

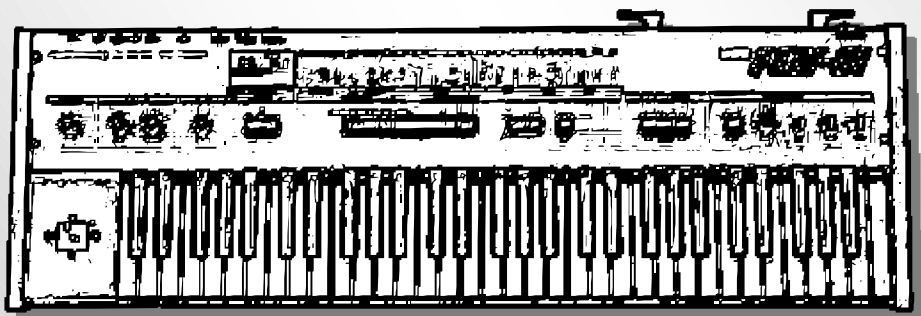
P61-KBD

MIDI Interface for

KORG POLY-61

Keyboard

Model 8-435
ver. 1.0



MIDI SYSTEM EXCLUSIVE COMMUNICATION
Rev. 2



© 2018 CHD Elektroservis



Contents

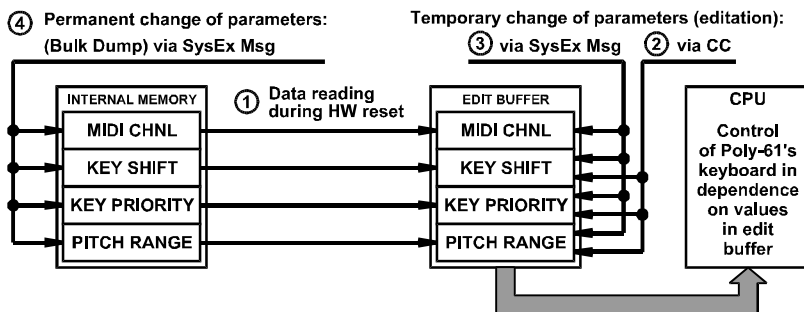
page

1	SYSTEM EXCLUSIVE COMMUNICATION	3
2	SYSEX MESSAGES STRUCTURE	3
2.1	"DEVICE ID" BYTE	3
2.2	"ADDRESS" BYTE	4
2.3	DATABLOCK	4
2.3.1	DATA FOR "MIDI CHANNEL" PARAMETER	4
2.3.2	DATA FOR "KEY SHIFT" PARAMETER	4
2.3.3	DATA FOR "KEY PRIORITY" PARAMETER	5
2.3.4	DATA FOR "PITCH BEND RANGE" PARAMETER	5
2.3.5	DATABLOCK FOR PERMANENT CHANGE OF ALL PARAMETERS	5
2.4	CHECKSUM BYTE	5
3	SYSEX MESSAGE CREATION	5
3.1	EXAMPLE 1	5
3.2	EXAMPLE 2	6
3.3	CHECKSUM CALCULATOR	7
3.4	SYSEX MESSAGES GENERATOR	7

1 SYSTEM EXCLUSIVE COMMUNICATION

P61-KBD interface disposes of system of System Exclusive communication which enables to receive a SysEx Messages for changes of all parameters in edit buffer and in permanent memory bank (see pic. 1).

Pic. 1 – Structure of interface’s memory



2 SYSEX MESSAGES STRUCTURE

TR2-KBD recognizes own specific messages for changes of parameters setting etc. with this structure:

[hex]	[bin]	byte
F0	11110000	Start SysEx
00	00000000	Manufacturer ID (always 3 bytes)
20	00100000	
21	00100001	
ii	0iiiiiii	Device ID
59	01011001	Model ID (P61-KBD)
aa	0aaaaaaa	Address ¹
dd	0ddddd	Data ²
..	
dd	0ddddd	
xx	0xxxxxxx	Checksum
F7	11110111	End SysEx

2.1 "DEVICE ID" BYTE

"Device ID" byte is equal to number of just active MIDI channel (00h for channel Nr. 1, 01h for channel Nr. 2 etc.).

Any of value from 00h to 0Fh of the Device ID is accepted if OMNI mode is active.

