IBM Watson IoT – T1 – MQTT Client

Commissioning task T1

In this commissioning task a normal computer with Windows operating system and a network connection to Internet will be needed. No special hardware or software will be needed.

1. Environment for completing this exercise – IBM Bluemix Watson IoT

Please study the instruction sheet ” IBM Watson - MKR1000 – Getting Started”. According to these instructions you will be registering yourself to IBM Bluemix and you will be defining an “organization” and creating two IoT devices.

The instruction ask you to create Arduino MKR1000 –development card as your device. Anyway, this time we will be creating a “device”, the role of which we can play with almost any MQTT client. MQTT client apps are available for Windows, Android and iOS.

Please create a “device” or two according to the instructions. The type can be for example “A_DeskTop_Editor”. Please remember to save in safe place the just created Organization ID, Device Type, Device ID, Authentication method, Authentication token! IT IS EXTREMELY IMPORTANT TO SAVE ON A TEXT FILE THE AUTHENTICATION TOKEN !

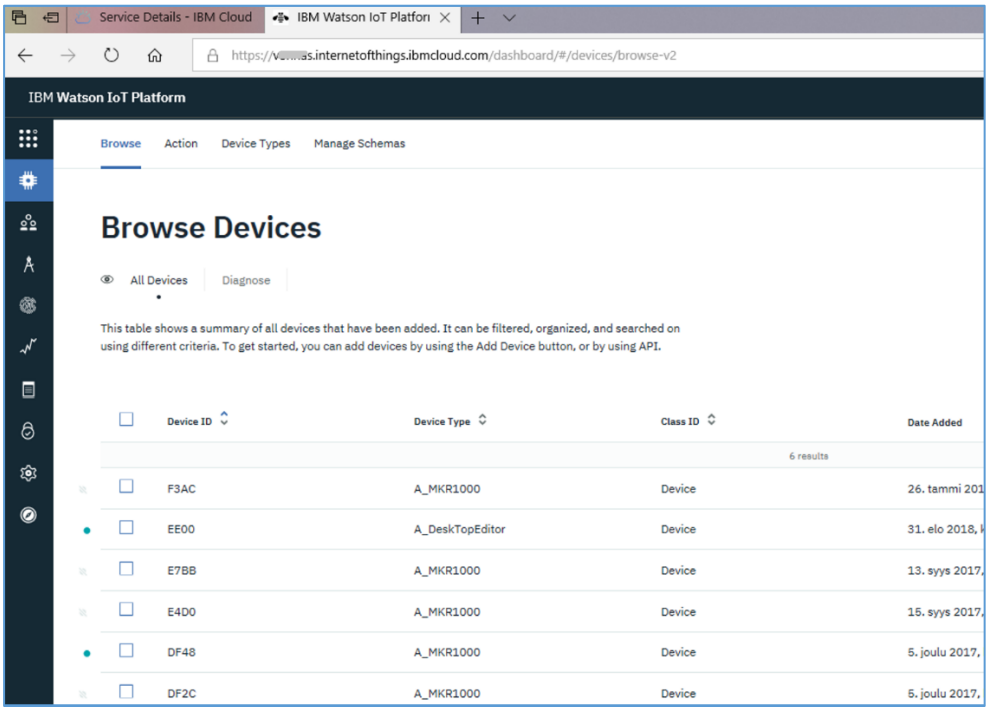

Fig 1.1 One desk top application and several real devices visible in IBM Watson IoT device list. Those marked with green dot are online at the moment.

2. Monitoring MQTT broker with browser extension, Windows app, Android App or iOS App.

2.1 Selecting an app

Web browsers have extensions for various purposes. Open your browser and search for “browser extension mqtt”. As a result you should get a link to for example MQTTBox Chrome Web Store. Follow the link for installation.

