

IDO-EVB3566-V1 Linux开发手册

1 SDK下载

1.1 源码下载

1.2 源码解压

2 安装SDK编译依赖环境

3 SDK编译

3.1 配置选择

3.2 完整编译

3.2.2 编译uboot/kernel



IDO-EVB3566-V1

Linux 开发手册

文档修订历史

版本	PCBA版本	修订内容	修订	审核	日期
V1.0	V1B	创建文档	HWC	IDO	2024/07/11

1 SDK下载

1.1 源码下载

从以下地址下载IDO-EVB3566的SDK：

链接：<https://pan.baidu.com/s/15Al-FJcegmrCjEAWH5J-NA?pwd=6666>

提取码：6666

1.2 源码解压

由于SDK打包后体积较大，按照4GB大小进行了分割，因此下载后需要合并，命令如下：

```
Shell |
1 $ cat ido_evb3566_5.10_linux.tar.gz.* > ido_evb3566_5.10_linux.tar.gz
```

合并后，进行解压，解压得到的ido_evb3566_5.10_linux即为sdk目录。

```
Shell |
1 $ tar -zxvf ido_evb3566_5.10_linux.tar.gz
```

注意：不要使用sudo解压，否则会导致后面编译出问题，命令如下：

2 安装SDK编译依赖环境

建议使用Ubuntu22.04及以上版本系统编译EVB3566 SDK，在编译前执行以下命令安装依赖环境，命令如下：

```
Shell |
1 $ sudo apt-get install repo git ssh make gcc libssl-dev liblz4-tool \
2 expect g++ patchelf chrpath gawk texinfo chrpath diffstat binfmt-support \
3 qemu-user-static live-build bison flex fakeroot cmake \
4 unzip device-tree-compiler python-pip ncurses-dev python-pyelftools expect-
dev
```

3 SDK编译

3.1 配置选择

SDK配置选择命令如下：

```
Shell |
1 //进入sdk目录
2 $ ./build.sh lunch
3 Pick a defconfig:
4
5 1. evb3566-edp-1920x1080-buildroot_defconfig
6 2. evb3566-edp-1920x1080-debian_defconfig
7 3. evb3566-edp-1920x1080-ubuntu_defconfig
8 4. evb3566-hdmi-buildroot_defconfig
9 5. evb3566-hdmi-debian_defconfig
10 6. evb3566-hdmi-ubuntu_defconfig
11 7. evb3566-hdmi-ubuntu_docker_defconfig
12 8. evb3566-lvds-1280x800-buildroot_defconfig
13 9. evb3566-lvds-1280x800-debian_defconfig
14 10. evb3566-lvds-1280x800-ubuntu_defconfig
15 11. evb3566-mipi-1200x1920-buildroot_defconfig
16 12. evb3566-mipi-1200x1920-debian_defconfig
17 13. evb3566-mipi-1200x1920-ubuntu_defconfig
18
```

根据屏幕（mipi/lvds/edp/hdmi）和文件系统（buildroot/debian/ubuntu），输入编号以选择对应的配置。如选择屏hdmi+buildroot文件系统，则输入4。

对应的配置文件在//kernel/arch/arm64/configs/rockchip_linux_evb3566_defconfig。

3.2 完整编译

首次编译请在sdk顶层目录执行build.sh，编译完成，将在rockdev/目录生成完整升级固件update.img。

```
Shell |
1 $ ./build.sh
```

注意：不要使用sudo编译，否则会导致编译失败。

3.2.2 编译uboot/kernel

在调试时，可单独编译uboot或kernel部分：

```
1 //编译uboot
2 $ ./build.sh uboot
3
4 //编译kernel
5 $ ./build.sh kernel
```

编译完成后，将在`rockdev/`目录生成对应的`uboot.img`。