Linphone交叉编译过程（坑较多）：

基于版本： origin/release/4.2

在服务器上121.40.242.234路径： /home/app\_comm/linphone/linphone-desktop（编译环境是好的，不要破坏）

编译命令： make ARCH=arm

增加内容： linphone-sdk\mediastreamer2\src\otherfilters\t3audiofilter.c

在centos7上交叉编译海思版本linphone （origin/release/4.2）使用过程：

经常需要执行命令：

find . -name "\*.sh"|sudo xargs chmod +x

配置命令：

./configure --host=arm-linux --prefix=/home/app\_comm/arm-compile/armlib/ CC=/opt/hisi-linux/x86-arm/arm-himix200-linux/bin/arm-himix200-linux-gcc

编译使用命令：

./prepare.py desktop-raspberry -DENABLE\_OPENH264=ON -DENABLE\_WEBRTC\_AEC=OFF -DENABLE\_UNIT\_TESTS=OFF -DENABLE\_MKV=OFF -DENABLE\_FFMPEG=ON -DENABLE\_CXX\_WRAPPER=OFF -DENABLE\_NON\_FREE\_CODECS=ON -DENABLE\_VCARD=OFF -DENABLE\_BV16=OFF -DENABLE\_V4L=OFF

使用linphone提供的树莓派版本编译的方式，但是树莓派arm版本和海思arm版本不太一致，所以部分cmake配置需要手动修改

Arm版本信息：

执行（arm-himix200-linux-gcc –help可以获取一些关于工具相关环境信息）

arm-himix200-linux-gcc -print-sysroot

获取: /opt/hisi-linux/x86-arm/arm-himix200-linux/host\_bin/../target

编译过程依赖2个cmake配置文件，需要修改， 分别是：

1. linphone-desktop\linphone-sdk\cmake-builder\toolchains\toolchain-raspberry.cmake

2. linphone-desktop\linphone-sdk\cmake-builder\configs\config-desktop-raspberry.cmake

在2文件中添加

set(CMAKE\_SYSTEM\_PROCESSOR "armv7")

修改2文件关于ffmpeg的配置：

lcb\_builder\_cross\_compilation\_options(ffmpeg

"--prefix=${CMAKE\_INSTALL\_PREFIX}"

"--enable-cross-compile"

- "--cross-prefix=arm-linux-gnueabihf-"

+ "--cross-prefix=arm-himix200-linux-"

"--arch=arm"

"--target-os=linux"

)

修改1文件 注释掉关于环境的判断，添加自己交叉编译gcc路径：

-if("$ENV{RASPBERRY\_VERSION}" STREQUAL "")

set(RASPBERRY\_VERSION 1)

-else()

- if($ENV{RASPBERRY\_VERSION} VERSION\_GREATER 3)

- set(RASPBERRY\_VERSION 3)

- else()

- set(RASPBERRY\_VERSION $ENV{RASPBERRY\_VERSION})

- endif()

-endif()

-if("$ENV{RASPBIAN\_ROOTFS}" STREQUAL "")

- message(FATAL\_ERROR "Define the RASPBIAN\_ROOTFS environment variable to point to the raspbian rootfs.")

-else()

- set(SYSROOT\_PATH "$ENV{RASPBIAN\_ROOTFS}")

-endif()

-set(TOOLCHAIN\_HOST "arm-linux-gnueabihf")

+#if("$ENV{RASPBIAN\_ROOTFS}" STREQUAL "")

+# message(FATAL\_ERROR "Define the RASPBIAN\_ROOTFS environment variable to point to the raspbian rootfs.")

+#else()

+# set(SYSROOT\_PATH "$ENV{RASPBIAN\_ROOTFS}")

+#endif()

+#set(TOOLCHAIN\_HOST "arm-linux-gnueabihf")

+set(TOOLCHAIN\_HOST "arm-himix200-linux")

修改1文件, 因为arm版本不一样，需要查询自己编译的arm版本的c,c++flag的参数配上，添加-std=c++11 ，在编译btctoolbox一直报错，分析后发现是c++11标准参数没有添加上，添加后解决

-if(RASPBERRY\_VERSION VERSION\_GREATER 2)

