

Министерство науки и высшего образования РФ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Уральский государственный педагогический университет»
Институт математики, физики, информатики и технологий
Кафедра информационно-коммуникационных технологий в образовании

ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС ДЛЯ УПРАВЛЕНИЯ ДОМАШНИМИ УМНЫМИ УСТРОЙСТВАМИ

*Выпускная квалификационная работа
бакалавра по направлению подготовки
09.03.02 – Информационные системы и технологии*

Исполнитель: студент группы ИСИТ-1501
Кошак Л.В.

Руководитель: к.п.н., доцент кафедры ИКТО
Арбузов С.С.

Работа допущена к защите
«____» _____ 2019 г.
Зав. кафедрой _____

Екатеринбург – 2019

Реферат

Кошак Л.В. ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС ДЛЯ УПРАВЛЕНИЯ ДОМАШНИМИ УМНЫМИ УСТРОЙСТВАМИ, выпускная квалификационная работа: 57 стр., рис. 12, библи. 39 назв.

Ключевые слова: ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС, УМНОЕ УСТРОЙСТВО, ДИСТАНЦИОННОЕ УПРАВЛЕНИЕ, СИСТЕМА УПРАВЛЕНИЯ.

Предмет разработки – программно-аппаратный комплекс для управления домашними умными устройствами.

Цель работы – разработать программно-аппаратный комплекс для управления домашними умными устройствами объединяющий умные устройства разных производителей в одной экосистеме.

В настоящей работе описаны результаты проектирования и программной реализации системы управления устройствами, позволяющей на основе современных веб-технологий объединять устройства в одном интерфейсе.

Система реализована путем интеграции VK API в качестве интерфейса и сервиса IFTTT в скрипт, расположенный на сервере.. Для реализации поставленной цели были проанализированы современные технологии проектирования и механизмы реализации систем домашней автоматизации.

Составлена техническая и сопроводительная документация по применению программно-аппаратного комплекса.

Оглавление

ВВЕДЕНИЕ.....	4
ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ РАЗРАБОТКИ АВТОМАТИЗИРОВАННЫХ СИСТЕМ	6
1.1 ОСНОВЫ РАЗРАБОТКИ СИСТЕМ ДОМАШНЕЙ АВТОМАТИЗАЦИИ	6
1.2 АНАЛИЗ ТЕХНОЛОГИЙ УПРАВЛЕНИЯ ДОМАШНИМИ УМНЫМИ УСТРОЙСТВАМИ...	10
1.3 ФОРМАЛИЗОВАННОЕ ОПИСАНИЕ ТЕХНИЧЕСКОГО ЗАДАНИЯ.....	20
ГЛАВА 2. РАЗРАБОТКА ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА.....	25
2.1 СЕРВИСЫ И ИНСТРУМЕНТЫ ИСПОЛЬЗУЕМЫЕ В РАЗРАБОТКЕ	25
2.1.1 Разработка интерфейса	25
2.1.2 Написание PHP скрипта	29
2.1.3 Настройка сервиса IFTTT	32
2.1.4 Настройка умного устройства	37
2.2 ОПИСАНИЕ РЕЗУЛЬТАТОВ РАЗРАБОТКИ. МОДЕЛЬНОЕ ПРЕДСТАВЛЕНИЕ ПРОЦЕССОВ 37	
2.3 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ.....	42
ЗАКЛЮЧЕНИЕ	46
СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ.....	48
ПРИЛОЖЕНИЯ	52
<i>ПРИЛОЖЕНИЕ 1.....</i>	<i>52</i>
<i>ПРИЛОЖЕНИЕ 2.....</i>	<i>55</i>

Введение

В настоящее время глобальная сеть Интернет является одним из важнейших изобретений за всю историю человечества. Развиваются системы автоматизации управления электронными устройствами, как на крупных предприятиях, так и в быту. Многие люди мечтают подключить к Интернету свои домашние устройства и управлять ими дистанционно с персонального мобильного устройства. Активно идут обсуждения проектов «умных домов», общего информационного пространства и других вариантов объединения разрозненных электронных компонентов в единую систему. При оснащении своего жилища современными умными устройствами может возникнуть проблема согласованной работы устройств разных производителей.

На сегодняшний день, практически у каждого более-менее крупного производителя умных устройств имеется своя, так называемая, экосистема, управление элементами и передача информации которой, по умолчанию, происходит внутри самой системы. Для человека, не являющегося специалистом в области электроники, программирования и микрокомпьютеров не предоставляется никаких возможностей слияния разных экосистем в одну, свою собственную. Поэтому идея создания программно-аппаратного комплекса для управления домашними умными устройствами разных производителей является весьма актуальной.

Предмет разработки: программно-аппаратный комплекс для управления домашними умными устройствами.

Цель работы – разработать программно-аппаратный комплекс для управления домашними умными устройствами объединяющий умные устройства разных производителей в одной экосистеме.

Для реализации поставленной цели необходимо решить следующие задачи:

1. Проанализировать основные способы построения и использования программно-аппаратных систем домашней автоматизации.

2. Произвести анализ умных устройств, рассмотреть их возможности и предлагаемые производителями способы управления.
3. В соответствии с техническим заданием провести разработку программно-аппаратного комплекса для управления домашними умными устройствами объединяющий умные устройства разных производителей в одной экосистеме.
4. Подготовить техническую и сопроводительную документацию по применению программно-аппаратного комплекса для управления домашними умными устройствами.

Глава 1. Теоретические основы разработки автоматизированных систем

1.1 Основы разработки систем домашней автоматизации

«Умный дом» или же «домашняя автоматизация» также можно встретить «интеллектуальное здание» – все это термины обозначающие современные жилые и офисные здания, квартиры, дома с единой автоматизированной системой управления и мониторинга сетей и систем, таких как:

- Освещение.
- Микроклимат.
- Безопасность.
- Коммуникационные системы.

Умным домом, в современном понятии этого слова, можно назвать дом, предоставляющий удобное управление всеми системами входящими в него, а также возможность их интеграции друг с другом, что повышает функциональность каждой из них и обеспечивает согласованную работу.

К ряду основных задач домашней автоматизации можно отнести:

- Повышение комфорта проживания.
- Повышение уровня безопасности. Система дает знать о возможных взломах или возникающих рисках.
- Экономия ресурсов за счет рационального расхода и распределения энергии.
- Возможность интеграции в систему диспетчерского контроля и управления. Такого рода функционал популярен среди административных и офисных зданий, но в некоторых случаях также используется в частных домах и квартирах.

На основе вышесказанного можно сделать вывод, что умный дом со временем станет важным элементом в вопросах удобства и жизнедеятельности большинства людей. Однако такой дом является высокотехнологичным изобретением и должен иметь хорошо спланированное построение системы.

На сегодняшний день можно выделить три варианта построения системы умного дома[1, 2]:

- централизованные системы;
- децентрализованные системы;
- смешанные.

Централизованные системы построены на основе центрального контроллера, который, по сути, является единым мощным процессором. Включая в себя исполнительные блоки с добавлением различных панелей управления. Такие системы распространены в основном в частном секторе. Все устройства подключаются к контроллеру, который является центром системы. Отдельные компоненты в системе могут иметь свои микроконтроллеры, но осуществляет управление и отвечает за взаимодействие всей системы центральный контроллер который может принимать и посылать сигналы управления по различным каналам. Главным преимуществом таких систем является не высокая стоимость всей системы, а недостаток в большой зависимости большинства устройств от центрального блока, который может выйти из строя.

Децентрализованные системы отличаются от централизованных отсутствием центрального контроллера. Все устройства соединяются непосредственно друг с другом при помощи шины и оснащены своим автономным контроллером. Недостаток таких систем в относительной сложности и дороговизне каждого отдельного устройства, но окупается это надежностью, а так же меньшим количеством или практически полным отсутствием проводных сетей.

Смешанные системы включают в себя элементы и принципы обеих вышеперечисленных систем. Как пример смешанной системы, можно привести X10. X10 – широко используемый в области домашней автоматизации стандарт определяющий протокол и метод передачи управляющих сигналов по беспроводным каналам или обычной электропроводке на электронные модули, к которым подключены управляемые бытовые устройства.

Помимо построения системы умного дома необходимо также решить, каким будет его управление.

Существуют следующие методы управления автоматизированными системами:

- управление на основе датчиков;
- управление с помощью панели управления и пульта;
- удаленное управление.

Самым простым способом управления являются кнопочные панели, где каждая кнопка соответствует какому-либо прибору, группе приборов или означает последовательность действий, которые нужно выполнить. На кнопках может быть световая индикация состояния соответствующего прибора.

Таким образом, под контролем находится практически все, от электросиловой структуры дома в целом до состояния каждого рабочего элемента в отдельности и количества потребляемой приборами мощности.

Необходимо понимать, что «умный дом» предназначен для умных людей. Между излишней автоматизацией и контруправлением человека следует найти разумный компромисс. Важным моментом является то, что человек должен иметь возможность в любой момент вмешаться в управление – с помощью пульта, Интернета или обычного телефона. Для этого все данные выводятся на табло, находящееся в самом доме или за его пределами, ведь автоматическая система регулирования подразумевает и множество обратных связей. А с помощью веб-камер можно не только управлять, но и видеть, что происходит дома.

Чтобы правильно распределить управление автоматизацией нужно разобраться в том, из чего может состоять система умного дома.

Система умного дома включает в себя несколько типов элементов. Контроллеры – это управляющие устройства, которые соединяют все части системы друг с другом и с внешним миром. К контроллерам подключаются датчики (например, датчики движения, уровня освещённости, детекторы дыма, видеокамеры), передающие информацию о внешних условиях и событиях, и актуато-

ры (электроприводы, реле, соленоидные приводы и т. д.), служащие для приведения в действие устройств, подключённых к системе «умный дом». Для системы «умный дом» потребуются как минимум один контроллер, несколько датчиков и некоторое число актуаторов, соответствующее числу таких подключённых устройств, как автоматические шторы или жалюзи, гаражные ворота, отопительный котёл и некоторые другие. Плюс ещё несколько дополнительных устройств, например блоки питания или, скажем, устройства приёма и передачи данных по радиоканалу или инфракрасному каналу (например, если у вас проводная система, а какой-то её элемент невозможно подключить по проводам).

Несмотря на множество плюсов, система умного дома имеет также и ряд минусов. Главной проблемой современного умного дома является фрагментированность рынка. Это значит, что многие смарт-устройства могут вообще не работать друг с другом. А в лучшем случае необходимо установить множество приложений на смартфоне для каждого из них. Это не самый удобный вариант. Некоторые фирмы предлагают комплексные системы «под ключ», где всё идеально работает, однако в таком случае не будет возможности объединить понравившиеся устройства: например, умную лампу от Xiaomi и увлажнитель от Apple.

Для решения этой проблемы необходимо разобраться в последнем современном и мощном способе общения с «интеллектуальным домом» – посредством интернета. С любого компьютера, подключенного к глобальной сети, можно узнать о состоянии дома, посмотреть записи камер видеонаблюдения и даже текущее изображение с них. При необходимости система может регулярно посылать своему хозяину по электронной почте или на его телефон, отчет о состоянии дома, о присутствии людей, о погоде и о многом другом. Конечно же, эта информация будет доступна только владельцам дома. А так же, используя технические возможности современных мобильных устройств, можно создавать различные сценарии работы домашних устройств, на основе взаимодействия пользователя с окружающей средой.

Таким образом, представляется актуальной разработка программно-аппаратного комплекса для управления домашними умными устройствами разных производителей. Для чего необходимо проанализировать существующие технологии и готовые решения.

1.2 Анализ технологий управления домашними умными устройствами

На сегодняшний день мобильные устройства, смартфоны и планшеты обладают высокими техническими характеристиками, а мобильный Интернет и облачные вычисления развиваются с огромной скоростью. Такие тенденции все больше соответствуют концепции «Интернет вещей». Согласно трактовке аналитической компании Gartner, понятие «Интернет вещей» означает «сеть физических объектов, содержащих встроенную технологию, которая позволяет этим объектам измерять параметры собственного состояния или состояния окружающей среды, использовать и передавать эту информацию» [3, 4].

В рамках проектирования умного дома концепцию Интернет вещей можно реализовать передачей данных в облако с множества используемых в системе устройств, где эти данные будут обрабатываться и храниться, а также путем передачи команд от пользователя на конечные управляемые устройства. В свою очередь пользователю будет доступен интерфейс (через браузер или мобильное приложение), предназначенный для мониторинга данных с датчика, а также для управления устройствами. Таким образом, пользователи получают инструмент удаленного управления и мониторинга всей системы умного дома.

