

第8章 使用 HTTP 协议与中国移动物联网平台通信

8.1 基础知识

书中 8.1.8 节介绍的内容已经无法使用，读者可以跳过。但是 8.1.1 节~8.1.7 节仍有实践价值，读者务必阅读。

由于本章的内容适配了新版的 OneNET 平台，因而此处提供了新版平台的操作方法，读者可以按照下列介绍创建项目和设备。

8.1.1 中国移动 OneNET 物联网平台的使用

1. OneNET 物联网平台简介

OneNET 是中国移动打造的高效、稳定、安全的物联网开放平台。它提供了丰富的 API 和应用模板，可以支撑各类行业应用和智能硬件的开发，能有效降低物联网应用开发和部署的成本，满足物联网领域设备连接、协议适配、数据存储、数据安全以及大数据分析等平台级服务需求。

OneNET 具备接入增强、边缘计算、增值能力、AI、数据分析、一站式开发、行业能力、生态开放等特点。可提供设备接入、设备管理、位置定位（LBS）、远程升级（OTA）、消息队列（MQ）、数据可视化（View）、人工智能（AI）、视频能力、边缘计算等功能。

2. 在 OneNET 平台中新建项目

新版 OneNET 平台上线后，停止了原有的多协议接入服务。下面以新版平台的“物联网开放平台”为例，介绍其使用方法。

在 OneNET 平台的首页点击“开发者中心”，进入“物联网开放平台”主页，如图 8-1 所示。



图 8-1 “物联网开放平台”主页

点击左侧的“产品开发”，可以看到当前的产品列表，如图 8-2 所示。



图 8-2 “产品开发”页面

点击右侧的“创建产品”。依次选择产品种类，如智慧生活-环境电器-环境监测盒子。智能化方式选择“设备接入”，并依次设置设备信息（节点类型：直连设备，接入协议：HTTP，数据协议：数据协议，联网方式：任意，开发方案：自定义方案），如图 8-3 所示。

选择智能化方式 重新选择

设备接入

3 填写信息

* 产品名称

简易气象站

* 所属地市

江苏省

南京市

* 节点类型

直连设备

网关设备

子设备

* 接入协议

HTTP

* 数据协议

OneJson

* 联网方式

蜂窝

Wi-Fi

NB

以太网

其他

* 开发方案

标准方案

自定义方案

图 8-3 填写设备信息

确定后，在产品开发页面便可以看到新建的产品。点击产品右侧的“产品开发”，可以打开产品的详细信息页面，如图 8-4 所示。



图 8-4 产品的详细信息页面

下面需要为产品设置“物模型”，也就是设置产品的功能、数据等。点击图 8-4 右侧的“设置物模型”按钮，选择“添加自定义功能点”，打开“添加自定义功能点”界面。在简易气象站中，一共有温度、湿度等 7 个物理量，因此需要添加 7 个功能点。图 8-5 是添加“温度”功能点的参数。其它参数亦同。

添加自定义功能点



* 功能类型: 属性类型

* 功能名称: 温度

* 标识符: temp

* 数据类型: float(单精度浮点型)

* 取值范围: -20 - 45

步长: 请输入数据精度; 如身高需要精确到1, 则输入1

单位: 请选择单位

* 读写类型: 只读

提示: 请输入设备信息

新增 取消

图 8-5 添加“温度”功能点

图 8-6 是定义好的功能点列表。定义完成后，点击列表右上角的保存按钮即可。

功能类型	功能类别	功能名称	标识符	数据类型	数据值定义	读写类型
属性	自定义	温度	temp	float(单精度浮点型)	取值范围: -20-45	只读
属性	自定义	照度	illumi	int32(整数型)	取值范围: 0-200000	只读
属性	自定义	湿度	humi	int32(整数型)	取值范围: 0-100	只读
属性	自定义	海拔	alti	int32(整数型)	取值范围: -200-9000	只读
属性	自定义	气压	pres	float(单精度浮点型)	取值范围: 90-110	只读
属性	自定义	风速	windspeed	float(单精度浮点型)	取值范围: 0-20	只读
属性	自定义	风向	winddirection	int32(整数型)	取值范围: 0-360	只读

