



การทดลองที่ **1** ติดตั้ง MQTT Broker

ในที่นี่จะใช้ MQTT Broker ของ Mosquitto



Mosquitto

An Open Source MQTT v3.1/v3.1.1/v5.0 Broker

ในการติดตั้ง MQTT Broker จะติดตั้ง package 2 ตัวดังนี้

- **mosquitto** is the MQTT broker (i.e. server)
- **mosquitto-clients** are the command-line clients

Repository

- โดยปกติ เวลาเราติดตั้งโปรแกรมบน **Windows** คือเราจะต้องโหลดโปรแกรมมาติดตั้งเอง จากเว็บไหนก็ได้ ซึ่งบางทีอาจได้ของแถมมาด้วย แต่การติดตั้งแอปพลิเคชันบน **Ubuntu** หรือ **OS** ตระกูล **Linux** จะต่างกัน โดย **Ubuntu** จะใช้การติดตั้งซึ่งดึงไฟล์แอปพลิเคชันจากเว็บไซต์โดยตรง ซึ่งเราเรียก เว็บที่เก็บโปรแกรมพวกนั้นว่า **repository** ซึ่งเวลาที่มีนักพัฒนาแอปพลิเคชันส่งขึ้นไปบนเว็บเหล่านี้จะมีการตรวจสอบก่อนที่จะปล่อยให้ผู้ใช้ทั่วไปติดตั้ง ซึ่งโดยปกติแอปพลิเคชันที่ถูกส่งไปบน **repository** มักจะเป็นโปรแกรมที่สมบูรณ์ ไม่ใช่เวอร์ชัน **Alpha, Beta** และเป็นโปรแกรมที่ถูกรวมเข้าไปใน **default repository**
- ที่มา <http://www.ubuntuthailand.com/>

การติดตั้ง package แบบ online

- ดาวน์โหลด **key** ในการ **install**
- โหลดแล้วให้ใช้ **key** ด้วยคำสั่ง **`$sudo apt-key add file.key`**
- **apt-get** คือ การติดตั้ง **package** แบบออนไลน์ จาก **Server** ที่ให้บริการได้โดยตรง
- ไฟล์ที่ระบุ **Server** ปลายทางที่ให้บริการ นั่นก็คือไฟล์ **`/etc/apt/sources.list`**
- ก่อนที่จะติดตั้ง **package** ควรจะเรียกใช้คำสั่ง **`$sudo apt-get update`** เพื่อ **update**ฐานข้อมูลของ **package** ให้เป็นปัจจุบัน
- ติดตั้ง **package** โดยใช้คำสั่ง **`$sudo apt-get install ชื่อ package`**

การทดลองที่ 1 ติดตั้ง MQTT Broker

1. add key repository ของ mosquitto โดยใช้คำสั่ง

```
$wget http://repo.mosquitto.org/debian/mosquitto-repo.gpg.key
```

```
$sudo apt-key add mosquitto-repo.gpg.key
```

2. เข้าไฟล์เดอร์ source.list.d เพื่อเตรียมติดตั้ง repository ใหม่โดยใช้คำสั่ง

```
$cd /etc/apt/sources.list.d/
```

