安卓8~开启btsnoop

setprop persist.bluetooth.btsnoopenable true

setprop persist.bluetooth.btsnooplogmode full

setprop persist.bluetooth.btsnoopdefaultmode full

setprop persist.bluetooth.btsnoopsize 0xfffffff

setprop persist.bluetooth.btsnooppath /data/misc/bluetooth/logs/btsnoop\_hci.log

Thinkpad 有关电池

https://blog.csdn.net/dianxiang0791/article/details/101604493

Git

https://backlog.com/git-tutorial/cn/stepup/stepup2\_3.html

Bluez

https://blog.csdn.net/zhuo\_lee\_new/article/details/106626680

<https://blog.csdn.net/qq_21949217/article/details/49739005>

<https://blog.csdn.net/hwh295/article/details/113409389>

<https://wenku.baidu.com/view/33c6add8cbd376eeaeaad1f34693daef5ff71349.html>

Pci相关总线

Ubuntu安装，/boot EFI系统？

<https://blog.csdn.net/qq_42030961/article/details/82740315>

<https://www.cnblogs.com/111testing/p/13668275.html>

BLE速率：<https://www.jianshu.com/p/a5f626d09567>

[https://mp.weixin.qq.com/s/12YaDn4sF5f3xTBYA9e2fA](https://mp.weixin.qq.com/s/12YaDn4sF5f3xTBYA9e2fA" \t "_blank)

//Jira账户

<https://synaextjira.atlassian.net/jira/dashboards/last-visited>

luke.chen@ampak.com.tw/Gb9619Gb9619

//vpn

user: jay-wu

pw: jay-wu@7340

211.75.123.148:443

svn co <http://10.10.29.8/svn/SiP/Broadcom/>

Update

https://www.cnblogs.com/liuxiaowei/p/8297268.html

新安装的minicom默认打开/dev/modem，如果/dev下没有这个设备的话，就会报错打不开

解决办法，执行

ln -s /dev/ttyS0 /dev/modem

第一种：使用 “nohup”，即 "nohup ./test.sh &"，忽略hangup信号，防止终端关闭时程序退出。

运行状态可以用 “jobs -l” 来查看。

第二种：使用 “setsid” 即 “setsid ./test.sh” 。

第三种：最简单，将命令放入括号当中，比如说：（./test &），这样就行了。

a2dp切换sink

You can turn on all the profiles you need from packages/apps/Bluetooth/ res/values/config.xml and set default disabled profiles def\_bluetooth\_disabled\_profiles from frameworks/base/packages/SettingsProvider/res/values/defaults.xml.

For example, host role (a2dp src/hfp ag) is the default one, so we need to disable a2dp sink and hf client. Hence, the value should be 2^16+2^11 = 67548.  you can refer to frameworks/base/ core/java/android/bluetooth/BluetoothProfile.java for the bitmask for all the profiles.

For dynamically changed the role, below command is ready for the job.

adb shell settings put global bluetooth\_disabled\_profiles <value>

安卓aosp源码刷机等。

# [AOSP下载编译并刷机最新分步骤详解](https://www.cnblogs.com/ciml/p/13158225.html)

#### 我的环境

Ubuntu 18.04.5 LTS

清华源

https://mirrors.tuna.tsinghua.edu.cn/help/AOSP/

安装repo, 3种方式

[IMG_256](javascript:void(0);)

# 1. 无需安装, 下载初始化包中自带最好用  
  
# 2. ubuntu apt-get  
sudo apt-get install repo

# 3. 单独脚本的方式(不推荐, 至少在我的电脑上, 会报一些python脚本兼容性错误, 手动改了又会遇到uncommit change错误)  
mkdir ~/bin

PATH=~/bin:$PATH

curl https://storage.googleapis.com/git-repo-downloads/repo > ~/bin/repochmod a+x ~/bin/repo

[IMG_257](javascript:void(0);)

#### 使用每月更新的初始化包

[IMG_258](javascript:void(0);)

# 下载初始化包, 查看包列表 https://mirrors.tuna.tsinghua.edu.cn/aosp-monthly/, 由于最新包太大了, 我只需要Android-8.0的包  
wget -c https://mirrors.tuna.tsinghua.edu.cn/aosp-monthly/aosp-20190101.tar  
# 或下载最新包8-90G吧wget -c https://mirrors.tuna.tsinghua.edu.cn/aosp-monthly/aosp-latest.tar

# 解压得到的 AOSP 工程目录tar xf aosp-latest.tar

cd AOSP

# 这时 ls 的话什么也看不到，因为只有一个隐藏的 .repo 目录  
# 正常同步一下,即可得到完整目录

repo sync

# 或 repo sync -l 仅checkout代码(这种方式比较快,不会fetch最新的远程仓库,也就是如果我下载的是20190101.tar,则最新的修改就到这天,之后至今天的修改,不同步.)  
# 如果不加 -l 选项, 则更新本地仓库为最新上百G了.

[IMG_259](javascript:void(0);)

#### 查看分支列表

# 进入Repo目录，查看所有分支

cd .repo/manifests

git branch -a  
  
# 如果像我一样使用repo sync -l选项同步, 会发现分支最新只到android-9.0

#### 切换到需要的目标分支

查看 https://source.android.com/setup/start/build-numbers#source-code-tags-and-builds 找到你想使用的分支, 这里我选择 android-8.1.0\_r35 , 原谅我手上只有一台Pixel 2. 所以要选择支持我这台设备的分支, 为后续刷机作准备

# 切换分支

repo init -b android-8.1.0\_r35

**如果提示无法连接到 gerrit.googlesource.com，请参照**[git-repo的帮助页面](https://mirrors.tuna.tsinghua.edu.cn/help/git-repo)**的更新一节。**

#### 环境配置

sudo apt-get updatesudo apt-get install openjdk-8-jdk  
  
# 安装必要程序包 (我就不装)  
sudo apt-get install git-core gnupg flex bison gperf build-essential zip curl zlib1g-dev gcc-multilib g++-multilib libc6-dev-i386 lib32ncurses5-dev x11proto-core-dev libx11-dev lib32z-dev ccache libgl1-mesa-dev libxml2-utils xsltproc unzip

#### 编译

source build/envsetup.sh  
lunch aosp\_arm64-eng  
  
# 若想查看所有lunch选项, 可以输入lunch 回车

我们先编译一个模拟器版的, 因为AOSP缺少vendor目录, 即不包含实体设备的驱动

输入2 回车, 接下来

make -j8

# 我的电脑是4核8线程的, 一次全编 它可以从早编到晚 ~~

不出意外会编译成功. 看下输出目录, 会有各种img

ls $OUT

 编译完成后可以运行模拟器看下效果

emulator

[下一篇我们来刷ROM到实体手机](https://www.cnblogs.com/ciml/p/13714036.html" \t "_blank)

#### 关于lunch构建目标

| 构建类型 | 说明 |
| --- | --- |
| user | 权限受限；适用于生产环境 |
| userdebug | 与“user”类似，但具有 root 权限和调试功能；是进行调试时的首选编译类型 |
| eng | 具有额外调试工具的开发配置, 会关闭用于提供良好用户体验的各种优化 |

有意外发生 ~

#### 遇到的错误

[IMG_260](javascript:void(0);)

# 报python语法错误  
acer@acer-TravelMate-P249-G3-MG:~/aosp-test/aosp$ repo sync -l

Traceback (most recent call last):

File "/home/acer/aosp-test/aosp/.repo/repo/git\_config.py", line 198, in GetBranch

b = self.\_branches[name]

KeyError: 'default'

During handling of the above exception, another exception occurred:

Traceback (most recent call last):

File "/home/acer/aosp-test/aosp/.repo/repo/main.py", line 531, in <module>

\_Main(sys.argv[1:])

File "/home/acer/aosp-test/aosp/.repo/repo/main.py", line 507, in \_Main

result = repo.\_Run(argv) or 0

File "/home/acer/aosp-test/aosp/.repo/repo/main.py", line 158, in \_Run

copts, cargs = cmd.OptionParser.parse\_args(argv)

File "/home/acer/aosp-test/aosp/.repo/repo/command.py", line 67, in OptionParser

self.\_Options(self.\_optparse)

File "/home/acer/aosp-test/aosp/.repo/repo/subcmds/sync.py", line 189, in \_Options

self.jobs = self.manifest.default.sync\_j

File "/home/acer/aosp-test/aosp/.repo/repo/manifest\_xml.py", line 360, in default

self.\_Load()

File "/home/acer/aosp-test/aosp/.repo/repo/manifest\_xml.py", line 400, in \_Load

b = m.GetBranch(m.CurrentBranch).merge

File "/home/acer/aosp-test/aosp/.repo/repo/project.py", line 814, in GetBranch

return self.config.GetBranch(name)

File "/home/acer/aosp-test/aosp/.repo/repo/git\_config.py", line 200, in GetBranch

b = Branch(self, name)

File "/home/acer/aosp-test/aosp/.repo/repo/git\_config.py", line 725, in \_\_init\_\_

self.merge = self.\_Get('merge')

File "/home/acer/aosp-test/aosp/.repo/repo/git\_config.py", line 768, in \_Get

return self.\_config.GetString(key, all\_keys = all\_keys)

File "/home/acer/aosp-test/aosp/.repo/repo/git\_config.py", line 131, in GetString

v = self.\_cache[\_key(name)]

File "/home/acer/aosp-test/aosp/.repo/repo/git\_config.py", line 248, in \_cache

self.\_cache\_dict = self.\_Read()

File "/home/acer/aosp-test/aosp/.repo/repo/git\_config.py", line 254, in \_Read

d = self.\_ReadGit()

File "/home/acer/aosp-test/aosp/.repo/repo/git\_config.py", line 299, in \_ReadGit

for line in d.decode('utf-8').rstrip('\0').split('\0'): # pylint: disable=W1401

AttributeError: 'str' object has no attribute 'decode'

[IMG_261](javascript:void(0);)

解决办法: 试试用 .repo/repo/repo sync -l

[IMG_262](javascript:void(0);)

# 遇到错误

[490/999] including ./hardware/qcom/keymaster/Android.mk ...

[491/999] including ./hardware/qcom/media/Android.mk ...

hardware/qcom/media/msm8998/mm-video-v4l2/vidc/venc/Android.mk:60: warning: "venc: PQ compiled out"

[492/999] including ./hardware/qcom/msm8998/Android.mk ...

including hardware/qcom/msm8998/gpt-utils/Android.mk ...

including hardware/qcom/msm8998/json-c/Android.mk ...

including hardware/qcom/msm8998/thermal-engine/Android.mk ...

including hardware/qcom/msm8998/time-services/Android.mk ...

[493/999] including ./hardware/qcom/power/Android.mk ...

[494/999] including ./hardware/qcom/sdm845/Android.mk ...

./hardware/qcom/power/Android.mk:3: error: ./hardware/qcom/sdm845/Android.mk: No such file or directory15:33:38 ckati failed with: exit status 1

[IMG_263](javascript:void(0);)

 解决: 进入hardware/qcom/sdm845, ls -l 可见一些无效的符号链接, 直接删除即可

参考:

https://source.android.com/source/initializing

<https://source.android.com/setup/build>

https://www.cnblogs.com/ciml/p/13714036.html

刷机

我的设备是\*鱼上淘的Pixel 2, 2015年之后的设备刷写步骤基本一致(Pixel 2XL, Pixel 3, Pixel 3XL, Pixel 4)

[上一篇](https://www.cnblogs.com/ciml/p/13158225.html)博客讲了如何下载编译AOSP, 但只能编译出模拟器版本, 缺少厂商驱动(没有vendor目录), 不能用来刷机, 因此首先要下载厂商驱动

### 下载驱动

https://developers.google.com/android/drivers 包含AOSP非master分支Nexus and Pixel设备的驱动. 这里我们下载 [驱动](https://developers.google.com/android/drivers" \l "walleyeopm2.171026.006.g1" \t "_blank) 与上篇提到的分支 android-8.1.0\_r35 对应

将下载的2个tar文件放到aosp根目录 解压, 安装驱动

[IMG_256](javascript:void(0);)

# 解压得到2个sh自释放文件

extract-google\_devices-walleye.sh extract-qcom-walleye.sh

# 安装 sh extract-google\_devices-walleye.shsh extract-qcom-walleye.sh

# 看完license输入 I ACCEPT 安装完成, 此时ls会看到vendor文件夹

[IMG_257](javascript:void(0);)

### 重新编译AOSP

source build/envsetup.sh  
lunch aosp\_walleye-userdebug  
# 经历漫长的等待后, 编译出镜像

### 刷写镜像

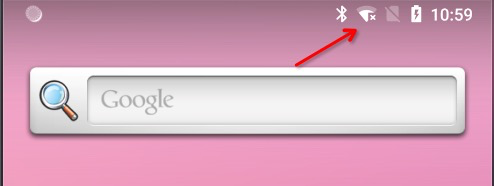
#### 1. 解锁

参考 [https://www.cnblogs.com/ciml/p/13949911.html](https://www.cnblogs.com/ciml/p/13949911.html" \t "_blank)

#### 2. 刷写设备

# -w选项意思wipe data, 会清空手机上的所有数据  
# 注意！ 此命令需要在刚才编译源码的同一终端窗口内执行， 或者执行过lunch的终端中， 否则环境变量中没有如 $OUT 的变量

刷好设备, 即可体验原版Android系统了, 但原版系统连上WiFi后, 总是提示网络连接不可用



为啥嗫-因为原版系统的网络探针访问的是Google的服务器,不够科xue的网络肯定是连不上了. 这个WiFi图标让人很不爽,下面我们来改点东西修复它.

