



教育部
高等职业教育示范专业规划教材
(电气工程及自动化类专业)

工业控制组态软件 及应用

主编 许志军



教育部高等职业教育示范专业规划教材

(电气工程及自动化类专业)

工业控制组态软件及应用

主 编 许志军

副主编 程龙泉 胡汉辉 刘宝玖

参 编 满海波 向守均

主 审 谢永春



机械工业出版社

本书精选了目前广泛应用的 WinCC 5.0 和 MCGS 两大软件。全书共分 8 章, 内容包括组态软件概述、WinCC C 语言基础、变量的组态、画面组态、WinCC 编辑器、组态软件通信、组态软件网络操作系统、MCGS 组态软件及应用。

本书在教学实践的基础上编写而成。在编写过程中, 重点突出实用性、适用性和先进性。本书通俗易懂、条理清晰, 既有对菜单命令的详细讲解, 又有大量精选例题, 以便进行实际操作训练。

本书既可作为高职高专电类相关专业工业控制组态软件方面的教材, 也可供进行工业控制组态软件设计的工作人员参考。

图书在版编目 (CIP) 数据

工业控制组态软件及应用/许志军主编. —北京: 机械工业出版社, 2005.6
教育部高等职业教育示范专业规划教材. 电气工程及自动化类专业
ISBN 7-111-16743-0

I. 工 ... II. 许 ... III. 过程控制软件—高等学校: 技术学校—教材
IV. TP317

中国版本图书馆 CIP 数据核字 (2005) 第 063686 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

责任编辑: 于宁

责任印制: 洪汉军

北京原创阳光印业有限公司印刷·新华书店北京发行所发行

2005 年 7 月第 1 版第 1 次印刷

787mm×1092mm 1/16·16.25 印张·401 字

定价: 24.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

本社购书热线电话 (010) 68326294

封面无防伪标均为盗版

前 言

WinCC 5.0 和 MCGS 两大软件是目前广泛应用的工业控制组态软件。SIMATIC WinCC 是第一个使用最新的 32 位技术的过程监视系统,具有良好的开放性和灵活性。无论是单用户系统,还是冗余多服务器/多用户系统,WinCC 均是最佳选择。通过 ActiveX、OPC、SQL 等标准接口,WinCC 可以非常方便地与其他软件进行通信。WinCC 还提供了许多功能强大的选件,如实现在工业应用中通过 Internet 监视生产过程的网页浏览器可选软件包,连接 MES 和 ERP 可选软件包等。MCGS 是全中文工业自动化控制组态软件,MCGS 可稳定运行于 Windows95/98/NT 操作系统,集动画显示、流程控制、数据采集、设备控制与输出、网络数据传输、双机热备、工程报表、数据与曲线等诸多强大功能于一身,并支持国内外众多数据采集与输出设备。

全书共分 8 章,第 1~7 章详细介绍了 WinCC 5.0 监控软件的界面、菜单、组态方法及实际应用;第 8 章详细介绍了 MCGS 组态软件的界面、菜单、组态方法及实际应用。内容包括:第 1 章 组态软件概述,第 2 章 WinCC C 语言基础,第 3 章 变量的组态,第 4 章 画面组态,第 5 章 WinCC 编辑器,第 6 章 组态软件通信,第 7 章 组态软件网络操作系统,第 8 章 MCGS 组态软件及应用。

本书的第一个特点是它全面地介绍了 WinCC 5.0 和 MCGS 的功能,几乎每一个菜单命令都得到详细介绍;第二个特点是精选了 WinCC 5.0 和 MCGS 的实际应用实例。

本书可作为高职高专院校工业控制组态软件课程的教材。课程教学安排为:课堂教学 26 学时,实际操作练习 14 学时。

本书也可以作为课程设计的教材,适合多于一周时间的课程设计,在教师给出设计题目后,学生进行工业控制组态设计。

全书由许志军副教授主编并统稿,程龙泉副教授、胡汉辉副教授、刘宝玖副教授担任副主编。第 1、2 章由程龙泉编写,第 3 章由程龙泉、刘宝玖编写,第 4 章由程龙泉、许志军编写,第 5 章由满海波编写,第 6、7 章由向守均编写,第 8 章由胡汉辉编写。

全书由攀钢集团公司谢永春博士担任主审。他对本书提出了许多宝贵意见,在此表示衷心的感谢。

在本书编写过程中还得到许多朋友和同事的帮助,在此表示感谢。

由于作者水平有限,书中内容不免存在疏漏和不妥之处,敬请读者批评指正。

编 者

目 录

前言

第1章 组态软件概述.....	1
1.1 概述.....	1
1.2 组态项目实例的规定.....	2
1.2.1 WinCC 项目名称.....	2
1.2.2 脚本和动作.....	2
1.2.3 用户界面.....	2
1.2.4 控制概念.....	3
1.2.5 更新周期和用户权限.....	4
1.2.6 报警.....	4
1.2.7 关于执行过程.....	4
1.3 用 WinCC 组态时的特性.....	5
1.3.1 怎样设置更新周期.....	5
1.3.2 关于更新周期应用的信息.....	5
1.3.3 在 WinCC 中添加动态.....	6
1.3.4 画面模块技术.....	9
1.4 项目管理器.....	12
1.4.1 启动.....	12
1.4.2 WinCC 项目管理器的结构.....	12
1.4.3 项目类型.....	14
1.4.4 计算机的属性.....	15
1.4.5 创建项目的步骤.....	16
1.4.6 激活项目.....	17
1.4.7 组态一个项目的步骤.....	19
1.4.8 WinCC 的编辑器.....	19
1.4.9 WinCC 基本选项.....	20
1.4.10 变量管理器.....	20
1.4.11 图形编辑器.....	22
第2章 WinCC C 语言基础.....	24
2.1 C 脚本的开发环境.....	24
2.1.1 图形编辑器的动作编辑器.....	24

