

基于 OPC UA 的工业设备数据采集系统

禹鑫燧 殷慧武 施甜峰 唐权瑞 柏继华 欧林林

浙江工业大学信息工程学院 杭州 310023

(yuxinyinet@163.com)

摘要 为了解决工业设备协议多样性给工厂数据采集和统一监控带来的难题,研究了基于 OPC UA 的数据采集系统。以 PLC、工业机器人和数控机床等工业设备为研究对象,通过工业以太网建立与本地监控服务器的连接。在本地监控服务器根据不同的工业设备通信协议设计不同的数据采集驱动和数据转换插件,并统一管理。基于 OPC UASDK 和配置界面生成的 XML 文件构建 OPC UA 地址空间,从而建设设备的 OPC UA 服务器。OPC UA 服务器用以存储采集并转换后的实时数据和 OPC UA 客户端交互,同时将转换后的数据上传到云端存储系统供进一步数据分析。系统基于 .Net 平台开发,采用 C# 语言和 .net framework 框架构建整个本地监控服务器端,并且采用 WPF 设计本地监控服务器端的监控和配置界面。云端结合 Redis 和 MySQL 实现对运行数据的存储。最后通过实验验证了所提系统的可行性和完整性。

关键词: 工业设备;OPC UA;Redis;数据采集;. Net

中图法分类号 TP242.6

Data Acquisition System of Industrial Equipment Based on OPC UA

YU Xin-yi, YIN Hui-wu, SHI Tian-feng, TANG Quan-rui, BAI Ji-hua and OU Lin-lin

College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China

Abstract In order to solve the problem of data collection and unified monitoring caused by various industrial equipment protocols, a data collection system based on OPC UA is studied. Taking industrial equipments, such as PLC, industrial robot and CNC machine tool as research object. Connection between industrial equipment and local monitoring server is established through industrial Ethernet. On the local monitoring server, different data acquisition drivers and data conversion plug-ins are designed and unified management according to different industrial equipment communication protocols. Based on the OPC UASDK and the XML file generated by the configuration interface, the OPC UA address space is constructed to build the OPC UA server, which is used to store the converted data and interact with the OPC UA client. Meanwhile, the collected converted data is uploaded to the cloud storage system for further data analysis. The system is developed based on .net platform, using C# language and .net framework to build the whole local monitoring server, using WPF to design the monitoring and configuration interface of the local monitoring server. The cloud combines Redis and MySQL to realize the storage of operating data. Finally, the feasibility and integrity of the system are verified through experiments.

Keywords Industrial equipment, OPC UA, Redis, Data collection, . Net

1 引言

信息技术的高速发展不断推动互联网与工业的融合,促使智能制造业进入了新的阶段,我国于 2015 年正式出台了“中国制造 2025”战略,旨在大力发展制造业实现制造业强国,其中重要一步就是大量使用自动化工业设备以取代人力劳动^[1],实现工厂智能化、无人化。但是工业设备品牌多样,各设备厂家独立采用不同的通信协议和规范,这使得工厂获取设备数据的成本大大提高。为了解决多设备协议带来的设备兼容性以及成本问题,需要统一设备协议,OPC UA(OLE for Process Control Unified Architecture)规范^[2-3]由此诞生。

OPC UA 规范主要用于解决当前工业自动化领域存在的兼容性、开放性、通用性等难题。

OPC UA 由 OPC 发展而来,是对 OPC 的继承与升级,解决了传统 OPC 对 Windows 平台的依赖以及对分布式系统信息交换不适用等主要问题^[4]。该规范规定了不同软件产品间进行数据交换的标准通信接口,将 OPC UA 技术运用到数据采集系统^[5],解决了系统因为工业设备通讯协议多样带来上位机软件开发复杂多样的困难。

国内外对于 OPC UA 协议规范已经有了很多研究,国外西门子、ABB 等公司均已推出了支持 OPC UA 规范的产品,也有很多成熟的 OPC UA SDK,国内对于 OPC UA 研究较

基金项目:国家重点研究发展计划“智能机器人”重点项目(2018YFB1308000);机器人技术与系统国家重点实验室开放基金项目(SKLR2013MS06)

This work was supported by the “Intelligent Robot” Key Project of National Key Research and Development Plan (2018YFB1308000) and Open Fund Project of National Key Laboratory of Robotics and Systems (SKLR2013MS06).

