

OpenHarmony应用开发手册

本文档旨在为开发者提供一个完整的操作指南，详细介绍如何在x86_64架构的Windows PC上使用DevEco Studio开发调试OpenHarmony应用，并最终将其成功部署到龙芯（LoongArch）设备上。

在开始之前，建议开发者对 OpenHarmony 应用开发有基本的了解。您可以参考以下官方文档，快速掌握基础知识：

- (1) 华为开发者联盟文档: 《从零开始开发 HarmonyOS 应用（ETS 语言）》
- (2) NDK 开发指南: 《NDK 开发概述》

1.1 准备工作

开发环境要求：

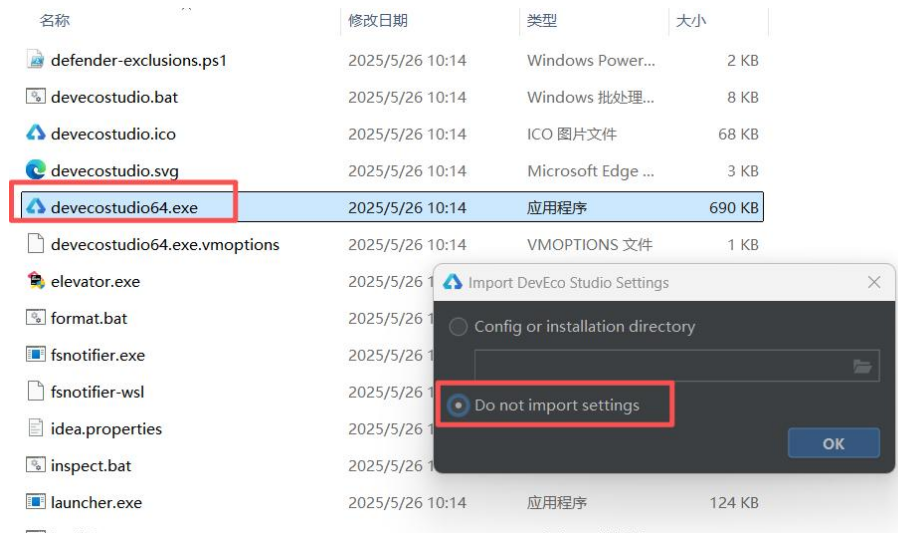
- (1) 操作系统：Windows10 或 Windows11
- (2) 开发工具：DevEco Studio NEXT 5.1
- (3) SDK版本：SDK API Level 12+

开发工具和SDK可直接从网盘获取：

<https://pan.baidu.com/s/1f2OdRemHyfDUF2b4k-CCUg?pwd=tuyW>

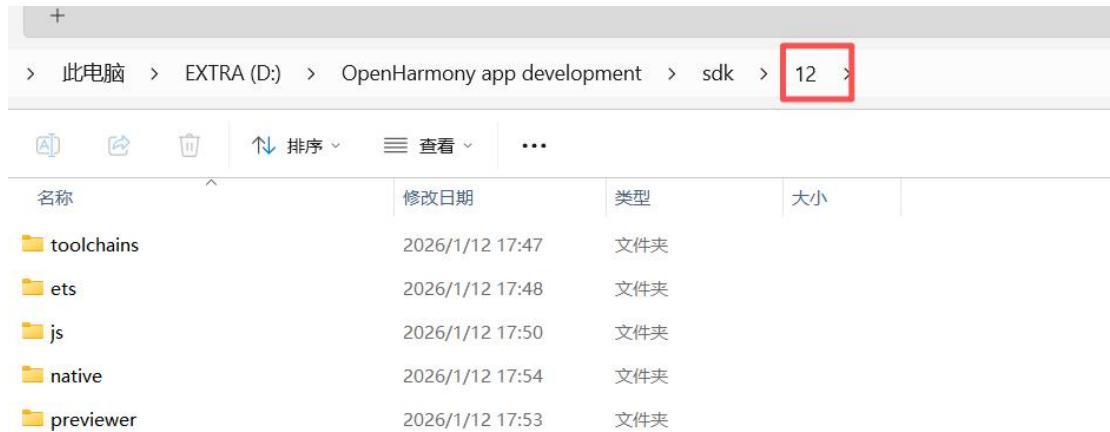
1.1.1 下载 DevEco Studio for LoongArch

解压devecostudio-windows-5.1.0.828-la.zip到本地任意路径，应用程序即为bin/devecostudio64.exe，双击打开即可。



1.1.2 下载 OpenHarmony SDK for LoongArch

在本地新建名为12的文件夹（此处12代表SDK API Level 21），将ohos-sdk.tar.gz压缩包中的五个包（ets, js, native, previewer, toolchains）解压到12目录下。如下图所示：



