Honeywell

ControlEdge PLC

ControlEdge RTU

Release 174.1

ControlEdge Builder Function and Function Block Reference

RTDOC-X286-en-174A
December 2022

DISCLAIMER

CONTENTS

CHAPTER

1 ABOUT THIS GUIDE

Revision history

Revision	Date	Description
A	December 2022	Initial release of this document

Intended audience

This documentation is intended for the following audience: Users who plan, install, configure, operate, or maintain ControlEdge™ 900 and 2020 controllers running the eCLR (IEC 61131-3) execution environment.

Prerequisite skills

Knowledge of SCADA systems and experience of working in a Microsoft Windows environment are required.

Introduction to ControlEdge Technology

Item	Description
ControlEdge PLC	ControlEdge 900 controllers running the eCLR (IEC 61131-3) execution environment with PLC software options configured with ControlEdge Builder.
ControlEdge RTU	ControlEdge 2020 controllers running the eCLR (IEC 61131-3) execution environment with RTU software options configured with ControlEdge Builder.
ControlEdge UOC	ControlEdge 900 controllers running the Honeywell control execution environment (CEE) configured with Experion Control Builder.

Special terms

The following table describes some commonly used industry-wide and Honeywell-specific terminology:

Terminology	Description
AI	Analog Input
AO	Analog Output
ControlEdge Builder	A integrated configuration tool to design, configure, program and maintain ControlEdge controllers.
DI	Digital Input
DO	Digital Output
EFM	Electronic Flow Measurement
Experion® PKS	Experion® Process Knowledge System
HART-IP	HART-IP extends the HART protocol to Ethernet connected nodes. This facilitates host level systems and asset management applications to access and integrate measurement and device diagnostics information from HART-enabled field devices using the existing plant networking infrastructure.
Modbus	A communication protocol supports communication between Modbus responder devices and Modbus master devices via serial port or Ethernet port.
OPC UA	An industrial machine-to-machine (M2M) communication protocol is developed by the OPC Foundation, which provides a path forward from the original OPC communications model (namely the Microsoft Windows only process exchange COM/DCOM) to a cross-platform service-oriented architecture (SOA) for process control, while enhancing security and providing an information model.
PI	Pulse Input
SCADA	Supervisory Control and Data Acquisition

Related documents

The following list identifies publications that may contain information relevant to the information in this document.

- ControlEdge Builder Software Installation User’s Guide
- ControlEdge Builder Software Change Notice
- ControlEdge PLC and ControlEdge RTU Getting started

- ControlEdge Builder User’s Guide
- ControlEdge 900 Platform Hardware Planning and Installation Guide
- ControlEdge 2020 Platform Hardware Planning and Installation Guide
- ControlEdge Builder Protocol Configuration Reference Guide
- ControlEdge PLC and ControlEdge RTU Network and Security Planning Guide
- ControlEdge EtherNet/IP User's Guide
- ControlEdge RTU and PLC DNP3 Device Profile
- ControlEdge Bulk Configuration User’s Guide
- ControlEdge PLC PROFINET User’s Guide
- ControlEdge RTU Electronic Flow Measurement User's Guide
- Firmware Manager User Guide

CHAPTER

2 OVERVIEW OF HONEYWELL PROVIDED LIBRARIES

The following IEC61131–3 libraries are provided with ControlEdge Builder. For more information about how to use libraries in ControlEdge Builder projects, see the embedded online help.

Licensed By	Library Name	Library Type	Short Description
Base	HWFBLib	User	HWFBLib provides library of common regulatory and device control function blocks based on Honeywell's mature control products.
	Funclib	Firmware	Funclib provides some common utility functions for processing values such as max, min, linearization, etc.
	HART and HART_V2	Firmware	These function blocks access the HART field devices connected to HART-enabled AI/AO channels. HART command 3 and command 48 are supported.
	UnitConversionlib	Firmware	Converts temperature from Fahrenheit to Kelvin, temperature from Celsius to Kelvin, temperature from Fahrenheit to Rankine, etc.
	Utilitylib	Firmware	Utility sets the controller Real Time Clock by a provided Timestamp value and reads out the current time and date from the real-time clock and presents them as parameters.
	MODBUS	Firmware	Modbus is a serial communication protocol developed by Modicon published by Modicon® in 1979 for use with its programmable logic controllers (PLCs). In simple terms, it is a method used for transmitting information over serial lines between electronic devices.

Licensed By	Library Name	Library Type	Short Description
	OPC UA	Firmware	OPC UA is a machine to machine communication protocol for industrial automation developed by the OPC Foundation.
	OPCUAFBHelpers	User	Honeywell Provided OPCUA Function blocks.
	CRC_16	Firmware	Calculates CRC-16
	User Defined	Firmware	Receives user defined data from the target device and send user defined data to the target device.
	ETHERNETIP	Firmware	Reads a variable value from a peer to peer controller and writes a value to a peer to peer controller through the tag name.
	MDIS	Firmware	The MDIS library has a set of OPC UA function blocks representing all the MDIS OPC UA object types as defined in the MDIS OPC UA Companion Specification V1.2. The MDIS OPC UA Object function blocks are used to obtain data from MDIS OPC UA compliant Servers.
	ELEPIU	User	Connects to the ELEPIU MUX board and provides the temperatures in a data structure for SCADA or PCDI connections.
	Energy Control	Firmware	EnergyControl library provides a set of function blocks for controlling charge/discharge of an Energy Storage System considering various constraints.
Gas and Liquid Metering Calculation Library	AGALib and AGALib_V2 and AGALib_V3	Firmware	Calculates Gas Super compressibility, Density and Compressibility at standard and flowing conditions using heating value/ without heating value/ all 21 gas elements; calculates Gas energy using gas heating value/

