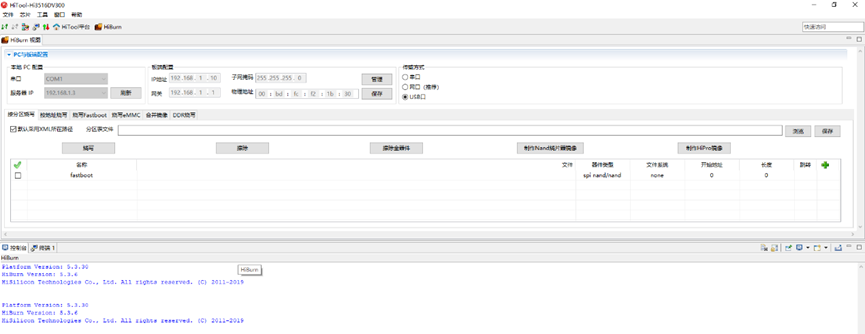
鸿蒙OHOS烧录方法总结

# 烧录工具

理解OHOS驱动框架，用户态APP和内核态驱动之间的通信机制。

OHOS烧录使用的海思自研的烧录工具Hitool，使用起来相对来说比较简单。整体界面如图所示：



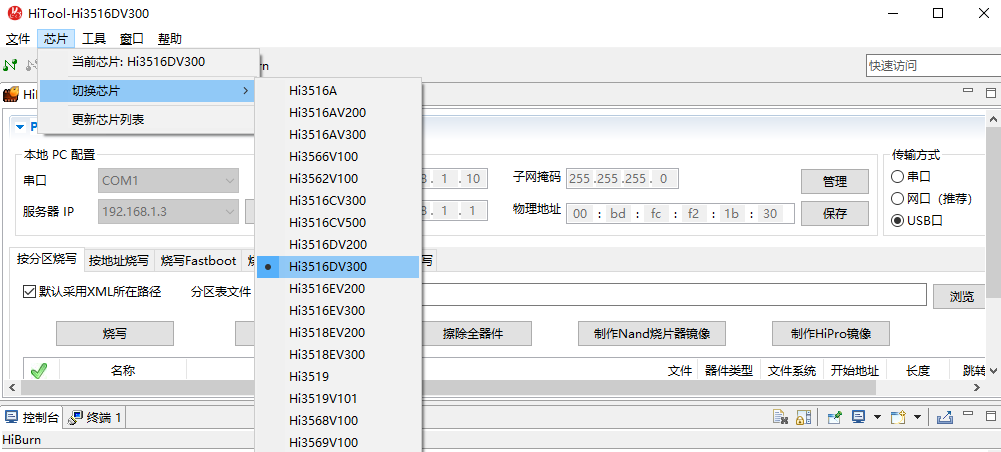
# 烧录内容

烧录内容主要分为两部分，一部分是引导文件fastboot烧录，另一部分为项目文件烧录。这些文件分别是：OHOS\_Image.bin可执行文件,rootfs.img和userfs.img镜像文件。

# 烧录方法

本文以sample\_test demo 为例讲解烧录方法，开发板类型是Hi3516DV300.

烧录的第一步是打开Hitool工具，选择对应的芯片类型。如果你的开发板使用的是Hi3518ev300的，请选择对应的芯片。



## 串口烧录

### fastboot烧录

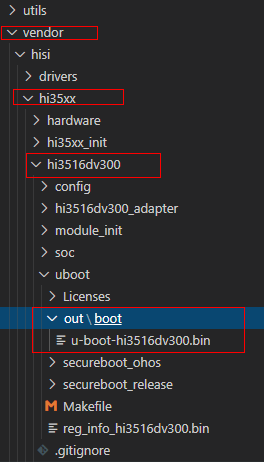
Hiburn请按照如图进行设置：

1. 我手上用的开发板是Hi3516DV300的，它的外挂flash是spi nor类型的，你们自己的设置请根据实际情况确定。
2. 本示例所用的串口设备，计算机识别成COM1，你们做开发时请设置成相应的端口号。
3. 烧录方式选择fastboot。

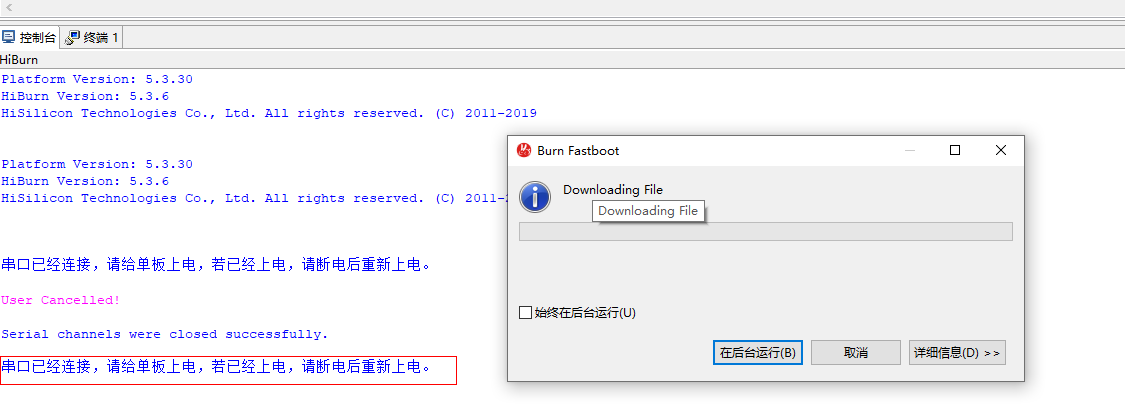




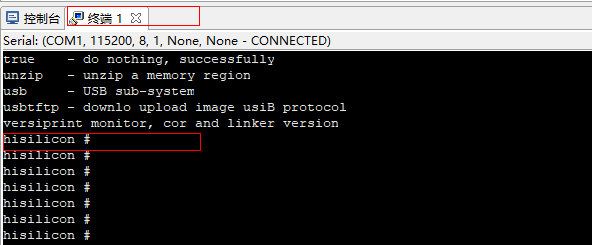
Uboot所在目录(如果uboot不能正常启动，请从其他地方寻找相对老一点的版本uboot)：



