MIDI THING V2

USER MANUAL

befaco

Version 1.1 (Jun 2024)

Bug fixes:

- Clock is processed from the first MIDI click received. So no 1 step lag should happen.
- St/sp message will make clock start on first midi click received.
- ADSR CC Mapping now works as intended.
- ADSR stage times over 9000 ms not broken anymore.
- LFO generating offsets when triggered with a specific Note.
- Pre-def list now have a back option.
- Note priority now affects to two different note ranges.
- Gates changing from voices fixed.
- Several items not saved fixed now.
- Memory corruption solved.

Changes:

- New pre-defs.
- LFO starts now in 0 volts on positive ranges when attenuated.
- LFO offset implemented.
- LFO MIDI Mapping.
- Flip screen option.
- After 5 mins of inactivity screensaver will kick in, switching screen OFF.

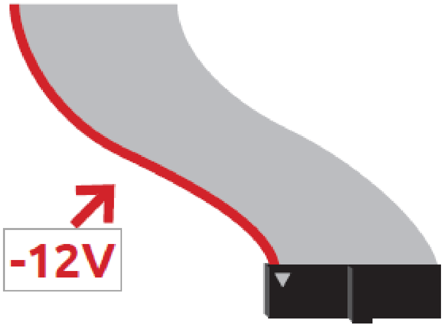
Known issues:

- ADSR visualization glitches. Although ADSR works fine.

POWERING THE MODULE

THANKS FOR PURCHASING A MODULE FROM BEFACO!
BEFORE YOU PLUG THIS MODULE IN...

1. Disconnect your cabinet from the mains.
2. Triple check the power cord polarity. The coloured line on the cable (pin number one) is the -12V rail.
3. If you plug the module backwards you might burn it out and unfortunately this is not covered by the warranty.
4. If you have any questions about this product feel free to contact us support@befaco.org

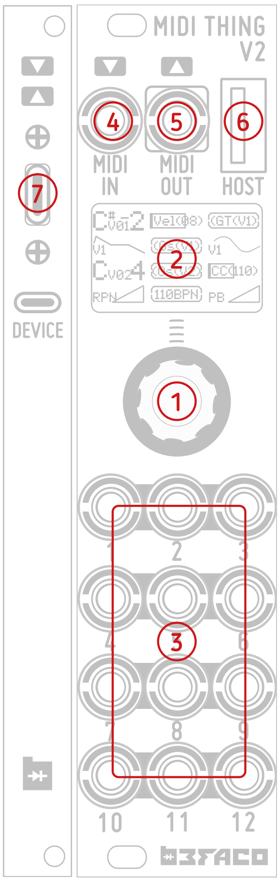


1. INTRODUCTION

MIDI THING V2 is a flexible MIDI to CV converter. Allowing polyphonic notes handling, envelope, Ocillator and LFO generation as well as all available MIDI messages to be converted into CV.

This is a huge upgrade from our previous beloved MIDI Thing, which adds a screen for easy configuration,12 assignable ports, TRS, USB Host and Device, MIDI merge OUT, a web configuration tool, and a VCV rack bridge counterpart.

2. CONTROLS AND CONNECTIONS



1. ENCODER CONTROL
   Rotary encoder control.
2. PERFORMANCE SCREEN AND PORT MAPPING
3. ASSIGNABLE PORTS OUT
4. TRS MIDI IN
5. TRS MIDI OUT
6. USB HOST
   USB port to connect USB MIDI controllers. this port provides up to 500 ma. power.
7. USB DEVICE VIA EXPANSION
   You can access MIDI Device connection using provided expander and USB-Cable.

CONNECTING MIDI

Midi Thing V2 can connect to MIDI devices using MIDI IN TRS, MIDI HOST, and MIDI DEVICE. You also have an available MIDI OUT, that merges all signals coming from TRS In, HOST and device and outputs as a MIDI OUT signal.

TRS MIDI IN

Type A TRS MIDI connector. Use provided adapter or a trusted Type-A adapter.

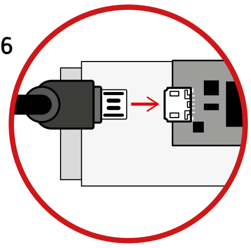
TRS MIDI OUT

Type A TRS MIDI connector. Use provided adapter or a trusted Type-A adapter.
This output can merge all module's MIDI ins (TRS, Host and device) making it a MIDI converter/forwarder or act as a MIDI thru for MIDI IN TRS. This can be configured in Global menu.

USB HOST

USB port to connect USB MIDI controllers. This port will draw current from your 5v rail.
Please mind tour powerr supply 5v max current draw when connecting a USB host device.
We strongly recommend to plug any *USB Host controller if it has a Power input adapter.

***Any MIDI Host connected will add its power consumption to this number. For example, a Keystep 37 will add around 96 mA. Do not draw more than 100 mA from this port. Meaning only small USB host controlleres. (For example Arturia Keystep 37 draws 90 mA).**



USB-C expander

USB MIDI DEVICE

In order to connect USB device, you will need to acces Teensy USB port in the back of the module. You have two options:

- **Using expansion panel:** Connect expansion panel MICRO-USB cable to Teensy USB port on the back of the PCB. Do this with power turned off.
- **Connecting to Befaco 7U Case.** Use a regular micro USB cable in order to connect Teensy micro USB conector to IO board USB A port. Do this with power turned off.

Internal Type-A USB socket

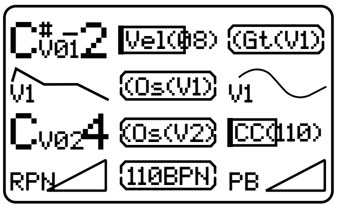
How it works

Once expansion is connected to the module, you can connect to a computer using a USB-C to USB-B cable*. If you are connecting MIDI Thing V2 to Befaco 7U Case**, use a regular USB-A to USB-B cable in the back of the case.
Upon connection, your computer should detect MIDI Thing V2 as a MIDI Device. From where you can map connections normally in your preferred DAW.