Licensed By	Library Name	Library Type	Short Description
			all 21 gas elements; calculates corrected flow rates for Orifice meter; calculates corrected flow rates for Turbine meter; calculates corrected flow rates for Ultrasonic meter; calculates corrected flow rate for Coriolis meter.
	API 11.1 Lib	Firmware	API 11.1 calculates for Crude Oil, Lubricating Oil, Refined Products, Special Products with Alternate conditions and or Observed conditions in both US units and Metric Units.
	API 21.1 Lib and API 21.1 Lib_V2	Firmware	This library provides function blocks to support API21.1 for electronic gas measurement systems. These function blocks provide flow measurement, reporting and change management logs required for accurate and auditable gas measurement.
	API 21.2 Lib and API 21.2 Lib_V2	Firmware	This library provides function blocks to support API 21.2 measures the liquid flow for the configured meter type. The function block calculates meter density, Gross and Net standard volume, Sediments and Water volume, mass flow rate, averaging and Totalization based on the input parameters. It generates events, alarms, hourly and daily QTR's which can be read from SCADA using MODBUS or DNP3 protocol.
	apingl lib	Firmware	The basic function of API NGL block when set for line to base operation is to calculate standard density and associated volume correction factor from an observed density,

Licensed By	Library Name	Library Type	Short Description
			temperature and pressure with an option to either calculate a vapor pressure or use an operator entered value. The basic function of API NGL block when set for base to line operation is to calculate meter density and associated volume correction factor from an observed density, temperature and pressure with an option to either calculate a vapor pressure or use an operator entered value.
	ISO5167DualLib	Firmware	ISO 5167 is an international standard covering the measurement of fluid flow by means of pressure differential devices such as orifice plates and venturis. When some parameters are known, ISO 5167 allows other variables to be calculated. The most common usage is to calculate mass flow rate from differential pressure, static pressure and density. ISO 5167 is widely used in most areas of the world except North America. The basic function of the ISO 5167 block is to calculate mass flow rate from primary element DP and other required inputs.
	ISO5167DualJTLib	Firmware	ISO 5167JT is an international standard covering the measurement of fluid flow by means of pressure differential devices such as orifice plates and venturis. When some parameters are known, ISO 5167 allows other variables to be calculated. The most common usage is to calculate mass flow rate from differential pressure, static pressure and density. ISO 5167 is widely used in most areas of the world except

Licensed By	Library Name	Library Type	Short Description
			North America. The basic function of the ISO 5167 block is to calculate mass flow rate from primary element DP and other required inputs.
	ISO6976lib	Firmware	ISO 6976 block calculates for Calorific value on a molar, mass and volumetric basis; Calorific value on a superior and inferior basis; Calculation of values on an ideal and a real basis; Standard density and compressibility at the 15 deg C and 1.01325 bara conditions regardless of the chosen combustion/ metering.

AGA/API standard library version supported in ControlEdge RTU:

Library	Specification	
Metering Calculation Library	**Gas**	
	AGA 3 (1992)	Orifice Meter
	AGA 3 (2012)	Orifice Meter
	AGA 5 (2009)	Volume to Energy Calculation
	AGA 7 (1996)	Turbine Meter
	AGA 8 (1994)	Gas Compressibility
	AGA 8 (2017)	Gas Compressibility
	AGA 9 (1996)	Ultrasonic Meter
	AGA 11 (2013)	Coriolis Meter
	ISO 6976 (1995)	Natural gas: calorific value density, relative density and Wobbe Index
	Liquid	
	API 11.1 (2004)	Volume Correction Factor
	API 11.2.2/M (1986)	Compressibility Factors
	API 11.2.4 (2007)	Temperature Correction
	API 11.2.5 (2007)	Correction Factor for pressure
	Gas & Liquid	
	ISO 5167 (1991, 1997, 2003)	Pressure differential devices such as orifice plates and Venturis
	AGA 3 (2012)	Orifice Meter

CHAPTER

3 AGA

The following libraries of AGA Function Blocks are supported:

Library	Description
AGALib	Basic version of AGA function block library. It supports AGA 3 (1992), AGA5 (1996), AGA8 (1994), AGA 7(1996), AGA 11 (2013) and AGA 9 (1996) Do not support clarity between Super-compressibility factor at base condition and standard conditions.
AGALib_ V2	It is supported from R151 release. Added clarity between Super-compressibility factor at base condition and standard conditions.
AGALib_ V3	It is supported from R161.2 release. This upgrade supports AGA3(2012), AGA8(2017) including GERG method and AGA5 (2009) and AGA3 orifice method (2013) supporting Liquid measurement. Function block with structure input is not supported and will not be available in this library. **NOTE:** When the outcode for any of the function block in AGALib_V3 is error, other corresponding output parameters from the same function block are invalid. Hence caution must be taken in project engineering when processing of the output parameters in conjunction with the outcode.

The following AGA function blocks are available:

Function Block	Apply to	Description
AGA8_ GrossMethod1	AGALib, AGALib_V2 and AGALib_V3	They calculate: • Gas Compressibility at base, standard and flowing conditions (temp & pressure) • Density of gas at base, standard and flowing conditions (temp & pressure)

<table>
<tr><th>Function Block</th><th>Apply to</th><th>Description</th></tr>
<tr><td></td><td></td><td rowspan="2">Gas Super-compressibility at standard temp & pressure</td></tr>
<tr><td>AGA8_ GrossMethod1_st</td><td>AGALib and AGALib_ V2</td></tr>
<tr><td>AGA8_ GrossMethod2</td><td>AGALib, AGALib_V2 and AGALib_V3</td><td rowspan="2">They calculate:
Gas Compressibility at base, standard and flowing conditions (temp & pressure)
Density of gas at base, standard and flowing conditions (temp & pressure)
Gas Super-compressibility at standard temp & pressure</td></tr>
<tr><td>AGA8_ GrossMethod2_st</td><td>AGALib and AGALib_ V2</td></tr>
<tr><td>AGA8_DetailMethod</td><td>AGALib, AGALib_V2 and AGALib_V3</td><td rowspan="2">They calculate:
Gas Compressibility at base, standard and flowing conditions (temp & pressure)
Density of gas at base, standard and flowing conditions (temp & pressure)
Gas Super-compressibility at standard temp & pressure</td></tr>
<tr><td>AGA8_ DetailMethod_st</td><td>AGALib and AGALib_ V2</td></tr>
<tr><td>AGA8_GERGMethod</td><td>AGALib_V3</td><td>This function block is based on AGA 8 (2017) upgrade and available only from R161.2 release.

