智能通用控制器16路

产品手册

产品型号：UNI-KZQ-TY-16

本产品支持私有化部署，支持自建消息服务器，可运行在纯局域网环境

接口采用HTTP协议，使得任何支持HTTP请求的编程语言均可调用。

接口简单、清晰、友好，仅需在接口携带签名、以及设备ID，即可向设备下发命令。

支持接入任何形式的软件项目：Web、APP/小程序、窗体软件、以及SaaS/低代码等平台。

设备使用WiFi 2.4G无线网络，采用 WiFi 直接方式，无需网关支持。

可设定5组 WiFi 网络，优先连接信号最强的进行连接。

本手册首次发布时间：2024-11-08 | 最近修改时间：2025-01-24

如发现本手册的任何错误，希望您能批评指正，非常感谢！

芯步

目录

1.1 产品特性

1.2 产品外观

1.3 产品规格与包装

2. 产品定义

了解产品定义（物模型），可以帮助您更加准确、灵活的调用接口，接入设备。

2 .1 状态属性

状态属性是产品的基础物模型，决定了产品的功能。通过下发命令、人为操作可改变这些状态属性；如果是传感器类或具备采样功能的产品，环境变化也会设备状态属性变化。

线路1（power1）

选项	值	说明
通	1	
断	0	

线路2（power2）

选项	值	说明
通	1	
断	0	

线路3（power3）

选项	值	说明
通	1	
断	0	

线路4（power4）

选项	值	说明
通	1	
断	0	

线路5（power5）

选项	值	说明
通	1	
断	0	

线路6（power6）

选项	值	说明
通	1	
断	0	

线路7（power7）

选项	值	说明
通	1	
断	0	

线路8（power8）

选项	值	说明
通	1	
断	0	

线路9（power9）

选项	值	说明
通	1	
断	0	

线路10（power10）

选项	值	说明
通	1	
断	0	

线路11（power11）

选项	值	说明
通	1	
断	0	

线路12（power12）

选项	值	说明
通	1	
断	0	

线路13（power13）

选项	值	说明
通	1	
断	0	

线路14（power14）

选项	值	说明
通	1	
断	0	

线路15（power15）

选项	值	说明
通	1	
断	0	

线路16（power16）

选项	值	说明
通	1	
断	0	

2. 产品定义

当设备属性状态值因为下面两个原因发生改变时触发：
1. 自身状态变化（如传感器的数值发生变化）；
2. 人为操作（按下了设备按钮）
事件被触发后，平台会携带设备当前状态参数，实时上报消息
具体机制在' 第6节：消息推送 '有详细描述

公共事件（所有产品均支持）

事件	说明
开机	设备加电、云端登记后触发，上报时携带当前状态（各功能属性的值）

本产品的事件

事件	名称	说明
按下按钮	btn1	

2. 产品定义

所有的配置项均保存在设备的Flash中，因为Flash有擦写次数限制，所以平台未开放批量修改设备配置的接口，您可以在控制台来修改设备的配置项。
所有配置项均有默认值，如不了解配置项目意义，请勿随意修改；修改配置项只有在设备在线时，才能成功下发并保存，并且不需要重启设备。