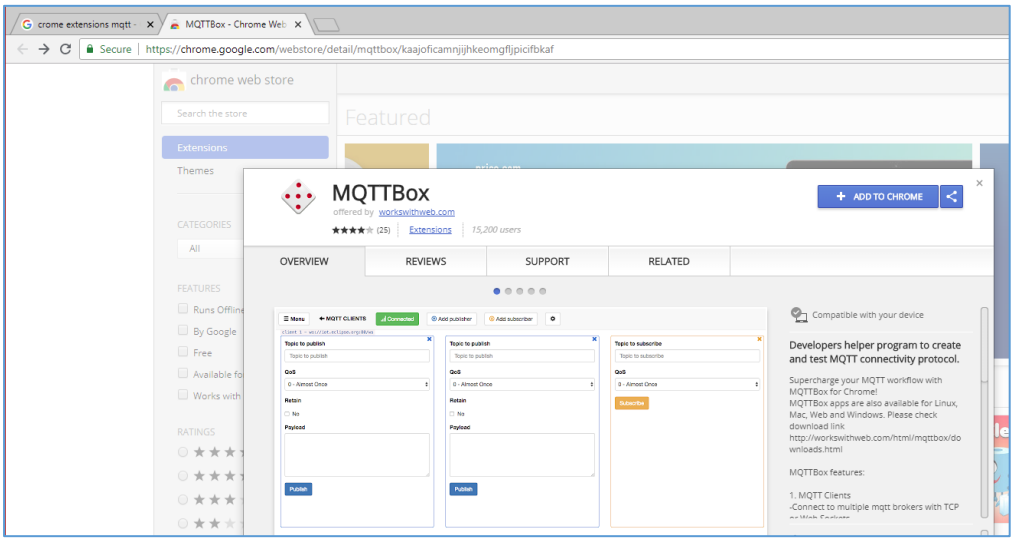

Fig 2.1.1 MQTTBox as browser extension / 2018

An MQTT APP can be installed on Windows computer, too. In the Microsoft Store search for mqtt in department Apps.

