|  |  |
| --- | --- |
| A picture containing graphical user interface  Description automatically generated |  |

**Detailed Design Document**

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Number** |  | : | N/A |
| **File Name** |  | : | DDS Redundancy System V1.3.docx |
| **Version Number** |  | : | 1.3 |
| **Status** |  | : | Draft |
| **Release Date** |  | : |  |
| **Originators** |  | : | Orbis MES – Dairy.com |
| **Authors** |  | : | Sun-Chil Choi |
| **Controlled Document** |  | : | Yes |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Author / Responsible** | **Revision** | **Comments** |
| **2022-09-28** | Sun-Chil Choi | 1.3 | Initial |
| **2022-11-07** | Sun-Chil Choi | 1.3 |  |

**Reference Documents**

|  |  |  |
| --- | --- | --- |
| **Date** | **Title** | **Revision** |
|  |  |  |
|  |  |  |

**TABLEOF CONTENTS**

Contents

[1. Introduction 4](#_Toc118889330)

[2. Redundancy System Overview 5](#_Toc118889331)

[3. Redundancy System Architecture 6](#_Toc118889332)

[3.1. Failover Tag in PLC for Redundancy 6](#_Toc118889333)

[3.2. Aspentech IP21 Redundancy 7](#_Toc118889334)

[3.3. E-Batch Interface Redundancy 7](#_Toc118889335)

[3.3.1. Writing cycle information to E-batch 7](#_Toc118889336)

[3.3.2. Automatic Configure Timers in PLC using E-batch 8](#_Toc118889337)

[3.4. Aspentech IP21 Steam Management 9](#_Toc118889338)

[3.5. OPS Batch System Integration Redundancy 9](#_Toc118889339)

[3.6. OPS Batch System Database Redundancy 9](#_Toc118889340)

[3.7. OPS Web service and UI Redundancy 9](#_Toc118889341)

[3.8. SSRS Report Redundancy 11](#_Toc118889342)

[3.8.1. Electronic Signature 12](#_Toc118889343)

[3.9. IP.21 and E-Batch Secondary Workflow 16](#_Toc118889344)

[4. Wonderware Intouch HMI 16](#_Toc118889345)

[4.1. User Interface in Intouch 16](#_Toc118889346)

[4.2. Kepware 17](#_Toc118889347)

[4.3. Wonderware FS Gateway 19](#_Toc118889348)

[4.4. Access name 20](#_Toc118889349)

[4.5. Tagname Dictionary 20](#_Toc118889350)

[4.6. Intouch Graphic 22](#_Toc118889351)

[4.6.1. Symbol AP\_MainDetail 24](#_Toc118889352)

[4.7. Intouch QuickFunctions 25](#_Toc118889353)

[4.7.1. DISABLE\_COMMS and ENABLE\_COMMS Tag 26](#_Toc118889354)

[5. Appendix 27](#_Toc118889355)

# Introduction

This document outlines the redundancy system architecture to improve the reliability and availability of Baxter MES system.

The Sterilizer Management System has been in successful operation at the Castlebar Plant since 2021. The system will be upgraded to ensure that, in the event of the primary server(s) not being available, the sterilization process can continue.

Currently, IECAAPP014 is the primary SMS historian server. IECAAPP017 is the secondary historian server, which contains critical tags. If the existing server, IECAAPP014 was unavailable, the cycle data would be viewed from the secondary server, IECAAPP017.

If IECAAPP014 was unavailable, it would not be possible to set up new sterilizer cycles, by sending e-batch information to the Sterilizer PLC's via the SMS or to read Cycle start times, Cycle OK / NOT Ok or Rules/Alerts broken and send to E-batch system.

Three new SMS servers will be introduced as part of this upgrade:

* IECAAPPSMS001 (Aspentech IP21 Secondary)
* IECADBSSMS001 (Microsoft SQL Server Secondary)
* IECAWEBSMS001 (Web Server Secondary)

IECAAPP017 will be replaced by IECAAPPSMS001, which will have the same functionality as IECAAPP014. Following the upgrade IECAAPP017 no longer be in service.

# Redundancy System Overview

The redundant system will consist of a **secondary** Aspentech IP21 server, Microsoft SQL server and Web server.

Diagram

Description automatically generated

Figure 1: Previous SMS system with SSRS Reporting

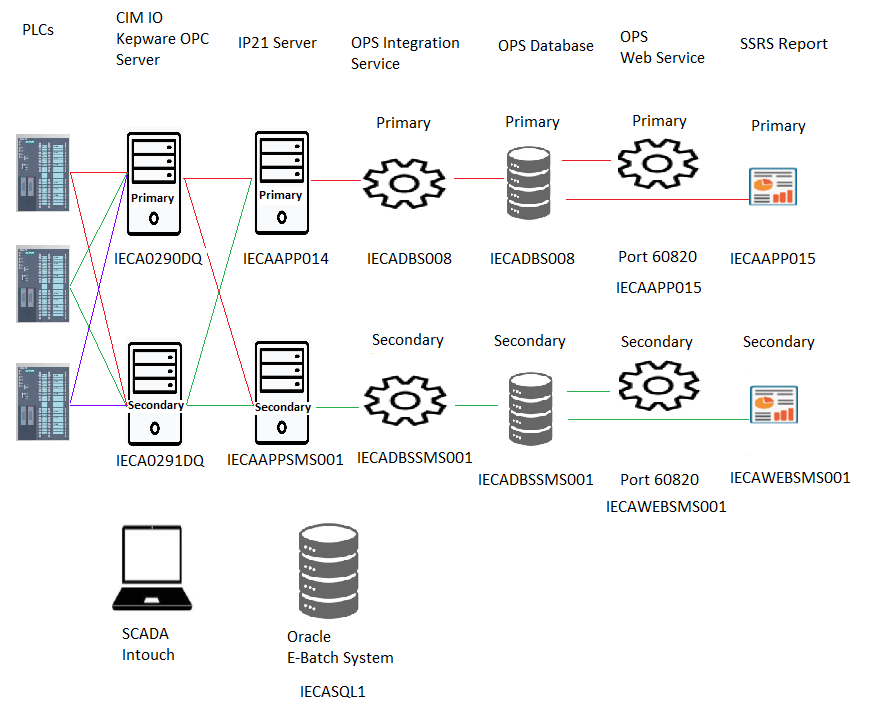


Figure 2: NEW SMS Redundancy system with SSRS Reporting

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **IP21 Historian Server (VM)** | **SQL Database Server (VM)** | **Application / Web server (VM)**  **SSRS Server** | **CIMIO / Kepware Server** | **Oracle  E-Batch** |
| **Primary** | IECAAPP014 | IECADBS008 | IECAAPP015 | IECA0290DQ | IECASQL1 |
| **Secondary** | IECAAPPSMS001 | IECADBSSMS001 | IECAWEBSMS001 | IECA0291DQ |

Table 1: SMS Redundancy System servers

# Redundancy System Architecture

## Failover Tag in PLC for Redundancy

A tag will be set up in the Steam PLC. This tag will be used to indicate when primary IP21 is active or secondary IP21 is active.

