

Innovation And Embedded System (539305)

ภาคการศึกษาที่ 2 ปีการศึกษา 2564

อาจารย์ ดร.มานพ ธิมาปะโท
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์ มหาวิทยาลัยเทคโนโลยีสุรนารี

Suranaree University of Technology

Course

Compro 1&2

Microprocessor and internet of thing
539206 5(4-3-14)

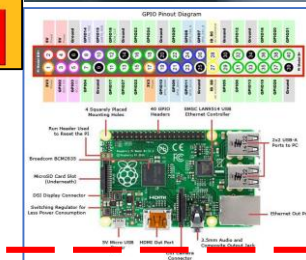
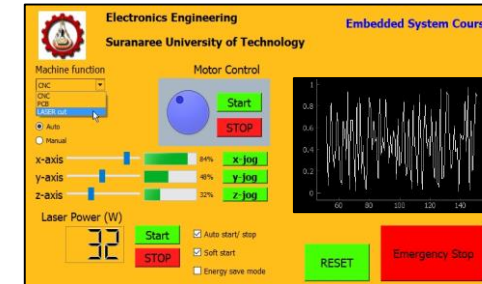
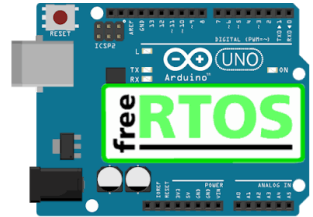
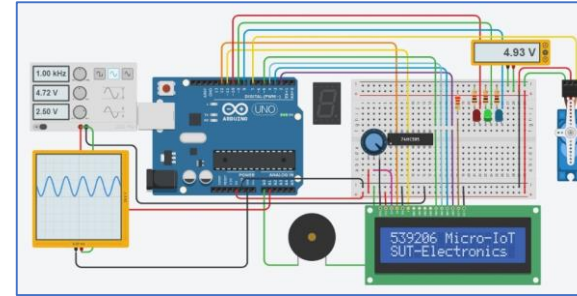
- ☐ C++ programming
- ☐ Arduino
- ☐ IoT ESP32
- ☐ HW interface

Embedded system and application
539302 5(4-3-14)

- ☐ Embedded Linux
- ☐ Python Programing
- ☐ Python GUI
- ☐ Arduino + Python app

Innovation and embedded system design
539305 2(0-6-9)

- ☐ Python GUI+IoT
 - 1 project
- ☐ LoRaWAN
- ☐ Node-red
 - 1 project



Course Outline 539305 2(0-6-9)

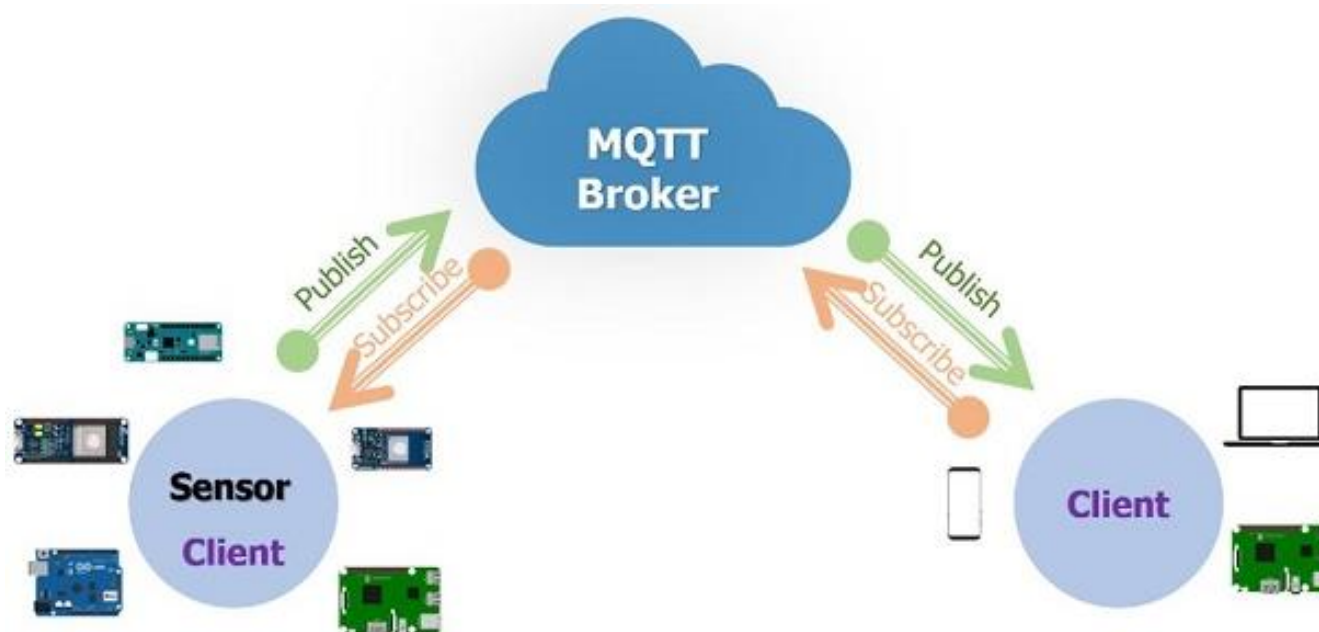
Category	Item	Topic	Week																							
			0		1		2		3		4		5		6		7		8		9		10		11	
			0.1	0.2	1.1	1.2	2.1	2.2	3.1	3.2	4.1	4.2	5.1	5.2	6.1	6.2	7.1	7.2	8.1	8.2	9.1	9.2	10.1	10.2	11.1	11.2
Introduction	1	Introduction (Outline, Score)					C	-																		
	2	Project					C	-																		
MQTT	3	MQTT Broker							C	L																
	4	MQTT pub sub client							C	L																
JSON	5	JSON									C	L														
	6	Arduino JSON									C	L														
Python MQTT	7	Python MQTT											C	L												
	8	Pub sub											C	L												
	9	Python MQTT Application											C	L												
Python-MQTT Project	10	Work													M	-		P								
	11	Present													M	-		P								
LoRAWAN	12	LoRa																	C	C						
	13	LoRaWAN																	C	C						
	14	LoRaWAN practical																	C	C						
Node-red	15	Node-red																			C	L				
	16	JS																			C	L				
	17	Node-red LoRaWAN application																			C	L				
Node-red Project	18	Start																					F	-		P
	19	Update																					F	-		P
	20	Final																					F	-		P

C = Class
L = Lab
M=Midterm

P = Present
F= Final

MQTT

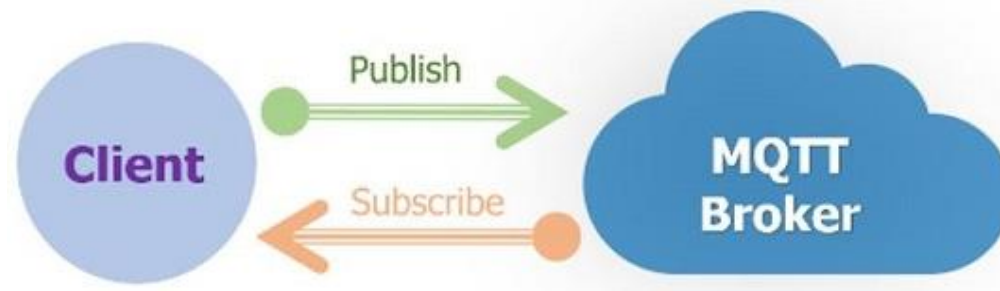
- ❑ MQTT ย่อมาจาก Message Queuing Telemetry Transport เป็นโพรโตคอลสำหรับใช้ในสื่อสารข้อมูลระหว่าง Machine to Machine (M2M)
- ❑ ถูกคิดค้นขึ้นในปี ค.ศ. 1999 โดย Dr Andy Stanford-Clark จาก IBM และ Arlen Nipper จาก Arcom (now Eurotech)
- ❑ ออกแบบมาเพื่อใช้สื่อสารในระบบเครือข่ายที่มีทรัพยากรค่อนข้างจำกัด ใช้งานแบนด์วิธต่ำ
- ❑ สามารถ publish-subscribe ข้อมูลระหว่าง Device เพื่อสื่อสารกันระหว่างอุปกรณ์
- ❑ ถ้ามองในด้านที่เกี่ยวกับ Internet of Things จะสามารถประยุกต์ให้อุปกรณ์ต่างๆเชื่อมต่อกันผ่านเครือข่ายของอินเทอร์เน็ตได้
- ❑ เราสามารถสร้างสรรค์โครงการที่เกี่ยวกับการติดตามอุปกรณ์ เช่น มอนิเตอร์อุปกรณ์ผ่านอินเทอร์เน็ต ควบคุมอุปกรณ์ผ่านอินเทอร์เน็ต เป็นต้น



