

TubeOhm  
AMBIKA-C MIDI Controller

Manual english  
V 1.0



## About the AMBIKA MIDI controller:

The MIDI controller for the AMBIKA DIY synthesizer by Mutable Instruments is a TOOL for quick sound programming. No more and no less.

All sound giving parameters for controlling the filter, VCO'S, VCA, LFO'S and the modulation matrix are directly controllable on the main page and send the exact numerical values via NRPN direct to AMBIKA.

A sophisticated random function allows individual parameters by a 'controlled random' to change.

With this, new sounds are programmed in seconds.

The controller automatically transmits these sounds in the AMBIKA, these can be in the AMBIKA be saved.

We speak here only once exclusively on the sound parameters.

Unfortunately some VST host does not support some MIDI NRPN parameter above 128 or send all MIDI data of all different MIDI channels on one (all) MIDI channel .

Thus, the sequencer section features 6 sequencers was programmed independently of the internal AMBIKA sequencer. Data of the controller sequencer so are not in the AMBIKA transmitted.

Furthermore, global parameters such as can Voices will launch their own MIDI channels, not be controlled because Mutable Instruments has made this not controllable.

**Attention to allow the AMBIKA can also be controlled via the controller checked before that the MIDI Filter is turned off in AMBIKA!**

The MIDI NRPN transmission:

Midi works with 31 K bit.  $31K \text{ bits} / 8 = 3875 \text{ byte}$ . A NRPN parameter contains  $4 \times 7$  bits.

The controller sends 106 parameters, each with  $4 \times 7$  bit. It will be so with each program change

approximately 424 bytes (7). And it takes a little. The time that the controller needs to ensure all the data to send is 0.3 .. 0.5 seconds.

So wait a bit after the program change and not wildly step through the programs .

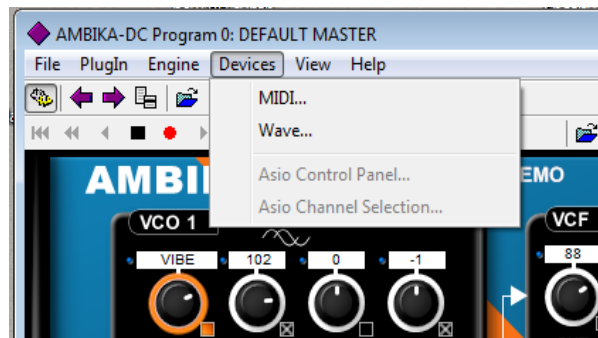
TubeOhm 28.05.2014

Here we go.

The AMBIKA controller works standalone with SAVIHost by Hermann Seib.  
 But it is also possible to supply the controller with any other standalone host program  
 B. z. Cantabile 32 bit + to operate 4free.  
 The Kontroller.DLL is designed as a VST and can be eg: Cubase 32 or 64 (with J-Bridge)  
 as well as in Ableton and FL operate as a VST.

Let us now therefore to the basic settings of stand alone version.

The setting of the MIDI and AUDIO devices



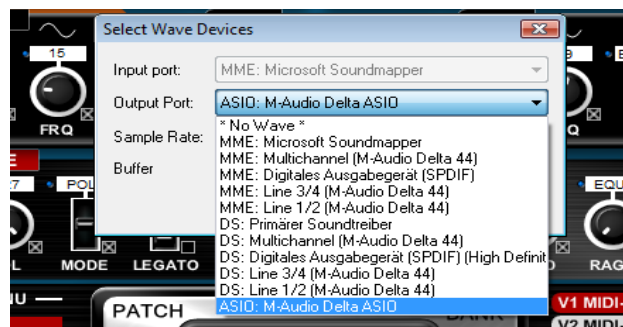
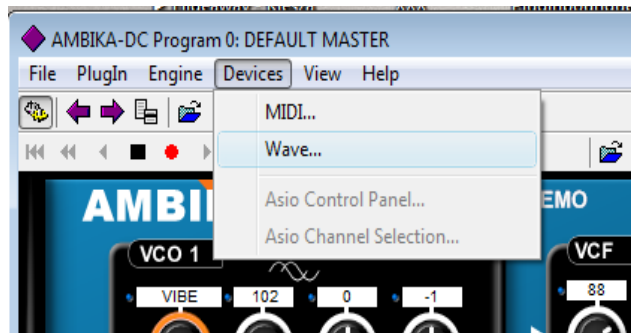
First, the MIDI IN port is set.



