МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«КРЫМСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ им. В. И. ВЕРНАДСКОГО»
ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ
Кафедра компьютерной инженерии и моделирования

СИСТЕМА УПРАВЛЕНИЯ УМНЫМ ДОМОМ ПО WI-FI С ПОМОЩЬЮ ESP8266

Курсовая работа
по дисциплине «Компьютерные сети»
студента 2 курса группы ИВТ-б-о-201(1)
Мазлова Ивана Денисовича
направления подготовки 09.03.01 «Информатика и вычислительная техника»

Научный руководитель _____
(оценка)
старший преподаватель кафедры
компьютерной инженерии и моделирования _____ Тимофеева С.В.
(подпись, дата)

Симферополь, 2022

АННОТАЦИЯ

«Система управления умным домом по Wi-Fi с помощью ESP8266» Симферополь: ФТИ КФУ им. В. И. Вернадского, 2022. - 28 с., 23 ил., 12 ист.

Объект исследования – NodeMCU на базе ESP8266 как основа создания систем «Умного дома». Описание понятия «Умный дом», процессора ESP8266 и методов работы с ним.

Цель работы – популяризовать системы «Умного дома», создать рабочий макет и реализовать программный код Wi-Fi-сервера для управления устройствами «Умного дома» по локальной сети.

Ключевые слова: «Умный дом», IoT, ESP8266, NodeMCU, Wi-Fi, АСУ, C++, JAVASCRIPT

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Системы «Умного дома», IoT (Internet of Things — Интернета вещей) и АСУ (автоматизированных систем управления) представляют собой разного рода автоматизацию на основе контрольных и активных устройств. Они приобретают всё большую популярность за счет того, что делают нашу жизнь более комфортной, безопасной и информированной. Доступность и функционал таких систем стремительно растут, и то, что раньше казалось невозможным, можно уже сегодня воплотить в жизнь своими руками у себя дома.

Сетевые технологии позволяют информации преодолевать любые расстояния, но в вопросах управления реальными электрическими приборами всё равно встанет необходимость преобразовывать полученную посредством сетевых протоколов информацию в непосредственно управляющие электрические сигналы.

В данной работе будут рассмотрены основные аспекты систем «Умного дома»: применение, преимущества, протоколы связи, используемое «железо» и настройки роутера точки доступа для удобного управления системой.

Основная часть этой работы посвящена описанию практического создания сервера и программному коду, реализованному для него на сервер-стороне (C++) и клиент-стороне (javascript). Детальное описание логики работы сервера сопровождается фрагментами программного кода и скриншотов реализации пользовательского интерфейса. Вместе с работой будет представлен собранный, запрограммированный и полностью рабочий макет сервера для дистанционного управления электроприборами.

1. СЕТЕВАЯ ОРГАНИЗАЦИЯ «УМНОГО ДОМА»

3.1. Умный дом — это удобно

Бытовые заботы часто кажутся нам скучными, но от них очень зависит качество нашей повседневной жизни. Автоматизация повседневных дел может сделать жизнь более приятной и комфортной.

Современные системы «Умного дома» (Smart Home) включают в себя не только бытовые приборы, но и автоматическое управление ими в соответствии с задачами, поставленными перед ними владельцем. Причем это управление можно осуществлять не только непосредственно на месте установки этих бытовых приборов, но и дистанционно.

Прежде чем создавать или модернизировать то или иное устройство для умного дома, нужно определиться со своими целями и задачами.

Системы умного дома могут помочь:

- регулировать микроклимат — температуру и влажность — в помещениях, управляя отоплением, увлажнителями или осушителями воздуха;
- контролировать состав воздуха в помещениях или в наружной атмосфере, считывая показания датчиков газов и пыли, управляя вентиляцией, фильтрами, баллонами и т. д.;
- рационально расходовать ресурсы: электричество, природный газ, воду и прочее;
- управлять освещением, включая его, где нужно в конкретный момент, и отключая там, где необходимость в нем отпадает;
- контролировать шумы и вибрации, если это требуется для соблюдения санитарных или технических норм;
- следить за протечками воды из труб, раковины, ванной, душа, стиральной или посудомоечной машинки;
- обеспечивать пожаробезопасность, следя за задымлением, температурой и газами, характерными для возгораний;

- осуществлять видео-, фото- и иное наблюдение за домом и придомовой территорией;
- вовремя и в нужных объемах поливать домашние растения, кормить и поить домашних животных.

3.2. Безопасность, эргономичность и дизайн систем умного дома

Многие системы умного дома сами по себе предназначены для повышения безопасности жизни в этом доме. Например, такими системами является разного рода сигнализация: от проникновения нежелательных субъектов, от возгораний, от вредных химических веществ, от протекания воды и других жидкостей и т. д. Например, в доме можно продумать автоматическое пожаротушение, отключение электричества от сети или, наоборот, подключение резервного электропитания, если отключается основное.

Желательно, чтобы устройства умного дома были не только безопасными, но и удобными, эргономичными — ведь для удобства они и создаются.

На удовольствие от пользования устройствами умного дома, а также на их продажи может влиять даже их дизайн, эстетика. Это менее значимый фактор, чем безопасность и удобство, однако, всё равно, если есть возможность, то желательно делать такие устройства еще и красивыми.

3.3. Способы монтажа устройств «умного дома»

В зависимости от того, насколько постоянное размещение устройств умного дома хочет пользователь, можно условно выделить стационарные (встроенные или закрепленные), полустационарные и мобильные устройства. У каждого из этих способов есть свои преимущества и недостатки.

3.3.1. Стационарная установка устройств

Системы умного дома можно встраивать еще во время постройки или ремонта дома. В таком случае можно вмонтировать устройства в стены, скрыть их там, чтобы они были малозаметными. Преимуществом такого монтажа является

возможность вписать приборы в общий дизайн дома, расположить их компактно, чтобы они не мешали и чтобы их было почти невозможно случайно задеть. Однако недостатком встраивания приборов является сложность их переделки и переустановки, модернизации.

3.3.2. Полустационарная установка устройств

Полустационарно могут монтироваться устройства среднего или небольшого размера, которые перемещаются не очень часто. Полустационарными могут быть, например, крупные колонки для воспроизведения звука, мониторы, телевизоры среднего размера, микроволновые печи, чайники и т. д., которые можно поставить на пол или на достаточно прочную и устойчивую мебель.

3.3.3. Переносные устройства

Переносные (портативные) устройства — это устройства, которые пользователь может носить с собой или легко перекладывать с места на место. Как правило, это устройства небольшого размера или такие, которые легко и быстро складываются до компактных размеров и раскладываются обратно.

Переносные устройства должны иметь довольно прочный корпус, стойкий к износу и ударам хотя бы средней силы, в отдельных случаях — особо ударопрочный и водостойкий корпус, поскольку такие устройства иногда выпадают из рук, попадают под дождь, а в условиях Крыма могут упасть в море, на камни, засориться пылью.

