

## Использование протокола MQTT для управления умными устройствами на базе Arduino

*Бокач Никита Александрович*

*Приамурский государственный университет имени Шолом-Алейхема*

*Студент*

### Аннотация

В данной статье исследуется применение протокола MQTT (Message Queuing Telemetry Transport) для управления интеллектуальными устройствами на базе Arduino в контексте Интернета вещей (IoT). Поскольку платформа Интернета вещей продолжает расширяться, потребность в эффективных и стандартизированных коммуникационных протоколах становится все более очевидной, особенно при взаимодействии с платформами микроконтроллеров, такими как Arduino.

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

### Using the MQTT protocol to control Arduino-based smart devices

*Bokach Nikita Aleksandrovich*

*Sholom-Aleichem Priamursky State University*

*Student*

### Abstract

This article explores the application of the MQTT (Message Queuing Telemetry Transport) protocol to control Arduino-based smart devices in the context of the Internet of Things (IoT). As the Internet of Things platform continues to expand, the need for efficient and standardized communication protocols is becoming increasingly evident, especially when interacting with microcontroller platforms such as Arduino.

**Keywords:** MQTT, Arduino, Smart devices, Internet of Things devices.

## 1 Введение

### 1.1 Актуальность

В условиях постоянно развивающегося Интернета вещей (IoT) бесперебойная связь и управление интеллектуальными устройствами имеют первостепенное значение. В этой статье рассматривается использование протокола MQTT в качестве надежного средства управления интеллектуальными устройствами на базе Arduino. Поскольку наш мир становится все более взаимосвязанным, потребность в эффективных и стандартизированных коммуникационных протоколах никогда не была более

острой, особенно при взаимодействии с системами на базе микроконтроллеров, такими как Arduino.

## 1.2 Обзор исследований

В работе Т. В. Костеннова предложен подход, направленный на уменьшение использования сетевых ресурсов, основанный на агрегации данных на шлюзах-агрегаторах. [1]. В статье Q. HIDAYATI и N. JAMAL авторами была создана система мониторинга сети датчиков затопления автомобильных дорог. Система подключается к сети Интернет через модуль NodeMCU ESP8266 и отправляет данные датчиков на сервер по протоколу Message Queuing Telemetry Transport (MQTT) [2]. С. Г. Майхуб рассмотрел вариант стека протоколов для сетей IoT, содержащий протокол маршрутизации RPL, протокол уровня адаптации 6LoWPAN и протокол обмена сообщениями MQTT [3]. Д. И. Дикий и В. Д. Артемьева в статье рассматривают вопросы безопасности в интернете вещей, а именно организация безопасного разграничения доступа при использовании протокола MQTT [4].

## 1.3 Цель исследования

Основной целью этой научной студенческой статьи является изучение и разъяснение практических применений и последствий использования протокола MQTT (Message Queuing Telemetry Transport) для управления интеллектуальными устройствами на базе платформы Arduino.

## 2 Обзор и работа с протоколом MQTT

Протокол MQTT (Message Queuing Telemetry Transport) представляет собой легковесный протокол обмена сообщениями, разработанный для эффективной передачи данных в сетях с ограниченной пропускной способностью. Основная архитектурная особенность протокола — модель "публикация/подписка". Устройства, функционирующие как издатели, отправляют сообщения на определенные темы (topics), а устройства-подписчики получают сообщения по темам, на которые они подписаны. Центральным элементом в архитектуре MQTT является брокер, который координирует обмен сообщениями между устройствами.

Протокол MQTT был разработан и опубликован в 1999 году Andy Stanford-Clark из IBM и Arlen Nipper из Arcom (теперь Eurotech) и стал OASIS стандартом в 2014 году. MQTT специально разработан для сетей с ограниченной пропускной способностью и нестабильным соединением, что часто встречается в мобильных и спутниковых связях.

Основные особенности протокола MQTT:

- Легковесность: MQTT оптимизирован для работы в условиях ограниченных ресурсов устройств, обеспечивая эффективную передачу данных.