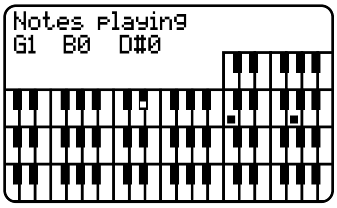
***** Mac users, USB-C to USB-C cables will not work.
****** Please refer to the Befaco 7U Case user manual for further explanation.

3. USER INTERFACE

When the module is powered ON you will see the performance screen, where you get feedback about the feature configured in each port and the activity.
Ports are displayed as they are located in the front panel. Use encoder to navigate the screen, each port will be highlighted then you can select it with encoder button.

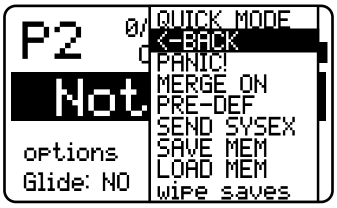


Turn the encoder counterclockwise to access the Notes MIDI monitor.
This screen will show the notes that the module is currently receiving.



Module has a built in screen saver. Screen will Dim down intensity after 5 minutes of inactivity.

3.1 GLOBAL MENU



Keep the encoder pressed to get to Global settings. This menu provides global configuration and several other features.

PANIC!: Sends All Note-OFF in all channels.

MERGE ON: By default MIDI OUT works as a MIDI thru, forwarding all data coming to MIDI IN. If Merge is set to ON, all data from USB device and MIDI host will also be forwarded to MIDI OUT, making the module a USB to MIDI converter.

FLIP VIEW: This option will flip screen 180º allowing to place the module with minijacks facing up.

PRE-DEF:
Loads a pre-def configuration. You can select among the following:

Predef 1: Polyphony
MIDI Channel1
4 voces: CV+Gate+Velocity

Predef 2: Multi-timbric
CV+Gate+Velocity MIDI Channel 1
CV+Gate+Velocity MIDI Channel 2
CV+Gate+Velocity MIDI Channel 3
CV+Gate+Velocity MIDI Channel 4

Predef 3: CCs preset
CC numbers:
14, 15, 16,17,18,19, 20, 21 ,22, 23, 24.25

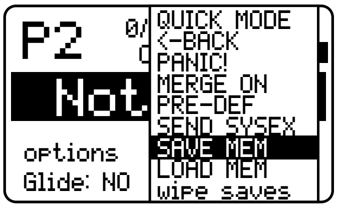
Predef 4: Bridge Bend
All Pitch bends MIDI Channel 1 - 12

Predef 5: Drum mode
MIDI Channel 10
Notas: 35,38,39,41,42,43,44,45,46,47,48,49

Predef 6: Drum mode mix
MIDI Channel 10
Notas: 35, 38, 39, 41, 42, 43, 44, 45, 46, 47, 48, 49

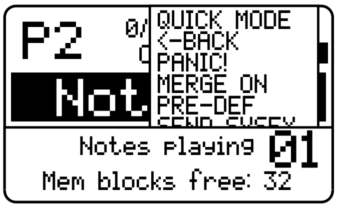
Predef 7: Blank preset
All No function

SEND SYSEX: This will send current module configuration via Sysex. Useful for our web editor.



SAVE MEM: Save your current configuration by selecting a slot. Incremental save means bigger files as the initial config is changed.

LOAD MEM: Load a saved config

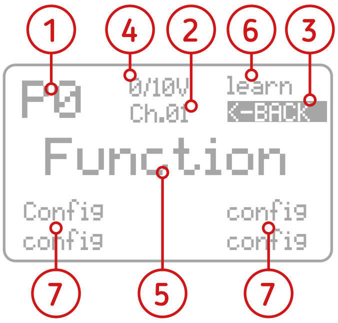


WIPE SAVES: Deletes all data saved in the module. If you are interested in backing this information up, you can use our web editor.
WARNING: All data wiped this way will be permanently lost.

On Startup, MIDI Thing will load last saved state by default.
If your Memory has any issue or you want module not to load this config, power on the module with the encoder button pressed.

3.2 PORT CONFIG SCREEN

In performance screen you can enter in port configuration by pressing the encoder.
Use encoder to move between configuration options

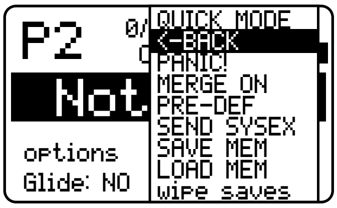


1. **P1:** Indicates the port we are configuring.
2. **MIDI CHANNEL:** This port will be listening to.
3. **BACK:** Go back to the previous screen.
4. **RANGE:** Selects CV range: 0/+10, 0/+8, 0/+5 and -5/+5.

5. **FUNCTION:** Shows function mapped to this port. Clicking opens menu to select function.
6. **LEARN:** Click port will listen to any MIDI message arriving. Then it will configure the module depending on relevant function.
7. **EXTRA CONFIGURATIONS:** Lower section of the screen will change depending on the function being configured.

