

Оглавление

Скачивание необходимых исходников	1
Запуск и настройка виртуальной машины.....	2
Установка git	4
Установка текстового редактора Gedit	4
Увеличение максимального количества “loop device”.....	5
Компиляция образов	6
Тестирование	12
Заключение	16

Скачивание необходимых исходников

Armbian_OK3568.zip архив со всеми необходимыми файлами доступен по [ссылке](#).

Архив содержит:

- образ виртуальной машины;
- три собранных образа (Desktop, Standard, Minimal);
- дерево устройств и исправленный Makefile для ядра;
- дерево устройств и defconfig для Uboot;
- специальный конфигурационный файл ОК3568-С.csc для проекта Armbian;
- папку с файлами для создания переходной платы для подключения LVDS дисплея ZW-T101HIEWA-03 в формате KiCad (принципиальная схема, список компонентов, файлы дизайна печатной платы, файлы Gerber);
- инструкцию по использованию переходной платы.

Запуск и настройка виртуальной машины

После скачивания и распаковки архива необходимо запустить виртуальную машину. Для этого подойдут такие программы, как VMware workstation или Virtualbox. Далее рассмотрим вариант работы с программой VMware workstation.

После установки и запуска VMware workstation:

- Создаем новую виртуальную машину выбрав «Create a New Virtual»



Create a New Virtual Machine

Create a new virtual machine, which will then be added to the top of your library.

- Указать путь до файла ubuntu-22.04.2-desktop-amd64.iso из нашего архива

● Installer disc image file (iso):
D:\ubuntu-22.04.2-desktop-amd64.iso Browse...
Ubuntu 64-bit 22.04.2 detected.
This operating system will use Easy Install. [\(What's this?\)](#)

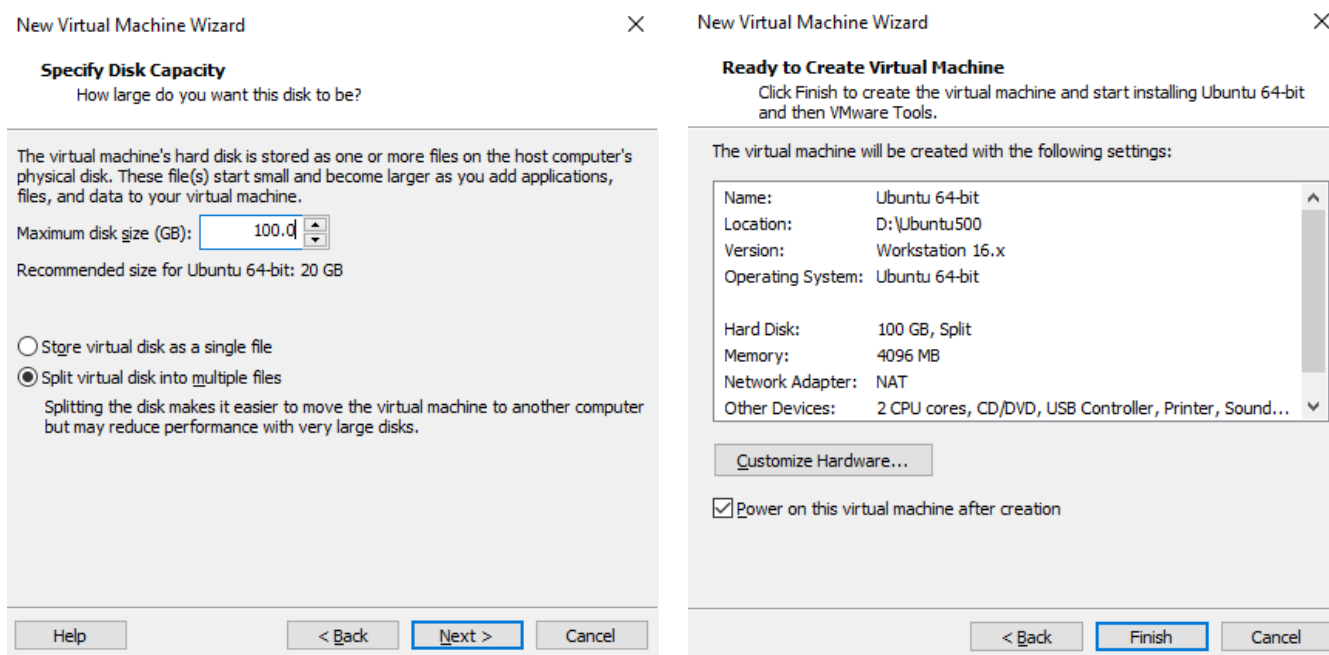
- Ввести предпочтительные имя пользователя и пароль

Personalize Linux
Full name:
User name:
Password:
Confirm:

- Дать имя виртуальной машине и создать папку на диске для хранения

Virtual machine name:
Ubuntu 64-bit
Location:
F:\Ubuntu500GB Browse...

- **Задать размер виртуальной машины и настроить аппаратную часть**



После нажатия кнопки **Finish** начнётся процесс первого запуска Ubuntu. Следуем указаниям системы, пока не попадём на рабочий стол.

Установка git

Запускаем терминал и вводим команду «sudo apt install git»

```
lcgitc-virtual-machine:~/Downloads$ sudo apt install git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  git
0 upgraded, 1 newly installed, 0 to remove and 178 not upgraded.
Need to get 3 165 kB of archives.
After this operation, 18,9 MB of additional disk space will be used.
Get:1 http://ru.archive.ubuntu.com/ubuntu jammy-updates/main amd64 git amd64 1:2.34.1-1ubuntu1.11 [3 165 kB]
Fetched 3 165 kB in 2s (1 570 kB/s)
Selecting previously unselected package git.
(Reading database ... 211955 files and directories currently installed.)
Preparing to unpack .../git_1%3a2.34.1-1ubuntu1.11_amd64.deb ...
Unpacking git (1:2.34.1-1ubuntu1.11) ...
Setting up git (1:2.34.1-1ubuntu1.11) ...
lcgitc-virtual-machine:~/Downloads$
```

Установим удобный текстовый редактор, который далее будем использовать для настройки образа виртуальной машины.