- Темы (Topics): Система топиков позволяет организовывать сообщения по категориям, облегчая структурирование и управление обменом информацией.
- Качество Обслуживания (QoS): Протокол предоставляет три уровня QoS, определяющих степень гарантированной доставки сообщений.
- Сессии: MQTT поддерживает сессии, что обеспечивает сохранение состояния соединения между устройствами и брокером.
- Эффективность Энергопотребления: Оптимизированный протокол минимизирует энергопотребление, что делает его подходящим для устройств с ограниченной батареей.

MQTT широко используется в сценариях IoT благодаря своей способности обеспечивать эффективное взаимодействие между устройствами. Он применяется в умных домах, системах мониторинга, промышленных устройствах и других областях, где требуется надежный обмен данными.

Далее рассмотрим интеграцию и работу протокола mqtt на Arduino.

Для начала работы с протоколом MQTT на Arduino, установим библиотеку PubSubClient. Затем настроим переменные и пропишем адрес брокера логин и пароль, топики в которые устройство будет отправлять информацию и топик в котором будут публиковаться команды для устройства (рис. 1).

```
// MQTT
const char* mqtt_server = "m2.wqtt.ru";
const int mqtt_port = 12549;
const char* mqtt_user = "u_WQFUTC";
const char* mqtt_password = "Rpass";

// 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 1 – Переменные и адреса топиков

Переменные mqtt\_topic и mqtt\_topic2 служат для отправки устройством, состояния реле в нужных комнатах. Переменная relay\_topic нужна для получения команд на устройство. Рассмотрим обработку команд на устройстве (рис. 2).

```
void callback(char* topic, byte* payload, unsigned int length) {

    String data_pay;
    for (int i = 0; i < length; i++) {
        data_pay += String((char)payload[i]);
    }

    Serial.println(data_pay);

    if( String(topic) == relay_topic ){
        if(data_pay == "KON") vanna_on = true, kyxnyaya = "KON";
        if(data_pay == "KOFF") vanna_on = false, kyxnyaya = "KOFF";
        if(data_pay == "VON") kyxnya_on = true, vannaya = "VON";
        if(data_pay == "VOFF") kyxnya_on = false, vannaya = "VOFF";
        if(data_pay == "enginestart") start_on = true;
        if(data_pay == "enginestop") start_off = true;
        if(data_pay == "pair") pair = true;
    }

    updateStatePins();
    updateStatePins2();
    startkey();
    stopkey();
    pairkey();
    handle_state();
}
```

Рисунок 2 – Обработка полученных команд на устройство

Программа микроконтроллера обрабатывает все полученные команды по адресу /topic она сравнивает команды с заложенными в ней, если найдена схожая команда, то микроконтроллер приступает к выполнению данной команды. Если устройство получило команду KON, то оно включит реле, отвечающее за кухню. Для отслеживания состояния реле, устройство отправляет текущее состояние реле (рис. 3).

```
void handle_state()
{
    // Получение температуры с датчика

    // Отправка данных в MQTT топик
    const char* payload = kyxnyaya;
    client.publish(mqtt_topic, payload);

    // Отправка данных в MQTT топик
    const char* payload2 = vannaya;
    client.publish(mqtt_topic2, payload2);

