

Актуальность выбранной темы:

Огромная часть жизни современного человека уделена повседневной деятельности. Согласно исследованиям сотрудников Института социальной политики НИУ ВШЭ [1], люди уделяют домашней работе порядка 3 часов в сутки. 1095 часов в год на выполнение рутинных обязанностей по дому. Или, если продолжить расчёты, 76650 часов в среднем за жизнь.

Закономерным вопросом будет: «Как сократить время, затрачиваемое на домашние дела, без потери качества самой жизни?». Переложить часть обязанностей с себя на автоматизированную систему – умный дом, или сделать контроль за многочисленными приборами и управление ими более доступными и энергоёмкими для человека. Умный дом – это единая система управления устройствами: вентиляцией, водоснабжением, безопасностью, - постоянно окружающими нас в повседневной жизни. Умный дом является частью более широкой области под названием – Internet Of Things (IoT) или Интернет вещей. Интернет вещей — это концепция сети передачи данных между физическими объектами («вещами»), оснащёнными встроенными средствами и технологиями для взаимодействия друг с другом или с внешней средой [2].

Однако, для среднестатистического пользователя в нашей стране, порог вхождения в большинство систем управления умным домом - выше среднего. Это послужило основанием для выбора темы дипломного проекта.

Цель дипломного проекта: спроектировать веб-интерфейс для системы управления умным домом, который будет прост в управлении.

Задачи дипломного проекта: в соответствии с поставленной целью в процессе работы были поставлены следующие задачи:

1. Спроектировать и создать модуль умного дома – «умная розетка» для тестирования системы.
2. Разработать базовый дизайн веб-интерфейса
3. Спроектировать веб-интерфейс для объединения всех модулей умного дома и их управления.

ПОСТАНОВКА ЗАДАЧИ

Описание предметной областиИнтернет вещей является концепцией, сформулированной в 1999 году в Массачусетском технологическом институте. Концепцией взаимодействия физических объектов по средствам цифрового обмена данными. В отличие от стандартной модели, где устройство привязано к пользователю и способно обмениваться информацией только с ним, в идею интернета вещей заложена возможность объектам (вещам) обмениваться данными между собой, образуя единую сеть. С этой технологией связаны как большие надежды, так и серьезные риски безопасности, связанные с всепроникающей тенденцией превращения обычных вещей в интернет-узлы для передачи данных. Таких как: автомобиль, бытовая и электронная техника, мебель и даже товарные упаковки.

Для удобства и более четкой классификации, основатель Европейского совета по «Интернету вещей», ведущий эксперт в области цифровизации и автор концепции Интернета вещей Роб Ван Краненбург ввел четыре уровня Интернета вещей по охвату области:

1. BAN (Body Area Network) — все, что так или иначе находится на человеке: умные часы, футболки, кроссовки, очки и так далее.
2. LAN (Local Area Network) — по сути это умный дом: различные устройства в доме, объединенные в одну сеть.
3. WAN (Wide Area Network) — умные города: общественный транспорт, объединенный в одну сеть и имеющий выход в интернет; электростанции, объединенные в одну сеть и автоматически перераспределяющие нагрузку и так далее.
4. VWAN (Very Wide Area Network) — умная планета, где каждое устройство может взаимодействовать с любым другим.

В данном распределении просматривается закономерность объединения мелких сетей в более крупные, до тех пор, пока они не сольются в единую глобальную сеть обмена данными.

Различные независимые сети на одном уровне объединяются с использованием защищенных протоколов шифрования, аналитики и контролем за передаваемыми данными. Например, на уровне WAN представляющем собой умные города, возникнет необходимость в связующем звене между обособленными системами: транспортными, энергетическими, бизнес, образовательными и другими. Данная идея хорошо передана на рисунке 1.

Умный дом являет собой совокупность автоматизированных сценариев, а также «умных» вещей, связанных в единую сеть. В представленной иерархии, он занимает второй уровень – LAN (Local Area Network) или зону локальной сети. Это посредник между «умными» объектами в доме и человеком, получающим всю необходимую информацию в удобном виде: графика, кнопки переключения и так далее, - а также в удобном формате вывода: на персональном компьютере, телефоне, телевизоре или любом другом устройстве с дисплеем для отображения данных.

Для упрощения настройки подобных соединений, существуют платформы с интуитивно-понятным интерфейсом и быстрой установкой, что обеспечивает низкий порог вхождения.

Функции системы

Существует ряд требований, обеспечивающих будущую работоспособность создаваемой системы:

Масштабируемость. Система должна быть гибкой при добавлении новых устройств в сеть.

Зона покрытия. Должна охватывать весь участок дома.

