



# 网络编程提升篇

如何连接到 onenet 云

# 目录

- OneNET 介绍
- 具体示例
- 创建应用
- 注意事项



# OneNET 介绍

# OneNET 介绍

- [OneNET](#) 是由中国移动打造的PaaS物联网开放平台。平台能够帮助开发者轻松实现设备接入与设备连接，快速完成产品开发部署，为智能硬件、智能家居产品提供完善的物联网解决方案。
- OneNET 软件包是 RT-Thread 针对 OneNET 平台连接做的的适配，通过这个软件包，可以让设备在 RT-Thread 上非常方便的连接 OneNet 平台，完成数据的发送、接收、设备的注册和控制等功能。

# OneNET 软件包功能特点

- 软件包具有以下优点：
- 断线重连
  - RT-Thread OneNET 软件包实现了断线重连机制，在断网或网络不稳定导致连接断开时，会维护登陆状态，重新连接，并自动登陆 OneNET 平台。提高连接的可靠性，增加了软件包的易用性。
- 自动注册
  - RT-Thread OneNET 软件包实现了设备自动注册功能。不需要在 web 页面上手动的一个一个创建设备，输入设备名字和鉴权信息。当开启设备注册功能后，设备第一次登陆 OneNET 平台时，会自动调用注册函数向 OneNET 平台注册设备，并将返回的设备信息保存下来，用于下次登陆。

# OneNET 软件包功能特点

- 自定义响应函数
  - RT-Thread OneNET 软件包提供了一个命令响应回调函数，当 OneNET 平台下发命令后，RT-Thread 会自动调用命令响应回调函数，用户处理完命令后，返回要发送的响应内容，RT-Thread 会自动将响应发回 OneNET 平台。
- 自定义 topic 和回调函数
  - RT-Thread OneNET 软件包除了可以响应 OneNET 官方 topic 下发的命令外，还可以订阅用户自定义的 topic，并为每个 topic 单独设置一个命令处理回调函数。方便用户开发自定义功能。
- 上传二进制数据
  - RT-Thread OneNET 软件包除了上传数字和字符串外，还支持二进制文件上传。当启用了 RT-Thread 的文件系统后，可以直接将文件系统内的文件以二进制的方式上传至云端。



# 具体示例

# 准备工作

- 设备接入 OneNET 云之前，需要在平台注册用户账号，
- OneNET 云平台地址：<https://open.iot.10086.cn>



# 创建产品

- 账号注册登录成功后，点击**开发者中心**进入开发者中心界面；
- 点击 **创建产品**，
- 输入产品基本参数，页面最下方设备接入协议选择 **MQTT** 协议，如下图所示：

OneNET 中移物联网开放平台

首页 案例与伙伴 开发文档 发现 社区 动态

开发者中心

0 个公开协议产品 0 个私有协议产品

创建产品

模组选择：

☒ 其他 ☐ 安信可 ESP8266-XX系列

☐ 庆科 EMW3081

设备接入协议：

MQTT

MQTT 协议基本功能介绍：

1. 上报传感器数据点
2. 订阅自定义TOPIC
3. 接收实时消息或离线消息

确定 取消


# 创建产品

- 产品创建成功之后，可以在开发者中心左侧**产品概况**中查看产品基础信息
- （如**产品ID**，接入协议，创建时间，**产品 APIkey** 等，后面配置工程时会用到）。



# 添加设备

- 在开发者中心左侧 **设备管理** 中点击 **添加设备** 按钮添加设备



The screenshot shows a horizontal bar with a search input field on the left containing the placeholder text "输入设备ID或者设备名称". To the right of the input field is a magnifying glass icon. Further right is a blue button with a plus icon and the text "添加设备", which is highlighted with a red rectangular border. To the right of this button is another button with a document icon and the text "导出".

