ADS Linux Bootloader Manual

Applied Data Systems

www.applieddata.net

9140 Guilford Road
Columbia MD 21046
301-490-4007

Revision History

REV	DESCRIPTION	DATE	BY
1	Initial release	04 NOV 02	JL
2	Update to ADS Linux Bootloader Ver. 4.3	03 MAR 03	JL
A	Add document number	03 MAR 03	HW
B	Update for PXA Bootloader.	29 SEP 03	JL

1 Overview

The ADS Linux Bootloader loads and boots Linux from On-Board Flash or from an ATA CF or PCMCIA card[1]. 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 on the Graphics Client Plus and Graphics Master is provided in a ROM chip (an Atmel 29V1024). On all other Linux boards (Bitsy, Bitsy Plus, BitsyX, Advanced Graphics Client, AGX, etc.), 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).

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 Linux partitions in On-Board Flash.
• ProgramADSBootROM.ads!	Program the bootloader partition in On-Board Flash.
• GotoBootROMMenu.ads!	Display the bootloader menu and prompt for a selection.
• 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 and identify the type, manufacturer, and model of the card. (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 is booted 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:

```
0xC0010000 0x00000042 ; set 32 bit word at 0xC0010000 to 42 hex
0xC0010000 0x0042     ; set 16 bit half word at 0xC0010000 to 42 hex
0xC0010000 0x42       ; set 8 bit byte at 0xC0010000 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 bits per pixel and display resolution. Visit our website to download register.txt files for various panels and pixel depths including files for the Epson frame buffer used on the Advanced Graphics Client.

4.1.1 SA1100 Frame Buffer

Five registers control the SA1110 frame buffer. Here is an example for a Sharp VGA panel:

```
; Sharp LQ64343 panel
0xB0100000  0x00000080   ; LCCR0
0xB0100020  0x2a2a1270   ; LCCR1
0xB0100024  0x182021df   ; LCCR2
0xB0100028  0x00300003   ; LCCR3
0xB0100010  0xFFFF1000   ; DBAR1 (8bpp)
```

Tip:

The register at 0xB0100010 (DBAR1) is normally a pointer to a memory location where the frame buffer look up tables and screen data reside. The first word of this memory location pointed to by DBAR1 has the number of bits per pixel encoded in bits 12 and 13 (0x0001000 for 8bpp, 0x0002000 for 16bpp). However, the driver checks the 16 most significant bits and if they are 0xFFFF, the 8bpp or 16bpp value is taken from the DBAR1 instead of indirectly.

4.1.2 PXA Frame Buffer

Four registers control the PXA frame buffer. Here is an example for a Sharp VGA panel: (Note: The LCCR settings for the SA1100 and the PXA frame buffers are not interchangeable.)

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

5 Kernel Command Line

The Linux kernel command line may be specified in a cmdline.txt file. The command line is used to configure the console, root file system, MTD partitioning, etc.

When programming the On-Board Flash, the bootloader checks the cmdline.txt file and repartitions the On-Board Flash according to the command line configuration if necessary. The command line is stored with the zImage in On-Board Flash (see Programming On-Board Flash).

The bootloader generates a command line if a cmdline.txt file is not provided. You can type

```
cat /proc/cmdline
```

after Linux starts to see the command line that is being used.

5.1 *Standard Command Lines*

Use these command lines as a starting place if you want to specify your own kernel command line. Enter the text in a single line on the first line of the cmdline.txt file.

For boards with a bootloader ROM chip, the command line for the typical configuration is:

```
console=ttySA0 rw mtdparts=sa1100:1m(zImage)ro,3m(ramdisk.gz)ro,-(flashfs1)
ramdisk_size=8192 initrd=0xC0800000,3m root=/dev/ram
```

On these boards, the zImage must be in the first partition.

For boards with the bootloader in On-Board Flash and an SA1110 CPU, the typical command line is:

```
console=ttySA0 rw mtdparts=sa1100:256k(adslinux.rom)ro,1m(zImage)ro,3m(ramdisk.gz)ro,-(flashfs1)
ramdisk_size=8192 initrd=0xC0800000,3m root=/dev/ram
```

On these boards, the bootloader must be in the first partition, the first partition’s size must be 256 kilobytes, and the zImage must be in the second partition.

For boards with an Xscale (PXA) CPU, the typical command line 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
```

On these boards, the bootloader must be in the first partition, the first partition’s size must be 256 kilobytes, and the zImage must be in the second partition.

On the BitsyX, the debug port is ttyS0. On other PXA boards it is ttyS1. The initial ramdisk starting address is 0xA0800000 on PXA boards. Be sure to substitute these values as needed in the examples that follow when using them on a PXA board. The pxa keyword in mtdparts can be used beginning with Rev 4.7 of the PXA bootloader. (The sa1100 keyword is also accepted on PXA boards.)

Tip:

> The mtdparts parameters shown above specify the default On Board Flash partitions. There are two configurations, one for boards with bootloader ROM chips and one for boards with the bootloader in On Board Flash. If you use a cmdline.txt file, you only need to include the mtdparts parameters if you are using non-standard partitioning.

5.2 *Command Line Examples*

Here are some examples of useful kernel command lines. Enter the command line in a single line on the first line of the cmdline.txt file.

