

文章编号: 1000-1220(2001)11-1294-05

面向 Agent 生命周期法的 LonWorks 现场总线控制系统产品 CS2000 设计与实现

金敏 沈德耀 周翔

(中南大学 信息科学与工程学院 湖南长沙 410083)

摘要: 针对现有 LonWorks 产品只提供 Neuron C 指令语言编程环境的缺陷, 设计开发研制成功一套提供可视化图形控制组态软件平台的 LonWorks 现场总线控制系统产品 CS2000。文中给出了该系统产品研制过程中采用的面向 Agent 生命周期法的技术路线, 说明了系统分析与设计的实施步骤, 并具体阐述了硬件节点和控制组态软件平台的关键实现技术。该系统产品目前已成功应用于楼宇自动化和工业自动化的工程项目中, 具有产业化价值。

关键词: LonWorks 现场总线控制系统; 面向 Agent; 控制组态软件; 网络控制节点

中图分类号: TP273; TP18

文献标识码: A

1 引言

我国民族工控产品如 PLC、DCS 长期落后于国外, 根本原因是国外少数几家大公司为保持其垄断经营的地位对其工控产品采用封闭式结构, 我国对其关键核心技术只有通过引进消化后方可掌握, 因此贻误了市场时机而总是步人后尘、受制于人。现场总线技术的开放性策略无疑为我国工控界在国际市场上的发展带来了一个千载难逢的平等竞争机遇, 使大家同时都能参与到 LonWorks 技术消化吸收和应用开发行列。在众多现场总线标准中, 美国 Echelon 公司推出的 LonWorks 现场总线^[1]以其在楼宇自动化和工业自动化等领域的突出业绩而倍受瞩目, 包括 Honeywell、IBM、AT&T 等工业巨头在内的 2500 家生产商竞相推出了自己的 LonWorks 控制模块和软件产品。国外目前大多数 LonWorks 产品开发商仍停留在集成 Neuron C 编程环境阶段, 欲用 Neuron C 语言开发出高效的控制应用程序, 不仅需要熟悉这种高级指令语言本身, 而且必须对 Neuron 芯片的内部硬件结构、存储器的地址空间分配等各个技术细节有深入详细的了解和掌握, 这些对现场工程技术人员来说显然是不现实的要求。也有少数几家 LonWorks 产品开发商已阔步跨进了可视化图形编程语言先进行列, 如美国 Dayton General System 公司已推出图形化编程语言。遗憾的是, 在实际工程应用中, 这些可视化的图形编程语言暴露出控制功能模块不够专业化、组态参数的非可视化和界面不太友好等缺陷。鉴于此, 我们紧紧抓住现场总线技术给中国工控界发展带来的这一历史良机, 及时消化掌握了 LonWorks 的全套技术, 并针对 LonWorks 只提供 Neuron C 指令语言编程环境的缺陷, 设计开发研制成功包括通用型网络智能节点和可视化图形控制组态软件平台在内的 LonWorks 现场总线控制系统 CS2000, 该系统产品的特色在于提供一可视图形化的控制组态软件编程环境, 它将控制程序变为工程技术人员所构思的控制策略的图形化映象, 而图

元间的连接关系则表示控制算法的数据流向, 且内置多个控制领域典型控制算法和高级控制算法, 直观可视化地表示出各功能块组态参数的物理意义以引导用户进行正确的参数设置, 彻底实现了可视化图形编程的思想, 且用户界面友好, 有经验的现场控制工程技术人员无须专门培训即可短期内完成控制策略的编制, 方便高效。

在 CS2000 系统的研制开发过程中, 我们依据面向 Agent 生命周期法的软硬件工程一般方法论^[2]为指导思想, 以系统不确定性和开放性为出发点, 以软硬件 Agent 一般性模型为统一描述范式, 通过采用自顶向下的分析设计与自底向上的实现交汇融合的开放技术路线, 从而成功的在半年时间内高效完成了这套系统产品的研制开发任务。