### 修改Captive Portal Server

打开 vim frameworks/base/services/core/java/com/android/server/connectivity/NetworkMonitor.java 可以看到如下两行代码

// Default configuration values for captive portal detection probes.private static final String DEFAULT\_HTTPS\_URL = "https://www.google.com/generate\_204";private static final String DEFAULT\_HTTP\_URL =

"http://connectivitycheck.gstatic.com/generate\_204";

这就是AOSP默认的检测网络是否畅通的探针地址, AOSP默认使用https (如果你没有通过adb修改的话)

现在修改为如下:

// https://captive.v2ex.co/generate\_204 is also acceptable   
// 这里我们改为小米的, V2EX的也可以, 我测试这里小米的快一点private static final String DEFAULT\_HTTPS\_URL = "https://connect.rom.miui.com/generate\_204";private static final String DEFAULT\_HTTP\_URL =

"http://connect.rom.miui.com/generate\_204";

 重新编译, 生成亲的系统镜像, 因为framework位于system.img, 所以这里我们单独刷system分区就可以.

# 重新编译, 成功后 可见类似输出

[ 99% 615/616] Install system fs image: out/target/product/walleye/system.img  
out/target/product/walleye/system.img+ maxsize=2740531200 blocksize=135168 total=1077064032 reserve=27709440  
[100% 616/616] Target vbmeta image: out/target/product/walleye/vbmeta.img

### 单刷system分区

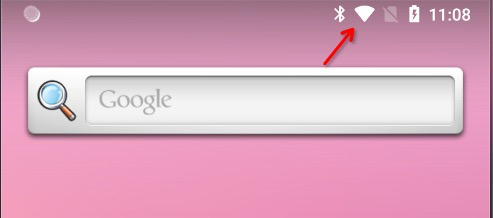
fastboot flash system system.img

fastboot flash system\_a system\_other.img

fastboot set\_active b　　　　　　　　　　　　　　# 将slot b设为启动分区

fastboot reboot

 由于Android8之后采用A/B更新, 所以有2套分区, 刷分区方式和以往不同, 参考: https://wertherzhang.com/android8-partiton-table/



补充一下: 一般修改framework目录的代码, 重新编译单刷system分区即可, 但有时可能会更新到其它分区的镜像, 要以make的最终结果为准, 比如上面我的日志, vbmeta.img镜像也有更新, 则vbmeta.img也要刷入设备, 还有一种方法是到out目录下, 执行 ls -tl 看哪些镜像的修改时间是最新的, 就刷哪些.  如果有漏刷则可能导致无法进入Android系统.

解锁

进入bootloader

# 1.开机状态, 执行

adb reboot bootloader

# 2. 关机状态, 按住音量-, 然后按开机键

解锁

[IMG_256](javascript:void(0);)

# 连接USB后, 先看看设备是否连接成功

fastboot devices

# 解锁 (用于2015年及之后的设备)

fastboot flashing unlock

# 老设备解锁 (2014及以前的设备)

fastboot oem unlock

[IMG_257](javascript:void(0);)

重新上锁 (会wipe data, 擦除用户数据, 谨慎操作)

# 进入bootloader模式, 执行

fastboot flashing lock  
  
# 老设备  
fastboot oem lock

刷写设备

fastboot flashall -w

参考：

　　https://source.android.google.cn/setup/build/running

**fastboot 介绍**

Fastboot，英语翻译意思是快速启动。 在安卓手机中fastboot是一种比recovery更底层的刷机模式（俗称引导模式）。 就是使用USB数据线连接手机的一种刷机模式。 相对于某些系统（如iOS）卡刷来说，线刷更可靠，安全。

进入fastboot有俩种方式：

在关机的情况下，同时按下power键和音量键，在boot驱动界面 手机插上USB线，进入CMD命令:

fastboot devices -- 查看设备是否连上

二．在开机情况下,手机插上USB线，进入CMD命令:

adb reboot bootloader -- 进入fastboot模式

fastboot devices -- 查看设备是否连上

将服务器上的img，拷贝到本地,打开命令行执行下面的步骤:

fastboot flashing unlock

fastboot flash boot E:\aosp\boot.img

fastboot flash ramdisk E:\aosp\ramdisk.img

fastboot flash ramdisk-recovery E:\aosp\ramdisk-recovery.img

fastboot flash system E:\aosp\system.img

fastboot flash userdata E:\aosp\userdata.img

fastboot flash vendor E:\aosp\vendor.img

fastboot flash system\_other E:\aosp\system\_other.img

fastboot reboot

执行完毕，手机将会自动重启。

全刷：

刷写设备

fastboot flashall -w

1.查看当前系统原码版本

$ apt-cache search linux-source  
linux-source - Linux kernel source with Ubuntu patches  
linux-source-2.6.35 - Linux kernel source for version 2.6.35 with Ubuntu patches

2.下载源码  
$ sudo apt-get install linux-source-2.6.35

下载完成后，在/usr/src目录中，可以看到压缩包  
$ ls  
linux-headers-2.6.35-22 linux-source-2.6.35  
linux-headers-2.6.35-22-generic linux-source-2.6.35.tar.bz2

3.编译  
$ sudo tar -xjvf linux-source-2.6.35.tar.bz2  
$ sudo make menuconfig

HOSTCC scripts/basic/fixdep  
HOSTCC scripts/basic/docproc  
HOSTCC scripts/basic/hash  
HOSTCC scripts/kconfig/conf.o  
scripts/kconfig/conf.c: In function ‘conf\_askvalue’:  
scripts/kconfig/conf.c:105: warning: ignoring return value of ‘fgets’, declared with attribute warn\_unused\_result  
scripts/kconfig/conf.c: In function ‘conf\_choice’:  
scripts/kconfig/conf.c:307: warning: ignoring return value of ‘fgets’, declared with attribute warn\_unused\_result  
HOSTCC scripts/kconfig/kxgettext.o  
\*\*\* Unable to find the ncurses libraries or the  
\*\*\* required header files.  
\*\*\* 'make menuconfig' requires the ncurses libraries.  
\*\*\*  
\*\*\* Install ncurses (ncurses-devel) and try again.  
\*\*\*  
make[1]: \*\*\* [scripts/kconfig/dochecklxdialog] 错误 1  
make: \*\*\* [menuconfig] 错误 2  
  
  
缺少ncurses库，安装  
$ sudo apt-get install libncurses5-dev

重新编译成功。也可以使用系统原来的配置来编译，oldconfig

$ sudo make oldconfig #编译内核

本文主要讲述，如何根据当前运行的Ubuntu版本，找到对应的内核源码。

首先，按照下面链接里的内容，下载对应的内核源码仓库。

https://wiki.ubuntu.com/Kernel/Dev/KernelGitGuide

如果觉得链接里的内容太长了，可以试下如下命令。

git clone git://git.launchpad.net/~ubuntu-kernel/ubuntu/+source/linux/+git/$(lsb\_release -cs)

该命令会根据你当前的Ubuntu版本下载对应的内核代码。

如果这个命令没报错，说明一切顺利，只要等待下载完成就行了。

Ubuntu内核代码下载完成之后，默认为master分支。该分支通常并不是精确对应到我们当前运行的Ubuntu版本，所以我们要切换分支。

先通过如下命令，找到当前运行的Ubuntu的精确版本号。

➜  bionic git:(master) cat /proc/version\_signature

Ubuntu 4.15.0-45.48-generic 4.15.18

其中，-generic之前的信息就对应为Ubuntu内核源码的tag，不过要把Ubuntu后的空格换成中划线。比如上面命令输出对应的tag就是 Ubuntu-4.15.0-45.48 。

然后，我们进入到下载好的内核源码目录，执行如下命令，把源码切换到该tag对应的版本。

➜  bionic git:(master) git checkout Ubuntu-4.15.0-45.48

HEAD is now at ffdd392b8196 UBUNTU: Ubuntu-4.15.0-45.48

好了，现在我们就拥有了当前运行的Ubuntu版本的内核源码，可以开始尽情读源码，做实验了。

<https://blog.csdn.net/wxh0000mm/article/details/94555238>

FASTBOOT

Adb

### 打开和关闭蓝牙BT：

adb root  
adb shell svc bluetooth enable  
adb shell svc bluetooth disable

### 查询蓝牙开启状态：

adb shell settings get global bluetooth\_on

=》0: 未开启    1: 已开启

### 获取mac地址：

方法1：

adb shell settings get secure bluetooth\_address

=》如：00:00:00:00:00:01

方法2：

adb shell adb shell service call bluetooth\_manager 13

## 5. 安装apk文件

adb install apk路径

1、打开蓝牙

adb shell service call bluetooth\_manager 6

2、关闭蓝牙

adb shell service call bluetooth\_manager 8

3、查看蓝牙信息

adb shell dumpsys bluetooth\_manager

4、允许被发现---需要按键确认

adb shell am start -a android.bluetooth.adapter.action.REQUEST\_DISCOVERABLE

5、获取蓝牙开关状态

adb shell settings get global bluetooth\_on

返回1表示开启，0表示关闭

6、获取蓝牙MAC地址

adb shell settings get secure bluetooth\_address

# [adb之打开蓝牙hci log](https://www.cnblogs.com/yingxiaoniu/p/11015971.html)

手机连接电脑，开启debug

adb shell测试设备已连接成功

指令：setprop persist.bluetooth.btsnoopenable true  开启蓝牙log

log一般是存放在/data/misc/bluetooth/logs下

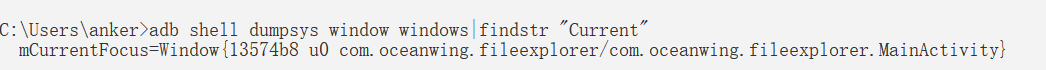
需要先root设备再push出来

# [adb指令之超级好用的获取当前app的package&activity](https://www.cnblogs.com/yingxiaoniu/p/11101436.html)

将设备（一般是手机），打开cmd指令输入以下指令

adb shell dumpsys window windows | findstr “Current”

 如下图所示，/之前的就是package，之后的就是当前的activity，如下图红线的部分。对于大多数的App，这个当前的activity就是am命令需要的launch activity， 但也有些应用启动时包含了欢迎页之类的，在你打开App的时候就自动切换了activity，那用这种方法就获取不了需要的package。



1.在cmd中，进入shell指令模式

　adb shell

2.进入ROOT指令模式

    su

3. 打开WIFI指令

   svc wifi enable

5. 关闭WIFI指令

   svc wifi disable

1、首先连接路由器和电脑

2、打开设备无线，连接路由器，（5G，2.4G）

3、首先启动WiFitest.bat

4、然后打开一个运行，控制的是设备端，

5、adb root，

6、adb remount,adb push C:\Users\Administrator\Desktop\WFI吞吐量测试\WIFI\iPerf\iperf /system/bin

7、更改权限chmod 777 /system/bin/iperf

（设备端已经就绪）

8、PC端：cd C:\Users\Administrator\Desktop\WFI吞吐量测试\WIFI\ iPerf\iperf-2.0.4-win32（路径）

9、进入路由器官网查看两个设备的IP地址。

10、测试MID下行速率：

(1) PC运行：

iperf -c x.x.x.x -i 1 -w 1M -t 60

iperf -c 192.168.1.102 -i 1 -w 1M -t 60

其中x.x.x.x为MID的IP地址

(2) MID运行：

system/bin/iperf -s

测试MID上行速率：

(1) PC运行：

iperf -s

(2) MID运行：

system/bin/iperf -c x.x.x.x -i 1 -w 1M -t 60

system/bin/iperf -c 192.168.50.192 -i 1 -w 1M -t 60

其中x.x.x.x为PC的IP地址

————————————————

版权声明：本文为CSDN博主「世联芯科技有限公司」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/ShiLianXin/article/details/109305973>

//Aops 蓝牙做sink

<https://blog.csdn.net/Jason_Lee155/article/details/115611730>

<https://blog.csdn.net/chaoli_chen/article/details/108285847>

https://blog.csdn.net/yyl1993/article/details/108061422?spm=1001.2101.3001.6650.1&utm\_medium=distribute.pc\_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-1.pc\_relevant\_default&depth\_1-utm\_source=distribute.pc\_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-1.pc\_relevant\_default&utm\_relevant\_index=2

device/qcom/msmxxx\_64/system.prop

添加如下配置

//persist.service.bt.a2dp.sink=true

//persist.service.bt.hfp.client=true

a2dp切换sink

You can turn on all the profiles you need from packages/apps/Bluetooth/ res/values/config.xml and set default disabled profiles def\_bluetooth\_disabled\_profiles from frameworks/base/packages/SettingsProvider/res/values/defaults.xml.

