

Содержание

Руководство пользователя	3
Общая информация	3
Состав программной части Q7_KOMDIV_MK	3
Подготовка окружения	4
Сборка VapeBox	4
Сборка образа системы	4
Прошивка загрузчика в SPI-флешку	5
Linux (Пример для Ubuntu 24.04)	5
Windows	6
Подготовка загрузочной флешки	6
Первый запуск	7

Руководство пользователя

Страница содержит руководство по подготовке программной части, сборке rootfs и ядра Linux для модуля **Q7_KOMDIV_MK**.

Общая информация

Q7_KOMDIV_MK используется как вычислительный модуль в составе встраиваемых и серверных систем. Для работы модуля требуется подготовить загрузчик, ядро Linux и корневую файловую систему.

В рамках данной страницы используется следующее окружение:

Компонент	Версия / описание
Хостовая ОС	Ubuntu 24.04
Rootfs	Buildroot 2025.02.7 + buildroot-external-inmys
Ядро Linux	k5500vk018-linux
Архитектура	KOMDIV-64 (MIPS64)
Сборка rootfs	через Docker-контейнер komdiv-sdk

Состав программной части Q7_KOMDIV_MK

Для работы модуля требуется три основных компонента:

Компонент	Описание	Доступ к исходну кода	Место прошивки
Загрузчик	Barebox	Приватный репозиторий с доступом по токену	SPI FLASH 4MB на модуле
Ядро Linux	Linux с поддержкой k5500vk018	Приватный репозиторий с доступом по токену	eMMC 4GB на модуле/внешняя SD-карта
Rootfs	Buildroot 2025.02.7 + buildroot-external-inmys	Публичный GitHub Inmys	eMMC 4GB на модуле/внешняя SD-карта



На данный момент исходный код распространяется через репозитории с приватным доступом.

Для получения токена обратитесь на почту info@inmys.ru

Подготовка окружения

Сборка выполняется на **Ubuntu 24.04**.

Перед началом работы необходимо обновить список пакетов и установить кросс-компилятор для MIPS (требуется для сборки barebox):

```
sudo apt update
sudo apt install -y cpp-mips64el-linux-gnuabi64
```

Также предполагается, что на хостовой машине установлен Docker.

Сборка BareBox



Исходники загрузчика будут доступны в августе

```
git clone gitlab.inmys.online ...
make ARCH=mips CROSS_COMPILE=mips64el-linux-gnuabi64- srisa_defconfig
make ARCH=mips CROSS_COMPILE=mips64el-linux-gnuabi64- -j24
```

Результат сборки находится по пути `images/barebox-rbm_clk5500vk018-el.img`

Сборка образа системы

Введите логин и токен вместо XXXXX и YYYYYY и выполните команду:

```
LOGIN="XXXXX"; TOKEN="YYYYY"; printf "machine
gitlab.inmys.online\nlogin %s\npassword %s\n" "$LOGIN" "$TOKEN" >
.gitlab.netrc
```

Выполните следующие команды:

```
mkdir -p container
cd container
wget
https://nextcloud.inmys.online/nextcloud/index.php/s/ZJFE4wx3pMwf9yn/do
wnload -O Dockerfile
sudo docker build -t komdiv-sdk .
cd ..
wget https://buildroot.org/downloads/buildroot-2025.02.7.tar.gz
tar -xf buildroot-2025.02.7.tar.gz
git clone https://github.com/inmys/buildroot-external-inmys.git -b
nms-q7-k5500vk018
sudo docker run -it -e USER=$USER -e USERID=$UID -v $(pwd):/BR -v
```

```
"$(pwd)/.gitlab.netrc:/root/.netrc:ro" --cpus=12 -t komdiv-sdk bash
```

В Docker-контейнере выполнить:

```
make BR2_EXTERNAL=$PWD/buildroot-external-inmys -C buildroot-2025.02.7  
0=$PWD/output br_defconfig  
cd output  
export FORCE_UNSAFE_CONFIGURE=1  
make -j12
```

После завершения сборки необходимо выйти из контейнера сочетанием клавиш Ctrl + D

Результаты сборки находятся в директории output/images/

Прошивка загрузчика в SPI-флешку

Осуществляется программатором FT2232H-56Q Mini Module с установленным переходником FT2232H-56Q Mini Module_Adapter