* Failover = 0 (Primary is active)
* Failover = 1 (Secondary is active)

This tag will be read by both Primary (IECAAPP014) and Secondary (IECAAPPSMS001) historians and monitored for active (0) and standby (1) states.

## Aspentech IP21 Redundancy

IECAAPP014 (Primary) and IECAAPPSMS001 (Secondary) are running in parallel. Both servers are identical and collect signal (Tags) from Active CIM IO.

## E-Batch Interface Redundancy

### Writing cycle information to E-batch

There is an Oracle 32-bit ODBC connection called PCASXFP set up on IECAAPP014 to connect to E-batch Oracle system. This connection allows Aspentech IP21 to send to E-Batch the following details about the sterilisation cycle: Cycle OK, Cycle Start Time and Cycle Aberrations.

This Oracle ODBC connection will be set up on new historian server IECAAPPSMS001. This will allow the secondary historian IECAAPPSMS001 to transfer the sterilisation details: Cycle OK, Cycle Start Time and Aberrations when it is the active historian.

The following outlines the Oracle 32-bit connection set up on IECAAPP014.

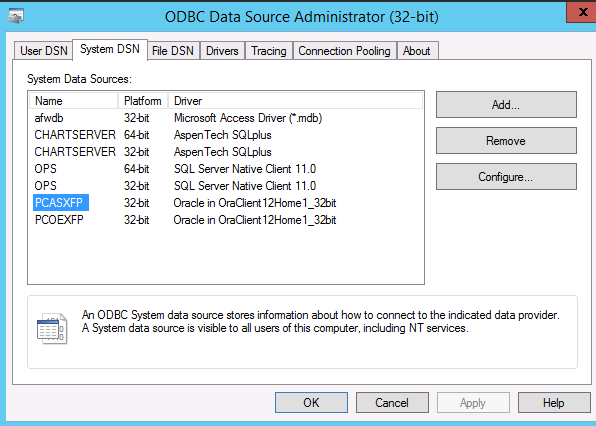


Figure 3: Oracle – E-batch ODBC connection

The Oracle ODBC connection uses the Oracle TNSNAMES.ORA file at this location on IECAAPP014.

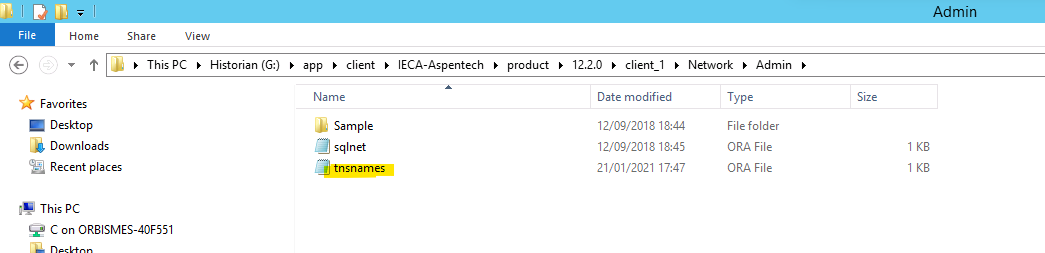


Figure 4: TNSNAMES.ORA File Location

This TNSNAMES.ORA file is updated with the Oracle E-Batch information Host IP address, Port, and service name.

Graphical user interface, text, application, email

Description automatically generated

Figure 5: TNSNAMES.ORA PCASXFP details

The following Compqueries outlined in table below are Aspentech IP21 scripts that interface with the E-Batch system. These scripts connect to E-Batch using ***PCASXFP*** ODBC connection. They send to E-Batch the following details about the sterilisation cycle: Cycle OK, Cycle Start Time and Cycle Aberration

The scripts will be disabled when IP21 server is in Standby mode and will be enabled when the server become Active server. So only the Active IP21 server will communicate with E-Batch System.

* The Primary server **IECAAPP014** will be active when the ***Failover tag value = 0***. The compqueries on IECAAPP014 will check the Failover tag value = 0 and then execute the script. When the Failover tag value = 1, the primary will be in standby mode and the compqueries will be disabled.
* The secondary server **IECAAPPSMS001** will be active when the ***Failover tag value = 1***. The compqueries on IECAAPPSMS001 will check the Failover tag value = 1 and then execute the script. When the Failover tag value = 0, the secondary will be in standby mode and the compqueries will be disabled.

|  |  |  |
| --- | --- | --- |
| **Interface to E-Batch** | **IP21 Queries** | **Ref** |
| Cycle start time | ***EbatchWriteCycSt***  (compquery) | Cycle\_Start\_Timetable |
| Cycle OK Flag | ***EbatchWriteCycOK***  (Compquery) | Cycle OK status |
| Writing Rules/Alerts Broken | ***EbatchWriteAberr***  (Compquery) | Cycle Aberations |

Table 2: Aspentech IP21 Compqueries interface to Ebatch

### Automatic Configure Timers in PLC using E-batch

The SMS system is set up to automatically configure timers in the PLC. This is achieved by the Aspentech IP.21 compquery ***EbatchConfigTime.***  The compquery is executed when the E-Batch steriliser tag VesselLoad is active provided the steriliser cycle has not started. The compquery will write the E-batch timer values to the steriliser PLC.

This compquery will be disabled when IP21 server is in Standby mode and will be enabled when the server become Active server. So only Active IP21 server will communicate with PLC.

* The Primary server **IECAAPP014** will be active when the ***Failover tag value = 0***. The ***EbatchConfigTime*** compquery on IECAAPP014 will check the Failover tag value = 0 and then execute the script. When the Failover tag value = 1, the primary will be in standby mode and the compquery will be disabled.
* The secondary server **IECAAPPSMS001** will be active when the ***Failover tag value = 1***. The ***EbatchConfigTime*** compquery on IECAAPPSMS001 will check the Failover tag value = 1 and then execute the script. When the Failover tag value = 0, the secondary will be in standby mode and the compquery will be disabled.

