

厂区级环境保障系统状态转换 智能化控制设计实现

蔡维利¹, 谢志敏², 梁春芳¹

(1. 中国人民解放军 91515 部队, 海南 三亚 572000;

2. 海军军事海洋环境建设办公室, 北京 100000)

摘要: 针对某厂区环境保障系统状态转换人工操作程序复杂、耗时长、配合难度大, 无法全面掌握设备运行情况, 转换后缺乏快速核验手段和无报表生成等问题; 开发工艺批量处理指令模块, 利用 OPC 技术建立与 WinCC 的通讯, 实现了系统状态一键转换; 利用 Excel 的公式数据处理能力, 对后台数据“变量池”进行梳理, 实现了设备运行和通讯状态的实时感知; 利用 ADO 技术创建了与 Access 数据库的通讯, 通过合理设置定时程序的循环写时间间隔和读时间区间, 结合 VBA 编程和录制宏, 实现了转换结果核验和报表生成功能; 经模拟和实际应用验证, 该程序将系统状态转换时间由 60 分钟以上缩短至 6 分钟以内, 效率提升 90% 以上。

关键词: 设备状态; 一键转换; OPC; WinCC; Excel; Access; VBA

Design and Implementation of Intelligent Control for State Transitions in Factory-level Environmental Support Systems

CAI Weili¹, XIE Zhimin², LIANG Chunfang¹

(1. 91515 Unit, PLA, Sanya 572000, China;

2. Naval Military Marine Environment Construction Office, Beijing 100000, China)

Abstract: Aiming at the problems in an industrial plant's environmental control system, including complex manual operation procedures, time-consuming processes, difficulty in coordination, inability to fully grasp the operational status of the equipment, lack of rapid verification methods, and absence of report generation after conversion, a batch processing command module for the Technology of Product (TOP) is developed. The OLE for Process Control (OPC) technology is used to establish communication with Windows Control Center (WinCC) to achieve the one-click conversion of the system state. By using the Excel's formula calculation capabilities, a "variable pool" of the backend data is organized to realize the real-time perception of equipment operation and communication status. The ActiveX Data Object (ADO) technology is used to communicate with an Access database. By reasonably setting the loop-write time interval and read-time interval of the timer program, and combined with the Visual Basic for Application (VBA) programming and macro recording, it achieves the verification of conversion results and automatic report generation. Simulation and practical application demonstrate that this solution reduces the transition time of the system state from over 60 minutes to less than 6 minutes, improving an efficiency of over 90%.

Keywords: equipment status; one-click conversion; OPC; WinCC; Excel; Access; VBA

0 引言

厂区环境保障系统在平时可能因市政电网电力故障、上级电站设备故障或电站维护等原因导致停电, 进而导致内部各系统及设备停机。自备电站发电或市电恢

复供电后, 需按照工艺要求启动设备, 即将各系统恢复至断电前的运行状态。可能遭受核生化武器袭击, 需快速、准确地实现清洁式通风、滤毒式通风和隔绝式通风之间的三防转换^[1-2]。这些操作涵盖通风空调、给排水和供配电等多个专业, 涉及多设备的批量操作及各系统

收稿日期: 2025-04-16; 修回日期: 2025-05-30。

作者简介: 蔡维利(1989-), 男, 硕士, 工程师。

引用格式: 蔡维利, 谢志敏, 梁春芳. 厂区级环境保障系统状态转换智能化控制设计实现[J]. 计算机测量与控制, 2025, 33(8): 188-196, 248.

的顺序启动和联动控制。例如, 日常停电恢复的顺序为: 关闭风阀, 启动供配电、冷却水、除湿空调机组和风阀等。人工操作需要二三十余人参与、通过电话进行配合操作易出现操作失误、耗时 60 min 以上、环境温湿度超标率达 90% 以上、超标时长达 30 min。人工操作程序复杂、耗时长、配合难度大。实现系统状态的一键转换不仅能缩短系统启动时间, 减轻操作人员负担, 减少人为失误带来的损失还能提高环境保障能力水平。

环境保障控制系统普遍采用集散控制系统 (DCS, distributed control system), 其“集中管理、分散控制”模式虽技术成熟, 但存在系统组态和升级改造困难的固有缺陷。为突破传统架构限制, 可以将一种扁平化、无中心的新型建筑平台技术应用于国防工程智能化领域, 基于群智能建筑图形化编程语言, 开发三防转换的图形化编程^[3]。但采用的图形化编程语言还停留在理论和设计层面, 尚未形成完整的工程应用体系。也可以利用动态数据交换 (DDE, dynamic data exchange) 技术实现 WinCC (Windows Control Center) 与 VB 的交互通讯^[4], 但需注意自 WinCC V7.0 SP2 版本起, 不再支持 Windows DDE 通道, 现有系统升级采用 OPC (OLE for Process Control) 标准通讯协议。

某厂区环境保障系统, 基本实现了各系统设备远程监控和集成, 但仍然存在缺陷: 系统状态转换自动化程度不足, 停电恢复和三防转换仍需人工现场或远程逐一操作, 存在操作流程复杂、人员协同效率低等问题; 缺少设备整体运行情况; 缺少转换后结果核验和报表生成统计等。现有研究多聚焦于三防转换控制, 对日常设备状态一键恢复和运维数据分析重视不足, 亟需构建设备全生命周期状态监测并强化数据统计分析功能。尽管 WinCC 平台具备变量记录能力, 可通过定时任务实现设备运行状态记录, 但持续累积的数据将显著增加系统负荷, 不当的系统配置极易引发项目崩溃风险。同时, 前端设备种类繁多、通讯协议多样, 开发直连设备的监控软件导致开发周期难以控制、技术风险显著增加。而且 WinCC 在统计整体运行情况和智能报表生成方面存在功能局限, 难以满足智能化运维需求^[5-7]。基于上述原因, 提出基于现有 WinCC 架构的轻量化改造方案。通过 VBA (Visual Basic for Application) 脚本开发 OPC 客户端, 构建转换控制程序与 WinCC 的实时通讯, 通过 SyncWrite 函数实现了系统状态一键转换。利用 Excel 公式提取 WinCC 变量管理器数据, 梳理后台数据“变量池”, 采用订阅式读取策略, 实现了设备运行和通讯状态的实时感知。利用 ADO (ActiveX Data Objects) 技术, 创建与 Access 数据库的连接, 通过合理设置定时程序的循环写时间间隔和读时间区间以及动态数据维护机制 (定时备份与过期数据删除) 确保数据库查询效

