

---

# 达芬奇 DM6467 评估板 系统软件平台

构  
建  
方  
法

---

Revision Table				
Date	Rev.	Author	Content of revision	Approval
1.0	2009-09-01	Andy,LIU	Original	

TIRVIDEO

---

1 系统介绍.....	4
2 开发环境的建立.....	4
3 网络文件系统的建立.....	5
4 tftp 服务的建立.....	5
5 UBOOT 的使用.....	6
6 启动方式说明.....	8
7 硬盘的恢复.....	9
8 uboot 和 kernel 的编译.....	9
9 在 nand flash 上使用 uboot 烧写 kernel.....	10

TIRVIDEO

---

# 1 系统介绍

建立 DM6467 开发平台是进行软件研发的基础，DM6467 EVM 以 LINUX 为操作系统，文件系统可以放置在 FLASH 中，也可以放置在硬盘中，可以根据具体的应用进行选取。本文告诉用户如何构建自己的 DM6467 的开发平台。

## 2 开发环境的建立

先建立 linux 开发环境，如 redhat，然后准备好以下文件：

- dvsdk\_setuplinux\_#\_#\_#\_#.bin
- mvl\_4\_0\_1\_demo\_lsp\_setuplinux\_#\_#\_#\_#.bin
- xdc\_setuplinux\_#\_#\_#\_#.bin
- bios\_setuplinux\_#\_#\_#\_#.bin
- TI-C6x-CGT-v#.#.#.#.bin
- data.tar.gz
- SoCAnalyzer\_#.#.#.#.exe

将上述四个文件拷贝到/tmp 路径下。

确认刚刚安装好的linux的X图像显示没有问题，并且建立DISPLAY的环境变量，例如：

csh下命令：

```
host $ setenv DISPLAY cnabc0314159d1:0
```

ksh下命令：

```
host $ export DISPLAY=cnabc0314159d1:0
```

第一步：进入/tmp 路径下，依次执行下面两个程序：

```
host $ ./mvl_4_0_1_demo_sys_setuplinux.bin
```

```
host $ ./mvl_4_0_1_demo_target_setuplinux.bin
```

```
host $ ./mvl_4_0_1_demo_lsp_setuplinux_#_#_#_#.bin
```

软件会开始自解压过程，其安装的默认路径为/opt。

第二步：进入/opt/mv\_pro\_4.0.1，会发现有以下两个文件：

- mvltools4.0.1-no-target.tar.gz
- mvl4.0.1-target\_path.tar.gz
- DaVinciLSP#\_#\_#\_#.tar.gz

将DaVinciLSP#\_#\_#\_#.tar.gz也拷贝到/opt/mv\_pro\_4.0.1下，然后开始按照下面的方式进行自解压：

```
host $ tar xzf mvltools4.0.1-no-target.tar.gz
```

```
host $ tar xzf mvl4.0.1-target_path.tar.gz
```

```
host $ tar xzf DaVinciLSP#_#_#_#.tar.gz
```

第三步：进入/tmp 路径下，执行下面命令：

```
host $ ./dvsdk_setuplinux_#_#_#_#.bin
```

---

软件会开始自解压过程，其安装的默认路径为/opt。

最后，在/root 下，以 root 身份打开.bashrc，设置以下路径：

```
PATH="/opt/mv_pro_4.0.1/montavista/pro/devkit/arm/v5t_le/bin:  
/opt/mv_pro_4.0.1/montavista/pro/bin:  
/opt/mv_pro_4.0.1/montavista/common/bin:$PATH"
```

设置好路径后，重新启动 LINUX，开发环境安装完成。

### 3 网络文件系统的建立

在研发中，为了节省开发时间，常常用到网络文件系统，本节以/root/armfs 为网络文件系统路径，PC 机的 IP 为 192.168.0.60，DM6467 EVM 板的地址为 192.168.0.32 为例子，建立网络文件系统的开发环境。

第一步：在/root 下建立 armfs 目录，作为网络文件系统的根目录。

第二步：执行以下命令：

```
host $ cp -a /opt/mv_pro_4.0/montavista/pro/devkit/arm/v5t_le/target/*  
/root/armfs
```

将网络系统所需要的文件拷贝到/root/armfs 下，拷贝可能要花比较长的时间，拷贝完成后，可以和/opt/mv\_pro\_4.0/montavista/pro/devkit/arm/v5t\_le/target 路径下的文件对照一下，检查是否正确。

第三步：设置/etc/exports 文件，输入

```
/root/armfs *(rw,no_root_squash,no_all_squash,sync)
```

