Modernization of Mold Heating/Cooling Pump Package Operator Interface (MHC PP OI)

Federal Manufacturing & Technologies

J. A. Salmond
Department 8A4

KCP-613-8601

Distributed July 2009

Final Report

Á

07] | [ç^âÁ | Á ~ à|æÁ^|æ^Lã dã~ ç } /æ Á } |ã ã^â

Prepared under prime contract DE-ACO4-01AL66850 for the
United States Department of Energy

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process or service by trade names, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

All data prepared, analyzed and presented has been developed in a specific context of work and was prepared for internal evaluation and use pursuant to that work authorized under the reference contract. Reference herein to any specific commercial product, process or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation or favoring by the United States Government, any agency thereof or Honeywell Federal Manufacturing & Technologies, LLC.

Printed in the United States of America.

This report has been reproduced from the best available copy.

Available to DOE and DOE contractors from the Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, Tennessee 37831; prices available from (865) 576-8401, FTS 626-8401.

Available to the public from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal, Rd., Springfield, Virginia 22161, (703) 487-4650.

Modernization of Mold Heating/Cooling Pump Package Operator Interface (MHC PP OI)

J. A. Salmond
Department 8A4

KCP-613-8601

Distributed July 2009

Final Report

Contents

<i>Section</i>	<i>Page</i>
Abstract	6
Summary	7
Discussion	8
Scope and Purpose	8
Activity	9
Operator Interface	9
Controller	9
Documentation	9
Accomplishments	10
Future work	10
Distribution	11
Attachment 1: SQA Plan	1-1
Attachment 2: Design and Maintenance Document	2-1
Attachment 3: Customer Guide	3-1

Abstract

The modernization of the Mold Heating and Cooling Pump Package Operator Interface (MHC PP OI) consisted of upgrading the antiquated single board computer with a proprietary operating system to off-the-shelf hardware and off-the-shelf software with customizable software options. The pump package is the machine interface between a central heating and cooling system that pumps heat transfer fluid through an injection or compression mold base on a local plastic molding machine. The operator interface provides the intelligent means of controlling this pumping process. Strict temperature control of a mold allows the production of high quality parts with tight tolerances and low residual stresses. The products fabricated are used on multiple programs.

Summary

The purpose of the MHC PP OI system is to control the fluid and/or metal temperature of a plastic injection or compression mold.

Cyclic temperature control of compression and injection molded parts has been an integral part of plastics fabrication at the Kansas City Plant (KCP) since the mid 1960s. Temperature controlled fluid flowing through mold sections, called zones, yield high-quality parts with tight tolerances and low residual stresses. Cyclic control has evolved from human operators actuating manual valves to analog controls manipulating proportional control valves. The next steps were to direct digital control using a centralized computer and finally to microprocessors implemented at each molding site.

This evolution will be continued by replacing the current microprocessors that have been in operation since early 1982. New control systems provide many improvements compared to the current configuration. Among these improvements are a faster processor speed, larger volatile and non-volatile memory, rugged industrial chassis, data logging, graphical user interface, windows-like screen navigation, and user friendliness. The old components are antiquated and nearly impossible to replace.

A programmable logic controller (PLC) was chosen as the controller as it offers extreme manufacturing process flexibility in controlling what a piece of equipment does, when it performs an operation, and how to incorporate operator input at only the right time. The software can also be modified more quickly than text based programming.

An industrial computer with a touch screen monitor was used for the operator interface. The interface software provides building block functions such as screen construction, screen objects and buttons, a scripting language, timed tasks, and hardware communication worksheets. These building blocks are used to create a customizable software interface.

The entire old cabinet and hardware were replaced; nothing was reused.

Detailed documentation for the development and design will be located in the appendices rather than repeating the text in the body of this report.

Discussion

Scope and Purpose

The old operator interface was antiquated to the point where the only hardware spares existing were those in-house. Therefore, in the event of a failure, one machine was raided to get another machine running. The user interface was challenging, text based, required specific text inputs, and required some knowledge to be used effectively. The goal of the project was to correct those issues by replacing the hardware and software.

The purpose of the MHC PP OI is to control the fluid and/or metal temperature of a plastic injection or compression mold. A mold may have 2 or 4 zone sections. The temperature of the fluid in a zone is controlled to the set-point by a standard PID feedback algorithm which modulates the control valves that supply hot and cold fluid. Metal temperature control uses a special control algorithm developed by the KCP to account for special non-linear features of this mode of control. The custom control algorithm's control loop is cascaded with the PID loop. The development of this new algorithm enabled the input of the desired mold temperature profile into the recipe, rather than having to produce the desired profile through trial and error of fluid set-points.

The molding machines are used in support of multiple programs. The molding capability would still be necessary if the manufacturing facility were to be relocated. A project request was submitted to the KCP Campaigns department and funds were granted for the work, project number 703030.

Project activity 70303020 was used for labor charging and to purchase development material. The development material was later used for the production release on a LaRose 750 ton compression molding press with a 2-zone pump package in the plastics molding department at the KCP.

It was desired to update system components to current industry standards, with as little customization as possible. A low degree of customization offers a more easily maintainable system over the life of the system. The hardware used is readily available and has a historically-proven reliability. However, the software must be customized to the application. A base software package was chosen which allowed customizable modules to be configured to handle all aspects of the user interface requirements.

Activity

Operator Interface

The operator interface was constructed to be similar to a Windows[®]-type experience. Screens are very graphical, colorful, and contain pertinent information. Screens were constructed so that the user would not have to navigate between many screens to perform given tasks. User accounts were created, allowing different levels of access to different screens depending on the user's production role.

The touch screen enables the user to perform operations without the use of other input devices such as a mouse and/or keyboard. This greatly simplifies the using experience. Buttons were configured so that there is little chance of a user pressing the wrong button. Many times the user is given a chance to cancel the selected operation.

Controller

The PLC takes over once the user selects a recipe to run; controlling the production cycle and valve actuation. The logic design for the controller operation was conceived using state machine logic. State machine logic breaks down the operations of a machine into small state sets. For a given state set, only one state can be active. The state design was converted to flowchart logic employed by the PLC programming software.

Documentation

When a project of this size is initiated, documentation is important to the success.

Since this project involved the development of custom software, a Software Quality Assurance (SQA) plan was generated and adhered to throughout the endeavor. The SQA plan can be found in Attachment 1. The SQA plan mandates that a design document be released to capture the design intent. The SQA plan also mandates that a maintenance document be kept to capture further changes to the software. The SQA plan permits the design and maintenance document to be combined; this document is found in Attachment 2. The SQA plan also mandates that a customer guide shall be written; this document is found in Attachment 3.

Accomplishments

The upgrade to the MHC PP OI provided a robust, user-friendly replacement. The operator interface is now more meaningful and intuitive. An old system can easily be converted to the new replacement with minimal investment. The new system should provide tens of years of longevity.

Future work

As needed, old systems will be upgraded to the implemented replacement.

Distribution

KCP:

J. Salmond

G. Domer

A. Brown

J. Vincent

M. Trimmer

B. Spieker

D. Prigel

Technical Information Center

THIS PAGE INTENTIONALLY LEFT BLANK

SQA PLAN
Mold Heating and Cooling (MHC) Machine Interface Controls Upgrade

NNR project 7030303
Activity 70303020

Prepared by:
J. A. Salmond

Approved By:

Date:

J. A. Salmond

02/15/2008

Reviewed By:

Date:

2/15/2008

2/15/2008

2/18/2008

Page intentionally left blank.

SQA PLAN

MHC Machine Interface Controls Upgrade

Table of Contents

1. INTRODUCTION.....	3
1.1. PURPOSE.....	3
1.2. SCOPE.....	3
1.3. DEFINITIONS, ACRONYMS, AND ABBREVIATIONS.....	3
1.4. REFERENCES.....	4
2. GENERAL APPLICATION REQUIREMENTS.....	4
2.1. GENERAL.....	4
2.2. SYSTEM PERSPECTIVE	4
2.3. PRODUCT PERSPECTIVE.....	4
2.4. USER CHARACTERISTICS	5
2.5. GENERAL CONSTRAINTS	5
3. SPECIFIC APPLICATION SQA REQUIREMENTS.....	6
3.1. REQUIREMENTS DOCUMENT.....	6
3.2. DETAILED DESIGN DOCUMENT	6
3.3. MAINTENANCE DOCUMENT.....	7
3.4. CODE DOCUMENTATION.....	7
3.5. REVIEW HISTORY'S.....	7
3.6. CODE/TOUCH AND FEEL REVIEWS	7
3.7. TEST PLAN	8
3.8. TESTING AND VALIDATION.....	8
3.9. APPLICATION VERSION CONTROL	8
3.10. CONFIGURATION MANAGEMENT	9
3.11. CHANGE CONTROL	9
3.12. BACK UPS	9
3.13. CUSTOMER GUIDE.....	9

SQA PLAN

MHC Machine Interface Controls Upgrade

1. Introduction

1.1. Purpose

This document is the SQA Plan covering the heating and cooling controls for the plastic injection and compression molds used in Department 26 at the NNSA's Kansas City Plant operated by Honeywell FM&T. The Process Automation Group is responsible for the Mold Heating and Cooling Central System as well as each machine interface to the system. The intended audiences for this document are members of the Process Automation Group, management, quality, manufacturing, and engineering personnel. This document is intended to define the general application requirements and the specific SQA requirements for the application. The defined requirements include: design, development, documentation, tests, reviews, validation, version control, release, maintenance, back up, and change.

1.2. Scope

The software covered by this plan will be used to provide continuous and discrete, as well as automatic and manual control of the MHC Control System. A HMI will be provided to change the setup parameters for the continuous, discrete, automatic, and manual controls, and to provide status of the system. The controller software will be created using Entivity's Think&Do Live Programming Pack application utilizing flow-chart programming running on a Direct Logic 205 Series, Model H2-WPLC3 Windows CE PLC (WinPLC) processor from AutomationDirect.com. The HMI will be created using InduSoft's Web Studio running on a PC with a touch screen and a Windows XP operating system. The WinPLC and PC will communicate system parameters and status via OPC protocol over an Ethernet connection.

1.3. Definitions, Acronyms, and Abbreviations

ACRONYMS-ABBREVIATIONS

FM&T	Federal Manufacturing and Technologies
HMI	Human-Machine Interface (operator interface)
KCP	Kansas City Plant
MHC	Mold Heating and Cooling
MTL	custom mold temperature control algorithm
NNR	Non-Nuclear Readiness campaigns
NNSA	National Nuclear Security Administration
OLE	Object Linking and Embedding
OPC	OLE for Process Control
PAG	Process Automation Group
PLC	Programmable Logic Controller
PC	Personal Computer
SQA	Software Quality Assurance

SQA PLAN

MHC Machine Interface Controls Upgrade

DEFINITIONS

MHC Controls	For this document, this represents the controls necessary to control the temperature of a mold at each molding machine.
Mold Heating and Cooling Central System	Consists of a heater for the “hot” loop and a heat exchanger for the “cold” loop that provides the heating and cooling fluid to each molding machine.

1.4. References

Process Automation Group Policy *Software Requirements Document Outline*, November 1990.

Process Automation Group Policy *Documenting Software*, October 1990.

2. General Application Requirements

This section will attempt to define the general requirements for the application over and above any specific requirements that may be generated by the requesting customer.

2.1. General

The MHC Control System is required to run user defined recipes to cycle the temperature of a mold on a plastic injection or compression molding machine. A mold can consist of two to four zones. Each mold zone's recipe and control parameters can be unique. The recipe can be defined to control the temperature of either the heat transfer fluid, or the mold. A temperature recipe is used to control the heating and cooling profile needed to successfully mold a plastic part to tight tolerances and reduce material creep over the life of the part. The recipe is executed by an algorithm that adjusts a control valve to let hot or cold heat transfer fluid into the mold zone to attain its defined temperature set-point. The recipe is developed and stored on the PC, then transmitted to the WinPLC at runtime.

2.2. System Perspective

The MHC Control System will be designed to run on a Direct Logic 205 Model PLC base, using a WinPLC processor with the HMI running on a touch screen with a Windows XP PC. The WinPLC and PC will communicate system parameters and status via OPC protocol over an Ethernet connection.

2.3. Product Perspective

The MHC Control System must be able to provide the functionality that was available on the single board computer controller that was installed in the early

SQA PLAN

MHC Machine Interface Controls Upgrade

1980's. The goal is to provide that functionality and modernize the controls and HMI to make the operation easier and more intuitive.

2.4. User Characteristics

The MHC Control System will be used primarily by engineers and operators/technicians. The HMI will be accessed via a touch screen panel that will be designed to be Windows-like in appearance. It shall be expected that users have a working knowledge of personal computers.

2.5. General Constraints

The primary purpose of the system will be to provide continuous and discrete control of the MHC Control System. A HMI will be provided to change the setup parameters for the continuous control and to initiate the discrete controls. Status screens will also be provided to display information about the system. There will be three access levels on the HMI:

- Operator, the most basic access. Will be able to select a recipe and run it, and view process status and history.
- Engineer. In addition to operator access, will be able to define and manipulate recipe file set-points and control parameters, and enter the MTL auto-tuning routine.
- PAG, the highest access level. Will be able to access the control valves directly, set 2 or 4 zone machine configuration, and verify other system parameters.

The continuous control will be used to control the mold temperature to a defined mold or fluid set-point and will be implemented using feedback control.

Concepts include:

- thermocouples to transmit the fluid temperature and mold temperature;
- error signal will be calculated based on fluid control or mold control;
- PID algorithm applied to fluid error signal to manipulate zone control valves; and
- custom metal algorithm (MTL) applied to mold error signal to manipulate zone control valves.

The discrete control will be used to perform actions not covered by the continuous control. Actions include:

- manipulating indicator lights on the HMI;
- initiating a cycle start, cycle abort, or dry cycle;
- monitoring when the machine is ready to start the recipe; and
- emergency shutdown.

The automatic mode will follow the chosen recipe to process a part. Concepts include:

- 90 recipes can be defined;

SQA PLAN

MHC Machine Interface Controls Upgrade

- 20 segments per zone can be defined, each segment consists of defining a time set-point, a temperature set-point, and whether the segment should be a dwell or ramp segment;
- change constant temperature set-point for a zone on the fly;
- define PID/MTL algorithm parameters for each zone;
- a part is processed by selecting a recipe from a list then loading the recipe on the WinPLC; and
- process status and defined parameters available via a HMI screen.

The manual mode will be used very rarely. It will control the actual position of the control valve for a zone. This mode can only be accessed by PAG users.

The HMI will provide the means to enter and/or display information for the control system. This information includes:

- percent output for control valves;
- temperature sensor values;
- trend graph traces of temperature sensors;
- visual indication of activity (e.g., color changes or indicators for various state changes);
- soft “push buttons” for discrete events;
- PID/MTL constants;
- set-points;
- alarm or warning conditions;
- help screens; and
- an auto-tune function to determine MTL algorithm parameters.

3. Specific Application SQA Requirements

This section defines the specific SQA requirements for the MHC Control System.

3.1. Requirements Document

A system requirements document will be prepared defining the specific requirements to control the MHC Control System. The system requirements document shall follow the outline defined by Process Automation Group Policy for *Software Requirements Document* Outline. This document will be frozen after initial release such that any new requirements or changes to the design will be documented in the Maintenance Document.

3.2. Detailed Design Document

A detailed design document will be prepared defining the initial design concepts and philosophies of the MHC Control System. The detailed design document shall include, as necessary, state transition diagrams, procedure level structured English, user interface screen formats, and reporting formats. The detailed

SQA PLAN

MHC Machine Interface Controls Upgrade

design document may be combined with the maintenance document in a Design and Maintenance Document.

3.3. Maintenance Document

A maintenance document shall be required to document the "As Built" application. The maintenance document may be combined with the detailed design document in a Design and Maintenance Document. The maintenance document shall include the following, as a minimum.

- 3.3.1. Listing of and general description of all modules developed for the application.
- 3.3.2. Detailed discussion of the interaction of the application with external systems.
- 3.3.3. Detailed discussion of the program flow.
- 3.3.4. Detailed discussion of states/subroutines/functions performing unusual or intricate activities that may not be apparent to a reviewer/maintainer.
- 3.3.5. Detailed discussion of the format of all data files.
- 3.3.6. Listings of the operator interface documentation.
- 3.3.7. Code listing.

Revision history information may also be included, but shall be treated only as a reference.

3.4. Code Documentation

Code documentation shall be per Process Automation Group *Documenting Software*. Revision histories of the WinPLC and HMI code shall be maintained in each application. Revision histories shall commence when the code is released for production.

3.5. Review History's

Electronically maintained history files (i.e., punch list, testing results, etc.) shall be maintained of touch and feel/testing reviews with a dated developer response to each item.

3.6. Code/Touch and Feel Reviews

Code reviews shall be performed by a PAG peer. Touch and feel reviews shall be conducted by non-developer personnel of the PAG. Touch and feel reviews shall be primarily concerned with:

SQA PLAN
MHC Machine Interface Controls Upgrade

- ease, consistency, and intuitiveness of the application's HMI and
- completeness of the application.

3.7. Test Plan

It is expected that the developer and application testers will thoroughly test the application prior to release (see Testing and Validation below for details).

3.8. Testing and Validation

The application shall be thoroughly tested for bugs that would cause "unexpected" application failure and validity of the data collected, by the developer, prior to testing by PAG personnel. Simulations will be performed in parallel with the code development to predict the validity of the control algorithm. PAG personnel not involved with the application development/coding shall conduct at least one review to test the application attempting to cause it to fail. Documentation of any discrepancies, failures, or suggested enhancements shall be maintained in the form of a punch list. For each discrepancy and failure, the developer shall indicate below the item in the electronically maintained punch list, that the item has been corrected, give an account of corrective action, and date and electronically initial the entry. For each suggested enhancement, the developer shall indicate below the punch list item, whether the suggestion was incorporated or not and, if not, what rationale was used for rejection. Multiple tests by PAG personnel are strongly encouraged.

3.9. Application Version Control

The version numbering schema for the application shall be as follows. The version number shall consist of three components **X**, **Y**, and **Z** (e.g., **X.Y.Z**, **v3.1.6**). The **X** component is reserved to reflect major additions or deletions of requirements for the application. The **X** component shall be numbered beginning with 3 at initial production release. The current released version of the software is 2.x, and although it is not directly portable to the upgrade, the upgrade will continue this sequence. The **Y** component shall be incremented when major changes to application structure or flow result in a new release. The **Y** component shall be numbered beginning with 0 at initial production release. The **Z** component shall be incremented when minor bug fixes or changes in philosophy require a new release. The **Z** component shall be numbered beginning with 0 at initial production release.

The application's version shall be maintained in the application. Separate version numbers will be maintained for the WinPLC logic and the HMI. Both version numbers will be visible on a HMI screen. Both versions will be numbered 3.0.0 at initial production release.

SQA PLAN
MHC Machine Interface Controls Upgrade

3.10. Configuration Management

Any time the **X**, **Y**, or **Z** component is incremented, a new directory shall be created with indication of the new application version embedded within the directory name. A copy of the original modules shall be made in the new directory. All subsequent changes shall be made in the new directory, preserving the original code in the old directory. Changes may be freely made in the new directory by any PAG application developer. Code in the old directory shall not be changed unless that version of the application is the current production release and then only to fix minor bugs.

When a new production release is made, a copy of all code modules/files, make files, PAG generated libraries, and any other pertinent files that comprise the release shall be made on tape or disk, labeled appropriately, and stored in the PAG area. The release tape/disks shall be preserved until any new release is field proven rendering the old tape/disks archaic.

3.11. Change Control

It shall be the responsibility of the PAG developer(s) to maintain control of the non-released and released code.

3.12. Back Ups

While released/development code is being changed, it shall be the responsibility of the PAG developer(s) to ensure that daily backups are made. At least once a week, a full back up shall be made of all directories containing application code and application dependent code (e.g., compilers, DLL's). Full back ups shall be retained for not less than one month.

3.13. Customer Guide

In addition to the on-line help screens, a printed "customer guide" shall be developed when the application is ready for initial release. PAG personnel shall be responsible for developing the "customer guide".

***Design and Maintenance Document
Mold Heating and Cooling Pump Package Interface and Controls Upgrade
(MHC PP OI)***

***NNR project 703030
Activity 70303020***

Prepared by:

J. A. Salmond

Josh A. Salmond 06/26/2009

Approved By:

[Signature]

Date:

6/26/09

Reviewed By:

[Signature]

Date:

6/26/09

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

PAGE INTENTIONALLY LEFT BLANK

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

Table of Contents

1.0	INTRODUCTION	6
1.1.	PURPOSE	6
1.2.	SCOPE	6
1.3.	GENERAL INFORMATION	6
1.4.	DEFINITIONS, ACRONYMS, AND ABBREVIATIONS	7
1.5.	REFERENCES	8
2.0	GENERAL REQUIREMENTS	9
2.1.	PROJECT PERSPECTIVE	9
2.2.	SOFTWARE FUNCTIONS	18
2.3.	USER CHARACTERISTICS	19
2.4.	GENERAL CONSTRAINTS	19
2.5.	SECURITY PLANS	19
2.5.1.	Classified Information Security	19
2.5.2.	Physical Security	19
3.0	FUNCTIONAL REQUIREMENTS	19
3.1.	SYSTEM FUNCTIONAL REQUIREMENTS	19
3.1.1.	Provide Features Equivalent to Demand Functions	19
3.1.2.	Ability to Run 4-Zone PID and Metal Control Loops	20
3.1.3.	Separate HMI and Controller Function	21
3.1.4.	Creating, Editing and Storing Recipes	21
3.1.5.	Collect and Store Trend Data	21
3.1.6.	Backup Capability	21
3.2.	SOFTWARE FUNCTIONAL REQUIREMENTS	21
3.2.1.	Software Operating Requirements	21
3.2.2.	HMI Software Functional Requirements	21
	Navigation & Status	23
	Recipe Viewer	24
	Recipe Status	25
	Trend Graph	26
	Recipe Editing – Recipe Segments	27
	Recipe Editing – Rec PID/MTL Constants	28
	Manual Constant Temperature	29
	Event Viewer	30
	PAG Utilities	31
3.2.3.	PLC Software Functional Requirements	31
	Control Logic	32
	Continuous Control	32
	Human Machine Interface Communication	34
3.3.	HARDWARE FUNCTIONAL REQUIREMENTS	35
3.3.1.	HMI Hardware Functions	35
3.3.2.	PLC Hardware Functions	35
3.4.	PERFORMANCE REQUIREMENTS	36
3.5.	DESIGN CONSTRAINTS	36
3.6.	ATTRIBUTES	36
3.7.	EXTERNAL INTERFACE REQUIREMENTS	36
3.8.	HARDWARE AND SOFTWARE INSTALLATION	37
4.0	STATE SETS	37

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

5.0	LESSONS LEARNED	37
5.1.	ENTITY THINK & DO ERRORS.....	37
5.1.1.	No Default Printer Defined	37
5.1.2.	Socket Timeout	37
5.2.	ENTITY DATAPACK v7.3	38
5.3.	WINDOWS FIREWALL.....	38
5.4.	TEMPERATURE AS PERCENT OF RANGE.....	39
5.5.	PRELOADING THE INTEGRATOR IN THE PID ALGORITHM.....	39
5.6.	PRELOADING THE MTL PIPELINE.....	40
5.7.	ALGORITHM CALCULATIONS DURING RAMP SEGMENTS.....	41
5.8.	WINDOWS XP USER PRIVILEGES	41
5.9.	OPC COMMUNICATION.....	41
5.10.	SEGMENT TIMER ACCURACY TEST	42
6.0	DESIGN CHANGE LOG	43
6.1.	HMI v3.0.0 AND PLC v3.0.0, 31-OCT-2008	43
6.2.	HMI v3.1.0, 08-APR-2009	43
6.3.	PLC v3.0.1, 08-APR-2009.....	43
APPENDIX A: SOFTWARE OPERATING REQUIREMENTS		44
APPENDIX B: HARDWARE & SOFTWARE INSTALLATION GUIDE		53
APPENDIX C: STATE SET CONVERSION TO THINK & DO FLOWCHARTS		90
APPENDIX D: WINPLC CODE LISTING.....		95
APPENDIX E: HMI CODE LISTING.....		96

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

List of Figures

<i>Figure 1: Typical Desired Plastic Compression Mold Temperature Cycle</i>	<i>10</i>
<i>Figure 2: Typical Compression Mold and Fluid Temperature Cycle</i>	<i>10</i>
<i>Figure 3: Typical Desired Plastic Injection Mold Temperature Cycle.....</i>	<i>11</i>
<i>Figure 4: Typical Injection Mold Temperature Cycle.....</i>	<i>11</i>
<i>Figure 5: Mold Heating and Cooling Central System Hot and Cold Oil Supply.....</i>	<i>12</i>
<i>Figure 6: One Mold Zone of an Individual Pumping Subsystem</i>	<i>13</i>
<i>Figure 7: Schematic Diagram of Controller/Computer for a Subsystem.....</i>	<i>14</i>
<i>Figure 8: Single Zone Schematic of a Four-Zone System</i>	<i>15</i>
<i>Figure 9: Single Zone Schematic of a Two-Zone System</i>	<i>16</i>
<i>Figure 10: Zone Control Valve Operation.....</i>	<i>16</i>
<i>Figure 11: Existing MHC PP OI Controller</i>	<i>17</i>
<i>Figure 12: Example Temperature Cycle</i>	<i>18</i>
<i>Figure 13: Cascaded Control Loop Block Diagram</i>	<i>33</i>

List of Tables

<i>Table 1: Demand Functions</i>	<i>20</i>
<i>Table 2: MHC PP OI Controls I/O Requirements</i>	<i>35</i>

List of Appendices

<i>Appendix A: Software Operating Requirements.....</i>	<i>44</i>
<i>Appendix B: Hardware & Software Installation Guide</i>	<i>53</i>
<i>Appendix C: State Set conversion to Think & Do Flowcharts</i>	<i>90</i>
<i>Appendix D: WinPLC Code Listing</i>	<i>95</i>
<i>Appendix E: HMI Code Listing.....</i>	<i>96</i>

List of Appendix Figures

<i>Figure C - 1: Reset Action.....</i>	<i>91</i>
<i>Figure C - 2: Generic State Set.....</i>	<i>92</i>
<i>Figure C - 3: Generic State Set as Subroutine.....</i>	<i>93</i>
<i>Figure C - 4: Loopback State.....</i>	<i>94</i>

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

1.0 Introduction

1.1. Purpose

This document contains the design based on the System Requirements for replacement of the Mold Heating and Cooling (MHC PP OI) Pump Package Interface and Control systems in the plastics molding department (currently D/26) at the NNSA's Kansas City Plant operated by Honeywell FM&T. This document will contain all documentation related to the requirements, design, and maintenance of the project. The primary purpose of this software governed by this document is to provide an operator interface with the pump package. This document defines the specific design steps for the:

- pump package zone controls; and
- operator interface.

This document will cover the production release of the software. The production release entails HMI version 3.1.0 and PLC version 3.0.1. A pre-production pilot release, v3.0.0 for both the HMI and PLC was the initial release. Any design changes after the production release will be documented in the section, *Design Change Log*.

The first system was installed on the LaRose 750 ton compression press, press CE#68980, pump package CE#69299; a 2-zone system, on April 8th, 2009.

1.2. Scope

The software and hardware described by this document will be used to:

- provide manual and automatic control of the pump package,
- control the temperature of a molding zone via a feedback control algorithm,
- provide an operator interface to change the profile parameters for the manual and automatic cycles,
- provide an operator interface to initiate the discrete controls, and
- provide an operator interface to display the status of the system.

1.3. General Information

Project Title:	MHC Controls Upgrade
Platform:	WinPLC and PC
Classification:	Unclassified
Funding:	NNR 703030, Activity 70303020
Commissioners:	Brian Spieker, Matthew Trimmer
Developers:	Josh Salmond
Sites/Machines:	FM&T KCP Plastics Molding Department
Platform specifics:	

DESIGN AND MAINTENANCE DOCUMENT

MHC PP OI Controls Upgrade

The control software will be created using Entivity Think&Do Live! Programming Pack v5.6 from Automation Direct running on a Direct Logic 205 Series Model H2-WPLC3 WinPLC using the WinCE v2.12 operating system. The runtime software on the WinPLC will be Entivity DataPack v7.3.

The operator interface will be created using InduSoft Web Studio v6.1+SP4 from InduSoft.com running on a Windows XP+SP2 industrial computer with a USB touch screen interface. Separate development and runtime licenses will be used.

The controller and the HMI will communicate via OPC protocol over standard Ethernet cabling.

1.4. Definitions, Acronyms, and Abbreviations

ACRONYMS

A/D	Analog to Digital Converter
CE	Capital Equipment
COTS	Commercial Off The Shelf
CRT	Cathode Ray Tube
FM&T	Federal Manufacturing and Technologies
HMI	Human Machine Interface (Operator Interface)
I/O	Input/Output
I/P	Current to Pneumatic
KCP	Kansas City Plant
MHC	Mold Heating and Cooling
MHC PP OI	Mold Heating and Cooling Pump Package Operator Interface
NNR	Non-Nuclear Readiness Campaigns Program
NNSA	National Nuclear Security Administration
OPC	Object Linking and Embedding (OLE) for Process Control
PAG	Process Automation Group
PC	Personal Computer
PID	Proportional Integral Derivative
PLC	Programmable Logic Controller
PP	Pump Package
WinPLC	WinCE (Windows Embedded) based PLC processor

DEFINITIONS

Greenback	A published formal report documenting projects or the development of new processes at the KCP, which is distributed across the Nuclear Weapons Complex.
MHC PP OI Controls	For this document, this represents the controls necessary to control the temperature of a mold at each molding machine.
Mold Heating and Cooling Central System	Consists of a heater for the “hot” loop and a heat exchanger for the “cold” loop that provides the heating and cooling fluid to each molding machine, see <i>Figure 5</i> .

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

MTL	Custom metal temperature control algorithm.
Process Variable	PV, Measured value of a parameter that affects a process, e.g., pressure, temperature, velocity, etc.
Recipe	Data file that contains a pre-assigned group of process set-points and parameters.
Set-Point	SP, Desired value of the process variable.

1.5. References

The following greenbacks were extensively consulted in the development and design of the system, and are referenced in the text via a superscript:

¹ X-248	R.H. Floersch, Mold Heating and Cooling Systems	May 1970
² BDX-613-369	R.H. Floersch, Theory and Operation of the Bendix Mold Heating and Cooling Equipment	June 1971
³ BDX-613-742	R.H. Floersch, Fluid Flow Analysis in the Mold Heating and Cooling Control Loop	Sept 1972
⁴ BDX-613-1990	R.H. Floersch, Mold Heating and Cooling Systems	Sept 1978
⁵ BID-A461	R.H. Floersch, Hardware Selection and Design Considerations for Microprocessor Controllers for Mold Heating and Cooling	June 1983
⁶ BDX-613-3181	R.H. Floersch, Microprocessor Controller Software Documentation for Mold Heating and Cooling	Nov 1984
⁷ BDX-613-3390	R.H. Floersch, Microprocessor Controller for Mold Heating and Cooling	Jan 1986

The Demand Function User Manual, published February 19, 1985, was also used in transitioning to the new system. These manuals were provided for each machine upon implementation. An abbreviated copy of the text in the user manuals can be found in BID-A461⁵.

The following greenbacks also contain historical information about Mold Heating and Cooling. These documents were consulted but did not contain any new information affecting the design:

• BDX-613-1111	R.H. Floersch, Experimental Testing for Mold Temperature Responses	May 1975
• BDX-613-1764	R.H. Floersch, Computer Simulation of the Mold Heating and Cooling Control Loop, Using Conventional Control Strategies	Feb 1977
• BDX-613-1979	R.H. Floersch, Mold Heating and Cooling Systems	Mar 1978

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

- BDX-613-1980 R.H. Floersch, Digital Controller Design, Analysis, and Implementation for the Mold Heating and Cooling System Sept 1978
- BDX-613-2974 R.H. Floersch, Hardware Selection and Design Considerations for Microprocessor Controllers for Mold Heating and Cooling Nov 1983
- BDX-613-3477 R.H. Floersch, Microprocessor Controller for Mold Heating and Cooling Apr 1986
- BDX-613-5427 D.P. Hoffman, Mold Heating and Cooling Microprocessor Conversion Aug 1994

2.0 General Requirements

This section provides an overall view of the project.

2.1. Project Perspective

The purpose of the MHC PP OI system is to control the fluid and/or metal temperature of a plastic injection or compression mold.

Cyclic temperature control of compression and injection molded parts has been an integral part of plastics fabrication at the KCP since the mid 1960s. Temperature controlled fluid flowing through two or four mold sections, called zones, yield high-quality parts with tight tolerances and low residual stresses. Cyclic control has evolved from human operators actuating manual valves to analog controls manipulating proportional control valves. The next steps were to direct digital control using a centralized computer and finally to microprocessors implemented at each molding site⁷.

This evolution will be continued by the replacement of the current microprocessors that have been in operation since early 1982. New control systems provide many improvements compared to the current configuration. Among them are a faster processor speed, larger volatile and non-volatile memory, rugged industrial chassis, data logging, graphical user interface, windows-like screen navigation, and user friendliness.

The temperature of the fluid in a zone is controlled to the set-point by a standard PID feedback algorithm which modulates the control valves that supply hot and cold fluid. Metal Temperature control uses a special control algorithm developed by Dick Floersch of the KCP to account for special non-linear features of this mode of control^{5,6,7}. The custom control algorithm's control loop is cascaded with the PID loop, reference *Figure 13*. The advantage of using the metal algorithm can be seen in *Figure 1–Figure 4* which illustrate how the fluid set-points were previously manipulated to obtain a desired mold temperature profile^{3,7}. The development of this new algorithm enabled the input of the desired mold temperature profile into the recipe, rather than having to produce the desired

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

profile through trial and error of fluid set-points.

Figure 1: Typical Desired Plastic Compression Mold Temperature Cycle

Figure 2: Typical Compression Mold and Fluid Temperature Cycle

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

Figure 3: Typical Desired Plastic Injection Mold Temperature Cycle

Figure 4: Typical Injection Mold Temperature Cycle

The current interface control hardware is antiquated. It was deployed in the early 1980's, with an expected lifetime of less than 20 years. Purchasing replacement parts for the systems is impossible, and the KCP's supply of replacement parts is dwindling. In order

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

to keep equipment functional, control system hardware must be updated. Along with hardware replacement, new software will be written to support the new platform because the current system's code cannot be ported into newer platforms.

Figure 5 through Figure 10 will illustrate the components of Mold Heating and Cooling. *Figure 5* illustrates the fluid interface of individual subsystems with the Central Heating and Cooling system^{1,4}. The existing Central System uses a boiler for the heater and a cooling tower for the heat exchanger as it was designed to handle some fifteen subsystems.

Figure 5: Mold Heating and Cooling Central System Hot and Cold Oil Supply

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

Figure 6 provides more detail of the fluid line interface of an Individual Pumping Subsystem, depicted as the upper boxes in Figure 5^{1,4}. An Individual Pumping Subsystem may have two or four mold zones depending on what type of mold and molding machine it interfaces with.

Figure 6: One Mold Zone of an Individual Pumping Subsystem

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

Figure 7 depicts the computer, or controller, interface of one mold zone at an individual pumping subsystem^{1,4}. The box labeled computer is precisely what this project will upgrade; this is what is referred to as the MHC PP OI Controls Upgrade. In actuality, the controller controls all of the mold zones at a given subsystem.

Figure 7: Schematic Diagram of Controller/Computer for a Subsystem

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

Figure 8 and Figure 9 provide more detail of the control elements needed at an individual molding machine to interface with the Mold Heating and Cooling Central System². Note that the figures only represent one zone of a local system, and that the only difference between the two figures is that one is a four-zone machine, while the other is a two-zone machine.

Figure 8: Single Zone Schematic of a Four-Zone System

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

Figure 9: Single Zone Schematic of a Two-Zone System

Figure 10 displays in detail how the pneumatic control valves for the zones are configured.

Figure 10: Zone Control Valve Operation

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

The PID Algorithm outputs 0 to 100% and is clamped at both ends; it can output no less than 0%, and no greater than 100%. Full Cold Fluid enters the zone loop at a valve setting of 0%, or 4 milliAmps, or 0 bits. Full Hot Fluid enters the zone loop at a valve setting of 100%, or 20 milliAmps, or 4095 bits. At a 50% valve setting, the fluid in the zone loop recirculates with no hot or cold fluid introduced. Therefore, a control feedback sample error of zero should result in a valve setting of ~50%. A PID output of 100% will result in a valve setting of 100%; a PID output of 0% will result in a valve setting of 0%. The supply and return of fluid is a balanced mass system; thereby balancing the valve operation such that if a supply valve is at 14%, the return valve will be at 14%.

A picture of the existing controller used to interface with the Mold Heating and Cooling system can be seen in *Figure 11*.

Figure 11: Existing MHC PP OI Controller

The MHC PP OI is required to run user defined recipes to cycle the temperature of a mold on a plastic injection or compression molding machine. Typically, a four-zone machine will be an injection molding machine. A two-zone machine may be either a compression or injection molding machine. An example of how mold zones may be

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

configured for a four-zone machine might be: Zone 1 - punch base, Zone 2 - punch insert, Zone 3 - cavity insert, Zone 4 - cavity base.

Each mold zone recipe and control parameters are unique. The zone recipe can be defined to control the temperature of either the heat transfer fluid, or the mold. A zone recipe consists of up to twenty segments. Each segment consists of a temperature set-point (mold or fluid), time duration, and whether or not the segment should be a ramp or dwell segment. The first two segments of a recipe, 0 and 1, are reserved for mold and machine conditioning, called Preheat. An artificial profile of a mold temperature cycle can be seen in *Figure 12*.

Figure 12: Example Temperature Cycle

The entire recipe is executed by the selected control algorithm that controls a valve that lets hot or cold heat transfer fluid into the zone to attain its defined recipe set-point. The recipe is developed and stored on the PC, then transmitted to the controller at run-time. It should also be noted that the segment is defined by the time; meaning that when the segment time set-point has expired, control will continue to the next segment. The temperature is merely a target to be reached within the time specified. Intimate knowledge or recipe experimentation must be used to configure a recipe that can be successfully controlled to its set-points.

2.2. Software Functions

The primary purpose of the system software will be to provide continuous and discrete control of the Pump Package interface to the Mold Heating and Cooling Central System. Software will reside on both the HMI, and the WinPLC. HMI software will have interface information, a recipe editor, and process monitoring capabilities. The controller

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

software will run all control algorithms, handle program execution, read data from and write data to the HMI, and aid screen navigation.

2.3. User Characteristics

The controls will be used primarily by process engineers, operators/technicians, and PAG personnel. The HMI will employ a touch-screen to enable all capabilities, select and define tasks, and navigation through screens. The HMI will provide the capability to press a graphical button to select the major tasks. It shall be expected that users have a working knowledge of and the requisite skills to operate the machine.

2.4. General Constraints

The principal constraint on the replacement MHC PP OI Controls is to include all current capabilities, and control algorithms.

2.5. Security Plans

2.5.1. Classified Information Security

The application acquires unclassified data; therefore, no security plan is required.

2.5.2. Physical Security

The controller will be located inside a cabinet. Personnel directly involved with the MHC PP OI Controls will be allowed physical access.

3.0 Functional Requirements

This section documents the design requirements used and the implemented design specifics for the MHC PP OI Controls Upgrade.

3.1. System Functional Requirements

The System Functional Requirements for the replacement MHC PP OI Controls are the umbrella requirements that apply to both hardware and software. These requirements were derived from the current system and from the technical reports listed in Section 1.5. The current system performs sufficiently; any replacement system should perform as well or better. Along with pre-existing functionality, new requirements were added. These new requirements were based on standard features found on most modern control systems.

3.1.1. Provide Features Equivalent to Demand Functions

The current MHC PP OI Controls provide the user with many “Demand Functions”. Demand Functions are text commands that users type into the CRT terminal to communicate with the machine. *Table 1* lists the Demand Functions and a brief description of their task⁵:

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

DF1	Change Control Output Percentage
DF2	Change/Display Control Set-Point
DF3	Set Control Mode (PID, Manual, or Metal)
DF4	Activate/Deactivate Recipe Zones
DF5	Edit/List Recipes
DF6	Display Process Monitor
DF7	Read/Write Bubble Memory
DF8	Change/Display Abort Temperatures
DF9	Change/Display Metal Temperature Limits
DF10	Change/Display PID Constants
DF11	Put Demand Functions in Default Conditions
DF12	Change/Display Metal Dynamic Constants
DF13	Change/Display/Start Time Constant Test
DF14	Change Controller Sample Periods
DF15	Change/Display Monitor Time Periods
DF16	Change Baud Rate
DF17	Display Metal and Fluid Integral Square Error
DF18	Enter Time
DF19	Reset Console Lights
DF20	Display Demand Function Menu

Table 1: Demand Functions

These are the Demand Functions used by the operator and engineers. A higher set of functions (DF26-30) exist for administrator use. The higher set of Demand Functions were used for computer and A/D diagnostics. Diagnostic interfaces are provided with the new hardware, thus these DF's will not be needed.

The new machine will employ a touch screen, thus eliminating the need for the user to type in text commands. Even though the Demand Function paradigm will be eliminated, the new system will provide features that are equivalent to those existing.

Demand Functions that will be excluded in the new implementation are 11, 15, 16, and 19 because they will not apply. DF14 will intentionally not be implemented because the sample periods that exist now have been running successfully for years. The reason DF14 was implemented in the previous version was because these values had to be determined during implementation.

3.1.2. Ability to Run 4-Zone PID and Metal Control Loops

The system will be capable of processing the control algorithms for both the PID and MTL control loops for up to a 4-zone machine. The Metal Temperature

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

Algorithm is calculation intensive and will be executed in real-time. The system will be configurable to set as a 2-zone or a 4-zone machine.

3.1.3. Separate HMI and Controller Function

Should the HMI crash during recipe execution, the controller will continue to function independently, at least until the end of the currently running recipe.

3.1.4. Creating, Editing and Storing Recipes

The system will allow for the creation, editing, and storing of temperature cycle recipes.

3.1.5. Collect and Store Trend Data

The system will be capable of collecting and storing temperature vs. time data files, which includes recipe temperature set-points for reference.

3.1.6. Backup Capability

The system will provide backups of recipe and trend data. This requirement will be satisfied by having an additional hard drive on the industrial computer. A Windows Scheduled Task will be set up that executes a Command File Script to backup selected directories that the necessary data files.

3.2. Software Functional Requirements

This section defines the software functional requirements for the MHC PP OI Controls Upgrade. Software requirements for both HMI and WinPLC will be addressed.

3.2.1. Software Operating Requirements

The first set of high level operating requirements were generated as a bulleted list at the project start and used as the design basis for developing the PLC State Sets and HMI functionality. These are listed in *Appendix A: Software Operating Requirements*.

3.2.2. HMI Software Functional Requirements

The HMI provides a way for the user to communicate with the controller. It consists of a touch screen monitor and an industrial computer. The functional requirements for the HMI software were derived from section 3.1 *System Functional Requirements*, *Appendix A: Software Operating Requirements*, and from running and observing the currently installed system.

The HMI is used to

- Store recipe data on disk
- Store historical trend data
- Edit recipes

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

- Select and start a recipe
- Initiate discrete logic by pressing a screen image
- Monitor process variables
- Manually enter set-points.

Three HMI logins are available to the user. Valid logins are Operator, Engineer, and PAG. Operator does not require a password. Engineer and PAG logins are password protected. The password for Engineer login, mh&c, is case insensitive.

The screens available to the user depend on who the user is logged-in as at run time. Operator has access to screens: *Recipe Viewer*, *Recipe Status*, *Trend Graph*, and *Historical Trend Graph*. Engineer has access to Operator screens plus: *Recipe Editing – Recipe Segments*, *Recipe Editing – Rec PID/MTL Constants*, and *Event Viewer*. PAG has access to Operator and Engineer screens plus the *PAG Utilities* screen.

An abbreviated list of the main screens is discussed below. A short description is provided with each screen. More information is available including all screens, their relationships, and their functions in *Appendix E: HMI Code Listing* or refer to the Customer Guide.

DESIGN AND MAINTENANCE DOCUMENT

MHC PP OI Controls Upgrade

Navigation & Status

The *Navigation & Status* screens will always be visible to the user. The *Navigation* screen is the toolbar on the left that houses an array of buttons to navigate to other available screens. The *Status* screen provides indication of the WinPLC status, E-STOP condition, login functions, and a button to exit the program.

DESIGN AND MAINTENANCE DOCUMENT

MHC PP OI Controls Upgrade

Recipe Viewer

RECIPE VIEWER - FOR OPERATOR VERIFICATION AND TO START RECIPES

Select Recipe

Start Recipe

Close Recipe

Run Preheat

Skip Preheat

ZONE

Zone 1

Zone 2

Zone 3

Zone 4

SEGMENTS

Seg 0-9

Seg 10-19

View Segment Grapher

RECIPE NAME Viewing Recipe from Disk

myRecipe

PART NUMBER(S) RECIPE DESCRIPTION

45678, 123456-78 This part is used to make a whachamigger for the whachamacallit.

Est. Cycle Time

00:33:00

SEG#	TIME	TEMP	Ramp	Ramp Rate	Segment Comment
0	300	100	☐	0.000000	This segment is used to preheat the mold.
1	300	175	☒	0.250000	Ramp up the temperature, then place the part in the mol...
2	600	275	☐	0.000000	Process the part for 5 minutes.
3	180	290	☐	0.000000	Process the part at a slightly higher temperature for 3 ...
4	600	100	☒	-0.316667	Ramp to cooldown and remove the part.
5	0	50	☐	0.000000	
6	0	50	☐	0.000000	
7	0	50	☐	0.000000	
8	0	50	☐	0.000000	
9	0	50	☐	0.000000	

Manual Preheat

Cycle Abort Temp SP

Last SP

Mold PreHT Temp Limits

Upper Limit:

Lower Limit:

Control Mode

Fluid Recipe Profile

Fluid Consts:

K_P 12.0

K_I 1.5

K_D 0.1

Mold Dynamic Constants:

Dead Time (sec) θ

Time Constant (sec) τ

Closed Loop Time Constant (sec) λ

Entered Gain Default: 0.33 K'

Steady State Gain K_m

The *Recipe Viewer* screen is used by the Operator to select and run a recipe. It displays a read-only copy of the recipe parameters.

DESIGN AND MAINTENANCE DOCUMENT

MHC PP OI Controls Upgrade

Recipe Status

The *Recipe Status* screen shows status for the system as a whole. This screen is available only when running in recipe mode. Process variables shown are:

- Recipe name
- Control mode of the zone
- Current recipe set-points for all zones
- Metal temperature for all zones
- Fluid temperature for all zones

DESIGN AND MAINTENANCE DOCUMENT

MHC PP OI Controls Upgrade

Trend Graph

The *Trend Graph* screen will display real-time process variables on a digital chart recorder during an active running recipe or a completed recipe. These process variables include:

- Historical trend data file name in the title bar, of format: YYYY-MM-DD_HHMM_RecipeName. Where YYYY is the four digit year, MM is the two digit month, DD is the two digit day, HHMM is the hour and minutes, and RecipeName is the name of the recipe used.
- Control mode of the zone
- Current recipe set-points for all zones
- Metal temperature for all zones
- Fluid temperature for all zones

As this data is collected it will be stored for later viewing on the *Historical Trend Graph* screen. The historical screen is similar to the *Trend Graph* screen and is used to retrieve historical information via the above filename.

DESIGN AND MAINTENANCE DOCUMENT

MHC PP OI Controls Upgrade

Recipe Editing – Recipe Segments

RECIPE EDITOR - SEGMENT SETPOINT DEFINITION FOR ZONES

New

Open

Close

Save

Save As

Delete

RECIPE NAME: **myRecipe**

RECIPE DESCRIPTION: This part is used to make a whachamigger for the whachamacallit.

PART NUMBER(S): 45678, 123456-78

Generate RTF Report File for Recipe: "myRecipe"

Start Recipe

View Segment Grapher

Est. Cycle Time: 00:33:00

Control Mode: **Fluid Recipe Profile**

Cycle Abort Temp SP

Last SP

Upper Limit:

Mold PreHT Temp Limits:

Lower Limit:

SEG#	TIME (sec)	TEMPERATURE (deg F)	RAMP? (Y / N)	RAMP RATE (deg F/sec)	SEGMENT COMMENT(S)
0	300	100			This segment is used to preheat the mold.
1	300	175	YES	0.250	Ramp up the temperature, then place the part in the mold after the segment has completed.
2	600	275	NO		Process the part for 5 minutes.
3	180	290	NO		Process the part at a slightly higher temperature for 3 minutes.
4	600	100	YES	-0.317	Ramp to cooldown and remove the part.
5	0				
6					
7					
8					
9					

PREHEAT

Manual Preheat Auto Preheat

Run Preheat Skip Preheat

ZONE

Zone 1 Zone 2

Zone 3 Zone 4

SEGMENTS

Seg 0-9 Seg 10-19

Insert Delete

The *Recipe Editing – Recipe Segments* screen will work in unison with the *Recipe Editing – Rec PID/MTL Constants* screen. A recipe, as it is currently defined, is a listing of parameters for a pre-determined temperature cycle of the mold for a process. The parameters include the control mode and constants, and 20 definable “segments” per zone, for up to four zones. Each segment has three user-selectable values. These values are dwell time, temperature set-point, and whether or not the segment is a ramp segment or a dwell segment. This screen will allow the user to change the control mode, create new segments, edit existing segments, copy segments, insert or delete segments, set abort temperatures, and set metal temperature limits.

DESIGN AND MAINTENANCE DOCUMENT

MHC PP OI Controls Upgrade

Recipe Editing – Rec PID/MTL Constants

RECIPE EDITOR - PID/MTL CONTROL CONSTANTS DEFINITION FOR ZONES

New

Open

Close

Save

Save As

Delete

RECIPE NAME
myRecipe

RECIPE DESCRIPTION
This part is used to make a whachamigger for the whachamacallit.

PART NUMBER(S)
45678, 123456-78

Generate RTF Report File for Recipe: "myRecipe"

Start Recipe **View Segment Grapher** **Est. Cycle Time** 00:33:00

		Zone 1	Zone 2	Zone 3	Zone 4
Control Mode		Fluid Recipe Profile	Mold Recipe Profile	Mold Const Temp	Fluid Const Temp
Const Temp SP		Use Segments	Use Segments	170	170
Fluid Constants:	K_P	12.0	12.0	12.0	12.0
	K_I	1.5	1.5	1.5	1.5
	K_D	0.05	0.05	0.05	0.05
	Dead Time (sec) θ		21	16	
Mold Dynamic Constants:	Time Constant (sec) τ		47	32	
	Closed Loop Time Constant (sec) λ		47	47	
	Entered Gain Default: 0.33 K'		0.3300	0.3300	
	Steady State Gain K_m		0.9960	0.9890	

Run Metal Tuning Test to Establish Metal Constants

The *Recipe Editing – Rec PID/MTL Constants* screen will work in unison with the *Recipe Editing – Recipe Segments* screen. This screen will allow the user to change the control mode, constant temperature set point if applicable, PID gains, mold dynamic constants that can be defined for up to four zones. This screen also provides an avenue to access the *MTL Tuning* screen.

DESIGN AND MAINTENANCE DOCUMENT

MHC PP OI Controls Upgrade

Manual Constant Temperature

NON-RECIPE CONSTANT TEMPERATURE CONTROL, FLUID OR MOLD						
		Zone 1	Zone 2	Zone 3	Zone 4	
Control Mode		Fluid Const Temp	Fluid Const Temp	Mold Const Temp	Mold Const Temp	
Const Temper SP		220	80	80	65	
Mold Temp		TC OK 80.8	TC OK 78.7	TC OK 78.4	TC OK 79.4	
Fluid Temp		TC OK 231.3	TC OK 79.9	TC OK 80.5	TC OK 77.1	
Valve Output %		88	0	50	82	
PID Consts Defined		YES	NO	NO	YES	
Fluid Consts:	PID Default1 K_P	1.5	0.0	0.0	12.0	
	PID Default2 K_I	4.5	3.0	3.0	1.5	
	K_D	0.05	0.00	0.00	0.05	
MTL Consts Defined				NO	YES	
Metal Dynamic Constants:	Dead Time (sec) θ			0	13	
	Time Constant (sec) τ			0	27	
	Closed Loop Time Constant (sec) λ			0	27	
	Entered Gain K' <small>Default: 0.33</small>			0.0000	0.3300	
	Steady State Gain K_m			0.0000	0.9940	
	Calculated Gain K_{cs}			0.0000	8.0394	
	P_0			0.0000	0.3352	
	P_1			0.0000	0.2453	
	P_5			0.0000	0.6905	
	P_8			0.0000	0.2284	
P_9			0.0000	0.0811		
Run Metal Tuning Test to Establish Metal Constants						

Zone 3 is under PID control until the Mold Constants are fully defined.

The *Manual Constant Temperature* screen is available to the Operator; however, its appearance differs depending on whether the Operator or Engineer is logged in. This screen is used when it is not desirable to run a pre-configured recipe, or when developing a recipe. The manual nature of this mode means that the set point must be changed manually. The set point will stay fixed until changed again, and it cannot be ramped. The control mode and the control constants can also be modified.

DESIGN AND MAINTENANCE DOCUMENT

MHC PP OI Controls Upgrade

Event Viewer

EVENT VIEWER							
Event Time	Type	Message	Tag Name	Value	Previous	User	Comment
02/25/2009 09:55:50	Display	Open Screen:events.scr		0	0	Engineer	
02/25/2009 09:55:50	Display	Close Screen:recipe editor-constants.scr		0	0	Engineer	
02/25/2009 09:55:50	Display	Close Screen:tempclear.scr		0	0	Engineer	
02/25/2009 09:55:50	Display	Open Screen:tempclear.scr		0	0	Engineer	
02/25/2009 09:55:25	Tags Ch...	Z3 Const Temper SP changed	hnSPTemperConstZ[3]	170	50	Engineer	
02/25/2009 09:55:25	Tags Ch...	ControlMode set by Static Tag Integer: 0=CV, 1=PL...	hmControlModeZ[2]	2	1	Engineer	
02/25/2009 09:55:25	Tags Ch...	ControlMode set by Static Tag Integer: 0=CV, 1=PL...	hmControlModeZ[1]	2	1	Engineer	
02/25/2009 09:55:25	Tags Ch...	Z4 Const Temper SP changed	hnSPTemperConstZ[4]	170	50	Engineer	
02/25/2009 09:55:25	Tags Ch...	Z4 Const Temper SP changed	hnSPTemperConstZ[4]	50	170	Engineer	
02/25/2009 09:55:25	Tags Ch...	Z3 Const Temper SP changed	hnSPTemperConstZ[3]	50	170	Engineer	
02/25/2009 09:55:25	Tags Ch...	ControlMode set by Static Tag Integer: 0=CV, 1=PL...	hmControlModeZ[2]	1	2	Engineer	
02/25/2009 09:55:25	Tags Ch...	ControlMode set by Static Tag Integer: 0=CV, 1=PL...	hmControlModeZ[1]	1	2	Engineer	
02/25/2009 09:55:25	Display	Open Screen:recipe editor-constants.scr		0	0	Engineer	
02/25/2009 09:55:24	Display	Close Screen:tempclear.scr		0	0	Engineer	
02/25/2009 09:55:24	Display	Close Screen:recipe editor-segments.scr		0	0	Engineer	
02/25/2009 09:55:24	Display	Open Screen:tempclear.scr		0	0	Engineer	
02/25/2009 09:54:28	Security ...	LogOn		0	0	Engineer	
02/25/2009 09:54:10	Security ...	Invalid attempt to log:		0	0	pag	
02/25/2009 09:54:02	Tags Ch...	Z3 Const Temper SP changed	hnSPTemperConstZ[3]	170	50	pag	
02/25/2009 09:54:02	Tags Ch...	ControlMode set by Static Tag Integer: 0=CV, 1=PL...	hmControlModeZ[2]	2	1	pag	
02/25/2009 09:54:02	Tags Ch...	ControlMode set by Static Tag Integer: 0=CV, 1=PL...	hmControlModeZ[1]	2	1	pag	
02/25/2009 09:54:02	Tags Ch...	Z4 Const Temper SP changed	hnSPTemperConstZ[4]	170	50	pag	
02/25/2009 09:54:02	Tags Ch...	Z4 Const Temper SP changed	hnSPTemperConstZ[4]	50	170	pag	
02/25/2009 09:54:02	Tags Ch...	Z3 Const Temper SP changed	hnSPTemperConstZ[3]	50	170	pag	
02/25/2009 09:54:02	Tags Ch...	Z2 Const Temper SP changed	hnSPTemperConstZ[2]	50	200	pag	
02/25/2009 09:54:02	Tags Ch...	Z1 Const Temper SP changed	hnSPTemperConstZ[1]	50	100	pag	
02/25/2009 09:54:02	Display	Open Screen:recipe editor-segments.scr		0	0	pag	
02/25/2009 09:54:02	Display	Close Screen:tempclear.scr		0	0	pag	
02/25/2009 09:54:02	Display	Open Screen:tempclear.scr		0	0	pag	
02/25/2009 09:54:01	Display	Close Screen:metal tuning.scr		0	0	pag	
02/25/2009 09:53:19	Display	Open Screen:metal tuning.scr		0	0	pag	
02/25/2009 09:53:19	Display	Close Screen:metal tuning.scr		0	0	pag	
02/25/2009 09:50:19	Display	Open Screen:metal tuning.scr		0	0	pag	
02/25/2009 09:50:19	Display	Close Screen:tempclear.scr		0	0	pag	
02/25/2009 09:50:19	Display	Close Screen:non-recipe consttemper.scr		0	0	pag	
02/25/2009 09:50:19	Display	Open Screen:tempclear.scr		0	0	pag	
02/25/2009 09:50:04	Display	Close Screen:splash.scr		0	0	pag	
02/25/2009 09:50:03	Security ...	LogOn		0	0	PAG	
02/25/2009 09:50:01	Display	Open Screen:splash.scr		0	0	Operator	
02/25/2009 09:49:25	Display	Open Screen:non-recipe consttemper.scr		0	0	Operator	
02/25/2009 09:49:25	Display	Close Screen:tempclear.scr		0	0	Operator	
02/25/2009 09:49:25	Display	Open Screen:tempclear.scr		0	0	Operator	
02/25/2009 09:49:23	Display	Close Screen:non-recipe consttemper.scr		0	0	Operator	
02/25/2009 09:49:23	Display	Close Screen:tempclear.scr		0	0	Operator	
02/25/2009 09:49:23	Display	Open Screen:tempclear.scr		0	0	Operator	
02/25/2009 09:49:23	Security ...	LogOn		0	0	Operator	
02/25/2009 09:48:42	Display	Open Screen:non-recipe consttemper.scr		0	0	PAG	
02/25/2009 09:48:41	Display	Close Screen:tempclear.scr		0	0	PAG	
02/25/2009 09:48:41	Display	Close Screen:trendhistorical.scr		0	0	PAG	
02/25/2009 09:48:41	Display	Open Screen:tempclear.scr		0	0	PAG	
02/25/2009 09:48:39	Display	Open Screen:trendhistorical.scr		0	0	PAG	
02/25/2009 09:48:39	Display	Close Screen:trend.scr		0	0	PAG	
02/25/2009 09:48:39	Display	Close Screen:trendhistorical.scr		0	0	PAG	

The *Event Viewer* screen displays the history of what screens the user has visited, what variables the user changed, custom messages, and login/logoff activity. This information may be helpful in diagnosing an unexpected condition.

DESIGN AND MAINTENANCE DOCUMENT

MHC PP OI Controls Upgrade

PAG Utilities

Process Automation Group (PAG) Utilities -- software developed by Josh Salmond, 2008

WinPLC Station: MHCWinPLC		PLC state is 4 zone		MTL Sample Time: 10 [sec]		PID Sample Time: 2 [sec]	
PLC Peak I/O Time (ms)	8	<div style="display: flex; justify-content: space-around;"> <div>Two Zone Machine</div> <div>Valve Position Z1 [% open]</div> <div>Valve Position Z2 [% open]</div> <div>Valve Position Z3 [% open]</div> <div>Valve Position Z4 [% open]</div> </div>					
PLC Average Logic Solve Time (ms)	16						
PLC Peak Logic Solve Time (ms)	27						
PLC Average Scan Interval (ms)	49						
PLC Peak Scan Interval (ms)	52						
OverScans / ScanCount	0.035 %						
PLC Number of OverScans	55						
PLC Scan Count	158033						
<div>MsgBox to display PLC calendar and clock values</div> <div>DailyRunScript</div>							
Reserved10 Status: (32 bit integer) b3 b2 b1 b0 0 0 0 0 b3 hi=battery missing or dead or low b2 hi=Flag forces b1 hi=Input forces b0 hi=Output forces							
Put editable HMI Heartbeat here if have comm timeout problems....							
Computer IP Address: 10.0.0.11 MAC ID: # found: 1 ID: 00:08:9B:B3:52:5D MAC Adapter: Generic Marvell Yukon Chipset based Gigabit Ethernet Controller Computer Name: OEMCOMPUTER c:\ GB free: 67.7 m:\ GB free: 74.3 (MHC_Backup) Operating System: 2 (2=>Win 2000/XP/Vista) IWS Product Path: C:\Program Files\InduSoft Web Studio v6.1\IWS IWS Product Version: 6.000000 ??this should be 6.1/doesn't work at runtime?? IWS Product Version: v6.1+SP4, Build 1042 (ProductPath\BIN\Program Settings.ini) IWS Application Path: C:\Documents and Settings\MHCoper\MoldHeatingAndCooling\HMI\IWS_v3.1.0\IWS IWS HardKey Model: Local Interface - CNF IWS HardKey S/N: 2737.88							

Slot0 D-In

3 0 4

D2-08ND3

Slot1 D-Out

3 0 4

D2-08TR

Slot2 A-Out

16-28 4VDC 8 CHANNEL

F2-08DA-1

Slot3 Fluid

F2-04THM

Slot4 Mold

F2-04THM

Slot5 Central

F2-04THM

Showing actual PLC value as read by the module.
Actual temperature = (PLC value)/10 [degF]

CH1+	CH1	CH2+	CH2	CH3+	CH3	CH4+	CH4
Z1: 1181	Z1: 633	Z2: 705	Z2: 630	Z3: 671	Z3: 676	Z4: 681	Z4: 680

mapped to PLC state set, status bit, & 2/4zone switch
J-type TC: '+'=white wire, '-'=red wire

Calculated constants when using Recipe MTL; was previously on Rec-Const screen

Calculated Gain	K _{cs}	P0	P1	P5	P8	P9
Show calculated values						

These values are only accurate after the current recipe has been downloaded.

Exit MHC to Windows

The *PAG Utilities* screen provides a peek into the WinPLC performance and status, the PC system information, and calculated MTL constants. The screen also provides PAG users the ability to stroke the control valves directly by changing their % open position. This screen also provides an exit feature out of the HMI to diagnose PC problems as they arise.

3.2.3. PLC Software Functional Requirements

This section defines the specific software functional requirements of the controller software for handling control logic, continuous control, and communication with the HMI.

The controller performs both discrete and continuous operations and handles all I/O. Discrete control handles execution flow by looking at digital inputs and setting digital outputs based on input conditions. Continuous control drives the process variable to the requested set-points by running analog process error through control algorithms and adjusting analog outputs to drive control elements. The PLC will execute the recipe segments when a recipe is started.

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

Control Logic

The control logic will be designed using state machine logic. The state machines will be derived from current modes and methods of operation. See section 4.0 *State Sets* for more information about state sets. BDX-613-3181⁸ was referred to extensively to ensure that the new controller was programmed similar to the existing controller as much as possible.

Continuous Control

The controller has two modes of continuous control: the first mode being fluid PID; the second mode being Metal Temperature control^{5,6}. Fluid PID uses a standard form of the discrete PID equation:

$$F_n = K_p \left[e_n + \frac{K_I T_1}{60} \sum_{i=1}^n e_i + \frac{60 K_D}{T_1} (e_n - e_{n-1}) \right]$$

where:

- F_n → PID controller output at the nth sample period to the control valve position as a percentage
- e_n → PID error at the nth sample period = Set-Point minus Process Value (SP-PV), as a percentage of range
- T_1 → PID sample rate in seconds (fixed at 2 seconds)
- K_P → proportional gain constant, unit-less
- K_I → integral gain constant, units of rate/minute
- K_D → derivative gain constant in minutes

The PID equation's output is a percentage (unit-less). Therefore, the errors are converted to percent of range, thereby making the equation unit-less, before calculating the output. The constant 60 is needed to convert T_1 from units of seconds to minutes. K_I has units of rate/minute; K_D has units of minutes.

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

Metal Temperature control uses a special control algorithm developed by Dick Floersch^{5,6,7}. Some subscripts and variable notation were changed from the previously published algorithm to provide clarity. When using the MTL mode of automatic control, the MTL and PID algorithm are combined in a cascaded control loop as illustrated in *Figure 13*.

Figure 13: Cascaded Control Loop Block Diagram

The special Metal Temperature control law equation is as follows:

$$M_m = K' K_{CS} (e_m - P_0 e_{m-1} - P_1 e_{m-2}) + P_5 M_{m-1} + P_8 M_{m-N-1} + P_9 M_{m-N-2}$$

where:

$$K' = K_{entered} = \text{default } 0.33; \text{ all current recipes use } 0.33.$$

$$K_{CS} = \frac{\tau_m (1 - e^{-fT/\lambda})}{K_m e^{-fT/\tau_m}} \quad K_m = \frac{\text{MetalTemperature}}{\text{FluidTemperature}} @ 5 * \tau_m$$

$$P_0 = e^{-T/\tau_m} - \frac{e^{-fT/\lambda} - e^{-T/\lambda}}{1 - e^{-fT/\lambda}} \quad P_1 = \frac{e^{-T/\tau_m} (e^{-fT/\lambda} - e^{-T/\lambda})}{1 - e^{-fT/\lambda}}$$

$$P_5 = e^{-T/\lambda} \quad P_8 = 1 - e^{-fT/\lambda} \quad P_9 = e^{-fT/\lambda} - e^{-T/\lambda}$$

$$N = \frac{\theta}{T}, \text{ integer value} \quad f = N + 1 - \frac{\theta}{T}, \text{ fractional decimal value}$$

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

M_m	→	metal controller output at the m^{th} sample period to the control valve position as a percentage; M_{m-1} is the metal controller output one sample period previous, etc.
e_m	→	metal error at m^{th} sample period = SP-PV, as a percentage of range; e_{m-1} is the metal error one sample period previous, etc.
λ	→	Lambda, specified closed loop time constant in seconds
τ_m	→	Tau, mold metal temperature (plant) time constant in seconds, defined as the elapsed time, after dead time, for the response to reach 63.2% of the input
K_m	→	mold metal temperature (plant) steady state gain. It should be noted that a variation of the calculation of this gain is the substitution of the Fluid SP for the Fluid Temperature; however, at five times the mold time constant, these two numbers are nearly identical.
θ	→	Theta, dead time in seconds, defined as the elapsed time before the sensing element sees a response from the input
T	→	Metal sample rate (sample period) in seconds, also called T_2 (fixed at 10 seconds)
N	→	truncated integer value of dead time divided by sample period (example: if $\theta = 77$, and $T = 10$, then $N = 77/10 = 7.7 = 7$)
f	→	fractional position (decimal value) of the dead time between periods (example: if $\theta = 77$, and $T = 10$, then $f = 7 + 1 - 7.7 = 0.3$)

The MTL equation's output is a percentage. Therefore, the errors are converted to percent of range, thereby making the equation unit-less, before calculating the output.

Human Machine Interface Communication

The controller and HMI must communicate discrete events, numeric values, and recipe data. OPC using Ethernet cabling was the obvious choice to enable the exchange of variables between the controller and HMI as their manufacturers were different. The HMI manufacturer, InduSoft, had an outdated and unsupported Think&Do driver that no longer functioned. Building a custom communication driver was not considered, as it would be time consuming and costly.

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

3.3. Hardware Functional Requirements

This section defines the hardware functional requirements for the MHC PP OI Controls Upgrade. Hardware requirements for both HMI and PLC will be addressed. The hardware will reside in a control cabinet next to each machine that uses the Mold Heating and Cooling Central System. New cabinets will be fabricated.

3.3.1. HMI Hardware Functions

The touch screen and computer will be industrial grade COTS. The computer will have backup capability in the form of an additional hard drive. The computer will have standard removable media read/write devices.

3.3.2. PLC Hardware Functions

The PLC form factor will be modular, expandable, and COTS. The PLC will manage all I/O. The replacement I/O is required to handle the signals listed in *Table 2*. More I/O may be added as necessary.

I/O Type	Qty	Interface	Range
Analog Inputs	4	Fluid Thermocouples	0-600 deg F
Analog Inputs	4	Mold Thermocouples	0-600 deg F
Analog Inputs	4	Central System Thermocouples	0-600 deg F
Analog Outputs	4	I/P Valve Controller	4-20 mA
Digital Inputs	2	Relay from Machine, PP ESTOP	On/Off
Digital Outputs	1	Press Inhibit	On/Off

Table 2: MHC PP OI Controls I/O Requirements

The Central System Thermocouples have been included in the design, but may not be implemented in the field. The thermocouples consist of Hot Supply, Hot Return, Cold Supply, and Cold Return. This would be useful information to have; however, it would require a major piping modification to include these items.

The Press Inhibit was considered in the design and a placeholder made; however, it was not implemented. Excerpt from BID-A461⁵:

Two relays are used in the design, one 5 VDC and the other a 110 VAC relay. The 5 VDC relay is used to activate the 110 VAC relay. The 5 V relay is activated by one bit of the 8255 output port. The 110 V relay is used to break the control power to the injection or compression machine preventing the press from operating when the relay is activated. During execution of a certain firmware code, activation of the press cycle could cause perturbations that would upset the process yielding unacceptable parts. To prevent that

DESIGN AND MAINTENANCE DOCUMENT

MHC PP OI Controls Upgrade

occurrence, the press inhibit relay is activated which keeps the press from closing until the relay is deactivated.

Other documentation of the Press Inhibit feature was fairly sparse, and seemed to only be implemented on injection molding machines. Therefore, since injection processes have been outsourced; and even more so, the need to have MHC PP OI on special injections processes has been discontinued, the need for Press Inhibit seemed superfluous. After reviewing drawings and actual wiring from the existing 750 ton compression press, the Press Inhibit was not being used. Press Inhibit is referenced in the code for Demand Functions 3, 4, 5, 10, 12, and 13. Currently the press is inhibited if control is 'inside' one of these Demand Functions, and is released once the Demand Function has been resolved. A good use for Press Inhibit might be during Preheat if Mold Preheat Temperature Limits are specified, preventing electrical signals for mold closure until the requirement is satisfied.

3.4. Performance Requirements

The only known performance requirements are that the PID control loop shall execute every 2 seconds, and the MTL control loop shall execute every 10 seconds.

A self imposed performance requirement was to minimize recipe download time from the HMI to the WinPLC. A significant amount of time was spent on this effort, reducing the initial transfer time from over 3 minutes to less than 30 seconds. Up to 400 variables can constitute a recipe.

3.5. Design Constraints

The MHC PP OI will be modular to the MHC PP OI Central System. Any number of machines may be added up to the capacity of the central system. The design will be portable to any other central system configuration that may be added or changed in the future. All the functionality of the current system will be accounted for in the new system. The control algorithms and their sample periods will remain unchanged.

3.6. Attributes

There are no special circumstances that will necessitate the need for additional software requirements; therefore, this section does not apply.

3.7. External Interface Requirements

The user external interface will be the touch screen HMI. The need for a keyboard and mouse should only be for administrator use.

The hardware interface is a standard stand-alone Ethernet network via a hub. The PC and WinPLC are connected to and communicate via the hub. A laptop may be connected to the hub to download new software to the WinPLC and to debug the WinPLC code real-time. A DVD, floppy, and flash drive are available on the PC.

DESIGN AND MAINTENANCE DOCUMENT

MHC PP OI Controls Upgrade

Thermocouple connections may be modified by the user without having to open the electrical cabinet. A thermocouple plug and socket panel is available on the side of the cabinet and can be reconfigured as necessary.

3.8. Hardware and Software Installation

See *Appendix B: Hardware & Software Installation Guide* for detailed instructions on how to setup the hardware and software for the system.

4.0 State Sets

This section documents the state sets that exist for the PLC software. A complete listing of the state sets can be found in *Appendix D: WinPLC Code Listing*, in the section labeled “State Design”. The state machine was developed with Xilinx StateCAD Release 7.1i software. Refer to *Appendix C* for documentation on converting the StateCAD design to Think&Do.

5.0 Lessons Learned

This section is meant to document lessons learned during development in an effort to prevent a repeat occurrence of these topics in the future.

5.1. Entity Think & Do Errors

Phoenix-Contact purchased Entity. A contact named Mike Schoonmaker, 800-722-6875, mschoonmaker@phoenixcon.com, was very helpful and knowledgeable of the Think&Do products.

5.1.1. No Default Printer Defined

An error message box would popup when a tag cross reference report was run with a caption “No Default Printer Defined.” The problem lies with Crystal Reports which is added on as the report generator for Think&Do. Files ‘CS_DT.RPT’ and ‘CS_FC.RPT’ must be placed in the \THINKNDO\BIN\ directory. See also <http://support.entity.com AnswerID 682>.

5.1.2. Socket Timeout

Communication of the PLC programming computer with the WinPLC would break during the download of a newly compiled runtime software version. The following message boxes would appear.

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

To fix the connection, use a registry editor and search for "SocketTimeout". Increase the time of keys "SocketTimeoutReceive" and "SocketTimeoutSend" from the default of 5 (decimal value). A time of 30 (decimal) was used with success.

5.2. Entity DataPack v7.3

Entity Think&Do is the development environment software that is used to write and debug the WinPLC code. Entity DataPack is the runtime software used to execute the developed software from Think&Do. DataPack provides a WinPLC interface and an OPC runtime server with unlimited installs.

DataPack was purchased from AutomationDirect.com, part number PC-ENT-DAP. The supplied software was version 7.2. Version 7.2 was not able to communicate over the OPC network. After consultation with Phoenix-Contact technical support, it was discovered that v7.2 did indeed have OPC communication issues. Phoenix-Contact had already released v7.3, but the supplier, Automation Direct, was years late in obtaining the latest release. Version 7.3 fixed all OPC communication problems.

DataPack does not allow the use of the debugging application AppTracker. Therefore, an Ethernet hub was used to allow a laptop with Think&Do installed to connect to the network to enable debugging of the WinPLC code as necessary.

5.3. Windows Firewall

The Windows Firewall must be turned OFF in order for the programming PC and the HMI PC to connect with the WinPLC on initial setup. Limited testing was performed to try to turn on the firewall after initial setup, with seemingly no problems. However, since all development and implementation was done with the firewall off, this is the

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

recommended setup. Phoenix-Contact recommends keeping the Windows Firewall turned off.

5.4. Temperature as Percent of Range

In order to make the PID Algorithm calculations unit-less, and the ensuing output in the range of 0-100%, the measured variable, temperature, was converted to a percent of range. The range chosen for the 1980's implementation of the MHC PP OI was 0%=32°F and 100%=700°F. To convert a temperature to percent range, the linear equation must be derived, $y=mx+b$. The slope is $(100-0\%)/(700-32^\circ\text{F})$. Using one of the two known endpoints to solve for the b intercept, one finds that $b=-4.79$. Therefore the equation to convert a given temperature to percent range is

$$\text{temperature} _ \%R = \frac{100}{668} * ^\circ F - 4.79$$

The MTL Algorithm output was capped at 1400°F to -700°F, correlating to a percent range output of 204.79% to -109.58%. The new implementation caps the MTL output to 200 to -100% for simplicity.

Note that when calculating error, SP-PV, the b intercept cancels out. Therefore, the equation for error, calculated as percent range, can be abbreviated to

$$\text{error} _ \%R = \frac{100}{668} * (SP - PV)$$

5.5. Preloading the Integrator in the PID Algorithm

Manual control valve mode is only available to PAG users. In this mode, the valve's percent open is directly controlled. During development, when changing from manual control value control to PID control, if K_p was set at zero, the PID output would be zero. The trick is in the math. The textbook illustration of the PID equation is

$$F_n = K_p \left[e_n + \frac{K_I T_1}{60} \sum_{i=1}^n e_i + \frac{60 K_D}{T_1} (e_n - e_{n-1}) \right]$$

In the textbook illustration, it is easy to see that if K_p was zero, the entire output would be zero. However, the implemented PID equation is

$$F_n = K_p e_n + \sum_{i=1}^n \frac{K_p K_I T_1}{60} e_i + \frac{60 K_p K_D}{T_1} (e_n - e_{n-1})$$

Notice that K_p has been multiplied through the equation. The other subtlety, which is not evident in the above equation, is how the integrator is actually calculated.

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

$$\text{Integrator_term} = \sum_{i=1}^n \frac{K_p K_I T_1}{60} e_i \rightarrow \text{Integrator} = \text{Integrator} + \frac{K_p K_I T_1}{60} e_n$$

When changing from manual control value control to PID control, the Integrator term is set to the current value of the control valve percent open. Therefore, to enable a bumpless transfer from control valve control to PID control, the first iteration of the Integrator is:

$$\text{Integrator} = CV\% + \frac{K_p K_I T_1}{60} e_n$$

Rewritten, the implemented PID equation is

$$F_n = K_p e_n + \left(CV\% + \frac{K_p K_I T_1}{60} e_n \right) + \frac{60 K_p K_D}{T_1} (e_n - e_{n-1})$$

Therefore, in the special case where K_p , K_I and K_D could be set to zero, or the error is zero, the output will be:

$$F_n = 0 + (CV\% + 0) + 0$$

This creates a truly bumpless transfer of control.

5.6. Preloading the MTL Pipeline

In the same way the PID Algorithm is preloaded, the MTL algorithm's initialization must also be considered when transferring to MTL control. Remember that the MTL Algorithm equation is

$$M_m = K K_{CS} (e_m - P_0 e_{m-1} - P_1 e_{m-2}) + P_5 M_{m-1} + P_8 M_{m-N-1} + P_9 M_{m-N-2}$$

The previous outputs, M_{m-x} , are stored in an array, or pipeline, so that the correct value, as determined by subscript N , can be retrieved. But if transferring to MTL control, the previous MTL outputs may be undefined, or held in memory from a previous use of the algorithm.

To preload the MTL Algorithm, the previous MTL outputs are populated with the current fluid temperature as percent range. The fluid temperature is used, rather than the metal temperature, because it is assumed that is the temperature the user wants the mold to reach.

The HMI was written such that the MTL constants must be defined before allowing a transfer to MTL control. On initial transfer to the MTL Algorithm, the error should be zero. However, populating the previous MTL outputs will prevent the output of the MTL Algorithm from going to zero. Remember that the MTL output becomes the set point for

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

the PID Algorithm, so on initial transfer, if the error is zero, the MTL output should closely reflect the current fluid temperature (depending on the value of the calculated constants P_5 , P_8 , and P_9), thereby effecting a bumpless transfer of control.

5.7. Algorithm Calculations during Ramp Segments

During development it seemed desirable to turn off the I and D portions of the PID control during a ramp segment, due to the constant changing nature of the set point. The original PID controller calculated all terms of the PID during a ramp segment. Bench tests showed that proportional only control could not keep up with the ramp. It was decided to write in a programming switch to enable or disable this feature, to allow for further testing when the software was commissioned on a machine.

The MTL Algorithm is unchanged during a ramp segment.

5.8. Windows XP User Privileges

There are two Windows users available when the computer is powered up: MHCOper and MHCAdmin.

MHCOper is used for production, engineering, or any other required access group. It does not require a password to login.

During development MHCOper was setup with XP User privileges. However, it was discovered that by default, an XP User was not allowed OPC communication rights without making custom modifications. Therefore, MHCOper was given XP Power User rights, rather than trying to setup custom controls for an XP User.

MHCAdmin has XP Administrator rights. PAG personnel are the only users that have access to this password protected login.

5.9. OPC Communication

After dealing with many issues on when, how, and why OPC variables are updated, it was determined that a different approach to OPC might be taken on future projects using Think&Do and InduSoft. Some of the issues could be due to the lack of configurability of the Think&Do OPC Server. The following shows how a PLC tag is currently updated from a HMI tag change:

- the HMI tag value is changed
- the OPC configuration sheet in InduSoft links the HMI tag name to the PLC tag name
- the OPC Server/Client relationship updates the PLC tag value

To better control variables, or tags, an alternate method would be to create HMI tags, PLC tags, and OPC tags, when using the Think&Do and InduSoft combination. The OPC tags would be the link between the HMI and PLC, rather than having the OPC Server and Client directly change HMI and PLC tags. An example of how a PLC tag would be updated from the HMI tag change might be as follows:

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

- the HMI tag value is changed
- a routine updates the value of the HMI-OPC tag from the HMI tag value
- the OPC configuration sheet in InduSoft links the HMI-OPC tag name to the PLC-OPC tag name
- the OPC Server/Client relationship would update the PLC-OPC tag value
- the PLC would run a routine and see a difference in the PLC tag versus the PLC-OPC copy of the tag
- the PLC would update the value of the tag from the PLC-OPC tag accordingly
- handshake the change if necessary

The disadvantages of this method are duplicate tags, which may nearly double the tag count needed on both the HMI and PLC licenses, and slower update rates. The tag count issue could cause a problem if the software license is based on how many tags are used, as the InduSoft licenses are. Advantages are that the HMI and PLC logic are kept separate, and the programmer controls when tag updates occur.

The interaction of Windows Firewall with OPC doesn't seem to be a problem; limited testing was done after development and implementation. Since limited testing was performed, it is recommended to keep the firewall off.

5.10. Segment Timer Accuracy Test

A 24 hour test was performed to test the accuracy of the segment timers. Zone 1 was setup with a 0 second segment 0, a 0 second segment 1, and a 24 hour segment 2. A special chart was constructed using the Think&Do system variable RollingMsecCounter, the starting value was recorded at cycle start (777170), and the ending value recorded at cycle completion (87177007). Cycle start was defined as being in states sRecipeActive and sSpSegLoadedZ1. Cycle completion was defined as state sSegTimerDoneZ1.

For the 24 hour test, the number of milliseconds recorded was 86399837, which equates to 23:59:59.837, a shortage of 163 milliseconds in 24 hours, and an error of 0.0002%. The average scan time of the PLC code is 50 milliseconds, the maximum scan time can be as high as 70 milliseconds; therefore, this timing error is in the range of 2-3 scan times for a 24 hour period. This error was deemed acceptable by the developer.

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

6.0 Design Change Log

This section will document changes in the design requirements of the software. The following paragraphs will contain a high-level summary of the change. A more detailed description of the change can be found in the Global Procedures interface of InduSoft for HMI changes, and on the Documentation flowchart of Think&Do for the WinPLC changes.

6.1. HMI v3.0.0 and PLC v3.0.0, 31-Oct-2008

This version was the initial release of the software. It was a pre-production version used to demonstrate the capabilities of the system. The end user recommended several changes to implement for the production version. Several programming methodologies also needed to be fixed to increase robustness.

6.2. HMI v3.1.0, 08-Apr-2009

This version was extensive. This version fixed many runtime issues with the different modules of InduSoft. This version changed many screen graphics. Help documentation was added. The tag license was upgraded from 1500 to 4000.

6.3. PLC v3.0.1, 08-Apr-2009

This version addressed minor bugs from the initial release, and added code documentation to charts. Added E-STOP functionality. Corrected control mode change from PID to MTL to stay in PID until MTL constants are defined. Created new chart to update the PLC clock.

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

Appendix A: Software Operating Requirements

Appendix A: Software Operating Requirements

Mold Heating and Cooling Pump Package Operator Interface (MHC PP OI) software operating requirements, REVISION 6

Derived from BDX-613-3181, BID-A461, and current system operation.

- A programmable logic controller (PLC) will be executing the recipe, controlling the fluid valves, reading the thermocouples, etc
- A personal computer (PC) will be running the human-machine interface (HMI) software and a touch-screen user interface panel. The PC will store the recipe files as a text file.
 - HMI functions:
 - Exiting the HMI will logoff the current Windows user.
 - Operator access: StartRecipe_PB, Cycle Abort, select recipe file...
 - Engineer access: In addition to Operator access, Recipe profile manipulation, and Recorded Events.
 - PAG access: PAG Utilities screen, plus all of above.
- OPC communication via Ethernet will be used to synchronize the PC and PLC
- Relationship of MHC PP OI to the machine Press:
 - Cycle time on the Press is independent of cycle time on MHC PP OI
 - The Press will have a relay (MHCRelay), that when energized, signals MHC PP OI to start its recipe (start Seg2). DryCycle_SS will bypass MHCRelay.
 - Example: When the two PBs on the Press are energized to advance the press off of its home position limit switch to close the mold, the relay will be energized.
- There will be two distinct modes of controlling mold temperatures
 - Automatic; which will be called "Recipe"
 - All four zones must be accounted for when running a recipe. The zones must be controlled by Constant Temperature, or by a Temperature Profile. The only user interaction with a running recipe is to change the SP of a zone controlled by Const Temper, or to Abort. If a zone is not going to be used, it still must be defined.
 - Manual; which will be called "Non-Recipe Constant Temperature"
 - Zones can be controlled independently by specifying a temperature SP and whether the SP will be using the PID or MTL algorithm. PID/MTL constants will have to be defined; default PID constants will be available.
 - PAG members will also have the option to operate the Control Valve directly.

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

Appendix A: Software Operating Requirements

- Segments 0, 1, and 2 must be defined, for all zones, before a recipe can be activated and downloaded to the PLC (Setting a zone to CT, will auto define these segments)
 - Segment 0 and 1 are the Preheat Segments
 - Segment 0 is a dwell only segment (no ramp can be specified)
 - There are two Preheat modes: Preheat (now will be called ManualPreheat) and Auto Wraparound (now will be called AutoPreheat)
 - ManualPreheat requires user interaction to start segment 0 via the StartRecipe_PB (start segment 0 timer). (Usually used for compression parts.)
 - AutoPreheat is an automatic type of preheat such that when all zone's have completed their last segment, the recipe automatically “wraps around” and executes Seg0 and Seg1. (Usually on injection parts.)
 - SkipPreheat is a one-shot question that affects only the first part of the production run (same functionality of DF4), or after a Cycle Abort and applies to all Zones
 - SkipPreheat will actually skip to the end of segment 1 (will use Seg1 SP and Seg1 timer will be expired) and will wait for DryCycle|MHCRelay Ready for Cycle Start (RFCS) and Metal Temp Limits are not affected by SkipPreheat. These requirements must still be satisfied before proceeding to segment 2.
 - Segments 2 through last are the “production” cycle segments
 - Plastic part material is usually introduced to the press at the start of segment 2

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

Appendix A: Software Operating Requirements

- Zone recipe operations/segments will execute independent of one another, except for:
 - A recipe will start when:
 - All 4 zones must be accounted for, by specifying control method – PID Profile or Const Temp, or MTL Profile or Const Temp. PID constants must be defined for every zone, MTL constants need to be defined for zones using the MTL algorithm.
 - StartRecipe_PB is pressed and all zones are IDLE
 - StartRecipe_PB will be disabled if the recipe in HMI memory has been changed (set flag to signal a change has been made, reset flag after recipe has been saved). Changes can be made during operation. This button will only be available when AllZonesIDLE.
 - If the recipe in HMI memory has been changed, the user must either resave the recipe in HMI memory to the same file, or save the HMI recipe as a new file, before the recipe can be executed. File manipulation only with engineer permissions.
 - Log when cycle has been started, completed, aborted, and which recipe filename was used.
 - All zones (with recipes) must complete segments 0 and 1, before proceeding to segment 2
 - Starting segment 0, with All zones IDLE, will be either:
 - Pressing the StartRecipe_PB if ManualPreheat_SS is selected (this serves the same function as the old Init_Preheat_PB)
 - Pressing the StartRecipe_PB if AutoPreheat_SS is selected
 - Automatic if AutoPreheat_SS is selected, and after a previous cycle has completed
 - Proceeding to segment 2 will be:
 - All zones RFCS
 - Metal temperature limits are within those specified (DF9). This is just an indicator in the old system; it will now be a condition before allowing segment 2 to begin.
 - If none are specified, then ignore this requirement and set RFCS indicator.
 - Limits can be specified regardless of PID or MTL control.
 - These limits correspond with segment 1. When the defined temperature limit is reached during segment 1, the RFCS indicator will be true.
 - When all zones are within their metal temperature limits, the Ready for Cycle Start (RFCS) indicator will illuminate allowing the press to close when initiated by the operator. Which is, press must activate MHCRelay, or the operator must press the DryCycle_SS.
 - MHCRelay OR DryCycle_SS
 - All zones (with recipes) will wait at their last segment, until all zones have completed their last segment
 - If AutoPreheat_SS, then segments 0 and 1 will automatically execute for all zones, and the zones will wait at the end of segment 1 until the MHCRelay is energized or DryCycle_SS is selected.

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

Appendix A: Software Operating Requirements

- If ManualPreheat_SS, then segment 0 will not start until StartRecipe_PB is pressed.
- When CycleAbort_SS is selected:
 - Log event.
 - IDLE state is set.
 - Cycle Abort is only effective/available when a recipe is running.
 - If Cycle Abort temperature set points (zone based) have been specified (DF8), the zones will aim to achieve this temperature; otherwise zones will jump to the temperature defined in the zone's last segment. This will move the controller to PID or MTL control with a constant temperature set point; control will match that of the recipe. (Example: if a recipe has Z1&Z4 at CT, Z2 defined as PID, and Z3 as MTL, the Cycle Abort temperature limits – whether they are defined, or use the last segment's SP – will be PID control for Z2, MTL control for Z3, and have no affect on Z1&Z4.)
 - The CycleAbort_SS on the HMI should be a maintained style switch. Cycle Abort state is released when NormalCycle state set is achieved; then the segment counter is reset to zero; however, the temperature SP will continue to be the CA temperSP.
- Makeup of Recipe:
 - Recipe header field (description of recipe)
 - Part number(s) field
 - Control Mode (PID, MTL) for all zones and PID, MTL dynamic constants
 - Recipe Segments
 - Set points if constant temperature PID or MTL is used
 - Code will set Seg0 and Seg1 time to 0, Seg2 time to 1 second. Code will set Seg0, Seg1, Seg2 temperature to the CT SP.
 - Up to 20 segment set points for each zone if recipe is used
 - Each segment consists of: dwell, temp, ramp-yes/no, ramp rate, comment
 - The system capability for max ramp rate for Metal & Fluid is not known and depends highly on the Central System settings; therefore, the ramp rate will not be checked for validity. The recipe will have to be run to verify that the ramp rate can be sustained.
 - Metal preheat temperature limits for each zone
 - Cycle Abort temperature set points for each zone
 - Date and time recipe was saved – via inherent file properties.
 - Have a print capability from HMI and from computer. → Add-on enhancement.

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

Appendix A: Software Operating Requirements

- Transferring between control modes will be bump-less. This really only comes into play when in Manual Non-Recipe Constant Temperature mode. However, a recipe cannot be bumpless. The recipe must immediately start controlling to the given SP.
 - Transferring from CV to PID, will make PID set point the current fluid temperature, and preload the PID integrator with the current CV setting.
 - Transferring from CV to MTL, will make MTL set point the current metal temperature and set the MTL algorithm output equal to the current fluid temperature for the cascaded PID set point input, and preload the MTL output pipeline array with the current fluid temperature, as %range.
 - Transferring from PID to MTL, will make MTL set point the current metal temperature and set the MTL algorithm output equal to the current fluid temperature for the cascaded PID set point input, and preload the MTL output pipeline array with the current fluid temperature, as %range.
 - Transferring from MTL to PID, will make PID set point the current fluid temp
 - Transferring from MTL to CV, will set CV at 50%
 - Transferring from PID to CV, will set CV at 50%
 - Dick's method was to set CV at 0%. My rationale is that setting to 50% keeps existing fluid in the loop, which would be the least amount of stress on the process/molds. 0%=full cold; 50%=closed, recirculation loop; 100%=full hot.
- Control modes available:
 - Control Valve setting as a percent of range; only available to PAG users
 - Constant Fluid temperature by specifying a SP, and controlling via PID control
 - Constant Metal temperature by specifying a SP, and controlling to that SP via cascade PID/MTL control
 - Cycling of Fluid temp via PID control and a specified recipe profile
 - Cycling of Metal temp via cascade PID/MTL control and a specified recipe profile
- Cycle status — Zone basis: (only valid during a recipe)
 - Idle
 - MAN-CV, MAN-PID, or MAN-MTL.
 - MAN-CV is the Powerup default state, with CV=0% (Full Cold).
 - Recipe control when waiting at the end of SegLast, or waiting at Seg0 for StartRecipe_PB.
 - Preheat – currently running Preheat segments 0 or 1
 - RFCS (Ready For Cycle Start)
 - Run - currently running a recipe; segment number = 2 or greater
 - Done – all segments for the zone have completed
 - *Cycle Aborted is not a zone state, it is a machine state. However, it will be the reset action for the zone state.*
- Cycle status — Machine basis (based on status of all zones): (only valid during a recipe)
 - Idle – all zones Idle.
 - Init – PLC initialization operations.
 - Preheat – at least one zone still in Preheat.
 - RFCS – all zones RFCS.
 - Run – at least one zone still running a recipe. (Other zones may be in Done state.)
 - Cycle Aborted

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

Appendix A: Software Operating Requirements

- During the cycle:
 - The current segment's temperature set point can be changed (DF2) on the cycle status screen – only if the zone is under constant temperature control.
 - Trending screen will graph the defined recipe's SPs as well as the actual temperatures and show some limited zone and zone segment information.
- ESTOP mushroom PB currently mounted on each Press
 - Shuts down the pump package and the computer cabinet
 - New implementation will be to NOT shut down the HMI station, rather monitor the contact and react accordingly, the pumps will continue to be shut down (effectively setting CV=0%).
 - Set all zones to CV=0% Full Cold.
- ESTOP (Zone Relay) currently mounted in areas of D/26 and Bldg90 (master shutdowns)
 - There are three Zone Relays, each of which control a specific group of presses
 - Acts as Fire Alarm safety feature
 - Shuts down Central System – heaters, pumps, cooling towers, etc (not sure?)
 - Shuts down air handlers
 - Shuts down all control power to the Presses in the Zone
 - Shuts down all Press pump packages in the Zone (which closes valves)
 - Shuts down all computer cabinets in the Zone
- If lose Analog to Digital signals – Thermocouples
 - Abort recipe, set control to MAN-CV, close control valves (set CV=0)
 - *New system, for new building KCRIMS, may control the pump package*
- Integral Squared Error (ISE)
 - Calculated as a summation... $ISE = \sum_{i=1}^n e_i^2$
 - Calculated for zones with active MTL or PID control, and is only valid for segments 2 through last segment.
 - Reset to zero at beginning of new segment 2.
 - Stop accumulation of calculation when the last defined segment has timed out.
 - Units are percent of range squared.
 - This is a measure of how well controllers are tuned and provides a repeatability measure. The smaller the value, the better control.

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

Appendix A: Software Operating Requirements

- PID/MTL control methodology
 - All calculations (inputs, outputs, errors, etc) in the algorithms are based on percent of range rather than engineering units.
 - Temperatures (set points, error, etc) are set to a fixed range of 32-700°F, corresponding to a percent range of 0-100%.
 - User input temperatures are set to a range of 50-600°F (2.7-85 %Range)
 - If user input is outside of 50-600 range do not allow entry.
 - Convert errors to %Range before using in algorithms. Keep temperatures as °F as much as possible; TCs deliver a °F value.
- PID Algorithm; used for MAN-PID and PID modes
 - $$F_n = K_p e_n + \frac{K_p K_I T_1}{60} \sum_{i=1}^n e_i + \frac{60 K_p K_D}{T_1} (e_n - e_{n-1})$$
 - Output is a percentage; therefore errors must be converted to %range before calculating the output. Only the PID algorithm output affects the position of the control valves.
 - Number of data slots needed for error will be 3 (current, previous, sum of all previous)
 - Anti-reset windup
 - Clamp integrator portion of algorithm between 0-100%
 - Clamp controller output between 0-100%
 - After calculating, update ISE
 - Percent output will translate to 4-20 mA that transmits to the I/P converter which sends 3-15 psi to the CVs
 - Sample period, T_1 , is set at 2 seconds; can only be changed by PAG. The default, obtained through historical experimentation, should be displayed on the HMI.
 - Default constants: $K_p=1.5$, $K_I=4.5$, $K_D=0$; OR $K_p=12$, $K_I=1.5$, $K_D=0.05$. These values were obtained from historical recipes.
 - **Open Item:** Turn OFF Integrator and Derivative Control during Ramp segments, or some other variation of control? Left this option open for flexibility. Have documented with the PLC code as such.
 - **Open Item:** Provide new functionality for loop tuning of PID parameters → Add-on enhancement
 - Loop Tuning Fundamentals, Vance J. VanDoren, Control Engineering-November 1, 2003
 - Use what tuning method?
 - Ziegler-Nichols? $K_p=(1.2*\tau)/(Gain*\theta)$, $K_I=1/(2*\tau)$, $K_D=\theta/2$
 - Gain = Ending SP – Starting SP ??
 - This method would be very similar to what already doing for finding MTL dynamic parameters

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

Appendix A: Software Operating Requirements

- MTL Algorithm; used for MAN-MTL and MTL modes
 - $M_m = K' K_{CS} (e_m - P_0 e_{m-1} - P_1 e_{m-2}) + P_5 M_{m-1} + P_8 M_{m-N-1} + P_9 M_{m-N-2}$
 - Will always be cascaded with PID control, if user specifies MTL control
 - Dick does not use Derivative control for the cascaded PID loop in his derivation, but does allow the derivative to be used in the implementation, according to the documented code of BDX-613-3181. Shouldn't need derivative because MTL algorithm is a predictor which helps to reduce the derivative type of error. Will continue to allow derivative to be used, as it does not adversely affect the outcome.
 - Output is a percentage; therefore errors must be converted to %range before calculating the output. This output is the PID's input when cascading.
 - Output is the needed temperature, as a percent of range, of the heat transfer fluid to change the metal mold temperature to reach its defined set point by adjusting the control valves. Clamp output at +200 and -100%Range (Actual percent range values for +2x and -1x range are 204.79 [1400°F] to -109.58 [-700°F], which Dick used, but we will round to 200 [1368°F] to -100 [-636°F]) → Use +200 to -100 MTL output range clamp for PID set point when cascading (use actual MTL output if it is within the 200 to -100 limits). Again, the PID output controls the position of the control valves.
 - 'm' subscript differentiates MTL algorithm variables from PID variables; 'm' is current sample. See System Req. doc for further definition.
 - Number of data slots needed for past controller outputs, M, will be N+2
 - N = dead time / sample period → Maximum known dead time from all available old cards is 229 seconds.
 - Dick allotted 25 storage locations, such that N≤23; therefore, max dead time = 230 seconds. Dick says poorly designed molds, for heat transfer, have dead times greater than 200 seconds. I will have the ability to increase this number significantly to 50 locations; therefore allow for a max dead time = 480 seconds.
 - If N<1, then N=0. Even though this condition is unlikely, it will be accounted for.
 - Number of data slots needed for error will be 3 (current, plus 2 previous)
 - After calculating, update ISE
 - Sample period, T₂, is set at 10 seconds; can only be changed by PAG. The default, obtained through historical experimentation, should be displayed on the HMI. If in the future, T₂ must change, then T₂ ≥ 5*T₁.
 - Default constant: K'=0.33.
 - The closed loop time constant on all zones should be equal to the longest mold time constant, as a minimum value (from DF12 notes and BDX-613-1980). This is not the case in some historical recipes – allow these differences, but display a warning notation. At any case, for each zone at a minimum requirement, the closed loop time constant must be greater than or equal to the mold time constant, $\lambda \geq \tau_m$.
 - Need to provide functionality of DF13 to automatically determine the metal dynamics. Determines θ , τ_m , K_m . Calculates K_{CS} , λ , P0, P1, P5, P8, P9. Sets K'=0.33. Sample clock should be set at a minimum of 1 second.
 - Press is inhibited to prevent closure of the mold during the test.

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

Appendix A: Software Operating Requirements

- Zones to be tested must be in PID control, with no recipe running, before starting test. Can test multiple or single zones.
- Specify desired zones to calculate constants for.
- Specify desired Begin Metal Temperature (usually equal to Preheat Temp).
- Specify desired End Metal Temperature (usually equal to Cure Temp).
 - Designed for calculating heating constants ($T_{\text{MetalEnd}} > T_{\text{MetalBegin}}$).
 - Ability to accommodate cooling has been added.
- Specify Starting Fluid Temperature.
 - Desirable to set Starting Fluid Temperature at:
 - $T_{\text{MetalBegin}} * 0.97 < T_{\text{FluidStart}} < T_{\text{MetalBegin}} * 1.03$
- 5 minute stabilization period before beginning test for ALL zones
 - Stabilization $\Rightarrow T_{\text{MetalBegin}} * 0.97 < T_{\text{MetalActual}} < T_{\text{MetalBegin}} * 1.03$
- Dead time, θ = elapsed time from start of test (end of 5 minute stabilization period) until $T_{\text{metal}} \geq T_{\text{MetalBegin}} * 1.03$
- Time constant, τ_m = elapsed time, after dead time, for response to be 63.2% of the input = $0.632 * (T_{\text{MetalEnd}} - T_{\text{MetalBegin}}) + T_{\text{MetalBegin}}$
- Steady State Gain, $K_m = T_{\text{Metal}} / T_{\text{Fluid}}$ at $5 * \tau_m$
- Provide a test monitor.
- Provide Abort capability.
- End of test is when the maximum $5 * \tau_m$ for all zones has been reached. Hold zone at T_{MetalEnd} until all zones done. Calculate other parameters here.
- At end of test, set PID SP = 55°F to enable CV=0% and save energy.
- Inputs to PLC DL205, CPU WinPLC H2-WPLC3
 - Digital, D2-08ND3
 - MHC PP OI Relay
 - Analog, 3x F2-04THM
 - 8 thermocouples (Fluid & Metal temp for each of 4 zones)
 - 4 thermocouples (Temp of Hot and Cold Supply and Return lines)
- Outputs from PLC
 - Digital, D2-08TR
 - None defined. Press Inhibit explicitly excluded.
 - Analog, F2-08DA-1
 - 4 outputs 4-20mA to I/P converter

Press Inhibit

A conscious decision was made to exclude this feature. Documentation on this feature was somewhat sparse. As best we could tell, this feature was only implemented on injection molding machines.

Recipe OPC transfer from HMI to PLC.

See “recipe OPC xfer.xls”, which is also contained in the Code Listing Appendix files.

