目のドレミは、二度目よりさらに半音下がったドレミです。
四度目のドレミは、ずれのないドレミになります。

2.12. @D : デチューン

記述 @D[1],[2]

種別 MML コマンド

【解説】

細かい音程のずれを与えます。

指定する引数は次の通りです。

引数[1]...音程のずれ (detune)

引数[2]...解像度 (reso)

引数はカンマで区切って指定します。カンマとresoの組は省略できます。トラック先頭におけるデフォルトは、@D0,100 です。

引数[1] (detune)

正数で音程を上げ、負数で音程を下げます (単位はresoに依存します)。

(この引数[1]では、加減算記述も可能です)

デフォルトはゼロ (音程ずれ無し) です。

少数以下の指定も受け付けます。

設定範囲は、reso が 100 であれば (半音を100分割)、
-9600 ~ +9600 (-96 ~ +96 の reso倍)

になります。

引数[2] (reso)

デチューン解像度 (半音の音程比を何分割するか) を指定します。

1 2 平均律のピッチスケールにのみ有効です。

デフォルトは 100 (1 cent単位) です。

resoは省略可能で、省略した場合は以前に設定された値が採用されます。

設定範囲は 10 ~ 1000 の整数で、範囲外の指定は無視されます。

トラック先頭で1度 reso を指定したら、以降のデチューン設定は reso を省略する使い方を想定しています。解像度が 1 cent で構わない場合は、トラック先頭から reso を省略してください。

【例】

```
@D0,64 CDEFG
```

```
@D6 GFEDC
```

上記 @D6 記述後の音程は、半音の 6/64 だけ音程が上がります。

2.13. _ : ポルタメント

記述	_ (アンダーバー)
----	------------

種別	MML コマンド
----	----------

【解説】

ポルタメント（音程の滑らか変化）です。
音名を _ で繋げるとポルタメントになります。
ピッチスケールが1 2 平均律モードの場合に使用できます。

【例1】

```
C_>C4
```

ドからオクターブ上のドまで4分音符の長さで滑らかに変化させます。

【例2】

```
C1&C_>C4&C2.
```

ドを全音符で鳴らした後、オクターブ上のドまで4分音符の長さで滑らかに変化させ、その後符点2分音符の長さで音程を維持します。

【例3】

```
C_>C1&1&1&1
```

全音符以上の長さで変化をさせたい場合はタイ（音長の延長）で実現できます。この例では、全音符4つ分の長さで、ドからオクターブ上のドまで滑らかに変化させています。

<次のような記述は出来ません>

【例4】

```
C_<C4_>C4
```

この場合はエラーになります。

2.14. @U：MIDI風ポルタメント

記述	@U[1],[2]
種別	MMLコマンド

【解説】

MIDI風のポルタメントです。

指定する引数は次の通りです。

引数[1]...機能番号 (code)

引数[2]...パラメータ (param)

引数はカンマで区切って指定します。

トラック先頭におけるデフォルトは、@U0 です。

codeは必須で、0、1、2、3 の4種類で指定します。

codeの値によって、機能とparamの指定内容が変わります。

code=0のとき (param不要)

MIDI風ポルタメントをOFFにします。

MIDIで言う所の CC#65 に相当します。

【例】

```
@U0
```

code=1のとき (param不要)

MIDI風ポルタメントをONにします。

MIDIで言う所の CC#65 に相当します。

【例】

```
@U1
```

code=2のとき (paramは数値で指定)

MIDI風ポルタメントの変化割合を指定します。

paramは数値で指定し、範囲は 1 ~ 127 で、少数以下の指定も受け付けます。

1 では早く、127 ではゆっくり音程変化します。

MIDIで言う所の CC#5 に相当します。

【例】

```
@U2,40 @U1 CG<C
```

code=3のとき (paramは音程番号か音程文字)

MIDI風ポルタメントの開始音程を指定します。

MIDIで言う所の CC#84 に相当します。

通常の音程変化は直前の音程からのスタートですが、このコマンドを使用すると、paramによって変化の開始音程を指定することが出来ます。

変化の開始音程が有効になるのは、このコマンドが指定された直後の音程のみで、その次からはまた直前の音程からのポルタメントとなります。

paramによる変化の開始音程の指定方法は次の 2 種類があります。

- ・ 音程番号を直接指定する方法 (音程番号とは)
- ・ 音程を文字として指定する方法

【例 1】

```
04E @U3,48 GAB
```

G の音程は、04Eからではなく、04Cからの音程変化になります。

【例 2】

```
04E @U3,04D- GAB
```

G の音程は、04Eからではなく、04D-からの音程変化になります。

開始音程の文字指定の場合は、

オクターブ、

音名、

フラット・シャープ (-,+,#)

のみが使用できます。

相対オクターブ変更記号は使用出来ないので注意してください。

2.15. #MB:PITCHSCALE : ピッチスケール定義

記述 #MB:PITCHSCALE [1] {[2]}

種別 メタデータ定義

【解説】

ピッチスケールコマンド (PS) で使用する、ユーザー定義のピッチスケールパターンを定義します。

指定する内容は次の通りです。

- [1]...パターン番号 【1個の整数】
- [2]...パターンデータ 【複数のパラメータ】

パターンデータの開始は「 { 」で認識され、終了は「 } 」で検知されます。パターンデータの各パラメータは、カンマ区切りで記述します。スペースや改行は読み飛ばされます。(最後のパラメータの後にカンマがあっても可)

パターン番号 :

パターン番号は、PSコマンドの引数で使用する番号と合致するように定義します。例えばMMLで「PS7」のように記述した場合の「7」を、パターン番号に使用します。

パターン番号の設定範囲は 0 ~ 127 です。

パターンデータ :

パターンデータのフォーマットは、まず最初のパラメータで指定する種別番号 (TypeID) で変わります。

基本になるフォーマットは、次のように TypeID と ModeID が必ず先頭に配置された状態になります。

```
#MB:PITCHSCALE パターン番号 {
    TypeID,
    ModeID,
    [TypeIDとModeIDに従ったパラメータ群]...
}
```


・TypeID=0 のとき（1 2 平均律）

フォーマットは次の通りです。

```
#MB:PITCHSCALE パターン番号 {
    TypeID,
    ModeID,
    BASE_NOTE,
    BASE_FREQ,
}
```

ModeID :

現在指定可能なModeIDは 0 または 1 です。

0 の場合、デチューン解像度の少数以下が「有効」です。

（例えば @d3.352 と記述して少数以下が有効）

1 の場合、デチューン解像度の少数以下が「無効」です。

BASE_NOTE :

基準になる音程番号を整数で指定します。（音程番号とは）

設定可能範囲は、0 ~ 119 です。

BASE_FREQ :

基準になる音程に割り当てる周波数を指定します。小数以下の指定も受け付けません。設定結果、o4a に割り当たる周波数が 195Hz ~ 988Hz の範囲外の場合は、デフォルト設定（o4a = 440Hz）のままになります。

【例】

```
#MB:PITCHSCALE 1 {
    0,                //TypeID
    0,                //ModeID
    57,               //BASE_NOTE
    438.0,           //BASE_FREQ
}
```

上の例では、オクターブ4のAに438Hzを割り当て、それを基準にした1 2 平均律を適用しています。

・TypeID=10 のとき (F-Number)

フォーマットは次の通りです。

```
#MB:PITCHSCALE パターン番号 {
    TypeID,
    ModeID,
    MASTER_CLOCK,
    PRESCALE,
    START_KEY_OFS,
    KEY00,
    KEY01,
    KEY02,
    KEY03,
    KEY04,
    KEY05,
    KEY06,
    KEY07,
    KEY08,
    KEY09,
    KEY10,
    KEY11,
}
```

ModeID :

現在指定可能なModeIDは 0 のみです。

MASTER_CLOCK :

F-Number 計算のもとになるマスタークロックを指定します。小数以下の指定も受け付けます。例えば、3993600.0 などになります。

PRESCALE :

F-Number 計算のもとになるプリスケール値を指定します。小数以下の指定も受け付けます。

これはサンプリング周波数を得るためのプリスケール値になるため、OPNのプリスケール値そのものではなく、12倍済みの、例えば 72.0 などになります。

START_KEY_OFS :

KEY00～KEY11にて指定する F-Number値 は、Block (オクターブ) が0である前提です。その上で KEY00 に割り当てる音程番号を、-24 ～ 24 で指定します。0 の時、KEY00 は オクターブ0の C と見なされます。

1 増えるごとに半音上がり、1 下がるごとに半音下がります。通常 0 を利用します。

KEY00 : START_KEY_OFS + 0半音 のF-Number
KEY01 : START_KEY_OFS + 1半音 のF-Number
KEY02 : START_KEY_OFS + 2半音 のF-Number
KEY03 : START_KEY_OFS + 3半音 のF-Number
KEY04 : START_KEY_OFS + 4半音 のF-Number
KEY05 : START_KEY_OFS + 5半音 のF-Number
KEY06 : START_KEY_OFS + 6半音 のF-Number
KEY07 : START_KEY_OFS + 7半音 のF-Number
KEY08 : START_KEY_OFS + 8半音 のF-Number
KEY09 : START_KEY_OFS + 9半音 のF-Number
KEY10 : START_KEY_OFS + 10半音 のF-Number
KEY11 : START_KEY_OFS + 11半音 のF-Number

※KEY00～KEY11の各F-Numberは整数11ビットで指定します。ただし、0は使用できません。

※KEY00 から順に、数値は大きくなるように設定する必要があります。

※F-Numberの数値は10進数でも指定できますが、接頭辞「0x」をつけて16進数で記述することもできます。

【備考】周波数計算は次の要領で行われます。

$$fMUS = (F-Number * (2^{(b-1)}) * fSAM) / (2^{20})$$

fMUS : 周波数 (音程)

fSAM : 再生サンプリング周波数。

$$(fSAM = MASTER_CLOCK / PRESCALE)$$

b : オクターブ番号 (0～7)

【例】 F-Numberピッチスケール

```
#MB:PITCHSCALE 10 {  
  10,           //TypeID  
  0,           //ModeID  
  3993600.0,   //MASTER CLOCK  
  72.0,       //PRESCALE  
  0,          //START KEY OFFSET  
  0x026A,     //F-Number KEY00...KEY11  
  0x028F,  
  0x02B6,  
  0x02DF,  
  0x030B,  
  0x0339,  
  0x036A,  
  0x039E,  
  0x03D5,  
  0x0410,  
  0x044E,  
  0x048F  
}
```

・TypeID=11 のとき (PSG分周比)

フォーマットは次の通りです。

```
#MB:PITCHSCALE パターン番号 {
    TypeID,
    ModeID,
    MASTER_CLOCK,
    PRESCALE,
    DIVRT_MASK,
    DIVRT_OFFSET,
    START_KEYCODE,
    KEY00,
    KEY01,
    KEY02,
    KEY03,
    KEY04,
    KEY05,
    KEY06,
    KEY07,
    KEY08,
    KEY09,
    KEY10,
    KEY11,
}
```

ModeID :

現在指定可能なModeIDは 0 または 1 です。

0 の場合、オクターブ計算のシフト演算を、デチューン計算の後に行います。

1 の場合、オクターブ計算のシフト演算を、デチューン計算の前に行います。

MASTER_CLOCK :

PSG分周比計算のもとになるマスタークロックを指定します。小数以下の指定も受け付けます。例えば、3993600.0 などになります。

PRESCALE :

PSG分周比計算のもとになるプリスケール値を指定します。小数以下の指定も受け付けます。

これはマスタークロックへのプリスケール値になるため、PSGのプリスケールそのものではなく、8倍済みの、例えば 32.0 などになります。

DIVRT_MASK :

PSG分周比による周波数計算を行う場合に有効なビット幅を決定する、AND演算に使用するマスク値を整数で指定します。10進数でも指定できますが、接頭辞「0x」をつけて16進数で記述することもできます。

PSGでは通常、分周比は12ビットですので、0x0FFF のように指定します。
DCSGでは通常10ビットですので、0x03FF のように指定します。

DIVRT_OFFSET :

PSG分周比に対する加算値（オフセット）を整数で指定します。通常は0を指定します。MSX-SCC など、オフセットが必要な場合などに利用します。

START_KEYCODE :

KEY00～KEY11にて指定する PSG分周比 について、KEY00 の PSG分周比が、どの音程番号に相当するかを指定します。

設定範囲は 0 ～ 108 です。

0 の時、オクターブ0の C の分周比が定義されていると解釈します。1 増えるごとに半音上がります。通常は 0 を利用します。

KEY00 :	START_KEYCODE + 0半音のPSG分周比
KEY01 :	START_KEYCODE + 1半音のPSG分周比
KEY02 :	START_KEYCODE + 2半音のPSG分周比
KEY03 :	START_KEYCODE + 3半音のPSG分周比
KEY04 :	START_KEYCODE + 4半音のPSG分周比
KEY05 :	START_KEYCODE + 5半音のPSG分周比
KEY06 :	START_KEYCODE + 6半音のPSG分周比
KEY07 :	START_KEYCODE + 7半音のPSG分周比
KEY08 :	START_KEYCODE + 8半音のPSG分周比
KEY09 :	START_KEYCODE + 9半音のPSG分周比
KEY10 :	START_KEYCODE + 10半音のPSG分周比
KEY11 :	START_KEYCODE + 11半音のPSG分周比

※KEY00～KEY11の各PSG分周比は整数31ビット以下で指定します。ただし、0は使用できません。

※KEY00 から順に、数値は小さくなるように設定する必要があります。

※PSG分周比は10進数でも指定できますが、接頭辞「0x」をつけて16進数で記述することもできます。

【備考】周波数計算は次の要領で行われます。

$$f_{\text{TONE}} = f_m / (f_p * (T_p + \text{ofs}))$$

f_{TONE} : 周波数 (音程)

f_m : MASTER_CLOCK

f_p : PRESCALE

T_p : PSG分周比

ofs : DIVRT_OFFSET

($T_p + \text{ofs}$) の部分は、DIVRT_MASKでAND演算マスクがなされた結果の数値になります。

【例】PSG分周比ピッチスケール

```
#MB:PITCHSCALE 11 {
  11,           //TypeID
  0,           //ModeID
  3993600.0,    //MASTER CLOCK
  32.0,        //PRESCALE
  0x0FFF,      //div-ratio MASK
  0,           //div-ratio OFFSET
  0,           //START KEY CODE
  0x0EE8,      //div-ratio KEY00...KEY11
  0x0E12,
  0x0D48,
  0x0C89,
  0x0BD5,
  0x0B2B,
  0x0A8A,
  0x09F3,
  0x0964,
  0x08DD,
  0x085E,
  0x07E6
}
```

・TypeID=12 のとき (PSG分周比Full Keyモード)

フォーマットは次の通りです。

```
#MB:PITCHSCALE パターン番号 {
    TypeID,
    ModeID,
    MASTER_CLOCK,
    PRESCALE,
    DIVRT_MASK,
    DIVRT_OFFSET,
    K000,K001,K002,K003,K004,K005,K006,K007,K008,K009,K010,K011,
    K012,K013,K014,K015,K016,K017,K018,K019,K020,K021,K022,K023,
    K024,K025,K026,K027,K028,K029,K030,K031,K032,K033,K034,K035,
    K036,K037,K038,K039,K040,K041,K042,K043,K044,K045,K046,K047,
    K048,K049,K050,K051,K052,K053,K054,K055,K056,K057,K058,K059,
    K060,K061,K062,K063,K064,K065,K066,K067,K068,K069,K070,K071,
    K072,K073,K074,K075,K076,K077,K078,K079,K080,K081,K082,K083,
    K084,K085,K086,K087,K088,K089,K090,K091,K092,K093,K094,K095,
    K096,K097,K098,K099,K100,K101,K102,K103,K104,K105,K106,K107,
    K108,K109,K110,K111,K112,K113,K114,K115,K116,K117,K118,K119,
    K120,
}
```

ModeID :

現在指定可能なModeIDは 0 のみです。

MASTER_CLOCK :

PSG分周比計算のもとになるマスタークロックを指定します。小数以下の指定も受け付けます。例えば、3993600.0 などになります。

PRESCALE :

PSG分周比計算のもとになるプリスケール値を指定します。小数以下の指定も受け付けます。

これはマスタークロックへのプリスケール値になるため、PSGのプリスケラーそのものではなく、8倍済みの、例えば 32.0 などになります。

DIVRT_MASK :

PSG分周比による周波数計算を行う場合に有効なビット幅を決定する、AND演算に使用するマスク値を整数で指定します。10進数でも指定できますが、接頭辞「0x」をつけて16進数で記述することもできます。

PSGでは通常、分周比は12ビットですので、0x0FFF のように指定します。DCSGでは通常10ビットですので、0x03FF のように指定します。

DIVRT_OFFSET :

PSG分周比に対する加算値（オフセット）を整数で指定します。通常は0を指定します。

MSX-SCC など、オフセットが必要な場合などに利用します。

K000～K120 :

K000 : オクターブ0のCに割り当てるPSG分周比

K001 : オクターブ0のC+に割り当てるPSG分周比

K002 : オクターブ0のDに割り当てるPSG分周比

:

K057 : オクターブ4のAに割り当てるPSG分周比

:

K119 : オクターブ9のBに割り当てるPSG分周比

K120 : オクターブ10のCに割り当てるPSG分周比

※各PSG分周比は整数31ビット以下で指定します。ただし、0は使用できません。

※PSG分周比は10進数でも指定できますが、接頭辞「0x」をつけて16進数で記述することもできます。

【備考】周波数計算は次の要領で行われます。

$$f_{\text{TONE}} = f_m / (f_p * (T_p + \text{ofs}))$$

f_{TONE} : 周波数（音程）

f_m : MASTER_CLOCK

f_p : PRESCALE

T_p : PSG分周比

ofs : DIVRT_OFFSET

(T_p+ofs) の部分は、DIVRT_MASKでAND演算マスクがなされた結果の数値になります。

【例】PSG分周比Full Keyモードピッチスケール

```
#MB:PITCHSCALE 12 {  
  12,          //type  
  0,          //mode  
  1789772.5,  //clock  
  16.0,       //prescale  
  0x0FFF,     //div-rate-mask  
  0,          //offset  
  0x0FFE,0x0FFE,0x0FFE,0x0FFE,0x0FFE,0x0FFE,0x0FFE,0x0FFE,0x0FFE,0x0FD8,0x0EF7,0x0E20,  
  0x0D5D,0x0C9C,0x0BE7,0x0B3C,0x0A9B,0x0A02,0x0973,0x08EB,0x086B,0x07F2,0x0780,0x0714,  
  0x06AE,0x064E,0x05F4,0x059E,0x054D,0x0501,0x04B9,0x0475,0x0435,0x03F9,0x03C0,0x038A,  
  0x0357,0x0327,0x02FA,0x02CF,0x02A7,0x0281,0x025D,0x023B,0x021B,0x01FC,0x01E0,0x01C5,  
  0x01AC,0x0194,0x017D,0x0168,0x0153,0x0140,0x012E,0x011D,0x010D,0x00FE,0x00F0,0x00E2,  
  0x00D6,0x00CA,0x00BE,0x00B4,0x00AA,0x00A0,0x0097,0x008F,0x0087,0x007F,0x0078,0x0071,  
  0x006B,0x0065,0x005F,0x005A,0x0055,0x0050,0x004C,0x0047,0x0043,0x0040,0x003C,0x0039,  
  0x0035,0x0032,0x0030,0x002D,0x002A,0x0028,0x0026,0x0024,0x0022,0x0020,0x001E,0x001C,  
  0x001B,0x0019,0x0018,0x0016,0x0015,0x0014,0x0013,0x0012,0x0011,0x0010,0x000F,0x000E,  
  0x0002,0x0002,0x0002,0x0002,0x0002,0x0002,0x0002,0x0002,0x0002,0x0002,0x0002,  
  0x0002  
}
```

・TypeID=20 のとき (WSGステップレート)

フォーマットは次の通りです。

```
#MB:PITCHSCALE パターン番号 {
    TypeID,
    ModeID,
    MASTER_RATE,
    REGBIT_WIDTH,
    REGBIT_MASK,
    D_CULC_WIDTH,
    START_KEYCODE,
    KEY00,
    KEY01,
    KEY02,
    KEY03,
    KEY04,
    KEY05,
    KEY06,
    KEY07,
    KEY08,
    KEY09,
    KEY10,
    KEY11,
}
```

ModeID :

現在指定可能なModeIDは 0、1、2 の3択です。

0 の場合、ピッチLFO演算を、通常モードで行います。

1 の場合、ピッチLFO演算を、特殊モードで行います。

このモードのピッチLFOは、LFOの form にユーザー定義テーブルしか使用できません。加えて、テーブルで指定される内容が、相対デチューンの値として解釈されます。

2 の場合、1 の場合とほぼ同様ですが、ユーザー定義テーブルで定義する値のフォーマットが、符号あり8bit整数の4バイトパック方式になります。例えば、0xFF03と定義したサンプルは、相対デチューンが0、0、-1、+3のコマンドとして処理されます。

MASTER_RATE :

WSGステップレート周波数計算のもとになる再生周波数を指定します。小数以下の指定も受け付けます。例えば、24000.0 などになります。

REGBIT_WIDTH :

WSGステップレート周波数計算のもとになるビット幅を整数で指定します。例えば 20 などになります。

REGBIT_MASK :

WSGステップレート周波数計算を行う場合に有効なビット幅を決定する、AND演算に使用するマスク値を整数で指定します。10進数でも指定できますが、接頭辞「0x」をつけて16進数で記述することもできます。WSGでは通常 20ビットですので、0x0FFFFFF のように指定します。

D_CULC_WIDTH :

WSGステップレートにおけるデチューン計算は、音程ごとのステップレート値の $1/n$ 倍の値を演算に使用しますが、このn値を 2の何乗にするかを指定します。通常 8 を指定します。

START_KEYCODE :

KEY00～KEY11にて指定する WSGステップレート について、KEY00 の WSGステップレートが、どの音程番号に相当するかを、0 ～ 108 で指定します。96 の時、オクターブ8の C のステップレートが定義されていると解釈します。1 増えるごとに半音上がります。通常、96 を利用します。

KEY00 :	START_KEYCODE + 0半音のステップレート
KEY01 :	START_KEYCODE + 1半音のステップレート
KEY02 :	START_KEYCODE + 2半音のステップレート
KEY03 :	START_KEYCODE + 3半音のステップレート
KEY04 :	START_KEYCODE + 4半音のステップレート
KEY05 :	START_KEYCODE + 5半音のステップレート
KEY06 :	START_KEYCODE + 6半音のステップレート
KEY07 :	START_KEYCODE + 7半音のステップレート
KEY08 :	START_KEYCODE + 8半音のステップレート
KEY09 :	START_KEYCODE + 9半音のステップレート
KEY10 :	START_KEYCODE + 10半音のステップレート
KEY11 :	START_KEYCODE + 11半音のステップレート

※各WSGステップレートは整数31ビット以下で指定します。ただし、0は使用できません。

※KEY00 から順に、数値は大きくなるように設定する必要があります。

※WSGステップレートは10進数でも指定できますが、接頭辞「0x」をつけて16進数で記述することもできます。

【備考】周波数計算は次の要領で行われます。

$$\text{Freq} = \text{mRATE} * \text{wsgVAL} / (2^{\text{regWIDTH}})$$

Freq：周波数（音程）

mRATE：MASTER_RATE

wsgVAL：WSGステップレート

regWIDTH：REGBIT_WIDTH

wsgVAL の部分は、REGBIT_MASK で AND演算マスクがなされた結果の数値になります。

【例】WSGステップレートピッチスケール

```
#MB:PITCHSCALE 20 {
    20,          //type
    1,          //mode
    24000.0,    //master rate
    20,        //register bit width
    0x0FFFFFF, //register bit mask
    8,        //detune-culc bit width
    96,      //start key-code
    0x02C578,
    0x02EFCB,
    0x031C82,
    0x034BC8,
    0x037DF6,
    0x03B335,
    0x03EB87,
    0x042717,
    0x046669,
    0x04A950,
    0x04F050,
    0x053B68
}
```

2.16. #ML:REVERSE-OCTAVE : 相対オクターブ記号反転

記述	#ML:REVERSE-OCTAVE
----	--------------------

種別	メタデータ指定
----	---------

【解説】

相対オクターブ変更記号の上下反転を行います。

メタデータとして

#ML:REVERSE-OCTAVE

と記述すると、相対オクターブの上げ下げ（不等号）の機能が反転します。

【例】

```
#ML:REVERSE-OCTAVE
```

3. 音長関連

3.1. L : デフォルト音長

記述	L[1]
種別	MML 擬似コマンド

【解説】

デフォルト音長（音名の後の音長指定を省略した場合の音長）を設定します。トラック先頭におけるデフォルトは、L4（4分音符）です。

引数[1]には、通常1以上の整数(n)を指定します。この場合、デフォルト音長は、(n)分音符になります。例えば L16 と記述すると、デフォルト音長が16分音符になります。

特例として、「%」付きの数値(%n)で記述することも出来ます。この場合、デフォルト音長は、(%n)のtickカウント数になります。例えば L%96 と記述すると、デフォルト音長は96(ticks)になります。

(n)分音符に指定できる数値は、「#ML:TICKUNIT」の設定により変わります。例えば「#ML:TICKUNIT 384」の場合、tickカウントが1あたり384分音符になるので、使えるデフォルト音長のラインナップは、384が整数で割り切れるものになります。列挙すると、
384, 192, 128, 96, 64, 48, 32, 24, 16, 12, 8, 6, 4, 3, 2, 1
となります。

【例】

```
L8 CDEFG
```

CDEFGが全て8分音符で演奏されます。

3.2. Q : ゲートタイム TYPE-1

記述 Q[1],[2]

種別 MML 擬似コマンド

【解説】

ゲートタイムの割合を設定します。

ある長さの音符を演奏するとき、実際に指定された音長のうち、何分の何を発音し、残りを休符とするか、を決定します。

スタッカートなどの表現に利用します。

指定する引数は次の通りです。

引数[1]...発音長倍率の分子 (num)

引数[2]...発音長倍率の分母 (denom)

引数はカンマで区切って指定します。カンマとdenomの組は省略できます。

トラック先頭におけるデフォルトは、Q16,16 です。

引数[1] (num)

発音長倍率の分子を、符号無しの整数で指定します。

引数[2] (denom)

発音長倍率の分母を、符号無しの整数で指定します。

denom のデフォルトは 16 です。

denom は省略可能で、省略した場合は以前に設定された値が採用されます。

ゲートタイムの設定範囲は、

分子÷分母が 0より大きく、1以下となる範囲です。

範囲外の場合は、設定そのものが無視されます。

トラック先頭で1度 denom 付きで設定したら、以降の設定は denom を省略する使い方を想定しています。

分母が 16 で構わない場合は、トラック先頭から denom を省略してください。

【例】

```
L4 @Q0 Q%0 Q32,32 CDEFG
Q27 GFEDC
```

この場合、CDEFG はゲートが掛からない「32/32 の発音」となり、GFEDC は「27/32 の発音 + 5/32 の休符」になります。

【注】

本機能と Q%（ゲートタイムTYPE-2）は排他制御が掛かります。
Q% の指定値が 0 以外の場合、Q% が優先され、本機能は無効になります。
ただし、本機能の指定値は保存されているので、Q% に 0 を指定して無効化すると、保存されていた指定値で本機能が有効になります。
本機能と @Q（ゲートタイムTYPE-3）は排他制御が掛からず、同時に作用します。

3.3. Q% : ゲートタイム TYPE-2

記述	Q@[1]
種別	MML 擬似コマンド

【解説】

ゲートタイムを、ノートオンからの tick カウント数 で設定します。
ある長さの音符を演奏するとき、実際に指定された音長のうち、何 tick カウントを発音して残りを休符とするか、を決定します。
スタッカートなどの表現に利用します。
tick カウントあたりの発音長は、「#ML:TICKUNIT」による設定の影響を受けます。

引数[1]には、発音する側の tick カウント数を 0 以上の整数で指定します。
0 は本機能を無効にする特別な指定値です。
トラック先頭におけるデフォルトは、Q%0 です。

【注】

本機能と Q (ゲートタイムTYPE-1) は排他制御が掛かります。
Q を使いたい場合、本機能を Q%0 としてください。
本機能と @Q (ゲートタイムTYPE-3) は排他制御が掛からず、同時に作用します。

【例】

```
#ML:TICKUNIT 192  
@Q0 Q%40 C4 G2;
```

この場合、
C は 40ticks 発音 + 8ticks 休符、
G は 40ticks 発音 + 56ticks 休符、
として演奏されます。

3.4. @Q : ゲートタイム TYPE-3

記述	@Q[1]
種別	MML 擬似コマンド

【解説】

ゲートタイムを、ノートオフを早める tickカウント数 で指定します。

Q または Q% で指定されたゲートタイムから、さらに、指定数値の tickカウント数 だけ引いた発音長にします。tickカウントあたりの発音長は、「#ML:TICKUNIT」による設定の影響を受けます。

引数[1]には、ノートオフを早める tickカウント数を、0 以上の整数で指定します。トラック先頭におけるデフォルトは、@Q0 です。

【注】

Q または Q% と、@Q により、計算上の発音長が1 tickカウント未満になってしまう場合は、実際の発音長は1 tickカウントに矯正されます。

【例】

```
#ML:TICKUNIT 192
Q%0 Q16,16 @Q5 C4 G2;
```

この場合、
C は 43ticks発音+5ticks休符、
G は 91ticks発音+5ticks休符、
として演奏されます。

3.5. #ML:TICKUNIT : TICKカウント単位設定

記述 #ML:TICKUNIT [1]

種別 メタデータ指定

【解説】

tickカウントの単位を設定します。

引数[1]には、tickカウント数1あたりを、何分音符とみなすかを指定します。デフォルト設定は 192 です（1 tick = 192分音符）。

設定条件は次の通りです。

- ・ 48以上、4608以下（48×96以下）の整数であること。
- ・ 4で割り切れる（4分音符が作れる）こと。

4分音符の生成が必須条件になっているのは、tempoコマンドにおけるBPM計算が「4分音符を1ビート」と定義しているためです。

【備考1】

4分音符などの音長は、tickカウント数への換算が自然数になるように作られています。これは、テンポずれ起こさないようにするためです。

「tickカウント換算が自然数」についてもう少し考えてみます。

例えば、192分音符単位での tickカウント換算は、このようになります。

音符	192	96	64	48	32	24	16	12	8	6	4	3	2	1
tick	1	2	3	4	6	8	12	16	24	32	48	64	96	192

これらは tickカウント換算が自然数になる物のうち、付点音符を含まないものです。付点音符においても、tickカウント換算が自然数になるものを使います。

【備考2】

連符の考え方について。例えば4分音符の3連符の場合。

まず4分音符は、全音符の1/4です。

さらに、4分音符の3分割は、

$$(1/4) \times (1/3) = (1/12)$$

となるので12分音符が作れば良い、と考えていきます。

例えば、4分音符の長さの9連符が欲しい場合、36分音符が必要になります。しかし36分音符はデフォルトの192分音符単位では音長ラインナップに無いので作れません。

そこで、144分音符単位

#ML:TICKUNIT 144

を考えてみます。

144分音符単位の音長ラインナップは、
144, 72, 48, 36, 24, 18, 16, 12, 9, 8, 6, 4, 3, 2, 1
ですので、36分音符が使えることになります。

このように、特別な音長単位を使いたい場合、曲目で使用する音長ラインナップを全て含む「#ML:TICKUNIT」を設定します。

4. 音量関連

4.1. 音量設定の前提事項

【解説】

V2MMLでは、音声バッファを $-1.0 \sim +1.0$ の浮動小数点数（ステレオ）で用意しています。

音声バッファにレンダリングする前の、音源モジュール段階の波形も、 $-1.0 \sim +1.0$ の浮動小数点数（モノラル）で用意しています。

このため、レベル調整しないと複数トラックであつという間に音割れが発生してしまいます。

レベル調整は大まかに分けて3つの段階があります。

音源モジュールの出力を音声バッファにレンダリングする際は、次の順でレベル調整を行います。

- (1) ミキシングボリューム演算（基準の 0dB 指定で振幅は 0.5 倍に設定）
- (2) ボリューム、エンベロープ演算（最大で振幅が 1.0 倍に設定）
- (3) ボリュームL、音量LFO演算（最大で振幅が 1.0 倍に設定）
- (4) パンポット演算 (@P および P)

となっています。

@P のパンポット演算では注意が必要です。倍率計算を抜粋して挙げると、

@P設定	左側の振幅倍率	右側の振幅倍率
@P-100	1.0 倍	0.0 倍
@P-50	0.75 倍	0.25 倍
@P0	0.5 倍	0.5 倍
@P50	0.25 倍	0.75 倍
@P100	0.0 倍	1.0 倍

となります。@P では、**左右の最大振幅が、中央設定の2倍**になっています。

一方 P のレガシーパンポットは、中央設定より大きくはなりません。

P設定	左側の振幅倍率	右側の振幅倍率
P-1	1.0 倍	0.0 倍
P0	1.0 倍	1.0 倍
P1	0.0 倍	1.0 倍

したがって、

- ・ミキシングボリュームが 0dB
- ・ボリューム、エンベープ、ボリュームL、音量LFOが最大
- ・パンポットが中心

という条件では、左右双方、音源モジュール出力の振幅の 0.25 倍になります。

この条件のとき、同じ波形、同じ音程、同じ位相でトラックを重ねると、

4トラックまでは音割れせず、

5トラック以上では音割れする

ことになります。

以上を踏まえて、最終的に、全トラックの音声を加算した結果が、

-1.0 ~ +1.0 の振幅に収まるように、

- ・ミキシングボリューム
- ・ボリューム、エンベープ、ボリュームL、音量LFO
- ・パンポット

における倍率を設定していくことになります。

意図せず音割れを起こしてしまった場合に備えて、KeyDispウィンドウには

ClipCount(L)

ClipCount(R)

の数値表示を設けてあります。

このカウンタは、演奏開始から数える、音割れでクリッピングしたサンプル数です。この数値が1以上になったら音割れが発生していますので、ミキシングボリュームを下げたり等の調整を検討してください。

4.2. デシベル計算

【解説】

V 2 MML で扱うデシベル (dB) は、相対量としてのデシベルで、基準量に対する倍率を表すものです。

通常、デシベル計算による音量の振幅倍率は次の式で得られます。

$$\text{倍率} = 10^{(\text{dB}/20)}$$

しかし、V 2 MML では特例で、次の式を採用しています。

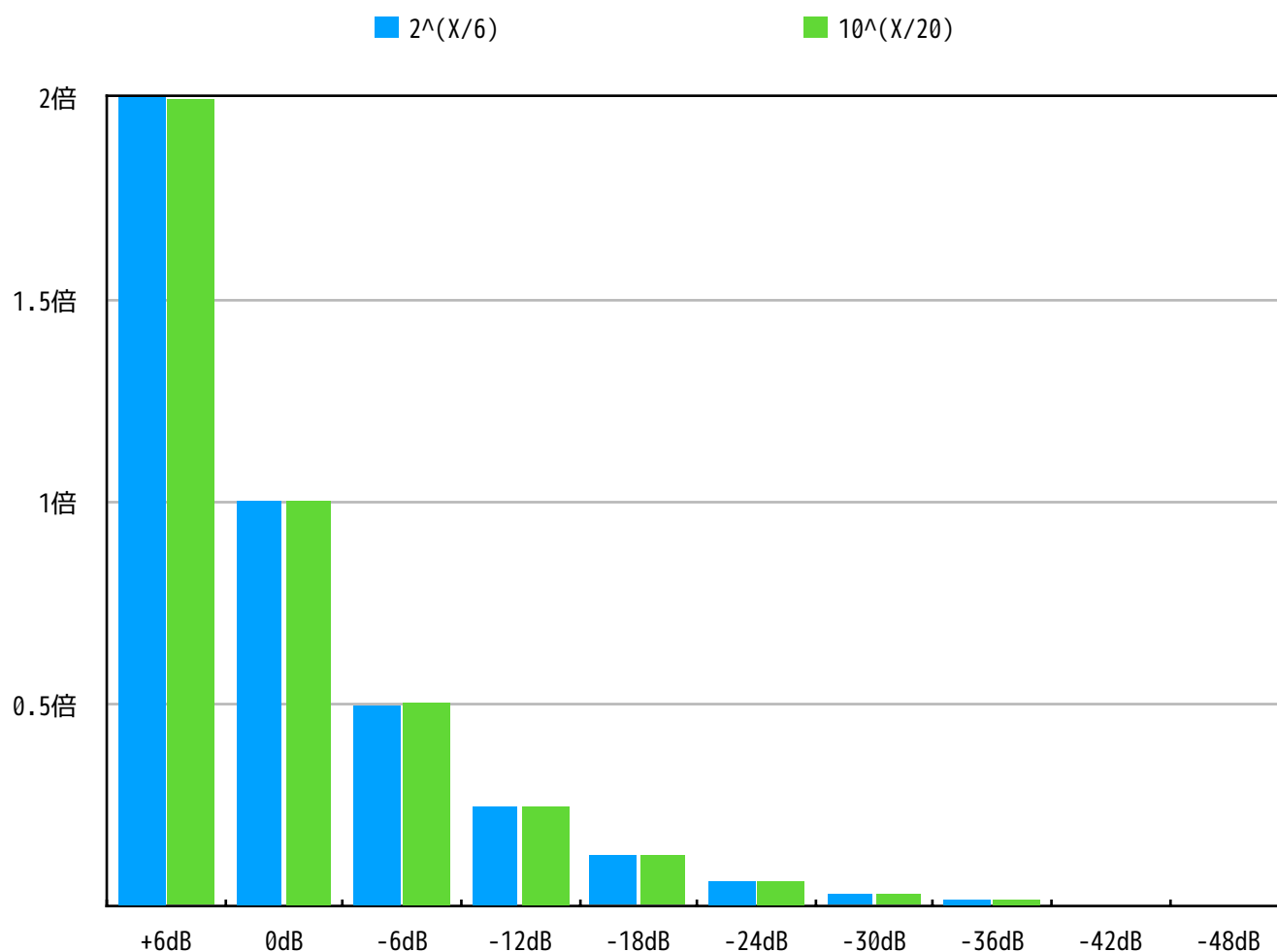
$$\text{倍率} = 2^{(\text{dB}/6)}$$

この計算は通常のデシベル計算とよく似た結果になりますが、相違点は、

- ・ 通常dBは、20増えるごとに丁度10倍、20減るごとに丁度1/10倍。
- ・ 特例dBは、6増えるごとに丁度 2倍、6減るごとに丁度 1/2倍。

となります。

これをグラフと表にしたものは次の通りです。



dB	倍率 : $2^{(dB/6)}$	倍率 : $10^{(dB/20)}$
24	16	15.8489319246111
18	8	7.94328234724282
12	4	3.98107170553497
6	2	1.99526231496888
0	1	1
-6	0.5	0.501187233627272
-12	0.25	0.251188643150958
-18	0.125	0.125892541179417
-24	0.0625	0.0630957344480193
-30	0.03125	0.0316227766016838
-36	0.015625	0.0158489319246111
-42	0.0078125	0.00794328234724281
-48	0.00390625	0.00398107170553497
-54	0.001953125	0.00199526231496888
-60	0.0009765625	0.001
-66	0.00048828125	0.000501187233627273
-72	0.000244140625	0.000251188643150958
-78	0.0001220703125	0.000125892541179417
-84	0.00006103515625	0.0000630957344480193
-90	0.000030517578125	0.0000316227766016838
-96	0.0000152587890625	0.0000158489319246111

4.3. MV：ミキシングボリューム

記述	MV[1]
種別	MML コマンド

【解説】

ミキシングボリュームを設定します。

引数[1]には、ミキシングボリュームを数値で指定します。
この引数では加減算記述も可能です。
トラック先頭におけるデフォルトは、MV0 (0dB) です。

設定範囲は -100 ~ 48 で、小数以下の指定も受け付けます。
単位はデシベル (dB) です。 (デシベルとは)

-100 以下の場合、特例で無音とみなし、振幅を 0 倍にします。
正の値を指定した場合は利得を持つこととなります。

【備考1】

音量設定の前提事項

【備考2】

MV0の指定で等倍の扱いになります。
ただし、この等倍には基準倍率として「0.5倍」が掛かっています。
つまり、等倍のとき、音源モジュールの波形出力に対して、振幅が「0.5倍」となりますが、これがミキシングボリュームの基準の 0dB になっています。

【備考3】

音量計算は統一的に 64bit 浮動小数計算で行っています。そのため、各種音量計算による音質ロスの心配は、ほぼありません。

4.4. @P：パンポット

記述 @P[1]

種別 MML コマンド

【解説】

パンポット（音声出力の左右バランス）を設定します。

引数[1]には、左右バランスを表す数値を指定します。
設定範囲は -100 ~ 100 で、小数以下の指定も受け付けます。
トラック先頭におけるデフォルトは、@P0（中心）です。

設定に対応する左右のバランスは次の通りです。

@P設定	左側の振幅倍率	右側の振幅倍率
@P-100（左側に最大）	1.0 倍	0.0 倍
@P-50（左側に50%）	0.75 倍	0.25 倍
@P0（中央設定）	0.5 倍	0.5 倍
@P50（右側に50%）	0.25 倍	0.75 倍
@P100（右側に最大）	0.0 倍	1.0 倍

（@Pでは、**左右の最大振幅が、中央設定の2倍**になっている点に注意）

【備考】

音量設定の前提事項

4.5. P：レガシーパンポット

記述	P[1]
種別	MML コマンド

【解説】

レガシーパンポット（左右の音声出力のスイッチング）を設定します。

引数[1]には、左右のスイッチングを表す整数を指定します。

設定範囲は、-1、0、1 の三択です。

トラック先頭におけるデフォルトは、P0（左右両方出力）です。

設定に対応する左右のスイッチングは次の通りです。

P設定	左側の振幅倍率	右側の振幅倍率
P-1（左側のみ出力）	1.0 倍	0.0 倍
P0（左右両方出力）	1.0 倍	1.0 倍
P1（右側のみ出力）	0.0 倍	1.0 倍

【備考】

レガシーパンポットは、機械的に左右のオンオフを行うだけです。

@P のパンポットを中心に設定した状態で使用することを想定しています。

※@Pコマンドでは片方に完全に寄せた場合、中心と比較して振幅が倍になりますが、Pコマンドでは左右のオンオフを行うだけなので、左右に振っても中心の振幅を超えることはありません。

4.6. VS : ボリュームスケール

記述	VS[1]
種別	MML コマンド

【解説】

ボリュームスケールを設定します。

ボリュームスケールでは、Vと@EA および VLと@LA から求める音量振幅を算出する方式を選択します。

指定する引数は次の通りです。

引数[1]...ボリュームスケール種別 (vsType)

引数[2]...ボリュームスケール番号 (vsNo)

引数はカンマで区切って指定します。カンマとvsNoの組は省略できます。

トラック先頭におけるデフォルトは、VS2 です。

引数[1] (vsType)

vsTypeは、0、1、2 のいずれかで指定します。

vsType=0 の場合

テーブルモードで音量を計算します。

vsType=1 の場合

線形モードで音程を計算します。

vsType=2 の場合

dBモードで音程を計算します。

当コマンドを指定しなかった場合のデフォルト設定は、このモードになります。

引数[2] (vsNo)

vsNoの指定は省略することができます。

省略した場合は、システム側で持っている、vsTypeに応じたデフォルトパターンに設定されます。

vsNoを指定する場合は、あらかじめ「#MB:VOLUMESCALE」でボリュームスケールパターンをユーザー定義し、その定義番号をvsNoに指定します。

4.7. V : ボリューム (for @EA)

記述	V[1]
種別	MML コマンド

【解説】

ボリューム（音量エンベロープ用の振幅倍率）を設定します。

引数[1]にはボリューム値を数値で指定します。
数値の範囲や、数値に対応する振幅倍率は、VSコマンドの設定内容に従います。

トラック先頭におけるデフォルトは、V13 です。
(デシベルモード、V範囲0~15、減衰3dB単位、V0は無音、の設定)

Vコマンドの振幅倍率には、音量エンベロープ(@EA)の時間変化が掛かります。

線形音量、dB音量では、指定数値が大きくなるほど音量が大きくなります。
また、VSコマンドにて、Vコマンドの小数以下を丸めない設定では、Vコマンドの小数以下の指定も有効になります。

テーブル定義音量では、テーブル設定内容に従った音量変化になります。
この場合、Vコマンドの指定値はテーブルの何番目を参照するかの数値になりますので、Vコマンドの小数以下の指定は必ず丸められます。小数以下の丸め方は、VSコマンドの設定に従います。

【例】

```
V15 CDE V12 FGA
```

この場合、ボリューム15で CDE を演奏し、
ボリューム12で FGA を演奏します。

4.8.) または (: 相対ボリューム (for @EA)

記述) または (

種別 MML 擬似コマンド

【解説】

当コマンド指定時の V コマンド (ボリューム) の設定状態から、相対的に数値変更したボリュームに設定します。

) ボリューム値を 1 上げます。(V10 だった場合 V11 にする)

(ボリューム値を 1 下げます。(V10 だった場合 V9 にする)

2 以上変更したい場合は、複数個記述するか、数字を添えて記述します。

)5 ボリューム値を 5 上げます。

(5 ボリューム値を 5 下げます。

数字を添える記述では、小数以下の指定も受け付けますが、演算を行った結果のボリューム設定 (最大・最小制限、小数以下の処理) は、VS コマンドの設定による制限に従います。

小カッコの向きと上下の対応を反転したい場合は、「#ML:REVERSE-VOLUME」によって反転できます。

【例】

```
L8 V10 C)D)E)F)G ((A (3 B
```

この場合、各音符の音量設定は次のようになります。

C:V10

D:V11

E:V12

F:V13

G:V14

A:V12

B:V9

4.9. VL : ボリューム L (for @LA)

記述	VL[1]
種別	MML コマンド

【解説】

ボリューム L (音量 L F O用の振幅倍率) を設定します。

引数[1]にはボリューム L 値を数値で指定します。
数値の範囲や、数値に対応する振幅倍率は、VSコマンドの設定内容に従います。

トラック先頭におけるデフォルトは、VL15 です。
(デシベルモード、VL範囲0~15、減衰3dB単位、VL0は無音、の設定)

VLコマンドの振幅倍率には、音量 L F O (@LA)の時間変化が掛かります。

【注】音量 L F O (@LA)が掛かっている間だけ、VLコマンドが有効です。
(@LAのdelay時間の最中は、@LAが無効のためVLも無効です)

線形音量、dB音量では、指定数値が大きくなるほど音量が大きくなります。
また、VSコマンドにて、VLコマンドの小数以下を丸めない設定では、VLコマンドの小数以下の指定も有効になります。

テーブル定義音量では、テーブル設定内容に従った音量変化になります。
この場合、VLコマンドの指定値はテーブルの何番目を参照するかの数値になりますので、VLコマンドの小数以下の指定は必ず丸められます。小数以下の丸め方は、VSコマンドの設定に従います。

【例】

```
#ML:LF0@A 0, 4,24, 0, 0, 0  
l2 v15 @la0 vl15 c vl13 c vl11 c vl9 c vl7 c;
```

この場合、ボリューム L が15から 2 ずつ下がって音符が演奏されます。
@la0の音量 L F Oにはdelayが掛かっていないため VLコマンドがノートオン直後から効いています。

4.10. VF：ボリュームファンクション

記述 VF[1],[2]

種別 MML コマンド

【解説】

音量に関する特殊機能の設定を行います。

指定する引数は次の通りです。

引数[1]...特殊機能の種別番号 (funcType)

引数[2]...設定値 (param)

引数はカンマで区切って指定します。

トラック先頭におけるデフォルトは、どのfuncTypeも無効に設定されています。

引数[1] (funcType)

現状サポートされているfuncTypeは、次の通りです。

- ・ 20 (エンベロープがリリース中のみ、Vを強制的に最大として処理)
- ・ 21 (エンベロープがリリース中のみ、VLを強制的に最大として処理)

引数[2] (param)

funcTypeによって指定内容や効果が変わります。

funcType = 20の場合のparam

エンベロープがリリース中のみ、V コマンドの音量を最大値として扱う特殊機能の有効無効を切り替えます。

VF20,0 機能の無効 (デフォルト)

VF20,1 機能の有効

funcType = 21の場合のparam

エンベロープがリリース中のみ、VL コマンドの音量を最大値として扱う特殊機能の有効無効を切り替えます。

VF21,0 機能の無効 (デフォルト)

VF21,1 機能の有効

4.11. @F0：フェードアウト

記述	@F0[1],[2]
種別	MML コマンド

【解説】

フェードアウト機能です。

当コマンドを指定したトラックのみ、音量を徐々に減衰させます。

指定する引数は次の通りです。

引数[1]...フェードアウト時間 (time)

引数[2]...音量変化させる範囲 (range)

引数はカンマで区切って指定します。

引数[1] (time)

フェードアウトを行う時間を 0 以上の数値で指定します。

単位は秒です。小数以下の指定も受け付けます。

0 を指定した場合は特例で、フェードアウト処理中であっても、フェードアウト機能を中断して、通常音量(0dB)に戻ります。この場合、range 指定は無効です。

引数[2] (range)

フェードアウトによって減衰する音量範囲を指定します。

範囲は -110 ~ 0 の整数で、単位はデシベルです。

減衰量の指定なのでマイナス数値であることに注意してください。

0 未満の場合、指定デシベルまで指数変化で減衰します。

0 の場合は特例で、無音まで線形変化で減衰します。

0 より大きい指定は、0 を指定したものとみなされます。

range は省略可能で、省略した場合は 0 を指定したものとみなされます。

【備考1】

最後に指定したフェードコマンドが優先されます。

例えば、長いフェードアウトの途中で、フェードインに切り替えて動作させることもできます。

【備考2】

フェードアウトのゼロ時間指定と、フェードインのゼロ時間指定 (@F00 と @FI0) の内部処理は同一です。

【例】

@F03, -48 CDEFG

@F03, -48記述以降の演奏に対し、3 秒間かけて、デシベルスケールで 0dB から -48dB まで減衰させます。

4.12. @FI : フェードイン

記述	@FI[1],[2]
種別	MML コマンド

【解説】

フェードイン機能です。

当コマンドを指定したトラックのみ、音量を徐々に増加させます。

- ・ 増加する音量に利得は含みません。
- ・ 最低音量から通常音量までの増加です。

指定する引数は次の通りです。

引数[1]...フェードイン時間 (time)

引数[2]...フェードイン開始時点の最低音量 (range)

引数はカンマで区切って指定します。

引数[1] (time)

フェードインを行う時間を 0 以上の数値で指定します。

単位は秒です。小数以下の指定も受け付けます。

0 を指定した場合は特例で、フェードイン処理中であっても、フェードイン機能を中断して、通常音量(0dB)に戻ります。この場合、range 指定は無効です。

引数[2] (range)

フェードイン開始時点の最低音量を指定します。

指定範囲は -110 ~ 0 の整数で、単位はデシベルです。

最低音量の指定なので、マイナス数値であることに注意してください。

0 未満の場合、指定デシベルから通常音量まで指数変化で増加します。

0 の場合は特例で、無音から通常音量まで線形変化で増加します。

0 より大きい指定は、0 を指定したものとみなされます。

range は省略可能で、省略した場合は 0 を指定したものとみなされます。

【備考1】

最後に指定したフェード機能が優先されます。

例えば、長いフェードアウトの途中で、フェードインに切り替えて動作させることもできます。

【備考2】

フェードアウトのゼロ時間指定と、フェードインのゼロ時間指定 (@F00 と @FI0) の内部処理は同一です。

【例】

@FI3, -48 CDEFG

@FI3, -48記述以降の演奏に対し、3 秒間かけて、デシベルスケールで -48dB から 0dB まで増加させます。

4.13. #MB:VOLUMESCALE : ボリュームスケール定義

記述	#MB:VOLUMESCALE [1] {[2]}
種別	メタデータ定義

【解説】

ボリュームスケールコマンド (VS) で使用する、ユーザー定義のボリュームスケールパターンを定義します。

指定する内容は次の通りです。

- [1]...パターン番号 【1個の整数】
- [2]...パターンデータ 【複数のパラメータ】

パターンデータの開始は「 { 」で認識され、終了は「 } 」で検知されます。パターンデータの各パラメータは、カンマ区切りで記述します。スペースや改行は読み飛ばされます。(最後のパラメータの後にカンマがあっても可)

パターン番号 :

パターン番号は、VSコマンドの引数で使用する番号と合致するように定義します。例えばMMLで「VS7」のように記述した場合の「7」を、パターン番号に使用します。

パターン番号の設定範囲は 0 ~ 127 です。

パターンデータ :

パターンデータのフォーマットは、まず最初のパラメータで指定する種別番号 (TypeID) で変わります。

基本になるフォーマットは、次のように TypeID と ModeID が必ず先頭に配置された状態になります。

```
#MB:VOLUMESCALE パターン番号 {
    TypeID,
    ModeID,
    [TypeIDとModeIDに従ったパラメータ群]...
}
```

・TypeID=0 のとき（配列定義モード）

フォーマットは次の通りです。

```
#MB:VOLUMESCALE パターン番号 {
    TypeID,
    ModeID,
    V_DENOM,
    V_ROUND,
    VL_DENOM,
    VL_ROUND,
    VH_DENOM,
    V0_MAGNI,           //V0からVMAXまでの
    V1_MAGNI,           //音量倍率の配列個数は
    V2_MAGNI,           //VH_DENOMで指定する
    :                   //数値+1の個数を
    VMAX_MAGNI,         //定義します。
}
```

ModeID :

現在指定可能なModeIDは 0 のみです。

V_DENOM :

Vコマンドで指定する最大値を指定します。

設定範囲は 3 ~ 10000 の整数です。

V_ROUND :

Vコマンドの指定値と、@EAエンベロープ出力 (0.0~1.0) を掛けた結果(*A) を、整数に丸めるモードを指定します。

0または1のとき：

(*A)の小数以下を四捨五入します。

2のとき：

(*A)に1を足し小数以下を切り捨てます。

切り捨て結果が V を超える場合は V に制限します。

3のとき：

(*A)の小数以下を切り捨てます。

10のとき（平行移動）：

(*A)から、V 最大値と V 指定値の差を引きます。

VL_DENOM :

VLコマンドで指定する最大値を指定します。

設定範囲は 3 ~ 10000 の整数です。

VL_ROUND :

VLコマンドの指定値と、@LA音量 L F O出力 (0.0~1.0) を掛けた結果(* B) を、整数に丸めるモードを指定します。

0または1のとき：

(* B) の小数以下を四捨五入します。

2のとき：

(* B) に 1 を足し小数以下を切り捨てます。

切り捨て結果が VL を超える場合は VL に制限します。

3のとき：

(* B) の小数以下を切り捨てます。

10のとき (平行移動) ：

(* B) から、VL 最大値と VL 指定値の差を引きます。

VH_DENOM :

仮想音量ハード (この場合配列定義された音量倍率) の音量最大値を指定します。

設定範囲は、3 ~ 1023 の整数です。

Vコマンドの指定値と、@EAエンベロープ出力と、V_ROUND設定によって得られた音量指定値(* C)と、

VLコマンドの指定値と、@LA音量 L F O出力と、VL_ROUND設定によって得られた音量指定値(* D)は、

共通の仮想音量ハードを想定して音量倍率を適用します。

まず(* C)をインデックスに使用し、音量倍率テーブルを引いて得た値を振幅に乗算します。

@LAコマンドが有効の場合は、さらに(* D)をインデックスに使用し、音量倍率テーブルを引いて得た値を振幅に乗算します。

V0_MAGNI~VMAX_MAGNI :

仮想音量ハードの、インデックス 0~最大値 で使用する倍率値を指定します。

V0_MAGNIからVMAX_MAGNIまでの定義個数は、

VH_DENOM値+1(個)

になります。

指定可能な倍率値の範囲は、0~1の浮動小数点数です。

特別な表記方法として、

- ・分数による指定 (例 : 1.0/3.0)

