# Quick&Dirty FreeBSD on Alix (without PXE boot)

 It's been a while since my last post and I've been quiet busy writing on my graduation essay.

Meanwhilst my colleague Steven donated me a somewhat dated PC Engines Alix computer. I though to put it to some good use as a packet generator for my new network playground I'm currently building up.

Funny anectode: While googl'ing around on some docs about Alix computers, I stumled accross his 2009 original post on FreeBSD installs.

But then I read that I'd need to go through config hell for DHCP, PXE, NFS for a one-time install ... Oh boy, must be kidding ...

It quickly came to me to actually use VMware to perform the FreeBSD install in 15 minutes or so.
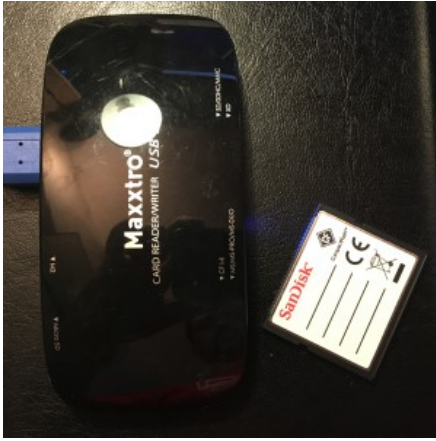Of course I had a test VM around, but first I went out to configure VMware's USB settings:
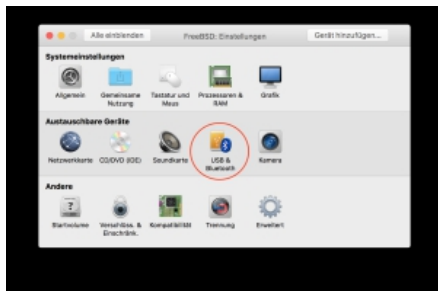
There I selected the USB card reader and chose to connect it exclusively to the VM.
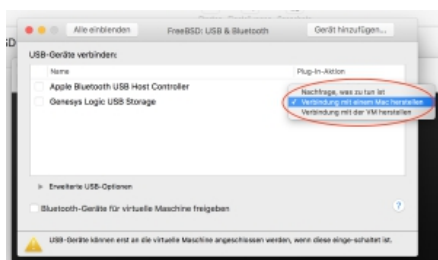
After booting into FreeBSD, I checked using **camcontrol devlist** command wether the card reader was visible to the OS

So it's about time to insert the CF card.

Good, so let's reinitialize the CF card with a new DOS-compatible partition table and an appropriate disk label.
Note that GPT should not be used.

```
fdisk -B -I /dev/da0
bsdlabel -B -w /dev/da0s1
```

The above will write a standard label, with just one single partition. Since my CF carf is only 1 GiB in size, I'm not going for swap or separate partitions besides a root filesystem.
So just make a new filesystem on the **a** partition:

```
newfs /dev/da0s1a
```

Good, so next is to mount the partition and populate FreeBSD to it. You should have either an ISO image or an install CD ready, or go for online sources, which is what I did.

```
mount /dev/da0s1a /mnt
fetch ftp://ftp.freebsd.org/pub/FreeBSD/releases/i386/i386/10.2-RELEASE/kernel.txz -o- | tar -xzpvf- -C /mnt
fetch ftp://ftp.freebsd.org/pub/FreeBSD/releases/i386/i386/10.2-RELEASE/base.txz -o- | tar -xzpvf- -C /mnt
```

And now some final configuration steps, otherwise the install won't boot properly (perform all steps from within a chroot):

        - Configure a root password.
 create another user for SSH login, which is member of **wheel**
        - disable virtual terminals to prevent errors
        - enable getty on serial console instead
        - enable serial console for the kernel as well

```
chroot /mnt
```

```
passwd root
adduser
echo '/dev/md0s1a / ufs rw 1 1' > /etc/fstab
sed -i.bak -E -e 's/(ttyv.*xterm.*)on/1off/g' /etc/ttys
echo 'ttyu0 "/usr/libexec/getty std.9600" vt100 on secure' >> /etc/ttys
echo 'console="comconsole"' >> /boot/loader.conf
exit
```

Some additional configuration is needed in **/etc/rc.conf**.
I choose to place /tmp on a MFS device (memory file system, aka RAM disk), despite it takes away some RAM.
Also, automatic file system checks are enforced if unclean file systems are encountered.
Yet, DHCP is enabled on all interfaces.

```
sshd_enable="YES"
tmpmfs="YES"
tmpsize="20m"
fsck_y_enable="YES"
hostname="alix.localdomain"
ifconfig_DEFAULT="DHCP"
```

That's it, umount the CF card, stick it into the Alix and have fun.
Mission accomplished in record time, box is up and running :-)