BBPOS

1.取代码：

wisepad2:

旧代码，裸机: git clone git@52.74.102.170:firmware\_fdk

新代码，freertos: git clone git@52.74.102.170:wisepad2\_freertos

wisepos:

ssh到服务器以后，

ssh kevindai@59.152.251.6

git clone git@localhost:m8527

money store 源码地址：git@52.74.102.170:money\_store

使用winscp连接，账户：[kevindai@59.152.251.6](mailto:kevindai@59.152.251.6) 密码：bbpos

开发者选项，设置里面连击版本号五次，密码是：BBPPOSDEV

<http://52.74.102.170/mantisbt/my_view_page.php>

密码：bbpos@kevin

2.取服务器代码时，用字符代替服务器IP地址。

在~/.ssh目录下新建一个config文件

文件内容如下，

Host BBPOS //字符”BBPOS即表示替代其后IP地址”

HostName 52.74.102.170

User git

IdentityFile ~/.ssh/id\_rsa

Host BBPOSSCM

HostName 59.152.251.6

User android

IdentityFile ~/.ssh/id\_rsa

取代码时候可以写成：git clone git@BBPOS:wisepad2\_freertos

来替换原始的：git clone [git@52.74.102.170:wisepad2\_freertos](mailto:git@52.74.102.170:wisepad2_freertos)

3.FreeOS 延时函数：

vTaskDelay(2000/portTICK\_RATE\_MS); //2000为延时ms数

需要添加头文件：

#include "FreeRTOS.h"

#include "task.h"

4.创建任务函数

xTaskCreate(usb\_VBUSPolling, "USB\_VBUS Polling Task",configMINIMAL\_STACK\_SIZE, NULL ,(UBaseType\_t)1U, NULL);

回调函数需要用死循环括起来，任务创建后，一直执行该回调函数。回调函数最后需要执行延时函数，让出CPU给其他任务使用。

void usb\_VBUSPolling(void \*argv) {

while(1) //任务创建后，一直执行该回调函数

{

switch (usb\_VBUSPollState) {

case 0:

SIM\_SCGC6 |= SIM\_SCGC6\_USBDCD\_MASK;

SIM\_SCGC4 |= SIM\_SCGC4\_USBOTG\_MASK;

USBDCD\_CONTROL |= USBDCD\_CONTROL\_SR\_MASK;

USBDCD\_TIMER0 = 0x1F400FF;

USBDCD\_TIMER1 = 0x1F400FF;

USBDCD\_TIMER2 = 0x1F4000A;

USBDCD\_CONTROL |= USBDCD\_CONTROL\_START\_MASK;

USBDCD\_CONTROL |= USBDCD\_CONTROL\_IACK\_MASK;

usb\_VBUSPollState = 1;

break;

case 1:

if ((USBDCD\_CONTROL & USBDCD\_CONTROL\_IF\_MASK) != 0)

usb\_VBUSPollState = 2;

break;

case 2:

USBDCD\_CONTROL |= USBDCD\_CONTROL\_IACK\_MASK;

if ((USB0\_OBSERVE & 0x90) == 0x90)

usb\_IsVBUSConnected = 1;

else

usb\_IsVBUSConnected = 0;

usb\_VBUSPollState = 0;

USBDCD\_CONTROL |= USBDCD\_CONTROL\_SR\_MASK;

break;

}

taskYIELD(); //延时函数

}

}

5.git ignore 配置(作用：设置git 文件管理中忽略的文件)

在git目录创建.gitignore文件

Vim .gitignore

文件内容：

\*.uvprojx //表示忽略所有以.uvprojx结尾的文件

/amadis-emvco //表示忽略该文件夹

WisePad2.uvprojx //表示忽略该文件

6.git mergetool 配置（git merge 中产生冲突时，调用此处设置的工具编辑解决冲突文件）

设置beyondcompare为默认的git mergetool，打开 .gitconfig 配置文件，输入以下内容:

[diff]

tool = bc3

[difftool "bc3"]

path = D:/Program Files (x86)/Beyond Compare 3/BCompare.exe

[merge]

tool = bc3

[mergetool "bc3"]

path = D:/Program Files (x86)/Beyond Compare 3/BCompare.exe

在git merge 有冲突时，执行git mergetool,便会调用beyondcompare。

在beyondcompare中出现四个窗口，上面并排的三个依次是logcal(当前分支文件内容),base,remote(需要merge分支内容)，最下面的窗口为合并后的最终文件，合并完成后，点击保存，便可以消除该文件冲突。合并后会产生一个记录文件变动的.orig结尾的临时文件，可以删除。

解决完冲突以后还是会显示（bracnch名称|MERGING），此时再执行git commit ,状态变为（branch名称）才表示合并完成

7.Android 单独编译Launcher3

执行：./mk -o cayan\_m8527\_kk(project) packages/apps/Launcher3(location)

编译OK后会生成对应的.apk文件

编译OK后打包生成.img文件，执行./mk cayan\_m8527\_kk(project) systemimage

8.隐藏APP的icon

1).在

/m8527/oversea/alps/mediatek/custom/cayan\_m8527\_kk/resource\_overlay/generic/packages/apps/Launcher3/res/values/arrays.xml 文件中加入需要隐藏的icon名称，内容如下：

<?xml version="1.0" encoding="utf-8"?>

<resources xmlns:xliff="urn:oasis:names:tc:xliff:document:1.2">

<string-array name="array\_hide\_app\_icon" translatable="false">

<item>com.mediatek.FMRadio</item>

<item>com.google.android.gm</item>

<item>com.google.android.apps.plus</item>

<item>com.google.android.apps.maps</item>

<item>com.android.videoeditor</item>

<item>com.android.music</item>

<item>com.android.soundrecorder</item>

<item>com.mediatek.todos</item>

</string-array>

</resources>

2).在

**/oversea/alps/packages/apps/Launcher3/src/com/android/launcher3/LauncherModel.java文件中修改如下内容（蓝色字体为修改内容）：**

public void run() {

Log.d(TAG, "ChunkThread: start = " + mStart + ", end = " + mEnd + ", this = " + this);

+ // Jim: add for hide app icon

+ String[] hideAppIconArray = mContext.getResources().getStringArray(R.array.array\_hide\_app\_icon);

+ Log.d(TAG, "hide apps:");

+ for (String str : hideAppIconArray) {

+ Log.d(TAG, str);

+ }

+ // Jim: end add

ArrayList<AppInfo> list = new ArrayList<AppInfo>();

for (int i=mStart; i<mEnd; i++) {

+ // Jim: for hide app icon

+ ResolveInfo app = mApps.get(i);

+ ComponentName componentName = getComponentNameFromResolveInfo(app);

+ String packageName = componentName.getPackageName();

+ Log.d(TAG, "package name:" + packageName);

+ boolean needHide = false;

+ for (String str : hideAppIconArray) {

+ if (str != null && str.equals(packageName)) {

+ needHide = true;

+ break;

+ }

+ }

+ if (needHide) {

+ mApps.remove(i);

+ i--;

+ mEnd--;

+ continue;

+ }

+ // Jim: end add

// This builds the icon bitmaps.

list.add(new AppInfo(mPackageManager, mApps.get(i),

mIconCache, mLabelCache));

@@ -2753,9 +2779,32 @@ public class LauncherModel extends BroadcastReceiver {

}

} else {

Log.d(TAG, "ChunkThread: single-thread");

+ // Jim: add for hide app icon

+ String[] hideAppIconArray = mContext.getResources().getStringArray(R.array.array\_hide\_app\_icon);

+ Log.d(TAG, "hide apps:");

+ for (String str : hideAppIconArray) {

+ Log.d(TAG, str);

