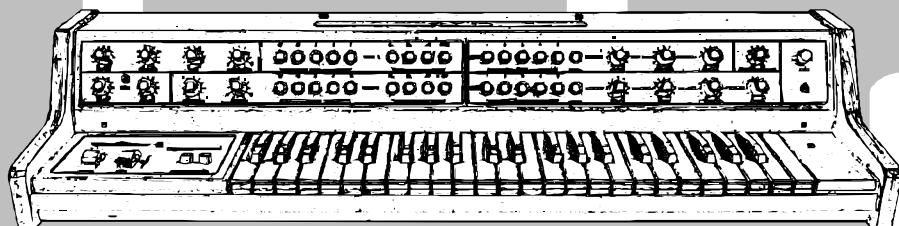


# VS-MIDI

MIDI Interface for  
Vermona Synthesizer

Model 8-434  
Version 2.0



MIDI SysEx Communication



© 2022 CHD Elektroservis  
[www.chd-el.cz](http://www.chd-el.cz)



## Contents

	Page
<b>1 SUPPORT SOFTWARE .....</b>	<b>3</b>
1.1 SYSEX MESSAGES GENERATOR .....	3
1.1.1 GLOBAL DATA SETTING .....	3
1.1.2 GLOBAL DATA REQUEST .....	3
1.1.3 PATCH DATA SETTING .....	4
1.1.4 PATCH DATA REQUEST .....	4
1.1.5 SYSTEM FUNCTIONS PROCESSING .....	4
1.1.6 OTHER CONTROLLERS .....	5
1.2 RECOMMENDED THIRD PARTY SOFTWARE .....	5
1.2.1 SETTING UP THE COMPUTER AND SOFTWARE .....	5
1.2.2 SEND THE TEXT SYSEX MESSAGE TO THE INTERFACE .....	6
1.2.3 REQUEST THE SYSEX DATA FROM THE INTERFACE .....	6
1.2.4 ALTERNATIVE SOFTWARE SOLUTIONS .....	7
1.3 CHECKSUM CALCULATOR .....	7
<b>2 SYSTEM EXCLUSIVE COMMUNICATION DESCRIPTION .....</b>	<b>8</b>
2.1 SYSEX MESSAGES SYSTEM .....	8
2.2 SYSEX MESSAGES STRUCTURE .....	9
2.2.1 "SOX" BYTE .....	9
2.2.2 "MANUFACTURER ID" BYTES .....	9
2.2.3 "DEVICE ID" BYTE .....	9
2.2.4 "MODEL ID" BYTE .....	9
2.2.5 "COMMAND" BYTE .....	9
2.2.6 "ADDRESS" BYTE .....	9
2.2.7 "DATA" BYTES .....	9
2.2.8 "CHECKSUM" BYTE .....	9
2.2.9 "EOX" BYTE .....	9
<b>3 SYSEX COMMANDS OVERVIEW .....</b>	<b>10</b>
3.1 BULK DUMP - DATA REQUEST .....	10
3.1.1 GLOBAL DATA REQUEST .....	10
3.1.2 PATCH DATA REQUEST .....	10
3.2 BULK DUMP - DATA LOAD .....	11
3.2.1 GLOBAL DATA LOAD .....	11
3.2.2 PATCH DATA LOAD .....	11
3.3 SYSTEM FUNCTIONS .....	12
3.3.1 PATCH Nr. CHANGE / REQUEST .....	12
3.3.2 PATCH SAVE .....	13
3.3.3 INTERFACE RESET .....	13
3.3.4 FIRMWARE VERSION REQUEST .....	13
3.4 SERVICE FUNCTIONS .....	13
<b>4 EXAMPLES OF SYSEX MESSAGES .....</b>	<b>14</b>
4.1 EXAMPLE 1 – CHANGE GLOBAL PARAMETERS .....	14
4.2 EXAMPLE 2 – CHANGE PATCH PARAMETERS .....	14
4.3 EXAMPLE 3 – SAVE EDIT BUFFER .....	15
4.4 EXAMPLE 4 – FACTORY RESET .....	16

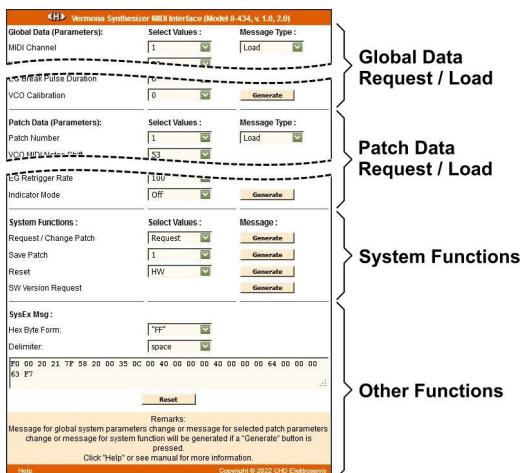
## 1 SUPPORT SOFTWARE

### 1.1 SYSEX MESSAGES GENERATOR

Special software generator for an easy creation of any necessary SysEx messages for control and programming of the VS-MIDI interface can be used. Visit our web site and download the “**VS-MIDI\_syxgen.zip**” archive. Expand the archive to a selected folder on your computer’s hard drive (i.e. “**VS-MIDI.html**” and “**VS-MIDI\_help.html**” files and “**media**” sub-folder).

To launch the SysEx messages generator, simply open the “**VS-MIDI.html**” file in your web browser (e.g. by clicking on the file icon). The generator window opens (fig. 1).

