Advantech AE Technical Share Document

Date	2018/10/16	**SR#**	1-3456257768
Category	□FAQ ■SOP	**Related OS**	N/A
Abstract	How to connect Mosquitto with iSensing MQTT?		
Keyword	WISE, Mosquitto, cloud upload, push notification, control end device		
Related Product	WISE-40XX series, WISE-42XX, WISE-44XX		

■ Problem Description:

This document shows that how to connect with Mosquitto and make sure upload/push data successfully. Utilize 3rd party MQTT broker to monitor the pushed data and control the status of WISE.

The iSensing MQTT for WISE is a bit different from ADAM-4000 and ADAM-6000. It supports MQTT with **websocket** format and **SSL** encryption package.

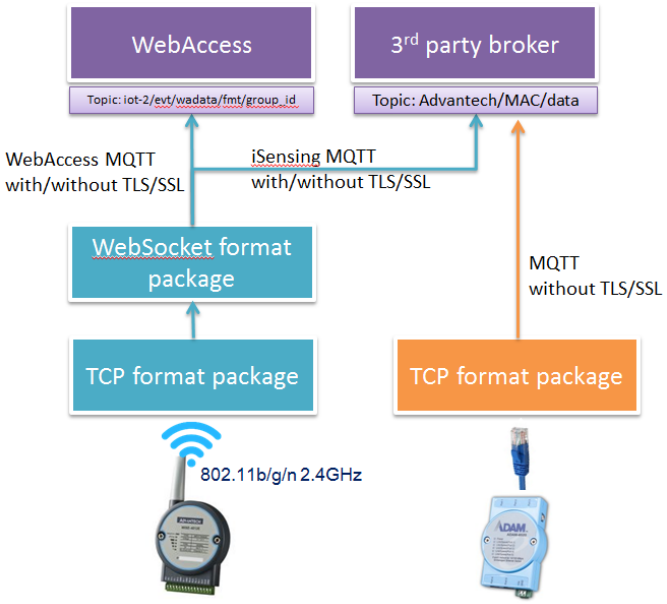


Figure 1. The MQTT for WISE and ADAM.

■ Brief Solution - Step by Step:

SOP video:

https://www.dropbox.com/s/nl31nhocf9d1uca/Advantech%20IIOT%20tutorial_iSensingMQTT.mp4?dl=0

NOTATION: Mosquitto holds TCP connection as default protocol. The attached Mosquitto .zip file is compiled with plug-in websocket library. Some of the setting is defined in the "mosquitto.tls.conf" file. **Please storage the broker under the exactly "D:\" folder in the demonstration steps.**

```
# http data as well. Set http_dir to a directory which contains the files you
# wish to serve. If this option is not specified, then no normal http
# connections will be possible.
#http_dir
http_dir D:\mosquitto\htdocs

# Set use_username_as_clientid to true to replace the clientid that a client
# connected with with its username. This allows authentication to be tied to
# the clientid, which means that it is possible to prevent one client
# disconnecting another by using the same clientid.
# If a client connects with no username it will be disconnected as not
# authorised when this option is set to true.
# Do not use in conjunction with clientid_prefixes.
# See also use_identity_as_username.
#use_username_as_clientid

# ----------------------------------------------------------------
# Certificate based SSL/TLS support
# ----------------------------------------------------------------
# The following options can be used to enable SSL/TLS support for
# this listener. Note that the recommended port for MQTT over TLS
# is 8883, but this must be set manually.
#
# See also the mosquitto-tls man page.

# At least one of cafile or capath must be defined. They both
# define methods of accessing the PEM encoded Certificate
# Authority certificates that have signed your server certificate
# and that you wish to trust.
# cafile defines the path to a file containing the CA certificates.
# capath defines a directory that will be searched for files
# containing the CA certificates. For capath to work correctly, the
# certificate files must have ".crt" as the file ending and you must run
# "c_rehash <path to capath>" each time you add/remove a certificate.
# cafile D:\mosquitto\certs\ca.crt
cafile D:\mosquitto\certs\m2mqtt_ca.crt
#capath

# Path to the PEM encoded server certificate.
#certfile D:\mosquitto\certs\server.crt
certfile D:\mosquitto\certs\m2mqtt_srv.crt

# Path to the PEM encoded keyfile.
#keyfile D:\mosquitto\certs\server.key
keyfile D:\mosquitto\certs\m2mqtt_srv.key

# This option defines the version of the TLS protocol to use for this listener.
# The default value allows v1.2, v1.1 and v1.0, if they are all supported by
```

path of listening file is defined

path of certification files are defined

Figure. The execute path are pre-defined.