通信作者:欧林林(linlinou@zjut.edu.cn)

晚,但已有较多应用。例如,文献[6]提出的基于 OPC UA 的 MES 数据管理系统将 OPC UA 技术应用到生产执行制造系统中,作为 MES 系统的底层数据采集技术核心,借此实现原始数据到上层管理的传输;文献[7]针对数控机床 828D,采用自带的 OPC UA 服务器实现底层设备与监控系统的数据交换,客户端通过直接读取服务器数据实现与数据库的数据同步;文献[8]以 OPC UA 技术的地址空间和信息模型为基础,重点分析了数据采集监控系统的需求和框架,基于 OPC UA SDK 实现了 SCADA 系统的关键模块设计。国内对于 OPC UA 的研究大多集中在基础的功能模块,应用方面则大多采用国外成熟的 SDK 进行开发,并且只是针对特定的工业设备,不具备通用性。

为了解决工厂设备和协议多样带来的数据采集和上位机开发难题,本文设计了基于 OPC UA 的数据采集系统,实现工业设备协议到 OPC UA 协议的转换,通过驱动开发和简单配置就可以实现设备的兼容。本文设计了系统的总体方案,该系统由 3 部分构成:本地工业设备网络、本地监控服务器和云端数据处理服务器。系统以本地监控服务器为核心,向下采集工业设备网络数据并进行 OPC UA 协议转换,建立 OPC UA 服务器和本地监控,向上结构化存储工业设备数据到云端服务器。首先根据功能需求设计本地监控服务器,基于 OPC UA SDK 搭建 OPC UA 服务器,采用接口的方式标准化数据采集驱动和数据解析插件,实现工业机器人协议到 OPC UA 的转换,同时使用 WPF 设计可视化本地监控和配置界面;然后基于 Redis 和 MySQL 建立云端数据库,便于进一步进行数据开发和基于机器学习算法进行故障预测性分析;最后基于 UR5 和 ABB120 设计数据采集驱动和数据解析插件,使用开源的 OPC UA 客户端和虚拟调试技术测试建立的本地监控服务器的完整性和可行性。

2 系统需求分析与总体方案设计

本文设计的基于 OPC UA 的数据采集系统,旨在为所有未提供 OPC UA 协议功能的设备建立通用的 OPC UA 转换机制,实现不同工业设备通信协议到 OPC UA 协议的转换。通过针对不同的工业设备设计标准的数据采集驱动,将工业设备采集的数据进一步分解加工为含有语义信息的 OPC UA 格式信息,从而映射建立 OPC UA 地址空间,为不具备 OPC UA 功能的工业设备建立 OPC UA 服务器,实现设备协议标准化,建立设备的互通信,降低工厂设备和协议多样带来的上位机统一监控难题。

系统的总体构成如图 1 所示,主要包括多种工业设备组成的工业设备网络、实现协议转换核心功能的本地监控服务器以及对采集数据存储并处理的云端数据处理服务器。

本地监控服务器通过以太网与工业设备网络连接,以工业设备网络为服务端,搭载本地监控服务器的 PC 机为客户端,构成 C/S(Client/Server)结构的本地数据采集和监控系统。采用多线程分别采集各工业设备数据,将采集的数据解析后映射到建立好的 OPC UA 地址空间,建立工业设备的 OPC UA 服务器。以本地监控服务器为服务端与远程监控客户端构成 C/S 结构的远程监控系统。采集转换后的数据上传至云端服务器,云端服务器对数据进行结构化存储以便进

一步分析处理,实现故障预测。为满足不同需求,云端服务器基于 Redis 和 Mysql 建立数据库架构存储设备数据,通过关系型数据库存储了设备的历史数据以及设备间的层级关系,提高历史数据的查找效率;采用 Redis 缓存实时数据,解决了实时数据访问的速度问题以及系统并发请求的效率问题。

