



CSZ
DIMENSION®
Series 60
User's Manual

Publication No. HA090933U002

October 1998

Table of Contents

Manual Scope	v
Manual Organization and Contents	v
Conventions	vi
How to Get Help	vii
 Section 1 Operator Interface Description	
Section Objectives	1—1
Introduction	1—1
Front Panel Controls	1—2
Help Key	1—3
Diamond Key (Four-way Keypad)	1—3
Multi-Function Knob	1—4
Function Key Labels	1—5
Memory Card Slot	1—5
Using Memory Cards	1—5
Memory Card Operation	1—6
Memory Card Information	1—8
Screen Display Format	1—9
Display Summary	1—11
 Section 2 Operation	
Process Control	2—18
Control Loop Definition	2—18
Process Control	2—18
Terminology	2—19
PROCESS CONTROL LOOPS	2—22
PID Loop Tuning	2—27
Tuning the Loop	2—28
Fine Tuning the Process	2—29
Unit Setup	2—30
Alarm Acknowledgment	2—33

Section 3 Dimension Programmer

Description	3—1
Terminology	3—1
Recipe Definition & Planning	3—3
Entering a Recipe into the Dimension Programmer	3—6
Executing/Starting a Recipe	3—8
Monitoring Program Execution	3—9
Advanced Recipe/Programmer features	3—11
Guaranteed Soak	3—12
Segment Recycles	3—13
Graphic Display of Selected Profile	3—14
Display Profile	3—14

Section 4 Configuration

Description	4—1
Configuration Access	4—1
Unit Configuration	4—3
Tag Name Configuration	4—4
Patch I/O Configuration	4—6
Display Configuration	4—9
Input and Output Configuration and Calibration	4—11
Analog Inputs	4—12
Remote Analog Inputs	4—13
Analog Output Configuration and Calibration	4—14
Communications	4—16

Appendix A Hardware/Wiring Connections

Description	A—1
Trouble Shooting	A—4
Multiple Alarms	A—4
Chamber Limit	A—5
Back-Up Limits	A—5
Back-Up Product	A—5
Back-Up 404A And 23	A—6
MCB Trouble Shooting Aids	A—6

Appendix B Communications

Serial Communications Overview	B—1
Electrical Connections	B—2
RS-232 Connections	B—2
RS-422 Connections	B—3
Shield Connection	B—4
Terminal Mode	B—5
Software Drivers	B—5
Error Checking	B—6
Setting Up the Terminal Mode	B—7
Terminal Mode Data Access	B—8
Network Mode	B—8
Software Driver	B—8
Setting Up the Network Mode	B—9
Network Mode Request and Response Messages	B—10
Network Mode Request and Response Message Formats	B—11
End of Text Control Character	B—14
Checksum	B—15
Error Checking	B—17
Test Program	B—17
PC to Dimension Communications – Optimized for Speed	B—18
System Variable Data Access	B—19
System Variable Commands	B—19
PRINT Commands	B—20
LET Commands	B—21
LIST Commands	B—22
STAG Commands	B—24
CTAG Commands	B—24
Reading & Changing the Status of the System Functions via Digital Communication	B—25
System Variable Descriptions	B—26
Example of System Variable Use	B—82
Error Messages	B—84
IEEE-488 Option Installation and Configuration	B—85
IEEE-488 Introduction	B—85
IEEE-488 and Dimension	B—85
IEEE-488 Installation	B—85
Cabling and Connections	B—85

IEEE-488 Configuration	B—86
Dimension 8705 Configuration/Set up	B—86
Dimension 8725 Configuration/Set up	B—87
System Controller Configuration	B—88
Primary GPIB Address	B—89
Secondary GPIB Address	B—89
Timeout Setting	B—89
EOS Byte	B—89
Terminate Read on EOS	B—90
Set EOI with EOS on Write	B—90
Type of Compare on EOS	B—90
Set EOI with Last Byte of Write	B—90
Repeat Addressing	B—91
IEEE-488 and Data Transfer	B—91
Data Transfer	B—91
IEEE-488 Hardware Bus	B—91
Handshake Lines	B—92
Interface Management Lines	B—93
Driver Software	B—93
IEEE-488 Bus Commands	B—94
Application Software	B—96
Dimension Commands	B—96
Sample Programs	B—96

Appendix C Verification/Calibration Procedure

Required Prerequisites	C—1
Required Equipment	C—1
Equipment Specifications	C—2
Dimension Input Verification	C—4
Cold Junction - Calibration	C—6
Input Calibration	C—7
Thermocouple Error	C—11

Index

Preface

This preface:

- Briefly describes the scope, organization, and contents of this manual.
- Identifies and describes related documentation.
- Lists and defines the conventions and terminology used in Dimension® documentation and software.
- Explains how to get help.

Manual Scope

This manual contains information on the following:

- Using the Dimension Controller to operate the CSZ chamber.
- Dimension hardware component functions.

Your system may or may not include all of the products and features discussed in this manual. If your system includes components other than those listed above, refer to related documentation shipped with your system.

Manual Organization and Contents

This manual contains information you will need in order to configure and operate the CSZ Dimension Controller. Manual contents are as follows:

Section 1. Operator Interface

A descriptive overview of Dimension's Graphic Operator Interface, including Operation and Configuration screens, description of graphic icons and menu screens, how to navigate the Graphic Operator Interface, and terminology related to Graphic Operator Interface.

Section 2. Operation

Description of the functions on the Dimension controller used to control the CSZ chamber. Definition, terminology and operation of PID process control loops.

Section 3. Programmer

Describes Dimension Programmer functions and capabilities, includes Programmer terminology, and explains how to enter, execute and monitor a setpoint program.

Section 4 . Configuration

Description of the setup and configuration functions that can be used to customize the Dimension controller.

Appendix A. Hardware/Wiring Connections

Explains how to install and set up Dimension system hardware, and how to make external field wiring and communication connections.

Appendix B. Communications

Explains how to communicate with Dimension via a serial link. Includes all parameters needed for communications. Also describes IEEE-488 communications.

Appendix C. Verification/Calibration Procedure

Provides procedures for calibration of your Dimension and verification of inputs.

Conventions

In Dimension documentation and on-screen instructions, you will be instructed to enter various system commands and information using Graphic Operator Interface front panel controls. The following conventions are used:

- Pressing the operator interface multi-function knob has the same effect as pressing the Enter or Return key on a computer keyboard. When you see “press **ENTER**” printed in an instruction, press the knob to enter a selection or command into the system.

- The operator interface multi-function knob can be turned clockwise or counterclockwise to scroll through or select information. In the documentation, CW = clockwise and CCW = counterclockwise.
- The Graphic Operator Interface (OI) continuously displays a set of graphic icons on the right-hand side of the screen. The meanings and use of these graphic symbols are defined in section 3 of this manual.
- The operator interface displays a graphic CURSOR which indicates the currently active data entry field or selection on a screen. In its "home" position, the cursor is displayed as a blinking square graphic object between the screen title and time indicator on the top line of the screen.
- When a window is displayed on a screen, the cursor disappears and another box with a smaller box inside is displayed.
- When an icon or data entry field is displayed in reverse video on a screen, that icon or field is currently selected.
- When an alarm becomes active, the alarm icon begins to blink. When all alarms are acknowledged on the Alarm Status screen, the icon stops blinking.
- Dimension display parameters are shown in bold type. Example: Configur_Select.

How to Get Help

On many operator interface displays, messages are displayed that will tell you how to proceed or how to correct an error. Also, pop-up "help" windows can be called up by simply pressing the **Help** key (?). Press the Help key again, or the knob, to make the window disappear.

A dark gray rectangular box with a thin black border. Inside the box, the word "Section" is written in a white, sans-serif font on the left, and a large white number "1" is on the right.

Operator Interface Description

Section Objectives

Section 1 provides the following information:

- Description of Dimension Graphic Operator Interface functions
- Functions of Interface graphic icons and menu screens
- How to navigate the Graphic Operator Interface
- Terminology related to Interface Operator and Configuration screens

Introduction

The Dimension Industrial Process Manager graphic operator interface is a self-contained, intelligent unit equipped with its own on-board microprocessor and memory. The operator interface provides all the tools needed for system configuration, operation, and monitoring of a Dimension Industrial Process Manager.

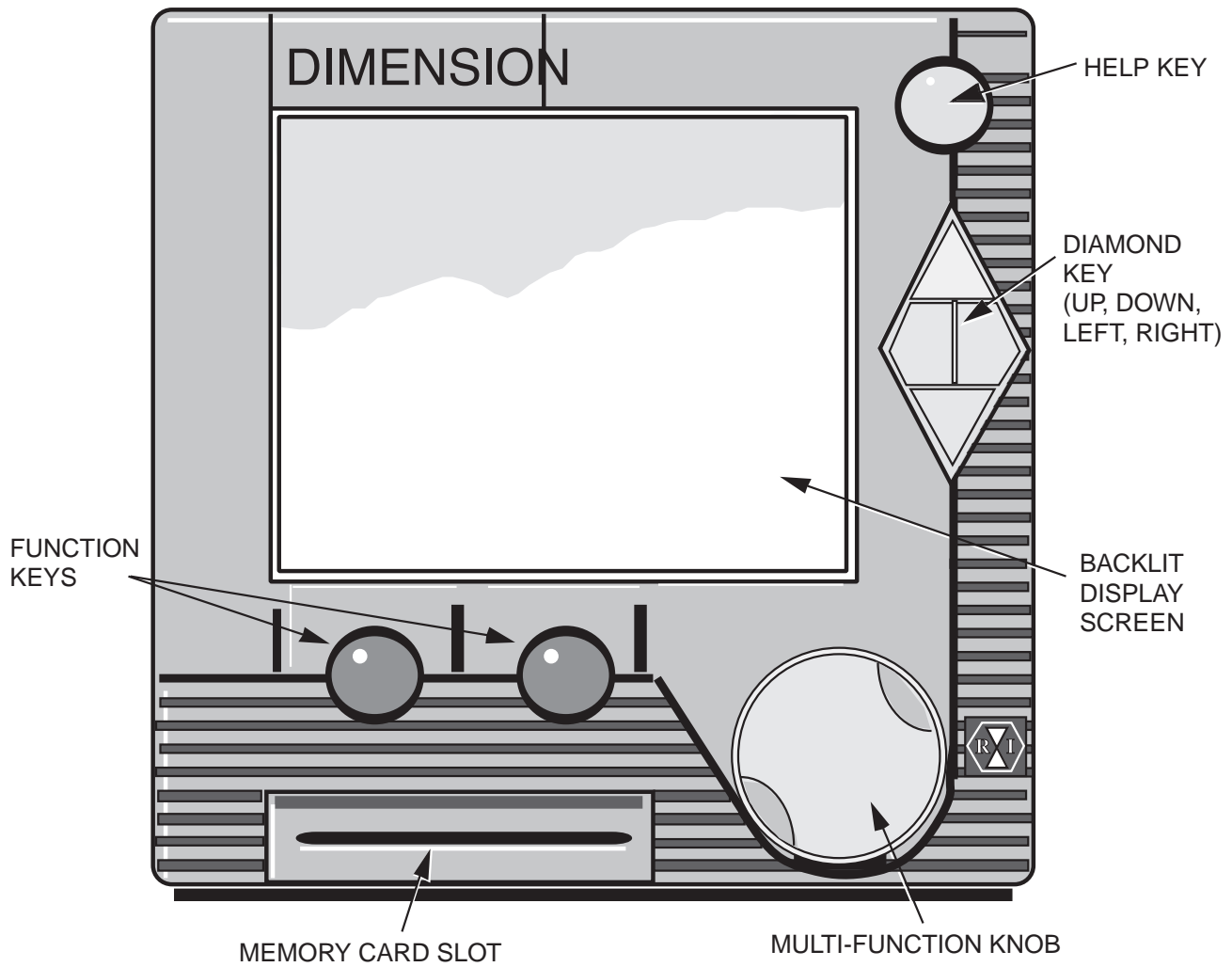
In all hardware configurations, operator interface functions and features are identical. Operator interface features include:

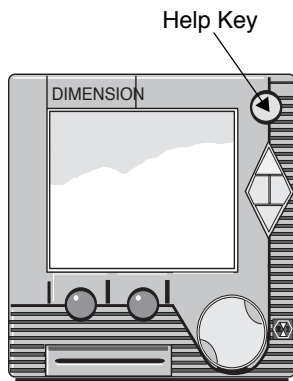
- An easy-to-use set of controls for both configuration and operation
- A set of highly readable, backlit screen displays using both text and graphics
- Dedicated "help" key
- Multi-function knob for data entry and parameter selection
- Two user-programmable "function keys" for combining frequently used multiple steps into one function
- User definable displays
- Screens in text/numeric, X-Y charts, bar charts
- User definable screens and process terminology
- An integrated memory card slot that can read from and write to removable data storage memory cards

Front Panel Controls

The following paragraphs identify and describe the Dimension graphic operator interface front panel controls shown in Fig. 1-1.

Figure 1-1.
Dimension Graphic Operator Interface Front Panel





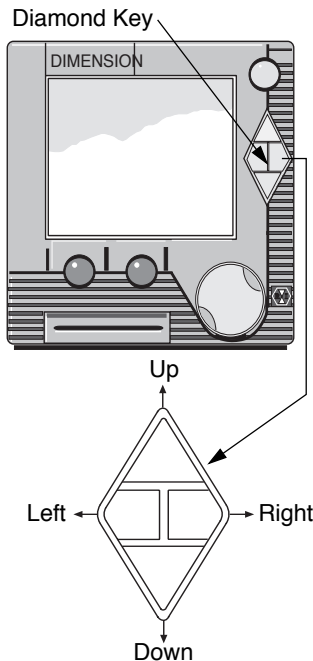
Help Key

Located in the upper right-hand corner of the front panel, the help key is designed to perform two functions:

- Cancel an entry. Pressing the help key when entering a value will cancel the entry.
- Provide on-screen help. Pressing the help key opens a “help” window containing instructions for the currently accessed field. Pressing the help key a second time, or the multi-function knob, closes the window.

➔ **Note:** During some operations, help messages are displayed automatically if you attempt to perform actions not permitted on the currently displayed screen.

Most help messages are an integral part of system software and are not user-definable. However, on the Alarm Display Configuration screen, you may enter alarm-specific messages that will display when the help key is pressed at the Alarm Status screen.



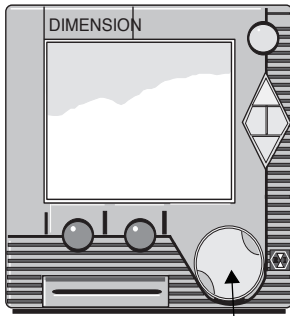
Diamond Key (Four-way Keypad)

The graphic process display screen is organized in a horizontal/vertical hierarchy. A set of six graphic icons displayed on the right side of the screen shows where you are located vertically in the screen display hierarchy by displaying the icon in inverse video (See Fig. 1-3 on page 10).

The diamond key located directly below the help key is used to perform one function: to exit the currently displayed screen and call up another screen. The diamond key has four selections:

- UP** – Accesses the next icon up and displays the first screen for the icon.
- DOWN** – Accesses the next icon down and displays the first screen for the icon.
- LEFT** – Displays the next screen to the left.
- RIGHT** – Displays the next screen to the right.

See “Display Summary” on page 1-11.



Multi-Function Knob

Multi-Function Knob

The multi-function knob (hereafter called the knob) is used to move the graphic “cursor” from field to field on Dimension screen displays. In its “home” (inactive) position, the cursor is located between the screen title and time indicator on the top line of the screen. If a window is displayed, a second, smaller box is displayed inside another box to the left of the cursor (see Fig. 1-3 on page 10).

The following sequence explains how you can make selections and enter or change information on Dimension screen displays.

1. When a screen is first displayed after being called up, the cursor is located in its “home” position on the top row of the screen.
2. If you press the **Help** key, a help window displays that describes screen and knob functions.
3. Press the **Help** key again, or the knob, to close the help window.
4. If any items on the screen are selectable, the knob can be turned CW to move the cursor to the first selectable item on the screen.
5. If the knob is pressed, the item is selected and one of two actions occurs:
 - A window opens for further input.
 - The selected item is displayed in inverse video and you can change the item in one of the following ways:

CHARACTER ENTRY – Display one character at a time by turning the knob CW or CCW and press **ENTER** when the character you wish to select is displayed. This input method displays the full ASCII alphanumeric character set as the knob is turned.

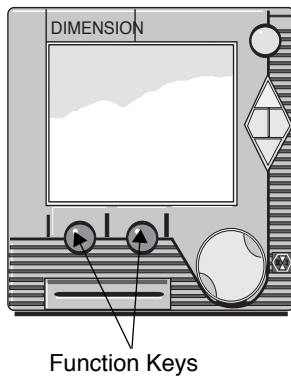
NUMBER ENTRY – Display one number at a time by turning the knob CW or CCW and press **ENTER** when the number you wish to select is displayed. This input method displays numbers in the correct format as the knob is turned. Parameters such as time, date, and passcodes are set with limits, as are control parameters and variables.

➡ **Note:** *If a number is entered outside a field's designated range, the number will not be accepted. The previous value will redisplay in the data entry field. Length varies from field to field.*

PARAMETER SELECTION – As the knob is turned CW or CCW, a list of selectable items is displayed; for example, “Run, Hold, Start.” When the correct entry is displayed, press **ENTER** to select the item.

FUNCTION SELECTION – Some entries are functions that will be executed when **ENTER** is pressed. For example, scrolling to the **CLOSE** entry in a window and pressing **ENTER** closes the window.

➔ **Note:** If an item is selectable (cursor can be moved to it), the help key can be pressed and a help message for the item in the cursor will be displayed. If an item is selected by pressing **ENTER**, a screen change cannot be made until the cursor reappears.



Function Key Labels

PgmStart - push button

- to view current active recipe
- view current segment
- select new recipe to start
- select memory card to save recipes

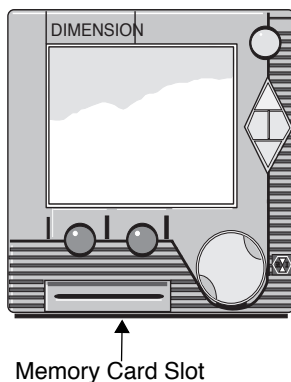
Run/Hold - After recipe is started label will display HOLD. Push button to HOLD (stop clock). Label will now read RUN. Push bottom to resume program.

Memory Card Slot

A slot is provided on the graphic operator interface front panel for insertion of a credit card-sized memory card. The operator interface can read from, write to, delete, or format memory cards.

Information that can be stored on Dimension memory cards includes:

- Programmer profile programs 1–15 (Recipes)
- Total memory image of Dimension configuration and operational parameters (Mem_Img)
- Display configuration (DspConf_)
- Function key configuration (MacroKy_)
- Optional Custom Control Program (CLCstart)



Using Memory Cards

A slot is provided on the operator interface front panel for insertion of credit card-sized memory cards. Configuration, control parameters and data can be retrieved from or stored on these cards.

➔ **Note:** *The memory card's battery life is approximately five years under normal operating condition. Batteries can be replaced; call CSZ customer service department for ordering information.*

Information that can be stored on Dimension memory cards includes:

- Setpoint profile programs (R e c i p e s)
- Total memory image of Dimension configuration and operational parameters (M e m - I m g)
- Display configuration (D S P C o n f)
- Function key configuration (M a c r o k y _)
- Custom features (C L C s t a r t)

Whenever you make changes to Dimension parameters or data, the new information should be saved on a memory card.

Memory Card Operation

The following procedure explains how you can save your new configuration data onto removable memory cards. With your information saved on one or more memory cards, you can load the data into Dimension system memory at any time.

Save your data onto memory cards as follows:

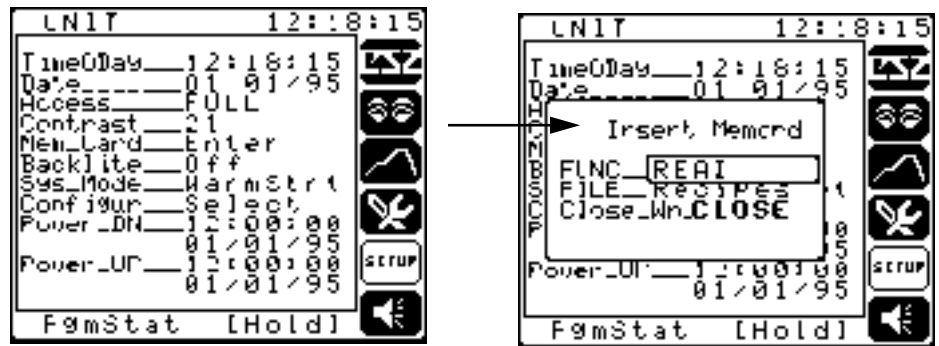
1. Use the diamond key to access the setup icon. Scroll down to the "unit" parameter and press ENTER. The system will display the Unit Operation screen, shown in figure 1-2.
 2. Insert a memory card into the card slot on the operator interface front panel.
- ! **Important:** If you wish to write data to a memory card, be sure that the write-protect switch on the card is set to the "NO" position.
3. Use the knob to select the MEM_CARD field. The system will display the Memory Card Operation Window shown in Fig. 1-2. The following information and functions are provided in this window:
 - The first line describes the current status of the memory card slot: INSERT MEMORY CARD or UNFORMATTED MEMORY CARD. If a card is not installed, "Insert Memcard" is displayed. If an installed card has not been initialized and is unreadable, "Unformat Memcard" will be displayed.



- The function (FUNC) selection field permits you to scroll through and select from a list of functions including READ, WRITE, DIRECT(ory), FORMAT, and DELETE.
- When you select the DIRECT(ory) function, a window permits you to scroll through the directory of files on the card. The name, date, and size (Kbytes) of each file are displayed.
- If READ or WRITE is selected and a Memory card is installed, the selector box moves to the FILE field where the file to be written or read can be specified.
- A status line displays card status parameters including write-protect on/off, battery low/good, read/write fail, and file too large.

Figure 1-2.
Memory Card
Operation Window

Select Memory Card Parameter



A window opens to allow access to the Memory Card Read, Write, Directory Format, and Delete Functions

4. If necessary, format the installed memory card. This will delete all data on the card.
5. To write a file to a card, select the FUNC field. Select the WRITE function. The selector box will move to the FILE field. Select or enter the name of the file that is to be written to the card.

If additional information is required, a window will open with prompts for more information. Enter the additional information as required.

At this point, the data will be written to the memory card under the file name you specified. Repeat this for other files as necessary.

To read a memory card file into system memory, follow the above procedure but specify READ rather than WRITE functions as required.

Memory Card Information

The memory card files contain the following Dimension information:

1. `Recipes` (Recipes):
 - The programmer build (PgBuild) segment table of 256 segments.
 - Recipe Tag Names (ProgName) of the programmer glossary.
 - Starting Segment numbers (Start_Seg) associated with the eight program recipe tag names (default names Recipe_1 to Recipe_8).
2. `Mem_Img` (Memory Image):
 - Unit configuration: Full and limited access passwords, system mode, Function key tag names.
 - Glossary configuration: System glossary (GLOS_SYS), programmer glossary (GLOS_PRG) *excluding Recipe Tag Names* (ProgName), Loop glossary (GLOS_LOP), I/O glossary (GLOS_IO).
 - Process configuration: Programmer configuration (CONF_PRG), Loop configuration (CONF_LOP) *including Auto Tune parameters*, Alarm configuration (CONF_ALM), Output configuration (CONF_OUT).
 - Analog Input configuration: (not calibration).
 - Analog Output configuration: (not calibration).
 - Remote Output configuration.
 - Time Proportioning Output configuration.
 - Contact Output configuration.
 - Communication configuration.
 - The following Process Settings: Programmer Option settings (PgOption), Analog Input Alarm settings (InputAlm), Manual Reset (ManRst), Setpoint Source (SptSrc), Loop Mode (LpMode).
 - Loop PID parameters for PID groups 1–5: (Gain, Reset, Rate).
 - Loop Alarm settings for Alarm Groups 1–5: (Loop_Alm): (HiHiPV, LoLoPV, HiPV, LoPV, HiHiDV, LoLoDV, HiDev, LoDev)
3. `DspConf_` (Dimension Display Configuration):
 - Overview display screens (DCON_OVR).
 - Programmer display screens (DCON_PRG).
 - Loop display screens (DCON_LOP).
 - Logic display screens (DCON_LOG).
 - Alarm Messages (DCON_ALM)

4. MacroKy_ (Function Key Configuration):
 - Function key number 1 function configuration (MacroKy1) *excluding* tag name.
 - Function key number 2 function configuration (MacroKy2) *excluding* tag name.
5. CLCStart (custom features):
 - To be determined by CSZ engineering (optional)

Items Not Stored on Card:

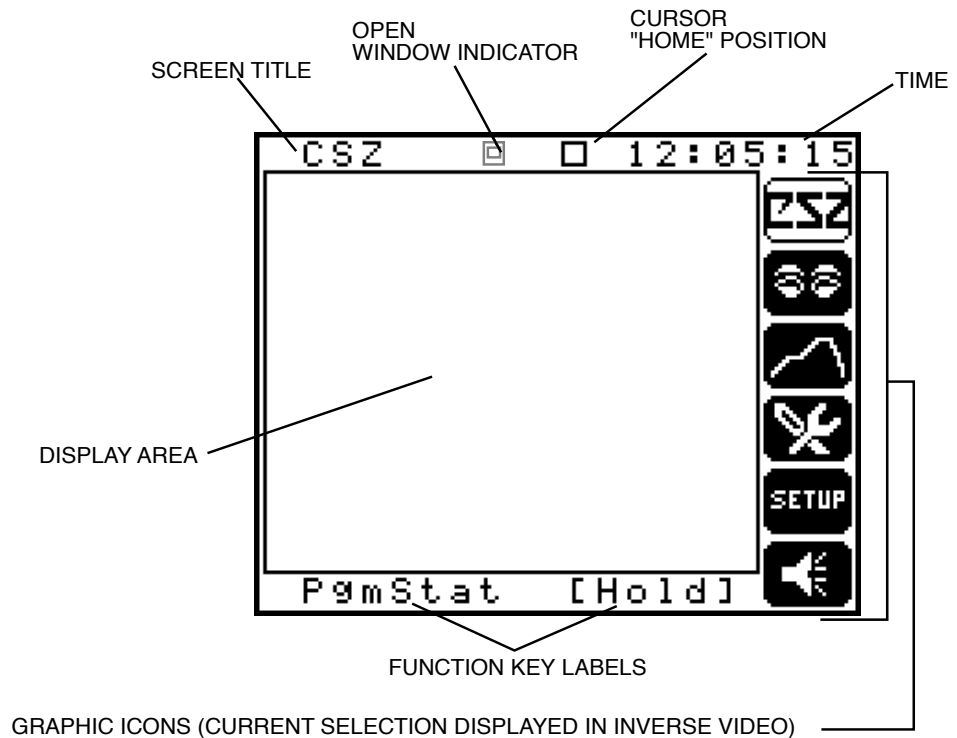
- Time of Day
- Date
- Contrast
- Backlight
- UNIT_CNF glossary name parameter
- Mnemonic (linear range characters for process variable units)
- EngUnits (Centigrade, Fahrenheit, Linear): The engineering units are not changed by memory card functions unless the "Input type" parameter selected in the Dimension is different than stored on the memory card. When the memory card restores the Input type to a Thermocouple or RTD, the EngUnits are set to Centigrade. When the memory card restores the input type to Current or Voltage ranges, the EngUnits are set to Linear.

Screen Display Format

As shown in Fig. 1-3, every Dimension screen display includes the following common elements:

Operator Interface Description

Figure 1-3.
Dimension Screen
Display Format



SCREEN TITLE—The title of the currently selected screen is displayed on the top row of the screen. (Most screen titles are user-definable.)

CURSOR "HOME" POSITION—A blinking graphic "box" displayed on the top row of the screen indicates that the cursor (controlled by turning the knob) is located in its "home" (inactive) position. Whenever you call up a new screen, the cursor is displayed in its home position.

OPEN WINDOW INDICATOR—A second, smaller box displayed inside another box to the left of the cursor "home" position indicates a window has been selected and is open in the display area. When the Open Window Indicator is active, the cursor home position indicator is not displayed (see Fig. 1-4).

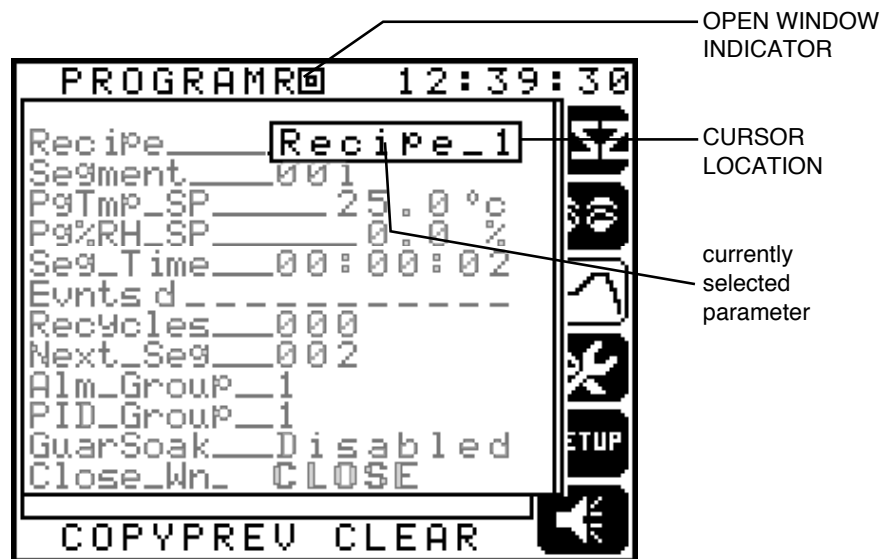
CURRENT TIME—The current time is displayed in the upper right-hand corner. Current time is displayed in standard 24-hour (HH:MM:SS) format.

GRAPHIC ICONS—The right-hand side of the screen displays six graphic icons, or symbols, which represent various system functions. The currently selected icon (selectable with diamond key) is displayed in inverse video.

FUNCTION KEY LABELS—The bottom row of the screen always displays user-definable labels for the two function keys located directly below the screen.

DISPLAY AREA—The contents of the current screen are visible in the “display area” shown in Fig. 1-3. The location of the cursor (when away from the “home” position) is depicted by a graphic box or rectangle surrounding the currently selected data entry field or selection, as shown in Fig. 1-4.

Figure 1-4.
Cursor Location, in This Case in an Open Window. Note Open Window Indicator.



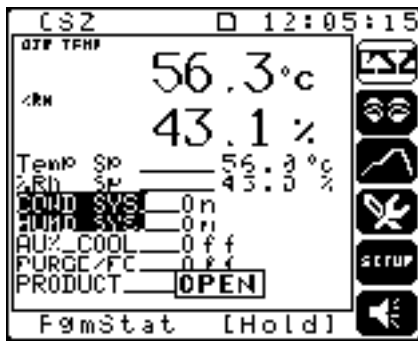
Display Summary







The Graphic Operator Interface provides all of the tools needed for Operation and the Configuration of a Dimension Industrial Process Manager. Operation can be described as the Dimension doing the work it was purchased to do, and Configuration as the process of setting up the unit to know how to do the work.

The Operation displays are divided into the icon/function groups described below and in Section 2 of this guide.

Operator Interface Description

Figure 1-5.
Operation Icons



- 
Unit Operation Displays current chamber air temp and relative humidity and their setpoints. Allows access to the chamber control functions and the product window.
- 
Overview Designed to be configured on a process basis to Displays show process parameters. Displays may be text, bar graph or xy graph format. Changing of process parameters may also be accomplished.
- 
Programmer Used to monitor the status of the setpoint/event Displays programmer and to create/edit program recipe segments.
- 
Utilities Used to monitor specific chamber functions and parameters.
- 
Setup Displays Used to monitor the status of control loops. Allows access to date/time entry, memory card functions and configuration access
- 
Alarm Displays Used to monitor the status of, acknowledge and set limits for loop and input alarms.

Access to the Configuration displays is accomplished by selecting the "setup" icon, then "entering" on the "unit" selection. Next select the Config parameter shown on the UNIT display. This will bring up the UNIT_CNF screen shown in Figure 1-6. From this screen, move up or down to access the desired configuration function. Configuration displays are divided into the areas described in Figure 1-6 and in Section 4, *Configuration*.

Figure 1-6.
Configuration Icons



Unit

Used to set access levels, define function key Configuration functions, enable/disable display groups and Display view/alter unit options.



Tagname

Used to modify the default factory terminology to Displays match that of your process and application.



Patch I/O

Used to assign input, output and function parameters to configure the Dimension to your application.



Display

Used to format and define operation displays to Configuration match your user, application and process needs.



I/O Calibration

Used to define input and output range and type information. Access to the calibration functions is also provided. Note that all Dimension inputs and outputs types are factory calibrated.



Communication

Used to set up parameters for the RS-232, RS-422 Displays and IEEE-488 communication ports.

A dark gray rectangular box with a thin black border. Inside the box, the text "Section 2" is written in a white, sans-serif font. The word "Section" is in a smaller font size than the large number "2".

Operation

This section describes the operation parameters available on the Dimension controller that are used to control the CSZ chamber.

This section assumes that the user is familiar with the conventions and operation of the Dimension Custom Logic controller. Refer to Section 1, *Operator Interface Description*, of this manual for basic operational information.

Fig. 2-1 and Fig. 2-2 are screen maps that illustrate the various Operations and Configuration screens of the Dimension.

Operation

Figure 2-1.
Map of Dimension Operations Screens

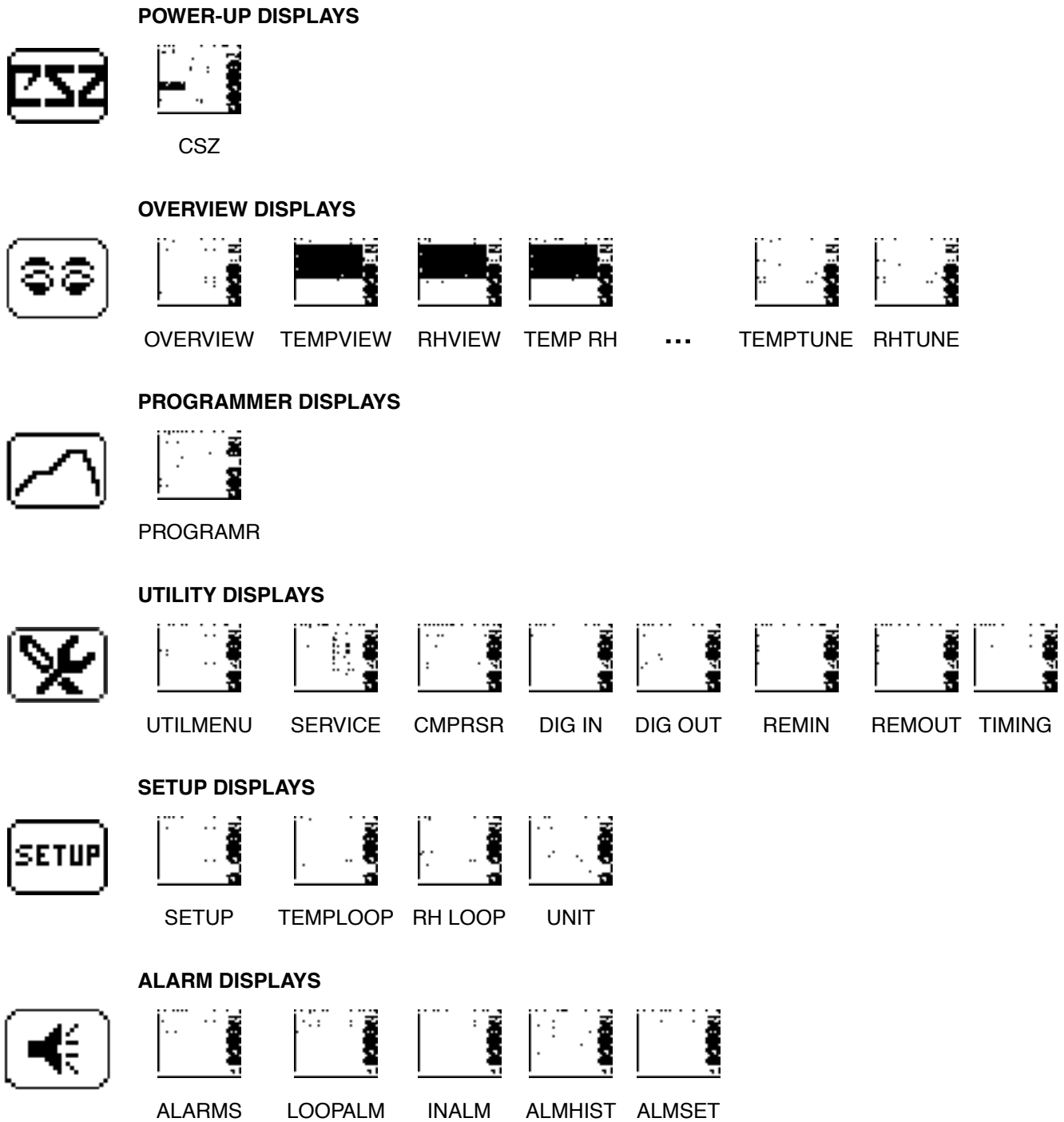
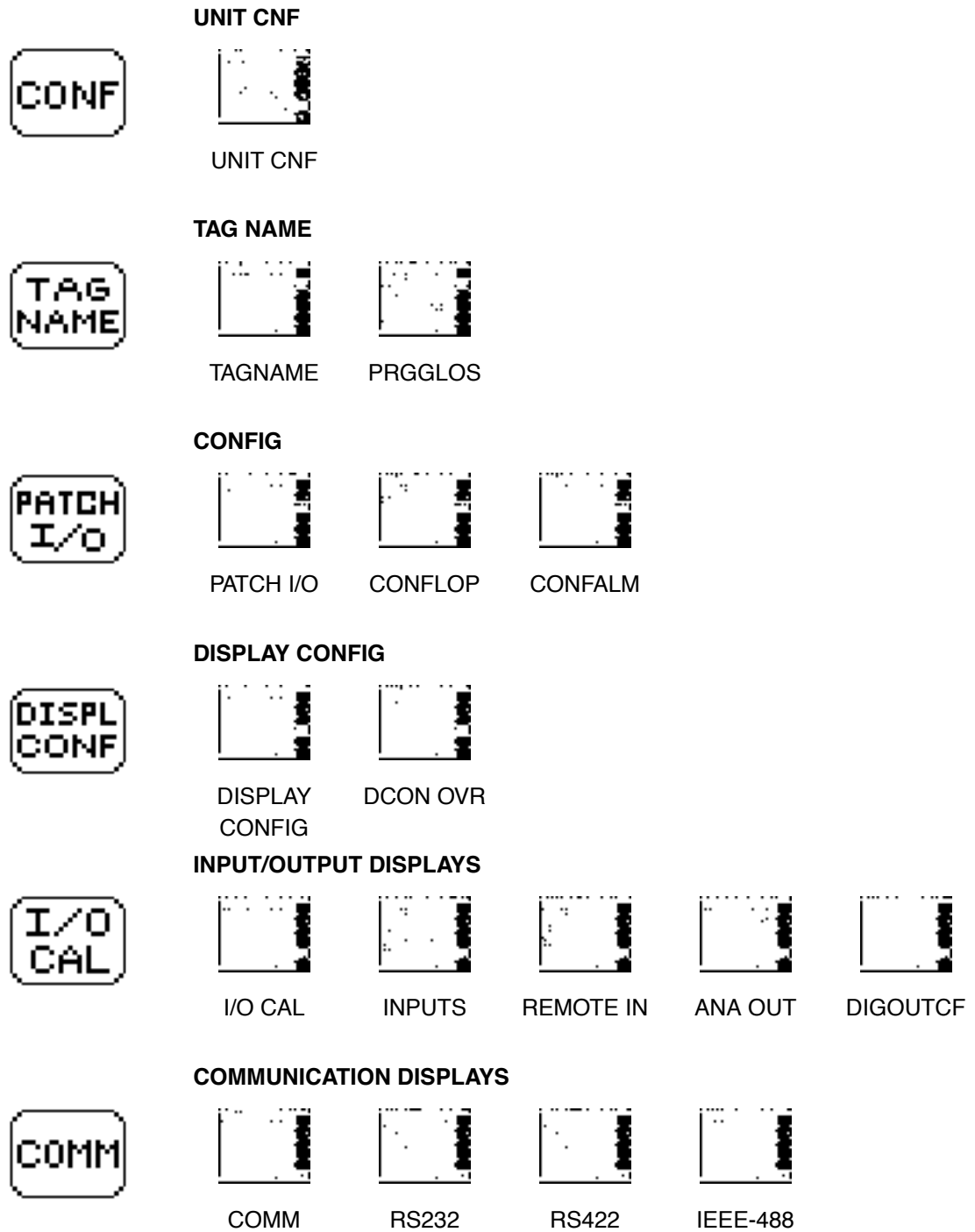


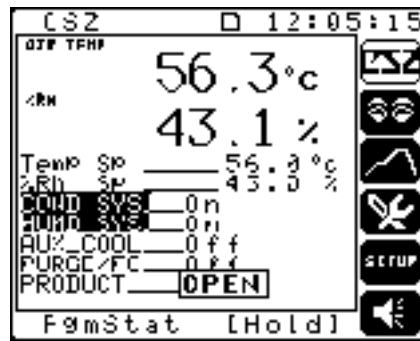
Figure 2-2.
Map of Dimension Configuration Screens



Operation

When the Dimension is powered up, the CSZ screen (Fig. 2-3) is the first screen displayed. This display contains the following parameters:

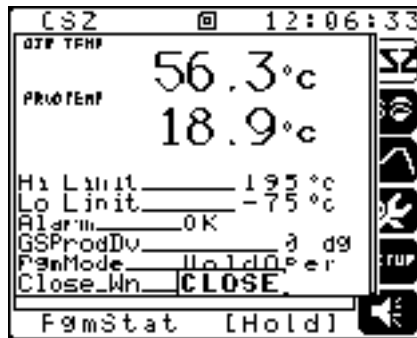
Figure 2-3.
CSZ Display



- AIR TEMP - Displays the current value of the chamber air temperature.
- %RH - Displays the current value of the relative humidity.
- Temp Sp - Set and display the current temperature setpoint that the chamber will try to control to when in the Automatic mode. This setpoint can be entered by the operator when the programmer is in a HoldOper condition or at program end. If the programmer is running, the setpoint will be controlled by the programmer.
- %Rh Sp - Set and display the current relative humidity setpoint the that the chamber will try to control to when in the Automatic mode. This setpoint can be entered by the operator when the programmer is in a HoldOper condition or at program end. If the programmer is running, the setpoint will be controlled by the programmer.
- COND SYS - Turns On and Off the heating and cooling functions. The following 3 modes are available:
 - On - Turns the conditioning system on and displays the COND SYS tagname in inverse video.
 - Off - Turns the conditioning system off and displays the COND SYS tagname in normal video.
 - Pgm - Allows the first event in the programmer (Event d) to control the status of the conditioning system and displays the COND SYS tagname in inverse video.
- HUMD SYS - Turns On and Off the relative humidity functions. The following 3 modes are available:
 - On - Turns the relative humidity system on and displays the HUMD SYS tagname in inverse video.
 - Off - Turns the relative humidity system off and displays the HUMD SYS tagname in normal video.