For example, host role (a2dp src/hfp ag) is the default one, so we need to disable a2dp sink and hf client. Hence, the value should be 2^16+2^11 = 67548.  you can refer to frameworks/base/ core/java/android/bluetooth/BluetoothProfile.java for the bitmask for all the profiles.

For dynamically changed the role, below command is ready for the job.

adb shell settings put global bluetooth\_disabled\_profiles <value>

Android 8.1开启Bluetooth A2DP sink功能

\*\*

Android O源码已经支持A2DP sink,只需要开启就可以了

一个是BTA\_AV\_SINK\_INCLUDED设为TRUE,BTA\_AV\_SINK\_INCLUDED在\system\bt\include的bt\_target.h

另外一个是bluetooth app这边的profile\_supported\_a2dp\_sink需设为true开启这个profile

但是实际上光开启这两个值A2DP sink是不能工作的，我用的是RK3399,不知道其他平台是不是OK的

调试发现这个btif\_av\_init只会执行一次，也就是加了source之后sink是加不进来了，因为这个if判断执行过一次就不会再成立了，这个设计的初衷不知道是不是source和sink只能有一个，因为我这边发现打开sink后，source是有问题的，看起来是不能使用了，目前是暂时把这个判断拿掉了，因为我们source端没什么要求应该是用不上，但是sink端是一定需要的

原文链接：<https://blog.csdn.net/marysun85/article/details/93877999>

export LC\_ALL=C

Page：这个子状态就是我们通常称为的连接(寻呼)，进行连接/激活对应的slave的操作我们就称为page。它是指：发起连接的设备（主设备）知道要连接设备的地址。所以可以直接传呼。（想想传呼机，要知道号码才行）。

Page scan：这个子状态是和page对应的，它就是等待被page的slave所处的状态，换句话说，若想被page到，我们就要处于page scan的状态。

inquiry：这就是我们通常所说的扫描状态，这个状态的设备就是去扫描周围的设备。它是不知道周围有什么设备，要去查询（调查），类似于广播（吆喝）。处于Inquiry Scan的设备可以回应这个查询。再经过必要的协商之后，它们就可以进行连接了。

此处需要说明的是：Inquiry之后，不需要进入Page就可以连接上设备。

inquiry scan：这就是我们通常看到的可被发现的设备。体现在上层就是我们在android系统中点击设备可被周围什么发现，那设备就处于这样的状态。

slave response：这个就是在page的过程中，slave收到了master的page msg，它会回应对应的page response msg，同时自己就进入到了slave response的状态。

master response：master收到slave response的msg后，他就会进入到master response的状态，同时他会发送一个FHS的packet。

inquiry response：就是在inquiry scan的设备在收到inquiry的msg后，就会发送inquiry response的msg，在这之后它就会进入到了inquiry response的状态了。

以上的各种状态可以总结到下面的寻呼过程中：即寻呼过程按照如下步骤进行：

1) 一个设备（源）寻呼另外一个设备（目的），此时处于寻呼状态。（Page state）

2) 目的设备接收到该寻呼，此时处于寻呼扫描状态。（Page Scan state）

3) 目的设备发送对源设备的回复，此时处于子设备响应状态。（Slave Response state）

4) 源设备发送FHS包到目的设备，此时处于主设备响应状态。（Master Response state。）

5) 目的设备发送第二个回复给源设备，此时处于子设备响应状态（Slave Response state。）

6) 目的和源设备切换并采用源信道的参数，此时处于主设备响应状态和子设备响应状态。

版权声明：本文为CSDN博主「zhaoshuzhaoshu」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/zhaoshuzhaoshu/article/details/51331741>

概述

\*　MTU: 最大传输单元（MAXIMUM TRANSMISSION UNIT） ， 指在一个PDU （Protocol Data Unit： 协议数据单元，在一个传输单元中的有效传输数

据）能够传输的最大数据量（多少字节可以一次性传输到对方）。

\* MTU 交换是为了在主从双方设置一个PDU中最大能够交换的数据量，通过MTU的交换和双方确认（注意这个MTU是不可以协商的，只是通知对方，双方在知道对方的极限后会选择一个较小的值作为以后的MTU，比如说，主设备发出一个150个字节的MTU请求，但是从设备回应MTU是23字节，那么今后双方要以较小的值23字节作为以后的MTU），主从双方约定每次在做数据传输时不超过这个最大数据单元

MTU交换通常发生在主从双方建立连接关系后（参见"一分钟读懂低功耗蓝牙连接数据包"）

做个对比就可以知道BLE MTU 比较小（不过新的BLE 标准MTU 已经大幅提升，详见即将发表在VIEWTOOL BBS上的后续文章）。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*“\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

以太网：1500

IEEE 802.3/802.2:　1492

X.25: 576

BLE: 23 => 这就是为什么WIFI 可以用于传输视频，传统蓝牙(BT)可以传输音频，而低功耗蓝牙（BTLE 或者BLE）只能够传输控制数据的原因了

插上U盘，打开命令行，使用命令df -h查看所有的挂载的硬盘U盘的信息（权限不足使用sudo）：

在这里插入图片描述https://csdnimg.cn/release/blogv2/dist/pc/themesSkin/skin-shirtblue/images/bg.png?v20200831

显示最后一条是我的U盘信息，U盘的文件系统是/dev/sdb4，挂载点是/media/charleswu/Ubuntu 16.0

使用命令卸载U盘（不要拔掉U盘）

sudo umount /media/charleswu/Ubuntu 16.0

1

修复文件系统故障

sudo dosfsck -v -a /dev/sdb4

1

此处的/dev/sdb4为我的U盘的文件系统，替换成你的即可。

这条指令的执行时间不定，可能很快完成，也可能花费几分钟，请耐心等待。

然后拔掉U盘，重新插一下看看是否已经修复。

————————————————

版权声明：本文为CSDN博主「CharlesWu123」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/m0_38007695/article/details/88534956>

 grep -nr NEON .

查找当前目录下包含关键字“NEON”的所有文件，并列出行号。

Termination 终止

现在手机校准和测试有两种模式：   
【1】signaling test mode   
【2】Non Signaling test mode;   
你知道这两种模式有何区别吗？  
signaling mode:信令模式，就是用CMU200或8960模拟基站，和手机建立起链接,仪表发出各种信令，手机此时相当于联上了网络。  
1）手机此时既要发射信号，又要接收来自仪表的各种信令，这种方式一般用于Final TEST。  
2）信令模式某种程度上可以说完全模拟了手机和基站注册、寻呼、以及MOC、MOT的呼叫过程以及通话过程。  
3）当然，这种和仪器建立的通信还是和真正的通信是有区别的，为了测试的需要，比如测试BER，有时需要手机环回（loop back）基站发出的数据，这在实际通信中是没有的。  
Non Signaling test mode：非信令模式，仪表此时并不发出信令去控制手机，或者仪表只是发射，手机接收；或者仪表只是接收，手机发射。当然为了支持这个模式，手机需要进入一种特殊的模式，能够支持只发，或只收。  
非信令模式主要用来手机校准（Calibration）或手机研发中故障定位等。比如排除接收的故障，单独看发射是不是好的。  
信令模式和非信令模式都可以用来测试RF指标，比如Phase Error、Frequency Error、Power/Time Template、Spectrum等，但是非信令模式不能测试BER，因为不能环回数据。

Permission denied 没有权限

基础 shell 脚本 [www.cnblogs.com/zhang-jun-jie/p/9266858.html](http://www.cnblogs.com/zhang-jun-jie/p/9266858.html)

<https://www.jianshu.com/p/a5f626d09567>

# BLE入门 20 蓝牙5速率分析

linux中的strip命令简介

一、下面是man strip获得到的信息，简单的说就是给文件脱掉外衣，具体就是从特定文件中剥掉一些符号信息和调试信息，使文件变小。

NAME

strip - Discard symbols from object files.

SYNOPSIS

strip [-F bfdname |--target=bfdname]

[-I bfdname |--input-target=bfdname]

[-O bfdname |--output-target=bfdname]

[-s|--strip-all]

[-S|-g|-d|--strip-debug]

[-K symbolname |--keep-symbol=symbolname]

[-N symbolname |--strip-symbol=symbolname]

[-w|--wildcard]

[-x|--discard-all] [-X |--discard-locals]

[-R sectionname |--remove-section=sectionname]

[-o file] [-p|--preserve-dates]

[--keep-file-symbols]

[--only-keep-debug]

[-v |--verbose] [-V|--version]

[--help] [--info]

objfile...

DESCRIPTION

GNU strip discards all symbols from object files objfile. The list of object files may include archives. At

least one object file must be given.

strip modifies the files named in its argument, rather than writing modified copies under different names.

2、实际测试。

I、测试代码strip.c:

#include <stdio.h>

void main()

{

printf("strip\n");

}

II、在liunx下执行gcc strip.c，会生成a.out。执行ls -l a.out看一下文件大小，用file命令来查看文件

基本信息的，用nm命令来列出一个目标文件中的各种符号。

III、执行strip a.out，然后执行ls -l a.out看一下文件大小，用file命令来查看文件基本信息

的，用nm命令 来列出一个目标文件中的各种符号。很明显，文件已经变小了，已经没有相关的符号信息了。

IV、执行./a.out ,说明strip后文件变小了，但还是可以正常的运行。

三、小结.

综上所述， strip命令用于脱掉文件的衣服， 文件会变小， 其中的符号信息会失去。 那这个strip有什么

用呢？ 很有用的！ 原来的a.out比较大， 可以执行。 在strip之后， 文件变小了， 也是可以执行， 这就节省了很多空间.

其实， strip不仅仅可以针对可执行文件， 还能针对目标文件和动态库等.