3.4. Компоненты умного дома

Система управления «умным домом» обычно состоит из следующих основных частей.

Источник питания — сеть электроснабжения или аккумуляторная батарея. Следует помнить, что электричество в сети иногда выключают, поэтому для бесперебойной работы устройств необходимо предусматривать независимые или резервные источники питания. Питание устройств умного дома необходимо

преобразовывать в ту форму, которая требуется этим устройствам (переменный или постоянный ток, соответствующие напряжение и сила тока). Для этого используются соответствующие выпрямители или инверторы, трансформаторы, резисторы и другие компоненты.

Пульт (контроллер) — центр управления «умным домом», снабженный, как правило, индикаторами или мониторами для определения состояния системы и кнопками или рычагами для принятия управляющих сигналов от пользователя.

Датчики (сенсоры, англ. sensors) — устройства, получающие информацию о состоянии среды, в которой они находятся. Современная промышленность выпускает следующие виды датчиков:

- датчики движения/присутствия;
- датчики вибрации (удара);
- датчики дыма;
- датчики утечки природного газа;
- датчики протечки воды;
- датчики открытия и закрытия дверей/окон;
- датчики температуры;
- датчики влажности;
- датчики освещенности;
- сенсорные панели/экраны, которые принимают управляющие сигналы от пользователя.

Сигналы с датчиков передаются на контроллер.

Средства передачи сигналов — провода или проводящая среда, позволяющая передачу разного рода излучений: радиоволн, инфракрасного и т. д.

Актуаторы (от английского слова «act» — действовать), то есть непосредственно **исполняющие устройства**. Их можно в том числе программировать под разные задачи.

3.5. Протоколы связи «умного дома»

Сигналы могут передаваться по одному или нескольким из современных протоколов:

3.5.1. 1-Wire

Магистралью для передачи данных выступает двунаправленная шина, двужильный провод. Один провод используется для питания и передачи данных, второй — для заземления. Все устройства «нанизываются» на один кабель, подобно бусинам на нитке. Такой стандарт прост, недорог и неприхотлив, однако неустойчив к отказам. Стандарт 1-Wire подойдет экономным пользователям, которым достаточно нескольких основных датчиков. Данные могут передаваться на расстояние до 300 м.

3.5.2. X10

Этому стандарту уже несколько десятков лет, но он универсальный и недорогой, поэтому активно используется до сих пор. Для передачи сигнала используется обычная электропроводка. Кроме того, могут использоваться трансиверы, которые ловят сигнал от беспроводных устройств, преобразуют его в нужный формат и передают в электрическую сеть. Эта функция используется для взаимодействия с датчиками и пультами дистанционного управления. Контроллеры программируются через компьютер и дальше работают самостоятельно. Скорость передачи данных такой сетью низкая, с задержкой ¾ секунды. Поэтому с помощью такого протокола трудно организовать сложные схемы взаимодействия.

3.5.3. KNX

Это дорогостоящий стандарт, популярный в Европе. Протокол KNX может использовать витую пару, электрическую сеть или радиоканал. Такая система должна иметь свой источник питания, но может обходиться без центрального контроллера, датчики и исполнительные устройства могут взаимодействовать

напрямую. Этот протокол позволяет объединять в одну сеть большое количество разнообразных устройств, подходит для автоматизации больших зданий. KNX мало подходит для самоучек, монтировать такие системы должны профессионалы.

3.5.4. ZigBee

Это протокол основан на стандарте IEEE 802.15.4. Он обеспечивает связь по радиоканалу и хорошо подходит для организации умного дома. Датчики при таком формате потребляют мало энергии, поскольку срабатывают быстро. Формат ZigBee поддерживает ячеистую структуру сети, сигналы могут передаваться от одного устройства к другому через компоненты-посредники. Такая структура позволяет сети самовосстанавливаться, и даже выход из строя отдельных элементов не всегда приводит к обрушению всей сети. Стандарт ZigBee характеризуется умеренной стоимостью, пригоден как для жилых домов, так и для больших производственных помещений. Однако не всегда ZigBee-устройства разных производителей совместимы друг с другом.

3.5.5. Z-Wave

Этот протокол похож на ZigBee. Однако разница состоит в том, что все устройства, поддерживающие Z-Wave, основываются на беспроводных модулях компании Sigma Designs, разработавшей этот протокол, поэтому совместимы друг с другом. Это недорогой и небыстрый протокол, однако он подходит для домашней автоматизации.

3.5.6. Insteon

Этот стандарт популярен в США. Он передает данные через электропроводку здания, как и X10, но также делает это и по радиосвязи. Проводная и беспроводная сеть функционируют одновременно, что повышает надежность системы. Структура сети ячеистая, она может работать без центрального контроллера. Все устройства для поддержки такого протокола стандартизированы фирмой Smartlabs [2].

3.5.7. BLE

Это стандарт Bluetooth с низким энергопотреблением (Bluetooth Low Energy) [11]. Он передает данные между двумя близко расположенными приборами. Дальность передачи составляет 10–30 м. BLE используется для датчиков, устройств и приложений, не требующих передачи больших объемов данных. Применяется в малопотребляющих устройствах с большими интервалами передачи данных и использует 40 радиоканалов.

3.5.8. Wi-Fi

Wi-Fi — это технология беспроводной локальной сети на основе стандартов IEEE 802.11 Wireless Networks («Беспроводные сети»).

Стандарты IEEE разрабатываются Институтом инженеров-электриков и электронщиков (The Institute of Electrical and Electronics Engineers) [6], штаб-квартира которого находится в г. Нью-Йорк, а офисы — в более чем 160 странах мира, в том числе и в Москве.

Аббревиатура Wi-Fi происходит от словосочетания Wireless Fidelity («беспроводная точность») и является торговой маркой Wi-Fi Alliance [5] — объединения крупных производителей компьютерной техники и беспроводных устройств, поддерживающих Wi-Fi. В настоящее время к сетям Wi-Fi во всем мире подключено около 16 миллиардов устройств.

Обычно протоколом Wi-Fi автоматизированную систему умного дома подключают к смартфону или планшету. Wi-Fi и Интернет — это не одно и то же. Wi-Fi создает локальную сеть, в которой устройства могут обмениваться данными между собой. Если подключить эту сеть к Интернет-провайдеру через маршрутизатор (роутер), то компьютерное и прочее оборудование получает доступ в Интернет и может обмениваться данными через Интернет.

Сеть Wi-Fi может раздаваться на расстояние до 50–100 метров — хотя возможно усилить сигнал еще больше. При этом Wi-Fi потребляет меньше энергии, чем мобильный Интернет. Чаще всего Wi-Fi работает на частотах 2,4 ГГц и 5 ГГц. Однако излучение при этом маломощное и не влияет на здоровье человека.

Недостатками Wi-Fi являются возможные задержки сигнала и его потеря при прохождении через препятствия (например, несколько толстых стен).