การทดลองที่ 1 ติดตั้ง MQTT Broker

3. สั่ง update apt information โดยใช้คำสั่ง

```
$sudo apt-get update
```

4. ติดตั้ง package mosquitto โดยใช้คำสั่ง

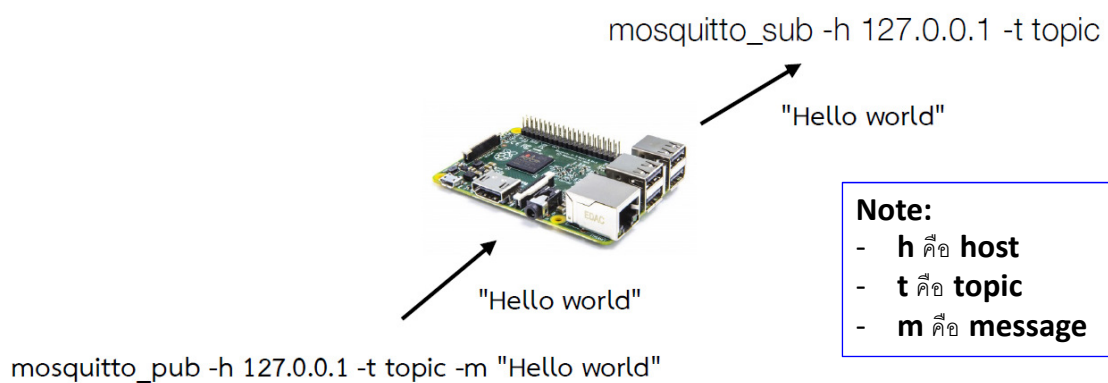
```
$sudo apt-get install mosquitto
```

5. ติดตั้ง package mosquitto-clients โดยใช้คำสั่ง

```
$sudo apt-get install mosquitto-clients
```

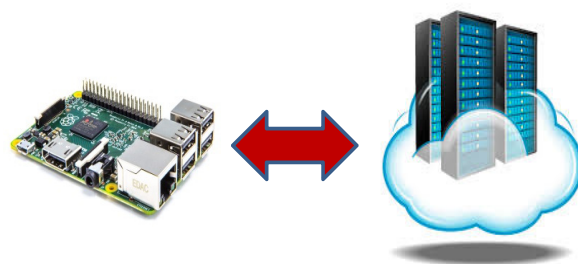
การทดลองที่ 2 ทดสอบการทำงานของ MQTT Broker

- ใช้โปรแกรม **putty** ทำการ **ssh** เพื่อเปิด **Terminal** หนึ่ง เพื่อทำการ **Subscribe** โดยใช้คำสั่ง
\$mosquitto_sub -h 127.0.0.1 -t topic
- และใช้โปรแกรม **putty** ทำการ **ssh** เพื่อเปิดอีก **Terminal** หนึ่ง เพื่อทำการ **Publish**
\$mosquitto_pub -h 127.0.0.1 -t topic -m "Hello world"
- เมื่อทำการ **Publish** แล้ว ให้สังเกตผลลัพธ์ที่หน้าจอ **Subscribe**



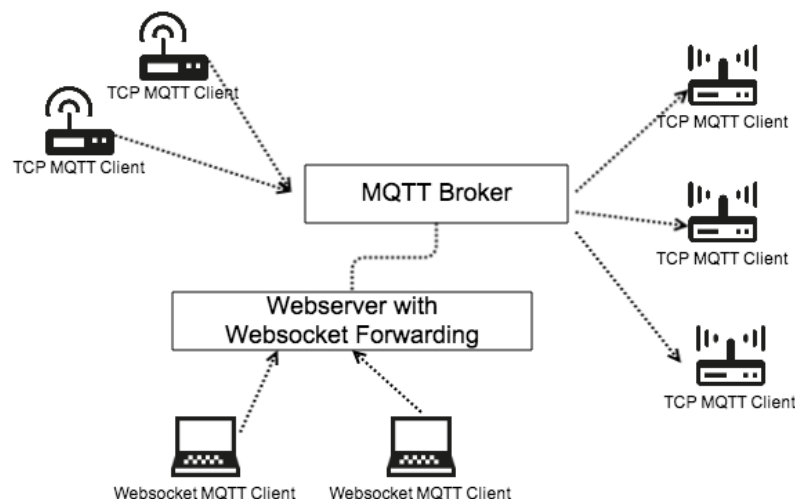
การทดลองที่ 3 การให้ Raspberry Pi เป็น Client