- 鉴权信息是为了区分每一个不同的设备，（这里仅为了测试就填写了当前时间作为鉴权信息）填完之后点击接入设备

# 添加设备

接入设备

MQTT协议

• 设备名称:

test1

• 鉴权信息:

201807171718

这里的鉴权信息每一个  
设备都要是不一样的

设备间不能设置相同的字符串，最多512个字母、数字或字母与数字组合的字符串。

• 数据保密性:



私有



公开

接入设备

取消

# 添加 APIkey

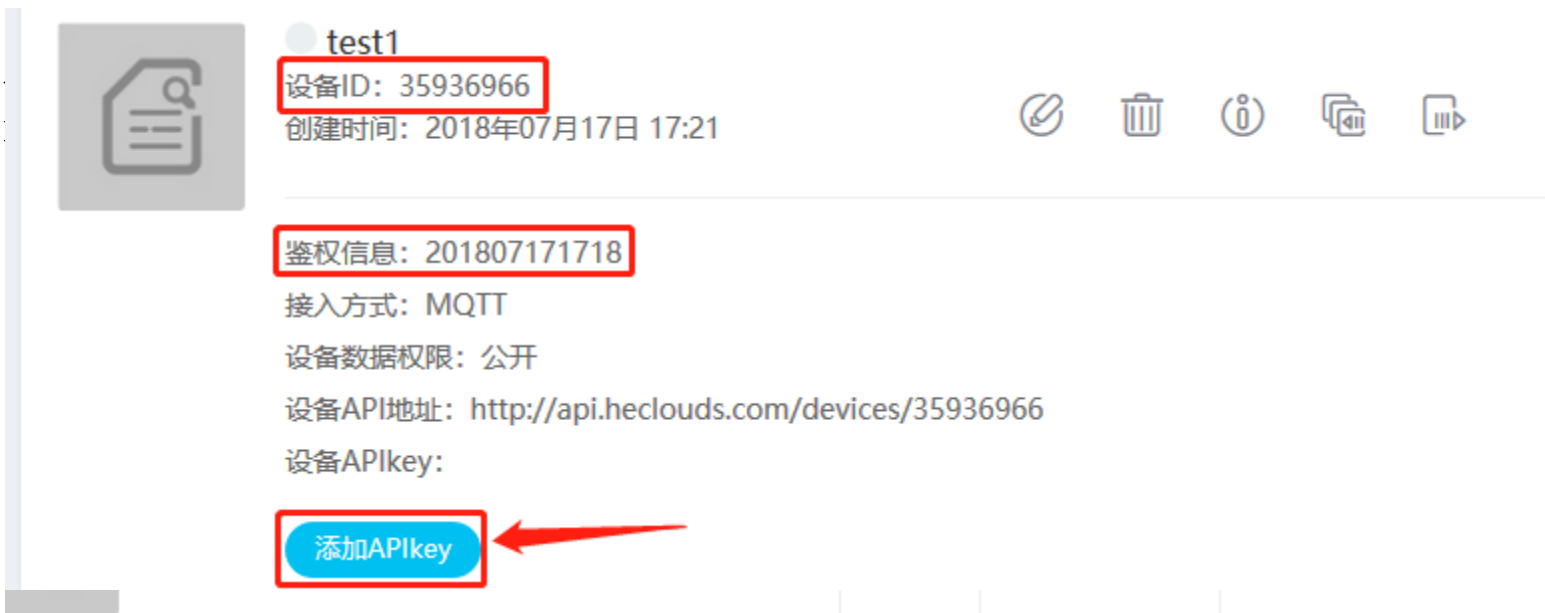
- 接入设备之后，可以看到设备管理的界面多了一个设备，设备的右边有一些操作设备的按钮，点击查看详情按钮



