文章编号：2095-2163(2021)09-0191-06 中图分类号：TP311.52 文献标志码：A

# 工控网络仿真靶场虚拟化场景的构建

孙 健，翟健宏

（哈尔滨工业大学 网络空间安全学院，哈尔滨 150000）

**摘 要**：工控网络仿真靶场对开展面向工业控制系统的安全测试、攻防演练和教学培训具有重要作用。当前，工控靶场以实物、半实物形态为主，建设成本高、维护代价大、可扩展性不强、场景单一固化等问题，不利于推广应用。本文基于虚拟化的工控网络仿真靶场，提出了一种场景构建方案，设计实现了 PCBA 智能制造场景。该场景符合工控系统规范，支持 OPC 统一架构，可扩展性强，适于开展工控网络安全测试和教学培训。

**关键词**：工控靶场；虚拟场景；OPC UA

## A virtual scene construction of industrial control network simulated range

SUN Jian, ZHAI Jianhong

(School of Cyberspace Science, Harbin Institute of Technology, Harbin 150000, China)

**【Abstract】** Industrial control network simulated range plays an important role in security test, attack and defense drill, teaching and training for industrial control system. At present, the industrial control range is mainly in the form of physical and semi physical with high construction cost, high maintenance cost, poor scalability and single scene, which is not conductive to the promotion and application. In this paper, based on the virtual industrial control network simulated range, a scene construction scheme is proposed and the PCBA intelligent manufacturing scene is designed and implemented. The scene conforms to the industrial control system specification, supports OPC unified architecture, and has strong scalability, which is suitable for industrial control network security test, teaching and training.

**【Key words】** industrial control range; virtual scene; OPC UA

## 0 引 言

工业互联网作为新型基础设施和战略新兴产业，是中国迈向制造业强国进程中的重要发展领域。《“十四五”规划和 2035 年远景发展目标纲要》提出：“要加快工业互联网建设，推动互联网、大数据、人工智能等各产业深度融合，推动先进制造业集群发展”，将工业互联网列为数字经济重点产业。

挑战与机遇并存。随着工控系统与互联网的深度融合，原来的“信息孤岛”也面临来自互联网的安全风险。由于工控系统广泛应用于能源、交通、制造等关系国计民生和国家安全的关键领域，保障工控系统安全的重要性十分突出。现运行的工控系统由于高可靠性和安全性要求，难以开展渗透测试、漏洞挖掘和攻防演练等带有破坏性的研究工作。因此，必须构建能够满足安全研究需求的工控网络仿真靶场，这对开展工控网络安全测试、组织面向工控网络的攻防演练、培育高水平工控安全人才队伍具有重要的现实意义[1-2]。

当前，工控网络仿真靶场的形态以全实物和半实物为主[3]。虽然实物类靶场具有仿真程度高的优点，但是投入成本高、建设周期长、维护代价高、可扩展性和灵活性低，场景单一、固化，不利于推广应用。有鉴于此，本文基于虚拟化工控网络仿真靶场，提出了一种场景构建方案，设计实现了 PCBA 智能制造场景，完善了工控网络仿真靶场的建设。

## 1 相关技术

### 1.1 工业控制系统

工业控制系统（Industrial Control Systems，ICS）是由各种自动化控制和过程控制组件构成的，确保工业基础设施自动化运行的业务流程管控系统。其核心组件主要包括：数据采集与监控系统（Supervisory Control and Data Acquisition，SCADA）、分布式控制系统（Distributed Control Systems，DCS）、可编程逻辑控制器（Programmable Logic Controller，PLC）、远程终

作者简介：孙 健（1990-），男，硕士研究生，主要研究方向：工控安全；翟健宏（1968-），男，硕士，副教授，硕士生导师，主要研究方向：网络内容安全、工业信息安全、云计算等。

通讯作者：翟健宏 Email：zhaijh@hit.edu.cn；

收稿日期：2021-05-26

端(Remote Terminal Unit,RTU)和人机交互界面 Human Machine Interface,HMI)等。

各组件的通信采用工控系统特有的协议。工控通信协议种类繁多,常见的标准协议有 Modbus、DNP3、IEC104、OPC 等;私有协议有西门子 S7、欧姆龙 FINS 等。本文构建的 PCBA 智能制造场景中,OPC Server 与各个节点、各个节点之间均采用 Modbus TCP 协议通信。

