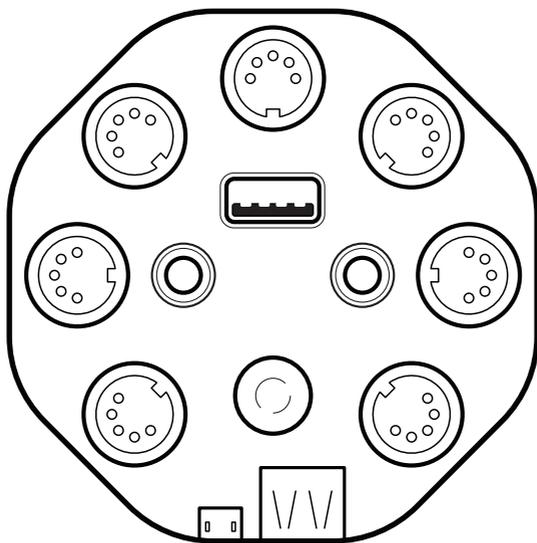


K004

MIDI MERGER/SPLITTER/CLOCK PROCESSOR



Manual v1.0 - June 2017

INTRODUCTION

This document contains the user manual of the **Retrokits RK-004 MIDI Merger/Splitter/-Clock Processor**. You will find basic user-operating instructions for the product as well as advanced MIDI implementation which allows you to shape the RK-004's operation mode.

Product summary

The RK-004 can be a central hub for various MIDI devices. In the **simplest configuration** you can use it to **distribute midi clock over 7 midi ports and the Sync out port**.

Just power the RK-004 and **tap the desired tempo** on the LED-button. After a few taps the RK-004 will have adopted the tempo and will start sending it out to its Sync and MIDI ports.

Apart from generating tempo on its own, it is also possible to **slave the RK-004 to a sync signal**. E.g. Volca's, Pocket Operators, the SyncControl app, or even older sequencers which operate on gate signals.

If most of your gear is operating in the MIDI domain, another very interesting feature is that the **MIDI ports are all 'auto sensing'**: They can all switch to MIDI-input if you just connect a MIDI device to it and send data to it. Once a port is switched to MIDI-IN, the RK-004 can not only sync from the connected MIDI device; it will **merge all MIDI data** which comes in and sends the merged MIDI data out to the remaining ports which are still in out mode.

Note: Every MIDI-Port can switch mode so you can **fully merge up to 6 MIDI devices**.

You can also set a MIDI-Port to just send out a **sync signal** if you just want to drive gate-sync gear. This may sound strange, but with the next feature of the RK-004 it can make sense:

Tempo modification

Every port can be configured to send out a modified master clock tempo. If you drive sequencers with the RK-004 you can make them run at different speeds, double, half, thirds,...

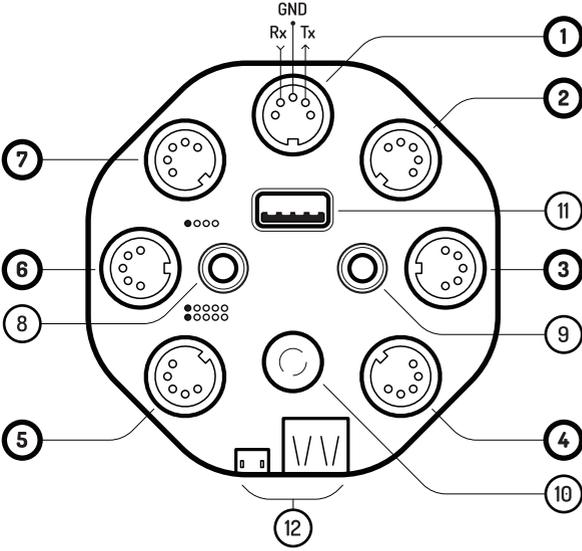
Serial I/O Port

MIDI Port 1 is special; It can be **changed into a serial I/O port**. This allows you to connect the RK-004 to an Arduino, WiFi board or RaspberryPi. The RK-004 will neatly merge the serial data you generate into the rest of the connected MIDI equipment but also send incoming MIDI through to the connected serial device.