4. MIDI THING FUNCTIONS

4.1 VOICE STRUCTURE



In order to convert MIDI notes into CV we will be using a voices structure. Like this MIDI Thing V2 can work as a polyphonic converter.
This is how classic synths managed multiple notes to act in monophonic or polyphonic modes. Where they used to have dedicated hardware to be able to play each note, allowing them to play as many notes at the same time as the number of voices they had in hardware

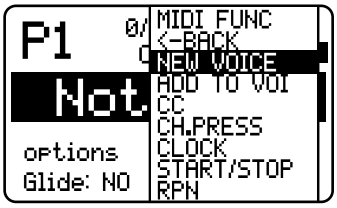
In MIDI Thing V2 we can create many voices, only with the limitation of the ports available.
A voice have the following information: CV, Gate, ADSR, Velocity, Drum, LFO and Oscillator. On voice creation, you can assign one or several of these features.
If you create several voices in the same MIDI channel, MIDI notes received will be assigned to the voices based on assignment and “stealing” algorithm selected. This can be configured in the Note screen, we will see this later.
Keep in mind that all items exist in the voice upon creation. But you will only have their output when configured to a port. So if, for example, you select ADSR to affect Oscillator amplitude, it will work although ADSR is not configured to any port. If you want to configure parameters of the ADSR then you need to assign it to a port to access the config.

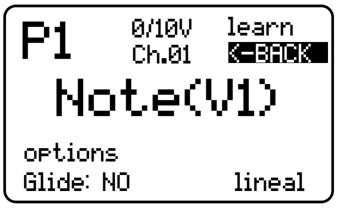
On the functions menu we have two options for voices: Create a new voice or adding features to an existing voice. Lets see them in detail.

NEW VOICE

This will create a new voice. On click, you are prompted to select which feature you want to map from that voice.

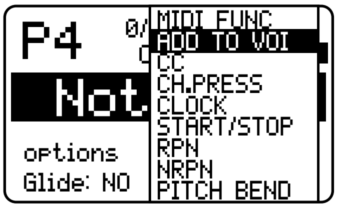
Voices will be named V1, V2, etc. You can create a maximum of 12 voices.
Voices will be automatically named in ascending port order.
So if you create a new voice on a port between two other ports with voices, the voices will be renamed. Same goes if you delete a voice, the rest of the voices from there will be renamed.
Also, in the same menu, you have macros to automatically map more features from the voice. For example, if you want to have.
Note + Vel+ Gate, CV will be mapped to the current port, velocity to the next one, and Gate to the third one.
For this example, if you are configuring this voice in Port 1, Note will be in 1, velocity will be automatically mapped to Port 2, and gate to Port 3 (as it follows port order for assignment).
Be warned that configuration will be overwritten on those ports.
When you have voices configured, when a MIDI note is received in the module, it will be assigned to the voice and the message converted to each function available in that voice.

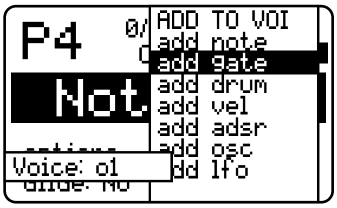




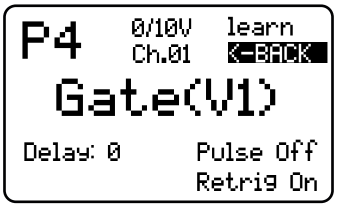
ADD TO VOICE

You can add more features to an existing voice.
On selection, you will be prompted to select the target voice.
Upon selection, you will see the feature selected / target voice.
If you add a feature already present in a voice, that feature will be erased and the current port will take it over.
So, following our previous example, if we add a gate to that voice in port4 you will see Gate(V1) in port 4. Then we will still have one voice, with Port 1 as Note, Port 4 as Gate, and velocity in Port 2, as adding a Gate to port 4 did erase the Gate we already had in Port 3.



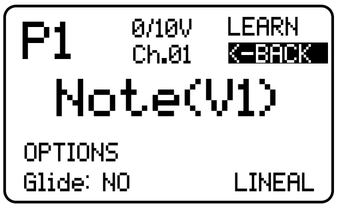


Example picture:
V1 by the function name indicates to which voice this feature belongs to.



VOICE ITEMS

NOTE CV



```
P1   0/10V  LEARN
     Ch.01  <-BACK
  Note(V1)
OPTIONS
Glide: NO      LINEAL
```

When a note is received and assigned to this voice, it will be converted to a CV value, following v/oct scaling. By default, at range 0/10v MIDI note 0 will be converted to 0 Volts, and note 120 will be converted to 10 v.

GLIDE

With this feature, you can select the time it takes the CV to change to a new value. Time is expressed in milliseconds. If its set to 0 the change of CV will be immediate, equivalent to being OFF.

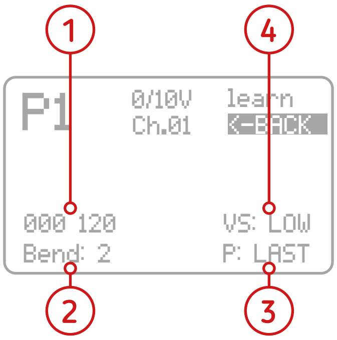
GLIDE MODE

There are three responses for the glide effect: Lineal, Fast, and slow.

LEARN

When pressed, MIDI Thing V2 will listen to the next MIDI note received. Then it will learn its MIDI channel and Will set that note as the new 0 volts point.
If there is another voice that had a learn and a range set, it will be respected. This is useful to split keyboards. We recommend doing this from top to bottom. For example, we do a Learn in an existing voice and send Note 75 in MIDI channel 1. So that voice will only process notes from 75 to 120.
Then we do learn in a second voice and send MIDI note 25. so learn will set that voice to start processing in note 25, but as there is another range set in that MIDI channel, the maximum note will be 74.

OPTIONS SCREEN
In this screen, you can configure the following extra parameters:



```
P1   0/10V  learn
     Ch.01  <-BACK

000 120       VS: LOW
Bend: 2       P: LAST
```

1. MIDI RANGE

You can select the first and last MIDI notes that Will be processed for this port.
This clips the range to be processed, but not scale it.
For example, if you select here 0 – 50, only those notes will generate voltages in this port. Then you can select range 51-127 in a different port to split a MIDI keyboard in two.
The voltage will be 0v. in the first note of the range.

