The following boot options are controlled by the PIC ID header, which can be changed with IC-Prog before “burning” the firmware:

1. ID[57..63] : SysEx Device ID (0x00 - 0x7f)

```
  choose a unique ID if you want to chain your MIDIboxes
```

1. ID[55] : if 0, UART will be initialized with common MIDI baudrate

```
            if 1, UART will be initialized with 38200 baud

            enable this setting your MIDIbox is connected via the

            to-Host Interface (RS232) to your computer
```

1. ID[48..51] : LCD type.

```
               Currently following LCDs are provided:

               0x0 : common character display

               0x1 : graphical LCD, KS0108 compatible

               0x2-0xf: reserved
```

Examples:

```
ID = 0000000000000000 -> Device ID = 0x00

                         common MIDI interface

                         character LCD

ID = 0000000000000012 -> Device ID = 0x12

                         common MIDI interface

                         character LCD

ID = 0000000000000142 -> Device ID = 0x42

                         to-Host Interface

                         character LCD

ID = 0000000000001000 -> Device ID = 0x00

                         common MIDI interface

                         graphical LCD (KS0108)
```

Note that the Bootstrap loader itself doesn't print a message on the LCD screen! After startup it requests a memory dump with the SysEx command: F0 00 00 7E 40 01 F7

If no operating system is installed, the loader repeats the request every 2 seconds.

During this time the converted dump file (*.syx) can be uploaded.

IMPORTANT: since writing to flash memory requires some time, you've to use a SysEx loader which allows to insert delays after every

F7 byte. Recommented tool: MIDI-Ox (http://www.midiox.com)

(Note: in difference to the PIC16F MIDIboxes, you don't have to specify an additional delay between every byte, only between every “F0 ... F7” block)

The MIOS bootstrap loader can fetch up to 1k at once. Thereafter a delay of ca. 750 mS has to be inserted before sending the next data block. The “hex2syx.pl” already takes care about this requirement by splitting the dump to 1k blocks.

After the code has been uploaded (or the operating system is already in flash memory), you don't need to use the primary bootstrap loader anymore, unless you want to update MIOS. Instead you will use the secondary bootstrap loader which is one of the background task of MIOS. This loader is similar to the primary one, but it prevents you from overwriting the operating system.

Performance:

1. upload of MIOS (12k): 9 seconds
1. upload of a typical application (4k): 3 seconds

[Thorsten.Klose@gmx.de]

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

Permanent link:
https://www.midibox.org/dokuwiki/doku.php?id=how_to_select_a_pic_id_header

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