In this document, we demonstrate how to make sure the connection between WISE and MQTT broker is successful. There are 2 demonstration topologies. One is using 3rd party MQTT client to subscribe the data published by WISE. Another one is to publish data, which can control the status of WISE.

The demonstration topology for DI channel is shown in figure 2, and DO channel control is shown in figure 3.

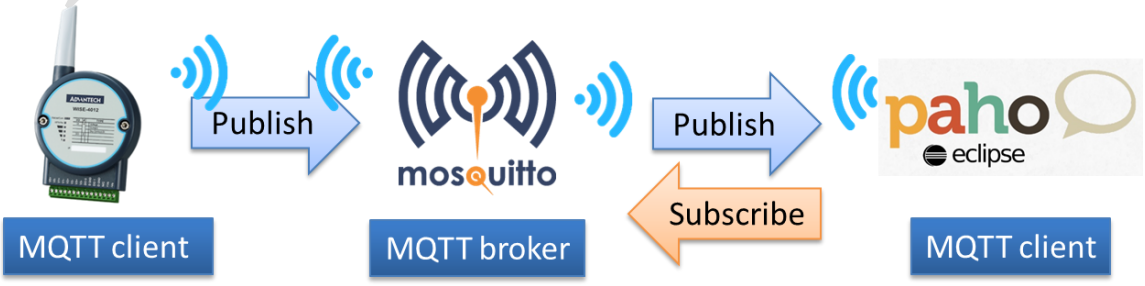


Figure 2. The demonstration topology – WISE is a publisher. (push DI)

Figure 3. The demonstration topology – WISE is a subscriber (control DO)

There are 3 topics for WISE modules:

- Advantech/MAC_of_WISE/Device_Status
- Advantech/MAC_of_WISE/data
- Advantech/MAC_address/ctl/do#

Video of this SOP document:
https://www.dropbox.com/s/899obg807d8xyxg/MQTT_iSensing.mp4?dl=0

Part I, WISE push DI channel status to MQTT broker.

Step1. Open command prompt from windows "Start Menu". Change the folder to the one which store Mosquitto. In the figure 4, the default port is 8443. Please note that this command should not be switched off during MQTT process.

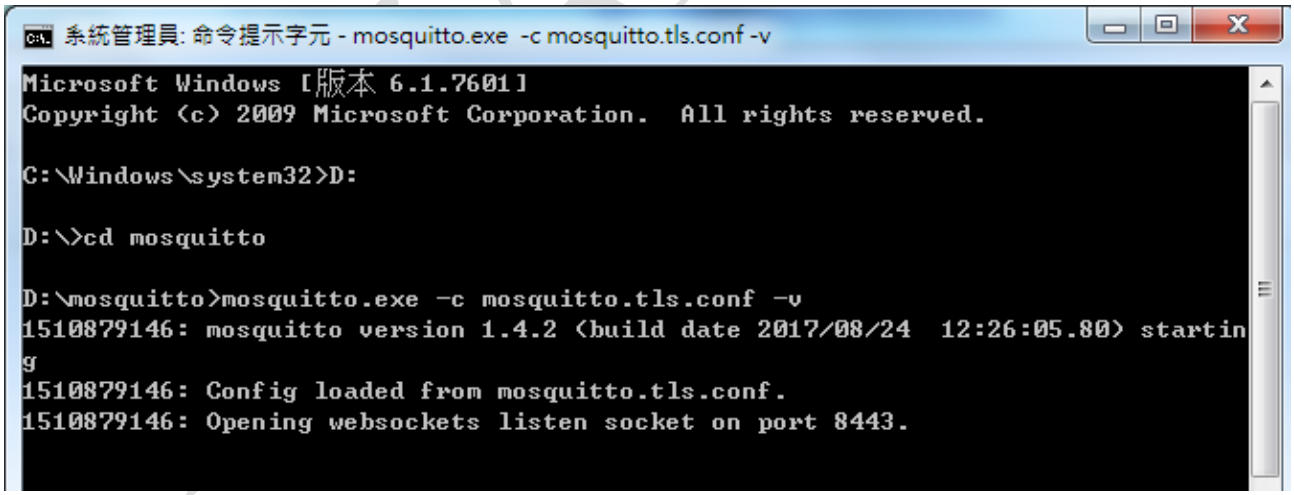


Figure 4. Command prompt.