点击开始烧录，开发板电源拔掉并重新接上。



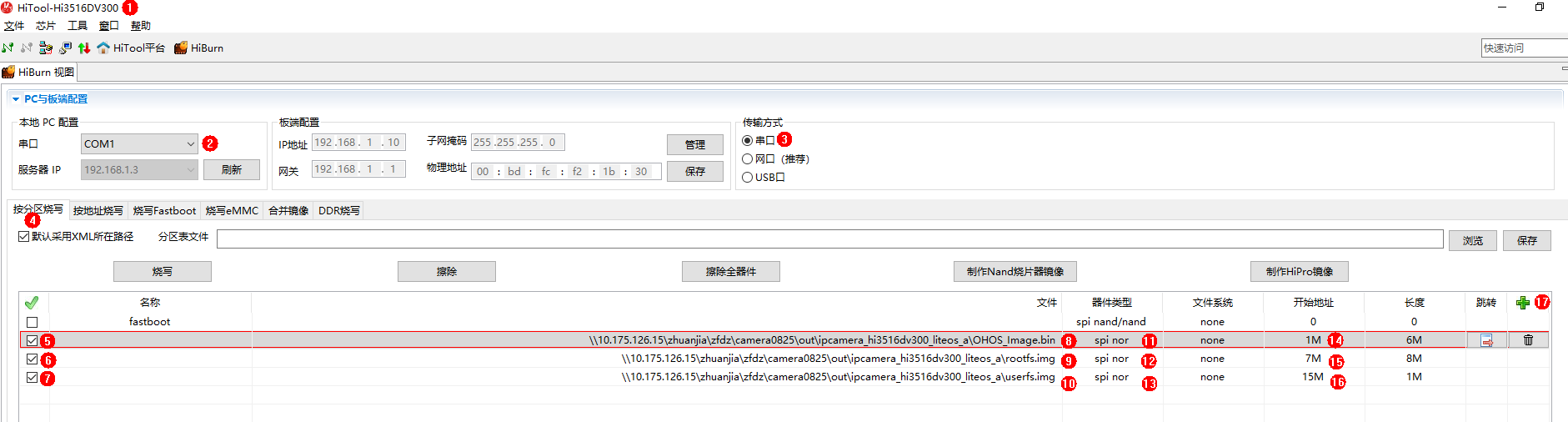
烧写成功后，打开终端，连接串口可以进入hisilion命令行。



### 应用文件烧录

#### Hitool设置：

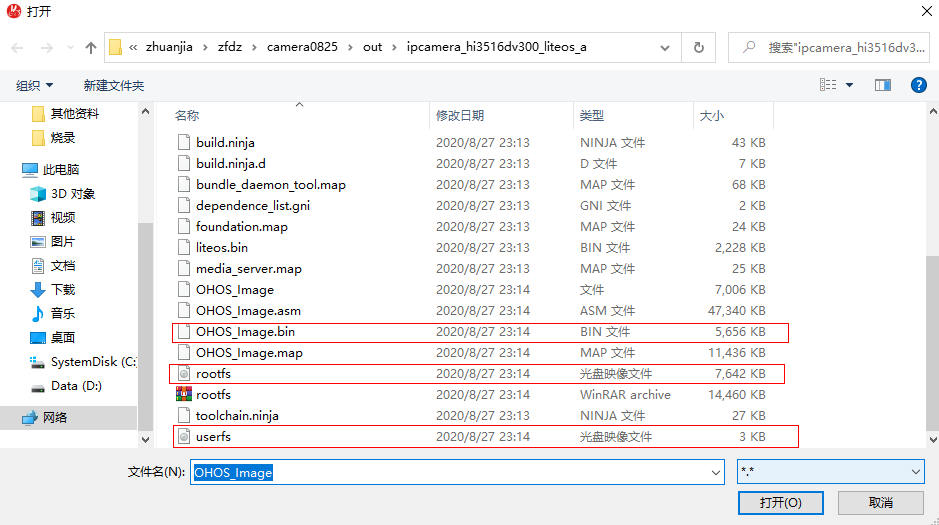
首先软件的设置如下图所示，请参考图例正确设置：



这里面比较重要的就是开始地址和文件长度的确定，这个部分的规则如下：

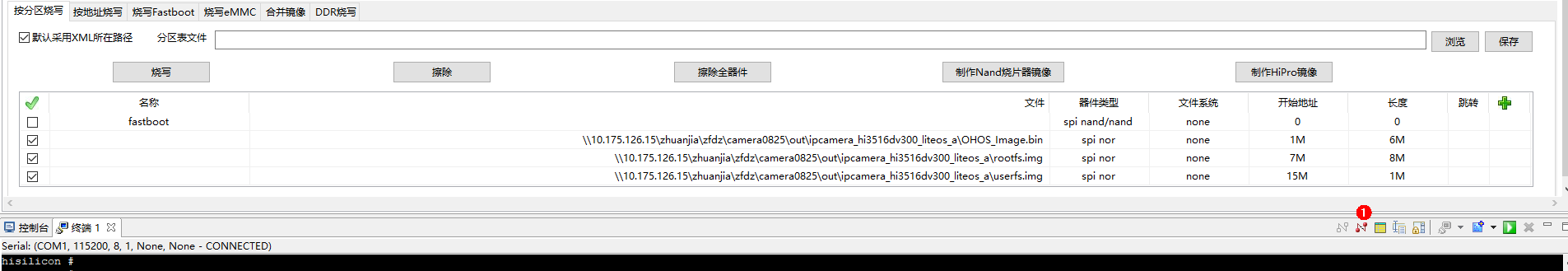
1. Uboot占用了1M的存储空间，所以OHOS\_Image.bin的开始地址是1M，
2. 文件大小不足1M的取值1M，比如本示例中OHOS\_Image.bin的大小是5656kb，Hiburn中它的长度设置为6M。所以rootfs.img烧录的起始地址是1M + 6M = 7M。
3. 其余情况，以此类推。

本示例中，各文件长度如下图所示：



#### 断开串口

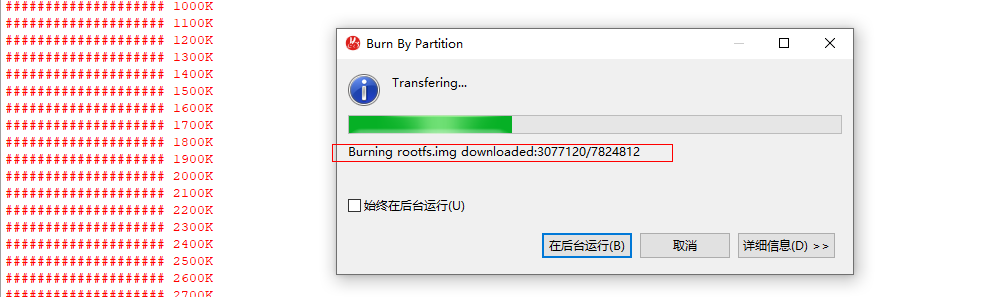
开始烧录前，首先先断开打印信息的串口链接，把串口腾出来给烧录器使用。



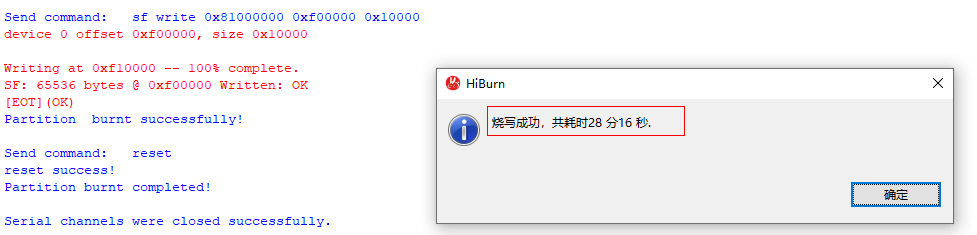
#### 开始烧录

点击“烧写”，并给开发板重新上电，烧录就开始了。16M的文件，用串口11500bps的烧写速度，是需要花费一段时间的，所以系统及应用文件的烧录一般不建议使用串口。

