

АО «НПК «АТРОНИК»

УТВЕРЖДЕН

РОФ.РУСВ.00001- 01 32 01-ЛУ

МОДУЛИ ЦЕНТРАЛЬНОГО ПРОЦЕССОРА МЦП1502/МЦП905

Руководство системного программиста

РОФ.РУСВ.00001- 01 32 01

**Версия 1.1
Листов 63**

Инв. № подл.	Подп. и дата
Инв. № дубл.	Взам. инв. №
Подп. и дата	Подп. и дата

2024

Содержание

1	Введение	4
1.1	Структура ПО поставки	4
1.2	Версия документа. Описание изменений	4
1.3	Справочная информация о доступных ресурсах референсной платформы	5
1.3.1	Доступные ресурсы референсной платформы	5
1.3.2	Перечень интерфейсов, выведенных в МЦП1502.....	6
2	Использование терминального режима отладки	9
2.1	USB to Serial adapter	10
2.2	Параметры подключения. Установка драйвера для Windows, запуск терминала	11
2.3	Использование терминального подключения под Ubuntu	13
3	Компиляция системного ПО. Архив комплекта разработчика	13
3.1	Ядро Linux	15
3.2	U-boot.....	16
3.3	Дерево устройств (dts).....	16
4	Загрузка системного программного обеспечения	16
4.1	Загрузка со встроенной eMMC-flash	18
4.2	Загрузка с microSD-flash	18
4.3	Загрузка с SSD	19
4.4	Загрузка с USB.....	19
5	Установка (обновление) системного программного обеспечения	19
5.1	Структура встроенной eMMC-flash	19
5.2	MaskROM upgrade mode из Linux	19
5.3	Loader upgrade mode	19
5.4	MicroSD-flash upgrade mode	20
5.5	Создание загрузочной microSD-flash.....	20
5.6	Использование загрузочной microSD-flash.....	23
6	Установка (обновление) пользовательского программного обеспечения	24
6.1	Обновление с USB-flash/microSD-flash	24
6.2	Обновление по сети через ssh.....	24
7	Функциональные блоки, интерфейсы и подключаемые устройства	24
7.1	CSI Camera.....	25
7.1.1	Full-mode (одна камера на 4 канала данных).....	26
7.1.2	Split-mode (две камеры по 2 канала данных на каждую)	26
7.2	Сети Ethernet	28
7.3	I2C-шина.....	28
7.3.1	Доступные I2C-контроллеры	29
7.3.2	Поиск устройств на I2C-шинах.....	29
7.3.3	Конфигурирование I2C-драйверов устройств	29
7.3.4	Чтение/запись информации через I2C интерфейс.....	30
7.4	Видеоподсистема GPU	30
7.4.1	HDMI	35
7.4.2	LVDS. На примере панели G070ACE-L01	36
7.4.2.1	Конфигурирование MIPI DTS	37
7.4.2.2	Pins configurations	37

7.4.2.4	Display timing configurations	38
7.4.3	eDP	39
7.4.4	DSI.....	40
7.5	GPIO.....	41
7.5.1	Описание вывода в dts	41
7.5.2	Программное управление линией GPIO (на примере USER-LED)	42
7.5.2.1	Програмное управление выводом драйвером gpio-leds.....	43
7.5.2.2	Программное управление выводом драйвером GPIO	44
7.5.3	Занятые выходы GPIO	45
7.6	SPI-шина.....	46
7.6.1	Подключение к шине SPI	48
7.6.2	Установленные устройства на шине SPI.....	49
7.7	UART	49
7.7.1	Список выведенных на разъёмы UART	50
7.7.2	Подключение ПК к выведенным на разъёмы UART	51
7.8	CAN. Описание и схема подключения	53
7.9	RTC. Настройка и получение текущего времени	55
7.10	Аудио I2S. Схема подключения микрофона и наушников.....	55
7.11	Измерение напряжения SARADC_Vin (mikroBUS)	58
7.12	Выдача напряжения через PWM (mikroBUS)	59
7.13	Получение температуры кристалла от Temperature sensor	60
7.14	Сторожевой таймер.....	61
7.15	Сопроцессор нейросетей NPU	62
7.16	PCI-E.....	62
7.17	SATA	62

1 Введение

Данный документ предназначен для разработчиков ПО, которые самостоятельно конфигурируют модуль процессора МЦП1502 и производят сборку образа операционной системы. Приведенные в документе примеры конфигурирования функциональных узлов относятся к модулю процессора МЦП905. Модуль МЦП905 состоит из модуля МЦП1502 и платы-носителя, на которую выведены интерфейсы модуля МЦП1502. МЦП905 используется в качестве референсной платформы для модуля процессора МЦП1502.

При изучении данного руководства следует также использовать документы:

- 467444.025РЭ_МЦП1502;

- 467444.026РЭ_МЦП905,

а также документы производителя микропроцессора:

- Rockchip RK3568J Datasheet V1.0-20220304;

- Rockchip RK3568 TRM Part1 V1.1-20210301;

- Rockchip RK3568 TRM Part2 V1.1-20210301.

1.1 Структура ПО поставки

МЦП905 поставляется с предустановленным на eMMC-flash ПО, включающим:

- загрузчик на основе u-boot. Расположен в начале встроенной eMMC-flash. Подробнее см. разделы 4, 5.

- ядро 4.19.232 и файл дерева устройств. Расположены на первом разделе встроенной eMMC-flash.

- файловую систему ОС Linux Ubuntu 20.04. Расположена на втором разделе встроенной eMMC-flash.

Пользователь также может загрузить модуль с microSD-flash карты. Подробнее – в подразделах 4.3, 5.4.

Подробнее информация представлена в разделе «Загрузка с других носителей».

Linux Ubuntu 20.04 построен на архитектуре arm64 и может обновляться из стандартных репозиториях Ubuntu, поддерживающих различные платформы (x86, x86_64, arm, arm64).

1.2 Версия документа. Описание изменений

Таблица 1 – История изменений

Версия	Дата	Изменения
1.0	22.01.2024	Создана первая версия
1.1	02.10.2024	Изменена ОС на модуле с Ubuntu 18.04 на Ubuntu 20.04. Удалена папка rockchip_official_scripts.

Исправлено имя образа microSD-flash, предоставляемого в архиве комплекта разработчика.

1.3 Справочная информация о доступных ресурсах референсной платформы

1.3.1 Доступные ресурсы референсной платформы

1. Процессор: RockChip RK3568J 4 ядра ARM Cortex-A55 1,8 ГГц.
2. ОЗУ: 4 Гб DDR4 с ECC.
3. NPU: Производительность до 1 Топс. Поддержка INT8/16, FP16/BFP16. Поддержка фреймворков: TensorFlow, TF-lite, Pytorch, Caffe, ONNX, MXNet, Keras, Darknet.
4. Подсистема хранения данных:
 - 32 Гбайт eMMC-flash стандарта версии 5.1;
 - разъем для подключения microSD-flash;
 - слот M.2 2242 для подключения накопителя SATA III.
5. Интерфейсы для подключения камер:
 - MIPI CSI2 2x2 lanes, 2,5 Gbps per lane; встроенный ISP.
6. Видеоконтроллер Mali-G52 800 МГц.
7. Видеокодеки:
 - H.265 до 4096x2304@60fps кодер/декодер;
 - H.264 до 4096x2304@60fps кодер/декодер;
 - JPEG кодер/декодер;
 - MPEG1/2/3/4 декодер.
8. Видеоинтерфейсы:
 - HDMI 2.0 до 4096x2304@60Hz. Выведен на стандартный разъем;
 - LVDS один канал 18/24 бита. Программно коммутируется с DSI0 до 1920x1080@60Hz;
 - eDP до 4096x2304@60Hz. Выведен на разъем для гибкого шлейфа.
9. Аудио:
 - микрофонный вход, выход на головные телефоны.
10. Два идентичных контроллера Ethernet. Поддержка скоростей 10/100/1000 Мбит/сек.

11. Два канала USB 2.0. Один с микропроцессора (USB0) выведен на разъем type A, второй (USB1) – через USB HUB.
12. Два канала USB 3.0. USB3.0 Host (USB2) выведен на разъем type A, USB 3.0 OTG (USB3) выведен на разъем type-C.
13. Два канала RS-422/RS-485. Гальванически изолированные. Напряжение изоляции 500 В.
14. Один канал RS-232. 2-проводной. Выведен на разъем IDC4. Уровни 3,3 В.
15. Два канала CAN. Гальванически изолированные. Напряжение изоляции 500 В. Поддержка CAN 2.0В.
16. GPIO: 3 линии. Уровни 1,8 В. Выведены на разъем mikroBUS.
17. Встроенный в микропроцессор сторожевой таймер. Он может генерировать прерывание или внутренний сброс процессора.
18. Два слота расширения miniPCIe с разъемами для SIM-карт.
19. Один слот расширения mikroBUS для установки модулей Click board или модулей собственной разработки.

1.3.2 Перечень интерфейсов, выведенных в МЦП1502

Таблица 2 – Интерфейсы

Обозначение интерфейса по спецификации SMARC	Интерфейс микропроцессора	Параметры инициализации в актуальной сборке ОС
LVDS0/eDP0/DSI0	MIPI_DSI/LVDS	Включен. Поддержка дисплея G070ACE-L01(7"LVDS_800x480)
LVDS1/eDP1/DSI1	eDP	Выключен. Разрешение до 4096x2304 60 Гц. Проверено на 3840x2160 60 Гц
HDMI/DP1	HDMI	Включен. Разрешение до 4096x2304 60 Гц. Проверено на 1920x1080 60 Гц
DP0	Отсутствует	
CSI0 x2	CSI_RX0, CSI_RX1, CSI_CLK0	Включен. Сенсор imx219 Разрешение видео 1920x1080 24 Гц
CSI1 x4	CSI_RX2, CSI_RX3, CSI_CLK1	Включен. Сенсор imx219 Разрешение видео 1920x1080 24 Гц
SDIO	SDMMC0	Включен. Объем microSD-flash до 128 Гб
SPI0 (2CS)	SPI3_M1	Включен. Выведен на разъём MikroBUS

Обозначение интерфейса по спецификации SMARC	Интерфейс микропроцессора	Параметры инициализации в актуальной сборке ОС
qSPI 1CS	FSPI	Выключен
I2S0	I2S1_M1	Включен. Звуковой интерфейс имеет выход для левого и правого каналов телефонов и одного канала микрофона
I2S1/HDA	Отсутствует	
I2C_PM	I2C0	Не подключен к разъёму SMARC
I2C_CAM0	I2C3_M0	Включен. Интерфейс выведен на разъём CSI0 и используется для управления камерой
I2C_CAM1	I2C2_M0	Включен. Интерфейс выведен на разъём CSI1 и используется для управления камерой
I2C_GP	I2C2_M0	Включен. Используется для управления камерой
I2C_LCD	I2C5_M0	Включен. Используется для поддержки click board
I2C HDMI_CTRL	HDMITX	Включен. Управление HDMI
SER0 4 wire	UART0	Включен. RS-422/485
SER1 2 wire	UART2_M0	Включен. Debug 1500000 8N1
SER2 4 wire	UART4_M0	Включен. RS-422/485
SER3 2 wire	UART5_M1	Включен. RS-232
CAN0	CAN0_M0	Включен Доступна скорость до 1Mbps
CAN1	CAN2_M1	Включен Доступна скорость до 1Mbps
USB0 (2.0) Host	USB2_HOST2	Включен
USB1(2.0) Host	USB2_HOST3_HUB0	Включен
USB2 (3.0) Host	USB3_HOST1	Включен
USB3 (3.0) OTG	USB3_OTG0	Включен
USB4 Host	USB2_HOST3_HUB1	Включен
USB5 Host	USB2_HOST3_HUB2	Включен
PCIe_A x1	PCIE30_0	Включен
PCIe_B x1	PCIE30_1	Включен
PCIe_C x1/SerDes0	Отсутствует	
PCIe_D x1/SerDes	Отсутствует	
SATA	SATA2	Включен
GBE0	GMAC0_M1	Включен. 10.173.216.250/24
GBE1	GMAC1_M1	Включен. 10.173.215.250/24
GPIO0 (CAM0_PWR#)	UART7_M1_TX/GPI O3_C4	Включен. MikroBUS 1,8В UART7 TX

Обозначение интерфейса по спецификации SMARC	Интерфейс микропроцессора	Параметры инициализации в актуальной сборке ОС
GPIO1 (CAM1_PWR#)	UART7_M1_RX/GPIO3_C5	Включен. MikroBUS 1,8В UART7 RX
GPIO2 (CAM0_RST#)	GPIO0_D5	Включен. MikroBUS interrupt pin. Доступен как GPIO
GPIO3 (CAM1_RST#)	GPIO0_D6	Включение питания LCD 3.3V
GPIO4 (HDA_RST#)	GPIO3_D3	Включен. MikroBUS reset pin. Доступен как GPIO
GPIO5 (PWM_OUT)	GPIO0_B7	Включение питания слота miniPCIe 1 (включено)
GPIO6 (TACHIN)	GPIO3_D4	Включение питания слота miniPCIe 2 (включено)
GPIO7	GPIO2_C6	Выключен
GPIO8	GPIO2_C5	Выключен
GPIO9	GPIO4_B2	Выключен
GPIO10	GPIO0_D4	Включен. CSI0 pwrn Сигнал активности
GPIO11	GPIO3_A4	Выключен
BATLOW#	GPIO3_A6	Выключен
CARRIER_PWR_ON	EXT EN от RK809	Включен
CARRIER_STBY#	GPIO3_A3	Включен. USER-LED запрограммирован непрерывно менять яркость, имитируя сердцебиение (trigger="heartbeat")
CHARGER_PRSENT#	GPIO2_D1	Не подключен
CHARGING#	GPIO2_D0	CSI0 pwrn Сигнал активности
VIN_PWR_BAD#	GPIO1_B2	Не подключен
SLEEP#	GPIO1_B1	Не подключен
SMB_ALERT#	GPIO4_C0	Не подключен
TEST#	GPIO3_B5	Не подключен
BOOT_SEL0#	SARADC_VIN3	Включен. MikroBUS АЦП вход 0 – 1.8V
BOOT_SEL1#	SARADC_VIN4	Не подключен
BOOT_SEL2#	SARADC_VIN5	Не подключен
FORCE_RECOV#	SARADC_VIN0	RECOVERY кнопка

Таблица 3 – Функциональные узлы

Функциональный узел	Параметры инициализации в актуальной сборке ОС
Neural Process Unit	Выключен (в разработке)
Trust Execution Environment system	Включен
Video Decoder	Включен
Video Encoder	Включен
JPEG decoder	Включен
JPEG encoder	Включен
IEP module	Включен
3D Graphics Engine	Включен
2D Graphics Engine	Включен
MIPI CSI RX Interface	Включен
VICAP	Включен
ISP	Включен
Video Output Processor	Включен
Audio Interface	Включен
Watchdog	Включен

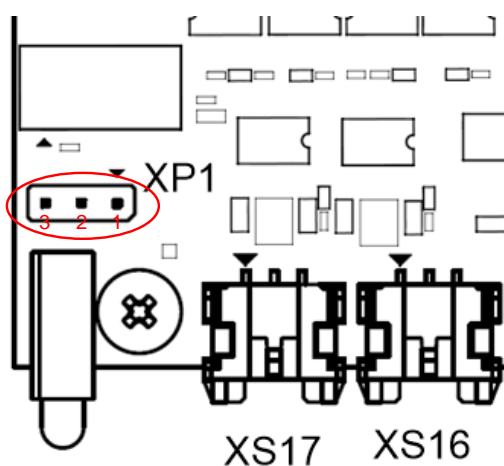
2 Использование терминального режима отладки

От момента подачи питания до появления информации на дисплее модуль МЦП905 запускает u-boot и ядро Linux. Для просмотра отладочных сообщений используется DEBUG-PORT. По умолчанию он включен. На Рис. 1 (fiq-debugger) отображены DTS настройки вывода отладочной информации в последовательный порт UART2. В нем настроена скорость 1,5 мегабит/сек. Здесь вы можете её поменять, но она поменяется только в отладчике ядра. Также вы можете её поменять в отладчике u-boot.

```
fiq_debugger: fiq-debugger {
    compatible = "rockchip,fiq-debugger";
    rockchip,serial-id = <2>;
    rockchip,wake-irq = <0>;
    /* If enable uart uses irq instead of fiq */
    rockchip,irq-mode-enable = <1>;
    rockchip,baudrate = <1500000>; /* Only 115200 and 1500000 */
    interrupts = <GIC_SPI 252 IRQ_TYPE_LEVEL_LOW>;
    pinctrl-names = "default";
    pinctrl-0 = <&uart2m0_xfer>;
    status = "okay";
};
```

Рисунок 1— fiq-debugger

DEBUG-PORT — это последовательный порт (UART). Используются логические уровни 1,8 В. Для подключения к ПК используется USB to Serial переходник (adapter).