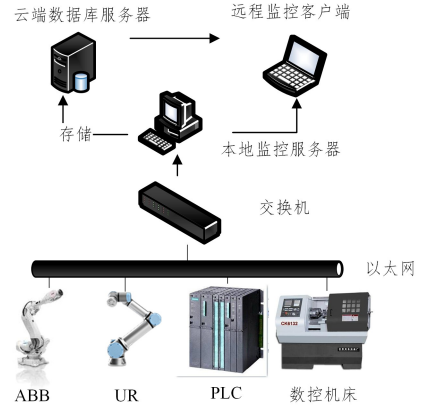


图 1 系统总体构成图

Fig. 1 Overall system composition diagram

本地监控服务器采用 WPF 设计监控界面,使用 MVVM (Model View View Model)框架实现界面和后台程序的分离,便于系统的拓展。通过任意具备 OPC UA 客户端功能的设备可以直接访问建立的 OPC UA 服务器从而达到远程监控。

3 本地监控服务器的设计

本地监控服务器为整个系统的开发核心,为简化其开发逻辑,根据本地监控服务器的各部分软件调用分层,将其设计为 5 个主要功能块,数据采集驱动开发、协议转换、OPC UA 服务器搭建、数据库存储、UI 交互,如图 2 所示,然后将各功能块进一步划分,细化系统的功能开发。

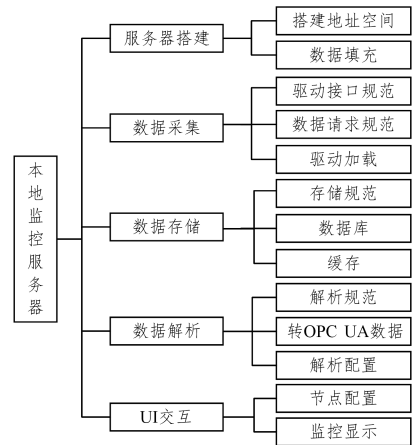


图 2 本地监控服务器功能图

Fig. 2 Function diagram of local monitoring server

如图 3 为本地监控服务器各主要功能模块连接图。以 UI 交互界面为起点,首先进行系统的节点配置和解析配置,分别生成两个 XML 文件保存配置信息,节点配置 XML 文件保存设备分类节点树,解析配置 XML 文件保存所有数据请求的解析规则,即请求数据的信息(数据类型、语义、长度、索引)。数据采集部分读取节点配置 XML 文件来获取设备连接信息和数据请求信息,调用相应的数据采集驱动实现对工业设备的数据采集;数据解析部分读取解析配置 XML 文件,

将采集的设备数据转换为具体的温度、电压、电流、关节等直观可视数据。对每个设备都采用独立的线程进行数据采集和数据解析。基于 OPC UA SDK 建立的 OPC UA 服务器加载节点配置和解析配置 XML 文件,将工厂树节点模型映射为自己的地址空间,建立 OPC UA 设备节点和值节点。解析后的数据填入建立好的 OPC UA 地址空间,完成 OPC UA 数据采集服务器的搭建,同时解析后的数据通过存储模块上传到云端服务器实现持久化,便于进一步数据处理,此外解析后的数据通过 MVVM 框架不断更新本地端 UI 界面显示,实现本地对数据的实时监控。

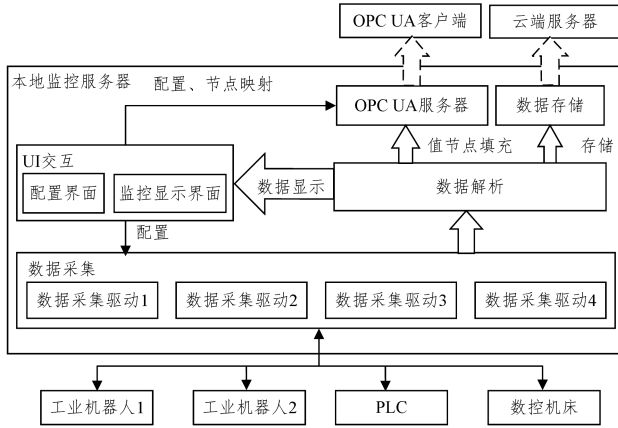


图3 本地监控服务器的功能连接

Fig. 3 Function-connection of local monitoring server

3.1 OPC UA 服务器的构建

OPC UA 服务器的构建主要有两种方式,即根据 OPC UA 目前的 13 种规范^[10-12] 直接开发或者使用成熟的 SDK 间接开发。前者可以根据实际需要选择性地实现相应功能,避免程序的冗余,但是需要深入理解 OPC UA 服务器的实现原理,比较耗时;后者基于已有的 SDK 开发,SDK 包含了 OPC UA 服务器所需要的全部方法,不需要开发人员对 OPC UA 有较深的理解,可以实现快速搭建。本文的 OPC UA 服务器就是基于 OPC UA 官方 SDK 开发,建立 OPCUAServer 类,在该类构造函数 OpcUaServer(string url) 中通过 SDK 给出的 API 接口对 OPC UA 服务器进行初始化并创建地址空间^[13-14],在服务器启动时,通过建立 OPCUAServer 对象开启服务器。其搭建流程如图 4 所示。