+ }

+ // Jim: end add

// Create the ApplicationInfos

for (int i = 0; i < apps.size(); i++) {

ResolveInfo app = apps.get(i);

+ // Jim: for hide app icon

+ ComponentName componentName = getComponentNameFromResolveInfo(app);

+ String packageName = componentName.getPackageName();

+ Log.d(TAG, "package name:" + packageName);

+ boolean needHide = false;

+ for (String str : hideAppIconArray) {

+ if (str != null && str.equals(packageName)) {

+ needHide = true;

+ break;

+ }

+ }

+ if (needHide) {

+ //apps.remove(i);

+ continue;

+ }

// Jim: end add

// This builds the icon bitmaps.

mBgAllAppsList.add(new AppInfo(packageManager, app,

mIconCache, mLabelCache));

9.使用scp拷贝文件到本地

scp kevin/m8527/oversea/alps/wisepos\_axis\_m8527\_kk\_eng\_MSAX\_WSC10\_0.59.714\_2016072112.zip "D:\Kevin\02\_img"

10．user版本手动打开log

在拨号界面输入\*#\*#31582525#\*#\*

11．读取Android机器目前状态值

1).adb shell

2).getprop

12.读取Android设备实时log

1) adb logcat > D:\log.txt

2) adb shell cat proc/kmsg > Desktop/kmsg.log

13.修改Android默认时区

1).设置默认时区

PRODUCT\_PROPERTY\_OVERRIDES += \

        persist.sys.timezone=Asia/Shanghai\

注：搜索“persist.sys.timezone”，并更改其值

persist.sys.timezone的值可以通过getprop获取得到

如果涉及到修改deflaut文件时，最好到相应的工程目录下修改，否则会影响其他工程，例如：

**需要修改/oversea/alps/frameworks/base/packages/SettingsProvider/res/values/defaults.xml文件时，**

可以直接在以下目录创建一个同名文件，只添加需要修改项即可。

**/oversea/alps/mediatek/custom/cayan\_m8527\_kk/resource\_overlay/generic/frameworks/base/packages/SettingsProvider/res/values/defaults.xml**

**编译时会覆盖过去。**

14．Mtk快速开机

1).将文件**/oversea/alps/frameworks/base/packages/SettingsProvider/res/values/mtk\_defaults.xml中**

- <bool name="def\_ipo\_setting" translatable="false">false</bool>

+ <!--Jim: default enable quick boot -->

+ <bool name="def\_ipo\_setting" translatable="false">true</bool>

修改为ture，默认打开快速开关机。

2).将文件

**/oversea/alps/packages/apps/Settings/res/xml/accessibility\_settings.xml中**

- android:persistent="false" />

+ android:persistent="true" />

修改为ture后，设置界面中的Quick boot选项默认为勾选。

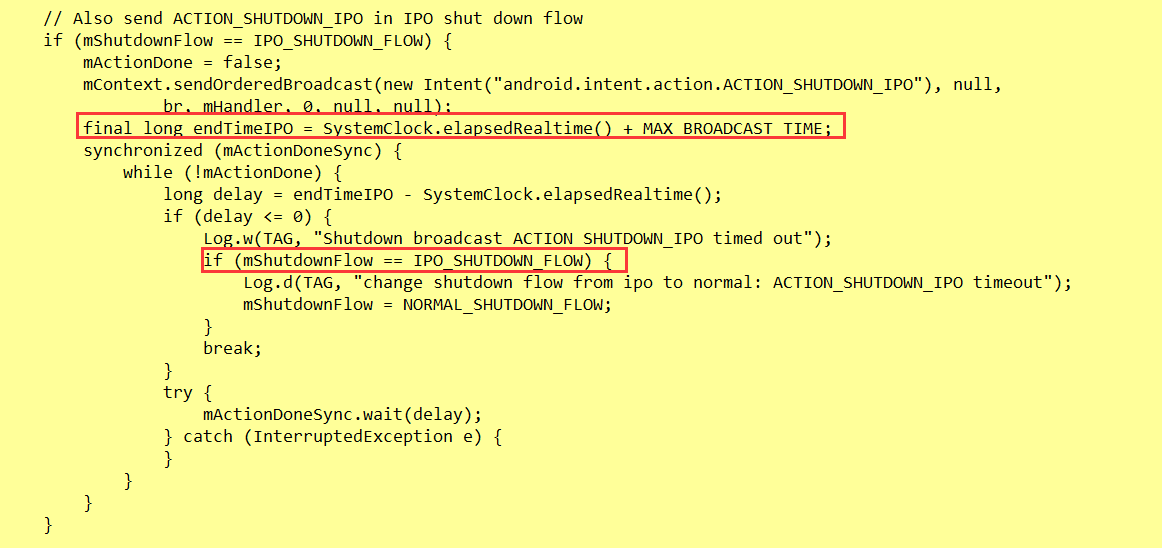
3).开启快速开关机后，如果在开机后马上关机，重启后快速开机不生效，此时可以修改如下文件

