About MIDIbox

What is a MIDIbox

A MIDIbox is an open-source, non-commercial, DIY, hardware MIDI device. It can perform various functions depending on the firmware that is installed and the hardware that is used. The MIDIbox DIY-projects consist of hardware (MBHP) and software (MIOS Operating System + Application). Some projects have special designed modules, others only require a Core module. Every device needs at least one Core.

The normal way to build a MIDIbox MIDI controller or synthesizer is:

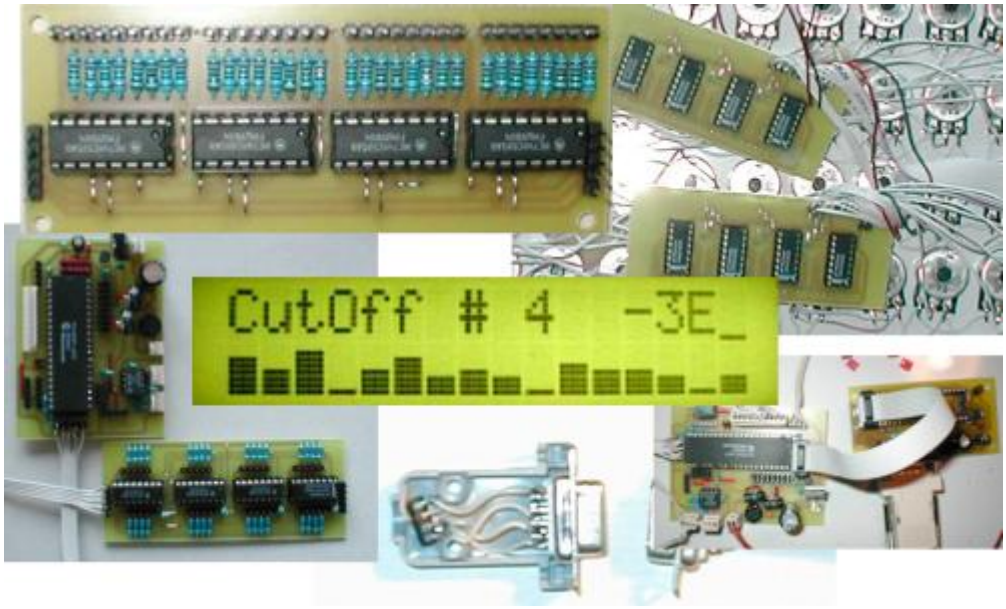
- Build the hardware (starting with a Core Module)
- Build the necessary I/O modules (eg LCD, DIN, DOUT)
- Connect the modules and add additional hardware (knobs, faders, encoders, buttons...)
- Burn the bootloader onto the microchip or get yourself a chip with a preburned bootloader from SmashTV or Mike
- Upload MIOS, the operating system (by MIDI; sometimes MIOS is already on the chip!)
- Upload the necessary application (eg MIDIbox64e.hex) by MIDI

If you have problems understanding the structure of a MIDIbox, just think of your PC:

- Processor ⇒ PIC18F
- Mainboard ⇒ Core Module
- Keyboard ⇒ DIN Module
- Soundcard ⇒ FM (OPL3) or SID Module
- Graphics Card ⇒ DOUT Module (with connected LEDs)
- Display ⇒ LCD Module (Liquid Crysal Display)
- BIOS ⇒ the Bootloader
- Operating Sytem (Windows, Linux, Macintosh) ⇒ MIOS.syx (MIDIbox Operating System)
- The application you work with ⇒ MB64e.syx, MBSID.syx or MBSeq.syx

Luckily - just as in the example - you don't need to open up your case and desolder the processor just to update “Word”: all Applications can be uploaded via MIDI by SysEx commands. You can even update new MIOS versions without having to burn a microcontroller! No PIC burning at all is required when you order your PIC-Microcontrollers at SmashTV or Mike.

MBHP - The MIDIbox Hardware Platform



The MIDIbox Hardware Platform is the consequential continuation of Thorsten Klose's work on MIDI controllers in the last years. It brings down every design to one standardised environment with reusable and exchangeable modules.

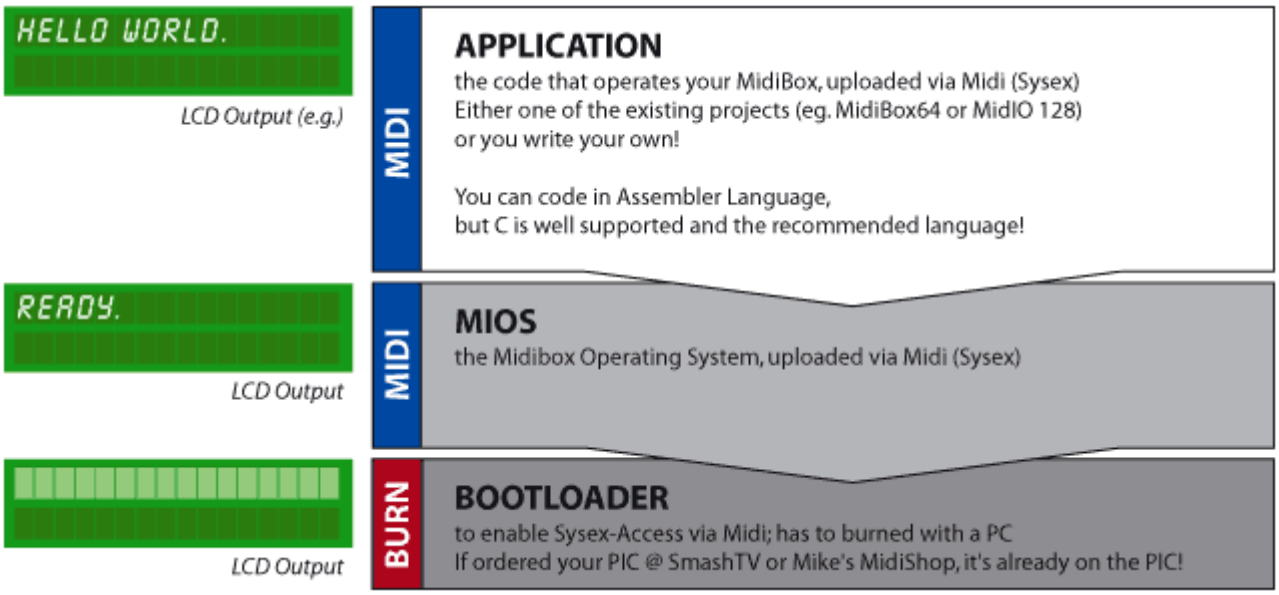
Well, this method is nothing new, did you ever saw the inner life of an old television, or a HIFI system, or a computer, or - of course - a modular synthesiser system? All are consisting of smaller or larger units, which interact over dedicated interfaces. This way of thinking, which simplifies the understanding of complex technical applications, is now also promoted by the MBHP.

