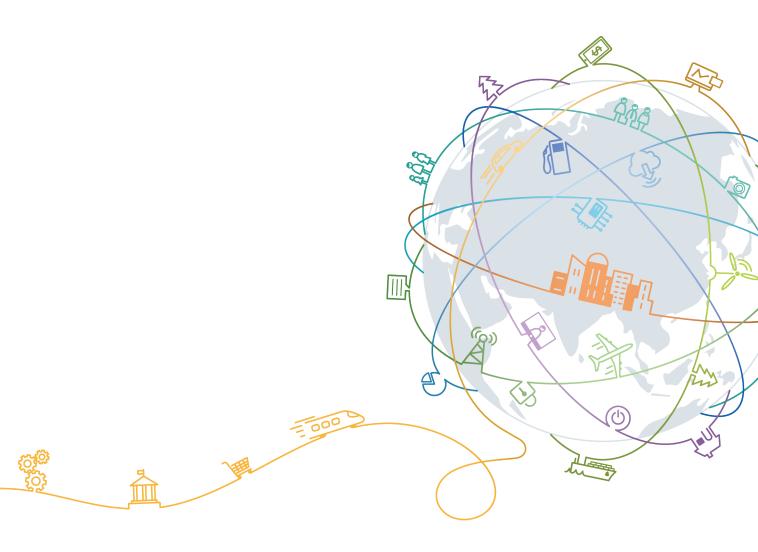
Atlas 500

DDK 安装指南 (Ubuntu, X86)

文档版本 01

发布日期 2019-09-30





版权所有 © 华为技术有限公司 2019。 保留一切权利。

非经本公司书面许可,任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部,并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。 本文档提及的其他所有商标或注册商标,由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束,本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定,华为公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因,本文档内容会不定期进行更新。除非另有约定,本文档仅作为使用指导,本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为技术有限公司

地址: 深圳市龙岗区坂田华为总部办公楼 邮编: 518129

网址: http://e.huawei.com

前言

读者对象

本文档主要适用于以下工程师:

- 华为技术支持工程师
- 渠道伙伴技术支持工程师
- ISV软件工程师

符号约定

在本文中可能出现下列标志,它们所代表的含义如下。

符号	说明
▲ 危险	用于警示紧急的危险情形,若不避免,将会导致人员死亡 或严重的人身伤害。
▲警告	用于警示潜在的危险情形,若不避免,可能会导致人员死 亡或严重的人身伤害。
△ 注意	用于警示潜在的危险情形,若不避免,可能会导致中度或 轻微的人身伤害。
注意	用于传递设备或环境安全警示信息,若不避免,可能会导致设备损坏、数据丢失、设备性能降低或其它不可预知的结果。 "注意"不涉及人身伤害。
□ 说明	用于突出重要/关键信息、最佳实践和小窍门等。 "说明"不是安全警示信息,不涉及人身、设备及环境伤害。

修改记录

文档版本	发布日期	修改说明
01	2019-09-30	第一次正式发布版本。

目录

前言	ii
- ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' '	
3 环境准备	
4 DDK 安装	
4.2 安装	
4.3 常用操作	
4.3.1 卸载 DDK	
4.3.2 查询 DDK 版本	8
4.3.3 配置 OpenPGP 公钥	8
5 FAQ	11
- 5.1 Mind Studio 或 DDK 安装过程中提示 pip2 或 pip 不可用	11
5.2 软件包完整性校验返回 WARNING 或 FAIL	
6 附录	14
6.1 软件包说明	
62 开源第三方库说明	15

1 简介

本文介绍独立部署DDK(Device Development Kit)的安装过程以及常见问题处理方法,适用于基于后台命令行的方式进行开发的场景。

DDK为用户提供基于NPU的数字开发者套件。DDK可以用于构建相关工程的编译环境。不同的发布包里集成了不同NPU形态的DDK。当前版本的DDK集成了TE、DVPP、流程编排等组件的依赖库和头文件,用户可以通过makefile编译相应的工程文件。

2 软件包获取

在工具安装前,请准备如下软件包。

DDK安装包:请访问https://support.huawei.com/enterprise,在搜索栏中搜索对应产品名进入产品节点下,单击"软件"页签,获取DDK安装包。

