

# 浙江大学



课程 \_\_\_\_\_ 计算机网络与现场总线 \_\_\_\_\_

题目 \_\_\_\_\_ 实验 1: MQTT 数据传输 \_\_\_\_\_

实验指导 \_\_\_\_\_ 杨春节 张建明 朱雄卓 \_\_\_\_\_

组员 \_\_\_\_\_ 曾豪辉 3180103783 \_\_\_\_\_

\_\_\_\_\_ 钱 煜 3180103948 \_\_\_\_\_

## 一、实验目的

实现物联网平台开发——终端与云平台的数据传输功能：基于 IoT Hub 组件完成 MQTT 数据通信。

## 二、实验工具

### 2.1 实验平台

#### 2.1.2 Aliyun Link Platform

#### 2.1.1 Develop Kit

MCU: STM32L496VGTX, Cortex-M4, 80MHz

Memory: 1 MB Flash, 320KB SRAM

Micro-USB: USB 2.0, USB1 for ST-Link USB2 for OTG

Module: Sensors, Camera, Key, LED, LCD, Wi-Fi, Audio

SD card: Support 32GB

OS: AliOS Things

Application: Remote control, measure and data restore

### 2.2 开发环境

Visual Studio Code (Windows, MacOS, Linux);

AliOS Things Studio;

IntelliJ IDEA;

## 三、实验原理

### 3.1 IoT Hub 及其特点

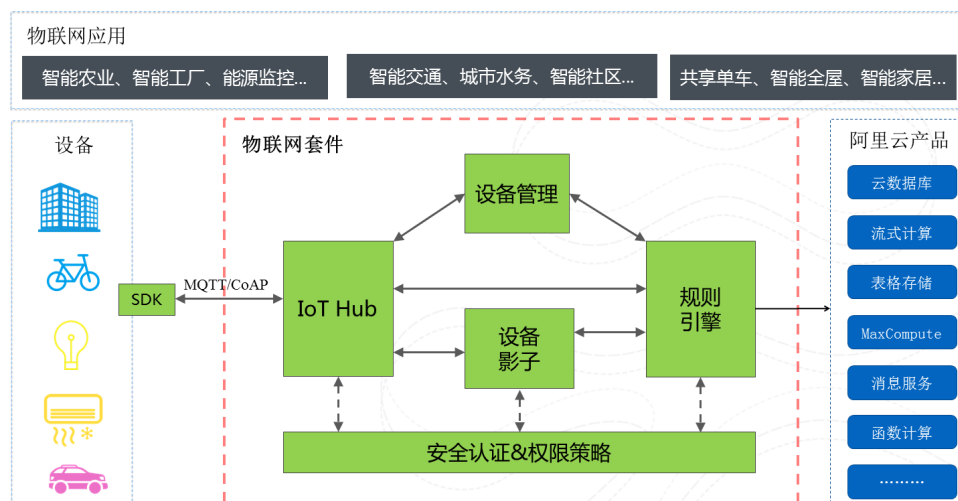


图 1 IoT Hub 与数据传输

如图 1，IoT Hub 通过 MQTT/CoAP 等协议与设备相连接，使其接入云端，作用类似设备网关。

其主要特点有：支持多种设备接入协议、支持多种通信模式、高性能扩展、全链路加密、消息实时到达、支持数据透传等。

### 3.2 MQTT

MQTT（Message Queuing Telemetry Transport Protocol）的全称是消息队列遥感传输协议的缩写，是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，为其提供有序、可靠、双向连接的网络连接保证。

