



物联网云平台综合设计实验指导书

信息物理系统课题组制

实验

1. 实验目的

- 熟悉阿里云物联网云平台
- 熟悉 Python 语言及其编程环境
- 了解 MQTT 协议特点
- 掌握物联网云平台上的应用设计实例

2. 实验环境

- Visual Studio Code 编程软件、python 语言环境、阿里云物联网平台。

3. 实验内容

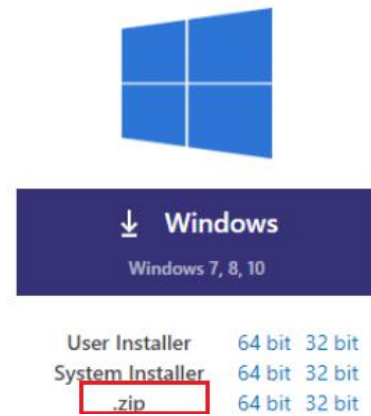
- 实验环境的安装
- 物联网云平台的注册及使用
- 物联网产品设计：要求多属性，并实现云端与多个设备间的数据通信。

4. 实验步骤

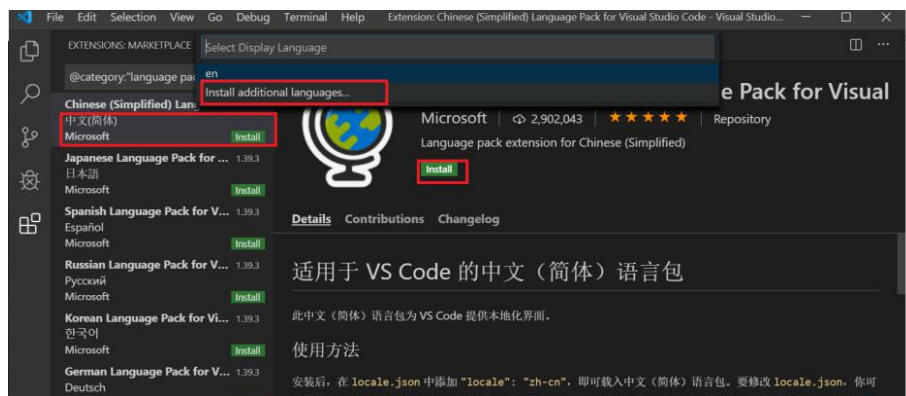
➤ 实验环境的安装

(1) 软件 Visual Studio Code 的安装及汉化

打开 VScode 官网，下载最新版软件，建议 zip 格式。解压到非系统盘，文件夹最好不要出现中文和空格。



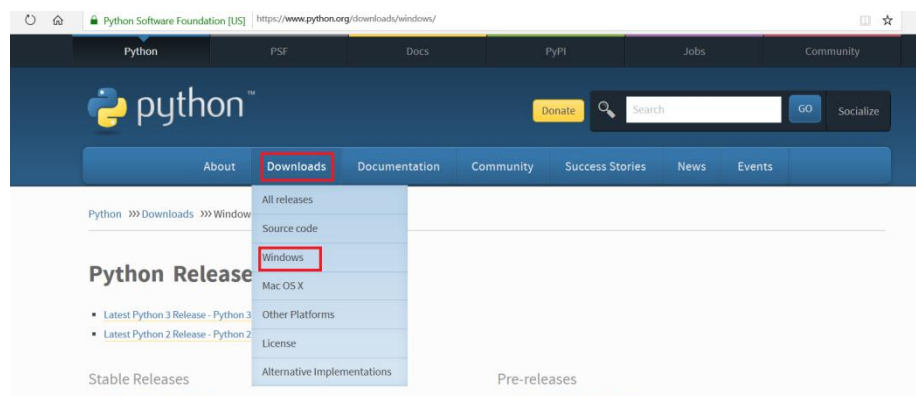
安装好后使用快捷键 Ctrl+Shift+P 弹出查找命令框，输入 language，找到 Configure Display Language，选择简体中文安装，如下图所示：



