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://69.56.171.55/~midibox/mios\_studio/MIOSStudio\_beta4.jar

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///MIOSStudio\_beta4.jar org.midibox.apps.virtualkeyboard.~VirtualKeyboard

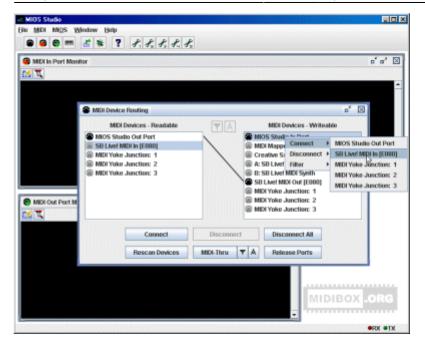
So far the program has been tested with Windows 98, 2000 and XP. In theory, it should also run on Linux and Mac but the old saying proves true, "write once, debug everywhere". Linux MIDI support is supposed to be working, but I have had many problems - possibly due to an older kernel and ALSA drivers. 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.

## **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...

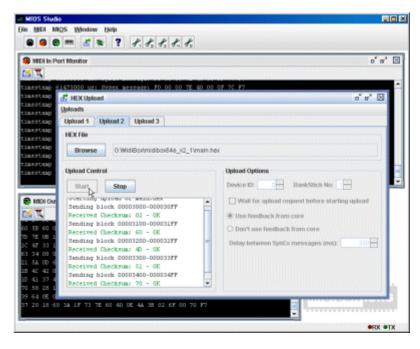
The readable port (MIDI IN of your interface) has to be routed to the MIOS Studio In Port, and the writable port (MIDI OUT of your interface) has to be routed to the MIOS Studio OUT Port:



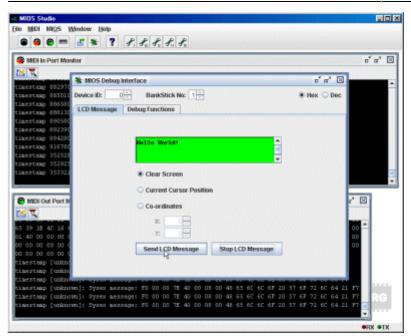
MIDI Filter: It's possible to filter MIDI events:

✓ Aftertouch	System Common	Channels		Control Change	
Control Change	MIDI Time Code	Channel 1	-	D: Bank Select	^
Note Off	🗹 Song Position Pointer	Channel 3		2 Breath Controller	
✓ Note On	✓ Song Select	Channel 4		3: Undefined     4: Foot Controller	
Pitch Bend	System Realtime	Channel 6		🕑 5: Portamento Time	
Polly Pressure	Active Sensing	Channel 7 Channel B	-		
🕑 Program Change	Continue	🕑 Channel 9		R Balance	
🗹 Tune Request	🕑 Start	Channel 10 Channel 11		Interfined Interfined Interfined	
V Sysex	✓ Stop	Channel 12		11: Expression Controller	
Meta Message	🗾 System Reset	Channel 13 Channel 14		<ul> <li>12: Effect Control 1</li> <li>13: Effect Control 2</li> </ul>	
	Timing Clock	Channel 15	*	14: Undefined	*

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



MIOS Debug LCD: LCD messages can be generated:



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

3/5

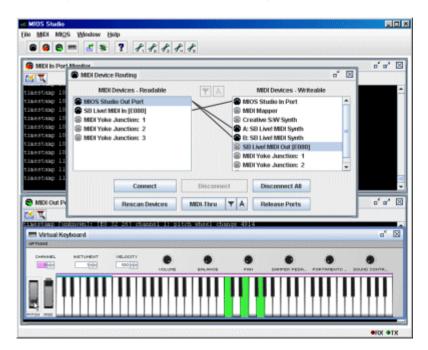
MIOS Debu	a Interface 0 + BankStick	No: 1÷			065555	ອ້ອ້ ⊛ Hex ⊂ t
LCD Message				000	<u> 1000</u>	
Function Built	Stop   Function der		SRAM Read	Delay (ms):	300	SRAM Read
NIOS DOUT	Function	WREG	MIOS_PAR.	MIDS_PAR. MID	PAR.	No. Bytes:
MIOS_DOUT		1	0			SRAM Write
MIOS_DOUT	_PinSet	2			Į.	Address: Data:
Return Value	unassumas					
WREG: 01 MREG: FD WREG: FB Done	NIOS_PARANI: 00 NIOS_PARANI: 01 NIOS_PARANI: 02	MIOS_PA	RAM2: 00 RAM2: 00 RAM2: 00	NIOS_PARAND: NIOS_PARAND: NIOS_PARAND:	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:

Virtual Keyboard				
ornoka	822828228282888888888888888888888888888			
MIDED	rvices - Readable	▼ ▲	MDI Devices - Writea	bie
SVirtual Keyboard Ou	t Port	9 VI	tual Keyboard in Port	-
SB Live! MIDI In [E08	0]		DI Mapper	
MDI Yoke Junction:			eative S/W Synth	
MDI Yoke Junction:	-		SB Live! MIDI Synth	-
MDI Yoke Junction:	3		SB Live! MIDI Synth	
			Live! MIDI Out (E080)	
			DI Yoke Junction: 1	
		@ MI	DI Yoke Junction: 2	*
	Connect	Disconnect	Disconnect All	
	Rescan Devices	MIDI-Thru TA	Release Ports	
CHPANEL NETLINENT				
	NOG HOLUN	E BRANKE FRA	DAMPER PEDA. PORTABELT	0 SOLAD CONTR
		••!	•••	**!***!
TOH 1900				الملطعا ملما

From: http://www.midibox.org/dokuwiki/ - **MIDIbox** 

Permanent link: http://www.midibox.org/dokuwiki/doku.php?id=miosstudio&rev=1142538080

Last update: 2006/10/15 09:35