- ・デシベルによる指定 (例 : -3.0dB) ※末尾に「dB」を付加するが可能です。

・TypeID=1 のとき（線形モード）

フォーマットは次の通りです。

```
#MB:VOLUMESCALE パターン番号 {
    TypeID,
    ModeID,
    V_DENOM,
    V_ROUND,
    VL_DENOM,
    VL_ROUND,
    VH_DENOM,
}
```

ModeID :

現在指定可能なModeIDは 0 のみです。

V_DENOM :

Vコマンドで指定する最大値を指定します。
設定範囲は 3 ~ 10000 の整数です。

V_ROUND :

Vコマンドの指定値と、@EAエンベロープ出力 (0.0~1.0) を掛けた結果(*E) を、整数に丸めるモードを指定します。

0 のとき :

(*E)の丸め処理を行いません。
小数以下の数値が有効になります。

1 のとき :

(*E)の小数以下を四捨五入します。

2 のとき :

(*E)に1を足し小数以下を切り捨てます。
切り捨て結果が V を超える場合は V に制限します。

3 のとき :

(*E)の小数以下を切り捨てます。

10のとき（平行移動） :

(*E)から、V 最大値と V 指定値の差を引きます。

VL_DENOM :

VLコマンドで指定する最大値を指定します。
設定範囲は 3 ~ 10000 の整数です。

VL_ROUND :

VLコマンドの指定値と、@LA音量 L F O出力 (0.0~1.0) を掛けた結果(*F) を、整数に丸めるモードを指定します。

・TypeID=2 のとき (デシベルモード)

フォーマットは次の通りです。

```
#MB:VOLUMESCALE パターン番号 {
    TypeID,
    ModeID,
    V_DENOM,
    V_ROUND,
    VL_DENOM,
    VL_ROUND,
    VH_DENOM,
    VH_RATE,
    VH_V0MODE,
}
```

ModeID :

現在指定可能なModeIDは 0 のみです。

V_DENOM :

Vコマンドで指定する最大値を指定します。
設定範囲は 3 ~ 10000 の整数です。

V_ROUND :

Vコマンドの指定値と、@EAエンベロープ出力 (0.0~1.0) を掛けた結果(* I) を、整数に丸めるモードを指定します。

0 のとき :

(* I) の丸め処理を行いません。

小数以下の数値が有効になります。

1 のとき :

(* I) の小数以下を四捨五入します。

2 のとき :

(* I) に 1 を足し小数以下を切り捨てます。

切り捨て結果が V を超える場合は V に制限します。

3 のとき :

(* I) の小数以下を切り捨てます。

10 のとき (平行移動) :

(* I) から、V 最大値と V 指定値の差を引きます。

VL_DENOM :

VLコマンドで指定する最大値を指定します。
設定範囲は 3 ~ 10000 の整数です。

VL_ROUND :

VLコマンドの指定値と、@LA音量 L F O出力 (0.0~1.0) を掛けた結果(* J) を、整数に丸めるモードを指定します。

0 のとき：

(* J)の丸め処理を行いません。

小数以下の数値が有効になります。

1 のとき：

(* J)の小数以下を四捨五入します。

2 のとき：

(* J)に1を足し小数以下を切り捨てます。

切り捨て結果が VL を超える場合は VL に制限します。

3 のとき：

(* J)の小数以下を切り捨てます。

10のとき (平行移動)：

(* J)から、VL 最大値と VL 指定値の差を引きます。

VH_DENOM：

仮想音量ハード (この場合デシベルパターンの音量倍率) の音量最大値を指定します。

設定範囲は、3 ~ 10000 の整数です。

Vコマンドの指定値と、@EAエンベロープ出力と、V_ROUND設定によって得られた音量指定値(* K)と、

VLコマンドの指定値と、@LA音量 L F O出力と、VL_ROUND設定によって得られた音量指定値(* L)は、

共通の仮想音量ハードを想定して音量倍率を適用します。

まず、(* K)とVH_RATE によるデシベル倍率を、振幅に乗算します。

@LAが有効の場合は、さらに (* L)とVH_RATE によるデシベル倍率を、振幅に乗算します。

VH_RATE：

仮想音量ハードの変化デシベル数を指定します。

設定範囲は、-24 以上、0 未満で、単位はデシベルです。

小数以下の指定も受け付けます。

例えば、-1.5と指定した場合、音量が1減るごとに、振幅が -1.5dB されます。

VH_V0MODE：

仮想音量ハードに音量0が指定された時に、無音とするか、デシベル計算するか (無音にならない) のモードを指定します。

設定は、0 または 1 で行います。

0 のとき、音量0で無音になります。

1 のとき、音量0で他の音量同様のデシベル計算を行います。

4.14. #ML:REVERSE-VOLUME : 相対ボリューム記号反転

記述	#ML:REVERSE-VOLUME
----	--------------------

種別	メタデータ指定
----	---------

【解説】

相対ボリューム変更記号の上下反転を行います。

メタデータとして

#ML:REVERSE-VOLUME

と記述すると、相対ボリュームの上げ下げ（小カッコ）の機能が反転します。

【例】

```
#ML:REVERSE-VOLUME
```


5. 制御関連

5.1. プリプロセッサの処理順序

【解説】

MMLの演奏要求（PLAY）を受け付けると、MMLテキストがコンパイルされて演奏が始まります。

ここで言うコンパイルとは、MMLテキストを演奏できる形式に翻訳することですが、この翻訳の直前に、MMLテキスト全体に対し前処理を行なっていて、この前処理を行う機能のことをプリプロセッサと呼んでいます。

プリプロセッサの処理順序は次の通りです。

- (1) コメントブロック処理
- (2) コメントライン処理
- (3) メタデータ処理
- (4) マクロ処理
- (5) 空白を除去し、アルファベットを全て小文字化
- (6) 繰り返しの展開処理

上記の処理を順に経てからMMLコンパイルに入ります。

そのため、

- ・コメントブロック中のコメントライン処理は無効
 - ・メタデータやマクロでは大文字小文字が区別される
- など、解釈の挙動に特徴があります。

5.2. T：テンポ

記述	T[1],[2]
種別	MML コマンド

【解説】

テンポを設定します。

指定する引数は次の通りです。

引数[1]...テンポ (bpm)

引数[2]...分母 (denom)

引数はカンマで区切って指定します。カンマとdenomの組は省略できます。トラック先頭におけるデフォルトは、T120 です。

引数[1] (bpm)

4分音符を1ビートとした、1分間のビート数 (BPM) を指定します。小数以下の指定も受け付けます。

また、記号付きの数値を指定することで、前回テンポ設定値に対する相対指定を行うことができます。

(記号1) + (プラス)	前回設定値への加算
(記号2) - (マイナス)	前回設定値への減算
(記号3) * (アスタリスク)	前回設定値への乗算

ただし、この相対指定における「前回」とは同一トラック内での指定値を指しません。MMLコンパイラの都合、他のトラックにおける過去指定を認識できないので、利用シーンは同一トラック内で徐々にゆっくりさせる指定を行う場合などです。

引数[2] (denom)

denom は、bpm で指定した数値を除算したい場合に指定します。

denom は省略可能で、省略した場合は denom に1を指定したものとみなされます。

【例1】通常のテンポ指定

```
T100 CDEFG;
```

BPMを100に設定します。

【例 2】除算のあるテンポ指定

```
T3600,28 CDEFG;
```

BPMを128.5714...に設定します。

これは、tickカウント1あたり1 / 60秒の前提で、
16分音符のtickカウントが7になるようなテンポにしたい場合、
4分音符は28 ticksになるので、秒数は $28 \times 1 / 60$ 。
よって、
 $\text{tempo} = 60 / (28/60) = 3600/28$ という計算です。

【例 3】だんだんゆっくりにするテンポ指定

```
L4 T120 cde T-20 def T-20 efg T+40 c1;
```

T120 で開始しているので、
T-20 def は、この場合 T100 で演奏、
T-20 efg は、この場合 T80 で演奏、
T+40 c1 は、この場合 T120 で演奏されます。

【例 4】だんだんゆっくりにするテンポ指定 (2)

```
L4 T120 cde T*0.5 def T*0.5 efg T*4 c1;
```

T120 で開始しているので、
T*0.5 def は、この場合 T60 で演奏、
T*0.5 efg は、この場合 T30 で演奏、
T*4 c1 は、この場合 T120 で演奏されます。

【備考 1】

テンポコマンドでは、同時点の全てのトラックに対し影響します。
テンポ同期の内部手順は次の通り。

(1) MML コンパイル中に受け取ったテンポコマンドは、テンポ専用のシステム管理トラックに、一旦全て記録する (全トラック分を集約)。

(2) 全てのトラックのMML コンパイルが終わった後、テンポ専用トラックを読み、各トラックの同時点に対し、テンポコマンドを挿入する。

...以上です。

同時点かどうかの管理には、再生開始からの累積tickカウント数を使用しています。

【備考 2】

テンポの精度は、内部におけるtickカウント数あたりの再生サンプル数 (整数) をいくつとするかの計算に依存しています。

【備考3】

複数トラックで、同時点に、指定値の違うテンポコマンドを記述した場合、後のトラックに記述したテンポコマンドが使われます。

内部的に、同時点のテンポコマンドは、先に定義されたトラックのものから順に各トラックに配信されるため、結果的に最後のトラックに記述したテンポコマンドが有効になります。

【備考4】

トラックごとにテンポの同期を厳守しようとする設計思想になっています。この設計は、無限リピートの仕様にも影響しています。

5.3. ; (セミコロン) :トラック区切り

記述	; (セミコロン)
種別	MML 擬似コマンド/マクロ定義終端

【解説】

セミコロンは、次の用途で使います。

- (1) トラック定義終端
- (2) マクロ定義終端

(1) の用途は、複数トラックを定義したい場合の区切りです。トラック終端にセミコロンを設けると、それまでのMML記述が1つのトラックとして認識され、以降を次のトラックの先頭として扱います。(トラックとトラックが区切られます)

最終トラックにはセミコロンはあってもなくても構いませんが、トラックがいつでも追加できるよう準備をしておくという観点から、最終トラック含め、全てのトラックの終端にはセミコロンを記述しておくことをお勧めします。

【備考1】

最終トラック記述後、セミコロン無しでテキストが終わった場合は、テキスト終端がトラック終端として扱われます。

【備考2】

最終トラック記述後、セミコロン有りでテキストが終わった場合、テキスト終端で何のMMLイベントも無いトラックが終端処理されますが、イベントの無いトラックは定義されずにその場で破棄されます。

【備考3】

MML記述のないままセミコロンで終端を迎えたトラックは、MMLイベントのないトラックとみなされ、定義されずに破棄されます。セミコロンを連続記述したとしても、空のトラックは全て破棄されます。

(2) の用途は、マクロ定義、
\$...=.....;
の終端記述です。

5.4. /* ... */ : コメントブロック

記述	/* ... */
種別	プリプロセッサコマンド

【解説】

コメントブロックを設定します。

/* と */ に囲まれた文字列を、コメントエリアとして読み飛ばします。
コメントエリア内には、全角文字や改行を含んでも問題ありません。

【備考】

プリプロセッサの処理順序により、コメントライン処理よりも、コメントブロック処理が優先されます。

【例】

```
/* ----- // ----- */
```

この場合、コメントブロックが優先的に処理されるため、行の途中のコメントライン開始記述は、コメントブロック内のコメントとして処理されます。

5.5. // : コメントライン

記述	//
----	----

種別	プリプロセッサコマンド
----	-------------

【解説】

コメントラインを設定します。

```
//
```

を記述して以降、改行までをコメントエリアとして読み飛ばします。
コメントエリア内には全角文字を含んでも問題ありません。

【備考】

プリプロセッサの処理順序により、メタデータ処理よりも、コメントライン処理が優先されます。

【例】

```
#ML:TITLE テストタイトル//test title
```

この場合、「//test title」の文字列は、コメントライン処理により、「#ML:TITLE」によるタイトル文字列には含まれないこととなります。

5.6. |: ... :| : 繰り返し

記述	:[1]... ...:
種別	プリプロセッサコマンド

【解説】

有限回数の繰り返し演奏に使用します。

引数[1]には、繰り返し回数を0以上の整数で指定します。

繰り返し記述を行うと、「|:」から「:|」の間が、引数[1] の回数だけ繰り返しになります。

ただし、繰り返し内に「|」がある場合は、繰り返し最終回するとき、

「|」から「:|」の間がスキップされます（繰り返し脱出）。

繰り返し脱出の「|」は省略可能です。

引数[1]の繰り返し回数は、省略可能です。省略時の繰り返し回数は2です。

【備考】

この繰り返し機能は、プリプロセッサにより処理されます。

つまり単純な文字列のコピーペーストによる実装なので、無限リピートは出来ません。

【例1】

```
|:3 cde | fg :|
```

この場合「cde fg cde fg cde」と同じになります。

【例2】

```
|: cde | fg :|
```

この場合「cde fg cde」と同じになります。

【例3】

```
|:1 cde | fg :|
```

この場合、「cde」と同じになります。

【例4】

```
|:0 cde | fg :|
```

この場合、繰り返しの中身が空となります（コメントアウト状態）。

この特徴は、部分的にMML記述の有効・無効を切り替えることに利用できます。

5.7. @RP：無限リピートエントリ

記述	@RP
種別	MML コマンド

【解説】

無限リピートエントリを登録します。

このコマンドを記述したトラックでは、演奏がトラック終端に達したら、@RP を記述した時点で演奏ポイントが無条件ジャンプします。そのため、結果的に無限リピートになります。

無限リピートコマンドを使用する場合は、対象となるトラック内で「1回だけ」記述してください。

無限リピートコマンドは、テンポコマンドの使用状況と大きく関係しています。理由は、各トラック間でのテンポ同期を確保するためです。

テンポ同期の都合、無限リピートの処理モードを、テンポコマンドの配置の仕方によって変えています。

(1) 規制緩和モード

このモードは、テンポコマンドが、演奏先頭 (0 tick時点) にのみ存在する場合に該当します。

【規制緩和モードの場合の @RP 使用条件】

(1.1) 各トラックの tickカウント合計の状況は問われません。

(1.2) @RP コマンドは、どのトラックでも自由な時点で配置できます。

以上により、無限リピートの自由度が高まりますが、音楽的におかしなリピートも可能になるため、ユーザー側でリピート状況を確認する必要があります。

(2) 厳密モード

このモードは、テンポコマンドが、演奏先頭だけでなく、それ以外の時点にも存在する場合に該当します。厳密モードでは、テンポ同期を厳守する目的で、無限リピートに関する制限を厳しくしています。

【厳密モードの場合の @RP 使用条件】

(2.1) 無限リピート対象の、全てのトラックは、演奏tickカウント数の合計が一致している必要があります。

(2.2) 無限リピート対象の、全てのトラックにおいて、@RP を配置する時点（演奏開始からのtickカウント数）が一致している必要があります。

(2.3) 無限リピートしないトラックが混在する場合、無限リピートしないトラックの演奏tickカウント数の合計は、無限リピートするトラックのもの以下でなければなりません。

厳密モードによる制限の理由：

テンポの変更は、演奏サンプリング周波数(192kHz)に依存した、内部音長である「絶対tickカウント数」の設定変更により実装されています。絶対tickカウント数の操作は、テンポ計算にのみ任されているので、普段は意識する必要はありません。

しかし、複数回テンポ変更する無限リピート対象トラックで、厳密モードによって制限しなかった場合、見かけ上テンポのつじつまを合わせたとしても、使用する音長の種類と個数の違いによって、簡単に内部音長である絶対tickカウント数が合わなくなって、テンポを同期させることができなくなります。
(システム都合のテンポ非同期が発生します)

そのため、厳密モードでは、無限リピートで何周しても同期するよう、テンポ変更のタイミング、無限リピートエントリの位置、トラックの演奏tickカウント数の合計を合わせることを条件にしています。

(2.3) の制限は、2周目以降のリピートに対応する同期が、無限リピートしないトラックに対して行えないために設けています。

【備考1】@RPの優先配置

@RP コマンドは、MML コンパイル時、同時点（演奏開始からの累積tickカウント数が同じ時点）に存在する他のコマンドより前に配置されます。

例えば、

```
@@pls v8 cdefg @@saw v12 @RP gfedc;
```

という記述の場合、見た目では @@saw v12 が無限リピート開始時に実行されないように見えますが、内部的には、

```
@@pls v8 cdefg @RP @@saw v12 @RP gfedc;
```

このように同時点コマンド群の先頭に自動配置されるため、同時点のコマンドはすべて無限リピートエントリ時に実行の対象になります。

この、自動配置の目的は、テンポコマンドの実行もれ防止です。

@RPと同時点にテンポコマンドが存在した場合の、実行もれを防止することで、テンポ同期が取れない状況が発生しないようにしています。

【備考2】 2周目以降に向けたMML擬似コマンドは無効

無限リピート開始時点において、MML擬似コマンドによる演奏設定は、常に無限リピート初回演奏の条件での適用になります。

MML擬似コマンドとは、例えば L4 などです。

【例1】 音長指定

```
L4 cdef @RP gab>c L1;
```

この場合、リピート始めの g は、常に L4 で演奏されてしまいます。これは、MMLコンパイル時点で g が L4 でコンパイルされて音長が確定していて、トラック終端に L1 指定があったとしても、無限リピートでジャンプした先にある、既に生成済みの音符の音長に作用できないためです。

同様に、他のMML擬似コマンドも作用できない性質を持っています。

これは無限リピート機能特有の問題です。

【例2】 相対ボリューム

```
V13 L4 cdef @RP gab>c (2;
```

相対ボリューム（小カッコ）は、Vコマンドの固定値に変換されているため、トラック終端に置いて無限リピートする場合、置いた1回分のみ有効です。

この例では記述上、繰り返すたびにボリュームが小さくなっていくかのように見えますが、実際には、

```
V13 L4 cdef @RP gab>c V11;
```

このように記述したのと同様の動作になります。

【例3】 音程の補助指定

0

<

>

NS

これら音程関連のコマンドは、音符生成時に参照され、音程番号に変換されて演奏データになるため、トラック終端に置いても、音長指定同様、リピート開始時の音符に作用できません。

【例4】 指定値の分母設定関連

Q の分母指定

@W の分母指定

これらの分母指定も、MMLコンパイル時点で参照されて、コマンド生成されるため、トラック終端で分母を設定変更しても無限リピート開始時に作用できません。

5.8. !: MML ステータス表示

記述	![1]
種別	MML 擬似コマンド

【解説】

MML 編集中のデバッグ表示として、当コマンド指定時の各種情報を、Warning ウィンドウに表示します。

引数[1]は整数で指定します。指定値が、情報表示のID番号になります。指定しなかった場合、0 を指定したとみなされます。

【例】

```
V13 cde (( !10 fg (( !11 ab (( !
```

警告表示領域に、ID:10、ID:11、ID:0 の順に、当コマンド記述時点での各種ステータスが表示されます。

5.9. !#0 : 音符ログ

記述	!#0,[1]
種別	MML 擬似コマンド

【解説】

MML 編集時のデバッグ表示として、当コマンド記述以降の音符の履歴情報（ログ）を、Warningウィンドウに表示します。

引数[1]は整数で指定します。

- ・ 指定値が1のとき、音符ログの開始
- ・ 指定値が0のとき、音符ログの停止

指定しなかった場合、0を指定したとみなされます。

【例】

```
!#0,1 cdefgr  
!#0,0 gfedcr;
```

この場合、前半のcdefgrの音符部分のログのみが表示されます。

5.10. !#1 : 休符ログ

記述	!#1,[1]
種別	MML 擬似コマンド

【解説】

MML 編集時のデバッグ表示として、当コマンド記述以降の休符の履歴情報（ログ）を、Warning ウィンドウに表示します。

引数[1]は整数で指定します。

- ・ 指定値が1のとき、休符ログの開始
- ・ 指定値が0のとき、休符ログの停止

指定しなかった場合、0 を指定したとみなされます。

【例】

```
!#1,1 cdefgr  
!#1,0 gfedcr;
```

この場合、前半のcdefgrの休符部分のログのみが表示されます。

5.11. @DSP : KeyDispゲージへの表示指示

記述	@DSP[1],[2],[3]
種別	MML コマンド

【解説】

KeyDispの表示領域における、
ボリューム (V) ゲージ
エンベロープ (@EA) ゲージ
ボリュームL (VL) ゲージ
音量LFO (@LA) ゲージ

に対して、強制的に上書きで表示指示を行います。
これは表示のみで、音には全く影響を及ぼしません。

例えば、YコントロールでFM音源モジュールのTLを操作した後、手動で音量ゲージを表示させる手段として利用することが出来ます。

指定する引数は次の通りです。

引数[1]...ゲージ表示レベルの分子 (num)
引数[2]...ゲージ表示レベルの分母 (denom)
引数[3]...ゲージ表示の宛先 (dest)
引数はカンマで区切って指定します。

引数[1] (num)

ゲージ表示させたい大きさの分子を指定します。
ゲージ表示させるには、分子÷分母の結果を 0~1 の範囲に収めてください。
ゲージ表示を解除するには、分子÷分母が負数になるよう指定してください。
(解除になると、通常通りの表示を行います)
接頭辞「0x」をつけて記述すると、16進数整数として解釈されます。
この引数では加減算記述も可能です。
例えば、「127-3」といった表記ができます。

引数[2] (denom)

ゲージ表示させたい大きさの分母を指定します。
分母に 0 は指定できません。
接頭辞「0x」をつけて記述すると、符号なし16進数整数として解釈されます。

引数[3] (dest)

ゲージ表示の宛先を、2文字の識別子で指定します。

【@DSPコマンドの宛先となる識別子一覧】

宛先の識別子	内容
VE	ボリューム (V) のゲージ
EA	エンベロープ (@EA) のゲージ
VL	ボリューム L (VL) のゲージ
LA	音量 LFO (@LA) のゲージ

(識別子は大文字・小文字を問わず)

【例 1】

```
@dsp13,15,VE
```

ボリュームゲージの表示を、
全体の 13/15 の大きさで表示します。

【例 2】

```
@dsp127-5,127,VE
```

ボリュームゲージの表示を、
全体の 122/127 の大きさで表示します。

5.12. \$名前=内容; : マクロ定義

記述 \$名前=内容;

種別 プリプロセッサコマンド

【解説】

マクロ機能は、頻繁に使うMMLコマンド群に「名前」をつける機能です。一度「名前」つければ（マクロ定義）、後はその名前を書くだけ（マクロ参照）でMMLコマンド群を呼び出せるので、記述が楽になります。

マクロの定義方法：

トラック先頭（MMLコマンドを1つも記述していない状態のトラック）で、
\$名前=内容;

このように、「\$」で始めてマクロ名を書き、次に「=」を書いてからMMLコマンド群を書いて、最後に「;」で締めくくすることでマクロを定義します。マクロの内容には改行を含んでも良いので、長いフレーズの定義も可能です。また、トラック先頭であれば、複数のマクロを定義できます。

すでに定義されたマクロを、以降のマクロ定義で参照するような、入れ子的なマクロ定義も可能です。

マクロ定義はセミコロン「;」で締め括りますが、トラック定義とは別物と考えてください。トラック先頭において「\$」で始まる部分は、マクロ定義モードと解釈されます。しかし、マクロ定義後に、1つでもMMLコマンドを記述すると、トラック定義モードとなり、「\$」は定義済みマクロの参照開始とみなされますので、MML記述中にマクロを定義することは出来ません。

マクロの名前の付け方には次の制限があります。

- ・ 名前の1文字目はアルファベット又はアンダーライン(_)。
- ・ 名前の2文字目以降は、
アルファベット、アンダーライン(_)、
数字、4種の記号【 + # () 】
のいずれかです。

マクロの参照方法：

マクロ定義されたトラック以降で、

```
$名前
```

のように記述します。こうすると、「\$名前」が「内容」に置き換えられます。この置き換えは、プリプロセッサにより行われます。

【マクロの定義と参照の例】

トラック先頭の状態、次のように書いた場合、

```
$C=ecgc;
$C $C $C;
```

上記のトラック定義は、

```
ecgc ecgc ecgc;
```

と同じことになります。

MMLコンパイル時、プリプロセッサにより、マクロ参照部分がマクロ定義内容に置き換えられています。

マクロ定義部分は、プリプロセッサによる置き換え処理の際に、除去される格好です。

入れ子的にマクロを利用する場合は、

```
$C=ecgc;
$D=fdad;
$M= $C $D;
T150 L16 05 $M;
```

といった感じになります。

マクロ機能は単純な文字列の置き換えなので、ちょっと変わった使い方もできます。

例えば、次のように書いた場合、

```
$C=cd;
L16 $C2$C4$C8;
```

上記のトラック定義は、

```
L16 cd2cd4cd8;
```

と同じこととなり、dの音長だけを変えて記述しています。

マクロの定義は、トラック定義の先頭部分に集まっていなければならない制限があります。

使用できる記述例：

```
$T1=@5 v10;  
$T2=@8 v12;  
$T1 cgeg $T2 cgeg $T1 egcgc;  
$T3=@10 v7;  
$T4=@11 v12;  
$T1 cccc $T3 cccc $T4 cccccc;
```

上記では、マクロの定義が1トラック目と2トラック目の先頭部分に集められていて、2トラック目では、それまでに定義した全てのマクロが参照できることが確認できます。

エラーになる例：

```
@0 cgeg  
$T1=@5 v10;  
$T1 egcgc;
```

上記では、MML 定義が始まってからマクロ定義をしようとしており、エラーになります。

これは、MML 記述が始まった時点で、そのトラックでは「\$」文字がマクロ定義開始ではなく、マクロ参照開始と認識されるからです。

MML 記述が始まると、セミicolonもマクロ内容終端ではなく、MMLトラック終端として認識されます。

・ 引数つきマクロ定義

引数つきマクロの定義方法：

```
$名前{引数1,引数2,...}=内容;
```

このように、マクロ名につづけて中カッコを書き、その中に引数名を書いて、「=」後に内容を書くと、引数つきマクロが定義できます。

引数につけられる名前の書式は、マクロ名と同じです。

そして、内容の定義内で、引数を使いたいところ（引数参照）では、「%引数名」と書くことで引数が参照できます。

引数つきマクロの参照方法：

```
$名前{引数1,引数2...}
```

このように、マクロ名につづけて中カッコを書き、その中に引数を書くことで、引数つきマクロが参照できます。

【引数つきマクロの定義と参照の例】

トラック先頭の状態、次のように書いた場合、

```
$m{note}=v13 %note16 v8 %note16 R8;  
$m{C} $m{D} $m{E} $m{F} $m{G};
```

上記のトラック定義は、

```
v13 C16 v8 C16 R8 v13 D16 v8 D16 R8 v13 E16 v8 E16 R8 v13 F16 v8 F16 R8  
v13 G16 v8 G16 R8 ;
```

と同じこととなります。同じマクロを %note の部分を変更しながら使い回すことができます。

引数は複数指定することもできます。

```
$m{vol,note}=v%vol %note;  
$m{13,CDE} $m{10,EFG};
```

上記のトラック定義は、

```
v13 CDE v10 EFG;
```

と同じこととなります。

5.13. #ML:REPORT-TICKS : 総TICKカウント数の表示

記述	#ML:REPORT-TICKS
種別	メタデータ指定

【解説】

定義済みトラックの tickカウント合計情報を表示します。

メタデータとして、

#ML:REPORT-TICKS

と記述すると、集計された tickカウント合計情報を、Warningウィンドウに表示します。

表示内容は次の通りです。

- ・ 演奏tickカウント数の合計。
- ・ 無限リピート時の、リピートエントリ時点の tickカウント数。
- ・ 無限リピート時の、リピート対象部分の tickカウント数の合計。

以上が、定義済みトラックごとに表示されます。

【例】

```
#ML:REPORT-TICKS
```

6. 音色関連

6.1. @@ : 使用音源モジュール指定

記述 @@[1]-[2]

種別 MML コマンド

【解説】

使用する音源モジュールを指定します。

指定する引数は次の通りです。

引数[1]...音源モジュール名の識別子 (form)

引数[2]...サブモジュール番号 (subform)

引数はハイフンで区切って指定します。

ハイフンと subform の組 (枝番指定) は省略できます。

トラック先頭におけるデフォルトは、@@PLS-0 です。

引数[1] (form)

使用する音源モジュール名を 3 文字の識別子で指定します。

識別子の一覧は次の通りです。

SIN	サイン波	NZP	PSG ノイズ
SAW	ノコギリ波	NZF	FC ノイズ
TRI	三角波	NZG	GB ノイズ
PLS	パルス波	NZC	波形メモリノイズ
WVM	波形メモリ	DPC	DPCM
FMS	FM 音源	PCM	PCM
NZW	ホワイトノイズ	ADP	ADPCM

(識別子は大文字・小文字を問わず)

引数[1] (subform)

form で指定した音源モジュールに伴うサブモジュール番号を、0 以上の整数で指定します。

ハイフンと subform のセットは省略可能で、省略した場合は subform に 0 を指定したものとみなされます。

【例】

```
@@sin CDE  
@@saw EFG  
@@pls-1 GAB;
```

この場合、サイン波でCDEを演奏し、次にノコギリ波でEFGを演奏、最後にノイズ混じりのパルス波でGABを演奏します。

6.2. @ : 音源サブモジュール番号指定

記述	@[1]
種別	MML コマンド

【解説】

当コマンド指定時、既に選択されている音源モジュールに対し、サブモジュール番号だけを指定します。

引数[1]には、0 以上の整数を指定します。
トラック先頭におけるデフォルトは「@@PLS-0」のため、サブモジュール番号のデフォルトは「0」です。

【例】

```
@@fms-0 CDE  
@18 EFG  
@7 GAB;
```

F M音源モジュールの、サブモジュール 0番 に登録した音色でCDEを演奏し、次に F M音源の 18番 の音色でEFGを演奏し、最後に F M音源の 7番 の音色でGABを演奏します。

6.3. @@SIN : サイン波音源

記述 @@SIN-[1]

種別 MML コマンド

【解説】

使用する音源モジュールを、サイン波音源にします。

引数[1]は、枝番指定で行う音源サブモジュール番号指定です。

枝番指定は省略可能です。省略した場合は 0 を指定したものとみなされます。

@@SIN と @@SIN-0 は全く同じ指定内容になります。

【特殊指定】

当音源は、マルチオペレータモードをサポートしています。マルチオペレータモードとは、仮想の発振器を複数用意し、コーラスや和音を行うモードです。

当音源のオペレータモードの切り替えは、音源サブモジュール番号指定 (@番号) で行います。

@0 : 1オペレータモード。1声(OP1)で発声します。

@20 : 2オペレータモード。2声(OP1-OP2)で発声します。

@30 : 3オペレータモード。3声(OP1-OP3)で発声します。

@40 : 4オペレータモード。4声(OP1-OP4)で発声します。

@50 : 5オペレータモード。5声(OP1-OP5)で発声します。

@60 : 6オペレータモード。6声(OP1-OP6)で発声します。

@70 : 7オペレータモード。7声(OP1-OP7)で発声します。

マルチオペレータモードでは、発音時の音程と位相が一致する場合、純粹に振幅がオペレータの数だけ倍になるため、音割れ防止のため振幅（音量）の抑制が必要になります。

【Yコントロールにおける、音源モジュール設定】

当音源宛の Yコントロールは、Y[F],[P],[D] のように書きます。

各引数は次の通りです。

[F] : 機能番号 (0以上の整数)

[P] : パラメータ (浮動小数点数。少数以下切り捨てで整数利用も有り。)

[D] : 宛先となる音源モジュールの識別子

例えば、Y3,8,SIN のように書きます。

Yコントロールの記述の2回目以降は宛先「,SIN」は省略できます。

以降の説明では宛先省略で記述します。

Y1,[P] :
音源サブモジュール番号を [P] に設定します。

Y3,[P] :
マルチオペレータモード時の音程比を、
コーラスモード [P] cent に設定します。

OP1=0 cent

OP2=[P] cent

OP3=(-[P]) cent

OP4=([P]/2) cent

OP5=(-[P]/2) cent

OP6=([P]/4) cent

OP7=(-[P]/4) cent

に設定します。

(デフォルトは8、範囲は -100~100)。

Y4,[P] :
全てのオペレータの音量比を [P] dB に設定します。

Y9,[P] :
マルチオペレータモードにおける音割れ防止の目的で、音量比のオートバランサー機能の有効無効を指定します。

0 の場合、オートバランサー無効

1 の場合、オートバランサー有効 (デフォルト)

オートバランサーによる合計振幅への倍率計算は、

倍率 = 全有効オペレータの音量倍率の合計の逆数
により求められます。

Y10,[P] :
全オペレータの、音程比をコーラスモード [P] cent、
音量比を 0 dB に設定します。

Y11,[P] : オペレータ1の音程比を、[P] cent に設定します。

Y12,[P] : オペレータ1の音量比を、[P] dB に設定します。

Y21,[P] : オペレータ2の音程比を、[P] cent に設定します。

Y22,[P] : オペレータ2の音量比を、[P] dB に設定します。

Y31,[P] : オペレータ3の音程比を、[P] cent に設定します。

Y32,[P] : オペレータ3の音量比を、[P] dB に設定します。

Y41,[P] : オペレータ4の音程比を、[P] cent に設定します。

Y42,[P] : オペレータ4の音量比を、[P] dB に設定します。

Y51,[P] : オペレータ5の音程比を、[P] cent に設定します。

Y52,[P] : オペレータ5の音量比を、[P] dB に設定します。

Y61,[P] : オペレータ6の音程比を、[P] cent に設定します。

Y62,[P] : オペレータ6の音量比を、[P] dB に設定します。

Y71,[P] : オペレータ7の音程比を、[P] cent に設定します。

Y72,[P] : オペレータ7の音量比を、[P] dB に設定します。

各オペレータの音程比の範囲は、-9600 ~ 9600 [cent]で、
音量比の範囲は、-100 ~ 0 [dB]です。

6.4. @@TRI : 三角波音源

記述 @@TRI-[1]

種別 MML コマンド

【解説】

使用する音源モジュールを、三角波音源にします。
引数[1]は、枝番指定で行う音源サブモジュール番号指定です。
枝番指定は省略可能です。省略した場合は 0 を指定したものとみなされます。
@@TRI と @@TRI-0 は全く同じ指定内容になります。

【特殊指定】

当音源は、マルチオペレータモードをサポートしています。マルチオペレータモードとは、仮想の発振器を複数用意し、コーラスや和音を行うモードです。当音源のオペレータモードの切り替えは、音源サブモジュール番号指定 (@番号) で行います。

@0 : 1オペレータモード。1声(OP1)で発声します。
@20 : 2オペレータモード。2声(OP1-OP2)で発声します。
@30 : 3オペレータモード。3声(OP1-OP3)で発声します。
@40 : 4オペレータモード。4声(OP1-OP4)で発声します。
@50 : 5オペレータモード。5声(OP1-OP5)で発声します。
@60 : 6オペレータモード。6声(OP1-OP6)で発声します。
@70 : 7オペレータモード。7声(OP1-OP7)で発声します。

マルチオペレータモードでは、発音時の音程と位相が一致する場合、純粹に振幅がオペレータの数だけ倍になるため、音割れ防止のため振幅（音量）の抑制が必要になります。

【Yコントロールにおける、音源モジュール設定】

当音源宛の Yコントロールは、Y[F],[P],[D] のように書きます。

各引数は次の通りです。

[F] : 機能番号 (0以上の整数)

[P] : パラメータ (浮動小数点数。少数以下切り捨てで整数利用も有り。)

[D] : 宛先となる音源モジュールの識別子

例えば、Y3,8,TRI のように書きます。

Yコントロールの記述の2回目以降は宛先「,TRI」は省略できます。

以降の説明では宛先省略で記述します。

Y1,[P] :
音源サブモジュール番号を [P] に設定します。

Y3,[P] :
マルチオペレータモード時の音程比を、
コーラスモード [P] cent に設定します。

OP1=0 cent

OP2=[P] cent

OP3=(-[P]) cent

OP4=([P]/2) cent

OP5=(-[P]/2) cent

OP6=([P]/4) cent

OP7=(-[P]/4) cent

に設定します。

(デフォルトは8、範囲は -100~100)。

Y4,[P] :
全てのオペレータの音量比を [P] dB に設定します。

Y9,[P] :
マルチオペレータモードにおける音割れ防止の目的で、音量比のオートバランサー機能の有効無効を指定します。

0 の場合、オートバランサー無効

1 の場合、オートバランサー有効 (デフォルト)

オートバランサーによる合計振幅への倍率計算は、

倍率 = 全有効オペレータの音量倍率の合計の逆数
により求められます。

Y10,[P] :
全オペレータの、音程比をコーラスモード [P] cent、
音量比を 0 dB に設定します。

Y11,[P] : オペレータ1の音程比を、[P] cent に設定します。

Y12,[P] : オペレータ1の音量比を、[P] dB に設定します。

Y21,[P] : オペレータ2の音程比を、[P] cent に設定します。

Y22,[P] : オペレータ2の音量比を、[P] dB に設定します。

Y31,[P] : オペレータ3の音程比を、[P] cent に設定します。

Y32,[P] : オペレータ3の音量比を、[P] dB に設定します。

Y41,[P] : オペレータ4の音程比を、[P] cent に設定します。

Y42,[P] : オペレータ4の音量比を、[P] dB に設定します。

Y51,[P] : オペレータ5の音程比を、[P] cent に設定します。

Y52,[P] : オペレータ5の音量比を、[P] dB に設定します。

Y61,[P] : オペレータ6の音程比を、[P] cent に設定します。

Y62,[P] : オペレータ6の音量比を、[P] dB に設定します。

Y71,[P] : オペレータ7の音程比を、[P] cent に設定します。

Y72,[P] : オペレータ7の音量比を、[P] dB に設定します。

各オペレータの音程比の範囲は、-9600 ~ 9600 [cent]で、
音量比の範囲は、-100 ~ 0 [dB]です。

6.5. @@SAW : ノコギリ波音源

記述 @@SAW-[1]

種別 MML コマンド

【解説】

使用する音源モジュールを、ノコギリ波音源にします。
引数[1]は、枝番指定で行う音源サブモジュール番号指定です。
枝番指定は省略可能です。省略した場合は 0 を指定したものとみなされます。
@@SAW と @@SAW-0 は全く同じ指定内容になります。

【特殊指定】

当音源は、マルチオペレータモードをサポートしています。マルチオペレータモードとは、仮想の発振器を複数用意し、コーラスや和音を行うモードです。当音源のオペレータモードの切り替えは、音源サブモジュール番号指定 (@番号) で行います。

@0 : 1オペレータモード。1声(OP1)で発声します。
@20 : 2オペレータモード。2声(OP1-OP2)で発声します。
@30 : 3オペレータモード。3声(OP1-OP3)で発声します。
@40 : 4オペレータモード。4声(OP1-OP4)で発声します。
@50 : 5オペレータモード。5声(OP1-OP5)で発声します。
@60 : 6オペレータモード。6声(OP1-OP6)で発声します。
@70 : 7オペレータモード。7声(OP1-OP7)で発声します。

マルチオペレータモードでは、発音時の音程と位相が一致する場合、純粹に振幅がオペレータの数だけ倍になるため、音割れ防止のため振幅（音量）の抑制が必要になります。

【Yコントロールにおける、音源モジュール設定】

当音源宛の Yコントロールは、Y[F],[P],[D] のように書きます。

各引数は次の通りです。

[F] : 機能番号 (0以上の整数)

[P] : パラメータ (浮動小数点数。少数以下切り捨てで整数利用も有り。)

[D] : 宛先となる音源モジュールの識別子

例えば、Y3,8,SAW のように書きます。

Yコントロールの記述の2回目以降は宛先「,SAW」は省略できます。

以降の説明では宛先省略で記述します。

Y1,[P] :
音源サブモジュール番号を [P] に設定します。

Y3,[P] :
マルチオペレータモード時の音程比を、
コーラスモード [P] cent に設定します。

OP1=0 cent

OP2=[P] cent

OP3=(-[P]) cent

OP4=([P]/2) cent

OP5=(-[P]/2) cent

OP6=([P]/4) cent

OP7=(-[P]/4) cent

に設定します。

(デフォルトは8、範囲は -100~100)。

Y4,[P] :
全てのオペレータの音量比を [P] dB に設定します。

Y9,[P] :
マルチオペレータモードにおける音割れ防止の目的で、音量比のオートバランサー機能の有効無効を指定します。

0 の場合、オートバランサー無効

1 の場合、オートバランサー有効 (デフォルト)

オートバランサーによる合計振幅への倍率計算は、

倍率 = 全有効オペレータの音量倍率の合計の逆数
により求められます。

Y10,[P] :
全オペレータの、音程比をコーラスモード [P] cent、
音量比を 0 dB に設定します。

Y11,[P] : オペレータ1の音程比を、[P] cent に設定します。

Y12,[P] : オペレータ1の音量比を、[P] dB に設定します。

Y21,[P] : オペレータ2の音程比を、[P] cent に設定します。

Y22,[P] : オペレータ2の音量比を、[P] dB に設定します。

Y31,[P] : オペレータ3の音程比を、[P] cent に設定します。

Y32,[P] : オペレータ3の音量比を、[P] dB に設定します。

Y41,[P] : オペレータ4の音程比を、[P] cent に設定します。

Y42,[P] : オペレータ4の音量比を、[P] dB に設定します。

Y51,[P] : オペレータ5の音程比を、[P] cent に設定します。

Y52,[P] : オペレータ5の音量比を、[P] dB に設定します。

Y61,[P] : オペレータ6の音程比を、[P] cent に設定します。

Y62,[P] : オペレータ6の音量比を、[P] dB に設定します。

Y71,[P] : オペレータ7の音程比を、[P] cent に設定します。

Y72,[P] : オペレータ7の音量比を、[P] dB に設定します。

各オペレータの音程比の範囲は、-9600 ~ 9600 [cent]で、
音量比の範囲は、-100 ~ 0 [dB]です。

6.6. @@PLS : パルス波音源

記述	@@PLS-[1]
種別	MML コマンド

【解説】

使用する音源モジュールを、パルス波音源にします。

引数[1]は、枝番指定で行う音源サブモジュール番号指定です。

枝番指定は省略可能です。省略した場合は 0 を指定したものとみなされます。

@@PLS と @@PLS-0 は全く同じ指定内容になります。

【備考】

パルス波の音源モジュールは、@W コマンドにより、デューティ比の変更を受け付けます。デューティ比のデフォルト設定は 50% です。

【特殊指定】

当音源は、マルチオペレータモードをサポートしています。マルチオペレータモードとは、仮想の発振器を複数用意し、コーラスや和音を行うモードです。当音源のオペレータモードの切り替えは、音源サブモジュール番号指定 (@番号) で行います。

@0 : 1オペレータモード。1声(OP1)で発声します。

@1 : 1オペレータモード。ただし、ノイズ混じりのパルス波にします。ミックスされるノイズ音源モジュールは、@@NZP (PSGノイズ) です。

@@NZP に対しノイズサイクルを指定すれば、ミックスされるノイズサイクルの設定にもなります。

@2 : 1オペレータモード。ただし、PSGノイズにします。

@@NZP と同じ状態ですが、このモードは、YコントロールLFO (@LY) で矩形波とノイズを行き来させるような使用を想定しています。

@@NZPに対しノイズサイクルを指定すれば、@@PLS-2 のノイズサイクルの設定にもなります。

@20 : 2オペレータモード。2声(OP1-OP2)で発声します。

@30 : 3オペレータモード。3声(OP1-OP3)で発声します。

@40 : 4オペレータモード。4声(OP1-OP4)で発声します。

@50 : 5オペレータモード。5声(OP1-OP5)で発声します。

@60 : 6オペレータモード。6声(OP1-OP6)で発声します。

@70 : 7オペレータモード。7声(OP1-OP7)で発声します。

マルチオペレータモードでは、発音時の音程と位相が一致する場合、純粹に振幅がオペレータの数だけ倍になるため、音割れ防止のため振幅（音量）の抑制が必要になります。

【Yコントロールにおける、音源モジュール設定】

当音源宛の Yコントロールは、Y[F],[P],[D] のように書きます。

各引数は次の通りです。

[F] : 機能番号（0以上の整数）

[P] : パラメータ（浮動小数点数。少数以下切り捨てで整数利用も有り。）

[D] : 宛先となる音源モジュールの識別子

例えば、Y3,8,PLS のように書きます。

Yコントロールの記述の2回目以降は宛先「,PLS」は省略できます。

以降の説明では宛先省略で記述します。

Y1,[P] :

音源サブモジュール番号を [P] に設定します。

Y3,[P] :

マルチオペレータモード時の音程比を、
コーラスモード [P] cent に設定します。

OP1=0 cent

OP2=[P] cent

OP3=(-[P]) cent

OP4=([P]/2) cent

OP5=(- [P]/2) cent

OP6=([P]/4) cent

OP7=(- [P]/4) cent

に設定します。

（デフォルトは8、範囲は -100~100）。

Y4,[P] :

全てのオペレータの音量比を [P] dB に設定します。

Y5,[P] :

デューティ比（パルス幅）を設定します。

設定範囲は 0.005 以上 0.995 以下です。（0.5%~99.5%）

範囲外の場合は、0.005 未満は 0.005、0.995 超えは 0.995 に矯正されます。

（このデューティ比設定では、全てのオペレータが設定対象になります）

Y9,[P] :

マルチオペレータモードにおける音割れ防止の目的で、音量比のオート balanser 機能の有効無効を指定します。

0 の場合、オート balanser 無効

1 の場合、オート balanser 有効 (デフォルト)

オート balanser による合計振幅への倍率計算は、

倍率 = 全有効オペレータの音量倍率の合計の逆数
により求められます。

Y10,[P] :

全オペレータの、音程比をコーラスモード [P] cent、
音量比を 0 dB に設定します。

Y11,[P] : オペレータ1の音程比を、[P] cent に設定します。

Y12,[P] : オペレータ1の音量比を、[P] dB に設定します。

Y21,[P] : オペレータ2の音程比を、[P] cent に設定します。

Y22,[P] : オペレータ2の音量比を、[P] dB に設定します。

Y31,[P] : オペレータ3の音程比を、[P] cent に設定します。

Y32,[P] : オペレータ3の音量比を、[P] dB に設定します。

Y41,[P] : オペレータ4の音程比を、[P] cent に設定します。

Y42,[P] : オペレータ4の音量比を、[P] dB に設定します。

Y51,[P] : オペレータ5の音程比を、[P] cent に設定します。

Y52,[P] : オペレータ5の音量比を、[P] dB に設定します。

Y61,[P] : オペレータ6の音程比を、[P] cent に設定します。

Y62,[P] : オペレータ6の音量比を、[P] dB に設定します。

Y71,[P] : オペレータ7の音程比を、[P] cent に設定します。

Y72,[P] : オペレータ7の音量比を、[P] dB に設定します。

各オペレータの音程比の範囲は、-9600 ~ 9600 [cent]で、
音量比の範囲は、-100 ~ 0 [dB]です。

6.7. @@WVM：波形メモリ音源

記述	@@WVM-[1]
種別	MML コマンド

【解説】

使用する音源モジュールを、波形メモリ音源にします。

引数[1]は、枝番指定で行う音源サブモジュール番号指定です。

枝番指定は省略可能です。省略した場合は 0 を指定したものとみなされます。

@@WVM と @@WVM-0 は全く同じ指定内容になります。

【備考】

波形メモリ音源では、後述するメタデータ定義「#ML:WAVEM」によって、波形データに波形番号を割り当てて登録したものを使用します。

波形データの選択は、音源サブモジュール番号 (@番号) によって、波形番号を指定して行います。

【特殊指定】

当音源は、マルチオペレータモードをサポートしています。マルチオペレータモードとは、仮想の発振器を複数用意し、コーラスや和音を行うモードです。当音源のオペレータモードの切り替えは、@0 コマンド または、Y コントロールによって行います。

マルチオペレータモードでは、発音時の音程と位相が一致する場合、純粋に振幅がオペレータの数だけ倍になるため、音割れ防止のため振幅（音量）の抑制が必要になります。

【Yコントロールにおける、音源モジュール設定】

当音源宛の Y コントロールは、Y[F],[P],[D] のように書きます。

各引数は次の通りです。

[F]：機能番号（0以上の整数）

[P]：パラメータ（浮動小数点数。少数以下切り捨てで整数利用も有り。）

[D]：宛先となる音源モジュールの識別子

例えば、Y3,8,WVM のように書きます。

Y コントロールの記述の2回目以降は宛先「,WVM」は省略できます。

以降の説明では宛先省略で記述します。

Y1,[P] :

音源サブモジュール番号（波形番号）を [P] に設定します。

（この設定では、全てのオペレータの波形番号が設定対象になります）

Y2,0 :

1オペレータモード。1声(OP1)で発声します。

Y2,20 :

2オペレータモード。2声(OP1-OP2)で発声します。

Y2,30 :

3オペレータモード。3声(OP1-OP3)で発声します。

Y2,40 :

4オペレータモード。4声(OP1-OP4)で発声します。

Y2,50 :

5オペレータモード。5声(OP1-OP5)で発声します。

Y2,60 :

6オペレータモード。6声(OP1-OP6)で発声します。

Y2,70 :

7オペレータモード。7声(OP1-OP7)で発声します。

Y2,[P]の Yコントロールは、@0コマンド から呼び出されます。

パラメータの仕様は、宛先WVMの @0コマンド と同様です。

Y3,[P] :

マルチオペレータモード時の音程比を、
コーラスモード [P] cent に設定します。

OP1=0 cent

OP2=[P] cent

OP3=(-[P]) cent

OP4=([P]/2) cent

OP5=(- [P]/2) cent

OP6=([P]/4) cent

OP7=(- [P]/4) cent

に設定します。

（デフォルトは8、範囲は -100~100）。

Y4,[P] :

全てのオペレータの音量比を [P] dB に設定します。

Y5,[P] :

オーバーサンプリングモードを、[P] に設定します。

デフォルトは2です（2倍オーバーサンプリング）。

範囲は1以上の整数で、上限は「#ML:OVSWAVEM」で指定できる範囲と同様です。

Y9,[P] :

マルチオペレータモードにおける音割れ防止の目的で、音量比のオート balans 機能の有効無効を指定します。

0 の場合、オート balans 無効

1 の場合、オート balans 有効 (デフォルト)

オート balans による合計振幅への倍率計算は、

倍率 = 全有効オペレータの音量倍率の合計の逆数

により求められます。

Y10,[P] :

全オペレータの、音程比をコーラスモード [P] cent、音量比を 0 dB に設定します。

Y11,[P] : オペレータ1の音程比を、[P] cent に設定します。

Y12,[P] : オペレータ1の音量比を、[P] dB に設定します。

Y13,[P] : オペレータ1の波形番号を、[P] に設定します。

Y21,[P] : オペレータ2の音程比を、[P] cent に設定します。

Y22,[P] : オペレータ2の音量比を、[P] dB に設定します。

Y23,[P] : オペレータ2の波形番号を、[P] に設定します。

Y31,[P] : オペレータ3の音程比を、[P] cent に設定します。

Y32,[P] : オペレータ3の音量比を、[P] dB に設定します。

Y33,[P] : オペレータ3の波形番号を、[P] に設定します。

Y41,[P] : オペレータ4の音程比を、[P] cent に設定します。

Y42,[P] : オペレータ4の音量比を、[P] dB に設定します。

Y43,[P] : オペレータ4の波形番号を、[P] に設定します。

Y51,[P] : オペレータ5の音程比を、[P] cent に設定します。

Y52,[P] : オペレータ5の音量比を、[P] dB に設定します。

Y53,[P] : オペレータ5の波形番号を、[P] に設定します。

Y61,[P] : オペレータ6の音程比を、[P] cent に設定します。

Y62,[P] : オペレータ6の音量比を、[P] dB に設定します。

Y63,[P] : オペレータ6の波形番号を、[P] に設定します。

Y71,[P] : オペレータ7の音程比を、[P] cent に設定します。

Y72,[P] : オペレータ7の音量比を、[P] dB に設定します。

Y73,[P] : オペレータ7の波形番号を、[P] に設定します。

各オペレータの音程比の範囲は、-9600 ~ 9600 [cent]で、

音量比の範囲は、-100 ~ 0 [dB]です。

6.8. @@FMS : F M音源

記述	@@FMS-[1]
種別	MML コマンド

【解説】

使用する音源モジュールを、FM音源にします。

引数[1]は、枝番指定で行う音源サブモジュール番号指定です。

枝番指定は省略可能です。省略した場合は 0 を指定したものとみなされます。

@@FMS と @@FMS-0 は全く同じ指定内容になります。

【備考1】

FM音源では、後述するメタデータ定義

#MB:FM@SS60P (6オペレータType-SS)

#MB:FM@SS40P (4オペレータType-SS)

#MB:FM@SS20P (2オペレータType-SS)

#MB:FM@OPM (OPM互換の4オペレータ)

#MB:FM@OPNA (OPNA互換の4オペレータ)

#MB:FM@OPN (OPN互換の4オペレータ)

のいずれかによって、音色データに音色番号を割り当てて登録したものを使用します。

音色データの選択は、音源サブモジュール番号 (@番号) によって、音色番号を指定して行います。

【備考2】

当FM音源モジュールは、OPMの上位互換になっていて、内部にオペレータを6個持っています。

音色定義の書式は複数ありますが、複数の独立した音色定義領域を持っている訳ではなく、実体は、6オペレータモード用の記憶領域があるだけです。

どの音色記述方法でも、6オペレータモード用の記憶領域に設定されるため、音色番号3番は6オペモード、音色番号8番は4オペモードというように、オペレータモードが混在した定義になります。

同じ番号に複数の音色定義をした場合、最後の定義が有効になるので、音色番号の重複定義には注意してください。

【備考3】

有効な音程の範囲は、実際のOPMでは 3.58MHz 駆動の場合、

「o0C+ から o8C まで」ですが、当FM音源モジュールでは、MMLオクターブ指定における範囲と同等の範囲で受付可能です。

ただし、音色データ内のKS（キースケーリング）は、実際のOPM相当における最低音程以下には最低音程、最高音程以上には最高音程とみなして動作します。

・Yコントロール（1）音源モジュール設定

当音源宛の Yコントロールは、Y[F],[P],[D] のように書きます。

各引数は次の通りです。

[F]：機能番号（0以上の整数）

[P]：パラメータ（浮動小数点数。少数以下切り捨てで整数利用も有り。）

[D]：宛先となる音源モジュールの識別子

例えば、Y5,0,FMS のように書きます。

Yコントロールの記述の2回目以降は宛先「,FMS」は省略できます。

以降の説明では宛先省略で記述します。

Y1,[P]：

音源サブモジュール番号（音色番号）を [P] に設定します。

Y2,[P]：

波形生成モードの設定を、0 または 1 で行います。

0 のとき、ハイレゾ（192kHz）にします。

1 のとき、ネイティブにします。この場合、生成レートは「#ML:FMDLOCK」の設定に従い、 $\text{clock} \div \text{prescale}$ のレートになります。

Y5,[P]：

ノートオン時の初期エンベロープレベル（IEL）の設定を、

-1 または 0 ~ 128 で行います。デフォルトは -1 です。

-1 のとき、IEL は音色データ内の設定に従います。

0 ~ 128 のとき、全オペレータの IEL を指定値にします（音色データ内の IEL 設定は無視します）。

Y6,[P]：

フィードバックレベルへのバイアス調整値を [P] にします。

値の範囲は「#ML:FMFBBIAS」による指定と同様です。

範囲外の値を指定した場合、最大値または最小値に制限されます。

Y7,[P]：

ハードLFOの設定を行います。

このYコントロールは、@MH または @MHA により内部で呼び出すために使用されますので、通常は使用しないでください。

Y8,[P]：

エンベロープダンパー（消音機能）の設定、または、実行を行います。

エンベロープダンパーでは、休符途中でのリリースレートを一時的に操作することで消音を行います。

ダンパー実行後は、次のノートオンで元のリリースレートに自動復帰します。指定値による動作は次の通りです。

-1 のとき、設定済み内容でダンパーを実行します。

0 ~ 15 のとき、ダンパー実行の際、この指定値をキャリアのリリースレートに適用されるよう設定します（設定のみで消音は実行しません）。

16 のとき、ダンパー実行の際、全てのオペレータのリリースレートを 15 にするよう設定します（設定のみで消音は実行しません）。

Y10, [P] :

全オペレータへの各種加算値設定を 0 にリセットします。

0 のとき、detune3、TL、Y16 を 0 にリセットします。

1 のとき、detune3 加算値を 0 にリセットします。

2 のとき、TL 加算値を 0 にリセットします。

3 のとき、Y16 設定値を 0 にリセットします。

Y11, [P] :

音色設定内の AME が 1 になっているオペレータのみ、TL値が、音色設定の TL に [P] を加算した値になります。

このコマンドは、Y31~36の設定内容を、AME が 1 のオペレータを対象に上書きするものです。

Y15, [P] :

全オペレータの、現在のエンベロープ出力レベルに対し、[P] を直ちに加算します。（@mzコマンドと同様）

Y16, [P] :

全オペレータの、現在のエンベロープ出力レベルに対し、ノートオンの都度、ノートオンの直前に [P] を加算します。（@MZEコマンドと同様）

Y21, [P] : オペレータ1の detune3 に、[P] cent 加算します。

Y22, [P] : オペレータ2の detune3 に、[P] cent 加算します。

Y23, [P] : オペレータ3の detune3 に、[P] cent 加算します。

Y24, [P] : オペレータ4の detune3 に、[P] cent 加算します。

Y25, [P] : オペレータ5の detune3 に、[P] cent 加算します。

Y26, [P] : オペレータ6の detune3 に、[P] cent 加算します。

Y31, [P] : オペレータ1の TL に、[P] 加算します。

Y32, [P] : オペレータ2の TL に、[P] 加算します。

Y33, [P] : オペレータ3の TL に、[P] 加算します。

Y34,[P] : オペレータ4の TL に、[P] 加算します。

Y35,[P] : オペレータ5の TL に、[P] 加算します。

Y36,[P] : オペレータ6の TL に、[P] 加算します。

※Y21~26、Y31~36における加算では、加算値を記憶し、その値に現在設定されている音色データから対応する値を呼び出して加算した結果を音源モジュールにセットします。ただし、detune3 は -9600~9600、TL は 0~127 に制限されます。

Y50,[P] :

FM音源のハードLFOモードの選択を行います。

-1 のとき、ハードLFO処理を行いません。

0 のとき、OPMモード。

1 のとき、OPNAモード。

ハードLFO処理を行わない場合、その分処理が軽くなります。

Y51,[P] : OPM HARD LFO wave form (0~3)

Y52,[P] : OPM HARD LFO frequency (0~255)

Y53,[P] : OPM HARD LFO pmd (0~127)

Y54,[P] : OPM HARD LFO amd (0~127)

Y55,[P] : OPM HARD LFO pms/ams

pmsは0~7で上位4bit、amsは0~3で下位4bit の書式で指定します。例えば、pms=7、ams=3のときは、0x73なので、10進数では115で指定します。

Y56,[P] : OPM HARD LFO sync-mode (0~1)

Y57,[P] : OPM HARD LFO SpeedAdj.