图 8-6 定义好的功能点

3. 在项目中添加设备

按照 OneNET 的资源模型，每一个物理设备都应该添加到平台上。在实际应用中，往往是设备在激活时调用平台的 API（Application Programming Interface，应用程序编程接口），

从而自动完成自动注册。此处作为例子，设备的数量较少，因此采用手工注册的方式。

点击图 8-4 所示页面左侧的“接入设备管理”-“设备管理”，打开设备列表，如图 8-7 所示。



图 8-7 设备列表

点击列表右侧的“添加设备”按钮，在弹出的界面中输入设备信息（所属产品和设备名称必填，其余选填），如图 8-8 所示。输入完成后，点击确定按钮即可。

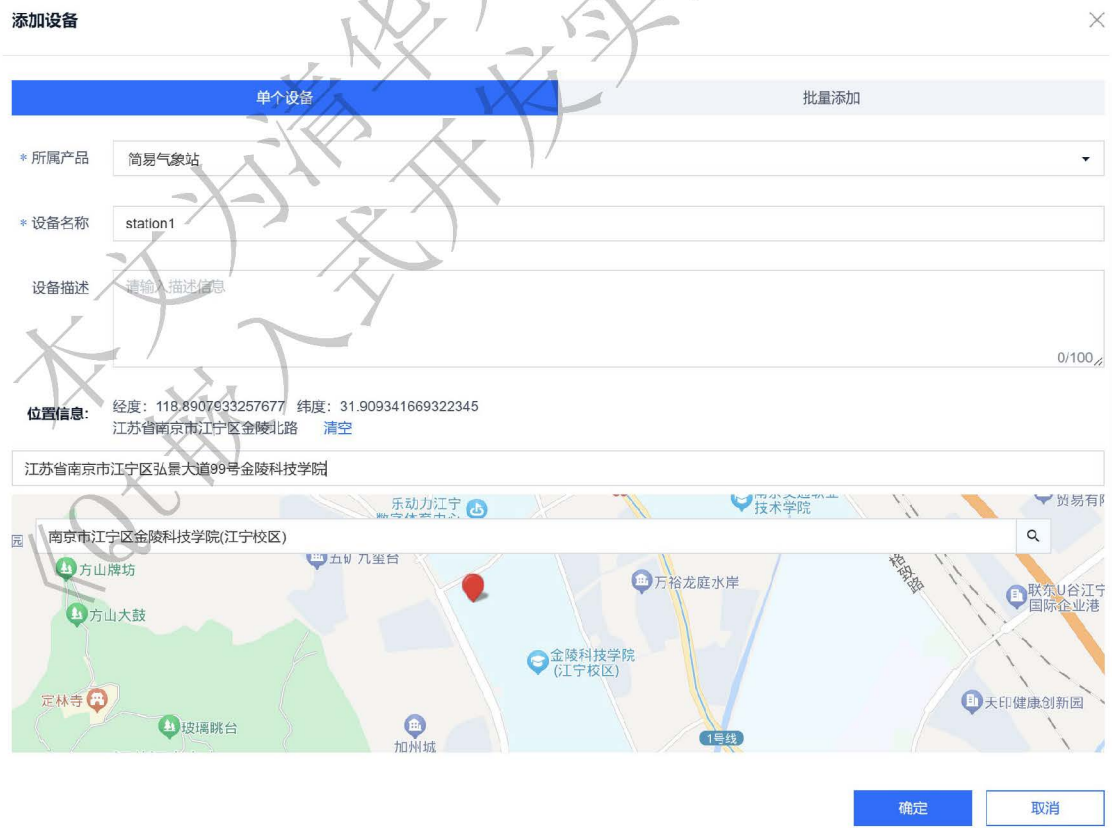


图 8-8 添加设备界面

图 8-9 是添加完设备后的设备列表中。点击设备右侧的详情按钮，可以查看设备的详细信息，如图 8-10 所示。

设备列表

批次列表

若您无“NB”套件、MQTT套件的存量设备，则新增设备后，列表不再展示“设备ID”

