Installing and Configuring NitrOS-9 and HDB-DOS

Boisy G. Pitre – June 1, 2008

Cloud-9's customers have had the convenience of NitrOS-9 and HDB-DOS residing on the same storage device (be it hard disk or CompactFlash card) for some time now. It is the ultimate CoCo setup, since it allows for floppy-less booting into NitrOS-9, not to mention that all of your BASIC and M/L software can reside on the same physical device. Yet setting up a storage device in this manner requires a detailed understanding of the intricacies of HDB-DOS, NitrOS-9 and SuperDriver.

Given the number of customers requesting how to set up their own systems, it's time to unveil the mystery behind this powerful configuration. Now, with the information provided in this document, you can set up your own storage device to achieve floppy-less booting and have a dual NitrOS-9/HDB-DOS partition.

Some assumptions are made throughout the document:

1. The system is a CoCo 3 with at least 512K of RAM
2. There is a Multi-Pak with a 5.25" double sided floppy drive as drive 0
3. HDB-DOS 1.1D and SuperDriver 2.1 is used
4. NitrOS-9/6809 Level 2 v3.2.8 is the operating system to be installed
5. The target storage device is a SCSI hard drive (IDE users, replace references of /s0 with /i0 and /sh with /ih)

Phase 1: Gathering Information

Before I begin, there are two key numbers that I must know in order to prepare my storage device for NitrOS-9 and HDB-DOS:

1. The number of sectors that the storage device has
2. The sector size (in bytes) that the storage device uses

In this document, I will walk you through the process that I take to set up a Quantum 40MB SCSI hard drive as the storage device to hold HDB-DOS and NitrOS-9. Your storage device may be a different hard drive or a CompactFlash device, so the numbers I use will almost certainly NOT be the numbers you will use. Substitute where necessary.

The technical information about the Quantum 40MB hard drive states that it has $01406D (82,029) sectors, and its sector size is 512 bytes. Again, your numbers will be different for your storage device.

Phase 2: Setting up HDB-DOS

As I turn on my CoCo 3 system, I see the familiar Disk BASIC sign on screen and OK prompt. I then insert the HDB-DOS disk into the drive, and type:

RUN "WIZARD"

The Wizard allows me to configure a custom version of HDB-DOS, tailored for my system. One important note: you must be running Disk BASIC 2.1 for the Wizard to work. Disk BASIC 2.0 is not supported by HDB-DOS.

At startup, Wizard will show a welcome screen. I press a key, and Wizard saves my Disk BASIC ROM to the floppy in preparation for changing it over to HDB-DOS. Once this is complete, I press a key to move onto the next screen.

Wizard now asks for me to select a controller. I select TC^3 and that overlay loads. (Select your controller or interface based upon your configuration) Once the loading is completed, I press a key to move to the next screen.

The next step is to select the address where the controller is mapped. Be sure this is correct! Wizard reports my choice and verifies the address. I select $FF74 and press a key.

Now Wizard asks for me to select my floppy step rate (usually 6 milliseconds is fine). I select it and press a key.

Next, I am asked which SCSI ID that I want to be used at startup. I choose 0, and I recommend your main SCSI startup drive also has an ID of 0 (IDE setups should use the master drive as the boot drive). After selecting the ID, I press a key to continue on to the next screen.

Finally I am asked if I want to allow an OS-9 partition on the drive. That's the whole point of this exercise, isn't it? So I press the Y key.

An explanation comes up describing the HDB-DOS offset. I press a key, then am prompted for the number of sectors my device has. I enter in the number 82029 to correspond with my number of sectors on the hard drive. Here, you will put in the number of sectors for your storage device.

Once I've entered the number of sectors, Wizard will then ask the sector size. Because my drive has 512 byte sectors, I need to select that option.

Wizard tells me the size of the storage device in megabytes, and asks how many HDB-DOS drives I would like. HDB-DOS can have up to 256 virtual drives, and depending upon the size of the storage device, it may make sense to put in the

maximum of 256. However, keep in mind that on a device with 256 byte sectors, a full span of 256 HDB-DOS virtual drives will require 41,287,680 bytes, or approximately 41MB. On a device with 512 byte sectors, HDB-DOS requires double that space! Since I have a 40MB hard drive with 512 byte sectors, I do not have enough space for 256 HDB-DOS drives. On top of that I need to reserve some space for NitrOS-9. Therefore, I will select 64 as the number of HDB-DOS drives for this device.