Установка текстового редактора Gedit

Вводим команду «sudo apt install gedit»

```
lcgitc-virtual-machine:~/Downloads$ sudo apt install gedit
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  gedit-plugins
The following NEW packages will be installed:
  gedit
0 upgraded, 1 newly installed, 0 to remove and 178 not upgraded.
Need to get 434 kB of archives.
After this operation, 1 827 kB of additional disk space will be used.
Get:1 http://ru.archive.ubuntu.com/ubuntu jammy/main amd64 gedit amd64 41.0-3 [434 kB]
Fetched 434 kB in 1s (822 kB/s)
Selecting previously unselected package gedit.
(Reading database ... 212673 files and directories currently installed.)
Preparing to unpack .../gedit_41.0-3_amd64.deb ...
Unpacking gedit (41.0-3) ...
Setting up gedit (41.0-3) ...
update-alternatives: using /usr/bin/gedit to provide /usr/bin/gnome-text-editor (gnome-text-editor) in auto mode
Processing triggers for mailcap (3.70+nmubuntu1) ...
Processing triggers for desktop-file-utils (0.26-1ubuntu3) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu3) ...
lcgitc-virtual-machine:~/Downloads$
```

Увеличение максимального количества “loop device”

Необходимо для корректной работы компилятора Armbian*

- Вводим «sudo gedit /etc/default/grub»

```
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
# info -f grub -n 'Simple configuration'

GRUB_DEFAULT=0
GRUB_TIMEOUT_STYLE=hidden
GRUB_TIMEOUT=0
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash max loop=64"
GRUB_CMDLINE_LINUX="find_preseed=/preseed.cfg auto noprompt priority=critical locale=en_US"

# Uncomment to enable BadRAM filtering, modify to suit your needs
# This works with Linux (no patch required) and with any kernel that obtains
# the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)
#GRUB_BADRAM="0x01234567,0xfefefefefe,0x89abcdef,0xefefefef"

# Uncomment to disable graphical terminal (grub-pc only)
#GRUB_TERMINAL=console

# The resolution used on graphical terminal
# note that you can use only modes which your graphic card supports via VBE
# you can see them in real GRUB with the command `vbeinfo'
#GRUB_GFXMODE=640x480

# Uncomment if you don't want GRUB to pass "root=UUID=xxx" parameter to Linux
#GRUB_DISABLE_LINUX_UUID=true

# Uncomment to disable generation of recovery mode menu entries
#GRUB_DISABLE_RECOVERY="true"

# Uncomment to get a beep at grub start
#GRUB_INIT_TUNE="480 440 1"
```

- Строку «GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"» исправляем на «GRUB_CMDLINE_LINUX_DEFAULT="quiet splash max_loop=64"».
- Прописываем команду «sudo update-grub» и перезапускаем систему.

Компиляция образов

После перезапуска заходим в папку Downloads и запускаем из неё командную строку.

- Вводим `git clone --depth=1 --branch=main https://github.com/armbian/build`
- Переносим папку Kernel, папку Uboot и файл ОК3568-С.csc из нашего архива на виртуальную машину в папку Downloads.



*В дальнейшем “.../Downloads/” в указании пути до файла или папки будет обозначать путь до папки Downloads в вашей системе. В нашем случае это “/home/ic/Downloads/”.

- Копируем файл ОК3568-С.csc в папку `.../Downloads/build/config/boards`
- Удаляем файл:
`.../Downloads/build/patch/kernel/rk35xx-vendor-6.1/0000.patching_config.yaml`
- Переходим в папку build и запускаем оттуда командную строку.
- Прописываем `./compile.sh uboot-patch BOARD=ОК3568-С BRANCH=vendor RELEASE=jammy` и вводим `sudo` пароль
- Дожидаемся, когда компиляция остановится:

```
[ ⚠ ] Press <ENTER> after you are done [ edit ]
Press ENTER to show a preview of your patch,
```

- Открываем новую вкладку консоли и копируем дерево устройств для uboot

```
«sudo cp .../Downloads/Uboot/ОК3568-С.dts
.../Downloads/build/cache/sources/u-boot-worktree/u-boot-rockchip64/stable-4.19-rock3/arch/arm/dts»
```


- Копируем defconfig в исходники uboot:
«sudo cp ../Downloads/Uboot/OK3568-rk3568_defconfig
../Downloads/build/cache/sources/u-boot-worktree/u-boot-
rockchip64/stable-4.19-rock3/configs»
- Возвращаемся в первую вкладку, где происходила компиляция.
Нажимаем enter, затем вводим букву “y” и нажимаем enter.
Результатом станет патч, который находится в папке
../Downloads/OK3568_Armbian/build/output/patch
- Приступаем к патчу для ядра Linux.

Вводим «./compile.sh kernel-patch BOARD=OK3568-C BRANCH=vendor
RELEASE=jammy»

По аналогии с uboot ожидаем остановки компиляции и открываем
новую вкладку консоли.

A screenshot of a terminal window with a dark background. It shows a yellow warning icon in a square followed by the text: "[Warning] Press <ENTER> after you are done [edit] Press ENTER to show a preview of your patch,". The text is in a light-colored monospace font.