It calculates Gas Compressibility, Density and Gas Super-compressibility at base, standard and flowing condition that is flowing temperature and pressure based on the input parameters defined below. Apart from this, it also calculating speed of sound in gas that can be used in health monitoring of Ultrasonic meters. It is used when all 21 gas composition elements are available to get more accurate densities.</td></tr>
<tr><td>AGA3_Orifice</td><td>AGALib, AGALib_V2</td><td>They calculate the volumetric flow rate</td></tr>
</table>

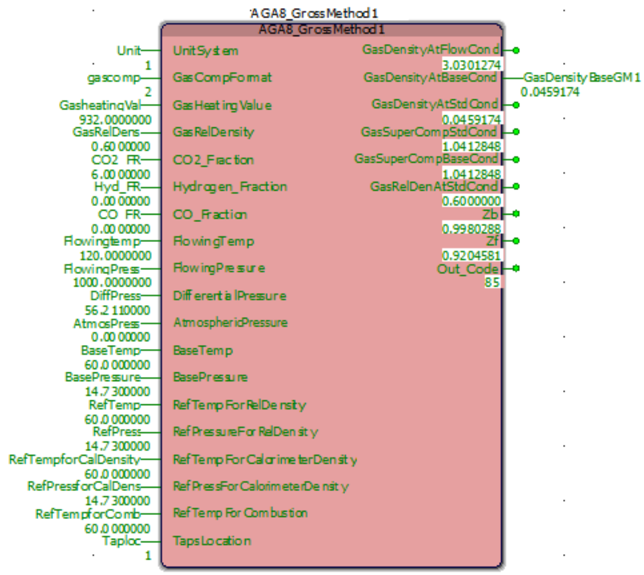
Function Block	Apply to	Description
	and AGALib_V3	for an orifice meter using either flange or pipe tap.
AGA3_Orifice_st	AGALib and AGALib_V2	
AGA3_Orifice_LIQ	AGALib_V3	This standard is based on AGA 3 (2012) update and this block is available from R161.2 release. It calculates the volumetric flow-rate of Liquid for an orifice meter using flange or pipe tap based on the input parameters defined below. It is used along with any one of the API 11.1 function blocks since it requires densities @flowing, standard and base conditions.
AGA7_Turbine & AGA9_Ultrasonic	AGALib, AGALib_V2 and AGALib_V3	• AGA7_Turbine and AGA_Turbine_st correct measured volume at flowing conditions read by turbine to volume at base conditions • AGA9_Ultrasonic and AGA9_Ultrasonic_st correct measured volume at flowing conditions read by ultrasonic to volume at base conditions
AGA7_Turbine_st & AGA9_Ultrasonic_st	AGALib and AGALib_V2	
AGA11_Coriolis	AGALib, AGALib_V2 and AGALib_V3	AGA11_Coriolis converts gas mass to volume. Gas mass is directly measured from Coriolis Meter.
AGA5_HV_CONSTANT	AGALib, AGALib_V2 and AGALib_V3	AGA5_HV_CONSTANT calculates the gas flow energy.
AGA5_DETAIL	AGALib, AGALib_V2 and AGALib_V3	They calculate the gas flow energy and Heating value.
AGA5_DETAIL_st	AGALib and AGALib_V2	

AGA8_GrossMethod1

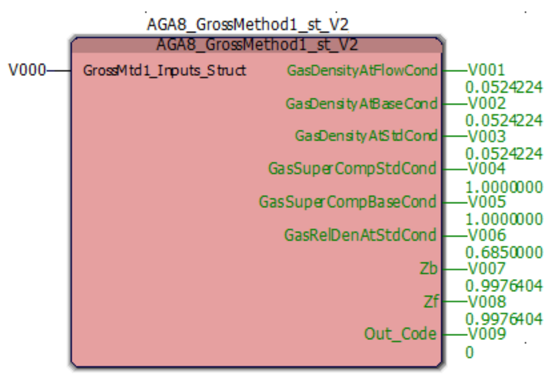
Description

AGA8_GrossMethod1 and AGA8_GrossMethod1_st calculate Gas Compressibility, Density and Gas Super-compressibility at base, standard and flowing condition that is flowing temperature and pressure based on the input parameters defined below.

- AGA8_GrossMethod1



- AGA8_GrossMethod1_st (This function block is with input parameters in the form of structures to make function block organized and compact.)



> **TIP:** AGA8_GrossMethod1 expects the input parameters either to be in US unit system or Metric unit system.

Input

Input Parameter	Data types	Description
UnitSystem	INT	{1} for US unit system and {2} for Metric unit system
GasCompFormat	INT	Gas Composition Format: • {1} for Mole Fraction • {2} for Percentage **NOTE:** It is recommended to use 2 percentage as a default option.
GasHeatingValue	LREAL	It's US unit is BTU/FT^3 and Metric unit is MJ/M^3.
GasRelDensity	LREAL	It is unitless number.
CO2_Fraction	LREAL	It can be in Mole Fraction or Percentage.
Hydrogen_Fraction	LREAL	It can be in Mole Fraction or Percentage.
CO_Fraction	LREAL	It can be in Mole Fraction or Percentage.
FlowingTemp	LREAL	It is in FAHRENHEIT for US unit and Celcius for Metric unit.
FlowingPressure	LREAL	It is in PSIA for US unit and Kpa for Metric unit.
DifferentialPressure	LREAL	It is in INH2O for US unit and in Kpa for Metric unit. It is used to adjust flowing pressure when the tap location is DOWNSTREAM.
AtmosphericPressure	LREAL	It is in PSIA for US unit and Kpa for Metric unit. Atmospheric pressure is used to make

Input Parameter	Data types	Description
		Flowing Pressure absolute when flowing pressure is measured by a pressure gauge. If flowing pressure is already absolute then it can be left zero.
BaseTemp	LREAL	It is in FAHRENHEIT for US unit and Celcius for Metric unit. The recommended default is 60 ° Fahrenheit.
BasePressure	LREAL	It is in PSIA for US unit and Kpa for Metric unit. The recommended default is 14.73 PSIA.
RefTempForRelDensity	LREAL	It is in FAHRENHEIT for US unit and Celcius for Metric unit. The recommended default is 60 ° Fahrenheit for reference temperature.
RefPressureForRelDensity	LREAL	It is in PSIA for US unit and Kpa for Metric unit. The recommended default is 14.73 PSIA for reference pressure.
RefTempForCalorimeterDensity	LREAL	It is the reference temperature for Calorimeter Density. It is in FAHRENHEIT for US unit and Celcius for Metric unit. Recommended default value is 60 ° Fahrenheit.
RefPressForCalorimeterDensity	LREAL	It is the reference pressure for Calorimeter Density. It is in PSIA for US unit and Kpa for Metric unit. Recommended default value is 14.73 PSIA.
RefTempForCombustion	LREAL	It is reference temperature for combustion. It is in FAHRENHEIT for US unit and Celcius for Metric unit. Recommended default value is 60 ° Fahrenheit.
TapsLocation	INT	It is unitless number. 1 is for UPSTREAM location and 2 is for DOWNSTREAM

Input Parameter	Data types	Description
		location.