软件包说明如表2-1所示。

表 2-1 软件包说明

安装包	包说明	软件包完整性校验文件
MSpore_DDK-{version}- x86_64.ubuntu16.04- x86_64.ubuntu16.04- aarch64.miniOS.tar.gz	非开发者板形态(Atlas 300等)场景对应的 DDK安装包。	MSpore_DDK-{version}-x86_64.ubuntu16.04-x86_64.ubuntu16.04-aarch64.miniOS.tar.gz.asc
MSpore_DDK-{version}-x86_64.ubuntu16.04-aarch64.ubuntu16.04-aarch64.ubuntu16.04.tar.gz	开发者板形态(Atlas 200 DK)场景对应的 DDK安装包。	MSpore_DDK-{version}-x86_64.ubuntu16.04-aarch64.ubuntu16.04-aarch64.ubuntu16.04.tar.g

□说明

{version}为DDK版本号。

3 环境准备

环境要求

安装DDK的环境,所要求的硬件及操作系统要满足以下条件。

表 3-1 Ubuntu 系统版本信息

类别	版本 限制	获取方式	注意 事项
操作系统	16.04.	请从http://old-releases.ubuntu.com/releases/16.04.3/网站下载对应版本软件进行安装,可以下载桌面版: "ubuntu-16.04.3-desktop-amd64.iso",或Server版: "ubuntu-16.04.3-server-amd64.iso"。	-
Pyt hon	Pytho n2: 2.7+ Pytho n3: 3.5+	请参见 安装依赖 。	-

创建 DDK 安装用户

安装前需要创建好DDK用户,并使用该用户安装。本节所有操作请在root用户下执行。

目前推荐使用非root用户安装,且确保该用户\$HOME目录权限为750,支持任意非root 用户安装,如果需要使用新用户安装DDK,则使用如下命令创建新用户。

- 1. 执行以下命令创建DDK安装用户并设置该用户的\$HOME目录。useradd -d /home/username -m username
- 2. 执行以下命令设置密码。 passwd username
- 3. 执行以下命令设置权限,进入"/home"目录。chmod 750 /home/username

□ 说明

username为安装Mind Studio的用户名,该用户的umask值不能大于0027:

- 若要查看umask的值,则执行命令: umask
- 若要修改umask的值,则执行命令: umask 新的取值

若创建的普通用户为**HwHiAiUser**,且**DDK**部署在host侧服务器,此时**DDK**安装用户和host普通用户为同一个,使用该用户进行离线模型转换时,日志默认输入到host侧var/dlog目录中,您也可以通过设置如下环境变量,使日志直接输出到屏幕上。

export SLOG_PRINT_TO STDOUT=1

配置 DDK 安装用户权限

DDK安装过程中需要使用sudo apt-get权限,请以root用户执行如下操作。

1. 打开 "/etc/sudoers" 文件:

chmod u+w /etc/sudoers
vi /etc/sudoers

2. 在该文件"#User privilege specification"下面增加如下内容:

username ALL=(ALL:ALL) NOPASSWD:SETENV:/usr/bin/apt-get

"username"为执行安装脚本的普通用户名。

□ 详明

请确保"/etc/sudoers"文件的最后一行为"#includedir/etc/sudoers.d",如果没有该信息,请手动添加。

- 3. 添加完成后,执行:wq!保存文件。
- 4. 执行以下命令取消 "/etc/sudoers" 文件的写权限:

chmod u-w /etc/sudoers

检查源

DDK安装过程需要下载相关依赖,请确保安装DDK的服务器能够连接网络。

请在root用户下执行如下命令检查源是否可用。

apt-get update

◯◯说明

如果命令执行报错,则检查网络是否连接或在"/etc/apt/sources.list"文件中更换可用的源。

安装依赖

请切换到DDK安装用户执行如下操作,安装DDK工具依赖的gcc、python等组件。

步骤1 执行以下命令安装DDK相关依赖。

sudo -E apt-get install gcc g++ cmake make python-pip python3-pip python3 python

步骤2 安装 Python开发环境

- 1. Python2环境安装
 - 环境部署依赖pip、setuptools;安装TE软件包依赖: numpy, decorator, 执行如下命令分别安装:
 - \$ pip2 install setuptools --user
 - \$ pip2 install numpy --user
 - \$ pip2 install decorator --user

- Python2环境检查

使用如下命令分别检查python、pip、setuptools:

- \$ python2 --version
- \$ pip2 --version
- \$ easy_install --version

若分别返回如下所示信息,则说明安装成功:

Python 2. 7. 12

pip 19.0.1 from /usr/local/lib/python2.7/dist-packages/pip (python 2.7) setuptools 40.6.3 from /home/username/.local/lib/python2.7/site-packages (Python 2.7)

