



Table of Contents

Overview	S-1
Upgrading to Version 4.30	S-1
Engineering Unit Conversion	S-1
PID Control	S-2
Specifications	S-2
Configuring the CO Variable for “Host Source”	S-2
Host Sourcing of “Controller Input Variables”	S-4
Bumpless Transfer	S-5
Host Sourcing of “Controller Constants”	S-5
CO Variable as “Source” to AO Variable	S-5
Accessing Data From a Host	S-6
Data Format for Host Access	S-7
Configuring CO with other variables as “sources”	S-7
Other MDS Variables Sourcing “Controller Input Variables”	S-8
Other MDS Variables Sourcing “Controller Constants”	S-9
Application Example: Proportional-only Ratio Control	S-11
Control Strategy	S-11
MDS Configuration	S-11
Operation	S-12
Programming Toolkit	S-13
Discrete Output with Deadband	S-13
Exponents	S-14
Removing an Open Input From an Averaging Calculation	S-15
Rate of Change Alarm	S-16
Time Delay Deadband	S-17
Sensor Linearization	S-18
Sensor Transfer	S-19
Totalizing	S-20
Resettable Functions	S-21
Example: Manual Reset Alarm	S-21
Example: Variable Frequency Output	S-22
Event or Edge Triggered Latching	S-23
Using an AI as a Discrete Input	S-24
ASCII Protocol Description	S-25
ASCII Message Format	S-25
Field Definitions:	S-25
ASCII Instruction Set	S-26
I/O EQUATION STATION® PID Viewer Program	S-28
Requirements	S-28
Operation	S-28
MDS Communication Settings	S-28
Starting MDSPID	S-28
Trend Options	S-28
Controller Parameters	S-29
Dynamic Data Exchange (DDE)	S-29

List of Tables and Figures

Table S-1. CO Variables Database Map	S-6
Table S-2. Sample Linearization Points	S-18
Table S-3. Query without Checksum	S-25
Table S-4. Query with Checksum	S-25
Table S-5. Response without Checksum	S-25
Table S-6. Response with Checksum	S-25
Table S-7. Start-Delimiter Characters	S-25
Table S-8. ASCII Queries and Responses	S-26
Table S-9. ASCII Field Codes	S-26
Table S-10. Device Identification	S-27
Table S-11. Device Information	S-27
Table S-12. Status Information	S-27
Table S-13. Subsystem Errors	S-27
Table S-14. Variable Configuration	S-27
Table S-15. Coding Variable Types, Binary Format, and Reset	S-27
Table S-16. Data Addresses within MDSPID	S-29
Figure S-1. Entering Engineering Units	S-1
Figure S-2. Converting Engineering Units	S-1
Figure S-3. Configuring a Host Sourced System	S-2
Figure S-4. Configuring the MDS-MAO	S-3
Figure S-5. Converting Input Range and Units	S-3
Figure S-6. Host Sourcing Controller Input Variables	S-4
Figure S-7. Host Sourcing Controller Constants	S-5
Figure S-8. Using a CO Variable as "Source" to an AO Variable	S-5
Figure S-9. Overview of Example Modifications	S-7
Figure S-10. Adding Potentiometers to the Configuration	S-7
Figure S-11. Configuring MDS Variables as Sources	S-8
Figure S-12. Sourcing Controller Constants with MDS Variables	S-9
Figure S-13. Configuring Controller Constants	S-10
Figure S-14. Overview of Example	S-11
Figure S-15. Configuring the MDS with Proportional-Only Control	S-11
Figure S-16. Inputting Controller Input Variables	S-12
Figure S-17. Viewing the Measure Window	S-12
Figure S-18. Configuring a Pump	S-13
Figure S-19. Calculating Exponential Equations	S-14
Figure S-20. Removing Open Input	S-15
Figure S-21. Configuring a Rate of Change Alarm	S-16
Figure S-22. Configuring a Time Delay Deadband	S-17
Figure S-23. Configuring Linearization	S-19
Figure S-24. Totalizing a Flow of Liquid	S-20
Figure S-25. Resetting a Function	S-21
Figure S-26. Using the "max" Function to Latch an Alarm	S-21
Figure S-27. Resetting an Integral using Process Result	S-22
Figure S-28. Triggering Latching with an Event	S-23
Figure S-29. Using an AI as a Discrete Input	S-24

MDS Supplement

Overview

This document explains how to use the features added to the **I/O EQUATION STATION**® with version 4.30 of the configuration software, MDSCNFG. Version 4.30 features PID control and a new engineering units conversion scheme for analog input variables.

Upgrading to Version 4.30

If your **I/O EQUATION STATION**® modules were received with an older version of the configuration software MDSCNFG, you will need to download software into the modules to use the added features. If the modules came with MDSCNFG version 4.30, skip this section. The software version can be found in the “About” screen or on the diskette label.

Note:

Do not begin the upgrade until the modules can be upgraded individually (with only one module on the comm link at a time) and the upgrade can be performed with a direct connection (i.e. not through any type of modem link).

First, upgrade the configuration software on your PC by following the instructions on the “Installation Diskette” for MDSCNFG Version 4.30.

Run the new version of MDSCNFG and, with the **I/O EQUATION STATION**® (MDS) connected to the PC (see page 78 of the manual), select “Utilities” from the main menu bar, then “Reprogram Module”. If the connections from PC to MDS (including RS232-RS485 converter) are correct, the “Module Selection” screen will show the communication details of the MDS being upgraded. (If not, try “Search All”.) Note the address.

Click on “OK”. From “Reprogramming Files”, select the Monitor/Update V3.10 file, and click on “OK”. Select “Utilities-Reprogram Module”. At the “Module Selection” screen, highlight the address of the module being reprogrammed (even though no module information appears beside the address) and then click OK”. “Reprogramming Files” appears. Select the Modbus-RTU V4.30 file for modules with the –MD or –MAO option or version 4.31 for the standard MDS (or Profibus/ASCII V4.30, if Profibus or ASCII is desired), and click on “OK”.

Engineering Unit Conversion

Version 4.30 has a simplified scheme for engineering conversion.

After adding an AI variable and selecting the desired sensor, click on the “Range/Error” field. Fill in the “minimum” and “maximum” range values in terms of the units of the selected sensor. (e.g., units of mA for “Current” sensor, units of V for “Voltage” sensor.)

The conversion to engineering units is defined in the “Format” field (see Figure S-1).

Figure S-1. Entering Engineering Units

Enter the engineering units under “Unit” (in this case, “psig”) and click on “Conversion”. The unit conversion screen appears (see Figure S-2).

Figure S-2. Converting Engineering Units

The sensor in this example is Current, so we convert the fundamental measurement units—or “Input” units—of the selected sensor (in this case Amperes) to psig. Type in the 0 and 100% engineering units values (in this case, 3 and 15) corresponding to the 0 and 100% “Input Units” values.

MDS Supplement

PID Control

One of the MDS variables available with version 4.30 is the Controller variable (CO), which executes a PID equation. The PID equation is a function of input variables and controller constants. Its real time result is the value of the CO variable.

$$CO = K \cdot \{ (SP - PV) + (1/T_i) \cdot \int (SP - PV) dt + T_d \cdot d(PV)/dt \}$$

where K , T_i and T_d are controller constants. K is Proportional Gain, T_i is Integral Time (in minutes), T_d is Derivative Time (in minutes), and SP (setpoint) and PV (process variable) are input variables.

The CO variable may then be used by any other MDS variable (as the source for Analog Output or Digital Output variables, for instance) or as setpoint input for another CO variable if cascade control is required.

Another parameter associated with the CO variable is **Mode**. Mode is a "switch" that can be set to Auto or Manual. In Auto (closed-loop) mode, the value of CO is the result of the above equation. In Manual (open-loop) mode the CO result (independent of controller inputs SP and PV) may be controlled directly.

The CO variable may only be configured in an MDS equipped with either the -MD or -MAO option. Each module can have a maximum of two CO variables.

Specifications

Sample Period: 1 second.

Resolution: 0.01% of control range

Gain Range: 0 - 655.35

Integral Time Range: 0 - 655.35 minutes
(0 means no Integral action)

Derivative Time Range: 0 - 655.35 minutes

Integral Wind-up prevented by non-linear feedback of CO result to Integrator.

"Bumpless Transfer" between Auto and Manual modes if SP and CO are "Host sourced".

Configuring the CO Variable for "Host Source"

All inputs to the CO variable, except PV, can be "sourced" by a host computer (i.e., a host controls their values) or sourced by other variables in the MDS configuration (an Analog Input variable, for instance, configured for potentiometer, to adjust K or SP). Process variable input *must* be sourced by another MDS variable.

The following example demonstrates how to use the MDS for PID control. This configuration requires a host computer to source all inputs to the CO variable (except process variable PV). Later the configuration is modified so that adjustments made from external signal sources are fed into the MDS as Analog Input or Digital Input variables. Figure S-3 gives an overview of a host sourced system.

Figure S-3. Configuring a Host Sourced System

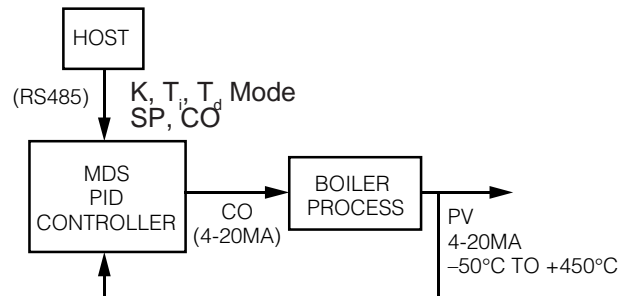


Figure S-4 shows the beginning of an MDS-MAO configuration. Variable 1 is an AI variable configured for 4-20mA input and named ProcVar. A CO variable is then added, and named COVar. (See page 16 of the manual to start a new configuration and add variables.)