OPMモードのハードLFOの速度調整を行います。

設定値 [P] の範囲は「#ML:FMHLF0S1」による指定と同様です。範囲外の値を指定した場合は、最大値または最小値に制限されます。

Y61,[P] : OPNA HARD LFO frequency (0~7)

Y62,[P] : OPNA HARD LFO ams/pms

amsは0~3で上位4bit、pmsは0~7で下位4bit の書式で指定します。例えば、ams=3、pms=7のときは、0x37なので、10進数では55で指定します。

Y63,[P] : OPNA HARD LFO sync-mode (0~1)

Y64,[P] : OPNA HARD LFO SpeedAdj.

OPNAモードのハードLFOの速度調整を行います。

設定値 [P] の範囲は「#ML:FMHLFOS2」による指定と同様です。範囲外の値を指定した場合は、最大値または最小値に制限されます。

・ Yコントロール（2）音色レジスタアクセス体系

FM音源の音色レジスタに直接設定する際に使用する、レジスタ番号アクセス体系を、3つのブロックにまとめてあります。

（A）ネイティブモードの音色レジスタ：Y101～202

この番号体系は、ネイティブモード（6オペレータモード）の音色定義の順番に対応させたものです。

ただし、ネイティブモードでの「CON」はオペレータモード変更に関わる都合、用意されていません。

※OPM/OPN(A)互換モードの「CON」では、0～7のみ指定可能です。

（B）OPM互換モードの音色レジスタ：Y256～Y511（Y0x100～Y0x1FF）

この番号体系は、OPM互換のレジスタ配列の順番に対応させたものです。

OPMレジスタ番号に256（0x100）加算した番号体系です。

ただし、有効なのはハードLFO機能および音色データ関連機能のみです。

また、どのチャンネル宛に設定しても自分自身のチャンネルとして受け付けます。

（C）OPN(A)互換モードの音色レジスタ：Y512～Y767（Y0x200～Y0x2FF）

この番号体系は、OPN(A)互換のレジスタ配列の順番に対応させたものです。

OPN(A)レジスタ番号に512（0x200）加算した番号体系です。

ただし、有効なのはハードLFO機能および音色データ関連機能のみです。

また、どのチャンネル宛に設定しても自分自身のチャンネルとして受け付けます。

【ネイティブモードの音色レジスタ：Y101～202】

OP1		OP2		OP3		OP4		OP5		OP6	
101	IPH	118	IPH	135	IPH	152	IPH	169	IPH	186	IPH
102	IEL	119	IEL	136	IEL	153	IEL	170	IEL	187	IEL
103	AR	120	AR	137	AR	154	AR	171	AR	188	AR
104	D1R	121	D1R	138	D1R	155	D1R	172	D1R	189	D1R
105	D2R	122	D2R	139	D2R	156	D2R	173	D2R	190	D2R
106	RR	123	RR	140	RR	157	RR	174	RR	191	RR
107	D1L	124	D1L	141	D1L	158	D1L	175	D1L	192	D1L
108	TL	125	TL	142	TL	159	TL	176	TL	193	TL
109	KS	126	KS	143	KS	160	KS	177	KS	194	KS
110	MUL	127	MUL	144	MUL	161	MUL	178	MUL	195	MUL
111	DT1	128	DT1	145	DT1	162	DT1	179	DT1	196	DT1
112	DT2	129	DT2	146	DT2	163	DT2	180	DT2	197	DT2
113	DT3	130	DT3	147	DT3	164	DT3	181	DT3	198	DT3
114	DT4	131	DT4	148	DT4	165	DT4	182	DT4	199	DT4
115	FB	132	FB	149	FB	166	FB	183	FB	200	FB
116	AME	133	AME	150	AME	167	AME	184	AME	201	AME
117	MSK	134	MSK	151	MSK	168	MSK	185	MSK	202	MSK

【OPM互換モードの音色レジスタ：Y256～Y511（Y0x100～Y0x1FF）】

	+0	+1	+2	+3	+4	+5	+6	+7
0x118	LFRQ	PMD/AMD		WF				
0x120	FB/CON							
0x138	PMS/AMS							
0x140	OP1: DT1/MUL							
0x148	OP3: DT1/MUL							
0x150	OP2: DT1/MUL							
0x158	OP4: DT1/MUL							
0x160	OP1: TL							
0x168	OP3: TL							
0x170	OP2: TL							
0x178	OP4: TL							
0x180	OP1: KS/AR							
0x188	OP3: KS/AR							
0x190	OP2: KS/AR							
0x198	OP4: KS/AR							
0x1A0	OP1: AME/D1R							
0x1A8	OP3: AME/D1R							
0x1B0	OP2: AME/D1R							
0x1B8	OP4: AME/D1R							
0x1C0	OP1: DT2/D2R							
0x1C8	OP3: DT2/D2R							
0x1D0	OP2: DT2/D2R							
0x1D8	OP4: DT2/D2R							
0x1E0	OP1: D1L/RR							
0x1E8	OP3: D1L/RR							
0x1F0	OP2: D1L/RR							
0x1F8	OP4: D1L/RR							

【OPN(A)互換モードの音色レジスタ：Y512～Y767 (Y0x200～Y0x2FF)】

	+0	+1	+2	+3	+4	+5	+6	+7
0x230	OP1: DT/MUL				OP3: DT/MUL			
0x238	OP2: DT/MUL				OP4: DT/MUL			
0x240	OP1: TL				OP3: TL			
0x248	OP2: TL				OP4: TL			
0x250	OP1: KS/AR				OP3: KS/AR			
0x258	OP2: KS/AR				OP4: KS/AR			
0x260	OP1: AME/DR				OP3: AME/DR			
0x268	OP2: AME/DR				OP4: AME/DR			
0x270	OP1: SR				OP3: SR			
0x278	OP2: SR				OP4: SR			
0x280	OP1: SL/RR				OP3: SL/RR			
0x288	OP2: SL/RR				OP4: SL/RR			
0x290								
0x298								
0x2A0								
0x2A8								
0x2B0	FB/CON							

・ Yコントロール（3）特殊動作対応：Y768(Y0x300)以降

Y0x300,[P] :

system1 opm-volume support

master volumeとch.volumeの設定です。引数からTL値を計算してキャリアのTLに音量の書き込みを行います。

[P]: 0xaabbccdd

aa: 未使用

bb: 未使用

cc: master volume 0-255

dd: ch.volume 0-255

master volume及びch.volumeは、書き込んだ値が記憶され、0x301もしくは0x302のコマンドで利用されます。

master volume及びch.volumeの初期値は双方255です。

負数を指定した場合、現状のmaster volume及びch.volumeにてTL値を再計算して設定します。負数指定は、音色の変更があった直後の使用を想定しています。

Y0x301,[P] :

system1 opm-volume support

master volume の設定です。引数からTL値を計算してキャリアのTLに音量の書き込みを行います。

[P]: master volume 0-255

ch.volumeは、以前書き込んだ値が読み出されて設定されます。ch.volumeの初期値は255です。

負数を指定した場合、現状のmaster volume及びch.volumeにてTL値を再計算して設定します。負数指定は、音色の変更があった直後の使用を想定しています。

Y0x302,[P] :

system1 opm-volume support

ch.volume の設定です。引数からTL値を計算してキャリアのTLに音量の書き込みを行います。

[P]: ch.volume 0-255

master volume は、以前書き込んだ値が読み出されて設定されます。master volume の初期値は255です。

負数を指定した場合、現状のmaster volume及びch.volumeにてTL値を再計算して設定します。負数指定は、音色の変更があった直後の使用を想定しています。

Y0x303, [P] :

system1 opm HLF0 support

HLF0パラメータのPMDとAMDの2つを1度に設定します。

[P]: 0xaabb

aa: PMDパラメータ

bb: AMDパラメータ

6.9. @@NZW : ホワイトノイズ音源

記述	@@NZW-[1]
種別	MML コマンド

【解説】

使用する音源モジュールを、ホワイトノイズ音源にします。
引数[1]は、枝番指定で行う音源サブモジュール番号指定です。
枝番指定は省略可能です。省略した場合は 0 を指定したものとみなされます。
@@NZW と @@NZW-0 は全く同じ指定内容になります。
今のところ @@NZW のサブモジュール番号は 0 のみです。

このホワイトノイズ音源では、変位がランダムに変更されます。
変位が更新される間隔を @N コマンドで変更することによって、ノイズの質感を
変えることができます。

【Yコントロールにおける、音源モジュール設定】

当音源宛の Yコントロールは、Y[F],[P],[D] のように書きます。

各引数は次の通りです。

[F] : 機能番号 (0以上の整数)

[P] : パラメータ (浮動小数点数。少数以下切り捨てで整数利用も有り。)

[D] : 宛先となる音源モジュールの識別子

例えば、Y5,1,NZW のように書きます。

Yコントロールの記述の2回目以降は宛先「,NZW」は省略できます。

以降の説明では宛先省略で記述します。

Y5,[P] :

ホワイトノイズの変位の更新間隔を設定します。

このYコントロールは、@Nコマンド から呼び出されます。

パラメータの仕様は、宛先NZWの @Nコマンド と同様に、

何サンプルごとに変位をリフレッシュするかを決定します。

例えば、パラメータに 1.5 と設定した場合には、1, 2, 1, 2... という具合に、
平均 1.5 サンプル毎に変位がリフレッシュされます。

6.10. @@NZIP : PSG ノイズ音源

記述 @@NZIP-[1]

種別 MML コマンド

【解説】

使用する音源モジュールを、PSG ノイズ音源にします。
引数[1]は、枝番指定で行う音源サブモジュール番号指定です。
枝番指定は省略可能です。省略した場合は 0 を指定したものとみなされます。
@@NZIP と @@NZIP-0 は全く同じ指定内容になります。
今のところ @@NZIP のサブモジュール番号は 0 のみです。

PSG ノイズは、パルス波のパルス幅が擬似ランダムに変化する「2値ノイズ」の一種で、パルス幅更新間隔と計算手順により固有の風合いを持っています。ただし、更新間隔が非常に短い2値ノイズは再生サンプリング周波数を超えた密度になるため、過密部分を相加平均処理しています。そのため、相加平均部分がホワイトノイズのように出力されます。

PSG ノイズ音源では、パルス幅の更新間隔を @N コマンド で変更することで、ノイズの質感を変えることができます。
加えて、@NC コマンド で、PSG ノイズの動作モードを変更することができます。デフォルトの動作モードは PC88 です。

【Yコントロールにおける、音源モジュール設定】

当音源宛の Yコントロールは、Y[F],[P],[D] のように書きます。

各引数は次の通りです。

[F]：機能番号（0以上の整数）

[P]：パラメータ（浮動小数点数。少数以下切り捨てで整数利用も有り。）

[D]：宛先となる音源モジュールの識別子

例えば、Y5,1,NZIP のように書きます。

Yコントロールの記述の2回目以降は宛先「,NZIP」は省略できます。

以降の説明では宛先省略で記述します。

Y5,[P]：

PSG ノイズのパルス幅更新間隔を設定します。

このYコントロールは、@N コマンド から呼び出されます。

パラメータの仕様は、宛先NZIPの @N コマンド と同様です。

Y8,[P] :

シフトレジスタ値の直接書き換えを行います。
このYコントロールは、@NRコマンド から呼び出されます。
パラメータの仕様は、宛先NZPの @NRコマンド と同様です。

Y10,[P] :

動作モードを設定します。
このYコントロールは、@NCコマンド から呼び出されます。
パラメータの仕様は、@NCコマンド と同様です。

6.11. @@NZF : FCノイズ音源

記述	@@NZF-[1]
種別	MMLコマンド

【解説】

使用する音源モジュールを、FCノイズ音源にします。

引数[1]は、枝番指定で行う音源サブモジュール番号指定です。

枝番指定は省略可能です。省略した場合は 0 を指定したものとみなされます。

@@NZF と @@NZF-0 は全く同じ指定内容になります。

@@NZF では、音源サブモジュール指定で、長周期モードと短周期モードの変更ができます。指定値とモードの対応は、

@@NZF-0 : 長周期モード

@@NZF-1 : 短周期モード

となります。

FCノイズは、パルス波のパルス幅が擬似ランダムに変化する「2値ノイズ」の一種で、パルス幅更新間隔と計算手順により固有の風合いを持っています。ただし、更新間隔が非常に短い2値ノイズは再生サンプリング周波数を超えた密度になるため、過密部分を相加平均処理しています。そのため、相加平均部分がホワイトノイズのように出力されます。

FCノイズ音源では、パルス幅の更新間隔を @Nコマンド で変更することで、ノイズの質感を変えることができます。

【Yコントロールにおける、音源モジュール設定】

当音源宛の Yコントロールは、Y[F],[P],[D] のように書きます。

各引数は次の通りです。

[F] : 機能番号 (0以上の整数)

[P] : パラメータ (浮動小数点数。少数以下切り捨てで整数利用も有り。)

[D] : 宛先となる音源モジュールの識別子

例えば、Y5,1,NZF のように書きます。

Yコントロールの記述の2回目以降は宛先「,NZF」は省略できます。

以降の説明では宛先省略で記述します。

Y1,[P] :

音源サブモジュール番号 (周期モード) を [P] に設定します。

指定値と周期モードの対応は次の通りです。

0 のとき、長周期モード。

1 のとき、短周期モード。

Y5,[P] :

FCノイズのパルス幅更新間隔を設定します。

このYコントロールは、@Nコマンド から呼び出されます。

パラメータの仕様は、宛先NZFの @Nコマンド と同様です。

Y8,[P] :

シフトレジスタ値の直接書き換えを行います。

このYコントロールは、@NRコマンド から呼び出されます。

パラメータの仕様は、宛先NZFの @NRコマンド と同様です。

6.12. @@NZG : GB ノイズ音源

記述 @@NZG-[1]

種別 MML コマンド

【解説】

使用する音源モジュールを、GB ノイズ音源にします。
引数[1]は、枝番指定で行う音源サブモジュール番号指定です。
枝番指定は省略可能です。省略した場合は 0 を指定したものとみなされます。
@@NZG と @@NZG-0 は全く同じ指定内容になります。

@@NZG では、音源サブモジュール指定で、長周期モードと短周期モードの変更ができます。指定値とモードの対応は、

@@NZG-0 : 長周期モード

@@NZG-1 : 短周期モード

となります。

GB ノイズは、パルス波のパルス幅が擬似ランダムに変化する「2値ノイズ」の一種で、パルス幅更新間隔と計算手順により固有の風合いを持っています。ただし、更新間隔が非常に短い2値ノイズは再生サンプリング周波数を超えた密度になるため、過密部分を相加平均処理しています。そのため、相加平均部分がホワイトノイズのように出力されます。

GB ノイズ音源では、パルス幅の更新間隔を @N コマンド で変更することで、ノイズの質感を変えることができます。

【Yコントロールにおける、音源モジュール設定】

当音源宛の Y コントロールは、Y[F],[P],[D] のように書きます。

各引数は次の通りです。

[F] : 機能番号 (0以上の整数)

[P] : パラメータ (浮動小数点数。少数以下切り捨てで整数利用も有り。)

[D] : 宛先となる音源モジュールの識別子

例えば、Y5,1,NZG のように書きます。

Y コントロールの記述の2回目以降は宛先「,NZG」は省略できます。

以降の説明では宛先省略で記述します。

Y1,[P] :

音源サブモジュール番号 (周期モード) を [P] に設定します。

指定値と周期モードの対応は次の通りです。

0 のとき、長周期モード。

1 のとき、短周期モード。

Y5,[P] :

GB ノイズのパルス幅更新間隔を設定します。

このYコントロールは、@Nコマンド から呼び出されます。

パラメータの仕様は、宛先NZGの @Nコマンド と同様です。

6.13. @@NZC : 波形メモリノイズ音源

記述	@@NZC-[1]
種別	MML コマンド

【解説】

使用する音源モジュールを、波形メモリノイズ音源にします。

引数[1]は、枝番指定で行う音源サブモジュール番号指定です。

枝番指定は省略可能です。省略した場合は 0 を指定したものとみなされます。

@@NZC と @@NZC-0 は全く同じ指定内容になります。

波形メモリノイズ音源は、後述するメタデータ定義「#ML:WAVEM」によって、波形データに波形番号を割り当てて登録したものを利用したノイズ音源です。波形データの選択は、音源サブモジュール番号 (@番号) によって、波形番号を指定して行います。

波形メモリノイズで使用する波形データは、定義されている先頭 32 サンプルのみ使用されます。32 サンプルに満たない波形定義の場合は、残りの部分が変位 0 (中心点) で埋められて使用されます。

波形メモリノイズでは、線形帰還シフトレジスタの演算を利用して、擬似ランダムに波形サンプルを取り出して変位に採用します。

波形メモリノイズ音源では、変位を更新する間隔を @N コマンド で変更することで、ノイズの質感を変えることができます。

【Yコントロールにおける、音源モジュール設定】

当音源宛の Yコントロールは、Y[F],[P],[D] のように書きます。

各引数は次の通りです。

[F] : 機能番号 (0以上の整数)

[P] : パラメータ (浮動小数点数。少数以下切り捨てで整数利用も有り。)

[D] : 宛先となる音源モジュールの識別子

例えば、Y5,1,NZC のように書きます。

Yコントロールの記述の 2 回目以降は宛先「,NZC」は省略できます。

以降の説明では宛先省略で記述します。

Y1,[P] :

音源サブモジュール番号 (波形番号) を [P] に設定します。

使用する波形番号を設定します。

Y5,[P] :

波形メモリノイズの変位の更新間隔を設定します。
このYコントロールは、@Nコマンド から呼び出されます。
パラメータの仕様は、宛先NZCの @Nコマンド と同様です。

Y8,[P] :

シフトレジスタ値の直接書き換えを行います。
このYコントロールは、@NRコマンド から呼び出されます。
パラメータの仕様は、宛先NZCの @NRコマンド と同様です。

Y10,[P] :

波形メモリノイズのステップレート倍率を指定します。
このYコントロールは、宛先NZCの @Nコマンド による設定に影響を与えます。
パラメータの仕様は、「#ML:NZCSTEPM」による設定と同様ですが、このYコントロールを使った場合の効果は、使用トラックにのみ影響します。

6.14. @@DPC : DPCM音源

記述	@@DPC-[1]
種別	MMLコマンド

【解説】

使用する音源モジュールを、DPCM音源にします。
 DPCMのデータ形式は、FC内蔵音源のDPCM互換です。
 引数[1]は、枝番指定で行う音源サブモジュール番号指定です。
 枝番指定は省略可能です。省略した場合は0を指定したものとみなされます。
 @@DPC と @@DPC-0 は全く同じ指定内容になります。

発音は、o0c ~ o1d+ の16通りの音程指定で行います。
 o0c が一番低いDPCM再生周波数での発音になります。
 音程を上げると16段階までのピッチ変更が出来ます。

DPCM音源では、後述するメタデータ定義「#ML:DPCM」によって、
 DPCM波形データに波形番号を割り当てて登録したものを使用します。
 波形データの選択は、音源サブモジュール番号(@番号)によって、波形番号を
 指定して行います。

【Yコントロールにおける、音源モジュール設定】

当音源宛の Yコントロールは、Y[F],[P],[D] のように書きます。
 各引数は次の通りです。
 [F]：機能番号（0以上の整数）
 [P]：パラメータ（浮動小数点数。少数以下切り捨てで整数利用も有り。）
 [D]：宛先となる音源モジュールの識別子
 例えば、Y1,0,DPC のように書きます。
 Yコントロールの記述の2回目以降は宛先「,DPC」は省略できます。
 以降の説明では宛先省略で記述します。

Y1,[P]：
 音源サブモジュール番号（波形番号）を [P] に設定します。

6.15. @@PCM : P C M音源

記述	@@PCM-[1]
種別	MML コマンド

【解説】

使用する音源モジュールを、P C M音源にします。
 P C Mのデータ形式は、Unsigned 8bit PCM です。
 引数[1]は、枝番指定で行う音源サブモジュール番号指定です。
 枝番指定は省略可能です。省略した場合は 0 を指定したものとみなされます。
 @@PCM と @@PCM-0 は全く同じ指定内容になります。

P C M音源では、後述するメタデータ定義「#ML:U8PCM」によって、
 P C M波形データに波形番号を割り当てて登録したものを使用します。
 波形データの選択は、音源サブモジュール番号 (@番号) によって、波形番号を
 指定して行います。

【Yコントロールにおける、音源モジュール設定】

当音源宛の Yコントロールは、Y[F],[P],[D] のように書きます。
 各引数は次の通りです。
 [F] : 機能番号 (0以上の整数)
 [P] : パラメータ (浮動小数点数。少数以下切り捨てで整数利用も有り。)
 [D] : 宛先となる音源モジュールの識別子
 例えば、Y5,0,PCM のように書きます。
 Yコントロールの記述の2回目以降は宛先「,PCM」は省略できます。
 以降の説明では宛先省略で記述します。

Y1,[P] :
 音源サブモジュール番号 (波形番号) を [P] に設定します。

Y2,[P] :
 補完モードの選択を行います。
 [P] が 0 のとき、再生を0次補完モードで行います。
 [P] が 1 のとき、再生を1次補完モードで行います。
 デフォルトは1です。
 このYコントロールは、@0コマンド から呼び出されます。
 パラメータの仕様は、宛先PCMの @0コマンド と同様です。

Y5,[P] :

波形再生終端処理の選択を行います。

[P] が 0 のとき、ループしない再生の終端は、最終変位を維持します。

[P] が 1 以上のとき、変位 0(中心) までに収束するサンプル数 (1/192000sec 単位) として扱われます。

デフォルトは 3 です。

6.16. @@ADP : ADPCM音源

記述	@@ADP-[1]
種別	MMLコマンド

【解説】

使用する音源モジュールを、ADPCM音源にします。

ADPCMのデータ形式は、4bit ADPCM です。

引数[1]は、枝番指定で行う音源サブモジュール番号指定です。

枝番指定は省略可能です。省略した場合は 0 を指定したものとみなされます。

@@ADP と @@ADP-0 は全く同じ指定内容になります。

ADPCM音源では、後述するメタデータ定義「#ML:ADPCM」によって、ADPCM波形データに波形番号を割り当てて登録したものを使用します。波形データの選択は、音源サブモジュール番号 (@番号) によって、波形番号を指定して行います。

【Yコントロールにおける、音源モジュール設定】

当音源宛の Yコントロールは、Y[F],[P],[D] のように書きます。

各引数は次の通りです。

[F] : 機能番号 (0以上の整数)

[P] : パラメータ (浮動小数点数。少数以下切り捨てで整数利用も有り。)

[D] : 宛先となる音源モジュールの識別子

例えば、Y5,0,ADP のように書きます。

Yコントロールの記述の2回目以降は宛先「,ADP」は省略できます。

以降の説明では宛先省略で記述します。

Y1,[P] :

音源サブモジュール番号 (波形番号) を [P] に設定します。

Y2,[P] :

補完モードの選択を行います。

[P] が 0 のとき、再生を0次補完モードで行います。

[P] が 1 のとき、再生を1次補完モードで行います。

デフォルトは1です。

このYコントロールは、@0コマンド から呼び出されます。

パラメータの仕様は、宛先ADPの @0コマンド と同様です。

Y5,[P] :

波形再生終端処理の選択を行います。

[P] が 0 のとき、ループしない再生の終端は、最終変位を維持します。

[P] が 1 以上のとき、変位 0(中心) までに収束するサンプル数 (1/192000sec 単位) として扱われます。

デフォルトは 3 です。

6.17. @MH : @@FMS用 : OPM互換HLFO

記述 @MH[1],[2],[3],[4],[5],[6],[7]

種別 MMLコマンド

【解説】

F M音源モジュール (@@FMS) 用の、OPM互換HLFO (hardware low frequency oscillator) 機能を指定します。
@MHコマンド と @MHAコマンド は排他制御が掛かります。

排他制御により、@MH を使うと、それまでの @MHA 指定は無効になり、
@MHA を使うと、それまでの @MH 指定は無効になります。
(両方同時にHLFOは掛かりません)

指定する引数は次の通りです。

引数[1]...波形 (wf)	[wave form]
引数[2]...周波数 (lfrq)	[lfo frequency]
引数[3]...音程深度 (pmd)	[pitch modulation depth]
引数[4]...音量深度 (amd)	[amplitude modulation depth]
引数[5]...音程感度 (pms)	[pitch modulation sensitivity]
引数[6]...音量感度 (ams)	[amplitude modulation sensitivity]
引数[7]...同期モード (sync)	[synchronization]

引数はカンマで区切って指定します。

引数[1]の wf 以外は省略可能で、省略した場合は後ろから順に省略したものとみなされます。省略時にはデフォルト値 (0) が採用されます。

トラック先頭におけるデフォルトは、当機能無効 (@MHZ指定状態) です。

引数[1] (wf)

HLFOで使用する波形を指定します。

範囲は 0~3。デフォルトは 0 です。

0: のこぎり波

1: 矩形波

2: 三角波

3: ランダム波 (ノイズ)

引数[2] (lfrq)

HLFOの周波数を指定します。

範囲は 0~255。デフォルトは 0 です。

引数[3] (pmd)

音程 L F O の深度を指定します。
範囲は 0~127。デフォルトは 0 です。

引数[4] (amd)

音量 L F O の深度を指定します。
範囲は 0~127。デフォルトは 0 です。

引数[5] (pms)

音程 L F O の感度を指定します。
範囲は 0~7。デフォルトは 0 です。

引数[6] (ams)

音量 L F O の感度を指定します。
範囲は 0~3。デフォルトは 0 です。

引数[7] (sync)

ノートオン同期モードを指定します。
範囲は 0 または 1。デフォルトは 0 です。
0 を指定すると、ノートオンで L F O 波形の位相はリセットされません。
1 を指定すると、ノートオンの都度、L F O 波形の位相が最初からスタートします。

【例】

```
@MH2,80,127,0,3,0,0
```

6.18. @MHA : @@FMS用 : OPNA 互換HLFO

記述 @MHA[1],[2],[3],[4]

種別 MML コマンド

【解説】

FM音源モジュール (@@FMS) 用の、OPNA 互換HLFO (hardware low frequency oscillator) 機能を指定します。
@MHAコマンド と @MHコマンド は排他制御が掛かります。

排他制御により、@MHA を使うと、それまでの @MH 指定は無効になり、
@MH を使うと、それまでの @MHA 指定は無効になります。
(両方同時にHLFOは掛かりません)

指定する引数は次の通りです。

引数[1]...周波数 (lfrq)	[lfo frequency]
引数[2]...音程感度 (pms)	[pitch modulation sensitivity]
引数[3]...音量感度 (ams)	[amplitude modulation sensitivity]
引数[4]...同期モード (sync)	[synchronization]

引数はカンマで区切って指定します。

引数[1]の lfrq 以外は省略可能で、省略した場合は後ろから順に省略したものとみなされます。省略時にはデフォルト値 (0) が採用されます。

トラック先頭におけるデフォルトは、当機能無効 (@MHZ指定状態) です。

引数[1] (lfrq)

HLFOの周波数を指定します。

範囲は 0~7。デフォルトは 0 です。

引数[2] (pms)

音程HLFOの感度を指定します。

範囲は 0~7。デフォルトは 0 です。

引数[3] (ams)

音量HLFOの感度を指定します。

範囲は 0~3。デフォルトは 0 です。

引数[4] (sync)

ノートオン同期モードを指定します。

範囲は 0 または 1。デフォルトは 0 です。

0 を指定すると、ノートオンでLFO波形の位相はリセットされません。
1 を指定すると、ノートオンの都度、LFO波形の位相が最初からスタートします。

【例】

```
@MHA3,4,0,0
```

6.19. @MHZ : @@FMS用 : H L F O無効

記述	@MHZ
種別	MMLコマンド

【解説】

F M音源モジュール (@@FMS) 用の、H L F O (Hardware Low Frequency Oscillator) 機能を無効に設定します。

H L F O機能が無効になると、F M音源の動作がその分軽くなります。トラック開始時のデフォルト設定では、H L F O機能は無効になっています。

H L F O機能が無効の状態では、Yコントロールを使用して直接H L F O設定を行っても動作しないので注意してください。

H L F O機能を利用する場合、@MHコマンド または @MHAコマンド を使用すると、必ずH L F O機能が有効化されますので確実です。

6.20. @MR : @@FMS用 : A R 初期レベルモード

記述	@MR[1],[2]
種別	MML コマンド

【解説】

F M音源モジュール (@@FMS) 用の、ノートオンごとのエンベロープでの、A R (アタックレート) の初期レベルについて、モード設定を行います。

指定する引数は次の通りです。

引数[1]...モード番号 (mode)

引数[2]...パラメータ (param)

引数はカンマで区切って指定します。

トラック先頭におけるデフォルトは、@MR0 です。

引数[1] (mode)

modeに 0 を指定すると、A R初期レベルは、音色設定内の指定に従います。

modeが 0 の場合、param は省略できます (param を指定しても無効)。

音色を OPM / OPNA / OPN モードで定義している場合は、いずれもA R初期レベルを変更できませんが、全てのオペレータで「ノートオン直前までのレベルを引き継ぐモード」に設定されています。

modeに 1 を指定すると、A R初期レベルは、音色設定内容に関わらず、全てのオペレータで強制的に param で指定したA R初期レベルになります。

modeが 1 の場合に param を省略すると、A R初期レベルは全オペレータで 126 にセットされます。

引数[2] (param)

modeが 1 の場合に、全オペレータにセットするA R初期レベルを、0 から 128 の整数で指定します。

128に近づくほど、無音からのエンベロープ開始になります。

【例】

```
@MR0 cdefg
@MR1 cdefg
@MR1,64 cdefg;
```


6.21. @MZ : @@FMS用 : カレントENV-LVの即時減衰機能

記述	@MZ[1]
種別	MMLコマンド

【解説】

F M音源モジュール (@@FMS) 用の、カレントエンベロープレベル（現在発音中のエンベロープ出力）を、全オペレータで即時減衰させます。

似た機能のエンベロープダンパー (@Zコマンド) は、リリースレートを操作する方式のため、実際出力減衰には、ダンパー実行から、わずかに休符時間を必要とします。

当コマンドでは、カレントエンベロープレベルを即時減衰させます（減衰に休符時間は不要）。

当コマンドをノートオン直前に使用すれば、ノートオン時の初期エンベロープレベル（IEL）が引き継ぎモードになっていて、なおかつ休符やゲートによる隙間がない場合でも、減衰済みのエンベロープレベルを引き継がせることができます。

引数[1]には、減衰量を 0 ~ 128 で指定します。

小数以下の指定も受け付けますが、小数以下9ビットの固定小数点数の精度に丸められて応答します。これは内部の振幅値テーブルサイズ（解像度）の都合によるものです。

減衰量は、TL のスケールと同様です。

例えば @MZ20 とした場合、カレントエンベロープレベルが TL 換算で 20 加算（出力が減衰）したのと同様の効果が即時適用されます。

【例】

```
CCC @MZ20 G @MZ20 A @MZ20 @MZ20 B;
```

このように書くと、Gは20、Aは20、Bは40の出力を減衰してノートオンします。（@MZ20を2回書くことで、20+20=40の減衰になっている点に注意）

6.22. @MZE : @@FMS用 : @MZ機能の継続適用

記述	@MZE[1]
種別	MML コマンド

【解説】

F M音源モジュール (@@FMS) 用の、カレントエンベロープレベル (現在発音中のエンベロープ出力) を、全オペレータで即時減衰させる機能 (@MZコマンド) を、ノートオンの都度、ノートオン直前に適用させます。

当コマンドを使用すれば、ノートオン時の初期エンベロープレベル (IEL) が引き継ぎモードになっていて、なおかつ休符やゲートによる隙間が全くない場合でも、ノートオンの都度、減衰済みのエンベロープレベルを引き継がせることができます。

引数[1]には、減衰量を 0 ~ 128 で指定します。

小数以下の指定も受け付けますが、小数以下9ビットの固定小数点数の精度に丸められて応答します。これは内部の振幅値テーブルサイズ (解像度) の都合によるものです。

減衰量は、TL のスケールと同様です。

例えば @MZE20 とした場合、カレントエンベロープレベルが TL 換算で 20 加算 (出力が減衰) したのと同様の効果が、ノートオンの都度適用されます。

【例】

```
@MZE20 CCCGGG;
```

このように書くと、CCCGGGの音符がいずれも、ノートオン直前に20、出力を減衰したレベルを引き継いでノートオンします。

6.23. @PH：ノートオン時の位相リセット

記述	@PH[1],[2],[3]
種別	MML コマンド

【解説】

ノートオン時に、位相リセットをサポートする音源モジュールの位相を、リセットするかどうかの指定を行います。

指定する引数は次の通りです。

引数[1]...モード番号 (mode)

引数[2]...位相値 (phase)

引数[3]...宛先の音源モジュール (module)

引数はカンマで区切って指定します。

@PHコマンドを受け付けた直後のノートオンは、mode に関わらず、必ず位相が phase の内容でリセットされます。

引数[1] (mode)

ノートオン時の、位相リセットモードを整数で指定します。

省略した場合は、-1 を指定したものとみなされます。

mode が 0 のとき

ノートオンの都度、位相をリセットしません。

mode が 1 のとき

ノートオンの都度、位相をリセットします。

mode が 2 のとき

今回ノートオンが直前ノートオフと同時刻の場合（休符やゲートでの間隔が無い場合）、位相をリセットしません。

今回ノートオンが直前ノートオフと同時刻でない場合（休符やゲートでの間隔がある場合）、位相をリセットします。

mode が 3 のとき

無音状態（エンベロープが完全に終了した状態）からのノートオンの場合、位相をリセットします。

無音状態でない場合でのノートオンの場合、位相をリセットしません。

mode が 4 のとき

ノートオンの都度、位相をリセットしませんが、「@番号」による音源サブモジュール指定コマンドを使用した直後のノートオンのみ、位相をリセットします。

mode に 0 以上の値を指定した場合で、上記 mode 以外の値を指定した場合、mode に 1 が指定されたものとみなされます。

mode が -1 のとき

既に設定済みの mode と phase を変更せずに、直後のノートオンのみ対象に、強制的に、既に設定済みの phase に位相をリセットします。

mode 指定が -1 のときは、phase の指定は不要で、指定したとしても無視されません。

-1 以外の負数の指定の場合は、-1 が指定されたものとみなされます。

引数[2] (phase)

mode に従ったタイミングで位相リセットする際の、位相値を設定します。

書式 1 : 0 以上 1 未満の浮動小数点数

ノートオン時にリセットする際の位相を、0 以上 1 未満の浮動小数点数で指定します。省略した場合は、0 を指定したものとみなされます。

(0~1は、0~2*pi に対応)

書式 2 : 負数 (-1)

phase に -1 と指定すると、ノートオン時の位相をランダムな位置にリセットします。

書式 3 : 「#」付きの整数

phase に #番号 と指定すると、ノートオン時の位相をランダム値パレットの内容の位置に位相リセットされます。

ランダム値パレットとは、MML コンパイル開始時に毎回 1 回初期化される、0~1 の浮動少数点数のランダム値を格納する配列で、0 番~63 番の 64 個が用意されています。例えば、

```
@PH0,#0
```

と指定すると、直後のノートのみ、0 番のランダム値パレットで位相リセットします（この場合の宛先はカレント音源モジュール）。

これは、複数トラックで開始位相のランダム値を一致させたい場合の利用を想定しています。

引数[3] (module)

位相リセット設定の宛先となる音源モジュール名を、3文字の識別子で指定します。有効な宛先は、

- SIN (サイン波音源)
- TRI (三角波音源)
- SAW (ノコギリ波音源)
- PLS (パルス波音源)
- WVM (波形メモリ音源)

のいずれかです（大文字、小文字は問われません）。

宛先の指定は、省略することもできます。

宛先を省略した場合は、当コマンド指定時に使用中の音源モジュールが宛先に指定されたものとみなされます。

【備考1】

音源モジュールごとに、デフォルト設定されている内容は次の通りです。

@@SIN	@PH1,0	@@PLS	@PH0,-1
@@TRI	@PH1,0	@@WVM	@PH1,0
@@SAW	@PH1,0		

【備考2】

F M音源モジュールにおけるノートオン時の位相リセットは、F M音源の音色データ定義や、Yコントロールで、オペレータごとに個別に行うよう変更になりました。

【例】

```
@WVM @0 @PH0,0 CDE;
```

この場合、最初の C のノートオンでは位相0スタート、以降のノートオンでは位相リセットされません。

6.24. @0 : オペレートモード

記述	@0[1],[2]
種別	MML コマンド

【解説】

特定の音源モジュールに対し、オペレートモードの指定を行います。

指定する引数は次の通りです。

引数[1]...モード番号 (mode)

引数[2]...宛先の音源モジュール (module)

引数はカンマで区切って指定します。

引数[1] (mode)

音源モジュールに対するオペレートモード番号を整数で指定します。

(宛先別の、mode番号に対応する機能については後述)

引数[2] (module)

オペレートモード設定の宛先となる音源モジュール名を、

3文字の識別子で指定します。有効な宛先は、

WVM (波形メモリ音源)

FMS (FM音源)

PCM (PCM音源)

ADP (ADPCM音源)

のいずれかです。(識別子は大文字・小文字を問わず)

宛先の指定は、省略することもできます。

宛先を省略した場合は、当コマンド指定時に使用中の音源モジュールが宛先に指定されたものとみなされます。

無効な宛先に対して @0コマンド を使用した場合、エラーになります。

・ WVM宛のmode番号に対応する機能

宛先音源モジュールが WVM（波形メモリ音源）のときの、mode番号に対応する機能は次の通りです。

mode が 0 のとき：

1オペレータモード。1声(OP1)で発声します。

mode が 20 のとき：

2オペレータモード。2声(OP1-OP2)で発声します。

mode が 30 のとき：

3オペレータモード。3声(OP1-OP3)で発声します。

mode が 40 のとき：

4オペレータモード。4声(OP1-OP4)で発声します。

mode が 50 のとき：

5オペレータモード。5声(OP1-OP5)で発声します。

mode が 60 のとき：

6オペレータモード。6声(OP1-OP6)で発声します。

mode が 70 のとき：

7オペレータモード。7声(OP1-OP7)で発声します。

【例】

```
@@WVM @1 @030,WVM CDEFG;
```

この場合、波形メモリ音源は、3オペレータモードで発声します。

・ FMS宛のmode番号に対応する機能

宛先音源モジュールが FMS（FM音源）のときの、mode番号に対応する機能は次の通りです。

mode が 0 のとき：

波形生成モードをハイレゾモード（192kHz）にします。

mode が 1 のとき：

波形生成モードをネイティブモードにします。

生成レートは「#ML:FMDLOCK」の設定に従い、

clock ÷ prescale のレートになります。

【例】

```
@@FMS @1 @01,FMS CDEFG;
```

この場合、FM音源は、ネイティブモードで発声します。

・ PCM宛のmode番号に対応する機能

宛先音源モジュールが PCM（PCM音源）のときの、mode番号に対応する機能は次の通りです。

mode が 0 のとき：

波形演奏を 0 次補完モードで行います。

mode が 1 のとき：

波形演奏を 1 次補完モードで行います。

【例】

```
@@PCM @1 @00,PCM CDEFG;
```

・ ADP宛のmode番号に対応する機能

宛先音源モジュールが ADP（ADPCM音源）のときの、mode番号に対応する機能は次の通りです。

mode が 0 のとき：

波形演奏を 0 次補完モードで行います。

mode が 1 のとき：

波形演奏を 1 次補完モードで行います。

【例】

```
@@ADP @1 @00,ADP CDEFG;
```

【音源モジュール別 @0 デフォルト】

@@WVM	@00	@@PCM	@01
@@FMS	@00	@@ADP	@01

6.25. @W : @@PLS用 : デューティ比

記述	@W[1],[2]
種別	MML コマンド

【解説】

パルス波音源 (@@PLS) に対して、デューティ比 (パルス幅) を指定します。

指定する引数は次の通りです。

引数[1]...デューティ比の分子 (num)

引数[2]...デューティ比の分母 (denom)

引数はカンマで区切って指定します。カンマとdenomの組は省略できます。

トラック先頭におけるデフォルトは、@W4,8 (デューティ比50%) です。

引数[1] (num)

デューティ比の分子を指定します。

少数以下の指定も受け付けます。

引数[2] (denom)

デューティ比の分母を1以上の数値で指定します。

少数以下の指定も受け付けます。

denom は省略可能で、省略した場合はそれまでに設定された値が採用されます。

当コマンドの設定範囲は、分子÷分母が、0.005 以上 0.995 以下の範囲です。

範囲外の場合は、デューティ比設定そのものが無視されます。

一度 denom 付きで設定したら、以降は denom を省略する使い方を想定していません。分母が8で構わない場合は、トラック先頭から denom を省略してください。

【例】

```
@@PLS @W3,8 CDEFG
```

```
@W1 GFEDC;
```

この場合、デューティ比 37.5% で CDEFG を演奏してから、次にデューティ比 12.5% で GFEDC を演奏します。

6.26. @N：ノイズ音源用のサイクル

記述 @N[1],[2]

種別 MML コマンド

【解説】

ノイズ音源モジュールに、ノイズサイクルを指定します。

指定する引数は次の通りです。

引数[1]...サイクルパラメータ (cycle)

引数[2]...宛先のノイズ音源モジュール (module)

引数はカンマで区切って指定します。

引数[1] (cycle)

ノイズサイクルを数値で指定します。

宛先のノイズ音源モジュールによって、指定値の有効範囲が変わります。

@@NZW が宛先の場合

cycleは 1~192000000 で指定します。小数以下の指定も受け付けます。

指定値は何サンプルごとに変位をリフレッシュするかを決定します。

(備考：最大値 = $192000 \times 1000 = 1000$ 秒)

例えば、1.5 と指定した場合には、1, 2, 1, 2... という具合のサンプル数でリフレッシュされます。

@@NZP が宛先の場合

cycleは 1~2048 で指定します。小数以下の指定も受け付けます。

指定値は、プリスケール後の分周比として使用されます。プリスケール設定は @NC コマンドによる設定で決まります。

指定値 1 ~ 31 が PSG 互換です。指定値 1 未満は 1 とみなされます。

指定値 31 を超える設定値、および少数を含む設定値は、1から31の整数における計算の延長線上での値で動作します。

指定値が大きくなるほど、低いサイクル（リフレッシュサイクルが長い）になります。

SN76489の2値ノイズは、サイクルが基本的に3択ですが、4 MHz のSN76489と X1 モードの PSG 同士であれば、ノイズサイクルを 16、32、64 とすると、とてもよく似た音になります。

@@NZF が宛先の場合

cycleは 0~15 で指定します。少数以下は切り捨てられます。

指定値は、あらかじめ決められた 16 通りのサイクルの呼び出し番号に使用されます。値が大きくなるほど、低いサイクルになります。

@@NZG が宛先の場合

cycle は 0~157 で指定します。小数以下は切り捨てられます。

GB ノイズ音源では、サイクルのパラメータを、

0~7 のレートと、0~15 のシフト数の、

2つのパラメータで決定します。

10進数で、一の位の桁をレート、十から百の位の桁をシフト数に見立てて指定します。

一の位の、8~9 は、7 としてみなされます。

例えば、指定値 105 は、レート 5、シフト数 10 です。

傾向として、指定値が大きくなるほど、おおむね低いサイクルのノイズになります。

@@NZC が宛先の場合

cycle は 1~1024 で指定します。小数以下の指定も受け付けます。

指定値は変位をリフレッシュするタイミングを決定します。

(リフレッシュされるサンプル数 = $1024 \div$ 指定値)

V 2 MML のサンプリング周波数は 192kHz のため、

1 サンプルあたりの時間は 1/192000 秒で計算します。

どの宛先であっても、cycle 指定値は、

最小値を下回る指定は最小値、最大値を超えた指定は最大値に制限されます。

引数[2] (module)

宛先になる音源モジュール名を 3 文字の識別子で指定します。

トラック開始時のデフォルトは NZP です。

省略した場合、最後に指定したモジュール名が指定されたとみなされます。

有効な音源モジュールの識別子は、次のノイズ系音源のみになります。

NZW	ホワイトノイズ	NZG	GB ノイズ
NZP	PSG ノイズ	NZC	波形メモリノイズ
NZF	FC ノイズ		

(識別子は 大文字・小文字を問わず)

【例】

```
@@PLS-1 @N15,NZP 07 CCC  
@@NZW @N10,NZW CCC  
@N60 CCC;
```

ノイズ混じりパルス波のノイズサイクルに 15 を指定して演奏し、次にホワイトノイズに切り替えて、ノイズサイクル 10 を設定し演奏、その後ノイズサイクルに 60 を設定して演奏しています。

6.27. @NR：ノイズ音源用のシフトレジスタ

記述	@NR[1],[2]
種別	MML コマンド

【解説】

シフトレジスタを使用するタイプのノイズ音源モジュールに対し、シフトレジスタの値を設定します。

通常はこのコマンドを使用する必要はありません。

利用シーンは、例えば、2トラックを使用してそれぞれレガシーパンポットで左右に振り、異なるエンベロープをつけて音量推移は別々にしたいけれど、ノイズパターンは左右一致させたい場合、などです。

指定する引数は次の通りです。

引数[1]...シフトレジスタパラメータ (register)

引数[2]...宛先のノイズ音源モジュール (module)

引数はカンマで区切って指定します。

引数[1] (register)

シフトレジスタの値を指定します。

宛先のノイズ音源モジュールによって、指定値の有効範囲が変わります。

@@NZP が宛先の場合

次の3通りの指定方法があります。

- (1) 0より大きく、1未満の浮動小数点数
- (2) 1以上、131071未満の整数
- (3) #付きの64種類の整数、#0 ~ #63

PSGノイズのシフトレジスタは17bitになっています。

(1) の場合、指定値に131071を乗算し、結果が1未満なら1、131070より大きければ131070に制限してシフトレジスタに設定されます。

(2) の場合、値がそのままシフトレジスタに設定されます。

(3) の場合、指定番号に応じたランダム値パレットを呼び出し、131071を乗算し、結果を1~131070に制限してシフトレジスタに設定されます。

ランダム値パレットとは、MMLコンパイル開始時に毎回1回初期化される、0~1の浮動少数点数のランダム値を格納する配列で、0番~63番の64個が用意されています。

@@NZF が宛先の場合

次の4通りの指定方法があります。

- (1) 0より大きく、1未満の浮動小数点数
- (2) 1以上、32767未満の整数（ロングモード時）
- (3) 1以上、15未満の整数（ショートモード時）
- (4) #付きの64種類の整数、#0 ~ #63

FCノイズのシフトレジスタは15bitになっています。

(4)の場合、指定番号に応じたランダム値パレットを元にシフトレジスタに設定されます。

@@NZC が宛先の場合

次の3通りの指定方法があります。

- (1) 0より大きく、1未満の浮動小数点数
- (2) 1以上、65535未満の整数
- (3) #付きの64種類の整数、#0 ~ #63

波形メモリノイズのシフトレジスタは16bitになっています。

(1)の場合、指定値に65535を乗算し、結果が1未満なら1、65534より大きければ65534に制限してシフトレジスタに設定されます。

(2)の場合、値がそのままシフトレジスタに設定されます。

(3)の場合、指定番号に応じたランダム値パレットを呼び出し、65535を乗算し、結果を1~65534に制限してシフトレジスタに設定されます。

引数[2] (module)

宛先になる音源モジュール名を3文字の識別子で指定します。

トラック開始時のデフォルトはNZPです。

省略した場合、最後に指定したモジュール名が指定されたとみなされます。

有効な音源モジュールの識別子は、次のノイズ系音源のみになります。

NZP	PSGノイズ	NZC	波形メモリノイズ
NZF	FCノイズ		

(識別子は大文字・小文字を問わず)

【例】

```
@NR#3,NZC AAAAAA  
@NR3423,NZC AAAAAA  
@NR0.234,NZC AAAAAA;
```

6.28. @NC : @@NZP用 : ノイズサイクル設定

記述	@NC[1]
種別	MML コマンド

【解説】

PSGノイズ音源モジュール (@@NZP) に対して、サイクルモードを設定します。
トラック先頭におけるデフォルトは、@NC0 です。

引数[1]には、0 ~ 3 の整数を指定します。

対応するモードは次の通りです。

0 : PC88モード(デフォルト)

1 : X1モード

2 : MSXモード

3 : FM7モード

6.29. Y : Y コントロール

記述	Y[1],[2],[3]
種別	MML コマンド

【解説】

各音源モジュールに対し直接、制御コマンドを発行します。

指定する引数は次の通りです。

引数[1]...機能番号 (func)

引数[2]...パラメータ (param)

引数[3]...宛先の音源モジュール (module)

引数はカンマで区切って指定します。

引数[1] (func)

各音源モジュール内でサポートしている機能番号を指定します。

各音源共通で予約されているコマンド番号は次の通りです。

0 : No Operation (NOP。何もしません)

1 : 音源サブモジュール番号指定

2 : 波形レンダリングモード指定 (一部モジュールのみ)

3 : コーラスモード用デチューン指定 (一部モジュールのみ)

4 : マルチオペレータ向け音量比指定 (一部モジュールのみ)

音源別の独自コマンド番号は、各音源別の説明項目にて記述します。

機能番号に接頭辞「0x」をつけて記述すると、符号なし16進数整数による指定として解釈されます。

引数[2] (param)

func に対応したパラメータを指定します。

小数以下の指定も受け付けますが、機能番号によっては、整数受付のものは少数以下を切り捨てて受付されます。

接頭辞「0x」をつけて記述すると、符号なし16進数整数による指定として解釈されます。

この引数では加減算記述も可能です。

引数[3] (module)

Yコントロールを行う宛先となる音源モジュールを、
3文字の識別子で指定します。

宛先の指定は、省略することもできます。

宛先を省略した場合は、前回指定した宛先と同様とみなされます。

トラック先頭時における module のデフォルト宛先は PLS になります。

宛先となる識別子は次の通りです。

【音源モジュール宛の識別子】

SIN	サイン波	NZP	PSGノイズ
SAW	ノコギリ波	NZF	FCノイズ
TRI	三角波	NZG	GBノイズ
PLS	パルス波	NZC	波形メモリノイズ
WVM	波形メモリ	DPC	DPCM
FMS	FM音源	PCM	PCM
NZW	ホワイトノイズ	ADP	ADPCM

【LFO波形モジュール宛の識別子】

波形名称 \ LFO宛先	LF0@P	LF0@A	LF0@B	LF0@F	LF0@Y
サイン波	LP0	LA0	LB0	LF0	LY0
リニアバンド	LP1	LA1	LB1	LF1	LY1
三角波	LP2	LA2	LB2	LF2	LY2
パルス波	LP3	LA3	LB3	LF3	LY3
ホワイトノイズ	LP4	LA4	LB4	LF4	LY4
ユーザー定義テーブル	LP5	LA5	LB5	LF5	LY5
ノンリニアバンド	LP6	非対応	LB6	LF6	LY6

(識別子は大文字・小文字を問わず)

【例】

```
@WVM @1 Y2,20,WVM Y3,10 CDE;
```

波形メモリ音源を選択し、波形番号1を指定、

Yコントロールで波形メモリを2オペレータモードにし、

波形メモリのコーラスモードをデチューン値10にして、CDEを演奏しています。

・ L F O 波形モジュールへの適用

Y コントロールの宛先は、各種 L F O の波形モジュールも対象になります。

(宛先の識別子一覧は前項参照)

現状有効な L F O 波形モジュールへの制御は次の通りです。

(1) パルス波パターンのデューティ比設定

form : 3 (LP3 / LA3 / LB3 / LF3 / LY3)

func : 5

param : 0.05~0.995 (デフォルト 0.5)

機能 :

パルス波パターンへのデューティ比を設定します。

【例】

Y5,0.25,LP3

ピッチ L F O のパルス波パターンのデューティ比を、25%に指定します。

(2) ノートオフ同期のユーザー定義テーブル番号指定

form : 5 (LP5 / LA5 / LB5 / LF5 / LY5)

func : 5

param : 定義済みのユーザー定義テーブル番号 (#MB:LFOTBL@R)

機能 :

ユーザー定義テーブルで L F O シーケンスを行う場合に、ノートオフに同期して、「#MB:LFOTBL@R」にて定義済みのテーブルに切り替えます (ノートオン同期のテーブルとノートオフ同期のテーブルが交互に切り替わるようになります)。ここで指定するテーブル番号を -1 とすると、ノートオフ同期のテーブル切り替え機能が無効になります。

【例】

Y5,1,LA5

(3) ユーザー定義テーブルの処理モード指定

form : 5 (LP5 / LA5 / LB5 / LF5 / LY5)

func : 9

param : 0 または 1 (デフォルト 0)

機能 :

