Write-up	Correctness of Program	Documentation of Program	Viva	Timely Completion	Total	Dated Sign of Subject Teacher
2	2	2	2	2	10	

Assignment No. 05 Date of Performance: ……

Title: Introduction to MQTT/ CoAP and sending sensor data to cloud using Raspberry-Pi/Beagle board/Arduino.

Objective: To provide knowledge for IoT related protocols such as MQTT / CoAP etc.

Theory:

- MQTT (Message Queue Telemetry Transport) is a lightweight messaging protocol which is ideal for communication of IoT connected devices.
- MQTT has three components: broker, publisher, and subscriber. A broker is an intermediary entity that handles the communication going on between devices. A publisher is a device that sends messages. A subscriber listens to the messages sent by the publisher.

Client –

- A program or device that uses MQTT.
- A Client always establishes the Network Connection to the Server.
- It can Publish Application Messages that other Clients might be interested in.
- Subscribe to request Application Messages that it is interested in receiving.

Server –

- A program or device that acts as an intermediary between Clients which publish Application Messages and Clients which have made Subscriptions. A Server
- Accepts Network Connections from Clients.
- Accepts Application Messages published by Clients.

Publish/Subscribe - The MQTT protocol is based on the principle of publishing messages and subscribing to topics, or "pub/sub". Multiple clients connect to a broker and subscribe to topics that they are interested in. Clients also connect to the broker and publish messages to topics. Many clients may subscribe to the same topics and do with the information as they please. The broker and MQTT act as a simple, common interface for everything to connect to. This means that you if you have clients that dump subscribed messages to a database, to Twitter, Cosm or even a simple text file, then it becomes very simple to add new sensors or other data input to a database, Twitter or so on. Temperature upload over MQTT using Raspberry Pi and DHT22 sensor and sending data to ThingsBoard platform.

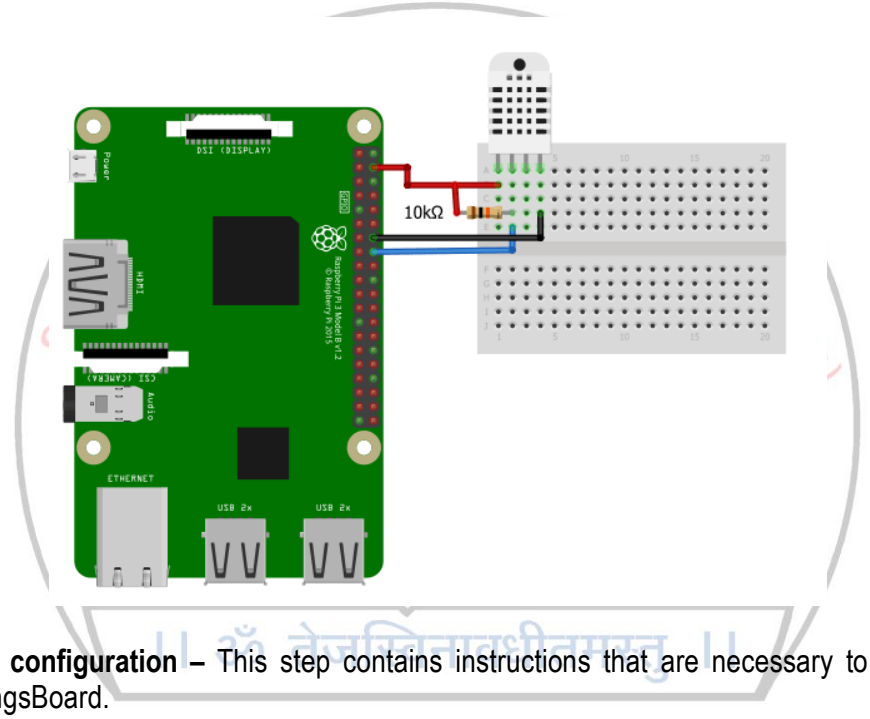
ThingsBoard –

- It is an open-source server-side platform that allows you to monitor and control IoT devices. It is free for both personal and commercial usage and you can deploy it anywhere.
- This sample application performs collection of temperature and humidity values produced by DHT22 sensor and further visualization on the real-time web dashboard. Collected data is pushed via MQTT to ThingsBoard server for storage and visualization.
- The DHT22 sensor is connected to Raspberry Pi. Raspberry Pi offers a complete and self-contained Wi-Fi networking solution. Raspberry Pi push data to ThingsBoard server via MQTT

protocol by using paho mqtt python library. Data is visualized using built-in customizable dashboard. The application that is running on Raspberry Pi is written in Python.

