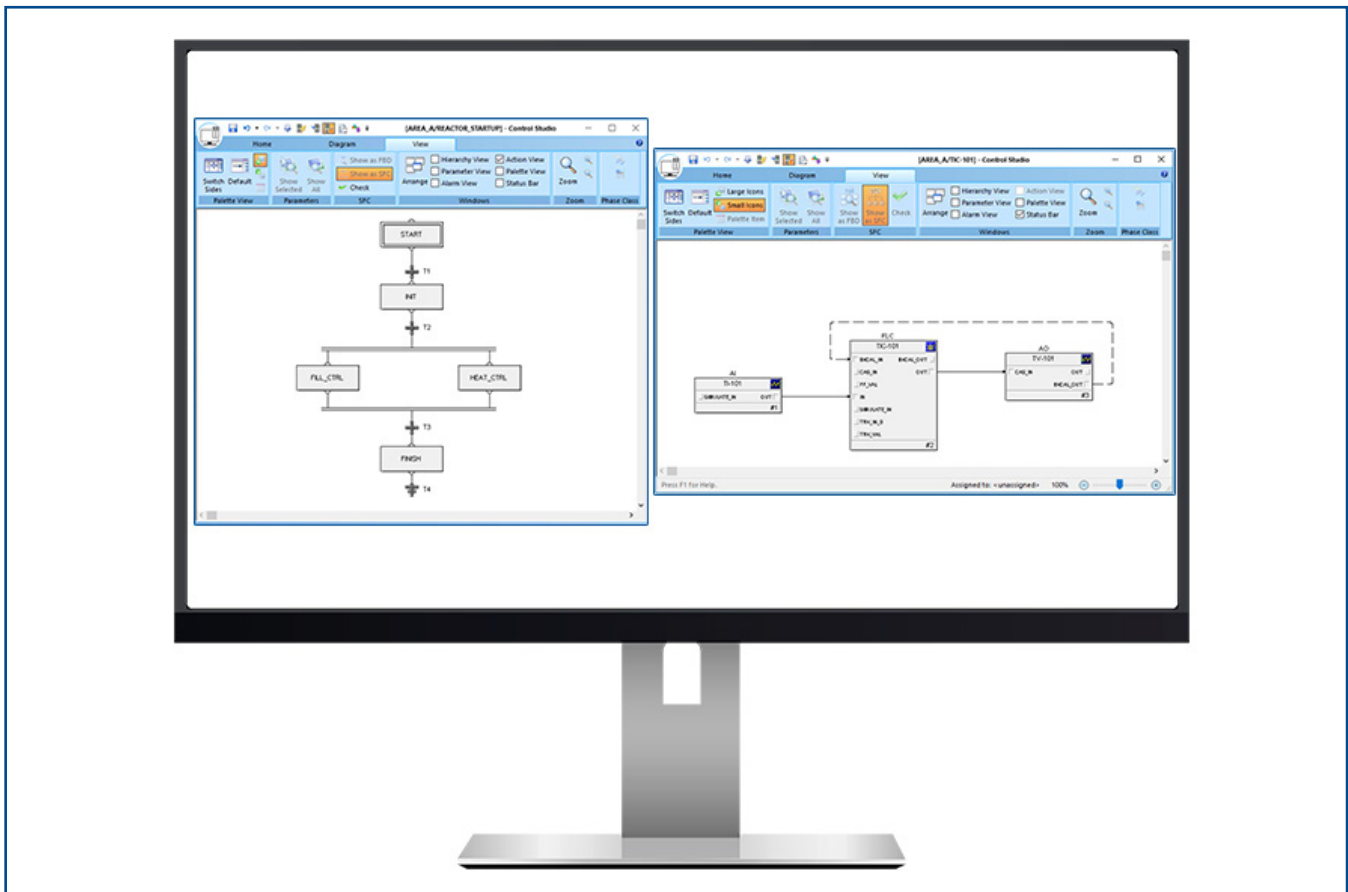


Monitor and Control Software



DeltaV™ allows you to use the right control language for the right control task.