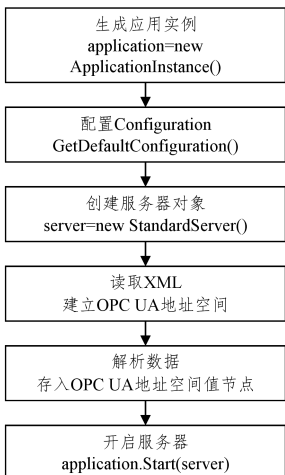


图4 OPC UA 服务器搭建流程

Fig. 4 OPC UA server setup process

服务器初始化主要使用 SDK 中的接口 ApplicationInstance, ApplicationConfiguration 和 StandardServer。首先通过 ApplicationInstance 建立应用实例 application;接着对应用实例进行基础配置,即对 ApplicationConfiguration 进行初始化赋值,利用生成的 xml 节点文件和 StandardServer 生成服务器的地址空间,从而建立服务器对象 server;最后利用应用实例开启服务器对象运行服务器 application.Start(server)。

通过配置界面生成的 XML 文件结合 SDK 接口实现地址空间的建立是服务器构建的关键,在设备配置和解析配置界面建立的节点模型会存储为 XML 文件,通过加载该 XML 节点模型文件,遍历每个节点,使用 SDK 提供的接口将节点模型映射为 OPC UA 节点,并对每一个节点建立对应的 NodeID 和 Description,从而完成地址空间的建立。通过数据采集解析后的数据根据 OPC UA 节点名称一一存入 OPC UA 地址空间的值节点中构成完整的 OPC UA 服务器,OPC UA 客户端可以通过浏览服务器端的地址空间直接查看对应数据。

3.2 数据采集与解析模块的设计

数据采集与解析模块采用 C# 语言,基于 Visual Studio 和 .netFramework 框架设计构建。首先对整体程序需求进行分析,数据采集分为设备连接和数据请求。设备连接即通过获取配置的设备 IP 地址和端口号等信息初始化设备数据采集驱动,建立与设备的 Socket 连接;数据请求则需要知道一次数据读取的字节数或者寄存器数目以及读取数据的起始地址等关键信息。解析模块则是将每次数据请求获取的原始字节码数据流分割为独立的数据,再将每个独立数据进行数据类型转换,转为标准的可直观理解数据,因此需要数据流中数据的数目以及每个数据的数据类型。

由于不同的工业设备所使用的通讯协议以及官方提供的 SDK 各不相同,并且设备中存储数据的编码格式略有差别,需要对工业设备分别设计数据采集和解析程序。为了对众多的驱动程序实现统一管理和调用,采用接口的方式规范驱动的调用。根据需求分别建立设备接口和解析接口,设备接口主要实现设备连接和数据读取方法,解析接口则实现数据类型转换方法,不同的设备独立实现接口方法。

以 UR5 工业机器人为例,采用 Modbus 协议与其通信,基于 Modbus 协议设计 UR5 的数据采集和数据解析模块。继承设备接口实现 UR5 设备的数据采集驱动程序 URDeviceBase,在 URDeviceBase 中实现接口 Connect, Read, Disconnect 和 StartEngine 等方法。Connect 方法通过 Socket 与 UR5 建立连接,连接所需要的 IP 地址和端口号等属性信息通过加载配置界面生成的 XML 文件获取。Read 方法根据参数“起始地址”和“字节长度”打包数据读取的报文,通过建立的 Socket 通道发送到 UR5 工业机器人控制器,返回字节数组数据,Disconnect 则断开 Socket 连接通道,同时结束整个线程释放所有资源。StartEngine 方法首先建立新的线程,接着调用 Connect 方法建立连接,连接成功后不断执行 Read 方法,并且将返回的字节数组除去报文头存入属性 Data 字节数组中。通过 Modbus 协议获取的每个 UR5 数据都占用两个字节,而标准的数据类型中 Int 占用 4 字节,Float 占用 4 字节,Double 占用 8 字节,因此在数据类型转换方法中以 0 补满字节位,从而建立 UR5 的 Modbus 协议数据转换规则,