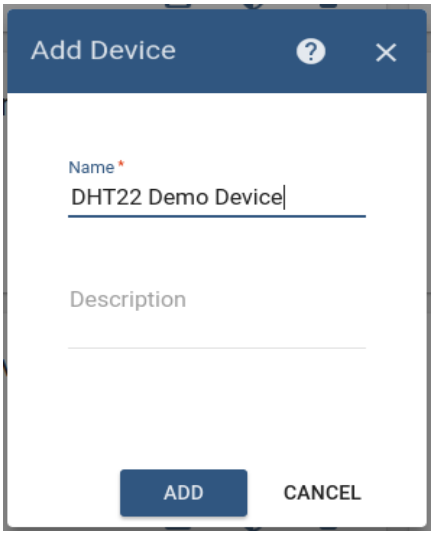
Circuit Diagram and Explanation – Make the connections of the DHT22 with the Raspberry pi as described below:

DHT22	**Raspberry pi**
VCC	5v
Data pin	Connect to GPIO 23 and also connect to 5V through 10K resistor
GND	GND

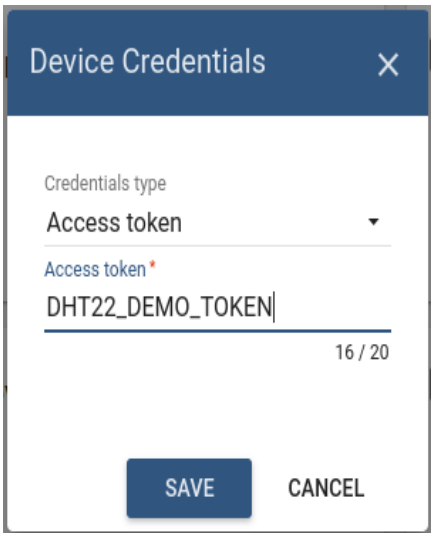


ThingsBoard configuration – This step contains instructions that are necessary to connect your device to ThingsBoard.

- Open ThingsBoard Web UI (http://localhost:8080) in browser and login as tenant administrator
- login: tenant@thingsboard.org
- password: tenant
- Goto “Devices” section. Click “+” button and create device with name “DHT22 Demo Device”.



Once device created, open its details and click “Manage credentials”. Copy auto-generated access token from the “Access token” field. Please save this device token. It will be referred to later as **$ACCESS_TOKEN**.



Click “Copy Device ID” in device details to copy your device id to the clipboard. Paste your device id to some place, this value will be used in further steps.

Download the dashboard file using this link. Use import/export instructions to import the dashboard to your ThingsBoard instance.

MQTT library installation – The following command will install MQTT Python library:

```
sudo pip install paho-mqtt
```

Adafruit DHT library installation –

Install python-dev package:

```
sudo apt-get install python-dev
```

Downloading and install the Adafruit DHT library:

```
git clone https://github.com/adafruit/Adafruit_Python_DHT.git
cd Adafruit_Python_DHT
sudo python setup.py install
```