Основные преимущества использования облачных технологий в системе умного дома следующие:

- Через облачный сервер возможно предоставление единого интерфейса управления системами умного дома в случае использования устройств от разных производителей с различными протоколами передачи данных [5].
- В любое время и в любом месте пользователю будет доступна вся информация об обстановке в доме.

- Сама система умного дома становится более гибкой, намного легче будет интегрировать новые устройства, лишь обеспечив им доступ в Интернет, не перестраивая всю систему заново.

Создание умного дома невозможно без системы управления им (автоматизированной системы управления зданием). При построении системы управления обычно выделяют три уровня автоматизации [6]:

1. Уровень управления и мониторинга. На данном уровне обеспечиваются управление и мониторинг всех систем автоматизируемого объекта через специальный графический пользовательский интерфейс, организованный на базе компьютерных средств и специального программного обеспечения.

2. Уровень автоматического управления. Основным компонент данного уровня – контроллер (специализированное устройство, к которому подключаются различные датчики и управляемые устройства). Контроллер предназначен для получения данных с датчиков и для управления исполнительными механизмами (приводы, задвижки, реле) подключенных устройств. Иногда к контроллеру применяется термин «модуль расширения», но только не надо путать его с модулями расширения, предназначенными для подключения к контроллерам с целью увеличения количества портов устройства. Если рассматривать проводные протоколы автоматизации, то количество подключаемого конечного оборудования весьма ограничено. В таком случае для подключения большого количества датчиков и управляемых устройств понадобится множество контроллеров.

3. Уровень конечного оборудования. Данный уровень включает в себя датчики и управляемые устройства. Датчики предназначены для отслеживания различных параметров (температура, освещенность и т.д.). К ним следует отнести и обычные кнопки, используемые для управления исполнительными устройствами (сигнал с них передается через контроллер). Управляемые устройства – это непосредственно устройства, управление которыми осуществляется

системой умного дома (например, устройства освещения, отопления, вентиляции и т.д.).

Устройства управления могут быть *мобильными и стационарными*.

Мобильные устройства – смартфоны или планшеты на мобильных операционных системах Android, iOS или Windows Phone. С мобильных устройств возможны передача команд на облачный сервер, а также получение от туда информации о состоянии датчиков. Отправка и получение данных могут осуществляться либо через нативное приложение, либо через браузер при заходе на сайт, предоставляющий услуги облачного сервиса. Кроме того, при нахождении пользователя непосредственно в автоматизированном доме можно организовать взаимодействие мобильных устройств по домашнему Wi-Fi или через Bluetooth [7].

Стационарные устройства – персональные компьютеры, ноутбуки. Работа с системой умного дома возможна либо через облачный сервис, либо по локальной сети через домашний сервер. Облачный сервер предназначен для хранения данных о состоянии датчиков и устройств в БД. Он выступает также в качестве посредника между удаленными устройствами управления и домашним сервером. Данная функция осуществляется путем передачи команд, например, с мобильных устройств в облако, где они обрабатываются и передаются дальше на домашний сервер.

Домашний сервер предназначен для получения команд от облачного сервера и их дальнейшей передачи на контроллеры, а также передачи в обратном порядке данных с датчиков, поступающих на контроллер. Через домашний сервер также обеспечивается управление системой умного дома по локальной сети как с самого домашнего сервера или другого домашнего компьютера, так и с мобильных устройств по Wi-Fi и Bluetooth.

В роли домашнего сервера можно использовать обычный персональный компьютер или тонкий клиент.

Существует и вариант отказа от домашнего сервера. Большинство производителей устройств автоматизации, использующихся в системах умного дома, выпускают широкий спектр устройств, среди которых есть контроллеры, способные напрямую получать доступ к сети Интернет. В этом случае облачный сервер может напрямую взаимодействовать с управляющими контроллерами.

Что касается управления такими контроллерами непосредственно из дома, то в данном случае осуществлять взаимодействие с ними можно по локальной сети.

На основе представленной выше аппаратной составляющей архитектуры системы управления умным домом, интегрированной в облако, рассмотрим программную составляющую. Она включает следующие уровни.

1. Серверная часть облака: веб-сервер, БД, скрипты для обработки и передачи данных.
2. Клиентская часть облака, предоставляющая графический пользовательский интерфейс управления и мониторинга. Строится на трех основных технологиях создания веб-приложений: HTML, CSS и JavaScript.
3. Серверное домашнее приложение. Комплектация аналогична серверной части облака.
4. Клиентское домашнее приложение. Комплектация аналогична клиентской части облака.
5. Мобильное приложение.
6. Программа для контроллера.

В целях эффективной работы предлагаемой системы управления умным домом необходимо обеспечить надежное взаимодействие между облаком и домашним сервером, а также между удаленными устройствами управления и облаком.

Для обеспечения взаимодействия в сети Интернет различных приложений и узлов существует технология веб-служб (веб-сервисов). В основном приме-

няются два способа реализации веб-служб: протокол SOAP и архитектурный стиль REST.

SOAP представляет собой протокол обмена структурированными сообщениями в формате XML в распределенной вычислительной среде [10].

REST является архитектурным стилем, при котором взаимодействие компонентов распределенного приложения в сети Интернет осуществляется посредством HTTP-запросов (в основном GET или POST) [11]. Для обозначения веб-сервисов, реализуемых с использованием HTTP и принципов REST, применяется термин «веб-сервис RESTful» [12]. Стил REST по сравнению с протоколом SOAP обладает целым рядом преимуществ, главные из которых следующие:

Кэширование. Для запроса данных в REST-архитектуре используются GET-запросы. В этом случае ресурсы, возвращаемые в ответ на запрос GET, можно кэшировать множеством разных способов, что увеличивает скорость и масштабируемость.

Функциональная совместимость. Для некоторых языков и сред до сих пор нет инструментальных средств SOAP. Для REST требуется только доступность библиотеки HTTP для большинства операций. Встраивание REST-сервисов в Интернет-сервис – относительно простой процесс по сравнению с усилиями, необходимыми для развертывания инфраструктуры и кода с целью поддержания SOAP-коммуникаций [13].

Время обработки запросов. В архитектуре REST гораздо меньше, чем при обработке сообщений, передаваемых через SOAP, соответственно при передаче больших объемов данных возрастает энергоэффективность системы умного дома [14].

Два основных HTTP-запроса – это GET и POST. Метод GET предназначен для получения ресурса с указанным URL. Получив запрос GET, сервер должен прочитать указанный ресурс и включить код ресурса в состав ответа клиенту. URL ресурса может указывать на исполняемый код программы, кото-

рый при соблюдении определенных условий должен быть запущен на сервере. В этом случае клиенту возвращаются данные, сгенерированные в процессе выполнения кода [15].

Метод POST предназначен для передачи данных на сервер. В данном случае запрос GET будет использоваться для запроса данных о состоянии устройств, входящих в систему умного дома. В свою очередь, передача управляющих команд будет осуществляться посредством POST-запроса. Для передачи данных через HTTP-запросы и получения ответа необходим легкий и удобный формат обмена данными. Как правило, передача данных в веб-службах осуществляется в одном из текстовых форматов: XML и JSON. В [16] проведено сравнительное исследование этих двух форматов. Установлено, что передача данных в формате JSON осуществляется намного быстрее, чем в формате XML. Кроме того, данные, представленные в формате JSON, гораздо легче обрабатывать. Также в большинстве современных языков программирования присутствует поддержка этого формата. JSON-текст представляется в одной из двух структур: объект – неупорядоченный набор пар ключ/значение, массив – упорядоченная коллекция значений [17].

Умный дом есть нечто большее, нежели группа IoT-вещей (умные лампы, термостаты, датчики охраны) или так называемых connected devices. Разница между умными бытовыми приборами и полноценной инженерной системой «умный дом» заключается в том, что второе можно дополнить первым, но только на умных устройствах построить полноценный умный дом невозможно! Во всяком случае, пока. По сути, мы имеем дело с необычными интересными игрушками, но это не умный дом, а отдельные умные устройства. Высокая степень автоматизации подключённых устройств и систем, а также и более высокая эффективность их работы достигается только с помощью инженерных решений типа LoXone, KNX и др., но точно не относится к Интернету вещей. Отдельные умные устройства хороши сами по себе, но из них не создать надёжную систему. К тому же, чтобы их интегрировать между собой, нужен

ХАБ (hub). При отключённом роутере уже ничего работать не будет. А в инженерном решении будет, так как там контроллер «лезет» в сеть только для интерфейса пользователя и сервисов типа погоды, а коммуникация между выключателем и лампочкой идёт через контроллер и шину, а не сеть.

Существует также множество готовых решений для создания умного дома. В каждом из данных решений представлены как плюсы, так и минусы, что позволяет создать максимально точное мнение о них.

LogicMachine, Российское решение, представляющее собой настоящий комбайн для умного дома [18]. Разные версии контроллера отличаются друг от друга набором интерфейсов. Последняя обладает самыми широкими возможностями среди всего семейства LogicMachine:

- Ethernet – семейство технологий пакетной передачи данных между устройствами для компьютерных и промышленных сетей [19].
- KNX/TP (TPUART2) – является «нервной системой» для проектов автоматизации. Сильными сторонами являются широчайшие возможности по поддержке различных сценариев и устройств, стандартизация и сертификация оборудования, особое внимание к надежности решений и активное сообщество профессионалов.
- USB – (англ. Universal Serial Bus – «универсальная последовательная шина») – последовательный интерфейс для подключения периферийных устройств к вычислительной технике.
- RS-485 – интерфейс осуществляющий связь между устройствами;
- DALI – сокращение от Digital Addressable Lighting Interface, универсальный стандарт управления освещением в зданиях. В особенности, для таких объектов, как офисы, магазины, рестораны или гостиницы.
- CEC – информационная шина, входящая в спецификацию HDMI интерфейса. Стандарт позволяет управлять несколькими устройствами через HDMI интерфейс.

- 1-Wire – распространённая полевая шина для множества применений.
- IR (Out) – предназначен для подключения инфракрасного переходника.

LogicMachine не имеют фиксированных встроенных алгоритмов работы, но могут быть запрограммированы под конкретную логику вашего проекта. Так что без знания программирования здесь не обойтись. В целом, на LogicMachine можно построить любой сценарий работы умного дома, написав свой собственный скрипт на языке Lua. Но возможности платформы позволяют использовать эти контроллеры даже в масштабах крупных предприятий. К примеру, Logic Machine 2 используется для управления светом в многоэтажных офисных зданиях.

Archos – недорогая система, представляющая собой планшет на базе Android с набором дополнительных устройств: мини-камера, датчик движения, датчик температуры и влажности, умные розетки [20]. Система беспроводная, что значительно облегчает монтаж, но зона покрытия небольшая: если вам нужно охватить большую двухэтажную дачу, то возникнут сложности. Есть встроенные сценарии поведения, а также простой и не требующий навыков программирования редактор. Сценарии простенькие, например, получить письмо на электронную почту и снимок с камеры наблюдения, когда кто-то подходит ко входной двери.

Z-Wave.Me [21]. Z-wave – это протокол беспроводной связи для создания умных домов. На его основе существует множество решений: от обычного центрального контроллера Mi Casa Verde Vera 3 до платы, превращающей Raspberry Pi в контроллер умного дома.

Для тех, кто предпочитает собирать «что-то своё» на основе Arduino, стоит обратить внимание на контроллер Fibaro Home Center [22]. Он находится на стыке между DIY-решениями для гиков и готовыми системами для обычных пользователей. В нем есть как встроенные автоматические сценарии, которые можно активировать за несколько кликов, так и редактор, позволяющий напи-

сать собственный скрипт на LUA. При этом контроллер очень прост в использовании и позволяет вообще не забивать себе голову какими-то скриптами.

К другим возможностям Fibaro Home Center можно отнести:

- разделение дома на зоны с разными сценариями поведения устройств;
- push-уведомления и система SMS-оповещений;
- поддержка удаленного доступа и других протоколов умных домов, что позволяет надстроить систему над уже существующим у вас решением.

Prestigio. Приложение под Android, позволяющее управлять яркостью и цветом свечения лампочек того же производителя. У Prestigio несколько линеек умных ламп: одни способны изменять яркость и цвет, а другие – только яркость.

Wink – это самое простое решение, если умный дом – «слишком сложно», дорого или бессмысленно [23].

Девиз этого устройства: «Управляем всеми гаджетами, которыми только можно управлять». Данный хаб совместим с сотнями устройств десятков различных производителей, поддерживает связь по протоколам Bluetooth, Wi-Fi, Z-Wave, Zigby, Lutron и Kidde. К сожалению, не все решения для умного дома хороши: от пользователей Wink поступило много негативных комментариев, а на Amazon оценка всего 2,6 балла.