Next possible value of the Device ID is 7Fh - universal Device ID recognized whenever independently on just active MIDI channel number or OMNI mode.

¹ Address "aa" specifies parameter or memory area in dependence on type of SysEx Msg – see below.

² Block of "dd...dd" data bytes contents values of parameters. Number of bytes in the datablock is different in dependence on type of SysEx message - 1 or 4 data bytes.



For all others values of Device ID (from 10h to 7Eh), the message is evaluated as invalid and it is ignored by the interface.

2.2 “ADDRESS” BYTE

“Address” byte specifies parameter to be changed. Allowed address range is from 00h to 04h:

- aa** = 00h → **MIDI Channel** – temporary change of receiving MIDI channel in edit buffer
- aa** = 01h → **Key Shift** – temporary change of keyboard transpose in edit buffer
- aa** = 02h → **Key Priority** – temporary change of key priority parameter in edit buffer
- aa** = 03h → **Pitch Bend Range** – temporary change of pitch bend range in edit buffer
- aa** = 04h → **All Parameters** – permanent change of all parameters in internal permanent memory

For addresses **aa** = 00h to 03h, value of one parameter is changed only temporarily (in edit buffer – see ③ on pic. 1). New value is valid until next change is done or the instrument is turned off. These messages are equivalent to control via CCs (see ② on pic. 1).

For address **aa** = 04h, values of all parameters are changed permanently. New values are stored in internal memory (see ④ on pic. 1). Performed changes of the parameters values become evident **till after next turning of the instrument** (i.e. during the interface reset – see ① on pic. 1).

If the address is out of range 00h to 04h, the message is evaluated as invalid and it is ignored by the interface.

2.3 DATABLOCK

Data bytes specify new value of just adjusted parameter / parameters. If **aa** address is from 00h to 03h, datablock contents just one byte which represents value of selected parameter. For **aa** address equal to 04h, datablock contents four bytes which represent values of all parameters (in order: MIDI Channel, Key Shift, Key Priority, Pitch Bend Range).

2.3.1 DATA FOR “MIDI CHANNEL” PARAMETER

If **aa** Address is equal to 00h, one **dd** Data byte specifies new number of receiving MIDI channel. Valid value of **dd** byte is from 00h to 10h.

Values from 00h to 0Fh represent the MIDI channel numbers from 1 to 16. When the value is equal to 10h, OMNI mode will be active.

For values from 11h to 7Fh, the message is evaluated as invalid and it is ignored by the interface.

2.3.2 DATA FOR “KEY SHIFT” PARAMETER

If **aa** Address is equal to 01h, one **dd** Data byte specifies keyboard transpose – it means relation between MIDI Note number and assigned key. Valid value of **dd** byte is from 00h to 67h.

The value specifies number of MIDI note which will be assigned to first key from left (i.e. to lowest C) of the instrument’s keyboard.

For values from 68h to 7Fh, the message is evaluated as invalid and it is ignored by the interface.



2.3.3 DATA FOR “KEY PRIORITY” PARAMETER

If **aa** Address is equal to 02h, one **dd** Data byte specifies selected key priority mode. Valid value of **dd** byte is from 00h to 03h.

“Last Key” priority is selected for value equal to 00h, “Higher Key” priority is selected for value equal to 01h, “Lower Key” priority is selected for value equal to 02h and “None” priority is selected for value equal to 03h.

For values from 04h to 7Fh, the message is evaluated as invalid and it is ignored by the interface.

2.3.4 DATA FOR “PITCH BEND RANGE” PARAMETER

If **aa** Address is equal to 03h, one **dd** Data byte adjusts maximal range of MIDI Pitch Wheel (Pitch Bend) controller. Valid value of **dd** byte is from 00h to 18h.

For value 00h, Pitch Bend controller will be inactive – its movement will have no effect. Values from 01h to 18Fh represent allowed range of Pitch Wheel controller in semitones (from ± 1 to ± 24 semitones).

For values from 19h to 7Fh, the message is evaluated as invalid and it is ignored by the interface.

2.3.5 DATABLOCK FOR PERMANENT CHANGE OF ALL PARAMETERS

If **aa** Address is equal to 04h, **dd...dd** datablock contains four bytes which represent values of all parameters. Conditions for values of partial parameters are described above. Order of parameters in block is: MIDI Channel, Key Shift, Key Priority, Pitch Bend Range.

