

OpenHarmony 轻量设备嵌入式编程开发 环境搭建

文档版本 01
发布日期 2026-01-26

目录

目录.....	3
3 轻量设备嵌入式编程开发环境搭建	4
3.1 实验介绍.....	4
3.1.1 实验背景.....	4
3.1.2 实验原理.....	4
3.1.3 实验目的.....	4
3.1.4 实验环境.....	4
3.2 实验任务.....	5
3.2.1 任务一 Windows Subsystem for Linux 2 环境安装	5
3.2.2 任务二 从仓库拉取 OpenHarmony 5.0.3 release 轻量系统代码	7
3.2.4 任务三 烧录编译的 hi3861 固件.....	10

1 轻量设备嵌入式编程开发环境搭建

1.1 实验介绍

1.1.1 实验背景

搭建 HiSpark hi3861 开发板的 OpenHarmony 5.0 Release 的南向系统开发环境。

1.1.2 实验原理

搭建 WSL2 环境，拉取 OpenHarmony 5.0 Release mini 源码，通过 build.sh 脚本执行源码编译。

1.1.3 实验目的

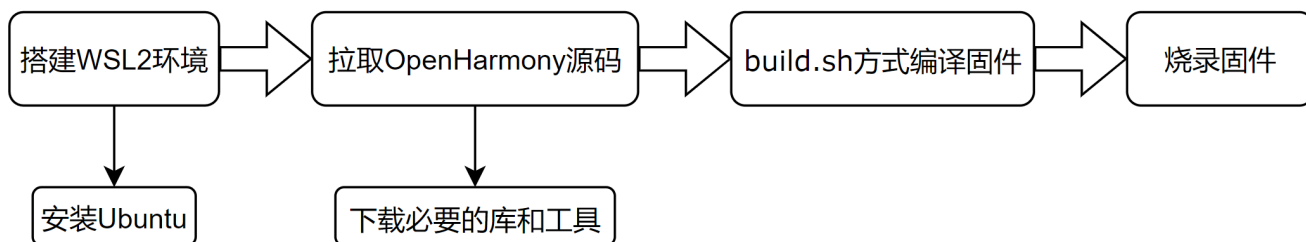
熟练掌握 OpenHarmony 南向系统源码开发环境安装流程和步骤。

1.1.4 实验环境

本实验只讲解 Windows 下的开发环境安装。

- 操作系统：Windows 10/11 64 位
- 对于 x64 系统：版本 1903 或更高版本，内部版本为 18362 或更高版本。
- 内存：16GB 及以上
 - 硬盘：100GB 及以上

1.1.5 实验流程



1.2 实验任务

1.2.1 任务一 Windows Subsystem for Linux 2 环境安装

1.启用适用于 Linux 的 Windows 子系统

需要先启用“适用于 Linux 的 Windows 子系统”可选功能，然后才能在 Windows 上安装 Linux 分发。

以管理员身份打开 PowerShell（“开始”菜单 > “PowerShell” > 单击右键 > “以管理员身份运行”），然后输入以下命令：

```
dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart
```

2.启用虚拟机功能

安装 WSL 2 之前，必须启用“虚拟机平台”可选功能。计算机需要虚拟化功能才能使用此功能。

以管理员身份打开 PowerShell 并运行：

```
dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart
```

安装完成之后必须要重启电脑

3.下载 Linux 内核更新包 下载最新包：适用于 x64 计算机的 WSL2 Linux 内核更新包



wsl.2.5.10.0.x64.msi 傻瓜式安装即可

4.将 WSL 2 设置为默认版本

使用命令默认设置为 WSL 2。

```
wsl --set-default-version 2
```

5.安装所选的 Linux 分发

执行下面命令安装 Ubuntu-22.04

```
wsl --install -d Ubuntu-22.04
```

6.设置 Ubuntu 用户名和密码以及 root 用户密码，切换 Ubuntu 下载源为华为云镜像

```

jiajiahao@jiajiahao: ~
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: jiajiahao
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
操作成功完成。
Installation successful!
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

jiajiahao@jiajiahao:~$

