

# **ЯДРО LINUX ДЛЯ 1892ВМ14Я. РУКОВОДСТВО СИСТЕМНОГО ПРОГРАММИСТА**

**Версия v2.8  
29.03.2018**

## ОГЛАВЛЕНИЕ

<b>1</b>	<b>О документе</b>	<b>3</b>
<b>2</b>	<b>Подсистема управления тактовыми сигналами</b>	<b>4</b>
<b>3</b>	<b>Драйвер контроллера PWM <i>pwm-tcom</i></b>	<b>5</b>
<b>4</b>	<b>Драйвер контроллера дисплея <i>vrout-drm</i></b>	<b>6</b>
<b>5</b>	<b>Драйвер фреймбуфера <i>vroutfb</i></b>	<b>7</b>
<b>6</b>	<b>Драйвер VPU <i>avico</i></b>	<b>9</b>
<b>7</b>	<b>Драйвер контроллера Ethernet <i>arasan-gemac</i></b>	<b>11</b>
<b>8</b>	<b>Драйвер управления частотой ядер CPU <i>cpufreq-dt</i></b>	<b>12</b>
<b>9</b>	<b>Механизм CPU hotplug</b>	<b>13</b>

## 1. О ДОКУМЕНТЕ

Документ содержит описание основных подсистем и драйверов ядра Linux, реализованных для поддержки аппаратуры СнК 1892ВМ14Я и модулей на базе СнК.

Ядро Linux поддерживает модули следующих ревизий:

- Салют-ЭЛ24Д1 r1.3;
- Салют-ЭЛ24Д1 r1.4;
- Салют-ЭЛ24Д1 r1.5;
- Салют-ЭЛ24Д2 r1.1;
- Салют-ЭЛ24ОМ1 r1.1 с установленным Салют-ЭЛ24ПМ1 r1.1.

Файлы DTS \*.dtsi, \*.dts расположены в дереве исходных кодов U-Boot arch/arm/dts/\*.dts\*. Пути до прочих файлов приведены относительно корня дерева исходных кодов Linux.

## 2. ПОДСИСТЕМА УПРАВЛЕНИЯ ТАКТОВЫМИ СИГНАЛАМИ

Управление тактовыми сигналами и частотами в ядре Linux реализовано с использованием [Common Clock Framework](#).<sup>1</sup> Тактовые сигналы микросхемы описаны в виде дерева в файле `mcom.dtsi`. Для управления тактовыми сигналами и частотами используются 4 драйвера, описанные в `drivers/clockdev/clk-elvees/clk-mcom.c`:

- `mcom-clk-gate`;
- `mcom-clk-divider`;
- `mcom-clk-pll`;
- `mcom-smctr`.

Для корректного управления тактовыми сигналами каждый драйвер устройства, входящий в состав ядра Linux, должен реализовывать:

1. При инициализации драйвера:
  1. Захват необходимого для устройства тактового сигнала, используя функцию `clk_get()`.
  2. Включение тактового сигнала, используя функцию `clk_enable()`.
2. При удалении драйвера:
  1. Выключение тактового сигнала, используя функцию `clk_disable()`.

При инициализации подсистемы управления тактовыми сигналами происходит начальная настройка всех PLL и делителей частот микросхемы. Устанавливаемые при инициализации значения множителей PLL и делителей частот описаны в файле `mcom.dtsi`.

---

<sup>1</sup> <https://www.kernel.org/doc/Documentation/clockdev.txt>

### 3. ДРАЙВЕР КОНТРОЛЛЕРА PWM PWM-MCOM

Драйвер *pwm-mcom* управляет контроллером PWM 1892BM14Я. Драйвер реализует стандартный интерфейс PWM<sup>2</sup>

Файл с исходным кодом драйвера — `drivers/pwm/pwm-mcom.c`. Описание DTS bindings представлено в файле `Documentation/devicetree/bindings/pwm/elvees,mcom-pwm.txt`.

Ограничения драйвера:

1. Не реализовано управление каналами OUTB.
2. Не поддерживается режим счёта PWM-контроллера (PWM API не поддерживает данный режим).
3. Не реализовано управление предделителем.

---

<sup>2</sup> <https://www.kernel.org/doc/Documentation/pwm.txt>

## 4. ДРАЙВЕР КОНТРОЛЛЕРА ДИСПЛЕЯ VPOUT-DRM

Данный раздел применим к драйверу контроллера дисплея VPOUT СнК 1892ВМ14Я для подсистемы DRM — *vpout-drm*.

Документация, описывающая текущую версию подсистемы DRM, доступна по ссылке [Linux GPU Driver Developer's Guide](#)<sup>3</sup>.

Исходный код драйвера содержится в директории `drivers/gpu/drm/vpout`.

Драйвер обеспечивает следующие возможности:

1. Разрешение экрана до 1920x1080 пикселей;
2. Поддержка внешнего HDMI передатчика NXP TDA998x;
3. Поддержка внешних панелей с заданием параметров дисплея через DTS;
4. Чтение Extended Display Identification Data (EDID);
5. Эмуляция фреймбуфера через устройство `/dev/fb0`.

