浙大城市学院实验报告

课程名称　物联网技术与应用　实验项目 实验九 MQTT 客户端设计

专业班级 _____ 学号 _____ 姓名 _____

指导老师（签名 ）　蔡建平　日期 _____　实验成绩 _____

一、实验目的：

学习 MQTT 的 Java 资源，掌握 MQTT 客户端设计，连接并完成数据库读写操作。

二、实验内容：

1. 读取订阅的 topic 消息，显示在控制台;
2. 读取订阅的 topic 消息，将 topic、QoS 和 payload 字段，时间戳分别写入数据表中;
3. 订阅并解析来自 WebSocket 客户端的 JSON 格式的消息，并将内容写入数据表中;

三、实验步骤：

1. 读取订阅的 topic 消息，显示在控制台;

接收消息主题 ： toCJP/1248	将主题改为姓名缩写/学号后 4 位
接收消息 Qos ： 1	
接收消息内容 ： close	
客户端掉线！	
接收消息内容 ： {"msg": "Hello, CJP20201129!" }	在 Payload 载荷部分有姓名缩写和日期信息

客户端代码:

```
package mqtt;

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

public class Client {
    public static final String HOST = "ws://47.100.136.15:8083/mqtt";
```

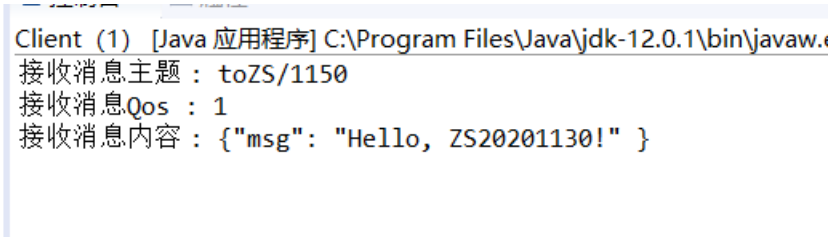
```
public static final String TOPIC = "toZS/1150";
private static final String clientid = "client1150";
private MqttClient client;
private MqttConnectOptions options;
private String userName = "";
private String passWord = "";
public void start() {
    try {

        client = new MqttClient(HOST, clientid, new MemoryPersistence());

        options = new MqttConnectOptions();

        options.setCleanSession(true);

        options.setUserName(userName);

        options.setPassword(passWord.toCharArray());

        options.setConnectionTimeout(10);

        options.setKeepAliveInterval(20);

        client.setCallback(new PushCallback());
        MqttTopic topic = client.getTopic(TOPIC);

        options.setWill(topic, "close".getBytes(), 2, true);

        client.connect(options);

        int[] Qos  = {1};
        String[] topic1 = {TOPIC};
        client.subscribe(topic1, Qos);


    } catch (Exception e) {
        e.printStackTrace();
    }
}
public static void main(String[] args) throws MqttException {
    Client client = new Client();
    client.start();
}
```

```
}
```

回调函数代码:

```
package mqtt;

import org.eclipse.paho.client.mqttv3.IMqttDeliveryToken;
import org.eclipse.paho.client.mqttv3.MqttCallback;
import org.eclipse.paho.client.mqttv3.MqttMessage;



public class PushCallback implements MqttCallback {

    public void connectionLost(Throwable cause) {

        System.out.println("客户端掉线！");
    }

    public void deliveryComplete(IMqttDeliveryToken token) {
        System.out.println("deliveryComplete---------" + token.isComplete());
    }
    public void messageArrived(String topic, MqttMessage message) throws
Exception {
        System.out.println("接收消息主题 : " + topic);
        System.out.println("接收消息Qos : " + message.getQos());
        System.out.println("接收消息内容 : " + new
String(message.getPayload()));
        String str=new String(message.getPayload());

        if(str.equals("close"))
            System.out.println("客户端掉线！");
        else {

            SaveMysql save=new SaveMysql();
            save.savedate(topic,message.getQos(),str);
            //save.savedate2(str);

        }
    }

  }
```

数据接收截图:



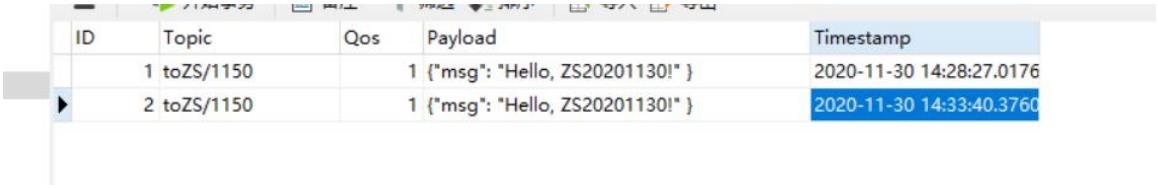
2. 读取订阅的 topic 消息，将 topic、QoS 和 payload 字段，时间戳分别写入数据表中;

