**ASM9260T使用遇到的问题汇总**

**作者:向仔州**

文件系统存储文件问题…………………………………………………………………………………………………2

文件系统烧写自己做的文件，下载问题.........................................................................2

TFTP下载文件问题............................................................................................................3

Asm9260t Alsa操作问题....................................................................................................4

网口使用udhcpc无法获取DNS问题...............................................................................5

内核编译后生成kernel烧写文件遇到的问题.................................................................5

Wifi RTL8188eus编译，安装.............................................................................................6

ASM9260T USB烧写程序注意事项...................................................................................8

交叉编译器里面库文件和头文件路径问题.....................................................................8

ASM9260T-GPIO使用方法.................................................................................................9

GPIO作为中断的使用方法...............................................................................................11

ASM9260T使用RTL8188EUS，启动AP功能方法..........................................................14

ASM9260T驱动网卡芯片KSZ8041RNL网线接口检测问题…………………………………………18

VBUS电源内核报错arch\_interrupt 85 : VBUS error workaround……….19

ASM9260T挂载SD卡问题…………………………………………………………………………………………..20

ASM9260T不同linux2.6.32内核版本的SD卡内核启动无法产生/dev节点问题……..21

SD卡出现只能读无法写数据的问题…………………………………………………………………………..22

SocketCAN总线移植到linux2.6.32上的方法………………………………………………………………23

ASM9260T CAN引脚映射设置，比如我想其它规定的GPIO也有CAN功能……………24

电路板启动后文件系统自启动程序流程……………………………………………………………………..24

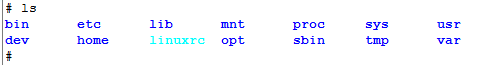
增大yaffs2文件系统存储容量，需要修改uboot………………………………………………………..26

擦除清空yaffs2目录下的文件……………………………………………………………………………………..27

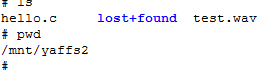
**文件系统存储文件问题**

CRAMFS文件系统是专门针对闪存设计的只读压缩的文件系统，其容量上限为256M,采用zlib压缩，文件系统类型可以是EXT2或EXT3

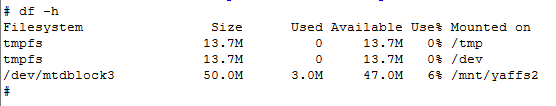
所以你在文件系统任意目录下存放的文件在开发板重启后都会被删除



但是asm9260t又在/mnt下自动挂载了yaffs2文件系统用来存放写入和读出的文件

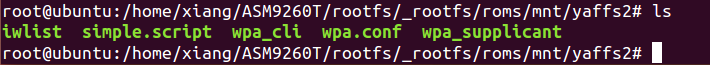
在/mnt/yaffs2下可以存储文件

检查磁盘空间大小



发现yaffs2只能存储50M的东西，但是后面可以修改因为Nandflash是128M的

**文件系统烧写自己做的文件，下载问题**



只有在开发板上用U盘把自己做的文件拷贝在roms/mnt/yaffs2目录上是可以的

如果你是想先把自己做的文件拷贝在roms文件系统的/mnt/yaffs2目录下，然后烧写编译进去，这样的话开发板yaffs2目录还是没有你拷贝的文件

yaffs2只支持你开发板u盘拷贝的文件，如果是在PC机上做文件系统烧录，那你必须把文件放在除了yaffs2以外的其他地方，这样你烧录进开发板文件才不会丢失。

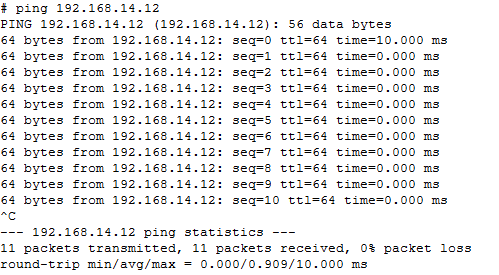


你的命令工具可以都放在除yaffs2以外的其他目录下执行。但是你的读写文本文件，conf后缀文件，必须放在yaffs2可读可写目录下。

**TFTP下载文件问题**

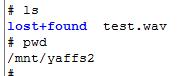


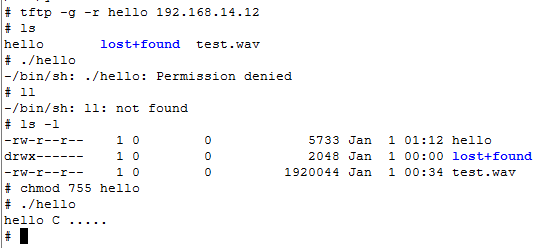
但是我ping主机是通的



而且主机自己tftp localhost测试也是通的。

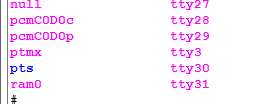
然后进入yaffs2目录





最后发现可以了，看来和文件系统cramfs问题一样，其他位置都是只读的，只要yaffs2可以自由读写

**Asm9260t Alsa操作问题**

ALSA的设备节点是在/dev目录下的，但是我们的aplay，amixer，arecord软件都是默认在/dev/snd目录下去找，所以我要重新映射设备节点的硬连接到snd目录下