Appendix B: Hardware & Software Installation Guide

Hardware and Software Installation Guide
for Mold Heating and Cooling Pump Package Operator Interface

Industrial PC setup, and WinXP+SP2 setup section:

Appendix B: Hardware & Software Installation Guide

Change backup drive letter and name, change CD drive letter

Appendix B: Hardware & Software Installation Guide

Appendix B: Hardware & Software Installation Guide

Appendix B: Hardware & Software Installation Guide

Appendix B: Hardware & Software Installation Guide

Appendix B: Hardware & Software Installation Guide

My Computer showing drive letters and names

set a directory folder view to Details

Tools>Folder Options>View> check option--> Display the contents of system folders
 Tools>Folder Options>View> check option--> Display the full path in the title bar
 Tools>Folder Options>View> check option--> Show hidden files and folders
 Tools>Folder Options>View> uncheck option--> Hide extensions for known file types

Issue Date: 06/26/2009
 Revision Date: N/A

Appendix B: Hardware & Software Installation Guide

Tools>Folder Options>View> uncheck option--> Hide protected operating system files
Click Apply to All Folders

Appendix B: Hardware & Software Installation Guide

Control Panel>User Accounts>Change an account (Administrator) OR
 go thru ManageComputer>Local Users and Groups>Set password
 select Create a password ---> PAG standard password
 Answer NO to question Do you want to make your files and folders private?
 My Computer>Manage>Local User and Groups> properties of Administrator
 Description: PAG standard password; added to existing description
 Make sure option is checked: Password never expires

Appendix B: Hardware & Software Installation Guide

Control Panel>User Accounts>Create a new account ---> MHCAdmin (only option should be to select computer administrator)

select Create a password ---> PAG standard password

My Computer>Manage>Local User and Groups> properties of MHCAdmin

Description: PAG standard password

Make sure option is checked: Password never expires

Make sure is member of Administrators

Appendix B: Hardware & Software Installation Guide

Control Panel>User Accounts>Create a new account ---> MHCOper ---> Limited Account
My Computer>Manage>Local User and Groups> properties of MHCOper

Description: no password
Make sure option is checked: User cannot change password
Make sure option is checked: Password never expires
Make sure is member of Power Users

Control Panel>User Accounts>Change the way users log on or off
uncheck option Use Fast User Switching
check option Use the Welcome screen (this will make the Administrator account appear to not exist)

Log in as MHCAdmin
set a directory folder view to Details
Tools>Folder Options>View> check option--> Display the contents of system folders
Tools>Folder Options>View> check option--> Display the full path in the title bar
Tools>Folder Options>View> check option--> Show hidden files and folders
Tools>Folder Options>View> uncheck option--> Hide extensions for known file types
Tools>Folder Options>View> uncheck option--> Hide protected operating system files
Click Apply to All Folders

Copy installation CD files into directory structure

C:\Documents and Settings\MHCAdmin\My Documents\MoldHeatingAndCooling_MHC install apps CD vYYYY-MM-DD\

Place shortcut to this directory on desktop pointing to

C:\Documents and Settings\MHCAdmin\My Documents\MoldHeatingAndCooling\

Install Cyber Security Warning Banner from ntbanner.reg file by double clicking on the file from the \software execs\CyberSecurity\ directory on the CD

Appendix B: Hardware & Software Installation Guide

Taskbar settings for MHCAdmin

Appendix B: Hardware & Software Installation Guide

Turn on screen saver with password protection after 15 minutes

Under Monitor power options, edit Home/Office Desk profile to turn off monitor after 2 hours

Change screen resolution to 1280x1024

Settings>Advanced>Intel GraphicsMedia Accelerator Driver for Mobile>Graphics Properties>
Change options on Monitor tab, Single Display section to Monitor
Select Display Settings, set Color Quality to 32 Bit
set Screen Resolution to 1280x1024

Appendix B: Hardware & Software Installation Guide

set Refresh Rate to 60 Hertz
Settings> change resolution slider to 1280x1024

Appendix B: Hardware & Software Installation Guide

Change size of font in message box.

Appendix B: Hardware & Software Installation Guide

Change size of Windows scroll bars to 26 for easier touchscreen usage.

Appendix B: Hardware & Software Installation Guide

Log in as MHCOper to populate Documents and Settings folders
 Tools>Folder Options>View> uncheck option--> Hide extensions for known file types
 Set up ScreenSaver, MessageBox, and ScrollBars same as above

Create Logoff command and icon for MHCOper

In c:\ root create a new text document and rename it to LogOff.cmd

Edit the cmd file with the following lines

rem This command will Logoff the current user.

shutdown -l

rem This command is intended for MHCOper, via a shortcut in MHCOper desktop

Create shortcut in C:\Documents and Settings\MHCOper\Desktop\

Log in as MHCAdmin

move C:\Documents and Settings\All Users\Start Menu\Programs\Realtek Sound Manager\ from
 All Users to MHCAdmin

delete AvRack shortcut from C:\Documents and Settings\All Users\Desktop\

move Nero Setup shortcut from C:\Documents and Settings\All Users\Desktop to
 C:\Documents and Settings\MHCAdmin\Desktop\

delete Windows Media Player shortcut from C:\Documents and Settings\MHCOper\Desktop\

create directory C:\Documents and Settings\All Users\Start Menu\Programs\Tools\

move TZEDIT.EXE from C:\Documents and Settings\MHCAdmin\Desktop to
 C:\Documents and Settings\All Users\Start Menu\Programs\Tools\

move XyCom SysInfo shortcut to C:\Documents and Settings\All Users\Start Menu\Programs\Tools\

Appendix B: Hardware & Software Installation Guide

Appendix B: Hardware & Software Installation Guide

Software installation section

ELO Touch Drivers v4.40

Install from installation CD \hardware execs\SlimAgeTouchScreen\EloSetup.exe

Select English language.

Select checkbox for Install USB Touchscreen Drivers

Turn off checkbox for Show tool tray utility

Appendix B: Hardware & Software Installation Guide

Create shortcut to desktop from C:\Program Files\EloTouchSystems\EloTouch.cpl , rename to EloTouch Setup

Create shortcut to desktop from C:\Program Files\EloTouchSystems\EloVa.exe , rename to EloTouch Alignment

Copy the alignment shortcut and paste to

C:\Documents and Settings\All Users\Start Menu\Programs\Tools\

May need to edit the properties of the icon, to add Everyone to the Security.

Kensington Dual Monitor Adapter

don't install, but keep to possibly install later from \hardware execs\KensingtonDualMonitorAdapter\

AdobeReader8 -- all defaults

run exe file from \software execs\AdobeReader8\

copy shortcut on desktop and place in

C:\Documents and Settings\All Users\Start Menu\Programs\Tools\

Primo PDF

run exe file from \software execs\misc\

Convert and Printkey

copy exe files from \software execs\misc\ to C:\Program Files\

create shortcut to desktop

copy shortcut from desktop and paste in

C:\Documents and Settings\All Users\Start Menu\Programs\Tools\

Edit the properties of the icons, to add Everyone to the Security

Appendix B: Hardware & Software Installation Guide

Appendix B: Hardware & Software Installation Guide

InduSoft Web Studio (IWS) v6.1 + SP4

First time install of HMI software:

Turn off Firewall and Automatic Updates.

Un-install any previous versions of InduSoft Web Studio (IWS)

Install from installation CD \software execs\HMI-InduSoft\IWSv6.1+SP4 bundled\setup.exe

Install IWS in C:\Program Files\InduSoft Web Studio v6.1\ (default location)

uncheck "WinCE" and "Demo Files" options when installing.

You will be asked to install Microsoft .NET Framework 2.0, choose Next.

move the shortcut to the software on the desktop for All Users to MHCAdmin desktop

move IWS from start menu of All Users to start menu of MHCAdmin

get copy of IWS runtime shortcut from >Start Menu>IWS>Startup>; rename runtime shortcut to "MHC"

put IWS runtime shortcut in C:\Documents and Settings\All Users\Desktop\

and C:\Documents and Settings\MHCOper\Start Menu\Programs\Startup\

Place IWS runtime files from installation CD \software execs\HMI-InduSoft\IWS_vX.Y.Z\ in directory

C:\Documents and Settings\MHCOper\MoldHeatingAndCooling\HMI\IWS_vX.Y.Z\

Place BLANK copy of IWS runtime project under HMI directory to use when changing or updating the version. It may be necessary to point IWS to the blank instead of the IWS_v.X.Y.Z directory.

USB Hardkey with Runtime only should be used for operator stations.

Start IWS development software to re-populate the ini files in MHCAdmin directory; an error will show that it could not find NT Demo, navigate to the new IWS runtime directory on the c drive or the BLANK project and open the project.

Launch IWS runtime via 'MHC' shortcut to point it to the correct runtime directory. Close MHC app via PAG Utilities screen to get back to Windows.

When creating a new version of HMI runtime software:

Delete old version directory

create new directory IWS_vX.Y.Z under C:\Documents and Settings\MHCOper\MoldHeatingAndCooling\HMI\

where directory name is "IWS_vX.Y.Z", where X.Y.Z is the revision number of the HMI software

issue

Leave a backup copy of HMI runtime files somewhere in

\MHCAdmin\My Documents\MoldHeatingAndCooling\

Delete ini files in directories MHCAdmin and \MHCOper\Local Settings\Application Data\InduSoft Web Studio v6.1\

Start IWS development software to re-populate the ini files in MHCAdmin directory; an error will show that it could not find NT Demo, navigate to the new IWS runtime directory or BLANK project and open the project.

Launch IWS runtime 'MHC' Startup to point it to the correct runtime directory.

Configure IP Addresses:

Start>Control Panel>Network and Internet Connections> pull 'Network Connections' shortcut to desktop

Open shortcut

Appendix B: Hardware & Software Installation Guide

Right click on 'Local Area Connection' and select Properties

Both NWLink options may be deselected if desired.

Appendix B: Hardware & Software Installation Guide

Select Internet Protocol (TCP/IP), then select Properties button
Set IP to 10.0.0.11, Set Subnet mask to 255.255.255.0

The screenshot shows the 'Internet Protocol (TCP/IP) Properties' dialog box with the 'General' tab selected. The dialog box has a blue title bar with a question mark icon and a close button. Inside, there is a text box explaining that IP settings can be assigned automatically or manually. Two radio buttons are present: 'Obtain an IP address automatically' (unselected) and 'Use the following IP address:' (selected). Below the selected radio button, there are three input fields: 'IP address' (containing '10 . 0 . 0 . 11'), 'Subnet mask' (containing '255 . 255 . 255 . 0'), and 'Default gateway' (empty). Below these, there are two more radio buttons: 'Obtain DNS server address automatically' (unselected) and 'Use the following DNS server addresses:' (selected). Below the selected radio button, there are two input fields: 'Preferred DNS server' (empty) and 'Alternate DNS server' (empty). At the bottom right of the dialog box, there is an 'Advanced...' button. At the very bottom, there are 'OK' and 'Cancel' buttons.

Internet Protocol (TCP/IP) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

☐ Obtain an IP address automatically

☒ Use the following IP address:

IP address: 10 . 0 . 0 . 11

Subnet mask: 255 . 255 . 255 . 0

Default gateway: . . .

☐ Obtain DNS server address automatically

☒ Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

Advanced...

OK Cancel

Appendix B: Hardware & Software Installation Guide

Go back to Network Connections and edit properties for Local Area Connection 2

Go back to Network Connections and select Local Area Connection 2, right click, select 'Disable'
Go to the menu pick Advanced>Advanced Settings..., change the order of Connections such that LAN is first and LAN2 is second.

Appendix B: Hardware & Software Installation Guide

You will be asked to restart.

To reduce confusion in the future, place an empty ethernet RJ-45 plug into the LAN2 connection on the IPC chassis.

Appendix B: Hardware & Software Installation Guide

Entivity Think & Do (TND) Datapack Release 7.3

First time install of PLC software:

Make sure Firewall on computer is turned off!!!!!!

Phoenix-Contact/Entivity DataPack software, obtained from automationdirect.com, p/n PC-ENT-DAP, is v7.2. DO NOT USE. There are problems with its versions of the OPC files.

Use v7.3. Software is unlimited installs.

IMPORTANT!!! IT IS ESSENTIAL TO FOLLOW THE NEXT STEPS EXACTLY, OTHERWISE THE MHCOper's OPC CONNECTIVITY MAY NOT FUNCTION PROPERLY

- TEMPORARILY CHANGE MHCOper TO XP ADMIN USER RIGHTS, INSTALL ENTIVITY, CHANGE SOCKETTIMEOUT, CONNECT TO WINPLC
- CHANGE MHCOper BACK TO XP POWER USER RIGHTS
- THEN LOGIN AS MHCAdmin, CONNECT TO WINPLC, VERIFY SOCKETTIMEOUT
(steps are explained in detail below)

Install from installation CD \software execs\PLC-ThinkNDo\EntivityDataPackv7.3\Setup.exe

Choose to create a desktop icon.

Restart computer.

Use CE Watch to connect to WinPLC via OPC for the first time. (it's just like the ESP application for development)

Choose 'Studio Station' tab, 'Select CE Station' button

A dialog will pop-up stating that 'No Windows CE Runtime(s) found on the network.'

In the pop-up dialog, choose 'Reset using IPX', you will have to get the MAC address off of the WinPLC, dismount it from the DirectLogic 205 rack, it is on a label on the circuit board.

On the WinPLC circuit board, turn dip switch S0 to the ON position. This will allow the project to run normally on powerup.

You might have to get into this dialog more than once.

Appendix B: Hardware & Software Installation Guide

Appendix B: Hardware & Software Installation Guide

Rename the WinPLC name to 'MHCWinPLC', since this is name configured in the IWS software
Set description to 'Mold Heating and Cooling WinPLC processor for Pump Package Interface Control'
Configure the IP address of the WinPLC device to 10.0.0.10, Subnet Mask to 255.255.255.0, Default Gateway 0.0.0.0
Choose the Select button, although the WinPLC may not be recognized yet.
You may have to get back into 'Select CE Station' and hit the Select button again.
This process does not seem to be repeatable or predictable; just have to jack with it until it works.
The Power, Run, and Link lights eventually will be blinking continuously, this usually means that all you have to do is cycle power on the WinPLC and then the CE Watch will finally see it.
You may have to cycle power on WinPLC several times throughout this connection process.
After connection has been made, the date/time of the WinPLC processor will be updated to the WinXP date/time via a HMI script, the next time the HMI is started.

Appendix B: Hardware & Software Installation Guide

The WinPLC firmware may need to be updated. This can be done before downloading the project or during. The PLC runtime logic is located on the installation CD in directory \software execs\PLC-ThinkNDo\TND_vX.Y.Z\ In CE Watch select 'Load & Start' button to update the runtime logic. Navigate to the above directory that contains the MHC WinPLC project, select the MHC.cfg file and download.

Appendix B: Hardware & Software Installation Guide

You will get an error about older version, choose 'Yes' to Continue anyway.

Appendix B: Hardware & Software Installation Guide

Appendix B: Hardware & Software Installation Guide

The CE OPC Server may be set to 'High';

'Above Normal' and 'Normal' seemed to work fine as of 01/21/2009, left at 'Normal'

Entivity tech support recommends running at lowest priority possible and said Normal should be sufficient.

Appendix B: Hardware & Software Installation Guide

Change registry entries for SocketTimeout

Find/search for SocketTimeout

Change both Send and Receive fields to 30, via the decimal edit.

Appendix B: Hardware & Software Installation Guide

Issue Date:
Revision Date:

Appendix B: Hardware & Software Installation Guide

Delete start/programs/games C:\Documents and Settings\All Users\Start Menu\Programs\Games\
Clean up any other start menu shortcuts and start menu group/folder picks for All Users, MHCAdmin, MHCOper, Administrator.

The MHCAdmin user's desktop icons should look something like this:

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

Appendix B: Hardware & Software Installation Guide

Create a directory on MHCAdmin's desktop to house the backup command scripts.
Copy the scripts from the installation CD \software execs\bkup_cmd_scripts\ to the desktop directory.
Schedule the scripts to run in >Start>Settings>Control Panel>Scheduled Tasks>Add Scheduled Task>
(The backup scripts copies the HMI files to the backup drive.)

MAC addresses and IP setup

WinPLC

00.E0.62.80.18.FA (D/26 cabinet)

00.E0.62.80.13.C0 (PAG development)

factory settings

IP: 10.0.1.100

SM: 255.0.0.0

DG: 0.0.0.0

MHCWinPLC NEW SETTINGS

IP: 10.0.0.10

SM: 255.255.255.0

DG: 0.0.0.0

INDUSTRIAL PC

IP: 10.0.0.11, 10.0.0.12 (address 12 external port should be plugged to avoid confusion)

SM: 255.255.255.0

DG: 0.0.0.0

HMI v3.1.0 date can be verified by the file dbversio.bin, which should be timestamped 04/08/2009 13:05.

PLC v3.0.1 date can be verified by the file mhc.map, which should be timestamped 04/08/2009 13:45:28.

Josh Salmond

u60479

April 2009

End of Document: Hardware and Software Installation Guide
for Mold Heating and Cooling Pump Package Operator Interface

Issue Date: 06/26/2009

Revision Date: N/A

Page 89

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

Appendix C: State Set conversion to Think & Do Flowcharts

Appendix C: State Set conversion to Think & Do Flowcharts

Since the StateCAD code cannot be run on the Think&Do platform, the state design had to be converted to the Think&Do programming logic.

In order to establish a standard, generic state cases were generated and guidelines formed for conversion. Some special cases had to be accounted for, but for the most part these standards were followed. If the transition condition contained an 'AND' or 'OR' condition, or there were multiple transitions, another decision block would be added on the NO path of the first decision block. The order of the decision blocks determined the priority of the transition.

The following graphics illustrate the standards:

Figure C - 1: Reset Action handles the reset condition of all states that use that specific reset condition. If a state set did not have a specific reset condition, the StateCAD design used "First Scan" notation to account for the reset condition. In the Think&Do implementation, the state that contained the First Scan reset action was equated to one (ON) in the tags database, and all other related states in the set were equated to zero (OFF). Therefore, on the first scan, and/or at system startup, this state would be active.

Figure C - 2: Generic State Set shows the generic conversion from state logic to flowchart logic accounting for state actions and reset actions.

Figure C - 3: Generic State Set as Subroutine shows the conversion of a group of state sets that are of the same configuration, with only the only difference being the state variable names. This technique was very useful for this project as a given state set dealing with zones would have four very similar state sets. In Think&Do, a sub-chart, or subroutine, could be written with generic variables. The generic variables would be matched with the specific project variables in the calling block of the flowchart. Another advantage of this method was that during development, if a state set design changed, only the sub-chart had to be updated instead of four different charts.

Figure C - 4: Loopback State deals with the situation of performing a specific task while in an active state, but is desirable to not leave that state.

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

Appendix C: State Set conversion to Think & Do Flowcharts

Figure C - 1: Reset Action

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

Appendix C: State Set conversion to Think & Do Flowcharts

Figure C - 2: Generic State Set

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

Appendix C: State Set conversion to Think & Do Flowcharts

Figure C - 3: Generic State Set as Subroutine

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

Appendix C: State Set conversion to Think & Do Flowcharts

Figure C - 4: Loopback State

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

Appendix D: WinPLC Code Listing

Appendix D: WinPLC Code Listing

An Adobe pdf file was created that displays the PLC code listing. Due to file size, file length, and incompatibilities, the pdf was not posted in this Appendix. The pdf file can be found at
\\Public\pag\PAG Content\MHC Upgrade\documentation\Design and Maintenance\CodeListings.

Mold Heating and Cooling Pump Package Operator Interface

State Design and PLC Code Listing

**State Sets designed with Xilinx StateCAD
Release 7.1i**

**PLC designed with Entivity Think&Do Live!
Programming Pack v5.6 from
AutomationDirect.com**

Project 703030, Activity 70303020

April 2009

Josh Salmond

Mold Heating and Cooling Pump Package Operator Interface

State Design

MHC v3.0.1

Layout is meant for B size Landscape

This is a copy of the master. Changes must be made to file "MHC3_0_1.DIA"

Color scheme:
Yellow State = Reset State
Orange State = Error, Warning, or Alarm State
Blue State = all others

Orange Comment = Question or Resolution needed
Blue Comment = additional code or logic

Variable Naming Convention --> See Excel Spreadsheet: VariableMapping TND&IWS.xls

PLC - Entity Think & Do (TND) Variables, prefix:
s = State Flag
f = Discrete Flag etc

HMI - InduSoft Web Studio (IWS) Variables, prefix:
same as TND but additional "h" prefix
hf = Discrete Flag etc

Warnings that appear on StateCAD for a state that has 2 or more transitions leaving it can be ignored, because the TND platform will analyze the transition conditions in a serial manner; therefore, if the first transition is true, then control will move to that state without checking the conditions of the second transition.

"Check" number is order of question blocks when checking transition condition (to transition to another state) when programming TND flowchart

StateCAD variable = TND Variable or Logic

sMTLTimerIncr = sMTLTimerIncrement
nMTLAccum_lt_SP = nMTLAccum less than T2 (10 seconds)
T1 (2 seconds)
nMTLAccum_eq_SP = nMTLAccum equals T2

While in state s__TimerIncr, increment the number timer tag, n__Accum, or nSegElapsedTimeZx, or nCycleElapsedTime.

StateCAD variable = TND Variable or Logic

nCycElapTime_eq0 = (output action) reset nCycleElapsedTime to zero

StateCAD variable = TND Variable or Logic

nSegElapZx_lt_SP = nSegElapsedTimeZx is less than the current segment's recipe Set Point for the zone
nSegElapZx_gteSP = nSegElapsedTimeZx greater than or equal to recipe Set Point
nSegElapZ1_eq0 = reset nSegElapsedTimeZx to zero IN STATE SET sSpSeg !!!

"Start" (hfStartRecipe) not available until a recipe is loaded on HMI.
Hide "Start" button once in sRecipeActive.
Hide "Start" button if in Manual Mode CV control for ANY zone.

hfManualPreheat = 1
hfAutoPreheat = !hfManualPreheat = 0

Manual Preheat is same as Dick's PREHEAT
Auto Preheat is same as Dick's AUTO WRAP AROUND

StateCAD variable = TND Variable or Logic
hstFileCreateTrg = hstFileCreateTrigger

In state sRXInitPLC
- move zeros into recipe segment array registers
- check hfRXHandshake before continuing

In state sRXTransSegData OR ConData
- The transition criteria for Seg# or Zone# depends on 2-zone or 4-zone machine. If 2-zone, then Seg#=39; if 4-zone, then Seg#=79.
- turn OFF fDataStored
- wait for Handshake ON from HMI
- move data from HMI to PLC tags via OPC temporary tags
- increment Seg# if not done
- turn ON fDataStored
- wait for Handshake OFF from HMI
- loop back to "turn OFF fDataStored" until done

XFER of DATA: In IWS script:
- enable OPC communication sheet
- assign first set of data tags to temporary tags
- turn ON Handshake
- wait for fDataStored ON from PLC
- assign next set of data tags
- turn OFF Handshake
- wait for fDataStored OFF from PLC
- loop back to "turn ON Handshake" until done

This state chart is merely a representation of the actual implementation of the Recipe Transfer. The Recipe Transfer's original state design did not fit well with the Think&Do implementation.

StateCAD variable = TND Variable or Logic

fSpSegUpDoneZ1 = fSpSegUpdateDoneZ1
SegNumZ1_gteLast = current segment number, cSegmentNumberZ1 >= hnSegLastZ1
nSegElapZ1_eq0 = reset nSegElapsedTimeZ1 to zero

NOTE!!: nSegElapZ1_eq0 must be in this state (as opposed to sSegTimer) to keep the value populated, so the HMI user knows that time has expired at the end of a segment when waiting on other segments to finish. Also cannot consolidate the action as a state action in sSpSegUpdateZ1; the action must be a transition action because the time needs to be reset at that instant; also had a problem resetting the time when it was a state action during AutoWraparound recipes.

Control Mode established at HMI, array of integers.
Index, i=1,2,3,4 represents the Zone#
hnControlModeZ[i]= 0 =Control Valve control
hnControlModeZ[i]= 1 =Fluid Const Temper control
hnControlModeZ[i]= 3 =Mold Const Temper control
hnControlModeZ[i]= 2 =Fluid Recipe control
hnControlModeZ[i]= 4 =Mold Recipe control

had to change values so that I could use > and < operators in TND decision blocks

In Manual Control, the user (PAG only) changes an HMI tag for the valve position.
The HMI tag is transported to the PLC via OPC.
The PLC tag then enables a write to the AnalogOut card to change the valve position.

In Auto Control (PID or MTL, Const Temp or Recipe), the PLC tag enables a write to the AnalogOut card to change the valve position.

All 4 zones must either be controlled by Const Temp or by a defined profile before the Recipe can be saved. (Don't want to allow the possibility of CV control while a recipe is running.)

**Mold Heating and Cooling
Pump Package
Operator Interface**

PLC Code Listing

MHC TND_v3.0.1 Flow Chart Listing

MHC.fcn

Documentation	sIDLE	sValveManualControlZ1
sTwoZoneMachine	sCycleINIT	sValveManualControlZ2
sFourZoneMachine	sPREHEAT	sValveManualControlZ3
sHMI_Heartbeat_Init	sRFCS	sValveManualControlZ4
sHMI_Heartbeat_PulseUP	sRUN	sValveAutoControlZ1
sHMI_Heartbeat_PulseDN	sIdleZ1	sValveAutoControlZ2
sHMIAlive	sIdleZ2	sValveAutoControlZ3
sHMIDead	sIdleZ3	sValveAutoControlZ4
sCycleNormal	sIdleZ4	sSpMTLOFFZ1
sCycleAbortRecipe	sPreheatZ1	sSpMTLOFFZ2
sCycleAbortControl	sPreheatZ2	sSpMTLOFFZ3
sAccumOFF	sPreheatZ3	sSpMTLOFFZ4
sAccumON	sPreheatZ4	sSpMTLUpdateZ1
sAccumWait	sRFCSZ1	sSpMTLUpdateZ2
sMTLTimerIdle	sRFCSZ2	sSpMTLUpdateZ3
sMTLTimerIncrement	sRFCSZ3	sSpMTLUpdateZ4
sMTLTimerDone	sRFCSZ4	sMTLAlgOFFZ1
sPIDTimerIdle	sRunZ1	sMTLAlgOFFZ2
sPIDTimerIncrement	sRunZ2	sMTLAlgOFFZ3
sPIDTimerDone	sRunZ3	sMTLAlgOFFZ4
sSegTimerIdleZ1	sRunZ4	sMTLAlgCalcZ1
sSegTimerIdleZ2	sDoneZ1	sMTLAlgCalcZ2
sSegTimerIdleZ3	sDoneZ2	sMTLAlgCalcZ3
sSegTimerIdleZ4	sDoneZ3	sMTLAlgCalcZ4
sSegTimerIncrementZ1	sDoneZ4	sSpPIDOFFZ1
sSegTimerIncrementZ2	sRecipeNotLoaded	sSpPIDOFFZ2
sSegTimerIncrementZ3	sRecipeDownloadNew	sSpPIDOFFZ3
sSegTimerIncrementZ4	sRecipeCurrent	sSpPIDOFFZ4
sSegTimerHoldZ1	sRecipeIdle	sSpPIDUpdateZ1
sSegTimerHoldZ2	sRecipeActive	sSpPIDUpdateZ2
sSegTimerHoldZ3	sRecipeDone	sSpPIDUpdateZ3
sSegTimerHoldZ4	sRecipeAutoWrapAround	sSpPIDUpdateZ4
sSegTimerDoneZ1	sRecipeXferOFF	sPIDAlgOFFZ1
sSegTimerDoneZ2	sRXInitPLC	sPIDAlgOFFZ2
sSegTimerDoneZ3	sRXTransferSegData	sPIDAlgOFFZ3
sSegTimerDoneZ4	sRXTransferConData	sPIDAlgOFFZ4
sCycleTimerIdle	sRXPLCLoaded	sPIDAlgCalcZ1
sCycleTimerIncrement	sSpSegOFFZ1	sPIDAlgCalcZ2
sCycleTimerDone	sSpSegOFFZ2	sPIDAlgCalcZ3
sSkipPreheatIdle	sSpSegOFFZ3	sPIDAlgCalcZ4
sSkipPreheat	sSpSegOFFZ4	sTC_OK_FluidZ1
sSkipPreheatDone	sSpSegUpdateZ1	sTC_OK_FluidZ2
sMoldTemperLimitsBreachZ1	sSpSegUpdateZ2	sTC_OK_FluidZ3
sMoldTemperLimitsBreachZ2	sSpSegUpdateZ3	sTC_OK_FluidZ4
sMoldTemperLimitsBreachZ3	sSpSegUpdateZ4	sTC_Open_FluidZ1
sMoldTemperLimitsBreachZ4	sSpSegLoadedZ1	sTC_Open_FluidZ2
sMoldTemperLimitsOKZ1	sSpSegLoadedZ2	sTC_Open_FluidZ3
sMoldTemperLimitsOKZ2	sSpSegLoadedZ3	sTC_Open_FluidZ4
sMoldTemperLimitsOKZ3	sSpSegLoadedZ4	
sMoldTemperLimitsOKZ4		

MHC TND_v3.0.1 Flow Chart Listing

MHC.fcn

sTC_Short_FluidZ1	sMTTauldleZ1
sTC_Short_FluidZ2	sMTTauldleZ2
sTC_Short_FluidZ3	sMTTauldleZ3
sTC_Short_FluidZ4	sMTTauldleZ4
sTC_OK_MoldZ1	sMTTtauTestZ1
sTC_OK_MoldZ2	sMTTtauTestZ2
sTC_OK_MoldZ3	sMTTtauTestZ3
sTC_OK_MoldZ4	sMTTtauTestZ4
sTC_Open_MoldZ1	sMTTtauDoneZ1
sTC_Open_MoldZ2	sMTTtauDoneZ2
sTC_Open_MoldZ3	sMTTtauDoneZ3
sTC_Open_MoldZ4	sMTTtauDoneZ4
sTC_Short_MoldZ1	sMT5TauldleZ1
sTC_Short_MoldZ2	sMT5TauldleZ2
sTC_Short_MoldZ3	sMT5TauldleZ3
sTC_Short_MoldZ4	sMT5TauldleZ4
sMTLTuneIdle	sMT5tauTestZ1
sMTLTuneInit	sMT5tauTestZ2
sMTLTuneStabilizeDone	sMT5tauTestZ3
sMTLTuneDone	sMT5tauTestZ4
sMTLTuneAbort	sMT5tauDoneZ1
sMTStblzIdleZ1	sMT5tauDoneZ2
sMTStblzIdleZ2	sMT5tauDoneZ3
sMTStblzIdleZ3	sMT5tauDoneZ4
sMTStblzIdleZ4	UpdatePLCClock
sMTStblzInitZ1	z-end
sMTStblzInitZ2	
sMTStblzInitZ3	
sMTStblzInitZ4	
sMTStblzTestZ1	
sMTStblzTestZ2	
sMTStblzTestZ3	
sMTStblzTestZ4	
sMTStblzDoneZ1	
sMTStblzDoneZ2	
sMTStblzDoneZ3	
sMTStblzDoneZ4	
sMTThetaIdleZ1	
sMTThetaIdleZ2	
sMTThetaIdleZ3	
sMTThetaIdleZ4	
sMTThetaTestZ1	
sMTThetaTestZ2	
sMTThetaTestZ3	
sMTThetaTestZ4	
sMTThetaDoneZ1	
sMTThetaDoneZ2	
sMTThetaDoneZ3	
sMTThetaDoneZ4	

MHC TND_v3.0.1 Tag Listing

MHC.map

1	I-0	xESTOP	49	F-34	sSegTimerHoldZ3
2	I-1	xMHCRelay	50	F-35	sSegTimerHoldZ4
3	I-2	x2	51	F-36	sCycleTimerIdle
4	I-3	x3	52	F-37	sCycleTimerIncrement
5	I-4	x4	53	F-38	sCycleTimerDone
6	I-5	x5	54	F-40	sSkipPreheatIdle
7	I-6	x6	55	F-41	sSkipPreheat
8	I-7	x7	56	F-42	sSkipPreheatDone
9	I-20	xTC_Open_FluidZ1	57	F-43	hfSkipPreheat
10	I-21	xTC_Open_FluidZ2	58	F-45	hfManualPreheat
11	I-22	xTC_Open_FluidZ3	59	F-51	sRecipeNotLoaded
12	I-23	xTC_Open_FluidZ4	60	F-52	sRecipeCurrent
13	I-24	xTC_Open_MoldZ1	61	F-53	sRecipeDownloadNew
14	I-25	xTC_Open_MoldZ2	62	F-56	hfStartRecipe
15	I-26	xTC_Open_MoldZ3	63	F-58	hfCancelDownload
16	I-27	xTC_Open_MoldZ4	64	F-59	sRecipeXferOFF
17	I-28	xTC_Open_CentralHotSupply	65	F-60	sRXInitPLC
18	I-29	xTC_Open_CentralHotReturn	66	F-61	F-61
19	I-30	xTC_Open_CentralColdSupply	67	F-62	sRXTransferSegData
20	I-31	xTC_Open_CentralColdReturn	68	F-63	sRXPLCLoaded
21	I-33	--- end INPUT definitions ---	69	F-64	hfRXHandshake
22	F-0	Flag0	70	F-65	F-65
23	F-2	sTwoZoneMachine	71	F-66	fRXSegDataStored
24	F-3	sFourZoneMachine	72	F-67	F-67
25	F-6	AlwaysOFF	73	F-68	F-68
26	F-9	sAccumOFF	74	F-69	sRXTransferConData
27	F-10	sAccumON	75	F-70	fRXConDataStored
28	F-11	sAccumWait	76	F-74	sCycleAbortControl
29	F-13	sMTLTimerIdle	77	F-75	sCycleNormal
30	F-14	sMTLTimerIncrement	78	F-76	sCycleAbortRecipe
31	F-15	sMTLTimerDone	79	F-77	fCycleAbort
32	F-16	sPIDTimerIdle	80	F-78	hfCycleAbort
33	F-17	sPIDTimerIncrement	81	F-79	hfDryCycle
34	F-18	sPIDTimerDone	82	F-81	hstFileCreateTrigger
35	F-20	sSegTimerIdleZ1	83	F-82	sRecipeIdle
36	F-21	sSegTimerIncrementZ1	84	F-83	sRecipeActive
37	F-22	sSegTimerDoneZ1	85	F-84	sRecipeDone
38	F-23	sSegTimerIdleZ2	86	F-85	sRecipeAutoWrapAround
39	F-24	sSegTimerIncrementZ2	87	F-86	sIdleZ1
40	F-25	sSegTimerDoneZ2	88	F-87	sPreheatZ1
41	F-26	sSegTimerIdleZ3	89	F-88	sRFLCSZ1
42	F-27	sSegTimerIncrementZ3	90	F-89	sRunZ1
43	F-28	sSegTimerDoneZ3	91	F-90	sIdleZ2
44	F-29	sSegTimerIdleZ4	92	F-91	sPreheatZ2
45	F-30	sSegTimerIncrementZ4	93	F-92	sRFLCSZ2
46	F-31	sSegTimerDoneZ4	94	F-93	sRunZ2
47	F-32	sSegTimerHoldZ1	95	F-94	sIdleZ3
48	F-33	sSegTimerHoldZ2	96	F-95	sPreheatZ3

MHC TND_v3.0.1 Tag Listing

MHC.map

97	F-96	sRFCSZ3	145	F-162	sSpPIDOFFZ1
98	F-97	sRunZ3	146	F-163	sSpPIDUpdateZ1
99	F-98	sIdleZ4	147	F-164	sSpMTLOFFZ2
100	F-99	sPreheatZ4	148	F-165	sSpMTLUpdateZ2
101	F-100	sRFCSZ4	149	F-166	sSpPIDOFFZ2
102	F-101	sRunZ4	150	F-167	sSpPIDUpdateZ2
103	F-104	sIDLE	151	F-168	sSpMTLOFFZ3
104	F-105	sCycleINIT	152	F-169	sSpMTLUpdateZ3
105	F-106	sPREHEAT	153	F-170	sSpPIDOFFZ3
106	F-107	sRFCS	154	F-171	sSpPIDUpdateZ3
107	F-108	sRUN	155	F-172	sSpMTLOFFZ4
108	F-111	sHMI_Heartbeat_Init	156	F-173	sSpMTLUpdateZ4
109	F-112	sHMI_Heartbeat_PulseUP	157	F-174	sSpPIDOFFZ4
110	F-113	sHMI_Heartbeat_PulseDN	158	F-175	sSpPIDUpdateZ4
111	F-114	hfHMIHeartbeat	159	F-181	sValveManualControlZ1
112	F-115	sHMIAlive	160	F-182	sValveAutoControlZ1
113	F-116	sHMIDead	161	F-183	sValveManualControlZ2
114	F-118	sDoneZ1	162	F-184	sValveAutoControlZ2
115	F-119	sDoneZ2	163	F-185	sValveManualControlZ3
116	F-120	sDoneZ3	164	F-186	sValveAutoControlZ3
117	F-121	sDoneZ4	165	F-187	sValveManualControlZ4
118	F-127	sMoldTemperLimitsBreachZ1	166	F-188	sValveAutoControlZ4
119	F-128	sMoldTemperLimitsOKZ1	167	F-190	sPIDAlgOFFZ1
120	F-129	sMoldTemperLimitsBreachZ2	168	F-191	sPIDAlgCalcZ1
121	F-130	sMoldTemperLimitsOKZ2	169	F-192	sMTLAlgOFFZ1
122	F-131	sMoldTemperLimitsBreachZ3	170	F-193	sMTLAlgCalcZ1
123	F-132	sMoldTemperLimitsOKZ3	171	F-194	sPIDAlgOFFZ2
124	F-133	sMoldTemperLimitsBreachZ4	172	F-195	sPIDAlgCalcZ2
125	F-134	sMoldTemperLimitsOKZ4	173	F-196	sMTLAlgOFFZ2
126	F-136	hfMoldTemperLimitsZ1	174	F-197	sMTLAlgCalcZ2
127	F-137	hfMoldTemperLimitsZ2	175	F-198	sPIDAlgOFFZ3
128	F-138	hfMoldTemperLimitsZ3	176	F-199	sPIDAlgCalcZ3
129	F-139	hfMoldTemperLimitsZ4	177	F-200	sMTLAlgOFFZ3
130	F-140	hfRXData_MoldTemperLimits	178	F-201	sMTLAlgCalcZ3
131	F-146	fBumplessMTLZ1	179	F-202	sPIDAlgOFFZ4
132	F-147	fBumplessPIDZ1	180	F-203	sPIDAlgCalcZ4
133	F-148	fBumplessCVZ1	181	F-204	sMTLAlgOFFZ4
134	F-149	fBumplessMTLZ2	182	F-205	sMTLAlgCalcZ4
135	F-150	fBumplessPIDZ2	183	F-207	hfMTLConstsDefinedZ1
136	F-151	fBumplessCVZ2	184	F-208	hfMTLConstsDefinedZ2
137	F-152	fBumplessMTLZ3	185	F-209	hfMTLConstsDefinedZ3
138	F-153	fBumplessPIDZ3	186	F-210	hfMTLConstsDefinedZ4
139	F-154	fBumplessCVZ3	187	F-211	hfRXData_ConstsDefined
140	F-155	fBumplessMTLZ4	188	F-212	fSPRampYNZ1
141	F-156	fBumplessPIDZ4	189	F-213	fSPRampYNZ2
142	F-157	fBumplessCVZ4	190	F-214	fSPRampYNZ3
143	F-160	sSpMTLOFFZ1	191	F-215	fSPRampYNZ4
144	F-161	sSpMTLUpdateZ1	192	F-216	hfRXData_RampYN_1

MHC TND_v3.0.1 Tag Listing

MHC.map

193	F-217	hfRXData_RampYN_2	241	F-275	sTC_Open_CCS
194	F-218	hfRXData_RampYN_3	242	F-276	sTC_Short_CCS
195	F-219	hfRXData_RampYN_4	243	F-277	hfHMIConfirm_CCS
196	F-220	fSPRampDoneZ1	244	F-278	sTC_OK_CCR
197	F-221	fSPRampDoneZ2	245	F-279	sTC_Open_CCR
198	F-222	fSPRampDoneZ3	246	F-280	sTC_Short_CCR
199	F-223	fSPRampDoneZ4	247	F-281	hfHMIConfirm_CCR
200	F-230	sTC_OK_FluidZ1	248	F-284	sSpSegOFFZ1
201	F-231	sTC_Open_FluidZ1	249	F-285	sSpSegUpdateZ1
202	F-232	sTC_Short_FluidZ1	250	F-286	sSpSegLoadedZ1
203	F-233	hfHMIConfirm_FluidZ1	251	F-287	sSpSegOFFZ2
204	F-234	sTC_OK_MoldZ1	252	F-288	sSpSegUpdateZ2
205	F-235	sTC_Open_MoldZ1	253	F-289	sSpSegLoadedZ2
206	F-236	sTC_Short_MoldZ1	254	F-290	sSpSegOFFZ3
207	F-237	hfHMIConfirm_MoldZ1	255	F-291	sSpSegUpdateZ3
208	F-239	sTC_OK_FluidZ2	256	F-292	sSpSegLoadedZ3
209	F-240	sTC_Open_FluidZ2	257	F-293	sSpSegOFFZ4
210	F-241	sTC_Short_FluidZ2	258	F-294	sSpSegUpdateZ4
211	F-242	hfHMIConfirm_FluidZ2	259	F-295	sSpSegLoadedZ4
212	F-243	sTC_OK_MoldZ2	260	F-296	fSpSegUpdateDoneZ1
213	F-244	sTC_Open_MoldZ2	261	F-297	fSpSegUpdateDoneZ2
214	F-245	sTC_Short_MoldZ2	262	F-298	fSpSegUpdateDoneZ3
215	F-246	hfHMIConfirm_MoldZ2	263	F-299	fSpSegUpdateDoneZ4
216	F-248	sTC_OK_FluidZ3	264	F-300	sMTLTuneIdle
217	F-249	sTC_Open_FluidZ3	265	F-301	sMTLTuneInit
218	F-250	sTC_Short_FluidZ3	266	F-302	sMTLTuneStabilizeDone
219	F-251	hfHMIConfirm_FluidZ3	267	F-303	sMTLTuneDone
220	F-252	sTC_OK_MoldZ3	268	F-304	sMTLTuneAbort
221	F-253	sTC_Open_MoldZ3	269	F-305	hfMTLTuneStart
222	F-254	sTC_Short_MoldZ3	270	F-306	hfMTLTuneAbort
223	F-255	hfHMIConfirm_MoldZ3	271	F-307	hfMTLTuneNoTestZ1
224	F-257	sTC_OK_FluidZ4	272	F-308	hfMTLTuneNoTestZ2
225	F-258	sTC_Open_FluidZ4	273	F-309	hfMTLTuneNoTestZ3
226	F-259	sTC_Short_FluidZ4	274	F-310	hfMTLTuneNoTestZ4
227	F-260	hfHMIConfirm_FluidZ4	275	F-312	sMTStblzIdleZ1
228	F-261	sTC_OK_MoldZ4	276	F-313	sMTStblzInitZ1
229	F-262	sTC_Open_MoldZ4	277	F-314	sMTStblzTestZ1
230	F-263	sTC_Short_MoldZ4	278	F-315	sMTStblzDoneZ1
231	F-264	hfHMIConfirm_MoldZ4	279	F-316	sMTThetaIdleZ1
232	F-266	sTC_OK_CHS	280	F-317	sMTThetaTestZ1
233	F-267	sTC_Open_CHS	281	F-318	sMTThetaDoneZ1
234	F-268	sTC_Short_CHS	282	F-319	sMTTauIdleZ1
235	F-269	hfHMIConfirm_CHS	283	F-320	sMTTauTestZ1
236	F-270	sTC_OK_CHR	284	F-321	sMTTauDoneZ1
237	F-271	sTC_Open_CHR	285	F-322	sMT5TauIdleZ1
238	F-272	sTC_Short_CHR	286	F-323	sMT5TauTestZ1
239	F-273	hfHMIConfirm_CHR	287	F-324	sMT5TauDoneZ1
240	F-274	sTC_OK_CCS	288	F-326	sMTStblzIdleZ2

MHC TND_v3.0.1 Tag Listing

MHC.map

289	F-327	sMTStblzInitZ2	337	F-389	placeholder PID tuning0002
290	F-328	sMTStblzTestZ2	338	F-390	placeholder PID tuning0003
291	F-329	sMTStblzDoneZ2	339	F-391	placeholder PID tuning0004
292	F-330	sMTThetaIdleZ2	340	F-392	placeholder PID tuning0005
293	F-331	sMTThetaTestZ2	341	F-393	placeholder PID tuning0006
294	F-332	sMTThetaDoneZ2	342	F-394	placeholder PID tuning0007
295	F-333	sMTTauIdleZ2	343	F-395	placeholder PID tuning0008
296	F-334	sMTTauTestZ2	344	F-396	placeholder PID tuning0009
297	F-335	sMTTauDoneZ2	345	F-397	placeholder PID tuning0010
298	F-336	sMT5TauIdleZ2	346	F-398	placeholder PID tuning0011
299	F-337	sMT5TauTestZ2	347	F-399	placeholder PID tuning0012
300	F-338	sMT5TauDoneZ2	348	F-400	placeholder PID tuning0013
301	F-340	sMTStblzIdleZ3	349	F-401	placeholder PID tuning0014
302	F-341	sMTStblzInitZ3	350	F-402	placeholder PID tuning0015
303	F-342	sMTStblzTestZ3	351	F-403	placeholder PID tuning0016
304	F-343	sMTStblzDoneZ3	352	F-404	placeholder PID tuning0017
305	F-344	sMTThetaIdleZ3	353	F-405	placeholder PID tuning0018
306	F-345	sMTThetaTestZ3	354	F-406	placeholder PID tuning0019
307	F-346	sMTThetaDoneZ3	355	F-407	placeholder PID tuning0020
308	F-347	sMTTauIdleZ3	356	F-408	placeholder PID tuning0021
309	F-348	sMTTauTestZ3	357	F-409	placeholder PID tuning0022
310	F-349	sMTTauDoneZ3	358	F-410	placeholder PID tuning0023
311	F-350	sMT5TauIdleZ3	359	F-411	placeholder PID tuning0024
312	F-351	sMT5TauTestZ3	360	F-412	placeholder PID tuning0025
313	F-352	sMT5TauDoneZ3	361	F-413	placeholder PID tuning0026
314	F-354	sMTStblzIdleZ4	362	F-500	fRampYN0000
315	F-355	sMTStblzInitZ4	363	F-501	fRampYN0001
316	F-356	sMTStblzTestZ4	364	F-502	fRampYN0002
317	F-357	sMTStblzDoneZ4	365	F-503	fRampYN0003
318	F-358	sMTThetaIdleZ4	366	F-504	fRampYN0004
319	F-359	sMTThetaTestZ4	367	F-505	fRampYN0005
320	F-360	sMTThetaDoneZ4	368	F-506	fRampYN0006
321	F-361	sMTTauIdleZ4	369	F-507	fRampYN0007
322	F-362	sMTTauTestZ4	370	F-508	fRampYN0008
323	F-363	sMTTauDoneZ4	371	F-509	fRampYN0009
324	F-364	sMT5TauIdleZ4	372	F-510	fRampYN0010
325	F-365	sMT5TauTestZ4	373	F-511	fRampYN0011
326	F-366	sMT5TauDoneZ4	374	F-512	fRampYN0012
327	F-377	- PID tuning is TBD -	375	F-513	fRampYN0013
328	F-378	sPIDTuneIdle	376	F-514	fRampYN0014
329	F-379	sPIDTuneInit	377	F-515	fRampYN0015
330	F-380	sPIDTuneStabilize	378	F-516	fRampYN0016
331	F-381	sPIDTuneTesting	379	F-517	fRampYN0017
332	F-382	sPIDTuneDone	380	F-518	fRampYN0018
333	F-383	sPIDTuneAbort	381	F-519	fRampYN0019
334	F-385	hfPIDTuneStart	382	F-520	fRampYN0020
335	F-387	placeholder PID tuning	383	F-521	fRampYN0021
336	F-388	placeholder PID tuning0001	384	F-522	fRampYN0022

MHC TND_v3.0.1 Tag Listing

MHC.map

385	F-523	fRampYN0023	433	F-571	fRampYN0071
386	F-524	fRampYN0024	434	F-572	fRampYN0072
387	F-525	fRampYN0025	435	F-573	fRampYN0073
388	F-526	fRampYN0026	436	F-574	fRampYN0074
389	F-527	fRampYN0027	437	F-575	fRampYN0075
390	F-528	fRampYN0028	438	F-576	fRampYN0076
391	F-529	fRampYN0029	439	F-577	fRampYN0077
392	F-530	fRampYN0030	440	F-578	fRampYN0078
393	F-531	fRampYN0031	441	F-579	fRampYN0079
394	F-532	fRampYN0032	442	F-581	fZero0000
395	F-533	fRampYN0033	443	F-582	fZero0001
396	F-534	fRampYN0034	444	F-583	fZero0002
397	F-535	fRampYN0035	445	F-584	fZero0003
398	F-536	fRampYN0036	446	F-585	fZero0004
399	F-537	fRampYN0037	447	F-586	fZero0005
400	F-538	fRampYN0038	448	F-587	fZero0006
401	F-539	fRampYN0039	449	F-588	fZero0007
402	F-540	fRampYN0040	450	F-589	fZero0008
403	F-541	fRampYN0041	451	F-590	fZero0009
404	F-542	fRampYN0042	452	F-591	fZero0010
405	F-543	fRampYN0043	453	F-592	fZero0011
406	F-544	fRampYN0044	454	F-593	fZero0012
407	F-545	fRampYN0045	455	F-594	fZero0013
408	F-546	fRampYN0046	456	F-595	fZero0014
409	F-547	fRampYN0047	457	F-596	fZero0015
410	F-548	fRampYN0048	458	F-597	fZero0016
411	F-549	fRampYN0049	459	F-598	fZero0017
412	F-550	fRampYN0050	460	F-599	fZero0018
413	F-551	fRampYN0051	461	F-600	fZero0019
414	F-552	fRampYN0052	462	F-601	fZero0020
415	F-553	fRampYN0053	463	F-602	fZero0021
416	F-554	fRampYN0054	464	F-603	fZero0022
417	F-555	fRampYN0055	465	F-604	fZero0023
418	F-556	fRampYN0056	466	F-605	fZero0024
419	F-557	fRampYN0057	467	F-606	fZero0025
420	F-558	fRampYN0058	468	F-607	fZero0026
421	F-559	fRampYN0059	469	F-608	fZero0027
422	F-560	fRampYN0060	470	F-609	fZero0028
423	F-561	fRampYN0061	471	F-610	fZero0029
424	F-562	fRampYN0062	472	F-611	fZero0030
425	F-563	fRampYN0063	473	F-612	fZero0031
426	F-564	fRampYN0064	474	F-613	fZero0032
427	F-565	fRampYN0065	475	F-614	fZero0033
428	F-566	fRampYN0066	476	F-615	fZero0034
429	F-567	fRampYN0067	477	F-616	fZero0035
430	F-568	fRampYN0068	478	F-617	fZero0036
431	F-569	fRampYN0069	479	F-618	fZero0037
432	F-570	fRampYN0070	480	F-619	fZero0038

MHC TND_v3.0.1 Tag Listing

MHC.map

481	F-620	fZero0039	529	O-4	y4
482	F-621	fZero0040	530	O-5	y5
483	F-622	fZero0041	531	O-6	y6
484	F-623	fZero0042	532	O-7	y7
485	F-624	fZero0043	533	O-18	--- end OUTPUT definitions ---
486	F-625	fZero0044	534	C-2	cRXSegNumber
487	F-626	fZero0045	535	C-4	cSegNumberZ1
488	F-627	fZero0046	536	C-5	cSegNumberZ2
489	F-628	fZero0047	537	C-6	cSegNumberZ3
490	F-629	fZero0048	538	C-7	cSegNumberZ4
491	F-630	fZero0049	539	C-9	cRXZoneNumber
492	F-631	fZero0050	540	C-20	-- end COUNTER definitions --
493	F-632	fZero0051	541	N-0	nMTLSampleTime_T2
494	F-633	fZero0052	542	N-1	nPIDSampleTime_T1
495	F-634	fZero0053	543	N-2	nMTLAccum
496	F-635	fZero0054	544	N-3	nPIDAccum
497	F-636	fZero0055	545	N-4	hn2_4zoneMachineSwitch
498	F-637	fZero0056	546	N-5	hnControlModeZ1
499	F-638	fZero0057	547	N-6	hnControlModeZ2
500	F-639	fZero0058	548	N-7	hnControlModeZ3
501	F-640	fZero0059	549	N-8	hnControlModeZ4
502	F-641	fZero0060	550	N-10	nControlValve
503	F-642	fZero0061	551	N-11	nFluidConstTemper
504	F-643	fZero0062	552	N-12	nMoldConstTemper
505	F-644	fZero0063	553	N-13	nFluidRecipe
506	F-645	fZero0064	554	N-14	nMoldRecipe
507	F-646	fZero0065	555	N-16	nTemperRangeLowerBound
508	F-647	fZero0066	556	N-17	nTemperRangeUpperBound
509	F-648	fZero0067	557	N-18	nTemperRange
510	F-649	fZero0068	558	N-19	nTemperMinHMIInput
511	F-650	fZero0069	559	N-20	nTC_FluidZ1
512	F-651	fZero0070	560	N-21	nTC_FluidZ2
513	F-652	fZero0071	561	N-22	nTC_FluidZ3
514	F-653	fZero0072	562	N-23	nTC_FluidZ4
515	F-654	fZero0073	563	N-24	nTC_MoldZ1
516	F-655	fZero0074	564	N-25	nTC_MoldZ2
517	F-656	fZero0075	565	N-26	nTC_MoldZ3
518	F-657	fZero0076	566	N-27	nTC_MoldZ4
519	F-658	fZero0077	567	N-28	nTC_CentralHotSupply
520	F-659	fZero0078	568	N-29	nTC_CentralHotReturn
521	F-660	fZero0079	569	N-30	nTC_CentralColdSupply
522	F-662	--- end FLAG definitions ---	570	N-31	nTC_CentralColdReturn
523	T-10	tHMI_Heartbeat	571	N-34	hnUpdateClock
524	T-20	--- end TIMER definitions ---	572	N-35	nPLCClockUpdated
525	O-0	yPressInhibit	573	N-40	nCycleElapsedTime
526	O-1	y1	574	N-42	nSegElapsedTimeZ1
527	O-2	y2	575	N-43	hnSegLastZ1
528	O-3	y3	576	N-44	N-44

MHC TND_v3.0.1 Tag Listing

MHC.map

577	N-45	N-45	625	N-114	nMTL_N_Z2
578	N-46	nSPDwellTimeTargetZ1	626	N-115	hnMTL_Theta_Z2
579	N-47	nSPTemperTargetZ1	627	N-116	hnMTL_Tau_Z2
580	N-48	N-48	628	N-117	hnMTL_Lambda_Z2
581	N-49	nRampYN_Z1	629	N-120	nMTL_N_Z3
582	N-50	hnMoldTemperLimitUpperZ1	630	N-121	hnMTL_Theta_Z3
583	N-51	hnMoldTemperLimitLowerZ1	631	N-122	hnMTL_Tau_Z3
584	N-52	hnSPTemperCycleAbortZ1	632	N-123	hnMTL_Lambda_Z3
585	N-53	hnSPTemperConstZ1	633	N-126	nMTL_N_Z4
586	N-54	hnSPControlValveManualZ1	634	N-127	hnMTL_Theta_Z4
587	N-55	nSP_CV_Z1	635	N-128	hnMTL_Tau_Z4
588	N-58	nSegElapsedTimeZ2	636	N-129	hnMTL_Lambda_Z4
589	N-59	hnSegLastZ2	637	N-141	nCV_Output_Z1
590	N-62	nSPDwellTimeTargetZ2	638	N-142	nCV_Output_Z2
591	N-63	nSPTemperTargetZ2	639	N-143	nCV_Output_Z3
592	N-65	nRampYN_Z2	640	N-144	nCV_Output_Z4
593	N-66	hnMoldTemperLimitUpperZ2	641	N-145	Ch5
594	N-67	hnMoldTemperLimitLowerZ2	642	N-146	Ch6
595	N-68	hnSPTemperCycleAbortZ2	643	N-147	Ch7
596	N-69	hnSPTemperConstZ2	644	N-148	Ch8
597	N-70	hnSPControlValveManualZ2	645	N-151	nZone1
598	N-71	nSP_CV_Z2	646	N-152	nZone2
599	N-74	nSegElapsedTimeZ3	647	N-153	nZone3
600	N-75	hnSegLastZ3	648	N-154	nZone4
601	N-78	nSPDwellTimeTargetZ3	649	N-158	nMTStblzTimeLimit
602	N-79	nSPTemperTargetZ3	650	N-160	hnFluidStartTemperZ1
603	N-81	nRampYN_Z3	651	N-161	hnMoldStartTemperZ1
604	N-82	hnMoldTemperLimitUpperZ3	652	N-162	nMoldStartTemperHiLimitZ1
605	N-83	hnMoldTemperLimitLowerZ3	653	N-163	nMoldStartTemperLoLimitZ1
606	N-84	hnSPTemperCycleAbortZ3	654	N-164	hnMoldEndTemperZ1
607	N-85	hnSPTemperConstZ3	655	N-165	nMTStblzTimeZ1
608	N-86	hnSPControlValveManualZ3	656	N-166	nMTThetaTimeZ1
609	N-87	nSP_CV_Z3	657	N-167	nMTTauTimeZ1
610	N-90	nSegElapsedTimeZ4	658	N-168	nMT5TauTimeTargetZ1
611	N-91	hnSegLastZ4	659	N-169	nMT5TauTimeZ1
612	N-94	nSPDwellTimeTargetZ4	660	N-170	nMTTauTemperZ1
613	N-95	nSPTemperTargetZ4	661	N-172	nMTL_LambdaTune_Z1
614	N-97	nRampYN_Z4	662	N-173	nMTL_NTune_Z1
615	N-98	hnMoldTemperLimitUpperZ4	663	N-180	hnFluidStartTemperZ2
616	N-99	hnMoldTemperLimitLowerZ4	664	N-181	hnMoldStartTemperZ2
617	N-100	hnSPTemperCycleAbortZ4	665	N-182	nMoldStartTemperHiLimitZ2
618	N-101	hnSPTemperConstZ4	666	N-183	nMoldStartTemperLoLimitZ2
619	N-102	hnSPControlValveManualZ4	667	N-184	hnMoldEndTemperZ2
620	N-103	nSP_CV_Z4	668	N-185	nMTStblzTimeZ2
621	N-108	nMTL_N_Z1	669	N-186	nMTThetaTimeZ2
622	N-109	hnMTL_Theta_Z1	670	N-187	nMTTauTimeZ2
623	N-110	hnMTL_Tau_Z1	671	N-188	nMT5TauTimeTargetZ2
624	N-111	hnMTL_Lambda_Z1	672	N-189	nMT5TauTimeZ2

MHC TND_v3.0.1 Tag Listing

MHC.map

673	N-190	nMTTauTemperZ2	718	N-316	hnRXData_TemperSP_2
674	N-192	nMTL_LambdaTune_Z2	719	N-317	hnRXData_DwellTimeSP_3
675	N-193	nMTL_NTune_Z2	720	N-318	hnRXData_TemperSP_3
676	N-200	hnFluidStartTemperZ3	721	N-319	hnRXData_DwellTimeSP_4
677	N-201	hnMoldStartTemperZ3	722	N-320	hnRXData_TemperSP_4
678	N-202	nMoldStartTemperHiLimitZ3	723	N-321	hnRXData_DwellTimeSP_5
679	N-203	nMoldStartTemperLoLimitZ3	724	N-322	hnRXData_TemperSP_5
680	N-204	hnMoldEndTemperZ3	725	N-323	hnRXData_DwellTimeSP_6
681	N-205	nMTStblzTimeZ3	726	N-324	hnRXData_TemperSP_6
682	N-206	nMTThetaTimeZ3	727	N-325	hnRXData_DwellTimeSP_7
683	N-207	nMTTauTimeZ3	728	N-326	hnRXData_TemperSP_7
684	N-208	nMT5TauTimeTargetZ3	729	N-327	hnRXData_DwellTimeSP_8
685	N-209	nMT5TauTimeZ3	730	N-328	hnRXData_TemperSP_8
686	N-210	nMTTauTemperZ3	731	N-329	hnRXData_DwellTimeSP_9
687	N-212	nMTL_LambdaTune_Z3	732	N-330	hnRXData_TemperSP_9
688	N-213	nMTL_NTune_Z3	733	N-500	nDwellTimeSP0000
689	N-220	hnFluidStartTemperZ4	734	N-501	nDwellTimeSP0001
690	N-221	hnMoldStartTemperZ4	735	N-502	nDwellTimeSP0002
691	N-222	nMoldStartTemperHiLimitZ4	736	N-503	nDwellTimeSP0003
692	N-223	nMoldStartTemperLoLimitZ4	737	N-504	nDwellTimeSP0004
693	N-224	hnMoldEndTemperZ4	738	N-505	nDwellTimeSP0005
694	N-225	nMTStblzTimeZ4	739	N-506	nDwellTimeSP0006
695	N-226	nMTThetaTimeZ4	740	N-507	nDwellTimeSP0007
696	N-227	nMTTauTimeZ4	741	N-508	nDwellTimeSP0008
697	N-228	nMT5TauTimeTargetZ4	742	N-509	nDwellTimeSP0009
698	N-229	nMT5TauTimeZ4	743	N-510	nDwellTimeSP0010
699	N-230	nMTTauTemperZ4	744	N-511	nDwellTimeSP0011
700	N-232	nMTL_LambdaTune_Z4	745	N-512	nDwellTimeSP0012
701	N-233	nMTL_NTune_Z4	746	N-513	nDwellTimeSP0013
702	N-299	nRXNumOfSegments	747	N-514	nDwellTimeSP0014
703	N-300	hnRXData_DwellTimeSP_0	748	N-515	nDwellTimeSP0015
704	N-301	hnRXData_TemperSP_0	749	N-516	nDwellTimeSP0016
705	N-302	hnRXData_RampYN	750	N-517	nDwellTimeSP0017
706	N-303		751	N-518	nDwellTimeSP0018
		hnRXData_MoldTemperLimitUpper	752	N-519	nDwellTimeSP0019
707	N-304		753	N-520	nDwellTimeSP0020
		hnRXData_MoldTemperLimitLower	754	N-521	nDwellTimeSP0021
708	N-305		755	N-522	nDwellTimeSP0022
		hnRXData_SPTemperCycleAbort	756	N-523	nDwellTimeSP0023
709	N-306	hnRXData_SPTemperConst	757	N-524	nDwellTimeSP0024
710	N-307	hnRXData_SegLast	758	N-525	nDwellTimeSP0025
711	N-308	hnRXData_CtlMode	759	N-526	nDwellTimeSP0026
712	N-309	hnRXData_Theta	760	N-527	nDwellTimeSP0027
713	N-310	hnRXData_Tau	761	N-528	nDwellTimeSP0028
714	N-311	hnRXData_Lambda	762	N-529	nDwellTimeSP0029
715	N-313	hnRXData_DwellTimeSP_1	763	N-530	nDwellTimeSP0030
716	N-314	hnRXData_TemperSP_1	764	N-531	nDwellTimeSP0031
717	N-315	hnRXData_DwellTimeSP_2	765	N-532	nDwellTimeSP0032

MHC TND_v3.0.1 Tag Listing

MHC.map

766	N-533	nDwellTimeSP0033	814	N-582	nTemperSP0001
767	N-534	nDwellTimeSP0034	815	N-583	nTemperSP0002
768	N-535	nDwellTimeSP0035	816	N-584	nTemperSP0003
769	N-536	nDwellTimeSP0036	817	N-585	nTemperSP0004
770	N-537	nDwellTimeSP0037	818	N-586	nTemperSP0005
771	N-538	nDwellTimeSP0038	819	N-587	nTemperSP0006
772	N-539	nDwellTimeSP0039	820	N-588	nTemperSP0007
773	N-540	nDwellTimeSP0040	821	N-589	nTemperSP0008
774	N-541	nDwellTimeSP0041	822	N-590	nTemperSP0009
775	N-542	nDwellTimeSP0042	823	N-591	nTemperSP0010
776	N-543	nDwellTimeSP0043	824	N-592	nTemperSP0011
777	N-544	nDwellTimeSP0044	825	N-593	nTemperSP0012
778	N-545	nDwellTimeSP0045	826	N-594	nTemperSP0013
779	N-546	nDwellTimeSP0046	827	N-595	nTemperSP0014
780	N-547	nDwellTimeSP0047	828	N-596	nTemperSP0015
781	N-548	nDwellTimeSP0048	829	N-597	nTemperSP0016
782	N-549	nDwellTimeSP0049	830	N-598	nTemperSP0017
783	N-550	nDwellTimeSP0050	831	N-599	nTemperSP0018
784	N-551	nDwellTimeSP0051	832	N-600	nTemperSP0019
785	N-552	nDwellTimeSP0052	833	N-601	nTemperSP0020
786	N-553	nDwellTimeSP0053	834	N-602	nTemperSP0021
787	N-554	nDwellTimeSP0054	835	N-603	nTemperSP0022
788	N-555	nDwellTimeSP0055	836	N-604	nTemperSP0023
789	N-556	nDwellTimeSP0056	837	N-605	nTemperSP0024
790	N-557	nDwellTimeSP0057	838	N-606	nTemperSP0025
791	N-558	nDwellTimeSP0058	839	N-607	nTemperSP0026
792	N-559	nDwellTimeSP0059	840	N-608	nTemperSP0027
793	N-560	nDwellTimeSP0060	841	N-609	nTemperSP0028
794	N-561	nDwellTimeSP0061	842	N-610	nTemperSP0029
795	N-562	nDwellTimeSP0062	843	N-611	nTemperSP0030
796	N-563	nDwellTimeSP0063	844	N-612	nTemperSP0031
797	N-564	nDwellTimeSP0064	845	N-613	nTemperSP0032
798	N-565	nDwellTimeSP0065	846	N-614	nTemperSP0033
799	N-566	nDwellTimeSP0066	847	N-615	nTemperSP0034
800	N-567	nDwellTimeSP0067	848	N-616	nTemperSP0035
801	N-568	nDwellTimeSP0068	849	N-617	nTemperSP0036
802	N-569	nDwellTimeSP0069	850	N-618	nTemperSP0037
803	N-570	nDwellTimeSP0070	851	N-619	nTemperSP0038
804	N-571	nDwellTimeSP0071	852	N-620	nTemperSP0039
805	N-572	nDwellTimeSP0072	853	N-621	nTemperSP0040
806	N-573	nDwellTimeSP0073	854	N-622	nTemperSP0041
807	N-574	nDwellTimeSP0074	855	N-623	nTemperSP0042
808	N-575	nDwellTimeSP0075	856	N-624	nTemperSP0043
809	N-576	nDwellTimeSP0076	857	N-625	nTemperSP0044
810	N-577	nDwellTimeSP0077	858	N-626	nTemperSP0045
811	N-578	nDwellTimeSP0078	859	N-627	nTemperSP0046
812	N-579	nDwellTimeSP0079	860	N-628	nTemperSP0047
813	N-581	nTemperSP0000	861	N-629	nTemperSP0048

MHC TND_v3.0.1 Tag Listing

MHC.map

862	N-630	nTemperSP0049	910	N-679	nZero0017
863	N-631	nTemperSP0050	911	N-680	nZero0018
864	N-632	nTemperSP0051	912	N-681	nZero0019
865	N-633	nTemperSP0052	913	N-682	nZero0020
866	N-634	nTemperSP0053	914	N-683	nZero0021
867	N-635	nTemperSP0054	915	N-684	nZero0022
868	N-636	nTemperSP0055	916	N-685	nZero0023
869	N-637	nTemperSP0056	917	N-686	nZero0024
870	N-638	nTemperSP0057	918	N-687	nZero0025
871	N-639	nTemperSP0058	919	N-688	nZero0026
872	N-640	nTemperSP0059	920	N-689	nZero0027
873	N-641	nTemperSP0060	921	N-690	nZero0028
874	N-642	nTemperSP0061	922	N-691	nZero0029
875	N-643	nTemperSP0062	923	N-692	nZero0030
876	N-644	nTemperSP0063	924	N-693	nZero0031
877	N-645	nTemperSP0064	925	N-694	nZero0032
878	N-646	nTemperSP0065	926	N-695	nZero0033
879	N-647	nTemperSP0066	927	N-696	nZero0034
880	N-648	nTemperSP0067	928	N-697	nZero0035
881	N-649	nTemperSP0068	929	N-698	nZero0036
882	N-650	nTemperSP0069	930	N-699	nZero0037
883	N-651	nTemperSP0070	931	N-700	nZero0038
884	N-652	nTemperSP0071	932	N-701	nZero0039
885	N-653	nTemperSP0072	933	N-702	nZero0040
886	N-654	nTemperSP0073	934	N-703	nZero0041
887	N-655	nTemperSP0074	935	N-704	nZero0042
888	N-656	nTemperSP0075	936	N-705	nZero0043
889	N-657	nTemperSP0076	937	N-706	nZero0044
890	N-658	nTemperSP0077	938	N-707	nZero0045
891	N-659	nTemperSP0078	939	N-708	nZero0046
892	N-660	nTemperSP0079	940	N-709	nZero0047
893	N-662	nZero0000	941	N-710	nZero0048
894	N-663	nZero0001	942	N-711	nZero0049
895	N-664	nZero0002	943	N-712	nZero0050
896	N-665	nZero0003	944	N-713	nZero0051
897	N-666	nZero0004	945	N-714	nZero0052
898	N-667	nZero0005	946	N-715	nZero0053
899	N-668	nZero0006	947	N-716	nZero0054
900	N-669	nZero0007	948	N-717	nZero0055
901	N-670	nZero0008	949	N-718	nZero0056
902	N-671	nZero0009	950	N-719	nZero0057
903	N-672	nZero0010	951	N-720	nZero0058
904	N-673	nZero0011	952	N-721	nZero0059
905	N-674	nZero0012	953	N-722	nZero0060
906	N-675	nZero0013	954	N-723	nZero0061
907	N-676	nZero0014	955	N-724	nZero0062
908	N-677	nZero0015	956	N-725	nZero0063
909	N-678	nZero0016	957	N-726	nZero0064

MHC TND_v3.0.1 Tag Listing

MHC.map

958	N-727	nZero0065	1002	FP-49	rMTL_P1_Z1
959	N-728	nZero0066	1003	FP-50	rMTL_P5_Z1
960	N-729	nZero0067	1004	FP-51	rMTL_P8_Z1
961	N-730	nZero0068	1005	FP-52	rMTL_P9_Z1
962	N-731	nZero0069	1006	FP-55	rMTL_KprimeTune_Z1
963	N-732	nZero0070	1007	FP-56	rMTL_KmTune_Z1
964	N-733	nZero0071	1008	FP-57	rMTL_KcsTune_Z1
965	N-734	nZero0072	1009	FP-58	rMTL_P0Tune_Z1
966	N-735	nZero0073	1010	FP-59	rMTL_P1Tune_Z1
967	N-736	nZero0074	1011	FP-60	rMTL_P5Tune_Z1
968	N-737	nZero0075	1012	FP-61	rMTL_P8Tune_Z1
969	N-738	nZero0076	1013	FP-62	rMTL_P9Tune_Z1
970	N-739	nZero0077	1014	FP-67	rMTLError_e.m_Z1
971	N-740	nZero0078	1015	FP-68	rMTLError_e.m-1_Z1
972	N-741	nZero0079	1016	FP-69	rMTLError_e.m-2_Z1
973	N-743	--- end NUMBER definitions ---	1017	FP-70	rMTL_ISE_Z1
974		All RMUs	1018	FP-73	rMTLOutput_M.m_Z1
975	FP-9		1019	FP-74	rMTLOutput_M.m-1_Z1
		rMTLOuput_M.m_UpperBound	1020	FP-75	rMTLOutput_M.m-N-1_Z1
976	FP-10		1021	FP-76	rMTLOutput_M.m-N-2_Z1
		rMTLOuput_M.m_LowerBound	1022	FP-93	hrPID_Kp_Z2
977	FP-12	rPIDOutput_F.n_UpperBound	1023	FP-94	hrPID_Ki_Z2
978	FP-13	rPIDOutput_F.n_LowerBound	1024	FP-95	hrPID_Kd_Z2
979	FP-14		1025	FP-96	rSP_PID_Z2
		rPIDError_Integral_UpperBound	1026	FP-97	rSPpr_PID_Z2
980	FP-15		1027	FP-98	rSPCascadeZ2
		rPIDError_Integral_LowerBound	1028	FP-99	rPIDPropPortionZ2
981	FP-26	hrPID_Kp_Z1	1029	FP-100	rPIDIntegPortionZ2
982	FP-27	hrPID_Ki_Z1	1030	FP-101	rPIDDerivPortionZ2
983	FP-28	hrPID_Kd_Z1	1031	FP-102	rPIDError_e.n_Z2
984	FP-29	rSP_PID_Z1	1032	FP-103	rPIDError_e.n-1_Z2
985	FP-30	rSPpr_PID_Z1	1033	FP-105	rPIDOutput_F.n_Z2
986	FP-31	rSPCascadeZ1	1034	FP-106	rPID_ISE_Z2
987	FP-32	rPIDPropPortionZ1	1035	FP-107	rSPRampRateZ2
988	FP-33	rPIDIntegPortionZ1	1036	FP-108	rSPTemperRampZ2
989	FP-34	rPIDDerivPortionZ1	1037	FP-110	rSP_MTL_Z2
990	FP-35	rPIDError_e.n_Z1	1038	FP-111	rSPpr_MTL_Z2
991	FP-36	rPIDError_e.n-1_Z1	1039	FP-112	hrMTL_Kprime_Z2
992	FP-38	rPIDOutput_F.n_Z1	1040	FP-113	hrMTL_Km_Z2
993	FP-39	rPID_ISE_Z1	1041	FP-114	rMTL_Kcs_Z2
994	FP-40	rSPRampRateZ1	1042	FP-115	rMTL_P0_Z2
995	FP-41	rSPTemperRampZ1	1043	FP-116	rMTL_P1_Z2
996	FP-43	rSP_MTL_Z1	1044	FP-117	rMTL_P5_Z2
997	FP-44	rSPpr_MTL_Z1	1045	FP-118	rMTL_P8_Z2
998	FP-45	hrMTL_Kprime_Z1	1046	FP-119	rMTL_P9_Z2
999	FP-46	hrMTL_Km_Z1	1047	FP-122	rMTL_KprimeTune_Z2
1000	FP-47	rMTL_Kcs_Z1	1048	FP-123	rMTL_KmTune_Z2
1001	FP-48	rMTL_P0_Z1	1049	FP-124	rMTL_KcsTune_Z2

MHC TND_v3.0.1 Tag Listing

MHC.map

1050	FP-125	rMTL_P0Tune_Z2	1098	FP-203	rMTLError_e.m-2_Z3
1051	FP-126	rMTL_P1Tune_Z2	1099	FP-204	rMTL_ISE_Z3
1052	FP-127	rMTL_P5Tune_Z2	1100	FP-207	rMTLOutput_M.m_Z3
1053	FP-128	rMTL_P8Tune_Z2	1101	FP-208	rMTLOutput_M.m-1_Z3
1054	FP-129	rMTL_P9Tune_Z2	1102	FP-209	rMTLOutput_M.m-N-1_Z3
1055	FP-134	rMTLError_e.m_Z2	1103	FP-210	rMTLOutput_M.m-N-2_Z3
1056	FP-135	rMTLError_e.m-1_Z2	1104	FP-227	hrPID_Kp_Z4
1057	FP-136	rMTLError_e.m-2_Z2	1105	FP-228	hrPID_Ki_Z4
1058	FP-137	rMTL_ISE_Z2	1106	FP-229	hrPID_Kd_Z4
1059	FP-140	rMTLOutput_M.m_Z2	1107	FP-230	rSP_PID_Z4
1060	FP-141	rMTLOutput_M.m-1_Z2	1108	FP-231	rSPpr_PID_Z4
1061	FP-142	rMTLOutput_M.m-N-1_Z2	1109	FP-232	rSPCascadeZ4
1062	FP-143	rMTLOutput_M.m-N-2_Z2	1110	FP-233	rPIDPropPortionZ4
1063	FP-160	hrPID_Kp_Z3	1111	FP-234	rPIDIntegPortionZ4
1064	FP-161	hrPID_Ki_Z3	1112	FP-235	rPIDDerivPortionZ4
1065	FP-162	hrPID_Kd_Z3	1113	FP-236	rPIDError_e.n_Z4
1066	FP-163	rSP_PID_Z3	1114	FP-237	rPIDError_e.n-1_Z4
1067	FP-164	rSPpr_PID_Z3	1115	FP-239	rPIDOutput_F.n_Z4
1068	FP-165	rSPCascadeZ3	1116	FP-240	rPID_ISE_Z4
1069	FP-166	rPIDPropPortionZ3	1117	FP-241	rSPRampRateZ4
1070	FP-167	rPIDIntegPortionZ3	1118	FP-242	rSPTemperRampZ4
1071	FP-168	rPIDDerivPortionZ3	1119	FP-244	rSP_MTL_Z4
1072	FP-169	rPIDError_e.n_Z3	1120	FP-245	rSPpr_MTL_Z4
1073	FP-170	rPIDError_e.n-1_Z3	1121	FP-246	hrMTL_Kprime_Z4
1074	FP-172	rPIDOutput_F.n_Z3	1122	FP-247	hrMTL_Km_Z4
1075	FP-173	rPID_ISE_Z3	1123	FP-248	rMTL_Kcs_Z4
1076	FP-174	rSPRampRateZ3	1124	FP-249	rMTL_P0_Z4
1077	FP-175	rSPTemperRampZ3	1125	FP-250	rMTL_P1_Z4
1078	FP-177	rSP_MTL_Z3	1126	FP-251	rMTL_P5_Z4
1079	FP-178	rSPpr_MTL_Z3	1127	FP-252	rMTL_P8_Z4
1080	FP-179	hrMTL_Kprime_Z3	1128	FP-253	rMTL_P9_Z4
1081	FP-180	hrMTL_Km_Z3	1129	FP-256	rMTL_KprimeTune_Z4
1082	FP-181	rMTL_Kcs_Z3	1130	FP-257	rMTL_KmTune_Z4
1083	FP-182	rMTL_P0_Z3	1131	FP-258	rMTL_KcsTune_Z4
1084	FP-183	rMTL_P1_Z3	1132	FP-259	rMTL_P0Tune_Z4
1085	FP-184	rMTL_P5_Z3	1133	FP-260	rMTL_P1Tune_Z4
1086	FP-185	rMTL_P8_Z3	1134	FP-261	rMTL_P5Tune_Z4
1087	FP-186	rMTL_P9_Z3	1135	FP-262	rMTL_P8Tune_Z4
1088	FP-189	rMTL_KprimeTune_Z3	1136	FP-263	rMTL_P9Tune_Z4
1089	FP-190	rMTL_KmTune_Z3	1137	FP-268	rMTLError_e.m_Z4
1090	FP-191	rMTL_KcsTune_Z3	1138	FP-269	rMTLError_e.m-1_Z4
1091	FP-192	rMTL_P0Tune_Z3	1139	FP-270	rMTLError_e.m-2_Z4
1092	FP-193	rMTL_P1Tune_Z3	1140	FP-271	rMTL_ISE_Z4
1093	FP-194	rMTL_P5Tune_Z3	1141	FP-274	rMTLOutput_M.m_Z4
1094	FP-195	rMTL_P8Tune_Z3	1142	FP-275	rMTLOutput_M.m-1_Z4
1095	FP-196	rMTL_P9Tune_Z3	1143	FP-276	rMTLOutput_M.m-N-1_Z4
1096	FP-201	rMTLError_e.m_Z3	1144	FP-277	rMTLOutput_M.m-N-2_Z4
1097	FP-202	rMTLError_e.m-1_Z3	1145	FP-302	hrRXData_Kprime

MHC TND_v3.0.1 Tag Listing

MHC.map

1146	FP-303	hrRXData_Km	1194	FP-625	rZero0044
1147	FP-310	hrRXData_Kp	1195	FP-626	rZero0045
1148	FP-311	hrRXData_Ki	1196	FP-627	rZero0046
1149	FP-312	hrRXData_Kd	1197	FP-628	rZero0047
1150	FP-581	rZero0000	1198	FP-629	rZero0048
1151	FP-582	rZero0001	1199	FP-630	rZero0049
1152	FP-583	rZero0002	1200	FP-631	rZero0050
1153	FP-584	rZero0003	1201	FP-632	rZero0051
1154	FP-585	rZero0004	1202	FP-633	rZero0052
1155	FP-586	rZero0005	1203	FP-634	rZero0053
1156	FP-587	rZero0006	1204	FP-635	rZero0054
1157	FP-588	rZero0007	1205	FP-636	rZero0055
1158	FP-589	rZero0008	1206	FP-637	rZero0056
1159	FP-590	rZero0009	1207	FP-638	rZero0057
1160	FP-591	rZero0010	1208	FP-639	rZero0058
1161	FP-592	rZero0011	1209	FP-640	rZero0059
1162	FP-593	rZero0012	1210	FP-641	rZero0060
1163	FP-594	rZero0013	1211	FP-642	rZero0061
1164	FP-595	rZero0014	1212	FP-643	rZero0062
1165	FP-596	rZero0015	1213	FP-644	rZero0063
1166	FP-597	rZero0016	1214	FP-645	rZero0064
1167	FP-598	rZero0017	1215	FP-646	rZero0065
1168	FP-599	rZero0018	1216	FP-647	rZero0066
1169	FP-600	rZero0019	1217	FP-648	rZero0067
1170	FP-601	rZero0020	1218	FP-649	rZero0068
1171	FP-602	rZero0021	1219	FP-650	rZero0069
1172	FP-603	rZero0022	1220	FP-651	rZero0070
1173	FP-604	rZero0023	1221	FP-652	rZero0071
1174	FP-605	rZero0024	1222	FP-653	rZero0072
1175	FP-606	rZero0025	1223	FP-654	rZero0073
1176	FP-607	rZero0026	1224	FP-655	rZero0074
1177	FP-608	rZero0027	1225	FP-656	rZero0075
1178	FP-609	rZero0028	1226	FP-657	rZero0076
1179	FP-610	rZero0029	1227	FP-658	rZero0077
1180	FP-611	rZero0030	1228	FP-659	rZero0078
1181	FP-612	rZero0031	1229	FP-660	rZero0079
1182	FP-613	rZero0032	1230	FP-661	rZero0080
1183	FP-614	rZero0033	1231	FP-700	rMTLOutputArrayZ1_0000
1184	FP-615	rZero0034	1232	FP-701	rMTLOutputArrayZ1_0001
1185	FP-616	rZero0035	1233	FP-702	rMTLOutputArrayZ1_0002
1186	FP-617	rZero0036	1234	FP-703	rMTLOutputArrayZ1_0003
1187	FP-618	rZero0037	1235	FP-704	rMTLOutputArrayZ1_0004
1188	FP-619	rZero0038	1236	FP-705	rMTLOutputArrayZ1_0005
1189	FP-620	rZero0039	1237	FP-706	rMTLOutputArrayZ1_0006
1190	FP-621	rZero0040	1238	FP-707	rMTLOutputArrayZ1_0007
1191	FP-622	rZero0041	1239	FP-708	rMTLOutputArrayZ1_0008
1192	FP-623	rZero0042	1240	FP-709	rMTLOutputArrayZ1_0009
1193	FP-624	rZero0043	1241	FP-710	rMTLOutputArrayZ1_0010

MHC TND_v3.0.1 Tag Listing

MHC.map

1242	FP-711	rMTLOutputArrayZ1_0011	1290	FP-808	rMTLOutputArrayZ2_0008
1243	FP-712	rMTLOutputArrayZ1_0012	1291	FP-809	rMTLOutputArrayZ2_0009
1244	FP-713	rMTLOutputArrayZ1_0013	1292	FP-810	rMTLOutputArrayZ2_0010
1245	FP-714	rMTLOutputArrayZ1_0014	1293	FP-811	rMTLOutputArrayZ2_0011
1246	FP-715	rMTLOutputArrayZ1_0015	1294	FP-812	rMTLOutputArrayZ2_0012
1247	FP-716	rMTLOutputArrayZ1_0016	1295	FP-813	rMTLOutputArrayZ2_0013
1248	FP-717	rMTLOutputArrayZ1_0017	1296	FP-814	rMTLOutputArrayZ2_0014
1249	FP-718	rMTLOutputArrayZ1_0018	1297	FP-815	rMTLOutputArrayZ2_0015
1250	FP-719	rMTLOutputArrayZ1_0019	1298	FP-816	rMTLOutputArrayZ2_0016
1251	FP-720	rMTLOutputArrayZ1_0020	1299	FP-817	rMTLOutputArrayZ2_0017
1252	FP-721	rMTLOutputArrayZ1_0021	1300	FP-818	rMTLOutputArrayZ2_0018
1253	FP-722	rMTLOutputArrayZ1_0022	1301	FP-819	rMTLOutputArrayZ2_0019
1254	FP-723	rMTLOutputArrayZ1_0023	1302	FP-820	rMTLOutputArrayZ2_0020
1255	FP-724	rMTLOutputArrayZ1_0024	1303	FP-821	rMTLOutputArrayZ2_0021
1256	FP-725	rMTLOutputArrayZ1_0025	1304	FP-822	rMTLOutputArrayZ2_0022
1257	FP-726	rMTLOutputArrayZ1_0026	1305	FP-823	rMTLOutputArrayZ2_0023
1258	FP-727	rMTLOutputArrayZ1_0027	1306	FP-824	rMTLOutputArrayZ2_0024
1259	FP-728	rMTLOutputArrayZ1_0028	1307	FP-825	rMTLOutputArrayZ2_0025
1260	FP-729	rMTLOutputArrayZ1_0029	1308	FP-826	rMTLOutputArrayZ2_0026
1261	FP-730	rMTLOutputArrayZ1_0030	1309	FP-827	rMTLOutputArrayZ2_0027
1262	FP-731	rMTLOutputArrayZ1_0031	1310	FP-828	rMTLOutputArrayZ2_0028
1263	FP-732	rMTLOutputArrayZ1_0032	1311	FP-829	rMTLOutputArrayZ2_0029
1264	FP-733	rMTLOutputArrayZ1_0033	1312	FP-830	rMTLOutputArrayZ2_0030
1265	FP-734	rMTLOutputArrayZ1_0034	1313	FP-831	rMTLOutputArrayZ2_0031
1266	FP-735	rMTLOutputArrayZ1_0035	1314	FP-832	rMTLOutputArrayZ2_0032
1267	FP-736	rMTLOutputArrayZ1_0036	1315	FP-833	rMTLOutputArrayZ2_0033
1268	FP-737	rMTLOutputArrayZ1_0037	1316	FP-834	rMTLOutputArrayZ2_0034
1269	FP-738	rMTLOutputArrayZ1_0038	1317	FP-835	rMTLOutputArrayZ2_0035
1270	FP-739	rMTLOutputArrayZ1_0039	1318	FP-836	rMTLOutputArrayZ2_0036
1271	FP-740	rMTLOutputArrayZ1_0040	1319	FP-837	rMTLOutputArrayZ2_0037
1272	FP-741	rMTLOutputArrayZ1_0041	1320	FP-838	rMTLOutputArrayZ2_0038
1273	FP-742	rMTLOutputArrayZ1_0042	1321	FP-839	rMTLOutputArrayZ2_0039
1274	FP-743	rMTLOutputArrayZ1_0043	1322	FP-840	rMTLOutputArrayZ2_0040
1275	FP-744	rMTLOutputArrayZ1_0044	1323	FP-841	rMTLOutputArrayZ2_0041
1276	FP-745	rMTLOutputArrayZ1_0045	1324	FP-842	rMTLOutputArrayZ2_0042
1277	FP-746	rMTLOutputArrayZ1_0046	1325	FP-843	rMTLOutputArrayZ2_0043
1278	FP-747	rMTLOutputArrayZ1_0047	1326	FP-844	rMTLOutputArrayZ2_0044
1279	FP-748	rMTLOutputArrayZ1_0048	1327	FP-845	rMTLOutputArrayZ2_0045
1280	FP-749	rMTLOutputArrayZ1_0049	1328	FP-846	rMTLOutputArrayZ2_0046
1281	FP-750	rMTLOutputArrayZ1_0050	1329	FP-847	rMTLOutputArrayZ2_0047
1282	FP-800	rMTLOutputArrayZ2_0000	1330	FP-848	rMTLOutputArrayZ2_0048
1283	FP-801	rMTLOutputArrayZ2_0001	1331	FP-849	rMTLOutputArrayZ2_0049
1284	FP-802	rMTLOutputArrayZ2_0002	1332	FP-850	rMTLOutputArrayZ2_0050
1285	FP-803	rMTLOutputArrayZ2_0003	1333	FP-900	rMTLOutputArrayZ3_0000
1286	FP-804	rMTLOutputArrayZ2_0004	1334	FP-901	rMTLOutputArrayZ3_0001
1287	FP-805	rMTLOutputArrayZ2_0005	1335	FP-902	rMTLOutputArrayZ3_0002
1288	FP-806	rMTLOutputArrayZ2_0006	1336	FP-903	rMTLOutputArrayZ3_0003
1289	FP-807	rMTLOutputArrayZ2_0007	1337	FP-904	rMTLOutputArrayZ3_0004

MHC TND_v3.0.1 Tag Listing

MHC.map

1338	FP-905	rMTLOutputArrayZ3_0005	1386	FP-1002	rMTLOutputArrayZ4_0002
1339	FP-906	rMTLOutputArrayZ3_0006	1387	FP-1003	rMTLOutputArrayZ4_0003
1340	FP-907	rMTLOutputArrayZ3_0007	1388	FP-1004	rMTLOutputArrayZ4_0004
1341	FP-908	rMTLOutputArrayZ3_0008	1389	FP-1005	rMTLOutputArrayZ4_0005
1342	FP-909	rMTLOutputArrayZ3_0009	1390	FP-1006	rMTLOutputArrayZ4_0006
1343	FP-910	rMTLOutputArrayZ3_0010	1391	FP-1007	rMTLOutputArrayZ4_0007
1344	FP-911	rMTLOutputArrayZ3_0011	1392	FP-1008	rMTLOutputArrayZ4_0008
1345	FP-912	rMTLOutputArrayZ3_0012	1393	FP-1009	rMTLOutputArrayZ4_0009
1346	FP-913	rMTLOutputArrayZ3_0013	1394	FP-1010	rMTLOutputArrayZ4_0010
1347	FP-914	rMTLOutputArrayZ3_0014	1395	FP-1011	rMTLOutputArrayZ4_0011
1348	FP-915	rMTLOutputArrayZ3_0015	1396	FP-1012	rMTLOutputArrayZ4_0012
1349	FP-916	rMTLOutputArrayZ3_0016	1397	FP-1013	rMTLOutputArrayZ4_0013
1350	FP-917	rMTLOutputArrayZ3_0017	1398	FP-1014	rMTLOutputArrayZ4_0014
1351	FP-918	rMTLOutputArrayZ3_0018	1399	FP-1015	rMTLOutputArrayZ4_0015
1352	FP-919	rMTLOutputArrayZ3_0019	1400	FP-1016	rMTLOutputArrayZ4_0016
1353	FP-920	rMTLOutputArrayZ3_0020	1401	FP-1017	rMTLOutputArrayZ4_0017
1354	FP-921	rMTLOutputArrayZ3_0021	1402	FP-1018	rMTLOutputArrayZ4_0018
1355	FP-922	rMTLOutputArrayZ3_0022	1403	FP-1019	rMTLOutputArrayZ4_0019
1356	FP-923	rMTLOutputArrayZ3_0023	1404	FP-1020	rMTLOutputArrayZ4_0020
1357	FP-924	rMTLOutputArrayZ3_0024	1405	FP-1021	rMTLOutputArrayZ4_0021
1358	FP-925	rMTLOutputArrayZ3_0025	1406	FP-1022	rMTLOutputArrayZ4_0022
1359	FP-926	rMTLOutputArrayZ3_0026	1407	FP-1023	rMTLOutputArrayZ4_0023
1360	FP-927	rMTLOutputArrayZ3_0027	1408	FP-1024	rMTLOutputArrayZ4_0024
1361	FP-928	rMTLOutputArrayZ3_0028	1409	FP-1025	rMTLOutputArrayZ4_0025
1362	FP-929	rMTLOutputArrayZ3_0029	1410	FP-1026	rMTLOutputArrayZ4_0026
1363	FP-930	rMTLOutputArrayZ3_0030	1411	FP-1027	rMTLOutputArrayZ4_0027
1364	FP-931	rMTLOutputArrayZ3_0031	1412	FP-1028	rMTLOutputArrayZ4_0028
1365	FP-932	rMTLOutputArrayZ3_0032	1413	FP-1029	rMTLOutputArrayZ4_0029
1366	FP-933	rMTLOutputArrayZ3_0033	1414	FP-1030	rMTLOutputArrayZ4_0030
1367	FP-934	rMTLOutputArrayZ3_0034	1415	FP-1031	rMTLOutputArrayZ4_0031
1368	FP-935	rMTLOutputArrayZ3_0035	1416	FP-1032	rMTLOutputArrayZ4_0032
1369	FP-936	rMTLOutputArrayZ3_0036	1417	FP-1033	rMTLOutputArrayZ4_0033
1370	FP-937	rMTLOutputArrayZ3_0037	1418	FP-1034	rMTLOutputArrayZ4_0034
1371	FP-938	rMTLOutputArrayZ3_0038	1419	FP-1035	rMTLOutputArrayZ4_0035
1372	FP-939	rMTLOutputArrayZ3_0039	1420	FP-1036	rMTLOutputArrayZ4_0036
1373	FP-940	rMTLOutputArrayZ3_0040	1421	FP-1037	rMTLOutputArrayZ4_0037
1374	FP-941	rMTLOutputArrayZ3_0041	1422	FP-1038	rMTLOutputArrayZ4_0038
1375	FP-942	rMTLOutputArrayZ3_0042	1423	FP-1039	rMTLOutputArrayZ4_0039
1376	FP-943	rMTLOutputArrayZ3_0043	1424	FP-1040	rMTLOutputArrayZ4_0040
1377	FP-944	rMTLOutputArrayZ3_0044	1425	FP-1041	rMTLOutputArrayZ4_0041
1378	FP-945	rMTLOutputArrayZ3_0045	1426	FP-1042	rMTLOutputArrayZ4_0042
1379	FP-946	rMTLOutputArrayZ3_0046	1427	FP-1043	rMTLOutputArrayZ4_0043
1380	FP-947	rMTLOutputArrayZ3_0047	1428	FP-1044	rMTLOutputArrayZ4_0044
1381	FP-948	rMTLOutputArrayZ3_0048	1429	FP-1045	rMTLOutputArrayZ4_0045
1382	FP-949	rMTLOutputArrayZ3_0049	1430	FP-1046	rMTLOutputArrayZ4_0046
1383	FP-950	rMTLOutputArrayZ3_0050	1431	FP-1047	rMTLOutputArrayZ4_0047
1384	FP-1000	rMTLOutputArrayZ4_0000	1432	FP-1048	rMTLOutputArrayZ4_0048
1385	FP-1001	rMTLOutputArrayZ4_0001	1433	FP-1049	rMTLOutputArrayZ4_0049

MHC TND_v3.0.1 Tag Listing

MHC.map

1434	FP-1050	rMTLOutputArrayZ4_0050	1482	SYS-46	ExtendedFunctionBlockError
1435	FP-1052	--- end FLOAT definitions ---	1483	SYS-47	ScreenViewDataEntryDisable
1436	SYS-0	ScanInterval	1484	SYS-48	Reserved48
1437	SYS-1	PeakScanInterval	1485	SYS-49	Reserved49
1438	SYS-2	AvgScanInterval	1486	SYS-50	
1439	SYS-3	OverScanRatio			LastExtendedFunctionBlockTime
1440	SYS-4	LogicTime	1487	SYS-51	
1441	SYS-5	PeakLogicTime			PeakExtendedFunctionBlockTime
1442	SYS-6	AvgLogicTime	1488	SYS-52	
1443	SYS-7	Reserved7			AvgExtendedFunctionBlockTime
1444	SYS-8	Reserved8	1489	SYS-53	Spare1
1445	SYS-9	Reserved9	1490	SYS-54	Spare2
1446	SYS-10	Reserved10	1491	SYS-55	Spare3
1447	SYS-11	ArrayIndexError	1492	SYS-56	Spare4
1448	SYS-12	DoLoopExit	1493	SYS-57	LiveCaptureStatus
1449	SYS-13	ScanCount	1494	Arr-0	Array_nDwellTimeSP
1450	SYS-14	Reserved14	1495	Arr-1	Array_nTemperSP
1451	SYS-15	DataLoggingStatus	1496	Arr-2	Array_fRampYN
1452	SYS-16	RollingMsecCounter	1497	Arr-5	Array_nZero
1453	SYS-17	LastShutdown	1498	Arr-6	Array_fZero
1454	SYS-18	Reserved18	1499	Arr-7	Array_rZero
1455	SYS-19	PowerFailShutdown	1500	Arr-11	Array_rMTLOutputZ1
1456	SYS-20	ScreenViewCurrentPassword	1501	Arr-12	Array_rMTLOutputZ2
1457	SYS-21	ScreenViewLevel1Password	1502	Arr-13	Array_rMTLOutputZ3
1458	SYS-22	ScreenViewLevel2Password	1503	Arr-14	Array_rMTLOutputZ4
1459	SYS-23	ScreenViewLevel3Password	1504	Arr-15	Array_rMTLErrorZ1
1460	SYS-24	ScreenViewLevel4Password	1505	Arr-16	Array_rMTLErrorZ2
1461	SYS-25	ScreenViewLevel5Password	1506	Arr-17	Array_rMTLErrorZ3
1462	SYS-26	ScreenViewLevel6Password	1507	Arr-18	Array_rMTLErrorZ4
1463	SYS-27	ScreenViewLevel7Password	1508	Str-0	wMHC_PLC_VersionRevision
1464	SYS-28	Reserved28	1509	Str-1	hwRecipeName
1465	SYS-29	OverScans	1510	Str-10	--- end STRING definitions ---
1466	SYS-30	LastIOTime			
1467	SYS-31	PeakIOTime			
1468	SYS-32	MinClock			
1469	SYS-33	SecClock			
1470	SYS-34	100msClock			
1471	SYS-35	50msClock			
1472	SYS-36	FirstScan			
1473	SYS-37	AltScan			
1474	SYS-38	IOStatus			
1475	SYS-39	MathError			
1476	SYS-40	ScreenViewDataEntryKeypad			
1477	SYS-41	StringError			
1478	SYS-42	ConversionError			
1479	SYS-43	WriteRetentiveData			
1480	SYS-44	Reserved44			
1481	SYS-45	Reserved45			

Flowchart : Documentation

Printed: Thursday, April 09, 2009 @ 11:11
Project: MHC_TND_v3.0.1

0	This chart will contain an index of documentation for the PLC portion of the project.				need to have this dummy chart complete in order for the project to compile correctly
	---	---	---	---	
1	V3.0.0 date 10/31/2008 JAS -initial release	- see example flowcharts on which the standards were developed to translate a drawn state set into a TND flowchart. Flowcharts are on the "Unused" tab of the ProjectBinder, with an "Example: ..." prefix.	COMMENT BLOCK	COMMENT BLOCK
	V3.0.1 date 04/08/2009 JAS - affected charts documented via note block /REV 3.0.1/ within chart - sRXInitPLC - sRecipeActive - sRecipeAutoWrapAround - sSpPIDUpdateZx Sub - sSpMTLUpdateZx Sub - sTwoZoneMachine				
2	/REV 3.0.1/ continued - sMTLTimerIncrement - sPIDTimerIncrement - added chart UpdatePLCClock - sCycleAbortControl - deleted OPC 350 tags in Number Grid - deleted example state charts that were kept in Unused tab - deleted tags used in state examples	- Reset Action of FirstScan is taken care of by setting the Initial Value column of data type Flag to 1 for the respective state that will be reset on the FirstScan	COMMENT BLOCK	COMMENT BLOCK	
3	/REV 3.0.1/ continued - added flowcharts for Central System thermocouples - sMTLAlgCalcZx, sMTLAlgOFFZx - sPIDAlgCalcZx, sPIDAlgOFFZx - added use of xESTOP input, sCycleNormal, sCycleAbortControl, sCycleAbortRecipe - sRecipeNotLoaded - sValveManualControlZx - sMTLTuneldie		COMMENT BLOCK	COMMENT BLOCK	
4					
5					
6					

Flowchart : sTwoZoneMachine

Printed: Tuesday, January 06, 2009 @ 14:53

Project: MHC_TND_v3.0.1

Printed: Tuesday, January 06, 2009 @ 14:41
Project: MHC_TND_v3.0.1

Flowchart: sHMI_Heartbeat_Init

Printed: Tuesday, January 06, 2009 @ 14:41

Project: MHC_TND_v3.0.1

Flowchart: sHMI_Heartbeat_PulseUP

Printed: Tuesday, January 06, 2009 @ 14:55

Project: MHC_TND_v3.0.1

Flowchart: sHMI_Heartbeat_PulseDN

Printed: Tuesday, January 06, 2009 @ 14:55
 Project: MHC_TND_v3.0.1

Flowchart : sHMIAlive

Printed: Tuesday, January 06, 2009 @ 14:55
Project: MHC_TND_v3.0.1

Flowchart : sHMIDead

Printed: Tuesday, January 06, 2009 @ 14:56
Project: MHC_TND_v3.0.1

Flowchart: sCycleNormal

Printed: Thursday, February 26, 2009 @ 15:27

Project: MHC_TND_v3.0.1

FREEZE: NO CRITERIA

/REV 3.0.1/ added use of
xESTOP. ESTOP should be on
a normally closed contact,
therefore if NOT ON, then the
ESTOP has been pressed.

>Fluid Recipe means must be
either Mold Const Temper or
Mold Recipe

ATTACHMENT 2

2-142

If not 4 zone, then go back to top decision block; otherwise check z3 & z4, but check hfCycleAbort.

/REV 3.0.1/ Moved hfCycleAbort check to 2nd priority over the ESTOP and TC errors.

<-- these lines criss/cross, they are not connected or related

<-- these lines criss/cross, they are not connected or related

1
5

1
6

1
7

1
8

1
9

2
0

2
1

2

<-- these lines criss/cross, they are not connected or related

Flowchart: sCycleAbortControl

Printed: Thursday, February 26, 2009 @ 15:28

Project: MHC_TND_v3.0.1

Block 1: {N-5 hnControlModeZ1} = {N-10 nControlValve}
 Block 2: {N-6 hnControlModeZ2} = {N-10 nControlValve}
 Block 6: {N-7 hnControlModeZ3} = {N-10 nControlValve}
 Block 9: {N-8 hnControlModeZ4} = {N-10 nControlValve}

6

7

8

9

10

11

12

1

ATTACHMENT 2

/REV 3.0.1/ had to add a wait time in here because the PLC was turning OFF fCycleAbort faster than OPC could catch it.

This also allows the notification on the HMI to blink a little while. This also allows the state sets time to reset.

This does not halt execution of the program, only this chart.