- Pgm - Allows the second event in the programmer (Event e) to control the status of the relative humidity system and displays the HUMD SYS tagname in inverse video.
- AUX_COOL - Turns On and Off the optional CO2 and LN2 boost cooling functions. The following 3 modes are available:
 - On - Turns the auxiliary cooling system on and displays the AUX_COOL tagname in inverse video.
 - Off - Turns the auxiliary cooling system off and displays the AUX_COOL tagname in normal video.
 - Pgm - Allows the third event in the programmer (Event f) to control the status of the auxiliary cooling system and displays the AUX_COOL tagname in inverse video.
- PURGE/FC - Turns On and Off the optional GN2 purge functions. The following 3 modes are available:
 - On - Turns the purge on and displays the PURGE/FC tagname in inverse video.
 - Off - Turns the purge off and displays the PURGE/FC tagname in normal video.
 - Pgm - Allows the fourth event in the programmer (Event g) to control the status of the purge and displays the PURGE/FC tagname in inverse video.
- PRODUCT - Opens a window (Fig. 2-4) that allows the setting of product High and Low limits along with the product guaranteed soak deviation value. This display contains the following parameters:

Figure 2-4.
Product Display

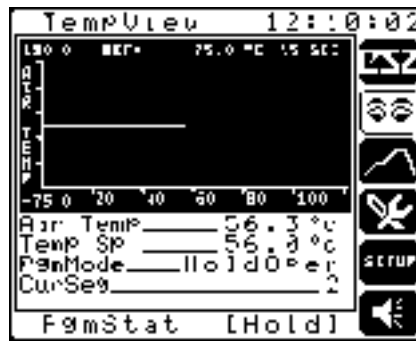


- AIR TEMP - Displays the current value of the chamber air temperature.
- PROTEMP - Displays the current value of the product temperature.
- Hi Limit - Set and display the value at which the product limit high temperature alarm will shut down the chamber. A critical alarm message will appear.

- OVIEW__4 - OVIEW_10 - User configurable overview screens.
- TempTune - Contains the active parameters to be used when tuning the temperature control loop.
- RH Tune - Contains the active parameters to be used when tuning the humidity control loop.

When TempView is selected from the Overview display, the screen shown in Fig. 2-6 is displayed. This screen contains the following parameters:

Figure 2-6.
TempView Display

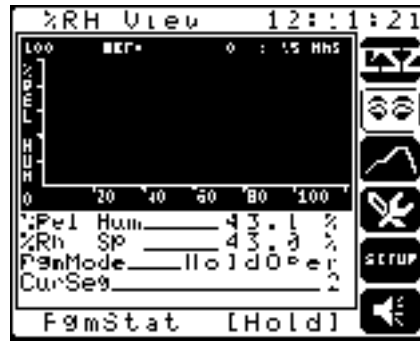


- Air Temp - Displays the current air temperature of the chamber.
- Temp Sp - Displays the current temperature setpoint to which the chamber will try to control to when in Auto mode. This setpoint can be entered by the operator when the programmer is in a HoldOper condition or at program end. If the programmer is running, the setpoint will be controlled by the programmer.
- PgmMode - Set and display the current mode of the programmer. The mode will be one of the following values:
 - Run - The programmer is currently executing its sequence of segments.
 - HoldOper - Execution of the programmer has been stopped due to operator intervention.
 - HoldSoak - Execution of the programmer has been stopped due to a guaranteed soak condition (Display Only).
 - PG_END - The programmer has completed executing its segments. The next segment must be set equal to the current segment for the programmer to stop (Display Only).
- CurSeg - Displays the current segment being executed by the programmer.

Operation

When %RH View is selected from the Overview display, the screen shown in Fig. 2-7 is displayed. This screen contains the following parameters:

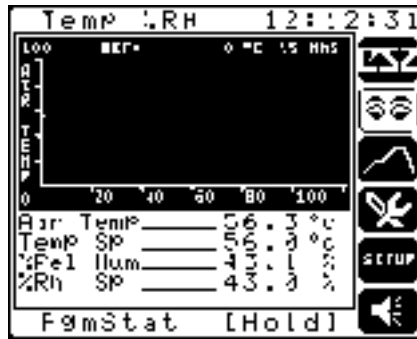
Figure 2-7.
%RH View Display



- %Rel Hum - Displays the current relative humidity for the chamber.
- %Rh Sp - Displays the current humidity setpoint to which the chamber will try to control to when in Auto mode. This setpoint can be entered by the operator when the programmer is in a HoldOper condition or at program end. If the programmer is running, the setpoint will be controlled by the programmer.
- PgmMode - Set and display the current mode of the programmer. The mode will be one of the following values:
 - Run - The programmer is currently executing its sequence of segments.
 - HoldOper - Execution of the programmer has been stopped due to operator intervention.
 - HoldSoak - Execution of the programmer has been stopped due to a guaranteed soak condition (Display Only).
 - PG_END - The programmer has completed executing its segments. The next segment must be set equal to the current segment for the programmer to stop (Display Only).
- CurSeg - Displays the current segment being executed by the programmer.

When Temp %RH is selected from the Overview display, the screen shown in Fig. 2-8 is displayed. This screen contains the following parameters:

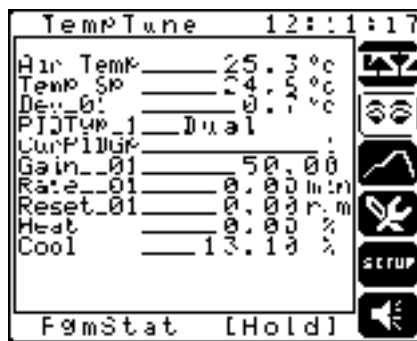
Figure 2-8.
Temp %RH Display



- Air Temp - Displays the current air temperature of the chamber.
- Temp Sp - Displays the current temperature setpoint to which the chamber will try to control to when in Auto mode. This setpoint can be entered by the operator when the programmer is in a HoldOper condition or at program end. If the programmer is running, the setpoint will be controlled by the programmer.
- %Rel Hum - Displays the current relative humidity for the chamber.
- %Rh Sp - Displays the current humidity setpoint to which the chamber will try to control to when in Auto mode. This setpoint can be entered by the operator when the programmer is in a HoldOper condition or at program end. If the programmer is running, the setpoint will be controlled by the programmer.

When TempTune is selected from the Overview display, the screen shown in Fig. 2-9 is displayed. This screen contains the following parameters:

Figure 2-9.
TempTune Display



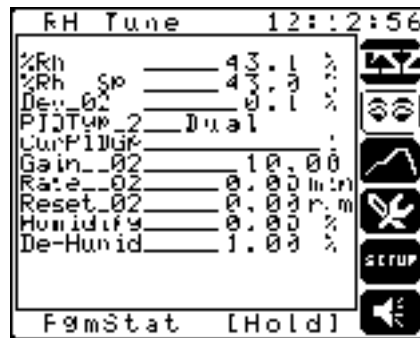
- Air Temp - Displays the current air temperature of the chamber.
- Temp Sp - Displays the current temperature setpoint to which the chamber will try to control to when in Auto mode.

Operation

- Dev_01 - Displays the difference between the actual air temperature and the temperature setpoint.
- PIDTyp_1 - Displays the current PID type. When in Dual, PID group 1 is for the heat output and PID group 2 is for the cool output.
- CurPIDGp - Displays the active PID group.
- Gain__01 - Set and display the PID gain value for the currently accessed PID group.
- Rate__01 - Set and display the PID rate value for the currently accessed PID group.
- Reset_01 - Set and display the PID rate value for the currently accessed PID group.
- Heat - Displays the calculated heating output.
- Cool - Displays the calculated cooling output.

When RH Tune is selected from the Overview display, the screen shown in Fig. 2-10 is displayed. This screen contains the following parameters:

Figure 2-10.
RH Tune Display

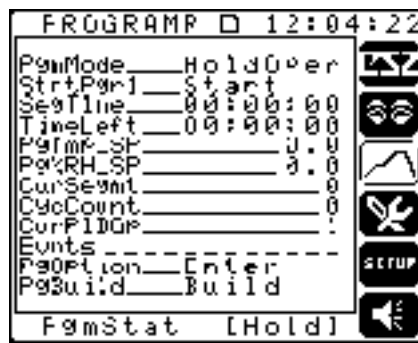


- %Rh - Displays the current relative humidity of the chamber.
- %Rh Sp - Displays the current humidity setpoint to which the chamber will try to control to when in Auto mode.
- Dev_02 - Displays the difference between the actual humidity and the humidity setpoint.
- PIDTyp_2 - Displays the current PID type. When in Dual, PID group 1 is for the humidify output and PID group 2 is for the de-humidify output.
- CurPIDGp - Displays the active PID group.
- Gain__02 - Set and display the PID gain value for the currently accessed PID group.
- Rate__02 - Set and display the PID rate value for the currently accessed PID group.
- Reset_02 - Set and display the PID rate value for the currently accessed PID group.

- Humidify - Displays the calculated humidify output.
- De-Humid - Displays the calculated de-humidify output.

The programmer icon is used to build, modify and monitor a real time program. For a more detailed description of the programming functions, refer to Section 4, *Configuration*, in this manual. This display contains the following parameters:

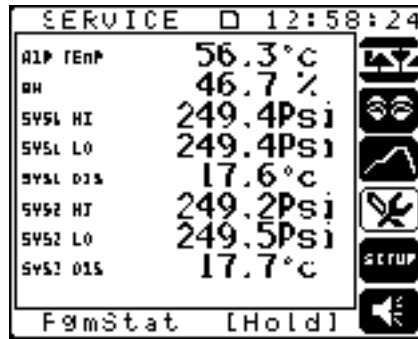
Figure 2-11.
Programmer Display



- PgmMode - Set and display the current status of the programmer. These functions may also be done by pressing the first Macro key "PgmStat". The mode will be one of the following values:
 - Run - The programmer is currently executing its sequence of segments.
 - HoldOper - Execution of the programmer has been stopped due to operator intervention.
 - HoldSoak - Execution of the programmer has been stopped due to a guaranteed soak condition (Display Only).
 - PG_END - The programmer has completed executing its segments. The next segment must be set equal to the current segment for the programmer to stop (Display Only).
- StrtPgr1 - Opens a window to select which Recipe to run, to start and stop the current recipe selected and to display the status of the current recipe.
- SegTime - Displays the time remaining in the currently active segment.
- TimeLeft - Display the total time remaining in the currently active recipe.
- PgTmp_Sp - Displays the current temperature setpoint that the chamber will try to achieve during the execution of the segment.
- Pg%RH_SP - Displays the current humidity setpoint that the chamber will try to achieve during the execution of the segment.
- CurSegmt - Displays the number of the current segment being executed.

When SERVICE is selected from the UTILITY display, the screen shown in Fig. 2-13 will be displayed. This screen has the following parameters:

Figure 2-13.
Service Display

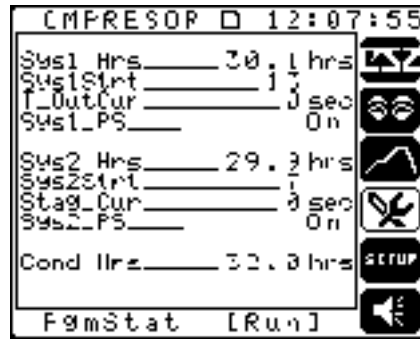


- AIR TEMP - Displays the current value of the chamber air temperature.
- RH - Displays the current relative humidity of the chamber.
- SYS1 HI - Displays the high pressure reading taken from the transducer located on the discharge side of the compressor. A backup mechanical pressure switch is also installed for the protection of the compressor.
- SYS1 LO - Displays the low pressure reading taken from the transducer located on the suction side of the compressor.
- SYS1 DIS - Displays the discharge line temperature reading taken from the thermocouple located on the discharge line of the compressor.
- SYS2 HI - Displays the high pressure reading taken from the transducer located on the discharge side of the compressor. A backup mechanical pressure switch is also installed for the protection of the compressor.
- SYS2 LO - Displays the low pressure reading taken from the transducer located on the suction side of the compressor.
- SYS2 DIS - Displays the discharge line temperature reading taken from the thermocouple located on the discharge line of the compressor.

Operation

When CMPRESOR is selected from the UTILITY display, the screen shown in Fig. 2-14 will be displayed. This screen has the following parameters:

Figure 2-14.
Cmpresor Display



- Sys1 Hrs - Displays the total number of hours the Stage 1 compressor has been running.
- Sys1Strt - Displays the number of times the Stage 1 compressor has been started.
- T_OutCur - Displays the time remaining before the compressors will be shut down.
- Sys1_PS - Displays the status of the oil pressure safety switch.
- Sys2 Hrs - Displays the total number of hours the Stage 2 compressor has been running.
- Sys2Strt - Displays the number of times the Stage 2 compressor has been started.
- Stag_Cur - Displays the time remaining before the Stage 2 compressor will be started.
- Sys2_PS - Displays the status of the oil pressure safety switch.
- Cond Hrs - Displays the total number of hours that the air circulators (conditioning system) have been running on the chamber.

When DIG IN is selected from the UTILITY display, the screen shown in Fig. 2-15 will be displayed. This screen has the following parameters:

Figure 2-15.
Dig In Display



- ExtLimit - Displays the status of the external high/low limit control. If the input is On, the limit control is in a good state.
- Sys1_PS - Displays the status of the system 1 oil pressure safety switch.
- Sys2_PS - Displays the status of the system 2 oil pressure safety switch.

When DIG OUT is selected from the UTILITY display, the screen shown in Fig. 2-16 will be displayed. This screen has the following parameters:

Figure 2-16.
Dig Out Display

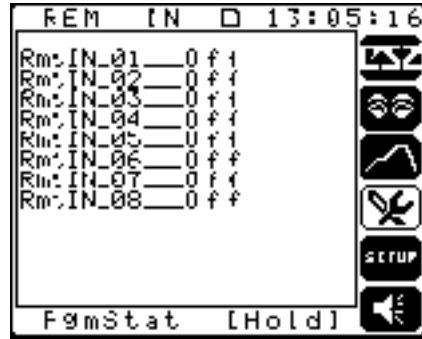


- Fans - Displays the status of the digital output that controls the air circulator fans.
- Compr #1 - Displays the status of the digital output that controls the Stage 1 compressor.
- Heater - Displays the status of the digital output that controls the heating control relay.
- Cooling - Displays the status of the digital output that controls the cooling solenoid.
- By-Pass - Displays the status of the digital output that controls the by-pass solenoid.
- Compr #2 - Displays the status of the digital output that controls the Stage 2 compressor.
- Hum Comp - Displays the status of the digital output that controls the humidify air compressor.
- Humidify - Displays the status of the digital output that controls the humidity solenoid.
- De-Humid - Displays the status of the digital output that controls the de-humidify wet coil solenoid.
- RH Cool - Displays the status of the digital output that controls the cooling solenoid (RH Mode).
- Max Heat - Indicated whether the chamber is calling for maximum heat (used only with a dual stage heating system).
- Horn - Displays the status of the sonalert audible alarm output.

Operation

When REM IN is selected from the UTILITY display, the screen shown in Fig. 2-17 will be displayed. This screen has the following parameters:

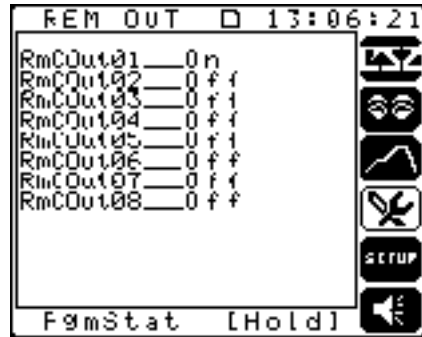
Figure 2-17.
REM IN Display



- RmtIN_01 - RmtIN_08 - Displays the status of the 8 optional remote digital inputs.

When REM OUT is selected from the UTILITY display, the screen shown in Fig. 2-18 will be displayed. This screen has the following parameters:

Figure 2-18.
REM OUT Display



- RmCOut01 - RmCOut08 - Displays the status of the 8 optional remote digital outputs. These outputs are tied to Events 1 - 8 of the programmer.

When ANALG IN is selected from the UTILITY display, the screen shown in Fig. 2-19 will be displayed. This screen has the following parameters:

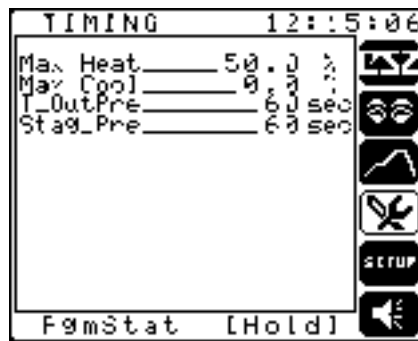
Figure 2-19.
ANALG IN Display



- Air Temp - Displays the air temperature in the chamber from a thermocouple located in the air stream at the discharge of the plenum.
- %Rel Hum - Displays the percentage of relative humidity reading from a sensor located in the air stream at the discharge of the plenum.
- ProdTemp - Displays the current product temperature reading from a thermocouple that is located inside the chamber and dedicated for the customer's part.
- Hi Limit - Displays the chamber air temperature reading from a thermocouple located above the heating element behind the plenum. This is a redundant sensor for chamber protection only.
- Unused 6 - An unused analog input.

When TIMING is selected from the UTILITY display, the screen shown in Fig. 2-20 will be displayed. This screen has the following parameters:

Figure 2-20.
TIMING Display



- Max Heat - Set the staged heating output value to cycle on max heat. When set to 0%, the max and min cycle together.
- Max Cool - Set the staged cooling output value to cycle on max cool. When set to 0%, the max and min cycle together.
- T_OutPre - The time-out timer will shut down the compressors when cooling has not been used for the set number of seconds. The compressors will not shut down while the humidity system is on. The range is 30-180 seconds.
- Stag_Pre - Stagger start timer that starts the Stage 2 compressor the set number of seconds after the Stage 1 compressor starts. The range is 30-120 seconds.

Process Control

Control Loop Definition

This topic is intended to provide a common base of terminology and control loop theory to help you configure and operate a control loop using the CSZ Dimension Custom Logic Controller. A brief discussion of the theory of process control is provided here, using the control of a home furnace as an example.

Process Control

The fundamentals of process control affect your daily life in many ways.

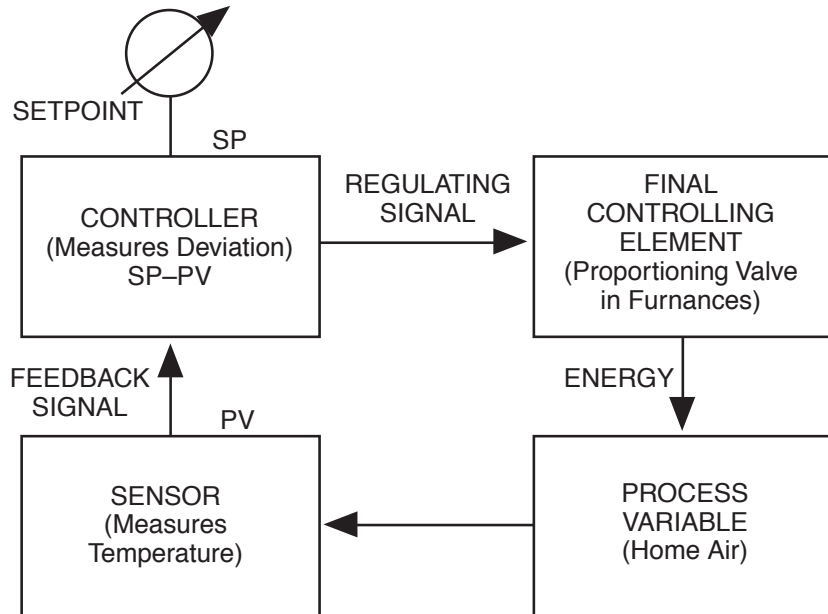
For example, the furnace that heats your home operates under the principles of process control. You control the temperature of the air in your home by choosing a setting on the thermostat, which in turn controls the furnace. In process control, a setting is referred to as a "setpoint." A setpoint is defined as "the desired value of a process variable." In this example, the process variable is the air temperature.

By establishing a desired air temperature setting for your home (setpoint), you expect that gas or oil will be supplied to the furnace to heat the air. The furnace is turned on when the temperature falls below the setpoint, and turned off when the temperature rises above the setpoint. This type of control action is often referred to as *two position* or "on/off" control.

In more sophisticated thermostats, the amount of fuel delivered to the furnace is regulated according to the number of degrees the actual air temperature differs from the desired temperature. This difference is the "error" or "deviation." The deviation is detected by the thermostat, which controls the amount of heat delivered by the furnace by controlling the fuel valve feeding the burner. The valve in turn opens proportionately. If the setpoint is reached, the valve receives a small signal or no signal at all. It closes and very little heat, if any, is generated.

The device that senses the air temperature in your home (sensor) sends a signal (feedback signal) to the temperature controller (thermostat). The signal portrays the actual air temperature (Process Variable, or PV). The thermostat compares the signal received with the setpoint (SP). A corresponding action (valve opens or closes) occurs depending on the magnitude of the deviation (SP-PV). This process is called a "closed loop" process because there is a feedback signal and the controller adjusts the output automatically (Fig. 2-21).

Figure 2-21.
Closed Loop Control



Terminology

Following are terms used in this Manual and by Dimension when Configuring or Operating a control loop.

- **Alarm** - A function that provides an indication (display or signal) if a specified limit is exceeded. Dimension provides 2 levels of High/Low process variable and deviation alarms, high/low analog input alarms, and Open sensor alarms for thermocouples.
- **Analog Input (AI)** - Refers to a sensor such as a thermocouple, RTD or pressure sensor that is connected to the Dimension. This input is used by Dimension for monitoring, control and display functions. This parameter is referred to as `AnIglIn` on the Dimension.
- **Configuration** - This is the process of selecting Inputs, Outputs, Limits, Constants, Enabling/Disabling features.
- **Control Action** - The control action of a loop defines what type of **control output** the loop provides. Choices are REVERSE, DIRECT, and BIMODAL and can be defined as follows:
 - Reverse output is selected when the process requires an increasing output signal when the *setpoint* is **greater** than the *process variable*. A typical example would be a heating output control signal.
 - Direct output is selected when a process requires an increasing output signal when the *setpoint* is **less** than the *process variable*. A typical example would be a cooling output control signal.

- **Bimodal** output is selected when a process requires both a *reverse* or a *direct* output. An example of this would be an environmental test chamber that has both heating and cooling capabilities.
- **Control Output** - The Control Output is the output signal from the controller that is used to control an element (valve, heater) that affects the process. The source of this output can be either the operator or the calculated PID (see next page) output.
- **Deviation** -The deviation is the difference between the desired process variable (setpoint) and the actual process variable. This value is used with alarms and in the PID output calculation.
- **Loop Mode** - The Loop Mode may be either **Automatic** or **Manual**. In **Automatic** (closed loop) mode the control loop calculates a control output based on the current PID constants and the current deviation from setpoint. In **Manual** (open loop) mode, the Operator is responsible for adjusting the control output signal.
- **Operator** - Refers to the user of the Dimension making a change to cause something within the instrument to happen. Examples are changing Loop Mode from Auto to Manual, Setpoint Source from Operator to Programmer.
- **PID - Proportional, Integral and Derivative control.** The term PID control is used to describe the calculation that is done using the proportional term (gain), the Integral term (reset), the Derivative term (rate), the Setpoint, the Process Variable and the Span of the control loop. The result of the calculation is the **control output**. It is the user's responsibility to select the proper PID constants (gain, reset, and rate) based upon the application that is being controlled. Selecting these constants is referred to as TUNING the control loop (see "PID Loop Tuning" on page 2-27).
- **Gain** - Gain is a multiplication term expressing how the control loop output varies with respect to the control loop deviation (process variable - setpoint). The effect of gain on the control loop output can be calculated from the following equation:

$$\text{Output}\% = \text{Deviation} / \text{Input Span} \times \text{Gain}.$$
- **Reset** -The reset factor is used to correct for process losses. Reset adds or subtracts from the control loop output. The effect of Reset can be calculated by the following equation:

$$\text{Output}\% = \sum \text{Reset} \times \text{Elapsed Time in minutes} \times \text{Deviation} / \text{Inputs span} \times 100.$$
- **Rate** - The rate factor is used to anticipate and correct for the process lag and adjust the output to avoid overshoot. The effect of Rate on the control loop output can be calculated by the following formula:

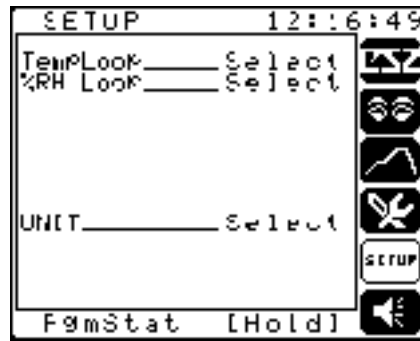
$$\text{OutPut}\% = \frac{\text{Rate} \times \text{Current} - \text{PV \% Last}}{\text{Minutes between PV Current and PV last}}$$

- **PID Type** - PID Type defines which group of PID settings is used in a process. There are up to 5 different PID groups that may contain different values for gain, reset, and rate. Single, Dual, Programmed, or Select Adaptive types are available.
 - Single PID is used when one set of PID constants is to be used at all times. Group 1 is always used if single is selected.
 - Dual PID is used when one set of PID constants is to be used for Reverse output (group #1) and another set is to be used for Direct output (group #2).
 - Programmed PID is used when the real-time programmer (see Section 4, *Configuration*) is used to select which PID group is to be used on each segment.
 - Select Adaptive PID is used when you wish to alter PID constants based upon the value of the process variable.
- **Real-Time Programmer** - A function within Dimension to generate a Setpoint or event that can vary with time. The programmer can be used to ramp a setpoint from one value to another or turn an event on or off over a user-specified time segment. Multiple time segments/setpoints/events are used to make up a real-time program.
- **Setpoint (SP)** - Setpoint is defined as the *desired value* for the PV. This is the value the Dimension tries to achieve when in automatic control.
- **Setpoint Source** - Defines what provides the setpoint to a control loop. In a Dimension this may be the *Operator*, a *Real-Time Programmer* or a *Remote Source*. A remote source may be an external device (such as another controller) or another control loop within the Dimension.
- **Span** - The difference between the upper and lower range values of the process variable.
- **Tagname** - A descriptive abbreviation used to label a function or variable in the Dimension. It is typically a single word that is made up from one or more words to describe something. An example is ProcVr for Process Variable. In the Dimension most tagnames may be altered by the user to fit the process to be controlled or monitored. An example of this would be to modify ProcVar to be AirTemp to describe Chamber Temperature.
- **Thermocouple** - A temperature sensor that produces a millivolt output signal that is proportional to the temperature. This input voltage is then linearized by Dimension to provide a temperature reading (process variable) in degrees.
- **Time Proportioned Output** - A control output that provides periodic on/off pulses. The proportion of *on time* to *off time* is determined by the control output percent.
- **Tuning Parameters** - Refers to the PID constants of Gain, Reset and Rate that are used to calculate the process control output signal.

PROCESS CONTROL LOOPS

The setup icon is used to access the process control loop and the unit displays. When this icon is selected the display shown in Fig. 2-22 will be displayed. From this display contains the following parameters:

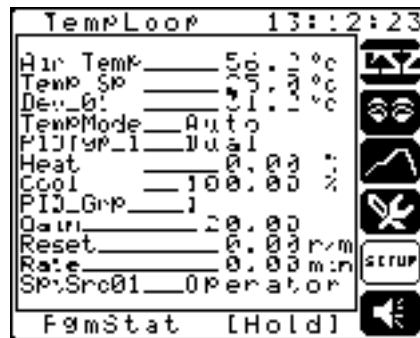
Figure 2-22.
SETUP Display



- TempLoop - Allows access to the temperature loop screen.
- %RH Loop - Allows access to the relative humidity loop screen.
- UNIT - Allows access to Dimension global parameters including setting Time-of-Day, date, memory card functions and access to the configuration side of the Dimension.

When TempLoop is selected from the SETUP display, the screen shown in Fig. 2-23 will be displayed. This screen has the following parameters:

Figure 2-23.
TempLoop Display



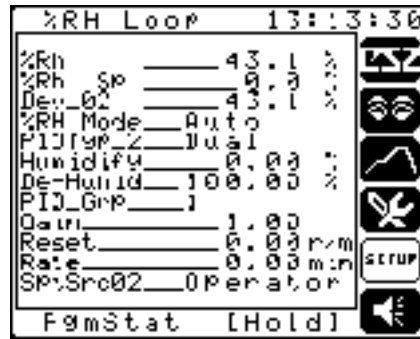
- Air Temp - Displays the current air temperature of the chamber. This value is calculated and may not be accessed by the operator.
- Temp Sp - Displays the desired value for the air temperature. This is the value at which you want the chamber to control. This value can be entered by the operator when the programmer is in a HoldOper condition or at program end. If the programmer is running, the value will be controlled by the programmer.

- Dev_01 - Displays the difference between the actual temperature and the desired temperature. This value is calculated and may not be accessed by the operator.
- TempMode - Displays the current mode of the temperature control loop. The default setting is Auto (closed loop). When on this screen only, the mode may be changed by the operator to Manual (open loop).
- PIDTyp_1 - Determines which group of PID settings are used for control of the chamber. There are up to 5 different PID groups that may contain different values for gain, reset, and rate. The type available are; Single, Dual, Programmed, or Select Adaptive (see Table 2-1 and Fig. 2-25).
- Heat - Displays the current heating output value for the control loop. If the TempMode is set to Manual, this parameter may be adjusted by the operator. If in the Auto mode, this parameter is a calculated value.
- Cool - Displays the current cooling output value for the control loop. If the TempMode is set to Manual, this parameter may be adjusted by the operator. If in the Auto mode, this parameter is a calculated value.
- PID_Grp - Allows the selection of the PID group that is currently being displayed in the Gain, Reset, and Rate parameters below it.
- Gain - Set and display the PID gain value for the currently accessed PID group.
- Reset - Set and display the PID reset value for the currently accessed PID group.
- Rate - Set and display the PID rate value for the currently accessed PID group.
- SptSrc01 - Displays the currently selected setpoint source for the control loop. This parameter may be altered to select either Operator, Programmer, or Remote. When the programmer is in the Run mode, the setpoint source will be set to Programmer automatically. When the programmer is in a hold condition or at program end the setpoint source will be changed to operator to allow for the operator to enter in a new setpoint if desired. Set the setpoint source to Remote to override the automatic changing of this parameter and input a remote signal for the setpoint. The actual source for Remote is set on the CONF_LOP screen (Refer to Section 4, *Configuration*).

Operation

When %RH Loop is selected from the SETUP display, the screen shown in Fig. 2-24 will be displayed. This screen has the following parameters:

Figure 2-24.
%RHLoop Display



- %Rh - Displays the current relative humidity of the chamber. This value is calculated and may not be accessed by the operator.
- %Rh Sp - Displays the desired value for the relative humidity. This is the value at which you want the chamber to control. This value can be entered by the operator when the programmer is in a HoldOper condition or at program end. If the programmer is running, the value will be controlled by the programmer.
- Dev_02 - Displays the difference between the actual humidity and the desired humidity. This value is calculated and may not be accessed by the operator.
- %RH Mode - Displays the current mode of the humidity control loop. The default setting is Auto (closed loop). When on this screen only, the mode may be changed by the operator to Manual (open loop).
- PIDTyp_2 - Determines which group of PID settings are used for control of the chamber. There are up to 5 different PID groups that may contain different values for gain, reset, and rate. The type available are; Single, Dual, Programmed, or Select Adaptive (see Table 2-1 and Fig. 2-25).
- Humidify - Displays the current humidify output value for the control loop. If the %RH Mode is set to Manual, this parameter may be adjusted by the operator. If in the Auto mode, this parameter is a calculated value.
- De-Humid - Displays the current de-humidify output value for the control loop. If the %RH Mode is set to Manual, this parameter may be adjusted by the operator. If in the Auto mode, this parameter is a calculated value.
- PID_Grp - Allows the selection of the PID group that is currently being displayed in the Gain, Reset, and Rate parameters below it.
- Gain - Set and display the PID gain value for the currently accessed PID group.

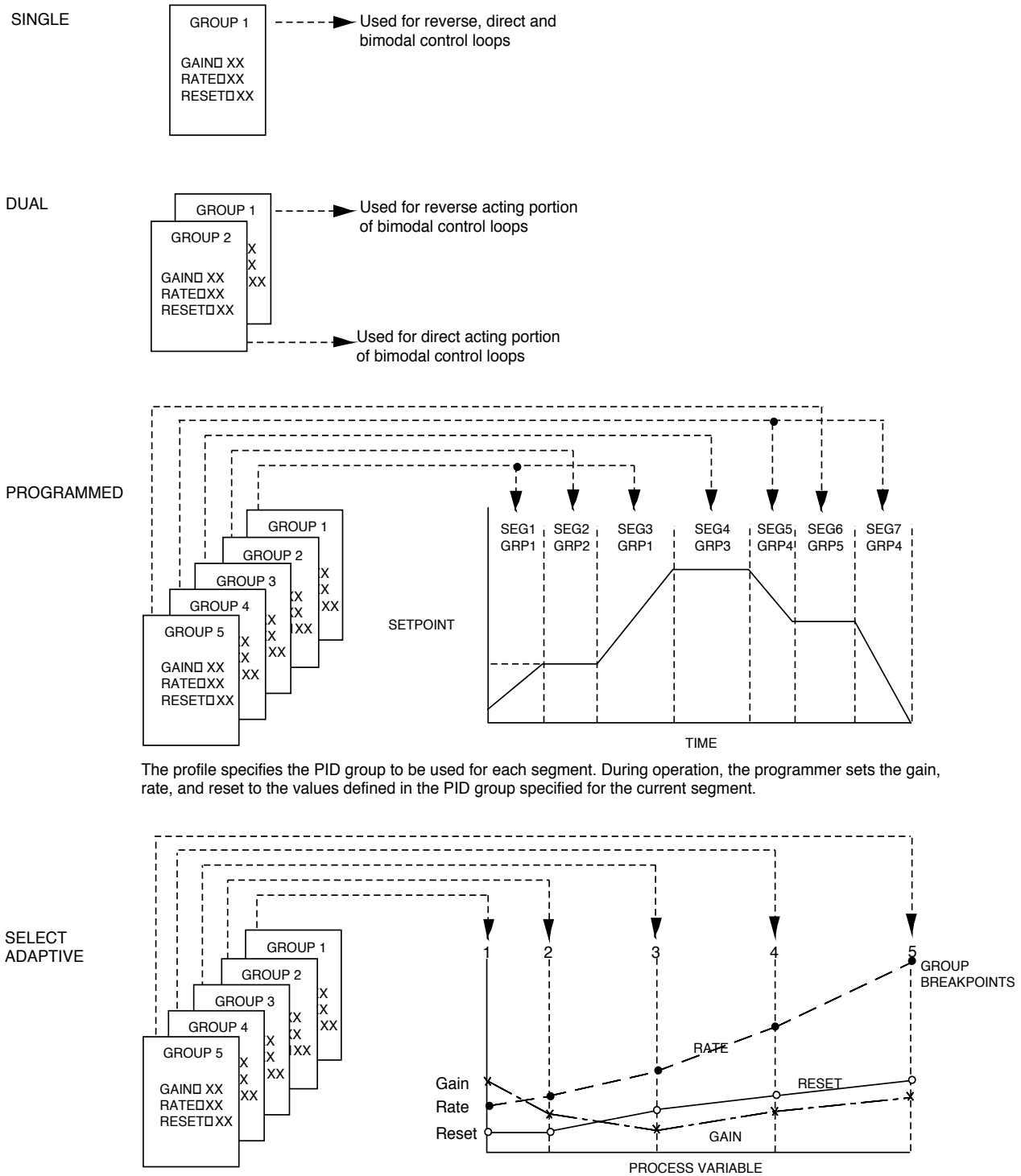
- Reset - Set and display the PID reset value for the currently accessed PID group.
- Rate - Set and display the PID rate value for the currently accessed PID group.
- SptSrc02 - Displays the currently selected setpoint source for the control loop. This parameter may be altered to select either Operator, Programmer, or Remote. When the programmer is in the Run mode, the setpoint source will be set to Programmer automatically. When the programmer is in a hold condition or at program end the setpoint source will be changed to operator to allow for the operator to enter in a new setpoint if desired. Set the setpoint source to Remote to override the automatic changing of this parameter and input a remote signal for the setpoint. The actual source for Remote is set on the CONF_LOP screen (Refer to Section 4, *Configuration*).

Table 2-1. Active PID Groups by Type Selected

PID Type	PID Groups				
	1	2	3	4	5
Single	X	Not Active			
Dual	X	X	Not Active		
Program	X	X	X	X	X
Select Adaptive	X	X	X	X	X
X = Active					

Operation

Figure 2-25.
PID_Type Group Selection



The profile specifies the PID group to be used for each segment. During operation, the programmer sets the gain, rate, and reset to the values defined in the PID group specified for the current segment.

Each PID group specifies the value of gain, rate, and reset at the associated break-point in the range of the process variable. During operation, the gain, rate, and reset are calculated by interpolating between the values at the breakpoints above and below the current values of the process variable.

PID Loop Tuning

The Dimension uses PID control. PID is an acronym meaning proportional, integral, and derivative. The calculations provided by the three functions are mathematically summed together to provide an output signal to control your process.

Before tuning your process, keep the following points in mind:

- The GAIN parameter adjusts the *proportional* PID term. Increasing the GAIN value will increase the output response to the process deviation.
- The RESET parameter adjusts the *integral* PID term. Increasing the RESET value will increase the output response to the process deviation.
- The RATE parameter adjusts the *derivative* PID term. Increasing the RATE parameter will increase the output response to a process variable change.
- Setting the GAIN, RESET, or RATE values to 0 will disable the proportional, integral, or derivative functions respectively.
- The derivative function is usually not used in applications where there is an immediate response to the process variable from the control output signal. The derivative function by nature will cause the output signal to appear unstable, as it responds to the changing process variable signal. Input filtering (FilterVal set greater than 0) is always required when using derivative. Using derivative may not be desirable in an application where the output controls a mechanical valve.
- The Dimension uses the difference (input span) between the input high limit (InptHiLim) and input low limit (InptLowLim) for the PID calculations. If setpoint limiting is desired, change the input high and low limit values before starting your loop tuning.

Should you have tuned your loop already, and are required to change the input high or low limit values, change your PID parameters as follows:

$$\frac{\text{New input high limit} - \text{New input low limit}}{\text{Old input high limit} - \text{Old input low limit}} \times \text{Old PID values} = \text{New PID values}$$

- Be sure the control action (CntrlAct) is set properly:
Set to Reverse if an increasing process variable should decrease the output (heating). Set to Direct if an increasing process variable should increase the output (cooling). Set to Bimodal if there will be 2 physical outputs for both reverse and direct control actions.
- When the PID type (PIDType) is set to Single, only PID group 1 is used. If PID type is set to Dual, PID group 1 is used for reverse control action and PID group 2 is used for direct control action.
- We do not advise to perform loop tuning while a real-time programmer is changing the setpoint. If the process is not controlling to a steady state condition, controlling to a changing setpoint will likely not occur.

Tuning the Loop

Proportional Adjustments (GAIN):

Gain ($\text{Gain} = \frac{1}{\text{PB}}$) adjusts the proportional control action by modifying the control loop's output in response to the amount of deviation from the setpoint. A low gain value changes the output gradually over a wide deviation band. A high gain value changes the output rapidly over a narrower deviation band. Gain is the inverse of proportional band, so control action with a low gain setting is like control action with a large proportional band.

When loop tuning from scratch, adjust the gain value first (reset and rate values to 0). Increase the gain until a small amount of oscillation of the process variable occurs. Lower the gain until the oscillation stops. Note that the oscillation may occur at a level lower than the setpoint (heating application).

Typical gain settings are from 15 - 60 in a temperature control application. Typical gain settings are 0.3 - 1.5 in applications such as pressure or mass flow control.

Integral Adjustments (RESET):

Reset adjusts the integral action by integrating the output in a positive or negative direction based on deviation. For example, in a heating application the output is increased over time if the temperature is low and the output is decreased over time if the temperature is high. The output is not changed if there is no deviation. In most control applications, proportional control alone cannot achieve zero deviation. For example, in a typical heating application, the process temperature stabilizes below the setpoint at a deviation value where the control loop provides just enough heat output to balance the heat losses from the process. This is called proportional droop. Reset action is used to eliminate this offset between the process variable and the setpoint.

Reset values are expressed in terms of repeats per minute. One repeat per minute is a measurement of integration speed of 100 percent output over 1 minute of time, *when the deviation is at maximum* (input high limit - input low limit). A higher value repeats per minute, the faster the output will integrate.

After the gain has been adjusted, adjust the reset. Increase the reset to eliminate the proportional droop achieving 0 deviation. Make small setpoint changes watching the response of the process. A small overshoot may be normal as long as oscillation of the process variable does not occur.

Too small of reset value will cause the process to have a sluggish response. Too high of reset value will cause the process variable to oscillate.

Typical reset settings are from 3 - 10 in a temperature control application. Typical reset settings are 5 - 15 in applications such as pressure or mass flow control.

Derivative Adjustments (RATE):

Rate adjusts the derivative action by changing the output in a positive or negative direction based on process variable change. Derivative opposes change of the process variable even if the process variable is under or over setpoint. For example, in a heating application the output is increased if the temperature is falling and the output is decreased if the temperature is rising. The output is not changed if the temperature is steady. Derivative action is used primarily in control applications where the process changes very slowly (where there are large time lags in response to changes in the controller output). Also a small amount of derivative may be used to eliminate overshooting of the process variable while still maintaining a good response time to setpoint changes.

Rate values are expressed in terms of minutes of duration. A rate of 1 describes a future process variable value in one minutes time based on the current process variable change. The future process variable value is used with proportional action to offset the output based on where the process variable will be one minute from now. The higher rate value, the higher the output will be offset.

After the reset has been adjusted, adjust the rate. Increase the rate to eliminate process overshooting. Make small setpoint changes watching the response of the process. Too high of rate value will cause the process variable to oscillate.

Typical rate settings are from .5 - 5 in a temperature control application.

Fine Tuning the Process

Here are some hints for fine tuning your process:

- Small adjustments may be made to any one parameter at a time giving enough time to allow for the process to stabilize after the change.
- Keep in mind the effect of the PID parameter values. For example, doubling the response of proportional action is done by doubling the gain from 1 to 2 or doubling the gain from 100 to 200.

- Watch the output percent simultaneously with watching the process variable. Observe the nominal output values required to maintain various process variable values (this may be done in manual mode also). The reset value may be adjusted higher or lower so the output percent is close to where it should be when the corresponding process variable is achieved.

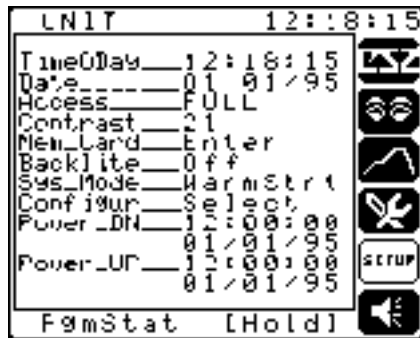
An approximate gain value for temperature control loops may be determined by this formula:

$$\text{GAIN} = .02 * ((\text{output at temperature 1} - \text{output at temperature 2}) / ((\text{temperature 1} - \text{temperature 2}) / (\text{Input high limit} - \text{Input low limit})))$$

Unit Setup

When UNIT is selected from the SETUP display, the screen shown in Fig. 2-26 will be displayed. This screen has the following parameters:

Figure 2-26.
UNIT Display

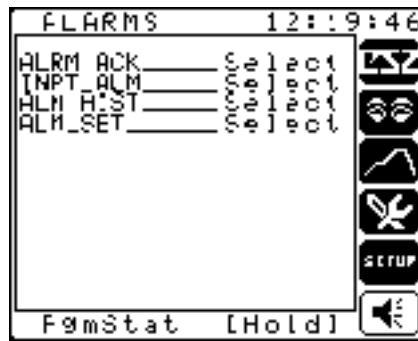


- TimeOfDay - Set and display the real time in standard 24-hour HH:MM:SS format.
- Date - Set and display the current date in standard MM/DD/YY format.
- Access - Displays current access level (FULL, LIMITED, NONE). A window will open to accept passcode entry.
- Contrast - Displays a value representing the current display contrast/brightness.
- Mem_Card - A window opens that permits the user to read from, write to, format, delete files or read the directory of a memory card. See "Memory Card Slot" in Section 1, *Operator Interface Description*.
- BackLite - When selected, a window opens in which the operator may select SET and then select ON (backlight always on), OFF (backlight always off), or CYCLE (backlight turns on and off at specified times). The ON and OFF fields allow the user to set the on/off cycle times.