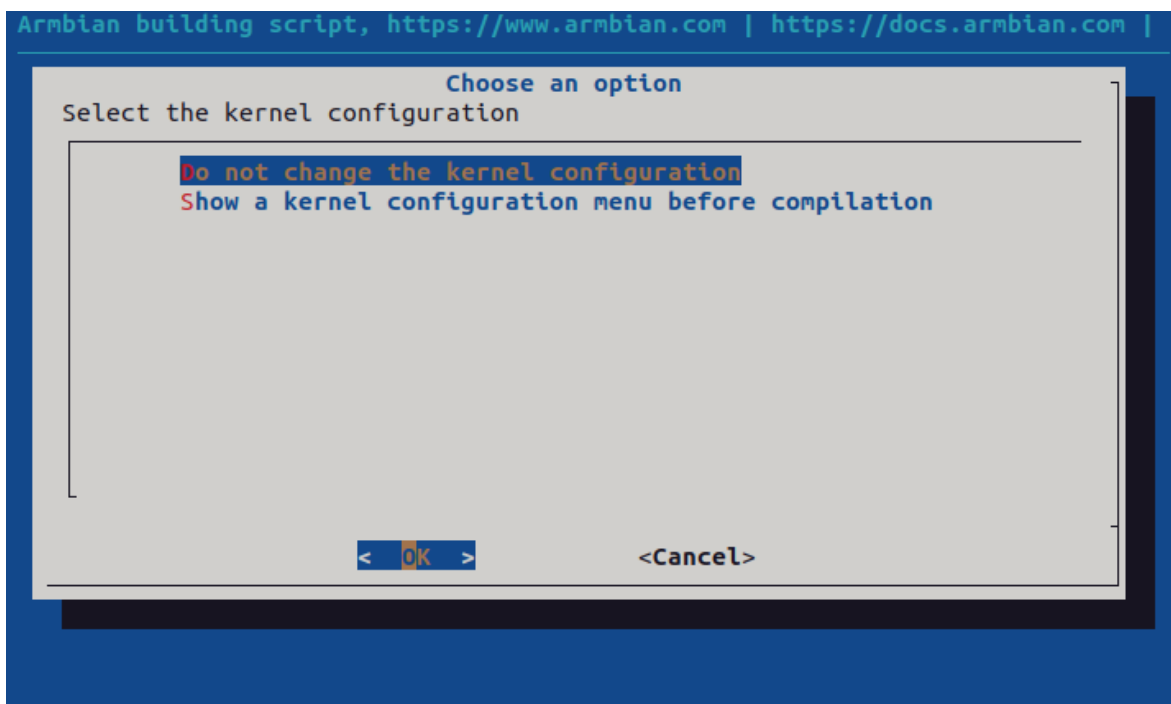
- Копируем файл дерева устройств.
sudo cp ../Downloads/Kernel/OK3568-C.dts
../Downloads/build/cache/sources/linux-kernel-
worktree/6.1__rk35xx__arm64/arch/arm64/boot/dts/rockchip
- Копируем исправленный Makefile, чтобы наше дерево устройств было
скомпилировано в .dtb файл.

«sudo cp ../Downloads/Kernel/Makefile
../Downloads/build/cache/sources/linux-kernel-
worktree/6.1__rk35xx__arm64/arch/arm64/boot/dts/rockchip»

- Возвращаемся в первую вкладку, где происходила компиляция.
Нажимаем enter, затем вводим букву “y” и нажимаем enter. Результатом
станет патч, который находится в папке ../Downloads/build/output/patch

- Копируем получившиеся патчи.
 1. Патч «u-boot-rk35xx-vendor.patch» в папку
.../Downloads/build/userpatches/u-boot/legacy
 2. Патч «kernel-rk35xx-vendor.patch» в папку
.../Downloads/build/userpatches/kernel/rk35xx-vendor-6.1
- Начинаем процесс компиляции

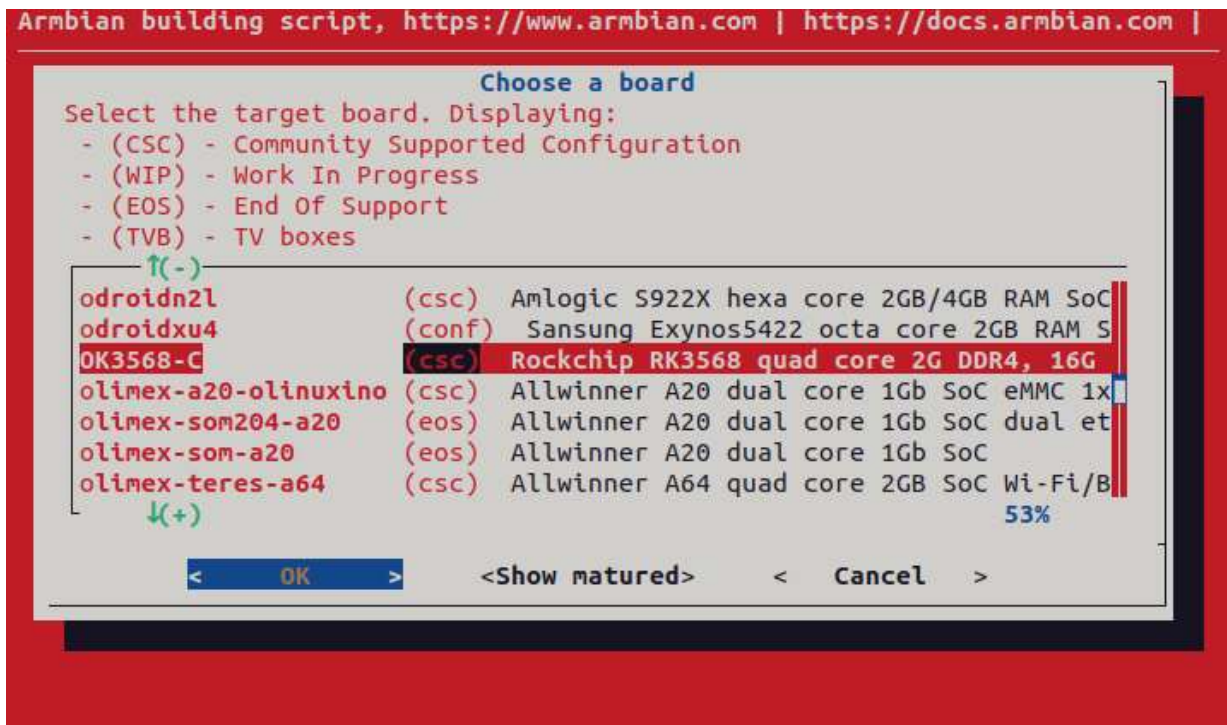
В папке build вводим ./compile.sh и sudo пароль.



