

## Управление умными устройствами на базе Arduino с помощью сервиса WQTT

*Бокач Никита Александрович*

*Приамурский государственный университет имени Шолом-Алейхема  
Студент*

### Аннотация

В данной статье рассматривается сервис WQTT для управления умными устройствами на базе Arduino. А также интеграция этих устройств в разные голосовые ассистенты Алиса и Маруся.

**Ключевые слова:** MQTT, Arduino, сервис wqtt, умные устройства, устройства Интернет вещей.

### Managing Arduino-based smart devices using the WQTT service

*Bokach Nikita Aleksandrovich*

*Sholom-Aleichem Priamursky State University  
Student*

### Abstract

This article discusses the WQTT service for managing Arduino-based smart devices. As well as the integration of these devices into different voice assistants Alice and Marusya.

**Keywords:** MQTT, Arduino, wqtt service, smart devices, Internet of Things devices.

## 1 Введение

### 1.1 Актуальность

В современном мире, переплетенном с технологическими инновациями и стремительными изменениями в области информационных технологий, интернет вещей (IoT) становится важной составляющей повседневной жизни. Умные устройства, способные обмениваться данными и взаимодействовать между собой, не только улучшают уровень комфорта, но и предоставляют новые перспективы для автоматизации и оптимизации различных сфер человеческой деятельности.

### 1.2 Обзор исследований

В работе А. В. Малышев и К. К. Корнеев рассмотрели актуальные вопросы реализации «умного дома» посредством платы контроллера ESP32. [1]. В статье К. А. Гончаров демонстрирует возможность управления собственными устройствами умного дома, сконструированными на микроконтроллере Arduino при помощи MQTT сервера, а также раскрывает

тему преимущества и способов использования данного протокола общения устройств “интернета вещей” [2]. С. П. Орсик рассматривает вопрос создания паркингной системы с помощью микроконтроллеров семейств ESP32 и Arduino, которая обеспечивает контроль транспортных средств путём распознавания регистрационных номерных знаков [3]. Д. А. Файден в статье рассматривает возможность использования платы WeMos для работы с MQTT-брокером [4].

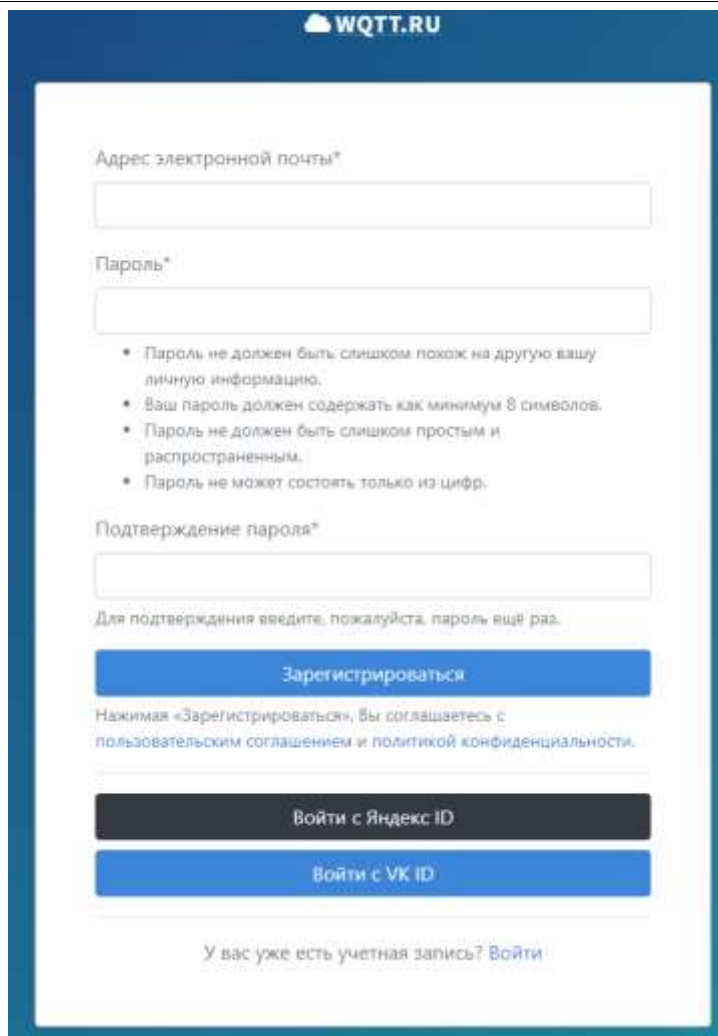
### **1.3 Цель исследования**

Цель данной статьи заключается в исследовании и анализе возможностей управления умными устройствами на базе Arduino с использованием сервиса WQTT [5]. Мы рассмотрим особенности данного сервиса, и его интеграцию с голосовыми помощниками Алиса [6] и Маруся [7].

