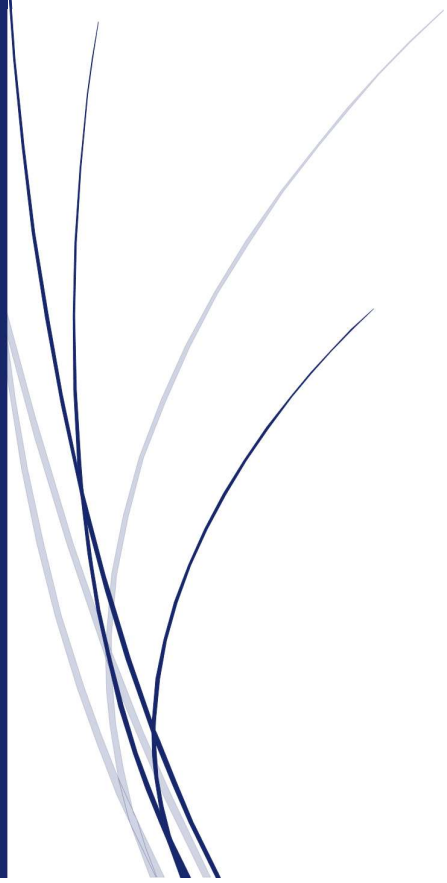
5/2/2023

Ge0sequencer

TB-Seq & Desktop Synths



V1.5.0b

GEOSYNCHRONOUS SYNTHS

Table of Contents

Overview *3*

MIDI-2-CV & Sequencer Module **3**

Feature Summary **3**

Connections *4*

PPQN Definition 4

TB-Seq Connections **4**

CV Outputs 5

Desktop Connections **5**

Device & Sequencer *6*

Sequence Slots **6**

Home Page **6**

Modifying Settings from the Device **7**

Settings Menu 8

Scale Factor 9

Factory Reset 10

About 10

Manipulating Sequences from the Device **10**

Activate Seq 11

Clear Seq 11

Save Active Seq 11

Edit Active Seq 12

Using the Ge0sync Sequence Editor Application *13*

Installing the Ge0sync Sequence Editor **13**

Application Sections 14

Application Settings (1) **15**

Testing Communication 15

Global Device Settings (2) **16**

Device In MIDI Channel 16

MIDI Note Priority 16

MIDI CC to send to Mod Output 16

CV Clock PPQN Input 17

CV Clock PPQN Output 17

Delay after Gate Off 17

Mod Output CV Duration 18

Trigger Output CV Duration 18

Clock Output CV Duration 18

Sending Settings to the Device **19**

Creating, Sending & Activating Sequences *20*

Sequence Details (3) **20**

Sequence Slide Type 20
Sequence Steps (1-16) 22
Device Sequence Tasks (4) 24
Send Sequence to Device 24
Activate existing Sequence 24
Local File Management (5) 25
Updating Device Firmware 26
Uploading Using an Arduino Uno 26
Upload Using avrdude 27

Overview

MIDI-2-CV & Sequencer Module

The TB-Seq eurorack module and integrated MIDI interface in the Ge0synchronous desktop semi-modular synthesisers combine standard MIDI note to CV V/Oct, gate and trigger conversion as well as MIDI to CV clock with configurable PPQN (pulse per quarter note) – both in and out.

The module also provides a basic “TB” style 1 to 16 step, sixteenth note, sequencer with eight persistent memory slots.

Partnering the module itself is a Windows or Mac based application that can be used to create, save and upload sequences to the module memory slots. The application also allows device settings to be sent to the module. Data transfer between the host computer and module is by means of MIDI SysEx messages via the 5-pin DIN MIDI Input.

Feature Summary

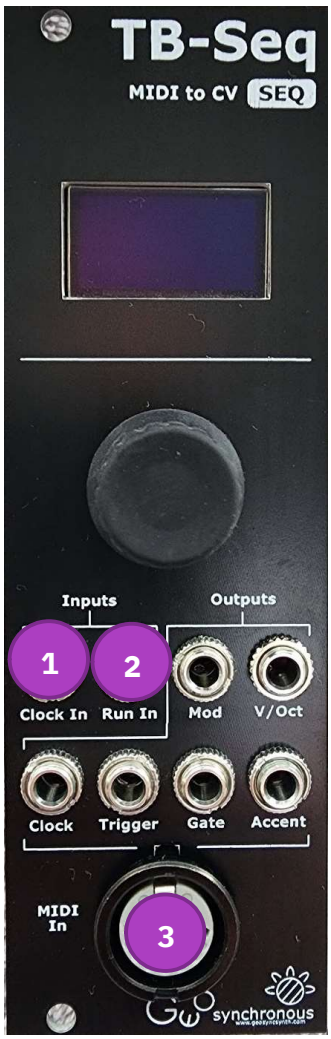
- 5 pin DIN MIDI Input.
- CV Input for CV Clock input pulses and CV Run to start sequencer. (Incoming clock configurable PPQN from 1 to 24)
- CV Output for MIDI to CV Clock (configurable PPQN from 4 to 24)
- CV Output for gate, trigger and accent. Gate and trigger correspond to incoming MIDI note or sequencer steps.
- CV Output for modulation – value set via configurable MIDI CC.
- 8x 1 to 16 step sequencer slots
- Integrated display and menu to configure device settings, and activate, copy or clear sequencer slots.
- Standalone PC based application to create, save and upload sequences and device settings via MIDI SysEx messages.

