

Introduction

The Octopus firmware may be accessed from a computer terminal using the built in USB port of Octopus. This connection type offers a channel to update the OS, in addition to using MIDI system exclusive transmission.

About this document

Once someone receives a machine-readable form of the Octopus OS, they will need to somehow embed it into the Octopus for execution. This document describes precisely the steps required to do that.

The described method of updating the software is entirely disconnected of the MIDI upgrade procedure using SYSEX files, since it is relying entirely on the boot loader that comes pre-installed on every Octopus machine. The boot loader is the program that gets executed on the lowest software level, just after power-on of the machine.

Technology overview

The Octopus internal memory is made up of a volatile (RAM) section and a permanent (FLASH) section.

Upon power-on (or reset) for that matter, a boot loader is started from the FLASH section. The boot loader is set up such that it then automatically loads the Octopus software from the FLASH into the RAM and starts executing it.

To replace the Octopus OS software, five steps are required:

- 1. Connect Octopus to your computer
- 2. Interrupt the boot loader prior to execution of Octopus OS
- 3. Load the new Octopus OS onto the Octopus hardware
- 4. Write the new OS file to Octopus flash
- 5. Boot into the new Octopus OS

NOTICE: unless otherwise specified, all internally saved data will be lost after an OS upgrade! Also, until further notice, externally saved data (via SYSEX) is currently not compatible across OS versions. Neither up- nor downward compatibility is provided at the moment. This may change with future versions of the OS.

Updating the Octopus OS

Updating the Octopus OS requires completion of all steps described in this chapter. Please make sure to follow the instructions precisely as described herein. Failure to do so may result in rendering your hardware inoperable.

1. Connecting Octopus to your computer

Octopus comes with a USB connector on board, which may be used as a computer interface. Following the instructions below you will build a serial connection between your computer and Octopus, and then use a simple terminal program to talk to the Octopus hardware.

This procedure was tested and documented for Apple Mac OS X (10.4.9 and later) and Microsoft Windows XP.

USB virtual COM port driver

In the first step you have to install a driver that will let you use your computer's USB ports as a serial interface to Octopus – which carries an FTDI USB chip of the type FT232xx.

This is a one-time install for your platform, which means that unless you uninstall the driver at some point in time you will not need to do this again for future software updates via the USB port.

The driver we need is called a Virtual COM Port (VCP) driver for the Octopus USB chip and may be downloaded freely from the following internet address on the FTDI site:

<http://www.ftdichip.com/Drivers/VCP.htm>

The installation of the driver is documented in detail in the documents at the following address on the FTDI site:

<http://www.ftdichip.com/Documents/InstallGuides.htm>

Make sure you only read the relevant section of the installation document, which is covering more than just the chip we are using.

Terminal connection

Once you have installed the virtual COM port driver on your computer, you are ready to connect to Octopus using a simple terminal emulation program. This may be Minicom on Linux, zTerm on Apple OS X, or Hyperterm on Windows machines. In the following we will use the Apple OS X and Windows XP examples showing how to do it.

Connecting from Apple Mac OS X

1. Install the FTDI virtual COM port driver

As mentioned above, the first step we need to do is install the FTDI virtual COM port driver. Before we start, disconnect the Octopus from the Mac, if it is connected. The installation should be relatively easy, you need to download and install the driver from <http://ftdichip.com/Drivers/VCP.htm>

On the site choose the right driver for your system (depending on whether you are on an Intel Mac or a PPC Mac). The installation is wizard driven, so not much to do wrong there. You will need to re-boot your machine as you are prompted.

2. Verify that the FTDI virtual COM port driver is installed.

In the Finder, go to your root drive/System/Library/Extensions. Check that the file FTDIUSBSerialDriver.kext is listed.

Next, verify that Octopus is recognized. You may do this in the System Profiler. Under the tab Hardware / USB make sure to open up all twisties - you should find an Octopus entry. Notice that it may be there regardless of Octopus being on/off or connected at all.

3. Connect your Octopus to your Mac

After installing the FTDI virtual COM port driver connect Octopus to the Mac using a standard USB cable. Make sure Octopus is turned on. If you have an USB lamp handy, it would be great to use as a control that the power supply is working and the machine is turned on.

