

墨子号科技给用户介绍下如何接入 ONENET 平台采用 MQTT 协议接入说明。首先在百度当中输入 ONENET 然后点击进入 ONENET 平台后，就可以进入到平台端了。第一次使用需要注册，现在一般都是实名制要求，用户注册的时候，根据注册需要，来进行操作即可。



ONENET 登录界面

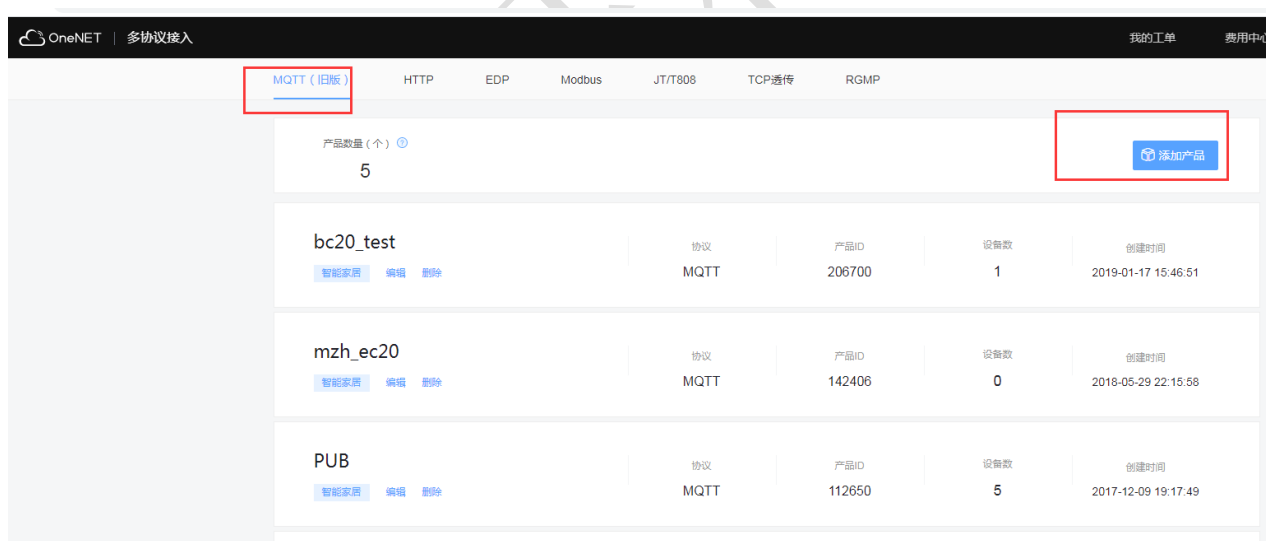
登录进入之后，就可以进行设备产品添加了，注意这里的登录协议采用 MQTT 协议，不是 MQTTS，因为现在 ONENET 改版，推荐的是 MQTTS，但是 MQTTS 目前不适用我们应用。所以用户在使用的时候，请务必选择 MQTT 这个功能的类目接入平台。



在产品服务里面，找到 MQTT 物联网套件，然后点击立即使用。进入到 MQTT 产品界面端。



进入到界面之后，因为这个界面我们有添加过很多设备，新注册的用户打开这个界面不一定一样，不过可以看到下面有一行原 MQTT 产品接入迁移的说明，告知用户需要使用 MQTT 协议可以点击前往旧版的标签，这样注册的产品就是 MQTT 协议了。



点击进入之后，可以看到他有很多的协议类型接入，根据不同的产品来实现。对于我们的用法来说，将采用 MQTT 协议接入，其他的协议是用不到的，有兴趣的用户可以选择尝试使用其他的方式来进行数据接入。

添加产品

产品信息

• 产品名称：
1-16个字符

• 产品行业：
请选择

• 产品类别：
请选择 请选择 请选择

产品简介：
1-200个字符

技术参数

• 联网方式：
☐ wifi ☐ 移动蜂窝网络

• 设备接入协议：
MQTT(旧版)

若要创建其他协议套件的产品请前往相应协议套件下创建

• 操作系统：
☐ Linux ☐ Android ☐ VxWorks ☐ µC/OS ☐ 无

然后用户就可以添加产品了，在添加产品的地方，红色星号都是要求用户必须填写的，一般最好都是用英文字符数字代替，尽量不要出现中文名称。联网方式选择移动蜂窝网络，操作系统选择无即可。



添加好产品之后，就可以添加设备了。点击立即添加设备，因为产品下面可以包含多个设备，并且这样的设备就是对应的硬件。所以一定要添加设备方可使用。

添加设备里面的有个鉴权信息，这个非常重要，后面在设备登录的时候，需要用到，显示应该是可以自己定义的数据，具体鉴权信息用户可以自己添加，使用数字与符号，尽量不要用一些特殊字符代替。

这样就注册好了，然后设备会显示离线，这时只要单片机端控制模块发送数据到 ONENET 平台进行激活就可以了。

4.9.2 单片机程序实现

下面就是对照程序即可。这里采用的是 EC20 内置的 MQTT 代码规范来实现的。MQTT 协议用户可以参考 EC20 MQTT 手册来进行学习。具体是

“Quectel_EC2x&EG9x&EM05_MQTT_Application_Note_V1.1” 来看相关指令。

```
printf("AT+QMTCFG=\"version\",0,4\r\n");//设备版本
delay_ms(500);
printf("AT+QMTOPEN=0,\"183.230.40.39\",6002\r\n");//通过TCP方式去连接MQTT服务器
delay_ms(500);
strx=strstr((const char*)RxBuffer,(const char*)"+QMTOPEN: 0,0");//看下返回状态
while(strx==NULL)
{
    strx=strstr((const char*)RxBuffer,(const char*)"+QMTOPEN: 0,0");//确认返回值正确
}
Clear_Buffer();
```