- Sys|_Mode - Displays the system power-up mode.
 - Warmstrt - All current settings are saved. All chamber systems (conditioning system, etc.) are disabled. The programmer is set to a hold condition.
 - ColdStrt - Only I/O calibrations settings are saved. Factory defaults are restored into all other parameters.
 - AutoStrt - All current settings are saved.
 - H-Astart - All current settings are saved and the programmer (if running) is set to a hold condition.
- Configur - Allows the operator to switch between the Operation and the Configuration screens in FULL access mode only.
- Power_DN - Displays the time and the date of the last time the Dimension was powered down.
- Power_UP - Displays the time and the date of the last time the Dimension was powered up.

The alarm icon is used to access the alarm status and limit screens. When this icon is selected the display shown in Fig. 2-27 will be displayed. From this display contains the following parameters:

Figure 2-27.
Alarm Display

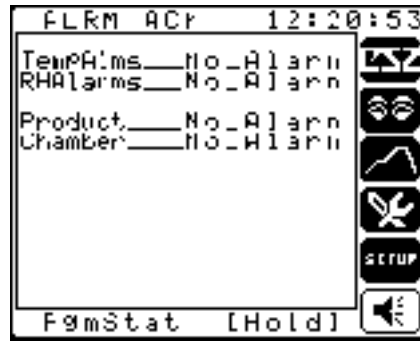


- ALRM ACK - When selected, allows access to the loop alarm status and acknowledge display.
- INPT_ALM - When selected, allows access to the input alarm status and acknowledge display.
- ALM HIST - When selected, allows access to the critical alarm history display.
- ALM_SET - When selected, allows access to the alarm limit setup display.

Operation

When ALRM ACK is selected from the ALARMS display, the screen shown in Fig. 2-28 will be displayed. This screen has the following parameters:

Figure 2-28.
ALRMACK Display



- TempAlms - Displays the status of the temperature loop process variable and deviation alarms.
- RHAlarms - Displays the status of the relative humidity loop process variable and deviation alarms.
- Product - Indicates whether a product high or a product low alarm condition has occurred in the chamber. The limits for this alarm are adjustable by the operator in the PRODUCT window.
- Chamber - Indicates that a high temperature condition has occurred in the chamber. This limit is set by the factory and cannot be changed by the operator. It is used to protect the chamber.

When INPT_ALM is selected from the ALARMS display, the screen shown in Fig. 2-29 will be displayed. This screen has the following parameters:

Figure 2-29.
INPT_ALM Display



- Air Temp - Displays the status of the Air Temp input alarm.
- %Rel Hum - Displays the status of the RH sensor input alarm.
- ProdTemp - Displays the status of the Product Temp input alarm.
- Hi Limit - Displays the status of the chamber high limit input alarm.
- Dewpoint - Displays the status of the dewpoint input alarm.

Alarm Acknowledgment

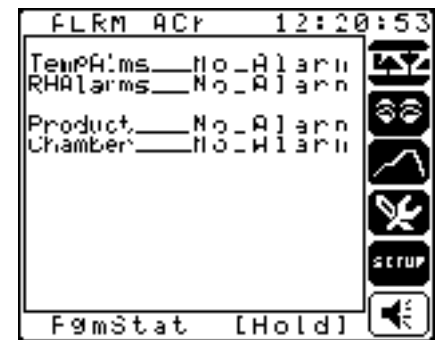
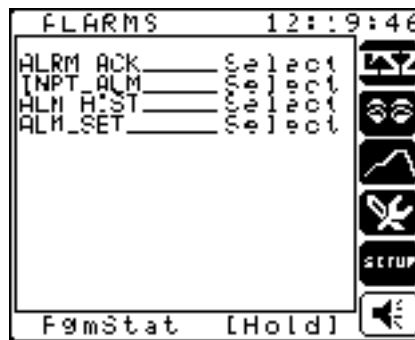
When an alarm becomes active the Alarm icon begins to flash, regardless of which Dimension screen is currently displayed. For unlatched alarms, the Alarm icon stops flashing when the condition that caused the alarm clears. Latched alarms require operator acknowledgment of the alarm to stop the flashing alarm icon.

Acknowledge alarms as follows:

1. From the operation group use the diamond key to access the Alarm Displays icon. The system will display the Alarm Status screen shown in Fig. 2-30. Select ALRM ACK and the ALRM ACK screen will be displayed as shown in Fig. 2-31.

Figure 2-30.
Alarm Display

Figure 2-31.
Alarm Status Display

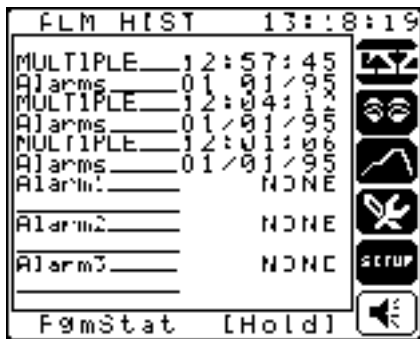


2. If necessary to access the active alarm, select INPT_ALM from the alarms display to access the Input Alarm Status Display.
3. Use the knob to select the alarm you wish to acknowledge. The Product and Chamber Alarm cannot be acknowledged here.
4. Press **ENTER** to acknowledge the alarm. If the alarm cannot be acknowledged (example: an open sensor), the alarm description will not change and the icon will continue to flash. If the alarm is acknowledged, the description will change to "NO ALARM" and the icon will stop flashing.

Operation

When ALM HIST is selected from the ALARMS display, the screen shown in Fig. 2-32 will be displayed. This screen has the following parameters:

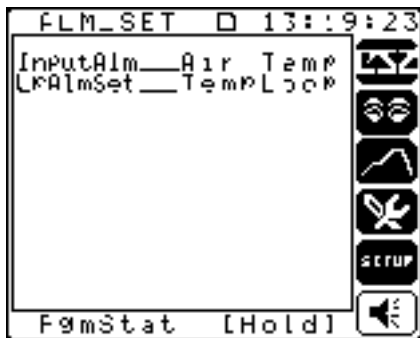
Figure 2-32.
ALMHIST Display



- The six most recent critical alarms along with the date and time that they occurred are logged here.

When ALM_SET is selected from the ALARMS display, the screen shown in Fig. 2-33 will be displayed. This screen has the following parameters:

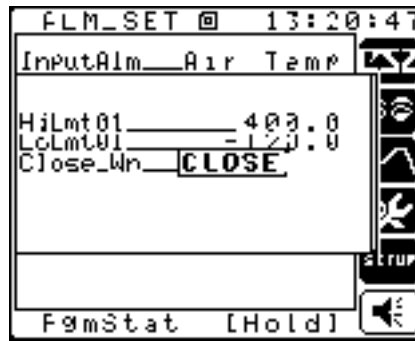
Figure 2-33.
ALM_SET Display



- InputAlm - This parameter allows the user to select the analog input on which to set up the alarms. When the input is selected, a window opens that allows the setting of the alarm limits.
- LpAlmSet - This parameter allows the user to select the specific control loop on which to set up alarms. When the loop is selected, a window opens that provides access to the alarm limit values for the selected loop: HI and LO process variable and deviation alarms. Each loop has up to five different alarm groups.

When and analog input has been selected from the ALM_SET screen, the window shown in Fig. 2-34 is displayed. It contains the following parameters:

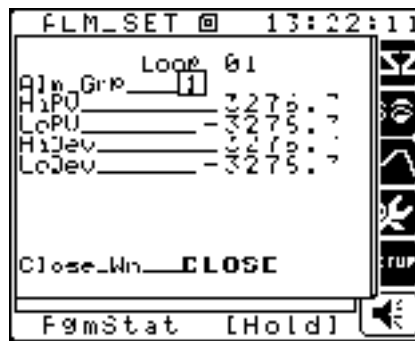
Figure 2-34.
And Analog Input Display



- HiLmt0X - Defines the Hi alarm limit value for the selected input. When the input value exceeds this value, an alarm will be set.
- LoLmt0X - Defines the Lo alarm limit value for the selected input. When the input value is less than this value, an alarm will be set.

When and loop has been selected from the ALM_SET screen, the window shown in Fig. 2-35 is displayed. It contains the following parameters:

Figure 2-35.
And Loop Display



- Alm_Grp - This parameter selects the alarm group in which to set up the limits. There are 5 alarm groups available for each control loop.
- HiPV - This parameter is used to set the Hi process variable limit for the selected loop/ alarm group.
- LoPV - This parameter is used to set the Lo process variable limit for the selected loop/ alarm group.
- HiDev - This parameter is used to set the Hi deviation limit for the selected loop/ alarm group.
- LoDev - This parameter is used to set the Lo deviation limit for the selected loop/ alarm group.

Operation

Section 3

Dimension Programmer

Description

A Real-Time Programmer is a sequence of steps (Segments) that can be programmed to control a setpoint(s) and/or event(s) over a period of time. This allows the setpoint to be adjusted or event to be turned on or off without the need for operator intervention. The generated setpoint may be used as the control loop setpoint or retransmitted as an analog output, for example, to a chart recorder. Events that can turn on digital outputs may be programmed into each segment. These outputs may be used to turn on an external device such as a circulating fan. Each segment may select one of five different alarm groups to allow the use of different alarm limits at various times in the profile. One of five different PID groups may also be used on a per segment basis to achieve better control of the process as it runs through its cycle. The sequence can also be placed in a "Guaranteed Soak" hold condition based on deviation limit.

Terminology

- Alarm Group - Select 1 of 5 different sets of alarm limits that will be active during the current segment. The alarm limits may be set on the ALM_SET screen.
- Current Segment - Displays the current segment being executed by the active recipe.
- Cycle Count - Displays the number of recycles remaining.
- Events (Evnts) - Twelve events are available that may be ON or OFF depending upon the segment. Four of the events, labeled d, e, f, and g are dedicated to the Conditioning System, Humidity System, Auxiliary Cooling and the Purge System. Events 1-8 are optionally available for end user functions.
- Guaranteed Soak - The condition where Deviation limit values are used to put the programmer into "Hold," allowing the process to "catch-up" with a change in the setpoint. The use of Guaranteed Soak will speed-up the test cycle and still maintain correct times for the soak period.

- **Next Segment** - The next segment for the programmer to execute after completing the current segment. In a program recipe setting, the next segment parameter equal to the current segment will cause the recipe to end after the segment completes.
- **PID_Group** - Select 1 of 5 sets of PID parameters (Gain, Reset, Rate) that will be active during the current segment. The PID parameters are set up on the CONF_LOP display. Programmed PID groups must be selected for PID_Type to enable this function.
- **Program Mode** - Displays the current status of the programmer, Run, Hold, Start, PG_END. The selected status may also be entered to change the state of the programmer. The mode will be one of the following values:
 - **Run** - The state of the programmer when it is currently executing its sequence of segments.
 - **Hold** - The state of the programmer when its execution has been stopped. May be due to operator intervention or due to a guaranteed soak condition.
 - **Start** - Begin programmer execution at the selected starting segment.
 - **PG_END** - Displayed when the program completes execution of a segment where the next segment number is equal to the current segment number.
- **Ramp** - The action of increasing or decreasing the setpoint over a period of time. An example of a Ramp is to heat a chamber from 100 °F to 200 °F over a 30 minute time period.
- **Recipe** - A name associated with a sequence of steps (ramp/soak) that begins at a corresponding starting segment. 15 recipes are available in Dimension, each with 50 segments. The recipe name may be altered by the user to be descriptive of the test to be run.
- **Recycles** - The number of repeats for a group of segments. After the number of recycles is completed, the programmer will index to the next sequential segment.
- **Segment** - A step in the recipe sequence. Each step will have its own unique assigned parameters. Dimension has 15 Recipes each with 50 segments available.
- **Segment Time** - The amount of time for the current segment to run. From 1 second to 99 hours.
- **Soak** - The action of maintaining the setpoint at a constant value over a period of time. An example of a soak is to maintain the chamber at 200 °F for 30 minutes.
- **Start Date/Time** - The Date and Time on which you would like the programmer to start. This is used with the Time of Day Start function to start a program at the set date/time.

- Time Left in Segment - Displays the time remaining in the current segment of the programmer.
- Time Left in Program - Displays the time remaining in the current program.
- Time of Day Start - The time of day that you would like to start a programmer running. This allows you to start a process without requiring an operator being present.
- Time of Day Start Segment - The segment number from which to start when the Time of Day start is executed.
- Time of Day Start Enable - Enables the Time of Day start function.

Recipe Definition & Planning

The first step in creating a recipe or setpoint program sequence for your process is to define the individual segments or steps that need to be executed.

This is best accomplished using the following three steps:

1. Create a graphical representation of the recipe showing time & process setpoint values broken down into segments or steps. Refer to Fig. 3-1 for an example recipe graph.
2. Translate the graph steps into a tabular recipe worksheet. Refer to Fig. 3-2 for an example worksheet. Refer to Fig. 3-11 at the end of this section for blank program worksheets that may be copied and used to help create and document recipes.
3. Enter the information from the worksheet to the Dimension. Note that this example recipe is factory loaded into recipe 1 of your Dimension.

Dimension Programmer

Figure 3-1.
Graph of Test to be Performed

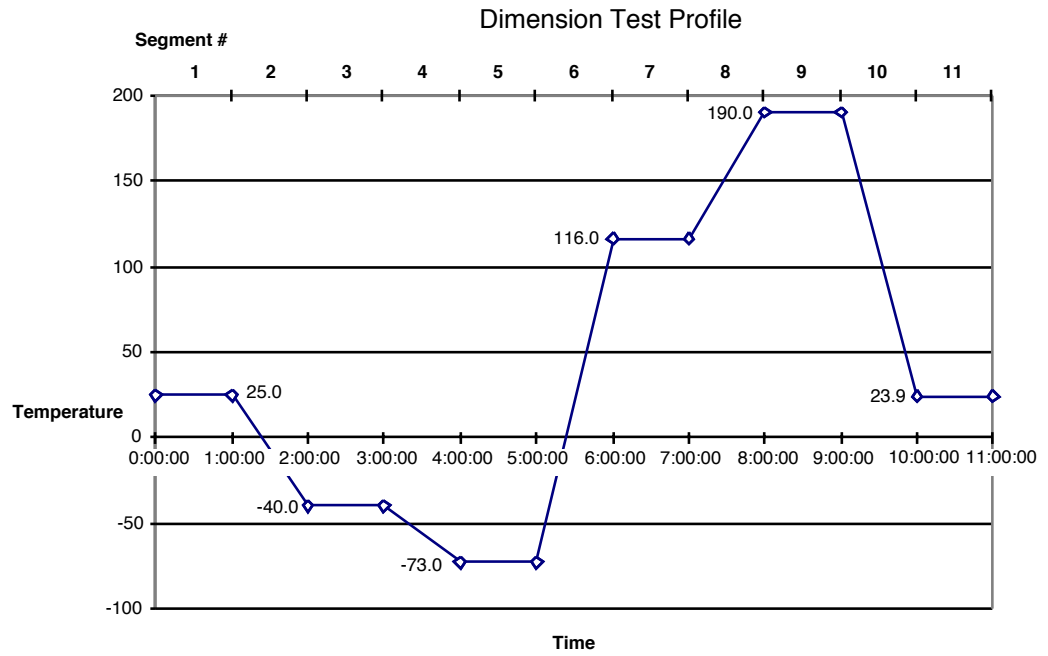


Figure 3-2.
Example Program Worksheet

Program # Recipe (1-15)		1											Description: CSZ Temperature Test Program (°C)	
Segment # (1-50)		1	2	3	4	5	6	7	8	9	10	11		
SetPoint														
AirTemp	1	25.0	25.0	-40.0	-40.0	-73.0	-73.0	116.0	116.0	190.0	190.0	23.9		
Humidity	2													
Segment Time 00:00:00 to 99:59:59		2 sec.	1 hr.	1 hr.	1 hr.	1 hr.	1 hr.	1 hr.	1 hr.	1 hr.	1 hr.	1 hr.	1 hr.	
Events														
Conditioning System	d	x	x	x	x	x	x	x	x	x	x			
Humidity	e													
Aux Cool	f													
Purge	g													
	1													
	2													
	3													
	4													
	5													
	6													
	7													
	8													
Recycles (0-999)												1		
Next Segment (1-51)		2	3	4	5	6	7	8	9	10	1	11		
Alarm Group (1-5)														
PID Group (1-5)														
Guaranteed Soak														
Temp														
Humidity														
Product														
Temp/Hum														

Entering a Recipe into the Dimension Programmer

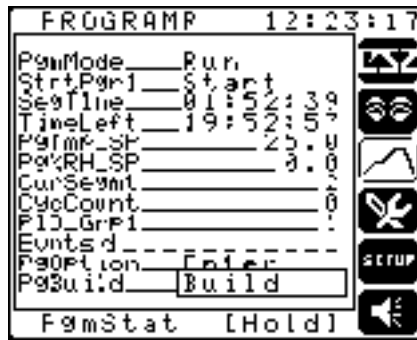


Programmer
Icon

Once you have defined your recipe as described in the previous section the information from the worksheets must be entered into the Dimension. This is accomplished by accessing the programmer icon. When accessed the display shown in Fig. 3-3 will be displayed. This display contains detailed information regarding the programmer, refer to the section on *Monitoring Program Execution* for added details on this display.

Figure 3-3.
Programmer Display(L)

Figure 3-4.
Build Display(R)



From this display select the **Build** parameter, which will open up a window as shown in Fig. 3-4. Enter the information from the worksheet into the corresponding segment parameter as described below:

- **Recipe** - This parameter is used to select which of the 15 available recipes to program.
- **Segment** - This parameter is used to select which segment of the recipes to program. This value can range from 1 to 50 for a selected recipe. A selection of segment 0 will allow programming of the *Program Stop* segment.
- **PgTmp_SP** - This parameter is used to set the desired ending Temperature Setpoint value for this segment. Note: If your Dimension contains more than two control loops selecting this parameter will open a window to allow the entry of this parameter.
- **Pg%RH_SP** - This parameter is used to set the desired ending Relative Humidity Setpoint value for this segment. Note: If your Dimension contains more than two control loops selecting this parameter will open a window to allow the entry of this parameter.
- **Seg_Time** - This parameter is used to define the execution time of the selected segment. This time may range from 0 seconds to 99 hours, 59 minutes and 59 seconds in length.
- **Evnts** - This parameter allows the entry of programmed on/off events. The first four events, entered as *d*, *e*, *f* & *g* are used to turn on/off the Conditioning sub-system and optionally the *Humidity* sub-system, *Auxiliary Cooling* sub-system and the *Purge* sub-system respectively during the recipe. Note that these subsystems must be set to

programmed from the main CSZ display as described in section 2. Refer to your chamber documentation for the applicability of these selections. Events 1 thru 8 are optional user events that can be used to control external on/off devices during the test. Refer to your chamber documentation for the applicability of these selections.

- **Recycles** - This parameter is used to enter the number of recycles or repeats of a series of segments. An entry of 0 indicates no recycling. After the number of recycles is completed the programmer will advance to the next sequential segment. Refer to the section on *Advanced Programmer features* for additional information on using recycles.
- **Next_Seg** - This parameter is used to direct the program to the next segment to execute after the completion of the current segment. Typically this should be the next sequential segment number although this is not a requirement. An entry of a next segment value that is equal to the current segment will cause the program to end. *Note: To link the selected recipe to the next recipe (i.e. recipe 2, segment 50 to recipe 3, segment 1) set the next segment number to 51.*
- **Alm_Group** - This is used to select which one of the 5 available alarm groups will be active during this segment. Refer to the section on *Advanced Programmer features* for additional information on using Alarm Groups.
- **PID_Group** - This is used to select which one of the 5 available PID groups will be active during this segment. (Note that Programmed PID types must be selected to enable this function. Refer to Section 4, Loop configuration for selecting this feature.) Refer to the section on *Advanced Programmer features* for additional information on using PID Groups.
- **GuarSoak** - This parameter is used to enable or disable the Guaranteed Soak hold function for this segment. Guaranteed Soak is used to hold the programmer if a selected control loops deviation from setpoint exceeds a limit value. Choices for this parameter are:
 - **Disabled** - No guaranteed soak for this segment. Typically selected when ramping values.
 - **Temp** - Selected when a guaranteed soak hold is desired based on *chamber temperature* deviation.
 - **Humidity** - Selected when a guaranteed soak hold is desired based on *chamber relative humidity* deviation.
 - **Product** - Selected when a guaranteed soak hold is desired based on *product temperature* deviation.
 - **When a guaranteed soak hold is desired based on chamber temperature deviation Temp/Hum**- Selected when a guaranteed soak hold is desired based on *chamber temperature or humidity* deviation.

The limit values for Chamber Temperature, Humidity and Product temperature are absolute values (+-) and are entered from the *PgOption* screen as described below.

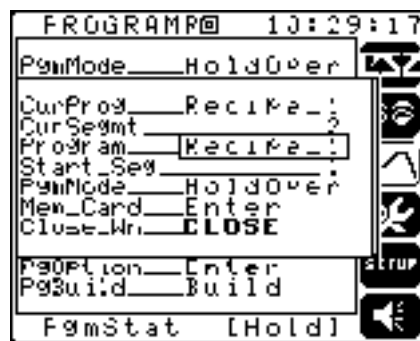
➔ **Note:** When in the program build window the two macro keys located below the display and the diamond key to the left have dedicated functions. The macro key labeled **COPYPREV** may be used to copy the contents of the previous segment into the current segment. The macro key labeled **CLEAR** may be used to clear the current segment parameters. The gold **diamond key** may be used to quickly advance from segment to segment of the selected recipe. Pressing the key to the left will move to the next sequential segment, to the right will move to the previous sequential segment. Use of these functions can speed up, simplify and reduce errors in the recipe definition process.

It is also recommended that all recipes be saved on the Dimension memory card. This will allow quick transfer of recipes if multiple tests are to be run or in the event of a controller failure.

Executing/Starting a Recipe

Once a recipe has been loaded into the Dimension it may be executed from any the Dimension displays. This is accomplished by pressing the gray macro key labeled *PgmStat*. When this key is pressed the window shown in Fig. 3-5 will be displayed. The following parameters are on this display window:

Figure 3-5.
Pgm Stat Display



- CurProg - This parameter *displays* the current or last executed recipe. This parameter is used to identify a currently executing recipe and is not accessible.
- CurSegmt - This parameter *displays* the currently active segment number. This parameter is not accessible.

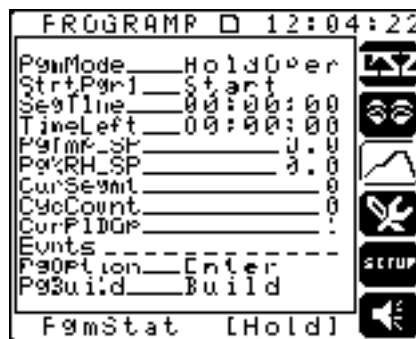
- Program - This parameter is used to **select** which recipe (Recipe_1 thru Recipe_15) to execute.
- Start_Seg - This parameter defines which segment of the selected recipe to start at. Typically this should be segment 1 but can be used to start a recipe in the middle rather than at the beginning.
- PgmMode - This parameter is used to alter the operating mode of the programmer. The following choices are available:
 - Start - Set the parameter to *Start* to initiate the programmer at the selected recipe and starting segment as described above.
 - Run - Set the parameter to *Run* to switch the programmer from a *HoldOper* mode. Note that the programmer cannot be put to *run* from the *HoldSoak* mode.
 - HoldOper - Set the parameter to *HoldOper* to put the programmer into a *hold* mode. When in hold the programmer will remain in it's current state until resumed by the operator.
 - ProgStop - Set the parameter to *ProgStop* to execute the program stop segment zero. This can be used to stop an executing recipe and to place the controller in a known mode as programmed into segment zero.
- Mem_Card - This parameter may be selected to read *recipes* saved on a memory card. Note the *Read* is the only function available from this window. Refer to section X for additional information on using the memory card.

Monitoring Program Execution



To monitor the status of the Programmer in the Dimension access the Programmer display using the Programmer Icon. This display will show details of the state of the programmer as shown in Fig. 3-6. This display has the following parameters:

Figure 3-6.
Programmer Display



- **PgmMode** - This parameter will display the current mode of the programmer and may be used to switch the programmer between *Run* and *HoldOper* modes. *Note: The right macro key may also be used to toggle the programmer between Run and HoldOper modes. If the programmer is in Run mode the macro key will be labeled Hold and when pressed will place the programmer in the HoldOper mode. If the programmer is in HoldOper mode the key will be labeled Run and will resume the programmer operation when pressed.*
- **StrtPgr1** - Select this parameter to open the program start window described in section x.x above. This selection is identical to pressing the *PgmStat* macro key.
- **SegTime** - This parameter displays the time remaining in the currently active segment.
- **TimeLeft** - This parameter displays the time remaining in the currently active recipe.
- **PgTmp_SP** - This parameter displays the current setpoint for the temperature control loop.
- **Pg%RH_SP** - This parameter displays the current setpoint for the relative humidity control loop.
- **CurSegmt** - This parameter displays the currently active segment number within the active recipe.
- **CycCount** - This parameter displays the current recycle count of the active recipe. A value of zero indicates no recycles. Refer to the section on advanced programmer features for a description of recycles.
- **CurPIDGp** - This parameter displays the currently active PID group. Note that this is only applicable if *Programmed PID Groups* are configured for the loops. Refer to Section 4, *Configuration*, of this manual for details on configuring this.
- **Evnts** - This parameter will display the current state of the on/off events for the active segment.
- **PgOption** - Selecting this parameter will open a window (see Fig. 3-7) that allows the setting of the following programmer related parameters:
 - **TODSEnbl** - This parameter is used to *Enable* and *Disable* the Time of Day programmer start function. When enabled the programmer will start at the time/date and recipe defined in the parameters below.
 - **TODStart** - This parameter is used to set the *time* for the time of day start.
 - **StrtDate** - This parameter is used to set the *date* for the time of day start.
 - **TODSRec#** - This parameter is used to select which recipe to execute in the time of day start function.

Figure 3-7.
PgOption Display



- GSTempDv - This parameter is used to define the Guaranteed Soak deviation limit for the Temperature loop.
- GS_RH_Dv - This parameter is used to define the Guaranteed Soak Hold deviation limit for the Relative Humidity loop.
- GSProdDV - This parameter is used to define the Guaranteed Soak Hold deviation limit for the Product temperature loop.
- Dsp_Profile - Select this to graphically display a selected recipe. This may be used to verify that a recipe matches what was intended. Refer to the section in Advanced features for details on Viewing a Recipe.
- PgBuild - This parameter is used to open the program build window. Refer to the section on Entering a setpoint program for details on the program build window.

Advanced Recipe/Programmer features

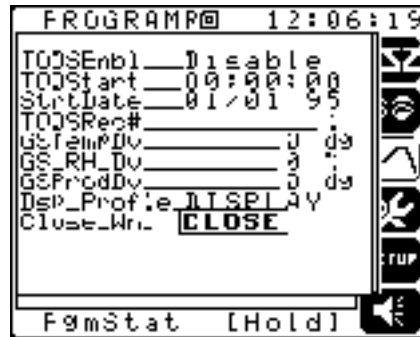
The programmer in the Dimension contains the following features that may be used to shorten program run time, ease program entry and increase control precision

1. Guaranteed Soak
2. Alarm Groups
3. PID Groups
4. Segment Recycles
5. Graphic Display of selected profile.

Guaranteed Soak

Guaranteed Soak is a feature in the Dimension in which a deviation tolerance limit is used to change the programmer between the Run mode and the Hold mode. The most common use of this function is when a test requires that the chamber or the product in the chamber “soak” at a specific temperature (or humidity) for a minimum amount of time. An example of this is a test that ramps temperature from 100 dg to 500 dg and then requires a soak at 500 dg for four hours. Without guaranteed soak the chamber would need to be ramp up at a slow enough rate to ensure that it was at 500 dg when the soak segment started thus ensuring that the four hour soak would be met. When using the guaranteed soak feature the program could step the chamber temperature setpoint from 100 dg to 500 dg the enter the soak segment with guaranteed soak enabled. As soon as the soak segment was entered the program would hold (no program time elapsing) until the chamber reaches 500 dg. At this time the soak segment would resume and start the four hour soak. At any time during the soak if the chamber temperature deviation exceeded the guaranteed soak limit the soak would be held until the temperature came back within tolerance.

Figure 3-8.
PgOption Display



The Dimension guaranteed soak feature can be used with chamber temperature, chamber relative humidity, either temperature or humidity or product (part inside the chamber) temperature. The feature may be enabled or disabled on a per segment basis in the program build window. The deviation limits are set from the program option window and are a +/- limit from the setpoint.

Alarm Groups

The Dimension contains 5 alarm groups for each control loop (temperature, humidity, product temperature). Each alarm group contains a set of alarm limit values for High PV, Low PV, High Deviation and Low Deviation. A common use of the alarm groups is to vary the tolerance to alarms based on what type of test or what portion of a test is being

executed. A typical use is to have wide alarm limits during a ramp of the process and tighter limits during a soak. *Note: The last alarm group selected during a recipe execution remains active even after a recipe has reached program end!*

The Dimension programmer provides an entry in each segment, `Alm_Group`, that is used to select which group is active during that segment.

PID Groups

The Dimension contains 5 PID groups for each control loop (temperature, humidity, product temperature). Each group contains *gain*, *reset* and *rate* tuning constants. The Dimension programmer allows the user to select the active PID group on a per segment basis. This may be used to set PID values on a segment by segment basis to achieve better control as the process ramps and soaks at differing process values. Different PID values may be required to control accurately over the entire process range. To use programmed PID groups the desired control loop's PID type must be set to Programmed. Refer to the Section 4 for details on loop configuration.

The Dimension programmer provides an entry in each segment, `PID_Group`, that is used to select which group is active during that segment. *Note: The last PID group selected during a recipe execution remains active after a recipe has reached program end!*

Segment Recycles

Recycles are used in the programmer to repeat a segment or a group of segments a selected number of times. This can be used to simplify the recipe build process when the test calls for repeating a cycle for a selected number of times. The number of recycles is set on a per segment basis during the recipe build process by setting the *Recycles* parameter in the segment. The recycle value can range from 0 to 255 times. The following items must be followed when using recycles:

1. A maximum of 25 segments may contain recycles. This is a total of 25 segment in all recipes, not per recipe. If more than 25 segments contain recycles the programmer will go to program end immediately after being started.
2. Recycles cannot be used to repeat the current segment. The next segment parameter is used to direct the recipe from segment to segment and if set to recycle the current segment will cause the recipe to end.
3. The recipe will automatically advance to the next sequential segment number once the recycles have completed.

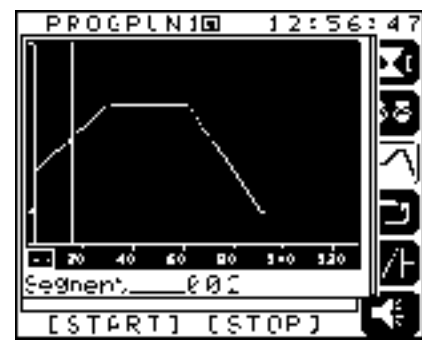
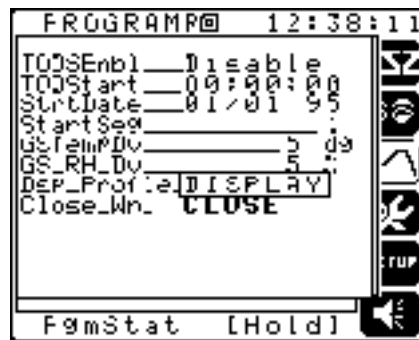
Graphic Display of Selected Profile

Display Profile

When first called up, the text-based Recipe Build screen displays default segment (001) parameters in tabular form. During the recipe building process, a window displaying an X-Y graph of the current state of the recipe can be called up by using the `Dsp_Profile_` from the program option window display (see Fig. 3-9). As shown in Fig. 3-10, this window dynamically displays the recipe. As the recipe develops, the graphic profile scrolls horizontally. When entries are made or edited, successive recipe segments are displayed according to the new values.

Figure 3-9.
Program Option Display (L)

Figure 3-10.
X-Y Display Graph(R)



At the bottom of the X-Y graph is a line of text that scrolls through the selections on the Recipe Build Window. Using the knob, the items scroll as if they were displayed on a full screen. These values can be modified by using the multi-function knob editing functions. As values are entered for profile segments, the profile is graphically displayed. If, for example, the process setpoint is chosen for editing, the text process variable displayed will change when the knob is turned. On the X-Y graph, a line moves up and down on the display representing the relative value of the process variable. When the segment time is changed, the X-Y graph displays that segment, expanding or contracting along the horizontal axis.

When the recipe grows larger than the display screen, the profile will scroll horizontally. The horizontal time scale can be changed to condense or expand the display. A value of 2 will condense the display by a factor of 2. If a segment that has following segments is edited, anything that follows that segment will be graphically erased. When the edited segment is entered, all segments will then display.

When you move the cursor to the small arrow field on the bottom left-hand corner of the display and press **ENTER**, a vertical reference line displays. This line can be moved horizontally by turning the knob. This will cause the text at the bottom of the display to increment or decrement the segment numbers as the line travels across the display. In addition, the macro key labels will be replaced with the time and temperature readings for the segment referenced. Pressing enter on the small arrow field again will enable you to move the cursor to the text line.

From the text line, segments can be selected and modified. Moving the cursor to the **CLOSE** field on the text line closes the graphic window and returns you to the Recipe Build window.

Dimension Programmer

Figure 3-11.
Program Worksheet

Program # Recipe (1-15)		Description:												
Segment # (1-50)														
SetPoint														
AirTemp	1													
Humidity	2													
Segment Time 00:00:00 to 99:59:59														
Events														
Conditioning System	d													
Humidity	e													
Aux Cool	f													
Purge	g													
	1													
	2													
	3													
	4													
	5													
	6													
	7													
	8													
Recycles (0-999)														
Next Segment (1-51)														
Alarm Group (1-5)														
PID Group (1-5)														
Guaranteed Soak														
Temp														
Humidity														
Product														
Temp/Hum														

Figure 3-12.
Program Worksheet

Program # Recipe (1-15)		Description:												
Segment # (1-50)														
SetPoint														
AirTemp	1													
Humidity	2													
Segment Time 00:00:00 to 99:59:59														
Events														
Conditioning System	d													
Humidity	e													
Aux Cool	f													
Purge	g													
	1													
	2													
	3													
	4													
	5													
	6													
	7													
	8													
Recycles (0-999)														
Next Segment (1-51)														
Alarm Group (1-5)														
PID Group (1-5)														
Guaranteed Soak														
Temp														
Humidity														
Product														
Temp/Hum														

Dimension Programmer

Figure 3-13.
Program Worksheet

Program # Recipe (1-15)		Description:												
Segment # (1-50)														
SetPoint														
AirTemp	1													
Humidity	2													
Segment Time 00:00:00 to 99:59:59														
Events														
Conditioning System	d													
Humidity	e													
Aux Cool	f													
Purge	g													
	1													
	2													
	3													
	4													
	5													
	6													
	7													
	8													
Recycles (0-999)														
Next Segment (1-51)														
Alarm Group (1-5)														
PID Group (1-5)														
Guaranteed Soak														
Temp														
Humidity														
Product														
Temp/Hum														

Section 4

Configuration

Description

This section describes the setup and configuration functions that are available to customize the Dimension controller for specific user installations or tests. It also provides access to diagnostic and unit option display screens.

Refer to Fig. 4-3 for a reference map of the configuration displays discussed in this section.

This section assumes that the user is familiar with the conventions and operation of the Dimension Custom Logic controller. Refer to Section 2 of this manual for basic operational information.

Configuration Access

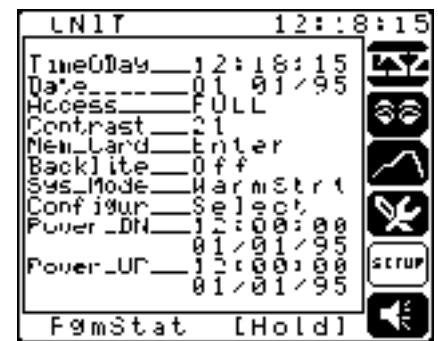
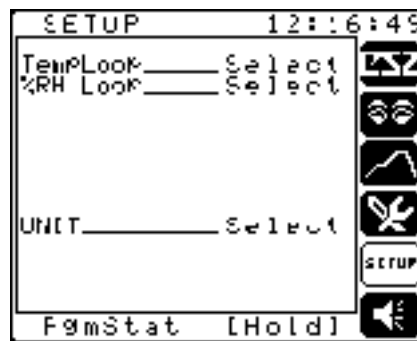


SETUP
ICON

The access to the unit configuration area of the Dimension is from Setup icon on the operator interface. From the display shown in Fig. 4-1, select the UNIT parameter. This will link directly to the UNIT display, as shown in Fig. 4-1, where you may select the Configur parameter obtain access to the display shown in Fig. 4-4 and begin the configuration and setup process.

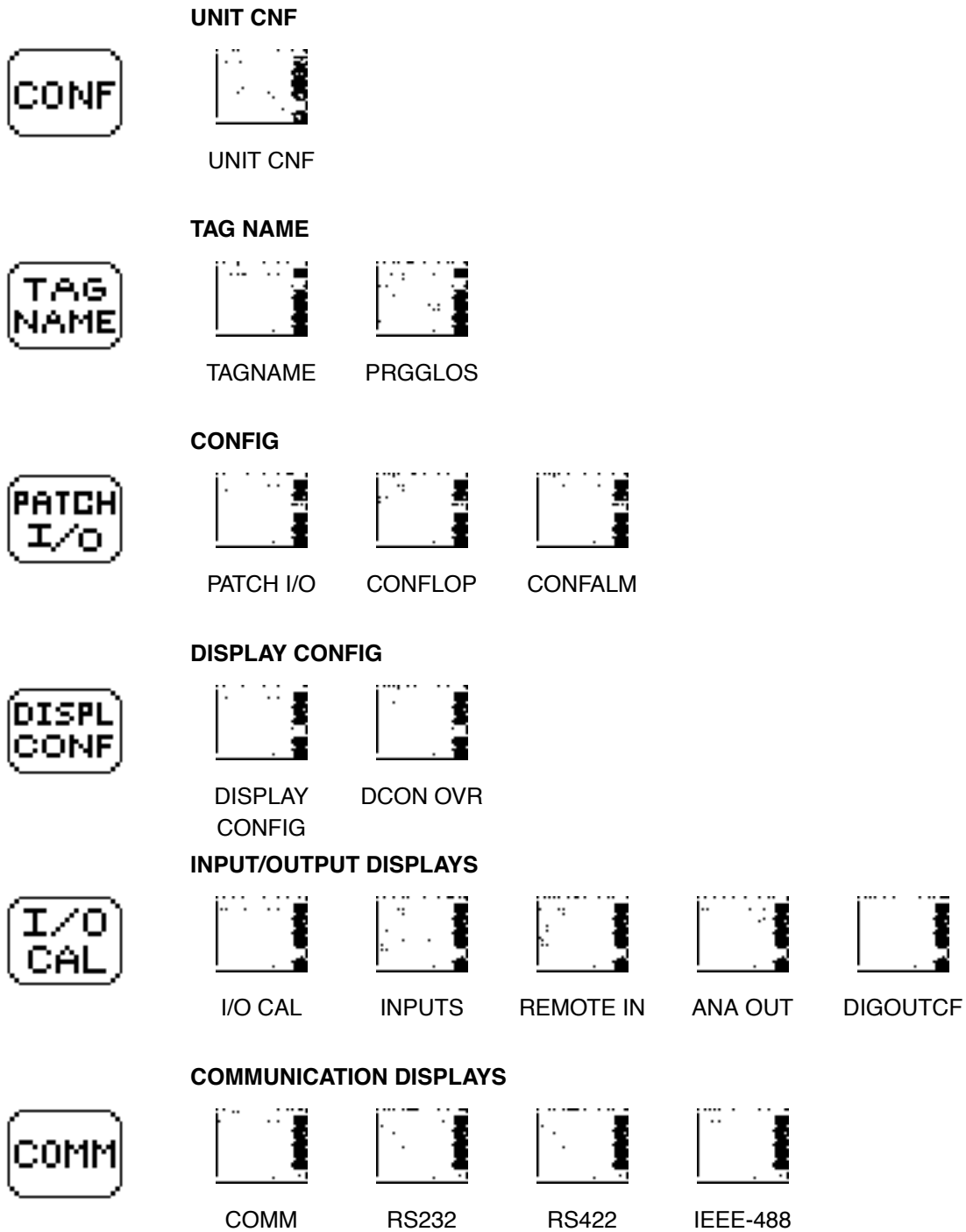
Figure 4-1.
The Setup and Unit Display
Screens (L)

Figure 4-2.
Unit Display (R)









Configuration

Figure 4-3.
Map of Dimension Configuration Screens



The Configuration area contains six different sections that are used to configure or setup various parts of the Dimension. Each of these are identified by the following icons:

-  • CONF icon is used to access Unit or Dimension related parameters.
-  • TAG NAME icon is used to modify selected display tag names to match the users process.
-  • PATCH I/O icon is used to modify control loop and alarm parameters.
-  • DISPL CONF icon is used to create/modify user overview displays.
-  • I/O CAL icon is used to configure and calibrate analog and digital input and output points.
-  • COMM icon is used to setup the RS-232/422 and IEEE-488 communications functions.

Unit Configuration

The unit configuration area is accessed from the UNIT ICON and contains the display shown in Fig. 4-4. This display contains the following parameters:

Figure 4-4.
Unit Configuration Display



- Full_Acc - This parameter is used to define the *Full Access* password value. Full access allows the user complete access to all parameters and is required to perform any configuration functions.
- Lmtd_Acc - This parameter is used to define the *limited access* password value. Limited access allows access to operating parameters only.
- Srvc_Acc - This parameter allows access to CSZ service and setup displays and functions when the top-secret password is entered.
- Chng_Lvl - This parameter is used to select the active access level. Selections are *Full*, *Limited* and *None*.

Configuration

- **Sys_Mode** - This parameter is used to define the action of the Dimension upon power up. Selections are *ColdStart* which will return the Dimension to a factory default condition, *WarmStart* which will save all current operating conditions/settings except the conditioning systems will be turned OFF and the programmer will be placed in Hold, *AutoStart* which will save all current operating conditions/settings including the conditioning systems and programmer status, and *H-AStart* which saves all current operating conditions but places the programmer in HOLD.
- **Diagnost** - This parameter is used to access diagnostic display information. When this parameter is selected a window will open to allow the selection of a specific diagnostic function. Choices are:
 - **OI_Diag** - This selection displays the software version number of the controller and to exercise the various knobs & keys on the Dimension operator interface.
 - **CC_Diag** - This selection will display the software version number of the analog I/O control cage cards.
 - **TimeDiag** - This selection displays the latest Power Down and Power Up time/dates for the Dimension.
- **Option 1,Option 2** - This selection displays a summary I/O and Option count for the Dimension.