2.1.2 全局脚本编辑器.....	27
2.2 变量.....	30
2.2.1 整数数据类型.....	31
2.2.2 整数 WinCC 变量.....	33
2.2.3 浮点数数据类型.....	34
2.2.4 浮点数 WinCC 变量.....	35
2.2.5 静态变量和外部变量.....	35
2.3 C 中的运算符和数学函数.....	36
2.3.1 基本的数学运算.....	37
2.3.2 自增和自减运算符.....	38
2.3.3 位运算.....	39
2.3.4 按字节循环移动.....	40
2.3.5 数学函数.....	41
2.4 指针.....	41
2.4.1 指针例.....	42
2.4.2 向量.....	43
2.4.3 指针与向量.....	43
2.4.4 字符串.....	44
2.4.5 WinCC 文本变量.....	45
2.5 循环和条件语句.....	45
2.5.1 循环概述.....	45
2.5.2 while 循环.....	47
2.5.3 do-while 循环.....	47
2.5.4 for 循环.....	48
2.5.5 无限循环.....	49
2.5.6 if-else 语句.....	50
2.5.7 switch-case 语句.....	50
2.6 函数.....	51
2.6.1 数值参数的传递.....	52
2.6.2 所传送地址域的写入.....	52
2.6.3 结果地址的返回.....	54

2.7 结构	55	3.3.4 通过复选框进行输入	87
2.8 WinCC API	57	3.4 对字中的位进行处理	88
2.8.1 通过 RT 函数创建一个变量连接	58	3.4.1 直接通过复选框和直接 连接进行置位	88
2.8.2 通过 CS 函数修改属性	59	3.4.2 选择一个位并更改其状态	89
2.9 项目环境	61	3.5 变量的间接寻址	91
2.9.1 项目文件的确定	61	3.5.1 通过直接连接进行间接寻址	91
2.9.2 确定用户名	62	3.5.2 使用间接寻址和 C 动作 进行多重显示	92
2.10 Windows API	62	3.5.3 使用 C 动作进行间接寻址	93
2.10.1 设置 Windows 属性	63	3.6 结构变量的使用	94
2.10.2 读取系统时间	63		
2.10.3 播放声音文件	64		
2.11 标准对话框	65	第 4 章 画面组态	96
2.11.1 变量选择	65	4.1 画面布局	96
2.11.2 出错框	66	4.2 画面切换	96
2.12 文件	66	4.2.1 实例 1: 通过直接连接 打开画面	96
2.12.1 保护数据	67	4.2.2 实例 2: 利用动态向导 打开画面	99
2.12.2 读取数据	68	4.2.3 实例 3: 通过内部函数 打开画面	100
第 3 章 变量的组态	70	4.2.4 实例 4: 利用直接连接 切换单个画面	101
3.1 变量的创建、分组和移动	70	4.2.5 实例 5: 通过对象名称和画面 名称的变量连接打开画面	101
3.1.1 变量组与变量	70	4.3 显示画面窗口	101
3.1.2 移动变量	71	4.3.1 实例 1: 画面窗口的隐藏 (撤消选择) 和显示 (选择)	102
3.1.3 WinCC 变量	71	4.3.2 实例 2: 对画面进行时控隐藏	103
3.1.4 结构变量	72	4.3.3 实例 3: 使用动态向导对 信息框进行组态	104
3.2 递增、递减、单击	73	4.3.4 实例 4: 显示用于文本 输入的对话框	106
3.2.1 更改设定值	74	4.4 操作控制权限	107
3.2.2 通过全局脚本更改设定值	75	4.4.1 实例 1: 退出运行系统或 整个系统	107
3.2.3 按钮	76	4.4.2 实例 2: 根据用户授权 执行画面切换	108
3.2.4 切换开关	78		
3.2.5 递增和递减	78		
3.2.6 通过全局脚本递增和递减	80		
3.3 通过 Windows 对象对变 量值进行修改	84		
3.3.1 通过带有直接连接的 滚动条进行输入	84		
3.3.2 通过滚动条和变量 连接进行输入	85		
3.3.3 通过选项组进行输入	85		