然后保存推出。

第四步：打开网络文件系统服务，执行下面命令：

```
host $service nfs restart.
```

第五步：对DM6467 EVM正确上电后，进入uboot，设置以下变量：

```
TirDavinci_EVM# setenv ipaddr 192.168.0.32
```

```
TirDavinci_EVM# setenv nfsroot 192.168.0.60:/root/armfs
```

```
TirDavinci_EVM# setenv bootargs console=ttyS0,115200n8 noinitrd rw  
ip=192.168.0.32:192.168.0.60 root=/dev/nfs nfsroot=$(nfsroot),nolock  
mem=120M davincihd_capture.channel0_numbuffers=4
```

然后执行 boot 重新启动，如果设置正确并且 KERNEL 没有问题，网络文件系统会正常启动，如果不能正常启动，请检查设置步骤。

### 4 tftp 服务的建立

如果 tftp 没有安装，请安装 tftp。

第一步：修改/etc/xinetd.d/tftp 如下：

```
service tftp  
{  
    disable = no
```

---

```
socket_type      = dgram
protocol         = udp
wait             = yes
user             = root
server           = /usr/sbin/in.tftpd
server_args      = -s /tftpboot -c
per_source       = 11
cps              = 100 2
flags            = IPv4
}
```

第二步：保存后执行如下命令：

```
host $service iptables stop
```

```
host $service xinetd restart
```

tftp 启动完成。

第三步：对 DM6467 EVM 正确上电后，进入 uboot，设置以下变量：

```
TirDavinci_EVM# setenv serverip 192.168.0.60
```

```
TirDavinci_EVM# setenv ipaddr 192.168.0.32
```

```
TirDavinci_EVM# setenv ethaddr ff:ff:ff:ff:ff:ff
```

```
TirDavinci_EVM# tftp 0x80000200 ulmage
```

```
TirDavinci_EVM# bootm 0x80000200
```

系统将从 192.168.0.60 机器上/tftpboot 目录下下载 ulmage 到 DM6467 EVM 的 0x80000200 地址上，然后从该地址启动 linux kernel，文件系统从默认位置启动，这可以用 printenv 命令看出，使 nfs 还是 hda。

## 5 UBOOT 的使用

UBOOT 的命令可以在系统命令提示符下，通过 help 命令来获得。下面是常用的命名的一些简单说明。

### 1) bootm [addr [arg ...]]

bootm 命令可以引导启动存储在内存中的程序映像。这些内存包括 RAM 和可以永久保存的 Flash。

第 1 个参数 addr 是程序映像的地址，这个程序映像必须转换成 U-Boot 的格式。

第 2 个参数对于引导 Linux 内核有用，通常作为 UBoot 格式的映像存储地址，也可以是传递给 Linux 内核的参数。

### 2) bootp

```
bootp [loadAddress] [bootfilename]
```

bootp 命令通过 bootp 请求，要求 DHCP 服务器分配 IP 地址，然后通过 TFTP 协议下载指定的文件到内存。

---

第 1 个参数是下载文件存放的内存地址。

第 2 个参数是要下载的文件名称，这个文件应该在开发主机上准备好。

### 3) erase

擦除 Flash 命令，参数必须指定 Flash 擦除的范围。有以下两种方式：

按照起始地址和结束地址，**start** 必须是擦除块的起始地址；**end** 必须是擦除末尾块的结束地址，这是最常用的一种方式。

按照组和扇区进行擦除。

### 4) flinfo

**flinfo** 命令打印全部 Flash 组的信息，也可以只打印其中某个组。一般嵌入式系统的 Flash 只有一个组。

### 5) go

**go addr [arg ...]**

**go** 命令可以执行应用程序。

第 1 个参数是要执行程序入口的地址。

第 2 个可选参数是传递给程序的参数，可以不用。

### 6) minfo

**minfo** 可以打印程序映像的开头信息，包含了映像内容的校验（序列号、头和校验和）等。

### 7) nfs

**nfs [loadAddress] [host ip addr:bootfilename]**

**nfs** 命令可以使用 NFS 网络协议通过网络启动映像。

### 8) printenv

**printenv** 命令打印环境变量。

### 9) setenv

**setenv name value ...**

**setenv** 命令可以设置环境变量。

第 1 个参数是环境变量的名称。

第 2 个参数是要设置的值，如果没有第 2 个参数，表示删除这个环境变量。

### 10) sleep

**sleep** 命令可以延迟多少秒钟执行。

### 11) tftpboot

**tftpboot [loadAddress] [bootfilename]**

**tftpboot** 命令可以使用 TFTP 协议通过网络下载文件。按照二进制文件格式下载。

另外使用这个命令，必须配置好相关的环境变量。例如 **serverip**、**ipaddr** 和 **ethaddr**。

---

第 1 个参数 `loadAddress` 是下载到的内存地址。

第 2 个参数是要下载的文件名称，必须放在 TFTP 服务器相应的目录下。

## 6 启动方式说明

UBOOT 可以设置多种启动方式，本文仅以三种方式加以说明，用户可以根据自己的具体使用情况进行调整。

本节假定：PC 机的 IP 为 192.168.0.60，DM6467 EVM 板的地址为 192.168.0.32。

1) tftp 启动 kernel，文件系统使用硬盘方式

```
TirDavinci_EVM# setenv serverip 192.168.0.60
```

```
TirDavinci_EVM# setenv ipaddr 192.168.0.32
```

```
TirDavinci_EVM# setenv ethaddr ff:ff:ff:ff:ff:ff
```

```
TirDavinci_EVM# tftp 0x80000200 ulmage
```

```
TirDavinci_EVM# bootm 0x80000200
```

