uniMatrix technical manual

rev. 1.2

Introduction

uniMatrix is a universal MIDI interface suitable for many synthesizers, drum machines and other gear. It can interface existing keyboard matrixes directly and provides MIDI input and output. Additionally it offers multiple inputs and outputs to control or read additional parameters both analog and digital.
These features allow it to control many aspects of a synthesizer, not just the keyboard. Often, sound or patch settings, filter cutoff, or similar can be controlled as well.

uniMatrix is freely configurable by using our configurator tool.

This manual explains how uniMatrix works and how it can be utilized to add midi to various types of gear. This is not a user guide on how to install uniMatrix in a specific machine, or how to use a uniMatrix modified machine. For specific machines, we do have install and user guides on our website.

If a manual exists for a specific machine, you probably do not need to read this document.

Features

- Midi input and (in some cases) output
- Freely configurable universal midi interface
- Can interface existing keyboard matrixes of any size up to 13×9, each contact freely configurable
- Supports Notes, CC, program change, and more
- Up to 22 I/O pins can be configured as digital inputs and outputs
- Outputs up to 22 MIDI clocks with adjustable divider and MIDI start/stop signals
- 2 additional CV-outputs to control filter or other parameters
- 6 pins configurable as CV-inputs
- Short-push mode to simulate short button presses
- Optional learn button to change the MIDI channel quickly
- Small size fits into almost any gear
- 3-12V supply voltage, 3-5V signal voltage
- All I/Os with over-voltage protection
- DIN or TRS Midi sockets available

Table of Contents

What is uniMatrix

uniMatrix is retrofit by Tubbutec designed to provide a MIDI interface to various types of synthesizers, drum machines or effects.

Specifically, it can interface keyboard diode matrixes, but can also provide digital and analog outputs to interface other kinds of architectures.
Furthermore, uniMatrix can be used output MIDI from key presses, or analog voltages. The turn of a knob of a synthesizer could be sent out via MIDI CC for example.

uniMatrix is not suited to interface string synthsizers or organs (unless they use a diode matrix). For this purpose, please have a look at our OrganDonor interface.
For drum machines, please see uniPulse.
Even though uniMatrix provides two CV-outputs, these are not designed to output pitch CV. An interface to control synthsizers via CV/Gate is in development.

Interfacing keyboard matrixes

Many electronic devices utilize a circuit called keyboard matrix, sometimes also referred to as diode matrix to read buttons and switches. Here, buttons are arranged in a grid defined by a number of output pins and input pins. Typically, each button has a diode in series to prevent cross talk. Each output is turned on one-by one and the inputs are read. This allows it to read many buttons with a relatively small amount of pins.

uniMatrix can interface this type of configuration by ‘listening’ to the outputs and injecting the correct stimulus signal into the devices inputs at the right time. uniMatrix supports both kinds of polarities. Up to 9 input (N) and 16 output (M) signals are supported, where N+M <= 22.
uniMatrix can also generate MIDI output by reading the buttons or keys connected to the keyboard matrix, debouncing and sending out corresponding midi messages.
In some cases, MIDI input and output can even work simultaneously, but this depends on various timing factors and can not be guaranteed to work.

Each button in a matrix can be freely configured to react to MIDI notes, CC or program change messages. Furthermore, uniMatix can simulate short button presses automatically.

Types of matrixes

A keyboard matrix has a number of outputs N and a number of inputs M forming a N x M grid. There are two types of matrix polarity: Active high and active low.

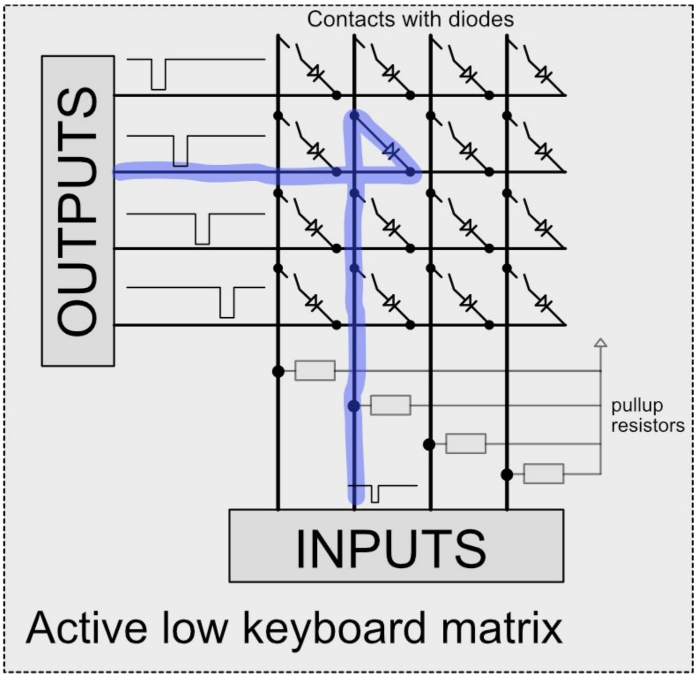
In active high matrixes, the signals at the outputs are normally low, and turn high sequentially to read on row of the matrix. At the input side, pulldown resistors are used to prevent the inputs from floating. If a button is pressed, the signal at the corresponding input becomes high via the path through the switch and diode.

In active low matrixes, the signals at the outputs are normally high, and turn low sequentially to read on row of the matrix. At the input side, pullup resistors are used to prevent the inputs from floating. If a button is pressed, the signal at the corresponding input gets pulled low via the path through the switch and diode.

