

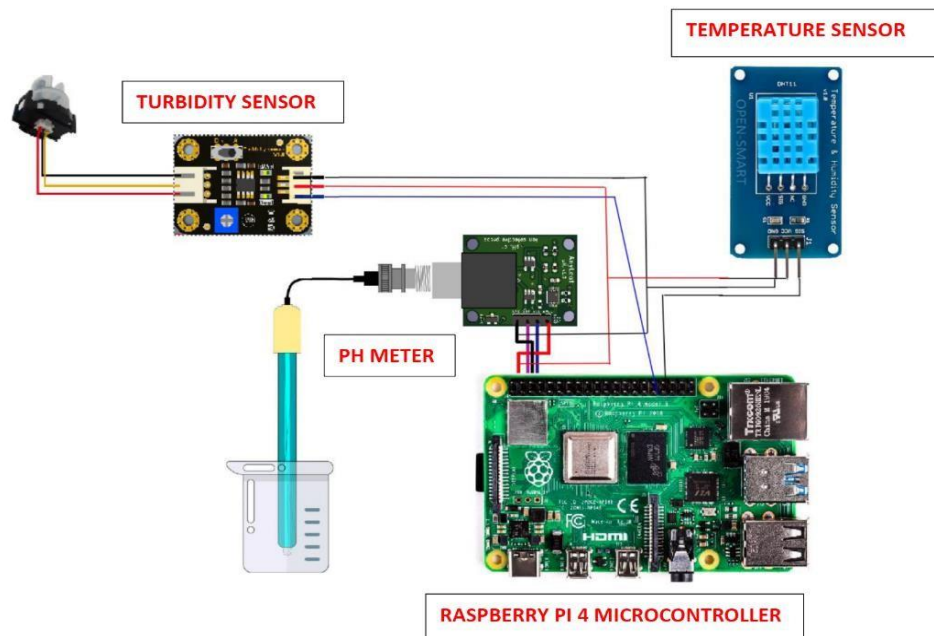
SPRINT 2

IBM CLOUD

The screenshot displays the IBM Watson IoT Platform interface. On the left, a code editor shows a Python script named `Test_python_3.7.4.py` that generates random data for pH, turbidity, and temperature, and publishes it to the IoT platform. The script includes a `while True` loop and a `def myOnPublishCallback()` function. The right side of the interface shows the 'Recent events' section, which lists the data being published. Below the events, a table shows the details of the events, including the Event ID, Value, and the device name 'Micro_controller_2'.

Event	Value
demo	("pH":12,"turbid":93,"temp":87)
demo	("pH":7,"turbid":873,"temp":94)
demo	("pH":3,"turbid":204,"temp":19)
demo	("pH":11,"turbid":304,"temp":77)
demo	("pH":13,"turbid":16,"temp":50)

CIRCUIT DAIGRAM



```
Untitled - Notepad
File Edit Format View Help
import ibmiotf.application
import ibmiotf.device
import time
import random
import sys
from twilio.rest import Client
import keys
Client = Client(keys.account_sid, keys.auth_token)

organization = "8n6m92"
deviceType = "Node"
deviceId = "868283"
authMethod = "use-token-auth"
authToken = "86828329"

pH = random.randint(1, 14)
turbidity = random.randint(1, 100)
temperature = random.randint(0, 100)

def myCommandCallback(cmd):
    print("Command Received: %s" % cmd.data['command'])
    print(cmd)

try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod,
                    "auth-token": authToken}
    deviceCli = ibmiotf.device.Client(deviceOptions)

except Exception as e:
    print("caught exception connecting device: %s" % str(e))
    sys.exit()

deviceCli.connect()

while True:
    pH = random.randint(1, 14)
    turbidity = random.randint(1, 100)
    temperature = random.randint(0, 100)
```

```
Untitled - Notepad
File Edit Format View Help
def myCommandCallback(cmd):
    print("Command Received: %s" % cmd.data['command'])
    print(cmd)

try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod,
                    "auth-token": authToken}
    deviceCli = ibmiotf.device.Client(deviceOptions)

except Exception as e:
    print("caught exception connecting device: %s" % str(e))
    sys.exit()

deviceCli.connect()

while True:
    pH = random.randint(1, 14)
    turbidity = random.randint(1, 100)
    temperature = random.randint(0, 100)

    data = {'pH': pH, 'turbid': turbidity, 'temp': temperature}
    def SMS():
        message = Client.messages.create(
            body="ALERT!! THE WATER QUALITY IS DEGRADED",
            from_=keys.twilio_number,
            to = keys.target_number)
        print(message.body)

    if temperature>70 or pH<6 or turbidity>500:
        SMS()

    def myOnPublishCallback():
        print("Published pH= %s" % pH, "Turbidity:%s" % turbidity, "Temperature:%s" % temperature)

    success = deviceCli.publishEvent("demo", "json", data, qos=0, on_publish=myOnPublishCallback)
    if not success:
        print("Not Connected to ibmiot")
        time.sleep(5)
    deviceCli.commandCallback = myCommandCallback

deviceCli.disconnect()
```