SmartThings от Samsung – Это аккуратная белая коробка-хаб, совместимая с несколькими сотнями устройств и оснащенная батареей, позволяющей работать до 10 часов. Хаб позволяет выстраивать сценарии поведения камер безопасности, термостатов, умных розеток и других гаджетов. Это не требует никаких специальных навыков программирования [24].

Iris. Система домашней автоматизации с поддержкой множества различных устройств: камер видеонаблюдения, детекторов дыма, датчиков утечки воды, замков и прочего. В состав Iris входит центральный хаб, несколько отдельных датчиков и модулей, а также ряд наборов для автоматизации и безопасности.

Wiren Board. Ещё один представитель класса бюджетной домашней автоматизации. По интерфейсу RS-485 поддерживает водо- и энергосчётчики, устройства управления подачей питания (релейные модули), цифровые и аналоговые датчики, устройства управления освещением. Измеряемые параметры: температура, влажность, освещённость, давление, наличие газов.

Благодаря облачному веб-интерфейсу системой можно управлять удаленно даже по GPRS.

Последняя на данный момент версия решает практически любую задачу автоматизации, к тому же она не требует высокого профессионализма. Для работы с контроллером достаточно опыта работы с автоматикой или электроникой на уровне пользователя + знание JavaScript для написания сценариев [25].

Velocio Ace. Миниатюрный (всего 2,5 x 2,5 см) контроллер с минимальным потреблением 5 В. Продвинутая версия с MiniUSB и RS-232 – \$170. За эти деньги вы получите 31 вход и выход для подключения различных датчиков и возможность загружать в контроллер собственные программы. Это решение для тех, кто готов программировать свой умный дом и возиться со множеством мелких проводов. В старших моделях уже до 450 входов/выходов: вместо умного дома можно сделать целый умный квартал.

Loxone Miniserver. Этому контроллеру недостаточно батарейки, нужен блок питания. И не удивительно: в нем предусмотрен разъем EIB/KNX, порт LAN, слот microSD, интегрированный WEB-сервер и многое другое. К разъемам Miniserver можно подключить, например, 8 групп света и 4 радиатора, а дальше всё зависит от наличия расширений. В Miniserver есть как встроенные сценарии поведения, так и возможность написать собственные скрипты.

MajorDoMo – это бесплатная открытая программная платформа для комплексного управления домашней автоматикой. В качестве хаба выступает персональный компьютер (на Windows или Linux), к которому нужно подключить умные датчики – заводские или самодельные.

Из интересных особенностей можно выделить поддержку динамических 3D-сцен, позволяющую создавать в редакторе помещение с различными объектами и камерами, и привязывать к ней отдельные элементы со сценариями.

Таким образом, современные умные устройства имеют разное программное обеспечение и веб сервисы, позволяющие управлять ими дистанционно. Проанализировав существующие технологии построения системы умного дома и готовые решения управления домашней автоматизацией, становится очевидной актуальность разработки программно-аппаратного комплекса, объединяющего умные устройства в одной экосистеме.

1.3 Формализованное описание технического задания

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

на разработку программно-аппаратного комплекса для управления домашними умными устройствами

Составлен на основе ГОСТ 34.602-89 «Техническое задание на создание автоматизированной системы»

1. Общие сведения.

1.1. Название организации-заказчика.

Проектирование и разработка программно-аппаратного комплекса для управления домашними умными устройствами осуществляется в рамках научно-исследовательской деятельности кафедры информационно-коммуникационных технологий в образовании Уральского государственного педагогического университета.

1.2. Название продукта разработки (проектирования).

Программно-аппаратный комплекс для управления домашними умными устройствами.

1.3. Назначение продукта.

Система предназначена для управления домашними умными устройствами разных производителей через один общий интерфейс. В системе должна

быть предусмотрена возможность добавления новых устройств и создание различных сценариев их взаимодействия с пользователем.

1.4. Плановые сроки начала и окончания работ.

Начало работ: 01 сентября 2018 г.;

Окончание работ 15 мая 2019 г.

2. Характеристика области применения продукта.

2.1. Процессы и структуры, в которых предполагается использование продукта разработки.

Продукт разрабатывается для домашнего использования.

3. Требования к продукту разработки.

3.1. Требования к продукту в целом.

3.1.1. Структура системы

Программно-аппаратный комплекс должен представлять собой систему состоящую из программного кода, написанного на языке программирования PHP, соединяющего интерфейс управления с веб сервисом, который, в свою очередь, осуществляет управление домашними устройствами, соединяясь с серверами производителей устройств. Взаимодействующие элементы:

3.1.2. Программный код на языке программирования PHP обеспечивает выполнение следующих функций:

- обработка JSON запросов интерфейса управления и отправка результатов выполнения запросов обратно;
- организация взаимодействия всех элементов системы;
- отправка команд управления в форме http запросов на сервер IFTTT;
- реализация логики взаимодействия пользователя с веб-интерфейсом.

3.1.3. Интерфейс управления должен обеспечивать:

- Понятное и удобное пользователю управление устройствами включенными в систему;

- Оповещение о состоянии устройств.

3.1.4. Интегрированное в систему средство управления сервисами IFTTT позволяет:

- Добавлять новые умные устройства в систему, через соединение с серверами производителей;
- Создавать новые сценарии взаимодействия пользователя с устройствами, такие как: триггер определенных функций интерфейса (например push-up уведомление) или элементов домашней автоматизации, в зависимости от геолокации пользователя;

3.1.5. Устройства включаемые в систему должны соответствовать следующим требованиям:

- Иметь собственный контролер;
- Подключаться к сети WiFi;
- Иметь родной сервис управления от производителя включенный в список сервисов IFTTT.

3.2. Аппаратные требования.

Для подключения устройств к системе и дальнейшего управления ими

- Необходим персональный компьютер с настроенным Интернет доступом или смартфон
- Домашний веб сервер с постоянным доступом в Интернет или сторонний веб хостинг

3.3. Указание системного программного обеспечения (операционные системы, браузеры, программные платформы и т.п.).

Персональные компьютеры под управлением ОС Windows, браузер (предпочтительно Google Chrome). Смартфон на базе операционной системы Android версии 4.4 и выше или IOS версии 10.3.3 и выше.

3.4. Указание программного обеспечения, используемого для реализации.

Операционная система Windows, редактор исходного кода Visual Studio Code.

3.5. Форматы входных и выходных данных.

Элементы разрабатываемой системы сообщаются между собой путем отправки http запросов, информация передается в виде JSON сообщений.

3.6. Источники данных и порядок их ввода в систему (программу), порядок вывода.

Данные (команды управления) вводятся пользователем через веб интерфейс.

3.7. Порядок взаимодействия с другими системами, возможности обмена информацией.

Порядок взаимодействия и интеграция системы со сторонними комплексами домашней автоматизации настраивается дополнительно через использование веб технологий.

3.8. Меры защиты информации.

Используются стандартные меры защиты информации и средства разграничения прав доступа к компонентам комплекса, предусмотренные производителями используемых сервисов и устройств.

4. Требования к пользовательскому интерфейсу.

4.1. Общая характеристика пользовательского интерфейса.

Интерфейс должен быть интуитивно понятным и комфортным для пользователя системы.

4.2. Особенности ввода информации пользователем, представление выходных данных.

- Формат вводимых данных должен быть удобным и комфортным для пользователя;
- Система должна осуществлять проверку корректности введенных данных.

5. Требования к документированию.

5.1. Перечень сопроводительной документации.

Техническое задание – главный документ, сопровождающий генератор БРС и отражающий все характеристики и требования к разрабатываемой системе.

5.2. Требования к содержанию отдельных документов.

Техническое задание – исходный документ для проектируемого объекта, содержащий технические требования, предъявляемые к объекту и исходные данные для его разработки; в документе указывается информация о назначении объекта, области его применения и сроках выполнения.

6. Порядок сдачи-приемки продукта.

Продукт считается принятым при его положительной оценке экспертами-преподавателями вуза.

Глава 2. Разработка программно-аппаратного комплекса

2.1 Сервисы и инструменты используемые в разработке

Разрабатываемая система представляет из себя несколько модулей взаимодействующих между собой посредством http запросов или web-хуков. Следующая схема (Рис. 1) демонстрирует эти модули и дает им краткое описание.

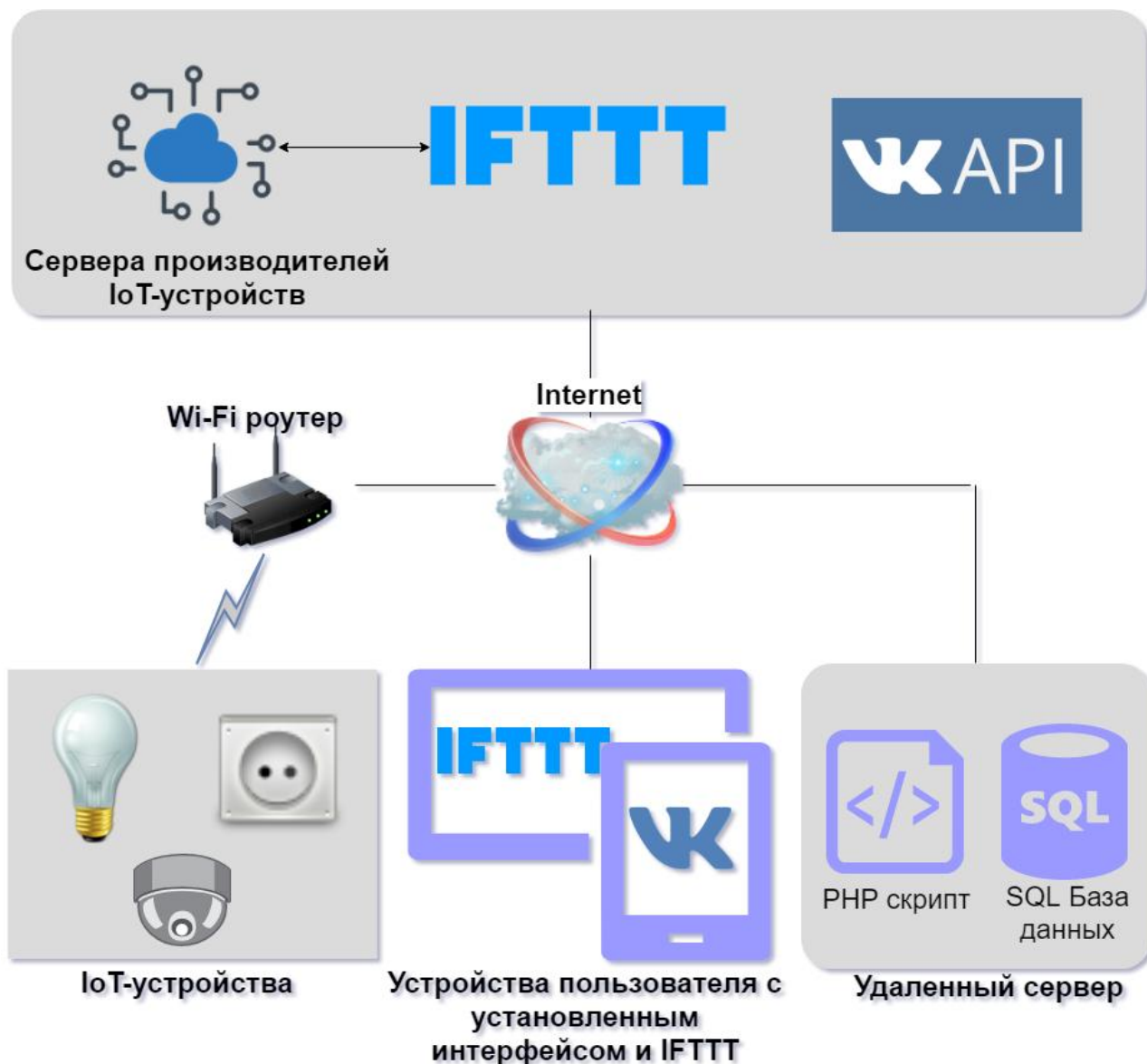


Рис. 1. Общая схема системы

Процесс разработки можно разделить на несколько этапов, каждый из которых подробно описан ниже.

2.1.1 Разработка интерфейса

В качестве интерфейса для управления домашними умными устройствами был выбран чат-бот в социальной сети «ВКонтакте».