4. Set up zTerm to talk to Octopus

Start zTerm and go to the menu Settings->Modem Preferences. This will bring up a window. In the Serial Port menu at the top you should be able to select something like "usbserial-0000101D" (this may vary on your machine). This is the virtual COM port for which we have installed the driver so select it.

In zTerm still, go to the menu Settings->Connection. Select a data rate of 38400, Data Bits: 8, Parity: None, Stop Bits 1, no handshake.

5. Reset Octopus

Reset the Octopus by using the reset button on the downside of the front panel. It is the red button roughly underneath the top right encoder. As soon as you hit the reset button you should see the boot sequence text come up in your zTerm window.

The rest of the procedure is the same for all platforms.

Connecting from Microsoft Windows XP

1. Install the FTDI virtual COM port driver

Please follow the driver installation that comes with the driver download from the FTDI site.

2. Connect your Octopus to your computer

Make sure that it is recognized as a device. Check this by following the description in the Driver installation document.

3. Set up Hyperterm to talk to Octopus

Start Hyperterm to which may be found under (START->All Programs->Accessories->Communications->Hyperterm). Upon start Hyperterm will ask you to create a connection.

First you are prompted to give the connection a name. The connection may be given any name, in our case it may be called "Octopus VCP". Once you enter the name you like, press OK and you should move on to the next screen.

Make sure that the field "Connect using" shows the COM port assigned to your USB port. Press OK again, to get to the property settings of your selected COM port.

Make sure to enter the Properties as follows: select a data rate of 38400, Data Bits: 8, Parity: None, Stop Bits 1, handshake NONE.

4. Reset Octopus

On the Octopus machine, press the red reset button on the back side of the front panel, next to the place where the cable hose enters the casing. You should see in your terminal window something similar to the picture below:

The rest of the procedure is the same for all platforms.

The boot sequence text

After successfully connecting Octopus to your computer and upon a reset, you should see the boot sequence text appear in your terminal window. The text will look similar to the following:

```
+
RedBoot(tm) bootstrap and debug environment [ROM]
Non-certified release, version UNKNOWN - built 04:27:46, Mar 12
2005

Copyright (C) 2000, 2001, 2002, 2003, 2004 Red Hat, Inc.

RAM: 0x00000000-0x00200000, [0x0000cfa0-0x001ee800] available
FLASH: 0x01800000 - 0x1a00000, 32 blocks of 0x00010000 bytes
each.

Adapted by genoQs Machines in 2006 as Octopus boot loader.

== Executing boot script in 1.000 seconds - enter ^C to abort
Octopus> fis load OCT_OS.elf -b 0x00010000
Octopus> go
```

Congratulations, you have just managed to successfully interface your computer to the Octopus hardware console and observe its boot procedure on your computer screen.

Obtaining the .elf OS file to be uploaded

After successfully connecting to your Octopus console you are ready to interact with it, including uploading an new OS file.

The OS file required for the update via USB is in the “.elf” format. This is different from the “.syx” (SYSEX) format and any attempt to upload a .syx format file will fail.

To obtain the .elf format OS file please use a modified URL of the .syx format file. In the URL simply replace the .syx extension with the .elf extension.

For example if the .syx format file URL is:

http://www.genoqs.com/downloads/OCT_OS_v162.syx

..the corresponding .elf format file URL will be:

http://www.genoqs.com/downloads/OCT_OS_v162.elf

Use the .elf file URL to download the .elf OS file in a convenient place on your machine. From here we will upload it to the Octopus memory shortly.

2. Interrupt the boot loader

Understanding the boot sequence text (optional)

Let's take a minute to discuss piece by piece what you see in the startup message of an Octopus machine like the following:

```
+
RedBoot(tm) bootstrap and debug environment [ROM]
Non-certified release, version UNKNOWN - built 04:27:46, Mar 12
2005
```

```
Copyright (C) 2000, 2001, 2002, 2003, 2004 Red Hat, Inc.
```

```
RAM: 0x00000000-0x00200000, [0x0000cfa0-0x001ee800] available
FLASH: 0x01800000 - 0x1a00000, 32 blocks of 0x00010000 bytes
each.
```

```
Adapted by genoQs Machines in 2006 as Octopus boot loader.
```