ให้ทำการทดลอง ให้ **Raspberry Pi** ทำการ **Subscribe** และ **publish**
ไปยัง **Broker** **broker.mqttdashboard.com, 1883**
โดยใช้ **command** **mosquitto_pub , mosquitto_sub**



broker.mqttdashboard.com, 1883

MQTT broker gets Websockets support



มีองค์ประกอบคือ

- Web Server ทำหน้าที่เก็บไฟล์ html ที่รองรับ websocket forwarding (javascript)
- Web Browser หรือ Web application ทำหน้าที่เรียกไฟล์ html จาก Web Server มาแสดงผล
- MQTT Broker ที่เปิด Web Socket เพื่อให้ Javascript ไปดึงข้อมูล มาแสดงผล

MQTT broker gets Websockets support

ติดตั้ง Websocket บน Raspberry Pi

เปิดไฟล์ mosquitto.conf โดยใช้คำสั่ง

```
$ sudo nano /etc/mosquitto/mosquitto.conf
```

แก้ไขไฟล์ โดยเพิ่มข้อความดังนี้

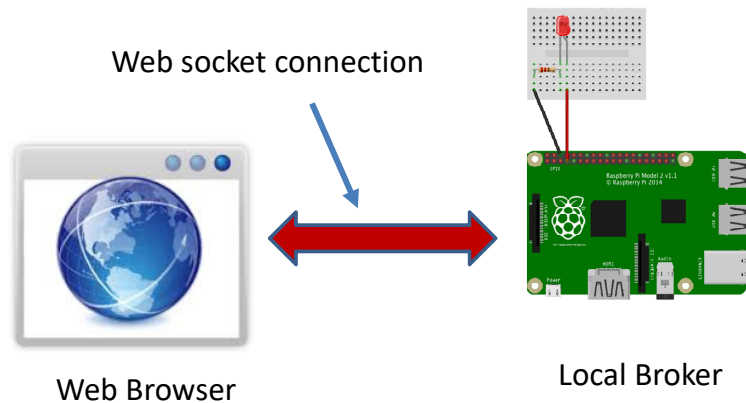
```
listener 9001
protocol websockets
```

Websocket ก็ใช้ port 9001 ในการติดต่อ
สามารถเปลี่ยนเป็น port อื่นๆ ได้

ทำการ Restart โดยใช้คำสั่ง

```
$ sudo /etc/init.d/mosquitto stop
$ sudo /etc/init.d/mosquitto start
```

