
ЯДРО LINUX ДЛЯ 1892ВМ14Я. РУКОВОДСТВО СИСТЕМНОГО ПРОГРАММИСТА

**Версия v2.4
10.08.2017**

ОГЛАВЛЕНИЕ

1	О документе	3
2	Подсистема управления тактовыми сигналами	4
3	Драйвер контроллера PWM <i>pwm-tcom</i>	5
4	Драйвер фреймбуфера <i>vroutfb</i>	6
5	Драйвер VPU <i>avico</i>	8

1. О ДОКУМЕНТЕ

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

Ядро Linux поддерживает отладочные 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](#).¹ Тактовые сигналы микросхемы описаны в виде дерева в файле `mcom.dtsi`. Для управления тактовыми сигналами и частотами используются 4 драйвера, описанные в `drivers/clockdev/elvees/clock-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`.

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

3. ДРАЙВЕР КОНТРОЛЛЕРА PWM PWM-MCOM

Драйвер *pwm-mcom* управляет контроллером PWM 1892BM14Я. Драйвер реализует стандартный интерфейс PWM²

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.

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

4. ДРАЙВЕР ФРЕЙМБУФЕРА VPOUTFB

Для вывода на экран графического окружения на СнК используется подсистема `FBDev`³ 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`.

³ <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`.

5. ДРАЙВЕР VPU AVICO

Драйвер *avico* управляет VPU VELcore-01 и реализует аппаратное сжатие видео по стандарту H.264. Драйвер реализован с использованием подсистемы V4L2⁴ и предоставляет стандартный программный интерфейс для сжатия и управления.

Возможности драйвера:

1. Поддерживаются входные кадры в формате M420⁵.
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.

⁴ <https://linuxtv.org/downloads/v4l-dvb-apis/v4l2spec.html>

⁵ <https://linuxtv.org/downloads/v4l-dvb-apis/V4L2-PIX-FMT-M420.html>

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