- 此设备的相关信息就都显示出来了，比如：设备 ID、鉴权信息、设备 APIkey，这些信息需要记下，在ENV配置时会用到。

# 添加 APIkey

- 接入设备  
作设备的



边有一些操

- 此设备的相关信息就都显示出来了，比如：设备 ID、鉴权信息、设备 APIkey，这些信息需要记下，在ENV配置时会用到。

# 添加 APIkey

- 点击按钮添加APIkey，APIKey 的名称一般和设备相关联，我们这里填入 test\_APIKey，关联设备写入我们刚刚创建的设备test1。

The screenshot shows a dialog box for adding an API key. It has two input fields: 'APIKey名称:' (API Key Name) and '关联设备:' (Associated Device). The first field contains 'test\_APIKey' and the second field contains 'test1'. A red arrow points from the text '关联我们刚才创建的设备' (Associate the device we just created) to the 'test1' input. At the bottom, there are two buttons: '添加APIKey' (Add API Key) and '取消' (Cancel). A red arrow points to the '添加APIKey' button. The background shows a blurred view of the '设备信息' (Device Information) page with a '基本信息' (Basic Information) section.

设备信息

• 基本信息

• APIKey名称:

test\_APIKey

• 关联设备:

test1 ← 关联我们刚才创建的设备

添加APIKey 取消

# 工程配置

- 开启 onenet 软件包
- 打开 env 工具输入 **menuconfig** 按照下面的路径开启 onenet 软件包

```
RT-Thread online packages
IoT - internet of things --->
  IoT Cloud --->
    [*] OneNET: China Mobile OneNet cloud SDK for RT-Thread
```

- 进入 onenet 软件包的配置菜单按下图所示配置，里面的信息依据自己的产品和设备的实际情况填写，并且开启示例代码

```
--- OneNET: China Mobile OneNet cloud SDK for RT-Thread
[*] Enable OneNET sample
[*] Enable support MQTT protocol
[ ] Enable OneNET automatic register device
(35936966) device id
(201807171718) auth info
(H3ak5Bbl0NxpW3QVVe33InnPx0g=) api key
(156418) product id
(dVZ=ZjVJvGjXIUDsbropzg1a8Dw=) master/product apikey
version (latest) --->
```



# 工程配置

- 配置项介绍
  - Enable OneNET sample: 开启 OneNET 示例代码
  - Enable support MQTT protocol: 开启 MQTT 协议支持
  - Enable OneNET automatic register device: 开启 OneNET 自动注册设备功能
  - device id: 配置云端创建设备时获取的 设备ID
  - auth info: 配置云端创建产品时 用户自定义的鉴权信息 (每个产品的每个设备唯一)
  - api key: 配置云端创建设备时获取的 APIkey
  - product id: 配置云端创建产品时获取的 产品ID
  - master/product apikey: 配置云端创建产品时获取的 主/产品APIKey

# 工程配置

- 利用 `env` 工具配置好 `onenet` 软件包之后，退出并保存
  - 更新软件包 `pkgs --update`
  - 编译工程 `scons`
  - 然后运行 `qemu`
- 
- 我们可以在工程的 `packages` 下的 `onenet-latest` 目录下的 `samples` 文件夹下看到 `onenet_sample.c` 文件。
- 
- 该文件是 **OneNET** 软件包的示例展示，主要是向用户展示如何使用 **OneNET** 软件包上传数据和接收命令。

# 运行示例代码

- 输入 **scons** 编译工程
- 然后输入 **qemu** 运行程序
- 系统运行起来后，先在 **msh** 命令行下输入 **onenet\_mqtt\_init** 命令初始化 onenet 。

```
msh> onenet_mqtt_init
```

- 运行结果如下图所示

```
msh />onenet_mqtt_init
[D/ONENET] (mqtt_connect_callback:85) Enter mqtt_connect_callback!
[D/MQTT] ipv4 address port: 6002
[D/MQTT] HOST = '183.230.40.39'
[I/ONENET] RT-Thread OneNET package(V0.2.0) initialize success.
msh />[I/MQTT] MQTT server connect success
[D/ONENET] (mqtt_online_callback:90) Enter mqtt_online_callback!
```

