

AM335x 的网口启动及 Flash 烧写用户指南

概述:

鉴于 AM335x 可以通过网口启动的方式将 MLO, u-boot.img, kernel 通过 TFTP 的方式下载至开发板, 并可通过 NFS 的方式完成文件系统的挂载, 因此, 对最小系统板而言, 仅需网口便可完成整个 Linux 的启动。本文基于网口的启动方式, 对网口启动的操作流程进行描述, 并着重介绍了如何使用网口启动方式进行 flash 的烧写。由于使用该方式进行 flash 烧写的操作是在 UBoot 环境中以脚本的方式实现, 因此, 用户可自定义具体的操作流程, 灵活性较好。

PC 环境:

主机: Windows 7

虚拟机: Virtual Box(version 4.1.12), Ubuntu 10.04.

Linux SDK 版本: 06.00.00.00

AM335x 开发板平台: GPEVM.

网口启动配置流程:

1. 在 Ubuntu 中安装 TFTP 服务和 NFS 服务, 确保可以正常 TFTP download 和挂载 NFS。(网上关于如何 TFTP kernel 以及挂载 NFS 的资料较多, 因此, 本文主要针对 MLO (SPL) 和 u-boot 阶段的设计进行描述, 对 kernel 和 NFS 的加载操作将不再赘述)。
本示例中, TFTP 目录为"/tftpboot", NFS 目录为"/nfs_share"。

2. 在 Ubuntu 中进行 DHCP 服务的安装和配置:

本示例中, 通过以下步骤将本机建成主机 IP 为 192.168.1.1 的服务器, 地址空间为 192.168.1.100 至 192.168.1.199。

- a. 使用 root 账户登录 Ubuntu, 使用该命令进行 DHCP 服务的安装:

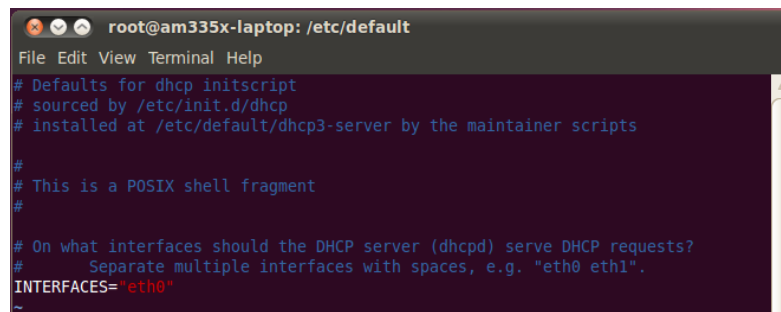
apt-get install dhcp3-server (本文中粗斜体为命令行输入内容, 下同)

- b. 增加 dhcp 服务的网卡端口 (本例中使用的网口为 eth0)。

安装完成后, 进入/etc/default 目录, 修改 dhcp3-server 文件:

- i. **vim /etc/default/dhcp3-server**

- ii. 在 INTERFACES="" 的参数中增加 eth0



```
root@am335x-laptop: /etc/default
File Edit View Terminal Help
# Defaults for dhcp initscript
# sourced by /etc/init.d/dhcp
# installed at /etc/default/dhcp3-server by the maintainer scripts
#
# This is a POSIX shell fragment
#
# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
# Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACES="eth0"
```

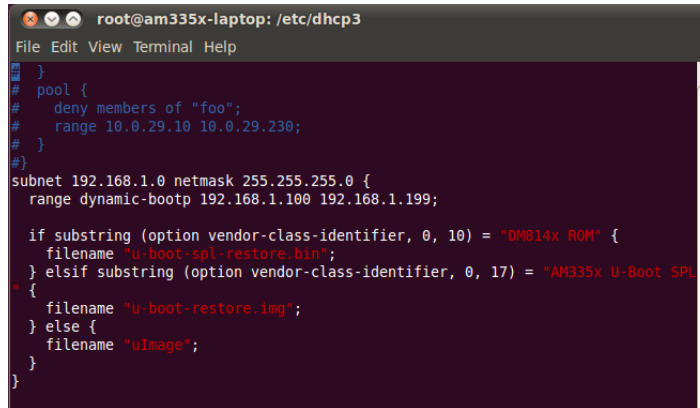
- iii. 保存后退出.