Output

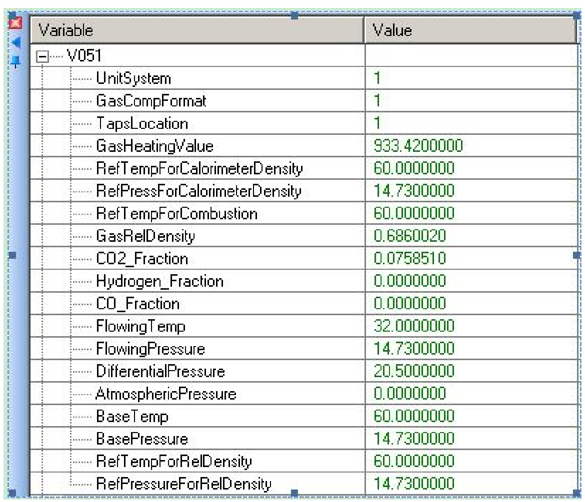
Output Parameter	Data types	Description
GasDensityAtFlowCond	LREAL	It is Gas Density at flowing temperature and pressure. It will be in KG/M^3 for (Metric System) & LBM/FT^3 for (US System). It is an input to AGA3 Function Block.
GasDensityAtBaseCond	LREAL	It is Gas Density at base temperature and pressure. It will be in KG/M^3 for (Metric System) & LBM/FT^3 for (US System). It is an input to AGA3 Function Block.
GasDensityAtStdCond	LREAL	It is Gas Density at standard temperature and pressure. It will be in KG/M^3 for (Metric System) & LBM/FT^3 for (US System). It is an input to AGA3 Function Block.
GasSuperComp	LREAL	It is gas super-compressibility factor. It is an input to AGA3 Function Block. It is unitless. The output is applicable only for AGALib.
GasSuperCompStdCond	LREAL	It is gas super-compressiblity factor at standard conditions. It is an input to AGA 7/9 function block and it is unitless. The output is applicable only for AGALib_V2 and AGALib_V3.
GasSuperCompBaseCond	LREAL	It is gas super-compressiblity factor at standard conditions. It is an input to AGA 3 function block and it is unitless. The output is applicable only for AGALib_V2 and AGALib_V3.
GasRelDenAtStdCond	LREAL	It is Gas Relative Density at standard temperature and pressure. It is an input to AGA3 Function Block. It is unitless.
Zb	LREAL	It is gas compressibility factor at base condition.
Zf	LREAL	It is gas compressibility factor at flowing

Output Parameter	Data types	Description
		condition.
Out_Code	INT	This out parameter returns success or fail code.

TIP: The output parameters must be in the same unit as of inputs.

For AGA8_GrossMethod1_st function block input structure is user defined data type. This is defined in Aga_Data Types under Data Types in IEC Programming Workspace.

Input Parameter	Data types
GrossMtd1_Inputs_Struct	AGA8_GrossMtd1_Inputs_STRUCT



Variable	Value
V051	
UnitSystem	1
GasCompFormat	1
TapsLocation	1
GasHeatingValue	933.4200000
RefTempForCalorimeterDensity	60.0000000
RefPressForCalorimeterDensity	14.7300000
RefTempForCombustion	60.0000000
GasRelDensity	0.6860020
CO2_Fraction	0.0758510
Hydrogen_Fraction	0.0000000
CO_Fraction	0.0000000
FlowingTemp	32.0000000
FlowingPressure	14.7300000
DifferentialPressure	20.5000000
AtmosphericPressure	0.0000000
BaseTemp	60.0000000
BasePressure	14.7300000
RefTempForRelDensity	60.0000000
RefPressureForRelDensity	14.7300000

Following is the table that describes different out code for both AGA8_GrossMethod1 and AGA8_GrossMethod2 function blocks:

Out Code	Description	Apply to
0	SUCCESS	All
5	ERROR: THE ROOT WAS NOT BOUNDED IN DGROSS	All
6	ERROR: NO CONVERGENCE IN DGROSS	All
7	ERROR: VIRGS SQURE ROOT NEGATIVE	All
8	ERROR: COMBINED VALUES OF GRGR, X[2] AND HV	All

Out Code	Description	Apply to
	NOT CONSISTENT	
9	ERROR: INVALID TERM IN VIRGS	All
11	ERROR: METHOD WAS NOT 1 OR 2	All
12	ERROR: FLOWING PRESSURE (PF) <= 0.0 OR > 1740.0 PSIA	All
13	ERROR: FLOWING TEMPERATURE (TF) < 14.0 OR > 149.0 DEG F	All
14	ERROR: HEATING VALUE (HV) < 477.0 OR > 1211.0 BTU/FT^3	AGA8_ GrossMethod1 AGA8_ GrossMethod1_V2 AGA8_ GrossMethod1_V3
15	ERROR: GAS RELATIVE DENSITY (GRGR) < 0.55 OR > 0.870	All
16	ERROR: MOLE FRACTION FOR N2 < 0.0 OR > 0.50 OR FOR CO2 < 0.0 OR > 0.30 OR FOR H2 < 0.0 OR > 0.10 OR FOR CO < 0.0 OR > 0.03	All
17	ERROR: REFERENCE TEMPERATURE < 32.0 OR > 77.0 DEG F	All
18	ERROR: REFERENCE PRESSURE < 13.0 OR > 16.0 PSIA	All
22	WARNING: FLOWING PRESSURE (PF) <= 0.0 OR > 1200.0 PSIA	AGA8_ GrossMethod1&2 AGA8_ GrossMethod1&2_ V2
23	WARNING: FLOWING TEMPERATURE (TF) < 32.0 OR > 130.0 DEG F	AGA8_ GrossMethod1&2 AGA8_ GrossMethod1&2_ V2