## Aspentech IP21 Steam Management

A scheduled compquerydef ***SteamManagement*** monitors the current steam availability, the steam requirements of currently running Sterilizers and the steam requirement of the Sterilizer that is currently in position 1 in the queue. If there is enough steam available to complete a cycle for this Sterilizer then the cycle will be started.

The compquery ***SteamManagement*** will be disabled when IP21 server is in Standby mode and will be enabled when the server become Active server. So only Active IP21 server will communicate with PLC.

* The Primary server **IECAAPP014** will be active when the ***Failover tag value = 0***. The ***SteamManagement*** compquery on IECAAPP014 will check the Failover tag value = 0 and then execute the script. When the Failover tag value = 1, the primary will be in standby mode and the compquery will be disabled.
* The Secondary server **IECAAPPSMS001** will be active when the ***Failover tag value = 1***. The ***SteamManagement*** compquery on IECAAPPSMS001 will check the Failover tag value = 1 and then execute the script. When the Failover tag value = 0, the secondary will be in standby mode and the compquery will be disabled.

## OPS Batch System Integration Redundancy

OPS Batch system Integration runs on both Primary and Secondary Database server and write OPS Batch to Primary and Secondary OPS Batch Database respectively.

## OPS Batch System Database Redundancy

Both Primary and Secondary runs in parallel and collect data from Primary and Secondary OPS Integration respectively.

* IECADBS008 (Primary)
* IECADBSSMS001 (Secondary) will be added.

## OPS Web service and UI Redundancy

There are two Web services and UIs as follows.

|  |  |  |
| --- | --- | --- |
|  | Web Service URL | OPS UI URL |
| Primary | http://IECAAPP015:60820/ws | http://IECAAPP015:60820/ui/app |
| Secondary | http://IECAWEBSMS001:60820/ws | http://IECAWEBSMS001:60820/ui/app |

Graphical user interface

Description automatically generated with medium confidence

Figure 6: Primary IIS Setting

A picture containing graphical user interface

Description automatically generated

Figure 7: Secondary IIS Setting

## SSRS Report Redundancy

SSRS report is currently on IECAAPP015 will be moved to IECADBS008 and IECADBSSMS001

Server and there will be two identical SSRS report on both IECADBS008 and IECADBSSMS001

SSRS on IECADBS008 will be pointing to primary OPS Database

SSRS on IECADBSSMS001will be pointing to secondary OPS Database

Graphical user interface, application, Teams

Description automatically generated

Figure 8: Primary SSRS Home Screen

Graphical user interface, application, Teams

Description automatically generated

Figure 9: Secondary SSRS Home Screen

Graphical user interface, text, application, email

Description automatically generated

Figure 10: SSRS Reports

### Electronic Signature

The e-sign page allows you to sign a chart with the following constraints:

* Only QACh and Supervisor users can fill missing data or sign
* If a user filled missing data, the same user cannot sign the chart
* Within 4 hours from the cycle starting time is impossible fill data or sign the chart
* Only QACh users can sign the chart if data is from tiway, but supervisors can fill data
* Only QACh users can sign a chart if the cycle is not OK
* Users who were logged “from (cycle\_start\_time - 30 min) to cycle\_end\_time” cannot sign the chart
* If the cycle has missing data, the user can only fill those data and confirm, the signing operation must be performed by another user selecting the cycle from the report page
* Cycles created by means of ‘CreateCycle’ page have batch id starting with M
* Before permanently storing data and/or sign check into the DB the user must re-input the password in the confirmation box
* The same signoff functionality/code is used for signoff of E-Batch and Non-Ebatch cycles

There is a special patch of the WS that allows to retrieve the group a user belongs. This is crucial to recognize Supervisor and QACh users. The patch is visible int he Accounts/CurrentUser API where the field GroupsName contains a list of groups for the user authenticated.

The e-sign page performs a series of queries in this order:

* /AttributeInstances/Batch where it retrieves the cycle data
* /AttributeInstances/Query only if the chart is ready to be signed (no missing attributes) to check if the current user has filled some of those
* /Accounts/CurrentUser to retrieve the current user
* /Sessions (intouch virtual app) only for supervisors to verify if the supervisor was logged into intouch during the cycle

For this last bullet, we created a small WS app where we read intouch WWALMDB database, more specifically the view V\_EventHistory searching for the name of the user and a time range.

Graphical user interface, application

Description automatically generated

Figure 11: Electronic signature

Graphical user interface, text, application

Description automatically generated

Figure 11: Cycle report Detail

Sign Chart button will open http:// IECAAPP015:60820/ui/reports/SignChart/ElectronicSignature.html

Graphical user interface, text, application, chat or text message

Description automatically generated

Figure 11: Electronic signature

Intouch web service that connect to Intouch WWALMDB

and electronic signature UI will on both IECAAPP015 and IECAWEBSMS001 server

Graphical user interface

Description automatically generated with medium confidence

Figure 12: Web Service for Intouch DB and Electronic Signature UI

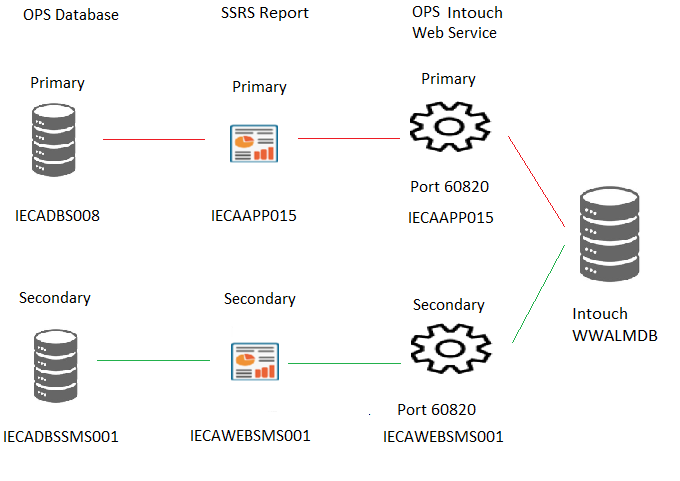


Figure 12: SSRS Redundancy architecture with Electronic Signature

## IP.21 and E-Batch Secondary Workflow

The workflow diagram below shows how both IP21 switched over to the secondary server.

Diagram

Description automatically generated

Figure 13: IP21 to E-Batch Redundancy

# Wonderware Intouch HMI

The Wonderware Intouch HMI application is a graphical interface that allows the user to monitor and administrate the manufacturing sterilisation processes.