4.5 画面缩放.....	110	6.1.6 总线系统的连接.....	153
4.5.1 实例 1: 在两种尺寸之间 改变画面几何结构.....	110	6.2 通信网络.....	154
4.5.2 实例 2: 连接更改 画面几何结构.....	111	6.2.1 工业通信概述.....	154
4.6 Windows 控制中心.....	113	6.2.2 工业通信子网.....	154
4.6.1 实例 1: 操作面板的访问.....	113	6.2.3 利用 MPI 的工业通信.....	154
4.6.2 实例 2: 自动输入检查.....	113	6.2.4 利用 PROFIBUS 的 工业通信.....	155
4.7 动态化.....	115	6.2.5 利用工业以太网的通信.....	157
4.7.1 实例 1: 颜色更改.....	115	6.2.6 OPC 接口标准.....	157
4.7.2 实例 2: 文本切换.....	116	6.3 通信组态.....	158
4.7.3 实例 3: 移动过程的动画.....	117	6.3.1 WinCC 过程通信.....	158
4.7.4 实例 4: 利用 C 动作创建 移动过程的动画.....	117	6.3.2 WinCC 通信组态.....	158
4.7.5 实例 5: 使用向导创建 移动过程的动画.....	118	6.3.3 通信方案选择.....	160
4.7.6 实例 6: 通过 C 动作更改颜色.....	119	6.3.4 规范.....	161
4.8 信息的显示和隐藏.....	120	6.3.5 组态注释.....	163
4.8.1 实例 1: 显示和隐藏对象.....	120	6.3.6 性能和数据.....	164
4.8.2 实例 2: 日期和时间的显示.....	121	6.4 通信连接的诊断.....	168
第 5 章 WinCC 编辑器.....	122	6.4.1 错误检测.....	168
5.1 变量记录.....	122	6.4.2 故障消除指南.....	169
5.1.1 周期连续的归档.....	122	第 7 章 组态软件网络操作系统.....	176
5.1.2 周期选择归档.....	131	7.1 多客户机系统.....	176
5.2 报警记录.....	137	7.1.1 多客户机的应用.....	176
5.2.1 创建、组态消息.....	137	7.1.2 服务器数据(数据包).....	176
5.2.2 组态 WinCC 报警显示.....	141	7.1.3 WinCC 多客户机项目中数据包.....	177
5.2.3 消息窗口.....	145	7.1.4 Project_MultiClient_Server 项目的创建.....	177
5.2.4 消息归档.....	147	7.1.5 Project_MultiClient_Client 项目的创建.....	179
第 6 章 组态软件通信.....	151	7.2 分布式服务器.....	181
6.1 通信基础.....	151	7.2.1 Project_DisServer_Server 项目的创建.....	182
6.1.1 通信.....	151	7.2.2 Project_DisServer_Client 项目的创建.....	184
6.1.2 网络拓扑.....	151	7.3 冗余控制.....	186
6.1.3 网络的分类.....	152	7.3.1 冗余的操作.....	186
6.1.4 访问方式.....	152	7.3.2 冗余用户归档.....	187
6.1.5 ISO-OSI 参考模型.....	152	7.3.3 Project_Redundancy_Server 项目的创建.....	188

第 8 章 MCGS 组态软件及应用.....	192	8.3.6 立体文字画面的制作.....	206
8.1 了解 MCGS 组态软件.....	192	8.3.7 水位控制系统动画制作.....	207
8.1.1 MCGS 组态软件的功能和特点	192	8.3.8 图像动起来画面增光彩.....	214
8.1.2 MCGS 组态软件的系统构成	193	8.3.9 编写控制流程.....	220
8.1.3 MCGS 组态软件的工作方式	194	8.3.10 曲线与数据显示.....	222
8.2 学习 MCGS 组态软件.....	195	8.3.11 报警显示.....	225
8.2.1 MCGS 组态软件常用术语	195	8.3.12 如何制作工程报表.....	227
8.2.2 MCGS 组态软件的操作方式	196	8.3.13 完善菜单按钮功能.....	233
8.2.3 MCGS 组态软件鼠标操作	197	8.3.14 工程运行与调试.....	234
8.2.4 组建工程的一般过程.....	198	8.3.15 建立安全机制.....	235
8.3 MCGS 工程实例详解.....	198	8.3.16 保护工程文件和用户权益.....	238
8.3.1 样例工程简介.....	199	8.3.17 完成设备连接.....	239
8.3.2 样例工程剖析.....	199	8.4 MCGS 在立体仓库中的应用.....	242
8.3.3 样例工程建立.....	200	8.4.1 概述.....	242
8.3.4 构造工程框架.....	201	8.4.2 立体仓库组态窗口.....	245
8.3.5 设计系统菜单.....	204	8.4.3 存取车策略.....	249

第 1 章 组态软件概述

1.1 概述

WinCC 是进行廉价和快速组态的 HMI 系统,从其他方面看,它是可以无限延伸的系统平台。WinCC 的模块性和灵活性为规划和执行自动化任务提供了全新的可能。

从组态的角度上来看,在 WinCC 中有三种解决方案:① 使用标准 WinCC 资源的组态;② 利用 WinCC 通过 DDE、OLE、ODBC 和 ActiveX 使用现有的 Windows 应用程序;③ 开发嵌入 WinCC 中的用户自己的应用程序(用 Visual C++或 Visual Basic 语言)。

WinCC 是基于 Microsoft 的 32 位操作系统(Windows NT4.0, Windows 2000 和 Windows 2000 XP)。该操作系统是 PC 平台上的标准操作系统。

WinCC 为过程数据的可视化、报表、采集和归档以及为用户自由定义的应用程序的协调集成提供了系统模块。此外,用户还可以合并自己的模块。

WinCC 的特点如下:

1. WinCC 的开放性 WinCC 对用户所添加的任何形式的扩充是绝对开放的。该绝对开放性是通过 WinCC 的模块结构及其强大的编程接口来获得。

2. 将应用软件集成到 WinCC 中 WinCC 提供了一些方法将其他应用程序和应用程序块统一地集成到用于过程控制的用户界面中。OLE 应用程序窗口和 OLE 自定义控件(32 位 OCX 对象)或 ActiveX 控件可以集成到 WinCC 应用软件中,就好像是真正的 WinCC 对象一样。