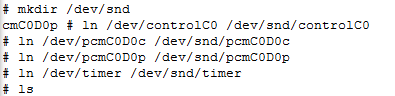
mkdir /dev/snd

ln /dev/controlC0 /dev/snd/controlC0

ln /dev/pcmC0D0c /dev/snd/pcmC0D0c

ln /dev/pcmC0D0p /dev/snd/pcmC0D0p

ln /dev/timer /dev/snd/timer



然后可以正常操作alsa了

芯片默认上电line和耳机是直通的

amixer cset numid=12 off

1.配置声卡线性输入

amixer cset numid=4 1

2.配置声卡playback输出

amixer cset numid=14 1

3.录音，该命令可录制10s长度48000波特率16位的wav音频文件

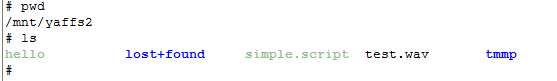
arecord -d 10 -fdat test.wav

4.播放音频文件

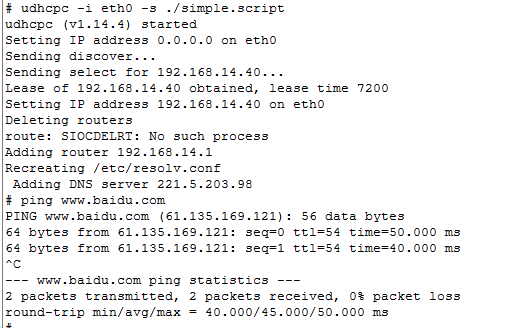
aplay test.wav

**网口使用udhcpc无法获取DNS问题**

要事先准备simple.script(为udhcpc所用)，可以放入/mnt/yaffs2，或者放到其它目录，因为紫芯的文件系统是cramfs只读文件系统，所以先放在yaffs2下做调试

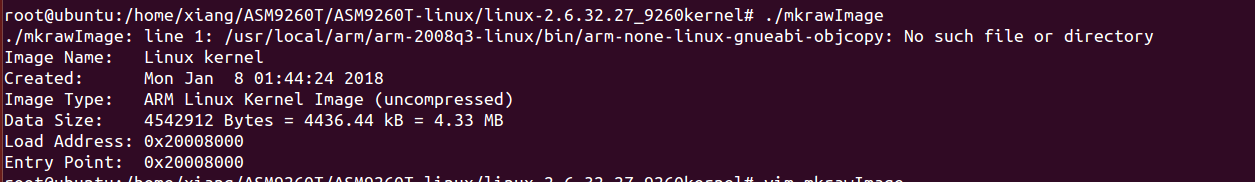


执行 udhcpc –i eth0 –s ./simple.script 就是可以了，然后ping百度没问题

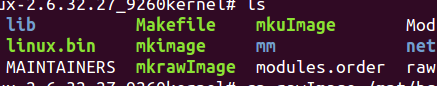


网口获取dhcp一定要点斜杠simple.script

**内核编译后生成kernel烧写文件遇到的问题**



找不到交叉编译路径的目录



这是因为官方的mkrawImage这个执行文件里面路径不是我们指定的交叉编译器路径，所以我们要修改mkrawImage里面的路径代码。打开文件看就知道怎么改了



我修改成了我自己交叉编译器的路径

**Wifi RTL8188eus编译，安装**

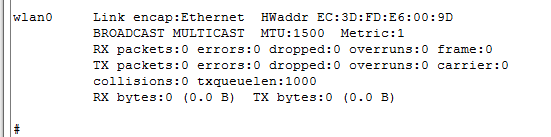
移植方法按照这个文档来。

要编译的代码在这个压缩包里面

按照文档要求编译

然后insmod就可以成功加载wifi了

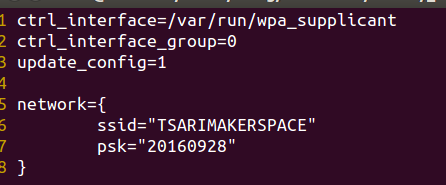




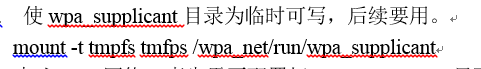
如果在insmod的时候发现

是因为8188驱动程序编译的时候linux内核路径没有指定对。

然后要修改你存放wifi用户名和密码文件的ctrl\_interface

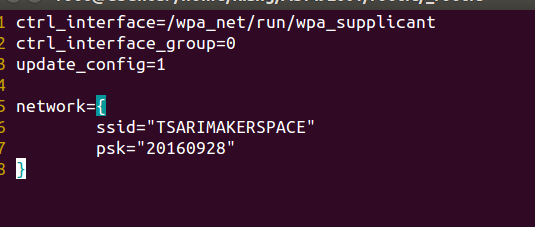


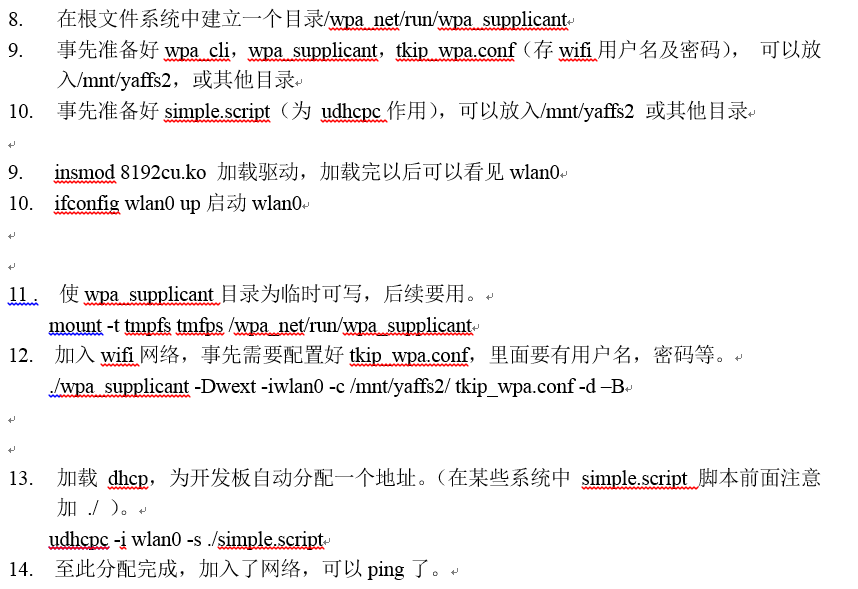
因为ASM9260T操作wifi RTL8188EUS文档下要求



要在根目录下创建的wpa\_net零时文件目录

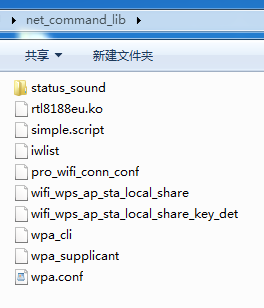
这样的话我们wpa\_supplicant操作wpa.conf文件里面ctrl\_interface接口路径也要改成一样

