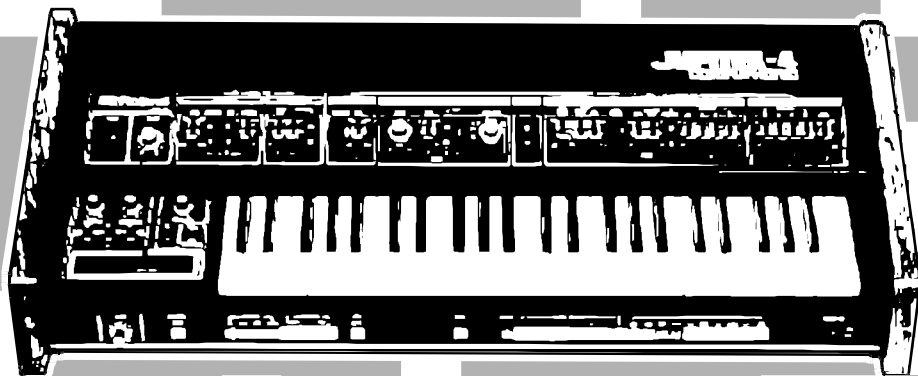


JP4-KBD

MIDI Interface for
Roland Jupiter – 4
Keyboard

Model 8-432
Version 2.0



SysEx Communication

Rev. 2



© 2019 CHD Elektroservis



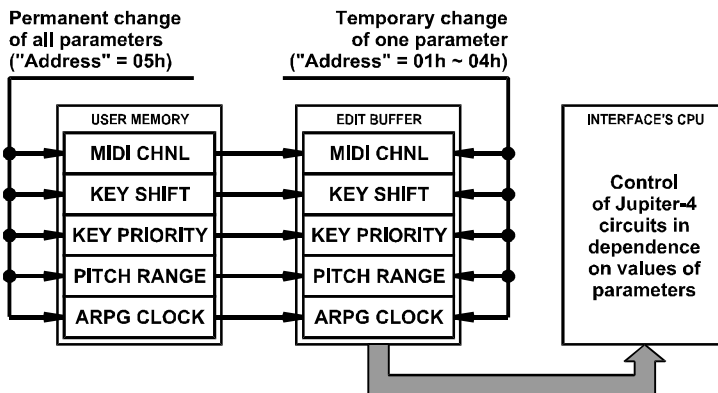
Contents:

| | | |
|----------|---|----------|
| 1 | SYSTEM EXCLUSIVE COMMUNICATION | 3 |
| 2 | SYSEX MESSAGES STRUCTURE | 3 |
| 2.1 | “DEVICE ID” BYTE..... | 3 |
| 2.2 | “ADDRESS” BYTE..... | 4 |
| 2.3 | “DATA” BYTE(S) | 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 | DATA FOR ARPEGGIO CLOCK RATE PARAMETER..... | 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..... | 6 |
| 3.4 | SYSEX MESSAGES GENERATOR..... | 7 |

1 SYSTEM EXCLUSIVE COMMUNICATION

JP4-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 (see fig. 1).

Figure. 1 – Structure of interface’s memory



2 SYSEX MESSAGES STRUCTURE

JP4-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 |
| 56 | 01010110 | Model ID |
| aa | 0aaaaaaa | Address ¹ |
| dd | 0ddddddd | Data ² |
| .. | | |
| dd | 0ddddddd | |
| 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 of the interface is active.

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

² Block "dd...dd" of data bytes contents values of parameters. Number of bytes in the datablock is different in dependence on the "Address" – 1 or 5 data bytes.

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

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(s) to be changed. Allowed address range is from 00h to 05h:

aa = 00h → MIDI Channel - Receiving MIDI channel temporary change in edit buffer

aa = 01h → Key Shift - Keyboard transpose temporary change in edit buffer

aa = 02h → Key Priority - Key priority parameter temporary change in edit buffer

aa = 03h → Pitch Wheel Range - Pitch bend range temporary change in edit buffer

aa = 04h → Arpeggio Clock Rate - Arpeggiator clock source selection temporary change in edit buffer

aa = 05h → All – permanent change of values of all parameters in internal memory of the interface

For addresses aa = 00h to 04h, value of only one parameter is changed in edit buffer temporarily - new value is valid until next change is done or the instrument is turned off.

For address aa = 05h, values of all parameters are changed permanently - new values are stored in internal memory.

If the address is out of range 00h to 05h, the message is evaluated as invalid and it is ignored by the interface.

2.3 “DATA” BYTE(S)

Data byte specifies new value for parameter selected by “aa” address. Its value depends on required value of the parameter.

For addresses 01h to 04h there is only one data byte which includes value of one parameters selected by the Address (see chapter 2.2).