**Fig. 1 – CHD SysEx Messages Generator**

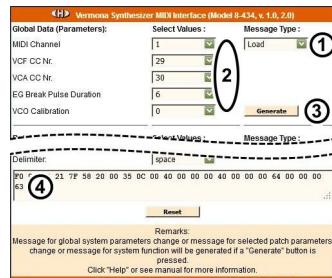


#### 1.1.1 GLOBAL DATA SETTING

To change / adjust the Global Data values (global parameters in global memory bank) see fig. 2:

1. Select “Load” in the Message Type select-box ①.
2. Select requested values of all global parameters<sup>1</sup> ②.
3. Click the “Generate” button ③.
4. The required hexadecimal SysEx message is generated as a text in text field on bottom of the window ④.
5. Copy the text in clipboard (CTRL+C) and paste (CTRL+V) to a MIDI Sysex software.
6. Send the message to the interface.
7. Switch the instrument off and then on again after a moment. Now the interface starts operation with the new global data settings.

**Fig. 2 – Global Data Load / Request**



#### 1.1.2 GLOBAL DATA REQUEST

To request actual Global Data values (global parameters from global memory bank) see fig. 2:

1. Select “Request” in the Message Type select-box ①.
2. Click the “Generate” button ③.
3. The required hexadecimal SysEx message is generated as a text in text field on bottom of the window ④.
4. Copy the text in clipboard (CTRL+C) and paste (CTRL+V) to a MIDI Sysex software.
5. Send the message to the interface.
6. The interface responds immediately – it sends “Bulk Dump - Global Data Load” type SysEx message.
7. Now you can save the received message in your computer for further use.

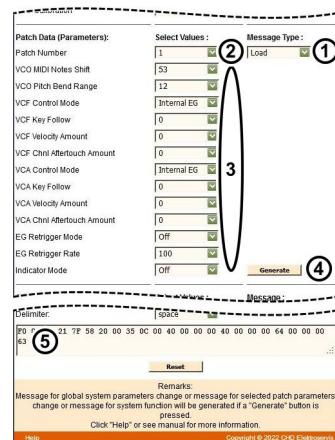
<sup>1</sup> For details about the global parameters see chapter 3.2.1.

### 1.1.3 PATCH DATA SETTING

To change / adjust the Patch Data values (patch parameters in a patch memory bank) see fig. 3:

1. Select “Load” in the Message Type select-box ①.
2. Select number of requested patch to be changed ②.
3. Select requested values of all patch parameters<sup>2</sup> ③.
4. Click the “Generate” button ④.
5. The required hexadecimal SysEx message is generated as a text in text field on bottom of the window ⑤.
6. Copy the text in clipboard (CTRL+C) and paste (CTRL+V) to a MIDI Sysex software.
7. Send the message to the interface.
8. The interface will work with the new patch data till after the changed patch is recalled.

**Fig. 3 – Patch Data Load / Request**



### 1.1.4 PATCH DATA REQUEST

To request actual Patch Data values (patch parameters from a patch memory bank) see fig. 3:

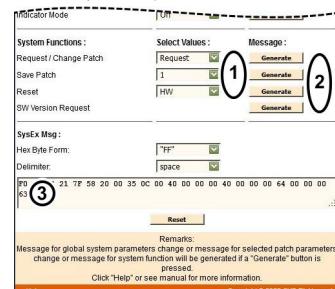
1. Select “Request” in the Message Type select-box ①.
2. Select number of requested patch ②.
3. Click the “Generate” button ④.
4. The required hexadecimal SysEx message is generated as a text in text field on bottom of the window ⑤.
5. **Copy** the text in clipboard (CTRL+C) and **paste** (CTRL+V) to a MIDI Sysex software.
6. Send the message to the interface.
7. The interface responds immediately – it sends “Bulk Dump - Patch Data Load” type SysEx message.
8. Now you can save the received message in your computer for further use.

### 1.1.5 SYSTEM FUNCTIONS PROCESSING

To execute a process with interface’s patches etc. (System Functions section of the generator window) see fig. 4:

1. Select value or type<sup>3</sup> of one requested system function ①.
2. Click the corresponding “Generate” button (in the same row) ②.
3. The required hexadecimal SysEx message is generated as a text in text field on bottom of the window ③.
4. Copy the text in clipboard (CTRL+C) and paste (CTRL+V) to a MIDI Sysex software.
5. Send the message to the interface.
6. The interface executes requested function immediately.

**Fig. 4 – System Functions**



<sup>2</sup> For details about the patch parameters see chapter 3.2.2.

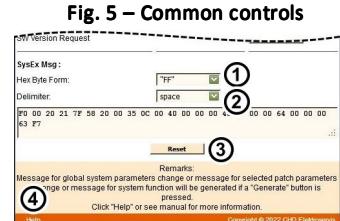
<sup>3</sup> Some functions don’t have a select box – skip this step. For details about System Functions, see chapter 3.3.



## 1.1.6 OTHER CONTROLLERS

The common controllers have the select / control elements on bottom of the window – see fig. 5:

1. Select “Hex Byte Form” ① and “Delimiter” character ② as required for your MIDI SysEx software<sup>4</sup>. Default setting of the generator is optimized for the recommended utility (see chapter 1.2). However some DAW or MIDI SysEx utilities require different format of the generated text message.
2. To clear the text field and return all values to their defaults, click the “Reset” button ③.
3. The “Help” link ④ opens new window with brief help.