2. PITCH BEND

Selects if pitch bend affects the voltage (o Bend) and sets the number of semitones that will be moved up and down.

3. NOTE PRIORITY

When a note is received the algorithm to assign it is as follows:
1- Check if there is a free Voice for this note based on MIDI channel and accepted note ranges. If several voices are free meeting the criteria, the voice is selected using the voice assignment method (see below)
2- If all voices are in use, the note stealing algorithm will determine which voice will be playing the new note or it will be ignored if no priority is selected.

Selects note priority when a new note is assigned to a voice. This applies in the case of new notes arriving and all defined voices are already in use.
You can select whether older or newer notes will be stolen or higher or lower voices. also no steal at all

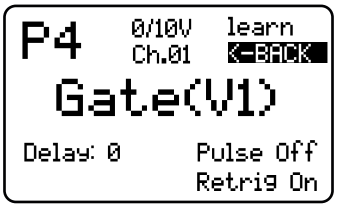
FIRST: Keeps the first notes received, stealing the last one played.
LAST: Keeps the last notes received, stealing the first played before the available voices ran out
HIGH: Keeps higher notes. The lowest will be stolen
LOW: Keeps lower notes. The highest note will be stolen
NO STEAL: When the polyphony runs out, no more notes will be played.

4. ASSIGN METHOD

We have two assign methods Lower To Higher (Low in screen) and Round Robin.
Default mode is Low, where as notes arrive, they will be assigned to existing voices in ascending voice order. If we receive a note off for an already assigned note, then a new note arrives it will be assigned to the first available voice after the last assigend.
For example, we have three voices configured. We receive two note ON, so they will be assigned to the first two voices: V1 and V2, then we receive a Note off for the note assigned to V1, and just after that a new Note ON arrives. The new Note will be assigned then to V3, following the ascending order.
If Round robin method is selected, notes will be assigned to the first available voice. So in the example mentioned before, the next note received will be assigned to V1, as it was free, before assigning it to V3.

GATE

Gate output of selected voice. Port linked is stated on screen.
The available ranges are: 0/10v, 0/5v and 0/8v
When Note ON arrives gate go up and will go down with a Note OFF.



```
P4   0/10V  learn
     Ch.01  <-BACK
  Gate(V1)
Delay: 0   Pulse Off
           Retrig On
```

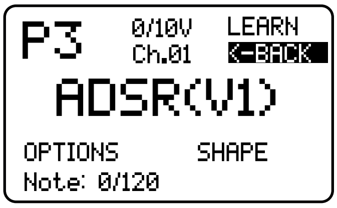
PULSE: If selected a trigger will be generated instead of a gate.

RETRIGGER: Mode will make the gate go down for a short time when a second Note ON is processed when the Note OFF of the previous note has not yet arrived.
By default is OFF, so gate only will go down when all Note OFF are processed.

DELAY: Will set a delay to the Gate against the CV. allowing to humanize a bit the notes played or correct some sequencer's delay between CV and Gate.

ADSR

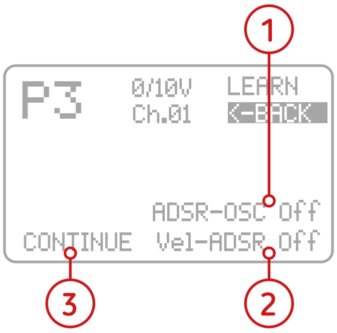
With this option you can generate an ADSR linked to this voice.
When note ON arrives, attack stage starts until it gets to maximum level, then decay stages going down to sustain level. Then maintain CV level set by sustain until a Note OFF arrives, so the release stage happens.



Note: Thai is a voice parameter, you can change the notes range accepted by the voice. notes outside the range will not be assigned to this voice.
TRICK: you can set this range to only one note, so when that note arrives you get an envelope generated!!

OPTIONS SCREEN

In this screen, you can configure the following extra parameters:



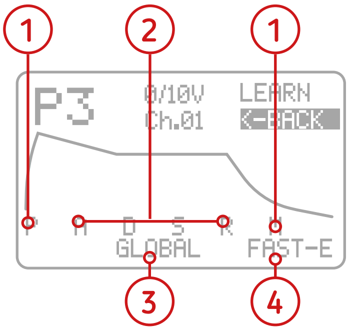
1. ADSR-OSC: Applies ADSR to Oscillator amplitude. So if you have an oscillator in this voice, it will only work when a Note arrives and ADSR cycle is started. By default is set to OFF

2. VEL-ADSR: Routes that voice's velocity to scale ADSR amplitude. Being 127 100% of the scale and 0 0%. By default is set to OFF.

3. RETRIGGER: Selects if the ADSR will retrigger with a new Note being assigned to this voice. Options are:

> **CONTINUE:** If a new note is assigned to this voice and previous note's NOTE OFF did not arrive, ADSR will continue from the point it was. This is the default mode.
>
> **RETRIGGER:** If a new note is assigned to this voice and previous note's NOTE OFF did not arrive, ADSR will be reset to zero and start over. When Note off for this new note arrives, ADSR will be triggered again. Same way as we do in Gate mode Retrigger.
>
> **NO RETRIGGER:** If a new note is assigned to this voice and previous note's NOTE OFF did not arrived, ADSR will not be triggered. In order to have a new ADSR triggered, previous one Need to be finished first.
>
> Any note that arrives during an ADSR cycle will be ignored.

SHAPE SCREEN



WARNING: If you are changing options in the menu while your ADSR is receiving notes, it might glitch. Better to change this option while you are not performing
In this screen you can select ADSR times and shape.

