

Unity PSG Player - MMLリファレンス

書式の基本

MMLは、基本的に「コマンド記号+パラメータ」のコマンドブロックの並びで構成され、1文字目から順に処理されます。

ブロック間にスペースや改行、またはコマンドに定義されていない文字やパラメータとして認識されない数字がある場合は無視されます。（※[1行コメント](#)や[タイ"&"](#)を除く。）

ただし、コマンド記号とパラメータの間にスペースなどが入ると、パラメータの値が認識されず、コマンドのデフォルト値が適用されます。

この制限を利用して、1拍ごとにスペースを入れたり、小節ごとに"
"などを入れて、可読性を良くすることができます。

コマンド記号としてのアルファベットは、大文字と小文字を区別して認識します。

同じアルファベットでも、大文字と小文字では機能が違うので注意が必要です。

パラメータとして扱う数値は、10進数でのみ処理できます。

また、パラメータの数値は整数のみで、0.5のような小数、1/3のような分数などは扱いません。

音長について

音符、休符に付随する<len>の音長表現は、楽譜表記のn分音符と同等になります。

全音符は「1」、2分音符は「2」、16分音符は「16」というように表します。

連符の場合、例えば3連8分音符は1小節の中に12音（4分音符の中に3音）入るので、12分音符と同等なので「12」と表します。

PSG Playerでは、デフォルトでは4分音符を960ティックとして計算するので、7連符のような960を割り切れない連符の場合は微妙にマルチチャンネルでズレが生じます。

特にループ再生の場合、段々とズレが大きくなっていきます。

これを回避するには、PSG PlayerのtickPerNote（分解能）を連符の倍数にするか、割り切れる長さで擬似的な連符にしてください。

（例 7連16分(28分) × 7 --> 3連32分(24分) × 3 + 32分 × 4）

また、数値の後に"."（ドット）をつけると、付点音符となり、音長に半分の長さが加算されます。"."は複数指定可能です。

（例 4... = 4分音符 + 8分音符 + 16分音符 + 32分音符）

コマンド一覧

- **音符、休符**
 - **c,d,e,f,g,a,b<sign><len>** 音符
 - **r<len>** 休符
 - **n<num>,<len>** ノートナンバー
- **時間、音長**
 - **t<num>** テンポ
 - **l<len>** デフォルト音長
 - **G<num>** ゲート

- **&** タイ
- **音程**
 - **o<num>** オクターブ
 - **>** オクターブ上げ
 - **<** オクターブ下げ
 - **T<num>** チューニング
 - **S<sign><num>** スイープ
 - **M<num>** LFO (ビブラート)
- **音量**
 - **v<num>** 音量
 - **V<num>** エンベロープ
- **音色**
 - **@<num>** 音色
- **制御**
 - **L** ループ
 - **[~]<num>** リピート
- **データ定義**
 - **V<num>{<vol>,~,<|>,~,<vol>}** エンベロープ定義
 - **M<num>{<delay>,<deapth>,<speed>}** LFO定義
- **コメント**
 - **/* ~ */** 範囲コメント
 - **; または //** 1行コメント
- **トラックのヘッダー**
 - **<ch> <string>** チャンネルMML

コマンド詳細

音符、休符

c,d,e,f,g,a,b<sign><len> 音符

- パラメーター
 - **<sign>** +,-半音上げ下げ (省略時: なし)
 - **<len>** 音長[1~128] (省略時: デフォルト音長)

指定の音階で発音します。

コマンドの直後に+をつけると半音上がった音になり、コマンドの直後に-をつけると半音下がります。

+や-は連続して記述することができ、ダブルシャープやダブルフラットなどに対応できます。

(例 **b-ag+gf+ = g+++g++g+gg- //** 左と右では同じ音階となる。)

なお、ノイズ音色の場合は16段階になるので、オクターブ4のド (o4c) を60としたノートナンバーに変換し、16で割った剰余 (0~15) で音程が決まります。

また、ノイズは高い周波数から順に割り当てられているので注意が必要です。

<len>は音長で、何分音符かを数字で書きます。

(例 **f1 //** 全音符、 **c8 //** 8分音符)

<len>を省略した場合は、"**I**"[コマンド](#)で設定したデフォルト音長になります。
付点音符は"."（ドット）で記述し、音長に半分の長さが加算されます。