24

1

19

21

22

25

27

29

20

23

26

28

30

ATTACHMENT 2

1
3

1
4

1
5

1
6

1
7

1
8

1
9

2

Flowchart : sAccumOFF

Printed: Tuesday, January 06, 2009 @ 14:58
Project: MHC_TND_v3.0.1

Flowchart : sAccumON

Printed: Tuesday, January 06, 2009 @ 14:59
Project: MHC_TND_v3.0.1

Flowchart: sAccumWait

Printed: Tuesday, January 06, 2009 @ 14:59
Project: MHC_TND_v3.0.1

Flowchart : sMTLTimerIdle

Printed: Tuesday, January 06, 2009 @ 14:59

Project: MHC_TND_v3.0.1

Flowchart : sMTLTimerIncrement

Printed: Tuesday, January 06, 2009 @ 15:00

Project: MHC_TND_v3.0.1

Block 0: $\{N-2 \text{ nMTLAccum}\} = \{N-2 \text{ nMTLAccum}\} + 1$

Flowchart : sMTLTimerDone

Printed: Tuesday, January 06, 2009 @ 15:00

Project: MHC_TND_v3.0.1

Flowchart : sPIDTimerIdle

Printed: Tuesday, January 06, 2009 @ 15:00
Project: MHC_TND_v3.0.1

Flowchart: sPIDTimerIncrement

Printed: Tuesday, January 06, 2009 @ 15:01
 Project: MHC_TND_v3.0.1

Block 0: $\{N-3 \text{ nPIDAccum}\} = \{N-3 \text{ nPIDAccum}\} + 1$

Flowchart : sPIDTimerDone

Printed: Tuesday, January 06, 2009 @ 15:02
 Project: MHC_TND_v3.0.1

Flowchart : sSegTimerIdleZ1

Printed: Tuesday, January 06, 2009 @ 15:02
Project: MHC_TND_v3.0.1

Flowchart : sSegTimerIdleZ3

Printed: Tuesday, January 06, 2009 @ 15:03
Project: MHC_TND_v3.0.1

Subchart : sSegTimerIdleZx

Printed: Tuesday, January 06, 2009 @ 15:03

Project: MHC_TND_v3.0.1

Flowchart : sSegTimerIncrementZ1

Printed: Tuesday, January 06, 2009 @ 15:04
Project: MHC_TND_v3.0.1

Subchart : sSegTimerIncrementZx

Printed: Tuesday, January 06, 2009 @ 15:04

Project: MHC_TND_v3.0.1

Block 0: {P-4 nSegElapsedTimeZx} = {P-4 nSegElapsedTimeZx} + 1

6
7
8
9
10
11
12
1

Flowchart: sSegTimerHoldZ1

Printed: Tuesday, January 06, 2009 @ 15:05

Project: MHC_TND_v3.0.1

Subchart : sSegTimerHoldZx

Printed: Tuesday, January 06, 2009 @ 15:05

Project: MHC_TND_v3.0.1

Block 3: {P-2 nSegElapsedTimeZx} = {P-2 nSegElapsedTimeZx} + 1

Flowchart : sSegTimerDoneZ1

Printed: Tuesday, January 06, 2009 @ 15:06
Project: MHC_TND_v3.0.1

Subchart : sSegTimerDoneZx

Printed: Tuesday, January 06, 2009 @ 15:06

Project: MHC_TND_v3.0.1

Flowchart: sCycleTimerIdle

Printed: Tuesday, January 06, 2009 @ 15:07

Project: MHC_TND_v3.0.1

Flowchart: sCycleTimerIncrement

Printed: Tuesday, January 06, 2009 @ 15:07

Project: MHC_TND_v3.0.1

Flowchart: sCycleTimerDone

Printed: Tuesday, January 06, 2009 @ 15:07
 Project: MHC_TND_v3.0.1

Block 1: {N-40 nCycleElapsedTime} = 0

Flowchart : sSkipPreheatIdle

Printed: Tuesday, January 06, 2009 @ 15:08
Project: MHC_TND_v3.0.1

Flowchart: sSkipPreheat

Printed: Tuesday, January 06, 2009 @ 15:08

Project: MHC_TND_v3.0.1

FREEZE: NO CRITERIA

Flowchart : sSkipPreheatDone

Printed: Tuesday, January 06, 2009 @ 15:08

Project: MHC_TND_v3.0.1

Flowchart : sMoldTemperLimitsBreachZ1

Printed: Tuesday, January 06, 2009 @ 15:09

Project: MHC_TND_v3.0.1

Flowchart : sMoldTemperLimitsBreachZ3

Printed: Tuesday, January 06, 2009 @ 15:09
 Project: MHC_TND_v3.0.1

Subchart : sMoldTemperLimitsBreachZx

Printed: Tuesday, January 06, 2009 @ 15:09

Project: MHC_TND_v3.0.1

ATTACHMENT 2

Flowchart : sMoldTemperLimitsOKZ1

Printed: Tuesday, January 06, 2009 @ 15:10
 Project: MHC_TND_v3.0.1

Subchart : sMoldTemperLimitsOKZx

Printed: Tuesday, January 06, 2009 @ 15:10

Project: MHC_TND_v3.0.1

6
7
8
9
10
11
12
1

Exit

```
Printed:    Tuesday, January 06, 2009 @ 15:11
Project:    MHC_TND_v3.0.1
```

Flowchart: sCycleINIT

Printed: Tuesday, January 06, 2009 @ 15:11
 Project: MHC_TND_v3.0.1

Flowchart : sPREHEAT

Printed: Tuesday, January 06, 2009 @ 15:11
Project: MHC_TND_v3.0.1

FREEZE: NO CRITERIA

Flowchart : sRFCS

Printed: Tuesday, January 06, 2009 @ 15:12
Project: MHC_TND_v3.0.1

Flowchart : sRUN

Printed: Tuesday, January 06, 2009 @ 15:12
 Project: MHC_TND_v3.0.1

FREEZE: NO CRITERIA

Flowchart: sIdleZ1

Printed: Tuesday, January 06, 2009 @ 15:12
Project: MHC_TND_v3.0.1

0
1
2
3
4
5
6

FREEZE: NO CRITERIA

Flowchart: sIdleZ3

Printed: Tuesday, January 06, 2009 @ 15:13
Project: MHC_TND_v3.0.1

FREEZE: NO CRITERIA

Subchart: sIdleZx

Printed: Tuesday, January 06, 2009 @ 15:13

Project: MHC_TND_v3.0.1

Flowchart: sPreheatZ1

Printed: Tuesday, January 06, 2009 @ 15:14
Project: MHC_TND_v3.0.1

Subchart : sPreheatZx

Printed: Tuesday, January 06, 2009 @ 15:14

Project: MHC_TND_v3.0.1

Flowchart : sRFCSZ1

Printed: Tuesday, January 06, 2009 @ 15:14

Project: MHC_TND_v3.0.1

Flowchart : sRunZ1

Printed: Tuesday, January 06, 2009 @ 15:15
Project: MHC_TND_v3.0.1

Flowchart : sDoneZ1

Printed: Tuesday, January 06, 2009 @ 15:15

Project: MHC_TND_v3.0.1

Flowchart : sRecipeNotLoaded

Printed: Thursday, February 26, 2009 @ 15:46

Project: MHC_TND_v3.0.1

Flowchart: sRecipeDownloadNew

Printed: Tuesday, January 06, 2009 @ 15:16
Project: MHC_TND_v3.0.1

FREEZE: NO CRITERIA

Flowchart : sRecipeCurrent

Printed: Tuesday, January 06, 2009 @ 15:16
 Project: MHC_TND_v3.0.1

Flowchart : sRecipeIdle

Printed: Tuesday, January 06, 2009 @ 15:17

Project: MHC_TND_v3.0.1

Flowchart : sRecipeActive

Printed: Tuesday, January 06, 2009 @ 15:17

Project: MHC_TND_v3.0.1

Flowchart : sRecipeDone

Printed: Tuesday, January 06, 2009 @ 15:17
Project: MHC_TND_v3.0.1

0

1

2

3

4

5

6

FREEZE: NO CRITERIA

Flowchart : sRecipeAutoWrapAround

Printed: Tuesday, January 06, 2009 @ 15:18

Project: MHC_TND_v3.0.1

Printed: Tuesday, January 06, 2009 @ 15:18
Project: MHC_TND_v3.0.1

Flowchart : sRXInitPLC

Printed: Friday, January 16, 2009 @ 15:56
Project: MHC_TND_v3.0.1

Block 6: {N-299 nRXNumOfSegments} = {N-4 hn2_4zoneMachineSwitch} * 20 - 1

TND_v3.0.1

Showing only the transfer of Segment information; the transfer of Constant information is essentially identical

Flowchart : sRXTransferSegData

Printed: Friday, January 23, 2009 @ 14:47

Project: MHC_TND_v3.0.1

have to put an additional check
for RecipeNotLoaded here so
that logic has escape plan

Flowchart : sRXTransferConData

Printed: Friday, January 23, 2009 @ 15:00

Project: MHC_TND_v3.0.1

enabled by a transition from
sRXTransferConDle

Moves value of HMI assigned
tags to PLC working tags.
There are no subchart variables
defined or mapped because the
subchart is dealing directly with
main program variables. It was
just easier to put all the moves in
4 like sub charts for each zone,
and keeps the higher level
sRXTransferConData chart
cleaner.

FREEZE: NO CRITERIA

ATTACHMENT 2

7

8

9

10

11

12

13

1

ATTACHMENT 2

1
5

1
6

1
7

1
8

1
9

2
0

2
1

2

Subchart : subRXConDataZ1

Printed: Friday, January 23, 2009 @ 15:04

Project: MHC_TND_v3.0.1

There are no subchart variables defined or mapped because the subchart is dealing directly with main program variables. It was just easier to put all the moves in 4 like sub charts for each zone, and keeps the higher level sRXTransferConData chart cleaner.

probably could do without nRampYN_Zx, but at least this way, I can verify that this number matches the HMI, to verify correct OPC xfer

Printed: Tuesday, January 06, 2009 @ 15:20
Project: MHC_TND_v3.0.1

```
Printed:    Tuesday, January 06, 2009 @ 15:20
Project:    MHC_TND_v3.0.1
```

Flowchart : sSpSegOFFZ3

Printed: Tuesday, January 06, 2009 @ 15:20

Project: MHC_TND_v3.0.1

Subchart : sSpSegOFFZx

Printed: Tuesday, January 06, 2009 @ 15:21

Project: MHC_TND_v3.0.1

Flowchart: sSpSegUpdateZ1

Printed: Tuesday, January 06, 2009 @ 15:21
Project: MHC_TND_v3.0.1

Flowchart : sSpSegLoadedZ1

Printed: Tuesday, January 06, 2009 @ 15:22
Project: MHC_TND_v3.0.1

Subchart : sSpSegLoadedZx

Printed: Friday, January 23, 2009 @ 15:05

Project: MHC_TND_v3.0.1

7
8
9
10
11
12
13
1

ATTACHMENT 2

Flowchart : sValveManualControlZ1

Printed: Tuesday, January 06, 2009 @ 15:23
Project: MHC_TND_v3.0.1

Flowchart : sValveManualControlZ3

Printed: Tuesday, January 06, 2009 @ 15:23
Project: MHC_TND_v3.0.1

Subchart: sValveManualControlZx

Printed: Thursday, February 26, 2009 @ 15:48

Project: MHC_TND_v3.0.1

7

On power up or abort, want to set CV=0 to flood lines with fluid to provide suction pressure to the loop pumps. Otherwise bumpless will be after some other change from PID or MTL and then want to set CV=50 so that the loop will truly see a bumpless transfer of temperature.

8

convert the nSP_PV from a percentage to number of bits for the output module to affect actuation of Control Valve

9

10

11

12

13

P-4
P-5

ATTACHMENT 2

2-223

Flowchart: sValveAutoControlZ1

Printed: Tuesday, January 06, 2009 @ 15:24
Project: MHC_TND_v3.0.1

Subchart : sValveAutoControlZx

Printed: Tuesday, January 06, 2009 @ 15:24

Project: MHC_TND_v3.0.1

Flowchart : sSpMTLOFFZ1

Printed: Tuesday, January 06, 2009 @ 15:25

Project: MHC_TND_v3.0.1

0

1

2

3

4

5

6

FREEZE: NO CRITERIA

Flowchart : sSpMTLOFFZ3

Printed: Tuesday, January 06, 2009 @ 15:25
Project: MHC_TND_v3.0.1

FREEZE: NO CRITERIA

Subchart : sSpMTLOFFZx

Printed: Tuesday, January 06, 2009 @ 15:26

Project: MHC_TND_v3.0.1

Flowchart : sSpMTLUpdateZ1

Printed: Tuesday, January 06, 2009 @ 16:02
Project: MHC_TND_v3.0.1

Subchart : sSpMTLUpdateZx

Printed: Thursday, February 26, 2009 @ 15:01

Project: MHC_TND_v3.0.1

Block 7: {L-0 c80elementArrayIndex} = {P-7 nZoneX} * 20 - 20 + {P-8 cSegNumberZx}

Block 14: {P-13 rSPRampRateZx} = ({P-10 nSPTemperTargetZx} - {L-1 nSPTemperPreviousZx}) / {P-9 nSPDwellTimeTargetZx}

Block 16: {P-14 rSPTemperRampZx} = {P-17 rSP_MTL_Zx} + {P-13 rSPRampRateZx} * 1

ATTACHMENT 2

P-17
P-18

Exit

Exit

RETURN BLOCK

1
3

1
4

1
5

1
6

1
7

1
8

1
9

2

Subchart : ClampTemperRamp

Printed: Tuesday, January 06, 2009 @ 16:03

Project: MHC_TND_v3.0.1

Subchart : ConvertTemper2%Range

Printed: Tuesday, January 06, 2009 @ 16:04

Project: MHC_TND_v3.0.1

0

1

2

3

4

5

0

Temp Range is 700-32
%Range is 100-0
Therefore, slope is 100/668.
Using y=mx + b to find the y
intercept with x=700, y=100,
yields the -4.79 offset

Exit

Exit

Block 0: {P-1 rTempAs%Range} = {P-0 rTemperature} * (100 / ({N-17 nTemperRangeUpperBound} - {N-16 nTemperRangeLowerBound})) - 4.79042

Flowchart : sMTLAlgOFFZ1

Printed: Tuesday, January 06, 2009 @ 16:04
Project: MHC_TND_v3.0.1

Flowchart : sMTLAlgOFFZ3

Printed: Tuesday, January 06, 2009 @ 16:05
Project: MHC_TND_v3.0.1

Subchart : sMTLAlgOFFZx

Printed: Thursday, February 26, 2009 @ 15:36

Project: MHC_TND_v3.0.1

Flowchart : sMTLAlgCalcZ1

Printed: Tuesday, January 06, 2009 @ 16:05
Project: MHC_TND_v3.0.1

Subchart : sMTLAlgCalcZx

Printed: Thursday, February 26, 2009 @ 15:34

Project: MHC_TND_v3.0.1

Block 2: {P-5 rSP_MTL_Zx} = {P-4 nTC_MoldZx} / 10

ATTACHMENT 2

Need to divide nTC_... by 10 to account for 1 implied decimal place. PLC module reads temperature, eg 758 = 75.8 degF. Put value in local subchart tag, so that it can be converted to %range next.

1
3

1
4

1
5

1
6

1
7

1
8

1
9

shift Output array to make room
for new 0th element value
(current value)

Set/change value of M.m-1 tag

Setup pointer to m-N-1 element
of MTLOuput Array.

If N=4, then m-N-1 = m-4-1 = m-5
th element, so the pointer is the
5th element (N+1).

N=Theta/T; therefore N will
always be 0 or greater.

Set/change value of M.m-N-1 tag

Setup pointer to m-N-2 element
of Array

Block 18: {L-0 cMTLOuputArrIndexPtr_m-N-1_Zx} = {P-22 nMTL_N_Zx} + 1
Block 20: {L-1 cMTLOuputArrIndexPtr_m-N-2_Zx} = {P-22 nMTL_N_Zx} + 2

20

21

22

23

24

25

26

2

Set/change value of M.m-N-2 tag
ATTACHMENT 2

shift Error array to make room for new 0th element value (current value) and to move previous values forward to populate the new 1st and 2nd previous errors on this iteration of the algorithm's calculation

calculate current error
this will automatically be placed in the 0th element of the MTL Error Array

populate 0th element of Output array

28
29
30
31
32
33
34

populate SP for Cascaded PID
this value is already in range
Send output to holding tag so
PID SP state set can pick it up, to
implement cascade control loop

Calculate ISE if in Run state

Block 28: (P-18 rMTL_ISE_Zx) = (P-18 rMTL_ISE_Zx) + (P-19 rMTLError_e.m_Zx) * (P-19 rMTLError_e.m_Zx)

Subchart : MTLouputAndError_Init

Printed: Tuesday, January 06, 2009 @ 16:06

Project: MHC_TND_v3.0.1

6

7

8

9

10

11

12

1

the array's length has 51 elements as specified on the Array datasheet, elements 0-50

zero MTL Error Array, because there are no previous errors yet

Subchart : ConvertTemper2%Range

Printed: Tuesday, January 06, 2009 @ 16:07

Project: MHC_TND_v3.0.1

0

1

2

3

4

5

0

Temp Range is 700-32
%Range is 100-0
Therefore, slope is 100/668.
Using y=mx + b to find the y
intercept with x=700, y=100,
yields the -4.79 offset

Exit

Exit

Block 0: {P-1 rTempAs%Range} = {P-0 rTemperature} * (100 / ({N-17 nTemperRangeUpperBound} - {N-16 nTemperRangeLowerBound})) - 4.79042

Subchart: MTL_Consts_Calc

Printed: Friday, January 23, 2009 @ 15:40

Project: MHC_TND_v3.0.1

Block 0: {P-0 N} = {P-1 Theta} \ {P-4 T}

Block 1: {L-3 f} = {P-0 N} + 1 - {P-1 Theta} / {P-4 T}

Block 2: {L-0 e^-T/tau} = exp(-{P-4 T} / {P-2 Tau})

Block 3: {L-1 e^-fT/L} = exp(-{L-3 f} * {P-4 T} / {P-3 Lambda})

6

7

8

9

10

11

12

$$\{L-2 e^{-T/L}\} = \exp(-\{P-4 T\} / \{P-3 \text{ Lambda}\})$$

$$\{P-6 Kcs\} = \{P-2 \text{ Tau}\} * (1 - \{L-1 e^{-fT/L}\}) / (\{P-5 Km\} * \exp(-\{L-3 f\} * \{P-4 T\} / \{P-2 \text{ Tau}\}))$$

$$\{P-7 P0\} = \{L-0 e^{-T/tau}\} - (\{L-1 e^{-fT/L}\} - \{L-2 e^{-T/L}\}) / (1 - \{L-1 e^{-fT/L}\})$$

$$\{P-8 P1\} = \{L-0 e^{-T/tau}\} * (\{L-1 e^{-fT/L}\} - \{L-2 e^{-T/L}\}) / (1 - \{L-1 e^{-fT/L}\})$$

$$\{P-9 P5\} = \{L-2 e^{-T/L}\}$$

Block 4: $\{L-2 e^{-T/L}\} = \exp(-\{P-4 T\} / \{P-3 \text{ Lambda}\})$
 Block 5: $\{P-6 Kcs\} = \{P-2 \text{ Tau}\} * (1 - \{L-1 e^{-fT/L}\}) / (\{P-5 Km\} * \exp(-\{L-3 f\} * \{P-4 T\} / \{P-2 \text{ Tau}\}))$
 Block 6: $\{P-7 P0\} = \{L-0 e^{-T/tau}\} - (\{L-1 e^{-fT/L}\} - \{L-2 e^{-T/L}\}) / (1 - \{L-1 e^{-fT/L}\})$
 Block 7: $\{P-8 P1\} = \{L-0 e^{-T/tau}\} * (\{L-1 e^{-fT/L}\} - \{L-2 e^{-T/L}\}) / (1 - \{L-1 e^{-fT/L}\})$
 Block 8: $\{P-9 P5\} = \{L-2 e^{-T/L}\}$

1
3

1
4

1
5

1
6

1
7

1
8

1
9

Block 9: $\{P-10\ P8\} = 1 - \{L-1\ e^{-fT/L}\}$
Block 10: $\{P-11\ P9\} = \{L-1\ e^{-fT/L}\} - \{L-2\ e^{-T/L}\}$

Subchart : ErrorAs%Range

Printed: Tuesday, January 06, 2009 @ 16:08
Project: MHC_TND_v3.0.1

0
1
2
3
4
5

PARAMS:
FP(r) rE rror
FP(r) rS P
N(r) nP V

LOCAL:
FP rPV

P-0
P-1
P-2

L-0

0

$$\{L-0 \text{ rPV}\} = \{P-2 \text{ nPV}\} / 10 * (100 / (\{N-17 \text{ nTemperRangeUpperBound}\} - \{N-16 \text{ nTemperRangeLowerBound}\})) - 4.79042$$

Need to divide nTC_xxx by 10 to account for 1 implied decimal place. PLC module reads temperature, eg 758 = 75.8 degF. Then convert to %Range.

Decided to keep decimal precision here so that error is more accurate. Decided not to truncate decimal and call sub "ConvertTemper2%Range" in this case, although it is similar.

1

$$\{P-0 \text{ rError}\} = \{P-1 \text{ rSP}\} - \{L-0 \text{ rPV}\}$$

%RangeError = SP - PV
since SP and PV are already in %Range

Exit

RETURN BLOCK

Block 0: $\{L-0 \text{ rPV}\} = \{P-2 \text{ nPV}\} / 10 * (100 / (\{N-17 \text{ nTemperRangeUpperBound}\} - \{N-16 \text{ nTemperRangeLowerBound}\})) - 4.79042$
Block 1: $\{P-0 \text{ rError}\} = \{P-1 \text{ rSP}\} - \{L-0 \text{ rPV}\}$

Subchart : MTL_Algorithm

Printed: Tuesday, January 06, 2009 @ 16:08

Project: MHC_TND_v3.0.1

Block 0: $\{P-0 M.m\} = \{P-1 K'\} * \{P-2 Kcs\} * (\{P-3 e.m\} - \{P-9 P0\} * \{P-4 e.m-1\} - \{P-10 P1\} * \{P-5 e.m-2\}) + \{P-11 P5\} * \{P-6 M.m-1\} + \{P-12 P8\} * \{P-7 M.m-N-1\} + \{P-13 P9\} * \{P-8 M.m-N-2\}$

Flowchart : sSpPIDOFFZ1

Printed: Tuesday, January 06, 2009 @ 16:09
Project: MHC_TND_v3.0.1

Flowchart : sSpPIDOFFZ3

Printed: Tuesday, January 06, 2009 @ 16:09
Project: MHC_TND_v3.0.1

0

1

2

3

4

5

6

FREEZE: NO CRITERIA

Subchart : sSpPIDOFFZx

Printed: Tuesday, January 06, 2009 @ 16:09

Project: MHC_TND_v3.0.1

Subchart : sSpPIDUpdateZx

Printed: Thursday, February 26, 2009 @ 14:58

Project: MHC_TND_v3.0.1

0

1

2

3

4

5

6

Block 16: $\{L-0\ c80elementArrayIndex\} = \{P-10\ nZoneX\} * 20 - 20 + \{P-11\ cSegNumberZx\}$

Block 18: $\{P-16\ rSPRampRateZx\} = \frac{\{P-13\ nSPTemperTargetZx\} - \{L-1\ nSPTemperPreviousZx\}}{\{P-12\ nSPDwellTimeTargetZx\}}$

2
2

2
3

2
4

2
5

2
6

2
7

2
8

RampRate calculation is:
 $SP_{new} = SP_{old} + RR * TimeBase$
RR is calculated at HMI and has units of [degF/sec]. Since this calculation is triggered when sAccumON is on, and sAccumON is triggered by SecondClock, that accounts for the TimeBase of 1.

CONDITION 6
/REV 3.0.1/ added condition when in-between PID and MTL before MTL constns are fully defined, still need to control to PID SP.
Previous condition was (CtlMode=3 & MTLTunIdle)

Block 23: $\{P-17 \text{ rSPTemperRampZx}\} = \{P-24 \text{ rSP_PID_Zx}\} + \{P-16 \text{ rSPRampRateZx}\} * 1$

29

30

31

32

33

34

35

3

Flowchart: sPIDAlgOFFZ1

Printed: Tuesday, January 06, 2009 @ 16:11

Project: MHC_TND_v3.0.1

0

1

2

3

4

5

6

FREEZE: NO CRITERIA

Flowchart: sPIDAlgOFFZ3

Printed: Tuesday, January 06, 2009 @ 16:11
Project: MHC_TND_v3.0.1

FREEZE: NO CRITERIA

Subchart : sPIDAlgOFFZx

Printed: Thursday, February 26, 2009 @ 15:44

Project: MHC_TND_v3.0.1

Flowchart: sPIDAlgCalcZ1

Printed: Tuesday, January 06, 2009 @ 16:12
 Project: MHC_TND_v3.0.1

Subchart : sPIDAlgCalcZx

Printed: Thursday, February 26, 2009 @ 15:41

Project: MHC_TND_v3.0.1

6
7
8
9
10
11
12

/REV 3.0.1/ First check, part 2 of 2.
Yes result goes directly to PID Alg Calc for first time through when the Recipe has changed the control mode. Shouldn't need to check that CtlMode=ControlValve, because the recipe must have PID or MTL configured for this zone.
----->

ATTACHMENT 2

Y 12 F-51 NOT ON

populate HMI tag ----->

P-5
P-4

P-5
P-6

Exit

9

P-7

=

N-10

N 11

Transition condition checks:
1st check is bumpless loopback
2nd check is transition to other state
3rd check is PID Alg loopback
this is 2nd check

P-0

P-1

Exit

11

Exit

N

Transition condition checks:
1st check is bumpless loopback
2nd check is transition to other state
3rd check is PID Alg loopback
this is 3rd check

Y 12

F-18 ON

Exit

P-17

P-18

13

P-17

P-6

P-3

14

set new previous error

calculate error

1
3

1
4

1
5

1
6

1
7

1
8

1
9

calculate integrator summation, I portion of PID

Decided not to do this:

If in a ramping segment, turn off the ID portion of control. Dick may or may not have done this, so may also want to look at following scenarios during 1st startup:
- If ramping with Fluid control, turn off ID.
- If ramping with Metal control, turn off D.
- Keep I on, but if reaches 100, then reset to 0.

However, may want to zero I&D when the ramp is done??

clamp I portion of PID

calculate P portion of PID

Block 14: {P-14 rPIDDerivPortionZx} = 60 * {P-9 hrPID_Kp_Zx} * {P-11 hrPID_Kd_Zx} * ({P-17 rPIDError_e.n_Zx} - {P-18 rPIDError_e.n-1_Zx}) / {N-1 nPIDSampleTime_T1}
Block 15: {P-13 rPIDIntegPortionZx} = {P-13 rPIDIntegPortionZx} + {P-9 hrPID_Kp_Zx} * {P-10 hrPID_Ki_Zx} * {N-1 nPIDSampleTime_T1} * {P-17 rPIDError_e.n_Zx} / 60
Block 23: {P-12 rPIDPropPortionZx} = {P-9 hrPID_Kp_Zx} * {P-17 rPIDError_e.n_Zx}

20

21

22

23

24

25

26

Block 24: {P-19 rPIDOutput_F.n_Zx} = {P-12 rPIDPropPortionZx} + {P-13 rPIDIntegPortionZx} + {P-14 rPIDDerivPortionZx}

Block 29: {P-21 nCV_Output_Zx} = {P-19 rPIDOutput_F.n_Zx} * 4095 / 100

Block 31: {P-20 rPID_ISE_Zx} = {P-20 rPID_ISE_Zx} + {P-17 rPIDError_e.n_Zx} * {P-17 rPIDError_e.n_Zx}

Subchart: PIDError_Zero

Printed: Tuesday, January 06, 2009 @ 16:13

Project: MHC_TND_v3.0.1

Flowchart : sTC_OK_FluidZ1

Printed: Tuesday, January 06, 2009 @ 16:14
Project: MHC_TND_v3.0.1

Flowchart: sTC_OK_FluidZ3

Printed: Tuesday, January 06, 2009 @ 16:14

Project: MHC_TND_v3.0.1

Subchart: sTC_OK_...Zx

Printed: Tuesday, January 06, 2009 @ 16:15

Project: MHC_TND_v3.0.1

Flowchart: sTC_Open_FluidZ1

Printed: Tuesday, January 06, 2009 @ 16:15

Project: MHC_TND_v3.0.1

0
1
2
3
4
5
6

FREEZE: NO CRITERIA

Subchart: sTC_Open_...Zx

Printed: Tuesday, January 06, 2009 @ 16:15

Project: MHC_TND_v3.0.1

Flowchart: sTC_Short_FluidZ1

Printed: Tuesday, January 06, 2009 @ 16:16

Project: MHC_TND_v3.0.1

Subchart: sTC_Short_...Zx

Printed: Tuesday, January 06, 2009 @ 16:16

Project: MHC_TND_v3.0.1

Flowchart : sTC_OK_MoldZ1

Printed: Tuesday, January 06, 2009 @ 16:17
Project: MHC_TND_v3.0.1

Flowchart: sTC_OK_MoldZ3

Printed: Tuesday, January 06, 2009 @ 16:18
Project: MHC_TND_v3.0.1

Flowchart : sTC_Open_MoldZ1

Printed: Tuesday, January 06, 2009 @ 16:19
Project: MHC_TND_v3.0.1

0

1

2

3

4

5

6

FREEZE: NO CRITERIA

Flowchart: sTC_Short_MoldZ1

Printed: Tuesday, January 06, 2009 @ 16:19

Project: MHC_TND_v3.0.1

Flowchart : sMTLTuneIdle

Printed: Thursday, February 26, 2009 @ 15:50
Project: MHC_TND_v3.0.1

Flowchart : sMTLTuneInit

Printed: Friday, January 23, 2009 @ 15:53

Project: MHC_TND_v3.0.1

7

8

9

10

11

12

13

1

Subchart: MTLTune_Consts_Zero

Printed: Tuesday, January 06, 2009 @ 16:21

Project: MHC_TND_v3.0.1

Block 0: {P-0 nMTStblzTimeZx} = 0

Block 1: {P-4 Lambda} = 0

Block 2: {P-8 P0} = 0

Block 3: {P-10 P5} = 0

Block 4: {P-1 Theta} = 0

Block 5: {P-5 Km} = 0

Block 6: {P-9 P1} = 0

Block 7: {P-11 P8} = 0

Block 8: {P-2 Tau} = 0

Block 9: {P-6 Kprime} = 0

Block 10: {P-12 P9} = 0

5

6

7

8

9

10

11

ATTACHMENT 2

Block 11: $\{P-3 \ 5\text{Tau}\} = 0$
Block 12: $\{P-7 \ Kcs\} = 0$

Flowchart : sMTLTuneStabilizeDone

Printed: Tuesday, January 06, 2009 @ 16:24

Project: MHC_TND_v3.0.1

Flowchart : sMTLTuneDone

Printed: Tuesday, January 06, 2009 @ 16:26

Project: MHC_TND_v3.0.1

Subchart : MTLTune_FindLambdaMax

Printed: Friday, January 23, 2009 @ 16:00

Project: MHC_TND_v3.0.1

Flowchart : sMTLTuneAbort

Printed: Tuesday, January 06, 2009 @ 16:27
 Project: MHC_TND_v3.0.1

7
8
9
10
11
12
13
1

9

Call
MTLTune_Consts_Zero(
 nMTStblzTimeZ1,
-165
 nMTThetaTimeZ1
-166
 nMTTauTimeZ1,
-167
 nMT5TauTimeZ1,
-169
 nMTL_LambdaTune
 _Z1,
-172
 rMTL_KmTune_Z1,
-56
 rMTL_KprimeTune
 _Z1,
-55
 rMTL_KcsTune_Z1
 ,
-57
 rMTL_P0Tune_Z1,
-58
 rMTL_P1Tune_Z1,
-59
 rMTL_P5Tune_Z1,
-60
 rMTL_P8Tune_Z1,
-61
 rMTL_P9Tune_Z1)
-62

N
N
N
N
N
N
FP
FP
FP
FP
FP
FP
FP
FP
FP
FP

zero out MTL parameters

10

Call
MTLTune_Consts_Zero(
 nMTStblzTimeZ2,
-185
 nMTThetaTimeZ2
-186
 nMTTauTimeZ2,
-187
 nMT5TauTimeZ2,
-189
 nMTL_LambdaTune
 _Z2,
-192
 rMTL_KmTune_Z2,
-123
 rMTL_KprimeTune
 _Z2,
-122
 rMTL_KcsTune_Z2
 ,
-124
 rMTL_P0Tune_Z2,
-125
 rMTL_P1Tune_Z2,
-126
 rMTL_P5Tune_Z2,
-127
 rMTL_P8Tune_Z2,
-128
 rMTL_P9Tune_Z2)
-129

N
N
N
N
N
N
N
FP
FP
FP
FP
FP
FP
FP
FP
FP
FP

11

Call
MTLTune_Consts_Zero(
 nMTStblzTimeZ3,
-205
 nMTThetaTimeZ3
-206
 nMTTauTimeZ3,
-207
 nMT5TauTimeZ3,
-209
 nMTL_LambdaTune
 _Z3,
-212
 rMTL_KmTune_Z3,
-190
 rMTL_KprimeTune
 _Z3,
-189
 rMTL_KcsTune_Z3
 ,
-191
 rMTL_P0Tune_Z3,
-192
 rMTL_P1Tune_Z3,
-193
 rMTL_P5Tune_Z3,
-194
 rMTL_P8Tune_Z3,
-195
 rMTL_P9Tune_Z3)
-196

N
N
N
N
N
N
N
FP
FP
FP
FP
FP
FP
FP
FP
FP
FP

12

Call
MTLTune_Consts_Zero(
 nMTStblzTimeZ4,
-225
 nMTThetaTimeZ4
-226
 nMTTauTimeZ4,
-227
 nMT5TauTimeZ4,
-229
 nMTL_LambdaTune
 _Z4,
-232
 rMTL_KmTune_Z4,
-257
 rMTL_KprimeTune
 _Z4,
-256
 rMTL_KcsTune_Z4
 ,
-258
 rMTL_P0Tune_Z4,
-259
 rMTL_P1Tune_Z4,
-260
 rMTL_P5Tune_Z4,
-261
 rMTL_P8Tune_Z4,
-262
 rMTL_P9Tune_Z4)
-263

N
N
N
N
N
N
N
FP
FP
FP
FP
FP
FP
FP
FP
FP
FP

had to add a wait time in here because the PLC was turning OFF hf__Abort faster than OPC could catch it. This also allows the notification on the HMI to blink a little while. This also allows the state sets time to reset. This does not halt execution of the program, only this chart.

transition action:
reset HMI Start and Abort flags

state transition

Decided to take out criteria that Mold TCs must be OK before transitioning back to MTLTuneIdle. Because if Mold TCs are not OK and a person tries to go thru the tuning routine, it will dump them back to Abort anyway. But at least the system will allow an escape out of MTLTuneAbort, as long as at least the Fluid TCs are OK.

Flowchart : sMTStblzIdleZ1

Printed: Tuesday, January 06, 2009 @ 16:28

Project: MHC_TND_v3.0.1

Flowchart : sMTStblzInitZ1

Printed: Tuesday, January 06, 2009 @ 16:29
 Project: MHC_TND_v3.0.1

Subchart : sMTStblzInitZx

Printed: Tuesday, January 06, 2009 @ 16:29

Project: MHC_TND_v3.0.1

Initialize values.

Block 0: $\{P-8 \text{ nMTTauTemperZx}\} = (0.632 * (\{P-5 \text{ hnMoldEndTemperZx}\} - \{P-4 \text{ hnMoldStartTemperZx}\})) + \{P-4 \text{ hnMoldStartTemperZx}\}$ Block 2: $\{P-7 \text{ nMoldStartTemperLoLimitZx}\} = 0.97 * \{P-4 \text{ hnMoldStartTemperZx}\}$ Block 5: $\{P-6 \text{ nMoldStartTemperHiLimitZx}\} = 1.03 * \{P-4 \text{ hnMoldStartTemperZx}\}$

Flowchart : sMTStblzTestZ1

Printed: Tuesday, January 06, 2009 @ 16:30
Project: MHC_TND_v3.0.1

Subchart : sMTStblzTestZx

Printed: Tuesday, January 06, 2009 @ 16:30

Project: MHC_TND_v3.0.1

6

7

8

9

10

11

12

ATTACHMENT 2

Block 6: {P-2 nMTStblzTimeZx} = {P-2 nMTStblzTimeZx} + 1
Block 7: {P-2 nMTStblzTimeZx} = 0

Flowchart : sMTStblzDoneZ1

Printed: Tuesday, January 06, 2009 @ 16:31
Project: MHC_TND_v3.0.1

Printed: Tuesday, January 06, 2009 @ 16:31
Project: MHC_TND_v3.0.1

6

7

8

9

10

11

12

ATTACHMENT 2

Flowchart : sMTThetaIdleZ1

Printed: Tuesday, January 06, 2009 @ 16:32
Project: MHC_TND_v3.0.1

0

1

2

3

4

5

6

enabled by 1) reset action:
setting the Initial Value to 1, this
will effectively enable the chart
on the system "FirstScan", or 2) a
transition from sMTThetaDoneZ1

FREEZE: NO CRITERIA

Subchart : sMTThetaIdleZx

Printed: Tuesday, January 06, 2009 @ 16:32

Project: MHC_TND_v3.0.1

Flowchart : sMTThetaTestZ1

Printed: Tuesday, January 06, 2009 @ 16:33
Project: MHC_TND_v3.0.1

Subchart : sMTThetaTestZx

Printed: Tuesday, January 06, 2009 @ 16:33

Project: MHC_TND_v3.0.1

Block 0: {L-0 rTC/10} = {P-3 nTC_MoldZx} / 10

6
7
8
9
10
11
12

If TC >= Hi and HeatUp then
done testing Theta, otherwise
increment timer.

OR

If TC <= Lo and CoolDown then
done testing Theta, otherwise
increment timer.

ATTACHMENT 2

Flowchart : sMTThetaDoneZ1

Printed: Tuesday, January 06, 2009 @ 16:34

Project: MHC_TND_v3.0.1

Subchart : sMTThetaDoneZx

Printed: Tuesday, January 06, 2009 @ 16:34

Project: MHC_TND_v3.0.1

Flowchart : sMTTaudleZ1

Printed: Tuesday, January 06, 2009 @ 16:34
Project: MHC_TND_v3.0.1

Subchart : sMTTaudleZx

Printed: Tuesday, January 06, 2009 @ 16:35

Project: MHC_TND_v3.0.1

Flowchart : sMTTtauTestZ1

Printed: Tuesday, January 06, 2009 @ 16:35
Project: MHC_TND_v3.0.1

Subchart : sMTTauTestZx

Printed: Tuesday, January 06, 2009 @ 16:35

Project: MHC_TND_v3.0.1

Block 0: {L-0 rTC/10} = {P-2 nTC_MoldZx} / 10

Block 7: {P-7 nMT5TauTimeTargetZx} = {P-6 nMTTauTimeZx} * 5

Block 9: {P-6 nMTTauTimeZx} = {P-6 nMTTauTimeZx} + 1

Flowchart : sMTTauDoneZ1

Printed: Tuesday, January 06, 2009 @ 16:36
Project: MHC_TND_v3.0.1

Subchart : sMTTauDoneZx

Printed: Tuesday, January 06, 2009 @ 16:36

Project: MHC_TND_v3.0.1

Flowchart : sMT5TauIdleZ1

Printed: Tuesday, January 06, 2009 @ 16:36

Project: MHC_TND_v3.0.1

Subchart : sMT5TauIdleZx

Printed: Tuesday, January 06, 2009 @ 16:37

Project: MHC_TND_v3.0.1

Flowchart : sMT5TauTestZ1

Printed: Tuesday, January 06, 2009 @ 16:37
Project: MHC_TND_v3.0.1

Block 0: {N-169 nMT5TauTimeZ1} = {N-167 nMTTauTimeZ1}

Subchart : sMT5TauTestZx

Printed: Tuesday, January 06, 2009 @ 16:38

Project: MHC_TND_v3.0.1

Block 4: {P-2 nMT5TauTimeZx} = {P-2 nMT5TauTimeZx} + 1

Block 5: {P-6 nMTL_KmTune_Zx} = {P-4 nTC_MoldZx} / {P-5 nTC_FluidZx}

Flowchart : sMT5TauDoneZ1

Printed: Tuesday, January 06, 2009 @ 16:38

Project: MHC_TND_v3.0.1

6

7

8

9

10

11

12

1

State transition criteria.

ATTACHMENT 2

Subchart : sMT5TauDoneZx

Printed: Tuesday, January 06, 2009 @ 16:39

Project: MHC_TND_v3.0.1

Printed: Tuesday, January 06, 2009 @ 16:40
Project: MHC_TND_v3.0.1

Flowchart: z-end

Printed: Friday, January 16, 2009 @ 15:58

Project: MHC_TND_v3.0.1

DataView - MHC_TND_v3.0.1

File Edit View Tools Live! Options Window Help

ATTACHMENT 2

Array

Create Type: Array

	Type	Starting Tagname	Length	Description
0	Number	nDwellTimeSP0000	80	working recipe data; N-500 (nDwellTimeSP0000) thru N-579 (nDwellTimeSP0079)
1	Number	nTemperSP0000	80	working recipe data; N-581 (nTemperSP0000) thru N-660 (nTemperSP0079)
2	Flag	fRampYN0000	80	working recipe data; F-500 (fRampYN0000) thru F-579 (fRampYN0079)
3				
4				
5	Number	nZero0000	80	array of zeros; N-662 (nZero0000) thru N-741 (nZero0079)
6	Flag	fZero0000	80	array of zeros; F-581 (fZero0000) thru F-660 (fZero0079)
7	Float	rZero0000	80	array of zeros; FP-581 (rZero0000) thru FP-660 (rZero0079)
8				
9				
10				
11	Float	rMTLOutputArrayZ1_0000	51	past values of MTL Alg Output for Zonel for use in calculating new Ouput M.m
12	Float	rMTLOutputArrayZ2_0000	51	TND Array starts at 0 and goes to 50; therefore 51 elements, length of 51
13	Float	rMTLOutputArrayZ3_0000	51	element 0 is reserved for the current Output, element 1 is the 1st previous
14	Float	rMTLOutputArrayZ4_0000	51	Z1: FP-700 to FP-750; Z2: FP-800 to FP-850; ... ; Z4: FP-1000 to FP-1050
15	Float	rMTLError_e.m_Z1	3	past values of MTL Error for use in MTL Alg
16	Float	rMTLError_e.m_Z2	3	TND array index values are 0,1,2; for a length of 3 variables
17	Float	rMTLError_e.m_Z3	3	element 0 is current, element 1 is 1st previous, element 2 is 2nd previous
18	Float	rMTLError_e.m_Z4	3	
19				
20				

For Help, press F1

Array (new)

NUM

Counter

Create Retentive Type: Counter

	*	ID	Tagname	Initial Value	Physical I/O	Description
0						
1						
2	*	C-2	cRXSegNumber	0		used in transferring temporary register recipe data from HMI to PLC as the index of 80 element array
3						
4	*	C-4	cSegNumberZ1	0		current segment number of Zone1, value 0-19
5	*	C-5	cSegNumberZ2	0		
6	*	C-6	cSegNumberZ3	0		
7	*	C-7	cSegNumberZ4	0		
8						
9	*	C-9	cRXZoneNumber	0		used in transferring control constants and parameters recipe data from HMI to PLC as the index of the zone class
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20	*	C-20	-- end COUNTER definitions --	0		end of TND Counter datasheet
21						
22						

For Help, press F1

Counter (new) NUM

Flag

Create

Retentive

Type: Flag

	*	ID	Tagname	Initial Value	Physical I/O	Description
0	*	F-0	Flag0	1		these are the states and state flags as designed by the MHC state machine set; prefix s=state flag, f=PLC discrete flag, hf=HMI flag
1						
2	*	F-2	sTwoZoneMachine	1		Compression or Injection Molding
3	*	F-3	sFourZoneMachine	0		typically Injection Molding
4						
5						
6	*	F-6	AlwaysOFF	0		used to disable a flowchart, or stop logic
7						
8						
9	*	F-9	sAccumOFF	1		uses system tag "SecClock" to accumulate timers for MTL and PID algorithms, for Segment elapsed times, and for Cycle elapsed times
10	*	F-10	sAccumON	0		on for lScan, used in other state sets to increment the accumulators
11	*	F-11	sAccumWait	0		looks for the transition of SecClock from 1 back to 0
12						
13	*	F-13	sMTLTimerIdle	1		waiting for sAccumON to increment Timer
14	*	F-14	sMTLTimerIncrement	0		increments the MTL Algorithm sample clock period (SP=10 seconds)
15	*	F-15	sMTLTimerDone	0		trigger for MTL alg to calculate
16	*	F-16	sPIDTimerIdle	1		
17	*	F-17	sPIDTimerIncrement	0		
18	*	F-18	sPIDTimerDone	0		trigger for PID alg to calculate
19						
20	*	F-20	sSegTimerIdleZ1	1		
21	*	F-21	sSegTimerIncrementZ1	0		increments the segment elapsed time
22	*	F-22	sSegTimerDoneZ1	0		trigger for end of Recipe Profile Segment for Zonel
23	*	F-23	sSegTimerIdleZ2	1		
24	*	F-24	sSegTimerIncrementZ2	0		
25	*	F-25	sSegTimerDoneZ2	0		
26	*	F-26	sSegTimerIdleZ3	1		
27	*	F-27	sSegTimerIncrementZ3	0		
28	*	F-28	sSegTimerDoneZ3	0		
29	*	F-29	sSegTimerIdleZ4	1		
30	*	F-30	sSegTimerIncrementZ4	0		
31	*	F-31	sSegTimerDoneZ4	0		
32	*	F-32	sSegTimerHoldZ1	0		state to prohibit seg# increment until all zones are ready for Seg2(RFCS)
33	*	F-33	sSegTimerHoldZ2	0		
34	*	F-34	sSegTimerHoldZ3	0		
35	*	F-35	sSegTimerHoldZ4	0		
36	*	F-36	sCycleTimerIdle	1		
37	*	F-37	sCycleTimerIncrement	0		
38	*	F-38	sCycleTimerDone	0		end of the plastic part's recipe cycle
39						
40	*	F-40	sSkipPreheatIdle	1		
41	*	F-41	sSkipPreheat	0		trigger for one-shot to skip Seg0&Seg1 on first part of production run, or CycleAbort
42	*	F-42	sSkipPreheatDone	0		
43	*	F-43	hfSkipPreheat	0		HMI tag: value of Skip Preheat as set by HMI, =1 to skip Preheat segments, will be a one-shot if running AutoPreheat for multiple parts

For Help, press F1

Flag (edit)

NUM

Flag

Create

Retentive

Type: Flag

	*	ID	Tagname	Initial Value	Physical I/O	Description
45	*	F-45	hfManualPreheat	0		HMI tag: 1=ManualPreheat=old INIT PRHT; 0=AutoPreheat=old Auto Wrap Around
46						
47						
48						
49						
50						
51	*	F-51	sRecipeNotLoaded	1		initial state when a recipe is not loaded, transition out when recipe is loaded on HMI
52	*	F-52	sRecipeCurrent	0		recipe on PLC is current and correct
53	*	F-53	sRecipeDownloadNew	0		need to download a new recipe to PLC
54						
55						
56	*	F-56	hfStartRecipe	0		HMI tag: Start/Load Recipe button was pushed
57						
58	*	F-58	hfCancelDownload	0		HMI tag: user canceled the download during the download wait screen
59	*	F-59	sRecipeXferOFF	1		power up state for transferring recipe data from HMI to PLC
60	*	F-60	sRXInitPLC	0		initialize PLC variables, zero arrays, zero counter, etc
61	*	F-61	F-61	0		deleted
62	*	F-62	sRXTransferSegData	0		transfer of temporary SEGMENT registers and assign to 80 element array
63	*	F-63	sRXPLCLoaded	0		signals that a recipe is successfully loaded on the PLC
64	*	F-64	hfRXHandshake	0		HMI tag: trigger working with PLC to xfer temporary registers
65	*	F-65	F-65	0		deleted
66	*	F-66	fRXSegDataStored	0		handshake that PLC has all recipe segment data
67	*	F-67	F-67	0		deleted
68	*	F-68	F-68	0		deleted
69	*	F-69	sRXTransferConData	0		transfer of CONTROL CONSTANTS and PARAMETERS
70	*	F-70	fRXConDataStored	0		handshake that PLC has all control constants and parameters data
71						
72						
73						
74	*	F-74	sCycleAbortControl	0		cycle has been aborted due to thermocouple failure
75	*	F-75	sCycleNormal	1		power up state
76	*	F-76	sCycleAbortRecipe	0		cycle has been aborted by button on HMI
77	*	F-77	fCycleAbort	0		PLC tag that signifies that PLC caused a CycleAbort
78	*	F-78	hfCycleAbort	0		HMI tag: button pressed on HMI to abort cycle
79	*	F-79	hfDryCycle	0		HMI tag: button pressed on HMI to bypass MHCRelay and start a cycle
80						
81	*	F-81	hstFileCreateTrigger	0		HMI tag: script trigger to generate history file name
82	*	F-82	sRecipeIdle	1		power up state, no recipe is running
83	*	F-83	sRecipeActive	0		a recipe is running
84	*	F-84	sRecipeDone	0		yup, you got it
85	*	F-85	sRecipeAutoWrapAround	0		state to capture a pause before allowing wraparound to occur
86	*	F-86	sIdleZ1	1		Zone is idle either because 1) Recipe Not Active, or 2) Recipe Active and completed last segment
87	*	F-87	sPreheatZ1	0		zone is currently in segment 0 or segment 1
88	*	F-88	sRFCSZ1	0		zone has completed segment 1 and is ready to start segment 2

For Help, press F1

Flag (edit)

NUM

Flag

Create

Retentive

Type: Flag

	*	ID	Tagname	Initial Value	Physical I/O	Description
89	*	F-89	sRunZ1	0		zone is running the rest of the recipe, segment 2 through segment last
90	*	F-90	sIdleZ2	1		
91	*	F-91	sPreheatZ2	0		
92	*	F-92	sRFCSZ2	0		
93	*	F-93	sRunZ2	0		
94	*	F-94	sIdleZ3	1		
95	*	F-95	sPreheatZ3	0		
96	*	F-96	sRFCSZ3	0		
97	*	F-97	sRunZ3	0		
98	*	F-98	sIdleZ4	1		
99	*	F-99	sPreheatZ4	0		
100	*	F-100	sRFCSZ4	0		
101	*	F-101	sRunZ4	0		
102						
103						
104	*	F-104	sIDLE	1		all zones idle
105	*	F-105	sCycleINIT	0		initialization of recipe sequence, waiting to transition to sPREHEAT sRFCs
106	*	F-106	sPREHEAT	0		at least one zone still in preheat
107	*	F-107	sRFCs	0		all zones done with preheat and ready for Seg2
108	*	F-108	sRUN	0		running Seg2+
109						
110						
111	*	F-111	sHMI_Heartbeat_Init	1		starting state, starts HB timer
112	*	F-112	sHMI_Heartbeat_PulseUP	0		pulse up, HMI tag=1
113	*	F-113	sHMI_Heartbeat_PulseDN	0		pulse down, HMI tag=0
114	*	F-114	hfHMIHeartbeat	0		HMI tag
115	*	F-115	sHMIAlive	1		HMI is communicating with PLC
116	*	F-116	sHMIDead	0		HMI is a goner
117						
118	*	F-118	sDoneZ1	0		belongs with sIdleZx, sPrntZx, sRFCsZx, sRunZx states, done with segments
119	*	F-119	sDoneZ2	0		
120	*	F-120	sDoneZ3	0		
121	*	F-121	sDoneZ4	0		
122						
123						
124						
125						
126						
127	*	F-127	sMoldTemperLimitsBreachZ1	1		the temperature of the mold is outside of the limits set; temp must be inside limits to proceed to RFCs
128	*	F-128	sMoldTemperLimitsOKZ1	0		temp is either 1)inside limits, or 2)option is turned off
129	*	F-129	sMoldTemperLimitsBreachZ2	1		
130	*	F-130	sMoldTemperLimitsOKZ2	0		
131	*	F-131	sMoldTemperLimitsBreachZ3	1		
132	*	F-132	sMoldTemperLimitsOKZ3	0		

For Help, press F1

Flag (edit)

NUM

Flag

Create

Retentive

Type: Flag

	*	ID	Tagname	Initial Value	Physical I/O	Description
133	*	F-133	sMoldTemperLimitsBreachZ4	1		
134	*	F-134	sMoldTemperLimitsOKZ4	0		
135						
136	*	F-136	hfMoldTemperLimitsZ1	1		HMI tag: button to turn ON/OFF Mold temper limit. 1=want mold to be within limits before proceeding to RFCS, 0=off so always LimitsOK
137	*	F-137	hfMoldTemperLimitsZ2	1		set all zones to 1 to prevent transition to sMoldTemperLimitsOKZx on FirstScan
138	*	F-138	hfMoldTemperLimitsZ3	1		
139	*	F-139	hfMoldTemperLimitsZ4	1		
140	*	F-140	hfRXData_MoldTemperLimits	0		temporary register for recipe xfer of control data, if Mold Temper Limits used, then =1
141						
142						
143						
144						
145						
146	*	F-146	fBumplessMTLZ1	0		trigger when entering control state to effect bumpless transition to MTL
147	*	F-147	fBumplessPIDZ1	0		trigger when entering control state to effect bumpless transition to PID
148	*	F-148	fBumplessCVZ1	0		trigger when entering control state to effect bumpless transition to ControlValve
149	*	F-149	fBumplessMTLZ2	0		
150	*	F-150	fBumplessPIDZ2	0		
151	*	F-151	fBumplessCVZ2	0		
152	*	F-152	fBumplessMTLZ3	0		
153	*	F-153	fBumplessPIDZ3	0		
154	*	F-154	fBumplessCVZ3	0		
155	*	F-155	fBumplessMTLZ4	0		
156	*	F-156	fBumplessPIDZ4	0		
157	*	F-157	fBumplessCVZ4	0		
158						
159						
160	*	F-160	sSpMTL0FFZ1	1		state set is OFF, meaning that control is Manual CV or PID Control
161	*	F-161	sSpMTLUpdateZ1	0		updating the MTL setpoint, in MTL control
162	*	F-162	sSpPID0FFZ1	1		state set is OFF, meaning that control is Manual via Control Valve
163	*	F-163	sSpPIDUpdateZ1	0		updating the PID setpoint, either in PID control or cascade MTL-PID control
164	*	F-164	sSpMTL0FFZ2	1		
165	*	F-165	sSpMTLUpdateZ2	0		
166	*	F-166	sSpPID0FFZ2	1		
167	*	F-167	sSpPIDUpdateZ2	0		
168	*	F-168	sSpMTL0FFZ3	1		
169	*	F-169	sSpMTLUpdateZ3	0		
170	*	F-170	sSpPID0FFZ3	1		
171	*	F-171	sSpPIDUpdateZ3	0		
172	*	F-172	sSpMTL0FFZ4	1		
173	*	F-173	sSpMTLUpdateZ4	0		
174	*	F-174	sSpPID0FFZ4	1		
175	*	F-175	sSpPIDUpdateZ4	0		
176

Flag

Create

Retentive

Type: Flag

	*	ID	Tagname	Initial Value	Physical I/O	Description
181	*	F-181	sValveManualControlZ1	1		Zonel is being controlled manually by controlling the Control Valve output directly
182	*	F-182	sValveAutoControlZ1	0		Zonel is being controlled by either the PID or MTL algorithm which calculates an appropriate Control Valve output to reach the desired SP
183	*	F-183	sValveManualControlZ2	1		
184	*	F-184	sValveAutoControlZ2	0		
185	*	F-185	sValveManualControlZ3	1		
186	*	F-186	sValveAutoControlZ3	0		
187	*	F-187	sValveManualControlZ4	1		
188	*	F-188	sValveAutoControlZ4	0		
189						
190	*	F-190	sPIDAlgOFFZ1	1		PID algorithm is not running on Zonel, which means sValveManualControl should be active
191	*	F-191	sPIDAlgCalcZ1	0		PID algorithm is active on Zonel, in PID and/or MTL control
192	*	F-192	sMTLAlgOFFZ1	1		MTL algorithm is not running on Zonel, either running CV or PID
193	*	F-193	sMTLAlgCalcZ1	0		MTL algorithm is active on Zonel
194	*	F-194	sPIDAlgOFFZ2	1		
195	*	F-195	sPIDAlgCalcZ2	0		
196	*	F-196	sMTLAlgOFFZ2	1		
197	*	F-197	sMTLAlgCalcZ2	0		
198	*	F-198	sPIDAlgOFFZ3	1		
199	*	F-199	sPIDAlgCalcZ3	0		
200	*	F-200	sMTLAlgOFFZ3	1		
201	*	F-201	sMTLAlgCalcZ3	0		
202	*	F-202	sPIDAlgOFFZ4	1		
203	*	F-203	sPIDAlgCalcZ4	0		
204	*	F-204	sMTLAlgOFFZ4	1		
205	*	F-205	sMTLAlgCalcZ4	0		
206						
207	*	F-207	hfMTLConstsDefinedZ1	0		HMI tag: flag to signal that the MTL consts for a zone are defined so that the MTL algorithm will perform correctly
208	*	F-208	hfMTLConstsDefinedZ2	0		
209	*	F-209	hfMTLConstsDefinedZ3	0		
210	*	F-210	hfMTLConstsDefinedZ4	0		
211	*	F-211	hfRXData_ConstsDefined	0		temporary register for recipe xfer of control data, if MTL Consts defined, then =1
212	*	F-212	fSPRampYNZ1	0		bit that signifies whether or not current segment SP is a ramp SP or not, 0=dwell, 1=ramp
213	*	F-213	fSPRampYNZ2	0		
214	*	F-214	fSPRampYNZ3	0		
215	*	F-215	fSPRampYNZ4	0		
216	*	F-216	hfRXData_RampYN_1	0		temporary register for recipe xfer of segment data
217	*	F-217	hfRXData_RampYN_2	0		
218	*	F-218	hfRXData_RampYN_3	0		
219	*	F-219	hfRXData_RampYN_4	0		
220	*	F-220	fSPRampDoneZ1	0		
221	*	F-221	fSPRampDoneZ2	0		
222	*	F-222	fSPRampDoneZ3	0		
223	*	F-223	fSPRampDoneZ4	0		
224						

For Help, press F1

Flag (new)

NUM

Flag

Create

Retentive

Type: Flag

	*	ID	Tagname	Initial Value	Physical I/O	Description
230	*	F-230	sTC_OK_FluidZ1	1		thermocouple for heat transfer fluid on Zonel is operating in normal fashion
231	*	F-231	sTC_Open_FluidZ1	0		thermocouple has an open leg, triggered via PLC mapped input bit
232	*	F-232	sTC_Short_FluidZ1	0		thermocouple is shorted out, reading 0 degrees F
233	*	F-233	hfHMIConfirm_FluidZ1	0		HMI tag: TBD, button on HMI must be pressed to confirm that TC is operating correctly before allowing transition back to sTC_OK state
234	*	F-234	sTC_OK_MoldZ1	1		thermocouple attached to the mold on Zonel
235	*	F-235	sTC_Open_MoldZ1	0		
236	*	F-236	sTC_Short_MoldZ1	0		
237	*	F-237	hfHMIConfirm_MoldZ1	0		HMI tag: TBD
238						
239	*	F-239	sTC_OK_FluidZ2	1		
240	*	F-240	sTC_Open_FluidZ2	0		
241	*	F-241	sTC_Short_FluidZ2	0		
242	*	F-242	hfHMIConfirm_FluidZ2	0		
243	*	F-243	sTC_OK_MoldZ2	1		
244	*	F-244	sTC_Open_MoldZ2	0		
245	*	F-245	sTC_Short_MoldZ2	0		
246	*	F-246	hfHMIConfirm_MoldZ2	0		
247						
248	*	F-248	sTC_OK_FluidZ3	1		
249	*	F-249	sTC_Open_FluidZ3	0		
250	*	F-250	sTC_Short_FluidZ3	0		
251	*	F-251	hfHMIConfirm_FluidZ3	0		
252	*	F-252	sTC_OK_MoldZ3	1		
253	*	F-253	sTC_Open_MoldZ3	0		
254	*	F-254	sTC_Short_MoldZ3	0		
255	*	F-255	hfHMIConfirm_MoldZ3	0		
256						
257	*	F-257	sTC_OK_FluidZ4	1		
258	*	F-258	sTC_Open_FluidZ4	0		
259	*	F-259	sTC_Short_FluidZ4	0		
260	*	F-260	hfHMIConfirm_FluidZ4	0		
261	*	F-261	sTC_OK_MoldZ4	1		
262	*	F-262	sTC_Open_MoldZ4	0		
263	*	F-263	sTC_Short_MoldZ4	0		
264	*	F-264	hfHMIConfirm_MoldZ4	0		
265						
266	*	F-266	sTC_OK_CHS	1		Central Hot Supply
267	*	F-267	sTC_Open_CHS	0		
268	*	F-268	sTC_Short_CHS	0		
269	*	F-269	hfHMIConfirm_CHS	0		
270	*	F-270	sTC_OK_CHR	1		Central Hot Return
271	*	F-271	sTC_Open_CHR	0		
272	*	F-272	sTC_Short_CHR	0		
273	*	F-273	hfHMIConfirm_CHR	0		

<

For Help, press F1

Flag (new)

NUM

Flag

Create

Retentive

Type: Flag

	*	ID	Tagname	Initial Value	Physical I/O	Description
274	*	F-274	sTC_OK_CCS	1		Central Cold Supply
275	*	F-275	sTC_Open_CCS	0		
276	*	F-276	sTC_Short_CCS	0		
277	*	F-277	hfHMIConfirm_CCS	0		
278	*	F-278	sTC_OK_CCR	1		Central Cold Return
279	*	F-279	sTC_Open_CCR	0		
280	*	F-280	sTC_Short_CCR	0		
281	*	F-281	hfHMIConfirm_CCR	0		
282						
283						
284	*	F-284	sSpSegOFFZ1	1		
285	*	F-285	sSpSegUpdateZ1	0		
286	*	F-286	sSpSegLoadedZ1	0		
287	*	F-287	sSpSegOFFZ2	1		
288	*	F-288	sSpSegUpdateZ2	0		
289	*	F-289	sSpSegLoadedZ2	0		
290	*	F-290	sSpSegOFFZ3	1		
291	*	F-291	sSpSegUpdateZ3	0		
292	*	F-292	sSpSegLoadedZ3	0		
293	*	F-293	sSpSegOFFZ4	1		
294	*	F-294	sSpSegUpdateZ4	0		
295	*	F-295	sSpSegLoadedZ4	0		
296	*	F-296	fSpSegUpdateDoneZ1	0		
297	*	F-297	fSpSegUpdateDoneZ2	0		
298	*	F-298	fSpSegUpdateDoneZ3	0		
299	*	F-299	fSpSegUpdateDoneZ4	0		
300	*	F-300	sMTLTuneIdle	1		MTL tuning routine, all zones Idle
301	*	F-301	sMTLTuneInit	0		Initialize tags and what not
302	*	F-302	sMTLTuneStabilizeDone	0		All zones stabilized
303	*	F-303	sMTLTuneDone	0		All zones done testing
304	*	F-304	sMTLTuneAbort	0		Tuning routing aborted to due to HMI button or fouled thermocouple
305	*	F-305	hfMTLTuneStart	0		HMI tag: button to start the MTL tuning process
306	*	F-306	hfMTLTuneAbort	0		HMI tag: button to cancel MTL test
307	*	F-307	hfMTLTuneNoTestZ1	0		HMI tag: exclude this zone from MTL Parameter Tune testing
308	*	F-308	hfMTLTuneNoTestZ2	0		
309	*	F-309	hfMTLTuneNoTestZ3	0		
310	*	F-310	hfMTLTuneNoTestZ4	0		
311						
312	*	F-312	sMTStblzIdleZ1	1		stabilize state set for zonel
313	*	F-313	sMTStblzInitZ1	0		
314	*	F-314	sMTStblzTestZ1	0		
315	*	F-315	sMTStblzDoneZ1	0		
316	*	F-316	sMTThetaIdleZ1	1		Theta time testing for zonel
317	*	F-317	sMTThetaTestZ1	0		

For Help, press F1

Flag (new)

NUM

Flag

Create

Retentive

Type: Flag

	*	ID	Tagname	Initial Value	Physical I/O	Description
318	*	F-318	sMTThetaDoneZ1	0		
319	*	F-319	sMTTauIdleZ1	1		Tau time testing for zonel
320	*	F-320	sMTTauTestZ1	0		
321	*	F-321	sMTTauDoneZ1	0		
322	*	F-322	sMT5TauIdleZ1	1		5Tau time testing for zonel
323	*	F-323	sMT5TauTestZ1	0		
324	*	F-324	sMT5TauDoneZ1	0		
325						
326	*	F-326	sMTStblzIdleZ2	1		
327	*	F-327	sMTStblzInitZ2	0		
328	*	F-328	sMTStblzTestZ2	0		
329	*	F-329	sMTStblzDoneZ2	0		
330	*	F-330	sMTThetaIdleZ2	1		
331	*	F-331	sMTThetaTestZ2	0		
332	*	F-332	sMTThetaDoneZ2	0		
333	*	F-333	sMTTauIdleZ2	1		
334	*	F-334	sMTTauTestZ2	0		
335	*	F-335	sMTTauDoneZ2	0		
336	*	F-336	sMT5TauIdleZ2	1		
337	*	F-337	sMT5TauTestZ2	0		
338	*	F-338	sMT5TauDoneZ2	0		
339						
340	*	F-340	sMTStblzIdleZ3	1		
341	*	F-341	sMTStblzInitZ3	0		
342	*	F-342	sMTStblzTestZ3	0		
343	*	F-343	sMTStblzDoneZ3	0		
344	*	F-344	sMTThetaIdleZ3	1		
345	*	F-345	sMTThetaTestZ3	0		
346	*	F-346	sMTThetaDoneZ3	0		
347	*	F-347	sMTTauIdleZ3	1		
348	*	F-348	sMTTauTestZ3	0		
349	*	F-349	sMTTauDoneZ3	0		
350	*	F-350	sMT5TauIdleZ3	1		
351	*	F-351	sMT5TauTestZ3	0		
352	*	F-352	sMT5TauDoneZ3	0		
353						
354	*	F-354	sMTStblzIdleZ4	1		
355	*	F-355	sMTStblzInitZ4	0		
356	*	F-356	sMTStblzTestZ4	0		
357	*	F-357	sMTStblzDoneZ4	0		
358	*	F-358	sMTThetaIdleZ4	1		
359	*	F-359	sMTThetaTestZ4	0		
360	*	F-360	sMTThetaDoneZ4	0		
361	*	F-361	sMTTauIdleZ4	1		

For Help, press F1

Flag (new)

NUM

Flag

Create

Retentive

Type: Flag

	*	ID	Tagname	Initial Value	Physical I/O	Description
362	*	F-362	sMTTauTestZ4	0		
363	*	F-363	sMTTauDoneZ4	0		
364	*	F-364	sMT5TauIdleZ4	1		
365	*	F-365	sMT5TauTestZ4	0		
366	*	F-366	sMT5TauDoneZ4	0		
367						
368						
369						
370						
371						
372						
373						
374						
375						
376						
377	*	F-377	- PID tuning is TED -	0		TED, setup would be similar to MTL tuning states, these have not been coded
378	*	F-378	sPIDTuneIdle	1		not tuning to get optimal parameters
379	*	F-379	sPIDTuneInit	0		Tunes all zones that are setup for tuning
380	*	F-380	sPIDTuneStabilize	0		Point in tuning routine that sets a baseline, then actual tuning follows
381	*	F-381	sPIDTuneTesting	0		Actual tuning algorithm
382	*	F-382	sPIDTuneDone	0		yup, you got it
383	*	F-383	sPIDTuneAbort	0		you know it
384						
385	*	F-385	hfPIDTuneStart	0		HMI tag: button to start the PID tuning process
386						
387	*	F-387	placeholder PID tuning	0		
388	*	F-388	placeholder PID tuning0001	0		
389	*	F-389	placeholder PID tuning0002	0		
390	*	F-390	placeholder PID tuning0003	0		
391	*	F-391	placeholder PID tuning0004	0		
392	*	F-392	placeholder PID tuning0005	0		
393	*	F-393	placeholder PID tuning0006	0		
394	*	F-394	placeholder PID tuning0007	0		
395	*	F-395	placeholder PID tuning0008	0		
396	*	F-396	placeholder PID tuning0009	0		
397	*	F-397	placeholder PID tuning0010	0		
398	*	F-398	placeholder PID tuning0011	0		
399	*	F-399	placeholder PID tuning0012	0		
400	*	F-400	placeholder PID tuning0013	0		
401	*	F-401	placeholder PID tuning0014	0		
402	*	F-402	placeholder PID tuning0015	0		
403	*	F-403	placeholder PID tuning0016	0		
404	*	F-404	placeholder PID tuning0017	0		
405	*	F-405	placeholder PID tuning0018	0		

For Help, press F1

Flag (new)

NUM

Flag

Create

Retentive

Type: Flag

	*	ID	Tagname	Initial Value	Physical I/O	Description
500	*	F-500	fRampYN0000	0		working recipe; l=ramp seg, 0=dwell seg, part of 80 element recipe definition
501	*	F-501	fRampYN0001	0		elements 00-19 = Zone1
502	*	F-502	fRampYN0002	0		elements 20-39 = Zone2
503	*	F-503	fRampYN0003	0		elements 40-59 = Zone3
504	*	F-504	fRampYN0004	0		elements 60-79 = Zone4
505	*	F-505	fRampYN0005	0		
506	*	F-506	fRampYN0006	0		
507	*	F-507	fRampYN0007	0		
508	*	F-508	fRampYN0008	0		
509	*	F-509	fRampYN0009	0		
510	*	F-510	fRampYN0010	0		
511	*	F-511	fRampYN0011	0		
512	*	F-512	fRampYN0012	0		
513	*	F-513	fRampYN0013	0		
514	*	F-514	fRampYN0014	0		
515	*	F-515	fRampYN0015	0		
516	*	F-516	fRampYN0016	0		
517	*	F-517	fRampYN0017	0		
518	*	F-518	fRampYN0018	0		
519	*	F-519	fRampYN0019	0		
520	*	F-520	fRampYN0020	0		
521	*	F-521	fRampYN0021	0		
522	*	F-522	fRampYN0022	0		
523	*	F-523	fRampYN0023	0		
524	*	F-524	fRampYN0024	0		
525	*	F-525	fRampYN0025	0		
526	*	F-526	fRampYN0026	0		
527	*	F-527	fRampYN0027	0		
528	*	F-528	fRampYN0028	0		
529	*	F-529	fRampYN0029	0		
530	*	F-530	fRampYN0030	0		
531	*	F-531	fRampYN0031	0		
532	*	F-532	fRampYN0032	0		
533	*	F-533	fRampYN0033	0		
534	*	F-534	fRampYN0034	0		
535	*	F-535	fRampYN0035	0		
536	*	F-536	fRampYN0036	0		
537	*	F-537	fRampYN0037	0		
538	*	F-538	fRampYN0038	0		
539	*	F-539	fRampYN0039	0		
540	*	F-540	fRampYN0040	0		
541	*	F-541	fRampYN0041	0		
542	*	F-542	fRampYN0042	0		
543	*	F-543	fRampYN0043	0

Flag

Create

Retentive

Type: Flag

	*	ID	Tagname	Initial Value	Physical I/O	Description
544	*	F-544	fRampYN0044	0		
545	*	F-545	fRampYN0045	0		
546	*	F-546	fRampYN0046	0		
547	*	F-547	fRampYN0047	0		
548	*	F-548	fRampYN0048	0		
549	*	F-549	fRampYN0049	0		
550	*	F-550	fRampYN0050	0		
551	*	F-551	fRampYN0051	0		
552	*	F-552	fRampYN0052	0		
553	*	F-553	fRampYN0053	0		
554	*	F-554	fRampYN0054	0		
555	*	F-555	fRampYN0055	0		
556	*	F-556	fRampYN0056	0		
557	*	F-557	fRampYN0057	0		
558	*	F-558	fRampYN0058	0		
559	*	F-559	fRampYN0059	0		
560	*	F-560	fRampYN0060	0		
561	*	F-561	fRampYN0061	0		
562	*	F-562	fRampYN0062	0		
563	*	F-563	fRampYN0063	0		
564	*	F-564	fRampYN0064	0		
565	*	F-565	fRampYN0065	0		
566	*	F-566	fRampYN0066	0		
567	*	F-567	fRampYN0067	0		
568	*	F-568	fRampYN0068	0		
569	*	F-569	fRampYN0069	0		
570	*	F-570	fRampYN0070	0		
571	*	F-571	fRampYN0071	0		
572	*	F-572	fRampYN0072	0		
573	*	F-573	fRampYN0073	0		
574	*	F-574	fRampYN0074	0		
575	*	F-575	fRampYN0075	0		
576	*	F-576	fRampYN0076	0		
577	*	F-577	fRampYN0077	0		
578	*	F-578	fRampYN0078	0		
579	*	F-579	fRampYN0079	0		
580						
581	*	F-581	fZero0000	0		used to zero out working recipe
582	*	F-582	fZero0001	0		
583	*	F-583	fZero0002	0		
584	*	F-584	fZero0003	0		
585	*	F-585	fZero0004	0		
586	*	F-586	fZero0005	0		
587	*	F-587	fZero0006	0		

<

For Help, press F1

Flag (edit)

NUM

Flag

Create

Retentive

Type: Flag

	*	ID	Tagname	Initial Value	Physical I/O	Description
620	*	F-620	fZero0039	0		
621	*	F-621	fZero0040	0		
622	*	F-622	fZero0041	0		
623	*	F-623	fZero0042	0		
624	*	F-624	fZero0043	0		
625	*	F-625	fZero0044	0		
626	*	F-626	fZero0045	0		
627	*	F-627	fZero0046	0		
628	*	F-628	fZero0047	0		
629	*	F-629	fZero0048	0		
630	*	F-630	fZero0049	0		
631	*	F-631	fZero0050	0		
632	*	F-632	fZero0051	0		
633	*	F-633	fZero0052	0		
634	*	F-634	fZero0053	0		
635	*	F-635	fZero0054	0		
636	*	F-636	fZero0055	0		
637	*	F-637	fZero0056	0		
638	*	F-638	fZero0057	0		
639	*	F-639	fZero0058	0		
640	*	F-640	fZero0059	0		
641	*	F-641	fZero0060	0		
642	*	F-642	fZero0061	0		
643	*	F-643	fZero0062	0		
644	*	F-644	fZero0063	0		
645	*	F-645	fZero0064	0		
646	*	F-646	fZero0065	0		
647	*	F-647	fZero0066	0		
648	*	F-648	fZero0067	0		
649	*	F-649	fZero0068	0		
650	*	F-650	fZero0069	0		
651	*	F-651	fZero0070	0		
652	*	F-652	fZero0071	0		
653	*	F-653	fZero0072	0		
654	*	F-654	fZero0073	0		
655	*	F-655	fZero0074	0		
656	*	F-656	fZero0075	0		
657	*	F-657	fZero0076	0		
658	*	F-658	fZero0077	0		
659	*	F-659	fZero0078	0		
660	*	F-660	fZero0079	0		
661						
662	*	F-662	--- end FLAG definitions ---	0		end of TMD Flag datasheet
663						

Float

Create

Retentive

Type: Float

	*	ID	Tagname	Initial Value	Physical I/O	Description
9	*	FP-9	rMTLOutput_M.m_UpperBound	200.		STATIC, Upper allowable limit of MTL Algorithm Output, if M.m exceeds this, clamp it. Approx 2x upper allowable temp of 700degF, 2*700=204.79%.
10	*	FP-10	rMTLOutput_M.m_LowerBound	-100.		STATIC, Upper allowable limit of MTL Algorithm Output, if M.m exceeds this, clamp it. Approx -1x lower allowable temp of -700degF=-109.58%.
11						
12	*	FP-12	rPIDOutput_F.n_UpperBound	100.		STATIC, Upper allowable limit of PID Algorithm Output, if F.n exceeds this, clamp it
13	*	FP-13	rPIDOutput_F.n_LowerBound	0.		STATIC, Lower allowable limit of PID Algorithm Output, if F.n exceeds this, clamp it
14	*	FP-14	rPIDError_Integral_UpperBound	100.		STATIC, Upper allowable limit of PID Integrator, if 'I' term exceeds this, clamp it
15	*	FP-15	rPIDError_Integral_LowerBound	0.		STATIC, Lower allowable limit of PID Integrator, if 'I' term exceeds this, clamp it
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26	*	FP-26	hrPID_Kp_Z1	0.		HMI tag: Proportional Gain Constant for PID algorithm for Zonel, default is 1.5
27	*	FP-27	hrPID_Ki_Z1	0.		HMI tag: Integral Gain Constant for PID algorithm for Zonel, default is 4.5
28	*	FP-28	hrPID_Kd_Z1	0.		HMI tag: Derivative Gain Constant for PID algorithm for Zonel, default is 0.0
29	*	FP-29	rSP_PID_Z1	0.		current working SP [degF] for use in PID Alg error calculations
30	*	FP-30	rSPPr_PID_Z1	0.		current working SP [%range] for use in PID Alg error calculations
31	*	FP-31	rSPCascadeZ1	0.		current cascade working SP [%range] for use in PID Alg error calculations while in MTL mode
32	*	FP-32	rPIDPropPortionZ1	0.		calculated entire P portion of PID algorithm for Zonel
33	*	FP-33	rPIDIntegPortionZ1	0.		calculated entire I portion of PID algorithm for Zonel, including summer
34	*	FP-34	rPIDDerivPortionZ1	0.		calculated entire D portion of PID algorithm for Zonel
35	*	FP-35	rPIDError_e.n_Z1	0.		current sample period error for Proportional portion of PID algorithm, Zonel
36	*	FP-36	rPIDError_e.n-1_Z1	0.		first previous sample period error for Derivative portion of PID algorithm, Zonel
37						
38	*	FP-38	rPIDOutput_F.n_Z1	0.		current sample period sum of P, I, & D portions of PID algorithm, calculated as a %Range of temperature, output will be SP for ControlValve Z1
39	*	FP-39	rPID_ISE_Z1	0.		Sum of PID Squared Error for Zonel, calculated for Seg2 through last
40	*	FP-40	rSPRampRateZ1	0.		current segment's RampRate, only used if segment is a ramp segment
41	*	FP-41	rSPTemperRampZ1	0.		temperSP[degF](of Fluid if PID, or Mold if MTL) for current segZ1 from recipe, was previously integer, but had to move to float to accurately control RR
42						
43	*	FP-43	rSP_MTL_Z1	0.		current working SP [degF] for use in MTL Alg error calculations
44	*	FP-44	rSPPr_MTL_Z1	0.		current working SP [%range] for use in MTL Alg error calculations
45	*	FP-45	hrMTL_Kprime_Z1	0.		HMI tag: Entered Gain Zonel, MTL algorithm fudge factor, default is 0.33, and has never been changed on any running MTL recipes
46	*	FP-46	hrMTL_Km_Z1	0.		HMI tag: Steady State Gain Zonel, ratio of MoldTemp/FluidTemp at 5*Tau
47	*	FP-47	rMTL_Kcs_Z1	0.		calculated: Calculated Gain
48	*	FP-48	rMTL_P0_Z1	0.		calculated: MTL algorithm constant
49	*	FP-49	rMTL_P1_Z1	0.		calculated: MTL algorithm constant
50	*	FP-50	rMTL_P5_Z1	0.		calculated: MTL algorithm constant
51	*	FP-51	rMTL_P8_Z1	0.		calculated: MTL algorithm constant
52	*	FP-52	rMTL_P9_Z1	0.		HMI tag, calculated: MTL algorithm constant

For Help, press F1

Float (new)

NUM

Float

Create

Retentive

Type: Float

	*	ID	Tagname	Initial Value	Physical I/O	Description
55	*	FP-55	rMTL_KprimeTune_Z1	0.		result of MTL tuning test
56	*	FP-56	rMTL_KmTune_Z1	0.		result of MTL tuning test
57	*	FP-57	rMTL_KcsTune_Z1	0.		result of MTL tuning test
58	*	FP-58	rMTL_P0Tune_Z1	0.		result of MTL tuning test
59	*	FP-59	rMTL_P1Tune_Z1	0.		result of MTL tuning test
60	*	FP-60	rMTL_P5Tune_Z1	0.		result of MTL tuning test
61	*	FP-61	rMTL_P8Tune_Z1	0.		result of MTL tuning test
62	*	FP-62	rMTL_P9Tune_Z1	0.		result of MTL tuning test
63						
64						
65						
66						
67	*	FP-67	rMTLError_e.m_Z1	0.		current sample period error [%Range] for MTL algorithm, Zonel
68	*	FP-68	rMTLError_e.m-1_Z1	0.		first previous sample period error for MTL algorithm, Zonel
69	*	FP-69	rMTLError_e.m-2_Z1	0.		second previous sample period error for MTL algorithm, Zonel
70	*	FP-70	rMTL_ISE_Z1	0.		Sum of MTL Squared Error for Zonel, calculated for Seg2 through last
71						
72						
73	*	FP-73	rMTLOutput_M.m_Z1	0.		current sample period MTL algorithm output [%CVopen] for Zonel
74	*	FP-74	rMTLOutput_M.m-1_Z1	0.		first previous sample period MTL algorithm output for Zonel
75	*	FP-75	rMTLOutput_M.m-N-1_Z1	0.		(N-1) previous sample period MTL algorithm output for Zonel, pointer index for correct past value in OutputArray is tag nMTLOutputArrayIndexPtr_N-1_Z1
76	*	FP-76	rMTLOutput_M.m-N-2_Z1	0.		(N-2) previous sample period MTL algorithm output for Zonel, pointer index for correct past value in OutputArray is tag nMTLOutputArrayIndexPtr_N-2_Z1
77						
78						
79						
80						
81						
82						
83						
84						
85						
86						
87						
88						
89						
90						
91						
92						
93	*	FP-93	hrPID_Kp_Z2	0.		
94	*	FP-94	hrPID_Ki_Z2	0.		
95	*	FP-95	hrPID_Kd_Z2	0.		
96	*	FP-96	rSP_PID_Z2	0.		
97	*	FP-97	rSPpr_PID_Z2	0.		
98	*	FP-98	rSPCascadeZ2	0.		

Float

Create

Retentive

Type: Float

	*	ID	Tagname	Initial Value	Physical I/O	Description
99	*	FP-99	rPIDPropPortionZ2	0.		
100	*	FP-100	rPIDIntegPortionZ2	0.		
101	*	FP-101	rPIDDerivPortionZ2	0.		
102	*	FP-102	rPIDError_e.n_Z2	0.		
103	*	FP-103	rPIDError_e.n-1_Z2	0.		
104						
105	*	FP-105	rPIDOutput_F.n_Z2	0.		
106	*	FP-106	rPID_ISE_Z2	0.		
107	*	FP-107	rSPRampRateZ2	0.		
108	*	FP-108	rSPTemperRampZ2	0.		
109						
110	*	FP-110	rSP_MTL_Z2	0.		
111	*	FP-111	rSPpr_MTL_Z2	0.		
112	*	FP-112	hrMTL_Kprime_Z2	0.		
113	*	FP-113	hrMTL_Km_Z2	0.		
114	*	FP-114	rMTL_Kcs_Z2	0.		
115	*	FP-115	rMTL_P0_Z2	0.		
116	*	FP-116	rMTL_P1_Z2	0.		
117	*	FP-117	rMTL_P5_Z2	0.		
118	*	FP-118	rMTL_P8_Z2	0.		
119	*	FP-119	rMTL_P9_Z2	0.		
120						
121						
122	*	FP-122	rMTL_KprimeTune_Z2	0.		
123	*	FP-123	rMTL_KmTune_Z2	0.		
124	*	FP-124	rMTL_KcsTune_Z2	0.		
125	*	FP-125	rMTL_P0Tune_Z2	0.		
126	*	FP-126	rMTL_P1Tune_Z2	0.		
127	*	FP-127	rMTL_P5Tune_Z2	0.		
128	*	FP-128	rMTL_P8Tune_Z2	0.		
129	*	FP-129	rMTL_P9Tune_Z2	0.		
130						
131						
132						
133						
134	*	FP-134	rMTLError_e.m_Z2	0.		
135	*	FP-135	rMTLError_e.m-1_Z2	0.		
136	*	FP-136	rMTLError_e.m-2_Z2	0.		
137	*	FP-137	rMTL_ISE_Z2	0.		
138						
139						
140	*	FP-140	rMTLOutput_M.m_Z2	0.		
141	*	FP-141	rMTLOutput_M.m-1_Z2	0.		
142	*	FP-142	rMTLOutput_M.m-N-1_Z2	0.		

Float

Create

Retentive

Type: Float

	*	ID	Tagname	Initial Value	Physical I/O	Description
143	*	FP-143	rMTLOutput_M.m-N-2_Z2	0.		
144						
145						
146						
147						
148						
149						
150						
151						
152						
153						
154						
155						
156						
157						
158						
159						
160	*	FP-160	hrPID_Kp_Z3	0.		
161	*	FP-161	hrPID_Ki_Z3	0.		
162	*	FP-162	hrPID_Kd_Z3	0.		
163	*	FP-163	rSP_PID_Z3	0.		
164	*	FP-164	rSPpr_PID_Z3	0.		
165	*	FP-165	rSPCascadeZ3	0.		
166	*	FP-166	rPIDPropPortionZ3	0.		
167	*	FP-167	rPIDIntegPortionZ3	0.		
168	*	FP-168	rPIDDerivPortionZ3	0.		
169	*	FP-169	rPIDError_e.n_Z3	0.		
170	*	FP-170	rPIDError_e.n-1_Z3	0.		
171						
172	*	FP-172	rPIDOutput_F.n_Z3	0.		
173	*	FP-173	rPID_ISE_Z3	0.		
174	*	FP-174	rSPRampRateZ3	0.		
175	*	FP-175	rSPTemperRampZ3	0.		
176						
177	*	FP-177	rSP_MTL_Z3	0.		
178	*	FP-178	rSPpr_MTL_Z3	0.		
179	*	FP-179	hrMTL_Kprime_Z3	0.		
180	*	FP-180	hrMTL_Km_Z3	0.		
181	*	FP-181	rMTL_Kcs_Z3	0.		
182	*	FP-182	rMTL_P0_Z3	0.		
183	*	FP-183	rMTL_P1_Z3	0.		
184	*	FP-184	rMTL_P5_Z3	0.		
185	*	FP-185	rMTL_P8_Z3	0.		
186	*	FP-186	rMTL_P9_Z3	0.		

Float

Create

Retentive

Type: Float

	*	ID	Tagname	Initial Value	Physical I/O	Description
189	*	FP-189	rMTL_KprimeTune_Z3	0.		
190	*	FP-190	rMTL_KmTune_Z3	0.		
191	*	FP-191	rMTL_KcsTune_Z3	0.		
192	*	FP-192	rMTL_P0Tune_Z3	0.		
193	*	FP-193	rMTL_P1Tune_Z3	0.		
194	*	FP-194	rMTL_P5Tune_Z3	0.		
195	*	FP-195	rMTL_P8Tune_Z3	0.		
196	*	FP-196	rMTL_P9Tune_Z3	0.		
197						
198						
199						
200						
201	*	FP-201	rMTLError_e.m_Z3	0.		
202	*	FP-202	rMTLError_e.m-1_Z3	0.		
203	*	FP-203	rMTLError_e.m-2_Z3	0.		
204	*	FP-204	rMTL_ISE_Z3	0.		
205						
206						
207	*	FP-207	rMTLOutput_M.m_Z3	0.		
208	*	FP-208	rMTLOutput_M.m-1_Z3	0.		
209	*	FP-209	rMTLOutput_M.m-N-1_Z3	0.		
210	*	FP-210	rMTLOutput_M.m-N-2_Z3	0.		
211						
212						
213						
214						
215						
216						
217						
218						
219						
220						
221						
222						
223						
224						
225						
226						
227	*	FP-227	hrPID_Kp_Z4	0.		
228	*	FP-228	hrPID_Ki_Z4	0.		
229	*	FP-229	hrPID_Kd_Z4	0.		
230	*	FP-230	rSP_PID_Z4	0.		
231	*	FP-231	rSPpr_PID_Z4	0.		
232	*	FP-232	rSPCascadeZ4	0.		

For Help, press F1

Float (new)

NUM

Float

Create

Retentive

Type: Float

	*	ID	Tagname	Initial Value	Physical I/O	Description
233	*	FP-233	rPIDPropPortionZ4	0.		
234	*	FP-234	rPIDIntegPortionZ4	0.		
235	*	FP-235	rPIDDerivPortionZ4	0.		
236	*	FP-236	rPIDError_e.n_Z4	0.		
237	*	FP-237	rPIDError_e.n-1_Z4	0.		
238						
239	*	FP-239	rPIDOutput_F.n_Z4	0.		
240	*	FP-240	rPID_ISE_Z4	0.		
241	*	FP-241	rSPRampRateZ4	0.		
242	*	FP-242	rSPTemperRampZ4	0.		
243						
244	*	FP-244	rSP_MTL_Z4	0.		
245	*	FP-245	rSPpr_MTL_Z4	0.		
246	*	FP-246	hrMTL_Kprime_Z4	0.		
247	*	FP-247	hrMTL_Km_Z4	0.		
248	*	FP-248	rMTL_Kcs_Z4	0.		
249	*	FP-249	rMTL_P0_Z4	0.		
250	*	FP-250	rMTL_P1_Z4	0.		
251	*	FP-251	rMTL_P5_Z4	0.		
252	*	FP-252	rMTL_P8_Z4	0.		
253	*	FP-253	rMTL_P9_Z4	0.		
254						
255						
256	*	FP-256	rMTL_KprimeTune_Z4	0.		
257	*	FP-257	rMTL_KmTune_Z4	0.		
258	*	FP-258	rMTL_KcsTune_Z4	0.		
259	*	FP-259	rMTL_P0Tune_Z4	0.		
260	*	FP-260	rMTL_P1Tune_Z4	0.		
261	*	FP-261	rMTL_P5Tune_Z4	0.		
262	*	FP-262	rMTL_P8Tune_Z4	0.		
263	*	FP-263	rMTL_P9Tune_Z4	0.		
264						
265						
266						
267						
268	*	FP-268	rMTLError_e.m_Z4	0.		
269	*	FP-269	rMTLError_e.m-1_Z4	0.		
270	*	FP-270	rMTLError_e.m-2_Z4	0.		
271	*	FP-271	rMTL_ISE_Z4	0.		
272						
273						
274	*	FP-274	rMTLOutput_M.m_Z4	0.		
275	*	FP-275	rMTLOutput_M.m-1_Z4	0.		
276	*	FP-276	rMTLOutput_M.m-N-1_Z4	0.		

Float

Create

Retentive

Type: Float

	*	ID	Tagname	Initial Value	Physical I/O	Description
277	*	FP-277	rMTL0utput_M.m-N-2_Z4	0.		
278						
279						
280						
281						
282						
283						
284						
285						
286						
287						
288						
289						
290						
291						
292						
293						
294						
295						
296						
297						
298						
299						
300						
301						
302	*	FP-302	hrRXData_Kprime	0.		temporary register for recipe xfer of control data
303	*	FP-303	hrRXData_Km	0.		
304						
305						
306						
307						
308						
309						
310	*	FP-310	hrRXData_Kp	0.		temporary register for recipe xfer of control data
311	*	FP-311	hrRXData_Ki	0.		
312	*	FP-312	hrRXData_Kd	0.		
313						
314						
315						
316						
317						
318						
319						
320						

Float

Create

Retentive

Type: Float

	*	ID	Tagname	Initial Value	Physical I/O	Description
700	*	FP-700	rMTLOutputArrayZ1_0000	0.		This number is equivalent to rMTLOutput_M.m_Z1 (current sample period output)
701	*	FP-701	rMTLOutputArrayZ1_0001	0.		Array to store past 50 values of MTL algorithm output so that m-N-1 and m-N-2 subscripts Outputs can be retrieved
702	*	FP-702	rMTLOutputArrayZ1_0002	0.		Increased from Dick's MHC v2.0 of 25 slots to 50;
703	*	FP-703	rMTLOutputArrayZ1_0003	0.		therefore, max deadtime allowable increased from 230 seconds to 480 seconds based on T2 of 10 seconds
704	*	FP-704	rMTLOutputArrayZ1_0004	0.		just increased number of slots because I could, and just in case T2 changed to something less than 10
705	*	FP-705	rMTLOutputArrayZ1_0005	0.		
706	*	FP-706	rMTLOutputArrayZ1_0006	0.		This number is equivalent to rMTLOutput_M.m-6_Z1 (output 6 sample periods ago)
707	*	FP-707	rMTLOutputArrayZ1_0007	0.		
708	*	FP-708	rMTLOutputArrayZ1_0008	0.		
709	*	FP-709	rMTLOutputArrayZ1_0009	0.		
710	*	FP-710	rMTLOutputArrayZ1_0010	0.		
711	*	FP-711	rMTLOutputArrayZ1_0011	0.		
712	*	FP-712	rMTLOutputArrayZ1_0012	0.		
713	*	FP-713	rMTLOutputArrayZ1_0013	0.		
714	*	FP-714	rMTLOutputArrayZ1_0014	0.		
715	*	FP-715	rMTLOutputArrayZ1_0015	0.		
716	*	FP-716	rMTLOutputArrayZ1_0016	0.		
717	*	FP-717	rMTLOutputArrayZ1_0017	0.		
718	*	FP-718	rMTLOutputArrayZ1_0018	0.		
719	*	FP-719	rMTLOutputArrayZ1_0019	0.		
720	*	FP-720	rMTLOutputArrayZ1_0020	0.		
721	*	FP-721	rMTLOutputArrayZ1_0021	0.		
722	*	FP-722	rMTLOutputArrayZ1_0022	0.		
723	*	FP-723	rMTLOutputArrayZ1_0023	0.		
724	*	FP-724	rMTLOutputArrayZ1_0024	0.		
725	*	FP-725	rMTLOutputArrayZ1_0025	0.		
726	*	FP-726	rMTLOutputArrayZ1_0026	0.		
727	*	FP-727	rMTLOutputArrayZ1_0027	0.		
728	*	FP-728	rMTLOutputArrayZ1_0028	0.		
729	*	FP-729	rMTLOutputArrayZ1_0029	0.		
730	*	FP-730	rMTLOutputArrayZ1_0030	0.		
731	*	FP-731	rMTLOutputArrayZ1_0031	0.		
732	*	FP-732	rMTLOutputArrayZ1_0032	0.		
733	*	FP-733	rMTLOutputArrayZ1_0033	0.		
734	*	FP-734	rMTLOutputArrayZ1_0034	0.		
735	*	FP-735	rMTLOutputArrayZ1_0035	0.		
736	*	FP-736	rMTLOutputArrayZ1_0036	0.		
737	*	FP-737	rMTLOutputArrayZ1_0037	0.		
738	*	FP-738	rMTLOutputArrayZ1_0038	0.		
739	*	FP-739	rMTLOutputArrayZ1_0039	0.		
740	*	FP-740	rMTLOutputArrayZ1_0040	0.		
741	*	FP-741	rMTLOutputArrayZ1_0041	0.		
742	*	FP-742	rMTLOutputArrayZ1_0042	0.		
743	*	FP-743	rMTLOutputArrayZ1_0043	0.		