Точка беспроводного доступа обеспечивает подключение устройств к беспроводной сети без использования кабелей [4]. Точка доступа использует полосу пропускания, которая предоставляется маршрутизатором, и распределяет ее таким образом, чтобы различные устройства могли получить доступ к сети. На смартфоне можно включить функцию мобильной точки доступа. Таким образом, другие устройства получают возможность совместно использовать подключение смартфона к Интернету.

2. ПЛАТА ESP8266 ДЛЯ УМНОГО ДОМА

ESP8266 – низкоэнергетический, высокоинтегрированный Wi-Fi-микроконтроллер, разработанный в Китае компанией Espressif Systems [9] в 2014 году.

На основе ESP8266 была создана NodeMCU, которая позволяет создавать разные устройства интернета вещей (IoT). Модуль позволяет отправлять и получать информацию в локальную сеть либо в Интернет при помощи Wi-Fi.

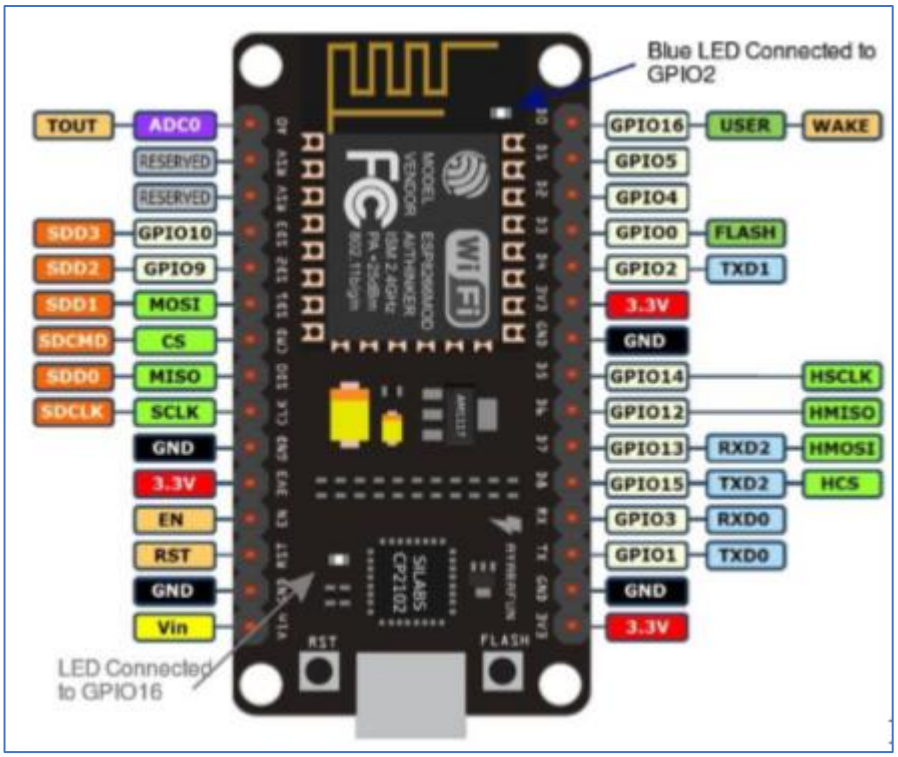

Рис. 1. Выводы NodeMCU на базе ESP8266.

Характеристики NodeMCU:

- Процессор Tensilica: L106 32-bit
- Частота процессора: 80-160 МГц
- Напряжение питания: 3,3 В
- Ток потребления: 300 мА при запуске и передаче данных, 35 мА во время работы, 80 мА в режиме точки доступа
- Максимальный ток вывода – 12 мА.
- Wi-Fi: 802.11 с поддержкой b/g/n
- Flash-память (память программы): 1 МБ

- Flash-память (файловое хранилище): 1-16 МБ
- EEPROM-память: до 4 кБ
- SRAM-память: 82 кБ
- GPIO: 11 выводов
- ШИМ: 10 выводов
- Прерывания: 10 выводов
- АЦП: 1 вывод с разрешающей способностью 1024
- I2C: 1 штука
- I2S: 1 штука
- SPI: 1 штука
- UART: 1,5 штуки
- Диапазон рабочих температур от - 40°C до +125°C.

ESP8266 может работать и как точка доступа, и как клиент. В настоящей курсовой работе ESP8266 выступает в роли клиента другой точки доступа.

Wi-Fi реализован синхронно, его обработчик должен постоянно вызываться во время работы программы не реже, чем каждые 20 мс (если Wi-Fi используется в программе).

3. ПРАКТИЧЕСКАЯ ЧАСТЬ

В качестве практической части задания выполнен прототип реально работающей системы управления Умным домом: на основе платы ESP8266 собрана система дистанционного управления электрическими приборами через сеть Wi-Fi и написано программное обеспечение для неё в Arduino IDE на языках программирования c++ и javascript.

3.1. Постановка задачи

Собрать макет устройства (ESP8266-сервера), дающего возможность дистанционно управлять устройствами Умного дома через Wi-Fi-подключение и Интернет.

В качестве условных физических устройств Умного дома, получающих управляющие сигналы, использовать светодиоды.

ESP8266-сервер подключается в качестве клиента по сети Wi-Fi к точке доступа (например, домашнему роутеру) и создает веб-сервер, доступный по сети. Сервер должен обслуживать обращения пользователей:

- Осуществлять авторизованный доступ к управлению системой Умного дома – до 5 ролей (отличаются уровнем доступа к устройствам и информированием пользователя о событиях).
- Обрабатывать переключение выключателей, выполненных в дизайне web-интерфейса, подачей с платы ESP8266 соответствующего управляющего сигнала на внешнюю плату, где в качестве сигнализации о полученном управляющем сигнале загорается светодиод. Каждый выключатель «привязан» к своему светодиоду.
- В зависимости от уровня доступа пользователя разрешать доступ к определенным переключателям (согласно таблице доступа, уникальной для каждого пользователя), а также информировать о том, когда и кто последним внес изменения в настройки системы.
- ESP8266-сервер должен работать от автономного источника питания и иметь статический IP-адрес в локальной сети.

3.2. Настройки роутера

Сервер ESP8266, управляющий умным домом, является стационарным устройством, которое должно быть постоянно подключено к точке доступа для возможности войти на него и получить доступ к управлению оборудованием. Поэтому использование им динамически выделяемого адреса не очень удобно. Конечно, можно подключить ESP8266 к какому-то компьютеру через USB-кабель и посмотреть, какой именно динамический IP-адрес был получен:

```
Serial.print("ESP IP: ");     Serial.print(WiFi.localIP());
Serial.print(", Gateway: "); Serial.println(WiFi.gatewayIP());
```

Рис. 2. Вывод IP-адреса ESP8266.

Но это не всегда удобно и ограничивает возможности расположения сервера. А в случае, если нет информации о новом IP, например, человек работает с сервером удаленно через NAT, войти на сервер будет невозможно. Для решения этой проблемы можно сделать привязывание по протоколу ARP IP-адреса к MAC-адресу Wi-Fi-карты ESP8266 и зарезервировать для этого конкретного MAC-адреса свой IP-адрес.