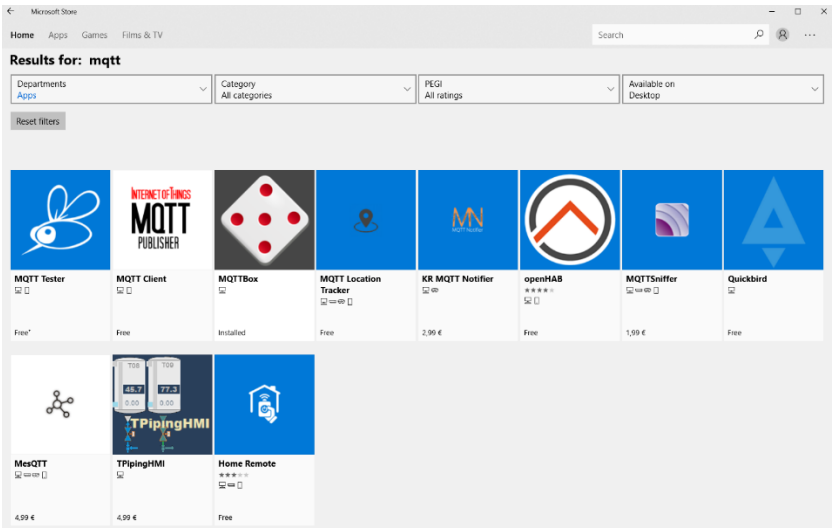

Fig 2.1.2. Search results in Microsoft Store for mqtt / 2018

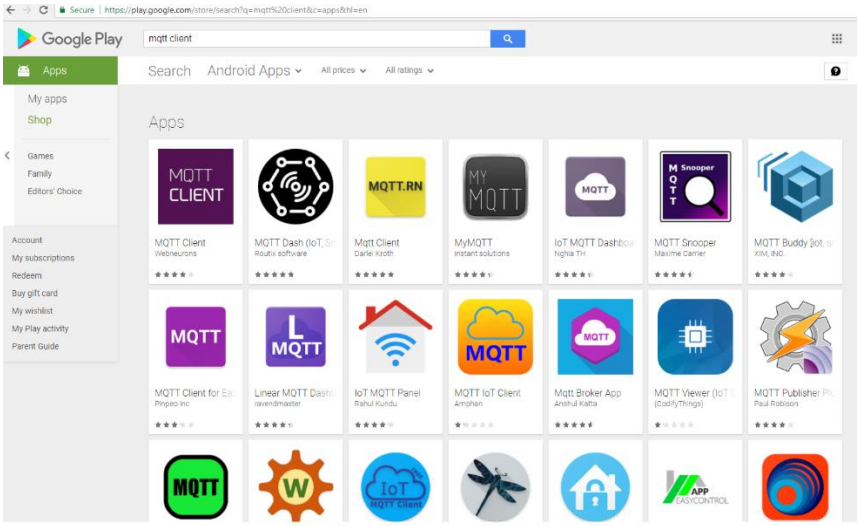

Fig 2.1.3 Android apps for MQTT client in Google Play / 2018

Install the MQTTbox either as browser extension or as a Windows APP. It seems to be easy to use.

The MQTTbox can be installed as well directly from an .exe file which is available at http://workswithweb.com/mqttbox.html

A number of apps are available for Android and iOS as well.

In opening the APP or extension you will be given a possibility to add a client. Please check out that in the app you selected you will be able to define in minimum:

- MQTT Client name
- MQTT Client Id
- Protocol
- Host
- Username
- Password
- Topic
- Quality of Service
- Payload Type
- Payload

Possibility on setting another port than standard 1883 would be a bonus!

2.2 Connecting to mosquitto mqtt broker

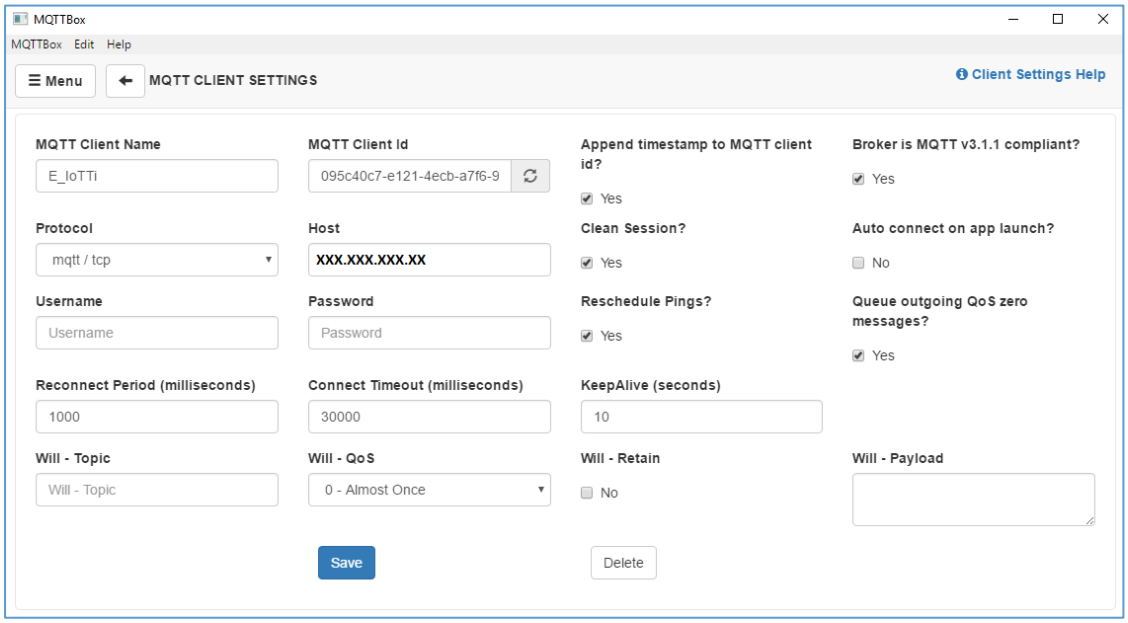

Fig. 2.2.1 MQTTBox settings for a new client.

