Message Queuing Telemetry Transport（MQTT）

|  |  |  |  |
| --- | --- | --- | --- |
| **版本号** | **修订日期** | **修订内容** | **说明** |
| V1.1 | 2015/9/8 | 草稿 |  |
| V2.0 | 2016/4/8 | 重构 | Liuyuan |
| V2.1 | 2016/4/19 | 丰富报文格式，业务流程 | Leihong |
| V2.2 | 2016/7/13 | 增加设备间订阅，创建topic功能 | Leihong |
| V2.3 | 2016/11/2 | 加入disconnect | Leihong |

目录

[1 说明 3](#_Toc456192642)

[2 接入流程 3](#_Toc456192643)

[3 Packet格式说明 4](#_Toc456192644)

[3.1 Fixed header 4](#_Toc456192645)

[3.2 Variable Header &Payload 4](#_Toc456192646)

[4 支持的packet 5](#_Toc456192647)

[4.1 CONNECT 5](#_Toc456192648)

[4.1.1 Fixed Header 5](#_Toc456192649)

[4.1.2 VariableHeader 5](#_Toc456192650)

[4.1.3 Payload 6](#_Toc456192651)

[4.2 CONNACK 6](#_Toc456192652)

[4.2.1 Fixed Header 6](#_Toc456192653)

[4.2.2 VariableHeader 6](#_Toc456192654)

[4.3 PUBLISH（client -> server） 7](#_Toc456192655)

[4.3.1 Fixed header 7](#_Toc456192656)

[4.3.2 VariableHeader 7](#_Toc456192657)

[4.3.3 Payload 8](#_Toc456192658)

[4.4 PUBLISH（server -> client） 8](#_Toc456192659)

[4.4.1 Fixed header 8](#_Toc456192660)

[4.4.2 VariableHeader 8](#_Toc456192661)

[4.4.3 Payload 8](#_Toc456192662)

[4.5 PUBACK 9](#_Toc456192663)

[4.4.1 Fixed header 9](#_Toc456192664)

[4.4.2 VariableHeader 9](#_Toc456192665)

[4.6 SUBSCRIBE 9](#_Toc456192666)

[4.5.1 Fixed header 9](#_Toc456192667)

[4.5.2 VariableHeader 9](#_Toc456192668)

[4.5.3 Payload 9](#_Toc456192669)

[4.7 SUBACK 10](#_Toc456192670)

[4.6.1 Fixed header 10](#_Toc456192671)

[4.6.2 VariableHeader 10](#_Toc456192672)

[4.6.3 Payload 10](#_Toc456192673)

[4.8 UNSUBSCRIBE 10](#_Toc456192674)

[4.7.1 Fixed header 10](#_Toc456192675)

[4.7.2 VariableHeader 11](#_Toc456192676)

[4.7.3 Payload 11](#_Toc456192677)

[4.9 UNSUBACK 11](#_Toc456192678)

[4.9.1 Fixed header 11](#_Toc456192679)

[4.9.2 VariableHeader 11](#_Toc456192680)

[5 接入流程 12](#_Toc456192681)

[5.1 连接鉴权 12](#_Toc456192682)

[5.2 消息发布 13](#_Toc456192683)

[5.2.1 数据点上报 13](#_Toc456192684)

[5.2.2 平台命令(下发&回复) 17](#_Toc456192685)

[5.2 创建Topic 19](#_Toc456192686)

[5.3 订阅 21](#_Toc456192687)

[5.4 取消订阅 21](#_Toc456192688)

[5.5 推送设备Topic 22](#_Toc456192689)

# 说明

MQTT协议详细内容请参见MQTT version 3.1.1官方文档，本文档对此不做详细说明，仅指明OneNet的要求、默认参数、以及当前实现与MQTT官方文档的差异。

该版本支持的功能:

* 鉴权;
* 数据点上报(平台指定topic);
* 创建topic;
* 获取项目的topic列表;
* 订阅/取消平台的topic;
* 设备间topic订阅;
* 平台命令下发;
* Qos0(c⬄S) ,Qos1(C->S);
* 连接保活