```

备份配置文件:

```
sudo cp -a /etc/apt/sources.list /etc/apt/sources.list.bak
```

修改 sources.list 文件, 将 <http://archive.ubuntu.com> 和 <http://security.ubuntu.com> 替换成 <http://repo.huaweicloud.com>, 可以参考如下命令:

```
sudo sed -i "s@http://.*archive.ubuntu.com@http://repo.huaweicloud.com@g" /etc/apt/sources.list
```

```
sudo sed -i "s@http://.*security.ubuntu.com@http://repo.huaweicloud.com@g" /etc/apt/sources.list
```

执行 `sudo apt-get update` 更新索引。

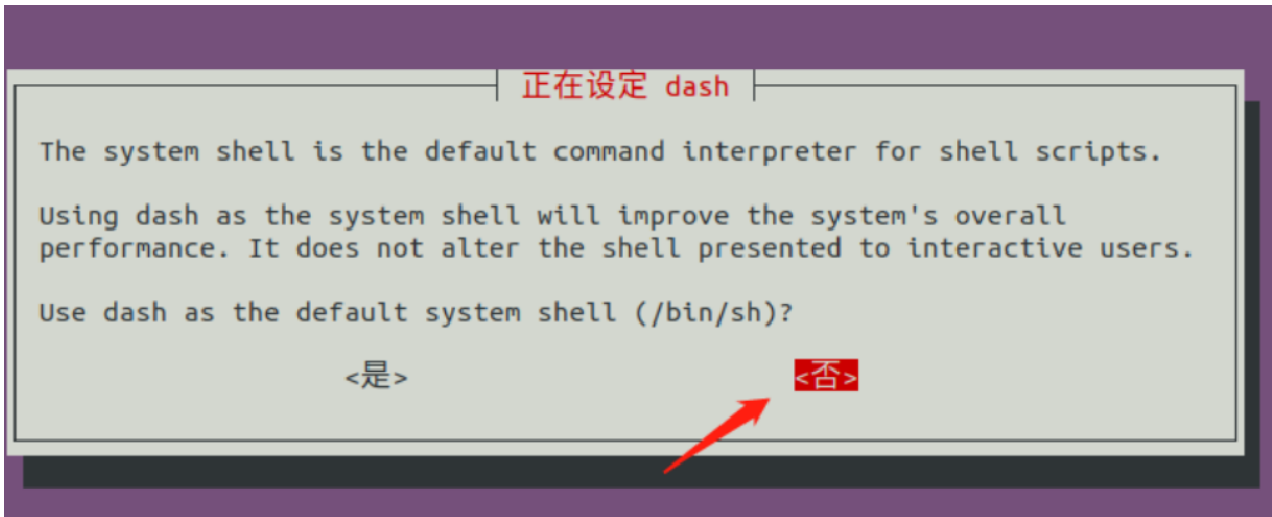
```

jiajiahao@jiajiahao:~$ sudo cp -a /etc/apt/sources.list /etc/apt/sources.list.bak
jiajiahao@jiajiahao:~$ sudo sed -i "s@http://.*security.ubuntu.com@http://repo.huaweicloud.com@g" /etc/apt/sources.list
jiajiahao@jiajiahao:~$ sudo sed -i "s@http://.*archive.ubuntu.com@http://repo.huaweicloud.com@g" /etc/apt/sources.list
jiajiahao@jiajiahao:~$ sudo apt-get update
Get:1 http://repo.huaweicloud.com/ubuntu bionic InRelease [242 kB]
Get:2 http://repo.huaweicloud.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://repo.huaweicloud.com/ubuntu bionic-backports InRelease [83.3 kB]
Get:4 http://repo.huaweicloud.com/ubuntu bionic-security InRelease [88.7 kB]
Get:5 http://repo.huaweicloud.com/ubuntu bionic/main amd64 Packages [1019 kB]
Get:6 http://repo.huaweicloud.com/ubuntu bionic/main Translation-en [516 kB]
Get:7 http://repo.huaweicloud.com/ubuntu bionic/restricted amd64 Packages [9184 B]
Get:8 http://repo.huaweicloud.com/ubuntu bionic/restricted Translation-en [3584 B]
Get:9 http://repo.huaweicloud.com/ubuntu bionic/universe amd64 Packages [8570 kB]
Get:10 http://repo.huaweicloud.com/ubuntu bionic/universe Translation-en [4941 kB]
Get:11 http://repo.huaweicloud.com/ubuntu bionic/multiverse amd64 Packages [151 kB]
Get:12 http://repo.huaweicloud.com/ubuntu bionic/multiverse Translation-en [108 kB]
