

论文独创性声明

1. 论文作者（若与其他作者共同合作完成作品，则论文作者已获得其他所有合作作者的授权发表此声明）在此声明所呈交论文系其独立或集体完成的作品。文中依法引用他人成果，并均已做出明确标注并得到许可。论文内容未包含法律意义上已属于他人而未被他人授权使用的任何形式的研究和创作成果。
2. 论文作者对稿件内容的真实性及思想观点负责，论文由论文作者自负文责。
3. 作者保证论文不会侵犯第三人的任何权利和利益，包括但不限于他人的著作权、商业秘密。
4. 论文作者未曾将本论文提交于以往各届的西门子工业专家会议。
5. 论文提交至西门子工业专家会议会务组即表明论文作者同意上述声明。

数字化解决方案SIMICAS在电子电器组装制造行业的应用

张健

西门子工厂自动化工程有限公司

北京

中文摘要:

企业成长期会遇到管理模糊、数据记录不透明及难以发现改善点的问题，因此一套可以实现生产 KPI 透明化及记录分析的管理工具成为企业中上层管理人员不可多得的法宝。基于产线建模的 SIMICAS 云应用，它专注于发现产线存在的问题，并通过首次投资小，实施快速敏捷，易用和免维护等先天优点吸引众多成长期企业加入数字化建设浪潮。SIMICAS Tool 组态工具对不同的产线进行模型化，通过组态完成 IO 映射，最后发布到智能网关 IOTBOX2040 和 MindSphere。智能网关 IOTBOX2040 采集产线模型相关的实时数据，将工业现场的传感器、PLC 和服务器数据以 KPI 的维度呈现在云端 SIMICAS APP 上，并通过数据趋势、报表为客户发现问题提供有力帮助。

中文关键词： 西门子MindSphere工业云，SIMICAS生产透镜和产效分析APP， 智能物联网网关 IOTBOX，数采系统及产线KPI等

Abstract

This paper introduces that enterprise will encounter problems of chaotic workshop management, opaque data records and difficult to find improvement points during the growth period. Therefore, a set of management tools that can realize the functions of the transparency of production KPIs, recording, analysis has become a rare magic weapon for middle and upper managers of enterprises. SIMICAS Cloud Application Based on production line modeling focuses on finding the problems existing in the production line and attracts many growing enterprises to join the wave of digital construction through the inherent advantages of small initial investment, fast and agile implementation, easy to use and maintenance free. SIMICAS Configuration Tool is designed to build models for different production lines, completes IO mapping through configuration, and finally releases them to intelligent gateways iotbox2040 and MindSphere. The intelligent gateway IOTBOX 2040 collects the real-time data related to the production line model, then presents the sensor, PLC and server data from the industrial filed on the cloud SIMICAS app in the dimension of KPI and provides powerful help for customers to find problems through data trends and reports.

Keyword: Siemens MindSphere Cloud, SIMICAS Metrics Performer and Performance Analyzer APP, Intelligent IOT BOX Gateway, Data Acquisition System and KPI of Manufacture Line.

一、项目简介

本文以 SNC（西门子南京数控有限公司）项目的后道测试包装工序为背景，展开介绍。SNC 以变频器和电机为主要产品，SIMICAS 分别成功应该用生产变频器的工厂和生产电机的工厂，产线主要工艺步骤主要有上料、模块化组装、测试、最终测试和自动化包装等，两个工厂位于不同的工业园区。

图 1 为 SNC 变频器组装测试产线模型，图 2 为 SNC 电机组装测试喷漆包装产线模型。

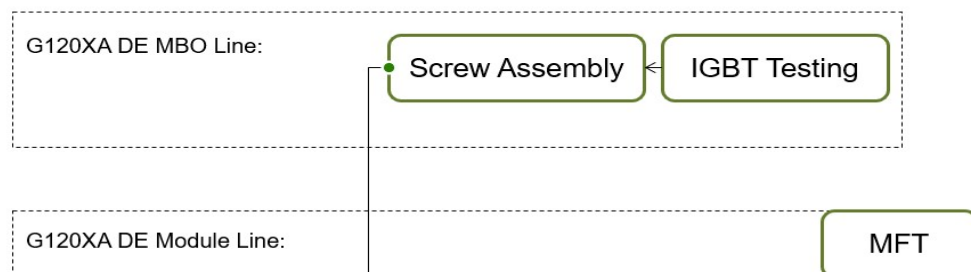


图 1. SNC 变频器组装生产线

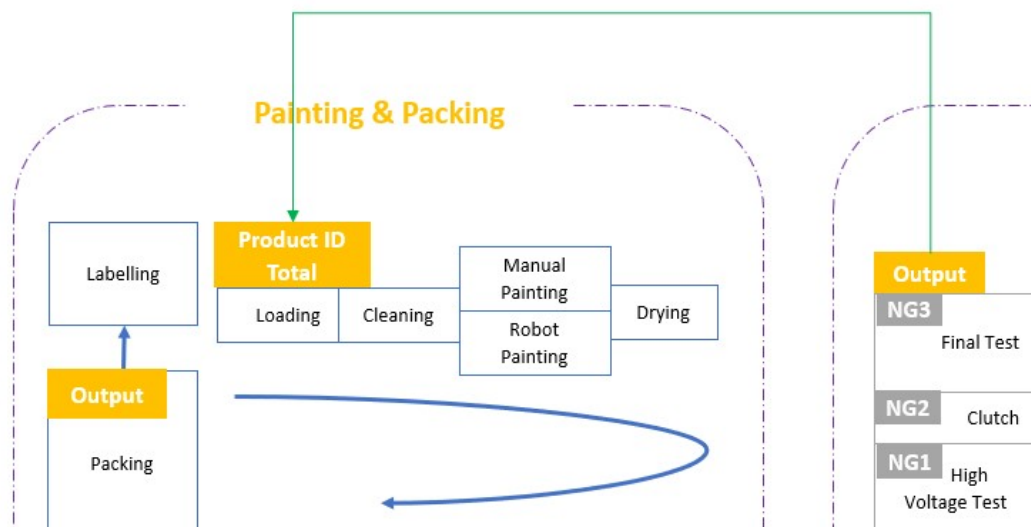


图 2. SNC 电机组装测试喷漆包装产线模型

二、系统结构

本项目数采箱硬件主要有：1 台 S7-1215C PLC、1 台 IOT2040、1 台 SCALANCE XB005 交换机、1 台 24V 直流电源、1 台华为路由器和一些按钮。如下图



图 3. MOF SIMICAS 数采箱照片

首先，电子器件组装厂的设备大部分由计算机控制，与我们平常见到 PLC 控制的产线网络有较大不同，这些主机都是在生产网络的管理下，所以项目选取一台华为路由器的 WAN 口接入客户生产网，其 LAN 口连接 PLC（TCP Server 端），建立与 Radiator、Assembly、MFT 和 Packing 工站 FMI 计算机（Client 端）TCP socket 连接。其次，通过 XB005 交换机建立起 PLC、SIMICAS BOX 和 Packing 机器人的稳定的局域网连接并进行 S7 通信。产线实时数据进入到 IOTBOX 后，通过其 X2 口连接到客户办公网，并通过 HTTP 代理服务器连接到 MindSphere。此外，要特别注意 SIMICAS Tool 的部署，要顺利将组态建模完成的产线模型发布到 IOTBOX 和云端并且 SIMICAS APP 能持续收到数据，该计算机必须随时与他们保持通讯连接。