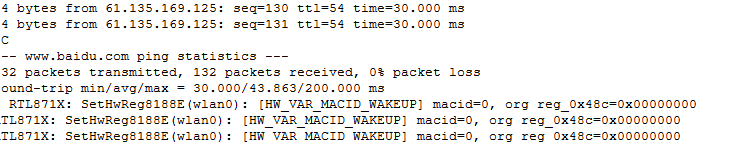




这个命令一定要执行

Asm9260t linux wifi工具在net\_command\_lib目录下

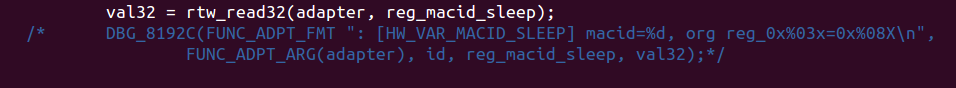


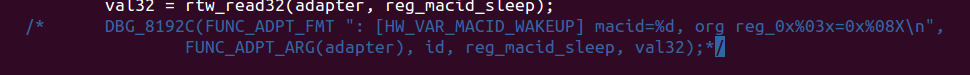


执行了wpa\_supplicant连接上网络后出现了WAKEUP周期唤醒功能，为了取消周期唤醒打印霸占调试屏幕的问题，需要屏蔽rtl8188代码里面的内容



去RTL8188EUS的目录下，找到rtl8188e\_hal\_init.c文件





这样就屏蔽了打印，但是周期唤醒功能还是存在，先用用看这个功能长时间用有没有问题

**ASM9260T USB烧写程序注意事项**



在目录找到这个压缩包解压，然后按照里面的文档安装。然后烧写程序安装官方文档操作

**交叉编译器里面库文件和头文件路径问题**

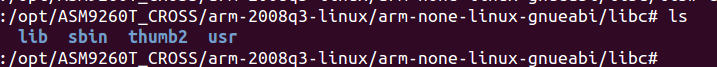


所有自己编译的include头文件和lib库文件都放在这个目录libc里面

常用的头文件库文件都放在这个里面



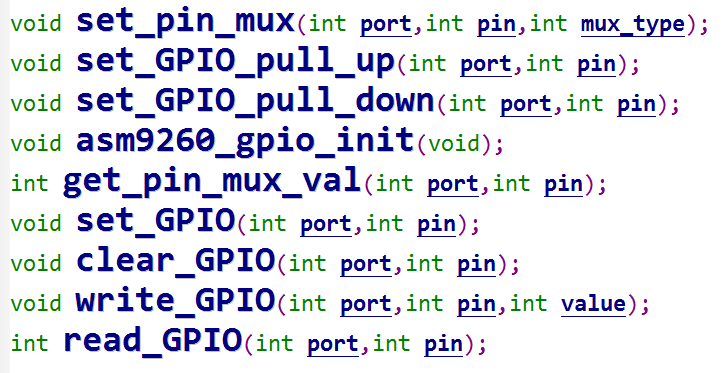
这是头文件的位置，在libc/usr/include目录下



库文件放在libc/目录下的lib里面

**ASM9260T-GPIO使用方法**

Pincontrol.h文件下有GPIO操作函数



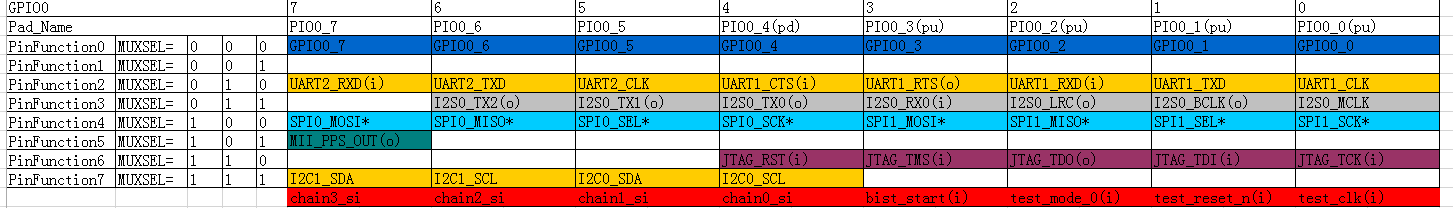
void set\_pin\_mux(int port,int pin,int mux\_type);

port:是指那组GPIO

pin:是指这组GPIO中的哪一个GPIO引脚

mux\_type：GPIO的管脚模式

GPIO管脚模式需要查表



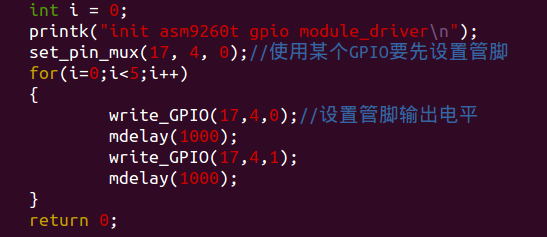
将里面的三位转换成16进制，填入set\_pin\_mux的mux\_type，就可以设置引脚复用功能

write\_GPIO(int port,int pin,int value); 该函数是写GPIO

port:是指那组GPIO

pin:是指这组GPIO中的哪一个GPIO引脚

value：写1高电平，写0底电平。



记住GPIO输出，要先设置GPIO管脚然后再输出电平

GPIO作为输入的方法

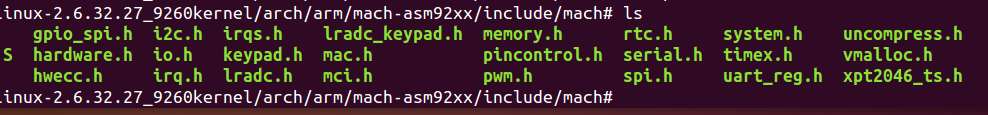
void set\_pin\_dir(int port, int pin, int bOut)

