



Программное обеспечение видеоаналитики "СмартКросс"

Руководство администратора по установке

АННОТАЦИЯ

В данном документе приведено руководство по установке Программного обеспечения видеоаналитики "СмартКросс" (далее – Модуль, СПО, Система) в составе Интеллектуальной транспортной системы, предназначенной для организации управления дорожным движением.

1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

1.1. Назначение и функции программы

Система предназначена для обеспечения мониторинга и оценки параметров транспортного потока на перекрестках.

В Системе реализован следующий функционал:

- построение и отображение направлений перемещения транспортных средств на перекрестке;
- отображение статистики по направлениям в виде круговой диаграммы;
- отображение статистики транспортный средств между рубежами контроля (створами) перекрестка в виде мнемосхемы;
- отображение цифры количества проездов на каждом направлении;
- фильтрация по параметрам;
- возможность привязки направлений к графу дорожной сети;
- выгрузка данных в файл;
- хранение фото- и видеоданных;
- возможность выгрузки индивидуальных проездов транспортных средств;
- предоставление диагностической информации о состоянии детектора транспорта.

1.2. Сведения о технических и программных средствах

Система может функционировать как в виртуальной среде, так и на физической аппаратной платформе. В качестве системного ПО используются операционные системы на основе ядра Linux, версии не ниже 2.6.

1.3. Требования к персоналу

Установка СПО должно производиться квалифицированным персоналом, имеющим профессиональные знания в области работы с локальными вычислительными сетями, сетями передачи данных и системами управления базами данных.

Администратор должен иметь опыт работы и представление о типах локальных сетей, об организации адресации в локальных сетях и в сетях передачи данных, знать основные понятия и термины, применяемые в цифровых способах передачи данных.

Администратор должен знать:

- основы обеспечения информационной безопасности, правила и способы организации доступа к информации, правила разграничения доступа с использованием штатных средств, предоставляемых средствами вычислительной техники или автоматизированными системами;

- абстрактное (формализованное или неформализованное) описание комплекса программно-технических средств и (или) организационных мер защиты от несанкционированного доступа;

- состояние защищенности информации, обрабатываемой средствами вычислительной техники или автоматизированной системы, от внутренних или внешних угроз;

- способности средств вычислительной техники или автоматизированной системы обеспечивать неизменность информации в условиях случайного и (или) преднамеренного искажения (разрушения);

- характеристики средств вычислительной техники, влияющие на защищенность по уровню, глубине в зависимости от класса защищенности средств вычислительной техники;

- комплекс организационных мер и программно-технических (в том числе криптографических) средств обеспечения безопасности информации в автоматизированных системах;

- основные функции работы активного сетевого оборудования, вспомогательных систем и агрегатов, отличия работы активного сетевого оборудования различных производителей;

- особенности эксплуатации операционных систем с открытым кодом на основе семейства архитектуры Linux, особенности пользовательских компонентов, оболочек, инструментов, сред для разработки;

- основные команды по работе с ядром и файловой средой операционной системы, назначение команд, интерфейс командной строки, особенности дистрибутивов операционной системы;

- основные приемы и методы программирования, уметь разбираться в коде программы, в средах разработки и программирования, понимать методы межпрограммного взаимодействия;

- принципы построены протоколов передачи данных по интернет, типы протоколов, различия в их версиях;

- стандартный стек протоколов (сетевая модель OSI), особенности их структуры и взаимодействия с верхним и нижним уровнем;

- взаимодействие приложений, включая понимание работы API, включая понимание сигнатур и семантики функций.

Администратор должен иметь представление об этапах разработки программного обеспечения, особенностях эксплуатации программного изделия на различных его этапах.

Администратор должен уметь оценивать основные эксплуатационные параметры сети передачи данных, анализировать производительности основных типов оборудования.

Администратор должен знать основные методы проведения эксплуатационных мероприятий, технического обслуживания, должен уметь спрогнозировать возможное развитие ситуации при реакции Системы на свои действия.

2. УСТАНОВКА ПРОГРАММЫ

Установка и запуск СПО осуществляется с правами администратора.

2.1. Установка

Весь процесс инсталляции отслеживается стандартными средствами операционной системы и сопровождается понятными и интерактивными сообщениями.