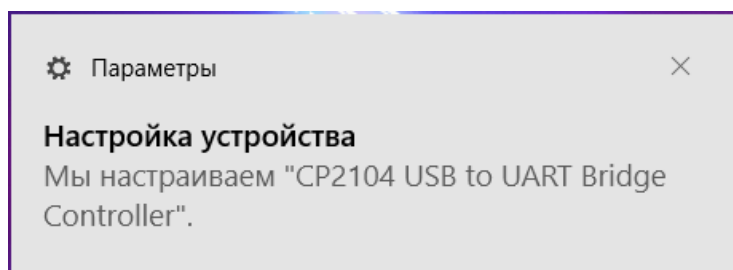


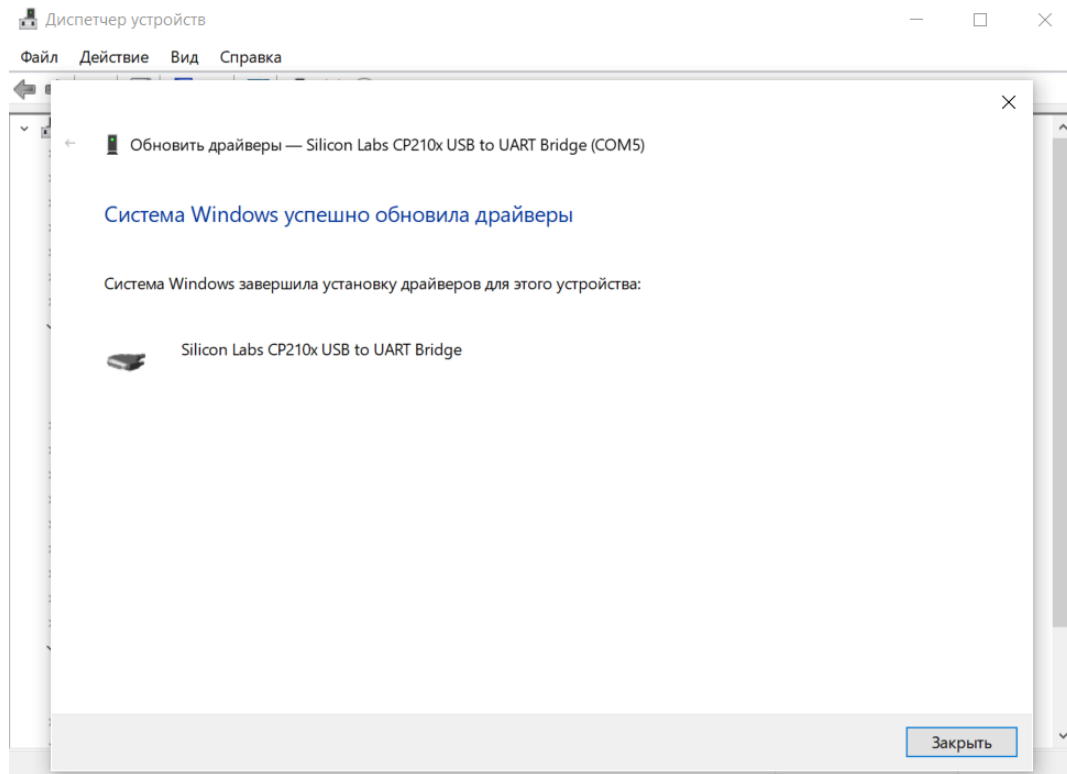
Для работы адаптера необходимо подключить три провода.

Обозначение на адаптере	Номер контакта в отладочном разъёме модуля
GND	МЦП905 3 (GND)
TX	2 (RX)
RX	1 (TX)

2.2 Параметры подключения. Установка драйвера для Windows, запуск терминала

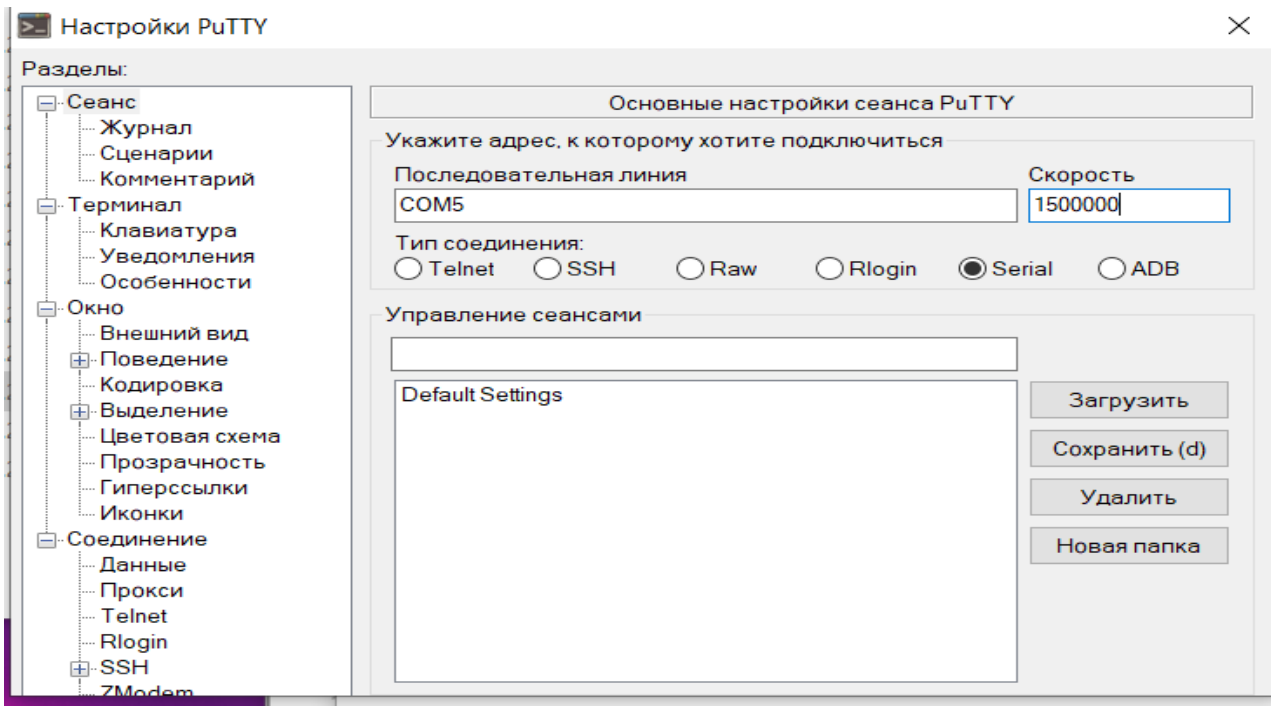
Windows требуется установить [драйвер](#) микросхемы CP2104.





Взаимодействие пользователя с микросхемой CP2104 происходит через терминал. Рекомендуем PuTTY.

В нём необходимо выбрать Serial. Выбрать COM5 (номером 5 он обозначен в диспетчере устройств), назначить скорость 1500000 и нажать <Enter>.



2.3 Использование терминального подключения под Ubuntu

В OS Ubuntu необходимо установить пакет minicom.

```
apt install minicom
```

Запускать с параметрами интерфейса и скорости.

```
minicom -D /dev/ttyUSB0 -b 1500000
```

3 Компиляция системного ПО. Архив комплекта разработчика

Архив комплекта разработчика можно скачать по [ссылке](#). Файл: mcp905_mcp1502_ddmму.zip, где ddmму – дата сборки архива. Архив запаролен. Для получения пароля обратитесь в службу технической поддержки support@atronik.ru.

В архиве размещены папки:

- build_os_img — набор файлов и скриптов для компиляции ядра ОС, загрузчика u-boot, для создания образа ОС референсной платформы.
- info_mcp905 — документация на устройство.
- os_img — образы установочных microSD-flash.
- ssh_key — ключ доступа по сети.

В папке build_os_img находятся:

- all.sh — ключевой скрипт сборки.
- bash_features.txt — вспомогательные bash-функции.
- boards — список известных плат и их файлы настроек.
- kernels — ядра ОС поддерживаемых плат.
- out — папка для сохранения результатов компиляции.
- special_devices_drivers — драйверы устройств, поддержка которых отсутствует в ядре.
- u-boot — исходники вторичного загрузчика.
- use_cases — скрипты автоматической компиляции и установки на целевое устройство.

Настройка целевого устройства прописана в файле `/build_os_img/use_cases/_device`. В переменной окружения `DEVICE` должно быть записано `mcp905`.

```
DEVICE='mcp905'  
export DEVICE
```

Перед подключением по `ssh` и запуском скриптов, имеющих в имени слово `apply`, необходимо настроить и проверить связь по сети. Для этого:

1. Подключите ваш компьютер к модулю МЦП905 сетевым кабелем (допускается использовать сетевые хабы и роутеры).
2. Проверьте, что модуль МЦП905 запитан и светится светодиод питания.
3. Настройте на своём компьютере `ip`-адрес, принадлежащий подсети модуля.

Настройка `ip`-адресов производится командами:

- `ip a` — показать существующие интерфейсы и `ip`-адреса.
- `ip a add 10.173.215.254/24 dev eth0` — добавить `ip`-адрес `10.173.215.254/24` в интерфейс `eth0`.
- `ip a add 10.173.216.254/24 dev eth0` — добавить `ip`-адрес `10.173.216.254/24` в интерфейс `eth0`.

По умолчанию в модуле настроены адреса `10.173.215.250/24` и `10.173.216.250/24` для первого и второго разъемов `RJ45` соответственно.

4. Запустите команду `ping 10.173.215.250` или `ping 10.173.216.250` в зависимости от того, к какому разъёму вы подключены.
5. Убедитесь, что программа `ping` может связаться с модулем МЦП905.
6. В файле конфигурации `ssh` соединения `~/.ssh/config` должен быть описан узел с тем же именем (`mcp905`). Файл ключа `/ssh_key/key` идёт в комплекте поставки. Если у Вас нет файла `~/.ssh/config`, то создайте его. В поле `HostName` должен быть указан `IP`-адрес, с которым смогла связаться программа `ping`. В поле `IdentityFile` указывается путь к файлу ключа.

```
Host mcp905  
  HostName 10.173.215.250  
  User root  
  Port 22  
  IdentityFile /ssh_key/key
```

Если вы не желаете пользоваться файлом ключа доступа по `ssh`, то используйте имя пользователя `user` и пароль `user` для доступа по сети. Команда доступа по имени пользователя и паролю: `ssh user@10.173.215.254`

Автоматическая установка рассчитана на доступ по ключу.

При настроенном ssh соединении через ключ, команда доступа по сети: `ssh mcp905`

Компиляция производится в OS Linux при помощи компилятора gcc8. Корректность компиляции иных версий gcc не гарантирована.

Для конфигурирования и компиляция системного ПО модуля МЦП905 используйте скрипты из папки use_cases:

- dts.sh — компиляция дерева устройств.
- kernel_menuconfig.sh — правка конфигурации ядра.
- kernel.sh — компиляция ядра.
- u-boot_menuconfig.sh — правка конфигурации загрузчика (в разработке).
- u-boot.sh — компиляция загрузчика (в разработке).

Для автоматической компиляции и автоматической установки через ssh-соединение (при успешной компиляции) необходимо использовать скрипты:

- dts_vs_apply.sh — компиляция дерева устройств, заливка обновлённого файла дерева устройств по ssh на модуль mcp905, перезагрузка модуля, просмотр системных сообщений.
- kernel_vs_apply.sh — компиляция ядра и его модулей, заливка по ssh на модуль mcp905, перезагрузка модуля, просмотр системных сообщений.

3.1 Ядро Linux

За основу взято ядро Linux 4.19.232 из ветки linux-kernel-4-19-232 с официального сайта производителя <https://github.com/rockchip-linux/kernel>

Ядро Linux поставляется в папке /build_os_img/kernels/mcp905.

Для изменения параметров конфигурации ядра необходимо его пересобрать. Для этого:

1. В папке use_cases запустите скрипт очистки kernel_clean_4_new_extraversion.sh с опцией, описывающей расширение версии ядра (например, `test_new`).

```
kernel_clean_4_new_extraversion.sh test_new
```

2. В папке use_cases запустите скрипт конфигурирования kernel_menuconfig.sh. В открывшемся окне произведите необходимые настройки и нажмите `Exit`
3. Сохраните бэкап прежнего конфига. В папке /boards/mcp905/ переименуйте файл kernel.config (например, в kernel.config_old).
4. Сделайте новый конфигурационный файл активным. Из папки /build_os_img/kernels/mcp905 переместите файл .config в папку /build_os_img/boards/mcp905/ под именем kernel.config
5. Запустите компиляцию ядра скриптом kernel.sh.

- Из папки out необходимо скопировать новое ядро, модули ядра и dtb-файл в соответствующие папки на файловой системе модуля МЦП905.

Таблица 4 – Компилируемые объекты ядра

Название	Расположение в свежесобранного на ПК	Расположение в файловой системе модуля МЦП905
Ядро	/build_os_img/out/mcp905/kernel	/boot
Модули ядра	/build_os_img/out/mcp905/kernel_modules	/lib/modules
Дерево устройств	/build_os_img/out/mcp905/kernel/k3568-mcp905.dtb	/boot

3.2 U-boot

Подраздел в разработке.

Для загрузки ОС Linux на архитектуре arm64 использован загрузчик с открытым исходным кодом u-boot 2017.09.

Сборка и установка загрузчика u-boot на ПК

Загрузчик поставляется в папке в виде исходных кодов.

Для сборки u-boot необходимо в папке use_cases запустить скрипт make_u-boot.sh.

Порядок записи загрузчика на носитель:

- Скопировать файл flash.bin на microSD-flash или на eMMC-flash модуля МЦП905.
- Запись производится из-под работающей системы командой:

```
dd if=./u-boot.itb of=/dev/mmcblk0 seek=16384
```

3.3 Дерево устройств (dts)

Дерево устройств описано в файле /build_os_img/dts/rk3568-mcp905.dts. Для его компиляции необходимо в папке use_cases запустить скрипт dts.sh. После завершения скопируйте файл /build_os_img/dts/k3568-mcp905.dtb в папку /boot модуля МЦП905.

4 Загрузка системного программного обеспечения

Загрузка модуля МЦП905 происходит в три этапа: первичный загрузчик, u-boot, ядро ОС. Первичный загрузчик закодирован в кристалл RK3568. Он ищет загрузчик u-boot на носителях информации: SPI-flash, eMMC-flash и microSD-flash. SPI-flash на референсной платформе МЦП905 не подключена. Поэтому в первую очередь происходит попытка загрузки с microSD-flash.

U-boot даёт возможность предложить пользователю меню выбора загрузки и загрузиться с любого доступного накопителя.

