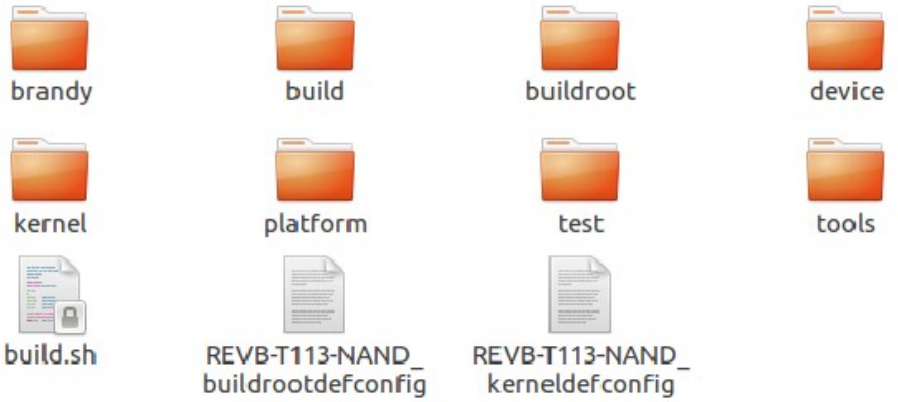
REVB-T113

Linux 系统编译说明

阅读本文前，请先查看原厂开发文档中的《T113_Longan_Linux_SDK 开发环境配置手册.pdf》第 3.1 节，Linux 开发环境搭建，搭建好开发环境后再去编译源码，如编译源码遇到问题，请检查开发环境。

REVB-T113 开发板的 Linux 源码，分为 NAND 版和 EMMC 版本。
包名：REVB-T113-NAND-日期.tar.gz　　REVB-T113-EMMC.tar-日期.gz
下载 SDK 包后，使用 md5sum 对比校验值，如果校验值不一致，重新下载

解压 SDK 源码包，得到如下目录



brandy:　　交叉编译器、uboot 源码等

build:　　编译脚本

buildroot:　　buildroot 文件系统源码

device:　　系统配置文件、分区文件、bootlogo 等

kernel:　　linux5.4 内核源码

platform:　　软件 demo、编解码库、QT 源码等

test:　　dragonboard 测试源码

tools:　　PC 端开发工具

系统配置文件：

NAND 版：device/config/chips/t113/configs/evb1_auto_nand/board.dts

EMMC 版：device/config/chips/t113/configs/evb1_auto /board.dts

uboot 配置文件：

NAND 版：device/config/chips/t113/configs/evb1_auto_nand/uboot-board.dts

EMMC 版：device/config/chips/t113/configs/evb1_auto /uboot-board.dts

生成的固件：

NAND 版：out/t113_linux_evb1_auto_nand_uart0.img

EMMC 版：out/t113_linux_evb1_auto _uart0.img

Linux 内核路径：kernel/linux-5.4

u-boot 路径：brandy/brandy-2.0/ u-boot-2018

Buildroot 路径：buildroot/buildroot-201902

QT 路径： platform/framework/qt/qt-everywhere-src-5.12.5

软件 Demo 路径：platform/framework/auto/sdk_demo

多媒体编解码库路径：platform/framework/auto/sdk_lib

Bootlogo 路径:

NAND 版：device/config/chips/t113/configs/evb1_auto_nand/longan/bootlogo.bmp

EMMC 版：device/config/chips/t113/configs/evb1_auto/longan/bootlogo.bmp

编译后生成的：

文件系统路径：out/ t113/evb1_auto_nand/longan/buildroot/target

linux 内核配置文件路径：out/kernel/build/.config

buildroot 配置文件路径：out/t113/evb1_auto_nand/longan/buildroot/.config

T113 平台代号：sun8iw20p1

注意：SDK 自带交叉编译器，在编译时由编译脚本从 brandy 目录自动安装到 out 目录，无需要使用外部的交叉编译器

NAND 版源码编译

首次编译，配置板型，仅执行一次

```
./build.sh autoconfig -c linux -o longan -k linux-5.4 -i t113 -b evb1_auto_nand -n default
```