在实际的开发中， 经常需要对动态库.so进行strip操作， 减少占地空间。 而在调试的时候（比如用addr2line）， 就需要符号了。 因此， 通常的做法是： strip前的库用来调试， strip后的库用来实际发布， 他们两者有对应关系。 一旦发布的strip后的库出了问题， 就可以找对应的未strip的库来定位。

例如某个动态库strip前是50M左右， strip后是20M左右， 可见， 脱脱衣服还是有明显好处的。

我们在调试过程中， 经常涉及到上传库， 库太大时， 很耗费传输时间， 所以还是先用strip来处理一下比较好。

————————————————

版权声明：本文为CSDN博主「海月汐辰」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/qq_37858386/article/details/78559490>

2016年蓝牙协议5.0版本中，低功耗蓝牙新增了编码物理层以及两种编码方案(s＝2，s＝8)，对应信息传输速率分别为500kb/s以及125kb/s。编码物理层增强了蓝牙信号传输的稳定性，在不提高发送功率的前提下低功耗蓝牙信号传输距离最高可以提升4倍，大大拓展了低功耗蓝牙在物联网的应用领域以及发展前景。

针对低功耗蓝牙的s＝8编码物理层，接收机利用维特比联合解调解码，将会极大地提高接收机的灵敏度，是一个非常具有实际意义的问题。

低功耗蓝牙 5.0 协议中，定义两种调制方式，强制的调制方式(1 Msym/s 调制)使用一个shaped, binary FM去减小传输的复杂度。符号速率是 1 Msym/s，另一种可选的调制方案( 2 Msym/s) 是类似的，但使用了 2 Msym/s的符号速率。

在 1 Msym/s 调制方式支持两种 PHY：

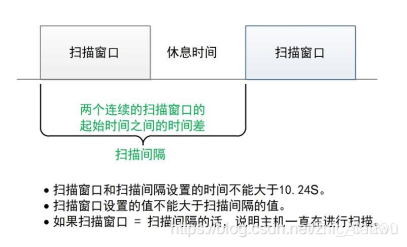
(1) LE Uncode PHY，传输速率就为 1 Mb/s。

(2) LE Coded PHY，信息数据编码方式，传输速率为 125 kb/s或者500 kb/s。

在 2 Msym/s 调制方式下支持单一 PHY：

LE 2MPHY，信息数据不编码，传输速率就为 2 Mb/s。

相对于早先的蓝牙协议标准来说，LE Coded PHY 和 LE 2M PHY 是新增加的。其中 LE Coded PHY 中引入了 FEC 编解码以及 Pattern Mapper 和 Demapper 部分, 以降低传输速率，增加接收机复杂度为代价，使得蓝牙设备能更远距离的传输信号，从而更好地适应物联网应用。



BLE扫描

连接间隔

       就是两个连接事件之间的时间间隔，连接间隔以1.25为单位，连接间隔的值为6(7.5ms)~3200(4s).不同的应用可能要求不同的时间间隔。长的时间间隔的优势是显著地节省功耗，因为设备可以在连接事件之间有较长时间的休眠，坏处是当设备有应用数据需要发送时，必须要等到下一个连接事件；短的时间间隔优势是两设备连接频发，可以更快地收发数据，不利之处是设备因连接事件的到来而被频繁的唤醒，会有较多功耗.

       在实际应用的时候，有时候需要发送数据快一点，这个时候我们可以把连接间隔改小一点，这样发送数据快，但是功耗大，当把数据发完的时候，可把连接间隔改小一点，这样可以降低功耗，达到一个均衡.比如穿戴设备连接上的时候，可能需要快速把本地数据发给手机，这个时候就可以把连接间隔设置小点。传送完毕后间隔设置大点。这样功耗就不会太大。

        这个间隔 是指成功连接后 的周期性通讯时间，主机会根据使用情况 在 这个取值范围内 选择合适的 间隔时间，这个具体值是不可控的，所以需要划定一个范围使得通讯响应在自己的可控范围之内。这个具体值会影响到下一次通讯数据包的响应时间，需要根据自己的情况来调整这个范围达到 既省电又匹配程序响应速度 的目的。

**比如：**[大数据](http://lib.csdn.net/base/20" \o "Hadoop知识库" \t "_blank)**传递时：通讯数据包是连续传递的，主机会选择min值来进行通讯。  
 无数据传递时：通讯是空闲状态，主机会选择max值来定期询问从机状态，以保持连接不中断。（在空闲时，由于使用max的值作为通讯周期，会影响到程序的下一个命令的发送时间）**

2

SLAVE\_LATENCY: （从机所存）

Slaver设备没有数据要发时，跳过一定数目的ConnectionEvent的值，Rang：0-499.

跳过的interval个数，设置为0则每次interval都连接。数字约小通信速度越快，但是功耗越高

注意：这个参数更重要的应用 ：比如在距离远或者干扰大的时候 无数据通信的时候。把这个值设大可以减少掉线概率。

比如防丢器：放在口袋里面假如这个参数设置为0 表示规定时间内必须响应从机，不然就以为是蓝牙断开了。假如设置为5。假如信号不好的时候，即使中间丢了4个，只接受了1个就表示连接了。因为他会跳过其中4个 ，保证了不掉线的概率

3.

TIMEOUT:（连接超时）

超时时间，就是两个设备在连接的这段时间没有发生通讯而导致连接自动断开的值。Range（10ms—–32s）

连接超时时间，用在信号不太好的情况下, 给对方一点时间。超过这个时间通信就建立失败

Ap6398sv 是43752

./brcm\_patchram\_plus --enable\_hci --no2bytes --tosleep 200000 --baudrate 115200 --patchram BCM4362A2\_001.003.006.1005.1011.hcd /dev/ttyUSB0 -d &

Rk 板子 bluez extended adv random 修改 0x03变0x01

diff --git a/net/bluetooth/hci\_request.c b/net/bluetooth/hci\_request.c

index 33dc78c24b73..6194fdcbad86 100644

--- a/net/bluetooth/hci\_request.c

+++ b/net/bluetooth/hci\_request.c

@@ -2054,7 +2060,8 @@ int \_\_hci\_req\_setup\_ext\_adv\_instance(struct hci\_request \*req, u8 instance)

hci\_req\_add(req, HCI\_OP\_LE\_SET\_EXT\_ADV\_PARAMS, sizeof(cp), &cp);

- if (own\_addr\_type == ADDR\_LE\_DEV\_RANDOM &&

+ if ((own\_addr\_type == ADDR\_LE\_DEV\_RANDOM ||

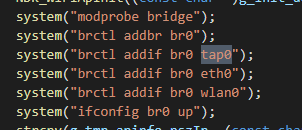
+ own\_addr\_type == ADDR\_LE\_DEV\_RANDOM\_RESOLVED) &&

bacmp(&random\_addr, BDADDR\_ANY)) {

struct hci\_cp\_le\_set\_adv\_set\_rand\_addr cp;

slave latency的意思是，slave端可以隔幾個包不回應。 slave可以進去睡覺

master latency是他會隔幾個不poll 不是。我們當client時



Ifconfig br0 物理网卡的ip up

“RFU”是“Reserved For Future Usage”的缩写，意思是“保留供将来使用”

<https://www.aoetc.com/130c6e.html>

麻烦帮忙把/etc/bluetooth/bt\_stack.conf里面的TRC\_XXX和LoggingV等级全改成6，重启后抓一下连接的logcat -v time和同步的btsnoop

Ubuntu 使用telnet

想要远程登录Ubuntu（linux）系统的电脑，可以使用ssh，也可以使用telnet。

ubuntu开启telnet服务

sudo apt-get install openbsd-inetd

sudo apt-get install telnetd

sudo /etc/init.d/openbsd-inetd restart

# 查看 telnet服务是否开启

sudo netstat -a | grep telnet

telnet登陆

# telnet address

telnet 192.168.x.x

如何退出呢？

**ctrl+],然后再按q就可以了。**

原文链接：<https://blog.csdn.net/zxc024000/article/details/81272714>

Ubuntu默认并没有安装ssh服务，如果通过ssh远程连接到Ubuntu，需要自己手动安装ssh-server。

判断是否安装ssh服务，可以通过如下命令进行：

ps -e|grep ssh

输出如下：

zht@zht-Ubuntu:~$ ps -e|grep ssh

2151 ? 00:00:00 ssh-agent

5313 ? 00:00:00 sshd

ssh-agent表示ssh-client启动，sshd表示ssh-server启动了。

如果缺少sshd，说明ssh服务没有启动或者没有安装。

安装ssh-client命令：sudo apt-get install openssh-client

安装ssh-server命令：sudo apt-get install openssh-server

安装完成以后，先启动服务：sudo /etc/init.d/ssh start

启动后，可以通过“ps -e|grep ssh”查看是否正确启动。

ssh服务默认的端口是22，可以更改端口，使用如下命令打开ssh配置文件：

sudo gedit /etc/ssh/sshd\_config

配置文件内容如下：

# Package generated configuration file

# See the sshd(8) manpage for details

# What ports, IPs and protocols we listen for

Port 22

# Package generated configuration file

# See the sshd(8) manpage for details

# What ports, IPs and protocols we listen for

修改端口号（Port）后，重启ssh服务即可生效，命令如下：

sudo /etc/init.d/ssh restart

ssh服务启动后，即可登陆，登陆命令格式为： ssh 帐号@IP地址

例如： ssh test@192.168.135.249

根据提示输入test的密码，即可远程登陆。

退出远程登陆命令：exit

原文链接：<https://blog.csdn.net/zht666/article/details/9340633>

安装ssh

sudo apt-get install openssh-server

检查是否安装成功，有 sshd，说明 ssh 安装成功

sudo ps -e | grep ssh

在这里插入图片描述

打开ssh服务

sudo service ssh start

查看ssh状态，是否已经打开，显示active说明已经打开

sudo service ssh status

关闭ssh服务：

sudo service ssh stop

复制文件

如果要将其他机器的文件复制到本机器，需要打开对方机器的ssh服务，与上述步骤相同

复制命令格式

scp -r -v username@IP:文件路径或者文件夹路径 PATH

例如

将jiang的opencv\_3.4.1文件夹下的内容递归复制到我的机器下的./Desktop/目录下，-r是递归 -v是打印信息。

scp -r -v jiang@10.5.91.160:./Documents/opencv/opencv\_3.4.1 ./Desktop/

原文链接：<https://blog.csdn.net/weixin_38907150/article/details/102925523>

在linux下一般用scp这个命令来通过[ssh](https://so.csdn.net/so/search?q=ssh" \t "_blank)协议传输文件。  
服务器需安装SSH服务器端[openssh](https://so.csdn.net/so/search?q=openssh" \t "_blank)-server，客户端需安装SSH客户端openssh-client  
ssh-server：sudo apt-get install openssh-server  
ssh-client：sudo apt-get install openssh-client  
ssh-server配置文件位于/etc/ssh/sshd\_config，在这里可以定义SSH的服务端口，默认端口是22，你可以自己定义成其他端口号，如2222。  
注：  
scp指定端口是大写P，  
ssh指定端口是小写p。

1、从服务器上下载文件  
scp username@serverip:/path/filename /var/www/local\_dir（本地目录）  
例如，

scp john@192.168.1.100:~/Desktop/MHN\_error\_solve ./

ssh默认是22端口，若出现错误ssh: connect to host 192.168.1.100 port 22: Connection refused，说明服务器端的ssh端口不是22了，这时需指定端口了：

scp -P2222 john@192.168.1.100:~/Desktop/MHN\_error\_solve ./

2、上传本地文件到服务器  
scp /path/filename username@serverip:/path

例如，

scp -P2222 test.txt john@192.168.1.100:~/

3、从服务器下载整个目录  
scp -r username@serverip:/var/www/remote\_dir/（远程目录） /var/www/local\_dir（本地目录）  
例如，

scp -P2222 -r john@192.168.1.100:~/mhn ./mhn\_copy

4、上传目录到服务器  
scp -r local\_dir username@serverip:remote\_dir  
例如，

scp -P2222 -r Study john@192.168.1.100:~/Study\_copy

查看服务器：

注：目标服务器要开启写入权限。

参考链接：  
1. SSH的详细安装步骤及登录提示:connection refused的解决办法  
blog.c[sdn](https://so.csdn.net/so/search?q=sdn" \t "_blank).net/jayxujia123/article/details/34125791  
2. sshscp 连接远程ssh非22端口的服务器方法  
https://wenku.baidu.com/view/965c281da76e58fafab003a2.html  
3. 利用ssh传输文件  
www.cnblogs.com/jiangyao/archive/2011/01/26/1945570.html  
  
原文链接：<https://blog.csdn.net/jackghq/article/details/64124062>

2.1连接间隔（Connection interval）

  连接事件被一个个连接间隔分开。从主设备发送数据包开始，每个连接事件可以持续进行，直至主设备或从设备停止响应。在连接事件之外，主从设备之间不发送任何数据，设置时间必须为1.25ms的整数倍。

2.2从设备延迟(Slave Latency)

  要确定从设备和主设备的实际交互间隔，需要用到从设备延迟这一参数，它实际上是一个连接间隔的倍数（即连接事件计数器），代表从设备在必须侦听之前可以忽略多少个连接事件。应当指出，从设备延迟必须短于监控超时时间,至少给从设备留出6次侦听的机会。

2.3超时时间（Supervision Timeout)

  为了让从设备收到数据包，主设备必须为其提供足够的机会 。不管从设备延迟是多少，都应该保证至少6次数据发送机会。因此超时时间应 >= 连接间隔 x （从设备延迟 + 1）x 6。