- IEC61131-3 languages
- Scalable software meets any needs
- Integrates multi-vendor real-time data into your DeltaV™ system
- Perform calculations on data from external sources
- Embedded advanced control

Introduction

Are you looking for a control system that allows you to use standard control languages and functions? Do you want to use different control languages that are appropriate for the control task within the same strategy? Do you want a control system that permits you to design where and how you perform monitoring and control, rather than forcing you to fit within the constraints of a fixed architecture? A DeltaV system gives you the flexibility to choose from IEC61131-3 control languages to design your monitoring and control strategies.

Need to consolidate your PLC, legacy systems, and gauging systems into a single operator interface? *OPC* may be used to tear down the barriers between your legacy control systems and your DeltaV system. The DeltaV system provides you with a real-time database where you can store third-party data for use anywhere within the DeltaV system. You can display, trend, alarm and even use it in control strategies. What's more, you can choose the location that is best for integrating the data into the DeltaV system, the workstation or the controller.

This document details the monitoring and control functions supported by a DeltaV system. These functions can run on DeltaV Application Stations and controllers.

Benefits

Uses IEC61131-3 control languages. A DeltaV system supports three languages, so you can always use the tools most appropriate for the job. Function Block Diagrams (FBD), Sequential Function Charts (SFC), and Structured Text (ST) are all available, making control strategy development both intuitive and easy.

Scalable software meets any needs. As your needs grow, your DeltaV system is ready to expand with you. Additional capacity and even additional control functionality can be added online while you continue to control your process. Your entire configuration remains intact; you simply add to existing control strategies with the newly available functions or configure new control modules to take advantage of higher capacity.

Integrates multi-vendor data. Bring data into your DeltaV system for control, display, and historical purposes. When executing in the controller, the monitoring software uses Classic I/O, HART, FOUNDATION Fieldbus, PROFIBUS DP, AS-i bus, DeviceNet and serial interfaces to access data. Running in workstations, OPC classic and OPC UA provides Data Acquisition control software with fast connections and data transfers to and from any system. The Ethernet I/O Card (EIOC) also supports Ethernet based devices that communicate via the following protocols: OPC UA client, EtherNet/IP, Modbus TCP and IEC61850 MMS. In addition, the PK controller has embedded support for EtherNet/IP and Modbus TCP protocols.

Performs calculations on data from external sources. Now you can put the data from an existing system to work for you in the DeltaV system. All DeltaV applications can access and use the external data as if it were native DeltaV data. And because you can execute function blocks directly in the workstation, you can really put this external data to work for you.

Embedded advanced control. DeltaV has native advanced control functionality built-in. Use DeltaV Fuzzy to improve performance of non-linear or noisy loops. The optional DeltaV Insight Tuning capabilities makes adjusting the Fuzzy rules and PID parameters a snap. For more demanding process control requirements, you can use DeltaV's embedded Model Predictive Control (MPC) function blocks. For process values typically only available via lab samples and analysis, DeltaV Neural can be used to easily implement real-time virtual sensors for use in your control strategies. Refer to the DeltaV Fuzzy, DeltaV Predict and Predict Pro, and DeltaV Neural product data sheets for details.

Product Description

DeltaV Monitoring and Control Software uses standard IEC61131-3 control languages to support control strategy development. These strategies may be graphically assembled and modified using standard drag-and-drop techniques and can be mixed and matched as-needed based on specific process requirements.. Strategy development is visually intuitive, making it easy for first-time users to quickly become productive. Context sensitive, online help is available for all functions. The IEC61131-3 control languages include Function Block Diagramming (FBD), Sequential Function Charts (SFC), and Structured Text (ST). Typical uses of each language are listed below.

Control Language Application Example Usages
Function Block Diagram (FBD)
Monitoring and alarming
Continuous calculations
Analog control (pressure, temperature, flow)
Motor and block valve control
Totalizers
Sequential Function Chart (SFC)
Charging systems
Startup/shutdown control
Batch sequences (fill, mix, heat, dump)
Structured Text (ST)
Advanced math functions
Complex calculations
If-then-else decisions looping
Bit manipulations

Function Block Diagrams. Your DeltaV system uses Function Block Diagrams to implement continuously executing calculations, process monitoring, and control strategies. Graphical “wires” are used to connect the different blocks within a diagram. Each wire transfers one or more pieces of data. All communications throughout the DeltaV system are handled automatically.

Sequential Function Charts. This language enables you to configure operator-independent time-variant actions. SFCs are best at controlling strategies with multiple states. They can be used for sequencing and simple batch applications.