Tag Name Configuration

The tagname configuration icon is used to alter the standard terminology of selected Dimension parameters to match a users specific process. When this icon is selected the display shown in Fig. 4-5 will be displayed. Selecting the programmer parameter will access the display in Fig. 4-6 that allows the modification of the following tagnames:

Figure 4-5.
Tagnames Display (L)



Figure 4-6.
Programmer Configuration Display (R)



- **ProgmrNm** - Used to select which programmer to configure. (This will typically be restricted to Progmr01)
- **ProgName** - This parameter may be selected modify the default Recipe_1 thru 15 tagnames to names that defines your process recipes.
- **Events** - This parameter may be selected to modify the tagnames of the 8 user defined on/off events to match your process event functions.
- **PgMode** - This parameter may be selected to modify the tagname of the programmer mode tagname. This parameters displays the status of the programmer as *Run*, *HoldOper*, *HoldSoak* or *PG_END*.
- **ProfileSP** - This parameter may be modified to change the programmer profile setpoint names. This may be used to change the setpoint names to match the control loop functions.
- **T_Seg** - This parameter may be modified to change the *Time Left in Segment* tagname. This parameter displays the time left to execute in the currently active segment of a running program.
- **T_Into** - This parameter may be modified to change the *Time Into Program* tagname. This parameter displays the amount of time an program has executed.
- **T_Left** - This parameter may be modified to change the *Time Left in Program* tagname. This parameter displays the amount of time an executing program has left to execute before it completes.
- **CurSeg** - This parameter may be modified to change the *Current Segment* tagname. This parameter displays the currently active segment number.
- **CycCnt** - This parameter may be modified to change the *Recycle Count* tagname. This parameter displays the number of recycle remaining in an active program.
- **Alm_Grp** - This parameter may be modified to change the *Current Alarm Group* tagname. This parameter displays the currently active alarm group.
- **PID_Grp** - This parameter may be modified to change the *Current PID Group* tagname. This parameter displays the currently active PID group when programmed PID groups are selected.

Patch I/O Configuration

This section allows access to the configuration of the control loops and alarm functions. When this icon is selected the display shown in Fig. 4-7 will be displayed. Select the **ContrLoop** parameter to configure control loop parameters and **Alarms** to setup analog input and control loop alarm parameters.

Figure 4-7.
Patch I/O Display (L)

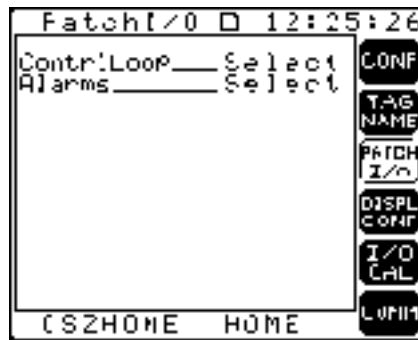
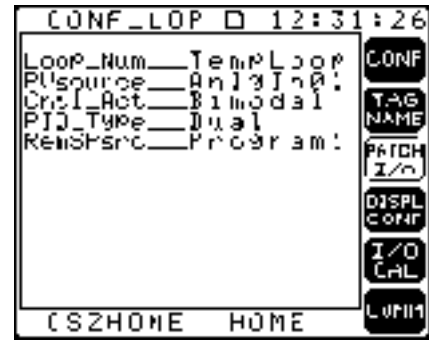


Figure 4-8.
Control Loop Display (R)



When the **ContrLoop** parameter is selected the display shown in Fig. 4-8 is shown and the following parameters may be altered:

- **Loop_Num** - This parameter is used to select which control loop to configure. The range of this will depend upon your chamber configuration.
- **PVSource** - This parameter is used to select which analog input is used as the process variable source for the selected control loop. This parameter is set at the factory and usually will not need to be altered by the user.
- **CntlAct** - This parameter is used to select the control action of the selected control loop. The default selection is **Bimodal**. Choices are:
 - **Bimodal** - Used for control loops the require both a *Reverse* (heating, humidify) and a *Direct* (cooling, dehumidify) output. This parameter is set at the factory and usually will not require user configuration.
 - **Reverse** - Used for control loops that require only a reverse acting output (heating) control signal.
 - **Direct** - Used for control loops that require only a direct acting (cooling) control signal.
 - **None** - Used to select no control output signal.

➔ **Note:** When selecting one of these choices the user may also set output high and low limits for the reverse or direct outputs. These limits may range from 0.0 to 100.0%.

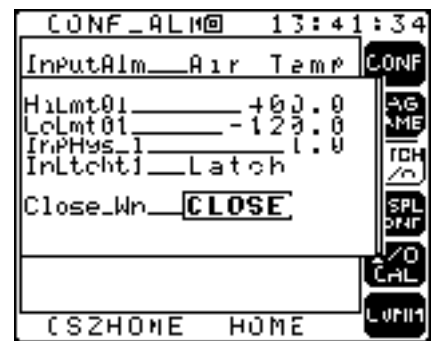
- **PID_Type** - This parameter is used to select the desired PID type. The default selection is DUAL. Choices are:
 - **Single** - Will provide 1 set of PID values (Gain, Reset & Rate) for both reverse and direct outputs.
 - **Dual** - Will provide a set of PID values for reverse (group 1) and a second set for direct (group 2).
 - **Progmmmed** - Allows one of five groups of PID values to be used depending upon the PID_Group defined in the active program.
 - **SelAdapt** - Allows five sets of PID values to be defined along with five breakpoints. The breakpoints are used to select a PID group based upon the control loop process variable. When between breakpoints the PID values will be interpolated.
- **RemSPsrc** - Allows the user to define the Remote Setpoint source value for the selected control loop. This parameter defaults to *Program1* but may be set to one of the following:
 - **Anlgn#** - Allows setting the remote setpoint source to an analog input. This may be used to get a setpoint from an external instrument.
 - **OutputL#** - Allows setting the setpoint from the output of another control loop. This may be used to implement cascade/blended cascade control. When selected the source for a remote setpoint and for ratio, blend and scale may also be entered.
 - **SetPntL#** - Allows the setpoint to be from another control loops setpoint. This allows two loops to be set to the same value with one entry.

When the Alarms parameter is selected the display shown in Fig. 4-9 will be displayed. When the **InputAlm** selection is made the display shown in Fig. 4-10 will be displayed and the following analog input alarm parameters may be set:

Figure 4-9.
Configure Alarm
Display (L)



Figure 4-10.
Input Alarm Display (R)

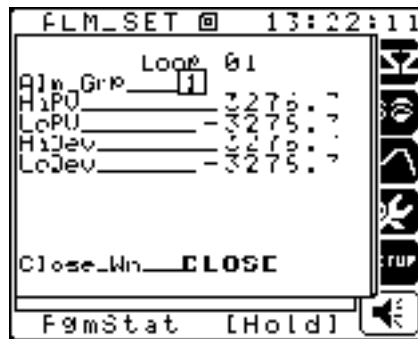


Configuration

- **HiLmt#** - This parameter is used to set a high limit alarm value for the selected analog input.
- **LoLmt#** - This parameter is used to set a low limit alarm value for the selected analog input.
- **InpHys#** - This parameter is used to set a hysteresis value for the selected analog input. Hysteresis value is a band value that the alarm must clear by before being reset.
- **InLtcht#** - This parameter allows the selection of *Latched* or *Unlatched* alarms. Latched alarms must be acknowledged by the user before they will clear, while unlatched alarms will clear without operator interaction whenever the alarm limit is not being exceeded.

When the **LpAlmSet** parameter is selected the window in Fig. 4-11 will be displayed and the following parameters may be set:

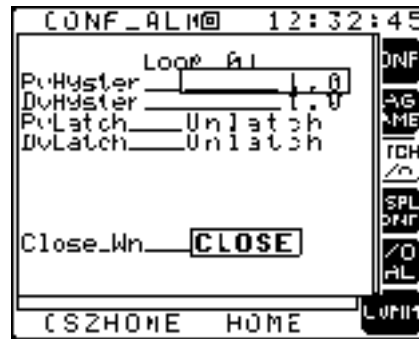
Figure 4-11.
Loop Alarm Set Display



- **Alm_Grp** - This parameter is used to select which alarm group (1 thru 5) to set the limits into. Multiple alarm groups are used to vary alarm limits depending upon the state of your process.
- **HiPV** - This parameter is used to set a high process variable alarm limit. If the process variable (temperature, humidity) exceeds this value an alarm will be set.
- **LoPv** - This parameter is used to set a low process variable alarm limit. If the process variable (temperature, humidity) is less than this limit value an alarm will be set.
- **HiDev** - This parameter is used to set a high deviation from setpoint alarm limit. If the deviation between process variable and the setpoint (PV - SP) exceeds this value an alarm will be set.
- **LoDev** - This parameter is used to set a low deviation from setpoint alarm limit. If the deviation between process variable and the setpoint (PV - SP) is less than this value an alarm will be set.

When the Loop_Alm parameter is selected the window in Fig. 4-12 will be displayed and the following parameters may be set:

Figure 4-12.
Loop Alarm Display



- PvHyster - This parameter is used to set the high/low process variable alarm hysteresis value.
- DvHyster - This parameter is used to set the high/low deviation from setpoint alarm hysteresis value.
- PvLatch - This parameter is used to select the latched/unlatched status of the process variable alarm. Latched alarms require user acknowledgment to clear while unlatch alarms will clear as soon as the alarm condition clears.
- DvLatch - This parameter is used to select the latched/unlatched status of the deviation from setpoint alarm. Latched alarms require user acknowledgment to clear while unlatch alarms will clear as soon as the alarm condition clears.

Display Configuration

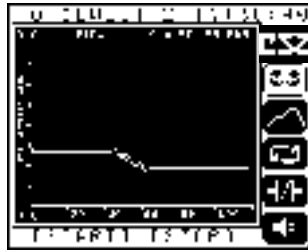
The Dimension contains seven Overview displays that are designed to be customizable by the user on a process or user basis to present process information in various formats. This flexibility allows displays to be formatted that will provide overview or detailed information and allow process interaction to occur from as few displays as possible. The displays shown below provide examples of the available overview display formats.

Configuration

Figure 4-13.
Overview Displays



XY Chart Display
Two parameters may be charted versus time and four parameters may be displayed as text.



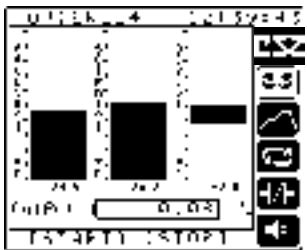
XY Display
Identical to XY Chart Display, but no text is shown.



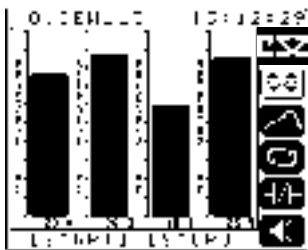
HOR_BAR
Two parameters are displayed in a horizontal bar format and five parameters may be displayed as text.



VER_BAR2
Two parameters may be displayed as vertical bar graphs, and four parameters may be displayed as text.



VER_BAR3
Three parameters may be displayed in bar graph format. One parameter may be displayed as text.



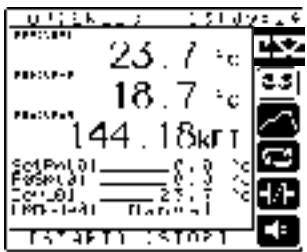
VER_BAR4
Four parameters may be displayed in bar graph format.



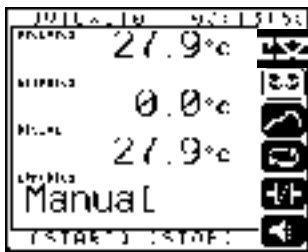
TEXT
Twelve parameters may be displayed in text format.



TEXT_2X
Eight parameters may be displayed in 2X text format.



TEXT_3X
Three parameters may be displayed in 3X text format and four parameters may be displayed as text.



TEXT_3X_4
Four parameters may be displayed in 3X text format.

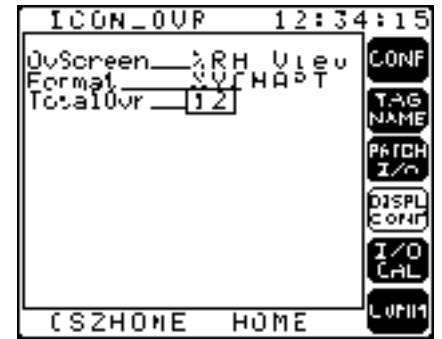


To configure displays access the Disp Conf icon. The screen shown in Fig. 4-14 will be displayed. From this display select the Overview parameter. This will bring up the display shown in Fig. 4-15 and the configuration process may begin. The first step is to select which screen to configure. This is accomplished from the OvScreen parameter, next move to the Format parameter and select the display type to assign to the selected overview display. Refer to Fig. 4-13 for a list and description of the available display types.

Figure 4-14.
Displays Display (L)



Figure 4-15.
Overview Display (R)



Input and Output Configuration and Calibration



The configuration and calibration of the Dimension analog and digital inputs and outputs is accomplished from the I/O Cal icon. When accessed the screen shown in Fig. 4-16 is displayed. Note that most configuration functions are set from the factory and require no user alterations. From this screen the following choices are available:

Figure 4-16.
I/O Calibration Display

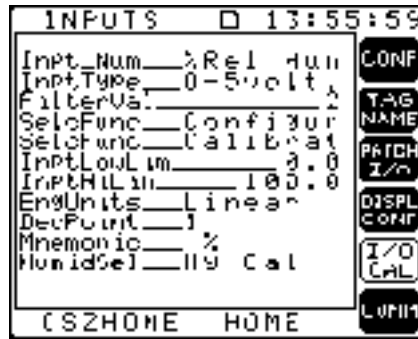


- AnalogIns - This selection allows access to the local analog inputs configuration and calibration display.
- RemoteIns - This selection allows access to the remote (machine control board) analog inputs configuration display.
- AnalogOuts - This selection allows access to the local analog output configuration and calibration display.

Analog Inputs

When the **AnalogIns** selection is made from the **I/O Calb** display the screen shown in Fig. 4-17 will be displayed. This screen has the following parameters:

Figure 4-17.
Inputs Display



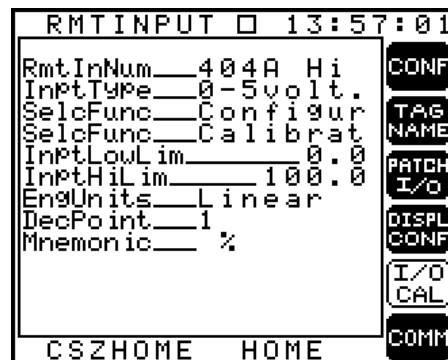
- **Inpt_Num** - This parameter is used to select which of the six local analog inputs to configure/calibrate.
- **InptType** - This parameter is used to define the input type (thermocouple, humidity sensor, voltage...) of the selected analog input. This is set at the factory and should not normally require user alterations.
- **FilterVal** - This parameter is used to set in the amount of digital filtering for the selected analog input. This parameter has a range from 0 to 100 and 0 is no filtering and 100 the maximum.
- **SelcFunc Configur** - Selecting this will open a window that is used to configure a linear inputs zero, span, decimal precision and mnemonic. This is typically set at the factory and should require no end user alterations.
- **SelcFunc Calibrat** - Selecting this will open a window that is used to calibrate the selected analog input. Refer to appendix X, Verification/Calibration Procedure for details on this screen and the calibration process.
- **InptLowLim** - This parameter defaults to the low limit of the selected input type. This parameter may be altered to limit the range of user entered values associated with this input such as setpoints and alarm limits.
- **InptHiLim** - This parameter defaults to the high limit of the selected input type. This parameter may be altered to limit the range of user entered values associated with this input such as setpoints and alarm limits.
- **EngUnits** - This parameter is used to set the engineering units of the selected input. Choices are Centigrade or Fahrenheit for thermocouples & RTDs and Linear for linear (current, voltage) inputs.

- **DecPoint** - This parameter is used to set the decimal precision for linear analog inputs and displays the precision for thermocouple and RTD inputs.
- **Mnemonic** - This parameter is used to define the engineering unit mnemonic for linear analog inputs and displays "C" or "F" for thermocouple and RTD inputs depending upon the *EngUnits* choice.
- **HumidSel** - This parameter is used by factory configuration experts to define the type of humidity sensor installed in the chamber. This parameter is normally display only.

Remote Analog Inputs

When the *RemoteIns* parameter is selected the display shown in Fig. 4-18 is displayed. Note that these inputs are factory configured and should not require end user modification. The following parameters are available:

Figure 4-18.
Remote In Display



- **RmtInNum** - This parameter is used to select which of the six remote inputs to configure/calibrate.
- **InptType** - This parameter is used to define the input type of the selected input.
- **SelcFunc Configur** - Selecting this will open a window that is used to configure a linear inputs zero, span, decimal precision and mnemonic. This is typically set at the factory and should require no end user alterations.
- **SelcFunc Calibrat** - Selecting this will open a window that is used to calibrate the selected analog input. Refer to appendix X, Verification/Calibration Procedure for details on this screen and the calibration process.
- **InptLowLim** - This parameter defaults to the low limit of the selected input type. This parameter may be altered to limit the range of user entered values associated with this input such as setpoints and alarm limits.

- **InptHiLim** - This parameter defaults to the high limit of the selected input type. This parameter may be altered to limit the range of user entered values associated with this input such as setpoints and alarm limits.
- **EngUnits** - This parameter is used to set the engineering units of the selected input. Choices are Centigrade or Fahrenheit for thermocouples & RTDs and Linear for linear (current, voltage) inputs.
- **DecPoint** - This parameter is used to set the decimal precision for linear analog inputs and displays the precision for thermocouple and RTD inputs.
- **Mnemonic** - This parameter is used to define the engineering unit mnemonic for linear analog inputs and displays "C" or "F" for thermocouple and RTD inputs depending upon the *EngUnits* choice.

Analog Output Configuration and Calibration

When the **AnalogOuts** parameter is selected the display shown in Fig. 4-19 is displayed and the following parameters are available:

- **AnaOutpt** - This parameter is used to select which of the four local analog outputs to configure/calibrate.
- **SelcFunc_Configur** - Selects the analog output configuration function. This function opens a window as shown in Fig. 4-19 to allow setting the output zero and span values. Each 0-5V or 0-20 mA analog output configuration defaults to:

Figure 4-19.
Configure Analog
Outputs Display



Zero = 0.00

Span = 100.00

For 4-20 mA output the configuration defaults to:

Zero = -25.00

Span = 100.00

➔ **Note:** Slopes CAN be negative (inverted) by configuring the output as shown:

$$\text{Zero} = 100.00$$

$$\text{Span} = 0.00$$

An output of 0.00 gives 20mA out

An output of 100.00 gives 4mA out

To calculate zero and span values for a 4-20mA output:

Span = Upper Limit of Range of variable that is patched to the output

$$\text{Zero} = \text{LowLim} - \frac{\text{Range}}{4}$$

Where: LowLim = Low Limit for the variable patched to the output.

Range = The absolute value of the difference between the High Limit and Low Limit [ABS (Hi-Lo)].

Example 1:

Retransmit Analog Input to Analog Output

Input Range is 0°–1000° F

$$\text{Span} = 1000$$

$$\text{Zero} = 0 - \frac{1000}{4}$$

$$\text{Span} = 1000$$

Example 2:

Retransmit Setpoint for a Loop to an Analog Output

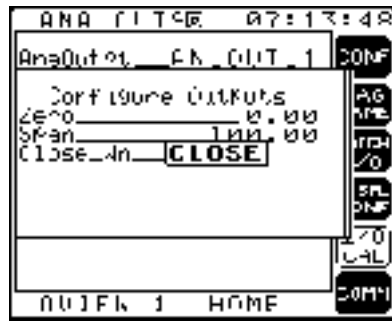
Setpoint Range is –80° C to 500° C.

$$\text{Span} = 500$$

$$\begin{aligned} \text{Zero} &= -80 - \frac{580}{4} \\ &= -225 \end{aligned}$$

Configuration

Figure 4-20.
Calibrate Analog Outputs



- SelcFunc_Calibrat – Opens a window to allow setting the zero and span calibration points (see Fig. 4-20). These points are different from the zero and span in the Config window. The calibration zero and span values calibrate the output to provide the correct voltage or current out. To calibrate, connect a volt meter to the appropriate terminals and adjust the span and zero values. When complete, save your calibration settings by selecting Save. If you wish to revert to the original settings select Abandon. Select CLOSE to close the window and return to the main analog outputs screen.
- Zero – Displays the zero value setup in the configuration window.
- Span – Displays the span value setup in the configuration window.
- CyclTime – Selects the cycle time in seconds. If the cycle time is set to 0 this output will function as an analog proportional output. If the cycle time is set to any value greater than 0, this output will function as a time proportioning output.

Communications

The Dimension is equipped with RS-232, RS-422 or optionally IEEE-488 communications interfaces to provide the ability to connect a supervisory computer system to the unit. Refer to appendix B, *Communications*, for details on the Dimension's communications capabilities and the setup of the communications interface.

Appendix A

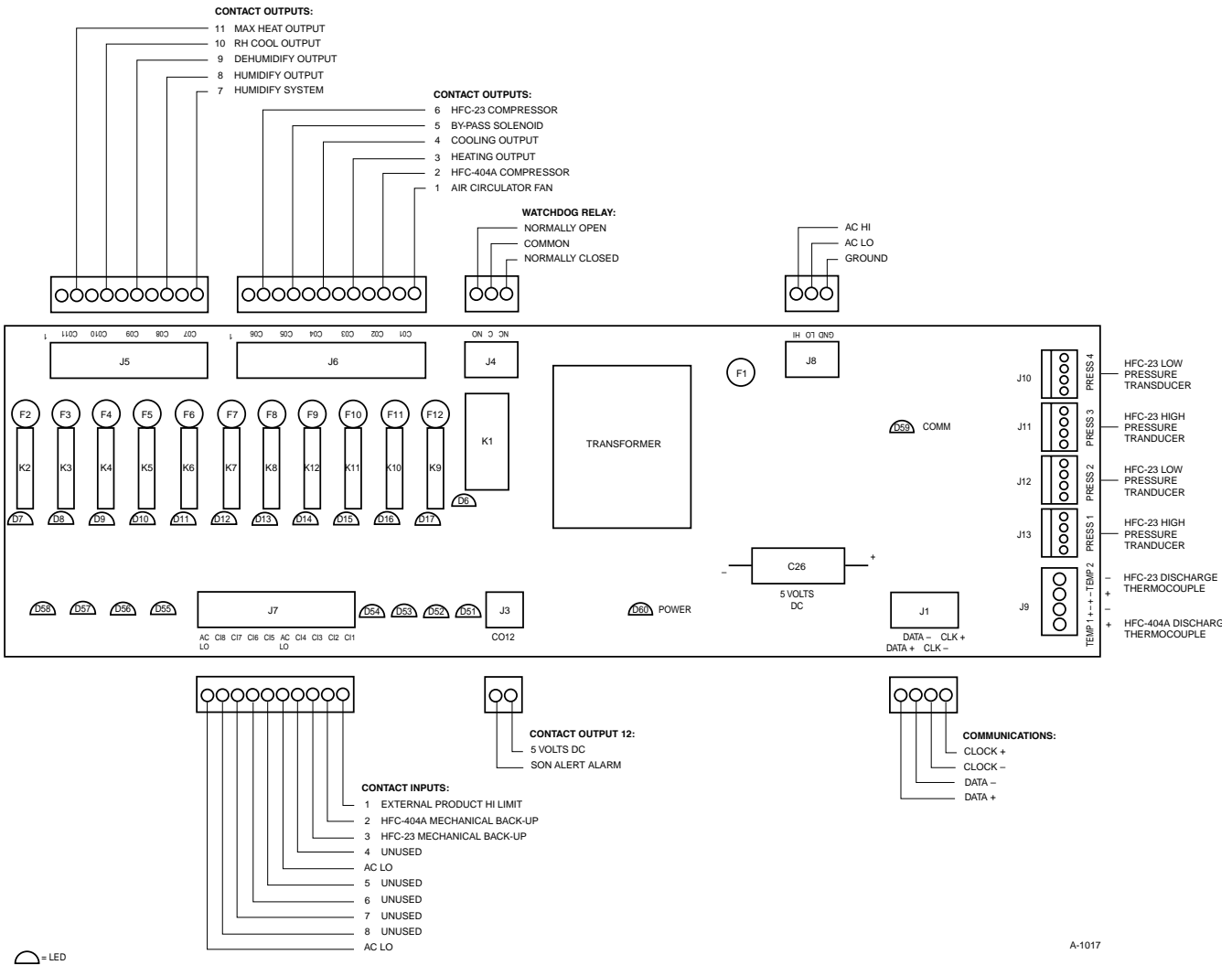
Hardware/Wiring Connections

Description

The typical connections for the CSZ Dimension (Z units) are shown in Fig. A-1 and Fig. A-2.

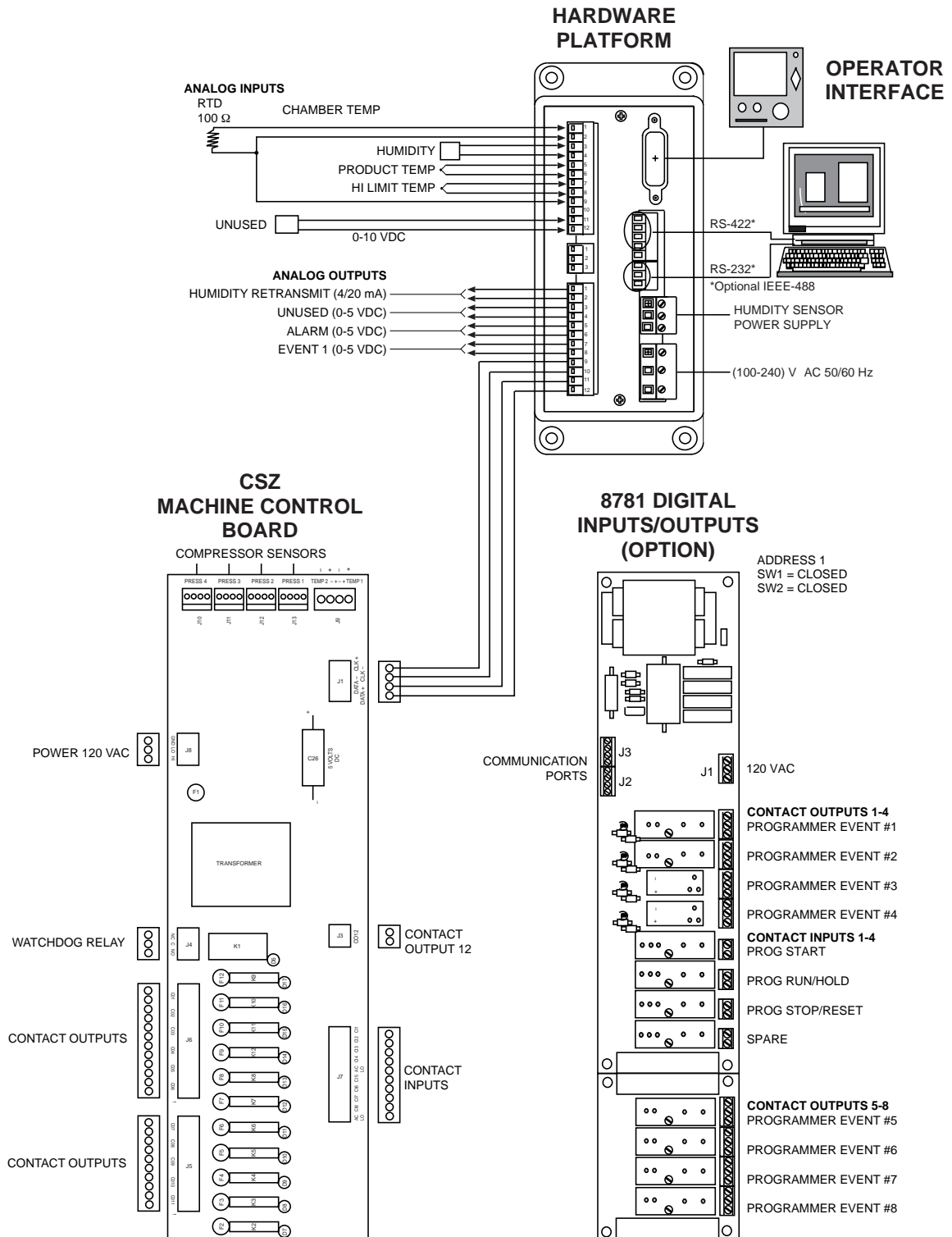
Hardware/Wiring Connections

Figure A-1.
CSZ Machine Control Board Connections



A-1017

Figure A-2.
Connections for Hardware Platform, Machine Control Board and Optional 8781 Digital I/O



A-1019

Trouble Shooting

Multiple Alarms

Multiple alarms are the messages the controller displays when it senses more than one alarm occurring at the same time. The probability of two alarms actually occurring simultaneously is remote.

There are several other conditions that can cause the controller to display two or more alarms without this being the case.

1. Loss of power to the inputs. If power is lost to the inputs, the controller will interpret this as multiple alarms. This could be caused by a pump down, not being initiated, or the chamber mechanical high limit opening up (Z units only).
2. Loss of power to the machine control board (MCB). The MCB is located on the sub-panel and it senses the pressures and all the 120 volt input and output signals.

If power is lost to this board, the controller gets no response from the MCB. The lack of response is interpreted as a 0 for pressure and 0 inputs are OK.

You can see if the controller is still communicating with the MCB by going to the UTILITY screen and selecting SERVICE. Look for pressures and discharge temperatures. If you have pressures or discharge temperatures for either stage, the MCB is running. If not, check power to the board and check the fuse on the MCB to see if it is OK.

*A new feature on the MCB is a LED to indicate power to the board. It will be lit when there is 5 volts DC on the MCB and off if there is not.

3. Loss of communication to the MCB. This occurs when one or more of the data lines (wires 20 through 23) are no longer connected. The symptoms are the same as item #2, but with power on the MCB. Check the wire connections.

*A new feature on the MCB is a LED to indicate communications to and from the MCB. This LED will blink continuously when the system is operating properly. A constant on or off indicates a problem. Check the wire connections.

4. Failure of the base unit. IF the base unit fails and will not communicate with the MCB, the symptoms are almost indistinguishable from item #3. Contact CSZ if this occurs to help resolve the problem.

* Feature added July 1997

Chamber Limit

The chamber limit is a built-in safety that protects the chamber from damage in extreme conditions. The chamber limit thermocouple comes into analog input #4. This T/C is located in the chamber plenum. The alarm settings are fixed and cannot be changed. If the T/C senses a temperature outside the alarms settings, typically +410 F (+216 C) for the high limit and -120 F (-85 C) for the low limit, the chamber will shut down and not restart until the condition has been cleared. If this alarm occurs and the chamber is not at extremes, the first thing to check would be the ANALG IN Screen. This screen is accessed from the UTILITY screen.

Check the status of the High Limit T/C. Is it displaying an open sensor? If it displays an open sensor, check the T/C junction - something may have happened to it and it needs to be repaired. This can be fixed by cutting off the old tip and soldering the wires together with high temperature solder. Does the temperature jump around? If the input behaves erratically, the T/C junction may be failing and need to be repaired. Does the temperature seem reasonable? A possible cause is a problem with the circulator fan. This can cause heat to build-up in the plenum and cause the sensor to register a high alarm.

Back-Up Limits

The back-up limits do not directly sense the alarm condition. An intermediate sensor senses the alarm and the CSZ Dimension sees this as a switch opening. The CSZ Dimension then shuts down the chamber until the alarm condition on the limit has been cleared.

Back-Up Product

This alarm will occur if you have purchased the independent/redundant product limit and it registers an alarm condition. The independent product high/low then opens a contact to the CSZ Dimension. The CSZ Dimension then shuts down the chamber until the alarm condition on the limit has been cleared.

Back-Up 404A And 23

This sensor is a mechanical pressure sensor. On Z chambers and most units with single phase hermetic compressors, the pressure switch is self-resetting/fixed at 350 psig. On larger units, any of the refrigeration safeties can cause this message, such as high pressure, low pressure, high discharge temperature, oil pressure, and (10 HP and larger) phase loss module.

During an alarm condition, the CSZ Dimension senses the switching and shuts down the chamber. The chamber cannot restart until the condition is cleared.

MCB Trouble Shooting Aids

- | | |
|----------------------------|---|
| Output On/Off Indicators | - Each Contact Output (CO1 to CO11) has a LED to indicate status.
Lit = ON Unlit = off |
| * Input On/Off Indicators | - Each Contact Input (CI1 to CI8) has a LED to indicate status.
Lit = ON Unlit = off |
| * Power Indicator | - LED to indicate power on/off status.
(5 volt DC) |
| * Communications Indicator | - This LED will blink continuously when the system is operating properly. A constant on or off indicates a problem. |

Appendix B

Communications

This appendix explains how to communicate with Dimension 8705 and 8725 Industrial Process Managers, via an RS-232 or RS-422 serial link or IEEE-488 General Purpose Interface Bus, from a terminal or host computer. All parameters needed for terminal or network mode communications are described.

Serial Communications Overview

Serial communications capability is available to link Dimension units with a terminal or a host computer. The following topics are discussed in this appendix:

- Communication types
- Data access methods

Communication types refer to handshaking techniques that permit communication to occur. Data access refers to the methods by which Dimension data can be viewed or changed.

The two communication types available for Dimension are:

- Terminal mode
- Network mode

In the terminal mode, a simple X-on/off handshaking procedure may, at the user's option, be used for data transfer. If used, this handshaking is enabled and disabled on a dedicated Dimension communications screen. In the network mode (also called Data Link Control, or DLC), data is transferred in blocks, and ASCII control characters are used to regulate data flow.

- ! **Important:** Network mode may be used only with a host computer.
- Terminal mode can be used by either a host computer or a terminal.

When a host computer is used, the user must write a software driver. This appendix provides sufficient information for a computer programmer to do so. It is presumed that the programmer already knows the requirements of the host computer's operating system.

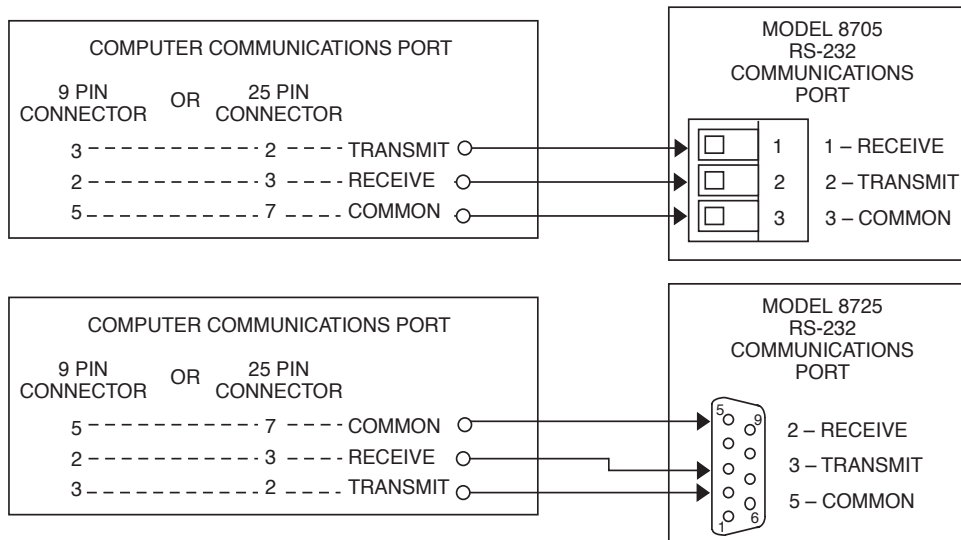
The Dimension may simultaneously use the RS-232 port and RS-422 port. RS-232 only supports one Dimension connected to a computer. RS-422 will support up to 31 Dimensions connected to a computer.

Electrical Connections

RS-232 Connections

RS-232 communication supports only one Dimension to a computer or CRT terminal. See Fig. B-1 for RS-232 connections. The Model 8725 Dimension uses a 9 pin D shell connector. A "NULL" modem cable may be used between a Model 8725 and computer. The maximum RS-232 cable length is 100 feet (30 meters).

Figure B-1.
RS-232 Connections



RS-422 Connections

RS-422 communication is useful in applications which require communicating over up to 4,000 feet (1219 meters) of interconnecting cable or when multiple Dimensions are connected in a distributed communications network. Up to 31 Dimensions may be connected in parallel to one computer. Fig. B-2 shows RS-422 communications from the computer to the first Dimension. Wire additional Dimensions to the first Dimension pin for pin and in a 'Daisy Chain' manner. See Fig. B-3.

Figure B-2.
RS-422 Connections

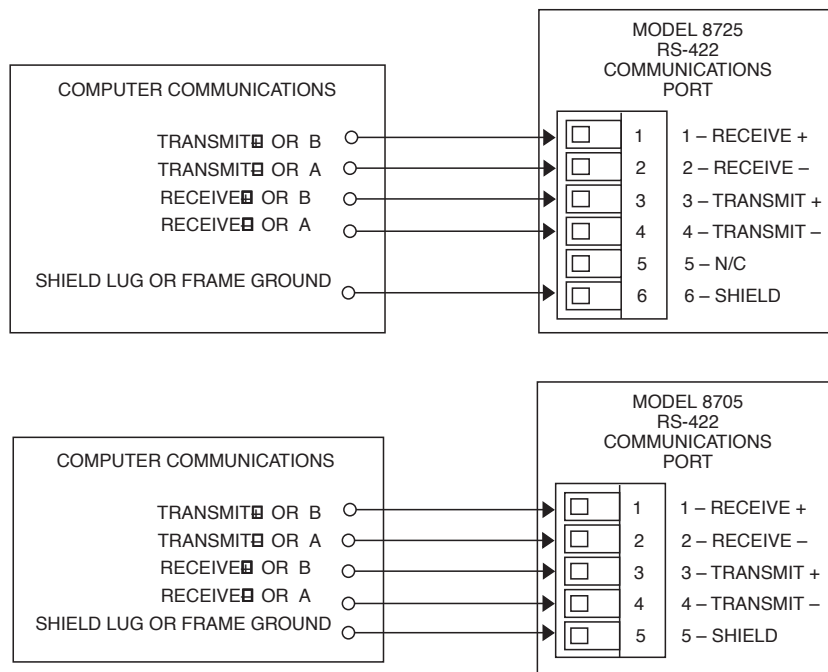
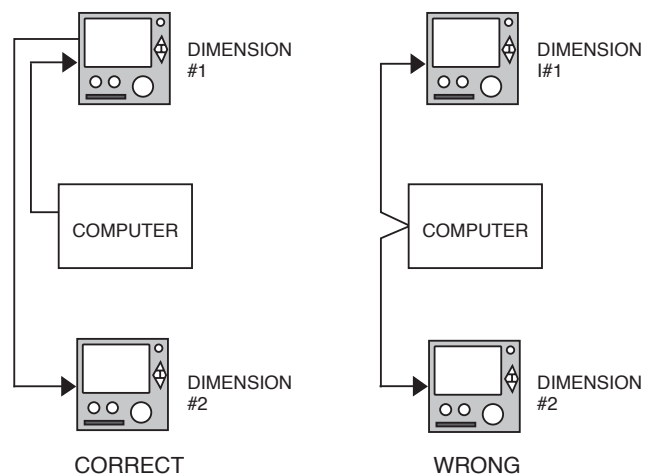


Figure B-3.
Daisy Chain Connections



Shield Connection

➔ *Note: The following shield connection procedure is recommended for typical Dimension applications. However, different connection procedures may be required in order to match a particular operating environment or host computer.*

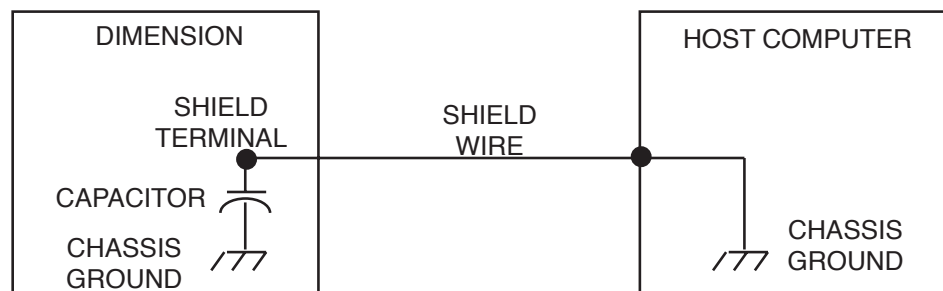
Use of the **SHIELD** connection minimizes pickup of electrical noise by the communications wiring.

The communications cable used for **RS-422** connections should include a metal sheath. This metal sheath is called the "shield." The shield surrounds the communications wires to keep out electrical interference.

In most cases, the shield functions best if DC current flow in the shield is blocked and AC voltages are effectively drained to ground. This is best achieved as follows:

1. To block DC current in the shield, connect the shield directly to chassis ground at the host computer and nowhere else.
2. To drain AC voltages to ground, connect the shield to the **SHIELD** terminal, which is connected to chassis ground through a capacitor (see Fig. B-4).
3. In a distributed network, connect the cable shield to chassis ground at the host computer and nowhere else.

Figure B-4.
SHIELD Terminal Wiring
Diagram



Terminal Mode

Terminal mode is a simple handshaking technique that, at the user's option, allows a terminal or a host computer to control the data flow from Dimension by using two ASCII control characters: X-on and X-off. Table B-1. illustrates different methods of documenting these control characters.

Table B-1. Dimension X-on/X-off Control Characters

Type	X-on	X-off
ASCII Name	DC1	DC3
HEX Number	11	13
Decimal Number	17	19

When Dimension receives an X-off character, it stops transmitting characters through the communication port. The Dimension remains in this state until it receives an X-on character, at which time it resumes transmission.

The X-on/off characters are the only special control characters used for terminal mode communication. A station address or checksum is not required.

When a computer terminal is connected to Dimension and is set for terminal mode, the X-on character is usually **Control Q** (press the **Control** key and then, while holding down the **Control** key, also press **Q**). The X-off character is usually **Control S** (press the **Control** key and, while holding down the **Control** key, also press **S**). Under most circumstances, a terminal does not need a software driver because most terminals have built-in X-on/off communications. **Control S** and **Control Q** are the only two control characters that are needed once the terminal is connected to the controller.

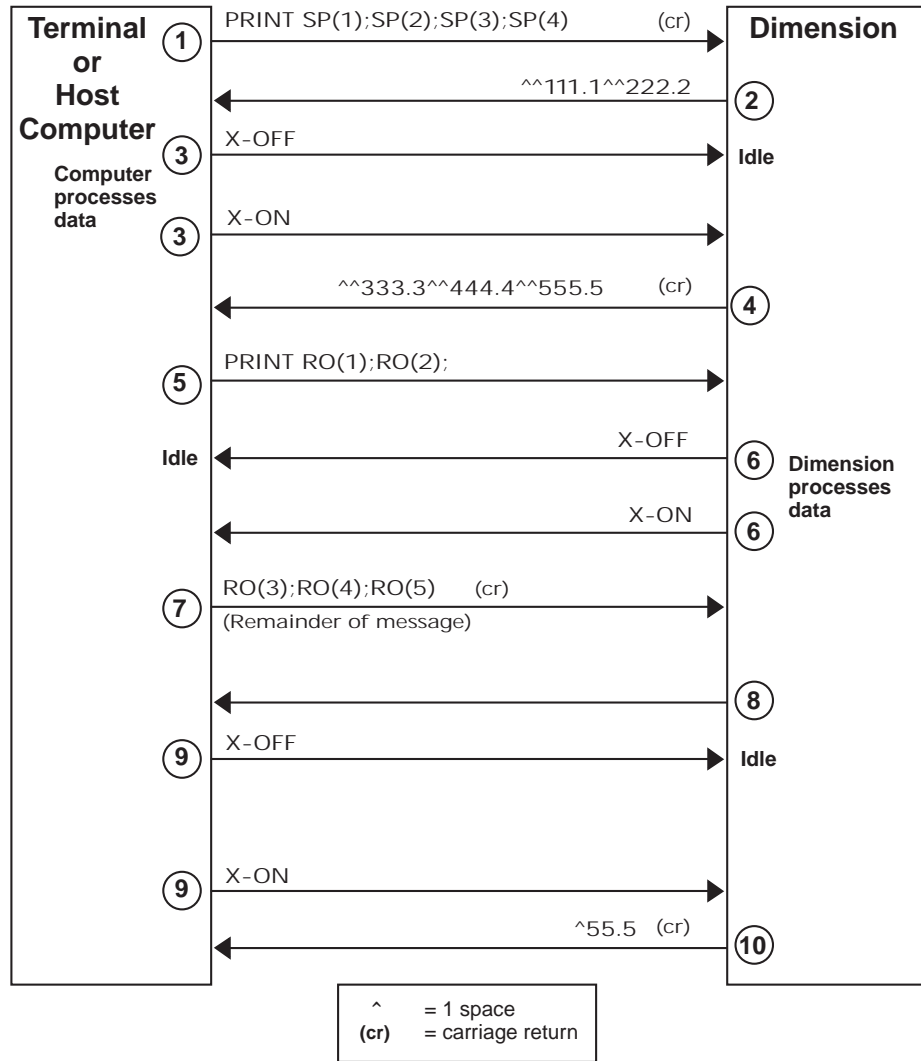
Fig. B-5 illustrates the communication sequence for a message exchanged using X-on/off communications.