В настоящее время чат-боты как современные инструменты коммуникаций стали широко использоваться во многих сферах жизнедеятельности человека с целью установления контакта с пользователями сети Интернет. Наибольшую популярность чат-боты получили, когда началось их использование в мессенджерах и социальных сетях (к примеру, в Telegram, Viber, Facebook, ВКонтакте). Благодаря новому инструменту виртуальной коммуникации появилась возможность узнавать о специальных предложениях онлайн, получать рассылку свежих новостей и специальных предложений в сфере товаров и услуг, совершать более сложные операции. В настоящее время наибольшее значение чат-боты имеют в формировании такой модели поведения, которая будет максимально приближена к человеческой.

Под чат-ботами в общем смысле обычно понимают специальные программы, осуществляющие Интернет-общение свою работу чаще всего с одним или несколькими пользователями, используя искусственный интеллект. Современный вид Интернет-коммуникаций способен выступать в качестве виртуального собеседника и повторять и воспроизводить письменный набор знаков человека, предоставляя запрограммированный ответ на заданные действия.

В силу того, что контакт с чат-ботами является не только интересным занятием, но и полезным для человека, с точки зрения получения новой информации и решения потребительских задач, стоит отметить, что Интернет-аудитория приняла их появление доброжелательно. [27]

Интерфейсные возможности программы достаточно просты и понятны. Алгоритм работы чат-бота основан на заранее загруженных шаблонах, командах и последующем выполнении каких-либо программных действий в зависимости от поступившего сообщения.

Понятие «чат-бот» пришло из английского языка, само слово состоит из двух частей: to chat – общение в сети Интернет и bot (robot) – сокращенно ро-

бот, из чего следует, что это роботы, предназначенные для осуществления коммуникаций с пользователями в сети Интернет, выполняющие действия согласно заложенному сценарию.

Для создания этого механизма и внедрения его в разрабатываемую систему, необходимо проанализировать возможности бесплатно предоставляемого платформой ВКонтакте VK API.

API ВКонтакте – это интерфейс, позволяющий получать информацию из базы данных vk.com посредством http-запросов к специальному серверу. Удобство заключается в том, что не нужно знать в подробностях, как устроена база, из каких таблиц и полей каких типов она состоит – достаточно того, что API-запрос об этом «знает». Синтаксис запросов и тип возвращаемых ими данных строго определены на стороне самого сервиса[28].

Так например, чтобы получить данные о пользователе с идентификатором 173139578 составляется запрос следующего вида:

`https://api.vk.com/method/users.get?user_id=173139578&v=5.52`

Подробнее обо всех его составляющих:

`https://` – широко распространенный протокол соединения и передачи данных, передача данных по которому, как правило, осуществляется через TCP/IP-соединения[29].

`api.vk.com/method` – адрес API-сервиса.

`users.get` – название метода API ВКонтакте. При помощи методов производятся действия с базой данных, например, такие как удаление или добавление записей.

Все методы разделены на секции. Например, для работы с сообществами используются методы секции `groups`, для работы с фотографиями – `photos`, и так далее.

`?user_id=173139578&v=5.52` – параметры запроса. После названия метода нужно передать его входные данные (если они есть) – как обычные GET-параметры в http-запросе. В данном примере серверу сообщается, что необхо-

димом получить данные о пользователе с `id=173139578` и формат этих данных должен соответствовать версии API 5.52. Входные параметры всегда перечислены на странице с описанием метода.

В ответ сервер вернет JSON-объект с запрошенными данными (или сообщение об ошибке, если что-то пошло не так).

Ответом на запрос будет:

```
{«response»:[{«id»:173139578,«first_name»:»Лев»,«last_name»:»Кошак»}]}
```

Структура ответа каждого метода также строго задана, при работе с API заранее определено, что в поле `id` придет число, а в поле `first_name` – строка и т.д. Все правила изложены на страницах с описанием метода и соответствующих объектов, которые он возвращает в ответе.

VK API предоставляет очень много возможностей, для большинства из которых необходимо зарегистрировать свое приложение. В рамках разработки ведущейся в данной ВКР достаточно ограничиться созданием своего сообщества и активации в нем режима разработчика. После чего необходимо указать версию используемого API (была выбрана 5.52) и указать адрес-ссылку на скрипт PHP. ВКонтакте сгенерирует два секретных ключа [30]:

- Ключ доступа (токен сообщества), который в дальнейшем будет использован в каждом запросе сервера к базе ВКонтакте;
- Ключ подтверждения, который необходимо вывести в своем скрипте в ответ на POST запрос, содержащий JSON:

```
{ «type»: «confirmation», «group_id»: 180130264 }
```

Данное действие производится для подтверждения сервисом нашего сервера.

А так же необходимо создать свой защищенный ключ, который используется для:

- получения ключа доступа по схеме Authorization Code Flow;
- получение ключа доступа в процессе прямой авторизации;
- получение сервисного ключа доступа по схеме Client Credentials Flow;

- получение сервисного ключа доступа в настройках приложения.

Для создания самого чат-бота, выполняющего определенные команды, необходимо воспользоваться Callback API. Это инструмент позволяющий отслеживать активность пользователей в созданном сообществе[31]. После подключения своего сервера, появляется возможность выбора типов событий данные о которых необходимо получать. Структура получаемого объекта в поле `object` зависит от типа уведомления. Тип события “входящее сообщение” `message_new` подходит для общения с ботом и отправки через него команд на сервер.

Настроив Callback API можно приступить к написанию скрипта расположенного на своем сервере, в результате работы которого чат-бот будет передавать на сервер определенный список команд и получать оттуда информацию.

2.1.2 Написание PHP скрипта

Центром, связывающим все элементы разрабатываемой системы, является программный файл (скрипт), написанный на языке программирования общего назначения PHP. Код был составлен на основе анализа литературы, посвященной синтаксису языка PHP [32, 33].

Скрипт принимает запросы от вышеописанного интерфейса, обрабатывает их и по технологии webhooks «общается» с сервисом IFTTT, что изображено на следующей схеме (Рис. 2):

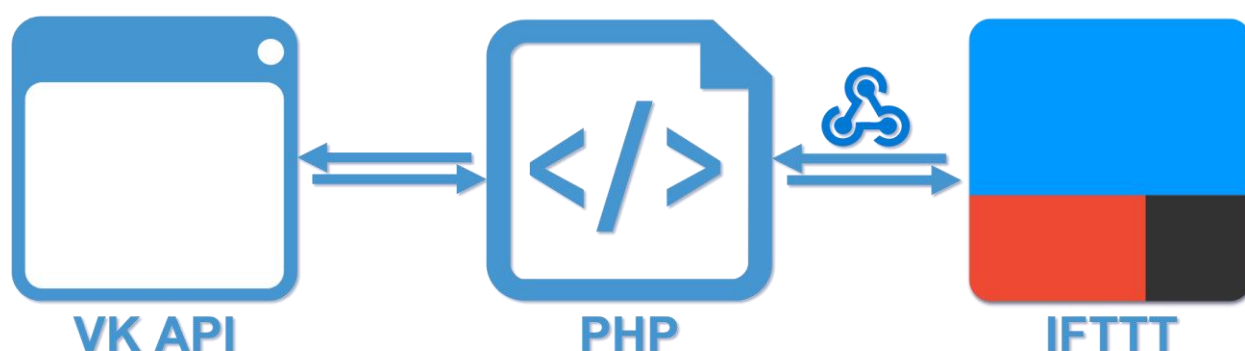


Рис. 2. Взаимодействие элементов управления

Используемые в скрипте методы:

- Конструкция `file_get_contents('php://input')` для получения POST запросов, обернута в метод `json_decode` преобразующий закодированную в JSON строку в переменную[34]. PHP по умолчанию не парсит запросы кроме `application/x-www-form-urlencoded multipart/form-data`, поэтому данные не попадают в суперглобальный массив `$_POST`, а тело запроса нужно парсить самостоятельно например с помощью `php://input`, так как туда поступают необработанные данные из тела запроса.
- Метод `http_build_query` генерирует URL-кодированную строку запроса из предоставленного ассоциативного (или индексированного) массива. С его помощью собирается запрос для отправки на сервер VK API, включающий полученные в ходе выполнения программы данные.

Для расширения возможностей системы было принято решение использовать базу данных, которая хранит состояние устройств и список пользователей, имеющих доступ к управлению.

Для подключения к базе данных и выполнения запросов используется MySQLi, расширение драйвера реляционных баз данных, используемого в языке программирования PHP.

Список методов MySQLi:

- `mysqli($db_host, $db_user, $db_password, $db_database)` – предоставляет связь между PHP и базой данных;
- `mysqli::query (string $query)` – выполняет запрос к базе данных;
- `mysqli_result::fetch_assoc` – извлекает результирующий ряд в виде ассоциативного массива.

На основе анализа литературного источника[35], были составлены следующие SQL запросы:

- `SELECT device_status, device_name FROM devices WHERE device_status=1`
- `SELECT user_vk_id FROM users`
- `UPDATE devices SET device_status = 1`
- `UPDATE devices SET device_status = 0`

Модельное представление работы базы данных выглядит следующим образом (Рис.3, 4).

