

# Purple Pi R1 系统编译

---

开发环境搭建

[从github搭建](#)

[源码下载](#)

[安装交叉编译链](#)

[从百度云网盘搭建](#)

[安装交叉编译链](#)

[安装SDK源码](#)

[安装编译需要的lib & tool](#)

源码编译



## Purple Pi R1 系统编译

深圳触觉智能科技有限公司

[www.industio.cn](http://www.industio.cn)

---

## 开发环境搭建

### 从github搭建

#### 源码下载

```
Plain Text |  
1  industio@industio$:git clone https://github.com/industio/PurPle-Pi-R1.git
```

#### 安装交叉编译链

```
Plain Text |  
1  industio@industio$:cd PurPle-Pi-R1/toolchain  
2  industio@industio$:./install_toolchain.sh
```

### 从百度云网盘搭建

从网盘中获取SDK 源码，拷贝到Ubuntu虚拟机中。

百度云网盘链接：<https://pan.baidu.com/s/1JCRKtwl2oojpnwfxDeWqvQ?pwd=ido6>

Ubuntu版本建议选用18.04版本，版本太低的Ubuntu可能存在区别，导致编译出问题。

SDK包括以下文件：

```
Plain Text |
1 boot.tar.bz2
2 kernel.tar.bz2
3 project.tar.bz2
4 sdk.tar.bz2
5 gcc-arm-8.2-2018.08-x86_64-arm-linux-gnueabihf.tar.gz
```

## 安装交叉编译链

- 解压gcc-arm-8.2-2018.08-x86\_64-arm-linux-gnueabihf.tar.gz

```
Plain Text |
1 industio@industio$:tar -xvf gcc-arm-8.2-2018.08-x86_64-arm-linux-gnueabihf.
tar.gz -C ./
```

- 修改环境变量

这里根据实际的路径来配置

```
Plain Text |
1 industio@industio$:vi ~/.bashrc
2 //在最后一行加入实际环境
3 export PATH=/home/bbelephant/work/ssd20x/gcc-arm-8.2-2018.08-x86_64-arm-lin
ux-gnueabihf/bin:$PATH
```

```
export ARCH=arm
export PATH=/home/bbelephant/work/ssd20x/gcc-arm-8.2-2018.08-x86_64-arm-linux-gnueabihf/bin:$PATH
```

- 验证交叉编译环境是否生效

```
Plain Text |
1 industio@industio$:source ~/.bashrc //使配置的环境生效
2 industio@industio$:which arm-linux-gnueabihf-gcc
```

```
bbelephant@ubuntu:~/work/ssd20x$ source ~/.bashrc
bbelephant@ubuntu:~/work/ssd20x$ which arm-linux-gnueabihf-gcc
/home/bbelephant/work/ssd20x/gcc-arm-8.2-2018.08-x86_64-arm-linux-gnueabihf/bin/
arm-linux-gnueabihf-gcc
```

## 安装SDK源码

将“源码下载”步骤下载的文件放在Linux系统同一个目录下，接着执行以下步骤完成SDK源码的安装。

#### 解压uboot源码

```
▼ Plain Text |
1  industio@industio$:tar -jxvf boot.tar.bz2 -C .
```

#### 解压kernel源码

```
▼ Plain Text |
1  industio@industio$:tar -jxvf kernel.tar.bz2 -C .
```

#### 解压project源码

```
▼ Plain Text |
1  industio@industio$:tar -jxvf project.tar.bz2 -C .
```

#### 解压sdk源码

```
▼ Plain Text |
1  industio@industio$:tar -jxvf sdk.tar.bz2 -C .
```

把所有文件设置权限属性，可避免不必要的问题：

```
▼ Plain Text |
1  industio@industio$:sudo chown industio:industio -R ./*
```

## 安装编译需要的lib & tool

注意：以下lib & tool，可以提前安装，否则会、编译失败

在编译前请确定Ubuntu，shell工具是否为bash，防止出现-e错误。

### 1、查看sh默认选项

```
1 industio@industio$:ls -l /bin/sh
```

```
bbelephant@ubuntu:~/work/ssd20x$ ls -l /bin/sh
lrwxrwxrwx 1 root root 4 May 27 01:11 /bin/sh -> dash
```

## 2、切换sh为bash

```
1 industio@industio$:sudo dpkg-reconfigure dash
```

```
bbelephant@ubuntu:~/work/ssd20x$ sudo dpkg-reconfigure dash
Removing 'diversion of /bin/sh to /bin/sh.distrib by dash'
Adding 'diversion of /bin/sh to /bin/sh.distrib by bash'
Removing 'diversion of /usr/share/man/man1/sh.1.gz to /usr/share/man/man1/sh.distrib.1.gz by dash'
Adding 'diversion of /usr/share/man/man1/sh.1.gz to /usr/share/man/man1/sh.distrib.1.gz by bash'
bbelephant@ubuntu:~/work/ssd20x$ ls -l /bin/sh
lrwxrwxrwx 1 root root 4 May 27 03:17 /bin/sh -> bash
```