Software Drivers

Most host computers have terminal drivers that support X-on/off. If not, the Network Mode topic on page B-9 contains the information needed to write a software driver using the network mode.

Communications

Figure B-5.
Terminal Mode (X-on/off) Communications Sequences



Error Checking

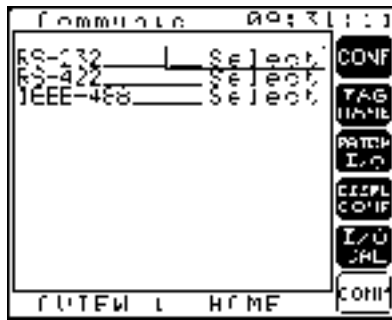
Terminal mode does not use error checking. The terminal or computer assumes that data is valid even if an error occurs during transmission. Error codes, however, are generated by Dimension for items such as syntax errors and number errors. (See "*LE PS(1)='R' (Run programmer 1 at current segment)*" on page B-83.)

Setting Up the Terminal Mode

Set up Dimension terminal mode as follows:

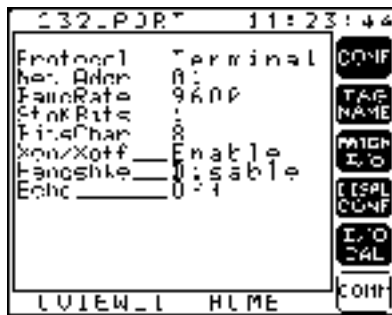
1. With the unit in the "FULL" access mode, use the diamond key to access the Communications Type Selection screen shown in Fig. B-6.

Figure B-6.
Communications Type Selection Screen



2. Use the diamond key to select the 232_Port Configuration screen for RS-232 communications or 422_Port Configuration screen for RS-422 communications. The screen shown in Fig. B-7 will display.

Figure B-7.
Communications Configuration Screen



3. Use the knob to select the Protocol field.
4. Select Terminal mode (no error checking).
5. Ignore the network address setting (network mode only), but set all other variables on the Communications Configuration screen to match the configuration of the terminal or host computer.

! **Important:** When the echo mode is enabled, all characters sent to the Dimension except X-on/off are immediately sent back to the terminal or computer. When using terminal mode, set the echo mode to **ON**. When the Dimension protocol is set for Network, set the echo to **OFF**.

6. Hardware handshake is not supported with Dimension. Set this parameter to Disable.
7. With the Dimension set for Warmstrt, remove AC power from Dimension.

8. Check that the host computer or terminal is powered up and electrically connected to the Dimension.
9. Apply AC power to the Dimension. The Dimension performs its startup routines, after which the terminal or host computer displays:
Dimension terminal mode v3.0
(c) 1997 Eurotherm Controls Inc.



***Note:** If there is no response on your terminal or host computer following step 9, it is possible that the settings given do not work with your application. Try other baud rates and settings, making sure that the settings on the Dimension Communications Setup screen are also set on your terminal or computer.*

Test the terminal mode by entering the following message at the terminal or computer keyboard:

PR TD (press **ENTER** or **RETURN**)

Response: 13:47:03 (current time)

Terminal Mode Data Access

Once the terminal mode is established on Dimension, the system variable data access method can be used as described under "System Variable Data Access" on page B-19.

Network Mode

Network mode (DLC) is a more advanced communication protocol intended for use with a host computer. This mode defines the sequence and format for communication with Dimension and consists of request and response messages, which are framed by control characters. These messages are transmitted as an entire block. If equipped with a user-developed software driver that understands the selected protocol, the computer dictates when Dimension transmits blocks of characters. Communication is always initiated by the computer; at no time can Dimension transmit unless the host computer requests data to be transmitted.

Software Driver

When using a host computer in the network mode, a software driver must be running on the host computer. The following topics explain how the protocol operates, and this information can be used to write a software communications driver for a host computer.

Setting Up the Network Mode

Set up Dimension network mode as follows:

1. With the unit in the "FULL" access mode, use the diamond key to access the Communications Type Selection screen shown in Fig. B-6.
2. Use the diamond key to select the 232_Port Configuration screen for RS-232 communications or 422_Port Configuration screen for RS-422 communications. The screen shown in Fig. B-7 will display.
3. Use the knob to select the Protocol field.
4. Select Network mode (error checking used).
5. Use the knob to select the Net_Addr field.
6. Select a network address number (01-99) for the Dimension. (This address is used only with Network selected and must be unique.)
7. Set other variables on the Communications Setup screen to match the configuration of the host computer. (Note that Xon/Xoff, Handshke and Echo have no effect in the network mode.)
8. With the controller set for Warmstrt, remove AC power from Dimension.
9. Check that the host computer is powered up and electrically connected to Dimension.
10. Connect AC power to Dimension. No response is displayed in either the Dimension or the host computer. Dimension simply performs its normal startup routines.
11. To test whether the network mode is set up correctly, start your software driver or run the 'DRIVER' program found on the Dimension Utility diskette:
 - A. On your computer, change to the DIMUTIL directory.
 - B. Type 'DRIVER' <CR>.
 - C. Select the COM port you are connected to.
 - D. Select the station address.
 - E. Select 2 seconds for Character Timeout.
 - F. Set "caps lock" ON on your keyboard.
 - G. At the Comnd: prompt, enter PR TD and press return.
Response: XX:XX:XX the Time of Day
 - H. Enter ! to end program or @ to start over.

If an error number appears rather than a number as a response to the command, refer to Table B-5. to identify the error that occurred.

➡ **Note:** *If there is no response on your host computer following step 11, the settings given may not work with your application. Try other baud rates and settings, making sure that the settings on the Dimension Communications Setup screen are also set to match your computer. Remember, the Dimension AC power must be cycled off and on with the Dimension in Warm Start for communications changes to take effect.*

Network Mode Request and Response Messages

Two types of messages are used with network mode protocol: request messages and response messages. The host computer initiates request messages; the Dimension answers with response messages. The Dimension always generates one message in response to a request message from the host computer. Both message types require ACK/NAK handshaking control characters for proper operation.

Network Mode Control Characters – Special handshaking control characters frame the messages between the computer and Dimension. The five control characters used in Dimension serial communication are:

STX(Start of TeXt: used to begin a request or response message)

ETX(End of TeXt: used to end a request or response message)

ENQ(ENQuiry: to prompt a response message from the Dimension)

ACK(ACKnowledge: to verify proper communication)

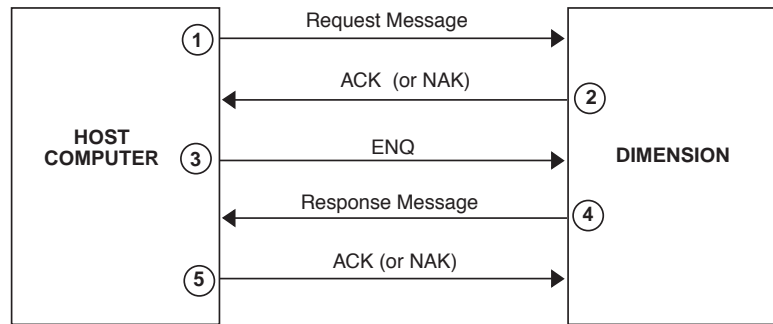
NAK(Not-AcKnowledge: to verify improper communication)

See Table B-3. for the numeric equivalents of these characters.

➡ **Note:** *Control characters may NOT be used in the station address or data field portions of a message.*

Network Mode Communication Sequence – Fig. B-8 illustrates the communication sequence for a message between a host computer and Dimension. Each communication begins with the transmission of a request message from the computer to Dimension, which then transmits either an ACK or a NAK control character back to the computer. An ACK confirms that the request message has been received without error. A NAK indicates that an error occurred and the request message should be retransmitted.

Figure B-8.
Dimension Serial
Communications Message
Sequence



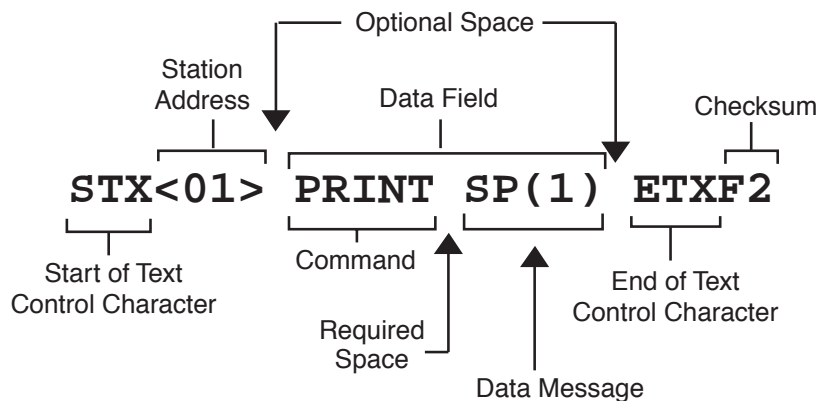
Once the request from the computer has been acknowledged, the Dimension is ready to transmit the response message. To request the response, the computer transmits an ENQ control character, upon receipt of which the Dimension transmits the response message and then waits for an ACK or NAK from the computer. If the Dimension receives a NAK, it automatically retransmits its response message up to four times. After the fourth NAK, the Dimension enters an idle state, waiting for another STX.

Once a response message is received and acknowledged by the computer, Dimension is ready to receive another request message.

Network Mode Request and Response Message Formats

Both request and response messages follow the same general format, as illustrated in Fig. B-9. The only difference is the structure of the data field. The following paragraphs explain in detail each element of the format.

Figure B-9.
Dimension Serial
Communications Request
Message Format



In the example of a message format shown in Fig. B-9, the checksum (F2) is calculated with the two optional spaces. If one optional space were removed from the message, the checksum would be D2; if both optional spaces were removed, the checksum would be B2.

Start of Text – The STX control character begins any request or response message. If communication with the Dimension is interrupted, a new STX must be sent by the computer to reestablish communication. An STX at any time during communication cancels the communication in progress and begins a new message sequence.

Station Address – A station address must be included in all request and response messages. Station addresses must be enclosed by brackets and be two digits long, e.g. <01>, <09>, <14>, etc. Spaces are not allowed between STX and the station address.

The station address for each Dimension is set using the Dimension Communications Setup screen Net_Addr field. Valid addresses are 00 through 99, even though only 31 stations may be used by an RS-422 network at a time.

Request Message Data Field – A request message requires that a *command* and a *data message* follow the station address. The request message format requires a space between the command characters and the data message (e.g. between PRINT and SP(1) in Fig. B-9).

Commands – The command is one of the two elements in the request message data field (Fig. B-9). There are five valid commands: PRINT, LET, LI (LIST), STAG and CTAG. The command determines whether the data message will monitor (PRINT) a system variable, change (LET) a system variable, display (LIST) all permissible parameters, show a display tagname (STAG) for a system variable, or change a display tagname (CTAG) for a system variable. If a command is not specified, LET is assumed. The Dimension will recognize the first two letters of PRINT, LET, and LIST, you can transmit LE for LET, PR for PRINT, or LI for LIST. However, STAG and CTAG cannot be abbreviated and LIST must be abbreviated. Also, a PRINT command may include comment text enclosed in quotes. (see "*System Variable Commands*" on page B-19). Comment text will be returned in the response message. For example:

Request message:

PR "The Dimension setpoint is"; SP(1)

Response message:

The Dimension setpoint is 10.0

Data Message – The data message is one of the two elements in the data field and can be up to 120 characters long. The data message always begins with a system variable that identifies the type of information (setpoint, deviation, process variable, etc.) to be changed or monitored. Also, several system variables may be included in the same data message by separating the system variables with semicolons, e.g. SP(1); PV(2); DV(2). The process of linking two or more system variables together in this manner is referred to as concatenation.

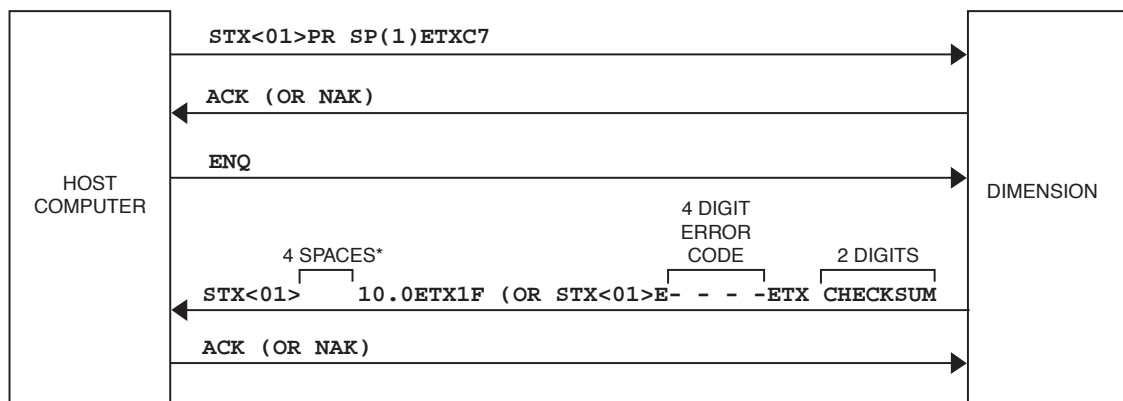
For a PRINT command, the data message consists of just the desired system variable(s) (see Fig. B-10). For example:

- To monitor the channel 2 (humidity) setpoint: PR SP(2).
- To monitor the channel 2 setpoint and deviation: PR SP(2); DV(2).

The PRINT command can also include comment text enclosed in quotes and placed in front of the system variable(s). For example:

- PR "The Ch 2 setpoint is: "; SP(2)
- PR "Ch 2 is: "; SP(2); "deviation 2 is: "; DV(2)
(one continuous message)

Figure B-10.
Dimension Serial Communications Message Format for a PRINT Command



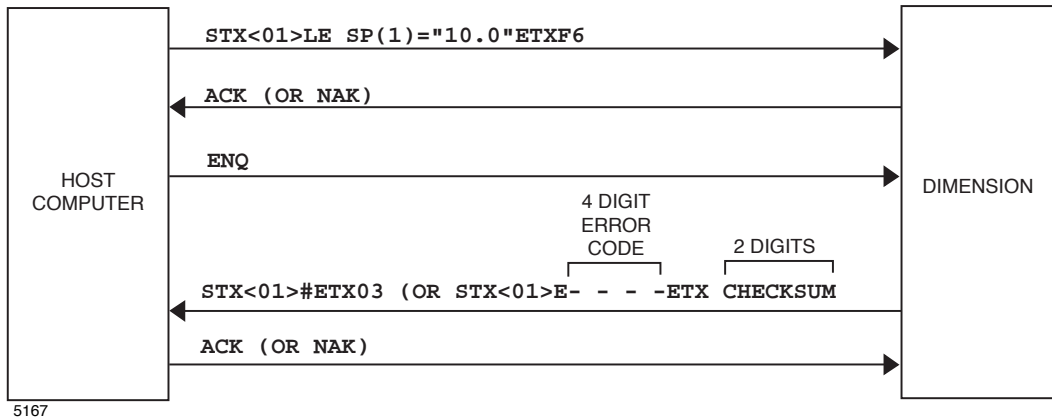
* This setpoint field length is eight characters. Since the setpoint data in this example is four digits long, Dimension transmits four spaces, followed by "10.0".

For a LET command, the data message contains the new system variable value, preceded by an equal sign and enclosed in single or double quotation marks (Fig. B-11). Note that a LET command cannot include comment text. For example, to enter a channel 2 (humidity) setpoint of 59.6:

LE SP(2)="59.6"

➡ *Note: The LE is optional; see example below.*

Figure B-11.
Dimension Serial Communications Message Format for a LET Command



5167

To enter a setpoint of 60.5 for both temperature and humidity channels:

SP(1)="60.5";SP(2)="60.5"

Since no command is specified in this example, LET is assumed.

Response Message Data Field – The format of the response message is determined by the type of request message command. If the response message is replying to a LET or CTAG command, a pound sign (#) or *error message* follows the station address. If the response message is replying to a PRINT, LIST, or STAG command, either a data message or an error message follows the station address. There are no spaces in a response message except those required by a system variable field length, error message, or spaces in the comment text included by the user. The pound sign (#), data messages, and error messages are explained under "System Variable Data Access" on page B-19.

End of Text Control Character

The ETX control character ends all messages. A space between the data field and the ETX message is optional. A space may not be used between the ETX and the checksum (a space after ETX will be interpreted as being part of the checksum).

Checksum

To verify the accuracy of each transmission, a two-character checksum is included at the end of each request and response message. The checksum is based on the 8-bit binary sum of all characters from the STX to and including the ETX. The checksum characters are the two least significant hex digits in the sum. Table B-2. illustrates the checksum generation process.

For information on hex-to-decimal and decimal-to-hex conversion, see Table B-3. .

Table B-2. Checksum Generation Example

Sample ASCII Message*: STX<01> LET SP(1)="56.3" ETX97		
MESSAGE CHARACTER	HEX EQUIVALENT	HEX RUNNING SUM
STX	02	02
<	3C	3E
0	30	6E
1	31	9F
>	3E	DD
SPACE	20	FD
L	4C	149
E	45	18E
T	54	1E2
SPACE	20	202
S	53	255
P	50	2A5
(28	2CD
1	31	2FE
)	29	327
=	3D	364
"	22	386
5	35	3BB
6	36	3F1
.	2E	41F
3	33	452
"	22	474
SPACE	20	494
ETX	03	497
		97 (CHECKSUM)

* The first and last spaces and the **T** in **LET** are optional; they are included in this example only for clarity. The only space required is between **LET** and **SP(1)**. This example message could also be written:

STX<01>LE SP(1)="56.3"ETX03

5168

Communications

Table B-3. ASCII Character Codes in Hexadecimal and Decimal

CHR	DEC	HEX	CHR	DEC	HEX	CHR	DEC	HEX
NUL	000	00	+	043	2B	V	086	56
SOH	001	01	,	044	2C	W	087	57
STX	002	02	-	045	2D	X	088	58
ETX	003	03	.	046	2E	Y	089	59
EOT	004	04	/	407	2F	Z	090	5A
ENQ	005	05	0	048	30	[091	5B
ACK	006	06	1	049	31	\	092	5C
BEL	007	07	2	050	32]	093	5D
BS	008	08	3	051	33	^	094	5E
HT	009	09	4	052	34	-	095	5F
LF	010	0A	5	053	35	.	096	60
VT	011	0B	6	054	36	a	097	61
FF	012	0C	7	055	37	b	098	62
CR	013	0D	8	056	38	c	099	63
SO	014	0E	9	057	39	d	100	64
SI	015	0F	:	058	3A	e	101	65
DLE	016	10	;	059	3B	f	102	66
DC1	017	11	<	060	3C	g	103	67
DC2	018	12	=	061	3D	h	104	68
DC3	019	13	>	062	3E	i	105	69
DC4	020	14	?	063	3F	j	106	6A
NAK	021	15	@	064	40	k	107	6B
SYN	022	16	A	065	41	l	108	6C
ETB	023	17	B	066	42	m	109	6D
CAN	024	18	C	067	43	n	110	6E
EM	025	19	D	068	44	o	111	6F
SUB	026	1A	E	069	45	p	112	70
ESCAPE	027	1B	F	070	46	q	113	71
FS	028	1C	G	071	47	r	114	72
GS	029	1D	H	072	48	s	115	73
RS	030	1E	I	073	49	t	116	74
US	031	1F	J	074	4A	u	117	75
SPACE	032	20	K	075	4B	v	118	76
!	033	21	L	076	4C	w	119	77
"	034	22	M	077	4D	x	120	78
#	035	23	N	078	4E	y	121	79
\$	036	24	O	079	4F	z	122	7A
%	037	25	P	080	50	{	123	7B
&	038	26	Q	081	51		124	7C
'	039	27	R	082	52	}	125	7D
(040	28	S	083	53	~	126	7E
)	041	29	T	084	54	DEL	127	7F
*	042	2A	U	085	55			

DEC= Decimal, HEX= Hexadecimal, CHR=Character, LF= Line Feed, FF= Form Feed,
CR= Carriage Return, DEL =Rubout.

Error Checking

The network mode contains a full set of error codes. Refer to Table B-5. for a listing and explanation of these codes.

Test Program

We have written the test program found in Fig. B-12 to test communications between a host computer and the Dimension using Network mode. This is not a complete communication driver program. This program was written using the Basic language and is compatible with QBASIC and GWBASIC.

Figure B-12.
Test Program

```

10 REM ***** FULL PROTOCOL COMMUNICATIONS TEST PROGRAM *****
20 OPEN "COM2:9600,N,8,1,RS,CS,DS" FOR RANDOM AS #1
30 CLS
40 PRINT #1, CHR$(2); "<01>SP(1)='10'"; CHR$(3); "F1";
50 GOSUB 100
60 PRINT #1, CHR$(2); "<01>PR SP(1)"; CHR$(3); "C7";
70 GOSUB 100
80 CLOSE #1
90 END
100 A = TIMER + 2
110 IF LOC(1) > 0 GOTO 140
120 IF TIMER > A GOTO 300
130 GOTO 110
140 B$ = INPUT$(1, #1)
150 IF ASC(B$) = 6 THEN PRINT "ACK"
160 IF ASC(B$) = 6 THEN PRINT #1, CHR$(5);
170 IF ASC(B$) = 21 GOTO 250
180 IF ASC(B$) = 2 THEN PRINT "STX"
190 IF ASC(B$) = 3 GOTO 220
200 IF ASC(B$) > 6 THEN PRINT B$;
210 GOTO 100
220 PRINT "ETX";
230 PRINT INPUT$(2, #1)
240 RETURN
250 PRINT "CONTROLLER SENT CHECKSUM ERROR (NAK)"
260 RETURN
300 PRINT "NO RESPONSE FROM CONTROLLER"
310 RETURN

```

The program changes loop 1 setpoint to 10 and then prints the setpoint. Correct output is displayed as follows:

```

ACK
STX
< Ø 1 > # ETX Ø 3
ACK
STX
< Ø 1 > ..... 1 Ø ETX Ø 1

```

Test Program output.

PC to Dimension Communications – Optimized for Speed

Variables are typically requested one at a time from Dimension. To cut some of the overhead associated with this method, a macro can be created in the PC and request a string of data with one command. The following is an example of a typical macro to implement this function.

1. Set to use the highest baud rate – 19.2 Kbaud
2. Use "SET" command to create macro in the PC. The following creates a macro which allows the user to transmit or receive blocks of information with one command.

PRINT Example

– Create Macro

```
SET $Z = "PR AI(1);AI(2);AI(3);AI(4);AI(5);AI(6)"
```

```
SET $Y = "PR AO(1);AO(2);AO(3);AO(4)"
```

```
SET $X = "PR CIN(1);CIN(2);CIN(3)"
```

– View Macro

```
SHOW $Z
```

```
SHOW $Y
```

```
SHOW $X
```

–Execute Macro

```
EX $Z
```

```
EX $Y
```

```
EX $X
```

LET Example

–Create Macro

```
SET $U = "LE PS(1)='H';PS(2)='H';PS(3)='H';PS(4)='H'"
```

–View Macro

```
SHOW $
```

–Execute Macro

```
EX $U
```

LIMITS: 255 characters per macro

\$ (U,V,W,X,Y,Z) only character available Macros.

System Variable Data Access

Access to Dimension parameters via terminal or network communications is accomplished using "System Variables". A System Variable is a two or three character abbreviation which corresponds to a certain parameter in the Dimension. In addition, many system variables take a numeric "subscript" surrounded by parentheses which specifies a specific member of a group of parameters. For example, the system variable for a setpoint is 'SP' and it takes a subscript which indicates the channel for which it is used. The full variable name would then be 'SP(x)' where x is a number from 1 to 3. Other system variables take no subscripts, and some take two. The System Variable Descriptions are listed in Table B-4. and described beginning on page B-26.

System Variable Commands

There are five valid commands for accessing system variables:

PRINT – Read a system variable

LET – Write a system variable

LIST – Display possible values for a system variable

STAG – Show the Tagname of a system variable

CTAG – Change the Tagname of a system variable

The PRINT, LET and LIST commands may be shortened to just two letters (PR, LE, LI), but the STAG and CTAG commands may not be truncated. All commands must be in capital letters, and if a command is not specified, LET is assumed. Each command is explained further below.

PRINT Commands

When using a PRINT command, you may either print only the desired system variable(s) or you may print both text and the desired system variables.

The following two examples use a PRINT command consisting of only the desired system variable(s):

- To monitor the channel 2 (humidity) setpoint: PR SP(2)
- To monitor the channel 2 setpoint and deviation: PR SP(2);DV(2)

The following two examples use a PRINT command with both text and the variables:

- PR "The channel 2 setpoint is:";SP(2)
- PR "Setpoint 2 is:";SP(2); "Deviation 2 is:";DV(2)

Response Message to a PRINT Command – When the response message is replying to a PRINT command, either a data message or an error message follows the station address. There are no spaces in a response message except those required by the system variable field length, error message, or spaces in comment text included by the user.

A response message contains the system variable information requested by a PRINT command. In addition, any comment text included in the request message will be returned in the response message.

Each system variable has a field length. Any unused characters in the field length are transmitted as spaces in the response message.

For example, the field length for the setpoint is eight characters. If the setpoint data requested is 10.0, the data response message would look like this:

```
^^^^10.0
```

where ^ = one space.

If the PRINT command is concatenated, the response data is also linked together. For example, the response data for setpoint (field length = 8), reverse output (field length = 8), loop status (field length = 1), and process variable (field length = 8) might look like this:

```
^^^^10.0^^^25.20A^^^35.63
```


If the PRINT command included text, the response data includes that text. For example, the response data for the message:

```
PR "The channel 2 setpoint is:";SP(2)
```

might look like this:

```
The channel 2 setpoint is:^^^^10.0
```

PRINT Request Error Messages – An error message indicates that the PRINT request message was unsuccessful. The error message is an E followed by a four-digit number. The number identifies what was wrong with the request message, e.g., a syntax error. All error messages are listed in Table B-5. .

LET Commands

For a LET command, the data message contains the new system variable value, preceded by an equals sign and enclosed in single or double quotation marks. A LET command cannot include comment text. Two examples are:

To enter a channel 2 (humidity) setpoint of 59.6:

```
LE SP(2)="59.6"
```

To enter a setpoint of 60.5 for temperature (channel 1) and 30 for humidity (channel 2):

```
SP(1)="60.5";SP(2)="30.0"
```

Response Message to a LET Command – If the response message is replying to a LET command, either a pound sign (#) or an error message follows the station address. A # response indicates that Dimension has correctly executed a LET command.

LET Request Error Message – An error message indicates that the LET request message was unsuccessful. The error message is an E, followed by a four digit number. The number identifies what was wrong with the request message, e.g. a syntax error. All error messages are listed in Table B-5. .

LIST Commands

The LIST command provides a useful “help” function that can be used during communications. To display all parameters (entries) on-screen that can be used with a PRINT or LET command (see Table B-4.), type LI followed by the 2- or 3-character system variable and the channel number in parentheses. The system will display all possible entries that can be used with the variable, capitalized correctly and separated by commas. For example, to list all parameters that can be used with the Loop Status system variable, type LI LS(1) and press **ENTER**. (This function works with system variables which have a finite number of possible settings.)

Table B-4. System Variables

Code	Title	Access	Code	Title	Access
AB	Abort Enable	R,W,P	CJ	Cold Junction Temperature	R,I
ACL	Access Level	R	CJE	Cold Junction Error	R,W,I
ACP	Accept Input	R,W,L	CM	Cascade Multiplier	R,W,L
ACY	Analog Output Time-Pro portioning Cycle Time	R,W,O	CO	Contact Out	R,C
AGP	Alarm Group	R,W,*	CP	Calculated Gain	R,L
AHY	Alarm Hysteresis	R,W,L	CPG	Current PID Group	R,L
AI	Analog Input	R,I	CRS	Calculated Reset	R,L
AK	Alarm Acknowledge	W,L,I	CS	Current Segment Number	R,P
AL	Alarm Screen Titles	R,W,T,S	CSP	Current Span Reference	R,W,I
ALL	All Input Range Names	R,*	CZE	Current Input Zero	R,W,I
ALT	Alarm Latch	R,W,L	DF	Digital Filter	R,W,I
AO	Analog Output	R,O	DH	Direct Output High Limit	R,W,L
AS	Alarm Status	R,L	DL	Direct Output Low Limit	R,W,L
ATD	Activate Time of Day Start	R,W,P	DO	Direct Output (in percent)	R,W,L
BC2	RS-232 Bits/Character	R,W	DOW	Day of Week Start	R,W,P
BC4	RS-422 Bits/Character	R,W	DT	Date	R,W
BD2	RS-232 Baud Rate	R,W	DV	Deviation	R,L
BD4	RS-422 Baud Rate	R,W	EO2	RS-232 Echo Mode	R,W
BEU	CLC Basic Engineering Units	R,W*	EO4	RS-422 Echo Mode	R,W
BF	Blend Factor	R,W,L	EU	Engineering Units	R,W,I
BHL	CLC Basic High Limit	R,W*	EV	Event Outputs	R,*
BLL	CLC Basic Low Limit	R,W*	FPC	Full Passcode	R,W
BLV	CLC Basic List Variable	R,W*	GAS	Global Alarm Status	R
BND	CLC Basic Number of Decimals	R,W*	GN	Gain	R,W,L
BNV	CLC Basic Numeric Variable	R,W*	HH2	RS-232 Hardware Handshaking	R,W
BSV	CLC Basic String Variable	R,W*	HH4	RS-422 Hardware Handshaking	R,W
CAG	Current Alarm Group	R,L	IAS	Input Alarm Status	R,I
CC	ReCycle Counter	R,P	IH	Input High Alarm	R,W,I
CCY	Contact Output Time-Pro portioning Cycle Time	R,W,C	IHL	Input High Limit	R,W,I
CD	Calculated Rate	R,L	IHY	Input Alarm Hysteresis	R,W,I
CEV	Current Events	R,P	IL	Input Low Alarm	R,W,I
CI	Configuration Image	R,W,*	ILL	Input Low Limit	R,W,I
CIN	Contact In	R,C	ILT	Input Alarm Latch	R,W,I
			LG	Logic Screen Titles	R,W,T,S

Code	Title	Access	Code	Title	Access
LND	Linear Analog Input Decimal Position	R,W,I	PS2	Programmer Start Segment 2 (Unused)	R,W,P
LNS	Linear Analog Input Span	R,W,I	PS3	Programmer Start Segment 3 (TOD Start)	R,W,P
LNZ	Linear Analog Input Zero	R,W,I	PS4	Programmer Start Segment 4 (Operator Start)	R,W,P
LP	Loop Screen Titles	R,W,T,S	PT	PID Type	R,W,L
LPC	Limited Passcode	R,W	PUD	Power Up Date	R
LS	Loop Status	R,W,L	PUT	Power Up Time	R
MC1	Macro Name #1	R,W,T	PV	Process Variable	R,L
MC2	Macro Name #2	R,W,T	QS2	RS-232 XON/XOFF Mode	R,W
MI	Memory Image	R,W,*	QS4	RS-422 XON/XOFF Mode	R,W
MR	Manual Reset	R,W,L	RH	Reverse Output High Limit	R,W,L
NA2	RS-232 Network Address	R,W	RI	Recipe Image	R,W,*
NA4	RS-422 Network Address	R,W	RL	Reverse Output Low Limit	R,W,L
NAI	Number of Analog Inputs	R	RO	Reverse Output (in percent)	R,W,L
NAO	Number of Analog Outputs	R	RS	Reset	R,W,L
NCI	Number of Contact Inputs	R	RSD	Response Speed	R,W,L
NCO	Number of Contact Outputs	R	RSS	Remote Start Segment	R,W,P
NL	Number of Loops	R	RT	Rate	R,W,L
NM2	RS-232 Protocol	R,W	SA	Setpoint Adjust	R,W,*
NM4	RS-422 Protocol	R,W	SB2	RS-232 Stop Bits	R,W
NP	Number of Programmers	R	SB4	RS-422 Stop Bits	R,W
NS	Number of Segments	R,P	SC	Scale	R,W,L
NTP	Number of Time-Proportioning Outputs	R	SG	Segment Data	R,W,*
OA	Output Action	R,W,L	SMD	System Mode	R,W
OC	Output Change	R,W,L	SP	Setpoint	R,W,L
OLS	Analog Output Linear Span	R,W,O	SPS	Setpoint Source	R,W,L
OLZ	Analog Output Linear Zero	R,W,O	SR	Sampling Rate	R,W,L
OT	Output	R,W,L	SS	System Status	R,W
OV	Overview Screen Titles	R,W,T,S	SY	System Screen Title	R,W,T
P1	Programmer 1 1 value	R,W,L	TCY	Time-Proportioning Cycle Time	R,W,C
P2	Programmer 2 Profile values	R,W,L	TD	Time-of-Day (TOD)	R,W
P3	Programmer 3 Profile values	R,W,L	TDD	Tuning Complete	R,L
P4	Programmer 4 Profile values	R,W,L	TDS	Time of Day Start	R,W,P
P5	Programmer 5 Profile values	R,W,L	TEN	Tuning Error Number	R,L
P6	Programmer 6 Profile values	R,W,L	TER	Tuning Error	R,L
P7	Programmer 7 Profile values	R,W,L	TL	Time Left in Segment (SEGtime)	R,P
P8	Programmer 8 Profile values	R,W,L	TLP	Time Left in Program	R,P
PDD	Power Down Date	R	TP	Time into Program	R,P
PDT	Power Down Time	R	TS	Tuning Status	R,W,L
PG	Programmer Screen Titles	R,W,T,S	TYP	Analog Input Type	R,W,I
PID	PID Group	R,W,*	UT	Engineering Units Type	R,W,I
PS	Programmer Mode Status	R,W,P			
PS1	Programmer Start	R,W,P			

Access Code Legend:

- R – Read
- W – Write
- T – Tagnames Only
- S – Screen Subscript
- L – Loop Subscript
- I –Input Subscript
- P –Programmer Subscript
- O –Output Subscript
- C –Contact Subscript
- * –Miscellaneous Subscript

STAG Commands

The Tagname of a system variable is the label used on the Dimension graphic operator interface for that variable. For example, the default tagname for the setpoint of loop 1 is "SetPntØ1". So PR SP(1) would read the value of the loop 1 setpoint, and STAG SP(1) would read the label used for the setpoint of loop 1. The STAG command is used to show the operator interface tagname used for a system variable. The data message contains the system variable abbreviation and a subscript if applicable. Multiple tagnames may be requested in a single command. The following examples show the use of the STAG command:

- To monitor the channel 1 process variable tagname: STAG PV(1)
- To monitor the channel 2 setpoint and process variable tagnames: STAG SP(2);PV(2)
- To monitor the first macro key title: STAG MC1

Response Message to a STAG Command – When the response message is replying to a STAG command, either a data message or an error message follows the station address. There are no spaces in a response message except those included in the system variable tagname, or the error message. Each response message contains the tagnames requested with no separators, and each tagname has a length of 8 characters. The following examples are the responses to the command examples above:

- Command:STAG PV(1)
Response: ProcVrØ1
- Command:STAG SP(2);PV(2)
Response: SetPntØ2ProcVrØ2
- Command: STAG MC1
Response: [START]^

where ^ = one space.

STAG Request Error Messages – An error message indicates that the STAG request message was unsuccessful. The error message is an E followed by a four digit number. The number identifies what was wrong with the request message, e.g., a syntax error. All error messages are listed in Table B-5. .

CTAG Commands

For a CTAG command, the data message contains the new system variable tagname, preceded by an equal sign and enclosed in single or double quotation marks. The new tagname **MUST** have a length of 8 characters. Some examples:

- To change the channel 1 process variable tagname:
CTAG PV(1)="Tempratr"
- To change the channel 2 setpoint tagname:
CTAG SP(2)"Target^^"
- To change the first macro key title:
CTAG MC1="RUN^^^^"

where ^ = one space. New tagnames will appear on Dimension graphic operator interface screens as soon as a new screen is displayed.

Response Message to a CTAG Command – If the response message is replying to a CTAG command, either a pound sign (#) or an error message follows the station address. A # response indicates that the Dimension has correctly executed the CTAG command.

CTAG Request Error Message – An error message indicates that the CTAG request message was unsuccessful. The error message is an E followed by a four digit number. The number identifies what was wrong with the request message, e.g., a syntax error. All error messages are listed in Table B-5. .

Reading & Changing the Status of the System Functions via Digital Communication

Four system functions are operated by Basic String Variables (BSV's). These BSV's can be read and changed using digital communications. The "Print" command is used to read the status and the LET command is used to get or change the status. The command format and BSV assignment is shown below.

Purpose: To monitor and set the state of the four main functions.

Use: PRINT format: PR BVS (#). where # is the system to be checked.
LET format: LE BSV (#) ="xxx", where # is the system and "xxx" is the status of the system, (Off, On, Pgm).

BSV(1)=Conditioning system event.

BSV(2)=Humidity system event.

BSV(3)=Auxiliary cool event.

BSV(4)=Purge event.

The values of these event can be "Off", "On", or "Pgm". Note that these values are case sensitive with the first character a capital.

Remarks: The format requires three characters and the first letter to be in the capital case.

Examples: Print request: PR BSV(!)
PRINT response: Off
LET request: LE BSV(1)="On"
LET response: #

System Variable Descriptions

Table B-4. lists all Dimension system variables that can be used for communications. The format and operation of all Dimension system variables are described on the following pages. The variables are organized alphabetically by system variable code, e.g., AB, ACL, ACP, etc. The following information is provided for each system variable:

System variable code: a two- or three-letter abbreviation for each system variable. For example, SP is the code for the "setpoint" system variable.


System variable name: For example, "setpoint."

Purpose: A description of the typical use of the variable.

Use: Describes valid commands (PR, LE, STAG, CTAG), field length, format.

Remarks: Information for the operation and use of the variable. If the format is complex, additional format information is provided.

Examples: Illustrate commands using the variable in a request message and the subsequent response message. When necessary, additional explanations are provided.

 **Note:** The caret symbol (^) indicates the position of spaces in commands and responses. In LET and CTAG commands, spaces must be typed after an alphabetical entry so the total number of characters matches the field length. Spaces are not required for numeric entries.

AB — Programmer Abort Enable

Purpose: To Enable or Disable programmer abort.

Use: PRINT format: PR AB(P), where P is programmer number

LET format:

LE AB(P)="Enable" – enables abort

LE AB(P)="Disable" – disables abort

Field length: 8

Remarks: Programmer number is 1 through 8. If an abort signal is sent to programmer P when it is running, it will immediately go to the abort segment (segment 0) if AB(P) is "Enable".

Examples: PRINT request: PR AB(1)
PRINT response: "Disable"
LET request: LE AB(1)="Enable"
LET response: #

ACL — Access Level

Purpose: To view controller security access (passcode) level.

Use: PRINT format: PR ACL

Field length: 8

Value: FULL,LIMITED,NONE

Remarks: Full access to all Operation and Configuration functions = FULL
Access to all Operation functions = LIMITED
View-only access to Operation screens = NONE

Examples: PRINT request: PR ACL
PRINT response: ^^^^FULL

ACP — Accept Input

Purpose: To view or change the status (accept, reject) of calculated PID autotuning parameters.

Use: PRINT format: PR ACP(L), where L is loop number.
 LET format: LE ACP(L)="x", where L is loop number, x is accept/reject status.
 Field length: 8
 Value: Yes, No

Remarks: Accept autotuning parameters = Yes
 Reject autotuning parameters = No

Examples: PRINT request: PR ACP(1)
 PRINT response: Yes^^^^
 LET request: LE ACP(1)="No"
 LET response: #

ACY — Analog Output Time-Proportioning Cycle Time

Purpose: To view or change the cycle time of an analog output that has been configured, via jumpers, to function as a time-proportioned output.

Use: PRINT format: PR ACY(L), where L is loop number.
 LET format: LE ACY(L)="XXX", where L is loop number, XXX is number of seconds.
 Field length: 3
 Value: XXX (1–255 seconds)

Remarks: Default (dead start) cycle time is 0 seconds (output functions as non-time-proportioned analog output). Range is 1 to 255 seconds.

Examples: PRINT request: PR ACY(1)
 PRINT response: 000
 LET request: LE ACY(1)="10"
 LET response: #

AGP — Alarm Group

Purpose: To view or change the settings of an entire alarm group.

Use: PRINT format: PR AGP(L,G), where L is loop number and G is group number.
 LET format: LE AGP(L,G)="A,B,C,D,E,F,G,H" where L is loop, G is group, and
 A is Hi Hi Process Variable alarm limit
 B is Lo Lo Process Variable alarm limit
 C is Hi Process Variable alarm limit
 D is Lo Process Variable alarm limit
 E is Hi Hi Deviation alarm limit
 F is Lo Lo Deviation alarm limit
 G is Hi Deviation alarm limit
 H is Lo Deviation alarm limit

Field length: 71

Remarks: Each Dimension control loop has five groups of values available to determine HiHiPvLmt, HiPvLmt, LoLoPvLmt, LoPvLmt, HiHiDevLmt, HiDevLmt, LoLoDevLmt, and LoDevLmt (group range is 1 to 5).

When changing those values, all eight positions must have a number even if all the numbers are the same value.

Examples: PRINT request: PR AGP(1,1)
 PRINT response:
 ^^3276.7,^-3276.7,^^3276.7,^-3276.7,
 ^^3276.7,^-3276.7,^^3276.7,^-3276.7
 LET request: LE AGP(1)="300,0,200,10,10,10,5,5"
 LET response: #

AHY — Alarm Hysteresis

Purpose: To view or change the hysteresis values of a Loop alarm.

Use: PRINT format: PR AHY(L), where L is loop number.

LET format: LE AHY(L)="A,B,C,D"

Where: A is Hi Hi & Lo Lo PV alarm hysteresis value
 B is Hi & Lo PV alarm hysteresis value
 C is Hi Hi & Lo Lo Deviation alarm hysteresis value
 D is Hi & Lo Deviation alarm hysteresis value

Field length: 35

Remarks: Hysteresis range for loop alarms is equal to the input range in use. All four values must be specified in a LET command.

Examples: PRINT request: PR AHY(1)
 PRINT response: ^^^^10.0,^^^^0.0,^^^^0.0,^^^^0.0
 LET request: LE AHY(1)="5,5,2,2"
 LET response: #

AI — Analog Input

Purpose: To view the current value of an analog input.

Use: PRINT format: PR AI(I), where I is analog input number.
Field length: 8
Value: XXXXXX.X (analog input value)

Examples: PRINT request: PR AI(1)
PRINT response: ^^^21.0

AK — Alarm Acknowledge

Purpose: This is a write-only parameter which acknowledges loop or analog input alarms.