Step 2. Open web browser and enter URL: https://IP_addr _of_PC:8443/. If the page shows as figure 5, which means Mosquitto open successfully and user can proceed to use MQTT client to connect to MQTT Broker. Click on Paho to open another tab for 3rd party MQTT client. Please note that this browser should **not** be turned off during MQTT process.

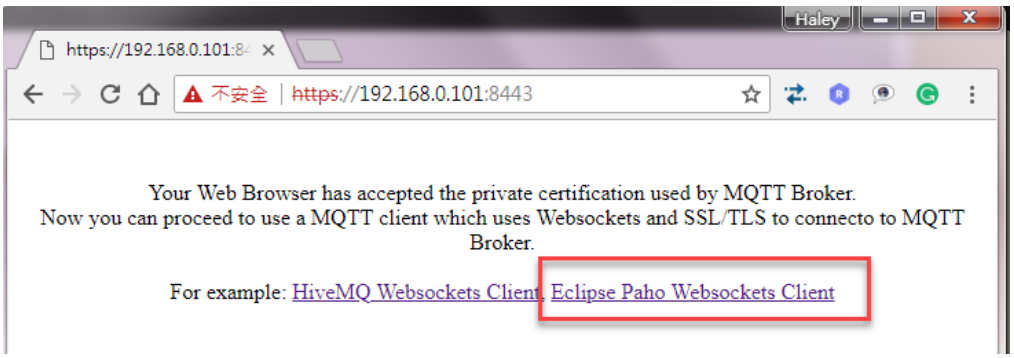


Figure 5. The message from browser.

If you get the result as figure 6. Which means the certification of Mosquitto we provided did not have the Website credentials. Just need simply go to advance option and go to the web address (figure 7).

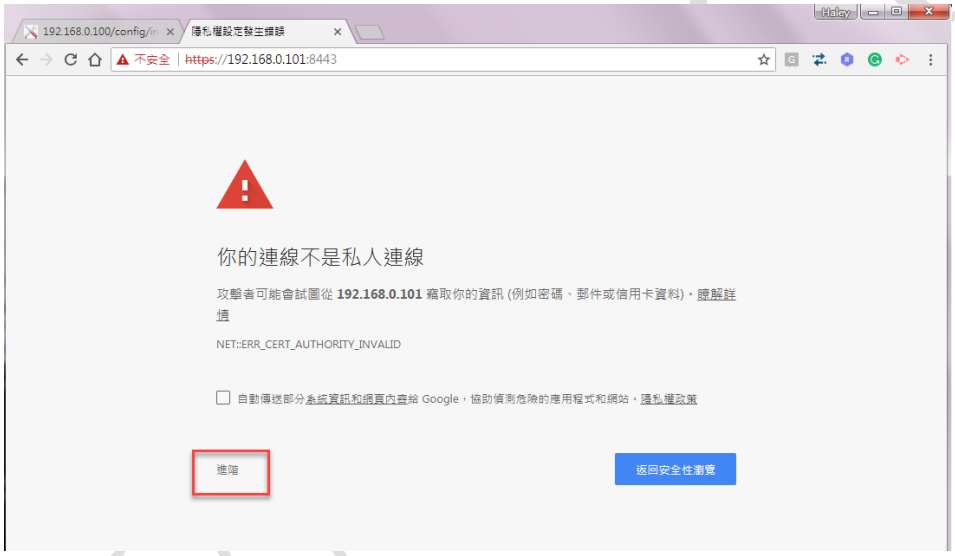


Figure 6. Error message.