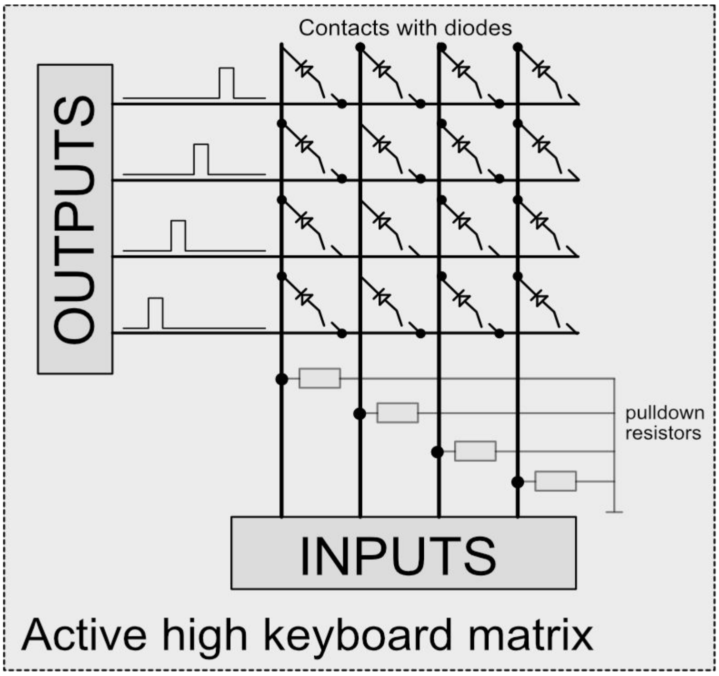
Identifying the matrix type

The simplest way to identify the matrix type is to check if the circuit uses pulldown or pullup resistors. This directly identifies the location and number of inputs and outputs and the matrix polarity.

In some cases however, the pull resistors are internal to an IC. Here, it is not directly possible to identify inputs and outputs. A multimeter, or even better an oscilloscope can be used to find outputs (they carry a signal), and inputs (they only carry a signal when a key is pressed). If an input measures high without pressing a key, the matrix is active low. If it measures low without pressing a key, it is active high.
The polarity of the diodes can also give some clues.



Active low keyboard matrix with pullup resistors. One switch is closed allowing output signal to drive an input high

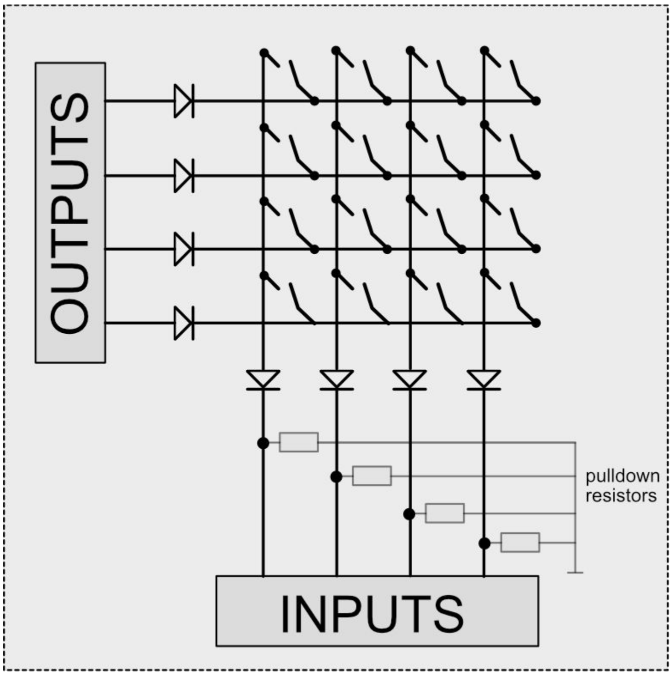


Active high keyboard matrix with pulldown resistors.

Fewer diodes

Some keyboard matrixes save on diodes, and only use them at the outputs and inputs (N+M), instead of for every key (N*M). This diode saving has some drawbacks:

- N*M diodes: All keys can be detected simultanously without interference
- N+M diodes: Can detect up to 2 key pressed correctly, but will create additional 'ghost' key pressed if more are pressed.



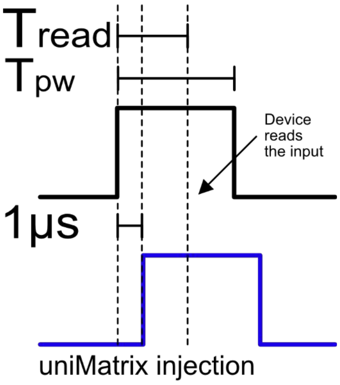
Keyboard matrix with only N+M diodes

Timing

MIDI input timing

The sequential output signals of a host device have a certain width of time T_{pw}. At some point while the signal is active, the device reads the state of its inputs. At this moment, the injection signal from uniMatrix needs to be present. Otherwise, the key press will not be detected. The typical delay between an output signal change and the injection signal generated by uniMatrix is about 1µS.

The time T_{read} is usually not known, so it is hard to predict if uniMatrix will work in advance. However, by looking at the pulse width T_{pw}, one can at least get an idea. If T_{pw} is much larger than 1µS, you can be pretty certain, that it will work.
In many devices, T_{read} is typically in the order of 10µS - 1ms, in order to allow correct keyboard reading and circumvent issues with keyboard capacitance.
However, we have also found an instance of a synthsizer with T_{read} < 1µS (a Kawai SX210). In such a case uniMatrix will not work and we recommend to use our OrganDonor interface instead.



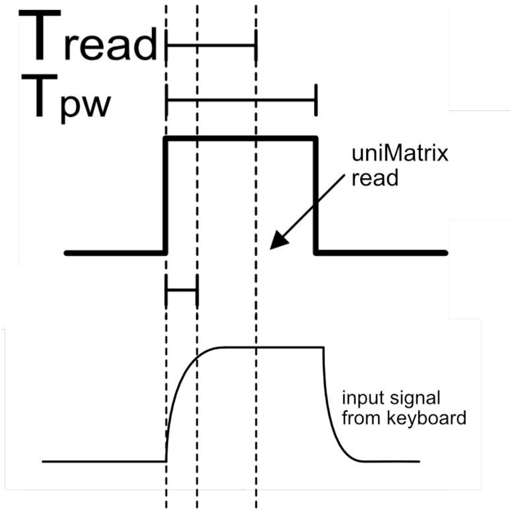
A successful injection as T_{read} is larger than 1µS

MIDI output timing

The same reason the host device needs to wait a little bit after sending the output signals, is also true for uniMatrix in MIDI output mode.