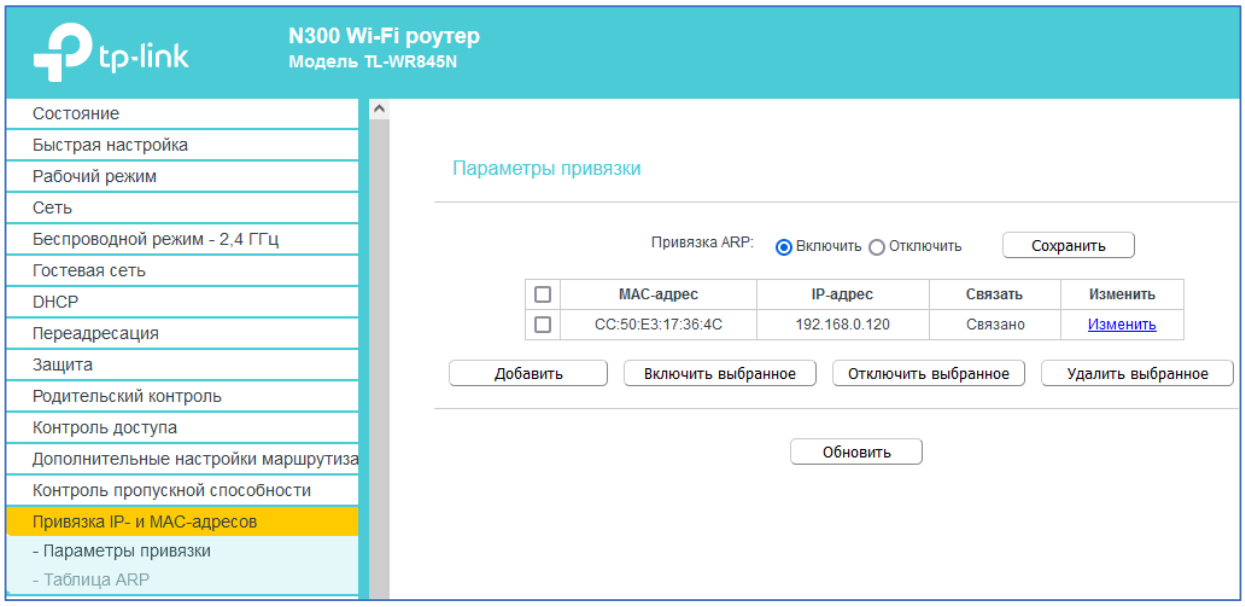

Рис. 3. Привязывание по протоколу ARP.

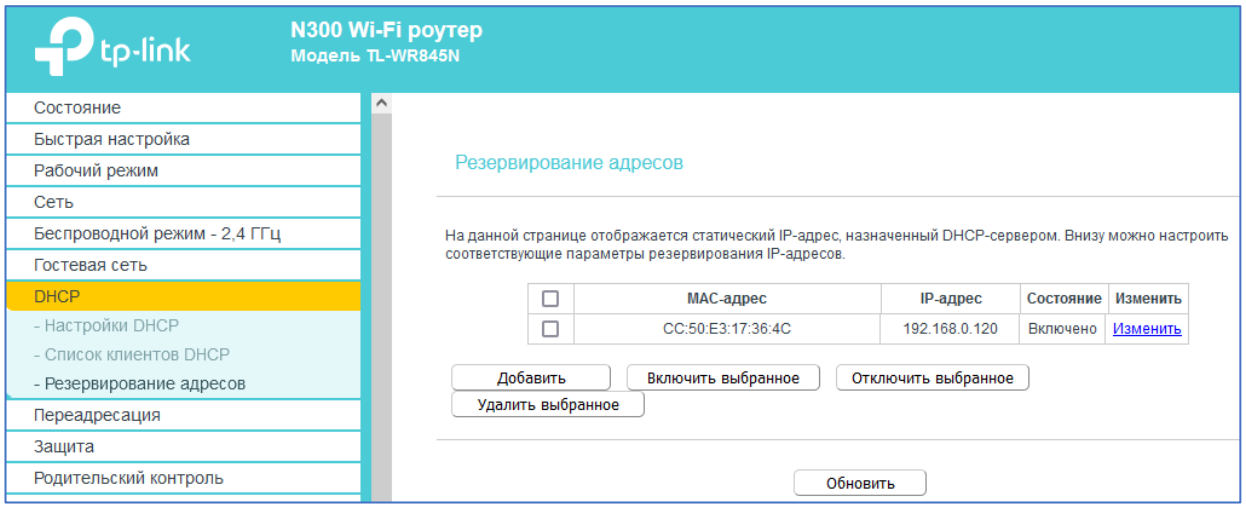

Рис. 4. Резервирование адреса для ESP8266.

Таким образом, после коннекта к точке доступа ESP8266 будет всегда иметь статичный назначенный IP-адрес. Это, в свою очередь, позволяет настроить NAT для статического отображения IP-адреса ESP8266 сервера на какой-то публичный IP-адрес и наоборот. Тогда управлять таким устройством и получать информацию о нем можно, находясь в любой точке мира, где есть Интернет.

3.3. Реализация

3.3.1. Техническая реализация (железо)

В качестве основы для сервера была выбрана плата ESP8266, описанная в части два настоящей работы. В качестве источника питания используется аккумулятор (power bank). Для увеличения количества управляющих выходов платы сервера, а также для защиты платы по нагрузке питания (так как максимальный ток по одному выводу, согласно документации, не может превышать 12мА, а светодиод может потреблять до 20мА) я распаял дополнительную мини-плату, на которой установил отдельный стабилизатор питания AMS1117-3.3 и запитал от него все светодиоды через микросхему “разветвитель” (shift register) 74hc595.

Обеспечивает питание для светодиодов и микросхемы стабилизатор питания.

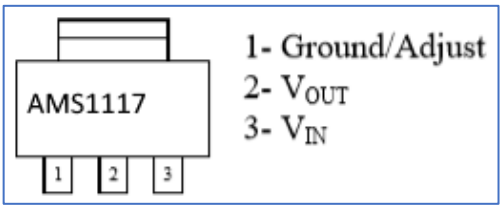

Рис. 5. Стабилизатор питания на AMS1117-3.3В.

Светодиод является индикатором наличия управляющего сигнала.

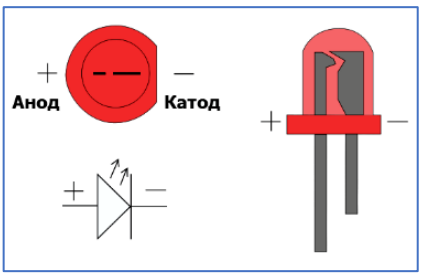

Рис. 6. Светодиод.

