# Linux obex传输

obexd版本：0.4

编译系统：ubuntu14，64位

交叉编译器：arm-linux-gnueabihf-gcc

运行平台：iRTK2（包括A8和A5）

编译生成目录：PC的/usr/local/armlib

编译Obexd-0.4需要准备的库及对应版本：

openobex-1.5，bluez-4.9.5，libical-0.48，libiconv-1.13，expat-2.0.1，libffi-3.2.1，dbus1.4.1，libusb-1.0.8，glib-2.24.2。

## 1、编译obexd

编译前在终端执行export NM=nm（解决编译时syntax error near unexpected token `|'的问题）

若要改变编译生成文件的保存路径则在configure时修改--prefix=

(1) 编译libdbus-1.so (依赖libexpat.so )

交叉编译 expat-2.0.1

CC=arm-linux-gnueabihf-gcc ./configure CFLAGS=-I/usr/local/armlib/include LDFLAGS=-L/usr/local/armlib/lib --prefix=/usr/local/armlib --host=arm-linux

make && make install

交叉编译 dbus1.4.1

CC=arm-linux-gnueabihf-gcc ./configure CFLAGS=-I/usr/local/armlib/include LDFLAGS=-L/usr/local/armlib/lib --prefix=/usr/local/armlib --host=arm-linux --enable-abstract-sockets --with-x=no

make && make install

(2) 编译libglib-2.0.so (依赖libiconv.so， libiffi.so)

交叉编译 libiconv-1.13

CC=arm-linux-gnueabihf-gcc ./configure CFLAGS=-I/usr/local/armlib/include LDFLAGS=-L/usr/local/armlib/lib --prefix=/usr/local/armlib --host=arm-linux

make && make install

交叉编译 libffi-3.2.1

CC=arm-linux-gnueabihf-gcc ./configure CFLAGS=-I/usr/local/armlib/include LDFLAGS=-L/usr/local/armlib/lib --prefix=/usr/local/armlib --host=arm-linux

make && make install

交叉编译 glib-2.24.2

CC=arm-linux-gnueabihf-gcc ./configure CFLAGS=-I/usr/local/armlib/include LDFLAGS=-L/usr/local/armlib/lib --prefix=/usr/local/armlib --host=arm-linux glib\_cv\_stack\_grows=no glib\_cv\_uscore=yes ac\_cv\_func\_posix\_getpwuid\_r=yes ac\_cv\_func\_posix\_getgrgid\_r=yes glib\_cv\_have\_qsort\_r=yes --with-libiconv

make && make install

(3) 编译 libbluez.so (依赖libdbus-1.so, libglib-2.0.so)

交叉编译 bluez-4.9.5

编译前先设置PKG\_CONFIG\_LIBDIR和PKG\_CONFIG\_PATH以便找到依赖库的pc文件

export PKG\_CONFIG\_LIBDIR=/usr/local/armlib/lib/pkgconfig/

export PKG\_CONFIG\_PATH=/usr/local/armlib/lib/pkgconfig/

./configure --prefix=/usr/local/armlib --host=arm-linux --target=arm-linux CC="arm-linux-gnueabihf-gcc -L/usr/local/armlib/lib" --disable-audio --enable-test

make && make install

(4) 编译 libopenobex.so （依赖libusb.so）

交叉编译 libusb-1.0.8

CC=arm-linux-gnueabihf-gcc ./configure CFLAGS=-I/usr/local/armlib/include LDFLAGS=-L/usr/local/armlib/lib --prefix=/usr/local/armlib --host=arm-linux

make && make install

交叉编译 openobex-1.5

CC=arm-linux-gnueabihf-gcc ./configure CFLAGS=-I/usr/local/armlib/include LDFLAGS=-L/usr/local/armlib/lib --prefix=/usr/local/armlib --host=arm-linux --disable-usb --enable-apps --enable-bluetooth

出现问题：

“ checking for USB ... yes

/usr/local/armlib/lib/pkgconfig/libusb.pc configure: error: cannot check for file existence when cross compiling ”

确定/usr/local/armlib/pkgconfig已经有了libusb-1.0.pc 修改configure 找到libusb.pc 将第11311行 # { (eit 1); exit 1; }; } 修改为 {(echo 1);echo 1;};}

重新配置 make && make install

(5) 编译obexd运行文件 （依赖libbluetooth.so, libopenobex.so, libical.so）

交叉编译obex4.0

CC=arm-linux-gnueabihf-gcc ./configure CFLAGS=-I/usr/local/armlib/include LDFLAGS=-L/usr/local/armlib/lib --prefix=/usr/local/armlib --host=arm-linux

make && make isntall

## 2 、移植obexd到固件中

按照以上的编译步骤，obexd和依赖库生成在PC的 /usr/local/armlib/里面

1、把obexd拷贝到iRTK2文件系统的/mnt/platform/bin/里

2、把依赖库拷贝到iRTK2文件系统的/mnt/platform/lib/里

3、在文件系统/etc/dbus-1/system.d/bluetooth.conf文件添加<allow own="org.openobex"/>

4、在/etc/init.d/loadperipherals.sh 开头加入

export DBUS\_SESSION\_BUS\_ADDRESS="unix:path=/var/run/dbus/system\_bus\_socket"，