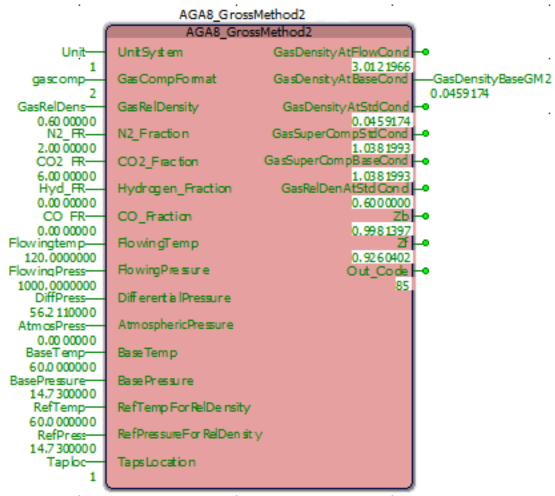
Out Code	Description	Apply to
24	WARNING: HEATING VALUE (HV) < 805.0 OR > 1208.0 BTU/FT^3	AGA8_ GrossMethod1 AGA8_ GrossMethod1_V2
25	WARNING: GAS RELATIVE DENSITY (GRGR) < 0.55 OR > 0.800	AGA8_ GrossMethod1&2 AGA8_ GrossMethod1&2_ V2
26	WARNING: MOLE FRACTION FOR N2 < 0.0 OR > 0.20 OR FOR CO2 < 0.0 OR > 0.20 OR FOR H2 < 0.0 OR > 0.0 OR FOR CO < 0.0 OR >0.0	AGA8_ GrossMethod1&2 AGA8_ GrossMethod1&2_ V2
81	WARNING: FLOWING PRESSURE (PF) > 1500.0 PSIA AGA8 2017 RANGE 1	AGA8_ GrossMethod1&2_ V3
82	WARNING: FLOWING TEMPERATURE (TF) < 17.01 OR > 143.0 DEG F AGA8 2017 RANGE 2 OR (TF) < 25.0 OR > 143.0 DEG F AGA8 2017 RANGE 1	AGA8_ GrossMethod1&2_ V3
83	WARNING: HEATING VALUE (HV) < 665.0 OR > 1100.0 BTU/FT^3 AGA8 2017 RANGE 2 OR (HV) < 930.0 OR > 1040.0 BTU/FT^3 AGA8 2017 RANGE 1	AGA8_ GrossMethod1_V3
84	WARNING: GAS RELATIVE DENSITY (GRGR) < 0.554 OR > 0.801 AGA8 2017 RANGE 2 OR (GRGR) < 0.554 OR > 0.630 AGA8 RANGE 1	AGA8_ GrossMethod1&2_ V3
85	WARNING: MOLE FRACTION FOR N2 > 0.20 AGA8 2017 RANGE 2 OR N2 > 0.07 AGA8 2017 RANGE 1 OR FOR CO2 > 0.25 AGA8 2017 RANGE 2 OR CO2 > 0.03 AGA8 2017 RANGE 1 OR FOR H2 < 0.0 OR > 0.0 AGA8 2017 RANGE 1 AND 2 OR FOR CO < 0.0 OR > 0.0 AGA8 2017 RANGE 1 AND 2	AGA8_ GrossMethod1&2_ V3

AGA8_GrossMethod2

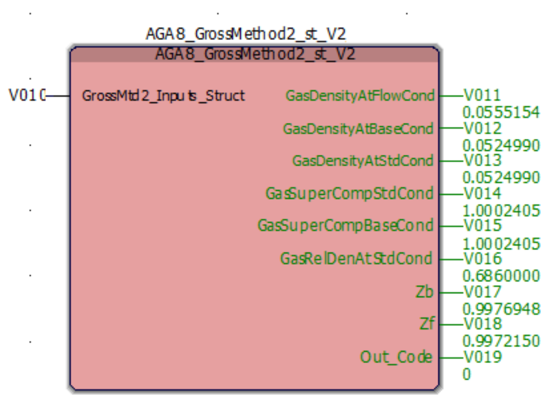
Description

This AGA8_GrossMethod2 function block calculates Gas Compressibility, Density and Gas Super-compressibility at base, standard and flowing condition that is flowing temperature and pressure based on the input parameters defined below. It takes Nitrogen fraction as input but does not take gas heating value. This description is applicable to following function blocks

- AGA8_GrossMethod2



- AGA8_GrossMethod2_st (This function block is having input parameters in the form of structures to make function block organized and compact)



This function block has input parameters in the form of structures to make the function block organized and compact.

Input

Input Parameter	Data types	Description
UnitSystem	INT	{1} for US unit system and {2} for Metric unit system
GasCompFormat	INT	Gas Composition Format - {1} for Mole Fraction and {2} for Percentage **NOTE:** It is recommended to use 2 percentage as a default option.
GasRelDensity	LREAL	It is unit less number
N2_Fraction	LREAL	It can be in Mole Fraction or Percentage
CO2_Fraction	LREAL	It can be in Mole Fraction or Percentage
Hydrogen_Fraction	LREAL	It can be in Mole Fraction or Percentage
CO_Fraction	LREAL	It can be in Mole Fraction or Percentage
FlowingTemp	LREAL	It is in FAHRENHEIT for US unit and Celcius for Metric unit.
FlowingPressure	LREAL	It is in PSIA for US unit and Kpa for Metric unit.
DifferentialPressure	LREAL	It is in INH2O for US unit and in Kpa for Metric unit. It is used to adjust flowing pressure when the tap location is DOWNSTREAM.
AtmosphericPressure	LREAL	It is in PSIA for US unit and Kpa for Metric unit. Atmospheric pressure is used to make Flowing Pressure absolute when flowing pressure is measured by a pressure gauge. If flowing pressure is already absolute then it can be left zero.
BaseTemp	LREAL	It is in FAHRENHEIT for US unit and Celcius for Metric unit. The recommended default is 60 Deg F.