重启 VSCode，可见页面已经实现汉化。

(2) python 安装流程

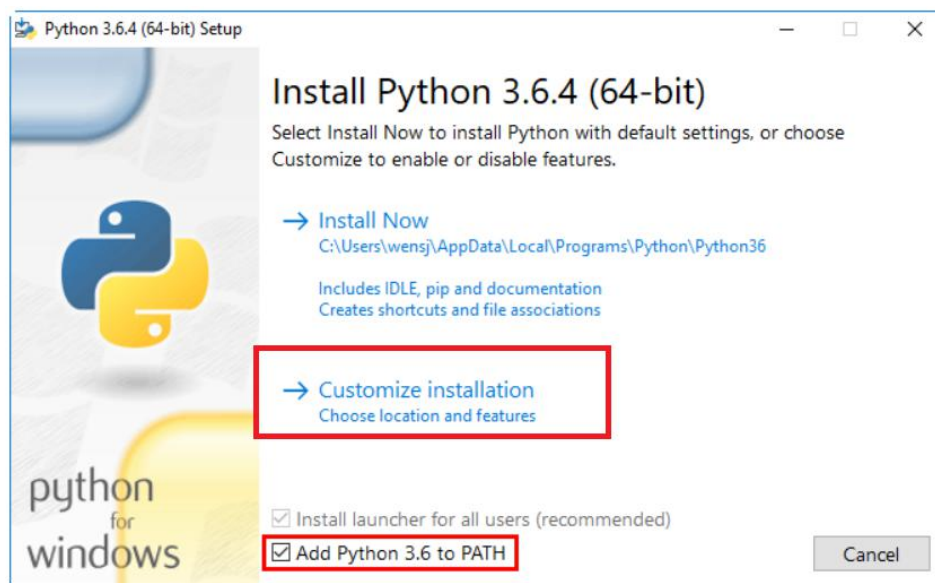
访问 python 官网，以 windows 系统为例。下载 executable installer，x86 表示 32 位，x86-64 表示 64 位。



Python Releases for Windows

- [Latest Python 3 Release - Python 3.7.0](#)
- [Latest Python 2 Release - Python 2.7.15](#)
- [Python 3.7.0 - 2018-06-27](#)
 - Download [Windows x86 web-based installer](#)
 - Download [Windows x86 executable installer](#)
 - Download [Windows x86 embeddable zip file](#)
 - Download [Windows x86-64 web-based installer](#)
 - Download [Windows x86-64 executable installer](#)
 - Download [Windows x86-64 embeddable zip file](#)
 - Download [Windows help file](#)
- [Python 3.6.6 - 2018-06-27](#)