## 1.2 RECOMMENDED THIRD PARTY SOFTWARE

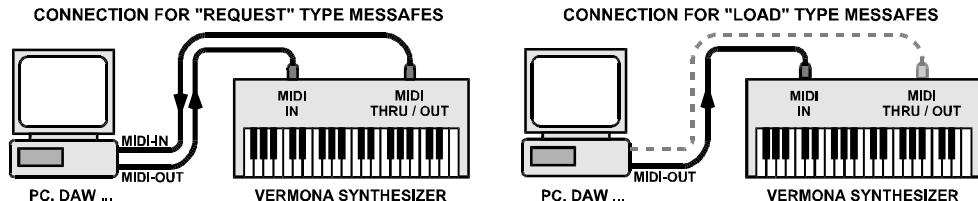
The hexadecimal MIDI SysEx messages created in a SysEx Generator (as described in chapter 1.1) are in plain text format. The text can not be saved as a \*.syx or \*.mid file directly, hence a text to SysEx capable utility or DAW is needed.

**Pocket MIDI**<sup>5</sup> is a utility that can be used to send the text as a SysEx message. It is a simple MIDI monitoring tool for both Windows and Mac OSX platforms. This is a freeware for commercial, non-profit or private use.

### 1.2.1 SETTING UP THE COMPUTER AND SOFTWARE

1. Download the **Pocket MIDI** utility at <https://www.morson.jp/pocketmidi-webpage/>
2. Install the utility in your computer.
3. Select the MIDI interface **Outputs** to device where the interface is connected to<sup>6</sup> (drop-down menu **Views** → **MIDI Settings** → **Input Port / Output Port**).
4. Connect the computer with the interface (accordingly to the fig. 6)<sup>7</sup>.
5. Send and receive required SysEx messages to / from the interface

Fig. 6 – Connection for Bulk Dump



<sup>4</sup> See the documentation of your DAW for required format. Form of hex bytes and delimiter must be set before a Generate button is pressed.

<sup>5</sup> Pocket MIDI is Copyright © MORSON JAPAN Co.,Ltd. All rights reserved.

<sup>6</sup> Computer's MIDI interface must be active / switched on and all necessary MIDI drivers correctly installed.

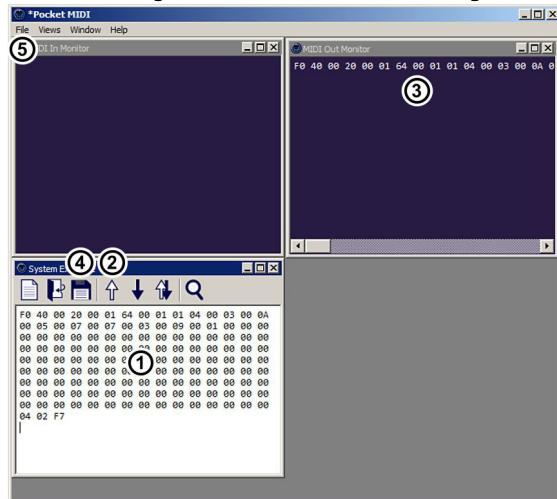
<sup>7</sup> MIDI cable from interface's MIDI output to computer's MIDI input is not necessary for "Load" type messages but it can remain connected.

**Fig. 7 – Pocket MIDI for data sending**

### 1.2.2 SEND THE TEXT SYSEX MESSAGE TO THE INTERFACE

Basic procedure to send any SysEx data to a MIDI device (fig. 7):

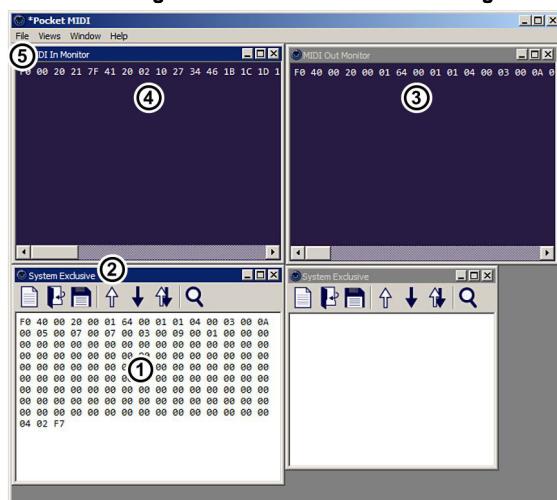
1. Generate required (“Load” or “Function” type) SysEx message in the SysEx Messages Generator (as described in Chapter 1)<sup>8</sup>.
2. Copy the text in clipboard (CTR+C) and paste (CTRL+V) in “System Exclusive” window①.
3. Click the “↑” (“Transmit”) arrow ② to send the data to the interface.
4. The sent message appears in the “MIDI Out Monitor” window ③.
5. Alternatively you can save the file for future use (either by the floppy icon ④ as a \*.txt file or in the drop-down “File” menu ⑤ as a \*.pocketmidi file).



### 1.2.3 REQUEST THE SYSEX DATA FROM THE INTERFACE

Basic procedure to request a SysEx data from a MIDI device and to record the received data (fig. 8):

1. Generate required (“Request”<sup>9</sup> or “Function”<sup>10</sup> type) SysEx message in the SysEx Messages Generator (as described in Chapter 1)<sup>11</sup>.
2. Copy the text in clipboard (CTR+C) and paste (CTRL+V) in “System Exclusive” window ①.
3. Click the “↑” (“Transmit”) arrow ② to send the data to the interface.
4. The sent message appears in the “MIDI Out Monitor” window ③.
5. The interface responds with “Load” type SysEx message. The received message appears in the “MIDI In Monitor” window ④.