- c. 增加地址空间定义。修改“/etc/dhcp3/dhcpd.conf”。

i. vim /etc/dhcp3/dhcpd.conf

- ii. 在文件的末尾，加入以下信息：

```
subnet 192.168.1.0 netmask 255.255.255.0 {
    range dynamic-bootp 192.168.1.100 192.168.1.199;
    if substring (option vendor-class-identifier, 0, 10) = "DM814x ROM" {
        filename "u-boot-spl-restore.bin";
    } elsif substring (option vendor-class-identifier, 0, 17) = "AM335x U-Boot SPL" {
        filename "u-boot-restore.img";
    } else {
        filename "ulmage";
    }
}
```



- iii. 保存并退出。

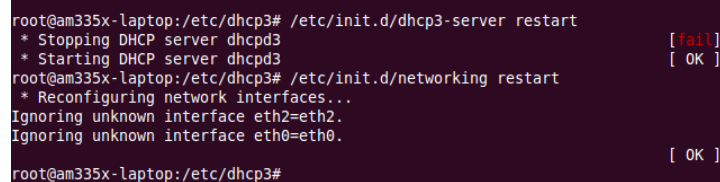
- d. 设置本机的 eth0 网卡 ip 为 192.168.1.1:

ifconfig eth0 192.168.1.1

- e. 重启 DHCP 服务:

/etc/init.d/dhcp3-server restart

/etc/init.d/networking restart



至此，Ubuntu 的 DHCP 服务建立已经完成。

注：DHCP 的服务，Ubuntu12.04 亦可以使用 isc-dhcp-server。

步骤 a 中对应的命令：apt-get install isc-dhcp-server;

步骤 b 中对应的文件为/etc/default/isc-dhcp-server;

步骤 c 中对应的文件为/etc/dhcp/dhcpd.conf;

步骤 e 中，对应的命令为/etc/init.d/isc-dhcp-server restart。

3. Linux 开发包对网口启动的相关设置:

- a. 在 Linux SDK 6.0 中, 在 uboot 中使用 am335x_evm 配置编译出 UBoot 的相关文件, 可参照链接: http://processors.wiki.ti.com/index.php/AM335x_U-Boot_User%27s_Guide
- b. 编译完成后, 将生成的 spl 目录下的 u-boot-spl.bin 文件拷贝至 TFTP 文件夹下, 并重命名为 u-boot-spl-restore.bin:
**cp ~/ti-sdk-am335x-evm-06.00.00.00/board-support/u-boot-2013.01.01-
psp06.00.00.00/spl/u-boot-spl.bin /tftpboot/u-boot-spl-restore.bin**
- c. 类似的, 将生成的 spl 目录下的 u-boot.img 文件拷贝至 TFTP 文件夹下, 并重命名为 u-boot-restore.img:
**cp ~/ti-sdk-am335x-evm-06.00.00.00/board-support/u-boot-2013.01.01-
psp06.00.00.00/u-boot.img /tftpboot/u-boot-restore.img**
- d. 至此, 主机的配置已经全部完成。使用网线将 AM335x 的开发板的网口 1 连接至 PC 机, 将 AM335x 的启动模式设置为带有 EMAC1 的模式 (如 SYS[4:0]=01010), 即可给开发板上电, 进行网口启动。

注意: 当使用 EMAC 启动时, 不要使用网线直连板子和电脑, 使用交换机或者路由器 (关闭 DHCP 服务), 确保网段内, 没有其他的 DHCP 服务器。AM335x 目前仅支持网口 1 的启动。