（例 c4. // 4分音符+ 8分音符の長さ）

付点は連続記述が可能で、2付点音符などに対応します。

（例 a2.. // 2分音符+ 4分音符+ 8分音符の長さ）

r<len> 休符

- パラメーター
 - <len> 音長[1~128]（省略時：デフォルト音長）

指定の時間だけ発音しません。<len>は音長です。

<len>を省略した場合は、"**I**"[コマンド](#)で設定したデフォルト音長になります。

n<num>,<len> ノートナンバー

- パラメーター
 - <num> ノートナンバー[1~127]（省略時：60）
 - <len> 音長[1~128]（省略時：デフォルト音長）

ノートナンバーで指定した音階で発音します。

ノートナンバーは、オクターブ4のドの音（o4c）を60として、鍵盤の各鍵に割り当てられた番号です。
ノイズ音色の場合は、ノートナンバーを16で割った剰余（0~15）で音程が決まります。

ノートナンバーと音長は","（カンマ）で区切ります。

<len>は音長です。

<len>を省略した場合は、"**I**"[コマンド](#)で設定したデフォルト音長になります。

また<len>を省略する場合、","も省略できます。

（例 I16 n60,8. n69）

時間、音長

t<num> テンポ

- パラメーター
 - <num> テンポ[1~255]（省略時：120）

テンポ（1分間に入る4分音符の数）を指定します。

l<len> デフォルト音長

- パラメーター
 - <len> デフォルト音長[1~128]（省略時：4）

デフォルト音長を設定します。
(音符、休符 の<len>を省略したときの音長です)

G<num> ゲート

- パラメーター
 - <num> ゲート割合 (%) [1~100] (省略時 : 100)

音符で指定した音長に対して、実際に発音する長さの割合を指定します。
音長はそのままなので、ゲートで指定した割合で発音した後は、残りの割合で無音になります。
(例 **G50 l4 c** は4分音符の50%の長さになるので **G100 l8 c r** と同じになる)

割合を小さくすることでスタッカートに対応できます。
また、音程によっては連続した音符が繋がって聞こえてしまう場合があります。
その場合、ゲートで僅かに発音を短くすることで分離できます。
(例 **t120 o4 l4 G100 aaaa G99 aaaa** // 前半は音が繋がって聞こえる)

& タイ

- パラメーター
 - なし

この記号の前の音と次の音をつなげます。
ただし、つながられるのは同じキーの音のみです。
(例 **a4& a16** // a(ラ)の音を4分音符 + 1 6 部音符分発音する。)

また、前の音の後にスペースなどを挟まずに"&"を記述すると、前の音のゲートを一時的に100%にします。
このとき、次の音の音長にのみゲートが有効になります。
(例 **G50 a4& a8** と **G50 a4.** は同じ音長ですが、実際の発音は前者が1.25拍、後者が0.75拍になります。)

なお、"&"で音をつなげると、スライドがリセットされません。
(例 **l4 S+10 c S-10 c** // ドから上がって、ドから下がる
l4 S+10 c& S-10 c // ドから上がって、上がったところから下がる)

音程

o<num> オクターブ

- パラメーター
 - <num> オクターブ [2~8] (省略時 : 4)

オクターブを直接指定します。

> オクターブ上げ

- パラメーター
 - なし

オクターブを1上げます。

< オクターブ下げ

- パラメーター
 - なし

オクターブを1下げます。

T<num> チューニング

- パラメーター
 - <num> o4aの周波数 (Hz) [400~480] (省略時: 440)

音階の基準となるオクターブ4のラ (o4a) の音の周波数を直接指定します。

複数チャンネルで同じ音色で同じ音を鳴らす場合、チューニングをずらすことで音に厚みができます。

S<sign><num> スイープ

- パラメーター
 - <sign> 正負記号 (+,-) (省略時: +)
 - <num> 変化量 (セント) [0~1200] (省略時: 0)

1/60秒ごとに指定した変化量で音の高さを変化させます。

変化量の単位はセントで、半音を100で割った値です。

変化量は正負の値を指定でき、正の場合は音が上がっていき、負の場合は音が下がっていきます。

スイープを止める場合は変化量0の"S0" ("S"のみでも可) を設定します。

M<num> LFO (ビブラート)

- パラメーター
 - <num> LFO番号[0~127] (省略時: 0)

