

## 4.3. Лабораторная работа: MQTT в Mbed

Site: [Samsung Innovation Campus](#)  
Course: Internet of Things  
Book: 4.3. Лабораторная работа: MQTT в Mbed

Printed by: Антон Файтельсон  
Date: Saturday, 21 October 2023, 7:38 PM

## Table of contents

[4.3.1. Описание лабораторной работы](#)

[4.3.2. Код демо-примера](#)

[4.3.3. Запуск демо-сценария](#)

### 4.3.1. Описание лабораторной работы

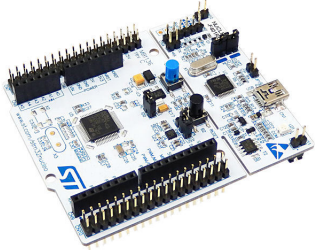
Конечно же, интересно теперь посмотреть, как отправлять и получать данные через MQTT, используя микроконтроллерную плату.

Сделаем следующий пример:

1. Отправка данных по WiFi на локальный MQTT-сервер
2. Получение данных по WiFi с локального MQTT-сервера

#### Список необходимых компонентов

- Плата STM32 Nucleo



- WiFi на этой плате (через ESP8266 или иной WiFi-модуль)



#### Список программных требований

- Для выполнения этого практикума у вас уже должен быть подключен WiFi к плате и всё должно работать, то есть должен выполняться пример `mbed-os-example-wifi`.
- Также у вас должен работать локальный MQTT-сервер на вашем компьютере, и вам должен быть известен его IP-адрес.

### 4.3.2. Код демо-примера

удем пробовать подключиться к MQTT-серверу, используя наш уже знакомый пример работы с WiFi. Считаем, что уже выполняется подключение к WiFi-сети и работа с сокетами.

Для работы примера в проект понадобится импортировать библиотеку mbed-mqtt:

<https://github.com/ARMmbed/mbed-mqtt>

Демо-функцию назовем mqtt\_demo. Это немного переработанный пример [HelloMQTT](#), его отличие в том, что

- а) из него убран лишний функционал в виде печати на LCD-экран
- б) исправлена ошибка с сокетами в соответствии с более новой документацией (видимо, пример писался для более старой версии библиотеки)
- в) убран пример с QoS 2, поскольку он по-видимому, нерабочий (другие пользователи тоже жалуются на проблемы с ним)

Суть примера очень простая. Создается MQTT-клиент, который подписывается на топик mbed-sample, и сам посылает туда же сообщения. Есть callback-функция messageArrived, которая срабатывает при получении нового сообщения и увеличивает счетчик. Демо-сценарий такой: клиент посылает в свой же топик два сообщения, и ждет третьего извне. Вы отправляете его сами, после чего работа программы завершается.

