# **MQTT (Message Queuing Telemetry Transport) protocol**

MQTT (Message Queuing Telemetry Transport) is a lightweight, publish-subscribe based messaging protocol designed to provide reliable transmission of data packets of all types for devices. It's a popular mechanism for devices constrained by power and computing. MQTT is commonly used for IoT applications to connect to and communicate with low-power devices such as sensors, actuators, and home appliances, efficiently managing the payload of data transmitted.

MQTT uses TCP.

TCP/IP port 1883 is reserved with IANA for use with MQTT. TCP/IP port 8883 is also registered, for using MQTT over SSL.

Uses a Broker/Server model.

# **MQTT Clients**

Because MQTT clients don't have addresses like email addresses, phone numbers etc. you don't need to assign addresses to clients like you do with most messaging systems.

For MQTTv3.1.1 there is client software available in almost all programming languages and for the main operating systems Linux, Windows, Mac from the <u>Eclipse Paho project.</u>

- Paho Python client.
- Node.js MQTT Client-Starting Guide
- JavaScript Websockets Client
- <u>Client for ESP32</u>

https://test.mosquitto.org/

### **Programming Challenge 3**

- 1) Upload and test the two MQTT demo sender and client programs on two ESP32 boards. These two demo programs use the free **test.mosquitto.org** MQTT server.
- 2) Modify the MQTT demo sender and receiver programs to turn on and off the built-in LED. The sender will send either an "ON" or "OFF" message under the same topic. Upon receiving the message, the receiver will act accordingly. Program any user interface you like for the sender.

Sample Arduino code to turn on and off the led and to flash the led.

#define statusLed 2 void setup() { pinMode(statusLed, OUTPUT); // init the port for output

```
digitalWrite(statusLed, HIGH); // turn on led
digitalWrite(statusLed, LOW); // turn off led
}
void loop() {
    // flash the led every 1000 millisecond
    digitalWrite(statusLed, HIGH); // to turn on
    delay(1000);
    digitalWrite(statusLed, LOW); // to turn off
    delay(1000);
```

Sample Arduino code to read bytes from the Serial console.

```
int incomingByte;
void setup() {
   Serial.begin(115200); // opens serial port, sets data rate to 115200 bps
   Serial.println("Enter something and press Send");
}
void loop() {
   if (Serial.available() > 0) {
      incomingByte = Serial.read(); // read the incoming byte
      Serial.print("Received ");
      Serial.print(incomingByte);
      Serial.print((char)incomingByte);
      Serial.println("]");
   }
}
```

- 3) Instead of using the free **test.mosquitto.org** server, install and configure your own MQTT server on any computer platform you like. See instructions below.
- 4) Do number 2) again but using your MQTT server.
- 5) Instead of running the demo sender program from number 2) on an ESP32, write your own sender/publisher program for your computer. Test it with the receiver program from number 2).

Here's a nice C example for both the sender/publisher and receiver/subscriber that you can use and follow.

https://github.com/LiamBindle/MQTT-C

# Setting up your own MQTT broker

Mosquitto is a free open source MQTT broker that runs on Windows and Linux.

https://mosquitto.org/

### For Linux

Install Mosquitto on Ubuntu

sudo apt-add-repository ppa:mosquitto-dev/mosquitto-ppa

sudo apt-get update

# For Windows

https://cedalo.com/blog/how-to-install-mosquitto-mqtt-broker-onwindows/#How to install\_Mosquitto\_MQTT\_Broker\_on\_Windows

Download from mosquitto.org and run the installation program.

Need to start the service on Windows. Search for services

🤹 Services			_		×
File Action View	Help				
	) 🗟 🛛 📷 🕨 🔲 II 🕨				
🤍 Services (Local)	Services (Local)				
	Mosquitto Broker	Name	Description	Status	Star
	Stop the service <u>Restart</u> the service Description: Eclipse Mosquitto MQTT v5/v3.1.1 broker	<ul> <li>Microsoft Edge Update Serv</li> <li>Microsoft Edge Update Serv</li> <li>Microsoft iSCSI Initiator Ser</li> <li>Microsoft Keyboard Filter</li> <li>Microsoft Office Diagnostic</li> <li>Microsoft Passport</li> <li>Microsoft Passport Container</li> <li>Microsoft Software Shadow</li> <li>Microsoft Storage Spaces S</li> <li>Microsoft Update Health Service</li> <li>Microsoft Update Health Service</li> </ul>	Keeps your Keeps your Manages In Controls ke Run portion Provides pr Manages Io Manages so Host service Provides inf Maintains U	Running	Aut Ma Disi Ma Ma Ma Ma Ma Disi
		Microsoft Windows SMS Ko Mosquitto Broker	Eclipse Mos Service for I	Running	Aut Ma
		Natural Authentication Net.Tcp Port Sharing Service	Signal aggr Provides abi		Ma Disi
	Extended Standard				

Need to punch hole through Windows' Defender Firewall for port 1883



Create a new Inbound Rule to allow incoming traffic through port 1883

🔗 Windows Defender Firewall wit	th Advanced Security					– 🗆 X						
File Action View Help												
Indows Defender Firewall with	Inbound Rules				Actions							
Connection Security Rules	Name	Group	Profile	Enabled	Action	Inbound Rules						
	🔇 мотт		All	Yes	Allow	📰 New Rule						
> Monitoring	🔮 Arduino IDE		Public	Yes	Allow	Filter by Profile						
,	🔇 Arduino IDE		Public	Yes	Allow							
	🛿 arduino_create_agent.exe		Public	Yes	Allow	Y Filter by State						
	🧭 arduino_create_agent.exe		Public	Yes	Allow	Filter by Group						
	CLion 2023.3.2		Public	Public Yes Allow	Allow	View 🕨						
	CLion 2023.3.2		Public	Yes	Allow Allow	a Refresh						
	🔮 computer1		Public	Yes		Export List						
	Computer1		Public	Yes	Allow	Export List						
	Computer I		Public	Yes	Allow	Help						
	Computer 1		Public	Yes	Allow	MQTT						
	Computer3		Public	Ver	Allow	Disable Rule						
	Google Chrome		Public	Ves	Allow	K c :						
	Google Chrome		Public	Yes	Allow	Cut						
	Ø mdns-discovery.exe		Public	Yes	Allow	Сору						
	Ø mdns-discovery.exe		Public	Yes	Allow	🔀 Delete						
	MSMPI-LaunchSvc		All	Yes	Allow	Properties						
	MSMPI-MPIEXEC		All	Yes	Allow							
	SMPI-SMPD		All	Yes	Allow	l incip						
	🔮 OpenJDK Platform binary		Public	Yes	Allow							
	* D/ Di-# Lin		n	V	A II							
	11											

Edit the mosquito.conf file in

C:/Program Files/mosquito/mosquitto.conf

And add the two uncommented lines shown in red below

# affected:
#
# acl\_file
allow\_anonymous true
listener 1883 0.0.0.0
# allow\_zero\_length\_clientid
# auto\_id\_prefix
# password\_file
# plugin

Stop the Mosquitto service and restart it again. The publisher and subscriber should work now. Make sure you're connecting to the correct Mosquitto Broker IP address.