ユーザー定義テーブルで L F O シーケンスを行う場合の、テーブル処理モード (#ML:LFOTBLPM相当) を指定します。

【例】

Y9,1,LP5

6.30. @DLY：ディレイエフェクト

記述	@DLY[1],[2]
種別	MML コマンド

【解説】

ディレイエフェクトを設定します。簡易的にエコー効果が得られます。

指定する引数は次の通りです。

引数[1]...遅延させる時間 (time)

引数[2]...減衰させるレベル (level)

引数はカンマで区切って指定します。

トラック先頭におけるデフォルトは、当機能無効です。

設定する time や level によっては、音が重なりすぎて音割れを起こす場合がありますので、ご注意ください。

引数[1] (time)

遅延させる時間を設定します。

timeを数値のみで指定した場合

遅延時間の単位はミリ秒になります。小数以下の指定も受け付けます。

timeを「#」付きの数値で指定した場合

遅延時間の単位はサンプル数 (1/192000sec) になります。

サンプル数は、整数のみ受け付けます。

数値のみの指定であっても、time はサンプル数で内部管理されます。

(a)time の設定範囲は、サンプル数換算で 4~384000(2sec) です。

(b)time に 0 を指定した場合、もしくは time 指定値を省略した場合、ディレイエフェクトが無効になります。

(a)または(b)以外の time 指定値はエラーとなります。

引数[2] (level)

遅延させた音の1次振幅を、元に比べて何デシベルにするかを設定します。

設定範囲は、-0.2 ~ -96.0 です。小数以下の指定も受け付けます。

省略した場合、-6dBとみなされます。

【例 1】

```
@DLY133.33,-6 CDEFG;
```

ディレイエフェクトが、133.33msのディレイ、-6dB のレベルで掛かります。

【例 2】

```
@DLY#32000,-6 CDEFG;
```

ディレイエフェクトが、 $32000/192000$ s (1/6秒) のディレイ、-6dB のレベルで掛かります。

6.31. @F：フィルタ

記述	@F[1]
種別	MML コマンド

【解説】

事前に定義されたフィルタ情報を呼び出して適用します。
トラック先頭におけるデフォルトは、フィルタ機能停止 (@FZ) です。

引数[1]には、フィルタ番号を 0 以上の整数で指定します。

フィルタでは、後述するメタデータ定義「#ML:FILTER」によって、
定義内容にフィルタ番号を割り当てて登録したものを使用します。

【例】

```
//          num sw amt  freq reso  
#ML:FILTER  1, 2,  0, 5000,  0  
@@SAW @F1 o4 CDEFGAB;
```

ノコギリ波音源に約 5 kHz のカットオフで L P F をかけます。

6.32. @FZ : フィルタ停止

記述	@FZ
種別	MML コマンド

【解説】

動作中のフィルタ指定を、当コマンドで機能停止できます。

【例】

```
@F3 CDEFG @FZ GFEDC;
```

前半のCDEFGにはフィルタ番号3番のフィルタが掛かりますが、後半のGFEDCにはフィルタが掛かりません。

6.33. @~ : フォルマントフィルタ

記述 @~[1]

種別 MML コマンド

【解説】

フォルマントフィルタ機能を指定します。
人の声を持つ特徴的な倍音を強調することで声のように聞かせます。
トラック先頭におけるデフォルト設定は、@~Z です。

引数[1]には、フォルマントフィルタの機能種別を、アルファベット1文字 (A,I,U,E,O,Z) で指定します。機能種別と内容の対応は次の通りです。

記述	強調する特徴
@~A	「あ」
@~I	「い」
@~U	「う」
@~E	「え」
@~O	「お」
@~Z	フォルマントフィルタ機能停止

6.34. #ML:FILTER : フィルタデータ定義

記述 #ML:FILTER [1],[2],[3],[4],[5]

種別 メタデータ定義

【解説】

フィルタデータをメタデータ定義します。

指定する引数は次の通りです。

引数[1]...フィルタ番号 (number)

引数[2]...フィルタの種類 (switch)

引数[3]...フィルタエンベロープ影響度 (amount)

引数[4]...カットオフ周波数 (frequency)

引数[5]...レゾナンス (resonance)

引数はカンマで区切って指定します。

引数[1] (number)

フィルタ番号を指定します。

指定可能範囲は 0 ~ 1023 です。

この番号は、@Fコマンドのフィルタ番号指定で使用します。

引数[2] (switch)

フィルタの種類を数値で指定します。

指定値と種類の対応は次の5種類です。

0 のとき、フィルタ機能無効

1 のとき、荒いLPF

2 のとき、高品質LPF

-1 のとき、荒いHPF

-2 のとき、高品質HPF

引数[3] (amount)

フィルタエンベロープ (@EF) の影響度を指定します。

設定範囲は -100 ~ 100 です。

小数以下の指定も受け付けます。

amount は、カットオフの内部管理値 (0 ~ 1) に対し、@EFエンベロープ結果値 (0 ~ 1) を、amount/100 倍して加算します。

フィルタに @EFエンベロープ変化をつける必要が無い場合には、amount に 0 を指定してください。

引数[4] (frequency)

基準になる、およそのカットオフ周波数を指定します。
設定範囲は 40 ~ 96000 です。小数以下の指定も受け付けます。

引数[5] (resonance)

レゾナンス（カットオフ周波数付近の強調）の量を指定します。
設定範囲は 0 ~ 100 です。小数以下の指定も受け付けます。
100 のときは発振します。

【例】

```
//          num sw amt  freq reso  
#ML:FILTER  1, 2,  0, 5000,  0  
@@SAW @F1 o4 CDEFGAB;
```

ノコギリ波音源に約 5 kHz のカットオフで L P F をかけます。

6.35. #ML:NZCSTEPM : @@NZC用 : 更新ステップ倍率設定

記述	#ML:NZCSTEPM [1]
種別	メタデータ指定

【解説】

波形メモリノイズ音源 (@@NZC) の、変位更新ステップ (ノイズサイクル) を、デフォルトの何倍にするかを設定します。

当メタデータ指定は、波形メモリノイズ音源への @Nコマンド によるノイズサイクル指定の結果に影響を与えます。

引数[1]には、変位更新ステップ倍率を指定します。
設定範囲は 0.125 ~ 8.0 で、浮動小数点数で指定します。
当メタデータ指定を使用しなかった場合のデフォルトは 1.0 です。

【例】

```
#ML:NZCSTEPM 0.5
```

波形メモリノイズ音源の、変位更新ステップ倍率が、通常 (1.0) の場合と比べて、同じノイズサイクル指定 (@Nコマンド) でも、半分 (0.5) のノイズサイクルになります。

6.36. #ML:OVSWAVEM : @@WVM用 : オーバーサンプリング 設定

記述	#ML:OVSWAVEM [1]
種別	メタデータ指定

【解説】

波形メモリ音源 (@@WVM) では、エイリアシングノイズ対策のため、簡易的なオーバーサンプリング処理を行っています。

当メタデータ指定により、全トラックの波形メモリ音源の初期設定を、何倍オーバーサンプリングにするか設定できます。

波形メモリ音源のオーバーサンプリング設定は、MML 記述でトラックごとに変更することもできます。(Yコントロールコマンド使用)

引数[1]には、オーバーサンプリング倍率を指定します。

設定範囲は 1 ~ 4 で、整数で指定します。

当メタデータ指定を使用しなかった場合のデフォルトは 2 です。

【備考】

多くのトラック定義などにより、デフォルトであっても波形メモリ音源演奏で処理が重い場合は、倍率を下げてみてください。

【例】

```
#ML:OVSWAVEM 4
```

波形メモリ音源の、全トラックのデフォルトのオーバーサンプリング倍率を 4 に設定します。

6.37. #ML:WAVEM : @@WVM用 : 波形データ定義

記述	#ML:WAVEM [1],[2],[3],[4]
種別	メタデータ定義

【解説】

波形メモリ音源 (@@WVM) の波形データを定義します。
定義する波形は「1 周期分の」波形データになります。

指定する引数は次の通りです。

引数[1]... 波形番号 (waveNo)

引数[2]... 1 サンプルあたりの文字数 (useChar)

引数[3]... 1 サンプルあたりのビット数 (bitWidth)

引数[4]... 1 周期分の波形データ (WAVEDATA)

引数はカンマで区切って指定します。

引数[1] (waveNo)

波形番号を整数で指定します。

指定可能範囲は 0 ~ 1023 です。

この番号は、@@WVM のサブモジュール番号 (@番号) で使用します。

引数[2] (useChar)

波形の 1 サンプルを表現するのに使用する文字数を、

1 または 2 で指定します。

引数[3] (bitWidth)

波形の 1 サンプルを表現するのに使用するビット幅を整数で指定します。

指定値の範囲は、

useChar が 1 のとき、1 ~ 4 で、

useChar が 2 のとき、1 ~ 8 になります。

引数[4] (WAVEDATA)

最大 2048 サンプルまでの PCM データを、1 周期分の波形として記述します。

波形定義終了は改行によって検知されます。

各サンプルは、半角の 16 進数表記で、符号無し整数で記述します。

(16 進数記述の英字は、大文字・小文字を問わず)

useChar で指定した桁ずつ、1 サンプルと解釈されますが、16 進数以外の文字の桁は「0」として認識されます。もし桁が中途半端な桁で終わる記述をした場合、最後のサンプルは無効になります。

スペース文字は読み飛ばされるので、各サンプルごとにスペースで区切って記述しても正しく認識されます。

【波形データ定義例】

```
#ML:WAVEM 0,2,8,ff00000000000000
#ML:WAVEM 1,2,8,ffff ffff ffff 8080 0000
#ML:WAVEM 3,2,5,1f1f1f1f 00000000
#ML:WAVEM 6,1,4,ffffffffffffffff0000000000000000
```

【備考】

変位の内部管理は $-1.0 \sim +1.0$ の浮動小数点数で行われており、これに整数値の変位を割り当てています。

変位最低の -1.0 には、最小値（ゼロ）。

変位最高の $+1.0$ には、最大値が割り当てられます。

変位最高と変位最低の間の数値は、リニアに割り当てられます。

この割り当て方を計算式で表すと次のようになります。

$$Y = ((X / (2^{\text{bitWidth}} - 1)) * 2.0) - 1.0$$

Yは変位、Xの値域は 0 から（2のbitWidth乗 - 1）までの整数です。

例えば、useCharが 2、bitWidthが 6 のときは、

1 サンプルは16進数 00 ~ 3F で表現され、

00 に変位 -1.0 が割り当てられ、

3F に変位 $+1.0$ が割り当てられます。

この場合、変位 0.0 に相当する16進数値は 1F と 20 の中間になるので、変位 0.0 ちょうどに割り当たる値は存在しないこととなります。

6.38. #ML:FMDLOCK : @@FMS用 : マスタークロック設定

記述	#ML:FMDLOCK [1],[2]
種別	メタデータ指定

【解説】

F M音源 (@@FMS) の、デフォルトのマスタークロックを設定します。

指定する引数は次の通りです。

引数[1]...マスタークロック (clock)

引数[2]...プリスケール (prescale)

引数はカンマで区切って指定します。

当メタデータ指定を使用しなかった場合のデフォルトは次の通りです。

```
#ML:FMDLOCK 3579545,64
```

引数[1] (clock)

マスタークロックを数値で指定します。少数以下の指定も受け付けます。

OPM想定の場合、3579545 や、4000000 などになります。

OPN想定の場合、PC88互換であれば、3993600 になります。

引数[2] (prescale)

分周比を数値で指定します。マスタークロックを割る数です。

OPM想定の場合、64と指定します。

OPN想定の場合、PC88互換であれば、72と指定します。

prescaleの指定は省略可能で、省略した場合 64 が指定されたものとみなされます。

【備考1】

clock ÷ prescale の値 (生成レート) が、40000~80000 でないと、エラーとなり設定されません。

4MHzのOPMの場合、clock ÷ prescale = 62500 です。

PC88互換OPN系の場合、およそ 55466.667 です。

【備考2】

当メタデータ指定の影響範囲は、

- ・ FM音源のエンベロープシーケンス速度
- ・ OPM互換ハードウェアLFOシーケンス速度 (@MH)
- ・ OPNA互換ハードウェアLFOシーケンス速度 (@MHA)
- ・ デチューン1計算
- ・ @01,FMS と指定した場合の波形生成レート

です。

音程計算は使用中のピッチスケールに従って計算するため、当メタデータ指定には依存しません。

影響範囲にある部分を、例えば4MHzモードに合わせ込みを行いたい場合などに、当メタデータ指定を使用します。

【例】

```
#ML:FMDLOCK 3993600,72
```

PC88互換の設定にしています。

6.39. #ML:FMFBBIAS : @@FMS用 : F B バイアス設定

記述	#ML:FMFBBIAS [1]
種別	メタデータ指定

【解説】

F M音源 (@@FMS) の、全トラックにおける、フィードバックレベルに対するバイアス調整値の初期値を設定します。

引数[1]には、フィードバックレベルに対するバイアス調整値を指定します。
設定範囲は 0.75 ~ 1.35 で、浮動小数点数で指定します。

(少数以下の桁数に制限は設けていません)

このバイアス調整値は、フィードバックレベルに対する倍率です。

デフォルトは 1.0 に設定されています。

指定値が大きくなるほど音色が明るくなり、小さくなるほど曇った感じになります。

【備考】

全くバイアス調整を行わない設定は 1.0 になります。

トラックごとに変更する場合は、Y6,[1],FMS による指定で行います。

6.40. #ML:FMHLFOS1 : @@FMS用 : H L F O速度調整 1

記述	#ML:FMHLFOS1 [1]
種別	メタデータ指定

【解説】

F M音源 (@@FMS) の、全トラックにおける、O P M互換H L F Oの速度調整の初期値を設定します。@MHコマンド の周期に影響します。

引数[1]には、O P M互換H L F Oの速度倍率を指定します。

設定範囲は 0.5 ~ 2.0 で、浮動小数点数で指定します。

小数点以下の桁数は制限していません。

デフォルトでは 1.0 (等倍) に設定されています。

トラックごとに速度変更したい場合は、Y57,[1],FMS による指定で行います。

例えば、3.58MHzモードで、4MHz の OPM で作り込まれたHLF0パラメータを使用する際に、Y57,1.117,FMS などと指定することを想定しています。

($4000000 \div 3579545 = 1.117460459360058$)

6.41. #ML:FMHLFOS2 : @@FMS用 : H L F O速度調整 2

記述	#ML:FMHLFOS2 [1]
----	------------------

種別	メタデータ指定
----	---------

【解説】

F M音源 (@@FMS) の、全トラックにおける、O P N A 互換 H L F O の速度調整の初期値を設定します。@MHAコマンド の周期に影響します。

引数[1]には、O P N A 互換 H L F O の速度倍率を指定します。

設定範囲は 0.5 ~ 2.0 で、浮動小数点数で指定します。

小数点以下の桁数は制限していません。

デフォルトでは 1.0 (等倍) に設定されています。

トラックごとに速度変更したい場合は、Y64,[1],FMS による指定で行います。

6.42. #MB:FM@SS60P : @@FMS用 : 音色データ定義 (6 オペレータ)

記述 #MB:FM@SS60P [1] {[2]}

種別 メタデータ定義

【解説】

F M音源 (@@FMS) の音色データを、SS60Pモードで定義します。
このモードは、6 オペレータ、フル機能モードになります。

指定する引数は次の通りです。

引数[1]...音色番号 (num) 【1個の整数】

引数[2]...音色データ (tone) 【複数のパラメータ】

引数[1] (num)

音色番号 (num) は、音源サブモジュール番号指定 (@番号) で、音色を指定する際に使用する番号です。

設定範囲は 0 ~ 1023 です。

音色番号 0 番には初期データが入っています (上書可)。

未定義の音色番号を参照した場合には、0 番が読み出されます。

引数[2] (tone)

音色データ (tone) の開始は「 { 」で認識され、終了は「 } 」によって検知されます。

音色データの各パラメータは、カンマで区切って記述します。

最終パラメータの後のカンマは、あってもなくても構いません。

スペースや改行は読み飛ばされます。

SS60Pモードでは、103個のパラメータ群により音色データを定義します。

【音色定義例】

```
#MB:FM@SS60P 1 {
  45,
// IPH IEL   AR D1R D2R  RR D1L  TL  KS   MUL DT1 DT2 DT3 DT4   FB AME MSK
   0, -1,   31,  0,  0,  0,  0, 21,  0,   1,  0,  0,  0,  0,   5,  0,  0,
   0, -1,   18, 15,  9,  5,  3,  0,  0,   1,  0,  0,  0,  0,   0,  0,  0,
   0, -1,   31,  0,  0,  0,  0, 20,  0,   1,  0,  0,  5,  0,   0,  0,  0,
   0, -1,   29, 15,  9,  4,  3,  0,  0,   1,  0,  0,  5,  0,   0,  0,  0,
  2/4, -1,   31,  0,  0,  0,  0, 21,  0,   1,  0,  0, -5,  0,   5,  0,  0,
   0, -1,   18, 15,  9,  5,  3,  6,  0,   1,  0,  0, -5,  0,   0,  0,  0,
}
@@FMS @1 V14 04CDE;
```

[SS60P]パラメータ番号表：

	1																
OP1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
OP2	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
OP3	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52
OP4	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69
OP5	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86
OP6	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103

[SS60P]パラメータ番号に対応する機能：

	CON																
OP1	IPH	IEL	AR	D1R	D2R	RR	D1L	TL	KS	MUL	DT1	DT2	DT3	DT4	FB	AME	MSK
OP2	IPH	IEL	AR	D1R	D2R	RR	D1L	TL	KS	MUL	DT1	DT2	DT3	DT4	FB	AME	MSK
OP3	IPH	IEL	AR	D1R	D2R	RR	D1L	TL	KS	MUL	DT1	DT2	DT3	DT4	FB	AME	MSK
OP4	IPH	IEL	AR	D1R	D2R	RR	D1L	TL	KS	MUL	DT1	DT2	DT3	DT4	FB	AME	MSK
OP5	IPH	IEL	AR	D1R	D2R	RR	D1L	TL	KS	MUL	DT1	DT2	DT3	DT4	FB	AME	MSK
OP6	IPH	IEL	AR	D1R	D2R	RR	D1L	TL	KS	MUL	DT1	DT2	DT3	DT4	FB	AME	MSK

6.43. #MB:FM@SS40P : @@FMS用 : 音色データ定義 (4 オペレータ)

記述	#MB:FM@SS40P [1] {[2]}
種別	メタデータ定義

【解説】

F M音源 (@@FMS) の音色データを、SS40Pモードで定義します。
このモードは、4 オペレータ、フル機能モードになります。

指定する引数は次の通りです。

引数[1]...音色番号 (num) 【1個の整数】

引数[2]...音色データ (tone) 【複数のパラメータ】

引数[1] (num)

音色番号 (num) は、音源サブモジュール番号指定 (@番号) で、音色を指定する際に使用する番号です。

設定範囲は 0 ~ 1023 です。

音色番号 0 番には初期データが入っています (上書可)。

未定義の音色番号を参照した場合には、0 番が読み出されます。

引数[2] (tone)

音色データ (tone) の開始は「 { 」で認識され、終了は「 } 」によって検知されます。

音色データの各パラメータは、カンマで区切って記述します。

最終パラメータの後のカンマは、あってもなくても構いません。

スペースや改行は読み飛ばされます。

SS40Pモードでは、69個のパラメータ群により音色データを定義します。

【音色定義例】

```
#MB:FM@SS40P 1 {
  4,
// IPH IEL   AR D1R D2R  RR D1L  TL  KS   MUL DT1 DT2 DT3 DT4   FB AME MSK
  0, -1,   31,  0,  0,  0,  0, 21,  0,   1,  0,  0,  0,  0,   5,  0,  0,
  0, -1,   18, 15,  9,  5,  3,  0,  0,   1,  0,  0,  0,  0,   0,  0,  0,
  0, -1,   31,  0,  0,  0,  0, 20,  0,   1,  0,  0,  5,  0,   0,  0,  0,
  0, -1,   29, 15,  9,  4,  3,  0,  0,   1,  0,  0,  5,  0,   0,  0,  0,
}
@@FMS @1 V14 04CDE;
```

[SS40P]パラメータ番号表：

	1																
OP1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
OP2	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
OP3	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52
OP4	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69

[SS40P]パラメータ番号に対応する機能：

	CON																
OP1	IPH	IEL	AR	D1R	D2R	RR	D1L	TL	KS	MUL	DT1	DT2	DT3	DT4	FB	AME	MSK
OP2	IPH	IEL	AR	D1R	D2R	RR	D1L	TL	KS	MUL	DT1	DT2	DT3	DT4	FB	AME	MSK
OP3	IPH	IEL	AR	D1R	D2R	RR	D1L	TL	KS	MUL	DT1	DT2	DT3	DT4	FB	AME	MSK
OP4	IPH	IEL	AR	D1R	D2R	RR	D1L	TL	KS	MUL	DT1	DT2	DT3	DT4	FB	AME	MSK

6.44. #MB:FM@SS20P : @@FMS用 : 音色データ定義 (2 オペレータ)

記述	#MB:FM@SS20P [1] {[2]}
種別	メタデータ定義

【解説】

F M音源 (@@FMS) の音色データを、SS20Pモードで定義します。
このモードは、2 オペレータ、フル機能モードになります。

指定する引数は次の通りです。

引数[1]...音色番号 (num) 【1個の整数】

引数[2]...音色データ (tone) 【複数のパラメータ】

引数[1] (num)

音色番号 (num) は、音源サブモジュール番号指定 (@番号) で、音色を指定する際に使用する番号です。

設定範囲は 0 ~ 1023 です。

音色番号 0 番には初期データが入っています (上書可)。

未定義の音色番号を参照した場合には、0 番が読み出されます。

引数[2] (tone)

音色データ (tone) の開始は「 { 」で認識され、終了は「 } 」によって検知されます。

音色データの各パラメータは、カンマで区切って記述します。

最終パラメータの後のカンマは、あってもなくても構いません。

スペースや改行は読み飛ばされます。

SS20Pモードでは、35個のパラメータ群により音色データを定義します。

【音色定義例】

```
#MB:FM@SS20P 1 {
    0,
// IPH IEL   AR D1R D2R  RR D1L  TL  KS   MUL DT1 DT2 DT3 DT4   FB AME MSK
    0, -1,   31,  0,  0,  0,  0, 21,  0,   1,  0,  0,  0,  0,   5,  0,  0,
    0, -1,   18, 15,  9,  5,  3,  0,  0,   1,  0,  0,  0,  0,   0,  0,  0,
}
@@FMS @1 V14 04CDE;
```

[SS20P]パラメータ番号表：

	1																
OP1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
OP2	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35

[SS20P]パラメータ番号に対応する機能：

	CON																
OP1	IPH	IEL	AR	D1R	D2R	RR	D1L	TL	KS	MUL	DT1	DT2	DT3	DT4	FB	AME	MSK
OP2	IPH	IEL	AR	D1R	D2R	RR	D1L	TL	KS	MUL	DT1	DT2	DT3	DT4	FB	AME	MSK

6.45. #MB:FM@S : @@FMS用 : 旧形式音色定義 (4 オペレータ)

記述	#MB:FM@S [1] {[2]}
種別	メタデータ定義

【解説】

FM音源 (@@FMS) の音色データを、Sモードで定義します。
このモードは、4 オペレータ、旧形式のモードで、一部機能が使用出来ません。
このモードの記述は廃止予定です。

指定する引数は次の通りです。

引数[1]...音色番号 (num) 【1個の整数】

引数[2]...音色データ (tone) 【複数のパラメータ】

引数[1] (num)

音色番号 (num) は、音源サブモジュール番号指定 (@番号) で、音色を指定する際に使用する番号です。

設定範囲は 0 ~ 1023 です。

音色番号 0 番には初期データが入っています (上書可)。

未定義の音色番号を参照した場合には、0 番が読み出されます。

引数[2] (tone)

音色データ (tone) の開始は「 { 」で認識され、終了は「 } 」によって検知されます。

音色データの各パラメータは、カンマで区切って記述します。

最終パラメータの後のカンマは、あってもなくても構いません。

スペースや改行は読み飛ばされます。

Sモードでは、61個のパラメータ群により音色データを定義します。

【音色定義例】

```
#MB:FM@S 5 {
  3,
// AR D1R D2R RR D1L TL KS MUL DT1 DT2 AME MSK FB DT3 IEL
  27, 14, 0, 7, 3, 34, 0, 8, 0, 0, 0, 0, 7, 0, -1,
  31, 10, 0, 8, 6, 54, 0, 2, 0, 0, 0, 0, 0, 0, -1,
  31, 19, 0, 5, 6, 15, 0, 0, 0, 0, 0, 0, 0, 0, -1,
  31, 6, 0, 9, 14, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1,
}
@@FMS @1 V14 04CDE;
```

[S]パラメータ番号表：

	1														
OP1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
OP2	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
OP3	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46
OP4	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61

[S]パラメータ番号に対応する機能：

	CON														
OP1	AR	D1R	D2R	RR	D1L	TL	KS	MUL	DT1	DT2	AME	MSK	FB	DT3	IEL
OP2	AR	D1R	D2R	RR	D1L	TL	KS	MUL	DT1	DT2	AME	MSK	FB	DT3	IEL
OP3	AR	D1R	D2R	RR	D1L	TL	KS	MUL	DT1	DT2	AME	MSK	FB	DT3	IEL
OP4	AR	D1R	D2R	RR	D1L	TL	KS	MUL	DT1	DT2	AME	MSK	FB	DT3	IEL

6.46. #MB:FM@OPM : @@FMS用 : 音色データ定義 (OPM 互換)

記述	#MB:FM@OPM [1] {[2]}
種別	メタデータ定義

【解説】

FM音源 (@@FMS) の音色データを、OPMモードで定義します。
このモードは、4 オペレータ、OPM互換モードです。

指定する引数は次の通りです。

引数[1]...音色番号 (num) 【1個の整数】

引数[2]...音色データ (tone) 【複数のパラメータ】

引数[1] (num)

音色番号 (num) は、音源サブモジュール番号指定 (@番号) で、音色を指定する際に使用する番号です。

設定範囲は 0 ~ 1023 です。

音色番号 0 番には初期データが入っています (上書可)。

未定義の音色番号を参照した場合には、0 番が読み出されます。

引数[2] (tone)

音色データ (tone) の開始は「 { 」で認識され、終了は「 } 」によって検知されます。

音色データの各パラメータは、カンマで区切って記述します。

最終パラメータの後のカンマは、あってもなくても構いません。

スペースや改行は読み飛ばされます。

OPMモードでは、47個のパラメータ群により音色データを定義します。

【音色定義例】

```
#MB:FM@OPM 1 {
// CON FB
    2, 7,
// AR D1R D2R RR D1L TL KS MUL DT1 DT2 AME MASK
    20, 31, 0, 10, 0, 28, 0, 1, 0, 0, 0,
    31, 31, 0, 15, 0, 48, 0, 1, 0, 0, 0,
    31, 31, 0, 15, 0, 42, 0, 1, 0, 0, 0,
    25, 31, 0, 10, 0, 16, 0, 1, 0, 0, 0, 15
}
@@FMS @1 V14 04CDE;
```

[OPM]パラメータ番号表：

	1	2											
OP1	3	4	5	6	7	8	9	10	11	12	13		
OP2	14	15	16	17	18	19	20	21	22	23	24		
OP3	25	26	27	28	29	30	31	32	33	34	35		
OP4	36	37	38	39	40	41	42	43	44	45	46	47	

[OPM]パラメータ番号に対応する機能：

	CON	FB											
OP1	AR	D1R	D2R	RR	D1L	TL	KS	MUL	DT1	DT2	AME		
OP2	AR	D1R	D2R	RR	D1L	TL	KS	MUL	DT1	DT2	AME		
OP3	AR	D1R	D2R	RR	D1L	TL	KS	MUL	DT1	DT2	AME		
OP4	AR	D1R	D2R	RR	D1L	TL	KS	MUL	DT1	DT2	AME	OM	

6.47. #MB:FM@OPNA : @@FMS用 : 音色データ定義 (OPNA 互換)

記述	#MB:FM@OPNA [1] {[2]}
種別	メタデータ定義

【解説】

FM音源 (@@FMS) の音色データを、OPNAモードで定義します。
このモードは、4オペレータ、OPNA互換モードです。

指定する引数は次の通りです。

引数[1]...音色番号 (num) 【1個の整数】

引数[2]...音色データ (tone) 【複数のパラメータ】

引数[1] (num)

音色番号 (num) は、音源サブモジュール番号指定 (@番号) で、音色を指定する際に使用する番号です。

設定範囲は 0 ~ 1023 です。

音色番号 0 番には初期データが入っています (上書可)。

未定義の音色番号を参照した場合には、0 番が読み出されます。

引数[2] (tone)

音色データ (tone) の開始は「 { 」で認識され、終了は「 } 」によって検知されます。

音色データの各パラメータは、カンマで区切って記述します。

最終パラメータの後のカンマは、あってもなくても構いません。

スペースや改行は読み飛ばされます。

OPNAモードでは、43個のパラメータ群により音色データを定義します。

【音色定義例】

```
#MB:FM@OPNA 1 {
// CON FB
    0, 7,
// AR DR SR RR SL TL KS MUL DT AME MASK
    31, 18, 0, 6, 2, 38, 0, 10, 3, 1,
    31, 14, 4, 6, 2, 44, 0, 0, 3, 1,
    31, 10, 4, 6, 2, 18, 1, 0, 3, 1,
    31, 10, 3, 6, 2, 0, 1, 0, 3, 1, 15
}
@@FMS @1 V14 04CDE;
```

[OPNA]パラメータ番号表：

	1	2										
OP1	3	4	5	6	7	8	9	10	11	12		
OP2	13	14	15	16	17	18	19	20	21	22		
OP3	23	24	25	26	27	28	29	30	31	32		
OP4	33	34	35	36	37	38	39	40	41	42	43	

[OPNA]パラメータ番号に対応する機能：

	CON	FB									
OP1	AR	D1R	D2R	RR	D1L	TL	KS	MUL	DT1	AME	
OP2	AR	D1R	D2R	RR	D1L	TL	KS	MUL	DT1	AME	
OP3	AR	D1R	D2R	RR	D1L	TL	KS	MUL	DT1	AME	
OP4	AR	D1R	D2R	RR	D1L	TL	KS	MUL	DT1	AME	OM

6.48. #MB:FM@OPN : @@FMS用 : 音色データ定義 (OPN 互換)

記述	#MB:FM@OPNA [1] {[2]}
種別	メタデータ定義

【解説】

FM音源 (@@FMS) の音色データを、OPNモードで定義します。
このモードは、4オペレータ、OPN互換モードです。

指定する引数は次の通りです。

引数[1]...音色番号 (num) 【1個の整数】

引数[2]...音色データ (tone) 【複数のパラメータ】

引数[1] (num)

音色番号 (num) は、音源サブモジュール番号指定 (@番号) で、音色を指定する際に使用する番号です。

設定範囲は 0 ~ 1023 です。

音色番号 0 番には初期データが入っています (上書可)。

未定義の音色番号を参照した場合には、0 番が読み出されます。

引数[2] (tone)

音色データ (tone) の開始は「 { 」で認識され、終了は「 } 」によって検知されます。

音色データの各パラメータは、カンマで区切って記述します。

最終パラメータの後のカンマは、あってもなくても構いません。

スペースや改行は読み飛ばされます。

OPNモードでは、39個のパラメータ群により音色データを定義します。

【音色定義例】

```
#MB:FM@OPN 1 {
// CON FB
    4, 5,
// AR DR SR RR SL TL KS MUL DT MASK
    31, 0, 0, 0, 0, 22, 0, 2, 7,
    18, 10, 0, 6, 1, 0, 0, 8, 7,
    31, 0, 0, 0, 0, 23, 0, 4, 3,
    18, 10, 0, 6, 1, 0, 0, 4, 3, 15
}
@@FMS @1 V14 04CDE;
```

[OPN]パラメータ番号表：

	1	2								
OP1	3	4	5	6	7	8	9	10	11	
OP2	12	13	14	15	16	17	18	19	20	
OP3	21	22	23	24	25	26	27	28	29	
OP4	30	31	32	33	34	35	36	37	38	39

[OPN]パラメータ番号に対応する機能：

	CON	FB								
OP1	AR	D1R	D2R	RR	D1L	TL	KS	MUL	DT1	
OP2	AR	D1R	D2R	RR	D1L	TL	KS	MUL	DT1	
OP3	AR	D1R	D2R	RR	D1L	TL	KS	MUL	DT1	
OP4	AR	D1R	D2R	RR	D1L	TL	KS	MUL	DT1	OM

6.49. FM音源の音色作成に関わる用語説明

V2MMLで扱うFM音源 (@@FMS) はOPMの上位互換です。

6つのオペレータを持っており、3種のオペレータモード (2 / 4 / 6) において音色を作成できます。

加えて、OPMのハードウェアに備わっていたLFO機能もサポートしており、この機能は便宜上ハードLFO (HLFO) と呼びます。

オペレータ (operator)

オペレータとは、1個の正弦波発生器です。

FM音源の音色作成では、各オペレータに対してパラメータを設定します。オペレータの各パラメータは、次の2つに大別されます。

エンベロープジェネレータ部 (振幅の時系列操作)
フェーズジェネレータ部 (周波数と位相の操作)

さらに、「オペレータ同士を接続」することで周波数変調を行います。

変調される側のオペレータを「キャリア」

変調する側のオペレータを「モジュレータ」

と呼びます。実際に音を出すオペレータは「キャリア」です。

・HLFO (hardware low frequency oscillator)

HLFOは、FM音源 (@@FMS) モジュール側でサポートするLFO機能です。

OPM互換モードと、OPNA互換モードがあります。

LFOとは、ビブラート (周波数をゆらす) や、トレモロ (振幅をゆらす) を行う機能です。

OPM互換モードのHLFOは、

WF、LFRQ、PMD、AMD、PMS、AMS、SYNC

の7種類で設定します。

WF : wave form

OPM互換HLFOで使用する波形を選択します。

指定範囲は 0~3。

0 : のこぎり波

1 : 矩形波

2 : 三角波

3 : ランダム波 (ノイズ)

LFRQ : lfo frequency

OPM 互換 H L F O の周波数。範囲は 0~255。

PMD : pitch modulation depth

音程深度を指定。範囲は 0~127。

AMD : amplitude modulation depth

音量深度を指定。範囲は 0~127。

PMS : pitch modulation sensitivity

音程感度を指定。範囲は 0~7。

AMS : amplitude modulation sensitivity

音量感度を指定。範囲は 0~3。

SYNC : synchronization

H L F O 位相のノートオン同期機能。範囲は 0~1。

OPNA 互換モードの H L F O は、

LFRQ、PMS、AMS、SYNC

の 4 種類で設定します。波形パターンは正弦波のみです。

LFRQ : lfo frequency

OPNA 互換 H L F O の周波数。範囲は 0~7。

PMS : pitch modulation sensitivity

音程感度を指定。範囲は 0~7。

AMS : amplitude modulation sensitivity

音量感度を指定。範囲は 0~3。

SYNC : synchronization

H L F O 位相のノートオン同期機能。範囲は 0~1。

・エンベロープジェネレータ部

エンベロープでは、オペレータごとに、振幅の時系列操作の設定を行います。
パラメータは、

AR、D1R、D2R、RR、D1L、TL、KS、AME、MSK、OM、IEL
の11種類です。

AR : attack rate (0~31)

アタックレートは、ノートオンから最大レベルになるまでの速さです。
設定値が大きいほど速い立ち上がりになります。
最大レベルとは、TLで設定する出力レベルに相当します。

D1R : decay 1 rate (0~31)

ディケイ1レートは、アタックレート終了直後の減衰速度です。
設定値が大きいほど速い減衰になります。
OPN系では「DR」と呼ばれます。

D2R : decay 2 rate (0~31)

ディケイ2レートは、ディケイ1レート終了直後の減衰速度です。
設定値が大きいほど速い減衰になります。
OPN系では「SR(sustain rate)」と呼ばれます。

RR : release rate (0~15)

リリースレートは、ノートオフから最小レベルになるまでの速度です。
設定値が大きいほど速い減衰になります。

D1L : decay 1 level (0~15)

ディケイ1レベルは、ディケイ1レートからディケイ2レートへ
移行する境界となる出力レベルです。
0 のとき 0dB です。(結果的にディケイ1レートがスキップされます)
1 あたり -3dB で、境界となるレベルが変化します。
15 のときは特例で、-93dB になります。(-45dB からさらに -48dB)
15 のときは結果的にディケイ2レートが無効になります。
OPN系では「SL(sustain level)」と呼ばれます。

TL : total level (0~127)

トータルレベルは、振幅の最大出力レベルです。
アタックレートにおける最大レベルに相当します。
設定値が大きいほど出力が小さくなります。
0 のとき 0dB です。(出力最大)
1 あたり -0.75dB で出力レベルが変化します。
127 のとき -95.25dBです。(出力最小)
TLは小数以下の指定も受け付けますが、小数以下9ビットの固定小数点数

の精度に丸められて応答します。

これは内部の振幅値テーブルサイズ（解像度）の都合によるものです。

KS : key scale (0~3)

キースケールは、発音音程に追従して、AR, D1R, D2R, RR のレートを自動的に内部で変化させる機能です。

0 でキースケールしません。

指定値が大きくなるほど、高い音程でエンベロープ変化が速くなります。

AME : amplitude modulation enable switch (0~1)

AMEは、HLFOの振幅変化の対象となるオペレータを設定します。

0 のとき、HLFO振幅変化の対象にしません。

1 のとき、HLFO振幅変化の対象とします。

この機能により、モジュレータのみ、もしくはキャリアのみに振幅変調を与えるような選択が可能になります。

MSK : operator mask (0~1)

オペレータマスクは、対象オペレータの出力をマスクするかどうかを設定します。

0 のとき、通常通りのオペレータ出力。

1 のとき、オペレータ出力をマスクします。

OM : operator mask (0~15)

こちらのオペレータマスクは、OPM/OPN(A)互換の定義内で使用する、1つのパラメータで4つのオペレータに対しマスク設定するものです。

0~15で表される4ビットのうち、

最下位からオペレータ1、2、3、4に対応します。

全ビット1 (15) のとき、全オペレータ通常通りの出力です。

全ビット0 (0) のとき、全オペレータがマスク状態になります。

IEL : initial envelope level (-1 または 0~128)

IELは、ノートオンによるエンベロープ開始時（アタック開始時）の出力レベルを設定します。

-1 のとき、ノートオン直前の出力レベルを引き継ぎます。

IELが使用できない音色定義では IEL は -1 に内部で設定されます。

0~128 の範囲では、アタック開始時の出力レベルを強制的に設定します。設定値と初期出力レベルの対応は、次の通りです。

0 のとき、0dB からアタック開始。

128 のとき、-96dB からアタック開始。

（1 増えるごとに -0.75dB された出力レベルからのアタック開始）

・フェーズジェネレータ部

オペレータごとに、周波数と位相の操作の設定を行います。

パラメータは、

MUL、DT1、DT2、DT3、DT4、FB、IPH

の7種類です。

MUL : multiple (0~32) (特殊指定 : -1)

マルチプルは、ノートオン周波数への倍率を指定します。

少数以下の指定、分数による指定も受け付けます。

ただし、0 は 0.5 倍とみなされます。(OPM/OPN(A)互換のため)

特殊指定で、-1 を指定すると 0倍 になります。

0倍の指定は、DT4 が使用できる音色設定において、

DT4 の周波数加算指定を、ノートオン周波数に左右されない

固定的な発音周波数として利用する使い方を想定しています。

(DT4 が使用できない音色定義では、-1 の特殊指定は使えません)

DT1 : detune 1 (0~7)

デチューン1は、周波数を微妙に ずらします。

1~3 のとき、数値が大きいほど周波数が高くなります。

5~7のとき、数値が大きいほど周波数が低くなります。

0,4のとき、周波数の変化を与えません。

※デチューン1による周波数ずれは微小です。

DT2 : detune 2 (0~3)

デチューン2は、周波数を大きく ずらします。

指定値と周波数倍率の対応は次の通りです。

0 のとき、1.00倍

1 のとき、1.41倍

2 のとき、1.57倍

3 のとき、1.73倍

DT3 : detune 3 (-9600~9600)

デチューン3は、周波数を cent単位で ずらします。

cent単位では、100あたり半音、1200あたり1オクターブずれます。

0 で周波数変化なしです。

小数以下の指定、分数による指定も受け付けます。

DT3 が使用できない音色定義では、DT3 は 0 として内部で設定されます。

DT4 : detune 4 (-1048576~1048576)

デチューン4は、周波数を Hz単位で ずらします。

0 で周波数変化なしです。

小数以下の指定、分数による指定も受け付けます。

MUL の特殊指定で、ノートオン周波数の 0 倍 になっている時は、DT4 で指定する周波数がそのままノートオン周波数になります。DT4 が使用できない音色定義では、DT4 は 0 として内部で設定されます。

FB : feedback level (0~15)

フィードバックは、自らのオペレータ出力を変調入力とする機能です。OPM/OPN(A)互換モードでは、オペレータ 1 のみが FB 機能の対象ですが、それ以外のモードでは、全オペレータが対象です。

0 を指定した場合は、FB は無効です。

0 より大きい数値を指定した場合、指定値を n とすると、

$$FB = 2^{(n-5)} \pi$$

となり、指定値が大きくなるほど強い FB が掛かります。

($n=7$ のとき $FB=4\pi$)

少数以下の指定、分数による指定も受け付けます。

IPH : initial phase (0.0以上~1.0未満 または -1.0以上~0.0未満)

イニシャル・フェーズは、ノートオン時の位相を設定します。

小数以下の指定、分数による指定も受け付けます。

0.0以上~1.0未満のとき

ノートオンの都度、位相を (指定値 $\times 2\pi$) にします。

OPM/OPN(A)互換の音色定義では、IPH は 0 として内部で設定されます。

-1.0以上~0.0未満のとき

ノートオンの都度、位相を設定しません。

ただし、音色設定時など、IPHを設定した時にだけ、

位相を (指定値 $\times (-1) \times 2\pi$) にします。

このとき、指定値が (-1) の場合、位相は 0 と同様です。

・オペレータ同士の接続

CON : connection

コネクションは、オペレータ同士の接続形態
(キャリアとモジュレータの決定、および変調パターン選択)
を指定します。

6 オペレータモードの接続形態

4 オペレータモードの接続形態

2 オペレータモードの接続形態

6.50. FM音源6オペレータモードの接続形態

6オペレータモード音色設定

#MB:FM@SS60P (6オペレータType-SS)

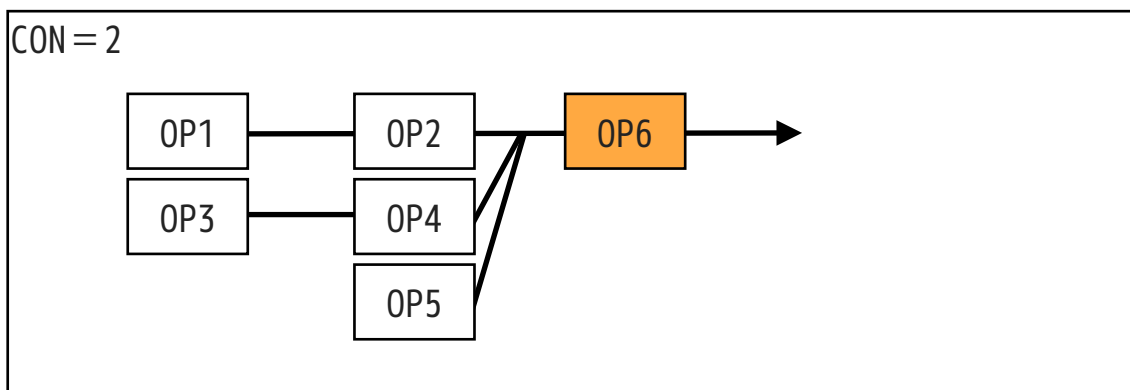
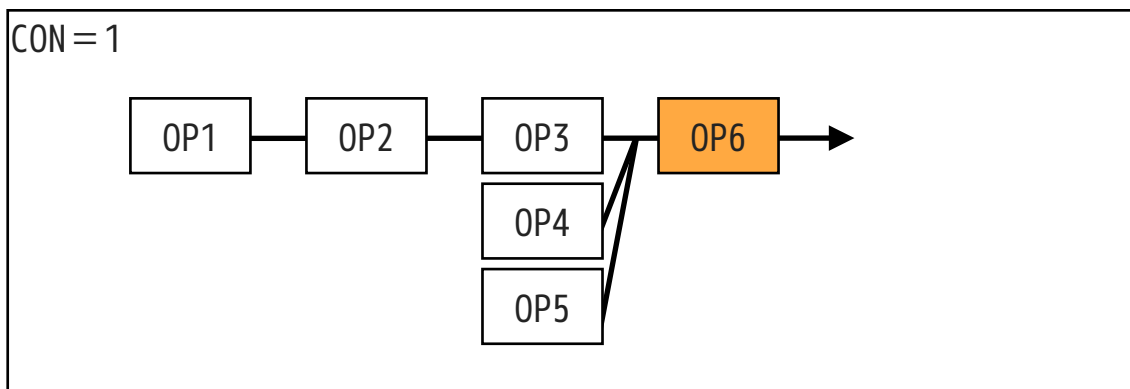
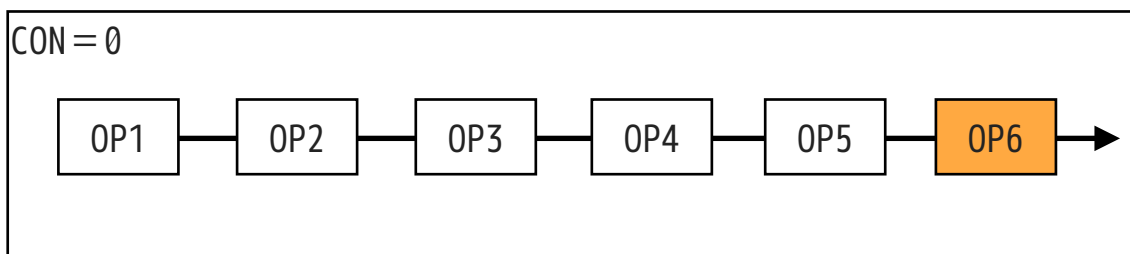
における、CON (connection: オペレータの接続パターン) の内容を示します。

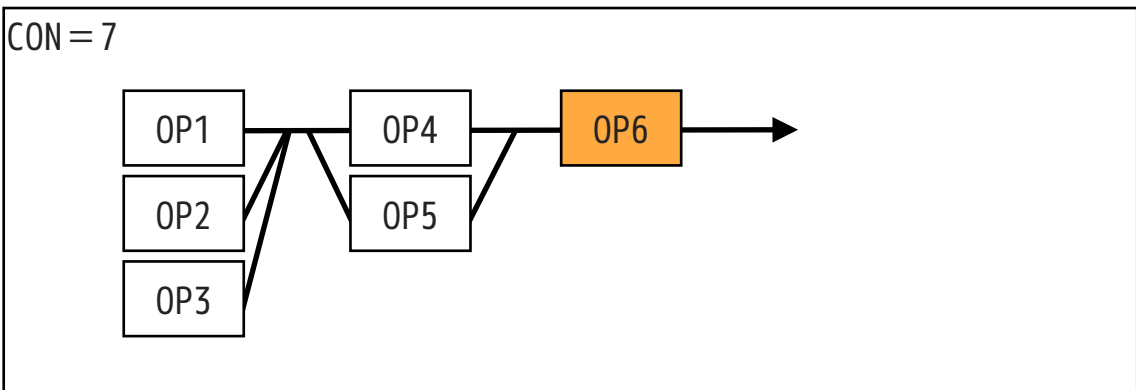
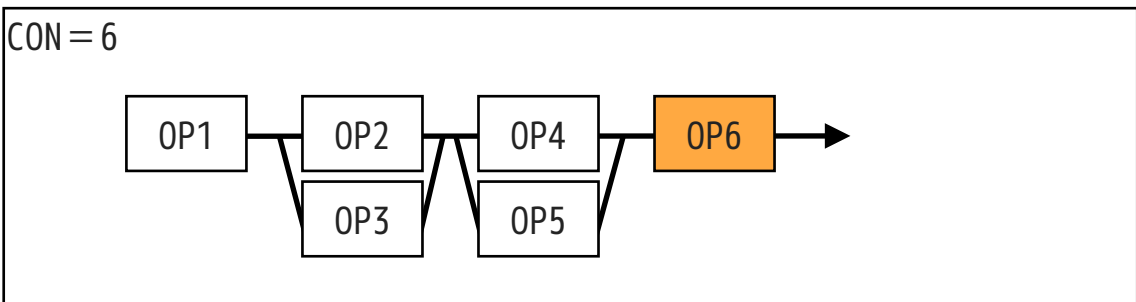
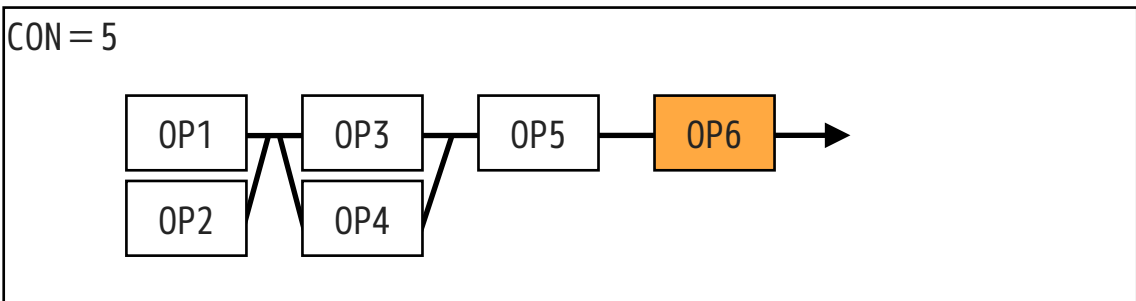
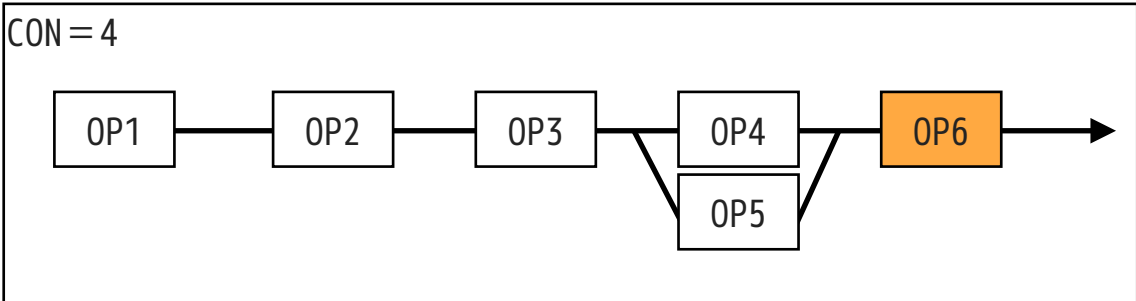
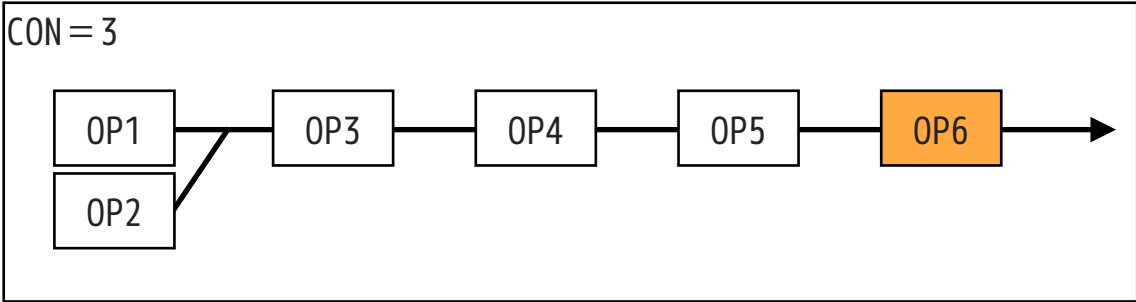
色付きのオペレータがキャリアで、それ以外がモジュレータです。

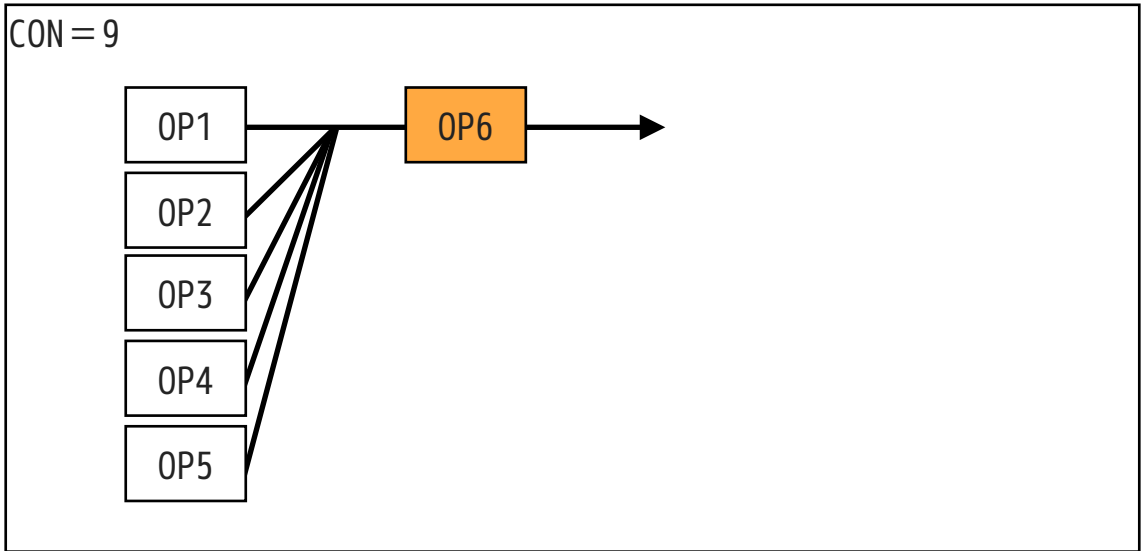
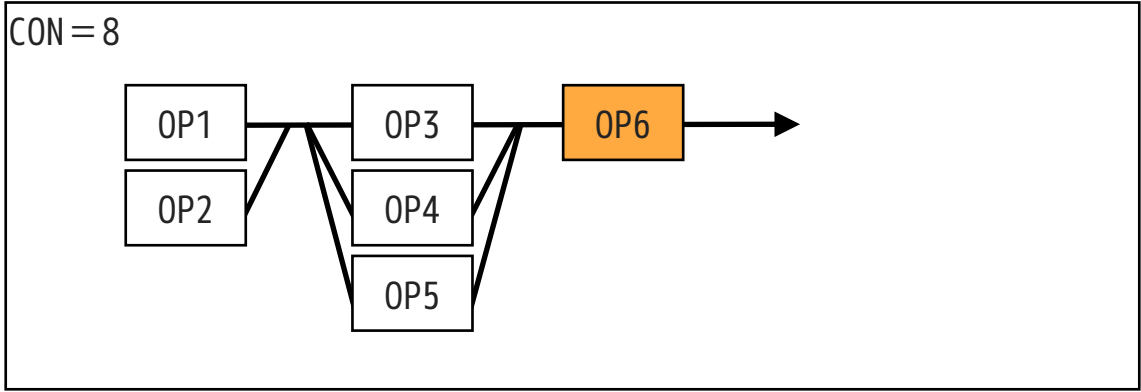
キャリアが複数ある CON では、コーラスや和音の効果を出すことも可能です。
全てのオペレータにセルフフィードバック機能があります。