配置项

按钮动作（btn1）

名称	内容	说明
全断	0	默认
全通	1	
禁用	disable	

1路开机状态（relay1）

名称	内容	说明
通	1	
断	0	默认

2路开机状态（relay2）

名称	内容	说明
通	1	
断	0	默认

3路开机状态（relay3）

名称	内容	说明
通	1	
断	0	默认

4路开机状态（relay4）

名称	内容	说明
通	1	
断	0	

5路开机状态（relay5）

名称	内容	说明
通	1	
断	0	

6路开机状态（relay6）

名称	内容	说明
通	1	
断	0	

7路开机状态（relay7）

名称	内容	说明
通	1	
断	0	

8路开机状态（relay8）

名称	内容	说明
通	1	
断	0	

9路开机状态（relay9）

名称	内容	说明
通	1	
断	0	

10路开机状态（relay10）

名称	内容	说明
通	1	
断	0	

11路开机状态（relay11）

名称	内容	说明
通	1	
断	0	

12路开机状态（relay12）

名称	内容	说明
通	1	
断	0	

13路开机状态（relay13）

名称	内容	说明
通	1	
断	0	

14路开机状态（relay14）

名称	内容	说明
通	1	
断	0	

15路开机状态（relay15）

名称	内容	说明
通	1	
断	0	

16路开机状态（relay16）

名称	内容	说明
通	1	
断	0	

1路离线状态（offline1）

名称	内容	说明
通	1	
断	0	
不理会	n	

2路离线状态（offline2）

名称	内容	说明
通	1	
断	0	
不理会	n	

3路离线状态（offline3）

名称	内容	说明
通	1	
断	0	
不理会	n	

4路离线状态（offline4）

名称	内容	说明
通	1	
断	0	
不理会	n	

5路离线状态（offline5）

名称	内容	说明
通	1	
断	0	
不理会	n	

6路离线状态（offline6）

名称	内容	说明
通	1	
断	0	
不理会	n	

7路离线状态（offline7）

名称	内容	说明
通	1	
断	0	
不理会	n	

8路离线状态（offline8）

名称	内容	说明
通	1	
断	0	
不理会	n	

9路离线状态（offline9）

名称	内容	说明
通	1	
断	0	
不理会	n	

10路离线状态（offline10）

名称	内容	说明
通	1	
断	0	
不理会	n	

11路离线状态（offline11）

名称	内容	说明
通	1	
断	0	
不理会	n	

12路离线状态（offline12）

名称	内容	说明
通	1	
断	0	
不理会	n	

13路离线状态（offline13）

名称	内容	说明
通	1	
断	0	
不理会	n	

14路离线状态（offline14）

名称	内容	说明
通	1	
断	0	
不理会	n	

15路离线状态（offline15）

名称	内容	说明
通	1	
断	0	
不理会	n	

16路离线状态（offline16）

名称	内容	说明
通	1	
断	0	
不理会	n	

负载时序（gap）

名称	内容	说明
无保护	0	
间隔1秒	1	
间隔2秒	2	
间隔3秒	3	

固件更新的原则是兼容性升级，即：新的固件版本会保留原有的功能和指令。
如果当前固件可满足业务需求，则可以不升级最新版本的固件。

最新固件

p103.20241108.v1

ID	大小	发布日期
id.785	279.71 KB	2024-11-08 12:59:14
id.817	279.71 KB	2024-11-20 16:01:52

3. 开放接口

平台开放了以下5类接口供开发者调用；
其中：设备类接口的'向设备下发指令'是用来控制设备的接口，是最常用，也是最重要的；
在'第4节 命令下发'中，有对此接口的详细说明。
如果您仅需控制少量设备，可以大致浏览后跳过本节。
另外需要注意的是，在私有化场景中，因为设备不再连接平台，需开发者自行实现设备管理与控制。

接口采用HTTP协议，使用任何可进行HTTP请求的编程语言均可；
接口调用的方法也完全一致，对于不同的设备，调用方式可复用，仅命令不同。因此：

如仅控制设备： 可以只封装一个函数，将设备ID(整形)和命令(数组)作为参数传入

如需其他管理操作： 可以封装一个类，将设备控制、分组、定时任务等做为类的方法

更加详尽的接口文档，请查看 《接口文档》

接口调用

接口地址：https://api.thingboot.com/{AppID}/{接口列表中的path}/?{其他参数}&sign={sign}&ts={ts}

{AppID} 为您的应用ID（由平台生成），请按 "准备工作" 的引导进入控制台，在开发设置页面查看。
{接口列表中的path} 为接口路径，如下发命令的路径为 device/control
{其他参数} 为允许get方式传入的参数，如lang=cn，则在错误返回时使用中文

请求方式：POST 【建议】，一些简单的命令，也可以使用GET方式

必传的参数

参数	名称	说明
sign	签名	所有请求物联网控制台接口，均需在url中携带此参数sign={sign} 取值方法：{sign} = md5(md5(开发者密码) + 上面的ts参数)，32位字符串
ts	时间戳	所有请求物联网控制台接口，均需在url中携带此参数ts={timestamp} 时间戳

