

under http://www.midibox.org/mios_studio and http://www.midibox.org/mios_studio/apps/miosstudio/gui/help/index.html

MIOS Studio was started by Wilba in November 2003, and later taken over by Adam King.

It is a java-based, platform-independent MIDI processing environment, which not only provides upload and debug functions for MIOS, but also advanced features like MIDI Port Routing, Filtering and a virtual Keyboard. It's written in a modular way and will be published under GPL later so that other programmers can make their contributions.

Adam hasn't finished all of his plans yet, therefore only a precompiled binary package is available. It can be downloaded from: http://www.midibox.org/mios_studio

He is interested in your inputs - how is it working, are there any problems, does the user interface need improvements, which features could be useful?

Installing MIOS Studio

To run MIOS Studio you will need the Java Runtime, at least version 1.5 installed. If you do not have it, you will need to go to the Sun site http://java.sun.com/j2se/1.5.0/download.jsp first, download the JRE 5.0 setup file and install it according to your platform's instructions.

Once Java is installed, all you will need to do is download MIOS Studio and save it to your PC. In Windows, you should be able to just double click on the .jar file to open the program. To start from a command line (or setup a shortcut), you can use the command

java -jar <path_to_jar_file>/MIOSStudio_beta4.jar

to open MIOS Studio.

As an added utility, a cutdown version with just a keyboard controller, MIDI routing/filtering and keyboard zone mapping is available. This can be started with the command

```
java -cp <path_to_jar_file>/MI0SStudio_beta4.jar
org.midibox.apps.virtualkeyboard.~VirtualKeyboard
```

So far the program has been tested with Windows 98, 2000 and XP. In theory, it should also run on Linux and Mac but the old saying proves true, "write once, debug everywhere". Linux MIDI support is supposed to be working, but I have had many problems - possibly due to an older kernel and ALSA drivers. If anyone can test on a newer version of Linux (with Java 1.5 installed) it would be greatly appreciated. At the moment, Java 1.5 is not officially available for the Mac, but if there are any Java/Mac developers in the community, please test it and report you findings.

I have been unsuccessful so far in trying to get MIOS Studio to recognize any actual MIDI inputs/outputs using OS X 10.4.6. The UI works fine, however... -nebula

Linux Installation Instructions

For Debian and distributions based on it, such as Ubuntu; you need to install three packages.

sudo apt-get install sun-java5-bin sun-java5-jre libgcj7-awt

Once this is done, you can start MIOS Studio by typing:

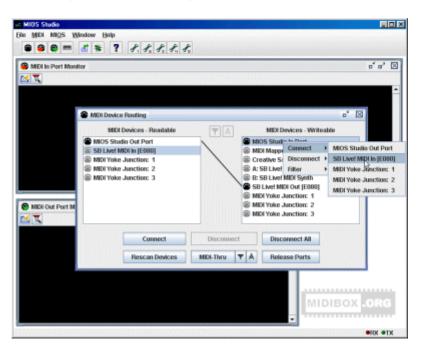
```
java -jar MIOSStudio(version).jar
```

Using MIOS Studio

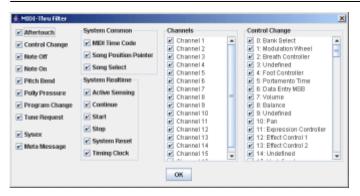
Below are examples of some of the main features of MIOS Studio. For further details, click on the links to go to the relevent section of the MIOS Studio Help File.

MIDI Device Routing: it allows you to forward and filter MIDI streams to any MIDI port. At the top of the list you will always find the MIOS Studio In and Out port. This is the port to which MIDI data has to be routed when you want to upload code, monitor it, send debug commands, etc...