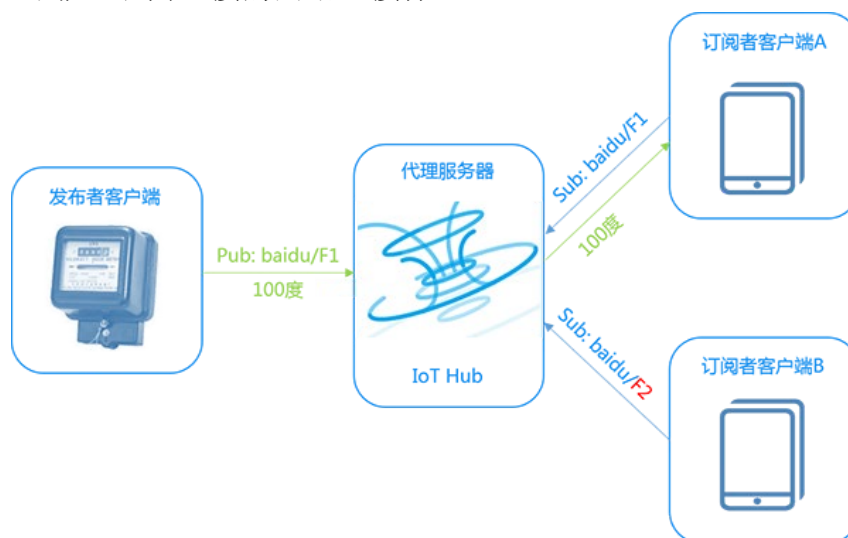


图 2 MQTT 与 Topic

如图 2，MQTT 协议中有三种角色：代理服务器、发布者客户端和订阅者客户端。

MQTT 通过“主题” (topic) 实现将消息从发布者客户端送达至接收者客户端。“主题”是附加在应用消息上的一个标签，发布者客户端将“主题”和“消息”发送至代理服务器，代理服务器将该消息转发至每一个订阅了该“主题”的订阅者客户端。

物联网平台与设备之间基于 Topic 进行消息的路由转发，每类 Topic 都具有自身的设备操作权限：发布或订阅，设备端通过向物联网平台发布或订阅消息实现通信。

为了方便海量设备基于海量 Topic 进行通信，简化授权操作，物联网平台增加了 Topic 类的概念。创建产品后，物联网平台会该产品自动创建默认的 Topic 类。并且，在创建设备后，会自动将产品 Topic 类映射到设备上。无需单独为每个设备授权 Topic。

**Topic 类格式：**

`/${productKey}/${deviceName}/update`

产品的 Topic 类不用于通信，只是定义 Topic。用于消息通信的是具体设备的 Topic。

**Topic 格式：**

`/${productKey}/device1/update`

## 四、实验过程

### 4.1 设备创建

#### 4.1.1 注册账号



图 3 阿里云物联网控制台界面

进入阿里云网站：<https://www.aliyun.com>，注册账号后进入开通控制台，界面如图 5，点击产品与服务中的物联网平台。

#### 4.1.2 创建产品和设备

单击创建产品，填写产品名称 (product1) 和产品描述 (图 4)。记录下对应的 ProductKey。

点击管理产品，并添加设备，如图 5 所示。点击查看设备可看到详细信息，记录对应的 ProductKey、DeviceName、DeviceSecret，如图 6 所示。

物联网平台 / 设备管理 / 产品

## 产品 (1)

<a href="#">创建产品</a>	<a href="#">快速入门</a>	<input type="text" value="请输入产品名称查询"/>	<input type="button" value="Q"/>	<input type="text" value="请选择产品标签"/>	<input type="button" value="v"/>
产品名称	ProductKey	节点类型	创建时间	操作	
product1	a1t1W6MTaUb	设备	2020/10/18 16:26:14	<a href="#">查看</a>	<a href="#">管理设备</a>   <a href="#">删除</a>

图 4 新建产品列表

添加设备 ?

×

特别说明: DeviceName 可以为空, 当为空时, 阿里云会颁发全局唯一标识符作为 DeviceName。

产品

product1

DeviceName ?

DeviceTest1

备注名称 ?

请输入备注名称

确认

取消

图 5 添加设备

物联网平台 / 设备管理 / 设备 / 设备详情

## ← DeviceTest1 离线

产品 product1 [查看](#)  
ProductKey a1t1W6MTaUb [复制](#)

DeviceSecret \*\*\*\*\* [查看](#)

<a href="#">设备信息</a>	<a href="#">Topic 列表</a>	<a href="#">物模型数据</a>	<a href="#">设备影子</a>	<a href="#">文件管理</a>	<a href="#">日志服务</a>	<a href="#">在线调试</a>	<a href="#">分组</a>
设备信息							
产品名称	product1	ProductKey	a1t1W6MTaUb <a href="#">复制</a>				
节点类型	设备	DeviceName	DeviceTest1 <a href="#">复制</a>				

图 6 设备详细信息

回到产品, 点击已创建的 product1, 选择 Topic 类列表的自定义 Topic, 定义 Topic 类如图 7。确定后可以有看到有四个 Topic, 如图 8 所示。

