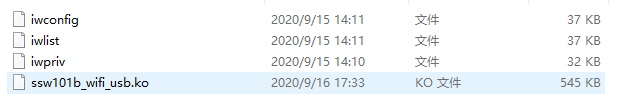
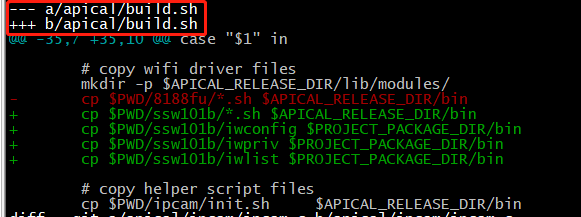
# wifi驱动 8188fu 和 101b

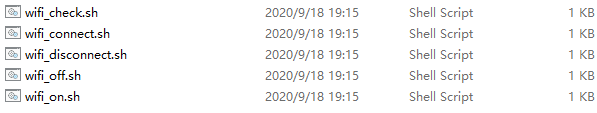




第一步：



第二步：在ssw101b目录新建五个shell脚本







其中这五个脚本中 包含着WiFi驱动的加载 以及 WiFi的连接 IP地址的获取、详细的内容请看脚本语言。

第三步：关于WiFi连接工具

wpa\_supplicant -D nl80211 -i wlan0 -c /etc/wpa\_supplicant.conf -B

-D nl80211：表示驱动类型名称为nl80211

-i wlan0：表示网络接口名称为wlan0

-c /etc/wpa\_supplicant.conf：表示配置文件为wpa\_supplicant.conf

-B：表示将守护进程wpa\_supplicant 挂到后台运行

其他的一些相关参数：

-i<ifname> : 网络接口名称

-c<conf>: 配置文件名称  
-C<ctrl\_intf>: 控制接口名称  
-D<driver>: 驱动类型名称  
-p<driver\_param>: 驱动参数  
-b<br\_ifname>: 桥接口名称

-d: 增加调试信息

第四步 说一下具体流程

操作和实现步骤的流程

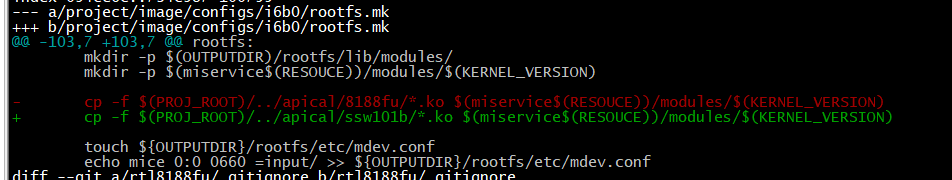
1、上电

2、安装 WiFi 驱动

3、用 wpa\_supplicant 连接 WiFi

4、 连接后 用 udhcpc 工具 自动的获取 IP地址

第五步：b/project/image/configs/i6b0/rootfs.mk 目录下的更改



关于udhcpc 的相关命令

-A N 失败后等待N秒（默认为20秒）

-i etho 要使用的网络接口

-r IP 请求IP地址

-T N 尝试获取N秒租约（默认为三秒）

-P N 使用端口号为N 而不是默认68

其他的下次遇到再百度吧。

# make 和 makefile

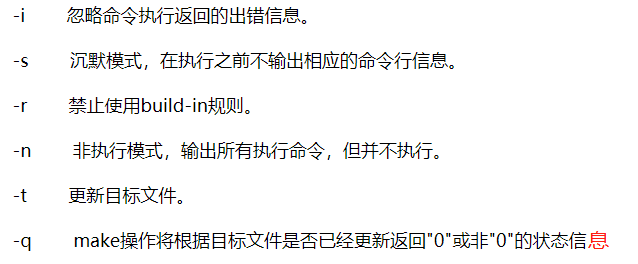
make 常见的标志参数：

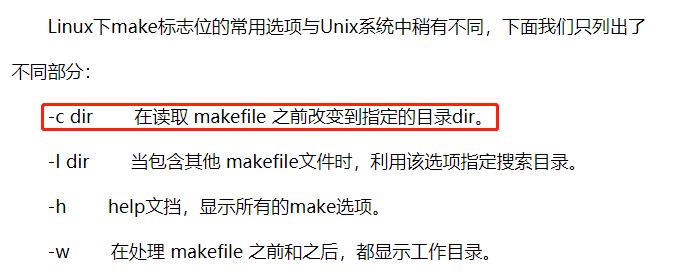
1、-f file　 指定file文件为描述文件，如果file参数为"-"符，那么描述文

件指向标准输入。如果没有"-f"参数，则系统将默认当前目录下名为makefile或者名为

Makefile的文件为描述文件。在Linux中， GNU make 工具在当前工作目录中按照

GNUmakefile、makefile、Makefile的顺序搜索 makefile文件。





之前个人学习总结的makefile 笔记：

http://note.youdao.com/noteshare?id=40f7141c682bdc0518d9b4e96697935e&sub=97F33B0BFB0C487AB02754A84696416F