**/oversea/alps/frameworks/base/services/java/com/android/server/power/ShutdownThread.java中**

- private static final int MAX\_BROADCAST\_TIME = 10\*1000;

+ private static final int MAX\_BROADCAST\_TIME = 20\*1000; //10\*1000; // Jim: improve ipo problem

因为，即便是打开了快速开机，开机后马上关机时，有些服务还没有完成初始化，如果立马关机，会有最长MAX\_BROADCAST\_TIME时间的延时去完成设备状态的获取，如果超时还没有获取到，还是会走正常关机流程NORMAL\_SHUTDOWN\_FLOW，所以此处可以加大一段时间的延时，等待设备完成状态的获取。



14．仅仅修改git上次提交的修改记录

1)git commit --amend -m ”YOUR-NEW-COMMIT-MESSAGE”

2)git push <remote> <branch> --force

注意：修改完成后，git status 是看不到修改状态的，此时必须执行步骤2才能将最新修改的内容push 到远程端。

15.Android设备IMEI号获取（AT+CGSN 获取IMEI号）

Adb log 可以看到IMEI相关log在radio\_log.boot中

在HAL层获取IMEI代码在 oversea/alps/mediate/hardware/ril/mkt-ril中ril\_oem.c文件中，相关代码如下：

extern void bootupGetImei(RILId rid) {

LOGE("bootupGetImei[%d]", rid);

ATResponse \*p\_response = NULL;

int err = at\_send\_command\_numeric("AT+CGSN", &p\_response, getDefaultChannelCtx(rid));

if (err >= 0 && p\_response->success != 0) {

err = asprintf(&s\_imei[rid], "%s", p\_response->p\_intermediates->line);

if(err < 0)

LOGE("bootupGetImei[%d] set fail", rid);

} else {

LOGE("bootupGetImei[%d] Fail", rid);

}

at\_response\_free(p\_response);

}

extern void bootupGetImeisv(RILId rid) {

LOGE("bootupGetImeisv[%d]", rid);

ATResponse \*p\_response = NULL;

int err = at\_send\_command\_singleline("AT+EGMR=0,9", "+EGMR:",&p\_response, getDefaultChannelCtx(rid));

if (err >= 0 && p\_response->success != 0) {

char\* sv = NULL;

char\* line = p\_response->p\_intermediates->line;

err = at\_tok\_start(&line);

if(err >= 0) {

err = at\_tok\_nextstr(&line, &sv);

if(err >= 0) {

err = asprintf(&s\_imeisv[rid], "%s", sv);

if(err < 0)

LOGE("bootupGetImeisv[%d] set fail", rid);

} else {

LOGE("bootupGetImeisv[%d] get token fail", rid);

}

} else {

LOGE("bootupGetImeisv[%d] AT CMD fail", rid);

