image**东软秘密**

文件编号：902EVSC21009-D00-D09-T09-001

NeuSAR aCore

平台编译出盘说明

东软睿驰汽车技术有限公司

**修 订 履 历**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **修改编号** | **版本** | **修改内容** | **状态** | **修改人/**  **日期** | **审批人/**  **日期** |
| 1 | 0.7 | 初版 | In review | 齐智/2021.1.12 |  |
| 2 | 1.0 | 版本发布 | Approved | 齐智/2021.1.29 | 李冰/2021.1.29 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**目 录**

[1 文档概述 1](#_Toc62826287)

[1.1 文档目的和范围 1](#_Toc62826288)

[1.1.1 术语/缩略语 1](#_Toc62826289)

[1.2 参考文档 1](#_Toc62826290)

[2 概述 1](#_Toc62826291)

[3 环境搭建 2](#_Toc62826292)

[3.1 环境准备 2](#_Toc62826293)

[4 平台应用编译及制作发布的SDK脚本 2](#_Toc62826294)

[4.1 步骤 2](#_Toc62826295)

[4.2 主要目录及文件简介 3](#_Toc62826296)

[4.2.1 build/config 3](#_Toc62826297)

[4.2.2 build/product/access/Makefile 3](#_Toc62826298)

[4.2.3 build/app/Makefile 4](#_Toc62826299)

[4.2.4 build/app/Makefile\_ARA 4](#_Toc62826300)

[4.2.5 build/app/rules 4](#_Toc62826301)

[4.2.6 build/app/applications 4](#_Toc62826302)

[4.2.7 build/app/ara-api 4](#_Toc62826303)

[4.2.8 build/app 4](#_Toc62826304)

[4.2.9 build/product/access/release/adaptive 4](#_Toc62826305)

[4.2.10 build/product/access/release/output/ 5](#_Toc62826306)

[4.2.11 build/scripts/sdk\_file 5](#_Toc62826307)

[4.2.12 build/scripts/make\_package.sh 5](#_Toc62826308)

[4.3 打开源代码包 5](#_Toc62826309)

[4.4 注意事项 5](#_Toc62826310)

[5 制作x86\_linux安装盘 6](#_Toc62826311)

[5.1 步骤 6](#_Toc62826312)

[5.2 输出结构介绍 6](#_Toc62826313)

[6 附加说明 6](#_Toc62826314)

# 文档概述

## 文档目的和范围

本文档为NeuSAR aCore平台编译平台应用，制作发布的SDK脚本，以及制作NeuSAR aCore平台的x86\_linux版本的ISO安装盘的说明文档，为平台开发人员提供编译平台环境提供指导和出盘的指导。

本文档的方法不是标准，仅供参考，可根据实际情况修改为符合自己特点的编译及出盘工具。

### 术语/缩略语

|  |  |  |
| --- | --- | --- |
| 序号 | 术语/缩略语 | 说明 |
| 1 | 平台 | NeuSAR aCore平台 |
| 2 |  |  |

## 参考文档

|  |  |  |
| --- | --- | --- |
| 序号 | 文档名 | 版本 |
| 1 |  |  |
| 2 |  |  |

# 概述

平台的开发人员，需要维护及开发平台应用程序，修改后，需要重新编译使修改生效，并使用修改后的程序重新制作安装盘，此文档即为介绍如何进行编译平台应用以及如何制作x86\_linux的ISO安装盘的文档。

本文档读者为平台的应用开发人员以及需要了解平台出盘以及需要对平台应用开发方法有进一步了解的人员。

使用本文档的读者理论上对平台了解较深的群体，而且本文档的内容为内部编译，所以，在输入限制等情况的异常校验可能会存在不足的情况，理论上由开发人员自行避免错误使用的场景。如后续扩展情况未得到支持，亦可自行进行调整以适应新场景。

# 环境搭建

平台作为一个基础软件平台，对于不同的硬件，需要使用相应硬件（板子）提供的原始工具链进行编译，才能最终将平台部署到对应的硬件上，以及生成包含平台的相应硬件的SDK，本文档以x86\_linux为例进行说明，其他硬件环境，理论上使用相应板子提供的原始工具链替换此说明中x86\_linux的原始工具链即可，但可能会存在不同的工具链所包含的底层库不同，导致平台编译失败的情况，需要根据实际问题再调查解决。

环境搭建步骤如下：

1. 宿主机安装文件传输工具，例如WinSCP，以便于与编译环境进行文件传输
2. 虚拟机安装linux系统，例如Ubuntu18.04.4；4核或以上，目前的编译选项是-j4，在4核上能达到最佳。
3. 虚拟机配置好网络
4. 创建编译目录，例如mkdir ~/NeuSAR\_aCore\_compile
5. 编译安装必要软件：sudo apt-get update; sudo apt install openssh-server gcc make git python gawk
6. 创建制作ISO目录，例如mkdir ~/NeuSAR\_aCore\_make\_iso
7. 制作ISO除上述编译需要的软件外，还需要安装：

* sudo apt install syslinux-utils
* sudo apt install python-pip ;pip install tenjin enum typing pathlib lxml