# 接入流程

* 1. 访问平台<http://open.iot.10086.cn/>注册用户；
  2. 用户根据业务情况，在“连接请求”章节中选择EDP登录方式（目前公测阶段，页面还未提供MQTT登录选项，登录方式与EDP兼容）；
  3. 登录需填写设备相关属性，在项目下新增设备，获取项目ID、设备ID，以及authinfo等信息；
  4. 设备发送TCP连接请求到以下地址，发送封装的报文与平台交互。

**平台服务器地址183.230.40.39，TCP 端口6002**

# Packet格式说明

包格式包含三部分：

|  |  |
| --- | --- |
| Fixed Header | 所有packet中都必须有 |
| Varable Header | 部分包含有 |
| Payload | 部分包含有 |

## 3.1 Fixed header

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bit** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| byte 1 | MQTT Packet Type | | | | 0 | 0 | 0 | 0 |
| byte2 - 5 | Remaining Length（该字段占用1-4个字节） | | | | | | | |

该版本支持的所有类型：

|  |  |  |  |
| --- | --- | --- | --- |
| 名字 | 值 | 流向 | 描述 |
| CONNECT | 1 | C->S | 客户端请求与服务端建立连接 |
| CONNACK | 2 | S->C | 服务端确认连接建立 |
| PUBLISH | 3 | C⬄S | 发布消息 |
| PUBACK | 4 | C⬄S | 收到发布消息确认 |
| SUBSCRIBE | 8 | C->S | 订阅请求 |
| SUBACK | 9 | S->C | 订阅确认 |
| UNSUBSCRIBE | 10 | C->S | 取消订阅 |
| UNSUBACK | 11 | S->C | 取消订阅确认 |
| PING | 12 | C->S | 客户端发送PING(连接保活)命令 |
| PINGRSP | 13 | S->C | PING命令回复 |
| DISCONNECT | 14 | C->S | 断开连接 |

## 3.2 Variable Header &Payload

|  |  |  |
| --- | --- | --- |
| 消息类型 | Variable Header | Payload |
| CONNECT | 有 | 有 |
| CONNACK | 有 | 有 |
| PUBLISH | 有 | 有 |
| PUBACK | 有 | 无 |
| SUBSCRIBE | 有 | 有 |
| SUBACK | 有 | 有 |
| UNSUBSCRIBE | 有 | 有 |
| UNSUBACK | 有 | 无 |
| PING | 无 | 无 |
| PINGRSP | 无 | 无 |
| DISCONNECT | 无 | 无 |

# 支持的packet

## CONNECT

### 4.1.1 Fixed Header

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bit** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| byte 1 | MQTT Packet Type | | | | 0 | 0 | 0 | 0 |
| byte2 - 5 | Remaining Length（该字段占用1-4个字节） | | | | | | | |

### 4.1.2 VariableHeader

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Description** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| byte 1-2 | ProtocolName Length | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| byte 3 | ‘M’ | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| byte 4 | ‘Q’ | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| byte 5 | ‘T’ | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| byte 6 | ‘T’ | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| Byte7 | Protocol Level | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| Byte8 | Connect Flag | User  flag | Password  flag | WillRetain  Flag | WillQos Flag | | WillFlag | CleanSession  Flag | Reserve |
| Byte9-10 | KeepAlive |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

1. **版本**

必须设置为4，平台只支持版本v 3.1.1，不支持更老的版本。使用第三方客户端时需要注意选择正确的版本。

1. **user flag与password flag**

平台不允许匿名登陆，因此这两个标志位在连接时必须设置为1，否则认为协议错误，平台将会断开连接。

1. **will flag与Willretainflag/Willqosflag**

平台暂不支持**will flag**，WillRetain Flag与 WillQos Flag必须设置为0。

1. **CleanSessionFlag**

若客户端将clean session标志位设置为0，当其断开后，平台将会保存session，session需保持的内容包含：

* 客户端订阅的topic列表.

客户端保存session的内容包含：

* 已经发送到服务端的但还没有收到确认的Qos1消息列表.
* 待发送的Qos0列表.