🔄

设备状态 (全部)

设备来源 (全部)

添加时间

请选择时间

📅

设备名称

请输入设备名称

🔍

🗑️

↶ 导出设备

+ 添加设备

<div>📄</div> 设备名称/ID	设备状态	所属产品/产品ID	最近在线时间	设备来源	操作
<div><input type="checkbox"/></div> station1	<div>🕒 未激活</div>	简易气象站 产品ID: 5k9cgZ094n	-	自主创建	<div>详情 / 删除</div>
<div><input type="checkbox"/></div> device1	<div>🟢 在线</div>	测试项目 产品ID: 0413K5s6Zx	2054-06-01 10:49:06	自主创建	<div>详情 / 删除</div>

图 8-9 新添加的设备已经出现在了设备列表中



设备接入管理 > 设备详情					
station1					
设备详情 属性 事件记录 服务记录 操作记录 文件管理 SIM卡查询					
设备信息					
设备密钥: N05EQUV1MnBwdXNEYJvrU... Auth_Code: - PSK: -					
创建时间: 2024-03-26 21:51:02 激活时间: - 最近在线时间: -					
设备状态: 未激活 操作系统版本号: - 设备位置: 江苏省南京市江宁区弘景大道99号金陵科技学院					
设备描述: -					
产品信息					
产品ID: 5k9cgZ094n 产品名称: 简易气象站 产品类型: 智慧生活 > 环境电器 > 环境检测盒子					
智能化方式: 设备接入 开发方案: 自定义方案 节点类型: 直连设备					
联网方式: Wi-Fi 接入协议: HTTP					

图 8-10 设备详细信息页面

在设备详细信息页面中，需要留意图中框出的三处信息。它们由上到下以此为：设备名称、设备密钥、产品 ID。在后续的操作中，需要用到这三处信息。

如果有更多设备，可以按照上述步骤继续添加。

4. Token 的计算

在新版 OneNET 平台中，对用户身份进行了严格校验。用户必须要提供一个特定的 Token 才能进行各种操作。平台提供了一个用于计算 Token 的工具。读者可以在配套工具和资料包中找到它。

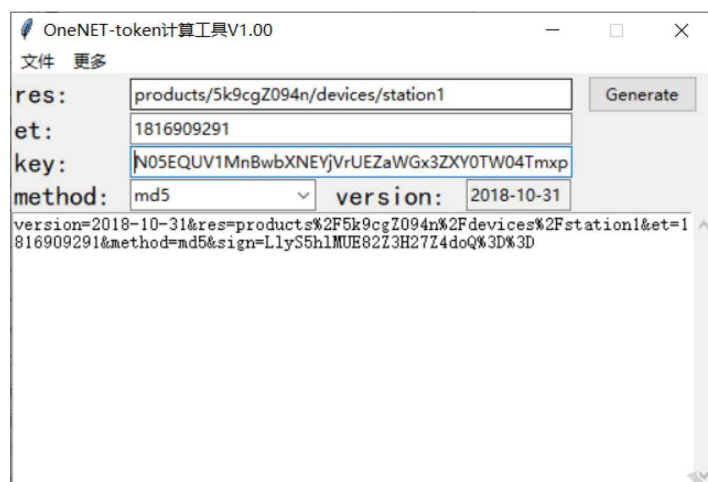


图 8-11 OneNET 平台的 Token 计算工具

图 8-11 是 Token 计算工具的界面。计算 Token 时，需要提供三个信息。res 的格式是固定的，只需要根据实际情况替换产品 ID 和设备名称即可：

products/产品 ID/devices/设备名称

et 是 Token 的过期时间，数值是 Unix 时间戳，即过期时间和 1970 年 1 月 1 日 00:00 之间相差的秒数。图 8-11 中的时间戳大约在 2027 年 7 月底过期。

