

ОБМЕН ИНФОРМАЦИЕЙ МЕЖДУ МОБИЛЬНЫМ ПРИЛОЖЕНИЕМ И МИКРОКОНТРОЛЛЕРОМ ЧЕРЕЗ ПРОТОКОЛ MQTT

А.В. ХАРЧЕНКО, В.С. ГАВРИЛЕНКО

Белорусский государственный университет информатики и радиоэлектроники, филиал «Минский радиотехнический колледж», Республика Беларусь

Поступила в редакцию 18 марта 2023

Аннотация. Рассмотрен протокол передачи информации между мобильными устройствами MQTT для передачи данных как между несколькими устройствами на микроконтроллерах, так и для подключения мобильных устройств на микроконтроллерах к смартфону. Показана, схема обмена информацией между клиентом и сервером MQTT протокола. Рассмотрены типы и строение сообщений для работы с протоколом.

Ключевые слова: протокол MQTT, IoT, MQTT сообщение, микроконтроллер, брокер, клиент, сервер.

Введение

Новые интернет технологии стремительно развиваются в современном мире. Широкую популярность, а, следовательно, и востребованность приобретают разработки на базе микроконтроллеров с возможностью управления мобильным приложением через беспроводную связь. На сегодняшний день существует множество различных протоколов, позволяющих подключать мобильные устройства (в том числе на базе микроконтроллеров) к интернету, и соединять их между собой. Одним из таких протоколов является протокол MQTT, который используются в Internet of Things (IoT).

MQTT (Message Queuing Telemetry Transport) – это протокол, сделанный конкретно для IoT. Он предназначен для обмена информацией между разными устройствами и модулями. Отвечает за безопасность соединения, скорость передачи данных и практическое функционирование систем и программ.

Описание принципа работы

Система связи, построенная на MQTT, состоит из сервера-издателя, сервера-брокера и одного или нескольких клиентов. Издатель не требует каких-либо настроек по количеству или расположению подписчиков, получающих сообщения. Кроме того, подписчикам не требуется настройка на конкретного издателя. В системе может быть несколько брокеров, распространяющих сообщения [1]. Схема обмена информации между клиентом и сервером представлена на рис. 1.



Рис. 1. Схема обмена информации между клиентом и сервером

Есть множество способов настройки клиента для подключения через брокера MQTT. Один из них представлен ниже в виде кода.

```
var options = {
  keepalive: 60,
  username: 'FIRST_HALF_OF_API_KEY',
  password: 'SECOND_HALF_OF_API_KEY',
  port: 8883
};
var client = mqtt.connect('mqtts:mqtt.ably.io', options);
```

Все данные опубликованные или полученные брокером MQTT, будут закодированы в двоичном формате, поскольку MQTT является бинарным протоколом. Это означает, что для получения исходного содержимого нужно интерпретировать сообщение. Ниже представлен способ получение информации с помощью Ably и JavaScript [2].

```
var ably = new Ably.Realtime('REPLACE_WITH_YOUR_API_KEY');
var decoder = new TextDecoder();
var channel = ably.channels.get('input');
channel.subscribe(function(message) {
  var command = decoder.decode(message.data);
});
```

Структура сообщений

Всего в протоколе MQTT существует 15 типов сообщений, которые представлены в табл. 1, где «К» – клиент, а «С» – сервер.

Табл. 1. Типы сообщений в фиксированном заголовке

Тип сообщения	Значение	Направление передачи	Описание
Reserved	0000 (0)	нет	Зарезервирован
CONNECT	0001 (1)	К → С	Запрос клиента на подключение к серверу
CONNACK	0010 (2)	К ← С	Подтверждение успешного подключения
PUBLISH	0011 (3)	К ← С, К → С	Публикация сообщения
PUBACK	0100 (04)	К ← С, К → С	Подтверждение публикации
PUBREC	0101 (5)	К ← С, К → С	Публикация получена
PUBREL	0110 (6)	К ← С, К → С	Разрешение на удаление сообщения
PUBCOMP	0111 (7)	К ← С, К → С	Публикация завершена
SUBSCRIBE	1000 (8)	К → С	Запрос на подписку
SUBACK	1001 (9)	К ← С	Запрос на подписку принят
UNSUBSCRIBE	1010 (10)	К → С	Запрос на отписку
UNSUBACK	1011 (11)	К ← С	Запрос на отписку принят
PINGREQ	1100 (12)	К → С	PING запрос
PINGRESP	1101 (13)	К ← С	PING ответ
DISCONNECT	1110 (14)	К → С	Сообщение об отключении от сервера
Reserved	1111 (15)	нет	Зарезервирован

MQTT сообщение состоит из нескольких частей: фиксированный заголовок, переменный заголовок и данные [3].

Фиксированный заголовок присутствует во всех сообщениях. в то время, как переменный заголовок и данные – присутствуют только в определенных сообщениях. Строение фиксированного заголовка представлено на рис. 2.