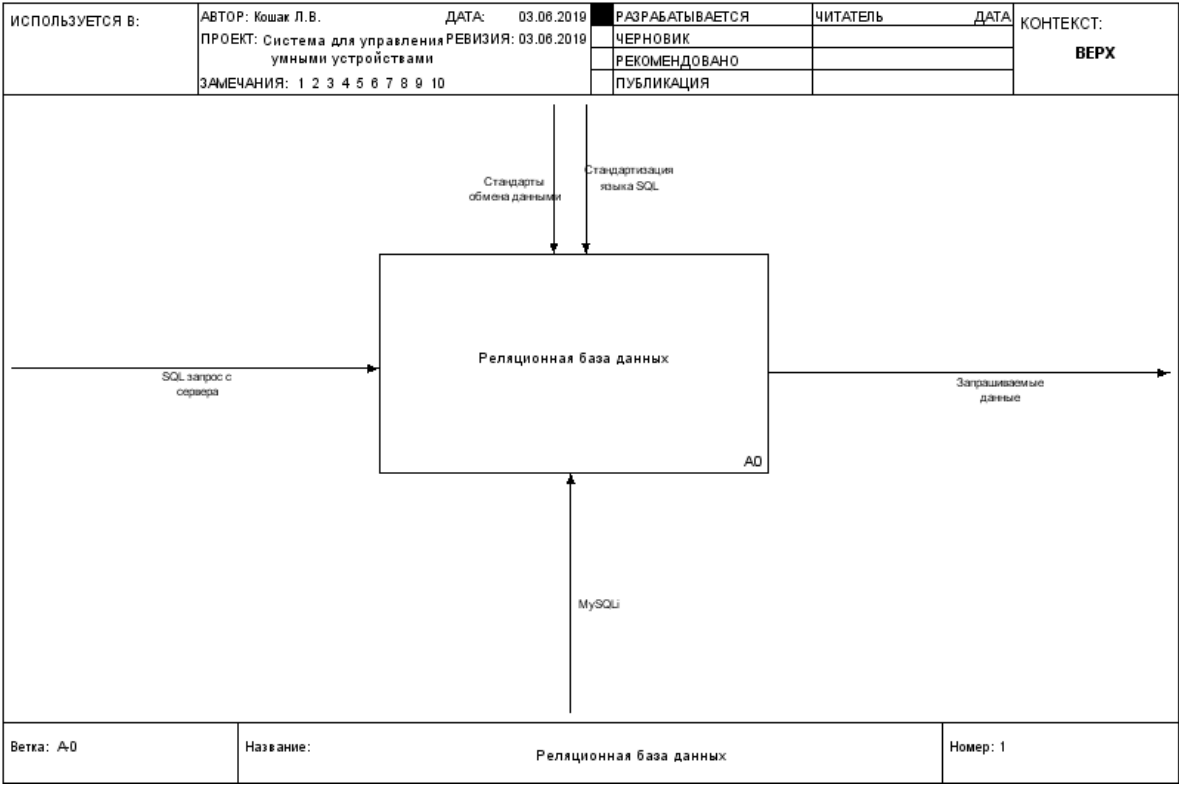


Рис. 3. Уровень A0 IDEF0 диаграммы работы БД

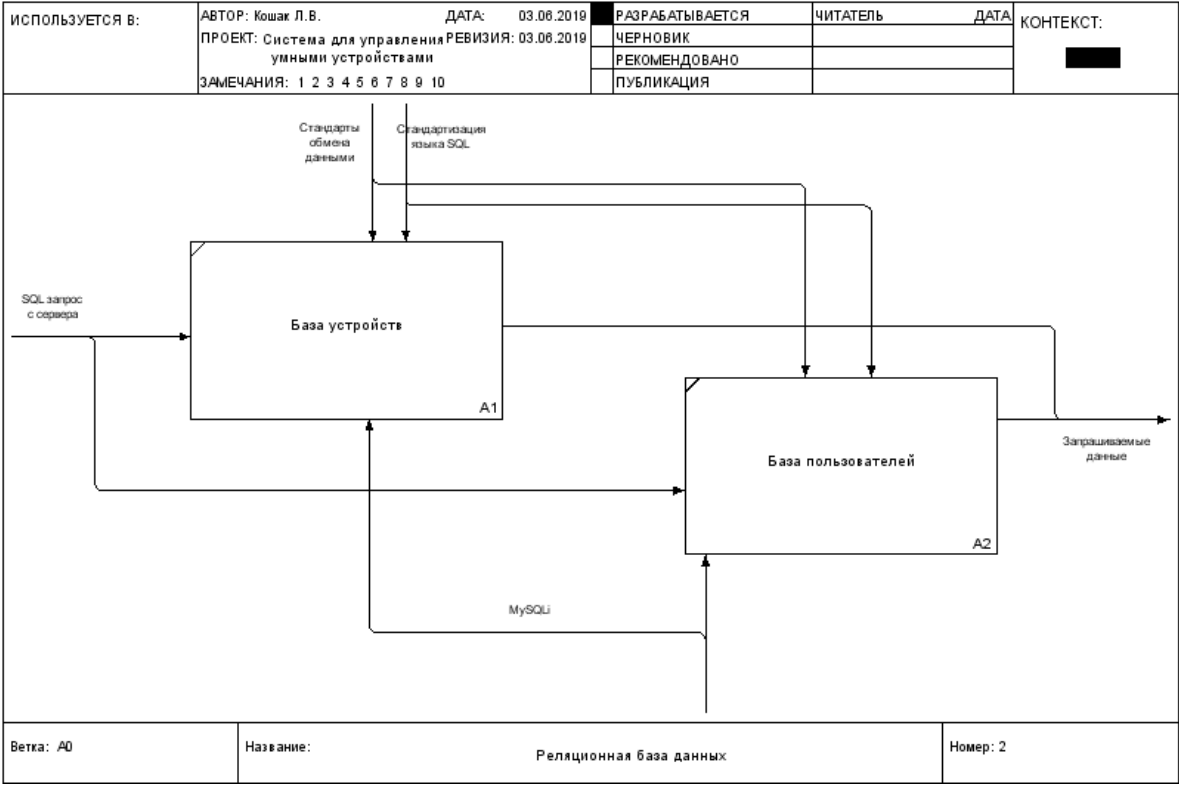


Рис. 4. Уровень A1 IDEF0 диаграммы работы БД

2.1.3 Настройка сервиса IFTTT

Большую часть задач, таких как создание новых сценариев автоматизации используя максимум технических возможностей оборудования, а самое главное – создание моста между разработчиком и серверами производителей умных устройств, в разрабатываемой системе выполняет бесплатный веб-сервис IFTTT.

IFTTT получил свое название от условного выражения программирования «if this, then that» (если это, то это). Компания предоставляет программную платформу, которая соединяет приложения, устройства и сервисы от разных разработчиков для запуска одной или нескольких автоматизаций, включающих эти приложения, устройства и сервисы.

Сценарии существуют самые разные, несколько примеров IFTTT автоматизации:

- Совершая звонок на своем телефоне Android, журнал этих звонков добавляется в электронную таблицу Google.
- При добавлении нового задания в свои задачи Amazon Alexa, они автоматически будут добавлено в приложение iOS Reminders.
- Если международная космическая станция пройдет над указанным домом, уведомление об этом придет на смартфон.

В настоящее время, по данным IFTTT, существует 54 миллиона апплетов.

История IFTTT.

Соучредители Тиббетс и Джесси Тейн основали компанию в 2010 году и официально запустили сервис в 2011 году. Базируясь в Сан-Франциско, IFTTT привлекла 63 млн. долларов венчурного финансирования от инвесторов, в том числе фирмы Andreessen Horowitz. Последний раунд финансирования в размере 24 млн. долл. США, проведенный Salesforce Ventures при участии IBM, Chamberlain Group и Fenox Venture Capital, поможет нанять новых сотрудников и расширить платформу. Тиббетс в настоящее время является генеральным директором IFTTT. В ноябре 2016 года IFTTT расширила свои «рецепты»

(recipes), которые связывали два устройства, приложения или службы, превратив их в апплеты (applets), способные соединять несколько устройств, приложений или служб.

Как устроена работа сервиса

Автоматизация осуществляется с помощью апплетов, похожих на макросы, которые соединяют несколько приложений для запуска автоматизированных задач. Можно включить или выключить апплет с помощью веб-сайта IFTTT или мобильных приложений (и / или виджетов IFTTT мобильных приложений). Также можно создавать свои собственные апплеты или вносить изменения в существующие через удобный и понятный интерфейс IFTTT.

Компании, такие как Ring и BMW, платят IFTTT ежегодную плату за предоставление апплетов. Существует так же партнерство с британским стартап-банком Monzo, которое, помимо прочего, позволяет сотрудникам банка автоматически выводить средства с накопительного счета «дождливый день», когда идет дождь, или «вознаграждать» себя каждый раз, когда они идут в спортзал.

Как правило, разработчики запускают свои «типовые» апплеты, а затем сообщество пользователей «строит то, чего разработчики никогда не ожидали». Апплеты IFTTT могут использовать JavaScript, HTTP запросы, расширенную фильтрацию и другие инструменты для создания новых взаимодействий. Поддержка JavaScript помогает партнерам IFTTT создавать надежные апплеты по сравнению с более ограниченными «рецептами» прошлых лет. Есть возможность создать некоторый пользовательский JavaScript, который будет фильтровать события автоматически, так что, например, апплет, включит несколько источников света в доме, по прибытию после 6 вечера или только свет на крыльце, по прибытию домой до 6 часов вечера.

На сегодняшний день IFTTT имеет более 550 партнерских сервисов. Сообщество 11 миллионов пользователей IFTTT использует более 1 миллиарда апплетов каждый месяц, согласно данным компании.

Начало работы с IFTTT

Сервис прост в использовании. Управление происходит через приложение (для Android или для Apple iOS), создав бесплатную учетную запись и можно сразу же начать работать с автоматизацией. IFTTT предлагает ряд самых популярных типовых и пользовательских апплетов. Коллекции объединяют апплеты для различных платформ – таких как iOS, Android и голосовые помощники – и демонстрируют все: от апплетов для служб новостей и прогнозов погоды до домашней автоматизации. Экран «Мои апплеты» позволяет пользователям управлять включенными в данный момент апплетами и отображает историю использовавшихся ранее. Пользователи могут создавать свои собственные апплеты, комбинируя различные «сервисы» приложения и устанавливая параметры триггера.

IFTTT в основном (но не исключительно) нацелен на потребителей, которые ищут простой способ извлечь больше пользы из своих устройств, услуг или приложений, уделяя особое внимание автоматизации IoT. Но этот сервис не единственный в своем роде.

Microsoft Flow объединяет 195 сервисов для создания средств автоматизации, называемых «потокami», с акцентом на производительность бизнеса. Бесплатный сервис позволяет запускать 750 потоков в месяц и создавать неограниченные потоки. Более продвинутая функциональность стоит 5 или 15 долларов на пользователя в месяц.

Zapier – это бесплатный сервис для потребителей, а также для деловых и опытных пользователей. Его Zaps – это рабочие процессы, которые связывают приложения, такие как Gmail и Dropbox, для запуска и автоматизации действий между ними. Создать двухшаговый Zaps можно бесплатно. Более сложные «запы» требуют ежемесячной платы. За 20 долларов в месяц подключаются три или более шагов ко всем 750 приложениям Zapier. План на 250 долларов в месяц разработан с учетом команд, предлагая неограниченное количество запоров и доступ ко всем премиальным функциям.

Как и любой сервис, IFTTT имеет ряд недостатков:

- Поиск нужных апплетов осложняется тем, что существует множество пользовательских апплетов, которые звучат одинаково.
- Некоторые из пользовательских апплетов «поломаны», но все-равно остаются в общем доступе.
- В работе с IFTTT может возникнуть ситуация, когда пользователь попытавшись настроить апплет, обнаружит, что поставщик услуг не предлагает необходимую функциональность.

Для интеграции вышеописанного сервиса в разрабатываемую систему были созданы апплеты на основе технологии webhooks, геолокации устройства Android и системных действий устройства Android, таких как соединение/рассоединение домашней сети WiFi.

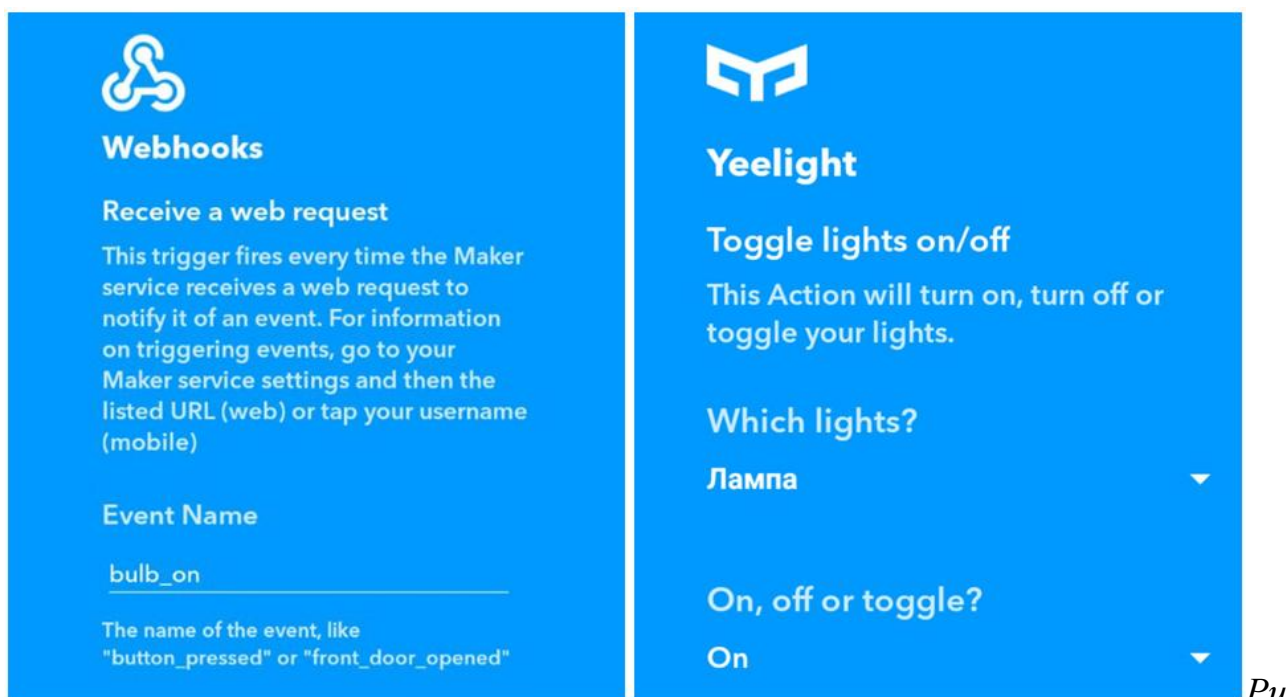
Веб-хук на сервер IFTTT выглядит следующим образом:

<https://maker.ifttt.com/trigger/{event}/with/key/dgHLhMYNwZjmb7c1NYPGAz>

Ссылка состоит из трех основных частей:

- <https://maker.ifttt.com/trigger/> – это адрес сервера;
- `trigger/{event}` – название события, которое пользователь придумывает при создании апплета;
- `with/key/dgHLhMYNwZjmb7c1NYPGAz` – уникальный ключ доступа, сгенерированный сервисом.

Одним из примеров работы такого апплета в разрабатываемой системе является получение запроса с именем события `bulb_on` и последующая отправка команды о включении на умное устройство (Рис. 5).



с.5. Сценарий использования приложения с технологией «Webhooks»

Так же, в ряде сценариев, используются веб-хуки в обратную сторону (с сервера IFTTT на свой сервер). Например, при покидании определенной геолокации пользователем мобильного устройства Android (Рис. 6).