Input Parameter	Data types	Description
BasePressure	LREAL	It is in PSIA for US unit and Kpa for Metric unit. The recommended default is 14.73 PSIA
RefTempForRelDensity	LREAL	It is in FAHRENHEIT for US unit and Celcius for Metric unit. The recommended default is 60 Deg F for reference temperature.
RefPressureForRelDensity	LREAL	It is in PSIA for US unit and Kpa for Metric unit. The recommended default is 14.73 PSIA for reference pressure.
TapsLocation	INT	It is unit less number. It is 1 for UPSTREAM location and 2 for DOWNSTREAM location

Ouput

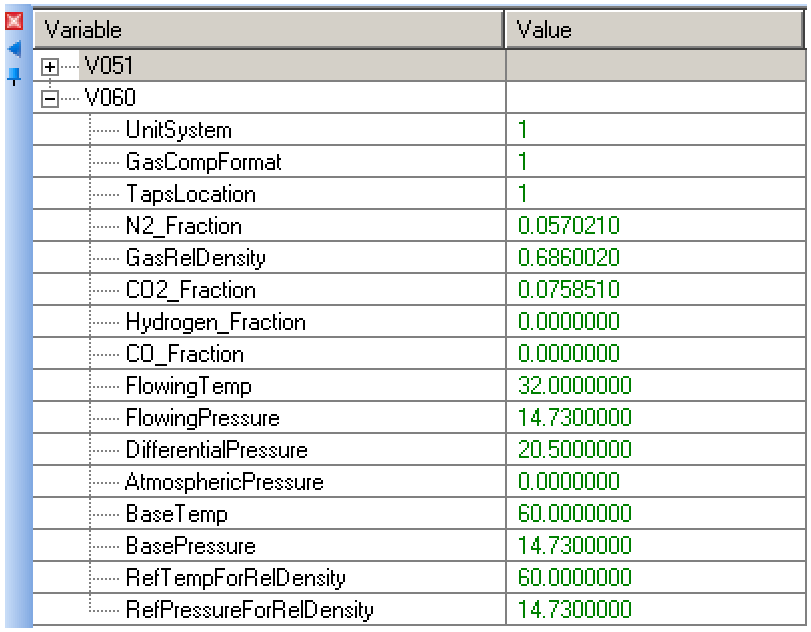
Output Parameter	Data types	Description
GasDensityAtFlowCond	LREAL	It is Gas Density at flowing temperature and pressure. It will be in KG/M^3 for (Metric System) & LBM/FT^3 for (US System).It is an input to AGA3 Function Block
GasDensityAtBaseCond	LREAL	It is Gas Density at base temperature and pressure. It will be in KG/M^3 for (Metric System) & LBM/FT^3 for (US System). It is an input to AGA3 Function Block
GasDensityAtStdCond	LREAL	It is Gas Density at standard temperature and pressure. It will be in KG/M^3 for (Metric System) & LBM/FT^3 for (US System). It is an input to AGA3 Function Block
GasSuperComp	LREAL	It is gas super compressibility factor. It is an input to AGA3 Function Block. It is unit less. The output is applicable only for AGALib.
GasRelDenAtStdCond	LREAL	It is Gas Relative Density at standard temperature and pressure. It is an input to AGA3 Function Block. It is unit less.

Output Parameter	Data types	Description
GasSuperCompStdCond	LREAL	It is gas supercompressiblity factor at standard conditions. It is an input to AGA 7/9 function block and it is unitless. The output is applicable only for AGALib_V2 and AGALib_V3.
GasSuperCompBaseCond	LREAL	It is gas supercompressiblity factor at standard conditions. It is an input to AGA 3 function block and it is unitless. The output is applicable only for AGALib_V2 and AGALib_V3.
Zb	LREAL	It is gas compressibility factor at base condition.
Zf	LREAL	It is gas compressibility factor at flowing condition.
Out_Code	INT	This out parameter returns success or fail code.

TIP: AGA8_GrossMethod2 expects the input parameters to be in either US unit system or Metric unit system. The output parameters would be in the same unit as of inputs.

For AGA8_GrossMethod2_st function block input structure is user defined data type. This is defined in Aga_Data Types under Data Types in IEC Programming Workspace.

Input Parameter	Data types
GrossMtd2_Inputs_Struct	AGA8_GrossMtd2_Inputs_STRUCT



Variable	Value
V051	
V060	
UnitSystem	1
GasCompFormat	1
TapsLocation	1
N2_Fraction	0.0570210
GasRelDensity	0.6860020
CO2_Fraction	0.0758510
Hydrogen_Fraction	0.0000000
CO_Fraction	0.0000000
FlowingTemp	32.0000000
FlowingPressure	14.7300000
DifferentialPressure	20.5000000
AtmosphericPressure	0.0000000
BaseTemp	60.0000000
BasePressure	14.7300000
RefTempForRelDensity	60.0000000
RefPressureForRelDensity	14.7300000

Following is the table that describes different out code for both AGA8_GrossMethod1 and AGA8_GrossMethod2 function blocks:

Out Code	Description	Apply to
0	SUCCESS	All
5	ERROR: THE ROOT WAS NOT BOUNDED IN DGROSS	All
6	ERROR: NO CONVERGENCE IN DGROSS	All
7	ERROR: VIRGS SQURE ROOT NEGATIVE	All
8	ERROR: COMBINED VALUES OF GRGR, X[2] AND HV NOT CONSISTENT	All
9	ERROR: INVALID TERM IN VIRGS	All
11	ERROR: METHOD WAS NOT 1 OR 2	All
12	ERROR: FLOWING PRESSURE (PF) <= 0.0 OR > 1740.0 PSIA	All
13	ERROR: FLOWING TEMPERATURE (TF) < 14.0 OR > 149.0 DEG F	All
14	ERROR: HEATING VALUE (HV) < 477.0 OR > 1211.0 BTU/FT^3	AGA8_ GrossMethod1 AGA8_