Надежность соединения. Система управления должна обеспечивать бесперебойную работу для исправного функционирования всех устройств. И должна быть независимой от внешних устройств, подключенных к ней.

Простота управления. Порог вхождения для новых пользователей должен быть приемлемым.

Технологии и элементная база

Система умного дома представляет из себя связку интеллектуальных устройств, обменивающихся данными.

Следовательно, для анализа системы управления, необходимо создать простейшую сеть объектов (вещей) способных передавать информацию в эту систему. Для создания подобных устройств нужна элементная база:

Микроконтроллеры

Микроконтроллер – это микросхема, предназначенная для управления электронными устройствами. Типичный микроконтроллер сочетает на одном кристалле функции процессора и периферийных устройств, содержит ОЗУ и (или) ПЗУ.

Строго говоря, это однокристальный компьютер, способный выполнять относительно простые задачи, такие как: открывания дверей, включения полива газонов, получение данных с датчиков, форматирование полученных данных для последующей отправки и многое другое.

Именно благодаря тому, что микроконтроллер способен выполнять столь разнообразный функционал, потребность в них ежегодно повышается, а производство во много раз превышает производство процессоров. Вторым фактором их распространенности является нужда производителей в относительно недорогом и сбалансированном модуле для встроенных систем, игрушек, домашней технике, где встраивание процессора существенно повысит итоговую стоимость продукта.

В ходе работы микроконтроллер считывает команды из памяти или порта ввода и исполняет их. Что означает каждая команда, определяется системой команд микроконтроллера. Система команд заложена в архитектуре микроконтроллера и выполнение кода команды выражается в проведении внутренними элементами микросхемы определенных микроопераций.

микроконтроллеры дают возможность гибко управлять различными электронными устройствами. Некоторые модели микроконтроллеров настолько мощны, что могут непосредственно переключать реле (для включения и выключения лампочки, например).

В данной дипломной работе будет использоваться микроконтроллер ESP8266 установленный на плату NodeMcu v3 Lua Wi-Fi. Данная плата с микроконтроллером была выбрана ввиду невысокой стоимости, а также возможности подключиться к беспроводной сети и передавать данные на устройства через Wi-Fi или превратить саму плату в точку доступа, для подключения модулей к ней напрямую.

Датчики

Для проверки работы, системы управления умным домом будут получать информацию с датчиков и, обрабатывая ее, отображать в понятном человеку виде через пользовательский интерфейс.

Датчик температуры и влажности будет замерять показатели в комнате и передавать их на плату NodeMcu, а та в свою очередь, подключаясь к системе управления, отобразит полученные данные на локальном сервере. Был выбран датчик DHT11 представленный на рисунке 2.

Датчик движения. Принцип действия основан на регистрации изменения инфракрасного (ИК) излучения, вызванного перемещением или деятельностью человека. По физической природе видимый свет и ИК излучение одинаковы. ИК излучение также можно сфокусировать линзой, как обычный свет. При попадании ИК излучения на фотозлемент, расположенный внутри датчика, он меняет свои параметры. При комнатной температуре в видимом свете тела не светятся, а в ИК диапазоне – сияют, что отчетливо видно на рисунке 3.

Для данного проекта был выбран модуль инфракрасного датчика движения HS-SR501 представленный на рисунке 4.

Релейный модуль

Реле — электромеханическое устройство, предназначенное для коммутации электрических цепей, цепей сигнализации и управления.

Электромагнитное реле представляет из себя катушку. Она состоит из основания из немагнитного материала, на которое намотан медный провод, как правило покрытый диэлектрическим лаком.

При подаче напряжения на катушку происходит втягивание металлического сердечника, связанного с толкателем, который приводит в движение контакты.

Данное реле будет использоваться в «умной» розетке. NodeMcu будет принимать команды от системы управления и передавать на реле через pin IN, та в свою очередь будет включать или выключать питание розетки. Реле, используемое в этом проекте показано на рисунке 5.

AC/DC преобразователь Преобразователь представляют из себя устройство, предназначенное для преобразования входного напряжения. Он может повышать или понижать его, преобразовывать постоянный электроток в переменный и наоборот. Соответственно, принцип функционирования оборудования зависит от его типа. Для создания «умной» розетки понадобится понижающий преобразователь с входным напряжением 220V и выходным 5V для питания управляющей платы NodeMcu от розетки. На рисунке 6 представлен такой преобразователь.

ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ МОДУЛЯ УМНОГО ДОМА В результате выполнения данного модуля, будет спроектирована розетка, управляемая пользователем удаленно. Схема представлена на рисунке 7. Блоком управления является плата NodeMcu. Питание подается от обычной розетки через вилку с заземлением. Напряжение от розетки преобразуется на понижающем ac/dc преобразователе со стандартных 220V до 5V, подходящих для питания платы. Управляющий модуль контролирует работу 1-канального реле. Реле контролирует подачу тока в розетку, к которой будет подключаться нагрузка в виде настольной лампы.