Connections

PPQN Definition

Most instrument clocks are defined in terms of Pulse Per Quarter Note. A quarter note usually equates to a “beat” in a standard 4/4 pattern. So for example, MIDI by default sends 24 PPQN, which means in a standard TR style 4/4 step sequencer this sends 6 pulses for each step – or in other words, 6 pulses per 1/16th note.

In terms of the Ge0Seq sequencer, each step is 1/16th note, and so the full 16 steps in a sequencer are a single note. With 4 steps per quarter note.



TB-Seq Connections

1 **Clock In** - A CV signal that sends a pulse on a regular interval. The number of pulses that are received before the sequencer moves on to the next step is determined by the **CV Clock In – PPQN** setting.

It maybe easier to understand this in terms of ***ticks per step***.

For example, with a PPQN in setting of **4**, the next step in the sequence will be played every time a pulse is detected on this input. The ***ticks per step*** being **1.**

With a setting of **16**, the next step will be played after 4 pulses have been received. The ***ticks per step*** here is **4**.

2 **Run In** – A trigger to “start” the sequencer. A short pulse on this input will start the sequencer playing and it will advanced through the steps based on the **Clock In** pulses and the **PPQN In** setting as described above.

3 **MIDI In** – Connect this port to your MIDI controller / MIDI interface or MIDI keyboard. This port is used to receive MIDI clock, and any start/stop signals sent via MIDI will result in the sequencer starting/stopping. This port is also used to received MIDI SysEx messages from the **Ge0sync Sequence Editor** application.

CV Outputs

The remaining ports on the module are all outputs.

- **V/Oct** This provides a CV note output. This will either send the note as played on your connected MIDI device, or will send the appropriate voltage as defined by the current sequencer step.
- **Mod** This sends a CV voltage, between 0 and 10v that corresponds to the incoming MIDI CC value for the setting defined by **MIDI CC Mod CV** – As MIDI CC sends a value between 0 and 127, the value will be scaled to spread across the 10v range. For example, if a value of 64 is send, the jack will output approximately 5v.
- **Clock** This sends a pulse that can be used as a CV clock. The number of pulses sent is dictated by the **CV Clock Out PPQN** setting. In the same way as the CV Input clock, the number of pulses that are sent per quarter note can be configured, but must be a logical division of 24 (MIDI PPQN) This output is only active when there is an incoming MIDI clock signal.
- **Trigger** This sends a pulse for a period of time (default 20ms – configurable in the settings) when a note is triggered, either by the attached MIDI keyboard, or the step sequencer.
- **Gate** This send a pulse that is maintained for the length of the note. Again, this gate is sent either while a key is pressed on the attached MIDI keyboard, or the step sequencer is playing a note.
- **Accent** A pulse is sent when the sequencer step has the Accent value enabled for this step. Like gate, this is sent for the duration of the note. The accent output is only enabled when the sequencer is playing.

Desktop Connections

<TBD>

Device & Sequencer

Sequence Slots

The device comes with a default sample sequence in slot “D” (Default). The other eight sequence slots are blank and must be “sent” a sequence via the companion Ge0sync Sequence Editor application.

The eight sequence slots can be used to store multiple sequences for later “activation”. Only the currently activated sequence is played when the sequencer is started by either MIDI or CV.

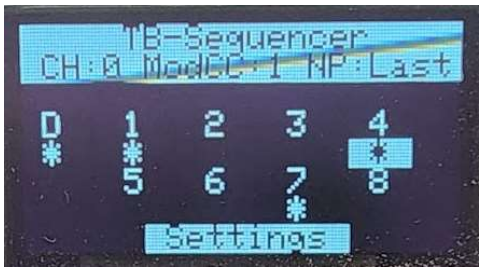
Home Page

When you power on the device, after the initial splash screen, the device will load the settings from the internal EEPROM. If this is the first time the device has been powered on, or if you have updated the firmware and erased the EEPROM, the default settings will be applied and the device will reboot.

