

Booting..txt
Booting ARM Linux
=====

Author: Russell King
Date : 18 May 2002

The following documentation is relevant to 2.4.18-rmk6 and beyond.

In order to boot ARM Linux, you require a boot loader, which is a small program that runs before the main kernel. The boot loader is expected to initialise various devices, and eventually call the Linux kernel, passing information to the kernel.

Essentially, the boot loader should provide (as a minimum) the following:

1. Setup and initialise the RAM.
2. Initialise one serial port.
3. Detect the machine type.
4. Setup the kernel tagged list.
5. Call the kernel image.

1. Setup and initialise RAM

Existing boot loaders:	MANDATORY
New boot loaders:	MANDATORY

The boot loader is expected to find and initialise all RAM that the kernel will use for volatile data storage in the system. It performs this in a machine dependent manner. (It may use internal algorithms to automatically locate and size all RAM, or it may use knowledge of the RAM in the machine, or any other method the boot loader designer sees fit.)

2. Initialise one serial port

Existing boot loaders:	OPTIONAL, RECOMMENDED
New boot loaders:	OPTIONAL, RECOMMENDED

The boot loader should initialise and enable one serial port on the target. This allows the kernel serial driver to automatically detect which serial port it should use for the kernel console (generally used for debugging purposes, or communication with the target.)

As an alternative, the boot loader can pass the relevant 'console=' option to the kernel via the tagged lists specifying the port, and serial format options as described in

Documentation/kernel-parameters.txt.

3. Detect the machine type

directly.

The zImage may also be placed in system RAM (at any location) and called there. Note that the kernel uses 16K of RAM below the image to store page tables. The recommended placement is 32KiB into RAM.

In either case, the following conditions must be met:

- Quiesce all DMA capable devices so that memory does not get corrupted by bogus network packets or disk data. This will save you many hours of debug.
- CPU register settings
 - r0 = 0,
 - r1 = machine type number discovered in (3) above.
 - r2 = physical address of tagged list in system RAM.
- CPU mode
 - All forms of interrupts must be disabled (IRQs and FIQs)
 - The CPU must be in SVC mode. (A special exception exists for Angel)
- Caches, MMUs
 - The MMU must be off.
 - Instruction cache may be on or off.
 - Data cache must be off.
- The boot loader is expected to call the kernel image by jumping directly to the first instruction of the kernel image.