3. WinCC 中的数据管理 WinCC 中的默认数据库 Sybase SQL Anywhere 从属于 WinCC,该数据库用于存储(事务处理保护)所有面向列表的组态数据(例如变量列表和消息文本),以及当前过程数据(例如消息、测量值和用户数据记录)。该数据库具有服务器的功能,WinCC 可以通过 ODBC 或作为客户通过开放型编程接口(C-API)来访问数据库,也可以将同样的权限授予其他程序。因此,不管应用程序是在同一台计算机上运行,还是在联网的工作站上运行,Windows 中的应用程序均可访问 WinCC 数据库的数据资源,在数据库查询语言 SQL 和相关连接的工具(例如 ODBC 驱动程序)的帮助下,其他客户端程序(例如 UNIX 数据库,Oracle、Informix、Ingres 等)也可以访问 WinCC 数据库的数据资源。

4. 在项目开始之前规定组态分类 在项目开始之前,组态规定分为:WinCC 项目的名称,变量的名称,WinCC 画面的名称,创建脚本和动作的规则,组态规则(共同标准、库函数、按组工作),归档项目和方法。

运行项目的规定:这些规定很大程度上取决于应用领域(例如冶金、汽车工业、机械制造等)。规定有:用户界面(画面安排、字体和字体大小、运行语言、对象显示等);控制概念(画面体系、控制原理、用户权限、有效键操作等);用于消息、限制值、状态、文本等的颜色;通信模式(连接类型、更新的周期和类型等);数量表(报警、归档值、趋势、客户端程序等的数目);消息和归档的方法。

1.2 组态项目实例的规定

制定用于组态项目实例的规定。创建用户自己的项目时, 这些规定可以作为模板类型使用。在大多数 WinCC 的编辑器中, 可以将某些属性设置为默认值。因此, WinCC 支持用户组态的特定样式, 从而能为指定的任务进行最优化的组态。

1.2.1 WinCC 项目名称

项目名称作为文件夹的默认名称, WinCC 项目中的所有数据都存储在该文件夹中, 但可以在创建项目初期或以后改变文件夹名称 (从 Windows Explorer 中)。除一些特殊字符 (例如 \? ! ; /) 之外, 文件夹名称允许使用所有的字符。还允许使用数字 0~9。

1. 变量名称 变量名称可以多于 8 个字符。但应尽量避免太长的名称。分配变量名称时应严格按照规则进行, 因为在组态时, 变量管理器的结构对于确保在运行时快速而有效地组态和高性能处理至关重要。在定义变量名称之前, 需要考虑一些与 WinCC 中变量管理器的结构有关的特殊字符。创建组态能影响组态时变量在变量管理器中的显示方式。组态名称将影响变量名称的惟一性。用于 WinCC 项目的变量名称必须是惟一的。系统将检验其惟一性。WinCC 可以帮助用户用不同的方法选择变量, 例如通过按列排序 (名称、创建日期等) 或通过使用过滤器。如果变量名称还包含了其他信息, 这对用户将非常有用。

2. 画面名称 如果想要在脚本或外部程序中寻址画面, 则使用固定的画面名称会非常有用。但是, 也需要考虑如何确定画面名称的长度。太长的名称 (文件名) 不容易识别 (列表框中的选择、脚本中的调用等)。根据经验表明, 长度最好不超过 28 个字符。画面名称应遵守以下限制条件: ① 最大长度为 255 个字符; ② 不使用某些特殊字符 (例如 /“\ : ? ”); ③ 画面名称中的字母不区分大小写。

1.2.2 脚本和动作

可以在 WinCC 项目中创建自己的脚本和动作。分配的名称应具有一定含义, 这样在以后使用脚本时会比较方便。在全局脚本编辑器中组态时, 使用比例字体可能会带来麻烦。因此, 选择宽度为常数字符的字体 (例如 Courier) 较容易读取。应为脚本配上恰当的注释, 便于帮助理解而且花费的时间不多。

1.2.3 用户界面

非常仔细地建立用户界面极为重要。在图形编辑器中创建的所有对象画面将显示在用户工作空间的画面上。所创建的画面是机器和用户之间的惟一界面, 也是操作员 (客户) 每天查看的画面。

画面显示系统中, 设备当前状态的信息通过显示画面单独呈现给用户, 因此界面画面显示必须尽可能地提供全面的以及容易理解的信息。WinCC 能够根据用户的要求准确地组态用户界面。可根据使用的硬件、处理要求和已有的规定来设置自己特定的用户界面。

组态用户界面时, 最需要关注的是用户, 因为组态毕竟是为用户而服务的。如果能成功而又清楚地提供给用户所需要的信息, 则会提高产品质量和减少故障。同时还能简化必要的维护工作。

1. 用户需要获得尽可能多的信息 使用这些数据作为基础,用户能够作出重要决定以保持过程以高质量运行。用户主要的工作不仅是对报警作出响应(此过程在这里已经不需要考虑),而是运用其经验、过程知识和操作系统提供的信息来预知过程发展的方向。用户应该能够在不规则事件发生前进行阻止。WinCC 提供了有效编辑和向用户显示这类信息的可能。

当确定应集成到画面的信息量时,为达到平衡,需要权衡以下两方面情况:

1) 如果画面包含的信息太多,读取信息就会有困难,而且信息的搜索可能会花费较多的时间,用户发生错误的概率也会随之增加。

2) 如果画面包含信息太少,将会增加用户的工作量。用户可能会找不到过程,只能频繁地切换画面以找到所需信息。这样会导致延迟响应、控制输入以及过程控制的不稳定。调查显示,有经验的用户希望每个画面中包含尽可能多的信息,从而不需要经常切换画面。