进行 SDK 编译环境配置 每次开新的 shell 都要执行，再进行开发

```
source build/envsetup.sh
```

编译整个 SDK

```
./build.sh
```

打包固件

```
./build.sh pack
```

替换 Linux 内核配置文件

首次编译 SDK 完成后，在根目录下执行：

```
cp REVB-T113-kerneldefconfig out/kernel/build/.config
```

替换 buildroot 配置文件

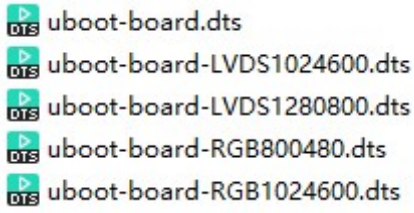
首次编译 SDK 完成后，在根目录下执行：

```
cp REVB-T113-buildrootdefconfig out/t113/evb1_auto_nand/longan/buildroot/.config
```

替换 2 种配置文件后，再用./build.sh 编译 SDK

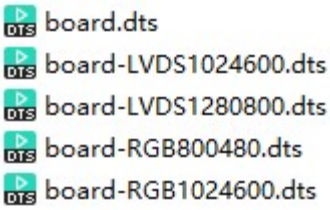
替换 uboot 配置文件

uboot 配置文件中包含 LCD 的参数，使用不同的屏幕，一定要替换相应的 uboot-board.dts，否则屏幕不会显示，系统只认名为 uboot-board.dts 的文件，将要使用的配置文件改名为 uboot-board.dts



替换系统配置文件

board.dts 是 linux 内核的配置文件，所有的硬件功能都在此文件中配置，系统只认名为 board.dts 的文件，将要使用的配置文件改名为 board.dts



EMMC 版源码编译

首次编译，配置板型，仅执行一次

```
./build.sh autoconfig -c linux -o longan -k linux-5.4 -i t113 -b evb1_auto -n default
```

进行 SDK 编译环境配置 每次开新的 shell 都要执行，再进行开发

```
source build/envsetup.sh
```

编译整个 SDK

```
./build.sh
```

打包固件

```
./build.sh pack
```

替换 Linux 内核配置文件、buildroot 配置文件、uboot 配置文件、系统配置文件和上面 NAND 版本的方式相同

编译 QT

```
./build.sh qt
```

编译 QT DEMO

```
cd platform/framework/auto/qt_demo
./build.sh
```

编译 cedar

```
cd platform/framework/auto/buildcedar
./T113_cedar_compile.sh
```

配置编译 Linux 内核

在根目录执行./build.sh menuconfig 可打开内核的图形化配置
或进入 out/kernel/build，执行 make menuconfig ARCH=arm
仅编译内核：./build.sh kernel

配置编译 Buildroot

NAND 版进入 out/t113/evb1_auto_nand/longan/buildroot
EMMC 版进入 out/t113/evb1_auto /longan/buildroot
执行 make menuconfig 可打开 buildroot 的图形化配置
仅编译 buildroot：./build.sh buildroot

编译 uboot

进入 brandy/brandy-2.0/ u-boot-2018
执行 make sun8iw20p1_config
编译：make -j

定制文件系统

路径：platform/framework/auto/rootfs
编译时将此路径的内容替换到 out/ t113/evb1_auto_nand/longan/buildroot/target 中
用户可将应用程序、库文件、配置文件等放到此路径
应用程序放到 rootfs /usr/bin、库文件放到 rootfs /lib、配置文件放到 rootfs /etc

自启动脚本：rootfs/etc/init.d/rcS
用户可将需要自启动的应用程序写到此脚本的最后一行，开机自动运行
开机加载驱动模块 insmod 命令也写到此脚本

QT 环境变量脚本：rootfs/etc/qtenv.sh
运行 QT 应用前需运行此脚本
例如在 rcS 的最后一行写：

```
. /etc/qtenv.sh && mainwindow &
```