使用对应的数据转换规则 Transform 将 Data 中的数据按照配置的解析规则分解转换为具体的电压、电流、关节等数据,最终转换的数据以键值的形式存入设备的数据字典 DictionaryData 中。

3.3 UI 交互界面的设计

UI 交互界面的主要功能是配置和监控显示,可实现系统通用功能、方便使用者操作该系统。本文基于 WPF 框架设计了 UI 交互界面,WPF 是微软推出的基于 Windows 的用户界面框架,真正分离了界面设计与开发,分别利用 Xmal 语言和 C# 语言设计纯图形界面和建立后台事件,采用 MVVM 模式最大限度降低界面 Xmal 文件与后台 CS 文件的耦合度,使得界面的大幅修改不会对逻辑代码产生较大影响。

UI 交互界面核心是配置功能,即节点配置、解析配置和 OPC UA 服务器配置,并且根据界面配置生成 Xml 文件。图 5 为设备节点配置界面,图 5(a)为节点配置界面,其中左侧框通过 TreeView 控件根据实际工厂建立树节点模型,即添加分类节点和设备节点,添加的每一个节点都要配置基本信息,右侧框显示选中的节点配置信息。图 5(b)对普通分类节点配置基本描述 Description 和名称 Name。图 5(c)对设备节点进行参数配置,包括连接工业机器人设备基本的端口号 Port、IP 地址 IpAddress 和超时时间 Timeout,以及通用的节点描述和节点名称。通过以上 3 个界面实现基本的树节点模型建立和节点信息配置,其中设备节点的配置用来建立系统和设备的 Socket 连接通道,而后台实现一次数据采集操作,发送的报文信息要包括基本的寄存器起始地址、采集数据长度(字节数或寄存器数,一个寄存器包含两字节数据)和数据请求超时时间。图 5(d)界面对设备添加数据请求并且配置数据请求的起始地址和字节长度,其中解析规则即建立的数据解析配置,每个数据请求都要提前建立其数据解析,即对采集的数据段添加描述,以便将其划分为具体的单个数据。

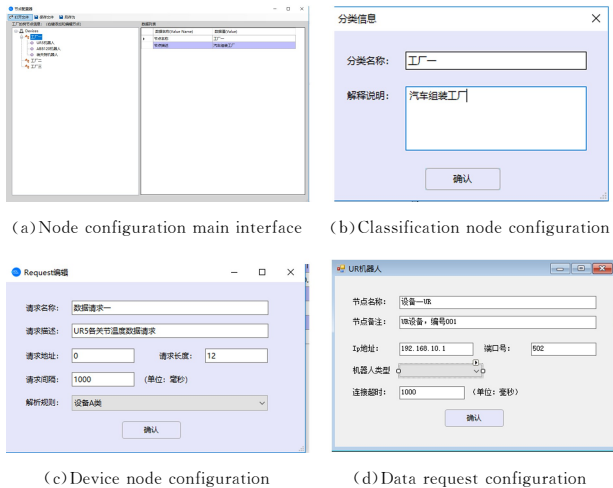


图 5 设备节点配置界面
Fig. 5 Device node configuration interface

解析规则配置界面是对数据请求的字节数组数据进行配置,由于数据地址在设备寄存器中多是连续的,每次数据请求都需要通过 Socket 通道发送一次数据读取的命令,采用一个数据发送一次命令的方式耗时过多,因此一次读取连续的一串数据,读出的数据如图 6 所示是一串字节数组,为了将一次数据请求读取的连续字节数组数据划分为具体的电压、电