For the settings fill in the text boxes as seen above.

In the **MQTT Client Name** you can create your own client name.

The **MQTT Client Id** is generated automatically. Selecting the **append time stamp to MQTT client id** helps avoiding to contact the server again with exactly the same client id.

In **Host** please type the IP address of the mqtt broker you are connecting to. TAMK - TUNI offers for our purposes the broker at address 193.167.167.59 . HAMK offers for education and research a broket at iot.research.hamk.fi .

In MQTTbox there is no option for port. The MQTT standard port 1883 will be used.

Select the Protocol mqtt/tcp.

User name and password can be left empty when there is no user authentication set in this MQTT broker.

For detailed information on the other settings please have a look at web page
https://www.hivemq.com/blog/mqtt-essentials-part-3-client-broker-connection-establishment

Once the connection has been established you can test it. To succeed with connecting you need to be in a network where communication to port 1883 is open in the firewall. In HAMK premises that is HAMKvisitor WLAN-network. Typically that port is open in home network or a network shared as WLAN from your own mobile phone.

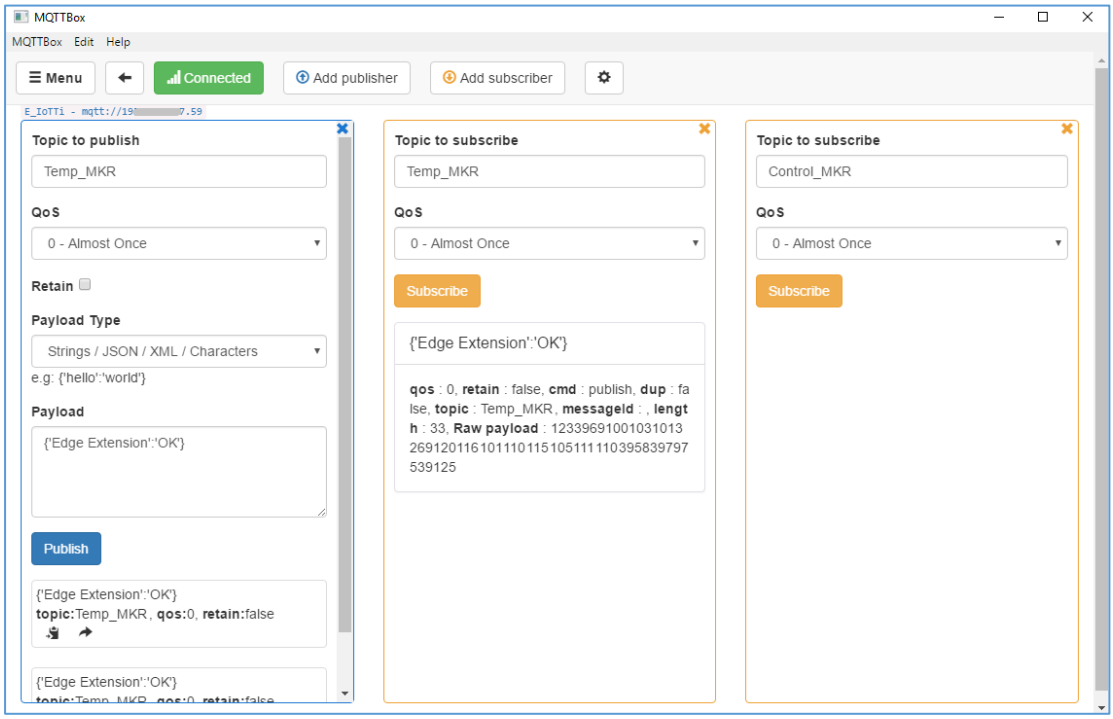

Fig 2.2.2 Broker tested with Windows APP MQTTBox. In browser extension MQTTBox the user interface is exactly the same.

