# 开发环境配置

1. **Android NDK：**

集成了android交叉编译器，针对不同版本android平台提供了相应的gcc、g++、ndk-build、ndk-gdb等一系列编译和调试工具。

安装ndk：

官网下载地址：<http://www.cnblogs.com/yaotong/archive/2011/01/25/1943615.html>

安装包选择：

选择安装包需要注意几点：

1. Android手机是32位还是64位，默认是32位，r9以前的只有32位，r10之后出现了64位的，需要注意,比如64位，则应该含有ndk64
2. 从r6开始各版本分为a,b,c,d系列，例如r9版本分为r9,r9b,r9c,r9d,一般选择r9或r10，r9d或r10d验证都存在一些问题
3. 编译主机系统是什么系统，32位还是64位的，比如是ubuntu 64位机，则应该是linux-x86\_64

比如，下载以64位linux系统作为主机的，32位android 平台的r10版本的ndk，下载地址为：http://dl.google.com/android/ndk/android-ndk32-r10-linux-x86\_64.tar.bz2

安装步骤(这里选ndk 32位android系统，主机系统为32位linux系统,r9版本)：

第一步：解压:tar jxvf android-ndk-r9-linux-x86.tar.bz2

第二步： 解压完成后进入ndk路径/build/tool/，开始配置环境，使用命令：

sudo  ./make-standalone-toolchain.sh  --platform=android-18  --ndk-dir=/home/ndk/  --install-dir=/opt/android-18-toolchain/ --toolchain=arm-linux-androideabi-4.6

说明：

android-18：使用的android API LEVEL，platform文件夹下可以看到，这里使用的是18

下表是android API 级别与android 版本的对应关系

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| API level | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Android 版本 | 4.4 | 4.3 | 4.2 | 4.1 | 4.03，4.04 | 4.0.0~4.02 | 3.2 | 3.1.x | 3.0.x | 2.3.3，2.3.4 | 2.3.0~2.3.2 | 2.2.x |

后面分别为ndk解压后的目录，和安装的目标目录。

第三步：配置路径，为了方便使用ndk的工具，可以将ndk的路径加入环境变量。终端输入：

export PATH=$PATH:/home/ndk,这样我们可以在任意地方直接使用ndk-build,ndk-gdb命令了。

export PATH=$PATH:/opt/android-18-toolchain/bin ,主要是交叉编译器工具集。

这样就配置完成了。

1. **Android SDK：**

提供了android sdk开发包直接将sdl库或执行文件与java代码一起编译成apk，但过程较繁琐，我们直接使用ant即可。还提供了adb、android模拟器等工具。

安装：adt-bundle-linux-x86-20140321.zip

推荐下载网址：http://pan.baidu.com/s/1AB9in

解压命令同上，解压完成后即可使用了，里面包含eclipse和sdk，不需要再进行配置其他，直接使用。

并将tools、platform-tools加入环境变量,这样就可以直接使用adb和模拟器了

1. **Ant工具：**

ant本来是sdk的一个子工具，但由于它集成了sdk的许多功能，用于一步打包成apk。

安装：apache-ant-1.9.4-bin.tar.gz

推荐下载网址：<http://download.csdn.net/download/sams_on361/7808511>

解压命令：tar -zxvf apache-ant1.9.4-bin.tar.gz

解压后，将bin目录加入环境变量，就可以直接使用ant命令了。

打包apk只需要在android工程目录下执行：ant release

1. **工具使用**
2. ndk交叉编译

在cmakelist里将编译器路径替换成ndk编译器路径即可编译成可以在android上运行的进程

SET(NDK\_PATH “/opt/android-18-toolchain”)

SET(CMAKE\_CXX\_COMPILER "${NDK\_PATH}/bin/arm-linux-androideabi-g++")

SET(CMAKE\_C\_COMPILER "${NDK\_PATH}/bin/arm-linux-androideabi-gcc")

1. 创建android工程

创建一个空文件夹，执行：

android create project --name proname --target android-19 --path ./ --package com.proname --activity activity\_name

比如创建一个sdl的android工程：

android create project --name SmartDevice --target android-18 --path ./ --package com.SmartDevice --activity MainActive

其中，target 想使用哪个，可以根据上表选择，但有些可能没有安装，可以通过命令：android list target 来查看

1. 更新android工程

从其他地方转移过来的android工程或想更新API LEVEL，或想更新项目名称可以使用以下命令：

android update project --name proname --target android-19 --path ./ --package com.proname --activity activity\_name

如想更新API LEVEL,只需要执行

android update project --target android-18 --path ./

1. 打包成apk

在android工程目录下，执行：ant release

1. 安装apk

Usb数据线连接好手机后，执行:adb install –r \*\*\*.apk

# Android sdl编译

1. **编译sdl**
2. 创建一个空目录
3. 进入目录，执行：

cmake -DSYSTEM\_NAME="Android" -DCMAKE\_BUILD\_TYPE="Release" -DCOMPILER\_PATH="/opt/android-18-toolchain" -DBUILD\_TARGET\_LIBRARY=ON <project path>

* 执行完后应该可以看到 current build system is Android这样的信息。
* -DSYSTEM\_NAME 是设置运行系统，这里必须设置成 "Android"
* -DCOMPILER\_PATH 是设置NDK的根目录，这里是"/opt/android-18-toolchain"

1. 执行make，开始编译
2. 等第3步完成后，执行make install，会生成一个bin目录，编译完成。

**然后把bin目录下的libsmartDeviceLinkCore.so 复制到<hmi\_sdk dir>/Library/android/sdl 下，其他文件复制到<hmi\_sdk dir>/Library/android/sdl/config/下。**