上面的是 MQTT 设置指令，需要注意 ONENET 使用的版本是 3.11 版本的 MQTT 协议，默认是 3.1 的，所以需要先设置下版本，否则是无法连接到 ONENET 平台

的。

设置好版本之后，就可以去连接 ONENET 的服务器了。其实这里的登录服务器与 TCP 登录没有区别，所以这里采用 TCP 指令或者是 MQTT 指令实现的效果都是一样的。

注意 IP 和端口，如果没有错的话，就会返回登录成功 0, 0。那么登录到服务器的第一步是成功了。

```
printf("AT+QMTCONN=0,\"31580807\", \"112650\", \"201808160002\"\\r\\n");//去登录MQTT服务器，设备ID，产品ID,鉴权.
delay_ms(500);
strx=strstr((const char*)RxBuffer, (const char*)"+QMTCONN: 0,0,0");//看下返回状态
while(strx==NULL)
{
    strx=strstr((const char*)RxBuffer, (const char*)"+QMTCONN: 0,0,0");//看下返回状态
}
Clear_Buffer();
```

登录到服务器，需要输入登录验证，因为 ONENET 为了规范，不可能让任意一个非法的用户登录，也会让每一个登录的用户找到自己的设备。所以需要输入相关的参数来接入，这个在后面的阿里云接入也是一样需要的。

这里的参数主要有设备 ID, 产品 ID, 鉴权三个元素组成。

那么此时可以查看下 ONENET 平台的参数找到这些参数。上面的参数值是我之前就设定好的设备，用户需要根据自己的参数来改动。

The screenshot displays the ONENET platform interface. At the top, there's a header for 'mzhec20' with tabs for '平板电视', '编辑', and '详情'. Below this, a table lists key parameters: '产品ID' (Product ID) is 310766, '用户ID' (User ID) is 70253, 'Master-APIkey' is '查看', 'access_key' is '查看', and '设备接入协议' (Device Access Protocol) is MQTT. Below the table, there are statistics: '设备数量(个)' (Device Count) is 1, '在线设备数(个)' (Online Device Count) is 0, and '设备注册码' (Device Registration Code) is Nvm70GoDzR8JvhUp. The bottom section shows a list of devices. The first device has '设备ID' (Device ID) 580515635, '设备名称' (Device Name) mzh001, '设备状态' (Device Status) 离线 (Offline), and '最后在线时间' (Last Online Time) -. The device ID 580515635 is highlighted with a red box. At the bottom, there's a pagination bar showing '共1项' (Total 1 item) and '1' page.

在产品里面有个设备列表，打开设备列表可以看到设备 ID。这个也是关键参数，需要填入到代码当中。



在右边的详情里面有一个鉴权信息，这个鉴权信息就是我们自己设定的，所以如果用户记不清了，就可以找到，然后填入进去。

这样三个参数就找完并填好后，编译代码即可。登录验证成功后，就可以发布数据到服务器端了。

```
DHT11_Read_Data(&temp,&humi) ;//获取温湿度传感器DHT11数据
json_len=MqttOnenet_Savedata(t_payload,temp,humi);//提交JSON数据
sprintf((char*)datastr,"%d",json_len);
printf("AT+QMTPUBEX=0,0,0,\"$dp\", \"%s\r\n\",datastr);//发布消息
delay_ms(500);
Uart3_SendDATA(t_payload,json_len);//发json数据到ONENET
strx=strstr((const char*)RxBuffer,(const char*)"+QMTPUBEX: 0,0,0");//发布成功 +QMTPUBEX: 0,0,0
while(strx==NULL)
{
    strx=strstr((const char*)RxBuffer,(const char*)"+QMTPUBEX: 0,0,0");//发布成功 +QMTPUBEX: 0,0,0
}
```

以上代码是获取到温湿度数据之后，然后拼接成 JSON 格式数据，然后发到服务器，注意 MQTT 发送数据都要带主题内容的。那么 ONENET 上显示的主题名称必须是“\$dp”，这个在 ONENET 的数据实现的教程里面有介绍，用户可以看他的手册来掌握设计的原理。

```
//访问ONENET需要提交JSON数据
u8 MqttOnenet_Savedata(u8 *t_payload,u8 temp,u8 humi)
{
    char json[]="{\"datastreams\": [{\"id\": \"temp\", \"datapoints\": [{\"value\": %d}]}, {\"id\": \"humi\", \"datapoints\": [{\"value\": %d}]}] }";
    // char json[]="{\"datastreams\": [{\"id\": \"location\", \"datapoints\": [{\"value\": \"%lon\": %2.6f, \"lat\": %2.6f}]}] }";
    char t_json[200];
    unsigned short json_len;
    sprintf(t_json, json, temp, humi);
    json_len = strlen(t_json)/sizeof(char);
    //type
    t_payload[0] = '\x01';
    //length
    t_payload[1] = (json_len & 0xFF00) >> 8;
    t_payload[2] = json_len & 0xFF;
    //json
    memcpy(t_payload+3, t_json, json_len);
    return json_len+3;
}
```

上面就是提交到 ONENET 的数据格式，是完全遵照 JSON 格式来进行设计的，分为数据流，ID 号，数据点，数据值，这样的表达方式。

发送成功后，会有发送数据成功响应，并可以在 ONENET 端看到数据了。那么可以验证下效果。



好的，在数据展现区里面看到了数据，他也有历史曲线图，非常的方便。用户只要根据上面的说明，更改下参数即可显示板子自带的温湿度传感器采集到的温湿度数据了。