So far you see that the Octopus boot loader is based on RedBoot, © by Red Hat Inc. RedBoot is a versatile, industrial-strength, free boot loader which may also be downloaded as source and customized to any hardware need you may have. We have made some customizations to the plain version of RedBoot to fit the Octopus hardware.

```
== Executing boot script in 1.000 seconds - enter ^C to abort
Octopus> fis load OCT_OS.elf -b 0x00010000
Octopus> go
```

This section shows that RedBoot is configured to execute a boot script, which is nothing else than loading the Octopus OS into the memory and starting it with the “go” command. What we will do next is interrupt the boot script in order to get control over RedBoot so we can use it for our purposes.

Interrupting the boot sequence

The boot script may be interrupted during a time interval of one second by pressing Control-C on your computer keyboard. Let's do just that in the next step.

Make sure to have the zTerm or Hyperterm window in focus - click on it just to be sure. Now press and hold Control-C on the computer keyboard, then press the Octopus reset button and hold it for 2 seconds. Release it, and you should get the following message:

```
RedBoot(tm) bootstrap and debug environment [ROM]
Non-certified release, version UNKNOWN - built 04:27:46, Mar 12
2005
```

```
Copyright (C) 2000, 2001, 2002, 2003, 2004 Red Hat, Inc.
```

```
RAM: 0x00000000-0x00200000, [0x0000cfa0-0x001ee800] available
FLASH: 0x01800000 - 0x1a00000, 32 blocks of 0x00010000 bytes
each.
```

```
Adapted by genoQs Machines in 2006 as Octopus boot loader.
== Executing boot script in 1.000 seconds - enter ^C to abort
^C
Octopus>
```

Now you may release the Control-C press on your computer keyboard. The additional “^C”s showing do not have any bad effect.

Important is that you get the last line of the message above: the prompt to the Octopus boot loader. With that done:

Congratulations, you now have an “Octopus>” prompt, which tells you that you may now operate on RedBoot as you wish.

Furthermore, the technically interested reader will find details on RedBoot at the following address:

<http://ecos.sourceforge.org/redboot/>

3. Loading the new OS file to RAM

Set Octopus to receive file transmission

We will now load the new OS file in .elf format onto the hardware. Assuming you have already downloaded it and have it handy, as described in an earlier section, there are two steps associated with this: first, loading it into the Octopus RAM and second, writing it to the Octopus FLASH.

Let's transmit the new .elf OS file to Octopus using the terminal program, in this example zTerm on Mac OS X. Others will behave in a similar fashion.

First, we will tell Octopus that we are about to send something, so Octopus will open a connection and wait for the file to be transmitted. We issue the following command (followed by pressing "Enter"):

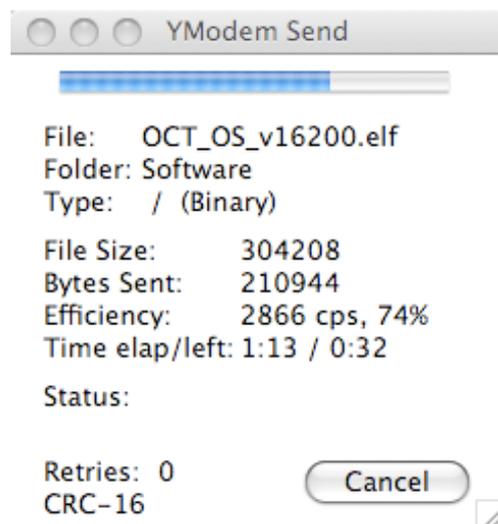
```
Octopus> load -m ymodem -b 0x00010000
```

Send the new OS file to Octopus

With Octopus waiting for the transmission (as we saw above) you should then proceed with sending the OS .elf file using the YModem protocol, supported by your terminal program. To do this, on zTerm we choose the menu "File -> Send Files -> YModem-ik". Other terminal programs may vary in terminology slightly.