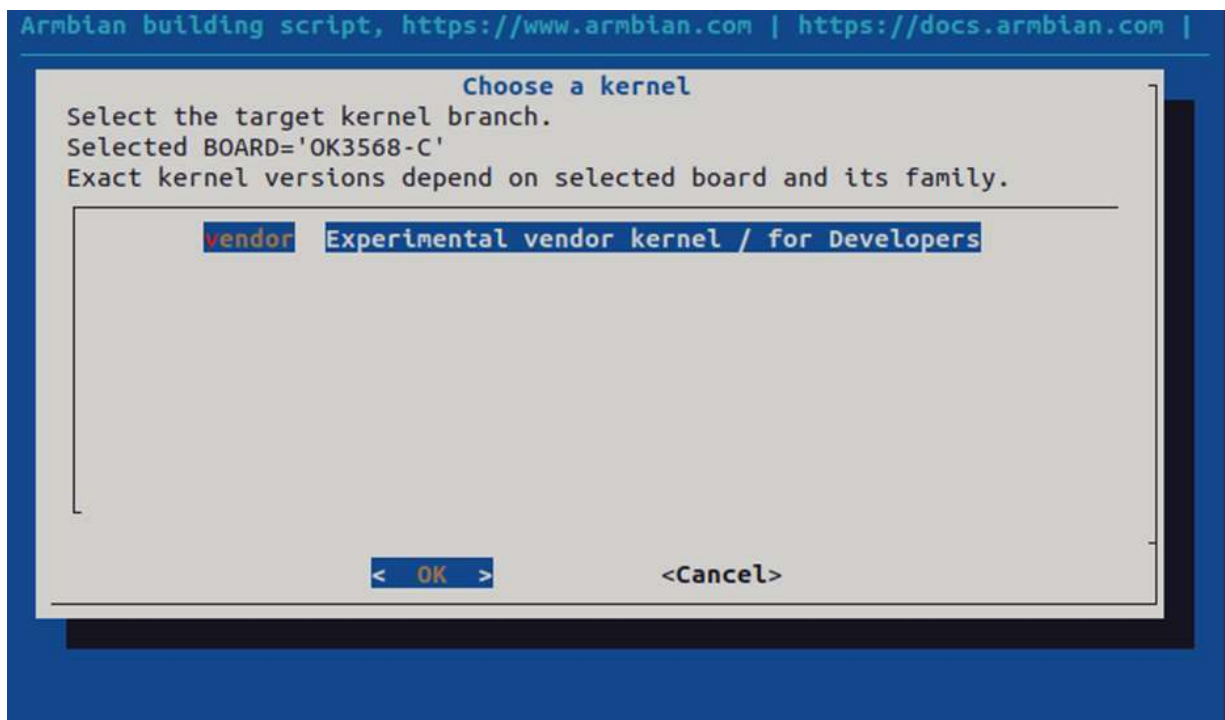
Если выбрать “Show a kernel configuration...”, menuconfig откроется перед началом компиляции ядра.

Также конфигурацию ядра можно отредактировать вручную, открыв файл .../Downloads/build/config/kernel/linux-rk35xx-vendor.config

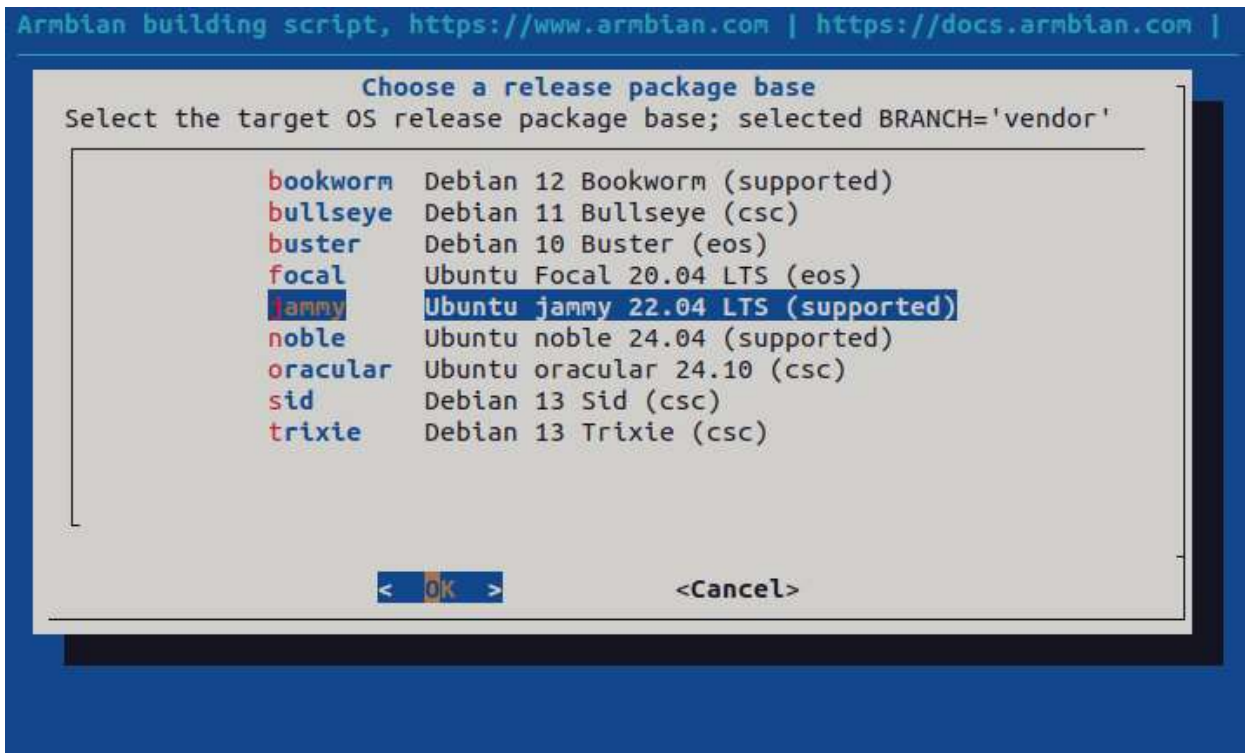
- Далее нужно найти нашу плату, она находится в списке “CSC/WIP/EOS/TVB”



