

**ESIEE**

**DRIO 4302B IoT Prototyping**

**Sahar Hosseini, Quoc Trung Pham**

April 2019

**IoT Project Report**

We open the virtual machine instant – contiki in VMware, then login, we plugged the launch pad on vmware; check the name and connectivity the sensor, we got this result:

```

user@instant-contiki: ~/contiki/tools
File Edit View Search Terminal Help
created a new RPL dag
Server IPv6 addresses:
fd00::212:4b00:7b1:6c01
fe80::212:4b00:7b1:6c01
Starting Contiki-3.x-3343-gbc2e445
With DriverLib v0.47020
TI CC2650 LaunchPad
IEEE 802.15.4: Yes, Sub-GHz: No, BLE: Yes, Prop: Yes
Net: sicslowpan
MAC: CSMA
RDC: ContikiMAC, Channel Check Interval: 16 ticks
RF: Channel 22
PAN_ID: eeee
Link layer addr: 00:12:4b:00:07:b1:6c:01
Node ID: 27649
RPL-Border router started
*** Address:fd00::1 => fd00:0000:0000:0000
Got configuration message of type P
Setting prefix fd00::
created a new RPL dag
Server IPv6 addresses:
fd00::212:4b00:7b1:6c01
fe80::212:4b00:7b1:6c01

```

We check the edge router address with firefox as belows:

```

user@instant-contiki:~/contiki/tools$
ContikiRPL - Mozilla Firefox
ContikiRPL
[fd00::212:4b00:7b1:6c01]
Neighbors
fe80::212:4b00:6a0:bd89
Routes
fd00::212:4b00:6a0:bd89/128 (via fe80::212:4b00:6a0:bd89) 17275

```

In order to run the MQTT broker first we need to stop the broker in vm machine so we did it as below:

**Note:** VM machine Ip address 147.215.189.15

```

user@instant-contiki:~/contiki/tools$ sudo service mosquitto stop
[sudo] password for user:
mosquitto stop/waiting
user@instant-contiki:~/contiki/tools$ 6tunnel -6 1883 147.215.189.15 1883

```

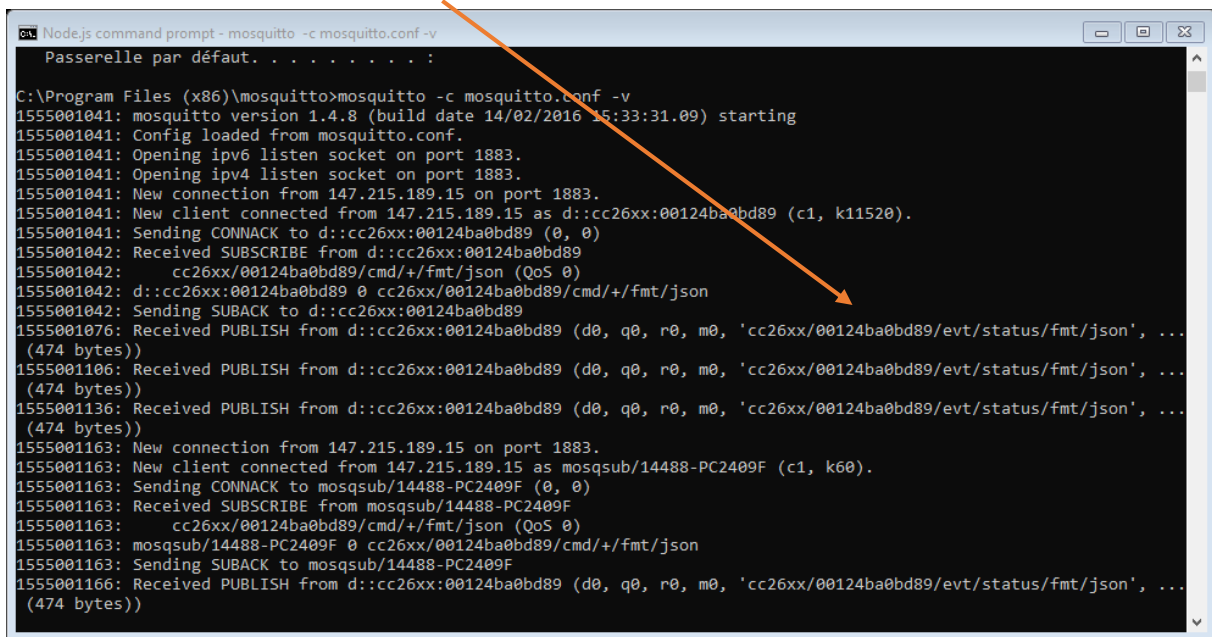
Then we run the MQTT broker on host machine to get the topic name of sensor with this command

```
Mosquitto -c mosquitto.conf -v
```

