

MXDRV 使用指南

夏普 X68000 电脑音源驱动使用教程

Ripo_2006

文档编写 | 2025/9/24

目录

| | |
|-------------------|----|
| 文件头 | 2 |
| 乐器制作与定义 | 2 |
| FM 乐器 | 2 |
| ADPCM 乐器 | 4 |
| MXC 编译器 | 6 |
| MXC 编译教程 | 6 |
| MXC MML 指令 | 6 |
| NOTE 编译器 | 13 |
| NOTE 编译教程 | 13 |
| 命令行开关 | 13 |
| NOTE MML 指令 | 15 |

文件头

```
#title "曲名"
```

```
#pcmfile "文件名.pdx"
```

设定一个 PDX 文件来播放 ADPCM，如需请查看“[PDX 文件制作](#)”章节来制作 PDX 文件并使用

乐器制作与定义

FM 乐器定义

格式：

```
@1~255={  
    AR, DR, SR, RR, SL, OL, KS, ML, DT1, DT2, AME,  
    AR, DR, SR, RR, SL, OL, KS, ML, DT1, DT2, AME,  
    AR, DR, SR, RR, SL, OL, KS, ML, DT1, DT2, AME,  
    AR, DR, SR, RR, SL, OL, KS, ML, DT1, DT2, AME,  
    CON, FL, OP  
}
```

解释：

AR = Attack Rate (0~31)

DR = Decay Rate (0~31)

SR = Sustain Rate (0~31)

RR = Release Rate (0~15)

SL = Sustain Level (0~15)

用法同等 ADSR；可以查阅[合成器基础：ADSR 包络线 - 知乎](#)

OL = Operator Level (0~127)

Operator 的音量大小，或调制等级

KS = Key Scaling (0~3)

ADSR 包络随音高缩放；音高越低包络缩放越长，音高越高包络缩放越短

ML = Multiplier (0~15)

Operator 的音高倍数；数字为对应倍数

DT1 = Detune 1 (0~7)

设置弱失谐；0为不失谐，1~4为偏高音失谐，5~7为偏低音失谐

DT2 = Detune 2 (0~3)

设置强失谐；数字越高越偏高音失谐

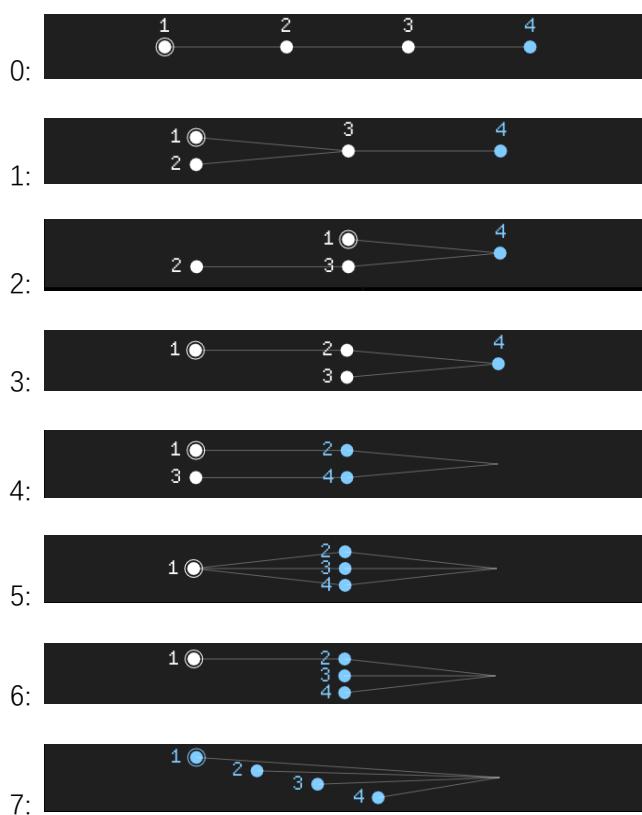
AME = Amplitude Modulation Enable (0~1)

调幅开关；0为关，1为开

CON = Connection. FM synthesis algorithm (0~7)

Operator的连接方式；不同数字对应的连接方式参考下图

OPM 拥有 4 个 Operator, 下面将以数字 1~4 代表展示 8 种不同的 CON



FL = Feedback Level (0~7)

设置 FB；只对 Operator 1 进行 FB，数字越高 FB 越强

OP = Operator Mask. 15 (binary %1111) turns on all 4 operators.