然后选择NO，可以解决脚本编译-e的问题。

- 安装需要的lib库

```
1 industio@industio$:sudo apt-get install libncurses5-dev libncursesw5-dev
2 industio@industio$:sudo apt-get install lib32z1
3 industio@industio$:sudo apt-get install lsb-core
4 industio@industio$:sudo apt-get install libc6-dev-i386
5 industio@industio$:sudo apt-get install lib32z1
6 industio@industio$:sudo apt-get install libuuid1:i386
7 industio@industio$:sudo apt-get install cmake
8 industio@industio$:sudo apt install bc
9 industio@industio$:sudo apt-get install xz-utils
10 industio@industio$:sudo apt-get install automake
11 industio@industio$:sudo apt-get install libtool
12 industio@industio$:sudo apt-get install libevdev-dev
13 industio@industio$:sudo apt-get install pkg-config
```

- 安装需要的tool

```
1 //安装ssh, 方便调试
2 industio@industio$:sudo apt-get install openssh-server
3 //安装xz压缩工具
4 industio@industio$:sudo apt-get install xz-utils
5 //安装 python
6 industio@industio$:sudo apt-get install python
7 //安装 git
8 industio@industio$:sudo apt-get install git
9 //安装 make
10 industio@industio$:sudo apt-get install make
11 //安装 gcc
12 industio@industio$:sudo apt-get install gcc
13 //安装 g++
14 industio@industio$:sudo apt-get install g++
```

我们用的是Ubuntu 18.04, 其他版本的Ubuntu安装的库可能存在差异。

## 源码编译

使用以下命令进行编译:

```
1 industio@industio$:./Release_to_customer.sh -f nand -p ssd202 -m 256
```

如果出现以下错误, 卸载libfdt-dev库, 让uboot强制使用内部的libfdt

```
1 industio@industio$:sudo apt-get remove libfdt-dev
```

```

hostcc tools/ftl_image.o
In file included from ./tools/./include/libfdt.h:54,
                 from ./tools/fdt_host.h:11,
                 from ./tools/mkimage.h:22,
                 from ./tools/./common/bootm.c:31,
                 from tools/common/bootm.c:1:
./usr/include/libfdt_env.h:27:30: error: conflicting types for 'fdt64_t'
 27 | typedef uint64_t FDT_BITWISE fdt64_t;
    |                               ^~~~~~
In file included from <command-line>:
././include/libfdt_env.h:19:16: note: previous declaration of 'fdt64_t' was here
 19 | typedef __be64 fdt64_t;
    |               ^~~~~~
In file included from ././include/libfdt_env.h:12,
                 from <command-line>:
./usr/include/libfdt_env.h:47:24: error: expected ')' before 'x'
 47 | static inline uint32_t fdt32_to_cpu(fdt32_t x)
    |                               ^~~~~~
././include/compiler.h:66:9: error: expected ')' before '&' token
 66 | (((x) & 0xff000000) >> 24) | \
    |           ^
././include/compiler.h:66:23: error: expected ')' before '>>' token
 66 | (((x) & 0xff000000) >> 24) | \
    |                               ^
././include/compiler.h:66:30: error: expected ')' before '|' token
 66 | (((x) & 0xff000000) >> 24) | \
    |                               ^
./usr/include/libfdt_env.h:51:23: error: expected ')' before 'x'
 51 | static inline fdt32_t cpu_to_fdt32(uint32_t x)
    |                               ^~~~~~
././include/compiler.h:66:9: error: expected ')' before '&' token
 66 | (((x) & 0xff000000) >> 24) | \
    |           ^

```

如下图所示，说明编译完成。

```

make[2]: Nothing to be done for 'boot_images'.
make[2]: Leaving directory '/home/bbelephant/work/ssd20x/project/image'
make scripts
make[2]: Entering directory '/home/bbelephant/work/ssd20x/project/image'
mkdir -p /home/bbelephant/work/ssd20x/project/image/output/images/scripts
make set_partition
make[3]: Entering directory '/home/bbelephant/work/ssd20x/project/image'
make[3]: Leaving directory '/home/bbelephant/work/ssd20x/project/image'
make cis_spinand_script ipl_spinand_script ipl_cust_spinand_script uboot_spinand_script logo_spinand_script kernel_spinand_script rootfs_spinand_ubifs_script miservice_spinand_ubifs_script customer_spinand_ubifs_script appconfigs_spinand_ubifs_script ubi_spinand_partition_script ubi_spinand_config_script
make[3]: Entering directory '/home/bbelephant/work/ssd20x/project/image'
kernel-image done!!!
make[3]: Nothing to be done for 'ubi_spinand_partition_script'.
##@echo reset >> /home/bbelephant/work/ssd20x/project/image/output/images/scripts/set_config
make[3]: Leaving directory '/home/bbelephant/work/ssd20x/project/image'
if [ "cis" != "" ]; then \
    echo estar scripts/[[cis.es >> /home/bbelephant/work/ssd20x/project/image/output/images/auto_update.txt; \
fi;
if [ "cis" != "" ]; then \
    echo estar scripts_bin/[[cis.es >> /home/bbelephant/work/ssd20x/project/image/output/images/auto_update_bin.txt; \
fi;
make[2]: Leaving directory '/home/bbelephant/work/ssd20x/project/image'
make[1]: Leaving directory '/home/bbelephant/work/ssd20x/project/image'
./split_partition.sh
split customer image
customer.es is not over size,do nothing!!

```

编译成功后，将在当前目录下生成images/目录，该目录下的文件即为烧录所使用的固件。