В DTS описан доступ из ОС Linux к eMMC-flash, microSD-flash и SPI-flash (SPI flash не установлена).

```
sdhci: sdhci@fe310000 {
    compatible = "rockchip,dwcshc-sdhci", "snps,dwcshc-sdhci";
    reg = <0x0 0xfe310000 0x0 0x10000>;
    interrupts = <GIC_SPI 19 IRQ_TYPE_LEVEL_HIGH>;
    assigned-clocks = <&cru BCLK_EMMC>, <&cru TCLK_EMMC>,
        <&cru CCLK_EMMC>;
    assigned-clock-rates = <200000000>, <24000000>, <200000000>;
    clocks = <&cru CCLK_EMMC>, <&cru HCLK_EMMC>,
        <&cru ACLK_EMMC>, <&cru BCLK_EMMC>,
        <&cru TCLK_EMMC>;
    clock-names = "core", "bus", "axi", "block", "timer";
    status = "disabled";
};
```

```
&sdhci {
    bus-width = <8>;
    supports-emmc;
    non-removable;
    max-frequency = <200000000>;
    status = "okay";
};
```

Рисунок 3 – sdhci (SD Host Controller). Интерфейс microSD-карты

```
sdmmc0: dwmmc@fe2b0000 {
    compatible = "rockchip,rk3568-dw-mshc",
        "rockchip,rk3288-dw-mshc";
    reg = <0x0 0xfe2b0000 0x0 0x4000>;
    interrupts = <GIC_SPI 98 IRQ_TYPE_LEVEL_HIGH>;
    max-frequency = <150000000>;
    clocks = <&cru HCLK_SDMMC0>, <&cru CLK_SDMMC0>,
        <&cru SCLK_SDMMC0_DRV>, <&cru SCLK_SDMMC0_SAMPLE>;
    clock-names = "biu", "ciu", "ciu-drive", "ciu-sample";
    fifo-depth = <0x100>;
    resets = <&cru SRST_SDMMC0>;
    reset-names = "reset";
    status = "disabled";
};
```

```
&sdmmc0 {
    max-frequency = <150000000>;
    supports-sd;
    bus-width = <4>;
    cap-mmc-highspeed;
    cap-sd-highspeed;
    disable-wp;
    num-slots = <1>;
    sd-uhs-sdr104;
    vmmc-supply = <&vcc3v3_sys>;
    vqmmc-supply = <&vccio_sd>;
    pinctrl-names = "default";
    pinctrl-0 = <&sdmmc0_bus4 &sdmmc0_clk
        &sdmmc0_cmd &sdmmc0_det>;
    status = "okay";
};
```

Рисунок 4 – sdmmc0. Встроенная EMMC flash

```
sfc: sfc@fe300000 {
    compatible = "rockchip,sfc";
    reg = <0x0 0xfe300000 0x0 0x4000>;
    interrupts = <GIC_SPI 101 IRQ_TYPE_LEVEL_HIGH>;
    clocks = <&cru SCLK_SFC>, <&cru HCLK_SFC>;
    clock-names = "clk_sfc", "hclk_sfc";
    assigned-clocks = <&cru SCLK_SFC>;
    assigned-clock-rates = <100000000>;
    status = "disabled";
};
```

```
&sfc {
    status = "okay";
};
```

Рисунок 5 – sfc (Serial Flash Controller) – доступ к SPI flash

4.1 Загрузка со встроенной eMMC-flash

Для загрузки со встроенной eMMC-flash необходимо извлечь microSD-flash и подать питание. Загрузится u-boot со встроенной eMMC-flash, который воспользуется файлом конфигурации запуска, лежащем на первом разделе, монтируемом в папку /boot.

Файл конфигурации запуска /boot/extlinux/extlinux.conf представляет из себя меню загрузки ОС. В меню загрузки перечислены варианты загрузки ядер ОС с параметрами. По умолчанию выбирается дефолтный пункт меню и загружается соответствующее ядро и дерево устройств.

```
menu title select kernel
default kernel-4.19
```

Заголовок варианта загрузки, выбранный по умолчанию

```
label kernel-4.19
    kernel /Image
    fdt /rk3568-mcp905.dtb
    append root=/dev/mmcblk0p2 earlyprintk console=ttyFIQ0,1500000n8
    console=tty1 consoleblank=0 loglevel=7 panic=10 rootwait rw init=/sbin/init
    rootfstype=ext4 cgroup_enable=cpuset cgroup_memory=1 cgroup_enable=memory
    swapaccount=1 swiotlb=1 coherent_pool=1m
```

Вариант загрузки под заголовком kernel-4.19

4.2 Загрузка с microSD-flash

Загрузка с microSD-flash используется для сброса eMMC-flash к заводским настройкам. В соответствии с подразделом 5.4. Возможно также загружать ОС и полноценно работать с microSD-flash. Однако скорость работы eMMC-flash равна 40 Мбайт/сек, а скорость microSD-flash равна ~12 Мбайт/сек.

Загрузка с microSD-flash полностью аналогична загрузке со встроенной eMMC-flash, см. подраздел 5.4.

4.3 Загрузка с SSD

Подраздел в разработке.

4.4 Загрузка с USB

Подраздел в разработке.

5 Установка (обновление) системного программного обеспечения

Операции по установке/обновлению системного программного обеспечения модуля МЦП905 выполняются за счёт замены содержимого встроенной eMMC-flash.

5.1 Структура встроенной eMMC-flash

На встроенной eMMC-flash есть три области данных:

- загрузчик u-boot;
- раздел boot, на котором размещается ядро ОС, дерево устройств и конфигурация загрузки, в которой описаны варианты загрузки с ядрами ОС и их параметрами
- корневой раздел ОС.

5.2 MaskROM upgrade mode из Linux

Подраздел в разработке.

Обновление системного ПО через USB-кабель

- Кабель USB type-C;
- Установка RKDevtool для Windows;
- Установка RK USB drive для Windows;
- Подключение модуля;
- Установка upgrade_tool для Linux.

5.3 Loader upgrade mode

Подраздел в разработке.

5.4 MicroSD-flash upgrade mode

В комплекте поставки есть iso образ /os_img/mcp905_mcp1502_install_vX.iso. Это образ microSD-flash, который сбросит содержимое встроенной в МЦП905 eMMC-flash в заводское состояние.

Для этого необходимо:

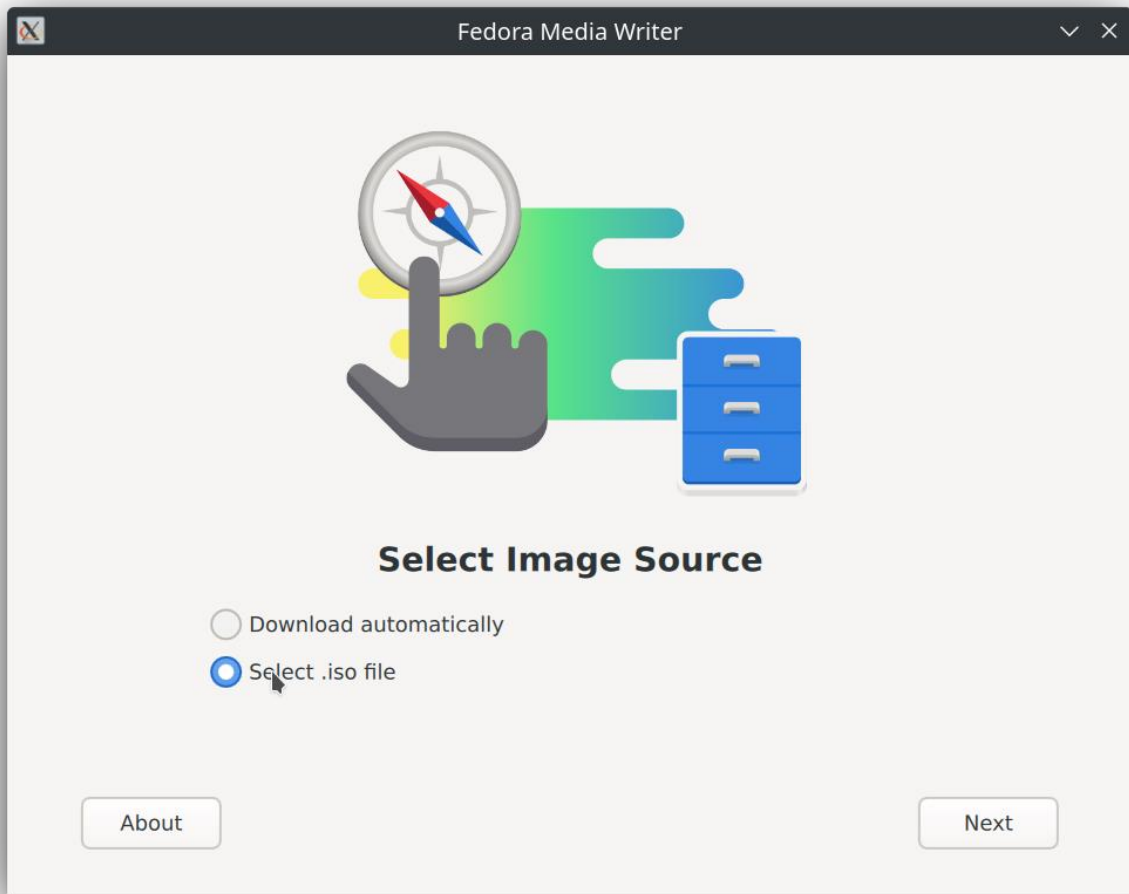
1. При помощи утилиты создания загрузочной flash установить /os_img/mcp905_mcp1502_install_vX.iso на microSD-flash размером не менее 8 Гб (см. подраздел 5.5.).
2. Загрузиться с microSD-flash (см. подраздел 5.6).

5.5 Создание загрузочной microSD-flash

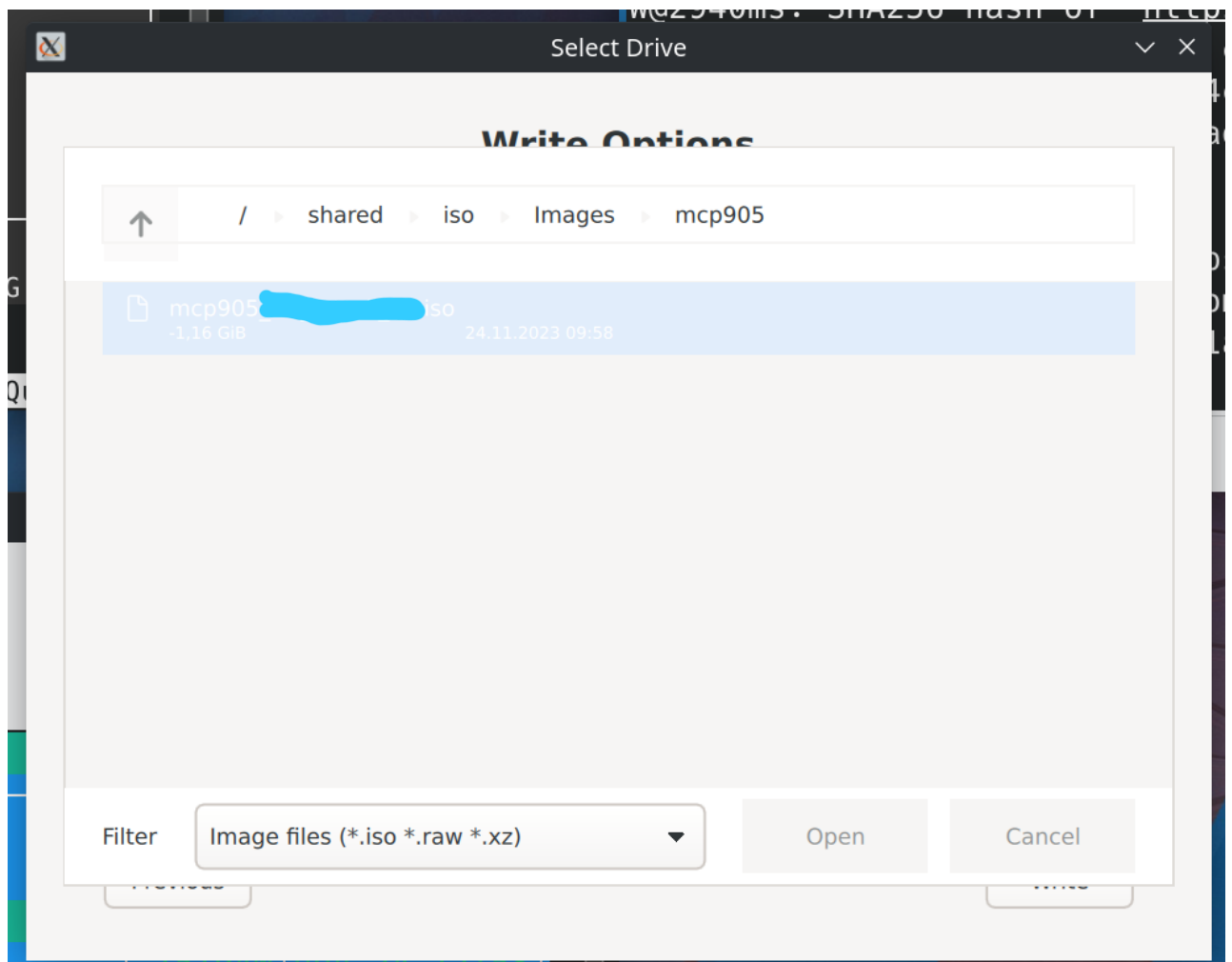
В комплекте поставки есть заархивированный iso образ /os_img/mcp905_mcp1502_install_vX.iso.gz. Это архив образа microSD-flash для обновления содержимого встроенной в МЦП905 eMMC-flash в заводское состояние (конфигурация референсной платформы). Разархивируйте его и получите iso образ microSD-flash.

Для обновления eMMC-flash необходимо:

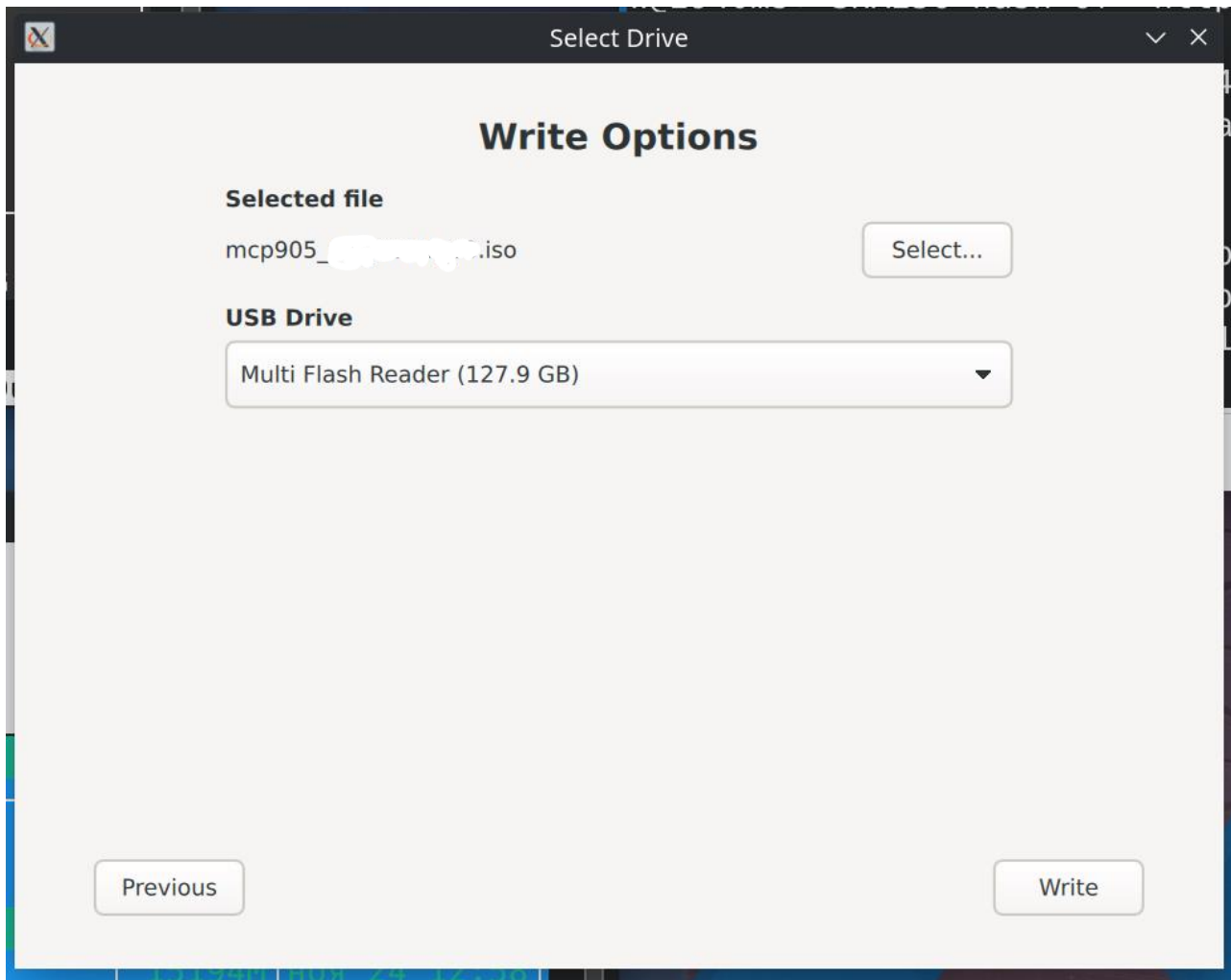
1. Запустите [программу](#) создания загрузочных USB-flash.



2. Выберите пункт Select .iso file.



3. Выберите файл /os_img/mcp905_mcp1502_install_vX.iso и свою microSD-flash.



6. Нажмите Write и ждите завершения.

5.6 Использование загрузочной microSD-flash

Созданная в подразделе 5.5 microSD-flash является загрузочной. Её можно вставить в разъём microSD и подать питание. Произойдёт загрузка с microSD-flash.