Out Code	Description	Apply to
		GrossMethod1_V2 AGA8_ GrossMethod1_V3
15	ERROR: GAS RELATIVE DENSITY (GRGR) < 0.55 OR > 0.870	All
16	ERROR: MOLE FRACTION FOR N2 < 0.0 OR > 0.50 OR FOR CO2 < 0.0 OR > 0.30 OR FOR H2 < 0.0 OR > 0.10 OR FOR CO < 0.0 OR > 0.03	All
17	ERROR: REFERENCE TEMPERATURE < 32.0 OR > 77.0 DEG F	All
18	ERROR: REFERENCE PRESSURE < 13.0 OR > 16.0 PSIA	All
22	WARNING: FLOWING PRESSURE (PF) <= 0.0 OR > 1200.0 PSIA	AGA8_ GrossMethod1&2 AGA8_ GrossMethod1&2_ V2
23	WARNING: FLOWING TEMPERATURE (TF) < 32.0 OR > 130.0 DEG F	AGA8_ GrossMethod1&2 AGA8_ GrossMethod1&2_ V2
24	WARNING: HEATING VALUE (HV) < 805.0 OR > 1208.0 BTU/FT^3	AGA8_ GrossMethod1 AGA8_ GrossMethod1_V2
25	WARNING: GAS RELATIVE DENSITY (GRGR) < 0.55 OR > 0.800	AGA8_ GrossMethod1&2 AGA8_ GrossMethod1&2_ V2
26	WARNING: MOLE FRACTION FOR N2 < 0.0 OR > 0.20 OR FOR CO2 < 0.0 OR > 0.20	AGA8_ GrossMethod1&2

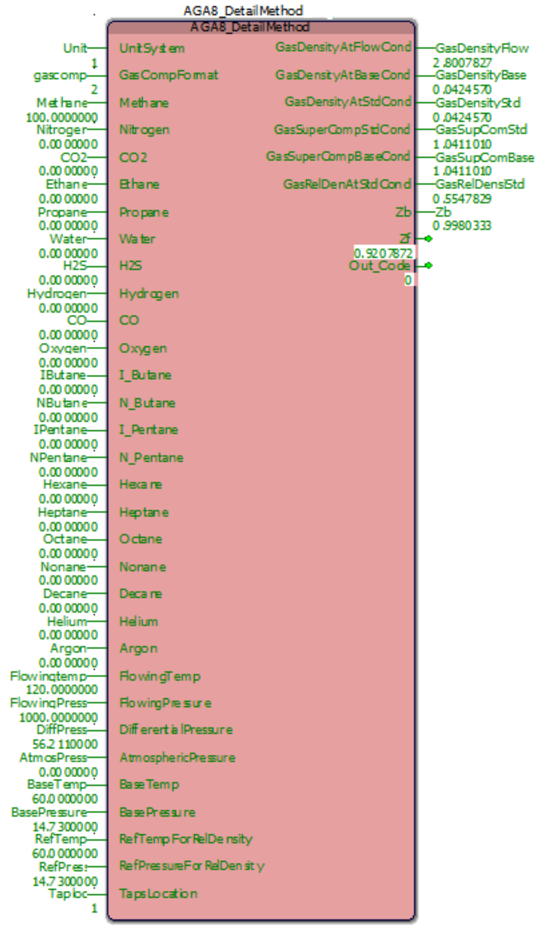
Out Code	Description	Apply to
	OR FOR H2 < 0.0 OR > 0.0 OR FOR CO < 0.0 OR >0.0	AGA8_ GrossMethod1&2_ V2
81	WARNING: FLOWING PRESSURE (PF) > 1500.0 PSIA AGA8 2017 RANGE 1	AGA8_ GrossMethod1&2_ V3
82	WARNING: FLOWING TEMPERATURE (TF) < 17.01 OR > 143.0 DEG F AGA8 2017 RANGE 2 OR (TF) < 25.0 OR > 143.0 DEG F AGA8 2017 RANGE 1	AGA8_ GrossMethod1&2_ V3
83	WARNING: HEATING VALUE (HV) < 665.0 OR > 1100.0 BTU/FT^3 AGA8 2017 RANGE 2 OR (HV) < 930.0 OR > 1040.0 BTU/FT^3 AGA8 2017 RANGE 1	AGA8_ GrossMethod1_V3
84	WARNING: GAS RELATIVE DENSITY (GRGR) < 0.554 OR > 0.801 AGA8 2017 RANGE 2 OR (GRGR) < 0.554 OR > 0.630 AGA8 RANGE 1	AGA8_ GrossMethod1&2_ V3
85	WARNING: MOLE FRACTION FOR N2 > 0.20 AGA8 2017 RANGE 2 OR N2 > 0.07 AGA8 2017 RANGE 1 OR FOR CO2 > 0.25 AGA8 2017 RANGE 2 OR CO2 > 0.03 AGA8 2017 RANGE 1 OR FOR H2 < 0.0 OR > 0.0 AGA8 2017 RANGE 1 AND 2 OR FOR CO < 0.0 OR > 0.0 AGA8 2017 RANGE 1 AND 2	AGA8_ GrossMethod1&2_ V3

AGA8_DetailMethod

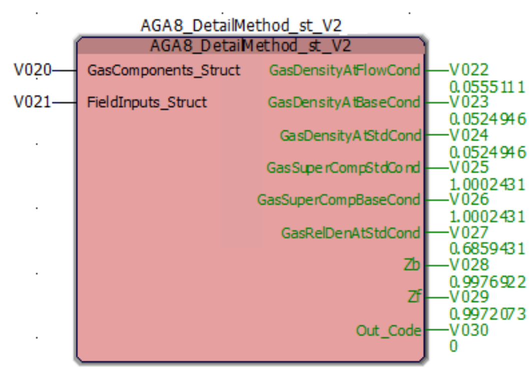
Description

This AGA8_DetailMethod function block calculates Gas Compressibility, Density and Gas Super-compressibility at base, standard and flowing condition that is flowing temperature and pressure based on the input parameters defined below. It is used when all 21 gas composition elements are available to get more accurate densities. This description is applicable to following function blocks

- AGA8_DetailMethod



- AGA8_DetailMethod_st (This function block is having input parameters in the form of structures to make function block organized and compact)