产线数采系统现场数据流如图 4 所示。

Data flow in SIMICAS System

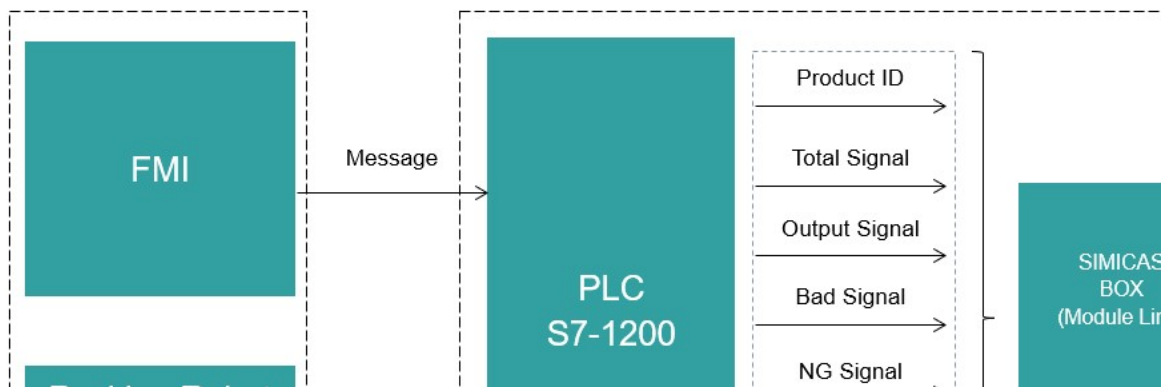


图 4. 产线数采现场数据流

为了实现上述的数据流，SIMICAS 部署于 Mindsphere 云端，具体网络规划如图 5 所示：

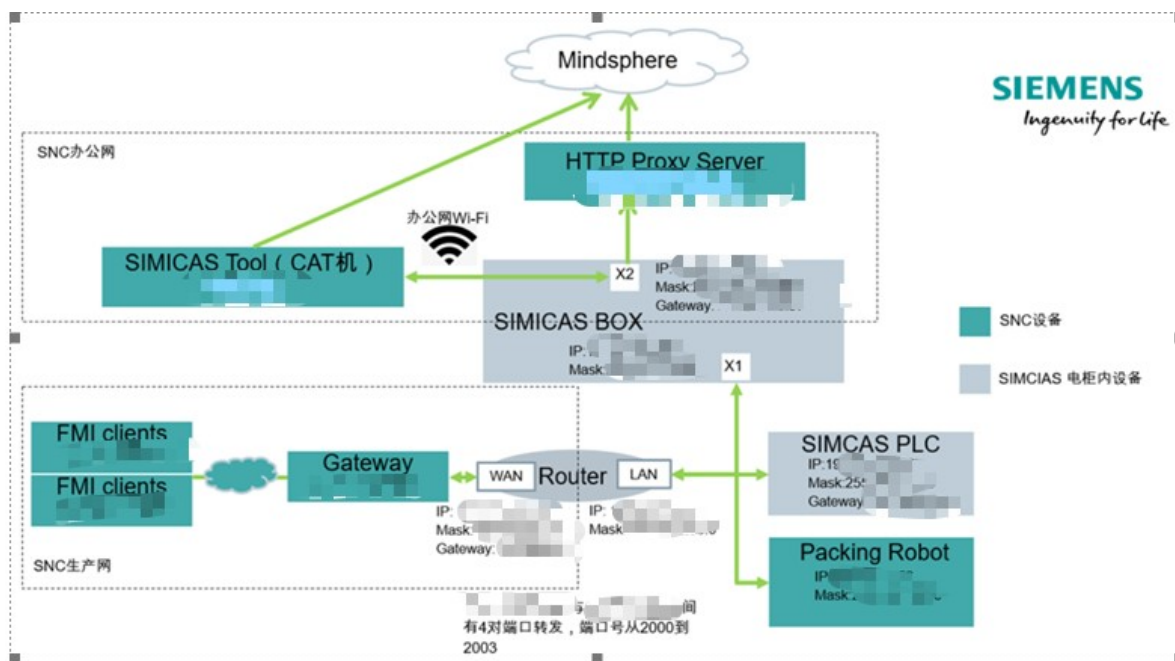


图 5. 网络规划

SIMICAS 解决方案和传统软件方案的根本区别。从功能上看，传统的企业管理软件，大多数的功能专注于固定的流程运作、无纸化或是代替手工工单和记录等，对于发现问题的作用微乎其微；从投资上看，传统大型软件管理系统首次投资都是百万级的，后续要支付昂贵

的维护费用。如果根据不同的产线定制不同的软件系统，虽然可以解决当下问题，但如果换一个生产线，又需要重新架构编程，软件复用性差，对于开发费用和周期是不小的考验。

三、功能与实现

3.1 产线建模和组态

为了完成图 4 的 KPI 指标，需要进行产线建模和组态。MindSphere 定义产线为 Asset，产线下的 KPI 为 Aspect，所以在定下客户需求时已经明确了产线模型应该如何组态。一方面，SIMICAS Tool 针对不同场景，对不同的 KPI 预定义了不同的计算公式，不同公式包含不同的计算因子，以及 KPI 显示多种数据格式等，如图 6。另一方面，根据这个特性，产线模型定义完成的同时规定了 IOTBOX 要采集的数据集以及 PLC 发送给 IOTBOX 的数据接口，比如说 DB 块等，如图 7。