# 平台应用编译及制作发布的SDK脚本

## 步骤

1. 将aCore，build，x86\_linux\_toolchain上传到~/NeuSAR\_aCore\_compile目录下。
2. cd build/
3. ./config
4. source toolchain/environment.config
5. cd product/access/
6. make all

## 主要目录及文件简介

以下介绍的目录结构以运行“4.1步骤”章节后的结构进行介绍。

### build/config

准备编译环境的脚本，主要使用x86\_linux\_toolchain目录下的工具生成原始的工具链，并将编译好的开源包放在相应的位置。将aCore中的源码链接到相应的位置供编译使用。生成一些编译时需要的文件等。

默认情况下，其会在build目录下生成toolchain目录，即为工具链的目录。首次./config后，后续再次编译时，并不需要再./config，只需要source toolchain/environment.config引用上工具链的环境，即可再次进行编译。

./config –help可以查看其帮助。以下为各参数的介绍：

* -c参数为清理，此参数只清理./config所做的链接，不清理./config的生成文件以及编译产生的文件，执行到此参数后，会退出脚本。执行清理后，再次编译会失败。
* -v参数可以为编译指定版本。
* -a可以选择编译动作，0为编译aCore代码，1为编译开源代码。0动作编译后会生成SDK脚本，供使用本平台的用户应用程序编译使用。生成的SDK脚本为product/access/release/output/deploySdk/sdk\_x86\_linux.sh。
* -f指定配置文件，配置文件里面需要配置aCore，原始工具链的目录。最好配置绝对目录，如果配置相对目录，为相对build的目录。
* -d调试模式。会将脚本执行的每步打印出来，通过set –x实现。
* -o输出工具链的路径，默认为build目录下的toolchain目录。对于开发人员来说，每次修改代码再生成sdk脚本，再释放脚本过于浪费时间，所以，在开发阶段，可以使用此目录作为用户应用的sdk脚本生成的目录使用。这样，每次make software-[模块名]后此目录就已经更新了。对于make software-[模块名]请继续往下看就会了解到了。

### build/product/access/Makefile

平台的编译入口文件。主要选项及说明：

* make software-[模块名] 编译指定模块，各模块的名都是什么，参见后面app/Makefile和app/Makefile\_ARA。
* make software-[模块名]-install 安装指定模块。
* make software-[模块名]-clean 清理指定模块编译产生的临时文件。
* make software 相当于make software全部模块。
* make release 相当于make software全部模块的install。
* make clean 相当于make software全部模块的clean。
* make distclean 除make clean外还清理编译时环境准备的产生物（非./config产生的）等。
* make all 整体编译及打sdk脚本

### build/app/Makefile

obj-y += [模块名] 其中模块名为平台所使用的开源代码的模块名。

### build/app/Makefile\_ARA

obj-y += [模块名] 其中模块名为平台应用或系统应用的模块名（经常被修改的开源代码会移到这里）。

### build/app/rules

[模块名].mk为相应模块的编译文件，模块的源码可以此文件的定义里找到。

### build/app/applications

系统应用及在上位机无法配置的部分出厂配置。

### build/app/ara-api

模块的源码目录。

### build/app

大部分目录为开源代码的源码。

### build/product/access/release/adaptive

make release的目录。

### build/product/access/release/output/

编译输出的目录，为客户提供的sdk在build/product/access/release/output/deploySdk目录下。

### build/scripts/sdk\_file

sdk打包的文件列表。

### build/scripts/make\_package.sh

运行环境需要的文件的打包的脚本。

## 重新制作开源代码包

* ./config –a 1
* source toolchain/environment.config
* cd product/access/
* make all
* cd release/adaptive
* mv etc usr/
* cd usr/
* echo “开源版本” > openSrcVersion
* tar czf openSrc.tgz --exclude=share \*
* mv -f openSrc.tgz ~/NeuSAR\_aCore\_compile/x86\_linux\_toolchain/
* rm –fr ~/NeuSAR\_aCore\_compile/toolchain
* 重新进行“4.1步骤”章节即为使用新的开源库进行编译

## 注意事项

* 如果用其它脚本对其进行使用时，尽量不要以软链接的形式使用，以免目录结构的部署出现问题。
* 同一台机器上，不要并发编译，以免出现问题（特别是开源代码的编译）。

# 制作x86\_linux安装盘

## 步骤

1. 将aCore，build，factory\_config，jsonGenerator，make\_iso，x86\_linux\_toolchain上传到~/NeuSAR\_aCore\_compile目录下。
2. cd make\_iso
3. ./make\_iso.sh -v aCore\_test

## 主要目录及文件简介

以下介绍的目录结构以运行“5.1步骤”章节后的结构进行介绍。

### make\_iso/make\_iso.sh

出盘脚本。

./make\_iso.sh –help可以查看其帮助。以下为各参数的介绍：

* -f指定配置文件，配置文件里面需要配置各目录。最好配置绝对目录，如果配置相对目录，为相对make\_iso的目录。
* -v参数可以为ISO指定版本。
* -c为清理生成的目录，清理后会退出脚本运行。

### make\_iso/rootfs

此目录为制作ISO的文件系统的目录

### make\_iso/output

输出目录，其中ISO目录为生成的安装盘目录，安装盘名为neusar\_20210129\_[version].iso

# 附加说明

出盘与编译尽量不要同时进行。