其中, 发布是指设备具有向 IoT 套件发送信息的权限; 订阅是指设备不具有向 IoT 套件发布消息的权限, 但是可以订阅该 Topic 从而获取 IoT 套件下发的信息。

至此, 在阿里云 IoT 套件端的设备创建完成。

定义 Topic 类

Topic 格式必须以“/”进行分层，区分每个类目。其中前三个类目已经规定好，第一个代表产品标识 ProductKey，第二个 \${deviceName} 通配 DeviceName，第三个 user 用来标识产品的自定义 Topic 类。简单来说，Topic 类：/a15T\*\*\*dhK/\${deviceName}/user/update 是具体 Topic：/a15T\*\*\*dhK/mydevice1/user/update 和 /a15T\*\*\*dhK/mydevice2/user/update 等的集合。

\* 设备操作权限

发布和订阅

\* Topic 类

/a1t1W6MTaUb/\${deviceName}/user/data

data

描述

请输入描述

0/100

确认

取消

图 7 Topic 类定义

自定义 Topic	操作权限
/a1t1W6MTaUb/\${deviceName}/user/data	发布和订阅
/a1t1W6MTaUb/\${deviceName}/user/update	发布
/a1t1W6MTaUb/\${deviceName}/user/update/error	发布
/a1t1W6MTaUb/\${deviceName}/user/get	订阅

图 8 四个 Topic 信息

4.2 新建工程

打开 VS Code，打开已经下载的 AliOS Things。工程界面如图 9。  
单击工程界面左下角，选择例程为 MQTT 例程，开发板选择“DeveloperKit”开发板。