MDS Supplement

Figure S-4. Configuring the MDS-MAO

The screenshot shows the 'MDS -MAO Configuration NONAME.ISK' window. The title bar includes 'File', 'Edit', 'Settings', 'Utilities', 'Communication', 'Typeface', and 'Help'. The status bar shows 'Address : ???' and '31.7.96 10:38 AM'. The main area contains a table with the following data:

Var. #	Variable Name	Sensor	Type of M.	Conn.	Term.	Format	Range/Error	Additional
1	AI ProcVar	Current	Current	...	AI 1 C	ffff.fff mA	4.000 20.000	No Filter
2	CO COVar					ffff.fff %	K=1.000 Ti=0.000 Td=0.000	PV = V1 SP = Host

Next, the input range and units of ProcVar are converted from 4-20mA to the engineering units range of -50°C to +450°C. See “Engineering Units Conversion” on page S-1 for conversion information. (Figure S-5 shows the configuration screen.)

Figure S-5. Converting Input Range and Units

The screenshot shows the 'MDS -MAO Configuration NONAME.ISK' window. The title bar includes 'File', 'Edit', 'Settings', 'Utilities', 'Communication', 'Typeface', and 'Help'. The status bar shows 'Address : ???' and '31.7.96 10:39 AM'. The main area contains a table with the following data:

Var. #	Variable Name	Sensor	Type of M.	Conn.	Term.	Format	Range/Error	Additional
1	AI ProcVar	Current	Current	...	AI 1 C	ffff.fff degC	-50.000 450.000	No Filter
2	CO COVar					ffff.fff %	K=1.000 Ti=0.000 Td=0.000	PV = V1 SP = Host

MDS Supplement

Host Sourcing of “Controller Input Variables”

Click on the Additional field of COVar to reveal the “Controller Input Variables” window. Select ProcVar as the source variable under “Process Variable Source” (see Figure S-6).

Figure S-6. Host Sourcing Controller Input Variables

The screenshot shows a configuration window titled "Controller Input Variables". It is divided into several sections:

- Setpoint Source:** Radio button "Host" is selected. The "on power-up" value is 125. There is also an option for "Source Variable" with a dropdown arrow.
- Process Variable Source:** Radio button "Host" is selected. The "Source Variable" dropdown menu shows "ProcVar".
- Control Range:** Two input fields are shown: "0%" with the value "-50" and "100%" with the value "450". To the right, it says "[in engineering units of SP / PV]".
- Mode Switch Source:** Radio button "Host" is selected. There are also radio buttons for "Manual" and "Auto", with "Auto" selected. Below these are options for "Digital Input 1", "Digital Input 2", and "Manual on Variable Result", each with a dropdown arrow.
- CO Manual Mode Source:** Radio button "Host" is selected. The "on power-up" value is 0 %. There is also an option for "Source Variable" with a dropdown arrow.

At the bottom of the window are two buttons: "Ok" and "Cancel".

The “Control Range” is entered in the engineering units of the Process Variable. This defines how the PID controller applies gain in terms of percent error vs. percent output. Although the range entered may differ from the range defined for ProcVar, for this example enter the same range: -50°C to $+450^{\circ}\text{C}$.

“Setpoint Source”, “Mode Switch Source” and “CO Manual Mode Source” must all be set to “Host” for this example. The associated “on power-up” values may also be changed at this time. These values apply until the Host writes new values to SP, Mode, and CO.

Set “on power-up” Mode to “Auto” and enter a Setpoint value of 125. The controller then defaults to Auto mode with a setpoint of 125°C on power-up. The entered value for Setpoint is interpreted in terms of Process Variable engineering units. If “on power-up” mode is set to “Manual”, then the “on power-up” value under “Co Manual Mode Source” becomes the default controller output value. Mode must be entered in percent of control range (i.e. 0-100).

Note:

CO Manual Mode Source controls CO when Mode is Manual. When Mode is Auto, CO is the result of the closed-loop PID equation.

Configured as above, the MDS will initialize as a PID controller in Auto mode with Setpoint at 125°C .

MDS Supplement

Bumpless Transfer

Bumpless transfer of control between Auto and Manual modes occurs when the Setpoint Source and CO Manual Mode Source are set to "Host". In bumpless transfer, the controller output does not change when Mode is switched. When Mode is switched from Manual to Auto, setpoint is updated to the value that maintains controller output at its present value.

Host Sourcing of "Controller Constants"

Clicking on the Range/Error field of COVar yields the "Controller Constants" window (see Figure S-7).

Figure S-7. Host Sourcing Controller Constants

Sources for adjustment of the PID constants may be set to "Host" or "Source Variable". When set to Host, the "initial settings" for the constants are used in the PID equation until the Host writes new values.

Note:

Values entered as "initial settings" for controller constants apply until a Host has written new values to K , T_i and T_d . On subsequent power ups, the last values written by the Host become the "initial settings".

The initial settings are undefined when "Source Variable" is selected as the adjustment source instead of "Host". If the values for K , T_i and T_d are known and will not be changed after installation, select Host and enter the values as initial settings.

Here, we settle for the default initial settings, $K = 1$ and $T_i = T_d = 0$, and assume we will be changing these values on-line from a Host.

CO Variable as "Source" to AO Variable

The value of COVar is the result of the PID control algorithm. This value can be used by other variables in the MDS configuration (i.e. as operand in a formula for an AR variable; as source to AL or DO variables; as source to another CO variable to achieve cascade control; or as source to an AO variable).

To complete the example configuration, an AO variable is added and named CtrlOut. COVar is declared as CtrlOut's source (by selecting "V2" in the Additional field of CtrlOut). Scaling of the output current is set so that 0-100% from COVar yields 4-20mA (by setting the source min & max values to 0 & 100, in the Range/Error field of CtrlOut.) Figure S-8 shows the screen after the addition of CtrlOut.

Figure S-8. Using a CO Variable as "Source" to an AO Variable

Var. #	Variable Name	Sensor	Type of M.	Conn.	Term	Format	Range/Error	Additional
1	AI ProcVar	Current	Current	---	AI 1	ffff.fff degC	-50.000 450.000	No Filter
2	CO COVar					ffff.fff %	K=1.000 Ti=0.000 Td=0.000	PV = V1 SP = Host
3	AO CtrlOut		Curr. Output	---	AO+ AO-	ffff.fff mA	4.000 20.000	Src.Var=V 2

MDS Supplement

Accessing Data From a Host

When this configuration is downloaded to a module, a Host computer can change the controller constants and setpoint, switch the controller mode between Auto and Manual, and control output directly in Manual mode.

The CO variables occupy fixed areas of the database. In Table S-1, “first CO variable” is the first CO variable added to a configuration and “Second CO variable” is the second CO variable added. Data can be accessed in integer or IEEE floating point format.

Note:

When using Modbus-RTU, the physical register address must be modified for SCADA software to a “30,000” or “40,000” decimal register address by adding 30,001 or 40,001 to the decimal physical register address.

Table S-1. CO Variables Database Map

Variables in IEEE, 4-byte Floating Point Format (STD 754-1985)		
Physical Register Address (hi-byte, lo-byte)	Content	Numerical Range
0600, 0601 (1536, 1537)	SP, first CO variable	User-defined Control Range, in engineering units
0602, 0603 (1538, 1539)	PV, first CO variable	User-defined Control Range, in engineering units
0604, 0605 (1540, 1541)	CO, first CO variable	0.00-100.00
0606, 0607 (1542, 1543)	K, first CO variable	0-655.35
0608, 0609 (1544, 1545)	T _i , first CO variable	0-655.35 (in minutes), 0 means no Integral
060A, 060B (1546, 1547)	T _d , first CO variable	0-655.35 (in minutes)
060C, 060D (1548, 1549)	Mode, first CO variable	0 = Auto, 1 = Manual
0700, 0701 (1792, 1793)	SP, second CO variable	User-defined Control Range, in engineering units
0702, 0703 (1794, 1795)	PV, second CO variable	User-defined Control Range, in engineering units
0704, 0705 (1796, 1797)	CO, second CO variable	0-100.00
0706, 0707 (1798, 1799)	K, second CO variable	0-655.35
0708, 0709 (1800, 1801)	T _i , second CO variable	0-655.35 (in minutes), 0 means no Integral
070A, 070B (1802, 1803)	T _d , second CO variable	0-655.35 (in minutes)
070C, 070D (1804, 1805)	Mode, second CO variable	0 = Auto, 1 = Manual
Variables in Integer Format		
Physical Register Address	Content	Numerical Range
0800 (2048)	SP, first CO variable	User-defined Control Range (in engineering units/100)
0801 (2049)	PV, first variable	User-defined Control Range (in engineering units/100)
0802 (2050)	CO, first CO variable	0-10000
0803 (2051)	K, first CO variable	0-65535 (in units of 1/100)
0804 (2052)	T _i , first CO variable	0-65535 (in minutes/100), 0 means no Integral
0805 (2053)	T _d , first CO variable	0-65535 (in minutes/100)
0806 (2054)	Mode, first CO variable	0 = Auto, 1 = Manual
0900 (2304)	SP, second CO variable	User-defined Control Range, in engineering units
0901 (2305)	PV, second CO variable	User-defined Control Range, in engineering units
0902 (2306)	CO, second CO variable	0-10000
0903 (2307)	K, second CO variable	0-65535 (in units of 1/100)
0904 (2308)	T _i , second CO variable	0-65535 (in minutes/100), 0 means no Integral
0905 (2309)	T _d , second CO variable	0-65535 (in minutes/100)
0906 (2310)	Mode, second CO variable	0 = Auto, 1 = Manual

MDS Supplement

Data Format for Host Access

Floating point data is in engineering units (for example, CO has range 0-100.00% and T_d has range 0-655.350 minutes) and integer data is in pseudo-engineering units (CO has range 0-10000 and T_d has range 0-65535 in units of minutes/100). Dividing the real (or floating point) number value by 100 before truncating to an integer allows finer resolution.

For SP and PV, numerical range is defined by the "Control Range" in the configuration program. (See "Host Sourcing of Controller Input Variables" on page S-4.) When accessing the integer registers for SP and PV, format the values in the units defined by "Control Range" and *divided by 100* to increase resolution.

The integer range of a 16-bit value is 0 to 65,535 for unsigned integers and -32,768 to 32,767 for signed integers. When accessing the integer values of SP and PV from a Host, "Control Range" must be limited to 0 to 655.35 for unsigned integer access, and -327.68 to 327.67 for signed integer access. If the process variable range exceeds these limits, the variable acting as "Process Variable Source" will have to be scaled to fit within the limits.