key 则是设备密钥，在创建设备时由系统自动生成。

填入上述信息后，点击 Generate 按钮，便可以得到 Token 字符串。

8.2 实践案例：简易气象站程序 V3.0 的实现

本节继续对 V2.0 版本的气象站程序进行更新，增加 HTTP 通信功能。通信时使用 JSON 数据格式将数据发送到 OneNET 物联网平台（参见示例代码 \ch8New\ch8-5SimpleWeatherStationV3.0\）。在本节开始前，需要在 OneNET 平台的多协议接入服务中新建一个 HTTP 协议工程，并在工程中新建一个设备（本章使用名为“气象站”的设备作为例子进行讲解）。具体新建过程与新建 TCP 透传工程类似。

8.2.1 JSON 数据的生成

向 OneNET 平台发送数据时，需要遵循平台指定的 OneJSON 格式。该格式的具体定义参见 OneNET 平台的帮助文档。下面是适用于本气象站的一组 JSON 数据：

```
{
  "id": "123",
  "params": {
    "temp": {
      "value": 25
    },
    "humi": {
      "value": 55
    }
  }
}
```

```

        "illumi": {
            "value": 8888
        },
        "alti": {
            "value": 10
        },
        "pres": {
            "value": 101.325
        }
    }
}

```

其中 `id` 是由用户自定义的数字，此处固定为 123。如果 JSON 中不增加 `id`，OneNET 服务器会返回错误。

有了数据格式后，就可以在 `ClassGY39` 类中增加 `dataToJSON()` 函数，用于将数据转化为 JSON 格式：

```

QByteArray ClassGY39::dataToJSON()
{
    cJSON *root;

    root = cJSON_CreateObject();

    cJSON_AddStringToObject(root, "id", "123"); //勿忘 id

    cJSON *params = cJSON_CreateObject();

    //重复下列代码段，为所有的物理量都创建 JSON Object
    cJSON *jsontemp = cJSON_CreateObject();
    cJSON_AddNumberToObject(jsontemp, "value", int(getTemperature() * 10)
/ 10.0);

    //重复下列代码段，将上两行创建的 JSON Object 添加到 root 中
    cJSON_AddItemToObject(params, "temp", jsontemp); //temp 是图 8-6 中定义的
温度变量的标识符

    cJSON_AddItemToObject(root, "params", params);

    char *cUnformattedJSON = cJSON_PrintUnformatted(root);
    QByteArray JSONData = QByteArray(cUnformattedJSON);
    free(cUnformattedJSON);
    cJSON_Delete(root);

    return JSONData;
}

```

类似的，也可以为 `ClassPR3000` 类增加一个 `dataToJSON()` 函数。该函数的代码与上面的代码类似，可以查看示例代码中的代码。

8.2.2 HTTP 发送函数的实现

按照上一章中生成 TCP 数据的思路,每个硬件类都能将自己的数据转换为 JSON 数据。但是两个硬件类生成的 JSON 数据无法简单地合并在一起。所以在实现 HTTP 通信时,分两次进行发送。

考虑到每次进行 HTTP 通信时都需要设置 URL、设置请求头,所以编写了函数 `slot_HTTPSendToOneNET()`,负责控制这两次发送。同时编写了函数 `HTTPSendJSON()`,用于将给定的 JSON 数据发送到服务器。

首先看 HTTP 发送函数 `HTTPSendJSON(QByteArray content)`。根据 OneNET 开发文档的说明,OneNET 对 POST 请求报文的格式要求如下:

```
POST
/fuse/http/device/thing/property/post?topic=$sys/**/$$$/thing/property/post
&protocol=http
Host:open.iot.10086.cn
Token:***
Content-Length:nnn

OneJSON 数据
```