So we run the broker and we press the button of sensor to publish some information on broker

We have seen this result on broker

**Note:** topic name includes the MAC address of sensor

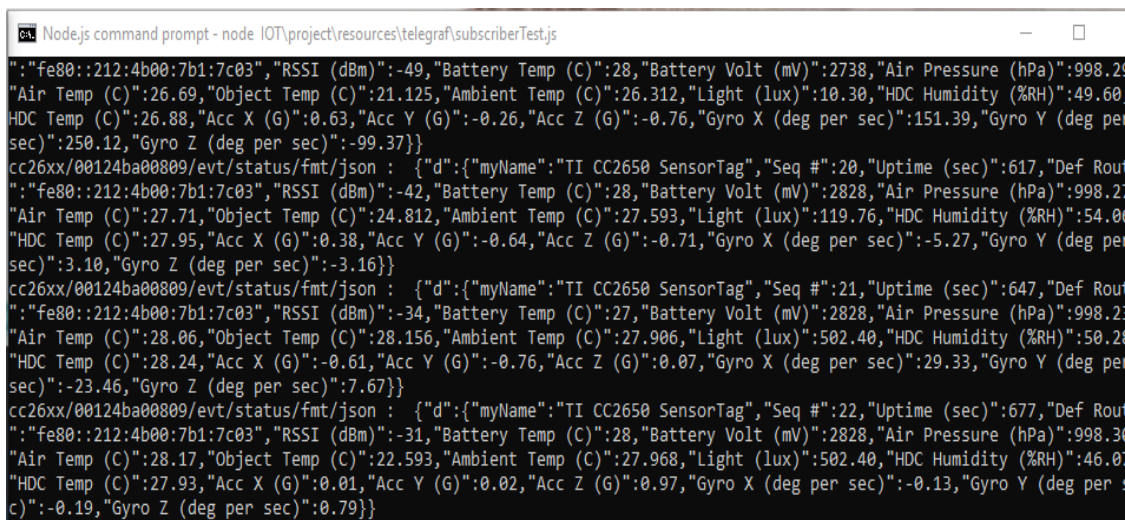


```
Node.js command prompt - mosquitto -c mosquitto.conf -v
Passerelle par défaut. . . . . :

C:\Program Files (x86)\mosquitto>mosquitto -c mosquitto.conf -v
1555001041: mosquitto version 1.4.8 (build date 14/02/2016 15:33:31.09) starting
1555001041: Config loaded from mosquitto.conf.
1555001041: Opening ipv6 listen socket on port 1883.
1555001041: Opening ipv4 listen socket on port 1883.
1555001041: New connection from 147.215.189.15 on port 1883.
1555001041: New client connected from 147.215.189.15 as d::cc26xx:00124ba0bd89 (c1, k11520).
1555001041: Sending CONNACK to d::cc26xx:00124ba0bd89 (0, 0)
1555001042: Received SUBSCRIBE from d::cc26xx:00124ba0bd89
1555001042: cc26xx/00124ba0bd89/cmd/+/fmt/json (QoS 0)
1555001042: d::cc26xx:00124ba0bd89 0 cc26xx/00124ba0bd89/cmd/+/fmt/json
1555001042: Sending SUBACK to d::cc26xx:00124ba0bd89
1555001076: Received PUBLISH from d::cc26xx:00124ba0bd89 (d0, q0, r0, m0, 'cc26xx/00124ba0bd89/evt/status/fmt/json', ...
(474 bytes))
1555001106: Received PUBLISH from d::cc26xx:00124ba0bd89 (d0, q0, r0, m0, 'cc26xx/00124ba0bd89/evt/status/fmt/json', ...
(474 bytes))
1555001136: Received PUBLISH from d::cc26xx:00124ba0bd89 (d0, q0, r0, m0, 'cc26xx/00124ba0bd89/evt/status/fmt/json', ...
(474 bytes))
1555001163: New connection from 147.215.189.15 on port 1883.
1555001163: New client connected from 147.215.189.15 as mosqsub/14488-PC2409F (c1, k60).
1555001163: Sending CONNACK to mosqsub/14488-PC2409F (0, 0)
1555001163: Received SUBSCRIBE from mosqsub/14488-PC2409F
1555001163: cc26xx/00124ba0bd89/cmd/+/fmt/json (QoS 0)
1555001163: mosqsub/14488-PC2409F 0 cc26xx/00124ba0bd89/cmd/+/fmt/json
1555001163: Sending SUBACK to mosqsub/14488-PC2409F
1555001166: Received PUBLISH from d::cc26xx:00124ba0bd89 (d0, q0, r0, m0, 'cc26xx/00124ba0bd89/evt/status/fmt/json', ...
(474 bytes))
```