Use: LET format: LE AK(X)="A" where X is a loop or input number
Field length: 8

Remarks: If desired, use alarm status (AS) to view current alarms for a loop, or input alarm status (IAS) to view alarms for an analog input. AK(X) will acknowledge all alarms for loop X and input X at the same time. If there is no loop X, then only alarms for input X will be acknowledged. The quotation marks should contain "A" or "ALL", although any characters will be accepted. There is no way to acknowledge some alarms for a loop and not others.

Examples: LET request: LE AK(1)="A" (for loop 1 and input 1)
LET response: #
LET request: LE AK(12)="A" (for input 12)
LET response: #

AL — Alarm Screen Titles

Purpose: To view or change the titles of Alarm Display screens.

Use: STAG format: PR AL(S), where S is alarm screen number.
CTAG format: LE AL(S)="X", where S is alarm screen number, X is screen title.
Field length: 8

Remarks: Default screen titles include LOOP_ALM, ALM_SET, INP_ALM, INP_ALM2.

Examples: STAG request: STAG AL(1)
STAG response: LOOP_ALM
CTAG request: CTAG AL(1)="ALM1^^^^"
CTAG response: #

ALL — All Input Range Names

- Purpose: To view all input ranges supported by an analog card.
 Use: PRINT format: PR ALL(A), where A is analog card number (1, 2 or 3).
 Field length: Variable
- Remarks: This command lists, in a string, all input ranges supported by the analog card.
- Examples: PRINT request: PR ALL(2)
 PRINT response: [RANGE NAME],[RANGE NAME],[RANGE NAME],....

ALT — Alarm Latch

- Purpose: To view or change whether an alarm condition for a given alarm will be latched or unlatched when an alarm occurs. Setting applies to both high and low alarms.
- Use: PRINT format: PR ALT(L), where L is loop number.
 LET format: LE ALT(L)="A,B,C,D", where:
 A is Hi Hi & Lo Lo PV alarm latch/unlatch value
 B is Hi & Lo PV alarm latch/unlatch value
 C is Hi Hi & Lo Lo Deviation alarm latch/unlatch value
 D is Hi & Lo Deviation alarm latch/unlatch value
 Field length: 35
 Value: Latch, Unlatch
- Remarks: Alarm condition latched = Latch
 Alarm condition unlatched = Unlatch
 All four values must be specified in a LET command.
- Examples: PRINT request: PR ALT(1)
 PRINT response:
 Latch^^^,Latch^^^,Unlatch^,Unlatch^
 LET request:
 LE ALT(1)="Unlatch^,Latch^^^,Unlatch^,Unlatch^"
 LET response:
 LE ALT(L)="Latch^^^,Latch^^^,Unlatch^,Unlatch^"

AO — Analog Output

- Purpose: To view or change the value of an analog output.
- Use: PRINT format: PR AO(O), where O is output number.
 Field length: 8
 Value: XXXXXX.X (0 to 100 percent)
- Remarks: Default value is 0 percent.
- Examples: PRINT request: PR AO(1)
 PRINT response: ^^^^25.0

AS — Alarm Status

Purpose: Read only parameter.

Use: PRINT format: PR AS(L), where L is loop number.
Field length: 8

Remarks: Will always return one of the following eight character strings.

Opensens	Open Sensor
Hi_Hi_PV	high high process variable alarm is active
Lo_Lo_PV	low low process variable alarm is active
High_PV	high process variable alarm is active
Low_PV	low process variable alarm is active
HiHi_Dev	high high deviation alarm is active
LoLo_Dev	low low deviation alarm is active
High_Dev	high deviation alarm is active
Low_Dev	low deviation alarm is active
No_Alarm	no alarm is active

Examples: PRINT request: PR AS(1)
PRINT response: No_Alarm

ATD — Activate Time of Day Start

Purpose: To view or change the enabled/ disabled status of clock-driven programmer time-of-day start.

Use: PRINT format: PR ATD(P), where P is programmer number.
LET format: LE ATD(P)="X", where X is Enable or Disable.
Field length: 8
Value: Enable, Disable

Remarks: Activate time-of-day programmer start = Enable
Disable time-of-day programmer start = Disable

Examples: PRINT request: PR ATD(1)
PRINT response: Disable^
LET request: LE ATD(1)="Enable^^"
LET response: #


BC2 — EIA-232 Bits/Character

Purpose: To view or change the number of bits per character used for EIA-232 communications.

Use: PRINT format: PR BC2
LET format: LE BC2="X", where X is 7 or 8.
Field length: 1
Value: 7, 8

Remarks: 7 bits per character = 7, 8 bits per character = 8

Examples: PRINT request: PR BC2
PRINT response: 7
LET request: LE BC2="8"
LET response: #

 *Note: Dimension must be powered down and then powered up again if serial communications parameters are changed.*


BC4 — EIA-422 Bits/Character

Purpose: To view or change the number of bits per character used for EIA-422 communications.

Use: PRINT format: PR BC4
LET format: LE BC4="X", where X is 7 or 8.
Field length: 1
Value: 7, 8

Remarks: 7 bits per character = 7, 8 bits per character = 8

Examples: PRINT request: PR BC4
PRINT response: 7
LET request: LE BC4="8"
LET response: #

 *Note: Dimension must be powered down and then powered up again if serial communications parameters are changed.*


BD2 — EIA-232 Baud Rate

Purpose: To view or change the baud rate used for EIA-232 communications.

Use: PRINT format: PR BD2
LET format: LE BD2="X", where X is baud rate.
Field length: 8
Value: 300, 1200, 2400, 4800, 9600, 19200

Remarks: Terminal and Dimension must have matching baud rate.

Examples: PRINT request: PR BD2
PRINT response: 4800^^^^
LET request: LE BD2="9600"
LET response: #

 *Note: Dimension must be powered down and then powered up again if serial communications parameters are changed.*


BD4 — EIA-422 Baud Rate

Purpose: To view or change the baud rate used for EIA-422 communications.

Use: PRINT format: PR BD4
LET format: LE BD4="X", where X is baud rate.
Field length: 8
Value: 300, 1200, 2400, 4800, 9600, 19200

Remarks: Terminal and controller must have matching baud rate.

Examples: PRINT request: PR BD4
PRINT response: 4800^^^^
LET request: LE BD4="9600"
LET response: #

 *Note: Dimension must be powered down and then powered up again if serial communications parameters are changed.*

BEU – CLC Basic Engineering Units

Purpose: To view or change the engineering units used with numeric CLC variables provided with a custom feature.

Use: PRINT format: PR BEU (V), where V is variable number (1–32).
LET format: LE BEU(V)-"XXX", where V is variable number and XXX is a three character string.

Field length: 3

Remarks: BEU(V) is used as the units for BNV(V), BHL(V), and BLL(V).
V can be any number from 1 to 32.

Examples: PRINT request: PR BEU (1)
PRINT response: F^^
LET request: LE BEU (23)="C^^"
LET response: #

BF – Blend Factor

Purpose: To view or change the blend factor value used with cascade control functions.

Use: PRINT format: PR BF(L), where L is loop number.
LET format: LE BF(L)="X", where X is the blend factor.
Field length: 8
Value: XXXXXX.X (-32767 to +32767)

Remarks: Blend factor is used as follows: (percent output * ratio + setpoint * blend factor) / scale factor

Examples: PRINT request: PR BF(1)
PRINT response: ^^^^^^5
LET request: LE BF(1)="6"
LET response: #

BHL — CLC Basic High Limit

Purpose: To view or change the highest acceptable value for the corresponding BNV variable.

Use: PRINT format: PR BHL(V), where V is variable number (1–32).
LET format: LE BHL(V)="X", where V is variable number, and X is a number.
Field length: 8

Remarks: BHL(V) is used to check operator input for BNV(V). No number higher than BHL(V) will be accepted for BNV(V). V can be any number from 1 to 32.

Examples: PRINT request: PR BHL(1)
PRINT response: ^^100.v
LET request: LE BHL(12)="250"
LET response: #

BLL — CLC Basic Low Limit

Purpose: To view or change the lowest acceptable value for the corresponding BNV variable (custom feature).

Use: PRINT format: PR BLL(V), where V is variable number (1–32).
LET format: LE BLL(V)="X", where V is variable number, and X is a number.
Field length: 8

Remarks: BLL(V) is used to check operator input for BNV(V). No number lower than BLL(V) will be accepted for BNV(V). V can be any number from 1 to 32.

Examples: PRINT request: PR BLL(1)
PRINT response: ^^10.0
LET request: LE BLL(12)="0"
LET response: #

BLV — CLC Basic List Variable

- Purpose:** To view or change the four selections available for the corresponding BSV variable (custom feature).
- Use:** PRINT format: PR BLV(V), where V is variable number (1–32).
 LET format:
 LE BLV(V)="AAAAAAAA,BBBBBBBB,CCCCCCCC,DDDDDDDD", where
 AAAAAAAAA,BBBBBBBB,CCCCCCCC,DDDDDDDD
 are 8-character strings.
 Field length: 35
- Remarks:** BLV(V) is the set of four string values which BSV(V) may take. The meaning and use of these strings is determined by the user via CLC programming. When changing BLV, the full 35 characters MUST be entered, with a comma as the 9th, 18th, and 27th characters. V can be any number from 1 to 32. PR BLV(V) is equivalent to LI BSV(V).
- Examples:** PRINT request: PR BLV(2)
 PRINT response: Run^^^^^^,Hold^^^^,Abort^^^,^^^^^^
 LET request:
 LE BLV(14)="Up^^^^^^,Down^^^^,Left^^^^,Right^^^^"
 LET response: #

BND — CLC Basic Number of Decimals

- Purpose:** To view or change the number of decimal places used for the corresponding BNV, BHL, and BLL (custom feature).
- Use:** PRINT format: PR BND(V), where V is variable number (1–32).
 LET format: LE BND(V)="X", where V is variable number, and X is a number.
 Field length: 1
- Remarks:** BND(V) is the number of decimal places which are shown when displaying BNV(V), BHL(V) and BLL(V). V can be any number from 1 to 32.
- Examples:** PRINT request: PR BND(1)
 PRINT response: 2
 LET request: LE BND(4)="1"
 LET response: #

BNV — CLC Basic Numeric Variable

Purpose: To view or change a CLC numeric variable (custom feature).

Use: PRINT format: PR BNV(V), where V is variable number (1–32).
 LET format: LE BNV(V)="X", where V is variable number, and X is a number.
 Field length: 8

Remarks: BNV(1) to BNV(32) are general-use numeric variables for CLC programming. Their meaning and use is determined by the user. BHL(V) and BLL(V) control acceptable input for BNV(V), and BND(V) and BEU(V) control how it is displayed on the Dimension operator interface.

Examples: PRINT request: PR BNV(4)
 PRINT response: ^^123.4
 LET request: LE BNV(1)="100"
 LET response: #

BSV — CLC Basic String Variable

Purpose: To view or change a CLC string variable (custom feature).

Use: PRINT format: PR BSV(V), where V is variable number (1–32).
 LET format: LE BSV(V)="XXXXXXXX", where V is variable number, and
 XXXXXXXX is an 8 character string.
 Field length: 8

Remarks: BSV(1) to BSV(32) are general-use string-type variables for CLC programming. Their meaning and use is determined by the user. BLV(V) controls the choices available on the Dimension operator interface for BSV(V).

Examples: PRINT request: PR BSV(1)
 PRINT response: Enable^^
 LET request: LE BSV(1)="Disable^^"
 LET response: #

CAG — Current Alarm Group

Purpose: To view the number of the currently active alarm group.

Use: PRINT format: PR CAG(P), where P is the programmer number.
 Field length: 8
 Value: 1–5

Remarks: Displays the number of the alarm group currently being utilized. The value of P is limited to the number of available programmers.

Examples: PRINT request: PR CAG(1)
 PRINT response: ^^^^^^3

CC — ReCycle Counter

- Purpose: To monitor the number of recycles left for a programmer.
- Use: PRINT format: PR CC(P), where P is programmer.
Value: XXX
Field length: 8
- Remarks: Dimension can recycle (repeat a segment sequence) up to 255 times, i.e., the recycle value can be from 0 to 255. When the recycle value reaches zero, the programmer ignores NEXT SEG and proceeds to SEG+1.
- Examples: PRINT request: PR CC(1)
PRINT response: ^^^^^10
(indicating that the programmer has 10 recycles left).

CCY — Contact Output Time-Proportioning Cycle Time

- Purpose: To view or change the cycle time of a contact output being used as a time-proportioned output.
- Use: PRINT format: PR CCY(C), where C is contact output number.
LET format: LE CCY(C)="XXX", where C is contact output number, XXX is number of seconds.
Field length: 3
Value: (1-255 seconds)
- Remarks: Default (dead start) cycle time is 0 seconds (output functions as non-time-proportioned contact output). Range is 1 to 255 seconds.
- Examples: PRINT request: PR CCY(1)
PRINT response: 000
LET request: LE CCY(1)="10"
LET response: #

CD — Calculated Rate

- Purpose: To view the rate calculated by the auto-tuning function.
- Use: PRINT format: PR CD(L), where L is loop number.
Field length: 8
Value: XXXXX.XX
- Remarks: Displays the rate that will be used if auto-tuning values are accepted.
- Examples: PRINT request: PR CD(1)
PRINT response: ^^^^2.00

CEV — Current Events

- Purpose:** To view or change the state into which a programmer is attempting to place the events.
- Use:** PRINT format: PR CEV(P), where P is programmer.
 LET format: LE CEV(P)="XXXXXXXXXXXX" where P is programmer, and XXXXXXXXXXXX is a 12 character string of event states.
 Field length: 12
 Value: If specific events (1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c) are displayed or included in the LET command, the programmer is set to turn those events on. If underscores "_" are displayed or included in the LET command, the programmer is set to turn events in those locations off.
- Remarks:** The LET command will only work if the programmer's status is Hold or PG_END. Otherwise CEV reverts to the event state programmed by the current segment. This variable is different from EV (Event Outputs) in that EV displays the actual current state of a single event, and CEV displays the state into which a specific programmer is programmed to set all events.
- Examples:** PRINT request: PR CEV(1)
 PRINT response: 1__4__7__b_
 LET request: LE CEV(1)="__45__8__c"
 LET response: #

CI — Configuration Image

- Purpose:** Allows uploading and downloading of EEPROM information (including operator interface and network configuration).
- Use:** PRINT format: PR CI(C,B), where C is CPU number, B is Block number.
 LET format: LE CI(C,B)="_____", where the information inside the quotation marks depends on the block being changed.
 Field length: Block 0 = 3, all other blocks = 128. All other blocks: 128
- Remarks:** CI is useful in network mode only when information is being uploaded or downloaded. Block 0 contains the number of blocks for the complete configuration image (e.g., 64). The information in all other blocks depends on the block being viewed. C is always 1.
- Examples:** PRINT request: PR CI(1,0)
 PRINT response: 64
 LET request: LE CI(1,1)="000AFC76ED..." (128 characters).
 LET response: #

CIN – Contact In

Purpose: To view the state of a contact input.

Use: PRINT format: PR CIN(C), where C is the contact input number

Field length: 8

Value: Tag name from one of the four possible sets of on/off labels for contacts.

Remarks: Each Dimension has four pairs of on/off labels. Each contact input or output can be configured to use one of these four sets of labels. So, the four pairs might be "On"/"Off", "Open"/"Closed", "True"/"False" and "Run"/"Stop" and the user can assign contact input 4 to read "Open" or "Closed". The CIN parameter returns the state of the contact input using the labels chosen for the particular contact.

Examples: PRINT request: PR CIN(3)

PRINT response: Off^^^^

CJ – Cold Junction Temperature

Purpose: To view the actual cold junction temperature for a thermocouple input.

Use: PRINT format: PR CJ(I), where I is the input number.

Field length: 8

Value: XXXXXX.X (temperature in degrees C)

Remarks: Thermocouple cold junction temperature is always expressed in degrees C.

Examples: PRINT request: PR CJ(1)

PRINT response: ^^^25.0

CJE – Cold Junction Error

Purpose: To view or change the cold junction error for a thermocouple input.

Use: PRINT format: PR CJE(I), where I is input number.

LET format: LE CJE(I)="X", where I is input number and X is amount of cold junction error compensation.

Field length: 8

Remarks: Cold junction error can be set from -10.0 to +10.0°C.

Examples: PRINT request: PR CJE(1)

PRINT response: ^^^^2.2

LET request: LE CJE(1)="2.5"

LET response: #

CM — Cascade Multiplier

Purpose: To view or change the multiplier used for cascade control with a remote setpoint.

Use: PRINT format: PR CM(L), where L is loop number.
 LET format: LE CM(L)="X", where L is loop number, X is the multiplier.
 Field length: 8
 Value: XXXXXXXX (-32767 to +32767)

Remarks: A decimal point in the multiplier is not permitted.

Examples: PRINT request: PR CM(1)
 PRINT response: ^^^^^^2
 LET request: LE CM(1)="3"
 LET response: #

CO — Contact Out

Purpose: To view the state of a contact output.

Use: PRINT format: PR CO(C), where C is the contact output number.
 Field length: 8
 Value: Tag name from one of the four possible sets of on/off labels for contacts.

Remarks: Each Dimension has four pairs of on/off labels. Each contact input or output can be configured to use one of these four sets of labels. So, the four pairs might be "On" / "Off", "Open" / "Closed", "True" / "False" and "Run" / "Stop" and the user can assign contact output 2 to read "Open" or "Closed". The CO parameter returns the state of the contact output using the labels chosen for the particular contact.

Examples: PRINT request: PR CO(2)
 PRINT response: Off^^^^^

CP — Calculated Gain

Purpose: To view the gain calculated by the auto-tuning function.

Use: PRINT format: PR CP(L), where L is loop number.
 Field length: 8

Remarks: Displays the gain that will be used if auto-tuning values are accepted.

Examples: PRINT request: PR CP(1)
 PRINT response: ^^^^1.00

CPG — Current PID Group

- Purpose: To view the currently active PID group for a programmer.
- Use: PRINT format: PR CPG(P), where P is the programmer number.
Field length: 8
Value: 1-5
- Remarks: Displays the number of the PID group currently being utilized. The value of P is limited to the number of available programmers.
- Examples: PRINT request: PR CPG(1)
PRINT response: ^^^^^^2

CRS — Calculated Reset

- Purpose: To view the reset calculated by the auto-tuning function.
- Use: PRINT format: PR CRS(L), where L is loop number.
Field length: 8
- Remarks: Displays the reset that will be used if auto-tuning values are accepted.
- Examples: PRINT request: PR CRS(1)
PRINT response: ^^^^1.00

CS — Current Segment Number

- Purpose: To monitor the current segment number for a given programmer.
- Use: PRINT format: PR CS(P), where P is programmer.
Field length: 8
- Remarks: Dimension units have segments 0 to 255 (0 to 749 with Extended Segments Option).
- Examples: PRINT request: PR CS(1)
PRINT response: ^^^^^^3
(The response indicates that the current segment for programmer 1 is segment 3.)

CSP — Current Span Reference

- Purpose: To view or change the span reference determined during calibration.
- Use: PRINT format: PR CSP(I), where I is analog input number.
LET format: LE CSP(1)="X", where I is analog input number,
X is span reference value.
Field length: 8
Value: XXXX.XXX
- Remarks: This reference number should not be changed unless an input channel is being recalibrated.
- Examples: PRINT request: PR CSP(1)
PRINT response: ^100.323
LET request: LE CSP(1)="100.356"
LET response: #

CZE — Current Input Zero

Purpose: To view or change the current input zero value determined during calibration.

Use: PRINT format: PR CZE(I), where I is analog input number.
 LET format: LE CZE(I)="X", where I is analog input number, X is input zero value.
 Field length: 8
 Value: XXXXXX.X

Remarks: This reference number should not be changed unless an input channel is being recalibrated.

Examples: PRINT request: PR CZE(1)
 PRINT response: ^^^^7.9
 LET request: LE CZE(1)="7.7"
 LET response: #

DF — Digital Filter

Purpose: To monitor or change the digital filter value of an analog input.

Use: PRINT format: PR DF(I), where I is input number.
 LET format: LE DF(I)="XXX", where I is input number, XXX is filter value.
 Field length: 8
 Value: XXX (0 to 100)

Remarks: The filter is used to "average" the input reading so that minor fluctuations are eliminated. The digital filter setting may be from 0 to 100; the higher the number, the greater the amount of filtering. When set to 0, the filtering feature is disabled.

Examples: PRINT request: PR DF(1)
 PRINT response: ^^^^^^0
 LET request: LE DF(1)="12"
 LET response: #

DH — Direct Output High Limit

Purpose: To monitor or change the direct output high limit (DH) for a control loop. This feature allows the user to set the maximum output percentage for direct outputs.

Use: PRINT format: PR DH(L), where L is loop.
 LET format: LE DH(L)="XXX.X"
 Field length: 8
 Value: XXX.X

Remarks: The high limit can be set anywhere from 00.0% to 100.0%. The direct output high limit can never be less than the direct output low limit.

Examples: PRINT request: PR DH(1)
 PRINT response: ^^100.00
 LET request: LE DH(1)="90.00"
 LET response: #

DL — Direct Output Low Limit

- Purpose:** To monitor or change the direct output low limit (DL) for a control loop. This feature allows the user to set the minimum output percentage for direct outputs.
- Use:** PRINT format: PR DL(L), where L is loop.
 LET format: LE DL(L)="XXX.X"
 Field length: 8
 Value: XXX.X
- Remarks:** The low limit can be set anywhere from 00.0% to 100.0%. The direct output low limit can never be greater than the direct output high limit.
- Examples:** PRINT request: PR DL(1)
 PRINT response: ^^^0.00
 LET request: LE DL(1)="10.0"
 LET response: #

DO — Direct Output (in percent)

- Purpose:** To monitor or change the direct output.
- Use:** PRINT format: PR DO(L), where L is loop.
 LET format: LE DO(L)="XXX.X"
 Field length: 8
 Value: XXX.X (0.00–100.00)
- Remarks:** The direct output may be monitored (displayed) at any time. Control outputs may be changed only when the control loop is in the manual mode.
- Examples:** PRINT request: PR DO(1)
 PRINT response: ^^^50.00
 LET request: LE DO(1)="75"
 LET response: #

DOW — Day of Week Start

- Purpose:** To view or change the day of the week on which a specific programmer is to start.
- Use:** PRINT format: PR DOW(P), where P is programmer number.
 LET format: LE DOW(P)="X", where P is programmer number, X is day of week.
 Field length: 8
 Value: See remarks.
- Remarks:** Enter days as follows: Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday.
- Examples:** PRINT request: PR DOW(1)
 PRINT response: Sunday^^
 LET request: LE DOW(1)="Monday^^"
 LET response: #

DT — Date

Purpose: To monitor or set the date.

Use: PRINT format: PR DT
 LET format: LE DT="MM/DD/YY"
 Field length: 8
 Value: MM/DD/YY

Remarks: MM = month, DD = day, YY = year.

Examples: PRINT request: PR DT 93
 PRINT response: 12/03/89
 LET request: LE DT="12/15/93"
 LET response: #

DV — Deviation

Purpose: To monitor the deviation.

Use: PRINT format: PR DV(L), where L is loop.
 Field length: 8
 Value: Depends on the range in use.

Remarks: The format for the deviation value matches the range in use.

Examples: PRINT request: PR DV(1)
 PRINT response: ^^^^21.3

E02 — EIA-232 Terminal Echo Mode

Purpose: To view or change the terminal echo mode (on/off) used during EIA-232 communications.

Use: PRINT format: PR E02
 LET format: LE E02="X", where X is echo mode.
 Field length: 8
 Value: See remarks.

Remarks: Echo mode on = ON
 Echo mode off = OFF


Examples: PRINT request: PR E02
 PRINT response: ON^^^^^^
 LET request: LE E02="OFF^^^^^^"
 LET response: #



Note: Dimension must be powered down and then powered up again if serial communications parameters are changed.

E04 — EIA-422 Terminal Echo Mode

- Purpose: To view or change the terminal echo mode (on/off) used during EIA-422 communications.
- Use: PRINT format: PR E04
 LET format: LE E04="X", where X is echo mode.
 Field length: 8
 Value: See remarks.
- Remarks: Echo mode on = ON
 Echo mode off = OFF
- Examples: PRINT request: PR E04
 PRINT response: ON^^^^^^
 LET request: LE E04="OFF^^^^^^"
 LET response: #

 *Note: Dimension must be powered down and then powered up again if serial communications parameters are changed.*

EU — Engineering Units

- Purpose: To monitor or select the engineering units of an analog input.
- Use: PRINT format: PR EU(I), where I is input.
 LET format: LE EV(I)="XXX", where I is input and XXX is the 3 alphanumeric characters.
 Field length: 3
- Remarks: F = Fahrenheit, C = Celsius, and XXX = Linear.
 Linear engineering units are user-configurable.
- Examples: PRINT request: PR EU(1)
 PRINT response: ^C^
 LET request: LE EV(3)="%^H"
 LET response: #

EV — Event Outputs

- Purpose: To monitor the actual state of the programmed event outputs.
- Use: PRINT format: PR EV(E), where E is event # (1–12).
 Field length: 8
 Value: "ON" or "OFF"
- Remarks: This reports the on/off state of the Dimension events. Each programmer may set them, but each event has only one state at a time.
- Examples: PRINT request: PR EV(1)
 PRINT response: OFF^^^^^^

FPC — Full Passcode

Purpose: To view or change the passcode required to change the security access level to "FULL" status.

Use: PRINT format: PR FPC
 LET format: LE FPC="X", where X is passcode.
 Field length: 8
 Value: XX-XX-XX

Remarks: The FULL passcode permits access to all system Operation and Configuration functions.

Examples: PRINT request: PR FPC
 PRINT response: 01-02-03
 LET request: LE FPC="01-03-05"
 LET response: #

GAS — Global Alarm Status

Purpose: To view global alarm status.

Use: PRINT format: PR GAS
 Field length: 8
 Value: No_Alarm, Alarm

Remarks: No active global alarm = No_Alarm
 Active global alarm = Alarm

Examples: PRINT request: PR GAS
 PRINT response: No_Alarm

GN — Gain

Purpose: To view or change the actual gain (proportional) value.


Use: PRINT format: PR GN(L), where L is loop.
 LET format: LE GN(L)="X", where L is loop and X is the new gain value.
 Field length: 8
 Value: XX.XX

Remarks: This system variable prints the actual gain value for a control loop. LET sets gain for the current PID group.

Examples: PRINT request: PR GN(1)
 PRINT response: ^^^3.50
 LET request: LE GN(2)="1.00"
 LET response: #


HH2 — EIA-232 Hardware Handshaking

- Purpose:** To view or change the hardware handshaking mode (on/off) used during EIA-232 communications.
- Use:** PRINT format: PR HH2
LET format: LE HH2="X", where X is handshaking mode.
Field length: 8
Value: Enable, Disable
- Remarks:** Hardware handshaking on = Enable, Hardware handshaking off = Disable
- Examples:** PRINT request: PR HH2
PRINT response: Disable^
LET request: LE HH2="Enable^^"
LET response: #

 *Note: Dimension must be powered down and then powered up again if serial communications parameters are changed.*

HH4 — EIA-422 Hardware Handshaking

- Purpose:** To view or change the hardware handshaking mode (on/off) used during EIA-422 communications.
- Use:** PRINT format: PR HH4
LET format: LE HH4="X", where X is handshaking mode.
Field length: 8
Value: Enable, Disable
- Remarks:** Hardware handshaking on = Enable, Hardware handshaking off = Disable
- Examples:** PRINT request: PR HH4
PRINT response: Disable^
LET request: LE HH4="Enable^^"
LET response: #

 *Note: Dimension must be powered down and then powered up again if serial communications parameters are changed.*

IH — Input High Alarm

- Purpose:** To view or change the input high alarm limit for an analog input.
- Use:** PRINT format: PR IH(I), where I is analog input number.
LET format: LE IH(I)="X", where I is analog input number, X is input high alarm limit value.
Field length: 8
- Remarks:** Range is dependent on input range.
- Examples:** PRINT request: PR IH(I)
PRINT response: ^^25.6
LET request: LE IH(1)="29.2"
LET response: #

IHL — Input High Limit

Purpose: To view or change the input high limit for an analog input.

Use: PRINT format: PR IHL(I), where I is analog input number.
 LET format: LE IHL(I)="X", where I is analog input number, X is input high limit value.
 Field length: 8

Remarks: The value range defaults to the range of the configured input type. The set-point and process variable will never exceed this limit. PID calculations and other limits are affected when a change is made.

Examples: PRINT request: PR IHL(I)
 PRINT response: ^^^^25.6
 LET request: LE IHL(1)="29.2"
 LET response: #

IHY — Input Alarm Hysteresis

Purpose: To view or change the hysteresis value of an input alarm.

Use: PRINT format: PR IHY(I), where I is input number.
 LET format: LE IHY(I)="XXXXXX.X", where I is input number, XXXXXX.X is hysteresis value.

Field length: 8
 Value: XXXXXX.X (-1392.8 to +1392.8)

Remarks: Range is same as input range (e.g. IHL ILL)

Examples: PRINT request: PR IHY(I)
 PRINT response: ^^^^11.0
 LET request: LE IHY(1)="13"
 LET response: #

IL — Input Low Alarm

Purpose: To view or change the input low alarm limit for an analog input.

Use: PRINT format: PR IL(I), where I is analog input number.
 LET format: LE IL(I)="X", where I is analog input number, X is input low alarm limit value.
 Field length: 8

Remarks: Range is dependent on input range.

Examples: PRINT request: PR IL
 PRINT response: ^^^^12.5
 LET request: LE IL(1)="14.2"
 LET response: #

ILL — Input Low Limit

- Purpose:** To view or change the input low limit for an analog input.
- Use:** PRINT format: PR ILL(I), where I is analog input number.
 LET format: LE ILL(I)="X", where I is analog input number, X is input low limit value.
 Field length: 8
- Remarks:** The value range defaults to the range of the configured input type. The set-point and process variable will never exceed this limit. PID calculations and other limits are affected when a change is made.
- Examples:** PRINT request: PR ILL(1)
 PRINT response: ^^^12.2
 LET request: LE ILL(1)="16.3"
 LET response: #

ILT — Input Alarm Latch

- Purpose:** To view or change whether an alarm condition for a given input will be latched or unlatched when an alarm occurs. Setting applies to both high and low alarms.
- Use:** PRINT format: PR ILT(I), where I is input number.
 LET format: LE ILT(I)="X", where I is input number, X is Latch or Unlatch.
 Field length: 8
 Value: Latch, Unlatch
- Remarks:** Alarm condition latched = Latch, Alarm condition unlatched = Unlatch
- Examples:** PRINT request: PR ILT(1)
 PRINT response: Latch^^^
 LET request: LE ILT(1)="Unlatch^"
 LET response: #

LG — Logic Screen Titles

- Purpose:** To view or change the titles of Logic Display screens.
- Use:** STAG format: STAG LG(S), where S is a logic screen number.
 CTAG format: CTAG LG(S)="X", where S is logic screen number, X is screen title.
 Field length: 8
- Remarks:** Default screen titles include LOGIC__1 through LOGIC__8.
- Examples:** STAG request: STAG LG(1)
 STAG response: LOGIC__1
 CTAG request: CTAG LG(1)="LOG1^^^^"
 CTAG response: #

LND — Linear Analog Input Decimal Position

Purpose: To view or change the decimal position for a linear analog input.

Use: PRINT format: PR LND(I), where I is analog input number.
 LET format: LE LND(I)="X", where I is analog input number, X is number representing decimal position.
 Field length: 1
 Value: decimal position (1–5)

Remarks: This variable can be changed only for linear analog inputs.

Examples: PRINT request: PR LND(1)
 PRINT response: 3
 LET request: LE LND(1)="2"
 LET response: #

LNS — Linear Analog Input Span

Purpose: To view or change the span for a linear analog input determined during calibration.

Use: PRINT format: PR LNS(I), where I is analog input number.
 LET format: LE LNS(I)="X", where I is analog input number, X is span value.
 Field length: 8
 Value: 0 to 100 percent

Remarks: This number defaults to 100 percent.

Examples: PRINT request: PR LNS(1)
 PRINT response: ^100.000
 LET request: LE LNS(1)="99.356"
 LET response: #

LNZ — Linear Analog Input Zero

Purpose: To view or change the input zero value for a linear analog input determined during calibration.

Use: PRINT format: PR LNZ(I), where I is analog input number.
 LET format: LE LNZ(I)="X", where I is analog input number, X is input zero value.
 Field length: 8
 Value: XXXXXX.X

Remarks: This value defaults to zero percent.

Examples: PRINT request: PR LNZ(1)
 PRINT response: ^^^^0.0
 LET request: LE LNZ(1)="2.2"
 LET response: #

LP – Loop Screen Titles

- Purpose: To view or change the titles of Loop Display screens.
- Use: STAG format: STAG LP(S), where S is a loop screen number.
 CTAG format: CTAG LP(S)="X", where S is loop screen number, X is screen title.
 Field length: 8
- Remarks: Default screen titles include LOOP__1 through LOOP__8.
- Examples: STAG request: STAG LP(1)
 STAG response: LOOP__1
 CTAG request: CTAG LP(1)="LP1^^^^^"
 CTAG response: #

LPC – Limited Passcode

- Purpose: To view or change the passcode required to change the security access level to "LIMITED" status.
- Use: PRINT format: PR LPC
 LET format: LE LPC="X", where X is passcode.
 Field length: 8
 Value: XX-XX-XX
- Remarks: The LIMITED passcode permits access to all system Operation functions.
- Examples: PRINT request: PR LPC
 PRINT response: 03-02-01
 LET request: LE LPC="05-03-01"
 LET response: #

LS – Loop Status

- Purpose: To set a control loop's mode of operation (auto or manual).
- Use: PRINT format: PR LS(L), where L is loop.
 LET format: LE LS(L)="M" or "A"
 Field length: 8
 Value: Manual or Auto
- Remarks: In the LET message above:
 A = Control loop is in the automatic mode.
 M = Control loop is in the manual mode.
 In the manual mode, the operator may adjust the output.
 No control action occurs in the manual mode.
 In the auto mode, the output is determined automatically by the PID feature.
- Examples: PRINT request: PR LS(1)
 PRINT response: Auto^^^^
 LET request: LE LS(1)="M"
 LET response: #

MC1 — Macro Name #1

Purpose: To view or change the tag name for Macro Key 1 displayed on the Operator Interface display.

Use: STAG format: STAG MC1
 CTAG format: CTAG MC1="X", where X is tag name.
 Field length: 8

Remarks: Default tag name is [START].

Examples: STAG request: STAG MC1
 STAG response: [START]^
 CTAG request: CTAG MC1="ON^^^^^^"
 CTAG response: #

MC2 — Macro Name #2

Purpose: To view or change the tag name for Macro Key 2 displayed on the Operator Interface display.

Use: STAG format: STAG MC2
 CTAG format: CTAG MC2="X", where X is tag name.
 Field length: 8

Remarks: Default tag name is [STOP].

Examples: STAG request: STAG MC2
 STAG response: [STOP]^
 CTAG request: CTAG MC2="OFF^^^^^^"
 CTAG response: #

MI — Memory Image

Purpose: Allows uploading and downloading of battery-backed memory (including patching and parameters).

Use: PRINT format: PR MI(C,B), where C is CPU number, B is Block number.
 LET format: LE MI(C,B)=" "
 where the information inside quotation marks varies according to the block being changed.
 Field length: Block 0 = 3, all other blocks = 128

Remarks: Memory image is useful in network mode only when downloading or uploading information.
 Block 0 contains the number of blocks for the complete memory image (e.g., 64). All other blocks contain different information depending on which block is viewed. C is always 1.

Examples: PRINT request: PR MI(1,0)
 PRINT response: 64
 LET request: LE MI(1,1)="000AFC76ED..." (128 characters)
 LET response: #

MR — Manual Reset

- Purpose: To monitor or change the manual reset (bias) value.
- Use: PRINT format: PR MR(L), where L is Loop.
 LET format: LE MR(L)="+-XX.X"
 Field length: 8
 Value: +-XX.X (-100.00 to 100.00)
- Remarks: Manual reset adds a fixed bias onto the control calculation when a control loop is in automatic mode.
- Examples: PRINT request: PR MR(1)
 PRINT response: ^^^^Ø.ØØ
 LET request: LE MR(1)="2.ØØ"
 LET response: #

NA2 — EIA-232 Network Address


- Purpose: To view or change the network address of the controller for EIA-232 communications in network mode.
- Use: PRINT format: PR NA2
 LET format: LE NA2="X", where X is network address number.
 Field length: 2
 Value: Network address 1 through 99.
- Remarks: This address (1–99) must match on both the host computer and the controller.
- Examples: PRINT request: PR NA2
 PRINT response: Ø1
 LET request: LE NA2="23"
 LET response: #



Note: Dimension must be powered down and then powered up again if serial communications parameters are changed.

NA4 — EIA-422 Network Address

- Purpose: To view or change the network address of the controller for EIA-422 communications in network mode.
- Use: PRINT format: PR NA4
 LET format: LE NA4="X", where X is network address number.
 Field length: 8
 Value: Network address 1 through 99.
- Remarks: This address (1–99) must match on both the host computer and the controller.
- Examples: PRINT request: PR NA4
 PRINT response: ^^^^^22
 LET request: LE NA4="23"
 LET response: #

 **Note:** Dimension must be powered down and then powered up again if serial communications parameters are changed.

NAI – Number of Analog Inputs

Purpose: To view the number of analog inputs used by the Dimension.

Use: PRINT format: PR NAI
Field length: 2

Remarks: Dimension accommodates up to six analog inputs per card, with a maximum of three cards per Dimension.

Examples: PRINT request: PR NAI
PRINT response: 04

NAO – Number of Analog Outputs

Purpose: To view the number of analog outputs used by the instrument.

Use: PRINT format: PR NAO
Field length: 2
Value: See remarks.

Remarks: Dimension accommodates up to four analog outputs per card, with a maximum of three cards per Dimension.

Examples: PRINT request: PR NAO
PRINT response: 04

NCI – Number of Contact Inputs

Purpose: To view the number of contact inputs used by the Dimension.

Use: PRINT format: PR NCI
Field length: 2
Value: See remarks.

Remarks: Dimension can have both local and remote contacts. This variable returns the total number of them configured as contact inputs.

Examples: PRINT request: PR NCI
PRINT response: 08

NCO – Number of Contact Outputs

Purpose: To view the number of contact outputs used by the Dimension.

Use: PRINT format: PR NCO
Field length: 2
Value: See remarks.

Remarks: Dimension can have both local and remote contacts. This variable returns the total number of them configured as contact outputs.

Examples: PRINT request: PR NCO
PRINT response: 04


NL — Number of Loops

- Purpose: To monitor the number of loops in a Dimension.
- Use: PRINT format: PR NL
Field length: 2
- Remarks: Dimension units have up to 8 PID control loops.
- Examples: PRINT request: PR NL
PRINT response: 02

NM2 — EIA-232 Network/Terminal Mode

- Purpose: To view or change the protocol (network/terminal) mode used for EIA-232 communications.
- Use: PRINT format: PR NM2
LET format: LE NM2="X", where X is mode.
Field length: 8
- Remarks: Terminal mode = Terminal


Network mode = Network
- Examples: PRINT request: PR NM2
PRINT response: Terminal
LET request: LE NM2="Network^"
LET response: #

 *Note: Dimension must be powered down and then powered up again if serial communications parameters are changed.*

NM4 — EIA-422 Network/Terminal Mode

- Purpose: To view or change the protocol (network/terminal) mode used for EIA-422 communications.
- Use: PRINT format: PR NM4
LET format: LE NM4="X", where X is mode.
Field length: 8
- Remarks: Terminal mode = Terminal

Network mode = Network
- Examples: PRINT request: PR NM4
PRINT response: Terminal
LET request: LE NM4="Network^"
LET response: #

 *Note: Dimension must be powered down and then powered up again if serial communications parameters are changed.*

NP — Number of Programmers

Purpose: To monitor the number of programmers in a Dimension.

Use: PRINT format: PR NP
Field length: 2
Value: XX

Remarks: Dimension units have up to 8 programmers.

Examples: PRINT request: PR NP
PRINT response: 04

NS — Number of Segments

Purpose: To monitor the number of segments in a Dimension.

Use: PRINT format: PR NS(P), where P is programmer.
Field length: 3
Value: XXX

Remarks: Dimensions have a single Table of 750 segments which are shared by all programmers. This parameter returns the same number no matter what P is.

Examples: PRINT request: PR NS(1)
PRINT response: 256

OA — Output Action

Purpose: To view or change the output action of a loop output for auto tuning.

Use: PRINT format: PR OA(L), where L is loop number.
LET format: LE OA(L)="X", where X is output action type.
Field length: 8
Value: Reverse, Direct

Remarks: Output action types = Reverse, Direct

Examples: PRINT request: PR OA(1)
PRINT response: Direct^^
LET request: LE OA(1)="Reverse"
LET response: #

OC – Output Change

- Purpose:** Used for auto tuning. To view or change an analog output value.
- Use:** PRINT format: PR OC(L), where L is loop number.
 LET format: LE OC(L)="X", where L is loop number and X is output change.
 Field length: 8
 Value: 0.0 to 100.0 percent
- Remarks:** To change an output value, type the percentage by which you wish to increase (bump) the output. The example below reflects an output increase from 5 to 15 percent.
- Examples:** PRINT request: PR OC(1)
 PRINT response: ^^^5.00
 LET request: LE OC(1)="10"
 LET response: #

OLS – Analog Output Linear Span

- Purpose:** To view or change the span for a linear analog output determined during calibration.
- Use:** PRINT format: PR OLS(O), where O is output number.
 LET format: LE OLS(O)="X", where O is loop number, X is span value.
 Field length: 8
 Value: XXXX.XXX (0.0 to 100.0 percent)
- Remarks:** This number defaults to 100 percent.
- Examples:** PRINT request: PR OLS(1)
 PRINT response: ^100.000
 LET request: LE OLS(1)="99.500"
 LET response: #

OLZ – Analog Output Linear Zero

- Purpose:** To view or change the zero value for a linear analog output determined during calibration.
- Use:** PRINT format: PR OLZ(O), where O is output number.
 LET format: LE OLZ(O)="X", where O is output number, X is zero value.
 Field length: 8
 Value: XXXX.XXX
- Remarks:** This number defaults to zero percent.
- Examples:** PRINT request: PR OLZ(1)
 PRINT response: ^^0.000
 LET request: LE OLZ(1)="1.400"
 LET response: #

OT — Output

Purpose: To view or change the output of a loop.

Use: PRINT format: PR OT(L), where L is loop number.
 LET format: LE OT(L)="X", where X is output value.
 Field length: 8
 Value: -100.0 to 100.0 percent

Remarks: Range is from -100 to +100 percent, where -100 is full reverse out and +100 is full direct out. In this way, the output of specific loops can be changed.

Examples: PRINT request: PR OT(1)
 PRINT response: ^^^^^100
 LET request: LE OT(1)="-100"
 LET response: #

OV — Overview Screen Titles

Purpose: To view or change the titles of Overview Display screens.

Use: STAG format: STAG OV(S), where S is overview screen number.
 CTAG format: CTAG OV(S)="X", where S is overview screen number, X is screen title.
 Field length: 8
 Value: See remarks.

Remarks: Default screen titles include OVIEW__1 through OVIEW__8.

Examples: STAG request: STAG OV(1)
 STAG response: OVIEW__1
 CTAG request: CTAG OV(1)="OVW1^^^^"
 CTAG response: #

P1 — Programmer 1 Profile Values

Purpose: To view or change a given profile value (loop setpoint) for the first programmer.

Use: PRINT format: PR P1(P), where P is profile number.
 LET format: LE P1(P)="X", where P is profile number and X is the setpoint value for profile P.
 Field length: 8
 Value: Any valid setpoint, depends on range in use.