率, 结合 VBA 编程和录制宏, 实现了转换结果核验和报表生成功能。该方案最大的优势在于无需修改现有 WinCC 项目底层架构, 通过外挂式程序既保障了原有系统的稳定性, 又实现了功能提升的技术目标。

1 环境保障及其控制系统介绍

1.1 系统组成及功能

该环境保障系统采用模块化设计, 由通风空调、给排水、供配电、防护密闭门以及控制系统共 5 个子系统构成。其中:

1) 通风空调系统集成空调除湿机组、风机、风阀等, 实现温湿度精准调控与三防通风模式切换;

2) 给排水系统包含水库、智能泵房、冷却塔及潜污泵集群, 通过 PLC 闭环控制实现冷却水循环、生活供水和消防用水应急保障;

3) 供配电系统部署市电和柴油机组发电双电路以及智能化照明网络;

4) 防护密闭门系统兼具车辆通行管控与冲击波和毒气防护双重功能;

5) 控制系统由西门子 S7-300/400/1500 系列 PLC、分布式 I/O 以及 WinCC 监控平台等组成, 负责汇总映射小型 S7-200 系列 PLC、采集数字量和模拟量信号以及实时监控系统的运行、事件追溯和故障报警等。

各子系统相互联系、协同工作。其中供配电系统为整个环境保障系统提供电能; 控制系统实现对环境保障系统的状态监控, 并在三防转换时实现对风、水、电设备的逻辑控制; 给排水系统主要为通风空调系统提供冷却供水; 通风空调系统主要实现对环境的温湿度控制, 并在三防转换时配合控制系统实现工作模式切换; 防护密闭门系统在三防转换时实现对有毒有害气体的密闭封锁。

例如, 在停电恢复时, 供配电系统首先恢复, 为环境保障系统提供电力支持, 关闭通风空调系统中自启动的除湿机和风阀等避免机组报故障和温湿度超标, 启动给排水系统为通风空调系统提供冷却水, 随后按预定顺序逐步恢复通风空调系统中除湿机、风机和风阀运行, 控制系统对环境保障系统实时监控, 流程如图 1 左图所示。

在三防转换时, 以清洁通风转换为隔绝通风为例, 首先关闭与外界关联的防护密闭门、通风空调系统中部分风机风阀、给排水系统中部分水阀等形成密闭环境, 供配电系统由市电切换为自备电站供电, 按顺序开启给排水和通风空调系统形成内部循环, 控制系统对三防转换实时监控, 流程如图 1 右图所示。

1.2 控制系统设计

控制系统采用“人机界面—通讯网络—过程控制”

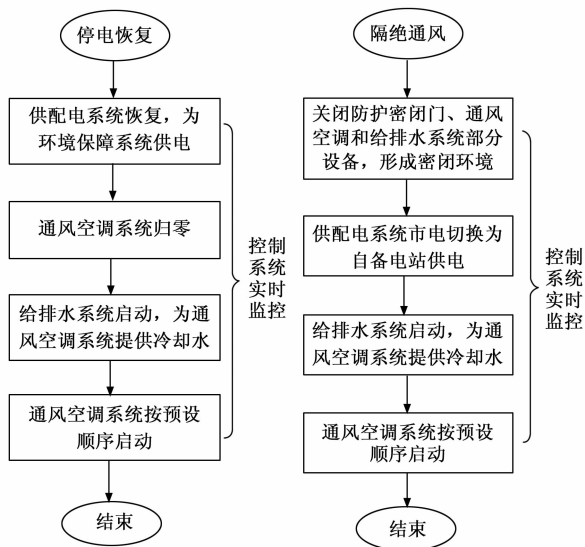


图 1 停电恢复和隔绝通风流程图

三层结构控制架构, 如图 2 所示。

1) 人机界面层基于客户机/服务器 (C/S, Client/Server) 架构部署 WinCC V7.3 监控平台, 采用双服务器热备冗余机制, 实现客户机远程数据交互与监控;

2) 通讯网络层通过西门子工业交换机构建网络, 支持 S7-1500/S7-300 主站与分布式 I/O 站等接入;

3) 过程控制层主站采用西门子 S7-300/400/1500 系列 PLC, 分站采用 ET200SP 分布式 I/O 模块实现多协议接入 (含三菱 FX1s 自由口协议、Modbus-RTU 设备), 通过 S7 协议集成 S7-200 智能化小系统, 并通过 OPC 协议集成 DVPS3000 电力监控^[8]和 iFIX 监控平台。