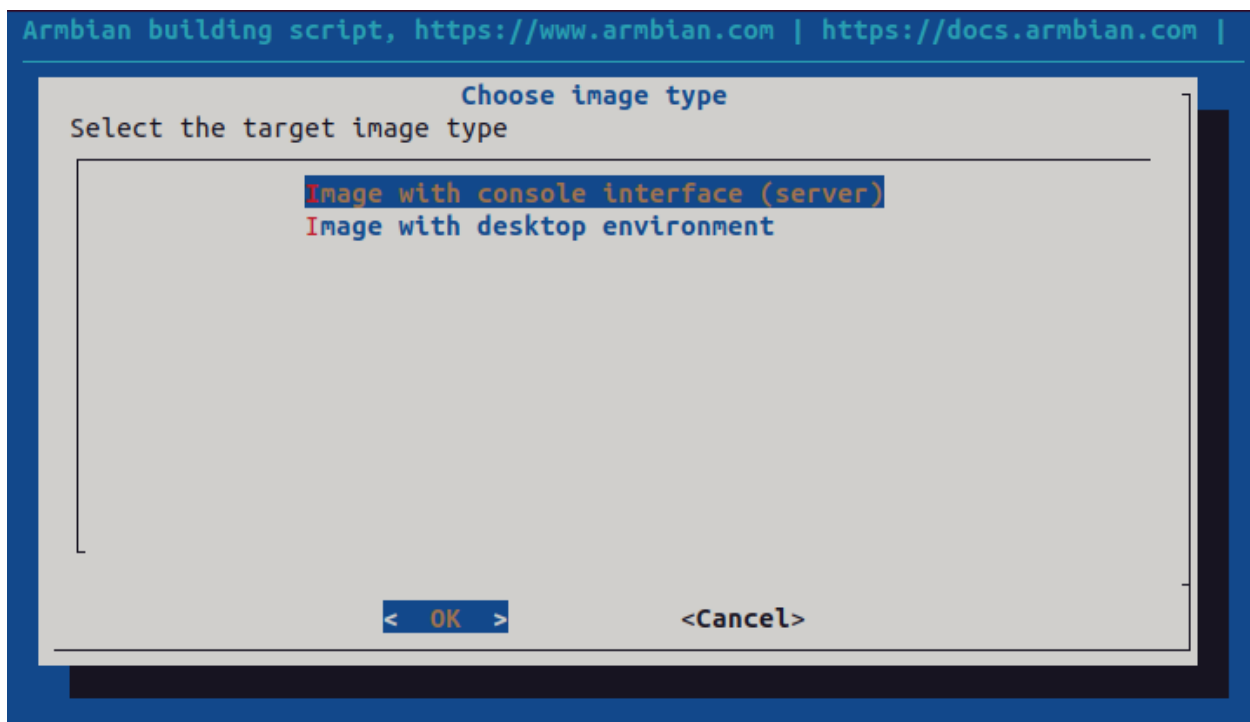
- Выбираем “vendor” опцию, это и будет наш kernel-6.1



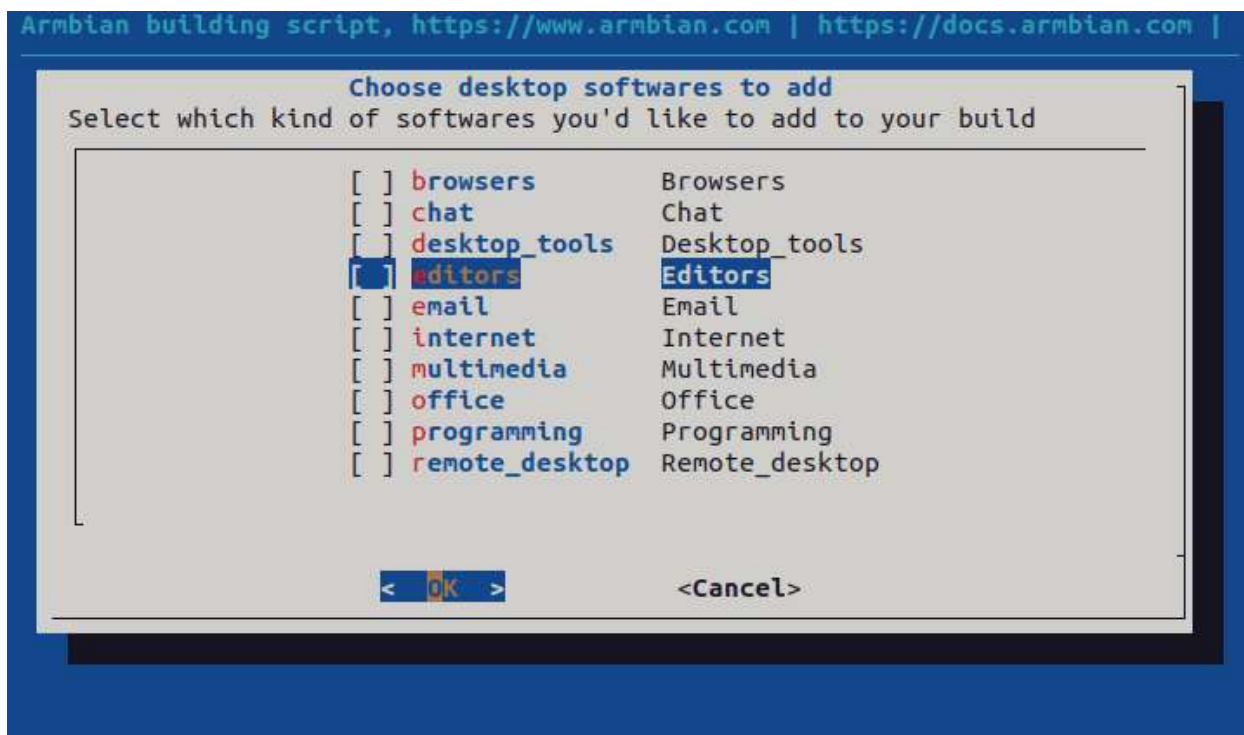
- Выбираем один из доступных дистрибутивов Ubuntu/Debian



