

---

---

# БИЗНЕС-ИНФОРМАТИКА

УДК 004

## **ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС ДЛЯ ЦЕНТРАЛИЗОВАННОГО УПРАВЛЕНИЯ СОСТОЯНИЕМ ВИДЕОПРОЕКТОРОВ В УЧЕБНЫХ АУДИТОРИЯХ**

**А.А. Лисс, А.И. Пестунов, Н.А. Линкевич, А.И. Семенков,  
Д.Д. Клипов, И.Е. Медведев, Т.М. Пестунова**

Новосибирский государственный университет  
экономики и управления «НИНХ»  
E-mail: pestunov@gmail.com

В статье изложена концепция и проектное решение программно-аппаратного комплекса (ПАК-УСВ), предназначенного для удаленного управления аудиторными видеопроекторами и мониторинга их состояния. Комплекс обеспечивает централизованный унифицированный доступ к разнотипным видеопроекторам с инфракрасным приемником и удобный способ включения и отключения проектора аутентифицированными пользователями. Основой комплекса является программно-аппаратное устройство (ПАУ), разработанное на базе микроконтроллера Arduino, к которому подключены инфракрасный приемник и инфракрасный передатчик для взаимодействия с целевым видеопроектором, а также RFID-считыватель для аутентификации пользователя посредством предъявления электронного пропуска. ПАУ связано с персональным компьютером, к которому подключен видеопроектор (ПК-ПРО). Для ПК-ПРО, ПАУ и сервера, через который ведется централизованное управление всеми видеопроекторами, разработано собственное программное обеспечение. Допускается расширение функционала ПАК-УСВ для управления другими устройствами (например, кондиционерами, плазменными панелями, DVD-плеерами), которые также оснащены инфракрасным приемником.

*Ключевые слова:* инфракрасный приемник, проектор, пульт дистанционного управления, микроконтроллер, Arduino, RFID.

## **SECURE COMPUTER APPLIANCE FOR REMOTE CONTROL AND MONITORING OF MULTIMEDIA PROJECTORS**

**A.A. Liss, A.I. Pestunov, N.A. Linkevich, A.I. Semenov,  
D.D. Klipov, I.E. Medvedev, T.M. Pestunova**

Novosibirsk State University of Economics and Management  
E-mail: pestunov@gmail.com

In this paper, we present a computer appliance, which is intended for remote control and monitoring of multimedia projectors. The computer appliance is developed and embedded in Novosibirsk State University of Economics and Management and provides easy

and unified access for different multimedia projectors with an infrared receiver. The developed computer appliance allows potential extension in order to control other devices with an infrared receiver (e.g. air conditionings, plasma display panels, DVD-players etc.). The main part of the computer appliance is an Arduino microcontroller with an infrared transmitter, an infrared receiver and a RFID-reader for users' authentication via demonstration an electronic pass. The microcontroller is connected with the personal computer, which, in its turn, is connected with the multimedia projector. Also, we developed special software for this computer, for the microcontroller, as well as for the server, which is intended for centralized control of all the available projectors.

*Keywords:* infrared receiver, projector, remote control, microcontroller, Arduino, RFID.

## 1. ВВЕДЕНИЕ

Современный уровень развития информационных технологий характеризуется стремительным ростом числа бытовой техники и устройств, оснащенных пультами дистанционного управления (далее – ПДУ) посредством передачи инфракрасных сигналов, в том числе в разнообразных социально значимых учреждениях (университетах, поликлиниках, детских садах и пр.). Полные функциональные возможности в таких устройствах, как правило, могут быть получены только через ПДУ, поэтому при отсутствии доступа к нему соответствующая техника часто не может быть использована в штатном режиме. Более того, если устройство расположено в труднодоступном месте (например, под потолком), то невозможными становятся даже базовые операции, такие как включение, выключение или регулировка (например, температуры в кондиционере). Отсутствие доступа к ПДУ может быть вызвано потерей, поломкой, отсутствием батарей питания и многими другими факторами. Замена недоступного ПДУ другим имеющимся в распоряжении пультом не всегда возможна, так как используемые в одной организации устройства разных производителей, разных моделей, с разным программным обеспечением и иной спецификой зачастую несовместимы между собой. Схожая проблема, хорошо известная многим пользователям, возникает при подключении DVD-плеера или видеомagneтофона к телевизору, когда требуются разные ПДУ для каждого из двух устройств.

В организациях, где доступ к устройствам необходим многим (но не всем) сотрудникам (например, только преподавателям) для предотвращения утраты или порчи ПДУ, они выдаются под подпись и (или) назначаются ответственные сотрудники, работа которых состоит в обеспечении работоспособности устройств и хранении ПДУ. Эти и подобные способы могут доставлять неудобства и замедлять рабочие процессы. Например, необходимо иметь систему информирования ответственных сотрудников о том, когда включать и выключать проекторы, преподавателям необходимо посещать специальную комнату, где выдаются ПДУ, и т.д. Кроме того, при ручном управлении сложно обеспечивать синхронное переключение каналов на всех устройствах для трансляции служебных или специальных сообщений. В ряде случаев требуется послать устройству не команду, а серию команд (сценарий), для того чтобы привести его в заданное состояние.

При ручном управлении время работы проектора может быть неоптимальным (например, в силу того, что проекторы простаивают во включенном состоянии), что приведет к более быстрому износу проектора и необходимости его более частого обслуживания. Прежде всего это касается замены достаточно дорогостоящих ламп. Сложно также узнать, сколько отработала лампа проектора и рассчитать ее оставшийся ресурс.

Создание ПАК-УСВ преследует следующие организационно-технические цели:

- повышение оперативности включения/выключения аудиторных проекторов;
- увеличение полезного времени работы проекторов;
- обеспечение защиты от несанкционированного включения аудиторных проекторов;
- автоматизация управления проекторами и мониторинга их состояния;
- резервирование ПДУ на случай поломки, кражи, выхода из строя;
- персонификация сотрудников, использующих проекторы;
- приблизительный учет времени использования проекторов сотрудниками.

Несмотря на обилие исследований в области удаленного управления техническими устройствами (см., например, [1–4]), прикладные решения аналогичного назначения в литературе не описаны и не представлены на рынке соответствующих технологий.

## **2. ЛОГИЧЕСКАЯ АРХИТЕКТУРА ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА**

### **2.1. Общая схема взаимодействия элементов**

Основным компонентом ПАК-УСВ является ПАУ на базе микроконтроллера Arduino с подключенными к нему инфракрасными приемником и передатчиком для взаимодействия с проектором, а также RFID-считывателем для аутентификации пользователя посредством предъявления электронного пропуска. ПАУ связано с персональным компьютером, к которому подключен проектор. Логическая схема ПАК-УСВ представлена на рис. 1.