## User Interface in Intouch

The Wonderware Intouch HMI will provide user interface to switch over between primary and secondary system. A new Intouch graphic called MES\_System\_Redundancy will be set up with buttons for ***Primary*** and ***Secondary***.

* When the user clicks the ***Secondary*** button – this will set the ***Failover tag = 1*** in the Steam PLC. The ***Secondary*** button will be highlighted in red. The Secondary Historian (***IECAAPPSMS001)*** is now active, the Primary Historian (**IECAAPP014**) is standby.
* When the user clicks the ***Primary*** button – this will set the ***Failover tag = 0*** in the Steam PLC. The ***Primary*** button will be highlighted in red. The Primary Historian (**IECAAPP014*)*** is now active, the Secondary Historian (***IECAAPPSMS001***) is standby.

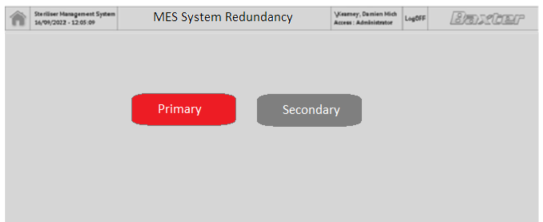


Figure 14: New Intouch user interface mimic

The Intouch system status graphic will be updated to display the Status of the Primary and Secondary Historian and highlight which historian is active and standby.

The active and standby will be decided by the Failover tag.

* Primary Active **Failover tag = 0**
* Secondary Active **Failover tag = 1**

A picture containing timeline

Description automatically generated

Figure 15: Modified System Status screen in Intouch with Status of Primary and Secondary Historians

The following sections will outline the modifications to allow for Intouch to display tag from Primary and Secondary historians. In summary

* When Primary historian is active (failover tag = 0) then Intouch will display IP21 tags from the Kepware ***IP21\_Ster*** OPC UA Channel.
* When Secondary historian is active (failover tag = 1) then Intouch will display IP21 tags from the Kepware ***IP21\_Ster\_Secondary*** OPC UA Channel.

## Kepware

There will be two channels (Primary historian and Secondary historian) set up using OPC UA Client Driver on both Primary (IECA0290DQ) and Secondary (IECA0291DQ) Kepware servers. Both channels (Primary historian and Secondary historian) will run in parallel.

***IP21\_Ster*** is the existing channel connected to Primary ***IECAAPP014*** historian. The channel connects to specific IP21 historian tags such as vessel stage, steam calculations and communication information. These tags are currently displayed on Intouch.

A new secondary channel called ***IP21\_Ster\_Secondary*** will be configured on both Primary (IECA0290DQ) and Secondary (IECA0291DQ) Kepware servers. The secondary channel will connect to specific IP21 historian tags such as vessel stage, steam calculations and communication information on the Secondary historian ***IECAAPPSMS001***.

Graphical user interface, text, application

Description automatically generated

Figure 16: Kepware Server Configuration with primary and secondary channels.

Each channel will be set up with the following OPC UA Endpoints URL. These endpoints allow the channel to connect to the Primary (***IECAAPP014***) and Secondary (***IECAAPPSMS001***) Historians.

|  |  |
| --- | --- |
| **OPC UA Channel** | **OPC UA Endpoint URL** |
| IP21\_Ster | opc.tcp:// **IECAAPP014**:63500/InfoPlus21/OpcUa/Server |
| IP21\_Ster\_Secondary | opc.tcp:// **IECAAPPSMS001**:63500/InfoPlus21/OpcUa/Server |

Table 3: OPC UA Endpoints

The following displays the properties of the IP21\_Ster and IP21\_Ster\_Secondary Endpoints URL configuration in Kepware software.

Graphical user interface, text, application

Description automatically generated

Figure 17: IP21\_Ster Endpoint URL

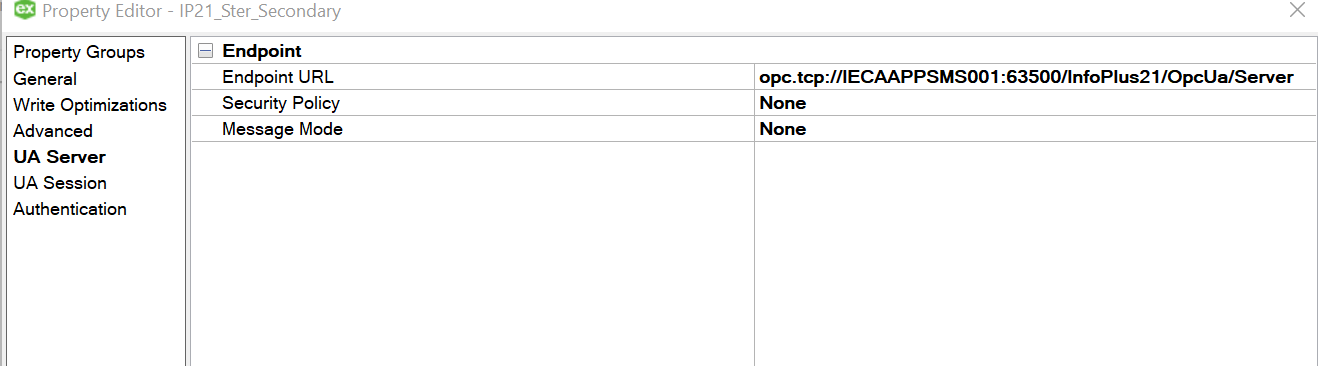


Figure 18: IP21\_Ster\_Secondary Endpoint URL

## Wonderware FS Gateway

Wonderware Intouch HMI communicates to Kepware using Wonderware FSGateway. Intouch will be connected to both channels ***IP21\_Ster*** and ***IP21\_Ster\_Secondary***.

Intouch scripts will check the Failover PLC tag, when

* failover tag = 0, Intouch will display the tags from IP21\_Ster Channel,
* failover tag = 1, Intouch will displays the tags from IP21\_Ster\_Seconday Channel.

The Wonderware FSGateway will be updated on Primary (IECA0290DQ) and Secondary (IECA0291DQ) Kepware servers with the new ***IP21\_Ster\_Secondary*** OPC UA. The connection will be configured in the same way as the existing ***IP21\_Ster*** connection as seen in figure below.