相反,如果一个画面中包含了太多的信息,对于初学者而言则会感到困惑,不能确定应该怎么做,他们可能找不到正确的信息或不能及时找到信息。

2. 显示的信息应该重要并容易理解 可以在某段信息(例如测量点标识符)不需要时将其隐含。显示信息模拟量数值时,可以用数字化的指针仪表来表示,数值的图形表达(例如指针仪、棒图……)可以使用户更容易、快捷地识别和掌握信息,且画面中包含的重要信息应总能立即识别出,因此颜色对比度的运用至关重要。

人眼识别颜色的速度要比识别文本快。使用颜色编码能够非常快速地建立各种对象的当前状态,但是必须建立并始终遵循颜色编码图表,用来显示项目状态的统一颜色必须采用统一的习惯标准。

为使文本更易于阅读,应该遵守如下一些简单的规则:

1) 文本的大小必须与文本中所包含信息的重要性相匹配,而且需要考虑用户与屏幕之间的距离。

2) 先使用小写字母。虽然大写字母在较远距离时也能阅读,但小写字母要比大写字母少占空间并易于读取。

3) 水平文本比垂直或斜置文本更易于读取。

4) 不同的信息类型使用不同的字体(例如测量点名称、注释等)。

无论决定使用什么概念,都要坚持让它贯穿整个项目。这样,就可以对过程画面进行直观控制,用户错误就会较少发生,这也同样适用于所使用的对象。例如,无论是在什么画面中,所绘制的电动机或泵总是一样的。

如果使用的是标准的 PC 监视器,则将画面分为三部分:总览部分、工作空间部分以及按钮部分。如果在特殊工业 PC 上或具有集成功能键的操作面板上运行应用程序,则对画面内容进行分割的方法并不一定适用。屏幕分成总览、按钮和空间画面三部分,各个画面的大小可以根据需要在固定范围内设置(最小 1×1 像素,最大 4096×4096 像素)。如果单用户系统使用 17" 监视器,则推荐使用的最大分辨率为 1024×768 像素;对于多用户系统(多台 VGA),常会使用较高的分辨率。

1.2.4 控制概念

用户使用常用的输入设备(如键盘、鼠标、触摸屏或工业操作杆)可以在 WinCC 下控制过程应用程序。如果计算机处于极个别不能使用的工业环境中,则可以组态 tab 顺序和 alpha

光标。通过 tab 顺序在可控制领域之间移动,通过 alpha 光标在输入域之间移动,可以锁定每个控制操作以防止未经授权的访问。

选择画面的概念取决于多个因素,最主要的因素是显示的画面数和过程结构。在较小的应用程序中,可将画面设计为循环或 FIFO 缓冲器。如果运行大量画面,则必须设计一个合理的结构体系来打开画面。结构体系使过程容易理解、能够简便地处理和提供信息并且快速访问详细信息。通常频繁使用的结构体系主要由三个层面组成:①层面 1 是总览画面,该层面主要包含不同的系统部分在系统中所显示的信息以及如何使用这些系统部分协同工作,还显示在较低层面是否有事件(消息)发生;②层面 2 是过程画面,该层面包含指定的过程部分的详细信息,并显示哪个设备对象属于该过程部分,该层面还显示了报警对应的设备对象;③层面 3 是详细画面,该层面提供各个设备对象的信息,例如控制器、阀、电动机等,并显示消息、状态和过程值。如果合适,则还包含与其他设备对象交换的信息。

1.2.5 更新周期和用户权限

1. 更新周期 当确定更新周期时,始终要从整体上考虑系统:需要考虑更新什么以及更新几次。不恰当的更新周期对 HMI 系统的性能具有反作用。当着眼于整个系统时(PLC-通信-HMI),可以在过程中(PLC)检测到改变的发生。在大多数情况下,总线系统是数据传输的瓶颈。当指定测量值的更新模式时,需要关注测量实际改变的速度。例如,对于容量为 5000L 的锅炉的温度控制,则以 500ms 的时间间隔进行实际值的更新是毫无意义的。由于 WinCC 是基于 Windows NT 的 32 位 HMI 系统,该操作系统已为事件驱动的控制操作进行优化,所以如果组态 WinCC 时考虑上述因素,则即使在处理大量的数据时都不会有性能方面的问题。

2. 用户权限 当操作设备时,需要保护某些操作员功能以防止未经授权的访问。进一步的要求是只有专门的人员才可以访问组态系统。可以指定用户和用户组,并在用户管理员中定义各种授权等级,这些授权等级可以连接到画面中的控制元素。可以分配基于个人的不同授权等级的用户组 and 用户。

1.2.6 报警

位消息步骤可以是任何自控系统报告的通用步骤。WinCC 监控所选择的二进制变量的信号边缘变化并产生消息事件。序列报表要求自控系统本身能够生成消息并以预定义格式向 WinCC 发送可能带有时间标志和过程值的消息。通过消息的操作步骤可以给来自不同自控系统的消息序列排序。当指定要报告所有事件和组态时,大多数人会根据其认为最安全的方式进行操作,设置软件来报告所有事件和状态改变。这样就要由用户来确定其首先要看的消息。如果在设备中报告的事件太多,则经验告诉我们只选择重要的消息,避免有的消息来不及查看。