Command line to suppress kernel messages on the debug port:

```
console=null rw ramdisk_size=8192 initrd=0xC0800000,3m root=/dev/ram
```

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

```
console=ttySA0 rw ramdisk_size=8192 initrd=0xC0800000,3m root=/dev/ram quiet_console=null
```

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

```
console=ttySA0 rw ramdisk_size=12288 initrd=0xC0800000,4m root=/dev/ram
```

The ramdisk_size is in kilobytes.

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

```
console=ttySA0 rw mtdparts=sa1100:1m(zImage)ro,4m(ramdisk.gz)ro,-(flashfs1)
ramdisk_size=12288 initrd=0xC0800000,4m root=/dev/ram
```

The ramdisk_size is in kilobytes. In this example, mtdparts is specified for a board with a bootloader ROM chip. If your board's bootloader is in On Board Flash, replace:

```
mtdparts=sa1100:1m(zImage)ro,4m(ramdisk.gz)ro,-(flashfs1)
```

with

```
mtdparts=sa1100:256k(adslinux.rom)ro,1m(zImage)ro,4m(ramdisk.gz)ro,-(flashfs1)
```

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

```
console=ttySA0 rw mtdparts=sa1100:1m(zImage)ro,-(jffs2_fs) root=/dev/mtdblock2 noinitrd
```

This example is for a board that has a bootloader ROM chip and is configured to eliminate the ramdisk partition. The JFFS2 partition completely fills the second partition. For boards with the bootloader in On-Board Flash, use:

```
console=ttySA0 rw mtdparts=sa1100:256k(adslinux.rom)ro,1m(zImage)ro,-(jffs2_fs) root=/dev/mtdblock3
noinitrd
```

Note that the mtdparts and the root device both differ.

Tip:

> You can use mkfs.jffs2 on a 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.

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=ttySA0 rw ramdisk_size=8192 initrd=0xC0800000,3m root=/dev/ram quiet_console=null
```

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

```
XXXXXXX=ttySA0 rw ramdisk_size=8192 initrd=0xC0800000,3m root=/dev/ram console=null
```

This prevents boot and error messages from appearing on the ttySA0 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 offset 0 on boards that have a bootloader ROM chip and 256 kilobytes into flash memory on boards with the bootloader in 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 (on PXA boards) or 0xC08000000 (on SA11x0 boards). 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 is reset.

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

- A zImage file must be on the card.
- The MTD partition files specified by name in the cmdline.txt file are copied from the card to flash partitions. If there is no cmdline.txt file on the card or there is no MTD partition information in the command line, then default values are used.
- The default partitioning, in order, is 256 kilobytes for the bootloader (only on boards without a bootloader ROM chip), 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*

For boards that have their bootloader stored in On-Board Flash memory, 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 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 Entering the Boot ROM Menu

The bootloader has several functions like dump memory, set memory and erase flash. To enter a menu to select one of these functions, create a "GotoBootROMMenu.ads!" signal file on an ATA CF or PCMCIA card formatted with FAT12 or FAT16.

Disable Secure mode (turn 1st dip switch on), insert the card, and reset the board. The bootloader will display the menu and prompt for a selection.

10 ADS Linux Bootloader Revision History

Prior to Version 4.0 there were different bootloaders for each board. None of these bootloaders used a version number higher than 3.x. Version 4.0 was selected as the first version number for the new common Linux bootloader described in this document to simplify the transition. The older bootloaders had some features in common as well as some differences. These versions are not described in this document.

Version 4.0

- Supports all ADS Linux boards.
- Supports command line and register initialization files.
- Supports Secure mode and Quiet mode.

Version 4.1

- Improves register initialization by supporting 8 and 16 bit writes.

Version 4.2

- Fixes problem that prevented booting from FAT16 that mis-ID themselves as FAT12.
- Removes limit on size of flash partitions that may be programmed.

Version 4.3

- Improves supports for 3.3V PCMCIA and CF cards by providing 3.3V power instead of 5V (except on the GC+ which only provides 5V). Most 3.3V cards are designed to operate on 5V safely. Cards that require 3.3V are keyed to prevent insertion in a 5V only slot.
- Improves coordination of Quiet mode with the kernel command line.
- Supports new board IDs for Bitsy+ and Advanced Graphics Client (AGC) and a new version of the AGC CPLD.

Version 4.4

- Not released.

Version 4.5

- Improves PCMCIA reset and checks for CIS data buffer overflow.

Version 4.5 PXA

- Supports BitsyX and AGX

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.

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 13.
2. If there is no ATA PCMCIA or CF card, go to step 13.
3. If there is a “GotoBootROMMenu.ads!” file, got to step 15.
4. If there is a “ProgramFlash.ads!” file, go to step 6.
5. If there is a “DontBootFromPCMCIA.ads!” file, go to step 13.
6. If the bootloader is not stored in On-Board Flash, go to step 10.
7. If there is no “ProgramADSBootROM.ads!” file, go to step 10.
8. If there is no adslinux.rom image, go to step 15.
9. Copy the adslinux.rom image to On-Board Flash.
10. If there is no zImage, go to step 15.
11. Copy the zImage and any file system files to On-Board Flash and go to step 13.
12. If a zImage is on the card, load the zImage and any root file system from the card and go to step 14.
13. Load the zImage and any root file system from On-Board Flash.
14. Boot the Linux kernel. The bootloader is done.
15. Display the bootloader menu and prompt for a selection.

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.