When the device is ready for use, it will show the the “home page”. This shows some basic MIDI configuration details in the header and a list of the slots showing which ones have sequences stored and the currently activated sequence.

Once sequences have been loaded into slots, the slot will show an “*” under the number. The currently activated sequence is denoted by a highlighted box around the corresponding sequence “*”

The following example shows sequences in slots D, 1, 4 and 7. With slot 4 being the currently activated sequence.



Modifying Settings from the Device

From the home page, click or turn the encoder to access the Settings menu.

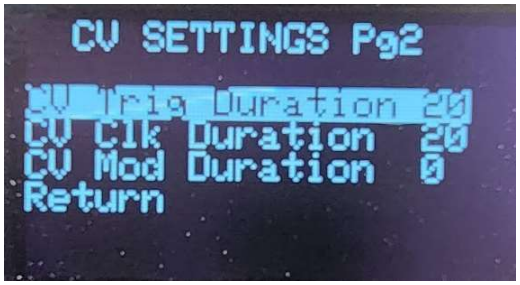
When any settings menu is displayed, the device will return to the home page is the encoder is not touched for 20 seconds.

Any menu with a “Return” option will take you back one level in the menu system. From the main Settings menu this will take you back to the home page.

Settings Menu

From here you can access the 3 sub menus :

- **CV Settings** Allows you to modify any of the CV settings, these are described in detail in section “



- *Global Device Settings (2)" on page 16*. The CV settings also has a second page of settings available by selecting ***More CV Settings***

- **MIDI Settings** Allows you to modify any of the MIDI settings and are also described in the same section above.

- **Sequence Settings** Allows you to active, copy or clear sequence slots as described below.

Scale Factor

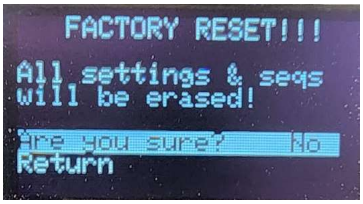
One setting that is not available in the Ge0sync Sequencer application is the **CV adjust mV** scale factor, this is only configurable on the device itself.

This allows you to adjust the "width" of note by a scaling factor. The device uses a digital to analog converter to convert the note value into a V/Oct value to be output on the V/Oct CV jack. This should be accurate by default, however depending on the quality of your Eurorack power supply you may need to adjust the scaling by using this setting.

The setting allows scaling between 0.9 and 1.1 in increments of 1/1000th of a percent. In general you should not need to adjust this setting from the default 1.000 value.

Factory Reset

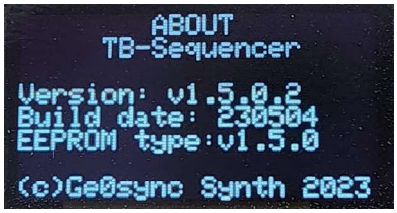
Should you wish to erase the contents of the EEPROM and return the device to the factory default un-initialised state, select this menu option and confirm your choice on the subsequent page.



WARNING! This will erase all sequences and return the device settings to the default values. The device will reboot twice during this process.

About

This menu item shows some basic information about the device, the firmware version, the date of build and the EEPROM format version. This can be used to confirm that the device and Ge0sync Sequence Editor application are using compatible versions.



Manipulating Sequences from the Device

The home page shows a single **Settings** button at the bottom.

To access the settings menu, either click (push) or turn the encoder.

To access the sequence sub-menu, scroll to and select **Sequence Settings**

From here you can :

- **Activate Seq** – select a slot to activate
- **Clear Seq** – select a slot to clear
- **Save Active Seq** – select a slot to save the currently active sequence into
- ***Edit Active Seq*** – not yet supported
- **Return** – go back to the main menu

To perform one of the actions, scroll to the menu item and click the encoder button. The encoder can then be used to scroll through the available sequence slots to select which slot to manipulate.

Activate Seq

This simply copies the sequence stored in the selected slot into the active memory on the device. This sequence will now play when MIDI start is received or when the CV RUN and CV CLOCK IN signals are received.

If you attempt to activate a slot that is empty, a suitable error is shown.

Clear Seq

No confirmation! The selected sequence slot will be cleared. If you select this option by accident, wait for 20 seconds and the menu will timeout and return you to the home page.

Save Active Seq

This option can be used to copy a sequence from one slot to another. Whatever the contents of the in memory active sequence, it will be saved to the selected slot.