Due to keyboard capacitance, it takes a while for the keyboard signal to be valid.

The wait delay of uniMatrix is adjustable in 50ns steps



The keyboard signal needs some time to stabilize.

Simultanous input and output

uniMatrix supports simultanous MIDI input and output. However, this feature is not guaranteed to work as its success greatly on the signals generated by the host device.

Essentially, uniMatrix will first read the state of the host device's input signals in order to generate MIDI output, and then injects its signals into the inputs.
The read must happen at a time when the keyboard signals are already stable, but still the injection must happen before the host device reads the state.

This can sometimes be tricky, especially when the host happens to read different keys at different times.

The problem with simultaneous MIDI input and output

When trying to use MIDi input and output simultaneously, depending on the way the host device scans the matrix problems can arise. We can not guarantee MIDI input and output to work on all devices simultaneously.

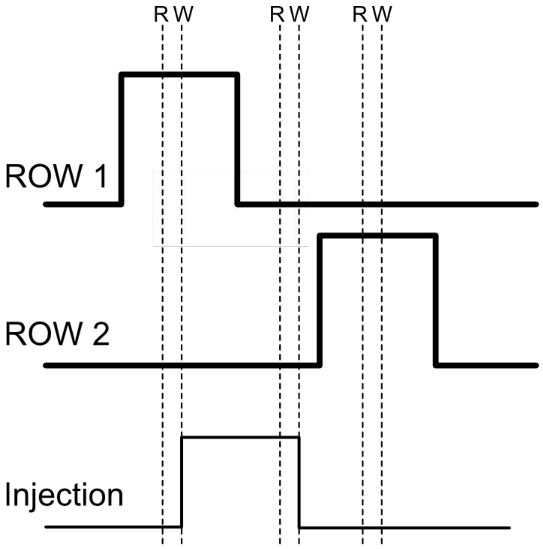


Figure 2: the injection signal turns off before the next row read

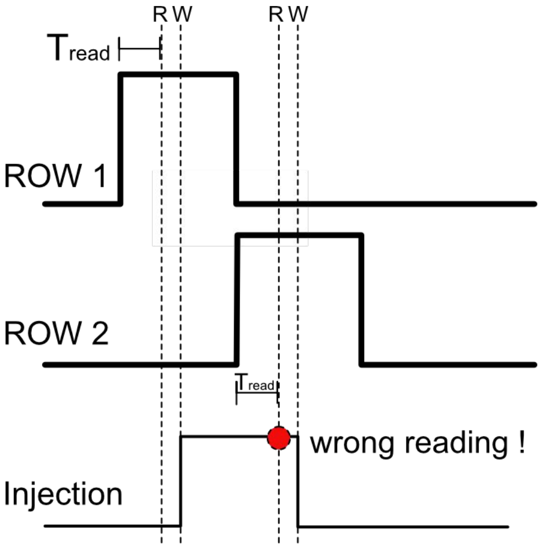


Figure 1: The read occurs before the injection signal turns off, resulting in an incorrect reading

Figure 1 shows a situation, where the host device outputs a selection signal on row 1, then tuns it off, and after a break turns on the activation signal for row 2.
You can see, that the uniMatrix injection signal is delayed by the configured T_{read} time and not active when row 2 is read. In this case, there is a well defines separation between read and write events and MIDI input and output possible simultaneously.

In figure 2 however, the host devices turns off row 1 and simultaneously turns on row 2. When uniMatrix reads the signals for ROW2, its own injection signal has not turned off yet. The result is an incorrect reading. In such a case, sending a note to uniMatrix, will result in a different note being returned from uniMatrix via MIDI out.

One way to fix this behavior is shown in figure 3:
Every time the row signals change, uniMatrix first turns off all injection signals, and only turns them back on after a reading.
In theory, this solves the above issue, but can cause other problems, mostly due to keyboard capacitance. It can be very difficult to find the right value for $T_{read.}$ In one synthesizer we tested, incorrect output notes were eliminated for some keys, but not for others.

This mode is available as “midi input and output v2” in the configurator.

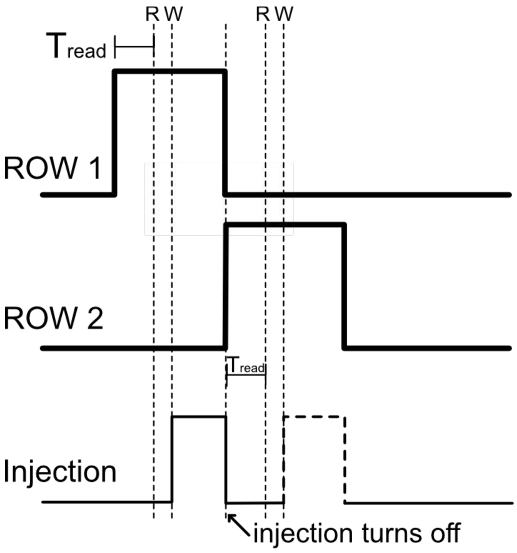
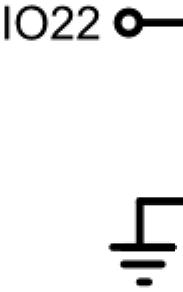


Figure 3: I/O mode v2

Switching between MIDI input and output

As an alternative to simultanous MIDI input and output, the user can select between input and output mode using a switch.