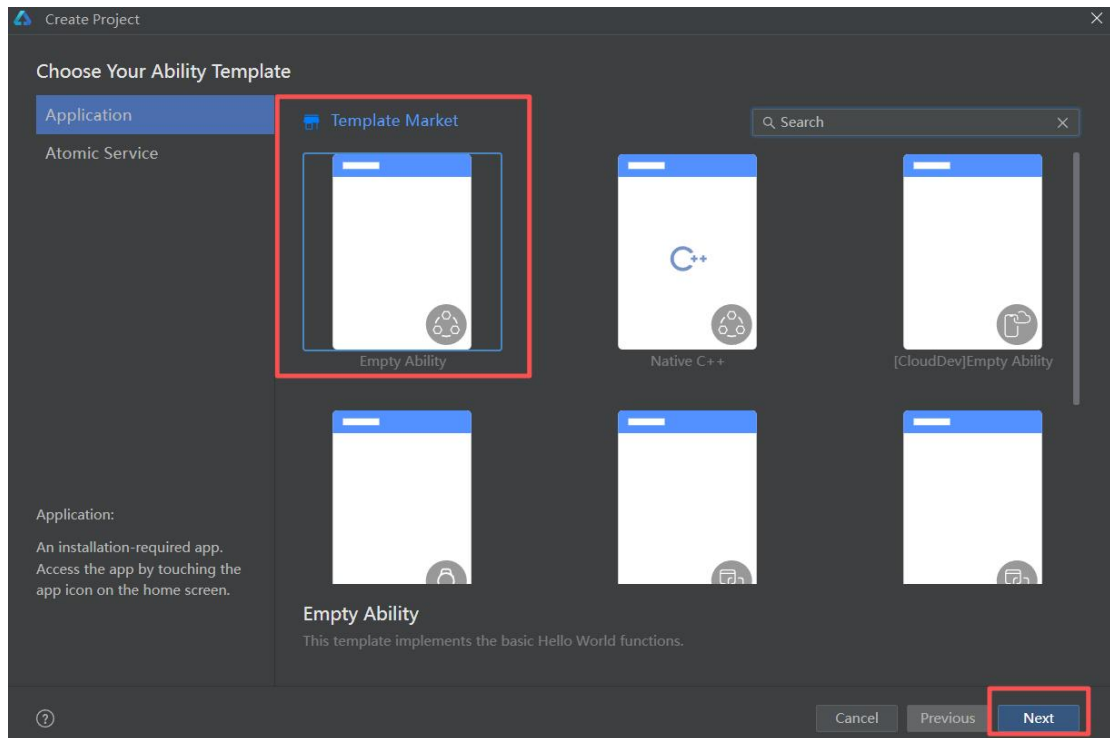
1.2 环境搭建

1.2.1 新建工程

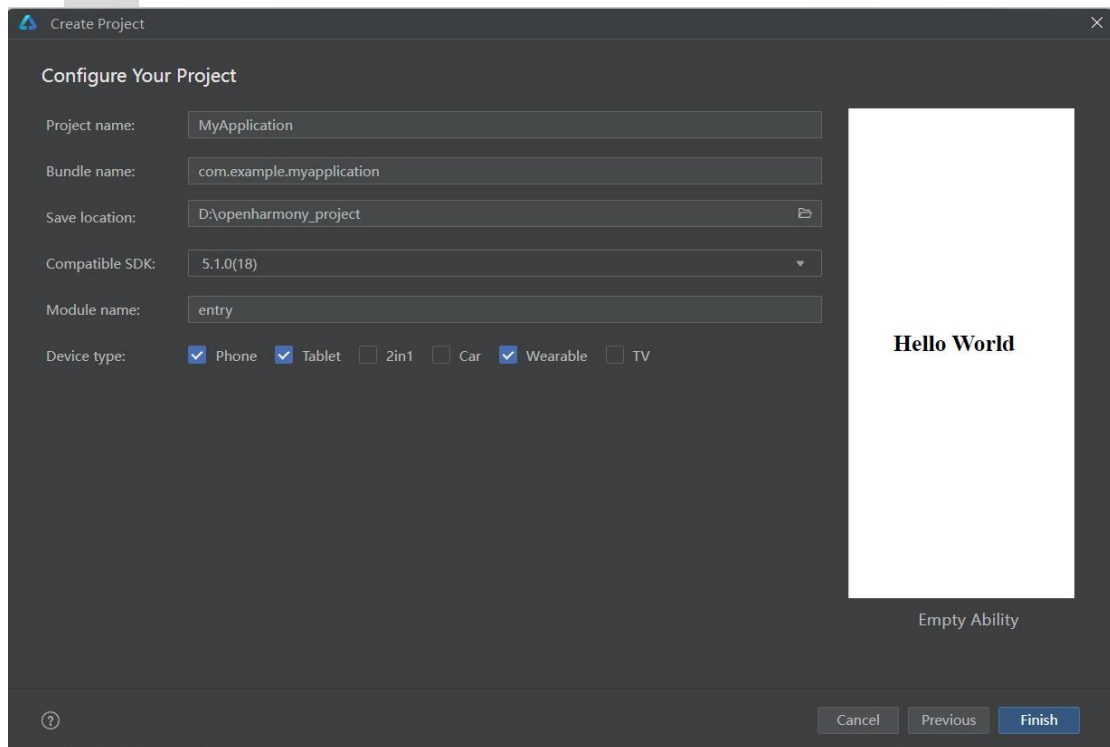
根据下图进行配置（具体可参考华为开发者联盟文档《[从零开始开发 HarmonyOS 应用（ETS 语言）](#)》）

若首次打开DevEco Studio，请点击Create Project创建工程。如果已经打开了一个工程，请在菜单栏选择File > New > Create Project来创建一个新工程。

选择Application应用开发（本文以应用开发为例，Atomic Service对应为元服务开发），选择模板Empty Ability，点击Next进行下一步配置。



点击Finish，工具会自动生成示例代码和相关资源，等待工程创建完成。



1.2.2 配置工程build-profile.json5文件

修改工程根目录下的build-profile.json5文件，确保应用能在OpenHarmony系统运行。

修改内容：将products标签下的runtimeOS字段修改为"OpenHarmony"，并将compatibleSdkVersion与compileSdkVersion调整为12。

```
{
  "app": {
    "signingConfigs": [],
    "products": [
      {
        "name": "default",
        "signingConfig": "default",
        "compatibleSdkVersion": 12,
        "compileSdkVersion": 12,
        "runtimeOS": "OpenHarmony",