1. **Reserve**

保留位，置0。

1. **KeepAlive保活时间**

每个客户端可自定义设置连接保持时间，最短120秒，最长65535秒。

### 4.1.3 Payload

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Description** | **是否必须存在** | **格式** |
| Field1 | Client Identifier | 是 | 2字节字串长度 + utf8字串 |
| Field2 | UserName | 是 | 2字节字串长度 + utf8字串 |
| Field3 | UserPassword | 是 | 2字节字串长度 + utf8字串 |

与鉴权相关的字段包含client id，username和password，支持鉴权方式。

|  |  |
| --- | --- |
| **字段设置** | **消息示例** |
| client\_id设置为平台创建设备时的设备id  username设置为“项目ID”  password设置为“鉴权信息（auth\_info）” | client\_id=”123”  username=”433223”  password=”注册的鉴权信息” |

各字段说明如下：

项目ID：在平台添加项目时平台生成的ID；

鉴权信息（auth\_info）：在平台申请设备时填写设备的auth\_info属性（数字+字母的字符串），该属性需要产品内具备唯一性；

## CONNACK

### 4.2.1 Fixed Header

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bit** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| byte 1 | MQTT Packet Type | | | | 0 | 0 | 0 | 0 |
| byte2 - 5 | Remaining Length（该字段占用1-4个字节） | | | | | | | |

### 4.2.2 VariableHeader

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Description** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| byte 1 | Acknowledge Flags | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Sp |
| byte 2 | Return Code | x | x | x | x | x | x | x | x |

Sp: Session Present Flag，session信息在服务器已保持，置1；未保存，置0。

**返回码说明：**

|  |  |
| --- | --- |
| 返回码 | 描述 |
| 0 | 成功 |
| 1 | 协议版本错误 |
| 2 | 非法的clientid |
| 3 | 服务不可用 |
| 4 | 用户名或密码错误 |
| 5 | 非法链接(比如token非法) |

**失败：**

\*如果connect包不符合协议内容约束，则直接断掉连接，而不需要发送connack包.

\*如果鉴权或授权失败，回复一个带非0错误码的connack包.

**成功：**

\*必须断掉重复的clientid.

\*执行cleansession 对应的操作.

\*必须回复一个connack，回复码为0.

\*开始消息传递，并加入keepalive的监视.

**PS：**客户端需要等到服务端的connack报文，才能发送后续的数据包.

## PUBLISH（client -> server）

### 4.3.1 Fixed header

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bit** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| byte 1 | MQTT Packet Type | | | | DUP flag | QoS Level | | RETAIN |
| byte2 - 5 | Remaining Length（该字段占用1-4个字节） | | | | | | | |

DUP:

QoS1：如果为0，则表示是第一次发送该包，如果为1，则表示为重复发送的包。

Qos0：DUP必须为0

QOS: 指定了该publish包的qos等级如下

RETAIN: 暂不实现

|  |  |  |  |
| --- | --- | --- | --- |
| Qos 值 | Bit2 | Bit1 | 描述 |
| 0 | 0 | 0 | 最多发送一次 |
| 1 | 0 | 1 | 至少发送一次 |

PS:该版本只实现Qos0，Qos1

### 4.3.2 VariableHeader

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Description** | **格式** | **是否必须** |
| Field1 | TopicName | 2字节字串长度 + utf8字串 | 是 |
| Field2 | PacketIdentifier | 2字节 | QoS0:否， QoS1:是 |

### 4.3.3 Payload

内容根据不同业务自定义.

## PUBLISH（server -> client）

### 4.4.1 Fixed header

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bit** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| byte 1 | MQTT Packet Type | | | | DUP flag | QoS Level | | RETAIN |
| byte2 - 5 | Remaining Length（该字段占用1-4个字节） | | | | | | | |

DUP:

QoS1：如果为0，则表示是第一次发送该包，如果为1，则表示为重复发送的包。

Qos0：DUP必须为0

QOS: 指定了该publish包的qos等级如下

RETAIN:

|  |  |  |  |
| --- | --- | --- | --- |
| Qos 值 | Bit2 | Bit1 | 描述 |
| 0 | 0 | 0 | 最多发送一次 |

PS:该版本只实现Qos0

### 4.4.2 VariableHeader

|  |  |  |
| --- | --- | --- |
|  | **Description** | **格式** |
| Field1 | TopicName | 2字节字串长度 + utf8字串 |

### 4.4.3 Payload

内容根据不同业务自定义.