MQTT

❑ MQTT ประกอบด้วย 2 ส่วน คือ

1. **MQTT Broker หรือ MQTT Server** เป็นซอฟต์แวร์สำหรับรับข้อมูลจาก MQTT Client ที่ได้ publish เข้ามาและสามารถ publish ข้อมูลจาก MQTT Broker ไปยัง MQTT Client ที่ได้ Subscribe ข้อมูลไว้ได้ หากมองในมุมมองของ Internet of Things อุปกรณ์นี้อาจจะเป็น Cloud Server ของค่ายต่างๆ เช่น CloudMQTT, NETPIE, Azure, AWS เป็นต้น หรือใช้ Single Board Computer เช่นบอร์ด Raspberry Pi, LattePanda, Beagle Bone, nanoPi, อื่นๆ แล้วติดตั้งซอฟต์แวร์เพิ่มเติมก็สามารถใช้งานได้เช่นกัน
2. **MQTT Client** เป็นส่วน publish ข้อมูลต่างๆ ขึ้นไปยัง MQTT Broker และสามารถ Subscribe ข้อมูลต่างๆจาก MQTT Broker ผ่านทาง TCP/IP Protocol ถ้ามองในมุมมองของ Internet of Things (IoT) อุปกรณ์จำพวกนี้จะเป็น Device ที่สามารถเชื่อมต่อกับระบบเครือข่ายได้ เช่น บอร์ด Arduino Uno Wifi 2, Arduino MKR Wifi 1010, บอร์ด ESP32, บอร์ด ESP8266, บอร์ด Raspberry Pi, เว็บไซท์, สมาร์ทโฟน และ คอมพิวเตอร์



10 Free Public MQTT Brokers(Private & Public)

Public MQTT Brokers

Name	Broker Address	TCP Port	TLS Port	WebSocket Port	Message Retention	Persistent Session	Sign Up Required	Link
Eclipse	mqtt.eclipse.org	1883	N/A	80, 443	YES	YES	NO	Link
Mosquitto	test.mosquitto.org	1883	8883, 8884	80	YES	YES	NO	Link
HiveMQ	broker.hivemq.com	1883	N/A	8000	YES	YES	NO	Link
Flespi	mqtt.flespi.io	1883	8883	80, 443	YES	YES	YES	Link
Dioty	mqtt.dioty.co	1883	8883	8080, 8880	YES	YES	YES	Link
Fluux	mqtt.fluux.io	1883	8883	N/A	N/A	N/A	NO	Link
EMQX	broker.emqx.io	1883	8883	8083	YES	YES	NO	Link

Private MQTT Brokers

Name	TCP Port	TLS Port	WebSocket Port	Message Retention	Persistent Session	QoS Levels	Free Limits	Link
Azure	NO	8883	443	NO	Limited	0, 1	8000 messages/day	Link
AWS	NO	8883	443	NO	Limited	0, 1	250,000/month	Link
CloudMQTT	Custom Port	Custom Port	Custom Port	NOT SURE	YES	0, 1, 2	5 Connections & 10 Kbit/s	Link

Mosquitto MQTT

- ❑ Eclipse Mosquitto is an open source (EPL/EDL licensed) message broker that implements the MQTT protocol versions 5.0, 3.1.1 and 3.1. Mosquitto is lightweight and is suitable for use on all devices from low power single board computers to full servers.
- ❑ one of the most famous MQTT broker.
- ❑ Its very easy to install and easy to use.
- ❑ It is lightweight and is suitable for use on all devices from low power single board computers to full servers.



<https://mosquitto.org/>



Mosquitto Installation

- ❑ เข้าสู่เว็บไซต์ <https://mosquitto.org/download/>
- ❑ เลือก Download ตามระบบปฏิบัติการที่ต้องการ ในบทเรียนนี้ให้ติดตั้งเวอร์ชันสำหรับ Windows

Source

- [mosquitto-2.0.10.tar.gz](#) (GPG signature)
- [Git source code repository](#) ([github.com](#))

Older downloads are available at <https://mosquitto.org/files/>

Binary Installation

The binary packages listed below are supported by the Mosquitto project. In many cases Mosquitto is also available directly from official Linux/BSD distributions.

Windows

- [mosquitto-2.0.10-install-windows-x64.exe](#) (64-bit build, Windows Vista and up, built with Visual Studio Community 2019)
- [mosquitto-2.0.10-install-windows-x32.exe](#) (32-bit build, Windows Vista and up, built with Visual Studio Community 2019)

Older installers can be found at <https://mosquitto.org/files/binary/>.

See also [README-windows.md](#) after installing.

Mac

Mosquitto can be installed from the homebrew project. See [brew.sh](#) and then use `brew install mosquitto`

Linux distributions with snap support

- `snap install mosquitto`

Debian

- Mosquitto is now in Debian proper. There will be a short delay between a new release and it appearing in Debian as part of the normal Debian procedures.
- There are also Debian repositories provided by the mosquitto project, as described at <https://mosquitto.org/2013/01/mosquitto-debian-repository>

Raspberry Pi

Mosquitto is available through the main repository.

There are also Debian repositories provided by the mosquitto project, as described at <https://mosquitto.org/2013/01/mosquitto-debian-repository/>

Mosquitto Installation

- ❑ หลังจากทำการติดตั้งแล้วให้เปิด CMD โดย Run as administrator
- ❑ จากนั้นพิมพ์คำสั่ง **net start mosquitto** เพื่อเริ่มการทำงาน
- ❑ หากเริ่มการทำงานได้สำเร็จจะขึ้นข้อความว่า **The Mosquitto Broker service was started successfully.** ดังตัวอย่าง

```
C:\> Administrator: Command Prompt

Microsoft Windows [Version 10.0.19042.867]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>net start mosquitto

The Mosquitto Broker service was started successfully.

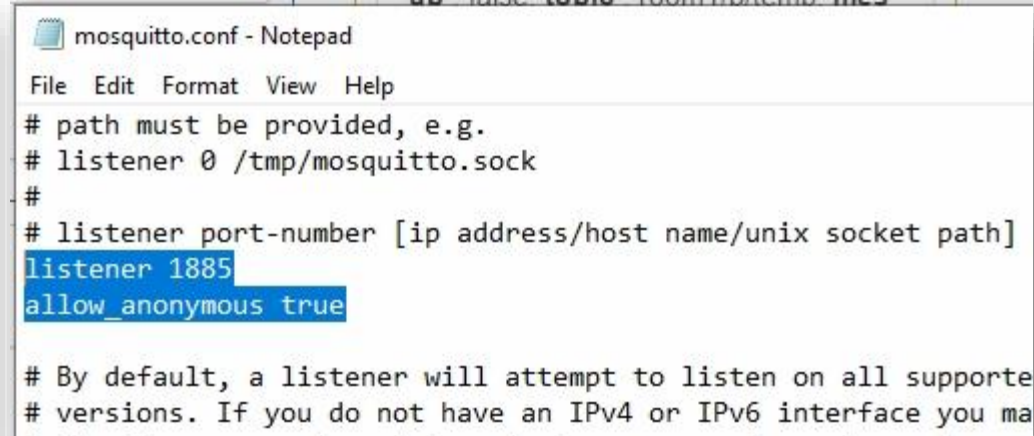
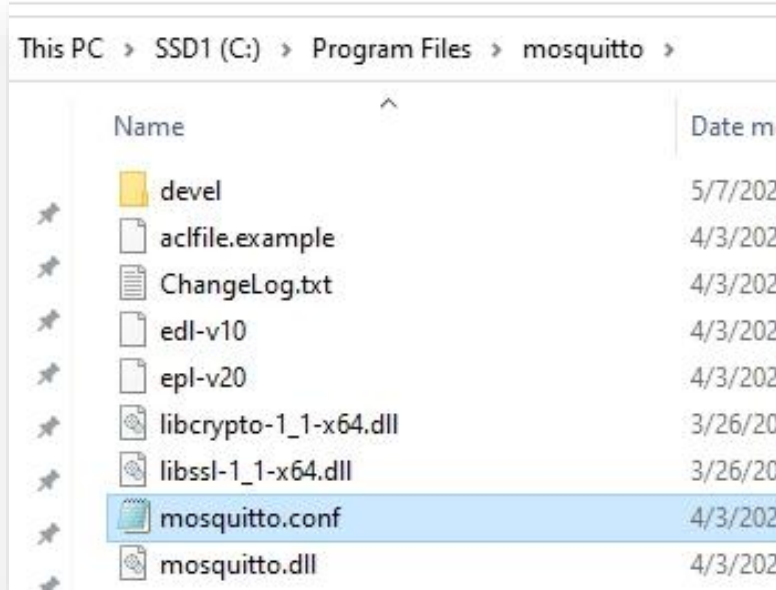
C:\WINDOWS\system32>
```