For example, you want to duplicate the sequence from slot 1 to slot 2.

1. First activate slot 1 to copy its sequence into the active memory.

2. Save Active Seq to slot 2, this will write the active sequence in memory to slot 2.

Edit Active Seq

Not yet supported. A suitable error message is shown.

This will be added in a subsequent release, allowing step by step editing of the active slot.

Using the Ge0sync Sequence Editor Application

Installing the Ge0sync Sequence Editor

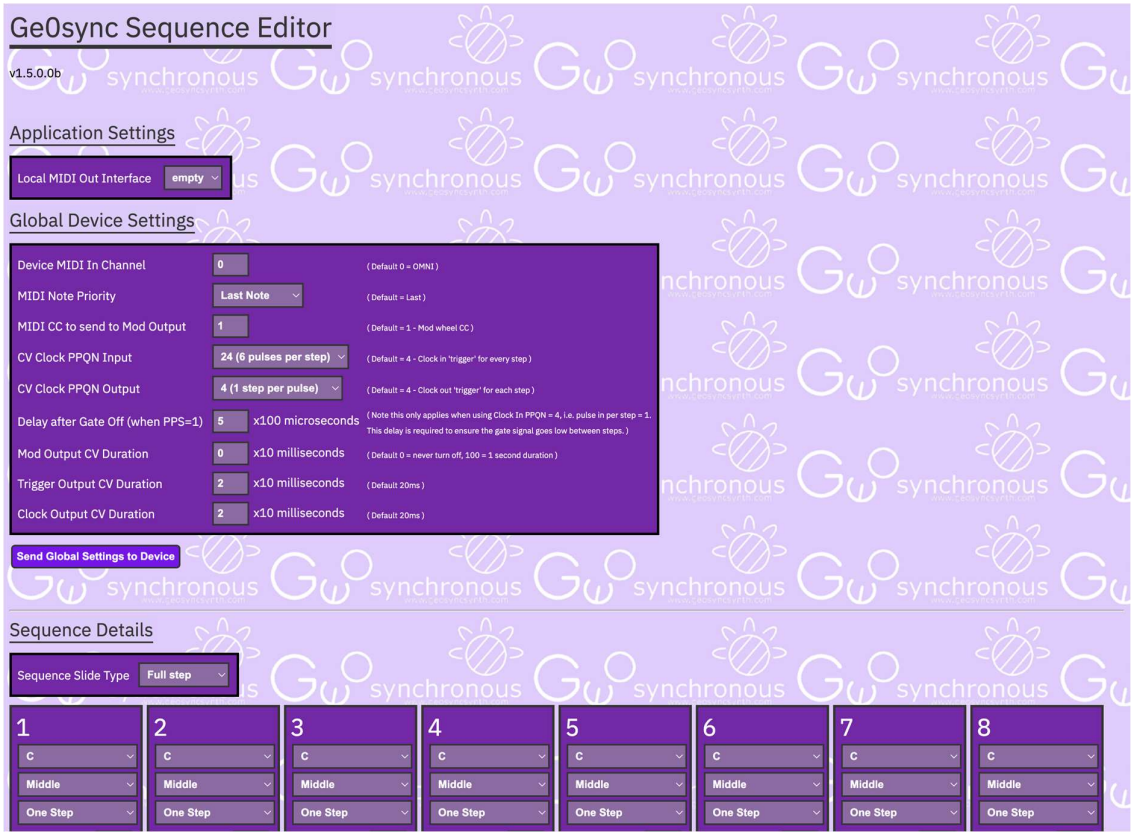
New sequences must be created using the Ge0sync Sequence Editor application running on either a Windows (7, 10, or 11) PC or a Mac (Intel or Apple Mx CPU).

Download, install and run the correct version of the application for your PC. The latest version of the application can be downloaded from :

https://geosyncsynth.com/documentation/

Note: You should download the version of the editor that matches the firmware running on your device. The current firmware version is displayed when the device boots.