An SFC comprises a series of steps and transitions. Each step contains a set of actions that affect the process. Transitions determine when processing moves on to the next step(s). Both single-stream and parallel execution of logic is supported within SFCs.




Structured Text. With structured text you can write sophisticated calculations using a wide array of algebraic and trigonometric functions and operators. In addition, you can build complex logic expressions using conditional and iterative structures.





Data Acquisition. Display, trend, alarm, and use third party data as if it were native DeltaV data. Now you can put the data from an existing system to work for you in a DeltaV system. All DeltaV applications can access and use the external data as if it were native DeltaV data. The data can be used on any DeltaV workstation. It can also be integrated into control strategies, consolidated into reports, historized, and much more. Whether your application utilizes the OPC Classic or OPC UA capabilities of the workstation, or the controller’s I/O subsystem—once accessed, the real-time database makes the data immediately available throughout the DeltaV system. In runtime, the *exception reporting* mechanism moves the data to the appropriate system application. The *global configuration database* makes management and use of this data easy.







Section 1: Function Blocks








The available DeltaV DCS function blocks and their descriptions are listed in the below tables. Not all listed function blocks can execute in the Ethernet I/O Card (EIOC). For the list of supported EIOC resident function blocks and quantities please refer to the EIOC product data sheet. For a list of function blocks available for use in DeltaV SIS™ Logic Solvers refer to the DeltaV SIS Engineering Tools product data sheet.

Math Blocks	
	Absolute Provides the absolute value of an integer or floating point input
	Add Sums the values of two to sixteen inputs and generates an output value.
	Arithmetic Provides standard algorithms for : Flow Compensation Linear, Flow Compensation Square Root, Flow Compensation Approximate, Btu Flow, Traditional Multiply and Divide, Average, Summer, Fourth Order Polynomial, and Simple HTG Compensate Level.
	Comparator Compares two values for the conditions 'greater than', 'less than', 'equal', 'not equal', and generates an output for each comparison (GT, LT, EQ, NEQ). It also checks whether the primary input is within the range defined by two inputs.
	Divide Divides one input value by another input value and generates an output value.
	Integrator Integrates one or two variables over time. The block compares the integrated or accumulated value to pre-trip and trip limits and generates discrete output signals when the limits are reached.
	Multiply Multiplies two to sixteen inputs and generates an output value.
	Subtract Subtracts one input value from another input value and generates an output value.















Input/Output Blocks	
	Alarm Detection Creates a set of standard alarms (HI_HI, HI, LO, LO_LO) on any system parameter. AI and PID blocks have this function built-in.
	Analog Input Accesses a single analog measurement value and status from an I/O channel. The input value can be a transmitter's 4-20mA signal, the digitally communicated FOUNDATION Fieldbus block, or the digitally communicated primary or non-primary variable from a HART transmitter.
	Analog Output Assigns an analog output value to a field device through a specified I/O channel or Fieldbus device.

	<p>Discrete Input Accesses a single discrete measurement value and status from a two-state field device and makes the processed physical input available to other function blocks.</p>
	<p>Discrete Output Takes a binary setpoint and writes it to a specified I/O channel to produce an output signal.</p>
	<p>Fieldbus Multiplexed Discrete Input Accesses discrete input values from two-state field devices and makes the processed physical inputs available through eight discrete output parameters.</p>
	<p>Fieldbus Multiplexed Analog Input Used to make up to 8 input parameters available to the Fieldbus I/O subsystem.</p>
	<p>H1 Carrier Multiple Discrete Input Accesses a DI card on the H1 carrier device and makes the individual channels available as an 8-bit integer output.</p>
	<p>H1 Carrier Multiple Discrete Output Takes an 8-bit integer and writes it to the individual channels of a DO card on the H1 carrier device.</p>
	<p>Fieldbus Multiplexed Analog Input Used to connect to higher density Fieldbus transmitters.</p>
	<p>Fieldbus Multiplexed Analog Output Used to connect up to 8 analog output signals for Fieldbus devices supporting this function block.</p>
	<p>Pulse Input Converts the rate parameter on the Pulse Input card to a value in engineering units.</p>
	<p>Smart HART Discrete Input Accesses a single discrete input variable with status from a Wireless Discrete field device and makes the processed physical input available to other function blocks in either a binary format or a named set output.</p>
	<p>Smart HART Discrete Output Converts a setpoint to a corresponding numeric value and writes it to a specified channel.</p>
	<p>TAGIO Converts to the appropriate TAGIO function block below, based on the type of the assigned control tag.</p> <p>Related Blocks</p> <ul style="list-style-type: none">  TAGAI Reads and displays the values of a TagAI_Monitor type control tag in a PLC.  TAGAO Reads and displays the values of a TagAO_Monitor type control tag in a PLC.  TAGDI Reads and displays the values of a TagDI_Monitor type control tag in a PLC.  TAGDO Reads and displays the values of a TagDO_Monitor type control tag in a PLC. <p><i>Note: TAGIO function blocks are only supported within the Ethernet I/O Card (EIOC).</i></p>