## **2 Обзор и работа с сервисом WQTT**

Сервис WQTT это управляемые серверы Mosquitto в облаке. Mosquitto реализует транспортный протокол MQ Telemetry Transport protocol, MQTT, который предоставляет облегченные методы обмена сообщениями с использованием модели издатель - подписчик.

Для начала работы с сервисом WQTT необходимо зарегистрироваться. Переходим на главную страницу сайта и нажимаем кнопку Регистрация. Далее проходим быструю регистрацию с помощью социальных сетей или электронной почты (рис.1).

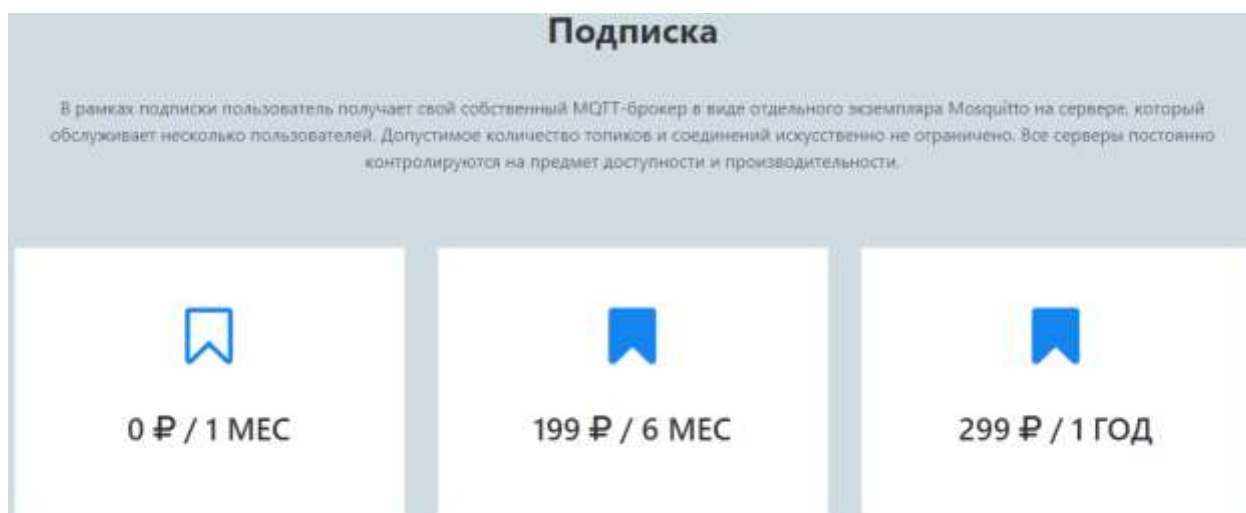


The image shows the registration page for WQTT.RU. At the top, the logo 'WQTT.RU' is displayed. The form includes the following fields and elements:

- Адрес электронной почты\***: A text input field for the email address.
- Пароль\***: A password input field with a list of requirements:
  - Пароль не должен быть слишком похож на другую вашу личную информацию.
  - Ваш пароль должен содержать как минимум 8 символов.
  - Пароль не должен быть слишком простым и распространенным.
  - Пароль не может состоять только из цифр.
- Подтверждение пароля\***: A second password input field.
- Для подтверждения введите, пожалуйста, пароль еще раз.**: A text instruction below the confirmation field.
- Зарегистрироваться**: A blue button to complete registration.
- Нажимая «Зарегистрироваться», вы соглашаетесь с пользовательским соглашением и политикой конфиденциальности.**: A text note below the registration button.
- Войти с Яндекс ID**: A dark grey button for social login.
- Войти с VK ID**: A blue button for social login.
- У вас уже есть учетная запись? Войти**: A link for existing users.

Рисунок 1 – Регистрация на сайте сервиса wqtt

Сервис WQTT предоставляет тестовый период 1 месяц для новых пользователей, а после, нужно приобретать подписку (рис. 2). После успешной регистрации открывается страница личного кабинета (рис. 3).