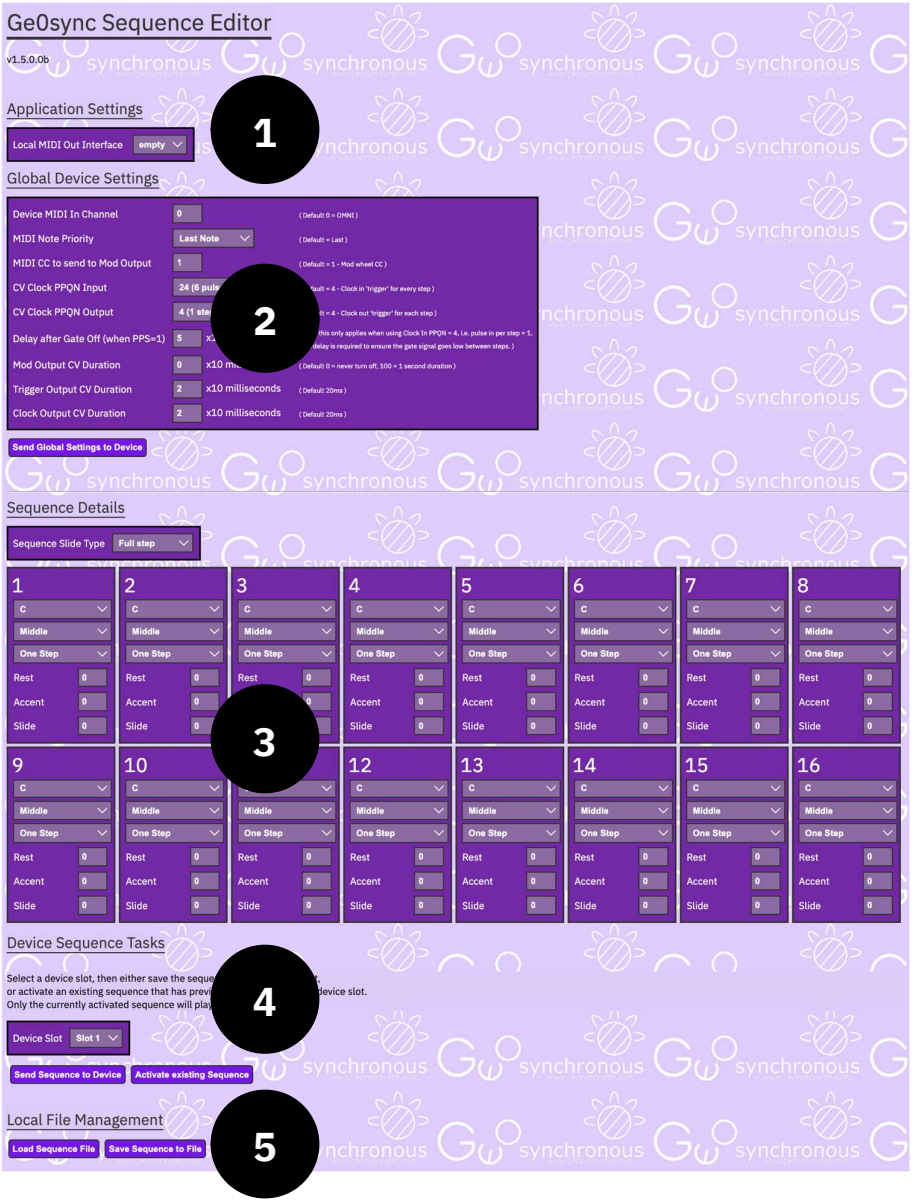
To update the firmware on your device, see the **Device Firmware** chapter.



Application Sections

The application is split into 5 sections :

1. Application Settings
2. Global Device Settings
3. Sequence Details
4. Device Sequence Tasks
5. Local File Management



Application Settings (1)

This section allows you to configure the application settings, specifically which local MIDI interface will be used to communicate with the device.

Testing Communication

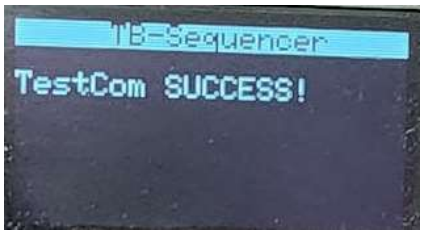
To update the device, for example to send a sequence to the device, you must have a MIDI connection between your PC and the device. Either connect your sound card/audio interface (with MIDI DIN connectivity) directly to the module MIDI IN DIN port, or via a MIDI OUT/thru chain.

You can also use a USB MIDI capable device, for example a keyboard or MIDI controller that connects to your PC via USB, but then provides a MIDI OUT/thru 5-pin DIN connection.

Once connected, you should see the PC based MIDI interface listed in the drop down in the **Application Settings -> Local MIDI Out Interface**

Select the interface you want to use and confirm communication by clicking the **Test Comms** button. This should show success on the GUI and at the same time a success message will be displayed on the device display.

This test also validates that the version of the application is compatible with the firmware on the device, if the versions differ sufficiently an error will be shown on the device. Update the application or device firmware accordingly.