Ограничения драйвера:

1. Не поддерживаются чересстрочные видеорежимы (не поддерживаются контроллером дисплея VPOUT);
2. Не поддерживаются HDMI передатчики отличные от NXP TDA998x;
3. Не реализована поддержка абстракции плоскостей (plane abstraction);
4. Не поддерживается атомарная установка видеорежима.

При использовании в качестве устройства вывода HDMI монитора драйвер устанавливает оптимальный для подключенного монитора видеорежим, определяемый по EDID. С помощью параметров ядра (kernel parameters) возможно установить фиксированный видеорежим. Например, следующая строка задает разрешение экрана в 1280×720 пикселей:

```
video=HDMI-A-1:1280x720
```

Подробное описание параметров ядра, задающих видеорежим, содержится в документе `Documentation/fb/modedb.txt`.

<sup>3</sup> <https://www.kernel.org/doc/html/latest/gpu/index.html>

## 5. ДРАЙВЕР ФРЕЙМБУФЕРА VPOUTFB

Для вывода на экран графического окружения на СнК используется подсистема `FBDev`<sup>4</sup> и драйвер `vputfb`. Директория с исходным кодом драйвера — `drivers/video/fbdev/vputfb`. Драйвер управляет контроллером VPOUT и HDMI-адаптером ITE IT66121.

Алгоритм работы драйвера:

1. Если в DTS в узле `output` присутствует свойство `compatible="ite,it66121"`, то выполнить настройку контроллера ITE IT66121, подключенного по I2C.
2. Считать из DTS видеорежим и настроить VPOUT для вывода в заданном видеорежиме.
3. Если в DTS отсутствует видеорежим или тайминги некорректны, или свойство `output` отсутствует, то настроить VPOUT для вывода в режиме 720p 60 FPS.

Вызов `ioctl FBIOPUT_VSCREENINFO` с заданием неподдерживаемого режима завершается с `-EINVAL`. (Следовательно, вызов `fbset` завершится с ненулевым кодом возврата).

Поддерживаются следующие `ioctl`:

- `FBIOGET_VSCREENINFO`;
- `FBIOPUT_VSCREENINFO`;
- `FBIOGET_FSCREENINFO`;
- `FBIOGETCMAP`;
- `FBIOPUTCMAP`;
- `FBIOBLANK`;
- `VPOUTFB_GET_MEMORY_ID`.

При появлении прерывания `OUT_FIFO_INT` блока VPOUT драйвер останавливает и переинициализирует VPOUT. При этом в `dmesg` печатается сообщение “`Caught OUT_FIFO_INT, reinitializing VPOUT`”.

В драйвере не реализовано:

1. Чтение EDID HDMI-монитора и ограничение возможных разрешений согласно данным из EDID.
2. Остановка/запуск VPOUT при отключении/подключении HDMI-монитора.

---

**Примечание:** Т.к. автоматическое определение подключения HDMI-монитора отсутствует, драйвер может быть выключен по умолчанию. Загрузка драйвера выполняется командой `modprobe vputfb`.

---

<sup>4</sup> <https://www.kernel.org/doc/Documentation/fb/api.txt>

---

**Примечание:** Для управления видеорежимами может использоваться утилита `fbset` и файл `fb.modes`.

---

Драйвер считывает видеорежим из DTS в соответствии с описанием в `Documentation/devicetree/bindings/video/display-timing.txt`. В DTS-файле `mcom.dtsi` описан формат цветowych компонентов изображения, устанавливаемые при инициализации драйвера. Подробное описание полей узла устройства VPOUT представлено в файле `Documentation/devicetree/bindings/fb/vpoutfb.txt`.



## 6. ДРАЙВЕР VPU AVICO

Драйвер *avico* управляет VPU VELcore-01 и реализует аппаратное сжатие видео по стандарту H.264. Драйвер реализован с использованием подсистемы V4L2<sup>5</sup> и предоставляет стандартный программный интерфейс для сжатия и управления.

Возможности драйвера:

1. Поддерживаются входные кадры в формате M420<sup>6</sup>.
2. Максимальная ширина кадра — 1920 пикселей.
3. Максимальная высота кадра — 4096 пикселей.
4. Возможность установки FPS видеопотока.

Ограничения драйвера:

1. Поддерживается только сжатие видео.
2. Поддерживается только один поток.
3. Требуется нестандартный формат пикселей на входе (M420).
4. Ширина и высота кадра должны быть кратны 16 пикселям.
5. Требуется 180 КиБ памяти XYRAM (работа совместно с DSP не тестировалась).
6. Нет возможности управления качеством сжатия.
7. Нет возможности сжатия с постоянный битрейтом.
8. Нет возможности менять FPS в процессе кодирования.

Для обхода проблемы `gf#1382` драйвер использует промежуточные буферы в XYRAM для восстановленных и сжатых данных. Всего используется 4 буфера по 45 КиБ (строка макроблоков для кадра шириной 1920 пикселей в формате M420) — 2 буфера для восстановленных данных и 2 для сжатых. В результате реализации обхода проблемы, максимальная ширина кадров ограничилась 1920 пикселями.