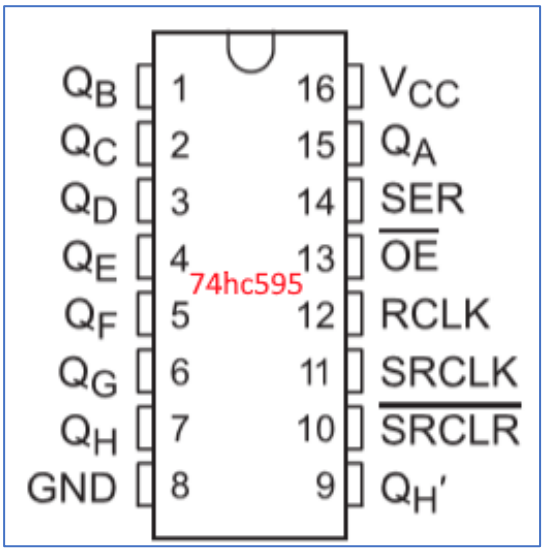

Рис. 7. 8-битный шифт-регистр 74hc595.

Используется для подключения большего количества светодиодов к плате ESP8266. Управляется шиной из трех проводов: SER -данные, RCLK - активация, SRCLK – синхроимпульсы. Создает уровень высокого напряжения на выходах Q_A-Q_H в зависимости от полученного бинарного кода по шине данных.

По сути, используя эту микросхему, мы «теряем» три вывода, но при этом получаем - восемь.

Общая схема системы.

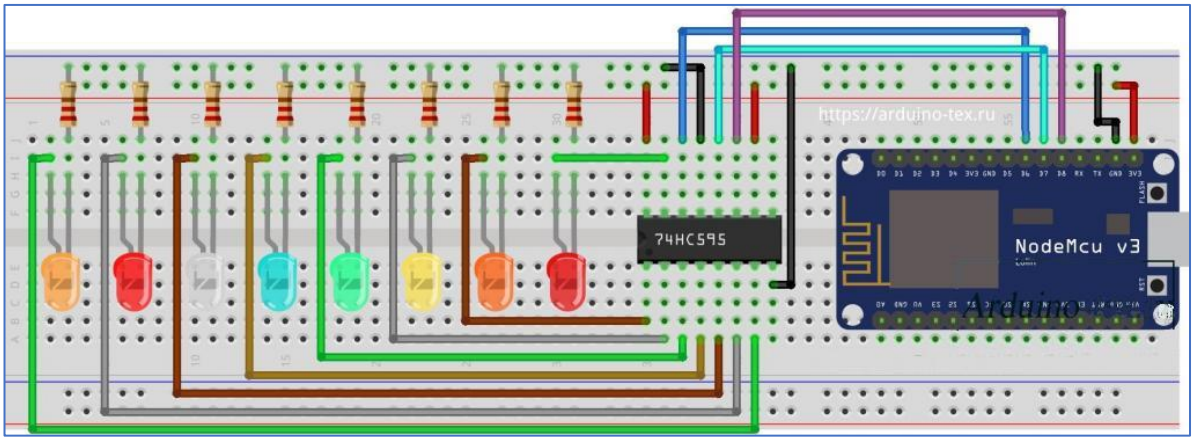

Рис. 8. Общая схема подключения светодиодов через 74ch595.

3.3.2. Программный модуль

Программный модуль состоит из двух частей:

- серверной, которая написана на языке программирования c++ и исполняется на плате ESP8266
- клиентской, которая написана на языке программирования javascript и исполняется в web-броузере на стороне пользователя

Серверная часть кода

Для реализации поставленной задачи было написано несколько классов таких как: Led, Tokens, UserAuthPassed и HC595. Каждый из этих классов содержит поля и методы для светодиодов, токенов, залогиненых пользователей и передачи данных на микросхему 74hc595 для включения и выключения светодиодов соответственно.

Логика работы сервера

По логике своей работы сервер работает следующим образом:

- подгружаются необходимые библиотеки:
- WiFi.h – для Wi-Fi подключения к точке доступа
- ESPAsyncTCP.h – работы с TCP протоколами
- ESPAsyncWebServer.h для запуска и обслуживания web-сервера
- "LoginPage.h" и "MainPage.h" – просто вынесенные в отдельные файлы тексты двух веб-страниц: авторизации и управления

```
#ifdef ESP32
  #include <WiFi.h>
  #include <AsyncTCP.h>
#else
  #include <ESP8266WiFi.h>
  #include <ESPAsyncTCP.h>
#endif
#include <ESPAsyncWebServer.h>
#include "LoginPage.h"
#include "MainPage.h"
```

Рис. 9. Подключение библиотек.

- инициализируются серийный порт и подключение по Wi-Fi к точке доступа. При этом используются заранее предопределенные ssid и пароль.

```
const char* ssid = "ssid вашей точки доступа";
const char* password = "пароль";

void setup(){
  Serial.begin(115200);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED){
    Serial.println("Connecting to WiFi..");
    delay(1000);}
  Serial.print("ESP IP: ");     Serial.print(WiFi.localIP());
  Serial.print(", Gateway: "); Serial.println(WiFi.gatewayIP());
```

Рис. 10. Инициалиазации. Вывод IP и GatewayIP ESP8266.

- настраиваются обработчики запросов Клиента, запускается сервер на порту номер 80.

```
AsyncWebServer server(80); // запускаем на 80 порту асинхронный WebServer
// root / стартовая страница: форма Login:Password
server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
  TOK.error = 0; request->send_P(200, "text/html", login_html, processor); });

// Обработка GET запроса от пользователя возврат информации о состоянии светодиодов.
server.on("/check", HTTP_GET, [](AsyncWebServerRequest *request){

// Изменение состояния светодиодов пользователем.
server.on("/update", HTTP_GET, [] (AsyncWebServerRequest *request) {

// обработка POST запроса от пользователя, проверка пересланного пароля
server.on("/", HTTP_POST, [](AsyncWebServerRequest *request){
  String loginn; String passs; int params = request->params();
  for (int i = 0; i < params; i++) {
      AsyncWebParameter *p = request->getParam(i);
      if (p->name()==PARAM_LOGIN){ loginn = p->value().c_str(); }
      if (p->name()==PARAM_PASS) { passs  = p->value().c_str(); }  } //for
  // Имеются вырезанные логин и пароль. Или их нет.
  int res = TOK.AddToken(loginn.c_str(), passs.c_str(), request->client()->remoteIP());
  if(res == -1) { request->send_P(200,"text/html",login_html,processor); }
  else          { request->send_P(200,"text/html",index_html,processor); TOK.dump(res); }
  });

server.begin();
```

Рис. 11. Настройка web-сервера ESP8266.