Global Device Settings (2)

This section allows you to modify device behavior settings. These settings are also configurable on the device itself – using the **Settings Menu** as described in *"Modifying Settings from the Device"* on page 7.

Device In MIDI Channel

This option sets the MIDI channel on the device. The device will respond to MIDI notes sent on this channel. Note that MIDI start/stop and SysEx are global, so non channel specific. By default, the device is set to OMNI mode, therefore will respond to MIDI notes sent on any channel.

To restrict the device to a specific channel, select a channel between 1 and 15.

To set the device back to OMNI mode, select 0.

MIDI Note Priority

Sets the note priority when more than one MIDI note is played at the same time.

The available options are **Last, Top** and **Bottom.** The default is Last.

- Last sets last not priority, the last note received will be played.
- Top sets top note priority, the highest note received will be played.
- Bottom sets bottom note priority, the lowest note received will be played.

MIDI CC to send to Mod Output

The device provides a Modulation Output CV jack. This output sends values between 0 and 10v as CV signals.

This setting allows you to select which MIDI CC controller value is used to set the modulation output voltage. The default is CC 1 which corresponds to the MIDI modulation wheel on your MIDI keyboard. To change the CC value, select one of the values from 1 to 127.

MIDI CC values have a range between 1 and 127. The device maps these to the voltages between 0 and 10v. For example, the midpoint in your modulation wheel will correspond to MIDI CC value 64, which will send a 5v signal to the Mod Output jack.

The duration of the Mod Out CV voltage "pulse" or signal can be modified via the **Mod Output CV Duration** setting.

CV Clock PPQN Input

The sequencer can be triggered and clocked via CV inputs. When the device receives a pulse (>5v) signal on the CV RUN input jack, the device will start playing via the incoming CV clock pulses. The number of pulses that correspond to a quarter note (4 steps) can be modified with this setting.

For example, MIDI uses 24 PPQN (therefore sends 24 pulses per quarter note) which means each sequencer step corresponds to 6 pulses (6 pulses per step)

This setting lets you set the number of PPQN when using the CV clock In. Supported values are

- 24 PPQN – 6 pulses per step
- 20 PPQN – 5 pulses per step
- 16 PPQN – 4 pulses per step
- 12 PPQN – 3 pulses per step
- 8 PPQN – 2 pulses per step
- 4 PPQN – 1 pulse per step

Note: 4PPQN, or 1 pulse per step is useful to allow a trigger input to advance the sequencer every time a pulse is received. Note also that because there are no 'sub-divisions' to this setting, some special handling of the Gate Output CV jack is needed with the **Delay after Gate Off** setting.

CV Clock PPQN Output

The device allows the conversion of the incoming MIDI clock to send a CV clock output. MIDI clock is fixed at 24 PPQN, and so this setting allows you to send the same number of pulses into your CV modules or to sub-divide the clock to less PPQN values.

- 24 PPQN – every MIDI pulse results in a CV pulse – 6 pulses per step
- 12 PPQN – every second MIDI pulse sends a CV pulse – 3 pulses per step
- 8 PPQN – every 3rd MIDI pulse sends a CV pulse – 2 pulses per step
- 4 PPQN – every 6th MIDI pulse sends a CV pulse – 1 pulse per step
- 1 PPQN – every 24th MIDI pulse sends a CV pulse – *4 steps per pulse*.

Delay after Gate Off

Note: Only used when ***CV Clock PPQN Input = 4*** *(i.e. 1 pulse per step)*

Because we only have 1 pulse per step incoming to the CV clock Input, we need to turn the CV Gate output off between steps to allow envelopes to re-trigger, etc. If we simply turn

the gate low, and then one high again, most euro modules don't notice this change because the digital processing of the device is so fast.

This setting specifies how long to delay after the gate has gone low before turning it on again. The value is set on the device to correspond with microseconds of delay. The default value is 500 microseconds.

Because we are using MIDI SysEx to send data to the device, the application sends a value between 1 and 10 and the device multiples this by 100 to get a delay value between 100us and 1ms (1000us).

Note: Settings as low as 100us may not be long enough of a delay for your eurorack module to detect the gate off.

Mod Output CV Duration

The modulation CV output jack can be configured to send a CV value for a specific period of time.

When an incoming MIDI CC value is received, the value is translated into a voltage between 0 and 10v and this voltage it sent on the Mod Output Jack for the duration specified by this setting.

The default setting is 0. This means to continue to send the voltage indefinitely, or until a new value is set. Otherwise, the value of this setting is multiplied by 10 inside the device and the Mod Output jack will send the value for that number of milliseconds. For example, a setting of 2 will result in the jack outputting its voltage for 20ms.