6オペレータモードの CON は、0~63 で指定します。

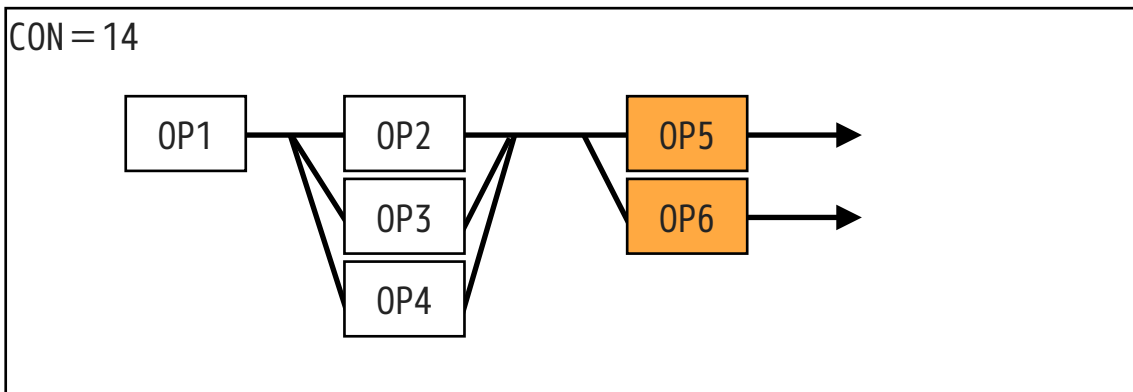
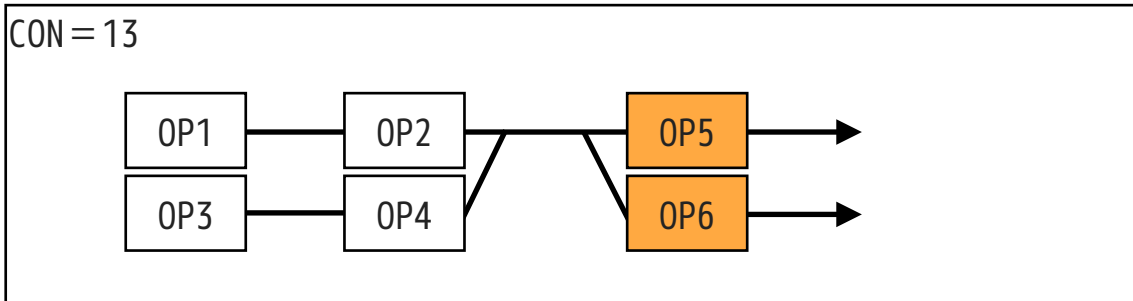
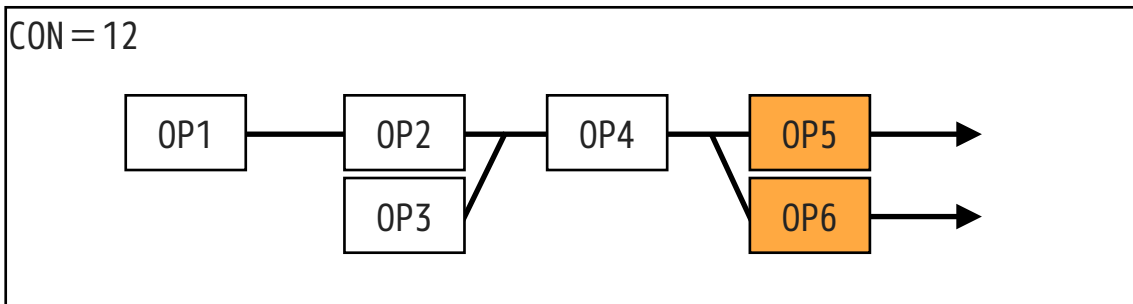
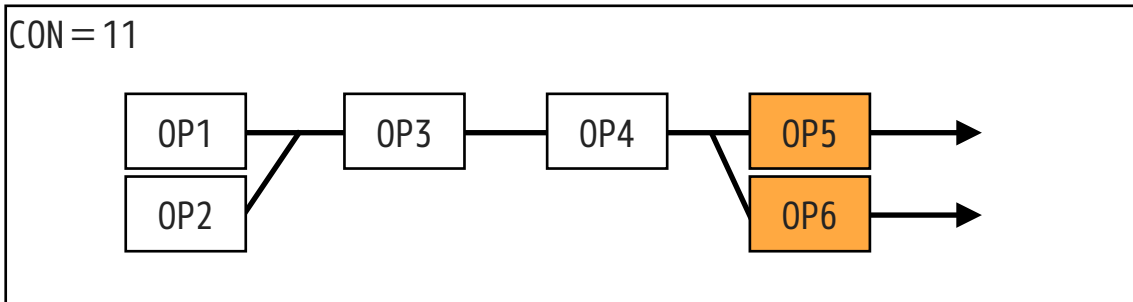
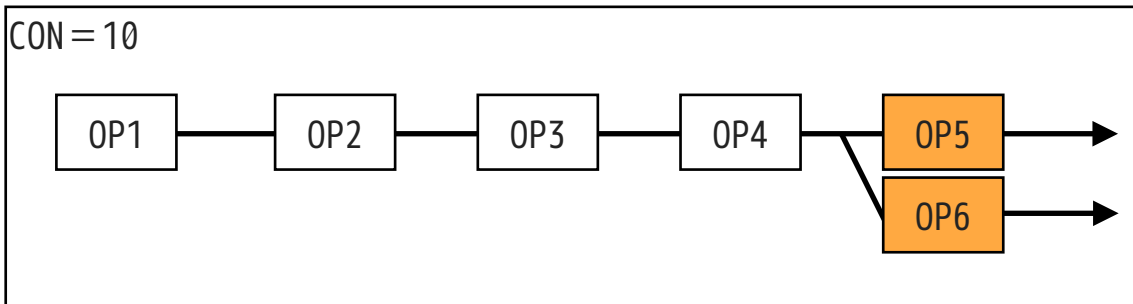
グループ1 : CON=0~9 : [1,2,3,4,5,6]

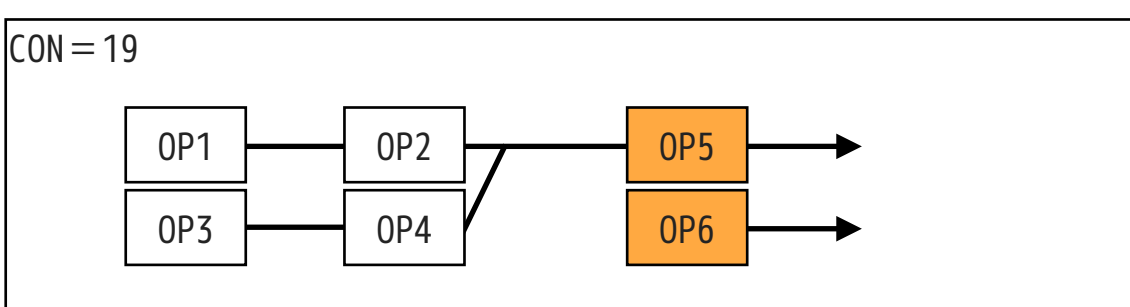
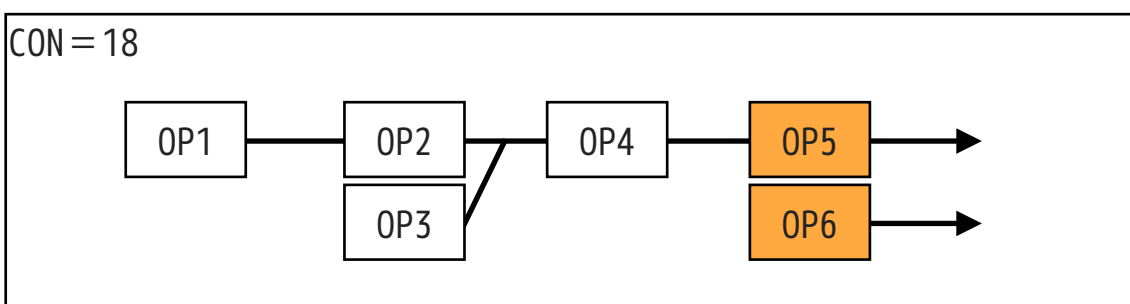
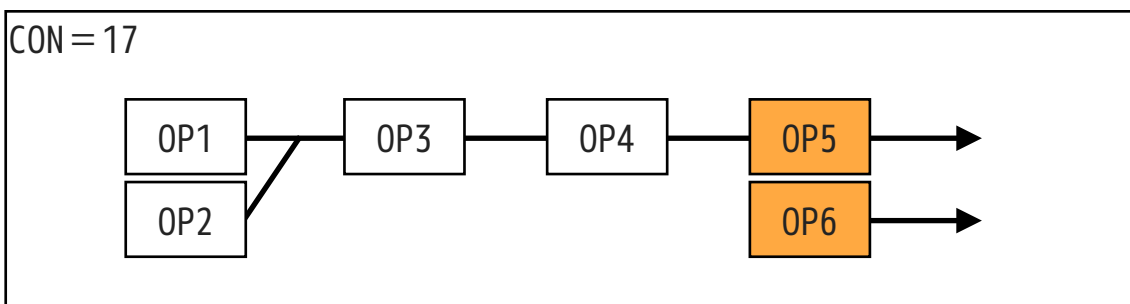
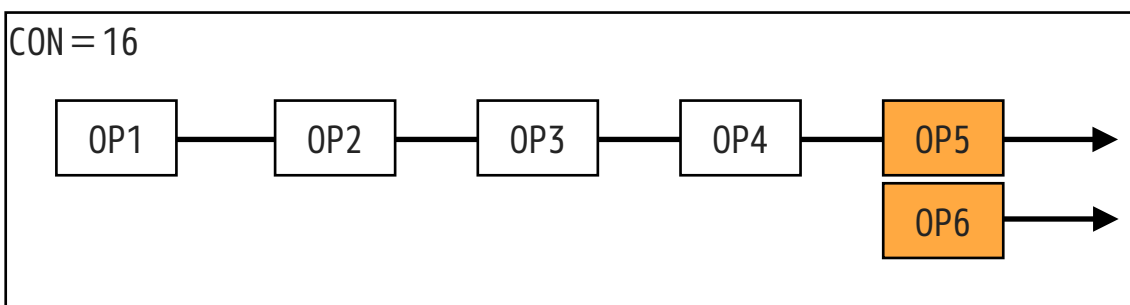
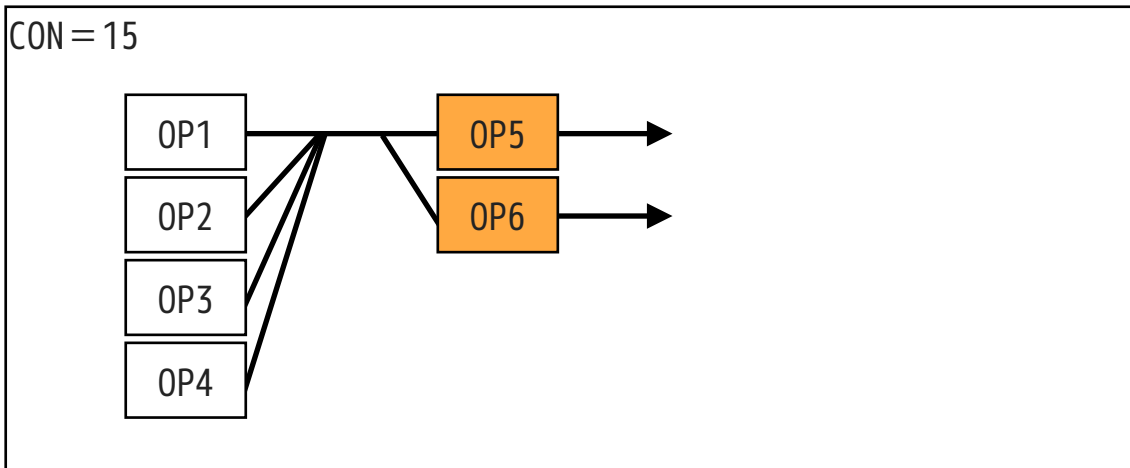




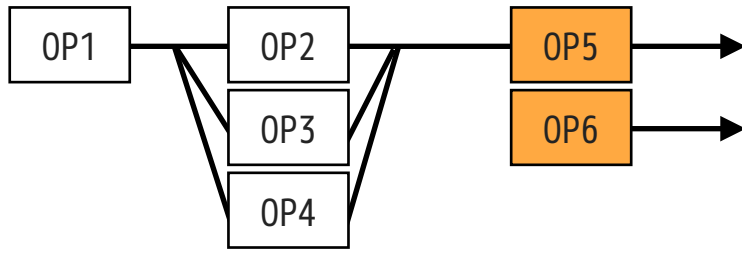


グループ 2 : CON=10~21 : [1,2,3,4,5][6]

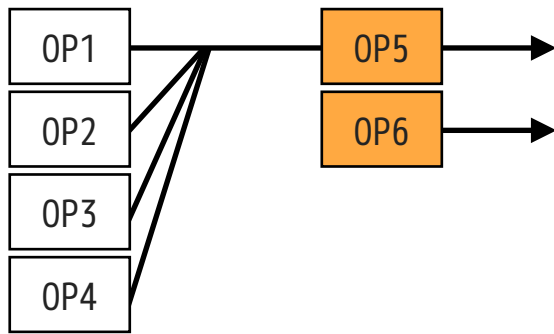




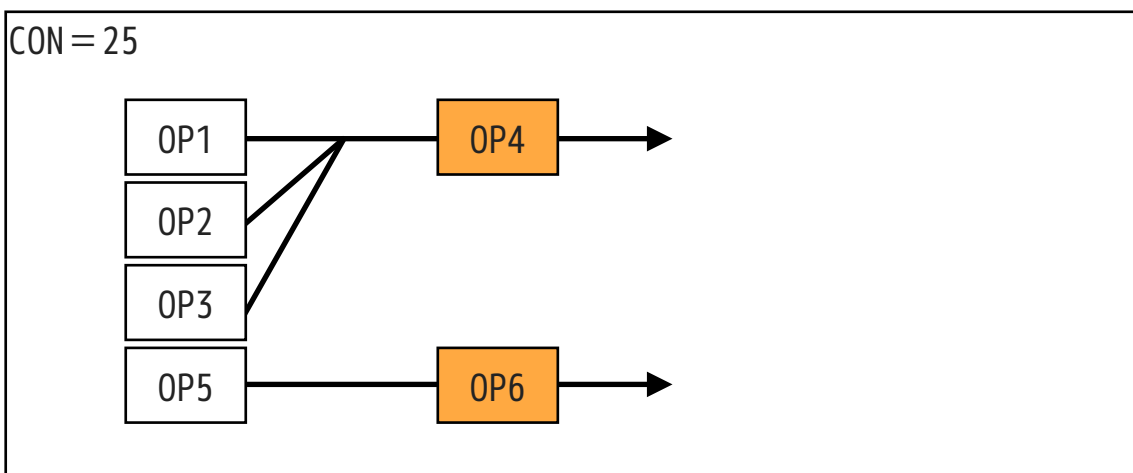
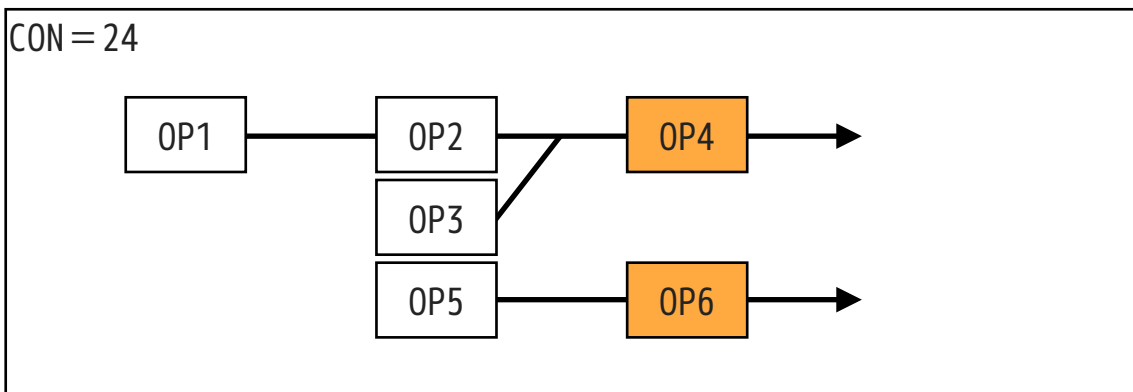
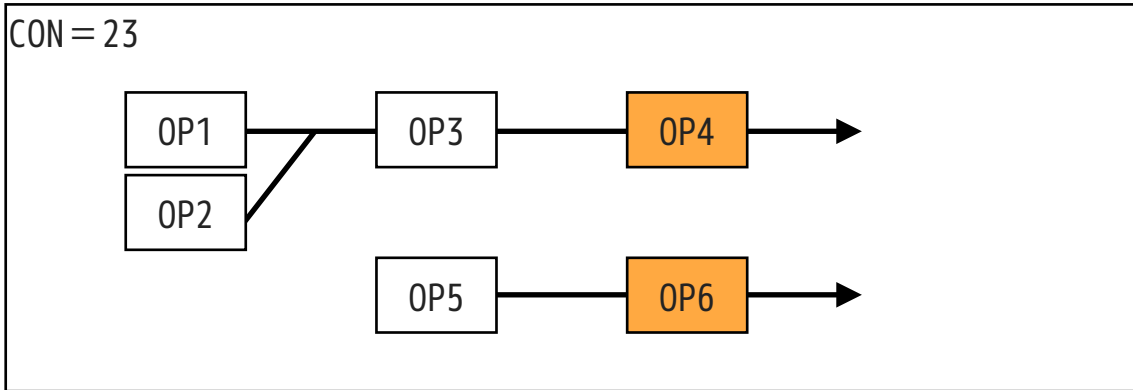
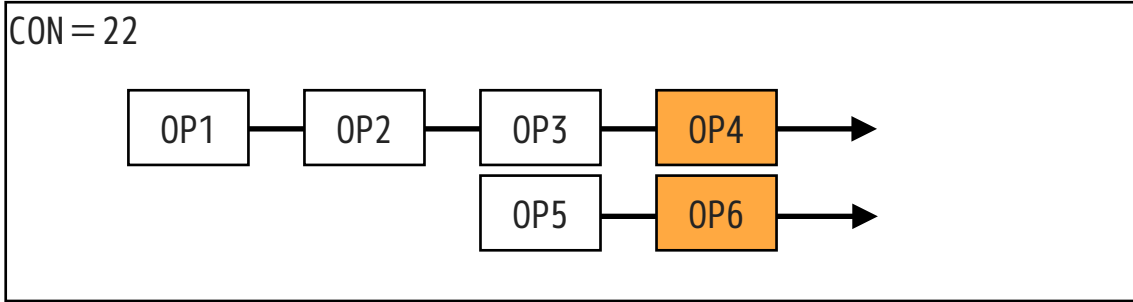
CON = 20



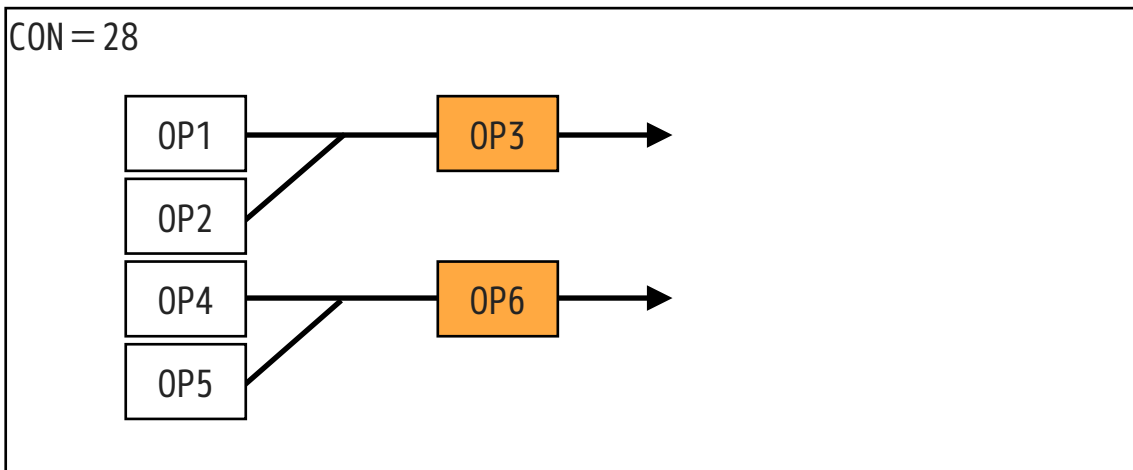
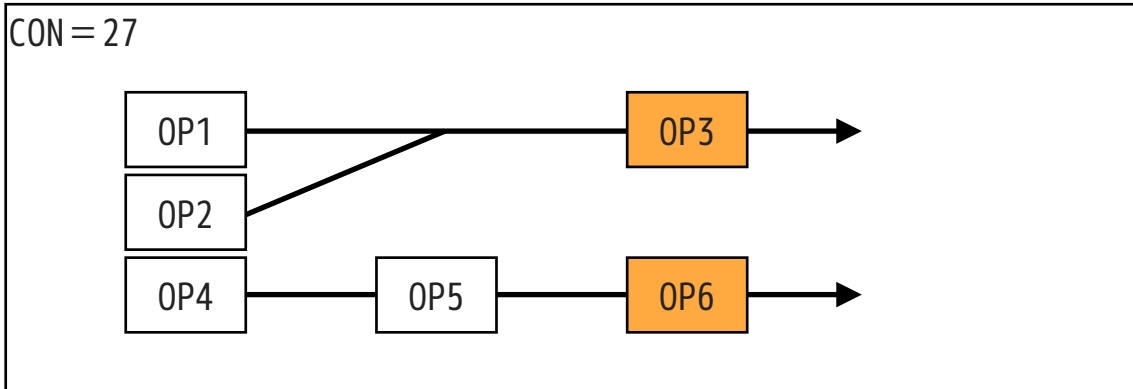
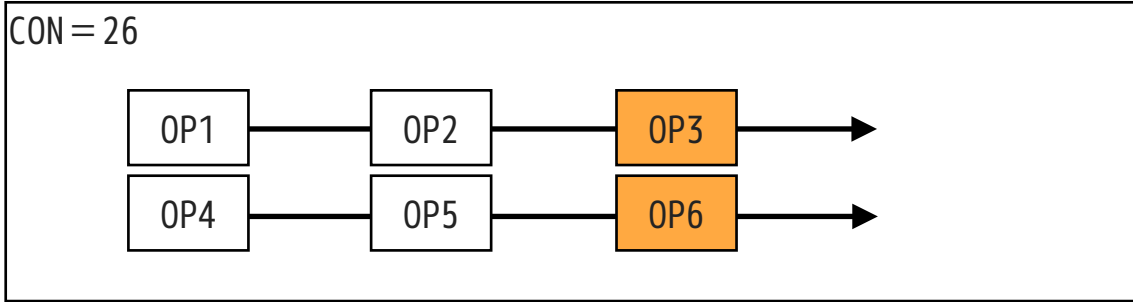
CON = 21



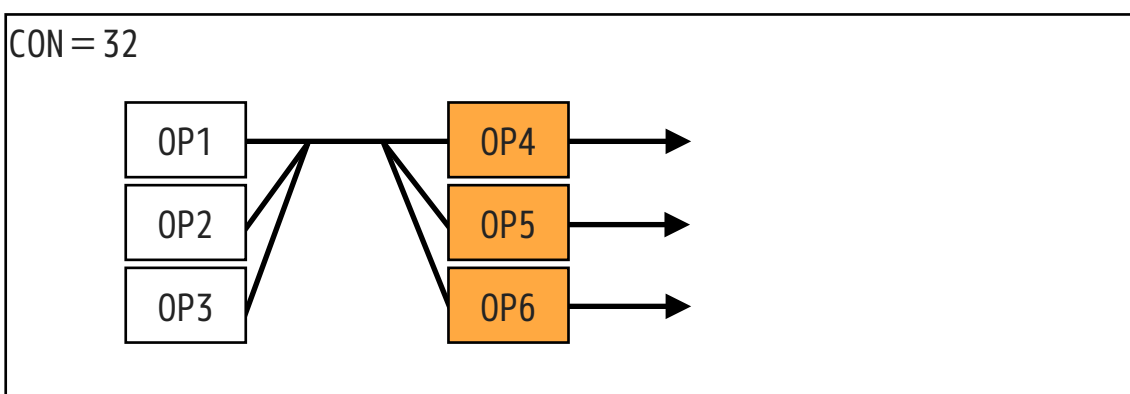
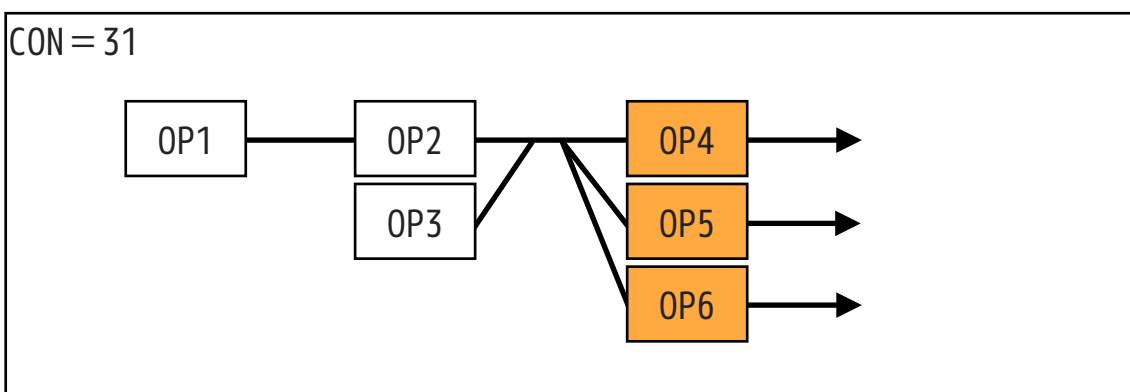
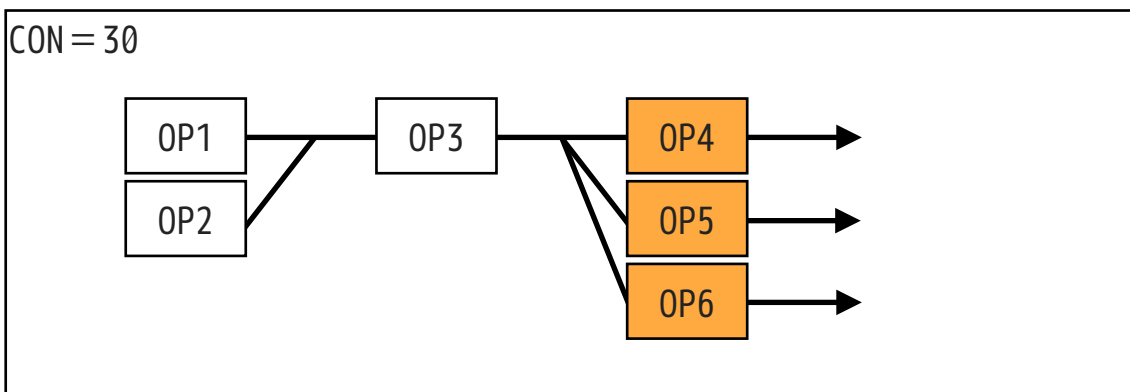
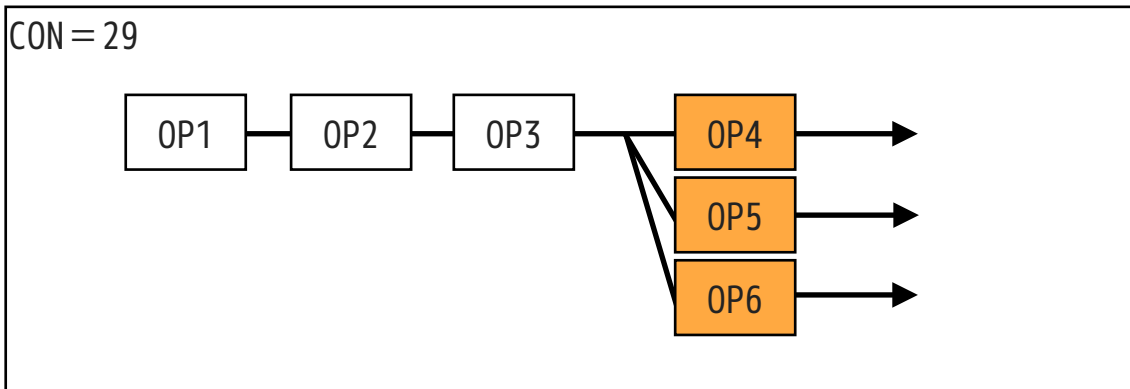
グループ 3 : CON=22~25 : [1,2,3,4][5,6]



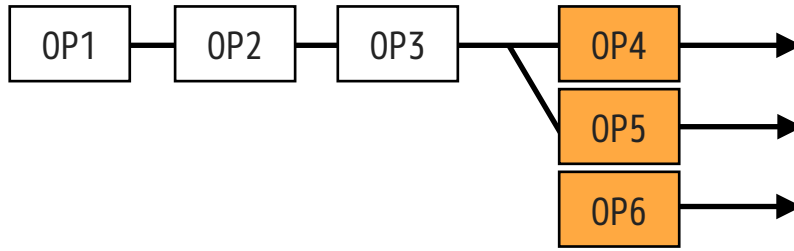
グループ4 : CON=26~28 : [1,2,3][4,5,6]



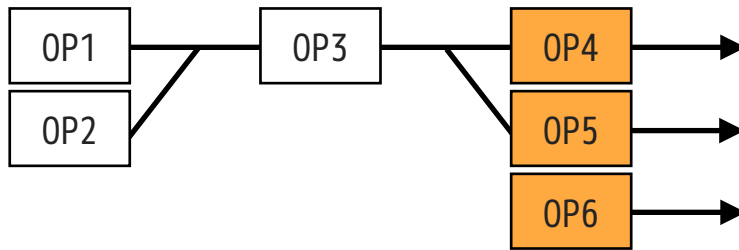
グループ 5 : CON=29~40 : [1,2,3,4][5][6]



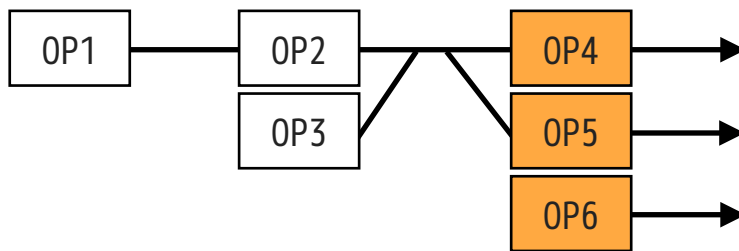
CON = 33



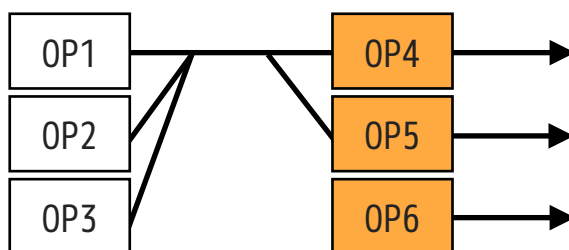
CON = 34

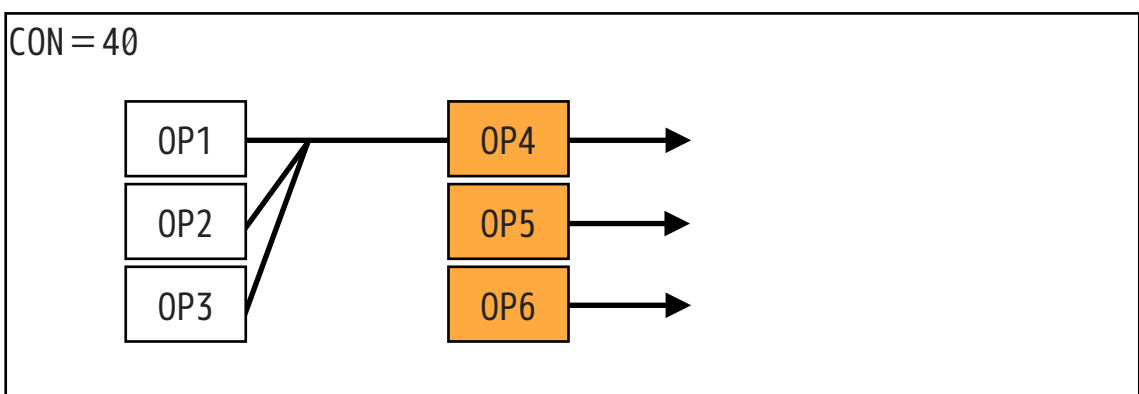
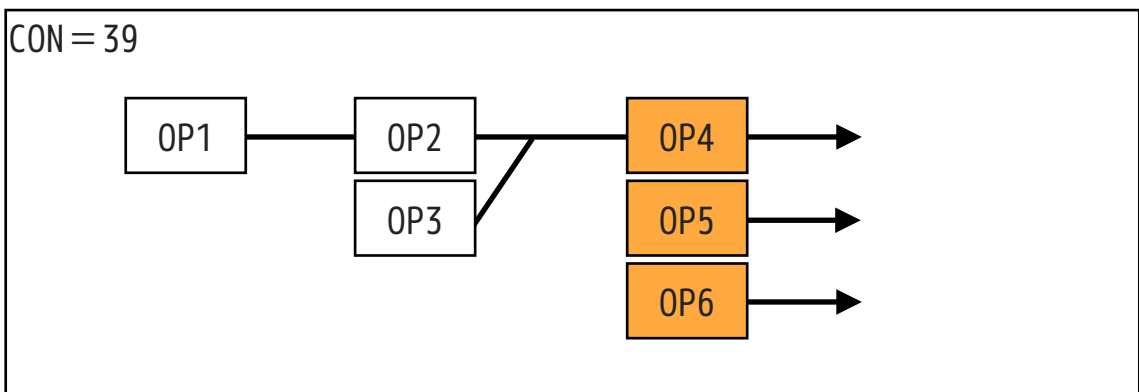
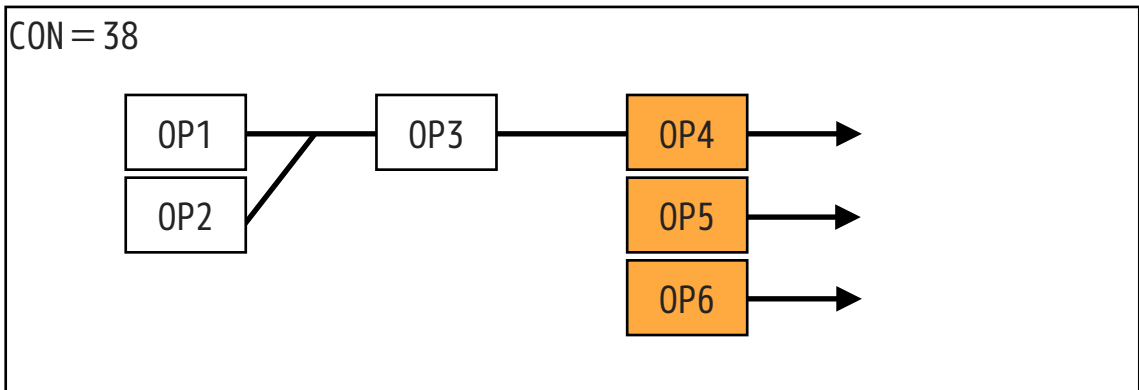
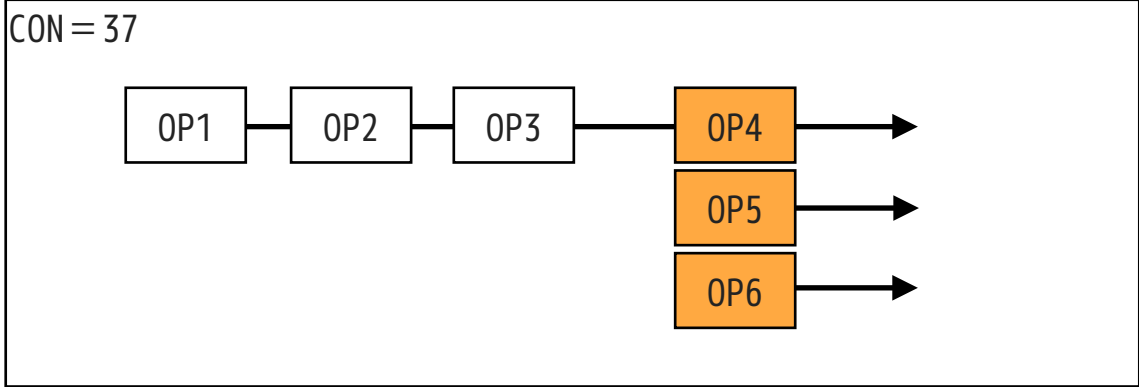


CON = 35

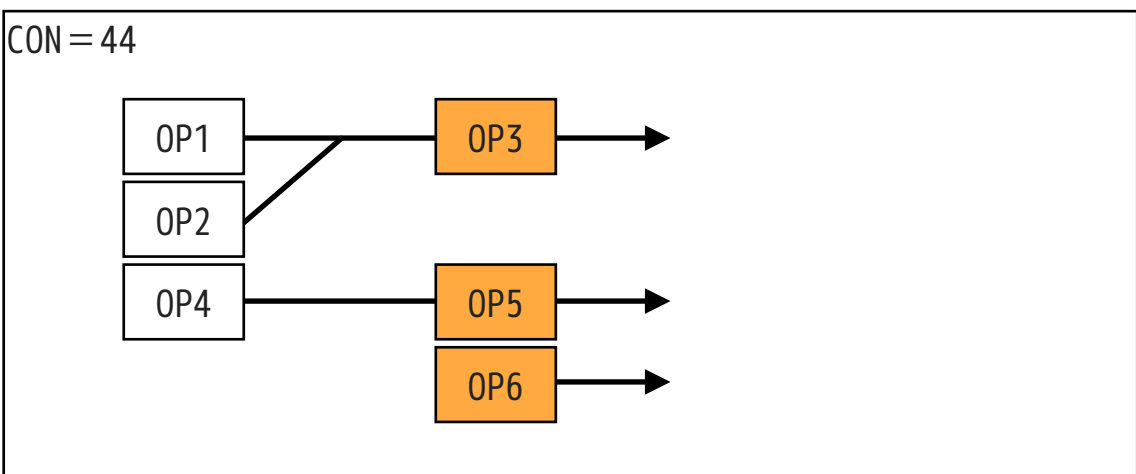
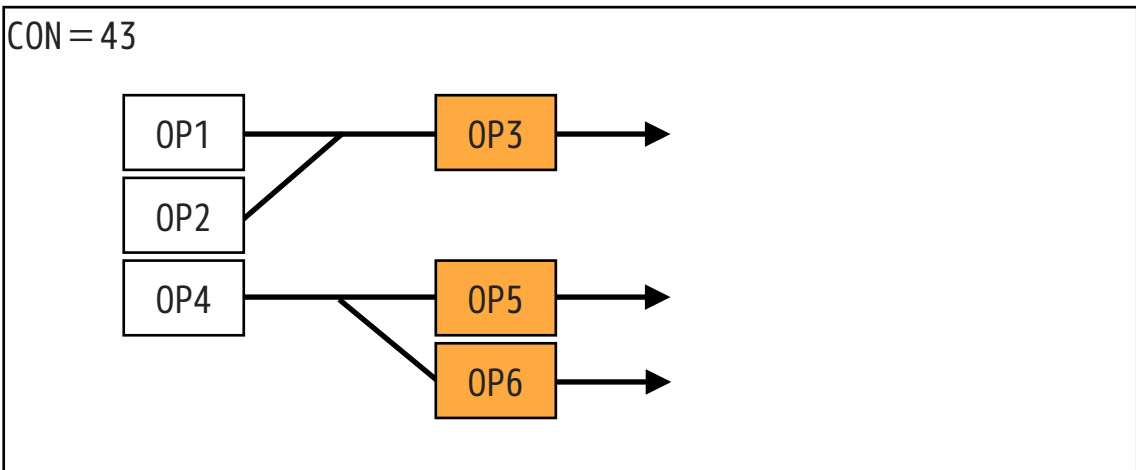
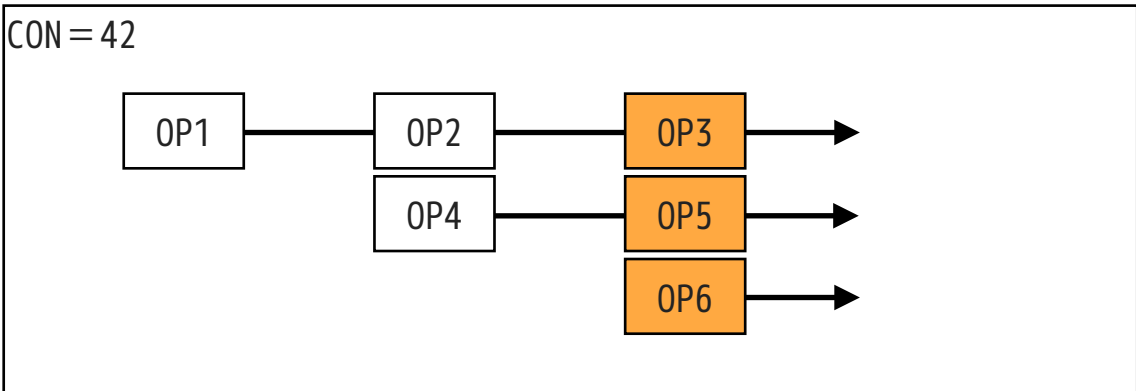
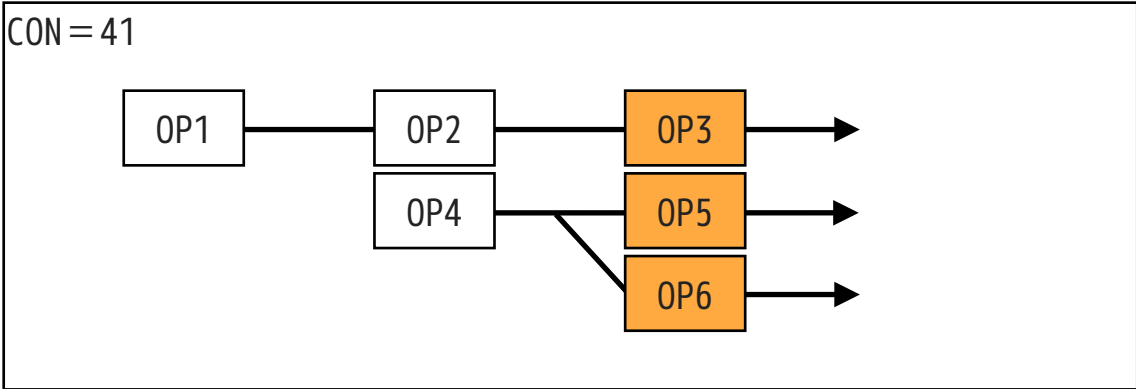


CON = 36

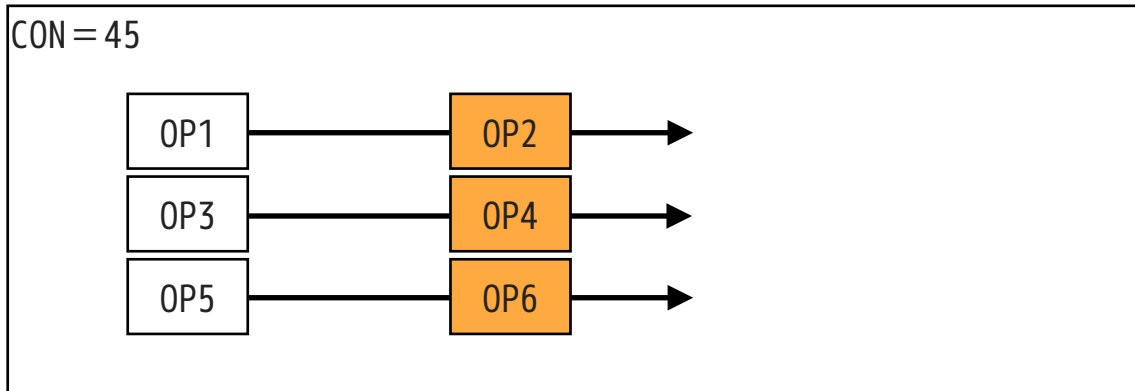




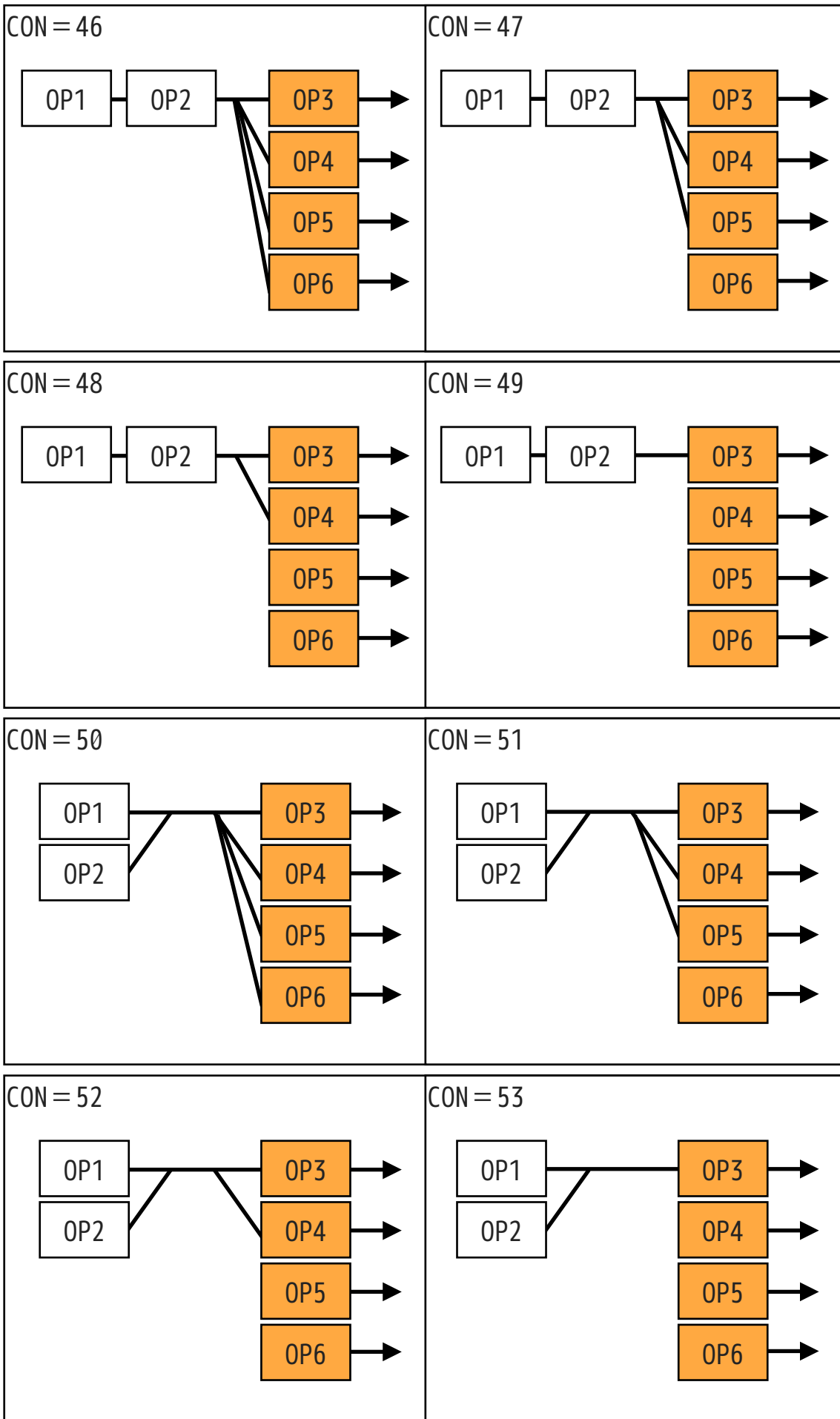
グループ 6 : CON=41~44 : [1,2,3][4,5][6]



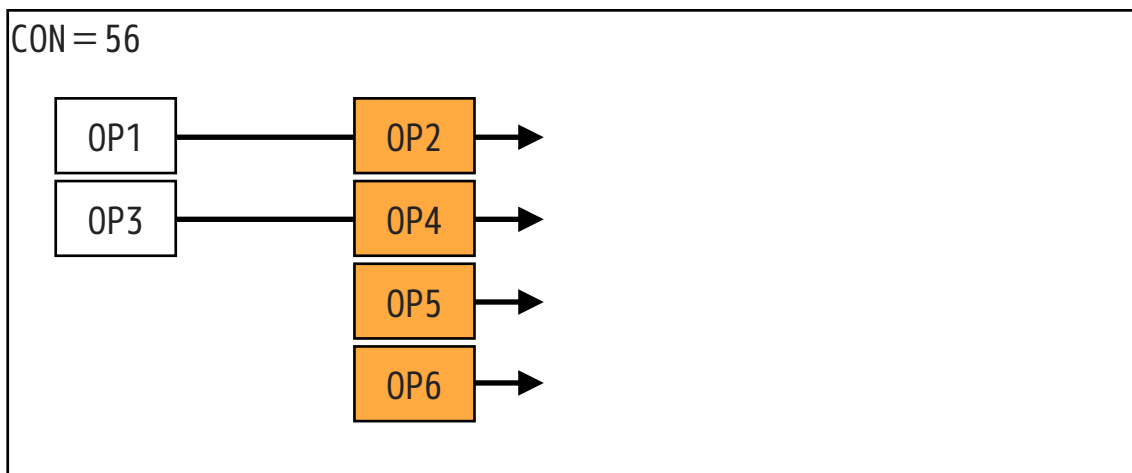
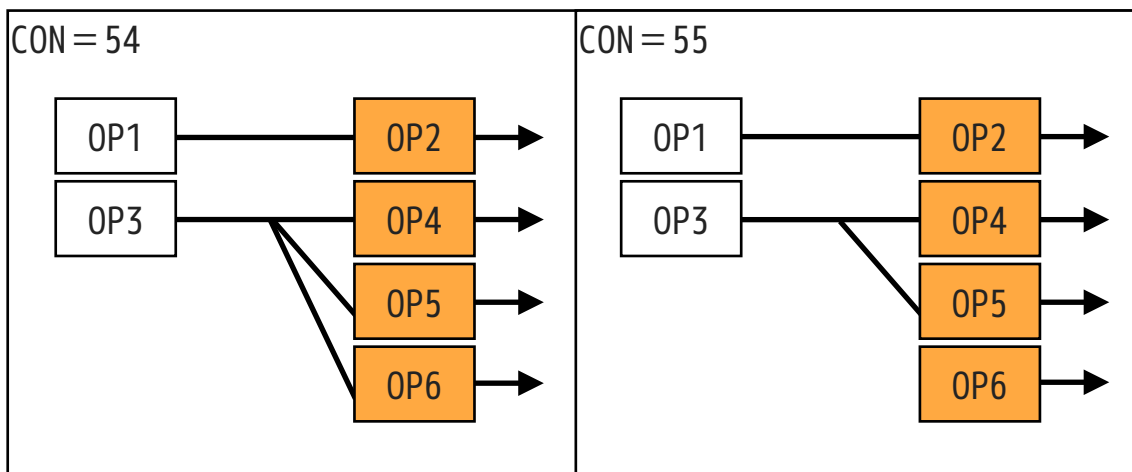
グループ7 : CON=45 : [1,2][3,4][5,6]



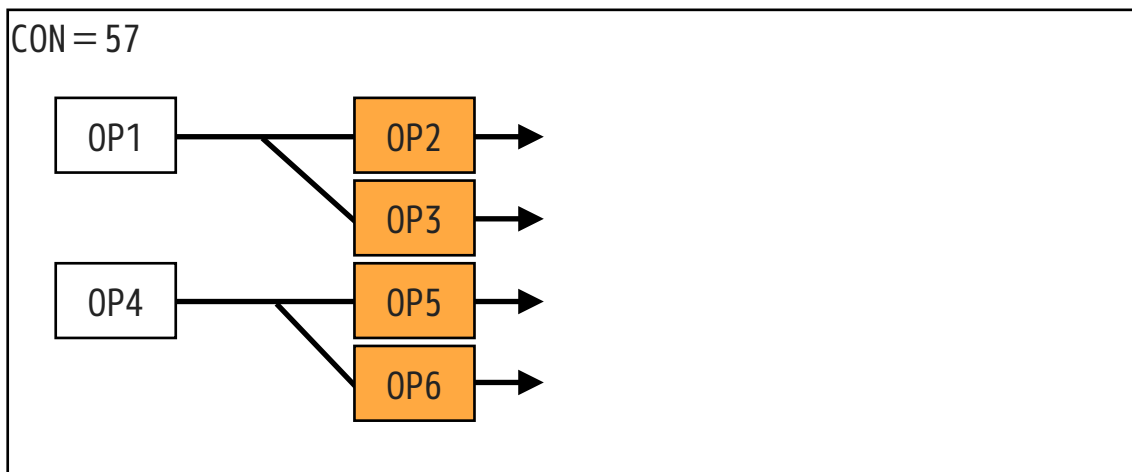
グループ8 : CON=46~53 : [1,2,3][4][5][6]



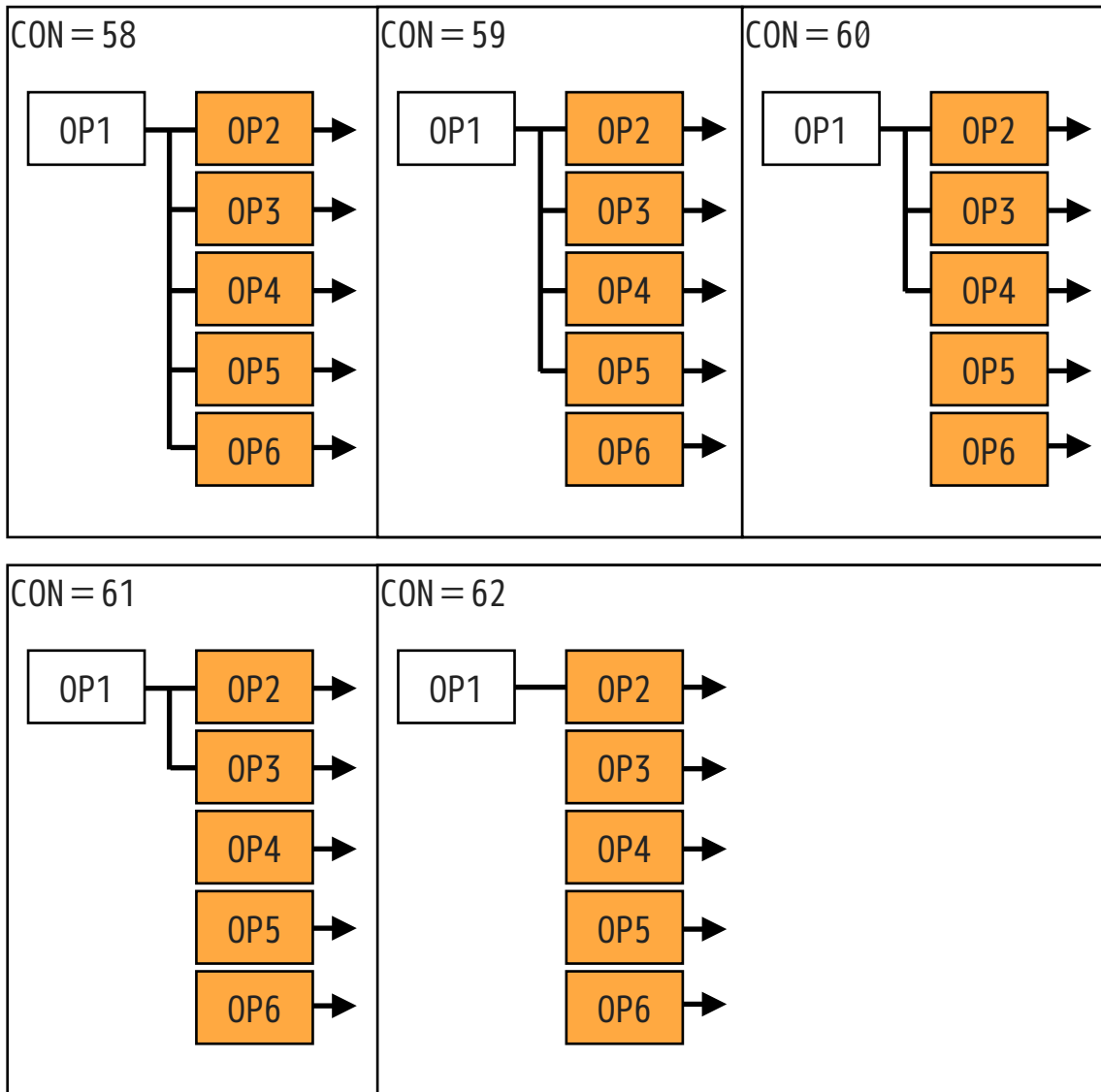
グループ9 : CON=54~56 : [1,2][3,4][5][6]