2.4 CHECKSUM BYTE

xx checksum byte confirms validity of System Exclusive message. It must be calculated so that seven-bit sum of bytes from “Model ID” to “Checksum” is equal to zero³ (i.e. checksum is 7-bit complement of sum from “Model ID” to “Datablock” bytes).

If checksum byte is invalid, whole message is invalid too and it is ignored by the interface.

3 SYSEX MESSAGE CREATION

3.1 EXAMPLE 1

Task:

Set parameters values in permanent interface’s memory to to these values:

- MIDI CHANNEL : 0Ah (i.e. MIDI Channel Nr. 11)
- KEY SHIFT : 24h (i.e. first acceptable MIDI Note is Note Nr. 36)
- KEY PRIORITY : 01h (i.e. Higher Key priority)
- PITCH BEND RANGE : 18h (i.e. ± 24 semitones)

Solution:

- We don’t know what number of MIDI channel is just active so universal Device ID (**7Fh**) will be used,
- Necessary address for changes of all parameters will be **04h**
- Datablock (4 bytes) will content values of parameters given by task: **0Ah 24h 01h 18h**

³ See chapter 3.3 for easy calculation of the Checksum byte.



Start of SysEx Msg will be then :

Start SysEx : **F0h**
Mfr ID : **00h 20h 21h**
Device ID : **7Fh**
Model ID : **59h**
Address : **04h**
Data bytes : **0Ah 24h 01h 18h**

- Now, checksum must be calculated as 7-bit complement of sum of bytes from 'Model ID' to 'Data', i.e.:
 $00h - (59h + 04h + 0Ah + 24h + 01h + 18h) = 5Ch$
- Form of whole required System Exclusive message is after checksum and 'End SysEx' byte refilling:
F0h 00h 20h 21h 7Fh 5Ah 04h 0Ah 24h 01h 18h 5Ch F7h

After that message is received, all parameters will be set to requested values but the interface will start working with newly stored values till next reset.

3.2 EXAMPLE 2

Task:

Set temporarily the receiving MIDI channel to Nr. 1

Solution:

- We don't know what number of MIDI channel is just active so universal Device ID (**7Fh**) will be used,
- Necessary address for temporary change of MIDI Channel number will be **00h**
- Requested MIDI channel number is 1 so data byte will be **00h**

Start of SysEx Msg will be then :

Start SysEx : **F0h**
Mfr ID : **00h 20h 21h**
Device ID : **7Fh**
Model ID : **59h**
Address : **00h**
Data byte : **00h**

- Now, checksum must be calculated as 7-bit complement of sum of bytes from 'Model ID' to 'Data', i.e.:
 $00h - (59h + 00h + 00h) = 27h$
- Form of whole required System Exclusive message is after checksum and 'End SysEx' byte refilling:
F0h 00h 20h 21h 7Fh 59h 00h 00h 27h F7h

Immediately after that message is received, the interface starts to accept MIDI messages on MIDI channel Nr. 1.

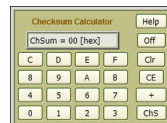


3.3 CHECKSUM CALCULATOR

If you want to create a SysEx message yourself, you need to calculate the 'Checksum' byte. This is difficult for most musicians because calculation with hexadecimal / binary numbers is necessary. For easy calculation of the checksum, special software **Checksum Calculator** is prepared.

The Checksum Calculator is based on Java scripts so it can run on any computer with web browser (Windows, OSX, etc.). Note that scripts and ActiveX elements must be enabled in the web browser for proper function of the calculator.

The Checksum Calculator is available at our website (<http://www.chd-el.cz>) on Support page.



The Checksum Calculator works on-line or it can be downloaded to your computer and then launched from it.

3.4 SYSEX MESSAGES GENERATOR

As a support for the users we have made special software generator to create any SysEx messages to control the interface. Usage of this generator is very easy for any user. Please see Owner's Manual of the interface for detailed description of SysEx Messages Generator.

The SysEx Generator is available at our website (<http://www.chd-el.cz>).

Document :
8435_syxcom_rev2

Manufacturer :
CHD Elektroservis
Nad kundratkou 27, 19000 Praha 9, Czech Republic

info@chd-el.cz www.chd-el.cz