We already have a **RaspberryPi image available** on our website which allows you to use the RK-004 and RPi as a USB MIDI Host combo but there are many other options to explore.

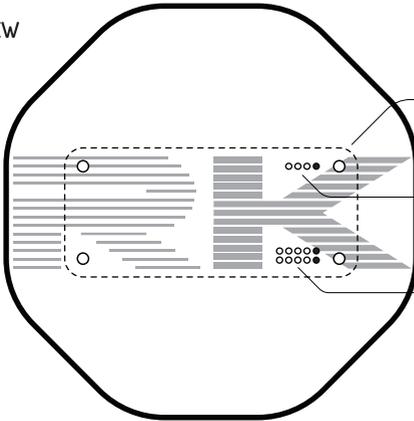
RK-004 CONNECTIONS

FRONT VIEW



- ①-⑦ MIDI/SYNC Ports
- ⑧ SYNC-IN / ACI Port
- ⑨ SYNC-OUT Port
- ⑩ LED Button
- ⑪ USB POWER OUT
- ⑫ USB POWER IN / USB DATA THROUGH

REAR VIEW

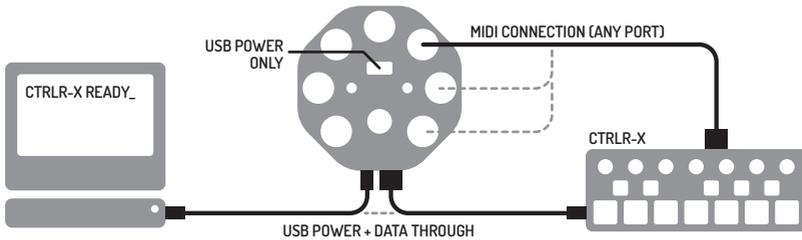


- RaspberryPi Zero outline**
Serial connection also available via DINS Port 1
- RPi Compatible pinout**
5V DC
GND
SERIAL DATA OUT [Port 1 Tx]
SERIAL DATA IN [Port 1 Rx]
- Euro rack Compatible pinout**
GATE
N/C
5V DC
N/C
GND

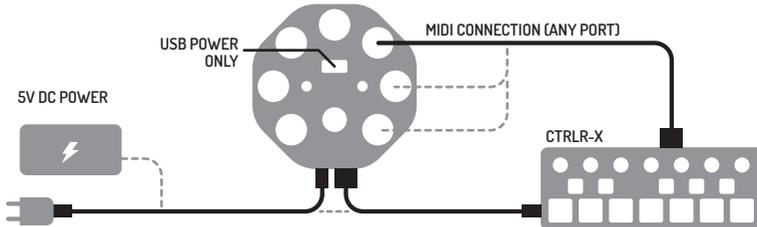
POWERING UP

The USB -ports on the RK-004 are power only. Micro USB connection ⑫ is designed to receive main 5V power. You can use a phone charger, battery pack or USB-host to power the RK-004.

If you have a USB host and you want to connect a USB client to the host you can use the RK-004 side ports to loop through the data and just 'borrow' power from the USB stream. Connect the devices like shown below and use the provided orange RK-USB cable to loop through your device:



If you have a MIDI device which also works on 5V power like the RK-004 you can also loop through the power on port ⑪ so you don't need an extra adapter to power both devices.



NOTE: The power consumption of the RK-004 is very low. In fact... it is so low that if you use a phone powerbank to power the RK-004, the majority of these banks will shut down after a number of seconds. This is because the powerbank does not detect any noticeable power consumption. To prevent this, loop the side USB through to another device or power other gear from the top USB port.