其中第 1 行 URL 中的***是项目 ID, \$\$\$是设备名称,需要根据不同的设备调整。Token 字段是鉴权信息。由于它不是 Qt 内置的字段,需要调用 `setRawHeader()`函数添加。综上, `HTTPSendJSON()`函数的代码如下:

```
QByteArray MainWindow::sendHTTPData(QByteArray content)
{
    QNetworkRequest requestInfo;

    // 设置 SSL 配置
    QSslConfiguration sslConfig =
    QSslConfiguration::defaultConfiguration();
    requestInfo.setSslConfiguration(sslConfig);

    QString url =
    QString("https://open.iot.10086.cn/fuse/http/device/thing/property/post?topi
c=$sys/产品 ID/设备名称/thing/property/post&protocol=http");
    requestInfo.setUrl(QUrl(url));
    requestInfo.setRawHeader("Token", "version=2018-10-
31&res=products%2F5k9cgZ094n%2Fdevices%2Fstation1&et=1816909291&method=md5&s
ign=FesGu0Dekb43PL4yzTQmpg%3D%3D");
    requestInfo.setHeader(QNetworkRequest::ContentTypeHeader,
    "application/json");

    QNetworkReply *reply = m_netManager->post(requestInfo, content);
    QEventLoop eventLoop;
    connect(reply, SIGNAL(finished()), &eventLoop, SLOT(quit()));
    QTimer::singleShot(1000, &eventLoop, SLOT(quit()));
}
```

```

eventLoop.exec();

if (reply->error() != QNetworkReply::NoError) {
    printLog("HTTPS 通信失败", reply->errorString());
    return QByteArray(reply->errorString().toUtf8());
}

QByteArray qbaResponse = reply->readAll();
printLog("HTTPS 服务器回应", qbaResponse);
reply->deleteLater();
return qbaResponse;
}

```

与前文介绍的 HTTP 通信操作相比，上述函数增加了 SSL（Secure Socket Layer，安全套接层）配置的内容。SSL 是在 TCP/IP 协议上实现的一种安全协议，采用公开密钥技术，保护通信的私密性。OneNET 平台服务器要求采用 SSL 通信，因而作为客户端需要配合服务器的要求。如果此处不使用 SSL 通信，则服务器会返回 301 错误。需要注意的是，Qt5.14.2 在使用 SSL 通信时，需要调用 OpenSSL 的库文件 libcrypto-1_1-x64.dll 和 libssl-1_1-x64.dll。读者可以在配套工具和资料包中找到这两个文件，并将它们复制到 Qt 安装路径中，如 D:\Qt\Qt5.14.2\5.14.2\mingw73_64\bin\。

槽函数 slot_HTTPSendToOneNET() 用于控制通信流程，内容相对而言比较简单：

```

void MainWindow::slot_HTTPSendData()
{
    QByteArray qbaJSONData = m_GY39Device->dataToJSON(); //生成 JSON
    HTTPSendJSON(qbaJSONData); //发送 JSON
    qbaJSONData = m_PR3000Device->dataToJSON(); //生成 JSON
    HTTPSendJSON(qbaJSONData); //发送 JSON
}

```

除了上述函数，还需要为 HTTP 通信开关控件 imageSwitchHTTP 添加鼠标单击事件处理函数。由于这部分内容前文已经多次提到，此处不再给出代码，可在示例代码中查看相关内容。

8.3 程序运行结果

这一节主要完成了程序的 HTTP 通信功能。该功能需要联合 OneNET 网站进行测试。

(1) 由于在代码中直接写入了产品 ID、设备 ID，因而无需填写界面上 HTTP 通信的参数。直接打开 HTTP 通信开关。日志区会输出日志，提示 HTTP 通信已打开，如图 8-12 所示。



图 8-12 在程序中打开 HTTP 通信开关

(2) 如图 8-13 所示, 在程序中读取一组测量数据, 程序会将数据转化为两条 JSON 数据上传到 OneNET 平台。如果数据的格式和内容正确, OneNET 平台会返回两组提示信息, 提示数据上传成功:

```
15:11:42 使用 HTTP 发送数据
15:11:42 HTTP 服务器回应 {"errno":0,"error":"succ"}
15:11:42 使用 HTTP 发送数据
15:11:42 HTTP 服务器回应 {"errno":0,"error":"succ"}
```