一般直接填 15 就好

$3 = 1 \text{ op} ; 7 = 2 \text{ op} ; 11 = 3 \text{ op} ; 15 = 4 \text{ op}$

ADPCM 乐器

PDX 文件制作

PDX 文件是 MXDRV 中用来制作 PCM 通道的重要文件，以下是如何从零制作 PDX 文件的教程

首先使用 WAV2ADP 来将.wav 文件转换为 raw ADPCM (.pcm) 文件

我们需要在终端窗口界面打入以下指令

```
run68 wav2adp [filename.wav]
```

如果你做对了的话，它会生成 .pcm 文件，在你搞定了所有采样之后，你需要把它们放进 PDX 文件中，不过在此之前，我们需要一个.PDL 文件来指定所有采样来对应各自的音高符号。

一个 PDL 文件就只是一个重命名的 txt 文件。

新建一个.txt 文件，然后往里填入下信息：

```
#ex-pdx 1  
  
@0  
1=文件名 1.pcm  
2=文件名 2.pcm  
3=文件名 3.pcm
```

接下来我会解释这个文件的具体用处

我们将会使用 ex-pdx，在 PDX 格式被创造后，它更新了一个 pdx 文件能够载入更多采样的功能，一个 bank 能够载入大约 90 个采样文件

之后，我们使用 @0 指令来为 bank 中的采样进行定义。我们分别将文件名 1.pcm 映射到了音符音高 1，文件名 2.pcm 映射到了音符音高 2，文件名 3.pcm 映射到了音符音高 3，以此类推，我们可以按照这个顺序来映射剩下的采样。

在完成了以上步骤，我们可以准备开始制作 PDX 文件了

接下来，我们在终端输入以下指令，或是直接将.PDL 文件拖入 MKPDX

```
mkpdx filename.pdl
```

如果你都做对了的话，你应该就做出来一个 PDX 文件了，你太牛逼了！现在你能够在曲子里嗯造 PCM 通道了

PDX 文件使用

要在 PCM 通道使用 PDX 十分简单。

PDX 只能在 PCM 通道播放 (OKIM6258)。

所有采样都在上文被我们以不同音高音符储存，如果你需要在目标轨道播放你想要的采样，你需要输入对应的音高，具体操作如下：

例 1: 一个四分音 kick 动次打次；假设 1=kick.pcm

```
o0 e4 e4 e4 e4  
o0l4[e]4
```

通过上面例子你会发现直接使用音符指令来播放有一点繁琐，因为你需要找到数值对应的音高，特别是遇到多个采样的情况，实在折磨人。不过别急，下面我们来学习更简单的方法！

我们可以直接通过 n# 指令来指定一个音符的数值音高，具体参考通用指令的“音符编写”指令，具体操作如下：

例 1: 一个四分音 kick, 八分音 hat; 在做一个 clap; 假设 1=kick.pcm 2=hat.pcm
3=clap.pcm

```
l8n1n2n3n2  
n1,8n2,8n3,8n2,8
```

关于不同编译器对 PCM 通道的注意事项：

如果你对 PCM 通道极度依赖或是需要更多 PCM 功能，我更推荐 NOTE 编译器

1. 对于 MXC 编译器：

1. PCM 通道只能播放同一频率的采样，你不能改变采样的音高。
2. 只拥有 1 个 PCM 通道
3. 不能调节 PCM 通道的音量。

2. 对于 NOTE 编译器：

1. PCM 可以扩充到 8 个通道
2. 可以调节 PCM 通道音量
3. 可以对采样进行播放不同音高的操作
4. 可以播放 PCM 采样而不是 ADPCM

具体区别请见 MXC 与 NOTE 编译器类目。

MXC 编译器

MXC 编译教程

MXC 编译器使用起来十分简单

在终端输入以下内容：

```
run68 mxc [你的 mml 文件.mml]
```

之后在 mml 目录下将会生成一份 mdx 文件，没啦。

MXC MML 指令

编写规则

1. 所有指令遵循严格的大小写写法，请注意
2. **#** 符号代表数值
3. Ticks 是序列中使用的最小计算单位 (此文档将该单位设为 %)
十六分音为%12；八分音为%24；四分音为%48；二分音为%96；一分音为%192

