

## Appendix J – Junk

### Raspberry Pi



#### Prebuilt image

sdcard image available for the raspberry pi:

[http://felloff.net/usr/cinap\\_lenrek/9front-pi-3125.9fcca5b8ac11.img.bz2.torrent](http://felloff.net/usr/cinap_lenrek/9front-pi-3125.9fcca5b8ac11.img.bz2.torrent)

<http://sciops.net/downloads/9front-pi-3125.9fcca5b8ac11.img.bz2>

I used Richard Miller's `pi.uboot.sd.img` as a base, you may want to update the firmware on the FAT partition with files from here:

<https://github.com/raspberrypi/firmware>

older versions:

9front 95b0756087eb from Sun Oct 20 22:38:35 CES 2013

[http://intma.in/downloads/9front\\_pi.img.gz.torrent](http://intma.in/downloads/9front_pi.img.gz.torrent)

[http://intma.in/downloads/9front\\_pi.img.gz](http://intma.in/downloads/9front_pi.img.gz)

#### Installation Instructions

If you don't want to use the provided image, follow these instructions to make your own. You don't need a computer capable of running 9front, qemu will suffice.

## Image

Make a FAT partition containing all the necessary files to boot the Raspberry Pi. An easy way to do this is to use Richard Miller's `pi.uboot.sd.img` (it's small). Write it to your SD card or do the following if you're using qemu (installing to an SD card directly from qemu was **very** slow for me):

```
# set count to the size you want your image to be
# (in 512b units, see dd manpage)
dd if=/dev/zero of=9front-pi.img count=3782016

# copy minimal image to beginning of the newly created image
dd if=pi.uboot.sd.img of=9front-pi.img conv=notrunc
```

## Boot and Installation

Now boot from the 9front CD. For qemu see FQA 3.3.1 - qemu, this is what I did:

```
qemu-system-i386 \
    -enable-kvm \
    -drive file=9front-pi.img,if=virtio \
    -cdrom 9front-3125.9fcca5b8ac11.iso \
    -boot d
```

Now do the installation (see FQA 4 - 9front Installation Guide for help), use `hjfs`. Since you don't need a 9fat partition, you may want to delete the automatically created partitions and create your own (start at sector 2 or `prepdisk` will complain). Note that since you've removed 9fat, the installer will prompt you to do the `prepdisk` step again, just type the next step in manually.

After the installation you have a 9front system that only has 386 binaries. The next step is compile the arm binaries.

## Compiling for ARM

Reboot and boot from CD again, but this time use the newly created partition as root (e.g. `local!/dev/sdF0/fs`)

Compile everything:

```
cd /sys/src
objtype=arm mk install
cd 9/bcm
mk install && mk install 'CONF=picpuf'
```

And copy the kernel to the FAT partition:

```
dosrv >/dev/null >[2]/dev/null
mount -c /srv/dos /n/dos /dev/sdF0/dos
cp /arm/9pif /arm/9picpuf /n/dos
```

Then set the kernel in `/n/dos/config.txt` to `9pif` or `9picpuf` for terminal or cpu kernel and boot the image with the Raspberry Pi. At this point you can delete the 386 binaries if you don't need them.

Once booted on the Raspberry Pi, you're able to mount the dos partition with:

```
9fs pidos
```

This will mount the dos partition on `/n/pidos`, similar to how `9fs 9fat` works.