

廈門大學



软件学院

物联网技术导论实验一

班 级 _____ 软工三班 _____
学 院 _____ 信息学院 _____
专 业 _____ 软件工程 _____
年 级 _____ 2021 级 _____
学 号 _____ 32420212202930 _____
姓 名 _____ 陈澄 _____

1 实验背景

随着汽车技术的不断发展，车辆的智能化程度越来越高。其中，温度传感器是一种广泛应用于汽车系统中的传感器之一，它能够实时监测车辆周围的环境温度，并将数据传输至服务器或其他控制单元，以供进一步分析和应用。

本实验旨在利用已部署在车辆中的温度传感器，以及与之连接的 MQTT 服务器，实现对车辆外观颜色的动态调节。具体而言，温度传感器每隔1秒将当前环境温度系数发送至 MQTT 服务器，请同学们实现对 MQTT 服务器的温度系数进行订阅并以此改变车辆的外观颜色，从而使车辆外观与环境温度保持一定的关联性。

2 实验内容

1.熟悉 ThingJS (<https://www.thingsjs.com/guide/>) 平台的开发模式，注册体验账户，或者寻找相似的 Web3d 开源组件用于可视化

2.针对 MQTT 协议进行学习并完成下述实验与实验报告

3.在软件仿真平台上实现（比如 ThingJS->低代码开发->数据->数据对接 MQTT）

a)温度系数的订阅

b)根据温度系数来改变车辆的颜色

i.0~10° C 红色

ii.10~20° C 绿色

iii.20~30° C 蓝色

3 实验步骤

1.完善 MQTT_subscriber 类

首先，完善 MQTT_subscriber 类中的 initConnection 方法，以便于成功建立连接后订阅特定的主题，并处理接收到的消息。

```
// 初始化连接
initConnection() {
  var _this = this;
  this.mqttClient = mqtt.connect(this.socketUrl, this.options);
  this.mqttClient.on('connect', function () {
    console.log("连接成功");
    // 订阅主题
    this.mqttClient.subscribe('/mqtt/WebOfThings/test', function (err) {
      if (!err) {
        console.log("订阅成功");
      }
    });
  });

  // 处理接收到的消息
  this.mqttClient.on('message', function (topic, message) {
    // message 是 Buffer
    const data = message.toString();
    console.log(`接收到数据: ${data}`);
    updateState(data); // 更新状态
  });
}
```

2. 完善 disconnection 方法

接下来，我们需要确保在断开连接时，能够正确关闭 MQTT 客户端。

```
// 关闭连接
disconnection() {
  if (this.mqttClient) {
    this.mqttClient.end();
    this.mqttClient = null;
    console.log('MQTT连接已关闭!');
  }
}
```