流、电流、数据类型(Float、Bool)等。而这些信息都因通信协议的不同而不同,需要从官方所支持的该通信协议文档中获取。

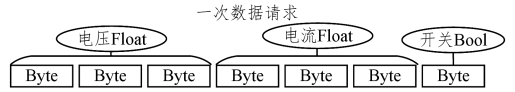
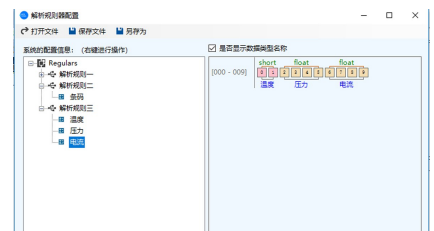


图 6 数据请求示意图

Fig. 6 Data request diagram

根据官方的说明文档将字节数组数据划分为单个值节点数据,并且配置好各值节点的数据名称、数据类型、字节长度等,以便解析模块通过该配置信息解析采集的工业机器人数据。图 7(a)为解析规则配置界面,界面左侧为数据请求以及各请求所包含的值节点名称,右侧展示一次数据请求所包含的值类型、长度和名称,图 7(b)为具体的值节点配置界面。



(a) Parse rule configuration main interface



(b) Value node configuration

图 7 解析配置界面

Fig. 7 Analytic configuration interface

4 云端数据处理系统设计

云端数据处理系统设计为数据存储和数据分析两部分,通过合理的数据库选型和存储结构搭建,将工业设备运行数据结构化存储在云端服务器,并利用云服务器的计算能力,结合机器学习算法对大量的历史数据和实时数据计算分析,实现一定程度的故障预测,减少突然停机带来的风险和损失。除此之外,基于云端数据存储,结合 Web 技术可以建立 B/S (Browser/Server) 结构的工业设备远程监控系统,利用存储的工业设备数据做进一步开发。

本系统的数据存储部分主要存储工业设备运行数据和工业设备相关参数信息,而且要具备一定的拓展性,除此之外,成本和使用便捷性也在考虑之列,结合实际需求,本文综合考虑 ElasticSearch, Redis, MySQL, MongoDB 等 4 个数据库的优缺点进行数据存储系统选型。由于系统基于 .NET 平台开发,而且要具有一定的存储容量,因此排除了使用 ElasticSearch 和 Redis 来存储工业设备的运行数据;再对比 MySQL 和 MongoDB,虽然 MySQL 的部署、管理和配置较为复杂,但是它的稳定的读写能力以及高可用、支持在线扩容等优点很符合本系统需求,因此选择 MySQL 作为系统的主体数据存

储数据库。除了使用 MySQL 存储工业设备大量的运行数据外,本系统还采用了非关系型数据库 Redis 作为系统的外部缓存,以提高系统热点数据的读取速度。Redis 数据库能够自动执行持久化操作,避免了服务器崩溃带来的热数据丢失,同时可以在服务器重启后自行恢复数据。

本系统采用 MySQL 和 Redis 相结合的数据存储方式,数据存储时经底层采集解析后的数据分别存入 Redis 和 MySQL 中。由于设备数据已由协议转换部分全部转换为 OPC UA 节点形式,每一个数据都有唯一的节点标识 NodeID,以 NodeID 为 Key,以值节点为 Value,将所有数据以 List 列表形式存入 Redis 中,同时对存入 Redis 中的数据设定过期时间和内存释放策略,基于时间局部性原理清除使用频率较低的数据,避免内存占用过高。数据在 MySQL 中的存储以分类节点建立表结构,中间表按照设备进行分类建立,最后以值节点的各属性值单独建立数据表。通过各表之间的关系清晰地反映节点的层次结构,以便上层软件的数据库查询操作。由于高访问的热点数据被转存在 Redis 中,数据查询时,首先从 Redis 中查找,Redis 查询无果则转向 MySQL 中搜索。

5 实验验证

5.1 OPC UA 客户端测试

基于 OPC UA 的数据采集和处理系统旨在实现多种工业设备数据的采集到 OPC UA 协议的转换,以达到使用同一个 OPC UA 客户端集中监控的目的,同时对采集的数据建立云端存储体系,便于进一步利用大量的工业设备运行数据实现故障预测分析。本文以 UR5 和 ABB120 两台工业机器人作为研究对象,验证整个系统的完整性、可靠性。针对 UR5 和 ABB120 分别采用 Modbus 协议和 PC SDK 按照系统的标准开发数据采集驱动和数据解析模块,将驱动装载入本地监控服务器端。