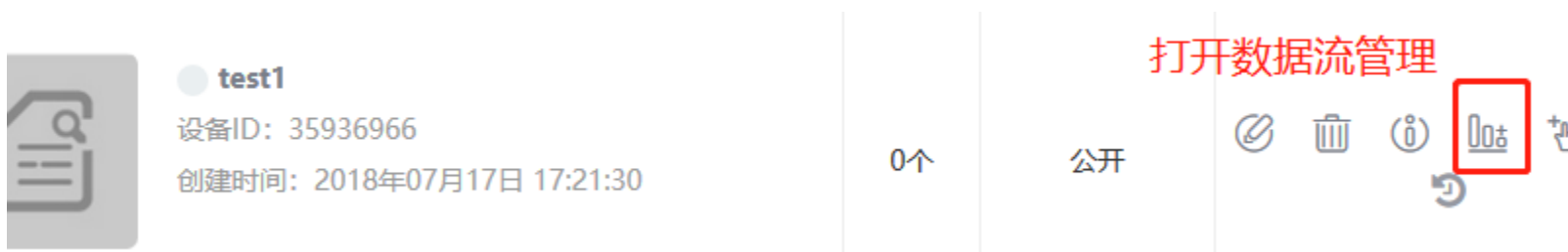
# 预期结果

- 然后输入命令 `onenet_upload_cycle` 运行示例程序，结果如下图所示：

```
msh />onenet_upload_cycle
msh />[D/ONENET] (onenet_upload_data:106) buffer : {"temperature":0}
[D/ONENET] (onenet_upload_data:106) buffer : {"temperature":56}
[D/ONENET] (onenet_upload_data:106) buffer : {"temperature":56}
[D/ONENET] (onenet_upload_data:106) buffer : {"temperature":19}
[D/ONENET] (onenet_upload_data:106) buffer : {"temperature":11}
[D/ONENET] (onenet_upload_data:106) buffer : {"temperature":51}
[D/ONENET] (onenet_upload_data:106) buffer : {"temperature":43}
[D/ONENET] (onenet_upload_data:106) buffer : {"temperature":5}
[D/ONENET] (onenet_upload_data:106) buffer : {"temperature":8}
[D/ONENET] (onenet_upload_data:106) buffer : {"temperature":93}
[D/ONENET] (onenet_upload_data:106) buffer : {"temperature":30}
[D/ONENET] (onenet_upload_data:106) buffer : {"temperature":66}
```

# 预期结果

- 在 onenet 上查看上传上去的数据流
- 点击网页上 设备管理 里面设备的 数据流管理 操作



- 刷新网页，能看到云端能接收到开发板发来的数据。





# 创建应用

# 创建应用

- 创建应用来展示接收到的数据流
- 点击左侧 **应用管理** 然后点击 **创建应用**



独立应用 模板应用

请输入应用名称进行搜索

要向您的应用列表中添加应用，请点击右上方的“创建应用”按钮。

创建应用

- 输入名称并点击创建



创建 取消

# 创建应用

页面 + ↑ ↓ 未命名 ×

未命名

元件库

基础元素

- 文本
- 折线图
- 柱状图
- 图片
- 链接
- 地图
- 仪表盘

控制元素

- 旋钮
- 开关
- 命令

室内温度

50  
40  
30  
20  
10  
0

08-15 14:53:01

属性 样式

轴颜色

轴颜色 #333

时间数据点格式

{y}-{m}-{d}

时间支持以下通配符

- {Y}: 四位年份
- {y}: 两位年份
- {m}: 两位月份
- {d}: 两位日期
- {h}: 两位小时
- {i}: 两位分钟
- {s}: 两位秒数

数据点个数

10

line-chart

属性 样式

图例

- 室内温度

☒ 显示图例 +

选择数据流

设备: test1

数据流: temperature

刷新频率

图层

输入关键字查找 查找

- line-chart





# 注意事项

# 注意事项

- 电脑需要关闭防火墙