

NRC/SAS
Kai Havukainen

13.9.2002

Nokia SP-MIDI Creator

VST Plugin Manual

Owner: NRC/SAS
Scope: Entertainmetn Audio
Originator: Kai Havukainen
Status: Final for beta release
Document ID: TBD
Location: TBD

Change History

Issue	Date	Handled by	Comments
1.0 beta	13.9.2002	Kai Havukainen	Beta release manual

Approved by

NRC/SAS
Kai Havukainen

13.9.2002

1.	Overview	3
1.1	Purpose	3
1.2	Requirements.....	3
2.	Installation	4
3.	SP-MIDI in a nutshell.....	5
4.	The plugin – Basic functionalities.....	7
4.1	Selecting target device.....	7
4.2	Using target device's polyphony limit.....	7
4.3	Channel polyphony meters.....	7
4.4	Maximum polyphony meters	8
4.5	Saving MIP messages	8
5.	Managing MIP messages	9
5.1	Inserting.....	9
5.2	Editing.....	9
5.2.1	Overview.....	9
5.2.2	Changing the channel priority order	10
5.2.3	Changing the MIP values	10
5.2.4	Real MIP values.....	10
5.2.5	Muting the channels.....	11
5.2.6	Changing MIP message position.....	11
5.2.7	Changing the view.....	11
5.3	Playback with multiple MIP messages.....	11
6.	Preferred way of creating SP-MIDI	12
7.	Known problems.....	13

NRC/SAS
Kai Havukainen

13.9.2002

1. OVERVIEW

1.1 Purpose

Nokia SP-MIDI Creator is a plugin for creating Scalable Polyphony MIDI (SP-MIDI) using professional MIDI sequencer programs that support Virtual Studio Technology instruments (VSTi). VST is a trademark of Steinberg Soft- und Hardware GmbH.



The plugin can not be used in real-time i.e. for playing it live, due to the considerable latency. The purpose is to simulate the playback of target devices and to insert MIP messages to convert standard MIDI files into SP-MIDI format.

1.2 Requirements

Operating system: Windows 98, 2000, XP

Hardware: 256MB RAM, CPU 500MHz

MIDI Sequencer: Cubase VST 5.0 or later

NRC/SAS
Kai Havukainen

13.9.2002

2. INSTALLATION

For installation, please follow these instructions carefully.

1. Copy the files named MSVCRTD.DLL and MSVCP60D.DLL to your System32 directory (c:\winnt\system32 or c:\windows\system32 or similar) if you don't have them there yet.
2. Create a new folder, e.g. "spmidi" for the plugin under the "Vstplugins" folder of your sequencer program. For Cubase the plugin folder could be "c:\steinberg\cubase\vstplugins\spmidi".
3. Copy the plugin file "SPMidiCreator.dll" file into the folder you just created.
4. Copy to the same folder also all the device description files, named *.spm, containing all necessary information about target terminals.

Now you can launch your sequence program, e.g. Cubase. There are couple of things you have to do before using the plugin.

1. Change the sample rate of the sequencer to 32kHz. In Cubase VST/32 version 5.0 this can be done from menu "Options -> Audio -> System -> Sample rate". Without this the playback would crackle.
2. Set the screen offset to zero. In Cubase VST/32 version 5.0 this can be done from menu "Options -> Synchronization" and setting "Bar display" to zero. This means that the first bar number at the display is number 1, that is same than assumed by the plugin.

Then you can open the VST-instrument view of the sequencer. In Cubase VST/32 version 5.0 it can be found from menu "Panels -> VST instruments". You should be able to locate the "SPMidiCreator" plugin from the drop-down menu. Select it, switch it on (red button) and press "Edit" to see the user interface.

Now when the plugin is active, you can lead MIDI messages from sequencer tracks to the plugin. Figure 1 shows how MIDI data is routed to VST instruments in Cubase. MIDI messages of every channel can be routed this way to computer's MIDI interface, sound card or plugin instruments.

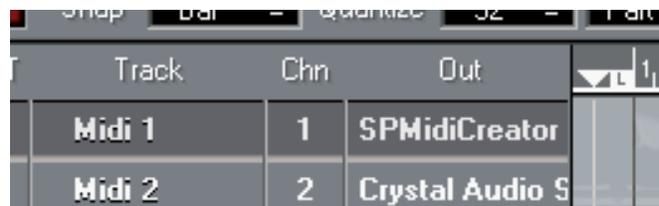


Figure 1: Routing MIDI messages to plugin

3. SP-MIDI IN A NUTSHELL

SP-MIDI was developed to give composers chance to decide how synthesizers will play MIDI songs that require more polyphony than the synthesizer is capable to offer.