- Выбираем между серверной и десктопной версией



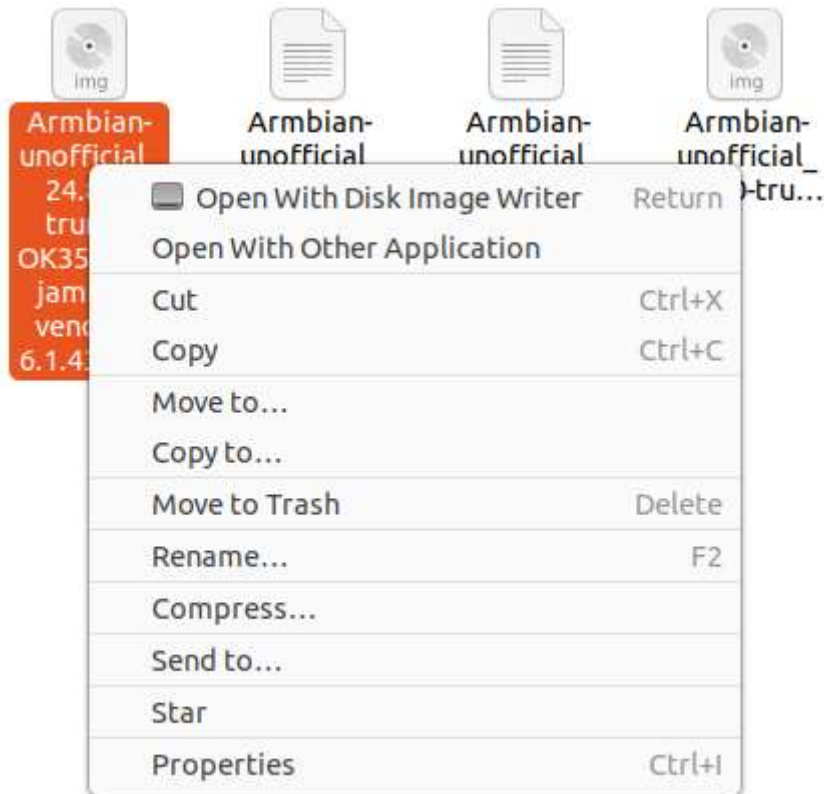
В случае десктопной версии нам будет предложено установить среду рабочего стола и дополнительный софт (браузер, офисные программы, почтовый клиент и т.д)



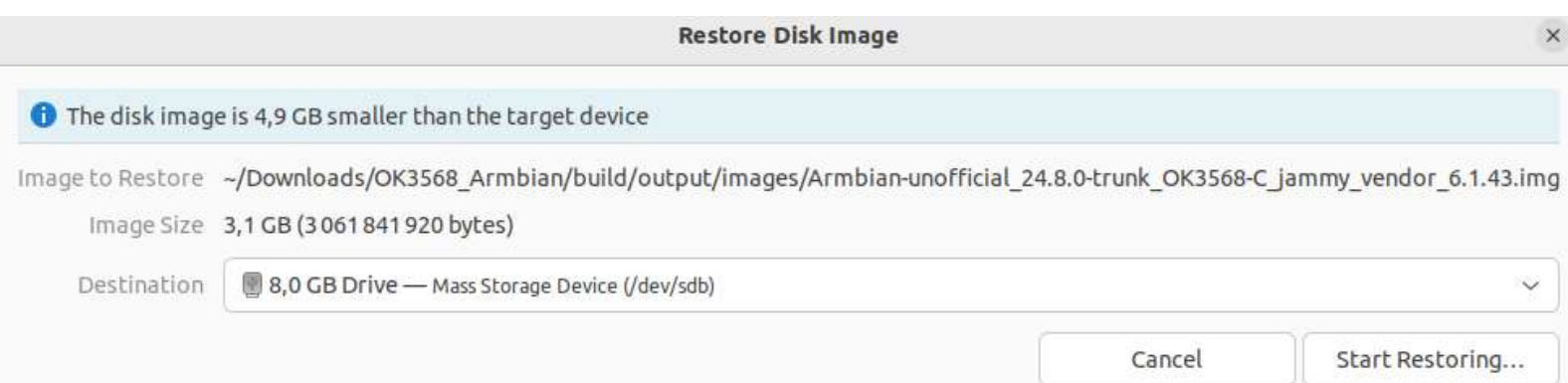
- После компиляции в папке `.../Downloads/ build/output/images` появится образ, который уже можно загружать на SD-карту.

Тестирование

- Загружаем образ на SD-карту при помощи встроенного в Ubuntu приложения Disk Image Writer



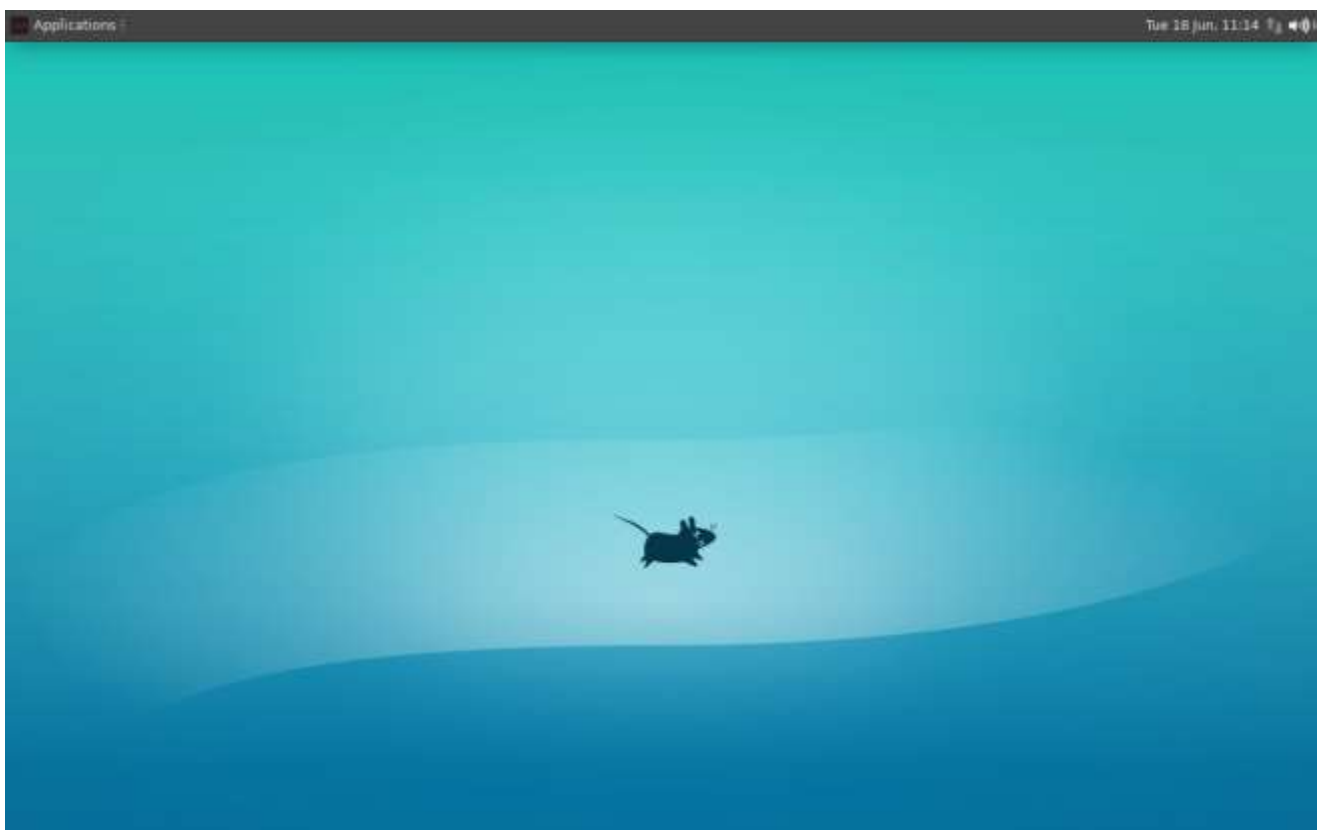
- Выбираем пустую SD-карту, вставленную через кард-ридер, нажимаем "Start Restoring". Дожидаемся окончания загрузки.