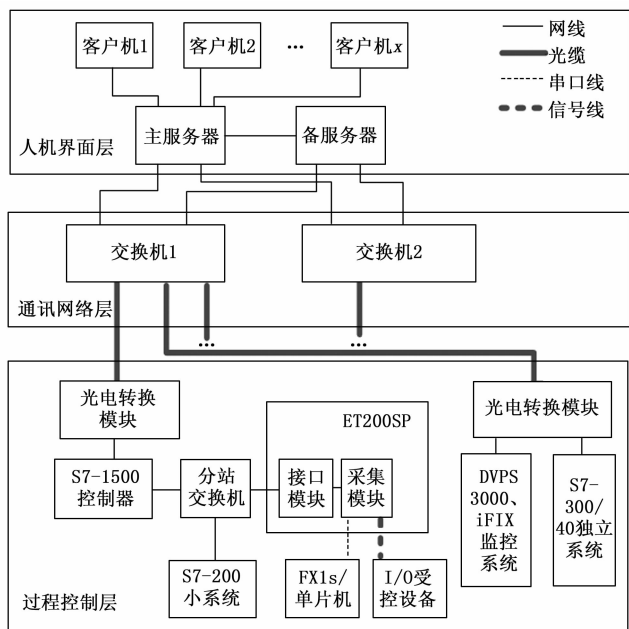


图 2 三层控制架构

1.3 环境保障系统运行痛点与功能需求

1) 常态运维挑战: 停电恢复效率瓶颈。环境保障系统在遭遇市电供电异常 (电网波动、电站维护等) 时, 系统进入停机状态。基于安全设计原则, 即便自备柴油机组或市电恢复供电, 各子系统仍保持停机状态。需要人工逐项恢复停电前运行参数, 造成温湿度调控响应延迟 (实测系统重启耗时 60 min 以上)。通风系统虽采用“一主一备”配置, 但因除湿机组相关联风阀采用手动控制, 无法通过 WinCC 平台实现远程切换, 维护人员需凭借经验手动操作或纸质记录远程逐台启动设备, 存在人力资源浪费和操作失误风险。

2) 应急需求: 三防转换协同控制。当遭受核生化武器袭击时, 需要快速、准确地实现清洁式、滤毒式和隔绝式通风的三防转换。该过程涉及风系统的风阀、风机和空调机组等, 水系统的水阀、冷却塔和冷却泵等, 以及防护密闭门等。转换为清洁式通风类似停电恢复, 转换为滤毒式和隔绝式通风需要各系统设备顺序联动控制, 控制逻辑复杂, 若是采用手动操作, 步骤繁琐、容易出错。

值得注意的是, 无论常态运维还是应急三防转换, 系统均存在时序批量控制、状态结果核验和报表生成等共同需求。

2 智能化转换控制程序实现

2.1 控制程序整体设计

控制程序基于 Excel 2010 利用 VBA 开发设计, 由人机界面、后台数据和报表共 3 张工作表以及 VBA 脚本组成, VBA 脚本又包括通讯连接、数据读写、数据库读写、定时延时和通讯监测等模块, 如图 3 所示。VBA 是微软公司专门针对 Word、Excel、Access 等 Office 应用软件而设计的基于 Visual Basic 语言的二次开发工具, 不但继承了 VB 的开发机制, 而且集成开发环境基本相同, 同时 Excel 可通过“录制宏”实现编程^[9-10]。VBA 脚本是控制程序的核心, “通讯连接”实现控制程序与 WinCC 通讯连接的建立和断开; “数据读写”实现控制程序与 WinCC 变量管理器中变量值同步, 并可批量向 WinCC 写入数值; “数据库读写”创建一个

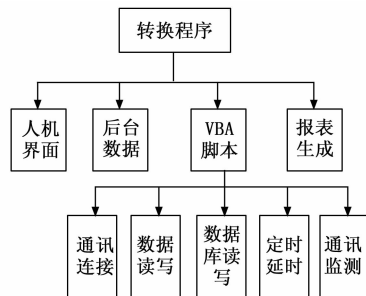


图 3 整体架构

Access 数据库, 将当前运行情况写入数据库, 并按时间读取数据库中运行情况, 并通过备份和删除的方式提高读取数据库的效率。“延时定时”实现定时循环和延时调用功能;“通信监测”实现控制程序与 WinCC 通信情况的实时监测。

利用 VBA 二次开发创建 OPC 客户端^[11], 实现与 WinCC “通讯连接”的建立和断开, 通过“数据写”和“延时”模块开发工艺批量处理指令, 实现系统状态一键转换; 通过“数据读”模块, 结合 Excel 强大的公式数据处理能力, 将实时值和质量代码转换为运行状态, 进而实现设备运行和通讯状态的实时感知; 通过“数据库读写”和“定时”模块, 实现历史数据的查询, 并与当前运行状态对比, 实现转换结果核验, 结合 VBA 编程和录制宏, 实现报表一键生成。

2.2 人机界面设计

停电一键恢复和三防一键转换人机界面, 基于工作表专为展示、控制及结果显示而设计, 界面采用直观的图形化展示, 操作按钮集中布局, 方便用户快速操作。同时, 通过颜色变化和图标提示, 实时反馈设备状态和操作结果, 提升用户体验。其结构布局如图 4 (a) 和 (b) 所示, 共涵盖 4 个板块:

1) 展示区。停电一键恢复界面动态展示了主站与独立智能设备 (诸如除湿机、潜污泵等) 实时通讯正常与异常数量以及百分比; 而三防一键转换界面则监视并展示了转换完成设备的数量与百分比。通过运用 COUNTIF 公式, 统计了通讯状态为“正常”的设备数量, 并以饼形图形式直观的实时呈现。

2) 控制区。用于与 WinCC 建立通讯、数据库读写、状态转换及转换结果核验等, 为确保操作安全, 现

场设备控制按钮设置了密码保护机制。唯有输入正确密码后, “右侧输入口令”单元格的背景颜色才由红色变为绿色, 按钮随之生效, 有效防止无权限人员或误操作引起的设备批量动作风险。

3) 结果核验区。在停电一键恢复界面, 系统能够对比当前与历史记录中运行设备及通讯异常设备的差异, 相同设备标记为“TRUE”, 减少的设备标记为“未启动”, 增加的设备标记为“多启动”, 以及通讯异常设备相对历史记录的“自行恢复”与“未恢复”情况。三防一键转换界面则专注于展示特定时刻参与设备、通讯异常设备及未完成转换设备的详细设备编号信息。