Generally speaking, SP-MIDI is much like General MIDI standards, defining instrument mapping to the MIDI program numbers, and requirements for MIDI messages to be supported. In addition, SP-MIDI defines a System Exclusive MIDI message that is used to define the cumulative polyphony levels of SP-MIDI songs. This message is called Maximum Instantaneous Polyphony (MIP) message, and SP-MIDI songs must always contain at least one MIP message, being the first message of the first MIDI track.

But how does the MIP message effect the playback? There are two kind of information inside MIP messages:

- MIDI channel priority order, and
- cumulative polyphony levels (MIP values) of the MIDI channels.

By the channel priority order, composer defines how important each channel is compared to the others. There can be all the 16 MIDI channels in arbitrary order, or only some of them. If a MIDI channel contains notes but the channel number is not in the MIP message, SP-MIDI synthesizers do not play the channel.

MIP values are also set by composer. They are integers that always make pairs with channel numbers of the priority list. Using MIP values composer tells synthesizers how much polyphony is needed to be able to play channels up to that channel priority level. When synthesizer notices that playing a channel requires more polyphony than it is capable of offer, it does not play any notes on that channel. This way randomized playback and cutting notes can be avoided.

Then, an illustrative example how a polyphony level 4 synthesizer analyses a MIP message. Let the MIP message contain information as follows:

Priority	MIDI channel	MIP value
Highest	1	1
	10	3
	2	4
	4	6
Lowest	3	8

When the synthesizer receives a SP-MIDI file with this MIP information, it “thinks” like this:

- “So, I’m capable of playing up to polyphony level 4”
- “Channel 1 has the highest priority and MIP value 1. I’m capable of playing it.”
- “If I want to play also channel 10, I should be able to play polyphony 3 music. I am, so I’ll play also channel 10.”
- “Also channel 2 seems to fit into my maximum polyphony, since it has MIP value 4, and that is what I’m capable of playing. I’ll play also it.”

NRC/SAS
Kai Havukainen

13.9.2002

- “Channel 4 has a MIP value over my maximum polyphony level. I do not play it nor any other MIDI channels with lower priority.”

Because the composer can set the MIP values as (s)he likes, they do not necessarily correspond exactly to the real cumulative polyphonies. For example, when 4 polyphony device plays the previous song, same number of channels would be played also with next MIP messages:

Priority	MIDI channel	MIP value
Highest	1	1
	10	1
	2	1
	4	6
Lowest	3	8

Priority	MIDI channel	MIP value
Highest	1	4
	2	4
	10	4
	4	6
Lowest	3	8

If the composer wants also the channel 4 to be played, (s)he can either modify the song reducing overlapping of notes on channel 4 and higher priority channels, or just set the MIP value of channel 4 to be 4. The latter alternative would probably cause some note-stealing but in many cases it is not audible or too disturbing.

4. THE PLUGIN – BASIC FUNCTIONALITIES

In this section, the basic functionalities of the plugin are introduced, excluding MIP message related topics. These are discussed in more detail in Section 5. Figure 2 shows the user interface after opening and adding one MIP message at position 43.3.1.0.