1.2.7 关于执行过程

执行项目时使用固定的结构来存储数据是特别有用的,在包含相关子文件夹的一个文件中存储项目的所有数据,这样在处理项目时具有优势,在备份数据时更是如此。指定文件夹时,除由 WinCC 创建的文件夹外,可根据需要为 Word、Excel 和临时文件创建其他的文件夹。

1.3 用 WinCC 组态时的特性

1.3.1 怎样设置更新周期

指定更新周期是在可视化系统中执行的最重要的设置步骤之一。设置将影响下列属性：画面结构，可视化站（图形编辑器）上当前打开的画面中的对象更新，后台脚本（全局脚本）的处理，数据管理和过程通信的激活，当测量值根据归档次数处理（变量记录）时设置其他的时间变量。

当前的变量值由数据管理（变量管理的主要管理器）根据设置的更新周期来请求。数据管理通过通信通道获得新过程数据，然后将这些数值提供给应用程序。

更新画面中对象的各个属性指的是打开画面之后已经动态化的对象。更新周期的任务是建立画面中特定对象的当前状态。组态人员或系统可以为下列动态类型设置动态化对象的更新周期：

- 1) 组态对话框：变量触发 2s 或事件触发（例如控件），自定义时间周期。
- 2) 动态向导：可以根据动态类型选择（事件触发、时间周期、变量触发），自定义时间周期、事件或变量。
- 3) 直接连接：事件触发。
- 4) 变量连接：变量触发 2s，自定义时间周期。
- 5) 动态对话框：时间周期 2s，自定义时间周期、变量触发器。
- 6) 关于属性的 C 动作：时间周期 2s，自定义时间周期、变量触发器，直接从 PLC 读取。
- 7) 对象属性：设置取决于动态编辑更新周期列。

要选择的更新由 WinCC 指定，并且可以由用户定义的时间周期来补充。

1.3.2 关于更新周期应用的信息

对于更新周期的应用，根据所选的周期类型，推荐使用下列设置，见表 1-1。

表 1-1 更新周期的设置

类 型	默认设置（时间）/s	推荐的组态
默认周期	2	动态对话框或 C 动作：如果变量是互相影响的，应该在所有的事件中使用变量触发。这样可以减少任务数量的变化和任务间的通信。变量触发设置为“一旦改变”只能有选择地使用，因为它会引起较大的系统负载！变量的变化情况会不断接受检查。这种循环检测机制总是会导致较大的系统负载。建议标准对象使用 1~2 s 的周期
时间周期	2	根据对象类型或对象属性设置时间周期。过程组件的惯性同样应该考虑。建议标准对象使用 1~2 s 的周期
变量触发	2	如果此更新选项可以组态（根据动态类型），则应该优先使用它！如果变量是互相影响的，则始终要考虑负责更改属性或执行动作的所有变量。只有那些包含在列表中的变量才能作为更新动态属性或动作的触发器。变量触发设置为“一旦改变”只应有选择地使用。一旦所选变量中有一个发生变化，就会触发该属性或动作的触发器
画面周期	2	只有动态画面对象本身的动态属性在较短的时间间隔内发生变化并因此而必须进行更新时，才应该缩短此周期。延长画面周期会减少系统负载
窗口周期		如果正在处理一个打开的画面窗口，则此设置对调整过程变量（过程框）会起作用。如果为了获得信息（例如画面布局）而不断显示画面窗口，则应该将窗口及其内容的更新设置为变量触发或时间周期
直接读取		用于同步读取过程值的内部函数只应有选择地使用。应用这些函数需要由系统进行循环检测，因此会导致较大的通信负载

1. 时间周期 除了已描述的默认周期和相关的 250ms~1h 的时间设置（或用户定义的周期 1~5s）之外，还有其他时间触发：每小时（分和秒），每天（时、分、秒），每周（星期、时、分、秒），每月（日、时、分、秒），每年（月、日、时、分、秒），可以在图 1-1 中进行选择。

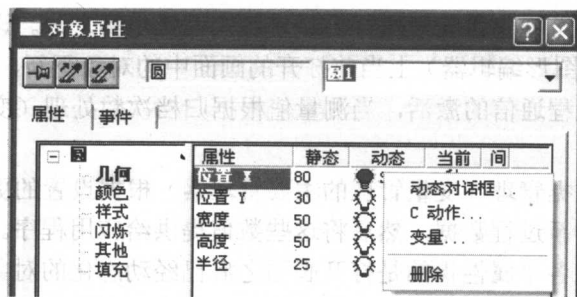


图1-1 属性动态化

2. 变量触发 如果动作用一个或多个变量来激活，则必须将事件触发器设置为变量触发。默认设置的周期为 2s。然而，组态人员可以设置时间因子来代替默认值。

1.3.3 在 WinCC 中添加动态

“添加动态”表示运行期间的状态（例如位置、颜色和文本等）变化以及对事件（例如鼠标单击、键盘操作和数值变化等）的响应。将图形窗口中的每个元素看作独立的对象。图形窗口本身同样是一种称为画面对象的对象。WinCC 图形系统中的每个对象都具有属性和事件。除少数情况外，大部分属性和事件都能够动态化。少数例外情况主要是指运行期间不受影响的属性和事件，它们没有显示其可以动态化的符号。