Float

Create

Retentive

Type: Float

	*	ID	Tagname	Initial Value	Physical I/O	Description
1011	*	FP-1011	rMTLOutputArrayZ4_0011	0.		
1012	*	FP-1012	rMTLOutputArrayZ4_0012	0.		
1013	*	FP-1013	rMTLOutputArrayZ4_0013	0.		
1014	*	FP-1014	rMTLOutputArrayZ4_0014	0.		
1015	*	FP-1015	rMTLOutputArrayZ4_0015	0.		
1016	*	FP-1016	rMTLOutputArrayZ4_0016	0.		
1017	*	FP-1017	rMTLOutputArrayZ4_0017	0.		
1018	*	FP-1018	rMTLOutputArrayZ4_0018	0.		
1019	*	FP-1019	rMTLOutputArrayZ4_0019	0.		
1020	*	FP-1020	rMTLOutputArrayZ4_0020	0.		
1021	*	FP-1021	rMTLOutputArrayZ4_0021	0.		
1022	*	FP-1022	rMTLOutputArrayZ4_0022	0.		
1023	*	FP-1023	rMTLOutputArrayZ4_0023	0.		
1024	*	FP-1024	rMTLOutputArrayZ4_0024	0.		
1025	*	FP-1025	rMTLOutputArrayZ4_0025	0.		
1026	*	FP-1026	rMTLOutputArrayZ4_0026	0.		
1027	*	FP-1027	rMTLOutputArrayZ4_0027	0.		
1028	*	FP-1028	rMTLOutputArrayZ4_0028	0.		
1029	*	FP-1029	rMTLOutputArrayZ4_0029	0.		
1030	*	FP-1030	rMTLOutputArrayZ4_0030	0.		
1031	*	FP-1031	rMTLOutputArrayZ4_0031	0.		
1032	*	FP-1032	rMTLOutputArrayZ4_0032	0.		
1033	*	FP-1033	rMTLOutputArrayZ4_0033	0.		
1034	*	FP-1034	rMTLOutputArrayZ4_0034	0.		
1035	*	FP-1035	rMTLOutputArrayZ4_0035	0.		
1036	*	FP-1036	rMTLOutputArrayZ4_0036	0.		
1037	*	FP-1037	rMTLOutputArrayZ4_0037	0.		
1038	*	FP-1038	rMTLOutputArrayZ4_0038	0.		
1039	*	FP-1039	rMTLOutputArrayZ4_0039	0.		
1040	*	FP-1040	rMTLOutputArrayZ4_0040	0.		
1041	*	FP-1041	rMTLOutputArrayZ4_0041	0.		
1042	*	FP-1042	rMTLOutputArrayZ4_0042	0.		
1043	*	FP-1043	rMTLOutputArrayZ4_0043	0.		
1044	*	FP-1044	rMTLOutputArrayZ4_0044	0.		
1045	*	FP-1045	rMTLOutputArrayZ4_0045	0.		
1046	*	FP-1046	rMTLOutputArrayZ4_0046	0.		
1047	*	FP-1047	rMTLOutputArrayZ4_0047	0.		
1048	*	FP-1048	rMTLOutputArrayZ4_0048	0.		
1049	*	FP-1049	rMTLOutputArrayZ4_0049	0.		
1050	*	FP-1050	rMTLOutputArrayZ4_0050	0.		
1051						
1052	*	FP-1052	--- end FLOAT definitions ---	0.		end of TND Float datasheet
1053						
1054						

Create		Type:	Input		
	*	ID	Tagname	Physical I/O	Description
0	*	I-0	xESTOP	WinPLC_Slot0i_B0	ESTOP for MHC Pump Package Power, should normally be ON (Normally Closed)
1	*	I-1	xMHCRelay	WinPLC_Slot0i_B1	existing relay in Press cabinet, signaling that the press is ready for the cycle to progress to Seg2 on MHC recipe profile. Contact closure sends input signal to PLC. Coil is energized by the press.
2	*	I-2	x2	WinPLC_Slot0i_B2	placeholder
3	*	I-3	x3	WinPLC_Slot0i_B3	placeholder
4	*	I-4	x4	WinPLC_Slot0i_B4	placeholder
5	*	I-5	x5	WinPLC_Slot0i_B5	placeholder
6	*	I-6	x6	WinPLC_Slot0i_B6	placeholder
7	*	I-7	x7	WinPLC_Slot0i_B7	placeholder
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20	*	I-20	xTC_Open_FluidZ1	WinPLC_Slot3_BrokenTx_Ch1	signals that the thermocouple has an open connection, is broken at line connections or extensions or at junction
21	*	I-21	xTC_Open_FluidZ2	WinPLC_Slot3_BrokenTx_Ch2	
22	*	I-22	xTC_Open_FluidZ3	WinPLC_Slot3_BrokenTx_Ch3	
23	*	I-23	xTC_Open_FluidZ4	WinPLC_Slot3_BrokenTx_Ch4	
24	*	I-24	xTC_Open_MoldZ1	WinPLC_Slot4_BrokenTx_Ch1	
25	*	I-25	xTC_Open_MoldZ2	WinPLC_Slot4_BrokenTx_Ch2	
26	*	I-26	xTC_Open_MoldZ3	WinPLC_Slot4_BrokenTx_Ch3	
27	*	I-27	xTC_Open_MoldZ4	WinPLC_Slot4_BrokenTx_Ch4	
28	*	I-28	xTC_Open_CentralHotSupply	WinPLC_Slot5_BrokenTx_Ch1	TBD
29	*	I-29	xTC_Open_CentralHotReturn	WinPLC_Slot5_BrokenTx_Ch2	TBD
30	*	I-30	xTC_Open_CentralColdSupply	WinPLC_Slot5_BrokenTx_Ch3	TBD
31	*	I-31	xTC_Open_CentralColdReturn	WinPLC_Slot5_BrokenTx_Ch4	TBD
32					
33	*	I-33	--- end INPUT definitions ---		end of TND Input datasheet
34					

For Help, press F1

Input (new)

NUM

DataView - MHC_TND_v3.0.1 - [Output]					
<div> <div>FileEditViewToolsLive!OptionsWindowHelp</div> <div> <div>2-34Z</div> <div>ATTACHMENT 2</div> </div> </div>					
<div> <div>CreateRetentive</div> <div>Type: Output</div> </div>					
	*	ID	Tagname	Pulse Duration	Physical I/O
0	*	0-0	yPressInhibit		WinPLC_Slotlo_B0
1	*	0-1	y1		WinPLC_Slotlo_B1
2	*	0-2	y2		WinPLC_Slotlo_B2
3	*	0-3	y3		WinPLC_Slotlo_B3
4	*	0-4	y4		WinPLC_Slotlo_B4
5	*	0-5	y5		WinPLC_Slotlo_B5
6	*	0-6	y6		WinPLC_Slotlo_B6
7	*	0-7	y7		WinPLC_Slotlo_B7
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18	*	0-18	--- end OUTPUT definitions ---		end of TMD Output datasheet
19					
<div> <div>For Help, press F1</div> <div>Output (new)</div> <div>NUM</div> </div>					

Number

Create

Retentive

Type: Number

	*	ID	Tagname	Initial Value	Physical I/O	Description
0	*	N-0	nMTLSampleTime_T2	10		set point for running MTL algorithm, default 10 seconds, set here
1	*	N-1	nPIDSampleTime_T1	2		set point for running PID algorithm, default 2 seconds, set here
2	*	N-2	nMTLAccum	0		accumulated time to trigger MTL algorithm
3	*	N-3	nPIDAccum	0		accumulated time to trigger PID algorithm
4	*	N-4	hn2_4zoneMachineSwitch	2		HMI tag: switch from HMI; 2=2-Zone, 4=4-Zone; InitValue=2 since startup state is 2-Zone
5	*	N-5	hnControlModeZ1	0		HMI tag: Current control mode Zone1, compare to static tags N-10 thru 14 below.
6	*	N-6	hnControlModeZ2	0		HMI tag
7	*	N-7	hnControlModeZ3	0		HMI tag
8	*	N-8	hnControlModeZ4	0		HMI tag
9						
10	*	N-10	nControlValve	0		STATIC, control mode value, INITIAL VALUE MUST MATCH IWS TAG DATABASE. Used to compare to hnControlModeZx.
11	*	N-11	nFluidConstTemper	1		STATIC, control mode value
12	*	N-12	nMoldConstTemper	3		STATIC, control mode value, changed from 2 to 3, 05/18/2007
13	*	N-13	nFluidRecipe	2		STATIC, control mode value, changed from 3 to 2, so could use > and < in TMD compare blocks to see if MTL control or not
14	*	N-14	nMoldRecipe	4		STATIC, control mode value
15						
16	*	N-16	nTemperRangeLowerBound	32		STATIC, minimum range of temperature = 32.0 degF, used to convert actual temperature to %range for PID/MTL algorithm calculations
17	*	N-17	nTemperRangeUpperBound	700		STATIC, maximum range of temperature = 700.0 degF, used to convert actual temperature to %range for PID/MTL algorithm calculations
18	*	N-18	nTemperRange	668		STATIC, range of temperature (700-32) degF, used to convert actual temperature to %range for PID/MTL algorithm calculations
19	*	N-19	nTemperMinHMIInput	50		STATIC, minimum HMI input temperature = 50 degF
20	*	N-20	nTC_FluidZ1	0	WinPLC_Slot3i_Ch1	Slot3iCh1: Actual temperature*10, as read by J-type thermocouple, for the heat transfer fluid going into the mold for Zone1
21	*	N-21	nTC_FluidZ2	0	WinPLC_Slot3i_Ch2	
22	*	N-22	nTC_FluidZ3	0	WinPLC_Slot3i_Ch3	
23	*	N-23	nTC_FluidZ4	0	WinPLC_Slot3i_Ch4	
24	*	N-24	nTC_MoldZ1	0	WinPLC_Slot4i_Ch1	Slot4iCh1: Actual temperature*10, as read by J-type thermocouple, for the mold on Zone1.
25	*	N-25	nTC_MoldZ2	0	WinPLC_Slot4i_Ch2	
26	*	N-26	nTC_MoldZ3	0	WinPLC_Slot4i_Ch3	
27	*	N-27	nTC_MoldZ4	0	WinPLC_Slot4i_Ch4	
28	*	N-28	nTC_CentralHotSupply	0	WinPLC_Slot5i_Ch1	Slot5iCh1: TBD: Actual temperature*10 of the fluid coming from the HOT side of the Central System, TC location also TBD
29	*	N-29	nTC_CentralHotReturn	0	WinPLC_Slot5i_Ch2	TBD: Actual temperature*10 of the fluid after it has passed through operating pump packages going back to the Central System, TC location TBD
30	*	N-30	nTC_CentralColdSupply	0	WinPLC_Slot5i_Ch3	TBD: Actual temperature*10 of the fluid coming from the COLD side of the Central System, TC location TBD
31	*	N-31	nTC_CentralColdReturn	0	WinPLC_Slot5i_Ch4	TBD: Actual temperature*10 of the fluid after it has passed through pump packages going back to the Central System, TC location TBD
32						
33						
34	*	N-34	hnUpdateClock	0		used to communicate with HMI to update PLC date/time with the HMI date/time
35	*	N-35	nPLCClockUpdated	0		PLC is done with a portion of the update, so tell HMI to move on to next portion
36						
37						
38						
39						
40	*	N-40	nCycleElapsedTime	0		overall elapsed time of the cycle
41						
42	*	N-42	nSegElapsedTimeZ1	0		elapsed running time of current segment in Zone1
43	*	N-43	hnSegLastZ1	0		HMI tag: number of last defined segment for Zone1

Number

Create

Retentive

Type: Number

	*	ID	Tagname	Initial Value	Physical I/O	Description
44	*	N-44	N-44	0		deleted
45	*	N-45	N-45	0		deleted
46	*	N-46	nSPDwellTimeTargetZ1	1		time SP [sec] for current segment in Zonel from recipe, set initial=1 so don't get error on RR calc
47	*	N-47	nSPTemperTargetZ1	0		temperature SP [degF] (of Fluid if PID, or Mold if MTL) for current segment in Zonel from recipe
48	*	N-48	N-48	0		deleted
49	*	N-49	nRampYN_Z1	0		tag contains binary encoded HMI values of REC[i].RampYN class members for each zone's 20 segments
50	*	N-50	hnMoldTemperLimitUpperZ1	0		HMI tag: Upper allowable limit for the Mold temperature on Zonel, Mold temperature must be between limits before RFCS [degF]
51	*	N-51	hnMoldTemperLimitLowerZ1	0		HMI tag: Lower allowable limit for the Mold temperature on Zonel, Mold temperature must be between limits before RFCS [degF]
52	*	N-52	hnSPTemperCycleAbortZ1	0		HMI tag: SP [degF] for Zonel in the event of a Cycle Abort
53	*	N-53	hnSPTemperConstZ1	0		HMI tag: temperature SP [degF] of Zonel when running Non-Recipe ConstTemper
54	*	N-54	hnSPControlValveManualZ1	0		HMI tag: set point for control valve [%open] when running manual (without recipe)
55	*	N-55	nSP_CV_Z1	0		current working SP [%open] of Control Valve when running manual (without recipe)
56						
57						
58	*	N-58	nSegElapsedTimeZ2	0		
59	*	N-59	hnSegLastZ2	0		
60						
61						
62	*	N-62	nSPDwellTimeTargetZ2	1		
63	*	N-63	nSPTemperTargetZ2	0		
64						
65	*	N-65	nRampYN_Z2	0		
66	*	N-66	hnMoldTemperLimitUpperZ2	0		
67	*	N-67	hnMoldTemperLimitLowerZ2	0		
68	*	N-68	hnSPTemperCycleAbortZ2	0		
69	*	N-69	hnSPTemperConstZ2	0		
70	*	N-70	hnSPControlValveManualZ2	0		
71	*	N-71	nSP_CV_Z2	0		
72						
73						
74	*	N-74	nSegElapsedTimeZ3	0		
75	*	N-75	hnSegLastZ3	0		
76						
77						
78	*	N-78	nSPDwellTimeTargetZ3	1		
79	*	N-79	nSPTemperTargetZ3	0		
80						
81	*	N-81	nRampYN_Z3	0		
82	*	N-82	hnMoldTemperLimitUpperZ3	0		
83	*	N-83	hnMoldTemperLimitLowerZ3	0		
84	*	N-84	hnSPTemperCycleAbortZ3	0		
85	*	N-85	hnSPTemperConstZ3	0		
86	*	N-86	hnSPControlValveManualZ3	0		
87	*	N-87	nSP_CV_Z3	0		

Number

Create

Retentive

Type: Number

	*	ID	Tagname	Initial Value	Physical I/O	Description
88						
89						
90	*	N-90	nSegElapsedTimeZ4	0		
91	*	N-91	hnSegLastZ4	0		
92						
93						
94	*	N-94	nSPDwellTimeTargetZ4	1		
95	*	N-95	nSPTemperTargetZ4	0		
96						
97	*	N-97	nRampYN_Z4	0		
98	*	N-98	hnMoldTemperLimitUpperZ4	0		
99	*	N-99	hnMoldTemperLimitLowerZ4	0		
100	*	N-100	hnSPTemperCycleAbortZ4	0		
101	*	N-101	hnSPTemperConstZ4	0		
102	*	N-102	hnSPControlValveManualZ4	0		
103	*	N-103	nSP_CV_Z4	0		
104						
105						
106						
107						
108	*	N-108	nMTL_N_Z1	0		calculated from MTL Consts Calc: Integer value of dead time interval. $N = \text{Theta} / T2(\text{SamplePeriod})$. Used in MTL algorithm to calculate past output subscri
109	*	N-109	hnMTL_Theta_Z1	0		HMI tag: Dead Time of mold response in seconds for Zonel
110	*	N-110	hnMTL_Tau_Z1	0		HMI tag: Mold Time Constant in seconds for Zonel
111	*	N-111	hnMTL_Lambda_Z1	0		HMI tag: specified Closed Loop Time Constant in seconds for Zonel, $\text{Lambda} \geq \text{Tau}$
112						
113						
114	*	N-114	nMTL_N_Z2	0		
115	*	N-115	hnMTL_Theta_Z2	0		
116	*	N-116	hnMTL_Tau_Z2	0		
117	*	N-117	hnMTL_Lambda_Z2	0		
118						
119						
120	*	N-120	nMTL_N_Z3	0		
121	*	N-121	hnMTL_Theta_Z3	0		
122	*	N-122	hnMTL_Tau_Z3	0		
123	*	N-123	hnMTL_Lambda_Z3	0		
124						
125						
126	*	N-126	nMTL_N_Z4	0		
127	*	N-127	hnMTL_Theta_Z4	0		
128	*	N-128	hnMTL_Tau_Z4	0		
129	*	N-129	hnMTL_Lambda_Z4	0		
130						
131						

Number

Create

Retentive

Type: Number

	*	ID	Tagname	Initial Value	Physical I/O	Description
141	*	N-141	nCV_Output_Z1	0	WinPLC_Slot2o_Ch1	Analog Output: output to affect actuation of Control Valve set on Zone1
142	*	N-142	nCV_Output_Z2	0	WinPLC_Slot2o_Ch2	Analog Output: output to affect actuation of Control Valve set on Zone2
143	*	N-143	nCV_Output_Z3	0	WinPLC_Slot2o_Ch3	Analog Output: output to affect actuation of Control Valve set on Zone3
144	*	N-144	nCV_Output_Z4	0	WinPLC_Slot2o_Ch4	Analog Output: output to affect actuation of Control Valve set on Zone4
145	*	N-145	Ch5	0	WinPLC_Slot2o_Ch5	A0 placeholder
146	*	N-146	Ch6	0	WinPLC_Slot2o_Ch6	A0 placeholder
147	*	N-147	Ch7	0	WinPLC_Slot2o_Ch7	A0 placeholder
148	*	N-148	Ch8	0	WinPLC_Slot2o_Ch8	A0 placeholder
149						
150						
151	*	N-151	nZone1	1		STATIC, used when calculating start index for zone's section of 80 element array
152	*	N-152	nZone2	2		STATIC
153	*	N-153	nZone3	3		STATIC
154	*	N-154	nZone4	4		STATIC
155						
156						
157						
158	*	N-158	nMTStblzTimeLimit	300		=5 minutes in units of seconds
159						
160	*	N-160	hnFluidStartTemperZ1	50		HMI tag for MTL tuning routine
161	*	N-161	hnMoldStartTemperZ1	50		HMI tag for MTL tuning routine
162	*	N-162	nMoldStartTemperHiLimitZ1	0		for MTL tuning routine, takes HMI Start Temper tag times 1.03
163	*	N-163	nMoldStartTemperLoLimitZ1	0		for MTL tuning routine, takes HMI Start Temper tag times 0.97
164	*	N-164	hnMoldEndTemperZ1	50		HMI tag for MTL tuning routine
165	*	N-165	nMTStblzTimeZ1	0		for MTL tuning routine, accumulation timer for stabilization period before allowing tuning to proceed
166	*	N-166	nMTThetaTimeZ1	0		for MTL tuning routine, accum timer for Theta time (mold dead time)
167	*	N-167	nMTTauTimeZ1	0		for MTL tuning routine, accum timer for Tau time (mold time const)
168	*	N-168	nMTSTauTimeTargetZ1	0		for MTL tuning routine, target time for end of test
169	*	N-169	nMTSTauTimeZ1	0		for MTL tuning routine, accum timer for 5*Tau (5 time consts=end of test)
170	*	N-170	nMTTauTemperZ1	0		for MTL tuning routine, temperature at which the response is 63.2%*SP
171						
172	*	N-172	nMTL_LambdaTune_Z1	0		result of MTL tuning test
173	*	N-173	nMTL_NTune_Z1	0		value of N (Theta/T) from tuning routine
174						
175						
176						
177						
178						
179						
180	*	N-180	hnFluidStartTemperZ2	50		
181	*	N-181	hnMoldStartTemperZ2	50		
182	*	N-182	nMoldStartTemperHiLimitZ2	0		
183	*	N-183	nMoldStartTemperLoLimitZ2	0		
184	*	N-184	hnMoldEndTemperZ2	50		

CreateRetentive

Type: Number

	*	ID	Tagname	Initial Value	Physical I/O	Description
185	*	N-185	nMTStblzTimeZ2	0		
186	*	N-186	nMTThetaTimeZ2	0		
187	*	N-187	nMTTauTimeZ2	0		
188	*	N-188	nMTSTauTimeTargetZ2	0		
189	*	N-189	nMTSTauTimeZ2	0		
190	*	N-190	nMTTauTemperZ2	0		
191						
192	*	N-192	nMTL_LambdaTune_Z2	0		
193	*	N-193	nMTL_NTune_Z2	0		
194						
195						
196						
197						
198						
199						
200	*	N-200	hnFluidStartTemperZ3	50		
201	*	N-201	hnMoldStartTemperZ3	50		
202	*	N-202	nMoldStartTemperHiLimitZ3	0		
203	*	N-203	nMoldStartTemperLoLimitZ3	0		
204	*	N-204	hnMoldEndTemperZ3	50		
205	*	N-205	nMTStblzTimeZ3	0		
206	*	N-206	nMTThetaTimeZ3	0		
207	*	N-207	nMTTauTimeZ3	0		
208	*	N-208	nMTSTauTimeTargetZ3	0		
209	*	N-209	nMTSTauTimeZ3	0		
210	*	N-210	nMTTauTemperZ3	0		
211						
212	*	N-212	nMTL_LambdaTune_Z3	0		
213	*	N-213	nMTL_NTune_Z3	0		
214						
215						
216						
217						
218						
219						
220	*	N-220	hnFluidStartTemperZ4	50		
221	*	N-221	hnMoldStartTemperZ4	50		
222	*	N-222	nMoldStartTemperHiLimitZ4	0		
223	*	N-223	nMoldStartTemperLoLimitZ4	0		
224	*	N-224	hnMoldEndTemperZ4	50		
225	*	N-225	nMTStblzTimeZ4	0		
226	*	N-226	nMTThetaTimeZ4	0		
227	*	N-227	nMTTauTimeZ4	0		
228	*	N-228	nMTSTauTimeTargetZ4	0		
229	*	N-229	nMTSTauTimeZ4	0		
230	*	N-230	nMTTauTemperZ4	0		

For Help, press F1

Number (new)NUM

CreateRetentive

Type:

Number

	*	ID	Tagname	Initial Value	Physical I/O	Description
299	*	N-299	hRXNumOfSegments	0		number of segments to tranfer based on 2 4 zone machine
300	*	N-300	hnRXData_DwellTimeSP_0	0		temporary register for recipe xfer of segment data
301	*	N-301	hnRXData_TemperSP_0	0		temporary register for recipe xfer of segment data
302	*	N-302	hnRXData_RampYN	0		temp reg for boolean encoded RampYN for each zones 20 segments
303	*	N-303	hnRXData_MoldTemperLimitUpper	0		temporary register for recipe xfer of control data
304	*	N-304	hnRXData_MoldTemperLimitLower	0		
305	*	N-305	hnRXData_SPTemperCycleAbort	0		
306	*	N-306	hnRXData_SPTemperConst	0		
307	*	N-307	hnRXData_SegLast	0		
308	*	N-308	hnRXData_CtlMode	0		
309	*	N-309	hnRXData_Theta	0		
310	*	N-310	hnRXData_Tau	0		
311	*	N-311	hnRXData_Lambda	0		
312						
313	*	N-313	hnRXData_DwellTimeSP_1	0		
314	*	N-314	hnRXData_TemperSP_1	0		
315	*	N-315	hnRXData_DwellTimeSP_2	0		
316	*	N-316	hnRXData_TemperSP_2	0		
317	*	N-317	hnRXData_DwellTimeSP_3	0		
318	*	N-318	hnRXData_TemperSP_3	0		
319	*	N-319	hnRXData_DwellTimeSP_4	0		
320	*	N-320	hnRXData_TemperSP_4	0		
321	*	N-321	hnRXData_DwellTimeSP_5	0		
322	*	N-322	hnRXData_TemperSP_5	0		
323	*	N-323	hnRXData_DwellTimeSP_6	0		
324	*	N-324	hnRXData_TemperSP_6	0		
325	*	N-325	hnRXData_DwellTimeSP_7	0		
326	*	N-326	hnRXData_TemperSP_7	0		
327	*	N-327	hnRXData_DwellTimeSP_8	0		
328	*	N-328	hnRXData_TemperSP_8	0		
329	*	N-329	hnRXData_DwellTimeSP_9	0		
330	*	N-330	hnRXData_TemperSP_9	0		
331						
332						
333						
334						
335						
336						
337						
338						
339						
340						
341						
342						
343						
344						

Create Retentive Type: Number

	*	ID	Tagname	Initial Value	Physical I/O	Description
500	*	N-500	nDwellTimeSP0000	0		working recipe, part of 80 element recipe definition
501	*	N-501	nDwellTimeSP0001	0		elements 00-19 = Zone1
502	*	N-502	nDwellTimeSP0002	0		elements 20-39 = Zone2
503	*	N-503	nDwellTimeSP0003	0		elements 40-49 = Zone3
504	*	N-504	nDwellTimeSP0004	0		elements 60-79 = Zone4
505	*	N-505	nDwellTimeSP0005	0		
506	*	N-506	nDwellTimeSP0006	0		
507	*	N-507	nDwellTimeSP0007	0		
508	*	N-508	nDwellTimeSP0008	0		
509	*	N-509	nDwellTimeSP0009	0		
510	*	N-510	nDwellTimeSP0010	0		
511	*	N-511	nDwellTimeSP0011	0		
512	*	N-512	nDwellTimeSP0012	0		
513	*	N-513	nDwellTimeSP0013	0		
514	*	N-514	nDwellTimeSP0014	0		
515	*	N-515	nDwellTimeSP0015	0		
516	*	N-516	nDwellTimeSP0016	0		
517	*	N-517	nDwellTimeSP0017	0		
518	*	N-518	nDwellTimeSP0018	0		
519	*	N-519	nDwellTimeSP0019	0		
520	*	N-520	nDwellTimeSP0020	0		
521	*	N-521	nDwellTimeSP0021	0		
522	*	N-522	nDwellTimeSP0022	0		
523	*	N-523	nDwellTimeSP0023	0		
524	*	N-524	nDwellTimeSP0024	0		
525	*	N-525	nDwellTimeSP0025	0		
526	*	N-526	nDwellTimeSP0026	0		
527	*	N-527	nDwellTimeSP0027	0		
528	*	N-528	nDwellTimeSP0028	0		
529	*	N-529	nDwellTimeSP0029	0		
530	*	N-530	nDwellTimeSP0030	0		
531	*	N-531	nDwellTimeSP0031	0		
532	*	N-532	nDwellTimeSP0032	0		
533	*	N-533	nDwellTimeSP0033	0		
534	*	N-534	nDwellTimeSP0034	0		
535	*	N-535	nDwellTimeSP0035	0		
536	*	N-536	nDwellTimeSP0036	0		
537	*	N-537	nDwellTimeSP0037	0		
538	*	N-538	nDwellTimeSP0038	0		
539	*	N-539	nDwellTimeSP0039	0		
540	*	N-540	nDwellTimeSP0040	0		
541	*	N-541	nDwellTimeSP0041	0		
542	*	N-542	nDwellTimeSP0042	0		
543	*	N-543	nDwellTimeSP0043	0		
544	*	N-544	nDwellTimeSP0044	0		
545	*	N-545	nDwellTimeSP0045	0		

Create Retentive Type: Number

	*	ID	Tagname	Initial Value	Physical I/O	Description
581	*	N-581	nTemperSP0000	0		working recipe, part of 80 element recipe definition
582	*	N-582	nTemperSP0001	0		elements 00-19 = Zone1
583	*	N-583	nTemperSP0002	0		elements 20-39 = Zone2
584	*	N-584	nTemperSP0003	0		elements 40-49 = Zone3
585	*	N-585	nTemperSP0004	0		elements 60-79 = Zone4
586	*	N-586	nTemperSP0005	0		
587	*	N-587	nTemperSP0006	0		
588	*	N-588	nTemperSP0007	0		
589	*	N-589	nTemperSP0008	0		
590	*	N-590	nTemperSP0009	0		
591	*	N-591	nTemperSP0010	0		
592	*	N-592	nTemperSP0011	0		
593	*	N-593	nTemperSP0012	0		
594	*	N-594	nTemperSP0013	0		
595	*	N-595	nTemperSP0014	0		
596	*	N-596	nTemperSP0015	0		
597	*	N-597	nTemperSP0016	0		
598	*	N-598	nTemperSP0017	0		
599	*	N-599	nTemperSP0018	0		
600	*	N-600	nTemperSP0019	0		
601	*	N-601	nTemperSP0020	0		
602	*	N-602	nTemperSP0021	0		
603	*	N-603	nTemperSP0022	0		
604	*	N-604	nTemperSP0023	0		
605	*	N-605	nTemperSP0024	0		
606	*	N-606	nTemperSP0025	0		
607	*	N-607	nTemperSP0026	0		
608	*	N-608	nTemperSP0027	0		
609	*	N-609	nTemperSP0028	0		
610	*	N-610	nTemperSP0029	0		
611	*	N-611	nTemperSP0030	0		
612	*	N-612	nTemperSP0031	0		
613	*	N-613	nTemperSP0032	0		
614	*	N-614	nTemperSP0033	0		
615	*	N-615	nTemperSP0034	0		
616	*	N-616	nTemperSP0035	0		
617	*	N-617	nTemperSP0036	0		
618	*	N-618	nTemperSP0037	0		
619	*	N-619	nTemperSP0038	0		
620	*	N-620	nTemperSP0039	0		
621	*	N-621	nTemperSP0040	0		
622	*	N-622	nTemperSP0041	0		
623	*	N-623	nTemperSP0042	0		
624	*	N-624	nTemperSP0043	0		
625	*	N-625	nTemperSP0044	0		
626	*	N-626	nTemperSP0045	0		

Create Retentive Type: Number

	*	ID	Tagname	Initial Value	Physical I/O	Description
700	*	N-700	nZero0038	0		
701	*	N-701	nZero0039	0		
702	*	N-702	nZero0040	0		
703	*	N-703	nZero0041	0		
704	*	N-704	nZero0042	0		
705	*	N-705	nZero0043	0		
706	*	N-706	nZero0044	0		
707	*	N-707	nZero0045	0		
708	*	N-708	nZero0046	0		
709	*	N-709	nZero0047	0		
710	*	N-710	nZero0048	0		
711	*	N-711	nZero0049	0		
712	*	N-712	nZero0050	0		
713	*	N-713	nZero0051	0		
714	*	N-714	nZero0052	0		
715	*	N-715	nZero0053	0		
716	*	N-716	nZero0054	0		
717	*	N-717	nZero0055	0		
718	*	N-718	nZero0056	0		
719	*	N-719	nZero0057	0		
720	*	N-720	nZero0058	0		
721	*	N-721	nZero0059	0		
722	*	N-722	nZero0060	0		
723	*	N-723	nZero0061	0		
724	*	N-724	nZero0062	0		
725	*	N-725	nZero0063	0		
726	*	N-726	nZero0064	0		
727	*	N-727	nZero0065	0		
728	*	N-728	nZero0066	0		
729	*	N-729	nZero0067	0		
730	*	N-730	nZero0068	0		
731	*	N-731	nZero0069	0		
732	*	N-732	nZero0070	0		
733	*	N-733	nZero0071	0		
734	*	N-734	nZero0072	0		
735	*	N-735	nZero0073	0		
736	*	N-736	nZero0074	0		
737	*	N-737	nZero0075	0		
738	*	N-738	nZero0076	0		
739	*	N-739	nZero0077	0		
740	*	N-740	nZero0078	0		
741	*	N-741	nZero0079	0		
742						
743	*	N-743	--- end NUMBER definitions ---	0		end of TND Number datasheet
744						
745						

DataView - MHC_TND_v3.0.1 - [String]

File

Edit

View

Tools

Live!

Options

Window

Help

ATTACHMENT 2

2-357

Create

Retentive

Type: String

	*	ID	Tagname	Max Length	Initial Setting	Physical I/O	Description
0	*	Str-0	wMHC_PLC_VersionRevision	40	PLC Version: 3.0.1		this is where the Version and Revision number of the PLC software are controlled, initial version 3.0.0
1	*	Str-1	hwRecipeName	80			recipe name downloaded to PLC from HMI
2							
3							
4							
5							
6							
7							
8							
9							
10	*	Str-10	--- end STRING definitions ---	1	0		end of TND String datasheet
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							

For Help, press F1

String (new)

NUM

DataView - MHC_TND_v3.0.1 - [Timer]

File Edit View Tools Live! Options Window Help

ATTACHMENT 2

2358

Create

Retentive

Type: Timer

	*	ID	Tagname	Preset Value (msecs)	Description
0					
1					
2					
3					
4					
5					
6					
7					
8					
9					
10	*	T-10	tHMI_Heartbeat	30000	Used in sHMI_Heartbeat state. If timer times out, then HMI must not be communicating with PLC.
11					
12					
13					
14					
15					
16					
17					
18					
19					
20	*	T-20	--- end TIMER definitions ---	1	end of TND Timer datasheet
21					
22					
23					
24					
25					

For Help, press F1

Timer (new)

NUM

DataView - MHC_TND_v3.0.1 - [System]

File

Edit

View

Tools

Live!

Options

Window

Help

ATTACHMENT 2

2359

Create

Type: System

	*	ID	Tagname	Description
0	*	SYS-0	ScanInterval	Read Only - Last scan interval (ms)
1	*	SYS-1	PeakScanInterval	Peak scan interval (ms)
2	*	SYS-2	AvgScanInterval	Read Only - Average scan interval (ms)
3	*	SYS-3	OverScanRatio	Read Only - Ratio of OverScans to ScanCount (in 10ths of a percent)
4	*	SYS-4	LogicTime	Read Only - Last amount of time it took to solve the logic and update the I/O (ms)
5	*	SYS-5	PeakLogicTime	Peak amount of time it took to solve the logic and update the I/O (ms)
6	*	SYS-6	AvgLogicTime	Read Only - Average amount of time it took to solve the logic and update the I/O (ms)
7	*	SYS-7	Reserved7	Read Only - Reserved for internal use by Think & Do RunTime
8	*	SYS-8	Reserved8	Read Only - Reserved for internal use by Think & Do RunTime
9	*	SYS-9	Reserved9	Read Only - Reserved for internal use by Think & Do RunTime
10	*	SYS-10	Reserved10	Read Only - Reserved for internal use by Think & Do RunTime
11	*	SYS-11	ArrayIndexError	Block, error code & chart # where last array index was out-of-range
12	*	SYS-12	DoLoopExit	Block, error code & chart # where last do loop was executed Max Loop Count times
13	*	SYS-13	ScanCount	Number of scans executed by the run-time
14	*	SYS-14	Reserved14	Read Only - Reserved for internal use by Think & Do RunTime
15	*	SYS-15	DataLoggingStatus	Status of data logging
16	*	SYS-16	RollingMsecCounter	Read Only - Counts the number of milliseconds (mod 2^32) since the project was started
17	*	SYS-17	LastShutdown	Read Only - Encoded time stamp giving the time of the restored retentive data
18	*	SYS-18	Reserved18	Read Only - Reserved for internal use by Think & Do RunTime
19	*	SYS-19	PowerFailShutdown	Setting this to 1 saves retentive data and stops flowchart execution and I/O scanning.
20	*	SYS-20	ScreenViewCurrentPassword	Current ScreenView password value
21	*	SYS-21	ScreenViewLevel1Password	ScreenView password value for security level 1
22	*	SYS-22	ScreenViewLevel2Password	ScreenView password value for security level 2
23	*	SYS-23	ScreenViewLevel3Password	ScreenView password value for security level 3
24	*	SYS-24	ScreenViewLevel4Password	ScreenView password value for security level 4
25	*	SYS-25	ScreenViewLevel5Password	ScreenView password value for security level 5
26	*	SYS-26	ScreenViewLevel6Password	ScreenView password value for security level 6
27	*	SYS-27	ScreenViewLevel7Password	ScreenView password value for security level 7
28	*	SYS-28	Reserved28	Read Only - Reserved for internal use by Think & Do RunTime

For Help, press F1

System (edit)

NUM

DataView - MHC_TND_v3.0.1 - [System]

File

Edit

View

Tools

Live!

Options

Window

Help

ATTACHMENT 2

Create

Type: System

	*	ID	Tagname	Description
29	*	SYS-29	OverScans	Number of scan intervals that were larger than the desired scan interval
30	*	SYS-30	LastIOTime	Read Only - Last amount of time it took to read the inputs and write the outputs (ms)
31	*	SYS-31	PeakIOTime	Peak amount of time it took to read the inputs and write the outputs (ms)
32	*	SYS-32	MinClock	Read Only - On for 30 seconds and off for 30 seconds
33	*	SYS-33	SecClock	Read Only - On for 0.5 seconds and off for 0.5 seconds
34	*	SYS-34	100msClock	Read Only - On for 50 milliseconds and off for 50 milliseconds
35	*	SYS-35	50msClock	Read Only - On for 25 milliseconds and off for 25 milliseconds
36	*	SYS-36	FirstScan	Read Only - On for the first scan, off from then on
37	*	SYS-37	AltScan	Read Only - On every other scan
38	*	SYS-38	IOStatus	Read Only - Status of the I/O
39	*	SYS-39	MathError	Block, error code & chart # where last error occurred
40	*	SYS-40	ScreenViewDataEntryKeypad	Enable ScreenView Data Entry Keypad
41	*	SYS-41	StringError	Block, error code & chart # where last error occurred
42	*	SYS-42	ConversionError	Block, error code & chart # where last error occurred
43	*	SYS-43	WriteRetentiveData	Setting this to 1 saves retentive data
44	*	SYS-44	Reserved44	Read Only - Reserved for internal use by Think & Do RunTime
45	*	SYS-45	Reserved45	Read Only - Reserved for internal use by Think & Do RunTime
46	*	SYS-46	ExtendedFunctionBlockError	Block, error code & chart # where last error occurred
47	*	SYS-47	ScreenViewDataEntryDisable	Disable ScreenView data entry
48	*	SYS-48	Reserved48	Read Only - Reserved for internal use by Think & Do RunTime
49	*	SYS-49	Reserved49	Read Only - Reserved for internal use by Think & Do RunTime
50	*	SYS-50	LastExtendedFunctionBlockTime	Read Only - Execution time (us) of the last Extended Function Block call
51	*	SYS-51	PeakExtendedFunctionBlockTime	Peak execution time (us) of all Extended Function Block calls
52	*	SYS-52	AvgExtendedFunctionBlockTime	Read Only - Average time (us) of all Extended Function Block calls
53	*	SYS-53	Spare1	Reserved for internal use by Think & Do RunTime
54	*	SYS-54	Spare2	Reserved for internal use by Think & Do RunTime
55	*	SYS-55	Spare3	Reserved for internal use by Think & Do RunTime
56	*	SYS-56	Spare4	Reserved for internal use by Think & Do RunTime
57	*	SYS-57	LiveCaptureStatus	Read Only - Status of Live! Capture

For Help, press F1

System (edit)

NUM

I/O Mappings

I/O Device Description	Data Type	Logical ID	Tagname
WinPLC_RS232Port	Comm		
WinPLC_Slot0i_B0	Input	I-0	xESTOP
WinPLC_Slot0i_B1	Input	I-1	xMHCRelay
WinPLC_Slot0i_B2	Input	I-2	x2
WinPLC_Slot0i_B3	Input	I-3	x3
WinPLC_Slot0i_B4	Input	I-4	x4
WinPLC_Slot0i_B5	Input	I-5	x5
WinPLC_Slot0i_B6	Input	I-6	x6
WinPLC_Slot0i_B7	Input	I-7	x7
WinPLC_Slot1o_B0	Output	O-0	yPressInhibit
WinPLC_Slot1o_B1	Output	O-1	y1
WinPLC_Slot1o_B2	Output	O-2	y2
WinPLC_Slot1o_B3	Output	O-3	y3
WinPLC_Slot1o_B4	Output	O-4	y4
WinPLC_Slot1o_B5	Output	O-5	y5
WinPLC_Slot1o_B6	Output	O-6	y6
WinPLC_Slot1o_B7	Output	O-7	y7
WinPLC_Slot2o_Ch1	Number	N-141	nCV_Output_Z1
WinPLC_Slot2o_Ch2	Number	N-142	nCV_Output_Z2
WinPLC_Slot2o_Ch3	Number	N-143	nCV_Output_Z3
WinPLC_Slot2o_Ch4	Number	N-144	nCV_Output_Z4
WinPLC_Slot2o_Ch5	Number	N-145	Ch5
WinPLC_Slot2o_Ch6	Number	N-146	Ch6
WinPLC_Slot2o_Ch7	Number	N-147	Ch7
WinPLC_Slot2o_Ch8	Number	N-148	Ch8
WinPLC_Slot3i_Ch1	Number	N-20	nTC_FluidZ1
WinPLC_Slot3i_Ch2	Number	N-21	nTC_FluidZ2
WinPLC_Slot3i_Ch3	Number	N-22	nTC_FluidZ3
WinPLC_Slot3i_Ch4	Number	N-23	nTC_FluidZ4
WinPLC_Slot4i_Ch1	Number	N-24	nTC_MoldZ1
WinPLC_Slot4i_Ch2	Number	N-25	nTC_MoldZ2
WinPLC_Slot4i_Ch3	Number	N-26	nTC_MoldZ3
WinPLC_Slot4i_Ch4	Number	N-27	nTC_MoldZ4
WinPLC_Slot5i_Ch1	Number	N-28	nTC_CentralHotSupply
WinPLC_Slot5i_Ch2	Number	N-29	nTC_CentralHotReturn
WinPLC_Slot5i_Ch3	Number	N-30	nTC_CentralColdSupply
WinPLC_Slot5i_Ch4	Number	N-31	nTC_CentralColdReturn

I/O View - [MHC : Configuration]

Configuration View Drivers Devices Tools Window Help

WinPLC H2-WPLC

Backplane I/O Driver

Serial Driver

WinPLC	Slot #0	Slot #1	Slot #2	Slot #3	Slot #4	Slot #5	Slot #6	Slot #7
H2-WPLC	8 In	8 Out	8 In 8 Out	4 In	4 In	4 In		

Refresh Grid

Attributes	Value
Name	Think & Do WinPLC Backplane I/O
Transport	Winsock
Protocol	IP
StartUp/ShutDown Output State	Zero

Board Info Board Status Mapping Module Info Module Status Mapping I/O Mapping

For Help, press F1

Refresh Grid

Attributes	Value
Module Type	H2-WPLC
EZ-EtherPlus (MODBUS TCP/IP) Port	Disabled

Board Info Board Status Mapping Module Info Module Status Mapping I/O Mapping

Fill Down Clear Mapping Refresh Grid

	Status Item Description	Data Type	Logical ID	Tagname	Value
1	WinPLC_ErrorExists	Input			0
2	WinPLC_ModuleErrorAtSlot	Number			-1

Board Info Board Status Mapping Module Info Module Status Mapping I/O Mapping

I/O View - [MHC : Configuration]

Configuration View Drivers Devices Tools Window Help

WinPLC H2-WPLC

Backplane I/O Driver

Serial Driver

Slot #0 8 In Slot #1 8 Out Slot #2 An 8 Out Slot #3 An 4 In Slot #4 An 4 In Slot #5 An 4 In Slot #6 Slot #7

Refresh Grid

Attributes	Value
Name	Digital DC 8 Input; D2-08ND3, 24volt
Physical Address	WinPLC_Slot0
I/O Description	WinPLC_Slot0

Board Info Board Status Mapping **Module Info** Module Status Mapping I/O Mapping

For Help, press F1

Fill Down Clear Mapping Refresh Grid

	Status Item Description	Data Type	Logical ID	Tagname	Value
1	WinPLC_Slot0_ErrorExists	Input			0
2	WinPLC_Slot0_ExtraModule	Input			0
3	WinPLC_Slot0_MissingModule	Input			0
4	WinPLC_Slot0_ModuleMismatch	Input			0

Board Info Board Status Mapping Module Info **Module Status Mapping** I/O Mapping

Fill Down Clear Mapping Refresh Grid

	I/O Device Description	Data Type	Logical ID	Tagname	Physical I/O
1	WinPLC_Slot0i_B0	Input	I-0	xESTOP	WinPLC_Slot0i_B0
2	WinPLC_Slot0i_B1	Input	I-1	xMHCRelay	WinPLC_Slot0i_B1
3	WinPLC_Slot0i_B2	Input	I-2	x2	WinPLC_Slot0i_B2
4	WinPLC_Slot0i_B3	Input	I-3	x3	WinPLC_Slot0i_B3
5	WinPLC_Slot0i_B4	Input	I-4	x4	WinPLC_Slot0i_B4
6	WinPLC_Slot0i_B5	Input	I-5	x5	WinPLC_Slot0i_B5
7	WinPLC_Slot0i_B6	Input	I-6	x6	WinPLC_Slot0i_B6
8	WinPLC_Slot0i_B7	Input	I-7	x7	WinPLC_Slot0i_B7

Board Info Board Status Mapping Module Info Module Status Mapping **I/O Mapping**

I/O View - [MHC : Configuration]

Configuration View Drivers Devices Tools Window Help

WinPLC H2-WPLC

Backplane I/O Driver

Serial Driver

Slot #0 8 In Slot #1 8 Out Slot #2 An 8 Out Slot #3 An 4 In Slot #4 An 4 In Slot #5 An 4 In Slot #6 Slot #7

Refresh Grid

Attributes	Value
Name	Relay AC/DC 8 Output; D2-08TR
Physical Address	WinPLC_Slot1
I/O Description	WinPLC_Slot1

Board Info Board Status Mapping **Module Info** Module Status Mapping I/O Mapping

For Help, press F1

Fill Down Clear Mapping Refresh Grid

	Status Item Description	Data Type	Logical ID	Tagname	Value
1	WinPLC_Slot1_ErrorExists	Input			0
2	WinPLC_Slot1_ExtraModule	Input			0
3	WinPLC_Slot1_MissingModule	Input			0
4	WinPLC_Slot1_ModuleMismatch	Input			0

Board Info Board Status Mapping Module Info **Module Status Mapping** I/O Mapping

Fill Down Clear Mapping Refresh Grid

	I/O Device Description	Data Type	Logical ID	Tagname	Physical I/O
1	WinPLC_Slot1o_B0	Output	0-0	yPressInhibit	WinPLC_Slot1o_B0
2	WinPLC_Slot1o_B1	Output	0-1	y1	WinPLC_Slot1o_B1
3	WinPLC_Slot1o_B2	Output	0-2	y2	WinPLC_Slot1o_B2
4	WinPLC_Slot1o_B3	Output	0-3	y3	WinPLC_Slot1o_B3
5	WinPLC_Slot1o_B4	Output	0-4	y4	WinPLC_Slot1o_B4
6	WinPLC_Slot1o_B5	Output	0-5	y5	WinPLC_Slot1o_B5
7	WinPLC_Slot1o_B6	Output	0-6	y6	WinPLC_Slot1o_B6
8	WinPLC_Slot1o_B7	Output	0-7	y7	WinPLC_Slot1o_B7

Board Info Board Status Mapping Module Info Module Status Mapping **I/O Mapping**

I/O View - [MHC : Configuration]

Configuration View Drivers Devices Tools Window Help

WinPLC H2-WPLC

Backplane I/O Driver

Serial Driver

WinPLC	Slot #0	Slot #1	Slot #2	Slot #3	Slot #4	Slot #5	Slot #6	Slot #7
H2-WPLC	8 In	8 Out	An 8 Out	An 4 In	An 4 In	An 4 In		

Refresh Grid

Attributes	Value
Name	Analog 4-20mA 8 Output; F2-08DA-1
Physical Address	WinPLC_Slot2
I/O Description	WinPLC_Slot2

Board Info Board Status Mapping **Module Info** Module Status Mapping I/O Mapping

For Help, press F1

Fill Down Clear Mapping Refresh Grid

	Status Item Description	Data Type	Logical ID	Tagname	Value
1	WinPLC_Slot2_ErrorExists	Input			0
2	WinPLC_Slot2_ExtraModule	Input			0
3	WinPLC_Slot2_MissingModule	Input			0
4	WinPLC_Slot2_ModuleMismatch	Input			0

Board Info Board Status Mapping Module Info **Module Status Mapping** I/O Mapping

Fill Down Clear Mapping Refresh Grid

	I/O Device Description	Data Type	Logical ID	Tagname	Physical I/O
1	WinPLC_Slot2o_Ch1	Number	N-141	nCV_Output_Z1	WinPLC_Slot2o_Ch1
2	WinPLC_Slot2o_Ch2	Number	N-142	nCV_Output_Z2	WinPLC_Slot2o_Ch2
3	WinPLC_Slot2o_Ch3	Number	N-143	nCV_Output_Z3	WinPLC_Slot2o_Ch3
4	WinPLC_Slot2o_Ch4	Number	N-144	nCV_Output_Z4	WinPLC_Slot2o_Ch4
5	WinPLC_Slot2o_Ch5	Number	N-145	Ch5	WinPLC_Slot2o_Ch5
6	WinPLC_Slot2o_Ch6	Number	N-146	Ch6	WinPLC_Slot2o_Ch6
7	WinPLC_Slot2o_Ch7	Number	N-147	Ch7	WinPLC_Slot2o_Ch7
8	WinPLC_Slot2o_Ch8	Number	N-148	Ch8	WinPLC_Slot2o_Ch8

Board Info Board Status Mapping Module Info Module Status Mapping **I/O Mapping**

I/O View - [MHC : Configuration]

Configuration View Drivers Devices Tools Window Help

WinPLC H2-WPLC

Backplane I/O Driver

Serial Driver

Slot #0 8 In Slot #1 8 Out Slot #2 An 8 Out Slot #3 An 4 In Slot #4 An 4 In Slot #5 An 4 In Slot #6 Slot #7

Refresh Grid

Attributes	Value
Name	Analog 4 Input, Thermocouple; F2-04THM
Physical Address	WinPLC_Slot3
I/O Description	WinPLC_Slot3

Board Info Board Status Mapping **Module Info** Module Status Mapping I/O Mapping

For Help, press F1

Fill Down Clear Mapping Refresh Grid

	Status Item Description	Data Type	Logical ID	Tagname	Value
1	WinPLC_Slot3_ErrorExists	Input			0
2	WinPLC_Slot3_ExtraModule	Input			0
3	WinPLC_Slot3_MissingModule	Input			0
4	WinPLC_Slot3_ModuleMismatch	Input			0
5	WinPLC_Slot3_AnalogInChFailure	Input			0
6	WinPLC_Slot3_BrokenTx_Ch1	Input	I-20	xTC_Open_FluidZ1	0
7	WinPLC_Slot3_BrokenTx_Ch2	Input	I-21	xTC_Open_FluidZ2	0
8	WinPLC_Slot3_BrokenTx_Ch3	Input	I-22	xTC_Open_FluidZ3	0
9	WinPLC_Slot3_BrokenTx_Ch4	Input	I-23	xTC_Open_FluidZ4	0

Board Info Board Status Mapping Module Info **Module Status Mapping** I/O Mapping

Fill Down Clear Mapping Refresh Grid

	I/O Device Description	Data Type	Logical ID	Tagname	Physical I/O
1	WinPLC_Slot3i_Ch1	Number	N-20	nTC_FluidZ1	WinPLC_Slot3i_Ch1
2	WinPLC_Slot3i_Ch2	Number	N-21	nTC_FluidZ2	WinPLC_Slot3i_Ch2
3	WinPLC_Slot3i_Ch3	Number	N-22	nTC_FluidZ3	WinPLC_Slot3i_Ch3
4	WinPLC_Slot3i_Ch4	Number	N-23	nTC_FluidZ4	WinPLC_Slot3i_Ch4

Board Info Board Status Mapping Module Info Module Status Mapping **I/O Mapping**

I/O View - [MHC : Configuration]

Configuration View Drivers Devices Tools Window Help

WinPLC H2-WPLC

Backplane I/O Driver

Serial Driver

Slot #0 8 In Slot #1 8 Out Slot #2 An 8 Out Slot #3 An 4 In Slot #4 An 4 In Slot #5 An 4 In Slot #6 Slot #7

Refresh Grid

Attributes	Value
Name	An 4 In
Physical Address	WinPLC_Slot4
I/O Description	WinPLC_Slot4

Board Info Board Status Mapping **Module Info** Module Status Mapping I/O Mapping

For Help, press F1

Fill Down Clear Mapping Refresh Grid

	Status Item Description	Data Type	Logical ID	Tagname	Value
1	WinPLC_Slot4_ErrorExists	Input			0
2	WinPLC_Slot4_ExtraModule	Input			0
3	WinPLC_Slot4_MissingModule	Input			0
4	WinPLC_Slot4_ModuleMismatch	Input			0
5	WinPLC_Slot4_AnalogInChFailure	Input			0
6	WinPLC_Slot4_BrokenTx_Ch1	Input	I-24	xTC_Open_MoldZ1	0
7	WinPLC_Slot4_BrokenTx_Ch2	Input	I-25	xTC_Open_MoldZ2	0
8	WinPLC_Slot4_BrokenTx_Ch3	Input	I-26	xTC_Open_MoldZ3	0
9	WinPLC_Slot4_BrokenTx_Ch4	Input	I-27	xTC_Open_MoldZ4	0

Board Info Board Status Mapping Module Info **Module Status Mapping** I/O Mapping

Fill Down Clear Mapping Refresh Grid

	I/O Device Description	Data Type	Logical ID	Tagname	Physical I/O
1	WinPLC_Slot4i_Ch1	Number	N-24	nTC_MoldZ1	WinPLC_Slot4i_Ch1
2	WinPLC_Slot4i_Ch2	Number	N-25	nTC_MoldZ2	WinPLC_Slot4i_Ch2
3	WinPLC_Slot4i_Ch3	Number	N-26	nTC_MoldZ3	WinPLC_Slot4i_Ch3
4	WinPLC_Slot4i_Ch4	Number	N-27	nTC_MoldZ4	WinPLC_Slot4i_Ch4

Board Info Board Status Mapping Module Info Module Status Mapping **I/O Mapping**

I/O View - [MHC : Configuration]

Configuration View Drivers Devices Tools Window Help

WinPLC H2-WPLC

Backplane I/O Driver

Serial Driver

WinPLC Slot #0 8 In Slot #1 8 Out Slot #2 An 8 Out Slot #3 An 4 In Slot #4 An 4 In Slot #5 An 4 In Slot #6 Slot #7

Refresh Grid

Attributes	Value
Name	An 4 In
Physical Address	WinPLC_Slot5
I/O Description	WinPLC_Slot5

Board Info Board Status Mapping **Module Info** Module Status Mapping I/O Mapping

For Help, press F1

Fill Down Clear Mapping Refresh Grid

	Status Item Description	Data Type	Logical ID	Tagname	Value
1	WinPLC_Slot5_ErrorExists	Input			0
2	WinPLC_Slot5_ExtraModule	Input			0
3	WinPLC_Slot5_MissingModule	Input			0
4	WinPLC_Slot5_ModuleMismatch	Input			0
5	WinPLC_Slot5_AnalogInChFailure	Input			0
6	WinPLC_Slot5_BrokenTx_Ch1	Input	I-28	xTC_Open_CentralHotSupply	0
7	WinPLC_Slot5_BrokenTx_Ch2	Input	I-29	xTC_Open_CentralHotReturn	0
8	WinPLC_Slot5_BrokenTx_Ch3	Input	I-30	xTC_Open_CentralColdSupply	0
9	WinPLC_Slot5_BrokenTx_Ch4	Input	I-31	xTC_Open_CentralColdReturn	0

Board Info Board Status Mapping Module Info **Module Status Mapping** I/O Mapping

Fill Down Clear Mapping Refresh Grid

	I/O Device Description	Data Type	Logical ID	Tagname	Physical I/O
1	WinPLC_Slot5i_Ch1	Number	N-28	nTC_CentralHotSupply	WinPLC_Slot5i_Ch1
2	WinPLC_Slot5i_Ch2	Number	N-29	nTC_CentralHotReturn	WinPLC_Slot5i_Ch2
3	WinPLC_Slot5i_Ch3	Number	N-30	nTC_CentralColdSupply	WinPLC_Slot5i_Ch3
4	WinPLC_Slot5i_Ch4	Number	N-31	nTC_CentralColdReturn	WinPLC_Slot5i_Ch4

Board Info Board Status Mapping Module Info Module Status Mapping **I/O Mapping**

End of Document

**Mold Heating and Cooling
Pump Package
Operator Interface**

**State Design
and
PLC Code Listing**

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

Appendix E: HMI Code Listing

Appendix E: HMI Code Listing

An Adobe pdf file was created that displays the HMI code listing. Due to file size, file length, and incompatibilities, the pdf was not posted in this Appendix. The pdf file can be found at
\\Public\pag\PAG Content\MHC Upgrade\documentation\Design and Maintenance\CodeListings.

Mold Heating and Cooling Pump Package Operator Interface

HMI Code Listing

Designed with InduSoft v6.1+SP4

Project 703030, Activity 70303020

April 2009

Josh Salmond

File Directory Structure

Directory \IWS_vX.Y.Z\ is referred to as the App Dir.

It's name is controlled by the SQA document, which states that the directory name shall reflect the version of the software.

For reference, notice the directory structure of a “Blank” application.

The custom portion of the software was written such that it will create the additional directories needed in the Startup Script.

File Menu: Project>Settings

Project Settings [X]

Identification | Options | Runtime Desktop | Communication | Web | Preferences

Application: C:_Josh\PAG\703030\MHC\software\MHC_Dev_v3\WVS_v3.1.0\MHC.APP

Description: Mold Heating & Cooling Pump Package Operator Interface

Revision: see Global Procedures for revision documentation

Company: Honeywell FM&T

Author: Josh Salmond

Field Equipment: Direct Logic 205 PLC with Entivity ThinkNDo WinPLC CE processor

Notes

[Empty text area]

OK Cancel

Project Settings [X]

Identification | Options | Runtime Desktop | Communication | Web | Preferences

Target system

Operator Workstation [v] Resolution: 1280x1024

4,000 tags; Runtime for WinNT/2K/XP

Automatic Translation

Translation File name: [Empty text box] Advanced...

☐ Enable Translation

Alarm History and Events

History Life Time (days): 0 History Format: Proprietary [v] Event Database... Alarm Database...

Default Database

Configure...

Shared Tags

Name: <None> [v] Configure...

OK Cancel

Project Settings

Identification Options Runtime Desktop Communication Web Preferences

☐ Titlebar: Mold Heating & Cooling Pump Package Operator Interface

☐ Minimize Box

☐ Maximize Box

☐ Close Box

☒ Start Maximized

☐ Menu Options...

☐ Resize Border

☐ Status Line

Startup screen: Startup.SG

☒ Show ???? when quality is not GOOD

☐ Hide Taskbar

☐ Enable ToolTips

☐ Auto Screen Scaling

Active area indication

☐ Show Object Edge

☐ Change Mouse Cursor

☒ Mouse Cursor

☒ Virtual Keyboard: Default: Keypad Scale: 300%

☒ Show Hint

☒ Show MIN/MAX fields

OK Cancel

Project Settings

Identification Options Runtime Desktop Communication Web Preferences

Driver and OPC: Send last state

TCP

Port: 1234 Send Period (ms): 200

☐ Enable binary control

Preloading tags from server

Timeout when executing on remote: 3000 ms

Timeout when executing on local: 1000 ms

Execution Environment

Timeout (ms): 30000

OK Cancel

Project Settings [X]

Identification Options Runtime Desktop Communication **Web** Preferences

Data Server IP Address: Send Period (ms):

☐ Disable Remote Client Commands ☒ Auto Screen Scaling
☒ Enable ToolTips ☐ Enable File Compression

Log
☐ Enable
FileName:

☐ Virtual Keyboard:
Default:
Scale:
☐ Show Hint
☐ Show Min/Max fields

Project Settings [X]

Identification Options Runtime Desktop Communication **Web** Preferences

Warning messages
☒ Display warning message before downloading screen to the target system
☒ Display confirmation message when renaming application tags
☒ Display confirmation message when changing the screen position and size
☒ Display warning message after saving symbols

Quality Feedback Service
☒ Generate a log file when an unexpected error happens
☒ Generate dump files when an unexpected error happens
☒ Enable log of memory

☐ Reset Tags Database when starting application
☐ Enforce Web functionality equivalence in local application displays
☐ Auto reload application on Viewer/Web Clients when it is changed

File Menu: Project>Status

Security Settings

Group Account

Group Account: Admin

Security Level - Development: 1 to 255

Security Level - Runtime: 1 to 255

Engineering Access:

- ☒ Project Settings
- ☒ Drivers, Data Sources
- ☒ Network Configuration
- ☒ Create, modify tags
- ☒ Create, modify screens
- ☒ Create, modify task sheets

Runtime Access:

- ☒ Start App
- ☒ Close App
- ☒ Database Spy (write)
- ☒ Task switch enabled
- ☒ CreateUser enabled
- ☒ Disable Task Manager

New... Delete Advanced... OK

Group Account

Group Account: Engineer

Security Level - Development: 1 to 150

Security Level - Runtime: 1 to 150

Engineering Access:

- ☐ Project Settings
- ☐ Drivers, Data Sources
- ☐ Network Configuration
- ☐ Create, modify tags
- ☐ Create, modify screens
- ☐ Create, modify task sheets

Runtime Access:

- ☒ Start App
- ☒ Close App
- ☐ Database Spy (write)
- ☒ Task switch enabled
- ☐ CreateUser enabled
- ☐ Disable Task Manager

New... Delete Advanced... OK

Group Account

Group Account: Operator

Security Level - Development: 1 to 20

Security Level - Runtime: 1 to 20

Engineering Access:

- ☐ Project Settings
- ☐ Drivers, Data Sources
- ☐ Network Configuration
- ☐ Create, modify tags
- ☐ Create, modify screens
- ☐ Create, modify task sheets

Runtime Access:

- ☒ Start App
- ☒ Close App
- ☐ Database Spy (write)
- ☒ Task switch enabled
- ☐ CreateUser enabled
- ☐ Disable Task Manager

New... Delete Advanced... OK

Group Account

Group Account: Guest

Security Level - Development: 1 to 20

Security Level - Runtime: 1 to 20

Engineering Access:

- ☐ Project Settings
- ☐ Drivers, Data Sources
- ☐ Network Configuration
- ☐ Create, modify tags
- ☐ Create, modify screens
- ☐ Create, modify task sheets

Runtime Access:

- ☒ Start App
- ☒ Close App
- ☐ Database Spy (write)
- ☒ Task switch enabled
- ☐ CreateUser enabled
- ☐ Disable Task Manager

New... Delete Advanced... OK

User Account

User: PAG Settings... New... Delete

Security Group: Admin ☐ User is blocked OK

User is password protected.

User Account

User: Engineer Settings... New... Delete

Security Group: Engineer ☐ User is blocked OK

User is password protected.

User Account [X]

User:
Operator [v] Settings... New...
Delete

Security Group:
Operator [v] ☐ User is blocked OK

User Account [X]

User:
Guest [v] Settings... New...
Delete

Security Group:
Operator [v] ☐ User is blocked OK

Tags Database

A text file is created when a cross-reference is performed in the software. The software uses this file to populate the XRef tab in the Output toolbar of the application when the cross-reference tool is performed for a tag. The file is stored at \<App Dir>\Database\xref.txt. This file can be viewed; however, it is not formatted or sorted to any particular fashion, other than by grouping the instances of use for each tag. It is not advisable to edit this file, only viewing is recommended.

Type a tag name or select a tag name, then click the 'XREF' tool button.

Instances of the tag are displayed in the Output window.

JAS March 2009
 Printed On: 3/30/2009 9:33 AM

Application

Name	Size	Type	Description	Scope	Startup	Min	Max	Unit	Retentive
MHCVersionRevision	0	String	tag to set HMI software version and revision number, set on Global Procedures	Server					
wMHC_PLC_VersionRevision	0	String	PLC tag: VersionRevision info of PLC software, set in tag database of PLC	Server					
WinPLCStationName	0	String	STATIC. Set here. Name of the WinPLC that IWS will communicate with via OPC.	Server	MHCWinPLC				
MACID	0	String	network MAC ID	Server					
MACAdapter	0	String	network adapter	Server					
UserPswd	0	String	temp tag to hold user input password for logging in	Server					
hn2_4zoneMachineSwitch	0	Integer	Two or Four Zone Machine; 2=2-zone, 4=4-zone:	Server	2	0	0		
sFourZoneMachine	0	Boolean	PLC state flag	Server					
nMTLSampleTime_T2	0	Integer	PLC tag: Timer for MTL algorithm sample time, default 10 seconds, set in PLC tag database	Server	0	0	0	seconds	
nPIDSampleTime_T1	0	Integer	PLC tag: Timer for PID algorithm sample time, default 2 seconds, set in PLC tag database	Server	0	0	0	seconds	
InputName	0	String	string input to IWS keypad	Server					
KeypadInput	0	Integer	/REV 3.1.0/ deleted/ input from user from keypad popup, temporary storage of input value, set limits using function switches	Server	0	0	0		
KeypadReturn	0	Integer	return value of IWS function KeyPad(); 0=Success, 1=Error	Server		0	0		
NavigationRadioButton	0	Integer	value corresponding to which button is pressed on NAV screen; don't use 0, since 0 is startup, which is assigned to no button	Server	0	0	0		
ZoneSelectedRadio	0	Integer	1=Z1, 2=Z2, etc	Server	1	0	0		
SegmentsDisplayed	0	Integer	SegmentsDisplayed: 0="Seg 0-9"; 1="Seg 10-19" on Segments screen; must use integer type for calculations to come out correct	Server	0	0	0		
SegmentOpsMessage	0	Integer	value determines display on smart message	Server		0	0		
RecipesFolderPath	0	String	set on Startup Script, location of directory that holds Recipe files	Server					
RecipeFiles	90	String	temp array used for FindFile() function to populate the class array for the grid control on RecipePopUp	Server					
RecipeNumFiles	0	Integer	number of recipes files returned by Find function in Recipes directory	Server		0	0		
RecipeList	90	RecipeList	used for populating grid control on Open Recipe Popup	Server					
SelectedRow	0	Integer	user selected row when selecting a recipe from the popup	Server		0	0		
TextFileInfo	40	String	carries info related to recipe text file; assigned in RecipeEditor ScreenScript. Reduce size after get Recipe.txt file figured.	Server					
RecipePathAndFileName	0	String	trigger tag for Recipe Script, and contains path+filename	Server					
RecipeName	0	String	recipe name without file extension	Server					
RecipeFileExtension	0	String	STATIC string: ".mhc"	Server	.mhc				
hwRecipeName	0	String	recipe downloaded to PLC from HMI	Server	empty				
RecipeHeader	0	String	Long description of recipe data file	Server					

JAS March 2009
Printed On: 3/30/2009 9:33 AM

Application

Name	Size	Type	Description	Scope	Startup	Min	Max	Unit	Retentive
RecipePartNumber	0	String	Part number(s) recipe is used for	Server					
RecipeIsSaved	0	Boolean	recipe has been saved, hide criteria for Load/Start Recipe button	Server	0				
RecipeLastViewed	0	String	last viewed file before leaving a screen	Server					
REC_MTL	4	MTL_Suite	MTL algorithm parameters and information for Recipe, element 1=Z1	Server					
REC_PID	4	PID_Suite	PID algorithm parameters and information for Recipe, element 1=Z1	Server					
REC	80	REC	Recipe Zone Segment SP Data	Server					
RecIndex	0	Integer	index position in 80 element REC class array	Server		0	0		
RecipeStartIndex	0	Integer	10x pointer in 80 element REC class array; =0,10,20...70	Server	0	0	0		
SegmentPagePointer	0	Integer	zone section pointer in 80 element REC class array; =0,20,40,60	Server		0	0		
hnRampYNZ	4	Integer	encoded integer containing the binary values of REC[i].RampYN for each zone, used in recipe xfer to PLC	Server		0	0		
setBitNumber	0	Integer	bit number (0-19) to set for consolidated hnRampYNZ[i] integer tag per zone	Server		0	0		
hnSegLastZ	4	Integer	last defined segment for a zone, established in CheckRecipe()	Server		0	0		
hnControlModeZ	4	Integer	ControlMode set by Static Tag Integer: 0=CV, 1=PIDCT, 2=PIDRec, 3=MTLCT, 4=MTLRec; Element [1]=Z1	Server	0	0	0		
ControlValve	0	Integer	STATIC=0. Used to compare to ControlModeZ[i]. INITIAL VALUE MUST MATCH TND TAG DATABASE.	Server	0	0	0		
FluidConstantTemper	0	Integer	STATIC=1. Used to compare to ControlModeZ[i]. INITIAL VALUE MUST MATCH TND TAG DATABASE.	Server	1	0	0		
MoldConstantTemper	0	Integer	STATIC=3. Used to compare to ControlModeZ[i]. INITIAL VALUE MUST MATCH TND TAG DATABASE.	Server	3	0	0		
FluidRecipe	0	Integer	STATIC=2. Used to compare to ControlModeZ[i]. INITIAL VALUE MUST MATCH TND TAG DATABASE.	Server	2	0	0		
MoldRecipe	0	Integer	STATIC=4. Used to compare to ControlModeZ[i]. INITIAL VALUE MUST MATCH TND TAG DATABASE.	Server	4	0	0		
BigButt	4	Boolean	used for increasing size of button to Big Button on "Control Mode Dialog.scr"	Server					
hnSPTemperConstZ	4	Integer	SP of zone under ConstTemper control, mold or fluid	Server	50	50	600		
hfMoldTemperLimitsZ	4	Boolean	Using Mold Temper Limits? listed in order Z1 Z2 Z3 Z4; 1=using specified MoldTemperLimits, 0=not using limits:	Server					
hnMoldTemperLimitUpperZ	4	Integer	If using Mold temper limits, the mold must be less than this limit before allowing RFCS, listed in order Z1 Z2 Z3 Z4:	Server	150	70	610		

JAS March 2009
Printed On: 3/30/2009 9:33 AM

Application

Name	Size	Type	Description	Scope	Startup	Min	Max	Unit	Retentive
hnMoldTemperLimitLowerZ	4	Integer	If using Mold temper limits, the mold must be greater than this limit before allowing RFCS, listed in order Z1 Z2 Z3 Z4:	Server	50	50	590		
hfMoldTemperLimitsChanged	0	Boolean	signal to user that limits were changed	Server					
sMoldTemperLimitsBreachZ	4	Boolean	PLC state tag: state of Mold Temper Limits criteria	Server					
hfSPTemperCycleAbortZ	4	Boolean	Using Cycle Abort SP? listed in order Z1 Z2 Z3 Z4; 1=using specified SP, 0=using last defined seg SP:	Server	0				
hnSPTemperCycleAbortZ	4	Integer	Cycle Abort Temperature Setpoint, listed in order Z1 Z2 Z3 Z4:	Server	50	50	600		
inputDwellTime	0	Integer	input Time SP from pop-up for a segment	Server	0	0	0	seconds	
inputHours	0	Integer		Server		0	100	hours	
inputMinutes	0	Integer		Server		0	59	minutes	
inputSeconds	0	Integer		Server		0	59	seconds	
inputTotalHours	0	Real		Server		0	100	hours	
inputTotalMinutes	0	Real		Server		0	6000	minutes	
inputTotalSeconds	0	Real		Server		0	360000	seconds	
TriggerSplashScreen	0	Boolean	trigger for script #2 to run timer on splash screen	Server					
Confirm	0	Boolean	trigger for Script #1	Server					
RecConfirmDlg	0	String	string descriptor used for comparison to run specific sections of Script#1: Confirm Dialog	Server					
calculateRampRate	0	Boolean	trigger for Script #3	Server					
hfStartRecipe	0	Boolean	trigger for PLC state change and Trend Task	Server					
hfManualPreheat	0	Boolean	Regarding Preheat Segments; 0=AutoPreheat=old Auto Wrap Around; 1=ManualPreheat=old INIT PRHT:	Server	0				
hfSkipPreheat	0	Boolean	trigger (=1) to skip Preheat segments, will be a one-shot if running AutoPreheat for multiple parts	Server					
hfDryCycle	0	Boolean	push button to allow a Dry Cycle	Server					
hfCycleAbort	0	Boolean	Cycle Aborted by HMI operator	Server					
sCycleNormal	0	Boolean	PLC state flag: cycle in normal state	Server	0				
fCycleAbort	0	Boolean	PLC flag: trigger script #5 signifies that the PLC caused a CycleAbort, probably due to a TC malfunction	Server					
hfRXHandshake	0	Boolean	Handshake with PLC to xfer segment data	Server					
cRXSegNumber	0	Integer	PLC tag: used xfer to increment segment# in 80 element array for transferring data from HMI to PLC	Server		0	0		
hnRXData_DwellTimeSP_0	0	Integer	temporary transfer register	Server		0	0		
hnRXData_TemperSP_0	0	Integer	temporary transfer register	Server		0	0		
hnRXData_DwellTimeSP_1	0	Integer	temporary transfer register	Server		0	0		
hnRXData_TemperSP_1	0	Integer	temporary transfer register	Server		0	0		
hnRXData_DwellTimeSP_2	0	Integer	temporary transfer register	Server		0	0		
hnRXData_TemperSP_2	0	Integer	temporary transfer register	Server		0	0		
hnRXData_DwellTimeSP_3	0	Integer	temporary transfer register	Server		0	0		
hnRXData_TemperSP_3	0	Integer	temporary transfer register	Server		0	0		
hnRXData_DwellTimeSP_4	0	Integer	temporary transfer register	Server		0	0		
hnRXData_TemperSP_4	0	Integer	temporary transfer register	Server		0	0		
hnRXData_DwellTimeSP_5	0	Integer	temporary transfer register	Server		0	0		

JAS March 2009
Printed On: 3/30/2009 9:33 AM

Application

Name	Size	Type	Description	Scope	Startup	Min	Max	Unit	Retentive
hnRXData_TemperSP_5	0	Integer	temporary transfer register	Server		0	0		
hnRXData_DwellTimeSP_6	0	Integer	temporary transfer register	Server		0	0		
hnRXData_TemperSP_6	0	Integer	temporary transfer register	Server		0	0		
hnRXData_DwellTimeSP_7	0	Integer	temporary transfer register	Server		0	0		
hnRXData_TemperSP_7	0	Integer	temporary transfer register	Server		0	0		
hnRXData_DwellTimeSP_8	0	Integer	temporary transfer register	Server		0	0		
hnRXData_TemperSP_8	0	Integer	temporary transfer register	Server		0	0		
hnRXData_DwellTimeSP_9	0	Integer	temporary transfer register	Server		0	0		
hnRXData_TemperSP_9	0	Integer	temporary transfer register	Server		0	0		
fRXSegDataStored	0	Boolean	PLC tag: handshake that PLC has received all recipe segment data	Server					
cRXZoneNumber	0	Integer	PLC tag: used in xfer to increment zone#	Server		0	0		
hnRXData_RampYN	0	Integer	temporary transfer register	Server		0	0		
hfRXData_ConstsDefined	0	Boolean	temporary transfer register	Server					
hfRXData_MoldTemperLimits	0	Boolean	temporary transfer register	Server					
hnRXData_MoldTemperLimitUpper	0	Integer	temporary transfer register	Server		0	0		
hnRXData_MoldTemperLimitLower	0	Integer	temporary transfer register	Server		0	0		
hnRXData_SPTemperCycleAbort	0	Integer	temporary transfer register	Server		0	0		
hnRXData_SPTemperConst	0	Integer	temporary transfer register	Server		0	0		
hnRXData_SegLast	0	Integer	temporary transfer register	Server		0	0		
hnRXData_CtiMode	0	Integer	temporary transfer register	Server		0	0		
hnRXData_Theta	0	Integer	temporary transfer register	Server		0	0		
hnRXData_Tau	0	Integer	temporary transfer register	Server		0	0		
hnRXData_Lambda	0	Integer	temporary transfer register	Server		0	0		
hrRXData_Kprime	0	Real	temporary transfer register	Server		0	0		
hrRXData_Km	0	Real	temporary transfer register	Server		0	0		
hrRXData_Kp	0	Real	temporary transfer register	Server		0	0		
hrRXData_Ki	0	Real	temporary transfer register	Server		0	0		
hrRXData_Kd	0	Real	temporary transfer register	Server		0	0		
fRXConDataStored	0	Boolean	PLC tag: handshake that PLC has received all recipe control data	Server					
sRXPLCLoaded	0	Boolean	PLC tag: a recipe has been xfered and is loaded on PLC	Server					
hfCancelDownload	0	Boolean	cancel the recipe download during the download wait screen	Server	0				
WaitProgress	0	Integer	progress bar when xfering recipe	Server		0	0		
WaitMessage	0	Integer	if don't come up with another use for wait progress bar, delete tag and just use text box for message	Server		0	0		
sRecipeActive	0	Boolean	PLC state flag: a recipe is active	Server					
sIDLE	0	Boolean	PLC state flag: ALL zones Idle	Server					
sCycleINIT	0	Boolean	PLC state flag: a cycle has been initiated	Server					
sPREHEAT	0	Boolean	PLC state flag: at least one zone still in Preheat	Server					
sRFCS	0	Boolean	PLC state flag: all zones done with Preheat and ready for Seg2	Server					
sRUN	0	Boolean	PLC state flag: running Seg2+	Server					
CycleStatus	0	Integer	IWS integer for smart message, based on status of PLC State Set Tags for display on Status.scr	Server		0	0		
ScriptTimeStatus	0	Integer	used for script scan rate on screen while loop	Server		0	0		

JAS March 2009
 Printed On: 3/30/2009 9:33 AM

Application

Name	Size	Type	Description	Scope	Startup	Min	Max	Unit	Retentive
NonRecipeConstTemper_MTL	4	MTL_Suite	parameters used if not running a recipe	Server					
NonRecipeConstTemper_PID	4	PID_Suite	parameters used if not running a recipe	Server					
NRCT_MTLConstsShow	0	Boolean	hide/show check box to display NRCT MTL consts	Server					
NRCT_PIDConstsShow	0	Boolean	hide/show check box to display NRCT PID consts	Server					
hnSPControlValveManualZ	4	Integer	SP for CV %open when running manual without recipe	Server		0	100		
hrTC_FluidZ	4	Real	TC/10; converted included decimal in integer to float	Server		0	0	degF	
hrTC_MoldZ	4	Real	TC/10; converted included decimal in integer to float	Server		0	0	degF	
nTC_FluidZ	4	Integer	PLC tag: value of thermocouple, degF/10	Server		0	0	degF/10	
nTC_MoldZ	4	Integer	PLC tag: value of thermocouple, degF/10	Server		0	0	degF/10	
nTC_CentralHotSupply	0	Integer	PLC tag: value of thermocouple, degF/10	Server		0	0	degF/10	
nTC_CentralHotReturn	0	Integer		Server		0	0	degF/10	
nTC_CentralColdSupply	0	Integer		Server		0	0	degF/10	
nTC_CentralColdReturn	0	Integer		Server		0	0	degF/10	
sTC_OK_FluidZ	4	Boolean	PLC state flag: state of Fluid TC, if not OK, then TC must be Open or Short	Server					
sTC_OK_MoldZ	4	Boolean	PLC state flag: state of Mold TCs, if not OK then must be Open or Short	Server					
sTC_OK_CHS	0	Boolean	PLC state flag: status of Central Hot Supply TC	Server					
sTC_OK_CHR	0	Boolean	PLC state flag: status of Central Hot Return TC	Server					
sTC_OK_CCS	0	Boolean	PLC state flag: status of Central Cold Supply TC	Server					
sTC_OK_CCR	0	Boolean	PLC state flag: status of Central Cold Return TC	Server					
yPressInhibit	0	Boolean	PLC tag: signal to PLC to turn on/off the Press Inhibit output	Server					
PLCRollingMsecCounter	0	Integer	PLC System tag	Server	0	0	0		
PLCSpare3	0	Integer	PLC System tag - individual bits are used to sync PLC clock with Windows clock	Server		0	0		
PLCReserved7	0	Integer	PLC System tag - contains WWSSMMHH as Hex	Server		0	0		
PLCReserved8	0	Integer	PLC System tag - contains DDMMYYYY as Hex	Server		0	0		
PLCReserved10	0	Integer	PLC System tag - bit 16 is PLC battery status	Server		0	0		
PLCHearbeatScan	0	Boolean	scheduler trigger for script #4 to see if PLC is alive	Server	0				
PLCPeakIOTime	0	Integer	PLC System tag - Peak amount of time it took to read the inputs and write the outputs (ms)	Server		0	0		
PLCAvgLogicTime	0	Integer	PLC System Tag - Average amount of time it took to solve the logic and update the I/O (ms)	Server		0	0		
PLCPeakLogicTime	0	Integer	PLC System Tag - Peak amount of time it took to solve the logic and update the I/O (ms)	Server		0	0		
PLCAvgScanInterval	0	Integer	PLC System Tag - Average Scan Interval (ms)	Server		0	0		
PLCPeakScanInterval	0	Integer	PLC System Tag - Peak scan interval (ms)	Server		0	0		
PLCOverScans	0	Integer	PLC System Tag - Number of scan intervals that were larger than the desired scan interval	Server		0	0		
ScanCount	0	Integer	PLC System Tag - count of scans on WinPLC	Server		0	0		
PreviousScanCount	0	Integer	comparison to PLC system tag ScanCount to check communications	Server		0	0		
ScanCountFailure	0	Integer	accumulator to compare to how many times the comm failed, if reaches successive limit then declare a comm failure	Server		0	0		

JAS March 2009
Printed On: 3/30/2009 9:33 AM

Application

Name	Size	Type	Description	Scope	Startup	Min	Max	Unit	Retentive
PLCHearbeatDead	0	Boolean	WinPLC is dead and not communicating via OPC with IWS	Server	0				
hfHMIHeartbeat	0	Boolean	scheduler pulse for PLC to check HMI heartbeat	Server	0				
SecondToggle	0	Boolean	scheduler toggles every second based on IWS clock	Server					
MHCVerRevSub	0	Boolean	global procedure; marker for IWS Xref tool	Server					
LogEvent	0	Boolean	marker for IWS xref tool to mark Trace() and SendEvent() usage locations	Server					
DailyRunSub	0	Boolean	global procedure; marker for IWS Xref tool	Server					
hstFileCreateSub	0	Boolean	global procedure; marker for IWS Xref tool, see also hstFileCreateTrigger	Server					
RecipeCMCTFunc	0	Boolean	global procedure; marker for IWS Xref tool	Server					
RecipeNameCheckFunc	0	Boolean	global procedure; marker for IWS Xref tool	Server					
RecipeNewSub	0	Boolean	global procedure; marker for IWS Xref tool	Server					
NewRecipeCreated	0	Boolean	oneshot to not ask overwrite question on 1st save of new recipe	Server					
RecipeSaveAsSub	0	Boolean	global procedure; marker for IWS Xref tool	Server					
RecipeOpenSub	0	Boolean	global procedure; marker for IWS Xref tool	Server					
RecipeOpenPLCSub	0	Boolean	global procedure; marker for IWS Xref tool	Server					
RecipeSaveThenActionSub	0	Boolean	global procedure; marker for IWS Xref tool	Server					
RecipeSaveSub	0	Boolean	global procedure; marker for IWS Xref tool	Server					
RecipeDeleteSub	0	Boolean	global procedure; marker for IWS Xref tool	Server					
LoadStartRecipe	0	Boolean	trigger for task script #10 to initiate RecipeLoadToPLC()	Server					
RecipeLoadToPLCSub	0	Boolean	global procedure; marker for IWS Xref tool	Server					
RecipeDownloadFunc	0	Boolean	global procedure; marker for IWS Xref tool	Server					
RecipeDownloadErrorMessage	0	String	AlarmWarningMessage for error msg created by RecipeDownload()	Server					
CheckRecipeFunc	0	Boolean	global procedure; marker for IWS Xref tool	Server					
CheckRecipeErrorMessage	0	String	AlarmWarningMessage for error msg created by CheckRecipe()	Server					
FindSegLastZxFunc	0	Boolean	global procedure; marker for IWS Xref tool	Server					
FindSegLastZxErrMsg	0	String	msg to display if errors found in Function	Server					
dwellCalcSub	0	Boolean	global procedure; marker for IWS Xref tool	Server					
resetALLFields2ZeroSub	0	Boolean	global procedure; marker for IWS Xref tool	Server					
resetTotalFields2ZeroSub	0	Boolean	global procedure; marker for IWS Xref tool	Server					
PopulateOpenRecipeGridControlSub	0	Boolean	global procedure; marker for IWS Xref tool	Server					
RecipeFileHeaderConstantsSub	0	Boolean	global procedure; marker for IWS Xref tool	Server					
SetMTLConstsDefinedFlagFunc	0	Boolean	global procedure; marker for IWS Xref tool	Server					
SegGraphFileCreateZoneSub	0	Boolean	global procedure; marker for IWS Xref tool	Server					
SetLambdaFunc	0	Boolean	global procedure; marker for IWS Xref tool	Server					
SetLambdaCTFunc	0	Boolean	global procedure; marker for IWS Xref tool	Server					
StartMTLTuningFunc	0	Boolean	global procedure; marker for IWS Xref tool	Server					
MTLTuningErrorMessage	0	String	AlarmWarningMessage for error msg on MTLTuning screen that setup is incorrect	Server					
SetPIDConstsFunc	0	Boolean	global procedure; marker for IWS Xref tool	Server					
WriteMTLTuningResults2NRCTSub	0	Boolean	global procedure; marker for IWS Xref tool	Server					
WriteMTLTuningResults2txtFileSub	0	Boolean	global procedure; marker for IWS Xref tool	Server					

JAS March 2009
Printed On: 3/30/2009 9:33 AM

Application

Name	Size	Type	Description	Scope	Startup	Min	Max	Unit	Retentive
WriteMTLTuningResults2recFileSub	0	Boolean	global procedure; marker for IWS Xref tool	Server					
Flag2OpenNSaveMTR	0	Boolean	notify came from MTL Tuning screen via Save2RecipeFile button, so open Rec file, assign consts, save recipe	Server					
CheckInputNameFunc	0	Boolean	global procedure; marker for IWS Xref tool	Server					
CCMRecFunc	0	Boolean	global procedure; marker for IWS Xref tool	Server					
CCMNRCTFunc	0	Boolean	global procedure; marker for IWS Xref tool	Server					
CCModes2NRCTSub	0	Boolean	global procedure; marker for IWS Xref tool	Server					
AlarmWarningMessage	0	Integer	value determines displayed text on Smart Message on AlarmWarning screen	Server	0	0	0		
TCErrormessage	0	String	AlarmWarningMessage for error msg during Load/Start routine that TCs are malfunctioning	Server					
TooManyRecipesErrorMessage	0	String	AlarmWarningMessage generated in populate grid control	Server					
OSErrorMessage	0	String	AlarmWarningMessage if on startup the OS is not what is expected	Server					
UpdatePLCClockSub	0	Boolean	global procedure; marker for IWS Xref tool	Server					
hnUpdateClock	0	Integer	integer trigger to sync PLC clock with HMI clock parameters, and used for OPC comm handshake	Server		0	0		
nPLCClockUpdated	0	Integer	PLC tag: used to handshake OPC comm	Server		0	0		
ConvertPLCClockSub	0	Boolean	global procedure; marker for IWS Xref tool	Server					
GetSNSub	0	Boolean	procedure screen script on "Input SN Dialog.scr"	Server					
disableSNScreenButtons	0	Boolean	hide input and close buttons on Input SN Dialog.scr	Server					
hfMTLTuneStart	0	Boolean	starts MTL tuning for selected zones	Server					
hfMTLTuneAbort	0	Boolean	abort MTL tuning test	Server					
hfMTLTuneNoTestZ	4	Boolean	exclude this zone from MTL tuning, element 1=Zone1	Server					
hnMoldStartTemperZ	4	Integer	SP for tuning routine, element 1=Zone1	Server	50	50	600		
hnMoldEndTemperZ	4	Integer	SP for tuning routine, element 1=Zone1	Server	50	50	600		
hnFluidStartTemperZ	4	Integer	SP for tuning routine, element 1=Zone1	Server	50	50	600		
nMTStblzTimeZ	4	Integer	PLC tag: elapsed time for stabilization, element 1=Zone1	Server		0	0		
sMTLTuneStabilizeDone	0	Boolean	PLC state flag: Stabilization criteria has been met for all zones	Server					
sMTStblzTestZ	4	Boolean	PLC state flag: for MTL Tuning routine status	Server					
sMTThetaTestZ	4	Boolean	PLC state flag: for MTL Tuning routine status	Server					
sMTTauTestZ	4	Boolean	PLC state flag: for MTL Tuning routine status	Server					
sMT5TauTestZ	4	Boolean	PLC state flag: for MTL Tuning routine status	Server					
MTL_TuneResults	4	MTL_Suite	gets results of MTL tuning test from PLC tags and populates this member's variables	Server					
PID_TuneResults	4	PID_Suite	TBD: gets results of PID tuning test	Server					
PID_MTLTuning	4	PID_Suite	PID consts to use for MTL Tuning	Server					
MTRPath	0	String	file directory path of MTL Tuning Results for those saved as txt files	Server					
MTRFileName	0	String	txt filename to store MTL Tuning Results to	Server					
OPCCommRecipeXfer	0	Boolean	script comm; trigger to enable=0/disable=1 OPC communication worksheet	Server	1				

JAS March 2009
Printed On: 3/30/2009 9:33 AM

Application

Name	Size	Type	Description	Scope	Startup	Min	Max	Unit	Retentive
OPCCommMTLTuning	0	Boolean	screen comm; trigger to enable=0/disable=1 OPC communication worksheet	Server	1				
OPCCommDiagnostic	0	Boolean	screen comm; trigger to enable=0/disable=1 OPC communication worksheet	Server	1				
OPCCommRecEdSegments	0	Boolean	screen comm; trigger to enable=0/disable=1 OPC communication worksheet	Server	1				
OPCCommRecEdConstants	0	Boolean	screen comm; trigger to enable=0/disable=1 OPC communication worksheet	Server	1				
OPCCommNonRecConstTemper	0	Boolean	screen comm; trigger to enable=0/disable=1 OPC communication worksheet	Server	1				
OPCCommPAGUtilities	0	Boolean	screen comm; trigger to enable=0/disable=1 OPC communication worksheet	Server	1				
OPCCommRecipeViewer	0	Boolean	screen comm; trigger to enable=0/disable=1 OPC communication worksheet	Server	1				
OPCCommStatus	0	Boolean	screen comm; trigger to enable=0/disable=1 OPC communication worksheet	Server	1				
OPCCommTrend	0	Boolean	screen comm; trigger to enable=0/disable=1 OPC communication worksheet	Server	1				
OPCCommDailyRun	0	Boolean	screen comm; trigger to enable=0/disable=1 OPC communication worksheet	Server	1				
DailyRunScript	0	Integer	trigger to run script#90 for daily run stuff, which runs DailyRun(): =1 then run script, =2 then script already done today	Server	0	0	0		YES
DailyRunScriptLastRunDate	0	String	retentive, same format as System Date tag	Server					YES
tag0	0	Boolean	junk for testing boolean logic commands	Server	0				
tag1	0	Boolean	same as tag0	Server					
myInput	0	String	temp tag for value of InputBox() function	Server					
myMsgAnswer	0	String	temp tag for value from answer to MsgBox() function	Server					
Integer	0	Integer	play around. examples of how to access tag properties/attributes on myNotes.scr	Server	4	0	0	none	
string	0	String	play around	Server					
AlarmEventTrialTag	0	Boolean	jack around tag to test out Alarm/Event control	Server	0				
vbTrigger	0	Boolean	trigger for task script to test run some vbscript code	Server	0				
myText	0	String	used for a dynamic debugging code section highlighter during scripts or subs, monitor in Log window	Server					
z	0	Integer	for IWS DB Spy debugging zone#	Server		0	0		
cSegNumberZ1	0	Integer	PLC tag: display on HMI the current seg# on PLC	Server		0	0		
cSegNumberZ2	0	Integer	can't use an array for these tags, because these tags are used for an array index into REC[]	Server		0	0		
cSegNumberZ3	0	Integer		Server		0	0		
cSegNumberZ4	0	Integer		Server		0	0		
nSegElapsedTimeZ1	0	Integer	PLC tag: to display on HMI	Server		0	0		
nSegElapsedTimeZ2	0	Integer		Server		0	0		
nSegElapsedTimeZ3	0	Integer		Server		0	0		
nSegElapsedTimeZ4	0	Integer		Server		0	0		

JAS March 2009
Printed On: 3/30/2009 9:33 AM

Application

Name	Size	Type	Description	Scope	Startup	Min	Max	Unit	Retentive
nCycleElapsedTime	0	Integer	PLC tag: elapsed time of cycle across all zones	Server		0	0		
CycleElapsedTimeEst	0	Integer	estimates the elapsed time once a defined cycle has been saved	Server		0	0		
rSP_MTL_Z	4	Real	PLC tag: display current temper SP if running MTL	Server		0	0		
rSP_PID_Z	4	Real	PLC tag: display current temper PID SP	Server		0	0		
nCV_Output_Z	4	Integer	PLC tag: current value of output to valve 0-4095	Server		0	0		
rPID_ISE_Z1	0	Real	PLC tag: to display on HMI	Server		0	0		
rPID_ISE_Z2	0	Real		Server		0	0		
rPID_ISE_Z3	0	Real		Server		0	0		
rPID_ISE_Z4	0	Real		Server		0	0		
rMTL_ISE_Z1	0	Real	PLC tag: to display on HMI	Server		0	0		
rMTL_ISE_Z2	0	Real		Server		0	0		
rMTL_ISE_Z3	0	Real		Server		0	0		
rMTL_ISE_Z4	0	Real		Server		0	0		
TrendScaleMaxY	0	Integer		Server	500	0	0	degF	
TrendScaleMinY	0	Integer		Server	50	0	0	degF	
TrendSegGrapherScaleMaxY	0	Integer		Server	550	0	0	degF	
TrendSegGrapherScaleMinY	0	Integer		Server	50	0	0	degF	
TrendSegGrapherScaleMaxX	0	Integer		Server	600	0	0	seconds	
TrendSegGrapherScaleMinX	0	Integer		Server	0	0	0	seconds	
TrendUpdate	0	Boolean	scheduler trigger to update the Trend Task file data	Server					
TrendTimeScale	0	String	sets the time scale on the trend control	Server					
TrendDuration	0	String	width of Trend Time Scale hh:mm:ss	Server					
TrendCursorPosition	0	Integer	0-100%	Server		0	0		
hstFileCreateTrigger	0	Boolean	trigger for script #6 creates hst filename and path, triggered by PLC state sRecipeActive	Server					
hstSerialNumber	0	String	SN only captured for Manual Prht parts	Server					
hstBatchFileName	0	String	filename to save trend graph data to *.hst	Server					
TrendGraphUpdate	0	Boolean	trigger script #7 to update/reload Trend Graph screen data; can't use TrendUpdate, cause script has to reset trigger	Server					
TrendGraphUpdateManual	0	Boolean	trigger script #7 manual option of previous, use if Recipe is not currently running	Server					
hstBatchFileReload	0	Boolean	reload data file on trend graph	Server					
TrendSaveSettings	0	Boolean	save look and feel of Trend Control; only have access to this in debug mode cause don't want operators changing the look	Server					
TrendLoadSavedSettings	0	Boolean	load saved settings from Trend Control to have look and feel they same	Server					
TrendCursorXValue	0	String	Ouputs the time value for the given cursor position of the Trend Graph	Server					
TrendTimeScaleHist	0	String	for Historical Trend Graph Screen	Server					
TrendDurationHist	0	String	for Historical Trend Graph Screen	Server					
hstBatchFileNameHist	0	String	for Historical Trend Graph Screen	Server					
TrendGraphUpdateHist	0	Boolean	script #9 trigger for Historical	Server					
hstBatchFileReloadHist	0	Boolean	for Historical Trend Graph Screen	Server					

JAS March 2009
 Printed On: 3/30/2009 9:33 AM

Application

Name	Size	Type	Description	Scope	Startup	Min	Max	Unit	Retentive
TrendSaveSettingsHist	0	Boolean	for Historical Trend Graph Screen	Server					
TrendLoadSavedSettingsHist	0	Boolean	for Historical Trend Graph Screen	Server					
DoLoopTimer	0	Integer	used as a timer in VBScript DO Loops, for Splash screen and Wait Dialog	Server		0	0		
xMHCRelay	0	Boolean	PLC tag: MHC Relay has been activated	Server					
xESTOP	0	Boolean	PLC tag: MHC ESTOP button has been pressed	Server					

JAS March 2009
 Printed On: 3/30/2009 9:33 AM

Class

Class: MTL_Suite

Name	Type	Description
Kprime	Real	Mold Control Parameters, listed in order Z1 Z2 Z3 Z4: K', Entered Gain, default 0.33
Km	Real	Mold Control Parameters, listed in order Z1 Z2 Z3 Z4: Km, Steady State Gain, ratio of MoldTemp/FluidTemp at 5* τ
Tau	Integer	Mold Control Parameters, listed in order Z1 Z2 Z3 Z4: Tau, Mold Time Constant [seconds]
Lambda	Integer	Mold Control Parameters, listed in order Z1 Z2 Z3 Z4: Lambda, Closed Loop Time Constant [seconds], $\lambda \geq \tau$
Theta	Integer	Mold Control Parameters, listed in order Z1 Z2 Z3 Z4: Theta, Dead Time of mold response [seconds]
Kcs	Real	Mold Control Parameters, listed in order Z1 Z2 Z3 Z4: Kcs, Calculated Gain
P0	Real	Mold Control Parameters, listed in order Z1 Z2 Z3 Z4: MTL algorithm constant
P1	Real	Mold Control Parameters, listed in order Z1 Z2 Z3 Z4: MTL algorithm constant
P5	Real	Mold Control Parameters, listed in order Z1 Z2 Z3 Z4: MTL algorithm constant
P8	Real	Mold Control Parameters, listed in order Z1 Z2 Z3 Z4: MTL algorithm constant
P9	Real	Mold Control Parameters, listed in order Z1 Z2 Z3 Z4: MTL algorithm constant
ConstsDefined	Boolean	1=MTL constants are defined

Class: PID_Suite

Name	Type	Description
Kp	Real	Fluid Control Parameters, listed in order Z1 Z2 Z3 Z4: Kp, Proportional Gain Constant, default is 1.5 [unit-less]
Ki	Real	Fluid Control Parameters, listed in order Z1 Z2 Z3 Z4: Ki, Integral Gain Constant, default is 4.5 [rate/minute]
Kd	Real	Fluid Control Parameters, listed in order Z1 Z2 Z3 Z4: Kd, Derivative Gain Constant, default is 0.0 [minutes]

Class: REC

Name	Type	Description
DwellTimeSP	Integer	desired elapsed time in seconds of the given segment
TemperSP	Integer	desired temperature in degF to obtain for the given segment
RampYN	Boolean	Is this Segment a Ramp? (=1), or a Dwell? (=0)
RampRate	Real	If a Ramp segment, this is the Ramp Rate in degF/sec
Comment	String	any comments the user may wish to associate with the segment

Class: RecipeList

Name	Type	Description
FileName	String	Windows file name of the recipe, without file extension
PartNumber	String	part number the recipe is associated with, as specified on HMI input field

JAS March 2009

Printed On: 3/30/2009 9:33 AM

System

Name	Size	Type	Description	Scope
Date	0	String	date: mm-dd-yyyy	Local
Time	0	String	time: hh:mm:ss	Local
Hour	0	Integer	hour (0 - 24)	Local
Minute	0	Integer	minute (0 - 59)	Local
Second	0	Integer	second (0 - 59)	Local
Day	0	Integer	day (0 - 31)	Local
Month	0	Integer	month(1 - 12)	Local
Year	0	Integer	year (1994 -)	Local
DayOfYear	0	Integer	day of the year (1 - 366)	Local
Weekday	0	Integer	day of the week (0=Sunday to 6=Saturday)	Local
Yesterday	0	Integer	yesterday (1 - 31)	Local
Tomorrow	0	Integer	tommorrow (1 - 31)	Local
Goto	0	String	Goto, math function	Local
Next	0	Integer	Next, math function	Local
Label	0	String	Label, math function	Local
Alarm	0	String	Most priority unacknowledge alarm	Server
Hint	0	String	Current displayed hint	Local
BlinkSlow	0	Boolean	Slow blink trigger	Local
BlinkFast	0	Boolean	Fast blink trigger	Local
Reserved__1	0	Boolean	Reserved tag	Local
Reserved__2	0	Boolean	Reserved tag	Local
AckAlr	0	Boolean	Ack the message displayed by tag "alarm"	Server
AckAll	0	Boolean	Ack all alarms	Server
LptOff	0	Boolean	Disable alarm printing	Local
BeepOff	0	Boolean	Disable alarm beep	Local
Beep	0	Boolean	Beep state, 1=ON 0=OFF	Local
UserName	0	String	Current User	Local
GroupName	0	String	Current User Group	Local
GroupLoLevel	0	Integer	Start of current security level range	Local
GroupHiLevel	0	Integer	end of current security level range	Local
InputOutOfRange	0	Boolean	Receive a toggle on edition out of range	Local
InputMinRange	0	Real	Minimum value related to InputOutOfRange	Local
InputMaxRange	0	Real	Maximum value related to InputOutOfRange	Local
GroupCNFLoLevel	0	Integer	Start of current security level range to CNF	Local
GroupCNFHiLevel	0	Integer	end of current security level range to CNF	Local
CrispInput	0	String	Crisp status-bar input value	Local

JAS March 2009

Printed On: 3/30/2009 9:33 AM

System

Name	Size	Type	Description	Scope
CrispOutput	0	String	Crisp status-bar output value	Local
CrispDisplay	0	Integer	Crisp status-bar output value	Local
AnalogValue_	0	Real	Simulation analog tag	Local
DigitalValue_	0	Boolean	Simulation digital tag	Local
LastCodeChar_	0	Integer	Last code char in the Viewer	Local
ServerTime_	0	String	Time of the server	Server
ServerDate_	0	String	Date of the server	Server
Reserved__8	0	Boolean	Reserved tag	Local
Reserved__9	0	Boolean	Reserved tag	Local
Reserved__10	0	Boolean	Reserved tag	Local

Event Settings

Event Settings

Event Settings

☒ Enable event logger Disable: Event Database...

Settings

☒ Security System ☐ Recipe ☒ Custom Messages

☒ Display ☐ Report ☒ System Warning

☒ Tags:

	Tag Name	Message
1	MHCVersionRevision	
2	WinPLCStationName	
3	hn2_4zoneMachineSwitch	{hn2_4zoneMachineSwitch->Descrip
4	hnControlModeZ[1]	{hnControlModeZ->Description}
5	hnControlModeZ[2]	{hnControlModeZ->Description}
6	hnControlModeZ[3]	{hnControlModeZ->Description}
7	hnControlModeZ[4]	{hnControlModeZ->Description}

OK Cancel

7	hnControlModeZ[4]	{hnControlModeZ->Description}
8	PLCHearbeatDead	{PLCHearbeatDead->Description}
9	hwRecipeName	{hwRecipeName->Description}
10	sRXPLCLoaded	state flag 1=loaded
11	hfDryCycle	
12	hfManualPreheat	
13	hfSkipPreheat	
14	hfStartRecipe	
15	hfCancelDownload	
16	fCycleAbort	cycle aborted by PLC
17	hfCycleAbort	cycle aborted by HMI user
18	cSegNumberZ1	
19	cSegNumberZ2	
20	cSegNumberZ3	
21	cSegNumberZ4	
22	hnSPTemperConstZ[1]	Z1 Const Temper SP changed
23	hnSPTemperConstZ[2]	Z2 Const Temper SP changed
24	hnSPTemperConstZ[3]	Z3 Const Temper SP changed
25	hnSPTemperConstZ[4]	Z4 Const Temper SP changed
26		

Workspace Toolbar: Tasks Tab>Trend

TREND002.TRD

Description:

History Format:

☒ Save On Trigger: ☐ Save On Tag Change

This worksheet will save data only while a batch is running (see Advanced configuration).

	Tag Name	Dead Band
1	hrTC_FluidZ[1]	
2	hrTC_FluidZ[2]	
3	hrTC_FluidZ[3]	
4	hrTC_FluidZ[4]	
5	hrTC_MoldZ[1]	
6	hrTC_MoldZ[2]	
7	hrTC_MoldZ[3]	
8	hrTC_MoldZ[4]	
9	rSP_MTL_Z[1]	
10	rSP_MTL_Z[2]	
11	rSP_MTL_Z[3]	
12	rSP_MTL_Z[4]	
13	rSP_PID_Z[1]	
14	rSP_PID_Z[2]	
15	rSP_PID_Z[3]	
16	rSP_PID_Z[4]	
*		

Trend Advanced Settings [X]

Batch

Start/Stop: Existent:

Name: Description:

Delete: ☐ Save data even if batch is not running.

Disk Space Control

History Life Time(days): Compress After(days):

Disable All Data Saving:

Workspace Toolbar: Tasks Tab>Recipes

Recipe 2

Recipe2.rcp

Description:

File Name: Register Number:

Options:
☐ Save As XML
☒ Unicode

	Tag Name	Number of Elements
1	RecipeName	0
2	RecipePartNumber	0
*		
*		
*		
*		
*		

Recipe 1

Recipe1.rcp

Description:

File Name: Register Number:

Options:
☐ Save As XML
☒ Unicode

	Tag Name	Number of Elements
1	RecipeName	
2	RecipePartNumber	
3	RecipeHeader	
4	TextFileInfo[0]	
5	TextFileInfo[1]	
6	TextFileInfo[0]	
7	TextFileInfo[2]	
8	TextFileInfo[3]	
9	TextFileInfo[0]	
10	TextFileInfo[4]	
11	hn2_4zoneMachineSwitch	
12	TextFileInfo[5]	
13	hnManualPreheat	
14	TextFileInfo[0]	
15	TextFileInfo[6]	//2 4switch can be 0 when \$Recipe("Init:Recipe1") command is called
16	hnControlModeZ[1]	{IF(hn2_4zoneMachineSwitch=0, 4, hn2_4zoneMachineSwitch)}
17	TextFileInfo[0]	
18	TextFileInfo[7]	
19	REC_MTL[1].Theta	{IF(hn2_4zoneMachineSwitch=0, 4, hn2_4zoneMachineSwitch)}
20	TextFileInfo[8]	
21	REC_MTL[1].Tau	{IF(hn2_4zoneMachineSwitch=0, 4, hn2_4zoneMachineSwitch)}
22	TextFileInfo[9]	
23	REC_MTL[1].Lambda	{IF(hn2_4zoneMachineSwitch=0, 4, hn2_4zoneMachineSwitch)}
24	TextFileInfo[10]	
25	REC_MTL[1].Kprime	{IF(hn2_4zoneMachineSwitch=0, 4, hn2_4zoneMachineSwitch)}
26	TextFileInfo[11]	
27	REC_MTL[1].Km	{IF(hn2_4zoneMachineSwitch=0, 4, hn2_4zoneMachineSwitch)}
28	TextFileInfo[0]	
29	TextFileInfo[12]	
30	REC_PID[1].Kp	{IF(hn2_4zoneMachineSwitch=0, 4, hn2_4zoneMachineSwitch)}
31	TextFileInfo[13]	

Recipe1.rcp

Description:

File Name: Register Number:

Options:
☐ Save As XML
☒ Unicode

	Tag Name	Number of Elements
32	REC_PID[1].Ki	{IF(hn2_4zoneMachineSwitch=0, 4, hn2_4zoneMachineSwitch)}
33	TextFileInfo[14]	
34	REC_PID[1].Kd	{IF(hn2_4zoneMachineSwitch=0, 4, hn2_4zoneMachineSwitch)}
35	TextFileInfo[0]	
36	TextFileInfo[15]	
37	hnSPTemperConstZ[1]	{IF(hn2_4zoneMachineSwitch=0, 4, hn2_4zoneMachineSwitch)}
38	TextFileInfo[0]	
39	TextFileInfo[16]	
40	hnSPTemperCycleAbortZ[1]	{IF(hn2_4zoneMachineSwitch=0, 4, hn2_4zoneMachineSwitch)}
41	TextFileInfo[17]	
42	hnSPTemperCycleAbortZ[1]	{IF(hn2_4zoneMachineSwitch=0, 4, hn2_4zoneMachineSwitch)}
43	TextFileInfo[0]	
44	TextFileInfo[18]	
45	hnMoldTemperLimitsZ[1]	{IF(hn2_4zoneMachineSwitch=0, 4, hn2_4zoneMachineSwitch)}
46	TextFileInfo[19]	
47	hnMoldTemperLimitUpperZ[1]	{IF(hn2_4zoneMachineSwitch=0, 4, hn2_4zoneMachineSwitch)}
48	TextFileInfo[20]	
49	hnMoldTemperLimitLowerZ[1]	{IF(hn2_4zoneMachineSwitch=0, 4, hn2_4zoneMachineSwitch)}
50	TextFileInfo[0]	
51	TextFileInfo[21]	
52	hnSegLastZ[1]	{IF(hn2_4zoneMachineSwitch=0, 4, hn2_4zoneMachineSwitch)}
53	TextFileInfo[0]	
54	TextFileInfo[27]	
55	TextFileInfo[32]	
56	REC[0].DwellTimeSP	20
57	TextFileInfo[33]	
58	REC[20].DwellTimeSP	20
59	TextFileInfo[34]	
60	REC[40].DwellTimeSP	{IF(hn2_4zoneMachineSwitch=4, 20, IF(hn2_4zoneMachineSwitch=0, 20, ...
61	TextFileInfo[35]	
62	REC[60].DwellTimeSP	{IF(hn2_4zoneMachineSwitch=4, 20, IF(hn2_4zoneMachineSwitch=0, 20, ...

Recipe1.rcp

Description:

Options:
☐ Save As XML
☒ Unicode

File Name: Register Number:

	Tag Name	Number of Elements
63	TextFileInfo[28]	
64	TextFileInfo[32]	
65	REC[0].TemperSP	20
66	TextFileInfo[33]	
67	REC[20].TemperSP	20
68	TextFileInfo[34]	
69	REC[40].TemperSP	{IF(hn2_4zoneMachineSwitch=4, 20, IF(hn2_4zoneMachineSwitch=0, 20, ...
70	TextFileInfo[35]	
71	REC[60].TemperSP	{IF(hn2_4zoneMachineSwitch=4, 20, IF(hn2_4zoneMachineSwitch=0, 20, ...
72	TextFileInfo[29]	
73	TextFileInfo[32]	
74	REC[0].RAMPYN	20
75	TextFileInfo[33]	
76	REC[20].RAMPYN	20
77	TextFileInfo[34]	
78	REC[40].RAMPYN	{IF(hn2_4zoneMachineSwitch=4, 20, IF(hn2_4zoneMachineSwitch=0, 20, ...
79	TextFileInfo[35]	
80	REC[60].RAMPYN	{IF(hn2_4zoneMachineSwitch=4, 20, IF(hn2_4zoneMachineSwitch=0, 20, ...
81	TextFileInfo[30]	
82	TextFileInfo[32]	
83	REC[0].RampRate	20
84	TextFileInfo[33]	
85	REC[20].RampRate	20
86	TextFileInfo[34]	
87	REC[40].RampRate	{IF(hn2_4zoneMachineSwitch=4, 20, IF(hn2_4zoneMachineSwitch=0, 20, ...
88	TextFileInfo[35]	
89	REC[60].RampRate	{IF(hn2_4zoneMachineSwitch=4, 20, IF(hn2_4zoneMachineSwitch=0, 20, ...
90	TextFileInfo[31]	
91	TextFileInfo[32]	
92	REC[0].Comment	20
93	TextFileInfo[33]

Recipe1.rcp

Description:

File Name: Register Number:

Options:
☐ Save As XML
☒ Unicode

	Tag Name	Number of Elements
94	REC[20].Comment	20
95	TextFileInfo[34]	
96	REC[40].Comment	{IF(hn2_4zoneMachineSwitch=4, 20, IF(hn2_4zoneMachineSwitch=0, 20, ...
97	TextFileInfo[35]	
98	REC[60].Comment	{IF(hn2_4zoneMachineSwitch=4, 20, IF(hn2_4zoneMachineSwitch=0, 20, ...
99	TextFileInfo[0]	
100	TextFileInfo[36]	
101	CycleElapsedTimeEst	
102	TextFileInfo[40]	
*		
*		

Workspace Toolbar: Tasks Tab>Reports

Output file: {RecipesFolderPath}\RecipeTextFiles\{RecipeName}_{year}-{month}-{day}.rtf

The report is configured by pressing on the 'Edit RTF file' which opens the Report*.rtf, where * is the same number as that shown in the 'Report*.rep' window. The RTF file is contained in directory \<App Dir>\Config\; this file is not to be edited outside of the InduSoft interface, as it seems that InduSoft has bundled an editor with its software. This directory also contains files 'Report1.rtf' file used for layout 4zone.rtf and 'Report2.rtf' file used for layout 2zone.rtf. These files were created by the designer and may be edited with a rich text format editor, such as Microsoft WordPad. It was found that editing this file, then cutting and pasting to the InduSoft file provided the most flexibility.