4) 程序运行监控区。该区域全面展示了数据的可调用起始时间、命令执行的开始时间以及预计执行时间。同时, 该区域还实时监视与 WinCC 通讯的心跳状态, 确保通讯连接正常。

2.3 后台数据处理

后台数据作为 Excel 与 WinCC 之间信息交互的桥梁, 其核心是由 WinCC 变量管理器中变量名称经过精心处理后构建而成的“变量池”。为了确保数据处理的准确性和高效性, 采取以下 3 步:

首先, 对参与转换的设备编号进行了全面梳理, 并按照命令、通讯、状态以及心跳变量的顺序进行排序。在命令变量中, 特意将“同一时刻动作设备尽量归并一处”(不区分先后顺序), 并尽量按照一键动作工艺的实际流程顺序进行排列; 状态变量则根据设备类别进行归类, 无特定先后顺序, 如图 5 所示。图中展示的实时数值, 正是与 WinCC 建立通讯后各变量对应的实时数据。系统通过 IF、AND 和 OR 公式对实时数值及质量代码进行综合解析, 实现对设备运行与通讯状态的精准判断。其核心逻辑如下: 除湿机的风机变量值若为 TRUE, 标记为“运行”, 否则记为“停止”; 除湿机通讯变量若质量代码≠192 或变量值=1, 则判定为“异常”, 否则为“正常”; 风阀开到位变量=1 且关到位变量=0 时, 判定为开到位 (程序中记为“运行”), 若开到位变量=0 且关到位变量=1, 则记为“关到位”, 其他情况均记为“未到位”。



(a) 停电恢复



(b) 三防转换

图 4 人机界面

	A	B	C	D	G	I	L	M
	变量名称	设备编号	位置	设备类型	实时数值	运行情况	通信情况	备注
1	AB001/EC001_CTRL_start_stop	EC001	PLCAB001	除湿机				序号1, 变频除湿机
2	AB001/EC002_CTRL_start_stop	EC002	PLCAB001	除湿机				
3	AB001/EC003_CTRL_start_stop	EC003	PLCAB001	除湿机				开机
4	AB001/EC004_CTRL_start_stop	EC004	PLCAB001	除湿机				
5	AB001/EC004_CTRL_start_stop	EC004	PLCAB001	除湿机				
219	AB002/FF001_stop_op	FF001	PLCAB002	风机				序号1, 新风机
220	AB002/FF002_stop_op	FF002	PLCAB002	风机				
221	AB002/FF003_stop_op	FF003	PLCAB002	风机				关机
222	AB002/FF004_stop_op	FF004	PLCAB002	风机				
223	AB002/FF005_stop_op	FF005	PLCAB002	风机				
415	X324_2_01P1	AB001	PLCAB001	主站		异常		主站通讯
416	X324_2_01P2	AB002	PLCAB002	主站		异常		
417	X324_2_02P1	AB003	PLCAB003	主站		异常		通讯
418	X324_2_02P2	AB004	PLCAB004	主站		异常		
588	AB001/EC001_fan_status	EC001	PLCAB001	除湿机		停止		除湿机空调机启动状态
589	AB001/EC002_fan_status	EC002	PLCAB001	除湿机		停止		
590	AB001/EC003_fan_status	EC003	PLCAB001	除湿机		停止		状态
591	AB001/EC004_fan_status	EC004	PLCAB001	除湿机		停止		
725	FF001_v_closed	FF001	PLCAB001	风阀				风阀运行情况 (运行代表开到位)
726	FF001_v_opened	FF001	PLCAB001	风阀		未到位		
727	FF002_v_closed	FF001	PLCAB001	风阀		未到位		
728	FF002_v_opened	FF001	PLCAB001	风阀		未到位		

图 5 后台数据

接下来,根据上述设备编号顺序,梳理并查找对应的变量名称、位置和设备类型等关键信息。为了简化变量创建过程,在 WinCC 项目中同一型号的设备多采用结构变量的形式,并在变量名称命名上遵循一定的规律。例如,变量“ABC01/KC001.CTRL_start_stop”通过点号被分为变量前缀和变量后缀两部分:变量前缀“ABC01/KC001”用于区分不同设备,其中“ABC01”表示站点或者控制柜位置,“KC001”表示设备编号(其中 KC 表示除湿机,001 为序列号一般用具体房间号表示);变量后缀“CTRL_start_stop”则明确指出该变量具体的含义。在 WinCC 变量管理器中,通过输入关键字如开关机“CTRL_start_stop”和“stop_op”,通讯“plc_comm”,以及状态相关的除湿机风机运行“fan_status”、风阀关到位“v_closed”、风阀开到位“v_opened”等,高效筛选出了所需变量,并将其复制到 Excel 电子表格中。随后,利用 Excel 中 MID 公式巧妙提取变量名称中的设备编号和设备位置信息,再结合 IF 公式根据设备编号准确判断设备类别。例如,通过下面的公式: =IF(MID(A2,7,2) = “KC”, “除湿机”, IF(MID(A2,7,2) = “WB”, “潜污泵”, “未知”)) 可提取 A2 单元格“ABC01/KC001.CTRL_start_stop”变量的设备类型。

最后,按照工艺顺序排序上述信息。灵活运用 INDEX 和 MATCH 公式,通过对设备编号进行排序查找,实现了信息的有序排列。例如,通过下面的公式:

=INDEX(Sheet1!A\$1:D\$5,MATCH(A2,Sheet1!B:B,0),1)

可在 Sheet1 中 A1:D5 绝对区域查找,行数为 A2 单元格数值与 Sheet1 中 B 列数值相等对应的行数,列数为第 1 列。

值得注意的是,这些公式只需输入一次,同一列的其他单元格可通过拖拽的方式自动填充公式。填充后检查填充结果,确保无误。若出现“#N/A”字样,则表示未找到对应的变量,需进行个别处理。对于极少数这种情况,设备编号对应的信息则通过 WinCC 图形编辑器逐一查找。经过上述一系列精心处理,最终得到如图 4 所示的“变量池”。