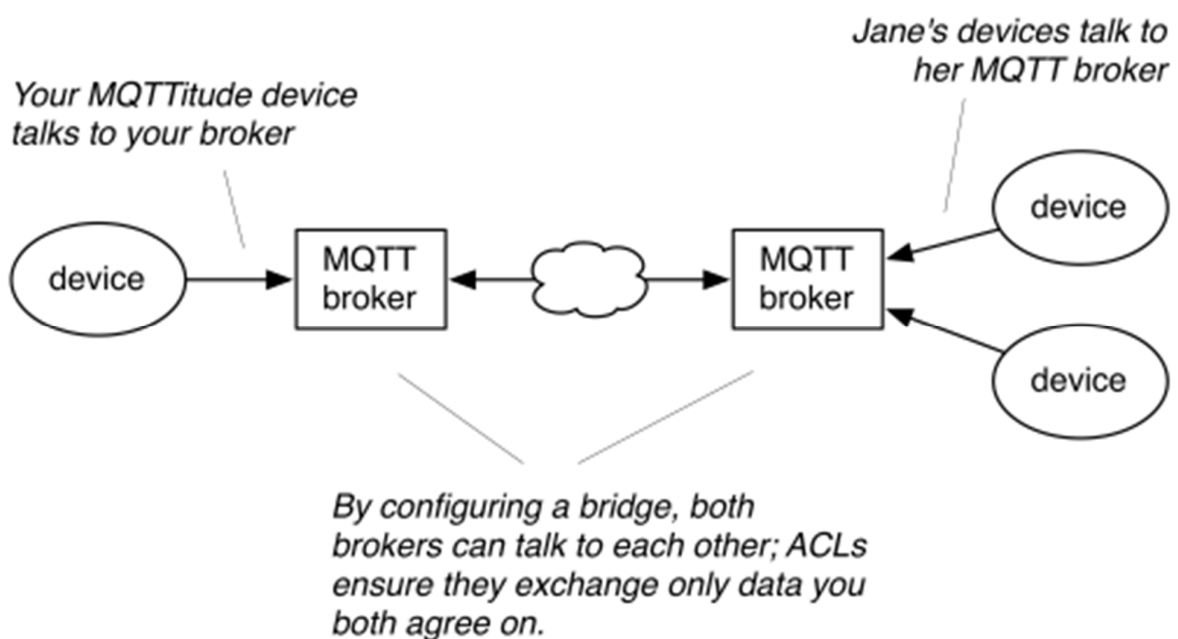
แบบฝึกหัด ทดสอบ Websocket ของ Rasp-pi



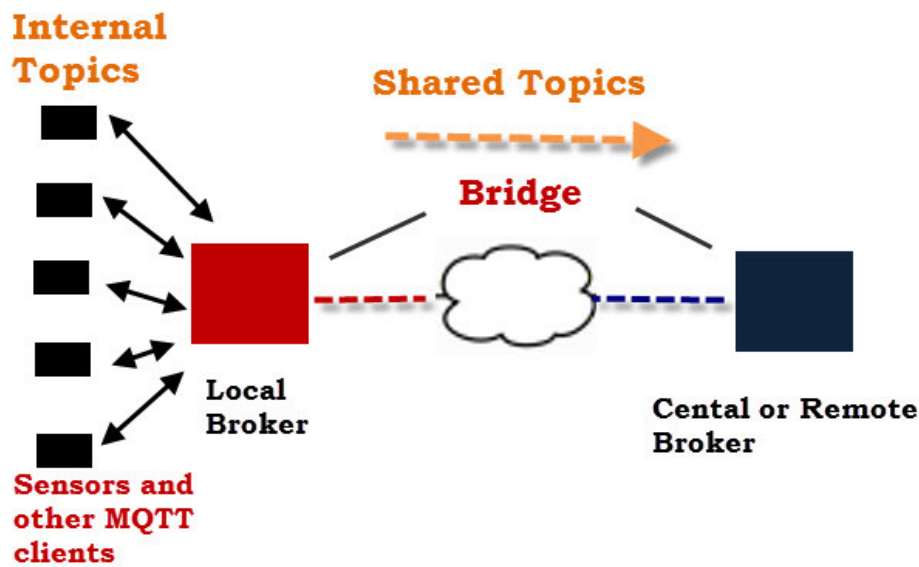
ทดสอบ Websocket

- แก้ website ที่ไฟล์ control.html ให้ระบุ IP เป็น IP ของ Raspberry Pi
- เขียนโปรแกรมบน Raspi ให้ Subscribe ค่าที่รับมา เพื่อมาปิด-เปิด LED และหรีไฟ LED ได้

Bridge Broker



Bridge Broker



MQTT Bridge Illustration

Configuring Bridges

- connection *name*
This variable marks the start of a new bridge connection. It is also used to give the bridge a name which is used as the client id on the remote broker.
- address *address[:port] [address[:port]]*, addresses *address[:port] [address[:port]]*
 - Specify the address and optionally the port of **the bridge to connect to**. This must be given for each bridge connection. If the port is not specified, the default of 1883 is used. Multiple host addresses can be specified on the address config.