Broker is tested with sending a payload in JSON format into a new topic. A subscriber is created to listen to that topic. It seems to work all right as seen on the picture above.

You may as well test the operation with IoT hardware. This is not necessary if the hardware is not available.

Arduino MKR1000 is powered up next. It starts publishing the temperature data. In MQTTBox a new subscriber is created to listen for topic Temp_MKR. The MKR1000 is publishing in JSON format under this topic. Seems to work all right.

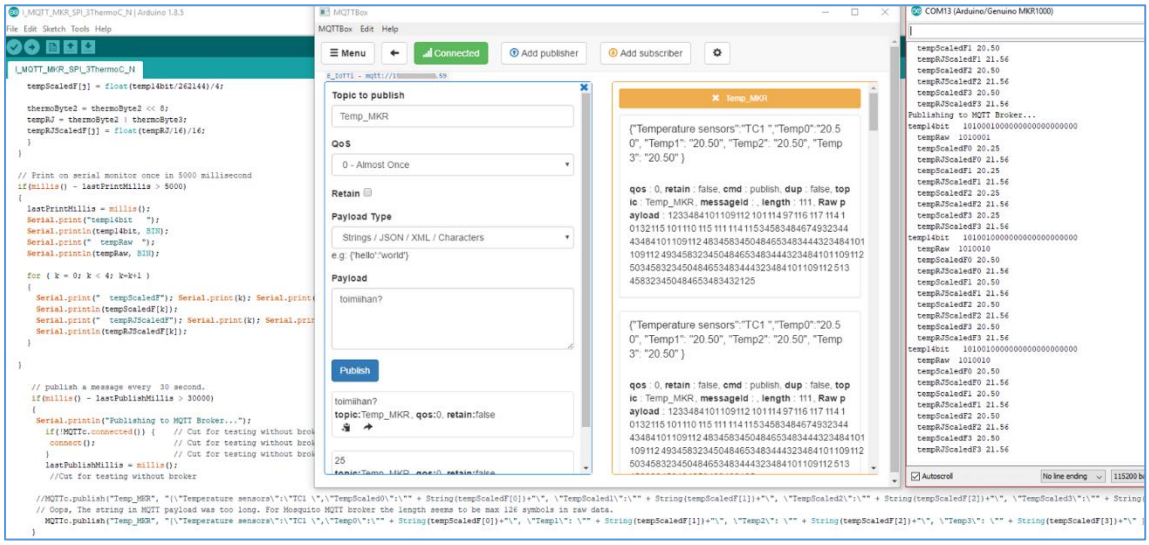

Fig 2.2.3. Publishing several temperature sensor values in JSON format under topic “Temp_MKR” to MQTT broker and subscribing to the topic “Temp_MKR” in MQTT client.

The maximum length of the MQTT payload seems to be 126 symbols in the raw data. Othervise the message seems not to be available for subscription. In your string it is rather difficult to count the exact number of characters. Although in MQTT the payload maximum size should be 256 mb!

JSON format is not necessary. Plain text or integer or float values can be used as well. The message payload will be sent as character string.

TASK A.

Please study from MQTT documentation what is said about MQTT Topic and topic hierarchy.

Please define for yourself a number of topics with some kind of hierarchy. Just write down your topics in a text sheet.

Publish messages to MQTT broker using the lowest level of your topic hierarchy.

Subscribe on MQTT broker on your topics. Try with different hierarchies.

In the report, write some lines about what you learned!

TASK B

Try to publish with different Quality of Service (QoS) levels.