Timer and Counter Blocks	
	<p>Counter Generates a discrete output value of True (1) when the count reaches a specified trip value. The block functions as an increment or decrement counter.</p>
	<p>Off-delay Timer Delays the transfer of a False (0) discrete input value to the output by a specified time period.</p>
	<p>On-delay Timer Delays the transfer of a True (1) discrete input value to the output by a specified time period.</p>
	<p>Retentive Timer Generates a True (1) discrete output after the input has been True for a specified time period. When the reset input is set True, the elapsed time and the output value are reset.</p>
	<p>Timed Pulse Generates a True (1) discrete output for specified time duration when the input makes a positive (False-to-True) transition. The output remains True even when the input returns to its initial discrete value; it returns to its original False value only when the output is True longer than the specified time duration.</p>
	<p>Date Time Event Generates a pulse at a preset date and time or at preset time intervals.</p>






Logic Blocks	
	<p>Action Evaluates a single structured text expression when the input value is True. Mathematical functions, logical operators, constants, parameter references, and I/O reference values may be used in the expression. Expressions are written using IEC 61131-3 Structured Text. See section 3 for Structured Text details.</p>
	<p>AND Generates a discrete output value based on the logical AND of two to sixteen discrete inputs.</p>
	<p>Bi-directional Edge Trigger Generates a True (1) discrete pulse output when the discrete input makes a positive (0-to-1, False-to-True) or a negative (1-to-0, True-to-False) transition since the last execution of the block. If there has been no transition, the discrete output of the block is False.</p>
	<p>Boolean Fan Input Generates a discrete output based on the weighted binary sum, binary coded decimal (BCD) representation, transition state, or logical OR of 1 to 16 discrete inputs.</p>
	<p>Boolean Fan Output Decodes a binary weighted floating point input to individual bits and generates a discrete output value for each bit (as many as sixteen outputs).</p>
	<p>Condition Evaluates a single structured text expression and generates a discrete output value when the expression is evaluated True (1) for longer than a specified time period.</p>
	<p>Device Control Provides setpoint control for multi-state discrete devices, such as motors, pumps, and block valves.</p>










	<p>Discrete Control Condition Evaluates up to 16 interlock conditions, 8 permissive conditions and 8 force setpoint conditions. Generated outputs can be used by the Enhanced Device Control function block.</p>
	<p>Enhanced Device Control Provides setpoint control for multistate discrete devices such as motors, pumps, and block valves. Accepts outputs from the Discrete Control Condition function block for interlocking, force setpoints, and permissives. Up to 4 discrete inputs and 4 discrete outputs can be associated with up to 6 states.</p>
	<p>Multiplexer Selects one input value from up to 16 input values and places it at the output.</p>
	<p>Negative Edge Trigger Generates a True (1) discrete pulse output when the discrete input makes a negative (1-to-0, True-to-False) transition since the last execution of the block. If there has been no transition (or a positive transition), the discrete output of the block is False (0).</p>
	<p>NOT Logically inverts a discrete input signal and generates a discrete output value. When the input is True (1), the output is False (0). When the input is False, the output is True.</p>
	<p>OR Generates a discrete output value based on the logical OR of 2 to 16 discrete inputs. When one or more of the inputs is True (1), the output is set True (1).</p>
	<p>Positive Edge Trigger Generates a True (1) discrete pulse output when the discrete input makes a positive (0-to-1, False-to-True) transition since the last execution of the block. If there has been no transition (or a negative transition), the discrete output of the block is False (0).</p>
	<p>Reset/Set Flip-flop Generates a discrete output value based on NOR logic of Reset and Set inputs. If the Reset input is False (0) and the Set input is True (1), the output is True. The output remains True regardless of the Set value until the Reset value is True. When Reset becomes True, the output is False.</p>
	<p>Set/Reset Flip-flop Generates a discrete output value based on NAND logic of Set and Reset inputs. When the Reset input is False (0) and the Set input is True (1), the output is True. The output remains True until the Reset input is True and the Set input is False. When the Reset input is True, the output is equal to the Set input.</p>
	<p>Transfer Selects one of two analog input signals and transfers the selected input to the output after a specified time. The transfer from one input to another is smoothed with a linear ramp.</p>








