

Introduction

Introduction

Hello there! This is the first English-language documentation for the KOLIN2 Advanced Sound BIOS by Akikaz for PC-9801. What follows is a brief primer, including some differences between the more popular PMD and this relatively little-known gem. After that, we'll be wrist deep in KOLIN2's MML.

I'd recommend checking out my PMD guide for a primer on OPN series and programming it in the context of other chips, and possibly also for some of the basic concepts of this MML, as they share some similarities that are a bit critical.

Coming to KOLIN2 from PMD

Differences From PMD include:

- Ability to escape loops early in a sequence with the / character. Ex: [e8r8e8r8 r2 /r1]3 will escape the loop on the 3rd iteration before the whole rest (r1)
- A different tempo system, 0-255. It does not seem to specify BPM but some sort of clock multiplier instead
- All channels available in the current driver mode (86 or 26) must be declared and given instructions. The compiler will complain about unused channels
- KOLIN2 includes its own PCM object linker that compiles a .PO8 file for use with 86 mode
- FM patch sizes do not include an AMS Mask flag like with PMD—patches used with PMD compositions must have this parameter removed before a file will compile
- Board mode must be specified in the MML. If 86 is chosen, the song may not play on 26 hardware at all even if only the YM2203 channels are used

Using KOLIN2

To initialize KOLIN2 in DOS V on PC-9801, navigate to the directory containing KOLIN2.COM and type KOLIN2 to load the driver.

To compile an MML source file to a playable driver instruction file for KOLIN2, type

```
klc [option] [directory of MML file]
```

To play a KOLIN2 driver instruction file, type in

```
klp [directory of compiled file]
```

Make sure to include the extension, as KOLIN2 compiles .MO8 and .MO2 files, depending on the driver mode specified in your source file.

KOLIN2 Basic Operation

KOLIN2 Preprocessor Directives

`#mode-86` specifies that the file is meant to be used on PC-9801-86 hardware (OPNA, .mo8 extension)

`#mode-26` specifies that the file is meant to be used on PC-9801-26K hardware (OPN, .mo2 extension)

Note: Mode 26 files will play on PC-9801-86 and more, but not the other way round, as the PC-9801-26K and Speak Board and other hardware lacks the required components to play 86 music.

`#octave-rev` reverses the direction of the octave incrementation operator notation.

`#volume-rev` reverses the direction of the volume incrementation operator notation.

`#detune-rev` reverses the direction of the detune incrementation operator notation.

`#include "Filename"` sets additional source files to include.

`#title "Title"` sets the Title metadata for the compiled music.

`#composer "Noyemi"` sets the Composer metadata for the compiled music.

`copyright "(c) 1979 Sunrise"` sets the Copyright metadata for the compiled music.

`#pcmfile "Filename.po8"` sets the PCM block file to be used with the compiled music and triggers PCM from that table.

KOLIN2 Basic Channel Elements

The available channels for the user will depend on the mode the source file is written in!

FM Channels:

- ABC (mode 26 or 86)
- DEF (mode 86 only)

SSG Channels:

- GHI (mode 26 or 86)

Rhythm/PCM (mode 86 only):

- R (RSS ROM)
- S (PCM)

As with PMD and other FM program APIs for some Japanese PCs, you program FM instrument patches by specifying an instrument offset and writing a parameter array for each operator:

```
@[1-255] = {  
    AR, DR, SR, RR, SL, TL, KS, ML, DT,           ;Operator 1  
    AR, DR, SR, RR, SL, TL, KS, ML, DT,           ;Operator 2  
    AR, DR, SR, RR, SL, TL, KS, ML, DT,           ;Operator 3
```

AR, DR, SR, RR, SL, TL, KS, ML, DT,
ALG, FB }

;Operator 4

These have the same limitations/values as PMD, with the exception of there being no AMS mask flag.

KOLIN2 Song Sequence Commands

Comments: ;, /*

Any characters on the line after the comment command are ignored.

Delimiter: |

A useful separator that is not read by the compiler, used to mark parts in a sequence for the programmer's benefit.

End Mark: !

A command that halts the playback on the channel when it is reached.

Loop Macro: L

Channel sequence information following this macro is looped infinitely.

Iterated Loop Macro/Escape: [... / ..]n

Information inside of [..] will be looped n times, and optionally escaped on the last iteration by / to cut the loop short.

Octave Macro: o[0-8]

Called before any note to shift the octave to the range indicated after the macro. > and < increment or decrement the octave as indicated by the presence of the #octave-rev preprocessor command if available. Default behaviour is the reverse of PMD (< increments and > decrements)

Length Macros: 1, 2, 4, 8, 16, 32, 64, 96, 192 or ., .., ... or %1-255

Putting an integer after the note will snap it to that length (whole, half, 4th, 8th, etc.) or it can be defined for an entire sequence by the length macro **lowercase l**. Dot macros extend the note by half of the previous extension (. extends an 8th note by 1/16, and a dot following will extend it by an additional 32nd).

Finally, the % macro uses the actual sound processor sequence clock for note length for even more granular control.

Rest Macro: r

This macro adds in a rest/keyoff. The length can be defined with the length macro in the same way you would with any other note.

Extension Macro: ^

This macro command extends the length of the note by an entire default length, unlike the dot operator. If the clock time exceeds 255, this may not work as expected.

Tie/Hammer-on/Pull-off Macro: &

The next note is played without retriggering the amplitude envelope, much like a guitar hammer-on.

Gate Time Quantization Macros: q[1-8] or @q[1-127]

Specifies gate length in terms of note length or absolute sound processor clock, respectively. k[0-127] will delay gate keyon time.

Key Transpose Macro: K[-127,127] and R[-127,127]

Absolute transposition of the sequence key by up to 127 semitones up or down. The R variant makes the transposition relative to the current absolute value.

Fine Detune Macros: D[-16383.16383] or {[0-16383], }[0-16383]

The D macro represents an absolute value for detune and { and } are increment/decrement macros. Each step represents 1/64 of a semitone.

Volume Control Macros: v[0-15] or @v[0-127] or ([1-127],)[1-127]

The default v macro selects an absolute volume for notes that follow, and @v controls this value to a finer degree. (and) are increment and decrement.