Trigger Output CV Duration

When a MIDI note is received or the sequencer sends a CV output to the V/Oct output jack, the trigger output is set high (>5v). The duration that this jack sends the high voltage is controlled by this setting. The device multiplies the value by 10 and will send the trigger output for that number of milliseconds. For example, the default setting of 2 corresponds to 20ms.

Clock Output CV Duration

A clock pulse is send on the CV Clock Out jack based on the required Clock Out PPQN setting. The duration that this jack sends the high voltage is controlled by this setting. The device multiplies the value by 10 and will send the clock output for that number of milliseconds. For example, the default setting of 2 corresponds to 20ms.

Sending Settings to the Device

To send all the changes you have made to the device settings in the application, click the **Send Global Settings to Device.**

Only settings that are different to those stored on the device will be updated.

Note: If you change the MIDI In Channel, the device will update all the other settings first, then reboot the device to restart the internal MIDI interface to pick-up the new MIDI channel.

Creating, Sending & Activating Sequences

Sequence Details (3)

The sequence details section defines the parameters for each of the 16 steps in your sequence.

The sequencer is loosely based on the classic TB-303 in terms of the settings per step. However, it follows a more logical, modern step-based programming style where the notes and duration area not input independently!

Sequence Slide Type

One of the classic 303 style parameters is the slide.

The Ge0Seq sequencer adds further configuration options to the slide. This lets you decide what percentage of the first node plays before it starts to incrementally slide up or down to the next note.

All steps in a sequence will use the same slide type.

How does it work?

A step can consist of multiple clock ticks – the *ticks per step* – which is calculated based on the MIDI clock in, or CV Clock in PPQN.

If there is more than one tick per step, then the slide can transition between the start note and end note in equal increments. The number of increments is dictated by the number of ticks per step, and the slide type setting.

For example, with the MIDI clock controlling the sequence, there are 6 *ticks per step*. Therefore, if the **slide type** is set to ***full step***, then the slide will transition between the start note and end note in 6 equal increments. If the slide type is set to ***half step***, then 3 increments will be used, and these will start in the *second half* of the note.

The device will attempt to honour the slide type as best it can, however some combinations are impossible to achieve and the best effort will be made to perform a slide. For example, if the device is configured in such a way that there ends up with only 1 *tick per step*, there are no ‘sub-steps’ and no increments can be calculated.

If this is the case, start note will play, followed by the end note. A fast slide will be heard because the gate is not turned off between sliding notes.

The slide type can be

- **Full step** – the slide will start at the beginning of the step and slide up/down in equal increments (where possible)
- **Half step** – the slide will start half-way through the step and slide/up down in equal increments (where possible)
- **Quarter step** – the slide will start between two-thirds and a three-quarters of the way through the step (where possible)

The following table shows the possible combinations of incoming PPQN and what the device will do based on the slide type setting.

PPQN	**4**	**8**	**12**	**16**	**20**	**24 (MIDI)**
Tick per Step	1	2	3	4	5	6
MIDI	No					Yes
CV	Yes					
Slides over last x ticks of the step						
Full	No	2	3	4	5	6
Half	No			2	2	3
Quarter	No					2

Sequence Steps (1-16)

Each step has the following attributes.

Note

The note to play, from C to B with a special value '*STOP*'.

STOP: The special note value of STOP is used to shoten the number of steps in a sequence.

For example, if you want a 4 step sequence, set step 5 to STOP and steps 5 through 16 will be hidden in the application.

Octave

The octave offset to apply. Options are

- Down 2
- Down
- Middle
- Up
- Up2

Setting Note: ***C*** and Octave: ***Down 2*** corresponds to MIDI node C1.

Step Length

The number of steps this note should be played. Options are 1, 2 or 4.

- **1** plays a 1/16th note
- **2** plays a 1/8th note
- **4** plays a ¼ note

For example, here Step 1 is 4 steps long, and step 5 is two steps long. The application hides the appropriate steps.

Rest, Accent & Slide

These settings can be enabled (1) or disabled (0) per step.

- **Rest** - turns the step off and no note is played.
- **Accent** - sends a pulse for the duration of the step on the CV Accent Output jack.
- **Slide** - causes this note to slide up or down to the next note. See the associated slide type description on page 20 for more details.

Device Sequence Tasks (4)

Select the device slot that you wish to work with from the **Device Slot** drop down menu and then select one of the tasks.