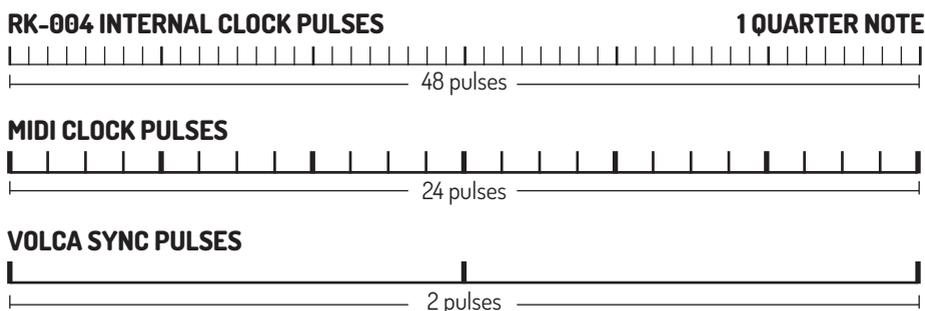
PORT 1-7 MIDI/SYNC

The MIDI Ports on the RK-004 are auto-sensing. On startup (unless default settings are overridden) the Ports are 'MIDI auto out', enabling to send clock signals to connected devices.

CLOCK SIGNAL MODIFICATION

Every Port can have it's own clock modification relative to master clock (see: **advanced configuration**). If you have multiple sequencers controlled by the RK-004's tempo, each can run relatively to the master clock. Apart from polyrhythm setups, this is also useful if you want to link older machines which run on a different clock/synchronisation speed.

RK-004s internal clock runs at 48ppqn. ppqn means: 'Pulses Per Quarter Note'. MIDI runs at 24ppqn (so half resolution of the RK-004 clock) and volca sync tempo runs at 2ppqn. The volca sync resolution is only a 12th of the regular midi clock resolution and a 24th of the RK-004s. Below you can see the resolution of the clock layed out in a quarter note:



We have introduced the 'PPSN parameter' which you can set for every port on the RK-004. It can scale the clock signal down to any value below 48ppqn. This will allow you to change clock speed between 0 (off) and 2x MIDI speed. You can find more information about the tempo modification in the '**advanced configuration**'-section.

PORT 1-7: MIDI/SYNC

SYNC MODE

Any of these Ports can also be configured to set in 'sync mode' - in combination with the clock modification you can have an intricate gearbox for your sync-based devices.

AUTO SENSING

If a MIDI device's out-port is connected to a Port on the RK-004, it will switch into a MIDI-input: It will be able to merge all incoming MIDI data and distribute it to the rest of the Ports which are still in out-mode.

Any combination of merging/splitting is possible, e.g. merge 2 MIDI streams into 5 out - or 6 streams to 1 out - and anything in between.

Note: Once a Port has received MIDI data and has switched to in-mode, it will stay that way until the power is pulled or you have pressed the tap-button +/- three seconds (=soft reset)

MIDI POWER

The RK-004 is able to MIDI-power devices like the RK-002 but not in default auto-sensing mode. Via the configuration parameters you can set the desired Port into a 'power out' mode, enabling MIDI powered devices as well (see '**advanced configuration**'-section).

LED-BUTTON

The LED button has multiple uses:

Signal indicator.

If a MIDI signal is fed into the RK-004 the button will pulse. This way you can troubleshoot if signals are coming in your MIDI chain.

Tempo generator

You can tap on the LED button to have the RK-004 generate a clock signal for all your devices. Tap the button about 4 times and it will adopt your tapped tempo.

Rescan/Hard reset

Hold the LED button for 3 seconds and the RK-004 will reset to the stored settings. Short pulses (like on startup) will indicate the RK-004 has reset. If you hold the LED button for about 10 seconds the RK-004 will reset to factory defaults. If factory reset is initiated, the LED will start glowing softly.

PORT 1: SERIAL I/O

Port 1 of the RK-004 has an extra mode: It can be set to Serial I/O mode.

If you activate this mode you will have a **Bi-directional** port which you use to interface your MIDI gear with development platforms like Arduino/Teensy or the RaspberryPi.

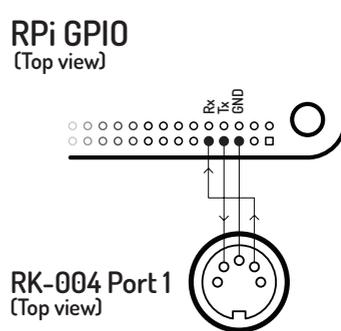
RaspberryPi

For the RaspberryPi we have an SD-card image ready at our website. The RaspberryPi Zero can be integrated directly onto the RK-004 with the pin header shown at the hardware layout in this manual. This is a modification which does require some tinkering skills.