Платформа и язык программирования

Для написания кода была выбрана интегрированная среда разработки Arduino IDE, предназначенная для создания и загрузки программ на Arduino-совместимые платы, а также на платы других производителей. Интерфейс показан на рисунке 8. В данной IDE, в качестве языка программирования, используется стандартный C++ с упрощающими написание кода особенностями. Интерфейс программы представлен на рисунке 8.

Подключение датчиков Для фиксации данных, получаемых с датчиков, будет использована вторая плата NodeMcu аналогичная первой. Питание подается через micro-usb порт от обычной розетки с помощью переходника. К плате подключается датчик движения и датчик температуры и влажности. Схема подключения DHT11 показана на рисунке 11. Для тестирования работы датчиков был написан код, представленный в приложении Б.1. В результате выполнения программы, данные передаются на локальный сервер, обновляясь с помощью ajax запросов. Ajax – это подход к построению интерактивных пользовательских интерфейсов веб-приложений, заключающийся в «фоновом» обмене данными браузера с веб-сервером. В результате при обновлении данных веб-страница не перезагружается полностью, и веб-приложения становятся быстрее и удобнее.

Датчики, подключенные к плате показаны на рисунке 12.

Плата NodeMcu использует домашний wi-fi для создания локального сервера. Таким образом, все девайсы, находящиеся в одной сети с NodeMcu, получают доступ к сайту, где в виде наглядного интерфейса отображаются данные: температура и влажность комнаты, движения в квартире. Смотреть статистику можно как с телефона – рисунок 13, так и с компьютера – рисунок 14.

ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ВЕБ-ИНТЕРФЕЙСА СИСТЕМЫ При проектировании поведения интерфейса был использован принцип цифровой доски или whiteboard: подключенные датчики визуально отображаются в виде прямоугольных блоков с информацией - фреймов. Фреймы могут быть помещены в любое место в пределах доски. Также они могут растягиваться на любую длину и ширину, чтобы вместить весь содержащийся внутри них контент. Один фрейм передает информацию только с одного подключенного модуля. Удаление отдельных фреймов не влияет ни на работу системы в целом, ни на отдельные ее части. Каждый фрейм содержит необходимый минимум информацию о подключенном модуле для корректной работы.

Рассмотрим систему более подробно.

Внешний вид интерфейса Один из примеров визуального оформления интерфейса представлен на рисунке 15.

Пользователь может изменить общий вид с помощью внутренних настроек, выбрав один из трех имеющихся стилей: классический, темный, светлый.

Переключение осуществляется при помощи radio-button во всплывающей вкладке настроек (рисунок 18). При подтверждении выбора, информация об id выбранного варианта отправляется в php файл, который отправляет sql-запрос в базу данных для обновления имеющейся в ней записи style в таблице pages. При загрузке html-страницы, в эту же таблицу отправляется запрос SELECT для получения данных о текущем стиле веб-страницы (рисунок 19). В соответствии с полученным значением, загружается тот или иной css файл.

При изменении стиля на одной странице, он сохранится и будет отображаться на всех.

Фреймы Фрейм — отдельный законченный HTML-документ, который вместе с другими HTML-документами может быть отображён в окне браузера. Фреймы по своей сути очень похожи на ячейки таблицы, однако более универсальны.

В данной работе, фреймы используются в качестве связующих звеньев между платами podemsi, с подключенными к ним устройствами, и веб-интерфейсом, располагающимся на локальном веб-сервере управляющего блока (компьютера).

Принцип работы:

На модуле (плате podemsi), который планируется подключить к системе, поднимается локальный сервер с прописанной в скетче html-страницей. Данная страница задает функционал конкретного модуля: переключение состояний реле при нажатии на html кнопку, отображение значений датчиков и т.п. Вся стилизация происходит на стороне модуля.

Плата переводится в Wi-Fi режим станции (STA) и подключается к существующей wi-fi сети, к которой уже подключен управляющий блок. Важно, чтобы все модули находились в единой, с блоком управления, сети.

В скетче модуля, при подключении к сети, прописывается следующая строка: Serial.println(WiFi.localIP()). При успешном подключении, она выведет в монитор порта IP адрес платы, который понадобится для добавления нового фрейма.

Добавив папку с проектом в каталог доменов локального сервера, по адресу http://openhome.espl будет доступен веб-интерфейс.

