

# 基于ZigBee与MQTT的物联网网关通信框架的设计与实现

谭方勇, 王 昂, 刘子宁

(苏州市职业大学, 江苏 苏州 215104)

**摘 要:** 提出了一种基于ZigBee与MQTT的物联网网关协议转换的通信框架的设计方法, 主要目的是解决多协议、多消息格式的兼容与扩展的问题。用C#语言定义了符合框架的传感器设备类和数据类, 并利用“观察者模式”和“简单工厂模式”的设计思想, 解决了多协议扩展和通信效率的问题。

**关键词:** ZigBee; MQTT; 物联网网关; 通信框架; JSON

**中图分类号:** TP393.03

**文献标识码:** A

## Design and Implementation of the IOT Gateway Communication Framework Based on ZigBee and MQTT

TAN Fangyong, WANG Ang, LIU Zining

(Suzhou Vocational University, Suzhou 215104, China)

**Abstract:** The paper proposes a design method of the communication framework for internet of things gateway protocol conversion based on ZigBee and MQTT. The main purpose is to solve the compatibility and expansion problems of multi-protocol and multi-message formats. It defines the framework of the sensor device class and data class with the C# language, and solves the low efficiency problem of multi-protocol extension and communication through the design philosophy of Observer Pattern and Simple Factory Pattern.

**Keywords:** ZigBee; MQTT; IOT Gateway; communication framework; JSON

### 1 引言(Introduction)

随着“智慧地球”“感知中国”等一系列有关物联网相关的口号的提出, 物联网的在各行各业应用方案也在不断地被提出并实施, 这也使人们的生活变得更加便利和智能化<sup>[1]</sup>。此外, 因为无线传感器结构简单、成本低廉的特点, 使得其在物联网中成为一个部分, 但因为其使用的ZigBee协议与现有的互联网的TCP/IP协议是属于两种不同的异构网络的协议<sup>[2]</sup>, 两者之间如有实时、可靠的传输数据, 也成了无线传感网, 乃至物联网能否快速发展的一个重要因素。本文将MQTT协议融合到物联网网关协议转换的设计中, 通过ZigBee协议与MQTT协议的对接, 解决物联网在不同网络结构、不同通信方式以及不同协议环境下的对接问题。

### 2 相关研究背景(Related research background)

物联网网关技术是物联网的关键技术之一, 近年来国内外的研究也较多, 在感知层的ZigBee协议与上层协议的对接转换上提出了很多网关的实现技术和方案, 如将ZigBee与

Wi-Fi进行对接来实现智能家居网关的设计<sup>[3]</sup>; 将ZigBee与TCP协议之间进行有效转换来实现网关软硬件设计<sup>[4]</sup>; 也有文献提出一种基于ZigBee技术和GPRS技术的网关设计方案, 实现监控中心对现地水位的实时监控<sup>[5]</sup>。以上的这类文献主要研究的是针对某种具体物联网的应用场合下对物联网网关的设计, 在一定程度上也提高了物联网网关的效率, 但对于异构网络的不同上层协议与底层ZigBee协议的对接的研究还比较少。本文将主要研究物联网网关设计中ZigBee与MQTT协议的对接, 从而实现物联网感知层与不同应用层协议之间的通信。

### 3 物联网网关相关协议分析(Analysis of internet of things gateway protocol)

ZigBee是物联网感知层的协议之一, 它具有近距离、低功耗、低复杂度、低数据传输率以及自组织等特点, 其低功耗的特点可以使其在有限能源供应的场合下能够存活更长的时间; 自组织的特点在物联网感知节点的部署中可以通过

多跳的方式来进行数据传输,而不需要专门部署固定的基础设施。相比于其他物联网感知层的短距离无线通信协议,如Wi-Fi、蓝牙等,虽然Wi-Fi组网方便,但传输快、流量大的特点会导致其具有太大的功耗与成本开销。对于物联网感知层来说,对传输的速度与流量并没有很大要求,反而更需要低流量的缓解流量开销的问题;蓝牙协议虽然也是可以做到比较低的功耗,但其组网上要比ZigBee复杂麻烦得多,不适合物联网下大量节点的组网与数据传输。