The image shows the 'Подписка' (Subscription) page. It features a header with the title 'Подписка' and a descriptive paragraph: 'В рамках подписки пользователь получает свой собственный MQTT-брокер в виде отдельного экземпляра Mosquitto на сервере, который обслуживает несколько пользователей. Допустимое количество топики и соединений искусственно не ограничено. Все серверы постоянно контролируются на предмет доступности и производительности.'

Below the text are three subscription options, each represented by a blue bookmark icon and a price:

Subscription Plan	Price
0 ₺ / 1 МЕС	0 ₺ / 1 МЕС
199 ₺ / 6 МЕС	199 ₺ / 6 МЕС
299 ₺ / 1 ГОД	299 ₺ / 1 ГОД

Рисунок 2 – Цены на подписку сервиса

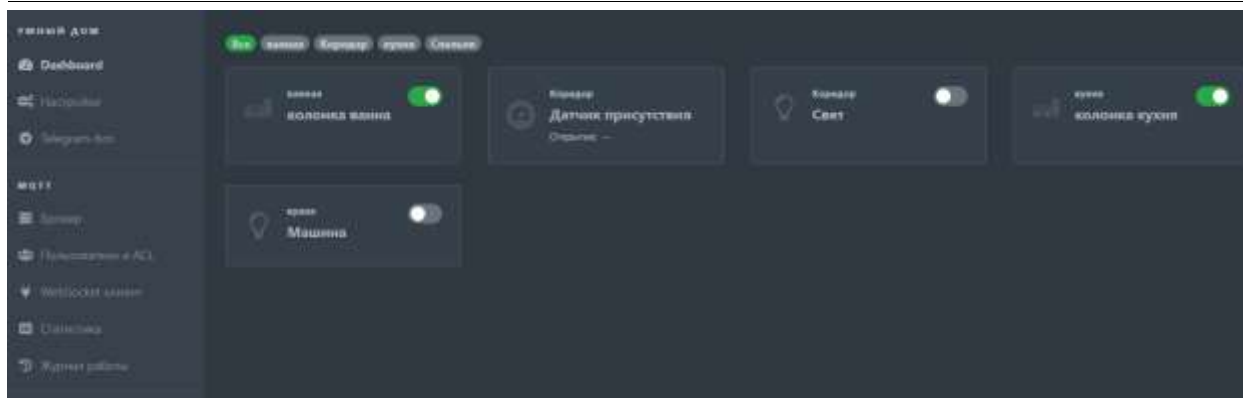


Рисунок 3 – Страница личного кабинета

Для подключения умных устройств на базе Arduino, перейдем на вкладку брокер в личном кабинете и скопируем данные для подключения к серверу (рис. 4).