Для включения проектора уполномоченный пользователь должен включить размещенный в аудитории компьютер ПК-ПРО, к которому подключен проектор, дождаться специального звукового сигнала от программно-аппаратного устройства ПАУ и прислонить свой персональный идентификатор (RFID-карту) к RFID-считывателю. Далее происходит процедура авторизации и в случае ее успешного прохождения посылается короткий монотонный звуковой сигнал с последующей отправкой на проектор необходимых инфракрасных команд, объединенных в сценарий. В момент прохождения процедуры авторизации происходит передача сообщения об этом событии на сервер. Данная информация содержит время события, его тип и хешированный номер карты. Хешированный номер карты заносится в локальный журнал клиентского компьютера ПК-ПРО и

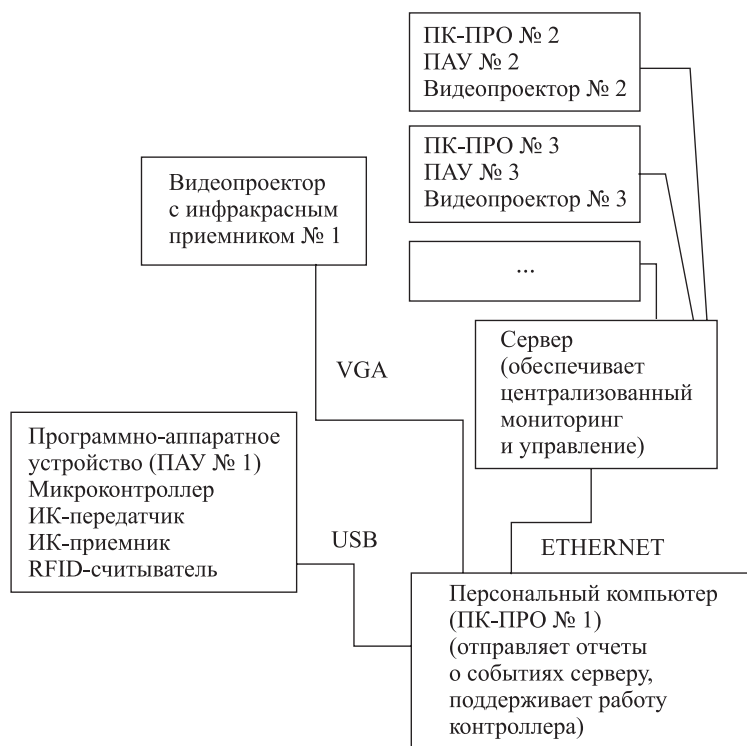


Рис. 1. Логическая схема программно-аппаратного комплекса

сервера. Сервер собирает информацию со всех клиентских компьютеров и выводит ее в виде Web-страницы.

Совместное функционирование всех элементов ПАК-УСВ обеспечивается авторским программным обеспечением (далее – ПО).

## 2.2. Функциональные требования к ПАК-УСВ

Исходя из описанной выше области применения комплекса, базовые функциональные возможности, которые должен предоставлять ПАК-УСВ, ориентированы на две категории пользователей. В первую категорию входят сотрудники, непосредственно использующие управляемые комплексом видеопроекторы при проведении учебных, научных и иных мероприятий в аудитории. Ко второй категории относится персонал, отвечающий за администрирование программно-аппаратной среды и информационной части комплекса.

При описании функций использованы термины «команда» и «сценарий». Под командой понимается элементарная команда, которая способна привести к атомарному действию проектора (например, включить, выключить или перейти в спящий режим). Для разных моделей проекторов команды могут быть различными.

Из команд формируются сценарии, представляющие собой их последовательности, приводящие к изменению состояния проектора. Сценарием, например, может быть следующая цепочка команд: включить, ввести PIN-код,

установить разрешение экрана и контрастность. Для кондиционера сценарий может состоять из команд включения, задания режима и температуры.

**Добавление (инициализация) новых проекторов** включает добавление модели проектора, ввод списка доступных команд для этой модели и времени задержки между отправкой команд. После этого добавляется конкретный проектор с указанием учебного корпуса, аудитории и пользователей, которым разрешено управлять проектором. Сценарии можно связать как с моделью проектора, так и с конкретным проектором.

**Управление учетными записями пользователей.** Данная функциональная возможность подразумевает право добавлять, изменять и удалять пользователей ПАК-УСВ. Для каждого пользователя указывается его фамилия, имя, отчество и хешированное значение номера RFID-карты, по которому будет осуществляться аутентификация. При создании пользователя, можно составить список проекторов, которыми он будет иметь право управлять. Администратор может создавать, изменять и удалять пользователей Web-интерфейса сервера. Пользователи Web-интерфейса представляют собой отдельные сущности, которые не связаны непосредственно с пользователями проекторов, поэтому добавляются отдельно. Для создания такого пользователя необходимо указать его электронную почту, пароль и роль.

**Управление состоянием проекторов через сервер** подразумевает возможность отправки команд проекторам. После прохождения процедуры авторизации пользователь получает доступ к панели управления, на которой выведен список всех проекторов. Пользователь может выделить один или несколько из них, выбрать один из доступных сценариев и инициировать его отправку на выбранные проекторы.

**Просмотр записей журнала и статистики.** Просмотр записей журнала допускает их фильтрацию по определенным критериям (дате, пользователю, конкретному проектору). Данный функциональный блок также предполагает возможность просмотра предполагаемого состояния аудиторных проекторов (включен, выключен, неисправен). Состояние предполагаемое, поскольку, подчеркнем это еще раз, проектор не имеет технической возможности отправить сигнал обратной связи. Состояние прогнозируется путем анализа записей журнала и обновляется по мере получения их от ПК-ПРО. Статистика включает в себя предполагаемое время наработки проектора и, кроме того, позволяет знакомиться со срезами различной информации (по датам, по пользователям и пр.).

**Управление состоянием проекторов через ПК-ПРО.** Данная функциональная возможность нацелена в основном на преподавателей, включающих и выключающих проекторы в аудитории. Подразумевается, что преподаватель, авторизуясь по своему RFID-пропуску, может отправлять доступные сценарии команд проектору.

