

родственные структуры: синтез, производство и применение: сб. статей. Тамбов, 2019. – С. 51-52.

3. Практикум по ионному обмену / В.М. Селеменев, Г.В. Славинская, В.Ю. Хохлов и др. – Воронеж: Воронежский государственный университет, 2014. – 160 с.

© Балашова Е.А.

УДК 621.397.424.2

Алексеев Денис Андреевич  
Alekseev Denis Andreevich  
Горбачев Иван Алексеевич  
Gorbachev Ivan Alexeyevich,  
Дубикайтис Анастасия Петровна  
Dubikaitis Anastasia Petrovna

Студент  
Student

Санкт-Петербургский политехнический университет Петра Великого  
Peter the Great St. Petersburg Polytechnic University

## СИСТЕМА ВИДЕОНАБЛЮДЕНИЯ НА ОСНОВЕ RASPBERRYPI

## VIDEO SURVEILLANCE SYSTEM BASED ON THE RASPBERRY PI

**Аннотация:** В статье изложена реализация прототипа домашней системы видеонаблюдения на основе Raspberry Pi. Продемонстрирована программная реализация с описанием каждого проделанного шага. Разработан скрипт на языке Python для интеграции в систему.

**Abstract:** The article describes the implementation of a prototype home video surveillance system based on the Raspberry Pi. The software implementation was demonstrated with a description of each step taken. A Python script was developed for integration into the system.

**Ключевые слова:** система видеонаблюдения, raspberry pi, motioneye, pushover.

**Key words:** video surveillance system, raspberry pi, motioneye, pushover.

Данная работа посвящена реализации прототипа домашней системы видеонаблюдения на основе одноплатного компьютера Raspberry Pi.

Рассчитаем нагрузку на сеть. В связи с тем, что в реализации проекта необходимо настраивать разрешение изображения, а также максимальное значение кадров в секунду, то нужно посчитать нагрузку на сеть с выбранными значениями: разрешение установлено 1024x768, а количество кадров в секунду 10.

В Motion используется формат сжатия h.264, а значит коэффициент сжатия будет равен 65,854. Также, стоит уточнить вес одного пикселя, учитывая, что RGB – это 3 цвета, следовательно 3 параметра по 256 оттенков, получается, что для кодирования пикселя используется 24 бита.

Формула расчета нагрузки на беспроводную сеть:

$$q = V_k \cdot n,$$

где  $q$  – нагрузка на сеть одной камеры;

$V_k$  – объем одного кадра изображения;

$n$  – количество кадров в секунду.

Расчет объема одного кадра изображения:

$$V_k = 1024 \cdot 768 \cdot \frac{24}{65,854} = 286609,29 \text{ бит}$$

Далее будет рассчитана нагрузка на беспроводную сеть:

$$q = 286609,29 \cdot \frac{10}{1024 \cdot 1024} = 2,73 \text{ Мбит/с}$$

Из расчета следует, что пропускная способность сети составляет 2,73 Мбит/с, что является допустимым значением и не существенным для домашней сети, скорость которой в среднем является не менее 20–30 Мбит/с.

Оценим объем архива, при записи видео на протяжении 24 часов с разрешением 1024x768 и количеством кадров в секунду равным 10.

Исходя из результата нагрузки на беспроводную сеть, будет использоваться уже вычисленным значением объема одного кадра изображения, равным 286 609,29 бит.

Формула расчета архива системы видеонаблюдения:

$$V = V_k \cdot n \cdot t,$$

где  $V$  – требуемый объем памяти на устройстве хранения,

$V_k$  – объем одного кадра изображения;

$n$  – количество кадров в секунду;

$t$  – время непрерывной съемки камеры.

Расчет количества кадров за час:

$$n = 10 \cdot 60 \cdot 60 = 36000 \text{ кадров}$$

Далее будет рассчитан необходимый объем памяти и переведем значение из бит в Гб:

$$V = 286\,609,29 \cdot 36000 \cdot \frac{24}{1024^3 \cdot 8} = 28,8 \text{ Гб}$$

Из расчета следует, что для записи видео с одной камеры на протяжении суток необходимо 28,8 Гб, что сильно заполняет память ненужными данными, в связи с чем в реализации проекта запись начинается только после обнаружения движения. Также стоит отметить, что объем карты microSD должен быть не менее 32 Гб, чтобы не возникало проблем с отсутствием памяти для хранения файлов.

Разработаем алгоритм работы системы (Рис. 1). После запуска Motion начинается транслирование картинки на выделенном порту. При обнаружении движения выполняется скриншот и начинается запись видеофрагмента, которые сохраняются на карте памяти microSD.