图 6. 产线及工站 KPI 组态

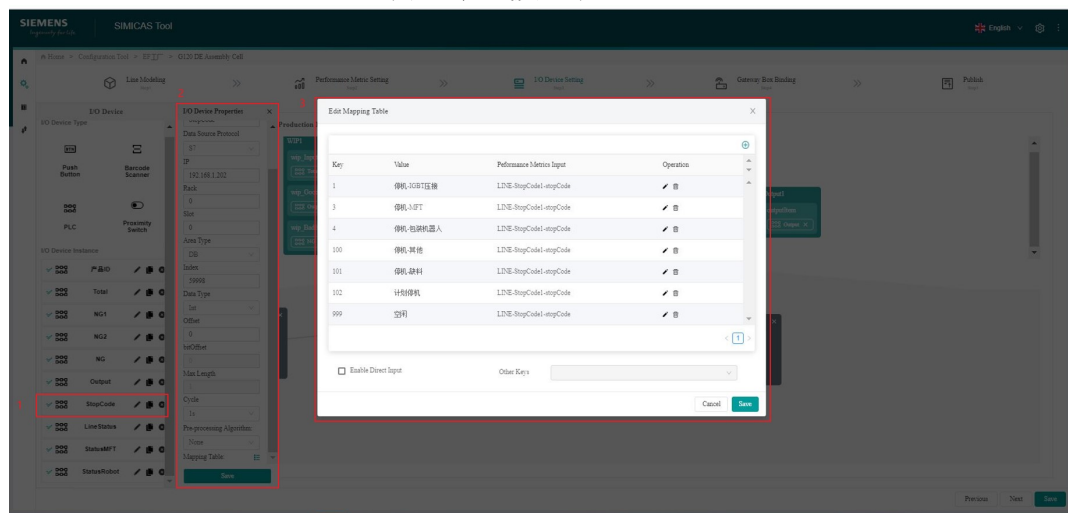


图 7. 计算因子 IO 实例配置及值的映射

上图中的一些关键字，解释如下：

标记 1，是 SIMICAS Tool 组态的数据变量，如 StopCode；标记 2，是变量 StopCode 对数据源进行采集的协议和地址设置；标记 3，是该变量的值与显示元素的映射关系。

3.2 PLC 程序设计

根据产线模型的定义，先设计出程序数据接口，如 Product ID、Total、Output、Line Status 等，如图 8。

PLC_SNC_Tiaoshi_20200328 ▸ PLC_SNC [CPU 1215C DC/DC/Rly] ▸ 程序块 ▸ TO_SIMICAS_DB [DB16]

保持实际值 快照 将快照值复制到起始值中 将起始值加载为实际值

TO_SIMICAS_DB										
	名称	数据类型	起始值	监视值	保持	可从 HMI...	从 H...	在 HMI ...	设定值	注释
1	Static									
2	PID	String	''			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
3	Total	Int	0			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
4	NG1	Int	0			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
5	NG2	Int	0			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
6	NG	Int	0			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
7	Output	Int	0			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
8	Stop_Code	Int	0			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
9	Line_Status	String	''			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
10	Status_MFT	Int	0			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
11	Status_Robot	Int	0			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