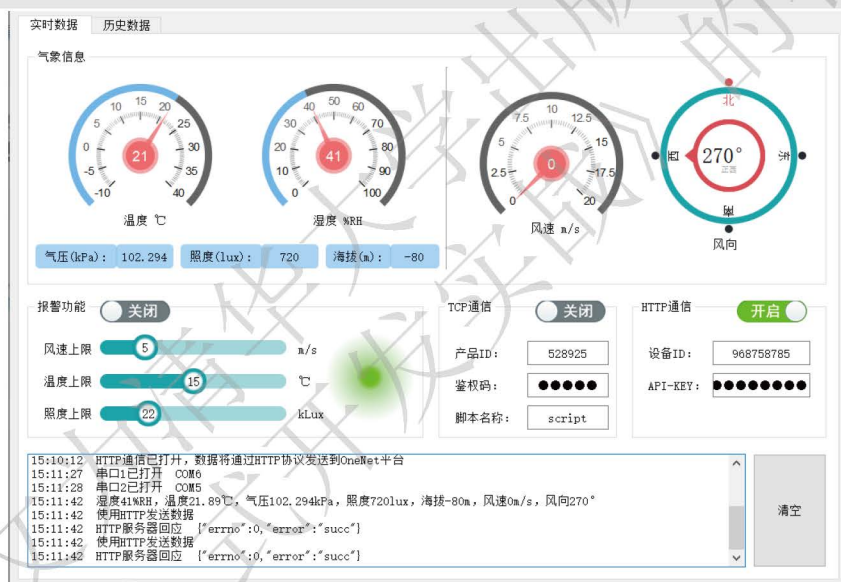


图 8-13 程序从硬件读取一组测量数据并发送到 OneNET 平台

同时在 OneNET 平台对应设备的属性页面中也会显示接收到的数据, 如图 8-14 所示。

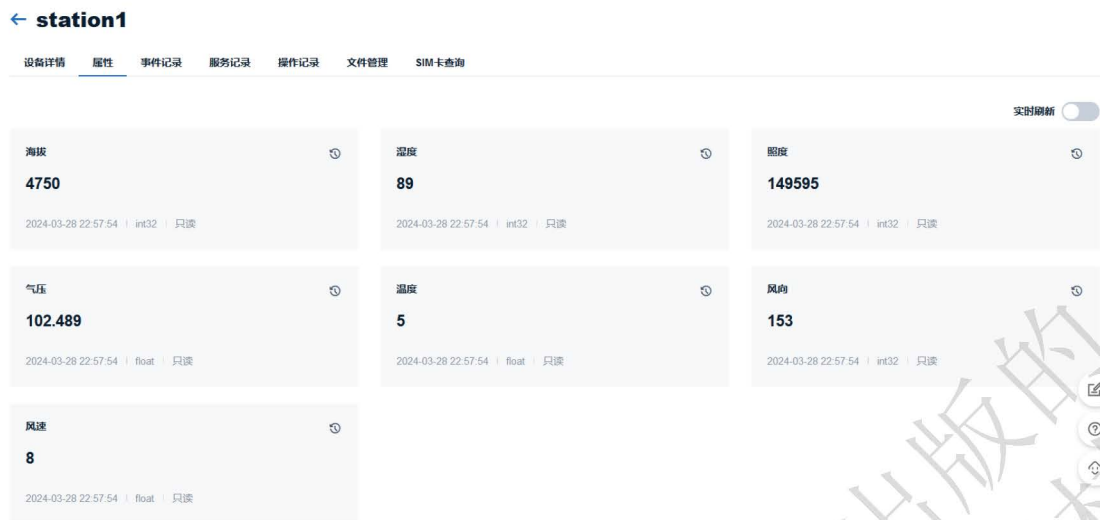


图 8-14 OneNET 接收到了程序发送的数据