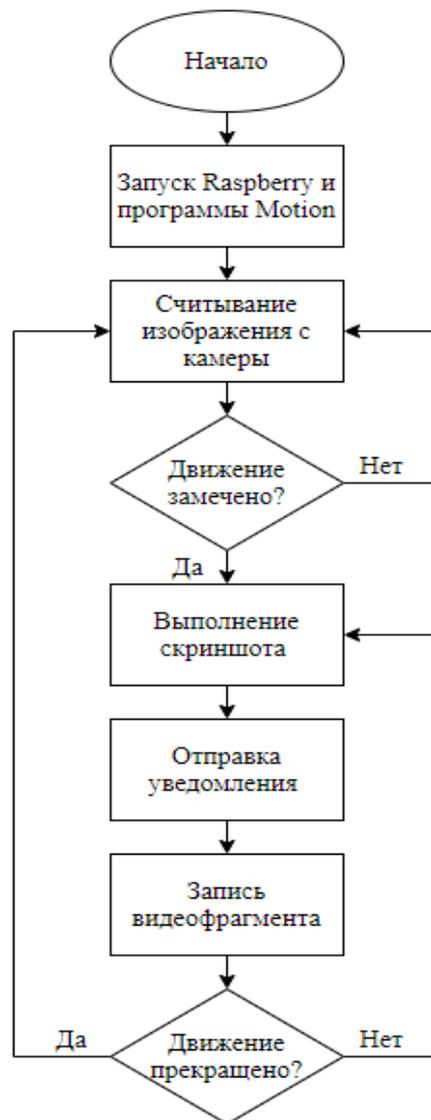


Рис. 1. Блок-схема алгоритма работы

Далее рассмотрим программную реализацию. Так как в реализации прототипа системы видеонаблюдения используется камера, подключенная через CSI порт, то в первую очередь необходимо включить ее в настройках: «sudo raspi-config»

В появившемся меню «camera» следует сменить опцию на «enabled», после чего Raspberry Pi нужно перезагрузить

После перезапуска для проверки работы и подключения камеры выполним следующую команду: «vcgencmd get\_camera»

Получен ответ «supported=1 detected=1», значит, камера функционирует должным образом.

Далее приступим к шагам по установке компонентов, последовательно вводя команды, описанные ниже.

Для начала необходимо обновить индекс пакетов, то есть файл, который содержит список программных пакетов. Затем следует обновить все программные пакеты, состоящие в системе до последних версий. Выполним следующие команды:

```
«sudo apt-get update»
```

```
«sudo apt-get upgrade»
```

Установка кодека ffmpeg и других зависимостей для программы Motion:

```
«sudo apt-get install ffmpeg libmariadb3 libpq5 libmicrohttpd12»
```

Установка программы Motion при помощи инструмента wget с github: «sudo wget https://github.com/MotionProject/motion/releases/download/release-4.2.2/pi\_buster\_motion\_4.2.2-1\_armhf.deb pkg-ipi\_buster\_motion\_4.2.2-1\_armhf.deb»

Установка менеджера пакетов и файлов заголовков Python, а также остальные необходимые пакеты:

```
«sudo apt-get install python-pip python-dev libssl-dev libcurl4-openssl-dev libjpeg-dev libz-dev »
```

Установка MotionEye при помощи утилиты pip, которая извлечет необходимые зависимости Python:

```
«sudo pip install motioneye»
```

Создание каталога для файла конфигурации:

```
«sudo mkdir -p /etc/motioneye»
```

Копирование файла конфигурации, на случай его редактирования:

```
«sudo cp /usr/local/share/motioneye/extra/motioneye.conf.sample /etc/motioneye/motioneye.conf»
```

Создание каталога для хранения, сделанных при обнаружении движения скриншотов и видеоснимков:

```
«sudo mkdir -p /var/lib/motioneye»
```

Копирование скрипта инициализации, который будет запускать MotionEye при включении Raspberry:

```
«sudo cp /usr/local/share/motioneye/extra/motioneye.systemd-unit-local  
/etc/systemd/system/motioneye.service»
```

Далее необходимо перезапустить фоновую программу и MotionEye:

```
«sudo systemctl daemon-reload»
```

```
«sudo systemctl enable motioneye»
```

```
«sudo systemctl start motioneye»
```

Затем надо обновить MotionEye до последней доступной версии и перезапустим:

```
«sudo pip install motioneye --upgrade»
```

```
«sudo systemctl restart motioneye»
```

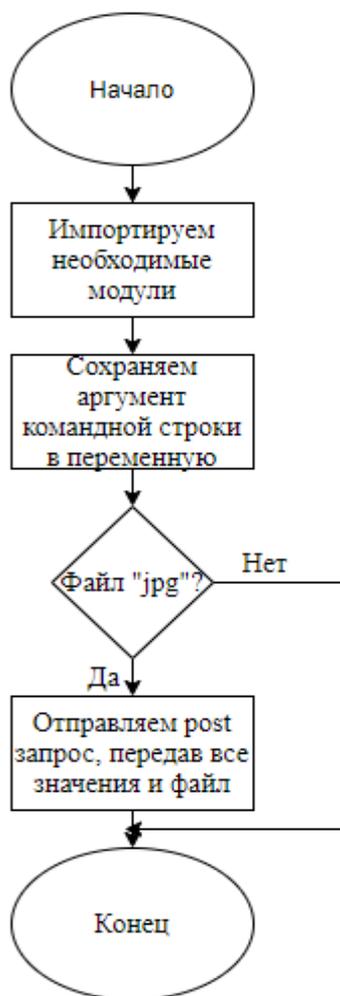
После установки всех необходимых компонентов, по средствам следующей команды будет известен ip устройства:

```
«sudo ifconfig»
```

Далее полученный ip будет использован для открытия в браузере графического интерфейса motionEye, введя следующий адрес «<https://192.268.0.220:8765>», где «8765» порт для доступа к интерфейсу.