图 8. PLC S7 通信 DB 块

接着，定义数采程序流程图，如图 9。

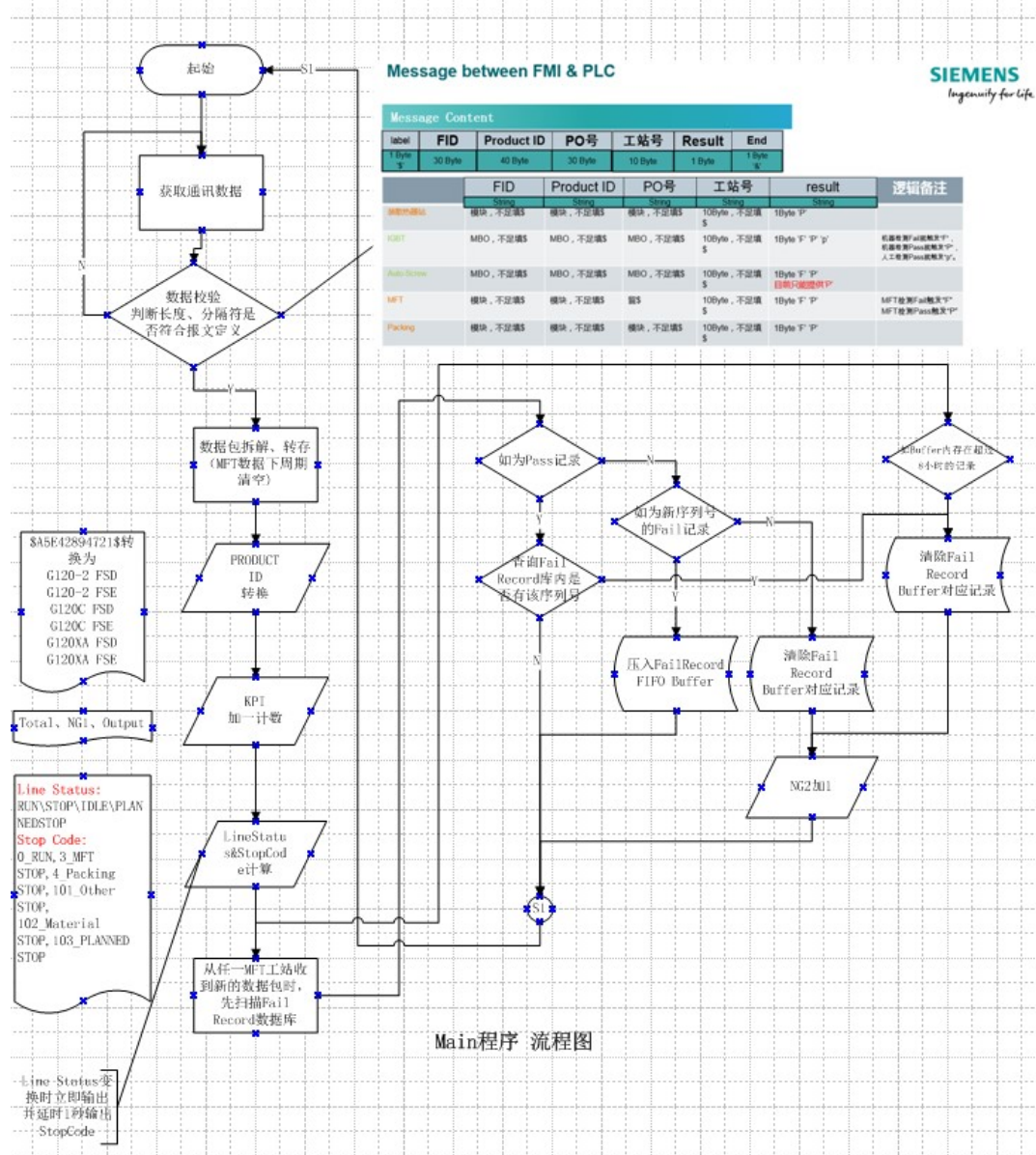


图 9. PLC Main 程序流程图

编写通信功能块用以接收 FMI 计算机的 TCP 通信数据（如图 10），Robot 机器人 S7 通信数据（如图 11）

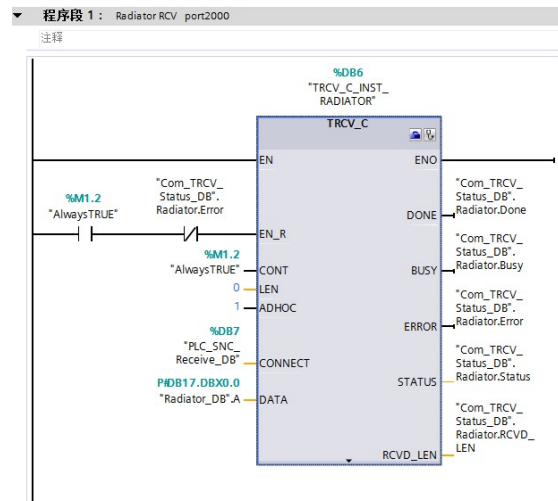


图 10. PLC 与 FMI 计算机 TCP SOCKET 通信

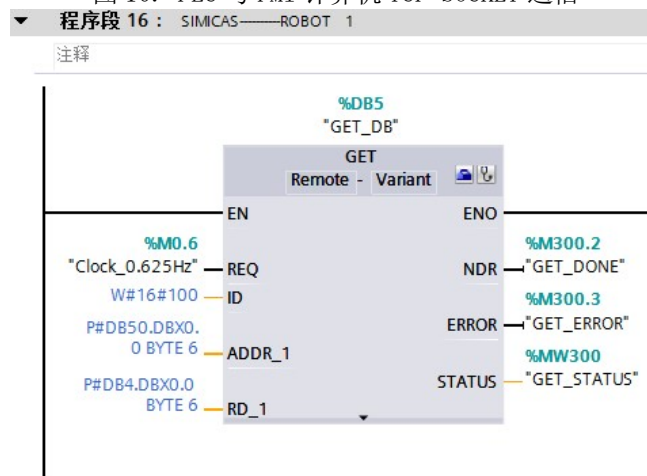


图 11. PLC 与 ROBOT PLC S7 通信

编写 Line Status& Stop Code 程序块，功能分为设置进入不同 Stop Code，Stop Code 清零，以及通过 Case of 选择进入不同的 Line Status。

编写数据校验拆解程序块，判断从伙伴发过来的数据包（字符串数组）是否符合要求，如符合通过 MOVE BLOCK VARIANT 转存到存储器中，同时在下一个周期对这些数据清空，如图 12 和图 13。