**Fig. 8 – Pocket MIDI for data receiving**


<sup>8</sup> The correct “FF” Hex Byte Form and “‘ (space)” for Delimiter are the initial values after the SysEx Generator is launched, so there is no need to change them for Pocket MIDI.

<sup>9</sup> i.e. Bulk Dump – Global Data Request or Bulk Dump – Patch Data Request message.

<sup>10</sup> i.e. Patch Nr. Request or Firmware Version Request message.

<sup>11</sup> The correct “FF” Hex Byte Form and “‘ (space)” for Delimiter are the initial values after the generator is launched, so there is no need to change them for Pocket MIDI.

6. Click on the “**MIDI In Monitor**” window heading and save the data (File downdrop menu **(5)**) as a \*.pocketmidi file for future use.
7. Alternatively you can copy the data (CTRL+C) from “**MIDI In Monitor**” window **(4)** and paste (CTRL+V) to any text editor and save as a \*.txt or document file.

#### 1.2.4 ALTERNATIVE SOFTWARE SOLUTIONS

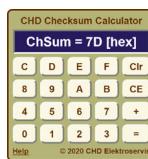
There are various DAW and MIDI utilites that can be used for the communication with VS-MIDI, both commercial and free. Among the others:

- MIDI-OX - advanced MIDI tool for MS Windows computers: <http://www.midiox.com/>
- Bome SendsX - simple and effective MS Windows MIDI utility: <https://www.bome.com/products/sendsx>
- InerziaSysEx - advanced MIDI tool for Mac OSX: available on iTunes
- Steinberg Cubase Pro - DAW for both MS Windows and Mac OSX computers: <http://www.steinberg.net>
- Apple Logic Pro - Mac OSX DAW with full sysex support: <https://www.apple.com/logic-pro/>
- Cakewalk Sonar - MS Windows DAW with full sysex support: <http://www.cakewalk.com/>
- etc.

### 1.3 CHECKSUM CALCULATOR

Fig. 9 – CHD Checksum Calculator

If you need to calculate simply the checksum byte necessary for any SysEx message, you can use a special software **CHD Checksum Calculator** (fig. 9). The calculator is based on Java script so it can be launched under any platform (PC, MAC, ...) which includes a web browser. The Calculator with guide how to use it is available at our “Support” web page.



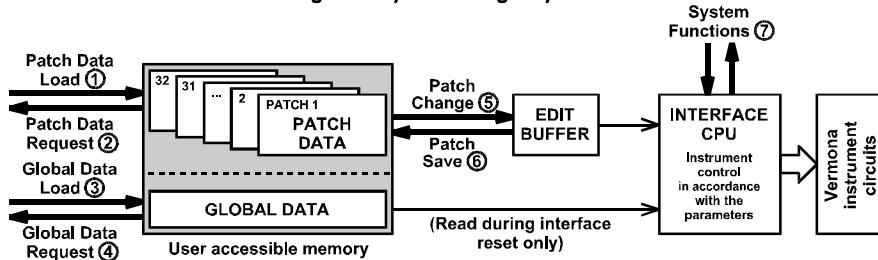


## 2 SYSTEM EXCLUSIVE COMMUNICATION DESCRIPTION

### 2.1 SYSEX MESSAGES SYSTEM

VS-MIDI interface uses System Exclusive communication to receive and to transmit a SysEx Messages for the control of all parameters in global and patch memory banks of the interface. It is also possible to list an actual memory banks content. These data can be backed up in the PC. Certain system and testing functions can be launched by SysEx commands as well.

Fig. 10 – SysEx Messages system



#### ① Bulk Dump - Patch Data Load

Response to the "Bulk Dump - Patch Data Request" message or change of the content of a patch in the internal interface's memory (chapter 3.2.2).

#### ② Bulk Dump - Patch Data Request

Request for the patch data in a patch in the internal interface's memory (chapter 3.1.2).

#### ③ Bulk Dump - Global Data Load

Response to the "Bulk Dump - Global Data Request" message or change of the content of global data memory bank in the internal interface's memory (chapter 3.2.1).

#### ④ Bulk Dump - Global Data Request

Request for the content of global data memory bank in the internal interface's memory (chapter 3.1.1).

#### ⑤ System Function – Request / Change Patch

Command for a change of actual interface patch number (chapter 3.3.1).

#### ⑥ System Function – Save Patch

Command for saving of the data from the edit buffer to an interface patch memory (chapter 3.3.2).

#### ⑦ System Functions - Others

Launches a system or a service function. Exact procedure is specified by the values of the "Address" and "Data" bytes (chapter 3.3).

## 2.2 SYSEX MESSAGES STRUCTURE

The VS-MIDI receives / transmits own specific messages with the following structure:

[hex]	[bin]	byte
<b>F0</b>	11110000	Start SysEx
<b>00 20 21</b>	00000000 00100000 00100001	Manufacturer ID (3 byte)
<b>ii</b>	Oiiiiiii	Device ID (00h ~ 0Fh or “Universal ID” 7Fh)
<b>58</b>	01011000	Model ID
<b>cc</b>	0ccc0000	Command (10h / 20h / 30h / 40h)
<b>aa</b>	Oaaaaaaaaa	Address
<b>dd ... dd</b>	0ddddddd ... 0ddddddd	Datablock (0 ~ 15 data bytes in dependence on the message type)
<b>xx</b>	0xxxxxxxxx	Checksum
<b>F7</b>	11110111	End SysEx

### 2.2.1 “SOX” BYTE

Each SysEx message must begin with the **F0h** “Start of SysEx” byte.