After entering the number, Wizard reports a summary of the proposed drive partitioning based on my previous input. There is also an important value: the HDB-DOS offset. For my hard drive, it is $00A2ED. Again, yours will be different. Write this number down because we will use it later.

I can either agree to the offset by pressing Y or redo the exercise by pressing N. I'm satisfied with my result, so I'll continue by pressing Y.

The next screen tells me that the OS-9 partition will have 83,418 256-byte sectors available. Remember, RBF thinks in 256 byte sectors. Our device really has 512 byte sectors. So we have to tell RBF that we have twice the number than we actually have in order for it to utilize all of the space we want for it.

I press a key a couple of times and two files will be saved: EPROM.DOS and HDB-DOS.BIN.

With that complete, I go ahead and jump into my newly configured HDB-DOS by typing:

```
LOADM"HDB-DOS":EXEC
```

Now that I am in HDB-DOS, the first order of business is to copy my HDB-DOS disk over to drive 0 of the storage device by typing:

```
DRIVE OFF
BACKUP 0 TO 4
DRIVE ON
BACKUP 4 TO 0
```

After these commands are executed, my HDB-DOS disk will reside on the HDB-DOS hard drive partition's virtual drive 0. I'm now ready to install NitrOS-9.

Phase 3: Installing NitrOS-9

Before I dive off into installing NitrOS-9, there is a very important principle that I want you to understand: RBF (the NitrOS-9 module responsible for managing disk space) sees EVERY storage device as having 256 byte sectors, even though that may not be the case (and it certainly isn't in this example here). HDB-DOS also sees devices as having 256 byte sectors. The BIG difference is that through SuperDriver, NitrOS-9 can utilize ALL bytes of a device's sector (up to 2,048 byte sectors). For backward-compatible reasons, HDB-DOS can only utilize the first 256 bytes of a

sector. This means that HDB-DOS "wastes" all storage in a sector PAST the first 256 bytes.

Ok, now it's time to set up the NitrOS-9 side of the storage device:

1. I'm booting the SuperDriver disk which puts me in NitrOS-9/6809 Level 2.
2. At the shell prompt, I type:

```
dmode /s0 typ=81 cyl=6c9e sid=3 sct=1 t0s=1
```

Let's break down this command's parameters so we can understand what it all means:

1. typ=81: this turns off the "auto-size" feature (bit 4) in the descriptor. Without this, the format command would use SuperDriver to interrogate the storage device for the number of sectors and format the entire disk for NitrOS-9, thus overwriting our HDB-DOS partition. By turning off this bit, the format command will use the values in the cyl, sid and sct/t0s fields to determine how much of the storage device will be formatted.
2. cyl=6c9e sid=3 sct=1 t0s=1: Remember HDB-DOS's Wizard said that my OS-9 partition would have 83,418 256-byte sectors? Well, I need to communicate that to the format command in order for it to allocate the right amount of space for my OS-9 partition.

   There is some slight trickery needed here. In order to coax the format command to initialize our storage device with exactly 83,418 256-byte sectors, I need to shoehorn that value into the cyl, sid and sct fields (sct and t0s will always be the same) so that the following formula:

   Total Sectors = (cyl * sid * sct) – sct + t0s

   will yield the desired number of sectors (83,418).

   The cyl field is 16 bits while the sid, sct and t0s fields are 8 bits in size. Since my number of sectors (83,418) is larger than 16 bits, I must use division to obtain a whole integer dividend. What I did in this example was to divide 83,418 by 3, to get a whole number: 27,806 ($6C9E). Since cyl holds a 16 bit value, I chose to set it to $6C9E. The sid and sct fields hold 8 bit values, thus I've set sid to 3, and left sct (and t0s) to 1. Multiply them all out and you get:

   TS = ($6C9E * 3 * 1) – 1 + 1 = $00A2ED = 83,418

   You'll have to perform some math with your number to get it to "fit" into the fields. A calculator is handy here.