The readable port (MIDI IN of your interface) has to be routed to the MIOS Studio In Port, and the writable port (MIDI OUT of your interface) has to be routed to the MIOS Studio OUT Port:



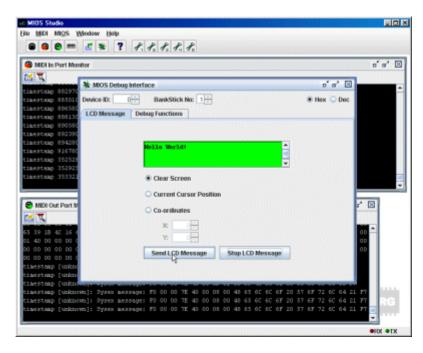
MIDI Filter: It's possible to filter MIDI events:



Hex Upload: The MIOS code upload is very comfortable - you don't need to generate a .syx anymore, HEX files can be uploaded directly:

n Port Monitor	
g 61479000 us: Sysex message: P0 00 00 72	40 00 07 70 F7
g Upleads	
Upload 1 Upload 2 Upload 3	
HEX File	
Browse G:WidiBarlmidibar64e_v2_1	Imain.hex
Start Stop	Upload Options Device ID: 0 BankSlick Nat 1
Sending block 00003000-000030FF Received Checksum: 02 - 0K Sending block 00003100-000031FF Received Checksum: 60 - 005	Wait for upload request before starting upload Bus feedback from core Don't use feedback from core
Semiing block 00003200-000032FF Received Checksum: 4D - 0K Semiing block 0000300-000033FF Received Checksum: 51 - 0K Semiing block 0000340-000034FF Received Checksum: 70 - 0K	Defay between SysEx messages (ms): 310

MIOS Debug LCD: LCD messages can be generated:



MIOS Debug Functions: The MIOS debug window allows to execute MIOS functions via remote:

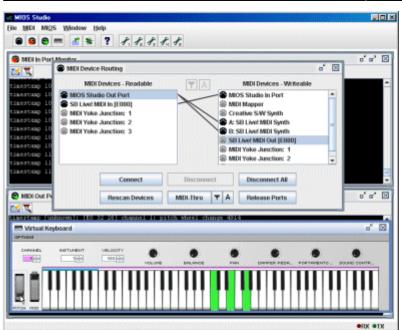
MIOS Debug Interface evice ID: 0 - 8	anikStick No: 1	-		S-900	e ^r i ⊛ Hex (
	unctions				
Start C Stop * Function Builder		SHOARE HER	Delay (ms		SRAM Read Address:
Function MIOS DOUT PinSet	W	REG MIOS_P	NR. MIOS_PAR.	MIOS_PAR.	No. Bytes:
MIOS_DOUT_PinSet	1	0			SRAM Write
MIOS_DOUT_PinSet	2				Address: Data:
Return Values					
	AM1: 01 MI	03_FARAM2: 00 05_FARAM2: 00 03_FARAM2: 00	MIOS_PARA	M3: 00	

It's also possible to read and write the memory.

Run External Commands: Buttons on the toolbar can be customised to run external commands:



MIDI Keyboard Controller: A virtual keyboard controller is also available:



Virtual Keyboard: A smaller standalone program is also available with just the features of the keyboard controller and MIDI Routing/Mapping:

MDI Desire	s - Readable		MDUDasics	rs - Writeable	
Virtual Keyboard Out Por		T A	Virtual Keyboard in Port		
SB Live! MIDI In [E080]			MIDI Mapper		
MIDI Yoke Junction: 1			Creative S/W Synth		
MDI Yoke Junction: 2		11.8	A: SB Live! MIDI Synth		
MIDI Yoke Junction: 3			B: SB Live! MIDI Synth		
		` `	SB Live! MIDI Out [E080]		
			MIDI Yoke Junction: 1 MIDI Yoke Junction: 2		
			NEX YORE SUPERIOR. 2		-
	Connect	Disconnect	Disconnect All		
	Rescan Devices	MDI-Thru 🔻	A Release Ports		
HWHEL BETUNENT	VELOCITY	MDI-Thru	A Release Ports	•	
		MDI-Thru T			O CONTR.
		•		e: PORTARENTO - ST	
		•			•

From: http://wiki.midibox.org/ - **MIDIbox**

Permanent link: http://wiki.midibox.org/doku.php?id=mios_studio&rev=1207391489

Last update: 2008/04/05 10:50

