**Bài 2. Trao đổi dữ liệu sử dụng giao thức MQTT**

**Yêu cầu.** Viết chương trình (bằng ngôn ngữ tùy ý: C#, Java, python) thực hiện

* Gửi (publish) dữ liệu lên một MQTT broker. Ví dụ dùng broker: **broker.hivemq.com:1883**
* Đóng gói dữ liệu bằng JSON. Ví dụ:

{

"devID": mqttbox01,

"packetno":126,

"temperature":30,

"humidity":60

}

* Nhận (subscribe) dữ liệu từ broker.
* Parse dữ liệu nhận được và hiển thị các trường dữ liệu tương ứng.
* Sử dụng công cụ MQTTBox để kiểm tra kết quả chạy chương trình.

**Yêu cầu nộp bài:**

Viết báo cáo mô tả ngắn gọn gồm: code chính của chương trình, ***chụp ảnh màn hình thực hiện chương trình và kết quả ở các yêu cầu***.

Submit: file báo cáo, file code

Sample code: (Java)

**package** mqttDemo;

**import** org.eclipse.paho.client.mqttv3.IMqttDeliveryToken;

**import** org.eclipse.paho.client.mqttv3.MqttCallback;

**import** org.eclipse.paho.client.mqttv3.MqttClient;

**import** org.eclipse.paho.client.mqttv3.MqttConnectOptions;

**import** org.eclipse.paho.client.mqttv3.MqttException;

**import** org.eclipse.paho.client.mqttv3.MqttMessage;

**import** org.eclipse.paho.client.mqttv3.persist.MemoryPersistence;

**public** **class** App {

**public** **static** **void** main( String[] args )

{

System.***out***.println( "MQTT Demo!" );

**new** App().doDemo();

}

**public** **void** doDemo() {

String subTopic = "/iot/in";

String pubTopic = "/iot/out";

String content = "\"{'deviceId':'esp123', 'temperature':25,'humidity':77}\"";

**int** qos = 1;

String broker1 = "tcp://broker.hivemq.com:1883";

String broker2 = "tcp://broker.mqtt-dashboard.com:1883";

String clientId = "ID\_OF\_CLIENT";

MemoryPersistence persistence = **new** MemoryPersistence();

**try** {

MqttClient client = **new** MqttClient(broker1, clientId, persistence);

// MQTT connection option

MqttConnectOptions connOpts = **new** MqttConnectOptions();

// retain session

connOpts.setCleanSession(**false**);

connOpts.setAutomaticReconnect(**true**);

connOpts.setConnectionTimeout(5000);

// set callback

client.setCallback(**new** OnMessageCallback());

// establish a connection

System.***out***.println("Connecting to broker: " + broker1);

client.connect(connOpts);

System.***out***.println("Connected");

// Subscribe

client.subscribe(subTopic);

// Required parameters for message publishing

MqttMessage message = **new** MqttMessage(content.getBytes());

message.setQos(qos);

System.***out***.println("Publishing message: " + content);

client.publish(pubTopic, message);

System.***out***.println("Message published");

//client.disconnect();

//System.out.println("Disconnected");

//client.close();

//System.exit(0);

}

**catch** (MqttException me) {

me.printStackTrace();

}

}

**public** **class** OnMessageCallback **implements** MqttCallback {

**public** **void** connectionLost(Throwable cause) {

// After the connection is lost, it usually reconnects here

System.***out***.println("disconnect，you can reconnect");

}

**public** **void** messageArrived(String topic, MqttMessage message) **throws** Exception {

// The messages obtained after subscribe will be executed here

System.***out***.println("Received message topic:" + topic);

System.***out***.println("Received message Qos:" + message.getQos());

System.***out***.println("Received message content:" + **new** String(message.getPayload()));

}

**public** **void** deliveryComplete(IMqttDeliveryToken token) {

System.***out***.println("deliveryComplete---------" + token.isComplete());

}

}

}

Thư viện paho mqtt client (file .jar)  
<https://download.eclipse.org/justj/?file=paho/releases/1.2.5/Java/plugins>

Thêm file thư viện jar vào project eclipse:

<https://www.wikihow.vn/Th%C3%AAm-JAR-v%C3%A0o-%C4%91%C6%B0%E1%BB%9Dng-d%E1%BA%ABn-x%C3%A2y-d%E1%BB%B1ng-d%E1%BB%B1-%C3%A1n-trong-Eclipse-(Java)>