**Разработка веб-интерфейса системы управления умным домом**

**Салаватов М.В.**

**ИВТ 1.1**

**Тезисы для защиты диплома**

**Введение**

Огромная часть жизни современного человека уделена повседневной деятельности. 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.

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

Умный дом – это единая система управления устройствами: вентиляцией, водоснабжением, безопасностью, - постоянно окружающими нас в повседневной жизни.

**Актуальность выбранной темы**, - существующие системы слишком сложны в понимании и использовании. Поэтому, важнейшим пунктом дипломной работы является гибкость системы и ее отказоустойчивость, и простота

**Практическая значимость:** Данная система не привязана к конкретным устройствам и может быть установлена в любом доме, где есть беспроводной интернет.

**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 дипломного проектирования 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.

**ЭЛЕМЕНТНАЯ БАЗА**

**Датчик движения.** Принцип действия основан на регистрации изменения инфракрасного (ИК) излучения, вызванного перемещением или деятельностью человека.

При попадании инфракрасного излучения на фотоэлемент, расположенный внутри датчика, он меняет свои параметры.

**Релейный модуль**. Реле — электромеханическое устройство, предназначенное для коммутации электрических цепей.

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

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

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

**Анализ платформы Blynk**

Работа в Blynk начинается с установки приложения.

Весь процесс интуитивно-понятен: вводится название проекта и выбирается плата, через которую осуществляется взаимодействие устройств с платформой.

После создания проекта, на почту, указанную при регистрации, высылается токен для авторизации. Этот токен нужен для привязки платы к проекту. Его нужно будет указать в исполняемой программе. Далее необходимо наполнить интерфейс.

Blynk обладает богатой библиотекой виджетов: кнопки, таймеры, дисплеи и многое другое.

Завершенные проекты можно опубликовать в качестве отдельного приложения.

**Минусы**

* Ограничения в интерфейсе
* платные дополнения
* Отсутствие гибкой настройки каждого модуля

**Плюсы**

* Простая установка
* Интуитивно-понятный интерфейс
* Большое количество поддерживаемых плат
* Зона покрытия охватывает всю планету, так как запросы передаются на сервер и только потом на устройства.
* Высокая надежность. Платформу поддерживает компания с большими ресурсами в сфере облачного хранилища.

**Анализ Home Assistant в связке с ESPHome**

Home Assistant – это бесплатное программное обеспечение для домашней автоматизации с открытым исходным кодом.

Home Assistant позволяет управлять всеми устройствами, не сохраняя данные в облаке

Основатели проекта делают упор на конфиденциальность и возможность организовать всю систему локально, без передачи данных на сторонние ресурсы. Это возможно благодаря операционной системе Home.io

После ее установки и запуска в VirtualBox, на устройстве пользования будет доступен локальный сервер с данной системой.

**Учет изменений в системе**

Home Assistant ведет постоянный учет изменений получаемых данных. На одном из рисунков видно, что после 16:00 6 апреля, Home Assistant зафиксировал переменную облачность

Home Assistant в связке с ESPHome – это более сложная в установке и настройке, чем Blynk, система. Но вместе с тем, Home.io дает более проработанную структуру с детальной настройкой каждого элемента.

**Плюсы**

* Большое сообщество разработчиков, развивающих проект, способных помочь при возникновении трудностей.
* Удобный интерфейс.
* Автоматическое добавление устройств на главный интерфейс Home Assistant.
* Масштабируемость. Добавление нового узла в систему происходит автоматически после прошивки платы. Это позволяет быстро внедрять новые узлы в систему.
* Ремонтопригодность. Благодаря OTA (Over the Air) дальнейшая перепрошивка устройств осуществляется по воздуху.

**Минусы**

* Относительная сложность установки.
* Необходимы дополнительные программы. (При установке на компьютер – VirtualBox, Python).
* Образ ОС занимает много пространства на диске.

**Проектирование интерфейса и реализации основного функционала**

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.

**Принцип работы:**

* На модуле (плате nodemcu), который планируются подключить к системе, поднимается локальный сервер с прописанной в скетче html-страницей. Данная страница задает функционал конкретного 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 понадобиться для добавления нового фрейма.
* Добавив папку с проектом в каталог доменов локального сервера (OpenServer), по адресу <http://openhome.esp/> будет доступен веб-интерфейс.
* Нажав на кнопку «+» в правом верхнему углу экрана, откроется окно для создания нового фрейма. Пользователю необходимо будет придумать frameID устройства и указать ip-адрес платы, который он получил прописав wifi.localip()

Фреймы можно обновлять и удалять, с помощью контекстного меню (открывается нажатием правой кнопки мыши)

**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.

**Рассказать, как создавать комнаты**

При нажатии на кнопку **Add,** введенные данные передадутся в php файл, который отправит в базу данных запрос на добавление новой комнаты в таблицу.

Перед отправкой данных, файл сгенерирует уникальное значение времени (**timestamp)** используя, как базис, время, в которое был выполнен запрос, с включением миллисекунд и сформирует на его основе уникальный **room\_id**.

Благодаря этому, две или более комнат с одинаковыми именами не будут конфликтовать между собой.

**Рассказать про контекстное меню**

Add Frame (Room)

Delete Frame

Update Frame

**Для настройки работы виджетов необходимо:**

* Зарегистрироваться на сайте openweathermap.org
* На сайте, во вкладке API, выбрать пункт One Call API (нажать кнопку Subscribe)
* Выбрать ценовой вариант Free (бесплатный)

(После этого на email, указанный при регистрации придет письмо с уникальным ключом api. Его нужно сохранить).

* Открыть файл weather.js в корневом каталоге проекта
* В строчку с «appid», после двоеточия, в двойных кавычках, вставить ключ api, пришедший на почту после шага 3.

**Для создания нового модуля умного дома:**

* Скачать и загрузить скетч для управляющей платы, на имеющийся микроконтроллер, предварительно внеся изменения в код, поменяв номера пинов для получения и отправки данных в соответствии с используемой схемой подключения. А также сменив ssid и пароль wi-fi, задав данные домашней сети (при использовании платы в роли станции).
* Загрузить скетчи для каждого модуля в соответствии с выполняемой им задачей (считывания показаний датчиков, переключение реле)
* Проверить работоспособность системы, подключив шлюз к компьютеру через usb и проверив данные в COM-порту.
* Получив ip-адрес из COM-порта, ввести его в поисковую строку любого используемого браузера. Данный адрес является ссылкой на локальный сервер модуля. При успешном отображении html-страницы с данными, можно подключать модуль к интерфейсу.

**Добавление нового модуля в интерфейс.** На данном этапе, должен быть установлен и запущен Open Server с разархивированной в нем папкой проекта. А также создан и проверен новый модуль, который планируется добавить в интерфейс управления.

* Открыть веб-интерфейс в браузере, перейдя по адресу – <http://openhome.esp>
* В открывшейся странице, нажать в правом верхнем углу по кнопке «+».
* Ввести в открывшейся popup-форме frameID – придумать имя устройства для отображения в интерфейсе, и Device IP – ip-адрес устройства, полученный при создании нового модуля (рисунок 53).
* Нажать кнопку **Add** или клавишу **Enter**, после чего страница обновится и новый фрейм автоматически добавиться на доску content.