Для комфортной и безопасной работы в системе реализована ролевая политика управления доступом, в рамках которой функциональные возможности сгруппированы в четыре роли: «Администратор», «Оператор», «Аудитор» и «Преподаватель». Администратор, оператор и аудитор – это пользователи серверной части, а преподаватель – это пользователь компьютера ПК-ПРО. Администратор обладает всеми функциональными воз-

возможностями. Эта роль назначается ответственному сотруднику, который обеспечивает администрирование программной среды и редактирование информационной части ПАК-УСВ, в том числе присвоение в системе ролей другим пользователям. Оператор может управлять состоянием проекторов через сервер (например, отключить проектор, если преподаватель забыл это сделать) и просматривать статистику. Аудитор имеет право на просмотр статистики. Преподаватель может управлять проекторами через ПК-ПРО. Поскольку пользователи серверной части и ПК-ПРО – это разные сущности, то, например, Администратору, чтобы управлять проекторами через ПК-ПРО, следует создать себе пользователя с ролью «Преподаватель».

### **2.3. Требования к составу и параметрам технических и программных средств**

Рассмотрим основные требуемые характеристики технических и программных средств. Согласно техническому заданию разрабатываемый программно-аппаратный комплекс должен удовлетворять следующим требованиям к надежности: время непрерывной работы должно составлять 144 ч; время ответа сервера ПАК-УСВ не должно превышать 2 с; ПАУ должно функционировать 7 дней в неделю согласно учебному расписанию; корпус и крепления ПАУ должны быть устойчивыми к непреднамеренным физическим воздействиям.

Следующие требования предъявлены к программно-аппаратному устройству (ПАУ): использование микроконтроллера ATmega с тактовой частотой не менее 16 МГц, объемом SRAM-памяти не менее 2 Кбайт, объемом Flash-памяти не менее 32 Кбайт и числом ШИМ-выходов не менее 6; рабочее напряжение ПАУ должно составлять 5 В; наличие инфракрасного передатчика (диода) для отправки команд на проектор; наличие RFID-считывателя карт (125 КГц); наличие корпуса для физической защиты устройства; наличие спикера для оповещения конечного пользователя; ПО для серверной стороны ПАК-УСВ поставляется в виде образа виртуальной машины VMware и должно иметь возможность запуска на виртуальной платформе операционной системы GNU/Linux Debian 8.

ПО для ПК-ПРО должно быть совместимо MS Windows 7 Pro SP1 (x86 либо x64-86) и должно иметь следующие функциональные возможности:

- распознавание ПАУ;
- получение от ПАУ хешированного номера идентификационной RFID-карты;
- принятие решения о разрешении включения проектора на основе списков доступа;
- ведение журнала событий;
- получение актуальной базы пропусков с центрального сервера;
- отправка журналов на центральный сервер;
- автономный режим с ограниченным функционалом при отсутствии связи с сервером.



## 2.4. Обеспечение безопасности программно-аппаратного комплекса

Кратко рассмотрим обеспечение безопасности ПАК-УСВ. Подробное описание модели угроз и мер защиты ПАК-УСВ приведено в [5]. Требования к безопасности разработаны исходя из следующих условий: идентификационная RFID-карта доступа находится у пользователя и не передается третьим лицам; устройство находится в закрытом корпусе; в ПО для ПАУ номера карт и команды включения проектора не хранятся; данные между ПК-ПРО и сервером, а также между пользователями Web-интерфейса и сервером передаются с помощью защищенных протоколов.

При выполнении перечисленных условий защищаемыми объектами являются хешированные значения номеров карт, фамилии и инициалы пользователей, соединение между сервером и ПК-ПРО. Нарушителями могут быть следующие лица: преподаватель или студент, который имеет возможность подключения сменных носителей и запуска с них стороннего ПО, ответственный за состояние аудиторных компьютеров и проекторов, а также администратор домена. Нарушители могут запускать вредоносное ПО, читать и изменять конфигурационные файлы ПАК-УСВ, подменять ПО, перехватывать трафик между узлами системы.

Основываясь на разработанной модели угроз, сформирован комплекс мер по обеспечению безопасности ПАК-УСВ.

## 3. ПРОГРАММНО-АППАРАТНАЯ АРХИТЕКТУРА КОМПЛЕКСА

### 3.1. Технологии для реализации серверной части

Клиентские компьютеры (ПК-ПРО), как правило, находятся за пределами механизма преобразования сетевых адресов (Network Address Translation), поэтому прямое подключение к ним со стороны сервера для отправки сценария на выполнение невозможно без дополнительной настройки перенаправления портов на маршрутизаторе клиентской подсети. Кроме того, ПК-ПРО может находиться под защитой файрвола, на котором заблокированы все порты, кроме стандартного для HTTPS № 443 и тех, что используются для служебных целей. Из этого следует, что, во-первых, подключение должен инициировать именно ПК-ПРО, а не сервер, во-вторых, соединение должно быть постоянным и, в-третьих, транспортный протокол должен использовать стандартные порты HTTPS. Этим критериям отвечает расширение протокола HTTP под названием WebSocket [8], являющееся протоколом полнодуплексной связи поверх TCP-соединения, предназначенное для обмена сообщениями между браузером (или иным клиентским приложением) и Web-сервером в режиме реального времени. Реализации сервера WebSocket существуют для многих языков программирования высокого уровня, в том числе Java, PHP и Python.

При разработке ПАК-УСВ удобство работы с WebSocket стало одной из основных причин выбора языка Python в качестве серверного. Под этот язык популярной библиотекой для сетевых приложений является gEvent [13] – высокопроизводительная и легковесная библиотека для асинхронной работы с сетью, написанная на языке программирования C. Для поддержки

WebSocket существует специальное расширение GEvent-WebSocket. В сфере Web-технологий Python несколько уступает в популярности языку PHP, но, на наш взгляд, выгодно отличается от последнего более продуманной архитектурой и инфраструктурой, а также универсальностью.

Если говорить про PHP, то для него реализацию протокола WebSocket предоставляет системная служба Ratchet, где взаимодействие с основными скриптами приложения, работающими под управлением Apache, осуществляется через локальный Unix- или TCP-сокеты, или даже через сторонний брокер сообщений. Иначе реализовать подобный функционал в PHP представляется технически сложной задачей, поскольку модель работы PHP предполагает запуск скриптов для обработки отдельных запросов, а не постоянно работающий сервис. Поэтому использование PHP для серверной части ПАК-УСВ оказывается неудобным.