- после запуска сервер ожидает одного из четырех вариантов запроса со стороны пользователя: логин, логаут, получение и изменение статуса
- когда пользователь заходит на «корень» сервера, возвращает заранее подготовленную страницу авторизации. Поле подстановки (placeholder) %LOGINRESULT% заменяется на информационное сообщение, соответствующее причине попадания на эту страницу

```
</style>
</head>
<body>
  <hl>Авторизация</hl>
    <div id="wrapper">
    <form id="signin" method="post" action="" autocomplete="off">
      <input type="text" id="user" name="setl" placeholder="Login" />
      <input type="password" id="pass" name="setp" placeholder="Password" />
      <button type="submit"> > </button>
      <p>%LOGINRESULT%</p>
    </form>
    </div>
</body>
```

Рис. 12. Основной фрагмент кода html страницы, возвращаемой пользователю страницы авторизации.

Так эта страница выглядит для пользователя.

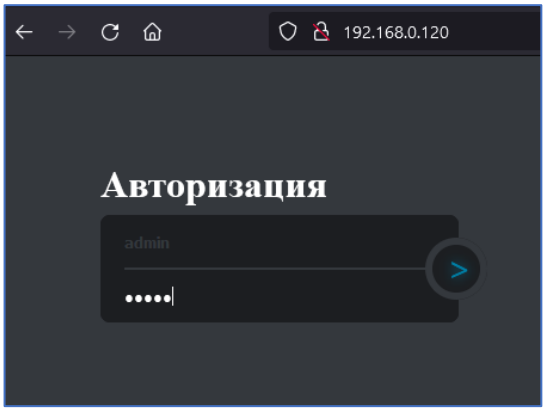

Рис. 13. Страница авторизации.

Если пользователь вводит логин и пароль, то обратно на сервер отсылается POST-запрос с этими эти данными. Сервер сравнивает их с (хардкорно) встроенной таблицей логинов и паролей. Таблица представляет из себя структуру.

```
struct user_types { const char* login;
                    const char* password;
                    const int led_access[MaxLedsInSystem];};

// Список ролей (уровней доступа пользователей)
// legend: 0b11 - два бита на один светодиод:
//                          1x - видит, x1-получает информацию
user_types ROLES[] = { {"admin", "admin", {0b11, 0b11, 0b11}},
                       {"chief", "admin", {0b11, 0b10, 0b00}},
                       {"user",  "12345", {0b10, 0b00, 0b00}},
                       {"dasha", "dasha", {0b11, 0b11, 0b10}},
                       {"mama",  "mama",  {0b11, 0b10, 0b11}}  };
```

Рис. 14. Структура данных: пользователи и уровни доступа.

На Рис.15 показан обмен пакетами с ESP8266-сервером в момент аутентификации и сразу после неё. IP-адрес сервера – 192.168.0.120, IP пользователя - 192.168.0.104.

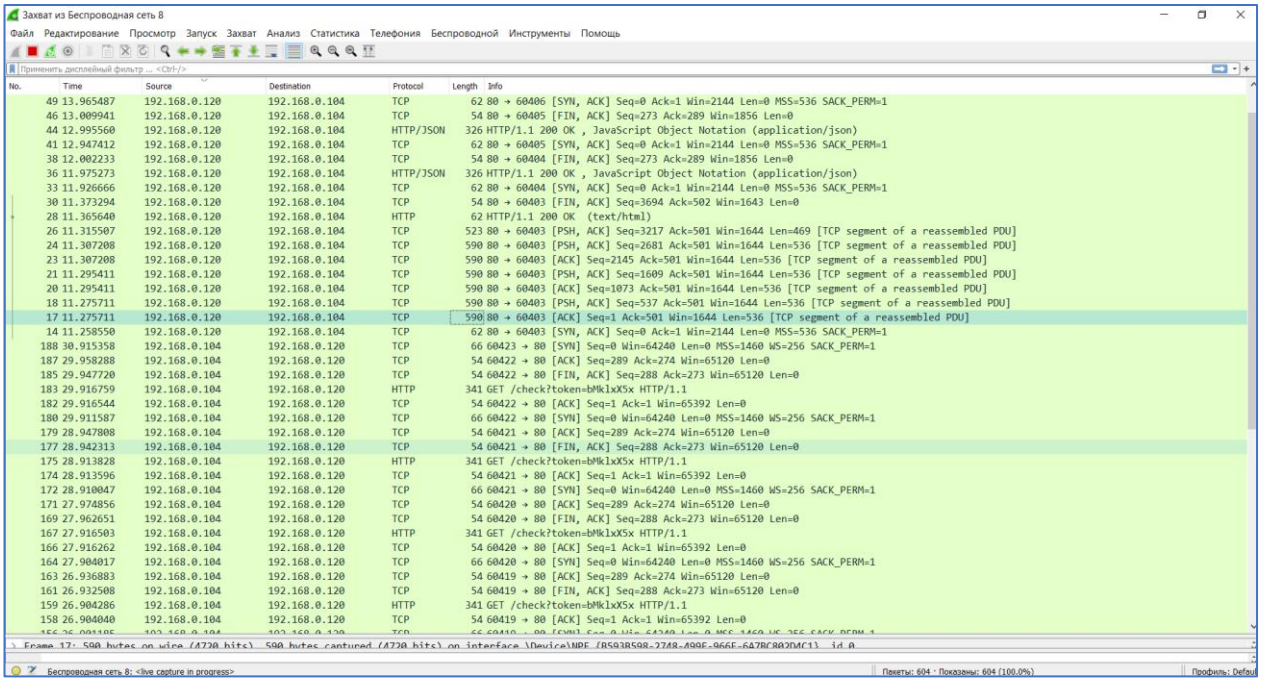

Рис. 15 Обмен пакетами с ESP8266-сервером в Wireshark

Если логин и пароль найдены в базе данных, то пользователь считается «прошедшим авторизацию». Дополнительно осуществляется проверка, не зашел ли уже кто-то под этим логином и паролем на сервер (в этом случае запрос отклоняется). Соответствующий номеру пользователя в базе светодиод включается – он сигнализирует, что пользователь находится в системе. IP адрес пользователя фиксируется в динамической базе данных авторизированных пользователей и случайным образом генерируется временный пароль – токен, используя который, он и продолжает дальше работать с сервером. Токен имеет ограниченный «срок жизни» - при генерации ему назначается таймаут, после которого он уже будет недействителен и не будет приниматься системой. Все операции на сервере (кроме логина) без корректного токена запрещены. Токен тоже сохраняется в базе данных.

```
const char alphabet[] = "0123456789ABCDE ... mnopqrstuvwxyz";
const int alphabetLen  = sizeof(alphabet)/sizeof(alphabet[0]);

void GenSetTokenToUser(int cli_idx){// генерирует Токен
      srand(millis()); UserOK[cli_idx].token[OneTokenLen] = 0;
      for (int i = 0; i < OneTokenLen; i++){
        UserOK[cli_idx].token[i] = alphabet[rand()%alphabetLen];} }
```

Рис. 16. Метод, генерирующий токен для пользователя.

```
// проверяет не истек ли срок действия токена
void CheckTokensTO(){
   for (int i = 0; i < UserOKLen; i++) {
    if (UserOK[i].role == -1) continue;
    UserOK[i].TokenTimeLeft(); } }

// ищет token по базе: -1 -нету, или индекс пользователя с этим токеном
void FindToken(const char * _token){
  for (int i = 0; i < UserOKLen; i++) {
    UserOK[i].TokenTimeLeft();
    if (UserOK[i].role == -1) continue;
    if ( cmpChar::c(_token, UserOK[i].token) ) {last_idx = i; return;}
  } last_idx = -1; }
```

Рис. 17 Методы класса токен:

проверка timeout и поиск пользователя с таким токеном.

