MIOS Studio

The original documentation can be found under MIOS Studio

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 the MIOS Studio beta7 forum thread.

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?

Installation Instructions

So far the program has been tested with Windows 98, 2000 and XP. In theory, it should also run on Vista, Linux and Mac but the old saying proves true, “write once, debug everywhere”.

Windows Installation Instructions

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.

Once Java is installed, all you will need to do is download MIOS Studio from the MIOS Studio beta7 forum thread and save it to your PC. Always use the latest version! Extract the zip file, and put the .jar file somewhere useful.

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_beta7_5.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_beta7_5.jar
org.midibox.apps.virtualkeyboard.~VirtualKeyboard
```

Mac Installation Instructions

MIOS Studio 7.5 Beta functions correctly under Mac OS X 10.4 (and, likely, onward).

You will first need to install either *Plumstone* (PPC only) or *Mandolane.* Either can be downloaded for free from Mandolane Limited: http://www.mandolane.co.uk/software.html (This software requires a manual install, so be sure to follow included instructions)

The “official” MIOS Studio 7.x thread is here: http://www.midibox.org/forum/index.php/topic,7238.0.html - Simply download MIOS Studio, copy the “jar” file to your applications folder, and double-click it to run! Please read the entire thread, especially if you have problems.

If you intend to continue to use Mandolane, don't forget to pay for it! It's the only way to bring MIOS Studio to your Mac, therefore it just saved your bacon! The cost is only £5, payable via PayPal.

Also don't forget to edit this page if you discover compatibility issues, or have anything to add.

Linux Installation Instructions

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.

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
```

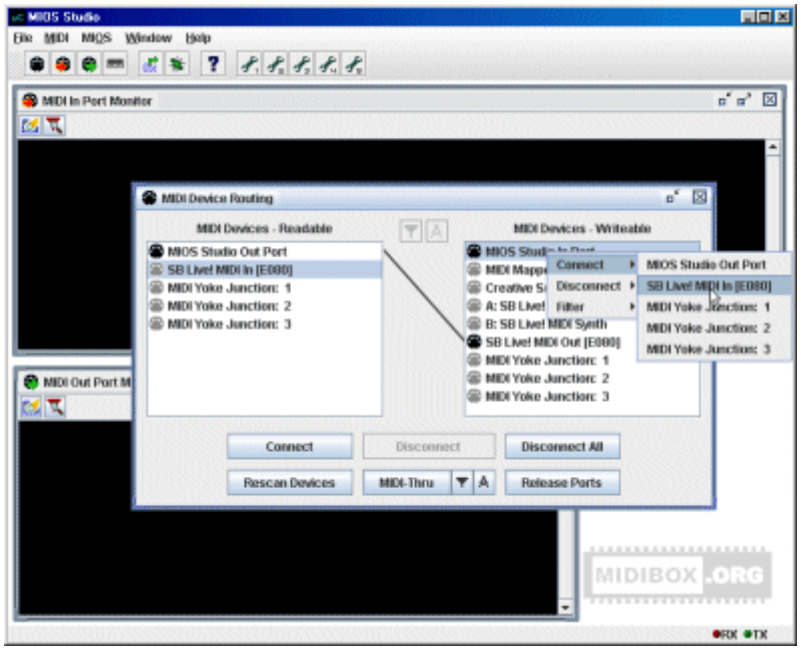
Using MIOS Studio

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 Routing

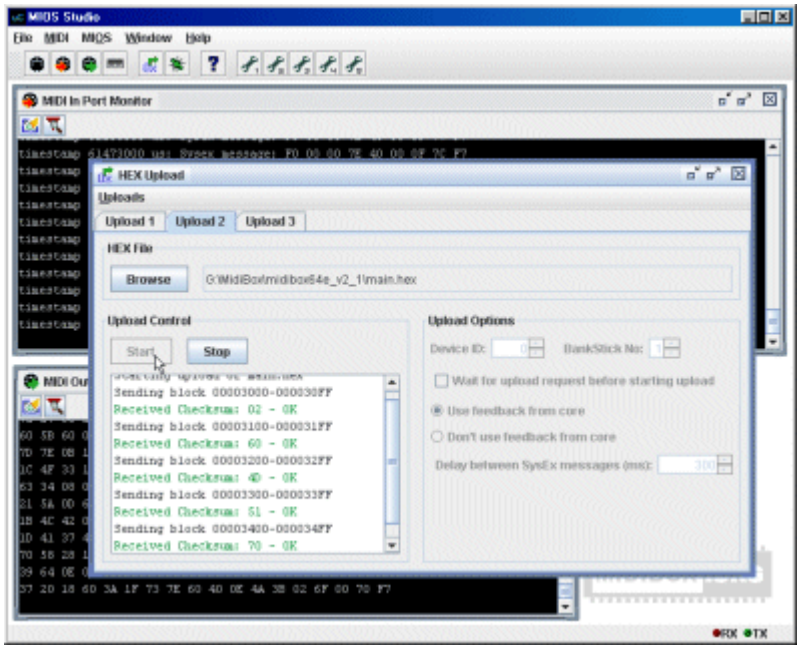
MIDI Device Routing: it allows you to forward and filter MIDI streams to any MIDI port. At the top of the list you will always find the MIOS Studio In and Out port. This is the port to which MIDI data has to be routed when you want to upload code, monitor it, send debug commands, etc...