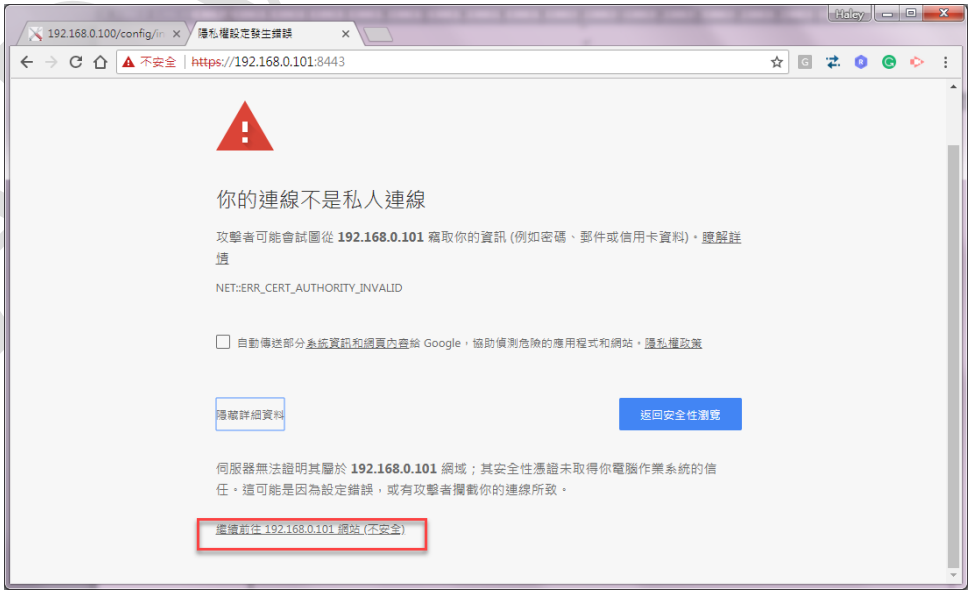


Figure 7. Go to the address of your broker.

Step 3. Please configure the WISE as figure 8. Remember to enable **SSL secure**.

In the .zip file of Mosquitto, there is no definition about user name and password. Please leave those parameters empty.

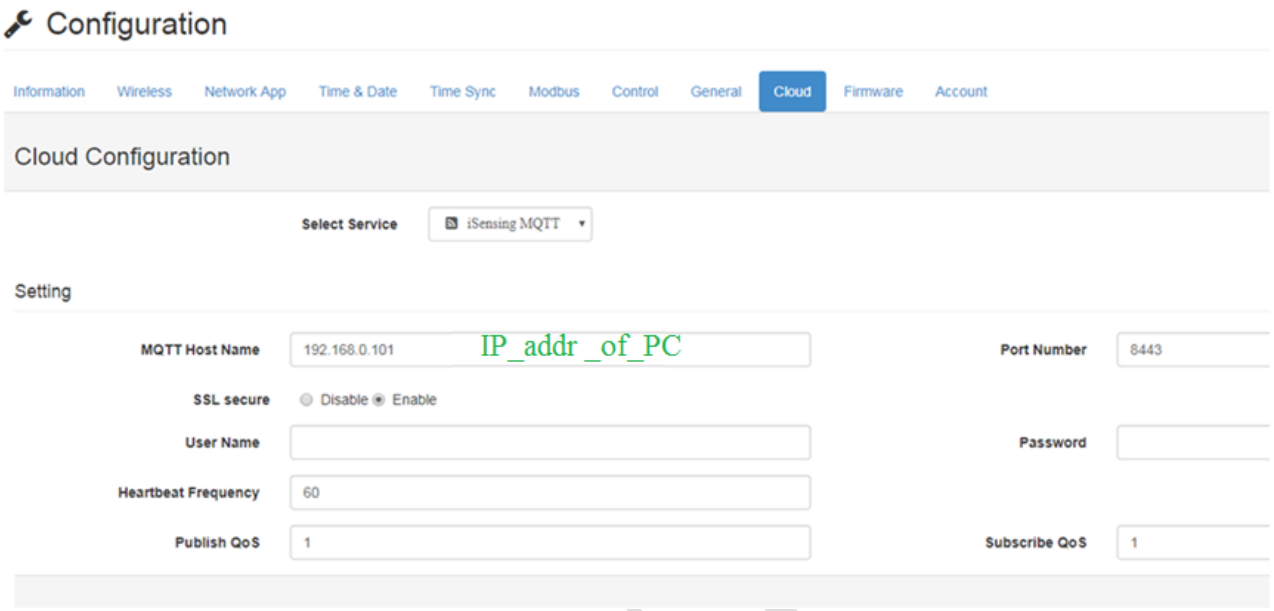


Figure 8. The configuration page of WISE.

Step 4. Enable push function of the WISE module.