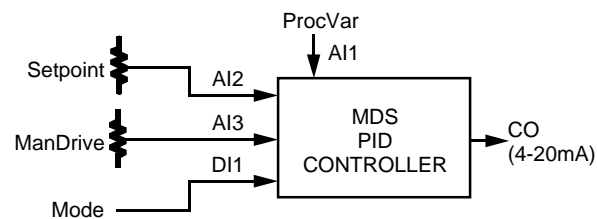
Note:

When accessing the registers holding integer values of SP, PV, CO, K, T_i and T_d , the values are formatted as the real number values divided by 100, and then truncated.

Configuring CO with Other Variables as "Sources"

Instead of having a Host computer as source, any input to the CO variable can be sourced by another variable in the MDS configuration program. To demonstrate, the example will be modified so that physical inputs to the MDS will: adjust the Setpoint from a 10-turn pot; drive the controller output directly with a 10-turn pot when in Manual mode; switch Mode from a discrete input; and adjust Gain **K** from a 10-turn pot. Figure S-9 gives an overview.

Figure S-9. Overview of Example Modifications



To accomplish this, first add 2 AI variables of sensor type Potentiometer, and name them Setpoint and ManDrive (see Figure S-9).

Figure S-10. Adding Potentiometers to the Configuration

MDS -MAO Configuration NONAME.ISK								
File Edit Settings Utilities Communication Typeface Help								
Address : ???							31.7.96 11:02 AM	
Var. #	Variable Name	Sensor	Type of M.	Conn.	Term.	Format	Range/Error	Additional
1	AI ProcVar	Current	Current		AI 1 C	ffff.fff degC	-50.000 450.000	No Filter
2	CO COVar					ffff.fff %	K=1.000 Ti=0.000 Td=0.000	PV = V1 SP = Host
3	AO CtrlOut		Curr. Output		AO+ AO-	ffff.fff mA	4.000 20.000	Src.Var=V 2
4	AI Setpoint	Potentiometer	potentiometri		EX AI 2 C	ffffff.f degC	-50.0 450.0	No Filter
5	AI ManDrive	Potentiometer	potentiometri		EX AI 3 C	ffffff.f %	0.0 100.0	No Filter

MDS Supplement

Other MDS Variables Sourcing “Controller Input Variables”

The ranges and units of variables “Setpoint” and “ManDrive” in Figure S-10 were converted. When Potentiometer is selected, it defaults to a unitless range of 0 to 1 representing the position of the potentiometer wiper as a fraction of end-to-end travel. In “Setpoint” this range has been converted to -50°C to +450°C, the same range and units as ProcVar.

Note:

When another MDS variable is the source for setpoint, that variable must be formatted in the units of the Process Variable.

The variable ManDrive will source the controller output in Manual mode, so the potentiometer range must be converted to 0-100%.

Note:

When another MDS variable is the Manual mode source of CO, that variable must be converted to a range of 0-100%.

Now click on the Additional field of COVar to bring up the “Controller Input Variables” window (shown in Figure S-11).

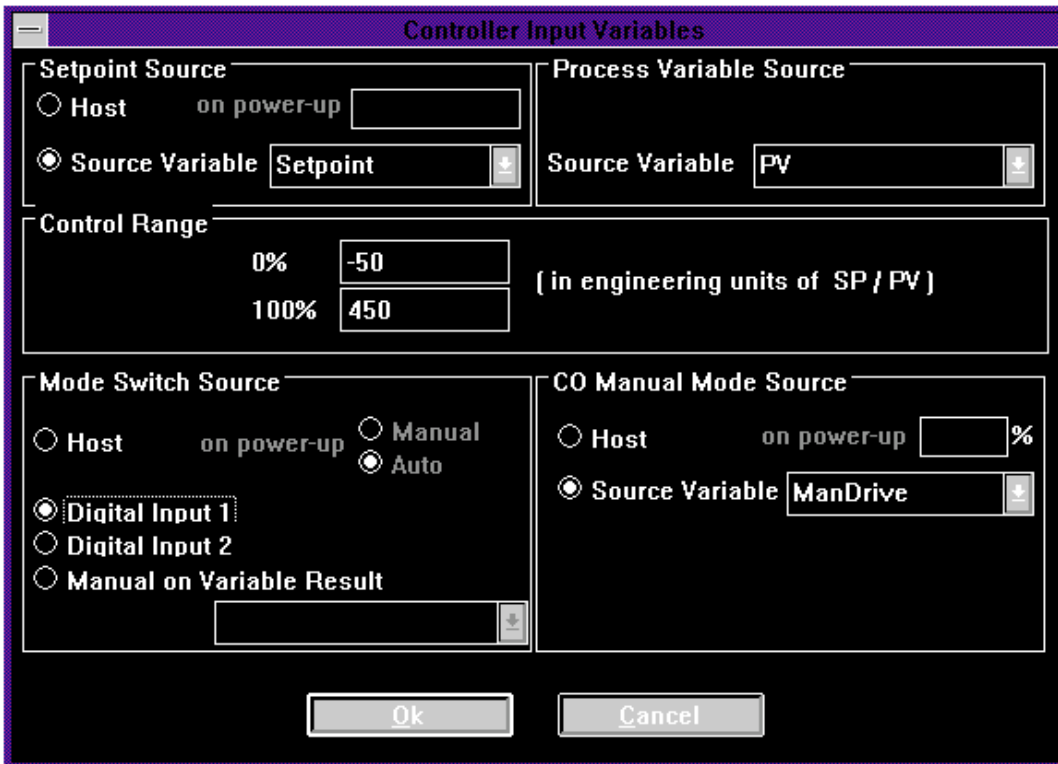
To configure the other MDS variables as “Setpoint Source” and “CO Manual Mode Source”, click the “Source Variable” button, then select a variable from the adjacent list. Select the variable “Setpoint” for the Setpoint Source, and the variable “ManDrive” for the CO Manual Mode Source.

For Mode Switch Source, click the button labeled “Digital Input 1”. Clicking “Manual on Variable Result” allows selection of another variable (an AL variable, for instance, configured to go high when the Process Variable reaches a high alarm level). In this case, the controller would be in Manual mode when the result of the selected variable is greater than 0.5, and in Auto mode otherwise.

Note:

When a “Digital Input” is selected as the Mode Switch source, an active (shorted) input puts the controller in Manual mode.

Figure S-11. Configuring MDS Variables as Sources



MDS Supplement

Other MDS Variables Sourcing “Controller Constants”

For the example, controller gain **K** is adjusted from a 10-turn pot, so another AI variable of sensor type Potentiometer is added and named “Gain” (see Figure S-12).

The default input range of variable “Gain” (0-1) has been converted to 0-10. The engineering unit range of **K**’s source variable must be within the overall range defined in the database map table on page S-7. (In this case, the range is 0-655.35.) This example limits the range of values that “Gain” can map into **K**, to include only the values 0-10.

Note:

*When another MDS variable is the adjustment source for controller constant **K**, T_i or T_d , that source variable’s range must fit within the range defined in Table S-1 on page S-6.*

In this example, T_i and T_d have no adjustment source variables. Their values will be fixed by entering “initial settings” of $T_i = 3$ minutes and $T_d = 0.75$ minutes.

Note:

If another MDS variable is the adjustment source for Integral Time T_i , then $T_i = 0$ is a discontinuity. (i.e. 0 means no integral action, while small T_i means large integral action.) Consequently, the source variable’s range should not include $T_i = 0$.

Figure S-12. Sourcing Controller Constants with MDS Variables

MDS -MAO Configuration NONAME.ISK								
File Edit Settings Utilities Communication Typeface Help								
Address : ???							31.7.96 11:04 AM	
Var. #	Variable Name	Sensor	Type of M.	Conn.	Term	Format	Range/Error	Additional
1	AI ProcVar	Current	Current		AI 1 C	ffff.fff degC	-50.000 450.000	No Filter
2	CO COVar					ffff.fff %	K=1.000 Ti=0.000 Td=0.000	PV = V1 SP = Host
3	AO CtrlOut		Curr. Output		AO+ AO- C	ffff.fff mA	4.000 20.000	Src.Var=V 2
4	AI Setpoint	Potentiometer	potentiometri		EX AI 2 C	ffffff.f degC	-50.0 450.0	No Filter
5	AI ManDrive	Potentiometer	potentiometri		EX AI 3 C	ffffff.f %	0.0 100.0	No Filter
6	AI Gain	Potentiometer	potentiometri		EX AI 4 C	ffffff.f	0.0 10.0	No Filter

MDS Supplement

Click on the “Range/Error” field of COVar: The “Controller Constants” field is shown (see Figure S-13).

Figure S-13. Configuring Controller Constants

Controller Constants

Proportional Gain Adjust

Host initial setting

Source Variable Gain

Integral Time Adjust

Host initial setting minutes

Source Variable

Derivate Time Adjust

Host initial setting minutes

Source Variable

To configure the variable “Gain” as an adjustment source for K , click the “Source Variable” button under “Proportional Gain” and select “Gain” from the list box. Set “Integral Time Adjust” and Derivative Time Adjust” to “Host”, then enter initial settings for T_i and T_d of 3 and 0.75 minutes, respectively.

Note:

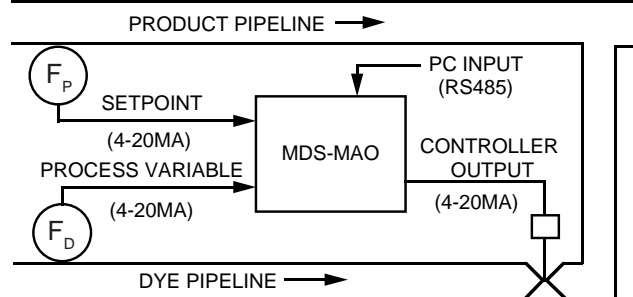
If the values for K , T_i and T_d are known and will not be changed after installation, select Host as adjustment source and enter the known values as “initial settings”.

Application Example: Proportional-only Ratio Control

Without Integral action in a closed-loop controller, non-zero error or a bias term is required to yield non-zero controller output.

Here, the flow of dye added to a chemical delivery pipeline is regulated by an MDS. The amount of dye added is proportional to the flow of product, but the proportions do not have to be exact. Ease of installation—including minimization of controller tuning effort—is more important, so Proportional-only control is used. Figure S-14 gives an overview of the process.