1. P: Set a pre-delay for the ADSR to start the cycle.

2. A, D, S, R: Change each stage time. **S** will change the sustain level.
Click **LINEAR** to change from linear to log response. This changes **A. D** and **R** curves.

1. M: Scales the maximum amplitude of the ADSR.

3. GLOBAL/LOCAL: By default voices are created as Global, meaning that in the same MIDI channel all ADSRs share the same profile. In this case, if any ADSR option is changed all ADSR from the channel will be changed. When local is selected, the voice will have an independent envelope.

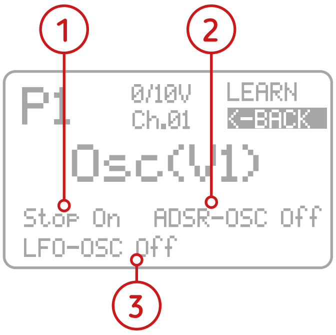
4. SHAPE: This will change the shape of the ADSR. Available options are: Lineal, Fast-E, Slow-E.

ADSR MIDI MAPPING:

By Default ADSR parameters are mapped to CC values. If you send these CCs in the same MIDI channel where the ADSR is configured, you will be able to change the parameters:

CC73 Attack, **CC75** Decay, **CC79** Sustain, **CC72** Release

OSCILLATOR



P1 0/10V Ch.01 LEARN <-BACK
Osc(V1)
Stop On ADSR-OSC Off
LFO-OSC Off

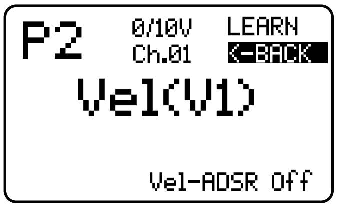
Provides a wave with frequency provided by the note mapped to the target port. Oscillator will keep the frequency until a new note ON arrives.
This is a bass-oriented Oscillator. As it's heavily filtered, after the third octave, you will notice filtering and some digital dirt. We recommend to stick to low notes. Nevertheless, higher notes are still usable, especially with ADSR enabled!

1. **STOP:** If enabled Oscillator will only oscillate when a note ON arrives. Then stops working when Note Off arrives.

2. **ADSR-OSC:** Applies ADSR to Oscillator amplitude. So if you have an oscillator in this voice, it will only work when a Note arrives and ADSR cycle is started. By default is set to OFF

3. **LFO-OSC:** LFO will be routed to this voice Oscillator´s amplitude.
By default is set to OFF

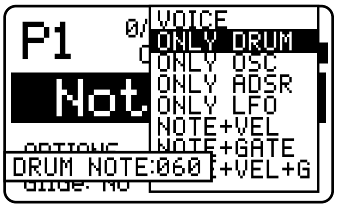
VELOCITY



P2 0/10V Ch.01 LEARN <-BACK
Vel(V1)
Vel-ADSR Off

Converts the Velocity associated to a note assigned to this voice into CV.
VEL-ADSR: Enables Velocity to control this voice ADSR amplitude.

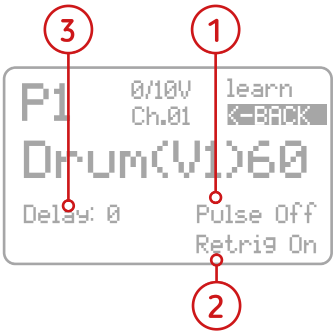
DRUM TRIGGER



P1 0/ C
Not
OPTIONS
Glide: No
VOICE
ONLY DRUM
ONLY OSC
ONLY ADSR
ONLY LFO
NOTE+VEL
NOTE+GATE
+VEL+G
DRUM NOTE:060

This feature is used to trigger drum modules using MIDI Notes.
Upon selection of this feature, you will be prompted to select to which note this port will react.

When a selected note is assigned to this voice, gate will be set to high. It Will go low when Note Off arrives.
This feature acts as a gate, it provides a gate out for an incoming note assigned to it. You can add other voice items related to this one.
When you add a Drum to a voice containing a Note, Note will be deleted.
If you add a Note to a voice that already has a Drum, Drum will be converted into a gate that reacts to every note assigned to the voice.
If you add a Note to a voice that already has a Drum, The Note will only react to the note of the Drum.



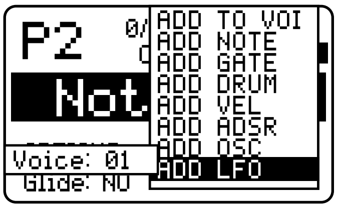
P1 0/10V Ch.01 learn <-BACK
Drum(V1)60
Delay: 0 Pulse Off
Retrig On

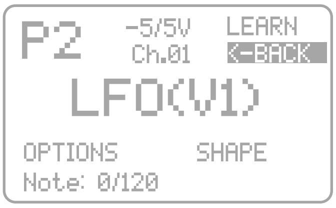
1. **PULSE:** Activates Trigger mod. So only a trigger will be sent when note on arrives.

2. **RETRIGGER:** This will make the gate go down for a short time when a second Note ON is processed when the Note OFF of the previous note has not yet arrived. By default is OFF, so gate only will go down when all Note OFF are processed.

3. **DELAY:** will set a delay to the Gate against the CV. allowing to humanize a bit the notes played or correct some sequencer's delay between CV and Gate.

LFO



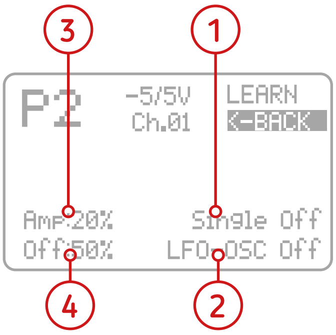


WARNING: If you are changing options in the menu while your LFO is receiving notes, it might glitch.