通用指令

A~H

设置 FM 通道标识符；OPM 拥有 8 个通道，分别对应字母 **A~H**

P

设置 PCM 通道标识符；OKIM6258 可以被分为 8 个通道，MXC 编译器只支持 1 个通道，如有多 PCM 通道需求请转至 NOTE 编译器栏目

在编写音符或是任何效果前，必须先以你想要编写的通道标识符为开头

t#

设置乐曲速度；以每秒的四分音符数量为单位

@t#

设置乐曲速度；直接设置 OPM register B 的数值，默认值为@t200

@#

= 乐器编号 (0 ~ 255)

设置乐器；乐器编号 0 为 PCM 通道专用，不可在 FM 通道使用

F#

= 0 ~ 4

设置 PCM 频率；默认值为 4，具体参考下列

0 = 3.9kHz

1 = 5.2kHz

2 = 7.8kHz

3 = 10.4kHz

4 = 15.6kHz

L

设置全局循环开始点；当前通道所有指令播放完后将会重新播放在 L 符号之后的所有指令

o#

设置八度音程；范围为 o0 – o8， 默认值为 o4

<

降低一个八度音程

>

升高一个八度音程

|#

设置音符默认长度， 默认值为 |4

v#

= FM (0 ~ 15)

设置音量；范围为 v0 ~ v15， 默认为 v8

@v#

= 0 ~ 127

以更高精度设定设置音量；范围为 @v0 ~ @v127

(

音量变化；降低一格当前通道音量

)

音量变化；提高一格当前通道音量

此命令不能通过后面添加数字来改变音量变化大小的幅度，但是你可以通过 []# 指令或者是通过叠加音量变化指令来改变音量大小变化幅度

p#

= 0 ~ 3

设置声像；p0=不输出，p1=左声像，p2=右声像，p3=左右声像，默认值为 p3

D#

= -32767 ~ 32767

设置失谐；以 1/64 的半音为单位

[]#

= 2 ~ 255

设置循环；将会循环 [] 符号内的指令 # 次

/

跳过循环；在 [] 指令内使用，在最后一次循环时，跳过 / 符号后的指令

!

忽略所有 ! 符号后的指令

;

行注释；在同一行的 ; 指令后的所有字符都会被注释

/* */

块注释；在 /* */ 指令内的所有字符都会被注释

音符编写：

格式：nal (占位字符详见下方解释)

例子：c+4.

n =

c d e f g a b

音符组；使用该数组来编写音符

r

休止符

a =

+

升高半音指令

-

降低半音指令

| =

1 2 4 8 12 16 24 32

x 分音组（时值）；使用该数组来编写每个音符的分音（时值）

□

付点符号；用于延长当前音符或休止符时值的一半

%#

= %

时值（Ticks）；直接使用 Ticks 来设置音符时值

例 1： c%48 = c4

n#

= 0 ~ 95

直接使用数字指定音高；n0 = o0d+，n95 = o8d。使用该方式编写音符时如需指定时值，请用□符号分隔后填入对应时值（该指令适合用于 PCM 通道）

&

绑定 macro；用于连接前后两个音符的 macro

例 1: c4&c4 = c2

例 2: c4&>c4

-

滑音；用于音符向另一个音符的滑动效果

例 1: c4_f = 音符 c 以四分音的速度滑向音符 f

q#

= 0 ~ 8

设置断音；设定音符断音长度，范围为 q0 ~ q8，默认值为 q8

@q#

= %

设置断音；以 Ticks 为单位设定音符断音长度

k#

= %

延迟音符；音符将会延迟 # 个 Ticks 播放

w#

= 0 ~ 31

设置 OPM 噪声频率；范围为 w0 ~ w31

y#1,#2

#1 = register

#2 = value

对 OPM register #1 写入数值 #2

W

等待同步；停止播放该通道指令，只有等到其他通道发送同步指令(S)后才会在该位置继续播放

S#

= A~H (FM 通道) 或 P (PCM 通道) / 0~8

发送同步；当播放到 S 指令时，继续播放指定的通道指令

LFO 相关指令

MP#1,#2,#3

#1= LFO 波形 (0=锯齿波, 1=方波, 2=三角波)

