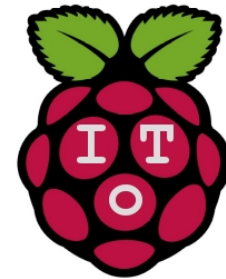




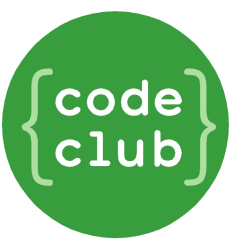
Internet of Things



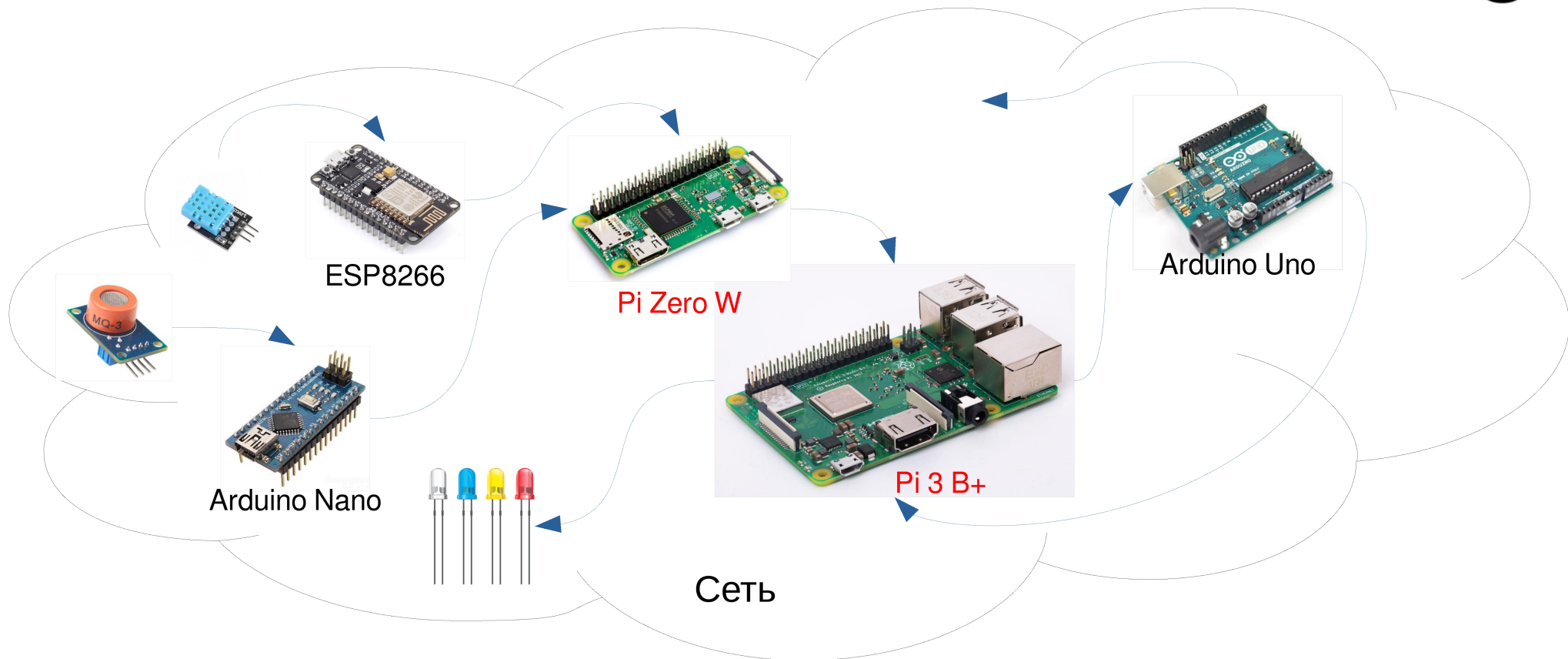
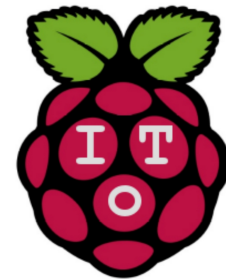
MQTT в сетях IoT

Шадринск
2018-2019

М. В. Шохирев

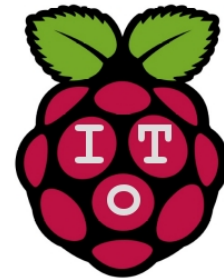


IoT = сеть контроллеров!