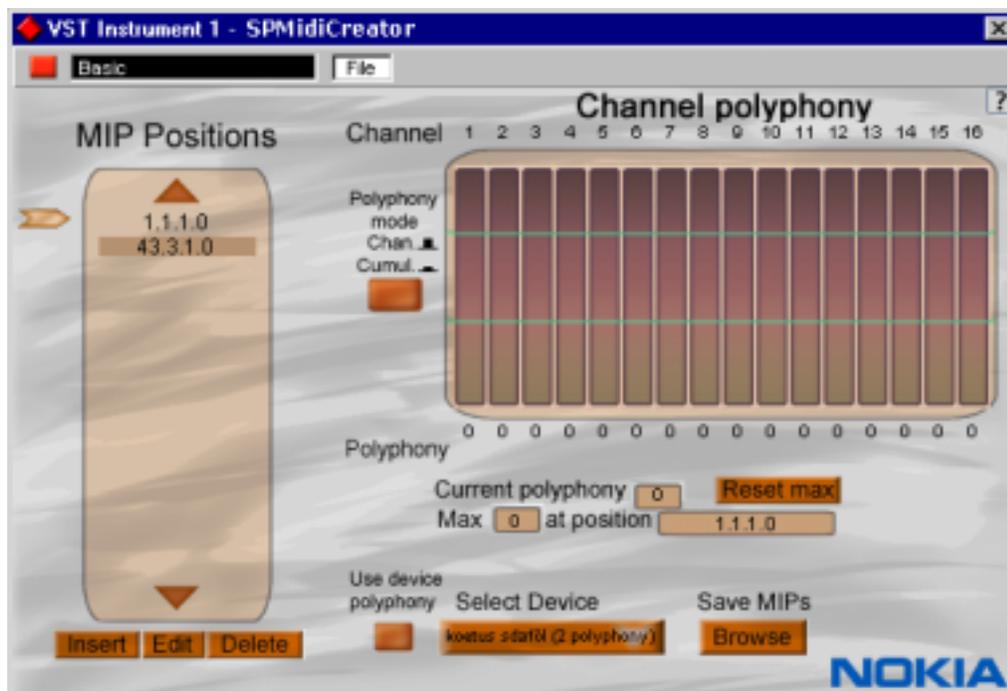


Figure 2: Plugin user interface with one additional MIP message

4.1 Selecting target device

Target device can be selected from *Select Device* popup menu. All supported target devices (all *.spm files in plugin folder) and their maximum polyphonies can be found from the menu list. When target device is selected, instruments are changed to the ones that are used in the selected device.

4.2 Using target device's polyphony limit

The button *Use device polyphony* allows user to switch on and off the polyphony limitation of the target device. This is practical when comparing the target device output to the total, unlimited playback. Also, when *Use device polyphony* button is off, the plugin keeps track of the total polyphonies of the channels (*Real MIP* values of the MIP edit view), not only played ones.

4.3 Channel polyphony meters

The biggest part of the UI is a row of LED bars, that show polyphonies of current playback. The MIDI channel numbers above the row are in the channel priority order (highest priority leftmost) of the current MIP message and the LED bar and polyphony value beneath are bound to the channel value. The channel order changes every time when a song is played to the position of a MIP that has different channel priority order than the previous one.

There are two polyphony modes of showing the polyphonies: cumulative and channel specific. The selection can be made by *Polyphony* mode button. Figure 3 and Figure 4 show polyphonies of a song at same position but with different polyphony mode.

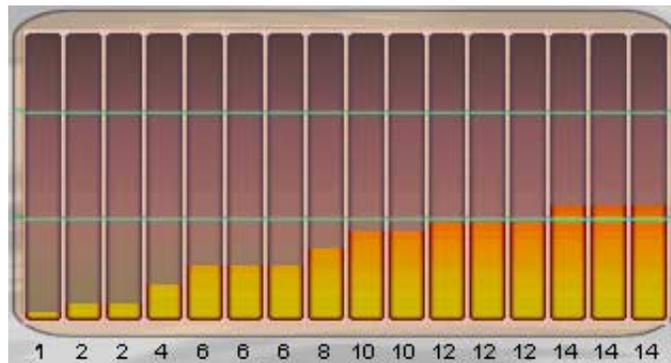


Figure 3: Cumulative polyphony mode

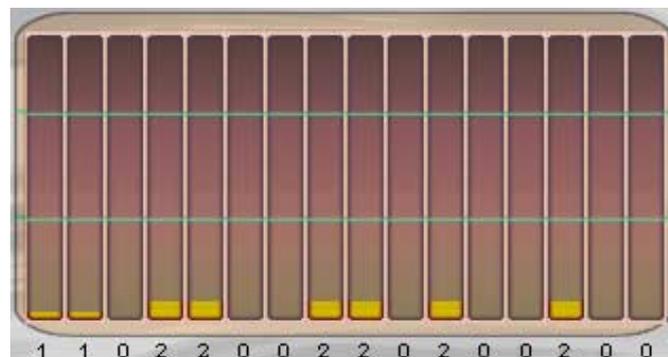


Figure 4: Channel specific polyphony mode

4.4 Maximum polyphony meters

When a song is played, the plugin continuously shows the *Current polyphony* of the song and the *Maximum polyphony* of the song. Also song position is showed for latter. These values for maximum polyphony can be reset by pressing *Reset max* button beside the meters.