#2= 设置 #2 Ticks 的 1/4 LFO 循环

#3= 设置 LFO 最大振幅

设置音高 LFO

MPON

开启音高 LFO；当设置 MP 指令后会自动开启

MPOF

关闭音高 LFO

MA#1,#2,#3

#1= LFO 波形 (0=锯齿波, 1=方波, 2=三角波)

#2= 设置 #2 Ticks 的 1/4 LFO 循环

#3= 设置 LFO 最大振幅

设置振幅 LFO

MAON

开启振幅 LFO；当设置 MA 指令后会自动开启

MAOF

关闭振幅 LFO

MD#

= 0 ~ 255

延迟 LFO；延迟 LFO 在一个音符开始的时间，单位为 Ticks，范围为 MD0 ~ MD255

当 MD 为 0 时，LFO 会忽略对通道进行持续 LFO；当 MD 大于 1 时，LFO 会因音符重新触发 LFO

该指令对 OPM LFO 指令无效

MH#1,#2,#3,#4,#5,#6,#7

#1= LFO 波形 (0=锯齿波, 1=方波, 2=三角波, 3=噪波)

#2= LFRQ

#3= PMD

#4= AMD

#5= PMS

#6= AMS

#7= key sync (0 = no sync, 1 = sync)

设置 OPM LFO

MHON

开启 OPM LFO；当设置 MH 指令后会自动开启

MHOFF

关闭 OPM LFO

NOTE 编译器

NOTE 编译教程

NOTE 编译器使用起来十分简单

在终端输入以下内容：

```
run68 NOTE [你的 mml 或 mus 文件.mml / mus]
```

之后在 NOTE 目录下将会生成一份 mdx 文件，没啦。

命令行开关

除此之外，NOTE 还提供了更多命令行开关编译选项，通过在 NOTE 指令后添加 -# 就可使用

下面是命令行开关 -# 功能的讲解：

- **-m<buffer size>**
指定输出 mdx 文件的缓冲区大小，单位为千字节 (Kbytes)，默认值为 64 千字节。
- **-x**
反转八度上下命令 <、> 的方向，效果等同于指定 #octave-rev。
- **-p**
指定 ex-pcm 模式，效果等同于指定 #ex-pcm。
- **-r**
当发生错误时，删除 mdx 文件，效果等同于指定 #remove。
- **-i<channel>**
声道掩码，用于指定不进行转换的声音。具体可指定 A~H、P~W，小写字母也可。
- **-b**
若发生错误，在转换结束后发出提示音，效果等同于指定 #beep (真机才有效)。
- **-l**
输出 pcm 使用情况文件 (pcmuse.map)。
- **-o**
对旧版 pcmuse.map 与新版 pcmuse.map 进行逻辑或运算 (需指定 -l 开关)。
- **-c [n]**

压缩音长，对可连接的休止符（音符）进行音长压缩。默认仅压缩休止符；若指定为 -cn，则音符也会被纳入压缩范围。但需注意，对音符进行压缩时，可能会改变量化程度；且在某些条件下，压缩可能无法正常生效。其中，连音、波形效果期间、带有 ‘_’ 的音符、滑音处理期间等情况，不进行压缩。

- **-z [dvqpt012]**
优化功能；对 D、v、@v、q、@q、p、@、MD、MP、MA 命令进行处理，删除或移动判定为不必要的命令。若不指定参数，上述所有命令均会被纳入优化范围；若指定 d（对应 D）、v（对应 v、@v）、q（对应 q、@q）、p、t（对应 @）、0（对应 MD）、1（对应 MP）、2（对应 MA）作为参数，则可指定具体需优化的命令。
- **-t [name]**
保存音色数据（二进制格式），在转换完成时，将音色的注册状态及音色数据以指定文件名保存。若省略文件名，将在当前目录下生成名为 tone.bin 的文件。
- **-w [name]**
保存波形数据（二进制格式），在转换完成时，将波形的注册状态及波形数据以指定文件名保存。若省略文件名，将在当前目录下生成名为 wave.bin 的文件。
- **-v [0|1]**
详细模式，可在一定程度上识别转换进度及错误位置。若不指定参数或指定 0，会将错误发生位置之后的内容更改颜色显示；若指定 1，会在错误位置的正前方显示 '→'。
- **-l**
在首次发生错误时，中止转换操作。
- **-e**
忽略错误并输出文件，即使发生转换错误，仍会通过 -t、-w、-l 开关执行文件输出操作。