安装 python 3.6。记得勾选添加路径：



快捷键 win+R，输入 cmd 调出命令提示符。输入 python，返回如下则安装成功。

```
C:\Windows\system32\cmd.exe - python
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

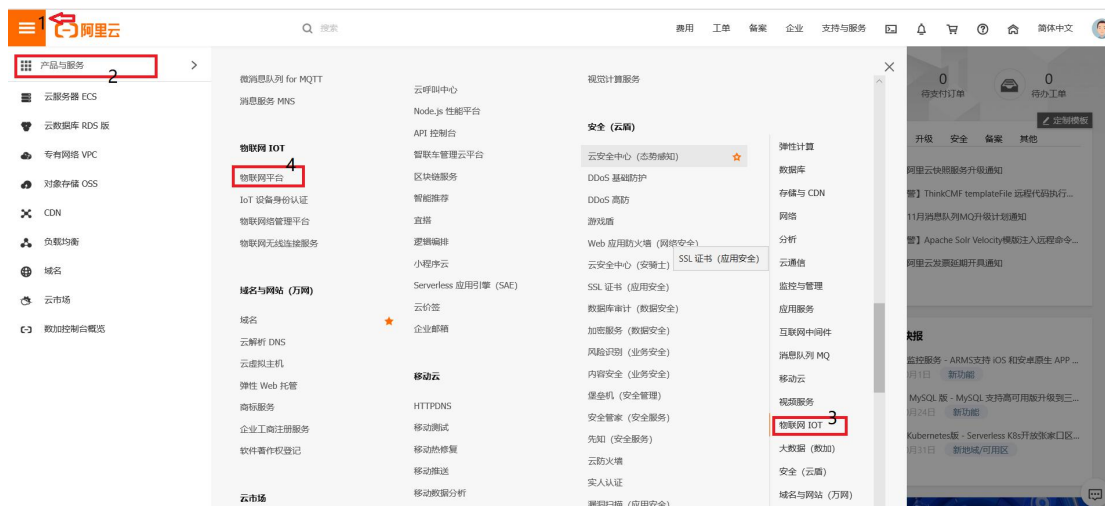
C:\Users\senbiao> python
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 17:54:52) [MSC v.1900 32 bit (Intel)]
on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

➤ 物联网云平台的注册及使用

进入阿里云官网，点击免费注册。注册成功后登录，点击控制台：



开通物联网平台：

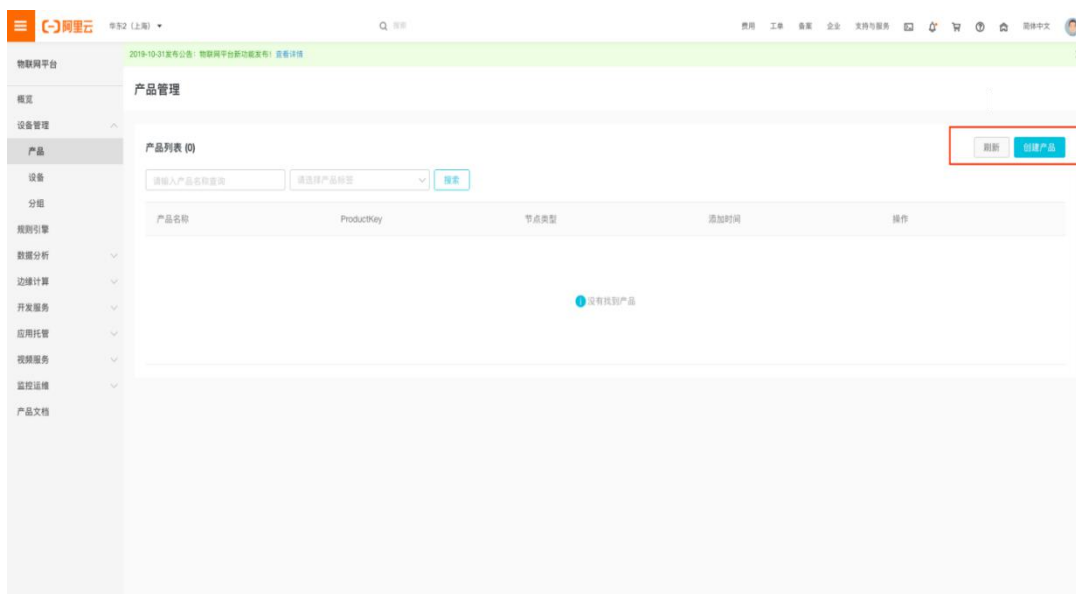


➤ 物联网产品设计实例

以下为设计流程实例：

(1) 产品及设备的创建

打开控制台-->点击设备管理 --> 产品 --> 创建产品



产品名称，所属产品选择“自定义品类”，点击保存。

产品管理 > 创建产品

创建产品 (设备模型)

产品信息 (设备模型)

* 产品名称

* 所属品类

☐ 标准品类 ☒ 自定义品类

* 节点类型

☒ 直连设备 ☐ 网关子设备 ☐ 网关设备

联网与数据

* 联网方式

* 数据格式

认证方式

更多信息

产品描述

创建产品后点击功能定义 --> 自定义功能 --> 添加自定义功能为产品添加功能。

物联网平台

概览

设备管理

产品

设备

分组

规则引擎

数据分析

边缘计算

开发服务

应用托管

视频服务

监控运维

产品文档

2019-10-31发布公告：物联网平台新功能发布！查看详情

产品管理 > 产品详情

Thermometer

ProductKey: atUw7G6ALJ 复制

ProductSecret: ***** 重置 复制

设备数: 1 前往管理

产品信息 Topic类列表 功能定义 服务编排 日志服务 在线调试

功能列表

标准功能

自定义功能

导入物模型

查看物模型

生成设备端代码

添加自定义功能

功能类型	功能名称	标识符	数据类型	数据定义	操作
无自定义功能					

选择功能类型、定义功能名称、标识符、数据类型、步长等，然后点击确认。
此处仅定义温度一个属性。

添加自定义功能

* 功能类型:

属性

服务

事件

* 功能名称:

温度

* 标识符:

temperature

* 数据类型:

int32 (整数型)

* 取值范围:

-30

~

30

* 步长:

1

单位:

请选择单位

* 读写类型:

☒ 读写

☐ 只读

描述:

请输入描述

0/100

确认

取消

点击：设备-->添加设备

物联网平台

2019-10-21发布公告：物联网平台新功能发布！查看详情

设备管理

全部产品

设备总数 0

连接设备 0

当前在线 0

刷新

设备列表

设备列表

DeviceName

请输入DeviceName

请选择设备类型

搜索

☐

DeviceName

设备所属产品

节点类型

状态/应用状态

最后上线时间

操作

没有找到设备

删除

禁用

启用

批量添加

添加设备

产品选择刚刚创建的产品名称。点击确认。



添加设备 ?

特别说明: deviceName可以为空, 当为空时, 阿里云会颁发全局唯一标识符作为deviceName。

* 产品:
Thermometer

DeviceName:
请输入DeviceName

备注名称:
请输入备注名称

确认 取消

点击确认后会弹窗设备证书, 包括公钥、设备名称、设备密钥的信息, 点击一键复制保存起来。



查看设备证书

设备证书用于云端对接入的设备做鉴权认证, 请妥善保管!

ProductKey	a1ufW7G6ALJ	复制
DeviceName	FOP5o96Vd3ltRkSWITII	复制
DeviceSecret	*****	显示

一键复制 关闭

(2) 用 python 模拟设备上传数据

以 windows 为例, 其他系统请参考文档:

https://help.aliyun.com/document_detail/98292.html?spm=a2c4g.11186623.6.6

[45.7ae971bcSx1KQD](#)

1) 打开 VScode 的终端控制台，依次输入以下命令行：

1. 在路径下创建文件夹：mkdir work_dir（默认在 c 盘创建）

以下步骤为创建和激活 python 虚拟环境，避免导入包时发生混乱：

2. cd work_dir

3. python3 -m venv test_env

4. test_env\Scripts\activate.bat

2) 安装 paho 和 linkkit，参看

https://help.aliyun.com/document_detail/98292.html?spm=a2c4g.11186623.6.645.7ae971bcSx1KQD

3) 创建 thermometer.py 文件模拟设备

4) 代码实现设备连接

1. 导入后文需用到的包