If “Midi input and output switch” is selected as matrix mode, IO22 is configured as an input to switch between input and output. In this mode, IO22 has an internal pullup. Pulling IO22 to GND, will result in MIDI output mode. Not pulling it to GND, will result in MIDI input.



Switch open: MIDI input, switch closed: MIDI output

Interfacing the matrix

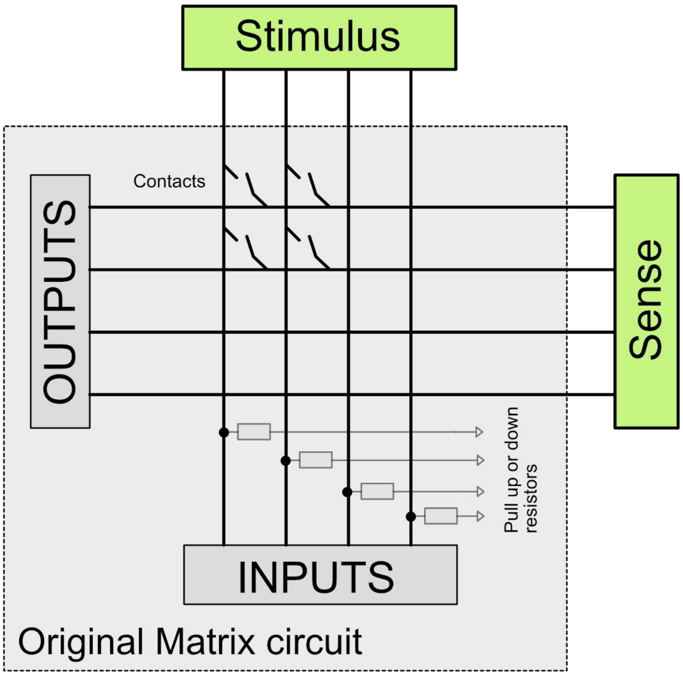
After the matrix type and dimensions has been identified, uniMatrix can be connected to its input and output signals. In the configurator, select the correct number of inputs and outputs. The configurator will then tell you which signal to connect. Starting from uniMatrix pin 1, first connect to the output signals, then the input signals.

uniMatrix will 'listen' to the host device's output signals via the sense lines and inject matching signals into the host device's input lines.

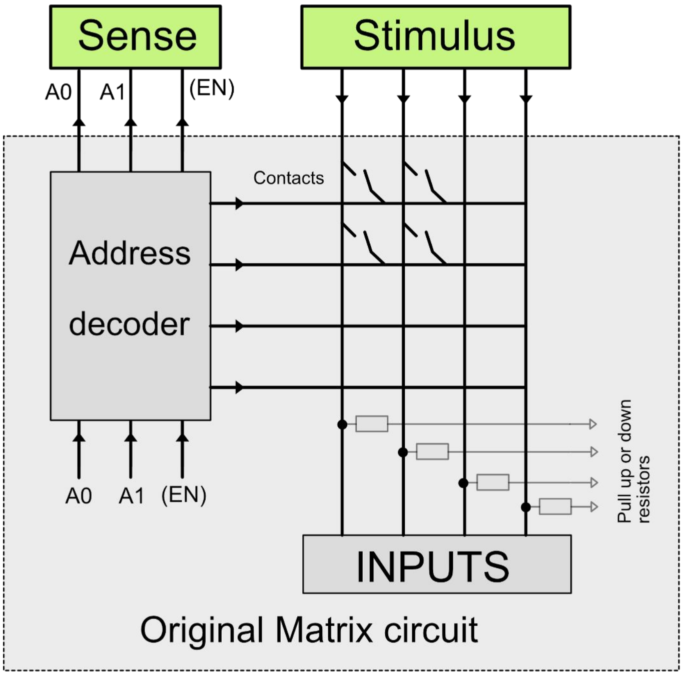
Output address mode

In some cases, the output signals are created using a multi-bit address signal decoded by an IC. For example, a circuit may use a 74xx137 to create keyboard matrix pulses using 3 address and one select signal.
In this case, uniMatrix can directly read the address and select signals instead of having to read the output signals themselves. This saves pins, and may also be a bit faster.

When using address mode, an optional enable signal is provided. The signal will only be injected if enable is active and the enable polarity can be selected.



uniMatrix sense and stimulus signals connected to a keyboard matrix



A matrix with address decoder. uniMatrix can interface the address signals directly saving pins

Short push

Some devices can only handle short presses of certain buttons. Longer presses would block block other functionality. An example are the instrument select buttons of many Casio keyboards.

uniMatrix can automatically simulate short presses, even if the midi signal would normally generate a long press, or just an 'on' signal.

The 'short press time' can be adjusted globally and the short press function activated for each key independently.

Midi output and debouncing

uniMatrix also supports MIDI output, turning a keyboard into a midi controller.

Here, an additional debouncing time can be selected. When a button is pressed, its signal typically changes polarity a few times before settling to its new state (bouncing). Debouncing filters this by only detecting a new state if two consecutive readings are identical.
The debouncing time is the reading interval - the time between two readings.

Matrix key configuration

After setting up the matrix parameters in the configurator, the individual keys or buttons of the matrix (now called marix contact) can be freely configured.

Each matrix contact can react to

- MIDI Note
- MIDI CC
- MIDI Program change

A global midi channel can be used for all contacts, or individual channels configured for each contact.

The note number, CC number or program change number can be defined as well.

A velocity/ CC value can be defined:

- Midi note input: Note must have at least this velocity value to turn on the note.
- Midi note output: Sends this velocity. If 0: sends 127
- CC input: received CC must have at least this value to turn on contact.
- CC output: Outputs this CC value when pressed, 0 when released. If 0: outputs 127

Additionally, you can select if the contact should be on/off or of short push type (see above).