对于无参数的开关，可采用如 -xpr 这样的合并写法

NOTE MML 指令

编写规则

1. 所有指令遵循严格的大小写写法, 请注意
2. `#` 符号代表数值
3. Ticks 是序列中使用的最小计算单位 (此文档将该单位设为 %)
十六分音为%12; 八分音为%24; 四分音为%48; 二分音为%96; 一分音为%192

全局指令

此类指令忽略大小写写法

`#title "曲名"`

设置 MDX 文件标题

`#pcmfile "文件名.mdx"`

设置指定的 PCM 文件

`#include "文件名"`

文件插入

将指定的外部文件内容插入到该指令所在的位置; 文件嵌套 (包含文件中再包含其他文件) 最多支持 16 层。

`#play "<命令> [参数..]"`

设置转换完成后执行的命令

例如设置为`#play "mxp sample"`, 转换完成后会自动演奏对应的乐曲。

以下情况该命令不会执行: 转换过程中发生错误、执行时按住 Shift 键。

命令会沿系统路径搜索, 但直接填写完整路径 (全路径) 可节省搜索时间, 缩短命令启动耗时。

`#ex-pcm`

启用扩展 PCM 模式, 激活 Q ~ W 通道

`#tps <数值>`

全局移调

参数范围: -24 ~ 24 (单位: 半音, 负数降低音高, 正数升高音高)

默认不作用于 PCM 声部

`#tps-all`

使 `#tps` 对所有声部生效

#detune <数值>

D 命令偏移 (-32768 ~ 32767), 仅影响 D 命令

仅当 MDX 文件中使用了 D 命令 (用于微调音高) 时, 该偏移量才会被加到 D 命令的原
始数值上; 不使用 D 命令则无效果

#octave-rev

将<>命令反转

原本用于调整八度的< (降低八度) 和> (升高八度) 命令功能互换
(即<变为升高、>变为降低)

#flat "音高"

指定音程始终附加降音

可指定的音程为 c、d、e、f、g、a、b 中的任意一种, 支持同时指定多个
(如#flat "c f"表示 c 和 f 音始终降半音)

#sharp "音高"

指定音程始终附加升音

可指定的音程为 c、d、e、f、g、a、b 中的任意一种, 支持同时指定多个
(如#sharp "c f"表示 c 和 f 音始终升半音)

#natural "音高"

取消指定音程的 #flat/#sharp 设置

可指定的音程为 c、d、e、f、g、a、b 中的任意一种, 支持同时指定多个
若省略“音程”参数 (如直接写#natural), 则解除所有音程的 #flat/#sharp 设置

#normal "音高"

同 #natural

#nlist

标记不进行转换的区域的起点

需与#list 配合使用, 两者之间的内容为“非转换区域”。

#list

标记需要进行转换的区域的起点

与#nlist 配对使用, #nlist ~ #list 包围的范围不会被转换 (即跳过该区间的内容)

#overwrite

优先使用后续的同编号音色定义 (覆盖前期定义)

即使之前已定义过相同编号的音色, 后续出现的同编号音色定义会忽略前期定义并覆盖;
该指令从自身所在行之后开始生效

#toneofs <数值>

为音色编号添加偏移量

该指令设置的数值会被加到“音色定义时的编号”和“音色调用时的编号”上
(即整体偏移音色索引)

#noretur

取消 K 指令（残响切断）后的音色重新定义

默认情况下，执行 K 指令（用于切断残响）后，系统会重新定义当前使用的音色；启用该指令后，K 指令执行后将不再重新定义音色。

#wavemem

分配波形处理所需的内存

分配约 128KB 的内存（可存储 128 个波形数据）。

仅当内存分配成功后，后续才能使用波形内存及相关指令（如波形定义、调用）。

#remove

转换错误时删除 MDX 文件

若转换过程中发生错误，不仅不会创建新的 MDX 文件，还会删除之前已生成的同名 MDX 文件（默认错误时仅不创建新文件，保留旧文件）

#beep

转换结束后若有错误，播放提示音（蜂鸣音，只在实机有用）

#reste [0|1]