Graphical user interface, text, application

Description automatically generated

Figure 19: Existing FSGateway Configuration with IP\_Ster Connection

## Access name

Intouch will be updated with the following new Access Names:

|  |  |
| --- | --- |
| **Access Name** | **Topic Name** |
| IP\_STER\_SECA | IP21\_STER\_SECONDARY\_IP21\_RAW\_IP\_AnalogDef |
| IP\_STER\_SECD | IP21\_STER\_SECONDARY\_IP21\_RAW\_IP\_DiscreteDef |
| IP\_STER\_SECT | IP21\_STER\_SECONDARY\_IP21\_RAW\_IP\_TextDef |

Each access name will have new FSGateway topic name connections to Secondary OPC UA ***IP21\_Ster\_Secondary***. The following figure displays current access name IP\_STERA.

Graphical user interface, text, application

Description automatically generated

Figure 20: Current Access Name for IP\_STERA in Intouch

## Tagname Dictionary

The Intouch Tagname Dictionary will be updated with the following tags from ***IP21\_Ster***\_***Secondary*** Channel connected to Secondary Historian ***IECAAPPSMS001.***

The following table outlines the new access names and the new tags.

|  |  |
| --- | --- |
| **New Access Name** | **Tags to be added to tagname dictionary and associated to this access name.** |
| IP\_STER\_SECD | CHARTOnPri, CHARTOnSec,  IP21OnPrim, IP21OnSec,  OPC1FSGATE, OPC1KEPWAR, OPC2FSGATE, OPC2KEPWAR, OPCUA, OPSBATCH,  PLC01STAT, PLC01STAT, PLC02STAT, PLC02STAT, PLC03STAT, PLC03STAT, PLC04STAT, PLC04STAT, PLC05STAT, PLC05STAT, PLC06STAT, PLC06STAT,  PLC07STAT, PLC07STAT, PLC08STAT, PLC08STAT, PLC09STAT, PLC09STAT, PLC10STAT, PLC10STAT, PLC11STAT, PLC11STAT, PLC12STAT, PLC12STAT,  PLC13STAT, PLC13STAT, PLC14STAT, PLC14STAT, PLC15STAT, PLC15STAT, PLC16STAT, PLC16STAT, PLC17STAT, PLC17STAT, PLC18STAT, PLC18STAT,  PLC19STAT, PLC19STAT, PLC20STAT, PLC20STAT, PLC21STAT, PLC21STAT, PLC22STAT, PLC22STAT, PLC23STAT, PLC23STAT, PLC24STAT, PLC24STAT,  PLC25STAT, PLC25STAT, PLC26STAT, PLC26STAT, PLC27STAT, PLC27STAT, PLC40STAT, PLC40STAT, PLC41STAT, PLC41STAT,  TER1ALM, TER2ALM, CHARTSERV, IP21BadTag, IP21GdTag, IP21LatTag, IP21SERV, IP21TotTag, MESDEVSERV, MSSQLSERV,  OPC1SERV, OPC2SERV, TER1SERV, TER2SERV, WEBSERV, WONDEVSERV  V01\_MainMode, V01\_SterIn2, V01\_Stage, V01\_EBatchEnab, V01\_QueuePosit, V01\_Trial,  V02\_MainMode, V02\_SterIn2, V02\_Stage, V02\_EBatchEnab, V02\_QueuePosit, V02\_Trial,  V03\_MainMode, V03\_SterIn2, V03\_Stage, V03\_EBatchEnab, V03\_QueuePosit, V03\_Trial,  V04\_MainMode, V04\_SterIn2, V04\_Stage, V04\_EBatchEnab, V04\_QueuePosit, V04\_Trial,  V05\_MainMode, V05\_SterIn2, V05\_Stage, V05\_EBatchEnab, V05\_QueuePosit, V05\_Trial,  V06\_MainMode, V06\_SterIn2, V06\_Stage, V06\_EBatchEnab, V06\_QueuePosit, V06\_Trial,  V07\_MainMode, V07\_SterIn2, V07\_Stage, V07\_EBatchEnab, V07\_QueuePosit, V07\_Trial,  V08\_MainMode, V08\_SterIn2, V08\_Stage, V08\_EBatchEnab, V08\_QueuePosit, V08\_Trial,  V09\_MainMode, V09\_SterIn2, V09\_Stage, V09\_EBatchEnab, V09\_QueuePosit, V09\_Trial,  V10\_MainMode, V10\_SterIn2, V10\_Stage, V10\_EBatchEnab, V10\_QueuePosit, V10\_Trial,  V11\_MainMode, V11\_SterIn2, V11\_Stage, V11\_EBatchEnab, V11\_QueuePosit, V11\_Trial,  V12\_MainMode, V12\_SterIn2, V12\_Stage, V12\_EBatchEnab, V12\_QueuePosit, V12\_Trial,  V13\_MainMode, V13\_SterIn48, V13\_Stage, V13\_EBatchEnab, V13\_QueuePosit, V13\_Trial,  V14\_MainMode, V14\_SterIn48, V14\_Stage, V14\_EBatchEnab, V14\_QueuePosit, V14\_Trial,  V15\_MainMode, V15\_SterIn48, V15\_Stage, V15\_EBatchEnab, V15\_QueuePosit, V15\_Trial,  V16\_MainMode, V16\_SterIn48, V16\_Stage, V16\_EBatchEnab, V16\_QueuePosit, V16\_Trial,  V17\_MainMode, V17\_SterIn48, V17\_Stage, V17\_EBatchEnab, V17\_QueuePosit, V17\_Trial,  V18\_MainMode, V18\_SterIn48, V18\_Stage, V18\_EBatchEnab, V18\_QueuePosit, V18\_Trial,  V19\_MainMode, V19\_SterIn48, V19\_Stage, V19\_EBatchEnab, V19\_QueuePosit, V19\_Trial,  V20\_MainMode, V20\_SterIn48, V20\_Stage, V20\_EBatchEnab, V20\_QueuePosit, V20\_Trial,  V21\_MainMode, V21\_SterIn48, V21\_Stage, V21\_EBatchEnab, V21\_QueuePosit, V21\_Trial,  V22\_MainMode, V22\_SterIn48, V22\_Stage, V22\_EBatchEnab, V22\_QueuePosit, V22\_Trial,  V23\_MainMode, V23\_SterIn48, V23\_Stage, V23\_EBatchEnab, V23\_QueuePosit, V23\_Trial,  V24\_MainMode, V24\_SterIn48, V24\_Stage, V24\_EBatchEnab, V24\_QueuePosit, V24\_Trial,  V25\_MainMode, V25\_SterIn48, V25\_Stage, V25\_EBatchEnab, V25\_QueuePosit, V25\_Trial,  V26\_SterIn2, V26\_Stage,  V27\_MainMode, V27\_SterIn48, V27\_Stage, V27\_EBatchEnab, V27\_QueuePosit, V27\_Trial,  V40\_MainMode, V40\_SterIn2, V40\_Stage, V40\_EBatchEnab, V40\_QueuePosit, V40\_Trial,  V41\_MainMode, V41\_SterIn2, V41\_Stage, V41\_EBatchEnab, V41\_QueuePosit, V41\_Trial |
| IP\_STER\_SECA | Boil2Cap, Boil6Cap  Boil7Cap, Boil9Cap  DesigSteam, TotalSteamAvailable  TotalSteamCapacity, 2TruckExposureSteam  2TruckHeatUpSteam, 4TruckExposureSteam  4TruckHeatUpSteam, 8TruckExposureSteam  8TruckHeatUpSteam, TotalSteamInUse  Boil6Cap, Boil7Cap, Boil9Cap |
| IP\_STER\_SECT | PLCTIWAY,  V01\_CycStTime, V02\_CycStTime, V03\_CycStTime, V04\_CycStTime,  V05\_CycStTime, V06\_CycStTime, V07\_CycStTime, V08\_CycStTime,  V09\_CycStTime, V10\_CycStTime, V11\_CycStTime, V12\_CycStTime, V13\_CycStTime, V14\_CycStTime, V15\_CycStTime, V16\_CycStTime, V17\_CycStTime, V18\_CycStTime, V19\_CycStTime, V20\_CycStTime, V21\_CycStTime, V22\_CycStTime, V23\_CycStTime, V24\_CycStTime, V25\_CycStTime, V27\_CycStTime, V40\_CycStTime, V41\_CycStTime |