Figure S-14. Overview of Example



Control Strategy

With product flow, F_p , as Setpoint, the MDS controls the dye so that dye flow, F_d , is proportional to F_p . The mixture ratio is determined by the relative scalings of the flow transmitters. For instance, with 50% output from the product flow transmitter, the correct mixture requires 50% output from the dye flow transmitter.

The MDS is configured to accept a bias adjustment at commissioning to nullify error at expected delivery flow.

MDS Configuration

From MDSCNFG, add two AI variables configured for sensor type Current, ranged to 4-20mA input with 0-100% engineering units, and named " F_p " and " F_d ".

Add two SP variables. Range the first, "BIAS", to allow host writes from 0-100%. Range the second, "MODE" (to switch controller mode between Auto and Manual), to write 0 or 1. Select "Host", and "Save to Non-volatile Memory", within the "Range/Error" field, to allow PC adjustments at commissioning (see page 50 of the manual). Figure S-15 shows the Configuration Screen.

The remaining variables are:

- "CTRLR", a CO variable computing control equation;
- "CTRLout", an AO (Analog Output) variable sourced by SUM and supplies the valve actuator current;
- "SUM", an AR (Arithmetic) variable that adds BIAS to CTRLR when MODE = 0 (Auto mode), and is simply BIAS when MODE = 1.

The formula entered for SUM is:

$$\text{SUM} = \text{BIAS} + (1 - \text{MODE}) \cdot \text{CTRLR}$$

Figure S-15. Configuring the MDS with Proportional-Only Control

MDS -MAO Configuration NONAME.ISK								
File Edit Settings Utilities Communication Typeface Help								
Address : 001		Prop-only w/ Bias				28.8.96 11:22 AM		
Var. #	Variable Name	Sensor	Type of M.	Conn.	Term.	Format	Range/Error	Additional
1	AI Fp	Current	Current	...	AI 1	ffff.fff %	0.000 100.000	No Filter
2	AI Fd	Current	Current	...	AI 2	ffff.fff %	0.000 100.000	No Filter
3	SP BIAS					ffff.fff %	Default Valu	
4	SP MODE					f	Default Valu	
5	CO CTRLR					ffff.fff %	K=5.000 Ti=0.000 Td=0.000	PV = V2 SP = V1
6	AR SUM					ffff.fff %	Reset	V3 + [1 - V4
7	AO CTRLout		Curr. Output	...	AO+ AO-	ffff.fff mA	4.000 20.000	Src.Var=V 6

MDS Supplement

Click on the "Range/Error" field of CTRLR and enter an "initial setting" of 5 for "Proportional Gain Adjust". A gain of 5, in this case, is a compromise between control loop stability and Setpoint variation error.

Click on CTRLR's Additional field to open the Controller Input Variables Window (see Figure S-16). F_p is Setpoint, and F_d is Process Variable. "Control Range" is equal to the flow ranges of 0-100%. "Mode Switch Source" is set to "Manual on Variable Result: MODE > 0.5". Writing 1 to MODE from a PC opens the control loop, writing 0 closes it.

Operation

At commissioning, the User can observe variables and change writable variables on-line from a PC running MDSCNFG. By clicking "Utilities" on the main menu bar and selecting "Measure", the Measure Window appears (see Figure S-17).

In Manual mode, $SUM = BIAS$. At commissioning, put the controller in Manual mode and set BIAS, the direct valve actuating signal, to the desired level. F_p is at operational level and error between F_p and F_d is zeroed by adjusting BIAS. In Auto MODE, the actuating signal is $SUM = CTRLR + BIAS$. Zero closed-loop error will occur with Setpoint, F_p , at operational flow level.

Figure S-17. Viewing the Measure Window

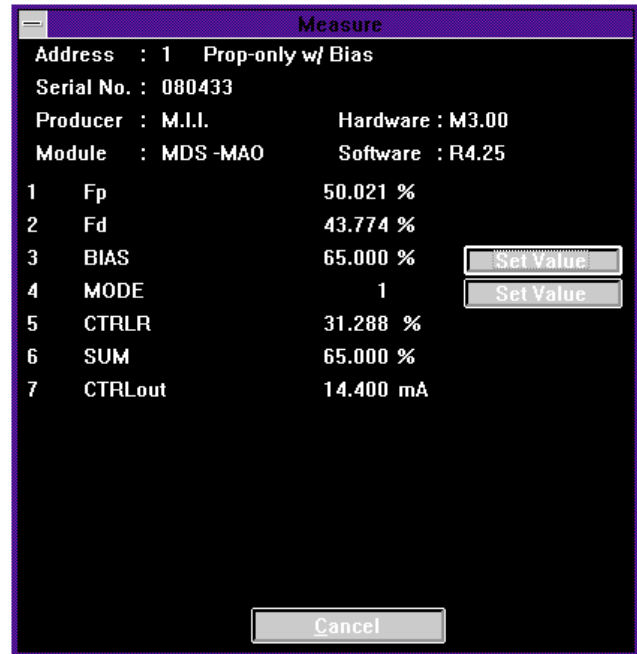
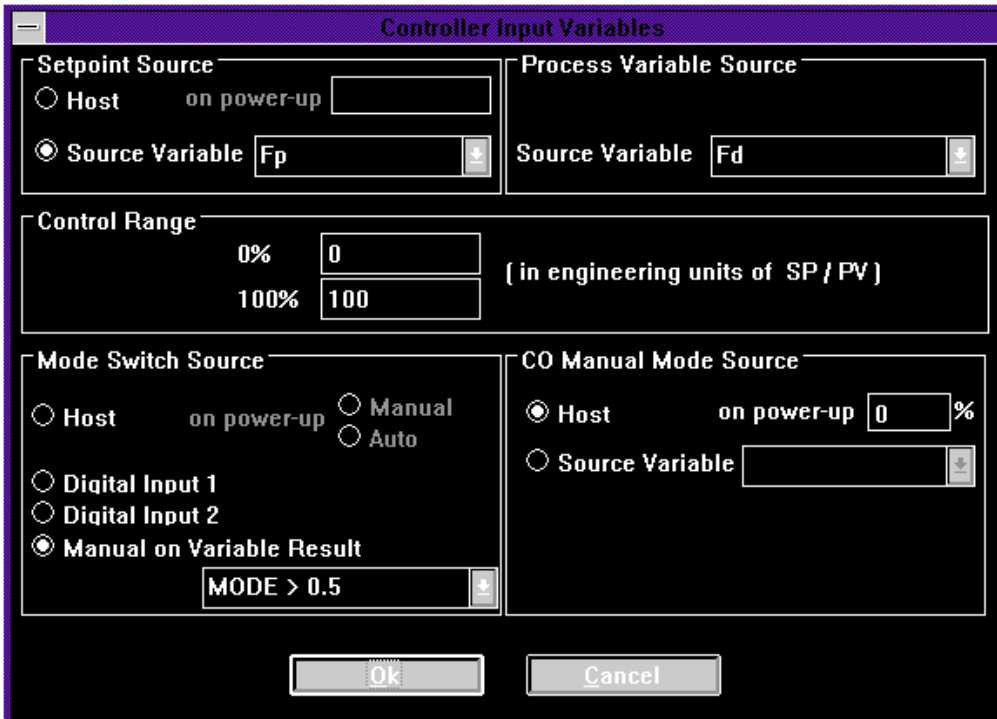


Figure S-16. Inputting Controller Input Variables



Programming Toolkit

This section contains several commonly used programming techniques for creating **I/O EQUATION STATION**[®] configurations with the configuration software MDSCNFG.

Discrete Output with Deadband

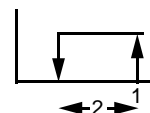
In alarm applications, or where a discrete output is used as an on/off controller, deadband is often required. Each discrete output, when configured as a “Process Out” measurement type, is equipped with four adjustable deadband configurations: High Failsafe, High Non-failsafe, Low Failsafe and Low Non-failsafe. These configurations require assigning values for “Threshold 1” and “Threshold 2”, where Threshold 1 is the trip point and Threshold 2 is the deadband.

In this example, the DO variable “Pump” will turn a pump on and off. The pump is activated when the level exceeds 60 feet and is de-activated when the level drops below 40 feet (see Figure S-18).

Figure S-18. Configuring a Pump

Var. #	Variable Name	Sensor	Type of M.	Conn.	Term.	Format	Range/Error	Additional
1	AI Level	Current	Current		AI 1 C	ffff.fff ft	0.000 100.000	No Filter
2	DO Pump		Process Out		+ DO 1	f	Independent	Thresholds

¹ Source: Level
 Threshold Type: High Non-failsafe
 Threshold1: 60
 Threshold2: 20



MDS Supplement

Exponents

Applications that require calculating exponential equations can use the logarithmic functions (i.e., “ln” and “exp”) provided in modules with the -MD or -MAO options. The following equation calculates exponential equations:

$$x^y = e^{(y \ln(x))}$$

In this example, 2 AI variables, “X” and “Y”, convert 4-20mA to engineering units and an AR variable, “A”, calculates x^y (see Figure S-19).

Figure S-19. Calculating Exponential Equations

Var. #	Variable Name	Sensor	Type of M.	Conn.	Term	Format	Range/Error	Additional's
1	AI X	Current	Current	⊕	AI 1	ffff.fff ft	0.000 100.000	No Filter
2	AI Y	Current	Current	⊕	AI 2	ffff.fff ft	0.000 100.000	No Filter
3	AR A					ffffff	Reset	exp (V1 * ln (

$$^1 A = \exp(X \cdot \ln(Y))$$

Removing an Open Input From an Averaging Calculation

In a signal averaging application, it is often desirable to obtain an average based only on good process signals. Upon open input, the system must detect the failure and eliminate the input from the averaging equation.

When an input to the **I/O EQUATION STATION**® opens, internal circuitry drives the reading out of range. Thermocouple and Voltage inputs are driven upscale while current inputs are driven down-scale. A multi-wire sensor (i.e. RTD, potentiometer, bridge) goes up- or downscale depending on which wire opens. To detect the failed sensor, an alarm (AL) variable changes state when an input reading goes out of range. This logic is then used to eliminate the sensor from the averaging equation.

