



08. Сетевое взаимодействие по MQTT

Практические задания.

Цель: научиться разрабатывать программы, которые отправляют и принимают данные через брокер MQTT.

Задача: Разработать программу чтения данных по протоколу MQTT.

№ 08.0

1. Установите брокер и клиенты Mosquitto на свою Raspberry Pi, выполнив в терминальном окне команды:
`sudo apt-get update`
`sudo apt-get install mosquitto mosquitto-clients`
2. Откройте 1-е терминальное окно. Запустите проверочный командный скрипт-подписчик, который ждёт сообщения в теме `sensor/temperature`:
`~/CodeClub-IoT/samples/mqtt_sub.sh`
3. Откройте 2-е терминальное окно. Запустите проверочный командный скрипт-издатель, который отправит сообщение в тему `sensor/temperature`:
`~/CodeClub-IoT/samples/mqtt_pub.sh`
4. В 1-м окне должно быть выведено значение сообщения: 25.5.
5. Закройте терминальное окно со скриптом-издателем, но пока не закрывайте окно, где работает скрипт-подписчик: он понадобится для проверки.

№ 08.1

1. Запустите редактор **Geany** из раздела «Программирование» в главном меню.
2. Откройте в редакторе **Geany** пример программы, публикующей сообщения на брокере MQTT: `~/CodeClub-IoT/samples/mqtt_pub.rb`.
3. Запустите программу на выполнение из раздела меню «Сборка», пункт «Execute» (выполнить) и проверьте её работу: в терминальном окне, где запущен скрипт-подписчик, должно появиться значение температуры.
4. Создайте в редакторе **Geany** новую программу, которая будет отправлять на брокер Mosquitto в заданную тему сообщения с показаниями температуры со встроенного термодатчика Raspberry Pi:

```
#!/usr/bin/ruby
require 'mqtt' # подключить библиотеку для работы с mqtt
require 'thermal_sensor' # библиотека аботы с термодатчиком
BROKER = '127.0.0.1' # адрес или имя брокера
sensor = RaspberryPi::ThermalSensor.new # создать объект «датчик»
while true do # в бесконечном цикле
  sensor.read_data # считать показание датчика
  t = sensor.celsius.to_s # преобразовать его в строку
  MQTT::Client.connect(BROKER) do |client| # подключиться к брокеру
    client.publish('sensor/temperature', t) # опубликовать
  end
  sleep 1
end
```

5. Сохраните её под именем `~/CodeClub-IoT/samples/mqtt_thermo.rb`
6. Запустите программу на выполнение из раздела меню «Сборка», пункт «Execute» (выполнить) и убедитесь, что она работает: в терминальном окне, где запущен скрипт-подписчик, должно появиться значение температуры. Не закрывайте терминальное окно, где работает `mqtt_thermo.rb`.
7. Откройте в редакторе **Geany** пример программы, читающей сообщения на брокере MQTT: `~/CodeClub-IoT/samples/mqtt_sub.rb`.

8. Запустите её, чтобы проверить, что она читает сообщения с брокера.
9. Измените её так, чтобы она читала сообщения и выводила их в бесконечном цикле.
10. Проверьте её работу.
11. Закройте все ненужные командные окна.

№ 08.2: Чат на MQTT.

1. Пусть одна из Raspberry Pi будет брокером MQTT. Узнайте IP-адрес её беспроводного интерфейса (wlan0) командой:

```
ip address | grep wlan0 | grep inet
```

2. Обменяйтесь сообщениями с товарищами через этот брокер, публикуя сообщения в определённой теме (договоритесь, как она будет называться), а затем читая сообщения из этой темы.

*Проще всего это сделать в 2-х командных окнах с помощью командных клиентов **mosquitto_sub** и **mosquitto_pub**. С какими параметрами вызывать эти утилиты, можно посмотреть в скриптах **~/CodeClub-IoT/samples/mqtt_sub.sh** и **~/CodeClub-IoT/samples/mqtt_pub.sh**.*

3. Проверьте, что можно читать сообщения из этой темы с разных узлов сети одновременно.