- ❑ ในการเริ่มต้นการทำงานนี้ เป็นการตั้งค่าทุกอย่างเป็น default
- ❑ หากต้องการเปลี่ยนแปลงค่า ให้เข้าไปแก้ไขไฟล์ที่ชื่อว่า mosquitto.conf
- ❑ รายละเอียดการกำหนดค่าสามารถศึกษาเพิ่มเติมได้จาก
- ❑ <https://mosquitto.org/man/mosquitto-conf-5.html#>



การเปลี่ยน default port

- ❑ Default Port ที่สามารถใช้งานได้คือ 1883 ซึ่งหาก Server มีการติดตั้ง MQTT Broker อื่นเช่น Thingsboard จำเป็นต้องมีการเปลี่ยน port ไม่ให้ซ้ำกัน
- ❑ สามารถเปลี่ยน port ได้ตามต้องการ โดยเข้าไปที่ mosquitto.conf โดยต้องเปิดโปรแกรม Notepad ด้วย administrator
- ❑ จากนั้นค้นหาคำว่า #listener ทำการลบเครื่องหมาย # ออก และเติมหมายเลข port ที่ต้องการลงไปเช่น 1885
- ❑ เพิ่มเติมคำว่า allow_anonymous true
- ❑ ทำการบันทึกไฟล์
- ❑ จากนั้นเริ่มต้นการทำงาน mosquitto อีกครั้ง
 - net stop mosquitto เพื่อหยุดการทำงาน
 - net start mosquitto เพื่อเริ่มการทำงานใหม่อีกครั้ง



ตัวอย่างการสื่อสารเบื้องต้นของ MQTT

□ ในกรณีต้องการมอนิเตอร์ค่าของเซ็นเซอร์ผ่านมือถือ มีขั้นตอนคร่าวๆ ดังนี้



1. กำหนดให้ Client Device ในตัวอย่างนี้จะเป็นสมาร์ทโฟน ทำการ Subscribe MQTT Broker ตาม Topic ที่ต้องการ ตัวอย่างนี้ติดตาม Topic "temp" ไว้



2. กำหนดให้ Client Device (Device ที่ต่อกับเซ็นเซอร์วัดอุณหภูมิ) แล้วทำการ Publish ค่าขึ้นไปยัง MQTT Broker



3. MQTT Broker Publish ไปยังอุปกรณ์ที่ Subscribe ไว้ ดังนั้นในตัวอย่างนี้สมาร์ทโฟนจะได้รับข้อมูลจาก Client ที่ต่ออยู่กับเซ็นเซอร์เรียบร้อยแล้ว



เส้นทาง (Topic)

- ❑ Topic หรือ เส้นทาง เปรียบเหมือนกับหัวข้อ หรือห้องแชทที่ต้องการจะคุย
- ❑ การคุยกันจะมีเฉพาะอุปกรณ์ที่อยู่ในห้องนั้น ๆ (Subscribe) ถึงจะสามารถได้รับข้อมูลที่มีการส่งไปในห้องนั้น ๆ
- ❑ ที่ถูกเรียกว่าเส้นทางเนื่องจากการใช้งานส่งข้อมูลและรับข้อมูลจะเหมือนกับเส้นทางในระบบไฟล์ เช่น /Room1/LED
- ❑ ซึ่งระบบเส้นทางนี้นอกจากอุปกรณ์จะสามารถรอการสนทนาในห้องตามเส้นทาง /Room1/LED ได้แล้ว ยังสามารถรอสนทนาเส้นทาง /Room1 ได้ด้วย หากเป็นการรอฟังในเส้นทาง (Subscribe) /Room1 จะหมายถึงการส่งข้อมูลใด ๆ ที่นำหน้าด้วย /Room1 เช่น /Room1/LED , /Room1/Value ผู้ที่รอฟัง (Subscribe) /Room1 อยู่จะได้รับข้อมูลเหล่านั้นด้วย

คุณภาพข้อมูล (QoS)

- ❑ แบ่งออกเป็น 3 ระดับดังนี้
- ❑ QoS0 – ส่งข้อมูลเพียงครั้งเดียว ไม่สนใจว่าผู้รับจะได้รับหรือไม่
- ❑ QoS1 – ส่งข้อมูลเพียงครั้งเดียว ไม่สนใจว่าผู้รับจะได้รับหรือไม่ แต่ให้ค่าที่ส่งล่าสุดไว้ เมื่อมีการเชื่อมต่อใหม่จะได้รับข้อมูลครั้งล่าสุดอีกครั้ง
- ❑ QoS2 – ส่งข้อมูลหลาย ๆ ครั้งจนกว่าปลายทางจะได้รับข้อมูล มีข้อเสียที่สามารถทำงานได้ช้ากว่า QoS0 และ QoS1

การส่งข้อมูล (Publish)

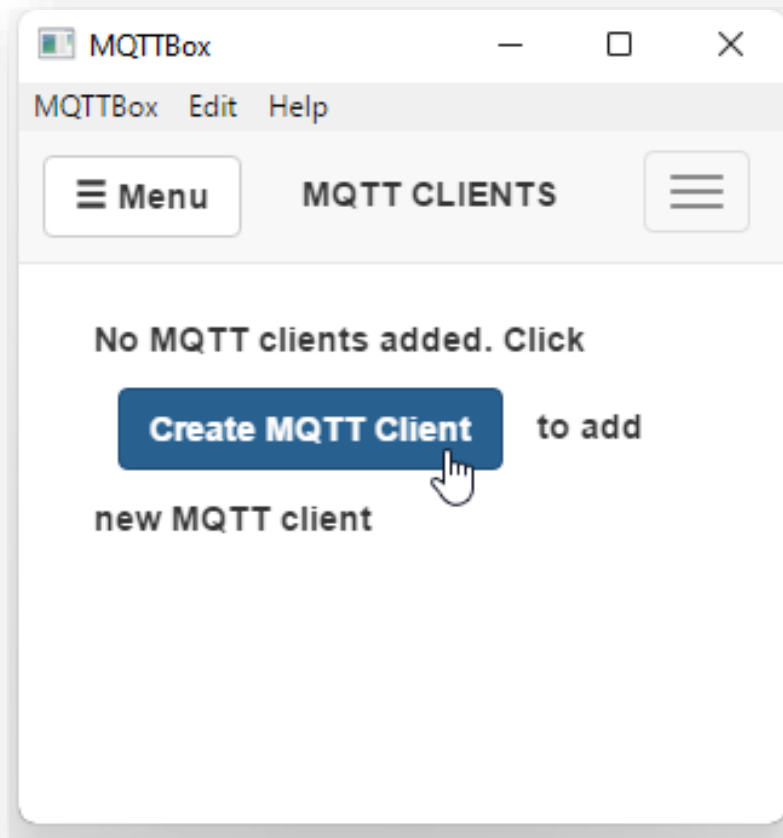
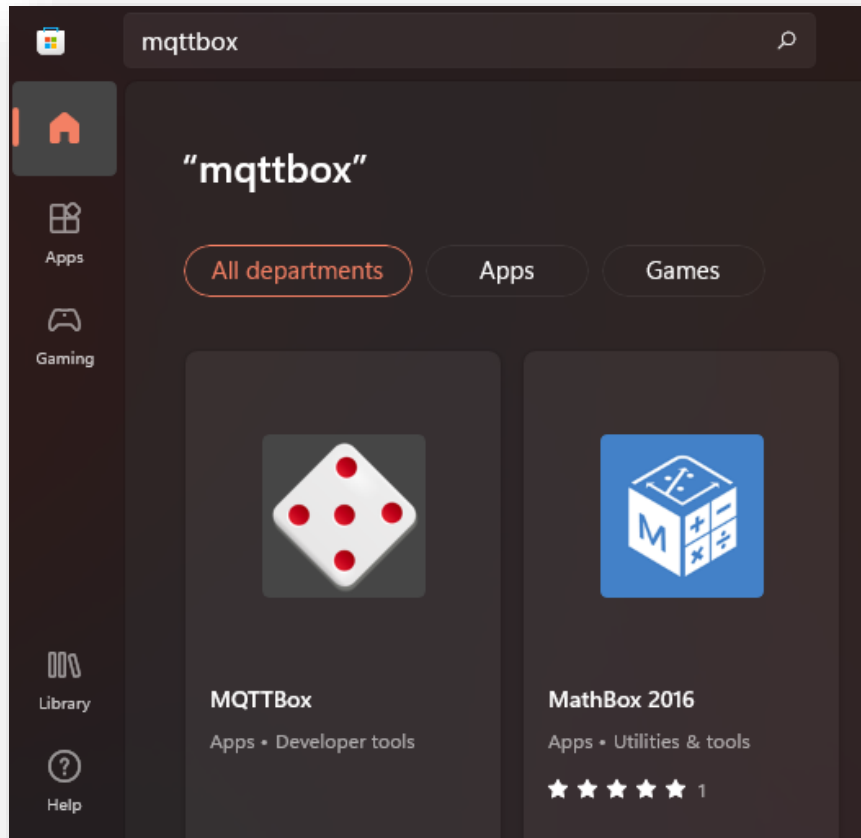
การส่งข้อมูลในแต่ละครั้งจะต้องประกอบไปด้วยเส้นทาง (Topic) ข้อมูล และคุณภาพข้อมูล ซึ่งการส่งข้อมูลจะเรียกว่า **Publish**