### 2.2.2 “MANUFACTURER ID” BYTES

The three Manufacturer ID bytes are always **00h 20h 21h**.

### 2.2.3 “DEVICE ID” BYTE

The device ID byte identifies the MIDI device exactly. It is equal to a number of an active MIDI channel (00h for channel Nr. 1, 01h for channel Nr. 2 etc. up to 0Fh for channel Nr. 16). The next possible value of the Device ID is 7Fh – universal Device ID always accepted by the interface regardless on an active MIDI channel number. For all other values of the Device ID (from 10h to 7Eh) the message is considered to be invalid and it is ignored by the interface.

### 2.2.4 “MODEL ID” BYTE

The Model ID byte is always **58h** for the VS-MIDI interface.

### 2.2.5 “COMMAND” BYTE

The “cc” command byte specifies the Bulk Dump message type or the interface function type (see chapter 3). It defines the interface activity after the whole message is received.

### 2.2.6 “ADDRESS” BYTE

The “aa” address byte specifies the memory area or selects the system function accordingly to a type of the SysEx message – see below.

### 2.2.7 “DATA” BYTES

The block of data bytes “dd...dd” contains the values of a parameters or a function defined by the SysEx message type. The number of bytes in the data block “dd...dd” is a different accordingly to a type of the SysEx message. It consists from none to 15 data bytes.

### 2.2.8 “CHECKSUM” BYTE

The “xx” checksum byte confirms validity of the System Exclusive message. It must be calculated as 7-bit complement of the sum of bytes from ‘Model ID’ to ‘Data’ (by other words, seven-bit sum of bytes from the “Model ID” to “Checksum” must be equal to zero). If the checksum byte is invalid, the whole message is invalid too and it is ignored by the interface.

### 2.2.9 “EOX” BYTE

Each SysEx message must be ended with the **F7h** “End of SysEx” byte.

### 3 SYSEX COMMANDS OVERVIEW

Valid values of the "cc" command byte are:

Table 1 – Commands

"cc" Byte	Command Name
10h	Bulk Dump – Data Request (Patch / Global)
20h	Bulk Dump – Data Load (Patch / Global)
30h	System Functions
40h	Service (user inaccessible – see chapter 3.4)

#### 3.1 BULK DUMP - DATA REQUEST

The "Bulk Dump - Data Request" command ("cc"=10h, ② and ④ on fig. 10) is request for content of a data memory bank. It launches the listing of the data from the selected memory bank (i.e. parameters stored in user memory).

##### 3.1.1 GLOBAL DATA REQUEST

Form of the message:

[hex]	[bin]	byte
F0	11110000	Start SysEx
00 20 21	00000000 00100000 00100001	Manufacturer ID (3 byte)
ii	Oiiiiiii	Device ID (00h ~ 0Fh or "Universal ID" 7Fh)
58	01011000	Model ID
10	00010000	Command (Bulk Dump - Global Data Request)
20	00100000	Address (always 20h)
xx	0xxxxxxx	Checksum
F7	11110111	End SysEx

The "aa" address is always 20h. The message is considered to be invalid for all other values and it is ignored by the interface.

Data bytes "dd" are not used in this type of the SysEx message.

Immediately after the "Bulk Dump - Global Data Request" message reception, the interface sends the "Bulk Dump - Global Data Load" message (see chapter 3.2.1) as a reply - data of all parameters of global memory bank are included in this message.

##### 3.1.2 PATCH DATA REQUEST

Form of the message:

[hex]	[bin]	byte
F0	11110000	Start SysEx
00 20 21	00000000 00100000 00100001	Manufacturer ID (3 byte)
ii	Oiiiiiii	Device ID (00h ~ 0Fh or "Universal ID" 7Fh)
58	01011000	Model ID
10	00010000	Command (Bulk Dump - Patch Data Request)
aa	000xxxxx	Address (00h ~ 1Fh → patch number 1 ~ 32)
xx	0xxxxxxx	Checksum
F7	11110111	End SysEx

The address "aa" is from 00h to 1Fh. It corresponds to the number from 1 to 32 of a requested patch.

Databytes "dd" are not used in this type of the SysEx message.

Immediately after the "Bulk Dump - Patch Data Request" message reception, the interface sends the "Bulk Dump - Patch Data Load" message (see chapter 3.2.2) as a reply - data of all parameters of requested patch from patch memory bank are included in this message.

## 3.2 BULK DUMP - DATA LOAD

Transmitting the "Bulk Dump - Data Load" message ("cc"=20h, ① and ③ on fig. 10) to the interface changes content of a memory bank saved in the user interface's memory. This SysEx message is also sent from the interface as a reply to the "Bulk Dump - Data Request" message (see chapter 3.1).