返回信息格式

```
{
   code: 200,     // 状态码，200成功，其他见下方对应描述
   data: {},      // 返回值，如有
   msg: 'ok'      // 消息文本
}
```

3.1 设备类接口

名称	功能
向设备下发指令	控制设备，向设备下发指令
获取设备列表	获取设备列表
获取设备详情	获取设备详情
获取设备配置信息	
维护设备标签	当存在很多设备时，您可能需要通过标签来将标记设备，一台设备可以关联多个标签。
维护设备分组	可以将设备添加到分组，统一管理
获取设备日志	获取设备日志列表

3.2 产品类接口

名称	功能
获取产品列表	获取平台和自定义产品列表
获取产品详情	获取产品详情
获取自有产品列表	获取已有（存量）设备的产品列表

3.3 分组类接口

名称	功能
获取分组列表	获取设备分组列表
创建分组	
修改分组	修改分组
删除分组	
执行命令或动作	控制分组执行命令或动作

3.4 标签类接口

名称	功能
获取标签列表	获取标签列表
创建标签	
修改标签	修改标签
删除标签	删除标签
执行命令或动作	控制标签执行命令或动作

3.5 任务类接口

名称	功能
获取任务列表	读取当前控制台的任务列表
任务创建	创建一个任务
任务详情	获取任务的详情信息
修改任务	修改任务信息
删除任务	删除任务
控制任务	控制任务的启停

4. 命令下发

在平台开放的所有接口中，向设备下发命令是最核心、也是最常用的命令。
初次接入设备时，也建议先实现『向设备下发命令』，再通过其他接口实现另外的功能。
在向设备下发命令前，请先确定设备已连接网络，并在控制台显示在线。

4.1 下发命令接口

接口地址： https://api.thingboot.com/{AppID}/device/control/?sign={sign}&ts={ts}

{AppID} 为您的应用ID（由平台生成），请进入控制台，在开发设置页面查看。
{sign} 和 {ts} 的算法，请查看上一节中的描述

请求参数

参数较长时建议使用POST方式

名称	必填	类型	说明
gateway	否	string	当发给设备的命令需要其关联的网关转发时，需要指定 网关的设备ID，唯一ID（在网关壳体上、控制台均可以找到）指定多个网关时请用间隔符(,或|)连接，最多可以同时指定5台网关
device	否	string	设备ID，设备唯一ID（在设备壳体上、控制台均可以找到） 指定多台设备时请用间隔符(,或|)连接 指定多个设备时，不要这些设备属于同一类产品，但必须有相同的指令
order	否	string	命令，可传JSON字符串(推荐)或直接传参 简单命令：可直接GET或POST：设备的属性名称=属性值 复杂或较长的命令：请POST {"order":{命令内容}} ---------- ---------- 在一些业务场景，需要在命令里携带一些特征信息：如订单号等 可在order中增加一个字段extra，如{"power":1,"extra":"T25030700001"} 在本条命令对应的异步消息推送中，会原样返回此特征信息。 extra只支持32位以内的大小写英文字母和数字（a-zA-Z0-9）

返回结果

需要注意的是，200仅代表平台接收到了合规的设备ID与参数名称，并成功向设备下发命令。

而设备可能已离线，或命令内容 / 参数有误，在设备上并没有看到预期效果。

在一些必须要求反馈的场景，请接收云端的消息推送，通过异步消息来标记设备是否成功执行命令。

名称	类型	说明
code	int	200 命令下发成功。 501 未指定设备ID，设备ID为一个整型数字，或由“|”或“,"连接的多个数字。设备列表接口中输出了此ID，在本控制台的设备列表中也可以查看。 502 设备不存在或没有可用设备，传入设备ID均不存在。请检查设备ID是否正确，设备是否删除。 503 指定了过多设备，一次最多指定100台设备ID 504 一次指定了多个设备ID，但其中有一些并不可用（不存在、无权限或已删除） 50xx 请见“全局错误代码”