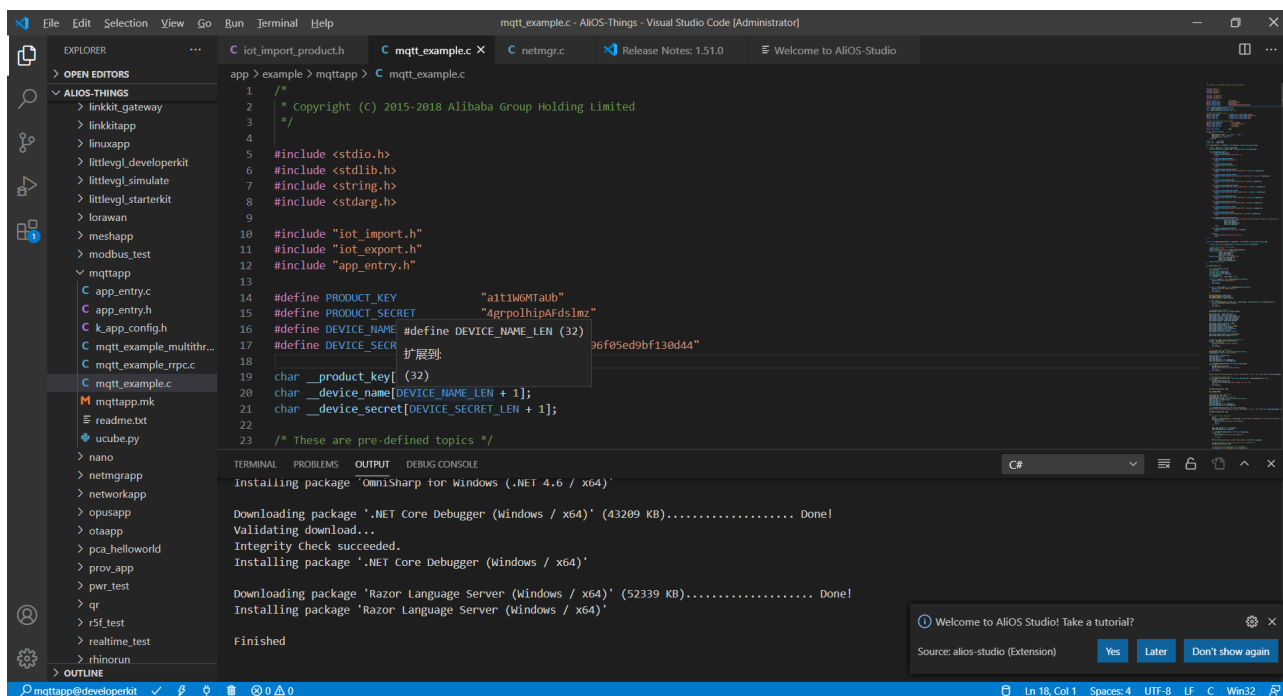


图 9 工程界面图

## 4.3 修改参数

### 4.3.1 修改设备相关参数

打开 mqtt-example.c 文件，将其中的设备信息修改为前述设备信息，具体修改如图 10。

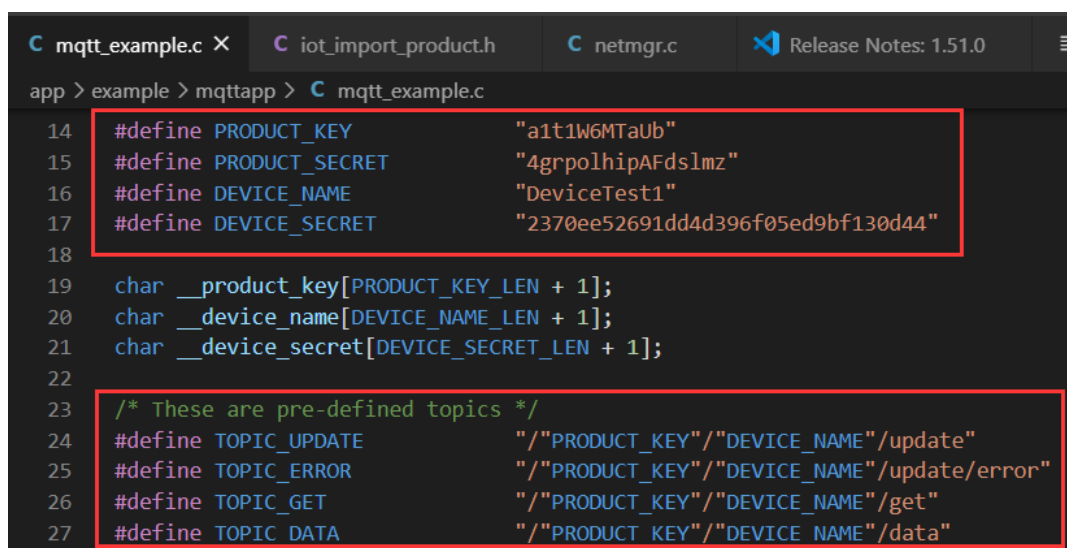
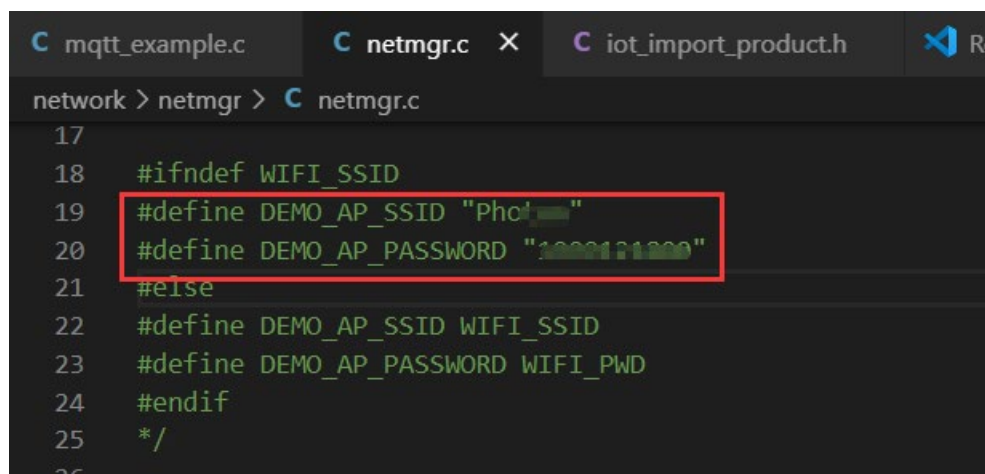


图 10 修改设备参数

### 4.3.2 修改 Wi-Fi 相关参数

Developer Kits 通过 Wi-Fi 连接网络，所以需要在代码中提供 Wi-Fi 的 SSID 和密码。打开 netmgr.c 文件，将其中的 DEMO\_AP\_SSID 和 DEMO\_AP\_PASSWORD 修改为可用的 Wi-Fi 名称和密码，如图 11。



```

17
18 #ifndef WIFI_SSID
19 #define DEMO_AP_SSID "Pho"
20 #define DEMO_AP_PASSWORD "1234567890"
21 #else
22 #define DEMO_AP_SSID WIFI_SSID
23 #define DEMO_AP_PASSWORD WIFI_PWD
24 #endif
25 */
26