烧录过程截图：



烧写完成，耗时28分钟。

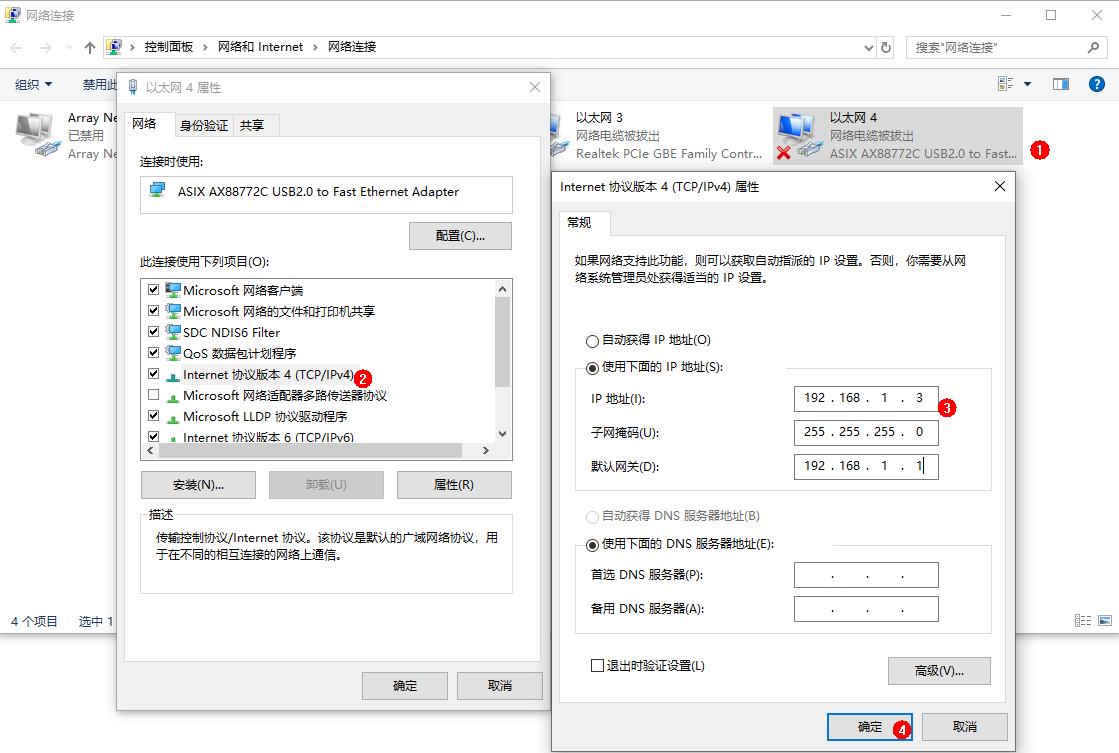


## 网口烧录

本示例网口烧录，硬件设备使用的是UGREEN的usb转RJ45网线接口，仅供参考。

### 网口增加IP地址

首先为开发板互联网口添加IP地址：192.168.1.3，增加方法如下：



1. 控制面板中找到与单板连接的网卡，右键打开属性。
2. 选择Internet协议版本4（TCP/IPv4）。
3. 按图配置IP地址和网关。
4. 点击“确定”完成配置保存。

### 单板IP配置

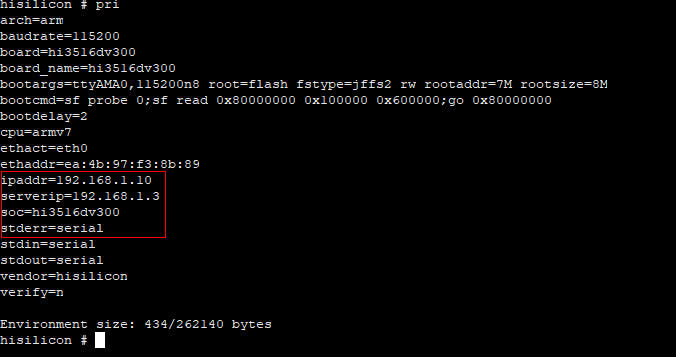
单板与PC间连接串口后，将单板上电，自动进入U-boot，按键盘"回车"键可在命令窗口输入命令，为单板配置192.168.1.10的IP地址，并指定192.168.1.3的服务器IP地址，命令如下。