4.2.1 产品命令

命令	名称	内容
线路1 power1	通	{"power1":"1"}
	断	{"power1":"0"}

命令	名称	内容
线路2 power2	通	{"power2":"1"}
	断	{"power2":"0"}

命令	名称	内容
线路3 power3	通	{"power3":"1"}
	断	{"power3":"0"}

命令	名称	内容
线路4 power4	通	{"power4":"1"}
	断	{"power4":"0"}

命令	名称	内容
线路5 power5	通	{"power5":"1"}
	断	{"power5":"0"}

命令	名称	内容
线路6 power6	通	{"power6":"1"}
	断	{"power6":"0"}

4. 命令下发

命令	名称	内容
线路7 power7	通	{"power7":"1"}
	断	{"power7":"0"}

命令	名称	内容
线路8 power8	通	{"power8":"1"}
	断	{"power8":"0"}

命令	名称	内容
线路9 power9	通	{"power9":"1"}
	断	{"power9":"0"}

命令	名称	内容
线路10 power10	通	{"power10":"1"}
	断	{"power10":"0"}

命令	名称	内容
线路11 power11	通	{"power11":"1"}
	断	{"power11":"0"}

命令	名称	内容
线路12 power12	通	{"power12":"1"}
	断	{"power12":"0"}

4. 命令下发

命令	名称	内容
线路13 power13	通	{"power13":"1"}
	断	{"power13":"0"}

命令	名称	内容
线路14 power14	通	{"power14":"1"}
	断	{"power14":"0"}

命令	名称	内容
线路15 power15	通	{"power15":"1"}
	断	{"power15":"0"}

命令	名称	内容
线路16 power16	通	{"power16":"1"}
	断	{"power16":"0"}

命令	名称	内容
批量控制 batch	全开	
	全关	
	线路1|3|5|7|9|11|13|15开	
	线路1|3|5|7|9|11|13|15关	
	线路2|4|6|8|10|12|14|16开	
	线路2|4|6|8|10|12|14|16关	

命令	名称	内容
先通后断 point	500毫秒，全部线路	
	1秒，线路1|3|5|7	
	2秒，线路2|4|6|8	

命令	名称	内容
先断后通 reset	500毫秒，全部线路	
	1秒，线路1|3|5|7	
	2秒，线路2|4|6|8	

4.2.2 系统命令

命令	名称	内容
system	获取网络信息	network
	获取设备状态	state
	连接WiFi	wifi_connect
	连接配置的WiFi	wifi_connect_multi
	断开WiFi连接	wifi_disconnect
	进入私有化模式（如配置）	private
	重启	restart

4. 命令下发

4.3 调用代码示例 PHP CURL

```
<?php

// First you need to prepare the following values
// 1. AppID (in your console development settings)
// 2. AppSecret (in your console development settings)
// 3. ts (timestamp of the current time, seconds)
// 4. YourSign = md5(md5(AppSecret)ts);(md5 is an encryption method, AppSecret is the AppSecret prepared above, ts is the timestamp prepared above, concatenate the string after the AppSecret after md5 encryption)
// In simple terms, the signature is md5(md5(your developer password)concatenate the above ts timestamp value); ts is the timestamp of the current timestamp; that is, to perform one MD5 on the developer password (AppSecret), then concatenate the timestamp, and then perform one MD5 on the entire concatenated string
// Core request address: api.thingboot.com/AppID/device/control/?sign=YourSign&&ts=ts;
// Request needs to pass two parameters device and order:
// device[string]: unique device ID, can pass multiple [separated by ,], can be viewed in the console, or pulled through the interface
// order[json string]: the command to be issued, for example:
// {"power":1}, usually to connect the circuit of the switch
// {"power3":0}, usually to close the third line of the switch or controller
// {"play:gbk:16":"hello, welcome to visit"}, let the voice speaker report the specified content
// The same type of product, the command is the same, different product types of commands, please view the product manual page of each product
// Note: you must replace the formal AppID and AppSecret with real-time timestamp calculation of the signature, the request must have device device ID and order command

$curl = curl_init();

$AppID = '开发者ID'; // 替换为实际的 AppID
$AppSecret = '开发者密码'; // 替换为实际的 AppSecret
$ts = time();
$sign = md5(md5($AppSecret) . $ts); //md5(md5(开发者密码)拼接上面的ts参数)
$url = 'https://api.yoyoiot.cn/{appid}/device/control/?sign={sign}&ts={ts}';
$url = str_replace(
    array('{appid}', '{sign}', '{ts}'),
    array($AppID, $sign, $ts),
    $url
);

// 请求体数据
$device = '1878'; // 替换为实际的设备ID；可传多个[用,间隔]
$order = '{"power1":1}'; // 替换为实际的命令
// 构建请求体
$postData = json_encode([
    'device' => $device,
    'order' => json_decode($order) // 确保命令是有效的 JSON 对象
]);
curl_setopt_array($curl, array(
    CURLOPT_URL => $url,
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_ENCODING => '',
    CURLOPT_MAXREDIRS => 10,
    CURLOPT_TIMEOUT => 0,
    CURLOPT_FOLLOWLOCATION => true,
    CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
    CURLOPT_POST => true, // 设置为 POST 请求
    CURLOPT_POSTFIELDS => $postData, // 设置请求体
    CURLOPT_HTTPHEADER => array(
        'Content-Type: application/json', // 设置请求头为 JSON 格式
    ),
));
$response = curl_exec($curl);
curl_close($curl);
echo $response;
```