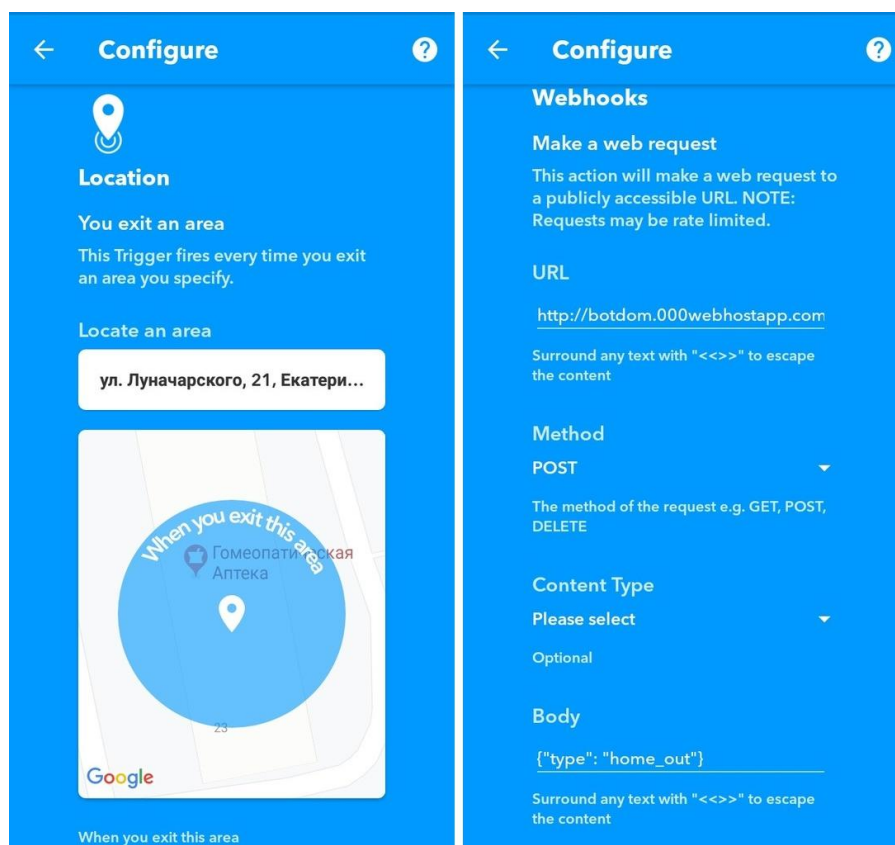


Рис. 6. Приложение на основе отслеживания геолокации

2.1.4 Настройка умного устройства

Для тестирования сценариев взаимодействия пользователя с системой было выбрано устройство «умная лампочка Xiaomi Yeelight LED Smart Light Bulb Color (YLDP02YL)».

Так как Smart LED Color Bulb является обладательницей цоколя E27, то ее можно вкручивать в любые светильники с аналогичным входом.

Процесс подключения состоит из следующих действий: после того, как лампочка была вкручена в цоколь, включить свет, активировать на телефоне Wi-Fi и скачать фирменное приложение «Mi Home», в качестве альтернативного варианта выступает приложение «Yeelight». Первое приложение позволяет объединить новое устройство с другими элементами «умного дома» Xiaomi, при их наличии, но не дает возможности настройки взаимодействия с устройствами других производителей. Второе приложение специфицируется исключительно на управлении конкретной лампочки.

2.2 Описание результатов разработки. Модельное представление процессов

В результате разработки была создана открытая система управления позволяющая добавлять различные умные устройства, имеющие встроенный контроллер автономно связывающий их с серверами производителей. Имеется возможность создавать ряд сценариев взаимодействия устройств друг с другом, с окружающей средой, пользователем и используемым им программным обеспечением.

Модельное представление выполнения команды в системе можно представить в виде IDEF0 диаграммы (Рис. 7, 8)

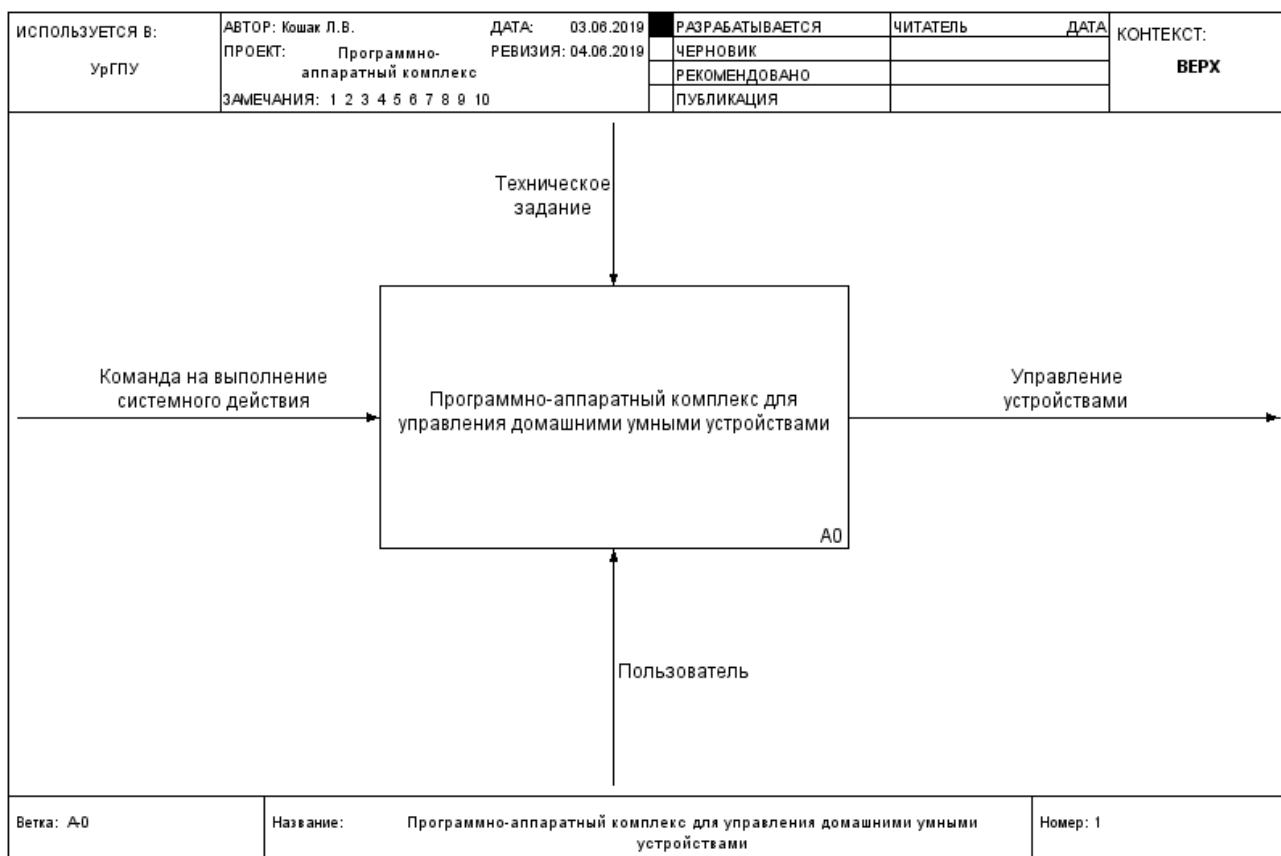


Рис. 7. IDEF0 диаграмма системы

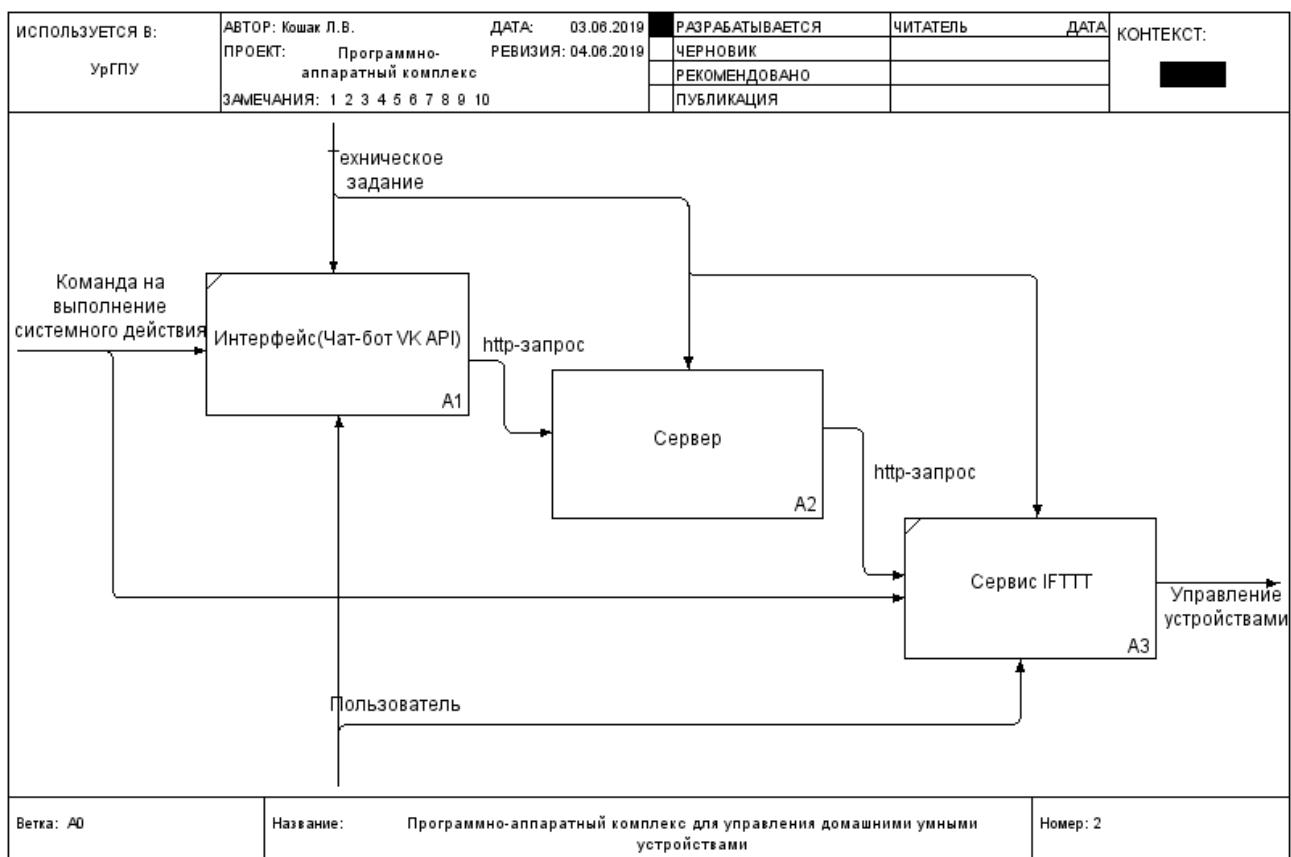


Рис. 8. Декомпозиция ветки A0

Следующая UML-диаграмма «Use Case» изображает возможные варианты взаимодействия пользователя с системой (Рис. 9).