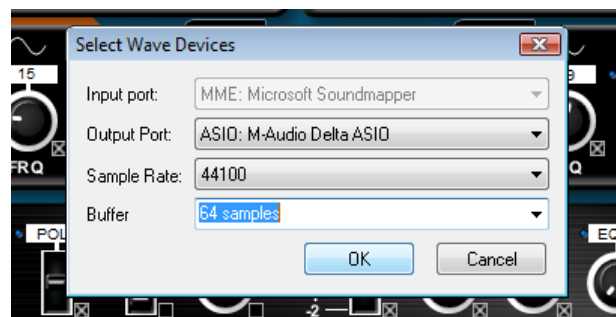
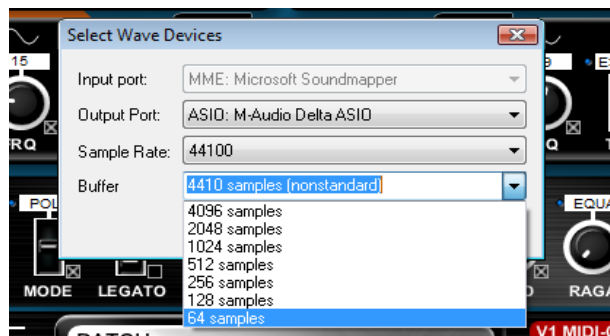
Then the MIDI OUT port



Set up your ASIO Sound card with less latency.

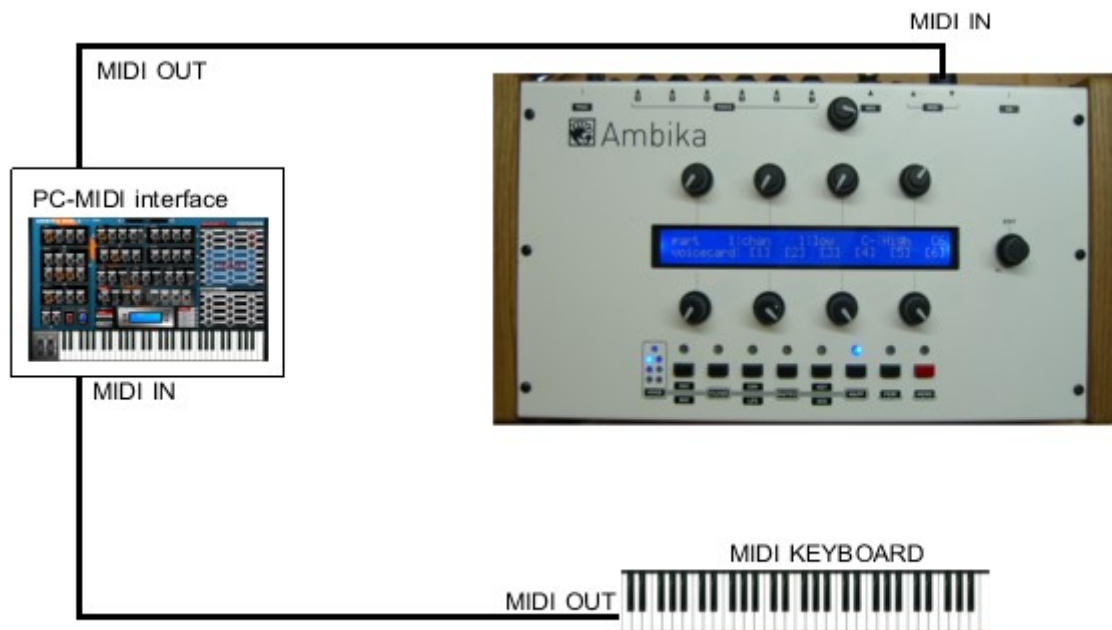


Since this is not a sound generator, the latency as small as possible be selected.



Now, the controller should receive and send MIDI signals .

## MIDI flow



### DEMO limitations

the DEMO works fully but

1 :) some random switches are off

2 :) the first stepper works complete with 16 steps, the second with 8, and the stepper 3...6 each with four Steps. (In the final version, all stepper work with 16 steps)

3 :) in the modulations matrix only the modulators are unlocked 1...2 and 13...14.

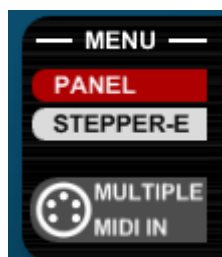
### The main panel

The main pannel contains all sound giving parameters for the AMBIKA.

VCO, VCF, LFO's and the 14-fold modulation matrix.

The STEPPER-E switch leads to the stepper Pannel.

If multiple MIDI IN on, so, the individual voices of the AMBIKA are controlled via a MIDI keyboard . (MIDI Channel 1-XX lower, MIDI channel 2-XX upper)



## Working with the random function



A special feature is the RANDOM FUNCTION. Every function button of the controller, the random function can be assigned. This is indicated by an orange / red ring around the button and can be switched on/off by the small rectangular switch .

Due to the large RANDOM START button each labeled button is now assigned a random value and transferred to AMBIKA. Depending on which buttons are provided with Randomize obtained variations of a sound or completely new sound sculptures.

## The refresh button

The refresh button to send all values of the controller with the specified MIDI channel manually to AMBIKA. For example, if the AMBIKA was turned on later , you can send all parameters manually to AMBIKA.

## Working with different voices on different MIDI channels



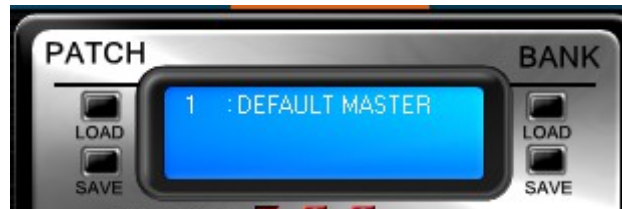
AMBIKA can put all 6 voices to different MIDI channels. In the picture above VOICE 1 is active on MIDI channel 1. All data of the controller are sent on MIDI channel 1.

