

作者：疯壳

注：文档和视频中所有的图片及代码截图皆为示意图，具体以 HarmonyOS 官网发布内容为准。

更多鸿蒙相关技术文章和资料，请关注 **HarmonyOS 社区**



HarmonyOS 应用程序开发

开发 Hi3516 第一个应用程序示例

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

1. 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");

    LOS_Sample(g_num);

    return 0;
}
```

2. 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
```

3. 烧写

h2U-boot 引导文件烧写方式

通常单板出厂时已经烧入 U-boot 引导文件，通常不需要烧写该文件，若 U-boot 引导文件因损坏而单板无法启动，可按照本节描述烧写 U-boot 引导文件。

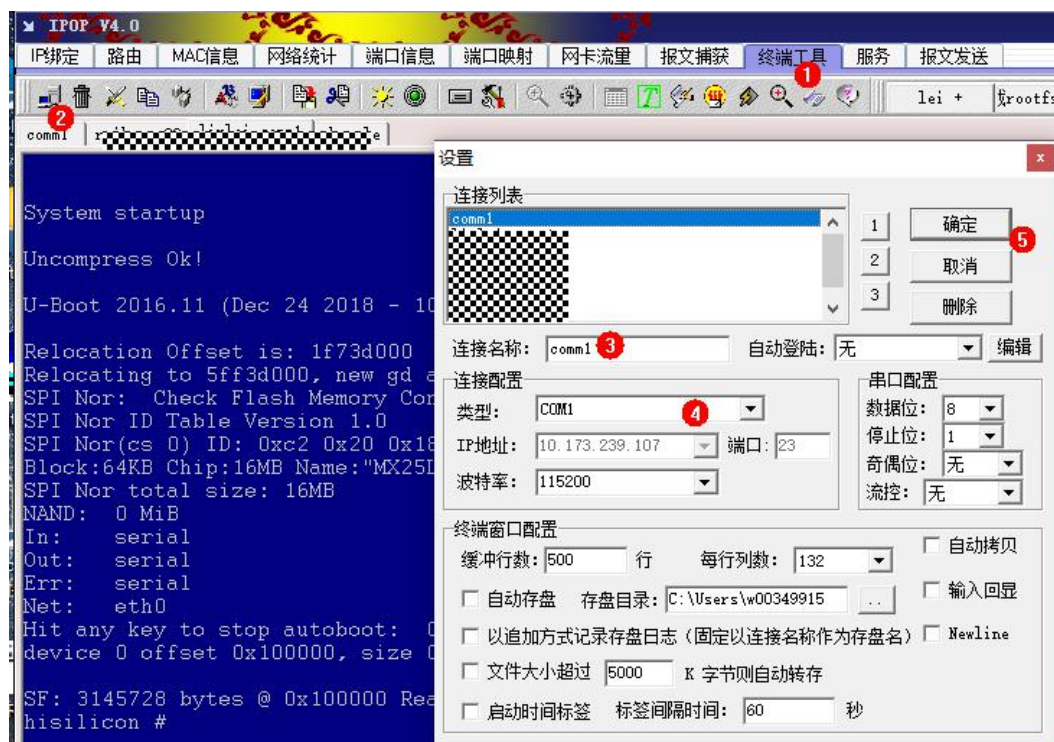
步骤 1 获取引导文件 U-boot。

说明

Hi3518EV300 系列单板的 U-boot 文件请在开源包中获取，示例路径为
vendor\hisi\camera\hi3518ev300\Hi3516EV200_SDK_V6.0.0.1\osdrv\pub\u-boot-hi3516ev200.bin，Hi3518EV300 单板可用。

步骤 2 连接串口，串口工具配置步骤。

图 1-1 串口连接流程图



1. 打开 IPOP 工具，点击终端工具。
2. 点击图标，弹出对话框。
3. 输入连接名称“comm1”。
4. 选择类型“COM1”。
5. 点击确认保存。
6. 界面将增加“comm1”页签，输入回车后，串口显示“hisilicon #”。
7. 点击标记 2 旁的“垃圾桶”图标，断开串口。