4.5 Saving MIP messages

When a song and MIP messages has been modified and tested, it has come time to add the MIP messages to a MIDI file to make it a SP-MIDI file. Before this, **you must export the song as a MIDI file from your sequencer**. You can save the MIPs by pressing *Browse* button that opens a file selecting dialog. Now find the MIDI file that you want to change to SP-MIDI and press *Save*. The dialog is closed and a new SP-MIDI file is born.

5. MANAGING MIP MESSAGES

5.1 Inserting

There must always be a MIP at the very beginning of the song (at position 1.1.1.0) and thus it cannot be removed or moved. For many SP-MIDI songs this one MIP is enough but the plugin supports also multiple MIP messages that can be inserted, edited and removed.

MIP messages can be inserted at desired position either by

- moving the sequencer's cursor to the desired position and pressing the *Insert* button below the *MIP Positions* box, or
- pressing the *Insert* button below the *MIP Positions* box, double-clicking the new message position from the *MIP Positions* box (or selecting it and pressing the *Edit* button) so that the MIP editing view is opened, and writing the desired position beside the *OK* button.

The MIP messages are listed in *MIP Positions* list in the order of their occurrence.

The format of the MIP position is <Bars>.<Beats>.<1/16th notes>.<Tics> that is a commonly used format for timing in sequencer programs. Another common format is <Bars>.<Beats>.<Tics> but this is not supported by the plugin. The <Tics> position is rounded towards nearest 100.

5.2 Editing

5.2.1 Overview

MIP editing view can be opened by double-clicking one of the MIP positions, or selecting one and pressing *Edit* button. When the MIP editing view is visible, you will see

- The numbering of *priorities* 1-16
- *MIDI channel numbers* corresponding to the priorities (these values can be modified)
- *User MIP values* corresponding to the MIDI channels, that user is able to edit (these are the MIP values that are included in MIP message and they can be modified)
- *Real MIP values* of the part of the song between the current MIP position, and the next MIP (or the end of the song). These are shown as reference for the composer, so that (s)he can see the real cumulative polyphonies of the song and adjust the *User MIP* values similarly or changing them slightly.
- *Mute* buttons for every channel

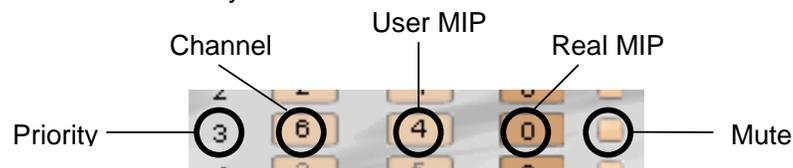


Figure 5: Priorities, channel numbers, MIPs and mute buttons

What actually is written to the SP-MIDI file are *MIDI channel numbers* in priority order and corresponding *User MIP values*. These tell the SP-MIDI synthesizer how many of the highest priority channels it is capable of playing without exceeding its maximum polyphony level. Only those MIDI channels that have smaller or same *User MIP* value compared to the synth's maximum polyphony are played.

This is also shown by the plugin UI: the *MIDI channel numbers* and *User MIP values* that fall outside the maximum polyphony of the currently selected target terminal are shown with grey font color whereas the others are shown black.

The modifications at this view take effect immediately and they can't be undone.

5.2.2 Changing the channel priority order

Each of the leftmost numbers 1-16 on the screen defines a priority (number 1 being the highest) of the MIDI channel beside it. If you want to set MIDI channel 4 to have the highest priority, click the *Channel* box of priority 1 and select number 4 from the drop-down list. The rest of the channels are shifted automatically to give space for the moved channel number.

5.2.3 Changing the MIP values

Composer can define which of the channels are played by certain target devices by changing the *User MIP* values. The MIP values are always cumulative i.e. lower priority MIP is always same or bigger than MIPs at higher priorities.

If you are targeting the SP-MIDI file e.g. for 4 polyphony device, some of the highest priority channels must have *User MIP* values 4 or less to be able to hear anything. The channels and *User MIP* values that are played by the target device are shown in black font colour, whereas the channels that are not played are shown grey.



Figure 6: Channel 6 is played by polyphony level 4 devices whereas channel 3 is not.