## PUBACK

### 4.5.1 Fixed header

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bit** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| byte 1 | MQTT Packet Type | | | | 0 | 0 | 0 | 0 |
| byte2 ~ 5 | Remaining Length（该字段占用1-4个字节） | | | | | | | |

### 4.5.2 VariableHeader

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Description** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| byte 1~2 | PacketIdentifier | PacketIdentifier | | | | | | | |

## SUBSCRIBE

### 4.6.1 Fixed header

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bit** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| byte 1 | MQTT Packet Type | | | | 0 | 0 | 0 | 0 |
| byte2 ~5 | Remaining Length（该字段占用1-4个字节） | | | | | | | |

### 4.6.2 VariableHeader

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Description** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| byte 1~2 | PacketIdentifier | PacketIdentifier | | | | | | | |

### 4.6.3 Payload

|  |  |  |
| --- | --- | --- |
|  | **Description** | **格式** |
| Byte1~n | TopicName | 2字节字串长度 + utf8字串 |
| Byten+1 | Reserved | 保留字节(兼容mqtt-v3.1.1) |

**topic说明**

可以包含一个或多个topic.

topic必须是数字、英文、反斜杠（/）的组合，目前不支持通配符。

每个客户端最多订阅100个topic；

以下topic被系统保留使用，（客户端不能订阅）：

|  |  |
| --- | --- |
| **类型** | **说明** |
| $开头 | 平台保留 |

## SUBACK

### 4.7.1 Fixed header

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bit** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| byte 1 | MQTT Packet Type | | | | 0 | 0 | 0 | 0 |
| byte2 ~ 5 | Remaining Length（该字段占用1-4个字节） | | | | | | | |

### 4.7.2 VariableHeader

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Description** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| byte 1~2 | PacketIdentifier | PacketIdentifier | | | | | | | |

### 4.7.3 Payload

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Description** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| byte 1 | retcode |  | | | | | | | |

**返回码说明：**

|  |  |
| --- | --- |
| 返回码 | 描述 |
| 0x00 | 成功 |
| 0x80 | 失败 |

## UNSUBSCRIBE

### 4.8.1 Fixed header

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bit** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| byte 1 | MQTT Packet Type | | | | 0 | 0 | 0 | 0 |
| byte2 ~ 5 | Remaining Length（该字段占用1-4个字节） | | | | | | | |

### 4.8.2 VariableHeader

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Description** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| byte 1~2 | PacketIdentifier | PacketIdentifier | | | | | | | |

### 4.8.3 Payload

|  |  |  |
| --- | --- | --- |
|  | **Description** | **格式** |
| Byte1~n | TopicName | 2字节字串长度 + utf8字串 |

可以包含一个或多个topic.

## 4.9 UNSUBACK

### 4.9.1 Fixed header

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bit** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| byte 1 | MQTT Packet Type | | | | 0 | 0 | 0 | 0 |
| byte2 ~ 5 | Remaining Length（该字段占用1-4个字节） | | | | | | | |

### 4.9.2 VariableHeader

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Description** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| byte 1~2 | PacketIdentifier | PacketIdentifier | | | | | | | |

## PING

### Fixed header

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bit** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| byte 1 | MQTT Packet Type | | | | 0 | 0 | 0 | 0 |
| byte2 ~ 5 | Remaining Length（该字段占用1-4个字节） | | | | | | | |

### 4.10.2 VariableHeader

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Description** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| byte 1~2 | NULL | NULL | | | | | | | |

注：PING命令无须VariableHeader

## PINGRSP

### Fixed header

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bit** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| byte 1 | MQTT Packet Type | | | | 0 | 0 | 0 | 0 |
| byte2 ~ 5 | Remaining Length（该字段占用1-4个字节） | | | | | | | |

### 4.11.2 VariableHeader

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Description** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| byte 1~2 | NULL | NULL | | | | | | | |

注：PINGRSP无VariableHeader

## DISCONNECT

### Fixed header

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bit** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| byte 1 | MQTT Packet Type | | | | 0 | 0 | 0 | 0 |
| byte2 ~ 5 | Remaining Length（该字段占用1-4个字节） | | | | | | | |

### 4.12.2 VariableHeader

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Description** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| byte 1~2 | NULL | NULL | | | | | | | |

注：DISCONNECT无须VariableHeader

# 接入流程

## 连接鉴权



* 设备向平台发起connect请求.connect中携带鉴权信息,具体参见(报文格式参考4.1)
* 平台拿到鉴权信息进行鉴权.
* 鉴权通过后，如果cleansession=0, 平台将会加载保存的设备的一些信息.如订阅列表(4.1.2中描述).