5. 配置下发

通过配置下发接口，可以远程修改设备的配置参数。
所有配置项均保存在设备的Flash中，因为Flash有擦写次数限制，所以平台未开放批量修改设备配置的接口。
修改配置项只有在设备在线时，才能成功下发并保存，并且不需要重启设备。

5.1 下发配置接口

接口地址： https://api.thingboot.com/{AppID}/device/config/?sign={sign}&ts={ts}

{AppID} 为您的应用ID（由平台生成），请进入控制台，在开发设置页面查看。
{sign} 和 {ts} 的算法，请查看"3. 开放接口"章节中的描述

请求参数：

名称	必填	类型	说明
device	是	string	设备唯一ID，支持传多个（用英文逗号分隔）
config	是	json	配置参数，格式为JSON对象，如：{"参数名":"参数值"}

调用示例：

```
POST https://api.thingboot.com/{AppID}/device/config/?sign={sign}&ts={ts}
Content-Type: application/json

{
   "device": "设备ID",
   "config": {"配置参数名": "参数值"}
}
```

5. 配置下发

以下列出本产品支持的所有配置参数，请根据实际需求选择需要修改的配置项。
所有配置项均有默认值，如不了解配置项目意义，请勿随意修改。

产品支持配置参数

按钮动作（btn1）

配置	名称	内容
btn1	全断	{"btn1":"0"}
btn1	全通	{"btn1":"1"}
btn1	禁用	{"btn1":"disable"}

1路开机状态（relay1）

配置	名称	内容
relay1	通	{"relay1":"1"}
relay1	断	{"relay1":"0"}

2路开机状态（relay2）

配置	名称	内容
relay2	通	{"relay2":"1"}
relay2	断	{"relay2":"0"}

3路开机状态（relay3）

配置	名称	内容
relay3	通	{"relay3":"1"}
relay3	断	{"relay3":"0"}

4路开机状态（relay4）

配置	名称	内容
relay4	通	{"relay4":"1"}
relay4	断	{"relay4":"0"}

5. 配置下发

5路开机状态（relay5）

配置	名称	内容
relay5	通	{"relay5":"1"}
relay5	断	{"relay5":"0"}

6路开机状态（relay6）

配置	名称	内容
relay6	通	{"relay6":"1"}
relay6	断	{"relay6":"0"}

7路开机状态（relay7）

配置	名称	内容
relay7	通	{"relay7":"1"}
relay7	断	{"relay7":"0"}

8路开机状态（relay8）

配置	名称	内容
relay8	通	{"relay8":"1"}
relay8	断	{"relay8":"0"}

9路开机状态（relay9）

配置	名称	内容
relay9	通	{"relay9":"1"}
relay9	断	{"relay9":"0"}

10路开机状态（relay10）

配置	名称	内容
relay10	通	{"relay10":"1"}
relay10	断	{"relay10":"0"}

11路开机状态（relay11）

配置	名称	内容
relay11	通	{"relay11":"1"}
relay11	断	{"relay11":"0"}

12路开机状态（relay12）

配置	名称	内容
relay12	通	{"relay12":"1"}
relay12	断	{"relay12":"0"}

13路开机状态（relay13）

配置	名称	内容
relay13	通	{"relay13":"1"}
relay13	断	{"relay13":"0"}

14路开机状态（relay14）

配置	名称	内容
relay14	通	{"relay14":"1"}
relay14	断	{"relay14":"0"}

15路开机状态（relay15）

配置	名称	内容
relay15	通	{"relay15":"1"}
relay15	断	{"relay15":"0"}

16路开机状态（relay16）

配置	名称	内容
relay16	通	{"relay16":"1"}
relay16	断	{"relay16":"0"}

5. 配置下发

1路离线状态（offline1）

配置	名称	内容
offline1	通	{"offline1":"1"}
