

开发Hi3516第一个应用程序示例

- [获取源码](#)
- [修改应用程序](#)
- [编译](#)
- [烧录](#)
- [镜像运行](#)
- [执行应用程序](#)

本节指导开发者在单板上运行第一个应用程序，其中包括修改应用程序、编译、烧写、运行等步骤，最终输出“Hello OHOS!”。

获取源码

开发者需要在Linux服务器上下载并解压一套源代码，获取Hi3516源码 ([下载链接](#))。更多源码获取方式，请见[源码获取](#)。

修改应用程序

源码目录applications/sample/camera/app/src内**helloworld.c**代码如下所示，用户可以自定义修改打印内容（例如：修改OHOS为World）。当前应用程序可支持标准C及C++的代码开发。

```
#include <stdio.h>
#include "los_sample.h"

int main(int argc, char **argv)
{
    printf("\n*****\n");
    printf("\n\t\tHello OHOS!\n");
    printf("\n*****\n\n");

    LOS_Sample(g_num);

    return 0;
}
```

编译

在linux服务器上，进入源码包根目录，目录内存放有build.py编译脚本，执行如下脚本编译源码包。结果文件生成在out/ipcamera_hi3516dv300目录下。

```
python build.py ipcamera_hi3516dv300 -b debug
```

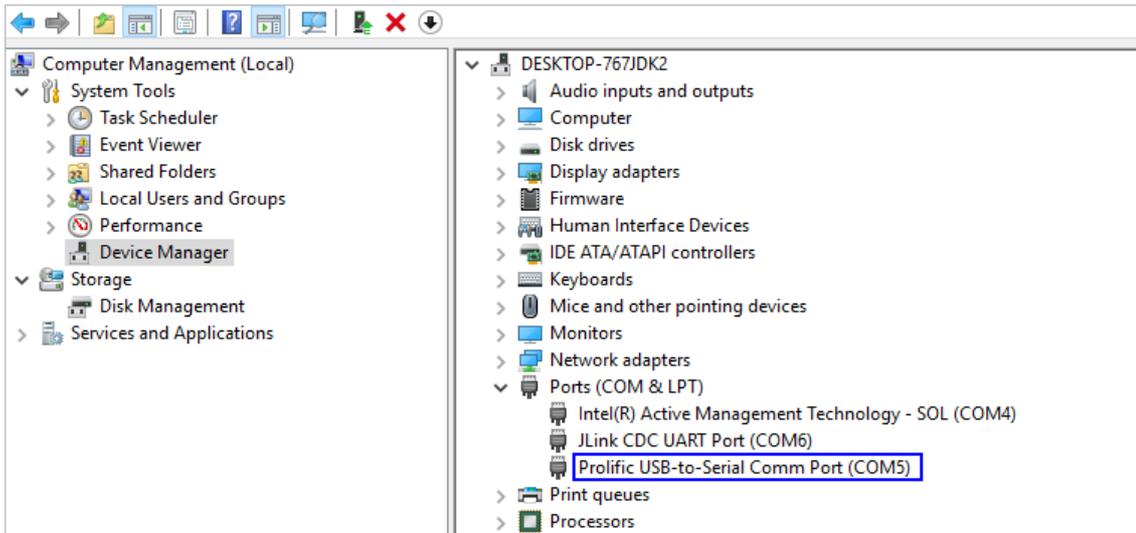
烧录

Hi3516开发板的代码烧录支持USB烧录、网口烧录和串口烧录三种方式。此处仅以网口烧录为例进行说明。

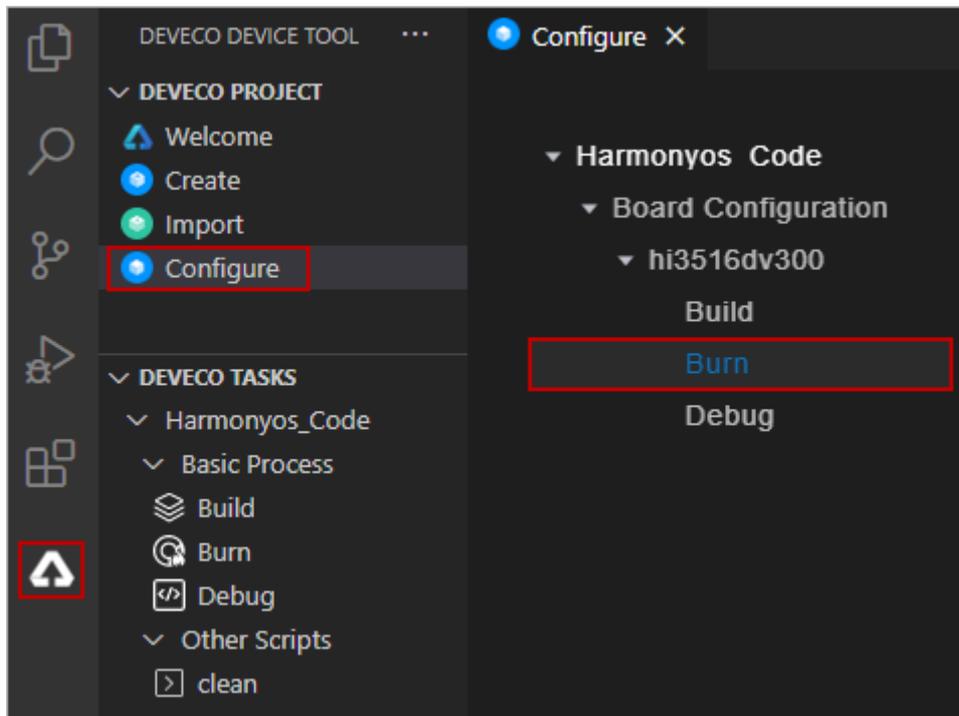
1. 请连接好电脑和待烧录开发板，以Hi3516DV300为例，需要同时连接串口、网口和电源，具体可参考[Hi3516开发板介绍](#)。
2. 打开电脑的设备管理器，查看并记录对应的串口号。

说明：

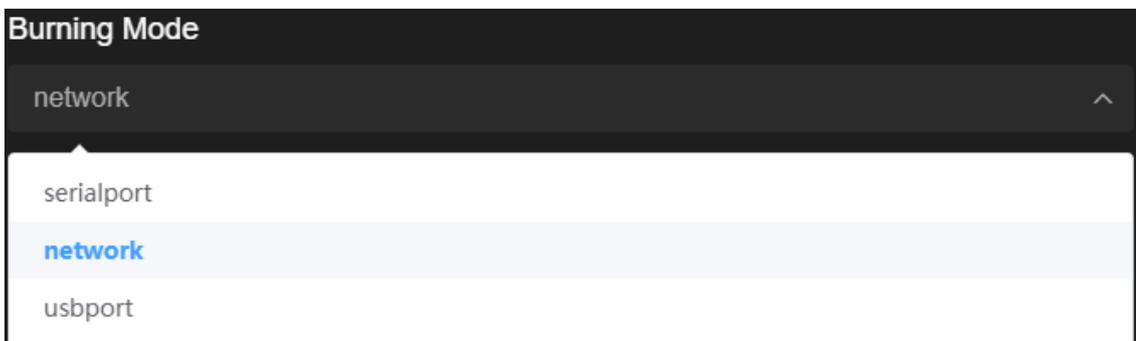
如果对应的串口异常，请根据[Hi3516/Hi3518系列开发板串口驱动安装](#)安装USB转串口的驱动程序。



