



SimFAS 深克斯中控系统

MQTTc 客户端使用

中控连入 MQTT 物联网云端服务器

MQTTC 模块的使用说明

内部文档,仅供合作伙伴使用,请勿公开.

上海深克斯科技有限公司 Copyright@2008-2022

SimFAS 使用 MQTT 客户端 MQTTc

<适用于中控专家模式 V7.26+以上>

MQTTc 简介:

用户可以在中控上使用 MQTTc 客户端连入各种主流的云端 MQTT 服务器,包括常见的阿里云,腾讯云,和自己创建的 MQTT 服务器. 可以收发并处理数据.

1. 支持验证: 支持用户名密码
2. 发布信息: 支持向特定的通道发布信息
3. 订阅信息: 支持订阅特定的通道信息.
4. Will 信息: 支持 will 断线通知信息.

开始使用:

在中控上的代码市场搜索 mqttc 客户端,并导入到中控主机上. (中控 WAN 口需先连入到外网上)

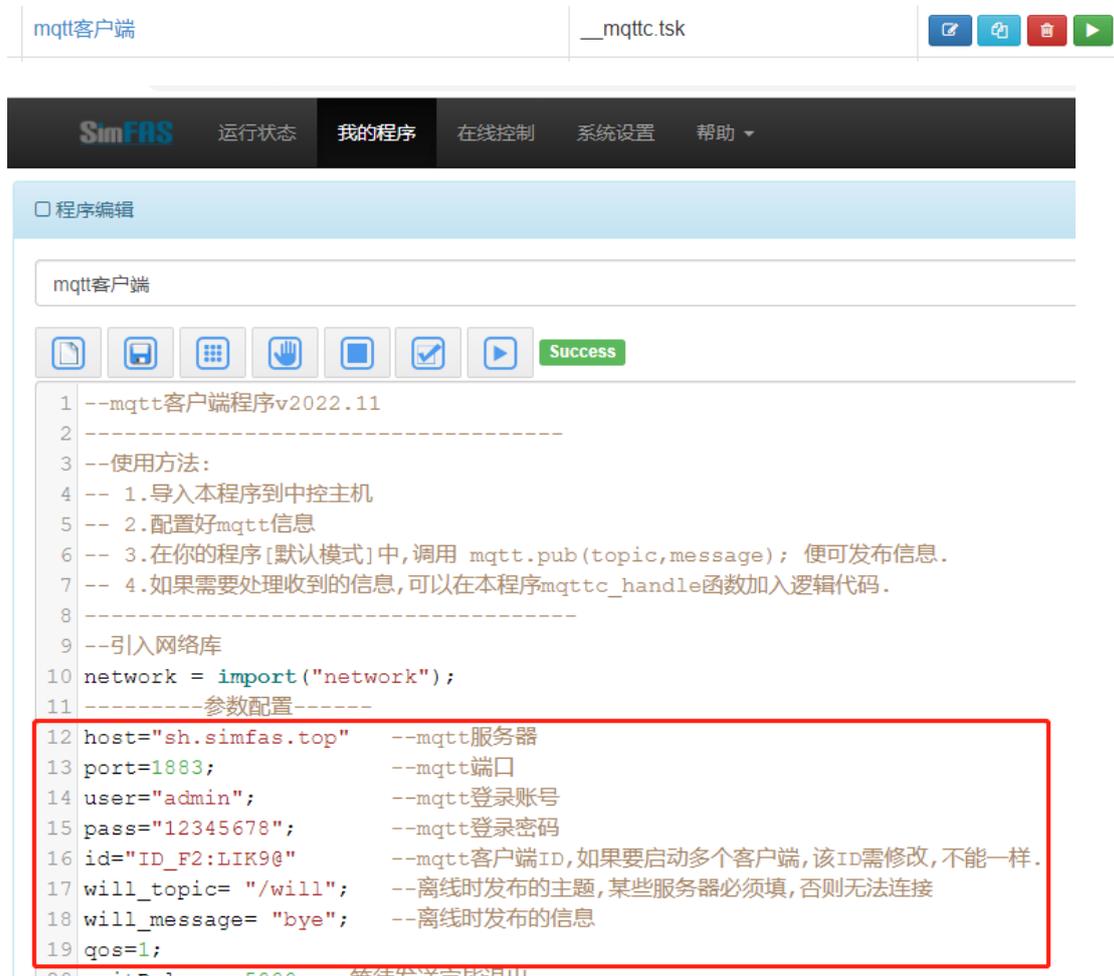


导入到中控主机:



配置 MQTT 登录信息:

打开刚刚导入的程序,并根据供应商给的 mqtt 登录信息,修改对应的登录服务器,端口,账号密码等信息.



