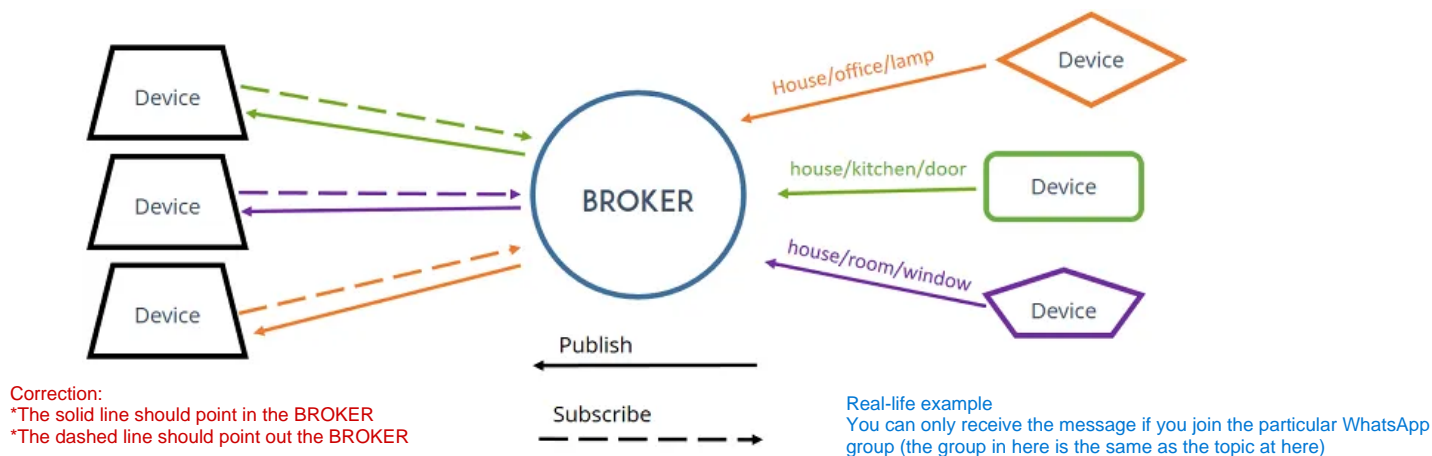


Practical 7

PART 1: Establish MQTT Broker and Client (Subscriber and Publisher)



MQTT = Message Query Telemetry Transport

Step1 is applicable if we use Raspberry Pi

Step 1: We use Paho MQTT, that is a simple library to implement message communication using MQTT protocol, pre-installed in our Raspberry Pi module

* If you are using your own Raspberry Pi, do install mosquitto and MQTT by typing:

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

```
sudo pip3 install paho-mqtt // This is the only command needed in order to run the Mosquitto MQTT on Windows
```

* or check if you have install mosquitto and MQTT by typing:

```
dpkg -l mosquitto mosquitto-clients
```

* Reference: <http://mosquitto.org/>

Step 2: Start the mosquitto broker automatically by typing this command in the terminal:

```
sudo systemctl enable mosquitto.service
```

- Check your own IP address by typing `ifconfig` in terminal
- Test your mqtt services by open two terminals and type the following command:
Terminal A as subscriber:
 - `mosquitto_sub -h [YOUR IF ADDRESS] -t 'test/topic'`
 Terminal B as publisher:
 - `mosquitto_pub -h [YOUR IF ADDRESS] -t 'test/topic' -m 'helloWorld'`

Note: Mosquitto Client is a publisher as well as subscriber

Step 3: In Thonny Python (ID), click “New” to create a new python file and Save As “test09.py”.
Type the following codes for subscriber and publisher:

```
test09.py ✖
1 from time import *
2 from grovepi import *
3 from paho.mqtt import publish
4
5 MQTT_BROKER = [BROKER IP ADDRESS]
6 MQTT_TOPIC = "test"
7
8 publish.single(MQTT_TOPIC, "Any question?", hostname=MQTT_BROKER)
```

```
test10.py ✖
1 #from time import *
2 from grovepi import *
3 from paho.mqtt.client import *
4
5 buzzer = 3
6 pinMode(buzzer, "OUTPUT")
7
8 MQTT_BROKER = [BROKER IP ADDRESS]
9 MQTT_TOPIC = "test"
10
11 def on_connect(client, userdata, flags, rc):
12     print("Connected with result code " + str(rc))
13     client.subscribe(MQTT_TOPIC)
14
15 def on_message(client, userdata, msg):
16     print(msg.topic + " " + str(msg.payload))
17     try:
18         i = int(msg.payload)
19         print(i)
20         if i > 0 and i < 256:
21             analogWrite(buzzer, i)
22     except:
23         analogWrite(buzzer,0)
24
25 client = Client()
26 client.on_connect = on_connect
27 client.on_message = on_message
28 client.connect(MQTT_BROKER, 1883, 60)
29 client.loop_forever()
```

Step 4: Check your raspberry pi IP address by typing this in the terminal:

hostname -I

MQTT by default is installed as a broker in each raspberry pi. Replace Broker IP with your raspberry pi IP.

Step 5: Run both codes to ensure the publish code

Task 1: Modify the code to cross control other sensors / devices with another raspberry pi.

