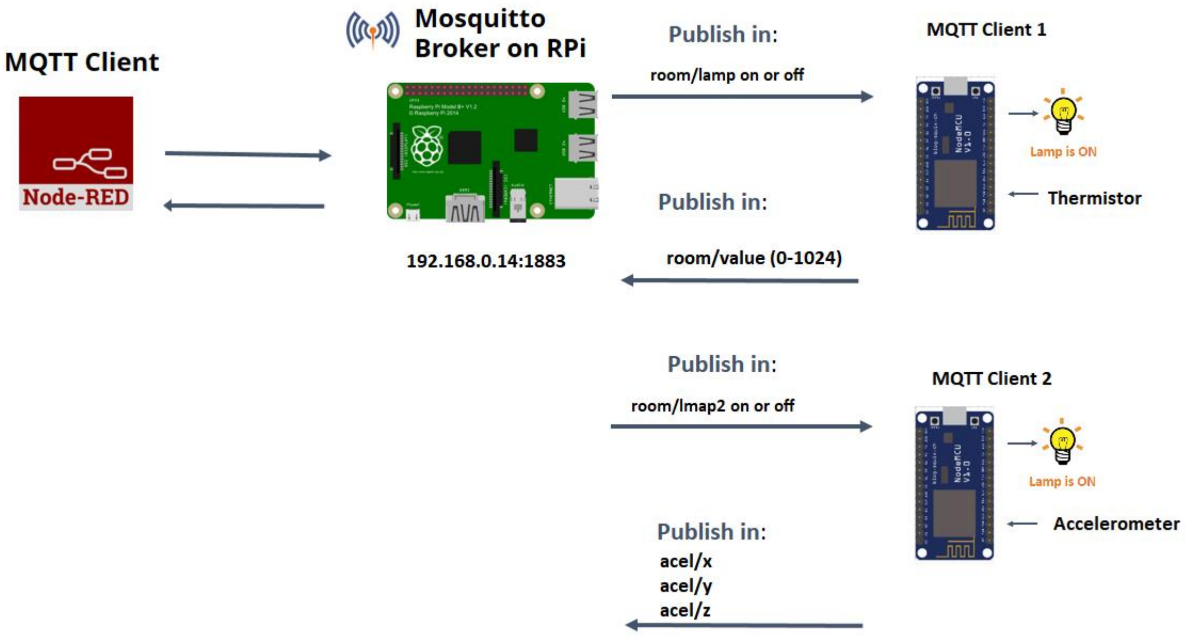
TECH3812
MQTT and Node-Red
Lab #10

Background: As we discussed in class, MQTT is a lightweight protocol designed to transmit data between a Client and Broker (or server). For this lab, the instructor will supply two MQTT Clients (created on ESP8266 microcontroller boards) and a Mosquitto Broker on a Raspberry Pi, in the following configuration:



You will be creating a new Client using Node-RED to subscribe to the broker, display info from the other clients and to also send info to the clients to turn LED's on and off.

Procedure:

Install Node-red as per https://www.brainboxes.com/faq/installing-and-running-node-red-on-a-windows-pc, **skipping** step 4.2 ("node-red-contrib-modbus" install). See step 5 "Running Node-RED" to run the Node-Red program and how to call up the interface within your browser.

Once you are in node-red, you will need to use the following nodes:

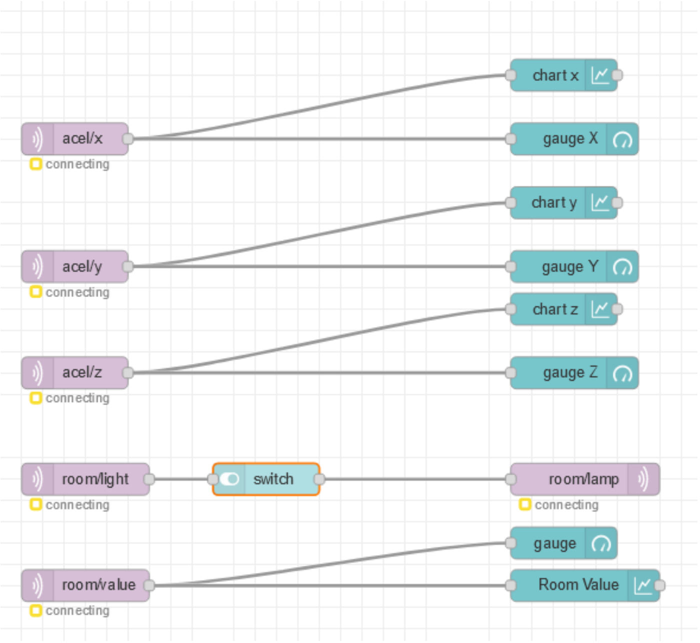
	This will connect to a server (broker) and subscribe you to one or more topics (I suggest using “subscribe to single topic”). You will need to configure the server option (pencil next to pull down) first time. Use the IP address provided in diagram above) on the connection tab (the security and messages tabs can be left as the defaults).
	This will display a point on the user interface as text/number (optional)
	This will display a point on the user interface as a gauge
	This will start a plot of incoming points.
	This allows you to create a slider switch and send info each time the status changes (for this experiment, you will define what is sent as a “String” and assign the values as “on” and “off” (no quotes)
	This node subscribes and sends a message on a topic to the broker

The video: https://www.youtube.com/watch?v=X8ustpkAJ-U will show you the basics of Node-Red and there are a lot of other helpful MQTT / Node-Red tutorials available.

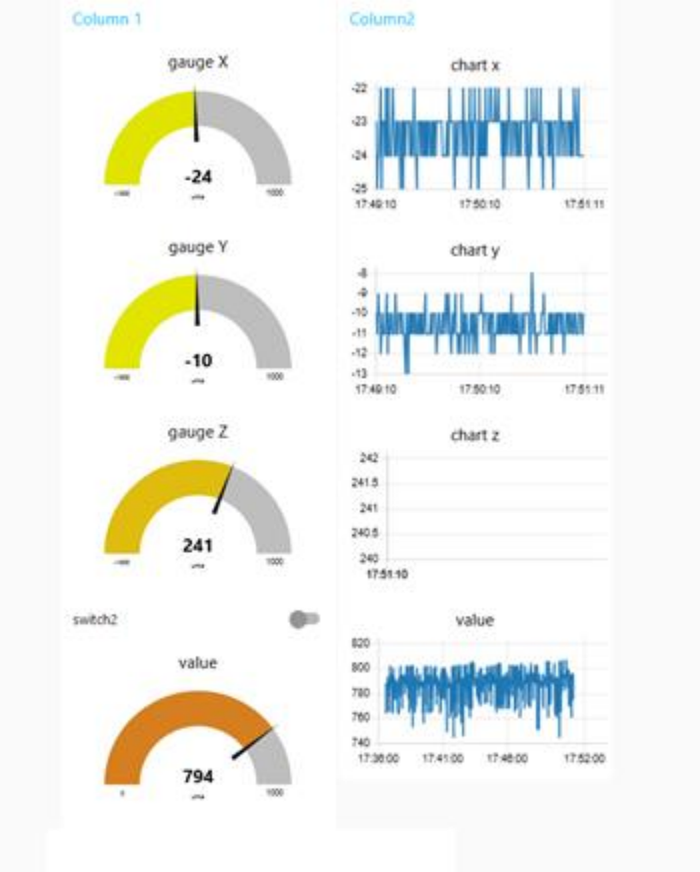
Display each data point (room/value, acel/x, acel/y, acel/z) as a graph, gauge and text.

Create a switch to control the led (room/lamp).

The final flow should look something like:



The final display should look something like the following:



Demo the working node-red flow and dashboard to the instructor for credit.

Export the Flow to a json file and submit the file.

Close Node-red. Open wire-shark and start a capture. Now re-open node-red. Open the Node-Red User Interface (and make sure you are getting data. Now stop the Wire-shark capture. Add a “MQTT” filter and File | Export Specified Packets. The first few packets should include “Connect”, “Connect Ack”, “Subscribe” and “Subscribe Ack” packets. Then you should see a number of “Publish Message” packets.

Submit this file as well.

<u>Note:</u> This lab just scratches to the surface of what node-red can do. Not only can it do MQTT, but it can also do HTTP, UDP and many other protocols. There are many youtube videos and other helpful websites to learn more about Node-Red.