说明

Windows 工作台通常含有多个串口，若 COM1 连接无反应，可参考问题 1:HiTool 工具烧写时上报如下错误。排查。

步骤 3 使用 HiTool 工具按照标号选择 U-boot 烧写选项，点击烧写按钮。

图 1-2 HiTool 工具 U-boot 烧写步骤图



步骤 4 断开单板电源，并重新上电，烧写完成后，连接串口，如下图所示。

图 1-3 HiTool 工具 U-boot 烧写完成串口显示



---结束

h2 内核镜像烧写方式

镜像烧写方法分为串口、USB 口、以太网口烧写，开发者均可按实际情况使用。

- Hi3518EV300 单板无网口，可使用串口烧写或 USB 烧写。
- Hi3516EV200 单板可使用串口、以太网烧写，注意单板类型不同，烧写芯片类型会不同。

串口烧写方式

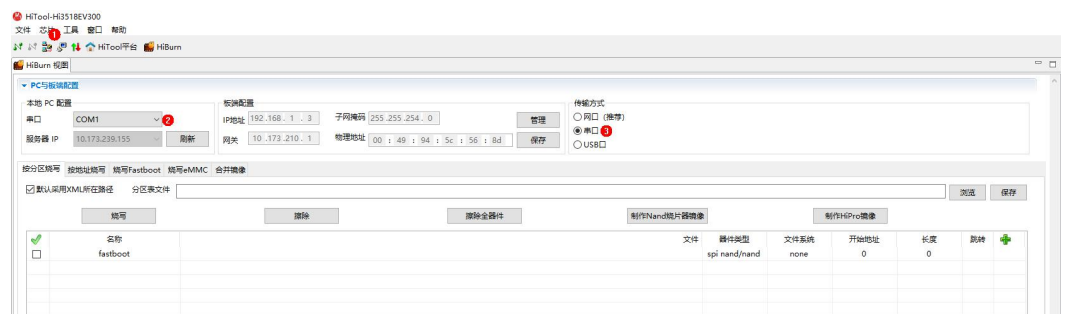
步骤 1 打开 HiTool 工具，按照标号顺序选择芯片类型"Hi3518EV300"，“本地 PC 配置”内选择“COM1”，“传输方式”内选择“串口”。



注意

请按实际情况选择芯片及串口号，否则无法执行烧写流程。

图 1-4 HiTool 工具串口配置步骤图



步骤 2 烧写：新增编译生成的 harmony_Image.bin 文件与 rootfs.img 文件，请仔细确认烧入文件的文件路径、器件类型、开始地址、长度。请按文件实际大小（向上取整，最小单位为 MB），修改相关参数，最后点击烧写并上电单板（已上电状态需要重新下电后再上电）。

图 1-5 HiTool 工具烧写步骤图



注意

标号 2 与 3 为 harmony_Image.bin 文件及 rootfs.img 文件添加时必须填写参数内容，请务必按实际大小填写。

步骤 3 烧写结果显示如下图所示。

图 1-6 HiTool 工具烧写成功图例



----结束

USB 烧写方式

步骤 4 安装 USB 驱动。

请参考发布包中上海海思 U-boot 配置文档《HiBurn 工具使用指南》，安装 USB 驱动，文档存放于 ReleaseDoc 目录内。

图 1-7 USB 驱动安装成功图



步骤 5 确认 Hi3518EV300 单板内含有 USB 下载功能的 U-boot 版本，若不支持，请将上海海思提供的最新 U-boot 文件烧入板中。

在 U-boot 界面输入"help"命令并回车，查看结果中是否存在 usbtftp 命令。

图 1-8 确认 U-boot 支持 USB 下载

```
usb      - USB sub-system
usbboot  - boot from USB device
usbtftp  - download or upload image using USB protocol
version  - print monitor, compiler and linker version
hisilicon #
```