MQTT(Message Queuing Telemetry Transport Protocol,消息队列遥测传输协议),它是IBM公司所推出的一种基于轻量级代理的发布/订阅模式的消息传输协议,运行在TCP协议栈之上,能够提供有序、可靠、双向连接的网络连接保证<sup>[6]</sup>。它具有开放、简单和易于实现等特点,因此更适合于嵌入式系统以及网络带宽低等资源受限的环境中应用,此外,它也被广泛用于遥感勘测、智能家居、能源监测和医疗应用程序等各个领域,对于M2M或物联网网关通信来说却是一个不错的选择。

## 4 基于ZigBee与MQTT协议的物联网网关总体设计(Design of internet of things gateway based on ZigBee and MQTT protocol)

### 4.1 网关系统总体结构设计

如图1所示,基于ZigBee/MQTT物联网网关位于网络传输层,其嵌入式网关/PC网关应用程序将串口传来的传感器数据转换为符合JSON消息格式的数据,通过MQTT的发布者角色发布到MQTT代理服务器。MQTT代理服务器一方面对数据进行持久化操作,将数据以文档形式存入MongoDB数据库。另一方面则将网关应用程序发布的数据消息推送到已经订阅的PC或移动端的客户端。而应用层则是PC或移动端的客户端接收服务器推送的下位机发布的传感器数据或发送控制传感器的命令给代理服务器。

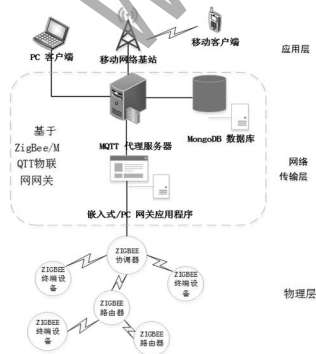


图1 基于ZigBee/MQTT物联网网关总体结构

Fig.1 The main architecture of IOT gateway based on ZigBee/MQTT

### 4.2 ZIGBEE与MQTT协议之间的通信框架设计

#### (1)定义传感器设备类

对于不同消息的格式,需要设计一个标准的格式,使得无论程序接收的是哪种消息格式,都可以统一转换成这个标准的格式,从而实现不同消息格式的统一化,方便消息格式的互相转换。经过对现有传感器产品特性的归纳、总结定义了如下数据类和传感器设备类,描述如下:

```
/* 定义数据类,描述并存放传感器设备状态值 */
3 个引用
public class StateValue
{
    9 个引用
    public string Describe { set; get; } //传感器数值描述
    11 个引用
    public object Value { set; get; } //传感器数值
}
/* 定义传感器设备类,描述设备信息 */
29 个引用
public class SensorDevice
{
    9 个引用
    public string Name { set; get; } //设备名称
    10 个引用
    public byte DeviceID { set; get; } //设备号
    10 个引用
    public byte AreaAddr { set; get; } //设备所在区域地址
    10 个引用
    public byte AreaID { set; get; } //设备所在区域号
    22 个引用
    public StateValue[] stateValue { set; get; } //传感器数据类
    10 个引用
    public SensorDevice(int number)
    {
        stateValue = new StateValue[number];
        for (int i = 0; i < number; i++)
        {
            stateValue[i] = new StateValue();
        }
    }
}
```

上述传感器设备类描述了传感器设备的类型、所在的区域地址、区域号、设备号,同时可以根据传感器的不同,无限扩充传感器采集的数据的描述与数值。

#### (2)“简单工厂模式”在传感器与协议中的应用

对于不同传感器或者协议,其实本质传输的内容并没有太大的改变,都是传感器相关的一系列属性。这种情况下,如果在编程上一次编写,然后根据不同的传感器类型进行相应不同的实例化,而使得系统具有更好的扩展性与较小的代码修改量,这是一个好的通信框架需要具备的。该模式能解决多种传感器与多种协议的扩展性问题。在定义了一个统一的标准传感器设备类后,可以根据不同的设备类型对其实例化。

#### (3)“观察者模式”设计

当前大多数物联网网关的通信都是轮询指令加同步通信的方式,但这种方式的通信会使数据传输有较大的延迟和阻塞,且随着设备数量的增加会更加明显。为了解决此问题,本网关的通信处理主要采用异步通信方式与“观察者模式”,这样使得通信在接收时不用一直处在等待阻塞状态,可以更实时的获取下位机传感器数据。在.NET下,使用委托可以很好的实现。C#委托的使用,如图2所示。