Для построения серверного приложения выбран Flask [10] – гибкий, легковесный, минималистичный и простой в использовании фреймворк, предназначенный для построения небольших сетевых приложений. Фреймворк представляет собой объект WSGI, разбирающий запрос клиента и передающий управление объекту (или функции), назначенному на такой запрос. Привязка функций-обработчиков осуществляется при помощи функций-декораторов. Фреймворк Flask использует язык шаблонов Jinja2 [11] для генерации кода HTML.

Персистентность данных на сервере обеспечивается библиотекой SQLAlchemy [12], служащей средством синхронизации объектов Python и записей реляционной базы данных с использованием технологии ORM. Она позволяет описывать таблицы баз данных и способы взаимодействия с ними без использования SQL. SQLAlchemy поддерживает работу со многими СУБД и позволяет переключаться между ними простым изменением конфигурации.

Использование технологии ORM для автоматической генерации SQL-кода имеет ряд преимуществ по сравнению с его ручным написанием. Параметры запросов автоматически экранируются, что делает SQL-инъекции маловероятными. Повышается вероятность преиспользования запроса к серверу баз данных, что может позволить тому в некоторых случаях применить повторно план выполнения запроса. Несмотря на стандартизацию SQL, между СУБД есть различия в реализации, абстрагироваться от которых и позволяет ORM. При использовании объектно-ориентированного языка, как правило, приходится реализовывать одни и те же структуры дважды: в SQL-коде и непосредственно в коде приложения, определяя связи между ними, что усложняет внесение изменений. ORM облегчает этот процесс.

В качестве СУБД выбрана MySQL как наиболее распространенная открытая РСУБД. Для построения пользовательского интерфейса используется библиотека Backbone.js [6], предназначенная для структурирования кода в интерактивных Web-приложениях, jQuery для работы с DOM-моделью страницы, а также набор инструментов Bootstrap. Для визуализации записей журнала применяется Charts.js [7] – библиотека на языке JavaScript, позволяющая визуализировать данные в виде различных диаграмм.



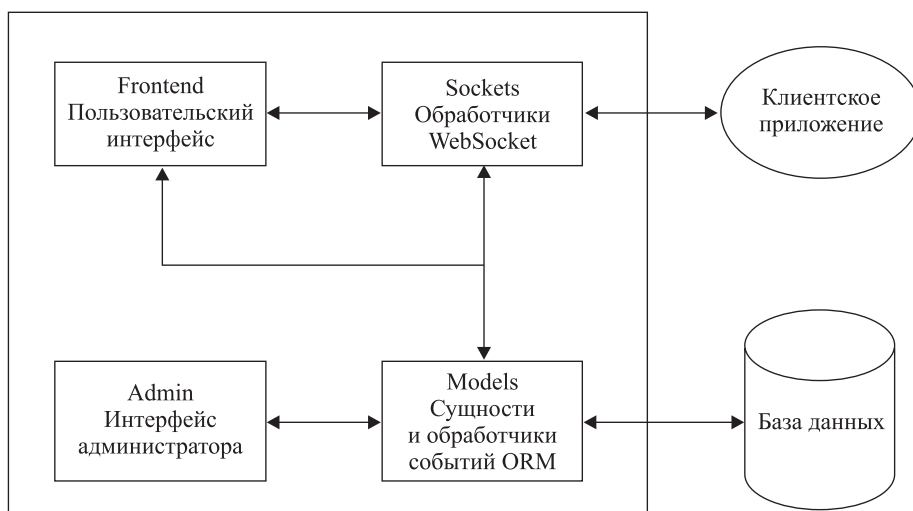


Рис. 2. Архитектура серверной части

Для разработки использовалась интегрированная среда PyCharm наряду с системой контроля версий Git. Архитектура серверной части представлена на рис. 2.

### 3.2. Технологии для реализации клиентской части

При разработке программного обеспечения для ПК-ПРО использованы язык программирования C# и фреймворк websocket-sharp. Программный код фреймворка websocket-sharp открыт, а распространяется он под лицензией MIT, позволяющей использовать его в составе закрытого (проприетарного) программного обеспечения.

Для создания аппаратной части ПАУ выбраны следующие компоненты: микроконтроллер Arduino Uno, RFID-считыватель Grove, зуммер Grove, ИК-передатчик Grove. Микроконтроллер Arduino обеспечивает высокую скорость разработки, имеет большое количество библиотек в сети Интернет и исчерпывающую документацию. Компоненты Grove также имеют исчерпывающую документацию, обладают единым интерфейсом подключения к микроконтроллеру и достаточно надежны.

### 3.3. Архитектура базы данных

Согласно результатам анализа предметной области, составлена концептуальная схема базы данных, развертываемой на сервере администратора. Выделены следующие сущности: пользователь, пользователь Web-интерфейса, роль пользователя Web-интерфейса, RFID-номер пользователя, помещение, здание, модель устройства, устройство, состояние устройства, команда, сценарий, журнал доступа, тип доступа, журнал действий, журнал устройства. Концептуальная схема базы данных представлена на рис. 3.

