

ADS

PXA Linux Bootloader

Manual

Applied Data Systems

www.applieddata.net

9140 Guilford Road
Columbia MD 21046
301-490-4007

© 2003 ADS

Revision History

REV	DESCRIPTION	DATE	BY
A	Initial release	29 Dec 03	JL

1 Overview

This document is the reference manual for the bootloaders on ADS single board computers that have an Xscale (PXA) CPU running Linux.

The ADS PXA Linux Bootloader loads and boots Linux from On-Board Flash or from an ATA CF or PCMCIA card¹. The bootloader can be used to copy files from an ATA card to On-Board Flash.

The bootloader permits registers and the Linux kernel command line to be set based on input from text files. It also supports repartitioning the On-Board Flash to match the kernel command line configuration. This provides flexibility in choosing a root file system – ramdisk, cramfs, jffs2, and so forth – and convenience in setting or changing the size of the partitions.

The bootloader is stored in the On-Board Flash memory and can be reprogrammed in place (without removing any chips).

2 DIP Switches

The bootloader uses the first and second dip switches to determine the boot mode. The first switch allows or prohibits On-Board Flash updates by the bootloader and the second switch enables or disables bootloader messages on the debug port.

2.1 Secure / Flash Update Mode

Secure mode (first dip switch off) only allows Linux to boot from On-Board Flash and does not allow the On-Board Flash to be programmed during boot. These operations are possible in Flash Update mode (first dip switch on). Use secure mode to prevent the system from booting from an ATA CF or PCMCIA card.

2.2 Normal / Quiet Mode

In Normal mode (second dip switch off), the bootloader sends messages to the debug port showing the boot progress, hardware detected, version information, etc. Quiet mode (second dip switch on) suppresses this output.

3 Signal Files

Many of the operations of the bootloader are triggered by the presence of one or more signal files. When the board is in Flash Update mode, the bootloader checks for an ATA CF or PCMCIA card and if it finds one, it looks for the following signal files to decide what to do next:

- ProgramFlash.ads! Program the partitions in On-Board Flash.
- ProgramADSBootROM.ads! Program the bootloader partition in On-Board Flash.
- DontBootFromPCMCIA.ads! Ignore the card. Load and boot from On-Board Flash.

If there is no signal file, the bootloader loads and boots Linux from the zImage on the card.

¹ To boot or load from an ATA card, the first partition must be formatted with FAT12 or FAT16. This is usually the case for new cards so try it before you reformat it. Visit our website for advice on formatting your card. If you have a card that will not work, please notify us.
(Note: There are many formats that work fine once Linux is running. These requirements are only for using the bootloader to boot or program On Board Flash from an ATA card.)

4 Register Settings

Before booting Linux, the bootloader initializes the board's registers. When the board boots Linux from On-Board Flash, the bootloader also sets registers according to register settings stored with the kernel zImage (see Programming On-Board Flash).

When booting from an ATA CF or PCMCIA card, the bootloader checks for a register.txt file. If it finds one, then it sets the registers specified in the file to the values provided before loading the zImage from the card. This permits various items such as the LCD panel to be configured without modifying the kernel or the bootloader.

Each line in the file specifies a register (address) and a value to set. Addresses must be physical addresses, not virtual addresses. Values must be hexadecimal and may be 8 bit, 16 bit, or 32 bit. Comments begin with a semicolon. For example:

```
0xA0010000 0x00000042 ; set 32 bit word at 0xA0010000 to 42 hex
0xA0010000 0x0042      ; set 16 bit half word at 0xA0010000 to 42 hex
0xA0010000 0x42        ; set 8 bit byte at 0xA0010000 to 42 hex
```

When programming the On-Board Flash, the register settings in the register.txt file are stored with the zImage.

4.1 Frame Buffer Registers

In linux drivers that support it, the video driver will read the frame buffer registers set by the bootloader to determine parameters such as the bits per pixel and the display resolution. Visit our website to download register.txt files for various panels and pixel depths. We have files for the Epson frame buffer which is an option on the AGX and for the PXA frame buffer available on all our PXA boards.

Here is an example for a Sharp VGA panel connected to the PXA Frame Buffer:

```
; SharpLQ64343 panel
0x44000000 0x003018f8 ; LCCR0
0x44000004 0x0f0e0e7f ; LCCR1
0x44000008 0x182021df ; LCCR2
0x4400000c 0x03300001 ; LCCR3 (8bpp)
```

Four LCCR registers control the PXA frame buffer. Our website also has register.txt files for the SA1100 frame buffer which also uses LCCR registers. Don't mix them up. The LCCR settings for the SA1100 and the PXA frame buffers are not interchangeable.