2 系统分析

2.1 需求分析

通过系统需求分析, 得到以下需求描述:

- (1) 系统应用领域为楼宇自动化和工业自动化等;
- (2) 系统由若干分散在现场的控制器组成, 各控制器在最大程度上完成各自控制目标的前提下, 协作完成系统总体目标;
- (3) 各控制器对复杂工业现场各地理位置分布的被控对象具有独立的闭环控制功能和监视管理功能;
- (4) 各控制器具有自校验与自诊断功能, 在出现故障情况下能自行关断, 不影响整个控制系统的继续运行, 保证系统的可靠性;
- (5) 任意两个控制器任意时刻无需第三方参与即可进行点对点的相互通信, 通信需达到实时控制系统所需求的实时性、安全性、可靠性和负载优化;
- (6) 控制策略的编制易实现性和运行高效性;
- (7) 低成本。

收稿日期: 2000-05-26 收修改稿日期: 2000-09-22 基金项目: “211 工程”重点实验室建设资助项目, 湖南省科技园入园项目。作者简介: 金敏, 博士研究生, 主要从事分布式 Multi-Agent 理论、LonWorks 现场总线控制系统、工业自动化及楼宇自动化等方面的研究。沈德耀, 教授, 博士生导师, 主要从事分布式人工智能、现场总线技术及过程自动化等方面的研究。周翔, 博士, 主要从事人工智能、Agent 理论等方面的研究。

2.2 可行性论证(略)

3 系统设计

3.1 粗粒度识别 Agent 实体

CS2000 系统中,从问题域中识别出的基本 Agent 实体类为通用型网络智能节点 Agent,另外为满足节点开发、网络安装监视维护等目标域而创建性的识别出的实体有五类,分别是节点开发组态 Agent、网络安装 Agent、SCADA Agent、现场网络安装维护 Agent 和路由器 Agent. 详细 Agent 字典见文献[2].

3.2 系统体系结构

CS2000 系统体系结构如 1 所示.

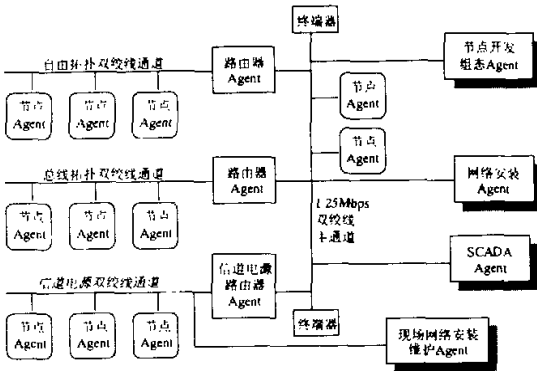


图 1 CS2000 系统体系结构

下面以网络智能控制节点 Agent 的节点硬件子 Agent 和节点开发组态 Agent 的可视化组态软件子 Agent 为例,说明单个硬件 Agent 和软件 Agent 的具体细粒度构建方法.

4 通用型网络智能节点硬件产品的研制开发

4.1 通用型网络智能控制节点硬件 Agent 实体细粒度构建

4.1.1 节点 Agent 形式化描述

根据软硬件 Agent 一般性形式化描述,硬件节点 Agent 可以形式化具体描述为以下七元组:

<节点 Agent> ::= (<Aid> <传感器> <通讯模块> <状态象> <事务处理器> <协调机构> <资源>)

其中:

- <Aid> ::= <通用型网络智能节点>
- <传感器> ::= (<模拟量检测通道> <开关量检测通道>)
- <通讯模块> ::= (<FTT-10A 收发器> <整流滤波电路> <静电放电保护电路> <LonTalk 通讯协议>)
- <状态象> ::= (<模拟量输出通道> <开关量输出通道> <节点地址设置单元>)
- <事务处理器> ::= <神经元芯片 Neuron 3150>
- <神经元芯片> ::= (<介质访问 CPU> <网络 CPU> <应用 CPU>)
- <协调机构> ::= (<firmware 固件> <神经元芯片应用 CPU>)
- <资源> ::= (<存储器> <知识库> <网络数据库> <实时时钟>)
- <存储器> ::= (<On-Chip RAM> <On-Chip EEPROM>