**setenv serverip 192.168.1.3**

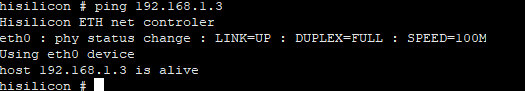
**setenv ipaddr 192.168.1.10**

**saveenv**

配置完成后使用**pri**命令确认查看配置结果，如下图所示。

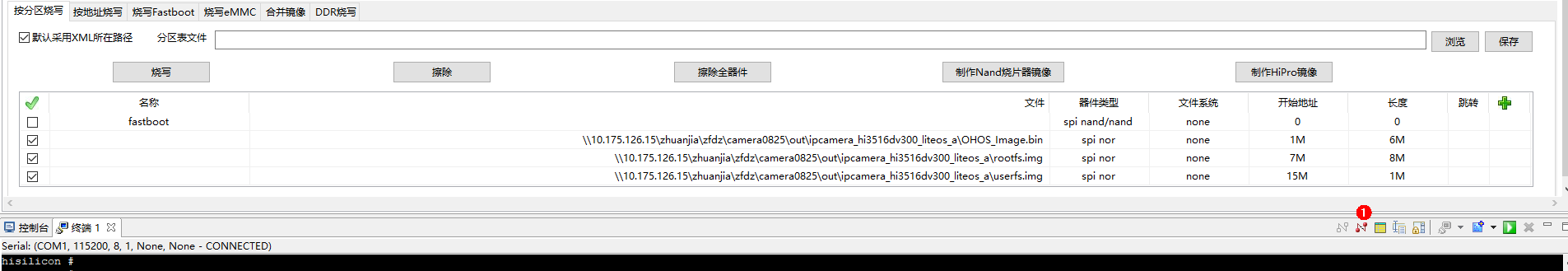


检查网络：在PC端使用ping命令确定单板与PC的网络，命令为"ping 192.168.1.3"。



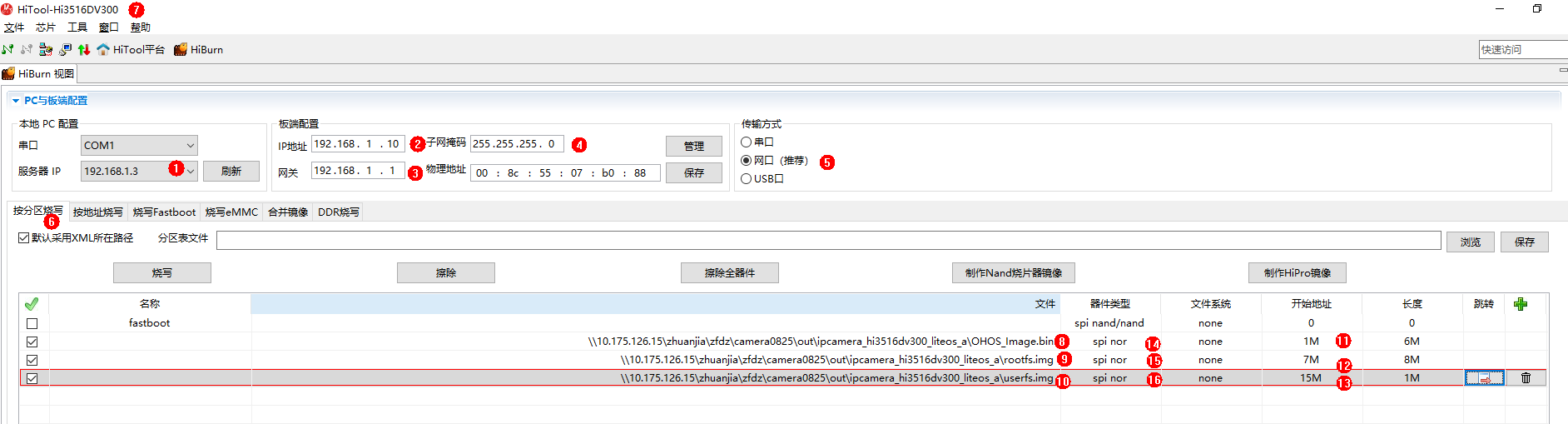
### 断开串口

开始烧录前，首先先断开打印信息的串口链接，把串口腾出来给烧录器使用。

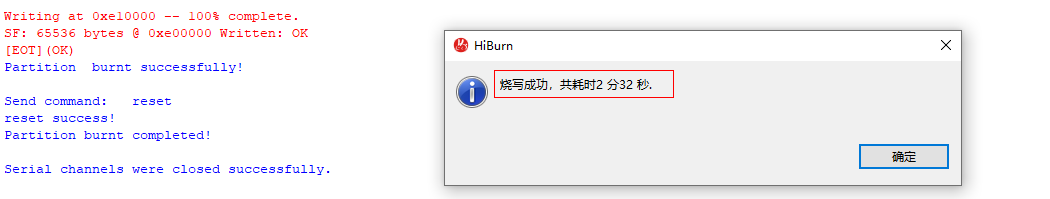


### 配置Hitool并烧录

Hitool配置如下，请参考。



配置完成后，点击烧录。烧录成功截图：



### 网口烧录的命令分析

网口烧录的命令删去中间的细节输出，整理如图所示：

start download process.

Boot started successfully!

Send command: getinfo version

version: U-Boot 2016.11

[EOT](OK)

Send command: **sf probe 0//选择spi flash 0**

[EOT](OK)

Send command: getinfo spi

Block:64KB Chip:16MB\*1

ID:0xC2 0x20 0x18

Name:"MX25L128XX"

[EOT](OK)

waiting phy ready, it will take 8s.

Send command: setenv serverip 192.168.1.3

[EOT](OK)

Send command: setenv ethaddr 00:8c:55:07:b0:88

[EOT](OK)

Send command: setenv ipaddr 192.168.1.10

[EOT](OK)

Send command: setenv netmask 255.255.255.0

[EOT](OK)

Send command: setenv gatewayip 192.168.1.1

[EOT](OK)

Tftp server Download Home switched to <\\10.175.126.15\zhuanjia\zfdz\camera0825\out\ipcamera\_hi3516dv300\_liteos\_a>

Send command: **mw.b 0x81000000 0xFF 0x590000//把内存中从0x81000000 开始的0x590000个字节全部置1**

[EOT](OK)

Send command: **tftp 0x81000000 OHOS\_Image.bin//把OHOS\_Image.bin文档加载到内存中去，起始地址为0x81000000.**

Hisilicon ETH net controler

eth0 : phy status change : LINK=UP : DUPLEX=FULL : SPEED=100M

Using eth0 device

TFTP from server 192.168.1.3; our IP address is 192.168.1.10

Filename 'OHOS\_Image.bin'.

Load address: 0x81000000