There are three QoS levels in MQTT:

At most once (0)

At least once (1)

Exactly once (2).

https://www.hivemq.com/blog/mqtt-essentials-part-6-mqtt-quality-of-service-levels

In the report, write some lines about what you learned!

3. Connecting to IBM Bluemix Watson IoT platform

In your MQTT client try to establish the client with parameters corresponding your “device” you defined in IBM Watson IoT.

Instruction are given here for:

IBM Bluemix MQTT Helper

MQTTBox application

a. IBM Bluemix MQTT Helper used for connecting the Watson IoT

IBM Bluemix has it's own testing tool called MQTT Helper. It is available at

http://mqtt-helper.mybluemix.net/

Fill in the authentication information you got when creating your organization ID and device:

Server:	<OrganizationID>.messaging.internetofthings.ibmcloud.com
Port:	1883
ClientID:	d: <OrganizationID>:<deviceType>:<DeviceID>
Username:	use-token-auth
Password:	<AuthenticationToken>
Clean Session:	OFF or ON
SSL:	OFF (and in Watson IoT please remember to select Security Settings, TSL Optional)

The message topic needs to be in certain format:

iot-2/evt/<topic>/fmt/json

In the Watson Topic it is defined that the message payload needs to be in JSON format. For example:

{ "name":"Pekka", "age":41, "city":"Oulu" }

b. MQTTBox application used for connecting the Watson IoT

An Application for MQTT messaging can be used for connecting to IBM Watson IoT platform. The client settings and topic settings just need to be created just in the correct format.