}

} else {

LOGE("bootupGetImeisv[%d] Fail", rid);

}

at\_response\_free(p\_response);

}

Framework 层代码在oversea/alps/frameworks/opt/telephony/src/java/com/android/internal/telephony/gsm/GSMPhone.java中，相关代码如下：

case EVENT\_GET\_IMEI\_DONE:

ar = (AsyncResult)msg.obj;

//MTK-START [mtk04070][111118][ALPS00093395]MTK modified

if (ar.exception != null) {

LOGW("Null IMEI!!");

setDeviceIdAbnormal(1);

break;

}

mImei = (String)ar.result;

LOGD("IMEI: " + mImei);

try {

Long.parseLong(mImei);

setDeviceIdAbnormal(0);

} catch (NumberFormatException e) {

setDeviceIdAbnormal(1);

LOGW("Invalid format IMEI!!");

}

//MTK-END [mtk04070][111118][ALPS00093395]MTK modified

break;

case EVENT\_GET\_IMEISV\_DONE:

ar = (AsyncResult)msg.obj;

if (ar.exception != null) {

break;

}

mImeiSv = (String)ar.result;

break;

//MTK-START [mtk04070][111118][ALPS00093395]MTK added

//Add by mtk80372 for Barcode Number

case EVENT\_GET\_BARCODE\_NUMBER:

LOGW("enter EVENT\_GET\_BARCODE\_NUMBER");

ar = (AsyncResult)msg.obj;

if (ar.exception != null) {

break;

}

LOGW("Barcode number is : " + (String)ar.result);

mSN = (String)(ar.result);

break;

//MTK-END [mtk04070][111118][ALPS00093395]MTK added

16.编译的时候避免服务器连接中断，尽量使用tmux进行编译。在tmux控制台下翻页

1).Ctrl+b

2).再按“{”，使用up、down就能翻页

17. 使用SD卡内置方式升级相关宏的设置

1).关于打开MTK\_2SDCARD\_SWAP 宏后MTK目前升级方案

如果设置宏MTK\_2SDCARD\_SWAP=yes；默认是优先从外卡获取升级包，外卡没有包，才会去内卡获取！

2).关于打开MTK\_SHARED\_SDCARD宏后MTK目前升级方案

如果设置宏MTK\_SHARED\_SDCARD=yes；默认是把升级包放到外置SD卡才能升级成功。即这种情况下不 支持内卡升级。

18.MTK下Android升级

1>.使用内置SD卡升级系统

**ENG版本：**

1).编译完整升级包

[SOLUTION]

Find the steps as the following:

1. Change directory to the root of code base

2. ./makeMtk <project> new

3. ./makeMtk <project> updatepackage

4. copy

oversea/alps/out/target/product/bbpos\_m8527\_kk/obj/PACKAGING/target\_files\_intermediates/ bbpos\_m8527\_kk-target\_files-eng.JB5.zip to

root directory of SD card and rename it to update.zip(The full OTA package size is about equal to boot.img size + system.img size.) （此步骤可以省略）

5. ./build/tools/releasetools/ota\_from\_target\_files out/target/product/bbpos\_m8527\_kk/obj/PACKAGING/target\_files\_intermediates/bbpos\_m8527\_kk-target\_files-eng.JB5.zip ota\_target\_update.zip (打包)

6. java -Xmx1024m -jar out/host/linux-x86/framework/signapk.jar -w build/target/product/security/bbpos\_m8527\_kk/releasekey.x509.pem build/target/product/security/bbpos\_m8527\_kk/releasekey.pk8 ota\_target\_update.zip sign\_ota\_target\_update.zip（给升级包签名）