Note to developer: configure the layout of the RTF file here, then cut and paste into the Report1.rtf (4zone) config file in IWS. JAS 2009

This file contains the text output of the parameters saved in a Recipe File used on the Mold Heating and Cooling (MHC) Pump Package Operator Interface (MHC PP OI) for plastic molded parts.

This document is for information only. The recipe file saved on the computer/controller is the governing data.

Units are degrees Fahrenheit for temperatures, and seconds for time, unless otherwise specified.

PLC Software: {TextFileInfo[2]}

HMI Software: {TextFileInfo[3]}

Date Created: {date} {time}

Recipe Name: {RecipeName}

Part Numbers: {RecipePartNumber}

Recipe Info: {RecipeHeader}

Estimated Cycle Time: {CycleElapsedTimeEst}

Two or Four Zone Machine: {hn2_4zoneMachineSwitch}

Type of Preheat (0=AutoPreheat, 1=ManualPreheat): {hfManualPreheat}

	Zone 1	Zone 2	Zone 3	Zone 4
Control Mode	{hnControlModeZ[1]}	{hnControlModeZ[2]}	{hnControlModeZ[3]}	{hnControlModeZ[4]}

(1=FluidConstantTemper, 2=FluidRecipe, 3=MoldConstantTemper, 4=MoldRecipe)

Mold Dead Time Const, Theta	{REC_MTL[1].Theta}	{REC_MTL[2].Theta}	{REC_MTL[3].Theta}	{REC_MTL[4].Theta}
Mold Time Const, Tau	{REC_MTL[1].Tau}	{REC_MTL[2].Tau}	{REC_MTL[3].Tau}	{REC_MTL[4].Tau}
Mold Closed Loop Time Const, Lambda	{REC_MTL[1].Lambda}	{REC_MTL[2].Lambda}	{REC_MTL[3].Lambda}	{REC_MTL[4].Lambda}
Mold Entered Gain, K'	{REC_MTL[1].Kprime 3}	{REC_MTL[2].Kprime 3}	{REC_MTL[3].Kprime 3}	{REC_MTL[4].Kprime 3}
Mold Steady State Gain, Km	{REC_MTL[1].Km 3}	{REC_MTL[2].Km 3}	{REC_MTL[3].Km 3}	{REC_MTL[4].Km 3}

Proportional Gain, Kp	{REC_PID[1].Kp 2}	{REC_PID[2].Kp 2}	{REC_PID[3].Kp 2}	{REC_PID[4].Kp 2}
------------------------------	-------------------	-------------------	-------------------	-------------------

Integral Gain, Ki	{REC_PID [1].Ki 2}	{REC_PID [2].Ki 2}	{REC_PID [3].Ki 2}	{REC_PID [4].Ki 2}
Derivative Gain, Kd	{REC_PID [1].Kd 2}	{REC_PID [2].Kd 2}	{REC_PID [3].Kd 2}	{REC_PID [4].Kd 2}

If the zone is configured as **Constant Temperature**, this is the temperature set point:

Const Temper SP	{hnSPTempe rConstZ[1]}	{hnSPTempe rConstZ[2]}	{hnSPTempe rConstZ[3]}	{hnSPTempe rConstZ[4]}
------------------------	---------------------------	---------------------------	---------------------------	---------------------------

Has a **Cycle Abort Temperature** been specified? (1=yes, 0=using last defined segment set point)

Yes/No (1/0)	{hfSPTempe rCycleAbort Z[1]}	{hfSPTempe rCycleAbort Z[2]}	{hfSPTempe rCycleAbort Z[3]}	{hfSPTempe rCycleAbort Z[4]}
Cycle Abort SP	{hnSPTempe rCycleAbort Z[1]}	{hnSPTempe rCycleAbort Z[2]}	{hnSPTempe rCycleAbort Z[3]}	{hnSPTempe rCycleAbort Z[4]}

Using **Mold Temperature Limits**? (used to hold in Segment 1 until within limits)(1=yes, 0=no)

Yes/No (1/0)	{hfMoldTem perLimitsZ [1]}	{hfMoldTem perLimitsZ [2]}	{hfMoldTem perLimitsZ [3]}	{hfMoldTem perLimitsZ [4]}
Upper Limit	{hnMoldTem perLimitUpp erZ[1]}	{hnMoldTem perLimitUpp erZ[2]}	{hnMoldTem perLimitUpp erZ[3]}	{hnMoldTem perLimitUpp erZ[4]}
Lower Limit	{hnMoldTem perLimitLow erZ[1]}	{hnMoldTem perLimitLow erZ[2]}	{hnMoldTem perLimitLow erZ[3]}	{hnMoldTem perLimitLow erZ[4]}

The last defined segment for each zone (19 is max):

Last Segment	{hnSegLastZ [1]}	{hnSegLastZ [2]}	{hnSegLastZ [3]}	{hnSegLastZ [4]}
---------------------	---------------------	---------------------	---------------------	---------------------

Segment Time Set Points:

	Zone 1	Zone 2	Zone 3	Zone 4
Segment 0	{REC [0].DwellTim eSP}	{REC [20].DwellTi meSP}	{REC [40].DwellTi meSP}	{REC [60].DwellTi meSP}

Segment 1	{REC [1].DwellTim eSP}	{REC [21].DwellTi meSP}	{REC [41].DwellTi meSP}	{REC [61].DwellTi meSP}
Segment 2	{REC [2].DwellTim eSP}	{REC [22].DwellTi meSP}	{REC [42].DwellTi meSP}	{REC [62].DwellTi meSP}
Segment 3	{REC [3].DwellTim eSP}	{REC [23].DwellTi meSP}	{REC [43].DwellTi meSP}	{REC [63].DwellTi meSP}
Segment 4	{REC [4].DwellTim eSP}	{REC [24].DwellTi meSP}	{REC [44].DwellTi meSP}	{REC [64].DwellTi meSP}
Segment 5	{REC [5].DwellTim eSP}	{REC [25].DwellTi meSP}	{REC [45].DwellTi meSP}	{REC [65].DwellTi meSP}
Segment 6	{REC [6].DwellTim eSP}	{REC [26].DwellTi meSP}	{REC [46].DwellTi meSP}	{REC [66].DwellTi meSP}
Segment 7	{REC [7].DwellTim eSP}	{REC [27].DwellTi meSP}	{REC [47].DwellTi meSP}	{REC [67].DwellTi meSP}
Segment 8	{REC [8].DwellTim eSP}	{REC [28].DwellTi meSP}	{REC [48].DwellTi meSP}	{REC [68].DwellTi meSP}
Segment 9	{REC [9].DwellTim eSP}	{REC [29].DwellTi meSP}	{REC [49].DwellTi meSP}	{REC [69].DwellTi meSP}
Segment 10	{REC [10].DwellTi meSP}	{REC [30].DwellTi meSP}	{REC [50].DwellTi meSP}	{REC [70].DwellTi meSP}
Segment 11	{REC [11].DwellTi meSP}	{REC [31].DwellTi meSP}	{REC [51].DwellTi meSP}	{REC [71].DwellTi meSP}
Segment 12	{REC [12].DwellTi meSP}	{REC [32].DwellTi meSP}	{REC [52].DwellTi meSP}	{REC [72].DwellTi meSP}
Segment 13	{REC [13].DwellTi meSP}	{REC [33].DwellTi meSP}	{REC [53].DwellTi meSP}	{REC [73].DwellTi meSP}
Segment 14	{REC [14].DwellTi meSP}	{REC [34].DwellTi meSP}	{REC [54].DwellTi meSP}	{REC [74].DwellTi meSP}

Segment 15	{REC [15].DwellTi meSP}	{REC [35].DwellTi meSP}	{REC [55].DwellTi meSP}	{REC [75].DwellTi meSP}
Segment 16	{REC [16].DwellTi meSP}	{REC [36].DwellTi meSP}	{REC [56].DwellTi meSP}	{REC [76].DwellTi meSP}
Segment 17	{REC [17].DwellTi meSP}	{REC [37].DwellTi meSP}	{REC [57].DwellTi meSP}	{REC [77].DwellTi meSP}
Segment 18	{REC [18].DwellTi meSP}	{REC [38].DwellTi meSP}	{REC [58].DwellTi meSP}	{REC [78].DwellTi meSP}
Segment 19	{REC [19].DwellTi meSP}	{REC [39].DwellTi meSP}	{REC [59].DwellTi meSP}	{REC [79].DwellTi meSP}

Segment Temperature Set Points:

	Zone 1	Zone 2	Zone 3	Zone 4
Segment 0	{REC [0].TemperS P}	{REC [20].Temper SP}	{REC [40].Temper SP}	{REC [60].Temper SP}
Segment 1	{REC [1].TemperS P}	{REC [21].Temper SP}	{REC [41].Temper SP}	{REC [61].Temper SP}
Segment 2	{REC [2].TemperS P}	{REC [22].Temper SP}	{REC [42].Temper SP}	{REC [62].Temper SP}
Segment 3	{REC [3].TemperS P}	{REC [23].Temper SP}	{REC [43].Temper SP}	{REC [63].Temper SP}
Segment 4	{REC [4].TemperS P}	{REC [24].Temper SP}	{REC [44].Temper SP}	{REC [64].Temper SP}
Segment 5	{REC [5].TemperS P}	{REC [25].Temper SP}	{REC [45].Temper SP}	{REC [65].Temper SP}
Segment 6	{REC [6].TemperS P}	{REC [26].Temper SP}	{REC [46].Temper SP}	{REC [66].Temper SP}

Segment 7	{REC [7].TemperS P}	{REC [27].Temper SP}	{REC [47].Temper SP}	{REC [67].Temper SP}
Segment 8	{REC [8].TemperS P}	{REC [28].Temper SP}	{REC [48].Temper SP}	{REC [68].Temper SP}
Segment 9	{REC [9].TemperS P}	{REC [29].Temper SP}	{REC [49].Temper SP}	{REC [69].Temper SP}
Segment 10	{REC [10].Temper SP}	{REC [30].Temper SP}	{REC [50].Temper SP}	{REC [70].Temper SP}
Segment 11	{REC [11].Temper SP}	{REC [31].Temper SP}	{REC [51].Temper SP}	{REC [71].Temper SP}
Segment 12	{REC [12].Temper SP}	{REC [32].Temper SP}	{REC [52].Temper SP}	{REC [72].Temper SP}
Segment 13	{REC [13].Temper SP}	{REC [33].Temper SP}	{REC [53].Temper SP}	{REC [73].Temper SP}
Segment 14	{REC [14].Temper SP}	{REC [34].Temper SP}	{REC [54].Temper SP}	{REC [74].Temper SP}
Segment 15	{REC [15].Temper SP}	{REC [35].Temper SP}	{REC [55].Temper SP}	{REC [75].Temper SP}
Segment 16	{REC [16].Temper SP}	{REC [36].Temper SP}	{REC [56].Temper SP}	{REC [76].Temper SP}
Segment 17	{REC [17].Temper SP}	{REC [37].Temper SP}	{REC [57].Temper SP}	{REC [77].Temper SP}
Segment 18	{REC [18].Temper SP}	{REC [38].Temper SP}	{REC [58].Temper SP}	{REC [78].Temper SP}
Segment 19	{REC [19].Temper SP}	{REC [39].Temper SP}	{REC [59].Temper SP}	{REC [79].Temper SP}

Was Ramp (*Yes/No*) selected for the Segment Set Points:

<i>(1=Yes, 0=No)</i>	Zone 1	Zone 2	Zone 3	Zone 4
Segment 0	{REC [0].RampYN}	{REC [20].RampY N}	{REC [40].RampY N}	{REC [60].RampY N}
Segment 1	{REC [1].RampYN}	{REC [21].RampYN }	{REC [41].RampYN }	{REC [61].RampYN }
Segment 2	{REC [2].RampYN}	{REC [22].RampY N}	{REC [42].RampY N}	{REC [62].RampY N}
Segment 3	{REC [3].RampYN}	{REC [23].RampY N}	{REC [43].RampY N}	{REC [63].RampY N}
Segment 4	{REC [4].RampYN}	{REC [24].RampY N}	{REC [44].RampY N}	{REC [64].RampY N}
Segment 5	{REC [5].RampYN}	{REC [25].RampY N}	{REC [45].RampY N}	{REC [65].RampY N}
Segment 6	{REC [6].RampYN}	{REC [26].RampY N}	{REC [46].RampY N}	{REC [66].RampY N}
Segment 7	{REC [7].RampYN}	{REC [27].RampY N}	{REC [47].RampY N}	{REC [67].RampY N}
Segment 8	{REC [8].RampYN}	{REC [28].RampY N}	{REC [48].RampY N}	{REC [68].RampY N}
Segment 9	{REC [9].RampYN}	{REC [29].RampY N}	{REC [49].RampY N}	{REC [69].RampY N}
Segment 10	{REC [10].RampYN }	{REC [30].RampY N}	{REC [50].RampY N}	{REC [70].RampY N}
Segment 11	{REC [11].RampYN }	{REC [31].RampYN }	{REC [51].RampYN }	{REC [71].RampYN }
Segment 12	{REC [12].RampYN }	{REC [32].RampY N}	{REC [52].RampY N}	{REC [72].RampY N}

Segment 13	{REC [13].RampYN }	{REC [33].RampY N}	{REC [53].RampY N}	{REC [73].RampY N}
Segment 14	{REC [14].RampYN }	{REC [34].RampY N}	{REC [54].RampY N}	{REC [74].RampY N}
Segment 15	{REC [15].RampYN }	{REC [35].RampY N}	{REC [55].RampY N}	{REC [75].RampY N}
Segment 16	{REC [16].RampYN }	{REC [36].RampY N}	{REC [56].RampY N}	{REC [76].RampY N}
Segment 17	{REC [17].RampYN }	{REC [37].RampY N}	{REC [57].RampY N}	{REC [77].RampY N}
Segment 18	{REC [18].RampYN }	{REC [38].RampY N}	{REC [58].RampY N}	{REC [78].RampY N}
Segment 19	{REC [19].RampYN }	{REC [39].RampY N}	{REC [59].RampY N}	{REC [79].RampY N}

The calculate **Ramp Rate** for the Segment Set Points:

	Zone 1	Zone 2	Zone 3	Zone 4
Segment 0	{REC [0].RampRate 3}	{REC [20].RampRate 3}	{REC [40].RampRate 3}	{REC [60].RampRate 3}
Segment 1	{REC [1].RampRate 3}	{REC [21].RampRate 3}	{REC [41].RampRate 3}	{REC [61].RampRate 3}
Segment 2	{REC [2].RampRate 3}	{REC [22].RampRate 3}	{REC [42].RampRate 3}	{REC [62].RampRate 3}
Segment 3	{REC [3].RampRate 3}	{REC [23].RampRate 3}	{REC [43].RampRate 3}	{REC [63].RampRate 3}
Segment 4	{REC [4].RampRate 3}	{REC [24].RampRate 3}	{REC [44].RampRate 3}	{REC [64].RampRate 3}

Segment 5	{REC [5].RampRate 3}	{REC [25].RampRate 3}	{REC [45].RampRate 3}	{REC [65].RampRate 3}
Segment 6	{REC [6].RampRate 3}	{REC [26].RampRate 3}	{REC [46].RampRate 3}	{REC [66].RampRate 3}
Segment 7	{REC [7].RampRate 3}	{REC [27].RampRate 3}	{REC [47].RampRate 3}	{REC [67].RampRate 3}
Segment 8	{REC [8].RampRate 3}	{REC [28].RampRate 3}	{REC [48].RampRate 3}	{REC [68].RampRate 3}
Segment 9	{REC [9].RampRate 3}	{REC [29].RampRate 3}	{REC [49].RampRate 3}	{REC [69].RampRate 3}
Segment 10	{REC [10].RampRate 3}	{REC [30].RampRate 3}	{REC [50].RampRate 3}	{REC [70].RampRate 3}
Segment 11	{REC [11].RampRate 3}	{REC [31].RampRate 3}	{REC [51].RampRate 3}	{REC [71].RampRate 3}
Segment 12	{REC [12].RampRate 3}	{REC [32].RampRate 3}	{REC [52].RampRate 3}	{REC [72].RampRate 3}
Segment 13	{REC [13].RampRate 3}	{REC [33].RampRate 3}	{REC [53].RampRate 3}	{REC [73].RampRate 3}
Segment 14	{REC [14].RampRate 3}	{REC [34].RampRate 3}	{REC [54].RampRate 3}	{REC [74].RampRate 3}
Segment 15	{REC [15].RampRate 3}	{REC [35].RampRate 3}	{REC [55].RampRate 3}	{REC [75].RampRate 3}
Segment 16	{REC [16].RampRate 3}	{REC [36].RampRate 3}	{REC [56].RampRate 3}	{REC [76].RampRate 3}
Segment 17	{REC [17].RampRate 3}	{REC [37].RampRate 3}	{REC [57].RampRate 3}	{REC [77].RampRate 3}
Segment 18	{REC [18].RampRate 3}	{REC [38].RampRate 3}	{REC [58].RampRate 3}	{REC [78].RampRate 3}

Segment 19	{REC [19].RampRa te 3}	{REC [39].RampRa te 3}	{REC [59].RampRa te 3}	{REC [79].RampRa te 3}
-------------------	------------------------------	------------------------------	------------------------------	------------------------------

Comments for the Segment Set Points:

Zone 1, Segment 0: {REC[0].Comment}
 Zone 1, Segment 1: {REC[1].Comment}
 Zone 1, Segment 2: {REC[2].Comment}
 Zone 1, Segment 3: {REC[3].Comment}
 Zone 1, Segment 4: {REC[4].Comment}
 Zone 1, Segment 5: {REC[5].Comment}
 Zone 1, Segment 6: {REC[6].Comment}
 Zone 1, Segment 7: {REC[7].Comment}
 Zone 1, Segment 8: {REC[8].Comment}
 Zone 1, Segment 9: {REC[9].Comment}
 Zone 1, Segment 10: {REC[10].Comment}
 Zone 1, Segment 11: {REC[11].Comment}
 Zone 1, Segment 12: {REC[12].Comment}
 Zone 1, Segment 13: {REC[13].Comment}
 Zone 1, Segment 14: {REC[14].Comment}
 Zone 1, Segment 15: {REC[15].Comment}
 Zone 1, Segment 16: {REC[16].Comment}
 Zone 1, Segment 17: {REC[17].Comment}
 Zone 1, Segment 18: {REC[18].Comment}
 Zone 1, Segment 19: {REC[19].Comment}

Zone 2, Segment 0: {REC[20].Comment}
 Zone 2, Segment 1: {REC[21].Comment}
 Zone 2, Segment 2: {REC[22].Comment}
 Zone 2, Segment 3: {REC[23].Comment}
 Zone 2, Segment 4: {REC[24].Comment}
 Zone 2, Segment 5: {REC[25].Comment}
 Zone 2, Segment 6: {REC[26].Comment}
 Zone 2, Segment 7: {REC[27].Comment}
 Zone 2, Segment 8: {REC[28].Comment}
 Zone 2, Segment 9: {REC[29].Comment}
 Zone 2, Segment 10: {REC[30].Comment}
 Zone 2, Segment 11: {REC[31].Comment}
 Zone 2, Segment 12: {REC[32].Comment}
 Zone 2, Segment 13: {REC[33].Comment}
 Zone 2, Segment 14: {REC[34].Comment}

Zone 2, Segment 15: {REC[35].Comment}
Zone 2, Segment 16: {REC[36].Comment}
Zone 2, Segment 17: {REC[37].Comment}
Zone 2, Segment 18: {REC[38].Comment}
Zone 2, Segment 19: {REC[39].Comment}

Zone 3, Segment 0: {REC[40].Comment}
Zone 3, Segment 1: {REC[41].Comment}
Zone 3, Segment 2: {REC[42].Comment}
Zone 3, Segment 3: {REC[43].Comment}
Zone 3, Segment 4: {REC[44].Comment}
Zone 3, Segment 5: {REC[45].Comment}
Zone 3, Segment 6: {REC[46].Comment}
Zone 3, Segment 7: {REC[47].Comment}
Zone 3, Segment 8: {REC[48].Comment}
Zone 3, Segment 9: {REC[49].Comment}
Zone 3, Segment 10: {REC[50].Comment}
Zone 3, Segment 11: {REC[51].Comment}
Zone 3, Segment 12: {REC[52].Comment}
Zone 3, Segment 13: {REC[53].Comment}
Zone 3, Segment 14: {REC[54].Comment}
Zone 3, Segment 15: {REC[55].Comment}
Zone 3, Segment 16: {REC[56].Comment}
Zone 3, Segment 17: {REC[57].Comment}
Zone 3, Segment 18: {REC[58].Comment}
Zone 3, Segment 19: {REC[59].Comment}

Zone 4, Segment 0: {REC[60].Comment}
Zone 4, Segment 1: {REC[61].Comment}
Zone 4, Segment 2: {REC[62].Comment}
Zone 4, Segment 3: {REC[63].Comment}
Zone 4, Segment 4: {REC[64].Comment}
Zone 4, Segment 5: {REC[65].Comment}
Zone 4, Segment 6: {REC[66].Comment}
Zone 4, Segment 7: {REC[67].Comment}
Zone 4, Segment 8: {REC[68].Comment}
Zone 4, Segment 9: {REC[69].Comment}
Zone 4, Segment 10: {REC[70].Comment}
Zone 4, Segment 11: {REC[71].Comment}
Zone 4, Segment 12: {REC[72].Comment}
Zone 4, Segment 13: {REC[73].Comment}
Zone 4, Segment 14: {REC[74].Comment}

Zone 4, Segment 15: {REC[75].Comment}

Zone 4, Segment 16: {REC[76].Comment}

Zone 4, Segment 17: {REC[77].Comment}

Zone 4, Segment 18: {REC[78].Comment}

Zone 4, Segment 19: {REC[79].Comment}

End of File.

Note to developer: configure the layout of the RTF file here, then cut and paste into the Report2.rtf (2zone) config file in IWS. JAS 2009

This file contains the text output of the parameters saved in a Recipe File used on the Mold Heating and Cooling (MHC) Pump Package Operator Interface (MHC PP OI) for plastic molded parts.

This document is for information only. The recipe file saved on the computer/controller is the governing data.

Units are degrees Fahrenheit for temperatures, and seconds for time, unless otherwise specified.

PLC Software: {TextFileInfo[2]}

HMI Software: {TextFileInfo[3]}

Date Created: {date} {time}

Recipe Name: {RecipeName}

Part Numbers: {RecipePartNumber}

Recipe Info: {RecipeHeader}

Estimated Cycle Time: {CycleElapsedTimeEst}

Two or Four Zone Machine: {hn2_4zoneMachineSwitch}

Type of Preheat (0=AutoPreheat, 1=ManualPreheat): {hfManualPreheat}

	Zone 1	Zone 2		
Control Mode	{hnControlModeZ[1]}	{hnControlModeZ[2]}		

(1=FluidConstantTemper, 2=FluidRecipe, 3=MoldConstantTemper, 4=MoldRecipe)

Mold Dead Time Const, Theta	{REC_MTL[1].Theta}	{REC_MTL[2].Theta}		
Mold Time Const, Tau	{REC_MTL[1].Tau}	{REC_MTL[2].Tau}		
Mold Closed Loop Time Const, Lambda	{REC_MTL[1].Lambda}	{REC_MTL[2].Lambda}		
Mold Entered Gain, K'	{REC_MTL[1].Kprime 3}	{REC_MTL[2].Kprime 3}		
Mold Steady State Gain, Km	{REC_MTL[1].Km 3}	{REC_MTL[2].Km 3}		

Proportional Gain, Kp	{REC_PID[1].Kp 2}	{REC_PID[2].Kp 2}		
------------------------------	-------------------	-------------------	--	--

Integral Gain, Ki	{REC_PID [1].Ki 2}	{REC_PID [2].Ki 2}		
Derivative Gain, Kd	{REC_PID [1].Kd 2}	{REC_PID [2].Kd 2}		

If the zone is configured as **Constant Temperature**, this is the temperature set point:

Const Temper SP	{hnSPTempe rConstZ[1]}	{hnSPTempe rConstZ[2]}		
------------------------	---------------------------	---------------------------	--	--

Has a **Cycle Abort Temperature** been specified? (1=yes, 0=using last defined segment set point)

Yes/No (1/0)	{hfSPTempe rCycleAbort Z[1]}	{hfSPTempe rCycleAbort Z[2]}		
Cycle Abort SP	{hnSPTempe rCycleAbort Z[1]}	{hnSPTempe rCycleAbort Z[2]}		

Using **Mold Temperature Limits**? (used to hold in Segment 1 until within limits)(1=yes, 0=no)

Yes/No (1/0)	{hfMoldTem perLimitsZ [1]}	{hfMoldTem perLimitsZ [2]}		
Upper Limit	{hnMoldTem perLimitUpp erZ[1]}	{hnMoldTem perLimitUpp erZ[2]}		
Lower Limit	{hnMoldTem perLimitLow erZ[1]}	{hnMoldTem perLimitLow erZ[2]}		

The last defined segment for each zone (19 is max):

Last Segment	{hnSegLastZ [1]}	{hnSegLastZ [2]}		
---------------------	---------------------	---------------------	--	--

Segment Time Set Points:

	Zone 1	Zone 2		
Segment 0	{REC [0].DwellTim eSP}	{REC [20].DwellTi meSP}		

Segment 1	{REC [1].DwellTimeSP}	{REC [21].DwellTimeSP}		
Segment 2	{REC [2].DwellTimeSP}	{REC [22].DwellTimeSP}		
Segment 3	{REC [3].DwellTimeSP}	{REC [23].DwellTimeSP}		
Segment 4	{REC [4].DwellTimeSP}	{REC [24].DwellTimeSP}		
Segment 5	{REC [5].DwellTimeSP}	{REC [25].DwellTimeSP}		
Segment 6	{REC [6].DwellTimeSP}	{REC [26].DwellTimeSP}		
Segment 7	{REC [7].DwellTimeSP}	{REC [27].DwellTimeSP}		
Segment 8	{REC [8].DwellTimeSP}	{REC [28].DwellTimeSP}		
Segment 9	{REC [9].DwellTimeSP}	{REC [29].DwellTimeSP}		
Segment 10	{REC [10].DwellTimeSP}	{REC [30].DwellTimeSP}		
Segment 11	{REC [11].DwellTimeSP}	{REC [31].DwellTimeSP}		
Segment 12	{REC [12].DwellTimeSP}	{REC [32].DwellTimeSP}		
Segment 13	{REC [13].DwellTimeSP}	{REC [33].DwellTimeSP}		
Segment 14	{REC [14].DwellTimeSP}	{REC [34].DwellTimeSP}		

Segment 15	{REC [15].DwellTi meSP}	{REC [35].DwellTi meSP}		
Segment 16	{REC [16].DwellTi meSP}	{REC [36].DwellTi meSP}		
Segment 17	{REC [17].DwellTi meSP}	{REC [37].DwellTi meSP}		
Segment 18	{REC [18].DwellTi meSP}	{REC [38].DwellTi meSP}		
Segment 19	{REC [19].DwellTi meSP}	{REC [39].DwellTi meSP}		

Segment Temperature Set Points:

	Zone 1	Zone 2		
Segment 0	{REC [0].TemperS P}	{REC [20].Temper SP}		
Segment 1	{REC [1].TemperS P}	{REC [21].Temper SP}		
Segment 2	{REC [2].TemperS P}	{REC [22].Temper SP}		
Segment 3	{REC [3].TemperS P}	{REC [23].Temper SP}		
Segment 4	{REC [4].TemperS P}	{REC [24].Temper SP}		
Segment 5	{REC [5].TemperS P}	{REC [25].Temper SP}		
Segment 6	{REC [6].TemperS P}	{REC [26].Temper SP}		

Segment 7	{REC [7].TemperS P}	{REC [27].Temper SP}		
Segment 8	{REC [8].TemperS P}	{REC [28].Temper SP}		
Segment 9	{REC [9].TemperS P}	{REC [29].Temper SP}		
Segment 10	{REC [10].Temper SP}	{REC [30].Temper SP}		
Segment 11	{REC [11].Temper SP}	{REC [31].Temper SP}		
Segment 12	{REC [12].Temper SP}	{REC [32].Temper SP}		
Segment 13	{REC [13].Temper SP}	{REC [33].Temper SP}		
Segment 14	{REC [14].Temper SP}	{REC [34].Temper SP}		
Segment 15	{REC [15].Temper SP}	{REC [35].Temper SP}		
Segment 16	{REC [16].Temper SP}	{REC [36].Temper SP}		
Segment 17	{REC [17].Temper SP}	{REC [37].Temper SP}		
Segment 18	{REC [18].Temper SP}	{REC [38].Temper SP}		
Segment 19	{REC [19].Temper SP}	{REC [39].Temper SP}		

Was Ramp (*Yes/No*) selected for the Segment Set Points:

<i>(1=Yes, 0=No)</i>	Zone 1	Zone 2		
Segment 0	{REC [0].RampYN}	{REC [20].RampY N}		
Segment 1	{REC [1].RampYN}	{REC [21].RampYN }		
Segment 2	{REC [2].RampYN}	{REC [22].RampY N}		
Segment 3	{REC [3].RampYN}	{REC [23].RampY N}		
Segment 4	{REC [4].RampYN}	{REC [24].RampY N}		
Segment 5	{REC [5].RampYN}	{REC [25].RampY N}		
Segment 6	{REC [6].RampYN}	{REC [26].RampY N}		
Segment 7	{REC [7].RampYN}	{REC [27].RampY N}		
Segment 8	{REC [8].RampYN}	{REC [28].RampY N}		
Segment 9	{REC [9].RampYN}	{REC [29].RampY N}		
Segment 10	{REC [10].RampYN }	{REC [30].RampY N}		
Segment 11	{REC [11].RampYN }	{REC [31].RampYN }		
Segment 12	{REC [12].RampYN }	{REC [32].RampY N}		

Segment 13	{REC [13].RampYN }	{REC [33].RampY N}		
Segment 14	{REC [14].RampYN }	{REC [34].RampY N}		
Segment 15	{REC [15].RampYN }	{REC [35].RampY N}		
Segment 16	{REC [16].RampYN }	{REC [36].RampY N}		
Segment 17	{REC [17].RampYN }	{REC [37].RampY N}		
Segment 18	{REC [18].RampYN }	{REC [38].RampY N}		
Segment 19	{REC [19].RampYN }	{REC [39].RampY N}		

The calculate **Ramp Rate** for the Segment Set Points:

	Zone 1	Zone 2		
Segment 0	{REC [0].RampRate 3}	{REC [20].RampRate 3}		
Segment 1	{REC [1].RampRate 3}	{REC [21].RampRate 3}		
Segment 2	{REC [2].RampRate 3}	{REC [22].RampRate 3}		
Segment 3	{REC [3].RampRate 3}	{REC [23].RampRate 3}		
Segment 4	{REC [4].RampRate 3}	{REC [24].RampRate 3}		

Segment 5	{REC [5].RampRate 3}	{REC [25].RampRate 3}		
Segment 6	{REC [6].RampRate 3}	{REC [26].RampRate 3}		
Segment 7	{REC [7].RampRate 3}	{REC [27].RampRate 3}		
Segment 8	{REC [8].RampRate 3}	{REC [28].RampRate 3}		
Segment 9	{REC [9].RampRate 3}	{REC [29].RampRate 3}		
Segment 10	{REC [10].RampRate 3}	{REC [30].RampRate 3}		
Segment 11	{REC [11].RampRate 3}	{REC [31].RampRate 3}		
Segment 12	{REC [12].RampRate 3}	{REC [32].RampRate 3}		
Segment 13	{REC [13].RampRate 3}	{REC [33].RampRate 3}		
Segment 14	{REC [14].RampRate 3}	{REC [34].RampRate 3}		
Segment 15	{REC [15].RampRate 3}	{REC [35].RampRate 3}		
Segment 16	{REC [16].RampRate 3}	{REC [36].RampRate 3}		
Segment 17	{REC [17].RampRate 3}	{REC [37].RampRate 3}		
Segment 18	{REC [18].RampRate 3}	{REC [38].RampRate 3}		

Segment 19	{REC [19].RampRa te 3}	{REC [39].RampRa te 3}		
-------------------	------------------------------	------------------------------	--	--

Comments for the Segment Set Points:

Zone 1, Segment 0: {REC[0].Comment}
 Zone 1, Segment 1: {REC[1].Comment}
 Zone 1, Segment 2: {REC[2].Comment}
 Zone 1, Segment 3: {REC[3].Comment}
 Zone 1, Segment 4: {REC[4].Comment}
 Zone 1, Segment 5: {REC[5].Comment}
 Zone 1, Segment 6: {REC[6].Comment}
 Zone 1, Segment 7: {REC[7].Comment}
 Zone 1, Segment 8: {REC[8].Comment}
 Zone 1, Segment 9: {REC[9].Comment}
 Zone 1, Segment 10: {REC[10].Comment}
 Zone 1, Segment 11: {REC[11].Comment}
 Zone 1, Segment 12: {REC[12].Comment}
 Zone 1, Segment 13: {REC[13].Comment}
 Zone 1, Segment 14: {REC[14].Comment}
 Zone 1, Segment 15: {REC[15].Comment}
 Zone 1, Segment 16: {REC[16].Comment}
 Zone 1, Segment 17: {REC[17].Comment}
 Zone 1, Segment 18: {REC[18].Comment}
 Zone 1, Segment 19: {REC[19].Comment}

Zone 2, Segment 0: {REC[20].Comment}
 Zone 2, Segment 1: {REC[21].Comment}
 Zone 2, Segment 2: {REC[22].Comment}
 Zone 2, Segment 3: {REC[23].Comment}
 Zone 2, Segment 4: {REC[24].Comment}
 Zone 2, Segment 5: {REC[25].Comment}
 Zone 2, Segment 6: {REC[26].Comment}
 Zone 2, Segment 7: {REC[27].Comment}
 Zone 2, Segment 8: {REC[28].Comment}
 Zone 2, Segment 9: {REC[29].Comment}
 Zone 2, Segment 10: {REC[30].Comment}
 Zone 2, Segment 11: {REC[31].Comment}
 Zone 2, Segment 12: {REC[32].Comment}
 Zone 2, Segment 13: {REC[33].Comment}
 Zone 2, Segment 14: {REC[34].Comment}

Zone 2, Segment 15: {REC[35].Comment}

Zone 2, Segment 16: {REC[36].Comment}

Zone 2, Segment 17: {REC[37].Comment}

Zone 2, Segment 18: {REC[38].Comment}

Zone 2, Segment 19: {REC[39].Comment}

End of File.

Workspace Toolbar: Tasks Tab>Scheduler

SCHED001.SCH

Description:
script triggers

	Event	Trigger	Time	Date	Tag	Expression	Disable
1	Change	Second			PLCHearbeatScan	1 //run script every second, script resets tag	0
2	Clock		00:00:10		PLCHearbeatScan	1 //run script every # seconds, script resets tag	1
3	Clock						
4	Change	Second			hfHMIHeartbeat	\$Toggle(hfHMIHeartbeat) //OFF for 1 second, ON for 1 second	1
5	Clock		00:00:10		hfHMIHeartbeat	\$Toggle(hfHMIHeartbeat) //OFF for # seconds, ON for # seconds; to reduce network traffic	0
6	Clock						
7	Change	Second			SecondToggle	\$Toggle(SecondToggle) //OFF for 1 second, ON for 1 second	0
8	Clock						
9	Clock		00:00:10		TrendUpdate	\$Toggle(TrendUpdate) //OFF for # seconds, ON for # seconds	0
10	Clock		00:00:10		TrendGraphUpdate	1 //run script every # seconds, script resets tag	0
11	Clock						
12	Change	Day			DailyRunScript	1 //run script every day if app already running, script resets tag	0
*	Clock						

Notice that some schedules are disabled, where the Disable column is set equal to 1.

SCHED002.SCH

Description:
TC retrieval and conversion

	Event	Trigger	Time	Date	Tag	Expression	Disable
1	Clock					//get TC value from PLC via OPC "Always Scan, TC update" sheet, divide by 10	
2	Clock		00:00:00.5		hrTC_MoldZ[1]	nTC_MoldZ[1]/10	0
3	Clock		00:00:00.5		hrTC_MoldZ[2]	nTC_MoldZ[2]/10	0
4	Clock		00:00:00.5		hrTC_MoldZ[3]	nTC_MoldZ[3]/10	0
5	Clock		00:00:00.5		hrTC_MoldZ[4]	nTC_MoldZ[4]/10	0
6	Clock		00:00:00.5		hrTC_FluidZ[1]	nTC_FluidZ[1]/10	0
7	Clock		00:00:00.5		hrTC_FluidZ[2]	nTC_FluidZ[2]/10	0
8	Clock		00:00:00.5		hrTC_FluidZ[3]	nTC_FluidZ[3]/10	0
9	Clock		00:00:00.5		hrTC_FluidZ[4]	nTC_FluidZ[4]/10	0
*	Clock						

Workspace Toolbar: Tasks Tab>Script

Note that the script files are stored in the \<App Dir>\Config\ directory. The files with “*.is” extensions can be viewed with a text editor. It is not advisable to edit these files, only viewing is recommended.

C:\Josh\PAG\703030\MHC\software\MHC Dev v3\IWS v3.1.0\Config\scriptstartup.gis

```
33: ' to convert to text files
34: Dim HstPath
35: HstPath = $InfoAppHstDir() & "Batch" 'sets string "c:\.....\<folder containing .APP
file>\Hst\Batch"
36: If $FindPath( HstPath ) = 0 Then $DirCreate( HstPath )
37:
38: //\\//\\//\\//\\//\\//\\ make sure MTL Tuning Results Folder is setup //\\//\\//\\//\\//\\//\\
39: 'This folder stores the resultant MTL dynamic constants found via the MTL tuning routine
40: $MTRPath = $RecipesFolderPath & "\MTL_TuneResults" 'sets string "c:\.....\<>\Recipes\MTL_TuneResults"
41: If $FindPath( $MTRPath ) = 0 Then $DirCreate( $MTRPath )
42:
43: //\\//\\//\\//\\//\\//\\ make sure Folder is setup where Recipe Reports are output to
//\\//\\//\\//\\//\\//\\
44: '/REV 3.1.0/ newly created
45: Dim rtfPath
46: rtfPath = $RecipesFolderPath & "\RecipeTextFiles"
47: If $FindPath( rtfPath ) = 0 Then $DirCreate( rtfPath )
48:
49: //////////////////////////////////////
////////////////////////////////////
50:
51:
52: ////////////////////////////////////// Daily Stuff
////////////////////////////////////
53: 'moved $DailyRunScript = 1 to Graphics Script, cause Startup Script was holding up the opening of the app
54: ' and this gives OPC time to start communicating, before the Graphics Tasks start
55:
56: '!? use function $DeleteOlderFiles(__,__ ) to delete files older than a specified time period, maybe
couple years old?
57: 'shouldn't have to worry about disc space, so just leave this in here as a reminder
58: //////////////////////////////////////
////////////////////////////////////
59:
60:
61: $Logon( "Operator" )
62:
63: 'End Startup Script
```

```
1: 'The code configured here is executed while the condition configured in the Execution field is TRUE.
2: '/REV 3.1.0/ moved Recipe manipulation (Open, Save, New, etc) to Global Procedures; kept Segment manipulation
   here
3:
4: 'Variables available only for this group can be declared here.
5: Dim i
6:
7: '//This script is broken out into sections.  A particular section will be executed based on the value of
   string tag $RecConfirmDlg.
8: '//The sections were not made into seperate routines that could be placed in the Global Procedures (as others
   were moved 12/2008 /REV 3.1.0/), because
9: '// there was no advantage, because am only doing tag manipulation for Segment Insert and Delete operations in
   a one-shot routine and
10: '// do not need any operator input in the form of msgbox or keypad popups, am only using my IWS custom screens,
    which always have focus.
11:
12: //////////////////////////////////////
   //////////////////////////////////////
13: //////////////////////////////////////          Delete Segment
   //////////////////////////////////////
14: //////////////////////////////////////
   //////////////////////////////////////
15: '//Code to handle deleting and shifting recipe segments,
16: '//$RecConfirmDlg set when "Delete" button is pressed on Segments box of profile editor
17: If $RecConfirmDlg = "Delete Segment" Then
18:
19:     '//take selected segment number 0-79 from ( $RecIndex ) from "Segment Operations Popup.scr"
20:     '//then move up all data in the zone group of 20 segments by 1 element to overwrite and effectively delete
   the chosen segment
21:         '$RecIndex is pointer to 80 element array; '$SegmentPagePointer is 0=z1,20=z2,40=z3,60=z4
22:         'only shift info for current 20 element section of array
23:         '//if RecIndex=1, then set (Seg1)=(Seg2's data), 2=3, ..... 18=19
24:         For i = $RecIndex To ($SegmentPagePointer + 18)
25:             $REC[i].DwellTimeSP = $REC[i + 1].DwellTimeSP
26:             $REC[i].TemperSP = $REC[i + 1].TemperSP
27:             $REC[i].RampYN = $REC[i + 1].RampYN
28:             $REC[i].RampRate = $REC[i + 1].RampRate
29:             $REC[i].Comment = $REC[i + 1].Comment
30:         Next
31:         '//initialize last segment in the zone group of 20 elements that the delete occurred
32:         $REC[$SegmentPagePointer + 19].DwellTimeSP = 0
33:         $REC[$SegmentPagePointer + 19].TemperSP = $REC[ ].TemperSP->Min
```

```
34:      $REC[$SegmentPagePointer + 19].RAMPYN = 0
35:      $REC[$SegmentPagePointer + 19].RAMPRATE = 0
36:      $REC[$SegmentPagePointer + 19].Comment = ""
37:
38:      'call Task Script #3 RampRateCalculation
39:      $calculateRampRate = 1
40:
41: End If      'If $RecConfirmDlg = "Delete Segment"
42:
43:
44: //////////////////////////////////////
45: //////////////////////////////////////                      Insert Segment
46: //////////////////////////////////////                      ( see also " Insert Segment Lose Last" )
47: //////////////////////////////////////
48: //////////////////////////////////////
49: //Code to handle inserting and shifting recipe segments,
50: //$RecConfirmDlg set when "Insert" button is pressed on Segments box of profile editor
51: 'receive $RecIndex from "Segment Operations Popup.scr"
52: 'insert a new segment AT the selected $RecIndex value, move current values to index+1
53: If $RecConfirmDlg = "Insert Segment" Then
54:     'check to make sure operator is not trying to insert a new last segment If $RecIndex = 19 | 39 | 59 | 79
55:     If $RecIndex = $SegmentPagePointer + 19 Then      '$RecIndex=0,1,2...79      $SegmentPagePointer=0,20,40,60
56:         $SegmentOpsMessage = 3
57:         $Open("Segment Operations Popup.scr")
58:         $Confirm = 0 'jump out of this script
59:     End If
60:
61:     'check to make sure there is room w/i each zone (group of 20 elements) for a new segment to be inserted,
    otherwise will lose last
62:     If $Confirm And ( $REC[$SegmentPagePointer + 19].DwellTimeSP > 0 ) Then
63:         'display error message that this segment's information will be lost if a new segment must be inserted
64:         'give option to continue or cancel
65:         'if "ok" on dialog then execute "Insert Segment Lose Last" code, otherwise quit and return to editor
66:         $SegmentOpsMessage = 2
67:         $Open("Segment Operations Popup.scr")
68:         $RecConfirmDlg = "Insert Segment Lose Last"
69:         $Confirm = 0 'jump out of this script
70:     End If
```



```

71:
72:   If $Confirm And $RecConfirmDlg = "Insert Segment" Then
73:     'move elements 'set 19=18, then 18=17.....
74:     'take selected segment number 0-79 from ( $RecIndex ) from "Segment Operations Popup.scr"
75:     '$RecIndex is pointer to 80 element array; '$SegmentPagePointer is 0=z1,20=z2,40=z3,60=z4
76:     'only shift info for current 20 element section of array
77:     'the following should be same code as "Insert Segment Lose Last"
78:     For i = ($SegmentPagePointer + 18) To $RecIndex Step -1
79:       $REC[i + 1].DwellTimeSP = $REC[i].DwellTimeSP
80:       $REC[i + 1].TemperSP = $REC[i].TemperSP
81:       $REC[i + 1].RampYN = $REC[i].RampYN
82:       $REC[i + 1].RampRate = $REC[i].RampRate
83:       $REC[i + 1].Comment = $REC[i].Comment
84:     Next
85:     'initialize new inserted segment
86:     $REC[$RecIndex].DwellTimeSP = 0
87:     $REC[$RecIndex].TemperSP = $REC[ ].TemperSP->Min 'set equal to minimum user input
88:     $REC[$RecIndex].RampYN = 0
89:     $REC[$RecIndex].RampRate = 0
90:     $REC[$RecIndex].Comment = " "
91:
92:     'call Task Script #3 RampRateCalculation, in case user cancels the input dwell dialog (cause successful
dwell input will also call RampRate)
93:     $calculateRampRate = 1
94:     'auto popup dwell input so that newly inserted seg's dwell must be input (dwell scr calls
calculateRampRate)
95:     $Open("Input Dwell Dialog.scr")
96:   End If   'If $Confirm And $RecConfirmDlg = "Insert Segment" Then
97:
98: End If   'If $RecConfirmDlg = "Insert Segment"
99:
100:
101: //////////////////////////////////////
////////////////////////////////////
102: //////////////////////////////////////          Insert Segment Lose Last
////////////////////////////////////
103: //////////////////////////////////////          ( see also " Insert Segment" )
////////////////////////////////////
104: //////////////////////////////////////
////////////////////////////////////
105: //Code to handle inserting and shifting recipe segments, $RecConfirmDlg set in above "Insert Segment" code
106: //If Last Segment is already defined, give option to continue or cancel

```

```
107: 'receive $RecIndex from "Segment Operations Dialog.scr"
108: 'insert a new segment AT the selected $RecIndex value, move current values to index+1
109: If $RecConfirmDlg = "Insert Segment Lose Last" And $Confirm Then
110:
111:     'move elements 'set 19=18, then 18=17.....
112:     'take selected segment number 0-79 from ( $RecIndex ) from "Segment Operations Dialog.scr"
113:     '$RecIndex is pointer to 80 element array; '$SegmentPagePointer is 0=z1,20=z2,40=z3,60=z4
114:     'only shift info for current 20 element section of array
115:     'the following should be same code as "Insert Segment"
116:     For i = ($SegmentPagePointer + 18) To $RecIndex Step -1
117:         $REC[i + 1].DwellTimeSP = $REC[i].DwellTimeSP
118:         $REC[i + 1].TemperSP = $REC[i].TemperSP
119:         $REC[i + 1].RampYN = $REC[i].RampYN
120:         $REC[i + 1].RampRate = $REC[i].RampRate
121:         $REC[i + 1].Comment = $REC[i].Comment
122:     Next
123:     'initialize new inserted segment
124:     $REC[$RecIndex].DwellTimeSP = 0
125:     $REC[$RecIndex].TemperSP = $REC[].TemperSP->Min 'set equal to minimum user input
126:     $REC[$RecIndex].RampYN = 0
127:     $REC[$RecIndex].RampRate = 0
128:     $REC[$RecIndex].Comment = " "
129:
130:     'call Task Script #3 RampRateCalculation, in case user cancels the input dwell dialog (cause successful
dwell input will also call RampRate)
131:     $calculateRampRate = 1
132:     'auto popup dwell input so that newly inserted seg's dwell must be input (dwell scr calls
calculateRampRate)
133:     $Open("Input Dwell Dialog.scr")
134:
135: End If 'If $RecConfirmDlg = "Insert Segment Lose Last"
136:
137:
138: //////////////////////////////////////
////////////////////////////////////
139: ////////////////////////////////////// Cleanup
////////////////////////////////////
140: //////////////////////////////////////
////////////////////////////////////
141: If $Confirm Then
142:     $SegmentOpsMessage = 0 'reset
143:     $RecConfirmDlg = "Confirm Script Done"
```

```
144:    $Confirm = 0
145: End If
146:
147: 'End of Script#1: Confirm Dialog
```

```
1: 'The code configured here is executed while the condition configured in the Execution field is TRUE.
2:
3:
4: 'Variables available only for this group can be declared here.
5:   Dim StartTime
6:
7:
8: 'Seems like it takes about 30 seconds for OPC to sync up HMI and PLC, so keep this screen open for
9: ' about that amount of time to try to keep the user from doing anything until OPC is synced.
10: 'The user can also close the Splash screen.
11:
12:
13:   $DoLoopTimer = 0           'reset timer; timer is used on the screen as a bar graph as visual time left
    before screen closes
14:   StartTime = $GetTickCount() 'IWS internal clock in milliseconds
15:   Do 'wait for XX milliseconds or if user closes screen, screen script will set trigger=0 on close
16:     $DoLoopTimer = $Abs( $GetTickCount() - StartTime )
17:     Loop Until ( $DoLoopTimer >= 30000 ) Or ( $TriggerSplashScreen = 0 )
18:
19:   $Close( "Splash.scr" ) 'if not already closed by the user
20:   $DoLoopTimer = 0       'reset timer
21:
22: 'reset execution tag for this script
23: $TriggerSplashScreen = 0
24:
25: 'End Script #2 Splash Screen
```

```
1: 'The code configured here is executed while the condition configured in the Execution field is TRUE.
2: //RampRates are not xfered to PLC. HMI calculates for operator reference. PLC calculates for SP control.
3: 'This script is called from:
4: ' "RECIPE EDITOR.scr": on TEMP input objects
5: ' "RECIPE EDITOR.scr": DWELL input objects call "_inputDwell.scr", which calls this script
6: ' "RECIPE EDITOR.scr": on RampYN buttons
7:
8:
9: 'Variables available only for this group can be declared here.
10: Dim RecIndexPlus1, RecIndexMinus1
11: RecIndexPlus1 = $RecIndex + 1
12: RecIndexMinus1 = $RecIndex - 1
13:
14: //calculation for current segment
15: If $REC[$RecIndex].TemperSP <> 0 And $REC[$RecIndex].DwellTimeSP <> 0 And $REC[$RecIndex].RAMPYN Then
16: $REC[$RecIndex].RampRate = ($REC[$RecIndex].TemperSP -
$REC[RecIndexMinus1].TemperSP)/$REC[$RecIndex].DwellTimeSP
17: Else
18: $REC[$RecIndex].RampYN = 0
19: $REC[$RecIndex].RampRate = 0
20: End If
21:
22: //calculation for next segment, because the next segment's RR is dependent on the current segment's
temperature
23: If $REC[RecIndexPlus1].TemperSP <> 0 And $REC[RecIndexPlus1].DwellTimeSP <> 0 And
$REC[RecIndexPlus1].RAMPYN Then
24: $REC[RecIndexPlus1].RampRate = ($REC[RecIndexPlus1].TemperSP -
$REC[$RecIndex].TemperSP)/$REC[RecIndexPlus1].DwellTimeSP
25: Else
26: $REC[RecIndexPlus1].RampYN = 0
27: $REC[RecIndexPlus1].RampRate = 0
28: End If
29:
30: //always make sure that segment zero is always defined for each zone
31: //zone 1
32: $REC[0].RAMPYN = 0
33: $REC[0].RampRate = 0
34: //zone 2
35: $REC[20].RAMPYN = 0
36: $REC[20].RampRate = 0
37: //zone 3
38: $REC[40].RAMPYN = 0
```

```
39: $REC[40].RampRate = 0
40: //zone 4
41: $REC[60].RAMPYN = 0
42: $REC[60].RampRate = 0
43:
44: 'call global procedure to find last seg, use this to show/hide input fields on RecEditor-Segments screen
45: Call FindSegLastZx( )
46:     $FindSegLastZxFunc 'marker for IWS Xref tool
47:
48: 'hide Load/Start Recipe button because must save recipe again before starting new part
49: $RecipeIsSaved = 0
50:
51: 'reset execution tag for this script
52: $calculateRampRate = 0
53:
54: 'End Script #3: Ramp Rate Calculation
```

```
1: 'The code configured here is executed while the condition configured in the Execution field is TRUE.
2: /////////// This script is triggered by Scheduler Task #2 ///////////
3:
4:
5: 'Variables available only for this group can be declared here.
6: Dim FailureLimit
7: 'number of times communication comparison failure is allowed before setting failure tag
8: 'in development, 30sec allowed enough time for a momentary loss of comm and recovery,
9: 'typically took ~23sec to recover
10: 'FailureLimit = ( Desired # seconds until admit to lack of comm ) / ( PLCHeartbeatScan time interval in
    seconds, Task2 )
11: FailureLimit = 30 '30/1
12:
13:
14: //////////another option to check that PLC is communicating with OPC is to check the quality of a PLC tag
15: //////////quality= 192 is Good, 64=Uncertain, 0=Bad
16:
17: 'check to see if PLC scan count is continually incrementing, which means PLC is running
18: 'ScanCount updates according to its OPC update rate configuration
19: If $ScanCount = $PreviousScanCount Or $ScanCount->Quality<>192 Then
20: 'increment counter
21: $ScanCountFailure = $ScanCountFailure + 1
22:
23: If $ScanCountFailure >= FailureLimit Then
24: 'set alarm flag that PLC is dead, either comm failure or some other PLC failure
25: $PLCHeartbeatDead = 1
26: 'use this bit to quit other scripts that are running, for example StartRecipe() checks this bit
27: 'set message for "AlarmWarning Popup.scr"
28: $AlarmWarningMessage = 1
29:
30: If $IsScreenOpen( "AlarmWarning Popup.scr" ) Then
31: 'do nothing; this prevents the screen from re-opening everytime
32: Else
33: $Open( "AlarmWarning Popup.scr" )
34: End If
35:
36: //do something if user says ok to Failure message x number of times, like close the application?
37: End If
38: Else
39: 'zero counter
40: $ScanCountFailure = 0
41: 'reset alarm flag
```

```
42:  $PLCHearbeatDead = 0
43:
44:  If $IsScreenOpen( "AlarmWarning Popup.scr" ) And $AlarmWarningMessage=1 Then
45:      'return of 1 from IsScreenOpen means screen is open
46:      'msg=1 is comm failure message number
47:      'this is to take care of if PLC has re-established comm
48:      $Close( "AlarmWarning Popup.scr" )
49:  End If
50:
51: End If
52:
53: 'set previous value for next comparison
54: $PreviousScanCount = $ScanCount
55: 'reset this script's trigger
56: $PLCHearbeatScan = 0
57:
58: 'End of Script #4 PLC Heartbeat
```



```
1: 'The code configured here is executed while the condition configured in the Execution field is TRUE.
2:
3: 'the cycle was aborted due to a TC failure, not due to pressing the Abort Cycle button, so alert the operator
4: '/REV 3.1.0/ added vb vars TCErrors and TCMsg, reconfigured the msg text to account for the ESTOP condition
5:
6:
7: 'Variables available only for this group can be declared here.
8: Dim msg, i, TCErrors, TCMsg
9: $myText = "s#5"
10:
11: TCErrors = False      'reset
12:
13: 'reset local variables
14: msg = "Control was ABORTED due to: " & vbCrLf _
15:      & " 1) Thermocouple (TC) errors, or" & vbCrLf _
16:      & " 2) The ESTOP was pressed." & vbCrLf & vbCrLf _
17:      & "If a recipe was active, it has also been aborted." & vbCrLf _
18:      & "If a zone is controlled by the MTL Algorithm (Mold Temperature), then its Mold TC must be functional
to execute." & vbCrLf _
19:      & "All zones must have a functional Fluid TC in order for the PID and/or MTL Algorithm to control the
process." & vbCrLf & vbCrLf
20:
21:
22: 'check thermocouples, if TCs not connected or not functioning then figure out which one
23: For i=1 To $hn2_4zoneMachineSwitch ' NumberOfZones
24:     If Not $sTC_OK_MoldZ[i] Then
25:         TCErrors = True
26:         'set up error msg to operator
27:         TCMsg = TCMsg & "    MOLD Thermocouple for Zone " & i & " is not functioning properly!" & vbCrLf
28:     End If
29:     If Not $sTC_OK_FluidZ[i] Then
30:         TCErrors = True
31:         'set up error msg to operator
32:         TCMsg = TCMsg & "    FLUID Thermocouple for Zone " & i & " is not functioning properly!" & vbCrLf
33:     End If
34: Next
35:
36: If TCErrors Then
37:     msg = msg & "These Thermocouple errors must be fixed to continue:" & vbCrLf & TCMsg
38: End If
39:
40: 'log TC errors
```

```
41: $LogEvent 'marker for IWS xref tool to find all $Trace() and $SendEvent() log actions
42: $SendEvent( msg ) : $Trace( msg )
43:
44: If $IsScreenOpen( "Wait Dialog.scr" ) Then
45:     $hfCancelDownload = 1
46:     $LogEvent 'marker for IWS xref tool to find all $Trace() and $SendEvent() log actions
47:     $SendEvent( "The download was cancelled due to a TC malfunction." )
48:     $Trace( "The download was cancelled due to a TC malfunction." )
49:     $Close( "Wait Dialog.scr" )
50: End If
51:
52: If $IsScreenOpen( "Non-Recipe ConstTemper.scr" ) Or $IsScreenOpen( "Metal Tuning.scr" ) Then
53:     'do nothing
54: Else
55:     'open diagnostic so it will be more evident to operator which TC is malfunctioning
56:     $Open( "Diagnostic.scr" )
57:     'set NAV radio integer to highlight Diagnostic button
58:     $NavigationRadioButton = 3
59: End If
60:
61: msg = msg & vbCrLf & vbCrLf & "Error Location: Script#5"
62:
63: 'display accumulated ErrorMessage on popup
64: $TCErrormessage = msg
65: 'set message for popup, message 3=TCErrormessage
66: $AlarmWarningMessage = 3
67: 'open popup window
68: $Open( "AlarmWarning Popup.scr" )
69:
70: Do 'wait until back to Normal before exiting out of script;
71:     ' otherwise will try to open AlarmWarning Popup continuously as the script would continue to execute
72:     //need a timeout here? Don't think so.
73: Loop Until $sCycleNormal
74:
75:
76: 'End of Script #5 fCycleAbort
```

```
1: 'The code configured here is executed while the condition configured in the Execution field is TRUE.
2: //This script is triggered when PLC State sRecipeActive turns ON hstFileCreateTrigger
3:
4:
5: 'Variables available only for this group can be declared here.
6:
7:
8:
9: Call hstFileCreate()
10:     $hstFileCreateSub    'marker for IWS xref tool
11: //had to put this code in a sub to be able to call it from the Confirm Dialog script as well during
    AutoWrapAround
12:
13:
14: 'reset execution tag for this script is performed in the above global procedure
15:
16: 'End Script #6 hstFile
```

```
1: 'The code configured here is executed while the condition configured in the Execution field is TRUE.
2: '$TrendGraphUpdate is triggered by the Scheduler Task #1
3: '$TrendGraphUpdateManual is triggered by a button and/or screen script on Trend.scr
4:
5:
6: 'Variables available only for this group can be declared here.
7: Dim StartTime
8:
9:
10: 'If a recipe is running then update the Graph, otherwise don't need to unless done manually
11: If ( $sRecipeActive And $TrendGraphUpdate ) Or $TrendGraphUpdateManual Then
12:
13:     'refresh/load the look and feel settings of the graph
14:     $TrendLoadSavedSettings = $Toggle( $TrendLoadSavedSettings )
15:
16:     StartTime = $GetTickCount() 'milliseconds
17:     Do 'wait for XX milliseconds before reloading the file, this gives the settings time to update
18:     Loop Until $Abs( $GetTickCount() - StartTime ) >= 500
19:
20:     'option 0 of the function returns the Start Time and Date
21:     $TrendTimeScale = $GetHstInfo( $hstBatchFileName, 0 )
22:     'option 2 returns the Duration (in hours) of the file, but it is stored in the file as seconds
23:     $TrendDuration = $Format( "%h", $GetHstInfo( $hstBatchFileName, 2)*3600 )
24:     'reload the Trend Graph
25:     $hstBatchFileReload = $Toggle( $hstBatchFileReload )
26:
27: End If
28:
29: 'reset manual trigger
30: $TrendGraphUpdateManual = 0
31: 'reset execution tag for this script
32: $TrendGraphUpdate = 0
33:
34: 'End Script #7 Current Runtime Trend Graph Update
```

```
1: 'The code configured here is executed while the condition configured in the Execution field is TRUE.
2: 'triggered by Screen_OnOpen script
3:
4:
5: 'Variables available only for this group can be declared here.
6: Dim StartTime
7:
8:
9: 'refresh/load the look and feel settings of the graph
10: $TrendLoadSavedSettingsHist = $Toggle( $TrendLoadSavedSettingsHist )
11:
12: StartTime = $GetTickCount() 'milliseconds
13: Do 'wait for XX milliseconds before reloading the file, this gives the settings time to update
14: Loop Until $Abs( $GetTickCount() - StartTime ) >= 500
15:
16: 'option 0 of the function returns the Start Time and Date
17: $TrendTimeScaleHist = $GetHstInfo( $hstBatchFileNameHist, 0 )
18: 'option 2 returns the Duration (in hours) of the file, but it is stored in the file as seconds
19: $TrendDurationHist = $Format( "%h", $GetHstInfo( $hstBatchFileNameHist, 2)*3600 )
20: 'reload the Trend Graph
21: $hstBatchFileReloadHist = $Toggle( $hstBatchFileReloadHist )
22:
23:
24:
25: 'reset execution tag for this script
26: $TrendGraphUpdateHist = 0
27:
28: 'End Script #9 Historical Trend Graph Update
```

```
1: 'The code configured here is executed while the condition configured in the Execution field is TRUE.
2:
3:
4: 'Variables available only for this group can be declared here.
5:
6: 'reset
7: $hfCancelDownload = 0
8:
9:
10: 'call global procedure
11: Call RecipeLoadToPLC( )
12:     $RecipeLoadToPLCSub    'marker for IWS Xref tool
13:
14:
15: 'reset execution tag for this script
16: $LoadStartRecipe = 0
17:
18:
19: 'End Script #10
```

```
1: 'The code configured here is executed while the condition configured in the Execution field is TRUE.
2:
3:
4: 'Variables available only for this group can be declared here.
5:
6: 'Dim counter, myNum
7: 'counter = 0
8: 'myNum = 20
9:
10: 'Do
11: ' myNum = myNum - 1
12: ' counter = counter + 1
13: 'Loop Until myNum=10
14: 'MsgBox "The loop made " & counter & " repetitions. myNum value is: " & myNum
15:
16:
17:
18:
19:
20: 'reset execution tag for this script
21: $vbTrigger=0
22:
23: 'End Script #50 vbscript test code area
```

```
1: 'The code configured here is executed while the condition configured in the Execution field is TRUE.
2: '/REV 3.1.0/ created this script
3: '$DailyRunScript is turned ON in Startup Script and Scheduler
4: '$DailyRunScript is a retentive tag: =1 means to run the script, =2 means the script has already ran today
5: 'scheduler sets equal to 1 on a $Day change, but only sees that during runtime
6:
7:
8: 'Variables available only for this group can be declared here.
9:
10:
11: Call DailyRun( )
12:     $DailyRunSub     'marker for IWS Xref
13:
14:
15: 'reset execution tag for the script done in global procedure sub
16:
17:
18: 'End Script #90
```


Workspace Toolbar: Graphics Tab>Graphics Script

Note that the graphic script file is stored in the \<App Dir>\Screen\ScriptGraphics.gis location. The file can be viewed with a text editor. It is not advisable to edit this file, only viewing is recommended.

```
1: 'Variables with local scope can be declared and initialized here.
2:
3:
4: 'Procedures with local scope can be implemented here.
5:
6:
7: 'This procedure is executed just once when the graphic module (Viewer Task) is started.
8: Sub Graphics_OnStart()
9:
10:  'close IWS splash screen, cause user doesn't care
11:  $CloseSplashWindow()
12:
13:  ////////////////////////////////// check operating system //////////////////////////////////
14:  Dim OSMsg
15:  If Not $GetOS() = 2 Then 'return of 2=2000/XP/Vista
16:    OSMsg = "The Mold Heating and Cooling operator interface software was developed to run on " _
17:    & "Microsoft Windows 2000/XP/Vista." & vbCrLf _
18:    & "This software was extensively tested on the XP platform." & vbCrLf _
19:    & "The author cannot guarantee compatibility with any other operating system."
20:    'MsgBox OSMsg, vbCritical, "Error: Check Operating System..."
21:    $OSErrorMessage = OSMsg
22:    $AlarmWarningMessage = 8 'set to display $OSErrorMessage
23:    $Open( "AlarmWarning Popup.scr" )
24:  End If
25:  //////////////////////////////////
26:
27:  'set nav button to none
28:  $NavigationRadioButton = 0
29:
30:  ////////////////////////////////// Daily Stuff //////////////////////////////////
31:  '/REV 3.1.0/ added Daily Stuff
32:  If $DailyRunScript = 0 Or $DailyRunScriptLastRunDate<>$Date Then
33:    $DailyRunScript = 1
34:    '$DailyRunScript is a retentive tag: =1 means to run the script, =2 means the script has already ran
today
35:    'is rententive so that the script will only run once if the application is started and ended several
time during a day
36:    'scheduler sets equal to 1 on a $Day change, but only sees that during runtime
37:    $myText = "daily run on startup"
38:  End If
39:  //////////////////////////////////
40:
```

```
41: End Sub
42:
43: 'This procedure is executed continuously while the graphic module is running.
44: Sub Graphics_WhileRunning()
45:
46: End Sub
47:
48: 'This procedure is executed just once when the graphic module is closed.
49: Sub Graphics_OnEnd()
50:
51: End Sub
```

Workspace Toolbar: Database Tab>Global Procedures

Note that the global procedures script file is stored in the
\\<App Dir>\Config\ScriptGlobal.gis
location. The file can be viewed with a text
editor. It is not advisable to edit this file,
only viewing is recommended.

```
1: Option Explicit
2: 'Keep the Option Explicit statement in the first line of this interface.
3: 'This statement forces the declaration of variables.
4:
5: 'use Ctrl key + mouse scroll wheel to zoom in/out on any of the script interfaces
6:
7: //////////////////////////////////////
8: ////////////////////////////////////// Global Procedures //////////////////////////////////////
9: //////////////////////////////////////
10: 'Global Procedures: Procedures with global scope can be implemented here.
11:
12: 'A HMI tag was created for each routine and placed with the calling location and definition location for the routine, for easy xref
13:
14: 'Use $RunGlobalProcedureOnServer( "ProcedureName" ) to run these procedures on server to make
15: ' them run multi-threaded on the Server; or use a Task Script to call the procedure.
16: ' The Server is same as BGTask which is the StudioManager.exe process.
17: ' In SP4 the Viewer was seperated from StudioManager as a separate process.
18: ' The Viewer is same as Graphics Task or Graphics Script (and anything on Graphics Tab), and is the Viewer.exe process.
19: ' Otherwise without $RunGlobalProcedureOnServer(), if one of these procedures is called from a button,
20: ' which runs on the Viewer Task, they will run single-threaded, which could slow performance.
21: ' However, if I need a msgbox or keypad popup, I must call the procedure from the Viewer to keep anything in-focus.
22:
23: 'General:
24: 'PLC tags have a lowercase prefix as follows: xInput, yOutput, sStateFlag, fDiscreteFlag, nNumber, rFloat, cCounter, wString, tTimer, aASCII
25: 'HMI tags, that communicate with the PLC, have the same lowercase prefix as PLCs preceeded by an h: hfDiscreteFlag, hnNumber, etc
26: 'most local HMI tags do not have a prefix at this time, one can refer to the HMI tag database to verify the type and use Xref to find where used
27:
28:
29: ////////////////////////////////////// Revision documentation //////////////////////////////////////
30: //Developed by Josh Salmond (u60479), 816.997.4980, for NNSA's Kansas City Plant operated by Honeywell FM&T
31: //Funded by NNR Project 703030, activity 70303020, FY2007-2009
32: //Mold Heating and Cooling Pump Package Operator Interface (MHC PP OI)
33: //Commisionned by Greg Domer of PAG; Brian Spieker, Matt Trimmer, and Pete Scheuer of Plastics
34: //Target Platform: HMI is targeted to Windows XP SP2, using InduSoft Web Studio v6.1+SP4
35: //Description of application
36: 'This software is designed to replace the exisiting MHC PP OI originally developed by Dick Floersch in the 1980's with at least all the same
37: ' functionality of the original interface.
38: 'It's scope is to give the user a means to interface with the fluid and mold temperatures via recipe or manual const temper control for each
   zone and to
39: ' provide temperature feedback of the zones.
40: 'This software works along with the PLC software, communication via OPC, to monitor the temperature SP of a
41: ' mold or fluid zone and provide a control output to the control valves on the pump package. The PP interfaces with the
```

```
42:   ' MHC Central System which provides the source of hot and cold heat transfer fluid.
43:
44: //----- document release notes here -----:
45: 'HMI and PLC revision numbers may not always be the same.
46: //----- Release v3.0.0 date 10/31/2008 JAS -----:
47: 'Initial Release
48:
49: //----- Release v3.1.0 date 04/08/2009 JAS -----:
50: 'interim rev control as such: v3.1dx.0, where d signifies development of the next rev(3.1), and where x is the intermediate dev rev
51: 'EXTENSIVE REVISION
52: 'Revision notes preceded by "/REV 3.1.0/"
53: 'Fixed issues with running MsgBox and Keypad on the Server as opposed to the Viewer (see above under Global Procedures heading)
54: 'Incorporated more operations in Global Procedures by adding/creating new routines
55: 'Moved several recipe tasks from $Confirm script to Global Procedures
56: 'Deleted "Confirm Dialog.scr" screen
57: 'Added "Input SN Dialog.scr" screen
58: 'Changed downloading scripting
59: 'Added time and date clock synchronization with PLC
60: 'Change SegGrapher screen from Replace(Complete) to Popup, so when it is closed, I don't OpenPrevious() which was causing the recipe to reload
    and set Z radio=1
61: 'Added checks to make sure Seg1 TemperSP and PRHT Mold Temp Limits complemented each other; added tag $hfMoldTemperLimitsChanged
62: 'Changed SegsDisplayed, Man/Auto-Prht, Skip/Run-Prht buttons from a single checkbox to 2 radio buttons; on screens RecEd-Seg and RecViewer
63: 'Changed CycleStatus display box on Status.scr to smart message, put old one on "zOld Items to Keep.scr"
64: ' had to change OPC sheets for sMoldTemperLimitsBreachZ[x]
65: 'Added indicator $xMHCRelay on Diagnostic.scr for when this PLC input comes on
66: 'Moved CycleStatus, DryCycle, AbortCycle objects and accompanying OPC data from "Status.scr" to "Diagnostic.scr", except sCycleNormal & sIDLE
67: 'Created OPC sheet 50, every 500ms, to accomodate sCycleNormal & sIDLE and potential others
68: 'Changed NAV procedure on NRCT, to update CtlMode based on what was running on PLC
69: 'Changed the way control modes were changed. Instead of toggling the button to change, now open screen "Control Mode Dialog.scr" to make
    selection.
70: 'Added xESTOP indicators
71: 'Added recipe report output to RTF file on PAG Utilities and every time the recipe is saved for historical documentation
72: 'Added use of help screens
73: 'Added more PLC modules to "PAG Utilities.scr"
74: 'Added documentation to global procedures where they are called from
75: 'Removed 'View PLC' button from 'Recipe Viewer.scr', Operator may only see what is currently running, if nothing running, then can open a recipe,
    and
76: ' will auto-load the last recipe that was downloaded to PLC
77: 'Changed screen scripts on 'Recipe Viewer.scr' and 'RecEd-Segs & RecEd-Consts'
78: 'Changed words 'Load/Start Recipe' to 'Start Recipe' on button on 'Recipe Viewer.scr, RecEd-Consts.scr, RecEd-Segs.scr'
79: 'Added realtime trend to NRCT
80: 'Added button to Diagnostic.scr to re-run the last recipe that was downloaded
```

```

81: 'Used custom msgbox on 'Abort Cycle' button instead of the IWS Confirm property on the Command dynamic
82: 'Upgraded to 4000 tag license.
83:
84: ////////////////////////////////// MHCVerRev( ) - software revision control //////////////////////////////////
85: 'this is where the Version and Revision numbers of the HMI software are controlled
86: 'Format is X.Y.Z, refer to SQA plan "MHC SQA.doc" paragraphs 3.9-3.12
87: 'X is major-like platform change, Y is significant changes to the app structure or code, Z is minor bug fixes or philosophy changes
88: 'Called from Task Startup Script
89: 'Called from Global RecipeFileHeaderConstants( )
90: Sub MHCVerRev( )
91:     $MHCVerRevSub    'marker for IWS Xref tool
92:     $MHCVersionRevision = "HMI Version: 3.1.0"
93: End Sub 'MHCVerRev( )
94: //////////////////////////////////////////////////////////////////////////////////
95: //////////////////////////////////////////////////////////////////////////////////
96:
97:
98: //////////////////////////////////////////////////////////////////////////////////
99: ////////////////////////////////// RecipeNameCheck( stringName ) //////////////////////////////////
100: //////////////////////////////////////////////////////////////////////////////////
101: //using previous string assignment to tagname $RecConfirmDlg as the function name
102: //Checks name for invalid characters thru CheckInputName( Name ) and checks if name already exists
103: //Called from Global RecipeNew( ), and Global RecipeSaveAs( )
104: '/REV 3.1.0/ moved from Confirm script to here because MsgBox, Keypad, and other graphics do not receive focus when called from Task Script
105: Function RecipeNameCheck( Name )
106:     $RecipeNameCheckFunc    'marker for IWS Xref tool
107:
108:     Dim fso, msg
109:
110:     //check name for invalid characters, call function on global
111:     If CheckInputName( Name ) = 1 Then    'function=1 is fault
112:         $CheckInputNameFunc    'marker for IWS Xref tool
113:         'new name error
114:         RecipeNameCheck = 1
115:     Else
116:         //Check to see if name is already existing, make user specify a new name.
117:         Set fso = CreateObject("Scripting.FileSystemObject")
118:         If ( fso.FileExists( $RecipesFolderPath & "\" & Name & $RecipeFileExtension ) Or Name = "" ) Then
119:             'new name error
120:             RecipeNameCheck = 1
121:             msg = "The name you specified already exists.      " & vbCrLf & "Please specify a new name." & vbCrLf _
122:                 & "The name input was (      " & Name & "      )." & vbCrLf _

```

```

123:         & vbCrLf & "Error Location: RecipeNameCheck( )"
124:     MsgBox msg, vbOKOnly + vbExclamation, "Recipe Name Error..."
125: Else
126:     'new name has met all previous checks
127:     RecipeNameCheck = 0
128: End If
129: End If
130:
131: End Function    'RecipeNameCheck( Name )
132: ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
133:
134:
135: ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
136: ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////          RecipeNew( )          ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
137: ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
138: //using previous string assignment to tagname $RecConfirmDlg as the function name
139: //creates a new recipe file on the disk
140: //Called from Global RecipeSaveThenAction( )
141: //Called from "Save" and "New" buttons on "Recipe Editor-Constans.scr" and "Recipe Editor-Segments.scr"
142: '/REV 3.1.0/ moved from Confirm script to here because MsgBox, Keypad, and other graphics do not receive focus when called from Task Script
143: Sub RecipeNew( )
144:     $RecipeNewSub    'marker for IWS Xref tool
145:
146:     Dim msg, msgReturn
147:
148:     'launch alphanumeric keypad for operator input of new name, and check
149:     $InputName = ""    'clear out tag, so keypad shows nothing
150:     $KeypadReturn = $Keypad( "$InputName", "AlphaNumeric", 0, "Input the Recipe Name for your New recipe.")
151:     If $KeypadReturn = 0 Then
152:
153:         'If new name has been successfully established, then create new recipe
154:         If RecipeNameCheck( $InputName ) = 0 Then    'return of 0 is success
155:             $RecipeNameCheckFunc    'marker for IWS Xref tool
156:             'clear out IWS tags by loading the "BLANK" recipe
157:             $RecipePathAndFileName = $GetAppPath() & "BLANK" & $RecipeFileExtension
158:             $Recipe( "Load:Recipe1" )
159:
160:             '/REV 3.1.0/ was confusing to see "BLANK" as the recipe name, so repopulate that tag
161:             $RecipeName = $InputName
162:
163:             'Need to set the #Zones. Let operator choose 2|4, default is 2.
164:             msg ="Is this a 2-zone machine?" & vbCrLf & "Please choose whether this is a 2-zone machine, or a 4-zone machine:" _

```



```

165:             & vbCrLf & vbCrLf & "Choose YES button for 2-zone. Choose NO button for 4-zone."
166:     msgReturn = MsgBox( msg, vbQuestion + vbYesNo, "Is this a 2-zone machine?" )
167:     If msgReturn = vbYes Then $hn2_4zoneMachineSwitch = 2 Else $hn2_4zoneMachineSwitch = 4
168:
169:     '/REV 3.1.0/ don't automatically save the recipe if creating new, force user to save
170:     $RecipeIsSaved = 0
171:     '/REV3.1.0/ set trigger that new recipe has been created, and if when 1st RecipeSave() is called, if trigger is true, skip overwrite
question
172:     ' reset trigger in RecipeSave()
173:     $NewRecipeCreated = True
174: End If
175: End If
176:
177: End Sub 'RecipeNew( )
178: //////////////////////////////////////
179:
180:
181: //////////////////////////////////////
182: ////////////////////////////////////// RecipeSaveAs( ) //////////////////////////////////////
183: //////////////////////////////////////
184: //using previous string assignment to tagname $RecConfirmDlg as the function name
185: //saves an existing recipe to disk as a new name specified by user
186: //Called from the "Save As" button on "Recipe Editor-Constans.scr" and "Recipe Editor-Segments.scr"
187: '/REV 3.1.0/ moved from Confirm script to here because MsgBox, Keypad, and other graphics do not receive focus when called from Task Script
188: Sub RecipeSaveAs( )
189:     $RecipeSaveAsSub 'marker for IWS Xref tool
190:
191:     Dim msg, msgReturn
192:     msg = "Save As: Make a copy of the current file? " & vbCrLf & "( "& $RecipeName &" ) "
193:     msgReturn = MsgBox( msg, vbQuestion + vbYesNo, "Save Recipe As...")
194:     If msgReturn = vbYes Then
195:
196:         'launch alphanumeric keypad for operator input of new name, and check
197:         $InputName = $RecipeName 'set equal to current, so user knows what current name is, and can make minor mods to the name if desired
198:         $KeypadReturn = $Keypad( "$InputName", "AlphaNumeric", 0, "Input the Recipe Name for your New recipe.")
199:         If $KeypadReturn = 0 Then
200:
201:             'If new name has been successfully established, then create new recipe
202:             If RecipeNameCheck( $InputName ) = 0 Then 'return of 0 is success
203:                 $RecipeNameCheckFunc 'marker for IWS Xref tool
204:                 'skip "overwrite question", by setting second param true
205:                 Call RecipeSave( $InputName, True )

```

```
206:             $RecipeSaveSub      'marker for IWS Xref tool
207:         End If
208:     End If
209: End If
210:
211: End Sub      'RecipeSaveAs( )
212: //////////////////////////////////////////////////
213:
214:
215: //////////////////////////////////////////////////
216: //////////////////////////////////////////////////      RecipeOpen( stringRecipeName )      //////////////////////////////////////////////////
217: //////////////////////////////////////////////////
218: //using previous string assignment to tagname $RecConfirmDlg as the function name
219: //opens an existing recipe file from disk
220: //Called from "Open" button on "Open Recipe Dialog.scr",
221: //Called from ScreenScripts on "Recipe Editor-Constans.scr" and "Recipe Editor-Segments.scr" and "Recipe Viewer.scr"
222: '/REV 3.1.0/ moved from Confirm script to here because MsgBox, Keypad, and other graphics do not receive focus when called from Task Script
223: Sub RecipeOpen( name )
224:     $RecipeOpenSub      'marker for IWS Xref tool
225:
226:     $myText = "open rec"
227:     Dim msg, msgReturn, StartTime
228:
229:     'clear out IWS tags by loading the "BLANK" recipe
230:     $RecipePathAndFileName = $GetAppPath() & "BLANK" & $RecipeFileExtension
231:     $Recipe( "Load:Recipe1" )
232:
233:     'Generate full path for selected file; $RecipeName will be populated when $Recipe( "Load:Recipe1" )
234:     $RecipePathAndFileName = $RecipesFolderPath & "\" & name & $RecipeFileExtension
235:     'run recipe script, Load selected recipe file
236:     $Recipe( "Load:Recipe1" )
237:     'may need to add error handling if the Recipe Load did not complete successfully
238:     'set recipe saved flag to unhide Load/Start Recipe button
239:     $RecipeIsSaved = 1
240:     'default to Segment Display, Zone 1, Seg 0-9
241:     $ZoneSelectedRadio = 1
242:     ' $SegmentsDisplayed = 0
243:
244:     'if flag is set that got here from MTL tuning screen button, reset flag, then call WriteResults2rec
245:     If $Flag2OpenNSaveMTR Then
246:         'reset flag
247:         $Flag2OpenNSaveMTR = 0
```

```
248:         'call sub on global to assign tags and save new data
249:         Call WriteMTLTuningResults2recFile()
250:         $WriteMTLTuningResults2recFileSub    'marker for IWS Xref tool
251:     End If
252:
253:     $Close( "Open Recipe Dialog.scr" )
254:
255: End Sub    'RecipeOpen( name )
256: //////////////////////////////////////////////////
257:
258:
259: //////////////////////////////////////////////////
260: //////////////////////////////////////////////////          RecipeOpenPLC( )          //////////////////////////////////////////////////
261: //////////////////////////////////////////////////
262: //using previous string assignment to tagname $RecConfirmDlg as the function name
263: //Code to handle opening a recipe file on the disk, but based on the name in the PLC
264: //Called from screen script on "Recipe Viewer.scr"
265: '/REV 3.1.0/ moved from Confirm script to here because MsgBox, Keypad, and other graphics do not receive focus when called from Task Script
266: '/REV 3.1.0/ changed screen script on 'Recipe Viewer.scr' to show the currently running recipe or nothing
267: Sub RecipeOpenPLC( )
268:     $RecipeOpenPLCSub    'marker for IWS Xref tool
269:
270:     Dim msg, msgReturn
271:     msg = "NOTICE:" & vbCrLf _
272:         & "You will be viewing the latest saved recipe from the disk, " & vbCrLf _
273:         & "based on the recipe name currently downloaded." _
274:         & vbCrLf & vbCrLf & "Current active recipe is: "& $hwRecipeName
275: '310     msgReturn = MsgBox( msg, vbOKCancel + vbExclamation, "View PLC recipe file")
276:
277: '310     If msgReturn = vbOK Then
278:         //clear out IWS tags by loading the "BLANK" recipe
279:         $RecipePathAndFileName = $GetAppPath() & "BLANK" & $RecipeFileExtension
280:         $Recipe( "Load:Recipe1" )
281:
282:         //Generate full path
283:         $RecipePathAndFileName = $RecipesFolderPath & "\" & $hwRecipeName & $RecipeFileExtension
284:         //run recipe script, Load recipe file
285:         $Recipe( "Load:Recipe1" )
286:         'may need to add error handling if the Recipe Load did not complete successfully
287:         //set recipe saved flag to unhide Load/Start Recipe button
288:         $RecipeIsSaved = 1
289:         //default to Segment Display, Zone 1, Seg 0-9
```

```
290:         $ZoneSelectedRadio = 1
291:         $SegmentsDisplayed = 0
292: '310     End If
293:
294: End Sub     'RecipeOpenPLC( )
295: //////////////////////////////////////////////////
296:
297:
298: //////////////////////////////////////////////////
299: ////////////////////////////////////////////////// Call RecipeSaveThenAction( string ) //////////////////////////////////////////////////
300: //////////////////////////////////////////////////
301: '/REV 3.1.0/ newly created
302: //used when changing screens, or when possibly changing the loaded recipe to ask user if want to Save before doing that action
303: //Actions are: "New", "Open", "Close", "Save"
304: //Called from "Recipe Viewer" button, "Recipe Status" button, "Trend Graph" button, "Historical Trend Graph" button,
305: // "Constant Temperature" button, "Event Viewer" button, "PAG Utilities" button; all on "NAVIGATION.scr"
306: //Called from "New" button, "Open" button, and "Close" buttons on "Recipe Editor-Constants.scr" and "Recipe Editor-Segments.scr"
307: //Called from "View Disk" and "Close Recipe" buttons on "Recipe Viewer.scr"
308: Sub RecipeSaveThenAction( Action )
309:     $RecipeSaveThenActionSub     'marker for IWS Xref tool
310:
311: Dim return
312:
313: If Action = "New" Then
314:     If $RecipeName<>" " Then
315:         return = MsgBox( "Would you like to save recipe ""& $RecipeName &"" first? ", vbQuestion + vbYesNo, "Save before creating new?" )
316:         If return = vbYes Then
317:             'call global procedure, skip the question
318:             Call RecipeSave( $RecipeName, True )
319:             $RecipeSaveSub     'marker for IWS Xref tool
320:             Call RecipeNew( )
321:             $RecipeNewSub     'marker for IWS Xref tool
322:         Else 'return=vbNo
323:             ' "close" the recipe
324:             $RecipeName = " "
325:             Call RecipeNew( )
326:             $RecipeNewSub     'marker for IWS Xref tool
327:         End If
328:     Else
329:         Call RecipeNew( )
330:         $RecipeNewSub     'marker for IWS Xref tool
331:     End If
```

```
332: End If
333:
334: If Action = "Open" Then
335:     return = MsgBox( "Would you like to save recipe ""& $RecipeName &"" first? ", vbQuestion + vbYesNo, "Save before opening?" )
336:     If return = vbYes Then
337:         'call global procedure, skip the question
338:         Call RecipeSave( $RecipeName, True )
339:         $RecipeSaveSub 'marker for IWS Xref tool
340:         $Open( "Open Recipe Dialog.scr" )
341:     Else 'return=vbNo
342:         ' "close" the recipe
343:         $RecipeName = ""
344:         $Open( "Open Recipe Dialog.scr" )
345:     End If
346: End If
347:
348: If Action = "Close" Then
349:     return = MsgBox( "Would you like to save recipe ""& $RecipeName &"" first? ", vbQuestion + vbYesNo, "Save before closing?" )
350:     If return = vbYes Then
351:         'call global procedure, skip the question
352:         Call RecipeSave( $RecipeName, True )
353:         $RecipeSaveSub 'marker for IWS Xref tool
354:         ' "close" the recipe
355:         $RecipeName = ""
356:     Else 'return=vbNo
357:         $RecipeName = ""
358:     End If
359: End If
360:
361: If Action = "Save" Then
362:     return = MsgBox( "Would you like to save recipe ""& $RecipeName &"" first? ", vbQuestion + vbYesNo, "Save Recipe?" )
363:     If return = vbYes Then
364:         'call global procedure, skip the question
365:         Call RecipeSave( $RecipeName, True )
366:         $RecipeSaveSub 'marker for IWS Xref tool
367:     Else 'return=vbNo
368:         '/REV 3.1.0/ if do not want to save then clear out the recipe
369:         '310 'call global
370:         '310 'Call RecipeOpen( $RecipeName )
371:         $RecipeName = ""
372:     End If
373: End If
```

```
374:
375: End Sub      'RecipeSaveThenAction( Action )
376: //////////////////////////////////////////////////
377:
378:
379: //////////////////////////////////////////////////
380: //////////////////////////////////////////////////      RecipeSave( stringName, bool )      //////////////////////////////////////////////////
381: //////////////////////////////////////////////////
382: //using previous string assignment to tagname $RecConfirmDlg as the function name
383: //saves the currently opened recipe file loaded in HMI tags to a disk file
384: //Called from "Save" button on "Recipe Editor-Constants.scr" and "Recipe Editor-Segments.scr"
385: //Called from Globals RecipeSaveAs( ), RecipeSaveThenAction( ), and WriteMTLTuningResults2recFile( )
386: '/REV 3.1.0/ moved from Confirm script to here because MsgBox, Keypad, and other graphics do not receive focus when called from Task Script
387: Sub RecipeSave( NameToSave, SkipQ )
388:     $RecipeSaveSub      'marker for IWS Xref tool
389:
390:     Dim msgReturn
391:     $myText = "save rec"
392:
393:     '/REV 3.1.0/ added SkipQ logic for new recipe
394:     If $NewRecipeCreated Then $NewRecipeCreated=False : SkipQ=True
395:
396:     If Not SkipQ Then
397:         msgReturn = MsgBox("Overwrite existing recipe file "& NameToSave &"?", vbYesNo, "Save Recipe...")
398:     End If
399:
400:     If msgReturn = vbYes Or SkipQ Then
401:         $RecipeName = NameToSave
402:         'set correct file path for Recipe Script
403:         $RecipePathAndFileName = $RecipesFolderPath & "\" & $RecipeName & $RecipeFileExtension
404:
405:         'call function on Global Procedures page in IF statement, and if function returns ok (value=0), then continue
406:         If CheckRecipe = 0 Then
407:             $CheckRecipeFunc      'marker for IWS Xref tool
408:             'call sub on global procedures
409:             Call RecipeFileHeaderConstants()
410:             $RecipeFileHeaderConstantsSub      'marker for IWS Xref tool
411:
412:             '/REV 3.1.0/ clear flag
413:             $hfMoldTemperLimitsChanged = 0
414:
415:             If $Recipe( "Save:Recipe1" ) = 0 Then      'Recipe( )=0 means the function return was successful
```

```
416:         //Set recipe saved flag to unhide Load/Start Recipe button
417:         $RecipeIsSaved = 1
418:         'MsgBox "recipe saved"
419:         '/REV 3.1.0/ create a new report for historical purposes to keep recipe info based on the yyyy-mm-dd they were saved
420:         If $hn2_4zoneMachineSwitch = 4 Then
421:             $Report( "Disk:Report1.rep" )
422:         Else
423:             $Report( "Disk:Report2.rep" )
424:         End If
425:
426:     Else If $Recipe( "Save:Recipe1" ) = 5 Then      'disk error
427:         $RecipeIsSaved = 0
428:         MsgBox "Recipe was not saved successfully due to disk error." & vbCrLf _
429:             & "Check that the filename does not contain the following special characters: / | \ ... what else? " & vbCrLf _
430:             & "Remove special characters from the filename and press the Save button again." & vbCrLf _
431:             & vbCrLf & "Error Location: RecipeSave( )", vbExclamation, "Save RecipeError..."
432:     Else
433:         $RecipeIsSaved = 0
434:         MsgBox "Recipe was not saved successfully due to some unknown error." & vbCrLf _
435:             & vbCrLf & "Error Location: RecipeSave( )", vbExclamation, "Save Recipe Error..."
436:     End If
437: End If
438: Else 'CheckRecipe = 1
439:     $RecipeIsSaved = 0
440: End If
441: End If
442:
443: End Sub      'RecipeSave( NameToSave, SkipQ )
444: //////////////////////////////////////////////////
445:
446:
447: //////////////////////////////////////////////////
448: //////////////////////////////////////////////////      RecipeDelete( )      //////////////////////////////////////////////////
449: //////////////////////////////////////////////////
450: //using previous string assignment to tagname $RecConfirmDlg as the function name
451: //Deletes the recipe file from disk, based on the current recipe loaded in HMI tags
452: //Called from "Delete" button on "Recipe Editor-Constants.scr" and "Recipe Editor-Segments.scr"
453: '/REV 3.1.0/ moved from Confirm script to here because MsgBox, Keypad, and other graphics do not receive focus when called from Task Script
454: Sub RecipeDelete( )
455:     $RecipeDeleteSub      'marker for IWS Xref tool
456:     Dim msgReturn
457:     msgReturn = MsgBox( "Delete recipe file: "& $RecipeName & " ?", vbYesNo + vbQuestion, "Delete recipe file?" )
```

```

458:     If msgReturn = vbYes Then
459:         //set correct file path for Recipe Script
460:         $RecipePathAndFileName = $RecipesFolderPath & "\" & $RecipeName & $RecipeFileExtension
461:         //Delete selected recipe file
462:         $Recipe( "Delete:Recipe1" )
463:         //clear out tag database name; also results in hiding items on Recipe screens
464:         $RecipeName = ""
465:         //set recipe saved flag to hide Load/Start Recipe button
466:         $RecipeIsSaved = 0
467:         //default to Segment Display, Zone 1, Seg 0-9
468:         $ZoneSelectedRadio = 1
469:         $SegmentsDisplayed = 0
470:     End If
471: End Sub 'RecipeDelete( )
472: //////////////////////////////////////
473:
474:
475: //////////////////////////////////////
476: //////////////////////////////////////           RecipeLoadToPLC( )           //////////////////////////////////////
477: //////////////////////////////////////
478: //using previous string assignment to tagname $RecConfirmDlg as the function name
479: //Code to handle loading recipe data to WinPLC when "Load/Start Recipe" button is pressed
480: //Called from Task Script 10 - Load/Start Recipe
481: '/REV 3.1.0/ moved from Confirm script to here because MsgBox, Keypad, and other graphics do not receive focus when called from Task Script
482: Sub RecipeLoadToPLC( )
483:     $RecipeLoadToPLCSub 'marker for IWS Xref tool
484:
485:     $myText = "RecipeLoadToPLC( )"
486:     Dim msg, i, TError, StartTime
487:     msg = ""
488:
489:     //set correct file path for Recipe Script
490:     $RecipePathAndFileName = $RecipesFolderPath & "\" & $RecipeName & $RecipeFileExtension
491:     //reload the recipe in case the user navigated to a screen that may have changed an HMI variable contained w/i the recipe, or if OPC
changed a var
492:     $Recipe( "Load:Recipe1" )
493:     'may need to add error handling if the Recipe Load did not complete successfully
494:
495:     TError = False
496:     'check thermocouples, if TCs not connected or not functioning then recipe cannot work
497:     'check fluid and if all not ok, show message. if recipe is mtl then check mtl too.
498:     For i=1 To $hn2_4zoneMachineSwitch ' NumberOfZones

```



```
499:         If $hnControlModeZ[i] >= $MoldConstantTemper And Not $sTC_OK_MoldZ[i] Then
500:             'set error flag if any one zone not good
501:             TCErrors = True
502:             'set up error msg to operator
503:             msg = msg & vbCrLf & "    MOLD Thermocouple for Zone " & i & " is not functioning properly!"
504:         End If
505:         If $hnControlModeZ[i] > $ControlValve And Not $sTC_OK_FluidZ[i] Then
506:             'set error flag if any one zone not good
507:             TCErrors = True
508:             'set up error msg to operator
509:             msg = msg & vbCrLf & "    FLUID Thermocouple for Zone " & i & " is not functioning properly!"
510:         End If
511:     Next
512:
513:     If TCErrors Then
514:         'msg TCs not ok, cannot proceed with recipe
515:         msg = "Thermocouple Errors:" & vbCrLf _
516:             & "Thermocouple errors must be fixed before a recipe can be run." & vbCrLf _
517:             & msg & vbCrLf _
518:             & vbCrLf & vbCrLf & "Error Location: RecipeLoadToPLC( )"
519:         'display accumulated ErrorMessage on popup
520:         $TCErrorsMessage = msg
521:         $AlarmWarningMessage = 3
522:         $Open( "AlarmWarning Popup.scr" )
523:     End If
524:
525:     If Not TCErrors And CheckRecipe = 0 Then 'return of 0 means function execution good, return of 1 means problem
526:         $CheckRecipeFunc 'marker for IWS Xref tool
527:         'set dynamic message for "Wait Dialog.scr"
528:         $WaitMessage = 0
529:         $WaitProgress = 0
530:         $Open( "Wait Dialog.scr" )
531:
532:         'call sub on Global Procedures page to start the download
533:         If RecipeDownload = 0 Then 'return of 0 means function execution good, return of 1 means problem
534:             $RecipeDownloadFunc 'marker for IWS Xref tool
535:             'start hst
536:             Call hstFileCreate() : $hstFileCreateSub 'marker for IWS xref tool
537:             'after returning from sub, close screen
538:             $Close( "Wait Dialog.scr" )
539:             'open diagnostic for cycle status and set NAV radio integer to highlight Diagnostic button
540:             $Open( "Diagnostic.scr" )
```

```
541:         $NavigationRadioButton = 3
542:         'trace which recipe was loaded
543:         $LogEvent 'marker for IWS xref tool to find all $Trace() and $SendEvent() log actions
544:         $SendEvent( "Recipe '" & $RecipeName & "' was loaded to controller." )
545:         $Trace( "Recipe '" & $RecipeName & "' was loaded to controller." )
546:         '/REV 3.1.0/ set tag for last recipe loaded, this is recipe that will load in Recipe Viewer screen
547:         $RecipeLastLoaded = $RecipeName
548:     Else 'comm error, or download cancelled by operator, or fCycleAbort and RecipeDownload=1
549:         $Close( "Wait Dialog.scr" )
550:         $hfCancelDownload = 1 'need to do this to effect PLC state change if comm error, otherwise would already be set by user
551:         $LogEvent 'marker for IWS xref tool to find all $Trace() and $SendEvent() log actions
552:         $SendEvent( "The download was cancelled either due to button hfCancelDownload OR comm error OR PLC fCycleAbort in RecipeDownload(
). " )
553:         $Trace( "The download was cancelled either due to button hfCancelDownload OR comm error OR PLC fCycleAbort in RecipeDownload( ). "
)
554:     End If
555: Else
556:     'If CheckRecipe = 1 then AlarmWarningPopup.scr should address the errors
557: End If
558:
559: $myText = "end of RecipeLoadToPLC( )"
560: End Sub 'RecipeLoadToPLC( )
561: //////////////////////////////////////
562:
563:
564: //////////////////////////////////////
565: ////////////////////////////////////// hstFileCreate( ) //////////////////////////////////////
566: //////////////////////////////////////
567: // code to create the history file path and file name to enable the Trend Task to save data to it
568: //This sub creates the string "\\<drive>\<folder containing .APP file>\Hst\Batch\YYYY-MM-DD_HHMM_RecipeName.hst"
569: // and puts that value in $hstBatchFileName; which is used by the Trend Task #2.
570: //Called from Task Script 6 - Create hst file and path
571: //Called from Global RecipeLoadToPLC( )
572: Sub hstFileCreate( )
573:     $hstFileCreateSub 'marker for IWS xref tool
574:
575:     If $Month < 10 Then
576:         'add a preceding zero to the month
577:         $hstBatchFileName = $InfoAppHstDir() & "Batch\" & $Year & "-0" & $Month
578:     Else
579:         $hstBatchFileName = $InfoAppHstDir() & "Batch\" & $Year & "-" & $Month
580:     End If
```

```
581:
582:  If $Day < 10 Then
583:      'add a preceding zero to the day
584:      $hstBatchFileName = $hstBatchFileName & "-0" & $Day
585:  Else
586:      $hstBatchFileName = $hstBatchFileName & "-" & $Day
587:  End If
588:
589:  If $Hour < 10 Then
590:      'add a preceding zero to the hour; hour is already 24 hour format
591:      $hstBatchFileName = $hstBatchFileName & "_0" & $Hour
592:  Else
593:      $hstBatchFileName = $hstBatchFileName & "_" & $Hour
594:  End If
595:
596:  If $Minute < 10 Then
597:      'add a preceding zero to the minute
598:      $hstBatchFileName = $hstBatchFileName & "0" & $Minute
599:  Else
600:      $hstBatchFileName = $hstBatchFileName & $Minute
601:  End If
602:
603:  'capture part serial number, only if running Manual Preheat (cause Auto Preheat has no pause point to capture multiple SNs
604:  If $hfManualPreheat Then
605:      'finish string
606:      $hstBatchFileName = $hstBatchFileName & "_" & $hwRecipeName & "_SN" & $hstSerialNumber & ".hst"
607:      '/REV 3.1.0/ change $SelectedFile to $hwRecipeName
608:  Else
609:      'finish string
610:      $hstBatchFileName = $hstBatchFileName & "_" & $hwRecipeName & ".hst"
611:      '/REV 3.1.0/ change $SelectedFile to $hwRecipeName
612:  End If
613:
614:  '/REV 3.1.0/ capture first set of data by calling Trend Script #2
615:  $TrendUpdate = 1
616:
617:  'reset script trigger
618:  $hstFileCreateTrigger = 0
619:
620: End Sub 'hstFileCreate( )
621: //////////////////////////////////////
622:
```

```

623: //////////////////////////////////////////////////
624: /////////////////////////////////// CheckInputName( string ) ///////////////////////////////////
625: ///////////////////////////////////
626: //////////////////////////////////////////////////
627: 'this function will check an alphanumeric string name input using the IWS keyboard for invalid filename characters
628: ' for filenames to be saved in the Windows XP file system
629: //Called from Screen Script on "Input SN Dialog.scr"
630: //Called from Globals RecipeNameCheck( ), and WriteMTLTuningResults2txtFile( )
631: Function CheckInputName( CheckName )
632:     $CheckInputNameFunc 'marker for IWS Xref tool
633:     Dim InvalidChar(9), ArrSum, i, msg
634:     Erase InvalidChar : ArrSum=0
635:
636:     'check for invalid characters, the return integer from the function corresponds to first occurrence of the specified char,
637:     'the occurrence can be thought of as the index position of the character in the string, if the string was an array of characters
638:     'so $StrStrPos( "ab\c", "\" ) would return 2 (zero based); therefore, the actual char position would be 3 (1 based, 2+1=3)
639:     'if no specific char found, then function returns -1, so change that to 0 by also adding 1, so can add all elements and check sum<=0
640:     'also if the first char is an invalid char, adding 1 to the return of 0 will also make the equation=1
641:     InvalidChar(0) = $StrStrPos( CheckName, "\" ) + 1
642:     InvalidChar(1) = $StrStrPos( CheckName, "/" ) + 1
643:     InvalidChar(2) = $StrStrPos( CheckName, ":" ) + 1
644:     InvalidChar(3) = $StrStrPos( CheckName, "*" ) + 1
645:     InvalidChar(4) = $StrStrPos( CheckName, "?" ) + 1
646:     InvalidChar(5) = $StrStrPos( CheckName, Chr(34) ) + 1 'chr(34) = " (double quote)
647:     InvalidChar(6) = $StrStrPos( CheckName, "<" ) + 1
648:     InvalidChar(7) = $StrStrPos( CheckName, ">" ) + 1
649:     InvalidChar(8) = $StrStrPos( CheckName, "|" ) + 1
650:
651:     MsgBox InvalidChar(0) & ", "& InvalidChar(1) & ", "& InvalidChar(2) & ", "& InvalidChar(3) & ", "& InvalidChar(4) & ", "& InvalidChar(5) _
652:     & ", "& InvalidChar(6) & ", "& InvalidChar(7) & ", "& InvalidChar(8) & ", "& InvalidChar(9)
653:
654:     For i = 0 To 8
655:         ArrSum = ArrSum + InvalidChar(i)
656:     Next
657:     If ArrSum > 0 Then
658:         'assign fault value to function name
659:         CheckInputName = 1
660:         msg = "One of the following invalid characters were found" & vbCrLf & "\" / : * ? " & Chr(34) & " < > |" & vbCrLf _
661:             & "Correct the name to continue." & vbCrLf & vbCrLf _
662:             & "The name input was ( " & CheckName & " )." & vbCrLf _
663:             & vbCrLf & "Error Location: CheckInputName( )"
664:         MsgBox msg, vbExclamation, "Invalid character found!"