This will add an LFO tied to the voice. You can select waveform and the LFO can be triggered by the Note assigned to the voice, or oscillate freely. Either in sync to MIDI clock or run freely as well.
Note: This is a voice parameter, you can change the notes range accepted by the voice. Notes outside the range will not be assigned to this voice.

**TRICK: you can set this range to only one note, so when that note arrives you get an LFO generated!*

OPTIONS SCREEN: Extra options for the LFO.



1. **SINGLE SHOT:** LFO will only do one cycle when triggered.

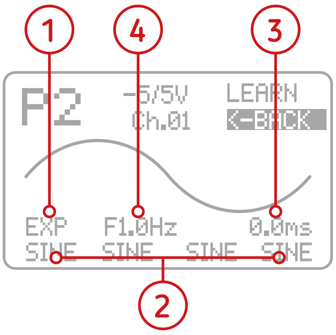
2. **LFO-OSC:** LFO will be routed to this voice Oscillator´s amplitude.

3. **ATTEN (Amp):** LFO attenuation.

4. **OFF-SET:** The LFO Offset allows to set the LFO in the following range:
Maximum offset 100% represent an LFO curve which maximum value is the maximum Port Range output
Minimum offset 0% represent an LFO curve which minimum values is the minimum Port Range output.

Be warned that LFO will nor clip. So if you apply an offset that will make the signal go out of range it will not allow it.

SHAPE SCREEN
Enter to LFO shape menu. You can select LFO waveshape per stage. Or the shape of all stages at once. Yo ucan also change time divisions.



1. **GLOBAL LFO SHAPE**: Options available are Exponential, Sine, Square, Ramp up, Ramp Down, Random and Triangle

2. **SECTOR LFO SHAPE**: Options available are Exponential, Sine, Square, Ramp up, Ramp Down, Random and Triangle

3. **PRE-DELAY:** Apply a delay before starting LFO cycle. By design, Pre-delay cannot be longer than LFO cycle time.

OTHER OPTIONS AVAILABLE IN THIS SCREEN ARE:

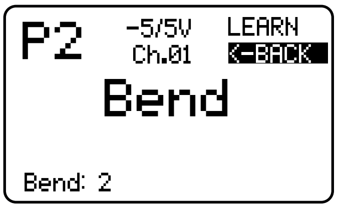
4. **CLOCKED/FREE:** Select if LFO is clocked or **FREE**. **C** or **F** in screen.
FREE give you the choice to set oscillation hertz speed.
SYNC LFO will be synced to MIDI clock.
PPQN DIVISION Sets ppqn (Pulses per quarter note) division. Values are: long, double, whole, half, qn, 8th, 16th, 32th and qn/24. Keep in mind MIDI clock send 24 Pulses per quarter note.
MULTIPLIER You can multiply the clock after setting ppqn division.
Pre-delay: apply a delay before starting LFO cycle.

LFO MIDI MAPPING:
By Default LFO parameters are mapped to CC values. If you send these CCs in the same MIDI channel where the LFO is configured, you will be able to change the parameters:
CC76: Controls the period of the LFO.
CC77: Controls LFO max level.
CC78: Controls LFO Pre Delay.

4.2 PITCH BEND

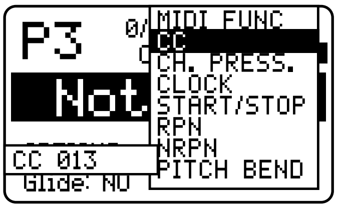
Converts pitch bend messages into CV. By default, values are converted to -5 /+5 . If 0/+10v is selected, 0 value will be 5 volts, then minimum value 0 and maximum 10v.

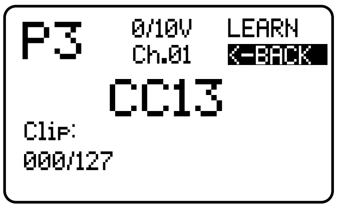
BEND: Select how many semitones


P2 -5/5V LEARN
Ch.01 <-BACK
Bend
Bend: 2

4.3 CC

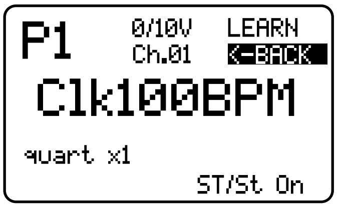
Converts selected CC value into CV. On selection a pop up will ask CC number to be processed. CC value will be converted to the selected output range.
CLIP: Select the range of CCs that will be processed. Clipping the rest of the values.
WARNING: There are some CC values hard coded to parameters of ADSR (check their definitions) You can map them anyway to this port, but keep in mind they will still affect those parameters.
*Note: Two CCs cannot be configured with the same value and MIDI channel.




P3 0/10V LEARN
Ch.01 <-BACK
CC13
Clip:
000/127

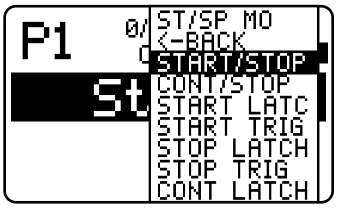
4.4 CLOCK

This feature will convert clock MIDI messages into modular-level clocks.
You will be selecting the ppqn division first, keeping in mind MIDI is 24 ppqn, then applying a multiplier to that division.