如果cleansession=1, 设备没有保存信息在平台，则不加载设备相关信息.

* 返回鉴权结果ConnAck(报文格式参考4.2).

## 消息发布

### 5.2.1 数据点上报

设备使用publish报文来上传数据点， 报文格式如下：

VariableHeader：

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Field名称** | **说明** | **格式** |
| Field1 | TopicName=”**$dp**” | $dp为系统上传数据点的指令 | 2字节字串长度 + utf8字串 |

Payload:

Payload包含真正的数据点内容,支持的格式如下：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **字节** | **说明\bit** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Byte 1 | Bit0-5数据类型指示，目前支持：  Type = 1 //可离散浮点数数据流  Type = 7  Bit6-7 flags 根据具体类型不同意义 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| Byte 2 | 根据数据类型指示不同 |  |  |  |  |  |  |  |  |
| 。。。 |  |  |  |  |  |  |  |  |
| Byte n |  |  |  |  |  |  |  |  |

**数据类型1格式说明：**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte 1 | 数据点类型值：1 //1： json格式1字符串 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| Byte 2 | //指示后面json字符串长度  固定两字节长度高位字节，值为0x00 |  |  |  |  |  |  |  |  |
| Byte 3 | 固定两字节长度低位字节，值为0x41 |  |  |  |  |  |  |  |  |
| Byte 4 | {  “datastreams”:[// 可以同时传递多个数据流  {  “id”:”temperature”,  “datapoints”:[  {  ”at”:”2013-04-22 22:22:22”,//可选  ”value”: 36.5//用户自定义  }  ]  },  {  “id”:”location”  “datapoints”:[…]  }, { … }  ]  } |  |  |  |  |  |  |  |  |
| …  …  …  …  …  …  …  …  …  … |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| Byte n |  |  |  |  |  |  |  |  |

**数据类型7(type == 7)格式说明**：（每次最多500个数据流的浮点数）

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte 1 | Bit0-5数据类型指示：type=7 //可离散浮点数数据流  Bit6：保留，置0  Bit7: 时间指示位，1，携带6字节时间 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| Byte 2 | 年（后两位），例如2016年，则该字节为16 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| Byte 3 | 月（1-12） |  |  |  |  |  |  |  |  |
| Byte 4 | 日（1-31） |  |  |  |  |  |  |  |  |
| Byte 5 | 小时（0~23） |  |  |  |  |  |  |  |  |
| Byte 6 | 分钟（0~59） |  |  |  |  |  |  |  |  |
| Byte7 | 秒（0~59） |  |  |  |  |  |  |  |  |
| Byte 8 | //数据流名称（取值范围1-65535）  高位字节，值为0x00 |  |  |  |  |  |  |  |  |
| Byte 9 | 低位字节，值为0x01 |  |  |  |  |  |  |  |  |
| Byte10 | //数据流个数（取值范围1-500）  高位字节，值为0x00 |  |  |  |  |  |  |  |  |
| Byte11 | 低位字节，值为0x01 |  |  |  |  |  |  |  |  |
| Byte 10 | 4字节float类型，低位在前，高位在后 |  |  |  |  |  |  |  |  |
| Byte 11 |  |  |  |  |  |  |  |  |
| Byte 12 |  |  |  |  |  |  |  |  |
| Byte 13 |  |  |  |  |  |  |  |  |
| ... |  |  |  |  |  |  |  |  |  |
| Byte n | //数据流名称（取值范围1-65535）  高位字节，值为0x24 |  |  |  |  |  |  |  |  |
| Byte n+1 | 低位字节，值为0x37 |  |  |  |  |  |  |  |  |
| Byte n+2 | //数据流个数（取值范围1-500）  高位字节，值为0x01 |  |  |  |  |  |  |  |  |
| Byte n+3 | 低位字节，值为0x00 |  |  |  |  |  |  |  |  |
| Byte n+2 | 4字节float类型，低位在前，高位在后 |  |  |  |  |  |  |  |  |
| Byte n+4 |  |  |  |  |  |  |  |  |
| Byte n+5 |  |  |  |  |  |  |  |  |
| Byte n+6 |  |  |  |  |  |  |  |  |

#### 5.2.1.1 Qos0(Client->Server)



* 设备发布Qos0消息(上报数据点)
* 平台收到上报数据点后保存起来.