3. 点击“Configure > Burn”，进入烧录配置界面，设置ARM系列芯片烧录信息。



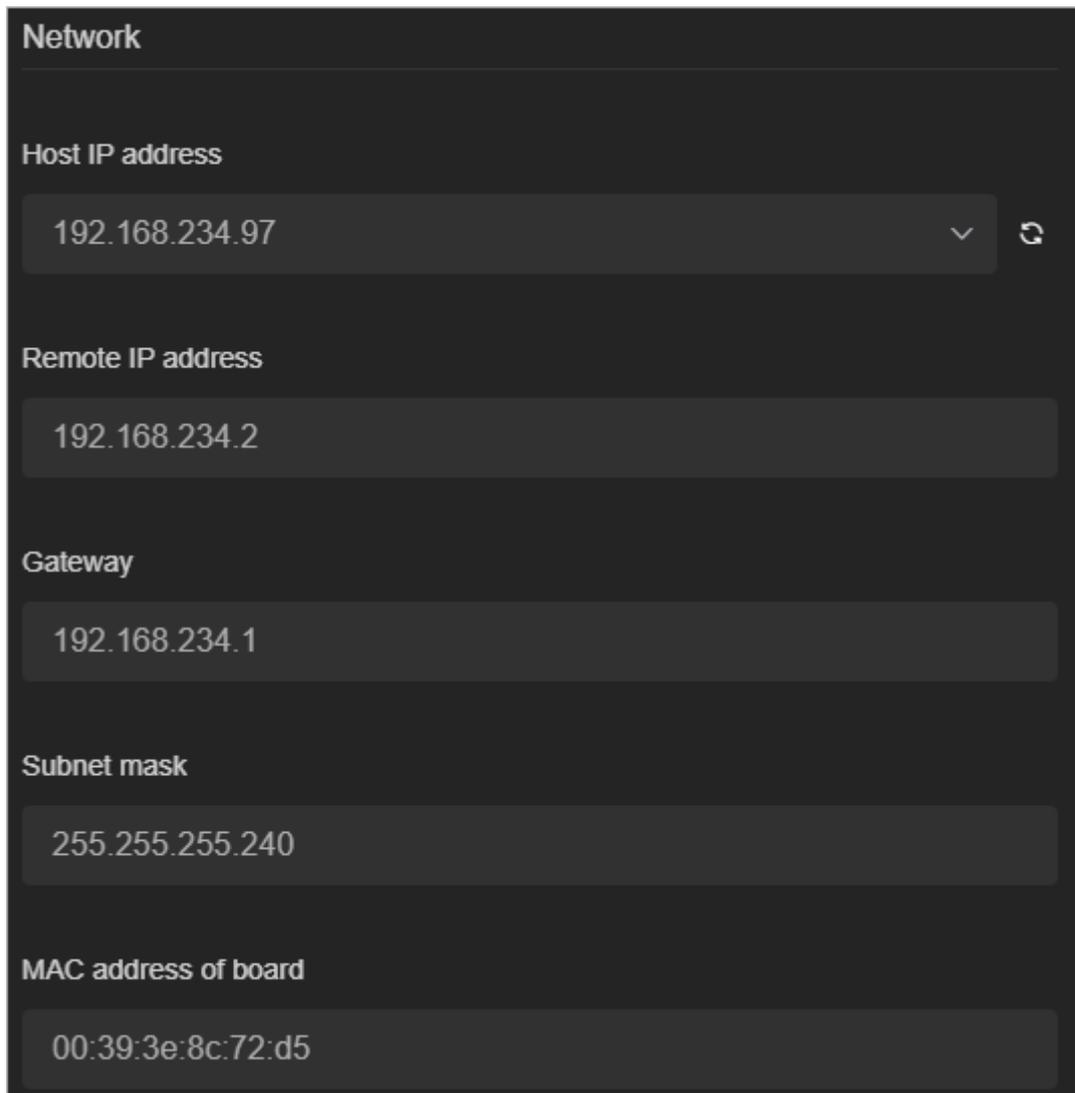
4. 选择烧录方式（Burning Mode），固定选择“network”。



5. 设置网口烧录方式的烧录参数。

- 设置Port number，请选择2中查询的串口号。
- 设置Baud rate和Data bits参数，已根据开发板进行适配，保持默认值即可。
- 设置网络连接信息：

- Host IP address: 选择对应本地PC端的IP地址, 可通过命令行工具中输入“ipconfig”命令查看。
- Remote IP address: 设置开发板的IP地址, 工具自动设置。
- Gateway: 对应开发板的网关, 工具自动设置。
- Subnet mask: 子网掩码, 工具自动设置。
- MAC address of board: 开发板的物理地址, 请手动设置。



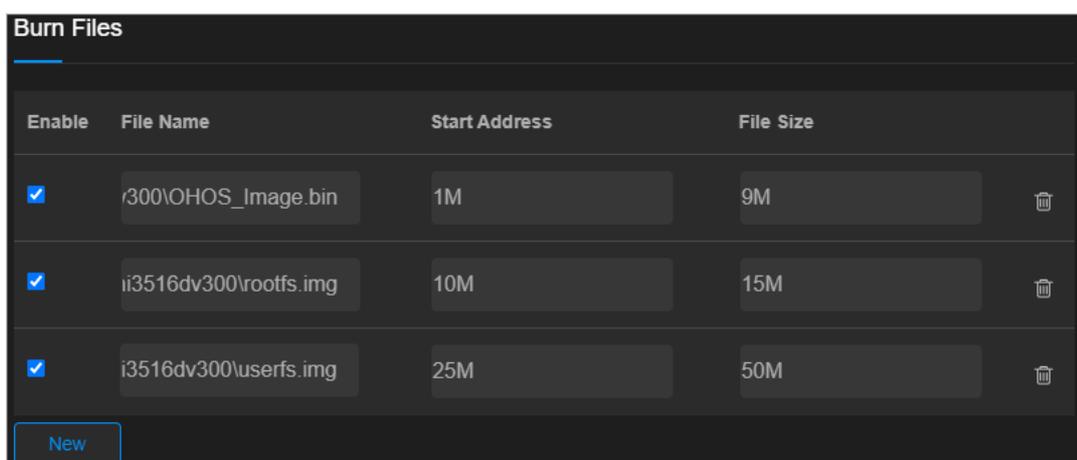
The image shows a dark-themed network configuration window titled "Network". It contains five input fields, each with a label and a value:

- Host IP address:** 192.168.234.97 (with a dropdown arrow and a refresh icon)
- Remote IP address:** 192.168.234.2
- Gateway:** 192.168.234.1
- Subnet mask:** 255.255.255.240
- MAC address of board:** 00:39:3e:8c:72:d5

- 设置器件类型 (Memory Type) , 包括spi nor、spi nand和emmc, 具体设置规则与硬件接收类型相关, 请查阅随硬件发货的硬件说明书。例如Hi3516DV300固定设置为emmc。
- 设置烧录文件信息: 请根据如下表格设置烧录文件, 网口烧录方式, 不支持烧录U-Boot文件。

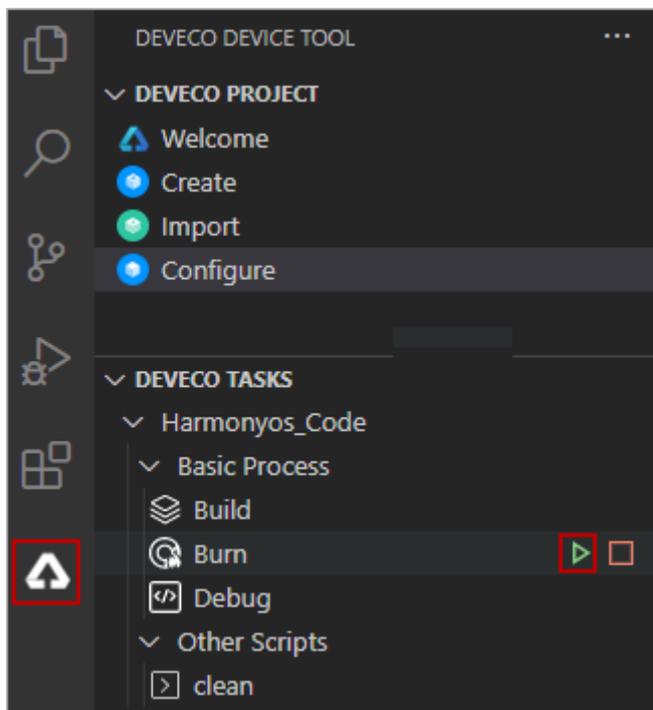
待烧录文件 (File Name)	开始地址 (Start Address)	Flash地址块大小 (File Size)	是否烧录 (Enable)
OH OS_Imag e.bin	1M	9M	第一次使用开发板烧录时, 必须同时烧录; 在后续烧录过程中, 如果修改了内核和驱动相关内容, 才需要烧录。
root fs  .img	10 M	15 M	第一次使用开发板烧录时, 必须同时烧录; 在后续烧录过

待烧录文件 (File Name)	开始地址 (Start Address)	Flash地址块大小 (File Size)	是否烧录 (Enable)
userfs.img	25M	50M	程 中, 如果 这两个 文件未 做修改, 可以不 烧录。 建议 每次 烧录时, 都烧录 这两个 文件。