步骤 6 按标号，确认如下内容。

图 1-9 USB 口烧写配置图

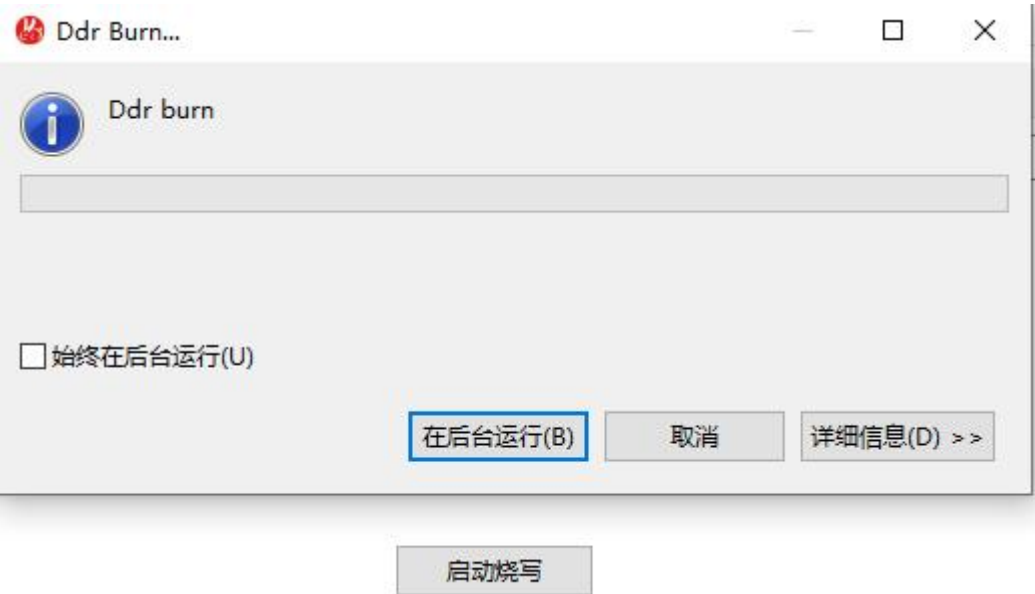


8. 选择芯片标号为“Hi3518EV300”。
9. 选择"DDR 烧写"。

- 传输方式选择"USB 口"
- 选择需要烧写的文件路径(路径包含 harmony_Image.bin 和 rootfs.img 文件)。
- 点击“启动烧写”。

步骤 7 启动烧写后，弹出界面，等待单板侧烧入。

图 1-10 烧写开始图



步骤 8 单板侧输入命令烧写 harmony_Image.bin 及 rootfs.img 文件，文件大小参数请参考“串口烧写”步骤 2。

表 8-1 USB 口烧写命令

执行命令	sf probe 0; mw.b 40000000 ff a00000; usbtftp 40000000 rootfs.img; sf erase 600000 a00000; sf write 40000000 600000 a00000; mw.b 40000000 ff 500000;usbtftp 0x40000000 harmony_Image.bin; sf erase 100000 500000; sf write 40000000 100000 500000;
命令解释	<p>命令原理，使用 USB 口将文件读取到内存，再从内存中读取写入 FLASH 中。</p> <p>sf probe 0;表示选择 FLASH 器件 0。</p> <p>写入 rootf.img 文件，mw.b 40000000 ff a00000，清空 0x40000000 开始，大小为 0xa00000 字节的空间。usbtftp 40000000 rootfs.img 将 rootfs.img 文件写入 0x40000000 的内存地址。sf erase 600000 a00000; sf write 40000000 600000 a00000; 擦除 0x600000 开始的大小为 0xa00000 字节的 FLASH 空间，并将内存中起始地址为 0x40000000，大小为 0xa00000 字节的数据复制到 0x600000 起始地址，大小为 0xa00000 字节的 FLASH 空间上。</p> <p>写入 harmony_Image.bin 文件，mw.b 40000000 ff 500000，清空 0x40000000 开始，大小为 0x500000 字节的空间。usbtftp 40000000 harmony_Image.bin 将 rootfs.img 文件写入 0x40000000 的内存地址。sf erase 100000 500000; sf write 40000000 100000 500000; 擦除 0x100000 开始的大小为 0x500000 字节的 FLASH 空间，并将内存中起始地址为 0x40000000，大小为 0x500000 字节的数</p>

据复制到 0x100000 起始地址，大小为 0x500000 字节的 FLASH 空间上。

步骤 9 烧写成功后显示 **Written:OK**。

----结束

镜像运行

- 步骤 1 连接串口：前步骤烧入成功后使用串口工具 IPOP 连接单板串口，输入回车，串口将显示"**hisilicon #**"字样。
- 步骤 2 修改 U-boot 的 bootcmd 及 bootargs 内容：该步骤为固化操作，可保存执行结果，若 U-boot 重新烧入，则需要再次执行。