设置波形效果是否作用于音符的休止部分

0= 因量化（Quantize）处理产生的音符休止部分，也会施加波形效果

1= 除休止部分外，休符本身也会施加波形效果

补充：同步模式下，若需在休符时保持波形相位，可使用&连接音符，或设置#cont 2

#nreste [0|1]

限制波形效果仅作用于音符的实际发声部分

0= 波形效果仅作用于音符的实际发声部分，且发声部分结束后，不恢复波形效果的初始状态

1= 波形效果仅作用于实际发声部分，且发声部分结束时，会将波形效果的状态恢复到施加前的初始状态

#coder

自动用休符补充通道标识符对应的音符 / 休符缺失部分

当“通道标识符的数量”多于“音符或休符的数量”时，系统会自动在缺失的位置填充休符，确保每个通道都有对应内容

#ncoder

不补充通道标识符对应的音符 / 休符缺失部分

当“通道标识符的数量”多于“音符或休符的数量”时，系统不会填充休符，后续的通道将无任何内容（即空缺）

#glide [0|1]

设置滑音（Glide）效果的处理范围

0= & 即使是&（音符连接符）后的音符，也会施加滑音处理

1= & 仅对&连接的多个音符中的第一个音符施加滑音处理，后续音符不处理

#load-tone "文件名"

加载指定的音色数据文件（该文件需通过-t命令行开关生成）

加载文件中包含的音色数据及音色注册状态

注意：即使之前已手动定义过音色，加载文件后会被覆盖并初始化（原音色设置失效）

#save-tone ["文件名"]

将当前的音色数据以二进制格式保存到文件

功能与-t命令行开关一致（用于导出音色数据）

命令行中通过-t指定的保存设置优先级高于该指令

若省略“文件名”参数，将使用默认文件名（需参考具体工具的默认规则）

#load-wave "文件名"

加载指定的波形数据文件（该文件需通过-w命令行开关生成）

加载文件中包含的波形数据及波形注册状态

注意：即使之前已手动定义过波形，加载文件后会被覆盖并初始化（原波形设置失效）

考虑到波形可重新定义的特性，系统会按“每个通道”分别执行加载操作

#save-wave ["文件名"]

将当前的波形数据以二进制格式保存到文件

功能与-w命令行开关一致（用于导出波形数据）

优先级：命令行中通过-w指定的保存设置优先级高于该指令

若省略“文件名”参数，将使用默认文件名（需参考具体工具的默认规则）

#cont [0|1|2]

设置&连接符对波形相位的影响（是否持续相位），默认为 0

0= &连接的音符 / 休符视为一个整体，音符中途不初始化 / 不更新波形相位（相位持续）

1= 忽略&的连接作用，对每个音符 / 休符单独初始化 / 更新波形相位（相位不持续）

2= 在 1 的基础上，额外取消“同步模式下休符的相位初始化”（休符时不重置相位）

#wcmd [0|1|2]

设置波形效果（AP、DT、VM）生效时，是否禁止写入相关通用命令，默认为 0

0= 正常写入相关通用命令（包括 p、D、v、@v、(、)、V，用于音量、音高微调等）

1= 不写入相关通用命令，但会保存命令的设置值，且该值仍会影响波形效果

2= 完全忽略相关通用命令（不写入也不保存设置值，对波形效果无影响）

#pcmlist [0|1]

生成 PCM 文件的使用情况报告（输出为 pcmuse.map 文件）

功能与命令行开关中的 -l（基础使用报告）、-l -o（详细使用报告）一致

0= 仅输出基础使用报告（与-l 效果相同）

1= 输出详细使用报告（与-l -o 效果相同）

#compress [0|1]

设置是否压缩音符的音长

功能与命令行开关中的-c[n]（音长压缩）一致

命令行中通过-c[n]指定的压缩设置优先级高于该指令

0= 启用基础音长压缩（与-c 效果相同）。

1= 启用进阶音长压缩（与-c[n]效果相同，n 为压缩等级，具体参考工具定义）。

生效范围：该指令从自身所在行之后开始生效，初始状态为“不压缩”。

#opt ["dvqpt012*"]

设置 MDX 文件的优化选项

无参数：关闭所有优化功能

参数为*：启用全部优化选项（等同于指定 dvqpt012）

可选参数说明（单个或组合指定，如#opt "dv"）：

d：优化音高微调（D 命令）

v：优化音量控制（v 命令）

q：优化量化处理

p：优化 PCM 相关指令

t: 优化音色调用

0/1/2: 优化特定通道或效果逻辑（具体参考工具文档）

#ver [0|1]