写入数据库的程序:

```
public class SaveMysql {
    private static String driver="com.mysql.jdbc.Driver";
    private static String
url="jdbc:mysql://47.100.136.15:3306/mqtt?serverTimezone=UTC";
    private static String user="root";
    private static String password="Sss991126/";
    Connection conn=null;
    Statement stmt=null;
    ResultSet rs=null;
    public void savedate(String topic,int Qos,String payload) {

        java.sql.Timestamp timestamp=new
java.sql.Timestamp(System.currentTimeMillis());
        String sql = "insert into
mqtt(Topic,Qos,payload,Timestamp)"+"values('"+topic+"','"+Qos+"','"+payload
+"','"+timestamp+"');";
        try {
            Class.forName(driver);
            conn=DriverManager.getConnection(url,user,password);
            stmt=conn.createStatement();
            stmt.executeUpdate(sql);
        }
        catch(Exception e) {
            e.printStackTrace();
        }
        finally {
            try {
                if(stmt!=null) stmt.close();
                if(conn!=null) conn.close();
            }
            catch(Exception e) {
```

```
                e.printStackTrace();
            }
        }
    }
```

截图:



ID	Topic	Qos	Payload	Timestamp
1	toZS/1150	1	{"msg": "Hello, ZS20201130!" }	2020-11-30 14:28:27.0176
2	toZS/1150	1	{"msg": "Hello, ZS20201130!" }	2020-11-30 14:33:40.3760

3. （选做）订阅并解析来自 WebSocket 客户端的 JSON 格式的消息，并将内容写入数据表中;

代码:

```
package mqtt;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import com.alibaba.fastjson.JSONObject;

public class SaveMysql {
    private static String driver="com.mysql.jdbc.Driver";
    private static String
url="jdbc:mysql://47.100.136.15:3306/IotZS?serverTimezone=UTC";
    private static String user="root";
    private static String password="Sss991126/";
    Connection conn=null;
    Statement stmt=null;
    ResultSet rs=null;


    public void savedate2(String payload) {
        JSONObject equipment = new JSONObject();

        java.sql.Timestamp time= new
java.sql.Timestamp(System.currentTimeMillis());
```

```
        try{

            equipment = JSONObject.parseObject(payload);
        }
        catch (Exception e){
            e.printStackTrace();
            System.out.println("Json格式错误");
            return ;
        }
        String sql = "insert into
IoT_test(Sid,Sname,Ename,Humidity,Temperature,Timesite,Username)"+"values('
"+equipment.getString("sid")+"','"

+equipment.getString("sname")+"','"+equipment.getString("Ename")+"','"+equi
pment.getDouble("humidity")+"','"+equipment.getDouble("temperature")
                +"','"+time+"','"+equipment.getString("clientid")+"');";
        try {
            Class.forName(driver);
            conn=DriverManager.getConnection(url,user,password);
            stmt=conn.createStatement();
            stmt.executeUpdate(sql);
        }
        catch(Exception e) {
            e.printStackTrace();
        }
        finally {
            try {
                if(stmt!=null) stmt.close();
                if(conn!=null) conn.close();
            }
            catch(Exception e) {
                e.printStackTrace();
            }
        }

    }

}
```

截图:

id	Sid	Sname	Ename	Humidity	Temperature	Timesite	Username
27	801150	zhangshuai	null	41.00	26.00	2020-11-30 15:	client 3180115(
28	801150	zhangshuai	null	40.00	26.00	2020-11-30 15:	client 3180115(
29	801150	zhangshuai	null	40.00	26.00	2020-11-30 15:	client 3180115(
30	801150	zhangshuai	null	39.00	26.00	2020-11-30 15:	client 3180115(
31	801150	zhangshuai	null	39.00	26.00	2020-11-30 15:	client 3180115(
32	801150	zhangshuai	null	39.00	26.00	2020-11-30 15:	client 3180115(
33	801150	zhangshuai	null	39.00	26.00	2020-11-30 15:	client 3180115(
34	801150	zhangshuai	null	39.00	26.00	2020-11-30 15:	client 3180115(

4. （选做）采集 ESP8266 的温度、湿度、时间等数据，构建 JSON 格式的消息，并写入到 MySQL 数据库中。

参考字段:

序号	学号	姓名	终端名称	湿度	湿度	时间	用户名

代码:

```
#include <ESP8266WiFi.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"
#define WLAN_SSID       "Mi 10000 Ultra"
#define WLAN_PASS       "88888888"
#define AIO_SERVER      "47.100.136.15"
#define AIO_SERVERPORT  1883
#define AIO_USERNAME    ""
#define AIO_KEY         ""
#include"SSD1306Wire.h"
SSD1306Wire display(0x3c,2,14);
#include<dht11.h>
#include<ArduinoJson.h>



dht11 DHT11;
// Create an ESP8266 WiFiClient class to connect to the MQTT server.
```

```
WiFiClient client;
// or... use WiFiClientSecure for SSL
//WiFiClientSecure client;

StaticJsonDocument<200> doc;
// Setup the MQTT client class by passing in the WiFi client and MQTT server and login details.
Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT, AIO_USERNAME, AIO_KEY);
// Setup a feed called 'photocell' for publishing.
// Notice MQTT paths for AIO follow the form: <username>/feeds/<feedname>
Adafruit_MQTT_Publish photocell = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "toZS/1150");

// Setup a feed called 'onoff' for subscribing to changes.
Adafruit_MQTT_Subscribe onoffbutton = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "toZS/1150");

// Bug workaround for Arduino 1.6.6, it seems to need a function declaration
// for some reason (only affects ESP8266, likely an arduino-builder bug).
void MQTT_connect();

void setup() {

Serial.begin(115200);
delay(10);
  display.init();

Serial.println(F("Adafruit MQTT demo"));
```

```
// Connect to WiFi access point.
Serial.println();
Serial.println();
Serial.print("Connecting to ");
Serial.println(WLAN_SSID);

WiFi.begin(WLAN_SSID, WLAN_PASS);
while (WiFi.status() != WL_CONNECTED) { delay(500);
Serial.print(".");
}
Serial.println();

Serial.println("WiFi connected");
Serial.println("IP address: "); Serial.println(WiFi.localIP());

// Setup MQTT subscription for onoff feed.
mqtt.subscribe(&onoffbutton);
}
uint32_t x=0; void loop() {
      int chk = DHT11.read(5);
// Ensure the connection to the MQTT server is alive (this will make the first
// connection and automatically reconnect when disconnected). See the
MQTT_connect
// function definition further below.
MQTT_connect();

// this is our 'wait for incoming subscription packets' busy subloop
// try to spend your time here

Adafruit_MQTT_Subscribe *subscription;
```

```
  while ((subscription = mqtt.readSubscription(5000))) { if (subscription == &onoffbutton)
{
    Serial.print(F("Got: "));
    Serial.println((char *)onoffbutton.lastread);

    display.flipScreenVertically();
    display.clear();
    display.drawString(0,10,"topic: toZS/1150");
    display.drawString(0,20,(char *)onoffbutton.lastread);
   // display.display();
    delay(2000);
  }
  }
  // Now we can publish stuff!
  Serial.print(F("\nSending photocell val ")); Serial.print(x);
  Serial.print("...");

  String output;

  doc["sid"] = "31801150";
  doc["sname"] = "zhangshuai";
  doc["ename"] = "test1";
  doc["clientid"] = "client 31801150";
  doc["humidity"] = (float)DHT11.humidity;
  doc["temperature"] = (float)DHT11.temperature;
  serializeJson(doc ,output);



  if (! photocell.publish(output.c_str())) { Serial.println(F("Failed"));
```

```
} else {
Serial.println(F("OK!"));

   display.flipScreenVertically();
   display.clear();
   display.drawString(0,10,"topic: iot/1");
   display.drawString(0,20,String(x));
//    display.display();
   delay(2000);
}

// ping the server to keep the mqtt connection alive
// NOT required if you are publishing once every KEEPALIVE seconds
/*
if(! mqtt.ping()) { mqtt.disconnect();
}
*/
}

// Function to connect and reconnect as necessary to the MQTT server.
// Should be called in the loop function and it will take care if connecting.
void MQTT_connect() {
int8_t ret;

// Stop if already connected.
if (mqtt.connected()) {
return;
}
Serial.print("Connecting to MQTT... "); uint8_t retries = 3;
while ((ret = mqtt.connect()) != 0) { // connect will return 0 for connected
```

```
    Serial.println(mqtt.connectErrorString(ret));
Serial.println("Retrying MQTT connection in 5 seconds..."); mqtt.disconnect();
delay(5000);    // wait 5 seconds retries--;
if (retries == 0) {
// basically die and wait for WDT to reset me
while (1);
}
}
Serial.println("MQTT Connected!");
}
```

截图:

id	Sid	Sname	Ename	Humidity	Temperature	Timesite	Username
27	801150	zhangshuai	null	41.00	26.00	2020-11-30 15:	client 3180115(
28	801150	zhangshuai	null	40.00	26.00	2020-11-30 15:	client 3180115(
29	801150	zhangshuai	null	40.00	26.00	2020-11-30 15:	client 3180115(
30	801150	zhangshuai	null	39.00	26.00	2020-11-30 15:	client 3180115(
31	801150	zhangshuai	null	39.00	26.00	2020-11-30 15:	client 3180115(
32	801150	zhangshuai	null	39.00	26.00	2020-11-30 15:	client 3180115(
33	801150	zhangshuai	null	39.00	26.00	2020-11-30 15:	client 3180115(