1. 使属性动态化 对象的属性（位置、颜色和文本等）可以静态地设置，并且可以在运行期间动态地改变。动态列中所有带有灯泡的属性都可以动态化。一旦属性被动态化，动态类型的彩色符号就代替白灯泡显示。已经动态化的主题（例如几何结构）以粗体字显示。在“对象属性”对话框中可使属性动态化，如图 1-1 所示。

2. 使事件动态化 对象的事件（例如鼠标单击、键盘操作和数值变化等）可以在运行期间检索，并且可以动态地进行判断。动作列中所有带闪电符号的事件都可以动态化。一旦事件被动态化，动态类型的彩色闪电就代替白色闪电显示。已经动态化的主题（例如“其他”）以粗体字显示。在“对象属性”对话框中可使事件动态化，如图 1-2 所示。

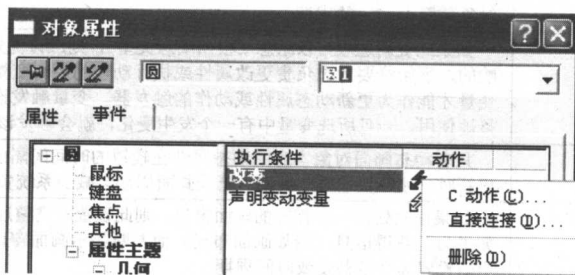


图1-2 事件动态化

3. 对象的动态化类型 设备画面的对象可以用许多不同的方法来动态化, 执行动态化的独立标准对话框面向不同的目标区域, 并且在某种程度上会导致不同的结果, 见表 1-2。

表 1-2 动态化类型的优缺点比较

动态类型	A ^①	B ^②	优 点	缺 点
动态向导	×	×	组态时以标准的方法提示	只适用于某些动态化的类型, 始终生成一个 C 动作
直接连接		×	使画面动态化的最快方法, 在运行系统中性能最佳	只限于一个连接, 并且只能在画面中使用
变量连接	×		易于组态	对动态化有所限制
动态对话框	×		快速且清楚: 用于数值范围或许多的选择项; 在运行系统中性能较高	并不适用于所有动态化类型
C 动作	×	×	由于脚本语言 (ANSI-C) 非常强大, 所以对动态化几乎没有限制	错误的 C 指令可能会导致产生错误, 与其他动态化类型相比性能较低

① A 为对象属性的动态化。

② B 为对象事件的动态化。

表 1-2 比较了动态化类型的优缺点, 下述为打开各种动态化对话框的方法:

1) “组态对话框”: 并不是所有对象都有这样的对话框来自动创建这些对象。在画面中选择对象, 单击鼠标右键打开其弹出式菜单, 在菜单中选择 “组态对话框”。

2) “动态向导”: 在 “查看” 菜单中打开 “工具栏...”, 在 “动态向导” 前加上复选标记。

在画面中选择对象, 在 “动态向导” 窗口 (如图 1-3 所示) 中选择 “标准向导”, 在列出的可选项中选择 “增加一个动态向导” 来启动 “动态向导”。

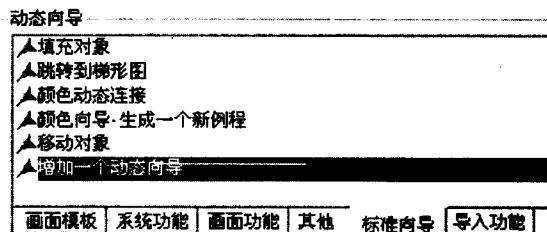


图1-3 动态向导窗口

3) “直接连接”: 在图 1-1 所示画面中选择对象, 单击鼠标右键打开其弹出式菜单, 在菜单中选择 “属性”, 显示 “对象属性” 对话框, 在 “对象属性” 对话框中选择 “事件” 标签, 选择 “动作” 列, 单击鼠标右键打开其弹出式菜单, 在打开的弹出式菜单中选择 “直接连接”。

4) “变量连接”: 在图 1-1 所示画面中选择对象, 单击鼠标右键打开其弹出式菜单, 在菜单中选择 “属性”, 显示 “对象属性” 对话框, 在 “对象属性” 对话框中选择 “属性” 标签, 选择 “动态” 列, 单击鼠标右键打开其弹出式菜单, 在打开的弹出式菜单中选择 “变量”, 在随后的对话框中, 按照提示选择并应用相应的变量。

5) “动态对话框”: 在图 1-1 所示画面中选择对象, 单击鼠标右键打开其弹出式菜单, 在菜单中选择 “属性”, 显示 “对象属性” 对话框, 在 “对象属性” 对话框中选择 “属性” 标签, 选择 “动态” 列, 单击鼠标右键打开其弹出式菜单, 在打开的弹出式菜单中选择 “动态对话框”, 在随后的对话框中, 按照提示选择并应用相应的动态。

6) “C 动作”: 在图 1-1 所示画面中选择对象, 单击鼠标右键打开其弹出式菜单, 在菜单中选择“属性”, 显示“对象属性”对话框, 在“对象属性”对话框中选择“属性”标签, 选择“动态”列, 单击鼠标右键打开其弹出式菜单, 在打开的弹出式菜单中选择“C 动作”, 在随后的对话框中, 按照提示选择并应用相应的“C 动作”。

上述各种动态化对话框的结果和表现形式如表 1-3 所示。

表 1-3 动态化对话框的结果和表现形式

对话框	结果	表现形式(符号)
动态向导	总是生成一个 C 动作	绿色闪电
直接连接		蓝色闪电
变量连接		绿色灯泡
动态对话框	自动生成 C 动作(InProc), 此 C 动作随后可以扩展, 但是会在过程中丢失性能优势	红色闪电, C 动作一旦改变, 就切换到绿色闪电
C 动作	已组态的 C 脚本	绿色闪电, 黄色闪电表示动作仍然要编译