```

```
665:      Exit Function
666:    Else
667:      'assign OK value to function (most of IWS functions return 0 if OK, so tried to keep that consistency)
668:      CheckInputName = 0
669:    End If
670: End Function 'CheckInputName( CheckName )
671: //////////////////////////////////////
672:
673:
674: //////////////////////////////////////
675: //////////////////////////////////////          CheckRecipe( )          //////////////////////////////////////
676: //////////////////////////////////////
677: // Code to check to make sure a recipe is defined and setup before allowing a save
678:     'checks the following recipe configurations
679:         'PID|MTL control for all zones, don't allow CV control via recipe
680:         'PID consts defined
681:         'MTL consts defined, set hfMTLConstsDefinedZx
682:         'segments defined correctly
683:         'find last defined segment
684:         'set CycleAbort Temperature SP
685:         'wrap up RampYN boolean for each segment into one zone integer
686:         'display any errors to operator
687: //Called from Globals RecipeSave( ), and RecipeLoadToPLC( )
688: Function CheckRecipe( )
689:     $CheckRecipeFunc 'marker for IWS Xref tool
690:     $myText = "CheckRecipe( )"
691:     Dim i, j, k, ControlModeError, PIDConstsError, MTLConstsError(4), SetPointsError, msg, PrhtMoldLimitError(4) ', NumberOfZones
692:     Dim ArrMaxZeroOne(4), ArrMaxTwoNineteen(4), MaxZeroOne, MaxTwoNineteen
693:     'reset tags
694:     ControlModeError = 0 : PIDConstsError = 0 : SetPointsError = 0
695:     Erase MTLConstsError : Erase ArrMaxZeroOne : Erase ArrMaxTwoNineteen : Erase PrhtMoldLimitError
696:     $CheckRecipeErrorMessage = "" : msg = ""
697:
698:     'For Loop to check that all zones have either PID|MTL control
699:     For i=1 To $hn2_4zoneMachineSwitch ' NumberOfZones
700:       If $hnControlModeZ[i] = $ControlValve Then
701:         'set control to Fluid Const Temper
702:         $hnControlModeZ[i] = $FluidConstantTemper
703:         $LogEvent 'marker for IWS xref tool to find all $Trace() and $SendEvent() log actions
704:         $SendEvent( "Zone " & i & " must be controlled by FluidConstTemper, FluidRecipe, MoldConstTemper, or MoldRecipe. FluidConstTemper was
selected for you. [CheckRecipe()" )
705:         $Trace( "Zone " & i & " must be controlled by FluidConstTemper, FluidRecipe, MoldConstTemper, or MoldRecipe. FluidConstTemper was
selected for you. [CheckRecipe()" )
```

```
706:         msg = msg & vbCrLf & "Z" & i & " must be controlled by FluidConstTemper, FluidRecipe, MoldConstTemper, or MoldRecipe. FluidConstTemper
was selected for you."
707:         ControlModeError = 1
708:     End If
709: Next
710:
711: 'For Loop to check PID/MTL consts
712: For i=1 To $hn2_4zoneMachineSwitch 'NumberOfZones
713:
714:     'check PID consts for zones
715:     If $REC_PID[i].Kp <= 0 Then
716:         PIDConstsError = 1
717:         'changed to not automatically set to the default, but alert the operator that this has to be set
718:         $LogEvent 'marker for IWS xref tool to find all $Trace() and $SendEvent() log actions
719:         $SendEvent( "Zone " & i & " Proportional Gain Constant (Kp) was not defined, or was a negative value. [CheckRecipe()]" )
720:         $Trace( "Zone " & i & " Proportional Gain Constant (Kp) was not defined, or was a negative value. [CheckRecipe()]" )
721:         msg = msg & vbCrLf & "Z" & i & " Gain Kp was not defined, or was a negative value."
722:     End If
723:     If $REC_PID[i].Ki < 0 Then 'check for erroneous negative sign
724:         PIDConstsError = 1
725:         $REC_PID[i].Ki = $Abs( $REC_PID[i].Ki )
726:         $LogEvent 'marker for IWS xref tool to find all $Trace() and $SendEvent() log actions
727:         $SendEvent( "Zone " & i & " Integral Gain Constant (Ki) was negative, changed to positive. [CheckRecipe()]" )
728:         $Trace( "Zone " & i & " Integral Gain Constant (Ki) was negative, changed to positive. [CheckRecipe()]" )
729:         msg = msg & vbCrLf & "Z" & i & " Gain Ki was negative, changed to positive."
730:     End If
731:     If $REC_PID[i].Kd < 0 Then
732:         PIDConstsError = 1
733:         $REC_PID[i].Kd = $Abs( $REC_PID[i].Kd ) 'check for erroneous negative sign
734:         $LogEvent 'marker for IWS xref tool to find all $Trace() and $SendEvent() log actions
735:         $SendEvent( "Zone " & i & " Derivative Gain Constant (Kd) was negative, changed to positive. [CheckRecipe()]" )
736:         $Trace( "Zone " & i & " Derivative Gain Constant (Kd) was negative, changed to positive. [CheckRecipe()]" )
737:         msg = msg & vbCrLf & "Z" & i & " Gain Kd was negative, changed to positive."
738:     End If
739:
740:     'check MTL consts, if MTL control defined
741:     'MoldConstantTemper(=3) or MoldRecipe(=4) control, so need MTL consts
742:     If $hnControlModeZ[i] >= $MoldConstantTemper And $REC_MTL[i].Kprime <= 0 Then
743:         $REC_MTL[i].Kprime = 0.33 'this value should be the same as the "Startup" value
744:         $LogEvent 'marker for IWS xref tool to find all $Trace() and $SendEvent() log actions
745:         $SendEvent( "Zone " & i & " Entered Gain Constant (K') was not defined, using default of 0.33. [CheckRecipe()]" )
746:         $Trace( "Zone " & i & " Entered Gain Constant (K') was not defined, using default of 0.33. [CheckRecipe()]" )
```

```
747:     msg = msg & vbCrLf & "Z" & i & " Gain K' was not defined, using default of 0.33."
748:     'don't set MTLConstsError flag for this miss
749: End If
750: If $hnControlModeZ[i] >= $MoldConstantTemper And $REC_MTL[i].Km <= 0 Then
751:     $LogEvent 'marker for IWS xref tool to find all $Trace() and $SendEvent() log actions
752:     $SendEvent( "Zone " & i & " Steady State Gain (Km) was not defined. [CheckRecipe()]" )
753:     $Trace( "Zone " & i & " Steady State Gain (Km) was not defined. [CheckRecipe()]" )
754:     msg = msg & vbCrLf & "Z" & i & " Gain Km was not defined."
755:     MTLConstsError(i) = 1
756: End If
757: If $hnControlModeZ[i] >= $MoldConstantTemper And $REC_MTL[i].Theta <= 0 Then
758:     $LogEvent 'marker for IWS xref tool to find all $Trace() and $SendEvent() log actions
759:     $SendEvent( "Zone " & i & " Dead Time (theta) was not defined. [CheckRecipe()]" )
760:     $Trace( "Zone " & i & " Dead Time (theta) was not defined. [CheckRecipe()]" )
761:     msg = msg & vbCrLf & "Z" & i & " Parameter Theta was not defined."
762:     MTLConstsError(i) = 1
763: End If
764: If $hnControlModeZ[i] >= $MoldConstantTemper And $REC_MTL[i].Tau <= 0 Then
765:     $LogEvent 'marker for IWS xref tool to find all $Trace() and $SendEvent() log actions
766:     $SendEvent( "Zone " & i & " Time Constant (tau) was not defined. [CheckRecipe()]" )
767:     $Trace( "Zone " & i & " Time Constant (tau) was not defined. [CheckRecipe()]" )
768:     msg = msg & vbCrLf & "Z" & i & " Paramater Tau was not defined."
769:     MTLConstsError(i) = 1
770: End If
771: If $hnControlModeZ[i] >= $MoldConstantTemper And $REC_MTL[i].Lambda <= 0 Then
772:     $LogEvent 'marker for IWS xref tool to find all $Trace() and $SendEvent() log actions
773:     $SendEvent( "Zone " & i & " Closed Loop Time Constant (lambda) was not defined. [CheckRecipe()]" )
774:     $Trace( "Zone " & i & " Closed Loop Time Constant (lambda) was not defined. [CheckRecipe()]" )
775:     msg = msg & vbCrLf & "Z" & i & " Parameter Lambda was not defined."
776:     MTLConstsError(i) = 1
777: End If
778:
779: 'reset bit which is part of condition that allows state transition to MTLAlgCalc
780: If MTLConstsError(i) Then
781:     $REC_MTL[i].ConstsDefined = 0
782:     $LogEvent 'marker for IWS xref tool to find all $Trace() and $SendEvent() log actions
783:     $SendEvent( " Zone " & i & " MTL parameters must be defined; or run the MTL Tuning Algorithm to establish. [CheckRecipe()]" )
784:     $Trace( " Zone " & i & " MTL parameters must be defined; or run the MTL Tuning Algorithm to establish. [CheckRecipe()]" )
785:     msg = msg & vbCrLf & " Z" & i & " MTL parameters must be defined; or run the MTL Tuning Algorithm to establish."
786: End If
787: If MTLConstsError(i)=0 And $hnControlModeZ[i] >= $MoldConstantTemper Then
788:     $REC_MTL[i].ConstsDefined = 1
```

```
789:     Else
790:         $REC_MTL[i].ConstsDefined = 0
791:     End If
792:
793: Next      'end of For Loop to check PID/MTL consts
794:
795: //call function on Global to find last defined segment
796: If FindSegLastZx = 1 Then      'return of 1 means error
797:     $FindSegLastZxFunc      'marker for IWS Xref tool
798:     SetPointsError = 1
799:     msg = msg & $FindSegLastZxErrMsg
800: End If
801:
802: //calculate the Estimated Cycle Time
803: If Not SetPointsError Then
804:     'put sum of all zone's preheat time Segs (Zero and One) into an array
805:     For i=1 To $hn2_4zoneMachineSwitch
806:         ArrMaxZeroOne(i) = $REC[ i*20-20 ].DwellTimeSP + $REC[ i*20-20+1 ].DwellTimeSP
807:     Next
808:     'find the max value of the array
809:     MaxZeroOne = $Max( 1, ArrMaxZeroOne(1), ArrMaxZeroOne(2), ArrMaxZeroOne(3), ArrMaxZeroOne(4) )
810:     'BUG IN MAX FUNCTION.  If the first numValue happens to be 0, the return is 0, and it doesn't find the max;
811:     ' so place an arbitrary numValue of 1 second in the first value
812:     'put sum of all zone's run time Segs (2 thru 19) into an array
813:     For i=1 To $hn2_4zoneMachineSwitch
814:         For j=(i*20-20+2) To (i*20-20+$hnSegLastZ[ i ])      'Seg2 to SegLast
815:             ArrMaxTwoNineteen(i) = ArrMaxTwoNineteen(i) + $REC[ j ].DwellTimeSP
816:         Next
817:     Next
818:     MaxTwoNineteen = $Max( 1, ArrMaxTwoNineteen(1), ArrMaxTwoNineteen(2), ArrMaxTwoNineteen(3), ArrMaxTwoNineteen(4) )
819:     'the cycle time should be the longest preheat plus the longest run segments
820:     $CycleElapsedTimeEst = MaxZeroOne + MaxTwoNineteen
821: End If      'est cyc time
822:
823: //establish the Cycle Abort Temperature Set Point
824: For i=1 To $hn2_4zoneMachineSwitch      'NumberOfZones
825:     If ( $hnControlModeZ[i] = $FluidRecipe Or $hnControlModeZ[i] = $MoldRecipe ) And $hfSPTemperCycleAbortZ[i] = 0 And SetPointsError = 0 Then
826:         'If running recipe AND check box not hit to specify CATemperSP AND no errors, then set CATemperSP to last defined segment's SP
827:         $hnSPTemperCycleAbortZ[i] = $REC[ $hnSegLastZ[i]+i*20-20 ].TemperSP
828:         'must add zone start index (i*20-20) to SegLast to find place in 80 element REC array
829:         ' MsgBox "for debugging:" & vbCrLf _
830:         '         & "i=" & i & vbCrLf _
```



```

831:      & "$hnSegLastZ[i] = " & $hnSegLastZ[i] & vbCrLf _
832:      & "$hnSegLastZ[i]+i*20-20 = " & $hnSegLastZ[i]+i*20-20 & vbCrLf _
833:      & "$REC[ $hnSegLastZ[i]+i*20-20 ].TemperSP = " & $REC[ $hnSegLastZ[i]+i*20-20 ].TemperSP & vbCrLf _
834:      & "$hnSPTemperCycleAbortZ[i] = " & $hnSPTemperCycleAbortZ[i]
835:      Else If ( $hnControlModeZ[i] = $FluidRecipe Or $hnControlModeZ[i] = $MoldRecipe ) And $hfSPTemperCycleAbortZ[i] = 0 And SetPointsError = 1
Then
836:          'if running recipe AND check box not hit AND had error defining SPs, set equal to last defined temporarily
837:          $hnSPTemperCycleAbortZ[i] = $REC[ $hnSegLastZ[i]+i*20-20 ].TemperSP
838:      Else If ( $hnControlModeZ[i] = $FluidConstantTemper Or $hnControlModeZ[i] = $MoldConstantTemper ) Then
839:          'set equal to const temper SP always; state set should not have to use this SP, but populate with this value anyway
840:          $hnSPTemperCycleAbortZ[i] = $hnSPTemperConstZ[i]
841:      Else
842:          'do nothing, SP should already have been specified by user because should be that $hfSPTemperCycleAbortZ[i] = 1
843:          'therefore will used specified $hnSPTemperCycleAbortZ[i]
844:          End If //second else if
845:      End If //first else if
846:  End If
847: Next
848:
849:  //wrap up RampYN boolean for each segment into one zone integer, integers are 32 bit, only using 20
850:  //bit B0=1 if $REC[0].RampYN = 1 (seg0), all the way through B19 for segment 19 of the zone
851:  'start by zeroing out old info, THIS IS A NECESSARY STEP, otherwise the old setBitNumbers will remain
852:  $hnRampYNZ[1] = 0 : $hnRampYNZ[2] = 0 : $hnRampYNZ[3] = 0 : $hnRampYNZ[4] = 0 : $hnRampYNZ[0] = 0
853:  For i=1 To $hn2_4zoneMachineSwitch 'NumberOfZones
854:      For j=( i*20-20 ) To ( i*20-1 ) 'Zone1 is Seg0-19, Zone2 is Seg20-39, etc
855:          $myText = "zone i=" & i & ", index j=" & j 'for debugging
856:          If $REC[j].RampYN Then
857:              ' j counter and setBitNumber have to be in the range of 0 to 19
858:              If j >= 0 And j < 20 Then $setBitNumber = j ':MsgBox "setBitNumber = j ->" & $setBitNumber
859:              If j >= 20 And j < 40 Then $setBitNumber = j - 20 ':MsgBox "setBitNumber = j-20 ->" & $setBitNumber
860:              If j >= 40 And j < 60 Then $setBitNumber = j - 40 ':MsgBox "setBitNumber = j-40 ->" & $setBitNumber
861:              If j >= 60 Then $setBitNumber = j - 60 ':MsgBox "setBitNumber = j-60 ->" & $setBitNumber
862:              'store accumulated bits in integer tag for each zone
863:              $SetBit( "$hnRampYNZ[" & i & "]", $setBitNumber )
864:          End If
865:      Next
866:  Next
867:
868:  'set function return value that recipe is not defined correctly if errors exist
869:  If ControlModeError Or SetPointsError Or PIDConstsError Or MTLConstsError(1) _
870:  Or MTLConstsError(2) Or MTLConstsError(3) Or MTLConstsError(4) _
871:  Or PrhtMoldLimitError(1) Or PrhtMoldLimitError(2) Or PrhtMoldLimitError(3) Or PrhtMoldLimitError(4) Then 'any other ORs to put in here?

```

```
872: ' MsgBox "debug msg: value of MTLConstsError(i)= " & MTLConstsError(0) & ", " & MTLConstsError(1) & _
873: '      ", " & MTLConstsError(2) & ", " & MTLConstsError(3) & ", " & MTLConstsError(4) & vbCrLf _
874: '      & "value of ControlModeError =" & ControlModeError & vbCrLf _
875: '      & "value of SetPointsError =" & SetPointsError
876: 'set function name equal to fault value
877: CheckRecipe = 1
878: $LogEvent 'marker for IWS xref tool to find all $Trace() and $SendEvent() log actions
879: $SendEvent( "The Recipe was not configured properly; therefore, it was not Saved or Started. [CheckRecipe()]" )
880: $Trace( "The Recipe was not configured properly; therefore, it was not Saved or Started. [CheckRecipe()]" )
881: msg = "The Recipe was not configured properly; therefore, it was not Saved or Started." & vbCrLf _
882:      & "These errors must be fixed before a Recipe can be run." _
883:      & " Errors are also listed in the Event Viewer for reference." & vbCrLf _
884:      & msg & vbCrLf & vbCrLf & "End of Error List."
885: 'display accumulated ErrorMessage on popup
886: $CheckRecipeErrorMessage = msg
887: $AlarmWarningMessage = 2
888: $Open( "AlarmWarning Popup.scr" )
889: Else
890: 'set function name equal to OK value
891: CheckRecipe = 0
892: End If
893:
894: $myText = "CheckRecipe done"
895: 'MsgBox "leaving CheckRecipe( ) function" & vbCrLf & "value of CheckRecipe=" & CheckRecipe
896: End Function 'CheckRecipe( )
897: //////////////////////////////////////
898:
899:
900: //////////////////////////////////////
901: ////////////////////////////////////// FindSegLastZx( ) //////////////////////////////////////
902: //////////////////////////////////////
903: '/REV 3.1.0/ moved out of CheckRecipe(). Code to find the last defined segment for each zone's recipe
904: '//the last defined segment must be the segment where Time>0 going backwards from Seg19 to Seg2; check Seg0 and Seg1 seperately
905: '//this means that you could have segments of 0 time; example Seg2Time=300, Seg3Time=0, Seg4Time=600;
906: '// this recipe will proceed thru Seg2,3,4 Temper setpoints also, maybe causing a valve swing if Seg3's Temper is different than 2 and 4
907: '//function return of 0 means good, return of 1 means errors found
908: '//Called from ScreenScript on "Recipe Editor-Segments.scr"
909: '//Called from Task Script 3 - Ramp Rate Calculation
910: '//Called from Globals CheckRecipe( ), and RecipeCMCT( )
911: Function FindSegLastZx( )
912:     $FindSegLastZxFunc 'marker for IWS Xref tool
913:
```

```

914:     Dim i, j, msg, FSLErr
915:     $FindSegLastZxErrMsg = ""      :      msg = ""
916:
917:     //For Loop to find SegLast for each zone
918:     For i=1 To $hn2_4zoneMachineSwitch 'NumberOfZones
919:         For j=( i*20-1 ) To ( i*20-20 + 2 ) Step -1      'Zone1 is Seg0-19, Zone2 is Seg20-39, etc
920:             'MsgBox "for debugging:" & vbCrLf & "i, Zone# = " & i & vbCrLf & "j, Seg# = " & j & vbCrLf & "TemperSP = " & $REC[j].TemperSP & vbCrLf &
"TimeSP = " & $REC[j].DwellTimeSP
921:             'the last defined segment must be the segment where Time>0 going backwards from Seg19 to Seg2; check Seg0 and Seg1 seperately
922:             If ( $hnControlModeZ[i] = $FluidRecipe Or $hnControlModeZ[i] = $MoldRecipe ) Then
923:                 If $REC[j].DwellTimeSP > 0 Then
924:                     $hnSegLastZ[i] = $Mod( j, 20 )
925:                     Exit For      'move on to next zone
926:                 Else
927:                     'If get all the way to seg2 and seg2 time=0, then error
928:                     If $Mod( j, 20 ) = 2 And $REC[j].DwellTimeSP = 0 Then
929:                         $hnSegLastZ[i] = 0
930:                         $LogEvent      'marker for IWS xref tool to find all $Trace() and $SendEvent() log actions
931:                         $SendEvent( "Zone " & i & " segment 2 Time SP must be greater than 0.  Segment 2 must be defined at a minimum, if recipe profile
control is desired. [FindSegLastZx( )]" )
932:                         $Trace( "Zone " & i & " segment 2 Time SP must be greater than 0.  Segment 2 must be defined at a minimum, if recipe profile
control is desired. [FindSegLastZx( )]" )
933:                         msg = msg & vbCrLf & "Z" & i & " segment 2 Time SP must be greater than 0.  Segment 2 must be defined at a minimum, if recipe
profile control is desired."
934:                         'set error flag (don't need to keep track of each zone's fault, cause will still run function regardless once I get a fault anyway)
935:                         FSLErr = 1
936:                     Else
937:                         'catch all, should be impossible to get here
938:                         $hnSegLastZ[i] = 0
939:                     End If
940:                 End If
941:             Else
942:                 If ( $hnControlModeZ[i] = $FluidConstantTemper Or $hnControlModeZ[i] = $MoldConstantTemper ) Then
943:                     If $hnSPTemperConstZ[i] < $hnSPTemperConstZ[]->Min Then
944:                         'check to make sure a SP has been input, if not, set to MIN
945:                         'setting to MIN will ensure that cold fluid is always flowing through the zone and CV will be 0%
946:                         'should be improbable to get here, maybe not impossible
947:                         $hnSPTemperConstZ[i] = $hnSPTemperConstZ[]->Min
948:                         $LogEvent      'marker for IWS xref tool to find all $Trace() and $SendEvent() log actions
949:                         $SendEvent( "Zone " & i & " Constant Temperature Set Point should be specified.  Using the MIN of " & $hnSPTemperConstZ[]->Min & "
degrees F. [FindSegLastZx( )]" )
950:                         $Trace( "Zone " & i & " Constant Temperature Set Point should be specified.  Using the MIN of " & $hnSPTemperConstZ[]->Min & "
degrees F. [FindSegLastZx( )]" )

```

```
951:         msg = msg & vbCrLf & "Z" & i & " Constant Temperature Set Point should be specified. Using the MIN of " & $hnSPTemperConstZ[]->Min
& " degF."
952:         'set error flag (don't need to keep track of each zone's fault, cause will still run function regardless once I get a fault anyway)
953:         FSLerr = 1
954:         'refer to $RecConfirmDlg = "Constant Temper Zone", when selecting const temper, Seg0,1,2 will be setup automatically
955:         $hnSegLastZ[i] = 2
956:         Exit For 'move on to next zone
957:     Else 'const temper zone must be setup correctly; therefore by definition SegLast=2
958:         $hnSegLastZ[i] = 2
959:         Exit For 'move on to next zone
960:     End If
961: Else 'This is the catch all catch all
962:     $hnSegLastZ[i] = 0
963: End If
964: End If
965: Next
966: Next
967:
968: //Seg0 and Seg1 can have DwellSP=0, but check to make sure that TemperSPs are defined
969: For i=1 To $hn2_4zoneMachineSwitch 'NumberOfZones
970:     For j=( i*20-20 ) To ( i*20-19 ) 'Zone1 is Seg0-19, Zone2 is Seg20-39, etc
971:         If $REC[j].TemperSP < $REC[j].TemperSP->Min Then
972:             $hnSegLastZ[i] = 0
973:             $LogEvent 'marker for IWS xref tool to find all $Trace() and $SendEvent() log actions
974:             $SendEvent( "Zone " & i & " segment "& $Mod( j, 20 ) & " Temper SP must be greater than or equal to "& $REC[j].TemperSP->Min & ", if
recipe profile control is desired. [FindSegLastZx( )]" )
975:             $Trace( "Zone " & i & " segment "& $Mod( j, 20 ) & " Temper SP must be greater than or equal to "& $REC[j].TemperSP->Min & ", if recipe
profile control is desired. [FindSegLastZx( )]" )
976:             msg = msg & vbCrLf & "Z" & i & " segment "& $Mod( j, 20 ) & " Temper SP must be greater than or equal to "& $REC[j].TemperSP->Min & ", if
recipe profile control is desired."
977:             'set error flag (don't need to keep track of each zone's fault, cause will still run function regardless once I get a fault anyway)
978:             FSLerr = 1
979:             End If
980:         Next
981:     Next
982:
983: //set function return
984: If FSLerr Then
985:     'set function return to fault value and export msg
986:     FindSegLastZx = 1
987:     $FindSegLastZxErrMsg = msg
988: Else
```

```

989:      'set function return to OK value and clear msg
990:      FindSegLastZx = 0
991:      $FindSegLastZxErrMsg = ""
992: End If
993:
994: End Function 'FindSegLastZx( )
995: //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
996:
997:
998: //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
999: //////////////////////////////////////////////////////////////////          RecipeDownload( )          //////////////////////////////////////////////////////////////////
1000: //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
1001: //Code to Transfer Recipe Data to PLC and Start Recipe
1002:     '//transferring ~350 tags all at once takes just over 2 minutes and was inconsistent, seemed to use polling method,
1003:     'which means xfer would start anywhere within the tag range; therefore,
1004:     'the following handshaking routine was developed to guarantee synchronization, chrono-xfer, and integrity of data
1005: //Called from Global RecipeLoadToPLC( )
1006: '/REV 3.1.0/ changed name from StartRecipe() to RecipeDownload()
1007: '/REV 3.1.0/ changed xfer tag assignments so that there is only one set of assignments, which is now only inside the DO loop,
1008: //    instead of the first set of assignments outside the DO loop, then the remaining assignments inside the DO loop
1009: Function RecipeDownload( )      'previously named StartRecipe( )
1010:     $RecipeDownloadFunc 'marker for IWS Xref tool
1011:     $myText = "RecipeDownload( )"
1012:
1013:     Dim ManPrhtTemp, ZoneSwitch, StartTime, StartMsecValue, msg, ControlLoopCount, i
1014:
1015:     'set zone and preheat choices, this temp assignment is necessary for HMI to overwrite PLC tags a few lines down
1016:     ' after OPC sheet is enabled. If when OPC sheet is enabled it may write the PLC value to the HMI, the Force() will
1017:     ' ensure that both the HMI and PLC will be synced.
1018:     ManPrhtTemp = $hfManualPreheat
1019:     ZoneSwitch = $hn2_4zoneMachineSwitch
1020:
1021:     'enable OPC worksheet to xfer segment variables
1022:     '!!! Place OPC enable before setting tags so that HMI will overwrite anything in the pipeline
1023:     'if OPC enable was after setting tags, TND OPC server would overwrite HMI tags when that wouldn't be desirable!!!
1024:     $OPCCommRecipeXfer = 0
1025:
1026:     '/REV 3.1.0/ set starting value of PLC counter after OPC sheet enabled
1027:     StartMsecValue = $PLCRollingMsecCounter
1028:     $DoLoopTimer = 0 :   StartTime = $GetTickCount()      'IWS internal clock in milliseconds
1029: Do 'wait for XX milliseconds or until OPC sees a change in its counter
1030:     'if see a change in counter, then know that OPC is updating; otherwise timer will time out, signifying a lack of comm

```

TND_v3.0.1

Showing only the transfer of Segment information; the transfer of Constant information is essentially identical

```
1031:     $DoLoopTimer = $Abs( $GetTickCount() - StartTime )
1032: '   $myText = "OPCcomm "& $DoLoopTimer
1033: Loop Until ( $DoLoopTimer >= 10000 ) Or ( $PLCRollingMsecCounter <> StartMsecValue ) Or $hfCancelDownload Or $fCycleAbort
1034:
1035: ' /REV 3.1.0/ if timed out, then means OPC comm is gone, so terminate xfer
1036: If $DoLoopTimer >= 10000 Then
1037:     msg = "OPC communications was not achieved and has timed out.      " & vbCrLf _
1038:         & "The timer is set to wait 10 seconds for a good connection to be established." & vbCrLf _
1039:         & "Please try again." & vbCrLf _
1040:         & vbCrLf & vbCrLf & "Error Location: RecipeDownload( )"
1041:     'display ErrorMessage on popup
1042:     $RecipeDownloadErrorMessage = msg
1043:     $AlarmWarningMessage = 5
1044:     $Open( "AlarmWarning Popup.scr" )
1045:     $OPCCommRecipeXfer = 1      'disable OPC worksheet
1046:     RecipeDownload=1          'set to error condition
1047:     Exit Function
1048: End If
1049:
1050: $DoLoopTimer = 0      'reset to populate text dynamic on "Wait Dialog.scr" bar graph
1051: ' /REV 3.1.0/ initialize OPC variables with HMI values, to make sure are reset; use Force to make sure OPC sees a change even if the value is
the same
1052: 'need to put tags here for IWS Xref, since the Xref doesn't find the tags within the Force function
1053: $hFRXHandshake : $cRXSegNumber : $cRXZoneNumber : $fRXSegDataStored : $fRXConDataStored
1054: $hfManualPreheat : $hn2_4zoneMachineSwitch : $hwRecipeName : $hfStartRecipe
1055: $ForceTagChange( "hFRXHandshake", 0 )
1056: $ForceTagChange( "cRXSegNumber", 0 )      'imperitive, so also initialize on PLC
1057: $ForceTagChange( "cRXZoneNumber", 1 )      'imperitive, so also initialize on PLC
1058: $ForceTagChange( "fRXSegDataStored", 0 )      ' /REV 3.1.0/ if dnld was canceled the value left may have been =1, can't rely on PLC resetting
before ripping thru the HMI code, so reset here
1059: $ForceTagChange( "fRXConDataStored", 0 )      ' /REV 3.1.0/ same as fRXSegDataStored
1060: $ForceTagChange( "hfManualPreheat", ManPrhtTemp )
1061: $ForceTagChange( "hn2_4zoneMachineSwitch", ZoneSwitch )
1062: $ForceTagChange( "hwRecipeName", $RecipeName )      'is on OPC sheet 52, every 2 seconds
1063:
1064: '-----transfer SEGMENT values-----
1065: Do      'Loop until all segments downloaded
1066:
1067:     If $cRXSegNumber < ($REC->Size*($hn2_4zoneMachineSwitch/4)-1) Then
1068:         'for 2-zone $REC->Size*($hn2_4zoneMachineSwitch/4)-1 = 80*2/4-1= 39 total segments
1069:         $hnRXData_DwellTimeSP_0 = $REC[$cRXSegNumber +0].DwellTimeSP
1070:         $hnRXData_TemperSP_0 = $REC[$cRXSegNumber +0].TemperSP
```

```
1071:      $hnRXData_DwellTimeSP_1 = $REC[$cRXSegNumber +1].DwellTimeSP
1072:      $hnRXData_TemperSP_1 = $REC[$cRXSegNumber +1].TemperSP
1073:      $hnRXData_DwellTimeSP_2 = $REC[$cRXSegNumber +2].DwellTimeSP
1074:      $hnRXData_TemperSP_2 = $REC[$cRXSegNumber +2].TemperSP
1075:      $hnRXData_DwellTimeSP_3 = $REC[$cRXSegNumber +3].DwellTimeSP
1076:      $hnRXData_TemperSP_3 = $REC[$cRXSegNumber +3].TemperSP
1077:      $hnRXData_DwellTimeSP_4 = $REC[$cRXSegNumber +4].DwellTimeSP
1078:      $hnRXData_TemperSP_4 = $REC[$cRXSegNumber +4].TemperSP
1079:      $hnRXData_DwellTimeSP_5 = $REC[$cRXSegNumber +5].DwellTimeSP
1080:      $hnRXData_TemperSP_5 = $REC[$cRXSegNumber +5].TemperSP
1081:      $hnRXData_DwellTimeSP_6 = $REC[$cRXSegNumber +6].DwellTimeSP
1082:      $hnRXData_TemperSP_6 = $REC[$cRXSegNumber +6].TemperSP
1083:      $hnRXData_DwellTimeSP_7 = $REC[$cRXSegNumber +7].DwellTimeSP
1084:      $hnRXData_TemperSP_7 = $REC[$cRXSegNumber +7].TemperSP
1085:      $hnRXData_DwellTimeSP_8 = $REC[$cRXSegNumber +8].DwellTimeSP
1086:      $hnRXData_TemperSP_8 = $REC[$cRXSegNumber +8].TemperSP
1087:      $hnRXData_DwellTimeSP_9 = $REC[$cRXSegNumber +9].DwellTimeSP
1088:      $hnRXData_TemperSP_9 = $REC[$cRXSegNumber +9].TemperSP
1089:
1090:      'convert to percent up to 50% using SegNum up to #segs, then remaining 50% using ZoneNum up to #zones
1091:      $WaitProgress = ($cRXSegNumber / ($REC->Size*($hn2_4zoneMachineSwitch/4)-1))*100/2
1092:
1093:      If $cRXSegNumber=0 Then
1094:          '/REV 3.1.0/ moved hfStartRecipe to occur after above assignments have been made
1095:          '/REV 3.1.0 con't/ this will give time for HMI to write to OPC and have PLC pick off OPC updates before making the PLC state
1096:          '/REV 3.1.0 con't/ changes to the downloading sequence
1097:          'set flag that button was pushed on HMI to Load/Start the recipe and enable state transition, tag sent via OPC to PLC
1098:          ' need to force here cause if dnld cancelled, HMI=1, but PLC=0. when dnld again, HMI writes 1, but is already 1, so OPC doesn't see
a change
1099:          $ForceTagChange( "hfStartRecipe", 1 )      'this should be the only place this flag is turned ON, this triggers the PLC state changes
to enable downloading
1100:          End If
1101:
1102:      End If
1103:
1104:      $hFRXHandshake = 1      'set handshake so PLC has the OK to assign above set of data
1105:
1106:      Do Until $fRXSegDataStored = 1 'waiting for PLC to store this set of Segment data
1107:          $myText= "loop SegStrd=1"
1108:          If $PLCRollingMsecCounter->Quality<>192 Or $hfCancelDownload Or $fCycleAbort Then
1109:              '/REV 3.1.0/ was previously looking at ScanCount->Quality, now specifically look at the quality of a tag on OPC worksheet used in
this script
```



```
1110:         $myText = "xF@fseg1" : RecipeDownload=1 : $OPCCommRecipeXfer = 1 : Exit Function
1111:     End If
1112: Loop
1113:
1114:     $hfRXHandshake = 0      'once HMI sees fRXSegDataStored, then reset handshake so PLC can loop back
1115:
1116:     Do Until $fRXSegDataStored = 0 'waiting for PLC to clear fRXSegDataStored, saying 'Ready for next set'
1117:     $myText= "loop SegStrd=0"
1118:         If $PLCRollingMsecCounter->Quality<>192 Or $hfCancelDownload Or $fCycleAbort Then
1119:             $myText = "xF@fseg0" : RecipeDownload=1 : $OPCCommRecipeXfer = 1 : Exit Function
1120:         End If
1121:     Loop
1122:
1123: Loop Until ( $cRXSegNumber >= ($REC->Size*($hn2_4zoneMachineSwitch/4)-1) ) Or $hfCancelDownload Or $fCycleAbort
1124: '-----END of transfer SEGMENT values-----
1125:
1126:
1127: '-----transfer CONTROL values-----
1128: 'need this variable cause cRXZoneNumber is incremented to by PLC at the bottom of this loop, but need the loop to return
1129: ' back to the top to assign the last zone's tags, then xfer, then exit out of loop
1130: ' so ControlLoopCount lags the value of cRXZoneNumber by 1
1131: ControlLoopCount = 0
1132: Do 'Loop until all zones downloaded
1133:
1134:     If ControlLoopCount < $hn2_4zoneMachineSwitch Then
1135:         ControlLoopCount = ControlLoopCount + 1
1136:
1137:         'set up RampYesNo REC class tag booleans to encoded integer; encoded in CheckRecipe()
1138:         $hnRXData_RampYN = $hnRampYNZ[$cRXZoneNumber]
1139:         'set up the remaining data transfers
1140:         $hfRXData_ConstsDefined = $REC_MTL[$cRXZoneNumber].ConstsDefined
1141:         $hfRXData_MoldTemperLimits = $hfMoldTemperLimitsZ[$cRXZoneNumber]
1142:         $hnRXData_MoldTemperLimitUpper = $hnMoldTemperLimitUpperZ[$cRXZoneNumber]
1143:         $hnRXData_MoldTemperLimitLower = $hnMoldTemperLimitLowerZ[$cRXZoneNumber]
1144:         $hnRXData_SPTemperCycleAbort = $hnSPTemperCycleAbortZ[$cRXZoneNumber]
1145:         $hnRXData_SPTemperConst = $hnSPTemperConstZ[$cRXZoneNumber]
1146:         $hnRXData_SegLast = $hnSegLastZ[$cRXZoneNumber]
1147:         $hnRXData_CtlMode = $hnControlModeZ[$cRXZoneNumber]
1148:         $hnRXData_Theta = $REC_MTL[$cRXZoneNumber].Theta
1149:         $hnRXData_Tau = $REC_MTL[$cRXZoneNumber].Tau
1150:         $hnRXData_Lambda = $REC_MTL[$cRXZoneNumber].Lambda
1151:         $hrRXData_Kprime = $REC_MTL[$cRXZoneNumber].Kprime
```

```
1152:      $hrRXData_Km = $REC_MTL[$cRXZoneNumber].Km
1153:      $hrRXData_Kp = $REC_PID[$cRXZoneNumber].Kp
1154:      $hrRXData_Ki = $REC_PID[$cRXZoneNumber].Ki
1155:      $hrRXData_Kd = $REC_PID[$cRXZoneNumber].Kd
1156:
1157:      $WaitProgress = 50 + ($cRXZoneNumber / $hn2_4zoneMachineSwitch)*100/2
1158:  End If
1159:
1160:  $hfRXHandshake = 1      'set handshake
1161:
1162:  Do Until $fRXConDataStored = 1 'waiting for PLC to store this set of Control data
1163:    $myText= "loop ConStrd=1, Z#=" & $cRXZoneNumber
1164:    If $PLCRollingMsecCounter->Quality<>192 Or $hfCancelDownload Or $fCycleAbort Then
1165:      $mytext = "xF@fcon1" : RecipeDownload=1 : $OPCCommRecipeXfer = 1 : Exit Function
1166:    End If
1167:  Loop
1168:
1169:  $hfRXHandshake = 0      'reset handshake
1170:
1171:  Do Until $fRXConDataStored = 0 'waiting for PLC to clear fRXConDataStored, saying 'Ready for next set'
1172:    $myText= "loop ConStrd=0, Z#=" & $cRXZoneNumber
1173:    If $PLCRollingMsecCounter->Quality<>192 Or $hfCancelDownload Or $fCycleAbort Then
1174:      $myText = "xF@fcon0" : RecipeDownload=1 : $OPCCommRecipeXfer = 1 : Exit Function
1175:    End If
1176:  Loop
1177:
1178:  Loop Until ( ControlLoopCount >= $hn2_4zoneMachineSwitch ) Or $hfCancelDownload Or $fCycleAbort
1179:  '-----END of transfer CONTROL values-----
1180:
1181:
1182:  'waiting for PLC to signal that it has a Recipe loaded
1183:  Do Until $sRXPLCLoaded
1184:    If $PLCRollingMsecCounter->Quality<>192 Or $hfCancelDownload Or $fCycleAbort Then
1185:      $myText = "xDoload" : RecipeDownload=1 : $OPCCommRecipeXfer = 1 : Exit Function
1186:    End If
1187:  Loop
1188:
1189:  'turn off OPC comm worksheet because don't need anymore
1190:  $OPCCommRecipeXfer = 1
1191:  'set function return to OK value
1192:  RecipeDownload = 0
1193:
```

```
1194: $myText= "end of RecipeDownload( )"
1195: End Function 'RecipeDownload( )
1196: //////////////////////////////////////////////////
1197:
1198:
1199: //////////////////////////////////////////////////
1200: //////////////////////////////////////////////////      StartMTLTuning( )      //////////////////////////////////////////////////
1201: //////////////////////////////////////////////////
1202: //Code to check MTL Tuning Setup Parameter Data    (incorporate CheckRecipe() concepts)
1203: 'this function is called from the 'Start MTL Tuning' button
1204: 'check Start, End, Fluid temps are set for all zones, or the Exclude button is pushed for zones not tested,
1205: ' and at least one zone has to be tested and have temps defined
1206: ' if setup not ok, msgbox error to notify user to change
1207: ' if setup ok, then enable $hfMTLTuneStart
1208: //Called from "Start Metal Tuning Test for Selected Zones" button on "Metal Tuning.scr"
1209: Function StartMTLTuning( )
1210:     $StartMTLTuningFunc 'marker for IWS Xref
1211:
1212:     Dim i, j, k, count, CountError, TemperSetupError, PIDConstsError, msg
1213:     'reset tags
1214:     count = 0 : CountError = 0 : TemperSetupError = 0 : PIDConstsError = 0
1215:     msg = "The following errors were found in the MTL Tuning setup: " & vbCrLf & vbCrLf
1216:
1217:     For i = 1 To $hn2_4zoneMachineSwitch
1218:         'check to make sure at least one zone is going to be tested
1219:         If $hfMTLTuneNoTestZ[i] Then count = count + 1
1220:     Next
1221:
1222:     If count = $hn2_4zoneMachineSwitch Then
1223:         CountError = 1
1224:         msg = msg & "At least one zone must be tested." & vbCrLf
1225:     End If
1226:
1227:     For i = 1 To $hn2_4zoneMachineSwitch
1228:         If Not $hfMTLTuneNoTestZ[i] And Not CountError Then
1229:             'check the following for zones that are to be tested
1230:             If $hnMoldStartTemperZ[i] = $hnMoldEndTemperZ[i] Then
1231:                 TemperSetupError=1
1232:                 msg = msg & "Specify a Mold Start and End Temperature that are different from each other, Zone " & i & "." & vbCrLf
1233:             End If
1234:             If $hnMoldEndTemperZ[i] = $hnMoldEndTemperZ->Min Then
1235:                 TemperSetupError=1
```

```
1236:         msg = msg & "Specify a Mold End Temperature greater than the minimum input, Zone " & i & "." & vbCrLf
1237:     End If
1238:
1239:     'check PID consts for zones to be tested
1240:     If $PID_MTLTuning[i].Kp <= 0 Then
1241:         PIDConstsError = 1
1242:         msg = msg & "Z" & i & " Gain Kp was not defined, or was a negative value." & vbCrLf
1243:     End If
1244:     If $PID_MTLTuning[i].Ki < 0 Then          'check for erroneous negative sign
1245:         PIDConstsError = 1
1246:         $PID_MTLTuning[i].Ki = $Abs( $PID_MTLTuning[i].Ki )
1247:         msg = msg & "Z" & i & " Gain Ki was negative, changed to positive." & vbCrLf
1248:     End If
1249:     If $PID_MTLTuning[i].Kd < 0 Then
1250:         PIDConstsError = 1
1251:         $PID_MTLTuning[i].Kd = $Abs( $PID_MTLTuning[i].Kd )    'check for erroneous negative sign
1252:         msg = msg & "Z" & i & " Gain Kd was negative, changed to positive." & vbCrLf
1253:     End If
1254: Else 'zone is not going to be tested so check if PID or CV
1255:     If $hnControlModeZ[i]=$ControlValve Or $hnControlModeZ[i]=$FluidConstantTemper Then
1256:         'ok, do nothing
1257:     Else
1258:         'set to CV=0%
1259:         $hnControlModeZ[i]=0
1260:         $hnSPControlValveManualZ[i]=0
1261:     End If
1262: End If
1263: Next
1264:
1265: If TemperSetupError Then
1266:     msg = msg & "The Fluid Start SP should be such that the Mold Temper can be maintained within 3% of the Mold Start Temper." & vbCrLf
1267: End If
1268:
1269: If CountError Or TemperSetupError Or PIDConstsError Then
1270:     'set function name equal to Error value
1271:     StartMTLTuning = 1
1272:     MsgBox msg, vbOKOnly, "MTL Tuning Setup Errors..."
1273:     $MTLTuningErrorMessage = msg
1274:     $AlarmWarningMessage = 6 'set to display $MTLTuningErrorMessage
1275:     $Open( "AlarmWarning Popup.scr" )
1276: Else
1277:     'set function name equal to OK value
```

```
1278:     StartMTLTuning = 0
1279:
1280:     'change Ctl Mode to PID Const Temper for zones to be tested, and set ConstTemper SP
1281:     For i = 1 To $hn2_4zoneMachineSwitch
1282:         If Not $hfMTLTuneNoTestZ[i] Then
1283:             $hnControlModeZ[i] = $FluidConstantTemper
1284:             'set const temper PID SP, so when tuning done, this is the SP that will drive the PID (when done, or aborted)
1285:             $hnSPTemperConstZ[i] = $hnFluidStartTemperZ[i]
1286:             ' was previously setting $hnSPTemperConstZ[i] = 55
1287:         End If
1288:     Next
1289:
1290: End If
1291:
1292: End Function 'StartMTLTuning( )
1293: //////////////////////////////////////
1294:
1295:
1296: //////////////////////////////////////
1297: ////////////////////////////////////// SetMTLConstsDefinedFlag( integer ) //////////////////////////////////////
1298: //////////////////////////////////////
1299: 'routine to check if MTL consts are defined, when running Manual Non-Recipe Const Temper
1300: 'and if so, then set flag that they are so that MTL control will be possible
1301: //Called from "Non-Recipe ConstTemper.scr" when a successful keypad entry is made in either Theta, Tau, Lambda, K', or Km fields
1302: //Called from Globals WriteMTLTuningResults2NRCT( ), and CCMNRCT( )
1303: Function SetMTLConstsDefinedFlag( ZoneX )
1304:     $SetMTLConstsDefinedFlagFunc 'marker for IWS Xref tool
1305:
1306:     If $NonRecipeConstTemper_MTL[ZoneX].Theta > 0 _
1307:         And $NonRecipeConstTemper_MTL[ZoneX].Tau > 0 _
1308:         And $NonRecipeConstTemper_MTL[ZoneX].Lambda > 0 _
1309:         And $NonRecipeConstTemper_MTL[ZoneX].Kprime <> 0 _
1310:         And $NonRecipeConstTemper_MTL[ZoneX].Km <> 0 Then
1311:
1312:         $NonRecipeConstTemper_MTL[ZoneX].ConstsDefined = 1
1313:     Else
1314:         $NonRecipeConstTemper_MTL[ZoneX].ConstsDefined = 0
1315:     End If
1316:
1317: End Function 'SetMTLConstsDefinedFlag( ZoneX )
1318: //////////////////////////////////////
1319:
```

```
1320:
1321: //////////////////////////////////////////////////
1322: ////////////////////////////////// SetPIDConsts( stringDescriptor ) //////////////////////////////////
1323: //////////////////////////////////////////////////
1324: 'Set PID Consts for MTL Tuning
1325: 'set either by from the screen which came from, or one of two buttons on tuning screen
1326: 'Descriptor is a text string to determine which IF statement to use
1327: //Called from "PID Default 1" or "PID Default 2" buttons on "Metal Tuning.scr" or "Non-Recipe ConstTemper.scr" or "Recipe Editor-Constants.scr"
1328: //Called from "Run Metal Tuning Test to Establish Metal Constants" button on "Non-Recipe ConstTemper.scr" or "Recipe Editor-Constants.scr"
1329: Function SetPIDConsts( Descriptor )
1330:     $SetPIDConstsFunc    'marker for IWS Xref tool
1331:     Dim i
1332:     For i=1 To $hn2_4zoneMachineSwitch
1333:         If Descriptor = "REC" Then
1334:             'set PID consts for MTL Tuning when press button for MTL Tuning from Recipe Editor-Constants screen
1335:             $PID_MTLTuning[i].Kp = $REC_PID[i].Kp
1336:             $PID_MTLTuning[i].Ki = $REC_PID[i].Ki
1337:             $PID_MTLTuning[i].Kd = $REC_PID[i].Kd
1338:         End If
1339:         If Descriptor = "NRCT" Then
1340:             'set PID consts for MTL Tuning when press button for MTL Tuning from Non-Recipe ConstTemper screen
1341:             $PID_MTLTuning[i].Kp = $NonRecipeConstTemper_PID[i].Kp
1342:             $PID_MTLTuning[i].Ki = $NonRecipeConstTemper_PID[i].Ki
1343:             $PID_MTLTuning[i].Kd = $NonRecipeConstTemper_PID[i].Kd
1344:         End If
1345:         If Descriptor = "Default1" Then
1346:             'set PID consts to Default 1; historical info from old machines
1347:             'used on MTL Tuning screen button
1348:             $PID_MTLTuning[i].Kp = 1.5
1349:             $PID_MTLTuning[i].Ki = 4.5
1350:             $PID_MTLTuning[i].Kd = 0.05
1351:         End If
1352:         If Descriptor = "Default2" Then
1353:             'set PID consts to Default 2; historical info from old machines
1354:             'used on MTL Tuning screen button
1355:             $PID_MTLTuning[i].Kp = 12.0
1356:             $PID_MTLTuning[i].Ki = 1.5
1357:             $PID_MTLTuning[i].Kd = 0.05
1358:         End If
1359:         If Descriptor = "Default1NRCT" Then
1360:             'set PID consts to Default 1; historical info from old machines
1361:             'used on NRCT screen button
```

```

1362: $NonRecipeConstTemper_PID[i].Kp = 1.5
1363: $NonRecipeConstTemper_PID[i].Ki = 4.5
1364: $NonRecipeConstTemper_PID[i].Kd = 0.05
1365: End If
1366: If Descriptor = "Default2NRCT" Then
1367:     'set PID consts to Default 2; historical info from old machines
1368:     'used on NRCT screen button
1369:     $NonRecipeConstTemper_PID[i].Kp = 12.0
1370:     $NonRecipeConstTemper_PID[i].Ki = 1.5
1371:     $NonRecipeConstTemper_PID[i].Kd = 0.05
1372: End If
1373: If Descriptor = "Default1Rec" Then
1374:     'set PID consts to Default 1; historical info from old machines
1375:     'used on Recipe Editor - Consts screen
1376:     $REC_PID[i].Kp = 1.5
1377:     $REC_PID[i].Ki = 4.5
1378:     $REC_PID[i].Kd = 0.05
1379: End If
1380: If Descriptor = "Default2Rec" Then
1381:     'set PID consts to Default 2; historical info from old machines
1382:     'used on Recipe Editor - Consts screen
1383:     $REC_PID[i].Kp = 12.0
1384:     $REC_PID[i].Ki = 1.5
1385:     $REC_PID[i].Kd = 0.05
1386: End If
1387: Next
1388:
1389: End Function      'SetPIDConsts( Descriptor )
1390: //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
1391:
1392:
1393: //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
1394: //////////////////////////////////////////////////////////////////// WriteMTLTuningResults2recFile( ) //////////////////////////////////////////////////////////////////
1395: //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
1396: //Code tied to button(s) on MTL Tuning screen to write parameter results to an existing recipe file
1397: //Called from "Save results to Open Recipe File "{RecipeName}" ?" button on "Metal Tuning.scr"
1398: //Called from Global RecipeOpen( )
1399: Sub WriteMTLTuningResults2recFile( )
1400:     $WriteMTLTuningResults2recFileSub    'marker for IWS Xref tool
1401: Dim i
1402: 'move results from test into MTL Class tags for specified recipe file
1403: For i = 1 To $hn2_4zoneMachineSwitch

```

```
1404:     If Not $hfMTLTuneNoTestZ[i] Then
1405:         'why go thru the routine if you are not going to use Mold Recipe, so set Ctl Mode to this
1406:         $hnControlModeZ[i] = $MoldRecipe
1407:
1408:         $REC_PID[i].Kp = $PID_MTLTuning[i].Kp
1409:         $REC_PID[i].Ki = $PID_MTLTuning[i].Ki
1410:         $REC_PID[i].Kd = $PID_MTLTuning[i].Kd
1411:
1412:         $REC_MTL[i].Theta = $MTL_TuneResults[i].Theta
1413:         $REC_MTL[i].Tau = $MTL_TuneResults[i].Tau
1414:         $REC_MTL[i].Lambda = $MTL_TuneResults[i].Lambda
1415:         $REC_MTL[i].Kprime = $MTL_TuneResults[i].Kprime
1416:         $REC_MTL[i].Km = $MTL_TuneResults[i].Km
1417:     End If
1418: Next
1419: 'save file; call global, set boolean true to skip "over-write" question
1420: Call RecipeSave( $RecipeName, True )
1421:     $RecipeSaveSub 'marker for IWS Xref tool
1422:
1423: End Sub 'WriteMTLTuningResults2recFile( )
1424: //////////////////////////////////////
1425:
1426:
1427: //////////////////////////////////////
1428: ////////////////////////////////////// WriteMTLTuningResults2txtFile( ) //////////////////////////////////////
1429: //////////////////////////////////////
1430: //Code tied to button on MTL Tuning screen to write paramater results to txt file
1431: //Called from "Save results to Text File ...?" button on "Metal Tuning.scr"
1432: Sub WriteMTLTuningResults2txtFile( )
1433:     $WriteMTLTuningResults2txtFileSub
1434:
1435: Dim strFileName, strWriteText, i, j, k
1436:
1437: 'clear MTRFileName, cause don't want user to over-write the last one that was input
1438: $MTRFileName = ""
1439: 'launch alphanumeric keypad for operator input of new name, and check
1440: $KeypadReturn = $Keypad( "$MTRFileName", "Alphanumeric", 0, "Specify filename to store results.... \\<>\Recipes\MTL_TuneResults\*.txt" )
1441: If $KeypadReturn = 0 Then '0=$ keypad success
1442:     'check name for invalid characters, call function on global
1443:     If CheckInputName( $MTRFileName ) = 1 Then 'function=1 is fault
1444:         $CheckInputNameFunc 'marker for IWS Xref tool
1445:     Exit Sub
```



```
1446:     End If
1447:
1448:     'if check was ok then proceed
1449:     strFileName = $MTRPath & "\" & $MTRFileName & ".txt"
1450:     'start setting up string tag
1451:     strWriteText = "Results of MTL Tuning Test " & vbCrLf
1452:     strWriteText = strWriteText & "This file was saved on " & $Date & " " & $Time & vbCrLf & vbCrLf
1453:     strWriteText = strWriteText & "Warning: Editing this file will cause text alignment problems." & vbCrLf & vbCrLf
1454:     'write data to file for each zone; option zero replaces the contents of the exisiting file
1455:     $FileWrite( strFileName, strWriteText, 0 )
1456:
1457:     For i = 1 To $hn2_4zoneMachineSwitch
1458:         'clear out string to write to file at beginning of each zone to be checked
1459:         strWriteText = ""
1460:         If Not $hfMTLTuneNoTestZ[i] Then
1461:             'start writing data for each zone
1462:             strWriteText = strWriteText & "ZONE "&i&" Results of MTL Tuning Test: " & vbCrLf
1463:             strWriteText = strWriteText & "    Temperature Setup as follows: " & vbCrLf
1464:             strWriteText = strWriteText & "        Mold Start Temperature = " & $hnMoldStartTemperZ[i] & vbCrLf
1465:             strWriteText = strWriteText & "        Mold End Temperature = " & $hnMoldEndTemperZ[i] & vbCrLf
1466:             strWriteText = strWriteText & "        Fluid Start Setpoint = " & $hnFluidStartTemperZ[i] & vbCrLf
1467:             strWriteText = strWriteText & "    PID consts used for the zone:" & vbCrLf
1468:             strWriteText = strWriteText & "        Kp = " & $PID_MTLTuning[i].Kp & vbCrLf
1469:             strWriteText = strWriteText & "        Ki = " & $PID_MTLTuning[i].Ki & vbCrLf
1470:             strWriteText = strWriteText & "        Kd = " & $PID_MTLTuning[i].Kd & vbCrLf
1471:             strWriteText = strWriteText & "    Mold Characteristic Results for the zone:" & vbCrLf
1472:             strWriteText = strWriteText & "        Theta, Dead Time = " & $MTL_TuneResults[i].Theta & vbCrLf
1473:             strWriteText = strWriteText & "        Tau, Time Constant = " & $MTL_TuneResults[i].Tau & vbCrLf
1474:             strWriteText = strWriteText & "        Lambda, Closed Loop Time Constant = " & $MTL_TuneResults[i].Lambda & vbCrLf
1475:             strWriteText = strWriteText & "        K', Entered Gain = " & $MTL_TuneResults[i].Kprime & vbCrLf
1476:             strWriteText = strWriteText & "        Km, Steady State Gain = " & $MTL_TuneResults[i].Km & vbCrLf
1477:             strWriteText = strWriteText & "    Calculated Mold Parameter Results for the zone:" & vbCrLf
1478:             strWriteText = strWriteText & "        Kcs = " & $MTL_TuneResults[i].Kcs & vbCrLf
1479:             strWriteText = strWriteText & "        P0 = " & $MTL_TuneResults[i].P0 & vbCrLf
1480:             strWriteText = strWriteText & "        P1 = " & $MTL_TuneResults[i].P1 & vbCrLf
1481:             strWriteText = strWriteText & "        P5 = " & $MTL_TuneResults[i].P5 & vbCrLf
1482:             strWriteText = strWriteText & "        P8 = " & $MTL_TuneResults[i].P8 & vbCrLf
1483:             strWriteText = strWriteText & "        P9 = " & $MTL_TuneResults[i].P9 & vbCrLf & vbCrLf
1484:
1485:             'write data to file for each zone; option one appends the contents of the exisiting file
1486:             $FileWrite( strFileName, strWriteText, 1 )
1487:         Else
```

```
1488:         'write that zone was not tested
1489:         strWriteText = "ZONE " &i& " was NOT tested." & vbCrLf & vbCrLf
1490:         $FileWrite( strFileName, strWriteText, 1 )
1491:     End If
1492: Next
1493: $FileWrite( strFileName, "End of File", 1 )
1494:
1495: Else '$KeypadReturn<>0
1496:     MsgBox "The file was not saved because a name was not specified.", vbExclamation, "File save error!"
1497:     Exit Sub
1498: End If
1499: End Sub 'WriteMTLTuningResults2txtFile( )
1500: //////////////////////////////////////
1501:
1502:
1503: //////////////////////////////////////
1504: ////////////////////////////////////// WriteMTLTuningResults2NRCT( ) //////////////////////////////////////
1505: //////////////////////////////////////
1506: 'move results from test into tags for NRCT setup
1507: //Called from "Save results to Manual Constant Temper setup?" button on "Metal Tuning.scr"
1508: Sub WriteMTLTuningResults2NRCT( )
1509:     $WriteMTLTuningResults2NRCTSub
1510:     Dim i, Answer, msg
1511:     msg = "This will navigate you to the Manual screen and populate the PID and MTL constants for the zones tested."
1512:     Answer = MsgBox(msg, vbQuestion + vbOKCancel, "Change to Manual Mode?")
1513:
1514:     If Answer = vbOK Then
1515:         For i = 1 To $hn2_4zoneMachineSwitch
1516:             If Not $hfMTLTuneNoTestZ[i] Then
1517:
1518:                 //OPC pages are setup such that PID for MTL Tuning and PID for NRCT use the same PLC tags, so when navigating to a
1519:                 //different screen, the PLC will write its PID parameters to the screen variable's PID parameters that is navigated to
1520:                 ' $NonRecipeConstTemper_PID[i].Kp = $PID_MTLTuning[i].Kp
1521:                 ' $NonRecipeConstTemper_PID[i].Ki = $PID_MTLTuning[i].Ki
1522:                 ' $NonRecipeConstTemper_PID[i].Kd = $PID_MTLTuning[i].Kd
1523:
1524:                 $NonRecipeConstTemper_MTL[i].Theta = $MTL_TuneResults[i].Theta
1525:                 $NonRecipeConstTemper_MTL[i].Tau = $MTL_TuneResults[i].Tau
1526:                 $NonRecipeConstTemper_MTL[i].Lambda = $MTL_TuneResults[i].Lambda
1527:                 $NonRecipeConstTemper_MTL[i].Kprime = $MTL_TuneResults[i].Kprime
1528:                 $NonRecipeConstTemper_MTL[i].Km = $MTL_TuneResults[i].Km
1529:
```

```

1530:         'call global
1531:     Call SetMTLConstsDefinedFlag( i )
1532:         $SetMTLConstsDefinedFlagFunc    'marker for IWS Xref tool
1533:
1534:     '      'Shouldn't have to check if mold TC ok, cause can't do MTL tuning if mold TC is malfunctioning anyway; however,
1535:     '      ' keep here as reminder, but comment out in case need to re-apply later
1536:     '      If $sTC_OK_MoldZ[i] Then
1537:         $hnControlModeZ[i]=$MoldConstantTemper
1538:     '      Else
1539:         msg = "Cannot change to Mold Constant Temperature because the Thermocouple is not functioning properly." & vbCrLf _
1540:             & "You may want to save these to a text file for later retrieval by the administrator."
1541:         MsgBox msg, vbExclamation, "Thermocouple Error for Zone " & i & " Mold"
1542:     '      End If
1543:     Else
1544:         $hnControlModeZ[i]=$FluidConstantTemper
1545:     End If
1546: Next
1547: $Open( "Non-Recipe ConstTemper.scr" )
1548: $NavigationRadioButton = 4
1549: End If
1550: End Sub 'WriteMTLTuningResults2NRCT( )
1551: //////////////////////////////////////
1552:
1553:
1554: //////////////////////////////////////
1555: //////////////////////////////////////          CCMRec( integer )          //////////////////////////////////////
1556: //////////////////////////////////////
1557: '/REV 3.1.0/ procedure made obsolete in this revision.  Added "Control Mode Dialog.scr" which now performs these actions
1558: 'Change Control Mode for Recipe
1559: 'This script will change the control mode when on the Recipe Editor-Segments or -Constants screen.
1560: Function CCMRec( ZoneX )
1561:     $CCMRecFunc    'marker for IWS Xref tool
1562:
1563:     '/REV 3.1.0/ throw up msgbox in case this procedure is called from somewhere that I forgot about
1564:     MsgBox "Error:  Called 'CCMRec( )' global procedure which is now obsolete.  Please contact the developer.  ", _
1565:         vbOKOnly + vbExclamation, "Obsolete procedure"
1566:
1567:     ' Dim CurrentControlMode, ControlModeNext, Answer, msg, NextRecipe, NextCT, keypadString, Index
1568:     ' 'set equal to the current value before a possible change is made, in case user decided to Cancel
1569:     ' CurrentControlMode = $hnControlModeZ[ZoneX]
1570:     ' 'toggle the choices btwn 'FluidConstantTemper, FluidRecipe, MoldConstantTemper, and MoldRecipe
1571:     ' 'only PAG users have access to set value to ControlValve (=0)

```

```
1572: ' ControlModeNext = CurrentControlMode +1
1573: ' If ControlModeNext > $MoldRecipe Then ControlModeNext = $FluidConstantTemper 'reset
1574: ' 'set NextRecipe and NextCT for msgbox clarity
1575: ' If CurrentControlMode=$FluidRecipe Then NextRecipe = "MOLD " : NextCT = "MOLD "
1576: ' If CurrentControlMode=$MoldRecipe Then NextRecipe = "FLUID " : NextCT = "FLUID "
1577: '
1578: ' 'throw up message if following conditions:
1579: ' ' if REC[seg0].temper<>seg1 OR seg1<>seg2, then means they have probably setup this zone as recipe
1580: ' ' so throw up warning that they are about to change the info
1581: ' '/REV 3.1.0/ added condition that SegLast<>2
1582: ' Index = ZoneX*20-20 '=0,20,40,60 same as $SegmentPagePointer but can't use that when on Consts screen
1583: ' If (ControlModeNext = $FluidConstantTemper Or ControlModeNext = $MoldConstantTemper ) _
1584: ' And ( ( $REC[Index+0].TemperSP<>$REC[Index+1].TemperSP ) Or ( $REC[Index+1].TemperSP<>$REC[Index+2].TemperSP ) _
1585: ' Or ($hnSegLastZ[ZoneX] <> 2) ) Then
1586: '
1587: ' msg = "You have selected to change control modes from RECIPE profile to "& NextCT &"CONSTANT TEMPERATURE. " & vbCrLf _
1588: ' & "It appears that you may have already defined some segment information." & vbCrLf _
1589: ' & "Changing the control mode from recipe to constant temperature will clear out all previously entered segment information." & vbCrLf &
vbCrLf _
1590: ' & "Do you want to change this zone from RECIPE to CONSTANT TEMPERATURE?" & vbCrLf & vbCrLf _
1591: ' & "Select 'YES' if you wish to continue." & vbCrLf _
1592: ' & "Select 'NO' to skip to " & NextRecipe & "RECIPE Control." & vbCrLf _
1593: ' & "Select 'Cancel' to cancel the change of control modes."
1594: ' Answer = MsgBox( msg, vbYesNoCancel + vbQuestion, "Change to Constant Temperature Control for Zone "& ZoneX &"?" )
1595: '
1596: ' If Answer = vbYes Then
1597: ' keypadString = "hnSPTemperConstZ[" & ZoneX & "]"
1598: ' $Keypad( keypadString, "Keypad", 0, "Zone " & ZoneX & " Const Temper SP [degF]" )
1599: ' $hnControlModeZ[ZoneX] = ControlModeNext
1600: ' 'initiate sequence to clear out REC Time, Temper, R?, RR info for the zone, call global
1601: ' Call RecipeCMCT( ZoneX ) : $RecipeCMCTFunc 'marker for IWS Xref
1602: ' End If
1603: ' If Answer = vbNo Then
1604: ' If CurrentControlMode=$FluidRecipe Then
1605: ' 'skip mold const temper and set equal to mold recipe
1606: ' $hnControlModeZ[ZoneX]=$MoldRecipe
1607: ' Else
1608: ' 'skip fluid const temper and set equal to fluid recipe
1609: ' $hnControlModeZ[ZoneX]=$FluidRecipe
1610: ' End If
1611: ' 'hide Load/Start Recipe button because must save recipe again before starting new part
1612: ' $RecipeIsSaved = 0
```

```

1613: ' End If
1614: '     If Answer = vbCancel Then
1615: '         'set back to what it was before
1616: '             $hnControlModeZ[ZoneX] = CurrentControlMode
1617: '     End If
1618: ' 
1619: ' Else
1620: '     $hnControlModeZ[ZoneX] = ControlModeNext
1621: '     'hide Load/Start Recipe button because must save recipe again before starting new part
1622: '     $RecipeIsSaved = 0
1623: ' End If
1624: End Function      'CCMRec( ZoneX )
1625: ////////////////////////////////////////
1626: 
1627: 
1628: ////////////////////////////////////////
1629: //////////////////////////////////////////          CCMNRCT( integer )           \////////////////////////////////////////
1630: //////////////////////////////////////////////////
1631: 'Change Control Mode Non Recipe Constant Temperature
1632: ' (CCMRec( ZoneX ) contains the routine to change control modes for Recipe configuration)
1633: '//Called when the Control Mode field for any zone is change on "Non-Recipe ConstTemper.scr"
1634: Function CCMNRCT( ZoneX )
1635:     $CCMNRCTFunc    'marker for IWS Xref tool
1636:     Dim MsgBoxReturn, ControlMode, ControlModeNext, msg, CurrentControlMode
1637:     CurrentControlMode = $hnControlModeZ[ZoneX]
1638:     If CurrentControlMode = $ControlValve Then
1639:         ControlMode = "Manual"
1640:         ControlModeNext = "Fluid/PID"
1641:     Else If CurrentControlMode = $FluidConstantTemper Then
1642:         ControlMode = "Fluid/PID"
1643:         ControlModeNext = "Mold/MTL"
1644:     Else If CurrentControlMode = $MoldConstantTemper Then
1645:         ControlMode = "Mold/MTL"
1646:         ControlModeNext = "Fluid/PID"
1647:     Else
1648:         ControlMode = "???"        'could get this condition if navigate from recipe to NRCT screen
1649:         ControlModeNext = "Fluid/PID"
1650:     End If
1651: End If
1652: End If
1653: 
1654: If ControlMode <> "???" Then
```

```
1655:     If ControlMode <> "Manual" Then
1656:         msg = "Do you really want to change control modes from "& ControlMode & " to " & ControlModeNext & " on Zone " & ZoneX & "?"
1657:         MsgBoxReturn = MsgBox( msg, vbQuestion + vbYesNo, "Change Control Modes Zone "& ZoneX & " ....?" )
1658:     End If
1659:     If MsgBoxReturn = vbYes Or ControlMode = "Manual" Then
1660:         'toggle the constant temperature control choices if TCs are working
1661:         If CurrentControlMode = $FluidConstantTemper Then
1662:             If $sTC_OK_MoldZ[ZoneX] Then
1663:                 'change modes
1664:                 $hnControlModeZ[ZoneX] = $MoldConstantTemper
1665:                 'see if all MTL consts are defined for the zone
1666:                 Call SetMTLConstsDefinedFlag(ZoneX)
1667:                 $SetMTLConstsDefinedFlagFunc 'marker for IWS Xref tool
1668:             Else
1669:                 msg = "Cannot change to Mold Constant Temperature because the Thermocouple is not functioning properly."
1670:                 MsgBox msg, vbExclamation, "Thermocouple Error for Zone " & ZoneX & " Mold"
1671:             End If
1672:         End If
1673:         If CurrentControlMode <> $FluidConstantTemper Then
1674:             If $sTC_OK_FluidZ[ZoneX] Then
1675:                 $hnControlModeZ[ZoneX] = $FluidConstantTemper
1676:             Else
1677:                 msg = "Cannot change to Fluid Constant Temperature because the Thermocouple is not functioning properly."
1678:                 MsgBox msg, vbExclamation, "Thermocouple Error for Zone " & ZoneX & " Fluid"
1679:             End If
1680:         End If
1681:     End If
1682:
1683: Else 'ctl mode = ????' 'could get this condition if navigate from recipe to NRCT screen
1684:     If $sTC_OK_FluidZ[ZoneX] Then
1685:         If $hnControlModeZ[ZoneX] = $FluidRecipe Then
1686:             '/REV 3.1.0/ need to perform "bumpless" cause PLC doesn't account for changing ctl modes when recipe was previously run.
1687:             ' what would happen would be that the last saved TemperConst is saved with the recipe and downloaded to the PLC; even if a zone was
recipe as
1688:             ' opposed to CT, when abort recipe and change to NRCT, the PLC would pick the last saved TemperConst as the SP, when it should
1689:             ' be bumpless, so use the last SP that the recipe was using
1690:             $hnSPTemperConstZ[ZoneX] = $rSP_PID_Z[ZoneX]
1691:             $hnControlModeZ[ZoneX] = $FluidConstantTemper
1692:         Else 'when changing from MoldRecipe to FluidConstantTemper, the PLC will perform bumpless
1693:             $hnControlModeZ[ZoneX] = $FluidConstantTemper
1694:         End If
1695:     Else
```

```

1696:         msg = "Cannot change to Fluid Constant Temperature because the Thermocouple is not functioning properly."
1697:         MsgBox msg, vbExclamation, "Thermocouple Error for Zone " & ZoneX & " Fluid"
1698:     End If
1699:
1700: End If
1701: End Function 'Function CCMNRCT( ZoneX )
1702: //////////////////////////////////////
1703:
1704:
1705: //////////////////////////////////////
1706: ////////////////////////////////////// CCModes2NRCT( ) //////////////////////////////////////
1707: //////////////////////////////////////
1708: '/REV 3.1.0/ newly added
1709: 'change control mode based on what is currently running on PLC, if recipe, then change to corresponding CT
1710: ' example: if running FluidRecipe, then change to FluidConstTemper
1711: '//Called from "Constant Temperature" button on "NAVIGATION.scr"
1712: Sub CCModes2NRCT( )
1713:     $CCModes2NRCTSub 'marker for IWS Xref
1714:     Dim i, StartTime, StartMsecValue
1715:
1716:     'open the TempClear screen to turn off the current screen's OPC connection,
1717:     ' so that the screen I really want to open has a clear OPC connection when it is opened.
1718:     $Open( "TempClear.scr" ) : $Close( "TempClear.scr" )
1719:
1720:     'enable OPC comm for screen variables
1721:     $OPCCommNonRecConstTemper = 0
1722:
1723:     StartMsecValue = $PLCRollingMsecCounter
1724:     StartTime = $GetTickCount() 'IWS internal clock in milliseconds
1725:     Do 'wait for XX milliseconds or until OPC sees a change in its counter
1726:         'if see a change in counter, then know that OPC is updating; otherwise timer will time out, signifying a lack of comm
1727:     Loop Until ( $Abs( $GetTickCount() - StartTime ) >= 10000 ) Or ( $PLCRollingMsecCounter <> StartMsecValue )
1728:     'could use $DoLoopTimer here to see how long it takes, but since this procedure is called to $RunGlobalProcedureOnServer, you won't
1729:     ' see the $DoLoopTimer tag update dynamically anyway
1730:
1731:     For i=1 To $hn2_4zoneMachineSwitch
1732:         If $hnControlModeZ[i] >= $MoldConstantTemper Then
1733:             $hnControlModeZ[i] = $MoldConstantTemper
1734:             $hnSPTemperConstZ[i] = $rSP_MTL_Z[i]
1735:         Else
1736:             $hnControlModeZ[i] = $FluidConstantTemper
1737:             $hnSPTemperConstZ[i] = $rSP_PID_Z[i]

```

```

1738:      End If
1739:      Next
1740:
1741:      $Open( "Non-Recipe ConstTemper.scr" )
1742:
1743: End Sub 'CCModes2NRCT( )
1744: //////////////////////////////////////
1745:
1746:
1747: //////////////////////////////////////
1748: //////////////////////////////////////      RecipeCMCT( integer )      //////////////////////////////////////
1749: //////////////////////////////////////
1750: //CMCT = Control Mode Constant Temperature
1751: //Code to handle setting the zone to be a constant temperature controlled zone (rather than recipe segment control)
1752: //Instead of creating separate code for a const temper SP, I set the recipe such that the Seg0,1,2 TemperSPs=ConstTemperSP
1753: // and then set Seg0,1 TimeSP=0seconds, and Seg2 TimeSP=1second
1754: //this will make the SP state states and other state sets still work with the Prht/RFCs/Run state sets in the PLC
1755: //Called from "OK" button on "Control Mode Dialog.scr" if a ConstTemper mode was chosen
1756: //Called when SetPoint field for ConstTemper Control has successfully been completed
1757: // on "Recipe Editor-Segments.scr" or "Recipe Editor-Constants.scr" screens
1758: '/REV 3.1.0/ moved from Confirm script to here because makes more sense to be a function
1759: Function RecipeCMCT( ZoneX )
1760:     $RecipeCMCTFunc    'marker for IWS Xref
1761:     Dim i, ZoneIndexStart
1762:     'ZoneIndexStart is same as $SegmentPagePointer: 0=z1,20=z2,40=z3,60=z4
1763:     ZoneIndexStart = ZoneX * 20 - 20      'ZoneX=1,2,3,4
1764:
1765:     //clear out all segments in the zone
1766:     For i = ZoneIndexStart To (ZoneIndexStart + 19)
1767:         $REC[i].DwellTimeSP = 0
1768:         $REC[i].TemperSP = $REC[i].TemperSP->Min
1769:         $REC[i].RampYN = 0
1770:         $REC[i].RampRate = 0
1771:         $REC[i].Comment = " "
1772:     Next
1773:     //set TemperSP for segments 0 and 1
1774:     $REC[ ZoneIndexStart + 0 ].TemperSP = $hnSPTemperConstZ[ ZoneX ]
1775:     $REC[ ZoneIndexStart + 1 ].TemperSP = $hnSPTemperConstZ[ ZoneX ]
1776:     //set segment 2 Time and Temper SP
1777:     $REC[ ZoneIndexStart + 2 ].DwellTimeSP = 1
1778:     $REC[ ZoneIndexStart + 2 ].TemperSP = $hnSPTemperConstZ[ ZoneX ]
1779:     //deselect Cycle Abort Temp SP and Prht Mold Temper Limits

```



```
1780:     $hfSPTemperCycleAbortZ[ ZoneX ] = 0
1781:     'global CheckRecipe() sets hnSPTemperCycleAbortZx, but keep don't reset, in case user changes their mind
1782:     $hfMoldTemperLimitsZ[ ZoneX ] = 0
1783:     'keep Upper and Lower Limit Values, don't reset
1784:
1785:     '/REV 3.1.0/ set last segment
1786:     Call FindSegLastZx( )      :    $FindSegLastZxFunc
1787:
1788:     'hide Load/Start Recipe button because must save recipe again before starting new part
1789:     $RecipeIsSaved = 0
1790:
1791: End Function 'Function RecipeCMCT( ZoneX )
1792: //////////////////////////////////////////////////
1793:
1794:
1795: //////////////////////////////////////////////////
1796: ////////////////////////////////////////////////// SetLambda( integer ) //////////////////////////////////////////////////
1797: //////////////////////////////////////////////////
1798: 'routine to check the value of the input Closed Loop Time Constant - Lambda
1799: 'Lambda on all zones should be equal to the longest mold time constant, Tau, as a minimum,
1800: ' per DF12 booklet explanation and BDX-613-1980.  Lambda >= TauMax
1801: 'Explanation from DF12:  <my notes>
1802: 'The closed loop time constant <lambda> on all zones should be changed, as a minimum, to the longest
1803: 'mold time constant <tau>.  If DF13 <tuning routine> were used and the mold time constants were
1804: '245 and 625 seconds for zones 1 and 2 'of a two zone mold, the closed loop time constant should be set
1805: 'to a minimum of 625 seconds for both zones.  Do not alter the mold, time constant <tau>, dead time <theta>,
1806: 'or steady state gains <Km>.  The only variables the user should alter are the closed loop gain <Kprime>
1807: 'and time constant <lambda>.
1808: 'Increasing the closed loop time constant or decreasing the entered gain have the effect of making the
1809: 'closed loop response more overdamped.  <meaning slower process response to a SP change>
1810: //Called when the keypad entry was successful for a Tau or Lambda input for any zone on "Recipe Editor-Constants.scr"
1811: Function SetLambda( ZoneY )
1812:     $SetLambdaFunc 'marker for IWS Xref tool
1813:
1814:     Dim ArrMaxTau(4), i, MaxTau, msg, answer
1815:     Erase ArrMaxTau
1816:
1817:     'find max defined Tau
1818:     For i=1 To $hn2_4zoneMachineSwitch
1819:         If $hnControlModeZ[i] >= $MoldConstantTemper Then
1820:             ArrMaxTau(i) = $REC_MTL[i].Tau
1821:         Else
```

```

1822:     ArrMaxTau(i) = 0
1823:     End If
1824: Next
1825: 'find the max value of the array
1826: MaxTau = $Max( 1, ArrMaxTau(1), ArrMaxTau(2), ArrMaxTau(3), ArrMaxTau(4) )
1827: 'BUG IN MAX FUNCTION. If the first numValue happens to be 0, the return is 0, and it doesn't find the max;
1828: ' so place an arbitrary numValue of 1 second in the first value; Tau should always be >1 so this should be ok.
1829: If $REC_MTL[ZoneY].Lambda < MaxTau Then
1830:     msg = "The Closed Loop Time Constant, Lambda, on all zones should be changed, as a minimum, to the longest " _
1831:         & "Mold Time Constant, Tau. " & vbCrLf _
1832:         & "The longest Mold Time Constant, Tau, is " & MaxTau & "." & vbCrLf & vbCrLf _
1833:         & "Select 'YES' if you wish to change Lambda for ALL zones to this value." & vbCrLf _
1834:         & "Select 'NO' if you wish to change Lambda for THIS zone (Zone " & ZoneY & ") to this value." & vbCrLf _
1835:         & "Selecting 'Cancel' will set Lambda>=Tau for this zone."
1836:
1837:     answer = MsgBox( msg, vbYesNoCancel + vbQuestion, "Value of the Closed Loop Time Constant, Lambda, for Zones" )
1838:
1839:     If answer = vbYes Then
1840:         For i=1 To $hn2_4zoneMachineSwitch
1841:             $REC_MTL[i].Lambda = MaxTau
1842:         Next
1843:     End If
1844:     If answer = vbNo Then
1845:         $REC_MTL[ZoneY].Lambda = MaxTau
1846:     End If
1847:     If answer = vbCancel And $REC_MTL[ZoneY].Lambda < $REC_MTL[ZoneY].Tau Then
1848:         $REC_MTL[ZoneY].Lambda = $REC_MTL[ZoneY].Tau
1849:         ' MsgBox "Unexpected results may occur!", vbExclamation, "I hope you'll reconsider."
1850:     End If
1851: End If
1852:
1853: End Function 'SetLambda( ZoneY )
1854: //////////////////////////////////////
1855:
1856:
1857: //////////////////////////////////////
1858: ////////////////////////////////////// SetLambdaCT( integer ) //////////////////////////////////////
1859: //////////////////////////////////////
1860: 'routine to check the value of the input Closed Loop Time Constant - Lambda, for Non-Recipe Const Temper control
1861: //Called when the keypad entry was successful for a Tau or Lambda input for any zone on "Non-Recipe ConstTemper.scr"
1862: Function SetLambdaCT( ZoneY )
1863:     $SetLambdaCTFunc 'marker for IWS Xref tool

```

```
1864:
1865: Dim ArrMaxTau(4), i, MaxTau, msg, answer
1866: Erase ArrMaxTau
1867:
1868: 'find max defined Tau
1869: For i=1 To $hn2_4zoneMachineSwitch
1870:     If $hnControlModeZ[i] >= $MoldConstantTemper Then
1871:         ArrMaxTau(i) = $NonRecipeConstTemper_MTL[i].Tau
1872:     Else
1873:         ArrMaxTau(i) = 0
1874:     End If
1875: Next
1876: 'find the max value of the array
1877: MaxTau = $Max( 1, ArrMaxTau(1), ArrMaxTau(2), ArrMaxTau(3), ArrMaxTau(4) )
1878: If $NonRecipeConstTemper_MTL[ZoneY].Lambda < MaxTau Then
1879:     msg = "The Closed Loop Time Constant, Lambda, on all zones should be changed, as a minimum, to the longest " _
1880:         & "Mold Time Constant, Tau. " & vbCrLf _
1881:         & "The longest Mold Time Constant, Tau, is " & MaxTau & "." & vbCrLf & vbCrLf _
1882:         & "Select 'YES' if you wish to change Lambda for ALL zones to this value." & vbCrLf _
1883:         & "Select 'NO' if you wish to change Lambda for THIS zone (Zone " & ZoneY & ") to this value." & vbCrLf _
1884:         & "Selecting 'Cancel' will set Lambda>=Tau for this zone."
1885:
1886:     answer = MsgBox( msg, vbYesNoCancel + vbQuestion, "Value of the Closed Loop Time Constant, Lambda, for Zones" )
1887:
1888:     If answer = vbYes Then
1889:         For i=1 To $hn2_4zoneMachineSwitch
1890:             $NonRecipeConstTemper_MTL[i].Lambda = MaxTau
1891:         Next
1892:     End If
1893:     If answer = vbNo Then
1894:         $NonRecipeConstTemper_MTL[ZoneY].Lambda = MaxTau
1895:     End If
1896:     If answer = vbCancel And $NonRecipeConstTemper_MTL[ZoneY].Lambda < $NonRecipeConstTemper_MTL[ZoneY].Tau Then
1897:         $NonRecipeConstTemper_MTL[ZoneY].Lambda = $NonRecipeConstTemper_MTL[ZoneY].Tau
1898:         ' MsgBox "Unexpected results may occur!", vbExclamation, "I hope you'll reconsider."
1899:     End If
1900: End If
1901:
1902: End Function 'SetLambdaCT( ZoneY )
1903: //////////////////////////////////////
1904:
1905:
```

```

1906: //////////////////////////////////////
1907: ////////////////////////////////////// SegGraphFileCreateZone( ) //////////////////////////////////////
1908: //////////////////////////////////////
1909: 'routine to setup data file for graphing the recipe's defined set points for graphical representation for the user
1910: 'this will create a string of data points, strWriteText
1911: 'one line graph section (each recipe's segment) accounts for 2 lines of the txt file
1912: 'for a 3 segment recipe there will be 6 data points; the endpoint of one segment's time must overlap the time of the next segment
1913: 'the XTime is a running calculated elapsed time for that particular zone
1914: '//Called from "View Segment Grapher" button on "Recipe Editor-Constants.scr" or "Recipe Editor-Segments.scr" or "Recipe Viewer.scr"
1915: '/REV 3.1.0/ changed file writing logic based on whether or not zone is Recipe or ConstTemper
1916: '/REV 3.1.0/ moved open screen into the sub instead of on the screen button
1917: '// !!!!!!!!!!! NEED TO FIX: GRAPHER CAN ONLY CONNECT DOTS ON SEGMENTS WITH TIME<=1.8 HOURS !!!!!!!!!!!!!!!
1918: '// !!! This is a bug that InduSoft is aware of, I will wait for their fix. !!!
1919: Sub SegGraphFileCreateZone( )
1920:     $SegGraphFileCreateZoneSub 'marker for IWS xref
1921:
1922:     Dim strFileName, strWriteText, XTime, i, j, k
1923:
1924:     For i = 1 To $hn2_4zoneMachineSwitch
1925:
1926:         XTime = 0
1927:         strFileName = $GetAppPath() & "SegmentGrapherZ" & i & ".txt"
1928:
1929:         If $hnControlModeZ[ i ] = $FluidRecipe Or $hnControlModeZ[ i ] = $MoldRecipe Then
1930:             'start setting up string tag, starting at Seg0: Time=0 and Seg0|20|40|60 Temper
1931:             strWriteText = 0 & " " & $REC[i*20-20].TemperSP
1932:             XTime = XTime + $REC[i*20-20].DwellTimeSP
1933:             strWriteText = strWriteText & vbCrLf & XTime & " " & $REC[i*20-20].TemperSP
1934:             For k = (i*20-20+1) To (i*20-20+$hnSegLastZ[ i ]) 'Seg1 thru SegLast
1935:                 If $REC[k].RampYN Then
1936:                     strWriteText = strWriteText & vbCrLf & XTime & " " & $REC[k-1].TemperSP
1937:                     XTime = XTime + $REC[k].DwellTimeSP
1938:                     strWriteText = strWriteText & vbCrLf & XTime & " " & $REC[k].TemperSP
1939:                 Else
1940:                     strWriteText = strWriteText & vbCrLf & XTime & " " & $REC[k].TemperSP
1941:                     XTime = XTime + $REC[k].DwellTimeSP
1942:                     strWriteText = strWriteText & vbCrLf & XTime & " " & $REC[k].TemperSP
1943:                 End If
1944:             Next
1945:         End If
1946:
1947:         If $hnControlModeZ[ i ] = $FluidConstantTemper Or $hnControlModeZ[ i ] = $MoldConstantTemper Then

```

```
1948:         'for graphing only setup segment 0 to "show" that it is a constant temper zone by putting
1949:         ' a duration that is as long as $CycleElapsedTimeEst
1950:         strWriteText = 0 & " " & $hnSPTemperConstZ[ i ]
1951:         strWriteText = strWriteText & vbCrLf & $CycleElapsedTimeEst & " " & $hnSPTemperConstZ[ i ]
1952:     End If
1953:
1954:     'write seg data to a file for each zone; option zero replaces the contents of the exisiting file
1955:     $FileWrite( strFileName, strWriteText, 0 )
1956: Next
1957:
1958: $Open( "Segment Grapher.scr" )
1959:
1960: End Sub 'SegGraphFileCreateZone( )
1961: //////////////////////////////////////
1962:
1963:
1964: //////////////////////////////////////
1965: ////////////////////////////////////// procedures for Input Dwell Dialog screen //////////////////////////////////////
1966: //////////////////////////////////////
1967: 'converts the user input to a dwell time for the recipe to seconds
1968: //Called when a successful keypad entry is made in any of the time input fields
1969: Sub dwellCalc( )
1970:     $dwellCalcSub 'marker for IWS Xref tool
1971:     If $inputHours>0 Or $inputMinutes>0 Or $inputSeconds>0 Then
1972:         'this is the clock section of the input "hh:mm:ss"
1973:         'inputDwell should be set as INTEGER in tag database to match REC[i].DWELL as INTEGER in seconds
1974:         $inputDwellTime = ( $inputHours*3600 + $inputMinutes*60 + $inputSeconds )
1975:
1976:     Else If $inputTotalHours>0 Then
1977:         'this is the input total hours, or total minutes, or total seconds section
1978:         'this input can be a decimal, then round to nearest integer to convert value to seconds
1979:         $inputDwellTime = $Round( $inputTotalHours*3600 )
1980:
1981:     Else If $inputTotalMinutes>0 Then
1982:         $inputDwellTime = $Round( $inputTotalMinutes*60 )
1983:
1984:     Else
1985:         $inputDwellTime = $Round( $inputTotalSeconds )
1986:
1987:     End If //else
1988: End If //else
1989: End If //main
```

```
1990: End Sub
1991: //////////////////////////////////
1992: //Called from "Reset ALL fields to ZERO" button, and any of the Total time input fields, and the ScreenScript
1993: Sub resetALLFields2Zero( )
1994:     $resetALLFields2ZeroSub 'marker for IWS Xref tool
1995:     $inputDwellTime = 0
1996:     $inputHours = 0 : $inputMinutes = 0 : $inputSeconds = 0
1997:     $inputTotalHours = 0 : $inputTotalMinutes = 0 : $inputTotalSeconds = 0
1998: End Sub
1999: //////////////////////////////////
2000: //Called when any of the HH:MM:SS fields are pressed
2001: Sub resetTotalFields2Zero( )
2002:     $resetTotalFields2ZeroSub 'marker for IWS Xref tool
2003:     $inputTotalHours = 0 : $inputTotalMinutes = 0 : $inputTotalSeconds = 0
2004: End Sub
2005: ////////////////////////////////// END procedures for Input Dwell Dialog screen //////////////////////////////////
2006:
2007:
2008: //////////////////////////////////
2009: ////////////////////////////////// PopulateOpenRecipeGridControl( ) //////////////////////////////////
2010: //////////////////////////////////
2011: 'this routine populates the recipe list for the user to choose a recipe name when the "Open" recipe button is pressed
2012: 'the user is limited to 90 recipes
2013: //Called from ScreenScript on "Open Recipe Dialog.scr"
2014: Sub PopulateOpenRecipeGridControl( )
2015:     $PopulateOpenRecipeGridControlSub 'marker for IWS Xref tool
2016:
2017:     Dim RecipeNameTemp, RecipePartNumberTemp, i, j, k, SearchCriteria, myMsg, RPAFTemp
2018:
2019:     //set existing name to temporary variable
2020:     RecipeNameTemp = $RecipeName
2021:     RecipePartNumberTemp = $RecipePartNumber
2022:     //auto highlight the first row, so know where "cursor" is
2023:     $SelectedRow = 1
2024:
2025:     ////////// Read defined recipe files to populate grid control section //////////
2026:     //clear out class arrays
2027:     For j = 0 To $RecipeList->size //90 elements
2028:         $RecipeFiles[i] = ""
2029:         $RecipeList[j].FileName = ""
2030:         $RecipeList[j].PartNumber = ""
2031:     Next
```

```
2032:
2033: //get recipe files with proper extension and place list in Array "RecipeFiles"
2034: SearchCriteria = $RecipesFolderPath & "*" & $RecipeFileExtension
2035: 'MsgBox "SearchCriteria = " & SearchCriteria
2036: $RecipeNumFiles = $FindFile( SearchCriteria, "RecipeFiles" )
2037:
2038: //error checking/messaging on 90 index limit to the the class array
2039: If $RecipeNumFiles > $RecipeList->size Then
2040:     myMsg = "oops, found " & $RecipeNumFiles & " Recipe files."
2041:     myMsg = myMsg & vbCrLf & "The array size for Class Tag $RecipeList is currently limited to " & $RecipeList->size & " elements."
2042:     myMsg = myMsg & vbCrLf & "The first " & ( $RecipeList->size - 1 ) & " files found, plus the last file found, will be shown."
2043:     'MsgBox myMsg, vbExclamation, "Too many recipes...."
2044:     $TooManyRecipesErrorMessage = myMsg
2045:     $AlarmWarningMessage = 7 'set to display $TooManyRecipesErrorMessage
2046:     $Open( "AlarmWarning Popup.scr" )
2047: End If
2048:
2049: //set existing to temp to restore later
2050: RPAFTemp = $RecipePathAndFileName
2051:
2052: //populate class array for grid control, start at index 1 because grid control row numbers start at 1 not 0
2053: For k = 1 To $RecipeNumFiles
2054:     //extract Name and PartNumber info from file to use in grid control by triggering Script Recipe2
2055:     $RecipePathAndFileName = $RecipesFolderPath & "\" & $RecipeFiles[k]
2056:     $Recipe( "Load:Recipe2" )
2057:     $RecipeList[k].FileName = $RecipeName
2058:     $RecipeList[k].PartNumber = $RecipePartNumber
2059:     ' MsgBox "Pause"
2060: Next
2061: /////////////// END Read defined recipe files to populate grid control section ///////////////
2062:
2063: //restore
2064: $RecipePathAndFileName = RPAFTemp
2065:
2066: //reset names back to what it was for display, because they just got overwritten by code to populate grid control
2067: $RecipeName = RecipeNameTemp
2068: $RecipePartNumber = RecipePartNumberTemp
2069:
2070: End Sub 'PopulateOpenRecipeGridControl( )
2071: //////////////////////////////////////
2072:
2073:
```

```
2074: //////////////////////////////////////
2075: //////////////////////////////////////      RecipeFileHeaderConstants( )      //////////////////////////////////////
2076: //////////////////////////////////////
2077: //Set constants used for Task Recipe1
2078: //These are used to set separations in the recipe data file, and used for notes in the recipe data file
2079: //Called from Global RecipeSave( )
2080: Sub RecipeFileHeaderConstants( )
2081:     $RecipeFileHeaderConstantsSub 'marker for IWS Xref tool
2082:
2083:     'call sub on Global Procedures because loading a recipe may have changed the value of the tag
2084:     MHCVerRev( )      :      $MHCVerRevSub 'marker for IWS Xref tool
2085:
2086:     //data structure of Recipe Text File to make the text file more readable
2087:     $TextFileInfo[0] = ""
2088:     $TextFileInfo[1] = "!!! DO NOT EDIT THIS FILE !!!  EDITING MUST BE DONE VIA THE HMI.  EDITING THIS FILE CAN CAUSE LOSS OF DATA !!!"
2089:     $TextFileInfo[2] = $wMHC_PLC_VersionRevision
2090:     $TextFileInfo[3] = $MHCVersionRevision
2091:     'must assign index's 2&3 because if read in a text file with older version of MHC the tag will be overwritten
2092:     'but since I run this routine right before the recipe is saved again, the new version info will be updated
2093:     $TextFileInfo[4] = $hn2_4zoneMachineSwitch->Description
2094:     $TextFileInfo[5] = $hfManualPreheat->Description
2095:     $TextFileInfo[6] = "ControlMode for each zone, listed in order Z1 Z2 Z3 Z4; {Value: 0=ControlValve, 1=FluidConstantTemper, 2=FluidRecipe,
3=MoldConstantTemper, 4=MoldRecipe}: "
2096:     $TextFileInfo[7] = "Mold Control Parameters, listed in order Z1 Z2 Z3 Z4; Theta, Dead Time Constant [seconds]: "
2097:     $TextFileInfo[8] = "Mold Control Parameters, listed in order Z1 Z2 Z3 Z4; Tau, Mold Time Constant [seconds]: "
2098:     $TextFileInfo[9] = "Mold Control Parameters, listed in order Z1 Z2 Z3 Z4; Lambda, Closed Loop Time Constant [seconds]: "
2099:     $TextFileInfo[10] = "Mold Control Parameters, listed in order Z1 Z2 Z3 Z4; K', Entered Gain: "
2100:     $TextFileInfo[11] = "Mold Control Parameters, listed in order Z1 Z2 Z3 Z4; Km, Steady State Gain: "
2101:     $TextFileInfo[12] = "Fluid Control Parameters, listed in order Z1 Z2 Z3 Z4; Kp, Proportional Gain Constant, default is 1.5: "
2102:     $TextFileInfo[13] = "Fluid Control Parameters, listed in order Z1 Z2 Z3 Z4; Ki, Integral Gain Constant, default is 4.5: "
2103:     $TextFileInfo[14] = "Fluid Control Parameters, listed in order Z1 Z2 Z3 Z4; Kd, Derivative Gain Constant, default is 0.0: "
2104:     $TextFileInfo[15] = "If using ConstTemper control (Mode 1 or 3), this is the last saved SP, listed in order Z1 Z2 Z3 Z4: "
2105:     $TextFileInfo[16] = $hfSPTemperCycleAbortZ->Description
2106:     $TextFileInfo[17] = $hnSPTemperCycleAbortZ->Description
2107:     $TextFileInfo[18] = $hfMoldTemperLimitsZ->Description
2108:     $TextFileInfo[19] = $hnMoldTemperLimitUpperZ->Description
2109:     $TextFileInfo[20] = $hnMoldTemperLimitLowerZ->Description
2110:     $TextFileInfo[21] = "The last defined segment for each zone (19 is max), listed in order Z1 Z2 Z3 Z4: "
2111:     $TextFileInfo[22] = "" 'not used
2112:     $TextFileInfo[23] = "" 'not used
2113:     $TextFileInfo[24] = "" 'not used
2114:     $TextFileInfo[25] = "" 'not used
```



```
2115: $TextFileInfo[26] = "" 'not used
2116: $TextFileInfo[27] = "--- Segment Time Set Points [seconds] ---"
2117: $TextFileInfo[28] = "--- Segment Temperature Set Points [deg F] ---"
2118: $TextFileInfo[29] = "--- Was Ramp selected for the segment? (1=Y, 0=N) ---"
2119: $TextFileInfo[30] = "--- The calculated Ramp Rate for the segment ---"
2120: $TextFileInfo[31] = "--- Comments for each segment ---"
2121: $TextFileInfo[32] = "      1111 Zone1 information 1111      (Seg0 Seg1 Seg2 ...)"
2122: $TextFileInfo[33] = "      2222 Zone2 information 2222      (Seg0 Seg1 Seg2 ...)"
2123: $TextFileInfo[34] = "      3333 Zone3 information 3333      (Seg0 Seg1 Seg2 ...)"
2124: $TextFileInfo[35] = "      4444 Zone4 information 4444      (Seg0 Seg1 Seg2 ...)"
2125: $TextFileInfo[36] = "Estimated Cycle Time [seconds]"
2126:
2127: $TextFileInfo[40] = "END OF FILE"
2128: End Sub 'RecipeFileHeaderConstants( )
2129: //////////////////////////////////////
2130:
2131:
2132: //////////////////////////////////////
2133: //////////////////////////////////////      DailyRun( )      //////////////////////////////////////
2134: //////////////////////////////////////
2135: '/REV 3.1.0/ newly created; checks PLC battery and syncs the PLC clock to HMI clock
2136: //Called by Task Script 90 - Daily Run Script
2137: Sub DailyRun( )
2138:     $DailyRunSub      'marker for IWS Xref
2139:     $myText = "DailyRunSub started"
2140:
2141:     Dim StartTime, StartMsecValue
2142:
2143:     'enable opc sheet
2144:     $OPCCommDailyRun = 0
2145:
2146:     'set starting value of PLC counter after OPC sheet enabled
2147:     StartMsecValue = $PLCRollingMsecCounter
2148:     StartTime = $GetTickCount() 'IWS internal clock in milliseconds
2149:     Do 'wait for XX milliseconds or until OPC sees a change in its counter
2150:         'if see a change in counter, then know that OPC is updating; otherwise timer will time out, signifying a lack of comm
2151:         '$myText = "wait "& $Abs( $GetTickCount() - StartTime )
2152:     Loop Until ( $Abs( $GetTickCount() - StartTime ) >= 10000 ) Or ( $PLCRollingMsecCounter <> StartMsecValue )
2153:     $myText = "opc good for DailyRunSub"
2154:
2155:     'check battery status from $PLCReserved10 as long as good quality on OPC
2156:     'info from Phoenix-Contact/Entivity says that b3=1 means the battery is low
```

```
2157: 'I checked this by removing the battery, b3 went hi, so could also mean dead or missing
2158: 'notify operator to replace battery, type CR2354 3V
2159: If $PLCReserved10->B3 And $PLCReserved10->Quality=192 Then
2160:     $AlarmWarningMessage = 4
2161:     $Open( "AlarmWarning Popup.scr" )
2162: End If
2163:
2164: 'if clock changing PLC tag is available on OPC, then run sync routine
2165: If $PLCSpare3->Quality=192 Then
2166:     'sync PLC date/time clock with HMI date/time clock
2167:     Call UpdatePLCClock()
2168:     $UpdatePLCClockSub      'marker for IWS Xref
2169: End If
2170:
2171: 'disable OPC comm
2172: $OPCCommDailyRun = 1
2173:
2174: 'reset execution tag for the Task Script
2175: $DailyRunScript = 2
2176: 'setup retentive date tag to compare in Startup Script, add a preceding zeroes to the month and day if <10
2177: If $Month < 10 Then
2178:     $DailyRunScriptLastRunDate = "0" & $Month
2179: Else
2180:     $DailyRunScriptLastRunDate = $Month
2181: End If
2182: If $Day < 10 Then
2183:     $DailyRunScriptLastRunDate = $DailyRunScriptLastRunDate & "/" & $Day
2184: Else
2185:     $DailyRunScriptLastRunDate = $DailyRunScriptLastRunDate & "/" & $Day
2186: End If
2187: $DailyRunScriptLastRunDate = $DailyRunScriptLastRunDate & "/" & $Year
2188:
2189: End Sub      'DailyRun( )
2190: ///////////////////////////////////////////////////
2191:
2192:
2193: ///////////////////////////////////////////////////
2194: ///////////////////////////////////////////////////      UpdatePLCClock( )      ///////////////////////////////////////////////////
2195: ///////////////////////////////////////////////////
2196: '/REV 3.1.0/ newly created
2197: // information from Entity on WinPLC system tag Spare3
2198: // To set the time ---> my code to do this is $hnUpdateClock = 1
```

```
2199:  'bits 31-24 = 1
2200:  'bits 23-16 = second (0-59)
2201:  'bits 15-8 = minute (0-59)
2202:  'bits 7-0 = hour (0-23)
2203:  // To set the day of the month ---> my code to do this is $hnUpdateClock = 4
2204:  'bits 31-24 = 2
2205:  'bits 7-0 = day (1-31)
2206:  // To set the month ---> my code to do this is $hnUpdateClock = 3
2207:  'bits 31-24 = 3
2208:  'bits 7-0 = month (1-12)
2209:  // To set the year ---> my code to do this is $hnUpdateClock = 2
2210:  'bits 31-24 = 4
2211:  'bits 15-0 = year
2212:  //Called by Global DailyRun( )
2213:  Sub UpdatePLCClock( )
2214:      $UpdatePLCClockSub      'marker for IWS Xref
2215:
2216:  Dim StartTime
2217:
2218:  'set the time of day
2219:  $hnUpdateClock = 1
2220:  If $hnUpdateClock = 1 Then
2221:      'reset possible HMI previous value of tag; PLC resets to 0 when sync is complete, but OPC doesn't always catch it
2222:      $PLCSpare3 = 0
2223:      'time
2224:      $PLCSpare3->B24 = 1      'bits 31-24 = 1
2225:      'hour bits 0-7
2226:      $PLCSpare3->B0 = $Hour->B0
2227:      $PLCSpare3->B1 = $Hour->B1
2228:      $PLCSpare3->B2 = $Hour->B2
2229:      $PLCSpare3->B3 = $Hour->B3
2230:      $PLCSpare3->B4 = $Hour->B4
2231:      $PLCSpare3->B5 = $Hour->B5
2232:      $PLCSpare3->B6 = $Hour->B6
2233:      $PLCSpare3->B7 = $Hour->B7
2234:      'minute bits 8-15
2235:      $PLCSpare3->B8 = $Minute->B0
2236:      $PLCSpare3->B9 = $Minute->B1
2237:      $PLCSpare3->B10 = $Minute->B2
2238:      $PLCSpare3->B11 = $Minute->B3
2239:      $PLCSpare3->B12 = $Minute->B4
2240:      $PLCSpare3->B13 = $Minute->B5
```

```
2241:      $PLCSpare3->B14 = $Minute->B6
2242:      $PLCSpare3->B15 = $Minute->B7
2243:      'second bits 16-23
2244:      $PLCSpare3->B16 = $Second->B0
2245:      $PLCSpare3->B17 = $Second->B1
2246:      $PLCSpare3->B18 = $Second->B2
2247:      $PLCSpare3->B19 = $Second->B3
2248:      $PLCSpare3->B20 = $Second->B4
2249:      $PLCSpare3->B21 = $Second->B5
2250:      $PLCSpare3->B22 = $Second->B6
2251:      $PLCSpare3->B23 = $Second->B7
2252:  End If
2253:  'wait for PLC to confirm update or timeout
2254:  StartTime = $GetTickCount() 'milliseconds up to 2^32 since init of OS
2255:  Do Until $nPLCClockUpdated = 1 Or ( $Abs( $GetTickCount() - StartTime ) >= 30000 )
2256:      '$myText = "wait4 n=1 " & $Abs( $GetTickCount() - StartTime )
2257:  Loop
2258:  $myText = "UpdatePLCClock( ) n=1 done"
2259:  'move to next
2260:  $hnUpdateClock = 2
2261:
2262:
2263:  'set the year
2264:  If $hnUpdateClock = 2 Then
2265:      'reset
2266:      $PLCSpare3 = 0
2267:      'date: year
2268:      $PLCSpare3->B26 = 1      'bits 31-24 = 4
2269:      'year
2270:      $PLCSpare3->B0 = $Year->B0
2271:      $PLCSpare3->B1 = $Year->B1
2272:      $PLCSpare3->B2 = $Year->B2
2273:      $PLCSpare3->B3 = $Year->B3
2274:      $PLCSpare3->B4 = $Year->B4
2275:      $PLCSpare3->B5 = $Year->B5
2276:      $PLCSpare3->B6 = $Year->B6
2277:      $PLCSpare3->B7 = $Year->B7
2278:      $PLCSpare3->B8 = $Year->B8
2279:      $PLCSpare3->B9 = $Year->B9
2280:      $PLCSpare3->B10 = $Year->B10
2281:      $PLCSpare3->B11 = $Year->B11
2282:      $PLCSpare3->B12 = $Year->B12
```

```
2283:      $PLCSpare3->B13 = $Year->B13
2284:      $PLCSpare3->B14 = $Year->B14
2285:      $PLCSpare3->B15 = $Year->B15
2286: End If
2287: 'wait for PLC to confirm update or timeout
2288: StartTime = $GetTickCount() 'milliseconds up to 2^32 since init of OS
2289: Do Until $nPLCClockUpdated = 2 Or ( $Abs( $GetTickCount() - StartTime ) >= 30000 )
2290:     '$myText = "wait4 n=2 " & $Abs( $GetTickCount() - StartTime )
2291: Loop
2292: $myText = "UpdatePLCClock( ) n=2 done"
2293: 'move to next
2294: $hnUpdateClock = 3
2295:
2296:
2297: 'set the month
2298: If $hnUpdateClock = 3 Then
2299:     'reset
2300:     $PLCSpare3 = 0
2301:     'date: month
2302:     $PLCSpare3->B25 = 1 'bits 31-24 = 3
2303:     $PLCSpare3->B24 = 1
2304:     'month
2305:     $PLCSpare3->B0 = $Month->B0
2306:     $PLCSpare3->B1 = $Month->B1
2307:     $PLCSpare3->B2 = $Month->B2
2308:     $PLCSpare3->B3 = $Month->B3
2309:     $PLCSpare3->B4 = $Month->B4
2310:     $PLCSpare3->B5 = $Month->B5
2311:     $PLCSpare3->B6 = $Month->B6
2312:     $PLCSpare3->B7 = $Month->B7
2313: End If
2314: 'wait for PLC to confirm update or timeout
2315: StartTime = $GetTickCount() 'milliseconds up to 2^32 since init of OS
2316: Do Until $nPLCClockUpdated = 3 Or ( $Abs( $GetTickCount() - StartTime ) >= 30000 )
2317:     '$myText = "wait4 n=3 " & $Abs( $GetTickCount() - StartTime )
2318: Loop
2319: $myText = "UpdatePLCClock( ) n=3 done"
2320: 'move to next
2321: $hnUpdateClock = 4
2322:
2323:
2324: 'set the day of the month
```

```
2325:  If $hnUpdateClock = 4 Then
2326:      'reset
2327:      $PLCSpare3 = 0
2328:      'date: day of month
2329:      $PLCSpare3->B25 = 1      'bits 31-24 = 2
2330:      'day
2331:      $PLCSpare3->B0 = $Day->B0
2332:      $PLCSpare3->B1 = $Day->B1
2333:      $PLCSpare3->B2 = $Day->B2
2334:      $PLCSpare3->B3 = $Day->B3
2335:      $PLCSpare3->B4 = $Day->B4
2336:      $PLCSpare3->B5 = $Day->B5
2337:      $PLCSpare3->B6 = $Day->B6
2338:      $PLCSpare3->B7 = $Day->B7
2339:  End If
2340:  'wait for PLC to confirm update or timeout
2341:  StartTime = $GetTickCount()      'milliseconds up to 2^32 since init of OS
2342:  Do Until $nPLCClockUpdated = 4 Or ( $Abs( $GetTickCount() - StartTime ) >= 30000 )
2343:      '$myText = "wait4 n=4 " & $Abs( $GetTickCount() - StartTime )
2344:  Loop
2345:  'done
2346:  'PLC sets hnUpdateClock = 0, which effectively disables the PLC chart
2347:  $myText = "PLC clock update done"
2348:
2349: End Sub      'UpdatePLCClock( )
2350: ///////////////////////////////////////////////////
2351:
2352:
2353: ///////////////////////////////////////////////////
2354: ///////////////////////////////////////////////////      ConvertPLCClock( )      ///////////////////////////////////////////////////
2355: ///////////////////////////////////////////////////
2356: '/REV 3.1.0/ newly created
2357: '//convert PLCReserved7 and PLCReserved8 from integers, to hex representations and parse out according to Entivity documentation
2358: '//Called by "MsgBox to display PLC calendar and clock values" button on "PAG Utilities.scr"
2359: Sub ConvertPLCClock( )
2360:     $ConvertPLCClockSub      'marker for IWS Xref
2361:
2362:     Dim WW, SS, MM, HH, DD, NN, YYYY, display
2363:
2364:     'tags come as integer from OPC, but values need to be evaluated in Hex
2365:     'VBScript Long Integer:  2147483647 = hex 7FFFFFFF      to      -2147483648 = hex 80000000
2366:     'to get WW, need to mask off using bitwise AND operator with hex FF000000, then shift right by the power of 2 bit location
```

```
2367:      'but since the longs are split across zero, have to do 2s inverse complement
2368:      'cause FF000000 = 4278190080 which overflows the upper end of the long integer for VBScript of 2147483647
2369:      'to get the 2s inverse complement of FF000000: invert bits, convert to decimal, add 1, change sign --> = -16777216
2370:
2371:      'parse out $PLCReserved7 = WWSSMMHH as Hex; Weekday seconds minutes hours
2372:      WW = ( $PLCReserved7 And -16777216 )/(2^24)
2373:      'need FF0000 to get SS
2374:      SS = ( $PLCReserved7 And 16711680 )/(2^16)
2375:      If SS < 10 Then SS = "0" & SS
2376:      'need FF00 to get MM
2377:      MM = ( $PLCReserved7 And 65280 )/(2^8)
2378:      If MM < 10 Then MM = "0" & MM
2379:      'need FF to get HH
2380:      HH = ( $PLCReserved7 And 255 )
2381:      If HH < 10 Then HH = "0" & HH
2382:
2383:      'parse out $PLCReserved8 = DDMMYYYY as Hex; day month year; use NN for MM here since can't reuse MM, so DDNNYYYY
2384:      DD = ( $PLCReserved8 And -16777216 )/(2^24)
2385:      If DD < 10 Then DD = "0" & DD
2386:      NN = ( $PLCReserved8 And 16711680 )/(2^16)
2387:      If NN < 10 Then NN = "0" & NN
2388:      YYYY = ( $PLCReserved8 And 65535 )
2389:
2390:      display = "Snapshot: " & vbCrLf _
2391:      & NN & "/" & DD & "/" & YYYY & vbCrLf & HH & ":" & MM & ":" & SS & vbCrLf & WeekdayName( WW, False, 0 ) _
2392:      & vbCrLf & vbCrLf & "The PLC clock time may be delayed based on OPC connectivity update rates." & vbCrLf _
2393:      & "The PLC clock is synchronized to the WinXP clock either daily during a runtime day change, or " & vbCrLf _
2394:      & "upon the first runtime startup of a day, or the DailyRunScript button here."
2395:      MsgBox display, vbOKOnly + vbInformation, "PLC system calendar and clock"
2396:
2397: End Sub 'ConvertPLCClock( )
2398: //////////////////////////////////////
2399:
2400:
2401: 'End Global Procedures
```

Workspace Toolbar: Graphics Tab>Screens

Screen Logic and Scripts

The following section will show each screen, its screen script, and attempt to address any major button scripts. For the most part, button scripts call a global procedure.

Note that the screen script files are integral in the screen definition files located at
 \<App Dir>\Screen\<ScreenName.scr>.

Reference the screen map to view screen dependency and navigation.

Screen objects, properties, definitions, and terminology:

An object on the screen can be a line, box, rectangle, circle/oval, button, or text.

A dynamic property can be placed on objects. Dynamic properties are Commands, Bar Graphs, Text Input, Color, Position, Size, or Rotation.

Command dynamics can include On Down, On While, and On Up; executing the dynamic when the button press action is performed. For the most part, the On Up dynamic was used throughout the project. The benefit of the On Up dynamic is that the action will not execute until the user releases the button. With the On Up dynamic, the user can press a button, but slide off the button to effectively cancel the button press. Notice that there are additional command options when the 'Config...' button is selected.

Small Command window

Config... view of Command window

Active objects are those written by the software manufacturer that are complex objects, such as Alarm/Event, Trend, ActiveX, Grid, Combo Box, Check Box, Radio Button, List Box, Smart Message, or Pushbutton. Active objects provide an interface to change both object properties and dynamic properties.

Screen Map

Mold Heating and Cooling Pump Package Operator Interface (MHC PP OI)

Tree representation of Screen Maps for revision 3.1.0, March 2009

Login: Operator

- Start Up
 - Splash Screen
 - Status Bar
 - Navigation
 - Recipe Viewer
 - Open Recipe Popup
 - View Segment Grapher
 - Input Serial Number Dialog
 - Wait Dialog
 - Help-Recipe Viewer
 - Recipe Status
 - Help-Recipe Status
 - Trend Graph
 - Historical Trend Graph
 - Manual Constant Temperature
 - MTL Tuning Routine
 - Help-MTL Tuning Routine
 - Help-Other Recipe Help Page2
 - Help-Manual Constant Temperature
 - Help-Other Recipe Help Page2
 - Splash Screen

Login: Engineer

- Start Up
 - Splash Screen
 - Status Bar
 - Navigation
 - Recipe Viewer
 - Open Recipe Popup
 - View Segment Grapher
 - Input Serial Number Dialog
 - Wait Dialog
 - Help-Recipe Viewer
 - Recipe Status
 - Help-Recipe Status
 - Trend Graph
 - Historical Trend Graph
 - Manual Constant Temperature
 - MTL Tuning Routine
 - Help-MTL Tuning Routine
 - Help-Other Recipe Help Page2
 - Help-Manual Constant Temperature
 - Help-Other Recipe Help Page2
 - Recipe Editor Segments
 - Open Recipe Popup
 - Input Time SP Dialog
 - Change Control Mode Dialog
 - Segment Operations Dialog
 - View Segment Grapher
 - Input Serial Number Dialog
 - Wait Dialog
 - Help-Recipe Editor Segments
 - Help-Recipe Editor Constants
 - Help-Other Recipe Help Page 1
 - Help-Other Recipe Help Page2
 - Recipe Editor Constants
 - Open Recipe Popup
 - Change Control Mode Dialog
 - Input Serial Number Dialog
 - Wait Dialog
 - MTL Tuning Routine
 - Help-MTL Tuning Routine
 - Help-Other Recipe Help Page2
 - Help-Recipe Editor Constants
 - Help-Recipe Editor Segments
 - Help-Other Recipe Help Page 1
 - Help-Other Recipe Help Page2
 - Event Viewer
 - Splash Screen

Login: PAG

- Start Up
 - Splash Screen
 - Status Bar
 - Navigation
 - Recipe Viewer
 - Open Recipe Popup
 - View Segment Grapher
 - Input Serial Number Dialog
 - Wait Dialog
 - Help-Recipe Viewer
 - Recipe Status
 - Help-Recipe Status
 - Trend Graph
 - Historical Trend Graph
 - Manual Constant Temperature
 - MTL Tuning Routine
 - Help-MTL Tuning Routine
 - Help-Other Recipe Help Page2
 - Help-Manual Constant Temperature
 - Help-Other Recipe Help Page2
 - Recipe Editor Segments
 - Open Recipe Popup
 - Input Time SP Dialog
 - Change Control Mode Dialog
 - Segment Operations Dialog
 - View Segment Grapher
 - Input Serial Number Dialog
 - Wait Dialog
 - Help-Recipe Editor Segments
 - Help-Recipe Editor Constants
 - Help-Other Recipe Help Page 1
 - Help-Other Recipe Help Page2
 - Recipe Editor Constants
 - Open Recipe Popup
 - Change Control Mode Dialog
 - Input Serial Number Dialog
 - Wait Dialog
 - MTL Tuning Routine
 - Help-MTL Tuning Routine
 - Help-Other Recipe Help Page2
 - Help-Recipe Editor Constants
 - Help-Recipe Editor Segments
 - Help-Other Recipe Help Page 1
 - Help-Other Recipe Help Page2
 - Event Viewer
 - PAG Utilities
 - Splash Screen

Mold Heating and Cooling Pump Package Operator Interface (MHC PP OI) screen map

STARTUP screen group includes NAVIGATION, STATUS, and SPLASH screens

NAVIGATION screen

File name: NAVIGATION.scr

This screen is most always visible and accessible.

Button scripts and properties:

All buttons are Radio Buttons and use Color and Command dynamics.

A Disable property is also used on the Radio Buttons. The color is green when pressed. The Command dynamic uses the On Up VBScript whose logic uses If-Then statements to call a Global Procedure, or open a screen.

The three sections of buttons are shown based on who is logged in during runtime. Admin/PAG has all buttons and all functions available. Engineering can also access the Operator buttons.

This screen does not use a screen script.

STATUS screen

File name: STATUS.scr

The status screen gives a quick reference to the recipe and PLC status, an avenue for logging in users, and the exit button to shutdown.

Button scripts and properties:

- The Recipe Loaded indicator uses a rectangle object with dynamic text as the caption, and a Color dynamic.
- The PLC OK/Dead indicator uses a Smart Message active control.
- The ESTOP indicator is a group containing a circle and static text objects. The group uses a Position dynamic; the objects are only shown when the ESTOP is active. The circle uses a Color Dynamic.
- The log in buttons are rectangle objects that use a Command dynamic and where necessary a Disable property. The command dynamic uses a keypad to obtain the user and password as necessary, then executes an InduSoft function if successful.
- The 'Exit MHC' button is a group containing a button, rectangle, and bitmap. The rectangle is used for the static text of the button, because it was hard to place the text otherwise. The bitmap is the icon on the button. The button uses a VBScript On Up Command dynamic to ask the user if they want to shutdown, and if yes, then shuts down the application and logs the user off of Windows.

This screen uses a screen script for additional logic.

```
1: 'Variables available on this screen can be declared and initialized here.
2:
3:
4: 'Procedures available on this screen can be implemented here
5:
6:
7: 'This procedure is executed just once when this screen is open.
8: Sub Screen_OnOpen()
9:
10:    'enable OPC comm for screen variables
11:    $OPCCommStatus = 0
12:
13: End Sub
14:
15: 'This procedure is executed continuously while this screen is open.
16: Sub Screen_WhileOpen()
17:
18: End Sub
19:
20: 'This procedure is executed just once when this screen is closed.
21: Sub Screen_OnClose()
22:
23:    'disable OPC comm for screen variables
24:    $OPCCommStatus = 1
25:
26: End Sub
```

SPLASH screen

File name: SPLASH.scr

This screen uses a screen script for additional logic.

```
1: 'Variables available on this screen can be declared and initialized here.
2:
3: 'Procedures available on this screen can be implemented here
4:
5:
6: 'This procedure is executed just once when this screen is open.
7: Sub Screen_OnOpen()
8:
9:     'trigger script #2
10:     $TriggerSplashScreen = 1
11:
12: End Sub
13:
14: 'This procedure is executed continuously while this screen is open.
15: Sub Screen_WhileOpen()
16:
17: End Sub
18:
19: 'This procedure is executed just once when this screen is closed.
20: Sub Screen_OnClose()
21:
22:     'this will end the DO Loop in script2
23:     $TriggerSplashScreen = 0
24:
25: End Sub
```


Recipe Viewer screen

File name: Recipe Viewer.scr

RECIPE VIEWER - FOR OPERATOR VERIFICATION AND TO START RECIPES

Select Recipe

Start Recipe

Close Recipe

Run Preheat Skip Preheat

ZONE

Zone 1

Zone 2

Zone 3

Zone 4

SEGMENTS

Seg 0-9 View Segment Grapher

RECIPE NAME
myRecipe

PART NUMBER(S)
45678, 123456-78

RECIPE DESCRIPTION
This part is used to make a whachamigger for the whachamacallit.

Est. Cycle Time
00:33:00

SEG#	TIME	TEMP	Ramp	Ramp Rate	Segment Comment
0	300	100	☐	0.000000	This segment is used to preheat the mold.
1	300	175	☒	0.250000	Ramp up the temperature, then place the part in the mol...
2	600	275	☐	0.000000	Process the part for 5 minutes.
3	180	290	☐	0.000000	Process the part at a slightly higher temperature for 3 ...
4	600	100	☒	-0.316667	Ramp to cooldown and remove the part.
5	0	50	☐	0.000000	
6	0	50	☐	0.000000	
7	0	50	☐	0.000000	
8	0	50	☐	0.000000	
9	0	50	☐	0.000000	

Manual Preheat

Cycle Abort Temp SP

Last SP

Mold PreHT Temp Limits

Upper Limit:

Lower Limit:

Control Mode

Fluid Recipe Profile

Fluid Consts:

K_P 12.0

K_I 1.5

K_D 0.1

Mold Dynamic Constants:

Dead Time (sec) θ

Time Constant (sec) τ

Closed Loop Time Constant (sec) λ

Entered Gain K' Default: 0.33

Steady State Gain K_m

This screen is used by the operator to view and start a recipe. The recipe cannot be changed on this screen. The only buttons available to the user are on the left side column.

Button scripts and properties:

- The 'Select Recipe' button uses an On Up VBScript Command dynamic with If-Then logic to either open a screen or call a global procedure; the command dynamic also uses the Disable property. It is also grouped with the bitmap icon.
- The 'View PLC' button uses an On Up VBScript Command dynamic to call a global procedure; it also uses the Disable property. It is also grouped with the bitmap icon.
- The 'Close Recipe' button uses an On Up VBScript Command dynamic with If-Then logic to either change a tag or call a global procedure; it also uses the Disable property. It is also grouped with the bitmap icon.
- The 'Run Preheat' and 'Skip Preheat' buttons are radio buttons that have Color dynamics. The Skip Preheat button uses an On Up VBScript Command dynamic to display a message box.
- The zone buttons are radio buttons that have Color dynamics. The zone 3 and 4 buttons also use Position dynamics to show/hide the buttons depending on the machine state.
- The 'Seg 0-9' and 'Seg 10-19' buttons are radio buttons that have Color dynamics.

- The 'View Segment Grapher' button is a group containing a button and bitmap. The button uses an On Up VBScript Command dynamic to call a global procedure when pressed. It also uses the Disable property.
- The 'Load/Start Recipe' button is a group containing a button and bitmap. The button uses an On Up VBScript Command dynamic to execute logic. A message box is presented to the user and if yes, will either start the download process by changing a tag value which calls a Task Script, or open the 'Input SN Dialog' screen; the action depends on the preheat mode of the recipe. The command dynamic also uses the Disable property.
- The help icon in the upper right hand corner of the screen is a bitmap object with an On Up VBScript Command dynamic that calls the help documentation for this screen.

This screen uses a screen script for additional logic.

```
1: 'Variables available on this screen can be declared and initialized here.
2:
3:
4: 'Procedures available on this screen can be implemented here
5:
6:
7: 'This procedure is executed just once when this screen is open.
8: Sub Screen_OnOpen()
9:
10:  '/REV 3.1.0/ change logic to open what is on PLC, if PLC is not empty; esle open the last recipe loaded to
    the PLC
11:  '310      If $RecipeName<>" " And $RecipeIsSaved Then
12:  '310      'reopen the last viewed to repopulate the recipe tags, if navigated from a screen that changed
    them
13:  '310      Call RecipeOpen( $RecipeLastViewed )
14:  '310      $RecipeOpenSub      'marker for IWS Xref
15:  '310      End If
16:  If $hwRecipeName<>" " And $hwRecipeName<>"empty" Then
17:  'call global procedure
18:  Call RecipeOpenPLC( ) : $RecipeOpenPLCSub      'marker for IWS Xref tool
19:  Else
20:  If $RecipeLastLoaded<>" " Then
21:  'open last recipe that was run and loaded to controller, $RecipeLastLoaded is set in GP
    RecipeLoadToPLC( )
22:  Call RecipeOpen( $RecipeLastLoaded )      : $RecipeOpenSub      'marker for IWS Xref tool
23:  Else
24:  $RecipeName = " "
25:  End If
26: End If
27:
28: 'When moving btwn 2 and 4 zone recipes, this could cause confusion, so default
29: If $ZoneSelectedRadio>2 And $hn2_4zoneMachineSwitch=2 Then
30:  $ZoneSelectedRadio = 2
31: End If
32:
33: 'enable OPC comm for screen variables
34: $OPCCommRecipeViewer = 0
35: 'once the screen is open this tag will recalculate, when the tag updates, the grid will update, this is
    necessary to get
36: ' the grid to reformat so that its alignment will match the Seg# column
37: $RecipeStartIndex = -1
38:
```

```
39: End Sub
40:
41:
42: 'This procedure is executed continuously while this screen is open.
43: Sub Screen_WhileOpen()
44:
45:     //Calculate the "RecipeStartIndex", this value provides a starting array pointer into the long recipe array
    for screen display
46:     $SegmentPagePointer = $ZoneSelectedRadio*20-20    '=0,20,40,60    '$ZoneSelectedRadio= 1=z1, 2=z2, 3=z3,
    4=z4
47:     $RecipeStartIndex = $SegmentPagePointer + $SegmentsDisplayed*10    '=0,10...60,70    '$SegmentsDisplayed=
    (0)=seg0-9 OR (1)=seg10-19
48:
49: End Sub
50:
51:
52: 'This procedure is executed just once when this screen is closed.
53: Sub Screen_OnClose()
54:
55:     'disable OPC comm for screen variables
56:     $OPCCommRecipeViewer = 1
57:     '/REV 3.1.0/ clear out recipe every time leave this screen, when open another screen dealing with recipes,
    it will re-populate the recipe name accordingly
58:     '310     'populate the recipe last viewed
59:     '310     '$RecipeLastViewed = $RecipeName
60:     $RecipeName = ""
61:
62: End Sub
63:
64:
65: 'End Screen Script
```

Recipe Status screen

File name: Diagnostic.scr

This screen provides the status of a currently running recipe as well as control status of the machine.

Controls available on this screen are the Dry Cycle and Abort Cycle buttons, which set an application tag. The only other control is if a zone is running as a constant temperature zone, the temperature set point may be adjusted here via the Command dynamic On Up VBScript.

Button scripts and properties:

- The 'Dry Cycle' button is a check box object with a Color dynamic. Its caption is dynamic.
- The 'Abort Cycle' button has a dynamic caption with a Color dynamic, and a On Up VBScript Command dynamic that sets tags and uses the Disable property.
- The only other buttons available are the temperature set points. These controls become available if the zone is configured as a constant temperature zone. The button is a rectangle object with a dynamic caption, using a Color dynamic, and a Command dynamic. The temperature set point may be adjusted via the Command dynamic On Up VBScript.
- The help icon in the upper right hand corner of the screen is a bitmap object with an On Up VBScript Command dynamic that calls the help documentation for this screen.

This screen uses a screen script for additional logic.

```
1: 'Variables available on this screen can be declared and initialized here.
2: Dim OneShot
3:
4: 'Procedures available on this screen can be implemented here
5:
6:
7: 'This procedure is executed just once when this screen is open.
8: Sub Screen_OnOpen()
9:
10:    'enable OPC comm for screen variables
11:    $OPCCommDiagnostic = 0
12:
13:    '$GetTickCount() milliseconds up to 2^32 since init of OS
14:    $ScriptTimeStatus = $GetTickCount()
15:
16: End Sub
17:
18: 'This procedure is executed continuously while this screen is open.
19: Sub Screen_WhileOpen()
20:
21:    '/REV 3.1.0/ newly added; Run logic every X milliseconds
22:    If ( $GetTickCount() - $ScriptTimeStatus ) >= 500 Then
23:        OneShot = True
24:
25:        Do While OneShot 'only reason for DO Loop is so I can exit the loop prior to analyzing another IF
                           statement, but failsafe to a oneshot loop
26:
27:            'set value of    $CycleStatus to display appropriate smart message box on screen
28:            ' 0 = ..?...      ' 4 = PRHT Wait Mold Limits
29:            ' 1 = IDLE        ' 5 = RFCS
30:            ' 2 = Initiated   ' 6 = RUN
31:            ' 3 = PREHEAT     ' 7 = ABORT
32:            If $hfCycleAbort Then $CycleStatus = 7 : Exit Do 'use Not $sCycleNormal instead of $hfCycleAbort??
33:            If $sIDLE Then $CycleStatus = 1 : Exit Do
34:            If $sCycleINIT And ( $sMoldTemperLimitsBreachZ[1] Or $sMoldTemperLimitsBreachZ[2] _
35:                Or ( $sFourZoneMachine And $sMoldTemperLimitsBreachZ[3] ) Or ( $sFourZoneMachine And
                           $sMoldTemperLimitsBreachZ[4] ) ) Then
36:                $CycleStatus = 4
37:                Exit Do
38:            End If
39:            If $sCycleINIT Then $CycleStatus = 2 : Exit Do
40:            If $sPREHEAT And ( $sMoldTemperLimitsBreachZ[1] Or $sMoldTemperLimitsBreachZ[2] _
```

```
41:         Or ( $sFourZoneMachine And $sMoldTemperLimitsBreachZ[3] ) Or ( $sFourZoneMachine And
$sMoldTemperLimitsBreachZ[4] ) ) Then
42:         $CycleStatus = 4
43:         Exit Do
44:     End If
45:     If $sPREHEAT Then $CycleStatus = 3 : Exit Do
46:     If $sRFCS Then $CycleStatus = 5 : Exit Do
47:     If $sRUN Then $CycleStatus = 6 Else $CycleStatus = 0 : Exit Do
48:     'if for some reason, did not Exit previously, then this is failsafe
49:     OneShot = False
50:
51:     Loop 'a one cycle do loop
52:
53:     $ScriptTimeStatus = $GetTickCount() 'reset tag for next loop
54: End If
55:
56:
57: End Sub
58:
59: 'This procedure is executed just once when this screen is closed.
60: Sub Screen_OnClose()
61:
62:     'disable OPC Comm for screen variables
63:     $OPCCommDiagnostic = 1
64:
65: End Sub
```

Trend Graph screen

File name: Trend.scr

The trend screen provides a temperature versus time graphical representation of the currently running cycle. The trend is an Active Object which has many built in controls available, such as zoom, layout, and pen properties.

Custom buttons available on this screen allow the user to adjust the trend cursor by inputting a specific location, or moving it to either end of the trend. The user can also adjust the Temperature scale, and reset the scale to default. The user also has the ability to reset the trend options and views, such as pens visible, zooms, etc., back to the default setup using the 'Format Graph' button. These buttons use simple VBScripts to set a tag, or obtain an input value from the user.

Button scripts and properties:

- The 'Cursor' section contains an input field, and arrow buttons. The input field button is a rectangle object with a dynamic caption and uses an On Up VBScript Command dynamic to launch a keypad for a new cursor position. The arrow button is a group containing a rectangle object and a drawn arrow object. An On Down VBScript Command dynamic is used to either set the cursor position to 0% or 100%.
- The 'Format Graph' button uses an On Up VBScript Command dynamic to set some tag values which will reformat the graph settings to their default.

- The 'Scale' section contains two input fields for the temperature scale, and a reset button. The input field buttons are rectangle objects with dynamic captions and use On Up VBScript Command dynamics to launch a keypad for a new scale value. Logic is also used to make sure that the max is greater than the min. The 'Reset' button uses an On Up VBScript Command dynamic to set the min and max scale values to their default setting.
- This screen uses a screen script for additional logic.

```
1: 'Variables available on this screen can be declared and initialized here.
2:
3:
4: 'Procedures available on this screen can be implemented here
5:
6:
7: 'This procedure is executed just once when this screen is open.
8: Sub Screen_OnOpen()
9:
10:    'enable OPC comm for screen variables
11:    $OPCCommTrend = 0
12:    'reload/refresh trend data.  Have to do this when navigating away from screen and then back.
13:    $TrendGraphUpdateManual = 1
14:
15: End Sub
16:
17: 'This procedure is executed continuously while this screen is open.
18: Sub Screen_WhileOpen()
19:
20: End Sub
21:
22: 'This procedure is executed just once when this screen is closed.
23: Sub Screen_OnClose()
24:
25:    'disable OPC comm for screen variables
26:    $OPCCommTrend = 1
27:
28: End Sub
```

TREND GRAPH SETTINGS for screen "Trend.scr"

AXES DIALOG SETUP

Cursor [X]

☒ Enable Position (0-100):

Color: Value output:

Position [X]

Placement: Orientation:
☒ Horizontal
☐ Vertical

Direction:

Format: Numeric [X]

Scale
 Width: Decimals:

☐ Truncate zeroes in the decimals

Number of Labels:

LEGEND DIALOG SETUP

Label="Pen Marker"=118 left, Current=68 center, Cursor=68 center, EU="Units"=73 center, Desc=348 left

POINTS DIALOG SETUP

Points

Point	Label	Color	Data Source	Tag/Field
1	Z1 PID SP		{hstBatchFileName}	rSP_PID_...
2	Z2 PID SP		{hstBatchFileName}	rSP_PID_...
3	{IF(sFourZoneMachine, "Z3 PID SP", "")}		{hstBatchFileName}	rSP_PID_...
4	{IF(sFourZoneMachine, "Z4 PID SP", "")}		{hstBatchFileName}	rSP_PID_...
5	Z1 FLUID TEMP		{hstBatchFileName}	hrTC_Flui...
6	Z2 FLUID TEMP		{hstBatchFileName}	hrTC_Flui...
7	{IF(sFourZoneMachine, "Z3 FLUID TEMP", "")}		{hstBatchFileName}	hrTC_Flui...
8	{IF(sFourZoneMachine, "Z4 FLUID TEMP", "")}		{hstBatchFileName}	hrTC_Flui...
9	Z1 MTL SP		{hstBatchFileName}	rSP_MTL_...
10	Z2 MTL SP		{hstBatchFileName}	rSP_MTL_...

< [] >

OK Cancel

Points

Point	Label	Color	Data Source	Tag/Field
8	{IF(sFourZoneMachine, "Z4 FLUID TEMP", "")}		{hstBatchFileName}	hrTC_Flui...
9	Z1 MTL SP		{hstBatchFileName}	rSP_MTL_...
10	Z2 MTL SP		{hstBatchFileName}	rSP_MTL_...
11	{IF(sFourZoneMachine, "Z3 MTL SP", "")}		{hstBatchFileName}	rSP_MTL_...
12	{IF(sFourZoneMachine, "Z4 MTL SP", "")}		{hstBatchFileName}	rSP_MTL_...
13	Z1 MOLD TEMP		{hstBatchFileName}	hrTC_Mold...
14	Z1 MOLD TEMP		{hstBatchFileName}	hrTC_Mold...
15	{IF(sFourZoneMachine, "Z3 MOLD TEMP", "")}		{hstBatchFileName}	hrTC_Mold...
16	{IF(sFourZoneMachine, "Z4 MOLD TEMP", "")}		{hstBatchFileName}	hrTC_Mold...
17			Tag	

< [] >

OK Cancel

Points

Point	Label	Color	Tag/Field	Min	Max	Style	Options	Hide
1	Z1 PID SP		rSP_PID_Z[1]				SP if ru...	
2	Z2 PID SP		rSP_PID_Z[2]				SP if ru...	
3	{IF(sFour...		rSP_PID_Z[3]				SP if ru...	
4	{IF(sFour...		rSP_PID_Z[4]				SP if ru...	
5	Z1 FLUID ...		hrTC_FluidZ[1]				value o...	
6	Z2 FLUID ...		hrTC_FluidZ[2]				value o...	
7	{IF(sFour...		hrTC_FluidZ[3]				value o...	
8	{IF(sFour...		hrTC_FluidZ[4]				value o...	
9	Z1 MTL SP		rSP_MTL_Z[1]				SP if ru...	
10	Z2 MTL SP		rSP_MTL_Z[2]				SP if ru...	

OK Cancel

Points

Point	Label	Color	Tag/Field	Min	Max	Style	Options	Hide
8	{IF(sFourZ...		hrTC_FluidZ[4]				value o...	
9	Z1 MTL SP		rSP_MTL_Z[1]				SP if ru...	
10	Z2 MTL SP		rSP_MTL_Z[2]				SP if ru...	
11	{IF(sFourZ...		rSP_MTL_Z[3]				SP if ru...	
12	{IF(sFourZ...		rSP_MTL_Z[4]				SP if ru...	
13	Z1 MOLD T...		hrTC_MoldZ[1]				value o...	
14	Z1 MOLD T...		hrTC_MoldZ[2]				value o...	
15	{IF(sFourZ...		hrTC_MoldZ[3]				value o...	
16	{IF(sFourZ...		hrTC_MoldZ[4]				value o...	
17							value o...	

OK Cancel

Points ✖

Point	Label	Options	Hide
1	Z1 PID SP	SP if running PID or is driven by MTL if running MTL, , , , 7200, , ...	☐
2	Z2 PID SP	SP if running PID or is driven by MTL if running MTL, , , , 7200, , ...	☐
3	{IF(sFo...	SP if running PID or is driven by MTL if running MTL, , , , 7200, , ...	☐
4	{IF(sFo...	SP if running PID or is driven by MTL if running MTL, , , , 7200, , ...	☐
5	Z1 FLUI...	value of thermocouple, , , , 600, , ...	☐
6	Z2 FLUI...	value of thermocouple, , , , 600, , ...	☐
7	{IF(sFo...	value of thermocouple, , , , 600, , ...	☐
8	{IF(sFo...	value of thermocouple, , , , 600, , ...	☐
9	Z1 MTL SP	SP if running MTL, , , , 7200, , ...	☐
10	Z2 MTL SP	SP if running MTL, , , , 7200, , ...	☐

< [] >

OK Cancel

Points ✖

Point	Label	Options	Hide
8	{IF(sFo...	value of thermocouple, , , , 600, , ...	☐
9	Z1 MTL SP	SP if running MTL, , , , 7200, , ...	☐
10	Z2 MTL SP	SP if running MTL, , , , 7200, , ...	☐
11	{IF(sFo...	SP if running MTL, , , , 7200, , ...	☐
12	{IF(sFo...	SP if running MTL, , , , 7200, , ...	☐
13	Z1 MOL...	value of thermocouple, , , , 600, , ...	☐
14	Z1 MOL...	value of thermocouple, , , , 600, , ...	☐
15	{IF(sFo...	value of thermocouple, , , , 600, , ...	☐
16	{IF(sFo...	value of thermocouple, , , , 600, , ...	☐
17		, , , , 7200, , ...	☐

< [] >

OK Cancel

DATA SOURCES DIALOG SETUP

The "Data Sources" dialog box is a standard Windows-style window with a title bar containing the text "Data Sources" and a close button (X). The main area contains several controls: a "Data Source:" label followed by a dropdown menu showing "Batch"; a "Source Type:" label followed by a dropdown menu showing "Batch"; an "X Axis field:" label followed by an empty text box; a "Reload:" label followed by a text box containing "hstBatchFileReload"; a "Max. Buffer:" label followed by a text box containing "1024"; a "Load Progress:" label followed by an empty text box; a checked checkbox labeled "Sort"; and a "Data Source Settings..." button. On the right side, there are four buttons: "New...", "Delete", "OK", and "Cancel", arranged vertically.

Data Source: Batch

Source Type: Batch

X Axis field:

Reload: hstBatchFileReload

Max. Buffer: 1024

Load Progress:

☒ Sort

Data Source Settings...

New...

Delete

OK

Cancel

The "Batch name" dialog box is a standard Windows-style window with a title bar containing the text "Batch name" and a close button (X). The main area contains a "Batch Name:" label followed by a text box containing "{hstBatchFileName}:". On the right side, there are two buttons: "Ok" and "Cancel", arranged vertically.

Batch Name: {hstBatchFileName}:

Ok

Cancel

TOOLBAR DIALOG SETUP

Toolbar

☒ Show toolbar

Command	Show	Activation Tag	Tooltip
Run/Stop	☐		
Period	☐		
Window Zoom	☒		
Horizontal Zoom	☒		
Vertical Zoom	☒		
Zoom In	☒		
Zoom Out	☒		
Cancel Zoom	☒		
Legend Properties	☐		

OK

Cancel

Toolbar

☒ Show toolbar

Command	Show	Activation Tag	Tooltip
Legend Properties	☐		
Pen Style	☐		
Add Pen	☐		
Remove Pen	☐		
Multiple Sections	☒		
Cursor	☒		
Auto Scale	☒		
Print	☐		Print

OK

Cancel

ADVANCED DIALOG SETUP

The image shows a software dialog box titled "Advanced" with a close button (X) in the top right corner. The dialog is organized into three main sections, each with a tab-like header:

- Run Mode Options:** Contains two text input fields: "Update trigger:" (empty) and "Update interval:" (containing "1"). Below these are four checkboxes:
 - ☒ Move to current time on run
 - ☐ Ignore X Filter
 - ☒ Retrieve bounding samples
 - ☒ Show Progress Bar
- Run-Time Config:** Contains three text input fields: "Save trigger:" (containing "TrendSaveSettings"), "Load trigger:" (containing "TrendLoadSavedSettings"), and "File Name:" (containing "TrendGraphSettings.stmp" and followed by a browse button "..."). Below these is a checked checkbox for "Auto Save".
- Custom Point Selection:** Contains a "Screen:" dropdown menu (set to "(Default)"), a "Point Number:" text input field (empty), and an "Add Indicator:" text input field (empty).

At the bottom of the dialog, there is a "VK:" label followed by an "AlphaNumeric" dropdown menu, and two checkboxes: ☒ Ext translation and ☐ Auto Format. On the right side of the dialog, there are "OK" and "Cancel" buttons.

End of Trend Graph Settings for screen "Trend.scr"

Historical Trend Graph screen

File name: TrendHistorical.scr

The historical trend screen provides a temperature versus time graphical representation of historical cycles. The trend is an Active Object which has many built in controls available, such as zoom, layout, and pen properties.

Button scripts and properties:

- The 'Open Historical Trend Graph' button uses an On Up VBScript Command dynamic to launch a user dialog to select the historical file, then load it into the trend object.
- The other buttons available on this screen are the same as those described on the real-time trend.

This screen uses a screen script for additional logic.

```
1: 'Variables available on this screen can be declared and initialized here.
2:
3:
4: 'Procedures available on this screen can be implemented here
5:
6:
7: 'This procedure is executed just once when this screen is open.
8: Sub Screen_OnOpen()
9:
10:   If $hstBatchFileNameHist <> "" Then
11:     'refresh and reformat the screen if have already loaded a historical file, or reload the last one that
    was opened
12:     $TrendGraphUpdateHist = 1
13:   End If
14:
15: End Sub
16:
17: 'This procedure is executed continuously while this screen is open.
18: Sub Screen_WhileOpen()
19:
20: End Sub
21:
22: 'This procedure is executed just once when this screen is closed.
23: Sub Screen_OnClose()
24:
25: End Sub
```

Manual Constant Temperature screen

File name: Non-Recipe ConstTemper.scr

Operator View

NON-RECIPE CONSTANT TEMPERATURE CONTROL, FLUID OR MOLD				
	Zone 1	Zone 2	Zone 3	Zone 4
Control Mode	Fluid Const Temp	Fluid Const Temp	Fluid Const Temp	Fluid Const Temp
Const Temper SP	100	200	170	170
Mold Temp	TC OK 79.6	TC OK 77.5	TC OK 77.2	TC OK 78.2
Fluid Temp	TC OK 220.3	TC OK 80.9	TC OK 79.5	TC OK 76.1
Valve Output %	0	100	100	100
Hidden PID Consts Defined	YES	YES	YES	YES
Hidden MTL Consts Defined				

Engineering View

NON-RECIPE CONSTANT TEMPERATURE CONTROL, FLUID OR MOLD				
	Zone 1	Zone 2	Zone 3	Zone 4
Control Mode	Fluid Const Temp	Fluid Const Temp	Mold Const Temp	Mold Const Temp
Const Temper SP	220	80	80	65
Mold Temp	TC OK 80.8	TC OK 78.7	TC OK 78.4	TC OK 79.4
Fluid Temp	TC OK 231.3	TC OK 79.9	TC OK 80.5	TC OK 77.1
Valve Output %	88	0	50	82
PID Consts Defined	YES	NO	NO	YES
Fluid Consts:				
PID Default1 K_P	1.5	0.0	0.0	12.0
PID Default2 K_I	4.5	3.0	3.0	1.5
K_D	0.05	0.00	0.00	0.05
MTL Consts Defined			NO	YES
Dead Time (sec) θ			0	13
Time Constant (sec) τ			0	27
Closed Loop Time Constant (sec) λ			0	27
Entered Gain K' Default: 0.33			0.0000	0.3300
Steady State Gain K_m			0.0000	0.9940
Calculated Gain K_{cs}			0.0000	8.0394
P0			0.0000	0.3352
P1			0.0000	0.2453
P5			0.0000	0.6905
P8			0.0000	0.2284
P9			0.0000	0.0811

Zone 3 is under PID control until the Mold Constants are fully defined.

Run Metal Tuning Test to Establish Metal Constants

This screen provides the user to operate the system without a recipe. Control of the zones can only be accomplished by changing the 'Const Temp SP' field. This method of control would be useful when developing recipes or for products whose time/temperature profiles do not have critical timing. Notice that the screen appearance differs depending on who is logged in at the time this screen opens. Also note that the Operator still has the ability to access the areas that appear for an Engineer by pressing the 'Hidden' button to show and change the PID/MTL constants.

Button scripts and properties:

- The 'Hidden' buttons are pushbutton objects that change the state of an internal tag value when pressed. This tag is then used on the Position dynamic of the PID/MTL constants section of the screen to show/hide those controls. The buttons also use a Position dynamic to show/hide the button.
- The 'Control Mode' fields are rectangle objects with dynamic captions that call a global procedure via a VBScript On Up Command dynamic. These fields also use a Color dynamic.
- The 'Const Temp SP' fields call the keypad via a VBScript On Up Command dynamic. These fields also use a Color dynamic, and are grouped with Text Display and Position dynamics.
- The 'Kp, Ki, Kd' fields use dynamic text in the caption of the rectangle to format the number to be displayed. A VBScript On Up Command dynamic is used to launch the keypad for user input.
- The ' θ , τ , λ , K', Km' fields are a group containing rectangle and text objects. The rectangle is used to provide a background to the text as well as using a Color dynamic, and a Command dynamic. The VBScript On Up Command dynamic contains logic to change the value via a keypad input, and if successful will call a global procedure, and if not successful may notify the user. The text object is used to display the value of the field. A Position dynamic on the text is used to show/hide the text depending on the value of the condition; this is the reason a text object was used rather than displaying the value dynamically in the rectangle caption.
- The indicator to the right of the metal constants is an indicator to show how the zone is being controlled when defining metal constants. The indicator is a rectangle using static caption text for display, and uses a Position and Color dynamic.
- The 'PID Default1' and 'PID Default2' buttons use an On Up VBScript Command dynamic to call a global procedure.
- The 'Run Metal Tuning Test to Establish Metal Constants' button uses an On Up VBScript Command dynamic to call a global procedure, then open the Metal Tuning screen. The Disable property is also used on the Command dynamic.
- The help icon in the upper right hand corner of the screen is a bitmap object with an On Up VBScript Command dynamic that calls the help documentation for this screen.

This screen uses a screen script for additional logic.

```
1: 'Variables available on this screen can be declared and initialized here.
2:
3:
4: 'Procedures available on this screen can be implemented here
5:
6:
7: 'This procedure is executed just once when this screen is open.
8: Sub Screen_OnOpen()
9:
10:    'enable OPC comm for screen variables
11:    'this is different from the other screens, the OPC comm enable bit is enabled in Global Procedure
    CCModes2NRCT()
12:
13: End Sub
14:
15: 'This procedure is executed continuously while this screen is open.
16: Sub Screen_WhileOpen()
17:
18:
19: End Sub
20:
21: 'This procedure is executed just once when this screen is closed.
22: Sub Screen_OnClose()
23:
24:    'disable OPC comm for screen variables
25:    $OPCCommNonRecConstTemper = 1
26:
27: End Sub
```


Recipe Segments screen

File name: Recipe Editor-Segments.scr

RECIPE EDITOR - SEGMENT SETPOINT DEFINITION FOR ZONES

RECIPE NAME
myRecipe

RECIPE DESCRIPTION
This part is used to make a whachamigger for the whachamacallit.

PART NUMBER(S)
45678, 123456-78

Generate RTF Report File
for Recipe:
"myRecipe"

Start Recipe

View Segment Grapher

Est. Cycle Time
00:33:00

Control Mode
Fluid Recipe Profile

Cycle Abort Temp SP

Last SP

Upper Limit:
Mold PreHT Temp Limits
Lower Limit:

SEG#	TIME (sec)	TEMPERATURE (deg F)	RAMP? (Y / N)	RAMP RATE (deg F/sec)	SEGMENT COMMENT(S)
0	300	100			This segment is used to preheat the mold.
1	300	175	YES	0.250	Ramp up the temperature, then place the part in the mold after the segment has completed.
2	600	275	NO		Process the part for 5 minutes.
3	180	290	NO		Process the part at a slightly higher temperature for 3 minutes.
4	600	100	YES	-0.317	Ramp to cooldown and remove the part.
5	0				
6					
7					
8					
9					

PREHEAT

Manual Preheat Auto Preheat

Run Preheat Skip Preheat

ZONE

Zone 1 Zone 2

Zone 3 Zone 4

SEGMENTS

Seg 0-9 Seg 10-19

Insert Delete

This screen gives the Engineering user a means to manage recipes. Its parameters are coupled with the Recipe Editor Constants screen.

Button scripts and properties:

- There are several groups of buttons that have a common Position dynamic for the group.
- The 'New', 'Open', 'Close', 'Save', 'SaveAs', and 'Delete' buttons are button objects with Command dynamics and grouped with a bitmap icon. The On Up VBScript Command dynamics call global procedures. Some of the buttons in this group employ the Disable property on the command dynamic.
- The 'Recipe Description' and 'Part Number(s)' fields are rectangle objects with a dynamic captions. The rectangle uses an On Up VBScript Command dynamic to launch the keypad to capture the freeform text input.
- The 'Load/Start Recipe' button is the same button and has the same functionality as the button on the Recipe Viewer screen.
- The 'View Segment Grapher' button is the same button and has the same functionality as the button on the Recipe Viewer screen.
- The 'Control Mode' button is a rectangle object with a dynamic text caption. The rectangle also uses an On Up VBScript Command dynamic to open a separate screen to change the

control mode. The rectangle also uses a Color dynamic to display a unique pattern for a given control mode.

- The 'SetPoint' field becomes active (and segment fields hidden) if the control mode selected is a constant temperature mode. The field is a group of rectangle and text objects. The rectangle uses an On Up VBScript Command dynamic to call the keypad, and if the keypad entry is successful will call a global procedure. The rectangle also uses the Disable property and a Color dynamic. The text object displays the tag value and uses a Position dynamic to show/hide the value.
- The 'Cycle Abort Temp SP' button is a checkbox object, and uses the Disable property. The checkbox also uses a Color dynamic and an On Up VBScript Command dynamic to change a tag value.
- The Cycle Abort Temp SP field is a group containing a rectangle and text object. The rectangle has a dynamic caption, and uses a Color dynamic. The rectangle uses an On Up VBScript Command dynamic to launch the keypad, and also uses the Disable property. The text object displays the tag value and uses a Position dynamic.
- The 'PreHT Mold Temp Limits' button is a checkbox object, and uses the Disable property. The checkbox also uses a Color dynamic and an On Up VBScript Command dynamic to change a tag value and execute logic to compare values of segment 1 with the upper and lower limits.
- The 'Upper Limit;' and 'Lower Limit;' fields are a group containing a rectangle and text object. The rectangle uses a Color dynamic, and an On Up VBScript Command dynamic to launch the keypad and execute logic to make sure that the upper limit is greater than the lower limit. The text object displays the tag value and uses a Position dynamic.
- The indicator next to the Upper and Lower Limit fields (not shown) is a group of two rectangles. The background rectangle uses a Position and Color dynamic. The foreground rectangle uses a static caption with the same Position dynamic condition, and uses an On Up VBScript Command dynamic to adjust the Upper and Lower Limits as necessary.
- The 'Manual Preheat' and 'Auto Preheat' buttons in the 'Preheat' section are radio button objects. They use a Color dynamic and an On Up VBScript Command dynamic to set a tag value.
- The 'Run Preheat' and 'Skip Preheat' buttons in the 'Preheat' section are the same buttons and have the same functionality as those on the Recipe Viewer screen.
- The 'Zone' buttons are the same buttons and have the same functionality as those on the Recipe Viewer screen.
- The 'Seg 0-9' and 'Seg 10-19' buttons in the 'Segments' section are the same buttons and have the same functionality as those on the Recipe Viewer screen.
- The 'Insert' and 'Delete' buttons in the 'Segments' section use an On Up VBScript Command dynamic to set tags and open the Segment Operations screen.
- The 'Seg#' column are text objects, each with their own Command and Position dynamics. The On Up VBScript Command dynamic sets the value of a tag.
- The 'Time' column fields are rectangle objects with a dynamic caption. The rectangles use a Position and an On Up VBScript Command dynamic to set tag values and open the Input Dwell screen.
- The 'Temperature' column fields are rectangle objects with a dynamic caption. The rectangles use a Position and an On Up VBScript Command dynamic to set tag values, launch the keypad, and if the keypad entry is successful will trigger a Task Script. Note that the segment 1 field contains additional logic dealing with the Preheat Mold Temperature Limits.
- The 'Ramp?' column buttons are checkbox objects that use a dynamic caption, the Disable property, and a Color, Position, and Command dynamic. The On Up VBScript Command dynamic sets tag values and triggers a Task Script.

- The 'Segment Comment(s)' fields are rectangle objects with a dynamic caption. The rectangles use a Position and an On Up VBScript Command dynamic to set tag values and launch the keypad.
- The help icon in the upper right hand corner of the screen is a bitmap object with an On Up VBScript Command dynamic that calls the help documentation for this screen.

This screen uses a screen script for additional logic.

```
1: 'Variables available on this screen can be declared and initialized here.
2:
3:
4: 'Procedures available on this screen can be implemented here
5:
6:
7: 'This procedure is executed just once when this screen is open.
8: Sub Screen_OnOpen()
9:
10:    '/REV 3.1.0/ change IF logic
11:    '310    If $RecipeName<>"" And $RecipeIsSaved Then
12:    If $RecipeLastViewed<>"" And $RecipeIsSaved Then
13:        'reopen the last viewed to repopulate the recipe tags, if navigated from a screen that changed them
14:        Call RecipeOpen( $RecipeLastViewed )
15:        $RecipeOpenSub    'marker for IWS Xref
16:    End If
17:
18:    'not sure I need to do this, but put here just in case, cause opening recipe should populate hnSegLast
19:    'call global procedure to find last seg, use this to show/hide input fields on RecEditor-Segments screen
20:    Call FindSegLastZx( )
21:    $FindSegLastZxFunc    'marker for IWS Xref tool
22:
23:    'enable OPC comm for screen variables
24:    $OPCCommRecEdSegments = 0
25:
26: End Sub
27:
28:
29: 'This procedure is executed continuously while this screen is open.
30: Sub Screen_WhileOpen()
31:
32:    '//Calculate the "RecipeStartIndex", this value provides a starting array pointer into the long recipe array
    for screen display
33:    $SegmentPagePointer = $ZoneSelectedRadio*20-20    '=0,20,40,60    '$ZoneSelectedRadio= 1=z1, 2=z2, 3=z3,
    4=z4
34:    $RecipeStartIndex = $SegmentPagePointer + $SegmentsDisplayed*10    '=0,10...60,70    '$SegmentsDisplayed=
    (0)=seg0-9 OR (1)=seg10-19
35:
36: End Sub
37:
38:
39: 'This procedure is executed just once when this screen is closed.
```

```
40: Sub Screen_OnClose()  
41:  
42:   'disable OPC comm for screen variables  
43:   $OPCCommRecEdSegments = 1  
44:   'populate the recipe last viewed  
45:   $RecipeLastViewed = $RecipeName  
46:  
47: End Sub  
48:
```

Rec PID/MTL Constants screen

File name: Recipe Editor-Constants.scr

RECIPE EDITOR - PID/MTL CONTROL CONSTANTS DEFINITION FOR ZONES

New

Open

Close

Save

SaveAs

Delete

RECIPE NAME
myRecipe

RECIPE DESCRIPTION
This part is used to make a whachamigger for the whachamacallit.

PART NUMBER(S)
45678, 123456-78

Generate RTF Report File for Recipe: "myRecipe"

Start Recipe **View Segment Grapher** **Est. Cycle Time**
00:33:00

		Zone 1	Zone 2	Zone 3	Zone 4
Control Mode		Fluid Recipe Profile	Mold Recipe Profile	Mold Const Temp	Fluid Const Temp
Const Temp SP		Use Segments	Use Segments	170	170
Fluid Constants:	PID Default1 K_P	12.0	12.0	12.0	12.0
	PID Default2 K_I	1.5	1.5	1.5	1.5
	K_D	0.05	0.05	0.05	0.05
Mold Dynamic Constants:	Dead Time (sec) θ		21	16	
	Time Constant (sec) τ		47	32	
	Closed Loop Time Constant (sec) λ		47	47	
	Entered Gain Default: 0.33 K'		0.3300	0.3300	
	Steady State Gain K_m		0.9960	0.9890	

Run Metal Tuning Test to Establish Metal Constants

This screen is used in concert with the Recipe Segments screen to manage recipe parameters. Explanations for buttons that are common with the Recipe Segments, Recipe Viewer, or Manual Constant Temperature screens will not be duplicated in this section.

Button scripts and properties:

- The 'PID Default1' and 'PID Default2' buttons use an On Up VBScript Command dynamic to call a global procedure and set a tag value; a Position dynamic is also used to show/hide the button.
- The help icon in the upper right hand corner of the screen is a bitmap object with an On Up VBScript Command dynamic that calls the help documentation for this screen.

This screen uses a screen script for additional logic.

```
1: 'Variables available on this screen can be declared and initialized here.
2:
3:
4: 'Procedures available on this screen can be implemented here
5:
6:
7: 'This procedure is executed just once when this screen is open.
8: Sub Screen_OnOpen()
9:
10:    '/REV 3.1.0/ change IF logic
11:    '310    If $RecipeName<>" " And $RecipeIsSaved Then
12:    If $RecipeLastViewed<>" " And $RecipeIsSaved Then
13:        'reopen the last viewed to repopulate the recipe tags, if navigated from a screen that changed them
14:        Call RecipeOpen( $RecipeLastViewed )
15:        $RecipeOpenSub    'marker for IWS Xref
16:    End If
17:
18: End Sub
19:
20: 'This procedure is executed continuously while this screen is open.
21: Sub Screen_WhileOpen()
22:
23: End Sub
24:
25: 'This procedure is executed just once when this screen is closed.
26: Sub Screen_OnClose()
27:
28:    'populate the recipe last viewed
29:    $RecipeLastViewed = $RecipeName
30:
31: End Sub
```

Event Viewer screen

File name: Events.scr

EVENT VIEWER							
Event Time	Type	Message	Tag Name	Value	Previous	User	Comment
02/25/2009 09:55:58	Display	Open Screen:events.scr		0	0	Engineer	
02/25/2009 09:55:58	Display	Close Screen:recipe editor-constants.scr		0	0	Engineer	
02/25/2009 09:55:58	Display	Close Screen:tempclear.scr		0	0	Engineer	
02/25/2009 09:55:58	Display	Open Screen:tempclear.scr		0	0	Engineer	
02/25/2009 09:55:25	Tags Ch...	Z3 Const Temper SP changed	hnSPTemperConstZ[3]	170	50	Engineer	
02/25/2009 09:55:25	Tags Ch...	ControlMode set by Static Tag Integer: 0=CV, 1=PI...	hnControlModeZ[2]	2	1	Engineer	
02/25/2009 09:55:25	Tags Ch...	ControlMode set by Static Tag Integer: 0=CV, 1=PI...	hnControlModeZ[1]	2	1	Engineer	
02/25/2009 09:55:25	Tags Ch...	Z4 Const Temper SP changed	hnSPTemperConstZ[4]	170	50	Engineer	
02/25/2009 09:55:25	Tags Ch...	Z4 Const Temper SP changed	hnSPTemperConstZ[4]	50	170	Engineer	
02/25/2009 09:55:25	Tags Ch...	Z3 Const Temper SP changed	hnSPTemperConstZ[3]	50	170	Engineer	
02/25/2009 09:55:25	Tags Ch...	ControlMode set by Static Tag Integer: 0=CV, 1=PI...	hnControlModeZ[3]	1	2	Engineer	
02/25/2009 09:55:25	Tags Ch...	ControlMode set by Static Tag Integer: 0=CV, 1=PI...	hnControlModeZ[1]	1	2	Engineer	
02/25/2009 09:55:25	Display	Open Screen:recipe editor-constants.scr		0	0	Engineer	
02/25/2009 09:55:24	Display	Close Screen:tempclear.scr		0	0	Engineer	
02/25/2009 09:55:24	Display	Close Screen:recipe editor-segments.scr		0	0	Engineer	
02/25/2009 09:55:24	Display	Open Screen:tempclear.scr		0	0	Engineer	
02/25/2009 09:54:28	Security ...	LogOn		0	0	Engineer	
02/25/2009 09:54:18	Security ...	Invalid attempt to log:		0	0	pag	
02/25/2009 09:54:02	Tags Ch...	Z3 Const Temper SP changed	hnSPTemperConstZ[3]	170	50	pag	
02/25/2009 09:54:02	Tags Ch...	ControlMode set by Static Tag Integer: 0=CV, 1=PI...	hnControlModeZ[2]	2	1	pag	
02/25/2009 09:54:02	Tags Ch...	ControlMode set by Static Tag Integer: 0=CV, 1=PI...	hnControlModeZ[1]	2	1	pag	
02/25/2009 09:54:02	Tags Ch...	Z4 Const Temper SP changed	hnSPTemperConstZ[4]	170	50	pag	
02/25/2009 09:54:02	Tags Ch...	Z4 Const Temper SP changed	hnSPTemperConstZ[4]	50	170	pag	
02/25/2009 09:54:02	Tags Ch...	Z3 Const Temper SP changed	hnSPTemperConstZ[3]	50	170	pag	
02/25/2009 09:54:02	Tags Ch...	Z2 Const Temper SP changed	hnSPTemperConstZ[2]	50	200	pag	
02/25/2009 09:54:02	Tags Ch...	Z1 Const Temper SP changed	hnSPTemperConstZ[1]	50	100	pag	
02/25/2009 09:54:02	Display	Open Screen:recipe editor-segments.scr		0	0	pag	
02/25/2009 09:54:02	Display	Close Screen:tempclear.scr		0	0	pag	
02/25/2009 09:54:02	Display	Open Screen:tempclear.scr		0	0	pag	
02/25/2009 09:54:01	Display	Close Screen:metal tuning.scr		0	0	pag	
02/25/2009 09:53:19	Display	Open Screen:metal tuning.scr		0	0	pag	
02/25/2009 09:53:19	Display	Close Screen:metal tuning.scr		0	0	pag	
02/25/2009 09:50:19	Display	Open Screen:metal tuning.scr		0	0	pag	
02/25/2009 09:50:19	Display	Close Screen:tempclear.scr		0	0	pag	
02/25/2009 09:50:19	Display	Close Screen:non-recipe consttemper.scr		0	0	pag	
02/25/2009 09:50:19	Display	Open Screen:tempclear.scr		0	0	pag	
02/25/2009 09:50:04	Display	Close Screen:splash.scr		0	0	pag	
02/25/2009 09:50:03	Security ...	LogOn		0	0	PAG	
02/25/2009 09:50:01	Display	Open Screen:splash.scr		0	0	Operator	
02/25/2009 09:49:25	Display	Open Screen:non-recipe consttemper.scr		0	0	Operator	
02/25/2009 09:49:25	Display	Close Screen:tempclear.scr		0	0	Operator	
02/25/2009 09:49:25	Display	Open Screen:tempclear.scr		0	0	Operator	
02/25/2009 09:49:23	Display	Close Screen:non-recipe consttemper.scr		0	0	Operator	
02/25/2009 09:49:23	Display	Close Screen:tempclear.scr		0	0	Operator	
02/25/2009 09:49:23	Display	Open Screen:tempclear.scr		0	0	Operator	
02/25/2009 09:49:23	Security ...	LogOn		0	0	Operator	
02/25/2009 09:48:42	Display	Open Screen:non-recipe consttemper.scr		0	0	PAG	
02/25/2009 09:48:41	Display	Close Screen:tempclear.scr		0	0	PAG	
02/25/2009 09:48:41	Display	Close Screen:trendhistorical.scr		0	0	PAG	
02/25/2009 09:48:41	Display	Open Screen:tempclear.scr		0	0	PAG	
02/25/2009 09:48:39	Display	Open Screen:trendhistorical.scr		0	0	PAG	
02/25/2009 09:48:39	Display	Close Screen:trend.scr		0	0	PAG	
02/25/2009 09:48:39	Display	Close Screen:trend.scr		0	0	PAG	

This screen is used to document the actions taken while running. Actions documented are screen changes, selected tag changes, log-on events, and custom message postings.

The event object is an Active Object; therefore has many built in functions.

There are no buttons or scripts to address on this screen.

This screen does not use a screen script for additional logic.

PAG Utilities screen

File name: PAG Utilities.scr

Process Automation Group (PAG) Utilities -- software developed by Josh Salmond, 2008

WinPLC Station: MHCWinPLC

PLC Peak I/O Time (ms)	8
PLC Average Logic Solve Time (ms)	16
PLC Peak Logic Solve Time (ms)	27
PLC Average Scan Interval (ms)	49
PLC Peak Scan Interval (ms)	52
OverScans / ScanCount	0.035 %
PLC Number of OverScans	55
PLC Scan Count	158033

MsgBox to display PLC calendar and clock values

DailyRunScript

Reserved10 Status: (32 bit integer)
b3 b2 b1 b0
0 0 0 0
b3 hi=battery missing or dead or low
b2 hi=Flag forces
b1 hi=Input forces
b0 hi=Output forces

Put editable tHMI_Heartbeat here if have comm timeout problems....

PLC state is 4 zone

MTL Sample Time: 10 [sec]

PID Sample Time: 2 [sec]

Two Zone Machine

Four Zone Machine

Valve Position Z1 [% open]

Valve Position Z2 [% open]

Valve Position Z3 [% open]

Valve Position Z4 [% open]

Valve Output%

100 40 28 76

Slot0 D-In

D2-08ND3

Slot1 D-Out

D2-08TR

Slot2 A-Out

F2-08DA-1

Slot3 Fluid

F2-04THM

Slot4 Mold

F2-04THM

Slot5 Central

F2-04THM

Showing actual PLC value as read by the module.
Actual temperature = (PLC value) / 10 [degF]

CH1+	CH1	CH2+	CH2	CH3+	CH3	CH4+	CH4
Z1: 1181	Z1: 633	Z2: 705	Z2: 630	Z3: 671	Z3: 676	Z4: 681	Z4: 680

mapped to PLC state set, status bit, & 2/4zone switch
J-type TC: '+'=white wire, '-'=red wire

Calculated constants when using Recipe MTL:
was previously on Rec-Const screen

Calculated Gain	K _{CS}	P0	P1	P5	P8	P9
Z1						
Z2						
Z3						
Z4						

These values are only accurate after the current recipe has been downloaded.

Exit MHC to Windows

Computer IP Address: 10.0.0.11
MAC ID: # found: 1 ID: 00-08-9B-B3-52-5D
MAC Adapter: Generic Marvell Yukon Chipset based Gigabit Ethernet Controller
Computer Name: OEMCOMPUTER
c:\ GB free: 67.7
m:\ GB free: 74.3 (MHC Backup)
Operating System: 2 (Z=>Win 2000/XP/Vista)
IWS Product Path: C:\Program Files\InduSoft Web Studio v6.1\
IWS Product Version: 6.000000 ??this should be 6.1! doesn't work at runtime??
IWS Product Version: v6.1+SP4, Build 1042 (ProductPath\BIN\Program Settings.ini)
IWS Application Path:
C:\Documents and Settings\MHCOper\MoldHeatingAndCooling\HMI\IWS_v3.1.0\
IWS HardKey Model: Local Interface - CNF
IWS HardKey S/N: 2737.88

This screen is available if logged in to the system as the PAG user. It provides administrative access to system parameters, PLC values and status tags, global settings such as 2-zone or 4-zone system, and the ability to control the Control Valve directly.

Button scripts and properties:

- The 'MsgBox to display PLC calendar and clock values' button uses an On Up VBScript Command dynamic to call a global procedure.
- The 'DailyRunScript' button has a dynamic caption and uses an On Up VBScript Command dynamic to trigger a Task Script. The button is also grouped with a rectangle object with a Color dynamic to signal when the script is running.
- The 'Two Zone Machine' and 'Four Zone Machine' buttons are radio buttons that use a Color dynamic.
- The 'Valve Position Z_ [% open]' buttons use a Color dynamic and use an On Up VBScript Command dynamic to display a message to the user and change a tag value; the Disable property is also used.
- The Valve Position field is a group containing a rectangle and text object. The rectangle uses an On Up VBScript Command dynamic to launch the keypad for user input. The text object displays the tag value and uses a Position dynamic to show/hide the value.

- The 'Show calculated values' button is a checkbox object; it uses the Disable property and a Color dynamic.
- The 'Exit MHC to Windows' button uses an On Up VBScript Command dynamic to set tags and close the application, rather than logging completely off of Windows.

This screen uses a screen script for additional logic.

```
1: 'Variables available on this screen can be declared and initialized here.
2:
3:
4: 'Procedures available on this screen can be implemented here
5:
6:
7: 'This procedure is executed just once when this screen is open.
8: Sub Screen_OnOpen()
9:
10:    'enable OPC comm for screen variables
11:    $OPCCommPAGUtilities = 0
12:
13: End Sub
14:
15: 'This procedure is executed continuously while this screen is open.
16: Sub Screen_WhileOpen()
17:
18: End Sub
19:
20: 'This procedure is executed just once when this screen is closed.
21: Sub Screen_OnClose()
22:
23:    'disable OPC comm for screen variables
24:    $OPCCommPAGUtilities = 1
25:
26: End Sub
```

Metal Tuning screen

File name: Metal Tuning.scr

METAL ALGORITHM TUNING ROUTINE TO ESTABLISH MOLD CONTROL CONTSTANTS

SETUP:

Fluid Consts:

	Zone 1	Zone 2	Zone 3	Zone 4
K_P	1.5	1.5	1.5	
K_I	4.5	4.5	4.5	
K_D	0.05	0.05	0.05	
Mold Start Temperature	150	175		
Mold End Temperature	350	375		
Fluid Start SP	160	180		
Do not test this zone	☐	☐	☒ Zone Excluded	☒ Zone Excluded
Ctl Mode for zones not tested			PID	CV
Const Temper SP			100	
CV SP %				0

Start Metal Tuning Test for Selected Zones

MONITOR:

	Zone 1	Zone 2	Zone 3	Zone 4
Control Mode	PID	PID	PID	CV
PLC SP Temp	55.0	81.0	100.0	
Valve Output%	0	99	57	50
Mold Temper	TC OK 80.6	TC OK 78.8	TC OK 78.5	TC OK 79.6
Fluid Temper	TC OK 231.7	TC OK 82.2	TC OK 80.9	TC OK 77.3
Stabilize Time	0	0	0	0
Dead Time θ	0	0	0	0
Time Constant τ	0	0	0	0
Status	Not Started	Not Started	n/a	n/a

Abort Testing and Retry

RESULTS:

ID	Theta	Tau	Lambda	K'	Km	Kcs	P0	P1	P5	P8	P9
1	0	0	0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
2	0	0	0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
3	0	0	0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
4	0	0	0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

Save results to Open Recipe File ... ?

Save results to an Existing Recipe File ... ?

Save results to Text File ... ?

Save results to Manual Constant Temper setup?

This screen provides the user with an interface to setup the tuning routine needed to find the mold control dynamic constants. These constants can only be obtained through this tuning procedure and should never be estimated. These constants are needed to run a zone using the Mold Recipe Profile control mode, or the Mold Constant Temperature control mode.

Button scripts and properties:

- The 'Close Screen' button uses an On Up VBScript Command dynamic to close the screen; the button also uses the Disable property.
- The input fields (K_p , K_i , K_d , Mold Start Temperature, Mold End Temperature, Fluid Start SP, Const Temp SP, CV SP%) are a group containing a rectangle and text objects. The rectangle is used to give the field appearance. The text object is Input Enabled, uses the Disable property, and uses a Position dynamic.
- The 'Do not test this zone' button is a checkbox object and uses the Disable property and a Color dynamic.
- The 'Ctl Mode for zones not tested' field is a rectangle with a dynamic caption and uses an On Up VBScript Command dynamic to set the control mode for the zone. The command dynamic also uses the Disable property.
- The 'PID Default1' and 'PID Default2' buttons use an On Up VBScript Command dynamic to call a global procedure and use the Disable property.

- The 'Start Metal Tuning Test for Selected Zones' button uses a Color and Command dynamic with a Disable property. The On Up VBScript Command dynamic displays a message, then if ok initiates the tuning procedure by calling a global procedure and setting a tag.
- The 'Abort Testing and Retry' button is a checkbox object and also uses a Disable property and a Color dynamic.
- The 'Save results to Open Recipe File " _____ " ?' button uses an On Up VBScript Command dynamic to call a global procedure; the button also uses the Disable property.
- The 'Save results to an Existing Recipe File ... ?' button uses an On Up VBScript Command dynamic to set a tag then open a screen; the button also uses the Disable property.
- The 'Save results to Text File ...?' button uses an On Up VBScript Command dynamic to call a global procedure; the button also uses the Disable property.
- The 'Save results to Manual Constant Temp setup?' button uses an On Up VBScript Command dynamic to call a global procedure; the button also uses the Disable property.
- The help icon in the upper right hand corner of the screen is a bitmap object with an On Up VBScript Command dynamic that calls the help documentation for this screen.

This screen uses a screen script for additional logic.

```
1: 'Variables available on this screen can be declared and initialized here.
2:
3:
4: 'Procedures available on this screen can be implemented here
5:
6:
7: 'This procedure is executed just once when this screen is open.
8: Sub Screen_OnOpen()
9:
10:    'enable OPC comm for screen variables
11:    $OPCCommMTLTuning = 0
12:
13: End Sub
14:
15: 'This procedure is executed continuously while this screen is open.
16: Sub Screen_WhileOpen()
17:
18: End Sub
19:
20: 'This procedure is executed just once when this screen is closed.
21: Sub Screen_OnClose()
22:
23:    'disable OPC comm for screen variables
24:    $OPCCommMTLTuning = 1
25:
26: End Sub
```

Segment Grapher screen

File name: Segment Grapher.scr

This screen provides the user with a graphical representation of the temperature versus time layout of the recipe. As of March 2009, a bug exists such that if a time segment is greater than 1.8 hours, the line between the dots of the endpoints of the segment is not visible. InduSoft has been made aware of this bug. All graphs will adjust their time and temperature scales simultaneously when the buttons on the left are used.

Button scripts and properties:

- The 'Close Screen' button uses an On Up VBScript Command dynamic to close the screen.
- The Time-Temperature Max-Min buttons use an On Up VBScript Command dynamic to launch the keypad for user input.
- The Reset button uses an On Up VBScript Command dynamic to reset the scale values to their default settings.

This screen uses a screen script for additional logic.

```
1: 'Variables available on this screen can be declared and initialized here.
2: Dim SetScale
3:
4: 'Procedures available on this screen can be implemented here
5:
6:
7: 'This procedure is executed just once when this screen is open.
8: Sub Screen_OnOpen()
9:
10:     SetScale = True
11:
12: End Sub
13:
14: 'This procedure is executed continuously while this screen is open.
15: Sub Screen_WhileOpen()
16:
17:     '/REV 3.1.0/ for some reason the trend dynamic wouldn't update the scale value unless the screen was open,
    so one shot it here
18:     If SetScale Then
19:         'set time scale to CycleElapsedTimeEst
20:         $TrendSegGrapherScaleMaxX = $CycleElapsedTimeEst
21:         $TrendSegGrapherScaleMinX = 0
22:         SetScale = False
23:     End If
24:
25: End Sub
26:
27: 'This procedure is executed just once when this screen is closed.
28: Sub Screen_OnClose()
29:
30: End Sub
```


Open Recipe Dialog screen

File name: Open Recipe Dialog.scr

Select Recipe to load on screen

Available Recipes	Part Number(s)
a2zone	123456-00, 654321-01, 8765309, 1...
AllFieldsFilled	123456-00, 654321-01, 8765309, 1...
AllMTLnoRamp	123456-00, 654321-01, 8765309, 1...
aRamp	123456-00, 654321-01, 8765309, 1...
aRampF	123456-00, 654321-01, 8765309, 1...
aStepF	123456-00, 654321-01, 8765309, 1...
aStepF1	123456-00, 654321-01, 8765309, 1...
myRecipe	45678, 123456-78
tTSegmentNumbers	123456-00
vb2	
w	
w2zone	1,2,3

Home

Page Up

Up

Down

Page Down

End

This screen lists the recipes available on the system and the respective part numbers that the recipe uses, if that field was configured in the recipe file.

Button scripts and properties:

- The 'Open' button uses an On Up VBScript Command dynamic to call a global procedure.
- The 'Cancel' button uses an On Up VBScript Command dynamic to set a tag and close the screen.
- The 'Home', 'Page Up', 'Up', 'Down', 'Page Down', 'End' buttons use an On Up VBScript Command dynamic to set the selected row tag, thus enable scrolling of the list.

This screen uses a screen script for additional logic.

```
1: 'Variables available on this screen can be declared and initialized here.
2:
3:
4: 'Procedures available on this screen can be implemented here
5:
6:
7: 'This procedure is executed just once when this screen is open.
8: Sub Screen_OnOpen()
9:     'call global procedure
10:     Call PopulateOpenRecipeGridControl()
11:     $PopulateOpenRecipeGridControlSub 'marker for IWS Xref tool
12: End Sub
13:
14: 'This procedure is executed continuously while this screen is open.
15: Sub Screen_WhileOpen()
16:
17: End Sub
18:
19: 'This procedure is executed just once when this screen is closed.
20: Sub Screen_OnClose()
21:
22: End Sub
```

Change Control Mode screen

File name: Control Mode Dialog.scr

Change Control Mode for Zone 1

Zone 1

**Fluid
Constant
Temperature**

Fluid
Recipe
Profile

Mold
Constant
Temperature

Mold
Recipe
Profile

Set Point

50

You have selected to change control modes from RECIPE profile to CONSTANT TEMPERATURE. It appears that you may have already defined some segment information. Changing the control mode from recipe to constant temperature will clear out all previously entered segment information when the OK button on this screen is pressed.

OK

Cancel

This screen enables the user to change the control mode for a zone. The red warning box appears if changing from a previously configured recipe profile to a constant temperature profile. The selected mode is highlighted by the bigger button.

Button scripts and properties:

- The mode buttons are a group containing a button with command dynamic and text. The group has a size dynamic property that sets the size of the button and the text on the button. The On Up VBScript Command dynamic sets tag values that control the button and text sizes.
- The 'OK' button uses the On Up VBScript Command dynamic to execute logic depending on which button has been selected. The logic may call a global procedure, will set the control mode, then close the screen.
- The 'Cancel' button uses an On Up VBScript Command dynamic to close the screen.

This screen uses a screen script for additional logic.

```
1: 'Variables available on this screen can be declared and initialized here.
2:
3:
4: 'Procedures available on this screen can be implemented here
5:
6:
7: 'This procedure is executed just once when this screen is open.
8: Sub Screen_OnOpen()
9:
10:    'Note: $ZoneSelectedRadio must be set manually if coming from "Recipe Editor-Constants.scr"
11:    If $hnControlModeZ[$ZoneSelectedRadio]=$FluidConstantTemper Then $BigButt[1]=True : $BigButt[2]=False :
$BigButt[3]=False : $BigButt[4]=False
12:    If $hnControlModeZ[$ZoneSelectedRadio]=$FluidRecipe Then $BigButt[1]=False : $BigButt[2]=True :
$BigButt[3]=False : $BigButt[4]=False
13:    If $hnControlModeZ[$ZoneSelectedRadio]=$MoldConstantTemper Then $BigButt[1]=False : $BigButt[2]=False :
$BigButt[3]=True : $BigButt[4]=False
14:    If $hnControlModeZ[$ZoneSelectedRadio]=$MoldRecipe Then $BigButt[1]=False : $BigButt[2]=False :
$BigButt[3]=False : $BigButt[4]=True
15:
16: End Sub
17:
18:
19: 'This procedure is executed continuously while this screen is open.
20: Sub Screen_WhileOpen()
21:
22:    Dim Index
23:    'throw up message if following conditions:
24:    ' if REC[seg0].temper<>seg1 OR seg1<>seg2, OR SegLast<>2 then means they have probably setup this zone as
recipe
25:    ' so throw up warning that they are about to change the info
26:    Index = $ZoneSelectedRadio*20-20    '=0,20,40,60 same as $SegmentPagePointer but can't use that when on
Consts screen
27:    If ( $BigButt[1] Or $BigButt[3] ) _
28:        And ( ( $REC[Index+0].TemperSP<>$REC[Index+1].TemperSP ) Or (
$REC[Index+1].TemperSP<>$REC[Index+2].TemperSP ) _
29:            Or ( $hnSegLastZ[$ZoneSelectedRadio] <> 2 ) ) Then
30:        'show warning message that seg info will be lost, use $BigButt since index zero is not used for anything
else
31:        $BigButt[0] = True
32:    Else
33:        $BigButt[0] = False
34:    End If
```

```
35:
36: End Sub
37:
38:
39: 'This procedure is executed just once when this screen is closed.
40: Sub Screen_OnClose()
41:
42:
43: End Sub
```

Input a Segment Time Set-Point screen

File name: Input Dwell Dialog.scr

Input dwell time set point for Segment 0

Input Option 1: hh:mm:ss 0 : 00 : 00

OR

Input Option 2: Total hours 0.0

OR

Input Option 3: Total minutes 0.0

OR

Input Option 4: Total seconds 0.0

Reset ALL fields to ZERO

Segment 0 0 (seconds)

Setpoint: 00 : 00 : 00 (hh:mm:ss)

Accept Set Point Cancel

This screen provides the user with the interface to intelligently set the segment time. Since the segment time is stored and displayed in seconds, this interface allows the flexibility for several input methods.

Button scripts and properties:

- The 'hh:mm:ss' fields are rectangle objects with dynamic text and uses an On Up VBScript Command dynamic to reset fields via a global procedure, then launch the keypad, and if the keypad entry is successful, will convert the entry into seconds via a global procedure.
- The total fields are a group containing a rectangle and text objects. The text object displays the tag value and uses an On Up VBScript Command dynamic to reset fields via a global procedure, then launch the keypad, and if the keypad entry is successful, will convert the entry into seconds via a global procedure.
- The 'Reset ALL fields to ZERO' button uses an On Up VBScript Command dynamic to call a global procedure.
- The 'Accept Set Point' button uses an On Up VBScript Command dynamic to close the screen, set the tag value, and call a task script.
- The 'Cancel' button uses an On Up VBScript Command dynamic to close the screen.

This screen uses a screen script for additional logic.

```
1: 'Variables available on this screen can be declared and initialized here.
2:
3:
4: 'Procedures available on this screen can be implemented here
5:
6:
7: 'This procedure is executed just once when this screen is open.
8: Sub Screen_OnOpen()
9:     'global procedure sub: set user dwell time inputs to zero
10:     Call resetALLFields2Zero()
11:     $resetALLFields2ZeroSub 'marker for IWS Xref tool
12: End Sub
13:
14: 'This procedure is executed continuously while this screen is open.
15: Sub Screen_WhileOpen()
16:
17: End Sub
18:
19: 'This procedure is executed just once when this screen is closed.
20: Sub Screen_OnClose()
21:
22: End Sub
```

Segment Operations screen

File name: Segment Operations Popup.scr

This screen is called when the Insert or Delete buttons are pressed on the Recipe Segments screen. It provides some narrative and indication for the segment that is to be changed.

Button scripts and properties:

- The 'OK' button uses an On Up VBScript Command dynamic to call a Task Script and to close the screen. The button uses a Position dynamic to show/hide the button.
- The 'Cancel' button uses an On Up VBScript Command dynamic to set tags and close the screen.

This screen does not use a screen script for additional logic.

Input Serial Number screen

File name: Input SN Dialog.scr

Input Serial Number

Input Serial Number

Input Result

Close

This screen is called if the preheat mode is set to Manual Preheat, and when the Start Recipe button is pushed.

Button scripts and properties:

- The 'Input Serial Number' field is a rectangle object and uses an On Up VBScript Command dynamic to call a global procedure; the command dynamic also uses the Disable property.
- The 'Close' button uses an On Up VBScript Command dynamic to display a message. If the message is canceled, the screen is closed. The command dynamic also uses the Disable property.

This screen uses a screen script for additional logic.

```
1: 'Variables available on this screen can be declared and initialized here.
2: '/REV 3.1.0/ created screen
3:
4: 'Procedures available on this screen can be implemented here
5:
6: Sub GetSN()
7:     $GetSNSub      'marker for IWS Xref
8:
9:     'hide input button and close button on screen
10:    $disableSNScreenButtons = 1
11:
12:    'launch keypad for operator input of new name, and check, but first clear it out, so they don't use the
    same one all the time
13:    $hstSerialNumber = ""
14:    $KeypadReturn = $Keypad( "$hstSerialNumber", "Alphanumeric", 0, "Input Serial Number" )
15:    If $KeypadReturn = 0 And $hstSerialNumber<>"" Then      'return of 0 is success
16:        'check name for invalid characters, call function on global
17:        If CheckInputName( $hstSerialNumber ) = 0 Then      'function=0 is ok
18:            $CheckInputNameFunc      'marker for IWS Xref tool
19:            'close screen and proceed with downloading the recipe
20:            $Close( "Input SN Dialog.scr" )
21:            'trigger Task script which calls downloading sub;
22:            ' running on Task Script seems to work better than calling $RunGlobalProcedureOnServer
23:            $LoadStartRecipe = 1
24:        Else
25:            'don't download, force user to input a good SN
26:            $hstSerialNumber = "error"
27:        End If
28:    Else '$KeypadReturn<>0 or SN=""
29:        'don't download, force user to input a SN
30:        $hstSerialNumber = "empty"
31:    End If
32:
33:    'show buttons on screen
34:    $disableSNScreenButtons = 0
35:
36: End Sub
37:
38:
39: 'This procedure is executed just once when this screen is open.
40: Sub Screen_OnOpen()
41:
```

```
42:      'clear out tag so nothing shows up in display field from previous entries
43:      $hstSerialNumber = " "
44:
45: End Sub
46:
47: 'This procedure is executed continuously while this screen is open.
48: Sub Screen_WhileOpen()
49:
50: End Sub
51:
52: 'This procedure is executed just once when this screen is closed.
53: Sub Screen_OnClose()
54:
55: End Sub
```

Wait screen

File name: Wait Dialog.scr

This screen is called when the Start Recipe button is pressed.

Button scripts and properties:

- The 'Cancel Download to Controller' button uses an On Up VBScript Command dynamic to set a tag and close the screen. A Position dynamic is also used to show/hide the button.

This screen uses a screen script for additional logic.

```
1: 'Variables available on this screen can be declared and initialized here.
2:
3:
4: 'Procedures available on this screen can be implemented here
5:
6:
7: 'This procedure is executed just once when this screen is open.
8: Sub Screen_OnOpen()
9:     'reset tag
10:     $WaitProgress = 0
11: End Sub
12:
13:
14: 'This procedure is executed continuously while this screen is open.
15: Sub Screen_WhileOpen()
16:
17:
18: End Sub
19:
20:
21: 'This procedure is executed just once when this screen is closed.
22: Sub Screen_OnClose()
23:     'reset tag
24:     $WaitProgress = 0
25: End Sub
```

Alarm/Warning/Error screen

File name: AlarmWarning Popup.scr

This screen is used to display the nature of an alarm, a warning, or an error.

Button scripts and properties:

- The 'OK' button uses an On Up VBScript Command dynamic to set a tag and close the screen.
- The 'Exit MHC' button (not shown) is used as a failsafe escape to logoff the user. This is the same button that is on the Status screen.

This screen does not use a screen script for additional logic.

Workspace Toolbar: Comm Tab>OPC

OPC Sheet 1

OPC OPCCL001.OPC

Description: Screen: Status Server Identifier: ThinknDo.TNDCEOPC Disable: OPCCommStatus

Read Update Rate (ms): 500 Percent Deadband: Status:

Remote Server Name: Browse... ☐ Read after writing
☒ Accept Tag Name in the Item column

	Tag Name	Item	Scan	Div	Add
*			Always		
*			Always		
*			Always		
*			Always		
*			Always		

OPC Sheet 2

OPC OPCCL002.OPC

Description: Segment and Control xfer Server Identifier: ThinknDo.TNDCEOPC Disable: OPCCommRecipeXfer

Read Update Rate (ms): 100 Percent Deadband: Status:

Remote Server Name: Browse... ☐ Read after writing
☒ Accept Tag Name in the Item column

	Tag Name	Item	Scan	Div	Add
1		//use PLCRollingMsecCounter to check OPC comm for the sheet	Always		
2	PLCRollingMsecCounter	{WinPLCStationName}.System.RollingMsecCounter	Always		
3			Always		
4	hfStartRecipe	{WinPLCStationName}.Flag.hfStartRecipe	Always		
5	hn2_4zoneMachineSwitch	{WinPLCStationName}.Number.hn2_4zoneMachineSwitch	Always		
6		//{WinPLCStationName}.String.hwRecipeName	Always		
7	hfManualPreheat	{WinPLCStationName}.Flag.hfManualPreheat	Always		
8	hfRXHandshake	{WinPLCStationName}.Flag.hfRXHandshake	Always		
9			Always		
10	fRXSegDataStored	{WinPLCStationName}.Flag.fRXSegDataStored	Always		
11	cRXSegNumber	{WinPLCStationName}.Counter.cRXSegNumber	Always		
12	hnRXData_DwellTimeSP_0	{WinPLCStationName}.Number.hnRXData_DwellTimeSP_0	Always		
13	hnRXData_TemperSP_0	{WinPLCStationName}.Number.hnRXData_TemperSP_0	Always		
14	hnRXData_DwellTimeSP_1	{WinPLCStationName}.Number.hnRXData_DwellTimeSP_1	Always		
15	hnRXData_TemperSP_1	{WinPLCStationName}.Number.hnRXData_TemperSP_1	Always		
16	hnRXData_DwellTimeSP_2	{WinPLCStationName}.Number.hnRXData_DwellTimeSP_2	Always		
17	hnRXData_TemperSP_2	{WinPLCStationName}.Number.hnRXData_TemperSP_2	Always		
18	hnRXData_DwellTimeSP_3	{WinPLCStationName}.Number.hnRXData_DwellTimeSP_3	Always		
19	hnRXData_TemperSP_3	{WinPLCStationName}.Number.hnRXData_TemperSP_3	Always		
20	hnRXData_DwellTimeSP_4	{WinPLCStationName}.Number.hnRXData_DwellTimeSP_4	Always		
21	hnRXData_TemperSP_4	{WinPLCStationName}.Number.hnRXData_TemperSP_4	Always		
22	hnRXData_DwellTimeSP_5	{WinPLCStationName}.Number.hnRXData_DwellTimeSP_5	Always		
23	hnRXData_TemperSP_5	{WinPLCStationName}.Number.hnRXData_TemperSP_5	Always		
24	hnRXData_DwellTimeSP_6	{WinPLCStationName}.Number.hnRXData_DwellTimeSP_6	Always		
25	hnRXData_TemperSP_6	{WinPLCStationName}.Number.hnRXData_TemperSP_6	Always		
26	hnRXData_DwellTimeSP_7	{WinPLCStationName}.Number.hnRXData_DwellTimeSP_7	Always		
27	hnRXData_TemperSP_7	{WinPLCStationName}.Number.hnRXData_TemperSP_7	Always		
28	hnRXData_DwellTimeSP_8	{WinPLCStationName}.Number.hnRXData_DwellTimeSP_8	Always		
29	hnRXData_TemperSP_8	{WinPLCStationName}.Number.hnRXData_TemperSP_8	Always		
30	hnRXData_DwellTimeSP_9	{WinPLCStationName}.Number.hnRXData_DwellTimeSP_9	Always		

OPC OPCCL002.OPC

Description: Segment and Control xfer Server Identifier: ThinknDo.TNDCEOPC Disable: OPCCommRecipeXfer

Read Update Rate (ms): 100 Percent Deadband: Status:

Remote Server Name: Browse... ☐ Read after writing ☒ Accept Tag Name in the Item column

	Tag Name	Item	Scan	Div	Add
31	hnRXData_TemperSP_9	{WinPLCStationName}.Number.hnRXData_TemperSP_9	Always		
32			Always		
33	hnRXData_RampYN	{WinPLCStationName}.Number.hnRXData_RampYN	Always		
34			Always		
35	fRXConDataStored	{WinPLCStationName}.Flag.fRXConDataStored	Always		
36	cRXZoneNumber	{WinPLCStationName}.Counter.cRXZoneNumber	Always		
37	hrRXData_ConstsDefined	{WinPLCStationName}.Flag.hrRXData_ConstsDefined	Always		
38	hrRXData_MoldTemperLimits	{WinPLCStationName}.Flag.hrRXData_MoldTemperLimits	Always		
39	hnRXData_MoldTemperLimitUpper	{WinPLCStationName}.Number.hnRXData_MoldTemperLimitUpper	Always		
40	hnRXData_MoldTemperLimitLower	{WinPLCStationName}.Number.hnRXData_MoldTemperLimitLower	Always		
41	hnRXData_SPTemperCycleAbort	{WinPLCStationName}.Number.hnRXData_SPTemperCycleAbort	Always		
42	hnRXData_SPTemperConst	{WinPLCStationName}.Number.hnRXData_SPTemperConst	Always		
43	hnRXData_SegLast	{WinPLCStationName}.Number.hnRXData_SegLast	Always		
44	hnRXData_CtlMode	{WinPLCStationName}.Number.hnRXData_CtlMode	Always		
45	hnRXData_Theta	{WinPLCStationName}.Number.hnRXData_Theta	Always		
46	hnRXData_Tau	{WinPLCStationName}.Number.hnRXData_Tau	Always		
47	hnRXData_Lambda	{WinPLCStationName}.Number.hnRXData_Lambda	Always		
48	hrRXData_Kprime	{WinPLCStationName}.Float.hrRXData_Kprime	Always		
49	hrRXData_Km	{WinPLCStationName}.Float.hrRXData_Km	Always		
50	hrRXData_Kp	{WinPLCStationName}.Float.hrRXData_Kp	Always		
51	hrRXData_Ki	{WinPLCStationName}.Float.hrRXData_Ki	Always		
52	hrRXData_Kd	{WinPLCStationName}.Float.hrRXData_Kd	Always		
*			Always		

OPC Sheet 3

OPC OPCL003.OPC

Description: Recipe Control Parameters Server Identifier: ThinknDo.TNDCEOPC Disable: OPCCommNonRecConstTei

Read Update Rate (ms): 2000 Percent Deadband: Status:

Remote Server Name: Browse... ☐ Read after writing ☒ Accept Tag Name in the Item column

	Tag Name	Item	Scan	Div	Add
1	PLCRollingMsecCounter	{WinPLCStationName}.System.RollingMsecCounter	Always		
2			Always		
3	hnControlModeZ[1]	{WinPLCStationName}.Number.hnControlModeZ1	Always		
4	hnSPTemperConstZ[1]	{WinPLCStationName}.Number.hnSPTemperConstZ1	Always		
5	NonRecipeConstTemper_PID[1].Kp	{WinPLCStationName}.Float.hrPID_Kp_Z1	Always		
6	NonRecipeConstTemper_PID[1].Ki	{WinPLCStationName}.Float.hrPID_Ki_Z1	Always		
7	NonRecipeConstTemper_PID[1].Kd	{WinPLCStationName}.Float.hrPID_Kd_Z1	Always		
8	NonRecipeConstTemper_MTL[1].Theta	{WinPLCStationName}.Number.hnMTL_Theta_Z1	Always		
9	NonRecipeConstTemper_MTL[1].Tau	{WinPLCStationName}.Number.hnMTL_Tau_Z1	Always		
10	NonRecipeConstTemper_MTL[1].Lambda	{WinPLCStationName}.Number.hnMTL_Lambda_Z1	Always		
11	NonRecipeConstTemper_MTL[1].Kprime	{WinPLCStationName}.Float.hrMTL_Kprime_Z1	Always		
12	NonRecipeConstTemper_MTL[1].Km	{WinPLCStationName}.Float.hrMTL_Km_Z1	Always		
13	NonRecipeConstTemper_MTL[1].Kcs	{WinPLCStationName}.Float.rMTL_Kcs_Z1	Always		
14	NonRecipeConstTemper_MTL[1].P0	{WinPLCStationName}.Float.rMTL_P0_Z1	Always		
15	NonRecipeConstTemper_MTL[1].P1	{WinPLCStationName}.Float.rMTL_P1_Z1	Always		
16	NonRecipeConstTemper_MTL[1].P5	{WinPLCStationName}.Float.rMTL_P5_Z1	Always		
17	NonRecipeConstTemper_MTL[1].P8	{WinPLCStationName}.Float.rMTL_P8_Z1	Always		
18	NonRecipeConstTemper_MTL[1].P9	{WinPLCStationName}.Float.rMTL_P9_Z1	Always		
19	NonRecipeConstTemper_MTL[1].ConstsDefined	{WinPLCStationName}.Flag.hrMTLConstsDefinedZ1	Always		
20			Always		
21	hnControlModeZ[2]	{WinPLCStationName}.Number.hnControlModeZ2	Always		
22	hnSPTemperConstZ[2]	{WinPLCStationName}.Number.hnSPTemperConstZ2	Always		
23	NonRecipeConstTemper_PID[2].Kp	{WinPLCStationName}.Float.hrPID_Kp_Z2	Always		
24	NonRecipeConstTemper_PID[2].Ki	{WinPLCStationName}.Float.hrPID_Ki_Z2	Always		
25	NonRecipeConstTemper_PID[2].Kd	{WinPLCStationName}.Float.hrPID_Kd_Z2	Always		
26	NonRecipeConstTemper_MTL[2].Theta	{WinPLCStationName}.Number.hnMTL_Theta_Z2	Always		
27	NonRecipeConstTemper_MTL[2].Tau	{WinPLCStationName}.Number.hnMTL_Tau_Z2	Always		
28	NonRecipeConstTemper_MTL[2].Lambda	{WinPLCStationName}.Number.hnMTL_Lambda_Z2	Always		
29	NonRecipeConstTemper_MTL[2].Kprime	{WinPLCStationName}.Float.hrMTL_Kprime_Z2	Always		
30	NonRecipeConstTemper_MTL[2].Km	{WinPLCStationName}.Float.hrMTL_Km_Z2	Always		

OPC OPCCL003.OPC

Description: Recipe Control Parameters Server Identifier: ThinkrDo.TNDCEOPC Disable: OPCCommNonRecConstTer

Read Update Rate (ms): 2000 Percent Deadband: Status:

Remote Server Name: Browse... ☐ Read after writing ☒ Accept Tag Name in the Item column

	Tag Name	Item	Scan	Div	Add
31	NonRecipeConstTemper_MTL[2].Kcs	{WinPLCStationName}.Float.rMTL_Kcs_Z2	Always		
32	NonRecipeConstTemper_MTL[2].P0	{WinPLCStationName}.Float.rMTL_P0_Z2	Always		
33	NonRecipeConstTemper_MTL[2].P1	{WinPLCStationName}.Float.rMTL_P1_Z2	Always		
34	NonRecipeConstTemper_MTL[2].P5	{WinPLCStationName}.Float.rMTL_P5_Z2	Always		
35	NonRecipeConstTemper_MTL[2].P8	{WinPLCStationName}.Float.rMTL_P8_Z2	Always		
36	NonRecipeConstTemper_MTL[2].P9	{WinPLCStationName}.Float.rMTL_P9_Z2	Always		
37	NonRecipeConstTemper_MTL[2].ConstsDefined	{WinPLCStationName}.Flag.hrMTLConstsDefinedZ2	Always		
38			Always		
39	hnControlModeZ[3]	{WinPLCStationName}.Number.hnControlModeZ3	Always		
40	hnSPTemperConstZ[3]	{WinPLCStationName}.Number.hnSPTemperConstZ3	Always		
41	NonRecipeConstTemper_PID[3].Kp	{WinPLCStationName}.Float.hrPID_Kp_Z3	Always		
42	NonRecipeConstTemper_PID[3].Ki	{WinPLCStationName}.Float.hrPID_Ki_Z3	Always		
43	NonRecipeConstTemper_PID[3].Kd	{WinPLCStationName}.Float.hrPID_Kd_Z3	Always		
44	NonRecipeConstTemper_MTL[3].Theta	{WinPLCStationName}.Number.hnMTL_Theta_Z3	Always		
45	NonRecipeConstTemper_MTL[3].Tau	{WinPLCStationName}.Number.hnMTL_Tau_Z3	Always		
46	NonRecipeConstTemper_MTL[3].Lambda	{WinPLCStationName}.Number.hnMTL_Lambda_Z3	Always		
47	NonRecipeConstTemper_MTL[3].Kprime	{WinPLCStationName}.Float.hrMTL_Kprime_Z3	Always		
48	NonRecipeConstTemper_MTL[3].Km	{WinPLCStationName}.Float.hrMTL_Km_Z3	Always		
49	NonRecipeConstTemper_MTL[3].Kcs	{WinPLCStationName}.Float.rMTL_Kcs_Z3	Always		
50	NonRecipeConstTemper_MTL[3].P0	{WinPLCStationName}.Float.rMTL_P0_Z3	Always		
51	NonRecipeConstTemper_MTL[3].P1	{WinPLCStationName}.Float.rMTL_P1_Z3	Always		
52	NonRecipeConstTemper_MTL[3].P5	{WinPLCStationName}.Float.rMTL_P5_Z3	Always		
53	NonRecipeConstTemper_MTL[3].P8	{WinPLCStationName}.Float.rMTL_P8_Z3	Always		
54	NonRecipeConstTemper_MTL[3].P9	{WinPLCStationName}.Float.rMTL_P9_Z3	Always		
55	NonRecipeConstTemper_MTL[3].ConstsDefined	{WinPLCStationName}.Flag.hrMTLConstsDefinedZ3	Always		
56			Always		
57	hnControlModeZ[4]	{WinPLCStationName}.Number.hnControlModeZ4	Always		
58	hnSPTemperConstZ[4]	{WinPLCStationName}.Number.hnSPTemperConstZ4	Always		
59	NonRecipeConstTemper_PID[4].Kp	{WinPLCStationName}.Float.hrPID_Kp_Z4	Always		
60	NonRecipeConstTemper_PID[4].Ki	{WinPLCStationName}.Float.hrPID_Ki_Z4	Always		

OPC OPCCL003.OPC

Description: Server Identifier: Disable:

Read Update Rate (ms): Percent Deadband: Status:

Remote Server Name: ☐ Read after writing
☒ Accept Tag Name in the Item column

	Tag Name	Item	Scan	Div	Add
61	NonRecipeConstTemper_PID[4].Kd	{WinPLCStationName}.Float.hrPID_Kd_Z4	Always		
62	NonRecipeConstTemper_MTL[4].Theta	{WinPLCStationName}.Number.hnMTL_Theta_Z4	Always		
63	NonRecipeConstTemper_MTL[4].Tau	{WinPLCStationName}.Number.hnMTL_Tau_Z4	Always		
64	NonRecipeConstTemper_MTL[4].Lambda	{WinPLCStationName}.Number.hnMTL_Lambda_Z4	Always		
65	NonRecipeConstTemper_MTL[4].Kprime	{WinPLCStationName}.Float.hrMTL_Kprime_Z4	Always		
66	NonRecipeConstTemper_MTL[4].Km	{WinPLCStationName}.Float.hrMTL_Km_Z4	Always		
67	NonRecipeConstTemper_MTL[4].Kcs	{WinPLCStationName}.Float.rMTL_Kcs_Z4	Always		
68	NonRecipeConstTemper_MTL[4].P0	{WinPLCStationName}.Float.rMTL_P0_Z4	Always		
69	NonRecipeConstTemper_MTL[4].P1	{WinPLCStationName}.Float.rMTL_P1_Z4	Always		
70	NonRecipeConstTemper_MTL[4].P5	{WinPLCStationName}.Float.rMTL_P5_Z4	Always		
71	NonRecipeConstTemper_MTL[4].P8	{WinPLCStationName}.Float.rMTL_P8_Z4	Always		
72	NonRecipeConstTemper_MTL[4].P9	{WinPLCStationName}.Float.rMTL_P9_Z4	Always		
73	NonRecipeConstTemper_MTL[4].ConstsDefined	{WinPLCStationName}.Flag.hrMTLConstsDefinedZ4	Always		
*			Always		

OPC Sheet 4

OPC OPCCL004.OPC

Description: Server Identifier: Disable:

Read Update Rate (ms): Percent Deadband: Status:

Remote Server Name: ☐ Read after writing
☒ Accept Tag Name in the Item column

	Tag Name	Item	Scan	Div	Add
1	hnControlModeZ[1]	{WinPLCStationName}.Number.hnControlModeZ1	Always		
2	hnControlModeZ[2]	{WinPLCStationName}.Number.hnControlModeZ2	Always		
3	hnControlModeZ[3]	{WinPLCStationName}.Number.hnControlModeZ3	Always		
4	hnControlModeZ[4]	{WinPLCStationName}.Number.hnControlModeZ4	Always		
5	cSegNumberZ1	{WinPLCStationName}.Counter.cSegNumberZ1	Always		
6	cSegNumberZ2	{WinPLCStationName}.Counter.cSegNumberZ2	Always		
7	cSegNumberZ3	{WinPLCStationName}.Counter.cSegNumberZ3	Always		
8	cSegNumberZ4	{WinPLCStationName}.Counter.cSegNumberZ4	Always		
*			Always		