По умолчанию начнётся запись образа (лежащего на microSD-flash) на eMMC-flash. Для отмены этой автоматической операции и работы в режиме LiveMicroSD card нажмите <CTRL> + <C>. В режиме LiveMicroSD card вы можете самостоятельно создать/развернуть iso-образ внутренней eMMC-flash.

6 Установка (обновление) пользовательского программного обеспечения

Операции по установке/обновлению пользовательского программного обеспечения модуля МЦП905 выполняются за счёт замены содержимого папки, содержащей пользовательское ПО на eMMC-flash. И перезагрузки модуля МЦП905.

6.1 Обновление с USB-flash/microSD-flash

После штатного запуска подключите внешнюю клавиатуру и USB-flash (или microSD-flash) к МЦП905. Нажмите <Ctrl> + <Alt> + <F2>. Откроется консоль. В ней необходимо примонтировать внешний накопитель и заменить папку с пользовательским ПО. Для этого:

1. Запустите команду `lsblk`
2. В списке найдите своё устройство (не /dev/mmcblk0*).
3. Примонтируйте свое устройство в папку /mnt.
4. Удалите папку пользовательского ПО.
5. Скопируйте из папки /mnt папку пользовательского ПО.
6. Перезагрузите модуль командой `reboot`.

6.2 Обновление по сети через ssh

После штатного запуска подключите внешнюю клавиатуру к модулю МЦП905. Нажмите <Ctrl> + <Alt> + <F2>. Откроется консоль. В ней:

1. Настройте IP-address для доступа по ssh, как описано в разделе 3.
2. Удалите папку пользовательского ПО.

На подключенном компьютере:

1. Настройте IP-address для доступа по ssh, как описано в разделе 3.
2. Запустите команду копирования по сети.
`scp -R /path/to/soft mcp905:/soft`
3. Дождитесь завершения копирования.
4. Выключите и включите питание модуля МЦП905.

7 Функциональные блоки, интерфейсы и подключаемые устройства

На модуле МЦП905 установлены интерфейсы:

- видео: HDMI, LVDS, 2 x CSI camera;
- память: слот для установки microSD-flash, слот M.2 key B для установки SATA SSD;
- шины: USB 3.0, USB 2.0, USB3.0-OTG (Type-c);
- модульные: I2C, SPI;
- последовательные порты: 2 x RS-485, RS-232, debug_uart;

- расширение: 2 x Mini PCI-E с выводами на SIM-Card, MikroBUS;
- сетевые: 2 x CAN, 2 x Ethernet.

7.1 CSI Camera

Микропроцессор RK3568 имеет один порт (MIPI CSI2 DPHY), который поддерживает два режима работы. Full mode, использующий 4 линии данных и Split mode, позволяющий организовать два интерфейса по две линии данных. Физически в МЦП1502 на линии разъема SMARC CSI0 выводятся линии 0/1 и на линии CSI1 линии 2/3. Соответственно, на МЦП905 установлены два разъема для подключения двух камер, использующих по две линии данных.

Также в разъёмах CSI есть интерфейс конфигурирования I2C и дискретные сигналы управления камерой (Рис. 6).

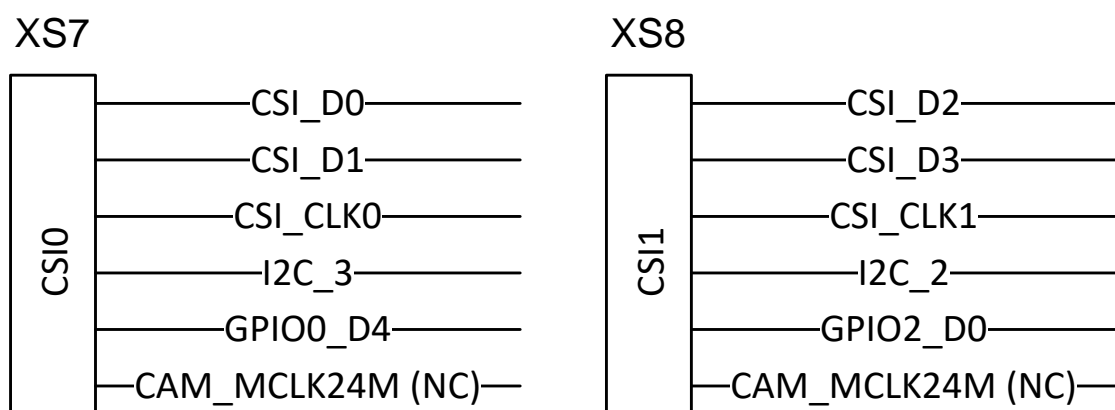


Рисунок 6 – CSI0, CSI1

Таблица 5 – Режимы подключения камеры

<i>Option</i>	<i>Sensor Lane</i>	<i>Interface</i>
<i>full mode</i>	<i>sensor1 x4Lane</i>	<i>MIPI_CSI_RX_D0-3</i> <i>MIPI_CSI_RX_CLK0</i>
<i>split mode</i>	<i>sensor1 x2Lane</i>	<i>MIPI_CSI_RX_D0-1</i> <i>MIPI_CSI_RX_CLK0</i>
	<i>+</i> <i>sensor2 x2Lane</i>	<i>MIPI_CSI_RX_D2-3</i> <i>MIPI_CSI_RX_CLK1</i>

В обработке изображения принимают участие:

- Sensor — матрица камеры, выдающая данные изображения в линии разъёма CSI.
- CSI D-PHY — физический уровень интерфейса CSI.
- ISP (Image Signal Processor) — встроенный в кристалл RK3568 процессор обработки изображений.

7.1.1 Full-mode (одна камера на 4 канала данных)

Обработка изображения происходит: sensor→csi2_dphy0→isp. На выходе ISP получается /dev/video0 интерфейс захвата видео.

7.1.2 Split-mode (две камеры по 2 канала данных на каждую)

Обработка изображения происходит по цепочкам:

sensor1->csi_dphy1->isp_vir0

sensor2→csi_dphy2→mipi_csi2→vicap→isp_vir1

На выходе isp_vir0 получается /dev/video0 и /dev/video1 интерфейсы захвата видео.

На выходе isp_vir1 получается /dev/video9 и /dev/video10 интерфейсы захвата видео.

Конфигурация isp описана на Рис. 6. Конфигурация камер описана на Рис. 8. Физический уровень интерфейса csi2_dphy – на Рис. 9.

```

&rkisp_vir0 {
    status = "okay";

    port {
        #address-cells = <1>;
        #size-cells = <0>;

        isp0_in: endpoint@0 {
            reg = <0>;
            remote-endpoint = <&csi2dphy1_out>;
        };
    };
};

&rkisp_vir1 {
    status = "okay";
    #address-cells = <1>;
    #size-cells = <0>;

    port {
        reg = <0>;
        #address-cells = <1>;
        #size-cells = <0>;

        isp1_in: endpoint@0 {
            reg = <0>;
            remote-endpoint = <&csi2dphy2_out>;
        };
    };
};

```

Рисунок 7– isp

```

&csi2_dphy1 {
    status = "okay";

    /*
     * dphy1 only used for split mode,
     * can be used concurrently with dphy2
     * full mode and split mode are mutually exclusive
     */
    ports {
        #address-cells = <1>;
        #size-cells = <0>;
        port@0 {
            reg = <0>;
            #address-cells = <1>;
            #size-cells = <0>;

            csi2dphy1_ucam0: endpoint@1 {
                reg = <1>;
                remote-endpoint = <&imx219_1_out>;
                data-lanes = <1 2>;
            };
        };

        port@1 {
            reg = <1>;
            #address-cells = <1>;
            #size-cells = <0>;

            csi2dphy1_out: endpoint@1 {
                reg = <1>;
                remote-endpoint = <&isp0_in>;
            };
        };
    };
};

```

```

&csi2_dphy2 {
    status = "okay";

    /*
     * dphy2 only used for split mode,
     * can be used concurrently with dphy1
     * full mode and split mode are mutually exclusive
     */
    ports {
        #address-cells = <1>;
        #size-cells = <0>;
        port@0 {
            reg = <0>;
            #address-cells = <1>;
            #size-cells = <0>;

            csi2dphy2_ucam0: endpoint@1 {
                reg = <1>;
                remote-endpoint = <&imx219_2_out>;
                data-lanes = <1 2>;
            };
        };

        port@1 {
            reg = <1>;
            #address-cells = <1>;
            #size-cells = <0>;

            csi2dphy2_out: endpoint@1 {
                reg = <1>;
                remote-endpoint = <&isp1_in>;
            };
        };
    };
};

```

Рисунок 8 – csi2_dphy1, csi2_dphy2

7.2 Сети Ethernet

1. gmac0 и gmac1 — интерфейсы ethernet (eth0, eth1).

```
&gmac0 {
    phy-mode = "rgmii";
    clock_in_out = "output";

    snps,reset-gpio = <&gpio3 RK_PB7 GPIO_ACTIVE_LOW>;
    snps,reset-active-low;
    /* Reset time is 20ms, 100ms for rtl8211f */
    snps,reset-delays-us = <0 20000 100000>;

    assigned-clocks = <&cru SCLK_GMAC0_RX_TX>, <&cru SCLK_GMAC0>;
    assigned-clock-parents = <&cru SCLK_GMAC0_RGMII_SPEED>;
    assigned-clock-rates = <0>, <125000000>;

    pinctrl-names = "default";
    pinctrl-0 = <&gmac0_miim
        &gmac0_tx_bus2
        &gmac0_rx_bus2
        &gmac0_rgmii_clk
        &gmac0_rgmii_bus>;

    tx_delay = <0x38>;
    rx_delay = <0x2d>;

    phy-handle = <&rgmii_phy0>;
    status = "okay";
};

&gmac1 {
    phy-mode = "rgmii";
    clock_in_out = "output";

    snps,reset-gpio = <&gpio3 RK_PB0 GPIO_ACTIVE_LOW>;
    snps,reset-active-low;
    /* Reset time is 20ms, 100ms for rtl8211f */
    snps,reset-delays-us = <0 20000 100000>;

    assigned-clocks = <&cru SCLK_GMAC1_RX_TX>, <&cru SCLK_GMAC1>;
    assigned-clock-parents = <&cru SCLK_GMAC1_RGMII_SPEED>;
    assigned-clock-rates = <0>, <125000000>;

    pinctrl-names = "default";
    pinctrl-0 = <&gmac1_miim
        &gmac1_tx_bus2
        &gmac1_rx_bus2
        &gmac1_rgmii_clk
        &gmac1_rgmii_bus>;

    tx_delay = <0x46>;
    rx_delay = <0x2d>;

    phy-handle = <&rgmii_phy1>;
    status = "okay";
};
```

Рисунок 9 – gmac0, gmac1

Микропроцессор RK3568 имеет два контроллера Ethernet. На модуле МЦП1502 установлены две микросхемы физического уровня Ethernet RTL8211. На модуле МЦП905 установлены разъемы со встроенными трансформаторами.

Конфигурация сетевых адресов содержится в файлах eth0.nmconnection и eth1.nmconnection, находящихся в папке /etc/NetworkManager/system-connections

У интерфейса eth0 настроен статический IP-адрес 10.173.216.250/24.

У интерфейса eth1 настроен статический IP-адрес 10.173.215.250/24.

7.3 I2C-шина

I2C-шина (Inter-Integrated Circuit) — последовательная асимметричная шина для связи между интегральными схемами внутри электронных приборов. Использует две двунаправленные линии связи (SDA и SCL), применяется для соединения низкоскоростных периферийных компонентов с процессором. На каждой шине есть нумерация адресов от 1 до 127. Из микропроцессора RK3568 выведены пять независимо работающих шин. Адрес каждого устройства, подключенного к шине I2C описан в документации на него. В случае подключения двух устройств с одним и тем же адресом, эти устройства необходимо разносить на разные шины.

7.3.1 Доступные I2C-контроллеры

Модуль МЦП905 содержит программно-доступные шины с номерами:

- #0, #2, #3, #5 — rk3x-i2c I2C adapter (встроенные в кристалл RK3568 адаптеры I2C)
- #6 — DesignWare HDMI I2C adapter (адаптер интерфейса HDMI).

Список доступных шин можно получить командой `i2cdetect -l`

7.3.2 Поиск устройств на I2C-шинах

Зная номера доступных шин, можно их просканировать командой `i2cdetect -y 2`

```
root@mcp905:~# i2cdetect -y 2
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  UU  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
```

На Рис. 54 мы видим UU на адресе 10. Это CSI камера.

7.3.3 Конфигурирование I2C-драйверов устройств

Шина I2C не имеет возможности автоматического определения устройств. Все подключенные устройства должны быть описаны в дереве устройств (dts). Например, на Рис. 55 CSI камера на **2** шине на адресе **10**. Также в конфигурации описывается имя драйвера в поле `compatible`.

```
&i2c2 {  
    status = "okay";  
    #address-cells = <1>;  
    #size-cells = <0>;  
  
    camera_imx219_2: camera-imx219-2@10 {  
        status = "okay";  
        compatible = "sony,imx219";  
        reg = <0x10>;  
        clock = <ext_cam_clk_imx219_2>;
```

Рисунок 11 – I2C device

7.3.4

Чтение/запись информации через I2C интерфейс

Для чтения/записи данных в устройствах рекомендуется использовать соответствующие драйвера, указанные в поле `compatible`. Однако OS Linux позволяет пользователю читать и записывать данные без участия драйвера устройства. Для этого используются команды `i2cset`, `i2cget`, `i2cdump` из пакета `apt-get install i2c-tools`.

7.4 Видеоподсистема GPU

В видеоподсистему входят подсистемы, отвечающие за:

- вывод на монитор (`display_subsystem`, Рис. 14).
- 3D-ускорение, 2D-ускорение.
- кодирование/декодирование видео, декодирование формата JPEG.
- обработку исходных изображений по корректировке яркости, контрастности и др.

```

display_subsystem: display-subsystem {
    compatible = "rockchip,display-subsystem";
    memory-region = <&drm_logo>, <&drm_cubic_lut>;
    memory-region-names = "drm-logo", "drm-cubic-lut";
    ports = <&vop_out>;
    devfreq = <&dmc>;

    route {
        route_dsi0: route-dsi0 {
            status = "disabled";
            logo,uboot = "logo.bmp";
            logo,kernel = "logo_kernel.bmp";
            logo,mode = "center";
            charge_logo,mode = "center";
            connect = <&vp0_out_dsi0>;
        };
        route_dsi1: route-dsi1 {
            status = "disabled";
            logo,uboot = "logo.bmp";
            logo,kernel = "logo_kernel.bmp";
            logo,mode = "center";
            charge_logo,mode = "center";
            connect = <&vp0_out_dsi1>;
        };
        route_edp: route-edp {
            status = "disabled";
            logo,uboot = "logo.bmp";
            logo,kernel = "logo_kernel.bmp";
            logo,mode = "center";
            charge_logo,mode = "center";
            connect = <&vp0_out_edp>;
        };
        route_hdmi: route-hdmi {
            status = "disabled";
            logo,uboot = "logo.bmp";
            logo,kernel = "logo_kernel.bmp";
            logo,mode = "center";
            charge_logo,mode = "center";
            connect = <&vp1_out_hdmi>;
        };
        route_lvds: route-lvds {
            status = "disabled";
            logo,uboot = "logo.bmp";
            logo,kernel = "logo_kernel.bmp";
            logo,mode = "center";
            charge_logo,mode = "center";
            connect = <&vp1_out_lvds>;
        };
        route_rgb: route-rgb {
            status = "disabled";
            logo,uboot = "logo.bmp";
            logo,kernel = "logo_kernel.bmp";
            logo,mode = "center";
            charge_logo,mode = "center";
            connect = <&vp2_out_rgb>;
        };
    };
};
};

```

```

&display_subsystem {
    status = "okay";
};

```

Рисунок 12 – Display subsystem. Описание в dts перечня возможных для подключения

```

gpu: gpu@fde60000 {
    compatible = "arm,mali-bifrost";
    reg = <0x0 0xfde60000 0x0 0x4000>;

    interrupts = <GIC_SPI 39 IRQ_TYPE_LEVEL_HIGH>,
                <GIC_SPI 41 IRQ_TYPE_LEVEL_HIGH>,
                <GIC_SPI 40 IRQ_TYPE_LEVEL_HIGH>;
    interrupt-names = "GPU", "MMU", "JOB";

    upthreshold = <40>;
    downdifferential = <10>;

    clocks = <&scmi_clk 1>, <&cru CLK_GPU>;
    clock-names = "clk_mali", "clk_gpu";
    power-domains = <&power RK3568_PD_GPU>;
    #cooling-cells = <2>;
    operating-points-v2 = <&gpu_opp_table>;

    status = "disabled";
    gpu_power_model: power-model {
        compatible = "simple-power-model";
        leakage-range = <5 15>;
        ls = <(-24002) 22823 0>;
        static-coefficient = <100000>;
        dynamic-coefficient = <953>;
        ts = <(-108890) 63610 (-1355) 20>;
        thermal-zone = "gpu-thermal";
    };
};