P1 0/10V LEARN
Ch.01 <-BACK
Clk100BPM
quart x1
ST/St On

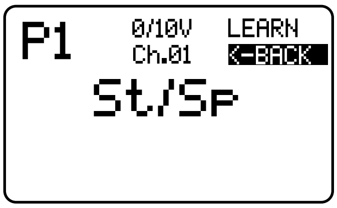
You can also configure multiple clocks to multiply the fun!
PPQN DIVISION: Sets ppqn (Pulses per quarter note) division. Values are: long, double, whole, half, qn, 8th, 16th, 32th and qn/24.
MIDI CLOCK: send 24 Pulses per quarter note.
MULTIPLIER: You can multiply the clock after setting ppqn division.
ST/STOP: Will toggle if clock processing stops when a Stop message is received.
The Start/Stop/Continue MIDI messages will change the running status in the module.
Start and Stop will respectively activate and deactivate the running status.
The Continue message will switch alternatively running status from on to off
The module inits with the Running status set to true.
This means that when clock messages are received they will be processed without the need to receive a Start or continue message.
This behaviour can be prevented by sending a Stop Message at the beginning of the setup.
When Clock ports are configured to ST/St On, the clock signal will be sent only when the module is in running status on.
When the start message is received, the clock port is configured with this option, it will be reset.
When continue is received, the port will stop sending pulses but will keep in sync with incoming clock messages.
So, when next continue message is received, the port will keep sending in the original sequence.
Clocks configured as ST/St Off will ignore the running status and will send pulses as long as clock messages are received.
WARNING: If settings are made while the clock is running, the clock may go out of sync.

4.6 START / STOP



P1 0/ C St

ST/SP MO
<-BACK
START/STOP
CONT/STOP
START LATC
START TRIG
STOP LATCH
STOP TRIG
CONT LATCH



P1 0/10V LEARN
Ch.01 <-BACK
St/Sp

This function will convert Start , stop and continue messages into logic levels. Available options are:

Start/Stop: You will get a high gate when Start message arrives. The gate will go down when a Stop message arrives.

Continue/Stop: You will get a high gate when Continue message arrives. The gate will go down when a Stop message arrives.

Start Trigger: You will get a trigger when a Start message arrives.

Stop Trigger: You will get a trigger when a Stop message arrives.

Continue Trigger: You will get a trigger when a Continue message arrives.

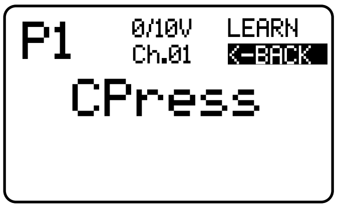
Start Latch: You will get a high gate when Start message arrives. The gate will go down when another Start message arrives.

Stop Latch: You will get a high gate when Stop message arrives. The gate will go down when another Stop message arrives.

Continue Latch: You will get a high gate when Continue message arrives. The gate will go down when another Continue message arrives.

4.7 CHANNEL PRESSURE

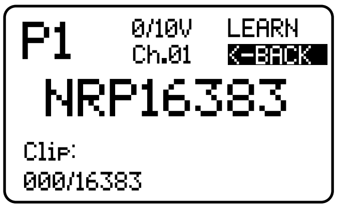
This function will provide a CV proportional to selected channel´s Channel Pressure message.



P1 0/10V LEARN
Ch.01 <-BACK
CPress

4.8 NRPN

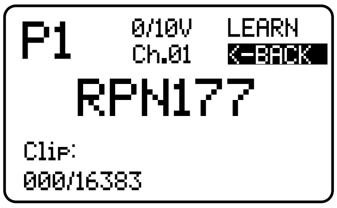
Converts selected NRPN value into CV. On selection a pop up will ask NRPN number to be processed. It will be converted to the selected output range.
CLIP: Select the range of NRPN that will be processed. Clipping the rest of the values.



P1 0/10V LEARN
Ch.01 <-BACK
NRP16383
Clip:
000/16383

4.9 RPN

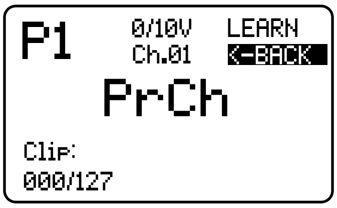
Converts selected RPN value into CV. On selection a pop up will ask RPN number to be processed. It will be converted to the selected output range.
Clip: Select the range of RPN that will be processed. Clipping the rest of the values



P1 0/10V LEARN
Ch.01 <-BACK
RPN177
Clip:
000/16383

4.10 PROGRAM CHANGE

This function will provide a CV proportional to the value of the program change received

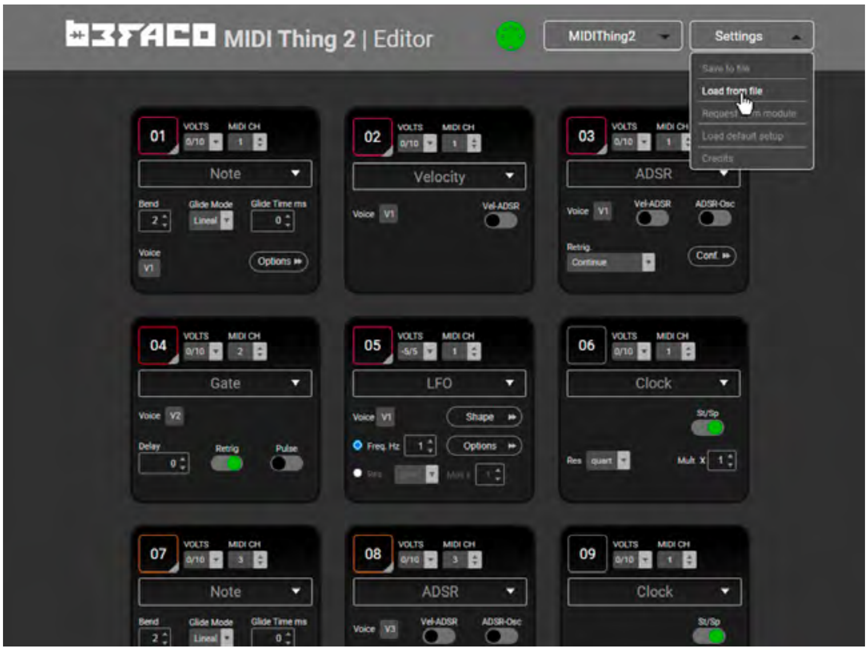


P1 0/10V LEARN
Ch.01 <-BACK
PrCh
Clip:
000/127

5. USING WEB CONFIGURATION TOOL

There is a configuration site available to easily configure MIDI Thing V2 from your browser.
Be warned that in Firmware version 1, you can only configure MIDI Thing via its USB Device connector.