5 Kernel Command Line

Linux kernel command line parameters may be specified in a `cmdline.txt` file. The command line is used to configure the console, root file system, MTD partitioning, etc. For more information about kernel command line parameters, visit:

<http://www.arm.linux.org.uk/armlinux/common/kernparams.html>

The bootloader uses the MTD partitioning information to determine how the On-Board Flash memory is partitioned.

When a zImage is written to the On-Board Flash (see Programming On-Board Flash), the command line is stored with it. Subsequent boots from On-Board Flash will use the stored command line.

The bootloader generates a default command line if a `cmdline.txt` file is not provided. The default command line assumes a ramdisk root file system is being used.

Tip:

To create your own command line, start by booting your board without any modifications. Mount an ATA flash card on `/mnt` and use this command

```
cat /proc/cmdline > /mnt/cmdline.txt
```

to create a `cmdline.txt` file that contains the standard (default) command line. Use this as a starting place for creating your own command line.

5.1 Standard Command Line

The kernel command line for the typical configuration is:

```
console=ttyS1,38400 rw mtdparts=pxa:256k(adslinux.rom)ro,1m(zImage)ro,3m(ramdisk.gz)ro,-(flashfs1)
ramdisk_size=8192 initrd=0xA0800000,3m root=/dev/ram
```

To specify your own kernel command line, start with the standard command line and modify it as needed. Enter the text in a single line on the first line of the `cmdline.txt` file.

The console specifies which serial device is the console and what baud rate to use. Typically, the debug port is used for this purpose and the rate is set to match the bootloader's (38400). On the BitsyX, the debug port is `ttyS0`. On other PXA boards it is `ttyS1`. (Substitute the desired serial device as needed in the examples that follow. A summary of the serial ports and the corresponding serial devices is provided on our website.)

The `rw` option makes the root file system writable (recommended).

The `mtdparts` parameters specify the On-Board Flash partitioning. The `pxa` keyword can be used beginning with Rev 4.7 of the PXA bootloader. (The keyword `sa1100` can also be used instead of `pxa` and is accepted on all PXA boards. You must use `sa1100` on PXA boards with bootloaders older than Rev 4.7.)

The parameters shown above are for the default On Board Flash partitions. If you use a `cmdline.txt` file, you only need to specify `mtdparts` if you want partitions that differ from those shown above. If you do specify your own partitions, be sure the bootloader (`adslinux.rom`) is in the first partition with a size of 256 kilobytes and the zImage is in the second partition.

The next two options tell the kernel to use and an initial ramdisk. Its uncompressed size in kilobytes is specified by `ramdisk_size` and `initrd` specifies the starting address in RAM (`0xA0800000` on PXA boards) and the (maximum) compressed size. The `root` option tells the kernel to use the ramdisk as the root file system.

5.2 Command Line Examples

Here are some examples of other useful kernel command lines. Enter the command line in a single line on the first line of the `cmdline.txt` file. The examples below are for PXA boards other than the BitsyX. The debug port on the BitsyX is `ttyS0` so replace `console=ttyS1` with `console=ttyS0` in the examples that follow to use the debug port (serial 3) on the BitsyX. Unless stated otherwise, these commands lines use a ramdisk as the root file system.

Command line to suppress kernel messages on the debug port:

```
console=null rw ramdisk_size=8192 initrd=0xA0800000,3m root=/dev/ram
```

Command line to suppress kernel messages in Quiet mode (see Quiet Mode Substitutions below):

```
console=ttyS1,38400 rw ramdisk_size=8192 initrd=0xA0800000,3m root=/dev/ram quiet_console=null
```

Command line for a 12MB ramdisk (4MB compressed) loaded from an ATA card:

```
console=ttyS1,38400 rw ramdisk_size=12288 initrd=0xA0800000,4m root=/dev/ram
```

The `ramdisk_size` is in kilobytes.

Command line for a 12MB ramdisk (4MB compressed) stored in On-Board Flash:

```
console=ttyS1,38400 rw mtdparts=pxa:256k(adslinux.rom)ro,1m(zImage)ro,4m(ramdisk.gz)ro,-(flashfs1)
ramdisk_size=12288 initrd=0xA0800000,4m root=/dev/ram
```

The `ramdisk_size` is in kilobytes.

Command line to mount a JFFS2 file system *instead* of a ramdisk as the root file system:

```
console=ttyS1,38400 rw mtdparts=pxa:256k(adslinux.rom)ro,1m(zImage)ro,-(jffs2_fs) root=/dev/mtdblock3
rootfstype=jffs2 noinitrd
```

In this example, the ramdisk partition has been eliminated altogether. The JFFS2 partition completely fills the third partition.