การรับข้อมูล (Subscribe)

- การรับข้อมูลในระบบ MQTT เรียกว่า **Subscribe**
- จะรับข้อมูลได้เฉพาะเมื่อมีการเรียกใช้การ Subscribe ไปยัง Topic ที่กำหนด อาจเปรียบได้กับการ Subscribe คือการเข้าไปนั่งรอเพื่อนในกลุ่ม Line ส่งแชทมาหา เมื่อมีการส่งข้อมูลเข้ามาจะเกิดสิ่งที่เรียกว่าเหตุการณ์ (Event) ให้เรากดเข้าไปดูข้อความที่เพื่อน ๆ ส่งเข้ามา
- จะเห็นได้ว่า MQTT ก็เปลี่ยนเสมือนห้องแชทของอุปกรณ์ที่จะสนทนาแลกเปลี่ยนข้อมูลกันแบบเรียลไทม์ผ่านเครือข่ายอินเทอร์เน็ต

ทดสอบการเชื่อมต่อ ด้วย MQTT box

- ❑ ทำการติดตั้ง MQTT box ตามขั้นตอนดังนี้
- ❑ เข้าไปที่ Microsoft Store ทำการค้นหาคำว่า mqtt box และติดตั้งโปรแกรม
- ❑ เปิด Application จากนั้น กด Create MQTT Client
- ❑ จากนั้นทำการตั้งค่าดังตัวอย่างในหน้าถัดไป



ทดสอบการเชื่อมต่อ ด้วย MQTT box

- ☐ Create MQTT client โดยกำหนดการเชื่อมต่อดังนี้
 - ☐ กำหนดชื่อ MQTT Client Name ตามต้องการ
 - ☐ เลือก Protocol mqtt/tcp
 - ☐ ใช้ host "electsut.trueddns.com" port "27860"
- ☐ กด save ถ้าเชื่อมต่อได้สำเร็จ จะแสดงคำว่า **Connected** สีเขียวดังรูป
- ☐ MQTT broker นี้ เป็น broker ที่จัดไว้ให้นักศึกษาจึงสามารถใช้งานได้ทุกที่
- ☐ หากตั้ง broker เอง จะสามารถใช้ได้ภายในเครือข่ายเดียวกันเท่านั้น เช่น ภายในบ้าน
- ☐ หากต้องการใช้ภายนอก ต้องทำการ forward port และตั้งค่าเพิ่มเติม

MQTT CLIENT SETTINGS

MQTT Client Name: n10

MQTT Client Id: b31c2388-70e7-4412-9

Append timestamp to MQTT client id? ☒ Yes

Broker is MQTT v3.1.1 compliant? ☒ Yes

Protocol: mqtt / tcp

Host: electsut.trueddns.com:27860

Clean Session? ☒ Yes

Auto connect on app launch? ☒ Yes

Username: Username

Password: Password

Reschedule Pings? ☒ Yes

Queue outgoing QoS zero messages? ☒ Yes

Reconnect Period (milliseconds): 1000

Connect Timeout (milliseconds): 30000

KeepAlive (seconds): 10

Will - Topic: Will - Topic

Will - QoS: 0 - Almost Once

Will - Retain: ☐ No

Will - Payload:

Save Delete

Connected

Topic to publish: Topic to publish

QoS: 0 - Almost Once

Retain: ☐

Payload Type: Strings / JSON / XML / Characters

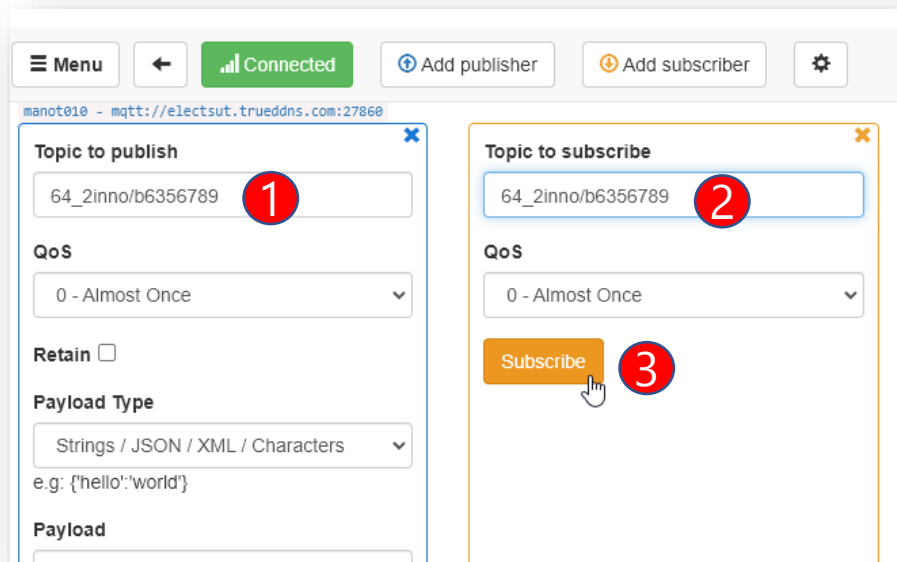
e.g: {'hello':'world'}

Payload:

Publish

ทดสอบ publish subscribe

- ❑ Create MQTT client โดยกำหนดการเชื่อมต่อดังนี้
 - ❑ กำหนดชื่อ MQTT Client Name ตามต้องการ
 - ❑ เลือก Protocol mqtt/tcp
 - ❑ ใช้ host "electsut.trueddns.com" port "27860"
- ❑ กด save ถ้าเชื่อมต่อได้สำเร็จ จะแสดงคำว่า **Connected** สีเขียวดังรูป
- ❑ MQTT broker นี้ ผู้สอนได้ติดตั้งไว้ให้นักศึกษาใช้ในรายวิชานี้ สามารถใช้งานเครือข่ายภายนอกเพื่อรับส่งข้อมูลไปที่ใดก็ได้
- ❑ หากตั้ง broker เอง จะสามารถใช้ได้ภายในเครือข่ายเดียวกันเท่านั้น เช่น ภายในบ้าน
- ❑ หากต้องการใช้ภายนอก ต้องทำการ forward port และตั้งค่าเพิ่มเติม