Send Sequence to Device

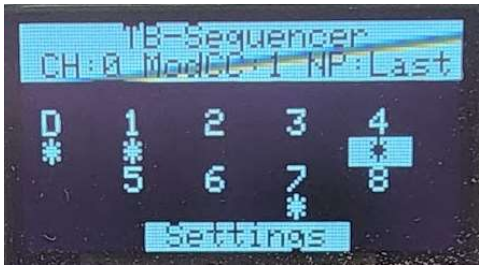
To send the sequence you have programmed in the application to the device, choose a slot location from the drop down and click the **Send Sequence to Device** button in the **Device Sequence Tasks** section.

WARNING: If an existing sequence is saved in the device slot it will be overwritten without any additional warning or confirmation. Be sure to select the correct slot before sending a sequence to the device.

The device display will show it is receiving incoming SysEx data, followed by a message to show it is saving the sequence data to the chosen device slot.

Note: When a sequence is saved to the device it will automatically become the active sequence.

When the device returns to the home page the slot should now show that it has a saved sequence "*" and that it is the active sequence (highlighted "*") as shown here for slot 4.



Activate existing Sequence

To activate an existing sequence that has been previously saved on the device, select the slot you wish to activate from the **Device Slot** drop down menu and click **Activate existing Sequence.**

The device display will show it is receiving incoming SysEx data, followed by a message to show it is activating the sequence data to the chosen device slot.

If there is no sequence stored in that device slot the device will show a suitable error message and the default sequence will become the active sequence.

Local File Management (5)

You can load and save sequence information to the local storage device on your PC for future use. This allows you to build a library of sequences and not be limited by the eight slots on the device.

To save the current sequence data to a file, select **Save Sequence to File.**

To load a sequence file into the application, select **Load Sequence File.**

Note these tasks only save or load the sequence from/to the application. If you want to send a saved sequence to the device you will need to

1. Load the sequence from a local file.
2. Send the sequence to the device.

Note: It is recommended you always save your sequences to the local filesystem so you can later re-upload them to the device if the slot memory has been cleared – for example after firmware updates.

Updating Device Firmware

Future updates may provide firmware update over MIDI SysEx connections, but at the time of writing updates must be performed via the ISP header on the device.

The device provides an ISP device header to upload firmware updates. You will need either an Arduino Uno or a serial in-circuit programmer such as a USBASP.

Download and install either the Arduino IDE, or avrdude.

WARNING: Uploading the firmware will erase the EEPROM contents and essentially perform a factory reset of the device. Make sure you have saved any sequences to the local filesystem on your PC before updating firmware.

WARNING: Uploading firmware can potentially brick the device, this can always be recovered, but may require removing re-flashing the bootloader and CPU fuse settings. Proceed with caution!

Uploading Using an Arduino Uno

To upload using an Arduino Uno you will need to download the source code sketch from github : https://github.com/orbist/Ge0Seq

Remove the Atmega328P from the device and insert into the Arduino Uno socket.

Open the Arduino IDE and load the Ge0Seq.ino sketch.

Edit the definitions.h file to ensure the correct “PRODUCT_NAME” is defined. Compile and upload the new version to the device using the Arduino IDE.

Replace the Atmega328P in the device and power back up.

Update the version of the Ge0sync Sequence Editor on your PC – either download from the Ge0sync website : https://geosyncsynth.com/documentation/ or from the GitHub releases page.

Upload Using avrdude

After installing avrdude and your ISP USB device, connect the corresponding MISO, MOSI, SCK, RST, GND and VCC pins from your ISP device to the header on the rear of the Ge0Seq module. Connect the 6 pins before connecting the USB device to your PC!

You should see the device power up as normal.

Once you are happy, download the latest hex firmware image from the Ge0sync website : https://geosyncsynth.com/documentation/ or from the GitHub releases page.

Open a terminal window and run the appropriate avrdude command, substituting “usbasp” for your programmer type

Note: Omit the -e flag from the first command if you do not want to erase the device first, thus leaving the sequences and settings on the device – be aware that the old format of the EEPROM may not be compatible but the device will fix this on next boot if that is the case (by re-formatting the EEPROM anyway!)

```
➢  ./avrdude -c usbasp -p m328p -B4 -e -U efuse:w:0xFD:m -U hfuse:w:0xDE:m -U
   lfuse:w:0xFF:m -U lock:w:0x0F:m
➢  ./avrdude -c usbasp -p m328p -b 19200 -U flash:w:.\TB-Seq-
   firmware_w_bootloader.hex:i
```

You should ensure that you also download the latest (matching) Ge0sync Sequence Editor application from the Ge0sync website : https://geosyncsynth.com/documentation/ or from the GitHub releases page.