
ELVEES OPENVX SDK ДЛ 1892ВМ14Я. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

**Версия v2.9
11.10.2018**

ОГЛАВЛЕНИЕ

1	О документе	3
2	Описание ELVEES OpenVX SDK	4
3	Запуск демонстрационных примеров	5
3.1	Пример запуска демонстрационного примера	6
4	Инструкция по разработке приложений на базе ELVEES OpenVX SDK	8

1. О ДОКУМЕНТЕ

Документ описывает SDK, предназначенный для разработки приложений на основе стандарта OpenVX версии 1.1.

2. ОПИСАНИЕ ELVEES OPENVX SDK

SDK предоставляет реализацию OpenVX под СнК 1892ВМ14Я. Реализация оптимизирована для параллельных вычислений на DSP-ядрах.

Состав SDK:

1. Библиотека, реализующая стандарт OpenVX. Бинарные артефакты библиотеки OpenVX распространяются в образе SD-карты Buildroot Linux для 1892ВМ14Я версии 2.9 и выше.
2. Заголовочные файлы OpenVX версии 1.1¹.
3. Дополнительные заголовочные файлы для получения расширенной статистики использования DSP-ядер.
4. Исходные коды демонстрационных примеров `openvx_demo`, а также их бинарные артефакты в составе образа SD-карты.

Полный набор пакетов Buildroot, из которых состоит SDK, перечислен в таблице 2.1.

Таблица 2.1. Артефакты библиотеки OpenVX

Пакет	Описание	Лицензия
<code>openvx-headers</code>	Заголовочные файлы стандарта OpenVX версии 1.1	MIT
<code>openvx</code>	Библиотека, предоставляющая функционал OpenVX	–
<code>openvx-demo</code>	Демонстрационные примеры OpenVX	–

¹ <https://www.khronos.org/registry/OpenVX/api/1.1/openvx-standard-headers-1.1.tar.bz2>

3. ЗАПУСК ДЕМОСТРАЦИОННЫХ ПРИМЕРОВ

Полный список демонстрационных примеров представлен в таблице 3.1. Для работы с изображениями используется формат PNG, с видео — YUV4MPEG2.

Таблица 3.1. Список демонстрационных примеров

Артефакт	Описание	Параметры командной строки
sampleCannyEdgeDetector	Детектор границ Канни	входное изображение, выходное изображение
sampleColorCorrection	Коррекция цвета изображения	входное изображение, выходное изображение
sampleFastCorners	Детектор углов	входное изображение, выходное изображение
sampleGaussianFiltering	Размытие Гауссовым окном размера 3x3	входное изображение, выходное изображение
sampleGaussianPyramid	Построение 4-х уровневой пирамиды изображений	входное изображение, 4 выходных изображения
sampleHarrisCorners	Детектор углов Харриса	входное изображение, выходное изображение
sampleMaskMake	Комбинированный алгоритм: гауссово размытие окном 3x3, пороговая фильтрация, эрозия	входное изображение, выходное изображение
sampleMedianFiltering	Медианная фильтрация окном 3x3	входное изображение, выходное изображение
sampleMinMaxLoc	Определение позиций максимального и минимального значений на изображении	входное изображение, выходное изображение
sampleMorphology	Морфологические алгоритмы, размер окна 3x3	входное изображение, выходное изображение эрозии, выходное изображение дилатации
sampleNoiseSuppression	Подавление шума с помощью аккумулялирования серии изображений	первое входное изображение серии, выходное изображение
sampleObjectCut	Вырезание объекта по порогу	входное изображение, выходное изображение
sampleOpticalFlowPyrLK	Определение оптического потока	входное изображение, сгенерированное смещённое изображение, выходное изображение
sampleSharpnessFiltering	Увеличение резкости, используется лапласиан 3x3	входное изображение, выходное разностное изображение, выходное изображение
sampleTemplate	Пример чтения/записи изображений	входное изображение, выходное изображение
sampleTranspose	Поворот на 90 градусов и горизонтальное отражение	входное изображение, выходное изображение
sampleVideo	Пример чтения/записи видео файла	входное видео, выходное видео

