

УДК 004.771

## СИСТЕМА УДАЛЕННОГО УПРАВЛЕНИЯ УСТРОЙСТВАМИ НА БАЗЕ МИКРОКОНТРОЛЛЕРА ESP32

*Маринович Д.А.<sup>1</sup>, учащийся НДТ,  
Андрейчук А.О.<sup>2</sup>, преподаватель дисциплин специального цикла,  
Сицко В.А.<sup>3</sup>, старший преподаватель кафедры ИСиТ*

*Учреждение образования «Национальный детский технопарк»<sup>1</sup>,  
Белорусский государственный университет информатики и радиоэлектроники,  
филиал Минский радиотехнический колледж<sup>2</sup>,  
Белорусский государственный университет информатики и радиоэлектроники,  
Институт информационных технологий<sup>3</sup>,  
г. Минск, Республика Беларусь*

*Сицко В.А. – магистр техн. наук*

**Аннотация.** В статье рассмотрены особенности протокола обмена сообщениями MQTT. Разработаны основные функции управления микроконтроллером ESP32. Предложена модель системы удаленного управления устройствами на базе микроконтроллера ESP32 с разработкой клиентской и серверной части веб-приложения.

**Ключевые слова.** Программирование микроконтроллеров, протокол MQTT, микроконтроллер ESP32

**Введение.** В современном мире технологии продвинулись до такого уровня, что можно управлять различными системами удаленно. Все больше людей нуждается в управлении устройствами без привязки к определенному месту. Довольно часто приходится готовить презентации и выставлять их на всеобщее обозрение. В связи с этим возникает проблема, связанная с возможностью управления сменой слайдов на экране удаленно. Существует аппаратное решение этого вопроса - физические пульты для управления презентацией, но постоянно их с собой носить не всегда есть возможность. Проектное решение позволит решить данную проблему путем программной организации передачи функций пульта мобильному устройству.

На данный момент существует два типа программ для управления презентацией: с мобильным приложением и с веб-управлением. Однако, у них есть весомые недостатки. Один из главных – цена этих мобильных приложений. Большинство качественных решений на мобильных устройствах за использование полного функционала требует купить платную версию. Вторым недостатком является тип выбранного подключения: Bluetooth и WiFi модем. Дальность распространения сигнала не превышает 10 метров, что может являться проблемой в больших аудиториях. Третий – IP-адрес. Подавляющее большинство решений использует именно его для соединения между устройствами, что снижает уровень безопасности данных на устройствах. Примерами таких программ являются: Remote Control Presentation, i-Clickr PowerPoint Remote, Clicer, Remote for PowerPoint Keynote.

В рамках проекта была поставлена задача разработать приложение для мобильных и компьютерных систем для управления устройствами через интернет на расстоянии. Кроме того, предлагаемое проектное решение позволит упростить и распространить эту технологию на большую аудиторию.

**Основная часть.** Для реализации предлагаемой системы удаленного управления устройствами были применены следующие программные средства:

- среда визуальной разработки приложений для ОС Android «App Inventor»;
- текстовый редактор VS Code с возможностью подключения различных плагинов, что дает возможность работать с разными языками программирования для разработки любого IT-продукта;
- PlatformIO – платформа с открытым исходным кодом на языке Python, удобная для разработки проектов «интернета вещей» (IoT);
- язык программирования Python;
- MQTT (или Message Queue Telemetry Transport) – упрощенный сетевой протокол, работающий поверх TCP/IP, ориентированный на обмен сообщениями между устройствами по принципу издатель-подписчик.

Основными особенностями протокола MQTT можно ответить: асинхронность, компактные сообщения, работа в условиях нестабильной связи на линии передачи данных, поддержка нескольких уровней качества обслуживания (QoS), легкая интеграция новых устройств. Вышеперечисленные достоинства позволяют применять его в системах M2M (Машинно-Машинное взаимодействие) и IIoT (Промышленный Интернет вещей).

Обмен сообщениями в протоколе MQTT осуществляется между клиентом (client), который может быть издателем или подписчиком (publisher/subscriber) сообщений, и брокером (broker) сообщений (например, Mosquitto MQTT). Издатель отправляет данные на MQTT брокер, указывая в сообщении определенную тему, топик (topic). Подписчики могут получать разные данные от

множества издателей в зависимости от подписки на соответствующие топики. Брокер можно создать на веб-ресурсах, на персональном компьютере, на микроконтроллере и т.п.

В проекте он реализован на удаленном сервере. В коде программы можно изменять время обмена данными. Следует учесть, что микроконтроллер отправляет данные каждые 2-3 секунды, так что, если будет потеря данных одной передачи, то сбоя не произойдет.

В качестве аппаратной платформы проекта была использована серия недорогих микроконтроллеров с низким энергопотреблением – ESP32, которые представляют собой систему на кристалле с интегрированным Wi-Fi и Bluetooth контроллерами.

Основное устройство, разработанное в рамках проекта, включает такие основные компоненты:

— Плата Espressif ESP32 Dev Module, которая имеет микроконтроллер ESP32, частоту 240МГц, флэш память 4МВ и RAM 320 KB, а также встроенный модуль WiFi, она является лучшей из известной в arduino uno.

— цифровой датчик влажности и температуры DHT11, состоящий из термистора и емкостного датчика влажности. Датчик содержит в себе АЦП для преобразования аналоговых значений влажности и температуры.

— Фоторезистор.

— DC-мотор (мотор постоянного тока). Электродвигатель работает только на постоянном токе от элементов питания.

— Светодиод – LED (диод, излучающий свет).

В программном коде для обработки показателей датчиков, фоторезистора, светодиода и DC-моторов используется библиотека "DHTesp.h".