Нажав на кнопку «+» в правом верхнем углу экрана, откроется окно для создания нового фрейма (рисунок 20). Пользователю необходимо будет придумать frameID устройства и указать ip-адрес платы, который он получил на шаге 3. При совершении всех перечисленных действий, в веб-интерфейс автоматически добавится новый фрейм, ссылающийся на страницу подключенной платы. Если модуль подключен к питанию, страница должна быть доступна в локальной сети wi-fi и фрейм отобразит содержимое, будь то кнопка управления реле (рисунок 21) или визуально оформленные данные, получаемые с датчиков (рисунок 22).

Если плата не подключена к питанию, ip-адрес будет ссылаться фрейм на несуществующую в данный момент страницу, что не нарушит работы остальных фреймов и веб-интерфейса в целом.

Фрейм, ссылающийся на несуществующую страницу, будет поддаваться масштабированию и перемещению как все остальные блоки.

При подключении модуля к сети и перезагрузке страницы, фрейм автоматически найдет и отобразит данные с платы.

Пример выключенного фрейма представлен на рисунках 23.

Фреймы, добавляемые пользователем, можно перемещать и масштабировать (изменять ширину и высоту) в пределах родительского блока content. При последующей загрузке страницы, фреймы останутся в том же положении и в тех же пропорциях.

Перемещения и изменения размеров фреймов сохраняются в локальном хранилище (LocalStorage) устройства и не требуют выгрузки значений в базу данных, что существенно замедлило бы работу системы или вовсе сделало бы ее невозможной.

На каждом устройстве, имеющем доступ к локальному серверу, веб-интерфейс будет иметь свои уникальные положения и размеры фреймов, заданные пользователем.

Пример изменений размера фрейма относительно других элементов на странице: рисунок 24 и 25.

Рисунок 24 – Начальное положение и размер фрейма «реле»

Данные фреймов, созданных пользователем, выводятся над ними в левом верхнем углу, при наведении на один из них (рисунок 26). Они включают в себя два пункта: url-адрес страницы, на которую ссылается фрейм и frameID, придуманное пользователем при инициализации.

Эти данные будут полезны, при обновлении фрейма. Если пользователь захочет сохранить размер и позиционирование блока, но изменить содержимое, ему будет достаточно использовать функцию update.

При нажатии правой кнопкой мыши на фрейм, выпадет контекстное меню с возможностью: удалить, обновить фрейм (рисунок 27).

Выбрав пункт Change Frame, в этом месте откроется всплывающая форма обновления фрейма (рисунок 28).

Введя новые данные и нажав кнопку update, в базу отправится sql-запрос для обновления последнего нажатого фрейма. Страница обновится и php-файл, получающий из базы все фреймы, принадлежащие данной странице, заново отобразит фрейм, с уже новым именем и ip-адресом заданными пользователем (рисунок 29 и 30).

Комнаты Комнаты, в разрабатываемой системе, представляют из себя дополнительные whiteboard-ы для структурирования фреймов по их местоположению в пространстве. Комнаты способны объединять в себе все фреймы, относящиеся к одной конкретной категории или месту в квартире.

Пользователь сам задает имя комнаты, но предпочтительно исходить из названия мест, в которых располагаются модули: кухня, гостиная, ванная и т.д.

Пример комнат показан на рисунке 31.




Для создания новой комнаты, необходимо нажать на соответствующую иконку в шапке сайта, после чего откроется рорир-форма создания комнаты (рисунок 32). Пользователю достаточно придумать название комнаты и выбрать любую иконку из предложенных. При нажатии на кнопку Add, введенные данные передадутся в php файл, который отправит в базу данных запрос на добавление (INSERT) новой комнаты в таблицу rooms (рисунок 33 и 34). Перед отправкой данных, файл сгенерирует уникальное значение timestamp используя время, в которое был выполнен запрос, с включением миллисекунд и сформирует на его основе уникальный goom_id. Благодаря этому, две или более комнат с одинаковыми именами не будут конфликтовать между собой.

7,8 баллов из 10
по шкале Главреда

165 предложений
2440 слов, 18900 знаков

268 стоп-слов. Основные проблемы: **необъективная оценка**
рекламный штамп **неопределенность** **усилитель** **фичеризм**

Главред помогает очистить текст от словесного мусора, проверяет на соответствие информационному стилю. Расскажите о Главреде всем, кто пишет:

 [Твитер](#)  [Вконтакте](#) 4383  [Гугл-плюс](#)

[О проекте](#) [Блог](#) [Курс](#) [АПИ](#) [Главред](#) [в Фейсбуке](#) и [Твитере](#)

Максим Ильяхов, Анатолий Буров. С 1 декабря 2014 года. Пишите: mail@glvrd.ru

Полезная рассылка о сильном тексте. Уроки, советы и примеры каждую неделю, без спама.