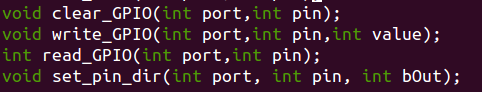
port:是指那组GPIO

pin:是指这组GPIO中的哪一个GPIO引脚

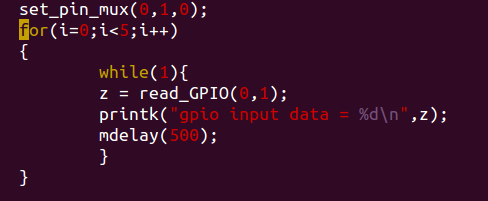
bOut 设置0为输入，设置1为输出，上下拉没法选择，I2C设备建议外部连接上来电阻

这里有个问题，set\_pin\_dir函数在C文件定义了，但是紫芯官方没在头文件定义，所以要自己将函数加进pincontrol.h





加上set\_pin\_dir函数,这个dir函数貌似用不到



这样就可以读单个GPIO了,每个GPIO低电平返回0，高电平返回1，这是固定的

我使用IMX6的时候每个GPIO低电平返回0，高电平返回GPIO编号，ASM9260T就不会这么装B，固定模式高电平返回0，低电平返回1

**GPIO作为中断的使用方法**

ASM9260T芯片GPIO作为中断引脚有几个地方要注意

比如我用做外部中断IO

第1步，申请GPIO为外部输入中断功能

void io\_irq\_enable\_edge(int port,int pin,int type) //GPIO中断功能申请函数

port:是指那组GPIO

pin:是指这组GPIO中的哪一个GPIO引脚

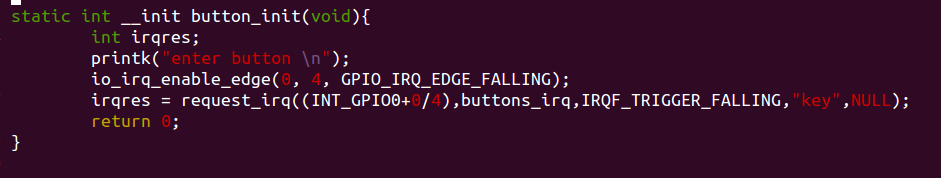
type：是指IO口触发那种中断

#define GPIO\_IRQ\_LEVEL\_LOW 0

#define GPIO\_IRQ\_LEVEL\_HIGH 1

#define GPIO\_IRQ\_EDGE\_FALLING 0

#define GPIO\_IRQ\_EDGE\_RISING 1



第2步，申请中断函数

request\_irq((INT\_GPIO0+portgroup/4),…………………)

关键是这个移位算法 (INT\_GPIO0+ portgroup /4)



因为ASM9260T中断源没有像IMX6，三星平台那种IO口静态映射表，所以要自己写数字

(INT\_GPIO0+ portgroup /4)

Portgroup：表示你外部中断的引脚属于GPIO哪一组

因为1个引脚是分到一个GPIO组里面，比如说GP0\_4，就是GPIO0组第4个引脚。GP1\_4就是第1组4号引脚，GPIO0\_IRQ就是包含GPIO0～GPIO3组的32个引脚共享中断号。

所以portgroup /4 不管你是用GPIO0,GPIO1,GPIO2,GPIO3组的任何一个引脚，都必须分配到GPIO0\_IRQ，也就是3号中断源，如果是GPIO4组就分配到GPIO1\_IRQ也就是4好中断源

request\_irq((INT\_GPIO0+portgroup/4),…………………)

(INT\_GPIO0+portgroup/4) 这个公式你只需要根据你使用的引脚编号对应上GPIO组，然后把GPIO组的编号写给portgroup变量就行了。其余的你不需要修改，/4除以4也不需要修改。

第3步：在中断函数里面执行判断IO口是否发生中断，和清除中断标志位

int get\_io\_irq\_status(int port,int pin) //gpio引脚是否发生中断再次确认函数

port:是指那组GPIO

pin:是指这组GPIO中的哪一个GPIO引脚

返回值为1表示，GPIO确实发生了外部中断，返回值为0表示没有发生外部中断

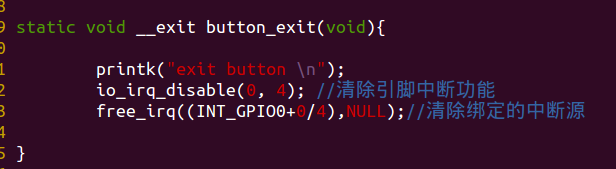
void io\_irq\_clr(int port,int pin) //中断函数执行完前一定要执行这句清除中断标志位，否则中断会不停的执行。

port:是指那组GPIO

pin:是指这组GPIO中的哪一个GPIO引脚



我在IMX6上就没有执行清除中断标志位，也没有什么问题，也许是官方SDK自动帮你清除了。这个还是根据实际使用平台来确定。



记得不使用要清除中断功能



这是ASM9260T整个中断申请，释放，使用流程

**ASM9260T使用RTL8188EUS，启动AP功能方法**

因为ASM9260T是用linux2.6.32版本的内核，而且ASM9260T的文件系统是Cramfs(只读文件系统)，所以有一些注意事项

因为供应商很多使用的RTL8188驱动是linux3.x以上的，所以我找到紫芯官方给的RTL8188EUS\_STA\_AP的驱动

解压压缩包

得到RTL8188EUS\_OLD文件目录



一定要看README,要结合紫芯的world文档看

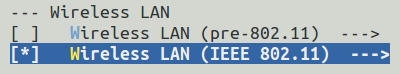
编译出来8188eu.ko的驱动

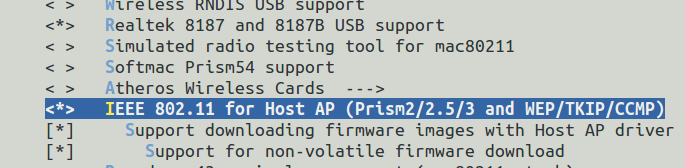
确认内核配置









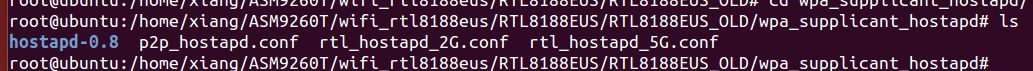
一定要把内核配置的AP模式选择上

编译hostapd

然后hostapd一定要使用紫芯官方给的源码包来编译，不要用网上的hostapd源码，因为紫芯官方和RTL8188厂商做了手脚。hostapd源码包也在RTL8188驱动包下