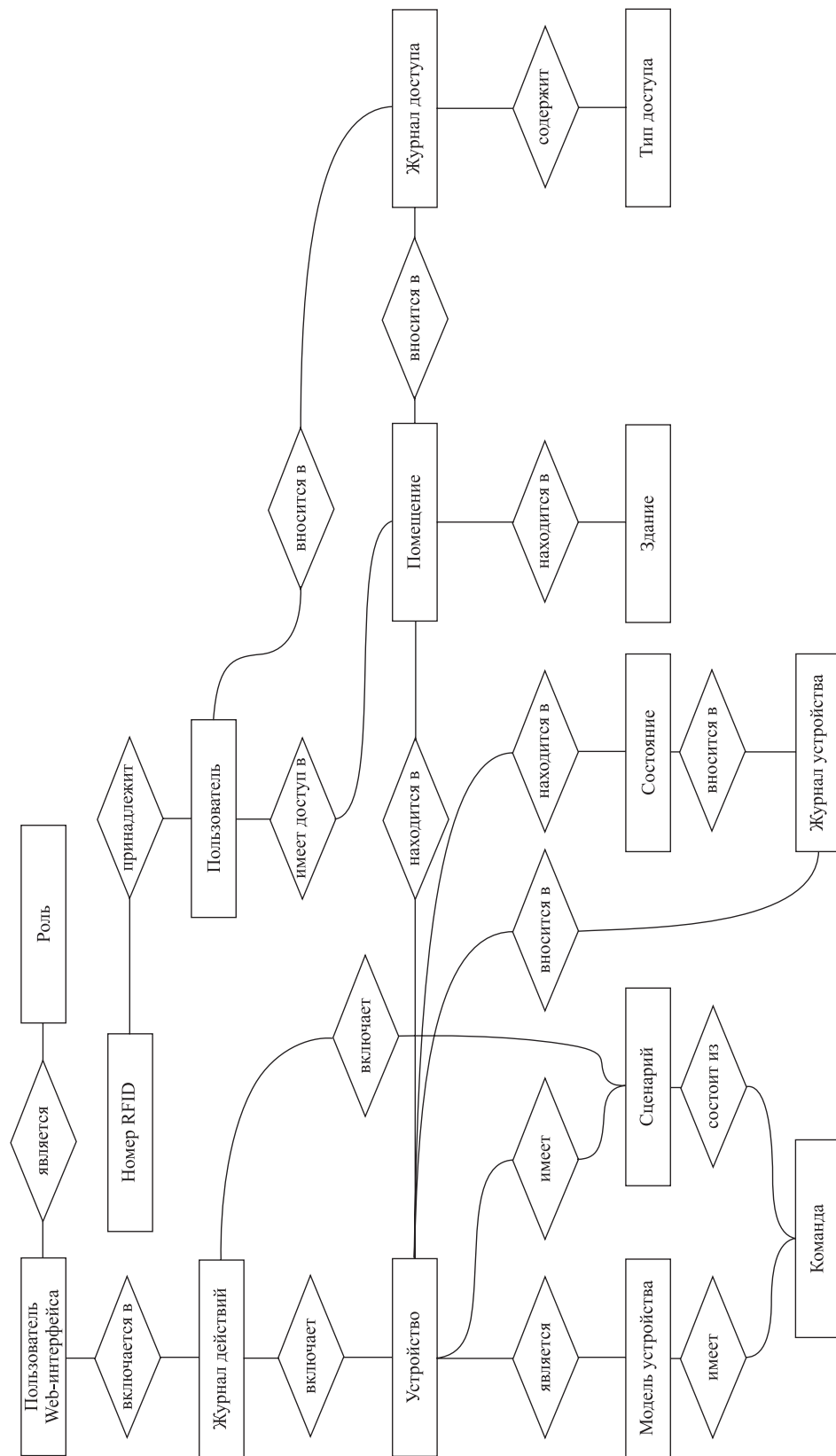


Рис. 3. Концептуальная схема базы данных

В таблице представлены классы, описывающие выделенные сущности, и их атрибуты.

**Сущности в базе данных**

Сущность	Описание	Атрибуты
State	Состояние устройства	Id – идентификатор состояния устройства; Name – наименование
Building	Строение (корпус)	Id – идентификатор строения; Name – наименование строения
AccessType	Тип доступа	Id – идентификатор типа доступа; Name – наименование
Room	Аудитория	Id – идентификатор помещения; Name – наименование помещения; BuildingId – внешний ключ для Building
Role	Роль пользователя Web-интерфейса	Id – идентификатор роли; Name – наименование роли; Description – описание роли
WebUser	Пользователь Web-интерфейса	Id – идентификатор пользователя Web-интерфейса; E-mail – электронная почта; Password – пароль пользователя; LastName – фамилия пользователя; FirstName – имя; MiddleName – отчество; Active – активен или нет; Roles – список ролей пользователя
Command	ИК-команда	Id – идентификатор команды; Content – тело команды; Name – наименование команды
Model	Модель устройства	Id – идентификатор модели устройства; Name – наименование модели; Delay – задержка между отправками команды; Commands – список доступных команд для модели
Script	Сценарий из ИК-команд	Id – идентификатор сценария; Name – наименование
Scripts-Commands-Association	Таблица связи между командой и сценарием	CommandId – внешний ключ для Command; ScriptId – внешний ключ для Script; Number – порядковый номер команды в сценарии; Cooldown – пауза, которую нужно выдержать, прежде чем перейти к следующей команде
Device	Устройство	Id – идентификатор устройства; Name – наименование устройства; Description – описание; ModelId – внешний ключ для Model; RoomId – внешний ключ для Room; StateId – внешний ключ для State; UpScriptId – внешний ключ для Script; DownScriptId – внешний ключ для Script; TotalWorkingTime – общее время наработки
User	Пользователь системы	Id – идентификатор пользователя; LastName – фамилия; FirstName – имя; MiddleName – отчество
RFID	Хешированный номер RFID-карты	Id – идентификатор; Hash – хешированное значение номера карты; UserId – внешний ключ для User
AccessLog	Запись в журнале доступа пользователей	Id – идентификатор записи; Date – дата и время события; RfidId – внешний ключ для RFID; RoomId – внешний ключ для Room
ExecLog	Запись в журнале действий пользователей Web-интерфейса	Id – идентификатор записи; Date – дата и время события; RoomId – внешний ключ для Room; UserId – внешний ключ для WebUser; ScriptId – внешний ключ для Script
DeviceLog	Запись в журнале изменения состояния устройства	Id – идентификатор записи; Date – дата и время события; DeviceId – внешний ключ для Device; From – внешний ключ для State; To – внешний ключ для State

#### 4. БИЗНЕС-ЛОГИКА ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА

На рис. 4 изображена схема взаимодействия элементов ПАК-УСВ. Далее в данном разделе рассматривается реализация представленных на этой схеме действий.

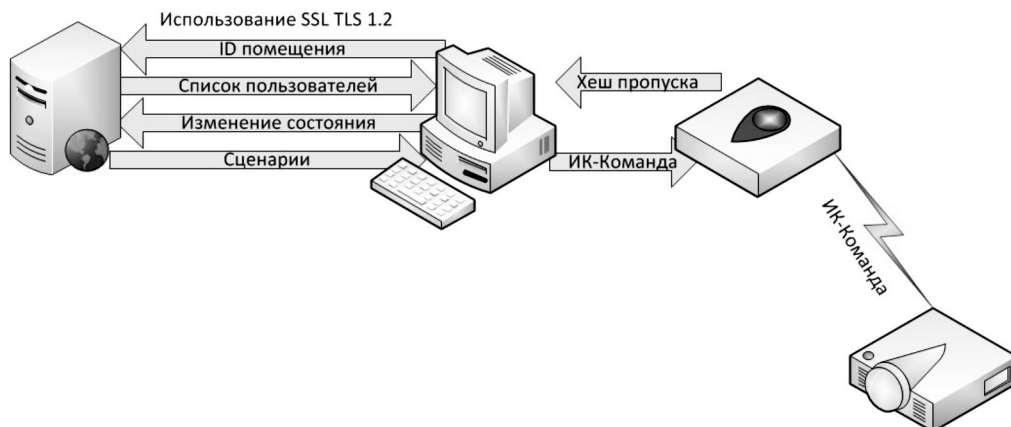


Рис. 4. Схема взаимодействия элементов ПАК-УСВ

##### 4.1. Взаимодействие ПК-ПРО и сервера

В модуле обмена сообщениями реализована событийно-ориентированная архитектура. Сообщения создаются в формате JSON в следующем виде:

```
{
  «event»: <название события>,
  «args»: <дополнительные параметры>
}
```