Bytes transferred = 5791744 (586000 hex)

[EOT](OK)

Send command: crc32 81000000 586000

crc32 for 81000000 ... 81585fff ==> 78b90ed7

[EOT](OK)

Send command: **sf probe 0//选择spi flash 0**

[EOT](OK)

Send command: **sf erase 0x100000 0x600000//把flash存储器中从0x100000开始的6M字节的存储空间清零。**

Send command: **sf write 0x81000000 0x100000 0x590000//把内存中从0x81000000开始的0x590000字节的内容copy到flash中，flash中存储的起始地址是0x100000**

device 0 offset 0x100000, size 0x590000

Send command: **mw.b 0x81000000 0xFF 0x780000//把内存中从0x81000000 开始的0x780000个字节全部置1**

[EOT](OK)

Send command: **tftp 0x81000000 rootfs.img//加载rootfs.img到内存中去，存储单位的起始地址是0x81000000**

Hisilicon ETH net controler

eth0 : phy status change : LINK=UP : DUPLEX=FULL : SPEED=100M

Using eth0 device

TFTP from server 192.168.1.3; our IP address is 192.168.1.10

Filename 'rootfs.img'.

Send command: crc32 81000000 772314

crc32 for 81000000 ... 81772313 ==> ba26d341

[EOT](OK)

Send command: **sf probe 0//选择spi flash 0**

[EOT](OK)

Send command: **sf erase 0x700000 0x800000//把flash存储器中从0x700000开始的8M字节的存储空间清零。**

Send command: **sf write 0x81000000 0x700000 0x780000//把内存中从0x81000000开始的0x780000字节的内容copy到flash中，flash中存储的起始地址是0x700000**

device 0 offset 0x700000, size 0x780000

Send command: **mw.b 0x81000000 0xFF 0x10000//把内存中从0x81000000 开始的0x10000个字节全部置1**

[EOT](OK)

Send command: **tftp 0x81000000 userfs.img//加载userfs.img到内存中去，存储单位的起始地址是0x81000000**

Hisilicon ETH net controler

eth0 : phy status change : LINK=UP : DUPLEX=FULL : SPEED=100M

Using eth0 device

TFTP from server 192.168.1.3; our IP address is 192.168.1.10

Filename 'userfs.img'.