2.4 关键代码

2.4.1 通讯连接的建立和断开

梳理 Excel“变量池”后,Excel 转换程序与 WinCC 两个平台软件之间还需建立通讯连接,才能实现信息的交互^[12-13]。WinCC 通过其集成的 OPC 服务器实现跨平台数据交互,其内置的 Siemens OPC DAAutomation 2.0 控件(sopcaeauto.dll)允许外部应用程序(如 Excel)作为 OPC 客户机端访问 WinCC 内部/外部变量。Excel 通过引用 Siemens OPC DAAutomation 2.0

控件创建 OPC 客户端,基于“变量池”实现动态数据绑定。采用循环批量将 num 个“变量池”中变量名称 ItemIDs(i) 和序号 ClientHandles(i) 对应添加到项集合;计算机名采用自动读取 WinCC 系统变量的方式,提升程序可移植性,通讯连接建立(启动 WinCC OPC 按钮)的关键脚本如表 1 所示。

表 1 通讯连接建立关键脚本

代码	注释
For i = 1 To num	批量绑定“变量池”到 OPC 项集合
ClientHandles(i) = i	
ItemIDs(i) = Sheet2.Range(“a”& (i + 1)).Value	
Next i	
GroupName = “MyGroup”	设置组名
Set DSNName = CreateObject(“CCHMIRuntime.HMIRuntime”)	自动获取本地计算机名称
NodeName = DSNName.Tags(“@LocalMachineName”).Read	
ServerName = “OPCServer.WinCC”	设置服务器名
Set MyOPCServer = New OpcServer	添加 OPC 服务器
MyOPCServer.ConnectServerName, NodeName	连接到服务器
Set MyOPCGroupColl = MyOPCServer.OPCGroups	添加组集合到服务
MyOPCGroupColl.DefaultGroupIsActive = True	为组集合设置缺省的激活状态
Set MyOPCGroup = MyOPCGroupColl.Add(GroupName)	添加组到组集合
Set MyOPCItemColl = MyOPCGroup.OPCItems	添加项集合到组
MyOPCItemColl.AddItemsnun, ItemIDs, ClientHandles, ServerHandles, Errors	添加 num 个项到项集合
MyOPCGroup.IsSubscribed = True	订阅信息组激活

通过释放 OPC 对象资源安全断开通讯连接,通讯连接断开(停止 WinCC OPC 按钮)的关键脚本如表 2 所示。

表 2 通讯连接断开的关键脚本

代码	注释
MyOPCGroupColl.RemoveAll	与服务器断开连接并清除
MyOPCServer.Disconnect	
Set MyOPCItemColl = Nothing	
SSetMyOPCGroup = Nothing	
Set MyOPCGroupColl = Nothing	
Set MyOPCServer = Nothing	

2.4.2 数据读写

通讯连接成功建立后, Excel 便可从 WinCC 中获取数据, 即“数据读”。为了高效且实时捕获数据变动, 采用订阅式读取策略, 即在上一步通信连接建立时将 MyOPCGroup 的 IsSubscribed 属性设定为 TRUE^[12]。采用 DataChange 函数, 其中 NumItems 为“变量池”发生变化的数量, ClientHandles 为序号或项句柄, ItemValues 为值, Qualities 为质量代码, TimeStamps 为时间戳。例如, 若 NumItems 为 2, 即有 2 个变量发生变化, 那么 OPC 项句柄 (ClientHandles) 中第 1 个和第 2 个元素将会分别记录这两个编号变量的句柄, 同时, 它们的值也更新为 ItemValues 数组中对应元素值, 即 ItemValues (1) 和 ItemValues (2)。随后, 将读取到的结果写入变量名称对应行相应的单元格中。实时数据读取的脚本如表 3 所示。

表 3 实时数据读取脚本

代码	注释
Private Sub MyOPCGroup_DataChange (ByVal TransactionID As Long, ByVal NumItems As Long, ClientHandles() As Long, ItemValues() As Variant, Qualities() As Long, TimeStamps() As Date)	发生变化的数量 NumItems, 项句柄 ClientHandles, 值 ItemValues, 质量代码 Qualities, 时间戳 TimeStamps
For ii = 1 To NumItems	根据发生变化的变量个数, 获取对应的值 ItemValues 和质量代码 Qualities, 并分 别存入变量对应的行, 这里 分别存入表 sheet2 中 g 和 k 列, 行号为 ClientHandles 加 1 的单元格中
sheet2. Range("g"& (ClientHandles(ii) + 1)). Value = CStr(ItemValues(ii))	
Sheet2. Range("k"& (ClientHandles(ii) + 1)). Value = Qualities(ii)	
Next ii	
End Sub	

Excel 向 WinCC 中发送数据, 即“数据写”, 通过 SyncWrite 函数同步批量写入。在调取此函数前, 需要预先将要写入的数据存入 Values 数组中。鉴于在梳理“变量池”时, 遵循了“同一时刻动作设备尽量归并一处”的原则, 因此, 可以使用 for 循环批量为 Values 赋值。为了更便捷调用 SyncWrite 函数, 通常会设定一个固定不变的数据写入数量 (即命令变量总个数)。例如, 若命令变量的总数为 400 个, 第 200~400 个变量代表开机命令, 现对这部分设备进行开机操作。具体操作流程如下: 首先, 使用 for 循环给 Values 数组第 200~400 个元素赋值为 1, 然后利用 SyncWrite 函数写入这 400 个变量; 在延时一段时间后, 再次使用 for 循环将这些元素赋值为 0, 并重新利用 SyncWrite 函数进行写