The screenshot shows the SimFAS IDE interface. At the top, there are tabs for 'mqtt客户端' and '__mqttc.tsk'. Below the tabs, there is a navigation bar with 'SimFAS', '运行状态', '我的程序', '在线控制', '系统设置', and '帮助'. The main area is titled '程序编辑' and contains a code editor for 'mqtt客户端'. The code editor shows the following code:

```
1 --mqtt客户端程序v2022.11
2 -----
3 --使用方法:
4 -- 1.导入本程序到中控主机
5 -- 2.配置好mqtt信息
6 -- 3.在你的程序[默认模式]中,调用 mqtt.pub(topic,message); 便可发布信息.
7 -- 4.如果需要处理收到的信息,可以在本程序mqttc_handle函数加入逻辑代码.
8 -----
9 --引入网络库
10 network = import("network");
11 ----- 参数配置-----
12 host="sh.simfas.top"    --mqtt服务器
13 port=1883;             --mqtt端口
14 user="admin";          --mqtt登录账号
15 pass="12345678";       --mqtt登录密码
16 id="ID_F2:LIK90"       --mqtt客户端ID,如果要启动多个客户端,该ID需修改,不能一样.
17 will_topic= "/will";  --离线时发布的主题,某些服务器必须填,否则无法连接
18 will_message= "bye";  --离线时发布的信息
19 qos=1;
```

The code editor also shows a 'Success' message and a toolbar with icons for file operations and execution.

启动程序:

导入 MQTT 客户端修改正确的信息后,可以通过下图的箭头指向的运行按钮,运行上述程序,程序运行后,默认情况下,会一直在后台运行,直到您停止它.



The screenshot shows the SimFAS IDE interface with the 'mqtt客户端' tab selected. The code editor is empty. The toolbar at the bottom right has a red arrow pointing to the green 'Run' button.

发布信息:

MQTT 客户端运行后,便可以在你的程序调用 `mqtt.pub(topic,data)`; 进行发布信息.



The screenshot shows the SimFAS IDE interface with the 'mqtt客户端' tab selected. The code editor contains the following code:

```
1 mqtt.pub("application/1/device/24E124147C326922/tx","hello");
```

The toolbar at the bottom right has a green 'OK' button.

接收信息:

MQTT 客户端收到信息后,会调用函数 `mqttc_handle`, 您可以在 39 行处加入您的逻辑代码,收到的主题和信息,可以通过变量 `cli.topic` 和 `cli.data` 访问.



```
程序编辑
mqtt客户端
--print("mqtt连接成功:",host);
mqttc1:mqttc_sub(sub_ch,qos);
mqttc1:mqttc_pub(pub_ch,pub_dat,qos);
print("mqtt发布信息",pub_ch);
end
if (cli.event=="message") then
39   print("收到:",cli.topic,cli.data);
40 end
41
```

程序停止:



该操作会重启所有用户应用程序.

用户服务器程序自启动:

程序调试完毕后,如果需要每次中控开机/重启后,服务器程序自动启动,那么可以在中控初始化的时候,调用您定义的服务器,其中函数是 `call`, 参数是您定义的服务器程序的控制代码, 步骤如下.

1. 打开并编辑下面程序:

» 4	程序初始化	ON_APP_INI.tsk	   
» 5	事件 系统设置	SYSTEM_SETTING	   

2. 加入启动程序代码 `call("__mqttc.tsk");`； --注意引号内参数为您 `mqtt` 程序的控制代码。

程序初始化









Success

```

1  --//该程序将用于初始化某些全局变量
2  --//如果需要设置中控通电自动运行，建议使用：系统设置--
3  --call("_RTC.SET.tsk?from=local");
4  sleep(2);
5  --启动用户服务器
6  call("__mqttc.tsk");|

```

常见问题:

Q1: 是否可以在回调函数中使用 `sleep` 延时等.