Note: Remember not to use MIDI Port 1 if the serial I/O mode is active!

If you use our SD-card image to boot the RaspberryPi it can be used as a headless system. If you do connect a monitor you will see a MIDI monitor. With the RK-004 RaspberryPi daemon running and connected you will be able to use and merge USB-MIDI devices into the MIDI stream of the RK-004.

It is also possible to use other RaspberryPi systems with the RK-004 by connecting the serial pins of the RPi to the RK-004's first MIDI port like shown on the right:



You can add our repository to your RaspberryPi system and use the RK-004 as a MIDI interface with this command:

```
curl -sLS https://www.retrokits.com/debian/add.sh | sudo bash
```

And install our RK-004 Daemon like this:

```
sudo apt-get install rk004d
```

On reboot, the RK-004 will show up as a MIDI device.

For more information about the RaspberryPi, check out <https://www.raspberrypi.org>.

PORT 8: SYNC IN /ACI

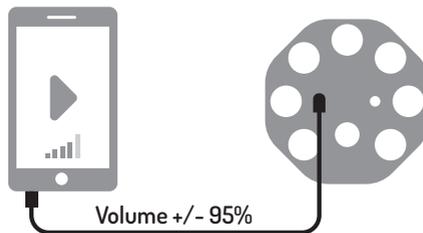
The Sync/ACI port can be used to generate a MIDI clock signal from a sync source. This can be e.g. a volca sync-out signal or an app like Synckontrol.

ACI

The RK-004 is completely upgradeable and configurable via MIDI System Exclusive messages (see 'configuration'-section). This is a proven method to manage MIDI devices. But because we wanted to have a configuration fallback in a completely DAWless environment we added an Audio Control Interface on the Sync-In port. This ACI allows you to send RK-004 configuration or MIDI signals through via a plain audio signal (you can use MP3s as a datafile)

Although trivial – via a simple audio signal you can set the Port behaviour or even update the RK-004 firmware.

Note: We also have the RK-002 firmware available as MP3 files to update via the RK-004s ACI port, enabling you to change the RK-002's firmware without the use of a computer.



We have a web-app on our site and on Android/iOS app store which also uses the audio connection for configuration.

Both sync and update can be done via an audio connection between the sender and the RK-004. For now we will assume a mobile phone.

1. Connect a cable between the audio out of the phone and the RK-004's ACI Port.
2. Set the volume about 95%
3. Play the audio file or use our web-app.

PORT 8: SYNC IN /ACI

If you perform an update of the RK-004 via ACI audio. The RK-004 will display a slow glowing signal on the LED. If all goes well the glowing will stop after the end of the file and the RK-004 will reboot with three flashes (the same signal as on startup)

If the signal is not received correctly, the RK-004 LED will flash fast. If this happens, adjust the volume and try playing the audio file again from the beginning. The RK-004 will not be useable until after you have completed the update correctly. It is not bricked, you just need to redo the update process.

NOTE: the ACI control is a fallback option which can be used if you don't have any other MIDI devices available to do configuration. We would recommend MIDI + Sysex control of the RK-004, you don't need to tune the audio level and the update is quite a bit faster.

PORT 9: SYNC OUT

By default, the Sync out Port is set to volca mode. This means a timebase of 2ppqn and a fairly short pulse to reduce sync skipping on the volca's. Via our configuration app you can set the properties of this port to a different timebase, inverse pulse level or lengthen the pulse for use with other gate-sync based devices like the Teenage Engineering's Pocket Operator series.

To see all available options for the Sync out Port, please go and check out the '**advanced configuration**'-section.

Sync/Clock priorities

Since all ports can be MIDI-In ports as well there needs to be a priority of what the RK-004 does with incoming (MIDI-)clock signals. Priority is given as follows.

- ① A clock signal on the sync in port overrides all other clock sources.
- ② MIDI clock priority is given to the port with the lowest number, so : Port 1 has priority over port 2, port 2 over three, etc - port 7 will be the least important clock.

ADVANCED CONFIGURATION

If you just want to use the RK-004 as a basic MIDI-merger or -splitter, the information you find hereafter might not be that important. The RK-004 will adapt to the signal you put in and is easy to reset (hold the LED button for 3 seconds) in case you want to start over building your MIDI setup.