Каждое событие определяется в некотором пространстве имен, и ему назначается функция-обработчик, которая вызывается каждый раз, как клиент инициирует такое событие. В качестве аргументов обработчик принимает объект подключения и список аргументов args. Выбор соответствующего обработчика происходит в результате работы функции-диспетчера. Кроме того, созданный обработчик позволяет указывать для события пространство имен и, таким образом, разделять несколько обработчиков событий с одинаковыми названиями для разных областей применения.

Полный список событий следующий: connection (осуществляется аутентификация), query\_allowed\_users (возвращает список хешированных номеров идентификационных карт, которым разрешено использование указанной комнаты), query\_scripts (возвращает список доступных сценариев работы с проектором), query\_device\_info (возвращает информацию об устройстве), send\_access\_log\_event (передает сообщение об использовании RFID-считывателя, которое сервер сохраняет в базу данных), request\_execute\_script\_event (инициируется сервером, передает идентификатор сценария).

При подключении клиент инициирует событие connection и передает в качестве аргумента идентификатор комнаты, в которой находится. Это необходимо для аутентификации клиента: если в дальнейшем он инициирует другое событие, не пройдя аутентификацию, то получит сообщение об

ошибке. Этот идентификатор уникален для клиента, и если другой клиент попытается подключиться, используя тот же идентификатор, например, из-за ошибки настройки, то также получит сообщение об ошибке, и соединение будет разорвано.

Затем клиент инициирует события `query_allowed_users`, `query_scripts` и `query_device_info`, получив, таким образом, списки хешированных номеров карт, для которых доступно выполнение сценариев. После этого клиент и сервер переходят в режим ожидания.

Если пользователь системы воспользовался RFID-считывателем, то клиент рассчитывает хешированный номер пропуска и проверяет, есть ли он во множестве допущенных пользователей. Соответственно, далее система его либо допускает, либо – нет. Затем клиент отправляет сообщение, содержащее хешированный номер и результат действия.

## 4.2. Интерфейс администратора

Для работы с базой данных требовался некоторый пользовательский интерфейс, который бы позволял осуществлять просмотр, вставку, обновление и удаление данных. Такой интерфейс предоставляет расширение фреймворка Flask библиотека Flask-Admin [5]. Эта библиотека позволяет в декларативном виде описать представления для выбранных моделей, после чего она по предоставленному HTML-шаблону генерирует интерфейс администратора. По таким декларативным представлениям генерируется HTML код и сопутствующий серверный API, необходимый для работы AJAX в представлениях. Пример сгенерированной страницы, на которой отображены доступные устройства, представлен на рис. 5.

## 4.3. Пользовательский интерфейс серверной части

Пользовательский интерфейс разработан в виде так называемого одностраничного приложения (single page application), которое состоит из единственной HTML-страницы, являющейся оболочкой для предоставляемого функционала. При этом активно используются библиотеки Backbone.js и jQuery, а также фреймворк Bootstrap. Для каждого устройства создается специальный блок пользовательского интерфейса, позволяющий управлять им (рис. 6).

Код синхронизации состоит из моделей (Device, Room, AccessLog и ExecLogs), описывающих синхронизируемые сущности, отображений, привязанных к элементам страницы, и коллекций, задающих логику получения и манипулирования множеством данных. После загрузки серверной страницы со стороны ПК-ПРО создается подключение WebSocket, и дальнейшее взаимодействие между ПК-ПРО и сервером осуществляется через него. Для моделей определены обработчики событий, рассылающие широковещательные сообщения для клиентов (используется шаблон проектирования «Наблюдатель»).

Для отображения статистики использована библиотека Chart.js, позволяющая строить интерактивные графики по предоставленным данным. На рис. 7 приведен фрагмент журнала событий, а на рис. 8 приведен пример

Remoto Главная Web-интерфейс Помещения Устройства Пользователи Номера карт root@example.com

Список (3) Создать Экспортировать С выбранным

Наименование	Модель	Комната	Общее время работы, мин	Описание
<input type="checkbox"/>	Epson 123	1-101	0	
<input type="checkbox"/>	HP 4321	1-202	0	
<input type="checkbox"/>	Epson 123	1-303	0	

Рис. 5. Пример страницы, сгенерированной при помощи Flask-Admin

## Устройства

5-614 EPSON EB-W18
▶ Включить
■ Выключить

Состояние: нет данных, клиент подключен

🕒 0 минут

5-612 ViewSonic PJD-6253
▶ Включить
■ Выключить

Состояние: нет данных

🕒 0 минут

5-600 Toshiba
▶ Включить
■ Выключить

Состояние: нет данных

🕒 0 минут

Рис. 6. Пользовательский интерфейс для управления проекторами



Логи		Статистика по пользователям		Статистика по комнатам	
Время	Комната	Пользователь	Комната	Пользователь	Событие
11.05.16 11:24:26	5-612	f750a92832b0d477301772c234b42861e	5-612	f750a92832b0d477301772c234b42861e	Не авторизован
11.05.16 11:24:02	5-612	bcdab6eb2c9473824245f6e4fccc1d9d	5-612	bcdab6eb2c9473824245f6e4fccc1d9d	Не авторизован
11.05.16 11:23:45	5-612	bcdab6eb2c9473824245f6e4fccc1d9d	5-612	bcdab6eb2c9473824245f6e4fccc1d9d	Не авторизован
11.05.16 11:19:18	5-612	Н.А. Линкевич	5-612	Н.А. Линкевич	Выключил
11.05.16 11:19:13	5-612	Н.А. Линкевич	5-612	Н.А. Линкевич	Включил
11.05.16 10:13:46	5-612	Н.А. Линкевич	5-612	Н.А. Линкевич	Выключил
11.05.16 10:13:37	5-612	Н.А. Линкевич	5-612	Н.А. Линкевич	Включил
11.05.16 10:13:25	5-612	bcdab6eb2c9473824245f6e4fccc1d9d	5-612	bcdab6eb2c9473824245f6e4fccc1d9d	Не авторизован
04.05.16 00:50:02	5-600	Н.А. Линкевич	5-600	Н.А. Линкевич	Включил
04.05.16 00:47:31	5-600	Н.А. Линкевич	5-600	Н.А. Линкевич	Выключил
04.05.16 00:47:04	5-600	Н.А. Линкевич	5-600	Н.А. Линкевич	Включил
29.04.16 16:55:56	5-612	Н.А. Линкевич	5-612	Н.А. Линкевич	Выключил
29.04.16 16:48:09	5-600	9b82a08beaest1d14f419b66b25407cda	5-600	9b82a08beaest1d14f419b66b25407cda	Не авторизован