Configuring Bridges

- topic *pattern* [[[out | in | both] qos-level] local-prefix remote-prefix]
 - Define a topic pattern to be shared between the two brokers. Any topics matching the pattern (which may include wildcards) are shared. The second parameter defines the direction that the messages will be shared in, so it is possible to import messages from a remote broker using *in*, export messages to a remote broker using *out* or share messages in both directions. If this parameter is not defined, the default of **out** is used. The QoS level defines the publish/subscribe QoS level used for this topic and defaults to **0**.
 - The *local-prefix* and *remote-prefix* options allow topics to be remapped when publishing to and receiving from remote brokers. This allows a topic tree from the local broker to be inserted into the topic tree of the remote broker at an appropriate place.
 - For incoming topics, the bridge will prepend the pattern with the remote prefix and subscribe to the resulting topic on the remote broker. When a matching incoming message is received, the remote prefix will be removed from the topic and then the local prefix added.
 - For outgoing topics, the bridge will prepend the pattern with the local prefix and subscribe to the resulting topic on the local broker. When an outgoing message is processed, the local prefix will be removed from the topic then the remote prefix added.

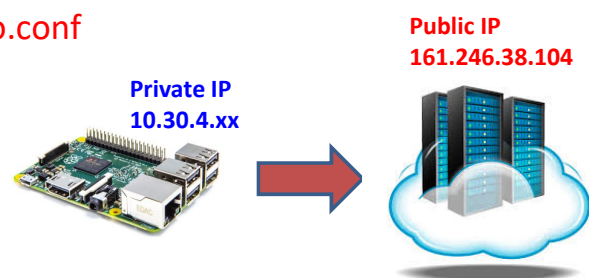
การทดลองที่ 4 Bridge Broker Configuration

เป็นการทดลองเชื่อม Rasp pi เข้ากับ Broker ของคณะ แบบ Single Direction

- เปิดไฟล์ `mosquitto.conf` โดยใช้คำสั่ง

\$ sudo nano /etc/mosquitto/mosquitto.conf

- แก้ไขไฟล์ โดยเพิ่มข้อความดังนี้
`connection bridge_test`
`address 161.246.38.104:1883`
`topic local`