*** We can use MQTTBox (Google Chrome Extension App) to run MQTT Publish and Subscribe action on the spot.**

Additional Tools:

Using NodeRed software to boost your IoT project development

Installation in Linux Distribution (e.g. Debian / Raspbian)

1) Refer to the reference:

<https://nodered.org/docs/getting-started/raspberrypi>

Installing NodeRed in Raspberry Pi with the following command:

```
bash <(curl -sL
https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)
```

```
pi@raspberrypi:~ $ bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)

This script will remove versions of Node.js prior to version 12.x, and Node-RED and
if necessary replace them with Node.js 12.x LTS (erbium) and the latest Node-RED from Npm.

It also moves any Node-RED nodes that are globally installed into your user
~/.node-red/node_modules directory, and adds them to your package.json, so that
you can manage them with the palette manager.

It also tries to run 'npm rebuild' to refresh any extra nodes you have installed
that may have a native binary component. While this normally works ok, you need
to check that it succeeds for your combination of installed nodes.

To do all this it runs commands as root - please satisfy yourself that this will
not damage your Pi, or otherwise compromise your configuration.
If in doubt please backup your SD card first.

Are you really sure you want to do this ? [y/N] ? ☐
```

2) After installed Node-RED, type

Node-red-start

```
Running Node-RED update for user pi at /home/pi on raspbian

This can take 20-30 minutes on the slower Pi versions - please wait.

Stop Node-RED                                ✓
Remove old version of Node-RED                ✓
Remove old version of Node.js                 ✓
Install Node.js LTS                           ✓   Node v12.22.1   Npm 6.14.12
Clean npm cache                               ✓
Install Node-RED core                         ✓   1.3.5
Move global nodes to local                    ✓
Install extra Pi nodes                        -
Npm rebuild existing nodes                    ✓
Add shortcut commands                         ✓
Update systemd script                         ✓

Any errors will be logged to /var/log/nodered-install.log
All done.
You can now start Node-RED with the command node-red-start
or using the icon under Menu / Programming / Node-RED
Then point your browser to localhost:1880 or http://{your_pi_ip-address}:1880

Started Wed 26 May 2021 08:40:58 PM +08 - Finished Wed 26 May 2021 08:47:24 PM +08

pi@raspberrypi:~ $ node-red-start

Start Node-RED
```

```
start node-red

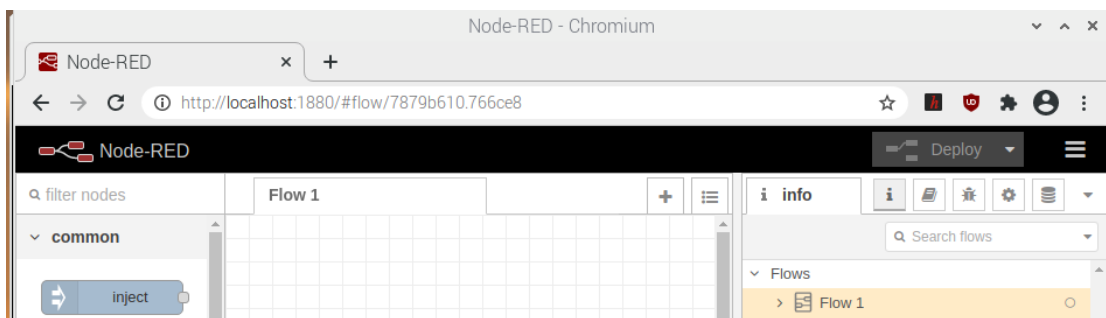
Once Node-RED has started, point a browser at http://192.168.1.108:1880
On Pi Node-RED works better with the Firefox or Chrome browser

Use  node-red-stop           to stop Node-RED
Use  node-red-start          to start Node-RED again
Use  node-red-log             to view the recent log output
Use  sudo systemctl enable nodered.service to autostart Node-RED at every boot
Use  sudo systemctl disable nodered.service to disable autostart on boot

To find more nodes and example flows - go to http://flows.nodered.org

Starting as a systemd service.
26 May 20:55:33 - [info]
Welcome to Node-RED
=====
26 May 20:55:33 - [info] Node-RED version: v1.3.5
26 May 20:55:33 - [info] Node.js version: v12.22.1
26 May 20:55:33 - [info] Linux 4.19.75-v7+ arm LE
26 May 20:55:34 - [info] Loading palette nodes
26 May 20:55:37 - [info] Settings file : /home/pi/.node-red/settings.js
26 May 20:55:37 - [info] Context store : 'default' [module=memory]
26 May 20:55:37 - [info] User directory : /home/pi/.node-red
26 May 20:55:37 - [warn] Projects disabled : editorTheme.projects.enabled=false
26 May 20:55:37 - [info] Flows file : /home/pi/.node-red/flows_raspberrypi.json
```

3) While Node-RED is running, turn on your browser with the localhost:1880, you will see the Node-RED interface.



For Windows

1) Download the node.js from here: <https://nodejs.org/en/>

- Use the recommended version: 14.17.0 LTS

and Run the following command in PowerShell

```
npm install -g --unsafe-perm node-red
```

2) Run the node-red by typing “node-red” in PowerShell.

In the case of unable to load node-red with the following error:

```
+ CategoryInfo          : SecurityError: (:) [], PSSecurityException
+ FullyQualifiedErrorId : UnauthorizedAccess
```

Type the following code and press Y to modify the execution policy, and run node-red again.

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned
```

3) Follow the URL shown in the PowerShell, example:

```
- [info] Server now running at http://127.0.0.1:1880/
```

Go to the page with your default browser to view the node-red interface.

4) Go through the Tutorial Flow 1 and Flow 2 from this Node-RED website:

<https://nodered.org/docs/tutorials/>

5) Add a new Network node (mqtt out), configure the mqtt broker to “broker.hivemq.com”, set the topic to “TARUC/WS/#”, print the output in debug mode.

Discover the feature of Node Red and utilize it for your project development purpose:

<https://developer.ibm.com/components/node-red/blogs/top-5-reasons-to-use-node-red-right-now/>