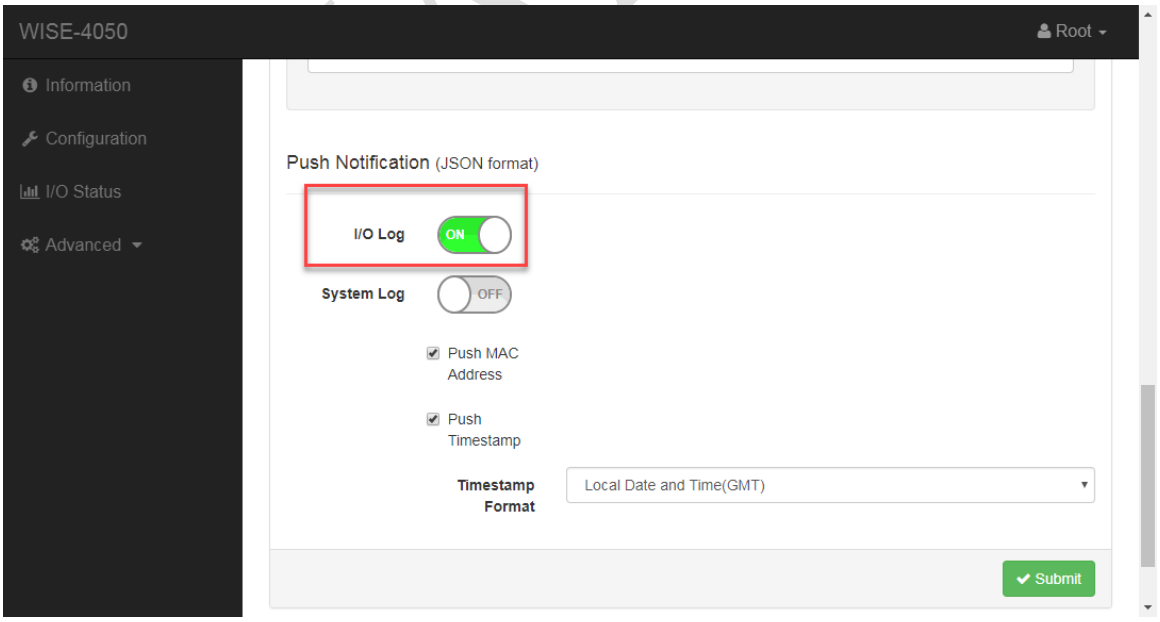


Figure 9. Enable push function of the WISE module

Step 5. Click on the Eclipe Paho option in the figure 10. The same as the link in red rectangle in

step 2, which is a 3rd party MQTT client. If the setting is correct, it will show "connected" after click "connect".

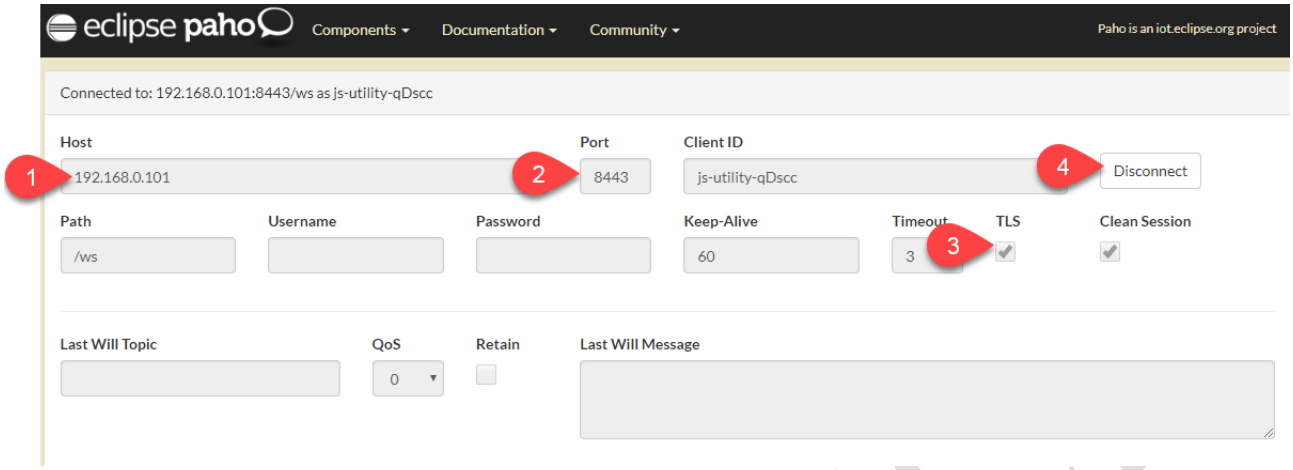


Figure 10. The configuration page of ECLIPE PAHO.