- Вставляем SD-карту в слот, подключаем дисплей ZW-T101HIEWA-03 через разъём P11 на отладочной плате.
Для подключения дисплея мы разработали специальную переходную плату со встроенными источниками питания для подсветки и микросхем дисплея. Исходники платы и инструкцию по установке можно найти в архиве Armbian_OK3568.zip.

При первом запуске система предлагает создать пользователя, задать root-пароль, выбрать часовой пояс.

После этого запустится рабочий стол или консоль в зависимости от версии дистрибутива



На этом этапе проверяем работоспособность USB-разъемов – подключаем мышку, клавиатуру, USB-накопители.

- Проверим инициализацию Ethernet-трансиверов через команду `dmesg`

```

Applications  ic@ОК3568-С: ~/Desktop
ic@ОК3568-С: ~/Desktop
ic@ОК3568-С: ~/Desktop
18.316054] rk_gmac-dwmac fe010000.ethernet eth0: Register MEM_TYPE_PAGE_POOL RxQ-0
18.317475] phy_rtl8211f_led_fixup in
18.327968] phy_rtl8211f_led_fixup in val=0x0a42
18.331928] phy_rtl8211f_led_fixup in
18.332098] phy_rtl8211f_led_fixup in val=0x0000
18.421325] rk_gmac-dwmac fe010000.ethernet eth0: PHY [stmmac-1:00] driver [RTL8211F Gigabit Ethernet]
(irq=POLL)
18.421977] dwmac4: Master AXI performs any burst length
18.422022] rk_gmac-dwmac fe010000.ethernet eth0: No Safety Features support found
18.422060] rk_gmac-dwmac fe010000.ethernet eth0: IEEE 1588-2008 Advanced Timestamp supported
18.422586] rk_gmac-dwmac fe010000.ethernet eth0: registered PTP clock
18.432806] rk_gmac-dwmac fe010000.ethernet eth0: configuring for phy/rgmii link mode
18.549603] rk_gmac-dwmac fe2a0000.ethernet eth1: Register MEM_TYPE_PAGE_POOL RxQ-0
18.550793] phy_rtl8211f_led_fixup in
18.550997] phy_rtl8211f_led_fixup in val=0x0000
18.551674] phy_rtl8211f_led_fixup in
18.551840] phy_rtl8211f_led_fixup in val=0x0000
18.641331] rk_gmac-dwmac fe2a0000.ethernet eth1: PHY [stmmac-0:00] driver [RTL8211F Gigabit Ethernet]
(irq=POLL)
18.642002] dwmac4: Master AXI performs any burst length
18.642061] rk_gmac-dwmac fe2a0000.ethernet eth1: No Safety Features support found
18.642098] rk_gmac-dwmac fe2a0000.ethernet eth1: IEEE 1588-2008 Advanced Timestamp supported
18.642581] rk_gmac-dwmac fe2a0000.ethernet eth1: registered PTP clock
18.645838] rk_gmac-dwmac fe2a0000.ethernet eth1: configuring for phy/rgmii link mode
21.504684] dwhdmi-rockchip fe0a0000.hdmi: Rate 0 missing; compute N dynamically
21.505400] dwhdmi-rockchip fe0a0000.hdmi: Rate 0 missing; compute N dynamically
21.534140] dwhdmi-rockchip fe0a0000.hdmi: Rate 0 missing; compute N dynamically
21.535119] dwhdmi-rockchip fe0a0000.hdmi: Rate 0 missing; compute N dynamically
22.276976] Bluetooth: BNEP (Ethernet Emulation) ver 1.3

```

Видим, что трансиверы RTL8211F успешно инициализированы. Скачиваем программу для дальнейшего тестирования аппаратного графического ускорителя. Вводим: `sudo apt install glmark2-es2`.

- Запускаем glmark2-es2. Видим, что программа использует ускоритель Mali-G52 для обработки 3D-объектов.

```

lc@OK3568-C:~$ glmark2-es2
=====
glmark2 2021.02
=====
OpenGL Information
GL_VENDOR:      Mesa
GL_RENDERER:    Mali-G52 r1 (Panfrost)
GL_VERSION:     OpenGL ES 3.1 Mesa 23.2.1-ubuntu3.1-22.04.2
=====
[build] use-ybo=false: FPS: 245 FrameTime: 4.082 ms
[build] use-ybo=true: FPS: 275 FrameTime: 3.636 ms
[texture] texture-filter=nearest: FPS: 424 FrameTime:
[texture] texture-filter=linear: FPS: 439 FrameTime:
[texture] texture-filter=mipmap: FPS: 452 FrameTime:
[shading] shading=gouraud: FPS: 205 FrameTime: 4.878
[shading] shading=blinn-phong-inf:
  