<Off-Chip NVRAM>

<Off-Chip EPROM>:

4.1.2 节点 Agent 结构模型(见图 2)

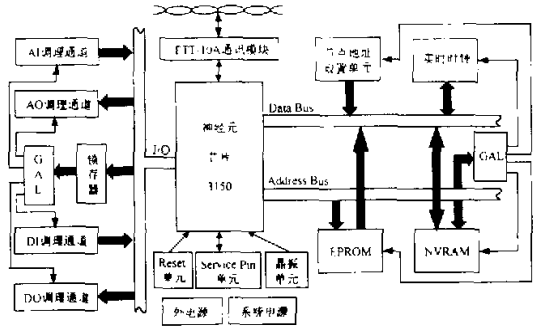


图 2 节点 Agent 结构模型

4.2 通用型网络智能节点硬件的实现

通用型网络智能节点硬件产品见图 3. 下面就该硬件产品研制过程中的一些关键技术要点予以阐述.

4.2.1 神经无芯片(Neuron Chip) 神经元芯片^[2]是本节点的控制和通讯中心枢纽,与一般微控制器相比,其显著的特点在于它所具有强大的网络通讯功能.

TMPN3150 神经元芯片的存储器包括片内和片外共 64K,其中片内 6K,片外 58K,系统 firmware 固件占据片外存储器的开始 16K 字节,用户实际能使用的片外存储器为 42K. 3150 允许的外接存储器类型有 EPROM、EEPROM、FLASH Memory (闪存存储器)和 NVRAM (非易失性 RAM),考虑到下装后的用户控制组态程序在节点掉电不致丢失,本节点的外接存储器采用 NVRAM,选用 Dallas 公司的 DS1230Y-120,它与 EPROM 共同构成本节点的片外存储器,该存储器映象在 NodeBuilder 节点开发工具中的 Memory Map 中定制.

TMPN3150 的通讯端口有 3 种工作模式,单端(Single-Ended Mode)、差分(Differential Mode)和专用模式(Special-purpose Mode);支持各种通信介质,包括双绞线、电力线、无线、红外、光纤等. 本节点采用廉价的双绞线通讯介质,工作模式为单端方式,详见 FTT-10A 通讯模块实现部分.

TMPN3150 的可编程 I/O 端口有 4 大类(直接、并行、串行、定时器/计数器)共 34 种预编程工作方式,且采用面向对象的编程方式,11 个 I/O 端口可配成 34 种不同的 I/O 对象. 本节点 D 神经元芯片 I/O 端口,采用并行 I/O 对象的 Muxbus 总线工作方式,作为模拟量/开关量输入输出的总线通道,定义方法为:

//输入输出通道(I/O口)定义
IO-0 muxbus IO-bus;

4.2.2 FTT-10A 通讯模块 FTT-10A 为自由拓扑双绞线收发器^[3],包含一隔离变压器和一曼彻斯特编码器,采用厚膜电路集成在一个芯片中. 该收发器在传输速率、通讯距离、节点容量和电气隔离性能等方面较 RS-485 均有明显的优势.

本节点采用 FTT-10A 作为收发器,与整流滤波和静电放电保护外围电路一起构成通讯模块,神经元芯片 TMPN3150 的通讯端口 CP0~CP1 配合工作在单端方式,完成节点的网络通讯功能.FTT-10A 与 TMPN3150 的连接框图见图 3.