# ipcam 分区的操作

用户自定义private分区：

1、找到对应的分区配置文件，如335在Z:\ext\msc333\project\image\configs\i6b0\nor.squashfs.partition.config

2、修改分区配置文件nor.squashfs.partition.config

修改IMAGE\_LIST ，添加新分区名字private， IMAGE\_LIST = mxp ipl ipl\_cust uboot kernel rootfs miservice customer private

修改USR\_MOUNT\_BLOCKS，添加用户需要挂载的block，添加private， USR\_MOUNT\_BLOCKS:=miservice customer private

修改mxp$(SYSTAB) ，添加$(private$(PATSIZE))(private)， mxp$(SYSTAB) = "$(kernel$(PATSIZE))(KERNEL),$(rootfs$(PATSIZE))(rootfs),$(miservice$(PATSIZE))(miservice),$(customer$(PATSIZE))(customer),$(private$(PATSIZE))(private)"

添加private分区属性：

private$(RESOUCE) = $(OUTPUTDIR)/private

private$(FSTYPE) = jffs2

private$(PATSIZE) = 0x100000

private$(MOUNTTG) = /private

private$(MOUNTPT) = /dev/mtdblock5

private$(OPTIONS) = rw

（可通过cat/proc/mtd指令来查看分区对应的block）

至此nor.squashfs.partition.config文件修改完毕

3、修改project/image/configs/i6b0/rootfs.mk，要新建private文件夹，为挂载分区做准备

mkdir -p $(OUTPUTDIR)/private

mkdir -p $(OUTPUTDIR)/rootfs/private

4、修改project/board/i6b0/boot/nor/partition/tool/mxp\_gen.c分区表，要同各个分区的大小对应好，不然会出错

5、project/onebin16M.sh，打包脚本也要把各个分区的大小对应好

分区表生成方法：

macaron\project\board\i6\boot\nor\partition目录下执行：

1、gcc -o mxp\_gen tool/mxp\_gen.c

2、./mxp\_gen MXP\_SF.bin

即在当前路径产生新的MXP\_SF.bin文件，同样适用于MXP\_SF\_8M.bin

# Busybox工具的使用

在linux中嵌入式平台中，经常需要使用很多工具， busybox是一个类似安卓市场或者软件管家这样的工具，集成了非常多的使用工具

在使用的时候 ，只需要 ln 一下生成链接，就可以使用对应点工具

llinux  busybox的移植，就是把交叉编译器修改为自己的文件系统交叉编译器    然后编译就可以

**busybox编译和移植**

     busybox 的编译与Linux内核的编译过程类似。从<http://www.busybox.net/downloads/> 下载最新的源码，解压后，通过以下几步，即可完成busybox的编译和移植：

**（1）make xxxxxxconfig**

         busybox提供了几种配置：defconfig (缺省配置)、allyesconfig（最大配置）、 allnoconfig（最小配置），一般选择缺省配置即可。

          这一步结束后，将生成.config

**(2)make menuconfig**

         这一步是可选的，当你认为上述配置中还有不尽如意的地方，可以通过这一步进行微调，加入或去除某些命令。

         这一步实际上是修改.config

**（3）make CROSS\_COMPILE=arm-linux-**

          这一步就是根据.config，生成busybox，当然你也可以指定其他的编译器， 如arm-linux-gnueabi-。（"make CROSS\_COMPILE="将用gcc编译PC机上运行的busybox.

 将编译生成的busybox  拷贝到linux arm平台 即可使用

**busybox的使用**

    busybox的使用很简单，有以下三种方式：

**（1） busybox后直接跟命令，如**

          busybox ls

          busybox tftp

     (**2)  直接将busybox重命名，如**

          cp busybox tftp

          cp busybox tar

          然后再执行tftp, tar

**（3）创建符号链接（symbolic link）， 如**

          ln -s busybox rm

          ln -s busybox mount

          然后就可以执行rm，mount等

参考文档：https://www.cnblogs.com/ChenChangXiong/p/11112619.html

# 修改bootargs和bootcmd

u-boot的环境参数中有两个bai和内核启动相du关的，它们是bootcmd和bootargs

.1、bootargs

这个参数设置要传递给内核的信息，主要用来告诉内核分区信息和根文件系统所在的分区。

示例：

root=/dev/mtdblock5 rootfstype=jffs2 console=ttyS0,115200 mem=35M mtdparts=nand.0:3840k(u-boot),4096k(kernel),123136k(filesystem)

