

=====

BOOTING FR-V LINUX KERNEL

=====

=====

PROVIDING A FILESYSTEM

=====

First of all, a root filesystem must be made available. This can be done in one of two ways:

(1) NFS Export

A filesystem should be constructed in a directory on an NFS server that the target board can reach. This directory should then be NFS exported such that the target board can read and write into it as root.

(2) Flash Filesystem (JFFS2 Recommended)

In this case, the image must be stored or built up on flash before it can be used. A complete image can be built using the mkfs.jffs2 or similar program and then downloaded and stored into flash by RedBoot.

=====

LOADING THE KERNEL IMAGE

=====

The kernel will need to be loaded into RAM by RedBoot (or by some alternative boot loader) before it can be run. The kernel image (arch/frv/boot/Image) may be loaded in one of three ways:

(1) Load from Flash

This is the simplest. RedBoot can store an image in the flash (see the RedBoot documentation) and then load it back into RAM. RedBoot keeps track of the load address, entry point and size, so the command to do this is simply:

```
fis load linux
```

The image is then ready to be executed.

(2) Load by TFTP

The following command will download a raw binary kernel image from the default server (as negotiated by BOOTP) and store it into RAM:

```
load -b 0x00100000 -r /tftpboot/image.bin
```

The image is then ready to be executed.

(3) Load by Y-Modem

The following command will download a raw binary kernel image across the serial port that RedBoot is currently using:

booting.txt

```
load -m ymodem -b 0x00100000 -r zImage
```

The serial client (such as minicom) must then be told to transmit the program by Y-Modem.

When finished, the image will then be ready to be executed.

## =====

### BOOTING THE KERNEL

## =====

Boot the image with the following RedBoot command:

```
exec -c "<CMDLINE>" 0x00100000
```

For example:

```
exec -c "console=ttySM0,115200 ip=:::::dhcp root=/dev/mtdblock2 rw"
```

This will start the kernel running. Note that if the GDB-stub is compiled in, then the kernel will immediately wait for GDB to connect over serial before doing anything else. See the section on kernel debugging with GDB.

The kernel command line <CMDLINE> tells the kernel where its console is and how to find its root filesystem. This is made up of the following components, separated by spaces:

(\*) console=ttyS<x>[,<baud>[<parity>[<bits>[<flow>]]]]

This specifies that the system console should output through on-chip serial port <x> (which can be "0" or "1").

<baud> is a standard baud rate between 1200 and 115200 (default 9600).

<parity> is a parity setting of "N", "O", "E", "M" or "S" for None, Odd, Even, Mark or Space. "None" is the default.

<stop> is "7" or "8" for the number of bits per character. "8" is the default.

<flow> is "r" to use flow control (XCTS on serial port 2 only). The default is to not use flow control.

For example:

```
console=ttyS0,115200
```

To use the first on-chip serial port at baud rate 115200, no parity, 8 bits, and no flow control.

(\*) root=/dev/<xxxx>

This specifies the device upon which the root filesystem resides. For example:

## booting.txt

```
/dev/nfs      NFS root filesystem
/dev/mtdblock3 Fourth RedBoot partition on the System Flash
```

(\*) rw

Start with the root filesystem mounted Read/Write.

The remaining components are all optional:

(\*) ip=<ip>::<>host>:<iface>:<cfg>

Configure the network interface. If <cfg> is "off" then <ip> should specify the IP address for the network device <iface>. <host> provide the hostname for the device.

If <cfg> is "bootp" or "dhcp", then all of these parameters will be discovered by consulting a BOOTP or DHCP server.

For example, the following might be used:

```
ip=192.168.73.12::<>frv:eth0:off
```

This sets the IP address on the VDK motherboard RTL8029 ethernet chipset (eth0) to be 192.168.73.12, and sets the board's hostname to be "frv".

(\*) nfsroot=<server>:<dir>[,v<vers>]

This is mandatory if "root=/dev/nfs" is given as an option. It tells the kernel the IP address of the NFS server providing its root filesystem, and the pathname on that server of the filesystem.

The NFS version to use can also be specified. v2 and v3 are supported by Linux.

For example:

```
nfsroot=192.168.73.1:/nfsroot-frv
```

(\*) profile=1

Turns on the kernel profiler (accessible through /proc/profile).

(\*) console=gdb0

This can be used as an alternative to the "console=ttyS..." listed above. It tells the kernel to pass the console output to GDB if the gdbstub is compiled in to the kernel.

If this is used, then the gdbstub passes the text to GDB, which then simply dumps it to its standard output.

(\*) mem=<xxx>M

Normally the kernel will work out how much SDRAM it has by reading the SDRAM controller registers. That can be overridden with this

booting.txt

option. This allows the kernel to be told that it has <xxx> megabytes of memory available.

(\*) init=<prog> [<arg> [<arg> [<arg> ...]]]

This tells the kernel what program to run initially. By default this is /sbin/init, but /sbin/sash or /bin/sh are common alternatives.

(\*) vdc=...

This option configures the MB93493 companion chip visual display driver. Please see Documentation/frv/mb93493/vdc.txt for more information.