



Amlogic Linux User Guide (1)

Boot Up

AMLOGIC, Inc.

3930 Freedom Circle

Santa Clara, CA 95054

U.S.A.

www.amlogic.com

AMLOGIC reserves the right to change any information described herein at any time without notice. AMLOGIC assumes no responsibility or liability from use of such information.

目 录

第一章 概述	1
1.1 关于 AMLOGIC LINUX.....	1
1.2 关于 AMLOGIC UBOOT.....	1
1.3 调试环境.....	1
1.3.1 平台	1
1.3.2 PC	2
1.3.3 Linux Server	2
1.4 TFTP 协议和 SERVER.....	2
1.5 OPEN LINUX 网站.....	3
第二章 烧录并启动 UBOOT	4
2.1 UBOOT 烧录包.....	4
2.2 准备工作.....	4
2.3 烧录 UBOOT.....	5
2.3.1 启动串口通讯和 tftpServer	5
2.3.2 通过 JTAG 加载 uboot	5
2.3.3 配置网络参数	6
2.3.4 下载 uboot.bin 并烧录到 SPI	6
2.3.5 其他加载 uboot.bin 的方式	7
2.4 UBOOT 的自动启动.....	7
第三章 配置并启动 LINUX	8
3.1 LINUX KERNEL 烧录包.....	8
3.2 KERNEL IMAGE 的不同加载方式.....	8
3.3 通过网络加载 KERNEL.....	8
3.3.1 准备工作	8
3.3.2 启动串口通讯和 tftpserver	8
3.3.3 启动平台并配置网络参数	9
3.3.4 执行命令加载 kernel	9
3.4 加载文件系统.....	9
3.4.1 文件系统加载方式	10
3.4.2 准备工作	10
3.4.3 在 uboot 配置启动参数	10
3.4.4 文件系统的自动加载	10
3.5 配置 LINUX 自动启动.....	10

第一章 概述

1.1 关于 Amlogic Linux

Amlogic Linux 指的是 Amlogic Inc.的嵌入式 Linux 系统 (kernel 版本 2.6.26)。它可以运行在 Amlogic 的 apollo, apollo-h 等平台上。Amlogic 将来的其他平台,也都支持运行 Linux。

本文档主要介绍如何在 Amlogic 的硬件平台上启动 Linux。

如不特殊说明,本文档中提到的 Linux 指的都是 Amlogic Linux 系统。

1.2 关于 Amlogic uboot

Amlogic Linux 要通过 Amlogic uboot 启动。

U-Boot, 全称 Universal Boot Loader, 是遵循 GPL 条款的开放源码项目。Amlogic uboot 是在 U-boot 2009.03 版本基础上移植过来的。

uboot 烧录在平台的 SPI Flash 中,具体烧录办法请参考第二章。

关于 Amlogic uboot 功能、特性、操作命令等等的详细介绍,请参考《uboot 说明.doc》。

如不特殊说明,本文档中提到的 uboot 指的都是 Amlogic uboot。

1.3 调试环境

1.3.1 平台

根据 CPU 芯片的不同, Amlogic 支持 linux 的平台有 apollo, apollo-h 等等,与之相应,uboot 和 linux 的 img 文件有多个版本(每个平台一个)。在使用时[请注意根据平台选择版本](#)。

平台要运行 Linux,一般需要 JTAG、UART、Ethernet 三个接口,分别用来连接 PC、网络交换机(并通过网络交换机连接 PC 和 Server),示意如下:

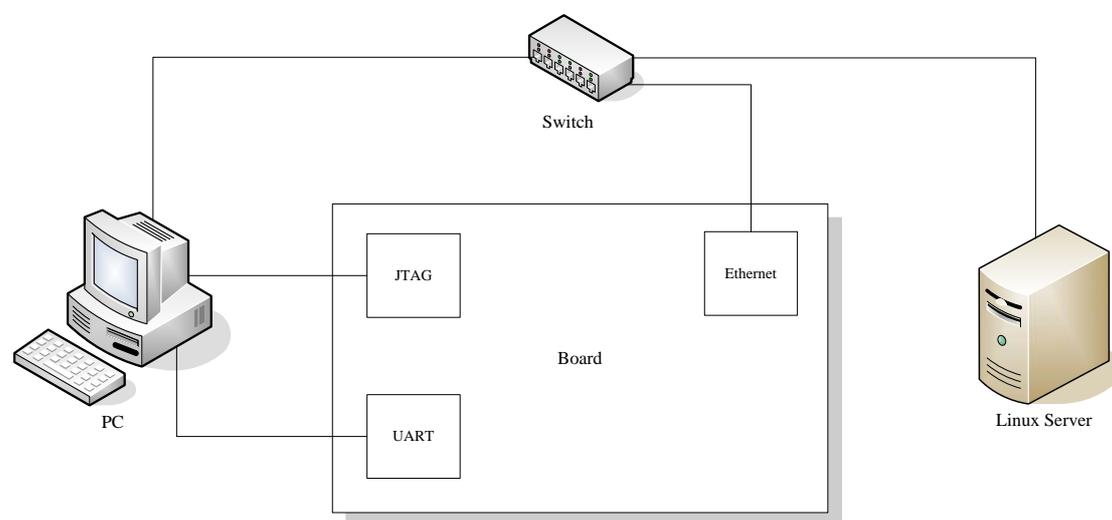


图 1-1. 硬件接口

其中:

JTAG 接口用来连接 PC 和 Board。烧录并运行 uboot、调试应用程序等, 都可以通过 JTAG 接口来完成。为此, 在 PC 端需要安装 JTAG 驱动, 并需要安装 Metaware Devkit for ARC。

UART 接口是通过串口连接 PC 和 Board。UART 接口主要用来调试 linux, 为此, 需要在 PC 端安装串口通讯工具 (比如 SecureCRT 等)。

Ethernet 接口主要用于网络通讯。一般情况下, 为了调试的方便, 可以把 kernel, file system 等 firmware 放到 Linux Server 或者 PC 上, 并使用 Ethernet 接口通过 TFTP 协议(参见 1.4)远程访问和加载。

1.3.2 PC

PC 端需要安装

- JTAG 驱动
- Metaware Devkit for ARC
- 串口通讯程序 (如 SecureCRT 等), 其中, 串口参数如下 (以 SecureCRT 为例):

Port: (视 PC 的具体接口而定, 一般是 COM1)

Baud rate: 115200

Data bits: 8

Parity: None

Stop bits: 1

Flow Control: None

Serial break: (视具体情况, 一般可以选择默认的 100ms)

- TFTP Server, 用来支持 Board 通过以太网下载 firmware 等。参见 1.4
- Linux Server 的远程访问工具 (如 SecureCRT)

1.3.3 Linux Server

需要安装:

- Linux ...
-
-

1.4 TFTP 协议和 Server

Amlogic Linux 在平台和 PC 之间、在平台和 Linux Server 之间, 都使用 TFTP 协议进行网络通讯。

TFTP (Trivial File Transfer Protocol,简单文件传输协议) 是 TCP/IP 协议族中的一个用来在客户机与服务器之间进行简单文件传输的协议, 提供不复杂、开销不大的文件传输服务。它比常用的与 FTP 协议要简单得多。

为了与平台通讯, PC 上需要安装支持 TFTP 协议的 Server。如 cisco tftp server 等等, 在 Internet 上有很多, 大家可以选择自己喜欢的下载。

1.5 Open Linux 网站

<http://openlinux.amlogic.com/> is Amlogic's Linux website; all the release packages will public here.

The site is currently split into six sections:

/Boot Loader

/Development Tools

/File Systems

/Kernel Info

/Platform Info

/Reference

在这个平台可以进行代码修改、Lib 发布、资料提交、文档修改、经验交流、Bug 反馈, 等等, 欢迎大家使用。

第二章 烧录并启动 uboot

在启动 Linux 之前，要首先确保 uboot 已经烧录并且可以正常启动。

本章主要描述如何往“裸板”上烧录 uboot。Board 上已经有可以正常工作的 uboot 的用户可以直接跳过本章。

本章只对涉及到的 uboot 的操作命令进行介绍，更多操作命令请参考《uboot 说明.doc》。

2.1 Uboot 烧录包

Uboot 的烧录包（当前版本是：uboot-2009-10-12-V372.7z）可以从 Open Linux 网站下载。烧录包中包括 apollo, apollo-h 等多个版本（请用户根据自己平台选用适用版本）。每个版本由以下文件组成：