}
```

Рисунок 3 – Отправка состояния устройства

Так данная программа отправляет данные о состоянии реле. Рассмотрим, как это выглядит со стороны брокера (рис. 4).

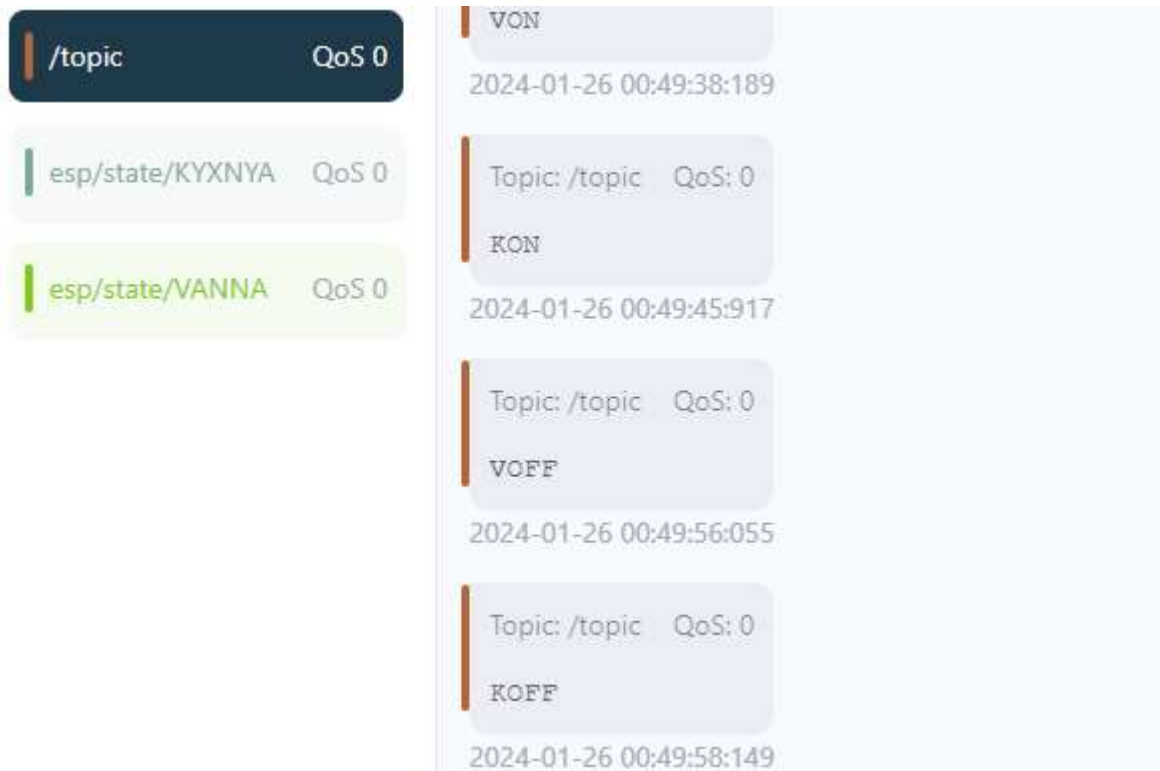


Рисунок 4 – Команды отправленные на брокер

Это команды отправленные на устройство, их может отправлять другое умное устройство или жн они могут выполняться по сценарию в системе умного дома. Рассмотрим ответ от устройства (рис. 5).



Рисунок 5 – Ответ на команды от устройства

Так устройство сообщает том, что на данный момент реле в конкретной комнате включенно или выключенно.

В данной статье был проведен обширный анализ использования протокола MQTT для управления умными устройствами, реализованными на базе Arduino. Результаты исследования подтвердили высокую эффективность применения MQTT в контексте умных устройств, отмечая его легковесность, гибкость и способность эффективно управлять обменом данными.

### **Библиографический список**

1. Костеннов Т. В. Об эффективности агрегации данных в сетях интернета вещей с использованием протокола MQTT // Математические структуры и моделирование. – 2023. – № 4(68).
2. Hidayati Q. Sistem monitoring pada jaringan sensor banjir jalan raya menggunakan protokol MQTT // JTEL (Jurnal Ilmiah Telekomunikasi, Elektronika, dan Listrik Tenaga). 2022. Vol. 2. No. 2. P. 119-128.
3. Майхуб С. Г. Модель сбора данных в сетях IoT на основе протокола обмена сообщениями MQTT // Фундаментальные проблемы радиоэлектронного приборостроения. 2017. Т. 17. № 4. С. 991-995.
4. Дикий Д. И. Протокол передачи данных MQTT в модели удаленного управления правами доступа для сетей Интернета // Научно-технический вестник информационных технологий, механики и оптики. – 2019. – Т. 19, № 1. С. 109-117.