2.4连接参数对通信速率和低功耗的影响

连接间隔设定越小，主从设备间通信次数多，数据吞吐量更高，通信速率更高，但功耗更大。

从设备延迟设定越小，每次连接事件回应数据次数越频繁，功耗上升，但通信速率会更高。

原文链接：https://blog.csdn.net/qq\_36749906/article/details/107730890

Ble slave latency

https://zhuanlan.zhihu.com/p/347986981

<https://blog.csdn.net/q1070292175/article/details/87892604>

Supervision Timeout > （1 +slaveLatency）\* （connectionInterval）\* 2

https://blog.csdn.net/zzfenglin/article/details/51304084?spm=1001.2101.3001.6650.5&utm\_medium=distribute.pc\_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7Edefault-5.no\_search\_link&depth\_1-utm\_source=distribute.pc\_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7Edefault-5.no\_search\_link&utm\_relevant\_index=7

deb是debian linus的安装格式，跟red hat的rpm非常相似，最基本的安装命令是：dpkg -i file.deb

dpkg 是Debian Package的简写，是为Debian 专门开发的套件管理系统，方便软件的安装、更新及移除。所有源自Debian的Linux发行版都使用dpkg，例如Ubuntu、Knoppix 等。

以下是一些 Dpkg 的普通用法：

1、dpkg -i <package.deb>

安装一个 Debian 软件包，如你手动下载的文件。

2、dpkg -c <package.deb>

列出 <package.deb> 的内容。

3、dpkg -I <package.deb>

从 <package.deb> 中提取包裹信息。

4、dpkg -r <package>

移除一个已安装的包裹。

5、dpkg -P <package>

完全清除一个已安装的包裹。和 remove 不同的是，remove 只是删掉数据和可执行文件，purge 另外还删除所有的配制文件。

6、dpkg -L <package>

列出 <package> 安装的所有文件清单。同时请看 dpkg -c 来检查一个 .deb 文件的内容。

7、dpkg -s <package>

显示已安装包裹的信息。同时请看 apt-cache 显示 Debian 存档中的包裹信息，以及 dpkg -I 来显示从一个 .deb 文件中提取的包裹信息。

8、dpkg-reconfigure <package>

重新配制一个已经安装的包裹，如果它使用的是 debconf (debconf 为包裹安装提供了一个统一的配制界面)。

原文链接：<https://blog.csdn.net/kevinhg/article/details/5934462>

在Ubuntu下安装deb包需要使用dpkg命令.  
Dpkg 的普通用法：

1、sudo dpkg -i <package.deb>

安装一个 Debian 软件包，如你手动下载的文件。

2、sudo dpkg -c <package.deb>

列出 <package.deb> 的内容。

3、sudo dpkg -I <package.deb>

从 <package.deb> 中提取包裹信息。

4、sudo dpkg -r <package>

移除一个已安装的包裹。

5、sudo dpkg -P <package>

完全清除一个已安装的包裹。和 remove 不同的是，remove 只是删掉数据和可执行文件，purge 另外还删除所有的配制文件。

6、sudo dpkg -L <package>

列出 <package> 安装的所有文件清单。同时请看 dpkg -c 来检查一个 .deb 文件的内容。

7、sudo dpkg -s <package>

显示已安装包裹的信息。同时请看 apt-cache 显示 Debian 存档中的包裹信息，以及 dpkg -I 来显示从一个 .deb 文件中提取的包裹信息。

8、sudo dpkg-reconfigure <package>

重新配制一个已经安装的包裹，如果它使用的是 debconf (debconf 为包裹安装提供了一个统一的配制界面)。

如果安装过程中出现问题,可以先使用命令:  
sudo apt-get update  
更新后再执行上面的命令.

在C语言中，或更精确地说成C标准函数库中表示文件结束符。

1.在while循环中以EOF作为文件结束标志，这种以EOF作为文件结束标志的文件，必须是文本文件。在文本文件中，数据都是以字符的ASCII代码值的形式存放。我们知道，ASCII代码值的范围是0~127，不可能出现-1，因此可以用EOF作为文件结束标志。

2.在Linux系统之中，EOF根本不是一个字符，而是当系统读取到文件结尾，所返回的一个信号值（也就是-1）。至于系统怎么知道文件的结尾，资料上说是通过比较文件的长度。

3.一般C在读取数据时，都是按流模式进行数据读操作，这里的流可以是文件，也可以是标准输入。EOF可以表示文件结尾，还可以表示标准输入的结尾。但是，标准输入与文件不一样，无法事先知道输入的长度，必须手动输入一个字符，表示到达EOF。

<https://zhuanlan.zhihu.com/p/262866442>

蓝牙耳机的编码格式

今天编程遇到了SIGSEGV错误，比较困惑，所以找了些资料，总结一下：

(1)官方说法是：

SIGSEGV --- Segment Fault. The possible [cas](https://so.csdn.net/so/search?q=cas" \t "_blank)es of your encountering this error are:

1.[buffer](https://so.csdn.net/so/search?q=buffer" \t "_blank) overflow --- usually caused by a pointer reference out of range.

2.stack over[flow](https://so.csdn.net/so/search?q=flow" \t "_blank) --- please keep in mind that the default stack size is 8192K.

3.illegal file access --- file operations are forbidden on our judge system.

(2)SIGBUS与SIGSEGV信号的一般区别如下:

1) SIGBUS(Bus error)意味着指针所对应的地址是有效地址，但总线不能正常使用该指针。通常是未对齐的数据访问所致。

2) SIGSEGV(Segment fault)意味着指针所对应的地址是无效地址，没有物理内存对应该地址。

SEGV\_MAPERR, 地址没有映射到对象，可能的原因是dangling pointer或者overflow，

比如

1. ptr1和ptr2指向同一段内存，但是某个线程某个时刻用ptr1将内存delete了，如果因为错误的设计或者假设导致认为ptr2还是指向合法的内存，使用时就会出错；

2. 某个数组有1个元素，但是传入的数组大小却是2，如果我们要用2作为长度来遍历这个数组，那当访问第二个元素时就会出错；

SEGV\_ACCERR, 对映射的对象没有权限

(3)Linux的mmap(2)手册页

--------------------------------------------------------------------------

使用映射可能涉及到如下信号

SIGSEGV

试图对只读映射区域进行写操作

SIGBUS

试图访问一块无文件内容对应的内存区域，比如超过文件尾的内存区域，或者以前有文件内容对应，现在为另一进程截断过的内存区域。

--------------------------------------------------------------------------

弄清楚错误以后，就要查找产生错误的根源，一般我用以下两种方法：

(1)gcc -g 编译

ulimit -c 20000

之后运行程序，等core dump

最后gdb -c core

来查调用栈

(2)使用strace execfile，运行程序，出错时会显示那个系统调用错

关于SIGSEGV错误及处理方法

http://blog.csdn.net/brace/article/details/1102422 今天编程遇到了SIGSEGV错误,比较困惑,所以找了些资料,总结一下: (1)官方说法是: SIG ...

js,java,浮点数运算错误及应对方法

js,java浮点数运算错误及应对方法 一,浮点数为什么会有运算错误 IEEE 754 标准规定了计算机程序设计环境中的二进制和十进制的浮点数自述的交换.算术格式以及方法. 现有存储介质都是2进制.2 ...

Linux系统Vsftp 传文件出现 553 Could Not Create File错误的解决方法

解决方法: 登录出现了这个错误提示:553 Could not create file SELinux设置如下 查看SELinux设置 ［root@localhost ~］# getsebool -a ...

[转]权限问题导致Nginx 403 Forbidden错误的解决方法

权限问题导致Nginx 403 Forbidden错误的解决方法 投稿:junjie 字体:[增加 减小] 类型:转载 时间:2014-08-22 这篇文章主要介绍了权限问题导致Nginx 403 F ...

今天早上刚刚碰到的一个问题oracle数据归档已满，只能进行内部连接,ORA-00257 archiver error. 错误的处理方法

archive log 日志已满ORA-00257: archiver error. Connect internal only, until freed 错误的处理方法1. 用sys用户登录  sq ...

ueditor上传大容量视频报http请求错误的解决方法

故障现象: 当使用百度编辑器ueditor上传大容量视频或大容量图片的时候,编辑器报"http请求错误"的解决方法详解: 原因分析: 目前很多CMS整合了百度的ueditor编辑器 ...

WCF项目中出现常见错误的解决方法：基础连接已经关闭: 连接被意外关闭

在我们开发WCF项目的时候,常常会碰到一些莫名其妙的错误,有时候如果根据它的错误提示信息,一般很难定位到具体的问题所在,而由于WCF服务的特殊性,调试起来也不是那么方便,因此往往会花费不少时间来进行跟 ...

随机推荐

Android性能优化之利用Rxlifecycle解决RxJava内存泄漏

前言: 其实RxJava引起的内存泄漏是我无意中发现了,本来是想了解Retrofit与RxJava相结合中是如何通过适配器模式解决的,结果却发现了RxJava是会引起内存泄漏的,所有想着查找一下资料学 ...

PHP 错误与异常 笔记与总结(7)将错误日志以邮件方式发送

当系统发生了很严重的问题,需要立刻发送给管理员.可以通过 error\_log() 将错误以邮件形式发送到邮箱. 在 php.ini 中设置: sendmail\_from = 472323087@qq. ...

mount源码分析 【转】

转自:http://blog.chinaunix.net/uid-10769062-id-3230811.html Busybox- 在util-linux/mount.c的line:1609行首先映 ...

在Windows 10上安装Oracle 11g数据库出现的问题及解决

在Windows 10上安装Oracle 11g数据库,并且很多次出现过:当安装的进度条进行到快要结束的时候弹出一个提示框.如下: [Java(TM)2 Platform Standard Editi ...

LinearLayout和RelativeLayout 区别

LinearLayout和RelativeLayout转自:http://blog.csdn.net/w176236767/article/details/6605848共有属性:java代码中通过b ...

Code for the Homework2 改进

1. 实现了到指定点各个关节的转角计算(多解性),并且所求解满足各个关节的最大角和最小角的限制条件. 2. 对方向向量进行了单位化,保证任意大小的向量都行 #include

只要把鼠标移上Div方框，方框就自动顺时针旋转

这是一个CSS3特效,IE下看不到效果.一个Div方框,在CSS3代码的作用下,只要把鼠标移上Div方框,方框就自动顺时针旋转.代码量不大,甚至有些简单,作为一个基础的CSS3实例,我想还是比较不错的 ...

python初识1

作者:武沛齐 出处:http://www.cnblogs.com/wupeiqi/ 本文版权归作者和博客园共有,欢迎转载,但未经作者同意必须保留此段声明,且在文章页面明显位置给出原文连接. 安装Pyt ...

CSS 专业技巧

使用CSS复位 CSS复位可以在不同的浏览器上保持一致的样式风格.您可以使用CSS reset 库Normalize等,也可以使用一个更简化的复位方法: \* { box-sizing: border- ...

linux-docker下安装禅道全部

友情提示:按照步骤走,99%的人会安装成功,1%的人可以咨询度娘 64位电脑安装禅道,满足发送邮件功能 第一步: docker ps 查看docker中的容器是否有禅道(docker ps -a    ...