### 3.2.1 GLOBAL DATA LOAD

Form of the message:

[hex]	[bin]	byte
F0	11110000	Start SysEx
<b>00 20 21</b>	00000000 00100000 00100001	Manufacturer ID (3 byte)
ii	Oiiiiiiii	Device ID (00h ~ 0Fh or "Universal ID" 7Fh)
<b>58</b>	01011000	Model ID
<b>20</b>	00100000	Command (Bulk Dump - Global Data Load)
<b>20</b>	00100000	Address (always 20h)
<b>d1...d8</b>	0ddddd... 0ddddd	Data block (8 data bytes – see table 2)
xx	0xxxxxxx	Checksum
<b>F7</b>	11110111	End SysEx

The "aa" address is always 20h. The message is considered to be invalid for all other values and it is ignored by the interface.

The data block "d1...d8" always contains 8 bytes with values of the global parameters:

Table 2 – Global Data

Byte order	Parameter	Valid Range	Byte order	Parameter	Valid Range
1 <sup>st</sup>	MIDI Channel	0 ~ 15	5 <sup>th</sup>	Calibration Constant	0 ~ 127
2 <sup>nd</sup>	VCF CC Nr.	0 ~ 119	6 <sup>th</sup>	[reserve]	0
3 <sup>rd</sup>	VCA CC Nr.	0 ~ 119	7 <sup>th</sup>	[reserve]	0
4 <sup>th</sup>	EG Break Pulse Duration	0 ~ 58	8 <sup>th</sup>	[reserve]	0

If received value of any parameter exceeds the valid range listed above, the parameter is not changed and its original value remains in the memory. For detailed description of the global parameters see the Owner's Manual.

**Remark:** Note that the interface works with changed parameters till after next interface reset!

### 3.2.2 PATCH DATA LOAD

Form of the message:

[hex]	[bin]	byte
F0	11110000	Start SysEx
<b>00 20 21</b>	00000000 00100000 00100001	Manufacturer ID (3 byte)
ii	Oiiiiiiii	Device ID (00h ~ 0Fh or "Universal ID" 7Fh)
<b>58</b>	01011000	Model ID
<b>20</b>	00100000	Command (Bulk Dump - Patch Data Load)
<b>aa</b>	000xxxxx	Address (00h ~ 1Fh → patch number 1 ~ 32)
<b>d1...d15</b>	0ddddd... 0ddddd	Data block (15 data bytes – see table 3)
xx	0xxxxxxx	Checksum
<b>F7</b>	11110111	End SysEx

The "aa" address is from 00h to 1Fh. It corresponds to the number from 1 to 32 of a requested preset.

The data block "d1...d15" always contains 15 bytes with values of the patch parameters:

Table 3 – Patch Data					
Byte order	Parameter	Valid Range	Byte order	Parameter	Valid Range
1 <sup>st</sup>	VCO MIDI Notes Shift	0 ~ 84	9 <sup>th</sup>	VCA Velocity Amount	0 ~ 127
2 <sup>nd</sup>	VCO Pitch Bend Range	0 ~ 12	10 <sup>th</sup>	VCA Chnl Aftertouch Amount	0 ~ 127
3 <sup>rd</sup>	VCF Control Mode	0 ~ 2	11 <sup>th</sup>	EG Retrigger Mode	0 ~ 2
4 <sup>th</sup>	VCF Key Follow	0 ~ 127	12 <sup>th</sup>	EG Retrigger Rate	0 ~ 127
5 <sup>th</sup>	VCF Velocity Amount	0 ~ 127	13 <sup>th</sup>	Indicator Mode	0 ~ 3
6 <sup>th</sup>	VCF Chnl Aftertouch Amount	0 ~ 127	14 <sup>th</sup>	[reserve]	0
7 <sup>th</sup>	VCA Control Mode	0 ~ 2	15 <sup>th</sup>	[reserve]	0
8 <sup>th</sup>	VCA Key Follow	0 ~ 127			

If received value of any parameter exceeds the valid range listed above, the parameter is not changed and its original value remains in the memory. For detailed description of the patch parameters see the Owner's Manual.

### 3.3 SYSTEM FUNCTIONS

The "System Function" commands ("cc"=70h, ⑤, ⑥, ⑦ on fig. 10) are direct commands for the execution of a system function.

Form of the message:

[hex]	[bin]	byte
F0	11110000	Start SysEx
00 20 21	00000000 00100000 00100001	Manufacturer ID (3 byte)
ii	Oiiiiiii	Device ID (00h ~ 0Fh or "Universal ID" 7Fh)
58	01011000	Model ID
40	01000000	Command (System Function)
aa	000000xx	Address (00h ~ 03h → the function type – see table 4)
dd	0ddddddd	Data block (1 data byte)
xx	0xxxxxxxxx	Checksum
F7	11110111	End SysEx

The type of the function is defined by the "aa" address:

Table 4 – System Functions		
"aa" Byte	Function Name	Meaning
00h	Patch Nr. Change / Request	Command for the actual interface patch change or request for the number of active interface patch number.
01h	Patch Save	Command for saving of the data from the edit buffer to an interface patch memory
02h	Interface Reset	Command for execution of warm HW reset or total factory reset of the interface.
03h	Firmware Version Request	Request for installed version of the interface firmware.

The datablock "dd" always contains only one databyte which specifies an activity of selected system function.

Immediately after the "Bulk Dump Prest Data Request" message reception, the interface executes the requested function.