После каждой строки макроблоков VPU останавливается и драйвер выполняет следующие действия:

1. Настраивает VPU на другой промежуточный буфер.
2. Запускает SDMA для копирования данных из промежуточного буфера в DDR.
3. Запускает VPU на обработку следующей строки макроблоков.

Для обхода проблемы `gf#2003` в обработчике прерывания используется задержка, состоящая из следующих действий:

1. Ожидание завершения чтения очередных данных исходного и референсного кадров.
2. Ожидание завершения 80-кратного чтения регистра EVENTS.

<sup>5</sup> <https://linuxtv.org/downloads/v4l-dvb-apis/v4l2spec.html>

<sup>6</sup> <https://linuxtv.org/downloads/v4l-dvb-apis/V4L2-PIX-FMT-M420.html>

- 
3. Ожидание снятия всех флагов регистра EVENTS, указывающих на текущую работу VDMA.

## 7. ДРАЙВЕР КОНТРОЛЛЕРА ETHERNET ARASAN-GEMAC

Драйвер *arasan-gemac* управляет контроллером Ethernet Arasan GEMAC. Драйвер реализует стандартный интерфейс `network devices`, описанный в `Documentation/networking/netdevices.txt`. Обработка RX-прерываний реализована с использованием интерфейса `NAPI`<sup>7</sup>.

Директория с исходным кодом драйвера — `drivers/net/ethernet/arasan`.

Драйвер поддерживает выполнение следующих операций из пространства пользователя:

1. Установка скорости (10/100/1000 МБит/с);
2. Установка дуплекса (full/half);
3. Установка уровня сообщений драйвера;
4. Установка MAC-адреса;
5. Перезапуск автосогласования;
6. Проверка физического подключения.

Драйвер не поддерживает:

1. Управление паузой;
2. Чтение и запись EEPROM;
3. Wake-on-Lan;
4. Управление объединением прерываний.

---

<sup>7</sup> <https://wiki.linuxfoundation.org/networking/napi>

## 8. ДРАЙВЕР УПРАВЛЕНИЯ ЧАСТОТОЙ ЯДЕР CPU *CPUFREQ-DT*

Штатный драйвер *cpufreq-dt*, позволяет управлять тактовой частотой ядер CPU0 и CPU1 через подсистему *CPUfreq*<sup>8</sup>.

Директория с исходным кодом драйвера — `drivers/cpufreq`. Список частот ядер CPU описан в DTS-файле `mcom.dtsi`. Описание DTS bindings представлено в файле `Documentation/devicetree/bindings/cpufreq/cpufreq-dt.txt`.

Возможности драйвера:

1. Регуляторы масштабирования тактовой частоты ядер CPU (*CPUfreq governors*):
  - `ondemand` (по-умолчанию) — устанавливает тактовую частоту в зависимости от нагрузки на ядрах CPU;
  - `conservative` — похож на `ondemand`, но более экономный (предпочтение отдаётся меньшим тактовым частотам);
  - `performance` — устанавливает тактовую частоту в максимальное значение;
  - `userspace` — позволяет устанавливать частоту из пространства пользователя.
2. Управление регуляторами и частотами через `sysfs`.

Ограничения драйвера:

1. Не поддерживается управление напряжением питания ядер CPU, т.к отсутствует поддержка в SnK;
2. Не поддерживается независимое управление частотой ядер CPU, т.к отсутствует поддержка в SnK.

Для установки тактовой частоты ядер из пространства пользователя необходимо:

1. Выбрать регулятор `userspace`:

```
echo userspace > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
```

2. Выбрать частоту из поддерживаемых. Список частот доступен в файле `/sys/devices/system/cpu/cpu0/cpufreq/scaling_available_frequencies`.
3. Установить частоту. Значение частоты передаётся в кГц, например:

```
echo 312000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_setspeed
```

<sup>8</sup> <https://www.kernel.org/doc/Documentation/cpu-freq/user-guide.txt>

## 9. МЕХАНИЗМ CPU HOTPLUG

Механизм `CPU hotplug`<sup>9</sup> позволяет включать и выключать процессорные ядра, не перезагружая систему, что может использоваться:

1. Для удаления отдельных незагрузочных процессорных ядер и устранения в них неисправностей;
2. Для перехода системы в ждущий режим.

Для выключения и включения процессорных ядер используются функции `cpu_down()` и `cpu_up()`, описанные в файле `kernel/cpu.c`.

Использование через `sysfs`:

1. Для отключения питания ядра CPU1 необходимо записать 0 в файл `/sys/devices/system/cpu/cpu1/online`:

```
echo 0 > /sys/devices/system/cpu/cpu1/online
smpboot: CPU 1 is now offline
```

2. Для включения питания ядра CPU1 необходимо записать 1 в файл `/sys/devices/system/cpu/cpu1/online`:

```
echo 1 > /sys/devices/system/cpu/cpu1/online
smpboot: Booting Node 0 Processor 1 APIC 0x1
```

<sup>9</sup> [https://www.kernel.org/doc/Documentation/core-api/cpu\\_hotplug.rst](https://www.kernel.org/doc/Documentation/core-api/cpu_hotplug.rst)