Changing the MIP values is done by clicking the value box, which allows you to write an integer number into it. The number must be bigger than the one of the higher priority. Writing can be completed by pressing Return (from the keypad) or clicking somewhere outside the box. If the number is bigger than the *User MIP* value of the next priority's, the next priority *User MIP* is also set to the bigger value. The same comparison is done also for the next priority level as long as all the *User MIP* values are updated.

5.2.4 Real MIP values

These values show the polyphony that really is playing. You can use these values as a reference for your *User MIP* values but in many cases songs sound pretty good with smaller *User MIP* values than what the *Real MIP* values show. The reason for this is that for *Real MIP* values two simultaneous voices are always considered to have polyphony of two, even if one of them would be shut down right after the other is launched.

NRC/SAS
Kai Havukainen

13.9.2002

The *Real MIP* values are updated all the time the playback is on and they remain the highest MIP value that has passed. However, if you change the channel priority order, the values below the changes are reset, since they are not valid anymore.

5.2.5 Muting the channels

When the *Mute* button of some priority level is switched on, that channel is not played and not included in the MIP message either. By muting the channels, you can make quick checks how channels effect the *Real MIP* values and disable channels without changing their position in the priority order.

5.2.6 Changing MIP message position

You can change the MIP position by typing integer values to the boxes at the bottom of the window. The format of the MIP position is <Bars>.<Beats>.<1/16th notes>.<Tics> that is a commonly used format for timing in sequenser programs. Another common format is <Bars>.<Beats>.<Tics> but this is not supported by the plugin.

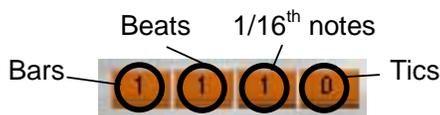


Figure 7: Position indicator

Because SP-MIDI defines that there is always a MIP message at the very beginning of songs, this position 1.1.1.0 cannot be moved. Also there can't be multiple MIP messages at a same position, so this is checked when the MIP settings are tried to be stored.

5.2.7 Changing the view

The MIP editing view can be changed back to MIP position view by pressing the *OK* button. At this point the validity check is done for the MIP position to avoid simultaneous MIP messages.

5.3 Playback with multiple MIP messages

When there are multiple MIP messages inserted and song is played, the MIP position pointer on the left side of *MIP Positions* list shows currently active MIP message (see Figure 8). When song reaches the position of some of the MIPs, the plugin sends the MIP message to it's synthesizer, just like the message would have been read from MIDI file in the case of file playback.

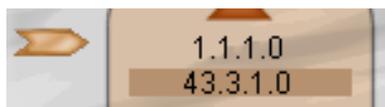


Figure 8: MIP position pointer showing that the first MIP is currently active.

6. PREFERRED WAY OF CREATING SP-MIDI

When playing a song, the MIP messages will have effect as long as the next MIP is reached. If different MIP settings are wanted to be set e.g. for verse and chorus parts of a song, the content creation requires a bit more attention than with only one MIP. However, each period between any two MIPs can be considered separately.

The recommended steps for creating SP-MIDI are as follows:

1. Compose the original MIDI file using MIDI sequencer program. Because of the latency of the plugin, it is recommended to use some other synthesizer for this step.
2. Set the plugin to be the MIDI output device for the MIDI tracks of the sequencer.
3. Insert the MIPs at desired positions as described earlier.
4. Set the channel priority order of the messages as described earlier.
5. Switch the *Use device polyphony* button off to enable full playback.
6. Play the song, or parts of it, through to get *Real MIP* values for the MIP messages.
7. Go to edit the *User MIP* values and try to adjust them to fit to your purposes. Use *Real MIP* values as reference.
8. Switch the *Use device polyphony* button on and play the song through to listen the final result.
9. If you are satisfied, export the MIDI file from the sequencer and save the MIPs to it as described earlier.

NRC/SAS
Kai Havukainen

13.9.2002

7. KNOWN PROBLEMS

- If the time signature changes during the playback, the MIP position may start showing wrong. The MIPs are located at right absolut position but the beat-based calculation looses.
- If MIP message is inserted while song is playing, it will not be located at the right position. This is due to the latency.
- If target device is selected during the playback, some of the instruments may stop playing if a Program Change is not sent for them. It is recommended to start playing from the beginning of the song in these cases.
- Muting the channels in MIP edit view is not currently working.