#### 3.3.1 PATCH NR. CHANGE / REQUEST

If the "aa" address is equal to 00h, the SysEx Msg is command for change of active interface patch number or it is request for the number of just active interface patch number in dependence on value of the "dd" databyte:

If value of the "**dd**" databyte is from 00h to 1Fh, the SysEx message is a command for the active interface patch change<sup>12</sup> (see ⑤ on fig. 10). The value of "**dd**" databyte (00h to 1Fh) specifies the number of newly selected interface patch (1 to 32).

If value of the "**dd**" databyte is from 20h to 7Fh, the message is inquiry for number of just active patch. In this case, the patch is not changed and the interface immediately transmits info about the active patch number to the MIDI output. The form of transmitted SysEx message is following:

**F0 00 20 21 ii 58 40 00 dd xx F7 [hex],**

where the "**dd**" databyte is the number from 00h to 1Fh which defines the number of the active patch number (1 to 32). If the request is received after the interface reset when none patch is still active, the "**dd**" databyte value is 7Fh.

### 3.3.2 PATCH SAVE

If the address "**aa**" is equal to 01h, the SysEx Msg is a command for saving of the data from the edit buffer to an interface patch memory (see ⑥ on fig. 10). Value of the "**dd**" databyte can be from 00h to 1Fh - it specifies the number of a patch in the memory (1 to 32) to which the data will be stored. For all other values of the "**dd**" databyte, the message is considered to be invalid and it is ignored by the interface.

### 3.3.3 INTERFACE RESET

If the address "**aa**" is equal to 02h, the SysEx Msg is the command for execution of the interface reset (see ⑦ on fig. 10). The value of databyte "**dd**" can only be 00h or 7Fh. For all other values, the message is considered to be invalid and it is ignored by the interface.

If the value of databyte "**dd**" is 00h, the "Warm Hardware Reset" of the interface is executed. The interface is set to the same status (Stand-by) as when the instrument is switched on.

If the value of databyte "**dd**" is 7Fh, the complete "Factory Reset" is executed. The interface is initialized to the factory status (all parameters in both global and patch memory banks are set to the default values).

**WARNING - All data stored by user in the internal memory are fully erased after the execution of the "Factory Reset" command.** It is recommended to make a back up of all user data before the "Factory Reset" execution.

### 3.3.4 FIRMWARE VERSION REQUEST

If the address "**aa**" is equal to 03h, the SysEx Msg is a request for the version number of the installed firmware (see ⑦ on fig. 10). The value of databyte "**dd**" can only be 00h. For all other values the message is considered to be invalid and it is ignored by the interface.

Immediately after the request is received, the interface transmits the information about the firmware version to the MIDI output. The form of transmitted SysEx message is following:

**F0 00 20 21 ii 58 40 03 dd xx F7 [hex],**

where nibbles of the "**dd**" databyte are identifying installed firmware version. For example: Byte 20h (i.e. nibbles 02h, 00h) means the version "2.0".

## 3.4 SERVICE FUNCTIONS

The "Service" command ("**cc**"= 40h) is used only for the debugging of the firmware and for the special service and test procedures of the interface hardware. The service functions are not expected to be used during regular operation so they are ignored in standard working modes of the interface.

<sup>12</sup> This message is an equivalent for the "Program Change" standard MIDI command with values from 0 to 31.

## 4 EXAMPLES OF SYSEX MESSAGES

### 4.1 EXAMPLE 1 – CHANGE GLOBAL PARAMETERS

#### Task:

Set the global data of the interface to the following values:

- MIDI Channel: 0Fh (i.e. channel Nr. 16)
- VCF CC Nr.: 76h (i.e. CC #118)
- VCA CC Nr.: 77h (i.e. CC #119)
- EG Break Pulse Duration: 06h (i.e. 4 ms)
- Calibration Constant: 40h (i.e. none shift of CV)

#### Solution:

- I don't know what number of the SysEx ID is actually active so the universal Device ID (**7Fh**) will be used.
- The command to be used is the “**Bulk Dump – Global Data Load**”, the command byte will be **20h**.
- The address byte will be **20h** for the global data bank.
- Start of the SysEx Msg will be then:

Start SysEx : **F0h**  
Mfr ID : **00h 20h 21h**  
Device ID : **7Fh**  
Model ID : **58h**  
Command : **20h**  
Address : **20h**

- All five parameter bytes of the datablock are given by the task:

**0Fh 76h 77h 06h 40h**

- The whole datablock has 8 byte – three reserve positions (6<sup>th</sup>, 7<sup>th</sup>, 8<sup>th</sup>) must be filled by 00h. Whole datablock will be:  
**0Fh 76h 77h 06h 40h 00h 00h 00h**
- Now, the checksum must be calculated as 7-bit complement of the sum of bytes from ‘Model ID’ to ‘Data’:  $00h - (58h + 20h + 20h + 0Fh + 76h + 77h + 06h + 40h + 00h + 00h + 00h) = 26h$
- The form of the whole required System Exclusive message is after the checksum and End SysEx byte refilling: **F0h 00h 20h 21h 7Fh 58h 20h 20h 0Fh 76h 77h 06h 40h 00h 00h 00h 26h F7h**

When the message is received, the values of all global parameters in the global memory bank will be set accordingly to the SysEx message data. Note that to use the newly stored data, the interface needs to be reset.