Если авторизация не прошла, пользователь информируется об этом, и страница авторизации перегружается.

После успешной авторизации пользователь попадает на главную страницу, где может управлять переключателями работы устройств, получать информацию в соответствии со своим уровнем доступа к системе.

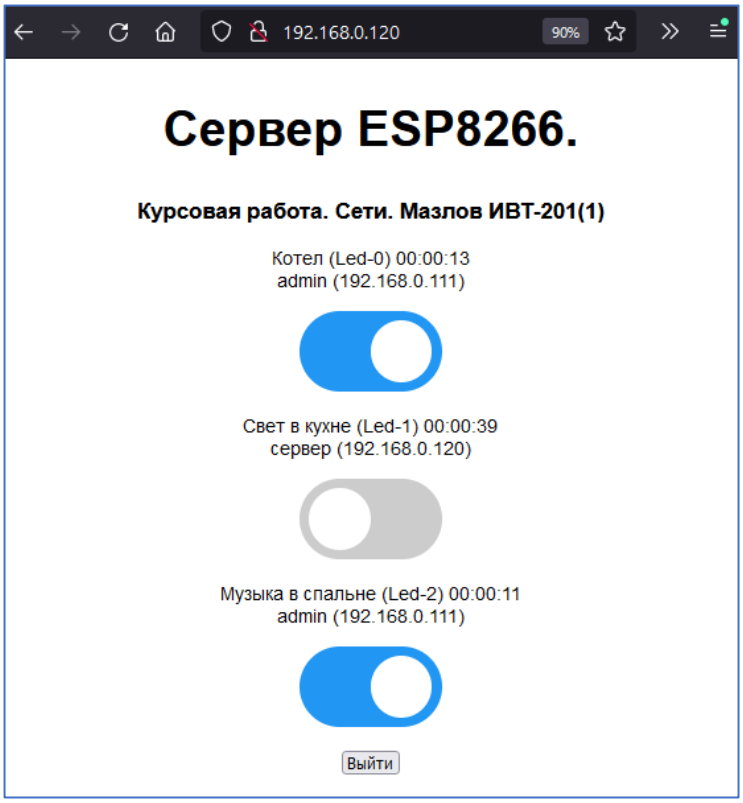

Рис. 18. Главная страница с элементами управления системой.

Один раз в секунду javascript на этой странице отсылает GET запрос на сервер и получает обновленный статус. Элементы страницы обновляются javascript. Изменения могли наступить в результате того, что другие авторизированные пользователи могли поменять настройки системы. Таким образом достигается синхронизация одинакового отображения настроек системы для всех пользователей (естественно, учитывая их уровень доступа). Ответ от сервера приходит в виде JSON объекта. Обработка запроса происходит только при передаче в параметрах запроса действительного токена авторизированного пользователя, который был выдан при аутентификации.

```
<script>
function callbackESP(){
    var xhr = new XMLHttpRequest();

    xhr.onload = function() {
    if (xhr.status != 200) { // проверка HTTP-статуса ответа, если статус не 200, то - ошибка
      alert(`Ошибка ${xhr.status}: ${xhr.statusText}`); location.href = '/'; }
    else if (xhr.responseText == "[]") {location.href = '/';}
    else { const obj = JSON.parse(xhr.responseText);
           for (let i = 0; i < obj.length; i++) {
             document.getElementById("ch" + obj[i][0]).checked   = obj[i][1];
             document.getElementById("state"  + obj[i][0]).innerHTML =
                (obj[i][1] ? "включено " : "выключено " ) + obj[i][2];}
         }  };
    xhr.open("GET", "/check?%TOKEN%", true); xhr.send();
  }

let timerId = setTimeout(function tick() {callbackESP(); timerId=setTimeout(tick,1000);},500);

</script>
```

Рис. 19. javascript код обновления страницы.

```
String LedInfo(int accessLVL) {          // "admin ip: 111.222.11.1" или " - "
  if (!(1 & accessLVL)) return "\" - \"";
  String res = "\"" + cmpChar::Mills2HHMMSS(millis() - last_changed);
  res+= "<br>" + String(login) + " (" + cmpChar::IP2String(ip) + ")\"";
  return res;
  }
```

Рис. 20. Метод класса LED: формирование ответа о состоянии переключателя по данным сервера.

Когда пользователь кликает по какому-то переключателю, на сервер направляется GET запрос с параметрами: токен, номер светодиода, установленное состояние. После проверки актуальности токена, информация о новом статусе фиксируется в базе светодиодов (class Led) и при помощи класса HC595 отправляется байт установок на микросхему 74hc595 и таким образом физически подается или убирается сигнал высокого уровня с вывода анода светодиода.

```
class HC595{ uint8_t mosi; uint8_t cs; uint8_t clk;
  public:  // отправка байта в регистр
    HC595(uint8_t _mosi, uint8_t _cs, uint8_t _clk){ // init
      pinMode(_mosi, OUTPUT);
      pinMode(_cs, OUTPUT);
      pinMode(_clk, OUTPUT);
      mosi = _mosi; cs = _cs; clk = _clk; return; Send(0);}

    void Send(uint8_t send_byte){
      digitalWrite(cs, LOW);
        shiftOut(mosi, clk, MSBFIRST, send_byte);
      digitalWrite(cs, HIGH); delay(1); }  };

HC595 HC = HC595(5,4,0);
// bits012:led012 (LEDS), connected: bits34567 - ROLES01234
uint8_t OUT = B00000000;
```

Рис. 21. Класс HC595. Управление уровнями напряжения на выходах микросхемы 74hc595.

```
// устанавливает новое значение бита "включенности" для светодиода,
// обновляет таймаут действия токена, физически включает светодиоды
  void   LedSet(int val, IPAddress _ip = no_ip, const char* _login){
    ip = _ip; login = _login; last_changed = millis();
    if (val){ OUT|= mask;}    // включаем
    else {OUT&= (~mask); }    // выключаем
    HC.Send(OUT);
    }

// возвращает текущее значение бита
  int    LedGet(){return ((OUT&mask)?1:0);}
```

Рис. 22. Методы Класса Led: LedSet() и LedGet().

Использование пользователем кнопки «Выход» отправляет HTTP_GET запрос на сервер с параметром %token%. Сервер удаляет пользователя из динамической базы данных авторизированных пользователей, обнуляет таймаут его токена и гасит соответствующий светодиод на информационной панели. Сессия

данного пользователя считается завершенной, и javascript переводит его на страницу ввода пароля и логина.