ทดสอบ publish subscribe

- ❑ ให้ทำการแก้ไข Topic to Publish และ Topic to Subscribe โดยใช้ topic 64_2/Bxxxxx โดยให้ใส่รหัสนักศึกษาของตนเอง
- ❑ จากนั้นกดปุ่ม Subscribe
- ❑ ให้ทำการ Publish โดยพิมพ์ข้อความว่า "hello" ในช่อง payload จากนั้นกด publish
- ❑ ถ้าทุกอย่างทำงานถูกต้องจะพบข้อความว่า "hello" ขึ้นมาทางฝั่ง subscribe

manot010 - mqtt://electsut.trueddns.com:27860

Topic to publish 64_2inno/b6356789 (1)

QoS 0 - Almost Once

Retain ☐

Payload Type Strings / JSON / XML / Characters

e.g: {'hello':'world'}

Payload

Topic to subscribe 64_2inno/b6356789 (2)

QoS 0 - Almost Once

Subscribe (3)

manot010 - mqtt://electsut.trueddns.com:27860

Topic to publish 64_2inno/b6356789

QoS 0 - Almost Once

Retain ☐

Payload Type Strings / JSON / XML / Characters

e.g: {'hello':'world'}

Payload hello (4)

Publish (5)

hello

topic:64_2inno/b6356789, qos:0, retain:false

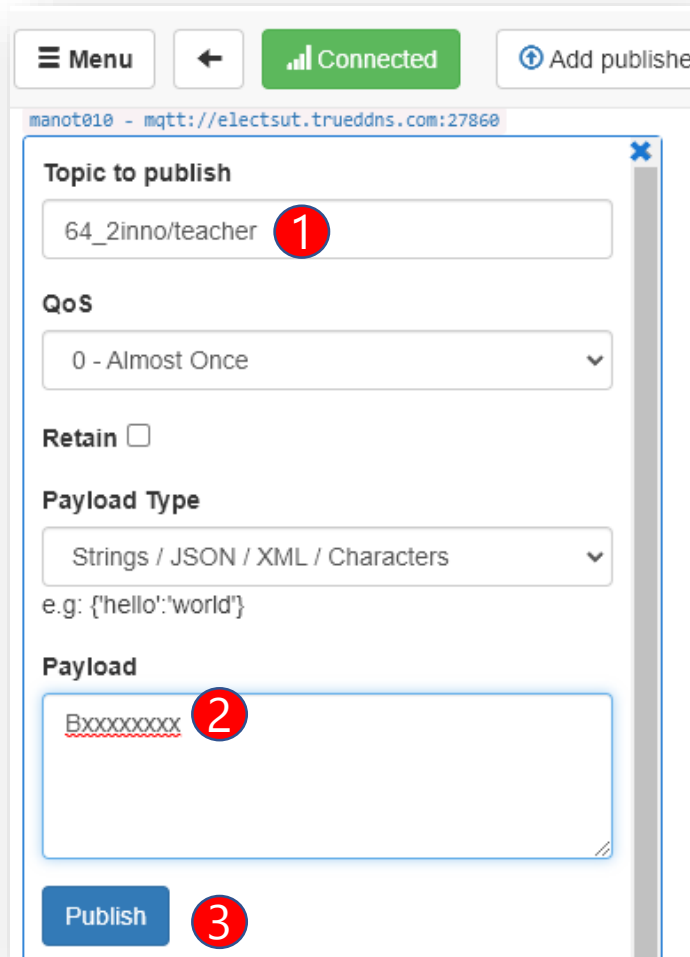
Message received on topic 64_2inno/b6356789 (6)

hello

qos : 0, retain : false, cmd : publish, dup : false, topic : 64_2inno/b6356789, messageid : , length : 27

ตัวอย่างการ publish subscribe

- ให้นักศึกษา publish ข้อความผ่านมาทาง topic "64_2inno/teacher" โดยส่งข้อความมาเป็นรหัสนักศึกษา



manot010 - mqtt://electsut.trueddns.com:27860

Topic to publish

64_2inno/teacher **1**

QoS

0 - Almost Once

Retain ☐

Payload Type

Strings / JSON / XML / Characters

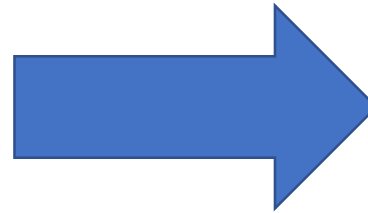
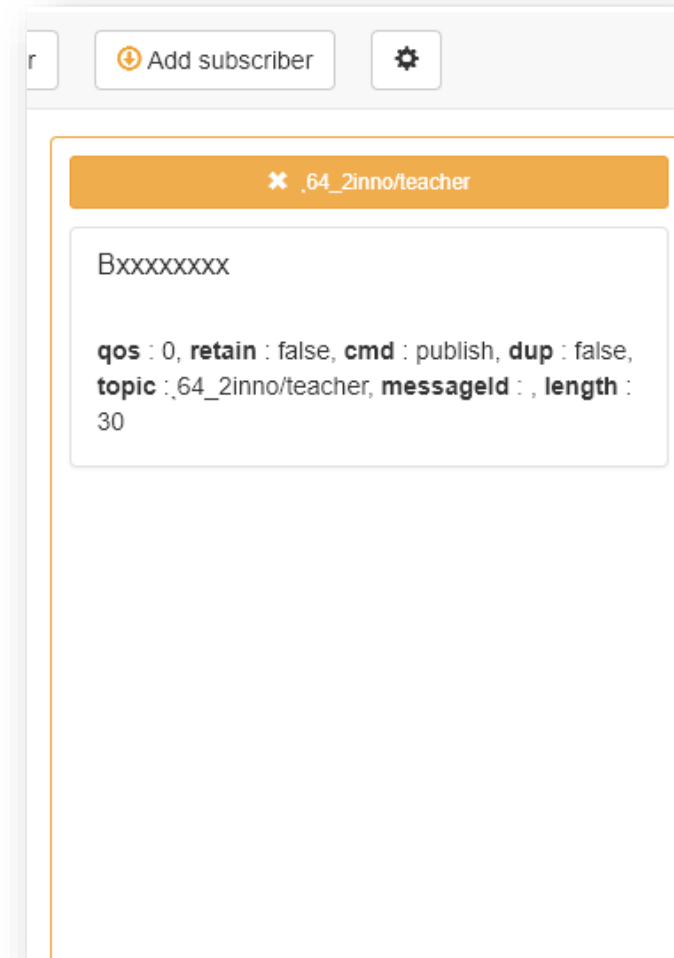
e.g: {"hello": "world"}

