

## Содержание

### **Подготовка программной части для модуля NMS-SM-RK3568, Kernel 5.10 от**

<b>Radxa</b> .....	3
Сборка Linux при помощи Buildroot и Docker .....	3
Подготовка окружения .....	3
Сборка rootfs .....	3
Сборка kernel .....	4
Сборка dtb .....	4
<b>Прошивка SD карты</b> .....	4
<b>Проверка</b> .....	5



# Подготовка программной части для модуля NMS-SM-RK3568, Kernel 5.10 от Radxa

## Сборка Linux при помощи Buildroot и Docker

Для выполнения следующей инструкции, на компьютере должно быть установлено [Docker окружение](#)

### Подготовка окружения

1. Указать параметры для git, если не дает скачать

```
export GIT_SSL_NO_VERIFY=1
```

2. Необходимо скачать папки:

- [buildroot](#)
- [external](#)
- [kernel](#)

```
wget -O buildroot_5.10.tar.gz --show-progress
https://doc.inmys.ru/open?hash=3634d88f557950df694f9b85d1e97463&fn=buildroot_5.10.tar.gz
wget -O external.tar.gz --show-progress
https://doc.inmys.ru/open?hash=4ef09a2f3d4e897a616fa0e314928006&fn=external.tar.gz
git clone https://gitlab.inmys.online/rk3568-kernel-5.10/kernel.git

tar xf buildroot_5.10.tar.gz
tar xf external.tar.gz
```

3. Перейти в папку container и собрать образ контейнера для работы с SDK следующей командой:

```
cd container
sudo docker build -t rk3568 .
```

На выходе будет образ **rk3568**, которым можно собирать все компоненты системы. Далее необходимо перейти в родительский каталог

```
cd ..
```

### Сборка rootfs

Команды сборки

- Использование конфигурации

```
sudo docker run -it -e USER=$USER -e USERID=$UID -v $(pwd):/BR -t rk3568 make rockchip_rk356x_inmys_defconfig -C buildroot
```

- Непосредственно сборка

```
sudo docker run -it -e USER=$USER -e USERID=$UID -v $(pwd):/BR -t rk3568 make -C buildroot
```

Сборка может длиться до часа, в зависимости от конфигурации компьютера, если возникла ошибка, связанная с host-gcc-final-XXX, то повторить команду сборки. Результаты будут лежать в папке buildroot/output/rockchip\_rk356x\_inmys/images.

## Сборка kernel

Команды сборки:

- Использование конфигурации

```
sudo docker run -it -e USER=$USER -e USERID=$UID -v $(pwd):/BR -t rk3568 make ARCH=arm64 CROSS_COMPILE=/BR/buildroot/output/rockchip_rk356x_inmys/host/bin/aarch64-buildroot-linux-gnu- rk3568_inmys_defconfig -C kernel
```

- Сборка образа

```
sudo docker run -it -e USER=$USER -e USERID=$UID -v $(pwd):/BR -t rk3568 make ARCH=arm64 CROSS_COMPILE=/BR/buildroot/output/rockchip_rk356x_inmys/host/bin/aarch64-buildroot-linux-gnu- Image -C kernel
```

Результаты будут лежать в папке kernel/arch/arm64/boot/Image.

## Сборка dtb

Команда сборки

```
sudo docker run -it -e USER=$USER -e USERID=$UID -v $(pwd):/BR -t rk3568 make ARCH=arm64 CROSS_COMPILE=/BR/buildroot/output/rockchip_rk356x_inmys/host/bin/aarch64-buildroot-linux-gnu- rockchip/rk3568-inmys-smarc-evm.dtb -C kernel
```

Результат будет лежать в папке kernel/arch/arm64/boot/dts/rockchip/rk3568-inmys-smarc-evm.dtb.

## Прошивка SD карты

Скачайте архив со скриптами для прошивки [burn.tar.gz](http://burn.tar.gz)

1. Перейти в каталог burn

```
cd burn
```

2. Создать временный каталог для монтирования карты памяти

```
mkdir /mnt/d4
```

3. Скопируйте файл `buildroot/output/rockchip_rk356x_inmys/images/rootfs.ext2` в каталог `burn/rootfs.ext2`
4. Скопируйте файл `kernel/arch/arm64/boot/Image` в `burn/pboot/Image`
5. `kernel/arch/aarch64/boot/dts/rockchip/rk3568-inmys-smarc-evm.dtb` в `burn/pboot/rk3568-inmys-smarc-evm.dtb`
6. Выполните команду:

```
cp burn/pboot/rk3568-inmys-smarc-evm.dtb burn/pboot/rk-kernel.dtb
```

7. Выполните команду

```
sudo ./burn_sd.sh /dev/sdX
```

### **ГДЕ ВМЕСТО X БУКВА КАРТЫ ПАМЯТИ**

## Проверка

Для проверки установки видео-драйверов, выполните следующие команды:

```
uname -a  
gst-inspect-1.0 | grep mpph
```

Корректный результат:

```
# uname -a  
Linux rk3568 5.10.160 #7 SMP Tue Jan 30 14:22:12 UTC 2024 aarch64  
GNU/Linux  
# gst-inspect-1.0 | grep mpph  
rockchipmpp: mpph264enc: Rockchip Mpp H264 Encoder  
rockchipmpp: mpph265enc: Rockchip Mpp H265 Encoder
```

Для записи видео, рекомендуется увеличить чувствительность камеры (для проверки используется `imx219`):

```
v4l2-ctl -c gain=2000
```

Запись в файл `output.mp4`:

```
gst-launch-1.0 -e v4l2src device=/dev/video1 ! videoconvert !  
mpph264enc ! h264parse ! qtmux ! filesink location=output.mp4
```