4. 符号和位图的传送 状态显示的符号或画面文件中的图形对象可以作为项目画面文件夹中的独立文件存储。可通过将期望的符号文件(*.emf 或*.gif)复制到新建项目的目标文件夹\GraCS 来完成。这些画面立即存在于状态显示或图形对象的选择列表中。在导入时, 可以将符号集成到画面中, 或通过“插入”菜单中的“导入”子菜单将符号直接复制到正在编辑的图形画面中。对于后面这种情况, 不一定要复制文件, 可以通过访问源项目的路径(\GraCS)直接导入想要的符号, 然后选择所期望的符号文件。

5. 传送项目库(带有预组态符号和自定义对象) 如果符号存储在项目库中, 则通过将文件 library.pxl 复制到 library 路径下就可以将这个库用于另一个项目。

如果仅在新建项目中使用一些项目中指定的符号, 则可以单独导出它们(符号文件*.emf)。传送符号的过程和步骤如下:

- 1) 打开库。
- 2) 使用鼠标选择期望的符号并按住鼠标键将符号拖到画面中(拖放)。
- 3) 选择“文件”菜单中的“导出...”子菜单, 打开用来保存符号的对话框。
- 4) 保存符号。

6. 新建项目库 这些导出的符号现在可以作为独立的符号文件使用并可以通过导入来单独使用。如果这些符号在项目中频繁使用, 则可以一次集成到新建的项目库中。可以通过调用符号库, 特别是项目库来进行。

在创建自己的符号文件时(例如借助库窗口工具栏的文件夹图标通过拖放将已导入的符号复制到该文件夹中), 可用这种方法来传送项目的部分符号并添加其他指定的符号, 以再次创建新的项目指定的库。

7. 动作传送 将项目中的动作或将要在不同项目动作间复制的动作存储为独立的文件, 这些文件存储在\GraCS 文件夹下, 其扩展名为.act(表示动作)。任何时候都可以通过将他们从源文件夹复制到目标文件夹来进行传送。

从 C 动作编辑器中通过工具栏按钮导出动作, 将动作文件存储到由用户命名的目标文件中。通过导入动作工具栏按钮, 将所存储的动作文件传送到新建项目画面中的对象动作

中。

8. 变量的传送 在将数据传送到目标项目前,需要说明目标项目的地址。如果在 WinCC 的变量管理器中已经有大量变量存在,则应该将 WinCC 变量列表导入到目标项目。内部变量必须始终从 WinCC 的变量管理器传送,可以通过 Var_Exim.exe 工具来完成。

(1) 使用动态向导传送 S7 数据变量 借助动态向导,可将使用 STEP 7 软件产生的数据区定义读入 WinCC 的变量管理器。其具体步骤如下:

1) 备份项目数据,并在数据库中作相应改变。

2) 使用 STEP 软件导出分配列表。创建名为 prj_zuli.SEQ 的文件。

3) 删除所有不要求从该导出文件导入 WinCC 的特殊符号(例如对于程序调用)。可以使用典型的文本编辑器,例如写字板来完成。分配列表不能包含任何空白行。

4) 打开 WinCC 资源管理器中的目标项目。项目必须处于组态模式下(运行系统没有激活)。

5) 打开图形编辑器。在任意画面中,调到动态向导,并选择导入功能标签。从此处,选择导入 S7 分配列表的功能。然后,必须指定(使用按钮)源文件(.seq)及其路径,还需要指定逻辑连接,在其中将放置对分配列表的变量描述,将数据输入 WinCC 变量管理器中。用于 WinCC 项目的变量名称必须是惟一的。将变量添加到已存在的 WinCC 变量管理器。

(2) 使用帮助程序传送变量 可以在任何时候将定义在变量管理器中的变量作为文本文件导出,以补充变量列表。然后,必须将生成的数据导回项目的变量管理器。已创建的文件使用 CSV(逗号分隔的数值)格式,可使用任何编辑程序读取并做进一步的处理。

导出或导入数据的步骤如下:

1) 打开 WinCC 资源管理器中的 WinCC 项目。

2) 定义当前不可用但稍后需要用于导入的连接(通道 DLL-逻辑连接-连接参数)。它只能在新建项目中进行。

3) 激活 Var_exim 程序。

1.3.4 画面模块技术

画面模块技术对于允许组态画面组建快速而简单的组态及其重复使用性和维护性至关重要。例如,已组态的过程框可用于若干同类的过程组件(例如阀或控制器)。将在项目中一起工作,并且可视化的控制模块重新使用到已组态的原始画面窗口中,可按照下列原则进行操作:

1) 复制画面窗口并且重新连接变量域。

2) 使用在调用时分配其变量域的画面窗口(间接连接)。

3) 应用具有原型的自定义对象和结果对象。

4) 创建原始画面并且进行集成。

5) 创建 OCX 画面模块并将它们集成起来作为 WinCC COCX 对象。

1. 不同技术的比较 这些技术存在很大的区别,不同技术的优缺点比较参见表 1-4。

如果项目中仅使用少量的简单画面模块,选择其中一种类型就能实现。用户对象特别适合于中低复杂性的简单对象和变量连接。如果能够预见对象需要进行一些改动,则原始画面