Analog Control Blocks	
	<p>Analog Tracking Evaluates up to 16 tracking conditions for use by downstream function blocks such as the PID. Each condition supports independent delay on time, delay off time, and enable. One or more conditions can also be optionally latched until a reset is triggered.</p>
	<p>Bias/Gain Provides adjustable gain capability by computing an output value from a bias setpoint, an input, and a gain value.</p>
	<p>Calc Allows you to specify an expression that determines the block's output. Mathematical functions, logical operators, constants, parameter references, and I/O reference values may be used in the expression. Expressions are written using IEC 61131-3 Structured Text. See section 3 for Structured Text details.</p>
	<p>Control Selector Selects one of three control signals to perform override control to a PID function block.</p>
	<p>Dead time Introduces a pure time delay in the value and status used in a signal path between two function blocks.</p>
	<p>Enhanced Control Selector Selects one of sixteen control signals to perform override control. Selection can be determined by operator selection or automatically based on high, low and middle select.</p>
	<p>Enhanced Ramp Ramps a target function block parameter to a fixed or variable endpoint over a specified time or at a specified rate. Supports ramping of target function blocks in the MAN, AUTO, RCAS, ROU, and LO modes. Ramps can be started, paused, and stopped via operator entry or external logic.</p>
	<p>Fieldbus Input Selector Extended Selects the first good, minimum, maximum, middle, or average of up to eight input signals. Supports operator-selected, mode, and signal disable.</p>
	<p>Filter Applies an equation to filter changes in the input signal and generates a smooth output signal.</p>
	<p>Input Selector Selects the first good, minimum, maximum, middle, or average of up to four input signals. Supports operator-selected, mode, and signal disable.</p>
	<p>Lead/Lag Provides dynamic compensation for an input value. The block can apply a lead time function, a lag time function, or a combination of the two. A specified gain is applied to the compensated value and the value is high/low-limited based on the block mode.</p>
	<p>Limit Limits an input value between two reference values.</p>
	<p>Manual Loader Allows the block output to be set by an operator.</p>
	<p>PID Combines all the necessary logic to perform analog input channel processing, proportional-integral-derivative (PID) control, and analog output channel processing within one function block.</p>

	<p>Ramp Creates a ramping output signal to increase or decrease a variable toward a specified target value at a defined rate.</p>
	<p>Rate Limit Limits the rate of change of the output value to specified limits. Also provides the derivative of the input signal.</p>
	<p>Ratio Applies an adjustable ratio setpoint to achieve a desired input/output relationship.</p>
	<p>Scaler Provides scaling and dimensional consistency between two values of different engineering units. The block converts the input value to the specified scale and generates an output value.</p>
	<p>Signal Characterizer Characterizes or approximates any function that defines an input/output relationship. The function is defined by configuring as many as twenty X, Y coordinates. The block interpolates an output value for a given input value using the curve defined by the configured coordinates. Two separate analog input signals can be processed simultaneously to give two corresponding separate output values using the same defined curve.</p>
	<p>Signal Generator Produces an output signal used to simulate a process signal.</p>
	<p>Signal Selector Selects the maximum, minimum, or average of as many as sixteen input values and places it at the output.</p>
	<p>Splitter Takes a single input and calculates two outputs based on specified coordinate values.</p>