```

```

&gpu {
    mali-supply = <&vdd_gpu>;
    status = "okay";
};

```

Рисунок 13 – GPU. Графический процессор Мали. Настройка драйвера

```

iep: iep@fdef0000 {
    compatible = "rockchip,iep-v2";
    reg = <0x0 0xfdef0000 0x0 0x500>;
    interrupts = <GIC_SPI 56 IRQ_TYPE_LEVEL_HIGH>;
    clocks = <&cru ACLK_IEP>, <&cru HCLK_IEP>, <&cru CLK_IEP_CORE>;
    clock-names = "aclk", "hclk", "sclk";
    resets = <&cru SRST_A_IEP>, <&cru SRST_H_IEP>,
            <&cru SRST_IEP_CORE>;
    reset-names = "rst_a", "rst_h", "rst_s";
    power-domains = <&power RK3568_PD_RGA>;
    rockchip,srv = <&mpp_srv>;
    rockchip,taskqueue-node = <5>;
    rockchip,resetgroup-node = <5>;
    iommu = <&iep_mmu>;
    status = "disabled";
};

```

```

&iep {
    status = "okay";
};

```

Рисунок 14 – IEP Image Enhancement Processor (IEP) преобразует форматы XRGB/RGB565/YUV420/YUV422 в формат RGB/RGB565/YUV420/YUV422

```

jpegd: jpegd@fded0000 {
    compatible = "rockchip,rkv-jpeg-decoder-v1";
    reg = <0x0 0xfdcd0000 0x0 0x400>;
    interrupts = <GIC_SPI 62 IRQ_TYPE_LEVEL_HIGH>;
    clocks = <&cru ACLK_JDEC>, <&cru HCLK_JDEC>;
    clock-names = "aclk_vcodec", "hclk_vcodec";
    rockchip,disable-auto-freq;
    resets = <&cru SRST_A_JDEC>, <&cru SRST_H_JDEC>;
    reset-names = "video_a", "video_h";
    iommu = <&jpegd_mmu>;
    rockchip,srv = <&mpp_srv>;
    rockchip,taskqueue-node = <1>;
    rockchip,resetgroup-node = <1>;
    power-domains = <&power RK3568_PD_RGA>;
    status = "disabled";
};

```

```

&jpegd {
    status = "okay";
};

```

Рисунок 15– jpegd. Jpeg аппаратный декодер

```

rk_rga: rk_rga@fdeb0000 {
    compatible = "rockchip,rga2";
    reg = <0x0 0xfdeb0000 0x0 0x1000>;
    interrupts = <GIC_SPI 90 IRQ_TYPE_LEVEL_HIGH>;
    clocks = <&cru ACLK_RGA>, <&cru HCLK_RGA>, <&cru CLK_RGA_CORE>;
    clock-names = "aclk_rga", "hclk_rga", "clk_rga";
    power-domains = <&power RK3568_PD_RGA>;
    status = "disabled";
};

```

```

&rk_rga {
    status = "okay";
};

```

Рисунок 16 – rk_rga RGA (2D raster graphic acceleration unit) - ускоритель 2D-графики

```

rkvdec: rkvdec@fdf80200 {
compatible = "rockchip,rkv-decoder-rk3568", "rockchip,rkv-decoder-v2";
reg = <0x0 0xdf80200 0x0 0x400>, <0x0 0xdf80100 0x0 0x100>;
reg-names = "regs", "link";
interrupts = <GIC_SPI 91 IRQ_TYPE_LEVEL_HIGH>;
interrupt-names = "irq_dec";
clocks = <&cru ACLK_RKVDEC>, <&cru HCLK_RKVDEC>,
        <&cru CLK_RKVDEC_CA>, <&cru CLK_RKVDEC_CORE>,
        <&cru CLK_RKVDEC_HEVC_CA>;
clock-names = "aclk_vcodec", "hclk_vcodec", "clk_cabac",
              "clk_core", "clk_hevc_cabac";
rockchip,normal-rates = <297000000>, <0>, <297000000>,
                       <297000000>, <600000000>;
rockchip,advanced-rates = <396000000>, <0>, <396000000>,
                          <396000000>, <600000000>;
rockchip,default-max-load = <2088960>;
resets = <&cru SRST_A_RKVDEC>, <&cru SRST_H_RKVDEC>,
        <&cru SRST_RKVDEC_CA>, <&cru SRST_RKVDEC_CORE>,
        <&cru SRST_RKVDEC_HEVC_CA>;
assigned-clocks = <&cru ACLK_RKVDEC>, <&cru CLK_RKVDEC_CA>,
                 <&cru CLK_RKVDEC_CORE>, <&cru CLK_RKVDEC_HEVC_CA>;
assigned-clock-rates = <297000000>, <297000000>, <297000000>, <297000000>;
reset-names = "video_a", "video_h", "video_cabac",
             "video_core", "video_hevc_cabac";
power-domains = <&power RK3568_PD_RKVDEC>;
operating-points-v2 = <&rkvdec_opp_table>;
vdec-supply = <&vdd_logic>;
iommu = <&rkvdec_mmu>;
rockchip,src = <&mpp_src>;
rockchip,taskqueue-node = <4>;
rockchip,resetgroup-node = <4>;
rockchip,sram = <&rkvdec_sram>;
/* rcb_iova: start and size */
rockchip,rcb-iova = <0x10000000 65536>;
rockchip,rcb-min-width = <512>;
rockchip,task-capacity = <16>;
status = "disabled";
};

```

```

&rkvdec {
    status = "okay";
};

```

Рисунок 17 – rkvdec. Видеодекодер для декодирования видео

```

rkvenc: rkvenc@fdf40000 {
compatible = "rockchip,rkv-encoder-v1";
reg = <0x0 0xdf40000 0x0 0x400>;
interrupts = <GIC_SPI 140 IRQ_TYPE_LEVEL_HIGH>;
interrupt-names = "irq_enc";
clocks = <&cru ACLK_RKVENC>, <&cru HCLK_RKVENC>,
        <&cru CLK_RKVENC_CORE>;
clock-names = "aclk_vcodec", "hclk_vcodec", "clk_core";
rockchip,normal-rates = <297000000>, <0>, <297000000>;
resets = <&cru SRST_A_RKVENC>, <&cru SRST_H_RKVENC>,
        <&cru SRST_RKVENC_CORE>;
reset-names = "video_a", "video_h", "video_core";
assigned-clocks = <&cru ACLK_RKVENC>, <&cru CLK_RKVENC_CORE>;
assigned-clock-rates = <297000000>, <297000000>;
iommu = <&rkvenc_mmu>;
node-name = "rkvenc";
rockchip,src = <&mpp_src>;
rockchip,taskqueue-node = <3>;
rockchip,resetgroup-node = <3>;
power-domains = <&power RK3568_PD_RKVENC>;
operating-points-v2 = <&rkvenc_opp_table>;
status = "disabled";
};

```

```

&rkvenc {
    vdec-supply = <&vdd_logic>;
    status = "okay";
};

```

Рисунок 18 – rkvenc. Видеоэнкодер для сжатия видео

```

vp0: port@0 {
    #address-cells = <1>;
    #size-cells = <0>;
    reg = <0>;

    vp0_out_dsi0: endpoint@0 {
        reg = <0>;
        remote-endpoint = <&dsi0_in_vp0>;
    };

    vp0_out_dsi1: endpoint@1 {
        reg = <1>;
        remote-endpoint = <&dsi1_in_vp0>;
    };

    vp0_out_edp: endpoint@2 {
        reg = <2>;
        remote-endpoint = <&edp_in_vp0>;
    };

    vp0_out_hdmi: endpoint@3 {
        reg = <3>;
        remote-endpoint = <&hdmi_in_vp0>;
    };
};

vp1: port@1 {
    #address-cells = <1>;
    #size-cells = <0>;
    reg = <1>;

    vp1_out_dsi0: endpoint@0 {
        reg = <0>;
        remote-endpoint = <&dsi0_in_vp1>;
    };

    vp1_out_dsi1: endpoint@1 {
        reg = <1>;
        remote-endpoint = <&dsi1_in_vp1>;
    };

    vp1_out_edp: endpoint@2 {
        reg = <2>;
        remote-endpoint = <&edp_in_vp1>;
    };

    vp1_out_hdmi: endpoint@3 {
        reg = <3>;
        remote-endpoint = <&hdmi_in_vp1>;
    };

    vp1_out_lvds: endpoint@4 {
        reg = <4>;
        remote-endpoint = <&lvds_in_vp1>;
    };
};

vp2: port@2 {
    #address-cells = <1>;
    #size-cells = <0>;
    reg = <2>;

    vp2_out_lvds: endpoint@0 {
        reg = <0>;
        remote-endpoint = <&lvds_in_vp2>;
    };

    vp2_out_rgb: endpoint@1 {
        reg = <1>;
        remote-endpoint = <&rgb_in_vp2>;
    };
};

```

Рисунок 19 – Video out ports. Виртуальные видеопорты для коммутации с физическим интерфейсом

7.4.1 HDMI

Описание видеовыхода HDMI состоит из настройки вывода звука (HDMI_sound, Рис. 21) и вывода видео (HDMI, Рис. 22). Для вывода HDMI-сигнала используется порт vp1.

```

hdmi_sound: hdmi-sound {
    compatible = "simple-audio-card";
    simple-audio-card,format = "i2s";
    simple-audio-card,mclk-fs = <128>;
    simple-audio-card,name = "rockchip-hdmi0";
    status = "okay";

    simple-audio-card,cpu {
        sound-dai = <&i2s0_8ch>;
    };
    simple-audio-card,codec {
        sound-dai = <&hdmi>;
    };
};

```

Рисунок 20 – HDMI_sound. Настройка вывода звука в HDMI-разъём

```

&hdmi {
    status = "okay";
    pinctrl-0 = <&hdmitx_scl &hdmitx_sda &hdmitxm0_cec>;
};

&hdmi_in_vp0 {
    status = "disabled";
};

&hdmi_in_vp1 {
    status = "okay";
};

&route_hdmi {
    connect = <&vp1_out_hdmi>;
};|

```

Рисунок 21 – Настройка порта для вывода HDMI-сигнала

7.4.2 LVDS. На примере панели G070ACE-L01

Размер панели 800x480. Диагональ 7 дюймов. Подключается к разъёму XP3.

7.4.2.3 Backlight configurations

Для работы подсветки матрицы необходимо указать вывод, подающий на неё питание (pwm6). PWM — широтно-импульсная модуляция. Она настраивается переменным значением от 0 (отсутствие напряжения) до 255 (3.3 V). Яркость подсветки зависит от напряжения. Однако 50 % напряжения не соответствует 50 % яркости, и требуется массив констант, обеспечивающий линейность отношения яркости подсветки к проценту открытия PWM.

Настройка яркости в графической оболочке осуществляется в меню настройки дисплея. Также настроить яркость можно с помощью консольных команд.

Чтение значения максимальной яркости:

```
cat /sys/classes/backlight/backlight-lvds/max_brightness
```

Вывело число 255.

Установка максимальной яркости:

```
echo 255 > /sys/classes/backlight/backlight-lvds/brightness
```

Установка минимальной яркости (отключение подсветки):

```
echo 0 > /sys/classes/backlight/backlight-lvds/brightness
```

7.4.2.4 Display timing configurations

Настройка формата видеосигнала производится в файле /build_os_img/dts/rk3568-mcp905-lvds.dtsi.

```

lvds_panel: lvds-panel {
    status = "okay";
    compatible = "simple-panel";
    backlight = <&backlight_lvds>;
    power-supply = <&vcc3v3_lcd_lvds>;
    pinctrl-names = "default";
    bus-format = <MEDIA_BUS_FMT_RGB888_1X24>;
    //rockchip,data-mapping = "vesa";

    enable-delay-ms = <20>;
    prepare-delay-ms = <20>;
    unprepare-delay-ms = <20>;
    disable-delay-ms = <20>;
    width-mm = <217>;
    height-mm = <136>;

    display-timings {
        native-mode = <&timing0>;

        timing0: timing0 {
            clock-frequency = <29500000>;
            hactive = <800>;
            vactive = <480>;
            hsync-len = <96>;
            hback-porch = <96>;
            hfront-porch = <24>;
            vsync-len = <7>;
            vback-porch = <10>;
            vfront-porch = <3>;
            hsync-active = <0>;
            vsync-active = <0>;
            de-active = <0>;
            pixelclk-active = <0>;
        };
    };
};

```

Рисунок 23 – LVDS panel timing

7.4.3 eDP

eDP (Embedded Display Port) — встроенный порт дисплея. Является стандартом VESA (Video Electronics Standard Association). Содержит 4 дифференциальных линии данных и одну дифференциальную линию синхронизации.

Для работы eDP в дереве устройств необходимо выбрать видеопорт (например, vp0). При этом иные видеointерфейсы не должны выводиться на этот порт. Также следует обратить внимание на доступность видеointерфейса на видеопорте (Рис. 29)

```

backlight_edp: backlight-edp {
    status = "okay";
    compatible = "pwm-backlight";
    pwms = <&pwm7 0 25000 0>;
    brightness-levels = <
        0 20 20 21 21 22 22 23
        23 24 24 25 25 26 26 27
        27 28 28 29 29 30 30 31
        31 32 32 33 33 34 34 35
        35 36 36 37 37 38 38 39
        40 41 42 43 44 45 46 47
        48 49 50 51 52 53 54 55
        56 57 58 59 60 61 62 63
        64 65 66 67 68 69 70 71
        72 73 74 75 76 77 78 79
        80 81 82 83 84 85 86 87
        88 89 90 91 92 93 94 95
        96 97 98 99 100 101 102 103
        104 105 106 107 108 109 110 111
        112 113 114 115 116 117 118 119
        120 121 122 123 124 125 126 127
        128 129 130 131 132 133 134 135
        136 137 138 139 140 141 142 143
        144 145 146 147 148 149 150 151
        152 153 154 155 156 157 158 159
        160 161 162 163 164 165 166 167
        168 169 170 171 172 173 174 175
        176 177 178 179 180 181 182 183
        184 185 186 187 188 189 190 191
        192 193 194 195 196 197 198 199
        200 201 202 203 204 205 206 207
        208 209 210 211 212 213 214 215
        216 217 218 219 220 221 222 223
        224 225 226 227 228 229 230 231
        232 233 234 235 236 237 238 239
        240 241 242 243 244 245 246 247
        248 249 250 251 252 253 254 255
    >;
    default-brightness-level = <200>;
    /*enable-gpios = <&gpio3 RK_PA5 GPIO_ACTIVE_HIGH>;*/
    power-supply = <&vcc3v3_lcd_edp>;
};

```

Рисунок 24– Подсветка панели eDP

```

edp_panel: edp-panel {
    status = "okay";
    compatible = "simple-panel";

    backlight = <&backlight_edp>;
    vsp-supply = <&vcc3v3_lcd_edp>;
    vsn-supply = <&vcc3v3_lcd_edp>;
    power-supply = <&vcc3v3_lcd_edp>;
    enable-gpio = <&gpio0 RK_PD6 GPIO_ACTIVE_HIGH>;

    display-timings {
        native-mode = <&edp_panel_timing1>;

        edp_panel_timing1: timing1 {
            clock-frequency = <80000000>;
            hactive = <3840>;
            vactive = <2160>;

            hfront-porch = <48>; // Simply guar
            hsync-len = <32>;
            hback-porch = <160>;

            vfront-porch = <3>; // simply guar
            vsync-len = <5>;
            vback-porch = <62>;

            hsync-active = <0>;
            vsync-active = <0>;
            de-active = <0>;
            pixelclk-active = <0>;
        };
    };

    ports {
        edp_panel_in: endpoint {
            remote-endpoint = <&edp_out>;
        };
    };
};

```

Рисунок 25 – Панель eDP

7.4.4 DSI

Display Serial Interface (DSI) — интерфейс, определяющий высокоскоростную дифференциальную передачу сигналов точка-точка по последовательной шине. Эта шина включает в себя одну линию высокой тактовой частоты и одну или несколько линий данных. По всем линиям сигналы передаются от хоста DSI к устройству DSI, за исключением первой линии данных (линия 0), которая поддерживает функцию разворота шины (bus turnaround, BTA), то есть может менять направление передачи.