Step 6. Subscribe the messages published by WISE refers to the topology in figure 2.

Add a new subscription with topic Advantech/MAC_of_WISE/data.

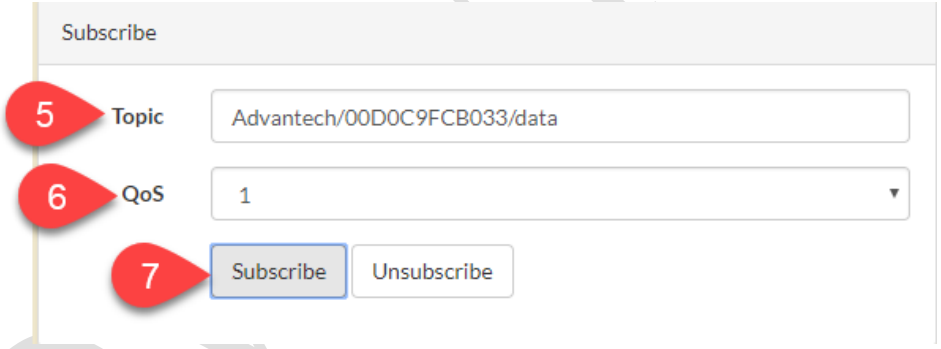
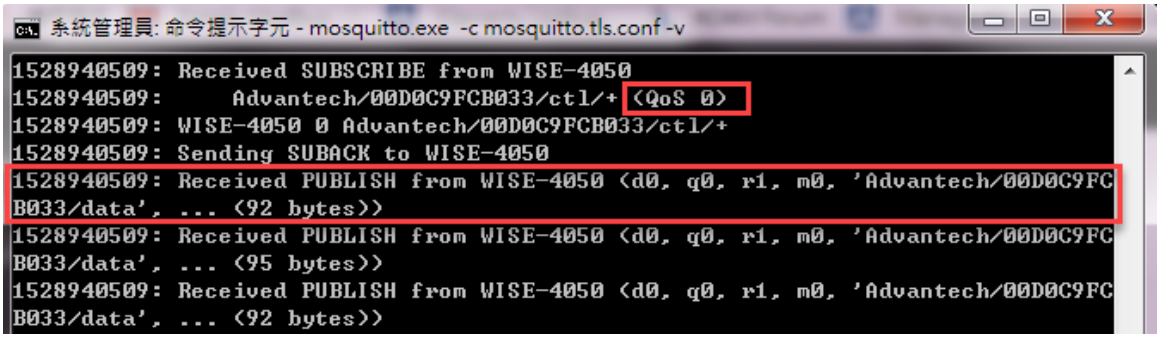


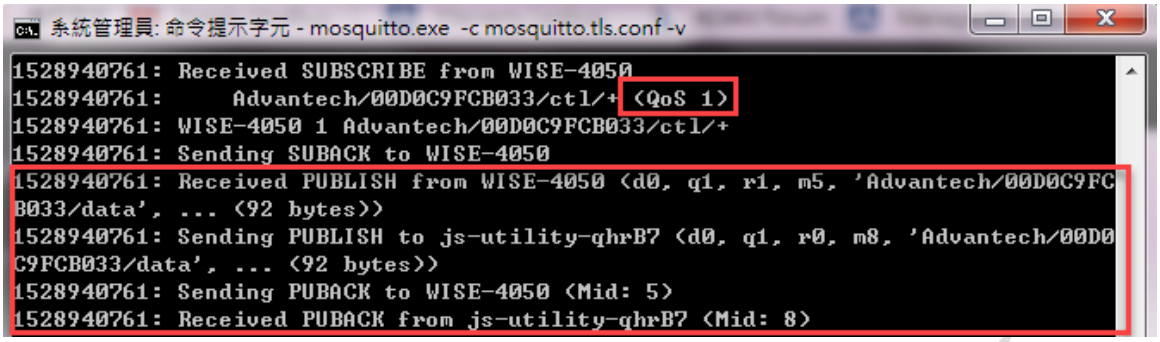
Figure 11. A new subscriptions.