Рис. 7. Фрагмент журнала событий

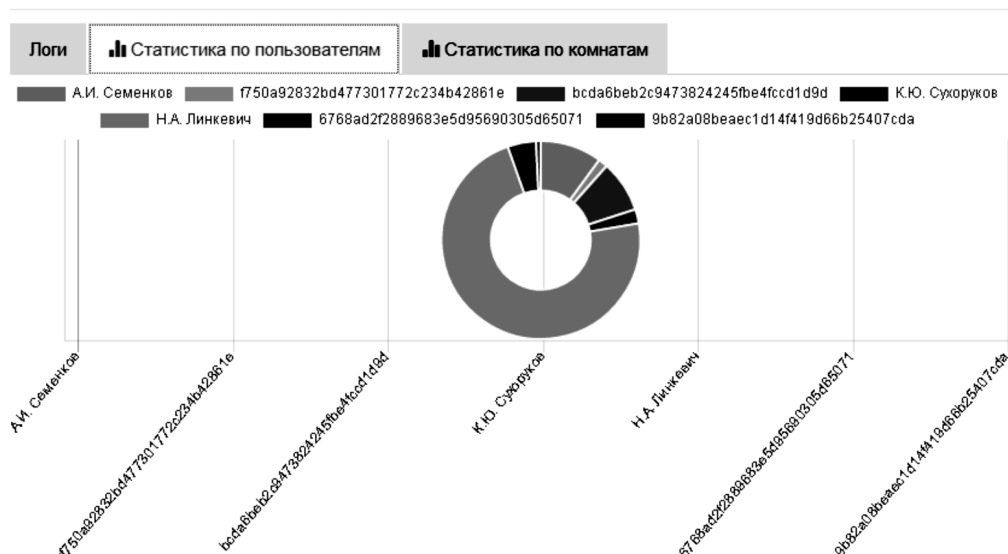


Рис. 8. Статистика времени использования проекторов по пользователям

отображения статистики времени использования проекторов по пользователям. Chart.js обеспечивает легкую расширяемость приложения. Так, для добавления нового вида статистики достаточно добавить функцию, генерирующую данные в формате JSON, вызвать ее из функции, возвращающей сводные данные для графиков, и добавить обработчик в соответствующее представление.

## 5. ТЕСТИРОВАНИЕ ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА И ЕГО ВНЕДРЕНИЕ

### 5.1. План тестирования

План тестирования ПАК-УСВ состоял из следующих основных тестов.

**Тест 1. Проверка сервера на общую устойчивость.** В течение 5 мин производить постоянный перезапуск программного обеспечения ПК-ПРО.

**Тест 2. Проверка времени реакции ПАУ.** В течение 2 мин постоянное поднесение нелегитимного пропуска.

**Тест 3. Проверка работы модуля принятия решения о легитимности пропуска.** В течение 2 мин поочередное поднесение легитимного и нелегитимного пропусков.

**Тест 4. Проверка корректности работы Web-интерфейса сервера.** Одновременное управление несколькими пользователями несколькими проекторами.

**Тест 5. Проверка работы модуля проверки целостности.** Преднамеренное искажение файлов со сценариями и хешированными номерами идентификационных карт.

**Тест 6. Проверка защищенности канала связи между ПК-ПРО и сервером.** Попытка реализации перехвата данных от ПК-ПРО до сервера посредством подмены сертификата.

## 5.2. Основные ошибки, выявленные в ходе тестирования

**Ошибка 1.** Во время внедрения выяснилось, что не все проекторы запускались при помощи ПАУ, хотя команда считывалась и посылалась верно. Оказалось, что это связано с тем, что некоторые модели проекторов требуют отправки команды два раза без задержек (чтобы убедиться, что команда считана ими верно).

**Ошибка 2.** Не всегда проверялись входные условия для данных, поступающих на ПК-ПРО, что приводило, в частности, к некорректному разбору данных в формате JSON.

**Ошибка 3.** При работе с ПАК-УСВ загрузка процессора ПК-ПРО доходила до 100 %.

**Ошибка 4.** Размер входного буфера ПАУ составлял 64 байта (это константа в заголовочном файле среды сборки), в то время как размер ИК-команды порядка 130 байт.

**Ошибка 5.** Ошибка в обработчике команд устройства при подаче неверных входных данных по UART-порту.

## 5.3. Устранение выявленных ошибок

Для устранения ошибок приняты следующие меры. Проведен мониторинг кода с последующим рефакторингом некоторых частей. Во все потоки встроена задержка 50 мс перед следующей итерацией, что позволило сократить пиковую нагрузку на процессор до 4–6 %. Некритичные по времени потоки переведены в фоновый режим. Создана специальная функция, контролирующая обмен сообщениями между потоками, что упростило отладку и повысило качество и читаемость кода. Модифицирована среда сборки программного обеспечения для ПАУ (она имеет открытый программный код) с целью увеличить входной буфер устройства до 256 байт. В результате ИК-команды стали помещаться в нем. Обработчик команд в ПАУ дополнен условием, очищающим входную очередь в случае невозможности распознать команды.

## 5.4. Установка программно-аппаратного комплекса

Установка программно-аппаратного комплекса в конкретной аудитории подразумевает выполнение следующих шагов.

Обследование помещения для выявления наилучшего местоположения ПАУ.

2. Сборка ПАУ и корректировка положения ИК-передатчика в нем.

3. Монтаж ПАУ в выбранном месте.

4. Инсталляция и настройка программного обеспечения на ПК-ПРО.

Сборка ПАУ подразумевает выполнение следующих шагов.

1. Установка RFID-считывателя.

2. Установка экрана из фольги между катушкой ИК-считывателя и микроконтроллером.

3. Установка зуммера.

4. Установка микроконтроллера Arduino и ИК-передатчика.

Процесс подготовки ПК-ПРО к работе производится следующим образом.