表 8-2 U-boot 修改参数解释

执行命令	setenv bootcmd "sf probe 0;sf read 0x40000000 0x100000 0x500000;"; setenv bootargs "console=ttyAMA0,115200n8 root=flash fstype=jffs2 rw rootaddr=6M rootsize=10M";go 0x40000000;; saveenv;
命令解释	<p>setenv bootcmd "sf probe 0;sf read 0x40000000 0x100000 0x500000;"表示选择 FLASH 器件 0，读取 FLASH 起始地址为 0x100000，大小为 0x500000 的内容到 0x40000000 的内存地址。</p> <p>setenv bootargs "console=ttyAMA0,115200n8 root=flash fstype=jffs2 rw rootaddr=6M rootsize=10M"; 表示设置 bootargs 为串口输出，rootfs 挂载在 FLASH 上，文件系统类型为 jffs2 rw，以支持可读写 JFFS2 文件系统。</p> <p>“rootaddr=6M rootsize=10M” 处对应填入实际 rootfs.img 的烧写起始位置与长度，此处应与 HiTool 新增文件时所填大小相同。</p> <p>[可选] “go 0x40000000” 默认配置已将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。</p> <p>saveenv;表示保存当前配置。</p>

- 步骤 3 若步骤 2 中使用回车打断自启动，需在 U-boot 的命令行中，输入 **“go 0x40000000”** 指令，启动内核进程，启动成功如下图。

图 1-11 系统启动图

```
load venc.ko for Hi3516EV200...OK!
load h264e.ko for Hi3516EV200...OK!
load h265e.ko for Hi3516EV200...OK!
load jpege.ko for Hi3516EV200...OK!
load ive.ko for Hi3516EV200...OK!
Do not support in hmos fun[osal_register_reboot_notifier]
insert audio
spi bus init ...
i2c bus init ...
load mipi_rx driver successful!
sensor i2c init OK.
load tde.ko for Hi3516EV200...OK!
Load hifb.ko OK!
SDK init ok...
wlan init ...
[1929][I:314]compiling time:Jun  6 2020 18:31:13
[1932][I:452]hi_wifi_plat_init vap_num[1], user_num[4]
app event id:0
sdio probe:pull up power on gpio
no sdio device
uhub0: 1 port with 1 removable, self powered
Link is Up - 100Mbps/Full
<3>[ERROR]:osal_sdio_host.c:osal_sdio_func_probe:2655:sdio enum timeout, reason[probe timeout]

sdio probe:pull down power on gpio
[11948][E:1373]hcc_host_init:: sdio_probe failed!
no sdio device
sdio probe:pull down power on gpio
[11957][W:631]hcc_host_init fail![1]
[11960][I:375]hi_wifi_host_init: hcc_hmac_init return error code: 1
[11966][E:339]wifi host initialize fail.
HMOS #
HMOS #
```

步骤 4 根目录下，在命令行输入指令“`./usr/bin/helloworld`”执行写入的 demo 程序，显示成功结果如下图所示，该文件生成请参考应用程序编译。

图 1-12 应用程序启动图

```
HMOS # ./usr/bin/helloworld
HMOS #
*****
                        Hello HarmonyOS!
*****
```

