

# 科学观察助手

## 功能需求：

如果我们想知道植物的生长情况与土壤湿度的关系，我们应该如何进行探究呢？中小学有很多关于观察植物的探究小实验，但是学生只能够通过观察植物每天的生长情况，以文字的形式记录在本子上，无法提供较为具体的监测数据。而现在，借助虚谷号、摄像头和土壤湿度传感器，我们能够将植物的生长情况进行实时记录，利用SIOT平台对数据进行可视化分析，让我们能够更加直观地看到植物在不同时间下的生长情况，有助于我们分析并了解植物的习性特点。

## 代码功能：

将拍照以及检测植物土壤湿度的相关数据，发送到物联网平台，具体实现过程为：利用python代码通过摄像头拍摄植物照片，读取传感器数值并写入图片，编码为Base64数据格式后上传到物联网平台SIoT。

代码编写：林淼焱，谢作如

## 器材准备

- 普通的USB摄像头\*1
- 土壤湿度传感器\*1
- Arduino扩展板（可选）

## 数据采集端

这段代码的作用，是将摄像头的画面和传感器数值合并起来。

In [ ]:

```
import cv2,base64,json,time,re,siot
from xugu import Pin # 从 xugu 库中导入 Pin类

p = Pin("A0", Pin.ANALOG) # 初始化 A0 引脚,设置为输入模式
filepath = r"sending.jpg" #即将发送的图片, 路径中不能带有中文
CLIENT_ID = "" #在SIoT上, CLIENT_ID可以留空
SERVER = "127.0.0.1" #MQTT服务器IP
IOT_pubTopic = 'DFRobot/linmy' #"topic"为"项目名称/设备名称"
IOT_UserName='scope' #用户名
IOT_PassWord='scope' #密码
siot.init(CLIENT_ID, SERVER, user=IOT_UserName, password=IOT_PassWord)

#设置发送反馈, 数据量过大, 因此不查看具体的信息
def sub_cb(client, userdata, msg):
    #print("\nTopic:" + str(msg.topic) + " Message:" + str(msg.payload))
    print("发送成功")

#摄像头拍照
def get_pic():
    cap=cv2.VideoCapture(0)
    sucess,img=cap.read()
    cv2.imwrite(filepath,img)
    cap.release()

#利用函数作画
def draw():
    img = cv2.imread(filepath)
    img = cv2.resize(img, (320,240), interpolation=cv2.INTER_AREA)
    high, width, _ = img.shape
    cv2.putText(img, 'Humidity', (int(width*3/4-50),int(high*3/4-5)), cv2.FONT_HERSHEY_SIMPLEX, 1, (0,0,255))
    cv2.rectangle(img,(int(width*3/4-40),int(high*3/4)), (int(width*3/4+100),int(high*3/4)), (0,0,255))
    cv2.putText(img, ('VALUE:' + str(value)), (int(width*3/4-40),int(high*3/4+20)), cv2.FONT_HERSHEY_SIMPLEX, 1, (0,0,255))
    #cv2.imshow('img',img)
    cv2.imwrite(filepath,img)

#图片→base64编码→json
def encode(value):
    with open(filepath, "rb") as f:
        base64_byte = base64.b64encode(f.read())
    #读取时间
    now = time.strftime("%Y-%m-%d-%H-%M-%S", time.localtime(time.time()))
    dictdata={"value":value,"time":now,"base64":str(base64_byte)}
    jsondata = json.dumps(dictdata)
    return jsondata

#主程序: 向服务器发送信息
if __name__ == '__main__':
    siot.connect()
    siot.set_callback(sub_cb)
    siot.getsubscribe(IOT_pubTopic)
    siot.loop()
    try:
        while True:
            get_pic()
            value= p.read_analog()
            draw()
            jsondata = encode(value)
            siot.publish(IOT_pubTopic,jsondata)
            time.sleep(360) #隔多少秒发送一次
```

```
except:
    siot.stop()
    print("disconnect seccused")
```

## ## 数据接收端

这段代码运行在普通的电脑上。

代码的作用是订阅（接收）MQTT服务器发回的消息，然后解码为图片。

**\*\*注意\*\***：SERVER的ip地址要改为虚谷号的ip。

In [ ]:

```
import json,base64,re
import siot

filepath = r'static\images' #保存图片的路径
filename = r"receive"       #图片名称的前缀
SERVER = "127.0.0.1"        #MQTT服务器IP（如虚谷号的IP地址）
CLIENT_ID = " "            #在SIoT上，CLIENT_ID可以留空
IOT_pubTopic = 'DFRobot/linmy' #“topic”为“项目名称/设备名称”
IOT_UserName='scope'        #用户名
IOT_PassWord='scope'        #密码
siot.init(CLIENT_ID, SERVER, user=IOT_UserName, password=IOT_PassWord)

def sub_cb(client, userdata, msg):
    print("\nTopic:" + str(msg.topic) + " Message:" + str(msg.payload))
    #如果数据格式是监控数据,则接受服务器信息,并储存到数据库中
    if len(str(msg.payload))>500:
        #将JSON转化为字符串
        jsontdata=msg.payload
        dictdata=json.loads(jsontdata)
        base64_str=dictdata["base64"]
        #将字符串格式的"base64code"转化为bytes格式
        base64_byte=base64_str.encode(encoding="utf-8")[-1]
        imgdata = base64.b64decode(base64_byte)
        #本地存取图片
        file = open(filepath+ '/' + filename + str(dictdata["time"]) + '.jpg','wb')
        file.write(imgdata)
        file.close()

if __name__ == "__main__":
    siot.connect()
    siot.subscribe(IOT_pubTopic, sub_cb)
    siot.loop()
    try:
        while True:
            pass
    except:
        siot.stop()
        print("disconnect seccused")
```

In [ ]:

