IDO-EVB3566-V1 Android开发手册

IDO-EVB3566-V1

Android 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	HJT	IDO	2024/05/08

1 源码获取

源码下载路径

链接：https://pan.baidu.com/s/1WGSk4WukhVchpeVanClelg?pwd=1234

提取码：1234

解压源码命令如下：

```bash
tar -xzvf ido_evb3566-v1b_android11_sdk_240605.tar.tar.gz -C your/target/path
```

2 Android_SDK编译环境配置

推荐编译主机配置如下：

1. Ubuntu22.04 操作系统
2. 64 位 CPU
3. 16GB 物理内存+交换内存
4. 250GB 空闲的磁盘空间

开发环境搭建，请参考SDK根目录下

```shell
$ sudo apt-get update
$ sudo apt-get install git gnupg flex bison gperf libsdl1.2-dev libesd-java \
libwxgtk3.0-dev squashfs-tools build-essential zip curl libncurses5-dev zlib1g-dev \
pngcrush schedtool libxml2 libxml2-utils xsltproc lzop libc6-dev schedtool g++-multilib \
lib32z1-dev lib32ncurses5-dev lib32readline-dev gcc-multilib libswitch-perl libssl-dev \
unzip zip device-tree-compiler liblz4-tool python-pyelftools python3-pyelftools -y
```

3 SDK编译

3.1 uboot编译步骤

进入 sdk 根目录执行命令如下：

```shell
$ cd u-boot
$ ./make.sh rk3566
```

3.2 kernel编译步骤

内核配置文件路径：kernel/arch/arm64/configs/rockchip_defconfig

设备树文件路径：kernel/arch/arm64/boot/dts/rockchip/

EVB3566-V1B开发板dts：ido-evb3566-v1a.dts

```Shell
/dts-v1/;

#include <dt-bindings/gpio/gpio.h>
#include <dt-bindings/display/media-bus-format.h>
#include <dt-bindings/pinctrl/rockchip.h>
#include "rk3566.dtsi"
#include "rk3568-android.dtsi"
#include "ido-sbc3566-core.dtsi"
//#include "ido-sbc3566-v1a-light-sensor.dtsi"

/* lcd */
// #include "ido-evb3566-v1a-lvds-1280-800.dtsi"  //单lvds屏
// #include "ido-evb3566-v1a-edp-1920-1080.dtsi"  //edp屏
#include "ido-evb3566-v1a-mipi-1200-1920.dtsi"    //mipi屏
```

3.2.1 完整编译kernel

以编译mipi屏内核为例（默认为hdmi），取消dts第14行注释，编译命令如下：

```Shell
$ cd kernel
$ make ARCH=arm64 rockchip_evb3566_defconfig
$ make ARCH=arm64 ido-evb3566-v1a.img -j10
```

以上方式编译完成后，kernel目录生成的boot.img文件不能直接用于烧录。内核分区烧录的文件是编译完Android所产生的rockdev/Image-rk3566_r/boot.img文件。

注意：编译如果提示选择io_domain电压，我司核心板vccio4是1.8V，其它为3.3V。

3.2.2 单独编译kernel

此处的编译方法的前提是，已存在rockdev/Image-rk3566_r/boot.img文件（即Android代码已经完全编译过一次）。

编译原理：在kernel目录下将编译生成的 kernel.img 和 resource.img 替换到旧的 boot.img 中，可以直接单独烧录。这样就能在调试驱动时，快速编译获得可直接烧录的内核镜像。

```Shell
$ cd kernel
$ make ARCH=arm64 rockchip_evb3566_defconfig
$ ./mk_kernel.sh ido-evb3566-v1a.img
```

使用此方法编译出kernel/boot.img文件可以直接用于烧录至boot分区。

注意：make ARCH=arm64 rockchip_defconfig，正常没有修改rockchip_defconfig配置文件的情况下只需要执行一次即可。

3.3 Android 编译及固件生成步骤

```Shell
$ source build/envsetup.sh
$ lunch rk3566_r-userdebug
$ make -jn
```

n: CPU内核数（按最大数编译一般为CPU数*2），user版本选择：lunch rk3566_r-user

4 固件打包

编译完成后，执行 SDK 根目录下的 mkimage.sh 脚本生成固件，所有烧写所需的各分区镜像会被拷贝至rockdev/Image-rk3566_r/目录，命令如下：

```Shell
$ ./mkimage.sh
```

将所有分区镜像合并成单个的镜像，命令如下：

```Bash
$ cd RKTools/linux/Linux_Pack_Firmware/rockdev/
$ ./mkupdate_rk356x.sh
```

执行mkupdate_rk356x.sh 命令后，会将各分区镜像合并成一个整包镜像文件：RKTools/linux/Linux_Pack_Firmware/rockdev/update.img。