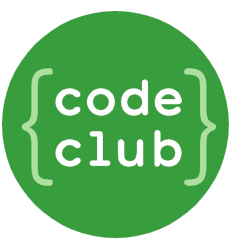
Сети IoT: всё плохо!



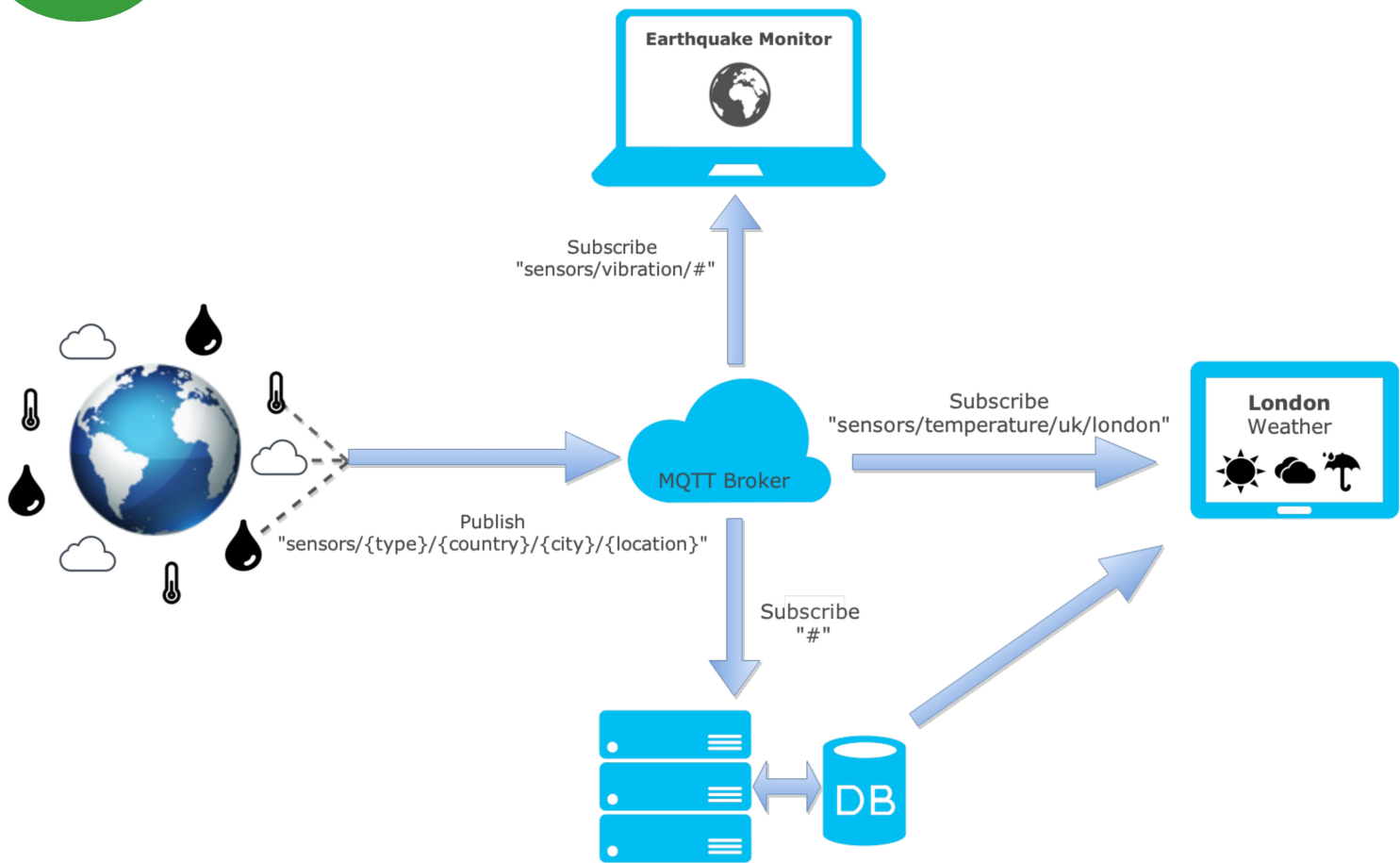
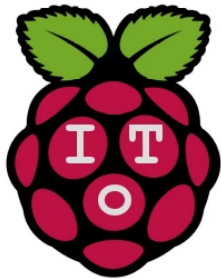
В сетях IoT типичны такие трудности:

- используются различные аппаратные платформы
- используются различные программные средства
- необходимо увязывать решения от разных производителей
- применяются разные сетевые технологии
- сеть состоит из отдельных удалённых фрагментов
- связь нестабильна: пропадает радио-сигнал и т. п.
- некоторые узлы сети могут быть временно недоступны
- случаются перебои в электроснабжении оборудования
- необходимо экономить заряд аккумулятора

Требуется повышенная надёжность работы: исправление ошибок, повторная отправка данных, дублирование каналов передачи данных, резервирование узлов сети, самовосстанавливаемость систем, защита от злоумышленников.