Tip:

You can use `mkfs.jffs2` on a Linux PC to create a JFFS2 image file containing the contents of a directory:

```
mkfs.jffs2 -e 0x40000 -p -r dir > jffs2_fs
```

The `mkfs.jffs2` program and directions for using it are available on our web site.

Tip 2:

You can create an empty jffs2 partition by booting Linux on your board and running the erase command:

```
erase /dev/mtdc6
```

Then you can mount and use the JFFS2 partition as needed:

```
mount -t jffs2 /dev/mtdb3 /flash
```

```
touch /flash/foo
```

```
umount /flash
```

Visit our website for more information about creating and using a JFFS2 flash file system.

Command line to use an ext3 file system on a microdrive in one of the slots as the root file system:

```
console=tty1,38400 ramdisk_size=3072 initrd=0xA0800000,1200k root=/dev/hda3
```

In this example, the ramdisk partition has not been eliminated but is only used as an initial ramdisk (initrd) to bootstrap mounting the file system on the microdrive as the root file system. Also in this example, the console is setup on the display and keyboard instead of the debug port.

5.3 Quiet Mode Substitutions

In Quiet mode, the bootloader will replace any console= parameter with the quiet_console= parameter in the command line. For example, if the following command line is used:

```
console=ttyS1 rw ramdisk_size=8192 initrd=0xA0800000,3m root=/dev/ram quiet_console=null
```

and the Quiet mode switch is on then the command line sent to the kernel will be:

```
XXXXXXXX=ttyS1 rw ramdisk_size=8192 initrd=0xA0800000,3m root=/dev/ram console=null
```

This prevents boot and error messages from appearing on the ttyS1 port. (If a cmdline.txt file is not used, this happens automatically.)

6 Booting from On-Board Flash

The bootloader loads and boots Linux from On-Board Flash when the board is in Secure mode or the bootloader cannot find a zImage on an ATA card (or there is a “DontBootFromPCMCIA.ads!” signal file on the card). To boot from On-Board Flash, the following apply:

- The On-Board Flash must be programmed. Ordinarily, boards are programmed in the factory. See Programming On-Board Flash for instructions on replacing this programming with your own.
- The zImage is required. It must be at located 256 kilobytes into flash memory. (The bootloader occupies the first 256 kilobytes of On-Board Flash.)
- There may be register and command line settings stored with zImage. The kernel uses a default command line if none is provided.
- A root file system is required. By default this is a ramdisk.
- If there is a partition named ramdisk.gz and no “initrd=<loc>,<size>” entry is provided in the kernel command line, then ramdisk.gz will be copied to DRAM address 0xA0800000. If this entry is provided and the loc parameter is in DRAM, then ramdisk.gz will be copied to the specified location. If “loc” is not in DRAM, the ramdisk is *not* copied to DRAM or anywhere else!
- The ramdisk_size is specified in the kernel command line, it should match the size of the uncompressed ramdisk. The bootloader ignores this parameter.

7 Programming On-Board Flash

To program the On-Board Flash memory, copy the zImage and any other files that may be needed (e.g. cmdline.txt, register.txt, ramdisk.gz, flashfs1, etc.) to an ATA CF or PCMCIA card formatted with FAT12 or FAT16. Create a “ProgramFlash.ads!” file on the card. Disable Secure mode (turn 1st dip switch on), insert the card, and reset the board.

The bootloader erases the necessary regions in On-Board Flash and copies the zImage and other files from the card to these regions. If a cmdline.txt or register.txt file is present, the information in these files is stored with the zImage.

After programming is complete, the bootloader will load and boot Linux from On-Board Flash using the new programming. Enable Secure mode, delete the signal file, or remove the card to prevent the On-Board Flash from being programmed again the next time the board boots from the ATA card.

When programming On-Board Flash from an ATA card, the following apply:

- The bootloader partition is not normally programmed. See below for additional steps necessary to program the bootloader partition.
- The bootloader determines the names of the partitions from Linux kernel command line mtd parameter or if the mtd parameter is not present, it uses the default names.
- The bootloader attempts to update each of the On-Board Flash partitions. Starting with the first partition (or skipping the first and starting with the second as is usually the case), the bootloader finds a file with the name of the partition on the ATA card and copies it to the partition. If the file is not found, the bootloader does not attempt to load any more partitions and continues with the normal boot process.
- The default partitioning, in order, is 256 kilobytes for the bootloader, 1 megabyte for the zImage, 3 megabytes for a ramdisk.gz partition, and the remainder for a flash file system (flashfs1) partition.
- The command line and register settings are stored with the zImage.
- An updated bootloader file is optional and is only programmed if an additional signal file is present (see below).