<http://cn.voidcc.com/question/p-egvcqmbk-bem.html>

Linux grep -r /// -R 的区别

<https://blog.csdn.net/daiyunping0402/article/details/88421058>

Sniff mode 讲解

<http://blog.chinaunix.net/uid-21411227-id-5791081.html>

A2dp相关资料

<https://blog.csdn.net/ylangeia/category_8664886.html>

蓝牙

<https://blog.csdn.net/zwc1725/article/details/52786644> ble报文

<https://blog.csdn.net/dahailinan/article/details/109243970?utm_medium=distribute.pc_relevant.none-task-blog-2~default~baidujs_title~default-0.pc_relevant_paycolumn_v2&spm=1001.2101.3001.4242.1&utm_relevant_index=3>

一些基础知识吧 不太好的

Mail.ampak.com.tw

Kevin.xiong@ampak.com.tw

<https://www.cnblogs.com/blogs-of-lxl/p/7010061.html>

<https://codeantenna.com/a/Z1f9BD1bXv>

<https://evilpan.com/2021/07/11/android-bt/>

<https://www.daimajiaoliu.com/daima/4796a0662100404>

<http://www.noobyard.com/article/p-oqrezvsv-ma.html>

安卓蓝牙

https://blog.csdn.net/weixin\_33743248/article/details/85747910?spm=1001.2101.3001.6650.3&utm\_medium=distribute.pc\_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7Edefault-3.pc\_relevant\_aa&depth\_1-utm\_source=distribute.pc\_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7Edefault-3.pc\_relevant\_aa&utm\_relevant\_index=6

有道笔记分享链接：[http://note.youdao.com/share/?id=950d00cefa9b7fd3c559eec349805b24&type=note](http://note.youdao.com/share/?id=950d00cefa9b7fd3c559eec349805b24&type=note" \t "_blank)

蓝牙技术

用我的account.  別提供其他人嘍J

帳號: luke.chen

密碼: 12007222

Cod

<https://www.ampedrftech.com/cod.htm>

Primary service

<https://cloud.tencent.com/developer/article/1689691>

int app\_read\_local\_device\_addr(BD\_ADDR \* bdaddr)

{

tBSA\_TM\_HCI\_CMD hci\_cmd;

tBSA\_STATUS bsa\_status;

UINT8 i;

BSA\_TmHciInit(&hci\_cmd);

hci\_cmd.opcode = HCI\_READ\_BD\_ADDR;

hci\_cmd.length = 0;

hci\_cmd.no\_opcode\_swap = TRUE;

BSA\_TmHciCmd(&hci\_cmd);

if(bsa\_status != BSA\_SUCCESS)

{

APP\_ERROR1("BSA\_TmReadVersion failed status:%d", bsa\_status);

return(-1);

}

memcpy(bdaddr,hci\_cmd.data,hci\_cmd.length);

#if 1

printf("HCI Server status:%d\n", hci\_cmd.status);

printf("opcode : 0x%04X\n", hci\_cmd.opcode);

printf("length : %02X\n", hci\_cmd.length & 0xff);

printf("status : %02X\n", hci\_cmd.status);

printf("Data : \n");

for(i=0; i<hci\_cmd.length; i++)

{

printf("%02X ",hci\_cmd.data[i]);

if(i!=0 && (i%15) == 0)

printf("\n");

}

printf("\n");

#endif

return 0;

}

Read local addr

可能有的人还没听过strstr函数，个人认为这个一个很实用的函数，strstr(str1,str2) 函数是字符串处理函数之一，位于头文件 “string.h”中。对于处理字符串的一些问题有很大的帮助。

**定义：strstr(str1,str2) 函数用于判断字符串str2是否是str1的子串。如果是，则该函数返回str2在str1中首次出现的地址；否则，返回NULL。**

源于百度百科：

<http://baike.baidu.com/link?url=ya28ba45_Sr6cFdJfNiQdHXl2jpZ_miXq-XnVOZurxXP_BEAck-q6ut3gX7X-cVe15lfkAWCv8z2dZQTi28VBa>

定义说的有点羞涩难懂。举个例子就知道了。   
比如：   
char str2 = “cdef”；   
char str1 = “abcdefgh”；   
则通过函数，将返回   
strstr(str1,str2) = cdefgh；

如果str1不包含有str2。   
char str2 = “cxef”；   
char str1 = “abcdefgh”；   
则通过函数，将返回   
strstr(str1,str2) = NULL；

PCM与I2S

<https://blog.csdn.net/newtonnl/article/details/7878260>

PCM（PCM-clock、PCM-sync、PCM-in、PCM-out）脉冲编码调制，模拟语音信号经过采样量化以及一定数据排列就是PCM了。理论上可以传输单声道，双声道立体声和多声道。是数字音频的raw data。

I2S是音频数字化后数据排列的一种格式，说传输的就是PCM，支持单声道和立体声。

SVN

<https://www.cnblogs.com/liuxiaowei/p/8297268.html>

<https://blog.csdn.net/mengdong_zy/article/details/78271247?utm_medium=distribute.pc_relevant.none-task-blog-2~default~baidujs_title~default-5.pc_relevant_default&spm=1001.2101.3001.4242.4&utm_relevant_index=8>

A2dp sink

<https://blog.csdn.net/marysun85/article/details/93877999>

Git

<https://blog.csdn.net/qq_38111015/article/details/84885809>

<https://blog.csdn.net/YJG7D314/article/details/104551896>

### 创建或解压tar.xz文件的方法

习惯了 tar czvf 或 tar xzvf 的人可能碰到 tar.xz也会想用单一命令搞定解压或压缩。其实不行 tar里面没有征对xz格式的参数比如 z是针对 gzip，j是针对 bzip2。

创建tar.xz文件：只要先 tar cvf xxx.tar xxx/ 这样创建xxx.tar文件先，然后使用 xz -z xxx.tar 来将 xxx.tar压缩成为 xxx.tar.xz

**解压tar.xz文件：先 xz -d xxx.tar.xz 将 xxx.tar.xz解压成 xxx.tar 然后，再用 tar xvf xxx.tar来解包。**

printenv PATH

本文介绍了linux下的压缩程式tar、gzip、gunzip、bzip2、bunzip2、compress 、uncompress、 zip、 unzip、rar、unrar等程式，以及如何使用它们对.tar、.gz 、.tar.gz、.tgz、.bz2、.tar.bz2、.Z、. tar.Z、.zip、.rar这10种压缩文件进行操作

Linux下最常用的打包程序就是tar了，使用tar程序打出来的包我们常称为tar包，tar包文件的命令通常都是以.tar结尾的。生成tar包后，就可以用其它的程序来进行压缩了，所以首先就来讲讲tar命令的基本用法：

tar命令的选项有很多(用man tar可以查看到)，但常用的就那么几个选项，下面来举例说明一下：

|  |  |
| --- | --- |
| 1 | # tar -cf all.tar \*.jpg |

这条命令是将所有.jpg的文件打成一个名为all.tar的包。-c是表示产生新的包 ，-f指定包的文件名。

|  |  |
| --- | --- |
| 1 | # tar -rf all.tar \*.gif |

这条命令是将所有.gif的文件增加到all.tar的包里面去。-r是表示增加文件的意思。

|  |  |
| --- | --- |
| 1 | # tar -uf all.tar logo.gif |

这条命令是更新原来tar包all.tar中logo.gif文件，-u是表示更新文件的意思。

|  |  |
| --- | --- |
| 1 | # tar -tf all.tar |

这条命令是列出all.tar包中所有文件，-t是列出文件的意思

|  |  |
| --- | --- |
| 1 | # tar -xf all.tar |

这条命令是解出all.tar包中所有文件，-x是解开的意思

以上就是tar的最基本的用法。为了方便用户在打包解包的同时可以压缩或解压文件，tar提供了一种特殊的功能。这就是tar可以在打包或解包的同时调用其它的压缩程序，比如调用gzip、bzip2等。

**1) tar调用**

gzip是GNU组织开发的一个压缩程序，.gz结尾的文件就是gzip压缩的结果。与gzip 相对的解压程序是gunzip。tar中使用-z这个参数来调用gzip。下面来举例说明一下：

|  |  |
| --- | --- |
| 1 | # tar -czf all.tar.gz \*.jpg |

这条命令是将所有.jpg的文件打成一个tar包，并且将其用gzip压缩，生成一个gzip压缩过的包，包名为all.tar.gz

|  |  |
| --- | --- |
| 1 | # tar -xzf all.tar.gz |

这条命令是将上面产生的包解开。

**2) tar调用bzip2**

bzip2是一个压缩能力更强的压缩程序，.bz2结尾的文件就是bzip2压缩的结果。

与bzip2相对的解压程序是bunzip2。tar中使用-j这个参数来调用gzip。下面来举例说明一下：

|  |  |
| --- | --- |
| 1 | # tar -cjf all.tar.bz2 \*.jpg |

这条命令是将所有.jpg的文件打成一个tar包，并且将其用bzip2压缩，生成一个bzip2压缩过的包，包名为all.tar.bz2

|  |  |
| --- | --- |
| 1 | # tar -xjf all.tar.bz2 |

这条命令是将上面产生的包解开。

**3)tar调用compress**

compress也是一个压缩程序，但是好象使用compress的人不如gzip和bzip2的人多。.Z结尾的文件就是bzip2压缩的结果。与 compress相对的解压程序是uncompress。tar中使用-Z这个参数来调用compress。下面来举例说明一下：

|  |  |
| --- | --- |
| 1 | # tar -cZf all.tar.Z \*.jpg |

这条命令是将所有.jpg的文件打成一个tar包，并且将其用compress压缩，生成一个uncompress压缩过的包，包名为all.tar.Z

|  |  |
| --- | --- |
| 1 | # tar -xZf all.tar.Z |

这条命令是将上面产生的包解开

**有了上面的知识，你应该可以解开多种压缩文件了，下面对于tar系列的压缩文件作一个小结**：

1)对于.tar结尾的文件

tar -xf all.tar

2)对于.gz结尾的文件

gzip -d all.gz

gunzip all.gz

3)对于.tgz或.tar.gz结尾的文件

tar -xzf all.tar.gz

tar -xzf all.tgz

4)对于.bz2结尾的文件

bzip2 -d all.bz2

bunzip2 all.bz2

5)对于tar.bz2结尾的文件

tar -xjf all.tar.bz2

6)对于.Z结尾的文件

uncompress all.Z

7)对于.tar.Z结尾的文件

tar -xZf all.tar.z

另外对于Window下的常见压缩文件.zip和.rar，Linux也有相应的方法来解压它们：

**1)对于.zip**

linux下提供了zip和unzip程序，zip是压缩程序，unzip是解压程序。它们的参数选项很多，这里只做简单介绍，依旧举例说明一下其用法：

|  |  |
| --- | --- |
| 1 | # zip all.zip \*.jpg |

这条命令是将所有.jpg的文件压缩成一个zip包

|  |  |
| --- | --- |
| 1 | # unzip all.zip |

这条命令是将all.zip中的所有文件解压出来

2)对于.rar

要在linux下处理.rar文件，需要安装RAR for Linux，可以从网上下载，但要记住，RAR for Linux 不是免费的；可从http://www.rarsoft.com/download.htm下载RARfor Linux 3.2.   
0，然后安装：

|  |  |
| --- | --- |
| 1  2  3 | # tar -xzpvf rarlinux-3.2.0.tar.gz  # cd rar  # make |

这样就安装好了，安装后就有了rar和unrar这两个程序，rar是压缩程序，unrar 是解压程序。它们的参数选项很多，这里只做简单介绍，依旧举例说明一下其用法：

|  |  |
| --- | --- |
| 1 | # rar a all \*.jpg |

这条命令是将所有.jpg的文件压缩成一个rar包，名为all.rar，该程序会将.rar 扩展名将自动附加到包名后。

|  |  |
| --- | --- |
| 1 | # unrar e all.rar |

这条命令是将all.rar中的所有文件解压出来

到此为至，我们已经介绍过linux下的tar、gzip、gunzip、bzip2、bunzip2、compress 、 uncompress、 zip、unzip、rar、unrar等程式，你应该已经能够使用它们对.tar 、.gz、.tar.gz、.tgz、.bz2、.tar.bz2、. Z、.tar.Z、.zip、.rar这10种压缩文   
件进行解压了，以后应该不需要为下载了一个软件而不知道如何在Linux下解开而烦恼了。而且以上方法对于Unix也基本有效。

本文介绍了linux下的压缩程式tar、gzip、gunzip、bzip2、bunzip2、compress 、uncompress、 zip、 unzip、rar、unrar等程式，以及如何使用它们对.tar、.gz 、.tar.gz、.tgz、.bz2、.tar.bz2、.Z、. tar.Z、.zip、.rar这10种压缩文件进行   
操作。

**以下补充**

tar

-c: 建立压缩档案   
-x：解压   
-t：查看内容   
-r：向压缩归档文件末尾追加文件   
-u：更新原压缩包中的文件

这五个是独立的命令，压缩解压都要用到其中一个，可以和别的命令连用但只能用其中一个。下面的参数是根据需要在压缩或解压档案时可选的。

-z：有gzip属性的   
-j：有bz2属性的   
-Z：有compress属性的   
-v：显示所有过程   
-O：将文件解开到标准输出 

下面的参数-f是必须的

-f: 使用档案名字，切记，这个参数是最后一个参数，后面只能接档案名。 

|  |  |
| --- | --- |
| 1  2 | # tar -cf all.tar \*.jpg这条命令是将所有.jpg的文件打成一个名为all.tar的包。-c是表示产生新的包，-f指定包的文件名。  # tar -rf all.tar \*.gif |