Необходимо подключить программатор FT2232H-56Q Mini Module к ПК и подключить прищепку к плате, красным проводом вверх. Правильное положение прищепки достигается, если плата Q7_KOMDIV_MK установлена в отладочную плату NMS-Q7-EVM-MINI



Linux (Пример для Ubuntu 24.04)

Установка зависимостей:

```
sudo apt update && sudo apt install flashrom -y
```

Прошивка производится с помощью скрипта [burn-barebox.sh](#):

```
chmod +x burn-barebox.sh  
./burn-barebox.sh barebox-rbm_clk5500vk018-el.img
```

```
4+0 records in  
8192+0 records out  
4194304 bytes (4,2 MB, 4,0 MiB) copied, 0,0105851 s, 396 MB/s  
1881+1 records in  
1881+1 records out  
963180 bytes (963 kB, 941 KiB) copied, 0,0058463 s, 165 MB/s  
flashrom unknown on Linux 6.14.0-28-generic (x86_64)  
flashrom is free software, get the source code at https://flashrom.org  
  
Using clock_gettime for delay loops (clk_id: 1, resolution: 1ns).
```

```
Found Winbond flash chip "W25Q32.V" (4096 kB, SPI) on ft2232_spi.  
===  
This flash part has status UNTESTED for operations: WP  
The test status of this chip may have been updated in the latest  
development  
version of flashrom. If you are running the latest development version,  
please email a report to flashrom@flashrom.org if any of the above  
operations  
work correctly for you with this flash chip. Please include the  
flashrom log  
file for all operations you tested (see the man page for details), and  
mention  
which mainboard or programmer you tested in the subject line.  
Thanks for your help!  
Reading old flash chip contents... done.  
Erasing and writing flash chip... Erase/write done.  
Verifying flash... VERIFIED.
```

Windows

- Скачать и установить flashrom для windows
https://github.com/therealdreg/flashrom_build_windows_x64
- Через Zadig заменить драйвер ftdi на winusb. Подходит вот эта инструкция:
<https://docs.mikron.ru/wiki/dev-tools/debuggers/ft2232.html>
- Открыть через командную строку flashrom.exe с правами администратора

Определение флешки программатором:

```
.\flashrom.exe -p ft2232_spi:type=2232H,port=A,divisor=4
```

Чтение файла из памяти:

```
.\flashrom.exe --progress -V -c "W25Q32JV" -p  
ft2232_spi:type=2232H,port=A,divisor=4 -r filename.bin
```

Запись файла в память:

```
.\flashrom.exe --progress -V -c "W25Q32JV" -p  
ft2232_spi:type=2232H,port=A,divisor=4 -w filename.bin
```

Подготовка загрузочной флешки

Скачать ресурсы: <https://disk.360.yandex.ru/d/11mHF4DiqqxNxQ>

Для первого запуска необходимо подготовить загрузочную флешку. Для прошивки флешки необходимо, чтобы в той же директории, что и скрипт, находились vmlinuz и файл .dtb.

Прошивка USB-флешки/SD-карты

```
sudo chmod +x burn-boot-usb.sh  
sudo ./burn-boot-usb.sh /dev/sdX #где X - буква прошиваемой флешки
```

В конце ожидайте вывод: Флешка подготовлена

Первый запуск

В Varebox установлен приоритет загрузки на USB или SD-карточку. Поэтому если вставлен внешний накопитель, то загрузка будет происходить с него.

- Вставьте флешку в отладочную плату и подайте питание
- Дождитесь конца загрузки и зайдите под root/root
- Установите уникальные MAC-адреса, разметьте EMMC и перенесите образ на EMMC следующими командами:

```
./auto-burn-macs.sh  
  
echo 'n p 1 +1G n p 2 +1G n p 3 +1G w' | tr ' ' '\n' | fdisk -u  
/dev/mmcblk0  
  
mke2fs /dev/mmcblk0p1  
mke2fs /dev/mmcblk0p2  
mke2fs /dev/mmcblk0p3  
  
mount /dev/mmcblk0p1 /boot  
mount /dev/sdc1 /mnt  
  
cp /mnt/k5500vk018-inmys-q7.dtb /boot  
cp /mnt/vmlinuz /boot  
sync
```

После того как команды были успешно выполнены, можно выключать питание и работать без флешки. Платформа готова к работе.