- set(CMAKE\_C\_FLAGS "-mcpu=cortex-a53 -mfpu=vfp -mfloat-abi=hard" CACHE STRING "Flags for Raspberry PI 3")

- set(CMAKE\_CXX\_FLAGS "${CMAKE\_C\_FLAGS}" CACHE STRING "Flags for Raspberry PI 3")

-elseif(RASPBERRY\_VERSION VERSION\_GREATER 1)

- set(CMAKE\_C\_FLAGS "-mcpu=cortex-a7 -mfpu=vfp -mfloat-abi=hard" CACHE STRING "Flags for Raspberry PI 2")

- set(CMAKE\_CXX\_FLAGS "${CMAKE\_C\_FLAGS}" CACHE STRING "Flags for Raspberry PI 2")

-else()

- set(CMAKE\_C\_FLAGS "-mcpu=arm1176jzf-s -mfpu=vfp -mfloat-abi=hard" CACHE STRING "Flags for Raspberry PI 1 B+")

- set(CMAKE\_CXX\_FLAGS "${CMAKE\_C\_FLAGS}" CACHE STRING "Flags for Raspberry PI 1 B+")

-endif()

-set(CMAKE\_FIND\_ROOT\_PATH "${CMAKE\_INSTALL\_PREFIX}" "${CMAKE\_SYSROOT}")

+ set(CMAKE\_C\_FLAGS "-mcpu=cortex-a7 -mfpu=neon-vfpv4 -mfloat-abi=softfp" CACHE STRING "Flags for Raspberry PI 1 B+")

+set(CMAKE\_CXX\_FLAGS "${CMAKE\_C\_FLAGS}" CACHE STRING "Flags for Raspberry PI 1 B+")

+set(CMAKE\_CXX\_FLAGS "${CMAKE\_CXX\_FLAGS} -std=c++11")

+ set(CMAKE\_FIND\_ROOT\_PATH "${CMAKE\_INSTALL\_PREFIX}" "${CMAKE\_SYSROOT}")

Ms2编译问题：

修改 ms2 CmakeLists.txt

cmake\_dependent\_option(ENABLE\_V4L "Enable Video4Linux support." YES "ENABLE\_VIDEO;LINUX\_OR\_BSD" NO)

改为：

cmake\_dependent\_option(ENABLE\_V4L "Enable Video4Linux support." NO "ENABLE\_VIDEO;LINUX\_OR\_BSD" NO)

添加：

set(ENABLE\_X11\_DEFAULT\_VALUE NO)

set(ENABLE\_GLX\_DEFAULT\_VALUE NO)

下载alsa源码编译成arm版本使用：

地址：https://www.alsa-project.org/main/index.php/Main\_Page

Tar –xvf .

./configure --host=arm-linux --prefix=/home/app\_comm/arm-compile/armlib/ CC=/opt/hisi-linux/x86-arm/arm-himix200-linux/bin/arm-himix200-linux-gcc

Make

配置linphone-desktop/linphone-sdk/mediastreamer2/cmake/FindALSA.cmake; 因为ALSA是自定义的安装路径，导致find\_package无法找到；在FindALSA.cmake文件中添加：

set(ALSA\_INCLUDE\_DIRS "/home/app\_comm/arm-compile/armlib/include")

set(ALSA\_LIBRARIES "/home/app\_comm/arm-compile/armlib/lib/libasound.so")

解决这个问题；

Linphone编译问题

1. Ms2/cmake/FindFFmpeg.cmake添加

set(HAVE\_FUN\_avcodec\_get\_context\_defaults3 1)

set(HAVE\_FUN\_avcodec\_open2 1)

库链接出错：

arm-linux-gnueabi/bin/ld: warning: libbellesip.so.0, needed by ../src/ liblinphone.so.9, not found (try using -rpath or -rpath-link)

linphone编译时会出现动态库生成可以成功，但是console和daemon文件夹下的demo程序出现链接报错的问题；经过分析应该是arm库在x86环境下无法自动关联自己依赖的库，需要修改 console/CmakeLists.txt 和daemon/CmakeLists.txt文件，帮他链接关联进去，具体修改：