offline1	断	{"offline1":"0"}
offline1	不理会	{"offline1":"n"}

2路离线状态（offline2）

配置	名称	内容
offline2	通	{"offline2":"1"}
offline2	断	{"offline2":"0"}
offline2	不理会	{"offline2":"n"}

3路离线状态（offline3）

配置	名称	内容
offline3	通	{"offline3":"1"}
offline3	断	{"offline3":"0"}
offline3	不理会	{"offline3":"n"}

4路离线状态（offline4）

配置	名称	内容
offline4	通	{"offline4":"1"}
offline4	断	{"offline4":"0"}
offline4	不理会	{"offline4":"n"}

5路离线状态（offline5）

配置	名称	内容
offline5	通	{"offline5":"1"}
offline5	断	{"offline5":"0"}
offline5	不理会	{"offline5":"n"}

6路离线状态（offline6）

配置	名称	内容
offline6	通	{"offline6":"1"}
offline6	断	{"offline6":"0"}
offline6	不理会	{"offline6":"n"}

7路离线状态（offline7）

配置	名称	内容
offline7	通	{"offline7":"1"}
offline7	断	{"offline7":"0"}
offline7	不理会	{"offline7":"n"}

8路离线状态（offline8）

配置	名称	内容
offline8	通	{"offline8":"1"}
offline8	断	{"offline8":"0"}
offline8	不理会	{"offline8":"n"}

9路离线状态（offline9）

配置	名称	内容
offline9	通	{"offline9":"1"}
offline9	断	{"offline9":"0"}
offline9	不理会	{"offline9":"n"}

10路离线状态（offline10）

配置	名称	内容
offline10	通	{"offline10":"1"}
offline10	断	{"offline10":"0"}
offline10	不理会	{"offline10":"n"}

11路离线状态（offline11）

配置	名称	内容
offline11	通	{"offline11":"1"}
offline11	断	{"offline11":"0"}
offline11	不理会	{"offline11":"n"}

12路离线状态（offline12）

配置	名称	内容
offline12	通	{"offline12":"1"}
offline12	断	{"offline12":"0"}
offline12	不理会	{"offline12":"n"}

13路离线状态（offline13）

配置	名称	内容
offline13	通	{"offline13":"1"}
offline13	断	{"offline13":"0"}
offline13	不理会	{"offline13":"n"}

14路离线状态（offline14）

配置	名称	内容
offline14	通	{"offline14":"1"}
offline14	断	{"offline14":"0"}
offline14	不理会	{"offline14":"n"}

15路离线状态（offline15）

配置	名称	内容
offline15	通	{"offline15":"1"}
offline15	断	{"offline15":"0"}
offline15	不理会	{"offline15":"n"}

5. 配置下发

16路离线状态（offline16）

配置	名称	内容
offline16	通	{"offline16":"1"}
offline16	断	{"offline16":"0"}
offline16	不理会	{"offline16":"n"}

负载时序（gap）

配置	名称	内容
gap	无保护	{"gap":"0"}
gap	间隔1秒	{"gap":"1"}
gap	间隔2秒	{"gap":"2"}
gap	间隔3秒	{"gap":"3"}

6. 消息推送

除了向设备下命令之外，平台会把上行消息（从设备到平台的消息）转发给您；
消息上报仅对传感类、或具备采样功能的产品是必需的。
因此，您可以选择是否接收设备的上报消息。