В модуле МЦП905 одновременное использование LVDS и DSI-интерфейсов невозможно, так как эти интерфейсы используют одни и те же выводы RK3568. Переключение режимов LVDS / DSI производится в Device Tree-файле.

7.5 GPIO

Интерфейс ввода/вывода общего назначения (англ. general-purpose input/output, GPIO) — интерфейс для связи между компонентами компьютерной системы. Линии (выводы) GPIO могут быть сконфигурированы как вход или выход. Линия, настроенная как вход, может генерировать прерывание. В RK3568 4 группы линий GPIO. Линия может иметь три состояния: высокое (уровень питания 3.3 V), низкое (уровень земли 0.0 V) и Z-состояние (напряжение не определено).

7.5.1 Описание вывода в dts

Все выводы, относящиеся к группам GPIO, описываются в файле rk3568-pinctrl.dtsi. После этого в dts-файле к выводу обращаются по имени. Пример: audiorpwm_lout. Находится в группе GPIO1. Индекс в группе A0. Функция 4 (AUDIOPWM_LOUT_P). Параметр &pcfg_pull_* описывает режим подтяжки вывода внутренними резисторами.

```
audiorpwm_lout: audiorpwm-lout {
    rockchip,pins =
        /* audiorpwm_lout */
        <1 RK_PA0 4 &pcfg_pull_none>;
};
```

Рисунок 26 – audio pwm lout pin



Рисунок 27– Вырезка из схемы подключения RK3568

Значения функций нужно смотреть в описании RK3568 или в схеме (на Рис. 27 представлена вырезка из схемы с подписями номеров функций).

Варианты подтяжки вывода:

- `pcfg_pull_[up, down, none]` (это сокращённая запись `pcfg_pull_up`, `pcfg_pull_down`, `pcfg_pull_none`)
- `pcfg_pull_[up, down, none]_drv_level_[0...15]`
- `pcfg_pull_[up, down, none]_smt`

`up` означает подтяжку через резистор к 3.3 V.

`down` означает подтяжку через резистор к GND.

`none` означает высокоимпедансное состояние вывода.

`drv_level` означает “drive strength” (силу тока на выходе).

`smt` означает подключение триггера Шмидта.

7.5.2 Программное управление линией GPIO (на примере USER-LED)

Рассмотрим схему подключения пользовательского светодиода. Из схемы видно, что он подключен к третьей группе GPIO, индекс A3. Для включения нужен высокий уровень.

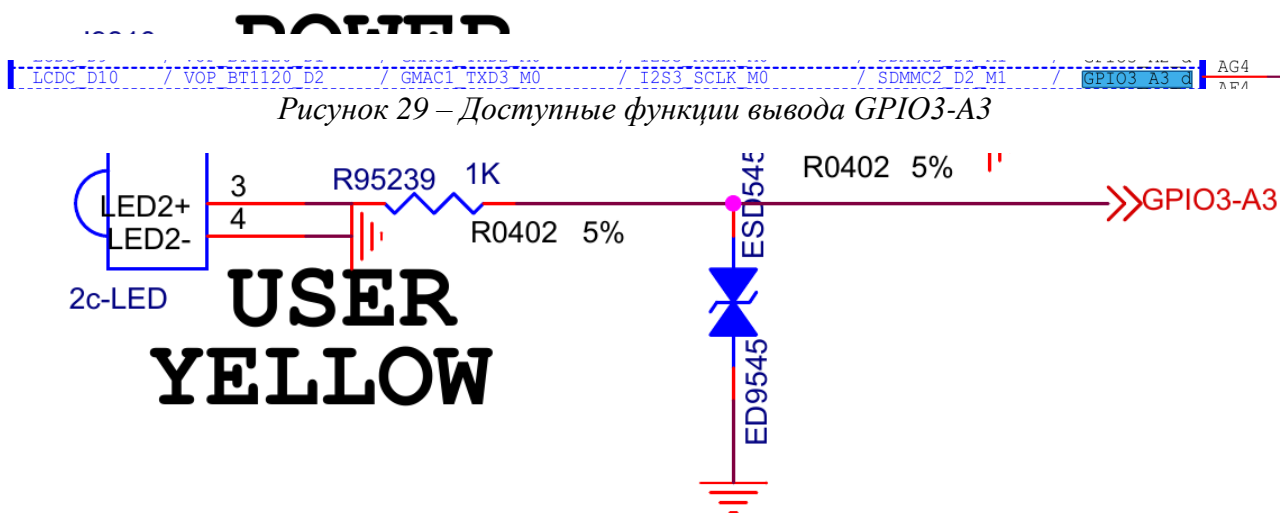


Рисунок 29 – Доступные функции вывода GPIO3-A3

Рисунок 28 – USER LED. Подключение

7.5.2.1 Программное управление выводом драйвером gpio-leds

Описание вывода пользовательского светодиода делается в файле `/build_os_img/dts/rk3568-pinctrl.dtsi`. Здесь необходимо написать, что вывод из третьей группы GPIO, индекс А3, функцию необходимо указать нулевую (RK_FUNC_GPIO).

```
leds {
    user_led: user-led {
        rockchip,pins = <3 RK_PA3 RK_FUNC_GPIO &pcfg_pull_none>;
    };
};
```

Рисунок 30 – USER LED. Конфигурация вывода

Для настройки драйвера gpio-leds используется поле user-led gpios. В нём описывается вывод и метод его включения GPIO_ACTIVE_HIGH (включение высоким уровнем). Есть вариант GPIO_ACTIVE_LOW (включение низким уровнем).

```
gpio_leds: gpio-leds {
    compatible = "gpio-leds";
    status = "okay";

    user-led {
        gpios = <&gpio3 RK_PA3 GPIO_ACTIVE_HIGH>;
        label = "disk-activity";
        /*linux,default-trigger = "heartbeat";*/
        linux,default-trigger = "activity";
        default-state = "on";
        pinctrl-0 = <&user_led>;
    };
};
```

Рисунок 31 – USER LED. Конфигурация драйвера

В описанном выше примере выводом управляет драйвер gpio-leds и зажигает светодиод в режиме индикации внутренней активности ЦПУ и активности дисков (режим activity).

Через управляющую папку `/sys` можно настроить драйвер gpio-leds. Путь к управлению выводом `/sys/class/leds/user-led`. Светодиод можно включить/выключить записью константы от 0 до 255 в файл, отвечающий за яркость.

```
echo 0 > ./brightness echo 255 > ./brightness
```

Режим «пульсации» можно поменять на один из доступных:

```
root@mcp905:/sys/class/leds/user-led# cat ./trigger
none rfkill-any rfkill-none kbd-scrolllock kbd-numlock kbd-capslock kbd-kanalock kb
d-shiftlock kbd-altgrlock kbd-ctrllock kbd-altlock kbd-shiftllock kbd-shiftrlock kb
d-ctrllock kbd-ctrlrlock usbport usb-gadget usb-host mmc1 timer oneshot disk-activ
ity disk-read disk-write ide-disk mtd nand-disk [heartbeat] backlight gpio cpu cpu0
cpu1 cpu2 cpu3 mmc0 activity default-on transient flash torch panic netdev
```

Например, светодиод можно настроить на отображение активности диска командой:

```
echo disk-activity > ./trigger
```

7.5.2.2 Программное управление выводом драйвером GPIO

```
/*gpio_leds: gpio-leds {
    compatible = "gpio-leds";
    status = "okay";

    user-led {
        gpios = <&gpio3 RK_PA3 GPIO_ACTIVE_HIGH>;
        linux,default-trigger = "heartbeat";
        default-state = "on";
        pinctrl-0 = <&user_led>;
    };
};*/
```

Рисунок 32 – USER LED. Отключенная конфигурация драйвера

Для управления выводом GPIO только в ручном режиме необходимо убрать все привязки вывода в dts и оставить только его описание в секции pinctrl.

Управление выводом производится из консоли по индексу. Ниже переведена таблица соответствия групп выводов и индексов (например, GPIO3 A3 соответствует индексу 99).

Таблица 6 – Соответствие вывода и номера

GPIO0		GPIO1		GPIO2		GPIO3		GPIO4	
A0	0	A0	32	A0	64	A0	96	A0	128
A1	1	A1	33	A1	65	A1	97	A1	129
A2	2	A2	34	A2	66	A2	98	A2	130
A3	3	A3	35	A3	67	A3	99	A3	131
A4	4	A4	36	A4	68	A4	100	A4	132
A5	5	A5	37	A5	69	A5	101	A5	133
A6	6	A6	38	A6	70	A6	102	A6	134
A7	7	A7	39	A7	71	A7	103	A7	135
B0	8	B0	40	B0	72	B0	104	B0	136
B1	9	B1	41	B1	73	B1	105	B1	137
B2	10	B2	42	B2	74	B2	106	B2	138
B3	11	B3	43	B3	75	B3	107	B3	139
B4	12	B4	44	B4	76	B4	108	B4	140
B5	13	B5	45	B5	77	B5	109	B5	141
B6	14	B6	46	B6	78	B6	110	B6	142
B7	15	B7	47	B7	79	B7	111	B7	143
C0	16	C0	48	C0	80	C0	112	C0	144
C1	17	C1	49	C1	81	C1	113	C1	145
C2	18	C2	50	C2	82	C2	114	C2	146
C3	19	C3	51	C3	83	C3	115	C3	147
C4	20	C4	52	C4	84	C4	116	C4	148
C5	21	C5	53	C5	85	C5	117	C5	149
C6	22	C6	54	C6	86	C6	118	C6	150
C7	23	C7	55	C7	87	C7	119	C7	151
D0	24	D0	56	D0	88	D0	120	D0	152
D1	25	D1	57	D1	89	D1	121	D1	153
D2	26	D2	58	D2	90	D2	122	D2	154
D3	27	D3	59	D3	91	D3	123	D3	155
D4	28	D4	60	D4	92	D4	124	D4	156
D5	29	D5	61	D5	93	D5	125	D5	157
D6	30	D6	62	D6	94	D6	126	D6	158
D7	31	D7	63	D7	95	D7	127	D7	159

Для корректного ручного управления выводом из консоли необходимо включить управление в драйвере gpio командой `echo 99 > /sys/class/gpio/export`

Выдача значения в вывод 99:

1. Командой `echo out > /sys/class/gpio/gpio99/direction` настроить вывод на выдачу уровня.
2. Командой `echo 1 > /sys/class/gpio/gpio99/value` настроить уровень на высокий. (0 - низкий)

Получение значения в вывод 99:

1. Командой `echo in > /sys/class/gpio/gpio99/direction` настроить вывод на чтение уровня.
2. Командой `cat /sys/class/gpio/gpio99/value` прочитать уровень (1 — высокий, 0 — низкий).

7.5.3 Занятые выводы GPIO

Просмотр используемых выводов, их состояний и привязанных к ним драйверов осуществляется командой `cat /sys/kernel/debug/gpio`

```

root@mcp905:/sys/kernel/debug# cat /sys/kernel/debug/gpio
gpiochip0: GPIOs 0-31, parent: platform/fdd60000.gpio, gpio0:
  gpio-28 (                |pwn          ) out lo

gpiochip1: GPIOs 32-63, parent: platform/fe740000.gpio, gpio1:

gpiochip2: GPIOs 64-95, parent: platform/fe750000.gpio, gpio2:
  gpio-88 (                |pwn          ) out lo
  gpio-94 (                |reset        ) out hi
  gpio-95 (                |vcc5v0_host ) out hi

gpiochip3: GPIOs 96-127, parent: platform/fe760000.gpio, gpio3:
  gpio-97 (                |reset        ) out hi
  gpio-104 (               |mdio-reset   ) out hi
  gpio-111 (               |mdio-reset   ) out hi
  gpio-124 (               |vcc3v3_pcie  ) out hi

gpiochip4: GPIOs 128-159, parent: platform/fe770000.gpio, gpio4:
gpiochip5: GPIOs 511-511, parent: platform/rk805-pinctrl, rk817-gpio, can sleep:

```

Включим вывод 99 и проверим его статус. Как видим, он под управлением драйвера sysfs. Это и есть драйвер управления через папку /sys.

```

root@mcp905:/sys/kernel/debug# echo 99 > /sys/class/gpio/export
root@mcp905:/sys/kernel/debug# cat /sys/kernel/debug/gpio
gpiochip0: GPIOs 0-31, parent: platform/fdd60000.gpio, gpio0:
  gpio-28 (                |pwn          ) out lo

gpiochip1: GPIOs 32-63, parent: platform/fe740000.gpio, gpio1:

gpiochip2: GPIOs 64-95, parent: platform/fe750000.gpio, gpio2:
  gpio-88 (                |pwn          ) out lo
  gpio-94 (                |reset        ) out hi
  gpio-95 (                |vcc5v0_host ) out hi

gpiochip3: GPIOs 96-127, parent: platform/fe760000.gpio, gpio3:
  gpio-97 (                |reset        ) out hi
  gpio-99 (                |sysfs        ) in hi
  gpio-104 (               |mdio-reset   ) out hi
  gpio-111 (               |mdio-reset   ) out hi
  gpio-124 (               |vcc3v3_pcie  ) out hi

gpiochip4: GPIOs 128-159, parent: platform/fe770000.gpio, gpio4:
gpiochip5: GPIOs 511-511, parent: platform/rk805-pinctrl, rk817-gpio, can sleep:

```

7.6 SPI-шина

SPI (Serial Peripheral Interface, SPI bus — последовательный периферийный интерфейс, шина SPI) — последовательный синхронный стандарт передачи данных в режиме полного дуплекса, предназначенный для обеспечения простого и недорогого высокоскоростного сопряжения микроконтроллеров и периферии. SPI также иногда называют четырёхпроводным (англ. four-wire) интерфейсом.

SPI является синхронным интерфейсом, в котором любая передача синхронизирована с общим тактовым сигналом (SCLK - Serial Clock), генерируемым ведущим устройством (процессором).

Данные передаются в дуплексном режиме сигналами: MOSI (Master Output Slave Input — ведущий передает, ведомый принимает) и MISO (Master Input Slave Output — ведущий принимает, ведомый передает). Принимающая (ведомая) периферия синхронизирует получение битовой последовательности с тактовым сигналом. К одному последовательному периферийному интерфейсу ведущего устройства-микросхемы могут присоединяться несколько микросхем. Ведущее устройство выбирает ведомое для передачи, активируя сигнал SS (Slave Select — выбор ведомого) на ведомой микросхеме. Периферия, не получившая данный сигнал, не принимает участия в передаче по SPI.

Есть три способа подключения Ведущего и Ведомых.

1. Ведомый один, одноименные выходы просто соединяются напрямую.

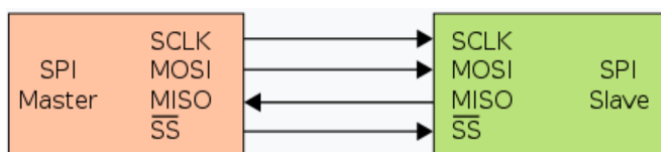


Рисунок 33 – Прямое подключение SPI

2. Несколько Ведомых соединяются с Ведущим параллельно шинами MISO, MOSI и SCK, а SS ведет к каждому из них индивидуально (классический случай).

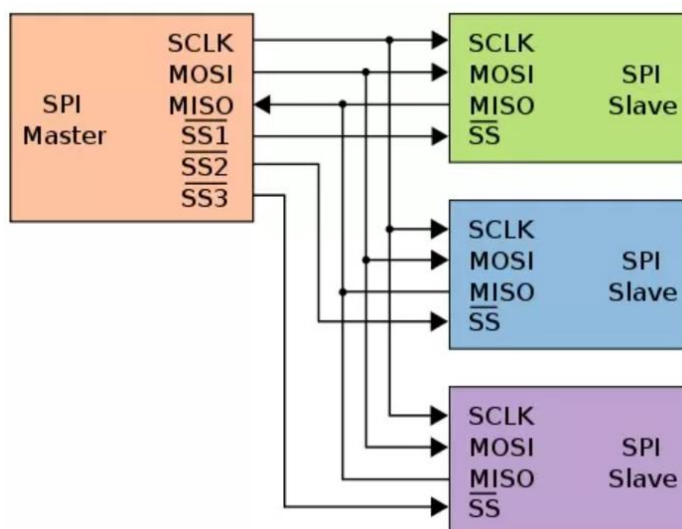


Рисунок 34 – Классическое подключение SPI

3. Цепочка. Он же кольцо. SS один на всех, но данные передаются как бы сквозь устройства с одного на другое. Пока SS в низком уровне, данные растекаются по своим местам, после подъема уровня SS устройства начинают работать с принятыми данными. Очевидный плюс: от контроллера требуется меньше выводов, но и большой минус: далеко не все устройства поддерживают сквозную передачу данных.