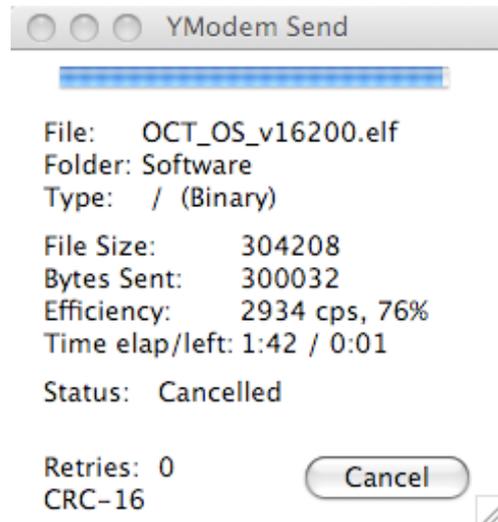
When prompted, select the .elf OS file to be sent from where you have stored it locally and press the "Open" button.

Once the transmission begins, a status window similar (do not worry if you see slightly different numbers) to the one below will show up to indicate the progress of the upload.



Once the transfer is complete, Octopus (as the remote system) will detect the end of the file transmitted and close the connection.

Knowing this, do not get intimidated by a “canceled” connection which in this case is fine, such as illustrated here:



Shortly thereafter the transmission status window will disappear and the terminal should show a message similar to the one below:

```
Octopus>
Octopus> load -m ymodem -b 0x00010000
CEntry point: 0x00010040, address range: 0x00010000-0x00054198
xyzModem - CRC mode, 1(SOH)/305(STX)/0(CAN) packets, 3 retries
Octopus>
```

Note the number of retries mentioned in the message above. This is nothing bad, this is typically just the number of times Octopus polled the transmission between the time you pressed enter on the command line and the time you actually selected the .elf file to be transmitted to the machine.

You have successfully loaded the new OS file into the RAM of the Octopus hardware.

In the next section we will write the OS file from RAM into the FLASH, so it does not get erased when we turn off the power of the unit.

4. Writing the new OS file to FLASH

WARNING: this step is extremely critical!
Please make sure it is executed precisely as indicated here, or you may render your hardware inoperable.

Write the OS file from RAM to FLASH

At the Octopus prompt, issue the following command.

Make sure the command is all in one line, and that no line breaks were inserted in case you copy and paste from this document.

```
> fis create -b 0x00010000 -l 0x00070000 -f 0x01820000 OCT_OS.elf
```

Double check in the command line above that “ -l “ is a lower-case letter “ L ” and not the character for number one (“ 1 ”).

When ready, complete by pressing “Enter”.

If a OS image is present in flash, you will get a message prompting you to confirm that you want to continue with the procedure. You do, so press “y” and “Enter” when asked. It looks like this:

```
An image named 'OCT_OS.elf' exists - continue (y/n)? y
```

If updating the Nemo then replace the above filename with ' NEMO_OS.elf '

Now your screen should produce output similar to the following:

```
Octopus> fis create -b 0x00010000 -l 0x00070000 -f 0x01820000
OCT_OS.elf
An image named 'OCT_OS.elf' exists - continue (y/n)? y
... Erase from 0x01820000-0x01890000: .....
... Program from 0x00010000-0x00080000 at 0x01820000: .....
... Erase from 0x019f0000-0x01a00000: .
... Program from 0x001f0000-0x00200000 at 0x019f0000: .
Octopus>
```

You have successfully written to FLASH the new Octopus OS file.

List the FLASH content (optional)

Just to make sure, you may optionally want to take a quick look at the flash contents issuing a “fis list” command:

```
Octopus> fis list
Name                FLASH addr  Mem addr    Length      Entry point
RedBoot             0x01800000  0x01800000  0x00020000  0x00000000
OCT_OS.elf          0x01820000  0x00010000  0x00070000  0x00010040
FIS directory       0x019F0000  0x019F0000  0x0000F800  0x00000000
RedBoot config      0x019FF800  0x019FF800  0x00000800  0x00000000
Octopus>
```

5. Boot and initialize the memory data

Boot without loading stored data

The last step is to make sure you boot the machine in the new OS, but without loading any sequencer data that may be stored in the FLASH. Hold the CLR button (third from top in the MUT column) and press the reset button to reboot the machine.

Save the clear memory state

Once the machine has booted, press CLR again, and then save the state of the machine by pressing GRID+Program. This will ensure that your data is in a state consistent with the new OS.

From here on, you are ready to go and enjoy the benefits of the newly installed OS!