В настоящем руководстве принято следующее – исходной директорией является директория на носителе данных, поставляемом для установки. Порядок установки указанных в настоящем руководстве компонент описан ниже (Таблица

Таблица 1 – Действия администратора

Действие	Описание
Скачайте содержимое исходной директории на локальный диск	
Установите зависимости (ОС - желательно ubuntu 20): – D	«Установка зависимостей» означает установку программного обеспечения или пакетов, от которых зависит работа приложения или системы на операционной системе Ubuntu версии 20. - открытая платформа для автоматизации развертывания, масштабирования и управления приложениями в контейнерах. Для установки Docker на Ubuntu 20 необходимо использовать команду `sudo apt-get install docker.io`
Проставьте нужные адреса в файле .env. Предупреждение - нельзя использовать адрес localhost.	См. пример ниже.
Добавьте docker образы в систему: s u d	Операция `sudo docker image load -i 'Умный перекресток/SmartIntersection '` загружает Docker-образ из файла в формате `.tar`.

Действие	Описание
	<p><code>`sudo`</code> - команда для выполнения операции с привилегиями суперпользователя (<code>root</code>).</p> <p><code>`docker image load`</code> - команда для загрузки Docker-образа.</p> <p>Умный перекресток/<code>SmartIntersection`</code> - опция, указывающая на входной файл с Docker-образом в формате <code>`tar`</code>. В данном случае</p> <p>Таким образом, данная операция загружает Docker-образ с именем <code>`Умный перекресток/SmartIntersection`</code> из файла в формате <code>`tar`</code>.</p>
<p>Настройте параметры конфигурации в файле <code>docker-compose.yml</code> (не обязательно)</p> <p><code>services.*.logging.loggers.level</code> - уровень логирования (<code>DEBUG</code>, <code>INFO</code>, <code>WARNING</code>, <code>ERROR</code>)</p>	<p>Операция настройки параметров конфигурации в файле <code>`docker-изменения значений соответствующих полей в разделе `services`</code>.</p> <p>Пример настройки параметра</p> <pre>''' services: myservice: logging: loggers: level: DEBUG</pre> <p><code>services`</code> - раздел, содержащий список сервисов, для которых нужно настроить параметры;</p> <p><code>myservice`</code> - имя сервиса, для которого нужно настроить параметры (может быть любым);</p> <p><code>logging`</code> - раздел, содержащий настройки логирования для данного сервиса;</p> <p><code>loggers`</code> - раздел, содержащий настройки логгеров для данного сервиса;</p>

Действие	Описание
	<p>level` - поле, содержащее уровень логирования. Здесь указывается одно из значений: `DEBUG`, `INFO`,</p> <p>В данном примере параметр `level` установлен на значение `DEBUG`, что означает, что будут записываться все доступные сообщения логирования.</p>
<p>Запустите СПО</p> <p>sudo docker compose up -d</p>	<p>Операция `sudo docker-compose up -d` используется для запуска контейнеров, описанных в файле `docker-compose.yml`, в фоновом режиме:</p> <p>sudo` - ключ, указывающий на выполнение команды с привилегиями суперпользователя (root);</p> <p>d</p> <p>o</p> <p>c</p> <p>up` - подкоманда, используемая для создания и запуска контейнеров, описанных в файле `docker-compose.yml`;</p> <p>compose` - ключ, указывающий на запуск Docker-сервера в фоновом режиме (detach) файла `docker-compose.yml`;</p> <p>При выполнении этой команды Docker будет проходить по файлу `docker-контейнеры` и запускать их. Контейнеры будут работать в фоновом режиме, и вывод команды будет показывать только идентификаторы контейнеров.</p> <p>Контейнеры запускаются, как сервисы, работающие в фоновом режиме без вывода на экран логов и информации об их работе.</p>
<p>Зайдите через веб браузер по IP адресу или через dns имя того сервера, на котором разворачивается Система (порт 80, либо без порта)</p>	

3. ПРОВЕРКА ПРОГРАММЫ

Программное обеспечение работоспособно, если в результате действий пользователя, изложенных в разделе 2, системой осуществляется мониторинг и оценка параметров транспортного потока на перекрестках.