Вам понадобится вставить следующий код в тело вашей программы - включение библиотек и две функции: одна call-back, вторая собственно демо-сценарий:

```
#define MQTTCLIENT_QOS2 1

#include "MQTTmbed.h"
#include "MQTTClientMbedOs.h"

int arrivedcount = 0;

void messageArrived(MQTT::MessageData& md)
{
    MQTT::Message &message = md.message;
    printf("Message arrived: qos %d, retained %d, dup %d, packetid %d\r\n", message.qos, message.retained, message.dup,
message.id);
    printf("Payload %.s\r\n", message.payloadlen, (char*)message.payload);
    ++arrivedcount;
}

void mqtt_demo(NetworkInterface *net)
{
    float version = 0.6;
    char* topic = "mbed-sample";

    TCPSocket socket;
    MQTTClient client(&socket);

    SocketAddress a;
    char* hostname = "192.168.0.11";
    net->gethostbyname(hostname, &a);
    int port = 1883;
    a.set_port(port);

    printf("Connecting to %s:%d\r\n", hostname, port);

    socket.open(net);
    printf("Opened socket\n\r");
    int rc = socket.connect(a);
    if (rc != 0)
        printf("rc from TCP connect is %d\r\n", rc);
    printf("Connected socket\n\r");

    MQTTPacket_connectData data = MQTTPacket_connectData_initializer;
    data.MQTTVersion = 3;
    data.clientID.cstring = "mbed-sample";
    data.username.cstring = "testuser";
    data.password.cstring = "testpassword";
    if ((rc = client.connect(data)) != 0)
        printf("rc from MQTT connect is %d\r\n", rc);

    if ((rc = client.subscribe(topic, MQTT::QOS2, messageArrived)) != 0)
        printf("rc from MQTT subscribe is %d\r\n", rc);

    MQTT::Message message;

    // QoS 0
    char buf[100];
    sprintf(buf, "Hello World!  QoS 0 message from app version %f\r\n", version);
    message.qos = MQTT::QOS0;
    message.retained = false;
    message.dup = false;
    message.payload = (void*)buf;
    message.payloadlen = strlen(buf)+1;
    rc = client.publish(topic, message);
    while (arrivedcount < 1)
        client.yield(100);

    // QoS 1
    sprintf(buf, "Hello World!  QoS 1 message from app version %f\r\n", version);
    message.qos = MQTT::QOS1;
    message.payloadlen = strlen(buf)+1;
```

```
rc = client.publish(topic, message);
while (arrivedcount < 2)
    client.yield(100);

while (arrivedcount < 3)
    client.yield(100);

if ((rc = client.unsubscribe(topic)) != 0)
    printf("rc from unsubscribe was %d\r\n", rc);

if ((rc = client.disconnect()) != 0)
    printf("rc from disconnect was %d\r\n", rc);

socket.close();

printf("Version %.2f: finish %d msgs\r\n", version, arrivedcount);

return;
}
```

И в main добавьте вызов этой функции в самом конце, после всех предыдущих тестов, перед отключением WiFi:

```
mqtt_demo(wifi);
```

Как видите, в коде ничего сложного нет, в значительной степени это повторение предыдущего демо-кода с сокетами.

### 4.3.3. Запуск демо-сценария

Соберите программу и запустите, не забыв прописать в переменной hostname адрес своего MQTT-сервера.

Если всё успешно, то увидите в консоли следующее:

```
Connecting to 192.168.0.11:1883
Opened socket
Connected socket
Message arrived: qos 0, retained 0, dup 0, packetid 1000
Payload Hello World! QoS 0 message from app version 0.600000
Message arrived: qos 1, retained 0, dup 0, packetid 1
Payload Hello World! QoS 1 message from app version 0.600000
```

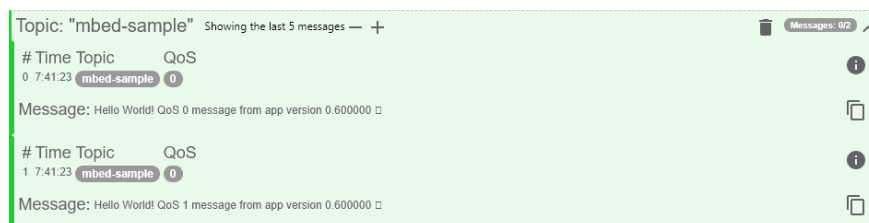
Если увидите такую ошибку:

```
rc from MQTT subscribe is -3004
```

То это означает, что вы скорее всего неправильно указали адрес сервера. Проверьте, что MQTT-сервер поднят и работает.

А если вы заново запустите эту программу, но уже подписавшись на топик mbed-sample, увидите следующее: программа отправила два сообщения Hello World в этот топик, одно с QoS 0 и одно с QoS 1.

Пример, как это выглядит в MQTTLens (можно смотреть в любом другом клиенте, графическом или консольном):



Дальше программа останавливается и ждет третьего сообщения. Отправьте его самостоятельно, и увидите в консоли, что ваша плата его получила и отчиталась об этом:

```
Message arrived: qos 0, retained 0, dup 0, packetid 1000
Payload hello
Version 0.60: finish 3 msgs
```

[Reset user tour on this page](#)