- ทำการ Restart โดยใช้คำสั่ง

\$ sudo /etc/init.d/mosquitto stop

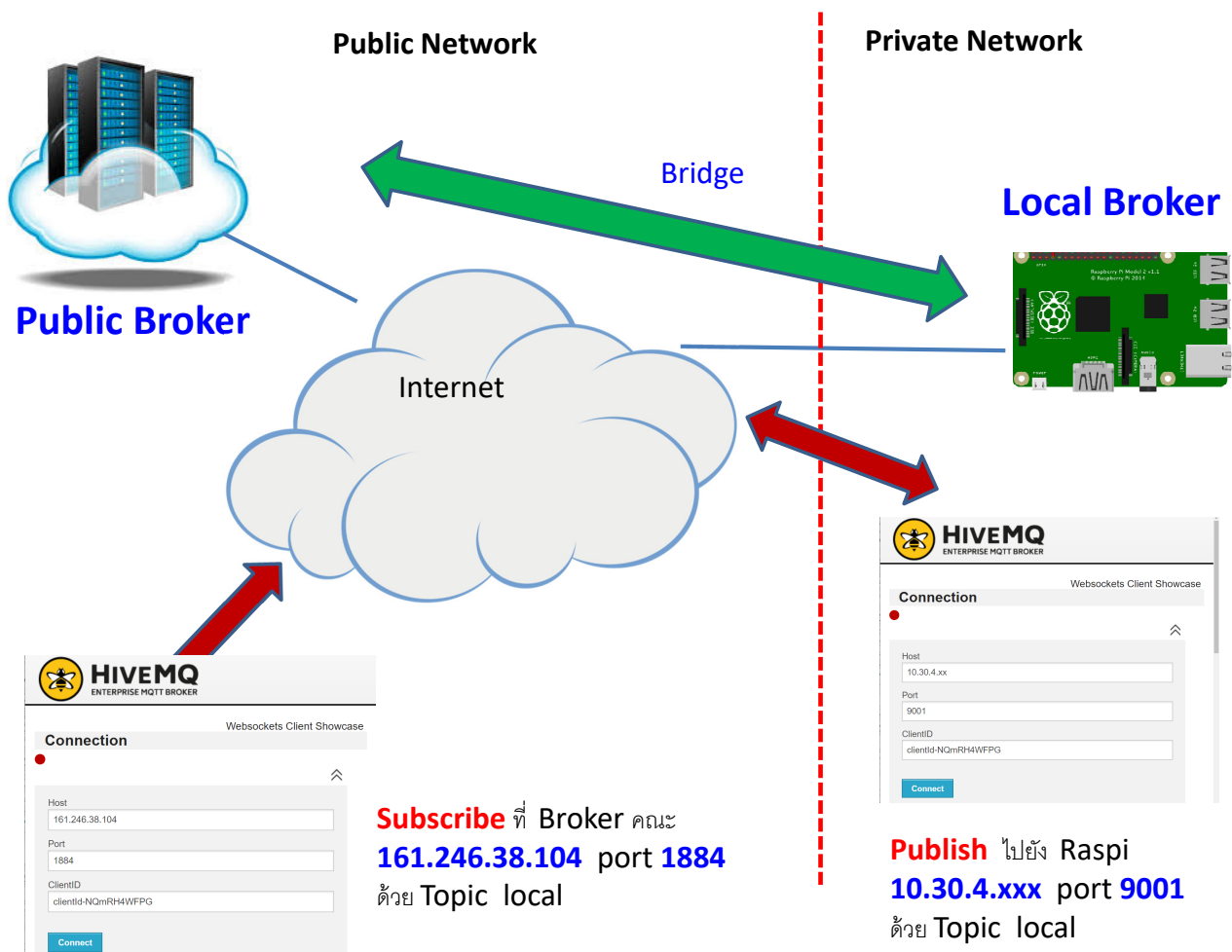
\$ sudo /etc/init.d/mosquitto start

- ทดสอบโดยการ

- Subscribe ที่ Broker คณะ ด้วย Topic local
- Publish ไปที่ Raspberry Pi Broker ด้วย Topic local

Note:

- ชื่อ Connection ต้องไม่ซ้ำกัน
- Address คือ address ของปลายทางที่เราไปเชื่อมต่อ
- export messages to a remote broker using *out* the default of **out**



การทดลองที่ 5 Bridge Broker Configuration

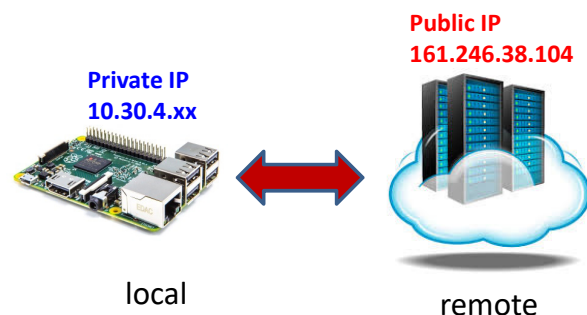
เป็นการทดลองเชื่อม **Rasp pi** เข้ากับ **Broker** ของคณะ แบบ **Dual Direction**

- เปิดไฟล์ **mosquitto.conf** โดยใช้คำสั่ง
\$ **sudo nano /etc/mosquitto/mosquitto.conf**

แก้ไขไฟล์ดังนี้

```
connection bridge_test
address 161.246.38.104:1883
topic # out 0
topic # in 0
```

- ทำการ **Restart** โดยใช้คำสั่ง
\$ **sudo /etc/init.d/mosquitto stop**
\$ **sudo /etc/init.d/mosquitto start**



map an entire topic tree

- ทดสอบโดยการ
 - Subscribe ที่ Broker คณะ ด้วย Topic อะไรก็ได้
 - Publish ที่ Raspberry Pi ด้วย Topic อะไรก็ได้ที่ตรงกัน
 - Subscribe ที่ Raspberry Pi ด้วย Topic อะไรก็ได้
 - Publish ที่ Broker คณะ ด้วย Topic อะไรก็ได้ที่ตรงกัน

Note:

-ชื่อ Connection ต้องไม่ซ้ำกัน