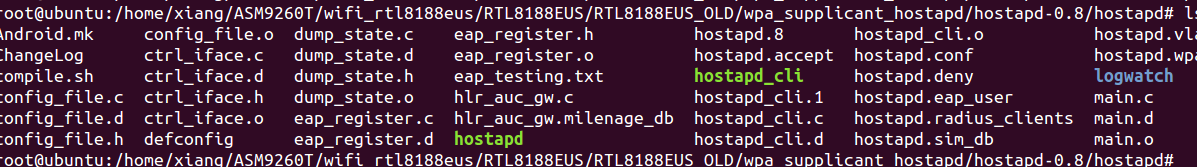


Wpa\_supplicant\_hostapd目录里面就是



这个是官方给的配置文件例子，直接复制粘贴到开发板先使用，然后再修改

这个就是hostapd软件源码包

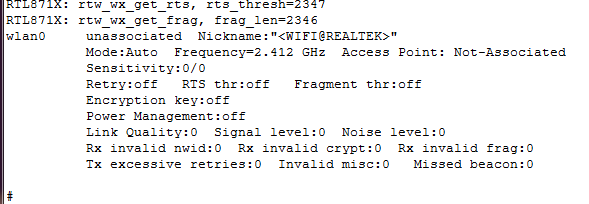


按照我写的linuxUSB\_wifi文档的方法吧hostapd软件编译出来

然后将rtl\_hostapd\_2G.conf和hostapd，hostapd\_cli拷贝到开发板上。

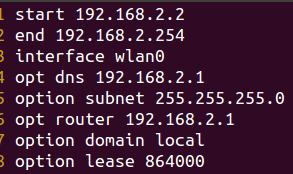
按照README文档来启动RTL8188的AP模式

加载rtl8188eus驱动



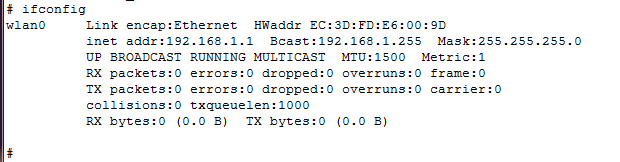
查看驱动是否添加成功

给wifi网卡配置IP地址，然后会自动启动

注意：这个给网卡分配的IP地址一定要根据udhcpd 的dhcpd.conf文件来配置

我的dhcpd.conf配置的地址范围是192.168.2.2～192.168.2.254，所以我们设置网卡IP地址为192.168.2.2本身或者以上的IP段都可以

这是启动完成



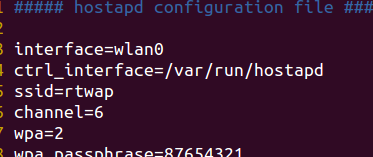
查看IP配置成功

执行hostapd服务，这样手机才能看到热点



hostapd启动失败，这是因为我们的ASM9260T平台的文件系统是只读系统

这点很关键



这个interface是使用hospapd\_cli查看有多少手机连接到我开发板网卡，使用的接口

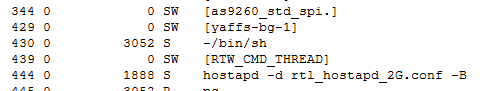
我们的rtl\_hostapd\_2G.conf文件里面ctrl\_interface指定的目录正好在Cramfs文件系统的只读路径下，不是在yaffs2可读写的目录。

mount -t tmpfs tmfps /var/run 我们用mount –t tmpfs命令将该路径下文件变成可写



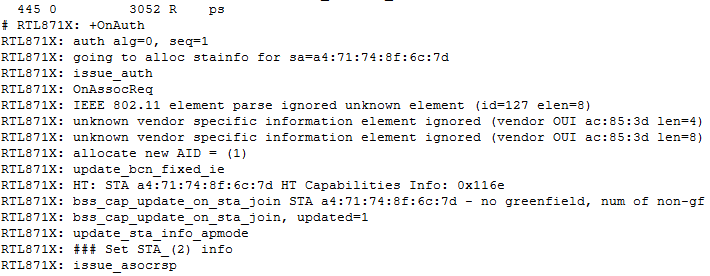
再次启动hostapd软件

hostapd启动成功



进程下能看到hostapd程序

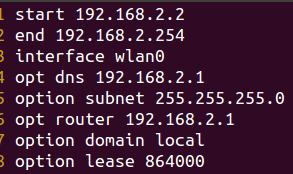
手机可以看到AP热点了，我用手机连接



出现了这个问题，最后手机显示连接失败。

看来我们要打开udhcpd来启动DNS自动分配IP功能

在文件系统/etc目录下创建udhcpd.conf

保存udhcpd.conf文件

然后将udhcpd.conf文件拷贝到开发板/etc目录下



在启动udhcpd的时候可能找不到leases文件，所以我们要在/var/lib/misc目录下创建leases文件。

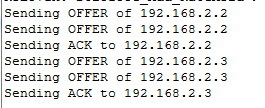
因为ASM9260T使用的是Cramfs只读文件系统，我们要将该路径下的目录设置成可写权限

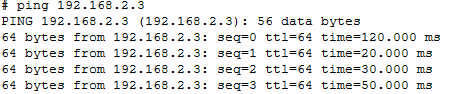
mount –t tmpfs tmfps /var/lib/misc



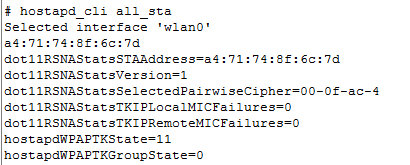
这样mount之后，本身在/var/lib/misc目录下的leases文件就不见了，所以我建议不要在文件系统烧写前将lease文件放在/var/lib/misc目录下，我们放在yaffs2目录下，然后在板子启动脚本执行的时候执行cp命令，将leases文件拷贝到/var/lib/misc目录下