6. 修改了相关配置后，请点击最下方的“Save”进行保存。

7. 在DevEco Device Tool中，点击Burn后的  按钮开始烧录。



8. 请在15秒内手动重启开发板(下电再上电)。
9. 等待烧录完成, 当控制台输出如下信息时, 表示烧录成功。

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

Open serial port: COM5
sf probe 0;sf erase 0x100000 0x300000;sf write 0x40000000 0x100000 0x300000;
Erasing at 0x400000 -- 100% complete.
SF: 3145728 bytes @ 0x100000 Erased: OK
device 0 offset 0x100000, size 0x300000
Writing at 0x400000 -- 100% complete.
SF: 3145728 bytes @ 0x100000 Written: OK
hisilicon #
Succeed to load and write images to flash.
```

10. 如果第一次使用开发板, 需要使用[串口工具](#)修改U-boot的bootcmd及bootargs内容, 具体可参考[Hi3516开发板镜像运行](#)。

镜像运行

1. 连接串口。



须知:

若无法连接串口, 请参考[常见问题](#)进行排查。

图 1 连接串口图

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
> Executing task: node serialterminal.js <

> The port name:
COM3
COM1
COM2
COM11
Pick the one for opening: com11
What is the Baud Rate? The default is 115200. Enter:
What is the Data Bits? The default is 8. Enter:
What is the Stop Bits? The default is 1. Enter:
Serial Info:
  The Port: com11
  The Baud Rate: 115200
  The Data Bits: 8
  The Stop Bits: 1
Serial com11 $

hisilicon #
```

1. 单击**Serial port**打开串口。
 2. 输入串口编号(按照烧录步骤中查询的串口号, 此处以com11举例), 并连续输入回车直到串口显示"hisilicon"。
 3. 单板初次启动或修改启动参数, 请进入[步骤2](#), 否则进入[步骤3](#)。
2. (单板初次启动必选) 修改U-boot的bootcmd及bootargs内容: 该步骤为固化操作, 若不修改参数只需执行一次。每次复位单板均会自动进入系统。



须知:

U-boot引导程序默认会有2秒的等待时间, 用户可使用回车打断等待并显示"hisilicon", 通过**reset**命令可再次启动系统。

表 1 U-boot修改命令

执行命令	命令解释
<pre>setenv bootcmd "mmc read 0x0 0x80000000 0x800 0x4800; go 0x80000000";</pre>	<p>读取FLASH起始地址为0x800（单位为512B，即1MB），大小为0x4800（单位为512B，即9MB）的内容到0x80000000的内存地址，该大小（9MB）与IDE中所填写OHOS_Image.bin文件大小必须相同。</p>
<pre>setenv bootargs "console=ttyAMA0,1 15200n8 root=emmc fstype=vfat rootaddr=10M rootsize=15M rw";</pre>	<p>表示设置启动参数，输出模式为串口输出，波特率为115200，数据位8，rootfs挂载于emmc器件，文件系统类型为vfat，</p> <p>“rootaddr=10M rootsize=15M rw”处对应填入rootfs.img的烧写起始位置与长度，此处与IDE中新增rootfs.img文件时所填大小必须相同。</p>
<pre>saveenv</pre>	<p>表示保存当前配置。</p>
<pre>reset</pre>	<p>表示复位单板。</p>



须知：

“go 0x80000000”为可选指令，默认配置已将该指令固化在启动参数中，单板复位后可自动启动。若想切换为手动启动，可在U-boot启动倒数阶段使用“回车”打断自动启动。

3. 输入“reset”指令并回车，重启单板，启动成功如下图，输入回车串口显示OHOS字样。

图 2 系统启动图

```
[DISPLAY I/] PrintLayerInfo: layerInfo:
[DISPLAY I/] PrintLayerInfo: type = 0
[DISPLAY I/] PrintLayerInfo: width = 960
[DISPLAY I/] PrintLayerInfo: height = 480
[DISPLAY I/] PrintLayerInfo: bpp = 16
[DISPLAY I/] PrintLayerInfo: pixFormat = 9
[DISPLAY I/] OpenGraphicLayer: open graphic layer
[DISPLAY I/] GfxInitialize: gfx initialize success
[UnRegisterDeathCallback : 959]Wrong cbId:-1.
GetInputInterface: enter
GetInputInterface: exit success
[UnRegisterDeathCallback : 959]Wrong cbId:-1.
OpenInputDevice: open /dev/input/event1 success
RegisterReportCallback: create monitor thread success
RegisterReportCallback: device1 register callback
OpenInputDevice: realpath fail
[UnRegisterDeathCallback : 959]Wrong cbId:-1.
[UnRegisterDeathCallback : 959]Wrong cbId:-1.

OHOS #

OHOS #
```

执行应用程序

根目录下，在命令行输入指令“./bin/camera_app”执行写入的demo程序，显示成功结果如下图所示。

图 3 应用程序启动图

```
OHOS #
./bin/camera_app
OHOS # ./bin/camera_app
OHOS #
*****
Hello OHOS!
*****
This is a sample: Param = 81
```