```

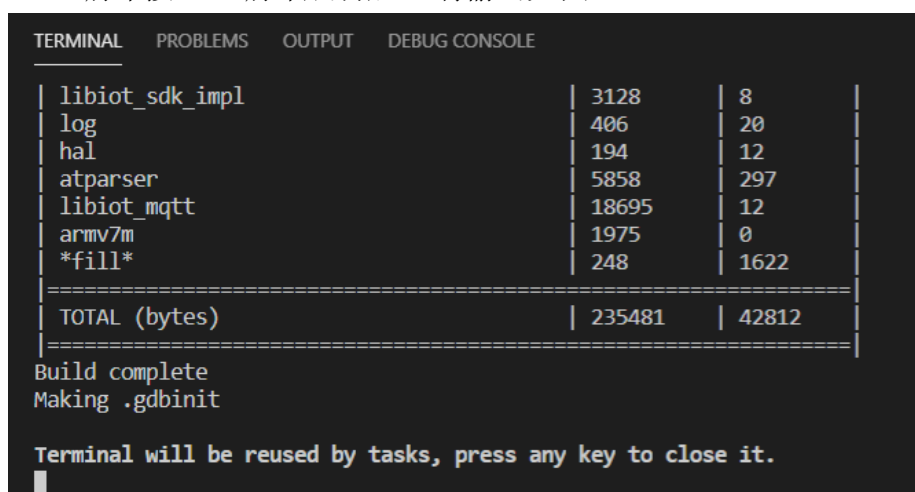
图 11 修改 Wi-Fi 参数

至此参数修改完成。

## 4.4 工程编译和下载

### 4.4.1 工程编译

点击 VS Code 编译按钮，编译成功后终端输出如图 12。



```

TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE

| libiot_sdk_impl          | 3128      | 8        |
| log                     | 406       | 20       |
| hal                     | 194       | 12       |
| atparser                | 5858      | 297      |
| libiot_mqtt              | 18695     | 12       |
| armv7m                  | 1975      | 0        |
| *fill*                   | 248       | 1622     |
|=====|
| TOTAL (bytes)            | 235481    | 42812    |
|=====|
Build complete
Making .gdbinit

Terminal will be reused by tasks, press any key to close it.

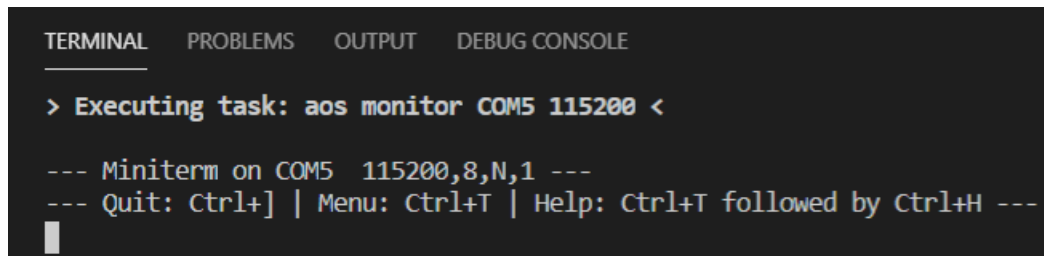
```

图 12 alios studio 的输出栏



## 4.4.2 设备连接和工程下载

连接开发板后单击 AliOS Studio: Connect Device 按钮，成功连接后终端输出如图 13。串口号为 COM5，波特率为 115200。



```

TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE

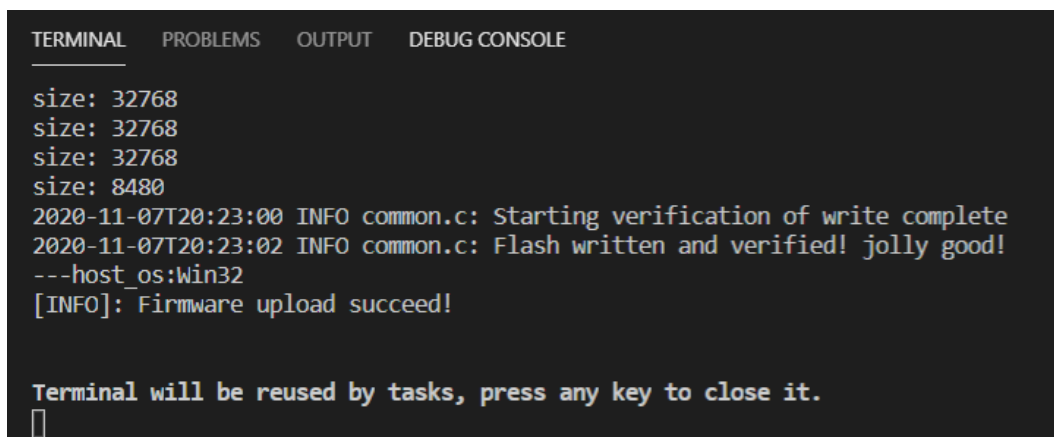
> Executing task: aos monitor COM5 115200 <

--- Miniterm on COM5 115200,8,N,1 ---
--- Quit: Ctrl+] | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H ---

```

图 13 设备连接

之后单击 AliOS Studio: Upload 按钮，烧录程序，完成后终端输出如图 14。



```

TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE

size: 32768
size: 32768
size: 32768
size: 8480
2020-11-07T20:23:00 INFO common.c: Starting verification of write complete
2020-11-07T20:23:02 INFO common.c: Flash written and verified! jolly good!
---host_os:Win32
[INFO]: Firmware upload succeed!

Terminal will be reused by tasks, press any key to close it.

```

图 14 烧录完成后界面

## 五、实验结果

启动串口并通过命令行输入命令：

```
1. netmgr Photon xxxxxxxxxxxx
```

设备成功连接到网络，并开始向云端发送数据。设备每个周期向云端发送数据，终端显示如图 15。在阿里云平台日志服务板块成功接收信息，如图 16。

TERMINAL   PROBLEMS   OUTPUT   DEBUG CONSOLE

```
mqtt_client|267 :: packet-id=155, publish topic msg={"attr_name":"temperature", "attr_value":"1"}
[dbg] iotx_mc_cycle(1663): PUBACK
event_handle|090 :: publish success, packet-id=154
[dbg] iotx_mc_cycle(1663): PUBACK
event_handle|090 :: publish success, packet-id=155
[dbg] MQTTPublish(303): WRITE COMPLETED: curr buf_send = 0x200126e0, curr buf_size_send = 1024
mqtt_client|267 :: packet-id=156, publish topic msg={"attr_name":"temperature", "attr_value":"1"}
[dbg] iotx_mc_cycle(1663): PUBACK
event_handle|090 :: publish success, packet-id=156
[dbg] MQTTPublish(303): WRITE COMPLETED: curr buf_send = 0x200126e0, curr buf_size_send = 1024
mqtt_client|267 :: packet-id=157, publish topic msg={"attr_name":"temperature", "attr_value":"1"}
█
```

图 15 设备发送信息

时间	TraceID	消息内容	DeviceName	业务类型(全部) ▾	操作 ⓘ	内容	状态 ⓘ
2020/11/07 22:31:32.872	0a30264a16047594928654828d5754	-	DeviceTest1	订阅	/a1t1W6MTaUb/DeviceTest1/user/data	["Content":"subscribe topic/a1t1W6MTaUb/Devi...	200
2020/11/07 22:31:32.691	0a30264a16047594926904696d5754	<a href="#">查看</a>	DeviceTest1	设备到云消息	/sys/a1t1W6MTaUb/DeviceTest1/thing/d...	["Content":"Publish message to...	200
2020/11/07 22:31:32.601	0a30264a16047594926004636d5754	<a href="#">查看</a>	DeviceTest1	设备到云消息	/sys/a1t1W6MTaUb/DeviceTest1/thing/d...	["Content":"Publish message to...	200
2020/11/07 22:31:32.616	0a30264a16047594926004636d5754	<a href="#">查看</a>	DeviceTest1	其它	/sys/a1t1W6MTaUb/DeviceTest1/thing/d...	-	200
2020/11/07 22:31:32.700	0a30264a16047594926904696d5754	<a href="#">查看</a>	DeviceTest1	其它	/sys/a1t1W6MTaUb/DeviceTest1/thing/d...	-	200
2020/11/07 22:31:32.278	0a30264a16047594922494390d5754	-	DeviceTest1	设备行为	online	["Content":"online, clientIp=112.17.238.85","cli...	200

图 16 云端接收信息