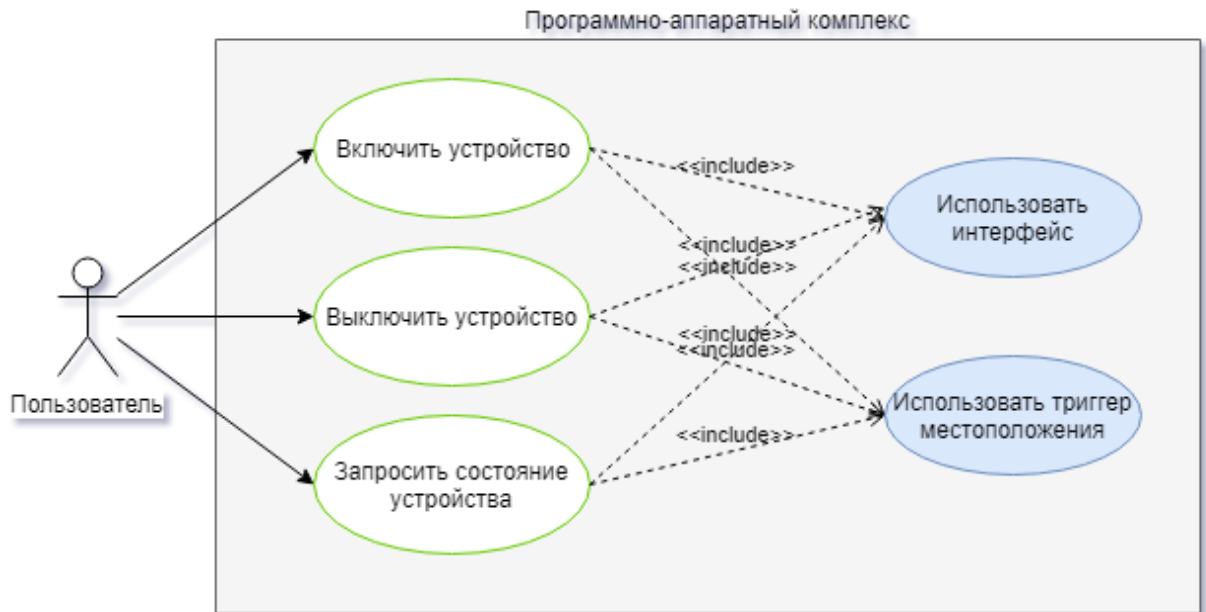


Рис. 9. Use case диаграмма

Список опробованных сценариев с модельным описанием в виде UML диаграмм активностей:

1. Включение/выключение света по набору команды чат-боту «включи свет» (Рис.10)

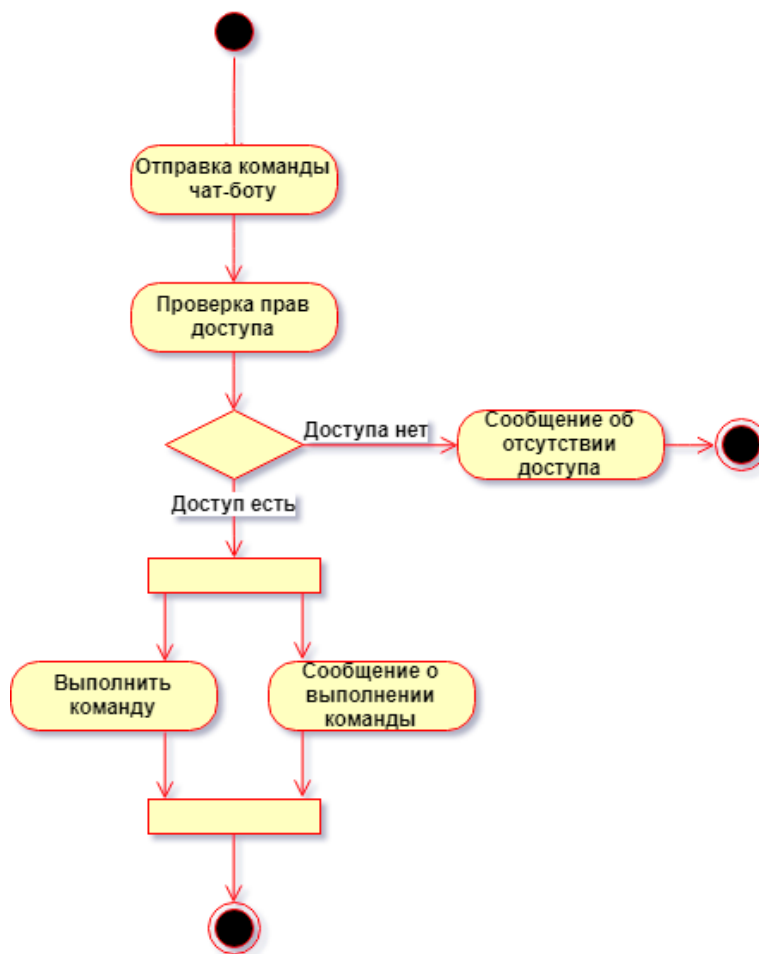


Рис. 10. Сценарий включения/выключения света

2. Отправка в базу запроса на проверку включенных устройств, при покидании дома (проверка проводится путем отслеживания геолокации мобильного устройства пользователя). При работающих устройствах, прислать оповещение об этом (Рис. 11).

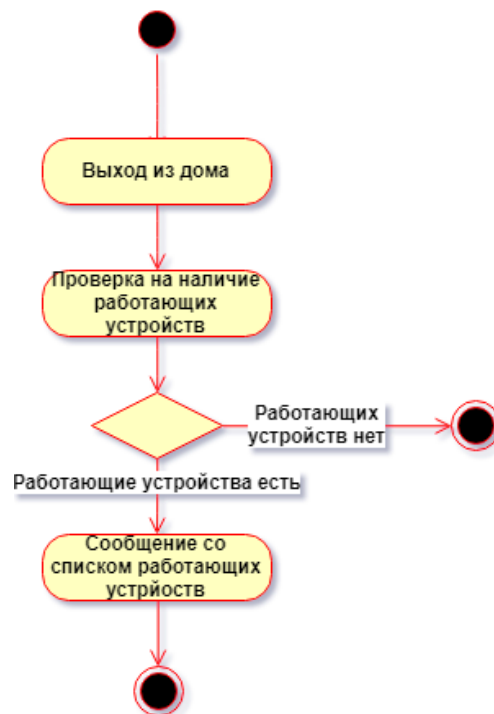


Рис. 11. Сценарий с использованием гео-триггера

3. Отправка в базу запроса на проверку включенных устройств, при покидании дома (триггер срабатывает когда мобильное устройство пользователя отключается от домашней сети Wi-Fi). При работающих устройствах, прислать оповещение об этом (Рис. 12).

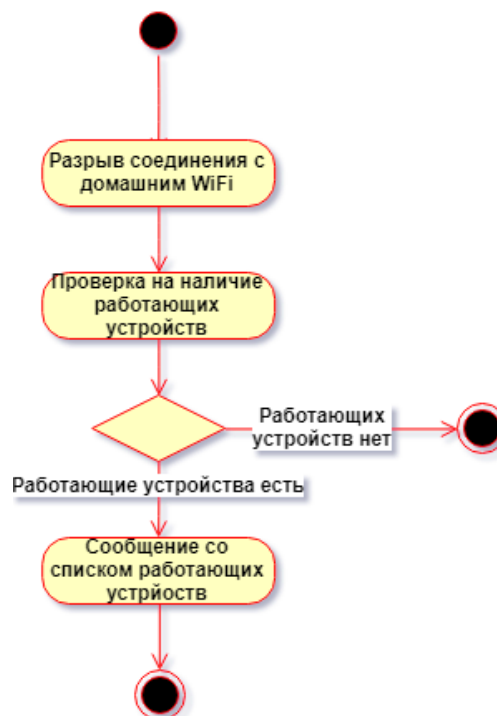


Рис. 12. Сценарий с использованием WiFi-триггера

2.3 Руководство пользователя

Составлен на основе ГОСТ «19.505-79 ЕСПД. Руководство оператора. Требования к содержанию и оформлению» [36]

1. ВВЕДЕНИЕ

1.1. Область применения

Системный продукт предназначен для офисного и домашнего использования

1.2. Краткое описание возможностей

Программно-аппаратный комплекс предоставляет следующие возможности:

- Управление своими домашними умными устройствами с одного интерфейса;
- Добавление новых устройств в систему;
- Создание сценариев взаимодействия устройств между собой и пользователем;

1.3. Уровень подготовки пользователя

Для управления системой пользователь должен владеть минимальным набором знаний и навыков работы с компьютером и электронными девайсами.

1.4. Перечень эксплуатационной документации

Настоящее «Руководство пользователя»

2. НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ

2.1. Назначение системы

Система предназначена для управления домашними умными устройствами разных производителей через один общий интерфейс.

2.2. Условия применения

Устройства включаемые в систему должны соответствовать следующим требованиям:

- Иметь собственный контролер;
- Подключаться к сети WiFi;

- Иметь родной сервис управления от производителя включенный в список сервисов IFTTT.

В качестве устройства управления может выступать любой ПК под управлением ОС Windows, с установленным браузером (рекомендуемо Google Chrome) и доступом в Интернет или мобильное устройство на операционной системе Android версии 4.4 и выше или IOS версии 10.3.3 и выше.

3. ПОДГОТОВКА К РАБОТЕ

3.1. Состав и содержание дистрибутивного носителя данных

Система не требует установки, все элементы управления находятся в сети Интернет и предоставляют доступ как с браузера, так и с приложений на мобильном устройстве.

3.2. Порядок загрузки данных и проверка работоспособности

Для того чтобы запустить систему:

1) Подключите умное устройство к своему смартфону через приложение производителя;

2) Установите приложение IFTTT и добавьте подключенное устройство в список своих устройств;

3) Установите приложение «ВКонтакте» на свой телефон или воспользуйтесь браузерной версией социальной сети

Порядок проверки работоспособности:

1) Откройте диалоговое окно с чат-ботом;

2) Напишите и отправьте слово «Привет», если в ответ придет сообщение «Привет, (Ваше имя)!» – вы подключены к системе;

3) Напишите и отправьте сообщение «Включи (название устройства)», если в ответ придет сообщение «Привет, Ваше_имя!» – вы подключены к системе.

4. ОПИСАНИЕ ОПЕРАЦИЙ

4.1. Описание операции 1

- Наименование:

Запуск работы умного устройства.

- Условия, при соблюдении которых возможно выполнение операции:

Устройство подключено к электросети и сети WiFi, сеть WiFi работает без перебоев.

- Подготовительные действия;
Запустить интерфейс.
- Основные действия в требуемой последовательности;
Написать чат-боту «Включи (название устройства)»
- Заключительные действия;
Отсутствуют
- Ресурсы, расходуемые на операцию.

Отсутствуют

4.2. Описание операции 2

- Наименование:
Проверка системы на наличие работающих устройств.
- Условия, при соблюдении которых возможно выполнение операции:

Устройства подключены к электросети и сети WiFi, сеть WiFi работает без перебоев.

- Подготовительные действия;
Настроить апплет в приложении IFTTT.
- Основные действия в требуемой последовательности;
Отключить мобильное устройство от домашней сети WiFi
- Заключительные действия;
Проверить наличие сообщения со списком работающих устройств.
- Ресурсы, расходуемые на операцию.

Отсутствуют

5. АВАРИЙНЫЕ СИТУАЦИИ

При сбое в работе системы восстановление нормальной работы должно производиться после:

- Перезагрузки неисправно работающих устройств
- Перезагрузки WiFi роутера

Если после перезагрузки и переходе на ручное управление одно из устройств работает неисправно – произвести сброс настроек устройства (сброс настроек каждого отдельного устройства производится согласно инструкции пользователя). Неисправная работа после сброса настроек говорит о поломке устройства.

При ошибках связанных с неисправным подключением к сети Интернет работа и функционирование устройств производится путем ручного управления

При отсутствии доступа к интерфейсу системы, управление устройствами может производиться через интерфейсы производителей или путем настройки в IFTTT альтернативных механизмов управления, например с помощью голосового управления Google.

При неверном вводе команд чат-боту, пользователю придет ответ о том, что такой команды в списке нет.

При попытке управления системой путем отправки сообщений чат-боту с постороннего аккаунта, пользователю придет ответ об отсутствии прав доступа.

6. РЕКОМЕНДАЦИИ ПО ОСВОЕНИЮ

Для успешного освоения системы необходимо иметь навыки использования ПК и мобильных устройств и изучить настоящее «Руководство пользователя».

Заключение

В результате работы, проанализировав основные способы домашней автоматизации, виды построения систем их особенности и недостатки, была выявлена проблема согласованной работы умных устройств разных производителей, связанная, в первую очередь с фрагментированностью рынка этих самых устройств.

Произведя анализ умных устройств и рассмотрев предлагаемые производителями способы управления, был спроектирован и разработан программно-аппаратный комплекс, в основе которого лежит ряд технологий:

- Технологии программирования на языке PHP.
- Технология создания базы данных и работы с ними при помощи языка SQL.
- Современные веб-технологии.
- Сервис типа SaaS (software as a service) IFTTT, позволивший управлять умными устройствами через свой собственный интерфейс.
- Интерфейс же реализован в виде чат-бота на основе технологии VK API.

Работа системы протестирована, а так же подготовлена техническая и сопроводительная документация в виде руководства пользователя.

Таким образом, задачи решены в полном объеме, цель достигнута – разработан программно-аппаратный комплекс для управления домашними умными устройствами, объединяющий умные устройства разных производителей в одной экосистеме.

В перспективе было бы интересно расширить систему путем интеграции множества разных устройств, а так же:

- Внедрить улучшенную систему защиты, для более безопасного использования.
- Управлять умными устройствами непосредственно через «родной» API, по другому, через сервера производителей, без использования вспомогательных сервисов, типа IFTTT.

- Интегрировать систему в другие, профессиональные системы домашней автоматизации. Например, платформу MajorDoMo. В таком случае разработанная система стала бы отличным расширением в виде нового интерфейса и ряда дополнительных сценариев автоматизации, путем добавления триггеров на основе взаимодействия пользователя с мобильным устройством и окружающей средой.