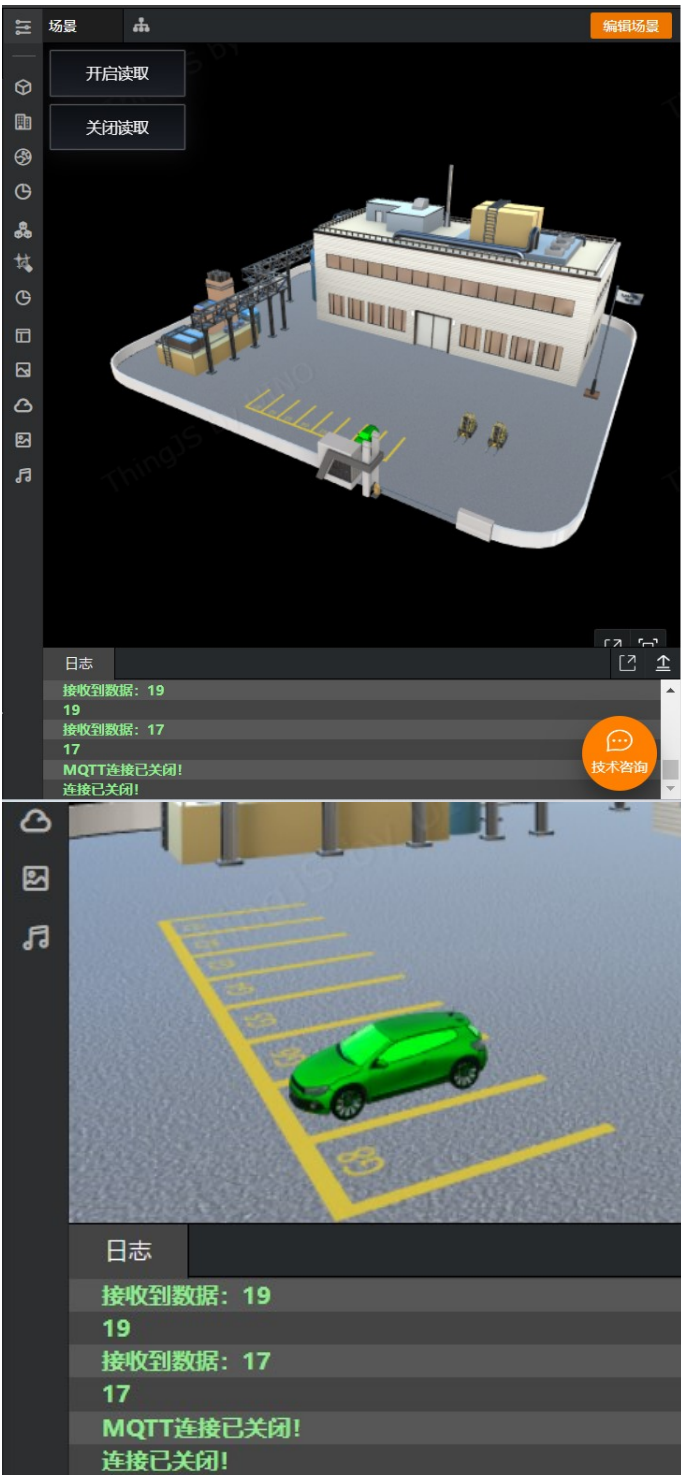
3. 修改 changeColor 方法

在 changeColor 方法中，实现车辆颜色随着温度变化

```
// 根据不同的温度系数对车辆的颜色进行控制
function changeColor(obj) {
  var temper = obj.getAttribute("monitorData/温度");
  var value = parseInt(temper);
  console.log(value);

  if (value >= 0 && value <= 10) {
    // 当温度在0~10℃时，变红
    obj.style.color = 'rgb(255,0,0)';
  } else if (value > 10 && value <= 20) {
    // 当温度在10~20℃时，变绿
    obj.style.color = 'rgb(0,255,0)';
  } else if (value > 20 && value <= 30) {
    // 当温度在20~30℃时，变蓝
    obj.style.color = 'rgb(0,0,255)';
  } else {
    // 其他情况恢复默认颜色
    obj.style.color = null;
  }
}
```

4.效果演示



4 我的体会

通过这次实验我了解到：

MQTT（Message Queuing Telemetry Transport）是一种轻量级的消息传输协议，专为小型设备和低带宽、不可靠或高延迟的网络环境设计。由于其设计简单、开销小、资源需求低等特点，MQTT 广泛应用于物联网领域，包括家居自动化、工业监控、医疗设备等场景。由于其轻量级和高效的特性，MQTT 被广泛应用于需要远程监控和控制的场景，如智能家居、工业自动化、车辆追踪、环境监测等。在这些场景中，设备可能位于网络条件较差的环境下，或者需要在电池供电的情况下尽可能减少能耗，MQTT 提供了一种有效的解决方案。

同时，我对 MQTT 的工作模式——发布/订阅模式也有了更深的理解，该模式这意味着客户端可以订阅一个或多个主题，并接收发布到这些主题的消息。同样，客户端也可以发布消息到一个特定的主题，该消息会被分发给订阅了该主题的所有客户端。这种模式解耦了消息的生产者和消费者。