1. Создание аудитории на сервере ПАК-УСВ и получение ее идентификатора.
2. Привязка конкретного проектора к конкретной аудитории.
3. Считывание команды с ПДУ проектора и ее занесение в список команд проектора.
4. Создание сценариев включения/выключения проектора и их привязка к проектору.
5. Добавление необходимых команд в сценарии.
6. Формирование конфигурационного файла для программного обеспечения ПК-ПРО.
7. Установка и настройка программного обеспечения для ПК-ПРО.

## 6. ЗАКЛЮЧЕНИЕ

Рассмотренные в настоящей статье логическая схема, архитектура, управляющее программное обеспечение и опытные образцы устройства ПАК-УСВ разработаны и апробированы в НГУЭУ. В ходе апробации и опытной эксплуатации были подтверждены все заявленные характеристики устройства. Существует принципиальная возможность расширения разработанного программно-аппаратного комплекса посредством добавления в него функциональности по удаленному управлению другими устройствами, в том числе с кондиционерами, при работе с которыми также активно используются пульта дистанционного управления с инфракрасным передатчиком.

## Литература

1. Вахнин С.А., Братанин А.М. Безопасное управление удаленным хостом через незащищенный сервер // Математические машины и системы. 2012. Т. 1. № 2. С. 102–106.
2. Горчаков Л.В., Пархоменко А.А. Удаленное управление лабораторной установкой с использованием встроенной системы // Дистанционное и виртуальное обучение. 2010. № 12. С. 37–55.
3. Даринцев О.В., Мигранов А.Б., Голенастов И.В. Удаленное управление микробототехническими комплексами через сеть Интернет: архитектура системы и особенности реализации // Известия ЮФУ. Технические науки. 2006. № 16. С. 74–79.
4. Курицын Д.Б., Соколов В.В. Мониторинг и управление современным промышленным оборудованием // Труды НГТУ им. Р.Е. Алексеева. 2014. № 3. С. 222–227.
5. Лисс А.А., Пестунов А.И., Линкевич Н.А. и др. Защищенный программно-аппаратный комплекс для удаленного управления состоянием аудиторных видеопрокторов // Информационная безопасность в современном обществе (Росинфоком-2016): мат-лы науч.-практ. конф. (Новосибирск, СибГУТИ, 02 сентября 2016 г.). Новосибирск: СибГУТИ, 2016. С. 79–94.
6. Backbone.js [Электронный ресурс]. URL: <http://backbonejs.org/> (дата обращения: 21.07.2016).
7. Chart.js – Open source HTML5 Charts for your website. [Электронный ресурс]. URL: <http://www.chartjs.org/> (дата обращения: 21.07.2016).

8. *Fette I., Melnikov A.* The WebSocket Protocol. [Электронный ресурс]. URL: <https://tools.ietf.org/html/rfc6455/> (дата обращения: 21.07.2016).
9. Flask-Admin. [Электронный ресурс]. URL: <https://flask-admin.readthedocs.io/en/latest/> (дата обращения: 21.07.2016).
10. Flask (A Python Microframework). [Электронный ресурс]. URL: <http://flask.pocoo.org/> (дата обращения: 21.07.2016).
11. Jinja (The Python Template Engine). [Электронный ресурс]. URL: <http://jinja.pocoo.org/> (дата обращения: 21.07.2016).
12. SQLAlchemy – The Database Toolkit for Python. [Электронный ресурс]. URL: <http://www.sqlalchemy.org/> (дата обращения: 21.07.2016).
13. What is gevent? [Электронный ресурс]. URL: <https://gevent.org/> (дата обращения: 21.07.2016).

### Bibliography

1. *Vahnin S.A., Bratanin A.M.* Bezopasnoe upravlenie udalennym hostom cherez nezashhishhennyj server // *Matematicheskie mashiny i sistemy*. 2012. T. 1. № 2. P. 102–106.
2. *Gorchakov L.V., Parhomenko A.A.* Udalennoe upravlenie laboratornoj ustanovkoj s ispol'zovaniem vstroennoj sistemy // *Distancionnoe i virtual'noe obuchenie*. 2010. № 12. P. 37–55.
3. *Darincev O.V., Migranov A.B., Golenastov I.V.* Udalennoe upravlenie mikrorobototekhnicheskimi kompleksami cherez set' Internet: arhitektura sistemy i osobennosti realizacii // *Izvestija JuFU. Tehnicheskie nauki*. 2006. № 16. P. 74–79.
4. *Kuricyn D.B., Sokolov V.V.* Monitoring i upravlenie sovremennym promyshlennym oborudovaniem // *Trudy NGTU im. R.E. Alekseeva*. 2014. № 3. P. 222–227.
5. *Liss A.A., Pestunov A.I., Linkevich N.A. i dr.* Zashhishhennyj programmo-apparatnyj kompleks dlja udalennogo upravlenija sostojaniem auditornyh videoproektorov // *Informacionnaja bezopasnost' v sovremennom obshhestve (Rosinfokom-2016): matly nauch.-prakt. konf. (Novosibirsk, SibGUTI, 02 sentjabrja 2016 g.)*. Novosibirsk: SibGUTI, 2016. P. 79–94.
6. Backbone.js [Jelektronnyj resurs]. URL: <http://backbonejs.org/> (data obrashhenija: 21.07.2016).
7. Chart.js – Open source HTML5 Charts for your website. [Jelektronnyj resurs]. URL: <http://www.chartjs.org/> (data obrashhenija: 21.07.2016).
8. *Fette I., Melnikov A.* The WebSocket Protocol. [Jelektronnyj resurs]. URL: <https://tools.ietf.org/html/rfc6455/> (data obrashhenija: 21.07.2016).
9. Flask-Admin. [Jelektronnyj resurs]. URL: <https://flask-admin.readthedocs.io/en/latest/> (data obrashhenija: 21.07.2016).
10. Flask (A Python Microframework). [Jelektronnyj resurs]. URL: <http://flask.pocoo.org/> (data obrashhenija: 21.07.2016).
11. Jinja (The Python Template Engine). [Jelektronnyj resurs]. URL: <http://jinja.pocoo.org/> (data obrashhenija: 21.07.2016).
12. SQLAlchemy – The Database Toolkit for Python. [Jelektronnyj resurs]. URL: <http://www.sqlalchemy.org/> (data obrashhenija: 21.07.2016).
13. What is gevent? [Jelektronnyj resurs]. URL: <https://gevent.org/> (data obrashhenija: 21.07.2016).