7.使用adb进行升级

a). adb push "d:\Kevin\02\_img\Newfolder\update.zip" sdcard/

b). adb shell "echo \"--update\_package=/sdcard/update.zip\" > cache/recovery/command"（如果使用adb时，对于USER 版本此处没有权限，需要修改init.rc文件中cache文件夹权限）

c). adb reboot recovery

**USER版本：**

1).编译完整升级包

1. source mbldenv.sh
2. ./mk -o=TARGET\_BUILD\_VARIANT=user bbpos\_m8527\_kk new
3. ./mk -o=TARGET\_BUILD\_VARIANT=user bbpos\_m8527\_kk updatepackage
4. ./build/tools/releasetools/ota\_from\_target\_files out/target/product/bbpos\_m8527\_kk/obj/PACKAGING/target\_files\_intermediates/bbpos\_m8527\_kk-target\_files-user.JB5.zip ota\_target\_update.zip (打包)
5. java -Xmx1024m -jar out/host/linux-x86/framework/signapk.jar -w build/target/product/security/bbpos\_m8527\_kk/releasekey.x509.pem build/target/product/security/bbpos\_m8527\_kk/releasekey.pk8 ota\_target\_update.zip update.zip（给升级包签名）（注意要根据实际情况来使用对应的key进行签名）

19.MTK 同时支持内置和外置SD 卡升级的方式

1).Add -DSUPPORT\_SDCARD2 in bootable/recovery/Android.mk，修改如下：

+#Kevin add for support external SDCARD in recover mode

+ifeq ($(WITH\_BACKUP\_RESTORE),true)

+LOCAL\_CFLAGS += -DSUPPORT\_DATA\_BACKUP\_RESTORE -DSUPPORT\_SDCARD2

+endif

+#Kevin add for support external SDCARD in recover mode

2). recovery.fstab的优先级是project > platform > common， 还请根据项目的具体情况来修改recovery.fstab。比如，project的fstab需要修改alps/mediatek/config/<project>/recovery\_xxx.fstab。

a) KK前的版本  
emmc为：  
/sdcard2vfat/dev/block/mmcblk1p1/dev/block/mmcblk1  
nand为：  
/sdcard2vfat/dev/block/mmcblk0p1/dev/block/mmcblk0  
b)KK版本  
emmc为： /dev/block/mmcblk1p1 /sdcard2 vfat defaults defaults  
nand为：  
/dev/block/mmcblk0p1 /sdcard2 vfat defaults defaults

MTK m8527 修改如下：

/oversea/alps/mediatek/config/bbpos\_m8527\_kk/recovery.fstab

+#Kevin add for support External SDCARD in recover mode

+/sdcard2 vfat /dev/block/mmcblk1p1

20.指定变量存储区域data段和bss段

data段和bss段的区别：

全局的未初始化变量存在于.bss段中，具体体现为一个占位符；全局的已初始化变量存于.data段中；而函数内的自动变量都在栈上分配空间。.bss是不占用.exe文件空间的，其内容由操作系统初始化（清零）；而.data却需要占用，其内容由程序初始化。

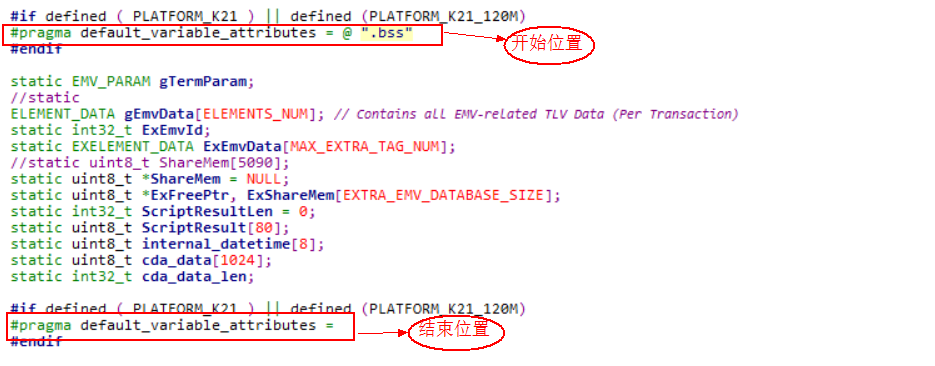
Wisepad项目，由于内存空间不足，需要将data区域空间释放一点出来时，可以指定将data段数据放在bss段。具体如下：