Go to https://befaco.org/configmidithing2 (Tested in Chrome in Windows and MAC and Chromium in Ubuntu.)
This tool will configure real time your MIDI Thing V2, make sure to save your changes in the module when finished. Or store your configs locally on your computer!



MIDI DEVICE SELECTOR

This will allow you to select MIDI device. Select MIDI Thing V2 from the list if you are using USB expansion to connect to your computer.

SETTINGS MENU

In this menu you will be able to pull/push configurations from/to Midi Thing V2. You can also save current configuration in the site to a file, in order to load it back in the future!

REQUEST FROM MODULE: this option will query current status of the module and display it in the website. This is useful if you make some changes in the module while using this site. Remember that changes made in the module will only be present in the site if you Request from module.

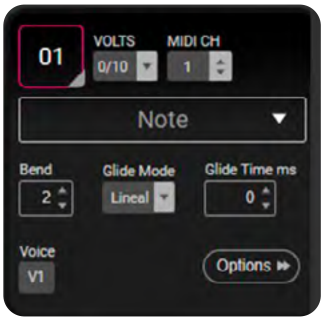
SEND TO MODULE: all changes made in the site will be automatically sent to the module. If there has been any communication hiccup, you can use this option to force the configuration into the module.

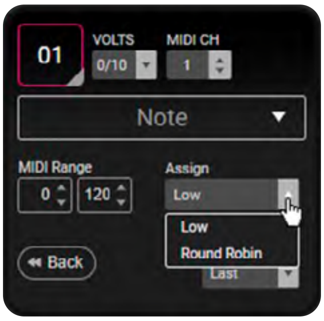
PORTS CONFIGURATION

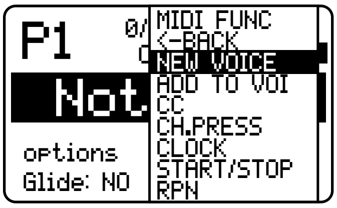
You have twelve cells to configure each of MIDI Thing 2 ports. You can configure all parameters available in the module.

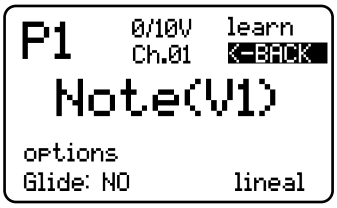
CONFIGURING VOICES

Creating voices in this tool follows the module structure. You will be prompted in Function dropdown for New Voice and Add to Voice.
When you Add a new feature to an existing voice, a pop up will ask to select target voice. Each voice will be color coded to identify to which voice each port is assigned.









6. FIRMWARE UPDATE PROCEDURE

Save states are not back compatible so you will need to delete them. Before updating your MIDI Thing wipe your Saves. Global Menu -> Wipe saves.
*Note: If you have several presets you will have to export them one by one.

To find out what firmware version is currently installed on your module, take a look at the splash screen that appears when you turn on the module.

1. Get latest firmware file from Hex folder in our github. Make sure you clone the repository, as downloading just the file won't work.
https://github.com/Befaco/Midithing2

2. Install Teensy uploader app found here
https://www.pjrc.com/teensy/loader.html

Or this other tool, just in case you want to dig in more Teensy funtionalities
https://koromix.dev/tytools

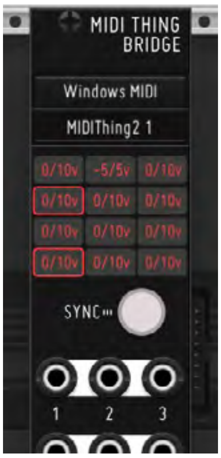
3. Connect Midi Thing V2 to your computer using a USB device.

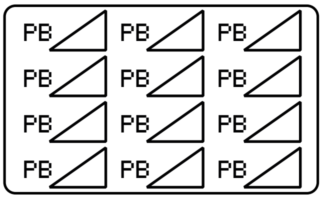
4. Run Teensy uploader and load hex file. If prompted by the app, press button at Teensy board in the back of the module to begin transfer...

Make sure module is plugged in.

7. VCV RACK MIDI THING BRIDGE

1. Set up your MIDIThing device or MIDI interface in the MIDI Thing Bridge.
2. Press the Sync button and the module will go into 12 port pitch bend mode, if that doesn't happen go to the Global Menu on the module and select the Predef. VCV Pitch Bends.
3. Plug control-rate signals at VCV side and they will appear in the module. Keep in mind that the higher the frequency of the signal, the lower the resolution you get in the module. This is due to MIDI speed rates limitation.
Please read manual of VCV module for further details.





Size: 6HP

Deeth: 30 mm

+12v: 30 mA

-12v: 5 mA

+5v : 120mA*****

Designed and developed by Befaco team.

Firmware by Sergio Retamero

Web configurator by Hugo Vazquez & Yago Nuchera

VCV Bridge by Ewan Hemmingway

Thanks a million to our Beta-Testing heros: Sergio Cañón, Alberto Navarro, Juan Cortina, Van Ta and Jon "Jessica Fletcher" Modular.

*Any MIDI Host connected will add its power consumption to this number. Do not draw more than 100 mA from this port. Meaning only small USB host controlleres.