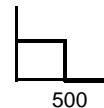
In Figure S-20, two type K thermocouple sensors connected to a boiler are averaged. Upon open input, the average temperature will be represented by the remaining active sensor.

The status of T1 and T2 are indicated by the AL variables T1GD and T2GD. This logic could also be made externally available by using DO variables configured as "Process Controlled" Type of Measurement instead of AL variables. Using this method, digital output 1 can be assigned to change state when T1 opens and digital output 2 can be assigned to T2.

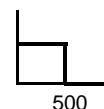
Figure S-20. Removing Open Input

Var. #	Variable Name	Sensor	Type of M.	Comp.	Term.	Format	Range/Error	Additional
1	AI T1	TC Type K	Differential	↘	AI 1 C	#####.1 °F	-0.0 500.0	No Filter
2	AI T2	TC Type K	Differential	↘	AI 2 C	#####.1 °F	-0.0 500.0	No Filter
3	AL T1GD					f		Thresholds
4	AL T2GD					f		Thresholds
5	AR #Good					#####	Reset	V3 + V4
6	AR Average					#####	Reset	(V3 * V1 + V
7	AI CJC	Cold Junc. Terminal CJ (2)		↘	AI 3 AI 4 C	#####.1 °C	-40.0 80.0	No Filter

¹ Source: T1
Change: Low Logic
Threshold: 500



² Source: T2
Change: Low Logic
Threshold: 500



³ Good = T1GD + T2GD

⁴ Average = [T1GD•T1+T2GD•T2] / #Good

MDS Supplement

Rate of Change Alarm

Analog signal rate of change can be measured by taking the derivative of an AI variable. The derivative function is available in -MD or -MAO modules.

The “deriv” function determines rate of change by taking the difference between two consecutive samples of the analog signal, and dividing by the sample period. The derivative’s time unit is seconds.

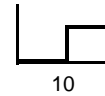
In this example, conveyor belt speed is monitored. The variable “ACCEL” is the derivative of belt speed (in units of RPM per second). “ABSVAL” is its absolute value. The MDS’s max function is a peak detector, so the variable “MAX” is the maximum rate of change.

The DO variable “ALARM” activates and latches when MAX exceeds the rate of change threshold of 10 RPM per second. When the DI variable “RST” is equal to 1 (activated by a contact closure input), MAX is reset, and ALARM becomes inactive (see Figure S-21).

Figure S-21. Configuring a Rate of Change Alarm

Var. #	Variable Name	Sensor	Type of M.	Conn.	Term.	Format	Range/Error	Additional
1	AI SPD	Current	Current		AI 1 C	ffff.f# RPM	0.000 100.000	No Filter
2	AR ACCEL					ffffff#	Reset	deriv (V1)
3	AR ABSVAL					ffffff#	Reset	abs (V2)
4	AR MAX					ffffff#	Reset	max (V3)
5	DO ALARM		Process Out		+ I/O 1	f	Independent	Thresholds
6	DI RST		Status		I/O 2	f		

- 1 ACCEL = deriv (SPD)
- 2 ABSVAL = abs(ACCEL)
- 3 MAX = max(ABSVAL); reset on RST.0.5
- 4 Source: MAX
Change: High Logic
Threshold: 10



MDS Supplement

Time Delay Deadband

In some alarm applications, it is desirable to ignore transients and noise spikes and only trip when the dc level exceeds the trip point. To do so, insert a timer in the logic path between detecting the alarm condition and the actual tripping of the alarm output. If the process variable exits alarm condition before the time delay period, the output will not trip and the timer is reset.

In Figure S-22, an AL (Alarm) variable, "ALM", goes high when the process variable, "TEMP", exceeds "TP" (for trip point). TP and the time delay variable, "DLY", are SP (Setpoint) variables, which are programmed to be written to by a Host computer. The written values are saved to non-volatile memory.

Since the integrator is based on time units of seconds, taking the integral of ALM, in variable "TMR", yields a timer when ALM goes high (i.e., 1) that counts off seconds. TMR is only enabled when the process variable is in alarm condition. The actual alarm output, the DO variable "ALOUT", is active only after the value of TMR exceeds DLY.

The Timer variable, TMR, requires a reset mechanism. Create another AL variable, called "NOTALM" (the logical inverse of variable ALM) and program TMR to "Reset on NOTALM > 0.5". Then, when TEMP goes below TP, TMR gets reset to 0. If the reset does not occur before TMR reaches DLY, ALOUT is not activated.

Figure S-22. Configuring a Time Delay Deadband

Var. #	Variable Name	Sensor	Type of M.	Conn.	Term	Format	Range/Error	Additional
1	AI TEMP	Pt 100	Resist. 3 wir		AI 1 AI 2 C	ffffff.f °F	-0.0 500.0	No Filter
2	SP TP					ffffff	Default Valu	
3	SP DLY					ffffff	Default Valu	
4	AI ALM					f		Thresholds
5	AR TMR					ffffff	Reset	integ (V4)
6	DO ALOUT		Process Out		+ VQ 1	f	Independent	Thresholds
7	AI NOTALM					f		Thresholds

1 0-500°F, Host-selected, Save to non-volatile

2 0-60 seconds, Host-selected, Save to non-volatile

3 Source: TEMP
Change: High Logic
Threshold: TP

4 TMR = integ(ALM), Reset on NOTALM > 0.5

5 Source: TMR
Change: High Logic
Threshold: DLY

6 Source: TEMP
Change: Low Logic
Threshold: TP

MDS Supplement

Sensor Linearization

The sensor definition function can linearize any standard sensor. You can create new RTD or Thermocouple tables or transform a signal from a level transmitter to volume of liquid in a container.

To define/linearize a sensor:

1. Click on the “sensor” field of an AI variable to access the Sensor Selection screen. Click on “Define” to access the Sensor Database screen.
2. From this screen add, copy, delete or change the sensors in the database. Click on “Add” to define a new sensor.
3. Enter the following information: Name of sensor, Principles (type of sensor), Unit, Min and Max Values of the input signal. For this example, a 1-5 volt signal from a level transmitter is converted to a non-linear volume, so a “1” is entered for Min Value, a “5” for Max Value, and “V” for Unit.
4. Click on “LINEARIZATION” to access the linearization table for the sensor.
5. Calculate the Measured and Displayed values (See Table S-2).

A list of ordered pairs (Level, Volume) defines the process. Each pair is converted to a new pair (Displayed Value, Measured Value) where the Displayed Value Range and the Measured Value range are both 1-5 volts.

Note:

The range of Displayed Values must equal the range of Measured Values.

To calculate the Measured Value from Level:

$$\text{Measured Value} = \text{Mmin} + [(\text{Mmax} - \text{Mmin}) \div (\text{Lmax} - \text{Lmin})] \bullet (\text{Level} - \text{Lmin}) = 1.00 + 0.02 \bullet (\text{Level} - 50)$$

where Mmin = minimum Measured Value; Mmax = maximum Measured Value; Lmax = maximum Level; and Lmin = minimum Level.

To calculate the Displayed Value from Volume:

$$\text{Displayed Value} = \text{Dmin} + [(\text{Dmax} - \text{Dmin}) \div (\text{Vmax} - \text{Vmin})] \bullet (\text{Volume} - \text{Vmin}) = 1.00 + 0.1081 \bullet (\text{Volume} - 8)$$

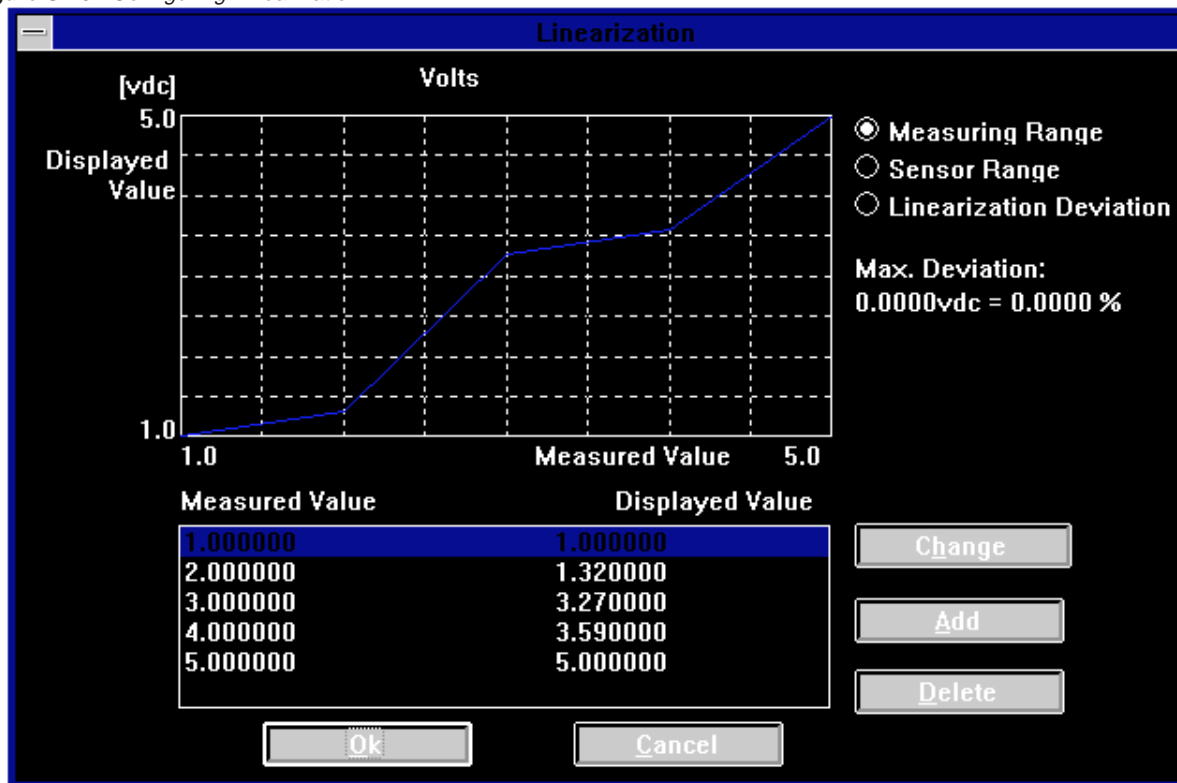
where Dmin = minimum Displayed Value; Dmax = maximum Displayed Value; Vmin = minimum Volume; and Vmax = maximum Volume.