- **uboot.bin**: bootloader 的烧写文件，烧录 uboot 指的就是把这个文件烧写到 Board 的 SPI Flash 中的指定位置。相应的，Board 启动时候，会从 SPI 把 bootloader 加载（并解压）到内存中的指定位置并开始执行，这也就是所谓的“启动 uboot”。
- **uboot**: bootloader 的 img 文件，可以通过 JTAG 接口把这个 img 文件加载到 Board 内存中的指定位置并执行。要往“裸板”烧录 uboot 的时候，这是必须的。
- **init.cmd**: CPU 寄存器的初始值集合。在加载 uboot 的 img 文件的时候，用来批量初始化寄存器。
- **uboot.bat**: 通过 JTAG 加载 img 文件的批处理命令。

```
@echo off

set METAWARE_INSTALL_DIR=c:\ARC\MetaWare
set CYGWIN_INSTALL_DIR=c:\cygwin
set
PATH=%METAWARE_INSTALL_DIR%;%METAWARE_INSTALL_DIR%\arc\bin;%CYGWIN_INSTALL_DIR%\bin;%PATH%
mdb -jtag ./u-boot -chipinit=init.cmd -OK
```

图 2-1. Uboot.bat 示例

2.2 准备工作

确保 PC 安装了 Metaware Devkit 以及 JTAG 驱动并且可以正常工作。

确保 PC 上安装了 SecureCRT 或者其他串口通讯工具并且可以正常工作。

确保 PC 上安装了 tftpServer 并且可以正常工作。

从 Open Linux 网站下载 uboot 烧录包到 PC 并解压，并把适用版本的 uboot.bin 放在

tftpserver 的工作目录。

用串口线、JTAG 正确连接 PC 和 Board。注意接口不要插反了。此外，大部分 Board 上的串口电压和 PC 不同，而且没有 RS-232 接口，需要通过相应的转接板进行连接。