Payload

Bxxxxxxx **2**

Publish **3**

Publisher

Add subscriber

64_2inno/teacher

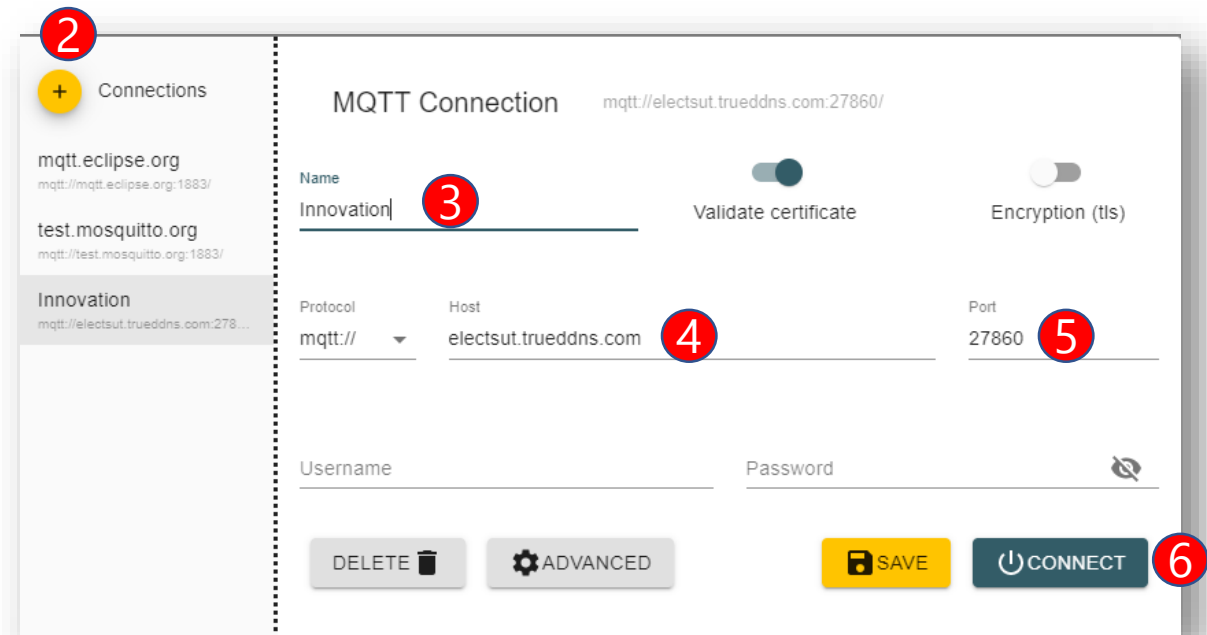
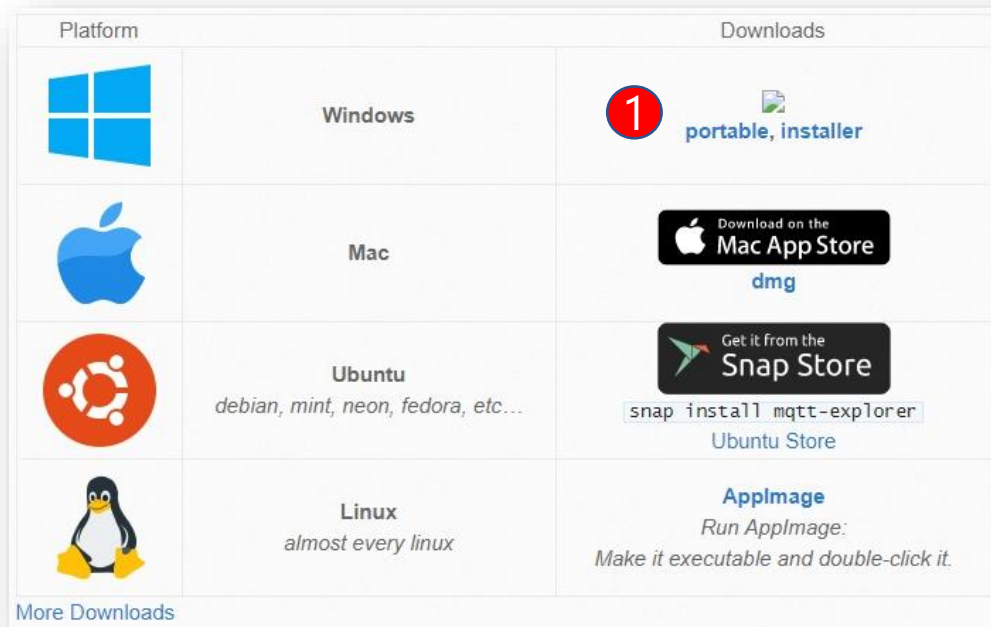
Bxxxxxxx

qos : 0, **retain** : false, **cmd** : publish, **dup** : false,
topic : 64_2inno/teacher, **msgid** : , **length** : 30

Subscriber

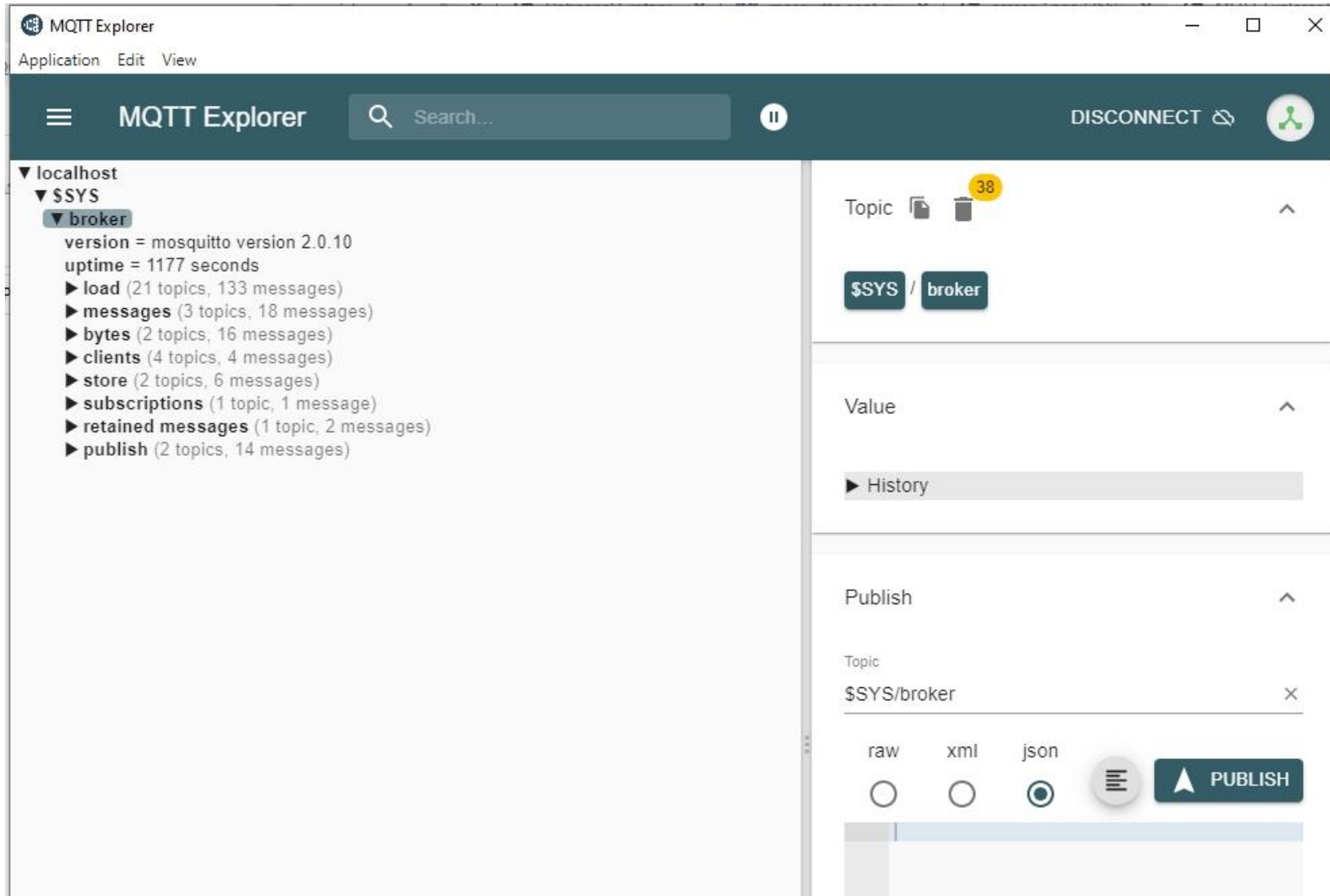
ทดสอบการเชื่อมต่อ ด้วย MQTT Explorer

- ❑ ทำการ download และติดตั้ง MQTT Explorer โดยเข้าไปที่ <http://mqtt-explorer.com/>
- ❑ เลือกระบบปฏิบัติการที่ต้องการ
- ❑ ในขั้นตอนนี้ เลือกระบบปฏิบัติการ Windows ซึ่งสามารถใช้ได้ทั้งแบบ Portable หรือ Installer
- ❑ กด + เพื่อสร้าง connection ใหม่
- ❑ ทำการกำหนด Hostname และ Port ดังรูปจากนั้นกด connect