启动udhcpd自动分配IP服务

地址分配成功，其实应该没什么问题了



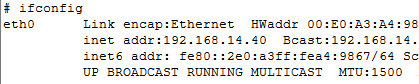
Ping下手机IP地址，数据包收发成功



为了保险起见我们用hostapd\_cli软件查看连接上我开发板的手机网卡地址，没问题

ASM9260T驱动网卡芯片KSZ8041RNL网线接口检测问题

我们用ifconfig或者cat来查看网线连接上主板网口没有



不管网线连没连接上主板，这里显示的都是RUNNING

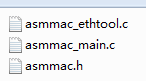


不管网线连没连接上主板，这里显示的都是1

这是因为ASM9260T网卡KSZ8041RNL驱动的问题

根据官方发过来的资料进行修改内核

在这个目录里面找到c和h文件

该文件用于linux-2.6.32版本

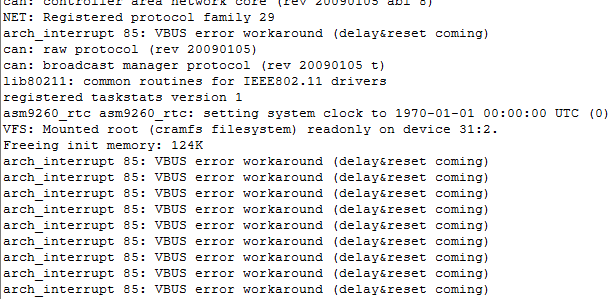


将这三个文件用来覆盖以前的内核版本，覆盖位置在内核目录下/drivers/net/asm92xx\_nac

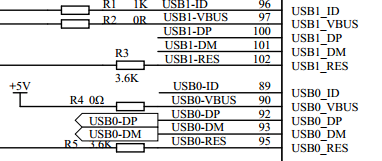
然后make

然后烧写内核进开发板就可以了

**内核报错 arch\_interrupt 85 : VBUS error workaround**



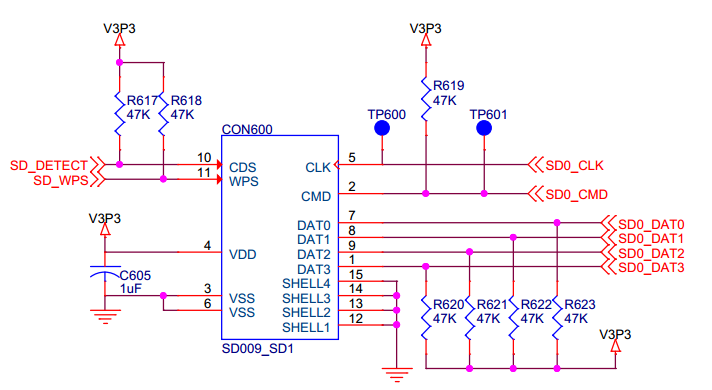
该报错的原因是硬件电路问题



该报错的原因是因为USB\_VBUS电源供电问题，电脑USB给主板供电很多都是4.7V，但是ASM9260T的USB\_VBUS供电必须在5V以上

所以为了保证USB\_VBUS的供电，我们可以在电源断开给个DC5V的电源适配器，也可以用3.3V升压到5V单独给USB\_VBUS供电

**ASM9260T挂载SD卡问题**



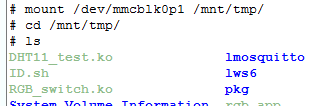
因为内核没有热插拔机制，所以SD卡是必须直接先插在主板上，然后启动内核才能识别到的，所以这两个引脚可以取消

只需要按照SDIO标准接SD卡就行了

如果内核启动成功的识别到SD卡会出现一下log



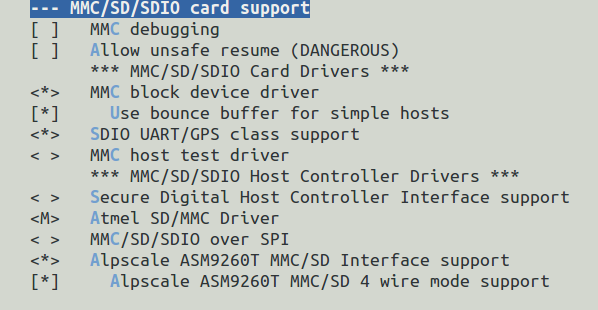
在/dev下面会产生mmc节点。记住是挂载mmcblk0p1

我挂载到tmp目录下，成功

**ASM9260T不同linux2.6.32内核版本的SD卡内核启动无法产生/dev节点问题**







保证以上内核配置完整

发现内核启动后没有自动生成SD卡节点

如果内核启动SD卡成功应该产生mmcblk0: p1设备节点





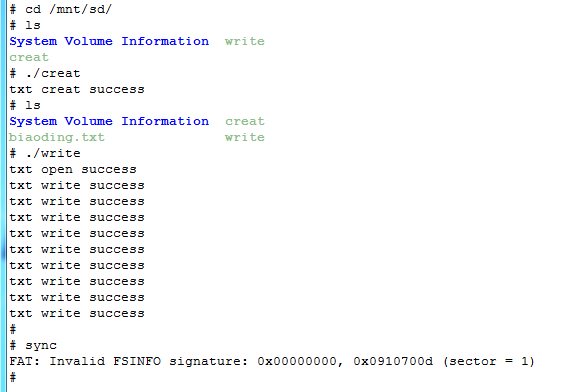
修改这个速率为25000000

重新编译内核，烧写进开发板，内核启动在/dev下面产生mmcblk0: p1节点，不用mount去挂载，内核自动挂载在/mnt/sd目录下，自己去/mnt/sd目录下看就是了