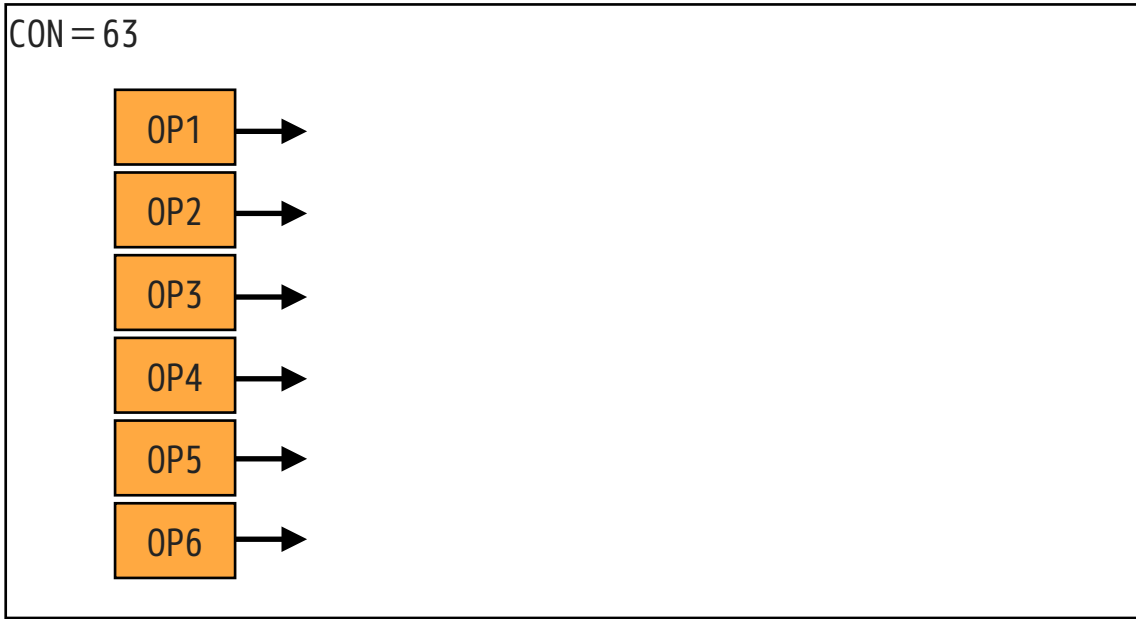
グループ10 : CON=57 : [1,2][3][4,5][6]



グループ 1 1 : CON=58~62 : [1,2][3][4][5][6]



グループ 1 2 : CON=63 : [1][2][3][4][5][6]



6.51. FM音源4オペレータモードの接続形態

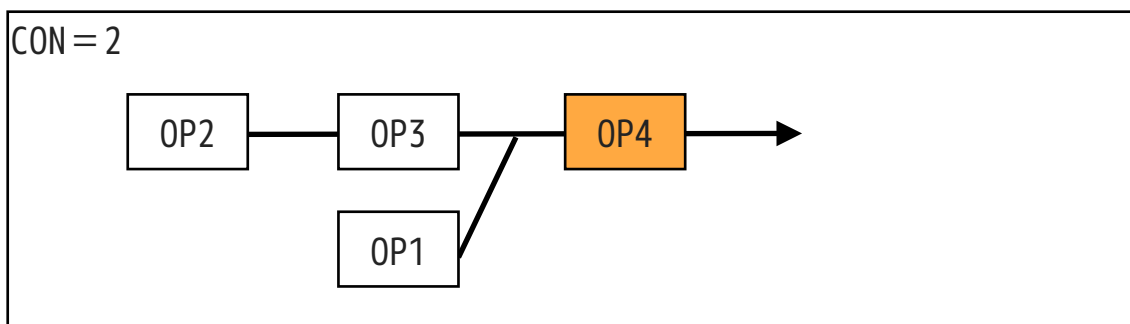
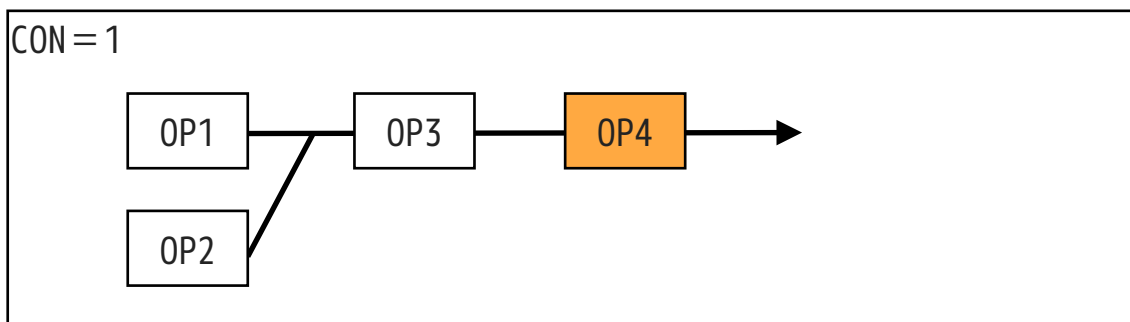
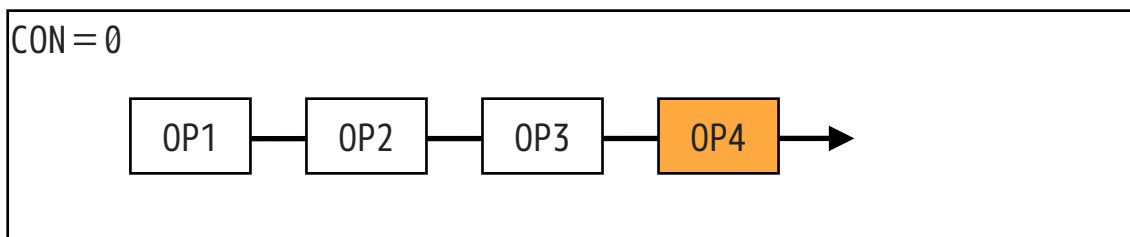
4オペレータモードの音色設定

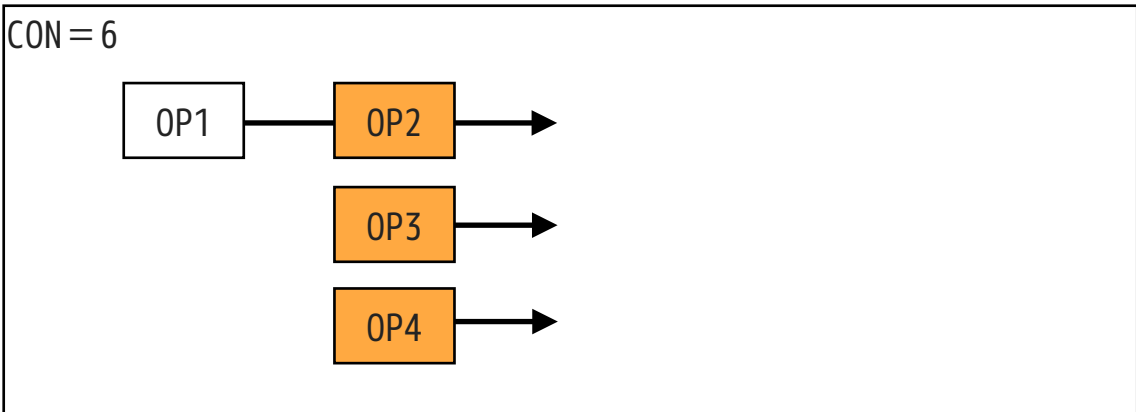
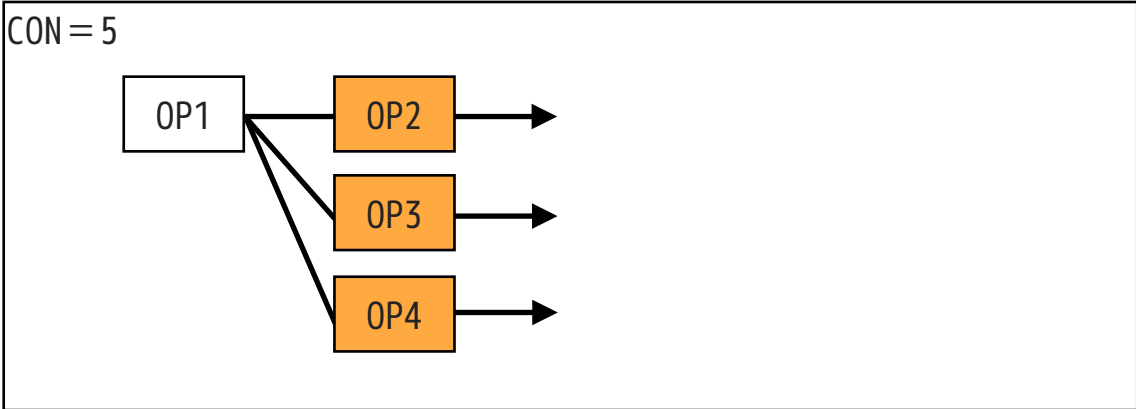
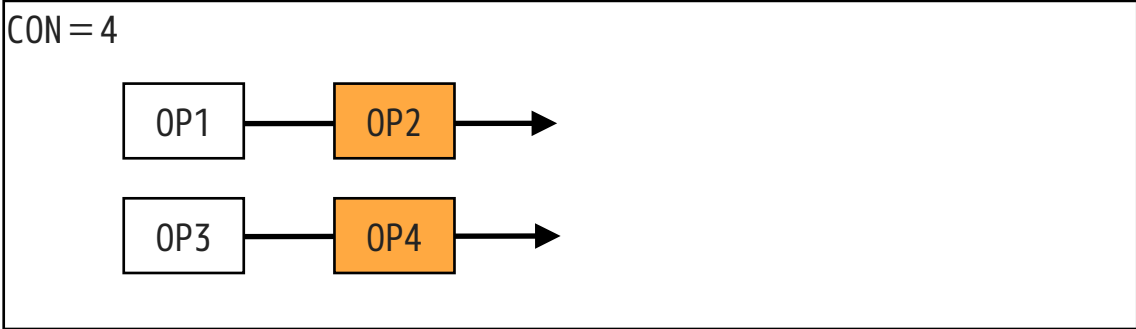
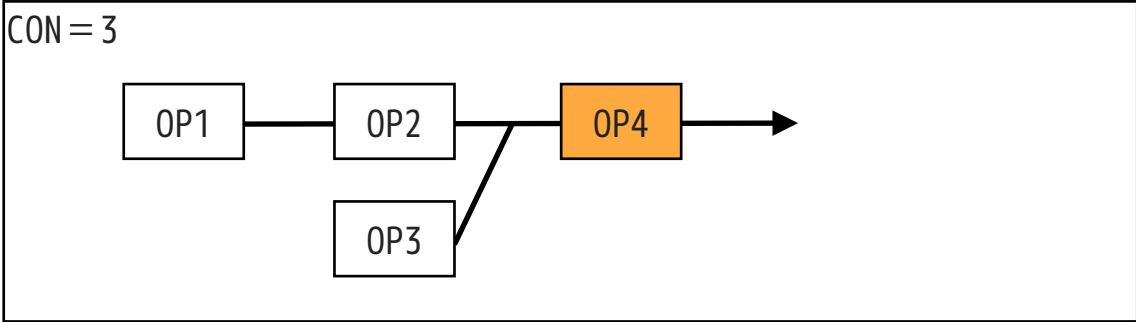
#MB:FM@SS40P (4オペレータType-SS)
 #MB:FM@S (4オペレータType-S)
 #MB:FM@OPM (OPM互換)
 #MB:FM@OPNA (OPNA互換)
 #MB:FM@OPN (OPN互換)

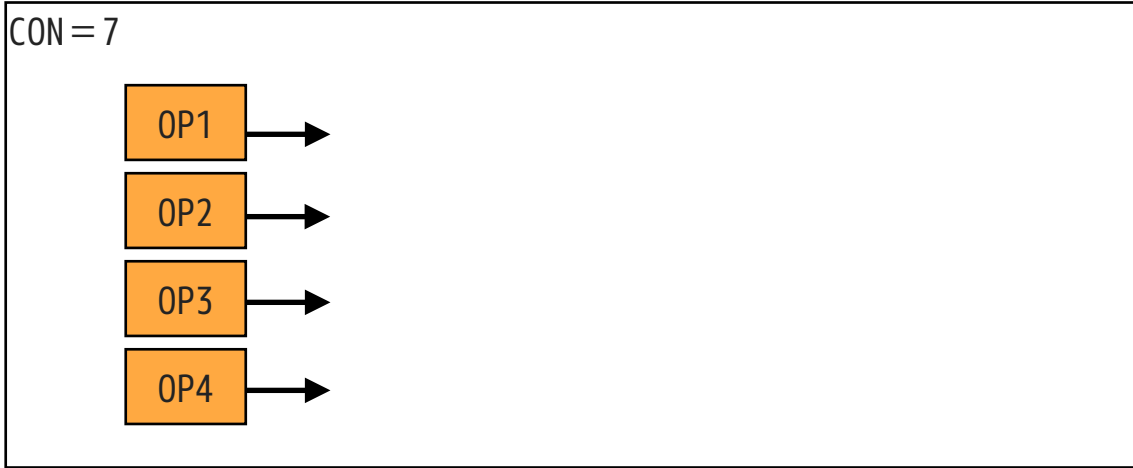
における、CON (connection: オペレータの接続パターン) の内容を示します。

色付きのオペレータがキャリアで、それ以外がモジュレータです。キャリアが複数あるCONでは、コーラスや和音の効果を出すことも可能です。全てのオペレータにセルフフィードバック機能があります。

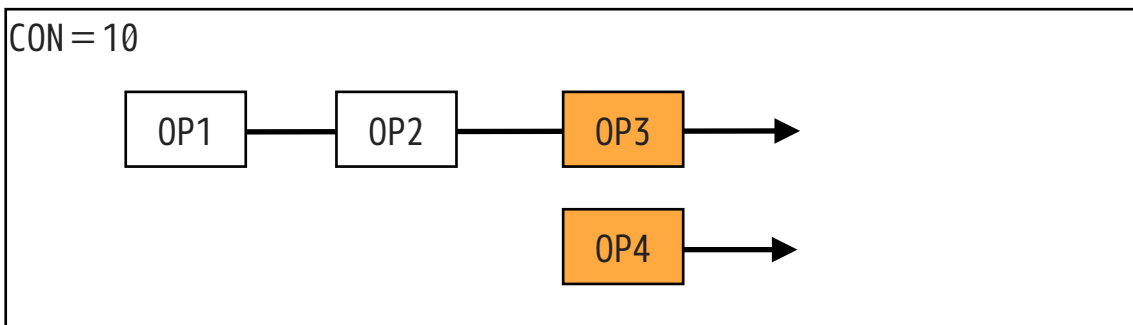
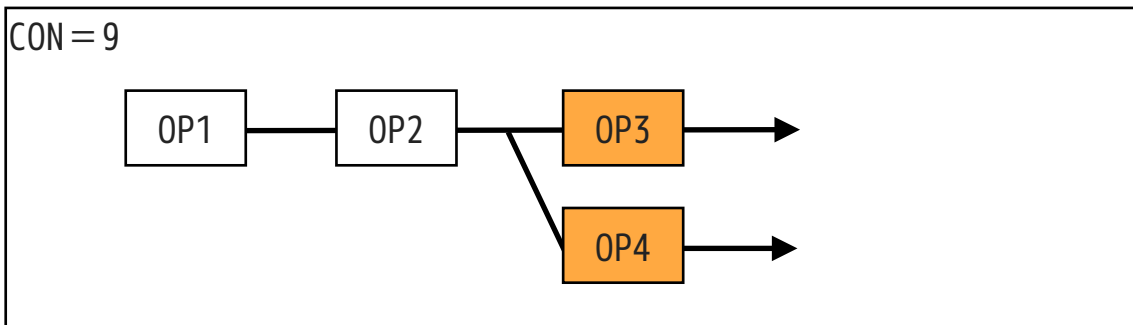
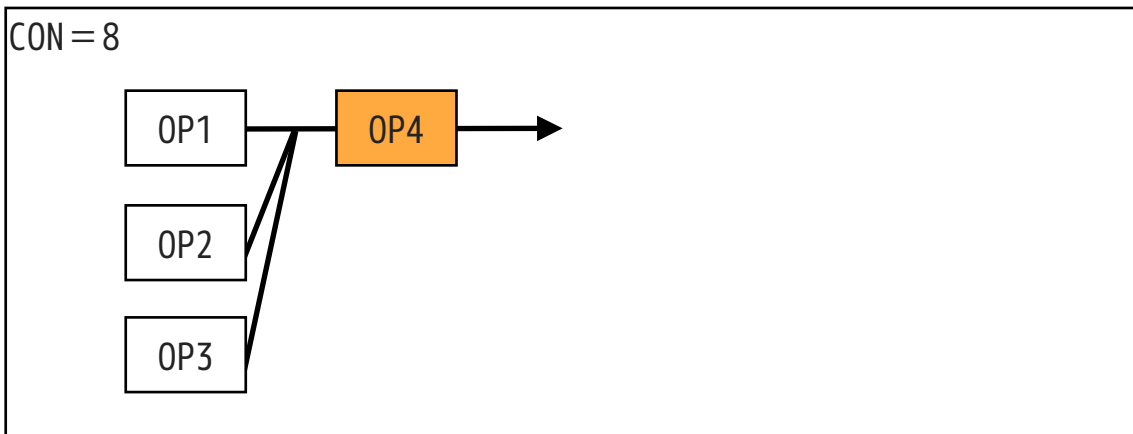
4オペレータモードのCONは、0~13で指定します。
 (通常0~7、拡張8~13)



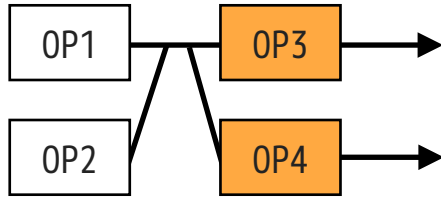




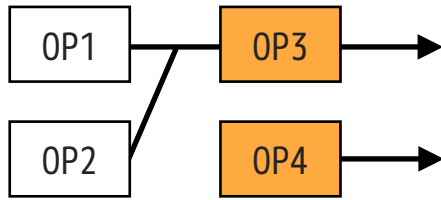
以下、独自拡張の接続形態（8～13）：



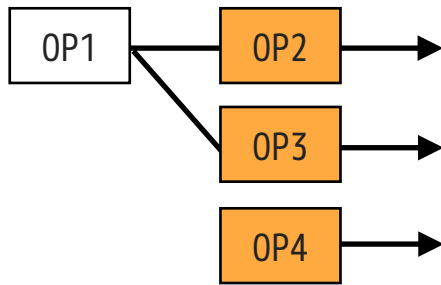
CON = 11



CON = 12



CON = 13



6.52. FM音源2オペレータモードの接続形態

4 オペレータモードの音色設定

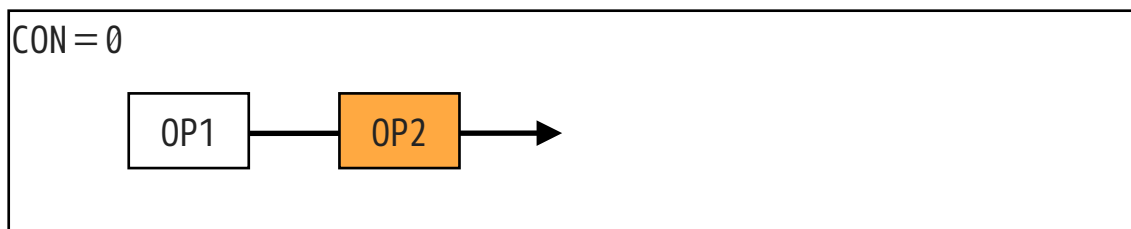
#MB:FM@SS20P (2オペレータType-SS)

における、CON (connection: オペレータの接続パターン) の内容を示します。

色付きのオペレータがキャリアで、それ以外がモジュレータです。キャリアが複数あるCONでは、コーラスや和音の効果を出すことも可能です。

全てのオペレータにセルフフィードバック機能があります。

2オペレータモードのCONは、0~1で指定します。



6.53. #ML:DPCM:@@DPC用：波形データ定義

記述	#ML:DPCM [1],[2],[3],[4],[5]
種別	メタデータ定義

【解説】

DPCM音源 (@@DPC) の波形データを定義します。

指定する引数は次の通りです。

引数[1]...波形番号 (waveNo)

引数[2]...初期変位 (initLv)

引数[3]...ループ再生フラグ (loopFg)

引数[4]...文字列のデコードモード (decMode)

引数[5]...波形データ (DPCMDATA)

引数はカンマで区切って指定します。

引数[1] (waveNo)

波形番号を整数で指定します。

指定可能範囲は 0 ~ 63 です。

この番号は、@@DPC のサブモジュール番号 (@番号) で使用します。

デフォルトで波形番号 0番 に無音の初期データが入っています (上書可)。

MML で未定義の波形番号を参照した場合は、0番 が読み出されます。

引数[2] (initLv)

DPCMの初期変位です。

initLv は DPCM差分計算の初期値になります。

設定範囲は 0 ~ 127 です。

変位が 0 となる中心値は 64 です。

引数[3] (loopFg)

ループ再生フラグです。ループ再生の可否を指定します。

0 のとき、ワンショット再生。

1 のとき、波形先頭からループ再生。

引数[4] (decMode)

DPCMDATA の文字列デコードモードを設定します。

0 のとき、プレーンテキストとして解釈します。

1 のとき、base64URLエンコードされた文字列として解釈します。

引数[5] (DPCMDATA)

DPCM波形データを記述します。

波形定義終了は改行によって検知されます。

DPCM波形データ形式はファミコン内蔵音源互換です。

8サンプル1バイトにパックされたデータを、そのまま16進数文字列で表記したプレーンテキスト形式と、base64URLエンコードした形式のいずれかで記述します。

【波形データ定義例】

```
#ML:DPCM 0,64,0,1,mA==
```

```
#ML:DPCM 0,64,0,1,mA==
```

```
#ML:DPCM 7,64,0,1,mA==
```

```
#ML:DPCM 15,64,0,0,98
```

1バイト分のDPCM (\$98) を、同内容で波形番号 0、7、15 に定義。

【備考1】

initLvは unsigned 7bit で受け付けますが、

再生時は下位 1bit を無視した 64段階で変位が変更されます。

【備考2】

DPCMバイナリのデコード処理では、各バイトのLSBから 1bit ずつ読み、0 なら変位を1段階下げ、1 なら変位を1段階上げています。

【備考3】

変位の内部管理は -1.0 ~ +1.0 の浮動小数点数で行われており、これに整数値の変位を割り当てています。割り当て内容は次の通り。

変位最低の -1.0 には、0 が割り当たります。

変位中心の 0.0 には、64 が割り当たります。

変位最高の +1.0 には割り当てはなく、整数最大の127 は 約 +0.98 に割り当たります。

6.54. #ML:U8PCM : @@PCM用 : 波形データ定義

記述	#ML:U8PCM [1],[2],[3],[4],[5],[6]
種別	メタデータ定義

【解説】

PCM音源 (@@PCM) の波形データを定義します。
データ形式は Unsigned 8bit PCM です。

指定する引数は次の通りです。

引数[1]...波形番号 (waveNo)

引数[2]...サンプリング周波数 (sFreq)

引数[3]...割当先の音程番号 (sKey)

引数[4]...ループ再生ポイント (loopPt)

引数[5]...文字列のデコードモード (decMode)

引数[6]...波形データ (PCMDATA)

引数はカンマで区切って指定します。

引数[1] (waveNo)

波形番号を整数で指定します。

指定可能範囲は 0 ~ 1023 です。

この番号は、@@PCM のサブモジュール番号 (@番号) で使用します。

デフォルトで波形番号 0番 に無音の初期データが入っています (上書可)。

MML で未定義の波形番号を参照した場合は、0番 が読み出されます。

引数[2] (sFreq)

PCMDATA のサンプリング周波数を指定します。少数点以下の指定も受け付けます。

引数[3] (sKey)

sFreq をどの音程番号に割り当てるかを指定します。(音程番号とは)

例えば、sFreq=8000、sKey=48 の場合、o4c でサンプリング周波数 8kHz 相当の再生になります。

引数[4] (loopPt)

ループ再生ポイントを何サンプル目にするかを指定します。

-1 のとき、ループ再生を無効にします。

0 のとき、先頭サンプルからループ再生します。

ループ有効時の設定可能範囲は、0 ~ (登録サンプル数-1) です。

引数[5] (decMode)

PCMDATA の文字列デコードモードを設定します。

0 のとき、プレーンテキストとして解釈します。

1 のとき、base64URLエンコードされた文字列として解釈します。

引数[6] (PCMDATA)

波形データを記述します。波形定義終了は改行によって検知されます。

波形データ形式は Unsigned 8bit PCM です。

1 サンプル1バイトのデータをそのまま16進数で表記したプレーンテキスト形式と、base64URLエンコードした形式のいずれかで記述します。

【波形データ定義例】

```
#ML:U8PCM 0,8000,48,0,0,ffffffff00000000
```

【備考】

変位の内部管理は -1.0 ~ +1.0 の浮動小数点数で行われており、これに整数値の変位を割り当てています。

変位最低の -1.0 には、0x00 が割り当たります。

変位最高の +1.0 には、0xFF が割り当たります。

変位中心は 0x7F と 0x80 の中間なので割り当てはありません。

6.55. #ML:ADPCM : @@ADP用 : 波形データ定義

記述	#ML:ADPCM [1],[2],[3],[4],[5],[6],[7]
種別	メタデータ定義

【解説】

ADPCM音源 (@@ADP) の波形データを定義します。
データ形式は 4bit ADPCM です。

指定する引数は次の通りです。

引数[1]...波形番号 (waveNo)

引数[2]...サンプリング周波数 (sFreq)

引数[3]...割当先の音程番号 (sKey)

引数[4]...ループ再生ポイント (loopPt)

引数[5]...文字列のデコードモード 1 (decMpde1)

引数[6]...文字列のデコードモード 2 (decMpde2)

引数[7]...波形データ (ADPCMDATA)

引数はカンマで区切って指定します。

引数[1] (waveNo)

波形番号を整数で指定します。

指定可能範囲は 0 ~ 1023 です。

この番号は、@@ADP のサブモジュール番号 (@番号) で使用します。

デフォルトで波形番号 0番 に無音の初期データが入っています (上書可)。

MML で未定義の波形番号を参照した場合は、0番 が読み出されます。

引数[2] (sFreq)

ADPCMDATA のサンプリング周波数を指定します。少数点以下の指定も受け付けます。

引数[3] (sKey)

sFreq をどの音程番号に割り当てるかを指定します。(音程番号とは)

例えば、sFreq=8000、sKey=48 の場合、o4c でサンプリング周波数 8kHz 相当の再生になります。

引数[4] (loopPt)

ループ再生ポイントを何サンプル目にするかを指定します。

-1 のとき、ループ再生を無効にします。

0 のとき、先頭サンプルからループ再生します。

ループ有効時の設定可能範囲は、0 ~ (登録サンプル数-1) です。

引数[5] (decMode1)

ADPCMDATA の文字列デコードモード (1) を設定します。

0 のとき、YM2608-ADPCM 互換形式として解釈します。

1 のとき、OKI-ADPCM 互換形式として解釈します。

引数[6] (decMode2)

ADPCMDATA の文字列デコードモード (2) を設定します。

0 のとき、プレーンテキストとして解釈します。

1 のとき、base64URL エンコードされた文字列として解釈します。

引数[7] (ADPCMDATA)

ADPCM波形データを記述します。

波形定義終了は改行によって検知されます。

ADPCM波形データ形式は 4bit ADPCM です。

1 サンプル 4 ビットのデータをそのまま 16 進数で表記したプレーンテキスト形式と、base64URLEncodeした形式のいずれかで記述します。

【波形データ定義例】

```
#ML:ADPCM 0,8000,48,0,0,0,8080808080808080
```

【備考】

ADPCMデータ受け取り後は、内部で Signed 16bit PCM に展開して保持します。

YM2608-ADPCM 互換形式では 16bit に展開され、

OKI-ADPCM 互換形式では 12bit に展開されます。

変位の内部管理は -1.0 ~ +1.0 の浮動小数点数で行われており、これに整数値の変位を割り当てています。

変位最低の -1.0 には、最小の整数値が割り当たります。

変位中心の 0.0 には、0 が割り当たります。

変位最高の +1.0 には割り当てが無く、

最大の整数値は、16bit展開時は 32767/32768、

12bit展開時は 2047/2048 に割り当たります。

6.56. #ML:EXBIN@FILE：バイナリファイル読み込み

記述 #ML:EXBIN@FILE [1]

種別 メタデータ定義

【解説】

一時バイナリ領域へ、指定のファイルを読み込みます。

一時バイナリ領域とは、メタデータ定義のために一時的にバイナリデータを保持するメモリ領域で、メタデータ定義が全て終了すると破棄されます。

読み込むファイルは、あらかじめ定められたフォーマットのバイナリデータです。（フォーマットとは）

当メタデータ定義は、1回だけ記述が可能で、複数回記述した場合には、2回目以降がエラーとなります。

引数[1] (fileName)

読み込むファイル名前を文字列で指定します。

読み込み対象になる場所は、プレイリストのカレントフォルダ配下です。

ファイル名の拡張子に制限は設けていませんが、

現状、標準の拡張子は「.v2b」です。

プレイリストのカレントフォルダは、PlayListウインドウの最下部に書かれています。

【例】

```
#ML:EXBIN@FILE v2mmlBinary.v2b
```

【備考】

当メタデータ定義は、プレイリスト内にあるMMLファイルを選択して演奏開始したとき、それが置かれた場所にあるバイナリファイルを読み込む動作を想定しています。

そのため、「開く…」から読み込む場合や、コピーペーストでMMLウインドウに貼り付けて演奏開始する場合などは、カレントフォルダの食い違いが発生する恐れがあるため注意が必要です。

6.57. 読み込むバイナリファイルのフォーマット

メタデータ定義のために一時バイナリ領域に読み込むファイルのフォーマットは、大きく分けて3つのチャンクから成り立っています。

- 第1 チャンク：ヘッダー
- 第2 チャンク：アロケーションテーブル
- 第3 チャンク：実データ群

また、バイナリデータファイル全体のサイズは2ギガバイトまでです。上限を超えるサイズの場合、演奏開始で読み込む際にエラーとなります。

第1 チャンク詳細（原則32バイト）

- (1)文字列16バイト（終端なし固定サイズ）：
 - 当ファイルの認識用ヘッダー。次のように定義。
V2MML-bin-XXXXXX
 - ヘッダー確認は現状先頭10文字で行われます。
 - 末尾6文字（XXXXXX）はデータフォーマット番号で、更新される可能性があります。
- (2)整数4バイト(big endian)：
 - ファイル先頭から第2チャンク先頭までのオフセット。
 - 原則 \$00000020 です。
- (3)整数4バイト(big endian)：
 - ファイル先頭から第3チャンク先頭までのオフセット。
 - 第2チャンクでの定義数に応じて変化します。
- (4)予約8バイト：
 - 拡張用の予約領域。

第2 チャンク詳細（サイズ可変）

データアロケーション1個あたり16バイト定義。
アロケーション個数は(2)と(3)から算出。

データアロケーション1個あたりの詳細

- (5)整数4バイト(big endian)：
 - 当アロケーションの開始位置。(3)からのオフセット。
- (6)整数4バイト(big endian)：
 - 当アロケーションのバイト数。
 - バイト数が0の場合は未定義とみなします。
 - (サンプル数ではなく、バイト数であることに注意)

(7) 予約 4 バイト :

拡張用の予約領域。

(8) 文字列 4 バイト (終端なし固定サイズ) :

当アロケーションの種別を表す文字列。

これはバインドされるファイルの拡張子で、
ドットで始まる 4 文字です。(例「.16l」)

(8)における現在のサポート拡張子 (7 種類) :

.16l (いち・ろく・える)

signed 16bit PCM、リトルエンディアン、モノラルデータです。

.16b (いち・ろく・びー)

signed 16bit PCM、ビッグエンディアン、モノラルデータです。

.u08 (ゆー・ぜろ・はち)

unsigned 8bit PCM、モノラルデータです。

.s08 (えす・ぜろ・はち)

signed 8bit PCM、モノラルデータです。

.ayb (えー・わい・びー)

YM2608互換 4bit ADPCM、ニブルの順番は上位から下位、モノラルデータです。

.amb (えー・えむ・びー)

OKI互換 4bit ADPCM、ニブルの順番は上位から下位、モノラルデータです。

.aml (えー・えむ・える)

OKI互換 4bit ADPCM、ニブルの順番は下位から上位、モノラルデータです。

第3チャンク詳細 (サイズ可変)

バイナリファイルの内容そのものが貼り付けられます。

各ファイル内容の開始位置は(5)、サイズ (バイト数) は(6)の通りです。

6.58. #ML:SMP@EXBIN：一時バイナリ領域からの波形データ定義

記述	#ML:SMP@EXBIN [1],[2],[3],[4],[5],[6]
種別	メタデータ定義

【解説】

「#ML:EXBIN@FILE」でファイルを読み込んだ結果の、一時バイナリ領域から、PCM音源 (@@PCM) または ADPCM音源 (@@ADP) に対し、波形データを定義します。

指定する引数は次の通りです。

引数[1]...ピックアップ番号 (pickUpNo)

引数[2]...データ形式 (dataType)

引数[3]...波形番号 (waveNo)

引数[4]...サンプリング周波数 (sFreq)

引数[5]...割当先の音程番号 (sKey)

引数[6]...ループ再生ポイント (loopPt)

引数はカンマで区切って指定します。

当コマンドの記述より前に、「#ML:EXBIN@FILE」のファイル読み込みコマンドを記述しておく必要があります。

波形データのサンプル数の上限は、現状12メガサンプルです。これを超えると12メガに制限されます。

引数[1] (pickUpNo)

一時バイナリ領域からピックアップするデータ番号を、整数で指定します。

データ番号は、アロケーションテーブルの先頭が0で、以後1ずつ増える番号になります。

設定範囲は、アロケーションテーブルに定義されているデータ数の範囲内です。

引数[2] (dataType)

一時バイナリ領域から読み込もうとするデータ形式を、キーワードとなる文字列で指定します。

キーワードは次の通りです。

キーワード	内容
PCM_S16L	@@PCM 宛の、signed 16bit PCM、リトルエンディアン、モノラルデータ
PCM_S16B	@@PCM 宛の、signed 16bit PCM、ビッグエンディアン、モノラルデータ
PCM_U8	@@PCM 宛の、unsigned 8bit PCM、モノラルデータ
PCM_S8	@@PCM 宛の、signed 8bit PCM、モノラルデータ
ADP_YMB	@@ADP 宛の、YM2608互換 4bit ADPCM、ニブルの順番は上位から下位、モノラルデータ
ADP_MSMB	@@ADP 宛の、OKI互換 4bit ADPCM、ニブルの順番は上位から下位、モノラルデータ
ADP_MSML	@@ADP 宛の、OKI互換 4bit ADPCM、ニブルの順番は下位から上位、モノラルデータ

引数[3] (waveNo)

波形番号を整数で指定します。

指定可能範囲は 0 ~ 1023 です。

この番号は、@@PCM / @@ADP のサブモジュール番号 (@番号) で使用します。デフォルトで波形番号 0番 に無音の初期データが入っています (上書可)。MML で未定義の波形番号を参照した場合は、0番 が読み出されます。

引数[4] (sFreq)

pickUpNoに対応する波形データのサンプリング周波数を指定します。少数点以下の指定も受け付けます。

引数[5] (sKey)

sFreq をどの音程番号に割り当てるかを指定します。(音程番号とは)

例えば、sFreq=8000、sKey=48 の場合、o4c でサンプリング周波数 8kHz 相当の再生になります。

引数[6] (loopPt)

ループ再生ポイントを何サンプル目にするかを指定します。

-1 のとき、ループ再生を無効にします。

0 のとき、先頭サンプルからループ再生します。

ループ有効時の設定可能範囲は、0 ~ (登録サンプル数-1) です。

【波形データ定義例】

```
#ML:EXBIN@FILE v2mmlPcm.v2b
#ML:SMP@EXBIN 1,PCM_S16L, 11,48000,48,-1
#ML:SMP@EXBIN 2,PCM_S16B, 12,48000,48,-1
#ML:SMP@EXBIN 3,PCM_U8, 13,16500,48,-1
#ML:SMP@EXBIN 4,PCM_S8, 14,16500,48,-1
#ML:SMP@EXBIN 5,ADP_YMB, 15,32000,48,-1
#ML:SMP@EXBIN 6,ADP_MSMB, 16,32000,48,-1
#ML:SMP@EXBIN 7,ADP_MSML, 17,32000,48,-1
```

ファイル「v2mmlPcm.v2b」を読み込み、その中に定義されている波形データを、音源モジュールに設定しています。

7. エンベロープ関連

7.1. @E0 : エンベロープオプション

記述	@E0[1],[2],[3],[4]
種別	MML コマンド

【解説】

エンベロープ (@EA,@EF) への、オプション設定を行います。

指定する引数は次の通りです。

引数[1]...エンベロープオプションの種類 (type)

引数[2]...時間単位 (unit)

引数[3]...時間単位への倍率の分子 (num)

引数[4]...時間単位への倍率の分母 (denom)

引数はカンマで区切って指定します。

引数[1] (type)

エンベロープオプションの種類を、英字 1 文字で指定します。

(英字は大文字・小文字を問わず)

「C」を指定した場合

エンベロープクロックの設定が対象になります。

エンベロープクロックは、エンベロープの 1 節での経過時間の単位です。

「#ML:ENVCLOCK」による設定を変更したい場合に使用します。

「R」を指定した場合

エンベロープ解像度の設定が対象になります。

エンベロープ解像度は、振幅計算更新の最少時間の単位です。

「#ML:ENVRESOL」による設定を変更したい場合に使用します。

引数[2] (unit)

時間単位を、1文字で指定します。
(英字の場合は大文字・小文字を問わず)

「S」を指定した場合

時間単位は「1秒」になります。
有効範囲は、1[秒] ~ 1/2400[秒]です。

「%」を指定した場合

時間単位は「1 tickカウント」になります。

「=」を指定した場合

「=」は、type に「R」を指定した場合にのみ使える指定です。
「=」を指定した場合、エンベロープ解像度が最大解像度となり、
時間単位は再生サンプリング周波数の1サンプルになります。
「=」を指定した場合、num と denom の指定は出来ません。

引数[3] (num)

unit によって指定した時間単位の倍率の分子を指定します。
小数以下の指定も受け付けます。

引数[4] (denom)

unit によって指定した時間単位の倍率の分母を指定します。
小数以下の指定も受け付けます。

【備考1】

num と denom の組は省略することが出来ます。
省略した場合、時間単位の倍率は等倍 (1/1倍) とみなされます。

【備考2】

時間単位が「1 tickカウント」の場合、テンポにより時間単位が変動します。
エンベロープ指定後にテンポを変更した場合、テンポ変更後のノートオンからエンベロープの時間計算が自動追従します。
つまり、ノートオン中にテンポ変更が掛かった場合は、そのノートオン中は時間計算の自動追従はできませんが、次のノートオンから自動追従します。

【例1】

@EOC,S,7,300

エンベロープクロック = $7/300$ [秒]

【例2】

@EOC,%,3,7

エンベロープクロック = 1 tickカウント × $3/7$

【例3】

@EOR,=

エンベロープ解像度 = 最大解像度

【例4】

@EOR,S,1,600

エンベロープ解像度 = $1/600$ [秒]

【例5】

@EOR,%,1,2

エンベロープ解像度 = 1 tickカウント × $1/2$

7.2. @EA：音量エンベロープ

記述	@EA[1]
種別	MML コマンド

【解説】

「#MB:ENV@A」で定義された音量エンベロープ情報を呼び出して適用します。トラック先頭におけるデフォルトは、システム側に用意された暫定内容の音量エンベロープが掛かっています。

引数[1]には、音量エンベロープ番号を整数で指定します。未定義の番号を指定すると、エラーになります。

音量エンベロープでは、後述するメタデータ定義「#MB:ENV@A」によって、定義内容に音量エンベロープ番号を割り当てて登録したものを使用します。

【注】 @@FMS（FM音源）には、@EAコマンドは無効です。
@@FMSでは音色データ定義内のエンベロープを使用します。

【例】

```
#MB:ENV@A 1 { 15,0,&, N,0,15, R,0,0 }  
@@PLS @EA1 o4 L8 CRDRER;
```

7.3. @EF：フィルタエンベロープ

記述	@EF[1]
種別	MML コマンド

【解説】

「#MB:ENV@F」で定義されたフィルタエンベロープ情報を呼び出して適用します。
@EF コマンドは、フィルタ機能 (@F) に依存しています。
そのため、トラック先頭においては、@F コマンドが無効につき、@EF コマンドも無効です。

引数[1]には、フィルタエンベロープ番号を整数で指定します。
未定義の番号を指定すると、エラーになります。

フィルタエンベロープでは、後述するメタデータ定義「#MB:ENV@F」によって、定義内容にフィルタエンベロープ番号を割り当てて登録したものを使用します。

【例】

```
#ML:FILTER 1,2,100,50,0
#MB:ENV@F 1 { 100,0,0, N,12,100, N,120,0, R,60,0 }
@@PLS @F1 @EF1
L2 04 CDEFGAB>C<;
```

7.4. ZS：音量エンベロープのダンパー設定

記述	ZS[1],[2]
種別	MML コマンド

【解説】

音量エンベロープの、ダンパー（消音機能）設定を行います。

当コマンドは、@Zコマンドで実行されるダンパー設定を変更したい場合に利用します。

トラック先頭におけるデフォルト設定は、「#ML:ENV DAMPD」「#ML:FMS DAMPD」による設定に従います。

指定する引数は次の通りです。

引数[1]...レート (rate)

引数[2]...宛先 (destination)

引数はカンマで区切って指定します。

引数[1] (rate)

音量エンベロープのダンパーレートを指定します。

destination が「ENV」のとき

rateは 0 以上の数値で指定します。小数以下の指定も受け付けます。

指定値はエンベロープ定義内における変化率（I のモード）によるレートとして受け付けられます。減衰時間はエンベロープクロック（時間単位）の設定に依存します。

destination が「FMS」のとき

rateは 0 ~ 16 の整数で指定します。

指定値が 0 ~ 15 だった場合、ダンパー実行時に、キャリアのリリースレートだけが、指定値に書き換えられます。

指定値が 16 だった場合、ダンパー実行時に、全てのオペレータのリリースレートが、15に書き換えられます。

引数[2] (destination)

エンベロープダンパー設定の宛先を、3文字の識別子で指定します。

【@ZS宛先の識別子一覧】

ENV	音量エンベロープ (@EA)
FMS	F M音源 (@@FMS) のエンベロープ

(識別子は大文字・小文字を問わず)

【例1】

ZS0.03,ENV

音量エンベロープのダンパーレートを 0.03 にします。

【例2】

ZS12,FMS

F M音源 (@@FMS) のエンベロープのダンパーレートを 12 にします。

7.5. @Z : エンベロープダンパー実行

記述	@Z
種別	MML コマンド

【解説】

エンベロープのダンパー（消音機能）を実行します。

引数はありません。

「#ML:ENV DAMPD」「#ML:FMS DAMPD」「@ZS コマンド」などで設定済みのレートにより、消音が実行されます。

当コマンドは、音量エンベロープ (@EA) または、FM音源 (@@FMS) エンベロープのうち、演奏中であるリリースを臨時的に書き換えて、即時無音までリリースを終結させる用途で使用します。書き換えたリリースは、次回ノートオンの際、自動的に元のレートに戻します。

例えば、ポリフォニックモードを有効にし、リリースも長く設定して、サスティンペダルを踏んでいるような状態のときに、ブレイクの休符前などに @Z を配置することで、即時消音（サスティンペダルを放す）したいときに使用します。

スラー中に @Z コマンドを使用した場合は、強制的にスラーが中断（ノートオフ）されて、@Z コマンド直後の音符があらためてノートオンされます。

7.6. #MB:ENV@A : 音量エンベロープデータ定義

記述	#MB:ENV@A [1] {[2]}
種別	メタデータ定義

【解説】

音量エンベロープのデータを定義します。

指定する引数は次の通りです。

引数[1]...音量エンベロープ番号 (envNum) 【1個の整数】

引数[2]...音量エンベロープデータ (envParam) 【複数のパラメータ】

引数[1] (envNum)

音量エンベロープ番号を、整数で指定します。

この番号は、@EAコマンドの音量エンベロープ番号指定で使用します。

設定範囲は 0 ~ 1023 です。

引数[2] (envParam)

音量エンベロープデータのパラメータ群 (envParam) の記述開始は

「 { 」で認識され、終了は「 } 」によって検知されます。

各パラメータは、カンマで区切って記述します。

スペースや改行は読み飛ばされます。

envParamは、

3 + (3 * n) 個

のパラメータ群で構成されています (n は 2 以上、2048 未満)。

パラメータ群の最初の3個は、ヘッダとして定義します。

以降は3個1組のパラメータセットを、ボディとして2セット以上定義します。

2セットとは、最低でもエンベロープ開始時点の節と、リリース開始時点の節が必要なためです。

以下、パラメータ群の各要素について説明します。

音量エンベロープのパラメータ要素：

ヘッダ部（3個）

[A] levelDenom

[B] levelOffset

[C] initLevel

ボディ部（3個×2セット以上）

[D] pointerMode

[E] rate,rateMode

[F] level

[A] levelDenom

levelDenomは、音量エンベロープの出力レベルの最大値を、0 より大きい値で指定します。小数以下の指定も受け付けます。出力レベルの計算結果は、0 ~ 1 の浮動小数点数で内部管理されています。そこで、データ定義内で指定される出力レベル値を、levelDenomで割って、0 ~ 1 に変換して使っています。ただし、出力レベルの計算結果は、 $(\text{level} + \text{levelOffset}) \div \text{levelDenom}$ の結果が 1 を超える場合は 1 に、0 を下回る場合は 0 に制限されます。

[B] levelOffset

levelOffsetは、音量エンベロープの出力レベルに対するオフセット値を指定します。オフセット値の加算は、levelDenom による除算を行う前に実行されます。

音量エンベロープでは、リリース中の無限演奏状態を避けるため、リリース最終節のレベルをゼロにしなければならない制限があります。そのため、オフセット値を 0 以外にした場合、オフセット値×(-1)のレベルでリリース最終節を終わらせる必要がある点に注意してください。

[C] initLevel

initLevelは、エンベロープ開始時点の出力レベルを、数値または &（アンパサンド）で指定します。

& を指定したとき：

ノートオン時の初期レベルは、直前までの演奏による出力レベルを引き継いだ値になります。

数値を指定したとき：

ノートオン時に、指定値に出力レベルを設定します。

数値の指定範囲は、0 から levelDenom 設定の範囲内になります。小数以下の指定も受け付けます。

[D] pointerMode

pointerModeは、エンベロープの節の種類を N、L、R の3択で指定します。

N：通常のエンベロープの節

L：ループエントリのエンベロープの節

R：リリースエントリのエンベロープの節

L の節は、R より前の節で定義しなければなりません。

L の節は、1つのエンベロープに1回だけ使用できますが、0回でも構いません。

R の節は、1つのエンベロープに1回だけ、必ず使用しなければなりません。

L は R が現れる直前の節ポイントをサスティンの末尾と解釈することで、末尾の節からループエントリへ無条件ジャンプするポイントになりますが、ループエントリがサスティン末尾であった場合、ループリクエストは無視されます。（自分自身にループジャンプはしません）

[E] rate,rateMode

rateは、今回のエンベロープの節における変化レートを、0 以上の数値で指定します。小数以下の指定も受け付けます。

変化レートの数値は、先頭1文字に rateMode によるモード指定を記述することで、2種類の解釈がなされます。

rateModeは、

rate の解釈について、英字1字でモード指定します。

（大文字・小文字問わず）

rateMode は省略可能で、省略した場合、T を指定したものとみなされます。

T（ティー）を指定したとき：

rate の解釈を、時間とします。

変化時間を固定的に指定します。

時間の単位は「#ML:ENVCLOCK」に依存します。

このモードでは、ディケイ時など、初期レベルと到達レベルの差が毎回固定的な場合は変化の傾きが一定になりますが、リリース時など、初期レベルと到達レベルの差が一定でない場合は、時間が固定される都合、変化の傾きが変わります。

I（アイ）を指定したとき：

rate の解釈を、変化率（勾配:incline）とします。

時間当たりの変化量を、最大変移量 ÷ rate で求めることで、固定化された変化量を設定できます。そのため、リリース時など、初期レベルと到達レベルの差が一定でない場合は、変化の傾きが固定される都合、時間が変わります。rate の値が大きいほどゆっくり推移します。0 のときは時間 0 で推移終了します。

【例 1】

I3

変化率 3 の記述

【例 2】

10

時間 10 の記述。この例では T が省略されている状態。

[F] level

level は、今回のエンベロープの節における到達レベルを数値で指定します。数値の指定範囲は、0 から levelDenom 設定の範囲内になります。小数以下の指定も受け付けます。

前回のレベルと今回のレベルが同じ場合で、rate が I0 または T0 の場合、時間 0 で何もせず次の節に移ります。

【音量エンベロープ定義例】

```
#MB:ENV@A 1 {
//  lv ofs init
    15,0,&,          /* (1) */
//  pt rate lv
    N,i3,15,        /* (2) */
    N,100,10,       /* (3) */
    N,10,12,        /* (4) */
    L,10,8,         /* (5) */
    N,10,12,        /* (6) */
    R,i30,0         /* (7) */
}
@ea1 l1
o4 cdefg;
```

エンベロープ定義例の動作解説：

- (1) レベルは0～15、ノートオン後の初期レベルは残存引継ぎでスタート
- (2) 変化率3でレベル15まで変化
- (3) 時間100でレベル10まで変化
- (4) 時間10でレベル12まで変化
- (5) (ループエントリ) 時間10でレベル8まで変化
- (6) 時間10でレベル12まで変化
- (7) (5)へジャンプ

この流れのどこでノートオフになるかは不定ですが、ノートオフ時から変化率30でレベル0になります。

7.7. #MB:ENV@F : フィルタエンベロープデータ定義

記述	#MB:ENV@F [1] {[2]}
種別	メタデータ定義

【解説】

フィルタエンベロープのデータを定義します。

指定する引数は次の通りです。

引数[1]...フィルタエンベロープ番号 (envNum) 【1個の整数】

引数[2]...フィルタエンベロープデータ (envParam) 【複数のパラメータ】

引数[1] (envNum)

フィルタエンベロープ番号を、整数で指定します。

この番号は、@EFコマンドのフィルタエンベロープ番号指定で使用します。

設定範囲は 0 ~ 1023 です。

引数[2] (envParam)

フィルタエンベロープデータのパラメータ群 (envParam) の記述開始は

「 { 」で認識され、終了は「 } 」によって検知されます。

各パラメータは、カンマで区切って記述します。

スペースや改行は読み飛ばされます。

envParamは、

$3 + (3 * n)$ 個

のパラメータ群で構成されています (n は 2 以上、2048 未満)。

パラメータ群の最初の3個は、ヘッダとして定義します。

以降は3個1組のパラメータセットを、ボディとして2セット以上定義します。

2セットとは、最低でもエンベロープ開始時点の節と、リリース開始時点の節が必要なためです。

以下、パラメータ群の各要素について説明します。

フィルタエンベロープのパラメータ要素：

ヘッダ部（3個）

[A] levelDenom

[B] levelOffset

[C] initLevel

ボディ部（3個×2セット以上）

[D] pointerMode

[E] rate,rateMode

[F] level

[A] levelDenom

levelDenomは、フィルタエンベロープの出力レベル（カットオフのレベル）の最大値を、0 より大きい値で指定します。小数以下の指定も受け付けます。出力レベルの計算結果は、0 ~ 1 の浮動小数点数で内部管理されています。そこで、データ定義内で指定される出力レベル値を、levelDenomで割って、0 ~ 1 に変換して使っています。

ただし、出力レベルの計算結果は、

$$(\text{level} + \text{levelOffset}) \div \text{levelDenom}$$

の結果が 1 を超える場合は 1 に、0 を下回る場合は 0 に制限されます。

[B] levelOffset

levelOffsetは、音量エンベロープの出力レベルに対するオフセット値を指定します。オフセット値の加算は、levelDenom による除算を行う前に実行されます。

フィルタエンベロープでは、音量エンベロープと違い、リリース最終節のレベルをゼロにする必要はありません。

[C] initLevel

initLevelは、エンベロープ開始時点の出力レベルを、数値または &（アンパサンド）で指定します。

& を指定したとき：

ノートオン時の初期レベルは、直前までの演奏による出力レベルを引き継いだ値になります。

数値を指定したとき：

ノートオン時に、指定値に出力レベルを設定します。

数値の指定範囲は、0 から levelDenom 設定の範囲内になります。小数以下の指定も受け付けます。

[D] pointerMode

pointerModeは、エンベロープの節の種類を N、L、R の3択で指定します。

N：通常のエンベロープの節

L：ループエントリのエンベロープの節

R：リリースエントリのエンベロープの節

L の節は、R より前の節で定義しなければなりません。

L の節は、1つのエンベロープに1回だけ使用できますが、0回でも構いません。

R の節は、1つのエンベロープに1回だけ、必ず使用しなければなりません。

L は R が現れる直前の節ポイントをサスティンの末尾と解釈することで、末尾の節からループエントリへ無条件ジャンプするポイントになりますが、ループエントリがサスティン末尾であった場合、ループリクエストは無視されます。（自分自身にループジャンプはしません）

[E] rate,rateMode

rateは、今回のエンベロープの節における変化レートを、0以上の数値で指定します。小数以下の指定も受け付けます。

変化レートの数値は、先頭1文字に rateMode によるモード指定を記述することで、2種類の解釈がなされます。

rateModeは、

rate の解釈について、英字1字でモード指定します。

（大文字・小文字問わず）

rateMode は省略可能で、省略した場合、T を指定したものとみなされます。

T（ティー）を指定したとき：

rate の解釈を、時間とします。

変化時間を固定的に指定します。

時間の単位は「#ML:ENVCLOCK」に依存します。

このモードでは、ディケイ時など、初期レベルと到達レベルの差が毎回固定的な場合は変化の傾きが一定になりますが、リリース時など、初期レベルと到達レベルの差が一定でない場合は、時間が固定される都合、変化の傾きが変わります。

I（アイ）を指定したとき：

rate の解釈を、変化率（勾配:incline）とします。

時間当たりの変化量を、最大変移量 ÷ rate で求めることで、固定化された変化量を設定できます。そのため、リリース時など、初期レベルと到達レベルの差が一定でない場合は、変化の傾きが固定される都合、時間が変わります。

rate の値が大きいほどゆっくり推移します。0 のときは時間 0 で推移終了します。

【例 1】

I3

変化率 3 の記述

【例 2】

10

時間 10 の記述。この例ではTが省略されている状態。

[F] level

levelは、今回のエンベロープの節における到達レベルを数値で指定します。数値の指定範囲は、0 から levelDenom 設定の範囲内になります。小数以下の指定も受け付けます。