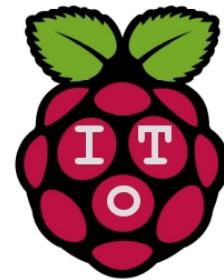
MQTT



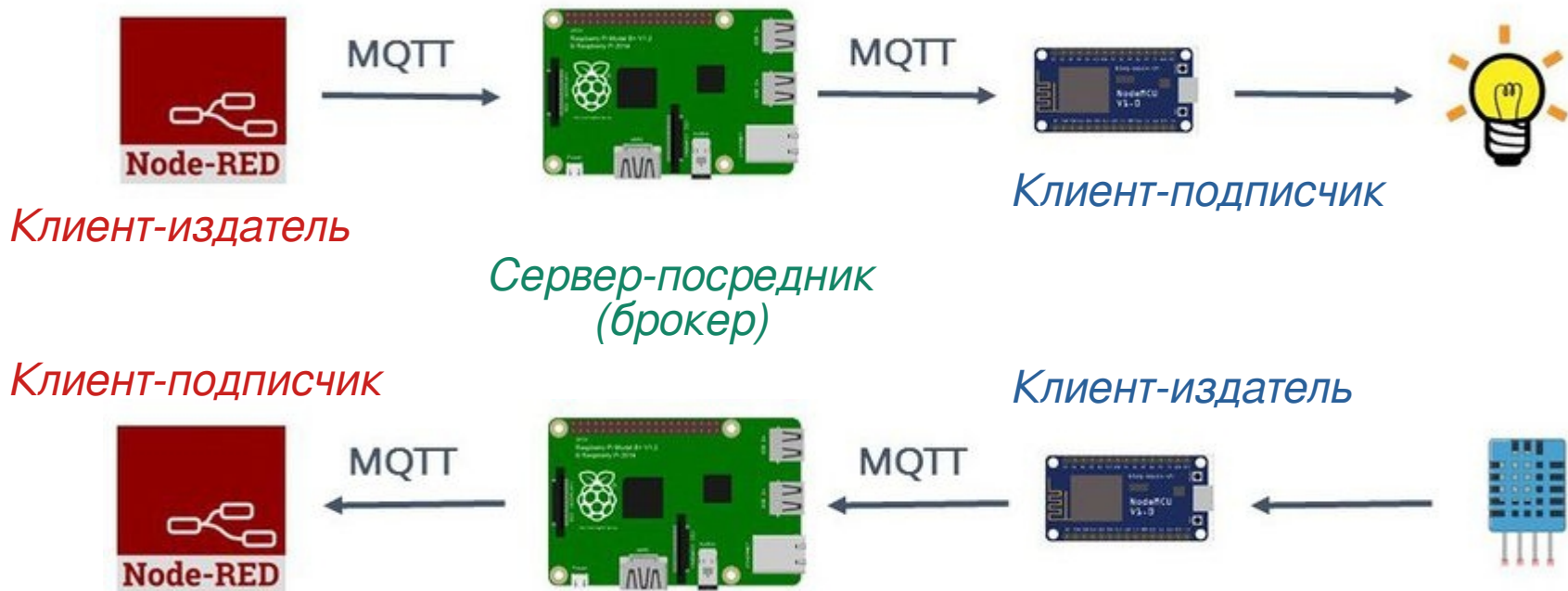
MQTT (Message Queuing Telemetry Transport) — сетевой протокол для передачи телеметрии при межмашинном взаимодействии (M2M = Machine-to-Machine). Его легко освоить и просто применять, он не сильно загружает каналы передачи данных, надёжен в работе при частых потерях связи, легко встраивается в любую систему, кросс-платформенный.



Модель MQTT

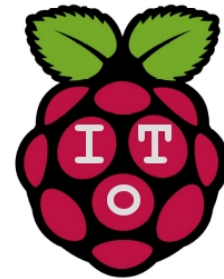


Протокол **MQTT** (Message Queuing Telemetry Transport) реализует обмен сообщениями по принципу «издатель-подписчик» с использованием посредника-брокера.