First by run this command we get the data from sensor

```
Mosquitto_sub -h 147.215.189.15 -p 1883 -t 'cc26xx/00124ba00809/evt/status/fmt/json'
```



```
Node.js command prompt - node IOT\project\resources\telegraf\subscriberTest.js

[{"fe80::212:4b00:7b1:7c03","RSSI (dBm)": -49,"Battery Temp (C)": 28,"Battery Volt (mV)": 2738,"Air Pressure (hPa)": 998.25,"Air Temp (C)": 26.69,"Object Temp (C)": 21.125,"Ambient Temp (C)": 26.312,"Light (lux)": 10.30,"HDC Humidity (%RH)": 49.60,"HDC Temp (C)": 26.88,"Acc X (G)": 0.63,"Acc Y (G)": -0.26,"Acc Z (G)": -0.76,"Gyro X (deg per sec)": 151.39,"Gyro Y (deg per sec)": 250.12,"Gyro Z (deg per sec)": -99.37}, {"d":{"myName": "TI CC2650 SensorTag","Seq #": 20,"Uptime (sec)": 617,"Def Rout": "fe80::212:4b00:7b1:7c03","RSSI (dBm)": -42,"Battery Temp (C)": 28,"Battery Volt (mV)": 2828,"Air Pressure (hPa)": 998.25,"Air Temp (C)": 27.71,"Object Temp (C)": 24.812,"Ambient Temp (C)": 27.593,"Light (lux)": 119.76,"HDC Humidity (%RH)": 54.04,"HDC Temp (C)": 27.95,"Acc X (G)": 0.38,"Acc Y (G)": -0.64,"Acc Z (G)": -0.71,"Gyro X (deg per sec)": -5.27,"Gyro Y (deg per sec)": 3.10,"Gyro Z (deg per sec)": -3.16}, {"d":{"myName": "TI CC2650 SensorTag","Seq #": 21,"Uptime (sec)": 647,"Def Rout": "fe80::212:4b00:7b1:7c03","RSSI (dBm)": -34,"Battery Temp (C)": 27,"Battery Volt (mV)": 2828,"Air Pressure (hPa)": 998.25,"Air Temp (C)": 28.06,"Object Temp (C)": 28.156,"Ambient Temp (C)": 27.906,"Light (lux)": 502.40,"HDC Humidity (%RH)": 50.28,"HDC Temp (C)": 28.24,"Acc X (G)": -0.61,"Acc Y (G)": -0.76,"Acc Z (G)": 0.07,"Gyro X (deg per sec)": 29.33,"Gyro Y (deg per sec)": -23.46,"Gyro Z (deg per sec)": 7.67}, {"d":{"myName": "TI CC2650 SensorTag","Seq #": 22,"Uptime (sec)": 677,"Def Rout": "fe80::212:4b00:7b1:7c03","RSSI (dBm)": -31,"Battery Temp (C)": 28,"Battery Volt (mV)": 2828,"Air Pressure (hPa)": 998.30,"Air Temp (C)": 28.17,"Object Temp (C)": 22.593,"Ambient Temp (C)": 27.968,"Light (lux)": 502.40,"HDC Humidity (%RH)": 46.07,"HDC Temp (C)": 27.93,"Acc X (G)": 0.01,"Acc Y (G)": 0.02,"Acc Z (G)": 0.97,"Gyro X (deg per sec)": -0.13,"Gyro Y (deg per sec)": -0.19,"Gyro Z (deg per sec)": 0.79}]
```

Then we use the subscriber json file to get the sensor data and parse it. In order to save the sensor data in influxdb we run the influxdb and create our db with below information:

Database name: mydb

Measurement name: weather

Also we configure the telegraf as data collector to get data from broker and write them in influxdb. We add topic names of data.

