一：顶层makefile

1. 确定编译模式，简单编译，静默编译，详细编译（根据命令行指令，看v的值）
2. 计算uboot环境变量，如cpu，arch，board(开发板名字),vendor(供应商),soc(芯片型号)等
3. 导出子Makefile所需的变量
4. 构建scripts\_basic依赖，%config目标
5. scripts\_basic依赖：编译出script/basic/fixdep 这个软件。

%config目标：使用xxx\_defconfig文件，将xxx\_defconfig文件的内容输出 到配置文件.config中，生成.config中。

二：uboot启动流程

A:引导bootloader加载器(flash,sd卡等)

B：开发板初始化：

1：\_start：

(1).编译后，入口为根目录下：u-boot.lds 。文件内容从\_start开始(vector里面 的start文件，后续存放中断向量表)

(2).\_start 中，最先转入reset函数，通过寄存器相关运算，使能开发板 的svc32模式(特权模式，可保护性修改开发板内配置)并取消FIQ 和IRQ中断

(3).调用cpu\_init\_cp15，cpu\_init\_crit ，初始化cpu的cp15 寄存器(系统控制协调处理器)，然后初始化cpu,在cup初始化中 调用了lowlevel\_init

a.lowlevel\_init:

1. 设置ocram首地址和大小(用于内存映射)
2. 设置iarm\_size
3. 调用s\_init(系统初始化)
4. .至此，完成reset，进入是 save\_boot\_params\_ret的\_main，  
    在\_main 函数里面调用了 board\_init\_f、relocate\_code、relocate\_vectors 和 board\_init\_r这4个函数

a.board\_init\_f:  
 ①、初始化一系列外设，比如串口、定时器，打印一些消息等。  
 ②、初始化 gd 的各个成员变量，uboot 会将自己重定位到 DRAM最后面 的地址区域，也就是将自己拷贝到 DRAM 最后面的内存区域中。这么 做的目的是给Linux 腾出空间，防止 Linuxkermel 覆盖掉 uboot， 将DRAM 前面的区域完整的空出来

b.relocate\_code:  
 将 uboot 拷贝到新的地方去,代码重定位

c.relocate\_vectors:  
 中断向量表重定位

d.board\_init\_r:(在这里面末尾调用了run\_main\_loop,用于开机进入boot命令)  
 board\_init\_f 并没有初始化所有的外设，还需要做一些后续工作，这

些后续工作就是由函数 board\_init\_r 来完成的,比如usb，lcd和一些杂七杂八的输入输出设备

(在run\_main\_loop中还涉及到cli\_loop,cmd\_process。cli是uboot 的命令行处 理函数，cmd来执行具体的命令)

1. uboot移植:

A.编译:

1.创建个 shell 脚本就行了，shell 脚本名为 mx6ull\_14x14\_emmc.sh，然后在 shell 脚本里面输入如下

1 #**!/bin/bash**

2 make ARCH**=**arm CROSS\_COMPILE**=**arm**-**linux**-**gnueabihf**-** distclean//每次都清除文件

3 make ARCH**=**arm CROSS\_COMPILE**=**arm**-**linux**-**gnueabihf

mx6ull\_14x14\_evk\_emmc\_defconfig //配置deconfig

4 make V**=**1 ARCH**=**arm CROSS\_COMPILE**=**arm**-**linux**-**gnueabihf**- -**j16//详细编译，16核(要chmod 777 更改.sh执行权限)

B.移植:

1.添加开发板默认配置文件:

先在 configs 目录下创建默认配置文件，复制 mx6ull\_14x14\_evk\_emmc\_defconfig，然后重命名为 mx6ull\_alientek\_emmc\_defconfig，将41x14的内容里面的名字换成重命名之后的

2.添加开发板对应头文件:

在目录 include/configs 下添加I.MX6ULL-ALPHA 开 发 板 对 应 的 头 文 件 ，复制include/configs/mx6ullevk.h,并重命名为 mx6ull\_alientek\_emmc.h，命令如下:

cp include/configs/mx6ullevk.h include/configs/mx6ull\_alientek\_emmc.h(后续更改内容查看驱动手册P867,从内容覆盖多种宏定义)

3.添加开发板对应的板级文件夹:

NXP 的 I.MX 系列芯片的所有板级文件夹都存放在 board/freescale 目录下，在这 个目录下 有个名为 mx6ullevk 的文件夹，这个文件夹就是 NXP 官方 I.MX6ULL EVK 开发板的板级文件 夹。复制 mx6ullevk，将其重命mx6ull\_alientek\_emmc

，然后对\_emmc目录文件中的内容进行修改  
 修改内容:

a.makefile文件(驱动手册P876)  
b.imximage文件

c.kconfig文件  
d.maintainers文件  
e.uboot图形界面配置文件(修改文件路径，开发板各不相同arch/arm/..)

4.使用新添加的板子配置编译uboot(步骤参考A.编译):

若成功编译,则完成第一步移植，后续则要根据实际情况更改去驱动，如LCD,网卡等(驱动手册P879-P902)，完成后续更改则完成移植，接下来可以测试移植后的uboot能不能启动linux(驱动手册P909)

注:文件 mx6ull\_alientek\_emmc.h 中的环境变量bootcmd,bootargs。分别是用来启动linux内核和保存uboot传递给linux内核的参数(consle,root,等)

四:uboot图形化配置:

1.操作:

通过键盘上的“↑”和“↓”键来选择要配置的菜单，按下“Enter”键进入子菜单。菜单 中高亮的字母就是此菜单的热键，在键盘上按下此高亮字母对应的键可以快速选中对应的菜单。 选中子菜单以后按下“Y”键就会将相应的代码编译进 Uboot 中，菜单前面变为“< \* >”。按下 “N”键不编译相应的代码，按下“M”键就会将相应的代码编译为模块，菜单前面变为“< M >”。 按两下“Esc”键退出，也就是返回到上一级，按下“?”键查看此菜单的帮助信息，按下“/”键打开搜索框，可以在搜索框输入要搜索的内容。

在配置界面下方会有五个按钮，这五个按钮的功能如下：

<Select>：选中按钮，和“Enter”键的功能相同，负责选中并进入某个菜单。

<Exit>：退出按钮，和按两下“Esc”键功能相同，退出当前菜单，返回到上一级。

<Help>：帮助按钮，查看选中菜单的帮助信息。

<Save>：保存按钮，保存修改后的配置文件。

<Load>：加载按钮，加载指定的配置文件。

注意:  
使用如下命令编译 uboot：

make ARCH=arm CROSS\_COMPILE=arm-linux-gnueabihf- -j16

千万不能使用如下命令：

./mx6ull\_alientek\_emmc.sh

因为 mx6ull\_alientek\_emmc.sh 在编译之前会清理工程，会删除掉.config 文件

2.添加自定义菜单(驱动手册P927)

: