

# SIEMENS潮州市西门子（授权）中国总代理- 西门子华南区一级总代理商

产品名称	SIEMENS潮州市西门子（授权）中国总代理- 西门子华南区一级总代理商
公司名称	广东湘恒智能科技有限公司
价格	.00/件
规格参数	变频器:西门子代理商 触摸屏:西门子一级代理 伺服电机:西门子一级总代理
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（仅限办公）（注册地址）
联系电话	18126392341 15267534595

## 产品详情

### 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文件获来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OPCUA设备节点和值节点。解析后的数据填入建立好的OPCUA地址空间，完成OPCUA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

OPCUA服务器的构建主要有两种方式，即根据OPCUA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OPCUA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OPCUA服务器所需要的全部方法，不需要开发人员对OPCUA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OPCUA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OPCUA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。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，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(

)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Pot、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。