```

1.2.3 配置模块module.json5文件

修改模块级别的module.json5文件，以适配目标设备类型。

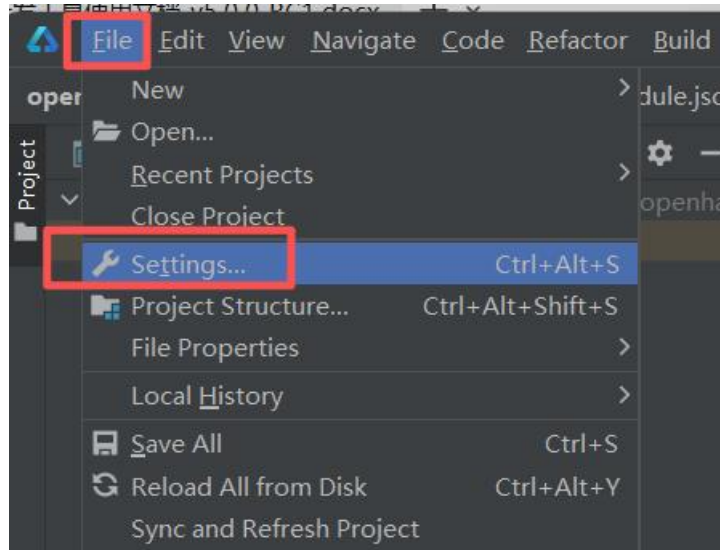
修改内容: 将deviceTypes字段修改为 "default" 和 "tablet"。

```
{
  "module": {
    "name": "entry",
    "type": "entry",
    "description": module description,
    "mainElement": "EntryAbility",
    "deviceTypes": [
      "default",
      "tablet",
    ],

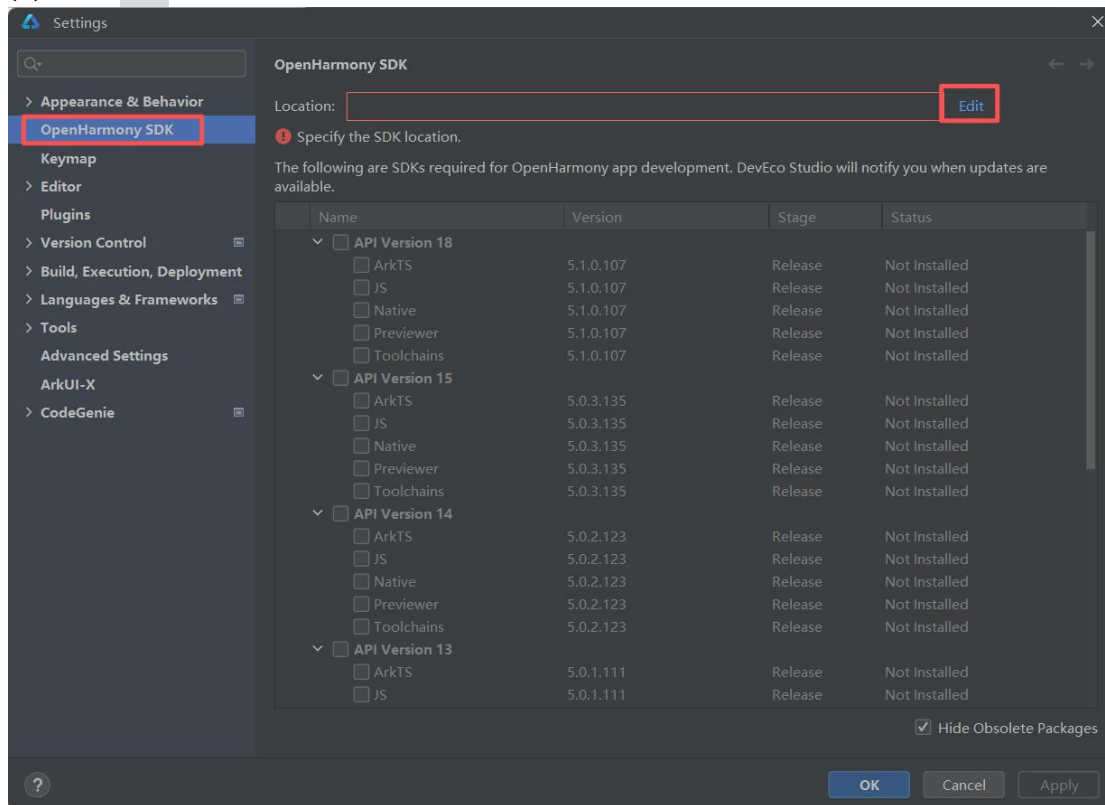
```