However, by making use of the advanced configuration of the RK-004 you will be able to play a bit more with your MIDI setup...

ADVANCED CONFIGURATION

The best way to explain all configuration options is by going through the **MIDI System exclusive** parameters which the RK-004 accepts. This way you will see the possibilities and have practical examples.

What is MIDI System Exclusive?

MIDI is a serial connection between instruments. It has a set of commands which are designed to trigger notes, change synth properties with controllers, synchronise tempo, etc. These are all pretty generic commands to make sound.

However, if you want to change specific settings in a synthesizer/module, manufacturers have the possibility to expand within a special block called System Exclusive. Within this block bigger chunks of data can be sent back and forth and checked on correct data interchange.

System Exclusive, 'Sysex' in short is what the RK-004 uses to do firmware upgrades and advanced settings.

RK-004 SysEx Format:



Sysex always starts with F0 and ends with F7. The Retrokits ID is an identifier which prevents the message to be interpreted by other MIDI gear. It has to be included on every Sysex message.

NOTE: As defined by MIDI sysex, <args> are 7-bit packed: 7 bytes of 7-bit data is prequelled by a byte containing the MSBs. Please refer to www.midi.org for more detailed information.

The 'command' section is what we will layout in the next pages, each command you send to it will be acknowledged with a response from the 004.

Example:

SETPARAM_REQ: F0 00 21 23 00 04 03 <paramnr> <paramval> F7

will return a:

SETPARAM_RSP: F0 00 21 23 00 04 43 <paramnr> <paramval> F7

Notice the 03 and 43 to indicate what command we're working with (send/response)

ADVANCED CONFIGURATION

Here the full list of RK-004 commands:

SETPARAM_REQ:

F0 00 21 23 00 04 03 <param> <val> F7

Sets parameter number <param> to value <val>*

SETPARAM_RSP:

F0 00 21 23 00 04 43 <param> <val> F7

Returns value <val> from parameter number <param>

GETPARAM_REQ:

F0 00 21 23 00 04 04 <param> F7

Retrieve a value from parameter number <param>

GETPARAM_RSP:

F0 00 21 23 00 04 43 <param> <value> F7

FACTORY_RESET_REQ:

F0 00 21 23 00 04 05 F7

Perform a RK-004 factory reset

FACTORY_RESET_RSP:

F0 00 21 23 00 04 45 F7

COMMIT_PARAMS_REQ:

F0 00 21 23 00 04 07 F7

Saves all current parameter settings to RK-004 startup configuration

COMMIT_PARAMS_RSP:

F0 00 21 23 00 04 47 F7

***NOTE:** Set parameters will instantly become active but not saved to the RK-004 startup configuration. You need to invoke a **COMMIT_PARAMS_REQ** to save your settings.

ADVANCED CONFIGURATION

RK-004 Parameter numbers:

NR (HEX)	NAME	VALUES		
00	0x00	SYNCIN_MODE	not used	
01	0x01	SYNCIN_PPSN	Sync-In PPSN (see MIDI-Out PPSN reference*)	
04	0x04	SYNCOUT_MODE	0	Positive Pulse, short
			1	Negative Pulse, short
			2	Positive Pulse, long
			3	Negative Pulse, long
			>=64	Battsynth-out mode
05	0x05	SYNCOUT_PPSN	Sync-Out PPSN (see MIDI-Out PPSN reference*)	
08	0x08	DINI_MODE	0	Sync-Out mode
				Positive Pulse, short
			1	Sync-Out mode
				Negative Pulse, short
			2	Sync-Out mode
				Positive Pulse, long
			3	Sync-Out mode
				Negative Pulse, long
			4	MIDI Auto detect mode
			5	MIDI-In mode
			6	MIDI-Out mode (MIDI power)
			Following modes are DIN1 only:	
			7	UART I/O Mode [38400Kbps]
			8	UART I/O Mode [31250Kbps]

See **UART Flow Control** note

ADVANCED CONFIGURATION

RK-004 Parameter numbers (continued):

NR (HEX)	NAME	VALUES
----------	------	--------

09	0x09	DIN2_MODE
----	------	-----------

10	0x0A	DIN3_MODE
----	------	-----------

11	0x0B	DIN4_MODE
----	------	-----------

12	0x0C	DIN5_MODE
----	------	-----------

13	0x0D	DIN6_MODE
----	------	-----------

14	0x0E	DIN7_MODE
----	------	-----------

DIN2 and subsequent port modes can be set the same values as DIN1 modes, with exception of the UART modes.