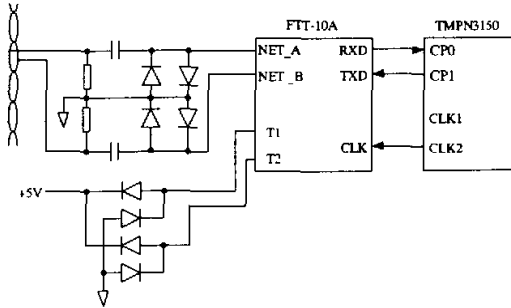


图 3 FTT-10A 通讯模块

4.2.3 本安性(Intrinsic Safety) 现场总线控制系统中的控制节点之所以能彻底分散到现场,前提是节点必须具备本安性.本安是一种危险区域的防爆技术^[3],与其它防爆技术相比,本安技术具有无需密封和磁耦合控制、维护方便,体积小、更高的可靠性和更广的应用场合等优点.

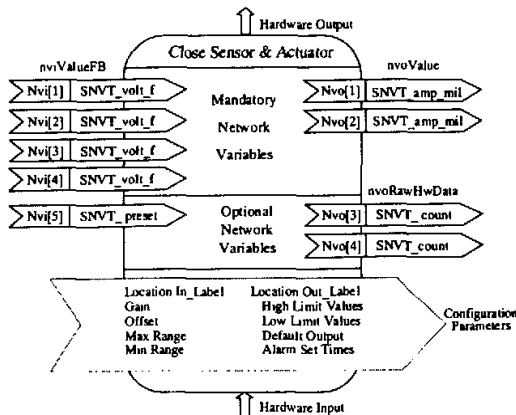


图 4 通用型网络智能节点 LonMark 对象

本安技术实现的机制是,允许可燃性介质进入电气设备时,通过认证的齐纳安全栅或电气隔离器来限制危险区域内进入电路设备的能量(电流、电压或功率),主要考虑两种点燃爆炸机制,即电火花和热表面.电路最大允许的电流(电压、功率)按下式计算:

$$\frac{\text{电路最大允许电流(电压或功率)}}{\frac{\text{最小点燃电流(电压或功率)}}{\text{安全系数}}} \quad (4.1)$$

IEC 标准定义了两个本安等级 ia 和 ib,规定在保证设备安全的前提下允许出现的故障数和安全系数.

典型本安节点的经验数据为:节点工作频率 5MHz,通讯速率 78Kbps,通讯距离 1000m(总线)/300m(自由拓扑),节点容量 2~3 时本安工作电流 25~40mA.

无论节点是否采用总线供电,在危险区域均有本安问题.CS2000 系统中的节点采用的是非总线供电形式,因此该类节点采用本安电源(I_{max} = 40mA)供电,且节点在本安电源提供的功率范围内正常工作即可;对于总线供电形式的节点,英国 MTL(测量技术有限公司)已开发出通过认证的 LonWorks 总线供电的本质安全通道系列产品 IS-78,包括同时起安全栅和中继作用的 MTL3054(输出功率率较大)及 MTL3055(输出功率率较小).IST-78(本安 LonTalk 收发器)和 ISC-78(本安神经元控制模块).

4.2.4 可互操作性 可互操作性使得不同 OEM 开发商的 LonWorks 节点产品能够集成在一个 LonWorks 网络系统中协同工作,其实现机制是通过应用层采用 LonMark 对象和表示层采用标准网络变量(SNVT)接口实现. LonMark 对象有五类,分别是 Node、Open Sensor、Close Sensor、Open Actuator、Close Actuator. CS2000 系统的通用型网络智能节点为实现可互操作性定义的 LonMark 对象属于 Close Sensor & Actuator 混合型,其功能文档框图见图 4.