前回のレベルと今回のレベルが同じ場合で、rate が I0 または T0 の場合、時間 0 で何もせず次の節に移ります。

【フィルタエンベロープ定義例】

```
#MB:ENV@F 1 {
//  lv  ofs  init
    100,  0,  0,      /* (1) */
//  pt  rate lv
    N,  15,  100,     /* (2) */
    N, 240,   0,      /* (3) */
    R,  60,   0,      /* (4) */
}
#ML:FILTER 0,2,100,40,0
@f0 @ef1 q13,16 l2
o4 cdefg;
```

エンベロープ定義例の動作解説：

- (1) レベルは0～100、ノートオン後の初期レベルは0でスタート
- (2) 時間15でレベル100まで変化
- (3) 時間240でレベル0まで変化
- (4) レベル0を維持

この流れのどこでノートオフになるかは不定ですが、
ノートオフ時から時間60でレベル0になります。

※フィルタエンベロープのレベルはリニアですが、カットオフに渡す際、対応するカットオフ周波数曲線に簡易変換しています。

7.8. #ML:ENVLOCK : エンベロープ時間単位

記述	#ML:ENVLOCK [1],[2],[3]
種別	メタデータ指定

【解説】

エンベロープ全種類への、エンベロープの時間単位を設定します。
この設定は、全トラックへの初期設定となります。

指定する引数は次の通りです。

引数[1]...時間単位 (unit)

引数[2]...時間単位の倍率の分子 (num)

引数[3]...時間単位の倍率の分母 (denom)

引数はカンマで区切って指定します。

当メタデータ指定を使用しなかった場合のデフォルト設定は次の通りです。

```
#ML:ENVLOCK S,1,127
```

引数[1] (unit)

時間単位を、1文字で指定します。

(英字は大文字・小文字を問わず)

「S」を指定した場合

時間単位は「1秒」になります。

有効範囲は、1[秒] ~ 1/2400[秒]です。

「%」を指定した場合

時間単位は「1 tickカウント」になります。

引数[2] (num)

unit によって指定した時間単位の倍率の分子を指定します。
小数以下の指定も受け付けます。

引数[3] (denom)

unit によって指定した時間単位の倍率の分母を指定します。
小数以下の指定も受け付けます。

【備考1】

num と denom の組は省略することが出来ます。
省略した場合、時間単位の倍率は等倍 (1/1倍) とみなされます。

【備考2】

時間単位が「1 tickカウント」の場合、テンポにより時間単位が変動します。
エンベロープ指定後にテンポを変更した場合、テンポ変更後のノートオンからエンベロープの時間計算が自動追従します。
つまり、ノートオン中にテンポ変更が掛かった場合は、そのノートオン中は時間計算の自動追従はできませんが、次のノートオンから自動追従します。

【例1】

```
#ML:ENVCLOCK S,7,300
```

エンベロープクロック = $7/300$ [秒]

【例2】

```
#ML:ENVCLOCK %,3,7
```

エンベロープクロック = 1 tickカウント × $3/7$

7.9. #ML:ENVRESOL : エンベロープ解像度

記述 #ML:ENVRESOL [1],[2],[3]

種別 メタデータ指定

【解説】

エンベロープ全種類への、エンベロープの解像度を設定します。
エンベロープ解像度とは、振幅計算更新の最小時間の単位です。
この設定は、全トラックへの初期設定となります。

指定する引数は次の通りです。

引数[1]...時間単位 (unit)

引数[2]...時間単位の倍率の分子 (num)

引数[3]...時間単位の倍率の分母 (denom)

引数はカンマで区切って指定します。

当メタデータ指定を使用しなかった場合のデフォルト設定は次の通りです。

#ML:ENVRESOL =

引数[1] (unit)

時間単位を、1文字で指定します。

(英字は大文字・小文字を問わず)

「S」を指定した場合

時間単位は「1秒」になります。

有効範囲は、1[秒] ~ 1/2400[秒]です。

「%」を指定した場合

時間単位は「1 tickカウント」になります。

「=」を指定した場合

エンベロープ解像度が最大解像度となり、

時間単位は再生サンプリング周波数の1サンプルになります。

「=」を指定した場合、num と denom の指定は出来ません。

引数[2] (num)

unit によって指定した時間単位の倍率の分子を指定します。
小数以下の指定も受け付けます。

引数[3] (denom)

unit によって指定した時間単位の倍率の分母を指定します。
小数以下の指定も受け付けます。

【備考1】

num と denom の組は省略することが出来ます。
省略した場合、時間単位の倍率は等倍 (1/1倍) とみなされます。

【備考2】

時間単位が「1 tickカウント」の場合、テンポにより時間単位が変動します。
エンベロープ指定後にテンポを変更した場合、テンポ変更後のノートオンからエンベロープの時間計算が自動追従します。
つまり、ノートオン中にテンポ変更が掛かった場合は、そのノートオン中は時間計算の自動追従はできませんが、次のノートオンから自動追従します。

【例1】

```
#ML:ENVRESOL =
```

エンベロープ解像度 = 最大解像度

【例2】

```
#ML:ENVRESOL S,1,600
```

エンベロープ解像度 = 1/600 [秒]

【例3】

```
#ML:ENVRESOL %,1,2
```

エンベロープ解像度 = 1 tickカウント × 1/2

7.10. #ML:ENV DAMPD : @EA用 : ダンパーデフォルト

記述	#ML:ENV DAMPD [1]
----	-------------------

種別	メタデータ指定
----	---------

【解説】

音量エンベロープダンパー (@EAコマンドの消音機能) の、全トラックへの初期値となるレートを設定します。

引数[1]には、0 以上の数値で指定します。小数以下の指定も受け付けます。指定値はエンベロープ定義内における変化率 (I のモード) によるレートとして受け付けられます。減衰時間はエンベロープクロック (時間単位) の設定に依存します。

当メタデータ指定を使用しなかった場合のデフォルト設定は次の通りです。

```
#ML:ENV DAMPD 0.03
```

7.11. #ML:FMSDAMPD : @@FMS用 : ダンパーデフォルト

記述	#ML:FMSDAMPD [1]
----	------------------

種別	メタデータ指定
----	---------

【解説】

F M音源のエンベロープダンパー（@@FMSの消音機能）の、全トラックへの初期値となるレートを設定します。

引数[1]は、0 ～ 16 の整数で指定します。

指定値が 0 ～ 15 だった場合

ダンパー実行時に、キャリアのリリースレートだけが、指定値に書き換えられます。

指定値が 16 だった場合

ダンパー実行時に、全てのオペレータのリリースレートが、15に書き換えられます。

当メタデータ指定を使用しなかった場合のデフォルト設定は次の通りです。

```
#ML:FMSDAMPD 16
```

8. L F O 関連

8.1. @L0 : L F O オプション

記述	@L0[1],[2],[3],[4]
種別	MML コマンド

【解説】

L F O (@LP,@LA,@LB,@LF,@LY) への、オプション設定を行います。

指定する引数は次の通りです。

引数[1]... L F O オプションの種類 (type)

引数[2]... 時間単位 (unit)

引数[3]... 時間単位への倍率の分子 (num)

引数[4]... 時間単位への倍率の分母 (denom)

引数はカンマで区切って指定します。

引数[1] (type)

L F O オプションの種類を、英字 1 文字で指定します。

(英字は大文字・小文字を問わず)

「C」を指定した場合

L F O クロックの設定が対象になります。

L F O クロックは、L F O 周期および L F O 遅延の時間単位を設定します。

「#ML:LFOCLOCK」による設定を変更したい場合に使用します。

「R」を指定した場合

L F O 解像度の設定が対象になります。

L F O 解像度は、L F O による情報更新の最少時間の単位です。

「#ML:LFORESOL」による設定を変更したい場合に使用します。

引数[2] (unit)

時間単位を、1文字で指定します。
 (英字の場合は大文字・小文字を問わず)

「S」を指定した場合

時間単位は「1秒」になります。
 有効範囲は、1[秒] ~ 1/2400[秒]です。

「%」を指定した場合

時間単位は「1 tickカウント」になります。

引数[3] (num)

unit によって指定した時間単位の倍率の分子を指定します。
 小数以下の指定も受け付けます。

引数[4] (denom)

unit によって指定した時間単位の倍率の分母を指定します。
 小数以下の指定も受け付けます。

【備考1】

num と denom の組は省略することが出来ます。
 省略した場合、時間単位の倍率は等倍 (1/1倍) とみなされます。

【備考2】

時間単位が「1 tickカウント」の場合、テンポにより時間単位が変動します。

【例1】

```
@LOC,S,7,300
```

LFOクロック = 7/300 [秒]

【例2】

```
@LOC,%,3,7
```

LFOクロック = 1 tickカウント × 3/7

【例3】

@LOR,S,1,600

LFO解像度 = 1/600 [秒]

【例4】

@LOR,%,1,2

LFO解像度 = 1 tickカウント × 1/2

8.2. @LP：ピッチ L F O

記述	@LP[1]
種別	MML コマンド

【解説】

「#ML:LF0@P」で定義されたピッチ L F O 情報を呼び出して適用します。トラック先頭におけるデフォルトは、ピッチ L F O 停止 (@LPZ) です。

引数[1]には、ピッチ L F O 番号を整数で指定します。未定義の番号を指定すると、エラーになります。

ピッチ L F O では、後述するメタデータ定義「#ML:LF0@P」によって、定義内容にピッチ L F O 番号を割り当てて登録したものを使用します。

【例】

```
#ML:LF0@P 1,50,20,10,0,0  
@LP1 C2;
```

8.3. @LA : 音量 L F O

記述	@LA[1]
種別	MML コマンド

【解説】

「#ML:LF0@A」で定義された音量 L F O 情報を呼び出して適用します。トラック先頭におけるデフォルトは、音量 L F O 停止 (@LAZ) です。

引数[1]には、音量 L F O 番号を整数で指定します。未定義の番号を指定すると、エラーになります。

音量 L F O では、後述するメタデータ定義「#ML:LF0@A」によって、定義内容に音量 L F O 番号を割り当てて登録したものを使用します。

【例】

```
#ML:LF0@A 1,4,18,9,0,0  
@LA1 C2;
```

8.4. @LB：パンポットバランスLFO

記述	@LB[1]
種別	MMLコマンド

【解説】

「#ML:LF0@B」で定義されたパンポットバランスLFO情報を呼び出して適用します。

トラック先頭におけるデフォルトは、パンポットバランスLFO停止 (@LBZ) です。

引数[1]には、パンポットバランスLFO番号を整数で指定します。未定義の番号を指定すると、エラーになります。

パンポットバランスLFOでは、後述するメタデータ定義「#ML:LF0@B」によって、定義内容にパンポットバランスLFO番号を割り当てて登録したものを使用します。

【例】

```
#ML:LF0@B 1,100,20,10,0,0  
@LB1 C2;
```


8.5. @LF：フィルタ L F O

記述	@LF[1]
種別	MML コマンド

【解説】

「#ML:LF0@F」で定義されたフィルタ L F O 情報を呼び出して適用します。トラック先頭におけるデフォルトは、フィルタ L F O 停止 (@LFZ) です。

引数[1]には、フィルタ L F O 番号を整数で指定します。未定義の番号を指定すると、エラーになります。

フィルタ L F O では、後述するメタデータ定義「#ML:LF0@F」によって、定義内容にフィルタ L F O 番号を割り当てて登録したものを使用します。さらに別途、@F コマンド でフィルタが掛かった状態で使用します。フィルタ L F O で操作するのは、適用されてる @F コマンド のカットオフが対象になります。

【例】

```
#ML:FILTER 1,2,0,50,0
#ML:LF0@F 1,20,30,1, 0,1
@F1 @LF1 C2;
```

8.6. @LY: YコントロールLFO

記述	@LY[1]
種別	MMLコマンド

【解説】

「#ML:LF0@Y」で定義されたYコントロールLFO情報を呼び出して適用します。トラック先頭におけるデフォルトは、YコントロールLFO停止 (@LYZ) です。

引数[1]には、YコントロールLFO番号を整数で指定します。未定義の番号を指定すると、エラーになります。

YコントロールLFOでは、後述するメタデータ定義「#ML:LF0@Y」によって、定義内容にYコントロールLFO番号を割り当てて登録したものを使用します。

【例】

```
#ML:LFORESOL S,1,60
#MB:ENV@A 1 { 15,0,&, N,0,15, R,0,7,N,32,7,N,1,0 }
/*          lp cp Do Dd Wm Wd */
#MB:LFOTBL@A 0 { 2, 0, 0, 1, 0, 1, 0.5, 0.5, 0.25 }
#MB:LFOTBL@R 0 { 0, 0, 0, 1, 0, 1, 0.125 }
/*          No Dp Wd Dl Md Yc tA tR */
#ML:LF0@Y 1, 1, 1, 0,pls, 5, 0, 0
T120 V13 @@PLS @EA1 @LY1
L8 05 CrErGrDrFrAr;
```

8.7. L F Oの停止

記述 @LPZ / @LAZ / @LBZ / @LFZ / @LYZ

種別 MML コマンド

【解説】

指定の L F O シーケンサを停止します。

@LPZ	ピッチ L F O を停止します。
@LAZ	音量 L F O を停止します。
@LBZ	パンポットバランス L F O を停止します。
@LFZ	フィルタ L F O を停止します。
@LYZ	Y コントロール L F O を停止します。

8.8. L F Oの再スタート

記述 @LPR / @LAR / @LBR / @LFR / @LYR

種別 MML コマンド

【解説】

指定のL F Oシーケンスを再スタートさせます。
スラー途中でL F Oを再スタートさせる場合や、L F Oのノートオン非同期モードで、任意のタイミングで再スタートさせる場合などの使用を想定しています。

@LPR	ピッチL F Oを再スタートさせます。
@LAR	音量L F Oを再スタートさせます。
@LBR	パンポットバランスL F Oを再スタートさせます。
@LFR	フィルタL F Oを再スタートさせます。
@LYR	YコントロールL F Oを再スタートさせます。

8.9. LFOテーブル番号変更

記述 @LPT[1],[2] / @LAT[1],[2] / @LBT[1],[2] / @LFT[1],[2] / @LYT[1],[2]

種別 MML コマンド

【解説】

指定のLFOにおいて使用中の、ユーザー定義テーブル番号を変更します。

@LPT[1],[2]	ピッチLFOで使用中の、ユーザー定義テーブル番号を変更します。
@LAT[1],[2]	音量LFOで使用中の、ユーザー定義テーブル番号を変更します。
@LBT[1],[2]	パンポットバランスLFOで使用中の、ユーザー定義テーブル番号を変更します。
@LFT[1],[2]	フィルタLFOで使用中の、ユーザー定義テーブル番号を変更します。
@LYT[1],[2]	YコントロールLFOで使用中の、ユーザー定義テーブル番号を変更します。

指定する引数は次の通りです。

引数[1]...ユーザー定義テーブルの種別 (target)

引数[2]...ユーザー定義テーブルの番号 (tableNum)

引数はカンマで区切って指定します。

引数[1] (target)

ユーザー定義テーブルの種別を 1 または 2 で指定します。

1: ノートオン時に使用するユーザー定義テーブル

2: ノートオフ時に使用するユーザー定義テーブル

引数[2] (tableNum)

使用するユーザー定義テーブル番号を指定します。

ここで指定する番号は、

target が 1 のとき、「#MB:LFOTBL@A」で定義済みの番号

target が 2 のとき、「#MB:LFOTBL@R」で定義済みの番号

を指定してください。

未定義番号を指定した場合、0 を指定したものとみなされます（0番は未定義でも内部保持のダミーデータを参照）。

target が 2 の場合は特別に、テーブル番号に -1 を指定することができます。

-1 を指定すると、ノートオフ同期でのテーブル番号変更機能が無効になります。（target が 1 の時の負数指定は 0 とみなされます）

【例 1】

@LPT1,20

ピッチ L F O で使用中の、ノートオン同期使用のテーブル番号を 20 に変更します。

【例 2】

@LAT1,10 @LAT2,101

音量 L F O で使用中の、ノートオン同期使用のテーブル番号を 10 に変更し、ノートオフ同期使用のテーブル番号を 101 に変更します。

8.10. #ML:LF0@P : ピッチ L F O データ定義

記述	#ML:LF0@P [1],[2],[3],[4],[5],[6],[7],[8]
種別	メタデータ定義

【解説】

ピッチ L F O のデータを定義します。

指定する引数は次の通りです。

引数[1]... L F O 番号 (num)

引数[2]... 揺らす振幅 (depth)

引数[3]... 揺らす時間幅 (width)

引数[4]... 開始までの遅延時間 (delay)

引数[5]... 揺らす波形番号 (form)

引数[6]... 揺らす波形番号枝番 (subform)

引数[7]... 揺らす波形への Y-control 番号 (exCmd)

引数[8]... 揺らす波形への Y-control データ (exCmdData)

引数はカンマで区切って指定します。

引数[1] (num)

L F O 番号を整数で指定します。

指定可能範囲は 0 ~ 1023 です。

@LP コマンド で参照される番号になります。

引数[2] (depth)

揺らす音程の振幅を数値で指定します。少数以下の指定も受け付けます。

単位は、PS コマンド で設定されている音程の解像度です。

ゼロを指定した場合、L F O が無効になります。

負数を指定した場合、変化パターンの上下が反転した動作になります。

振幅は「depth × form の波形変位」で得られます。

form の波形変位は、通常 -1 ~ 1 ですが、

ユーザー定義テーブルの場合は、自由に変位が定義できるため、振幅演算が「depth × テーブル値」となる点に注意してください。

引数[3] (width)

form の指定内容によって、width の時間単位が変わります。

formが0、1、2、3のとき：

width は form波形1周期の長さを数値で指定します。

1以上の値を指定してください。小数以下の指定も受け付けます。

周期の時間単位のデフォルトは「1 tickカウント」ですが、LFOクロック設定 (@L0コマンドなど)により、テンポに依存しない時間単位 (0.01秒など)にも出来ます。

formが4、5、6のとき：

width は LFOステップ数を数値で指定します。

1以上の値で指定してください。小数以下の指定も受け付けます。

LFOステップ数は、LFO解像度設定 (@L0コマンドなど)で決まる、LFO専用の最少時間単位です。

さらに、form が5の場合は、ユーザー定義テーブル内の設定で、widthの動作モードを、LFOステップ数とLFOシフト数の2つから選択します。

詳細は「#MB:LFOTBL@A」を参照してください。

引数[4] (delay)

ノートオンからLFO開始までの遅延時間を数値で指定します。

小数以下の指定も受け付けます。

時間単位は width 同様です。

delay が 0 以上のとき：

ノートオンの都度、LFOシーケンスが再スタートします。

(ノートオン同期モード)

delay が負数のとき：

ノートオンでLFOシーケンスが再スタートしない、ノートオン非同期モードとなります。いかなる負数の値も、このモード指定と判定されます。

ノートオン非同期モードにおいて、任意のタイミングでLFO位相のリセットを行いたい場合は、LFOの再スタートコマンド (@LPR)を使用します。

LFOコマンドの指定時には、同期・非同期モードに関わらずLFOシーケンスは位相ゼロからスタートします。

引数[5] (form)

引数[6] (subform)

変化パターンの波形を指定します。(波形パターン詳細)

波形の決定は、form と subform の番号の組み合わせによって行います。

form は波形のモジュール番号を指し、

subform はバリエーション番号を指します。

form=5であるユーザー定義テーブルの subform には、

「#MB:LFOTBL@A」で定義したテーブル番号を指定します。

form=6であるノンリアベンドの subform には、

変化率の引数として、1 ~ 512 を指定します。1未満もしくは512超えの指定は、それぞれ1、512に制限されます。

【form, subform, 波形の対応表】 (#ML:LFO@P)

form	subform	波形
0	0	サイン波
0	1	調整サイン波
1	0	リニアベンド
1	1	周期的リニアベンド
2	0	三角波
2	1	調整三角波
3	0	パルス波
3	1	調整パルス波
4	0	ホワイトノイズ
4	1	調整ホワイトノイズ
5	[番号]	ユーザー定義テーブル
6	[変化率]	ノンリニアベンド

引数[7] (exCmd)

引数[8] (exCmdData)

form で指定した波形モジュールに対して、
Yコントロールコマンドを発行します。

必要ない場合は、exCmd と exCmdData のセットは省略することができます。

LFO波形モジュールの、Yコントロール指定の詳細は[こちら](#)。

(exCmd は「func」、exCmdData は「param」に対応)

【例】

```
//          num depth width delay form subform exCmd exCmdData
#ML:LFO@P  1,   50,   20,   10,   0,     0
@LP1 C2;
```

この例では、exCmdとexCmdDataは省略しています。

8.11. #ML:LF0@A : 音量 L F O データ定義

記述	#ML:LF0@A [1],[2],[3],[4],[5],[6],[7],[8]
種別	メタデータ定義

【解説】

音量 (Amplitude) L F O のデータを定義します。

指定する引数は次の通りです。

引数[1]... L F O 番号 (num)

引数[2]... 揺らす振幅 (depth)

引数[3]... 揺らす時間幅 (width)

引数[4]... 開始までの遅延時間 (delay)

引数[5]... 揺らす波形番号 (form)

引数[6]... 揺らす波形番号枝番 (subform)

引数[7]... 揺らす波形への Y-control 番号 (exCmd)

引数[8]... 揺らす波形への Y-control データ (exCmdData)

引数はカンマで区切って指定します。

引数[1] (num)

L F O 番号を整数で指定します。

指定可能範囲は 0 ~ 1023 です。

@LA コマンド で参照される番号になります。

引数[2] (depth)

揺らす音量の振幅を設定します。単位は VS コマンド で指定する音量スケールです。少数点以下の指定も受け付けます。

ゼロを指定した場合、L F O が無効になります。

音量 L F O の場合、音量変化の変位は、0 で始まり、変位を減らす方向で推移させていて、利得は発生させない (リミッターが掛かる) 設計です。

音量変化の変位が 0 とは、指定の音量通りという状態です。

例えば、変化パターンのサイン波では 0 から (-1) までの間で変化するので、depth に 3 を指定していれば、音量変化は VS コマンド によるスケールで 0 ~ (-3) で揺れることになります。

depth に負数を指定した場合、変化パターンの上下が反転しますが、利得が発生しないよう制限されるので、意味の無い状態になります。

form がユーザー定義テーブルの場合、depth はテーブル値への倍率として動作します。この場合、テーブル値の内容によって自由に変位が定義できてしまうため、演算結果の範囲が 0 ~ 負数 になるようテーブル値への配慮が必要です（正の数は利得とみなされて 0 に制限されるため）。

引数[3] (width)

formの指定内容によって、widthの時間単位が変わります。

formが0、1、2、3のとき：

width は form波形1周期の長さを数値で指定します。

1以上の値を指定してください。小数以下の指定も受け付けます。

周期の時間単位のデフォルトは「1 tickカウント」ですが、LFOクロック設定 (@L0コマンドなど) により、テンポに依存しない時間単位 (0.01秒など) にも出来ます。

formが4、5のとき：

width は LFOステップ数を数値で指定します。

1以上の値で指定してください。小数以下の指定も受け付けます。

LFOステップ数は、LFO解像度設定 (@L0コマンドなど) で決まる、LFO専用の最少時間単位です。

さらに、form が5の場合は、ユーザー定義テーブル内の設定で、widthの動作モードを、LFOステップ数とLFOシフト数の2つから選択します。

詳細は「#MB:LFOTBL@A」を参照してください。

引数[4] (delay)

ノートオンからLFO開始までの遅延時間を数値で指定します。

小数以下の指定も受け付けます。

時間単位は width 同様です。

delay が 0 以上のとき：

ノートオンの都度、LFOシーケンスが再スタートします。

(ノートオン同期モード)

delay が負数のとき：

ノートオンでLFOシーケンスが再スタートしない、ノートオン非同期モードとなります。いかなる負数の値も、このモード指定と判定されます。

ノートオン非同期モードにおいて、任意のタイミングでLFO位相のリセットを行いたい場合は、LFOの再スタートコマンド (@LAR) を使用します。

LFOコマンドの指定時には、同期・非同期モードに関わらずLFOシーケンスは位相ゼロからスタートします。

引数[5] (form)

引数[6] (subform)

変化パターンの波形を指定します。(波形パターン詳細)

波形の決定は、form と subform の番号の組み合わせによって行います。

form は波形のモジュール番号を指し、

subform はバリエーション番号を指します。

form=5 であるユーザー定義テーブルの subform には、

「#MB:LFOTBL@A」で定義したテーブル番号を指定します。

【form, subform, 波形の対応表】 (#ML:LF0@A)

form	subform	波形
0	0	調整サイン波
1	0	周期的リニアバンド
2	0	調整三角波
3	0	パルス波
4	0	ホワイトノイズ
5	[番号]	ユーザー定義テーブル

引数[7] (exCmd)

引数[8] (exCmdData)

form で指定した波形モジュールに対して、

Yコントロールコマンドを発行します。

必要ない場合は、exCmd と exCmdData のセットは省略することができます。

LFO波形モジュールの、Yコントロール指定の詳細は[こちら](#)。

(exCmd は「func」、exCmdData は「param」に対応)

【例】

```
//          num depth width delay form subform exCmd exCmdData
#ML:LF0@A  1,    4,   18,    9,    0,    0
@LA1 C2;
```

この例では、exCmdとexCmdDataは省略しています。

8.12. #ML:LF0@B : パンポット L F O データ定義

記述	#ML:LF0@B [1],[2],[3],[4],[5],[6],[7],[8]
種別	メタデータ定義

【解説】

パンポットバランス L F O のデータを定義します。

指定する引数は次の通りです。

引数[1]... L F O 番号 (num)

引数[2]... 揺らす振幅 (depth)

引数[3]... 揺らす時間幅 (width)

引数[4]... 開始までの遅延時間 (delay)

引数[5]... 揺らす波形番号 (form)

引数[6]... 揺らす波形番号枝番 (subform)

引数[7]... 揺らす波形への Y-control 番号 (exCmd)

引数[8]... 揺らす波形への Y-control データ (exCmdData)

引数はカンマで区切って指定します。

引数[1] (num)

L F O 番号を整数で指定します。

指定可能範囲は 0 ~ 1023 です。

@LB コマンド で参照される番号になります。

引数[2] (depth)

揺らすパンポットの振幅を、-200 ~ 200 で指定します。

少数以下の指定も受け付けます。

ゼロを指定した場合、L F O が無効になります。

負数を指定した場合、変化パターンの上下が反転した動作になります。

パンポットが @P0 の状態で、depth が 100 で三角波の L F O をかけた場合、パンポットは最大幅で揺れます。上限の絶対値が 200 である理由は、パンポットを片方いっぱいにかけてから、変位がもう片方にしか振れない波形を使用する場合を想定しています。

パンポットの制限値を超える L F O をかけた場合は、超えた部分が制限値にリミットされながら L F O がかかります。

振幅は「depth × form の波形変位」で得られます。

form の波形変位は、通常 -1 ~ 1 ですが、

ユーザー定義テーブルの場合は、自由に変位が定義できるため、振幅演算が「depth × テーブル値」となる点に注意が必要です。

引数[3] (width)

formの指定内容によって、widthの時間単位が変わります。

formが0、1、2、3のとき：

width は form波形1周期の長さを数値で指定します。

1以上の値を指定してください。小数以下の指定も受け付けます。

周期の時間単位のデフォルトは「1 tickカウント」ですが、LFOクロック設定 (@L0コマンドなど)により、テンポに依存しない時間単位 (0.01秒など)にも出来ます。

formが4、5、6のとき：

width は LFOステップ数を数値で指定します。

1以上の値で指定してください。小数以下の指定も受け付けます。

LFOステップ数は、LFO解像度設定 (@L0コマンドなど)で決まる、LFO専用の最少時間単位です。

さらに、form が5の場合は、ユーザー定義テーブル内の設定で、widthの動作モードを、LFOステップ数とLFOシフト数の2つから選択します。

詳細は「#MB:LFOTBL@A」を参照してください。

引数[4] (delay)

ノートオンからLFO開始までの遅延時間を数値で指定します。

小数以下の指定も受け付けます。

時間単位は width 同様です。

delay が 0 以上のとき：

ノートオンの都度、LFOシーケンスが再スタートします。

(ノートオン同期モード)

delay が負数のとき：

ノートオンでLFOシーケンスが再スタートしない、ノートオン非同期モードとなります。いかなる負数の値も、このモード指定と判定されます。

ノートオン非同期モードにおいて、任意のタイミングでLFO位相のリセットを行いたい場合は、LFOの再スタートコマンド (@LBR)を使用します。

LFOコマンドの指定時には、同期・非同期モードに関わらずLFOシーケンスは位相ゼロからスタートします。

引数[5] (form)

引数[6] (subform)

変化パターンの波形を指定します。(波形パターン詳細)

波形の決定は、form と subform の番号の組み合わせによって行います。

form は波形のモジュール番号を指し、
 subform はバリエーション番号を指します。
 form=5 であるユーザー定義テーブルの subform には、
 「#MB:LFOTBL@A」で定義したテーブル番号を指定します。
 form=6 であるノンリニアベンドの subform には、
 変化率の引数として、1 ~ 512 を指定します。1未満もしくは512超えの指定
 は、それぞれ1、512に制限されます。

【form, subform, 波形の対応表】 (#ML:LF0@B)

form	subform	波形
0	0	サイン波
0	1	調整サイン波
1	0	リニアベンド
1	1	周期的リニアベンド
2	0	三角波
2	1	調整三角波
3	0	パルス波
3	1	調整パルス波
4	0	ホワイトノイズ
4	1	調整ホワイトノイズ
5	[番号]	ユーザー定義テーブル
6	[変化率]	ノンリニアベンド

引数[7] (exCmd)

引数[8] (exCmdData)

form で指定した波形モジュールに対して、
 Yコントロールコマンドを発行します。
 必要ない場合は、exCmd と exCmdData のセットは省略することができます。

LF0波形モジュールの、Yコントロール指定の詳細は[こちら](#)。
 (exCmd は「func」、exCmdData は「param」に対応)

【例】

```
//          num depth width delay form subform exCmd exCmdData  
#ML:LFO@B  1, 100,  20,  10,  0,    0  
@LB1 C2;
```

この例では、exCmdとexCmdDataは省略しています。

8.13. #ML:LF0@F : フィルタ L F O データ定義

記述	#ML:LF0@F [1],[2],[3],[4],[5],[6],[7],[8]
種別	メタデータ定義

【解説】

フィルタ L F O のデータを定義します。

指定する引数は次の通りです。

引数[1]... L F O 番号 (num)

引数[2]... 揺らす振幅 (depth)

引数[3]... 揺らす時間幅 (width)

引数[4]... 開始までの遅延時間 (delay)

引数[5]... 揺らす波形番号 (form)

引数[6]... 揺らす波形番号枝番 (subform)

引数[7]... 揺らす波形への Y-control 番号 (exCmd)

引数[8]... 揺らす波形への Y-control データ (exCmdData)

引数はカンマで区切って指定します。

引数[1] (num)

L F O 番号を整数で指定します。

指定可能範囲は 0 ~ 1023 です。

@LF コマンド で参照される番号になります。

引数[2] (depth)

揺らすカットオフの振幅を 0 ~ 100 で指定します。少数以下の指定も受け付けます。

ゼロを指定した場合、L F O が無効になります。

負数を指定した場合、変化パターンの上下が反転した動作になります。

振幅は「depth × form の波形変位」で得られます。

form の波形変位は、通常 -1 ~ 1 ですが、

ユーザー定義テーブルの場合は、自由に変位が定義できるため、振幅演算が「depth × テーブル値」となる点に注意してください。

引数[3] (width)

formの指定内容によって、widthの時間単位が変わります。

formが0、1、2、3のとき：

width は form波形1周期の長さを数値で指定します。

1以上の値を指定してください。小数以下の指定も受け付けます。

周期の時間単位のデフォルトは「1 tickカウント」ですが、LFOクロック設定 (@L0コマンドなど)により、テンポに依存しない時間単位 (0.01秒など)にも出来ます。

formが4、5、6のとき：

width は LFOステップ数を数値で指定します。

1以上の値で指定してください。小数以下の指定も受け付けます。

LFOステップ数は、LFO解像度設定 (@L0コマンドなど)で決まる、LFO専用の最少時間単位です。

さらに、form が5の場合は、ユーザー定義テーブル内の設定で、widthの動作モードを、LFOステップ数とLFOシフト数の2つから選択します。

詳細は「#MB:LFOTBL@A」を参照してください。

引数[4] (delay)

ノートオンからLFO開始までの遅延時間を数値で指定します。

小数以下の指定も受け付けます。

時間単位は width 同様です。

delay が 0 以上のとき：

ノートオンの都度、LFOシーケンスが再スタートします。

(ノートオン同期モード)

delay が負数のとき：

ノートオンでLFOシーケンスが再スタートしない、ノートオン非同期モードとなります。いかなる負数の値も、このモード指定と判定されます。

ノートオン非同期モードにおいて、任意のタイミングでLFO位相のリセットを行いたい場合は、LFOの再スタートコマンド (@LFR)を使用します。

LFOコマンドの指定時には、同期・非同期モードに関わらずLFOシーケンスは位相ゼロからスタートします。

引数[5] (form)**引数[6] (subform)**

変化パターンの波形を指定します。(波形パターン詳細)

波形の決定は、form と subform の番号の組み合わせによって行います。

form は波形のモジュール番号を指し、

subform はバリエーション番号を指します。

form=5であるユーザー定義テーブルの subform には、

「#MB:LFOTBL@A」で定義したテーブル番号を指定します。

form=6であるノンリニアベンドの subform には、変化率の引数として、1 ~ 512 を指定します。1未満もしくは512超えの指定は、それぞれ1、512に制限されます。

【form, subform, 波形の対応表】 (#ML:LFO@F)

form	subform	波形
0	0	サイン波
0	1	調整サイン波
1	0	リニアベンド
1	1	周期的リニアベンド
2	0	三角波
2	1	調整三角波
3	0	パルス波
3	1	調整パルス波
4	0	ホワイトノイズ
4	1	調整ホワイトノイズ
5	[番号]	ユーザー定義テーブル
6	[変化率]	ノンリニアベンド

引数[7] (exCmd)

引数[8] (exCmdData)

form で指定した波形モジュールに対して、Yコントロールコマンドを発行します。

必要ない場合は、exCmd と exCmdData のセットは省略することができます。

LFO波形モジュールの、Yコントロール指定の詳細は[こちら](#)。

(exCmd は「func」、exCmdData は「param」に対応)

【例】

```
//          num depth width delay form subform exCmd exCmdData
#ML:LFO@F  1,  20,  30,   1,   0,   1
#ML:FILTER 1,2,0,50,0
@F1 @LF1 C2;
```

この例では、exCmdとexCmdDataは省略しています。

8.14. #ML:LF0@Y : YコントロールL F Oデータ定義

記述	#ML:LF0@Y [1],[2],[3],[4],[5],[6],[7],[8]
種別	メタデータ定義

【解説】

YコントロールL F Oのデータを定義します。

指定する引数は次の通りです。

引数[1]...L F O番号 (num)

引数[2]...揺らす振幅 (depth)

引数[3]...揺らす時間幅 (width)

引数[4]...開始までの遅延時間 (delay)

引数[5]...宛先音源の識別子 (modType)

引数[6]...Yコントロールコマンド番号 (YcmdNum)

引数[7]...ノートオン時のテーブル番号 (table1)

引数[8]...ノートオフ時のテーブル番号 (table2)

引数はカンマで区切って指定します。

YコントロールL F Oでは、揺らす波形はユーザー定義テーブル方式に固定されていて、ユーザー定義テーブルに従ったYコントロールのコマンド発行シーケンスが行われます。

ユーザー定義テーブルには、

#MB:LF0TBL@A (ノートオン同期向け)

#MB:LF0TBL@R (ノートオフ同期向け)

の2種類があります。

引数[1] (num)

L F O番号を整数で指定します。

指定可能範囲は 0 ~ 1023 です。

@LYコマンド で参照される番号になります。

引数[2] (depth)

ユーザー定義テーブルから得られる値への倍率を数値で指定します。少数以下の指定も受け付けます。

ゼロを指定した場合、L F Oが無効になります。

引数[3] (width)

LFOステップ数を数値で指定します。

1以上の値で指定してください。小数以下の指定も受け付けます。

LFOステップ数は、LFO解像度設定 (@L0コマンドなど) で決まる、LFO専用の最少時間単位です。

また、widthの動作モードは、LFOステップ数とLFOシフト数の2つから選択します。

詳細は「#MB:LFOTBL@A」を参照してください。

引数[4] (delay)

ノートオンからLFO開始までの遅延時間を数値で指定します。

小数以下の指定も受け付けます。

時間単位は width 同様です。

delay が 0 以上のとき：

ノートオンの都度、LFOシーケンスが再スタートします。

(ノートオン同期モード)

delay が負数のとき：

ノートオンでLFOシーケンスが再スタートしない、ノートオン非同期モードとなります。いかなる負数の値も、このモード指定と判定されます。

ノートオン非同期モードにおいて、任意のタイミングでLFO位相のリセットを行いたい場合は、LFOの再スタートコマンド (@LYR) を使用します。

LFOコマンドの指定時には、同期・非同期モードに関わらずLFOシーケンスは位相ゼロからスタートします。

delay によりYコントロールLFOが遅延されている間は、Yコントロールコマンドが発行されないため、ノートオン直前まで発行されていたYコントロールが残ったままの形になり、想定外のシーケンスになる可能性があります。

このような問題を避けるには、delay を 0 とし、LFOテーブル内のデータ定義で、遅延に相当するデータを含むデータ列にします。

引数[5] (modType)

Yコントロールを行う宛先となる音源モジュールを3文字の識別子で指定します。識別子は音源モジュール指定と同様です。

引数[6] (YcmdNum)

Yコントロールのコマンド番号を整数で指定します。

引数[7] (table1)

ノートオン時に使用するユーザー定義テーブル番号を整数で指定します。
 「#MB:LFOTBL@A」にて定義済みの番号を指定してください。
 未定義番号を指定した場合、0 を指定したものとみなされます。
 (0番はテーブルが未定義でも内部でダミーデータを保持)

引数[8] (table2)

ノートオフ時に使用するユーザー定義テーブル番号を整数で指定します。
 「#MB:LFOTBL@R」にて定義済みの番号を指定してください。
 未定義の番号を指定した場合、0 を指定したものとみなされます。
 (0番はテーブルが未定義でも内部でダミーデータを保持)
 ノートオフに同期してユーザー定義テーブルを切り替えたくない場合、
 -1 (または負数)
 を指定してください。

【例】

```
#ML:LFORESOL S,1,60
#MB:ENV@A 1 { 15,0,&, N,0,15, R,0,7,N,32,7,N,1,0 }
/*          lp cp Do Dd Wm Wd */
#MB:LFOTBL@A 0 { 2, 0, 0, 1, 0, 1, 0.5, 0.5, 0.25 }
#MB:LFOTBL@R 0 { 0, 0, 0, 1, 0, 1, 0.125 }

/*          num depth width delay modType YcmdNum table1 table2 */
#ML:LFO@Y 1, 1, 1, 0, pls, 5, 0, 0
T120 V13 @@PLS @EA1 @LY1
L8 05 CrErGrDrFrAr;
```

8.15. #ML:LFOCLOCK : L F O 時間単位

記述 #ML:LFOCLOCK [1],[2],[3]

種別 メタデータ指定

【解説】

L F O 全種類への、時間単位を設定します。
時間単位は、width の周期時間と、delay に影響します。
この設定は、全トラックへの初期設定となります。

指定する引数は次の通りです。

引数[1]...時間単位 (unit)

引数[2]...時間単位の倍率の分子 (num)

引数[3]...時間単位の倍率の分母 (denom)

引数はカンマで区切って指定します。

当メタデータ指定を使用しなかった場合のデフォルト設定は次の通りです。

```
#ML:LFOCLOCK %
```

引数[1] (unit)

時間単位を、1文字で指定します。
(英字は大文字・小文字を問わず)

「S」を指定した場合

時間単位は「1秒」になります。
有効範囲は、1[秒] ~ 1/2400[秒]です。

「%」を指定した場合

時間単位は「1 tickカウント」になります。

引数[2] (num)

unit によって指定した時間単位の倍率の分子を指定します。
小数以下の指定も受け付けます。

引数[3] (denom)

unit によって指定した時間単位の倍率の分母を指定します。
小数以下の指定も受け付けます。

【備考1】

num と denom の組は省略することが出来ます。
省略した場合、時間単位の倍率は等倍 (1/1倍) とみなされます。

【備考2】

時間単位が「1 tickカウント」の場合、テンポにより時間単位が変動します。

【例1】

```
#ML:LFOCLOCK S,7,300
```

LFOクロック = $7/300$ [秒]

【例2】

```
#ML:LFOCLOCK %,3,7
```

LFOクロック = 1 tickカウント × $3/7$

8.16. #ML:LFORESOL : L F O 解像度

記述 #ML:LFORESOL [1],[2],[3]

種別 メタデータ指定

【解説】

LFO時間軸の解像度（LFO変位計算の最少時間単位）を設定します。
この設定は全トラックへの初期値となります。

この設定が有効になる対象は、

ピッチLFO

フィルタLFO

YコントロールLFO

の3種類です。

LFO解像度は、ポルタメント機能による周波数変更の最小時間単位にもなっています。

指定する引数は次の通りです。

引数[1]...時間単位 (unit)

引数[2]...時間単位の倍率の分子 (num)

引数[3]...時間単位の倍率の分母 (denom)

引数はカンマで区切って指定します。

当メタデータ指定を使用しなかった場合のデフォルト設定は次の通りです。

```
#ML:LFORESOL S,1,600
```

引数[1] (unit)

時間単位を、1文字で指定します。

(英字は大文字・小文字を問わず)

「S」を指定した場合

時間単位は「1秒」になります。

有効範囲は、1[秒] ~ 1/2400[秒]です。

「%」を指定した場合

時間単位は「1 tickカウント」になります。

引数[2] (num)

unit によって指定した時間単位の倍率の分子を指定します。
小数以下の指定も受け付けます。

引数[3] (denom)

unit によって指定した時間単位の倍率の分母を指定します。
小数以下の指定も受け付けます。

【備考1】

num と denom の組は省略することが出来ます。
省略した場合、時間単位の倍率は等倍 (1/1倍) とみなされます。

【備考2】

時間単位が「1 tickカウント」の場合、テンポにより時間単位が変動します。

【備考3】

音量とパンポットの L F O (@LA / @LB) は、常に1サンプル単位で動作するので、本機能の対象外になります。

【例1】

```
#ML:LFORESOL S,1,300
```

L F O解像度 = 1/300 [秒]

【例2】

```
#ML:LFORESOL %,1,2
```

L F O解像度 = 1 tickカウント × 1/2

8.17. #MB:LFOTBL@A : L F Oテーブル定義 (A)

記述 #MB:LFOTBL@A [1] {[2]}

種別 メタデータ定義

【解説】

L F Oの変化パターン、ユーザー定義テーブルモードで使用する、テーブルデータを定義します。

「#MB:LFOTBL@A」は、ノートオン同期用に使用されます。

また、ユーザー定義テーブルのデータ定義領域は、全てのL F Oで共有して利用するため、例えばピッチL F O用に定義した内容であっても、音量L F Oなど他のL F Oからでも利用できてしまう点に注意して下さい。

指定する引数は次の通りです。

引数[1]...ユーザー定義テーブル番号 (tableNum) 【1個の整数】

引数[2]...ユーザー定義テーブルデータ (tableData) 【複数のパラメータ】

引数[1] (tableNum)

ユーザー定義テーブル番号を、整数で指定します。

これは、各L F Oにおける、form=5 の場合の subform に記述する番号に相当します。

これは、「#ML:LF0@P」「#ML:LF0@A」「#ML:LF0@B」「#ML:LF0@F」の定義において、form=5 の場合の subform に記述する番号に相当します。

「#ML:LF0@Y」の定義では、table1 に記述する番号に相当します。

設定範囲は 0 ~ 4095 です。

未定義のテーブル番号を参照した場合は、0 番が読み出されます。

テーブル番号 0 番には、デフォルトではダミーデータが入っています（上書可）。

引数[2] (tableData)

ユーザー定義テーブルデータのパラメータ群 (tableData) の記述開始は

「 { 」で認識され、終了は「 } 」によって検知されます。

各パラメータは、カンマで区切って記述します。

スペースや改行は読み飛ばされます。

tableDataは、6つの設定値と、1個以上（最大2048個）のテーブル値で構成されます。そのため、最低7つの数値パラメータを定義する必要があります。

フォーマットは次の通りです。

```
#MB:LFOTBL@A テーブル番号 {
    loop, cpl, Dofs, Ddnm, Wmod, Wdnm,
    テーブル値
}
```

loop :

ループエントリインデックスを指定します。

テーブルシーケンスをループさせる場合は、ループエントリインデックスを整数（0以上2047以下）で指定します。

ループさせない場合は、-1と指定します。

ループさせない場合、テーブル末尾に到達後は、テーブル末尾の値が維持されます。

cpl :

テーブル値を読み出すときの補間モードを、0または1で指定します。

0 の場合 :

0次補間モードです。

0次補間モードでは、読み出し位置が少数以下切捨てる場所にあるテーブル値がそのまま使われるため、ガクガクした形になります。

このモードは、急峻な変化の表現に向いています。このモードで滑らかに変化させるには多くのテーブル値を要します。

1 の場合 :

1次補間モードです。

1次補完モードでは、読み出し位置（少数以下含む）に応じて、前後のテーブル値から線形補完した値が使われるため、少ないテーブル値でも滑らかに変化させられます。

Dofs :

テーブル値へのオフセット（加算値）を設定します。

小数以下の指定も受け付けます。

テーブル値の読み出しが、テーブル値に Dofs を加算した値で行われます。

Ddnm :

テーブル値への分母を設定します。

小数以下の指定も受け付けます。

0 より大きい数値を指定してください。

テーブル値の読み出しが、テーブル値を Ddnm で除算した値で行われます。

計算の優先順位は、オフセット加算が優先です。

$$\text{実際の値} = (\text{テーブル値} + \text{Dofs}) \div \text{Ddnm}$$

となります。

Wmod :

LFOの width で受け取る時間幅の扱いを、0 または 1 で指定します。

0 の場合 :

width をステップ数として扱います。

テーブルシーケンスは「#ML:LFORESOL」による解像度を 1 ステップとして動作しますが、何ステップ経過後にテーブルインデックスを 1 進めるかを決定するモードとなります。

この場合、LFOでの width 受付範囲は 1 以上になります。少数以下の指定も受け付けます。

1 の場合 :

width をシフト数として扱います。

テーブルシーケンスは「#ML:LFORESOL」による解像度を 1 ステップとして動作しますが、1 ステップ経過後に、テーブルインデックスをどれだけ進めるかを決定するモードとなります。

この場合、LFOでの width 受付範囲は、0 より大きく、Wdnm 以下になります。

(width は Wdnm で除算され、0 より大きく 1 以下の範囲で利用)

Wdnm :

Wmod で 1 を指定した場合のみ有効な、シフト数に対する分母を設定します。

小数以下の指定も受け付けます。

0 より大きい数値を指定してください。

テーブル値 :

1 個以上（最大2048個まで）のテーブル値を指定します。

各テーブル値では、小数以下の指定も受け付けます。

最大個数を超える定義を行うとエラーとなり、テーブル自体が定義されません。

【テーブル定義例】

```
#MB:LFOTBL@A 1 {  
/* loop cml Dofs Ddnm Wmod Wdnm */  
    0,  0,  0,  1,  0,  1,  
    0,  1,  0, -1  
}  
  
#MB:LFOTBL@A 2 {  
/* loop cml Dofs Ddnm Wmod Wdnm */  
    0,  1,  0,  1,  0,  1,  
    0,  1,  0, -1  
}  
#ML:LF0@P 1, 400,36,18, 5,1  
#ML:LF0@P 2, 400,36,18, 5,2  
T120 Q16,16 05 @@PLS L2  
@LP1 CEG  
@LP2 CEG;
```

@LP1ではギザギザのピッチ変化、@LP2では滑らかなピッチ変化を掛けています。

8.18. #MB:LFOTBL@R : L F Oテーブル定義 (R)

記述	#MB:LFOTBL@R [1] {[2]}
種別	メタデータ定義

【解説】

YコントロールL F Oデータ定義の、table2 で使用する、テーブルデータを定義します。

指定する引数は次の通りです。

引数[1]...ユーザー定義テーブル番号 (tableNum) 【1個の整数】

引数[2]...ユーザー定義テーブルデータ (tableData) 【複数のパラメータ】

「#MB:LFOTBL@R」は、ノートオフ同期専用に使われます。
定義できる数と、その定義内容は「#MB:LFOTBL@A」と同様です。

「#MB:LFOTBL@R」によるテーブル定義は、YコントロールL F O以外のL F Oでも使えます。

YコントロールL F O以外のL F Oでは、通常ノートオフに同期してテーブルデータを切り替えません。

しかし特別な用途、例えば音量L F Oをテーブルシーケンスさせて、擬似エンベロープのように使うような、ノートオフに同期させてテーブルを変更したい場合などが考えられます。

その際には、音量L F Oデータ定義時の exCmd と exCmdData の指定を使ったり、@LATコマンドを使うことにより、「#MB:LFOTBL@R」によるテーブル定義をノートオフ同期用に指定することができます。

【テーブル定義例】

```

#ML:LFORESOL S,1,60
#MB:ENV@A 0 { 15,0,&, N,0,15, R,0,7, N,32,7, N,1,0 }

#MB:LFOTBL@A 0 {
/* loop cmpL Dofs Ddnm Wmod Wdnm */
    2, 0, 0, 1, 0, 1,
    0.5, 0.5, 0.25
}
#MB:LFOTBL@R 0 {
/* loop cmpL Dofs Ddnm Wmod Wdnm */
    0, 0, 0, 1, 0, 1,
    0.125
}

/*          No Dp Wd DL Md Yc tA tR */
#ML:LFO@Y 0, 1, 1, 0,pls, 5, 0, 0
T150 Q8,16
@@PLS @EA0 @LY0
L2 05 CDEFGFEDCEGEC;

```

YコントロールLFOにて、パルス波のデューティ比を時間変化させています。ノートオン中は、最初の1/30[秒]がデューティ比 50% で、それ以後 25%。ゲート切れやノートオフになったらデューティ比を 12.5% にしています。

8.19. #ML:LFOTBLPM : L F Oテーブル処理モード

記述	#ML:LFOTBLPM [1]
----	------------------

種別	メタデータ指定
----	---------

【解説】

L F Oにおける、ユーザー定義テーブルのプロセスモード（処理モード）を設定します。

引数[1]には、0 または 1 でモード番号を指定します。

0 のとき、テーブル値を処理してから、ポインタを進めます。

1 のとき、ポインタを進めてから、テーブル値を処理します。

特に必要性が無い場合、0 を使用してください。

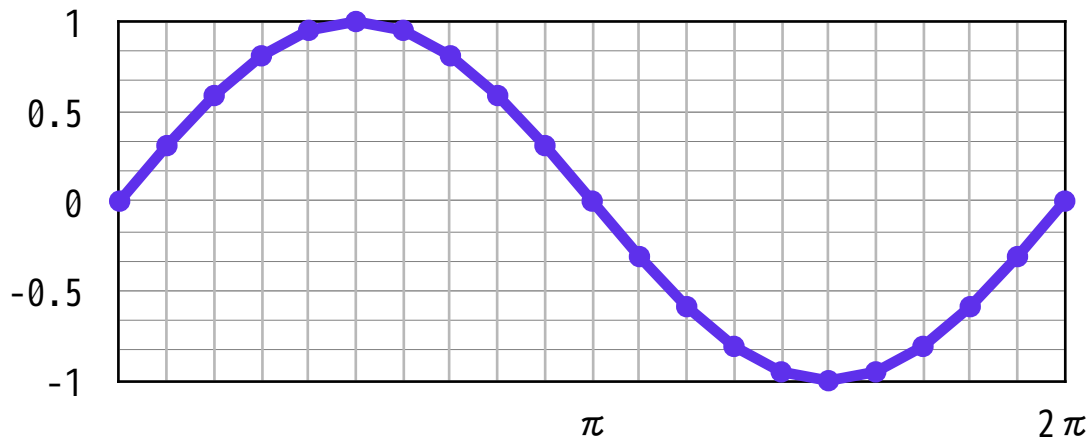
当メタデータ指定を使用しなかった場合のデフォルト設定は次の通りです。

```
#ML:LFOTBLPM 0
```

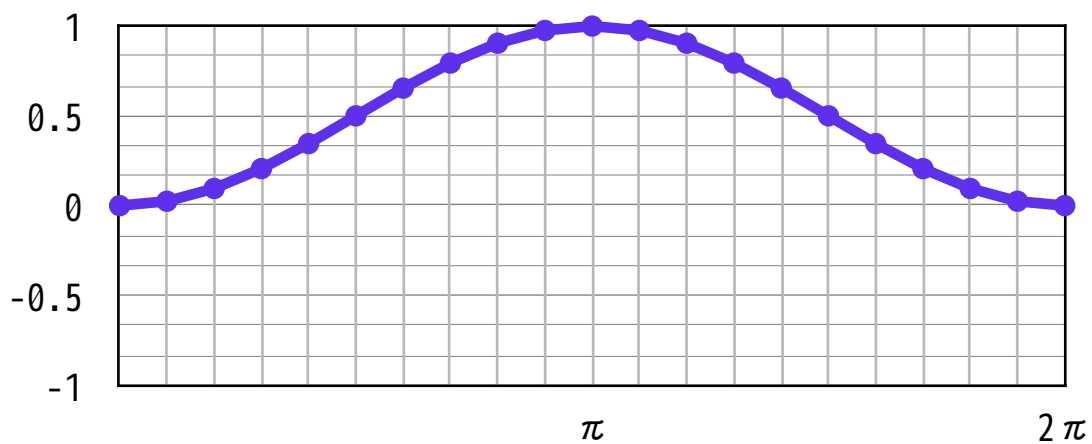
8.20. L F O 波形グラフ (1) (@LP/@LB/@LF)