## Intouch Graphic

The following Intouch graphics with symbol will be modified for the new **Secondary** **Historian** tags. The symbol will be modified to check the failover tag. When the ***failover tag = 0*** the symbol will display **Primary** **historian** tag. When the ***failover tag = 1*** the symbol will display the **secondary** **historian** tag.

|  |  |  |
| --- | --- | --- |
| **Intouch graphic Name** | **Intouch Symbol used on graphic** | **Following tags on these screens will switch to secondary tags when Failover tag = 1** |
| System\_Status | IP21READSTATUS  TIWAYCHECK  TerminalServerCheck  TiwayButton  IP21CHECK  IP21OPC\_UA\_Connected\_Kepware  OPSBatchCheck  OPCCheckBar | CHARTOnPri, CHARTOnSec,  IP21OnPrim, IP21OnSec,  OPC1FSGATE, OPC1KEPWAR, OPC2FSGATE, OPC2KEPWAR, OPCUA, OPSBATCH,  PLC01STAT, PLC01STAT, PLC02STAT, PLC02STAT, PLC03STAT, PLC03STAT, PLC04STAT, PLC04STAT, PLC05STAT, PLC05STAT, PLC06STAT, PLC06STAT,  PLC07STAT, PLC07STAT, PLC08STAT, PLC08STAT, PLC09STAT, PLC09STAT, PLC10STAT, PLC10STAT, PLC11STAT, PLC11STAT, PLC12STAT, PLC12STAT,  PLC13STAT, PLC13STAT, PLC14STAT, PLC14STAT, PLC15STAT, PLC15STAT, PLC16STAT, PLC16STAT, PLC17STAT, PLC17STAT, PLC18STAT, PLC18STAT,  PLC19STAT, PLC19STAT, PLC20STAT, PLC20STAT, PLC21STAT, PLC21STAT, PLC22STAT, PLC22STAT, PLC23STAT, PLC23STAT, PLC24STAT, PLC24STAT,  PLC25STAT, PLC25STAT, PLC26STAT, PLC26STAT, PLC27STAT, PLC27STAT, PLC40STAT, PLC40STAT, PLC41STAT, PLC41STAT  TER1ALM, TER2ALM  PLCTIWAY,  CHARTSERV, IP21BadTag, IP21GdTag, IP21LatTag, IP21SERV, IP21TotTag, MESDEVSERV, MSSQLSERV,  OPC1SERV, OPC2SERV, TER1SERV, TER2SERV,  WEBSERV, WONDEVSERV, |
| SMS\_Overview | IP21READSTATUS  TIWAYCHECK  AP\_MainDetail  AP\_MainDetail\_Bolt | V01\_Stage, V02\_Stage, V03\_Stage, V04\_Stage, V05\_Stage, V06\_Stage, V07\_Stage, V08\_Stage, V09\_Stage, V10\_Stage, V11\_Stage, V12\_Stage, V13\_Stage, V14\_Stage, V15\_Stage, V16\_Stage, V17\_Stage, V18\_Stage, V19\_Stage, V20\_Stage,  V21\_Stage, V22\_Stage, V23\_Stage, V24\_Stage, V25\_Stage, V26\_Stage, V27\_Stage, V40\_Stage, V41\_Stage  PLCTIWAY  PLC01STAT to PLC41STAT |
| Steam\_MGMT |  | Boil2Cap, Boil6Cap, Boil7Cap, Boil9Cap  DesigSteam, TotalSteamAvailable  TotalSteamCapacity, 2TruckExposureSteam  2TruckHeatUpSteam,4TruckExposureSteam  4TruckHeatUpSteam,8TruckExposureSteam  8TruckHeatUpSteam, TotalSteamInUse |
| 2Place\_SterIN | SterSelection | V01\_SterIn2, V02\_SterIn2, V03\_SterIn2, V04\_SterIn2, V05\_SterIn2, V06\_SterIn2, V07\_SterIn2, V08\_SterIn2, V09\_SterIn2, V10\_SterIn2, V11\_SterIn2, V12\_SterIn2, V40\_SterIn2, V41\_SterIn2 |
| 4\_8\_Place\_SterIN | SterSelection | V13\_SterIn48, V14\_SterIn48, V15\_SterIn48, V16\_SterIn48, V17\_SterIn48, V18\_SterIn48, V19\_SterIn48, V20\_SterIn48, V21\_SterIn48, V22\_SterIn48, V23\_SterIn48, V24\_SterIn48, V25\_SterIn48, V27\_SterIn48, |
| Ster01\_Control | TrialControllerButton  Queue\_Postion  SteriliserBanner | V01\_MainMode, V01\_QueuePosit, V01\_Trial |
| Ster02\_Control | V02\_MainMode, V02\_QueuePosit, V02\_Trial |
| Ster03\_Control | V03\_MainMode, V03\_QueuePosit, V03\_Trial |
| Ster04\_Control | V04\_MainMode, V04\_QueuePosit, V04\_Trial |
| Ster05\_Control | V05\_MainMode, V05\_QueuePosit, V05\_Trial |
| Ster06\_Control | V06\_MainMode, V06\_QueuePosit, V06\_Trial |
| Ster07\_Control | V07\_MainMode, V07\_QueuePosit, V07\_Trial |
| Ster08\_Control | V08\_MainMode, V08\_QueuePosit, V08\_Trial |
| Ster09\_Control | V09\_MainMode, V00\_QueuePosit, V09\_Trial |
| Ster10\_Control | V10\_MainMode, V10\_QueuePosit, V10\_Trial |
| Ster11\_Control | V11\_MainMode, V11\_QueuePosit, V11\_Trial |
| Ster12\_Control | V12\_MainMode, V12\_QueuePosit, V12\_Trial |
| Ster13\_Control | V13\_MainMode, V13\_QueuePosit, V13\_Trial |
| Ster14\_Control | V14\_MainMode, V14\_QueuePosit, V14\_Trial |
| Ster15\_Control | V15\_MainMode, V15\_QueuePosit, V15\_Trial |
| Ster16\_Control | V16\_MainMode, V16\_QueuePosit, V16\_Trial |
| Ster17\_Control | V17\_MainMode, V17\_QueuePosit, V17\_Trial |
| Ster18\_Control | V18\_MainMode, V18\_QueuePosit, V18\_Trial |
| Ster19\_Control | V19\_MainMode, V19\_QueuePosit, V19\_Trial |
| Ster20\_Control | V20\_MainMode, V20\_QueuePosit, V20\_Trial |
| Ster21\_Control | V21\_MainMode, V21\_QueuePosit, V21\_Trial |
| Ster22\_Control | V22\_MainMode, V22\_QueuePosit, V22\_Trial |
| Ster23\_Control | V23\_MainMode, V23\_QueuePosit, V23\_Trial |
| Ster24\_Control | V24\_MainMode, V24\_QueuePosit, V24\_Trial |
| Ster25\_Control | V25\_MainMode, V25\_QueuePosit, V25\_Trial |
| Ster27\_Control | V27\_MainMode, V27\_QueuePosit, V27\_Trial |
| Ster40\_Control | V40\_MainMode, V40\_QueuePosit, V40\_Trial |
| Ster41\_Control | V41\_MainMode, V41\_QueuePosit, V41\_Trial |