Чтобы получить полный доступ к интерфейсу достаточно ввести только имя пользователя «admin» и полный доступ к графическому интерфейсу будет получен.

Следующим этапом является разработка скрипта на языке программирования Python, который отправляет push-уведомления в приложение на телефоне о детектировании движения в границах камеры с прикрепленным скриншотом и ссылкой на трансляцию с камеры домашнего видеонаблюдения. Перед тем, как написать скрипт стоит составить алгоритм работы, на который стоит ориентироваться при разработке (Рис. 2).



**Рис. 2. Алгоритм работы скрипта**

Далее создадим файл скрипта на языке Python через терминал:

```
«touch ./pushover.py»
```

После создания файла изменим права для доступа к нему, установим значение «777» – что означает разрешение всех действий:

```
«sudo chmod 777 pushover.py»
```

Далее с помощью текстового редактора Nano будет открыт, созданный ранее файл и вставим, написанный на языке Python код:

```
«sudo nano pushover.py»
```

Ниже приведен исходный код:

```
#!/usr/bin/python
```

```
import sys,requests # импорт модулей
```

```
filename = sys.argv[1]; # список аргументов командной строки
```

```
if filename.endswith('.jpg'): # проверка на получение изображения
r = requests.post("https://api.pushover.net/1/messages.json", # post запрос
data={
"token":"axzm7rvbug22ygx89cmfzm2w42kosc", # API токен приложения
"user":"uhnwzvivrxd5y426i4m9fn3q4xu9zm", пользовательский токен
"message":"Motion detected!", # текст сообщения push-уведомления
"url": "http://192.168.0.220:8081", # ссылка на трансляцию с камеры
"url_title": "Stream link", # текст ссылки
"sound": "siren", # звуковой сигнал push-уведомления
},
files={
"attachment":open(filename,"rb") # открывает переданные параметры в
бинарном режиме только для чтения
})
```

Для применения данного скрипта будет использоваться графический интерфейс MotionEye, в разделе отправки уведомлений будет указан путь расположения скрипта на устройстве.

Для настройки параметров отображения надо открыть в ранее созданной директории «/var/lib/motioneye» файл конфигурации «motion.conf». В этом файле проверим наличие следующих компонентов:

«webcontrol\_localhost on» - данный параметр отвечает за ограничение управления графическим интерфейсом MotionEye, то есть управление интерфейсом будет доступно только на том компьютере, на котором запущена программа Motion.

«camera camera-1.conf» - в данном параметре указывается полный путь и имя файла конфигурационного файла камеры.

Следующим этапом необходимо настроить непосредственно конфигурацию камеры:

«stream\_maxrate 10» - это максимальное значение частоты кадров в секунду для трансляции, значение может быть выбрано от 1 до 100. В данном

случае стоит значение 12, как самое оптимальное для прототипа домашнего наблюдения.

«stream\_localhost off» - значение параметра «off» отвечает за возможность трансляции с камеры не только на том компьютере, где была запущена программа Motion.

«stream\_port 8081» - это TCP порт, который отправляет основной поток.

«height 768» - высота в пикселях для каждого кадра. Диапазон данного значения зависит от используемой камеры.

«width 1024» - ширина в пикселях для каждого кадра. Диапазон также зависит от камеры.

«minimum\_motion\_frames 5» - прежде чем кадры будут распознаны, как истинное движение, должно содержаться, как минимум указанное значение кадров. Так, при установлении в 1 будут обнаруживаться все движения. Допустимый диапазон для данного параметра от 1 до 1000, но для качественного функционирования стоит держать показатель от 1 до 5. А при использовании более высокого параметра Motion будет пропускать большое количество кадров.

«noise\_tune on» - функция постоянно регулирует порог шума для различия шума от движения.

«noise\_level 32» - данная функция устраняет изменения, вызванные электрическим шумом в камере, который особенно заметны в темноте, либо при плохом освещении. Значение параметра означает, что интенсивность пикселя должна изменяться больше, чем параметр порога шума для подсчета. Доступный диапазон от 1 до 255, но лучше оставить рекомендуемый по умолчанию в программе Motion показатель 32.

### **Библиографический список:**

1. Демьяновски В. CCTV. Библия видеонаблюдения. Цифровые и сетевые технологии/Пер, с англ. - М.: ООО «Ай-Эс-Эс Пресс», 2006, — 480 с.