Load address: 0x81000000

Loading: \*##

0 Bytes/s

done

Bytes transferred = 3020 (bcc hex)

[EOT](OK)

Send command: crc32 81000000 bcc

crc32 for 81000000 ... 81000bcb ==> 0361fc92

[EOT](OK)

Send command: **sf probe 0//选择spi flash 0**

[EOT](OK)

Send command: **sf erase 0xf00000 0x100000//把flash存储器中从0xF00000开始的1M字节的存储空间清零。**

Send command: **sf write 0x81000000 0xf00000 0x10000//把内存中从0x81000000开始的0x10000字节的内容copy到flash中，flash中存储的起始地址是0xf00000**

device 0 offset 0xf00000, size 0x10000

Writing at 0xf10000 -- 100% complete.

SF: 65536 bytes @ 0xf00000 Written: OK

[EOT](OK)

Partition burnt successfully!

Send command: reset

reset success!

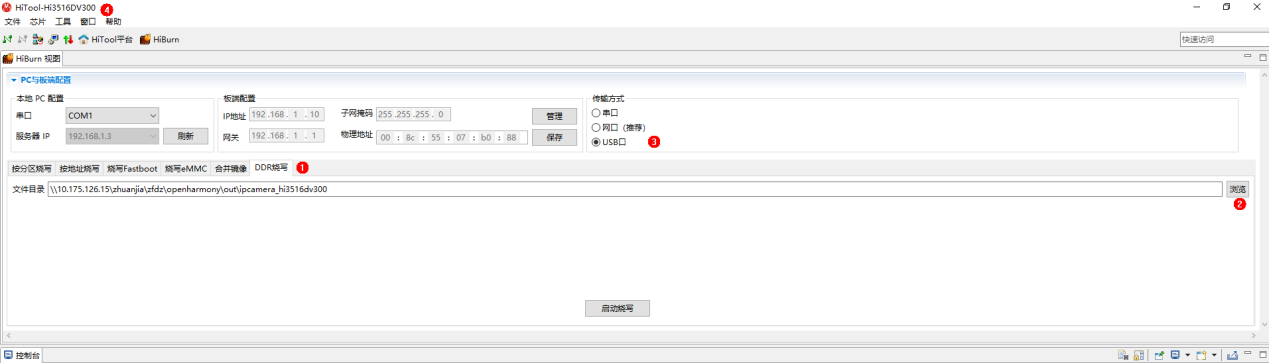
Partition burnt completed!

Serial channels were closed successfully.

## USB烧录

硬件使用的双母口USB线。

### Hitool工具配置



文件目录中只要包含OHOS\_Image.bin,rootfs.img,userfs.img三个文件即可，不限定位置。

烧写方式一定要选择DDR烧写。

### 烧录步骤：

#### 烧录OHOS\_Image.bin文件

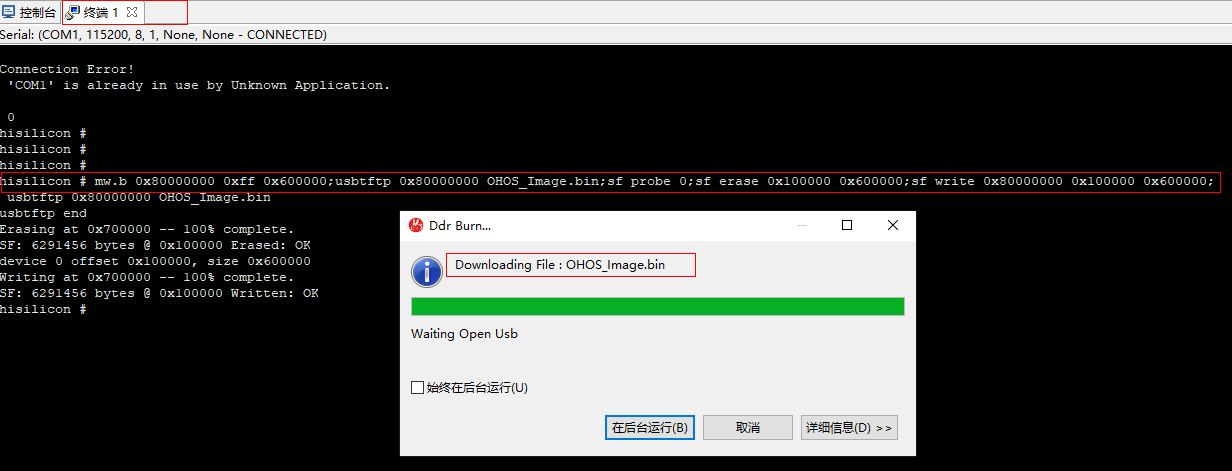
在串口终端中输入以下命令，每条命令的具体含义请参考网口烧录相关部分的描述：

mw.b 0x80000000 0xff 0x600000;**// 0x600000就是OHOS\_Image.bin所占的空间（取整）**

usbtftp 0x80000000 OHOS\_Image.bin;

sf probe 0;sf erase 0x100000 0x600000;sf write 0x80000000 0x100000 0x600000; **//起始地址为0x100000**