#### 5.2.1.2 Qos1(Client->Server)



* 设备发布Qos0消息(上报数据点)
* 平台收到上报数据点后保存起来.
* 平台给设备回复PubAck(报文格式参考4.5)

### 5.2.2 平台命令(下发&回复)

#### 5.2.2.1命令下发

平台使用publish报文来下发平台指令， 报文格式如下：

FixHeader：

参考4.3.1

VariableHeader：

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Field名称** | **说明** | **格式** |
| Field1 | TopicName=”**$creq/cmduuid**” | $creq为系统下发Cmd的指令，cmduuid为该条指令的uuid | 2字节字串长度 + utf8字串 |

Payload:

Payload包含真正的指令内容

##### 5.2.2.1.1 Qos0(Server->Client)



* 命令下发：

平台向设备发送topic为$creq的消息(该topic为平台命令).

设备收到topic为$creq的topic时，需将其作为平台下发的指令来处理.

#### 5.2.2.1命令回复

设备使用publish报文来回复平台指令， 报文格式如下：

FixHeader：

参考4.3.1

VariableHeader：

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Field名称** | **说明** | **格式** |
| Field1 | TopicName=”**$crsp/cmduuid**” | $crsp为系统处理设备回复cmd的指令，cmduuid为该条指令的uuid | 2字节字串长度 + utf8字串 |

Payload:

Payload包含真正回复的指令内容

##### 5.2.2.2.1 Qos0(Client->Server)



##### 5.2.2.2.1 Qos1(Client<-> Server)



如果设备回复响应时以Qos1回复，则平台需要给设备回复一个Puback消息

## 创建Topic



* 设备通过发送HTTP请求进行topic的创建操作.
* 平台收到请求后创建topic并返回结果.

请求及响应定义如下：

|  |  |
| --- | --- |
| HTTP方法 | POST |
| URL | http://api.heclouds.com/mqtt/topic |
| HTTP头部 | api-key:xxxx-ffff-zzzzz，必须master key |
| URL参数 |  |
| HTTP body内容 | {  “name”:”xxxxxx” //topic名字，只能由英文，下划线，反斜杠组成，最长1024个字符，同一产品下topic name不重复。  } |
| 成功返回 | {  "errno": 0,  "error":“succ”  } |
| 说明 | 每个产品最多100个topic |

## 订阅



* 设备发起订阅请求.(报文格式参考4.6)
* 平台收到请求后更新topic列表.
* 平台给设备回复SubAck. (报文格式参考4.7)

## 取消订阅



* 设备发起取消订阅请求. (报文格式参考4.8)
* 平台收到请求后更新topic列表.
* 平台给设备回复UnSubAck. (报文格式参考4.9)

## 推送设备Topic

### 5.6.1 Publish报文推送:



* 设备发起推送topic请求.(报文格式参考5.5.1)
* 平台收到请求后,将topic推送到相关订阅设备.(目前只支持在线推送)
* 平台返回推送ack(取决于qos).(报文格式参考4.5)

设备使用publish报文来推送Topic. 报文格式如下：

FixHeader：

参考4.3.1

VariableHeader：

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Field名称** | **说明** | **格式** |
| Field1 | TopicName | 填写设备订阅的topic | 2字节字串长度 + utf8字串 |

Payload：

Payload为设备自定义内容

### 5.6.2 HTTP请求推送



* 设备以HTTP的方式发起推送topic请求.
* 平台收到请求后,将topic推送到相关订阅设备.(目前只支持在线推送)
* 平台返回推送结果.

请求及响应定义如下：

|  |  |
| --- | --- |
| HTTP方法 | POST |
| URL | http://api.heclouds.com/mqtt |
| HTTP头部 | api-key:xxxx-ffff-zzzzz，必须master key |
| URL参数 | topic=“XXXXXX”，必选，在发送mqtt协议的topic. 必填 |
| HTTP body内容 | 用户自定义Json或二进制数据（小于64K） |
| 成功返回 | {  "errno": 0,  "error":“succ”  } |

PS: 欢迎访问设备云门户网站<http://open.iot.10086.cn/>注册用户，获取最新文档。