1.2.4 配置龙芯SDK

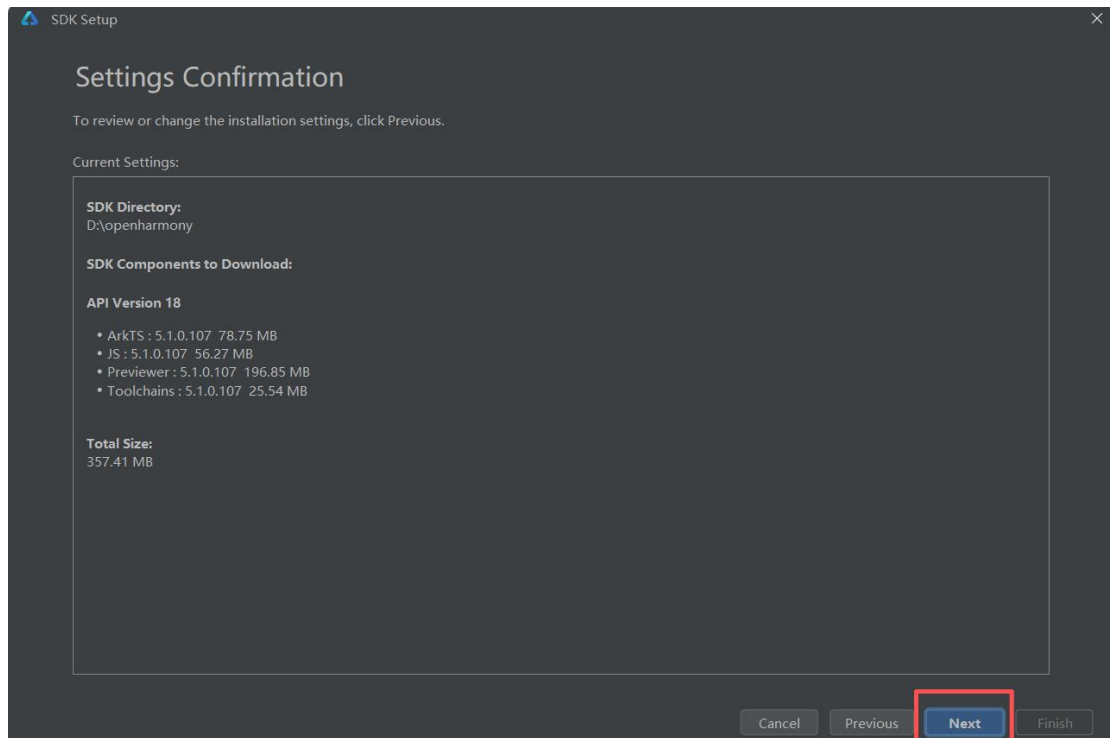
(1) 选择File->Settings-> OpenHarmony SDK



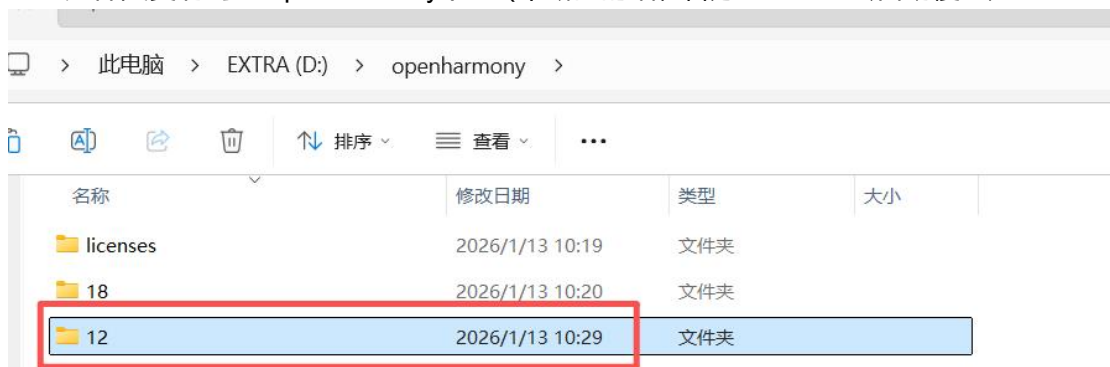
(2) 点击Edit



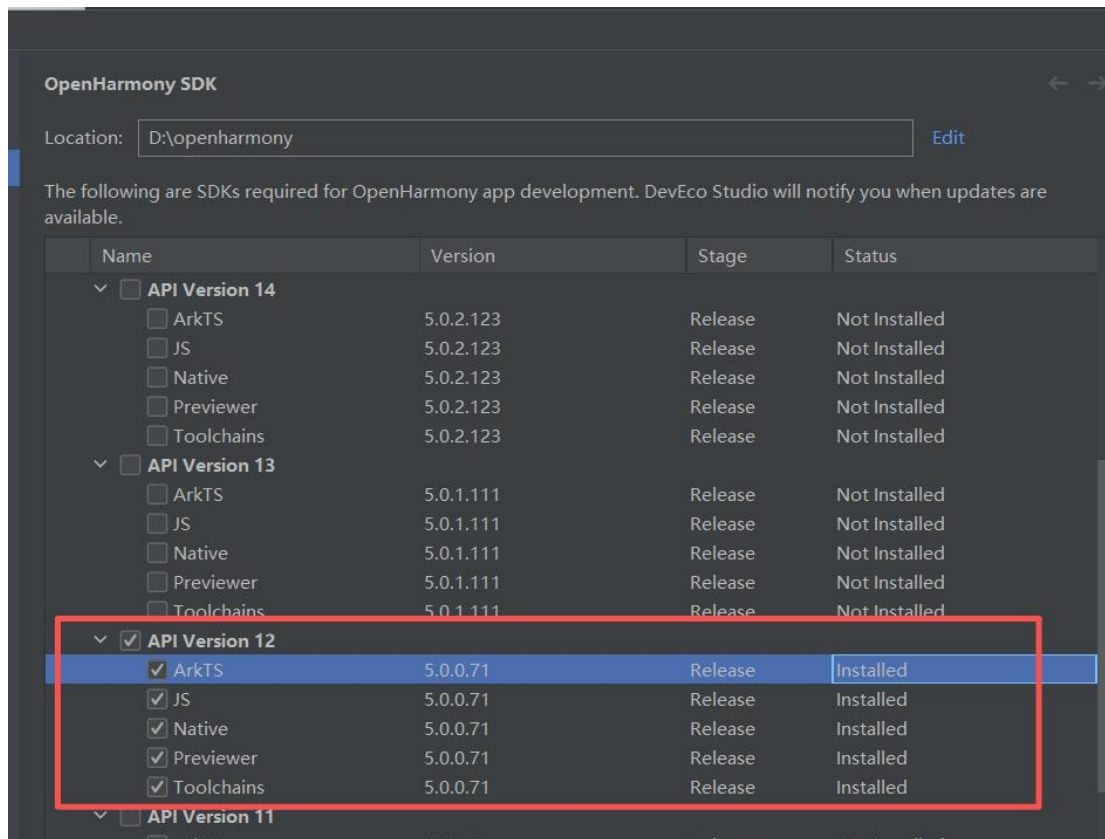
(3) 将SDK安装在任意自定义目录下（文档使用的SDK目录为D:\openharmony），