Results of receiving DI status.

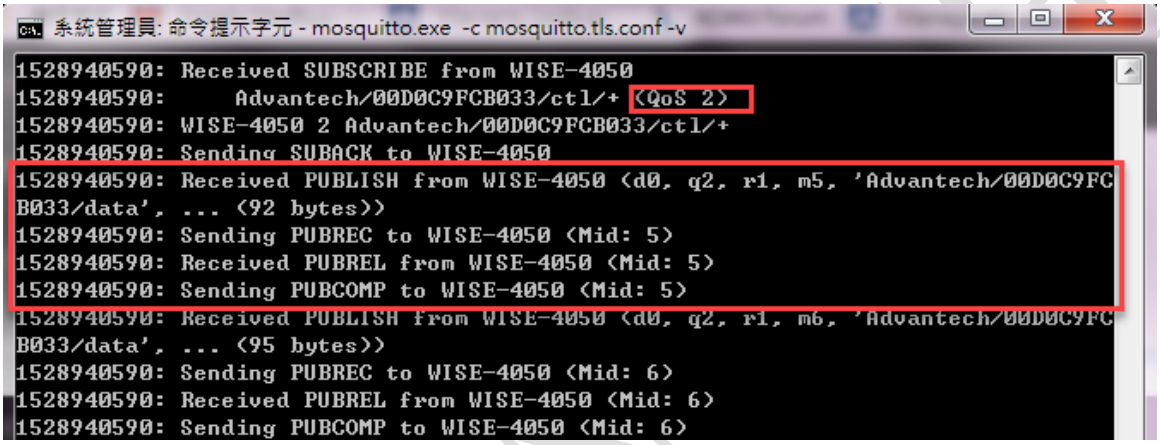
Monitor the data while WISE is uploading/pushing data to MQTT broker. As shown in figure 12, there will be lots of messages receiving from WISE.



(A) Messages received from WISE (QoS=0) on command panel.



(B) Messages received from WISE (QoS=1) on command panel.



(C) Messages received from WISE (QoS=2) on command panel.

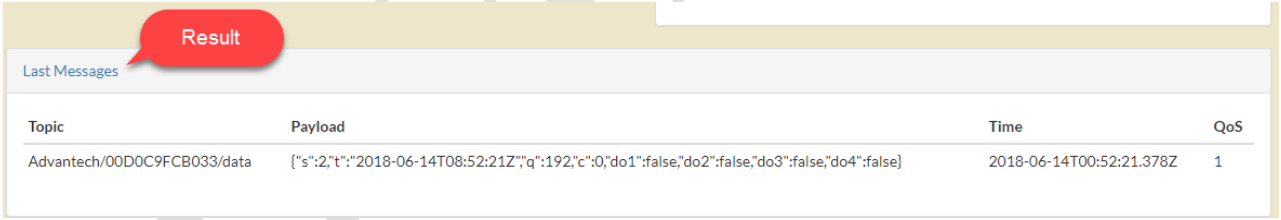


Figure 12. (D) Messages received from WISE on Paho (QoS=1).

Part II, WISE subscribe from MQTT broker and change DO channel status.

Step 1 – 5 are the same as in part I.

Step 6. Publish the messages, which can control the status of WISE module refers to the topology in figure 3.

Add a new topic Advantech/MAC_of_WISE/ctl/do1 (DO0 of WISE).

Add a new message {"v":true} / {"v":false}.

Description	Example MQTT Topic	Example JSON Data	Notes
Set the value of a digital output.	Advantech/0013430C981F/ctl/do1 Advantech/0013430C981F/ctl/do2 Advantech/0013430C981F/ctl/do3	{"v":true}	MQTT: do not set the retain bit when publishing messages to this topic; otherwise, an old retained message may change the state of the output.

Figure 13. The example MQTT topic and message.