Get:13 http://repo.huaweicloud.com/ubuntu bionic-updates/main amd64 Packages [3045 kB]
Get:14 http://repo.huaweicloud.com/ubuntu bionic-updates/main Translation-en [554 kB]
Get:15 http://repo.huaweicloud.com/ubuntu bionic-updates/restricted amd64 Packages [1347 kB]
Get:16 http://repo.huaweicloud.com/ubuntu bionic-updates/restricted Translation-en [187 kB]
Get:17 http://repo.huaweicloud.com/ubuntu bionic-updates/universe amd64 Packages [1915 kB]
Get:18 http://repo.huaweicloud.com/ubuntu bionic-updates/universe Translation-en [421 kB]
Get:19 http://repo.huaweicloud.com/ubuntu bionic-updates/multiverse amd64 Packages [25.6 kB]
Get:20 http://repo.huaweicloud.com/ubuntu bionic-updates/multiverse Translation-en [6888 B]
Get:21 http://repo.huaweicloud.com/ubuntu bionic-backports/main amd64 Packages [53.3 kB]
Get:22 http://repo.huaweicloud.com/ubuntu bionic-backports/main Translation-en [14.6 kB]
Get:23 http://repo.huaweicloud.com/ubuntu bionic-backports/universe amd64 Packages [18.2 kB]
Get:24 http://repo.huaweicloud.com/ubuntu bionic-backports/universe Translation-en [8668 B]
Get:25 http://repo.huaweicloud.com/ubuntu bionic-security/main amd64 Packages [2717 kB]
Get:26 http://repo.huaweicloud.com/ubuntu bionic-security/main Translation-en [467 kB]
Get:27 http://repo.huaweicloud.com/ubuntu bionic-security/restricted amd64 Packages [1317 kB]
Get:28 http://repo.huaweicloud.com/ubuntu bionic-security/restricted Translation-en [182 kB]
Get:29 http://repo.huaweicloud.com/ubuntu bionic-security/universe amd64 Packages [1303 kB]
Get:30 http://repo.huaweicloud.com/ubuntu bionic-security/universe Translation-en [308 kB]
Get:31 http://repo.huaweicloud.com/ubuntu bionic-security/multiverse amd64 Packages [19.8 kB]
Get:32 http://repo.huaweicloud.com/ubuntu bionic-security/multiverse Translation-en [3928 B]
Fetched 29.7 MB in 20s (1469 kB/s)
Reading package lists... Done

