

ELVEES OPENVX SDK ДЛҀА 1892ВМ14Я. РУКОВОДСТВО ПРОГРАММИСТА

**Версия
11.03.2022**

ОГЛАВЛЕНИЕ

1	О документе	3
2	Описание ELVEES OpenVX SDK	4
3	Запуск тестов OpenVX CTS	5
3.1	Пример запуска тестов	5
4	Запуск демонстрационных примеров	6
4.1	Пример запуска демонстрационного примера	7
5	Инструкция по разработке приложений на базе ELVEES OpenVX SDK	9

1. О ДОКУМЕНТЕ

Документ описывает SDK, предназначенный для разработки приложений на основе стандарта OpenVX версии 1.3.

2. ОПИСАНИЕ ELVEES OPENVX SDK

SDK предоставляет реализацию OpenVX под СнК 1892ВМ14Я. Ключевые особенности реализации:

- Соответствует стандарту OpenVX версии 1.3.
- Поддерживает функциональные наборы²:
 - базовый для организации графовой структуры;
 - Vision, совместимый со стандартом OpenVX версии 1.1;
 - Neural Network, совместимый со стандартом OpenVX версии 1.2.
- Проверка соответствия стандарту (OpenVX Conformance Test Suite).
- Содержит демонстрационные примеры с открытым исходным кодом.
- Имеет более высокую производительность по сравнению с эталонной реализацией от KhronosGroup³.

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 1.3¹.
3. Исходные коды демонстрационных примеров `openvx_demo`, а также их бинарные артефакты в составе образа SD-карты.
4. Тесты соответствия стандарту OpenVX Conformance Test Suite (OpenVX CTS) 1.3.

Полный набор пакетов Buildroot, из которых состоит SDK, перечислен в таблице 2.

Таблица 2.1. Артефакты библиотеки OpenVX

Пакет	Описание	Лицензия
<code>openvx-headers</code>	Заголовочные файлы стандарта OpenVX версии 1.3	MIT
<code>openvx</code>	Библиотека, предоставляющая функционал OpenVX	–
<code>openvx-demo</code>	Демонстрационные примеры OpenVX	–
<code>openvx-cts</code>	Тесты OpenVX CTS версии 1.3	Apache License 2.0

² https://www.khronos.org/registry/OpenVX/specs/1.3/vx_khr_feature_sets/1.1/html/vx_khr_feature_sets_1_1.html

³ <https://github.com/KhronosGroup/OpenVX-sample-impl>

¹ <https://www.khronos.org/registry/OpenVX/api/1.3/openvx-standard-headers-1.3.tar.bz2>

3. ЗАПУСК ТЕСТОВ OPENVX CTS

Запуск тестов OpenVX CTS осуществляется с помощью исполняемого файла `vx_test_conformance`. Для тестов используются тестовые данные из директории `/usr/share/cts/test_data` и библиотеки OpenVX.

3.1 Пример запуска тестов

1. Подать питание на модуль, дождаться загрузки ОС и подключиться по терминалу к модулю (подробнее см. Запуск модуля из Руководства программиста).
2. Запустить исполняемый файл:

```
vx_test_conformance
```

3. Дождаться окончания прохождения тестирования (около 30 минут).
4. Убедиться, что отчёт прохождения тестов не содержит ошибок и его окончание выглядит следующим образом:

```
[ ===== ]
[ ALL DONE ] 7042 test(s) from 75 test case(s) ran
[ PASSED   ] 7042 test(s)
[ FAILED   ] 0 test(s)
[ DISABLED ] 8190 test(s)

=====
OpenVX Conformance report summary
=====

To be conformant to the OpenVX baseline, 5862 required test(s) must pass.
↪5862 tests passed, 0 tests failed. PASSED.
To be conformant to the Neural Network extension, 1180 required test(s) must
↪pass. 1180 tests passed, 0 tests failed. PASSED.
To be conformant to the Vision conformance profile, 6167 required test(s)
↪must pass. 6167 tests passed, 0 tests failed. PASSED.
To be conformant to the Neural Networks conformance profile, 1180 required
↪test(s) must pass. 1180 tests passed, 0 tests failed. PASSED.
Note: The 8190 disabled tests are optional and are not considered for
↪conformance.

#REPORT: 20200922123136 ALL 16499 8190 7042 7042 7042 0 (version 1.3)
```

4. ЗАПУСК ДЕМОСТРАЦИОННЫХ ПРИМЕРОВ

Полный список демонстрационных примеров представлен в таблице 4.1. Для работы с изображениями используется формат PNG, с видео — YUV4MPEG2.

Таблица 4.1. Список демонстрационных примеров

Артефакт	Описание	Параметры командной строки
sampleCannyEdgeDetector	Детектор границ Канны	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 4.1 – продолжение с предыдущей страницы

Артефакт	Описание	Параметры командной строки
sampleVideoCanny	Детектор границ Канни	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

4.1 Пример запуска демонстрационного примера

1. Подать питание на модуль, дождаться загрузки ОС и подключиться по терминалу к модулю (подробнее см. Запуск модуля из Руководства 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 1057113088, avg 1057113088
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 34454470912, avg 137817883  
Done!
```

Файл `out.y4m` будет содержать результат обработки видео.

Все демонстрационные примеры выводят некоторую вспомогательную статистику запуска: количество обработанных кадров, потраченное время в наносекундах, среднее время обработки одного кадра.

5. ИНСТРУКЦИЯ ПО РАЗРАБОТКЕ ПРИЛОЖЕНИЙ НА БАЗЕ ELVEES OPENVX SDK

Инструкция предназначена для разработки приложений, использующих API стандарта OpenVX версии 1.3.

1. Создать пакет Buildroot⁴, 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 библиотеки и приложений, см., например, два предыдущих раздела.

⁴ <https://buildroot.org/downloads/manual/manual.html#adding-packages>