```

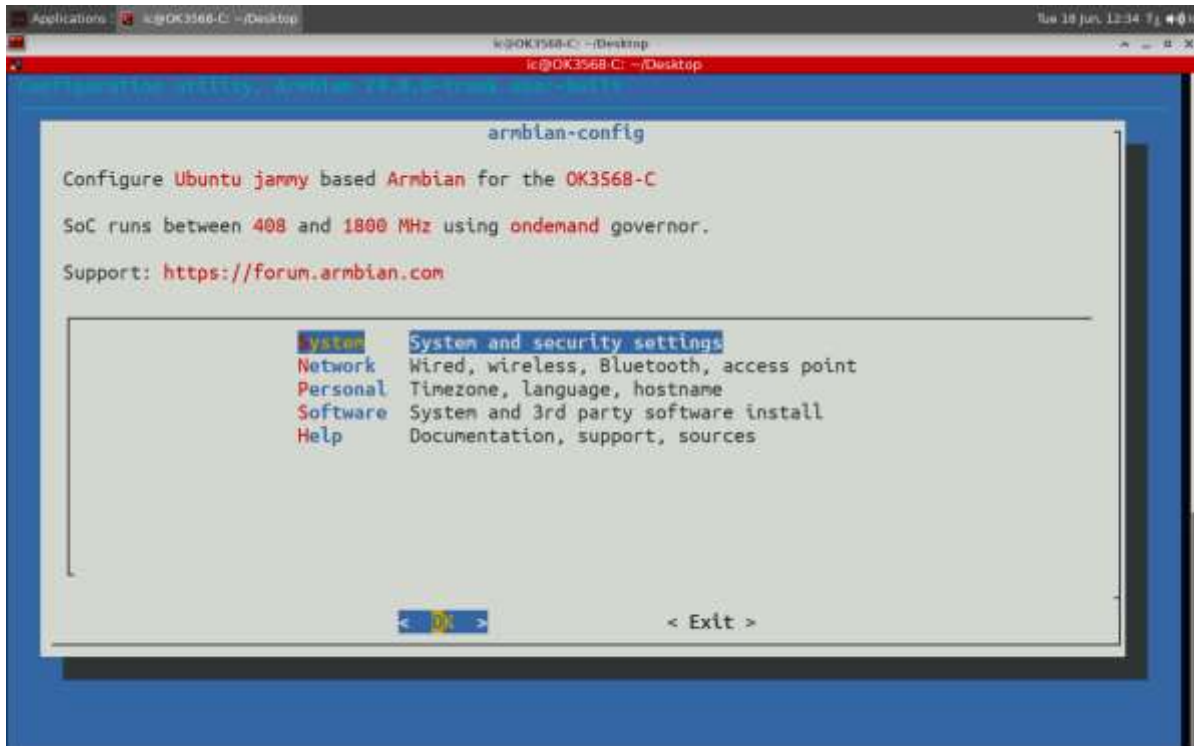
- Аппаратный видео-декодер H264 проверим через плеер MPV

```

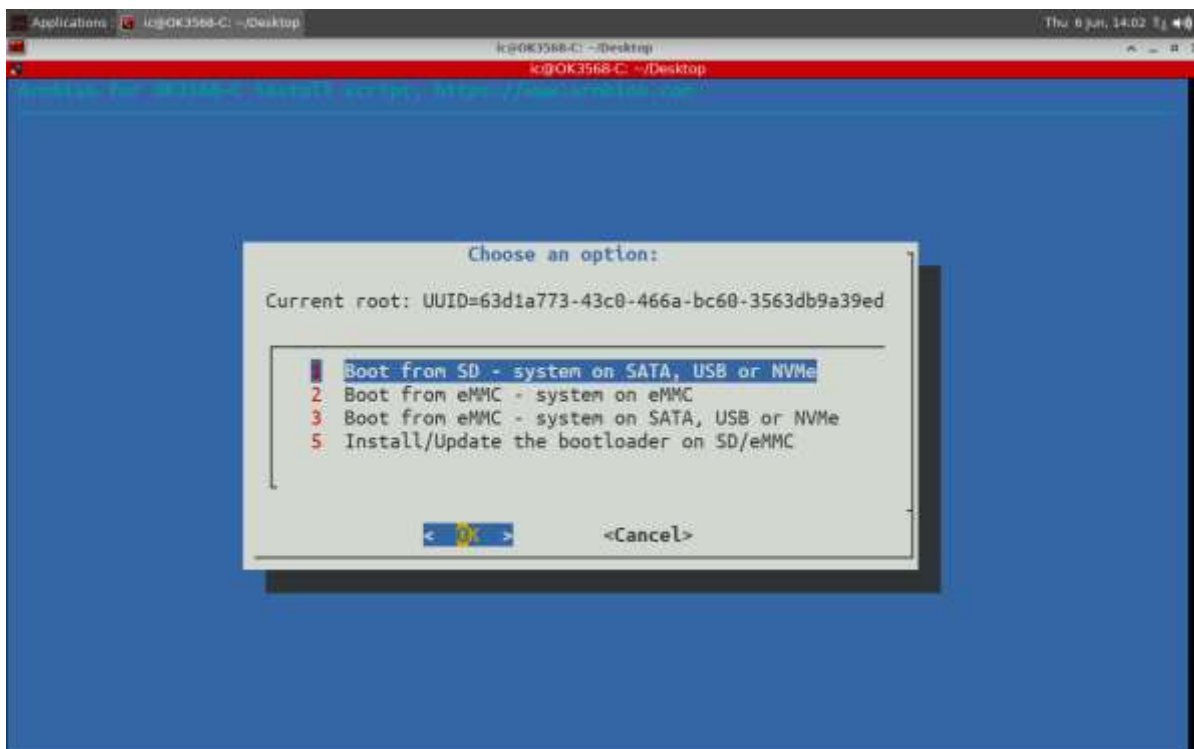
lc@OK3568-C:~/Downloads$ mpv winstar.mp4
(+) Video --vid=1 (*) (h264 1280x800 29.970fps)
(+) Audio --aid=1 --alang=eng (*) (aac 2ch 48000Hz)
Cannot load libcuda.so.1
AO: [pulse] 48000Hz stereo 2ch float
VO: [gpu] 1280x800 yuv420p
AV: 00:01:51 / 00:04:36 (40%) A-V: 0.000 Dropped: 21
  
```

Заключение

- В качестве последнего шага запишем получившийся образ в eMMC. Вводим команду `sudo armbian-config`



Переходим в System/Install и выбираем интересующую нас опцию



После выбора интересующей опции, Armbian-config сообщит нам, что вся информация из eMMC-памяти будет удалена и предложит выбрать формат файловой системы для eMMC-памяти.

После установки и перезагрузки системы все необходимые настройки завершены.

*Автор: Котельников Григорий, инженер по применению, ГК
Промэлектроника. kotelnikov.grigoriy@promelec.ru.*