设置是否启用 Barbados 兼容模式

功能与命令行开关-v（Barbados 模式）一致。命令行中通过-v 指定的模式优先级高于该指令

生效范围：该指令从自身所在行之后开始生效，初始状态为“不启用 Barbados 模式”

与-v 开关的区别：启用该指令后，不会显示当前正在处理的通道

（而-v 开关通常会显示通道信息）

通用指令

A~H

设置 FM 通道标识符；OPM 拥有 8 个通道，分别对应字母 A~H

P~W

设置 PCM 通道标识符；OKIM6258 可以被分为 8 个通道，分别对应字母 P~W

在编写音符或是任何效果前，必须先以你想要编写的通道标识符为开头

t#

=19 ~ 4882

设置乐曲速度；以每秒的四分音符数量为单位

@t#

=0 ~ 255

设置乐曲速度；直接设置 OPM register B 的数值

@#

设置乐器；FM 通道和 PCM 通道可以分别设置，PCM 通道默认为@0，FM 通道不定

@w#

设置波形；

@k#

音色编号映射定义；

@@@#

音色宏定义；

= 编号 (0 ~ 255)

F#

= 0 ~ 4

设置 PCM 频率；默认值为 4，具体参考下列

0 = 3.9kHz

1 = 5.2kHz

2 = 7.8kHz

3 = 10.4kHz

4 = 15.6kHz

= 5 ~ 6

设置 PCM 频率 (x68k 电脑需要安装 PCM8 v0.46 及以上版本驱动)

5= 16bit pcm / 15.6Khz

6= 8bit pcm / 15.6kHz

= 5 ~ 31

设置 PCM 频率 (x68k 电脑需要安装 pcm8++ 驱动)

L

设置全局循环开始点；当前通道所有指令播放完后将会重新播放在 L 符号之后的所有指令

C

伪循环开始点；当前通道所有在 C 指令之后的指令播放完后将会标记一次循环，但实际不循环，不可与 L 同时使用

o#

设置八度音程；范围为 o0 – o8， 默认值为 o4

<

降低一个八度音程

>

升高一个八度音程

|#

设置音符默认长度， 默认值为 |4

v#

= FM (0 ~ 15)

设置音量；范围为 v0 ~ v15， 默认为 v8

@v#

= 0 ~ 127

以更高精度设定设置音量；范围为 @v0 ~ @v127

V#

相对音量调整；类似 ()，但在当前音量基础上叠加数值后写入 v/@v，不考虑 () 的变化，音量到最高时自动限幅

VO#

= -127 ~ 127

音量命令偏移，为 v、@v、x、@x 设置偏移量，范围 -127 ~ 127， 默认 VO0

(#

音量变化；降低 # 格当前通道音量，#可以被忽略 (1 格)

)#

音量变化；提高 # 格当前通道音量，#可以被忽略 (1 格)

x#

#= 0 ~ 15

设定紧随其后的 1 个音符的音量；从第 2 个音符开始，恢复为原来的音量

@x#

#= 0 ~ 127

以更高精度设定紧随其后的 1 个音符的音量；从第 2 个音符开始，恢复为原来的音量

p#

= 0 ~ 3

设置声像； p0=不输出， p1=左声像， p2=右声像， p3=左右声像， 默认值为 p3

D#

= -32767 ~ 32767

设置失谐；以 1/64 的半音为单位

_D#

= -6144 ~ 6144

滑音失谐；用于音符后。以之前音符的音长为时长，按指定数值线性改变音高。数值单位为 1/64 半音

TR#

= -48 ~ 48

设置移调；对当前通道移调，对 PCM 通道同样有效且默认叠加 #tps 的修正效果，
默认值为 TR0

\$FLAT {音符}

指定降音；为 cdefgab 中指定的音符始终添加降号

\$SHARP {音符}

指定升音；为 cdefgab 中指定的音符始终添加升号

\$NORMAL {音符}

解除降音 / 升音设置

\$NATURAL {音符}

解除指定音符的 FLAT/\$SHARP 设置，省略 {音符} 时解除所有音符

[]#

= 2 ~ 255

设置循环；将会循环 [] 符号内的指令 # 次

/

跳过循环；在 [] 指令内使用，在最后一次循环时，跳过 / 符号后的指令

!