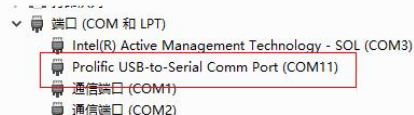
----结束

网络烧写方式

该方法只适合支持网口的单板（例如 Hi3516DV300），且 PC 主机必须与单板用网线连接并配置在于同一网络中。

1. 安装USB转串口驱动，并获取串口号。

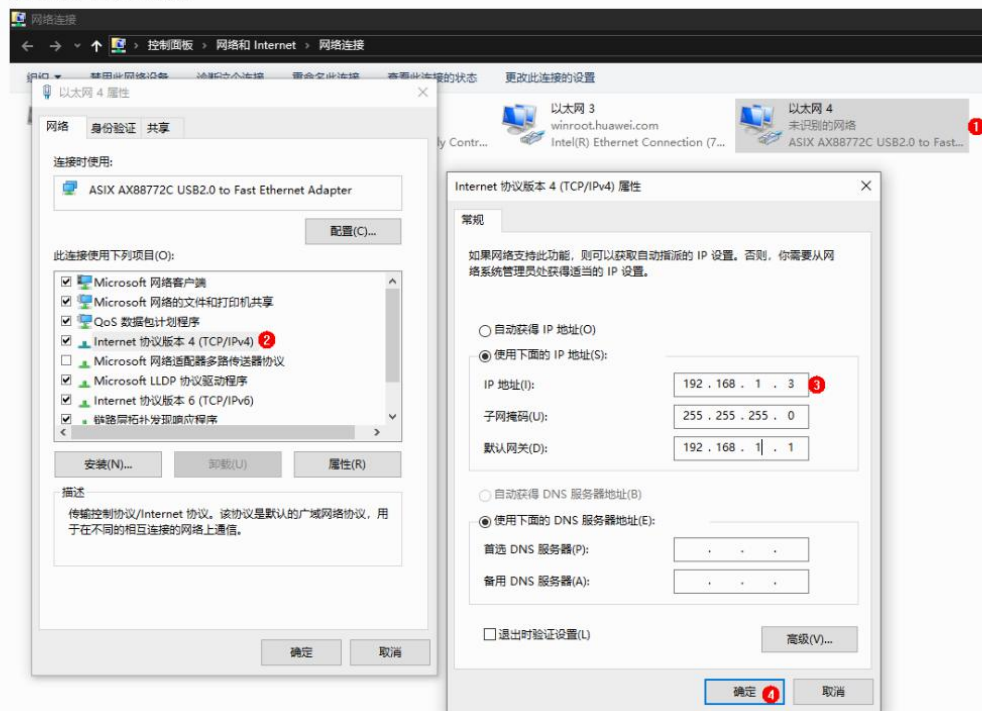
图1 驱动安装成功图



- a. 为单板上电，并将单板串口线连接Windows工作台。
 - b. 安装驱动，驱动[获取链接](#)。
 - c. 打开电脑的设备管理器，查看并记录“Prolific USB-to-Serial Comm Port” 串口号，此处为COM11。
驱动安装成功后，若设备图标存在警示图标，请右键点击设备并卸载驱动后，重新安装驱动，并按提示重启电脑。
2. Windows工作台为单板互联网口增加192.168.1.3的IP地址，增加方法如下。

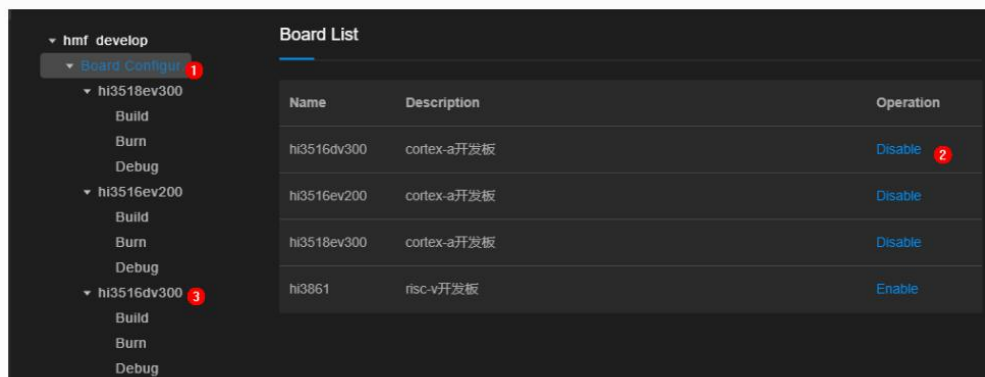
2. Windows工作台中为单板互联网口增加192.168.1.3的IP地址，增加方法如下。