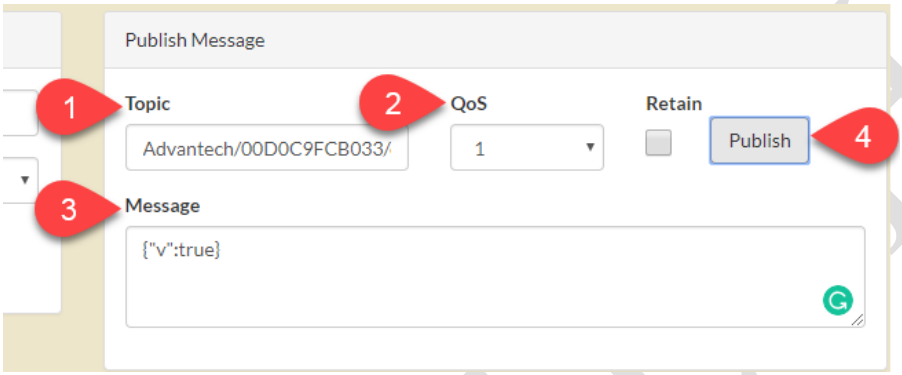


Figure 14. The example MQTT topic and message setting on ECLIPE PAHO.

Results of controlling DO status.

Monitor the DO channel of WISE while a user is publishing data from ECLIPE PAHO.

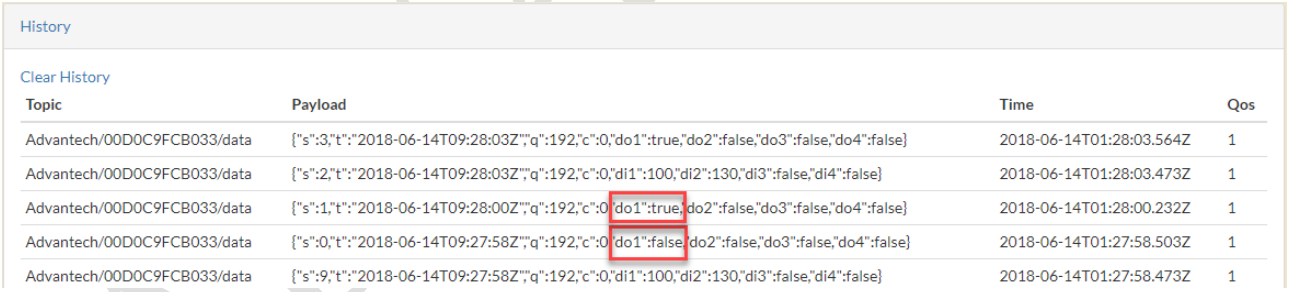


Figure 15. The results on Eclipe Paho.

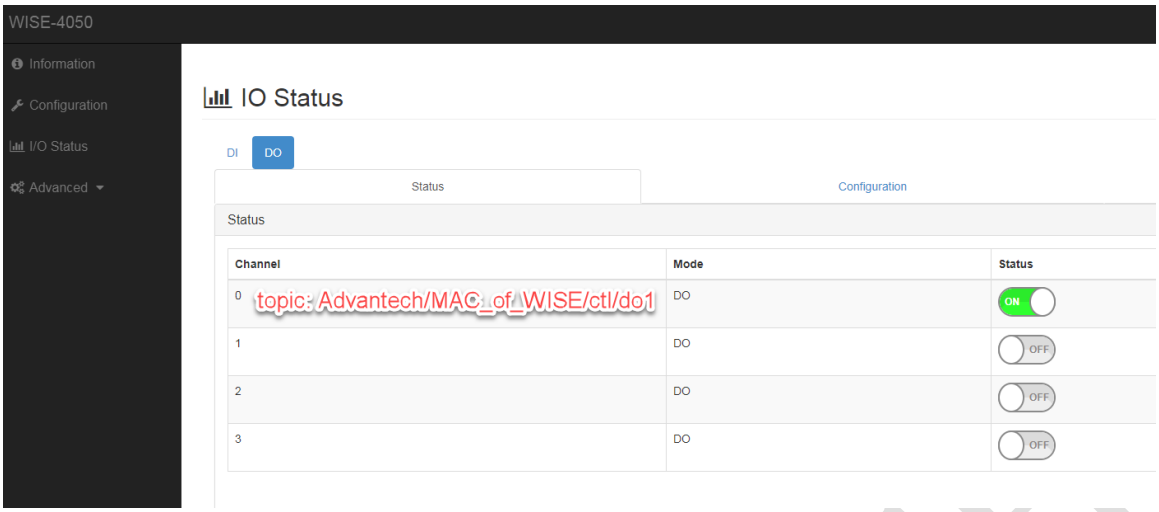


Figure 16. Ch0 is corresponding to do1 of the topic.