2.3 烧录 uboot

Uboot 烧录在平台的 SPI Flash 中。往 SPI 烧录 uboot 要使用 JTAG，相应的，在 PC 端要使用 Metaware Devkit。

2.3.1 启动串口通讯和 tftpServer

启动 SecureCRT，并 Connect 相应接口（如 COM1）。

启动 tftpserver，并确保 uboot.bin 文件在其工作目录。

2.3.2 通过 JTAG 加载 uboot

在 PC 上运行 uboot.bat，会打开 Metaware Debugger 并把 bootloader 的 img 加载到 Board 的内存，示意如下：

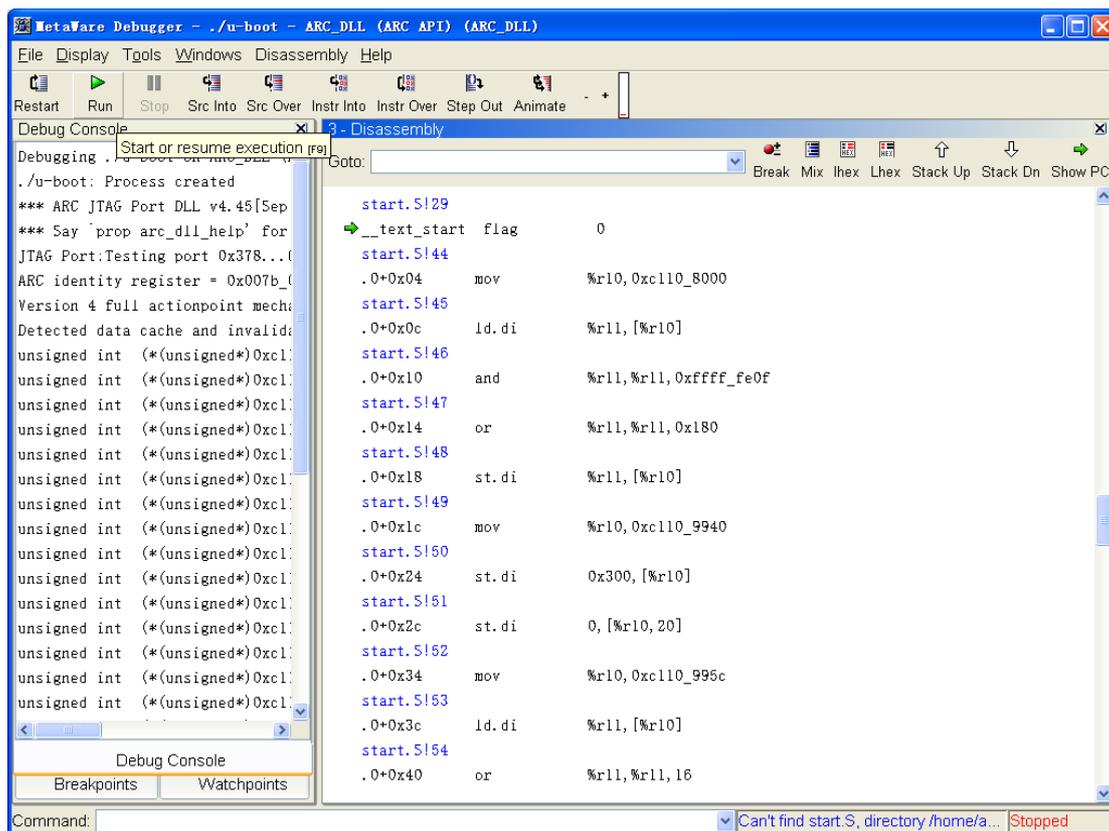


图 2-2. 加载 uboot 示例

然后点击 Run 启动 uboot，在串口通讯工具上可以系统启动的相关消息，示意如下：

```

Serial-COM1 - SecureCRT
File Edit View Options Transfer Script Tools Help
Serial-COM1
Global data pointer 81043840

U-Boot 2009.03-svn372 (Oct 12 2009 - 18:23:30)

running at 1560000000HZ
env_relocate[234] offset = 0x0
env_relocate[252] malloced ENV at 81096240
SF: Got idcode c2 20 15
SF: Detected M25X16 with sector size 4096, total 2097152 bytes
In:  serial
Out:  serial
Err:  serial
Global data pointer 81043840

U-Boot 2009.03-svn372 (Oct 12 2009 - 18:23:30)

running at 1560000000HZ
MODULE_SDIIC io 0 error! Use default config.
SD/SDHC/MMC: 0
MODULE_ETHERNET io 0 error! Use default config.
MODULE_ETHERNET io 0 error! Use default config.
MAC address is 00:01:02:03:04:75
duplex
100m
Apollo EMAC

NAND Device found: Samsung
NAND Device 0: NAND 128MiB 3.3V 8-bit, block size 128 KiB

Hit any key to stop autoboot: 0
MBX AML7266 A1H V1.0 # █

Ready          Serial: COM1    9, 24 33 Rows, 84 Cols VT100    NUM

```

图 2-3. 串口消息示例 (uboot 启动以后)

注意：启动 uboot 后，下文中所述的**操作命令**都是在串口通讯工具（如 SecureCRT）的命令行执行的。

2.3.3 配置网络参数

大致分为以下这些步骤：

1. 执行 print 命令，查看环境变量
2. 执行 setenv serverip 具体 IP 地址，配置 tftp server 的地址
3. 执行 setenv loadaddr 82000000，设置 uboot.bin 的加载内存位置为 0x82000000
4. 执行 savenv，保存上述修改

2.3.4 下载 uboot.bin 并烧录到 SPI

大致分为以下这些步骤：

1. 执行 tftpboot 82000000 uboot.bin，从 tftp server 下载 uboot.bin 文件到内存的指定位

置 (0x82000000)

2. 执行 `sf probe 2`, 探测 SPI Flash
3. 执行 `sf erase 0 60000`, 擦除 SPI 的从 0 开始的 0x60000 个字节 (384k)
4. 执行 `sf write 82000000 0 60000`; 把内存中从 0x82000000 开始的 0x60000 个字节写入 SPI

注意: 以上命令可以在一个命令行里批量执行, 示意如下:

```
tftpboot 82000000 u-boot.bin:sf probe 2:sf erase 0 60000:sf write 82000000 0 60000
```