入, 脚本如表 4 所示。这个过程类似于同时按下并随后松开这些设备开机按钮的操作当然, 若是恢复到某一时刻的运行状态, 还应调取并对比历史记录数据, 对历史记录中存在的 (即运行的) 设备对应的 Values 赋值为 1, 其余赋值为 0。

表 4 批量写入脚本

代码	注释
For i = 200 To 400 Values(i) = 1 Next i	第 200~400 个变量 赋值为 1, i 为变量序号。
MyOPCGroup. SyncWrite400, ServerHandles, Values, Errors	写入 400 个变量, 其中 1~199 写入 0, 200~400 写入 1。
Call delay(5000, 401)	调用延时函数 delay, 延时 5 秒, 并将写入时间记录在第 401 行
For i = 200~400 Values(i) = 0 Next i	第 200~400 个变量赋值为 0
MyOPCGroup. SyncWrite400, ServerHandles, Values, Errors	400 个变量全部写入 0

2.4.3 数据库读写

为了在停电或清洁通风后恢复设备的运行状态, 首先需要记录设备的运行情况, 以便在恢复时调用这些记录并与当前状态进行对比, 从而实现对设备的批量操作。记录设备运行情况的方法主要有两种: 一是利用 WinCC 现有数据库, 在变量记录中添加需要记录的变量。该方法需要对 WinCC 项目修改, 并对其数据库进行操作, 导致数据库记录数量大幅增加。再者, WinCC V6.2 以后, 数据均以压缩形式归档, 需要 WinCC 专门的查询语言进行访问^[14-15], 无法通过标准的 SQL 语句访问。二是新建一个独立的数据库, 该数据库与 WinCC 的数据库完全分离, 可以根据需求自定义表结构和字段, 可通过标准 SQL 语句访问^[16-17], 占用内存较小。这里采用第二种方法, 即新建 Access 数据库。Access 数据库是微软推出的一款轻量级关系型数据库, 介于 Excel 和专业数据库之间, 其安装包内置于 Office 套件中, 使用方便且功能强大。

首先, 新建一个名为“停电三防数据库. accdb”的数据库文件, 并在其中创建一张名为“三防”的表。于该表中添加“设备编号”(文本类型)、“通讯设备”(文本类型)和“写入时间”(日期/时间类型)三个字段。随后, 复制“三防”表结构, 并分别重命名为“停电”、“历史记录”和“空表格式备份”三张表。“停电”、“三防”、“历史记录”和“空表格式备份”四张表分别用于记录停电、三防和历史数据以及备份表的格式。

将需要写入数据库的变量分为通讯变量和非通讯变量。通讯变量用于监视设备通讯状态, 非通讯变量用于

监视设备运行状态。为了简化数据库记录，对通讯变量（如 PLC 通讯），仅记录通讯异常的设备编号及写入数据库的时间，并在“通讯设备”字段中标记为“是”。对于非通讯变量（如除湿机组运行和风阀开关到位），除湿机机组仅记录运行设备的编号，风阀仅记录开到位设备编号以及写入数据的时间，并在“通讯设备”字段中标记为“否”。在 Excel 中引用“Microsoft Activex Data Objects 2.0—6.1 Library”，利用 ADO 技术将图 5 后台数据中 L 列通讯异常数据写入数据库的关键程序脚本，脚本如表 5 所示。

表 5 写入 Access 数据库脚本

代码	注释
Set rst = New ADODB.Recordset	创建 Recordset 对象用于操作数据库记录
path = ActiveWorkbook.path&"\停电三防数据库.accdB"	将 Access 与 Excel 文件在同一目录下,自动获取 Access 文件路径
cnn.ConnectionString="Provider=Microsoft.ACE.OLEDB.12.0;data source=" + path + ";persist security info=false"	设置数据库连接字符串,office2007 版本以下将“ACE.OLEDB.12.0”替换为“Jet.OLEDB.4.0”
cnn.Open	打开数据库连接
ssql = “select * from 三防”	定义 SQL 查询语句
rst.Openssql, cnn, adOpenStatic, adLockOptimistic	打开记录集
For i = 415 To 587	
If Sheet2.Cells(i, “I”) = “异常”Then	检查 L 列是否为“异常”状态
rst.AddNew	在记录集中添加新记录
rst.Fields(1) = Sheet2.Cells(i, “b”)	设置第 1 个字段为设备编号
rst.Fields(2) = “是”	设置第 2 个字段为“是”
rst.Fields(3) = Date + Time()	设置第 3 个字段为写入数据的日期和时间
rst.MoveNext	移动记录指针
End If	
Next i	

在实际应用中，数据库的写入操作不可能时刻进行，而在查询某一时刻的历史记录时，用户选择的时间点可能与记录的时间点不完全一致，这可能导致数据记录了无法被查询到或查询到了但编号重复。为了解决这一问题，采用定时循环写入的方式，每 10 min 进行一次循环写入。在读取数据时，根据设定的时间查询该时间点往前 10 min 时间区间的数据，且该区间采用前开后闭的规则。同时，根据第一次和最近一次写入时间，利用公式自动计算可查询数据的时间区间，这样既保证了数据的可查询性又确保了数据的唯一性。以下是查询

历史数据的关键代码如表 6 所示。

表 6 查询 Access 数据库关键脚本

代码	注释
t_stop = Format(DTPicker1.Value, “yyyy-MM-ddhh:mm:ss”)	通过时间控件 DTPicker1 设定时间并格式化时间
t_start = Format(DateAdd("n", -10, t_stop), "yyyy-MM-ddhh:mm:ss")	开始时间为设定时间往前 10 分钟
ssql = "select * from 三防 where Format(写入时间, 'yyyy-mm-ddhh:mm:ss') > "&t_start&" and Format(写入时间, 'yyyy-mm-ddhh:mm:ss') <= "&t_stop&"	从数据库表“三防”中按“写入时间”字段查询,查询区间为前开后闭