Remarks: This parameter is the setpoint loop 1 will have if loop 1 is configured to take its setpoint from the selected programmer. A LET command for P1(P) will be overridden if programmer 1 is in the "Run" mode.

See the description of the SPS (Setpoint Source) parameter for more information on configuring a loop to take its setpoint from a given programmer.

Examples: PRINT request: PR P1(1)
 PRINT response: ^^^98.60
 LET request: LE P1(3)="100.7"
 LET response: #

P2 – Programmer 2 Profile Values

- Purpose:** To view or change a given profile value (loop setpoint) for the second programmer.
- Use:** PRINT format: PR P2(P), where P is profile number.
LET format: LE P2(P)="X", where P is profile number and X is the setpoint value for profile P.
Field length: 8
Value: Any valid setpoint, depends on range in use.
- Remarks:** This parameter is the setpoint loop 2 will have if loop 2 is configured to take its setpoint from the selected programmer. A LET command for P2(P) will be overridden if programmer 2 is in the "Run" mode.
- See the description of the SPS (Setpoint Source) parameter for more information on configuring a loop to take its setpoint from a given programmer.
- Examples:** PRINT request: PR P2(1)
PRINT response: ^^98.60
LET request: LE P2(3)="100.7"
LET response: #

P3 – Programmer 3 Profile Values

- Purpose:** To view or change a given profile value (loop setpoint) for the third programmer.
- Use:** PRINT format: PR P3(P), where P is profile number.
LET format: LE P3(P)="X", where P is profile number and X is the setpoint value for profile P.
Field length: 8
Value: Any valid setpoint, depends on range in use.
- Remarks:** This parameter is the setpoint loop 3 will have if loop 3 is configured to take its setpoint from the selected programmer. A LET command for P3(P) will be overridden if programmer 3 is in the "Run" mode.
- See the description of the SPS (Setpoint Source) parameter for more information on configuring a loop to take its setpoint from a given programmer.
- Examples:** PRINT request: PR P3(1)
PRINT response: ^^98.60
LET request: LE P3(3)="100.7"
LET response: #

P4 — Programmer 4 Profile Values

Purpose: To view or change a given profile value (loop setpoint) for the fourth programmer.

Use: PRINT format: PR P4(P), where P is profile number.
 LET format: LE P4(P)="X", where P is profile number and X is the setpoint value for profile P.
 Field length: 8
 Value: Any valid setpoint, depends on range in use.

Remarks: This parameter is the setpoint loop 4 will have if loop 4 is configured to take its setpoint from the selected programmer. A LET command for P4(P) will be overridden if programmer 4 is in the "Run" mode.

See the description of the SPS (Setpoint Source) parameter for more information on configuring a loop to take its setpoint from a given programmer.

Examples: PRINT request: PR P4(1)
 PRINT response: ^^98.60
 LET request: LE P4(3)="100.7"
 LET response: #

P5 — Programmer 5 Profile Values

Purpose: To view or change a given profile value (loop setpoint) for the fifth programmer.

Use: PRINT format: PR P5(P), where P is profile number.
 LET format: LE P5(P)="X", where P is profile number and X is the setpoint value for profile P.
 Field length: 8
 Value: Any valid setpoint, depends on range in use.

Remarks: This parameter is the setpoint loop 5 will have if loop 5 is configured to take its setpoint from the selected programmer. A LET command for P5(P) will be overridden if programmer 5 is in the "Run" mode.

See the description of the SPS (Setpoint Source) parameter for more information on configuring a loop to take its setpoint from a given programmer.

Examples: PRINT request: PR P5(1)
 PRINT response: ^^98.60
 LET request: LE P5(3)="100.7"
 LET response: #

P6 — Programmer Profile 6 Values

- Purpose:** To view or change a given profile value (loop setpoint) for the sixth programmer.
- Use:** PRINT format: PR P6(P), where P is profile number.
LET format: LE P6(P)="X", where P is profile number and X is the setpoint value for profile P.
Field length: 8
Value: Any valid setpoint, depends on range in use.
- Remarks:** This parameter is the setpoint loop 6 will have if loop 6 is configured to take its setpoint from the selected programmer. A LET command for P6(P) will be overridden if programmer 6 is in the "Run" mode.
- See the description of the SPS (Setpoint Source) parameter for more information on configuring a loop to take its setpoint from a given programmer.
- Examples:** PRINT request: PR P6(1)
PRINT response: ^^98.60
LET request: LE P6(3)="100.7"
LET response: #

P7 — Programmer 7 Profile Values

- Purpose:** To view or change a given profile value (loop setpoint) for the seventh programmer.
- Use:** PRINT format: PR P7(P), where P is profile number.
LET format: LE P7(P)="X", where P is profile number and X is the setpoint value for profile P.
Field length: 8
Value: Any valid setpoint, depends on range in use.
- Remarks:** This parameter is the setpoint loop 7 will have if loop 7 is configured to take its setpoint from the selected programmer. A LET command for P7(P) will be overridden if programmer 7 is in the "Run" mode.
- See the description of the SPS (Setpoint Source) parameter for more information on configuring a loop to take its setpoint from a given programmer.
- Examples:** PRINT request: PR P7(1)
PRINT response: ^^98.60
LET request: LE P7(3)="100.7"
LET response: #

P8 — Programmer 8 Profile Values

- Purpose: To view or change a given profile value (loop setpoint) for the eighth programmer.
- Use: PRINT format: PR P8(P), where P is profile number.
 LET format: LE P8(P)="X", where P is profile number and X is the setpoint value for profile P.
 Field length: 8
 Value: Any valid setpoint, depends on range in use.
- Remarks: This parameter is the setpoint loop 8 will have if loop 8 is configured to take its setpoint from the selected programmer. A LET command for P8(P) will be overridden if programmer 8 is in the "Run" mode.
- See the description of the SPS (Setpoint Source) parameter for more information on configuring a loop to take its setpoint from a given programmer.
- Examples: PRINT request: PR P8(1)
 PRINT response: ^^98.60
 LET request: LE P8(3)="100.7"
 LET response: #

PDD — Power Down Date

- Purpose: To view the date on which the Dimension was last powered down.
- Use: PRINT format: PR PDD
 Field length: 8
 Value: MM/DD/YY
- Remarks: Displayed in standard month/day/year format.
- Examples: PRINT request: PR PDD
 PRINT response: 08/15/90

PDT — Power Down Time

- Purpose: To view the time at which the Dimension was last powered down.
- Use: PRINT format: PR PDT
 Field length: 8
 Value: HH:MM:SS
- Remarks: Displayed in standard hours/minutes/seconds 24-hour time format.
- Examples: PRINT request: PR PDT
 PRINT response: 04:19:22

PG — Programmer Screen Titles

- Purpose:** To view or change the titles of Programmer Display screens.
- Use:** STAG format: STAG PG(S), where S is programmer screen number.
 CTAG format: CTAG PG(S)="X", where S is programmer screen number,
 X is screen title.
 Field length: 8
- Remarks:** Default screen titles include PROGRUN1 through PROGRUN8.
- Examples:** STAG request: STAG PG(1)
 STAG response: PROGRUN1
 CTAG request: CTAG PG(1)="PGM1^^^^"
 CTAG response: #

PID — PID Group

- Purpose:** To display or change the PID parameters for a specific loop and specific group number.
- Use:** PRINT format: PR PID(L,G), where L is loop, G is group.
 LET format: LE PID(L,G)"X,Y,Z", where L is loop, G is group, and X,
 Y, and Z are gain, reset and rate respectively.
 Field length: 26
- Remarks:** Each control loop has five different groups of PID parameters (1–5). If a loop has dual PID Type (PT), Group 1 is always the reverse-acting PID parameters and Group 2 is always the direct-acting PID parameters. If a loop has single PID Type, then Group 1 is used no matter whether reverse or direct acting. If the PID Type is programmed, then all 5 groups may be used, depending on the current programmer segment.
- Examples:** PRINT request: PR PID(1,1)
 PRINT response: ^^^^1.00,^^^^0.00,^^^^0.00
 LET request: LE PID(1,2)="3.5,2.5,.5"
 LET response: #

PS — Programmer Mode Status

Purpose: To monitor or change the programmer's mode of operation. This system variable can start or hold a programmer.

Use: PRINT format: PR PS(P), where P is programmer number.
 LET format: LE PS(P)="X", where P is programmer number and X is R, S, or H.
 Field length: 8

Remarks: The programmer mode system variable has two different formats: one for displaying (PRINT commands) and one for changing the status (LET commands).

Status display (PRINT commands):

Run^^^^ = Run mode.

Hold^^^^ = Hold mode.

PG_END^^ = Programmer in hold after completing a program run.

Programmer status change format (LET command):

R = PrgMode placed into run mode at current segment.

H = PrgMode placed into hold mode.

S = Starts a programmer run starting at segment in PS4 (operator start segment)

Dimension considers a start command from the network (using PS) as an Operator start. In order to start a programmer using the PS variable, the programmer must be configured for Operator start. The CONF_PRG screen for programmer X should have the OPStart field set to "OPStrt0X".

Examples: PRINT request: PR PS(1)
 PRINT response: Run^^^^
 LET request: LE PS(1)="S"
 LET response: #

PS1 — Programmer Start Segment 1 (Remote Start)

Purpose: This parameter is the same as RSS. See RSS.

PS2 — Programmer Start Segment 2

Purpose: Unused, reserved for future use.

PS3 — Programmer Start Segment 3 (Time-of-Day Start)

- Purpose: To view or change the state of the Time-of-Day Start segment of the selected programmer.
- Use: PRINT format: PR PS3(P), where P is programmer.
LET format: LE PS3(P)="X", where P is programmer number, X a segment number.
Field length: 8
Value: 0–255 (0–749 with Extended Segments Option)
- Remarks: This is the segment where the indicated programmer will start if started by the Time-of-Day Start feature.
- Examples: PRINT request: PR PS3(1)
PRINT response: ^^2
LET request: LE PS3(4)="20"
LET response: #

PS4 — Programmer Start Segment 4 (Operator Start)

- Purpose: To view or change the Operator Start segment for a particular programmer.
- Use: PRINT format: PR PS4(P), where P is programmer).
LET format: LE PS4(P)="X", where P is programmer number, X is a segment number.
Field length: 8
Value: 0–255 (0–749 with Extended Segments Option)
- Remarks: This is the segment where the indicated programmer will start if started by the operator from the Dimension graphic operator interface or by use of the command LE PS(P)="S".
- Examples: PRINT request: PR PS4(2)
PRINT response: ^^1
LET request: LE PS4(3)="5"
LET response: #

PT — PID Type

Purpose: To view or change the PID type for a control loop.

Use: PRINT format: PR PT(L), where L is loop.
 LET format: LE PT(L)="X", where X is S, D, P or A.
 Field length: 1

Remarks: PID type determines whether a control loop uses single PID (S), dual PID (D), programmed PID (P), or Select Adaptive (A).
 The **PID** system variable sets or views the values for each group, whereas the PT system variable sets or views the type of PID control.
 When programmed PID is selected, the programmer determines the PID group to be used for each segment in the programmer.

Examples: PRINT request: PR PT(1)
 PRINT response: Single
 LET request: LE PT(1)="D"
 LET response: #

Valid LET Requests: Single (or S)
 Dual (or D)
 Programmed (or P)
 AdaptSel (or A)

Dimension responses
 to PRINT requests: Single__
 Dual__
 Programmed
 AdaptSel

➡ *Note: Underscore represents trailing spaces.*

PUD — Power Up Date

Purpose: To view the date on which the Dimension was last powered up.

Use: PRINT format: PR PUD
 Field length: 8
 Value: MM/DD/YY

Remarks: Displayed in standard month/day/year format.

Examples: PRINT request: PR PUD
 PRINT response: 08/16/90

PUT — Power Up Time

Purpose: To view the time at which the Dimension was last powered up.

Use: PRINT format: PR PUT
 Field length: 8
 Value: HH:MM:SS

Remarks: Displayed in standard hours/minutes/seconds 24-hour time format.


Examples: PRINT request: PR PUT
 PRINT response: 06:21:01

PV — Process Variable

- Purpose: To monitor the process variable of a control loop.
- Use: PRINT format: PR PV(L), where L is loop.
Field length: 8
- Remarks: The format for the process variable value matches the range in use.
- Examples: PRINT request: PR PV(1)
PRINT response: ^^25.74


QS2 — EIA-232 Xon/Xoff Mode

- Purpose: To view or change the state of the Xon/Xoff mode for EIA-232 communications.
- Use: PRINT format: PR QS2
LET format: LE QS2="x", where x is the state of the Xon/Xoff communications mode.
Field length: 8
Value: Enable, Disable
- Remarks: Xon/Xoff enabled = Enable
Xon/Xoff disabled = Disable
- Examples: PRINT request: PR QS2
PRINT response: Disable^
LET request: LE QS2="Enable^^"
LET response: #

 **Note:** Dimension must be powered down and then powered up again if serial communications parameters are changed.

QS4 — EIA-422 Xon/Xoff Mode

- Purpose: To view or change the state of the Xon/Xoff mode for EIA-422 communications.
- Use: PRINT format: PR QS4
LET format: LE QS4="x", where x is the state of the XON/XOFF communications mode.
Field length: 8
Value: Enable, Disable
- Remarks: Xon/Xoff enabled = Enable
Xon/Xoff disabled = Disable
- Examples: PRINT request: PR QS4
PRINT response: Disable^
LET request: LE QS4="Enable^^"
LET response: #

 **Note:** Dimension must be powered down and then powered up again if serial communications parameters are changed.

RH — Reverse Output High Limit

Purpose: To monitor or change the reverse output high limit (RH). This feature allows the user to set the maximum output percentage for reverse outputs.

Use: PRINT format: PR RH(L), where L is loop.
 LET format: LE RH(L)="XXX.X", where L is loop and X is the limit.
 Field length: 8
 Value: XXX.X (0.0 to 100.0%)

Remarks: The reverse output high limit (RH) can never be less than the reverse output low limit (RL).

Examples: PRINT request: PR RH(1)
 PRINT response: ^^100.00
 LET request: LE RH(1)="50"
 LET response: #

RI — Recipe Image

Purpose: Permits uploading and downloading of recipe information.

Use: PRINT format: PR RI (C,B), where C is CPU number, B is block number.
 LET format: LE RI(C,B)="x", where C is CPU number, B is block number, x is the block of information being read or written.
 Field length: Block 0 = 3, all other blocks = 128
 Value: See remarks.

Remarks: RI is useful in the network mode only, when information is being uploaded or downloaded. This function creates a direct image of memory storing recipe values only (profiles, segment time, recycles, segments, alarm groups, PID groups). Block 0 contains the number of blocks for the complete recipe image. The information in all other blocks depends on the block being viewed. C is always 1.

Examples: PRINT request: PR QS(1,0)
 PRINT response: 128
 LET request: LE QS(1,1)="000AFC76ED..." (128 characters)
 LET response: #

RL — Reverse Output Low Limit

Purpose: To monitor or change the reverse output low limit (RL). This feature allows the user to set the minimum output percentage for reverse outputs.

Use: PRINT format: PR RL(L), where L is loop.
 LET format: LE RL(L)="XXX.X", where L is loop and X is the limit.
 Field length: 8
 Value: XXX.X (0.0 to 100.0%)

Remarks: The reverse output low limit (RL) can never be greater than the reverse output high limit (RH).

Examples: PRINT request: PR RL(1)
 PRINT response: ^^75.00
 LET request: LE RL(1)="25"
 LET response: #

RO — Reverse Output (in percent)

- Purpose:** To monitor or change the reverse output for a control loop.
- Use:** PRINT format: PR RO(L), where L is loop.
 LET format: LE RO(L)="XXX.X", where L is loop and X is the new reverse output value.
 Field length: 8
 Value: XXX.X (0.0 to 100.00%)
- Remarks:** The reverse output (RO) may be monitored (displayed) at any time. Control outputs may be changed only when the Dimension is in manual mode.
- Examples:** PRINT request: PR RO(1)
 PRINT response: ^^^^15.0
 LET request: LE RO(1)="20"
 LET response: #

RS — Reset

- Purpose:** To monitor or change the reset (integral) value currently in use for a control loop.
- Use:** PRINT format: PR RS(L), where L is loop.
 LET format: LET RS(1)="X", where L is loop and X is the reset value.
 Field length: 8
 Value: XX.XX
- Remarks:** This system variable prints or changes the actual reset value currently in use for a control loop.
- Examples:** PRINT request: PR RS(1)
 PRINT response: ^^^^1.00
 LET request: LE RS(1)="2.00"
 LET response: #

RSD — Response Speed

- Purpose:** To view or change the process speed (fast, medium, slow) for which you wish to auto-tune parameters.
- Use:** PRINT format: PR RSD(L), where L is loop number.
 LET format: LE RSD(L)="x", where L is loop number, x is response type.
 Field length: 8
 Value: Fast, Medium, Slow
- Remarks:** Before auto-tuning a loop, select the response speed of the process you are controlling. A process which responds quickly to a change in control output will be tuned differently from a process which responds slowly.
- Examples:** PRINT request: PR RSD(1)
 PRINT response: Fast^^^^
 LET request: LE RSD(1)="Medium^^"
 LET response: #

RSS — Remote Start Segment

Purpose: To change or view the remote start segment for a given programmer.

Use: PRINT format: PR RSS(P), where P is programmer.
 LET format: LE RSS(P)="X", where P is programmer and X is a segment number.
 Field length: 8
 Value: 0–255 (0–749 with Extended Segments Option)

Remarks: Each programmer has one Remote Start Segment (RSS) number. This is the segment where the programmer will start running if a remote start is executed. Note that a programmer start done via the network with the PS parameter is considered an Operator start, NOT a Remote start.

Examples: PRINT request: PR RSS(1)
 PRINT response: ^^^^^^3
 LET request: LE RSS(1)="2"
 LET response: #

RT — Rate

Purpose: To monitor or change the rate (derivative) value for a control loop.

Use: PRINT format: PR RT(L), where L is loop.
 LET format: LE RT(L)="X", where L is loop number, X is the new rate value.
 Value: XX.XX

Remarks: This system variable prints or changes the actual rate value currently in use for a control loop.

Examples: PRINT request: PR RT(1)
 PRINT response: ^^^^1.5Ø
 LET request: LE RT(4)="1.1"
 LET response: #

SA — Setpoint Adjust

Purpose: Displays or changes the biasing of a profile setpoint.


Use: PRINT format: PR SA(L), where L is loop number.
 LET format: LE SA(L)="X", where L is loop number, X is bias value.
 Field length: 8

Remarks: This number is added to the profile setpoint of a loop when its setpoint source is "Programr". The setpoint of the loop will be equal to its profile value in the current segment plus the setpoint adjust. This allows a whole setpoint program to be shifted up or down by a fixed constant amount.

Examples: PRINT request: PR SA(1)
 PRINT response: ^^^^1Ø.Ø
 LET request: LE SA(1)="1.5"
 LET response: #


SB2 — EIA-232 Stop Bits

- Purpose: To view or change the number of stop bits used during EIA-232 communications.
- Use: PRINT format: PR SB2
LET format: LE SB2="X", where X is the number of stop bits.
Field length: 1
Value: 1 or 2
- Remarks: Set number of stop bits to 1 or 2. This must match your host computer's setting.
- Examples: PRINT request: PR SB2
PRINT response: 2
LET request: LE SB2="1"
LET response: #

 **Note:** Dimension must be powered down and then powered up again if serial communications parameters are changed.

SB4 — EIA-422 Stop Bits

- Purpose: To view or change the number of stop bits used during EIA-422 communications.
- Use: PRINT format: PR SB4
LET format: LE SB4="X", where X is the number of stop bits.
Field length: 1
Value: 1 or 2
- Remarks: Set number of stop bits to 1 or 2. This must match your host computer's setting.
- Examples: PRINT request: PR SB4
PRINT response: 2
LET request: LE SB4="1"
LET response: #

 **Note:** Dimension must be powered down and then powered up again if serial communications parameters are changed.

SC — Scale

Purpose: To view or change the scale value used with cascade control functions.

Use: PRINT format: PR SC(L), where L is loop number.
 LET format: LE SC(L)="x", where L is loop number, x is the scale value.
 Field length: 8
 Value: XXXXXX.X (0 to 32767)

Remarks: Scale is used as follows: (percent output * ratio + setpoint * blend factor) / scale factor. How this value is used is configurable by the user.

Examples: PRINT request: PR SC(1)
 PRINT response: ^^^^^^6
 LET request: LE SC(1)="9"
 LET response: #

SG — Segment Data

Purpose: To view or change the values of a programmer segment.

Use: PRINT format: PR SG(P,S), where P is programmer, S is segment.
 LET format: LE SG(P,S) = ". . .", where P is Programmer, S is segment, and all segment values are listed between the quotes.
 Field length: 105

Remarks: This variable views or changes an entire segment in the controller's segment Table. Valid segment numbers are 0-749. The programmer number is irrelevant since all programmers use the same segment Table, but it must be specified in the command.

The format is as follows: P1,P2,P3,P4,P5,P6,P7,P8,HH:MM:SS,EEEEEEEEEEEE,CCC,NNN,A,P where

P1-P8 = The 8 profile setpoints. There are always 8 even if the controller has less than 8 loops. P1 corresponds to loop1, P2 to loop2, and so on. With the LET command, all 8 MUST be specified – use zeros for unused profiles.

HH:MM:SS = Segment time in Hours:Minutes:Seconds

EEEEEEEEEEEE = Events. All 12 MUST be specified – use 1-9, a, b, and c for events which should be on and underscores “_” for events which should be off.

CCC = Recycles (0-255)

NNN = Next segment

A = Alarm group

P = PID group

When using a LET command, ALL 14 parameters must be specified.

Examples: PRINT request: PR SG(1,15)
 PRINT response: (one continuous message):
 ^^^150.0,^^^200.0,^^^45.0,^^^0.0,^^^0.0,^^^0.0,
 ^^^0.0,^^^0.0,00:05:00,___34_6____b_,000,016,1,1
 LET request: (one continuous message):
 LE SG(1,1)="25,30,44.5,100,0,0,0,0,00:10:00, 1___6____,0,2,3,3"
 LET response: #

SMD – System Mode

- Purpose: To view or change the mode in which the Dimension will restart after the next power-down.
- Use: PRINT format: PR SMD
LET format: LE SMD="x", where x is mode.
Field length: 8
Value: WarmStrt, ColdStrt
- Remarks: WarmStrt = All current settings saved.
ColdStrt = Calibration settings saved, all others reset to factory defaults. For the LET command, "W" and "C" may be used.
- Examples: PRINT request: PR SMD
PRINT response: ColdStrt
LET request: LE SMD="WarmStrt"
LET response: #

SP – Setpoint

- Purpose: To monitor or change the setpoint.
- Use: PRINT format: PR SP(L), where L is loop.
LET format: LE SP(L)="+-XXX.", where L is loop.
Field length: 8
Value: Depends on the range in use.
- Remarks: The format for the setpoint (SP) value matches the range in use.
The setpoint may be changed only when the setpoint source (SPS) is set to Operator.
- Examples: PRINT request: PR SP(1)
PRINT response: ^^^54.0
LET request: LE SP(1)="87"
LET response: #

SPS — Setpoint Source

Purpose: To view or change the setpoint source for a loop.

Use: PRINT format: PR SPS(L), where L is loop.
 LET format: LE SPS(L)="X", where L is loop, X is O, R or P.
 Field length: 8
 Value: Operator, Remote, Programr

Remarks: This parameter determines whether a control loop receives its setpoint from the operator, a programmer, or a remote source. A calculated setpoint is considered remote.
 To configure a loop to take its setpoint from a given programmer, the Setpoint Source (SPS) must be set to Programr, and the loop's Remote Setpoint Source must be set to ProgramX where X is the programmer you would like to control the loop. On the Dimension graphic operator interface, the Setpoint Source appears as SptSrcØX in the window accessed by clicking on the LpModeØX field of the main screen for loop X. The Remote Setpoint Source appears as RemSPsrc on the CONF_LOP screen. There is no network parameter available to set the Remote Setpoint Source.

Examples: PRINT request: PR SPS(1)
 PRINT response: Operator
 LET request: LE SPS(1)="P"
 LET response: #

SR — Sampling Rate

Purpose: To view or change the sample rate of a loop input for tuning.

Use: PRINT format: PR SR(L), where L is loop number.
 LET format: LE SR(L) = "x", where L is loop number, x is sampling rate.
 Field length: 8
 Value: Range is 0 to 1000.

Remarks: The number of input samples taken should be at least 200. The sampling rate is in multiples of 200 milliseconds.

Examples: PRINT request: PR SR(1)
 PRINT response: ^^^^2Ø2
 LET request: LE SR(1)="22Ø"
 LET response: #

SS – System Status

- Purpose:** To indicate whether control functions are running. Also, to download information to Memory Image (MI) or Configuration Image (CI).
- Use:** PRINT format: PR SS
 LET format: LE SS="X"
 Field length: 1
 Value: R, I
- Remarks:** System Status (SS) indicates the current status of the functions:
 R indicates functions are running (functioning normally).
 I indicates idle or off line. When downloading Memory Image (MI) or Configuration Image (CI), change the SS status to I.
 When the downloading process is finished, change the SS status back to R.
- Examples:** PRINT request: PR SS
 PRINT response: I
 LET request: LE SS="R"
 LET response: #

SY – System Screen Title

- Purpose:** To view or change the title of the system (UNIT) screen.
- Use:** STAG format: STAG SY
 CTAG format: CTAG SY="x", where x is screen title.
 Field length: 8
 Value: See remarks.
- Remarks:** Default screen title is UNIT.
- Examples:** STAG request: STAG SY
 STAG response: UNIT^^^^
 CTAG request: CTAG SY="UNIT1^^^^"
 CTAG response: #

TCY – Time-Proportioning Cycle Time

- Purpose:** To view or change the time-proportioning cycle time of a relay contact output on an analog I/O card.
- Use:** PRINT format: PR TCY(C), where C is contact number.
 LET format: LE TCY(C)="XXX", where C is contact number, XXX is number of seconds.
 Field length: 3
 Value: XXX (1–255 seconds)
- Remarks:** Default (dead start) cycle time is 0 seconds (output functions as non-time-proportioned contact output). Range is 1 to 255 seconds.
- Examples:** PRINT request: PR TCY(1)
 PRINT response: 001
 LET request: LE TCY(1)="15"
 LET response: #

TD — Time-of-Day (TOD)

Purpose: To monitor or set the time-of-day (TOD) clock.

Use: PRINT format: PR TD
 LET format: LE TD="HH:MM:SS"
 Field length: 8
 Value: HH:MM:SS

Remarks: HH = hour, MM = minutes, and SS = seconds.
 The TOD clock displays military time and increments the date every 24 hours.

Examples: PRINT request: PR TD
 PRINT response: 02:49:34
 LET request: LE TD="14:55:00"
 LET response: #

TDD — Tuning Complete

Purpose: To view the status of loop auto-tuning.

Use: PRINT format: PR TDD(L), where L is loop number.
 Field length: 8
 Value: Yes, No

Remarks: Yes = auto-tuning complete
 No = auto-tuning not complete

Examples: PRINT request: PR TDD(1)
 PRINT response: No^^^^^^

TDS — Time of Day Start

Purpose: To view or change the time for the clock-driven programmer time-of-day start.

Use: PRINT format: PR TDS(P), where P is programmer number.
 LET format: LE TDS(P)="x", where P is programmer number, x is time of day.
 Field length: 8
 Value: HH:MM:SS

Remarks: Time is displayed or set in hours/minutes/seconds 24-hour format.

Examples: PRINT request: PR TDS(1)
 PRINT response: 04:14:12
 LET request: LE TDS(1)="04:15:00"
 LET response: #

TEN — Tuning Error Number

Purpose: To view the number of an error that occurred during loop auto-tuning.

Use: PRINT format: PR TEN(L), where L is loop number.
Field length: 8

Remarks: 600 = Loop not in manual mode.
604 = PV not responding to output change.
605 = Process not moving fast enough; increase sample rate.
606 = Manual output changed during tuning process.
607 = System error.
608 = Buffer full.
609 = User abort.

Examples: PRINT request: PR TEN(1)
PRINT response: ^^^^600

TER — Tuning Error

Purpose: To view the yes/no error status of loop auto-tuning.

Use: PRINT format: PR TER(L), where L is loop number.
Field length: 8
Value: Error, or blank

Remarks: Error = Error occurred during auto-tuning.
(Blank) = No errors during auto-tuning.

Examples: PRINT request: PR TER(1)
PRINT response: Error^^

TL — Time Left in Segment (SEG time)

Purpose: To monitor the amount of time left in the current segment.

Use: PRINT format: PR TL(P), where P is programmer.
Field length: 8
Value: HH:MM:SS

Remarks: This system variable monitors the amount of time left in the current segment for a specific programmer.

Examples: PRINT request: PR TL(1)
PRINT response: 00:08:34

TLP — Time Left in Program

Purpose: To view the amount of time left in a program.

Use: PRINT format: PR TLP(P), where P is programmer number.

Field length: 8

Value: HH:MM:SS or DD-HH:MM. A “-” between DD and HH instead of a “:” indicates days field in effect.

Remarks: Displays time in hours/minutes/seconds 24-hour time format or days-hours-minutes format if remaining time exceeds 24 hours. If end of program can't be found or if nested program recycles are present, will display XX:XX:XX.

Examples: PRINT request: PR TLP(1)
PRINT response: 00:09:13

TP — Time into Program

Purpose: To monitor the amount of time the current program has been running.

Use: PRINT format: PR TP(P), where P is programmer.

Field length: 8

Value: HH:MM:SS or DD-HH:MM

Remarks: This variable displays how long a program has been running. The DD-HH:MM format is used only when a program has been running longer than one day.

Examples: PRINT request: PR TP(1)
PRINT response: 01:30:23

TS — Tuning Status

Purpose: To view the status of, or initiate, or abort loop auto-tuning.

Use: PRINT format: PR TS(L), where L is loop number.

LET format: LE TS(L)="x", where L is loop, x is status.

Field length: 8

Value: Start, Abort

Remarks: If status is “Start”, auto tuning is not in progress and a command of LE TS(L)="S" will start it. If status is “Abort” auto tuning is in progress and command of LE TS(L)="A" will abort it.

Examples: PRINT request: PR TS(1)
PRINT response: Start^^^
LET request: LE TS(1)="A"
LET response: #

TYP — Analog Input Type

- Purpose: To view or change the types of inputs used on an analog I/O card.
- Use: PRINT format: PR TYP(I), where I is analog input number.
 LET format: LE TYP(I)="x", where I is analog input number, x is input type.
 Field length: 8
 Value: See remarks.
- Remarks: Use the command PR ALL(I), where I is the input number to display a list of all permissible input types.

! **Important:** Input type changes also require wiring and sensor changes.

- Examples: PRINT request: PR TYP(1)
 PRINT response: Type..B.
 LET request: LE TYP(1)="Type..G."
 LET response: #

UT — Engineering Units Type

- Purpose: To view or change the engineering unit type used with an analog input.
- Use: PRINT format: PR UT(I), where I is analog input number.
 LET format: LE UT(I)="x", where I is analog input number, x is engineering units.
 Field length: 8
 Value: Centigrd, Fahrenht, Linear
- Remarks: Centigrd = TC or RTD
 Fahrenht = TC or RTD
 Linear = Linear input
- Examples: PRINT request: PR UT(1)
 PRINT response: Centigrd
 LET request: LE UT(1)="Fahrenht"
 LET response: #

Example of System Variable Use

This section gives an example of getting something done using the system variables via a Dimension communications port. We explain the steps necessary to set up programmer 1 to control loops 1 and 2 (we'll call them Temperature and Humidity) through a three segment recipe which ramps up, soaks for a while, then ramps back down. We assume a few things here to make the example simpler:

- A. The Network communications mode is used, but we will leave out the mechanics of the protocol (STX, address, ETX, checksum). The commands shown will work via either the EIA-232 or EIA-422 ports.
- B. Correct tuning parameters have already been set for loops 1 and 2. If you wish to set tuning parameters over the network, see the description of the PID Group parameter.
- C. Loops 1 and 2 are configured to take their setpoints from programmer 1. Each loop should have its Setpoint Source (SptSrc) set to Programr and its Remote Setpoint Source (RemSPsrc) set to Program1. For help in configuring them as such, see Section 4. The description of the SPS parameter in this appendix is also helpful.
- D. Programmer 1 is configured to be started by the operator. Dimension considers a start command from the network (using the PS system variable) as an Operator start. The CONF_PRG screen for programmer 1 should have the OPStart field set to OPStrt01. (Section 3, Programmer).

Step 1: Place the Programmer in Hold.

```
LE PS(1)="H"
```

The programmer really should be in `Hold` or at `PG_END` before you start another recipe program, but this sets programmer 1 status to `Hold` if it happens to be in `Run`. If it is at `PG_END` it will remain at `PG_END`. If it is in `Run`, it will stop and hold its present setpoints. This should ensure that the following commands will be accepted.

Step 2: Create a Recipe in the Segment Table.

```
LE SG(1,1)="90,80,0,0,0,0,0,0,01:30:00,_____,0,2,1,1,"
LE SG(1,2)="90,80,0,0,0,0,0,0,04:00:00,_____,0,3,1,1,"
LE SG(1,3)="25,50,0,0,0,0,0,0,01:15:00,_____,0,3,1,1,"
```

This program will increase the temperature to 90 degrees and humidity to 80% over one hour thirty minutes, hold these settings for four hours, then ramp back down to 25 degrees and 50% humidity in one hour fifteen minutes. It sets no event outputs and it uses Alarm Group 1 and PID Group 1 for the entire program.

Step 3: Set the Operator Start Segment for the programmer.

LE PS4(1)="1" (Set the Operator Start Segment for programmer 1 to segment 1)

Step 4: Set modes of programmed loops to Auto.

LE LS(1)="A" (Set Loop Status for loop 1 to Auto)

LE LS(2)="A" (Set Loop Status for loop 2 to Auto)

Step 5: Issue the Start Programmer command.

LE PS(1)="S" (Start programmer 1 at segment in PS4(1))

Once this last command is given, programmer 1 will start running segment 1 and controlling the setpoints of loops 1 and 2. While the programmer is running, it may be held and resumed with the following commands:

LE PS(1)="H" (Hold programmer 1 at current segment)

LE PS(1)="R" (Run programmer 1 at current segment)

Error Messages

If the host computer makes an illegal request message to Dimension, or if a syntax error occurs, Dimension will return an error message to the host computer. All error messages use the same six-byte format: an E followed by a space, followed by a four-digit error identification number. Each type of error condition has a separate error identification number. For example, E 0403 is the error identification number for an illegal system variable. Table B-5. lists all error identification numbers and the error condition to which they call attention.

Table B-5. Dimension Communication Error Messages

Error Message	Error Condition
E 0215	Illegal pathname
E 0242	Write protect (read only)
E 0244	Read error
E 0245	Write error
E 0401	Bad assignment parameter
E 0402	Bad string variable
E 0403	Illegal system variable
E 0404	Illegal assignment
E 0405	Bad pathname
E 0406	Incomplete path list
E 0407	Read only parameter
E 0408	Buffer overflow
E 0409	Illegal tagname assignment

IEEE-488 Option Installation and Configuration

IEEE-488 Introduction

The IEEE-488 bus is also known as the "General Purpose Interface Bus" (or GPIB). The purpose of the IEEE-488 bus is to transfer information between two or more devices. A device may either be an instrument or a computer. Devices may 'Talk' (send), 'Listen' (receive), or be idle on the bus. Only one device can be 'addressed' (allowed) to Talk on the bus at a time, while one or more devices may Listen.

One device on the bus must be the System Controller. The Controller determines which device will talk and which devices will listen. The Controller is usually a computer containing a IEEE-488 interface card and application programs. The IEEE-488 specification allows the System Controller to pass its duties to another device, making it the 'Active Controller'.

There are two types of messages sent over the IEEE-488 bus: interface messages and device dependent messages. Interface messages manage the IEEE-488 bus, using commands to clear the interface, address devices to talk or listen, and so forth. Device dependent messages transfer commands specific to the device (Dimension Industrial Process Managers for example).

IEEE-488 and Dimension

The Dimension Industrial Process Manager handles the basic IEEE-488 commands. It uses a selectable primary address (0-31) and handles the Talker and Listener functions. Dimension will act as a peripheral device, answering information asked of it. Serial and parallel polling, and the Controller functions have not been implemented.

IEEE-488 Installation

Cabling and Connections

Dimension uses the standard 24 position IEEE-488 connector for the IEEE-488 bus. The recommended cable style to use with Dimension is an "In-Line" mating cable. See Fig. B-13. This allows clearance for other cables and connectors on the rear panel of the Dimension.

Cable lengths should be kept as short as possible and a good quality shielded cable is recommended. IEEE-488 specifies:

- (1) Maximum of 15 devices per system, and
- (2) Total combined length of all cables is 20 meters or less.

Figure B-13.
IEEE-488 In-Line
Mating Cable

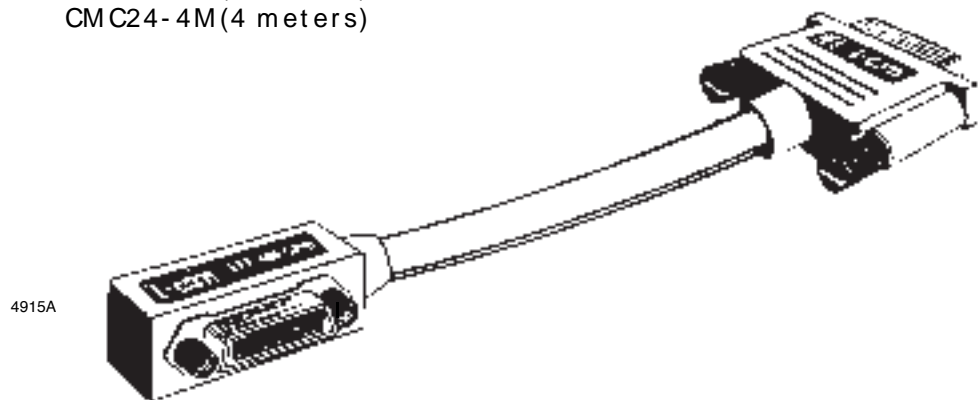
IN-LINE MATING CABLE

L-COMM (phone 800-343-1455)

CM C24- 1M (1 meter)

CM C24- 2M (2 meters)

CM C24- 4M (4 meters)



IEEE-488 Configuration

Configuration parameters must be selected for each of the devices on the IEEE-488 bus, including the System Controller. The parameters are adjusted to match the communication characteristics of each device. Frequently, the default settings of these parameters will work if the correct address has been selected.

Dimension 8705 Configuration/Set up

Dimension's 8705 configuration parameters are set from the IEEE-488 configuration screen. This screen, displayed in Fig. B-14, is accessed from the Communications Configuration Screen.

Figure B-14.
8705 IEEE-488
Configuration Screen



PrimAddr: Selects the Primary IEEE-488 Address of the Dimension, which is different than the Network Address. The address range is 0 to 31; address 4 is the factory default.

EOM_Char: Selects the END of Message Character. This is set to the factory default, CR-LF (Carriage Return and Line Feed). The choices are:

- CR-LF
- LF-CR
- LF
- CR

Dimension 8725 Configuration/Set up

Dimension's 8725 configuration parameters are also set from the IEEE-488 configuration screen. This screen, displayed in Fig. B-14, is accessed from the Communications Configuration Screen.

Figure B-15.
8725 IEEE-488
Configuration Screen



PrimAddr: Selects the Primary IEEE-488 Address of the Dimension, which is different than the Network Address. The address range is 0 to 31; address 4 is the factory default.

EOM_Char: Selects the END of Message Character. This is set to the factory default, CR-LF D of Message Character. This is set to the factory default, CR-LF (Carriage Return and Line Feed). The choices are:

CR-LF
 LF-CR
 LF
 CR
 None **** Uses the EOI hardware line to signal the End of Message.

ScndAddr: Enables or disables the use of Secondary addressing. This is set to "disable" from the factory.


ScndAddr: Selects the Secondary IEEE-488 Address of the Dimension. The choices are 96 to 126. This is set to 0 when secondary addressing is disabled.

System Controller Configuration

The System Controller will need to be configured for Dimension and the other devices connected to the IEEE-488 Bus. Below are the parameters that can be set when using a National Instruments AT-GPIB interface board and software, with an IBM AT compatible personal computer.

DEVICE CHARACTERISTICS Device : DEV4

<u>Parameter</u>	<u>Default Settings</u>	<u>Recommended Settings</u>
Primary GPIB Address	1-16	4
Secondary GPIB Address	None	None
Timeout Setting	T10s	T10s
EOS byte	00H	0AH
Terminate Read on EOS	no	yes
Set EOI with EOS on Write	no	no
Type of compare on EOS	7-bit	8-bit
Set EOI w/last byte of Write	yes	yes
Repeat Addressing	no	

 **Note:** Please consult the documentation supplied with your IEEE-488 interface card for more information.

Primary GPIB Address

The Primary GPIB Address is selectable from both Dimension and the National Instruments software. Each device on the IEEE-488 bus must have a unique primary address and will have its own Device Characteristic setup screen when using National Instruments software. The primary address for the Dimension must match the selected primary address in the Device Characteristics if they are to communicate.

Secondary GPIB Address


Secondary Addressing is typically used in special applications that require extended addressing. If used, the address selected must be the same as selected on the Dimension in order to communicate. The normal setting for this item is `Disable`.

Timeout Setting

The Timeout setting is the maximum length of time that an I/O operation can take before a timeout occurs. One way this is used is as a Time out Error on the IEEE-488 bus. The default timeout setting of 10 seconds is more than adequate for most devices. Shortening the timeout delay will decrease the length of time spent waiting for a communication error to be handled, too short of time may cause communication errors. Values of 10 msec. to 10 seconds will work for most Dimension applications.

EOS Byte

The End Of String (EOS) byte is an optional character that is sent at the end of the message string to indicate the end. Dimension defaults with Carriage Return and Line Feed characters (0D + 0A Hex) as the End of String byte.

 **Note:** *The selection made for this device characteristic on National Instruments software is for reference only.*

Messages can be terminated by an EOS byte, setting the EOI line, or by sending a specified number of characters. The application program used will determine which method should be selected.

Dimension will always set the EOI line when sending a message and will terminate the reception a message if the EOI line gets set. The EOS character(s) will be sent by Dimension if selected in the Configuration screen. The Sample Programs on page B-96 were written to use the CR-LF as the EOS byte.

Terminate Read on EOS

Terminate Read on EOS, if set, will cause the interface board to terminate a Read operation when the EOS byte has been received. Dimension can be used with this parameter set to either yes or no. If an EOS character on Dimension has been selected to be something other than 'None', this parameter should be set to 'yes'. However, the application software that will be used will determine the correct setting.

Set EOI with EOS on Write

Set EOI with EOS on Write, if set, will cause the interface board to set the EOI line whenever it sends the EOS byte. Dimension can be used with this set to either yes or no. The application software that will be used will determine the correct setting.

Type of Compare on EOS

Type of compare on EOS byte selects whether all 8 bits of the character are compared or if only the first 7 bits are compared to see if they match. Dimension comes from the factory set for 8 bit characters. Most applications will work in either setting.

Set EOI with Last Byte of Write

Set EOI with last byte of write, if set, will cause the interface board to set the EOI line with the last byte of every message sent. If Dimension has been set up with the EOS character set to 'None', this feature must be set to 'yes' to receive messages. For other the other selections of the EOS character on Dimension, this feature can be set to either yes or no. The recommended setting is 'Yes'.

Repeat Addressing

Normally, devices are addressed each time a read or write operation is performed. If no is selected, the read and write operations will not readdress the selected device if the same operation was just performed. This saves time when repeated operations are performed with the same device.