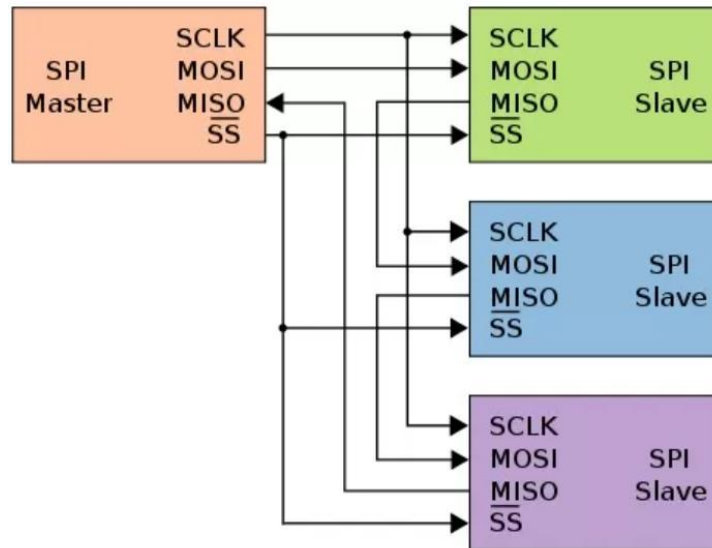


Рисунок 35 – Подключение SPI устройств кольцом

```
spi3: spi@fe640000 {
    compatible = "rockchip,rk3066-spi";
    /*compatible = "rockchip,spidev";*/

    reg = <0x0 0xfe640000 0x0 0x1000>;
    interrupts = <GIC_SPI 106 IRQ_TYPE_LEVEL_HIGH>;
    #address-cells = <1>;
    #size-cells = <0>;
    clocks = <&cru CLK_SPI3>, <&cru PCLK_SPI3>;
    clock-names = "spiclk", "apb_pclk";
    dmas = <&dmac0 26>, <&dmac0 27>;
    dma-names = "tx", "rx";
    pinctrl-names = "default", "high_speed";
    pinctrl-0 = <&spi3m1_cs0 &spi3m1_pins>;
    pinctrl-1 = <&spi3m1_cs0_hs &spi3m1_pins_hs>;
    status = "okay";
};
```

```
&spi3 {
    status = "okay";
    #address-cells = <1>;
    #size-cells = <0>;
    pinctrl-names = "default", "high_speed";
    pinctrl-0 = <&spi3m1_cs0 &spi3m1_pins>;
    pinctrl-1 = <&spi3m1_cs0 &spi3m1_pins_hs>;
    max-freq = <500000000>;

    spidev {
        compatible = "rockchip,spidev";
        status = "okay";
        reg = <0>;
        spi-max-frequency = <500000000>;
    };
};
```

Рисунок 36 – spi3. Описание шины SPI #3, выведенный на разъём mikroBUS (XS18/XS19)

7.6.1 Подключение к шине SPI

В модуле МЦП1502 одна шина SPI с двумя сигналами CS, построенная по классической схеме. В МЦП905 шина выведена на разъём MikroBUS с CS0.

```

spi3m1_pins: spi3m1-pins {
    rockchip,pins =
        /* spi3_clkm1 */
        <4 RK_PC2 2 &pcfg_pull_none>,
        /* spi3_misom1 */
        <4 RK_PC5 2 &pcfg_pull_none>,
        /* spi3_mosim1 */
        <4 RK_PC3 2 &pcfg_pull_none>;
};

```

Рисунок 37 – Описание контроллера SPI

7.6.2 Установленные устройства на шине SPI

В модуле МЦП905 нет встроенных устройств на шине SPI. Внешние вы можете подключить через разъем MikroBUS.

Также необходимо описать подключенное устройство в dts.

```

&spi3 {
    status = "okay";
    #address-cells = <1>;
    #size-cells = <0>;
    pinctrl-names = "default", "high_speed";
    pinctrl-0 = <&spi3m1_cs0 &spi3m1_pins>;
    pinctrl-1 = <&spi3m1_cs0 &spi3m1_pins_hs>;
    max-freq = <50000000>;

    spidev {
        compatible = "rockchip,spidev";
        status = "okay";
        reg = <0>;
        spi-max-frequency = <50000000>;
    };

    gyro: gyro@1 {
        status = "okay";
        compatible = "st,l3gd20-gyro";
        reg = <0>;

        vdd-supply = <&vcc3v3_sys>;
        vddio-supply = <&vcc3v3_sys>;
    };
};

```

Рисунок 38 – Описание подключения модуля GYRO click к шине SPI

Пример описания подключения модуля гироскопа MIKROE-1379, GYRO click.

7.7 UART

UART (Universal Asynchronous Receiver-Transmitter) — универсальный асинхронный приёмопередатчик) — узел вычислительных устройств, преобразующий передаваемые данные в последовательный вид так, чтобы передать их по одной физической цифровой линии другому аналогичному устройству.

Общепринятый ряд стандартных скоростей: 300; 600; 1200; 2400; 4800; 9600; 19200; 38400; 57600; 115200; 230400; 460800; 921600 бод.

На основе логического протокола UART были разработаны такие распространенные физические реализации, как токовая петля, RS-232, RS-485, RS-422.

TTL full-duplex работает по 3 проводам с уровнями напряжения (3.3/5 В): GND, RX, TX.

RS-232 full-duplex работает минимум по 3 проводам с уровнями сигналов $\pm 5 \dots \pm 12$ В: GND, RX, TX.

RS-422 full-duplex работает по двум дифференциальным парам с уровнями сигналов $\pm 7 \dots \pm 12$ В (всего 4 провода): A, B; Y, Z.

RS-485 half-duplex работает по одной дифференциальной паре с уровнями сигналов $\pm 7 \dots \pm 12$ В (всего 2 провода): A, B.

Тестирование работоспособности UART производится программой `minicom`. Для доступа к каждому порту должны быть запущены по одному экземпляру программы. В них настраивается символьное устройство для обмена (напр `/dev/ttyS0` для UART0) и скорость обмена.

7.7.1 Список выведенных на разъёмы UART

На модуле МЦП905 выведены интерфейсы:

1. Debug TTL (1,8 В): XP1 (UART2);
2. RS-485/RS422: XS14 (UART4), XS15 (UART0);
3. RS-232: XP10 (UART5);
4. mikroBUS TTL (1,8 В): XS18/XS19 (UART7).

```
uart5: serial@fe690000 {
    compatible = "rockchip,rk3568-uart", "snps,dw-apb-uart";
    reg = <0x0 0xfe690000 0x0 0x100>;
    interrupts = <GIC_SPI 121 IRQ_TYPE_LEVEL_HIGH>;
    clocks = <&cru SCLK_UART5>, <&cru PCLK_UART5>;
    clock-names = "baudclk", "apb_pclk";
    reg-shift = <2>;
    reg-io-width = <4>;
    dmas = <&dmac0 10>, <&dmac0 11>;
    pinctrl-names = "default";
    pinctrl-0 = <&uart5m0_xfer>;
    status = "disabled";
};
```

Рисунок 39 – UART5. RS232 на разъёме XP10

```

uart7: serial@fe6b0000 {
    compatible = "rockchip,rk3568-uart", "snps,dw-apb-uart";
    reg = <0x0 0xfe6b0000 0x0 0x100>;
    interrupts = <GIC_SPI 123 IRQ_TYPE_LEVEL_HIGH>;
    clocks = <&cru SCLK_UART7>, <&cru PCLK_UART7>;
    clock-names = "baudclk", "apb_pclk";
    reg-shift = <2>;
    reg-io-width = <4>;
    dmas = <&dmac0 14>, <&dmac0 15>;
    pinctrl-names = "default";
    pinctrl-0 = <&uart7m0_xfer>;
    status = "disabled";
};

```

Рисунок 414 – UART7. TTL выведен на разъём mikroBUS XS18/XS19

```

uart0: serial@fdd50000 {
    compatible = "rockchip,rk3568-uart", "snps,dw-apb-uart";
    reg = <0x0 0xfdd50000 0x0 0x100>;
    interrupts = <GIC_SPI 116 IRQ_TYPE_LEVEL_HIGH>;
    clocks = <&pmucru SCLK_UART0>, <&pmucru PCLK_UART0>;
    clock-names = "baudclk", "apb_pclk";
    reg-shift = <2>;
    reg-io-width = <4>;
    dmas = <&dmac0 0>, <&dmac0 1>;
    pinctrl-names = "default";
    pinctrl-0 = <&uart0_xfer>;
    status = "disabled";
};

```

```

&uart0 {
    status = "okay";
    pinctrl-0 = <&uart0_xfer &uart0_rtsn>;
};

```

Рисунок 405 – UART0. RS485/RS422 на разъёме XS15

```

uart4: serial@fe680000 {
    compatible = "rockchip,rk3568-uart", "snps,dw-apb-uart";
    reg = <0x0 0xfe680000 0x0 0x100>;
    interrupts = <GIC_SPI 120 IRQ_TYPE_LEVEL_HIGH>;
    clocks = <&cru SCLK_UART4>, <&cru PCLK_UART4>;
    clock-names = "baudclk", "apb_pclk";
    reg-shift = <2>;
    reg-io-width = <4>;
    dmas = <&dmac0 8>, <&dmac0 9>;
    pinctrl-names = "default";
    pinctrl-0 = <&uart4m0_xfer>;
    status = "disabled";
};

```

```

&uart4 {
    status = "okay";
    pinctrl-0 = <&uart4m0_xfer &uart4m0_rtsn>;
};

```

Рисунок 42. UART4. RS485/RS422 на разъёме XS14.

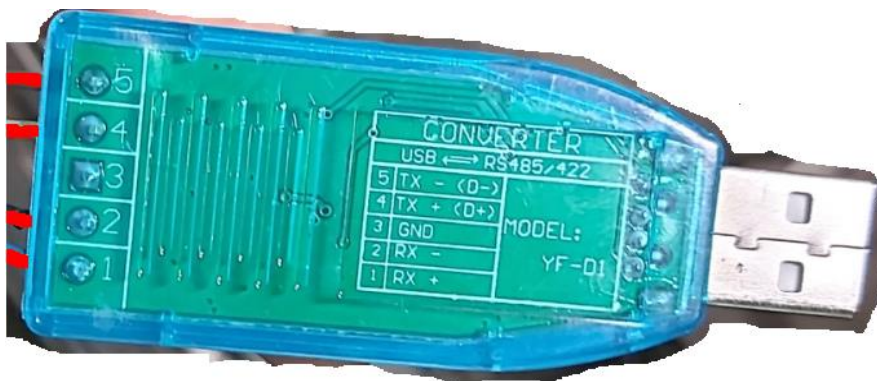
7.7.2 Подключение ПК к выведенным на разъёмы UART

Для подключения UART к ПК необходимы адаптеры, соответствующие типу интерфейса:

- TTL: Адаптер рассмотрен в подразделе 2.1 (USB to Serial adapter).
- [RS232: Usb To Rs232 Serial Adapter](#)



- RS485/RS422: USB в RS485/422 CH340, преобразователь.



7.8 CAN. Описание и схема подключения

CAN 2.0B (англ. Controller Area Network — сеть контроллеров) — стандарт промышленной сети, ориентированный, прежде всего, на объединение в единую сеть различных исполнительных устройств и датчиков. Режим передачи — последовательный, широкополосный, пакетный. На концах линии требуется включить терминальный резистор. Скорость обмена до 1 Мбит/с.

```
can0: can@fe570000 {
    compatible = "rockchip,rk3568-can-2.0";
    reg = <0x0 0xfe570000 0x0 0x1000>;
    interrupts = <GIC_SPI 1 IRQ_TYPE_LEVEL_HIGH>;
    clocks = <&cru CLK_CAN0>, <&cru PCLK_CAN0>;
    clock-names = "baudclk", "apb_pclk";
    resets = <&cru SRST_CAN0>, <&cru SRST_P_CAN0>;
    reset-names = "can", "can-apb";
    tx-fifo-depth = <1>;
    rx-fifo-depth = <6>;
    status = "disabled";
};
```

```
/* CAN bit rates <= 3M: set clock rates to 100M
 * CAN bit rates > 3M: set clock rates to 200M
 */
&can0 {
    compatible = "rockchip,can-1.0";
    assigned-clocks = <&cru CLK_CAN1>;
    assigned-clock-rates = <800000000>;
    pinctrl-names = "default";
    pinctrl-0 = <&can0m0_pins>;
    status = "okay";
};
```

Рисунок 43– CAN0. Описание первой шины CAN. Разъём XS17

```
can2: can@fe590000 {
    compatible = "rockchip,rk3568-can-2.0";
    reg = <0x0 0xfe590000 0x0 0x1000>;
    interrupts = <GIC_SPI 3 IRQ_TYPE_LEVEL_HIGH>;
    clocks = <&cru CLK_CAN2>, <&cru PCLK_CAN2>;
    clock-names = "baudclk", "apb_pclk";
    resets = <&cru SRST_CAN2>, <&cru SRST_P_CAN2>;
    reset-names = "can", "can-apb";
    tx-fifo-depth = <1>;
    rx-fifo-depth = <6>;
    status = "disabled";
};
```

```
&can2 {
    compatible = "rockchip,can-1.0";
    assigned-clocks = <&cru CLK_CAN2>;
    assigned-clock-rates = <800000000>;
    pinctrl-names = "default";
    pinctrl-0 = <&can2m1_pins>;
    status = "okay";
};
```

Рисунок 44 – CAN2. Описание второй шины CAN. Разъём XS16

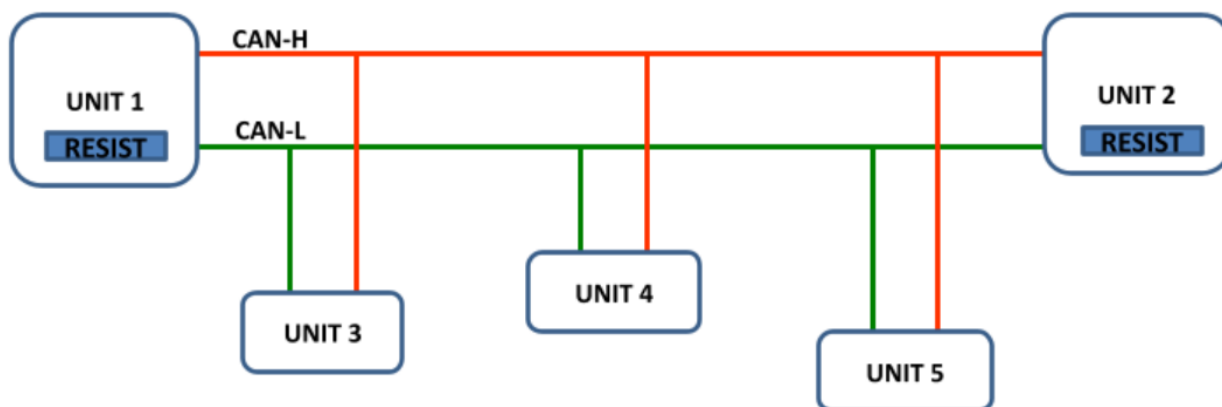


Рисунок 45 – Схема соединения абонентов интерфейса CAN

Пользователю доступны 2 CAN-интерфейса на разъёмах XS17 и XS16. Просмотр списка интерфейсов осуществляется командой `ip a`.

Для активации интерфейса и установки скорости следует выполнить команду:

```
ip link set up can0 type can bitrate 100000
```

```
ip link set up can1 type can bitrate 100000
```

Для деактивации интерфейса следует выполнить команду:

```
ip link set down can0 type can
```

```
ip link set down can1 type can
```

Для проверки работоспособности необходимо соединить одноименные линии интерфейсов CAN0_H к CAN1_H и CAN0_L к CAN1_L.