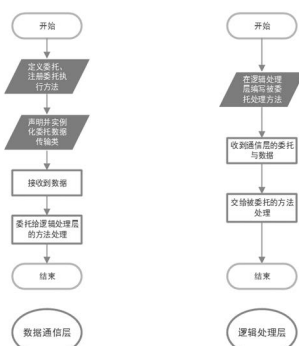


图2 C#委托使用流程图

Fig.2 C# commissioned flow chart

#### (4)数据传送中消息格式的确定

在数据交换中，JSON数据格式是一种轻量级别的“key/value”数据交换格式。与XML相比，JSON在描述同一设备的情况下，具有更少的容量，使得在传输中能够用较少的流量传输相同的数据。

### (5) 网关协议转换的实现流程

首先是ZIGBEE无线网络经过ZIGBEE协调器的串口与程序进行通信，程序的数据层在收到串口数据后，将数据传递给程序的逻辑层。逻辑层根据改写MODBUS协议的自定义协议解析成框架内统一的传感器设备对象实例，然后再经过JSON格式处理类将转化来的传感器设备实例打包成JSON格式字符串。再通过MQTT客户端将数据发布到MQTT服务器，从而推送到订阅者的客户端上，最终实现不同通信方式、不同协议、不同数据格式的转换与传输。

## 5 ZIGBEE与MQTT协议转换的实现(Implementation of ZIGBEE and MQTT protocol conversion)

在.NET开发中，M2Mqtt.Net是一个支持MQTT v3.1协议版本的MQTT客户端开源库。在通信框架核心网关程序中，使用M2Mqtt.Net开源库，并根据需求对源代码进行修改，来实现MQTT客户端上数据的发布与订阅。对于此类库的操作，如图3所示。

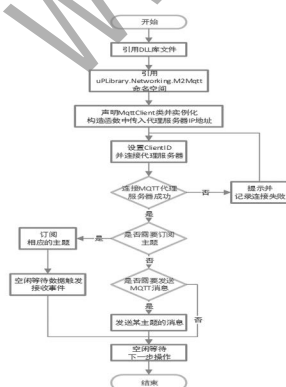


图3 PC端C#操作M2Mqtt开源库流程图

Fig.3 C# operating M2Mqtt open source library  
flow chartonPC

通过对MqttClient实例化并操作，可以实现客户端与MQTT代理服务器的连接、发布、订阅主题消息。发布消息是推送感知层数据，订阅消息则是接收应用层客户端的控制指令。

C#中有相应的JSON打包与解析开源库，如Newtonsoft.Json开源库。使用Newtonsoft.Json可以对一个设备对象类直接打包成JSON格式的字符串，也可以将JSON格式的字符串中解析为一个设备对象类，简化了开发过程、提高了开发效率。

这样就基本实现了将感知层数据收集后转换到JSON格式的MQTT协议中,进行推送,如图4所示。



图4 PC端MQTT协议的数据打包与推送

Fig.4 MQTT protocol data packed and pushed on PC

## 6 结论 (Conclusion)

物联网网关的通信框架中协议的数据传输与不同的协议转换是其关键所在。基于ZigBee/MQTT的物联网网关协议框架能很好的解决了多协议、多消息格式的兼容与扩展的问题。通信方式不同，根据不同的通信类获取数据后委托给逻辑处理层来处理，逻辑处理层把委托来的消息，通过不同的消息格式处理类来解析或打包消息，最终统一转化为框架内标准的传感器设备对象，用于后续的传输或界面显示。

### 参考文献(References)

- [1] 张艺粟,等.物联网ZigBee网关的设计与实现[J].计算机系统应用,2013,22(6):34-38.
- [2] 牛作领.物联网ZigBee网关的设计与实现方法[J].电子技术与软件工程,2016(24):19-19.
- [3] 岑荣滢,等.面向智能家居应用的ZigBee-WiFi网关[J].计算机系统应用,2017,3:232-235.
- [4] 程春潘,王可.基于Zigbee与TCP的物联网网关设计[J].电子信息与计算机科学,2014,9:1.
- [5] 杜岗,等.一种ZigBee-GPRS网关系统设计[J].自动化应用,2015,3:60-62.
- [6] 林湃,张家铭,杨海波.基于MQTT协议的即时消息业务设计与实现[J].计算机系统应用,2017,3:219-224.

### 作者简介:

谭方勇(1976-), 男, 硕士, 副教授. 研究领域: 物联网应用开发.

王 昂(1993-),男,大专,工程师.研究领域:物联网网关技术传感器.

刘子宁(1995-), 男, 大专生. 研究领域: 移动应用开发.