### 4.2 EXAMPLE 2 – CHANGE PATCH PARAMETERS

#### Task:

Set the patch data of the patch memory Nr. 1 to the following values:

- VCO MIDI Notes Shift: 29h (i.e. from MIDI Note Nr. 41)
- VCO Pitch Bend Range: 02h (i.e. ±2 semitones)
- VCF Control Mode: 01h (i.e. Gate)
- VCF Key Follow: 40h (i.e. none)
- VCF Velocity Amount: 40h (i.e. middle position)
- VCF Chnl Aftertouch Amount: 00h (i.e. none)
- VCA Control Mode: 01h (i.e. Gate)
- VCA Key Follow: 40h (i.e. none)
- VCA Velocity Amount: 00h (i.e. none)
- VCA Chnl Aftertouch Amount: 00h (i.e. none)
- EG Retrigger Mode: 01h (i.e. Fixed rate)
- EG Retrigger Rate: 48h (i.e. 4 Hz)
- Indicator Mode: 03h (i.e. Retrigger)

**Solution:**

- I don't know what number of the SysEx ID is actually active so the universal Device ID (**7Fh**) will be used.
- The command to be used is the "**Bulk Dump – Patch Data Load**", the command byte will be **20h**.
- Data will be stored to patch memory Nr. 1, the address byte will be **00h**.
- Start of the SysEx Msg will be then:

Start SysEx : **F0h**

Mfr ID : **00h 20h 21h**

Device ID : **7Fh**

Model ID : **58h**

Command : **20h**

Address : **00h**

- All 13 parameter bytes of the datablock are given by the task. Whole datablock will be:  
**29h 02h 01h 40h 40h 00h 01h 40h 00h 00h 01h 4Bh 03h**
- The whole datablock has 15 byte – two reserve positions (14<sup>th</sup>, 15<sup>th</sup>) must be filled by 00h. Whole datablock will be:  
**29h 02h 01h 40h 40h 00h 01h 40h 00h 00h 01h 4Bh 03h 00h 00h**
- Now, the checksum must be calculated as 7-bit complement of the sum of bytes from 'Model ID' to 'Data':  
**00h – (58h + 20h + 00h + 29h + 02h + 01h + 40h + 40h + 00h + 01h + 40h + 00h + 01h + 4Bh + 03h + 00h + 00h) = 4Ch**
- The form of the whole required System Exclusive message is after the checksum and End SysEx byte refilling:  
**F0h 00h 20h 21h 7Fh 58h 20h 00h 29h 02h 01h 40h 40h 00h 00h 01h 4Bh 03h 00h 00h 4CF7h**

When the message is received, the values of all preset parameters and patch name in permanent patch memory Nr. 1 will be set accordingly to the SysEx message data. If the patch Nr. 1 is just set as active, it must be recalled again so that the interface will start to work with the newly stored data.

### 4.3 EXAMPLE 3 – SAVE EDIT BUFFER

**Task:**

Save the data from edit buffer (e.g. set previously by CCs) to patch memory Nr. 32.

**Solution:**

- I don't know what number of the SysEx ID is actually active so the universal Device ID (**7Fh**) will be used.
- The command to be used is the "**System Functions**", the command byte will be **30h**.
- The function is "**Patch Save**" – the address byte will be **01h** as shown in table 6.
- Start of SysEx Msg will be then:

Start SysEx : **F0h**

Mfr ID : **00h 20h 21h**

Device ID : **7Fh**

Model ID : **58h**

Command : **30h**

Address : **01h**

- Databyte depends on number of required patch memory; for patch Nr. 32 it is **1Fh** (see chapter 3.3.2). Since the datablock has only one byte, whole datablock will be then **1Fh**
- Now, the checksum must be calculated as 7-bit complement of the sum of bytes from 'Model ID' to 'Data':  
**00h – (58h + 30h + 01h + 7Fh) = 58h**
- The form of the whole required System Exclusive message is after the checksum and End SysEx byte refilling:  
**F0h 00h 20h 21h 7Fh 58h 30h 01h 1Fh 58h F7h**

When the message is received, the actual values of all parameters in the edit buffer will be copied to permanent patch memory Nr. 64.

#### 4.4 EXAMPLE 4 – FACTORY RESET

##### Task:

Make factory reset of the interface.

##### Solution:

- I don't know what number of the SysEx ID is actually active so the universal Device ID (**7Fh**) will be used.
- The command to be used is the “**System Functions**”, the command byte will be **30h**
- The function is “**Interface Reset**”, the address byte will be **02h** (see table 4)
- Start of SysEx Msg will be then:

Start SysEx : **F0h**

Mfr ID : **00h 20h 21h**

Device ID : **7Fh**

Model ID : **58h**

Command : **30h**

Address : **02h**

- Databyte for complete factory reset is **7Fh** (see chapter 3.3.3). Since the datablock has only one byte, whole datablock will be then **7Fh**
- Now, the checksum must be calculated as 7-bit complement of the sum of bytes from ‘Model ID’ to ‘Data’: **00h – (58h + 30h + 02 + 7Fh) = 77h**
- The form of the whole required System Exclusive message is after the checksum and End SysEx byte refilling:  
**F0h 00h 20h 21h 7Fh 58h 30h 02h 7Fh 77h F7h**

When the message is received, the interface is set to factory status.

---

All documents and support software are available at manufacturer's web pages.

---



Vermona Synthesizer MIDI Interface  
Model VS-MIDI, Nr. 8-434, ver. 2.00  
Document: 843420\_syxcom, rev. 1

Manufacturer: CHD Elektroservis, Czech Republic

[www.chd-el.cz](http://www.chd-el.cz)    [info@chd-el.cz](mailto:info@chd-el.cz)