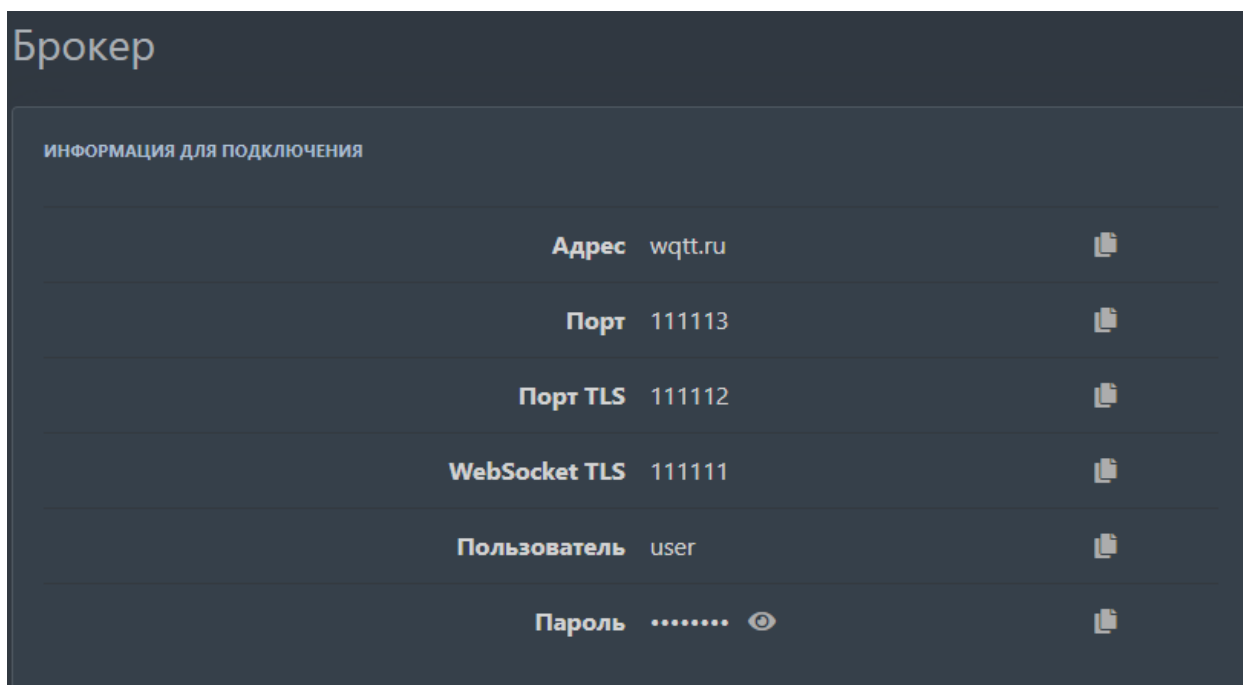


Рисунок 4 – Данные для подключения к серверу

Введем данные для подключения к серверу в прошивке устройства, а также укажем топики откуда устройство будет принимать команды (рис. 5).

```
// MQTT
const char* mqtt_server = "wqtt.ru";
const int mqtt_port = 11111;
const char* mqtt_user = "user";
const char* mqtt_password = "pass";

// MQTT топик
const char* mqtt_topic = "esp/state/KYXNYA";
const char* mqtt_topic2 = "esp/state/VANNA";

// RELAY
const String relay_topic = "/topic";
const int VANNA = D2;
const int KYXNYA = D3;
```

Рисунок 5 – Заполнение данных для подключения в прошивке устройства

Теперь, когда устройство подключилось к серверу и готово к работе, настроим его в сервисе wqtt. Для этого перейдем во вкладку настройки в личном кабинете. Далее нажимаем кнопку добавить устройство, затем необходимо выбрать тип этого устройства (рис. 6).