If you want to, for example, Edit Voice 2 to MIDI channel 2, so just enclose V2 active and

backed by a mouse click on the midi channel fixed this.

The MIDI ACTIVE switch will not be saved with a patch. Thus it is possible e.g. load a sound to MIDI channel 1, then switch to MIDI channel 2 and back to load a different sound there.

## Saving a sound Patch



Individual sound patches can be saved using SAVE / LOAD patch.

To do this award only one NAME by clicking on the font in the display.

The sound program is saved with the name used as an FXP file on the computer.

A bank contains 128 patches and sound is saved using SAVE / LOAD BANK.

Basically, a PATCH contains all controller data, including settings for the Step Sequencer.

## The Step Sequencer

The AMBIKA CONTROLLER includes 6 step sequencer with 16 steps.

These can be assigned to different MIDI IN and OUT channels.

The timing can be set separately for each step sequencer.

But there is also a BPM-ALL function which all sequencers the same TEMPO assigns.

All sequencer to be started at the same time by pressing any button of the MIDI KEYBOARD.

The sequence is automatically adjusted to the pitch played.



From left to right: Stepper ON, the individual stepper switches active

Step1 ... 6 switches by the stepper and allows the settings.

Semitone transpose the complete sequence.

Start defines the START-STEP, Stop the STOP STEP Steps = 1, each step is played, Steps = 2

the 1,3,5 ... quilting is played. Steps = 3, the 1,4,7, etc is played

Direction indicates the direction of the stepper.

MIDI IN is selected from which MIDI channel (all, 1 .. 16) of the stepper gets his pitch

information.

MIDI OUT determines which MIDI channel the stepper sends the data.

To create a sequence, you can use the small MIDI keyboard. (T) on the upper left side holds a note and allows for adjustment of the sequence without always pressing a button.

### Other features of the stepper:



'Active' turns on the STEPPER >>> when the MIDI ROUTING is set correctly!

More on that later.

'Free / Key' is turned on, so the stepper starts with a MIDI keyboard with the selected step.

Tip and Trick. Should order to get a sequence when you transpose with the keyboard always in the timing of the next note you play while the property prev NOTE still being held!

Thus, one can transpose the sequence within the STEPS. If you play only a new note as the stepper always starts with the selected START STEP on.

'Free' can just run the stepper. There is no transposition possible.

### The BPM setting:

Basically, the BPM setting can be done in three ways.

1 :) separately for each stepper

2 :) globally if the BPM ALL button is pressed all the steppers have the same tempo

3 :) externally, then the BPM of the VST host program is adopted.

About a numerical input the tempo divider or multiplier can be adjusted.

Example 120 BPM, div 2, the steppers running at 60 BPM.

Each stepper can have its own divider / multiplier.

SWING delayed every 2nd note



## The Velocity / Gate RANDOM function:

Thus, the stepper not as static sounds include all sequencer still a random function for the gate length and the velocity (VELO)

About OFFSET doing the basic value is set, strange represents a strength of the random signal from OFFSET to MAXIMUM.

example:

1 :) OFFSET is 0.5 STRANGE is 1 are random values from 0.5 .. 1 generates

2 :) OFFSET is 0 Strange is 1, there are random values generated from 0 to 1

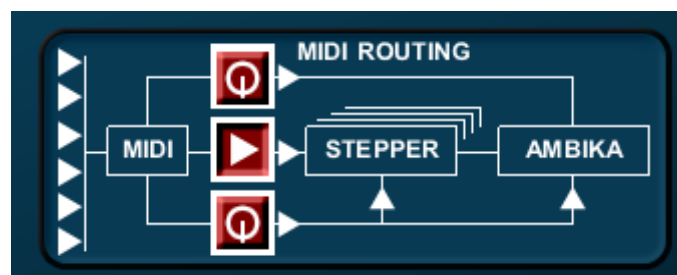
3 :) OFFSET is 0 Strange is 0.5, there are random values generated by 0 .0.5.

Need to hear this effect the AMBIKA Velo must be routed in the MOD MATRIX.  
Should listen to the gate Lenge the VCA release time should be very short!

## The most important conclusion:

### The MIDI ROUTING!

To get the steppers work, the MIDI routing must be set correct.



Three functions are possible. From top to bottom

1 :) the controller sends its data directly to AMBIKA

2 :) the controller sends its data directly to the stepper. All steppers are now switched on.

3 :) the controller sends its data to the stepper and simultaneously to AMBIKA

\*1:) all Stepper are switched off

\*2:) all Stepper are switched on ( if they are set active)

\*3:) all Stepper switched on AND you can play on a separate Midi chanel the Ambika without the stepper . A lead sound for example.

OK, thats it . I hope you will have fun with the AMBIKA-C controller

TubeOhm

28.05.2014