### 1.2 OpenStack

OpenStack 是一个开源的云计算管理平台,由美国国家航空航天局和 Rackspace 合作研究并发起,其目标是提供实施简单、可大规模扩展、丰富、标准统一的云计算管理平台。OpenStack 目前拥有近 30 个组件,其核心组件主要包括认证服务(Keystone)、计算服务(Nova)、网络服务(Neutron)、镜像服务(Glance)、对象存储服务(Swift)和块存储服务(Cinder)等。本文使用 OpenStack 框架的 Stein 版本搭建虚拟化平台,创建工控网络仿真靶场的基础环境。

### 1.3 PLC

PLC 是广泛用于自动化控制领域的数字运算控制器,由 CPU、指令及数据内存、输入/输出接口、电源、数字模拟转换等功能单元组成,可以将控制指令随时载入内存进行储存与执行。PLC 采用“顺序扫描,不断循环”的方式进行工作,一个扫描周期要经过输入采样、程序执行和输出刷新 3 个阶段。

PLC 是根据工控系统要求和实际业务流程,按照 PLC 编程语言规范设计实现的控制程序。根据国际电工委员会制定的工业控制编程语言标准(IEC1131-3),PLC 的 5 种标准编程语言是:梯形图语言(LD)、指令表语言(IL)、功能模块语言(FBD)、顺序功能流程图语言(SFC)、结构文本化语言(ST)。本文使用开源软件 OpenPLC Runtime[4-5] 作为 PLC 软实现,使用 ST 语言设计实现各节点的 PLC 程序。

### 1.4 OPC 统一架构

应用过程控制的对象连接和嵌入技术(OLE for Process Control,OPC),是工控领域的一种数据访问机制,用于为不同供应商生产的工控设备和应用程序之间提供标准化接口,解决数据交互的跨平台、跨协议问题。

早期 OPC 标准基于微软的 OLE/COM 技术,支持多种语言和代码重用,但由于对 Windows 平台的依赖性,已不能满足工控系统的发展需求。2006 年,OPC 基金会推出新一代技术标准:OPC 统一架构(OPC Unified Architecture,OPC UA)。与早期标准相比,OPC UA 不仅包括了数据访问、历史数据访问、报警和事件、安全等不同方面的功能,而且在其基础上集成了 Web 服务,构建了统一的数据模型,规范了复杂的地址空间,为协议实现、信息建模和服务器设计提供了支撑[6-8]。

本文基于 OPC UA 规范建立靶场场景的信息模型,使用开源的 Open62541 作为开发包完成 OPC Server 的设计与开发。

## 2 PCBA 场景建模

印刷电路板装配(Printed Circuit Board Assembly,PCBA)是对 PCB 空白板进行表面贴装的制造过程,主要包括载板、印刷、贴片、回流焊、插件、波峰焊和质检等流程。目前,PCBA 已经高度自动化、智能化,涵盖了机器人、传送带、传感器和控制器等智能制造领域的典型组件,具有要素齐全、代表性强、复杂度适中、规模可扩展的特点,适于构建工控网络仿真靶场虚拟化场景。本文选取某 PCBA 实验床为对象建立场景模型。

### 2.1 场景描述

如图 1 所示,空白 PCB 板从进料口进入,按逆时针方向依次完成 6 道工序后从出料口递出,整条流水线形成一个闭合环路。其中,6 条工位传送带各由一个驱动电机驱动,单向传送,2 个转接传送带组各由 4 个驱动电机驱动,双向传送。

### 2.2 场景模型

上述场景可以抽象为主控制器、工位和转接组 3 个组件,分别用 3 套控制算法描述各组件的运行逻辑:

#### 2.2.1 符号说明

$M$(驱动电机):$M_1 \sim M_6$ 表示 6 个工位传送带驱动器,$M_{A1} \sim M_{A4}$ 表示转接组 A 的 4 个驱动器,$M_{B1} \sim M_{B4}$ 表示转接组 B 的 4 个驱动器;

$R$(机器人):$R_1 \sim R_6$ 表示 6 个工位机器人;

$P$(定位传感器):$P_1 \sim P_6$ 表示 6 个工位定位传感器,$P_{A1} \sim P_{A3}$ 表示转接组 A 的 3 个定位传感器,$P_{B1} \sim P_{B3}$ 表示转接组 B 的 3 个定位传感器;

$L$(传输线):工位 1、2、3 和两个转接组构成传输线 $L_1$,工位 4、5、6 和 2 个转接组构成传输线 $L_2$,$signal\_L_1$ 和 $signal\_L_2$ 表示相应的控制信号。

#### 2.2.2 工位控制算法

工位根据控制信号和定位传感器状态,控制机器人和工位传送带驱动电机,其控制算法描述如下:

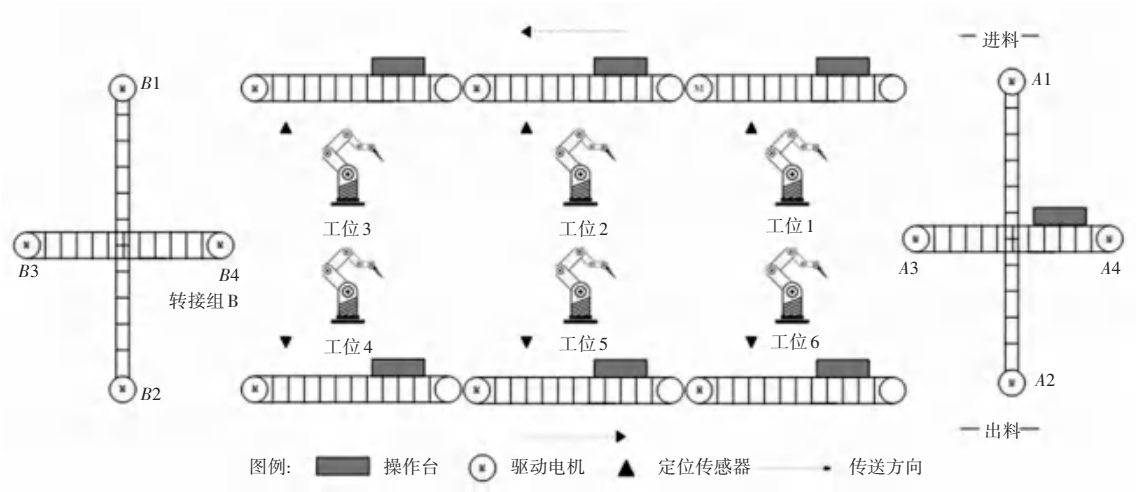


**图 1　PCBA 场景图**

**Fig. 1　PCBA Scene**

$P_n$ 为真时，停止 $M_n$，启动 $R_n$；

$R_n$ 完成预定工时后，停止 $R_n$，将 $P_n$ 复位；

$P_n$ 非真，$signal\_L_1$ 为真时，启动 $M_n$。

2.2.3　转接组控制算法

转接组根据控制信号和定位传感器状态，控制 4 台传送带驱动电机的运行，其控制算法描述如下（以转接组 A 为例）：

（1）$M_{A1}$ ~ $M_{A4}$ 分别驱动传送带向上、下、左、右 4 个方向转动；$P_{A1}$、$P_{A2}$ 分别判断转接组是否与传输线 $L_1$、$L_2$ 对齐，$P_{A3}$ 判断操作台是否居中就位；

（2）初始状态下，$P_{A1}$ 为真，$M_{A1}$ ~ $M_{A4}$、$P_{A2}$、$P_{A3}$ 非真；

（3）$signal\_L_1$ 为真时，启动 $M_3$；

（4）$P_1$ 为真时，停止 $M_3$，启动 $M_2$，复位 $P_{A1}$；

（5）$P_{A2}$ 为真时，停止 $M_2$；

（6）$signal\_L_2$ 为真时，启动 $M_{A4}$；

（7）$P_{A3}$ 为真时，停止 $M_{A4}$，启动 $M_{A1}$，复位 $P_{A2}$；

（8）$P_{A1}$ 为真时，停止 $M_{A1}$。

2.2.4　主控制器控制算法

主控制器根据各定位传感器状态，发出控制信号，其控制算法描述如下：

（1）$P_1$、$P_2$、$P_3$、$P_{A1}$、$P_{B1}$ 都为真时，将 $signal\_L_1$ 置真，否则复位；

（2）$P_4$、$P_5$、$P_6$、$P_{A2}$、$P_{B2}$ 都为真时，将 $signal\_L_2$ 置真，否则复位。

### 2.3　信息建模

完成场景建模后，根据 OPC UA 信息建模规范和流程，按照面向对象的方法对场景进行信息建模。为此定义以下 4 种类型：

（1）驱动电机类型：成员包括转速变量、运行状态变量、启动方法和停止方法；

（2）机器人类型：成员包括运行状态变量、启动方法和停止方法；

（3）工位类型：成员包括驱动电机对象、机器人对象、定位传感器变量；

（4）转接组类型：成员包括 4 个驱动电机对象和 3 个定位传感器变量。

场景信息模型如图 2 所示。


**图 2　信息模型**

**Fig. 2　The information model**

## 3 场景构建与部署

### 3.1 虚拟化靶场平台搭建

使用 OpenStack:stein 搭建靶场的虚拟化平台,平台硬件资源见表 1。

主机 controller 作为控制节点,部署有 keystone 组件、glance 组件、placement 组件、nova 组件、neutron 组件和 zun 组件,负责资源管理和调度;主机 compute1、compute2、compute3 作为计算节点,部署有 nova 组件、neutron 组件,负责提供计算和网络资源;其中主机 compute3 还安装了 docker 服务和 zun 组件,负责提供容器服务。

**表 1 靶场资源**

**Tab. 1 Range resources**

| 主机名 | CPU | RAM | 磁盘 | 内网 IP | 管理 IP | 角色 |
|---|---|---|---|---|---|---|
| controller | 24core | 32GB | 7TB | 192.168.54.2 | 192.168.0.2 | 控制节点 |
| compute1 | 24core | 64GB | 2TB | 192.168.54.5 | 192.168.0.5 | 计算节点 |
| compute2 | 24core | 64GB | 2TB | 192.168.54.6 | 192.168.0.6 | 计算节点 |
| compute3 | 24core | 64GB | 2TB | 192.168.54.7 | 192.168.0.7 | 计算节点 |

### 3.2 场景各节点构建

按照场景模型,创建 1 个节点作为主控制器,6 个节点作为工位,2 个节点作为转接组。各节点在虚拟化平台中通过容器实现,容器中运行 OpenPLC Runtime。使用 ST 语言将场景模型中各组件的控制算法编写成 PLC 程序,上传至 OpenPLC Runtime 中运行。各节点之间使用 Modbus TCP 协议通信,其中主控制器为 Modbus 主站,其余各节点均为 Modbus 从站。主从站结构如图 3 所示:

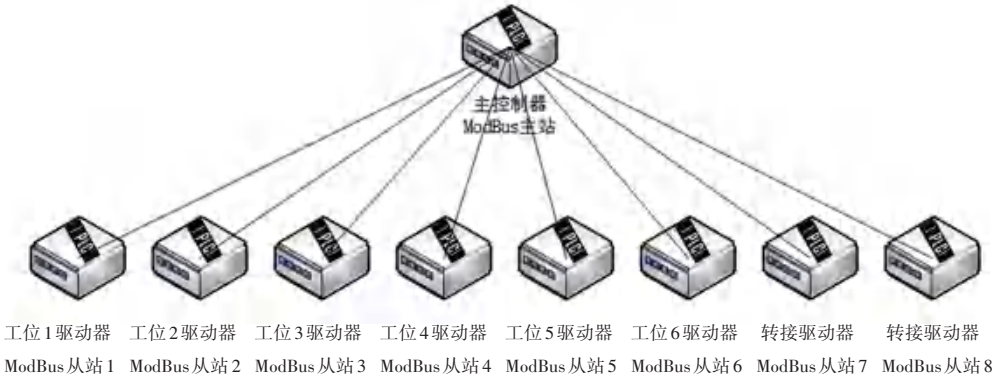


**图 3 主从站结构**

**Fig. 3 The structure of master and slave stations**

### 3.3 OPC Server 开发

OPC Server 的主要功能是监测和采集各节点数据提供给客户端,向各节点传达客户端指令。OPC Server 使用 VS2019 开发,主要开发流程如下:

(1)导入信息模型。将 2.3 中导出的 xml 文件载入服务器工程;

(2)添加对象节点。向服务器中添加 6 个工位对象节点和 2 个转接组对象节点;

(3)建立连接。使用 libmodbus 库与 3.2 中各节点建立 Modbus TCP 连接;

(4)添加监测项和回调函数。将各个对象中的状态变量设置为监测项,为对象中的启动方法和停止方法设置回调函数,在回调函数中完成对相应地址空间的读写操作。

### 3.4 部署运行

完成各节点构建和 OPC Server 开发后,按照以下步骤完成场景部署:

(1)按照工控 5 层网络模型,在虚拟化平台中创建虚拟网络,场景网络划分详情见表 2;

**表 2 场景网络划分**

**Tab. 2 Scene network division**

| 层级 | 功能 | 网络名称 | 地址划分 |
|---|---|---|---|
| 0 | 访问接入 | Provider | 10.10.10.0/24 |
| 1 | 企业管理 | Mes-network | 10.1.1.0/24 |
| 2 | 过程控制 | Scada-network | 10.1.2.0/24 |
| 3 | 现场控制 | PLC-network | 10.1.3.0/24 |
| 4 | 现场设备 | Device-network | 10.1.4.0/24 |

(2)创建各节点虚拟机和容器,如图 4 所示;

| name | image | status | task_state | addresses | ports |
|---|---|---|---|---|---|
| plc-slave-1 | openplc | Running | None | 10.1.4.101 | [] |
| plc-slave-2 | openplc | Running | None | 10.1.4.102 | [] |
| plc-slave-3 | openplc | Running | None | 10.1.4.103 | [] |
| plc-slave-4 | openplc | Running | None | 10.1.4.104 | [] |
| plc-slave-5 | openplc | Running | None | 10.1.4.105 | [] |
| plc-slave-6 | openplc | Running | None | 10.1.4.106 | [] |
| plc-slave-7 | openplc | Running | None | 10.1.4.107 | [] |
| plc-slave-8 | openplc | Running | None | 10.1.4.108 | [] |
| mes-webserver | cirros | Running | None | 10.1.1.119 | [] |
| plc-master | openplc | Running | None | 10.1.3.212 | [] |
| mes-DA | cirros | Running | None | 10.1.1.94 | [] |
| scada-HMI | cirros | Running | None | 10.1.2.232 | [] |
| scada-database | cirros | Running | None | 10.1.2.195 | [] |

**图 4　场景节点**

**Fig. 4　Scene nodes**

(3)启动 OPC Server,如图 5 所示;

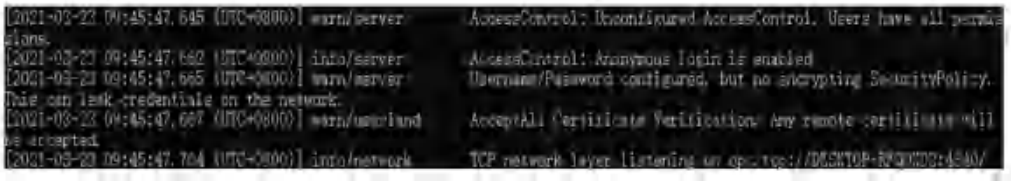


**图 5　OPC Server 运行**

**Fig. 5　OPC Server is running**

(4)运行与测试,如图 6 所示。

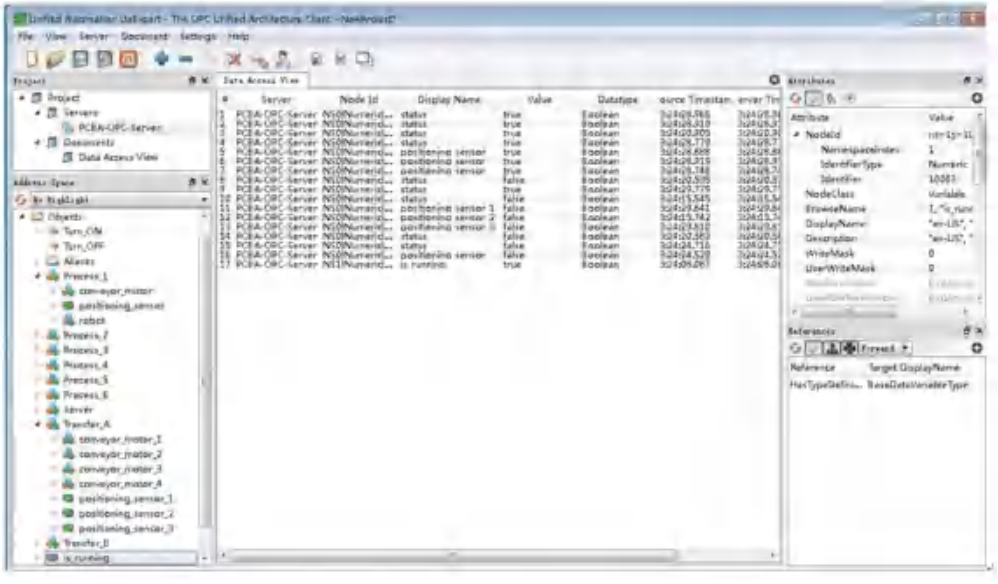

**图 6　使用 UaExpert 监控场景运行状态**

**Fig. 6　Using UaExpert to monitor the running state of the scene**

场景运行后,使用 UaExpert 软件作为客户端连接 OPC Server。在 UaExpert 中调用启动函数,发出启动指令,场景开始运行。通过观察 UaExpert 中的监测变量和 OpenPLC Runtime 中的变量变化状态,可以判断,变量变化状态符合控制逻辑和模型设计,场景运行状况良好。

## 4　结束语

本文以 PCBA 智能制造生产线为对象,通过场景建模、信息建模、PLC 程序设计和 OPC Server 开发等环节完成了一种工控虚拟化场景的设计与实现,并在工控网络仿真靶场中完成部署。通过运行

(下转第 199 页)