Now I'm ready to format the storage device, so I type:

```
format /s0
```

Format asks if I am certain that I want to format a hard disk. I am, so I answer Y. Being paranoid, format tells me that that it is a hard disk and asks if I am sure I want to format the disk. Again, I answer Y. Next, I am asked if I want to do a PHYSICAL format. I answer N to that question (physical formats usually aren't necessary). Now I am prompted to type in a disk name. I do so, and press ENTER. The last question asks if I want a physical verify done. If you're the perfectionist type, you would answer Y, but be prepared to wait a while for the process to complete, depending on how large your drive is. I've formatted this disk recently with a verify, and I "trust" the veracity of the storage device, so I'll answer N.

Once the storage device is formatted, I can copy the NitrOS-9 system onto it. First, I will eject the SuperDriver disk and insert the NitrOS-9 System Master disk in the floppy drive and type:

```
chd /dd
chx /dd/cmds
dsave -s56 /s0 ! shell -p
```

This saves the entire System Master disk to the storage device. It takes a while, but once it's completed, I eject the disk and insert the Modules/Basic09 disk (if you are using an 80 track 3.5" disk, you can skip this last step since that disk holds both the NitrOS-9 System Master and the Modules/Basic09 disks):

```
chd /dd
chx /dd/cmds
dsave -s56 /s0 ! shell -p
```

Just like before, the entire contents of that disk are copied to the storage device. Several ERROR #218's flash across the screen. That's ok. There are some duplicate files on the second disk, and I don't want to overwrite the ones I copied earlier.

Once the Modules/Basic09 disk has been copied, I eject it and insert the SuperDriver disk back into the drive and type:

```
chd /dd
chx /dd/cmds
dsave -s56 /s0 ! shell -p
```

I have just copied all the files from the NitrOS-9 and SuperDriver disks over to the storage device. This sets up the device to hold the system and will allow me to configure a custom system boot file next.

Phase 4: Customizing NitrOS-9

Now comes the final part: creating a bootfile that will boot from the hard drive. The NitrOS-9 boot file will actually reside on an HDB-DOS virtual drive, and NOT in the NitrOS-9 file system. In order to accomplish this, I'll use SuperDriver's ability to read/write HDB-DOS virtual drives with the /SH descriptor.

First, I must set up the /SH descriptor to know the virtual drive I want to write to, as well as my HDB-DOS offset. This lets SuperDriver know where the HDB-DOS partition starts so it can safely write to the correct area of the disk.

Since I have 64 virtual disks, I'll put the NitrOS-9 boot file on the last virtual disk: disk 63 ($3F). And my HDB-DOS offset is $00A2ED, so I type:

```
dmode /sh stp=3f wpc=0 ofs=a2ed
```

Now I run the mb.scsi script that was copied to the hard drive earlier:

```
chd /s0/nitros9/6809l2/scripts
mb.scsi
```

This script uses the scsi.bl bootlist file to build a custom boot on drive 63. The script will prompt me for the destination device that the finished boot disk will go on to. I type /sh and press ENTER, then another key to get the script going. Once the script has completed, I insert my HDB-DOS disk into drive 0 and reset my CoCo 3, then type:

```
LOADM"HDB-DOS":EXEC
```

Since I copied my HDB-DOS disk over to virtual drive 0 earlier, I will see a menu appear when HDB-DOS starts. If I press the '1' key at the menu, I will boot into NitrOS-9 from drive 63!

Additional Notes

- If you plan on making the storage device a permanent fixture on your CoCo system, you probably want to burn HDB-DOS into ROM or Flash. The EPROM.DOS file that was generated by Wizard is the file you want to burn.
- Setting up a NitrOS-9 Level 1 bootable system for a CoCo 2 can be done with these directions as well. A CoCo 3 is still needed to set up the storage device, but substitute the NitrOS-9/6809 Level 1 System Master and Modules disks, and change into the 6809l1 directory before running the mb.scsi script.
- The bootlist files that come with SuperDriver 2.1 use the software clock. If you have a real-time clock installed on your Cloud-9 hardware, you will need to edit either the scsi.bl or ide.bl file to comment out the clock2_soft line and uncomment the clock2_cloud9 line.

Conclusions

Following these instructions for your particular storage device should yield you the same results: a fully bootable hard drive or CompactFlash unit that you can use with both NitrOS-9 and HDB-DOS.

Good luck!