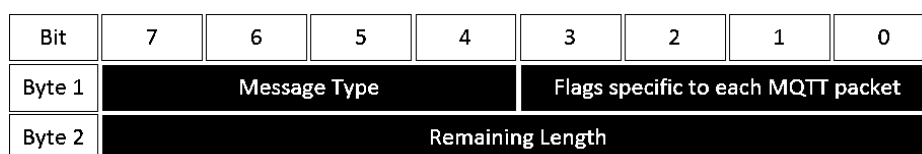


Рис. 2. Строение фиксированного заголовка

«Message Type» – это тип сообщения. «Flags specific to each MQTT packet» – эти 4 бита отведены под вспомогательные флаги, наличие и состояние которых зависит от типа сообщения. «Remaining Length» – представляет длину текущего сообщения, может занимать от 1 до 4 байта.

Четыре старших бита первого байта фиксированного заголовка отведены под специальные флаги, они изображены на рис. 3.

Bit	7	6	5	4	3	2	1	0
Byte 1	Message type				DUP	QoS	QoS	Retain
Byte 2	Remaining Length							

Рис. 3. Расположений страших битов

«DUP» – флаг дубликата устанавливается, когда клиент или MQTT брокер совершает повторную отправку пакета. При установленном флаге переменный заголовок должен содержать Message ID. «QoS»– качество обслуживания. «RETAIN»– при публикации данных с установленным флагом retain, брокер сохранит его. При следующей подписке на этот топик брокер незамедлительно отправит сообщение с этим флагом. Используется только в сообщениях с типом «PUBLISH».

Переменный заголовок содержится не во всех заголовках. В нем помещаются следующие данные. «Packet identifier» – идентификатор пакета, присутствующий во всех типах сообщений, кроме: «CONNECT», «CONNACK», «PUBLISH», «PINGREQ», «PINGRESP», «DISCONNECT». «Protocol name» – название протокола (только в сообщениях типа «CONNECT»). «Protocol version» – версия протокола (только в сообщениях типа «CONNECT»). «Connect flags» – флаги, указывающие на поведение клиента при подключении. Строение переменного заголовка представлено на рис. 4.

Bit	7	6	5	4	3	2	1	0
Byte 8	User name	Password	Will Retain	Will QoS		Will Flag	Clean Session	Reserved

Рис. 4. Строение переменного заголовка

«User name» – при наличии этого флага в «нагрузке» должно быть указано имя пользователя. «Password» – при наличии этого флага в «нагрузке» должен быть указан пароль. «Will Retain» – при установке в 1, брокер хранит у себя «Will Message». «Will QoS» – качество обслуживания для «Will Message», при установленном флаге «Will Flag», «Will QoS» и «Will retain» являются обязательными. «Will Flag» – при установленном флаге, после того, как клиент отключится от брокера без отправки команды «DISCONNECT» (в случаях непредсказуемого обрыва связи, например), брокер оповестит об этом всех подключенных к нему клиентов через «Will Message».

«Clean Session» – необходим для очистки сессии. При установленном «0» брокер сохранит сессию, все подписки клиента, а также передаст ему все сообщения с «QoS1» и «QoS2», которые были получены брокером во время отключения клиента, при его следующем подключении. Соответственно при установленной «1», при повторном подключении клиенту будет необходимо заново подписываться.

Содержание и формат данных, передаваемых в MQTT сообщениях, определяются в приложении. Размер данных может быть вычислен путем вычитания из «Remaining Length» длины переменного заголовка.

Заключение

MQTT предоставляет способ создания иерархии каналов связи – так называемую «ветвь с листьями». Всякий раз, когда у издателя есть новые данные для распространения среди клиентов, сообщение сопровождается примечанием контроля доставки. Клиенты более высокого уровня могут получать каждое сообщение, в то время как клиенты более низкого уровня могут получать

сообщения, относящиеся только к одному или двум базовым каналам, «ответвляющимся» в нижней части иерархии. Это облегчает обмен информацией размером от двух байт до 256 мегабайт. Данный протокол защищает от различных сбоев и неполадок при передаче данных позволяя обмениваться информацией между устройствами, а также выполнять систематизацию локальных сетей в интернете.

EXCHANGE OF INFORMATION BETWEEN THE MOBILE APP AND THE MCU VIA MQTT PROTOCOL

A.V. HARCHENKO, V.S. GAVRILENKO

Abstract. The protocol for transferring information between mobile devices MQTT is considered for data transfer both between several devices on microcontrollers and for connecting mobile devices on microcontrollers to a smartphone. The scheme of information exchange between the client and the server of the MQTT protocol is shown. The types and structure of messages for working with the protocol are considered.

Keywords: MQTT protocol, IoT, MQTT message, microcontroller, broker, client, server.

Список литературы

1. Chen W.J., Gupta R. // Responsive Mobile User Experience Using MQTT and IBM MessageSight, 2014.
2. Boyd B., Gauci J. // Building Real-time Mobile Solutions with MQTT and IBM MessageSight, 2014.
3. Pulver T. // Hands-On Internet of Things with MQTT: Build connected IoT devices with Arduino and MQ Telemetry Transport, 2019