```
import paho.mqtt.client as mqtt
import time
import hashlib
import hmac
import random
import json
```

2. 配置端口信息，注意替换成自己的设备信息

```
options = {
    'productKey': "a1ufW7G6ALJ",
    'deviceName': "F0P5o96Vd3ltRkSWlTlI",
    'deviceSecret': "zkg2vN0yY1hRFSk0RPakKbKayDZudb8b",
    'regionId': 'cn-shanghai'
}

HOST = options['productKey'] + '.iot-as-mqtt.' + options['regionId'] + '.aliyuncs.com'
PORT = 1883
PUB_TOPIC = "/sys/" + options['productKey'] + "/" + options['deviceName'] + "/thing/event/property/post";
```

3. 编写订阅功能

```
# The callback for when the client receives a CONNACK response from the server.
def on_connect(client, userdata, flags, rc):
    print("Connected with result code "+str(rc))

# The callback for when a PUBLISH message is received from the server.
def on_message(client, userdata, msg):
    print(msg.topic+" "+str(msg.payload))
```

4. 使用 hmacsha1 签名认证算法:

```
def hmacsha1(key, msg):  
    return hmac.new(key.encode(), msg.encode(), hashlib.sha1).hexdigest()
```

5. 获取阿里云物联网客户端信息

```
def getAliyunIoTClient():  
    timestamp = str(int(time.time()))  
    # CLIENT_ID = "paho.py|securemode=3,signmethod=hmacsha1,timestamp="+timestamp+"|"  
    CLIENT_ID = 'aaa'  
    CONTENT_STR_FORMAT = "clientIdpaho.pydeviceName"+options['deviceName']+"productKey"+options['productKey']+"timestamp"+timestamp  
    # set username/password.  
    USER_NAME = options['deviceName'] + "&" + options['productKey']  
    PWD = hmacsha1(options['deviceSecret'], CONTENT_STR_FORMAT)  
    client = mqtt.Client (client_id=CLIENT_ID, clean_session=False)  
    # client.username_pw_set (USER_NAME, PWD)  
    return client
```

CLIENT_ID 由自己来定义。

6. 定义主函数

```
if __name__ == '__main__':  
  
    client = getAliyunIoTClient()  
    client.on_connect = on_connect  
    client.on_message = on_message  
  
    client.connect(HOST, 1883, 300)
```

调用 publish 方法将数据上传到平台。保持监听状态。

```
print('send data to iot server: ' + str (payload_json))  
  
client.publish (PUB_TOPIC, payload=str (payload_json), qos=1)  
  
client.loop_forever()
```

主函数内部还需要设置负载 (payload) , 即订阅者具体接收的内容:

本例中上传的数据为-30 到 30 之间的温度数据, 标识符为: temperature。

```
payload_json = {  
    'id': int(time.time()),  
    'params': {  
        'temperature': random.randint(-30, 30),  
    },  
    'method': "thing.event.property.post"  
}
```

5) 运行

在 VSCode 控制台中输入指令: python thermometer.py

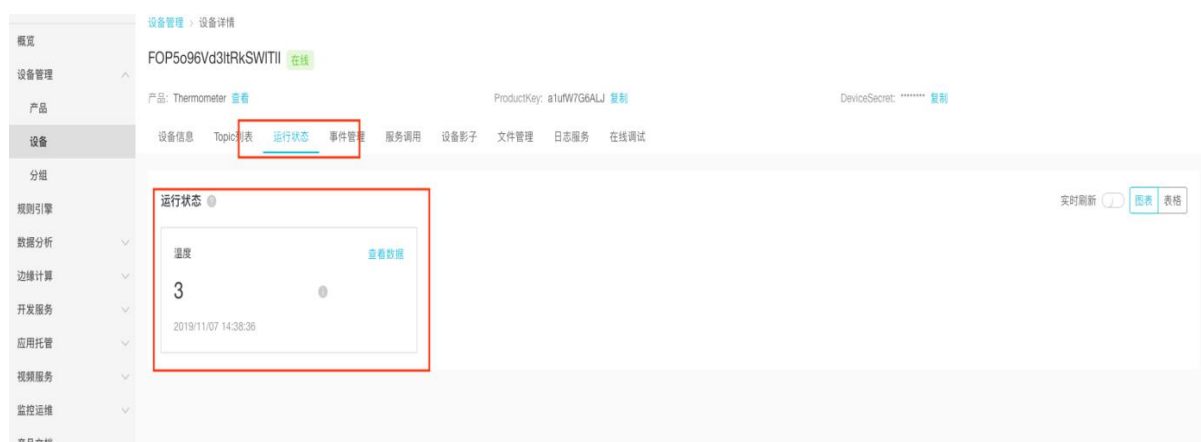
运行代码后，控制台输出信息：

```
send data to iot server: {'id': 1573108716, 'params': {'temperature': 3}, 'method': 'thing.event.property.post'}  
Connected with result code 0
```

然后在设备管理平台中点击查看设备。



在运行状态中即可看到当前 python 模拟设备上传的运行状态。



5. 实验任务

参考上述实验步骤中的物联网产品设计实例实现一个至少包含 3 个属性, 且同时连接至少三个设备 (建议按序取名, 如 Device01, Device02) 的物联网应用, 实现数据通信。

例如: 设计一个含有温度 (摄氏度)、湿度 (百分比)、功率 (双精度浮点型) 三个属性的产品, 并同时连接三个设备。

➤ 评分标准

根据所设计的应用的属性数、连接设备数、以及功能的丰富程度来评分。

6. 思考

- a) MQTT 协议构建于什么协议之上？有何特点？
- b) 代码中导入的 hmac 包作用是？
- c) 如何利用阿里云平台实现云端往设备发送实时数据？