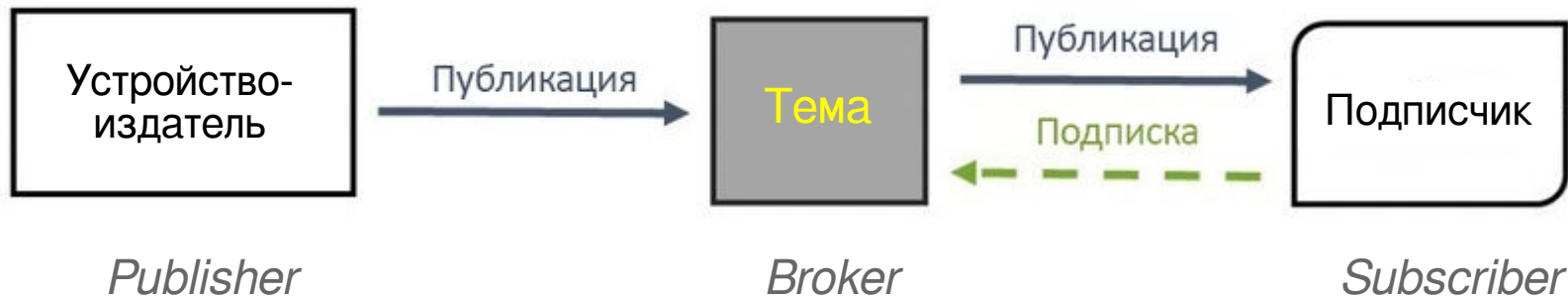


Издатель-подписчик

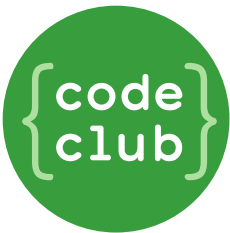


Модель «издатель-подписчик» означает:

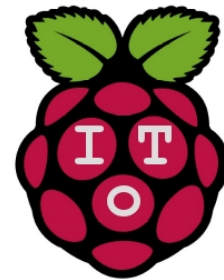
- одно устройство может публиковать (publish) сообщения для других устройств;
- устройство может подписаться (subscribe) на какую-нибудь тему (topic), чтобы получать интересующие его сообщения.



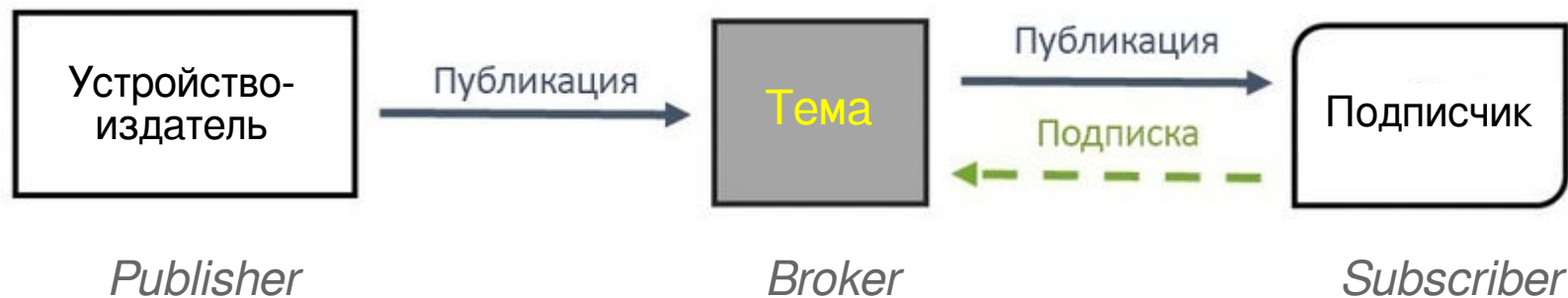
Когда одно устройство-издатель публикует сообщение в теме, а второе устройство-подписчик подписано на эту тему, то сообщение, опубликованное первым устройством, будет получено вторым устройством.



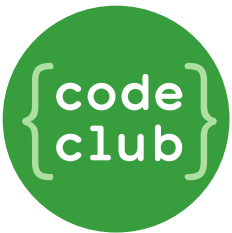
Темы



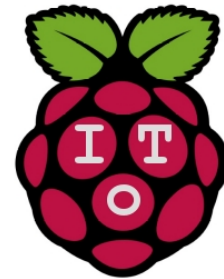
Темы (topic) — это способ группировать и структурировать сообщения, передаваемые через брокер. Имена тем и подтем разделяются символом «/». При именовании тем учитывается регистр букв (заглавные и строчные).