Список информационных источников

1. Новый уровень функционирования системы «умный дом» // cyberleninka.ru URL: <https://cyberleninka.ru/article/v/novyy-uroven-effektivnosti-funktsionirovaniya-sistemy-umnyu-dom> (дата обращения: 7.02.2019).
2. Анализ централизованных и децентрализованных систем автоматизированного управления «интеллектуальным» домом // cyberleninka.ru URL: <https://cyberleninka.ru/article/v/analiz-tsentralizovannyh-i-detsentralizovannyh-sistem-avtomatizirovannogo-upravleniya-intellektualnym-domom> (дата обращения: 9.02.2019).
3. Internet of Things // gartner.com URL: <https://www.gartner.com/it-glossary/internet-of-things/> (дата обращения: 11.02.2019).
4. Найдич А. «Интернет вещей» – реальность или перспектива? URL: <https://compress.ru/article.aspx?id=24290>.
5. Николаев П.Л. Применение облачных технологий в системах умного дома // Молодой ученый. 2014. № 13. С. 37–39.
6. Практика применения систем автоматизации и диспетчеризации в гостиничных комплексах. URL: <http://www.stroim-hotel.ru/files/pdf/metodology/metod/11.pdf> (дата обращения: 17.03.2019).
7. Николаев П. Л. Архитектура интегрированной в облачную среду системы управления умным домом // Программные продукты и системы. — 2015. — № 2 (110). — С. 65–69.
8. Пожалуйста, не забудьте правильно оформить цитату:
9. Кирсанов Н. С. Исследование и синтез системы управления умным зданием // Молодой ученый. — 2018. — №16. — С. 127-130. — URL <https://moluch.ru/archive/202/49486/> (дата обращения: 19.06.2019).
10. SOAP. Версия 1.2. Часть 0: учебник для начинающих. URL: <http://www.w3.org/2002/07/soap-translation/russian/part0.html> (дата обращения: 17.03.2019).

- 11.Фландерс Дж. Введение в службы RESTful с использованием WCF. URL: <http://msdn.microsoft.com/ru-ru/magazine/dd315413.aspx> (дата обращения: 18.03.2019).
- 12.И М.Х., Дун Ф.В. Создаем Web-сервисы RESTful при помощи Spring 3. URL: <http://www.ibm.com/developerworks/ru/library/wa-spring3webserv/index.html> (дата обращения: 18.03.2019).
- 13.Маквитти Л. REST как альтернатива SOAP. URL:http://www.ccc.ru/magazine/depot/07_01/read.html?0502.htm (дата обращения: 18.03.2019).
- 14.Yazar D., Dunkels A. Efficient Application Integration in IP-based Sensor Networks. In First ACM Workshop On Embedded Sensing Systems For Energy-Efficiency In Buildings (BuildSys), Berkeley, California, 2009. URL: <http://dunkels.com/adam/yazar09efficient.pdf> (дата обращения: 19.03.2019).]
- 15.Интеграция и взаимодействие в сети Веб. URL:<http://www.intuit.ru/studies/courses/485/341/lecture/8211> (дата обращения: 19.03.2019).
- 16.Nurseitov N., Paulson M., Reynolds R., Izurieta C. Comparison of JSON and XML Data Interchange Formats: A Case Study. 22nd Intern. Conf. on Computer Applications in Industry and Engineering 2009 (CAINE-2009), 2009, pp. 157–162.
- 17.Введение в JSON // json.org URL: <http://json.org/json-ru.html> (дата обращения: 23.03.2019).
- 18.Технические решения на LogicMachine // <http://logicmachine.net.ru> URL: <http://logicmachine.net.ru/solutions/> (дата обращения: 25.03.2019).
- 19.Ethernet // Национальная библиотека им. Н. Э. Баумана Bauman National Library URL: <https://ru.bmstu.wiki/Ethernet> (дата обращения: 23.03.2019).
- 20.ARCHOS Connected Home: открытая технология автоматизации дома // www.itweek.ru URL: <https://www.itweek.ru/mobile/news-company/detail.php?ID=169852> (дата обращения: 27.03.2019).

21. Домашняя автоматизация с Z-Wave – это доступно // z-wave.me URL: <https://rus.z-wave.me/> (дата обращения: 29.03.2019).
22. Контроллер домашней автоматизации Fibaro Home Center // fibaro.com URL: <https://www.fibaro.com/en/products/home-center-2/> (дата обращения: 29.03.2019).
23. О Wink // wink.in.ua URL: <http://wink.in.ua/help/faq/> (дата обращения: 29.03.2019).
24. SmartThings. Одно интеллектуальное приложение для всех интеллектуальных устройств. // samsung.com URL: <https://www.samsung.com/ru/apps/smartthings/> (дата обращения: 29.03.2019).
25. Wirenboard. Центр документации. // wirenboard.com URL: https://wirenboard.com/wiki/index.php/Центр_документации (дата обращения: 29.03.2019).
26. ГОСТ 34.602-89 Межгосударственный стандарт. «Информационная технология (ИТ). Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы» от июнь 2009 № 661 // М.: Стандартинформ, 2009 с изм. и допол. в ред. от 27.12.2006 г.
27. Смыслова Л. В. Чат-бот как современное средство Интернет-коммуникаций // Молодой ученый. – 2018. – №9. – С. 36-39. – URL <https://moluch.ru/archive/195/48623/> (дата обращения: 01.04.2019).
28. Знакомство с API ВКонтакте. Методы и объекты // vk.com URL: https://vk.com/dev/first_guide (дата обращения: 26.03.2019).
29. HTTP // wikipedia.org URL: <https://ru.wikipedia.org/wiki/HTTP> (дата обращения: 01.04.2019).
30. Получение ключа доступа // vk.com URL: https://vk.com/dev/access_token (дата обращения: 03.04.2019).
31. Callback API // vk.com URL: https://vk.com/dev/callback_api (дата обращения: 04.04.2019).
32. Каллум Хопкинс PHP. Быстрый старт. Екатеринбург: Эксмо, 2014.

33. Хадсон Пол, Лунин С. М. PHP. Справочник. СПб.: КУДИЦ-Пресс, 2006.
34. Функции JSON // php.net URL: <https://www.php.net/manual/ru/function.json-decode.php> (дата обращения: 07.04.2019).
35. Бен Форта Освой самостоятельно SQL. 10 минут на урок. Москва: Вильямс, 2006.
36. ГОСТ «19.505-79 ЕСПД. Руководство оператора. Требования к содержанию и оформлению (с Изменением N 1)» от 1980-01-01 № 74 // 2010 г.
37. Оформитель библиографических ссылок // SNOSKA.INFO URL: <http://snoskainfo.ru/> (дата обращения: 17.02.2019).
38. ГОСТ Р 7.0.83-2013 «Система стандартов по информации, библиотечному и издательскому делу. Электронные издания. Основные виды и выходные сведения» от 15.10.2013 № 1163-ст // Стандартиформ. 2014 г. с изм. и допол. в ред. от 12.09.2018.
39. ГОСТ 2.105-95. Межгосударственный стандарт «Единая система конструкторской документации. Общие требования к текстовым документам» от 08.08.1995 № 426 // Всероссийским научно-исследовательским институтом стандартизации и сертификации в машиностроении (ВНИИНМАШ) Госстандарта России. 1995 г.

Приложения

Приложение 1.

Листинг PHP скрипта

```
<?php
$db_host = 'localhost'; // адрес сервера
$db_database = 'u17631_botdom_db'; // имя базы данных
$db_user = 'u17631_botdom'; // имя пользователя
$db_password = 'KJKSZPJJJ0325HESYOYAMAYEZAKMI111'; // пароль
$connect = new mysqli($db_host, $db_user, $db_password,
$db_database);
$token =
'adb1ea3e0e0597b2a66b0e264df7e80c91b249495e380206bb9de77c0f2056498
c056c4823952837de1cf'; //ключ доступа(токен сообщества)
$key = 'ayezakmiHesoyam123'; //свой секретный ключ
$confirmation_key = 'd9b9a624'; //ключ подтверждения
$data = json_decode(file_get_contents('php://input'));
if(isset($data)){
    switch($data->type){
        case 'home_out':
            $user_id = '173139578';
            $sql = "SELECT device_status, device_name FROM
devices WHERE device_status = 1";
            //$rows = DB::prepare($sql)->execute([1]);
            $wd = $connect->query($sql);
            while(($row = $wd->fetch_assoc()) != FALSE){
                $device_name = $row['device_name'];
                $working_devices .= $device_name . " " . PHP_EOL;
            }
            $response = 'Вы вышли из дома, у вас работает:'.
PHP_EOL . $working_devices;
            break;
        case 'confirmation':
            echo $confirmation_key;
```

```

        break;
    case 'message_new':
        $user_id = $data->object->user_id;
        $user_info =
json_decode(file_get_contents("https://api.vk.com/method/users.get
?user_ids=".$user_id."&v=5.52&access_token=".$token));
        $user_name = $user_info->response[0]->first_name;
        $home_owner = FALSE;
        $sql = "SELECT user_vk_id FROM users";
        $uvkid = $connect->query($sql);
        while(($row = $uvkid->fetch_assoc()) != FALSE){
            if($row['user_vk_id'] == $user_id)
                $home_owner = true;
        }
        if($home_owner == FALSE){
            $response = "Привет, ".$user_name.", ты не
хозяин этого дома!";
            break;
        }
        $message = $data->object->body;
        switch($message){
            case 'Привет' :
            case 'привет' : $response = "Привет,
".$user_name."!";
            break;
            case 'Включи свет':
            case 'включи свет':
                $response =
file_get_contents('https://maker.ifttt.com/trigger/bulb_on/with/ke
y/dgHLhMYNwZjmb7c1NYPGAz');
                $sql = "UPDATE devices SET device_status = 1";
                $connect->query($sql);
                break;
            case 'Выключи свет':

```

```

        case 'выключи свет':
            $response = file_get_contents(
'https://maker.ifttt.com/trigger/bulb_off/with/key/dgHLhMYNwZjmb7c1NYPGAz');
            $sql = "UPDATE devices SET device_status = 0";
            $connect->query($sql);
            break;
        case 'мой айди':
        case 'мой id':
        case 'id':
        case 'Id':
        case 'ид':
        case 'айди': $response = $user_id."";
            break;
        default : $response = "Я не знаю такой команды";
    }
    break;
}
}

if(isset($response)&&isset($user_id)){
    $request_params = [
        'message' => $response,
        'user_id' => $user_id,
        'access_token' => $token,
        'v' => '5.0'
    ];
    $get_params = http_build_query($request_params);

    file_get_contents('https://api.vk.com/method/messages.send?'.
    $get_params);
    return false;
}

?>

```

Листинг базы данных SQL

```
-- phpMyAdmin SQL Dump
-- version 4.8.5
-- https://www.phpmyadmin.net/
--
-- Хост: localhost:3306
-- Время создания: Июн 20 2019 г., 09:14
-- Версия сервера: 10.3.14-MariaDB
-- Версия PHP: 7.3.2

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- База данных: `id9638524_botdom_db`
--
--
-- Структура таблицы `devices`
--
CREATE TABLE `devices` (
  `id` int(10) NOT NULL,
  `device_name` varchar(30) COLLATE utf8_unicode_ci NOT NULL,
  `device_status` varchar(10) COLLATE utf8_unicode_ci NOT NULL
```

```

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
--
-- Дамп данных таблицы `devices`
--
INSERT INTO `devices` (`id`, `device_name`, `device_status`) VAL-
UES
(1, 'Лампочка Xiaomi Yeelight', '0');

-----
--
-- Структура таблицы `users`
--
CREATE TABLE `users` (
  `id` int(10) NOT NULL,
  `user_name` varchar(60) COLLATE utf8_unicode_ci NOT NULL,
  `user_vk_id` int(30) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
--
-- Дамп данных таблицы `users`
--
INSERT INTO `users` (`id`, `user_name`, `user_vk_id`) VALUES
(1, 'Лев', 173139578);
--
-- Индексы сохранённых таблиц
--
--
-- Индексы таблицы `devices`
--
ALTER TABLE `devices`
  ADD PRIMARY KEY (`id`);
--
-- Индексы таблицы `users`
--
ALTER TABLE `users`

```

```

    ADD PRIMARY KEY (`id`);
--
-- AUTO_INCREMENT для сохранённых таблиц
--
--
-- AUTO_INCREMENT для таблицы `devices`
--
ALTER TABLE `devices`
    MODIFY `id` int(10) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;
--
-- AUTO_INCREMENT для таблицы `users`
--
ALTER TABLE `users`
    MODIFY `id` int(10) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

```