随着数据库记录的增加，Access 数据库查询效率会急剧下降^[18]。经测试，当数据量超过 50 万条时，查询速度会明显变慢。为了提高查询效率，采用“备份—删除”的方式对数据库进行优化处理。采用 INSERT INTO 语句将数据插入到历史记录表，然后再通过 DELETE 语句从查询数据的表（如停电、三防）中删除相应数据，并通过定时循环调用这一备份机制，关键程序脚本如表 7 所示，其中 Now 为调用备份程序的时间。通常，定时循环时间应比 sStop 语句中设置的时间间隔更短，以减少处理的数据量。

表 7 优化数据库关键脚本

代码	注释
sStop = DateAdd("d", -2, CDate(Now))	计算截止时间:当前时间往前推 2 天
ssql = "INSERT INTO 历史记录 SELECT * FROM 停电 where 写入时间< #"&sStop&"#"	将"停电"表中早于截止时间的记录插入到"历史记录"表中,“##”用于转换为字符串
rst.Openssql, cnn, adOpenStatic, adLockOptimistic	执行 SQL 插入操作
ssql = "delete FROM 停电 where 写入时间< #"&sStop&"#"	删除"停电"表中早于截止时间的记录
rst.Openssql, cnn, adOpenStatic, adLockOptimistic	执行 SQL 删除操作
rst.Close	关闭记录集对象

2.4.4 定时延时

在数据库写入和备份过程中，需要用到定时循环操作，但 Excel VBA 开发环境未提供与 VB 类似的 timer 组件，可采用 Application.OnTime 方法实现定时功能^[19]。然而，实验发现当 TimeValue 参数设置大于 1 秒时，在调用终止循环函数 endtimer1 时易触发“方法‘OnTime’作用于对象‘_Application’时失败”的系统异常。

为解决这一问题，将 TimeValue 的值设为 1 s，并通过引入计数 i 来实现 10 min 定时循环，循环开始和结束脚本如表 8 所示。

表 8 定时脚本

代码	注释
Sub starttimer1()	定时开始函数
If $i < 600$ Then	计数 i 在 600(10 分钟)
$i = i + 1$	i 自增 1
Application. OnTime Now + TimeValue("0:0:1"), "starttimer1"	设置 1 秒后再次执行本函数(实现定时循环)
Else	
$i = 0$	满 10 min 清零 i
starttimer1	重新启动定时器
End If	
End Sub	
Sub endtimer1()	定时结束函数
On Error Resume Next	忽略运行时错误继续执行下一句(防止定时器未启动时报错)
Application. OnTime Now + TimeValue("0:0:1"), "starttimer1", , False	取消预定的 starttimer1 执行计划
$i = 0$	重置计数器 i 为 0
End Sub	

为了确保定时程序不被多次调用，限制定时程序仅可通过按钮触发一次。若多次按下按钮，会导致循环程序被重复调用，造成数据重复写入，严重影响程序执行效率。为此引入计数 i 监视循环运行状态。正常情况下，计数每秒递增一次；若发现计数递增速度异常加快，则停止循环并重新启动。

根据工艺要求，设备分批次操作时需要引入延时功能。采用 timeGetTime 函数^[20]结合 DO 循环实现延时，记录延时执行完毕时间，并引入 temp1 参数实现任意时间提前终止延时程序，关键程序代码如表 9 所示。

表 9 延时脚本

代码	注释
Sub delay(t As Long, n As Integer)	延时程序(需要延时的毫秒数 t , 记录执行时间的行号 n)
Dim time1 As Long	延时开始时间
time1 = timeGetTime	
Do	
If temp1 = True Then	temp1 为 TRUE 提前终止延时
Exit Do	退出
End If	
DoEvents	
Loop While timeGetTime - time1 < t	循环条件:当前时间与起始时间的差值小于指定延时 t
Sheet2.Cells(n , "h") = Date + Time()	
End Sub	

2.5 Excel 与 WinCC 通信监测

为实时监测控制程序与 WinCC 的通信状态，在 WinCC 侧创建布尔型内部变量 maichong，并编写全局动作脚本对该变量取反，通过 1S 标准周期触发器激活全局动作^[21]。将该变量添加在“数据池”后，通过监视其数值在 0 和 1 之间周期性变化判断通信是否正常。若通讯异常，则点击“启动 WinCC OPC”按钮，重新建立通信连接。

2.6 报表功能

为记录转换的起始时间、持续时长及未完成转换的设备信息，设计了 Excel 报表模版，报表模版支持一键记录相关数据，并通过录制宏将生成的报表另存为 PDF 格式，避免人为修改确保数据的真实性和可追溯性。生成的停电恢复报表和三防转换报表样式如图 6 (a) 和 (b) 所示。

停电报表2024-10-25 08: 06: 35							
温湿度统计时间		2024-10-18 16: ▾	2024-10-18 16: ▾	清空数据 生成报表			
停电时间		10/18 08:06:35	10/18 10:06:36				
停电时长		2:00:01					
报表生成时间		24/10/18 21:54:33					
通讯异常	3	VB002 VB003 VB004					
通讯未恢复	1	VB004					
通讯自行恢复	1	VB001					
设备未启动	1	KC001					
设备多启动	1	KC007					
温湿度探头超标	2	TH001 TH002					
重要工房温湿度详情							
工房及标准 探头编号		湿度(%)		温度(℃)			
		最小	最大	最小	最大		
		001(18-27, 35-75)	TH001	40	45	22	38
		002(10-25, 0-75)	TH002	45	47	24	26

