MIOS FAQ

I plan to build a tiny foot controller, should I use a reduced PIC16F based MIDIbox, or should I use the new PIC18F452 controller?

You definitely want to use the PIC18F452, as it is available for the same price. Although MIOS offers mighty functions for much more complex applications, the handling will be much easier for you. Especially because of the MIOS Bootstrap Loader, which allows to reprogram the chip without a howto_program_a_pic

But for such a minimal project isn't this big chip wasted money?

Smaller chips like the PIC16F84 are cheaper to obtain, but the price difference of $3 US doesn't really matter in a sparetime project. Also the instruction set of the smaller chips is very reduced, therefore programming such CPUs won't be a lot of fun!
The use of cheaper chips only makes sense when the saved costs are higher than your wasted time!

But I want to produce my project in large quantities and sell it to other people for as much profit as possible - so could you send me a smaller MIOS version - ASAP?

Go away, this is a non-profit project!

Am I allowed to publish my modifications and improvements, or completely new applications based on MIOS?

You are allowed to share your applications with others, it's even allowed to publish them on the web or somewhere else, but please ask the copyright holder(s) of the program parts you are re-using in your application before doing this.

The copyright holders are listed in the headers of the source files, sometimes also in a README.txt or COPYRIGHT.txt file.

How do I burn the bootstrap loader into the PIC18F452?

Fortunately this can be made with a common PIC programmer, see also howto_program_a_pic. Once

the bootstrap loader has been written into the flash memory, OS and application updates can be made via MIDI. Therefore it isn't necessary anymore to build such a programmer by yourself for frequently firmware updates; you could ask a friend or somebody of the MIDIbox Forum if he does this job for you.

Also you can send your chip to SmashTV for free programming or you can buy them pre-programmed from the MIDIbox Store

Can I burn a MIOS application into the PIC without the bootstrap loader.

No, the bootstrap loader is a *must*. It's also required that MIOS is already running before uploading an application. The upload procedure is described under
http://www.ucapps.de/mios/mios_upload_procedure.gif

Can I use a PIC18C452 instead of a PIC18F452?

No, because the C device is only one-time-programmable - keep away from these OTP chips!

Can I use a PIC18LF452 instead of a PIC18F452?

Yes. The LF type is the low-power version which works equally well in a MIDIbox.

What is the difference between a PIC18F452I/P and PIC18F452I/PT?

The package type. (With integrated circuits, “package type” refers to *form factor.*) The PDIP package is absolutely recommended for DIY projects, and is the only type that will fit into the PIC socket on the Core), so use a PIC18F452-I/P or PIC18LF452-I/P.

How can I test if my PIC is running?

During the startup phase, the bootstrap loader will send following SysEx message over MIDI Out “F0 00 00 7E 40 00 01 F7”. This identifies that your PIC is running. The whole test procedure is described under http://www.ucapps.de/howto_debug_midi.html

How can I test if MIOS is running?

After the startup phase, you can send this message “F0 00 00 7E 40 00 0F F7” to the core, MIOS

should reply with the same message immediately. You could also try to upload an application, after every code block a checksum message will be send which acknowledges that the stream has been received correctly.

How do I connect MIDI In/Out LEDs, a MIDI Thru port and/or a COM interface to the core module?

See the MBHP_LTC page.

How do I connect an LCD to the core module?

See howto_connect_a_lcd.

Is a LCD required for MIOS?

No!

But using an LCD will be a BIG help in getting your project working. Connect an LCD to your core module unless you have enough experience to know that you can do without it. It doesn't have to be part of your panel, or even stay connected when the project is working. Just have it there when you are bringing up the project. Your first goal in a new project should be to get a working core module with LCD. *Jim Henry*

How do I connect buttons to the core module?

See the MBHP_DIN page.

How do I connect LEDs to the core module?

See the MBHP_DOUT page.

How do I connect pots or faders to the core module?

See the MBHP_AIN page. Note that MIOS supports also the direct (“unmuxed”) connection to the analog pins. So, if you only want to use 8 pots, you don't need to build the AIN multiplexer modules. The appr. configuration has to be made in the application main.asm file.

How do I connect motorfaders to the core module?

See the MBHP_MF page.

How do I connect a BankStick to the core module?

See the MBHP BankStick page.

How do I connect rotary encoders to the core module?

See the MBHP_DIN page.
(Note that MIOS needs to know which shift registers the encoders have been connected to. This change has to be made in mios_tables.inc)

How do I connect LED rings to the core module?

See the MBHP_DOUT page. The DOUT modules for the LED rings have to be connected to the same shift register output chain like the other LEDs (J8 of the core module, or cascaded to another DOUT module). Note that LED rings are handled by the user application, the setup has to be made in the main.asm file.

How do I connect a SID to the core module?

See the MBHP_SID page.

How do I connect CV outputs to the core module?

See the MBHP_AOUT and MBHP_AOUT_LC page.

How do I test my modules?

With the AIN DIN DOUT test application for pots/buttons/LEDs, or MIDIO128 for testing buttons/LEDs only. Motorfaders can be tested with the MF Calibration program, the SID and the BankStick with the SID application. LCDs with every application.

I want to build a controller with 128 pots, how can I achieve this?

You could build two core modules and link them together

How do I modify the application code and build a new .syx file?

Follow the instructions which are given in the “main.asm” file.

Where can I learn more about the PIC18F instructions?

See also Book Reviews

But before buying a book about PIC assembly programming, consider that in many cases the use of MIOS C (→ http://www.ucapps.de/mios_c.html) is sufficient for common applications, and therefore assembly knowledge is not required.

The most important informations (for instance a complete instruction list) can be found at the Microchip site. Search under Products→PICmicro controllers→PIC18 Microcontroller Family→PIC18F452→Datasheet. There is also a reference manual for the whole PIC18 family available under Engineer's Toolbox→Reference Manuals→PIC18C MCU. Some interesting informations are also given by the Application notes (but you will find no informations about MIOS applications there 😉. The PICList is also a nice information source.

Where can I find more informations about MIOS specifics?

Currently only in the source codes and in the MIOS Functions Reference. Some additional docs are available in the MIOS Download Section

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

Permanent link:
https://www.midibox.org/dokuwiki/doku.php?id=mios_faq&rev=1151682387

Last update: **2006/10/15 09:35**