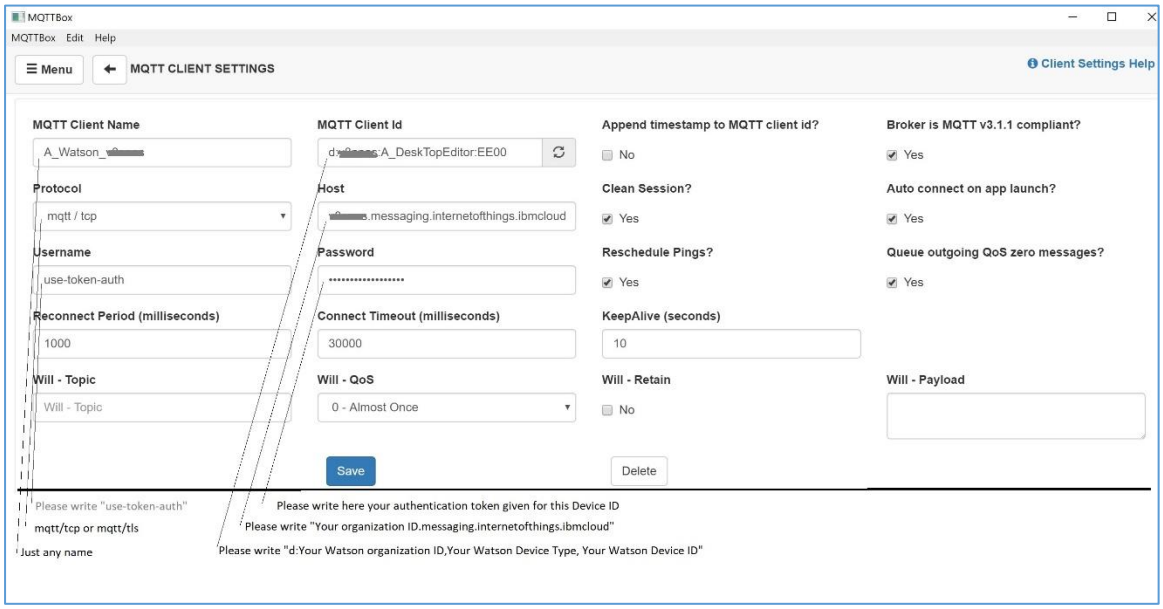

Fig 3.2.1 Settings on the MQTT Client window on MQTTBox application

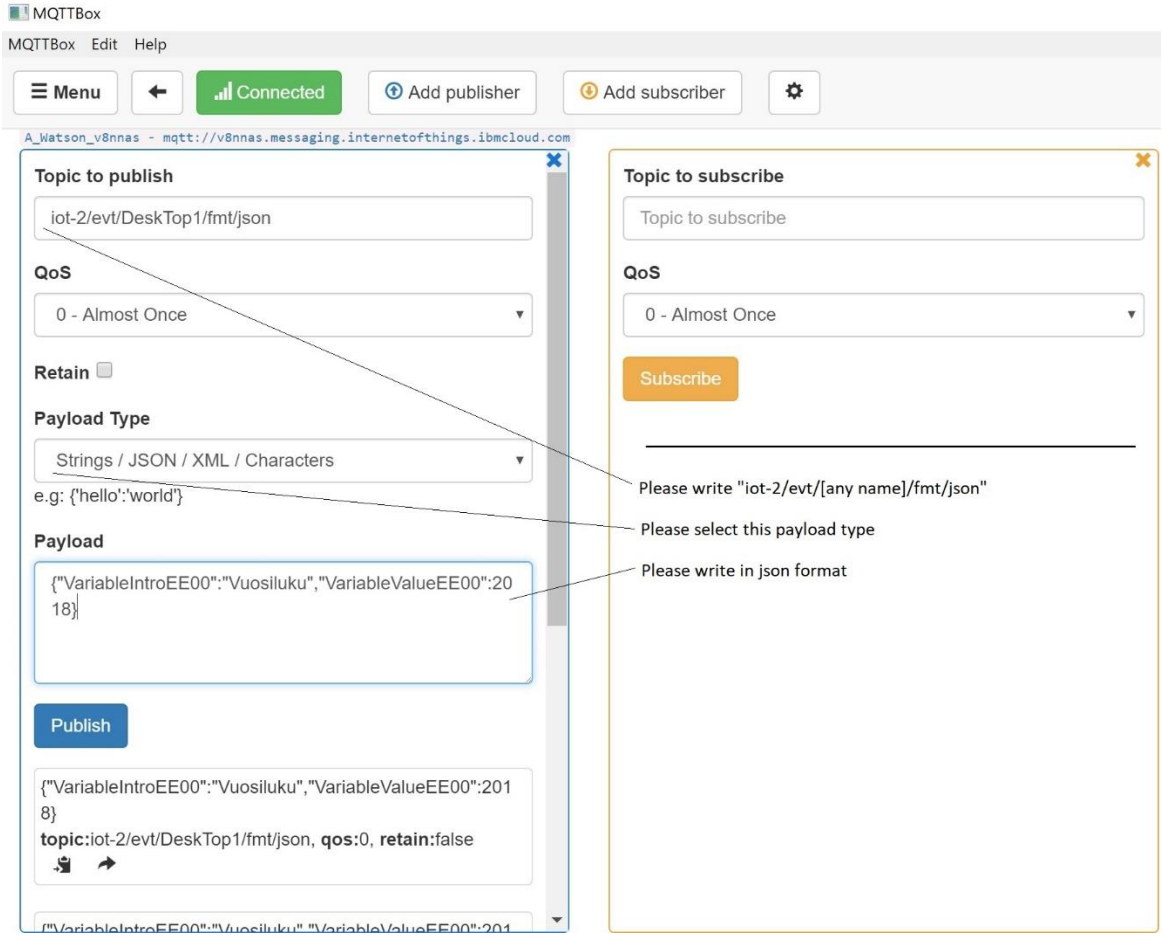

Fig 3.2.2 Definition of topic for publishing in Topic window on MQTTBox application

TASK C

Try out publishing data into the IBM Watson IoT. Can you publish a topic and a JSON formatted value? Can you publish text strings inside the JSON formatted payload? Can you publish number values inside the JSON formatted payload?

In the report, write some lines about what you learned!