Advanced Control Blocks	
	<p>Fuzzy Logic Easy to use alternative to PID control. Useful in non-linear and noisy processes. Also very suited to handling unmeasured disturbances.</p>
	<p>Inspect Monitors plant area control performance statistics for viewing, trending and history collection. Also provides for programmatic enable/disable of performance monitoring based on equipment or unit state.</p>
	<p>Model Predictive Control (MPC, MPCPro and MPCPlus) Model Predictive constrained multi-variable controller for processes with interacting variables, difficult dynamics, multiple constraints, and throughput optimization.</p> <p>Related Blocks</p> <ul style="list-style-type: none"> <li style="margin-bottom: 10px;">  <p>MPC Process Simulator Allows the process to be simulated to verify the model.</p> <li style="margin-bottom: 10px;">  <p>MPC Input Reference Easily connect input parameters (controlled variables, disturbance variables, and constraint variables) from multiple loops in a single MPC strategy. Useful for implementing MPC in a supervisory fashion.</p> <li style="margin-bottom: 10px;">  <p>MPC Output Reference Easily connect output parameters (manipulated variables) from multiple loops in a single MPC strategy. Useful for implementing MPC in a supervisory fashion.</p>  <p>MPC Analyzer template Used to pre-process sampled analyzer measurements before feeding them into the MPCPro block.</p>
	<p>Neural Network Build virtual sensors for on-line qualitative property estimation. Use anywhere samples are taken for analysis, or in place of or back-up to continuous analyzers.</p> <p>Related Blocks</p> <ul style="list-style-type: none">  <p>Lab Entry Allows operator input of offline laboratory analysis results to be used in training the neural network. Sample time and test results are easily logged right from the operator interface.</p>
	<p>Diagnostic The Diagnostic function block provides a method to monitor device alerts from non-fieldbus assets.</p>

Advanced Functions Blocks	
	<p>Analog Voter Monitors a number of analog input values and determines if there are enough votes to trip. If a configured number of the inputs vote to trip, the block trips, and the output of the block is set to zero (0).</p>
	<p>Discrete Voter Monitors a number of discrete input values and determines if there are enough votes to trip. If a configured number of the inputs vote to trip, the block trips, and the output of the block is set to zero (0).</p>
	<p>Cause and Effect Matrix Incorporates interlock and permissive logic for one or more final control elements. Up to 16 Causes (inputs) and up to 16 Effects (outputs) can be configured using the function block's matrix.</p>
	<p>State Transition Implements a user-defined state machine. The state machine describes the possible states, and the transitions between those states, that can occur.</p>
	<p>Step Sequencer Associates system states with actions to drive outputs based on the current state.</p>

Energy and Metering Blocks	
	<p>Isentropic Expansion Calculates the final enthalpy for isentropic expansion of steam to a given pressure for given entropy. If the entropy-pressure point lies in the two-phase region, the combined enthalpy of steam and water for constant entropy is provided. Steam quality (% moisture) and steam temperature are also calculated.</p>
	<p>Saturated Steam Properties Calculates steam enthalpy, entropy, specific volume and pressure for saturation conditions specified by a given temperature.</p>
	<p>Saturated Temperature Calculates steam temperature at saturation give steam pressure.</p>
	<p>Steam Density Ratio Calculates the square root of the ratio of steam density to the density of steam at the calibration conditions of the flow meter.</p>
	<p>Steam Properties Calculates steam enthalpy, entropy and specific volume for a give gauge pressure and a given temperature.</p>
	<p>Water Enthalpy Calculates enthalpy of water for a specified temperature.</p>
	<p>Water Entropy Calculates entropy of water for a specified temperature.</p>
	<p>Flow Metering in SI units Calculates mass flow, volumetric flow and energy flow for orifice and turbine meters.</p>
	<p>Flow Metering in US units Calculates mass flow, volumetric flow and energy flow for orifice and turbine meters.</p>

Special Items	
	<p>Custom Block Tool Adds various block types .</p> <p> Composite Block A grouping of individual function blocks. Composites can be of the Linked or Embedded type.</p> <p> Module Block A block on the diagram that represents another module within the system. It allows direct connection of parameters between this module and the referenced module.</p>
	<p>Input Parameter A value from another function block, a sequential function chart (SFC), or a field device supplied to a module for use in execution.</p>
	<p>Internal Read Parameter A value used within a module that supplies an output for connection to I/O or other module parameters.</p>
	<p>Internal Write Parameter A value used within a module that supplies an input for connection to I/O or other module parameters.</p>
	<p>Output Parameter A value calculated or triggered by the module and sent to other diagrams or field devices.</p>