ทดสอบการเชื่อมต่อ ด้วย MQTT Explorer

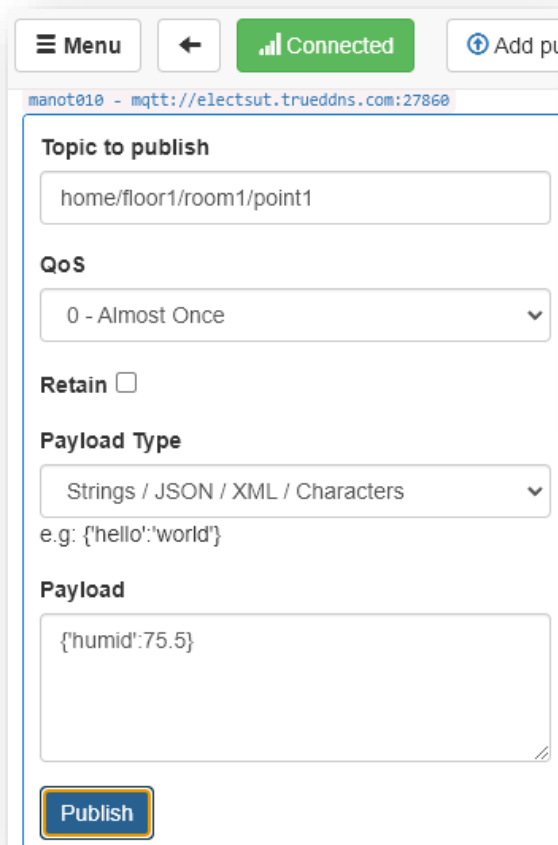
❑ หากเชื่อมต่อสำเร็จจะปรากฏหน้าต่างดังรูป



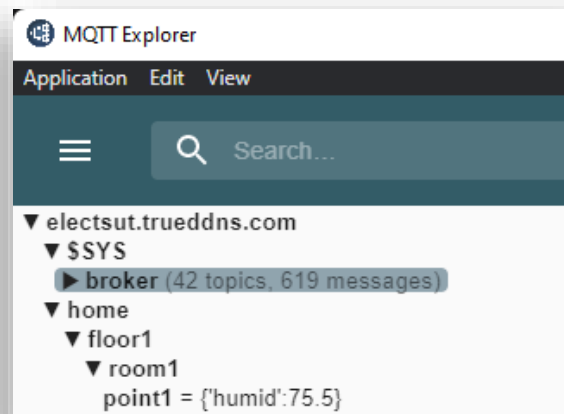
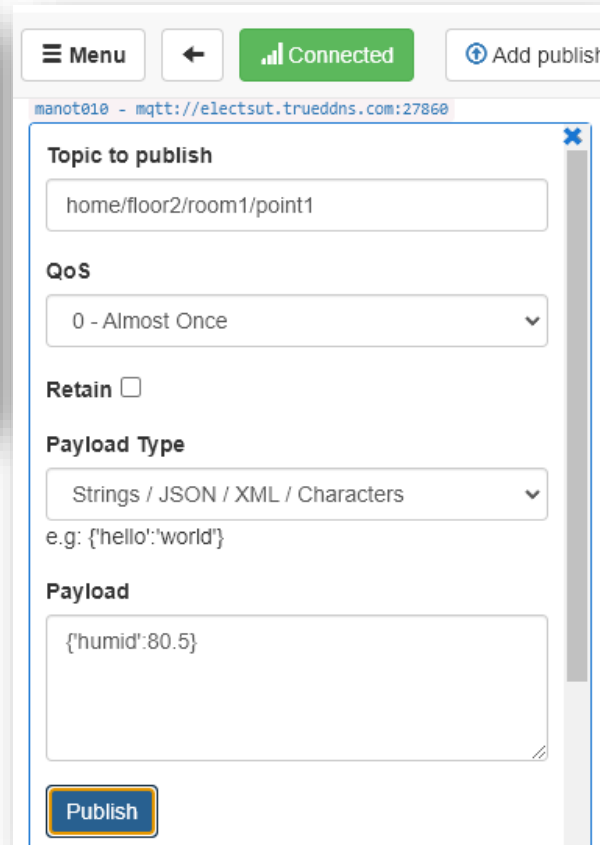
1 5
2 4
3

ทดสอบการ publish /subscribe

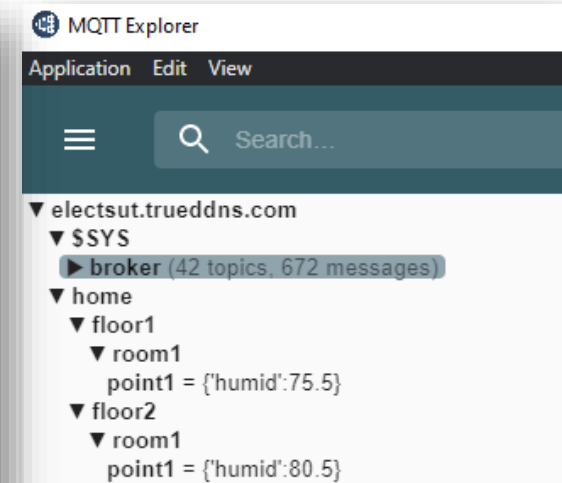
- ❑ ทำการทดสอบ publish ข้อมูลผ่านทาง MQTT box ไปยัง broker ด้วย topic "home/floor1/room1/point1"
- ❑ ส่งข้อความ payload ในรูปแบบของ Json format {'humid':75.5} ดังตัวอย่าง
- ❑ payload ดังกล่าว จะปรากฏขึ้นที่ MQTT Explorer
- ❑ ทดลองเปลี่ยน topic เป็น "home/floor2/room1/point1"



MQTT Explorer interface showing the Publish form. The status bar indicates 'Connected'. The 'Topic to publish' field contains 'home/floor1/room1/point1'. The 'QoS' dropdown is set to '0 - Almost Once'. The 'Retain' checkbox is unchecked. The 'Payload Type' dropdown is set to 'Strings / JSON / XML / Characters'. The 'Payload' field contains the JSON string {'humid':75.5}. A 'Publish' button is at the bottom.

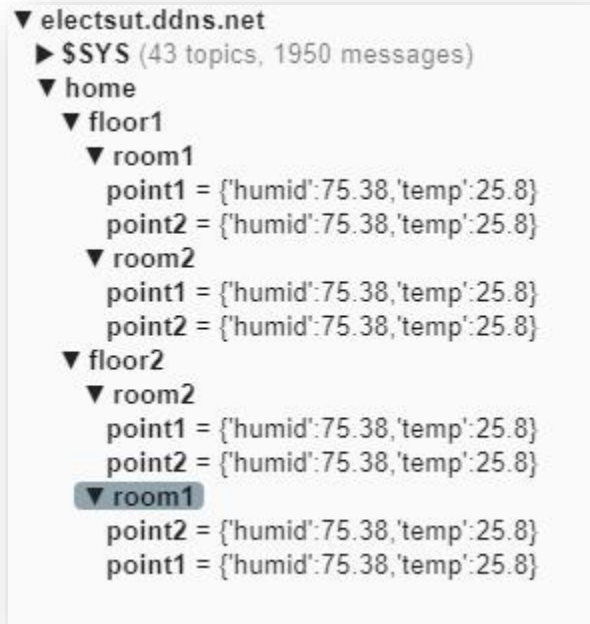



MQTT Explorer interface showing the Publish form. The status bar indicates 'Connected'. The 'Topic to publish' field contains 'home/floor2/room1/point1'. The 'QoS' dropdown is set to '0 - Almost Once'. The 'Retain' checkbox is unchecked. The 'Payload Type' dropdown is set to 'Strings / JSON / XML / Characters'. The 'Payload' field contains the JSON string {'humid':80.5}. A 'Publish' button is at the bottom.