2. Python3 (3.5+)环境安装

- 环境部署依赖pip3、setuptools;安装TE软件包依赖: numpy, decorator, 执行如下命令分别安装:

```
$ pip3 install setuptools --user
```

- \$ pip3 install numpy --user
- \$ pip3 install decorator --user
- Python3环境检查

使用如下命令分别检查python3、pip3、setuptools:

- \$ python3 --version
- \$ pip3 --version
- \$ easy install3 --version

若分别返回如下所示信息,则说明安装成功:

Python 3. 5. 2

pip 8.1.1 from /usr/lib/python3/dist-packages (python 3.5) setuptools 20.7.0 from /usr/lib/python3/dist-packages (Python 3.5)

----结束

□ 说明

- 如果依赖安装过程中pip安装异常,请参见5.1 Mind Studio或DDK安装过程中提示pip2或pip 不可用解决。
- 上述安装成功后返回信息中的版本号只是样例,请以实际环境中的返回信息为准。

4 DDK 安装

本节介绍DDK的安装方法以及常用操作。

- 4.1 安装前准备
- 4.2 安装
- 4.3 常用操作

介绍DDK安装以及使用过程中的常用操作。

4.1 安装前准备

上传安装包

使用DDK的安装用户将如下文件上传到\$HOME目录任意路径下。

- "MSpore_DDK****.tar.gz": DDK安装包
- "MSpore DDK****.tar.gz.asc": DDK安装包校验文件。

□ 说明

DDK具体使用包名请参见表2-1, "*"代表DDK具体安装包。以上两个文件请确保在同一目录。

软件包完整性校验

为了防止软件包在传输过程中由于网络原因或存储设备原因出现下载不完整或文件破坏的问题,在执行安装前,建议您对软件包的完整性进行校验。

在DDK安装包所在目录,执行如下操作:

步骤1 配置OpenPGP公钥信息,请参考4.3.3 配置OpenPGP公钥。

步骤2 使用DDK安装用户执行如下命令,检测软件包是否合法完整,如图4-1所示。

gpg --verify "MSpore_DDK****.tar.gz.asc"

图 4-1 软件包完整性检测

- 返回信息中"27A74824"为公钥ID。
- 提示信息返回"Good signature"且信息中无 WARNING 或 FAIL,表明此签名为有效签名,软件包完整性校验通过。
- 若提示信息存在 WARNING 或 FAIL,则表明验证不通过,请参见**5.2 软件包完整性校验返回WARNING或 FAIL**处理建议解决。

川说明

- 操作时,请将MSpore DDK****.tar.gz.asc替换为实际安装包对应的校验文件。
- 软件包和软件包.asc文件必须放在同一个路径,才能进行完整性校验。

----结束

4.2 安装

前提条件

已完成3环境准备和4.1安装前准备。

操作步骤

请切换到DDK安装用户执行如下操作。

步骤1 执行如下命令解压安装包。

tar -zxvf MSpore DDK***tar.gz

1288

操作时请将 $MSpore\ DDK*****.tar.gz$ 替换为实际安装包,解压后文件说明请参见表6-1。

步骤2 执行如下命令安装DDK。

bash install.sh DDK安装目录

门设明

DDK安装目录在执行安装时会自动创建,例如安装目录设置为"\$HOME/tools/che/ddk",则在安装过程中会自动创建"tools/che/ddk"目录。您也可以自己指定安装路径。

若系统出现"Successfully installed the DDK!"表明安装成功。

----结束

□□说明

DDK安装完毕后,安装路径下"\$HOME/tools/che/ddk/ddk/toolchains/profiler"的Profiling功能不可用,如果用户需要使用Profiling功能,则请参见《Ascend 310 Mind Studio工具安装指南》手册(请根据Mind Studio安装服务器选择对应形态的安装手册)安装Mind Studio与DDK。

4.3 常用操作

介绍DDK安装以及使用过程中的常用操作。

4.3.1 卸载 DDK

如果您想更新DDK版本,则请卸载当前版本然后再按照5.1-安装步骤安装新的版本,本章节给出DDK的卸载方法。

进入DDK安装路径,本示例为"\$HOME/tools/che/ddk/ddk/scripts",使用DDK安装用户执行如下命令卸载DDK:

```
./uninstall.sh
```

如果出现如下所示信息,则说明DDK卸载成功:

```
Info: ide_daemon pem uninstall succ Starting to remove ddk dir.
```