Additional inputs and outputs

Pins not used by the keyboard matrix can serve a variety of different functions.

The learn pin

One pin can be configured as 'learn' input. If a button is connected between this pin and GND it can be used to learn a new midi channel for uniPulse.
Press and hold the button for 2 seconds, then send a midi note to uniMatrix. The note's midi channel will be used as uniMatrix's global MIDI input channel.

Note output

Configuring a pin as Note-out will output a logic high signal if a MIDI on message is received and a low signal if a note off message is received.

It is possible to invert this behavior and to active short push mode.

CC output

Configuring a pin as Note-out will output a logic low signal if a MIDI CC value is below 64 and a high signal if above or equal 64.

It is possible to invert this behavior and to active short push mode.

Clock output

Configuring a pin as clock will output a clock derived from an incoming MIDI clock. 16 different clock dividers are available.

Using this feature, uniMatrix can output up to 22 different clocks simultaneously.

The clock can be inverted. Activating the short-push feature will result in a clock-pulse instead of a square wave clock signal.

Start / stop

Configuring a pin as Start / Stop will output a start stop signal derived from an incomcing MIDI clock.

The signal can be inverted.

Note input

When a high signal is detected at this pin, a 'note on' MIDI message is sent. When a low signal is detected, uniMatrix sends a 'note off' message.
This behaviour can be inverted.
Optionally, a pull up resistor can be activated, so a button can be directly connected between the pin and GND

CC input

When a high signal is detected at this pin, a CC message with value 127 is sent. When a low signal is detected, uniMatrix sends a CC message with value 0.
This behaviour can be inverted.
Optionally, a pull up resistor can be activated, so a button can be directly connected between the pin and GND

CV input

Pins 17-22 can be configured as analog inputs. The voltages at these inputs are translated to MIDI CC messages.

This can be used to send a knob's position back via MIDI, or to build a midi controller.

PLL output

IO21 can be configured to output a multiplied MIDI clock.
The multiplier factor can be adjusted.
See Clock multiplier PLL output.

CV outputs

Two additional CV outputs are available that can be used to control filters or similar parameters.

These are not suitable for pitch control.

The outputs can be mapped to

- MIDI controller / CC
- MIDI velocity (of the last received note)
- MIDI Aftertouch (of the last received note)
- MIDI Channel pressure

The CV outputs can also be inverted

Clock multiplier PLL output

Some synthesizer need a clock that is faster than the MIDI clock in order to be synced properly. IO21 can be configured to output a clock with a frequency that is a multiple of the MIDI clock. This is achieved using a PLL.

The MIDI clock runs at 24 PPQN, or 3 clocks each 32th note. Using the multiplier, this frequency can further be multiplied by an adjustable factor between 2 and 24.

Please note: Due to an unfortunate hardware limitation, using the clock multiplier will result in a ripple on the CV-outputs with a frequency of 612Hz. The CV-outputs can still be used, but need additional filtering to function properly.

Powering uniMatrix

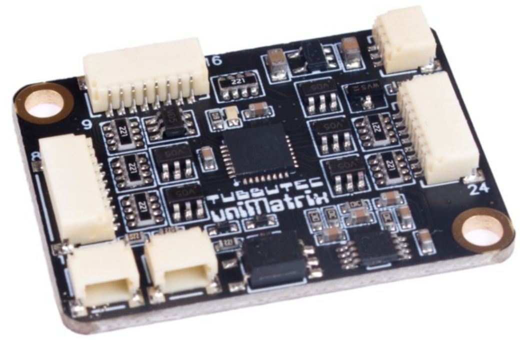
uniMatrix needs a single supply voltage between 3V and 12V. Current draw is about 18mA.

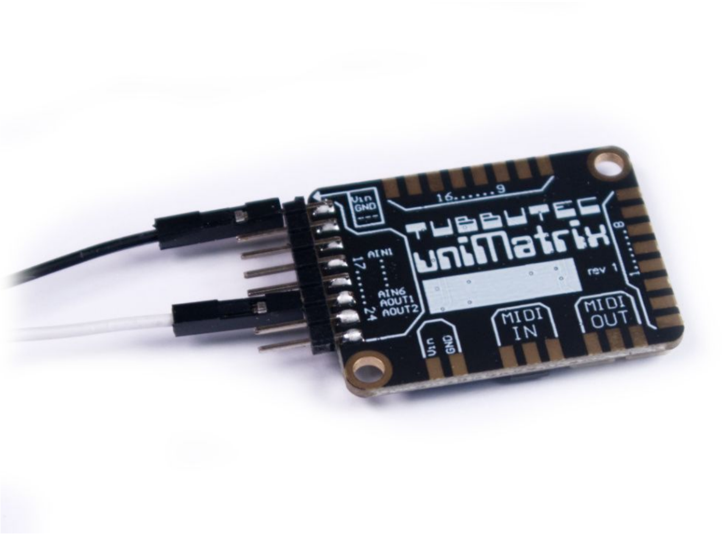
uniMatrix has a built-in voltage regulator. If powered at 5V or higher, it will run at 5V. Signal outputs are 0-5V. When the host devices' signals are greater then 5V, uniMatrix may need additional circuitry.

While inputs and outputs are protected against over voltage, the protection diodes will clap higher voltages to 5V. So it is not possible to directly connect an uniMatrix input to a 12V signal. Additional series resistors will be needed.

When powered from a supply between 3V - 5V, signal inputs and outputs are restricted to that supply voltage.

