# PLC模拟指令系统设计与实现

# PLC 指令模拟下发系统详细设计方案

## 一、设计目标

基于上传的两个文件（原片仓储.xlsx、上片机 - v003.xlsx）中不同模块的字段差异，设计一个灵活的 PLC 指令模拟下发系统，用于测试 MES 程序。该系统要能够根据不同模块的需求，准确下发 PLC 指令，并实现任务流转的自动化，同时尽可能提高代码的复用性和可维护性。

## 二、系统架构

采用分层架构设计，分为前端展示层、后端业务逻辑层和数据存储层，以及与外部的 MES 系统和 PLC 设备的交互接口。

### 2.1 前端展示层

提供友好的用户界面，让用户能够选择不同的模块（如原片仓储、上片等），配置该模块对应的 PLC 指令参数（如字段值、定时任务时间等），并实时监控任务的执行状态和结果。

### 2.2 后端业务逻辑层

负责处理前端传来的请求，根据不同模块的需求生成相应的 PLC 指令，并与 PLC 设备进行通信下发指令。同时，监控 MES 系统的响应，按照任务流转逻辑进行后续操作。

### 2.3 数据存储层

存储不同模块的配置信息（包括字段定义、定时任务设置等）、任务执行历史记录等数据，以便后续查询和分析。

### 2.4 外部交互接口

与 MES 系统进行交互，获取 MES 对 PLC 指令的响应；与 PLC 设备进行通信，实现指令的下发和状态的读取。

## 三、模块分析

### 3.1 上片模块

根据上片机 - v003.xlsx 文件，该模块在 PLC 下发指令时需要考虑完成数量和破损数量等字段。这意味着在生成 PLC 指令时，要将这些字段的值准确设置并下发。

### 3.2 原片仓储模块

原片仓储.xlsx 文件对应的模块，其所需字段与上片模块不同，在设计 PLC 指令下发逻辑时，要根据该模块的特点进行参数配置和指令生成。

## 四、核心功能设计

### 4.1 模块配置管理

* **模块定义**：在系统中定义不同的模块，如“上片模块”和“原片仓储模块”，每个模块有唯一的标识和对应的配置信息。

* **字段配置**：针对每个模块，配置其所需的 PLC 字段。可以通过文件解析（如解析上传的 Excel 文件）自动获取字段信息，也支持手动添加和修改。对于上片模块，添加完成数量和破损数量等字段；对于原片仓储模块，添加该模块特有的字段。

* **定时任务配置**：允许用户为每个模块配置定时任务参数，如任务执行的时间间隔、延迟时间等。可以设置固定时间执行，也可以设置周期性执行。

### 4.2 PLC 指令生成与下发

* **指令生成**：根据模块配置信息和用户设置的参数，生成相应的 PLC 指令。对于上片模块，将完成数量和破损数量等字段的值按照 PLC 协议要求进行编码，组成完整的指令。对于原片仓储模块，按照其模块特点生成指令。

* **指令下发**：通过与 PLC 设备的通信接口，将生成的指令下发到 PLC 设备。支持多种通信协议，如 Modbus、Profinet 等，根据实际 PLC 设备的协议进行选择和配置。

### 4.3 任务流转监控

* **MES 响应监听**：实时监听 MES 系统对 PLC 指令的响应。当 PLC 下发请求字后，等待 MES 发送相应的反馈（如发送字），并验证反馈的正确性。

* **任务状态更新**：根据 MES 的响应和任务流转逻辑，及时更新任务状态。例如，当 MES 发送发送字后，将任务状态更新为“MES 已接收任务”；当 PLC 收到汇报字后，将任务状态更新为“任务完成”。

* **异常处理**：如果在任务流转过程中出现异常情况（如 MES 未及时响应、PLC 通信故障等），系统要能够捕获并记录异常信息，同时采取相应的处理措施，如重试、告警等。

### 4.4 定时任务管理

不建议为每个模块创建完全独立的定时任务（如 loadGlassPlcTest、GlassStoragePlcTest），因为这样会导致代码重复和维护困难。更好的方法是设计一个通用的定时任务框架，根据模块配置动态加载和执行任务。