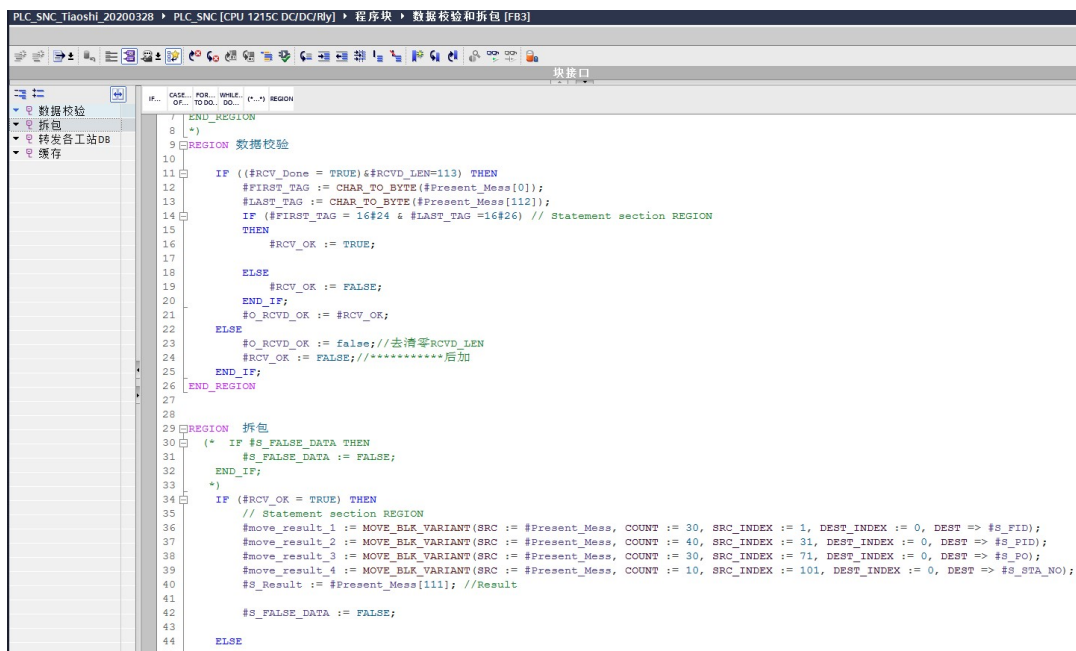


图 12. 数据校验拆解程序块（数据校验、提取数据）

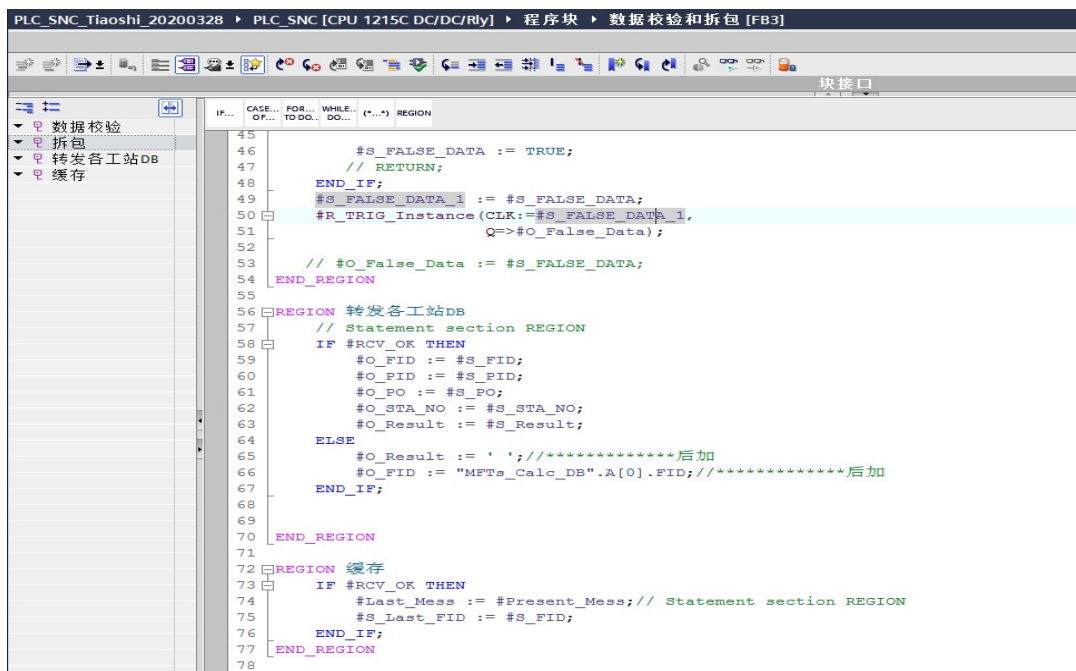


图 13. 数据校验拆解程序块（转发、缓存当前信息）

编写 Fail Record FIFO 数据库，存储每一条记录的产品序列号以及时间戳，该库上限为 100 条，如图 14 和图 15。

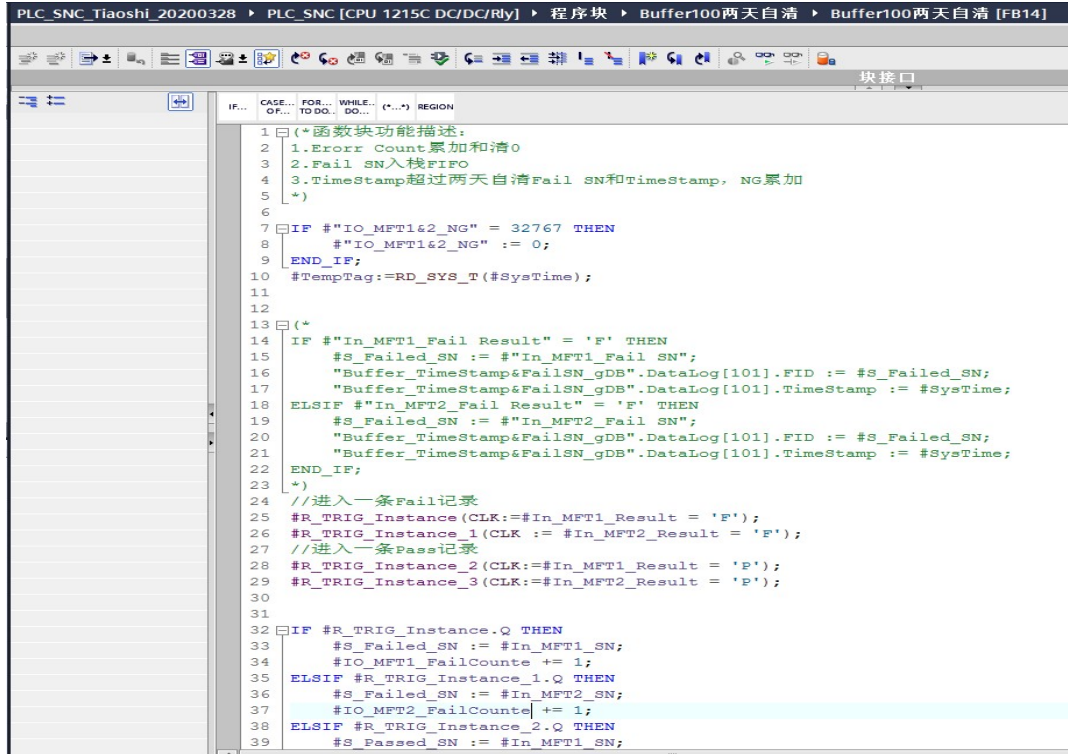


图 14. Fail Record FIFO 数据库（记录的读取和异常处理）

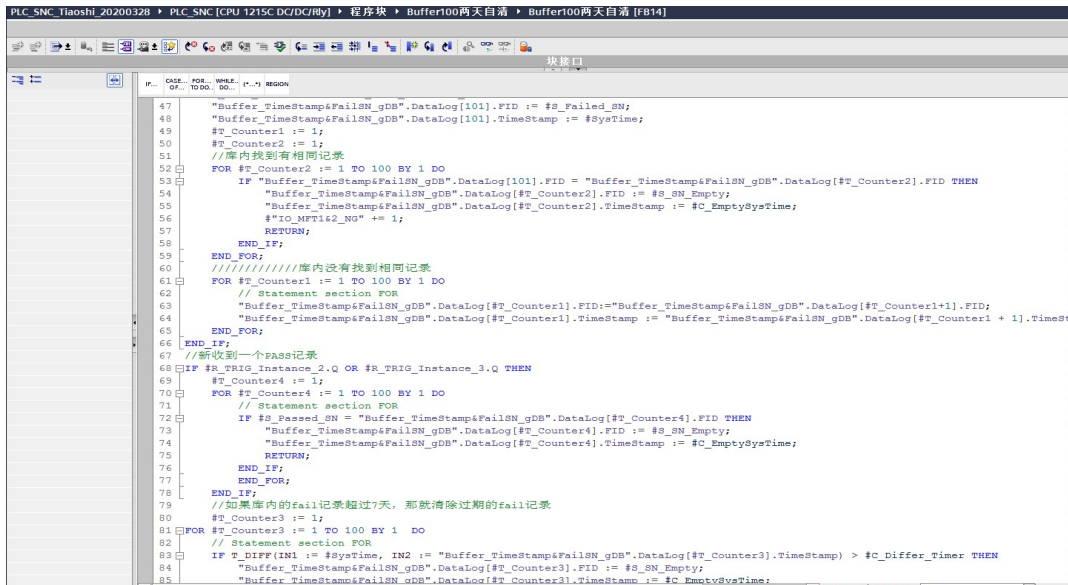


图 15. Fail Record FIFO 数据库（数据记录查询和处理机制）

3.3 SIMICAS APP 上数数据流

项目涉及到的软硬件比较多，有一些非常陌生如 IOTBOX 数据采集处理，客户自主开发的 FMI 和 MFT 软件，甚至路由器的设置等，所以必须清楚每个节点的数据格式，通信方式，上数的状态以及故障应对措施等，面对种种项目的难点，绘制出 SIMICAS APP 实施项目节点级上数数据流用以诊断是非常必要的，具体如图 16。