The data from subscriber need to change and be suitable to save in our db so we change the subscriber code as bellows:

```
var mqtt = require('mqtt');
```

```

// Create an MQTT client (here using static IP of server)
var client = mqtt.connect('mqtt://192.168.213.131:1883')
// Indicate what topics we care about
client.on('connect', function () {
// get data from vm subscriber sensor
client.subscribe('cc26xx/00124ba00809/evt/status/fmt/json');
})
// Respond to message on subscribed topic(s)
client.on('message',
function (topic, message) {
//our code to parse data and remove space between words
data=JSON.parse(message)
Object.keys(data.d).forEach(function(prop) {
    //object key change name
    console.log(prop.replace(/("[^"]*"|\\s/g, "$1"))
    console.log(data.d[prop])//value
    var topicName=prop.replace(/("[^"]*"|\\s/g, "$1");
    var value=data.d[prop]
}) ;
});

```

Then we need to publish data so we write publisher

```
client.publish(topicName,value.toString())
```

result of parsing data as bellows:

```
Node.js command prompt - node subscriberTest.js
27
BatteryVolt(mV)
2781
AirPressure(hPa)
1009.41
AirTemp(C)
26.57
ObjectTemp(C)
22.5
AmbientTemp(C)
26.125
Light(lux)
140.96
HDCHumidity(%RH)
38.28
HDCTemp(C)
26.42
AccX(G)
0.01
AccY(G)
0.03
AccZ(G)
0.99
GyroX(degpersec)
-0.74
GyroY(degpersec)
1.08
GyroZ(degpersec)
0.98
```

And we get these result from broker

```
Node.js command prompt - mosquitto -c mosquitto.conf -v
1555001418: New client connected from 147.215.189.15 as mosqsub/2404-PC2409F (c1, k60).
1555001418: Sending CONNACK to mosqsub/2404-PC2409F (0, 0)
1555001418: Received SUBSCRIBE from mosqsub/2404-PC2409F
1555001418: cc26xx/00124ba0bd89/cmd/+/fmt/json (QoS 0)
1555001418: mosqsub/2404-PC2409F 0 cc26xx/00124ba0bd89/cmd/+/fmt/json
1555001418: Sending SUBACK to mosqsub/2404-PC2409F
1555001423: Socket error on client mosqsub/2404-PC2409F, disconnecting.
1555001435: New connection from 147.215.189.15 on port 1883.
1555001435: New client connected from 147.215.189.15 as mosqsub/13388-PC2409F (c1, k60).
1555001435: Sending CONNACK to mosqsub/13388-PC2409F (0, 0)
1555001435: Received SUBSCRIBE from mosqsub/13388-PC2409F
1555001435: cc26xx/00124ba0bd89/evt/status/fmt/json (QoS 0)
1555001435: mosqsub/13388-PC2409F 0 cc26xx/00124ba0bd89/evt/status/fmt/json
1555001435: Sending SUBACK to mosqsub/13388-PC2409F
1555001445: Received PUBLISH from d::cc26xx:00124ba0bd89 (d0, q0, r0, m0, 'cc26xx/00124ba0bd89/evt/status/fmt/json', ... (474 bytes))
1555001445: Sending PUBLISH to mqttjs_30d42bd9 (d0, q0, r0, m0, 'cc26xx/00124ba0bd89/evt/status/fmt/json', ... (474 bytes))
1555001445: Sending PUBLISH to mosqsub/13388-PC2409F (d0, q0, r0, m0, 'cc26xx/00124ba0bd89/evt/status/fmt/json', ... (474 bytes))
1555001445: Received PUBLISH from mqttjs_30d42bd9 (d0, q0, r0, m0, 'myName', ... (19 bytes))
```

Topics name

telegraf collect data as belows:

```
1555001776: New connection from 147.215.189.15 on port 1883.
1555001776: New client connected from 147.215.189.15 as mqttjs_30d42bd9 (c1, k60).
1555001776: Sending CONNACK to mqttjs_30d42bd9 (0, 0)
1555001776: Received SUBSCRIBE from mqttjs_30d42bd9
1555001776: cc26xx/00124ba0bd89/evt/status/fmt/json (QoS 0)
1555001776: mqttjs_30d42bd9 0 cc26xx/00124ba0bd89/evt/status/fmt/json
1555001776: Sending SUBACK to mqttjs_30d42bd9
1555001796: Received PINGREQ from mosqsub/13388-PC2409F
1555001796: Sending PINGRESP to mosqsub/13388-PC2409F
1555001796: New connection from 127.0.0.1 on port 1883.
1555001796: New client connected from 127.0.0.1 as Telegraf-Consumer-QB47D (c1, k60).
1555001796: Sending CONNACK to Telegraf-Consumer-QB47D (0, 0)
1555001796: Received SUBSCRIBE from Telegraf-Consumer-QB47D
1555001796: AccY(G) (QoS 0)
1555001796: Telegraf-Consumer-QB47D 0 AccY(G)
1555001796: AccZ(G) (QoS 0)
1555001796: Telegraf-Consumer-QB47D 0 AccZ(G)
1555001796: GyroX(degpersec) (QoS 0)
1555001796: Telegraf-Consumer-QB47D 0 GyroX(degpersec)
1555001796: GyroY(degpersec) (QoS 0)
1555001796: Telegraf-Consumer-QB47D 0 GyroY(degpersec)
1555001796: GyroZ(degpersec) (QoS 0)
1555001796: Telegraf-Consumer-QB47D 0 GyroZ(degpersec)
1555001796: myName (QoS 0)
1555001796: Telegraf-Consumer-QB47D 0 myName
1555001796: AccX(G) (QoS 0)
1555001796: Telegraf-Consumer-QB47D 0 AccX(G)
1555001796: Sending SUBACK to Telegraf-Consumer-QB47D
1555001805: Received PUBLISH from d::cc26xx:00124ba0bd89 (d0, q0, r0, m0, 'cc26xx/00124ba0bd89/evt/status/fmt/json', ... (476 bytes))
```