### Symbol AP\_MainDetail

This section is to understand the modification that will occur on a symbol to reflect changeover from active to standby system. The following will take place on all symbols outlined in previous section.

The symbol AP\_MainDetail is used on SMS\_Overview for each steriliser to display the steriliser stage and timers.

Graphical user interface, application, Word

Description automatically generated

Figure 21: AP\_MainDetail Symbol on SMS\_Overview

The following will be modified for the Stage section on this symbol: Clicking on the Stage text – the Visibility and String will be modified to check the Failover tag to decide on displaying Primary server stage tag value or Secondary server stage tag value.

When the ***failover tag = 0*** the **primary historian** tags are used in the symbol, when the ***failover tag = 1*** the **secondary** **historian** tags are used in the symbol.

Diagram, table

Description automatically generated

Figure 22: Stage String on AP\_MainDetail symbol

* **String**: CALL ParseStageTagname(ST1\Unit\Stage)

**NEW String:** IF (ST1\Secondary\Failover==0) ST1\Unit\Stage; ELSE ST1\Secondary\Stage

* **Visibility**: ST1\Unit\Stage.QualityStatus == 3

**NEW Visibility**: IF (ST1\Secondary\Failover==0) ST1\Unit\Stage.QualityStatus == 3; ELSE ST1\Secondary\Stage.QualityStatus == 3

## Intouch QuickFunctions

These Intouch QuickFunctions are called on the following graphics. These functions will be modified to check the failover tag.

When the ***failover tag = 0*** the **primary historian** tags are used in the function, when the ***failover tag = 1*** the **secondary** historian tags are used in the function.

|  |  |  |
| --- | --- | --- |
| **QuickFunction** | **Graphic where function is called** | **Reason** |
| CheckPLCReads | System\_Overview System\_Status | Checks each of these tags:  PLC01STAT, PLC01STAT, PLC02STAT, PLC02STAT, PLC03STAT, PLC03STAT, PLC04STAT, PLC04STAT, PLC05STAT, PLC05STAT, PLC06STAT, PLC06STAT,  PLC07STAT, PLC07STAT, PLC08STAT, PLC08STAT, PLC09STAT, PLC09STAT, PLC10STAT, PLC10STAT, PLC11STAT, PLC11STAT, PLC12STAT, PLC12STAT,  PLC13STAT, PLC13STAT, PLC14STAT, PLC14STAT, PLC15STAT, PLC15STAT, PLC16STAT, PLC16STAT, PLC17STAT, PLC17STAT, PLC18STAT, PLC18STAT,  PLC19STAT, PLC19STAT, PLC20STAT, PLC20STAT, PLC21STAT, PLC21STAT, PLC22STAT, PLC22STAT, PLC23STAT, PLC23STAT, PLC24STAT, PLC24STAT,  PLC25STAT, PLC25STAT, PLC26STAT, PLC26STAT, PLC27STAT, PLC27STAT, PLC40STAT, PLC40STAT, PLC41STAT, PLC41STAT |
| CheckTiway | System\_Overview  System\_Status | PLCTIWAY |
| MaintenanceModeScript | Ster01\_Control, Ster02\_Control, Ster03\_Control, Ster04\_Control  Ster05\_Control, Ster06\_Control, Ster07\_Control, Ster08\_Control  Ster09\_Control, Ster10\_Control, Ster11\_Control, Ster12\_Control  Ster13\_Control, Ster14\_Control, Ster15\_Control, Ster16\_Control  Ster17\_Control, Ster18\_Control, Ster19\_Control, Ster20\_Control  Ster21\_Control, Ster22\_Control, Ster23\_Control, Ster24\_Control  Ster25\_Control, Ster27\_Control, Ster40\_Control  Ster41\_Control | V01\_MainMode, V02\_MainMode,  V02\_MainMode, V03\_MainMode,  V04\_MainMode, V05\_MainMode,  V06\_MainMode, V07\_MainMode,  V08\_MainMode, V09\_MainMode,  V10\_MainMode, V11\_MainMode,  V12\_MainMode, V13\_MainMode,  V14\_MainMode, V15\_MainMode,  V16\_MainMode, V17\_MainMode,  V18\_MainMode, V19\_MainMode,  V20\_MainMode, V21\_MainMode,  V22\_MainMode, V23\_MainMode,  V24\_MainMode, V25\_MainMode,  V27\_MainMode  V40\_MainMode, V41\_MainMode, |
| MaintenanceModeResetScript | Same as above | Same as above |

**Maintenance Mode Tags – need to discuss with Damien**

Maintenance mode – Every sterilisers – They can put the valve into maintenance. Tag in IP21

Need this to be moved to the PLC

**Tiway Setup**

Tiway Reset

Baxter need a method to re-initialize communications on the Tiway network.