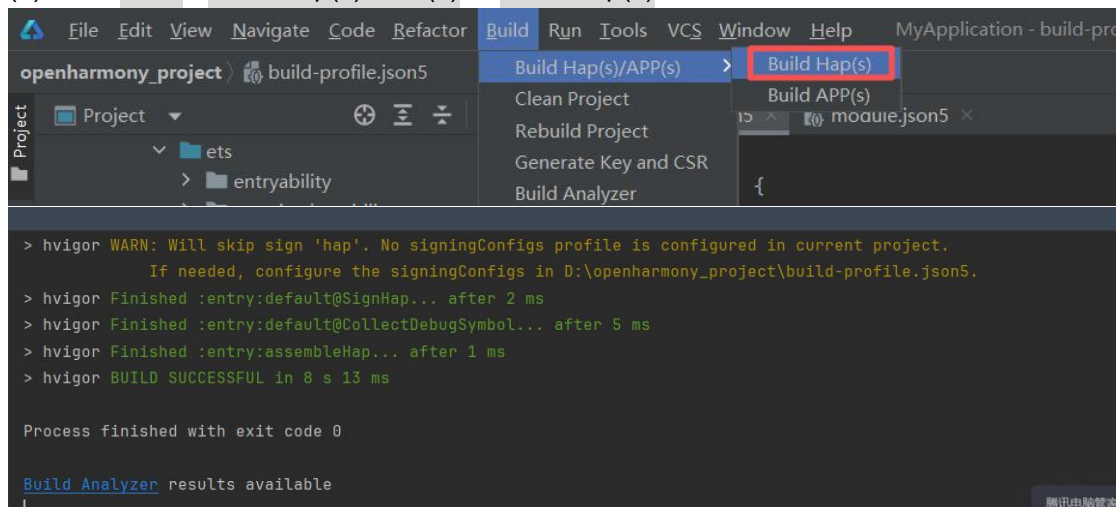
- (4) 点击Next，保持联网状态，等待SDK18下载安装完成。
- (5) **断网！！** 找到SDK目录，将[1.1.2 下载 OpenHarmony SDK for LoongArch](#) 解压的12文件夹复制到D:\openharmony下。（不断网的话，官方x86 SDK会自动覆盖龙芯 SDK）



如下图，已经识别到龙芯SDK12，如果不按照以上步骤可能会导致龙芯SDK被覆盖而无法正常编译。



(6) 点击Build->Build Hap(s)/APP(s) -> Build Hap(s)



编译成功，至此，DevEco Studio for loongarch基础环境搭建完成。

1.3 应用调试和部署

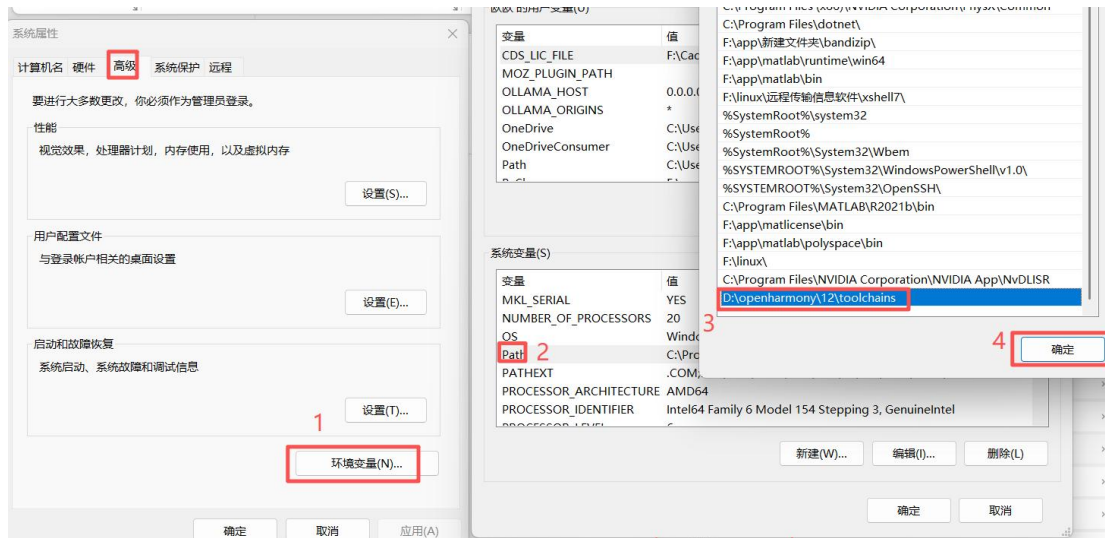
建议参考官方文档[hdc-调试命令-调测调优-系统 - 华为HarmonyOS开发者](#)