* **任务调度器**：采用一个集中的任务调度器（如 Spring 框架中的 @Scheduled 注解或 Quartz 框架），负责管理所有模块的定时任务。

* **任务配置加载**：任务调度器从数据存储层加载每个模块的定时任务配置信息，包括任务执行时间、执行频率等。

* **动态任务执行**：根据加载的配置信息，任务调度器动态创建和执行对应的任务。对于每个任务，调用相应模块的 PLC 指令生成和下发逻辑。例如，当定时任务触发上片模块的任务时，调用上片模块的指令生成方法，将完成数量和破损数量等字段考虑在内，生成并下发指令。

## 五、数据存储设计

### 5.1 模块配置表

| 存储每个模块的基本信息和字段配置。 |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 描述 |
| module\_id | int | 模块唯一标识 |
| module\_name | varchar | 模块名称（如“上片模块”“原片仓储模块”） |
| field\_config | text | 模块字段配置信息（以 JSON 格式存储字段名称、类型、DB 块地址等） |

### 5.2 定时任务配置表

| 存储每个模块的定时任务配置。 |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 描述 |
| task\_id | int | 任务唯一标识 |
| module\_id | int | 关联的模块 ID |
| task\_type | varchar | 任务类型（如“定时执行”“周期执行”） |
| execution\_time | datetime | 任务执行时间（对于定时执行任务） |
| interval\_time | int | 任务执行间隔时间（对于周期执行任务，单位：秒） |

### 5.3 任务执行记录表

| 记录每个任务的执行情况和结果。 |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 描述 |
| record\_id | int | 记录唯一标识 |
| task\_id | int | 关联的任务 ID |
| execution\_time | datetime | 任务实际执行时间 |
| status | varchar | 任务执行状态（如“成功”“失败”“执行中”） |
| result\_info | text | 任务执行结果详细信息（如异常信息、MES 响应数据等） |

## 六、界面设计

### 6.1 模块配置界面

* 提供模块选择下拉框，用户可以选择要配置的模块。

* 展示所选模块的字段列表，可对字段进行编辑、添加和删除操作。

* 提供定时任务配置区域，用户可以设置任务的执行时间和频率等参数。

### 6.2 任务监控界面

* **任务列表展示**：以表格形式呈现所有模块的任务列表，包含任务 ID、所属模块名称、任务状态（如等待执行、执行中、已完成、失败等）、最近执行时间等信息。表格支持排序、筛选功能，方便用户快速定位关注的任务。

* **任务详情查看**：用户点击任务列表中的某一任务，可查看该任务的详细信息，如任务配置参数（包括下发的 PLC 指令具体字段值、定时任务设置等）、执行历史记录（每次执行的时间、状态、结果信息等）。

* **实时状态更新**：通过 WebSocket 或定时轮询的方式，实时获取任务的最新状态，并在界面上动态更新展示。例如，当任务从“执行中”变为“已完成”时，界面立即刷新显示新状态。

* **任务进度可视化**：对于执行时间较长的任务，以进度条的形式直观展示任务的执行进度。比如，若任务有多个子步骤，可根据已完成子步骤的比例更新进度条。

* **异常告警提示**：当任务执行过程中出现异常（如 MES 未响应、PLC 通信故障等），界面以醒目的颜色（如红色）标记该任务，并弹出告警提示框，显示异常的具体信息，方便用户及时处理。

### 6.3 日志查询界面

* **日志列表呈现**：将系统的操作日志、任务执行日志等以列表形式展示，每条日志记录包含日志时间、日志类型（如信息、警告、错误等）、日志内容等信息。

* **日志筛选与搜索**：提供多种筛选条件，如日志时间范围、日志类型、所属模块等，用户可根据需求进行筛选。同时，支持关键词搜索功能，方便快速定位特定的日志记录。

* **日志详情查看**：用户点击某条日志记录，可查看该日志的详细内容，包括完整的异常堆栈信息（如果是错误日志）等。

* **日志导出功能**：允许用户将筛选后的日志记录导出为文件（如 CSV、Excel 等格式），以便后续进一步分析和存档。

‍