Интерфейс вывода данных представлен на рисунке 2.



Рисунок 2 – Интерфейс вывода выходных данных на смартфон

Из приложения со смартфона приходят данные об изменении режима светодиода, скорость работы вентиляторов, положении курсора на экране и уровня громкости. Например: изменение положения курсора вправо/ вниз/ влево/ вверх - "DPR/jXpl" / "DPR/jYpl" / "DPR/jXmn" / "DPR/jYmn"; нажатие левой кнопкой мыши - "DPR/sololeftclick"; двойное нажатие левой кнопкой мыши - "DPR/doubleclick"; минимальная/ максимальная громкость - "DPR/soundOff" / "DPR/soundOn"; увеличение/ уменьшение громкости - "DPR/soundPUp" / "DPR/soundDown"; режим светодиода – "DPR/light"; скорость DC моторов – "DPR/speed".

Устройство выводит информацию о режиме работы в приложение на смартфоне.

В телефон через протокол MQTT поступают следующие значения:

- Температура из раздела "DPR/temp";
- Влажность из топика " DPR/vlazh";
- Уровень освещенности из топика " DPR/svet";
- Скорость вращения вентилятора.

В программе для микроконтроллера реализованы такие пользовательские функции для управления микроконтроллером:

- scanWiFi() для поиска окружающих WiFi сетей;
- WiFi\_Other\_Connect() для подключения лучшей сети WiFi из заранее объявленных;
- WiFi\_Connect() для подключения к сети указанной в качестве аргумента функции;

- `callback(char* topic, byte* payload, unsigned int length)` для получения значений из топиков, на которые подписаны пользователи;
- `BrokerConnection()` для подключения к брокеру, используется после разрыва соединения;
- `setup()` основная функция для платы, в которой определяются начальные настройки;
- `loop()` функция для платы, в которой циклически выполняются действия;
- Для управления компьютером:
- `subscribe(client: mqtt_client)` подключение топиков для получения данных;
- `on_message(client, userdata, msg)` для получения значений из топиков;
- `run()` бесконечный цикл для непрерывной работы программы.

При проектировании устройства спроектирована и разработана схема подключения всех аппаратных компонент, которая представлена на рисунке 3.

Питание 3.3 вольта поступает на плату через usb кабель. Датчик температуры и влажности подключен через разъем (пин-коннектор) 14. Фоторезистор подключен на 36 разъем с широтно-импульсной модуляцией и подтянут через резистор к земле. Мотор подключен на 25 разъем с широтно-импульсной модуляцией. Светодиод подключен на разъем 33.

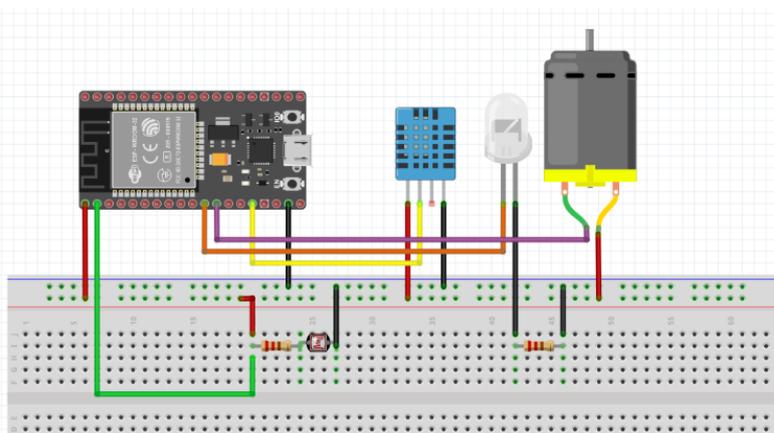


Рисунок 3 – Схема подключения устройств на монтажной плате

**Заключение.** Разработанная аппаратно-программная система предоставляет возможность удаленного управления различными устройствами, использовать, обрабатывать и передавать данные, расположенные в удаленном дата-центре, вне зависимости от местонахождения пользователя.

**Список использованных источников:**

1. Просто о протоколе MQTT в IIoT [Электронный ресурс]. – [Электронный ресурс] – Режим доступа: <https://zen.yandex.ru/media/id/5e16ce1278125e00b1ea677e/prosto-o-protokole-mqtt-v-iiot-5e25713ee6e8ef00b2d8938e> Дата доступа: 08.04.22
2. Espressif ESP32 Dev Module [Электронный ресурс] – [Электронный ресурс]. – Режим доступа: <https://docs.platformio.org/en/latest/boards/espressif32/esp32dev.html> Дата доступа: 08.04.22

UDC 004.771

## REMOTE DEVICE MANAGEMENT SYSTEM BASED ON THE ESP32 MICROCONTROLLER

*Marinovich D.A.<sup>1</sup>, Andreychuk A.O.<sup>2</sup>, Sitsko U.A.<sup>3</sup>*

*National Children's Technopark<sup>1</sup>,  
Affiliate «Minsk Radioengineering College» of the Belarusian State University of Informatics and Radioelectronics<sup>2</sup>,  
Institute of Information Technologies of the Belarusian State University of Informatics and Radioelectronics<sup>3</sup>,  
Minsk, Republic of Belarus*

*Sitsko U.A. – Master of Engineering Sciences*

**Annotation.** The article discusses the features of the MQTT messaging protocol. The main control functions of the ESP32 microcontroller have been developed. A model of a remote device management system based on the MSP 32 microcontroller with the development of the client and server parts of the web application is proposed.

**Keywords.** Programming of microcontrollers, MQTT protocol, ESP32 microcontroller