1.3.1 hdc路径环境配置

hdc工具位于D:\openharmony\12\toolchains\hdc.exe(该路径为1.2.4配置的龙芯sdk12路径)

hdc path环境配置如图所示。打开**设置**，选择**系统->系统信息->找到“高级系统设置”->高级->环境设置->系统环境变量中的path->输入“D:\openharmony\12\toolchains\”->一路“确定”**





接着打开cmd窗口，输入hdc。

```

PS C:\Users\欧文> hdc
Unknown operation command...

OpenHarmony device connector(HDC) ...

-----global commands:-----
-h/help [verbose]          - Print hdc help, 'verbose' for more other cmds
-v/version                 - Print hdc version
-t connectkey              - Use device with given connect key

-----component commands:-----
session commands(on server):
list targets [-v]          - List all devices status, -v for detail
start [-r]                 - Start server. If with '-r', will be restart server
kill [-r]                  - Kill server. If with '-r', will be restart server

service commands(on daemon):
target mount               - Set /system /vendor partition read-write
target boot [-bootloader|-recovery] - Reboot the device or boot into bootloader\recovery.
target boot [MODE]         - Reboot the into MODE.
smode [-r]                 - Restart daemon with root permissions, '-r' to cancel root
                             permissions
tmode usb                  - Reboot the device, listening on USB
tmode port [port]          - Reboot the device, listening on TCP port
  
```

可以看到，hdc环境配置完成。

1.3.2 hdc网络调试设备

(1) 开发板准备工作

首先需要确保windows主机和开发板处于同一个网段下，保证能ping通。可以使用wifi连接，也可以通过网线将开发板和主机连接起来。

接下来以将开发板和主机通过**网线连接**为例，**再通过串口进入开发板命令行终端**，修改开发板的ip地址为**192.168.1.10**（ip地址随意），windows主机ip地址为**192.168.1.2**，可以使用**自定ip地址**，确保两者能ping通即可。

```

# ifconfig eth0 192.168.1.10
# ifconfig
lo          Link encap:Local Loopback
           inet addr:127.0.0.1  Mask:255.0.0.0
           inet6 addr: ::1/128 Scope: Host
           UP LOOPBACK RUNNING  MTU:65536  Metric:1
           RX packets:0 errors:0 dropped:0 overruns:0 frame:0
           TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:0 TX bytes:0

eth0       Link encap:Ethernet  HWaddr 14:d3:4b:1c:4c:28  Driver dwmac-loongson-pci
           inet addr:192.168.1.10  Bcast:192.168.1.255  Mask:255.255.255.0
           inet6 addr: fe80::16d3:4bff:fe1c:4c28/64 Scope: Link
           UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
           RX packets:105 errors:0 dropped:0 overruns:0 frame:0
           TX packets:191 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:14590 TX bytes:50854
           Interrupt:47

eth1       Link encap:Ethernet  HWaddr 16:d3:4b:1c:4c:28  Driver dwmac-loongson-pci
           UP BROADCAST MULTICAST  MTU:1500  Metric:1
           RX packets:0 errors:0 dropped:0 overruns:0 frame:0
           TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:0 TX bytes:0
           Interrupt:64

```

配置hdc模式和端口

\$ param set persist.hdc.mode tcp #设置hdc以tcp的方式连接

\$ param set persist.hdc.port 8710 #port也可是其他的端口号

重启hdc服务

\$ service_control stop hdcd #关闭hdc服务

\$ service_control start hdcd #开启hdc服务

查看hdc状态

\$ param get | grep hdc

```

# param set persist.hdc.mode tcp
Set parameter persist.hdc.mode tcp success
# param set persist.hdc.port 8710
Set parameter persist.hdc.port 8710 success
# service_control stop hdcd
# service_control start hdcd
# param get | grep hdc
sys.usb.config = hdc
sys.usb.ffs.ready.hdc = 0
sys.usb.state = hdc
const.hdc.version = Ver: 3.0.0b
const.hdc.secure = 0
persist.sys.usb.config = hdc
persist.hdc.mode = tcp
persist.hdc.mode.usb = enable
persist.hdc.mode.tcp = enable
persist.hdc.mode.uart = disable
persist.hdc.port = 8710
persist.hdc.root = 1
persist.hdc.jdwp = 0
startup.servicectl.hdcd = 2
startup.servicectl.hdcd.pid = 4585
#

```

如果版本是5.0 release及以后，则参考以下hdc配置

配置hdc模式和端口

```
$ param set persist.hdc.mode tcp #设置hdc以tcp的方式连接
```

```
$ param set persist.hdc.port 8710 #port也可是其他的端口号
```

5.0release新增配置，必须设置：

```
$ param set persist.hdc.mode.tcp enable
```

```
$ param set persist.hdc.root 1
```

```
$ param set const.security.developermode.state true #开启开发者模式
```

重启hdc服务

```
$ service_control stop hdcd #关闭hdc服务
```

```
$ service_control start hdcd #开启hdc服务
```