Electrical connections

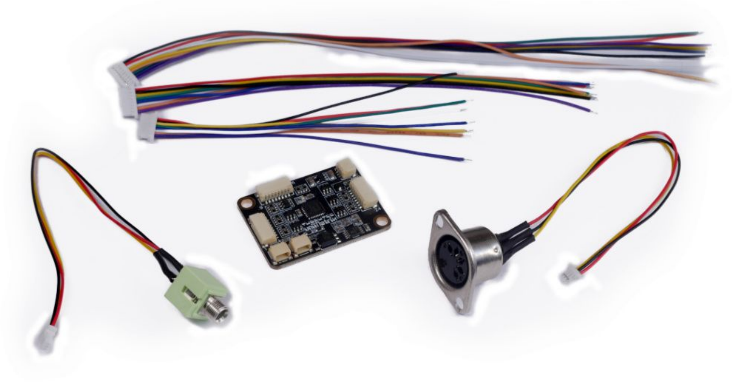




There are multiple ways to connect uniMatrix to a host device. Depending on the device, one of these might be the most preferred way.

UniMatrix features JST SH (1mm pitch) connectors on the top. We can supply prefabricated wires with various lengths. These can then be soldered to the nessesary points in the host device in order to get power and connect signals. MIDI sockets can also be connected this way.

Alternatively, one can use the solder pads on the back. These can be used to solder wires directly, or to install pin headers and use Dupont or jumper wires.



Mechanical

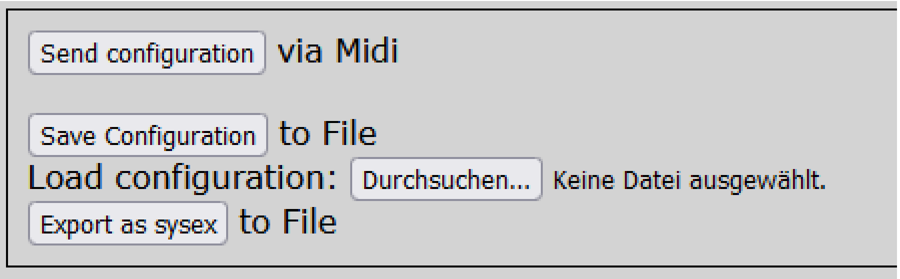
The board’s dimensions are 37x28mm (1.5×1.1 in) with a height 5mm (0.2in)
Two mounting holes are provided to fix the board inside a device. Alternatively, double sided tape can be used easily, as there are no components on the underside.

The configurator

The uniMatrix configurator comes in the form of a web-based application running in your browser.
In order to work directly, your browser needs to support web-MIDI, which most do.
Alternatively, a sysex file can be generated and sent to uniMatrix with 3rd parts tools.

The configurator can be found at
https://tubbutec.de/files/uniMatrix/uniMatrix-configurator.html
It is possible to download this filer and use it without an internet connection.

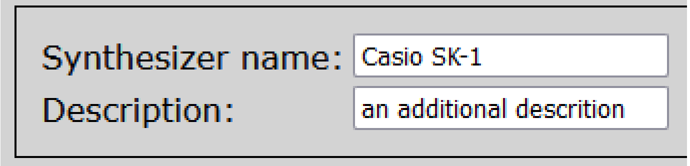
Loading, Saving, Sending



Keine Datei ausgewählt.

- Send configuration via MIDI: Send the current configuration to uniMatrix via MIDI sysex
- Save configuration to file: Save the configuration to a file in order to load it later. The file name is based on the synthesizer name (see below).
- Load configuration: Load a previously saved configuration
- Export as sysex: Export configuration to a sysex file for sending via a 3rd party tool

