

## DEVELOPING A PYTHON SCRIPT

Date	20 OCTOBER 2022
Team ID	PNT2022TMID52857
Project Name	IoT-Based Safety Gadget for Child Safety Monitoring and Notification

### Code for python script

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random

#Provide your IBM Watson Device Credentials
organization = "awb990"

deviceType = "NodeMCU"
deviceId = "12345"
authMethod = "token"
authToken = "12345678"

# Initialize GPIO

def myCommandCallback(cmd):

    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print("led is on")
    elif status== "lightoff":
        print("led is off")
    else :
```

```
print ("please send proper 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()
```

# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10 times

```
deviceCli.connect()
```

while True:

```
    #Get Sensor Data from DHT11
```

```
    temp=random.randint(90,100)
```

```
    Humid=random.randint(60,100)
```

```
    data = { 'temp' : temp, 'Humid': Humid }#print data
```

```
    def myOnPublishCallback():
```

```
        print ("Published Temperature = %s C" % temp, "Humidity = %s %" % Humid,
"to IBM Watson")
```

```
        success      =      deviceCli.publishEvent("IoTSensor",  "json",  data,  qos=0,
on_publish=myOnPublishCallback)
```

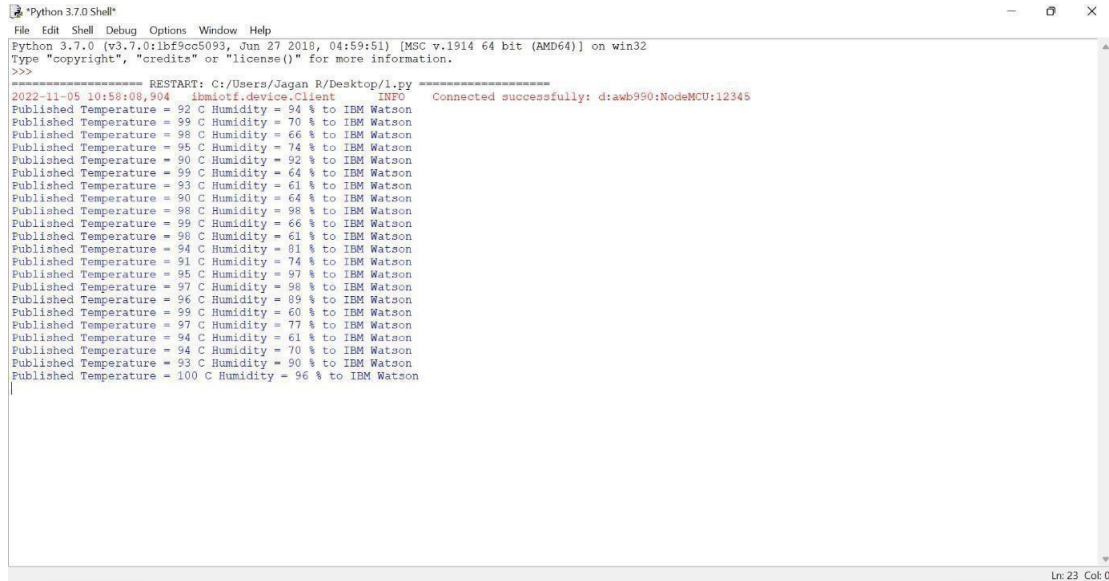
```
        if not success:
```

```
            print("Not connected to IoT")time.sleep(1)
```

```
deviceCli.commandCallback = myCommandCallback
```

# Disconnect the device and application from the cloud  
deviceCli.disconnect()

## OUTPUT:



```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2019, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Jagan R/Desktop/l.py =====
2022-11-05 10:58:08,904 ibmiotf.device.Client INFO Connected successfully: d:awb990:NodeMCU:12345
Published Temperature = 92 C Humidity = 94 % to IBM Watson
Published Temperature = 99 C Humidity = 70 % to IBM Watson
Published Temperature = 98 C Humidity = 66 % to IBM Watson
Published Temperature = 95 C Humidity = 74 % to IBM Watson
Published Temperature = 90 C Humidity = 92 % to IBM Watson
Published Temperature = 99 C Humidity = 64 % to IBM Watson
Published Temperature = 93 C Humidity = 61 % to IBM Watson
Published Temperature = 90 C Humidity = 64 % to IBM Watson
Published Temperature = 98 C Humidity = 98 % to IBM Watson
Published Temperature = 99 C Humidity = 66 % to IBM Watson
Published Temperature = 96 C Humidity = 61 % to IBM Watson
Published Temperature = 94 C Humidity = 81 % to IBM Watson
Published Temperature = 91 C Humidity = 74 % to IBM Watson
Published Temperature = 95 C Humidity = 97 % to IBM Watson
Published Temperature = 97 C Humidity = 98 % to IBM Watson
Published Temperature = 96 C Humidity = 89 % to IBM Watson
Published Temperature = 99 C Humidity = 60 % to IBM Watson
Published Temperature = 97 C Humidity = 77 % to IBM Watson
Published Temperature = 94 C Humidity = 61 % to IBM Watson
Published Temperature = 94 C Humidity = 70 % to IBM Watson
Published Temperature = 93 C Humidity = 90 % to IBM Watson
Published Temperature = 100 C Humidity = 96 % to IBM Watson
>>>
```

Figure 7.1 Output window

Figure 7.1 shows the output of the above code which executed in the Python software

## 7.1 Data publish to IBM cloud

WOKWI Simulation using ESP32 and Ultrasonic Sensor:

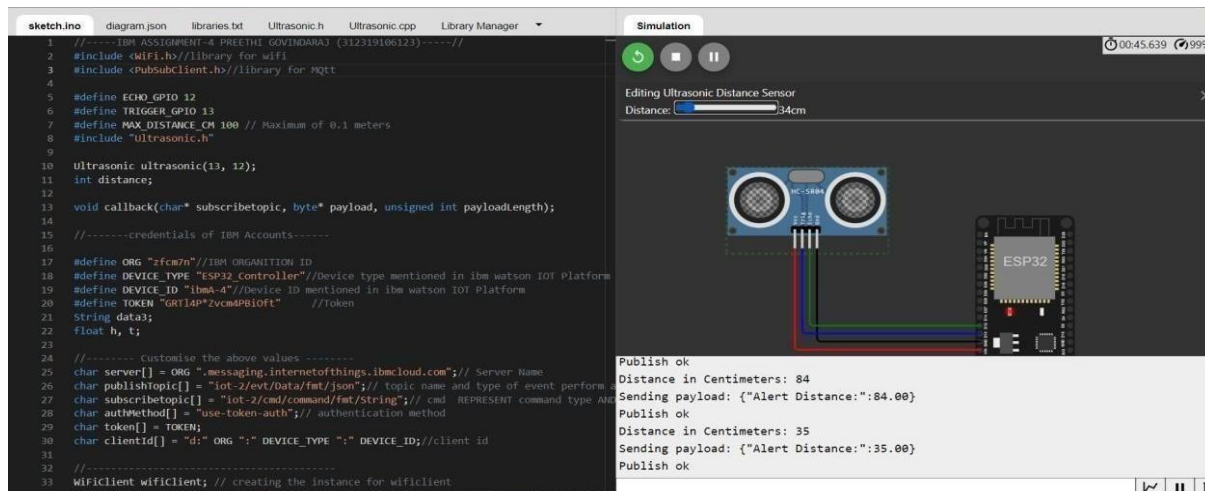
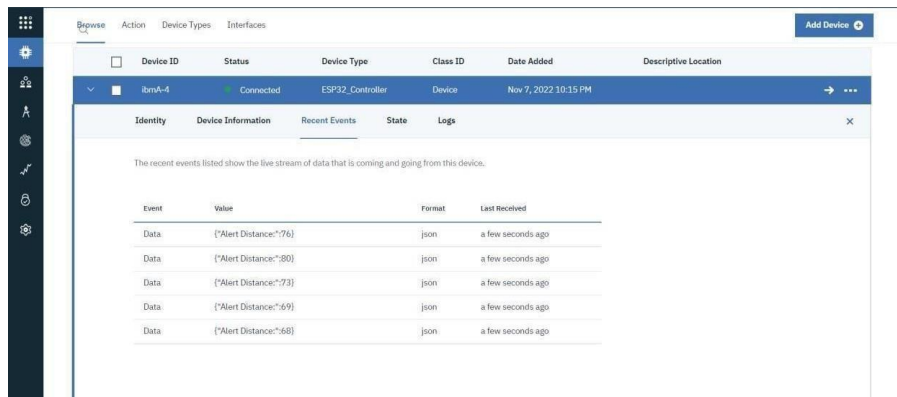


Figure 7.2 Wokwi simulation

Figure 7.2 shows the wokwi simulation of the ultrasonic sensor and wifi module

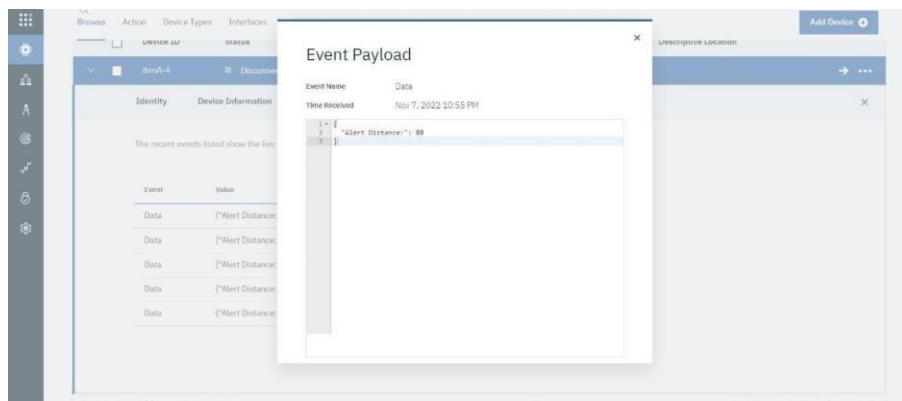


The screenshot shows the IBM Watson IoT Platform interface. A table lists recent events for a device named 'ibm4-4'. The table has columns for Event, Value, Format, and Last Received. The events are all 'Data' type with a value of '["Alert Distance":76]', '["Alert Distance":80]', '["Alert Distance":73]', '["Alert Distance":69]', and '["Alert Distance":68]'. The format is 'json' and the last received time is 'a few seconds ago'.

Event	Value	Format	Last Received
Data	["Alert Distance":76]	json	a few seconds ago
Data	["Alert Distance":80]	json	a few seconds ago
Data	["Alert Distance":73]	json	a few seconds ago
Data	["Alert Distance":69]	json	a few seconds ago
Data	["Alert Distance":68]	json	a few seconds ago

Figure 7.3 IBM Watson

Figure 7.3 shows the IBM Watson platform device-event log which shows different setof data.



The screenshot shows the IBM Watson IoT Platform interface with an 'Event Payload' dialog box open. The dialog box displays the event name 'Data', the time received 'Nov 7, 2022 10:55 PM', and the event payload '["Alert Distance":1 80]'. The background shows the same table of events as Figure 7.3.

Event Name	Data
Time Received	Nov 7, 2022 10:55 PM
Event Payload	["Alert Distance":1 80]

Figure 7.4 Event payload

Figure 7.4 shows the event payload that shows the alert system.