查看hdc状态

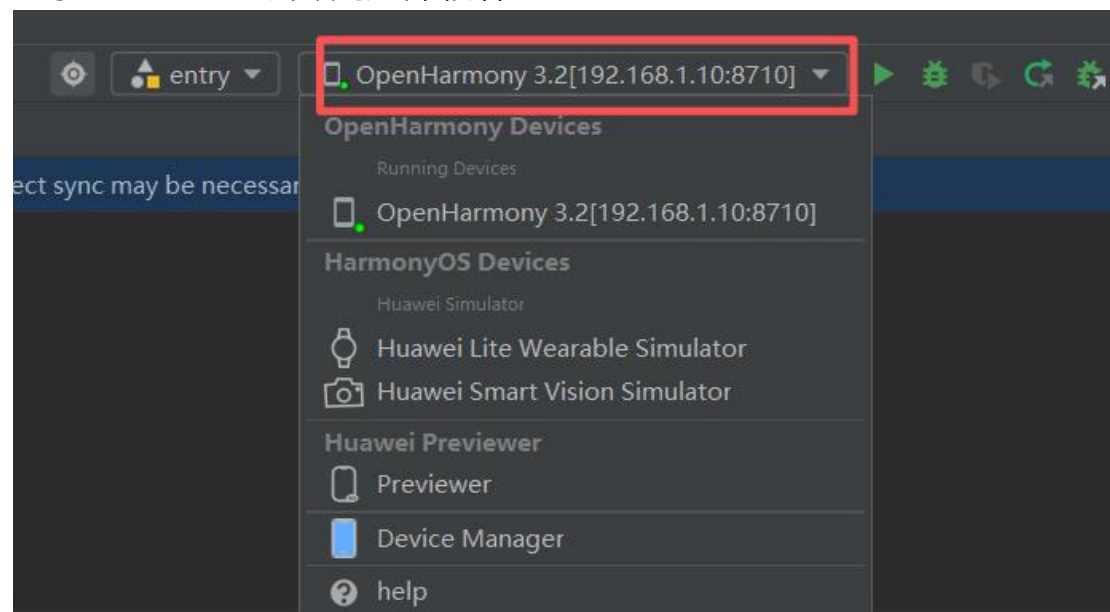
```
$ param get | grep hdc
```

(2) Windows主机连接OpenHarmony设备

默认端口号为8710。hdc tconn 192.168.1.10:8710

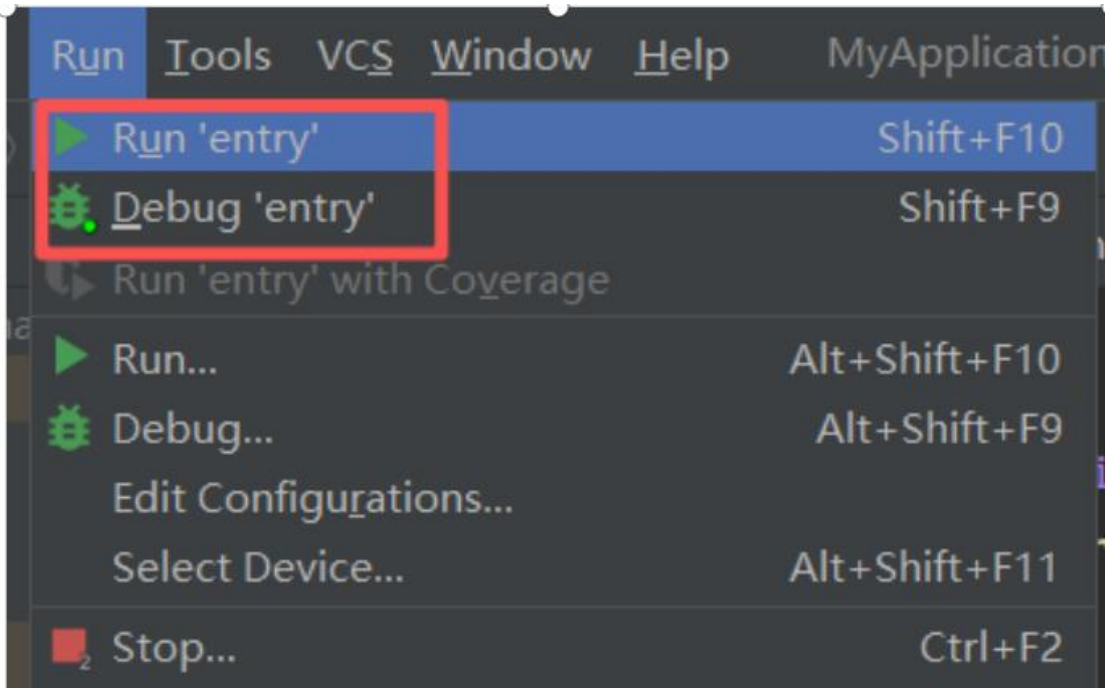
```
PS C:\Users\19643> hdc tconn 192.168.1.10:8710
Connect OK
PS C:\Users\19643> hdc list targets -v
192.168.1.10:8710      TCP      Connected      localhost      hdc
```