平台消息分为上行消息和下行消息两大类：

下行消息	由您通过HTTP请求发出，云平台来响应您的请求，并立即向设备下发命令。 如控制设备、修改设备分组等属性、创建定时任务等
上行消息	由设备发给平台，如果您设置了接收这些消息，云平台会将这些消息实时推送到这个地址； 如设备上下线、命令应答、事件（用户按下按钮）、状态上报（温度功率等）

6.1 消息推送机制

平台支持两种消息推送机制，建议使用MQTT方式

方式	说明
MQTT方式	实时上报，无频率限制，稳定性高
HTTP方式	上报频率 1秒 / 条，如存在带有计量功能的设备、或设备数量较大、对及时性要求较高时，请使用mqtt方式上报

6.2 开通消息推送

消息推送是可选的，当前也是免费的。

请在您的物联网控制台的左侧导航[开发设置]中，
随时打开 / 关闭消息推送。

6.3 推送的消息类型

名称	文档
上/下线消息	查看
指令执行消息	查看
设备触发的事件消息	查看
设备自主上报的状态消息	查看

7. 安装与配网

使用手机开放热点的方式给设备配网

7.1 正式使用场景设备配网

打开手机热点,将热点的名称修改为 tb-您的工作台ID;
您的工作台ID控制台右上角查看

如果您的工作台ID为XXXX,则将热点的名称改成：tb-XXXX;

密码设置为：12345678

请设置手机热点：

热点名称：	tb-XXXX
热点密码：	12345678

注：手机开放的热点频段必须为 2.4G，若设备长时间没有成功配网，尝试给设备断电后重新通电等待设备自行配网，或者使用其他手机链接该热点尝试是否可以上网

7.2 设备调试配网

需要设备进入调试模式时，通过该方式配网，设备将回到云平台，即可进行调试。
打开手机热点,将热点的名称修改为 tb;
设备调式方式联网时，设备会忽略私有化的配置信息

密码设置为：12345678

请设置手机热点：

热点名称：	tb
热点密码：	12345678

注：手机开放的热点频段必须为 2.4G，若设备长时间没有成功配网，尝试给设备断电后重新通电等待设备自行配网，或者使用其他手机链接该热点尝试是否可以上网

8. 私有化部署说明

芯步的所有智能设备出厂时，均默认连接芯步物联网平台。

物联网平台由Broker(代理/中间件）、消息服务器、各种监测模块组成，部署在公有云的若干台服务器上。

在一些特定的场景，需要私有化支持：

- 对设备数据/日志，有较高的统计/分析需求
- 无法访问公网（外网），无法访问芯步平台
- 依据政策/法规要求，设备数据存储必须本地化
- 已有物联网平台，需要接入自有平台，集中管理

8.1 私有化方式

芯步提供了两类方式，来实现私有化：

所有方式均支持公网和局域网部署

下面列出了两类方式的适用场景，请根据您的情形选择适合的方案。

方式	适用场景
网关方式 Gateway	适合少量设备（几十台）的私有化，尤其适合局域网 （部署简单、方便） **无网关**：直接修改设备配置 请注意：无网关方式无法接收设备的上行消息 通过设备自身接口：http://设备ip地址/control 来实现设备控制 **软网关**：在网络内的服务器上安装"UNI-Gateway"软件 通过 http://网关地址:网关端口/api/control 来控制指定设备 **硬网关**：在网络中安装芯步"智能网关"设备，工作原理与软网关类似 通过 http://网关地址:网关端口/api/control 来控制指定设备
代理方式 Broker	对设备规模数量无要求，更加灵活 （可在局域网，可以公网） **平台Broker**：在局域网或云服务器上安装"UNI-Broker"软件 通过与平台一致的http接口来控制设备 **自建Broker**：在局域网或云服务器上，新建或使用现有Broker 编程对接MQTT，实现设备控制

8.2 说明

芯步出品的所有智能设备均支持私有化，

请按照《私有化文档》进行私有化相应的操作与部署。

客服工程师会协助您完成私有化，为您提供相应的技术支持，

但由于物联网云平台的建设和调试较复杂，

我们无法免费为您提供物联网云平台的建设服务。