```

7.将 Ubuntu Shell 环境修改为 bash, 在终端输入如下命令:

```
sudo dpkg-reconfigure dash
```



执行 `ls -l /bin/sh` 后选择否

```
root@jiajiahao:/home/jiajiahao/OpenHarmony/3.2r_dayu200/src# ls -l /bin/sh
lrwxrwxrwx 1 root root 4 Jan 24 17:17 /bin/sh -> bash
```

更改成功，如上图所示

8.安装 git、git-lfs、ninja-build、curl、python3-pip，在终端输入如下命令

```
sudo apt-get install -y git git-lfs ninja-build curl python3-pip scons
```

9.安装码云 repo 工具，在终端输入如下命令

```
sudo curl https://gitee.com/oschina/repo/raw/fork_flow/repo-py3 -o /usr/local/bin/repo
```

```
sudo chmod a+x /usr/local/bin/repo
```

```
sudo pip3 install -i https://repo.huaweicloud.com/repository/pypi/simple requests
```

10.设置 git 全局用户名和电子邮件地址，在终端输入如下命令。

gitte_user 替换为 gitte 账号名，gitte_mail 替换为 gitte 绑定的邮箱，也可以不退换，直接复制下面的命令即可。

```
git config --global user.name "yourname"
git config --global user.email "your-email-address"
git config --global credential.helper store
```

1.2.2 任务二 从仓库拉取 OpenHarmony 5.0.3 release 轻量系统代码

1.下载 OpenHarmony-v5.0.3-Release 版本 hi3861 源码

```
sudo ln -sf /usr/bin/python3 /usr/bin/python //创建软连接
```

在/root/下新建一个用于存储代码的目录，比如 oh_5.0.3_mini，然后切换到该目录下执行下面的代码

```
repo init -u https://gitee.com/openharmony/manifest -b refs/tags/OpenHarmony-v5.0.3-Release -m chipsets/hi  
spark_pegasus.xml -g ohos:mini
```

2.执行 `repo sync -c` 和 执行最后报错则重新执行 `repo sync -c`，直到提示 `repo sync has finished successfully`。
执行 `repo forall -c 'git lfs pull'`。

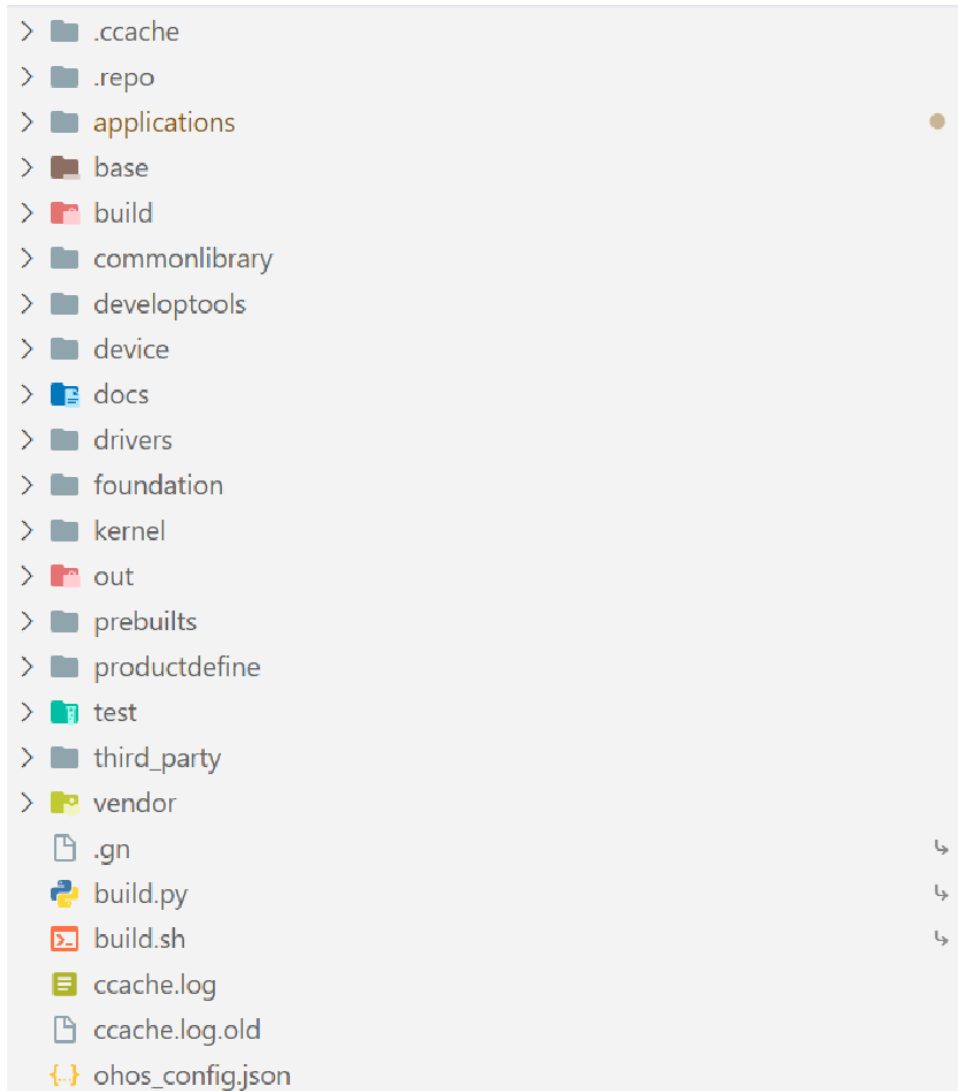
```
repo sync -c
```

```
repo forall -c 'git lfs pull'
```