Topics name

The in writing data to influxdb we got this unfortunately

```
["d":{"myName":"TI CC2650 SensorTag","Seq #":8,"Uptime (sec)":259,"Def Ro
,"Battery Temp (C)":27,"Battery Volt (mV)":2871,"Air Pressure (hPa)":1009
18,"Ambient Temp (C)":24.843,"Light (lux)":129.36,"HDC Humidity (%RH)":38
Y (G)":0.03,"Acc Z (G)":0.98,"Gyro X (deg per sec)":-1.00,"Gyro Y (deg pe
["d":{"myName":"TI CC2650 SensorTag","Seq #":9,"Uptime (sec)":289,"Def Ro
,"Battery Temp (C)":27,"Battery Volt (mV)":2871,"Air Pressure (hPa)":1009
50,"Ambient Temp (C)":24.781,"Light (lux)":106.24,"HDC Humidity (%RH)":38
Y (G)":0.03,"Acc Z (G)":0.98,"Gyro X (deg per sec)":-0.89,"Gyro Y (deg pe
["d":{"myName":"TI CC2650 SensorTag","Seq #":10,"Uptime (sec)":319,"Def Ro
,"Battery Temp (C)":27,"Battery Volt (mV)":2828,"Air Pressure (hPa)":1009
281,"Ambient Temp (C)":24.687,"Light (lux)":110.40,"HDC Humidity (%RH)":38
Y (G)":0.02,"Acc Z (G)":0.99,"Gyro X (deg per sec)":-1.04,"Gyro Y (deg pe
["d":{"myName":"TI CC2650 SensorTag","Seq #":11,"Uptime (sec)":349,"Def Ro
,"Battery Temp (C)":27,"Battery Volt (mV)":2828,"Air Pressure (hPa)":1009
12,"Ambient Temp (C)":24.625,"Light (lux)":137.75,"HDC Humidity (%RH)":39
Y (G)":0.03,"Acc Z (G)":0.99,"Gyro X (deg per sec)":-1.08,"Gyro Y (deg pe
["d":{"myName":"TI CC2650 SensorTag","Seq #":12,"Uptime (sec)":379,"Def Ro
7,"Battery T
656,"Ambient
Y (G)":0.03Your environment has been set up for using Node.js 10.13.0 (x64) and npm.
And Y (G)":0.03
1,"Battery T U:\IOT\project\resources\telegraf
687,"Ambient
Y (G)":0.02 U:\IOT\project\resources\telegraf>telegraf -config telegraf.conf
2019-04-11T16:56:36Z I! Starting Telegraf 1.9.4
2019-04-11T16:56:36Z I! Unable to create /Program Files/Telegraf/telegraf.log (open /Program Files/Telegraf/telegraf.log
2019-04-11T16:56:36Z I! Loaded inputs: inputs.mqtt_consumer inputs.win_perf_counters
2019-04-11T16:56:36Z I! Loaded aggregators:
2019-04-11T16:56:36Z I! Loaded processors:
2019-04-11T16:56:36Z I! Loaded outputs: influxdb
2019-04-11T16:56:36Z I! Tags enabled: host=PC2409F
2019-04-11T16:56:36Z I! [agent] Config: Interval:10s, Quiet:false, Hostname:PC2409F, Flush Interval:10s
2019-04-11T16:56:36Z I! [inputs.mqtt_consumer] Connected [tcp://localhost:1883]
2019-04-11T16:56:45Z E! [inputs.mqtt_consumer]: Error in plugin: strconv.ParseFloat: parsing \"SensorTag\": invalid syntax
```

In this mini project and other TPs we have learnt how to configure the electronic devices with internet and monitor devices; we have simulated with VMWare, connected, run broker(MQTT), publish data (MQTT), collect data(Telegraf), save data (influxdb) and visualize data (grafana). But unfortunately for above error we were not able to write data on influxdb and visualize them with grafana.