(a) 停电恢复报表

	A	B	C	D	E
1	三防报表2024101316214				
2	参与设备	3	KC001 KC002 KC003		
3	通讯异常设备	6	WB001 WB002 WB003 WB004 WB005 WB006		
4	清洁通风	命令开始时间	11:09:27	命令结束时间	11:14:52
5	未完成设备	2	KC001 KC002		
6	隔绝通风	命令开始时间	09:21:59	命令结束时间	09:23:14
7	未完成设备	1	KC003		
8	现通讯异常设备	2	WB001 WB002		
9	报表生成时间		2024-10-13 16:21:04		

(b) 三防转换报表

图 6 报表模板

3 验证情况

3.1 模拟验证

在控制程序正式投入使用前,需先通过模拟验证以确保其功能性和稳定性。测试用例 1:模拟停电恢复,验证通讯连接、数据读写和数据库读写功能。

测试过程:于 Excel“数据池”中用变量 tag1、tag2 等代替实际设备变量(如 ABC01/KC001.CTRL_start_stop、ABC01/KC002.CTRL_start_stop 等);在 WinCC 中创建与“数据池”变量数量相同的内部布尔型变量,变量名称为 tag1、tag2 等,并组态输入/输出域以连接这些内部变量。激活 WinCC 项目后,观察通讯监测变量 maichong 在 0 和 1 之间跳动;在 Excel 程序中点击“启动 WinCC OPC”按钮后,若“数据池”中对应的变量 maichong 也在 0 和 1 之间跳动,则表明通讯正常。验证实时数据读取功能时,在输入/输出域中更改变量值,观察“数据池”中对应变量是否同步更新。点击 Excel 程序中批量操作按钮,若对应输入/输出域随之发生改变,则表明批量写数据功能正常,如图 7 所示。通过定时循环写入数据库数据,并设定时间读数据库数据,验证数据库的读写是否正常。

测试结果:通讯正常,数据同步更新,批量写入功能正常,数据库读写正常。

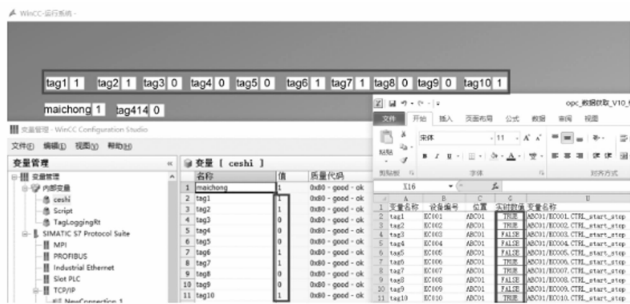


图 7 Excel 程序模拟验证

3.2 实际应用验证

模拟验证通过后,该控制程序在某厂区环境保障系统进行了实际应用验证。控制程序涉及的设备有 PLC 主站 * * 个、风机 * * 台、风阀 * * 台、除湿机 * * 台、污水泵 * * 台、水阀 * * 台、防护门 * * 道,测得恢复到停电前运行状态用时 5 分 20 秒 (5.33 min),转换为隔绝通风用时 1 分 15 秒 (1.25 min),转换为清洁通风用时 5 分 25 秒 (5.42 min),转换为滤毒通风用时 4 分 50 秒 (4.83 min),图 8 展示了控制程序在不同场景下的转换时间。实证数据表明,该控制程序在确保原有控制系统稳定运行的前提下,实现了状态转换、运行状态感知、结果核验和报表生成等功能。

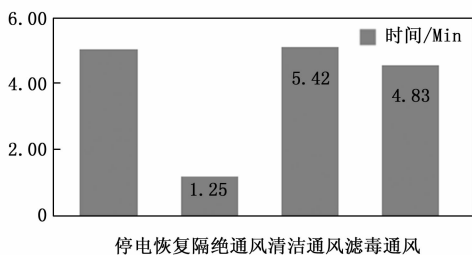


图 8 不同场景转换时间

4 结束语

针对厂区环境保障系统存在的状态转换人工操作复杂、耗时长、运行状态统计缺失、核验手段不足及报表功能不完善等问题。通过开发智能化转换控制程序,实现了系统状态的一键智能转换、实时通讯情况展示、转换结果核验及报表生成,显著提升了环境保障系统的自动化水平和状态转换效率。当前研究中,部分设备的远控集成尚未完成,影响了变量池的完整性。在未来的研究中,将进一步完善变量池,实现更多设备的远控集成,进一步提升系统的智能化水平,并采用用户窗口进一步优化人机界面。

参考文献:

- [1] 曹 晟,王昌栋,邢建春,等.基于 LORA 物联网的人防工程三防控制系统 [J].测控技术,2019,38(10):118-121.
- [2] 李超峰,毛 维,李 娟,等.防护通风系统现状及展望 [J].防护工程,2020,42(4):51-55.
- [3] 赵 硕,杨启亮,陈文杰,等.基于群智能图形化编程语言的三防转换控制程序设计与实现 [J].防护工程,2021,43(1):56-61.
- [4] 池行强,郭均瑶,肖 盼.基于 DDE 技术的 WinCC 与 VB 通讯研究 [J].工业控制计算机,2016,29(11):22-23.
- [5] 张 超,高 轶,刘俊华,等.某司锻轧线车轮轧机生产数据的本地存储方法 [J].制造业自动化,2023,45(11):152-156.
- [6] 霍迎科,杨 铮.基于 SQL SERVER 2005 的 WINCC 7.0 报表系统实现方法 [J].自动化技术与应用,2020,39(6):135-139.
- [7] 杨 林.基于 VBS 脚本及 SQL server 的 WinCC 报表二次开发 [J].电子制作,2020(22):63-64.
- [8] 王丰磊,程国防,魏宏建,等.阵地电力系统跨平台集中监测设计实现 [J].计算机测量与控制,2023,31(9):36-41.
- [9] 杨 洋.深入浅出 Excel VBA [M].北京:电子工业出版社,2019.

(下转第 248 页)