16	0x10	DIN1_PPSN
----	------	-----------

MIDI-Out PPSN

PPSN reference:

17	0x11	DIN2_PPSN
----	------	-----------

The Sync and MIDI timebase property is set up in PPSN; 'Pulse Per Sixteenth Note'.

18	0x12	DIN3_PPSN
----	------	-----------

The RK-004 has an internal clock with runs at twice midi clock speed: 48 PPQN, the PPSN parameter is 4 times PPQN so: **48PPQN = 196 PPSN** (48*4).

19	0x13	DIN4_PPSN
----	------	-----------

20	0x14	DIN5_PPSN
----	------	-----------

To recalculate 48PPQN to 2 PPQN (volca sync timebase) we need to divide 48 by (48/2) - 24, next we convert it to PPSN (*4) which gives **96**

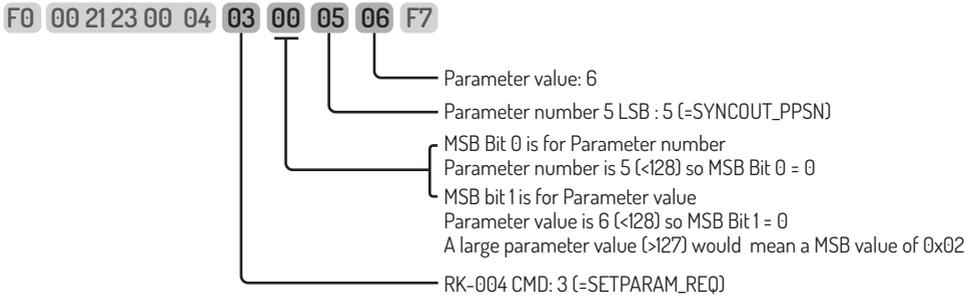
21	0x15	DIN6_PPSN
----	------	-----------

22	0x16	DIN7_PPSN
----	------	-----------

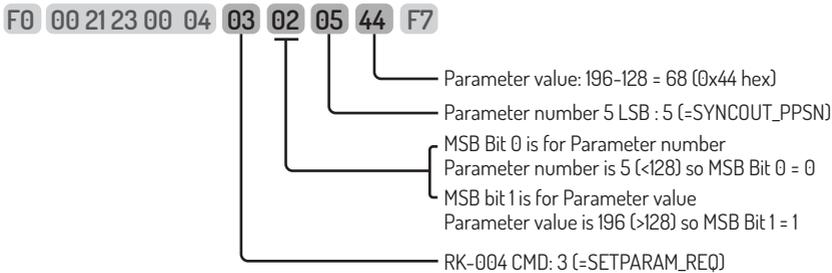
ADVANCED CONFIGURATION

Example Sysex

Below an example syntax message to set **SYNCOU_T_PPSN** to 6 (<paramnr>=5, <paramval>=6)



Setting **SYNCOU_T_PPSN** to value **196** would result in the following:



UART Flow Control

Especially when operating the RK-004 on the full-duplex uart port running at 38400bps, there is a realistic chance of overflowing the connection. Therefore the RK-004 emits XON/XOFF messages in order to flow-control the connection:

XON = 0xF9

XOFF = 0xFD

Both messages are 'undefined' in MIDI, and therefore are to be used solely for in-band flow-control.

PRODUCT SUMMARY

Company: Retrokits, www.retrokits.com
P.O. Box 36334
1050 MH Amsterdam, the Netherlands

Product Name: RK-004 MIDI Merger/Splitter/Clock Processor ('Medusa')
Web settings manager: <https://www.retrokits.com/rk004/settings/>

Product Development: A.J. Huitsing, G. Dijkstra