参考: u-boot-am33x/doc/am335x.net-spl/README. AM335x TRM

4. Linux 开发包对网口烧写 flash 的相关设置:

- a. 类似于 3 的步骤 a、b、c, 但在 uboot 编译时, 使用 am335x_evm_restore_flash 配置编译出 UBoot 的相关文件, 此时得到的 u-boot-spl.bin 和 u-boot.img 重命名后即用于网口烧写的启动文件。使用该配置编译生成的 uboot 与之前不同的是, 在启动后, 会调用脚本文件 debrick.scr, 进行 flash 的烧写操作。
- b. 脚本文件 debrick.scr 的制作 (以 NAND flash 作为示例):
 - i. 脚本文件 debrick.scr 的源文件, 可以参考 u-boot 文件夹下的 /doc/am335x.net-spl/debrick-nand.txt。源文件中可使用 UBoot 阶段的相关命令对 NAND flash 进行操作。类似的, 也可以对 SPI flash 进行操作。
 - ii. 使用 mkimage 工具将源文件 debrick-nand.txt 制作成为 debrick.scr (mkimage 位于 u-boot 文件夹下的 tools 文件夹中), 命令如下
/mkimage -A arm -O U-Boot -C none -T script -d debrick-nand.txt debrick.scr
 - iii. 将 debrick.scr 拷贝到 TFTP 文件夹中。在启动时, 会在 uboot 中自动调用。
- c. 将需要烧写到 NAND FLASH 的 MLO 和 u-boot.img 文件拷贝到 TFTP 文件夹中。
- d. 完成以上各步骤后, 即可设置 AM335 的启动模式并连接网线进行网口烧写 flash。

NAND FLASH 烧写成功示例:

```
cpsw Waiting for PHY auto negotiation to complete... done
link up on port 0, speed 100, full duplex
BOOTP broadcast 1
DHCP client bound to address 192.168.1.104
link up on port 0, speed 100, full duplex
Using cpsw device
TFTP from server 192.168.1.1; our IP address is 192.168.1.104
Filename 'debrick.scr'.
Load address: 0x80000000
Loading: #
          28.3 KiB/s
done
Bytes transferred = 389 (185 hex)
## Executing script at 80000000

NAND erase chip: device 0 whole chip
Erasing at 0xffe0000 -- 100% complete.
OK
link up on port 0, speed 100, full duplex
Using cpsw device
TFTP from server 192.168.1.1; our IP address is 192.168.1.104
Filename 'ML0'.
Load address: 0x81000000
Loading: #####
          276.4 KiB/s
done
Bytes transferred = 100549 (188c5 hex)
link up on port 0, speed 100, full duplex
Using cpsw device
TFTP from server 192.168.1.1; our IP address is 192.168.1.104
Filename 'u-boot.img'.
Load address: 0x81080000
Loading: #####
          372.1 KiB/s
done
Bytes transferred = 364376 (58f58 hex)

NAND write: device 0 offset 0x0, size 0x260000
2490368 bytes written: OK
U-Boot#
```

SPI FLASH 烧写成功示例:

```
USB Host mode controller at 47401800 using P10, IRQ 0
Net: <ethaddr> not set. Validating first E-fuse MAC
cpsw, usb_ether
Hit any key to stop autoboot: 0
cpsw Waiting for PHY auto negotiation to complete... done
link up on port 0, speed 100,
          BOOTP broadcast 1
DHCP client bound to address 192.168.1.104
link up on port 0, speed 100, full duplex
Using cpsw device
TFTP from server 192.168.1.1; our IP address is 192.168.1.104
Filename 'debrick.scr'.
Load address: 0x80000000
Loading: #
          32.2 KiB/s

Bytes transferred = 434 (1b2 hex)
## Executing script at 80000000
SF: Detected W2516 with page size 4 KiB, total 8 MiB
link up on port 0, speed 100, full duplex
Using cpsw device
TFTP from server 192.168.1.1; our IP address is 192.168.1.104
Filename 'ML0 spi'.
Load address: 0x80200000
Loading: #####
          359.4 KiB/s
done
Bytes transferred = 100549 (188c5 hex)
link up on port 0, speed 100, full duplex
Using cpsw device
TFTP from server 192.168.1.1; our IP address is 192.168.1.104
Filename 'u-boot.bin'.
Load address: 0x80200000
Loading: #####
          368.2 KiB/s
done
Bytes transferred = 364152 (58e78 hex)
SPI FLASH is written done!
U-Boot#
U-Boot#
```

By Steven Liu Jan.8th.2014