如果是Hi3518ev300的板子，请把命令中的0x80000000改成0x40000000。



#### 烧录rootfs.img文件

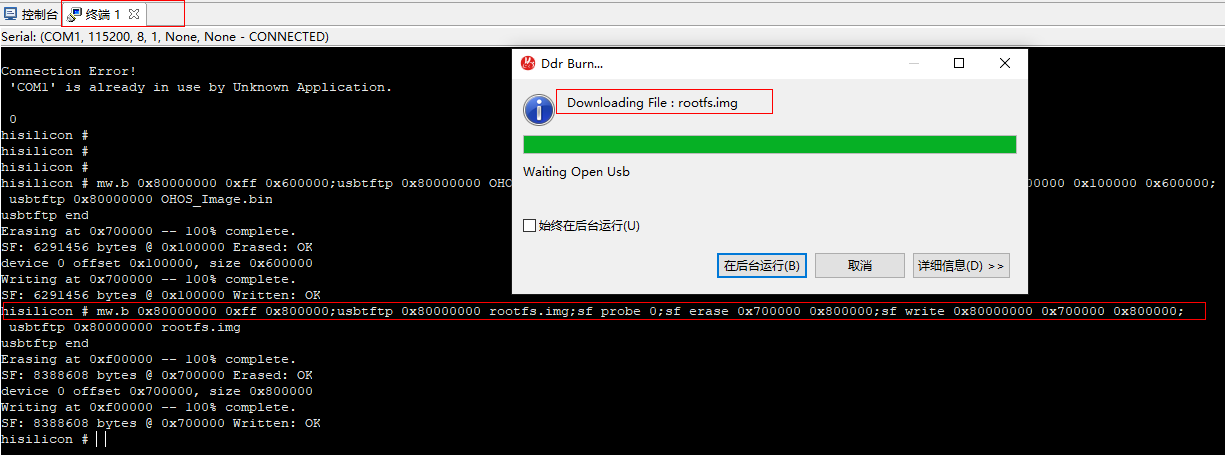
在串口终端中输入以下命令，每条命令的具体含义请参考网口烧录相关部分的描述：

mw.b 0x80000000 0xff 0x800000; **//0x800000就是rootfs.img所占的空间（取整）**

usbtftp 0x80000000 rootfs.img;

sf probe 0;sf erase 0x700000 0x800000;sf write 0x80000000 0x700000 0x800000; **//起始地址为0x100000 + 0x600000 = 0x700000**