4.3.2 查询 DDK 版本

DDK 安装成功后, 您可以在DDK 安装目录下查看DDK 版本号。

例如本示例DDK安装目录为"\$HOME/tools/che/ddk",则您可以在"\$HOME/tools/che/ddk/ddk"目录下执行如下命令查看DDK版本号:

```
cat ddk info
```

返回结果如图4-2所示。

图 4-2 查询 DDK 版本号

```
ascend@szvphicpra61963:~/tools/che/ddk/ddk$ cat ddk_info {
    "VERSION": "1.3.Tl3 =882",
    "NAME": "DDK",
    "TARGET": "ASIC"
}ascend@szvphicpra61963:~/tools/che/ddk/ddk$ ■
```

其中:

- VERSION:表示DDK版本号,查询结果请以实际环境中的为准。
- TARGET:表示DDK的运行环境,目前包括ASIC和Atlas DK。

4.3.3 配置 OpenPGP 公钥

前提条件

- 请使用DDK的安装用户配置公钥。
- Linux系统已经安装GnuPG工具。

检查方法:

- 若已经安装GnuPG工具,在 Shell中输入 **gpg** --version命令,可看到如**图4-3**所示的回显信息。

图 4-3 信息回显

- 若没有安装GnuPG工具,则在GnuPG的官方网站https://www.gnupg.org/,按照网站的指引,完成工具安装。

配置公钥

步骤1 获取公钥文件。

进入OpenPGP下载页面,单击下载链接,如图4-4所示,界面跳转到文件下载页面。

图 4-4 单击下载文件

版本	发布时间	是否过期
V100R001C00	2017-12-29	未过期

文件名为"KEYS"的文件为公钥文件,如图4-5所示。

图 4-5 选择 KEYS 文件

□ 软件名称	文件大小	发布时间	下载
■ KEYS.txt	1.26KB	2019-01-21	<u>*</u>
■ OpenPGP签名验证指南.pdf	1.72MB	2019-01-21	<u>*</u>
■ VerificationTools.rar	3.44MB	2019-01-21	<u>*</u>

下载

□说明

步骤2 将下载的KEYS.txt文件上传到DDK所在linux系统中。

例如传到"/home/username/openpgp/keys"新建目录中。

步骤3 导入公钥文件。

执行如下命令进入 KEYS 公钥文件所在的目录。

gpg --import "/home/username/openpgp/keys/KEYS.txt"

图 4-6 导入公钥文件

```
ascend@szvphicpra61963:~$ gpg --import "/home/ascend/openpgp/keys/KEYS.txt"
gpg: keyring '/home/ascend/,gnupg/secring.gpg' created
gpg: /home/ascend/,gnupg/trustdb.gpg: trustdb created
gpg: key 27A74824: public key "OpenPGP signature key for Huawei software (created on 30th Dec,2013) <support@huawei.com>" imported
gpg: Total number processed: 1
gpg: __imported: 1 (RSA: 1)
```

□说明

其中"/home/username/openpgp/keys"是公钥文件"KEYS"所在的绝对路径, username为DDK的安装用户名,请修改为实际路径。

步骤4 执行如下命令查看公钥导入结果。

gpg --fingerprint

图 4-7 查看结果

```
ascend@szvphicpra61963:~$ gpg --fingerprint
/home/ascend/.gnupg/pubring.gpg
pub 2048<mark>R/27A74824</mark> 2013-12-30
Key fingerprint = B100 0AC3 8C41 525A 19BD C087 99AD 81DF 27A7 4824
uid OpenPGP signature key for Huawei software (created on 30th Dec,2013) <support@huawei.com>
```

步骤5 验证公钥。

- OpenPGP 公钥的合法性需要根据公钥的 ID、指纹、uid 等信息与发布公钥的主体进行合法性验证。当前对外发布的OpenPGP公钥信息如下:
 - 公钥 ID: 27A74824
 - 公钥指纹(Key fingerprint): B100 0AC3 8C41 525A 19BD C087 99AD 81DF 27A7 4824
 - 用户 ID(uid): OpenPGP signature key for Huawei software (created on 30th Dec, 2013) < support@huawei.com>

完成信息核实后,可以对该公钥设置信任级别。

● 执行如下命令设置公钥的信任级别。 gpg --edit-key "OpenPGP signature key for Huawei software" trust