IARv6.7版本以后增加了两个新的pragma指令：#pragma default\_function\_attributes

和#pragma default\_variable\_attributes。在需要指定位置的函数或者变量前后，以一定格式加上这两个指令，就可以轻松地将多个函数或变量放在指定的位置。

在需要指定位置的程序组开始的位置，用”#pragma default\_function\_attributes = @“SECTION NAME” ”的格式说明要将下面的函数放到哪个SECTION。

然后在程序组结束的位置，用”#pragma default\_function\_attributes = “的格式说明从这里开始定义的函数不再放到前面指定的SECTION。



21.icf文件的作用

icf文件就是链接文件。其作用是:

1.定义了芯片存储空间的大小。

2.定义ROM的大小和起始、结束地址。

3.定义RAM的大小和起始、结束地址。

同时还可以:

1.链接时是否保存位块与符号

2.设置你的启动过程

3.设置你的堆与栈的大小和对齐方式

4.修改编译器默认的初始化过程。

5.关于默认库的处理等

22.map文件的作用

毕竟有些时候IDE提示的信息不够明白，使用MAP文件是一个调试的好帮手。通过map文件可以查找崩溃地址，然后就可以精确地定位到源代码中出错的代码行

MAP 文件是程序的全局符号、源文件和代码行号信息的唯一的文本表示方法，它可以在任何地方、任何时候使用，不需要有额外的程序进行支持。而且，这是唯一能找出**程序崩溃**的地方的救星。

22.gitlog的拓展命令， 加入以下文件加到.gitconfig

[alias]

#quick look at all repo

loggsa = log --color --date-order --graph --oneline --decorate --simplify-by-decoration --all

#quick look at active branch (or refs pointed)

loggs = log --color --date-order --graph --oneline --decorate --simplify-by-decoration

#extend look at all repo

logga = log --color --date-order --graph --oneline --decorate --all

#extend look at active branch

logg = log --color --date-order --graph --oneline --decorate

#Look with date

logda = log --color --date-order --date=local --graph --format=\"%C(auto)%h%Creset %C(blue bold)%ad%Creset %C(auto)%d%Creset %s\" --all

logd = log --color --date-order --date=local --graph --format=\"%C(auto)%h%Creset %C(blue bold)%ad%Creset %C(auto)%d%Creset %s\"

#Look with relative date

logdra = log --color --date-order --graph --format=\"%C(auto)%h%Creset %C(blue bold)%ar%Creset %C(auto)%d%Creset %s\" --all

logdr = log --color --date-order --graph --format=\"%C(auto)%h%Creset %C(blue bold)%ar%Creset %C(auto)%d%Creset %s\"

loga = log --graph --color --decorate --all

# For repos without subject body commits (vim repo, git-svn clones)

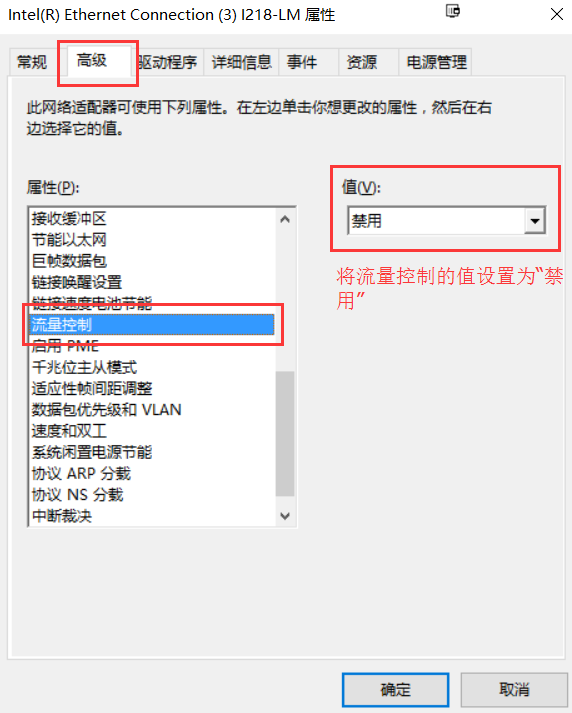
logt = log --graph --color --format=\"%C(auto)%h %d %<|(100,trunc) %s\"