A1: 不要在回调函数使用任何延时(`sleep/msleep/usleep`)指令,这会导致丢失无法及时处理客户端的数据请求和返回.

Q2: 提示错误 , 模块 `network` 不存在.









错误@行 10: 模块 'network' 不存在:

A2: 新建程序的时候,需要手动修改控制代码,前面以 2 个下划线开始 `__xxx.tsk`

程序引擎	专家模式
控制代码	<code>__mqttc.tsk</code>

以如果上述操作无效,请把中控更新到最新版本.

更多服务器端代码例子:

请在中控的代码市场搜索相关的关键字: `mqttc`

还有另外一个版本的 MQTT 客户端程序,
代码市场请搜索: **mqtt 客户端测试 v2023**

如您有疑问, 请联系我们:

提供实时免费的各种服务

 编程协助	 APP制作协助
 调试协助	 远程Debug

 177-0180-5710  simfas  cs@simfas.com

 上海深克斯科技有限公司
<http://www.simfas.cn/>

MQTTC 客户端库程序的源代码

```
--mqtt 客户端程序 v2022.11
-----
--使用方法:
-- 1.导入本程序到中控主机
-- 2.配置好 mqtt 信息
-- 3.在你的程序[默认模式]中,调用 mqtt.pub(topic,message); 便可发布信息.
-- 4.如果需要处理收到的信息,可以在本程序 mqttc_handle 函数加入逻辑代码.
-----

--引入网络库
network = import("network");
-----参数配置-----
host="sh.simfas.top"  --mqtt 服务器
port=1883;           --mqtt 端口
user="admin";        --mqtt 登录账号
pass="12345678";     --mqtt 登录密码
id="ID_F2:LIK9@"     --mqtt 客户端 ID,如果要启动多个客户端,该 ID 需修改,不能一样.
will_topic= "/will"; --离线时发布的主题,某些服务器必须填,否则无法连接
will_message= "bye"; --离线时发布的信息
qos=1;
exitDelayms=5000; --等待发送完毕退出
-----联机后发布和订阅信息-----
--发布通道
pub_ch="application/1/device/24E124147C326922/tx";
--发布消息 [[括号里面直接填写内容,不需要转义]]
pub_dat=[[{"confirmed":true,"fport":85,"data":"/ykR"}]];
--订阅信息
sub_ch="application/1/device/24E124147C326922/rx";

-----

--mqtt 客户端回调函数
function mqttc_handle(cli)
  if (cli.event=="open") then
    --print("mqtt 连接成功:",host);
    mqttc1:mqttc_sub(sub_ch,qos);
    mqttc1:mqttc_pub(pub_ch,pub_dat,qos);
    print("mqtt 发布信息",pub_ch);
  end
  if (cli.event=="message") then
    print("收到:",cli.topic,cli.data);
  end
end
```

```
if (cli.event=="error") then
    print("mqtt 错误:",host);
end
if (cli.event=="close") then
    print("mqtt 连接断开:",host);
end

if (cli.event=="timer") then
    --print("定时器任务");
end
end

function pub_message()
    if (GetKV("mqttc_new")==="Y") then
        --print("有新消息要发布");
        SetKV("mqttc_new","N");
        mqttc_topic=GetKV("mqttc_topic");
        mqttc_data=GetKV("mqttc_data");
        mqttc1:mqttc_pub(mqttc_topic,mqttc_data,0);
    end
end
--启动 mqtt 客户端

mqttc1 = network.start_mqtt_cli
(host,user,pass,port,id,will_topic,will_message,mqttc_handle);

loops=0;
--循环等待信息
while true do
    mqttc1:poll();
    loops=loops+1;
    pub_message();
    --if loops>exitDelayms then break; end
end
print("退出程序");
mqttc1:free();
```