在wlanhw语句后添加obexd -a -f –o。

5、把蓝牙模块的波特率文件480-0048-R3.0\_921600Baud.bts改名成TIInit\_7.6.15.bts，然后复制到/mnt/platform/lib/firmware/替换掉原来的TIInit\_7.6.15.bts。这样就能把波特率改变成921600（默认的波特率为3M）。

按照以上5个步骤完成obexd移植到固件。

目前obex传输使用921600或以下的波特率并且接收的文件保存在/tmp目录下，并未发生过传输失败的情况。

## 3、obex传输速度（文件默认保存在/tmp目录）

测试距离为1米以内的范围。

（1）直接使用安卓系统自带的opp功能传输并且不运行固件的情况下：

3M 波特率传输28M的固件平均使用了3分钟50秒（127kb/s）。

900k波特率传输28M的固件平均使用了6分钟左右的时间（77kb/s）。

直接使用安卓系统自带的opp传输功能传输固件速度比较稳定。时间一般偏差不到1分钟。

（2）运行固件同时使用hi- survey连接iRTK2进行固件传输的情况下：

3M波特率的情况下传输一个28M的固件一般使用8分钟（61kb/s）或更多时间。

900k波特率情况下传输一个28M的固件一般使用14分钟（34kb/s）或更多时间。

此情况传输时间浮动比较大，偏差从不到1分钟到10分钟的情况都有。

（3）运行固件同时使用hi- survey连接iRTK2进行固件传输（关闭wifi热点）的情况下：

3M 波特率传输28M的固件平均使用了3分钟50秒（127kb/s）。

900k波特率传输28M的固件平均使用了6分钟10秒的时间（76kb/s）。

此情况下速度比较稳定，和直接使用手机进行obex传输的速度差不多。

## 4、iRTK2（A8）上obex传输的问题

（1）、obex传输文件的时候若把文件存放在/tmp（内存构成的文件系统）以外的目录则会有几率出现“unknown packet type”的错误”如图1.1。并且波特率越快出现此问题的几率越高，出现此错误之后蓝牙就会出现异常，其他设备能搜索到该蓝牙但是该蓝牙不能再正常工作。执行hciconfig –a查看蓝牙设备会出错如图1.2。

针对此问题进行过以下两个测试：

一、接收obex的文件到内存缓冲区但是不写入到文件系统中，在传输过程中同时运行一个进程不断地对一个/tmp以外的目录下的一个文本文件不断写入文本也会影响到蓝牙出现“unknown packet type”的错误。

二、使用同样的obex和库的源代码在其他嵌入式芯片（超过两个不同的芯片）和其他蓝牙测试，传输速度即使到达 170kb/s也非常稳定并没有发现此问题。

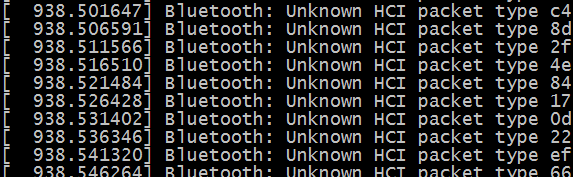


图1.1

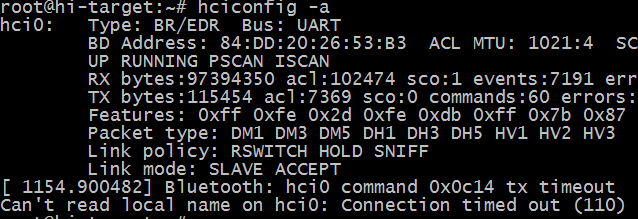


图1.2

（2）出现“unknown packet type”错误后只能执行重启系统和执行/etc/init.d/bluetooth restart重启蓝牙才能继续正常工作。但是执行reboot命令或者执行/etc/init.d/bluetooth restart重启之后会有很低的概率在之后的obex传输会连续出现“unknown packet type”的情况，断电重启就暂时没有发现这个情况。

## 5、本次测试结论：

（1）、obex传输的可用方案

一、目前obex传输使用921600或以下的波特率并且接收的文件保存在/tmp目录下，是最稳定的传输情况，平均速度约为77kb/s同时运行着固件的其他功能也并未测试出任何问题。

二、obex传输只有使用3M的波特率,接收的文件保存在/tmp目录下，传输速度约为127kb/s,

不运行固件的情况下传输并未出现问题，但是运行固件的情况下会有大约5%的概率会出现

“unknown packet type”的情况。

（2）、根据上面针对“unknown packet type”这个问题的测试可以看出此问题应该与obex的源代码和依赖库无关，极有可能和内核或蓝牙驱动相关。

（3）、在运行固件的情况下进行传输，wifi热点的进程hostapd会大大影响obex的传输速度，若把hostapd关掉，则传输速度会变稳定和不运行固件时用手机obex直传基本无差别。

补充：

1、iRTK2的A5版本最高波特率只能达到460800，但是经过多次测试暂未发现“unknown packet type”的错误。

2、改变obex的MTU大小会影响速度（MTU越大传输速度越快），但是A8平台改变MTU也解决不了“unknown packet type”的情况。

编写日期：2017年10月16日

编写人： 蔡展能