用户编译应用程序可参考 platform/framework/auto/sdk_demo 例程里的 Makefile 写法

Linux 硬件接口使用说明

GPIO：

T113 提供 4 个可编程 GPIO： PG0 PG2 PG3 PG4

使用命令导出设备文件：

```
echo 192 > /sys/class/gpio/export
echo 194 > /sys/class/gpio/export
echo 195 > /sys/class/gpio/export
echo 196 > /sys/class/gpio/export
```

对应的 GPIO 设备文件：

PG0 /sys/class/gpio/gpio192/value 和 direction
PG2 /sys/class/gpio/gpio194/value 和 direction
PG3 /sys/class/gpio/gpio195/value 和 direction
PG4 /sys/class/gpio/gpio196/value 和 direction

进入相应的 GPIO 目录操作示例：

```
echo in > direction      设置为输入状态
cat value                读出 1 是高电平，读出 0 是低电平
echo out > direction     设置为输出状态
echo 1 > value           设置高电平
echo 0 > value           设置低电平
```

I2C：

T113 提供了 2 路 I2C 供外部使用，是标准 I2CDEV 驱动
设备名：/dev/i2c-0 /dev/i2c-2
其中 I2C-0 默认用于电容触摸，I2C-2 默认用于 RTC 芯片

串口：

T113 提供了四路 UART，其中串口 3 仅用于调试
串口 1 /dev/ttyS1
串口 3 /dev/ttyS3
串口 4 /dev/ttyS4
串口 5 /dev/ttyS5
使用 Linux 系统通用的串口软件即可操作

PWM:

T113 提供 2 个可编程 PWM： PWM4 PWM6

使用命令导出设备文件：

```
echo 4 > /sys/class/pwm/pwmchip0/export
echo 6 > /sys/class/pwm/pwmchip0/export
```

对应的 PWM 设备文件：
/sys/class/pwm/pwmchip0/pwm4
/sys/class/pwm/pwmchip0/pwm6

进入相应的 PWM 目录操作示例：

```
echo 1000000 > period      设置 PWM 周期时间，单位是 ns，此时为 1KHz
echo 500000 > duty_cycle   设置 PWM 占空比时间，单位是 ns，此时为 50%
echo 1 > enable            使能 PWM 输出
```

INPUT 设备节点：

电阻触摸：/dev/input/event1
电容触摸：/dev/input/event3
按键：/dev/input/event0

网络设备节点：

有线网：eth0
WIFI：wlan0
使用 ifconfig 或其他网络命令、脚本等打开和配置网络

LINEIN 用法

默认已开启 Linein 和 mic，使用 tinycap 可录音，同时录 linein 和 mic 音源
使用 tinymix 开启/关闭 Linein
执行 `tinymix contents`
查看支持的混音器配置，可以看到 24 和 27 项是 LineinL 和 LineinR 的配置
开启 Linein：

```
tinymix set 24 1
tinymix set 27 1
```

关闭 Linein：

```
tinymix set 24 0
tinymix set 27 0
```

测试程序路径：/usr/bin
输入命令直接可运行，不需要加路径

CVBS 显示测试

执行 `tv_tester.sh`

通过 CVBS OUT 显示红绿蓝彩条
具体实现方法可打开 tv_tester.sh 文件查看

CVBS 视频输入测试

执行 `tvd_test_usrptr 4 0 720 576 /tmp/ 4 10000 25`

参数说明：
4 设备名/dev/video4
0 通道号 0 代表 tvin0，1 代表 tvin1
720 x 分辨率
576 y 分辨率
/tmp/ 保存 YUV 文件的路径（当采集次数不等于 10000 时，才会保存文件）
4 模式 固定写法
10000 采集次数 当=10000 时，是将图像通过 CVBS OUT 实时显示，当小于 10000 时是保存成 YUV 文件
25 帧率
测试 CVBS 输入并通过 CVBS 输出显示时，不能运行 CVBS 显示测试，否则显示会有问题

代码路径：platform/framework/auto/sdk_demo/tvd_test