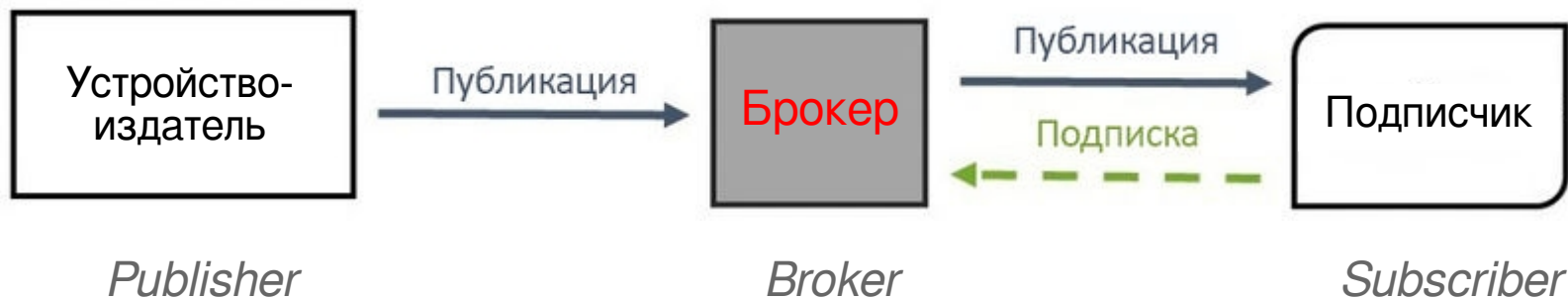
Темы могут называться строками на национальных языках, например, на русском, в кодировке UTF-8. Но рекомендуется называть темы осмысленными словами на английском языке в латинице.



Брокер (посредник)



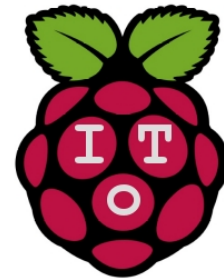
Брокер (Broker) – программа, которая организует публикацию сообщений в темах, получение сообщений от издателей, их отбор по подпискам и отправку подписчикам.



Имеется много разных программ-брокеров MQTT. Один из наиболее популярных брокеров, Mosquitto, реализован для **Raspberry Pi**.



Управление по MQTT



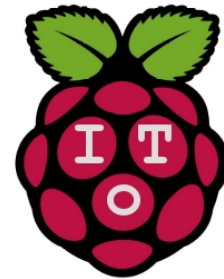
Например, чтобы включить светильник в офисе, нужно в управляющей клиентской программе, например, Node-RED на Raspberry Pi, опубликовать сообщение «ON» в теме `home/office/lamp` на брокере (который также развёрнут на Raspberry Pi).

Исполняющее клиентское устройство, например, модуль ESP8266, должно быть подписано на эту тему, чтобы получать сообщения, и когда оно получит опубликованное сообщение «ON», оно включит светильник.





Mosquitto



Mosquitto™ — брокер MQTT с открытым исходным кодом, развиваемый организацией Eclipse Foundation. Это легковесный сервер, который обеспечивает обмен сообщениями по протоколу MQTT. Он прекрасно подходит для обмена данными в сетях IoT между датчиками с низким энергопотреблением и мобильными устройствами, встраиваемыми компьютерами или микроконтроллерами. Он реализован для многих платформ, включая Raspberry Pi.

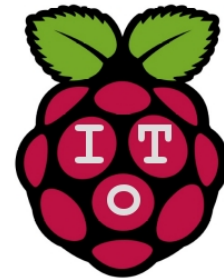
Проект **Mosquitto** также предоставляет библиотеку на языке C для реализации клиентов MQTT, а также популярные команды `mosquitto_pub` и `mosquitto_sub` для работы из командной строки.