Pseudo Code –

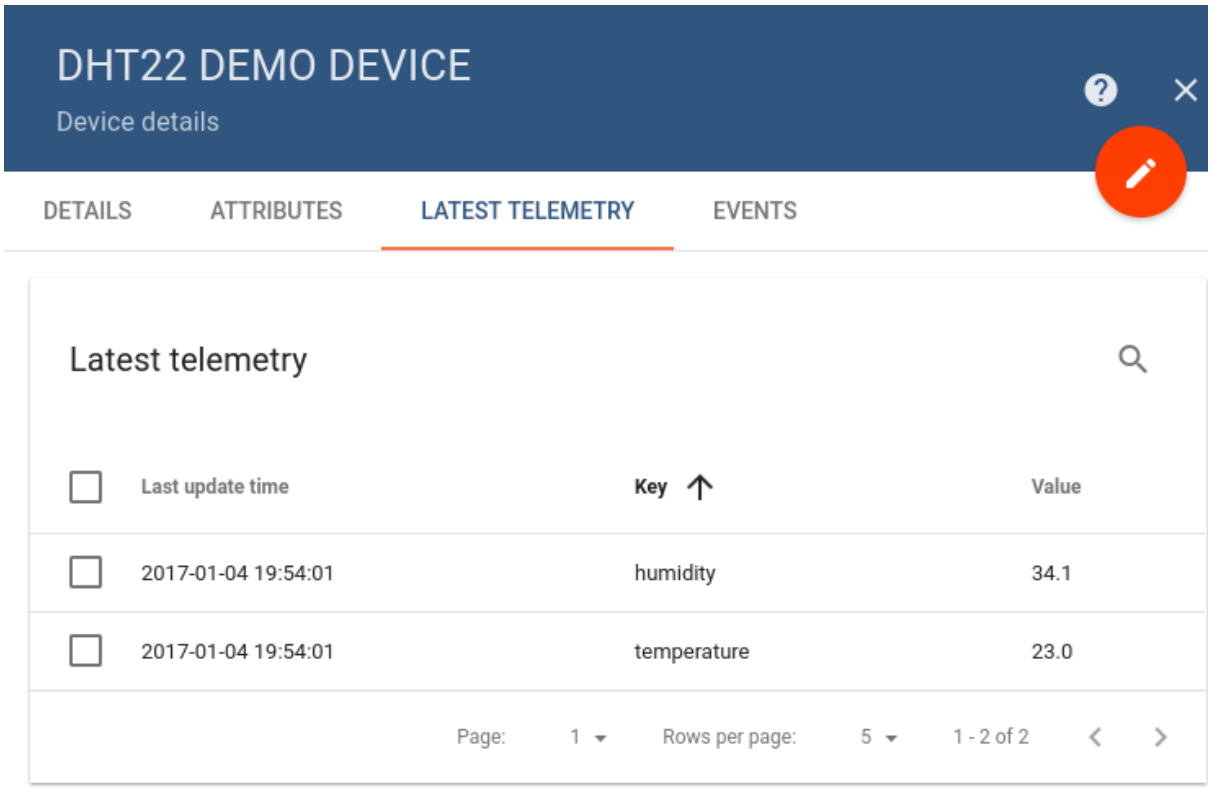
```
import os
import time
import sys
import Adafruit_DHT as dht
import paho.mqtt.client as mqtt
import json

THINGSBOARD_HOST='demo.thingsboard.io'
ACCESS_TOKEN='DHT22_DEMO_TOKEN'

# Data capture and upload interval in seconds. Less interval will eventually hang the DHT22.
INTERVAL=2

sensor_data={'temperature':0,'humidity':0}

next_reading=time.time()

client=mqtt.Client()

# Set access token
client.username_pw_set(ACCESS_TOKEN)

# Connect to ThingsBoard using default MQTT port and 60 seconds keepalive interval
client.connect(THINGSBOARD_HOST,1883,60)

client.loop_start()

try:
while True:
humidity,temperature=dht.read_retry(dht.DHT22,4)
humidity=round(humidity,2)
temperature=round(temperature,2)
print(u"Temperature: {:g}\u00b0C, Humidity: {:g}%".format(temperature,humidity))
sensor_data['temperature']=temperature
sensor_data['humidity']=humidity

# Sending humidity and temperature data to ThingsBoard
client.publish('v1/devices/me/telemetry',json.dumps(sensor_data),1)

next_reading+=INTERVAL
sleep_time=next_reading-time.time()
ifsleep_time>0:
time.sleep(sleep_time)
except KeyboardInterrupt:
pass

client.loop_stop()
client.disconnect()
```

Running the application – This simple command will launch the application:

```
python mqtt-dht22.py
```

Data visualization –

- Finally, open ThingsBoard Web UI. You can access this dashboard by logging in as a tenant administrator.
- Go to **"Devices"** section and locate **"DHT22 Demo Device"**, open device details and switch to **"Latest telemetry"** tab. If all is configured correctly you should be able to see latest values of *"temperature"* and *"humidity"* in the table.



Outcome: - Thus, we have studied how to interface cloud for uploading the environment properties measured with the help of DHT22 sensor and Raspberry Pi 3 using MQTT/CoAP.