Table S-2. Sample Linearization Points

Level (in inches)	Measured Value (in volts)	Volume (in cubic feet)	Displayed Value (in volts)
Lmin = 50	Mmin = 1.00	Vmin = 8	Dmin = 1.00
100	2.00	11	1.32
150	3.00	29	3.27
200	4.00	32	3.59
Lmax = 250	Mmax = 5.00	Vmax = 45	Dmax = 5.00

6. Click on “Add” and then enter the “Measured” and corresponding “Displayed” values. (See Table S-2.)
7. Click on “OK” to save and exit linearization. Once the table has been generated, the linearized value can be scaled to the desired engineering units (8-45 psig) in the “Format” field of the input variable.

Figure S-23. Configuring Linearization



Sensor Transfer

To transfer a user-defined sensor from one PC to another, create a configuration file (*.ISK) that uses the sensor and transfer the file. To transfer the sensor:

1. Select "File" from the main menu bar, then "New" to create a configuration file. Add an AI variable of the sensor type to be transferred.

Note:

Do not change any of the entries in the Range/Error or format fields.

2. Select "Save" from the "File" menu and then save the configuration file (*.ISK) to a floppy diskette.
3. From the PC running MDSCNFG, open the file with the User-defined sensor (select "Open" from the "File" menu). MDSCNFG will detect the new sensor and the following message will appear:

"The Sensor <new sensor's name> is not in the Sensor Database. Shall this Sensor be added to the Sensor Database? "

4. Click on "Yes".
5. Before creating a configuration using the new sensor, go to the Linearization screen and click on the "Sensor" field for the AI variable.
6. Click on "Define" and select the new sensor from "Sensor Database".
7. Click "Change" and then click "Linearization" in the "Change Sensor" screen.
8. The table and graph for the new sensor will be on the "Linearization" screen. Click on "OK".
9. Click "OK", "End" and "OK" to exit.

This new sensor can also be used in other configurations.

MDS Supplement

Totalizing

Totalizing integrates a signal proportional to, for instance, flow or power. The integral over a given period of time represents the amount of product or energy delivered during that time. The Integrator function is available in the extended math package of the **I/O EQUATION STATION**® (MDS with -MD or -MAO option).

Time units of seconds are used in the integrator. Thus, for a constant input, the integrator output increases in time at a rate of 1 Count / Unit Input / Second:

$$\text{integ}(C) = Ct,$$

where C is a constant and t is the time in seconds, and assuming $\text{integ}(C) = 0$ at time $t = 0$.

In the following example, flow of a liquid is totalized based on the linear signal provided by a flow meter (see Figure S-24).

An AI variable (FLOW) measures 4-20mA from the flow meter output and converts the signal to engineering units of 0-100 gallons per minute. If we assume the flow signal is at full scale (100 GPM), the integral of FLOW will increase at a rate of 100 counts/second. The count after 1 second will be 100 (assuming an initial condition with the integrator output zero), the count after 1 minute will be 6000, etc.

In engineering units (i.e., gallons), the amount of liquid transferred is calculated by dividing the integrator output by the number of seconds in the input variable's time base (i.e., 60, for gallons per *minute*).

A practical totalizing application requires resetting the integrator. Clicking on the "Range/Error" field of an AR variable brings up the "Reset" window, which applies if the formula for the AR variable contains integral, maximum or minimum functions. (See Example: Variable Frequency Output, on page S-22.)

Figure S-24. Totalizing a Flow of Liquid

Var. #	Variable Name	Sensor	Type of M.	Conn.	Term	Format	Range/Error	Additional
1	AI FLOW	Current	Current		AI 1 C	###.### GPM	0.000 100.000	No Filter
2	AR INTEGRAL					#####	Reset	Integ (V1)
3	AR TOTAL					#####	Reset	V2 / 60

¹ INTEGRAL = integ(flow)

² TOTAL = integrator / 60

MDS Supplement

Resettable Functions

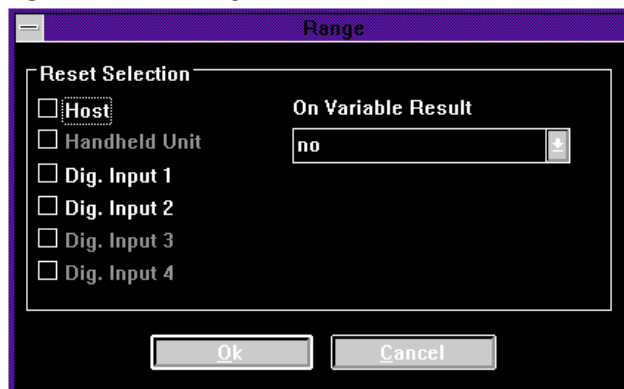
Three MDS math functions have a reset property: maximum, minimum and integral.

The max and min functions accept a single operand and are “peak” and “valley” detectors. So, if a configuration has an AI variable named “PV” and an AR variable named “MOST”, and $MOST = \max(PV)$, then MOST is the maximum value of PV since the last time MOST was reset. Upon reset, MOST is set to the present value of PV.

The integral function accepts a single operand and is the time integral of the operand. When an AR variable with an integ function is reset, the integral becomes 0.

To configure how the reset is generated, click on the “Range/Error” field of an AR variable with a resettable function. The following window appears:

Figure S-25. Resetting a Function



The reset can be generated in 3 different ways:

“**Direct Digital Input**” Reset allows a contact closure input to a digital I/O terminal to reset a resettable function by clicking on “Dig. Input 1” or “Dig. Input 2”.

“**Host**” Reset allows a reset when the Host writes to the variable containing the resettable function.

Note:

On a Host generated reset, the value the Host writes to the AR variable containing the resettable function is irrelevant. An integral is set to zero, and maximum or minimum is set to the present value of the operand.

“**On Variable Result**” Reset allows reset when another variable in the present configuration exceeds 0.5. Click on the arrow for the list of configured variables. You can select an AL (Alarm) variable to generate the reset when the AL variable changes state from 0 or 1.

Example: Manual Reset Alarm

In Figure S-26, the “max” function latches an alarm that is only reset by a contact closure to a discrete input.

The AL variable “ALARM” is high when “TMP” > 425°C. Using the max function, “LATCH” goes and stays high when ALARM is high. The max function in LATCH is reset by input to DI1 (i.e. if TMP goes below the high alarm threshold of 425°C and a contact closure has been applied to DI1). The DO variable “ALOUT” supplies a discrete output that is active when LATCH is high.

Figure S-26. Using the “max” Function to Latch an Alarm

Var. #	Variable Name	Sensor	Type of M.	Conn.	Term.	Format	Range/Error	Additional
1	AI TMP	Pt 100	Resist. 3 wir		AI 1 AI 2 C	ffffff.1 °C	0.0 500.0	No Filter
2	AL ALARM					f		Thresholds
3	AR LATCH					ffffff	Reset	max(V2)
4	DO ALOUT		Process Out		+ VO 1	f	Independent	Thresholds

¹ Source: TMP
Change: High Logic
Threshold: 425

² LATCH = max(ALARM), Reset on DI 1

³ Source: LATCH
Change: High Logic
Threshold: 0.5

MDS Supplement

Example: Variable Frequency Output

In this example, an integral is reset by a DO variable changing state from 0 to 1 (see Figure S-27). An AI variable configured for potentiometer input, "FRQ", is scaled to an engineering units range of 0-100 p/m (pulses per minute). "INT" takes the integral of FRQ. The integral time base is seconds, so if FRQ = 1, INT will ramp up at a rate of 1 count per second. If, for FRQ = 1, you want 1 pulse output after 1 minute, configure a DO variable, "SIXTY", to change state when INT reaches 60.

INT is configured to reset when SIXTY > 0.5 (i.e., when SIXTY changes state.). The **I/O EQUATION STATION**[®] executes each variable in sequence—variable 1, then 2, then 3; then back to 1. During each cycle, FRQ (the potentiometer input) is updated, then INT is updated. Next, the value of SIXTY is checked to see if the integral should be reset or updated, then SIXTY checks if INT > 60. Until this is true, INT = 0.

Eventually, an execution cycle occurs when the value of INT reaches 60. At that time, SIXTY changes its value from 0 to 1; FRQ is updated; INT checks its reset condition and, since SIXTY > 0.5 the integral is reset to 0. Next, SIXTY sees that INT < 60 and changes its value back to 0.

The output pulse duration is approximately 1 execution cycle, which in this configuration is 8 milliseconds. (See Table 2 on page 8 of the Manual.) The execution cycle time is also the resolution of the output waveform. The period of the output waveform can never be less than twice the execution cycle time.

Figure S-27. Resetting an Integral using Process Result

Var. #	Variable Name	Sensor	Type of M.	Conn.	Term.	Format	Range/Error	Additional
1	AI FRQ	Potentiometer	potentiomet		EX AI 1	ffffff.f p/m	0.0 100.0	No Filter
2	AR INT					ffffff	Reset	integ (V1)
3	DO SIXTY		Process Out		+ DO 1	f	Independent	Thresholds

¹ INT = integ(FRQ), Reset on 1 MIN > 0.5

² Source: INT
Change: High Logic
Threshold: 60

MDS Supplement

Event or Edge Triggered Latching

Latching can be triggered by a logic variable changing state. A DI variable representing a contact closure input or an AL variable that changes state when a given event takes place can trigger latching.

In this example, an event is defined by the tank level exceeding 70%. When this occurs, a monitored temperature is latched (see Figure S-28).

The AL variable "EDG" goes from low to high when "LVL" increases through 70%. "EDGNOT" is the logical inverse of EDG. "TRGR" (the product of EDG and EDGNOT) should always be 0. Since the **I/O EQUATION STATION**® executes each variable in sequence (i.e., variable 1, then 2, then 3, etc., and then back to 1), TRGR becomes 1 (i.e., for 1 execution cycle) as LVL increases through 70%.

LVL was less than 70% going into the execution cycle when the triggering event occurs, so EDG = 0, EDGNOT = 1 and TRGR = 0.