这条命令是将所有.gif的文件增加到all.tar的包里面去。-r是表示增加文件的意思。 

|  |  |
| --- | --- |
| 1 | # tar -uf all.tar logo.gif |

这条命令是更新原来tar包all.tar中logo.gif文件，-u是表示更新文件的意思。 

|  |  |
| --- | --- |
| 1 | # tar -tf all.tar |

这条命令是列出all.tar包中所有文件，-t是列出文件的意思 

|  |  |
| --- | --- |
| 1 | # tar -xf all.tar |

这条命令是解出all.tar包中所有文件，-x是解开的意思

**压缩**

tar –cvf jpg.tar \*.jpg //将目录里所有jpg文件打包成tar.jpg   
tar –czf jpg.tar.gz \*.jpg //将目录里所有jpg文件打包成jpg.tar后，并且将其用gzip压缩，生成一个gzip压缩过的包，命名为jpg.tar.gz   
tar –cjf jpg.tar.bz2 \*.jpg //将目录里所有jpg文件打包成jpg.tar后，并且将其用bzip2压缩，生成一个bzip2压缩过的包，命名为jpg.tar.bz2   
tar –cZf jpg.tar.Z \*.jpg //将目录里所有jpg文件打包成jpg.tar后，并且将其用compress压缩，生成一个umcompress压缩过的包，命名为jpg.tar.Z   
rar a jpg.rar \*.jpg //rar格式的压缩，需要先下载rar for linux   
zip jpg.zip \*.jpg //zip格式的压缩，需要先下载zip for linux

**解压**

tar –xvf file.tar //解压 tar包   
tar -xzvf file.tar.gz //解压tar.gz   
tar -xjvf file.tar.bz2 //解压 tar.bz2   
tar –xZvf file.tar.Z //解压tar.Z   
unrar e file.rar //解压rar   
unzip file.zip //解压zip

**总结**

1、\*.tar 用 tar –xvf 解压   
2、\*.gz 用 gzip -d或者gunzip 解压   
3、\*.tar.gz和\*.tgz 用 tar –xzf 解压   
4、\*.bz2 用 bzip2 -d或者用bunzip2 解压   
5、\*.tar.bz2用tar –xjf 解压   
6、\*.Z 用 uncompress 解压   
7、\*.tar.Z 用tar –xZf 解压   
8、\*.rar 用 unrar e解压   
9、\*.zip 用 unzip 解压

**Linux下解压RAR软件下载和解压.zip和.rar文件**

下载地址：http://www.rarsoft.com/download.htm（目前最新为RAR 3.71 for Linux）

以最新的为准。

对于Window下的常见压缩文件.zip和.rar，Linux也有相应的方法来解压它们：

**1:对于.zip**

linux下提供了zip和unzip程序，zip是压缩程序，unzip是解压程序。它们的参数选项很多，这里只做简单介绍，举例说明一下其用法：

|  |  |
| --- | --- |
| 1  2 | # zip all.zip \*.jpg(这条命令是将所有.jpg的文件压缩成一个zip包)  # unzip all.zip(这条命令是将all.zip中的所有文件解压出来) |

**2:对于.rar**

要在linux下处理.rar文件，需要安装RAR for Linux，可以从网上下载，但要记住，RAR for Linux不是免费的；可从http://www.rarsoft.com/download.htm下载RAR 3。60 for Linux ，然后安装其安装操作如下:

|  |  |
| --- | --- |
| 1  2  3 | # tar -xzpvf rarlinux-3.2.0.tar.gz  # cd rar  # make |

这样就安装好了，安装后就有了rar和unrar这两个程序，rar是压缩程序，unrar是解压程序。它们的参数选项很多，举例说明一下其用法

|  |  |
| --- | --- |
| 1 | # rar a all \*.jpg |

这条命令是将所有.jpg的文件压缩成一个rar包，名为all.rar，该程序会将.rar 扩展名将自动附加到包名后。

|  |  |
| --- | --- |
| 1 | # unrar e all.rar |

这条命令是将all.rar中的所有文件解压出来。

xxxx.tar.gz文件使用tar带zxvf参数，可以一次解压开。XXXX为文件名。例如：

|  |  |
| --- | --- |
| 1 | $tar zxvf xxxx.tar.gz |

**gzip -d**

语 法：gzip [-acdfhlLnNqrtvV][-S <压缩字尾字符串>][-<压缩效率>][--best/fast][文件...] 或 gzip [-acdfhlLnNqrtvV][-S <压缩字尾字符串>][-<压缩效率>][--best/fast][目录]

补充说明：gzip是个使用广泛的压缩程序，文件经它压缩过后，其名称后面会多出".gz"的扩展名。

参 数：

-a或--ascii 使用ASCII文字模式。   
-c或--stdout或--to-stdout 把压缩后的文件输出到标准输出设备，不去更动原始文件。   
-d或--decompress或----uncompress 解开压缩文件。   
-f或--force 强行压缩文件。不理会文件名称或硬连接是否存在以及该文件是否为符号连接。   
-h或--help 在线帮助。   
-l或--list 列出压缩文件的相关信息。   
-L或--license 显示版本与版权信息。   
-n或--no-name 压缩文件时，不保存原来的文件名称及时间戳记。   
-N或--name 压缩文件时，保存原来的文件名称及时间戳记。   
-q或--quiet 不显示警告信息。   
-r或--recursive 递归处理，将指定目录下的所有文件及子目录一并处理。   
-S<压缩字尾字符串>或----suffix<压缩字尾字符串> 更改压缩字尾字符串。   
-t或--test 测试压缩文件是否正确无误。   
-v或--verbose 显示指令执行过程。   
-V或--version 显示版本信息。   
-<压缩效率> 压缩效率是一个介于1－9的数值，预设值为"6"，指定愈大的数值，压缩效率就会愈高。   
--best 此参数的效果和指定"-9"参数相同。   
--fast 此参数的效果和指定"-1"参数相同。

1、\*.tar 用 tar –xvf 解压  
2、\*.gz 用 gzip -d或者gunzip 解压  
3、\*.tar.gz和\*.tgz 用 tar –xzf 解压  
4、\*.bz2 用 bzip2 -d或者用bunzip2 解压  
5、\*.tar.bz2用tar –xjf 解压  
6、\*.Z 用 uncompress 解压  
7、\*.tar.Z 用tar –xZf 解压  
8、\*.rar 用 unrar e解压  
9、\*.zip 用 unzip 解压

Linux下\*.tar.gz文件解压缩命令

**1.压缩命令：**

命令格式：tar -zcvf 压缩文件名.tar.gz 被压缩文件名

可先切换到当前目录下。压缩文件名和被压缩文件名都可加入路径。

**2.解压缩命令：**

命令格式：tar -zxvf 压缩文件名.tar.gz

解压缩后的文件只能放在当前的目录。

你可以使用 zip 命令同时处理多个文件和目录，方法是将它们逐一列出，并用空格间隔：

zip -r filename.zip file1 file2 file3 /usr/work/school 

压缩单个目录命令zip -r filename.zip filename

上面的命令把 file1、file2、 file3、以及 /usr/work/school 目录的内容（假设这个目录存在）压缩起来，然后放入 filename.zip 文件中。

<https://www.runoob.com/svn/svn-commit.html> svn

Linux nm 命令可以查看可执行文件和动态库、静态库里面的符号信息

<https://www.cnblogs.com/downey-blog/p/10477835.html>

host +C 默认的host健是右边的Ctrl，如果修改了，就使用自定义的host健+C。  
我已成功找回，哈哈哈……

转载于:https://blog.51cto.com/shujuliu/2171636

[Linux ldconfig命令](https://www.cnblogs.com/274914765qq/p/4567283.html)

**一、简介**

ldconfig是一个动态链接库管理命令，为了让动态链接库为系统所共享,还需运行动态链接库的管理命令--ldconfig。 ldconfig 命令的用途,主要是在默认搜寻目录(/lib和/usr/lib)以及动态库配置文件/etc/ld.so.conf内所列的目录下,搜索出可共享的动态 链接库(格式如前介绍,lib\*.so\*),进而创建出动态装入程序(ld.so)所需的连接和缓存文件.缓存文件默认为 /etc/ld.so.cache,此文件保存已排好序的动态链接库名字列表.

**二、使用注意事项**

[IMG_256](javascript:void(0);)

1. 在/lib和/usr/lib里面添加内容，不需修改/etc/ld.so.conf，但要执行ldconfig，否则library会找不到

2. 在上面两个目录以外添加内容，需要修改/etc/ld.so.conf，并且执行ldconfig，否则library会找不到

如安装mysql到/usr/local/mysql，mysql有一大堆library在/usr/local/mysql/lib下面，此时需要在/etc/ld.so.conf下添加/usr/local/mysql/lib，并执行ldconfig一下

3. 如果想在这两个目录以外放lib，又不想在/etc/ld.so.conf中添加内容，可以export一个全局变量LD\_LIBRARY\_PATH，然后运行程序的时候就会去这个目录中找library。一般来讲这只是一种临时的解决方案，在没有权限或临时需要的时候使用

4. ldconfig做的这些东西都与运行程序时有关，跟编译时一点关系都没有。编译的时候还是该加-L就得加，不要混淆了

5. 总之，就是不管做了什么关于library的变动后，最好都ldconfig一下，不然会出现一些意想不到的结果。不会花太多的时间，但是会省很多的事

注意：此命令只适用于broadcom 系列蓝牙芯片，例如RK903, AP6xxx 系列

1.第一种方式

# echo 1 > /sys/class/rfkill/rfkill0/state //打开蓝牙

# echo 0 > /sys/class/rfkill/rfkill0/state //关闭蓝牙

2.第二种方式

/frameworks/base/services/core/java/com/android/server/BluetoothManagerService.java

private static final String SECURE\_SETTINGS\_BLUETOOTH\_ADDRESS="bluetooth\_address";

private static final String SECURE\_SETTINGS\_BLUETOOTH\_NAME="bluetooth\_name";

# settings get secure bluetooth\_address //获取蓝牙Mac

# settings get secure bluetooth\_name //获取蓝牙名称

3.第三种方式

# service call bluetooth\_manager 6 //打开蓝牙

# service call bluetooth\_manager 8 //关闭蓝牙

//获取蓝牙mac地址

# service call bluetooth\_manager 10 //对于Android 4.4.4

# service call bluetooth\_manager 12// 适用于Android 5.0

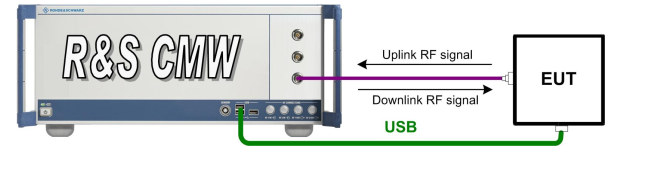
Adb 指令 <https://www.jianshu.com/p/65e80c60f656>

版权声明：本文为CSDN博主「慢慢的燃烧」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/u010164190/article/details/90294022>

BLE 信令测试：

1. 想将模块与host之间的uart断开，通过usb转串口板将模块和test device连接。
2. 通过pc使用bluetool将蓝牙fw download给模块，期间要给模块保持供电。
3. 将USB转串口板从PC端移除，接到test device，然后通过对test device操作进入信令测试模式，对BLE 蓝牙进行测试。



Ps：example CWM500

1、将host和模组端的uart断开。

2、用USB转串口板连接到模组。

3、用blue tool下载fw

4、将USB转串口板从PC端移除，接到CWM500

信令测试：由test device控制与模块双方的交互，所以都交给test device了。

BR/EDR 不用。剛剛說的，直接進DUT mode就好

BLE：LE的信令模式直接直接發LE TX/LE RX/LE test end, 這些命令都是SIG定義的，每家都一樣，所以儀器可以直接發.需要割线后，连接仪器。

（RX/TX/CTS/RTS/REG\_ON），其中模块的UART与串口转板的UART接线一一对应（即TX接TX，RX接RX，CTS接CTS，RTS接RTS），另外需要接GND，从主板接3.3V到串口板的VDD，并把VDD与REG\_ON连接上。然后用bluetool加载完Hcd后，测试BLE时，实验室的测试软件不能选com口，需要选hci来测试BLE部分