logta = log --graph --color --format=\"%C(auto)%h %d %<|(100,trunc) %s\" --all

logtsa = log --graph --color --format=\"%C(auto)%h %d %<|(100,trunc) %s\" --all --simplify-by-decoration

23.winscp 拷贝速度慢

解决方法：

找到计算机网卡，进行如下操作。

24.tux 后台保存任务

1).任务一执行过程中

2).按ctrl +Z 将当前任务保存在后台中

3).打开任务二

4).按ctrl+Z，输入jobs 可以查看当前后台运行任务，按fg +任务前面需要可以切换到相应的任务

25.查看挂载的module

1).adb shell

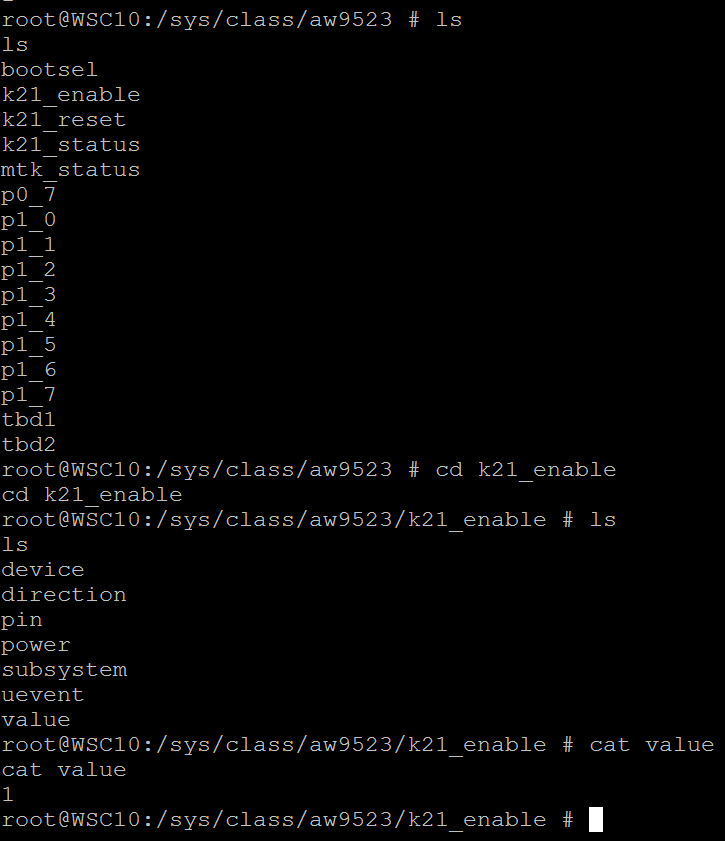
2).cd /system/lib/modules

26.查看设备节点

eg:查看aw9523驱动节点

cd sys/class/aw9523

cat k21\_enable/value



27 . IAR 打印heap大小

\_\_iar\_dlmalloc\_stats()

https://www.iar.com/support/tech-notes/general/iar-dlib-library-heap-usage-statistics/

28.IAR 使用\_\_FILE\_\_宏时不要打印出相对路径

--no\_path\_in\_file\_macros

## 29. MT6589 DSI Video Mode的屏，加入ESD机制后每隔2S会闪一下屏

关掉ESD check 功能