The focus of the platform:

- well defined and documented modules
- small, uncomplicated circuits
- realizable on PCBs (single-layer) and prototype boards
- PCBs designed with a freeware CAD program
- different firmwares for different configurations
- open for future extensions by other hobbyists
- programming examples in the PRG and MIOS Download section
- NON-COMMERCIAL!

More Informations: mbhp

MIOS - The MIDIbox Operating System



MIOS is the operating system of the Microchip PIC18F* chip, in a Core module. It can be uploaded by SysEx (via MIDI) thanks to a bootstrap loader, a special software that is already burned onto the chip if you order at SmashTV's or Mike's Shop.

MIOS provides the basic functionality for a bunch of different applications. There are dozens of precompiled, ready-to-use apps available, that you can upload and start making music! It is also quite easy to develop your own applications if you aren't afraid of coding in C.

Main features:

- Bootloader - no PIC burning required - upload applications by MIDI
- Easy to update
- Test new or other applications with existing hardware
- ASM or C
- Application Development toolchain is fully cross-platform compatible
- Well defined and documented MIOS functions, no low-level hardcore nerd programming required
- NON-COMMERCIAL!

For more information: MIOS

MIOS32 - The 32-bit MIDIbox Operating System

MIOS32 is the operating system of the STM32 in a Core32 module. It can be uploaded by SysEx (via MIDI) thanks to a bootstrap loader, a special software that is already burned onto the chip if you order at SmashTV's or Mike's Shop. It can also be uploaded and in-circuit debugged using a JTAG adapter.

MIOS32 provides the basic functionality for a bunch of different applications. There are dozens of precompiled, ready-to-use apps available, that you can upload and start making music! It is also quite easy to develop your own applications if you aren't afraid of coding in C.

Main features:

- Bootloader - no burning required - upload applications by MIDI

- Easy to update
- Test new or other applications with existing hardware
- Applications are coded in the C Programming language
- Application Development toolchain is fully cross-platform compatible
- Well defined and documented MIOS functions, no low-level hardcore nerd programming required
- NON-COMMERCIAL!

For more information: mios32

Applications

Just like the PC you are reading right now, a MIDIbox runs an application on top of the Operating System, to provide the hardware it's functionality.

For Information on the various applications available, see projects

Thoughts

A few things worth taking into consideration before you start....

A MIDIBox is a serious project. Thanks to the efforts of TK and a few assistants, it is accessible to anyone who wishes for one, but don't be fooled into thinking it's going to be a walk in the park.

The forum is not school. It is not warranty. It is not contracted 24/7 tech support ;) It is a place where you can find a bunch of cool people just like you. It is a place where everyone should help everyone else as much as possible. If you have a question, ask it! Don't worry if your response is not forthcoming in a short time, be patient, help will come along.

If someone on the forum says to 'go and read ucapps.de' or 'search the forum' or 'google it', please don't be offended. They aren't trying to get rid of you :) It just means that they know that the perfectly documented and most easily understood response to your question exists elsewhere, and they want you to know where you can find it. If they answered in the forum, they'd probably just be giving you a less complete answer than the one elsewhere. If you've read something that says it is your answer, but you don't understand it, first try googling it and then come ask on the forum. Don't be shy, but don't be lazy either :)

A MIDIBox is not a cheap way out. If you're thinking “Heck, why buy one when I can build one of my own for half price?” then you are barking up the wrong tree. Your first project will involve not only the purchase of a large range of expensive tools, but a collosal investment of your time and efforts. Even if you only earn a few bucks an hour, you're FAR better off investing your time in a second job or some extra shifts, and saving that money to buy a commercial offering.

Take your time. If you're in a rush, go buy a commercial offering. Do not be fooled, a MIDIBox will take at least a couple of months to build, with the exception of some of the very simple devices like the clockbox. Talk to your wife and kids about this thing, or they might wonder if you're having an affair or something ;)

Read. READ. READ!!! Read until your eyes can't take it any more. If you think you've read enough to get started, go and read it again. Seriously, information is imperative. I personally spend around 2 hours every day (at least - it's not unusual for me to spend over 20 hours in a weekend) reading and learning about the electronics involved - but I do this for fun, not the end result.

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

Permanent link:
https://midibox.org/dokuwiki/doku.php?id=home:about:midibox&rev=1229736268

Last update: **2008/12/20 01:24**