IEEE-488 and Data Transfer

Data Transfer

Data transfer on the IEEE-488 bus is handled by the following hardware and software:

- IEEE-488 hardware bus
- Driver software
- Application software

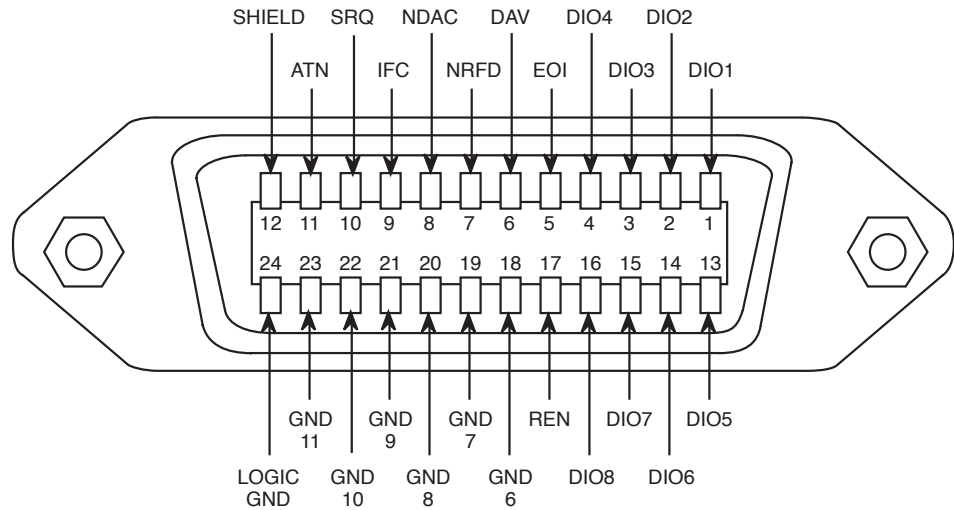
Application software controls the driver software which in turn controls the hardware bus. Each is explained in general terms in the following section. Detailed information on IEEE-488 can be found in the following documents:

ANSI/IEEE 488.1-1987: IEEE Standard Digital Interface for Programmable Instrumentation, and ANSI/IEEE 488.2-1987: IEEE Codes, Formats, Protocols, and Common Commands, and Standard Commands for Programmable Instruments.

IEEE-488 Hardware Bus

The IEEE-488 Bus consists of 24 wires or 'lines' (Fig. B-16): 8 data lines, 8 ground lines, 3 handshake lines, and 5 interface management lines.

Figure B-16.
IEEE-488 Standard
Connector



4915

Handshake lines:	DAV	-Data Valid
	NRFD	-Not Ready For Data
	NDAC	-Not Data Accepted

Interface Management lines:	ATN	-Attention
	IFC	-Interface Clear
	REN	-Remote Enable
	EOI	-End or Identify
	SRQ	-Service Request

Data lines: DIO1 through DIO8

Data lines carry both data and command messages to and from the devices.

Handshake Lines

Handshake lines control the transfer of messages between devices. The handshaking process guarantees that the bytes on the data lines are sent and received without transmission errors.

DAV (Data Valid) – indicates when the information on the data lines are stable or valid and can be accepted by devices. This line is controlled by the Talker when sending data messages and by the Controller when sending commands.

NRFD (Not Ready For Data) – indicates when a device is ready to receive a message. The line is controlled by all devices when receiving commands and by Listeners when receiving data messages. Note, this line will not be released (ready to accept data) until *all* devices have released it.

NDAC (Not Data Accepted) – indicates when a device has accepted the data or control byte on the data lines. The line is controlled by all devices when receiving commands and by Listeners when receiving data messages. Note, this line will not be released (data accepted) until *all* devices have released it.

Interface Management Lines

Interface management lines manage the flow of control and data bytes across the interface.

ATN (Attention) line is set by the Controller to indicate that it is putting an address or control byte on the data bus. ATN is released to allow the Talker to put data or status information on the bus.

IFC (Interface Clear) line is set by the Controller in order to initialize all devices to a known state.

REN (Remote Enable) line is set by the Controller and is used to place devices in remote or local program mode.

EOI (End of Identify) line is used in two ways. A Talker uses this line to indicate the end of a message. The Controller can use this line to tell devices to identify their response in a parallel poll.

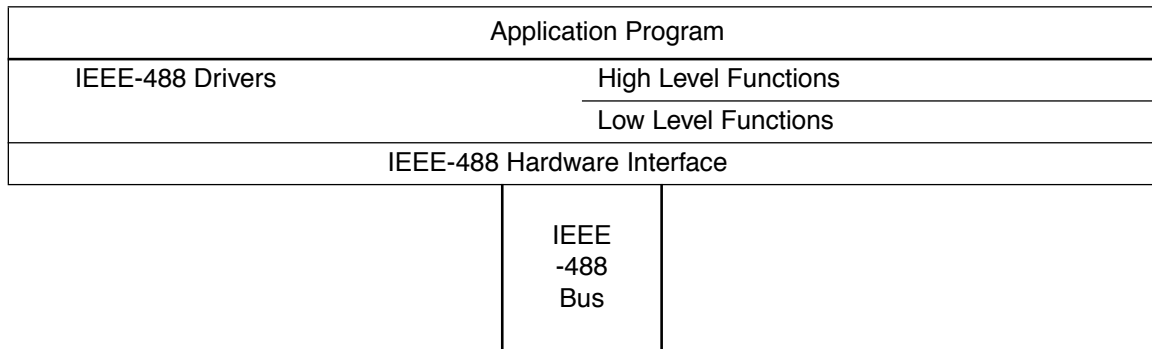
SRQ (Service Request) line can be set by any device to request that the Controller take some action. The Controller will in turn determine which of the devices is requesting service and then take the appropriate action.

Driver Software

The Driver software is the lowest level of IEEE-488 software which interfaces directly with the IEEE-488 hardware. Driver software automatically handles the IEEE-488 protocol and bus management details.

There are three popular software command sets for programming the IEEE-488 bus: HP style commands, IEEE-488.1 commands, and the enhanced IEEE-488.2-1987 commands. The command syntax varies based on the programming language, type of computer, and vendor of IEEE-488 hardware.

A pictorial representation of how the different levels of software and hardware fit together is shown below.



IEEE-488 Bus Commands

The commands associated with the IEEE-488 bus can be grouped into three general categories: Uniline Commands, Multiline Commands, and Device-Dependent Commands. See Table B-6. for a partial listing of these commands.

Uniline Commands – effect only one of the IEEE-488 bus lines. These commands are executed by setting or clearing the appropriate line.

Multiline Commands – effect more than one of the IEEE-488 bus lines. Multiline commands are sent by the Controller with the ATN line set.

Device-Dependent Commands – are those commands that are specific to a particular device, such as Dimension. These commands are transmitted by the data lines while the ATN line is not set. An example would be sending the characters "PR TD" over the data lines to the Dimension. This would request the time of day information from Dimension. See page B-19 for a list of system variable commands and Table B-4. , page B-22, for a list of the available system variables.

Table B-6. IEEE-488 Bus Command Summary

Command Type	Command	Description
UNILINE	REN (Remote Enable) EOI (End or Identify) IFC (Interface Clear) ATN (Attention) SRQ (Service Request)	Sets up device for remote operation. Marks end of transmission. Clears interface. Defines data bus contents Controlled by external device. (Currently not implemented on Dimension)
MULTILINE		
Universal	LLO (Local Lockout) DCL (Device Clear) SPE (Serial Poll Enable) SPD (Serial Poll Disable)	Locks out local operation. (No effect on Dimension) Returns device to default. (See "System Variable Commands" on page B-19) Enable serial polling. (No effect on Dimension) Disable serial polling. (No effect on Dimension)
Addressed	MTA (My Talk Address) MLA (My Listen Address) MSA (My Secondary Address) SDC (Selective Device Clear) GTL (Go To Local) GET (Group Execute Trigger)	Selects device to be talker. Selects device to be listener. Selects device's secondary address. Returns unit to default. (See "System Variable Commands" on page B-19) Returns device to local. Triggers device for reading.
Unaddressed	UNL (Unlisten) UNT (Untalk)	Removes all listeners from bus. Removes any talkers from bus.
DEVICED-DEPENDENT	Table B-4. , for list of commands.	

Application Software

Application software performs an overall process by interfacing with the IEEE-488 driver programs. Application software is typically written using a programming language such as 'Basic' or 'C' to control a process. The process may be as simple as taking a voltage measurement or complex enough to control an Environmental Chamber throughout its series of tests. Software vendors have many useful tools and libraries of functions to program the IEEE-488 bus.

Each programming language has its own language interface which contains the functions necessary for the language to access the IEEE-488 drivers. National Instruments has many language interface utilities available including a utility called Universal Language Interface (ULI). ULI is set up for users who are familiar with programming Hewlett-Packard controllers. It interfaces between MS-DOS and the IEEE-488 drivers using HP style functions.

Dimension Commands

Dimension uses five commands to access system variables (see "*System Variable Commands*" on page B-19).

Sample Programs

Listed is a short program for testing the communication from your IEEE-488 system to your Dimension. Three versions of the program have been listed, each using a different command set (HP style commands, IEEE-488.1 commands, and IEEE-488.2-1987 commands). The output of the program is shown in Fig. B-16.

The programs are written for the 'Basic' programming language. The test setup used was: the MS-DOS operating system on a AT style computer, National Instruments GPIB IEEE-488 hardware and corresponding software drivers, and Microsoft QuickBasic version 4.5. Information following the single quotes (') are program comments and do not need to be entered. Refer to pages B-96 through B-98 for sample programs.

Figure B-17. Test Program Output Screen

```

  DATE          TIME          LS          SP          PV          RO
  ==  ==  ==  ==  ==  ==  ==  ==  ==  ==  ==  ==
  01/01/98     17:00:00     MANUAL     0.0     Opensens     0.00

```

```

'   Sample Program Using IEEE-488.1 Commands
'   -----
'
'   THE DEFAULT IEEE-488 ADDRESS IS 4.
'   THE EOM CHARACTER FOR THE DIMENSION IS CR
'   USE THE ESCAPE KEY TO END THE PROGRAM.
'
'   Load file QBDECL.BAS - Quick Basic initialization code
'   REM $INCLUDE: 'C:QBDECL.BAS'
'
'   IEEEAD% = 4                ' Set the IEEE address to 4
'   IEEEAD$ = "4"
'   DEVNAME$ = "DEV" + IEEEAD$ ' Make device name DEV4 (default)
'
'   CALL IBFIND(DEVNAME$, D%) ' Find identifier for Dimension
'   CALL IBSIC(brd0%)         ' Initialize GPIB Bus
'   CALL IBCLR(D%)            ' Clears Dimension device (SDC)
'   CALL IBSRE(brd0%, D%)    ' Sets Dimension to remote
'   CLS                       ' Clears Screen
CYCLE:
'   Print header line starting at line 3 column 1.
'   LOCATE 3, 1
'   PRINT " DATE           TIME      LS           SP           PV           RO           "
'   PRINT "=====
'   PRINT
'
'   Send data to Dimension - Ask for parameter values.
'   CALL IBWRT(D%, "PR DT;' ';TD;' ';LS(1);' ';SP(1);' ';PV(1);' ';RO(1)")
'
'   Initialize data string to 61 spaces.
'   LDATA$ = SPACE$(61)
'
'   Address Dimension to talk and put data in the string.
'   CALL IBRD(D%, LDATA$)
'
'   Print the data.
'   PRINT LDATA$
'
'   Test for escape key pressed to end the program.
'   IF INKEY$ = CHR$(27) THEN END
'
'   GOTO CYCLE                ' Repeat this cycle again.
'   END

```

Communications

```

'   Sample Program Using IEEE-488.2-1987 Commands
'   -----
'
'   THE DEFAULT IEEE-488 ADDRESS IS 4.
'   THE EOM CHARACTER FOR THE DIMENSION IS CR.
'   USE THE ESCAPE KEY TO END THE PROGRAM.
'
'   Load file QBDECL.BAS - Quick Basic initialization code
'   REM $INCLUDE: 'C:QBDECL.BAS'
'
'   IEEEAD% = 4                ' Set the IEEE address to 4
'   IEEEAD%(0) = 4
'   CALL SendIFC(0)            ' Initialize GPIB Bus
'   CALL DevClear(0, IEEEAD%)  ' Clears Dimension device (SDC)
'   CALL EnableRemote(0, IEEEAD%()) ' Sets Dimension to remote
'   CLS                        ' Clears Screen
CYCLE:
'   Print header line starting at line 3 column 1.
'   LOCATE 3, 1
'   PRINT " DATE           TIME      LS           SP           PV           RO           "
'   PRINT "=====
'   PRINT
'
'   Send data to Dimension - Ask for parameter values.
'   CALL Send(0, IEEEAD%, "PR DT;'  ';TD;'  ';LS(1);'  ';SP(1);'  ';PV(1);'
';RO(1)" + CR$, DABend)
'
'   Initialize data string to 61 spaces.
'   LDATA$ = SPACE$(61)
'
'   Address Dimension to talk and put data in the string.
'   CALL Receive(0, IEEEAD%, LDATA$, STOPend)
'
'   Print the data.
'   PRINT LDATA$
'
'   Test for escape key pressed to end the program.
'   IF INKEY$ = CHR$(27) THEN END
'
'   GOTO CYCLE                ' Repeat this cycle again.
'   END

```

Appendix C

Verification/Calibration Procedure

Required Prerequisites

Allow a minimum of two hours of stabilization time for the Dimension, the Analog Input connector, and Verification/Calibration equipment.

Room ambient should be stable (± 3 degrees C) with minimum air flow at the Analog Input connector. Measure the ambient air temperature. This will be required to calculate the actual tolerance in step 2 of the Input Verification.

Required Equipment

- Digital Voltmeter
- Millivolt/Volt Calibration Source
- Temperature Calibrator/Simulator
- RTD Calibration Source/Simulator

When using a Temperature Calibrator/Simulator, use special limits thermocouple wire of 24 ga or heavier, 3 feet or less in length for calibration.

Equipment Specifications

Digital Voltmeter 6 1/2 Digit Example: Keithley Model 196

	ACCURACY $\pm(\%rdg+counts)$			TEMP COEF $\pm(\%rdg+counts)/^{\circ}C$	
	24 HOURS			1 YEAR	
RANGE	RESOL	INP RES.	23 \pm 1 $^{\circ}C$	18-28 $^{\circ}C$	0-18 & 28-50 $^{\circ}C$
300mV	100nV	>1G ohm	0.0020+20	0.008+20	0.0006 + 10
3V	1uV	>1G ohm	0.0013+10	0.0038+20	0.0004 + 1
30V	10uV	11 M ohm	0.0015+10	0.008+30	0.0013 + 3

Millivolt/Volt Calibration Source Example: EDC Model 521B

Output Specifications:

	<u>0-100 MV range</u>	<u>0-10 volt range</u>
Resolution (1PPM)	100 nV	10 uV

Accuracy Statement $\pm(0.002\%$ of setting + 0.0005% of range + 3 uV)

Stability: 1 year: $\pm 0.002\%$



Note: If The Digital Voltmeter Is Used To Measure And Set The Output Of The Calibrator, The Primary Spec Of The Calibrator Is The Stability.

Temperature Calibrator/Simulator Example: Ectron Model 1120

Accuracy specs = or better than published Dimension specs for thermocouple inputs.

For example:


At 22 degrees C ambient, type K thermocouple, spec = $\pm 0.45^{\circ}\text{C}$

At 22 degrees C ambient, type T thermocouple, spec = $\pm 0.35^{\circ}\text{C}$

RTD Calibration Source Example: ESI Model DB 62

Resistance Decade Box

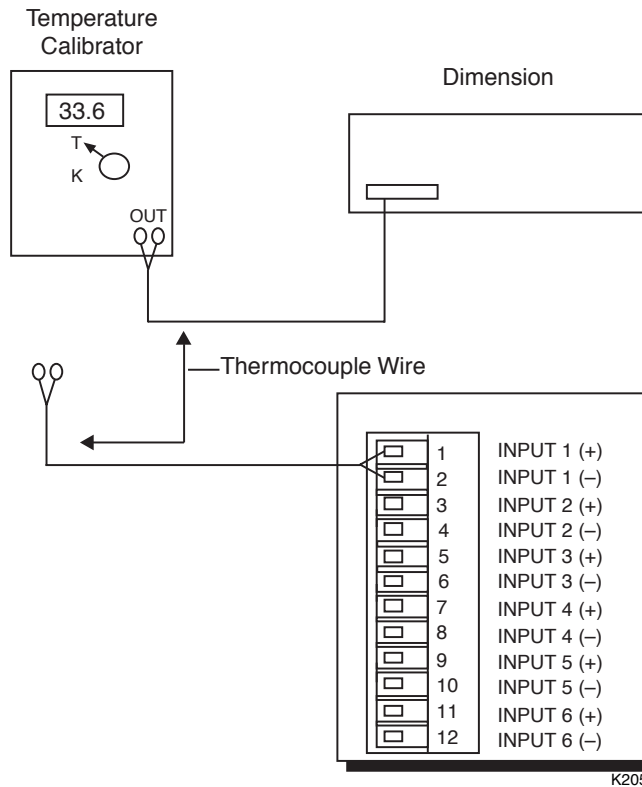
RANGE	ACCURACY		
	INITIAL %	LONG TERM %	TEMP COEF(PPM/C)
1000 ohm	0.01	0.02	5
100 ohm	0.012	0.025	15
10 ohm	0.03	0.07	20
1 ohm	0.2	0.5	60
0.1 ohm	2	5	400

 *Note: Rtd Simulators Must Be Capable Of Operating With 160 Microamps Of Excitation Voltage*

Dimension Input Verification

1. Connect a Temperature Calibrator with thermocouple wire to terminals 1 and 2 of the Analog input connector on the Dimension. See Fig. 1.

Figure C-1.
Temperature Calibrator
Connection



➔ **Note:** Allow two hours to stabilize operating temperature of the Dimension, analog input connector and the Temperature Calibrator.

For RTD's, connect an RTD source as shown in Fig. 8.

2. Pick several temperatures from the Temperature Calibrator and read the value of temperature on the Dimension for "T" thermocouple. See Fig. 2.

Tolerance: Dimension Accuracy + Dimension Temperature Coefficient + Tolerance of Calibrator + Thermocouple Leadwire Error

Assuming minimal or no error from the temperature calibrator and thermocouple wire, will leave just the Dimension tolerance. For example, the "T" thermocouple at a room temperature of 30° C would be:

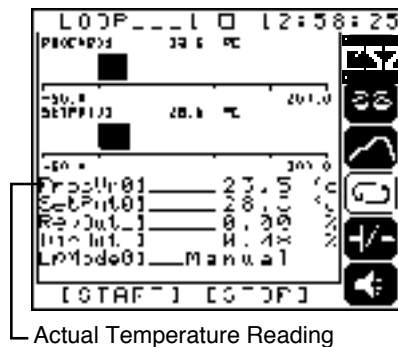
$$\text{Dimension Accuracy} + \text{Temp. Coeff.} = \text{Tolerance}$$

$$.35 + .16 = \pm .51^\circ \text{ C}$$

Temperature Coefficient is calculated from the specification given in Table C-1 and multiplying that by the difference between room temperature and 22° C (72° F). In the example, the room temperature was 30 and the temperature coefficient was .02.

$$(30 - 22) \times .02 = .16$$

Figure C-2.
Temperature Reading



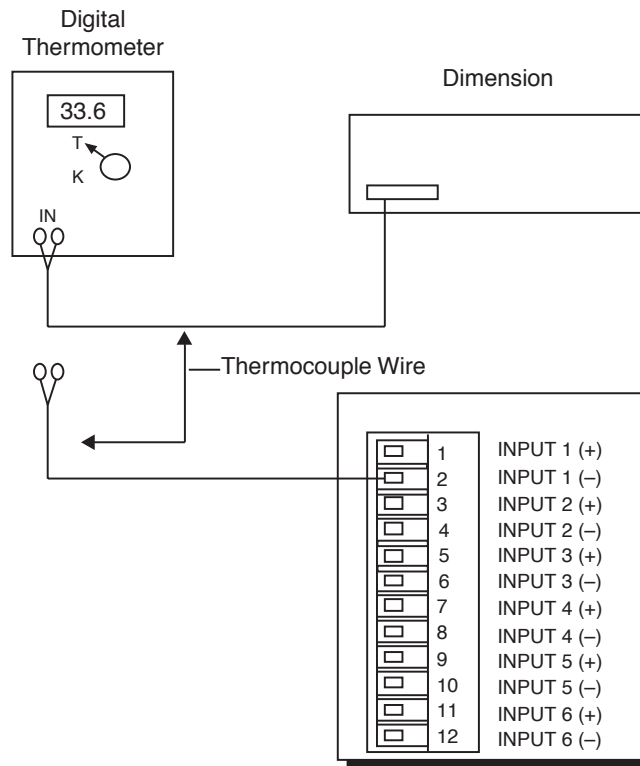
Note: The temperature range is -200° to +400° C for a "T" thermocouple.

3. If unit is found outside of specifications:
 - A. Allow additional time for instrument and connector stabilization (2-hour minimum time required).
 - B. Verify Cold Junction calibration – see "Cold Junction - Calibration" on page Appendix C-6.
 - C. See "Input Calibration" on page Appendix C-7.

Cold Junction - Calibration

1. Connect a digital thermometer to terminal #2 of the Analog input connector on the Dimension. See Fig. 3.

Figure C-3.
Digital Thermometer
Connection



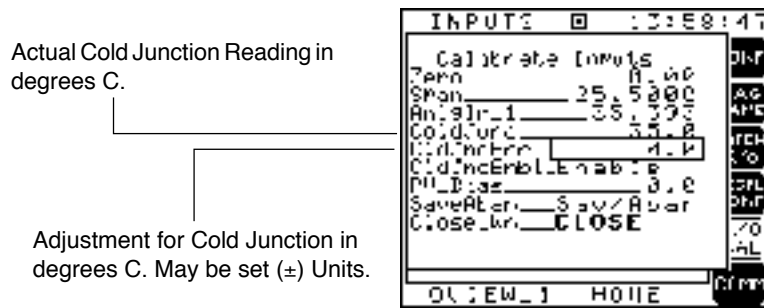
➔ **Note:** Allow two hours to stabilize operating temperature of the Dimension, analog input connector and the Digital thermometer.

Cold Junction on the Dimension is only displayed and able to be set in degrees C.

Tighten thermocouple in screw connector.

2. Adjust `CIdJncErr` (Cold Junction Error) so the actual value of `ColdJunc` matches the digital thermometer reading. See Fig. 4.
3. After `CIdJncErr` has been set, scroll down to `SaveAban`, select `Save` and press **ENTER**.

Figure C-4.
Cold Junction Reading
and Adjustment



➔ **Note:** Any inaccuracy in the cold junction calibration introduces a direct error in the Process Variable reading.

CldJncEnbl must be set to ENABLE when a thermocouple is selected as a sensor.

Input Calibration

1. If the Zero/Span calibration numbers have been changed from the factory settings, select “abandon” on the Save/Aban selection on the calibrate window. See Fig. 5. This will restore previous values *unless* Zero/Span has been changed and SAVED.

Figure C-5.
Calibrate Inputs Window



2. If Zero/Span numbers have been changed and saved they cannot be restored to the factory supplied numbers. Refer to Fig. 6 and the following procedure.
 - Select appropriate PG Channel in the Inpt Type field for the sensor type used on the input being selected.
 - If sensor type is RTD, select RTD. am.1.

	Input Range	Input Type
PG_CH_01	-17 to 109 Millivolts	J,K, E, N, t/c
PG_CH_02	Reserved for special ranges	Special
PB_CH_03	-1.0 to 10 Volts DC	Linear, Volt, Milliamp
PG_CH_04	-14 to 26 Millivolts	T, S, R, B, t/c
RTD	-200 to 630 °C	100 W American Alpha or European

Figure C-6.
Inputs Screen



Example: Zero/Span Calibration

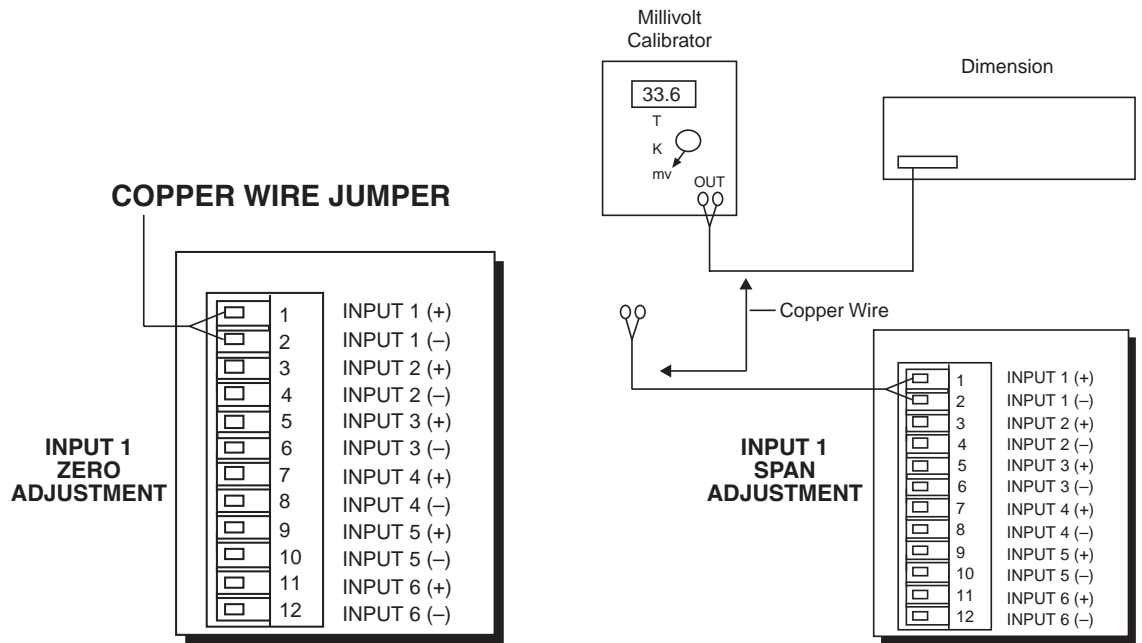
If calibrating a “T” thermocouple on channel 1 and if the factory calibration numbers have been changed, perform the following procedure:

1. Select the `InptType` to PG_CH_04 (Fig. 6)
2. Set `PV_Bias` = 0.0 (Fig. 5)
3. Install **jumper** in place of thermocouple on the Analog input connector, terminals 1, 2 (Fig. 7) (input #1). Note: jumper is used as a precision 0.0 millivolt input and may be used for the Zero Calibration for any of the PG_CH input ranges. No calibrator necessary.
4. Allow minimum **warm-up** time of 10 minutes.

➡ **Note:** 2 hours minimum warm-up time required for cold junction calibration.

5. Adjust `Zero` number on the Dimension Calibrate window (Fig. 5) so the `Anlgn_1` value reads 0.00. Remember units are in direct millivolts and the procedure is calibrating two points within the input range. Remove jumper when complete.
6. Connect **millivolt calibrator** to input number 1 terminals 1(+) and 2(-) (Fig. 7) and apply 25.000 millivolts. Note any millivolt value can be used within the PG_CH_04 range selected. Main concern is precision and stability of the input signal source.

Figure C-7.
Millivolt Calibrator Connection



7. Adjust `Span` number on the Dimension Calibrate window (Fig. 5) so the `AnlgIN_1` value reads 25.000 (if 25.000 mv applied).
8. When both `Zero/Span` adjustments have been completed, select `SAVE` on the `SaveAban` line within the calibrate window (Fig. 5), and click the knob to save the new calibration values.
9. Return `InptType` selection to Type T (Fig. 6).

➡ **Note:** The channel (analog input) has now been calibrated. Return now to "Dimension Input Verification Procedure."

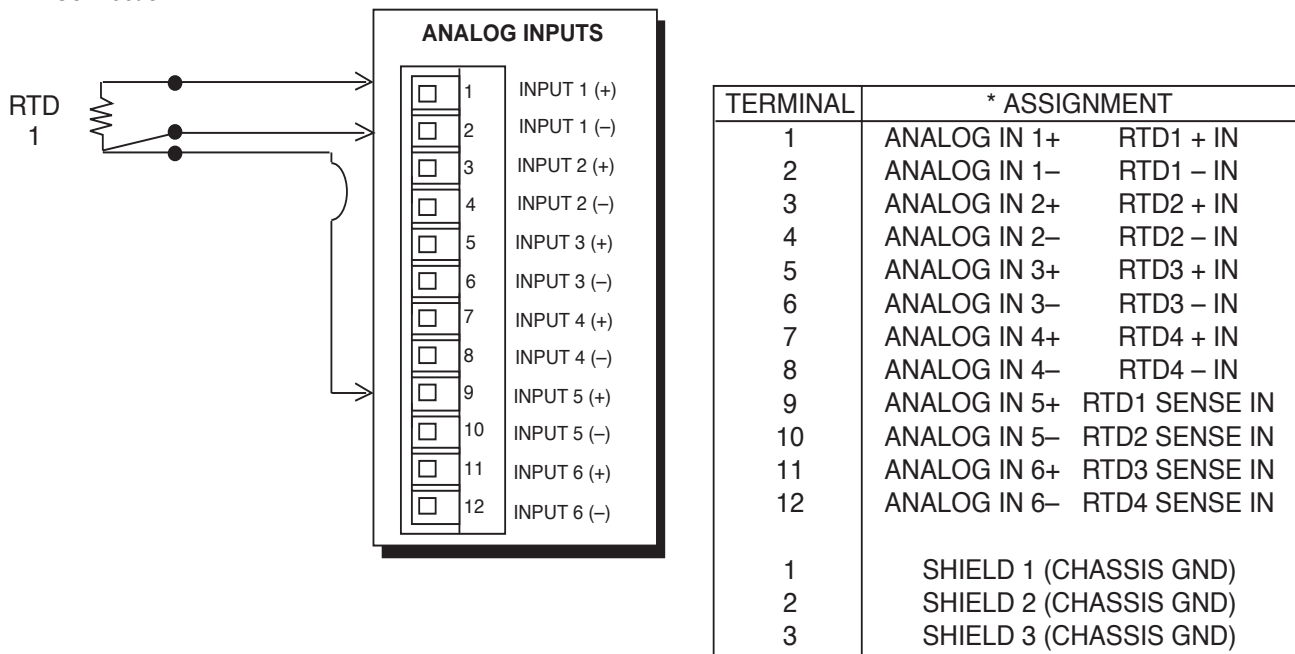
Example: Zero/Span RTD Calibration

If calibrating an RTD input on channel 1 and if the factory calibration numbers have been changed and saved, perform the following procedure:

1. Select the `InptType` to `RTD.am.1` (Fig. 6).
2. Set `PV_Bias` = 0.0 (Fig. 5).
3. Connect an "RTD" source as shown in Fig. 8.
4. Allow minimum **warm-up** time of 10 minutes.
5. Input (apply) 17.14 ohms on the "RTD" source. Adjust `Zero` number on the Dimension Calibrate window (Fig. 5) so the `Anlgn_1` value reads -200.00.
6. Input (apply) 327.02 ohms on the "RTD" source. Adjust the `Span` number on the Dimension Calibrate window (Fig. 5) so the `Anlgn_1` value reads 630.0.
7. When both `Zero/Span` adjustments have been completed, select `SAVE` on the `SaveAban` line within the calibrate window (Fig. 5), and click the knob to save the new calibration values.

➔ **Note:** The channel (analog input) has now been calibrated. Return now to "Dimension Input Verification Procedure."

Figure C-8.
RTD Connection



Thermocouple Error

The Dimension burnout circuit uses 10 Meg Ω resistors to $\pm 15V$ to sense an open thermocouple. Resistance in the thermocouple wire, especially in small gauge wire, will create an offset error on the input reading. The length and gauge of the thermocouple wire used will also affect accuracy. To calculate the error in millivolts, use the following formula:

$$\frac{30 \text{ volts}}{20 \text{ Meg } \Omega} \times R \times FT = \text{Millivolt Offset}$$

Where:

R is the thermocouple resistance per double ft. (See Table 1.)

FT is the length of thermocouple wire.

For example: 6 feet (1.8 meters) of "K" thermocouple, 36 gauge would be calculated as follows:

$$\frac{30 \text{ volts}}{20 \text{ Meg } \Omega} \times \frac{24.08 \text{ } \Omega}{FT} \times 6 \text{ FT} = 0.2167 \text{ mv}$$

At 25°C a "K" thermocouple has 0.040 mv/°C

$$\text{At } 25^\circ\text{C} \quad \frac{0.2167 \text{ mv}}{0.040 \text{ mv}/^\circ\text{C}} = 5.418 \text{ }^\circ\text{C offset.}$$

Setting the PV_Bias to -5.4 would correct for this error.

Verification/Calibration Procedure

Table C-1. Resistance Vs. Wire Diameter

(Resistance in Ohms per Double Foot at 68° F)

AWG No.	Diameter (Inches)	Type K Chrome I/ Alumel	Type J Iron/ Constantan	Type T Copper/ Constantan	Type E Chromel/ Constantan	Type S Pt/ Pt10% Rh	Type R Pt/ Pt13% Rh	Type C W5% Re/W26 % Re
6	0.162	0.023	0.014	0.012	0.027	0.007	0.007	0.009
8	0.128	0.037	0.022	0.019	0.044	0.011	0.011	0.015
10	0.102	0.058	0.034	0.029	0.069	0.018	0.018	0.023
12	0.081	0.091	0.054	0.046	0.109	0.028	0.029	0.037
14	0.064	0.146	0.087	0.074	0.175	0.045	0.047	0.058
16	0.051	0.230	0.137	0.117	0.276	0.071	0.073	0.092
18	0.040	0.374	0.222	0.190	0.448	0.116	0.119	0.148
20	0.032	0.586	0.357	0.298	0.707	0.185	0.190	0.235
24	0.0201	1.490	0.878	0.7526	1.78	0.464	0.478	0.594
26	0.0159	2.381	1.405	1.204	2.836	0.740	0.760	0.945
30	0.0100	5.984	3.551	3.043	7.169	1.85	1.91	2.38
32	0.0080	9.524	5.599	4.758	11.31	1.96	3.04	3.8
34	0.0063	15.17	8.946	7.66	18.09	4.66	4.82	6.04
36	0.0050	24.08	14.20	12.17	28.76	7.40	7.64	9.6
38	0.0039	38.20	23.35	19.99	45.41	11.6	11.95	15.3
40	0.00315	60.88	37.01	31.64	73.57	18.6	19.3	24.4
44	0.0020	149.6	88.78	76.09	179.2	74.0	76.5	60.2
50	0.0010	598.4	355.1	304.3	716.9	185	191	240
56	0.00049	2408	1420	1217	2816	740	764	1000

Excerpted from Omega Temperature Handbook #28

A

- Alarm
 - Defined 2—19
- Alarm Acknowledgment 2—33
- Alarm Group
 - Defined 3—1
- Alarm Icon
 - Flashing 2—33
- Analog Input (AI)
 - Defined 2—19
- AnlgIn 2—19
- Application Software
 - IEEE-488 B—96
- ATN - see Interface Management Lines, IEEE-488

B

- Bus Commands
 - IEEE-488 B—94

C

- Cabling and Connections
 - IEEE-488 B—85
- Calibration
 - Cold Junction C—6
 - Input C—7
- Character Codes
 - ASCII B—16
- Character Entry
 - on Graphic Operator Interface 1—4
- Checksum B—15
- Command Summary
 - IEEE-488 Bus B—95
- Commands
 - System Variable B—19
- Commands - see Message Formats, Network Mode
 - Request and Response
- Communication Sequence
 - Network Mode B—10
- Communications
 - PC to Dimension B—18
- Communications Network
 - Multiple Dimensions connected in a B—3

- Configuration
 - Dimension 8705
 - IEEE-488 B—86
 - Dimension 8725
 - IEEE-488 B—87
 - System Controller, IEEE-488 B—88
- Connection
 - Shield B—4
- Connections
 - Daisy Chain - See Connections
 - RS232
 - Electrical B—2
 - RS-232 B—2
 - RS-422 B—3
- Control
 - Two Position 2—18
- Control Action
 - Bimodal, defined 2—20
 - Defined 2—19
 - Direct, defined 2—19
 - Reverse, defined 2—19
- Control Character
 - End of Text B—14
- Control Characters
 - Network Mode B—10
- Control Output
 - Defined 2—20
- Controls
 - Front Panel, Graphic Operator Interface 1—2
- Conventions vi
- CTAG - see Commands
 - System Variable
- CTAG Command
 - Message B—25
- CTAG Request Error Message B—25
- Current Segment
 - Defined 3—1
- Cursor
 - Graphic Operator Interface 1—4
 - Home Position 1—4
 - Window in 1—4
- Cursor "Home" Position
 - Graphic Operator Interface 1—10
- Cycle Count
 - Defined 3—1

Index

- D**
- Data Link Control B—1
 - Data Message - see Message Formats, Network Mode Request and Response
 - Data Transfer
 - IEEE-488 B—91
 - DAV - see Handshake Lines, IEEE-488
 - Definition
 - Control Loops 2—18
 - Description
 - Real-Time Programmer 3—1
 - Deviation 2—18
 - Defined 2—20
 - Device-Dependent Commands - see Bus Commands, IEEE-488
 - Diamond Key
 - Four Way Keypad 1—3
 - Dimension Commands - see System Variable Commands
 - Display Area
 - Graphic Operator Interface 1—11
 - DLC - See Data Link Control
 - Driver Software
 - IEEE-488 B—93
 - Driver, Software
 - when a host computer is used B—8
 - Drivers
 - Software B—5
 - Dual PID - See PID Type
- E**
- Electrical Connections B—2
 - ENTER Function
 - On Graphic Operator Interface 1—4
 - EOI - see Interface Management Lines, IEEE-488
 - EOS Byte
 - IEEE-488 B—89
 - Error Checking B—17
 - Error Message
 - LET Request B—21
 - Error Messages
 - Communications B—83
 - Events (Evnts)
 - Defined 3—1
- Examples**
- Process Control 2—18
 - ZERO/SPAN calibration C—8
 - ZERO/SPAN RTD calibration C—10
- F**
- Fine Tuning
 - Your Process 2—29
 - Function Selection
 - on Graphic Operator Interface 1—4
- G**
- Gain - See PID Constants
 - GPIB Address
 - Primary, IEEE-488 B—89
 - Secondary, IEEE-488 B—89
 - Guaranteed Soak
 - Defined 3—1
- H**
- Handshake Lines
 - IEEE-488 B—92
 - Handshaking
 - X-on/off B—1
 - Hardware B—91
 - Hardware Bus
 - IEEE-488 B—91
 - Help
 - How to Get vii
 - Help Key 1—3
 - Graphic Operator Interface 1—4
- I**
- Icons
 - Graphic
 - Graphic Operator Interface 1—10
 - IEEE-488 Option B—85–B—98
 - Installation and Configuration B—85
 - IFC - see Interface Management Lines, IEEE-488
 - Input Verification
 - Dimension C—4
 - Interface Management Lines
 - IEEE-488 B—93

K

Knob

Multi-Function 1—4

L

Labels

Macro Key, Graphic Operator Interface 1—11

LET - see Commands

System Variable

LET Command

Message B—21

LIST - see Commands

System Variable

Loop Mode

Automatic 2—20

Defined 2—20

Manual 2—20

M

Macro Keys 1—5

Memory Card Information 1—8

Memory Card Operation 1—6

Memory Card Slot 1—5

Memory Cards

Information that can be stored on 1—5

Message Format for a PRINT Command B—13

Message Formats

Network Mode Request and Response B—11

Messages

Network Mode Request and Response B—10

Multiline Commands - see Bus Commands,
IEEE-488**N**

NDAC - see Handshake Lines, IEEE-488

Network Mode B—1, B—8

Setting Up the B—9

Next Segment

Defined 3—2

NRFD - see Handshake Lines, IEEE-488

Number Entry

on Graphic Operator Interface 1—4

O

Open Window Indicator 1—10

Operations Screens

Map of 2—2

Operator

Defined 2—20

P

Parameter Selection

on Graphic Operator Interface 1—4

PID

Defined 2—20

PID Constants

Gain 2—20

Rate 2—20

Reset 2—20

PID Type

Defined 2—21

Dual 2—21

Programmed 2—21

Select Adaptive 2—21

Single 2—21

PID_Group

Defined 3—2

PRINT - see Commands, System Variable

PRINT Command

Response Message to a B—20

Process Variable 2—18

Program Mode

Defined 3—2

Hold

Real-Time Programmer 3—2

PG-END

Real-Time Programmer 3—2

Run

Real-Time Programmer 3—2

Start

Real-Time Programmer 3—2

Programmed PID - See PID Type

Index

R

- Ramp
 - Defined 3—2
- Rate - See PID Constants
- Real-Time Programmer
 - Defined 2—21
- Recipe
 - Defined 3—2
 - Display Profile 3—14
 - View a 3—14
- Recycles
 - Defined 3—2
- REN - see Interface Management Lines, IEEE-488
- Repeat Addressing
 - IEEE-488 B—91
- Request Message Data Field - see Message Formats, Network Mode Request and Response
- Reset - See PID Constants
- Response Message Data Field - see Message Formats, Network Mode Request and Response
- RS-232
 - Cable Length B—2
- RS-232 Connections B—2
- RS-422 Connections B—3

S

- Screen Display Format
 - Graphic Operator Interface 1—9
- Screen Title
 - Graphic Operator Interface 1—10
- Segment
 - Defined 3—2
- Segment Time
 - Defined 3—2
- Select Adaptive PID - See PID Type
- Serial Communications Message Format for a LET
 - Command B—14
- Set EOI with EOS on Write
 - IEEE-488 B—90
- Set EOI with Last Byte of Write
 - IEEE-488 B—90
- Setpoint 2—18

- Setpoint (SP)
 - Defined 2—21
- Setpoint Source
 - Defined 2—21
- Single PID - See PID Type
- Soak
 - Defined 3—2
- Software Driver
 - When a Host Computer is Used B—2
- Span
 - Defined 2—21
- SRQ - see Interface Management Lines, IEEE-488
- STAG - see Commands, System Variable
- STAG Command
 - Message B—24
- STAG Request Error Messages B—24
- Start Date/Time
 - Defined 3—2
- Start of Text - see Message Formats, Network Mode Request and Response
- Station Address - see Message Formats, Network Mode Request and Response
- System Variable Descriptions B—26, B—26-B—81
- System Variable Use
 - Example of B—82

T

- Tagname
 - Defined 2—21
- Terminal Mode B—1, B—5
 - Communications B—5
 - Communications Sequences B—6
 - Error Checking B—6
 - Setting Up the B—7
- Terminal Mode Data Access B—8
- Terminate Read on EOS
 - IEEE-488 B—90
- Terminology
 - Control Loops 2—19
 - Real-Time Programmer 3—1
- Test Program
 - Communications B—17

T (continued)

Thermocouple

Defined 2—21

Time

Current, Display 1—10

Time Left in Program

Defined 3—3

Time Left in Segment

Defined 3—3

Time of Day Start

Defined 3—3

Time of Day Start Enable

Defined 3—3

Time of Day Start Segment

Defined 3—3

Time Proportioned Output

Defined 2—21

Timeout Setting

IEEE-488 B—89

Tuning

Derivative Adjustments (RATE) 2—29

Integral Adjustments (RESET) 2—28

Loop 2—27

Proportional Adjustments (GAIN) 2—28

Tuning Parameters

Defined 2—21

Type of Compare on EOS

IEEE-488 B—90

UUniline Commands - see Bus Commands,
IEEE-488

Using Memory Cards 1—5

V

Variables

System B—22

Verification/Calibration

Equipment Specifications C—2

Procedure C—1

Required Equipment C—1

Required Prerequisites C—1

W

Wiring Diagram

SHIELD Terminal B—4

Worksheets

Real-Time Program 3—5, 3—16, 3—17,
3—18**X**

X-on/X-off Control Characters

Dimension B—5