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:



HEX Upload

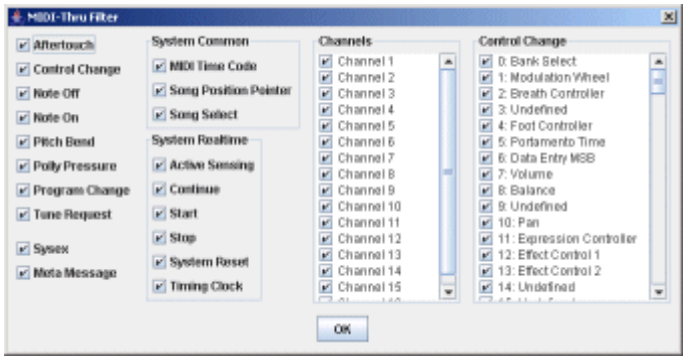
Hex Upload: The MIOS code upload is very comfortable - you don't need to generate a .syx anymore, HEX files can be uploaded directly:



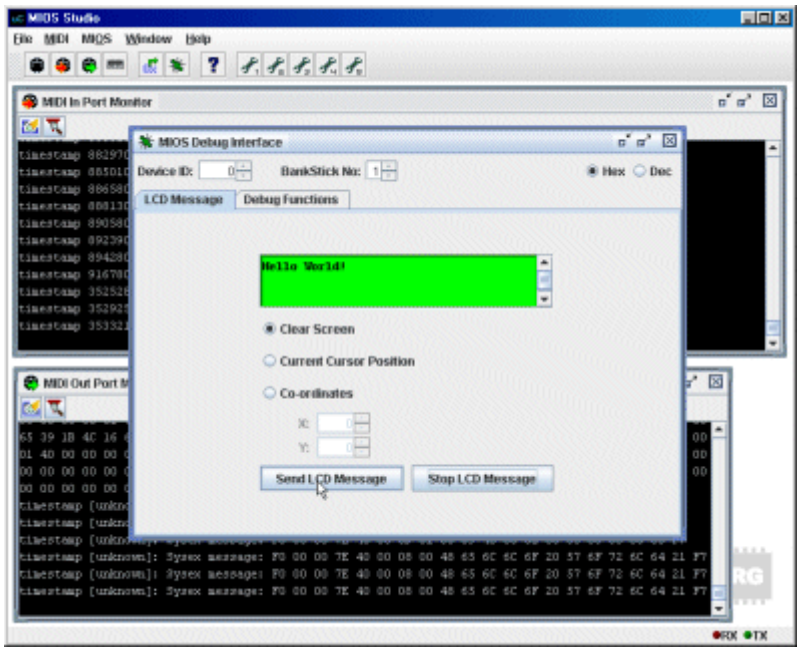
You should **ALWAYS** use 'Smart Mode', and enable 'Wait for Upload Request', and then power on your core after hitting 'Start'. **ALWAYS**. If this doesn't work, there's something wrong, and you should begin troubleshooting as per the uCApps MIDI Troubleshooting page. **ALWAYS**!!! :)

Other Tools

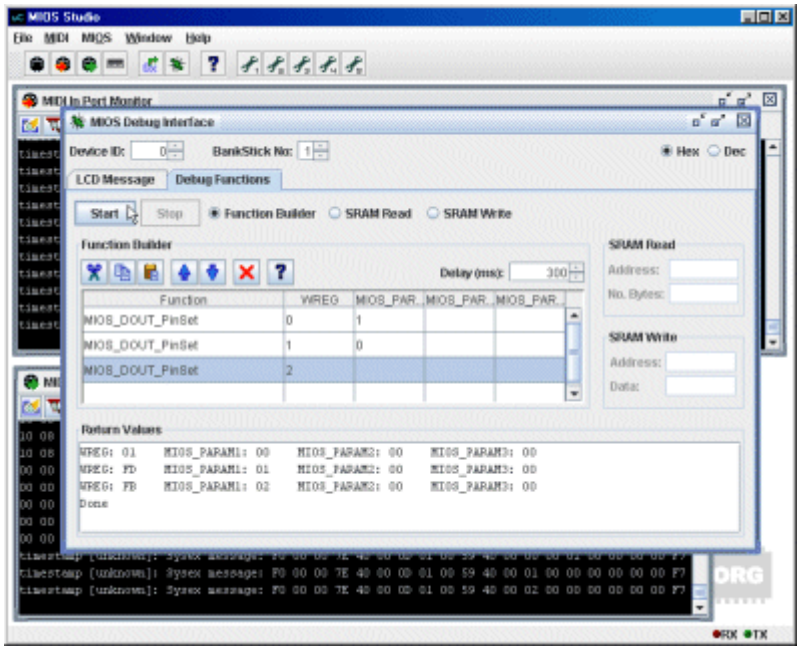
MIDI Filter: It's possible to filter MIDI events:



MIOS Debug LCD: LCD messages can be generated:

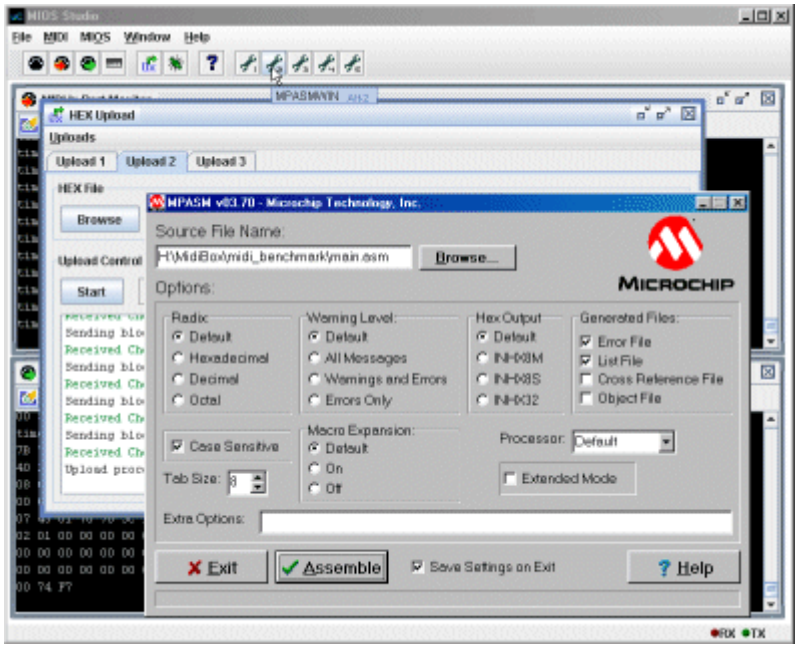


MIOS Debug Functions: The MIOS debug window allows to execute MIOS functions via remote:

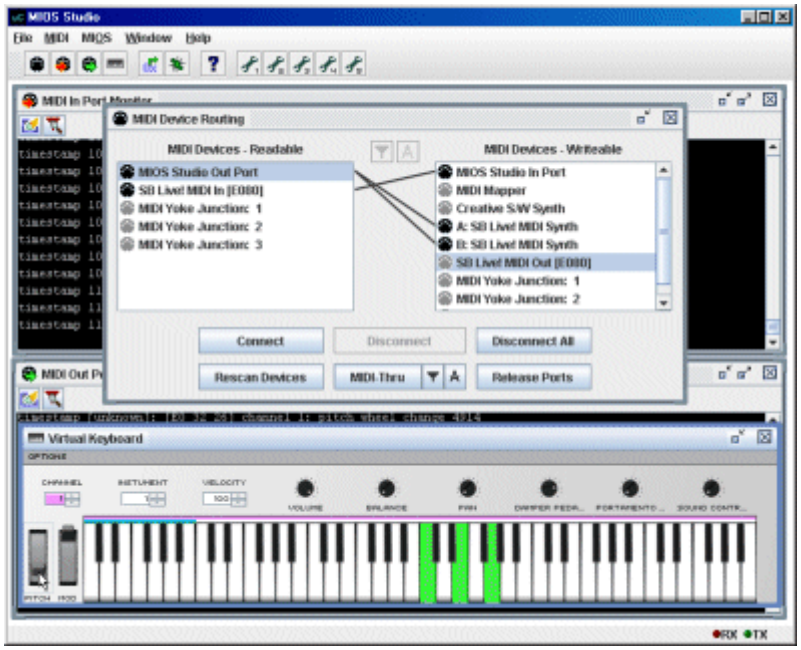


It's also possible to read and write the memory.

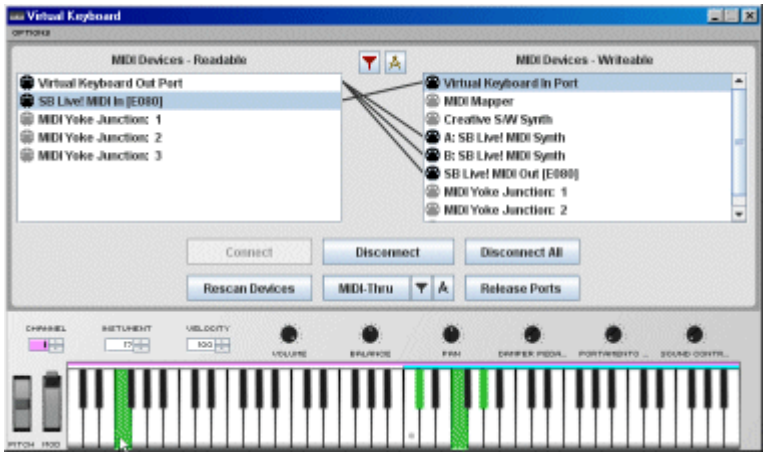
Run External Commands: Buttons on the toolbar can be customised to run external commands:



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:



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

Permanent link:
https://wiki.midibox.org/doku.php?id=mios_studio&rev=1224674531

Last update: **2008/10/22 11:22**