图2 Windows主机IP增加图例



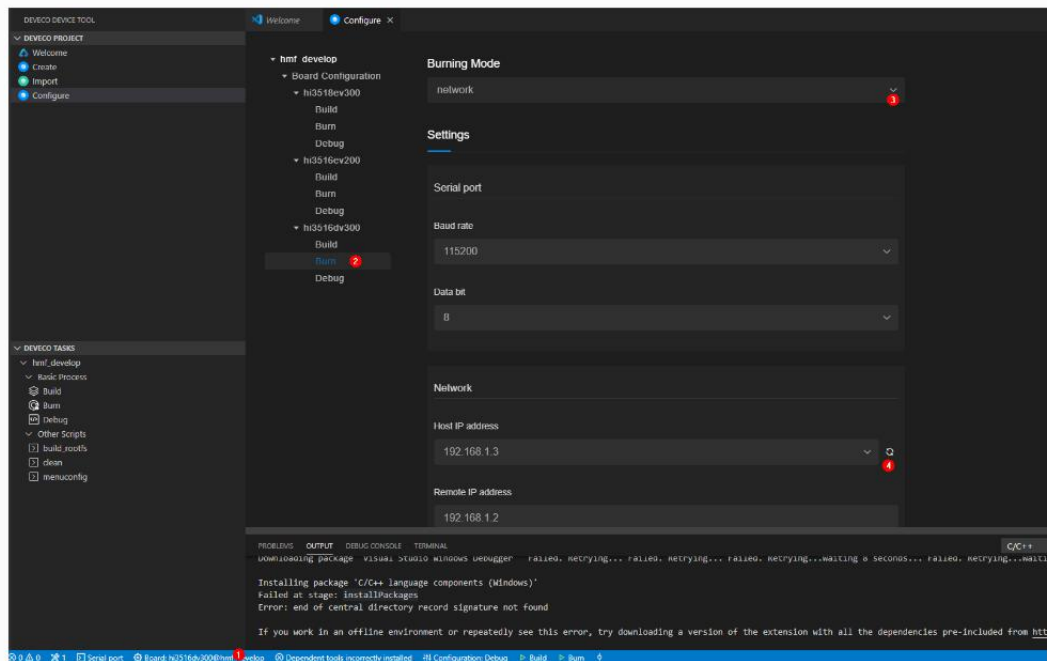
- 控制面板->网络和Internet->网络连接中找到与单板连接的网卡，右键打开属性。
 - 选择Internet协议版本4（TCP/IPv4）。
 - 按图配置IP地址和网关。
 - 点击“确定”完成配置保存。
3. 单板列表增加hi3516dv300，依次选择Board Configure，enable单板，自动新增单板配置表。

图3 增加hi3516dv300单板



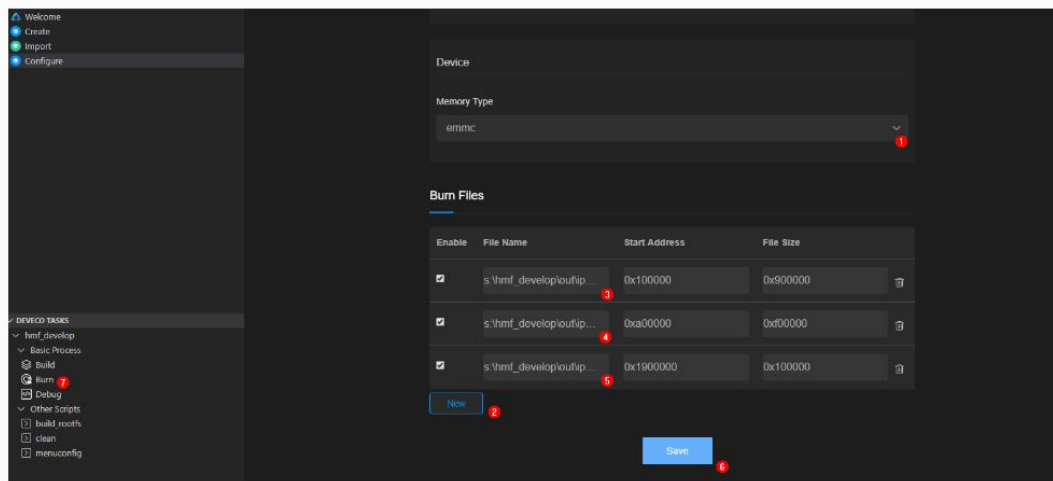
4. 打开IDE工具，按下图标号顺序配置网络烧写内容。

图4 IDE工具网络配置图例



- a. 单板类型选择Hi3516单板。
 - b. 单击"Burn"。
 - c. "Burning Mode"选择为"network"。
 - d. "Host IP Address"点击刷新后，在下拉框中选择步骤2中配置的IP地址"192.168.1.3"。
5. 选择需要烧写的FLASH芯片存储类型及烧写地址。