系统将从 192.168.0.60 机器上/tftpboot 目录下下载 ulmage 到 DM6467 EVM 的 0x80000200 地址上，然后从该地址启动 linux kernel，文件系统从默认位置启动，这可以用 `printenv` 命令看出，使 `nfs` 还是 `hda`。

2) kernel 和文件系统均由硬盘方式启动

```
TirDavinci_EVM# setenv ipaddr 192.168.0.32
```

```
TirDavinci_EVM#setenv bootargs mem=120M console=ttyS0,115200n8  
root=/dev/hda1 noinitrd rw ip=192.168.0.31:192.168.0.60:255.255.255.0  
davincihd_capture.channel0_numbuffers=4
```

```
TirDavinci_EVM# boot
```

系统启动后，将使用硬盘上的 `kernel` 和文件系统。

3) kernel 使用硬盘方式，使用网络文件系统

```
TirDavinci_EVM# setenv ipaddr 192.168.0.32
```

```
TirDavinci_EVM# setenv nfsroot 192.168.0.60:/root/armfs
```

```
TirDavinci_EVM# setenv bootargs console=ttyS0,115200n8 noinitrd rw  
ip=192.168.0.32:192.168.0.60 root=/dev/nfs nfsroot=$(nfsroot),nolock  
mem=120M davincihd_capture.channel0_numbuffers=4
```



---

TirDavinci\_EVM# boot

系统启动后，将使用 192.168.0.60 目录/root/armfs 下的文件系统。

## 7 硬盘的恢复

第一步：建立 nfs 环境，参见第 3 节。

第二步：在 nfs 主机上/root/armfs 下，进行数据准备工作。

建立/root/armfs/mnt/ restore目录，并且下列文件拷贝到该目录下：

- data.tar.gz
- dsa-install.sh
- mvl4.0.1-target.tar.gz
- prep-hdd
- dsa-install-cgi.tar.gz
- dsa-install.tar.gz
- overlay.tar.gz
- restore-hdd

第三步：网络文件启动后，在超级终端中执行以下命令：

```
host $ mount -t ext3 /dev/hda2 /mnt/restore
```

```
host $ date MMDDHHMMCCYY
```

```
host $ cd /mnt/restore
```

```
host $ chmod +x restore-hdd
```

```
host $ ./restore-hdd
```

系统会提示"This will destroy all data on /dev/hda1 - are you sure?"填入yes后按“Enter”键开始恢复操作

## 8 uboot 和 kernel 的编译

正确建立系统平台后，在/root下建立kernel目录，将uboot.tar.gz和kernel.tar.gz解压在该目录。

进入uboot目录，键入make后执行即可以编译成功。

进入ti-davinci目录，如果不修改config文件，则可以直接sh build.sh即可以编译，编译好的文件在arch/arm/boot下，uImage文件。如果修改了config文件，则执行host\$ **make ARCH=arm CROSS\_COMPILE=arm\_v5t\_le-checksetconfig**。

---

## 9 在 nand flash 上使用 uboot 烧写 kernel

使用ccs把uboot烧写进nand flash 后，可以使用uboot把kernel烧写进nand flash 中。

把步骤8中编译好的kernel拷贝到/tftpboot目录下。打开开发板电源，输入如下命令：

```
TirDavinci_EVM# setenv serverip 192.168.0.60
```

```
TirDavinci_EVM# setenv ipaddr 192.168.0.32
```

```
TirDavinci_EVM# setenv ethaddr ff:ff:ff:ff:ff:ff
```

```
TirDavinci_EVM# tftp 0x80700000 ulmage
```

系统将从 192.168.0.60 机器上/tftpboot 目录下下载 ulmage 到 DM6467 EVM 内存中 0x80000200 地址上，

```
TirDavinci_EVM# nand erase 0xa0000 0x155800
```

```
TirDavinci_EVM# nand write 0x80700000 0xa0000 0x155800
```

把 ulmage 烧写到 nand flash 的 0xa0000 地址上，ulmage 的大小是根据用户裁剪大小而变化的，所以烧写时请注意变化。

```
TirDavinci_EVM# boot
```

启动 kernel,文件系统使用硬盘中的文件系统。