**SD卡出现只能读无法写数据的问题**

第1步: 查看SD卡槽的写保护引脚是不是接了什么电平，被写保护了。

第2步，如果SD卡槽硬件写保护引脚没有被使能，就看软件。

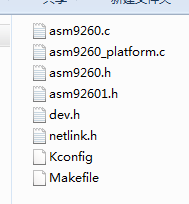


你看我写完数据之后执行sync，这样就把缓存的数据写入SD卡了。

还有就是有些内核sync后不会输出这个字符串，但是确实是存储进SD卡了的

**socketCAN总线移植到linux2.6.32上的方法**

这是官方给我的linux2.6.32上的can驱动

这就目录里面的文件

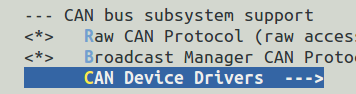
将这些文件完全覆盖到内核kernel/drivers/net/can/asm9260/ 目录下

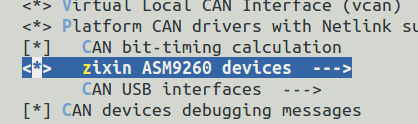


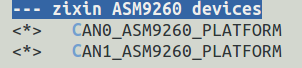
覆盖完成后去配置make menuconfig

选择网络支持

选择CAN总线子系统支持

选择CAN设备驱动支持

选择紫芯



虽然用的是CAN0但是一定要将CAN1也同时选择上，否则无法启动CAN0

cd /usr/sbin 进入usr的bin目录

执行./ip link set can0 up type can bitrate 500000



配置成功，然后你ifconfig就看得到CAN0设备节点了

但是移植后发现can还是用不了，官方又给了个新的内核，可以用CAN1的我把内核放在这个目录下了

**ASM9260T CAN引脚映射设置，比如我想其它规定的GPIO也有CAN功能**



打开C文件



修改这里

**电路板启动后文件系统自启动程序流程**

Cramfs文件系统分两个目录下的etc，

1.根目录下的/etc

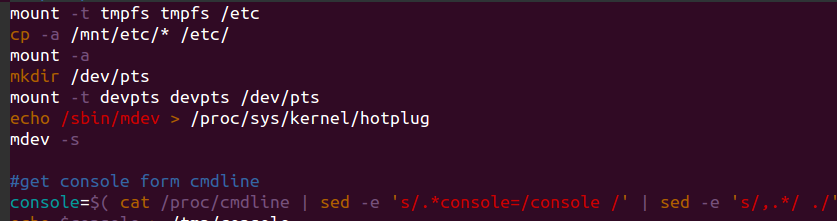


2./mnt目录下的etc



首先启动根目录下的/etc/init.d/rcS脚本文件

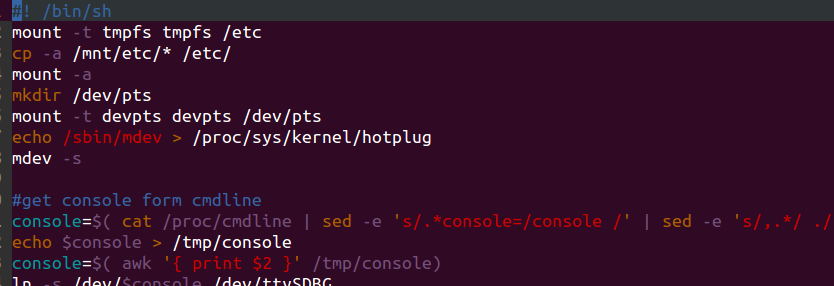




因为/etc目录下的文件涉及到读写权限问题，所以将/mnt/etc目录下一模一样的rcS文件拷贝过来执行

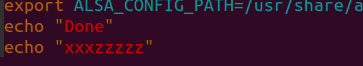
所以在根目录下/etc/init.d/rcs最后几行写自己定义的启动脚本就执行不了。

那我们就要去看/mnt/etc/init.d目录下的rcS



其实/mnt/etc/init.d目录下的rcS和/etc/init.d目录下的rcS是一样的，它们执行完后会自动启动profile文件。所以我们关键点是去修改profile文件，但是修改的profile文件必须是/mnt/etc目录下的profile





你看执行了自己定义的程序。

如果要执行你自己编写的hello程序，不能在X86上直接把程序放在/mnt/yaffs2上面，然后做出文件系统烧录进开发板，因为yaffs2目录下的文件在X86上无法做进文件系统的。

只能先在X86根目录下建立一个自己的文件

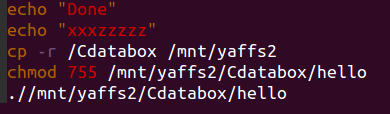




将你写的程序hello拷贝进自己在根文件系统建立的自定义目录Cdatabox

然你还不能在profile文件里面指定开机后去根目录下执行hello程序，因为根目录是cramfs文件系统，是只读属性。

你只能在profile上添加开机后自动拷贝Cdatabox目录到yaffs2文件系统



开机自动将Cdatabox目录放到可读写的yaffs2文件系统，然后给权限，执行yaffs2文件系统的应用程序

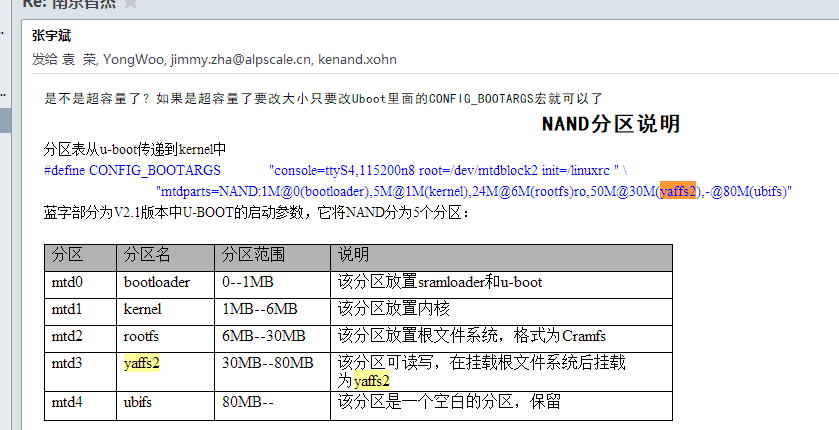
这样就可以正常实现开机自启动应用程序功能了。

所以Cramfs文件系统就是在X86把你自己写好的文件放在根目录下，然后交给shell脚本程序将自己的应用程序移动到yaffs2下执行。如果你要做在线自动更新你就只能用shell去判断yaffs2是否有以前的应用程序，如果有就停止拷贝根目录的应用程序覆盖yaffs2的应用程序，直接执行yaffs2上的应用程序去更新系统。