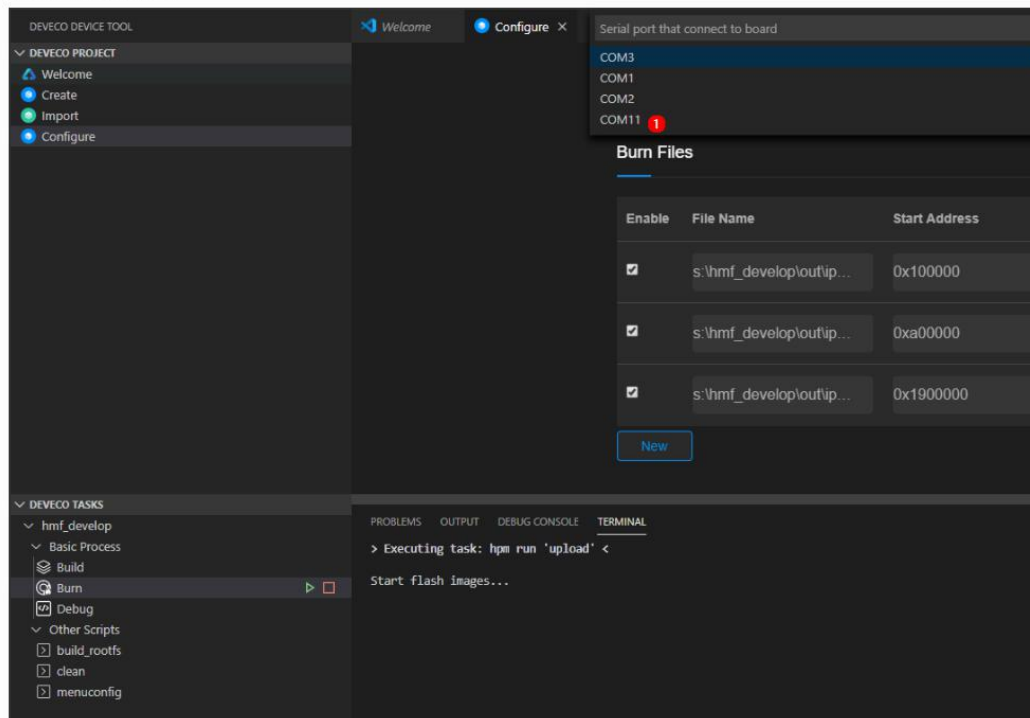
图5 烧写文件参数配置图



- "Memory Type"选择"emmc"。
- 点击"New"，新增至三个文件，并在文件路径中依次填入OHOS_Image.bin、rootfs.img、userfs.img文件，起始地址及文件长度按图填写，文件从out/ipcamera_hi3516dv300目录下获取。
- 点击"Save"保存。
- 点击左侧"Burn"开始烧写。