5 可视化图形控制组态软件平台 Visual Control Configuration(VCC)的研制开发

5.1 VCC 控制组态软件平台 Agent 实体的构建

5.1.1 组态平台 Agent 形式描述 根据软硬件 Agent 一般性形式化描述,组态平台 Agent 可以具体形式化描述为以下七元组:

- <组态平台 Agent> ::= {<Aid> <传感器> <通讯模块> <状态集> <事务处理器> <协调机构> <资源>} 其中, <Aid> ::= <可视化图形控制组态软件平台 Visual Control Configuration>
- <传感器> ::= (<模拟量输入通道功能块> <开关量输入通道功能块>)
- <通讯模块> ::= (<网络变量通讯功能块> <显示报文通讯功能块>)
- <状态集> ::= (<模拟量输出通道功能块> <开关量输出通道功能块> <算术运算功能块输出> <逻辑运算功能块输出>)
- <控制运算功能块输出>)
- <事务处理器> ::= {<运算事物处理器> <通讯事物处理器>}
- <运算事物处理器> ::= {<算术运算事物处理器> <逻辑运算事物处理器> <控制运算事物处理器>}
- <通讯事物处理器> ::= (<网络变量处理器> <显示报文处理器>)
- <协调机构> ::= (<事物调度功能块> <时序调度功能块> <冲突消解功能块>)
- <资源> ::= (<存储器资源> <函数库> <实时时钟功能块>)

5.1.2 组态平台 Agent 结构模型 采用分布式 Client/Server

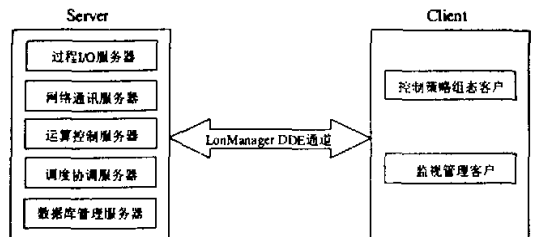


图 5 组态平台 Agent 的 C/S 结构模型

er 客户/服务器结构模式,如图5所示。其中:服务器位于组态软件的后台,提供模拟量/开关量等过程信息的输入/输出服务、网络变量及显示报文的通讯服务、运算控制服务、调度协调服务和数据库管理服务等;客户端位于组态软件的前台,为用户提供一个可视化的、图形化的控制策略编制开发环境和监视环境,属于MMI范畴;客户端与服务器之间的通讯通道采用LonManager DDE(Dynamic Data Exchange)。

5.2 可视化图形控制组态软件平台VCC的实现

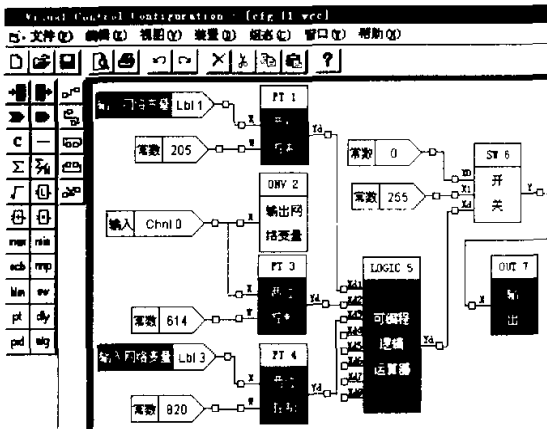


图6 控制组态软件平台界面

Visual Control Configuration 控制组态软件平台的系统功能界面如图6所示,图中Cfg-11.vcc为某用户在该平台上编制的控制组态程序。VCC组态软件平台的服务器端采用Neuron C语言(一种面向对象的扩展ANSI C语言)编制,客户端采用Visual Basic5.0开发。下面就该软件平台研制过程中的一些关键技术要点予以说明。

5.2.1 控制策略的可视编程环境 目前国内外大多数LonWorks产品开发商提供给用户的节点控制策略开发环境仍停留在Neuron C语言这种指令语言的不可视编程环境,欲用Neuron C语言开发出高效的控制应用程序,用户不仅需要熟悉这种指令语言本身,而且必须对神经元芯片的内部硬件结构、存储器的地址空间分配等各个技术环节有深入详细的了解和掌握,这些对于现场工程技术人员来说显然是不现实的要求。国际电工委员会的IEC1131-3(可编程控制器编程语言标准)中的功能方块图FBD(Function Block Diagram)编程标准为解决上述问题指明了一条新的途径。

