许春林, 曾培峰. 基于 Modbus 协议的工控设备数据虚拟化系统[J]. 智能计算机与应用,2024,14(6):163-168. DOI:10.20169/j.issn.2095-2163.240623

基于 Modbus 协议的工控设备数据虚拟化系统

许春林, 曾培峰
(东华大学 计算机科学与技术学院, 上海 201620)

摘 要: 针对工业控制现场网络不同设备的多链路通信数据传输问题,本文提出了一种基于 Modbus 协议的工控设备数据虚拟化方案。通过虚拟数据中心以及物理链路虚拟化、内外网的数据映射这两层虚拟化,实现主控机对现场网络内不同链路设备数据的统一流转与传输。该方案解决了不同链路数据传输的介质差异带来的协议变体问题,同时解决 Modbus 协议对于多字节类型数据难以区分的问题。在不改变原有协议标准的基础上增强了 Modbus 协议的数据表述能力,使得不同类型的数据能够在总线上灵活传输;对总线外部的数据访问请求提供统一的接口,降低了现场网络设备部署与维护的成本。

关键词: Modbus 设备虚拟化; 虚拟数据中心; 多链路数据传输; 数据映射
中图分类号: TP311 **文献标志码**: A **文章编号**: 2095-2163(2024)06-0163-06

Data virtualization system of industrial control equipment based on Modbus protocol

XU Chunlin, ZENG Peifeng
(College of Computer Science and Technology, Donghua University, Shanghai 201620, China)

Abstract: Aiming at solving the problem of multi-link communication data transmission between different devices in the industrial control network, a data virtualization method based on the Modbus protocol is proposed for the devices. This method is realized through virtual data center and two layers of virtualization including physical link virtualization and data mapping of internal and external networks. The master device transfers data among different devices in the unified manner in the research. This solution solves the problem of protocol variation caused by the media differences in data transmission in different links. Meanwhile, by enhancing the Modbus protocol without changing the original protocol standard, it also solves the problem for Modbus protocol to distinguish multi-byte data. The new data expression enables different types of data to be flexibly transmitted on the bus and the unified interface could be provided for data access requests outside the bus. Therefore, the proposal reduces the cost for field network device deployment and maintenance.

Key words: Modbus device virtualization; virtual data center; multilink data transmission; data mapping

0 引 言

目前工业界已形成了一种标准通信协议: Modbus 协议,国内也早在 2004 年发布了 Modbus 协议国家标准《基于 Modbus 协议的工业自动化网络规范》[1]。通过该协议,不同的工控设备可以在不同的数据链路层通信,如串口、以太网、无线网等。针对不同的传输介质,Modbus 协议也衍生出了很多种变体协议,如用于串口的 Modbus RTU、用于网络的 Modbus TCP/UDP 等,其在实现细节上略有不同,但在应用层面的协议功能是保持一致的。2012 年,Liu 等学者[2]介绍了一种基于 Modbus 协议的工业现场总线的平板度测控系统的实现方法,通过串行链路的 Modbus RTU 协议传输多通道涡流位移传感器采集平台的数据。2013 年,Liang 等学者[3]设计了一种基于 Modbus 协议的监测系统,实现对船舰环境的实时监测。2016 年,Chhatrawala 等学者[4]在现场可编程门阵列电路中通过 UART 端口传输 Modbus 协议数据,实现温度、电压、电流等数据的传输共享。2019 年,Pandey 等学者[5]则提出通过 WiFi 网络传输 Modbus 协议数据以实现水管理网络系统。上述的研究结果表明,Modbus 协议正成为各种硬件上的通用数据传输协议,应用前景十分广泛,但由于工业设备的复杂多样,现场网络主控机对不

作者简介: 许春林(1999-),男, 硕士研究生,主要研究方向:嵌入式系统,Email:2212547@mail.dhu.edu.cn;曾培峰(1964-),男,博士,教授,主要研究方向:嵌入式系统,图像处理。

收稿日期: 2023-04-24

同物理链路上的 Modbus 设备进行统一的连接并实现虚拟化是需要解决的一个重要问题。

2009 年,Yu 等学者[6]就已经提出了一种基于协调器的 ZigBee 协议/Modbus 协议转换方案,该方案利用网关的协议转换功能,将无线传感器网络节点虚拟成 Modbus 设备,从而扩展了 Modbus 协议的使用范围。2011 年,胡四泉等学者[7]也提出了无线传感器网络下类似的虚拟化方案。这 2 种方案均针对无线传感器网络下的传感器设备的虚拟化,即通过 Modbus 协议抽象传感器网络下的各种类型的设备,实现统一的数据收发,巧妙利用了 Modbus 协议的设备无关性,但传输介质仅为单一的无线传感器网络,并没有对多物理链路的设备进行处理。2022 年,Ocaña 等学者[8]提出了在以太网 TCP/IP 下的 Modbus 设备的进程虚拟化方案,该方法将 TCP/IP 网络下的不同 Modbus 设备抽象为虚拟的进程,通过网络实现一种跨域的分布式控制系统。2021 年,崔靖元等学者[9]提出基于 Modbus TCP 协议的地铁虚拟操纵台系统设计,实现电气试验车虚拟 Modbus 服务器,电气试验车实现硬件初始化,并提供各种控制服务处理等多个功能。该方案通过重新定义协议的方式虚拟了 Modbus 设备,并且只处理了串口与 TCP/IP 网络环境下的协议实现。从上述的研究可知,多物理链路下的 Modbus 设备虚拟化相关研究仍为空白,在不改变原有协议的基础上实现这样一层虚拟化的系统能够增强现场主控机功能性,使其能够灵活组网、连接更多的设备。

对于实际工控现场而言,主控机除了需要管理网络内部的设备以外还需要整合数据供外部网络的设备访问。面对杂乱的设备,系统需要进行整合对外提供一个统一的访问接口,来自内部网络的数据提供给外部访问的过程被称为 Modbus 数据的透传。2010 年,Wang 等学者[10]提出 Modbus 与 IEC 61850 的模型映射方法,包括 IEC 61850 对象引用与 Modbus 地址的映射、公共数据类型与 Modbus 参数类型的映射、抽象通信服务与 Modbus 协议数据单元的映射。2019 年,Silva 等学者[11]提出了一种将传统 Modbus RTU 传感器映射到 MQTT 物联网云的方法。以上研究为 2 种不同链路间数据映射提供了实现思路,具体就是实现数据从其他协议到 Modbus 协议的映射转换,但根据《基于 Modbus 协议的工业自动化网络规范》[1]所述,Modbus 协议数据与数据类型无关,以 16 位寄存器作为数据传输的最小单元进行传输,协议数据实际表示的内容由应用层软件自行定义,因此上述系统对不同的数据类型并没有做额外的工作处理。

在对以上相关文献和系统的研究基础上,本文综合了胡四泉等学者[7]在无线传感器网络下的虚拟化方案与 Ocaña 等学者[8]提出的在以太网 TCP/IP 下的 Modbus 设备的进程虚拟化方案,扩展了对串口、以太网、CAN 总线、无线网络等多种物理链路的统一处理方式,提出第一层虚拟化系统使系统在复杂的现场环境下能够灵活连接多路设备。同时,本文对 Wang 等学者[10]提出的数据映射方案进行了增强,提出第二层虚拟化系统,通过统一的 Modbus 协议实现内外数据的映射。综上所述,本文提出了 Modbus 协议设备的两层虚拟化:

(1)Modbus 设备物理链路传输虚拟化;

(2)设备寄存器内外映射及字节序虚拟化。

同时,为解决 Modbus 协议对于数据的表述能力较弱的问题,本文还提出一种虚拟数据中心(Virtual Data Center)的实现,将多链路数据以及本机数据统一流转至数据中心,对 4 种字节序:小端、大端、小端交换、大端交换进行处理和转换。由于目前绝大多数工控设备都是基于状态机模型设计的,虚拟数据中心也能满足这类设备的需求,有效与状态机系统结合,扩展原有系统的数据处理方式。

通过 2 层虚拟化及虚拟数据中心,能够有效解决 Modbus 设备种类繁杂的问题,降低工控现场网络设备的部署和维护成本,具有重要的意义和实际应用价值。

1 多链路设备二层虚拟化系统结构

本文提出的二层虚拟化系统包含三大子模块,分别是:虚拟数据中心;第一层虚拟化:物理链路协议报文转换;第二层虚拟化:内外网映射与多字节数据表述。

二层虚拟化系统结构如图 1 所示。由图 1 可看到,本系统以虚拟数据中心核心枢纽,存储多种类型的数据,内部网络设备与外部网络的设备均通过第一层虚拟化对不同物理链路报文进行处理后屏蔽掉底层链路的差异,并将数据传输到虚拟数据中心执行相关操作。虚拟数据中心中的数据经第二层虚拟化,通过使用一张映射表进行数据映射,将虚拟数据中心中已有的各种指定类型的数据以及内部不同的设备的数据提供给外部设备访问,并在实际映射的过程中自动完成多字节序列的处理与转换。

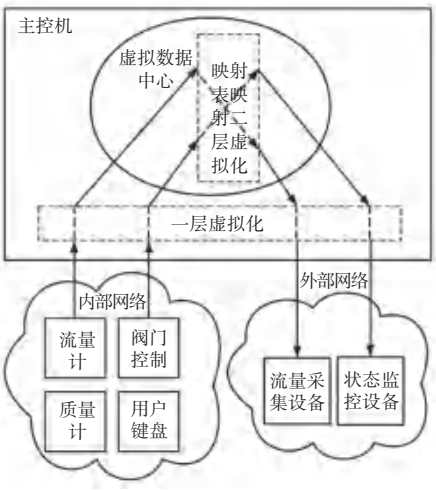

图 1　二层虚拟化系统结构图

Fig. 1　Structure of two-layers virtualization system

2　虚拟数据中心设计

2.1　整体设计

本文提出的二层虚拟化机制以虚拟数据中心作为数据的桥接中心，联系着内网、本机以及外部设备的数据。虚拟数据中心主要存储以下 3 种来源的数据，即：内网设备数据、主控机配置数据和多种数据运算处理后得到的新数据。

第二层虚拟化所需的映射表也在虚拟数据中心定义，通过访问虚拟数据中心，主控机的状态机程序或其他主控程序通过存取虚拟数据中心中的各种数据以控制主控机的工作状态和工作逻辑，实现内外部数据的交换与流转。

2.2　数据操作设计

虚拟数据中心将主控机的全部数据通过 *DataID* 进行划分，主控机 *DataID* 可由用户自行定义范围，每个 Data 对应一种类型的数据。虚拟数据中心数据类型见表 1。由表 1 分析可知，虚拟数据中心共包含 9 种数据类型：INT16、UINT16、INT32、UINT32、INT64、FLOAT32、FLOAT64、WCHAR、STRING，每种类型的数据均按照指定的字节序存储在虚拟数据中心。

从表 1 可知，前 8 种数据为数值型定长数据，而最后一种字符串 STRING 类型的数据则是不定长度非数值型数据。由于虚拟数据中心的 *DataID* 可映射到 Modbus 协议对应的 16 位寄存器，因此数据需要长度固定且为 2 的整数倍。而由于 STRING 类型不定长的特殊性，虚拟数据中心对该类型的数据使用特殊的方式进行处理，在该类型数据定义时通过设置“*Len*”属性来指定字符串类型的最大字节长度，字符串数据在进行存取时最后一个字节必须为 0，同时最大长度不能超过 *Len* 属性所定义的最大数据容量。

表 1　虚拟数据中心数据类型

Table 1　Types of data in virtual data center

数据类型	数据长度/byte	寄存器数	数据格式
INT16	2	1	字节序列
UINT16	2	1	字节序列
INT32	4	2	字节序列
UINT32	4	2	字节序列
INT64	8	4	字节序列
FLOAT32	4	2	IEEE 754
FLOAT64	8	4	IEEE 754
WCHAR	2	1	GBK 码点
STRING	Len 属性值	数据长度/2	GBK 字节序列

系统对表 1 中不同数据类型抽象出了一组通用数据操作接口，抽象接口包含了四则运算、移位操作、赋值操作、比较运算、逻辑判断、数据转换等基本运算。通用数据操作接口调用如图 2 所示，特定的数据类型需实现全部接口供虚拟数据中心调用对其进行运算。

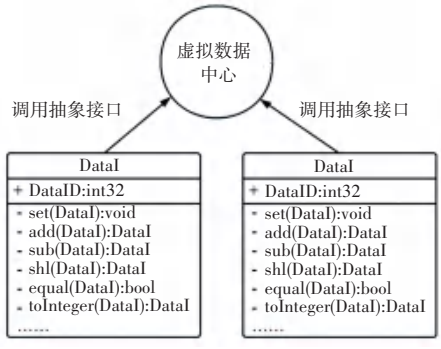

图 2　通用数据操作接口调用图

Fig. 2　General data operation interface call

分析图 2 可知，主控机虚拟数据中心存取、计算相关数据时以 *DataID* 作为唯一的标识符进行操作，通过调用抽象数据接口实现数据的运算。主控机虚拟数据中心通过 XML（EXtensible Markup Language）文件进行配置与定义，在 XML 配置中通过<Data />标签完成一个 Data 数据的定义。每个 Data 标签使用一系列属性配置该 Data 的相关数据与操作，共有如下 6 种属性的定义：

（1）*DataID*：定义该数据的唯一 ID 号；

（2）*Type*：定义数据类型；

（3）*Value*：定义数据初始值；

（4）*Method*：定义数据取值的运算中缀表达式；

（5）*ModReg*：定义数据所映射的 Modbus 寄存器号；

（6）*Len*：STRING 类型数据专有，定义最大数据长度。

其中，*ModReg* 属性在二层虚拟化中映射 Modbus 寄存器使用，而 *Len* 属性则定义 STRING 类型数据的最大数据长度。*Value* 属性与 *Method* 属性同时只能存在一个，有 *Value* 属性的数据是原生的数据，该数据可被自由读写；有 *Method* 属性的数据不是原生数据，该数据需要运用 *Method* 属性值定义的表达式通过调用其他数据的抽象接口进行运算得到，因此该类型数据只可读不可写。*Method* 中的表达式运算在 *C* 语言数学表达式的基础上拓展了如下所示多种表达式格式：

(1)“[DataID]”：取出 *DataID* 的数据；

(2)“(Type)”：对 Data 类型按值转换；

(3)“(*Type)”：对 Data 类型按实际字节转换。

其中，为了使虚拟数据中心的运算能够灵活取出数据，增加了取值 *DataID* 数据的运算符，通过方括号加 *DataID* 的方式来取出指定 *DataID* 的数据供后续运算。例如，“[1000] + [1001]” 表示取出 *DataID* 为 1000 与 1001 的数据执行加法运算。同时，方括号运算符可以进行多级嵌套运算，例如“[[1000]]” 是一个二级嵌套，运算时会先取出 *DataID* 为 1000 的数据，再将该数据的值作为 *DataID* 再次取值，实现类似 C 语言多级指针的效果。

通过上述 XML 文件可定义主控机内虚拟数据中心的数据格式，为2层虚拟化转换提供了统一的数据操作中心，后续工作的展开都将围绕虚拟数据中心进行。

3 二层虚拟化机制

3.1 第一层虚拟化：协议报文转换机制

根据《基于 Modbus 协议的工业自动化网络规范》[1]，Modbus 协议报文中与实际通信物理链路无关的是协议数据单元(Protocol Data Unit, PDU)[1]，而在实际应用中针对特定的物理链路，Modbus 协议应用数据单元(Application Data Unit, ADU)[1] 在 PDU 的基础上增加了一些附属域，ADU 数据帧如图 3 所示。

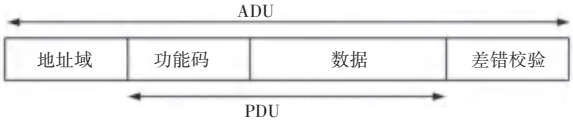

图 3 通用 Modbus 数据帧

Fig. 3 General data frame of Modbus

除了规范中的 ADU 如 Modbus RTU、Modbus TCP/UDP 以外，在实际应用中也存在很多自定义修改的 Modbus 协议用于其他规范未指定的物理链路，如 CAN 总线、无线传感器等，来协调不同物理链路的差异或实现其他协议扩展功能。Modbus RTU 与 Modbus TCP 在 3 号功能码读指令报文数据帧上的差异如图 4 所示。

协议	MBAP报文头	地址	功能码	寄存器地址	寄存器数	CRC
ModbusRTU	无	01	03	01 02	00 04	E4 35
ModusTCP	00 00 00 00 00 06 00	无	03	01 02	00 04	无

图 4 Modbus RTU 与 TCP 数据帧

Fig. 4 Data frames of Modbus RTU and TCP

由图 3、图 4 可以看出，实际协议应用数据单元的差异主要在于头部地址域与尾部校验域，这取决于实际传输链路是否可靠以及传输链路寻址的方式。针对实际应用场景也存在一些定制的协议应用数据单元，例如本系统主控机对标准 Modbus UDP 协议做了一定拓展，对于 UDP 协议不可靠传输的问题，在标准 UDP 协议应用数据单元尾部增加了 2 字节的 CRC 校验来保证报文较长时数据的正确。

因此，本系统的第一层虚拟化设计关键在于实现不同物理链路的数据传输后对不同的 Modbus ADU 进行区别和处理，第一层虚拟化框架如图 5 所示：

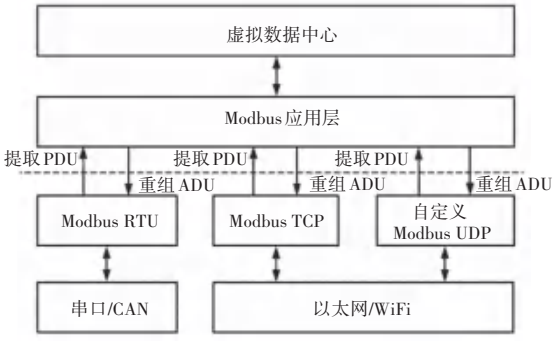

图 5 第一层虚拟化框架

Fig. 5 Architecture of the first layer of virtualization

图 5 中，第一层虚拟化对不同物理链路入站的

数据提取出实际的 Modbus PDU 并交给 Modbus 应用层，Modbus 应用层再根据功能码读写虚拟数据中心中对应的数据，由虚拟数据中心调用抽象数据接口完整数据的操作；对于出站的数据，Modbus 应用层从虚拟数据中心获取数据后再根据实际物理链路在 PDU 基础上加上头尾部的附属域后重组成完整的 ADU 报文并交由底层物理链路传输。

主控机通过实现第一层数据的虚拟化屏蔽不同物理链路下 Modbus 协议 ADU 的差异，使得系统能够同时接入不同物理链路的设备，对接入不同设备的数据能够进行统一的处理，提高了系统的灵活性。

3.2　第二层虚拟化：内外网映射与多字节数据传输

第二层虚拟化通过 *DataID* —寄存器映射表实现内外网 Modbus 设备数据的映射，将不同的内网 Modbus 设备寄存器进行重新分配或进行多数据重组，对外部设备提供统一的访问接口，并由虚拟数据中心实现多字节数据类型的字节变换。此时，通过第二层虚拟化主控机对外部设备来说是一个 Modbus Slave 设备，即主控机被动接收外部设备的数据请求，通过虚拟数据中心查找映射关系并调用抽象数据接口返回相应的数据。

图 6 为二层虚拟化完整数据流程。图 6 中，*D*1、*D*2 两个 *Data* 的定义如下：

< Data *ID* = "1" *Value* = "0" *Type* = "FLOAT32" *Comment* =" 流量计"/ >

< Data *ID* = "2" *Value* = "1" *Type* = "FLOAT32" *ModReg* = "01" *Comment* = " 流量系数"/ >

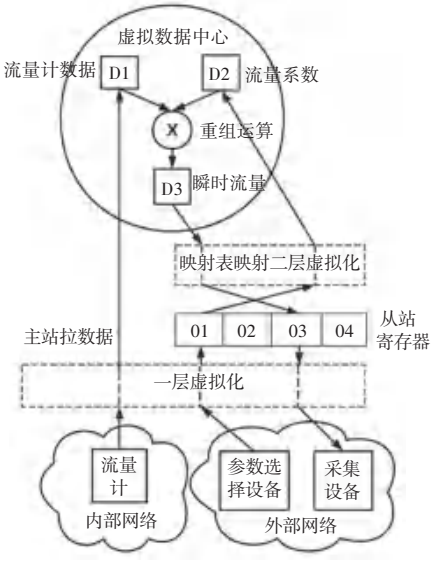

图 6　二层虚拟化数据流程图

Fig. 6　Flowchart of two-layers data virtualization

图 6 中，*D*1 表示保存流量计数据的 1 号 Data，主控机此时作为 Modbus 主站，通过第一层虚拟化从内网流量计指定寄存器中提取数据存储在 *D*1 中。*D*2 表示保存流量系数的 2 号 Data，通过 *ModReg* 属性映射到对外部提供访问的 1 号寄存器，由于 FLOAT32 类型数据长度为 4 字节，需要 2 个 Modbus 寄存器来存储，因此 1、2 号寄存器被占用保存着 1 号 Data 的数据，后续定义 Data 映射的 Modbus 寄存器则不能再使用 1 号和 2 号寄存器。外部设备此时可以通过读写这 2 个寄存器选择不同的流量系数。

*D*3 表示多数据重组后的新数据，定义如下：

< Data *ID* = "3" *Method* = "[1] * [2]" *Type* = "FLOAT32" *ModReg* = "03" *Comment* = " 瞬时流量"/ >

其中，*Method* 属性使用了表达式运算对数据进行重组，将 1 号与 2 号的数据相乘后得到新的数据，并将运算结果映射到 3 号、4 号寄存器。外部采集设备通过访问 3 号、4 号寄存器即可得到重组后的计算结果。使用数据重组的映射方式得到的新数据为只读，反向修改数据会被定义的数据依赖关系覆盖。

主控机通过实现第二层数据的虚拟化能够在表达式运算与数据抽象接口的帮助下完成对多种类型的多字节数据灵活重组，并将数据灵活映射至外部寄存器接口，供外部设备获取使用。内部实际存储的字节序列由虚拟数据中心维护，并根据系统的设定调整数据的字节序列，包括大端、小端、大端交换、小端交换四种字节序。实现第二层虚拟化能够有效解决 Modbus 协议不能够区分多字节类型数据的问题，充分利用 Modbus 协议的设备无关性灵活传输各种类型的数据。

4　实验与分析

本文提出的基于 Modbus 协议的工控设备数据二层虚拟化系统目前已在全志 A33 四核心 CortexTM-A7 架构 ARM 处理器下的 Android 系统实现，通过多线程的方式实现不同模块的功能。系统实现中虚拟数据中心定义了 65 536 个 Data 进行测试，系统全局创建的线程如下：

(1) 虚拟数据中心表达式运算与数据更新线程；

(2) 串口链路 Modbus 主站拉数据线程；

(3) 以太网链路 Modbus 主站拉数据线程；

(4) 串口链路 Modbus 从站监听线程；

(5) 以太网链路 Modbus 从站监听线程；

(6)通用 Modbus 应用层线程。

其中,虚拟数据中心的线程负责数据的运算与更新;Modbus 不同物理链路主站、从站线程用于实现第一层虚拟化,负责各自协议报文的转换,并与 Modbus 应用层线程互相传输与底层链路无关的应用层报文数据;Modbus 应用层线程则负责处理应用层报文,包括第二层虚拟化寄存器映射的数据传输,调用虚拟数据中心的抽象接口完成数据的读写操作。

在实际设计中,虚拟数据中心线程、Modbus 主站线程、Modbus 应用层线程均由非阻塞式循环实现,通过调节线程循环的 Sleep 间隔来平衡 CPU 负载。而 Modbus 从站监听线程属于阻塞式线程,因此不需要在循环中加入 Sleep 间隔,在没有连接时不占用 CPU 性能。系统线程参数与性能见表 2。

表 2 系统线程参数与性能

Table 2 Parameters and performance of system threads

线程类型	线程数	是否阻塞	Sleep 间隔	单次循环平均时间
表达式运算与数据更新	1	否	10 ms	23 ms
串口 Modbus 主站	1	否	50 ms	48 ms
以太网 Modbus 主站	4	否	50 ms	26 ms
Modbus 应用层	4	否	50 ms	34 ms
串口 Modbus 从站	2	是	无	阻塞等待
以太网 Modbus 从站	2	是	无	阻塞等待

由表 2 可知,表达式运算与数据更新占用一个线程,为了保证数据的刷新速度,线程 Sleep 间隔设置为 10 ms,完成一轮数据更新需要 23 ms 的时间,整体耗时满足实时刷新的要求。Modbus 主站循环时间受读写寄存器长度与报文数据长度影响,由于串口速率较慢,因此以太网 Modbus 主站循环耗时较小。实验中,采用 1 路串口主站、4 路以太网 Modbus 主站拉取数据,Modbus 从站最多使用 2 个线程响应连接,Modbus 应用层使用 4 线程线程池处理 Modbus 命令,在全志 A33 处理器下系统能够稳定运行,平均 CPU 占用率能够控制在 40%左右,内存占用在 50 MB 以下,满足工控设备需求。

5 结束语

本文提出了一种基于 Modbus 协议的工控设备数据二层虚拟化系统,针对工业控制现场网络不同设备的多链路通信数据传输问题,充分利用 Modbus 协议的设备无关性传输数据并扩展了 Modbus 协议的数据表述能力,灵活映射多种类型的数据。该系统在 Android 系统下实现并测试,运行效率良好,满足工控设备要求,可用于油库发油等现场作业。另外,系统多线程同步与协作效率仍需进一步优化,未来工作将对该问题做更加深入的探讨与研究。

参考文献

[1] 中国国家标准化管理委员会. 基于 Modbus 协议的工业自动化网络规范(第 1 部分):GB/Z19582.1-2004[S]. 北京:中国标准出版社,2004.

[2] LIU Xijun, HU Bing, AN Jinxin, et al. Design of surface plates flatness detection system based on Modbus protocol[C]// 2011 International Conference on Future Materials Engineering and Industry Application (ICFMEIA 2011). Bali Island, Indonesia: Trans. Tech. Publications Ltd., 2012, 365: 67-72.

[3] LIANG Sheng, PAN Gaofeng, ZENG Shanping, et al. Design and implementation of environmental monitoring system based on bus [J]. Advanced Materials Research, 2013, 712: 1999-2002.

[4] CHHATRAWALA J N, JASANI N, TILVA V. FPGA based data acquisition with Modbus protocol [C]//2016 International Conference on Communication and Signal Processing (ICCSP). Melmaruvathur, India: IEEE, 2016: 1251-1254.

[5] PANDEY B, FARULLA G A, INDACO M, et al. Design and review of water management system using ethernet, Wi-Fi 802.11n, Modbus, and other communication standards [J]. Wireless Personal Communications, 2019, 106(4): 1677-1699.

[6] YU Chengbo, LIU Yanfei, WANG Cheng. Research on ZigBee wireless sensors network based on Modbus protocol[J]. Wireless Sensor Network, 2009(1):43-47.

[7] 胡四泉,杨金阳,汪红兵,等. 基于无线传感器网络的 Modbus 虚拟设备[J]. 计算机应用,2011,31(S1):8-10,14.

[8] OCAÑA W S, RUBIO E A, LÓPEZ E T, et al. Design and implementation of an industrial multinetwork TCP/IP of a distributed control system with virtual processes based on IOT [C]//Proceedings of Sixth International Congress on Information and Communication Technology. Singapore: Springer, 2022: 797-812.

[9] 崔靖元,魏春光,曾洁. 基于 Modbus TCP 协议的地铁虚拟操纵台系统设计[J]. 自动化与仪表,2021,36(10):24-27,37.

[10] WANG Dewen, YAN Chunyu, BI Jiangang, et al. An approach to mapping between Modbus and IEC61850 for condition monitoring communication gateway in substations[J]. Automation of Electric Power Systems, 2010, 36(19): 78-84.

[11] SILVA C R M, SILVA F A C M. An IoT gateway for Modbus and MQTT integration [C]//2019 SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference (IMOC). Aveiro, Portugal: IEEE, 2019: 1-3.