4.下载编译工具

```
bash build/prebuilts_download.sh
```

- 下载好 OpenHarmony 5.0.3 release 轻量系统 hi3861 芯片源码目录结构如下



5. 下载需要的 python 库

```
pip install ecdsa -i https://pypi.tuna.tsinghua.edu.cn/simple
```

```
pip install scons -i https://pypi.tuna.tsinghua.edu.cn/simple
```

```
pip install pycryptodome -i https://pypi.tuna.tsinghua.edu.cn/simple
```

```
pip install genconfig -i https://pypi.tuna.tsinghua.edu.cn/simple
```

```
pip install kconfiglib -i https://pypi.tuna.tsinghua.edu.cn/simple
```

6. 配置 riscv 编译

下载 https://repo.huaweicloud.com/harmonyos/compiler/gcc_riscv32/7.3.0/linux/gcc_riscv32-linux-7.3.0.tar.gz

然后解压并配置环境变量

安装步骤如下：

1、下载 riscv32 gcc 镜像，执行如下命令

```
wget https://repo.huaweicloud.com/harmonyos/compiler/gcc_riscv32/7.3.0/linux/gcc_riscv32-linux-7.3.0.tar.gz
```

2、解压

```
sudo tar -xvf gcc_riscv32-linux-7.3.0.tar.gz -C /opt/
```

3、设置环境变量路径

```
vim ~/.bashrc
```

添加 `export PATH=/opt/gcc_riscv32/bin:$PATH` 到最后一行

5、source ~/.bashrc

6、再次执行：riscv32-unknown-elf-gcc -v

如果能够看到如下版本信息，则配置成功

```
rdw@STU23:~$ riscv32-unknown-elf-gcc -v
Using built-in specs.
COLLECT_GCC=riscv32-unknown-elf-gcc
COLLECT_LTO_WRAPPER=/opt/gcc_riscv32/bin/./libexec/gcc/riscv32-unknown-elf/7.3.0/lto-wrapper
Target: riscv32-unknown-elf
Configured with: ../riscv-gcc/configure --prefix=/home/yuanwenhong/gcc_compiler_riscv/gcc_riscv32/gcc_riscv32 --target=riscv32-unknown-elf --with-arch=rv32imc --with-abi=ilp32 --disable-__cxa_atexit --disable-libgomp --disable-libmudflap --enable-libssp --disable-libstdcxx-pch --disable-nls --disable-shared --disable-threads --disable-multilib --enable-poison-system-directories --enable-languages=c,c++ --with-gnu-as --with-gnu-ld --with-newlib --with-system-zlib CFLAGS='-fstack-protector-strong -O2 -D_FORTIFY_SOURCE=2 -Wl,-z,relro,-z,now,-z,noexecstack -fPIE' CXXFLAGS='-fstack-protector-strong -O2 -D_FORTIFY_SOURCE=2 -Wl,-z,relro,-z,now,-z,noexecstack -fPIE' LDFLAGS='-Wl,-z,relro,-z,now,-z,noexecstack' CXXFLAGS_FOR_TARGET=-Os -mmodel=medlow -Wall -fstack-protector-strong -Wl,-z,relro,-z,now,-z,noexecstack -Wtrampolines -fno-short-enums -fno-short-wchar' CFLAGS_FOR_TARGET=-Os -mmodel=medlow -Wall -fstack-protector-strong -Wl,-z,relro,-z,now,-z,noexecstack -Wtrampolines -fno-short-enums -fno-short-wchar' --with-headers=/home/yuanwenhong/gcc_compiler_riscv/gcc_riscv32/gcc_riscv32/riscv32-unknown-elf/include --with-mpc=/usr/local/mpc-1.1.0 --with-gmp=/usr/local/gmp-6.1.2 --with-mpfr=/usr/local/mpfr-4.0.2
Thread model: single
gcc version 7.3.0 (GCC)
rdw@STU23:~$
```