Продолжается на следующей странице

Таблица 3.1 – продолжение с предыдущей страницы

Артефакт	Описание	Параметры командной строки
sampleVideoCanny	Детектор границ Канны	входное видео, выходное видео
sampleVideoColorCorrection	Коррекция цвета видео	входное видео, выходное видео
sampleVideoFastCorners	Детектор углов на видео	входное видео, выходное видео
sampleVideoGaussianFiltering	Размытие Гауссовым окном размера 3x3	входное видео, выходное видео
sampleVideoMedian	Медианная фильтрация окном 3x3	входное видео, выходное видео
sampleVideoMinMaxLoc	Определение позиций максимального и минимального значений на видео	входное видео, выходное видео
sampleVideoMorphology	Морфологический алгоритм (эрозия), размер окна 3x3	входное видео, выходное видео
sampleVideoObjectCut	Вырезание объекта по порогу	входное видео, выходное видео
sampleVideoTranspose	Поворот на 90 градусов и горизонтальное отражение видео	входное видео, выходное видео

3.1 Пример запуска демонстрационного примера

1. Подать питание на модуль, дождаться загрузки ОС и подключиться по терминалу к модулю (подробнее см. Запуск модуля из Руководства системного программиста).
2. Сгенерировать изображение разрешением 1280x720:

```
ffmpeg -f lavfi -i "testsrc=size=1280x720" -frames 1 image.png
```

3. Запустить пример sampleCannyEdgeDetector:

```
sampleCannyEdgeDetector image.png out.png
```

Примерный вывод программы:

```
Read from "image.png"
Processing CannyEdgeDetector...
Write to "out.png"
Frames counter 1, time spent 242329, avg 242329
DMA: copy: 14909632, timing: 151599
DSP: total_count: 171, total_ticks: 17105995, dsp0: count: 163, ticks:
↪16761623, timings: 147376, dsp1: count: 8, ticks: 16762749, timings: 147828
Done!
```

Файл out.png будет содержать обработанное изображение.

4. Сгенерировать 10-ти секундное видео разрешением 640x480:

```
ffmpeg -f lavfi -i "life=s=640x480:mold=10:ratio=0.1:death_color=
↪#C83232:life_color=#00ff00" -pix_fmt yuv420p -t 10 life.y4m
```

5. Запустить пример sampleVideoCanny:

```
sampleVideoCanny life.y4m out.y4m
```

Вывод будет следующим:

```
Processing video with CannyEdgeDetector...
Frames counter 250, time spent 17090762, avg 68363
DMA: copy: 1235248000, timing: 10107926
DSP: total_count: 14000, total_ticks: 1402948579, dsp0: count: 12000, ticks:
↪1361500497, timings: 9348930, dsp1: count: 2000, ticks: 1319805206,
↪timings: 9139613
Done!
```

Файл out.y4m будет содержать результат обработки видео.

Все демонстрационные примеры выводят некоторую вспомогательную статистику запуска: потраченное время, количество запусков каждого ядра, количество обработанных кадров и др.

4. ИНСТРУКЦИЯ ПО РАЗРАБОТКЕ ПРИЛОЖЕНИЙ НА БАЗЕ ELVEES OPENVX SDK

Инструкция предназначена для разработки приложений, использующих API стандарта OpenVX версии 1.1.

1. Создать пакет Buildroot², см., например, `openvx-demo`.
2. Подключить заголовочные файлы и указать зависимости в скриптах сборки, см., например, `openvx-demo`.
3. Пересобрать Buildroot и образ SD-карты.
4. Проверить работоспособность приложения, см., например, предыдущий раздел.

² <https://buildroot.org/downloads/manual/manual.html#adding-packages>