如果是Hi3518ev300的板子，请把命令中的0x80000000改成0x40000000。



#### 烧录userfs.img文件

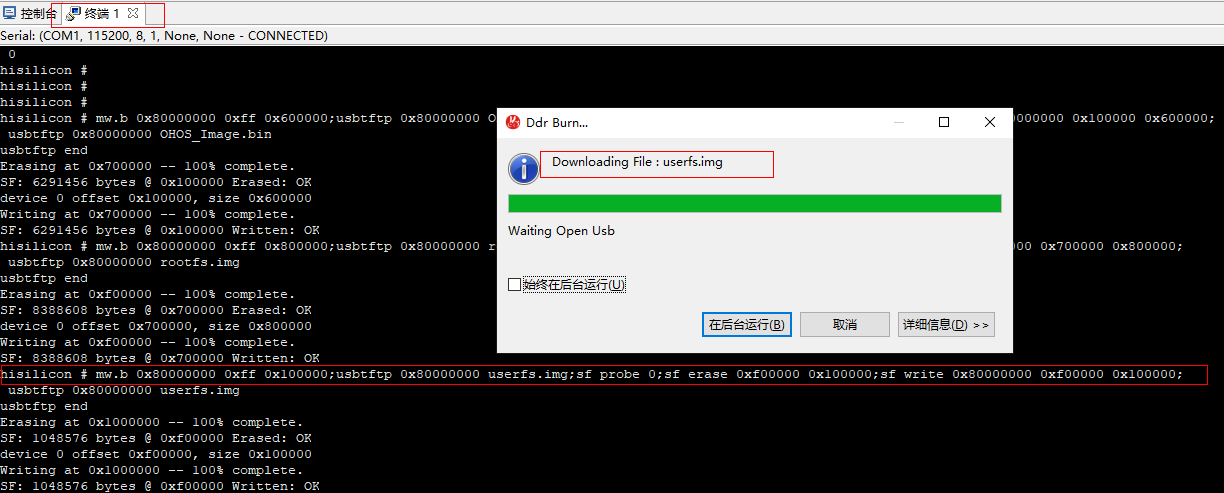
在串口终端中输入以下命令，每条命令的具体含义请参考网口烧录相关部分的描述：

mw.b 0x80000000 0xff 0x100000;**//0x100000就是userfs.img所占的空间（取整）**

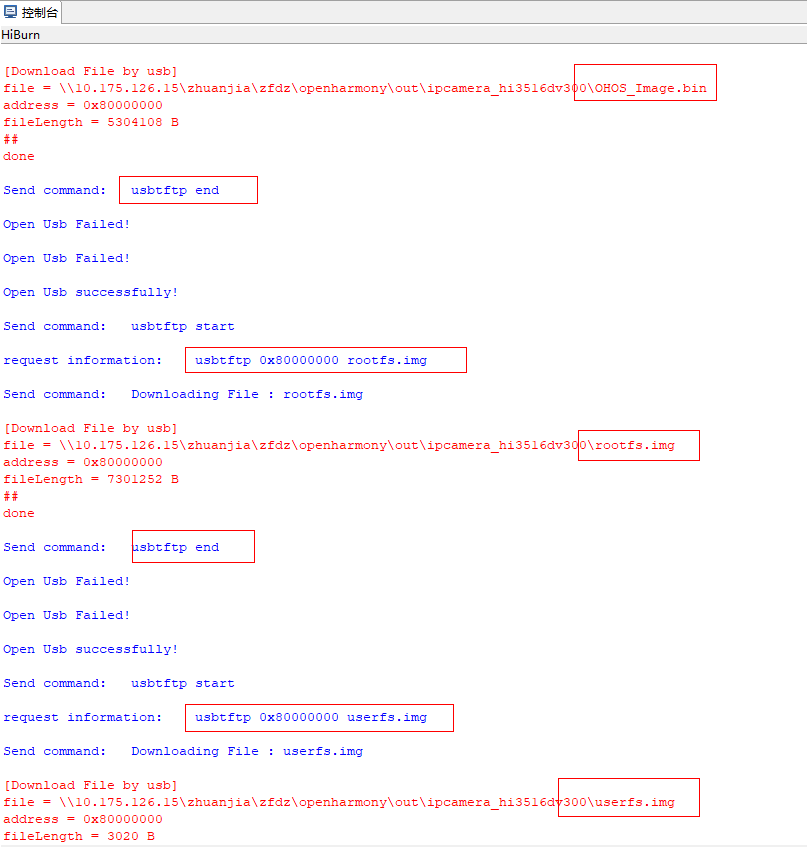
usbtftp 0x80000000 userfs.img;

sf probe 0;sf erase 0xf00000 0x100000;sf write 0x80000000 0xf00000 0x100000; **//起始地址为0x100000 + 0x600000 + 0x800000 = 0xf00000**

如果是Hi3518ev300的板子，请把命令中的0x80000000改成0x40000000。

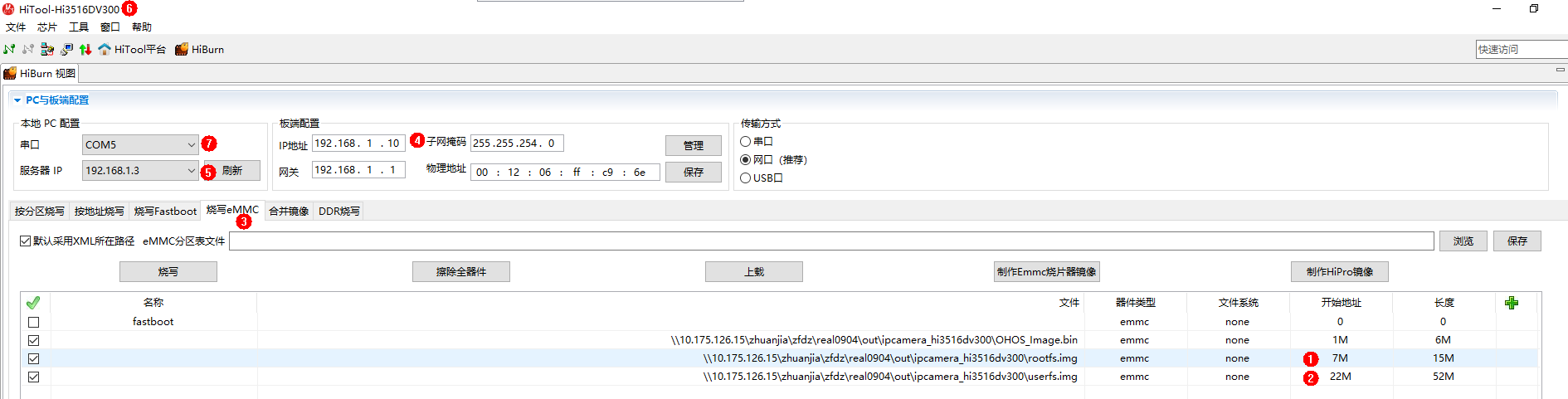


#### 烧录过程中控制台信息：



# 开源板emmc烧录

烧录的时候，选择eMMC方式。我们这里选择通过网口烧录eMMC，网口的配置参考上文。



烧录完成之后，uboot的启动命令还有所不同，在eMMC中，存储是以512B为单位的，如果我们的文件（比如是OHOS\_Image.bin）大小为6M（10进制），我们需要把它转换成16进制，具体转换的公式如下：

10进制中文件总大小：6 \* 1024 \* 1024 Byte=6291456Byte，那么总共是多少个512KB的块呢？我们用6291456/512= 12288个块（十进制），再把十进制中的12288个块转换成16进制，得0x3000。那么也就是说一共有0x3000个512Byte的块。

setenv bootcmd "sf probe 0;mmc read 0x0 0x80000000 0x800 0x3000;";

setenv bootargs "console=ttyAMA0,115200n8 root=emmc fstype=vfat rootaddr=7M rootsize=15M rw"; saveenv;

# 不同烧录方法之间的比较

Usb烧录方式和网口烧录方式想象，不同之处比较如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 烧录方式 | 文件导入命令 | 内存起始地址 | 命令输入方式 | 校验环节 |
| Usb烧录 | usbtftp | 0x80000000 | 手工 | 无 |
| 网口烧录 | tftp | 0x81000000 | 自动 | 有 |

网口烧录和USB烧录的速度基本是相同的，相比之下串口烧录就太慢了，所以正常情况下不推荐串口烧录。

综上比较，只从效率的角度看，烧录器的选择优先级为：

**网口烧录 > USB 烧录 > 串口烧录**

# 其他问题

我们现在用的开发板Flash只有16M，如果程序过大，则无法进行烧录。所以做项目的时候，外挂的flash尽可能大一些，留有足够的预备空间。