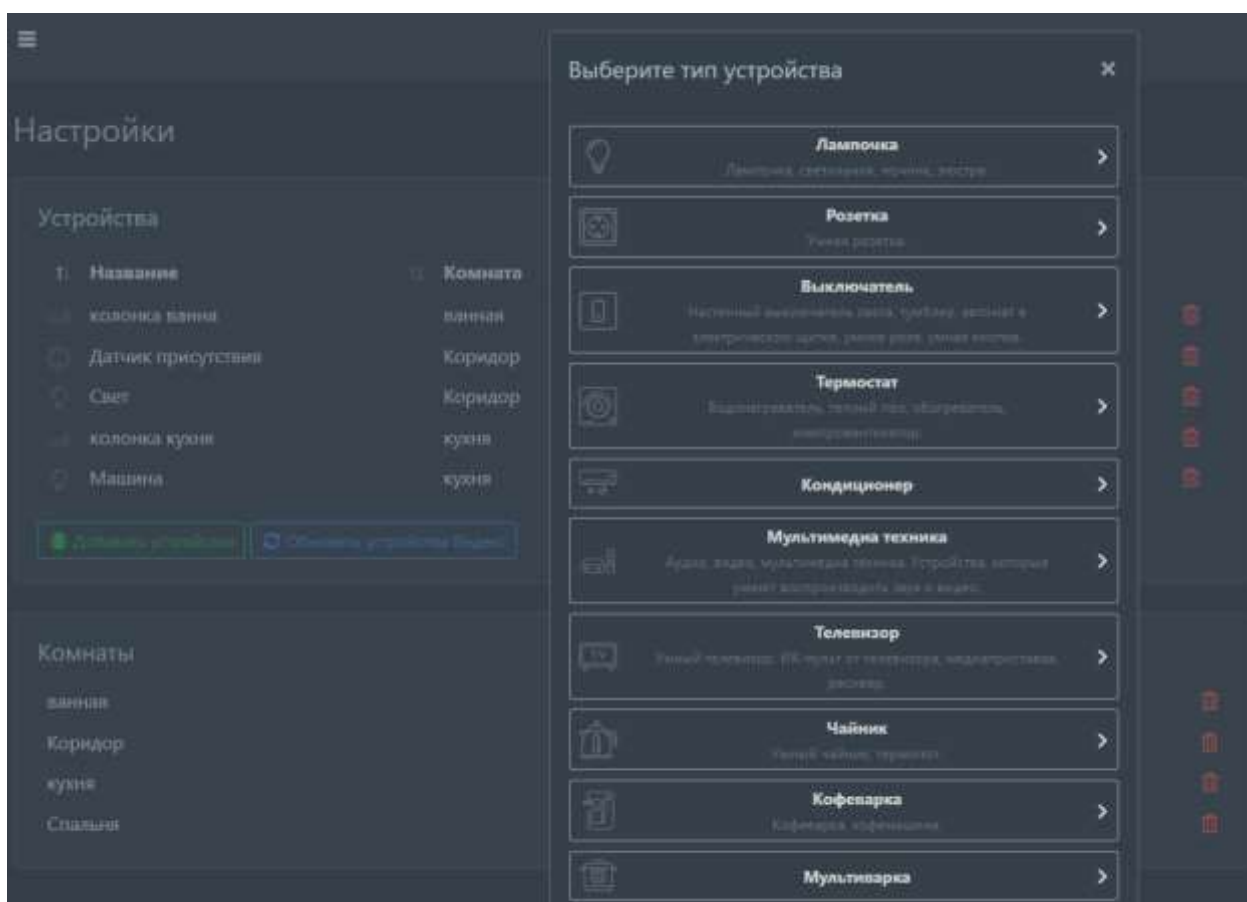


Рисунок 6 – Добавление и настройка нового устройства в личный кабинет

В качестве примера настроим устройство лампочка. После выбора типа устройства, открывается страница с настройкой этого устройства, где необходимо ввести название устройства, указать комнату, где оно

расположено, это нужно для последующего удобства и группировки устройств. Переходим к следующему пункту настройки устройства, а именно настройки органов управления им. Для включения и выключения лампы необходимо добавить соответствующую кнопку. Нажимаем кнопку добавить (рис. 7).