0. Установить сервер и клиентов Mosquitto:

```
sudo apt-get update  
sudo apt-get install mosquitto mosquitto-clients
```



Mosquitto ← команды

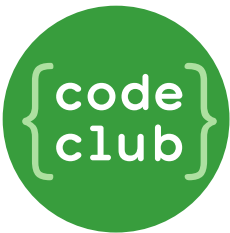


```
pi@RPi06: ~  
File Edit Tabs Help  
pi@RPi06:~ $ mosquitto_sub -h localhost -t "sensor/temperature"  
23.5  
23.4  
[ ]
```

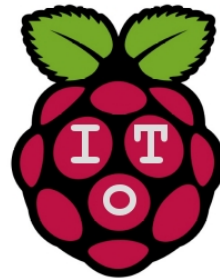
```
pi@RPi06: ~  
File Edit Tabs Help  
pi@RPi06:~ $ mosquitto_pub -h localhost -t "sensor/temperature" -m 23.5  
pi@RPi06:~ $ mosquitto_pub -h localhost -t "sensor/temperature" -m 23.4  
pi@RPi06:~ $
```

1. Подписаться (SUBscribe) на нужную тему:
`mosquitto_sub -h localhost -t "sensor/temperature"`

2. Опубликовать (PUBlish) в нужной теме:
`mosquitto_pub -h localhost -t "sensor/temperature" -m 23.5`
`mosquitto_pub -h localhost -t "sensor/temperature" -m 23.4`



Mosquitto ← bash



Подписаться (SUBscribe) на тему:

```
mosquitto_sub -h localhost -t "sensor/temperature"
```

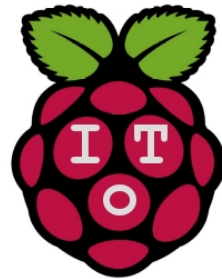
Командный файл, публикующий показания температуры со встроенного датчика Raspberry Pi:

```
#!/bin/bash
while true
do
    temperature=`cat /sys/class/thermal/thermal_zone0/temp`
    degree=$((temperature/1000))    # целая часть
    fraction=$((temperature%1000)) # дробная часть
    `mosquitto_pub -h localhost -t "sensor/temperature" -m $degree.$fraction`
    sleep 1
done
```

При запуске на одном компьютере следует сначала запустить брокера Mosquito, затем клиента-подписчика, затем клиента-издателя.



Mosquitto → Ruby



Приём из темы сообщений с показаниями температуры от брокера Mosquitto:

```
#!/usr/bin/ruby
# https://github.com/njh/ruby-mqtt
# sudo gem install mqtt
```

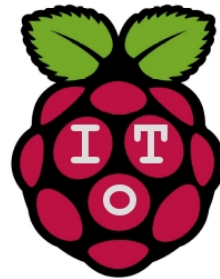
```
require 'mqtt'                                # подключить библиотеку для работы с mqtt
```

```
BROKER = '10.36.254.16' # 'localhost'           # адрес или имя брокера
PORT = 1883              # номер порта на брокере
```

```
MQTT::Client.connect(BROKER, PORT) do |client| # подключиться к брокеру
  client.subscribe('sensor/temperature')          # подписаться на тему
  while true do                                    # в бесконечном цикле
    topic, message = client.get                    # прочитать тему и сообщение
    puts "#{topic}: #{message}"                    # вывести значения на экран
  end
end
```



Mosquitto ← Ruby



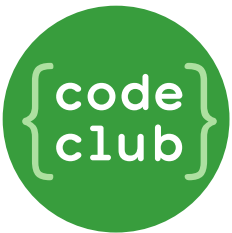
Отправка на брокер Mosquitto в заданную тему сообщений с показаниями температуры со встроенного термодатчика Raspberry Pi:

```
#!/usr/bin/ruby
require 'mqtt'           # подключить библиотеку для работы с mqtt
require 'thermal_sensor' # подключить библиотеку для работы с термодатчиком

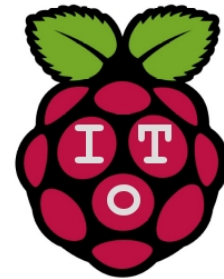
BROKER = '10.36.254.16' # адрес или имя брокера

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 в теме сообщение
  end
  sleep 1
end
```



Источники



Книги:

- Петин В. **Arduino и Raspberry Pi в проектах Internet of Things**, 2-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2018. - 432 с.: ил.

Интернет-ресурсы:

- Raspberry Pi:Настройка/Что_такое_протокол_MQTT_и_как_он_работает
- Платформа ARM и брокер MQTT, как современная основа решений для Интернета вещей
- MQTTBox
- Configuring MQTT on the Raspberry Pi
- MQTT gem для Ruby - <https://github.com/njh/ruby-mqtt>
- Програмируем управление освещением по датчикам движения и освещения на Node-RED