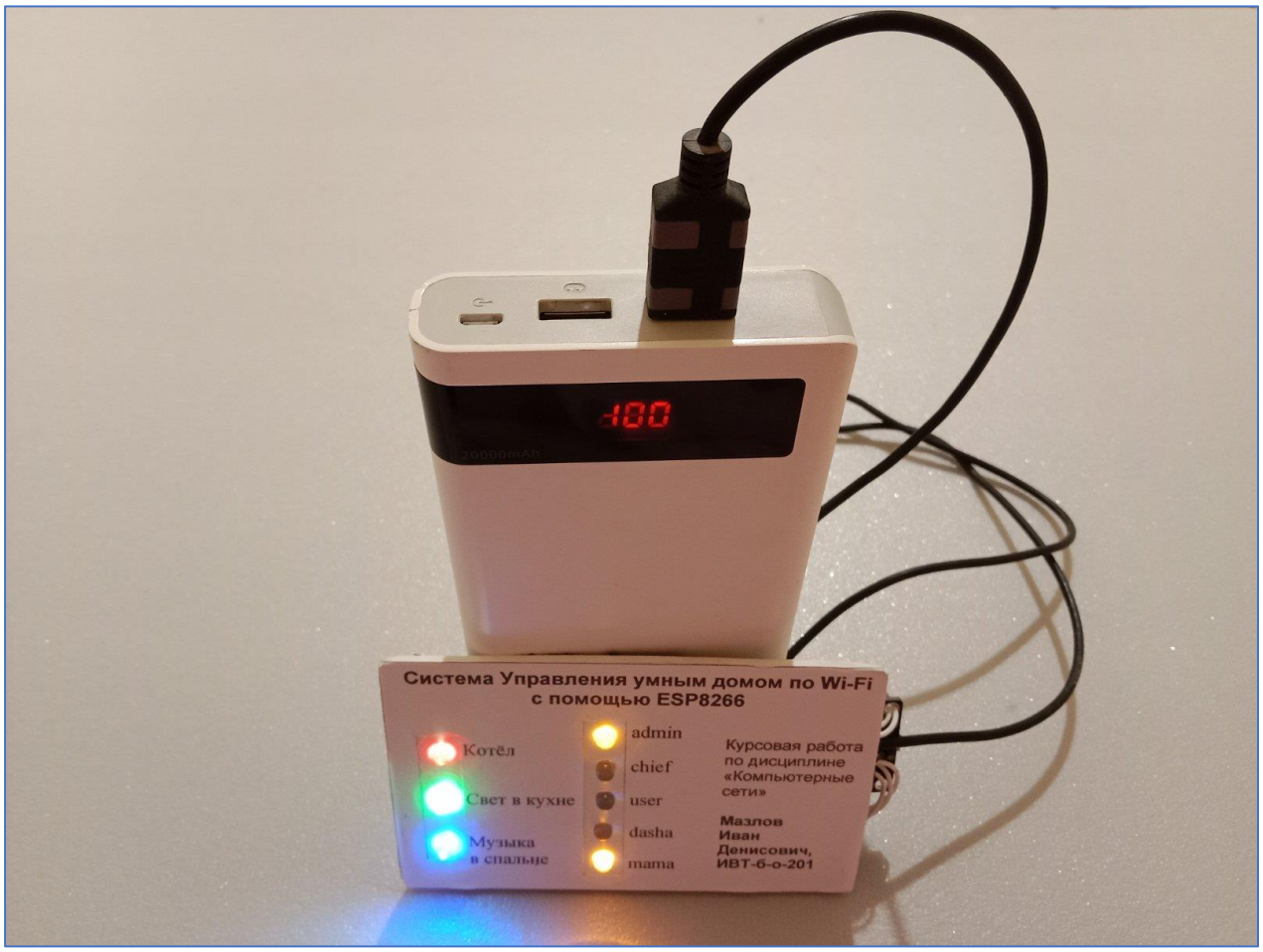

Рис. 23 Добро пожаловать на сервер!

ЗАКЛЮЧЕНИЕ

Создание систем «Умного дома» является не только полезным, но увлекательным, можно даже сказать – творческим процессом. Столько всего интересного можно придумать и реализовать - практически любой функционал по своему вкусу и желанию. В настоящее время уже широко доступен большой выбор датчиков (которые из себя, как правило, представляют небольшие собранные платы или микросхемы), сервомоторов для приведения любых элементов автоматизации в движение, регуляторов напряжения. Знание элементов программирования контроллеров позволяет реализовать любое взаимодействие всех этих датчиков, приборов, компьютерных сетей и любых других технологий и устройств.

Проект, представленный в настоящей курсовой работе, может быть развит в дальнейшем. Например, планируется добавить систему датчиков влажности и температуры, часы реального времени (RTC – Real Time Clock), часть управления перевести на автоматическое принятие решений, добавить регистрацию новых пользователей в системе, модернизировать интерфейс, добавить дополнительные информационные опции и уровни доступа.

Есть основания полагать, что в скором будущем значение систем автоматизации, робототехники и «Умного дома» возрастет, и их использование станет повсеместным. Так что будем и дальше развиваться в этом направлении, чтобы идти в ногу со временем!

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Manoj R. Thakur / NodeMCU ESP8266 Communication Methods and Protocols Programming with Arduino IDE.
2. Официальный сайт протокола Insteon. - URL: https://www.insteon.com/
3. aectaan / Bluetooth Low Energy: подробный гайд для начинающих. - 10 декабря 2020 г. - URL: https://habr.com/ru/post/532298/
4. Что такое Wi-Fi? - URL: https://www.cisco.com/c/ru_ru/products/wireless/what-is-wifi.html
5. Официальный сайт Wi-Fi Alliance. - URL: https://www.wi-fi.org/
6. Официальный сайт Института инженеров-электриков и электронщиков (The Institute of Electrical and Electronics Engineers). URL: https://www.ieee.org/
7. Универсальная энциклопедия Кирилла и Мефодия Megabook. - URL: https://megabook.ru/article/IEEE
8. xantorx / Как выбрать датчики для умного дома и какие они бывают. - URL: https://club.dns-shop.ru/blog/t-238-drugoe-dlya-umnogo-doma/59491-kak-vyibrat-datchiki-dlya-umnogo-doma-i-kakie-oni-byivaut/
9. Официальный сайт компании Expressif. - URL: https://www.espressif.com/en/products/modules/esp8266/
10. Intro to Bluetooth Low Energy. - URL: https://www.bluetooth.com/bluetooth-resources/intro-to-bluetooth-low-energy/
11. Vlek / Сети ZigBee. Зачем и почему? 17 октября 2012 г. - URL: https://habr.com/ru/post/155037/
12. Протоколы связи для "умного дома". - 29 июля 2014 г. - URL: https://www.ferra.ru/review/smarthome/SmartHome-Protocols.htm