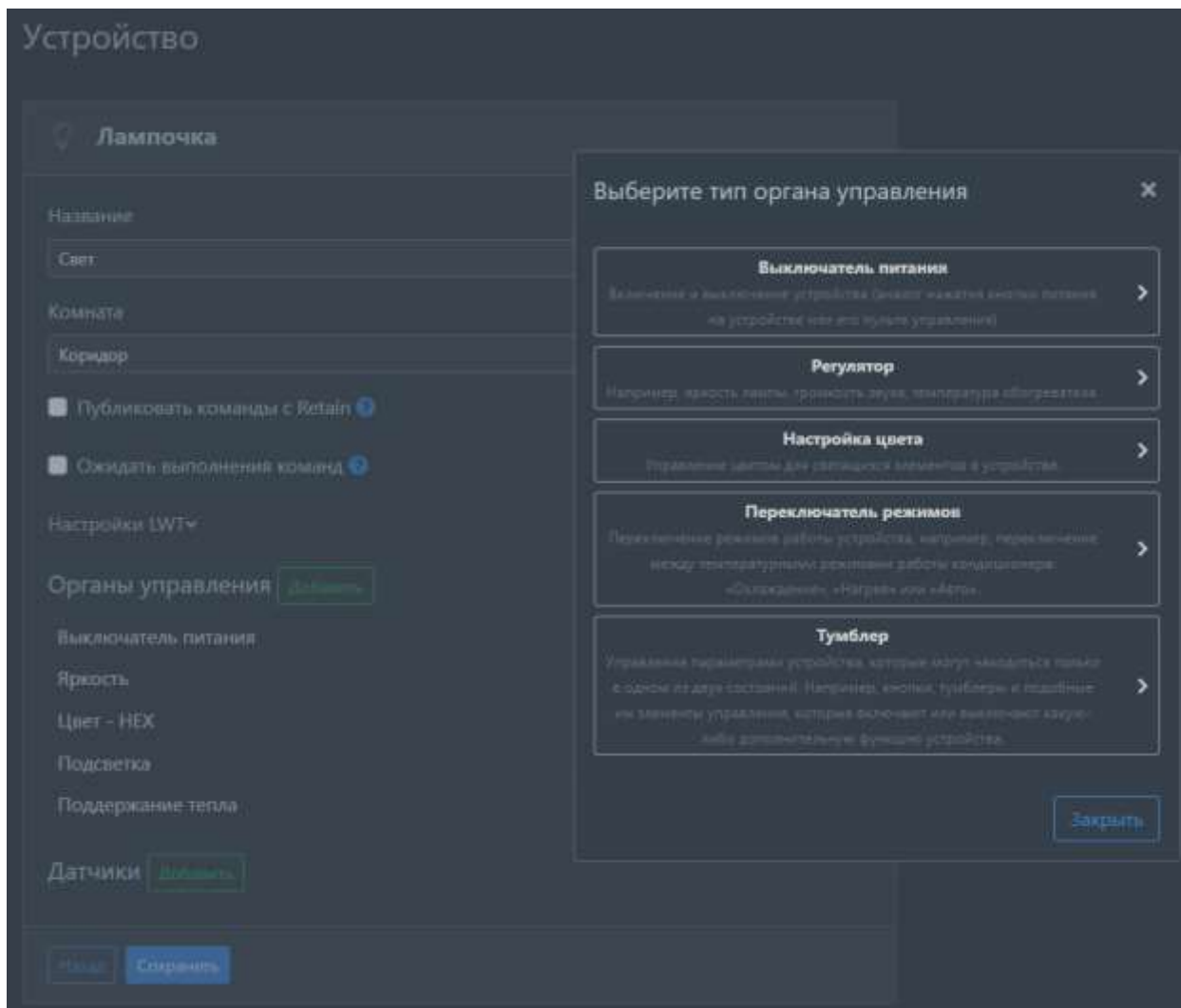


Рисунок 7 – Добавления кнопки для включения лампы

После выбора типа кнопки для управления лампой, открывается страница, на которой нужно ввести топик в который сервис будет отправлять команду, а также нужно указать сами команды для разных сценариев кнопки (рис. 8).

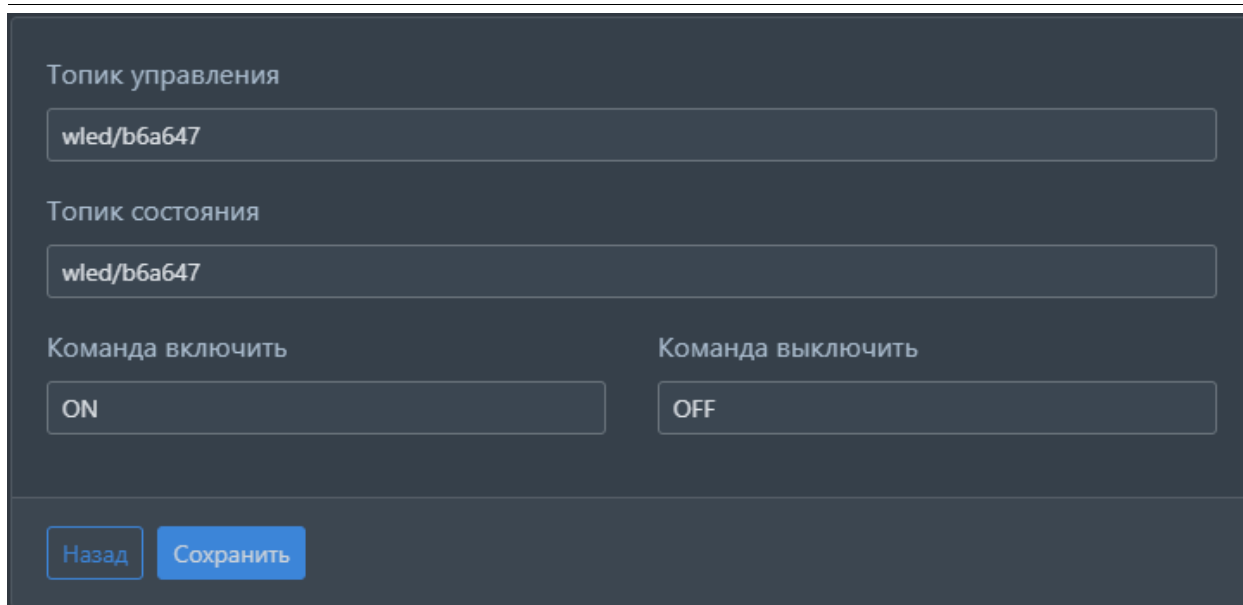


Рисунок 8 – Настройка кнопки управления питанием лампы

Когда все данные будут заполнены, кнопка сохранить станет активной. Сохраняем настройки и переходим на домашнюю страницу личного кабинета. Добавленное устройство должно появиться на этой вкладке. Можно нажать на устройство лапа, тогда откроется окно управления этой лампой (рис. 9).