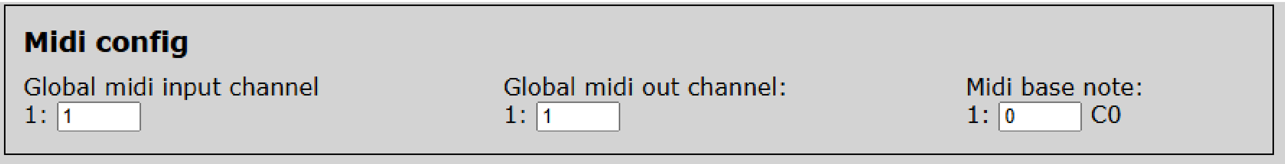
Name and description



Synthesizer name: Casio SK-1
Description: an additional descrition

Synthesizer name and an additional description

MIDI config



Midi config

Global midi input channel
1: 1

Global midi out channel:
1: 1

Midi base note:
1: 0 C0

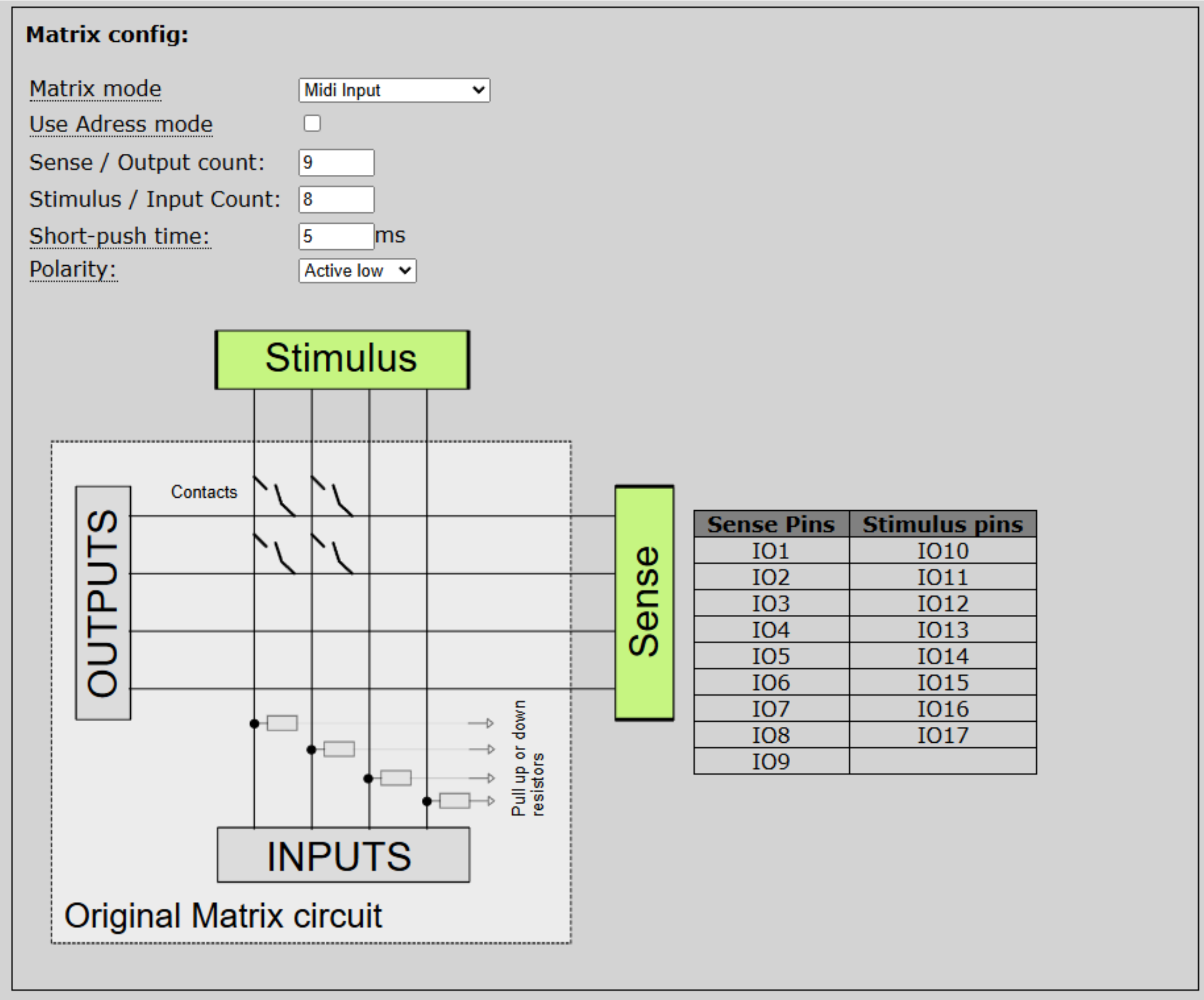
- Global midi input channel: MIDI channel to receive data unless specified otherwise at an individual contact level.
- Global midi output channel: MIDI channel to send data unless specified otherwise at an individual contact level.
- Midi base note: This base note get added to the notes configured in the matrix key configuration. It is the lowest note sent.

Matrix config

Here, the matrix can be configured. Select the type of matrix, input and output count, polarity and additional parameters. All parameters are explained in chapter Interfacing keyboard matrixes and a tooltip appears when hovering over the parameter names.

The configurator provides a list of pins and how to connect them based on the matrix dimensions.

If you do not want to interface a matrix, select matrix mode ‘OFF’.



Matrix config:

Matrix mode: Midi Input

Use Adress mode: □

Sense / Output count: 9

Stimulus / Input Count: 8

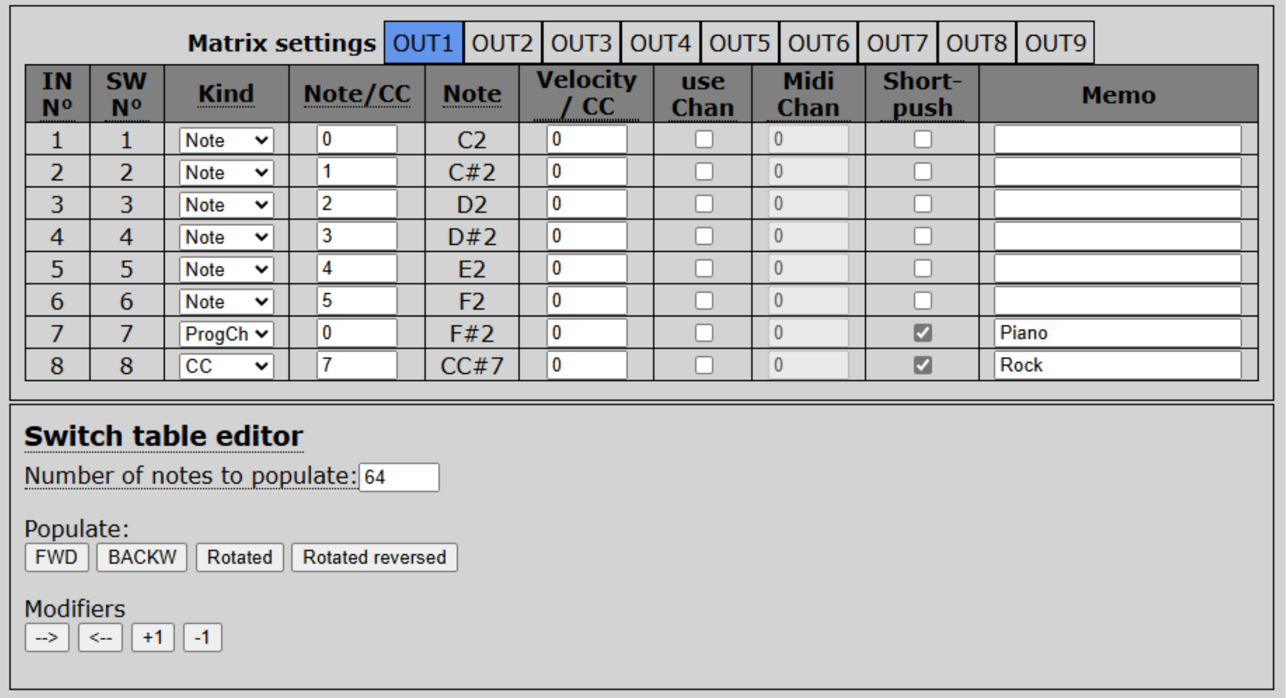
Short-push time: 5 ms

Polarity: Active low

Sense Pins	Stimulus pins
IO1	IO10
IO2	IO11
IO3	IO12
IO4	IO13
IO5	IO14
IO6	IO15
IO7	IO16
IO8	IO17
IO9	

Matrix key configuration

Each matrix contact can be configured individually. The parameters are explained in chapter Matrix key configuration.