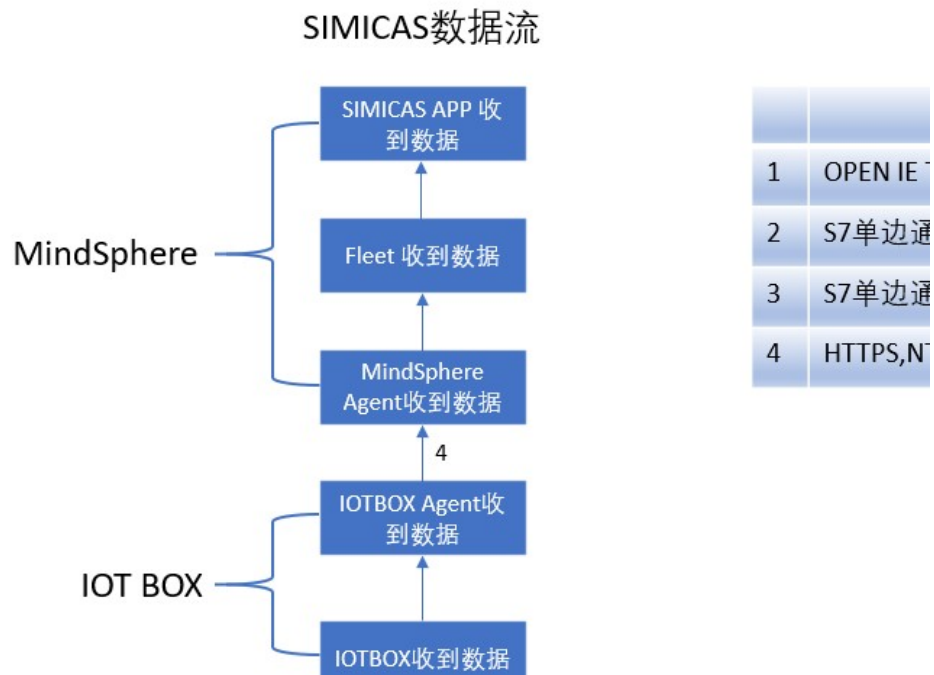


图 16. SIMICAS APP 上数据流

3.5 IOTBOX 采集并发布的数据

通过指令 `~/exec/mb_sub -t opcua/data` 和 `~/exec/mb_sub -t MessageTypePublished` 查看 IOTBOX 采数和计算后的发布到 Agent 的数据情况。具体如下图。值得说明的是，消息队列中的数据名称与 SIMICAS TOOL 组态保持一致，每一条数据都带有时间戳，如图 17 和图 18。

```

Receive Message from server, topic: opcua/data, content length: 79, content: {"key": "S7-f2ff312b707353ae/Total", "timestamp": "1588770066521", "value": "1301"}
Receive Message from server, topic: opcua/data, content length: 79, content: {"key": "S7-f2ff312b707353ae/Total", "timestamp": "1588770087893", "value": "1302"}
Receive Message from server, topic: opcua/data, content length: 79, content: {"key": "S7-f2ff312b707353ae/Total", "timestamp": "1588770910100", "value": "1303"}
Receive Message from server, topic: opcua/data, content length: 79, content: {"key": "S7-f2ff312b707353ae/Total", "timestamp": "1588770995202", "value": "1304"}
Receive Message from server, topic: opcua/data, content length: 92, content: {"key": "S7-f2ff312b707353ae/Product ID", "timestamp": "1588771005871", "value": "KUKA1FK708_P"}
Receive Message from server, topic: opcua/data, content length: 83, content: {"key": "S7-f2ff312b707353ae/LineStatus", "timestamp": "1588771005948", "value": "RUN"}
Receive Message from server, topic: opcua/data, content length: 79, content: {"key": "S7-f2ff312b707353ae/Output", "timestamp": "1588771006067", "value": "972"}
Receive Message from server, topic: opcua/data, content length: 87, content: {"key": "S7-f2ff312b707353ae/Delayed StopCode", "timestamp": "1588771007235", "value": "0"}
Receive Message from server, topic: opcua/data, content length: 79, content: {"key": "S7-f2ff312b707353ae/Output", "timestamp": "1588771007286", "value": "977"}
  
```

图 17. IOTBOX 采集数据

上图中的一些关键字，解释如下：

标记 1，是 IOTBOX 通过 OPC UA 协议采集的数据；标记 2，产线的总投入 Total；其余标记依次是：产线目前生产的产品 ID，产线状态，产量和停机原因。


```

ger_resetter", "timestamp": "1588771066132"}
Receive Message from server, topic : MessageTypePublished, content length : 125, content : {"value": {"category": "class", "action": "reset", "target": "FPY"}, "key": "trigger
_resetter", "timestamp": "1588771066159"}
Receive Message from server, topic : MessageTypePublished, content length : 125, content : {"value": {"category": "class", "action": "reset", "target": "WIP"}, "key": "trigger
_resetter", "timestamp": "1588771066167"}
Receive Message from server, topic : MessageTypePublished, content length : 171, content : {"value": {"ProductID": "KUKA1FK708_P"}, "ChangeOver": "true"}, "key": "ChangeOver",
"timestamp": "1588771066171", "hashtag": ["Station:LINE"], "PRODUCTID": "KUKA1FK708_P"}
Receive Message from server, topic : MessageTypePublished, content length : 133, content : {"value": {"Status": "Run"}, "key": "Status", "timestamp": "15887710661948", "hashtag
": ["Station:LINE"], "PRODUCTID": "KUKA1FK708_P"}
Receive Message from server, topic : MessageTypePublished, content length : 131, content : {"value": {"Output": "1"}, "key": "Output", "timestamp": "1588771066607", "hashtag":
["Station:LINE"], "PRODUCTID": "KUKA1FK708_P"}
Receive Message from server, topic : MessageTypePublished, content length : 109, content : {"value": "", "key": "CMD_RESETER", "timestamp": "1588771066623", "hashtag": [], "PR
ODUCTID": "KUKA1FK708_P"}
Receive Message from server, topic : MessageTypePublished, content length : 109, content : {"value": "", "key": "CMD_RESETER", "timestamp": "1588771066629", "hashtag": [], "PR
ODUCTID": "KUKA1FK708_P"}
Receive Message from server, topic : MessageTypePublished, content length : 131, content : {"value": {"Output": "1"}, "key": "Output", "timestamp": "1588771067286", "hashtag":
["Station:LINE"], "PRODUCTID": "KUKA1FK708_P"}
Receive Message from server, topic : MessageTypePublished, content length : 158, content : {"value": {"FPY": "100.0", "Total": "1", "Defect": "0"}, "key": "FPY", "timestamp":
"1588771186346", "hashtag": ["Station:LINE"], "PRODUCTID": "KUKA1FK708_P"}
Receive Message from server, topic : MessageTypePublished, content length : 158, content : {"value": {"FPY": "100.0", "Total": "2", "Defect": "0"}, "key": "FPY", "timestamp":
"1588771186546", "hashtag": ["Station:LINE"], "PRODUCTID": "KUKA1FK708_P"}
Receive Message from server, topic : MessageTypePublished, content length : 158, content : {"value": {"FPY": "100.0", "Total": "3", "Defect": "0"}, "key": "FPY", "timestamp":
"1588771202835", "hashtag": ["Station:LINE"], "PRODUCTID": "KUKA1FK708_P"}
Receive Message from server, topic : MessageTypePublished, content length : 134, content : {"value": {"Status": "Idle"}, "key": "Status", "timestamp": "1588771246379", "hashta
g": ["Station:LINE"], "PRODUCTID": "KUKA1FK708_P"}
Receive Message from server, topic : MessageTypePublished, content length : 140, content : {"value": {"StopCode": "空闲"}, "key": "StopCode", "timestamp": "1588771247568", "ha
shtag": ["Station:LINE"], "PRODUCTID": "KUKA1FK708_P"}
Receive Message from server, topic : MessageTypePublished, content length : 131, content : {"value": {"Output": "2"}, "key": "Output", "timestamp": "1588771370887", "hashtag":
["Station:LINE"], "PRODUCTID": "KUKA1FK708_P"}
Receive Message from server, topic : MessageTypePublished, content length : 131, content : {"value": {"Output": "3"}, "key": "Output", "timestamp": "1588771372559", "hashtag":
["Station:LINE"], "PRODUCTID": "KUKA1FK708_P"}

```

图 18. IOTBOX 发布到 Agent 的数据

上图中的一些关键字，解释如下：

标记 1，是 IOTBOX 往 MindsSphere 发布的数据，产品 ID，其余标记依次是状态，产量等。

3.4 排错和优化

SCL 程序下载报错问题

如图 15 的 Fail Record FIFO 数据库，工艺要求每条新收到的数据记录都要跟库内 100 条记录进行产品序列号的比较，序列号的数据格式为 Array of Char，在编写 if 语句条件判断新数据的序列号和旧序列号两字符数组是否相等程序时，遇到编译不报错，但下载提示报错问题。经过多次的测试，才把问题定位出来，SCL 无法进行两个字符数组的直接比较。之后，通过修改数据记录的数据类型为 String[35]+DTL 来存储一条数据记录的序列号和时间戳，解决此问题。