首先设置 IP 地址,在机器人控制器中将 UR5 设置为 192.168.10.10,ABB120 则设置为 192.168.10.11,搭载本地监控系统的主机 IP 设置为 192.168.10.1,从而使得三者在同一局域网内。如图 5 在节点配置界面中配置好 UR5 和 ABB120 设备节点的连接参数,并且分别添加六关节电流的数据请求,以 UR5 为例,数据请求的起始地址为 290,总长度为 6 个寄存器即 12 字节,在解析规则配置界面对该数据请求的 12 字节数据进行语义配置,设置其数据类型为 Float。如图 8 所示对 OPC UA 服务器系统进行设置,包括基本的登录用户名和密码、服务器地址以及安全代理“SecurityPolicyNone”,接着运行服务器。

如图 9 所示,通过官方开发完备的 OPC UA 客户端连接 OPC UA 服务器测试服务器的可靠性和完整性。



图 8 系统配置

Fig. 8 System configuration

在 Address 地址栏填入服务器的 URL 地址,建立连接成功后,客户端右侧框中显示服务器端配置的节点信息,选中

UR5 设备节点,查看设备的配置信息和数据请求的六关节电流数据,由此可以验证 OPC UA 服务器搭建成功,数据采集驱动和数据解析模块设计可行,系统可以实现工业机器人协议到 OPC UA 的转换,可以实现 OPC UA 客户端的远程监控功能。

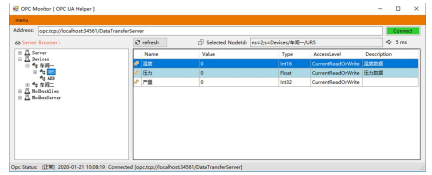


图 9 OPC UA 客户端

Fig. 9 OPC UA client

5.2 基于虚拟调试系统的测试

结合虚拟调试系统,测试数据采集系统采集数据和存储数据的准确性、完整性。虚拟调试用以测试运行在物理系统的程序逻辑,提高程序实际运行的安全性。在实验中,通过数据采集系统获取实际工业设备的运行数据,首先将其存储在云端数据库,然后虚拟调试系统从云端数据库获取实际工业设备全部的运行数据,通过这些数据驱动虚拟设备运行,观察虚拟设备是否能完成物理设备完整的操作。

采用如图 10 所示的虚拟调试系统,通过界面配置,首先连接虚拟设备,通过输入配置连接 Redis,获取数据采集系统采集的物理设备数据,通过输出配置将采集的物理设备数据导入虚拟设备中运行,对比虚拟设备和物理设备的运行情况。在 RoboDK 中导入机器人和车床的模型,然后构建虚拟平台,如图 11 所示。

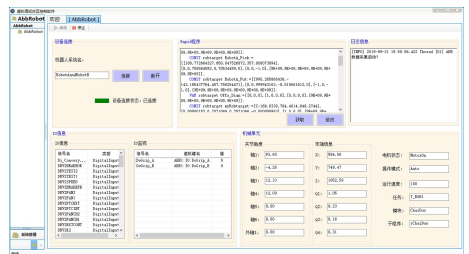


图 10 虚拟调试系统配置

Fig. 10 Virtual debug system configuration



图 11 RoboDK 虚拟设备

Fig. 11 RoboDK virtual equipment

通过物理设备的简单取件操作来测试系统,操作流程是机器人从仓库取出工件并放置于车床的卡盘上,然后车床门关闭。如图 12 所示,可以看到仿真机器人及时准确地从 Redis 中获取到了真实机器人的位置数据,并准确将工件放置在了卡盘上,虚拟设备通过采集的物理设备数据准确地复现了物理设备的操作,说明数据采集系统能准确完整地采集并且转换物理设备数据。