忽略所有 ! 符号后的指令

;

行注释；在同一行的 ; 指令后的所有字符都会被注释

/* */

块注释；在 /* */ 指令内的所有字符都会被注释

?

范围跳过；忽略 ? 包围范围内的实体命令输出，非实体命令 (o、<、>、! 等) 状态保留，但范围结束后需重新设置实体命令。该范围内语法检查放宽，且 L、C 等命令仍会被转换

音符编写：

格式：nal (占位字符详见下方解释)

例子：c+4.

n =

c d e f g a b

音符组；使用该数组来编写音符

r

休止符

a =

+

升高半音指令

-

降低半音指令

| =

1 2 4 8 12 16 24 32

x 分音组（时值）；使用该数组来编写每个音符的分音（时值）

.

付点符号；用于延长当前音符或休止符时值的一半

%#

= %

时值（Ticks）；直接使用 Ticks 来设置音符时值，NOTE 可以使用 # 命令代替

例 1： c%48 = c#48 = c4

n#

= 0 ~ 95

直接使用数字指定音高；n0 = o0d+，n95 = o8d。使用该方式编写音符时如需指定时值，请用□符号分隔后填入对应时值（该指令适合用于 PCM 通道）

&

绑定 macro；用于连接前后两个音符的 macro

若前后逻辑正确，可写在行首，但在波形效果中行首的 & 将被忽略

例 1: c4&c4 = c2

例 2: c4&>c4

_

滑音；用于音符向另一个音符的滑动效果

音高前仅可加 o、<、> 符号

若指定音长短于直前音符音长，采用直前音长；若更长，则过渡完成后维持最终音高至剩余时长

波形效果中使用时，波形效果将暂停。

例 1: c4_f = 音符 c 以四分音的速度滑向音符 f

{}

连音；将指定音长按 {} 内音符数量等分分配，{} 内最多包含 32 个音符/休符，可指定超过 %256 的音长。

{} 内不支持循环相关命令、波形效果、滑音、滑音后的音长指定。

q#

= 0 ~ 8

设置断音；设定音符断音长度，范围为 q0 ~ q8，默认值为 q8

@q#

= 0 ~ 192

设置断音；以 Ticks 为单位设定音符断音长度

Q#

波形专用量化，仅波形中有效

0：恢复由 q/@q 控制

1 ~ 256：以全音长的 1/256 为单位量化

-256 ~ -1：绝对值为实际发音步数，全音长≤绝对值时以全音长为发音时长

k#

= 0 ~ 255

延迟音符；音符将会延迟 # 个 Ticks 播放

w#

= 0 ~ 31

设置 OPM 噪声频率；范围为 w0 ~ w31，不设置时默认关闭

y#1,#2

#1 = register

#2 = value

对 OPM register #1 写入数值 #2

W

等待同步；停止播放该通道指令，只有等到其他通道发送同步指令(S)后才会在该位置继续播放

S#

= A~H (FM 通道) 或 P~W (PCM 通道) / 0~15

发送同步；当播放到 S 指令时，继续播放指定的通道指令

LFO 相关指令

MP#1,#2,#3

#1= LFO 波形 (0=锯齿波, 1=方波, 2=三角波)

#2= 设置 #2 Ticks 的 1/4 LFO 循环

#3= 设置 LFO 最大振幅

设置音高 LFO

MPON

开启音高 LFO; 当设置 MP 指令后会自动开启

MPOF

关闭音高 LFO

MA#1,#2,#3

#1= LFO 波形 (0=锯齿波, 1=方波, 2=三角波)

#2= 设置 #2 Ticks 的 1/4 LFO 循环

#3= 设置 LFO 最大振幅

设置振幅 LFO

MAON

开启振幅 LFO; 当设置 MA 指令后会自动开启

MAOF

关闭振幅 LFO

MD#

= 0 ~ 255

延迟 LFO; 延迟 LFO 在一个音符开始的时间, 单位为 Ticks, 范围为 MD0 ~ MD255

当 MD 为 0 时, LFO 会忽略对通道进行持续 LFO; 当 MD 大于 1 时, LFO 会因音符
重新触发 LFO

该指令对 OPM LFO 指令无效