

西门子广东省深江市（中国）授权总代理 一级代理

产品名称	西门子广东省深江市（中国）授权总代理 一级代理
公司名称	广东湘恒智能科技有限公司
价格	.00/件
规格参数	西门子变频器:西门子触摸屏 西门子伺服电机:西门子PLC 西门子直流调速器:西门子电缆
公司地址	惠州大亚湾澳头石化大道中480号太东天地花园2 栋二单元9层01号房
联系电话	18475208684 18475208684

产品详情

1 工业通信与OPC UA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研究较晚，但已有较多应用。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：<01>本地工业设备网络<02>本地监控服务器<03>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。图1 系统总体构成图Fig.1 Overall system composition diagram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。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OPCUA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图3

本地监控服务器的功能连接Fig.3 Function-connection 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OPCUASDK建立的OPCUA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。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服务器所需要的所有方法，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OPCUA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.net Framework 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。Co

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补满缺省字节位，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词典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础信息，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、开关状态等值数据，需要知道每段值节点在该数据请求获取的字节数组的索引值以及该值节点的字节长度、语义(电压、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。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。图8 系统配置Fig.8 System colfiguration图9 OPC UA客户端Fig.9 OPC UA client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。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。

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 colfiguration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