# MQTT 能流图数据解析与前端处理笔记

## 项目背景

当前系统基于 Vue2 构建，通过订阅设备 MQTT 数据（22位整型数组）实时渲染能流图页面，展示设备运行状态，包括 PV、电池、电网、负载等功率方向与大小。

MQTT 数据来源为设备通讯棒直传，解析需严格对齐后端 Java Hm6Page3 模块中的字段顺序、倍率处理、符号位判断及优化逻辑。

## 原始数据格式

数据类型：MQTT 整数数组（22项）

典型样例：[0,1,2,3,...]

映射字段示例：

电池功率 → 第 6 位（index 5）

电网功率 → 第 9 位（index 8）

## 数据解析规则

### 1. 字节解析

每个字段为 2 bytes，即一个 short 类型，需使用 twoBytesSigned(data[index]) 或 twoBytesUnsigned() 进行符号扩展。

### 2. 倍率系数（CoefficientTool）

后端处理倍数为 0.01，单位 KW，前端需同步：

value = twoBytesSigned(data[index]) \* 0.01;

### 3. 字段方向含义（功率）

batteryPower：

正值：充电

负值：放电

gridPower：

正值：向电网买电

负值：向电网卖电

loadPower 始终为正（设备当前总功耗）

### 4. 功率计算与展示

统一单位：kW

展示逻辑需考虑三种优化：

电池功率方向优化

电网功率买卖方向优化

负载计算修正（load = pv + battery + grid）

## 前端兼容性处理（Vue2）

由于 Vue2 环境使用较旧 Babel/webpack，需避免使用如下语法：

可选链 ?.

空值合并 ??

示例替代写法：

const pvObj = sources.find(s => s.key === "pv");

pv = pvObj ? pvObj.value : 0;

## 与后端逻辑的对齐

|  |  |  |
| --- | --- | --- |
| **字段/逻辑点** | **是否已实现** | **说明** |
| PV、电池、电网、负载功率解析 | ✅ | 使用与后端一致的两字节 + 倍率处理 |
| 字段方向逻辑 | ✅ | battery 正负表示充/放，grid 正负表示买/卖电 |
| 展示单位对齐 | ✅ | 均为 kW，\*0.01 系数 |
| 电池 SOC 解析 | ✅ | index 2，两字节 signed，单位 % |
| 优化逻辑对齐（部分） | ⚠️ | 需补充尖峰点平滑、突变抛帧等 |

## 待扩展与不可实现项

|  |  |  |
| --- | --- | --- |
| 功能 | 当前状态 | 说明 |
| SOC 报警判断 | ❌ 伪造为 false | 后续可通过告警字段/位处理 |
| 尖峰点平滑 | ❌ 未实现 | 后端有波动平滑处理，前端暂无依据 |
| 抛帧修正 | ❌ 无法判断 | MQTT 未提供序号/时间戳，前端无法判断是否丢包 |
| 启停突变逻辑 | ❌ 未实现 | 依赖状态边缘识别或跳变趋势，逻辑复杂 |
| 剩余充电上限 SOC | ❌ 未实现 | 暂无字段判断是否禁止充电 |

## 技术能力体现（可用于面试）

精通 MQTT 数据结构解析，能独立对齐设备端与后端协议字段

熟悉二进制处理与有符号/无符号位转换

理解电力系统功率正负方向定义，能根据设备运行逻辑判断数据合理性

具备前端旧框架环境兼容处理能力（Vue2/Babel 限制下的语法改写）

能协同后端梳理数据展示与真实运行状态间的映射

能识别系统优化点，并具备设计前端展示逻辑的能力