Section 2: Sequence Function Chart Tools

SFC Tools	
	<p>Input Parameter A value from another function block, a sequential function chart (SFC), or a field device supplied to a module for use in execution. Automatically available when the Module Block is used on another module’s diagram.</p>
	<p>Internal Read Parameter A value used within a module that supplies an output for connection to I/O or other module parameters.</p>
	<p>Internal Write Parameter A value used within a module that supplies an input for connection to I/O or other module parameters.</p>
	<p>Output Parameter A value calculated or triggered by the module and sent to other function blocks, SFC’s, or field devices. Automatically available when the Module Block is used on another module’s diagram.</p>
	<p>Step An element of an SFC that contains a set of actions. The actions define the rules and the behavior of the step. The step manages the execution of the actions they contain.</p>
	<p>Termination A transition that specifies a condition that represents a logical end to a sequence.</p>
	<p>Transition An element of a SFC that allows the sequence to proceed from one step to the next if a transition condition is met. A transition condition is the result of evaluating a Boolean expression.</p>
	<p>Sequence A number of steps and transitions linked together.</p>

Section 3: Structured Text Instruction Functions

Math	Description
ABS (x)	Absolute value of x
EXP	Exponential of x
EXPT (x, y)	Raise x to the power of y
FRACT (x)	Fractional part of a number
LN (x)	Natural logarithm of x
LOG (x)	Log base 10 of x
LOG2 (x, y)	Log base y of x
MAX (x, y)	Maximum of x and y
MIN (x, y)	Minimum of x and y
ROUND (x)	Round x to the nearest float
SQRT (x)	Square root of x
TRUNC (x)	Floor function
Trig	Description
ACOS (x)	Arc cosine of x
ASIN	Arc sine of x
ATAN	Arc tangent of x
COS	Cosine of x
SIN(x)	Sine of x
TAN (x)	Tangent of x
Bitwise	Description
ROL(x,y) / ROT	Rotate x to the left by y positions
ROTL16(i,n)	Rotate i to the left by n bits for a 16 bit integer
ROR(x,y) / ROT	Rotate x to the right by y positions
ROTR16(i,n)	Rotate i to the right by n bits for a 16 bit integer
SHL(x,y)	Shift x left by y positions, no carry
SHR(x,y)	Shift x right by y positions, no carry
ASR16(i,n)	Arithmetic shift right n bits by l positions for a 16 bit integer
STBT(i,b,n)	Set bit n in integer i to boolean value b
Time	Description
time_to_str	converts a time value to a time string
str_to_time	converts a time string to a time value

The DeltaV system uses IEC 61131-3 Structured Text to create custom functions. The language uses operators, functions, statements and operands in expressions. SFCs and the Action, Condition, Calc, Analog Tracking and Discrete Control Condition function blocks use expressions. Any system parameter may be referenced by these expressions.

Operator	Description
-	Unary minus
+	Unary plus
!, NOT	Logical NOT
~	Bitwise NOT
OR	Logical OR
XOR	Logical Exclusive OR
AND	Logical AND
	Bitwise OR
^	Bitwise XOR
&	Bitwise AND
=	Equality
!=, <, >, ~=	Inequality
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
+	Add
+	String concatenation
-	Subtract
*	Multiply
/	Divide
MOD, %	Modulus (remainder)
**	Raise to power
?:	Conditional evaluation

Miscellaneous	Description
EUP(e,p0,p100)	Conversion from engineering units to percent
PEU(p,p0,p100)	Conversion from percent to engineering units
Option Explicit	Forces all local variables to require explicit declaration
Logevent(x)	Logs the event string specified
Selstr(n,s1,s2,s3,s4,s5)	Returns the nth string
Sysstat(x)	Download, Power fail, and switchover event flags
Basepath(path_string)	Returns the base portion of a path string

Statements

Statement	Description
;	End of statement separator
:=	Assignment
If...Then... Else...End_if	Conditional execution
While...Do... End_While	Looping function
Exit	Exits the innermost While loop
Var...End_var	Declare variables
Option Explicit	Force variables to be declared
REM	Single line comment
(* ... *)	Comment (multi-line)

Operands

- Named set constants in the form “SetName:ValueName.”
- Inputs are floating-point-with-status data types that may be wired to function block parameter connectors or used in expressions.
- Outputs are floating-point-with-status data types that may be wired from function block parameter connectors or written from action expressions.
- Input device signal tags (DSTs) are references to signals and also contain signal status data.
- Constants- predefined, unchangeable values in expressions. These constants let you test values against DeltaV system values without having to know their internal representations. The DeltaV system supports the constants listed below:

Constant	Description
GOOD	Good status
BAD	Bad status
TRUE	Boolean True
FALSE	Boolean False
UNC	Uncertain status
LIMITED CONSTANT	Limited constant status
LIMITED HIGH	Limited High status
LIMITED LOW	Limited Low status
AUTO	Automatic mode
CAS	Cascade mode
IMAN	Initializing Manual mode
LO	Local Override mode
MAN	Manual mode
OS	Out of Service mode
RCAS	Remote Cascade mode
ROUT	Remote Out mode

Capacity

Your DeltaV system provides the exact monitoring and control functions your application requires. The software is also scalable in size. The system is sized according to the number of device signal tags (DSTs) and SCADA tags. A DST is a signal used as a measured variable or controlled output. A SCADA tag is a value brought into the DeltaV system for monitoring, display, alarming and history collection.

System Specifications	Description
Hardware	MQ, MX, SQ, SX, SZ, and PK Controllers Ethernet I/O Card (EIOC)
Capacity SCADA tags Device signal tags	25000 per workstation, 30000 per system
Data source	Classic I/O (AI, AO, DI, DO) HART (AI, AO) EtherNet/IP Modbus TCP FOUNDATION Fieldbus PROFIBUS DP AS-i bus DeviceNet Serial OPC Classic OPC UA

Classic and CHARMs I/O supports only one signal per device. HART, FOUNDATION Fieldbus, PROFIBUS DP, ASi bus and serial communications can support multiple signals per device, but in most cases only 1 DST is counted for system sizing purposes. OPC communications can support multiple tags per server.

Ordering Information

Controller function licensing is based on the number and type of I/O connected. The license limits are enforced on a systemwide basis and not on a per controller basis. For details, refer to the Controller Licensing white paper.

License Type	Model Number	License Size (DSTs)
DI	VE3111Sxxxxx	xxxxx = 00025 to 30000
DO	VE3112Sxxxxx	xxxxx = 00025 to 30000
AI	VE3113Sxxxxx	xxxxx = 00025 to 30000
AO	VE3114Sxxxxx	xxxxx = 00025 to 30000
ProfessionalPLUS Station	VE2101Sxxxxx-1	xxxxx = 00025 to 30000
ProfessionalPLUS Premium Station	VE2101PSxxxxx	xxxxx = 00025 to 30000
Standard-to-Premium ProfessionalPLUS Station upgrade	VE21USPSxxxxx	xxxxx = 00025 to 30000

Related Products

- DeltaV Fuzzy.** For difficult-to-control loops, fuzzy logic technology enables more robust control than standard PID algorithms. It improves loop performance by providing faster, tighter control with minimal overshoot and settling time. Fuzzy logic is ideal for controlling non-linear process loops, as well as improving control of simple loops.
- DeltaV Neural.** Easily create virtual sensors using neural networks.
- DeltaV InSight.** Control performance monitoring and loop tuning application embedded in DeltaV. Identifies control problems and improves control performance with automatic process learning, loop diagnostics, on-demand and adaptive tuning, and automatic report generation.
- DeltaV Predict and DeltaV PredictPro.** Obtain greater throughput, reduced variability, and increased profitability with easy to design, implement, operate, and maintain Model Predictive Control (MPC)
- Ethernet I/O Card.** Provides a platform to access data from intelligent field devices in the DeltaV system utilizing Modbus TCP, EtherNet/IP, IEC61850 MMS and OPC UA.
- OPC UA.** The different OPC UA servers and clients in the DeltaV system allow data reads and writes to and from 3rd party application in an easy, reliable and secure way

©2022, Emerson. All rights reserved.

The Emerson logo is a trademark and service mark of Emerson Electric Co. The DeltaV logo is a mark of one of the Emerson family of companies. All other marks are the property of their respective owners.

The contents of this publication are presented for informational purposes only, and while diligent efforts were made to ensure their accuracy, they are not to be construed as warranties or guarantees, express or implied, regarding the products or services described herein or their use or applicability. All sales are governed by our terms and conditions, which are available on request. We reserve the right to modify or improve the designs or specifications of our products at any time without notice.

Contact Us

www.emerson.com/contactus