L F O 波形グラフ (1) : @LP/@LB/@LF の変化パターン

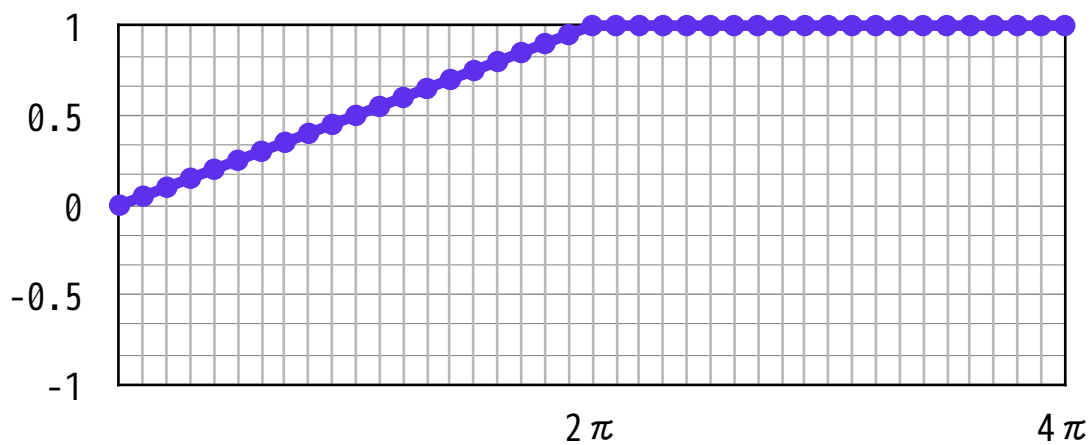
サイン波 : form=0, subform=0



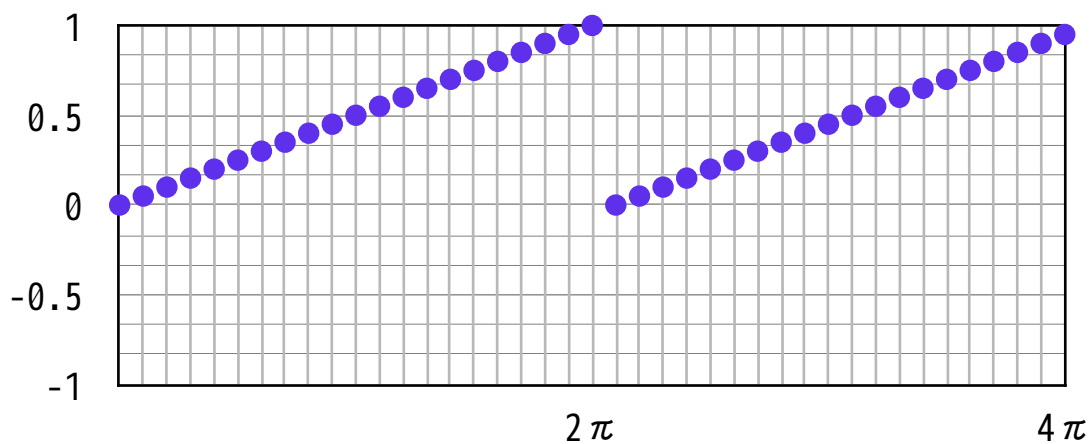
調整サイン波 : form=0, subform=1



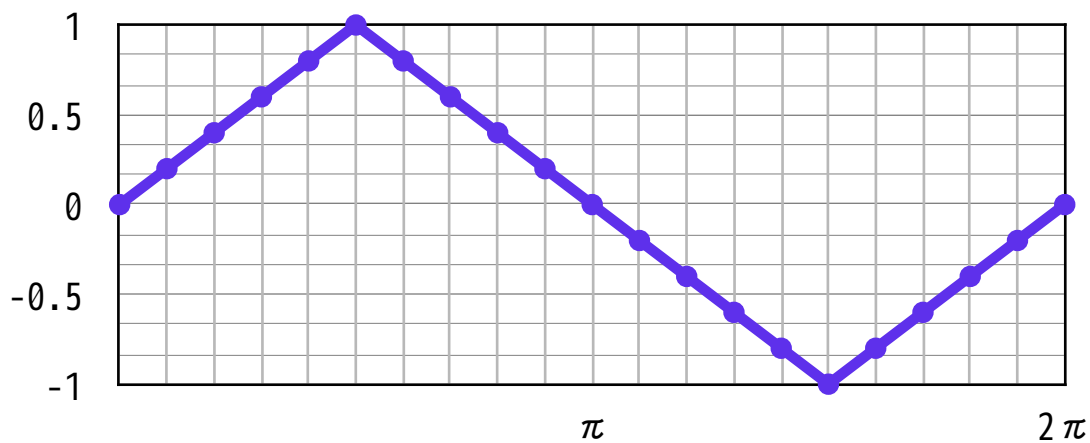
リニアベンド : form=1, subform=0



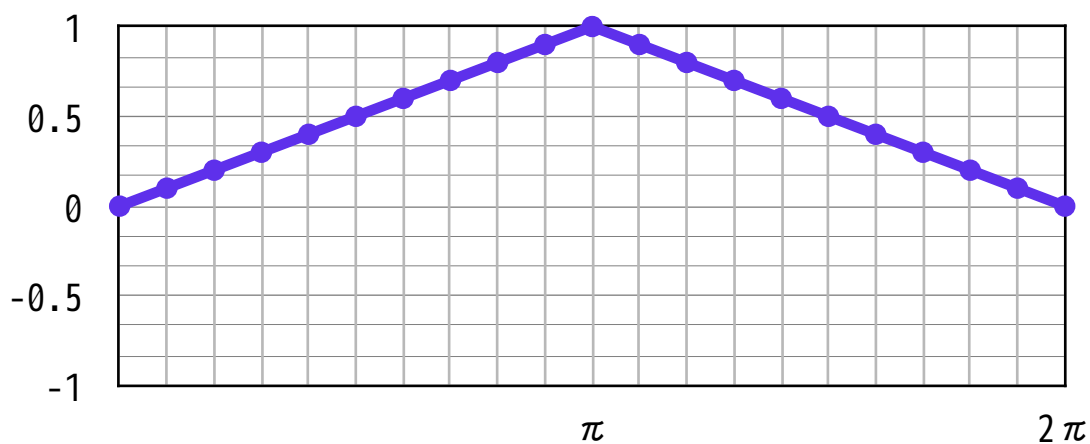
周期的リニアバンド : form=1, subform=1



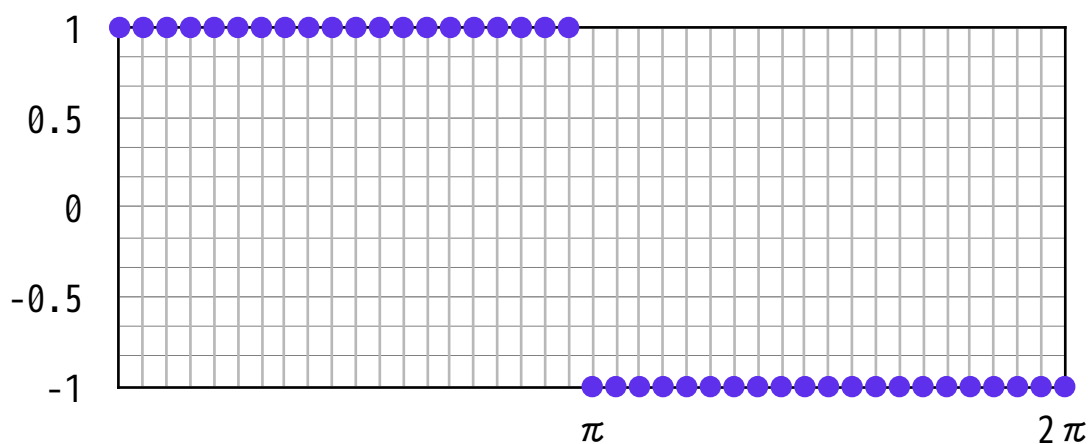
三角波 : form=2, subform=0



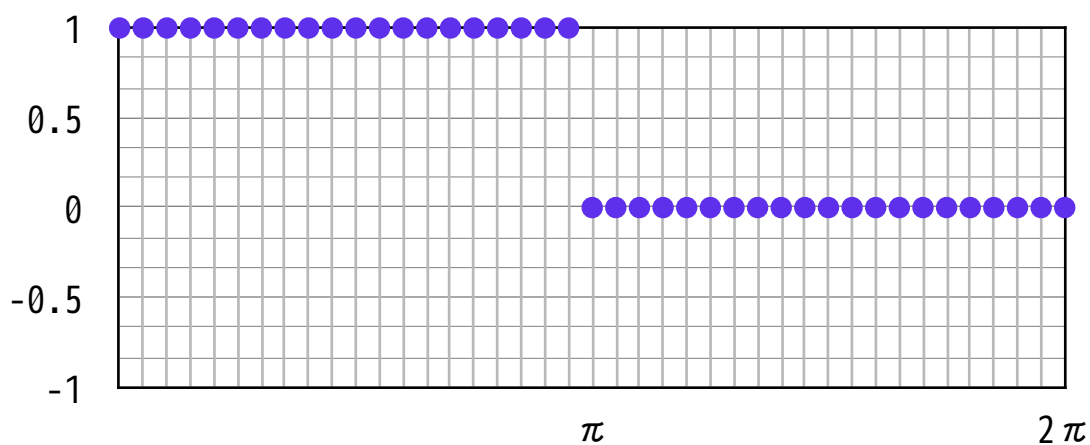
調整三角波 : form=2, subform=1



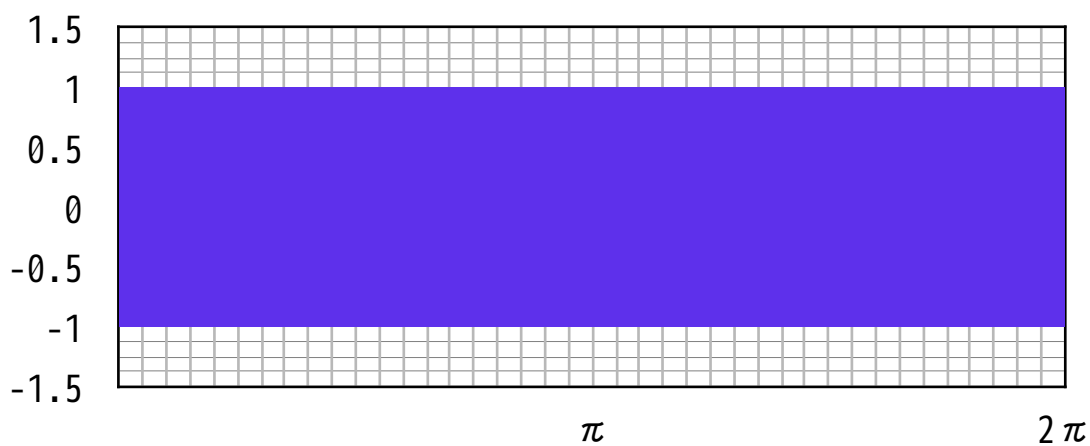
パルス波 : form=3, subform=0



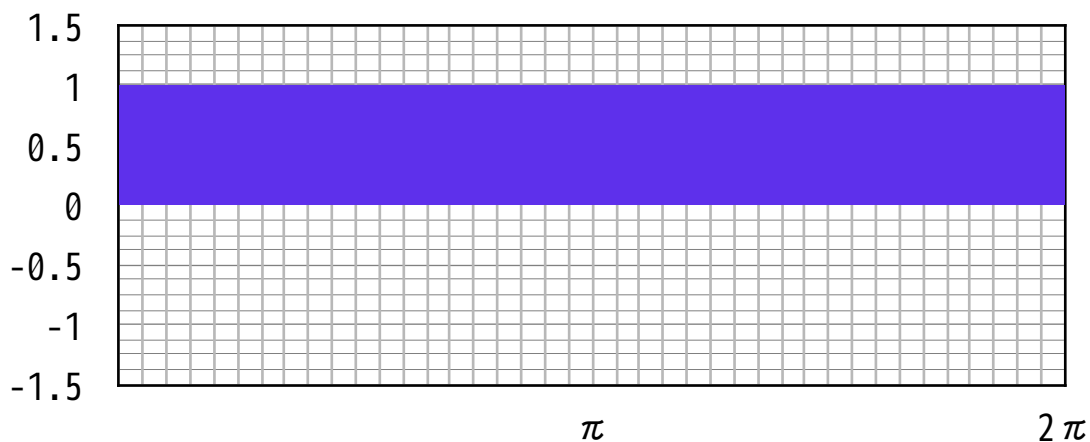
調整パルス波 : form=3, subform=1



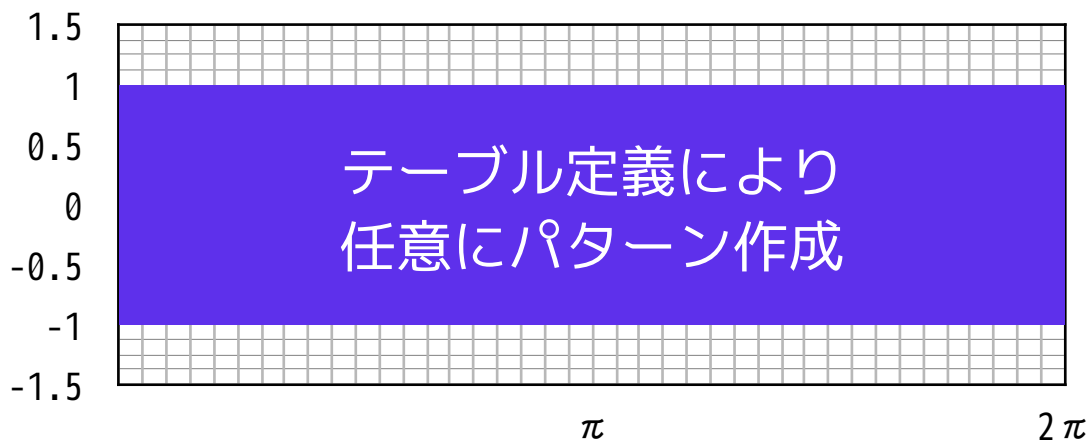
ホワイトノイズ : form=4, subform=0



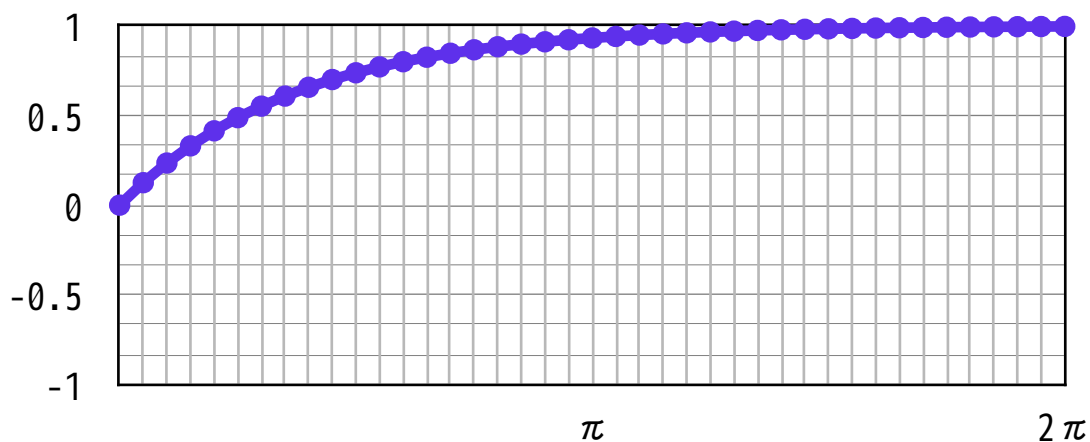
ホワイトノイズ B : form=4, subform=1



ユーザー定義テーブル : form=5, subform=x



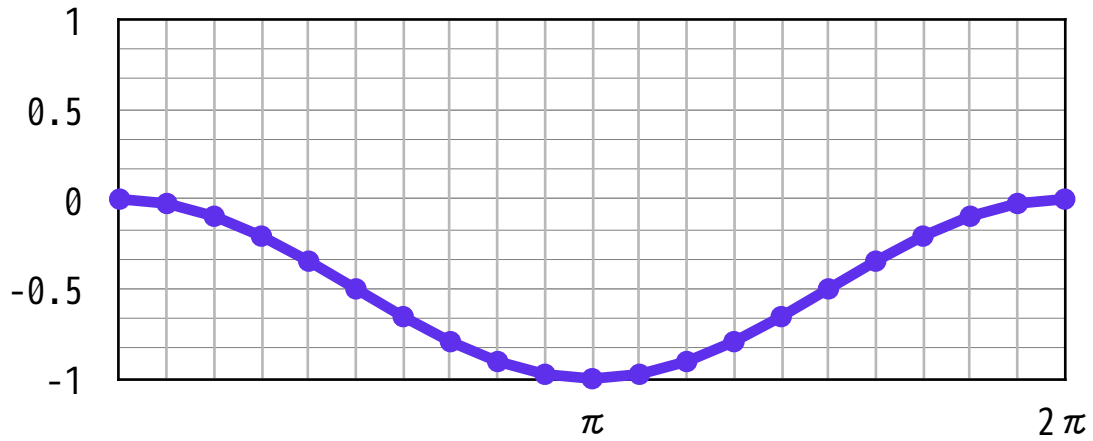
ノンリニアバンド : form=6, subform=128



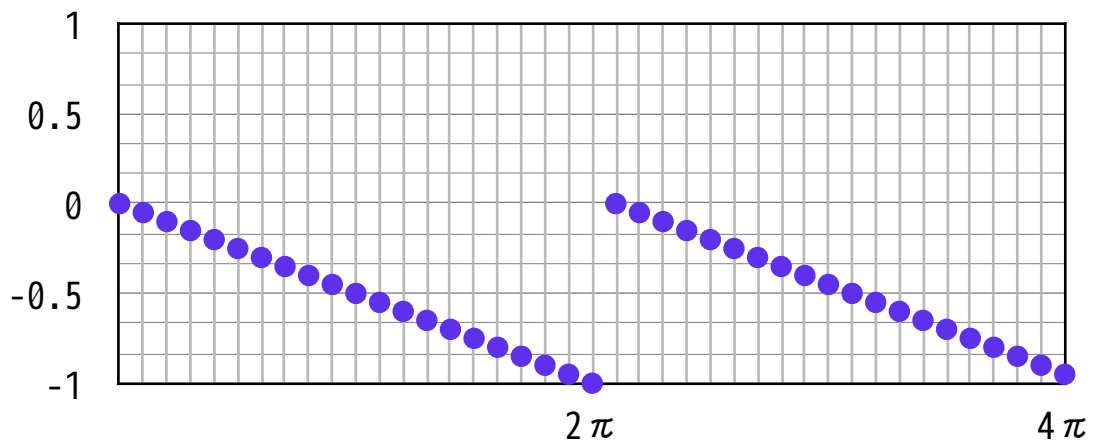
8.21. L F O 波形グラフ (2) (@LA)

L F O 波形グラフ (2) : @LA の変化パターン

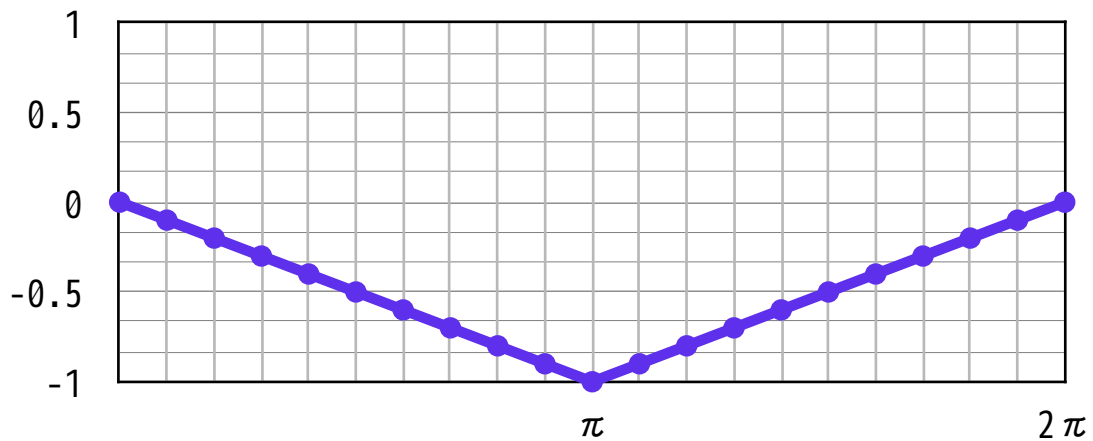
調整サイン波 : form=0, subform=0



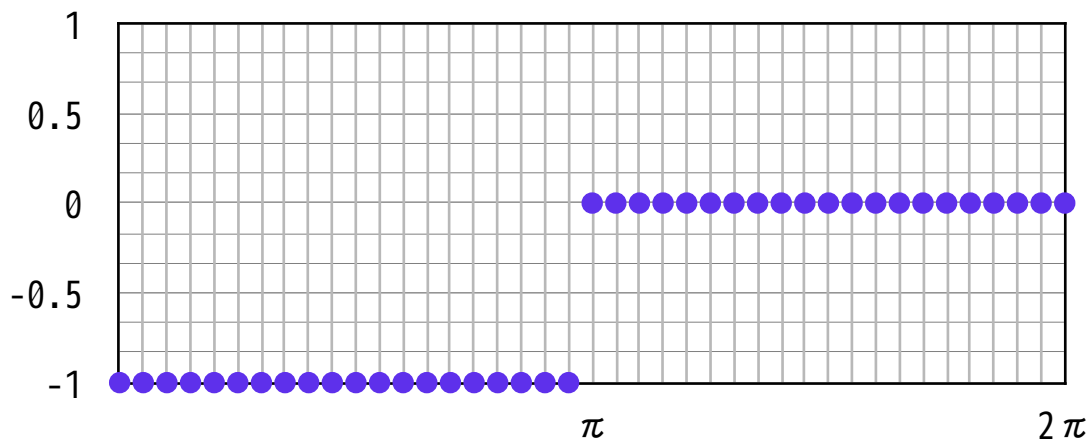
周期的リニアバンド : form=1, subform=0



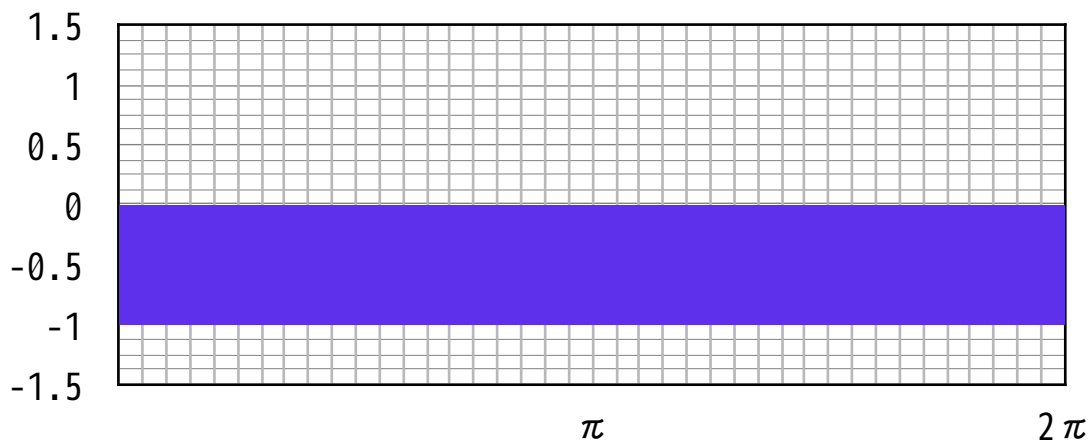
調整三角波 : form=2, subform=0



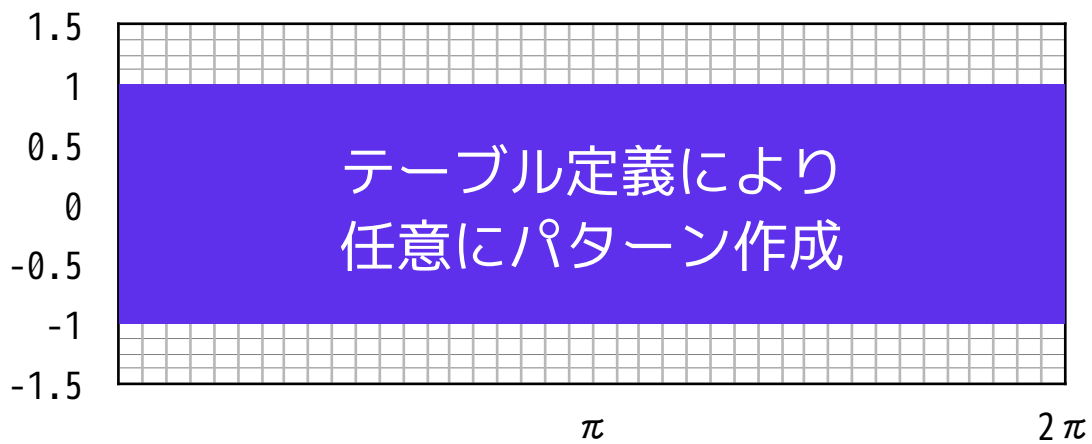
パルス波 : form=3, subform=0



ホワイトノイズ : form=4, subform=0



ユーザー定義テーブル : form=5, subform=x



9. 発音数関連

9.1. #ML:USING-POLY : ポリフォニックモード有効化

記述	#ML:USING-POLY [1],[2]
種別	メタデータ指定

【解説】

ポリフォニックモードを有効にします。

ポリフォニックモードとは、1つのトラックで複数の音を同時に鳴らすためのモードです。@PLコマンド と組み合わせて使用します。

ポリフォニックモードが有効になると、通常1トラックあたり1音のところを、指定音数のチャンネルを内部で確保し、複数音の同時発音を可能にします。その結果、1トラックでも、長いリリースの音を途切れさせずに重ねて演奏させたり、和音記法を用いて和音を演奏させることも出来ます。

指定する引数は次の通りです。

引数[1]...最大発音数 (maxPoly)

引数[2]...最大発音数の強制指示 (maxCmd)

引数はカンマで区切って指定します。

当メタデータ指定を使用しなかった場合のデフォルトは、1トラックあたり単音のモードになります。

引数[1] (maxPoly)

1トラックあたりの最大発音数を整数で指定します。

設定範囲は 1 ~ 64 です。

引数[2] (maxCmd)

maxCmdとして文字列「force」と記述すると、全てのトラックのデフォルトが、最大発音数での発音になります。(文字列「force」は大文字・小文字問わず) maxCmdは省略可能で、省略した場合は全てのトラックのデフォルトが、単音での発音になります。

【例1】

```
#ML:USING-POLY 4
#MB:ENV@A 1 { 15,0,&, N,0,15, R,150,0 }
@EA1 L8 @PL1 cegb dfa>d< bgegcrrr
      @PL cegb dfa>d< bgegcrrr;
```

force指定を省略しています。最初は単音で演奏し、その後「#ML:USING-POLY」で定義した最大数（この場合4）を上限にリリースが重なり合って演奏しています。

【例2】

```
#ML:USING-POLY 4,force
#MB:ENV@A 1 { 15,0,&, N,0,15, R,150,0 }
@EA1 L8 cegb dfa>d< bgegcrrr;
```

force指定があるため、最初、@PLコマンドが無くとも「#ML:USING-POLY」で定義した最大数（この場合4）を上限にリリースが重なり合って演奏しています。

9.2. @PL：ポリフォニックモード発動

記述	@PL[1]
種別	MML コマンド

【解説】

「#ML:USING-POLY」が、「force」指定なしで記述されている場合、@PL コマンドを1度でも記述したトラックだけが、ポリフォニックモードの発動対象となります。なおかつ、@PL コマンドによって、最大発音数の絞り込みが可能です。（絞り込みは重なり過ぎの抑制目的）

引数[1]には、同時発音数を整数で記述します。
ただし、「#ML:USING-POLY」で指定した最大発音数が上限となります。
引数[1]の下限は、1 です。

引数[1]は省略することができます。省略した場合、「#ML:USING-POLY」で指定した最大発音数と同じ値を指定したものとみなされます。

【備考1】

ポリフォニックモードが発動されたトラックでは、「#ML:USING-POLY」で指定した最大発音数の分だけ、内部的に発音チャンネルを固定的に確保するので、メモリ使用量が増えます。

ポリフォニックモードが発動されていないトラックでは、通常モードのままとなり、内部的に多重チャンネル化を行わないので、メモリ使用量は増えません。

【備考2】

「#ML:USING-POLY」で「force」指定されている場合は、@PL コマンドの記述がなくとも、全てのトラックで強制的にポリフォニックモードが発動し、なおかつ、トラック先頭における最大発音数は「#ML:USING-POLY」で指定した最大発音数と同じになります。

【備考3】

@PL コマンドの記述をトラック先頭に配置しなかった場合、たとえば中間に1つだけ配置した場合、そのトラック先頭における同時発音数は、強制的に最大数になります。

ポリフォニックモード発動対象トラックにおいて、@PL コマンドが現れるまでの同時発音数は不定のように見えますが、内部では最大発音数として処理される仕様になっているので注意してください。

記述の明確化のため、ポリフォニックモード発動トラックでは、トラック先頭位置での @PL コマンド記述は、省略しないようにしてください。

(トラック記述中での @PL コマンドによる最大発音数の変更は随時 OK です)

【備考 4】

「#ML:USING-POLY」が記述されていない場合、@PL コマンドを記述しても、ポリフォニックモードは発動せず、無視されます。

9.3. @PM：ポリフォニック方式の選択

記述	@PM[1]
種別	MML コマンド

【解説】

当コマンドは、ポリフォニックモードが発動しているトラックでのみ有効です。
(ポリフォニックモードが発動していないトラックでは無視されます)

ポリフォニックモードでは、同時発音数が上限を超えるノートオン要求が来たとき、一定の条件に従って古い発音をやめて新しい発音に上塗りしていきます。

当コマンドでは、引数[1]により、その「一定の条件」を、
0 または 1 で選択します。

@PM0 のとき：

デフォルトのポリフォニックモードでの上塗り方式です。

まず、無音チャンネルを検索し、見つければそこで発音させます。

無音チャンネルが見つからなかった場合、最も古い発音に、最も新しい発音を上塗りします。

最も古い発音が同時発音により複数存在する場合は、最初に見つかったものに機械的に上塗りされます。

@PM1 のとき：

(1) ノートオフされて余韻状態で発音しているチャンネルを検索し、その中から、これからノートオンしようとしている音程と同じ音程のものがあれば、新しい発音を上塗りします。

(2) (1)で見つからなければ、無音チャンネルを検索し、見つければそこで発音させます。

(3) (2)においても見つからなければ、最も古い発音に、最も新しい発音を上塗りします。

最も古い発音が同時発音により複数存在する場合は、最初に見つかったものに機械的に上塗りされます。

【備考】

@PM1 は、鍵盤楽器を意識したモードです。

9.4. [...] : 和音記法

記述 []

種別 MML 擬似コマンド

【解説】

和音記法です。1つのトラックで和音を記述することができます。

ポリフォニックモードが発動しているトラックで、音符を大カッコ [] で括ると、複数音を同時に鳴らすことができます。

Rコマンドによって、大カッコ内での発音タイミングをずらすこともできます。

ポリフォニックモードを発動するには、「#ML:USING-POLY」を記述し、対象となるトラックで @PLコマンドを記述します。

【機能制限】

- ・和音記法 [] 内では、スラーは使用できません。

【使用例】

L4 CEG

ピアノロール												
G												
E												
C												

L4 [CEG]

ピアノロール												
G												
E												
C												

L8 [C4.EG]E.

ピアノロール												
G												
E												
C												

L8 [C4.RERG]E.

ピアノロール												
G												
E												
C												

L8 [C4.RE4.RG]G.

ピアノロール												
G												
E												
C												

L8 [C4.EGRFARE]F.

ピアノロール												
A												
G												
F												
E												
D												
C												

10. 装飾関連

10.1. XA：パラメータへの加算

記述	XA[1],[2]
種別	MML 擬似コマンド

【解説】

宛先になるコマンドのパラメータ値に、指定値を加算します。
適用範囲は、当コマンド指定以降、当該トラック終了までです。

想定している用途は、既に取り上げたMMLトラックの文字列を、マクロ参照でそのまま再利用することです。再利用時、トラック先頭に当コマンドを書き足して、デチューン値やサブフォーム番号などをずらすのに使用します。

指定する引数は次の通りです。

引数[1]...加算数値 (addNum)

引数[2]...宛先 (destCmd)

引数はカンマで区切って指定します。

トラック先頭におけるデフォルトは、全ての宛先に対し加算値0です。

引数[1] (addNum)

加算したい数値を指定します。

負数や、小数以下の指定も記述することもできます。

ただし、加算後のパラメータが有効な設定値におさまるようにしてください。

加算後のパラメータが無効な設定値の場合、エラーは XAコマンド ではなく、宛先のコマンドに対して表示されます。

整数のみ受け付けるコマンドでは、加算値の少数以下は切り捨てが行われます。

引数[2] (destCmd)

加算対象となる宛先のコマンドを、2文字の識別子で指定します。

【XAコマンドの宛先となる識別子一覧】

識別子	宛先	内容
SF	@	音源サブフォーム（サブモジュール）番号
MV	MV	ミキシングボリューム
VE	V	ボリューム
VL	VL	ボリューム for @LA
NS	NS	ノートシフト
DT	@D	デチューン
EA	@EA	音量エンベロープ
EF	@EF	フィルタエンベロープ
LP	@LP	ピッチLFO
LA	@LA	音量LFO
LB	@LB	パンポットバランスLFO
LF	@LF	フィルタLFO
LY	@LY	YコントロールLFO
FL	@F	フィルタ

（識別子は大文字・小文字を問わず）

【例】

```
$A= @d0 cdefgfed @d-10 cegec;
$A;
xa8,dt $A;
```

1つのマクロ定義でトラックを2つ定義し、2トラック目全体のデチューン設定に+8加算しています。

10.2. XM：パラメータへの乗算

記述 XM[1],[2]

種別 MML 擬似コマンド

【解説】

宛先になるコマンドのパラメータ値に、指定値を乗算します。
適用範囲は、当コマンド指定以降、当該トラック終了までです。

想定している用途は、既にかき上げたMMLトラックの文字列を、マクロ参照でそのまま再利用することです。再利用時、トラック先頭に当コマンドをかき足して、パンポット値をずらすのに使用します。

指定する引数は次の通りです。

引数[1]...乗算数値 (mulNum)

引数[2]...宛先 (destCmd)

引数はカンマで区切って指定します。

トラック先頭におけるデフォルトは、全ての宛先に対し乗算しません。

引数[1] (mulNum)

乗算したい数値を指定します。

負数や、小数以下の指定も記述することもできます。

ただし、乗算後のパラメータが有効な設定値におさまるようにしてください。

乗算後のパラメータが無効な設定値の場合、エラーは XMコマンド ではなく、宛先のコマンドに対して表示されます。

整数のみ受け付けるコマンドでは、乗算値の少数以下は切り捨てが行われます。

引数[2] (destCmd)

乗算対象となる宛先のコマンドを、2文字の識別子で指定します。

【XMコマンドの宛先となる識別子一覧】

識別子	宛先	内容
PP	@P	パンポット
PL	P	パンポット・レガシーモード

(識別子は大文字・小文字を問わず)

【例】

```
$A= @p100 cdefgfed @p-100 cegec;  
$A;  
xm-1,pp $A;
```

1つのマクロ定義でトラックを2つ定義し、2トラック目全体のパンポット設定に(-1)を乗算し、左右反転しています。

10.3. @KT：キーイベント（トリガモード）

記述 @KT[1],[2]

種別 MML 擬似コマンド

【解説】

キーイベント機能をトリガモードで発動させます。
 キーイベント機能のトリガモードとは、ノートオンの都度、事前に定義した「#MB:KEYEVENT@TRG」によるテーブルに従って順番に、指定種別のコマンドを発行する機能です。異なる宛先であれば、複数の発動も可能です。

指定する引数は次の通りです。

引数[1]...キーイベント番号 (ktNum)

引数[2]...宛先 (destCmd)

引数はカンマで区切って指定します。

トラック先頭におけるデフォルトは、全ての宛先に対し当機能無効です。

引数[1] (ktNum)

「#MB:KEYEVENT@TRG」で定義済みのキーイベント番号を整数で指定します。
 未定義の番号を指定すると、エラーになります。

引数[2] (destCmd)

キーイベント機能発動の宛先となるコマンドを、2文字の識別子で指定します。

【@KTコマンドの宛先となる識別子一覧】

識別子	宛先	内容
SF	@	音源サブフォーム（サブモジュール）番号
VE	V	ボリューム
VL	VL	ボリューム for @LA
PP	@P	パンポット
PL	P	パンポット・レガシーモード
NZ	@N	ノイズサイクル
PW	@W	パルス幅（デューティ比）
DT	@D	デチューン
EA	@EA	音量エンベロープ

識別子	宛先	内容
EF	@EF	フィルタエンベロープ
LP	@LP	ピッチ L F O
LA	@LA	音量 L F O
LB	@LB	パンポットバランス L F O
LF	@LF	フィルタ L F O
LY	@LY	Yコントロール L F O
FL	@F	フィルタ

(識別子は大文字・小文字を問わず)

【キーイベントの発動をやめたい場合】

発動済みのキーイベント機能の停止するには、宛先ごとのコマンドを、あらためて発行します。

例えば、サブフォーム宛のキーイベント機能の発動と停止は次の通りです。

```
@KT1,SF CDEFG @1 GFEDC;
```

この場合、CDEFG はキーイベント機能有効、GFEDC は宛先「SF」のキーイベント機能が全て無効になって、@1 で演奏されます。

同様に、他の宛先でのキーイベント機能の発動と停止は次のようになります。

```
@KT1,VE CDEFG V13 GFEDC;
```

```
@KT1,VL CDEFG VL13 GFEDC;
```

```
@KT1,PP CDEFG @P0 GFEDC;
```

```
@KT1,PL CDEFG P0 GFEDC;
```

```
@KT1,NZ CDEFG @N1 GFEDC;
```

//ノイズは@N1,NZWなどで事前にノイズ宛先の指定が必要。

//ノイズ宛先を指定しなかった場合デフォルト宛先になる。

```
@KT1,PW CDEFG @W4,8 GFEDC;
```

```
@KT1,DT CDEFG @D0 GFEDC;
```

```
@KT1,EA CDEFG @EA1 GFEDC;
```

```
@KT1,EF CDEFG @EF1 GFEDC;
```

```
@KT1,LP CDEFG @LP0 GFEDC; //@LPZでも解除可能
```

```
@KT1,LA CDEFG @LA0 GFEDC; //@LAZでも解除可能
```

```
@KT1,LB CDEFG @LB0 GFEDC; //@LBZでも解除可能
```

```
@KT1,LF CDEFG @LF0 GFEDC; //@LFZでも解除可能
```

```
@KT1,LY CDEFG @LY0 GFEDC; //@LYZでも解除可能
```

```
@KT1,FL CDEFG @F1 GFEDC;
```

【使用例】

```
#MB:KEYEVENT@TRG 1 {  
    0, //repeat entry index  
    0,1,-1,1,-1,1,-1,0  
}  
l4 @KT1,PL  
o4 cdefgfedcdefgfedcdefgfedc;
```

この場合、音符が発音される都度、

P0、P1、P-1、P1、P-1、P1、P-1、P0

のパターンでパンポットレガシーコマンドが発行され、パターンの最後の次は、また先頭 (index:0) のP0から繰り返します。

10.4. @KC：キーイベント（音程モード）

記述 @KC[1],[2]

種別 MML 擬似コマンド

【解説】

キーイベント機能を音程モードで発動させます。

キーイベント機能の音程モードとは、ノートオンの都度、キーコードに応じて、事前に定義した「#MB:KEYEVENT@CODE」によるテーブルに従って、指定種別のコマンドを発行する機能です。異なる宛先であれば、複数の発動も可能です。

指定する引数は次の通りです。

引数[1]...キーイベント番号 (kcNum)

引数[2]...宛先 (destCmd)

引数はカンマで区切って指定します。

トラック先頭におけるデフォルトは、全ての宛先に対し当機能無効です。

引数[1] (kcNum)

「#MB:KEYEVENT@CODE」で定義済みのキーイベント番号を整数で指定します。未定義の番号を指定すると、エラーになります。

引数[2] (destCmd)

キーイベント機能発動の宛先となるコマンドを、2文字の識別子で指定します。

【@KCコマンドの宛先となる識別子一覧】

識別子	宛先	内容
SF	@	音源サブフォーム（サブモジュール）番号
VE	V	ボリューム
VL	VL	ボリューム for @LA
PP	@P	パンポット
PL	P	パンポット・レガシーモード
NZ	@N	ノイズサイクル
PW	@W	パルス幅（デューティ比）
DT	@D	デチューン
EA	@EA	音量エンベロープ

識別子	宛先	内容
EF	@EF	フィルタエンベロープ
LP	@LP	ピッチ L F O
LA	@LA	音量 L F O
LB	@LB	パンポットバランス L F O
LF	@LF	フィルタ L F O
LY	@LY	Yコントロール L F O
FL	@F	フィルタ

(識別子は大文字・小文字を問わず)

【キーイベントの発動をやめたい場合】

発動済みのキーイベント機能の停止するには、宛先ごとのコマンドを、あらためて発行します。

例えば、サブフォーム宛のキーイベント機能の発動と停止は次の通りです。

```
@KC1,SF CDEFG @1 GFEDC;
```

この場合、CDEFG はキーイベント機能有効、GFEDC は宛先「SF」のキーイベント機能が全て無効になって、@1 で演奏されます。

同様に、他の宛先でのキーイベント機能の発動と停止は次のようになります。

```
@KC1,VE CDEFG V13 GFEDC;
```

```
@KC1,VL CDEFG VL13 GFEDC;
```

```
@KC1,PP CDEFG @P0 GFEDC;
```

```
@KC1,PL CDEFG P0 GFEDC;
```

```
@KC1,NZ CDEFG @N1 GFEDC;
```

//ノイズは@N1,NZWなどで事前にノイズ宛先の指定が必要。

//ノイズ宛先を指定しなかった場合デフォルト宛先になる。

```
@KC1,PW CDEFG @W4,8 GFEDC;
```

```
@KC1,DT CDEFG @D0 GFEDC;
```

```
@KC1,EA CDEFG @EA1 GFEDC;
```

```
@KC1,EF CDEFG @EF1 GFEDC;
```

```
@KC1,LP CDEFG @LP0 GFEDC; //@LPZでも解除可能
```

```
@KC1,LA CDEFG @LA0 GFEDC; //@LAZでも解除可能
```

```
@KC1,LB CDEFG @LB0 GFEDC; //@LBZでも解除可能
```

```
@KC1,LF CDEFG @LF0 GFEDC; //@LFZでも解除可能
```

```
@KC1,LY CDEFG @LY0 GFEDC; //@LYZでも解除可能
```

```
@KC1,FL CDEFG @F1 GFEDC;
```


10.5. @KL : キーイベント (音長モード)

記述 @KL[1],[2]

種別 MML 擬似コマンド

【解説】

キーイベント機能を音長モードで発動させます。

キーイベント機能の音長モードとは、ノートオンの都度、tickカウント数に応じて、事前に定義した「#MB:KEYEVENT@LEN」によるテーブルに従って、指定種別のコマンドを発行する機能です。異なる宛先であれば、複数の発動も可能です。

指定する引数は次の通りです。

引数[1]...キーイベント番号 (klNum)

引数[2]...宛先 (destCmd)

引数はカンマで区切って指定します。

トラック先頭におけるデフォルトは、全ての宛先に対し当機能無効です。

引数[1] (klNum)

「#MB:KEYEVENT@LEN」で定義済みのキーイベント番号を整数で指定します。未定義の番号を指定すると、エラーになります。

引数[2] (destCmd)

キーイベント機能発動の宛先となるコマンドを、2文字の識別子で指定します。

【@KLコマンドの宛先となる識別子一覧】

識別子	宛先	内容
SF	@	音源サブフォーム (サブモジュール) 番号
VE	V	ボリューム
VL	VL	ボリューム for @LA
PP	@P	パンポット
PL	P	パンポット・レガシーモード
NZ	@N	ノイズサイクル
PW	@W	パルス幅 (デューティ比)
DT	@D	デチューン
EA	@EA	音量エンベロープ

識別子	宛先	内容
EF	@EF	フィルタエンベロープ
LP	@LP	ピッチ L F O
LA	@LA	音量 L F O
LB	@LB	パンポットバランス L F O
LF	@LF	フィルタ L F O
LY	@LY	Yコントロール L F O
FL	@F	フィルタ

(識別子は大文字・小文字を問わず)

【キーイベントの発動をやめたい場合】

発動済みのキーイベント機能の停止するには、宛先ごとのコマンドを、あらためて発行します。

例えば、サブフォーム宛のキーイベント機能の発動と停止は次の通りです。

```
@KL1,SF CDEFG @1 GFEDC;
```

この場合、CDEFG はキーイベント機能有効、GFEDC は宛先「SF」のキーイベント機能が全て無効になって、@1 で演奏されます。

同様に、他の宛先でのキーイベント機能の発動と停止は次のようになります。

```
@KL1,VE CDEFG V13 GFEDC;
```

```
@KL1,VL CDEFG VL13 GFEDC;
```

```
@KL1,PP CDEFG @P0 GFEDC;
```

```
@KL1,PL CDEFG P0 GFEDC;
```

```
@KL1,NZ CDEFG @N1 GFEDC;
```

//ノイズは@N1,NZWなどで事前にノイズ宛先の指定が必要。

//ノイズ宛先を指定しなかった場合デフォルト宛先になる。

```
@KL1,PW CDEFG @W4,8 GFEDC;
```

```
@KL1,DT CDEFG @D0 GFEDC;
```

```
@KL1,EA CDEFG @EA1 GFEDC;
```

```
@KL1,EF CDEFG @EF1 GFEDC;
```

```
@KL1,LP CDEFG @LP0 GFEDC; //@LPZでも解除可能
```

```
@KL1,LA CDEFG @LA0 GFEDC; //@LAZでも解除可能
```

```
@KL1,LB CDEFG @LB0 GFEDC; //@LBZでも解除可能
```

```
@KL1,LF CDEFG @LF0 GFEDC; //@LFZでも解除可能
```

```
@KL1,LY CDEFG @LY0 GFEDC; //@LYZでも解除可能
```

```
@KL1,FL CDEFG @F1 GFEDC;
```

【使用例】

```
#ML:TICKUNIT 192
#MB:KEYEVENT@LEN 1 {
  10000, 48, 12,
  10000, 24, 8,
  10000, 12, 15,
  2, 1, 3,
  1, 0, 10 }
t120 @@pls l4 @KL1,VE
o4 c4c4c4c4 c8c16c8c16c8c16 r2 c6c6c6c6
c%9 c%12 c%9 c%12 c%9 c%12 c%9 c%12;
```

この場合、音符が発音されるたびに、

- 4分音符だったら V12、
- 8分音符だったら V8、
- 16分音符だったら V15、
- tickカウント数が奇数だったら V3、
- 上記のいずれでもない場合は V10

がコマンド発行されます。

tickカウント数を割る数に10000と記述している部分は、使用される最大の音長よりも大きい数であれば良く、剰余が各音長のtickカウント数に合致するように指定します。

また、tickカウント数を1で割った剰余は必ず0になるため、条件の最後に、それまでのどの条件にも当てはまらなかった場合のコマンド発行を「1,0,10」などとして配置しています。

MML内で登場する音長の最大値は「report」チェックボックスを有効にして「PLAY」ボタンをクリックすることによるレポート表示で確認することが出来ます。

10.6. #MB:KEYEVENT@TRG : @KT用キーイベントデータ定義

記述	#MB:KEYEVENT@TRG [1] {[2]}
種別	メタデータ定義

【解説】

@KTコマンドで使用する、トリガモードのキーイベントパターンを定義します。

指定する内容は次の通りです。

- [1]...パターン番号 【1個の整数】
- [2]...パターンデータ 【複数のパラメータ】

パターンデータの開始は「 { 」で認識され、終了は「 } 」で検知されます。パターンデータの各パラメータは、カンマ区切りで記述します。スペースや改行は読み飛ばされます。（最後のパラメータの後にカンマがあっても可）

パターン番号：

パターン番号は、@KTコマンドの引数[1] (ktNum) で使用する番号と合致するように定義します。例えば「@KT3,SF」のように記述して使用したい場合は「3」をパターン番号にします。

パターン番号の設定範囲は 0 ~ 1023 です。

パターンデータ：

パターンデータのフォーマットは、最初の1個が繰り返しエントリ番号、それ以降は実際に使用される、宛先コマンドへのパラメータを数値で記述します。数値は少数以下の指定も受け付けます。

宛先コマンドが整数で受け付けるコマンドだった場合、少数以下は使用時に四捨五入で整数に丸められて処理されます。

繰り返しエントリ番号の設定範囲は、

0 ~ (パターン定義個数-1)

の整数になります。

特例で、繰り返しエントリ番号に「-1」と記述すると、繰り返しをしない設定になります。

宛先コマンドへのパラメータの定義個数は 1 ~ 2048個の範囲になります。

このため、パターンデータとしては2個以上の数値定義が必要です。

【例】

```
#MB:KEYEVENT@TRG 1 {  
    0, //repeat entry index  
    0,1,-1,1,-1,1,-1,0  
}  
l4 @KT1,PL  
o4 cdefgfedcdefgfedcdefgfedc;
```

この場合、音符が発音される都度、

P0、P1、P-1、P1、P-1、P1、P-1、P0

のパターンでパンポットレガシーコマンドが発行され、パターンの最後の次は、また先頭 (index:0) のP0から繰り返します。

10.7. #MB:KEYEVENT@CODE : @KC用キーイベントデータ 定義

記述	#MB:KEYEVENT@CODE [1] {[2]}
種別	メタデータ定義

【解説】

@KCコマンドで使用する、音程モードのキーイベントパターンを定義します。

指定する内容は次の通りです。

- [1]...パターン番号 【1個の整数】
- [2]...パターンデータ 【複数のパラメータ】

パターンデータの開始は「 { 」で認識され、終了は「 } 」で検知されます。パターンデータの各パラメータは、カンマ区切りで記述します。スペースや改行は読み飛ばされます。（最後のパラメータの後にカンマがあっても可）

パターン番号：

パターン番号は、@KCコマンドの引数[1] (kcNum) で使用する番号と合致するように定義します。例えば「@KC3,SF」のように記述して使用したい場合は「3」をパターン番号にします。

パターン番号の設定範囲は 0 ~ 1023 です。

パターンデータ：

パターンデータのフォーマットは、120個の数値を定義するだけです。

数値は少数以下の指定も受け付けます。

定義個数は120個より多くても少なくてもエラーになります。

宛先コマンドが整数で受け付けるコマンドだった場合、少数以下は使用時に四捨五入で整数に丸められて処理されます。

120個の各数値の割り当ては、1個目がオクターブ0の「ド」が呼ばれた際に使用される値となり、2個目がオクターブ0の「ド#」用の値になります。以降同様に半音ずつ上がっていき、120個目はオクターブ9の「シ」用の値になります。

12平均律のピッチスケールなどで、120個の範囲外の音程が呼ばれた場合、例えばオクターブ0の「ド」よりも低い音程が呼ばれた場合は、オクターブ0の

10.8. #MB:KEYEVENT@LEN : @KL用キーイベントデータ定義

記述	#MB:KEYEVENT@LEN [1] {[2]}
種別	メタデータ定義

【解説】

@KLコマンドで使用する、音程モードのキーイベントパターンを定義します。

指定する内容は次の通りです。

- [1]...パターン番号 【1個の整数】
- [2]...パターンデータ 【複数のパラメータ】

パターンデータの開始は「 { 」で認識され、終了は「 } 」で検知されます。パターンデータの各パラメータは、カンマ区切りで記述します。スペースや改行は読み飛ばされます。（最後のパラメータの後にカンマがあっても可）

パターン番号：

パターン番号は、@KLコマンドの引数[1] (klNum) で使用する番号と合致するように定義します。例えば「@KL3,SF」のように記述して使用したい場合は「3」をパターン番号にします。

パターン番号の設定範囲は 0 ~ 1023 です。

パターンデータ：

パターンデータのフォーマットは、3個1組の数値の連続で構成されます。

- (A) 1個目 (0より大きい整数) :
音長tickカウント数を割る数。除数。
- (B) 2個目 (0以上の整数) :
音長tickカウント数を(A)で割った「剰余」と比較する数。
- (C) 3個目 (整数または浮動小数点数) :
(B)での比較結果が「一致」だった場合に実行する宛先コマンドへのパラメータ。

以上の3個1組を最大256組まで定義できます。

(C)において、宛先コマンドが整数で受け付けるコマンドだった場合、少数以下は使用時に四捨五入で整数に丸められて処理されます。

【例】

```
#ML:TICKUNIT 192
#MB:KEYEVENT@LEN 1 {
  10000, 48, 12,
  10000, 24, 8,
  10000, 12, 15,
  2, 1, 3,
  1, 0, 10 }
t120 @@pls l4 @KL1,VE
o4 c4c4c4c4 c8c16c8c16c8c16 r2 c6c6c6c6
c%9 c%12 c%9 c%12 c%9 c%12 c%9 c%12;
```

この場合、音符が発音されるたびに、

- 4分音符だったら V12、
- 8分音符だったら V8、
- 16分音符だったら V15、
- tickカウント数が奇数だったら V3、
- 上記のいずれでもない場合は V10

がコマンド発行されます。

tickカウント数を割る数に10000と記述している部分は、使用される最大の音長よりも大きい数であれば良く、剰余が各音長のtickカウント数に合致するように指定します。

また、tickカウント数を1で割った剰余は必ず0になるため、条件の最後に、それまでのどの条件にも当てはまらなかった場合のコマンド発行を「1,0,10」などとして配置しています。

MML内で登場する音長の最大値は「report」チェックボックスを有効にして「PLAY」ボタンをクリックすることによるレポート表示で確認することが出来ます。

11. 付加情報関連

11.1. #ML:TITLE : 曲のタイトル

記述	#ML:TITLE [1]
種別	メタデータ指定

【解説】

曲のタイトル文字列を定義します。

引数[1]には、定義したい文字列（全角文字も可）を記述します。

改行で定義終了になります。

引数[1]の直前には、区切りとしてスペースが最低1文字は必要ですが、複数のスペースを置いても良く、置いたスペースは全て読み飛ばされます。結果的に、引数[1]の先頭文字は必ずスペース以外の文字になります。

「区切りとしてのスペース」とは、半角スペースまたはTAB文字になります。全角スペースは通常文字扱いとなり、「区切りとしてのスペース」としては認識されません。

曲のタイトルを複数行定義したい場合は、「#ML:TITLE」定義を複数、記述します。ただし、原則として表示保証対象は最初の行のみになります。

11.2. #ML:ARTIST : 曲のアーティスト

記述	#ML:ARTIST [1]
種別	メタデータ指定

【解説】

曲のアーティスト（作曲社名・編曲者名など）の文字列を定義します。引数[1]には、定義したい文字列（全角文字も可）を記述します。改行で定義終了になります。

引数[1]の直前には、区切りとしてスペースが最低1文字は必要ですが、複数のスペースを置いても良く、置いたスペースは全て読み飛ばされます。結果的に、引数[1]の先頭文字は必ずスペース以外の文字になります。

「区切りとしてのスペース」とは、半角スペースまたはTAB文字になります。全角スペースは通常文字扱いとなり、「区切りとしてのスペース」としては認識されません。

曲のアーティストを複数行定義したい場合は、「#ML:ARTIST」定義を複数、記述します。ただし、原則として表示保証対象は最初の行のみになります。

11.3. #ML:COMMENT : 曲のコメント

記述	#ML:COMMENT [1]
----	-----------------

種別	メタデータ指定
----	---------

【解説】

曲のコメント文字列を定義します。

引数[1]には、定義したい文字列（全角文字も可）を記述します。

改行で定義終了になります。

引数[1]の直前には、区切りとしてスペースが最低1文字は必要ですが、複数のスペースを置いても良く、置いたスペースは全て読み飛ばされます。

結果的に、引数[1]の先頭文字は必ずスペース以外の文字になります。

「区切りとしてのスペース」とは、半角スペースまたはTAB文字になります。全角スペースは通常文字扱いとなり、「区切りとしてのスペース」としては認識されません。

曲のコメントを複数行定義したい場合は、「#ML:COMMENT」定義を複数、記述します。ただし、原則として表示保証対象は最初の行のみになります。

11.4. #ML:CODING : MML 作成者名

記述	#ML:CODING [1]
種別	メタデータ指定

【解説】

MML 作成者名の文字列を定義します。

引数[1]には、定義したい文字列（全角文字も可）を記述します。

改行で定義終了になります。

引数[1]の直前には、区切りとしてスペースが最低1文字は必要ですが、複数のスペースを置いても良く、置いたスペースは全て読み飛ばされます。

結果的に、引数[1]の先頭文字は必ずスペース以外の文字になります。

「区切りとしてのスペース」とは、半角スペースまたはTAB文字になります。全角スペースは通常文字扱いとなり、「区切りとしてのスペース」としては認識されません。

MML 作成者名を複数行定義したい場合は、「#ML:CODING」定義を複数、記述します。ただし、原則として表示保証対象は最初の行のみになります。

[end of file]