添加：

set(LIBS

${BELLESIP\_LIBRARIES}

${MEDIASTREAMER2\_LIBRARIES}

${ORTP\_LIBRARIES}

${BCTOOLBOX\_CORE\_LIBRARIES}

${XML2\_LIBRARIES}

${BELR\_LIBRARIES}

${LIBXSD\_LIBRARIES}

)

if(WIN32 AND NOT CMAKE\_SYSTEM\_NAME STREQUAL "WindowsStore")

list(APPEND LIBS "Ws2\_32")

endif()

if(LIME\_FOUND)

list(APPEND LIBS ${LIME\_LIBRARIES})

endif()

if(ENABLE\_LIME)

list(APPEND LIBS ${BZRTP\_LIBRARIES})

endif()

if(ZLIB\_FOUND)

list(APPEND LIBS ${ZLIB\_LIBRARIES})

endif()

if(SOCI\_FOUND)

list(APPEND LIBS ${SOCI\_LIBRARIES})

endif()

if(SQLITE3\_FOUND)

list(APPEND LIBS ${SQLITE3\_LIBRARIES})

endif()

if(ICONV\_FOUND)

list(APPEND LIBS ${ICONV\_LIBRARIES})

endif()

if(ENABLE\_TUNNEL)

list(APPEND LIBS ${TUNNEL\_LIBRARIES})

endif()

if(MSVC AND NOT CMAKE\_SYSTEM\_NAME STREQUAL "WindowsPhone" AND NOT CMAKE\_SYSTEM\_NAME STREQUAL "WindowsStore")

list(APPEND LIBS ${LIBGCC} ${LIBMINGWEX})

endif()

if(WIN32 AND NOT CMAKE\_SYSTEM\_NAME STREQUAL "WindowsPhone" AND NOT CMAKE\_SYSTEM\_NAME STREQUAL "WindowsStore")

list(APPEND LIBS shlwapi)

endif()

if(INTL\_FOUND)

list(APPEND LIBS ${INTL\_LIBRARIES})

endif()

if(BELCARD\_FOUND)

list(APPEND LIBS ${BELCARD\_LIBRARIES})

endif()

link\_directories("/home/app\_comm/linphone/linphone-desktop/OUTPUT/desktop-raspberry/lib")

list(APPEND LIBS mbedtls mbedx509 mbedcrypto swscale vpx bzrtp srtp2 gsm opus speex speexdsp avcodec avutil swresample bcg729 asound stdc++)

修改：

add\_executable(linphonec ${LINPHONEC\_SOURCE\_FILES})

-target\_link\_libraries(linphonec ${LINPHONE\_LIBS\_FOR\_TOOLS} ${BCTOOLBOX\_CORE\_LIBRARIES} ${ORTP\_LIBRARIES} ${MEDIASTREAMER2\_LIBRARIES} ${XSD\_LIBRARIES})

+target\_link\_libraries(linphonec ${LINPHONE\_LIBS\_FOR\_TOOLS} ${LIBS})

set\_target\_properties(linphonec PROPERTIES LINK\_FLAGS "${LINPHONE\_LDFLAGS}")

set\_target\_properties(linphonec PROPERTIES LINKER\_LANGUAGE CXX)

@@ -54,7 +104,10 @@ if(WIN32)

endif()

add\_executable(linphonecsh ${LINPHONECSH\_SOURCE\_FILES})

-target\_link\_libraries(linphonecsh ${LINPHONE\_LIBS\_FOR\_TOOLS} ${BCTOOLBOX\_CORE\_LIBRARIES} ${ORTP\_LIBRARIES})

+#target\_include\_directories(linphonecsh PUBLIC ${LINPHONE\_INCLUDE\_DIRS})

+#target\_link\_libraries(linphonecsh ${LINPHONE\_LIBS\_FOR\_TOOLS} ${BCTOOLBOX\_CORE\_LIBRARIES} ${ORTP\_LIBRARIES} mbedtls mbedx509 mbedcrypto)

+target\_link\_libraries(linphonecsh ${LINPHONE\_LIBS\_FOR\_TOOLS})

+target\_link\_libraries(linphonecsh ${LIBS})

set\_target\_properties(linphonecsh PROPERTIES LINK\_FLAGS "${LINPHONE\_LDFLAGS}")

set\_target\_properties(linphonecsh PROPERTIES LINKER\_LANGUAGE CXX)

daemon/CmakeLists.txt按照相同方式修改；