7. 开始编译

```
./build.sh --product-name wifiot_hisparc_pegasus --ccache --no-prebuilt-sdk
```

编译完成之后的镜像文件所在位置

> Linux > Ubuntu-04 > root > oh_5.0.3_mini > out > hispark_pegasus > wifiot_hispark_pegasus >

名称	修改日期	类型	大小
.ninja_log	2025/11/6 18:34	NINJA_LOG 文件	166 KB
all_parts_host.json	2025/7/22 20:21	JSON 源文件	12 KB
all_parts_info.json	2025/11/6 18:34	JSON 源文件	13 KB
args.gn	2025/11/6 18:33	GN 文件	2 KB
binary_installed_parts.json	2025/11/6 18:34	JSON 源文件	1 KB
build.1753186889.8693316.log	2025/7/22 20:21	文本文档	311 KB
build.log	2025/11/6 18:34	文本文档	42 KB
build.ninja	2025/11/6 18:33	NINJA 文件	98 KB
build.ninja.d	2025/11/6 18:33	D 文件	20 KB
build.trace.1753186889.gz	2025/7/22 20:21	WinRAR 压缩文件	16 KB
build.trace.gz	2025/11/6 18:34	WinRAR 压缩文件	19 KB
Hi3861_boot_signed.bin	2025/11/6 18:34	BIN 文件	24 KB
Hi3861_boot_signed_B.bin	2025/11/6 18:34	BIN 文件	24 KB
Hi3861_loader_signed.bin	2025/11/6 18:34	BIN 文件	15 KB
Hi3861_wifiot_app.asm	2025/11/6 18:34	ASM 文件	16,495 KB
Hi3861_wifiot_app.map	2025/11/6 18:34	MAP 文件	4,960 KB
Hi3861_wifiot_app.out	2025/11/6 18:34	OUT 文件	1,553 KB
Hi3861_wifiot_app_burn.bin	2025/11/6 18:34	BIN 文件	756 KB
Hi3861_wifiot_app_flash_boot_ota.bin	2025/11/6 18:34	BIN 文件	25 KB
Hi3861_wifiot_app_ota.bin	2025/11/6 18:34	BIN 文件	434 KB
Hi3861_wifiot_app_vercfg.bin	2025/11/6 18:34	BIN 文件	1 KB
OHOS_Image.bin	2025/11/6 18:34	BIN 文件	771 KB
pcid.sc	2025/11/6 18:33	SC 文件	1 KB
sorted_action_duration.txt	2025/11/6 18:34	文本文档	106 KB
src_installed_parts.json	2025/7/22 20:21	JSON 源文件	12 KB
src_sa_infos_tmp.json	2025/7/22 20:21	JSON 源文件	1 KB
toolchain.ninja	2025/11/6 18:33	NINJA 文件	185 KB

这个 OHOS_Image.bin 就是烧录文件，它所在的路径根据自己 ubuntu 的实际代码路径来。

1.2.3 任务三 烧录编译的 hi3861 固件

1.使用 HiBurn 工具烧录二进制程序到 Hi3861 芯片 工具下载地址:

https://gitee.com/hihope_iot/docs/blob/master/HiSpark_WiFi_IoT/Software/tools/HiBurn.exe

2.安装 CH341SER 驱动 工具下载地址

https://gitee.com/hihope_iot/docs/blob/master/HiSpark_WiFi_IoT/Software/tools/CH341SER.ZIP

3.使用 USB 线连接到 3861 开发板，打开下载好的 HiBurn 工具

4.点击 select file 选择要下载的 OHOS_Image.bin 文件。勾选上 Auto burn 选项。单击 Connect 按钮,开始烧录