APP 设备状态栏出现实时显示设备停机状态，但不能同时显示停机原因，且对应停机类别的时间不会累加问题。

由于项目前期定义了如图 9 的 SIMICAS PLC（服务器端）DB 块作为与 IOTBOX S7 通信的数据接口。为何 PLC 程序一个周期内输出了 Line Status（STOP）和 Stop Code，但是 APP 却接收不到呢？经后期与产品开发沟通，APP 必须要先查询到 STOP 状态，之后再接收到停机原因 Stop Code（Packing 故障、MFT 故障等），才能为停机类别的时间进行统计和显示。而 PLC 程序的 Line Status 是根据 Stop Code 计算出来的，两者逻辑上相悖了。另外，由于习惯卸、写实时逻辑控制程序，对周期时许比较敏感能在一个周期内完成的任务绝不分两个周期做。惯性思维一个 DB 块内的变量在输出时谁先谁后是不确定，因此在调试过程中就进入了

死胡同。最后发现，这样的数采程序并没有严格的实时性要求，最终按 APP 端的要求编写出 Line Status 变化时立即输出同时延时一秒再输出 Stop Code 功能块，解决此问题。

APP 无法收到产线的数据

这种情况在以往调试中并非少见，其可能原因也非常多，此时应结合图 16（SIMICAS APP 上数数据流）至下而上的判断各数据节点的状态。举一例子，产线模型正常发布成功后，IOTBOX 以客户端角色通过 S7 通信周期（最快 500MS）读取 PLC DB 块接口数据，那么只有当 PLC 数据变化时 IOTBOX 才能获取一条新的数据记录。当 PLC 数据更新后，IOTBOX 是否收到数据。首先，打开调试工具 MobaXterm，通过 SSH 协议 22 端口进入 IOTBOX Linux 界面。然后，输入 `cd exec`，再输入 `./mb_sub -t opcua/data`。如果 IOTBOX 未收到数据，则可能需要执行 `Systemctl restart s7adapter`，也有可能未勾选 PLC-连接机制-允许远程对象的 PUT/GET 访问，或者重启下 IOTBOX 电源重启所有进程；如果 IOTBOX 收到数据，再查看 Agent 是否收到数据，此时输入 `cd exec`，再输入 `./mb_sub -t MessageTypePublished` 进行查询。

产线模型发布失败

产线模型经 SIMICAS Tool 组态完成后，需要向 IOTBOX 和 MindSphere 发布。此过程归结为三个步骤：1、IOTBOX 是否成功加载发布的配置；2、IOTBOX 上的 Agent 是否收到产线配置并且收到来自 MindSphere 的 Token5；3、云端 SIMICAS APP 是否成功收到产线配置。在发布过程中，每一步都可能遇到失败的情况，下面以第二步失败举例。根据上述过程，可以从以下几个方面着手判断。首先，检查 SIMICAS Tool 服务器与 IOTBOX 网络连接情况；其次，检查 SIMICAS Tool 服务器与 MindSphere 的网络连接状况，并校验 MindSphere 连接参数是否通过。如果以上都确认无误，可以通过 `curl -x http://代理服务器:8080`，例如 `http://gateway.cn1.mindsphere.in.cn` 确认 IOTBOX 确认 IOTBOX 能否连接到 MindSphere。如上述条件均满足的情况下还是未能发布，还得修正 SIMICAS Tool 服务器 Mongo DB 数据库设置合适的 APP 版本和 Token 的过期时间，有时候还得通过命令 `journalctl -f -u updater` 查看 IOTBOX 盒子的 Agent 服务，如果查询到失败需要重启所有服务 `stop/start-all.sh`（也可单独重启 Agent 服务）才能最终发布成功。

生产转换 changeover 导致产量不准

实施完成后，客户经过几天的验证发现 APP 记录的 Output 产量 KPI 不准，按照 SIMICAS 的逻辑以每一次 Product ID 变化作为生产转换的标志，而每次 Product ID 和 Output 也都被准确记录，那为何出现产量统计不准的情况。经过长时间观察验证，截取 Changeover 的过程，得出如下结论：1. 一次 Changeover 为 1S（标记 1 和 2）；2. 转换过程中的一次 Output

被丢弃（标记 3）；3. 由于 PLC 此前在一个周期内更新的 Product ID 和 Output。总结上述原因得出，每次转换都会丢掉一次产量的统计，在换型频繁的场景下，导致产量统计值的失准。那么对症下药，通过修改 PLC 程序使得得到新的 Product ID 后延时 2S 更新 Output 值，就规避了此问题，如图 19。

```
Receive Message from server, topic : MessageTypePublished, content length : 158, content : {"value": {"FPY": "100.0", "Total": "9", "Defect": "0"}, "key": "FPY", "timestamp": "1589079240491", "hashtag": ["Station:LINE"], "PRODUCTID": "MTS1FK703G_P"}
Receive Message from server, topic : MessageTypePublished, content length : 159, content : {"value": {"FPY": "100.0", "Total": "10", "Defect": "0"}, "key": "FPY", "timestamp": "1589079285270", "hashtag": ["Station:LINE"], "PRODUCTID": "MTS1FK703G_P"}
Receive Message from server, topic : MessageTypePublished, content length : 159, content : {"value": {"FPY": "100.0", "Total": "11", "Defect": "0"}, "key": "FPY", "timestamp": "1589079307849", "hashtag": ["Station:LINE"], "PRODUCTID": "MTS1FK703G_P"}
Receive Message from server, topic : MessageTypePublished, content length : 131, content : {"value": {"Output": "4"}, "key": "Output", "timestamp": "1589079328450", "hashtag": ["Station:LINE"], "PRODUCTID": "MTS1FK703G_P"}
Receive Message from server, topic : MessageTypePublished, content length : 131, content : {"value": {"Output": "5"}, "key": "Output", "timestamp": "1589079329562", "hashtag": ["Station:LINE"], "PRODUCTID": "MTS1FK703G_P"}
Receive Message from server, topic : MessageTypePublished, content length : 159, content : {"value": {"FPY": "100.0", "Total": "12", "Defect": "0"}, "key": "FPY", "timestamp": "1589079352136", "hashtag": ["Station:LINE"], "PRODUCTID": "MTS1FK703G_P"}
Receive Message from server, topic : MessageTypePublished, content length : 131, content : {"value": {"Output": "6"}, "key": "Output", "timestamp": "1589079378123", "hashtag": ["Station:LINE"], "PRODUCTID": "MTS1FK703G_P"}
Receive Message from server, topic : MessageTypePublished, content length : 148, content : {"value": {"ProductID": "MTS1FK708G_P", "key": "ProductID", "timestamp": "1589079458607", "hashtag": ["Station:LINE"], "PRODUCTID": "MTS1FK708G_P"}, {"value": {"Output": "7"}, "key": "Output", "timestamp": "1589079458648", "hashtag": ["Station:LINE"], "PRODUCTID": "MTS1FK708G_P"}
Receive Message from server, topic : MessageTypePublished, content length : 128, content : {"value": {"category": "class", "action": "reset", "target": "Output"}, "key": "trigger_resetter", "timestamp": "1589079459044", "hashtag": ["Station:LINE"], "PRODUCTID": "MTS1FK708G_P"}
Receive Message from server, topic : MessageTypePublished, content length : 125, content : {"value": {"category": "class", "action": "reset", "target": "FPY"}, "key": "trigger_resetter", "timestamp": "1589079459046", "hashtag": ["Station:LINE"], "PRODUCTID": "MTS1FK708G_P"}
Receive Message from server, topic : MessageTypePublished, content length : 125, content : {"value": {"category": "class", "action": "reset", "target": "WIP"}, "key": "trigger_resetter", "timestamp": "1589079459050", "hashtag": ["Station:LINE"], "PRODUCTID": "MTS1FK708G_P"}
Receive Message from server, topic : MessageTypePublished, content length : 171, content : {"value": {"ProductID": "MTS1FK708G_P", "ChangeOver": "true", "key": "ChangeOver", "timestamp": "1589079459067", "hashtag": ["Station:LINE"], "PRODUCTID": "MTS1FK708G_P"}
Receive Message from server, topic : MessageTypePublished, content length : 109, content : {"value": "", "key": "CMD_RESETER", "timestamp": "1589079459396", "hashtag": ["Station:LINE"], "PRODUCTID": "MTS1FK708G_P"}
Receive Message from server, topic : MessageTypePublished, content length : 109, content : {"value": "", "key": "CMD_RESETER", "timestamp": "1589079459396", "hashtag": ["Station:LINE"], "PRODUCTID": "MTS1FK708G_P"}
Receive Message from server, topic : MessageTypePublished, content length : 109, content : {"value": "", "key": "CMD_RESETER", "timestamp": "1589079459416", "hashtag": ["Station:LINE"], "PRODUCTID": "MTS1FK708G_P"}
Receive Message from server, topic : MessageTypePublished, content length : 134, content : {"value": {"Status": "Idle", "key": "Status", "timestamp": "1589079557210", "hashtag": ["Station:LINE"], "PRODUCTID": "MTS1FK708G_P"}
Receive Message from server, topic : MessageTypePublished, content length : 140, content : {"value": {"StopCode": "空闲", "key": "StopCode", "timestamp": "1589079558368", "hashtag": ["Station:LINE"], "PRODUCTID": "MTS1FK708G_P"}
```