非信令测试：由我们自己发送比如1000的指令给模块，然后计算性能。

Direct Test Mode

和ble信令模式差不多。

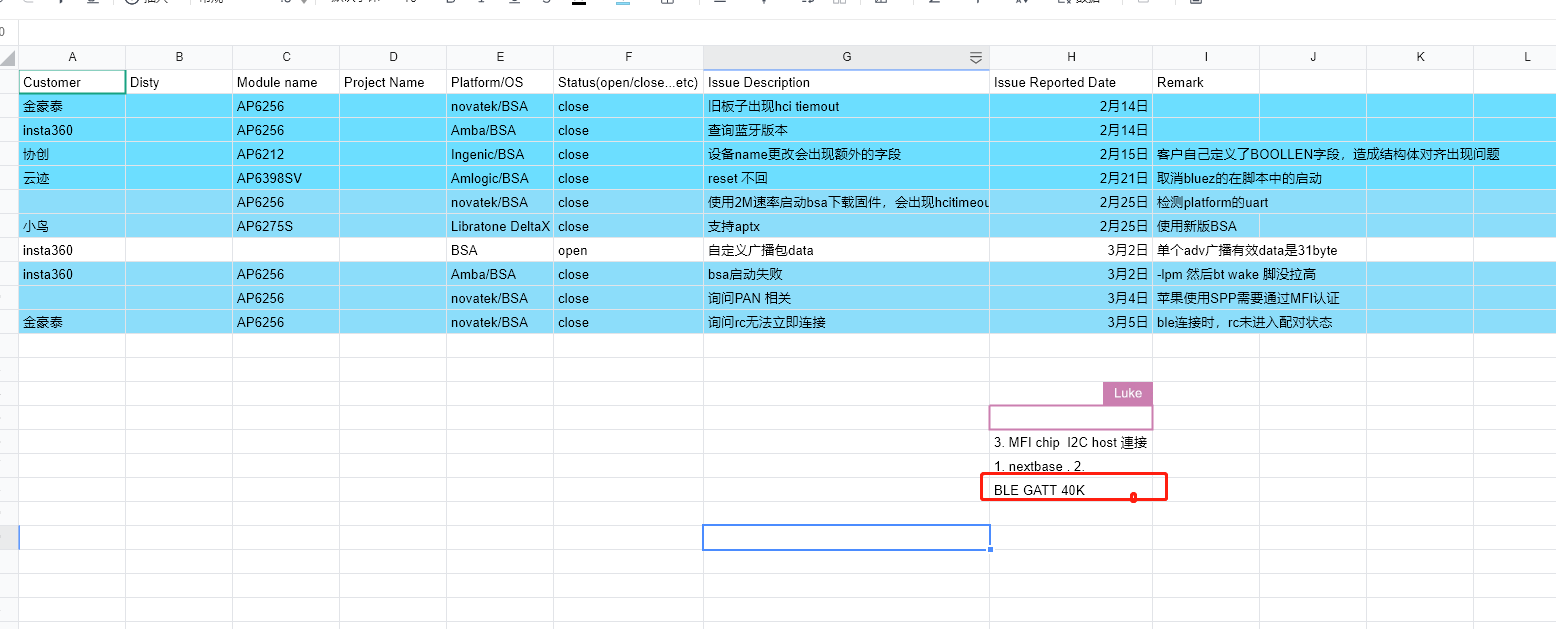
BLE 广播包

<https://blog.csdn.net/slimmm/article/details/100583655>

<https://support.apple.com/zh-tw/HT204387>

苹果 可以SUPPORT的PROFILE





IOS SPP 和 BLE 传文件相关。

#define BLE\_PRIVACY\_SPT

//#ifndef BLE\_PRIVACY\_SPT

#define BLE\_PRIVACY\_SPT TRUE

//#endif

通过对ifndef注释掉，可以验证平台的overload中是否有相同的宏定义。

ssh相关

<https://blog.csdn.net/jackghq/article/details/54974141>

<https://blog.csdn.net/future_ai/article/details/81701744>

<https://www.cnblogs.com/cookiewu/p/9664062.html>

<https://www.jb51.net/article/186472.htm>

启动ssh命令：service sshd start

停止ssh命令：service sshd stop

更改下载源头

<https://blog.csdn.net/weixin_33691817/article/details/93083043>

<https://www.cnblogs.com/yhjoker/p/12813423.html>

https://blog.csdn.net/XerxesSimon/article/details/75291927?spm=1001.2101.3001.6650.2&utm\_medium=distribute.pc\_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-2.pc\_relevant\_default&depth\_1-utm\_source=distribute.pc\_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-2.pc\_relevant\_default&utm\_relevant\_index=5

3.更新源列表

 sudo apt-get update

 sudo apt-get upgrade

Bluez <https://www.cnblogs.com/shinedream/p/14085847.html>

安卓蓝牙博客

<https://blog.csdn.net/liulei823581722/article/details/82733193>

<https://blog.csdn.net/joychanger/article/details/51434817>

Vim 看 二进制文件转  
[www.cnblogs.com/embedded-linux/p/5760745.html](http://www.cnblogs.com/embedded-linux/p/5760745.html)

Vim 可以用来查看和编辑二进制文件

vim -b egenea-base.ko   加上-b参数，以二进制打开

然后输入命令  :%!xxd -g 1  切换到十六进制模式显示

二  比较两个二进制文件，可以使用vimdiff。

vimdiff应该还是比较好用的，类似windows 平台的windiff

vim -bd base.ko base2.ko

打开后就可以在两个窗口里面显示两个文件

ctrl + W +L  把输入焦点切换到右边的窗口，激活右边的窗口后输入的命令就是针对右窗口了  
:%!xxd -g 1  切换成十六进制的一个字节的模式  
ctrl + W +H  把输入焦点切换到左边的窗口   
:%!xxd -g 1   
] + c  查找上一个不同点  
[ + c  查找下一个不同点  
 0012930: 89 df 68 77 01 00 00 e8 fc ff|  0012930: 89 df 68 78 01 00 00 e8 fc f

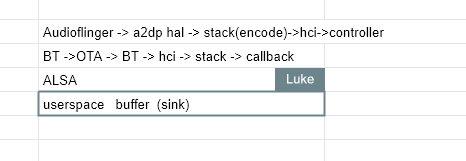
<https://backlog.com/git-tutorial/cn/intro/intro2_4.html>

<https://backlog.com/git-tutorial/cn/stepup/stepup2_8.html>

宝藏git

<https://www.code-nav.cn/>

Git

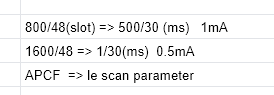


1.打开终端；   
2.sudo chown 用户名 filename；   
3.对于文件夹带锁

sudo chown 用户名 filename/-R；

关闭内核打印

echo 0 > /proc/sys/kernel/printk



lpm 功耗量測. FC27 , FC27+ le scan (bt wake asserted/deasserted. RTS status

Bluez

<https://blog.csdn.net/huohongpeng/article/details/115786953>

<https://blog.51cto.com/u_11626714/4881152>

<http://tongxinmao.com/Article/Detail/id/360>

Zip command

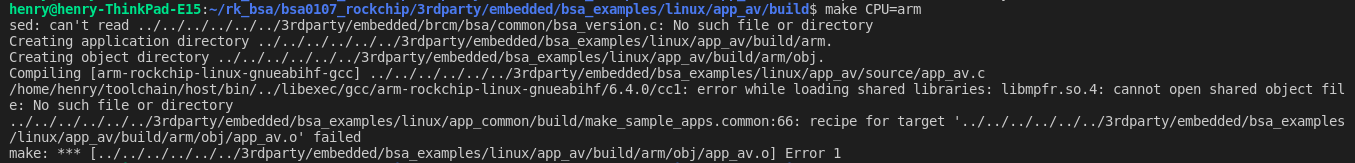


blog.csdn.net/songbinxu/article/details/80435665

Dbus 资料

<https://thebigdoc.readthedocs.io/en/latest/dbus/dbus.html>

sudo ln -s /usr/lib/x86\_64-linux-gnu/libmpfr.so.6 /usr/lib/x86\_64-linux-gnu/libmpfr.so.4



Bit rate

<https://www.lineageos.org/engineering/Bluetooth-SBC-XQ/>

电路设计以及PCB制作中，经常碰见电源符号：VCC、 VDD、VEE、VSS、VREF，他们具有什么样的关系那?

　　一、解释

　　VCC：C=circuit 表示电路的意思, 即接入电路的电压

　　VDD：D=device 表示器件的意思, 即器件内部的工作电压

　　VSS：S=series 表示公共连接的意思，通常指电路公共接地端电压

VEE：负电压供电；场效应管的源极(S)，或三极管的发射极（E）

VBAT：BAT=Battery 表示电池电压，链接电池正极

VREF：ref=reference 表示参考电压

svn账户

在目录~/.subversion/auth/svn.simple/下面的文件里

<https://blog.csdn.net/baidu_35381300/article/details/51736431>

Select函数

Repo

blog.csdn.net/nwpushuai/article/details/78778602

Curl

<https://www.cnblogs.com/linjiqin/p/5484910.html>

Repo

<https://www.jianshu.com/p/8e17af125ffb>

可以的，但需要保证是具有相同结构的结构体变量。

可以将一个结构体变量的值赋给另一个具有相同结构的结构体变量。如student1和student2都是student类型的变量，可以这样赋值：student1= student2；

假设申明了两个成员列表完全一样的两个结构体，编译器会当作两个完全不同的类型，即使他们的成员列表是一样的，令两个结构体的变量间赋值，则是非法的。

Ubuntu 降级

<https://www.modb.pro/db/148406>

<https://blog.csdn.net/qq_49814035/article/details/116035670>

<https://blog.csdn.net/qq_49814035/article/details/116035670>

https://blog.csdn.net/checknet/article/details/122257816?spm=1001.2101.3001.6650.14&utm\_medium=distribute.pc\_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7ERate-14.pc\_relevant\_default&depth\_1-utm\_source=distribute.pc\_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7ERate-14.pc\_relevant\_default&utm\_relevant\_index=18

Ubuntu 解压 压缩

<https://blog.csdn.net/weixin_39801202/article/details/110489045>

Sprintf() sscanf()

<https://blog.csdn.net/u013178472/article/details/53433939>

7z

答: sudo apt-get install p7zip

1. 解压

7zr x filename.7z

2. 压缩

7zr a -t7z -r filename.7z ~/filename/\*

各参数解析:

　　a 代表添加文件／文件夹到压缩包  
　　-t 是指定压缩类型 一般我们定为7z  
　　-r 表示递归所有的子文件夹，filename.7z 是压缩好后的压缩包名，~/filename是要压缩的目录，＊是表示该目录下所有的文件\

Ubuntu 7zip压缩使用：

安装：

apt-get install p7zip

使用：

压缩：

7zr a -r aa.7z /aa

a 代表添加文件/文件夹到压缩包

-r 是递归

解压：

7zr x aa.7z -r -o/root/

x 代表解压缩文件并按原始目录树解压（参数e也是解压缩文件，但其会将所有文件都解压到根下）

-o 是指定解压路径，-o后没有空格的，直接接目录

# iwlist/iwconfig/iw命令

<https://www.openwrt.pro/post-366.html>

1.

跳转到命令行行首:ctrl+a

2.

跳转到命令行行尾:ctrl+e

3.

删除行首到光标位置的字符:ctrl+u

4.

删除光标位置到行尾的字符:ctrl+k

Gdb

<https://blog.csdn.net/wangzhen209/article/details/111416542>

PCM

<https://zhuanlan.zhihu.com/p/212318683>

<https://www.cnblogs.com/zxc2man/p/11272435.html>

# **[PCM时序](https://www.cnblogs.com/zxc2man/p/11272435.html)**

PCM（Pulse Code Modulation），脉冲编码调制，PCM总线用于传输数字语音信号，包括4根信号线：FSYNC(同步)/PCLK(时钟)/DTX(发送)/DRX(接收)

PCM分为Master和Slave. Master提供PCLK及FSYNC。

PCM CLK是时钟信号，由master端提供，速率一般为512KHz到8.192MHz。每一个PCLK对应一位DTX/DRX的数据

PCM FSYNC为帧同步信号，其频率为PCM采样频率，一般为8KHz。一个帧同步信号的上升沿代表一帧(frame)数据的开始，一帧数据中分为若干个时隙（time slot），时隙对应channel。

DRX/DTX为数据收发，PCM数据格式有A-law/mu-law/linear三种，其中A-law/mu-law每个采样用8bit编码，即每个同步信号中需要传输8bit的数据，速率为8KHz\*8bit=64kbit/s。而linear每个采样16bit编码，每个同步信号中传输16bit数据，速率为8KHz\*16bit=128kbit/s

例如，PCLK为512KHz，FSYNC为8KHz，那么每个FSYNC能传输512/8=64位数据，假如使用16位线性PCM编码，那么一个同步信号能传输64/16=4个timeslot的数据，即4个channel。

线性PCM编码一个采样传输16bit数据，每个采样占0.125ms，20ms为一帧语音数据，包括160个采样，即160\*16bit=320Byte数据

宽带线性PCM编码（FSYNC为8k）一个采样传输32bit数据，每20ms传输640Byte数据