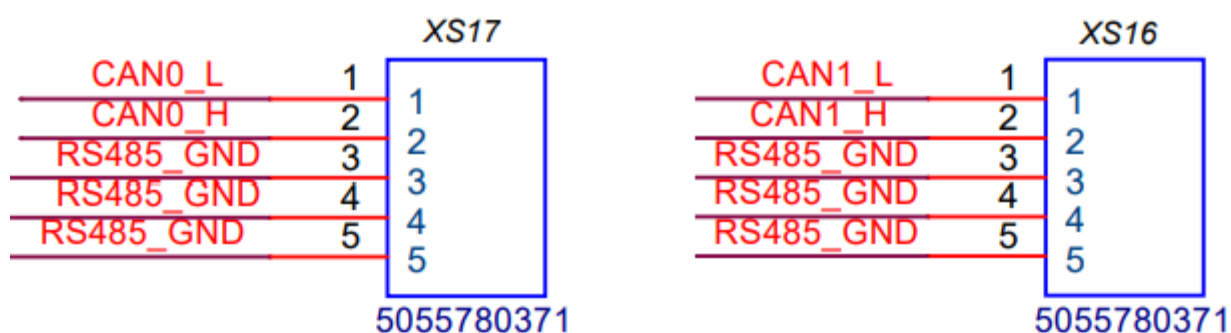


Рисунок 46 – Распиновка разъемов CAN

Для тестирования используются утилиты `cangen` и `sandump`.

7.9 RTC. Настройка и получение текущего времени

```
&i2c5 {
    status = "okay";
    #address-cells = <1>;
    #size-cells = <0>;

    hym8563: hym8563@51 {
        compatible = "haoyu,hym8563";
        reg = <0x51>;
        #clock-cells = <0>;
        clock-frequency = <32768>;
        clock-output-names = "hym8563";
        pinctrl-names = "default";
        pinctrl-0 = <&rtc_int>;

        interrupt-parent = <&gpio0>;
        interrupts = <RK_PD3 IRQ_TYPE_LEVEL_LOW>
    };
};
```

Рисунок 47– hym8563. Микросхема часов реального времени (RTC) подключена к интерфейсу i2c #5 адрес 0x51

RTC HYM8563 (CMOS_RTC) подключена к шине i2c #5.

Установка даты 28 ноября 2008 года производится командой `date +%Y%m%d -s "20081128"`

Установка времени 10 часов 13 мин производится командой `date +%T -s "10:13:13"`

Также для сохранения настройки времени в часы реального времени рекомендуется (в некоторых ОС это делается автоматически) запустить команду `hwclock -- systohc`

Чтение времени производится командой `date`

7.10 Аудио I2S. Схема подключения микрофона и наушников

Подраздел в разработке.

На разъёме XP5 выведены выводы для подключения стереонаушников и микрофона.

Наушники подключаются:

- левый: контакты 1, 4;
- правый: контакты 2, 4.

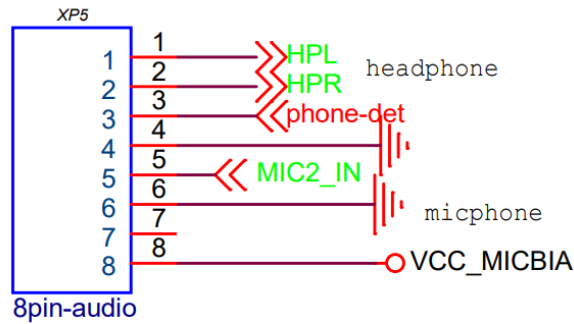


Рисунок 48 – Схема разъема аудио

Микрофон подключается контактами 5, 6, 8.

Автоматическое тестирование звука производится переключками: 1-5 контакты и 2-5 контакты. При этом в микрофоне будет слышно то, что приходит на динамики. Для этого осуществляется одновременное воспроизведение аудиодорожки и запись с микрофона. Успешным тестирование считается при совпадении записанной и воспроизведённой аудиодорожек.

Для ручного тестирования аудиоинтерфейса нужно подключить микрофон и наушники по схеме (Рис. 73). Программа для записи: agcord, для воспроизведения: aplay. Настройка громкости: alsamixer.

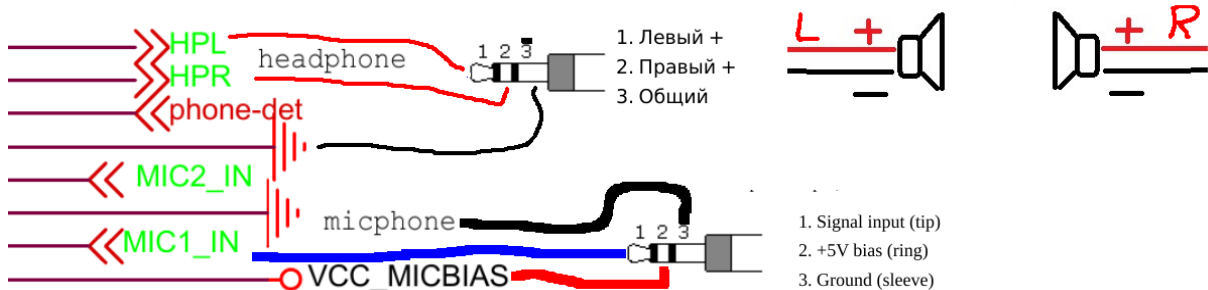


Рисунок 49 – Подключение микрофона и наушников

```

es8316_sound {
    compatible = "simple-audio-card";
    simple-audio-card,format = "i2s";
    simple-audio-card,name = "rockchip,rk-es8316-codec";
    simple-audio-card,mclk-fs = <256>;
    simple-audio-card,widgets =
        "Microphone", "Mic Jack",
        "Headphone", "Headphone Jack";
    simple-audio-card,routing =
        "Mic Jack", "MICBIAS1",
        "IN1P", "Mic Jack",
        "Headphone Jack", "HPOL",
        "Headphone Jack", "HPOR";
    simple-audio-card,cpu {
        sound-dai = <&i2s1_8ch>;
    };
    simple-audio-card,codec {
        sound-dai = <&es8316>;
    };
};

```

Рисунок 50 – es8316_sound. Настройка вывода звука в разъем для подключения Jack

```

i2s1_8ch: i2s@fe410000 {
    compatible = "rockchip,rk3568-i2s-tdm";
    reg = <0x0 0xfe410000 0x0 0x1000>;
    interrupts = <GIC_SPI 53 IRQ_TYPE_LEVEL_HIGH>;
    clocks = <&cru MCLK_I2S1_8CH_TX>, <&cru MCLK_I2S1_8CH_RX>, <&cru HCLK_I2S1_8CH>;
    clock-names = "mclk_tx", "mclk_rx", "hclk";
    dmas = <&dmac1 2>, <&dmac1 3>;
    dma-names = "tx", "rx";
    resets = <&cru SRST_M_I2S1_8CH_TX>, <&cru SRST_M_I2S1_8CH_RX>;
    reset-names = "tx-m", "rx-m";
    rockchip,cru = <&cru>;
    rockchip,grf = <&grf>;
    #sound-dai-cells = <0>;
    pinctrl-names = "default";
    pinctrl-0 = <&i2s1m0_sclktx
        &i2s1m0_sclkrx
        &i2s1m0_lrcktx
        &i2s1m0_lrckrx
        &i2s1m0_sdi0
        &i2s1m0_sdi1
        &i2s1m0_sdi2
        &i2s1m0_sdi3
        &i2s1m0_sdo0
        &i2s1m0_sdo1
        &i2s1m0_sdo2
        &i2s1m0_sdo3>;
    status = "disabled";
};

```

```

&i2s1_8ch {
    status = "okay";
    rockchip,playback-channels = <8>;
    rockchip,capture-channels = <8>;
    #sound-dai-cells = <0>;
    rockchip,clk-trcm = <1>;
    pinctrl-names = "default";
    pinctrl-0 = <&i2s1m1_sclktx
        &i2s1m1_lrcktx
        &i2s1m1_sdi0
        &i2s1m1_sdo0>;
};

```

Рисунок 51 – i2s1_8ch. Аудиодрайвер и описание аудиointерфейса

```

es8316_sound {
    compatible = "simple-audio-card";
    simple-audio-card,format = "i2s";
    simple-audio-card,name = "rockchip,rk-es8316-codec";
    simple-audio-card,mclk-fs = <256>;
    simple-audio-card,widgets =
        "Microphone", "Mic Jack",
        "Headphone", "Headphone Jack";
    simple-audio-card,routing =
        "Mic Jack", "MICBIAS1",
        "IN1P", "Mic Jack",
        "Headphone Jack", "HPOL",
        "Headphone Jack", "HPOR";
    simple-audio-card,cpu {
        sound-dai = <&i2s1_8ch>;
    };
    simple-audio-card,codec {
        sound-dai = <&es8316>;
    };
};

```

```

&i2c5 {
    status = "okay";

    es8316: es8316@11 {
        compatible = "everest,es8316";
        reg = <0x11>;
        clocks = <&cru I2S1_MCLKOUT>;
        clock-names = "mclk";
        pinctrl-names = "default";
        pinctrl-0 = <&i2s1m1_mclk>;
        assigned-clocks = <&cru I2S1_MCLKOUT>, <&cru I2S1_MCLK_TX_IOE>;
        assigned-clock-parents = <&cru I2S1_MCLKOUT_TX>, <&cru I2S1_MCLKOUT_TX>;
        assigned-clock-rates = <11289600>;
        #sound-dai-cells = <0>;
    };
};

```

Рисунок 52 – es8316. Аудиоконтроллер es8316 на шине i2c #5 адрес 0x11

7.11 Измерение напряжения SARADC_Vin (mikroBUS)

Подраздел в разработке.

```

saradc: saradc@fe720000 {
    compatible = "rockchip,rk3568-saradc", "rockchip,rk3399-saradc";
    reg = <0x0 0xfe720000 0x0 0x100>;
    interrupts = <GIC_SPI 93 IRQ_TYPE_LEVEL_HIGH>;
    #io-channel-cells = <1>;
    clocks = <&cru CLK_SARADC>, <&cru PCLK_SARADC>;
    clock-names = "saradc", "apb_pclk";
    resets = <&cru SRST_P_SARADC>;
    reset-names = "saradc-apb";
    status = "disabled";
};

```

Рисунок 53 – Saradc. Аналого-цифровой преобразователь, встроенный в rk3568

7.12 Выдача напряжения через PWM (mikroBUS)

Подраздел в разработке.

```
pwm3: pwm@fdd70030 {
    compatible = "rockchip,rk3568-pwm", "rockchip,rk3328-pwm";
    reg = <0x0 0xfdd70030 0x0 0x10>;
    interrupts = <GIC_SPI 82 IRQ_TYPE_LEVEL_HIGH>,
                <GIC_SPI 86 IRQ_TYPE_LEVEL_HIGH>;
    #pwm-cells = <3>;
    pinctrl-names = "active";
    pinctrl-0 = <&pwm3_pins>;
    clocks = <&pmucru CLK_PWM0>, <&pmucru PCLK_PWM0>;
    clock-names = "pwm", "pclk";
    status = "disabled";
};
```

Рисунок 54 – pwm3. pulse-width modulation - широтно-импульсная модуляция на выводе GPIO0, группа C2

Настройка PWM у вывода 1:

```
$ echo 1 | sudo tee /sys/class/pwm/pwmchip4/export
$ echo 1000000 | sudo tee /sys/class/pwm/pwmchip4/pwm1/period
$ echo 500000 | sudo tee /sys/class/pwm/pwmchip4/pwm1/duty_cycle
$ echo 1 | sudo tee /sys/class/pwm/pwmchip4/pwm1/enable
```

Отключение командой

```
$ echo 0 | sudo tee /sys/class/pwm/pwmchip4/pwm1/enable
```

7.13 Получение температуры кристалла от Temperature sensor

```
tsadc: tsadc@fe710000 {
    compatible = "rockchip,rk3568-tsadc";
    reg = <0x0 0xfe710000 0x0 0x100>;
    interrupts = <GIC_SPI 115 IRQ_TYPE_LEVEL_HIGH>;
    rockchip,grf = <&grf>;
    clocks = <&cru CLK_TSADC>, <&cru PCLK_TSADC>;
    clock-names = "tsadc", "apb_pclk";
    assigned-clocks = <&cru CLK_TSADC_TSEN>, <&cru CLK_TSADC>;
    assigned-clock-rates = <17000000>, <700000>;
    resets = <&cru SRST_TSADC>, <&cru SRST_P_TSADC>,
            <&cru SRST_TSADCPHY>;
    reset-names = "tsadc", "tsadc-apb", "tsadc-phy";
    #thermal-sensor-cells = <1>;
    nvmem-cells = <&tsadc_trim_base>, <&tsadc_trim_base_frac>;
    nvmem-cell-names = "trim_base", "trim_base_frac";
    rockchip,hw-tshut-temp = <120000>;
    rockchip,hw-tshut-mode = <0>; /* tshut mode 0:CRU 1:GPIO */
    rockchip,hw-tshut-polarity = <0>; /* tshut polarity 0:LOW 1:HIGH */
    pinctrl-names = "gpio", "otput";
    pinctrl-0 = <&tsadc_gpio_func>;
    pinctrl-1 = <&tsadc_shutorg>;
    #address-cells = <1>;
    #size-cells = <0>;
    status = "disabled";

    tsadc@0 {
        reg = <0>;
        nvmem-cells = <&cpu_tsadc_trim_l>, <&cpu_tsadc_trim_h>;
        nvmem-cell-names = "trim_l", "trim_h";
    };
    tsadc@1 {
        reg = <1>;
        nvmem-cells = <&gpu_tsadc_trim_l>, <&gpu_tsadc_trim_h>;
        nvmem-cell-names = "trim_l", "trim_h";
    };
};
```

Рисунок 55 – tsadc. АЦП датчика температуры

Получение сырых данных о температуре (49 градусов Цельсия) возможно через файловую систему /sys

```
cat /sys/class/thermal/thermal_zone*/temp
```

```
bosko@bosko:~$ cat /sys/class/thermal/thermal_zone*/temp
49000
```

Форматированный вывод возможен командой `paste <(cat /sys/class/thermal/thermal_zone*/type) <(cat /sys/class/thermal/thermal_zone*/temp) | column -s $'\t' -t | sed 's^(.)..$/\1°C/'`

```
bosko@bosko:~$ paste <(cat /sys/class/thermal/thermal_zone*/type) <(cat /sys/class/thermal/thermal_zone*/temp) | column -s '\t' -t | sed 's/\(.\)\.\.$/.1°C/'
x86_pkg_temp 49.0°C
```

7.14 Сторожевой таймер

Модуль МЦП905 содержит сторожевой таймер. Его параметры доступны суперпользователю в файловой системе `proc` по адресу `/proc/ctrl/watchdog`

Допустимый таймаут — от 1 до 255 секунд.

Порядок использования сторожевого таймера:

1. Установка таймаута записью количество секунд в файл файл `/proc/ctrl/watchdog/timeout_sec`.

```
echo 60 > /proc/ctrl/watchdog/timeout_sec
```

2. Включение сторожевого таймера записью `1` в файл `/proc/ctrl/watchdog/enable`

```
echo 1 > /proc/ctrl/watchdog/enable
```

3. Периодическое стробирование таймера записью `1` в файл `/proc/ctrl/watchdog/refresh`

```
echo 1 > /proc/ctrl/watchdog/refresh
```

Все параметры таймера можно считать из файла `/proc/ctrl/watchdog/state`.

```
cat /proc/ctrl/watchdog/state
```

7.15 Сопроцессор нейросетей NPU

Подраздел в разработке. Выключен.

```
rknpu: npu@fde40000 {
    compatible = "rockchip,rk3568-rknpu", "rockchip,rknpu";
    reg = <0x0 0xfde40000 0x0 0x10000>;
    interrupts = <GIC_SPI 151 IRQ_TYPE_LEVEL_HIGH>;
    clocks = <&scmi_clk 2>, <&cru CLK_NPU>, <&cru ACLK_NPU>, <&cru HCLK_NPU>;
    clock-names = "scmi_clk", "clk", "aclk", "hclk";
    assigned-clocks = <&cru CLK_NPU>;
    assigned-clock-rates = <600000000>;
    resets = <&cru SRST_A_NPU>, <&cru SRST_H_NPU>;
    reset-names = "srst_a", "srst_h";
    power-domains = <&power RK3568_PD_NPU>;
    operating-points-v2 = <&npu_opp_table>;
    iommu = <&rknpu_mmu>;
    status = "disabled";
};
```

```
&rknpu {
    rknpu-supply = <&vdd_npu>;
};
```

Рисунок 56 – Rknpu. Сопроцессор для нейронных сетей

7.16 PCI-E

Микропроцессор RK3568 имеет два контроллера PCIe gen.3 x1 lane (8 Gbit/s).

Поддерживаются режимы: 2 x1 lane и 1 x2 lane.

На модуле МЦП905 установлены два слота miniPCIe. На каждый выведена 1 линия PCIe.

7.17 SATA

Микропроцессор RK3568 имеет выводы для подключения SATA-накопителя. Они выведены на разъём M.2 NGFF_KEY_M.