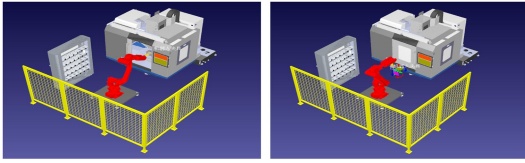


图 12 虚拟设备运行结果

Fig. 12 Virtual equipment running results

结束语 本文设计并且实现了基于 OPC UA 的数据采集系统,旨在实现不同工业设备协议到 OPC UA 标准协议的转换,以达到使用任意具有 OPC UA 客户端功能的设备就可以实现对工厂所有工业设备数据的远程查看和监控。本文通过设计数据采集接口来标准化工业设备不同协议的数据采集驱动。针对任意工业设备,只要按照标准接口设计好相应的驱动,在本地监控界面做简单的配置,就可以实现到 OPC UA 协议的转换。除此之外,本系统还建立了基于 Redis 和 MySQL 的云端数据存储结构,二者都具有良好的扩展性,存储的工业设备运行数据结合机器学习算法可以实现一定程度的故障预测分析,减少了设备突然停机带来的危害和损失。本文通过 UR5 和 ABB120 工业机器人验证了系统的可行性和完整性。所提系统解决了工业设备和协议多样带来的上位机软件多样化问题,具有较强的通用性。

参 考 文 献

- [1] TIAN C. Research on the current situation and development trend of industrial robots [J]. China Management Informatization, 2019, 22(20): 156-157.
- [2] WANG F D, ZHANG H L, YANG F. Research on robot information model based on OPC UA in the 70th lecture of industrial wireless communication technology lecture [J]. Instrument Standardization and Metrology, 2018(4): 12-15, 25.
- [3] LIU Y Y, SU X R, XU Z C, et al. Brief introduction of OPC UA technology in the 56th lecture on industrial wireless communication technology [J]. Instrument Standardization and Measurement, 2016(2): 21-24.
- [4] ZHAO Y H, NIE Y J, WANG Y L, et al. Overview of OPC UA technology [J]. Naval Chemical Defense, 2010(2): 33-37.
- [5] LI J X. Research on equipment data acquisition and remote monitoring system for intelligent factories [D]. Nanjing: Nanjing University of Aeronautics and Astronautics, 2018.
- [6] LIU W. Research on MES data management system based on OPC UA [D]. Beijing: Beijing University of Posts and Telecommunications, 2019.
- [7] ZHAN G C, WU M L, LU J Q, et al. Design of 828D CNC machine tool group remote monitoring system based on OPC UA [J]. Machinery Manufacturing & Automation, 2018, 47(6): 186-189.

- [8] XU B B. Design and implementation of key modules of data acquisition and monitoring system based on OPC UA [D]. Xi'an: Xi'an University of Science and Technology, 2017.
- [9] XU J M, PAN X F. Research on industrial robot monitoring system based on Socket communication [J]. Computer Measurement and Control, 2017, 25(7): 70-73.
- [10] LU H M, YAN Z F. research and development of OPC UA server address space key technology [J]. Power Automation Equipment, 2010, 30(7): 109-113.
- [11] WANG Z Z, XIE B H. design of OPC UA server based on MODBUS protocol [J]. Computer Application and Software, 2014, 31(2): 89-92, 131.
- [12] ZHAO Y L, YAN X F, LIU D. Design and implementation of embedded OPCUA server based on SDK [J]. Manufacturing Automation, 2017, 39(1): 49-52.
- [13] XU P F, QI G, SONG K, et al. Research on redis-based dual-channel real-time data acquisition system [J]. Science and Technology Innovation, 2019(25): 84-85.
- [14] LIU Y T, HU T N, LV Y L. Research on multi-protocol conversion system based on embedded technology [J]. Instrument Technology and Sensors, 2020(2): 62-66, 70.
- [15] XU M H, WANG F, ZHANG M. Application and research of Redis technology based on big data [J]. Information Technology and Informatization, 2019(11): 228-230.
- [16] CAVALIERI S, CHIACCHIO F, PUGLISI A D S. Integrating KNX and OPC UA Information Model [J]. Journal of Computers, 2014, 9(7): 1536-1541.
- [17] SCHÄFER F, BAKAKEU J, KLEINERT B, et al. Designing an OPC UA Based Ecosystem for Smarter Homes [J]. Advanced Engineering Forum, 2016, 4420: 83-91.
- [18] SHIN I J, SONG B K, EOM D S. Auto-Mapping and Configuration Method of IEC 61850 Information Model Based on OPC UA [J]. Energies, 2016, 9(11): 901-902.



YU Xin-yi, born in 1979, Ph.D, associate professor. His main research interests include embedded, industrial robot control and research.



OU Lin-lin, born in 1979, Ph.D, professor. Her main research interests include multi-intelligence coordination and so on.