在烧写新板子时一般yaffs2没有你的应用程序，所以shell判断成功，拷贝应用程序去覆盖yaffs2目录

**增大yaffs2文件系统存储容量，需要修改uboot**

记住容量不能超过nandflash的大小，要保留10M的余量

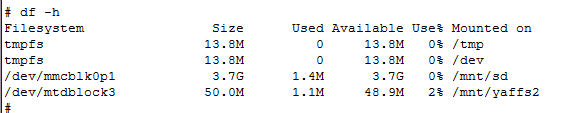


在uboot目录下



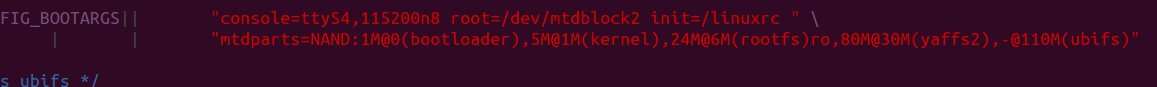




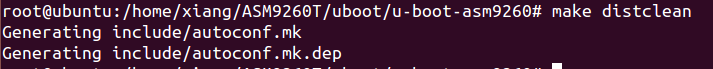


这是现在的版本yaffs2只有50M容量

修改CONFIG\_BOOTARGS的参数



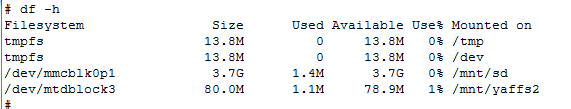
修改yaffs2为80M，然后保存退出，确定Makefile交叉编译器路径。





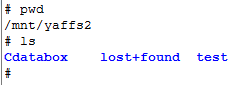


生成了u-boot.bin文件，然后把u-boot.bin烧写进开发板



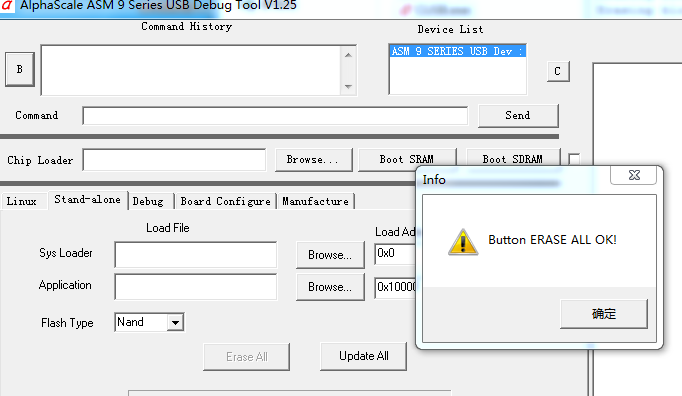
yaffs2扩容到了80M

**擦除清理yaffs2目录下的文件**

不论是我怎么下载内核，文件系统，yaffs2目录下的多余调试文件始终没有清空。

清空yaffs2目录下的文件方法是先清除整个flash，然后重新烧写uboot，内核，文件系统

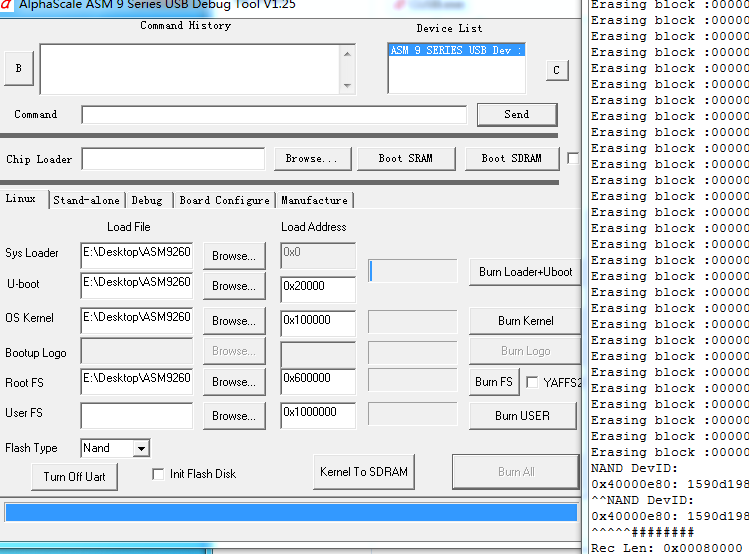
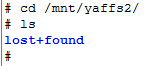
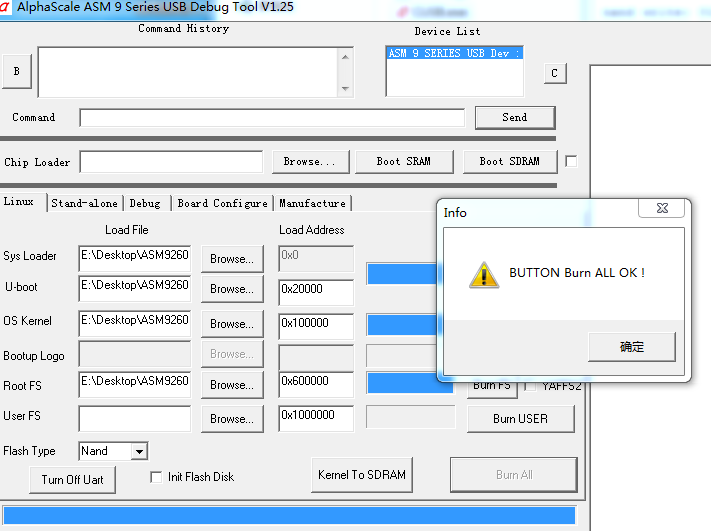
1.把拨码开关拨码到下载模式



然后点击擦除

最后显示擦除成功

先点击B

Yaffs2目录下文件已经干净了

最后重新烧写uboot，内核，和文件系统