First, LVL is updated to a value greater than 70%, then EDG changes from 0 to 1. Next, TRGR becomes 1, since EDG = 1 and EDGNOT = 1, and then EDGNOT changes from 1 to 0. In the next execution cycle, TRGR will go back to 0.

TRGR = 1 for only 1 unique execution cycle. (Note that TRGR will remain 0 when LVL *decreases* through 0 as well as during steady state.)

The variable "LTCH" is the sum of 2 terms, one of which is always 0. So long as TRGR = 0, LTCH = LTCH, (i.e. its value doesn't change). When TRGR = 1 (this is a momentary event as LVL increases through 70%), LTCH = T (i.e. it has a new value assigned to it).

Figure S-28. Triggering Latching with an Event

Var. #	Variable Name	Sensor	Type of M.	Comp.	Term.	Format	Range/Error	Additional
1	AI LVL	Current	Current		AI 1 C	ffff.fff %	0.00 100.000	No Filter
2	AI EDG					f		Thresholds
3	AR TRGR					fffffff	Reset	V2 * V4
4	AI EDGNOT					f		Thresholds
5	AI T	Pt 100	Resist. 3 wr		AI 2 AI 3 C	fffffff.1 °F	-0.0 500.0	No Filter
6	AR LTCH					fffffff	Reset	V3 * V5 + (1 -

¹ Source: LVL
Change: High Logic
Threshold: 70

² TRGR = EDG • EDGNOT

³ Source: LVL
Change: Low Logic
Threshold: 70

⁴ LTCH = TRGR • T + (1 - TRGR) • LTCH

MDS Supplement

Using an AI as a Discrete Input

Sometimes the *I/O EQUATION STATION*® would fit an application if the MDS had 1 or 2 more discrete inputs. An analog input channel can also determine the status of a contact closure input.

Add an Analog Input variable configured for sensor type “Resistance” and 2-wire “Type of Measurement”. Set the ohm range to 0-1000 ohms and convert it to an engineering units range of 1-0 (1 for 0 ohms, 0 for 1000 ohms). This yields the same logic as a DI status input (contact closure, or 0 ohms, is logic 1, while open input is logic 0).

In the following example (Figure S-29), a contact closure applied to the AI variable “STATUS” will result in STATUS = 1 when the contact is open and STATUS = 0 when the contact is closed.

Figure S-29. Using an AI as a Discrete Input

The screenshot shows a software window titled "MDS-MAO CONFIGURATION NONAME.ISK" with a menu bar (FILE, EDIT, SETTINGS, UTILITIES, COMMUNICATION, TYPEFACE, HELP) and a status bar (Address : ???, 4.2.97 3:28 PM). The main content is a table with the following data:

Var. #	Variable Name	Sensor	Type of M.	Conn.	Term.	Format	Range/Error	Additional
1 AI	STATUS	Resistance	Resist. 2 wire	↕	AI 1 C	#####	1 0	No Filter

ASCII Protocol Description

This section gives an overview of the ASCII protocol that can be used with the *I/O EQUATION STATION*[®]. To use ASCII protocol, the unit must be programmed with the Profibus/ASCII protocol file.

To program the module with the Profibus ASCII file, select "Utilities" from the main menu bar and then "Reprogram Module". If the connections from PC to MDS (including the RS232-RS485 converter) are correct, the "Module Selection" screen will show the communication details of the MDS being upgraded. (If not, try "Search All".) Click on "OK". From the "Reprogramming Files" list, select the Profibus/ASCII file and click on "OK".

ASCII Message Format

An ASCII message sent from a host to a module (Slave) is called a "Query". When a Slave receives a Query, it replies with a "Response" message.

The following tables show the Query and Response message formats, identifying the fields within the message and the number of characters in each field.

Table S-3. Query without Checksum

SD	DA	ReqData	ED
1	2	n	1

Table S-4. Query with Checksum

SD	DA	ReqData	CS	ED
1	2	n	2	1

Table S-5. Response without Checksum

SD	ResData	ED
1	n	1

Table S-6. Response with Checksum

SD	ResData	CS	ED
1	n	2	1

or

ACK
1

or

NAK
1

Field Definitions:

ACK Acknowledge (Length = 1 character) A Slave response to a query, that does not require return data, is a single "Acknowledge" character (HEX 06), signifying the orderly execution of the instruction.

CS Check Sum (Length = 2 characters) The Check Sum field, CS, is the binary byte-wise sum of the message, modulo 256. It is calculated by summing the Start-Delimiter (SD), Device Address (DA) and Data (ReqData or ResData) fields.

Checksum_ASCII = [SD+DA+Data] mod 256.

The value is given as a two digit ASCII coded HEX number: ASCII "00"..."FF".

DA Device Address. (Length = 2 characters) The Device Address field, DA, only appears in a Query message. It identifies the Slave being queried. Device Addresses are in the range 1 to 127. The value is given as a two digit ASCII coded HEX number (i.e.ASCII "01".."7F").

ED End Delimiter (Length = 1 character) The End Delimiter field, ED, identifies the end of the message. It is the carriage return character (HEX 0D).

NAK No Acknowledge (Length = 1 character) When a request to the Slave is invalid, the Slave responds with a "No Acknowledge" character (HEX 15).

ReqData Request Data (Length = 1 ... n characters) The Request Data field contains the host's instruction and data for the indicated Slave.

ResData Response Data (Length = 1 ... n characters) The Response Data field contains the Slave's reply to the host's instruction.

SD Start-Delimiter. (Length = 1 character) The Start-Delimiter field, SD, is a single ASCII character that marks the beginning of a message. Its value identifies whether or not a checksum will be included in the message.

Table S-7. Start-Delimiter Characters

SD	Query	Response
with checksum	#	>
without checksum	\$	=

MDS Supplement

ASCII Instruction Set

Table S-8 lists Queries and expected Slave Responses. Table S-9 explains the field codes used in Table S-8. Tables S-10-S-15 give Device Identification, Device Information, and Status Information.

Table S-8. ASCII Queries and Responses

Checksum	Query	Response with Orderly Execution	Response in Case of Error
Read Device Identification			
with	# aa V cc <cr>	>v..v cc <cr>	NAK
without	\$ aa V <cr>	= v..v <cr>	NAK
Read Device Information			
with	# aa S cc <cr>	>s..s cc <cr>	NAK
without	\$ aa S <cr>	=s..s <cr>	NAK
Read Status Information			
with	# aa Z cc <cr>	z..z cc <cr>	NAK
without	\$ aa Z <cr>	= z..z <cr>	NAK
Read Variable Information			
with	# aa B k cc <cr>	>b..b cc <cr>	NAK
without	\$ aa B k <cr>	=b..b <cr>	NAK
Read Variable Value			
with	# aa R k cc <cr>	>d..d cc <cr>	NAK
without	\$ aa R k <cr>	=d..d <cr>	NAK
Write Value to a Variable			
with	# aa W k d..d cc	ACK	NAK
without	\$ aa W k d..d <cr>	ACK	NAK
Reset Variable			
with	# aa D k cc <cr>	ACK	NAK
without	\$ aa D k <cr>	ACK	NAK

Table S-9. ASCII Field Codes

Code	Definition	Length	Range
#	Start delimiter for query with checksum	1	ASCII “#”
>	Start delimiter for response with checksum	1	ASCII “>”
\$	Start delimiter for query without checksum	1	ASCII “\$”
=	Start delimiter for response without checksum	1	ASCII “=”
<cr>	End delimiter (carriage return)	1	HEX 0D
ACK	Positive Acknowledge	1	HEX 06
NAK	No Acknowledge	1	HEX 15
aa	Destination Address	2	ASCII “01”..“7F”
cc	Checksum	2	ASCII “00”..“FF”
k	Variable Number	1	ASCII “1” “C”
d..d	value as formatted for the channel	max. 8	ASCII - String
v..v	Device Identification (see below)	26	ASCII - String
s..s	Device Information (see below)	27	ASCII - String
z..z	Status Information (see below)	2	ASCII - String
b..b	Variable Configuration (see below)	29	ASCII - String

MDS Supplement

Table S-10. Device Identification

Device Identification (v..v)	Length (26 Characters Total)
<vendor name>	ASCII 8 characters
<module name>	ASCII, 8 Characters
<hw-revision>	ASCII, 5 Characters ("_x.xx")
<sw-revision>	ASCII, 5 Characters ("_x.xx")

Table S-11. Device Information

Device Information (s..s)	Length (27 Characters Total)
<location>	ASCII, 20 Characters
<Serial Number>	ASCII, 6 Characters
<number of variables>	ASCII, 1 Character

Table S-12. Status Information

Status Information (z..z)	Length (2 Characters Total)
<byte 1>	ASCII, 1 Character ("variable status")
<byte 2>	ASCII, 1 Character ("module status")

<byte n> = b8 b7 b6 b5 b4 b3 b2 b1

If bit bn is set in byte 1, then an error occurred in variable n. A variable error occurs when the input to an Analog Input variable is outside of the configured range.

If a bit is set in byte 2, then an error occurred in one of the subsystems (b5-b8 are not used).

Table S-13. Subsystem Errors

Byte	Subsystem
b1	EEPROM
b2	Flash
b3	ADC
b4	Configuration

Table S-14. Variable Configuration

Variable Configuration (b..b)	Length (29 bytes total)
<variable type>	ASCII, 1 Character
<variable name>	ASCII, 20 Characters
<binary format>	ASCII, 1 Character
<field length>	ASCII, 1 Character
<decimals>	ASCII, 1 Character
<unit>	ASCII, 4 Characters
<reset>	ASCII, 1 Character

Table S-15. Coding Variable Types, Binary Format, and Reset

Code	Variable Type	Binary Format	Reset
ASCII "0"	Empty Variable (EM)	No Format	Reset Disabled
ASCII "1"	Analog Input Variable	Boolean	Reset Enabled
ASCII "2"	Arithmetic Variable (AR)	Integer	n/a
ASCII "3"	Digital Output Variable (DO)	Real	n/a
ASCII "4"	Digital Input Variable (DI)	n/a	n/a
ASCII "5"	Setpoint Variable (SP)	n/a	n/a
ASCII "6"	Alarm Variable (AL)	n/a	n/a

MDS Supplement

I/O EQUATION STATION **PID Viewer Program**

The PID Viewer program, MDSPID.EXE, allows you to view the PID parameters in bar graph and trend formats using Modbus-RTU protocol. MDSPID also acts as a DDE source to allow access to PID data from other Windows programs.