此时DevEco Studio可以看到如下图内容：

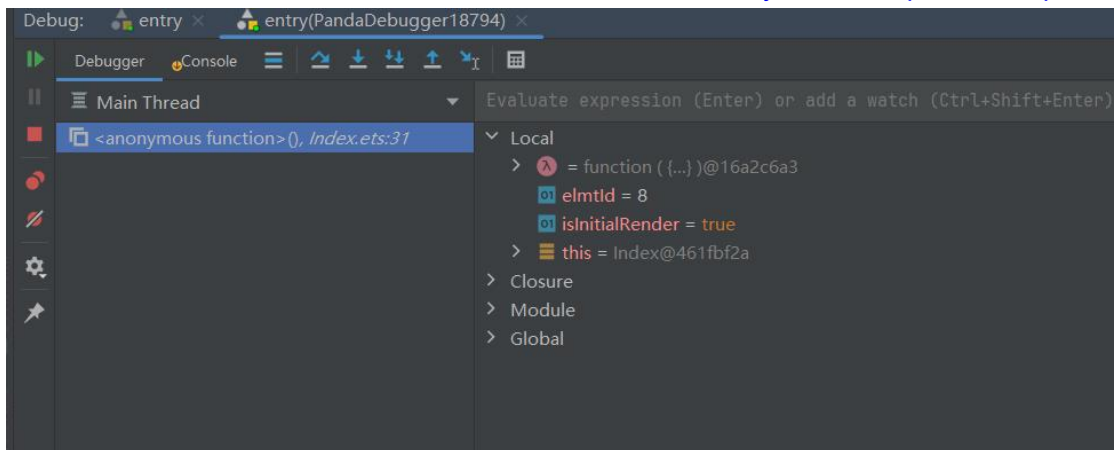


(3) 工程调试

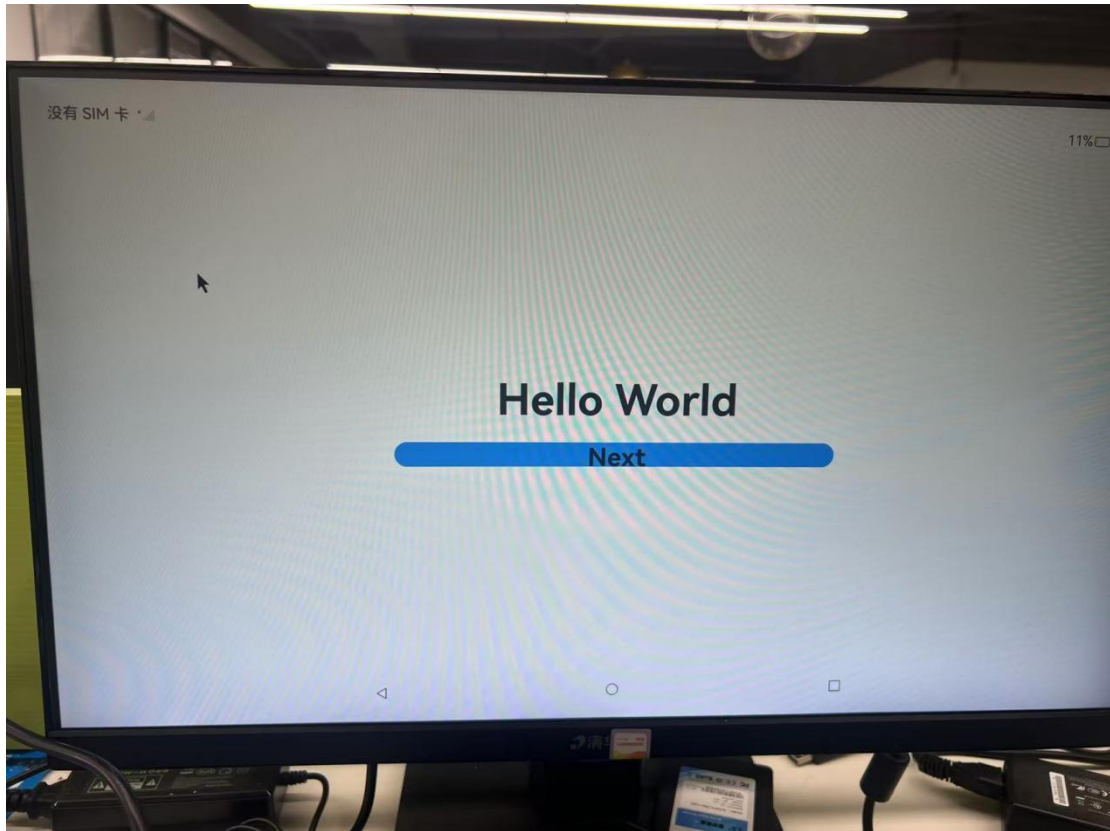
DevEco Studio选项栏 点击RUN -> Run 'entry' /Debug 'entry'



可以看到调试信息，工程例子参考：《从零开始开发 HarmonyOS 应用（ETS 语言）》

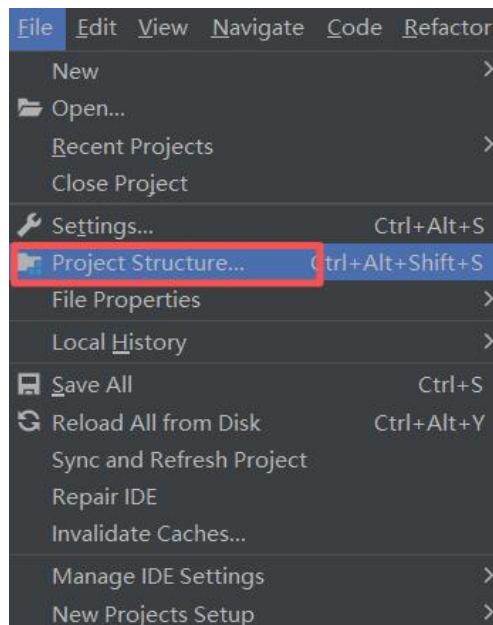


如下图，顺利运行应用。



1.3.3 应用部署

- (1) 应用安装前必须进行签名。使用DevEco Studio自动签名功能，进入File > Project Structure... > Project > Signing Configs界面，勾选“Automatically generate signature”，即可完成签名。



(2) 在cmd窗口上执行 `hdc install` 软件绝对路径，安装完成后，会在鸿蒙设备桌面显示对应的软件。（为方便测试部署功能，可直接使用网盘中的entry-default-signed.hap验证）

```
C:\Users\欧欧> hdc install D:\openharmony_project\entry\build\default\outputs\default\entry-default-signed.hap
[Info]App install path:D:\openharmony_project\entry\build\default\outputs\default\entry-default-signed.hap, queuesize:0,
msg:install bundle successfully.
AppMod finish
```