7.1 Updating the Bootloader

The PXA bootloader is stored in On-Board Flash memory so the bootloader can also be programmed. Be careful when replacing the bootloader. Failure during an update can make the board unbootable. If this happens, the board may have to be returned to our factory for reprogramming. Charges may apply.

Copy a bootloader file (available on our web site) to an ATA card. The file must be named “adslinux.rom”. Rename it if the downloaded file has a different name.

Create a “ProgramADSBootROM.ads!” file on the card *in addition to* the “ProgramFlash.ads!” file mentioned above. Disable Secure mode (turn 1st dip switch on), insert the card, and reset the board. The bootloader will erase and program the bootloader partition in On-Board Flash.

Enable Secure mode or remove the card and reset the board to execute the new bootloader.

8 Booting from an ATA PCMCIA or CF Card

To boot from an ATA PCMCIA or CF card, insert the card, disable Secure mode (turn 1st dip switch on), and reset the board. The card should be formatted with FAT12 or FAT16. The bootloader searches for the card on socket 0 first and then socket 1 if there is one.

If the bootloader finds a zImage file on the card, it loads the zImage from the card. Otherwise, it loads the zImage from On-Board Flash. Likewise, if a valid ramdisk.gz file is found, it loads the ramdisk image from the card. Finally, the bootloader boots the loaded Linux kernel.

When booting from an ATA card, the following apply:

- The zImage is required to be on the card.
- The cmdline.txt file is used if present.
- The register.txt file is used if present.
- A ramdisk.gz is not required.
- A “ramdisk_size=” command line parameter can be used to specify the size of the ramdisk for the kernel. This should match the uncompressed size of the ramdisk file. The bootloader does not use this parameter. It just passes it along to the kernel.
- An “initrd=<loc>,<size>” command line entry can be used to specify two ramdisk parameters for the kernel.

The size parameter is the size of the compressed ramdisk (which should be at least as big as the compressed ramdisk.gz file). The bootloader ignores the size.

The loc parameter is the location in DRAM where the ramdisk will be copied for decompression by kernel. If this location is not in DRAM, the ramdisk is *not* be copied to DRAM or anywhere else! The default location is 0xA0800000 on PXA boards and 0xC0800000 on SA11x0 boards.

- The bootloader will not load a file from a card that is bigger than its corresponding partition. To load a file from a card is bigger than its partition in On-Board Flash, specify the partitions using mtdparts= in the kernel command line.

9 ADS Linux Bootloader Revision History

To maintain commonality between PXA Linux bootloaders and the bootloaders for other ADS Linux products, the first PXA bootloader to be released was Version 4.5. Prior versions are not described in this document.

Version 4.5 PXA

- Supports BitsyX and AGX single board computers.

Version 4.6 PXA

- Fixes problem where only 32 MB of RAM was used when 64 MB was present.

Version 4.7 PXA

- Improves support for CF and PCMCIA cards.
- Supports Hitachi Microdrive (CF form factor hard drive).
- Adds support for the “pxa” keyword in the mtdparts parameters of kernel command line.

Version 4.8 PXA

- Adds support for Rev C BitsyX boards.
- Improves support for Hitachi Microdrive.
- Improves ATA support.
- Improves bootloader messages.

A Bootloader Operation

If you're not quite sure what will happen in some circumstance, such as having more than one signal file on a card, review this section to see exactly what steps the bootloader follows:

- 1.If the board is in Secure mode, go to step 10.
- 2.If there is no ATA PCMCIA or CF card, go to step 10.
- 3.If there is a "ProgramFlash.ads!" file, go to step 6.
- 4.If there is a "DontBootFromPCMCIA.ads!" file, go to step 10.
- 5.Go to step 9.
- 6.If there is no "ProgramADSBootROM.ads!" file, go to step 8.
- 7.Copy the adslinux.rom image to On-Board Flash.
- 8.Copy the zImage and any file system files to On-Board Flash and go to step 10.
- 9.If a zImage is on the card, load the zImage (and the ramdisk, if any) into RAM and go to step 11.
- 10.Load the zImage (and the ramdisk, if any) from On-Board Flash into RAM.
- 11.Boot the Linux kernel. The bootloader is done.

Ordinarily, the root file system is a ramdisk (compressed ramdisk.gz file) and an optional flash file system (flashfs1). See the Command Line Examples for other possibilities.