2.3.5 其他加载 uboot.bin 的方式

上面介绍的方法中, uboot.bin 是通过网络并使用 tftp 协议下载到 memory 的。

Uboot 还支持从 USB, MMC/SD 卡等设备上加载 uboot.bin, 具体请参考《uboot 说明.doc》。

2.4 uboot 的自动启动

完成 2.3 后, uboot 就已经烧录到平台了。

在平台每次断电重启后, uboot 都会自动启动。

第三章 配置并启动 Linux

本章介绍在有 uboot 并且可以正常工作的平台上烧录 Linux Kernel、以及加载文件系统的方法。如果平台上没有 uboot，请参考第二章首先烧录 uboot。

3.1 Linux Kernel 烧录包

Linux Kernel 的烧录包（当前版本是：k425）可以从 Open Linux 网站下载。烧录包由以下文件组成：

- vmlinux: kernel 的 elf 文件，需要使用 JTAG 烧录（本文档不做详细介绍）
- System.map: 与 elf 文件相对应的 map 文件，调试的时候使用。
- ucImage: 使用 ucl 压缩的 kernel img 文件。
- uImage: 通过 gzip 压缩的 kernel img 文件。

3.2 Kernel Image 的不同加载方式

在 uboot 正常启动后，可以采用以下几种方式加载 img：

- 通过以太网口和 tftp 协议，从 PC（或者 LinuxServer）上加载
- 把 img 写入 NAND Flash，从 NAND Flash 加载
- 把 img 写入 USB 设备，从 USB 加载
- 把 img 写入 SD/MMC 等外置设备，从外置设备加载
- 等等

本章主要介绍第一种方式。

3.3 通过网络加载 Kernel

3.3.1 准备工作

确保 PC 上安装了 SecureCRT 或者其他串口通讯工具并且可以正常工作。

确保 PC 上安装了 tftpServer 并且可以正常工作。

从 Open Linux 网站下载适用版本的 kernel 到 PC 并解压，并把 img 文件（如 ucImage）放在 tftpserver 的工作目录。

确保平台烧录了 uboot 并且可以正常工作。

用串口线正确连接 PC 和 Board。注意接口不要插反了。此外，大部分 Board 上的串口电压和 PC 不同，而且没有 RS-232 接口，需要通过相应的转接板进行连接。

3.3.2 启动串口通讯和 tftpserver

3.4.1 文件系统加载方式

目前，文件系统通过网络从 Linux Server 加载。

将来，可以考虑把文件系统烧录到 NAND、USB 设备或者 MMC/SD 卡，然后从这些设备加载。

3.4.2 准备工作

1. Linux Server 的准备工作，（创建用户，。。。）
2. 从 Open Linux 网站下载文件系统压缩包，放到当前用户的工作目录。

3.4.3 在 uboot 配置启动参数

启动平台进入 uboot，在命令行设置启动参数，示意如下：

```
MBX AML7266 A1H V1.0 # setenv bootargs root=/dev/nfs nfsroot=
10.68.11.57:/home/breezecheng/rootfs/rootfs rw noinitrd init=/init
ip=10.68.11.73:10.68.11.1:10.68.11.1:255.255.255.0:target:eth0:off
console=ttyS0,115200 mac=00:11:65:34:43:22
MBX AML7266 A1H V1.0 # saveenv
Saving Environment to SPI Flash...
Erasing SPI flash...Erase:0
Erase:1000
SF: MXIC: Successfully erased 33554432 bytes @ 0x1f0000
```

图 3-2. 配置 bootargs 示例

上述参数中，指定 filesystem 位于 Server(10.68.11.57)的/home/breezecheng/rootfs/rootfs 目录。其他参数为启动时候的 ip 地址设置等。

3.4.4 文件系统的自动加载

正确配置 bootargs 参数并保存后，下次再启动 Linux 的时候，就会自动从指定位置加载文件系统。

3.5 配置 Linux 自动启动

可以通过 uboot 的 bootcmd 和 bootdelay 两个环境变量，让 Linux 随着平台自动启动。

- setenv bootdelay 3 //设置 uboot 启动 3 秒后执行 bootcmd
- setenv bootcmd tftpboot ucImage //bootcmd 的执行效果是通过 tftp 协议加载 ucImage
- saveenv