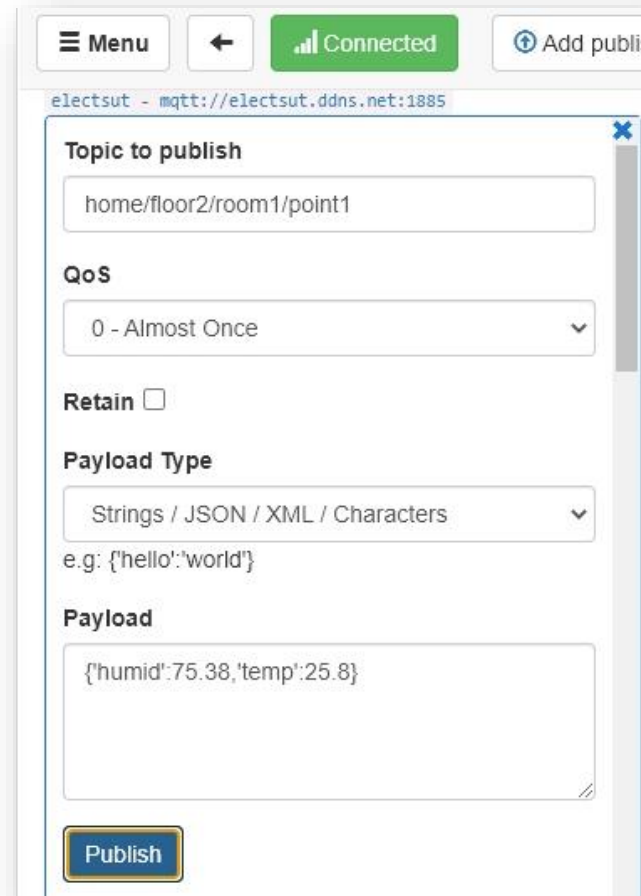


ทดสอบการ publish /subscribe ผ่าน server

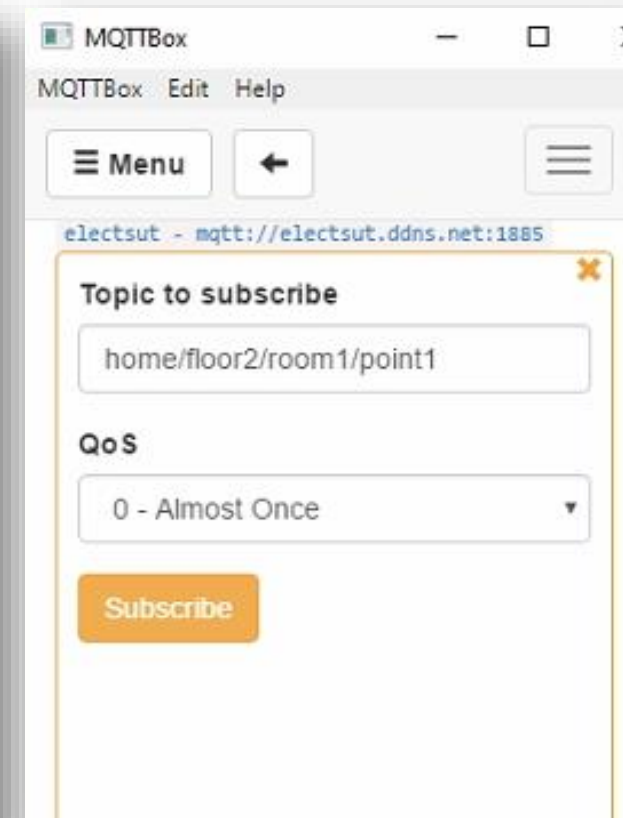
- ❑ สามารถรับข้อมูลที่ต้องการ ด้วยการ subscribe ไปยัง topic ที่ต้องการ
- ❑ ในที่นี้สมมติให้ในบ้านมี 2 ชั้นแต่ละชั้นมี 2 ห้องและแต่ละห้องมีเซ็นเซอร์จำนวน 2 จุด แต่ละจุดมีการวัดค่าจำนวน 2 ค่า ได้แก่ความชื้นและอุณหภูมิ



- ❑ ให้นักศึกษาทดลองทำการ Subscribe ข้อมูลที่อาจารย์ส่งไป



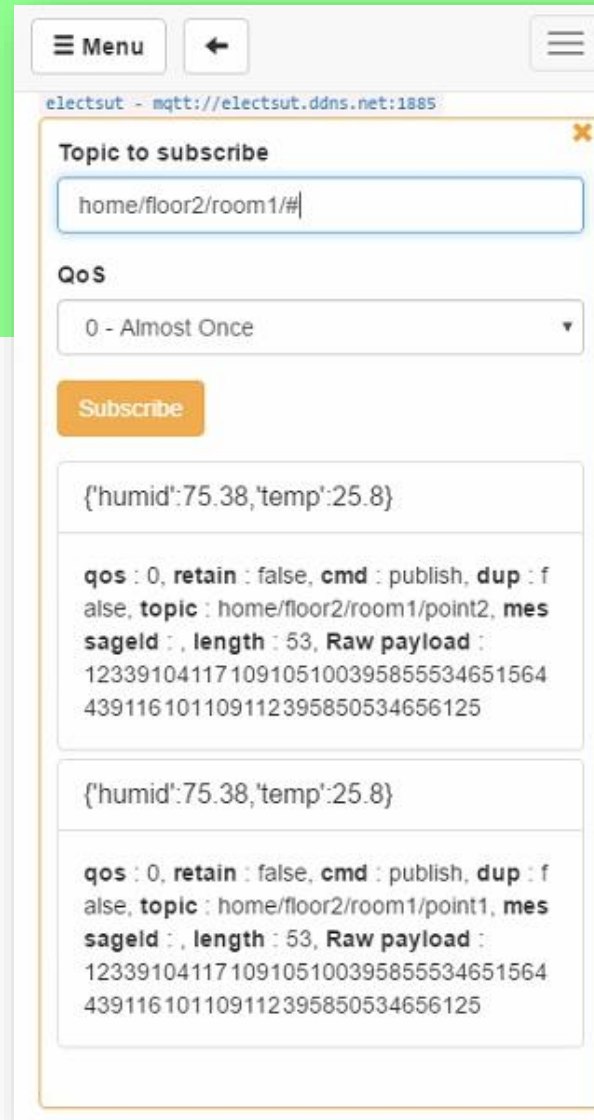
publisher



subscriber

การใช้เครื่องหมาย # ในการ subscribe

- ❑ การใช้เครื่องหมาย # จะเป็นการ subscribe ข้อมูลที่อยู่ใน topic ย่อยหลัง topic หน้าเครื่องหมาย # ทั้งหมด
- ❑ ถ้าต้องการข้อมูล ชั้น 2 ห้อง 1 **point1**
 - home/floor2/room1/point1
- ❑ ถ้าต้องการข้อมูล ชั้น 2 ห้อง 1 **point2**
 - home/floor2/room1/point2
- ❑ ถ้าต้องการข้อมูล ชั้น 2 ห้อง 1 **ทั้งสองจุด**
 - home/floor2/room1/#



```

▼ electsut.ddns.net
  ► $SYS (43 topics, 1950 messages)
  ▼ home
    ▼ floor1
      ▼ room1
        point1 = {'humid':75.38,'temp':25.8}
        point2 = {'humid':75.38,'temp':25.8}
      ▼ room2
        point1 = {'humid':75.38,'temp':25.8}
        point2 = {'humid':75.38,'temp':25.8}
    ▼ floor2
      ▼ room2
        point1 = {'humid':75.38,'temp':25.8}
        point2 = {'humid':75.38,'temp':25.8}
      ▼ room1
        point2 = {'humid':75.38,'temp':25.8}
        point1 = {'humid':75.38,'temp':25.8}
  
```


การใช้เครื่องหมาย + ในการ subscribe

- ❑ การใช้เครื่องหมาย + จะเป็นการ subscribe ข้อมูลที่อยู่ใน level ที่แทนด้วยเครื่องหมาย +
- ❑ ถ้าต้องการข้อมูล **ชั้น 2** ห้อง 1 point1
 - home/floor2/room1/point1
- ❑ ถ้าต้องการข้อมูล **ชั้น 1** ห้อง 1 point1
 - home/floor1/room1/point1
- ❑ ถ้าต้องการข้อมูล **ทั้งสองชั้น** ห้อง 1 point1
 - home/+/room1/point1

```

▼ electsut.ddns.net
▶ $SYS (43 topics, 1950 messages)
▼ home
  ▼ floor1
    ▼ room1
      point1 = {'humid':75.38,'temp':25.8}
      point2 = {'humid':75.38,'temp':25.8}
    ▼ room2
      point1 = {'humid':75.38,'temp':25.8}
      point2 = {'humid':75.38,'temp':25.8}
  ▼ floor2
    ▼ room2
      point1 = {'humid':75.38,'temp':25.8}
      point2 = {'humid':75.38,'temp':25.8}
    ▼ room1
      point2 = {'humid':75.38,'temp':25.8}
      point1 = {'humid':75.38,'temp':25.8}
  
```