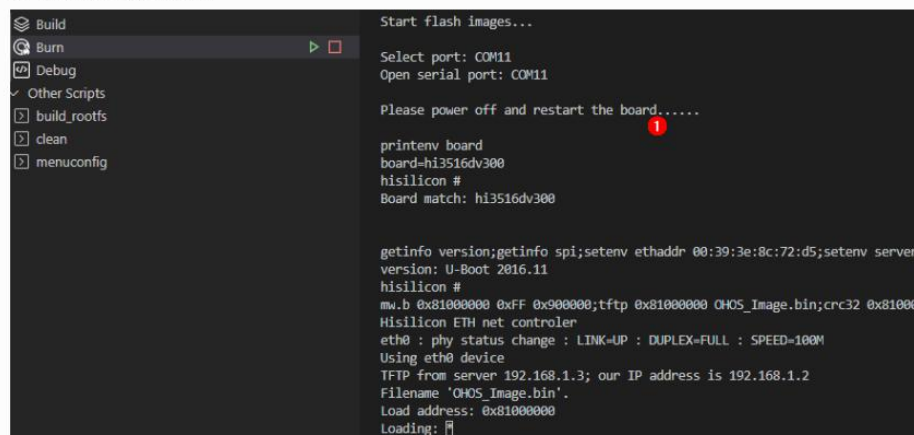
6. 上方输入框弹出下拉框后，选择串口号，例如COM11。

图6 选择与单板连接的串口



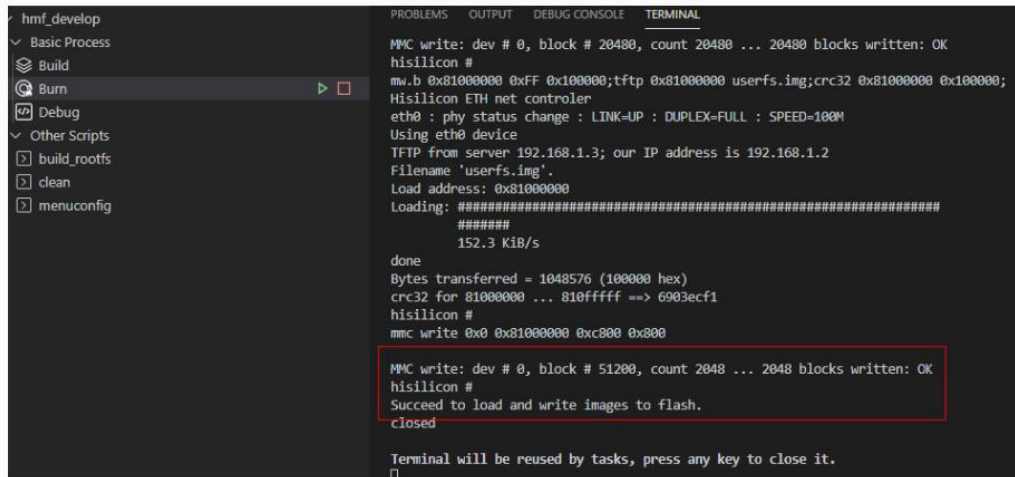
7. 烧录开始，若出现提示，需要手动重启单板(下电再上电)。

图7 提示下电并重新为单板上电



8. 烧写完成。

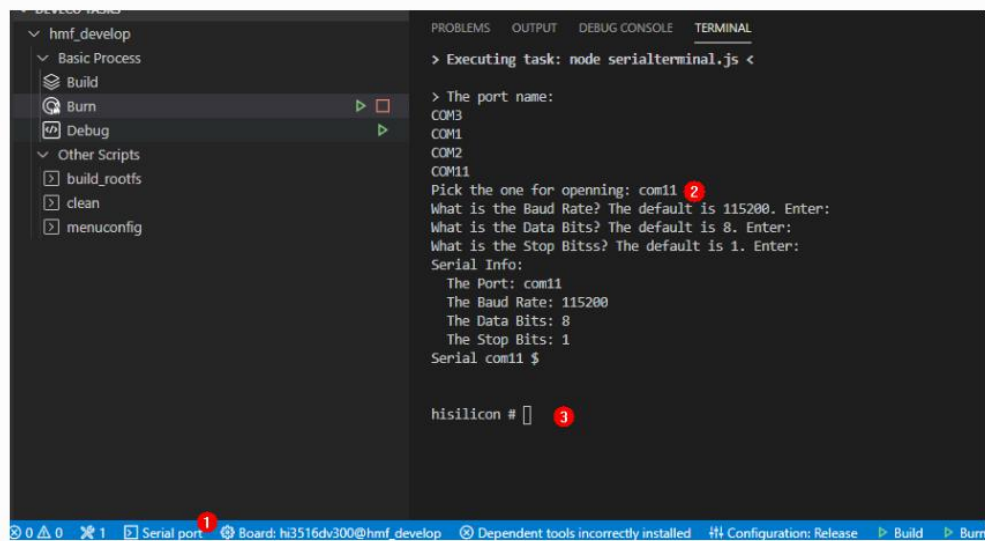
图8 烧写成功图



镜像运行

1. 连接串口。

图9 连接串口图



- 单击**Serial port**打开串口。
- 输入"com11"串口编号并连续输入回车直到串口显示"hisilicon"。
- 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。

- (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启动参数

执行命令	<pre>setenv bootcmd "sf probe 0:mmc read 0x0 0x80000000 0x800 0x4800; go 0x80000000"; setenv bootargs "console=ttyAMA0,115200n8 root=emmc fstype=vfat rootaddr=10M rootsize=15M rw"; saveenv reset</pre>
命令解释	<p>setenv bootcmd "mmc read 0x0 0x80000000 0x800 0x4800;go 0x80000000"; 表示选择FLASH器件0，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。</p> <p>setenv bootargs "console=ttyAMA0,115200n8 root=emmc fstype=vfat rootaddr=10M rootsize=15M rw"; 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，“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> <p>saveenv;表示保存当前配置。</p> <p>reset;表示复位单板</p> <p>[可选] “go 0x80000000” 默认配置已将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。</p>

- 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。

- 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10 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 succ
[UnRegisterDeathCallback : 959]wrong cbId:-1.
OpenInputDevice: open /dev/input/event1 succ
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11 应用程序启动图

```
OHOS #
./bin/camera_app
OHOS # ./bin/camera_app
OHOS #
*****

Hello OHOS!

*****

This is a sample: Param = 81
```