Рисунок 9 – Управление лампой из личного кабинета

Теперь устройство лампа, можно добавить в голосовые помощники Алиса и Маруся. Для этого переходим в приложение голосового помощника, для яндекс Алисы – это, приложение умный дом. Переходим в него и нажимаем кнопку добавить умное устройство, далее в поиске производителей устройств ищем WQTT. Затем авторизуемся в сервисе wqtt, готово (рис. 10). Устройство должно отобразиться на главном экране приложения умный дом (рис. 11). Теперь при вызове голосового помощника он сможет управлять этим устройством.

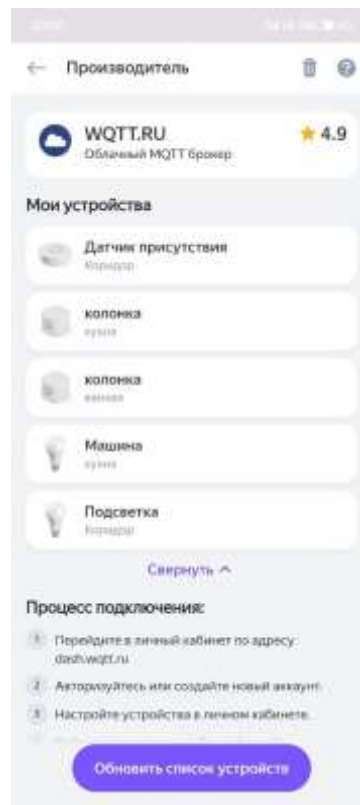


Рисунок 10 – Добавление устройств из сервиса wqtt

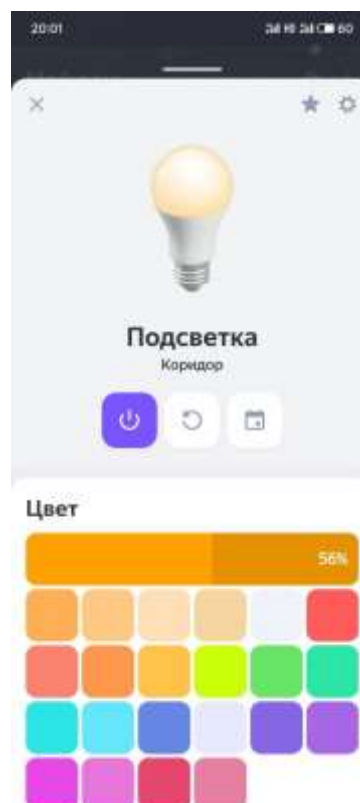


Рисунок 11 – Управление устройством лампы из приложения умный дом

В данной статье был проведен обзор на сервис WQTT и его интеграцию с Arduino и голосовыми помощниками. В результате использования сервиса WQTT можно отметить, что данная технологическая комбинация



предоставляет перспективное решение в контексте создания современных систем умного дома. Стоит отметить что данное решение работает на облачных серверах, и при отсутствующем интернет подключении или перебоями в работе сервиса, управлять умными устройствами не получится.

### **Библиографический список**

1. Малышев А. В. Реализация "умного дома" на ESP32 через протокол MQTT // КИП и автоматика: обслуживание и ремонт. 2021. № 1. С. 46-47.
2. Гончаров К. А. Реализация управления приборами как частями умного дома при помощи esp8266 и протокола MQTT // Энигма. 2021. № 29-2. С. 170-176.
3. Орсик С. П. Разработка паркинговой системы на базе микроконтроллера ESP32 и протокола передачи данных MQTT // Endless Light in Science. 2023. № 1-1. С. 310-315.
4. Файден Д. А. Использование платы типа wemos для связи с mqtt-сервером // Современные проблемы радиоэлектроники и телекоммуникаций. 2020. № 3. С. 117.
5. Сервис WQTT URL: <https://wqtt.ru/> (дата обращения: 27.01.2024)
6. Голосовой помощник Яндекс Алиса URL: <https://yandex.ru/alice> (дата обращения: 27.01.2024)
7. Голосовой помощник VK Маруся URL: <https://marusia.vk.com/> (дата обращения: 27.01.2024)