OPC Sheet 5

OPCCL005.OPC

Description: : T-couple Central System Server Identifier: ThinknDo.TNDCEOPC Disable: 0

Read Update Rate (ms): 2000 Percent Deadband: Status: 1

Remote Server Name: ☐ Read after writing
☒ Accept Tag Name in the Item column

	Tag Name	Item	Scan	Div	Add
1		//had to move Mold Fluid TCs to each screen cause ...	Screen		
2			Screen		
3	nTC_CentralHotSupply	{WinPLCStationName}.Number.nTC_CentralHotSup...	Screen		
4	nTC_CentralHotReturn	{WinPLCStationName}.Number.nTC_CentralHotRet...	Screen		
5	nTC_CentralColdSupply	{WinPLCStationName}.Number.nTC_CentralColdSu...	Screen		
6	nTC_CentralColdReturn	{WinPLCStationName}.Number.nTC_CentralColdRe...	Screen		
*			Always		

OPC Sheet 6

OPCCL006.OPC

Description: Screen: PAG Utilities Server Identifier: ThinknDo.TNDCEOPC Disable: OPCCommPAGUtilities

Read Update Rate (ms): 2000 Percent Deadband: Status: 1

Remote Server Name: ☐ Read after writing
☒ Accept Tag Name in the Item column

	Tag Name	Item	Scan	Div	Add
1	hn2_4zoneMachineSwitch	{WinPLCStationName}.Number.hn2_4zoneMachine...	Always		
2	hnSPControlValveManualZ[1]	{WinPLCStationName}.Number.hnSPControlValveM...	Always		
3	hnSPControlValveManualZ[2]	{WinPLCStationName}.Number.hnSPControlValveM...	Always		
4	hnSPControlValveManualZ[3]	{WinPLCStationName}.Number.hnSPControlValveM...	Always		
5	hnSPControlValveManualZ[4]	{WinPLCStationName}.Number.hnSPControlValveM...	Always		
6	hnControlModeZ[1]	{WinPLCStationName}.Number.hnControlModeZ1	Always		
7	hnControlModeZ[2]	{WinPLCStationName}.Number.hnControlModeZ2	Always		
8	hnControlModeZ[3]	{WinPLCStationName}.Number.hnControlModeZ3	Always		
9	hnControlModeZ[4]	{WinPLCStationName}.Number.hnControlModeZ4	Always		
10	PLCAvgLogicTime	{WinPLCStationName}.System.AvgLogicTime	Always		
11	PLCPeakLogicTime	{WinPLCStationName}.System.PeakLogicTime	Always		
12	PLCAvgScanInterval	{WinPLCStationName}.System.AvgScanInterval	Always		
13	PLCPeakScanInterval	{WinPLCStationName}.System.PeakScanInterval	Always		
14	PLCOverScans	{WinPLCStationName}.System.OverScans	Always		
15	PLCPeakIOTime	{WinPLCStationName}.System.PeakIOTime	Always		
16	PLCReserved10	{WinPLCStationName}.System.Reserved10	Always		
17	PLCReserved7	{WinPLCStationName}.System.Reserved7	Always		
18	PLCReserved8	{WinPLCStationName}.System.Reserved8	Always		
*			Always		

OPC Sheet 7

OPCCL007.OPC

Description: Screen: Diagnostic Server Identifier: ThinknDo.TNDCEOPC Disable: OPCCommDiagnostic

Read Update Rate (ms): 500 Percent Deadband: Status:

Remote Server Name: Browse... ☐ Read after writing ☒ Accept Tag Name in the Item column

	Tag Name	Item	Scan	Div	Add
1	hnControlModeZ[1]	{WinPLCStationName}.Number.hnControlModeZ1	Always		
2	hnControlModeZ[2]	{WinPLCStationName}.Number.hnControlModeZ2	Always		
3	hnControlModeZ[3]	{WinPLCStationName}.Number.hnControlModeZ3	Always		
4	hnControlModeZ[4]	{WinPLCStationName}.Number.hnControlModeZ4	Always		
5			Always		
6	cSegNumberZ1	{WinPLCStationName}.Counter.cSegNumberZ1	Always		
7	cSegNumberZ2	{WinPLCStationName}.Counter.cSegNumberZ2	Always		
8	cSegNumberZ3	{WinPLCStationName}.Counter.cSegNumberZ3	Always		
9	cSegNumberZ4	{WinPLCStationName}.Counter.cSegNumberZ4	Always		
10	nSegElapsedTimeZ1	{WinPLCStationName}.Number.nSegElapsedTimeZ1	Always		
11	nSegElapsedTimeZ2	{WinPLCStationName}.Number.nSegElapsedTimeZ2	Always		
12	nSegElapsedTimeZ3	{WinPLCStationName}.Number.nSegElapsedTimeZ3	Always		
13	nSegElapsedTimeZ4	{WinPLCStationName}.Number.nSegElapsedTimeZ4	Always		
14	nCycleElapsedTime	{WinPLCStationName}.Number.nCycleElapsedTime	Always		
15			Always		
16		// rSP_MTL_Z[1] on Always Scan, Every 2 seconds	Always		
17		// rSP_PID_Z[1] on Always Scan, Every 2 seconds	Always		
18	hnSPTemperConstZ[1]	{WinPLCStationName}.Number.hnSPTemperConstZ1	Always		
19	hnSPTemperConstZ[2]	{WinPLCStationName}.Number.hnSPTemperConstZ2	Always		
20	hnSPTemperConstZ[3]	{WinPLCStationName}.Number.hnSPTemperConstZ3	Always		
21	hnSPTemperConstZ[4]	{WinPLCStationName}.Number.hnSPTemperConstZ4	Always		
22			Always		
23	sMoldTemperLimitsBreachZ[1]	{WinPLCStationName}.Flag.sMoldTemperLimitsBreachZ1	Always		
24	sMoldTemperLimitsBreachZ[2]	{WinPLCStationName}.Flag.sMoldTemperLimitsBreachZ2	Always		
25	sMoldTemperLimitsBreachZ[3]	{WinPLCStationName}.Flag.sMoldTemperLimitsBreachZ3	Always		
26	sMoldTemperLimitsBreachZ[4]	{WinPLCStationName}.Flag.sMoldTemperLimitsBreachZ4	Always		
27			Always		

OPCCL007.OPC

Description: Screen: Diagnostic Server Identifier: ThinknDo.TNDCEOPC Disable: OPCCommDiagnostic

Read Update Rate (ms): 500 Percent Deadband: Status:

Remote Server Name: Browse... ☐ Read after writing ☒ Accept Tag Name in the Item column

	Tag Name	Item	Scan	Div	Add
28	rMTL_ISE_Z1	{WinPLCStationName}.Float.rMTL_ISE_Z1	Always		
29	rMTL_ISE_Z2	{WinPLCStationName}.Float.rMTL_ISE_Z2	Always		
30	rMTL_ISE_Z3	{WinPLCStationName}.Float.rMTL_ISE_Z3	Always		
31	rMTL_ISE_Z4	{WinPLCStationName}.Float.rMTL_ISE_Z4	Always		
32	rPID_ISE_Z1	{WinPLCStationName}.Float.rPID_ISE_Z1	Always		
33	rPID_ISE_Z2	{WinPLCStationName}.Float.rPID_ISE_Z2	Always		
34	rPID_ISE_Z3	{WinPLCStationName}.Float.rPID_ISE_Z3	Always		
35	rPID_ISE_Z4	{WinPLCStationName}.Float.rPID_ISE_Z4	Always		
36			Always		
37	hfManualPreheat	{WinPLCStationName}.Flag.hfManualPreheat	Always		
38	hfSkipPreheat	{WinPLCStationName}.Flag.hfSkipPreheat	Always		
*			Always		

OPC Sheet 8

OPCCL008.OPC

Description: : Recipe Editor - Segments Server Identifier: ThinknDo.TNDCEOPC Disable: OPCCommRecEdSegments

Read Update Rate (ms): 2000 Percent Deadband: Status: |

Remote Server Name: Browse... ☐ Read after writing
☒ Accept Tag Name in the Item column

	Tag Name	Item	Scan	Div	Add
1		//probably don't want any other tags for this screen enabled	Always		
2	hfSkipPreheat	{WinPLCStationName}.Flag.hfSkipPreheat	Always		
*			Always		

OPC Sheet 9

OPCCL009.OPC

Description: Ion: Recipe MTL Constants Server Identifier: ThinknDo.TNDCEOPC Disable: OPCCommRecEdConstants

Read Update Rate (ms): 2000 Percent Deadband: Status: |

Remote Server Name: Browse... ☐ Read after writing
☒ Accept Tag Name in the Item column

	Tag Name	Item	Scan	Div	Add
1		//these are read only; enabled by button on screen	Always		
2		//REV 3.1.0/ moved to PAG Utilities screen from MTL Consts screen	Always		
3	REC_MTL[1].Kcs	{WinPLCStationName}.Float.rMTL_Kcs_Z1	Always		
4	REC_MTL[1].P0	{WinPLCStationName}.Float.rMTL_P0_Z1	Always		
5	REC_MTL[1].P1	{WinPLCStationName}.Float.rMTL_P1_Z1	Always		
6	REC_MTL[1].P5	{WinPLCStationName}.Float.rMTL_P5_Z1	Always		
7	REC_MTL[1].P8	{WinPLCStationName}.Float.rMTL_P8_Z1	Always		
8	REC_MTL[1].P9	{WinPLCStationName}.Float.rMTL_P9_Z1	Always		
9			Always		
10	REC_MTL[2].Kcs	{WinPLCStationName}.Float.rMTL_Kcs_Z2	Always		
11	REC_MTL[2].P0	{WinPLCStationName}.Float.rMTL_P0_Z2	Always		
12	REC_MTL[2].P1	{WinPLCStationName}.Float.rMTL_P1_Z2	Always		
13	REC_MTL[2].P5	{WinPLCStationName}.Float.rMTL_P5_Z2	Always		
14	REC_MTL[2].P8	{WinPLCStationName}.Float.rMTL_P8_Z2	Always		
15	REC_MTL[2].P9	{WinPLCStationName}.Float.rMTL_P9_Z2	Always		
16			Always		
17	REC_MTL[3].Kcs	{WinPLCStationName}.Float.rMTL_Kcs_Z3	Always		
18	REC_MTL[3].P0	{WinPLCStationName}.Float.rMTL_P0_Z3	Always		
19	REC_MTL[3].P1	{WinPLCStationName}.Float.rMTL_P1_Z3	Always		
20	REC_MTL[3].P5	{WinPLCStationName}.Float.rMTL_P5_Z3	Always		
21	REC_MTL[3].P8	{WinPLCStationName}.Float.rMTL_P8_Z3	Always		
22	REC_MTL[3].P9	{WinPLCStationName}.Float.rMTL_P9_Z3	Always		
23			Always		
24	REC_MTL[4].Kcs	{WinPLCStationName}.Float.rMTL_Kcs_Z4	Always		
25	REC_MTL[4].P0	{WinPLCStationName}.Float.rMTL_P0_Z4	Always		
26	REC_MTL[4].P1	{WinPLCStationName}.Float.rMTL_P1_Z4	Always		
27	REC_MTL[4].P5	{WinPLCStationName}.Float.rMTL_P5_Z4	Always		
28	REC_MTL[4].P8	{WinPLCStationName}.Float.rMTL_P8_Z4	Always		
29	REC_MTL[4].P9	{WinPLCStationName}.Float.rMTL_P9_Z4	Always		
*			Always		

OPC Sheet 10

OPC OPCL010.OPC

Description: Screen: Recipe Viewer Server Identifier: ThinknDo.TNDCEOPC Disable: OPCCommRecipeViewer

Read Update Rate (ms): 2000 Percent Deadband: Status:

Remote Server Name: Browse... ☐ Read after writing
☒ Accept Tag Name in the Item column

	Tag Name	Item	Scan	Div	Add
1		//skip preheat should be only one allowed here probably	Always		
2	hfSkipPreheat	{WinPLCStationName}.Flag.hfSkipPreheat	Always		
*			Always		

OPC Sheet 40

OPC OPCL040.OPC

Description: Screen: Metal Tuning Server Identifier: ThinknDo.TNDCEOPC Disable: OPCCommMTLTuning

Read Update Rate (ms): 1000 Percent Deadband: Status:

Remote Server Name: Browse... ☐ Read after writing
☒ Accept Tag Name in the Item column

	Tag Name	Item	Scan	Div	Add
1	hfMTLTuneStart	{WinPLCStationName}.Flag.hfMTLTuneStart	Always		
2	sMTLTuneStabilizeDone	{WinPLCStationName}.Flag.sMTLTuneStabilizeDone	Always		
3	hfMTLTuneAbort	{WinPLCStationName}.Flag.hfMTLTuneAbort	Always		
4			Always		
5	hnSPControlValveManualZ[1]	{WinPLCStationName}.Number.hnSPControlValveManualZ1	Always		
6	hnSPControlValveManualZ[2]	{WinPLCStationName}.Number.hnSPControlValveManualZ2	Always		
7	hnSPControlValveManualZ[3]	{WinPLCStationName}.Number.hnSPControlValveManualZ3	Always		
8	hnSPControlValveManualZ[4]	{WinPLCStationName}.Number.hnSPControlValveManualZ4	Always		
9			Always		
10	hnSPTemperConstZ[1]	{WinPLCStationName}.Number.hnSPTemperConstZ1	Always		
11	hnSPTemperConstZ[2]	{WinPLCStationName}.Number.hnSPTemperConstZ2	Always		
12	hnSPTemperConstZ[3]	{WinPLCStationName}.Number.hnSPTemperConstZ3	Always		
13	hnSPTemperConstZ[4]	{WinPLCStationName}.Number.hnSPTemperConstZ4	Always		
14			Always		
15	hnControlModeZ[1]	{WinPLCStationName}.Number.hnControlModeZ1	Always		
16	PID_MTLTuning[1].Kp	{WinPLCStationName}.Float.hrPID_Kp_Z1	Always		
17	PID_MTLTuning[1].Ki	{WinPLCStationName}.Float.hrPID_Ki_Z1	Always		
18	PID_MTLTuning[1].Kd	{WinPLCStationName}.Float.hrPID_Kd_Z1	Always		
19	hnFluidStartTemperZ[1]	{WinPLCStationName}.Number.hnFluidStartTemperZ1	Always		
20	hnMoldStartTemperZ[1]	{WinPLCStationName}.Number.hnMoldStartTemperZ1	Always		
21	hnMoldEndTemperZ[1]	{WinPLCStationName}.Number.hnMoldEndTemperZ1	Always		
22	hfMTLTuneNoTestZ[1]	{WinPLCStationName}.Flag.hfMTLTuneNoTestZ1	Always		
23	sMTStblzTestZ[1]	{WinPLCStationName}.Flag.sMTStblzTestZ1	Always		
24	nMTStblzTimeZ[1]	{WinPLCStationName}.Number.nMTStblzTimeZ1	Always		
25	sMTThetaTestZ[1]	{WinPLCStationName}.Flag.sMTThetaTestZ1	Always		
26	sMTTauTestZ[1]	{WinPLCStationName}.Flag.sMTTauTestZ1	Always		
27	sMT5TauTestZ[1]	{WinPLCStationName}.Flag.sMT5TauTestZ1	Always		
28	MTL_TuneResults[1].Theta	{WinPLCStationName}.Number.nMTThetaTimeZ1	Always		
29	MTL_TuneResults[1].Tau	{WinPLCStationName}.Number.nMTTauTimeZ1	Always		
30	MTL_TuneResults[1].Lambda	{WinPLCStationName}.Number.nMTL_LambdaTune_Z1	Always		

OPC OPCCL040.OPC

Description: Screen: Metal Tuning Server Identifier: ThinknDo.TNDCEOPC Disable: OPCCommMTLTuning

Read Update Rate (ms): 1000 Percent Deadband: Status:

Remote Server Name: Browse... ☐ Read after writing
☒ Accept Tag Name in the Item column

	Tag Name	Item	Scan	Div	Add
31	MTL_TuneResults[1].Km	{WinPLCStationName}.Float.rMTL_KmTune_Z1	Always		
32	MTL_TuneResults[1].Kprime	{WinPLCStationName}.Float.rMTL_KprimeTune_Z1	Always		
33	MTL_TuneResults[1].Kcs	{WinPLCStationName}.Float.rMTL_KcsTune_Z1	Always		
34	MTL_TuneResults[1].P0	{WinPLCStationName}.Float.rMTL_P0Tune_Z1	Always		
35	MTL_TuneResults[1].P1	{WinPLCStationName}.Float.rMTL_P1Tune_Z1	Always		
36	MTL_TuneResults[1].P5	{WinPLCStationName}.Float.rMTL_P5Tune_Z1	Always		
37	MTL_TuneResults[1].P8	{WinPLCStationName}.Float.rMTL_P8Tune_Z1	Always		
38	MTL_TuneResults[1].P9	{WinPLCStationName}.Float.rMTL_P9Tune_Z1	Always		
39			Always		
40	hnControlModeZ[2]	{WinPLCStationName}.Number.hnControlModeZ2	Always		
41	PID_MTLTuning[2].Kp	{WinPLCStationName}.Float.hrPID_Kp_Z2	Always		
42	PID_MTLTuning[2].Ki	{WinPLCStationName}.Float.hrPID_Ki_Z2	Always		
43	PID_MTLTuning[2].Kd	{WinPLCStationName}.Float.hrPID_Kd_Z2	Always		
44	hnFluidStartTemperZ[2]	{WinPLCStationName}.Number.hnFluidStartTemperZ2	Always		
45	hnMoldStartTemperZ[2]	{WinPLCStationName}.Number.hnMoldStartTemperZ2	Always		
46	hnMoldEndTemperZ[2]	{WinPLCStationName}.Number.hnMoldEndTemperZ2	Always		
47	hrMTLTuneNoTestZ[2]	{WinPLCStationName}.Flag.hrMTLTuneNoTestZ2	Always		
48	sMTStblzTestZ[2]	{WinPLCStationName}.Flag.sMTStblzTestZ2	Always		
49	nMTStblzTimeZ[2]	{WinPLCStationName}.Number.nMTStblzTimeZ2	Always		
50	sMTThetaTestZ[2]	{WinPLCStationName}.Flag.sMTThetaTestZ2	Always		
51	sMTTauTestZ[2]	{WinPLCStationName}.Flag.sMTTauTestZ2	Always		
52	sMT5TauTestZ[2]	{WinPLCStationName}.Flag.sMT5TauTestZ2	Always		
53	MTL_TuneResults[2].Theta	{WinPLCStationName}.Number.nMTThetaTimeZ2	Always		
54	MTL_TuneResults[2].Tau	{WinPLCStationName}.Number.nMTTauTimeZ2	Always		
55	MTL_TuneResults[2].Lambda	{WinPLCStationName}.Number.nMTL_LambdaTune_Z2	Always		
56	MTL_TuneResults[2].Km	{WinPLCStationName}.Float.rMTL_KmTune_Z2	Always		
57	MTL_TuneResults[2].Kprime	{WinPLCStationName}.Float.rMTL_KprimeTune_Z2	Always		
58	MTL_TuneResults[2].Kcs	{WinPLCStationName}.Float.rMTL_KcsTune_Z2	Always		
59	MTL_TuneResults[2].P0	{WinPLCStationName}.Float.rMTL_P0Tune_Z2	Always		
60	MTL_TuneResults[2].P1	{WinPLCStationName}.Float.rMTL_P1Tune_Z2	Always		

OPC OPCCL040.OPC

Description: Screen: Metal Tuning Server Identifier: ThinknDo.TNDCEOPC Disable: OPCCommMTLTuning

Read Update Rate (ms): 1000 Percent Deadband: Status:

Remote Server Name: Browse... ☐ Read after writing
☒ Accept Tag Name in the Item column

	Tag Name	Item	Scan	Div	Add
61	MTL_TuneResults[2].P5	{WinPLCStationName}.Float.rMTL_P5Tune_Z2	Always		
62	MTL_TuneResults[2].P8	{WinPLCStationName}.Float.rMTL_P8Tune_Z2	Always		
63	MTL_TuneResults[2].P9	{WinPLCStationName}.Float.rMTL_P9Tune_Z2	Always		
64			Always		
65	hnControlModeZ[3]	{WinPLCStationName}.Number.hnControlModeZ3	Always		
66	PID_MTLTuning[3].Kp	{WinPLCStationName}.Float.hrPID_Kp_Z3	Always		
67	PID_MTLTuning[3].Ki	{WinPLCStationName}.Float.hrPID_Ki_Z3	Always		
68	PID_MTLTuning[3].Kd	{WinPLCStationName}.Float.hrPID_Kd_Z3	Always		
69	hnFluidStartTemperZ[3]	{WinPLCStationName}.Number.hnFluidStartTemperZ3	Always		
70	hnMoldStartTemperZ[3]	{WinPLCStationName}.Number.hnMoldStartTemperZ3	Always		
71	hnMoldEndTemperZ[3]	{WinPLCStationName}.Number.hnMoldEndTemperZ3	Always		
72	hmMTLTuneNoTestZ[3]	{WinPLCStationName}.Flag.hmMTLTuneNoTestZ3	Always		
73	sMTStblzTestZ[3]	{WinPLCStationName}.Flag.sMTStblzTestZ3	Always		
74	nMTStblzTimeZ[3]	{WinPLCStationName}.Number.nMTStblzTimeZ3	Always		
75	sMTThetaTestZ[3]	{WinPLCStationName}.Flag.sMTThetaTestZ3	Always		
76	sMTTauTestZ[3]	{WinPLCStationName}.Flag.sMTTauTestZ3	Always		
77	sMT5TauTestZ[3]	{WinPLCStationName}.Flag.sMT5TauTestZ3	Always		
78	MTL_TuneResults[3].Theta	{WinPLCStationName}.Number.nMTThetaTimeZ3	Always		
79	MTL_TuneResults[3].Tau	{WinPLCStationName}.Number.nMTTauTimeZ3	Always		
80	MTL_TuneResults[3].Lambda	{WinPLCStationName}.Number.nMTL_LambdaTune_Z3	Always		
81	MTL_TuneResults[3].Krn	{WinPLCStationName}.Float.rMTL_KrnTune_Z3	Always		
82	MTL_TuneResults[3].Kprime	{WinPLCStationName}.Float.rMTL_KprimeTune_Z3	Always		
83	MTL_TuneResults[3].Kcs	{WinPLCStationName}.Float.rMTL_KcsTune_Z3	Always		
84	MTL_TuneResults[3].P0	{WinPLCStationName}.Float.rMTL_P0Tune_Z3	Always		
85	MTL_TuneResults[3].P1	{WinPLCStationName}.Float.rMTL_P1Tune_Z3	Always		
86	MTL_TuneResults[3].P5	{WinPLCStationName}.Float.rMTL_P5Tune_Z3	Always		
87	MTL_TuneResults[3].P8	{WinPLCStationName}.Float.rMTL_P8Tune_Z3	Always		
88	MTL_TuneResults[3].P9	{WinPLCStationName}.Float.rMTL_P9Tune_Z3	Always		
89			Always		

OPC OPCL040.OPC

Description: Server Identifier: Disable:

Read Update Rate (ms): Percent Deadband: Status:

Remote Server Name: ☐ Read after writing
☒ Accept Tag Name in the Item column

	Tag Name	Item	Scan	Div	Add
90	hnControlModeZ[4]	{WinPLCStationName}.Number.hnControlModeZ4	Always		
91	PID_MTLTuning[4].Kp	{WinPLCStationName}.Float.hrPID_Kp_Z4	Always		
92	PID_MTLTuning[4].Ki	{WinPLCStationName}.Float.hrPID_Ki_Z4	Always		
93	PID_MTLTuning[4].Kd	{WinPLCStationName}.Float.hrPID_Kd_Z4	Always		
94	hnFluidStartTemperZ[4]	{WinPLCStationName}.Number.hnFluidStartTemperZ4	Always		
95	hnMoldStartTemperZ[4]	{WinPLCStationName}.Number.hnMoldStartTemperZ4	Always		
96	hnMoldEndTemperZ[4]	{WinPLCStationName}.Number.hnMoldEndTemperZ4	Always		
97	hMTLTuneNoTestZ[4]	{WinPLCStationName}.Flag.hMTLTuneNoTestZ4	Always		
98	sMTStblzTestZ[4]	{WinPLCStationName}.Flag.sMTStblzTestZ4	Always		
99	nMTStblzTimeZ[4]	{WinPLCStationName}.Number.nMTStblzTimeZ4	Always		
100	sMTThetaTestZ[4]	{WinPLCStationName}.Flag.sMTThetaTestZ4	Always		
101	sMTTauTestZ[4]	{WinPLCStationName}.Flag.sMTTauTestZ4	Always		
102	sMT5TauTestZ[4]	{WinPLCStationName}.Flag.sMT5TauTestZ4	Always		
103	MTL_TuneResults[4].Theta	{WinPLCStationName}.Number.nMTThetaTimeZ4	Always		
104	MTL_TuneResults[4].Tau	{WinPLCStationName}.Number.nMTTauTimeZ4	Always		
105	MTL_TuneResults[4].Lambda	{WinPLCStationName}.Number.nMTL_LambdaTune_Z4	Always		
106	MTL_TuneResults[4].Km	{WinPLCStationName}.Float.rMTL_KmTune_Z4	Always		
107	MTL_TuneResults[4].Kprime	{WinPLCStationName}.Float.rMTL_KprimeTune_Z4	Always		
108	MTL_TuneResults[4].Kcs	{WinPLCStationName}.Float.rMTL_KcsTune_Z4	Always		
109	MTL_TuneResults[4].P0	{WinPLCStationName}.Float.rMTL_P0Tune_Z4	Always		
110	MTL_TuneResults[4].P1	{WinPLCStationName}.Float.rMTL_P1Tune_Z4	Always		
111	MTL_TuneResults[4].P5	{WinPLCStationName}.Float.rMTL_P5Tune_Z4	Always		
112	MTL_TuneResults[4].P8	{WinPLCStationName}.Float.rMTL_P8Tune_Z4	Always		
113	MTL_TuneResults[4].P9	{WinPLCStationName}.Float.rMTL_P9Tune_Z4	Always		
*			Always		

OPC Sheet 50

OPC OPCL050.OPC

Description: Server Identifier: Disable:

Read Update Rate (ms): Percent Deadband: Status:

Remote Server Name: ☐ Read after writing
☒ Accept Tag Name in the Item column

	Tag Name	Item	Scan	Div	Add
1	xESTOP	{WinPLCStationName}.Input.xESTOP	Always		
2	xMHCRelay	{WinPLCStationName}.Input.xMHCRelay	Always		
3	hfDryCycle	{WinPLCStationName}.Flag.hfDryCycle	Always		
4	hfCycleAbort	{WinPLCStationName}.Flag.hfCycleAbort	Always		
5	sCycleNormal	{WinPLCStationName}.Flag.sCycleNormal	Always		
6	sIDLE	{WinPLCStationName}.Flag.sIDLE	Always		
7	sCycleINIT	{WinPLCStationName}.Flag.sCycleINIT	Always		
8	sPREHEAT	{WinPLCStationName}.Flag.sPREHEAT	Always		
9	sRFGS	{WinPLCStationName}.Flag.sRFGS	Always		
10	sRUN	{WinPLCStationName}.Flag.sRUN	Always		
*			Always		

OPC Sheet 51

OPC OPCCL051.OPC

Description: Server Identifier: Disable:

Read Update Rate (ms): Percent Deadband: Status:

Remote Server Name: Browse... ☐ Read after writing ☒ Accept Tag Name in the Item column

	Tag Name	Item	Scan	Div	Add
*			Always		
*			Always		
*			Always		
*			Always		
*			Always		

OPC Sheet 52

OPC OPCCL052.OPC

Description: Server Identifier: Disable:

Read Update Rate (ms): Percent Deadband: Status:

Remote Server Name: Browse... ☐ Read after writing ☒ Accept Tag Name in the Item column

	Tag Name	Item	Scan	Div	Add
1		#{WinPLCStationName}.Flag.hfStartRecipe	Always		
2	hfCancelDownload	{WinPLCStationName}.Flag.hfCancelDownload	Always		
3	sRXPLCLoaded	{WinPLCStationName}.Flag.sRXPLCLoaded	Always		
4	sRecipeActive	{WinPLCStationName}.Flag.sRecipeActive	Always		
5	hstFileCreateTrigger	{WinPLCStationName}.Flag.hstFileCreateTrigger	Always		
6	hwRecipeName	{WinPLCStationName}.String.hwRecipeName	Always		
7	fCycleAbort	{WinPLCStationName}.Flag.fCycleAbort	Always		
8		#{WinPLCStationName}.Flag.sRecipeAutoWrapAround	Always		
9	nCV_Output_Z[1]	{WinPLCStationName}.Number.nCV_Output_Z1	Always		
10	nCV_Output_Z[2]	{WinPLCStationName}.Number.nCV_Output_Z2	Always		
11	nCV_Output_Z[3]	{WinPLCStationName}.Number.nCV_Output_Z3	Always		
12	nCV_Output_Z[4]	{WinPLCStationName}.Number.nCV_Output_Z4	Always		
13	rSP_MTL_Z[1]	{WinPLCStationName}.Float.rSP_MTL_Z1	Always		
14	rSP_MTL_Z[2]	{WinPLCStationName}.Float.rSP_MTL_Z2	Always		
15	rSP_MTL_Z[3]	{WinPLCStationName}.Float.rSP_MTL_Z3	Always		
16	rSP_MTL_Z[4]	{WinPLCStationName}.Float.rSP_MTL_Z4	Always		
17	rSP_PID_Z[1]	{WinPLCStationName}.Float.rSP_PID_Z1	Always		
18	rSP_PID_Z[2]	{WinPLCStationName}.Float.rSP_PID_Z2	Always		
19	rSP_PID_Z[3]	{WinPLCStationName}.Float.rSP_PID_Z3	Always		
20	rSP_PID_Z[4]	{WinPLCStationName}.Float.rSP_PID_Z4	Always		
21			Always		
22	sTC_OK_FluidZ[1]	{WinPLCStationName}.Flag.sTC_OK_FluidZ1	Always		
23	sTC_OK_FluidZ[2]	{WinPLCStationName}.Flag.sTC_OK_FluidZ2	Always		
24	sTC_OK_FluidZ[3]	{WinPLCStationName}.Flag.sTC_OK_FluidZ3	Always		
25	sTC_OK_FluidZ[4]	{WinPLCStationName}.Flag.sTC_OK_FluidZ4	Always		
26	sTC_OK_MoldZ[1]	{WinPLCStationName}.Flag.sTC_OK_MoldZ1	Always		
27	sTC_OK_MoldZ[2]	{WinPLCStationName}.Flag.sTC_OK_MoldZ2	Always		
28	sTC_OK_MoldZ[3]	{WinPLCStationName}.Flag.sTC_OK_MoldZ3	Always		
29	sTC_OK_MoldZ[4]	{WinPLCStationName}.Flag.sTC_OK_MoldZ4	Always		
*			Always		

OPC Sheet 55

OPC OPCL055.OPC

Description: Always Scan, Every 5 Sec Server Identifier: ThinknDo.TNDCEOPC Disable: 0

Read Update Rate (ms): 5000 Percent Deadband: Status:

Remote Server Name: Browse... ☐ Read after writing ☒ Accept Tag Name in the Item column

	Tag Name	Item	Scan	Div	Add
1	hfHMIHeartbeat	{WinPLCStationName}.Flag.hfHMIHeartbeat	Always		
2	ScanCount	{WinPLCStationName}.System.ScanCount	Always		
3	sFourZoneMachine	{WinPLCStationName}.Flag.sFourZoneMachine	Always		
*			Always		

OPC Sheet 60

OPC OPCL060.OPC

Description: Always Scan, Every 10 Se Server Identifier: ThinknDo.TNDCEOPC Disable: 0

Read Update Rate (ms): 10000 Percent Deadband: Status:

Remote Server Name: Browse... ☐ Read after writing ☒ Accept Tag Name in the Item column

	Tag Name	Item	Scan	Div	Add
1	nMTLSampleTime_T2	{WinPLCStationName}.Number.nMTLSampleTime_T2	Always		
2	nPIDSampleTime_T1	{WinPLCStationName}.Number.nPIDSampleTime_T1	Always		
3	wMHC_PLC_VersionRevision	{WinPLCStationName}.String.wMHC_PLC_VersionRevision	Always		
4	sTC_OK_CHS	{WinPLCStationName}.Flag.sTC_OK_CHS	Always		
5	sTC_OK_CHR	{WinPLCStationName}.Flag.sTC_OK_CHR	Always		
6	sTC_OK_CCS	{WinPLCStationName}.Flag.sTC_OK_CCS	Always		
7	sTC_OK_CCR	{WinPLCStationName}.Flag.sTC_OK_CCR	Always		
*			Always		

OPC Sheet 70

OPC **OPCCL070.OPC**

Description: Server Identifier: Disable:

Read Update Rate (ms): Percent Deadband: Status:

Remote Server Name: ☐ Read after writing
☒ Accept Tag Name in the Item column

	Tag Name	Item	Scan	Div	Add
1		//convert to TC/10=hrTC, using scheduler	Always		
2	nTC_MoldZ[1]	{WinPLCStationName}.Number.nTC_MoldZ1	Always		
3	nTC_MoldZ[2]	{WinPLCStationName}.Number.nTC_MoldZ2	Always		
4	nTC_MoldZ[3]	{WinPLCStationName}.Number.nTC_MoldZ3	Always		
5	nTC_MoldZ[4]	{WinPLCStationName}.Number.nTC_MoldZ4	Always		
6	nTC_FluidZ[1]	{WinPLCStationName}.Number.nTC_FluidZ1	Always		
7	nTC_FluidZ[2]	{WinPLCStationName}.Number.nTC_FluidZ2	Always		
8	nTC_FluidZ[3]	{WinPLCStationName}.Number.nTC_FluidZ3	Always		
9	nTC_FluidZ[4]	{WinPLCStationName}.Number.nTC_FluidZ4	Always		
*			Always		

OPC Sheet 90

OPC **OPCCL090.OPC**

Description: Server Identifier: Disable:

Read Update Rate (ms): Percent Deadband: Status:

Remote Server Name: ☐ Read after writing
☒ Accept Tag Name in the Item column

	Tag Name	Item	Scan	Div	Add
1		//use this tag to check OPC comm for the sheet	Always		
2	PLCRollingMsecCounter	{WinPLCStationName}.System.RollingMsecCounter	Always		
3			Always		
4	PLCReserved10	{WinPLCStationName}.System.Reserved10	Always		
5	PLCSpare3	{WinPLCStationName}.System.Spare3	Always		
6	hnUpdateClock	{WinPLCStationName}.Number.hnUpdateClock	Always		
7	nPLCClockUpdated	{WinPLCStationName}.Number.nPLCClockUpdated	Always		
*			Always		

End of Document

**Mold Heating and Cooling
Pump Package
Operator Interface**

HMI Code Listing

DESIGN AND MAINTENANCE DOCUMENT
MHC PP OI Controls Upgrade

END OF DOCUMENT

***Customer Guide
Mold Heating and Cooling Pump Package Operator Interface Controls
(MHC PP OI)***

***NNR project 703030
Activity 70303020***

**Prepared by:
J. A. Salmond**

Jack A. Salmond 06/26/09

Approved By:

Date:

[Signature] 6/26/09

Reviewed By:

Date:

[Signature] 6/26/09

CUSTOMER GUIDE
MHC PP OI Controls

PAGE INTENTIONALLY LEFT BLANK

CUSTOMER GUIDE
MHC PP OI Controls

Table of Contents

1.0	INTRODUCTION	5
1.1.	PURPOSE	5
1.2.	SCOPE	5
1.3.	GENERAL INFORMATION	5
1.4.	DEFINITIONS, ACRONYMS, AND ABBREVIATIONS	6
1.5.	REFERENCES	6
2.0	GENERAL INFORMATION	8
2.1.	PRODUCT PERSPECTIVE	8
2.2.	SOFTWARE FUNCTIONS	17
2.3.	SOFTWARE TERMINOLOGY AND USAGE	17
3.0	HMI SCREENS	20
3.1.	SCREEN MAP	20
3.2.	STARTUP AND SPLASH SCREEN	22
3.3.	NAVIGATION & STATUS	23
3.4.	RECIPE VIEWER	24
3.5.	OPEN RECIPE DIALOG	25
3.6.	SEGMENT GRAPHER	26
3.7.	RECIPE STATUS	28
3.8.	TREND GRAPH	30
3.9.	HISTORICAL TREND GRAPH	32
3.10.	MANUAL CONSTANT TEMPERATURE	33
3.10.1.	Operator View	33
3.10.2.	Engineering View	34
3.11.	RECIPE EDITING – RECIPE SEGMENTS	36
3.12.	INPUT SEGMENT TIME SET-POINT DIALOG	38
3.13.	SEGMENT OPERATIONS DIALOG	39
3.14.	RECIPE EDITING – REC PID/MTL CONSTANTS	40
3.15.	MTL TUNING ROUTINE	41
3.16.	EVENT VIEWER	44
3.17.	PAG UTILITIES	45
3.18.	ALARMS, WARNINGS, OR ERRORS POPUP DIALOG	46
3.19.	INPUT SERIAL NUMBER DIALOG	47
3.20.	WAIT FOR DOWNLOAD DIALOG	48
3.21.	ALPHA-NUMERIC KEYPAD	49
3.22.	NUMERIC KEYPAD	49
4.0	WINPLC OPERATION	50
4.1.	CONTINUOUS CONTROL ALGORITHMS	50
4.1.1.	PID Algorithm	50
4.1.2.	MTL Algorithm	51
5.0	ENCLOSURE	53
5.1.	LOCATION RELATIVE TO PRESS	53
5.2.	THERMOCOUPLE INPUTS	54

CUSTOMER GUIDE
MHC PP OI Controls

List of Figures

Figure 1: Typical Desired Plastic Compression Mold Temperature Cycle.....	9
Figure 2: Typical Compression Mold and Fluid Temperature Cycle.....	9
Figure 3: Typical Desired Plastic Injection Mold Temperature Cycle.....	10
Figure 4: Typical Injection Mold Temperature Cycle.....	10
Figure 5: Mold Heating and Cooling Central System Hot and Cold Oil Supply.....	11
Figure 6: One Mold Zone of an Individual Pumping Subsystem	12
Figure 7: Schematic Diagram of Controller/Computer for a Subsystem.....	13
Figure 8: Single Zone Schematic of a Four-Zone System	14
Figure 9: Single Zone Schematic of a Two-Zone System	15
Figure 10: Zone Control Valve Operation.....	15
Figure 11: Example Temperature Cycle	16
Figure 12: HMI Screen Map	21
Figure 13: MTL Tuning Cycle Terminology	43
Figure 14: Cascaded Control Loop Block Diagram	51
Figure 15: MHC PP OI Enclosure Location	53
Figure 16: Thermocouple Inputs.....	54

CUSTOMER GUIDE
MHC PP OI Controls

1.0 Introduction

1.1. Purpose

This document is the Customer Guide, or User's Manual, explaining the operation of the Mold Heating and Cooling Pump Package Interface (MHC PP OI) and its control systems. The MHC PP OI is used in the plastics molding department (currently D/26) at the NNSA's Kansas City Plant operated by Honeywell FM&T.

1.2. Scope

The software described by this document will be used to:

- provide manual and automatic control of the pump package,
- control the temperature of a molding zone via a feedback control algorithm,
- provide an operator interface to change the profile parameters for the manual and automatic cycles,
- provide an operator interface to initiate the discrete controls, and
- provide an operator interface to display the status of the system.

1.3. General Information

Project Title:	MHC Controls Upgrade
Platform:	WinPLC and PC
Classification:	Unclassified
Funding:	NNR 703030, Activity 70303020
Commissioners:	Brian Spieker, Matthew Trimmer
Developers:	Josh Salmond
Sites/Machines:	FM&T KCP Plastics Molding Department

Platform specifics:

The control software was created using Entivity Think&Do Live! Programming Pack v5.6 from Automation Direct. It is running on a Direct Logic 205 Series Model H2-WPLC3 WinPLC using the WinCE v2.12 operating system. The runtime software on the WinPLC is Entivity DataPack v7.3.

The operator interface was created using InduSoft Web Studio v6.1+SP4 from InduSoft.com. It is running on a Windows XP+SP2 industrial computer with a USB touch screen interface.

The controller and the HMI communicate via OPC protocol over standard Ethernet cabling.

CUSTOMER GUIDE

MHC PP OI Controls

1.4. Definitions, Acronyms, and Abbreviations

ACRONYMS

FM&T	Federal Manufacturing and Technologies
HMI	Human Machine Interface (Operator Interface)
KCP	Kansas City Plant
MHC	Mold Heating and Cooling
MHC PP OI	Mold Heating and Cooling Pump Package Operator Interface
NNR	Non-Nuclear Readiness Campaigns Program
NNSA	National Nuclear Security Administration
OPC	Object Linking and Embedding (OLE) for Process Control
PAG	Process Automation Group
PC	Personal Computer
PID	Proportional Integral Derivative
PLC	Programmable Logic Controller
WinPLC	WinCE (Windows Embedded) based PLC processor

DEFINITIONS

Greenback	A published formal report documenting projects or the development of new processes at the KCP, which is distributed across the Nuclear Weapons Complex.
MHC PP OI Controls	For this document, this represents the controls necessary to control the temperature of a mold at each molding machine.
Mold Heating and Cooling Central System	Consists of a heater for the “hot” loop and a heat exchanger for the “cold” loop that provides the heating and cooling fluid to each molding machine, see <i>Figure 5</i> .
MTL	Custom metal temperature control algorithm.
Process Variable	PV, Measured value of a parameter that affects a process, e.g., pressure, temperature, velocity, etc.
Recipe	Data file that contains a pre-assigned group of process set-points and parameters.
Set-Point	SP, Desired value of the process variable.

1.5. References

The following greenbacks were extensively consulted in the development and design of the system, and are referenced in the text via a superscript:

¹ X-248	R.H. Floersch, Mold Heating and Cooling Systems	May 1970
² BDX-613-369	R.H. Floersch, Theory and Operation of the Bendix Mold Heating and Cooling Equipment	June 1971
³ BDX-613-742	R.H. Floersch, Fluid Flow Analysis in the Mold Heating and Cooling Control Loop	Sept 1972

CUSTOMER GUIDE
MHC PP OI Controls

⁴ BDX-613-1990	R.H. Floersch, Mold Heating and Cooling Systems	Sept 1978
⁵ BID-A461	R.H. Floersch, Hardware Selection and Design Considerations for Microprocessor Controllers for Mold Heating and Cooling	June 1983
⁶ BDX-613-3181	R.H. Floersch, Microprocessor Controller Software Documentation for Mold Heating and Cooling	Nov 1984
⁷ BDX-613-3390	R.H. Floersch, Microprocessor Controller for Mold Heating and Cooling	Jan 1986

The following greenbacks also contain historical information about Mold Heating and Cooling. These documents were consulted but did not contain any new information affecting the design:

• BDX-613-1111	R.H. Floersch, Experimental Testing for Mold Temperature Responses	May 1975
• BDX-613-1764	R.H. Floersch, Computer Simulation of the Mold Heating and Cooling Control Loop, Using Conventional Control Strategies	Feb 1977
• BDX-613-1979	R.H. Floersch, Mold Heating and Cooling Systems	Mar 1978
• BDX-613-1980	R.H. Floersch, Digital Controller Design, Analysis, and Implementation for the Mold Heating and Cooling System	Sept 1978
• BDX-613-2974	R.H. Floersch, Hardware Selection and Design Considerations for Microprocessor Controllers for Mold Heating and Cooling	Nov 1983
• BDX-613-3477	R.H. Floersch, Microprocessor Controller for Mold Heating and Cooling	Apr 1986
• BDX-613-5427	D.P. Hoffman, Mold Heating and Cooling Microprocessor Conversion	Aug 1994

CUSTOMER GUIDE
MHC PP OI Controls

2.0 General Information

This section provides an overall view of the project.

2.1. Product Perspective

Cyclic temperature control of compression and injection molded parts has been an integral part of plastics fabrication at the KCP since the mid 1960s. Temperature controlled fluid flowing through two or four mold sections, called zones, yield high-quality parts with tight tolerances and low residual stresses.

Therefore the purpose of the MHC PP OI system is to control the fluid and/or metal temperature of a plastic injection or compression mold. The temperature of the fluid in a zone is controlled to the set-point by a standard PID feedback algorithm which modulates the control valves that supply hot and cold fluid. Metal Temperature control uses a special control algorithm developed by Dick Floersch of the KCP to account for special non-linear features of this mode of control^{5,6,7}. The custom MTL control algorithm's control loop is cascaded with the PID loop, reference *Figure 14*. The advantage of using the metal algorithm can be seen in *Figure 1–Figure 4* which illustrate how the fluid set-points were previously manipulated to obtain a desired mold temperature profile^{3,7}. The development of this new algorithm enabled the input of the desired mold temperature profile into the recipe, rather than having to produce the desired profile through trial and error of fluid set-points.

CUSTOMER GUIDE
MHC PP OI Controls

Figure 1: Typical Desired Plastic Compression Mold Temperature Cycle

Figure 2: Typical Compression Mold and Fluid Temperature Cycle

CUSTOMER GUIDE
MHC PP OI Controls

Figure 3: Typical Desired Plastic Injection Mold Temperature Cycle

Figure 4: Typical Injection Mold Temperature Cycle

CUSTOMER GUIDE
MHC PP OI Controls

Figure 5 through Figure 10 will illustrate the components of Mold Heating and Cooling. Figure 5 illustrates the fluid interface of individual subsystems with the Central Heating and Cooling system^{1,4}. The existing Central System uses a boiler for the heater and a cooling tower for the heat exchanger as it was designed to handle some fifteen subsystems.

Figure 5: Mold Heating and Cooling Central System Hot and Cold Oil Supply

CUSTOMER GUIDE
MHC PP OI Controls

Figure 6 provides more detail of the fluid line interface of an Individual Pumping Subsystem, depicted as the upper boxes in Figure 5^{1,4}. An Individual Pumping Subsystem may have two or four mold zones depending on what type of mold and molding machine it interfaces with.

Figure 6: One Mold Zone of an Individual Pumping Subsystem

CUSTOMER GUIDE

MHC PP OI Controls

Figure 7 depicts the computer, or controller, interface of one mold zone at an individual pumping subsystem^{1,4}. The box labeled computer is the item of interest in the MHC PP OI Controls Upgrade. In actuality, the controller controls all of the mold zones at a given subsystem.

Figure 7: Schematic Diagram of Controller/Computer for a Subsystem

CUSTOMER GUIDE
MHC PP OI Controls

Figure 8 and Figure 9 provide more detail of the control elements needed at an individual molding machine to interface with the Mold Heating and Cooling Central System². Note that the figures only represent one zone of a local system, and that the only difference between the two figures is that one is a four-zone machine, while the other is a two-zone machine.

Figure 8: Single Zone Schematic of a Four-Zone System

CUSTOMER GUIDE MHC PP OI Controls

Figure 9: Single Zone Schematic of a Two-Zone System

Figure 10 displays in detail how the pneumatic control valves for the zones are configured.

Figure 10: Zone Control Valve Operation

CUSTOMER GUIDE

MHC PP OI Controls

The PID Algorithm outputs 0 to 100% and is clamped at both ends; it can output no less than 0%, and no greater than 100%. Full Cold Fluid enters the zone loop at a valve setting of 0%. Full Hot Fluid enters the zone loop at a valve setting of 100. At a 50% valve setting, the fluid in the zone loop recirculates with no hot or cold fluid introduced. Therefore, a control feedback sample error of zero should result in a valve setting of ~50%. A PID output of 100% will result in a valve setting of 100%; a PID output of 0% will result in a valve setting of 0%. The supply and return of fluid is a balanced mass system; thereby balancing the valve operation such that if a supply valve is at 14%, the return valve will be at 14%.

The MHC PP OI is required to run user defined recipes to cycle the temperature of a mold on a plastic injection or compression molding machine. Typically, a four-zone machine will be an injection molding machine. A two-zone machine may be either a compression or injection molding machine. An example of how mold zones may be configured for a four-zone machine might be: Zone 1 - punch base, Zone 2 - punch insert, Zone 3 - cavity insert, Zone 4 - cavity base.

Each mold zone recipe and control parameters are unique. The zone recipe can be defined to control the temperature of either the heat transfer fluid, or the mold. A zone recipe consists of up to twenty segments. Each segment consists of a temperature set-point (mold or fluid), time duration, and whether or not the segment should be a ramp or dwell segment. The first two segments of a recipe, 0 and 1, are reserved for mold and machine conditioning, called Preheat. An artificial profile of a mold temperature cycle can be seen in *Figure 11*.

Figure 11: Example Temperature Cycle

The entire recipe is executed by the selected control algorithm that controls a valve that lets hot or cold heat transfer fluid into the zone to attain its defined recipe set-point. The recipe is developed and stored on the PC, then transmitted to the controller at run-time. It

CUSTOMER GUIDE

MHC PP OI Controls

should also be noted that the segment is defined by the time; meaning that when the segment time set-point has expired, control will continue to the next segment. The temperature is merely a target to be reached within the time specified. Intimate knowledge or recipe experimentation must be used to configure a recipe that can be controlled to its set-points.

2.2. Software Functions

The primary purpose of the system software will be to provide continuous and discrete control of the Pump Package interface to the Mold Heating and Cooling Central System. Software will reside on both the HMI, and the WinPLC.

The HMI is used to

- Store recipe data on disk
- Store historical trend data
- Edit recipes
- Select and start a recipe
- Initiate discrete logic by pressing a screen image
- Monitor process variables
- Manually enter set-points.

Three HMI logins are available to the user. Valid logins are Operator, Engineer, and PAG. Operator does not require a password. Engineer and PAG logins are password protected.

The WinPLC controller software will run all control algorithms, handle program execution, read data from and write data to the HMI, and aid screen navigation.

2.3. Software Terminology and Usage

Preheat	Segments 0 and 1 are the preheat segments of the recipe profile. Segments 2 and greater, up to 19, are the run segments. A recipe will always wait at the end of segment 1 for a run signal (either Dry Cycle or Press Relay) before continuing to segment 2. The time durations for segments 0 and 1 may be 0. The time duration for segment 2 must be at least 1 second.
Manual Preheat	A type of recipe that requires user input to start each and every part that is made. This type of recipe also requires a serial number input. When this recipe has completed, all zones will continue to control to their last segment's temperature set point.
Auto Preheat	A type of recipe that only requires user input on the first part. After all segments of all zones on the first part's recipe has completed, the recipe will automatically restart segment 0 for all zones. Sometimes referred to as 'Auto Wrap Around'.

CUSTOMER GUIDE
MHC PP OI Controls

Skip Preheat	A recipe option that will skip the segment times of segments 0 and 1, its temperature set point will be segment 1's, and the recipe will wait for the run signal (either Dry Cycle or Press Relay) before continuing to segment 2. This option will skip preheat for only the first part if the recipe is in Auto Preheat mode. This option will skip preheat for every part if in the Manual Preheat mode. This option is useful if the mold has been warmed up and running an Auto Preheat part.
Run Preheat	Default option to use segments 0 and 1.
Dry Cycle	Manual run signal from screen push button to allow the recipe to start segment 2.
Press Relay	Electrical run signal to allow the recipe to start segment 2. This signal may be wired to a mold closure, a push button, etc.
Control Mode	Each zone has a control mode, which must be either Fluid Const Temp, Fluid Recipe Profile, Mold Const Temp, or Mold Recipe Profile.
Fluid Const Temp	A control mode that controls a zone's heat transfer fluid temperature to a fixed temperature.
Fluid Recipe Profile	A control mode that controls a zone's heat transfer fluid temperature to a defined segmented time-temperature profile.
Mold Const Temp	A control mode that controls a zone's actual mold temperature to a fixed temperature.
Mold Recipe Profile	A control mode that controls a zone's actual mold temperature to a defined segmented time-temperature profile.
K_p	Proportional gain tuning parameter that affects the controlling response of the control valve to a change in set point. A higher value will create a faster response with more overshoot.
K_i	Integral gain tuning parameter that affects the controlling response of the control valve to a change in set point. Used to remove steady state error. A higher value will dampen the response and may cause oscillations.
K_D	Derivative gain tuning parameter that affects the controlling response of the control valve to a change in set point. Temperature control needs very little derivative gain.
Dead Time, θ	MTL Algorithm control parameter. The length of time between a change in set point and the noticeable movement in the temperature of the mold in the direction of the SP change. A mold with a small dead time translates to a faster control response.
Time Constant, τ	MTL Algorithm control parameter. The time required for the process variable to achieve 63.2% of the SP change.

CUSTOMER GUIDE
MHC PP OI Controls

Closed Loop Time Constant, λ	MTL Algorithm control parameter. Increasing the closed loop time constant, or decreasing the entered gain, have the effect of damping the response.
Entered Gain, K'	MTL Algorithm control parameter. All historical recipes use a value of 0.33.
Steady State Gain, K_M	MTL Algorithm control parameter. The ratio of the metal temperature to the fluid temperature after five time constants.
Dwell Segment	A timed segment where the temperature stays constant through time duration.
Ramp Segment	A timed segment where the temperature either increases or decreases during the time duration. The segment completes when the time duration has expired, not when the final temperature has been reached.
Cycle Abort	A method to end the cycle before the programmed end has been reached.
Cycle Abort Temp SP	The temperature set point that a zone will be controlled to in the event of a Cycle Abort.
Mold PreHT Temp Limits	Choosing the option will require that the actual temperature of the mold must be between these limits at the conclusion of segment 1, before proceeding to segment 2.
Cycle Status	Will either be IDLE, Initiated, PREHEAT, PRHT Wait Mold Limits, RFCS (Ready For Cycle Start), RUN
Valve Output %	See <i>Figure 10: Zone Control Valve Operation</i>
ISE	Integral Squared Error (ISE) can be used as a measure of repeatability between parts and between sets of control parameters. Units are percent of range squared.

CUSTOMER GUIDE
MHC PP OI Controls

3.0 HMI Screens

The screens available to the user depend on who the user is logged-in as at run time. Three login users are available: Operator, Engineer, and PAG. Operator has access to screens: *Recipe Viewer*, *Recipe Status*, *Trend Graph*, *Historical Trend Graph*, and *Manual Constant Temperature*. Engineer has access to Operator screens plus: *Recipe Editing – Recipe Segments*, *Recipe Editing – Rec PID/MTL Constants*, and *Event Viewer*. PAG has access to Operator and Engineer screens plus the *PAG Utilities* screen.

The following syntax will be used to denote screen items: *Screen Name* in italics, [Button Name] and/or [Field Name] in brackets.

3.1. Screen Map

The screens available and their navigational relationships on the HMI can be seen in *Figure 12*.

[illegible]

Figure 12: HMI Screen Map

CUSTOMER GUIDE

MHC PP OI Controls

3.2. Startup and Splash Screen

Startup consists of *Navigation*, *Status*, and *Splash*. *Navigation* & *Status* are most always visible. *Splash* can be accessed by pressing [About ...] in the lower left area of the screen on the *Navigation* toolbar. *Splash* contains the SQA version numbers of the HMI and PLC software.

CUSTOMER GUIDE

MHC PP OI Controls

3.3. Navigation & Status

Navigation & Status will always be visible to the user.

Navigation screen is the toolbar on the left that houses an array of buttons to navigate to other available screens; buttons available depend on the user who is logged-in at the time.

Status screen provides indication of the WinPLC status, E-STOP condition, login functions, and a button to exit the program. To login as 'Operator': touch [Log On Operator]. To login as 'Engineer': touch [Log On Engineer] which will bring up a password dialog, entering the correct password will login the user as 'Engineer'. [Exit MHC] will exit Windows.

CUSTOMER GUIDE

MHC PP OI Controls

3.4. Recipe Viewer

RECIPE VIEWER - FOR OPERATOR VERIFICATION AND TO START RECIPES

Select Recipe

Start Recipe

Close Recipe

Run Preheat Skip Preheat

ZONE

Zone 1

Zone 2

Zone 3

Zone 4

SEGMENTS

Seg 0-9 Seg 10-19

View Segment Grapher

RECIPE NAME Viewing Recipe from Disk

myRecipe

PART NUMBER(S) RECIPE DESCRIPTION Est. Cycle Time

45678, 123456-78 This part is used to make a whachamigger for the whachamacallit. 00:33:00

SEG#	TIME	TEMP	Ramp	Ramp Rate	Segment Comment
0	300	100	☐	0.000000	This segment is used to preheat the mold.
1	300	175	☒	0.250000	Ramp up the temperature, then place the part in the mol...
2	600	275	☐	0.000000	Process the part for 5 minutes.
3	180	290	☐	0.000000	Process the part at a slightly higher temperature for 3 ...
4	600	100	☒	-0.316667	Ramp to cooldown and remove the part.
5	0	50	☐	0.000000	
6	0	50	☐	0.000000	
7	0	50	☐	0.000000	
8	0	50	☐	0.000000	
9	0	50	☐	0.000000	

Manual Preheat

Cycle Abort Temp SP

Last SP

Mold PreHT Temp Limits

Upper Limit:

Lower Limit:

Control Mode

Fluid Recipe Profile

Fluid Consts:

K_P

K_I

K_D

12.0

1.5

0.1

Mold Dynamic Constants:

Dead Time (sec) θ

Time Constant (sec) τ

Closed Loop Time Constant (sec) λ

Entered Gain Default: 0.33 K'

Steady State Gain K_m

Recipe Viewer is accessed via [Recipe Viewer] on the *Navigation* toolbar. This screen is used by the Operator to select, view, and run a recipe. It displays a read-only copy of the recipe parameters. What is shown in the bottom section depends on which user is logged-in.

The only buttons available are on the left side of the screen (except [Help]):

- [Select Recipe] will access *Open Recipe Dialog*.
- [Start Recipe] will either open *Input Serial Number Dialog* or *Wait for Download Dialog* depending on whether the recipe is setup as Manual Preheat (which requires the serial number), or as Auto Preheat.
- [Close Recipe] will close the current recipe and clear the screen.
- [Run Preheat] [Skip Preheat] to determine whether segments 0 and 1 are used.
- [Zone 1] [Zone 2] [Zone 3] [Zone 4] will display selected zones recipe parameters. Zone 3 and 4 buttons will be available if the recipe is a four zone recipe.
- [Seg 0-9] will show the first 10 segments for the highlighted zone.
- [Seg 10-19] will show the last 10 segments for the highlighted zone.
- [View Segment Grapher] will open the popup *Segment Grapher*.

CUSTOMER GUIDE

MHC PP OI Controls

- [Help] is located in the upper right corner of the screen; the question mark book icon. This button will open a popup window that contains help topics for this screen.

3.5. Open Recipe Dialog

Select Recipe to load on screen

Available Recipes	Part Number(s)
a2zone	123456-00, 654321-01, 8765309, 1...
AllFieldsFilled	123456-00, 654321-01, 8765309, 1...
AllMTLnoRamp	123456-00, 654321-01, 8765309, 1...
aRamp	123456-00, 654321-01, 8765309, 1...
aRampF	123456-00, 654321-01, 8765309, 1...
aStepF	123456-00, 654321-01, 8765309, 1...
aStepF1	123456-00, 654321-01, 8765309, 1...
myRecipe	45678, 123456-78
tTSegmentNumbers	123456-00
vb2	
w	
w2zone	1,2,3

Home

Page Up

Up

Down

Page Down

End

Open Recipe Dialog popup appears when the user has selected [Select Recipe] or [Open] from another screen. This screen is used to select which recipe the user would like to load into the HMI. The maximum number of recipes available is limited to 90.

Buttons available:

- The user may select any line.
- [Open] will load the selected recipe on the HMI.
- [Cancel] will close the dialog window.
- [Home] will scroll the user and highlight the top-most recipe.
- [Page Up] [Page Down] will scroll the user one page in the respective direction.
- [Up] [Down] will scroll the user one line in the respective direction.
- [End] will scroll the user and highlight the bottom-most recipe.

CUSTOMER GUIDE

MHC PP OI Controls

3.6. Segment Grapher

Segment Grapher is accessed from any of the recipe viewing or configuration screens. The screen shows a visual time–temperature profile for all zones. The graph shows whether the segment is a dwell or ramp segment, the time length of each segment, and the temperature of each segment denoted by the dots at the beginning and end of each segment. If the user adjusts the time or temperature scales, all graphs will update simultaneously. The user should also notice that the header contains the name of the recipe whose data points are being graphed. Two or four zones will be displayed depending on the recipe configuration.

Custom buttons available:

- [Close Screen] will return the user to the previous screen.
- [Time Max] will set the upper end of the X scale for all graphs.
- [Time Min] will set the lower end of the X scale for all graphs.
- [Temp Max] will set the upper end of the Y scale for all graphs.
- [Temp Min] will set the lower end of the Y scale for all graphs.
- [Reset Time/Temper Defaults] resets the X and Y scales to a default setting.

CUSTOMER GUIDE
MHC PP OI Controls

Buttons inherent in the trend object:

- [Window Zoom] allows the user to select a zoom box by touching the desired location of the upper left part of the box anywhere on the graph and dragging to the desired bottom right location. The scale change will affect all graphs.
- [Horizontal Zoom] allows the user to zoom into an X axis range without changing the Y scale. Touching anywhere on the graph will create a dotted line of one edge of the zoom, and then touch to either side of the temporary line to zoom. Once the button is pressed on a particular graph, the selections must be made on that graph; however, once completed the scale change will affect all graphs.
- [Vertical Zoom] allows the user to zoom into a Y axis range without changing the X scale. Touching anywhere on the graph will create a dotted line of one edge of the zoom, and then touch to either side of the temporary line to zoom. Once the button is pressed on a particular graph, the selections must be made on that graph; however, once completed the scale change will affect all graphs.
- [Zoom In] [Zoom Out] zooms in/out the X and Y scales of all graphs.
- [Cancel Zoom] is used when the user wants to cancel a [Window Zoom], [Horizontal Zoom], or [Vertical Zoom].
- [Cursor] toggles the availability of the cursor for each graph.
- [Scroll (arrow or bar)] adjusts the X scale for all graphs by adjusting one of them.

Other things to note:

- A segment's start and end dot both must be visible to see the connecting line between them.
- Use the reset button to get all graphs back to default if a zoom has changed the view sufficiently.
- When [Cursor] is on, the user may touch anywhere on the graph. The output of the cursor value is where the vertical cursor line intersects the graphed line, which is a Y scale value. The output is located in the 'Cursor' column underneath the graph. This function is unique to each graph.
- The type of control mode for each zone is displayed next to each graph.
- Warning: segments greater than 1.8 hours will not show a line between the dots – the software vendor is aware of this bug.

CUSTOMER GUIDE

MHC PP OI Controls

3.7. Recipe Status

RECIPE DIAGNOSTIC AND STATUS INDICATORS WHILE RUNNING

Active Recipe empty

Serial # 1234 Elapsed Cycle Time 00:00:00

Theoretical Cycle Time 00:33:00

Start Recipe Restart Recipe: "myRecipe" ?

Current Segment

	Zone 1	Zone 2	Zone 3	Zone 4
Control Mode	Fluid Recipe Profile	Mold Recipe Profile	Mold Const Temp	Fluid Const Temp
Segment	0	0	0	0
Segment Time	00:05:00	00:05:00	00:00:00	00:00:00
Elapsed Time	00:00:00	00:00:00	00:00:00	00:00:00
Segment Temp	100	100	170	78
Segment Ramp?	NO	NO	NO	NO
Mold PreHT	OFF	OFF	OFF	OFF
Temp Limits	OFF	OFF	OFF	OFF
Mold Temp	TC OK 78.0	TC OK 79.3	TC OK 78.8	TC OK 78.9
Fluid Temp	TC OK 159.8	TC OK 78.7	TC OK 78.4	TC OK 78.2
Active Temp SP	100.0	200.0	170.0	78.0
Valve Output%	0	100	100	49
PID ISE	9.01E+002	2.21E+005	0.00E+000	0.00E+000
MTL ISE	0.00E+000	1.74E+003	0.00E+000	0.00E+000

Next Segment

	Zone 1	Zone 2	Zone 3	Zone 4
Seg	1	1	1	1
Time	00:05:00	00:05:00	00:00:00	00:00:00
Temp	175	175	170	170
Ramp?	YES	NO	NO	NO

Real time thermocouple values
Thick lines are Fluid Temperature

Cycle Status: IDLE

Manual Preheat Press Relay: OFF

Dry Cycle Abort Cycle

CENTRAL SYSTEM [degrees F]

TC OK HS 87.2	TC OK HR 87.2	TC OK CS 87.1	TC OK CR 87.4
------------------	------------------	------------------	------------------

Recipe Status shows status for the system as a whole and for each zone. This screen is available only when running in recipe mode and is automatically displayed when a recipe has been started, or may be accessed via [Recipe Status]. The ISE and Central System temperature items are displayed depending on which user is logged-in. Two or four zones will be displayed depending on the recipe configuration.

Buttons and fields available:

- [Start Recipe] will appear after the current part's cycle has ended. The user can make another part with the noted recipe by pressing this button.
- [Segment Temp] may be entered while the recipe is running, only if the zone control mode is in Fluid Constant Temperature or Mold Constant Temperature, as signified by a white background field.
- [Dry Cycle] is employed when the user wants to manually advance the recipe to segment 2 to bypass the Press Relay run signal.
- [Abort Cycle] is used to prematurely end the cycle.
- [Help] is located in the upper right corner of the screen.

CUSTOMER GUIDE
MHC PP OI Controls

Other things to note:

- The red/green indicator next to 'Mold PreHT Temp Limits' for each zone will be green if the limits are off, or if the mold temperature is within the specified limits; the indicator will be red if the mold temperature is outside of the specified limits. This indicator will disappear once segment 2 has started.
- If a thermocouple is not working, its field will flash red and yellow.
- The blue/red color bar underneath each zone's 'Valve Output %' is a visual indication of how much cold/hot heat transfer fluid the valve is letting into the zone.
- ISE is available to the 'Engineer' login. It is calculated from segment 2 through last each time a recipe is run. It can be used as a measure of control and repeatability between parts using the same recipe. Once a baseline has been set for a particular recipe, the control gains and constants can be modified and this number monitored. A smaller ISE translates to better control.
- The Central System temperatures are not connected for existing systems because this would require extensive pipe work. This is a built-in upgrade for future systems.
- Notice that the colors of 'Control Mode', 'Mold Temp', 'Fluid Temp', and 'Active Temp SP' fields are related. For example: Zone 1 is set as Fluid Recipe Profile which is orange, so the user will probably be most interested in monitoring the orange Fluid Temp field; the Active Temp SP field is orange to note the same.

CUSTOMER GUIDE

MHC PP OI Controls

3.8. Trend Graph

Trend Graph is accessed via [Trend Graph] on the *Navigation* toolbar. This screen will display process variables on a digital chart recorder during an active running recipe or will show the last completed recipe's data. The chart on the left displays real-time temperature information which is updated every second. If the user navigates away from this screen and back, the chart on the left will restart. The chart on the right is updated every 10 seconds, such that all of the current recipe's data will be visible within the graphing window. The information on this chart is always available to the user. The 'Current' column represents the current set point value. The 'Cursor' column is an output field where the vertical cursor crosses the graphed line of a given pen. The user may touch anywhere on the graphing window to employ the cursor. The file name which is collecting data is in the title bar, of format: YYYY-MM-DD_HHMM_RecipeName{ _SNxxxx}. Where YYYY is the four digit year, MM is the two digit month, DD is the two digit day, HHMM is the hour and minutes when the recipe was started, RecipeName is the name of the recipe used, and SNxxxx is the serial number of the run, if applicable. This will be the same filename used when retrieving a historical chart on *Historical Trend Graph*. Zone 3 and 4 pen labels will be displayed if the recipe is a four zone recipe.

CUSTOMER GUIDE
MHC PP OI Controls

Custom buttons available:

- [Cursor (input field)] is used to move the cursor to a specific location in the graphing window as a percentage of the scale width. Alternately, the user may directly point to a location.
- [Cursor (left/right arrow)] is used to move the cursor in the respective direction to the window's limit. Left side location is 0%; right side is 100%.
- [Scale Max] [Scale Min] are used to set the Y temperature scale limits.
- [Scale Reset] is used to reset the Y scale to default values.
- [Format Graph] is used to reset the trend object to its default settings.

Buttons inherent in the trend object, in addition to those listed previously:

- [Multiple Sections] will create a separate horizontal section for each pen.
- [Auto Scale] will create a color coded scale on the left of the graphing window for each highlighted pen whose limits are the maximum and minimum for the active data set. All highlighted pens will be on the same graphing space.
- [Pen] will configure the pen settings.
- [Show/Hide Pen] (eye symbol) is used to show or hide the selected pen.

Other things to note:

- Setting changes made to the trend object will be overwritten once the user navigates away from this screen.
- Use [Format Graph] to reformat changes made by [Multiple Sections] and [Auto Scale].
- [Multiple Sections] and [Auto Scale] can be used in conjunction to create a more readable display of data.

CUSTOMER GUIDE

MHC PP OI Controls

3.9. Historical Trend Graph

Historical Trend Graph is accessed via [Historical Trend Graph] on the *Navigation* toolbar. This screen is used to retrieve historical chart data from previous runs. The user interface for this screen is very much like *Trend Graph*.

[Open Historical Trend Graph] will launch a dialog allowing the user to select a historical file for display.

CUSTOMER GUIDE

MHC PP OI Controls

3.10. Manual Constant Temperature

3.10.1. Operator View

NON-RECIPE CONSTANT TEMPERATURE CONTROL, FLUID OR MOLD				
	Zone 1	Zone 2	Zone 3	Zone 4
Control Mode	Fluid Const Temp	Fluid Const Temp	Fluid Const Temp	Fluid Const Temp
Const Temper SP	100	200	170	170
Mold Temp	TC OK 79.6	TC OK 77.5	TC OK 77.2	TC OK 78.2
Fluid Temp	TC OK 220.3	TC OK 80.9	TC OK 79.5	TC OK 76.1
Valve Output %	0	100	100	100
<input type="button" value="Hidden"/> PID Consts Defined	YES	YES	YES	YES
<input type="button" value="Hidden"/> MTL Consts Defined				

CUSTOMER GUIDE

MHC PP OI Controls

3.10.2. Engineering View

NON-RECIPE CONSTANT TEMPERATURE CONTROL, FLUID OR MOLD					
	Zone 1	Zone 2	Zone 3	Zone 4	
Control Mode	Fluid Const Temp	Fluid Const Temp	Mold Const Temp	Mold Const Temp	
Const Temper SP	220	80	80	65	
Mold Temp	TC OK 80.8	TC OK 78.7	TC OK 78.4	TC OK 79.4	
Fluid Temp	TC OK 231.3	TC OK 79.9	TC OK 80.5	TC OK 77.1	
Valve Output %	88	0	50	82	
PID Consts Defined	YES	NO	NO	YES	
<div style="border: 1px solid black; padding: 2px;">PID Default1</div> <div style="border: 1px solid black; padding: 2px;">PID Default2</div>	K_P	1.5	0.0	0.0	12.0
	K_I	4.5	3.0	3.0	1.5
	K_D	0.05	0.00	0.00	0.05
MTL Consts Defined			NO	YES	
Dead Time (sec) θ			0	13	
Time Constant (sec) τ			0	27	
Closed Loop Time Constant (sec) λ			0	27	
Entered Gain K' Default: 0.33			0.0000	0.3300	
Steady State Gain K_m			0.0000	0.9940	
Calculated Gain K_{cs}			0.0000	8.0394	
<div style="border: 1px solid black; padding: 2px;">Run Metal Tuning Test to Establish Metal Constants</div>	P_0		0.0000	0.3352	
	P_1		0.0000	0.2453	
	P_5		0.0000	0.6905	
	P_8		0.0000	0.2284	
	P_9		0.0000	0.0811	

Zone 3 is under PID control until the Mold Constants are fully defined.

Manual Constant Temperature is accessed via [Constant Temperature] in the 'Manual' section of the *Navigation* toolbar. The screen appearance differs depending on whether the Operator or Engineer is logged-in. However, the Operator still has access to PID and MTL control parameters. This screen is used when it is not desirable to run a pre-configured recipe, or when developing a recipe. The set point will stay fixed until changed again and cannot be ramped. The control mode and the control constants can be modified. *MTL Tuning Routine* is accessed by [Run Metal Tuning ...] and is only available to the Engineer login. Two or four zones will be displayed depending on the machine configuration.

Buttons available:

- [Control Mode] is used to toggle between Fluid and Mold Constant Temperature control.
- [Const Temp SP] is used to enter a temperature set point value for the zone.
- [Hidden/Visible] are shown when Operator is logged-in. Pressing [Hidden] will show the respective control parameter fields. These buttons are not shown when Engineer is logged-in.

CUSTOMER GUIDE

MHC PP OI Controls

- $[K_P]$ is used to adjust the proportional gain of the PID controller.
- $[K_I]$ is used to adjust the integral gain of the PID controller.
- $[K_D]$ is used to adjust the derivative gain of the PID controller.
- [PID Default1] can be used when the PID gains have not been established or tuned for a mold yet. This button will set $K_P = 1.5$, $K_I = 4.5$, and $K_D = 0.05$.
- [PID Default2] will set $K_P = 12.0$, $K_I = 1.5$, and $K_D = 0.05$.
- [Dead Time θ] is used to enter the MTL algorithm constant of the mold if it is known. This parameter can be established by running the *MTL Tuning Routine*.
- [Time Constant τ] is used to enter the MTL algorithm constant of the mold if it is known. This parameter can be established by running the *MTL Tuning Routine*.
- [Closed Loop Time Constant λ] is used to enter the MTL algorithm constant of the mold if it is known. This parameter can be established by running the *MTL Tuning Routine*.
- [Entered Gain K'] is used to enter the MTL algorithm constant of the mold if it is known. All historical recipes use a value of 0.33.
- [Steady State Gain K_m] is used to enter the MTL algorithm constant of the mold if it is known. This parameter can be established by running the *MTL Tuning Routine*.
- [Run Metal Tuning ...] will open *MTL Tuning Routine*.
- [Help] is located in the upper right corner of the screen.

Other things to note:

- Notice the warning in the above screen shot for Zone 3 that when the user changes control from Fluid to Mold, the zone will remain in PID control until all mold constants are fully defined.
- Green/Red-Yes/No indicators are used to signal when PID and MTL control parameters are fully defined. The minimum requirement for the PID algorithm is a K_P value. All mold constants must be defined.
- A real-time trend chart has been added to this screen.
- Refer to 2.3 *Software Terminology and Usage* for brief definitions of control parameters.
- Mold Temp and Fluid Temp field will blink red/yellow when there is a problem with a thermocouple.
- This screen should be used sparingly for production because the control constants are not saved. If a production part uses constant temperature control, it may be advantageous for the engineer to create a recipe that uses the constant temperature control mode. Using a recipe will record trend data and offers many more setup parameters that are saved with the recipe file.
- The default PID constants should be used sparingly. This functionality was only provided to give the user basic control of the set point.

CUSTOMER GUIDE

MHC PP OI Controls

3.11. Recipe Editing – Recipe Segments

RECIPE EDITOR - SEGMENT SETPOINT DEFINITION FOR ZONES

RECIPE NAME
myRecipe

RECIPE DESCRIPTION
This part is used to make a whachamigger for the whachamacallit.

PART NUMBER(S)
45678, 123456-78

Generate RTF Report File
for Recipe:
"myRecipe"

Control Mode
Fluid Recipe Profile

Start Recipe

View Segment Grapher

Est. Cycle Time
00:33:00

Cycle Abort Temp SP

Last SP

Upper Limit:
Mold PreHT Temp Limits

Lower Limit:

SEG#	TIME (sec)	TEMPERATURE (deg F)	RAMP? (Y / N)	RAMP RATE (deg F/sec)	SEGMENT COMMENT(S)
0	300	100			This segment is used to preheat the mold.
1	300	175	YES	0.250	Ramp up the temperature, then place the part in the mold after the segment has completed.
2	600	275	NO		Process the part for 5 minutes.
3	180	290	NO		Process the part at a slightly higher temperature for 3 minutes.
4	600	100	YES	-0.317	Ramp to cooldown and remove the part.
5	0				
6					
7					
8					
9					

PREHEAT

Manual Preheat Auto Preheat

Run Preheat Skip Preheat

ZONE

Zone 1 Zone 2

Zone 3 Zone 4

SEGMENTS

Seg 0-9 Seg 10-19

Insert Delete

Recipe Editing – Recipe Segments works in unison with *Recipe Editing – Rec PID/MTL Constants*. The screen is accessed via [Recipe Segments] in the ‘Recipe Editing’ section of the *Navigation* toolbar, if logged-on as Engineer. A recipe, as it is currently defined, is a listing of parameters for a pre-determined time–temperature cycle of the mold for a given process. The parameters include the control mode and constants, and 20 definable “segments” per zone, for up to four zones. Each segment has three user-selectable values. These values are dwell time, temperature set-point, and whether or not the segment is a ramp segment or a dwell segment. This screen will allow the user to change the control mode, create new segments, edit existing segments, insert or delete segments, set abort temperatures, and set metal temperature limits.

Buttons and fields available:

- [New] will prompt the user to name the new recipe file and create a new recipe.
- [Open] will launch *Open Recipe Dialog*.
- [Close] will close the recipe and clear the screen.
- [Save] will update the data in the recipe file.
- [Save As] will prompt the user to specify a new name to save the recipe file to.

CUSTOMER GUIDE

MHC PP OI Controls

- [Delete] will delete the recipe file from the computer disk.
- [Recipe Description] is a free form text field.
- [Part Number(s)] is a free form text field. This value of this field is displayed in *Open Recipe Dialog*.
- [Start Recipe] starts the sequence of events of sending the data file loaded in the HMI memory to the controller to start processing the part and will launch either *Input Serial Number Dialog* or *Wait for Download Dialog* depending on which preheat mode is used.
- [View Segment Grapher] launches *Segment Grapher*. The recipe must be saved before access to this button is granted.
- [Control Mode] is used to change the control mode to one of the four defined modes.
- [Set Point] is used when the control mode is set to a constant temperature mode; this field contains the constant temperature set point.
- [Cycle Abort Temp SP] is used to toggle between using the last defined set point as the cycle abort temperature set point, or using a user defined set point.
- [Cycle Abort Set Point] is the field used to set the cycle abort temperature set point.
- [Mold PreHT Temp Limits] enables the use of Mold Preheat Temperature Limits.
- [Upper Limit:] is the field used to set the upper limit of the Mold Preheat Temperature Limits.
- [Lower Limit:] is the field used to set the lower limit of the Mold Preheat Temperature Limits.
- [Segment Time] launches *Input Segment Time Set-Point Dialog* and is the field used to set the time duration of the segment.
- [Segment Temperature] is the field used to set the temperature set point for the segment.
- [Ramp?] is a toggle switch to enable the segment to be a ramping segment.
- [Segment Comment] is a free form text field to provide detail of the segment's purpose.
- [Manual Preheat] [Auto Preheat] toggles between recipe preheat modes.
- [Run Preheat] [Skip Preheat] to determine whether segments 0 and 1 are used.
- [Zone 1] [Zone 2] [Zone 3] [Zone 4] will display selected zones recipe parameters. Zone 3 and 4 buttons will be available if the recipe is a four zone recipe.
- [Seg 0-9] will show the first 10 segments for the highlighted zone.
- [Seg 10-19] will show the last 10 segments for the highlighted zone.
- [Insert] [Delete] launches *Segment Operations Dialog* to insert/delete a segment.
- [Generate RTF...] is available for PAG login.
- [Help] is located in the upper right corner of the screen.

Other things to note:

- Segment 1 temperature and the mold preheat temperature limits should complement each other. For example, if segment 1 temperature is 100°F, then if using mold preheat temperature limits, they should be banded around the 100°F set point.
- The user may navigate between *Recipe Editing – Recipe Segments* and *Recipe Editing – Rec PID/MTL Constants* freely without having to save the recipe; navigating to any other screen will prompt the user to save the recipe.

CUSTOMER GUIDE
MHC PP OI Controls

3.12. Input Segment Time Set-Point Dialog

Input dwell time set point for Segment 0

Input Option 1: hh:mm:ss : :

OR

Input Option 2: Total hours

OR

Input Option 3: Total minutes

OR

Input Option 4: Total seconds

Reset ALL
fields
to ZERO

Segment 0 0 (seconds)

Setpoint: 00:00:00 (hh:mm:ss)

Input Segment Time Set-Point Dialog is accessed when the user selects any [Segment Time] set point field. This dialog provides the user with an interface to intelligently set the segment time. Since the segment time is stored and displayed in seconds, this interface allows the flexibility for several input methods, and dynamically informs the user of their input before accepting it and returning to the segments screen.

Buttons and fields available:

- [hh:mm:ss] allows the user to individually input the hours, minutes, and second portions of the segment time set point.
- [hours] allows the user to input the set point time as a total amount of hours.
- [minutes] allows the user to input the set point time as a total amount of minutes.
- [seconds] allows the user to input the set point time as a total amount of seconds.
- [Reset ALL fields to ZERO] allows the user to clear all inputs.
- [Accept Set Point] will return the user to the segment screen and replace the given segment set point time with the time entered on the input screen.
- [Cancel] will cancel the input dialog and leave the given segment set point time unchanged.

CUSTOMER GUIDE
MHC PP OI Controls

3.13. Segment Operations Dialog

Segment Operations Dialog is accessed when the user presses [Insert] or [Delete] from *Recipe Editing – Recipe Segments*. When the screen is first called, the 'Selected Zone' and 'Selected Segment' fields will be empty, and [Cancel] only will be available. The user must select a segment to perform the insert/delete operation on before [OK] is made available. When selecting a segment, the user may select any of the following objects of the segment: SEG#, TIME, TEMPERATURE, or SEGMENT COMMENT.

CUSTOMER GUIDE

MHC PP OI Controls

3.14. Recipe Editing – Rec PID/MTL Constants

RECIPE EDITOR - PID/MTL CONTROL CONSTANTS DEFINITION FOR ZONES

New

Open

Close

Save

Save As

Delete

RECIPE NAME: **myRecipe**

RECIPE DESCRIPTION: This part is used to make a whachamigger for the whachamacallit.

PART NUMBER(S): 45678, 123456-78

Generate RTF Report File for Recipe: "myRecipe"

Start Recipe

View Segment Grapher

Est. Cycle Time: 00:33:00

		Zone 1	Zone 2	Zone 3	Zone 4
Control Mode		Fluid Recipe Profile	Mold Recipe Profile	Mold Const Temp	Fluid Const Temp
Const Temp SP		Use Segments	Use Segments	170	170
Fluid Constants:	PID Default1 K_P	12.0	12.0	12.0	12.0
	PID Default2 K_I	1.5	1.5	1.5	1.5
	K_D	0.05	0.05	0.05	0.05
	Dead Time (sec) θ		21	16	
Mold Dynamic Constants:	Time Constant (sec) τ		47	32	
	Closed Loop Time Constant (sec) λ		47	47	
	Entered Gain K' Default: 0.33		0.3300	0.3300	
	Steady State Gain K_m		0.9960	0.9890	

Run Metal Tuning Test to Establish Metal Constants

Recipe Editing – Rec PID/MTL Constants works in unison with *Recipe Editing – Recipe Segments*. The screen is accessed via [Rec PID/MTL Constants] in the ‘Recipe Editing’ section of the *Navigation* toolbar, if logged-on as Engineer. The buttons available on this screen perform the same actions as those listed previously. This screen will allow the user to change each zone’s control mode, constant temperature set point if applicable, PID gains, and mold dynamic constants. This screen also provides an avenue to access *MTL Tuning Routine*.

CUSTOMER GUIDE

MHC PP OI Controls

3.15. MTL Tuning Routine

METAL ALGORITHM TUNING ROUTINE TO ESTABLISH MOLD CONTROL CONTANTS

SETUP:

Close Screen

	Zone 1	Zone 2	Zone 3	Zone 4
K_P	1.5	1.5	1.5	
K_I	4.5	4.5	4.5	
K_D	0.05	0.05	0.05	
Mold Start Temperature	150	175		
Mold End Temperature	350	375		
Fluid Start SP	160	180		
Do not test this zone	☐	☐	Zone Excluded	Zone Excluded
Ctl Mode for zones not tested			PID	CV
Const Temper SP			100	
CV SP %				0

Start Metal Tuning Test for Selected Zones

MONITOR:

	Zone 1	Zone 2	Zone 3	Zone 4
Control Mode	PID	PID	PID	CV
PLC SP Temp	55.0	81.0	100.0	
Valve Output%	0	99	57	50
Mold Temper	TC OK 80.6	TC OK 78.8	TC OK 78.5	TC OK 79.6
Fluid Temper	TC OK 231.7	TC OK 82.2	TC OK 80.9	TC OK 77.3
Stabilize Time	0	0	0	0
Dead Time θ	0	0	0	0
Time Constant τ	0	0	0	0
Status	Not Started	Not Started	n/a	n/a

Abort Testing and Retry

RESULTS:

ID	Theta	Tau	Lambda	K'	Km	Kcs	P0	P1	P5	P8	P9
1	0	0	0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
2	0	0	0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
3	0	0	0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
4	0	0	0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

Save results to Open Recipe File ...?

Save results to an Existing Recipe File ...?

Save results to Text File ...?

Save results to Manual Constant Temper setup?

MTL Tuning Routine is accessed from either *Recipe Editing – Rec PID/MTL Constants* or *Manual Constant Temperature*. This screen provides the user with an interface to setup the tuning routine needed to find the mold control dynamic constants. These constants can only be obtained through this tuning procedure and should never be estimated. These constants are needed to run a zone using the Mold Recipe Profile control mode, or the Mold Constant Temperature control mode.

Buttons available:

- [Close Screen] will navigate the user to the previously viewed screen.
- [PID Default1] [PID Default2] sets PID constants to the same defaults as those listed previously.
- [K_P] [K_I] [K_D] sets PID gains for each zone under PID control.
- [Mold Start Temperature] for zones tested.
- [Mold End Temperature] for zones tested.
- [Fluid Start SP] for zones tested.
- [Do not test this zone] to toggle testing for each zone.

CUSTOMER GUIDE
MHC PP OI Controls

- [Ctl Mode for zones not tested] is toggled between 'PID' or 'CV'.
- [Const Temper SP] is used when the zone is not tested and the Control Mode is set to PID.
- [CV SP%] is used when the zone is not tested and the Control Mode is set to CV.
- [Start Metal Tuning ...] to start the test.
- [Abort Testing ...] to prematurely end the test.
- [Save results to Open Recipe File ...?] to save the MTL tuning results to the currently opened recipe on the HMI, if applicable.
- [Save results to an Existing Recipe File ...?] to save the MTL tuning results to an existing recipe file on the disk.
- [Save results to Text File ...?] to save the MTL tuning results to a plain text file on the disk.
- [Save results to Manual Constant Temp setup?] to send the results to *Manual Constant Temperature*, navigate to that screen, and run manually.
- [Help] is located in the upper right corner of the screen.

Tuning Procedure, reference *Figure 13*:

- Machine and recipe must be in the IDLE state.
- Set PID gains. PID gains should be tuned to the mold before using this test.
- Set desired mold start temperature. This is usually the preheat temperature.
- Set desired mold end temperature. This is usually the cure temperature.
- Set the fluid start set point needed to keep the mold temperature at the mold start temperature within $\pm 3\%$. The Fluid Start SP and Mold Start Temperature should be very close, if not the same, if the mold thermocouple has been properly placed in the mold.
- The user may elect to not test a particular zone; however, at least one zone must be tested.
- If a zone is not to be tested, its control mode and control set point must be set.
- Press [Start Metal Tuning ...].
- Press [Abort Testing ...] to stop the test anytime.
- Zones will attempt to stabilize their temperatures for 5 minutes.
- When all zones have been stable for a minimum of 5 minutes, the test will automatically change the set point to the end temperature.
- Theta (θ , Dead Time) is defined as the time needed for the mold temperature to move outside of 3% of its start temperature.
- Tau (τ , Time Constant) is defined as the time needed for the mold temperature to move 63.2% from its start temperature to its end temperature.
- Lambda (λ , Closed Loop Time Constant) for all zones will be set to the largest Tau for the zones tested.
- K' is automatically set to 0.33.
- The test continues to find K_m (Steady State Gain) after Tau has been established. K_m is calculated once five Tau time periods have elapsed.
- Results will be populated in the grid at the bottom of the screen.

CUSTOMER GUIDE
MHC PP OI Controls

Figure 13: MTL Tuning Cycle Terminology

Other things to note:

- Theta, Tau, Lambda, K' , and K_m are the mold constants needed to use the MTL control algorithm. K_{cs} , P0, P1, P5, P8, and P9 are calculated constants and are provided for information only.
- This tuning procedure was designed to determine heat-up dynamic constants. This tuning procedure can be used to determine cool-down dynamic constants; however, the user should be aware that these constants may be indeterminate. Also note that the user may only specify one set of mold dynamic constants per zone.
- When the tuning procedure has completed or aborted, the temperature set point for selected zones will return to the fluid start set point and will be in PID control. Zones not tested will remain in the control mode selected for those zones.
- Zones will remain at their Mold End Temperature set point until all zones K_m 's have been calculated; then the test will be completed.

CUSTOMER GUIDE

MHC PP OI Controls

3.16. Event Viewer

EVENT VIEWER							
Event Time	Type	Message	Tag Name	Value	Previous	User	Comment
02/25/2009 09:55:50	Display	Open Screen:events.scr		0	0	Engineer	
02/25/2009 09:55:50	Display	Close Screen:recipe editor-constants.scr		0	0	Engineer	
02/25/2009 09:55:50	Display	Close Screen:tempclear.scr		0	0	Engineer	
02/25/2009 09:55:50	Display	Open Screen:tempclear.scr		0	0	Engineer	
02/25/2009 09:55:25	Tags Ch...	Z3 Const Temper SP changed	hnSPTemperConstZ[3]	170	50	Engineer	
02/25/2009 09:55:25	Tags Ch...	ControlMode set by Static Tag Integer: 0=CV, 1=PL...	hmControlModeZ[2]	2	1	Engineer	
02/25/2009 09:55:25	Tags Ch...	ControlMode set by Static Tag Integer: 0=CV, 1=PL...	hmControlModeZ[1]	2	1	Engineer	
02/25/2009 09:55:25	Tags Ch...	Z4 Const Temper SP changed	hnSPTemperConstZ[4]	170	50	Engineer	
02/25/2009 09:55:25	Tags Ch...	Z4 Const Temper SP changed	hnSPTemperConstZ[4]	50	170	Engineer	
02/25/2009 09:55:25	Tags Ch...	Z3 Const Temper SP changed	hnSPTemperConstZ[3]	50	170	Engineer	
02/25/2009 09:55:25	Tags Ch...	ControlMode set by Static Tag Integer: 0=CV, 1=PL...	hmControlModeZ[2]	1	2	Engineer	
02/25/2009 09:55:25	Tags Ch...	ControlMode set by Static Tag Integer: 0=CV, 1=PL...	hmControlModeZ[1]	1	2	Engineer	
02/25/2009 09:55:25	Display	Open Screen:recipe editor-constants.scr		0	0	Engineer	
02/25/2009 09:55:24	Display	Close Screen:tempclear.scr		0	0	Engineer	
02/25/2009 09:55:24	Display	Close Screen:recipe editor-segments.scr		0	0	Engineer	
02/25/2009 09:55:24	Display	Open Screen:tempclear.scr		0	0	Engineer	
02/25/2009 09:54:20	Security ...	LogOn		0	0	Engineer	
02/25/2009 09:54:10	Security ...	Invalid attempt to log:		0	0	pag	
02/25/2009 09:54:02	Tags Ch...	Z3 Const Temper SP changed	hnSPTemperConstZ[3]	170	50	pag	
02/25/2009 09:54:02	Tags Ch...	ControlMode set by Static Tag Integer: 0=CV, 1=PL...	hmControlModeZ[2]	2	1	pag	
02/25/2009 09:54:02	Tags Ch...	ControlMode set by Static Tag Integer: 0=CV, 1=PL...	hmControlModeZ[1]	2	1	pag	
02/25/2009 09:54:02	Tags Ch...	Z4 Const Temper SP changed	hnSPTemperConstZ[4]	170	50	pag	
02/25/2009 09:54:02	Tags Ch...	Z4 Const Temper SP changed	hnSPTemperConstZ[4]	50	170	pag	
02/25/2009 09:54:02	Tags Ch...	Z3 Const Temper SP changed	hnSPTemperConstZ[3]	50	170	pag	
02/25/2009 09:54:02	Tags Ch...	Z2 Const Temper SP changed	hnSPTemperConstZ[2]	50	200	pag	
02/25/2009 09:54:02	Tags Ch...	Z1 Const Temper SP changed	hnSPTemperConstZ[1]	50	100	pag	
02/25/2009 09:54:02	Display	Open Screen:recipe editor-segments.scr		0	0	pag	
02/25/2009 09:54:02	Display	Close Screen:tempclear.scr		0	0	pag	
02/25/2009 09:54:02	Display	Open Screen:tempclear.scr		0	0	pag	
02/25/2009 09:54:01	Display	Close Screen:metal tuning.scr		0	0	pag	
02/25/2009 09:53:19	Display	Open Screen:metal tuning.scr		0	0	pag	
02/25/2009 09:53:19	Display	Close Screen:metal tuning.scr		0	0	pag	
02/25/2009 09:50:19	Display	Open Screen:metal tuning.scr		0	0	pag	
02/25/2009 09:50:19	Display	Close Screen:tempclear.scr		0	0	pag	
02/25/2009 09:50:19	Display	Close Screen:non-recipe consttemper.scr		0	0	pag	
02/25/2009 09:50:19	Display	Open Screen:tempclear.scr		0	0	pag	
02/25/2009 09:50:04	Display	Close Screen:splash.scr		0	0	pag	
02/25/2009 09:50:03	Security ...	LogOn		0	0	PAG	
02/25/2009 09:50:01	Display	Open Screen:splash.scr		0	0	Operator	
02/25/2009 09:49:25	Display	Open Screen:non-recipe consttemper.scr		0	0	Operator	
02/25/2009 09:49:25	Display	Close Screen:tempclear.scr		0	0	Operator	
02/25/2009 09:49:25	Display	Open Screen:tempclear.scr		0	0	Operator	
02/25/2009 09:49:23	Display	Close Screen:non-recipe consttemper.scr		0	0	Operator	
02/25/2009 09:49:23	Display	Close Screen:tempclear.scr		0	0	Operator	
02/25/2009 09:49:23	Display	Open Screen:tempclear.scr		0	0	Operator	
02/25/2009 09:49:23	Security ...	LogOn		0	0	Operator	
02/25/2009 09:48:42	Display	Open Screen:non-recipe consttemper.scr		0	0	PAG	
02/25/2009 09:48:41	Display	Close Screen:tempclear.scr		0	0	PAG	
02/25/2009 09:48:41	Display	Close Screen:trendhistorical.scr		0	0	PAG	
02/25/2009 09:48:41	Display	Open Screen:tempclear.scr		0	0	PAG	
02/25/2009 09:48:39	Display	Open Screen:trendhistorical.scr		0	0	PAG	
02/25/2009 09:48:39	Display	Close Screen:trend.scr		0	0	PAG	
02/25/2009 09:48:39	Display	Close Screen:tempclear.scr		0	0	PAG	

Event Viewer is accessed via [Event Viewer] on the *Navigation* toolbar, if logged-on as Engineer. This screen displays the history of what screens the user has visited, what variables the user changed, custom messages, and logon/logoff activity. This information may be helpful in diagnosing an unexpected condition. The history is available for the current day only.

CUSTOMER GUIDE

MHC PP OI Controls

3.17. PAG Utilities

PAG Utilities is accessed via [PAG Utilities] on the *Navigation* toolbar, if logged-on as a PAG user. This screen provides a peek into the WinPLC performance and variables status, the PC system information, and calculated MTL constants. The screen also provides PAG users the ability to stroke the control valves directly by changing their % open position. This screen also provides an exit feature out of the HMI to diagnose PC problems.

Buttons available:

- [MsgBox to display ...] will poll the PLC and display its calendar date and time information.
- [DailyRunScript] is used to manually run the scheduled script. The script checks the PLC's battery status, and updates the PLC's calendar date and time to the current Windows date and time.
- [Two Zone Machine] sets the HMI for 2 zones.
- [Four Zone Machine] sets the HMI for 4 zones.
- [Valve Position % open] enables the control mode to set the valve position.

CUSTOMER GUIDE
MHC PP OI Controls

- [(Valve Position %SP field)] sets the actual valve position set point from 0 to 100%.
- [Show calculated values] shows the calculated constants used in the MTL algorithm when running a recipe with Mold control selected.
- [Exit MHC to Windows] exits the InduSoft HMI platform and returns the user to Windows.

3.18. Alarms, Warnings, or Errors Popup Dialog

Alarms, Warnings, or Errors Popup Dialog appears to notify the user of a setup problem, a machine error, a setup warning, or any other message of this type.

CUSTOMER GUIDE
MHC PP OI Controls

3.19. Input Serial Number Dialog The dialog box has a blue title bar with the text "Input Serial Number". The main area has a yellow background. It contains two labels, "Input Serial Number" and "Input Result", each followed by a rectangular input field. The "Input Serial Number" field is white, and the "Input Result" field is gray. At the bottom center is a button with a red square containing a white "X" and the text "Close".

Input Serial Number Dialog appears when the recipe preheat mode is set to Manual Preheat. The user must enter a serial number to continue with production. The serial number field is a free form text field. The serial number will also be part of the historical trend data file name, which will aid the user when viewing historical data.

CUSTOMER GUIDE
MHC PP OI Controls

3.20. Wait for Download Dialog

Wait for Download Dialog appears when the user selects [Start Recipe]. The download process may be cancelled. The HMI first establishes a communication line with the PLC; once communication has been established, the download will proceed. The communication and download time for a recipe should take less than 30 seconds; most recipes will take approximately 15 seconds.

CUSTOMER GUIDE

MHC PP OI Controls

3.21. Alpha-Numeric Keypad

Alpha-Numeric Keypad appears when the user has pressed an input field that accepts alpha and numeric characters.

Hint: If a text field already has many characters, use the backspace key to keep the existing text and append more text to the end. Alternately, the user may click within the text field to insert text.

3.22. Numeric Keypad

Numeric Keypad appears when the user has pressed an input field that accepts only numeric characters.

CUSTOMER GUIDE
MHC PP OI Controls

4.0 WinPLC Operation

4.1. Continuous Control Algorithms

Continuous control drives the process variable to the requested set-points by running analog process error through control algorithms and adjusting analog outputs to drive control elements.

4.1.1. PID Algorithm

The controller has two modes of continuous control: the first mode being fluid PID; the second mode being Metal Temperature control^{5,6}. Fluid PID uses a standard form of the discrete PID equation:

$$F_n = K_p \left[e_n + \frac{K_I T_1}{60} \sum_{i=1}^n e_i + \frac{60 K_D}{T_1} (e_n - e_{n-1}) \right]$$

where:

- F_n → PID controller output at the n^{th} sample period to the control valve position as a percentage
- e_n → PID error at the n^{th} sample period = Set-Point minus Process Value (SP-PV), as a percentage of range
- T_1 → PID sample rate in seconds (fixed at 2 seconds)
- K_P → proportional gain constant, unit-less
- K_I → integral gain constant, units of rate/minute
- K_D → derivative gain constant in minutes

The PID equation's output is a percentage (unit-less). Therefore, the errors are converted to percent of range, thereby making the equation unit-less, before calculating the output. The constant 60 is needed to convert T_1 from units of seconds to minutes. K_I has units of rate/minute; K_D has units of minutes.

CUSTOMER GUIDE
MHC PP OI Controls

4.1.2. MTL Algorithm

Metal Temperature control uses a special control algorithm developed by Dick Floersch^{5,6,7}. Some subscripts and variable notation were changed from the previously published algorithm to provide clarity. When using the MTL mode of automatic control, the MTL and PID algorithm are combined in a cascaded control loop as illustrated in *Figure 14*.

Figure 14: Cascaded Control Loop Block Diagram

The special Metal Temperature control law equation is as follows:

$$M_m = K' K_{CS} (e_m - P_0 e_{m-1} - P_1 e_{m-2}) + P_5 M_{m-1} + P_8 M_{m-N-1} + P_9 M_{m-N-2}$$

where:

$$K' = K_{entered} = \text{default } 0.33; \text{ all current recipes use } 0.33.$$

$$K_{CS} = \frac{\tau_m (1 - e^{-fT/\lambda})}{K_m e^{-fT/\tau_m}} \quad K_m = \frac{\text{MetalTemperture}}{\text{FluidTemperature}} @ 5 * \tau_m$$

$$P_0 = e^{-T/\tau_m} - \frac{e^{-fT/\lambda} - e^{-T/\lambda}}{1 - e^{-fT/\lambda}} \quad P_1 = \frac{e^{-T/\tau_m} (e^{-fT/\lambda} - e^{-T/\lambda})}{1 - e^{-fT/\lambda}}$$

$$P_5 = e^{-T/\lambda} \quad P_8 = 1 - e^{-fT/\lambda} \quad P_9 = e^{-fT/\lambda} - e^{-T/\lambda}$$

$$N = \frac{\theta}{T}, \text{ integer value} \quad f = N + 1 - \frac{\theta}{T}, \text{ fractional decimal value}$$

CUSTOMER GUIDE
MHC PP OI Controls

M_m	→	metal controller output at the m^{th} sample period to the control valve position as a percentage; M_{m-1} is the metal controller output one sample period previous, etc.
e_m	→	metal error at m^{th} sample period = SP-PV, as a percentage of range; e_{m-1} is the metal error one sample period previous, etc.
λ	→	Lambda, specified closed loop time constant in seconds
τ_m	→	Tau, mold metal temperature (plant) time constant in seconds, defined as the elapsed time, after dead time, for the response to reach 63.2% of the input
K_m	→	mold metal temperature (plant) steady state gain. It should be noted that a variation of the calculation of this gain is the substitution of the Fluid SP for the Fluid Temperature; however, at five times the mold time constant, these two numbers are nearly identical.
θ	→	Theta, dead time in seconds, defined as the elapsed time before the sensing element sees a response from the input
T	→	Metal sample rate (sample period) in seconds, also called T_2 (fixed at 10 seconds)
N	→	truncated integer value of dead time divided by sample period (example: if $\theta = 77$, and $T = 10$, then $N = 77/10 = 7.7 = 7$)
f	→	fractional position (decimal value) of the dead time between periods (example: if $\theta = 77$, and $T = 10$, then $f = 7 + 1 - 7.7 = 0.3$)

The MTL equation's output is a percentage. Therefore, the errors are converted to percent of range, thereby making the equation unit-less, before calculating the output.

CUSTOMER GUIDE
MHC PP OI Controls

5.0 Enclosure

5.1. Location Relative to Press

The enclosure will be located in close proximity to the press and the pump cabinet, typically as shown in *Figure 15*.

An ON/OFF switch is located on the top right side of the cabinet.

Figure 15: MHC PP OI Enclosure Location

CUSTOMER GUIDE
MHC PP OI Controls

5.2. Thermocouple Inputs

Thermocouple inputs can be reconfigured on the left side panel of the enclosure to accommodate varying machine configurations. Plugs should be used to prevent open loops when a zone is not being used as shown in *Figure 16*.

Figure 16: Thermocouple Inputs

There should be no occasion for most users to access the inside of the enclosure. The software was written such that a keyboard and mouse will not be needed, as all functions can be accessed from the touch panel.

CUSTOMER GUIDE
MHC PP OI Controls

END OF DOCUMENT