For address 05h, there is block of five databytes which includes values of all parameters³ in this order:

- 1) MIDI Channel
- 2) Key Shift
- 3) Key Priority
- 4) Pitch Bend Range
- 5) Arpeggio Clock Rate

2.3.1 DATA FOR MIDI CHANNEL PARAMETER

Values from 00h to 0Fh represent the MIDI channel numbers from 1 to 16.

If the value is 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

Value of the data byte corresponds to the parameter value directly – valid range is from 00h to 4Fh (i.e. +0 to +79 semitone shift) thus.

For values from 50h to 7Fh, the message is evaluated as invalid and it is ignored by the interface.

³ Values of all parameters must be in acceptable range. If any of data bytes is out of range, the message is evaluated as invalid and it is ignored by the interface.



2.3.3 DATA FOR KEY PRIORITY PARAMETER

"**Last Key**" priority is selected for value of the data byte equal to 00h, "**Higher Key**" priority is selected for the value equal to 01h, "**Lower Key**" priority is selected for the value equal to 02h and "**None**" priority is selected for the 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

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 Bend 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 DATA FOR ARPEGGIO CLOCK RATE PARAMETER

If the value is 00h, internal instrument's arpeggio clock generator is selected.

For values from 01h to 7Fh, tempo of the arpeggiator is derived from MIDI Clock – see chapter 3.5 in Owner's Manual.

2.4 "CHECKSUM" BYTE

Checksum byte "**xx**" confirms validity of the System Exclusive message. It must be calculated so that seven-bit sum of bytes from "Model ID" to "Checksum" is equal to zero ⁴ (i.e. checksum is 7-bit complement of sum from "Model ID" to "Data" 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 receiving MIDI channel to channel Nr. 1 temporarily

Solution:

- We don't know what number of MIDI channel is just active so universal 'Device ID' (**7Fh**) will be used,
- Temporary change of MIDI Channel parameter is requested, 'Address' byte is **00h**
- MIDI channel Nr. 1 is required, message will content one 'Data' byte with value **00h**

Start of SysEx Msg is then :

Start SysEx : **F0h**
Mfr ID : **00h 20h 21h**
Device ID : **7Fh**
Model ID : **56h**
Address : **00h**
Data : **00h**

- Now, checksum must be calculated as 7-bit complement of sum of bytes from 'Model ID' to 'Data', i.e.:
00h – (56h + 00h + 00h) = 2Ah

⁴ See chapter 3.3 for easy calculation of the Checksum byte.

- Form of whole required System Exclusive message is after 'Checksum' and 'End SysEx' bytes refilling:
F0h 00h 20h 21h 7Fh 56h 00h 00h 2Ah F7h

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

Note that the interface will work with new MIDI Channel only temporarily.

3.2 EXAMPLE 2

Task: Set all parameters in permanent memory to these values:

- MIDI CHANNEL : 00h
- KEY SHIFT : 24h
- KEY PRIORITY : 01h
- PITCH BEND RANGE : 18h
- ARPG CLOCK RATE : 64h

Solution:

- We don't know what number of MIDI channel is just active so universal 'Device ID' (**7Fh**) will be used,
- Permanent change of all parameters is requested, 'Address' byte is **05h**
- 'Data' bytes (5 bytes) content values of parameters given by the task: **00h 24h 01h 18h 64h**

Start of SysEx Msg is then :

Start SysEx : **F0h**
Mfr ID : **00h 20h 21h**
Device ID : **7Fh**
Model ID : **56h**
Address : **05h**
Data : **00h 24h 01h 18h 64h**

- Now, checksum must be calculated as 7-bit complement of sum of bytes from 'Model ID' to 'Data', i.e.:
00h – (56h + 05h + 00h + 24h + 01h + 18h + 64h) = 04h
- Form of whole required System Exclusive message is after 'Checksum' and 'End SysEx' bytes refilling:
F0h 00h 20h 21h 7Fh 56h 05h 00h 24h 01h 18h 64h 04h F7h

After that message is received, engaged values of parameters will be stored to permanent memory.

Note that the interface will start working with newly stored data till the interface reset.

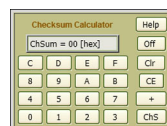
3.3 CHECKSUM CALCULATOR

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 generators to create any SysEx messages to control the interface editor. 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 Messages Generator is available at our website (<http://www.chd-el.cz>).



MIDI Interface for Roland Jupiter-4
Model JP4-KBD, Nr. 8-433, ver. 2.00
Document: 843220_syxcom_rev2

Manufacturer:
CHD Elektroservis, Czech Republic
www.chd-el.cz info@chd-el.cz