图 19. 产品 ID 转换导致产量计算不准

四、运行效果

此项目于 2020 年 5 月份正式投入使用，对于 KPI 的统计和 APP 的使用客户都有良好的评价。由于本项目的运行界面当时没有截图，但是前端显示风格类似，所以只好展示 Demo 项目中 PC 端 SIMICAS APP 的产线界面，具体如下图。

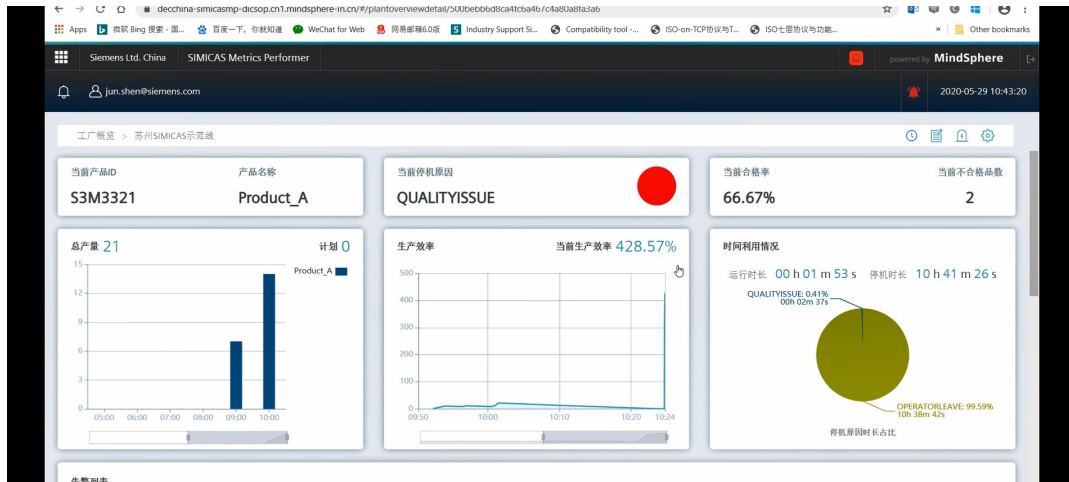


图 20. Demo1 产线界面

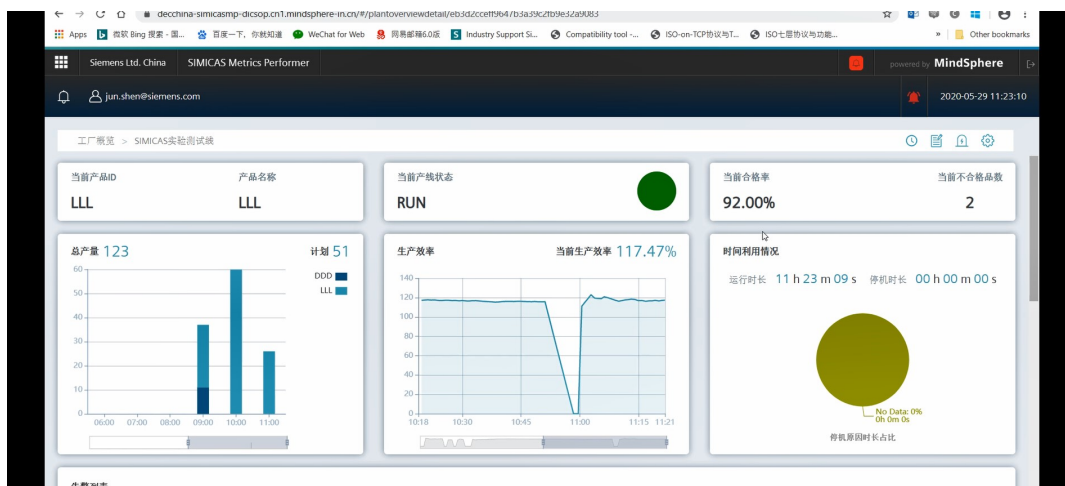


图 21. Demo2 产线界面

五、应用体会

作为项目实施者，我认为 SIMICAS 的主要特点和优劣势如下：

对不同离散行业产线建模时使用同一套建模工具，效率比较高，不需要高级语言编程；KPI 的自定义编辑功能，满足不同客户的口味和需求；多个客户的项目运行于同一个服务器之上，降低一次性投入；管理者可以在 PC、平板和手机等不同设备上使用该系统，并可以在微信小程序和 APP 市场上找到，使用门槛低。

相对来说，SIMICAS 的前端画面较固定和死板，同时也存在和其它云应用面临的年费问题等痛点。