The Tiway network is setup using.

### DISABLE\_COMMS and ENABLE\_COMMS Tag

IOGetDef requests with 1 tagname and with the tagname equal to ENABLE\_COMMS or DISABLE\_COMMS cause the Tiway interface to enable or disable communications with HIU on startup respectively. On startup of the interface, communications must be enabled using an IOGetDef record with a single Tagname equal to ENABLE\_COMMS.

# Appendix

LOCAL i INTEGER;

--LOCAL num\_reads INTEGER;

LOCAL a TIMESTAMP;

--

--macro active\_server = CIMIOSTATE.ACTIVE\_NODE;

--num\_reads = (select "#\_of\_selections" from select20def

-- where name like "&active\_server".read\_list);

--

--IF "&active\_Server".read\_list like 'IO1\_READS' THEN

-- -- Disable COMMS on TSK\_M\_TI3

-- update TI\_COMMS\_IO2 set io\_data\_processing = 'OFF';

-- update TI\_COMMS\_IO2 set io\_tagname[1] = 'DISABLE\_COMMS';

-- update TI\_COMMS\_IO2 set io\_data\_processing = 'ON';

-- update TI\_COMMS\_IO2 set "io\_activate?" = 'YES';

-- update TI\_COMMS\_IO2 set "io\_activate?" = 'YES';

--

-- -- Enable COMMS on TSK\_M\_TI1

-- update TI\_COMMS\_IO1 set io\_data\_processing = 'OFF';

-- update TI\_COMMS\_IO1 set io\_tagname[1] = 'ENABLE\_COMMS';

-- update TI\_COMMS\_IO1 set io\_data\_processing = 'ON';

-- update TI\_COMMS\_IO1 set "io\_activate?" = 'YES';

-- update TI\_COMMS\_IO1 set "io\_activate?" = 'YES';

--ELSE

-- -- Disable COMMS on TSK\_M\_TI1

-- update TI\_COMMS\_IO1 set io\_data\_processing = 'OFF';

-- update TI\_COMMS\_IO1 set io\_tagname[1] = 'DISABLE\_COMMS';

-- update TI\_COMMS\_IO1 set io\_data\_processing = 'ON';

-- update TI\_COMMS\_IO1 set "io\_activate?" = 'YES';

-- update TI\_COMMS\_IO1 set "io\_activate?" = 'YES';

--

-- -- Enable COMMS on TSK\_M\_TI3

-- update TI\_COMMS\_IO2 set io\_data\_processing = 'OFF';

-- update TI\_COMMS\_IO2 set io\_tagname[1] = 'ENABLE\_COMMS';

-- update TI\_COMMS\_IO2 set io\_data\_processing = 'ON';

-- update TI\_COMMS\_IO2 set "io\_activate?" = 'YES';

-- update TI\_COMMS\_IO2 set "io\_activate?" = 'YES';

--END

--

--

--

--FOR i = 1 to num\_reads DO

-- update iogetdef set io\_record\_processing = 'ON'

-- where name like (select select\_description from select20def

-- where name like "&active\_server".read\_list and occnum = i);

--

-- update iogetdef set "io\_activate?" = 'YES'

-- where name like (select select\_description from select20def

-- where name like "&active\_server".read\_list and occnum = i);

--

-- -- Wait for 0.2 seconds

--

-- "&active\_server".next\_cimio\_act = cast("time".schedule\_time as integer) + 2;

--

-- WHILE cast("time".schedule\_time as integer) <

-- cast("&active\_server".next\_cimio\_act as integer) DO

-- a = "time".schedule\_time;

-- END

--END

FOR i = 1 to 10 DO

UPDATE TI\_COMMS\_IO1 SET "IO\_ACTIVATE?"='YES';

UPDATE TI\_COMMS\_IO1 SET "IO\_ACTIVATE?"='YES';

UPDATE TI\_COMMS\_IO1 SET "IO\_ACTIVATE?"='YES';

UPDATE IOGETDEF SET "IO\_ACTIVATE?"='YES' WHERE NAME LIKE '%IO1';

UPDATE IOGETDEF SET "IO\_ACTIVATE?"='YES' WHERE NAME LIKE '%IO1';

UPDATE IOGETDEF SET "IO\_ACTIVATE?"='YES' WHERE NAME LIKE '%CR';

UPDATE TI\_COMMS\_IO1 SET "IO\_ACTIVATE?"='YES';

UPDATE TI\_COMMS\_IO1 SET "IO\_ACTIVATE?"='YES';

UPDATE TI\_COMMS\_IO1 SET "IO\_ACTIVATE?"='YES';

update TI\_READ\_P27\_IO1 SET "IO\_ACTIVATE?"='YES';

update TI\_READ\_C27\_IO1 SET "IO\_ACTIVATE?"='YES';

update TI\_READ\_Y27\_IO1 SET "IO\_ACTIVATE?"='YES';

update TI\_READ\_Y27\_IO2 SET "IO\_ACTIVATE?"='YES';

update TI\_READ\_H27\_IO1 SET "IO\_ACTIVATE?"='YES';

update ti\_read\_tank28 SET "IO\_ACTIVATE?"='YES';

update TI\_READ\_H27\_IO2 SET "IO\_ACTIVATE?"='YES';

update TI\_READ\_H27\_IO3 SET "IO\_ACTIVATE?"='YES';

update TI\_READ\_H27\_IO4 SET "IO\_ACTIVATE?"='YES';

-- Wait for 0.2 seconds

IO1\_TIME.IP\_INPUT\_TIME = cast("time".schedule\_time as integer) + 2;

WHILE cast("time".schedule\_time as integer) <

cast(IO1\_TIME.IP\_INPUT\_TIME as integer) DO

a = "time".schedule\_time;

END

END