Matrix settings | OUT1 | OUT2 | OUT3 | OUT4 | OUT5 | OUT6 | OUT7 | OUT8 | OUT9

IN Nº	SW Nº	Kind	Note/CC	Note	Velocity / CC	use Chan	Midi Chan	Short-push	Memo
1	1	Note	0	C2	0	□	0	□	
2	2	Note	1	C#2	0	□	0	□	
3	3	Note	2	D2	0	□	0	□	
4	4	Note	3	D#2	0	□	0	□	
5	5	Note	4	E2	0	□	0	□	
6	6	Note	5	F2	0	□	0	□	
7	7	ProgCh	0	F#2	0	□	0	☑	Piano
8	8	CC	7	CC#7	0	□	0	☑	Rock

Switch table editor

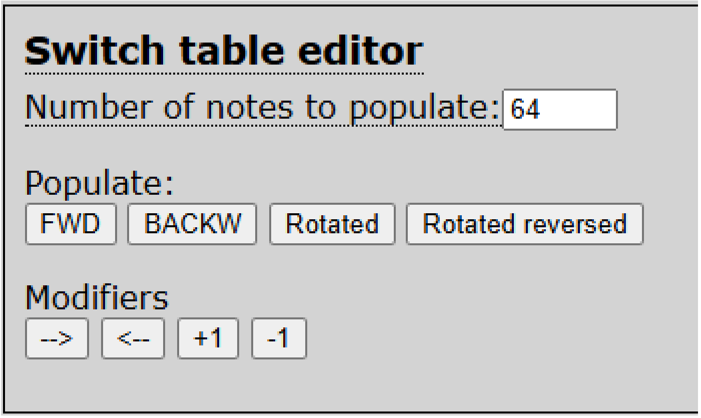
Number of notes to populate: 64

Populate:

FWD | BACKW | Rotated | Rotated reversed

Modifiers

--> | <-- | +1 | -1



Switch table editor

Number of notes to populate: 64

Populate:

FWD | BACKW | Rotated | Rotated reversed

Modifiers

--> | <-- | +1 | -1

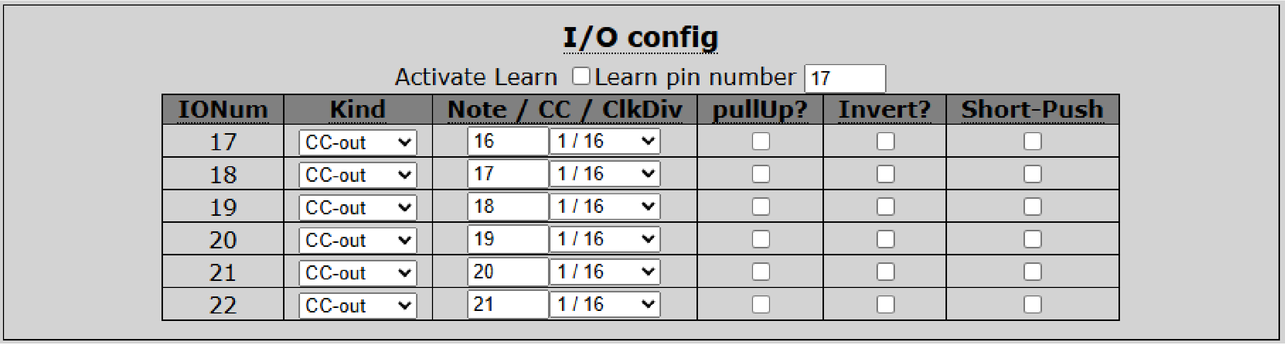
Switch table editor

Some tools are provided to quickly populate the switch table with MIDI notes.

Type in the number of notes and press one of the 4 populate options.

The modifiers can be used to shift, increment or decrement existing entries.

I/O and config



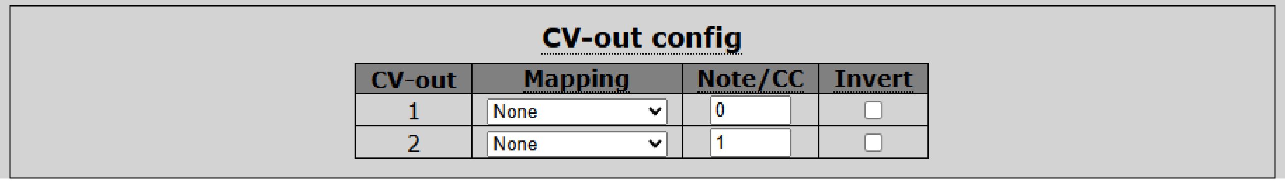
I/O config

Activate Learn □Learn pin number 17

IONum	Kind	Note / CC / ClkDiv		pullUp?	Invert?	Short-Push
17	CC-out	16	1 / 16	□	□	□
18	CC-out	17	1 / 16	□	□	□
19	CC-out	18	1 / 16	□	□	□
20	CC-out	19	1 / 16	□	□	□
21	CC-out	20	1 / 16	□	□	□
22	CC-out	21	1 / 16	□	□	□

Can be used to configure all pins not used by the matrix. Options are explained in chapter Additional inputs and outputs

CV-out config



CV-out config

CV-out	Mapping	Note/CC	Invert
1	None	0	□
2	None	1	□

Here, you can configure the two CV-ouputs provided by uniMatrix. Options are explained in chapter CV outputs.