Requirements

MDSPID is a Windows 3.1 program. MDSPID has the same hardware requirements as MDSCNFG. (See "Installation—Hardware/Software Requirements" on page 10 of the Manual.)

Although the ***I/O EQUATION STATION*** can be programmed to communicate via Profibus, ASCII or Modbus-RTU, it must be programmed for Modbus-RTU to run with MDSPID.

Operation

To try out MDSPID, install it from the Setup disk onto the same computer you use to run MDSCNFG.

Note:

If the MDSPID Setup program does not install successfully, exit all programs besides the Windows Program Manager, and run Setup again.

If all of the hardware (i.e. the comm port, cable, and RS232-485 converter) is set up and working correctly and MDSCNFG communicates successfully with the ***I/O EQUATION STATION***, then MDSPID should also communicate successfully.

MDS Communication Settings

Before starting the PID Viewer program, start MDSCNFG and select "Communication" from the main menu bar, then "Module Parameters". The "Module Selection" window will display serial communication information about the module on your comm link (If not, try "Search All").

The protocol must be "Modbus-RTU". Note the Baud Rate, Parity, Address and Comm Port because they will be set in the PID Viewer program.

Starting MDSPID

Start the PID Viewer program by clicking on the MDSPID icon inside the MDSPID program group. The ***I/O EQUATION STATION*** PID Viewer faceplate will appear.

Select "Communications" from the main menu bar, then "Comm Settings", and fill in the information to match the comm settings of the MDS module. ("Data Bits" should be set to 8.) Click "OK".

Again, select "Communications" from the main menu bar, then "Device Address", and move the slide bar until the correct address appears. Click "OK".

Click the "Start" button. The green "Communicating" indicator appears, and the bar graphs and trend display become active.

Note:

The configuration downloaded to the MDS module must contain at least one PID Controller (CO) variable, in order for MDSPID to run.

Trend Options

Select "Trend" from the main menu bar, then "Process Range". Here you can rescale the trend plots and bar graphs for the Process Variable and Setpoint to fit the engineering units of your application. This does not affect the Controller Output displays, since the Controller Output range is always 0-100%.

To modify the time width encompassed by the trend display, select "Trend" from the main menu bar and then select "Time Scale".

Controller Parameters

Under “Mode”, the radio buttons labeled “Manual” and “Auto” switch the controller between Manual and Auto modes.

Click on each of the two “Change” buttons in turn. One allows you to change the controller constants K , T_i and T_d . The other allows you to change Setpoint and Controller Output. Note that Controller Output can only be changed if the “Mode” selection is set to “Manual”.

Note:

From MDSPID, only Controller parameters that are “Host sourced” within the MDS configuration can be changed.

A single MDS can have up to two Controller (CO) variables configured to allow for cascade control. Under “CO Variable”, you can switch between “CO1” and “CO2”. When set to “CO1”, the faceplate displays the parameters of the first CO variable in the configuration (i.e., the CO variable with the lowest variable number). When set to “CO2”, the faceplate displays the parameters of the second CO variable.

Dynamic Data Exchange (DDE)

The Dynamic Data Exchange (DDE) allows another Windows program, running on the same PC and at the same time as MDSPID, to access the PID parameter data on a real time basis. MDSPID acts as a DDE source, so the other program must be a DDE destination. Data can be written to MDSPID by the destination program using a DDE “poke”.

The destination program must address registers in MDSPID using DDE’s “application|topic|item” format. The addresses of accessible data within MDSPID are listed in Table S-16.

A text value of “A” in the “Mode” register means controller mode is Auto, while “M” means Manual.

Table S-16. Data Addresses within MDSPID

Data	Address
Proportional Gain, K	MDSPID!MDSPID!Gain
Integral Time, T_i	MDSPID!MDSPID!IntTime
Derivative Time, T_d	MDSPID!MDSPID!DerivTime
Setpoint	MDSPID!MDSPID!Setpoint
Process Variable	MDSPID!MDSPID!ProcVar
Controller Output	MDSPID!MDSPID!CtrlOut
Mode	MDSPID!MDSPID!Mode

Moore Industries

STAR CENTER

*Providing Instrumentation fast
from our Quick-Ship Facilities!*

- Signal Transmitters
- Temperature Transmitters
- I/P and P/I Converters
- Isolators and Converters
 - Alarm Trips
- Integrators and Totalizers
 - Power Transducers
- Instrument Power Supplies
- Racks, Rails and Enclosures



**IN
STOCK**



**READY
TO
SHIP**

*If you need it today — It's on its way!
Call us NOW*

United States

1-800-999-2900

United Kingdom

0800 525107

Australia

(02) 525-9177



**FAST
REPAIRS
TOO!**

STAR Center is Service, Technical Assistance and Repair.

RETURN PROCEDURES

To return equipment to Moore Industries for repair, follow these four steps:

1. Call Moore Industries and request a Returned Material Authorization (RMA) number.

Warranty Repair –

If you are unsure if your unit is still under warranty, we can use the unit's serial number to verify the warranty status for you over the phone. Be sure to include the RMA number on all documentation.

Non-Warranty Repair –

If your unit is out of warranty, be prepared to give us a Purchase Order number when you call. In most cases, we will be able to quote you the repair costs at that time. The repair price you are quoted will be a "Not To Exceed" price, which means that the actual repair costs may be less than the quote. Be sure to include the RMA number on all documentation.

2. Provide us with the following documentation:
 - a) A note listing the symptoms that indicate the unit needs repair
 - b) Complete shipping information for return of the equipment after repair
 - c) The name and phone number of the person to contact if questions arise at the factory
3. Use sufficient packing material and carefully pack the equipment in a sturdy shipping container.
4. Ship the equipment to the Moore Industries location nearest you.

The returned equipment will be inspected and tested at the factory. A Moore Industries representative will contact the person designated on your documentation if more information is needed. The repaired equipment, or its replacement, will be returned to you in accordance with the shipping instructions furnished in your documentation.

WARRANTY DISCLAIMER

THE COMPANY MAKES NO EXPRESS, IMPLIED OR STATUTORY WARRANTIES (INCLUDING ANY WARRANTY OF MERCHANTABILITY OR OF FITNESS FOR A PARTICULAR PURPOSE) WITH RESPECT TO ANY GOODS OR SERVICES SOLD BY THE COMPANY. THE COMPANY DISCLAIMS ALL WARRANTIES ARISING FROM ANY COURSE OF DEALING OR TRADE USAGE, AND ANY BUYER OF GOODS OR SERVICES FROM THE COMPANY ACKNOWLEDGES THAT THERE ARE NO WARRANTIES IMPLIED BY CUSTOM OR USAGE IN THE TRADE OF THE BUYER AND OF THE COMPANY, AND THAT ANY PRIOR DEALINGS OF THE BUYER WITH THE COMPANY DO NOT IMPLY THAT THE COMPANY WARRANTS THE GOODS OR SERVICES IN ANY WAY.

ANY BUYER OF GOODS OR SERVICES FROM THE COMPANY AGREES WITH THE COMPANY THAT THE SOLE AND EXCLUSIVE REMEDIES FOR BREACH OF ANY WARRANTY CONCERNING THE GOODS OR SERVICES SHALL BE FOR THE COMPANY, AT ITS OPTION, TO REPAIR OR REPLACE THE GOODS OR SERVICES OR REFUND THE PURCHASE PRICE. THE COMPANY SHALL IN NO EVENT BE LIABLE FOR ANY CONSEQUENTIAL OR INCIDENTAL DAMAGES EVEN IF THE COMPANY FAILS IN ANY ATTEMPT TO REMEDY DEFECTS IN THE GOODS OR SERVICES, BUT IN SUCH CASE THE BUYER SHALL BE ENTITLED TO NO MORE THAN A REFUND OF ALL MONIES PAID TO THE COMPANY BY THE BUYER FOR PURCHASE OF THE GOODS OR SERVICES.

ANY CAUSE OF ACTION FOR BREACH OF ANY WARRANTY BY THE COMPANY SHALL BE BARRED UNLESS THE COMPANY RECEIVES FROM THE BUYER A WRITTEN NOTICE OF THE ALLEGED DEFECT OR BREACH WITHIN TEN DAYS FROM THE EARLIEST DATE ON WHICH THE BUYER COULD REASONABLY HAVE DISCOVERED THE ALLEGED DEFECT OR BREACH, AND NO ACTION FOR THE BREACH OF ANY WARRANTY SHALL BE COMMENCED BY THE BUYER ANY LATER THAN TWELVE MONTHS FROM THE EARLIEST DATE ON WHICH THE BUYER COULD REASONABLY HAVE DISCOVERED THE ALLEGED DEFECT OR BREACH.

RETURN POLICY

For a period of thirty-six (36) months from the date of shipment, and under normal conditions of use and service, Moore Industries ("The Company") will at its option replace, repair or refund the purchase price for any of its manufactured products found, upon return to the Company (transportation charges prepaid and otherwise in accordance with the return procedures established by The Company), to be defective in material or workmanship. This policy extends to the original Buyer only and not to Buyer's customers or the users of Buyer's products, unless Buyer is an engineering contractor in which case the policy shall extend to Buyer's immediate customer only. This policy shall not apply if the product has been subject to alteration, misuse, accident, neglect or improper application, installation, or operation. THE COMPANY SHALL IN NO EVENT BE LIABLE FOR ANY INCIDENTAL OR CONSEQUENTIAL DAMAGES.



WORLDWIDE • www.miinet.com

United States • info@miinet.com
Tel: (818) 894-7111 • FAX: (818) 891-2816

Australia • sales@mooreind.com.au
Tel: (02) 8536-7200 • FAX: (02) 9525-7296

Belgium • info@mooreind.be
Tel: 03/448.10.18 • FAX: 03/440.17.97

The Netherlands • sales@mooreind.nl
Tel: (0)344-617971 • FAX: (0)344-615920

China • sales@mooreind.sh.cn
Tel: 86-21-62491499 • FAX: 86-21-62490635

United Kingdom • sales@mooreind.com
Tel: 01293 514488 • FAX: 01293 536852