屏幕显示类似如下信息,其中红框部分需要手工输入,"Your decision?"后输入"5",表示"I trust ultimately"; "Do you really want to set this key to ultimate trust? (y/N)"后输入"y"。

图 4-8 设置公钥信任级别

步骤6 执行quit命令退出。

----结束

5 FAQ

- 5.1 Mind Studio或DDK安装过程中提示pip2或pip不可用
- 5.2 软件包完整性校验返回WARNING或 FAIL

5.1 Mind Studio 或 DDK 安装过程中提示 pip2 或 pip 不可用

问题描述

Mind Studio或DDK安装过程中提示pip2或pip不可用,并退出安装,提示信息如图5-1、图5-2所示。

图 5-1 pip2 不可用提示信息

```
Python 2.7.12

Traceback (most recent call last):
File "/usr/bin/pip2", line 9, in <module>
load_entry_point('pip=8.1.1', 'console_scripts', 'pip2')()
File "/usr/local/lib/python2.7/dist-packages/setuptools-0.6c11-py2.7.egg/pkg_resources.py", line 318, in load_entry_point
File "/usr/local/lib/python2.7/dist-packages/setuptools-0.6c11-py2.7.egg/pkg_resources.py", line 2221, in load_entry_point
File "/usr/local/lib/python2.7/dist-packages/setuptools-0.6c11-py2.7.egg/pkg_resources.py", line 2221, in load_entry_point
File "/usr/lib/python2.7/dist-packages/pip/_init__.py", line 18, in <module>
from pip.commands import get_summaries, get_similar_commands
File "/usr/lib/python2.7/dist-packages/pip/commands/_init__.py", line 8, in <module>
from pip.commands.freeze import FreezeCommand
File "/usr/lib/python2.7/dist-packages/pip/commands/freeze.py", line 8, in <module>
from pip.operations.freeze import freeze
File "/usr/lib/python2.7/dist-packages/pip/operations/freeze.py", line 11, in <module>
from pip._vendor.pkg_resources import RequirementParseError
ImportErforc.cannot import name RequirementParseError
#### Your pip2 has problem, please solve it before install MindStudio, Exit 1
```

图 5-2 pip 不可用提示信息

```
Traceback (most recent call last):
File "/usr/bin/pip", line 9, in <module>
from pip import main
ImportError: cannot import name main
```

可能原因

该问题可能是由于pip重新安装过程中,pip2没有正确更新。

解决方法1

步骤1 su root切换到root用户,执行**pip list**,如果返回信息没有错误提示,说明**pip**可用;执行**pip2 list**提示错误,说明**pip**2不可用。

步骤2 root用户下删除pip2,删除命令为: rm /usr/bin/pip2。

步骤3 将pip2软连接到pip, 命令为: ln -s pip pip2。

步骤4 再次执行pip2 list, 若返回信息没有错误提示,则说明问题解决。

若pip、pip2仍不可用,则参见解决方法2。

----结束

解决方法 2

如果依赖安装过程中pip安装异常,请依次执行如下命令。

sudo apt-get remove python-pip python3-pip
wget https://bootstrap.pypa.io/get-pip.py
python get-pip.py --user
python3 get-pip.py --user

5.2 软件包完整性校验返回 WARNING 或 FAIL

软件包完整性校验如果返回WARNING或 FAIL,则表示验证未通过,请参见表5-1处理建议解决。

表 5-1 场景举例

验证结果场 景	输出信息举例	验证 结果	处理建议
签名验证通 过,没有异 常	gpg: Signature made Thu Jan 9 15:29:06 2014 CST using RSA key ID 27A74824 gpg: Good signature from "OpenPGP signature key for Huawei software (created on 30th Dec, 2013) < support@huawei.com>"	PAS S	NA
签名验证失 败	gpg: Signature made Thu Jan 9 15:29:06 2014 CST using RSA key ID 27A74824 gpg: BAD signature from "OpenPGP signature key for Huawei software (created on 30th Dec, 2013) < support@huawei.com>"	FAIL	重新下载目 标文件。
找不到公钥	gpg: Signature made Thu Jan 9 15:20:01 2014 CST using RSA key ID 27A74824 gpg: Can't check signature: public key not found	FAIL	重新下载公 钥,请参见 "配置 OpenPGP公 钥-配置公 钥"章节。

