# Rom Utinity使用方法：

1. 将gcc-arm-none-eabi-4\_7拷贝到linux 下user\local下，makefile指定这个路径下面的GCC编译器

2. 进入到rom\_utinity文件夹下

make TARGET=mx6x //编译代码

make TARGET=mx6x usdhc=1 //支持EMMC

make clean //清除编译生成的文件

3. windows C盘根目录下面制作跟linux路径相同的文件夹名称

例如：linux下的路径为：mnt/hgfs/rom\_utility

那么将编译完成的rom\_utinity文件夹拷贝到C：[\\mnt\\hgfs](file:///\\\\mnt\\\\hgfs)

4. 将attach\_6sll\_rom\_utility.cmm 放置到C：[\\mnt\\hgfs\\rom\_utinity](file:///\\\\mnt\\\\hgfs\\\\rom_utinity)

修改参照如下：

data.load.elf C:\mnt\hgfs\rom\_utility\output\mx6x\bin\rom\_utility\_mx6x.elf /path "C:\mnt\hgfs\rom\_utility\output\mx6x\bin"

5. 使用光盘安装t32marm, 选择第二项安装

6. 调试器USB连接到电脑，打开调试器，打开i.max板。双击运行T32\\bin\\windows64\\t32marm.exe

7.点击file\runScript，选择4中修改好的CMM文件，运行

8. 弹出TERM.VIEW窗口，按F7运行。

# Boot ROM 编译下载

1. 建立ssh链接

Ip：10.192.241.185

用户名：maxx

密码：123456

可以使用git自带的ssh登录到这个账户

ssh [maxx@10.192.241.185](mailto:maxx@10.192.241.185)

2. cd /mcurom\_src/code

执行命令：make PRODUCT=imx6sll

3. 打开git bash，拷贝数据到本地

a. 使用sftp建立链接：sftp [maxx@10.192.241.185](mailto:maxx@10.192.241.185), 选择yes，输入密码123456

b. cd mcurom\_src

c. 将code文件夹拷贝到本地：

get –r code /f/boot\_rom

研发uboot的git：

http://sw-stash.freescale.net/projects/IMX

# 使用ROM utinity下载uboot

1. 根据[使用方法：](#_使用方法：)运行shell

2. 单击，停止shell

3. 点击file\runScript，加载ddr3\_script.cmm

4. 点击file\runScript，加载imxrom.cmm

imxrom.cmm里面修改正确的uboot.imx文件路径跟加载地址，加载完后uboot.imax就被加载到ddr中了。

5. 单击，运行shell

6. 输入命令 spinor init port 0 cs 0,进行初始化

7. 输入命令spinor es addr 0x400 len 0x100000,擦除nor flash offset 0x400开始的0x100000个字节。0x400是bootrom规定的nor 中app存放地址

8. 输入命令spinor write src 0x80000000 dst 0x400 len 0x100000, 写入0x80000000处存放的uboot

9. 重启板子，开串口终端，配置波特率为115200，数据位8，奇偶校验none，停止位1。流量控制none。即可看到ubbot启动了

注意，com口在背面，连接上后会显示两个com口，其中一个可用

# 源码分析

## 1. makefile 分析

## 2. code 分析

### Shell command 如何执行的？

#### 执行的流程大致如下：

1. 程序编译时Shell command在shell command 段中注册

2. 运行程序时，进入shell\_task\_loop()

3. 输入command，程序进行命令的识别处理，从shell command段中查找是否存在command。

若存在，执行相关command

否则，显示invalid command

#### 命令的执行语句位于：

shell\_task\_loop()

cmd->func(argc, argvs);

cmd是一个结构体变量，结构体中包含三部分。name, func, usage

name: shell command对应的指令

func: shell command指令的具体实现

usage: shell command 的使用说明

typedef struct {

char \*name;

int (\*func) (unsigned int argc, char \*argvs[]);

char \*short\_usage;

} shell\_cmd\_t, \*shell\_cmd\_p;

#### 注册shell command的宏：

#define SHELL\_CMD(name, func, short\_usage) \_\_attribute\_\_ ((section(".shell\_cmd"))) \

shell\_cmd\_t shell\_cmd\_##name = { \

#name, \

func, \

short\_usage \

};

#### 注册语句：

SHELL\_CMD(test, do\_test, "test\t .................")

将注册语句宏展开，变为：

\_\_attribute\_\_ ((section(".shell\_cmd"))) shell\_cmd\_t shell\_cmd\_test= {

‘test’,

do\_test,

"test\t ................."

}

shell\_cmd\_test结构体将会存储在.shell\_cmd段中，.shell\_cmd段的定义是在ld文件中实现的.

查阅makefile，可以发现使用的ld文件是src\psp\mx6x\mx6x.ld

.shell\_cmd :

{

. = ALIGN (4);

\_\_shell\_cmds\_beginning = .;

KEEP(\*(.shell\_cmd))

\_\_shell\_cmds\_end = .;

} >IRAM

# Imx6sll BootRom源码分析：

Pu\_irom\_vector.s中：

ldr pc, start\_addr

start\_addr

dcd pu\_reset\_handler

pu\_reset\_handler存在于Pu\_reset\_handler.s中

pu\_reset\_handler PROC

;Init all registers to zero (r0 used below)

;------------------------------------------

mov r1, #0x0

mov r2, r1

mov r3, r1

mov r4, r1

mov r5, r1

mov r6, r1

mov r7, r1

mov r8, r1

mov r9, r1

mov r10, r1

mov r11, r1

mov r12, r1

; Check Power Gating Status register

LDR R1, =power\_gating\_status\_reg

LDR R1, [R1]

; If not 0 branch to entry restore function

CMP R1, #0

BNE entry\_restore

;Set up the SCV stack. (Exception stacks not used by the rom.)

;------------------------------------------------------------

msr CPSR\_c, #SVC\_mode:OR:I\_bit:OR:F\_bit ; Enter SVC\_mode, no interrupts

ldr sp, =pu\_irom\_svc\_stack+STACK\_SIZE-4 ; Initialize stack pointer r13

LDR R0, =pu\_irom\_init

BX R0

首先清理R0-R12寄存器，

然后读取power\_gating\_status\_reg地址对应的值，做相应的判断

接着配置CPSR，使cpu进入SVC模式（管理模式）

配置栈后跳转到pu\_irom\_init

## Linkfile

F:\boot\_rom\code\build\link\imx6sll\_ads\_rom\_link.cmd

# Uboot

源代码下载地址：

<http://sw-stash.freescale.net/projects/IMX/repos/uboot-imx/browse>

可以编译的机器：

10.192.244.44

Fanny

Happy123

. /opt/poky/1.7+snapshot/environment-setup-cortexa7hf-vfp-neon-poky-linux-gnueabi

if you want to get uboot for spinor:

make mx6sll\_lpddr3\_arm2\_spinor\_defconfig && make

if you want to get uboot for SD card:

make mx6sll\_lpddr3\_arm2\_defconfig && make