※番号0はLFO解除

時間経過で音程を上下に変化するLFO (ビブラート) を設定します。

定義されていないLFO番号を指定した場合は有効になりません。

LFOの内容については[LFO定義](#)を参照してください。

LFOを解除する場合は"M0" ("M"のみでも可) を設定します。

音量

v<num> 音量

- パラメーター
 - <num> 音量[0~15]（省略時：15）

音量を設定します。

V<num> エンベロープ

- パラメーター
 - <num> エンベロープ番号[0~127]（省略時：0）

※番号0はエンベロープ解除

時間経過で音量が変化するソフトウェアエンベロープを設定します。

定義されていないエンベロープ番号を指定した場合は有効になりません（設定されている音量が継続します）。

エンベロープの内容についてはエンベロープ定義を参照してください。

エンベロープを解除する場合は、"V0"（"V"のみでも可）または"**v**"（**音量**）を設定します。

音色

@<num> 音色

- パラメーター
 - <num> 音色番号[0~6]（省略時：2）

音色を設定します。

番号	波形
0	パルス波（12.5%）
1	パルス波（25%）
2	矩形波（50%）
3	パルス波（75%）
4	三角波
5	ノイズ
6	短周期ノイズ

制御

L ループ

- パラメーター
 - なし

シーケンスが最後まで到達すると、"L"（ループ）の位置に戻ります。
ループは再生を停止するまで継続します。

[~]<num> リピート

- パラメーター
 - <num> リピート回数[1~128]（省略時：1）

"["（ブラケット）の間を指定した回数繰り返します。

リピート回数1は繰り返しません。

~~ネスト(入れ子)は出来ません。~~

v0.9.4beta リピートのネストができるようになりました。

データ定義

V<num>{<vol>, ~ ,<|>, ~ ,<vol>} エンベロープ定義

- パラメーター
 - <num> エンベロープ番号[1~127]（省略不可）
 - <vol> 音量[0-15]（省略時：最初は15、2番目以降は直前に設定された音量）
 - <|> ループ位置（省略時：最後の音量を維持）

※エンベロープ番号0は定義不可。

時間経過で音量を変化させるソフトウェアエンベロープを定義します。

エンベロープ番号を設定することで、"V"（エンベロープ）コマンドで呼び出します。

1/60秒ごとに"{"（ブレース）内の音量で左から順に変化します。

音量が省略された場合、最初の音量は15になり、2番目以降は直前に設定された音量となります。

（例 V1{,,14,,13,} = V1{15,15,14,14,13,13} // 左と右では同じエンベロープとなる。）

"{"の中に"|"がある場合、演奏時にエンベロープデータが最後まで到達すると"|"の位置に戻って処理を続けます。

（例 V1{15,13,11,|,9,10} // 音量15、13、11と変化し、9、10を繰り返す。）

"|"が省略された場合はエンベロープ変化後の音量を維持して発音します。

M<num>{<delay>,<deapth>,<speed>} LFO定義

- パラメーター
 - <num> LFO番号[1~127]（省略不可）
 - <delay> LFOがかかるまでの遅延[0~255]（省略時：0）
 - <deapth> 音程の上下幅[0~255]（省略時：0）
 - <speed> 音程の変化速度[1~255]（省略時：1）

※LFO番号0は定義不可。

時間経過で音程が上下に変化するLFO（ビブラート）を定義します。

LFO番号を設定することで、"M"（LFO）コマンドで呼び出します。

<delay>は音の鳴り始めからLFOがかかるまでの時間を1/120秒単位で設定します。

<depth>は音程の上下幅で、単位はセントです。

<speed>は変化速度で、LFOの周波数を1/16Hz単位で指定します。

コメント

`/* ~ */` 範囲コメント

`/* ~ */`で囲まれた中身をコメント扱いにします。

`;` または `//` 1行コメント

`;`、`"`、`"`、`/"`の文字以降から行末までをコメント扱いにします。

トラックのヘッダー

`<ch>` `<string>` チャンネルMML

- パラメーター
 - `<ch>` 送信チャンネル[A,B,C...]
 - `<string>` MML文字列

※*MMLSplitter*のみ有効

`<ch>`で送信するチャンネルを指定し、`<string>`は各チャンネルに送るMMLです。

チャンネルとMMLを区別するため、間にスペースを入れてください。

トラックのヘッダーは行毎に処理されるので、`<ch>`は行頭に記述する必要があります。

チャンネルはMMLSplitterで設定したチャンネル数で決まり、"A"から順に大文字アルファベットで割り当てられます。

また、`<ch>`は"ABC"のように複数連ねての記述も可能です。

この場合は、"A","B","C"各チャンネルに同じMMLが送信されます。

ヘッダーの無い行は、直前に設定された送り先に送信します。