验证结果场 景	输出信息举例	验证结果	处理建议
签名验证通 过,但是公 钥没有被设 置为完全信 任	gpg: Signature made Thu Jan 9 15:29:06 2014 CST using RSA key ID 27A74824 gpg: Good signature from "OpenPGP signature key for Huawei software (created on 30th Dec, 2013) < support@huawei.com>" gpg: WARNING: This key is not certified with a trusted signature! gpg: There is no indication that the signature belongs to the owner. Primary key fingerprint: B100 0AC3 8C41 525A 19BD C087 99AD 81DF 27A7 4824	WAR NIN G	确认KeyID为 27A74824 后,将华为 公钥设置为 可信,请置 见"配置 OpenPGP公 钥-配置公钥- 验证公钥" 章节。
找不到对应 的源文件	gpg: no signed data gpg: can't hash datafile: No data	FAIL	重新下载目 标文件。
签名已到期	gpg: Signature made 04/24/13 10:50:29 CST using RSA key ID 133B64E5 gpg: Expired signature from "OpenPGP signature test key <support@huawei.com>" gpg: Signature expired 04/25/13 10:50:29 CST</support@huawei.com>	FAIL	下载更新过 签名的目标 文件。
签名验证通 过,但是公 钥已被撤销	gpg: Signature made 06/13/13 11:14:49 CST using RSA key ID 133B64E5 gpg: Good signature from "OpenPGP signature test key <support@huawei.com>" gpg: WARNING: This key has been revoked by its owner! gpg: This could mean that the signature is forged. gpg: reason for revocation: Key is no longer used gpg: revocation comment:</support@huawei.com>	WAR NIN G	下载最新公 钥和更新了 签名的目标 文件。
源文件找不 到对应的签 名文件	无	WAR NIN G	下载目标文 件对应的签 名文件。

6 附录

- 6.1 软件包说明
- 6.2 开源第三方库说明

6.1 软件包说明

DDK安装包解压后包的详细内容请参见表6-1。

表 6-1 软件包说明

安装包	包内容	使用场景
MSpore_DDK-{version}- <uihost arch.os="">-<host arch.os>-<device arch.os>.tar.gz 各参数解释请参见表 6-2。</device </host </uihost>	ddk.tar.gz	DDK套件安装包。
	install.sh	安装脚本。
	check_sha.sh	校验以上两个文件的完整性,install.sh执行过程中会自动调用该脚本进行完整性校验。

表 6-2 DDK 安装包命名规范

参数	说明
{version}	版本号
<uihost arch.os=""></uihost>	UIhost侧 CPU架构和操作系统及版本,如: x86_64.ubuntu16.04、x86_64.centOS7.4等
<host arch.os=""></host>	Host侧 CPU架构和操作系统及版本,如: x86_64.ubuntu16.04、x86_64.centOS7.4等

参数	说明
<device arch.os=""></device>	Device侧 CPU架构和操作系统及版本,如: aarch64.ubuntu16.04、aarch64.miniOS等 ● 开发者板形态(Atlas 200 DK)device侧架构: aarch64.ubuntu16.04 ● 非开发者板形态(Atlas 300等)device侧架构: aarch64.miniOS

6.2 开源第三方库说明

cereal

cereal是一个开源的(BSD License)、轻量级的、支持C++11特性的、仅仅包含头文件实现的、跨平台的C++序列化库。它可以将任意的数据类型序列化成不同的表现形式,比如二进制、XML格式或JSON。

详细介绍可参考cereal官网: http://uscilab.github.io/cereal/

gflags

gflags (the Global Flags Editor)是google的一个开源的处理命令行参数的库,比 getopt 更 方便,更功能强大,使用c++开发,具备python接口。

详细介绍可参考gflags官网: https://github.com/gflags/gflags

glog

glog是Google的一个开源库,用于实现应用级别的logging。 它提供了一系列类似于C++流风格的logging API,以及一些预定义的宏。它有点类似于C里面的assert,但是比它具备更丰富的输出信息以及使用灵活性。

详细介绍可参考glog官网: https://github.com/google/glog

opencv

OpenCV的全称是Open Source Computer Vision Library,是一个跨平台的计算机视觉库。OpenCV可用于开发实时的图像处理、计算机视觉以及模式识别程序。

详细介绍可参考opencv官方网站: https://opencv.org/

Protobuf

Protobuf(Google Protocol Buffers)是Google提供一个具有高效的协议数据交换格式工具库;它是一种平台无关、语言无关、可扩展且轻便高效的序列化数据结构的协议,可以用于网络通信和数据存储。当前使用版本: 3.5.1

详细介绍可参考google官方网站: https://developers.google.com/protocol-buffers/