其中：

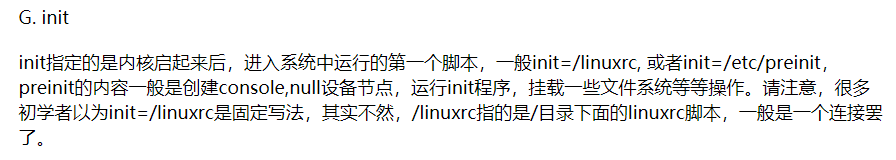
root=/dev/mtdblock5 表示根文件系统在第五分区

rootfstype=jffs2 表示根文件系统的类型是jffs2

console=ttyS0,115200 表示终端为ttyS0,串口波特率为115200

mem=35M 表示内存大小为35M

mtdparts告诉内核MTD分区情况，它在内核目录/drivers/mtd/cmdlinepart.c文件中有介绍



2、：

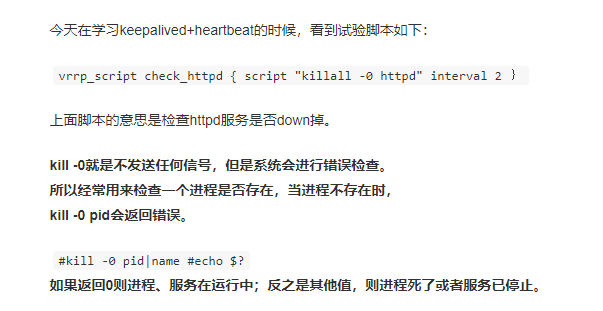
.bootcmd：

这个参数包含了一些命令，这些命令将在u-boot进入主循环后执行

示例：

bootcmd=boot\_logo;nand read 10000003c0000 300000;bootm 1000000

意思是启动u-boot后，执行boot\_logo显示logo信息，然后从nand flash中读内核映像到内存，然后启动内核。



# 机器死机的恢复方法

1. 使用 D:\tools\flash\_tool 工具
   1. 
2. 具体步骤参考 <file:///D:/tools/flash_tool/开发环境搭建参考.pdf> 16页开始
3. 按步骤操作完了，重启设备烧写one.bin
4. 烧写完如果不能进入系统，输入如下命令：

setenv bootcmd 'sf probe 0;sf read 0x21000000 50000 200000;bootm 0x21000000

setenv bootargs console=ttyS0,115200 root=/dev/mtdblock2 rootfstype=squashfs ro init=/linuxrc LX\_MEM=0x3fe0000 mma\_heap=mma\_heap\_name0,miu=0,sz=0x1C00000 mma\_memblock\_remove=1

saveenv

最后输入 reset 回车。

1. 正常的情况以上步骤基本可以了
2. 以下步骤是另一个验证方法：

1、"fatsize mmc 0 one.bin",

2、"fatload mmc 0 0x21000000 one.bin",

3、"crc32 0x21000000 $(filesize)",

4、"sf erase 0x0 0x7f0000",

5、 sf write 0x21000000 0x000000 $(filesize)",

6、mxp t.load

7、mxp r.info KERNEL

8、setenv sf\_kernel\_start $(sf\_part\_start)

9、setenv sf\_kernel\_size $(sf\_part\_size)

10、saveenv

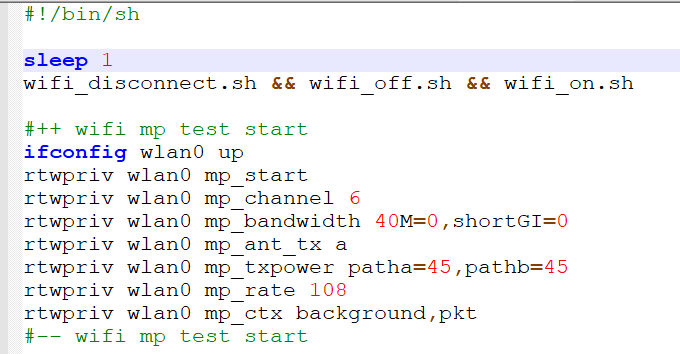
然后 reset

# 产测wifi定频测试功能

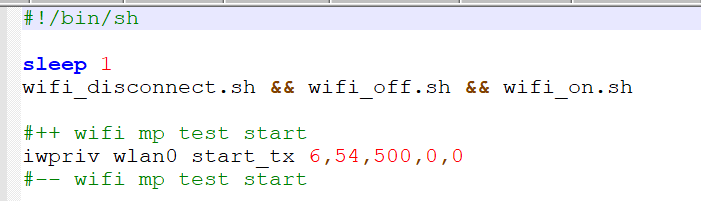
不同wifi驱动有着不同的执行的命令，目前主要用到8188fu 和 101b 驱动

核心的代码 在wifi\_test.sh 脚本。

1、8188的



2、101B



# OTA升级功能

主要代码在 tuya\_ipc\_system\_control.c 文件中

首选先运行 IPC\_APP\_Upgrade\_Inform\_cb 函数 ----》其次\_\_IPC\_APP\_get\_file\_data\_cb 函数-----》最后\_\_IPC\_APP\_upgrade\_notify\_cb函数。

最后一个函数运行一个脚本 update.sh 搞不懂为什么会死机？？？