可视编程环境即是对功能方块图这一编程理念的继承和发扬,通过将各功能方块图进一步变为可视化图元,可视编程环境将工程技术人员所构思控制算法和控制策略映射为各种图元及其组合,图元间的连接关系则表示控制算法的数据流向。这是一种基于数据流机制的环境模型,具体而言,它首先将一个控制任务划分成相互独立的具有控制计算能力的可视功能块,每个功能块视为带有输入/输出接口和组态参数设置

接口的封装体,每个功能块在接收到数据或事件后开始运行,然后把结果传输给与之相连的下一功能块,各功能块之间依靠数据传递保持联系。因而,在可视编程环境下,编制控制程序用户的精力不必再分散于“怎么做”,而可集中于“做什么”,具体“怎么做”交给编程环境实现,这是一种面向控制目标的编程环境,而不是传统的面向控制过程的编程环境。可视编程环境下表示控制任务实现方式的数据依赖关系图称为数据流图(Flow Graph),它包括以下要素:

- ① 可视功能块:为具有独立运算控制功能的实体,是构造数据流图的基本单元,每个功能块具有相应的数据输入/输出接口和(或)组态参数设置接口。Visual Control Configuration提供三类功能块,即过程I/O功能块,运算功能块和控制功能块。
- ② 管道线:它将一功能块的输出端与另一功能块输入端相连,负责把上一级功能块实体处理后的数据传递到下一级功能块实体处理,实现功能块实体间数据的有效流动。
- ③ 驱动方式:有数据驱动和事件驱动两种方式:在数据驱动方式下,当功能块的输入数据到达或变化时,功能块就被点火执行,其结果沿功能块输出端向下传递,形成系列点火执行顺序;事件驱动方式包括控制周期到达事件、客户请求事件等。
- ④ 时间调度机制:负责调度每个功能块实体执行的先后顺序。

VCC可视编程环境的运作方式为:在客户端前台软件中提供可视图元(包括功能块和管道线)界面组态环境,用户在客户端进行;控制组态程序编制完成后,客户端软件通过LonManager DDE通道将其分别下装至各网络控制器节点;在时序调度机制和协同机制的控制下,组态程序向后台服务器依次请求相关过程I/O服务、运算控制服务、网络通讯服务等实现系统控制目标。

5.2.2 “软集成芯片”—COM 部件编程技术 软件的可重用性、可扩展维护性是软件质量高低的两个重要衡量尺度。传统的软件程序通常被分割成若干个文件、模块、函数或类,然后被编译链接成一个铁板块式的二进制文件,这种软件由于模块之间的关联耦合性强,因此扩展维护性能很差,并且其模块、函数或封装类的重用性低,因为这类重用属于源代码级,源代码往往依赖于原操作系统、编程语言和应用项目,无法做到分布式跨平台、跨语言的重用,且重用前往往需要根据当前具体应用项目进行整理和修改,可能导致产生不可预见的错误或副作用。COM 部件编程技术的诞生很好的解决了软件的重用性和扩展维护性问题。部件COM(Component Object Module)是一段独立的二进制代码,通过使用标准接口,部件可实现跨越时间和地域限制的分布式的应用,不受系统支撑平台和编程语言的限制,因此极大的提高了软件的可重用性;部件可被动态插入或卸出应用程序,具有即插即用功能,并且通过一定的注册机制就可以解决部件升级时的版本控制问题,因而提高了软件的可扩展性和可维护性。软件领域“部件”的出现,与当年硬件领域“集成芯片”的出现如出一辙,作为一种“软集成芯片”,部件很好的解决了软件设计开发过程中的