Input

Input Parameter	Data types	Description
UnitSystem	INT	{1} for US unit system and {2} for Metric unit system
GasCompFormat	INT	Gas Composition Format: {1} for Mole Fraction and {2} for Percentage **NOTE:** It is recommended to use 2 percentage as a default option.
Methane	LREAL	It can be in Mole Fraction or Percentage
Nitrogen	LREAL	It can be in Mole Fraction or Percentage
CO2	LREAL	It can be in Mole Fraction or Percentage
Ethane	LREAL	It can be in Mole Fraction or Percentage
Propane	LREAL	It can be in Mole Fraction or Percentage
Water	LREAL	It can be in Mole Fraction or Percentage
H2S	LREAL	It can be in Mole Fraction or Percentage
Hydrogen	LREAL	It can be in Mole Fraction or Percentage
CO	LREAL	It can be in Mole Fraction or Percentage
Oxygen	LREAL	It can be in Mole Fraction or Percentage
I_Butane	LREAL	It can be in Mole Fraction or Percentage
N_Butane	LREAL	It can be in Mole Fraction or Percentage
I_Pentane	LREAL	It can be in Mole Fraction or Percentage
N_Pentane	LREAL	It can be in Mole Fraction or Percentage
Hexane	LREAL	It can be in Mole Fraction or Percentage
Heptane	LREAL	It can be in Mole Fraction or Percentage
Octane	LREAL	It can be in Mole Fraction or Percentage
Nonane	LREAL	It can be in Mole Fraction or Percentage
Decane	LREAL	It can be in Mole Fraction or Percentage

Input Parameter	Data types	Description
Helium	LREAL	It can be in Mole Fraction or Percentage
Argon	LREAL	It can be in Mole Fraction or Percentage
FlowingTemp	LREAL	It is in FAHRENHEIT for US unit and Celcius for Metric unit.
FlowingPressure	LREAL	It is in PSIA for US unit and Kpa for Metric unit.
DifferentialPressure	LREAL	It is in INH2O for US unit and in Kpa for Metric unit. It is used to adjust flowing pressure when the tap location is DOWNSTREAM.
AtmosphericPressure	LREAL	It is in PSIA for US unit and Kpa for Metric unit. Atmospheric pressure is used to make Flowing Pressure absolute when flowing pressure is measured by a pressure gauge. If flowing pressure is already absolute then it can be left zero.
BaseTemp	LREAL	It is in FAHRENHEIT for US unit and Celcius for Metric unit. The recommended default is 60 Deg F.
BasePressure	LREAL	It is in PSIA for US unit and Kpa for Metric unit. The recommended default is 14.73 PSIA
RefTempForRelDensity	LREAL	It is in FAHRENHEIT for US unit and Celcius for Metric unit. The recommended default is 60 Deg F for reference temperature.
RefPressureForRelDensity	LREAL	It is in PSIA for US unit and Kpa for Metric unit. The recommended default is 14.73 PSIA for reference pressure.
TapsLocation	INT	It is unit less number. It is 1 for UPSTREAM location and 2 for DOWNSTREAM location.

Ouput

Output Parameter	Data types	Description
GasDensityAtFlowCond	LREAL	It is Gas Density at flowing temperature and pressure. It will be in KG/M^3 for (Metric System) & LBM/FT^3 for (US System).It is an input to AGA3 Function Block
GasDensityAtBaseCond	LREAL	It is Gas Density at base temperature and pressure. It will be in KG/M^3 for (Metric System) & LBM/FT^3 for (US System). It is an input to AGA3 Function Block
GasDensityAtStdCond	LREAL	It is Gas Density at standard temperature and pressure. It will be in KG/M^3 for (Metric System) & LBM/FT^3 for (US System). It is an input to AGA3 Function Block
GasSuperComp	LREAL	It is gas super compressibility factor. It is an input to AGA3 Function Block. It is unit less. The output is applicable only for AGALib.
GasSuperCompStdCond	LREAL	It is gas supercompressiblity factor at standard conditions. It is an input to AGA 7/9 function block and it is unitless. The output is applicable only for AGALib_V2 and AGALib_V3.
GasSuperCompBaseCond	LREAL	It is gas supercompressiblity factor at standard conditions. It is an input to AGA 3 function block and it is unitless. The output is applicable only for AGALib_V2 and AGALib_V3.
GasRelDenAtStdCond	LREAL	It is Gas Relative Density at standard temperature and pressure. It is an input to AGA3 Function Block. It is unit less.
Zb	LREAL	It is gas compressibility factor at base condition. It is an input to AGA5_DETAIL function block.
Zf	LREAL	It is gas compressibility factor at flowing condition.
Out_Code	INT	This out parameter returns success or fail code.

> **TIP:** AGA8_Detail Method expects the input parameters to be either in US unit system or Metric unit system. It expects all the 21 elements of Gas composition coming out from Gas Analyzer or Chromatograph/ Flow Computer. Gas parameter should be either in mole fraction or in percentage.

The output parameters would be in the same unit as of inputs.

AGA8_DetailMethod_st function block input structures are user defined data type. They are defined in Aga_Data Types under Data Types in IEC Programming Workspace.

Input Parameter	Data types
GasComponents_Struct	DtlMtd_GasComps_STRUCT
FieldInputs_Struct	AGA8_DtlMtd_FldInputs_STRUCT

Variable	Value
V040	
Gas_Comps_Array	
[0]	0.8121100
[1]	0.0570200
[2]	0.0758500
[3]	0.0430300
[4]	0.0089500
[5]	0.0000000
[6]	0.0000000
[7]	0.0000000
[8]	0.0000000
[9]	0.0000000
[10]	0.0015100
[11]	0.0015200
[12]	0.0000000
[13]	0.0000000
[14]	0.0000000
[15]	0.0000000
[16]	0.0000000
[17]	0.0000000
[18]	0.0000000
[19]	0.0000000
[20]	0.0000000
V041	
UnitSystem	1
GasCompFormat	1
TapsLocation	1
FlowingTemp	32.0000000
FlowingPressure	14.7300000
DifferentialPressure	20.5000000
AtmosphericPressure	0.0000000
BaseTemp	60.0000000
BasePressure	14.7300000
RefTempForRelDensity	60.0000000
RefPressureForRelDensity	14.7300000

Following is the table that describes different out code for AGA8_ DetailMethod function blocks.

Out Code	Description	Apply to
0	SUCCESS	All
1	ERROR: PRESSURE HAS A NEGATIVE DERIVATIVE DEFAULT GAS DENSITY USED	All
2	WARNING: DENSITY IN BRAKET EXCEEDS MAXIMUM	All

Out Code	Description	Apply to
	DEFAULT PROCEEDURE USED	
3	ERROR: MAXIMUM ITERATIONS EXCEEDED IN BRAKET DEFAULT DENSITY USED	All
4	ERROR: MAXIMUM ITERATIONS IN DDETAIL EXCEEDED LAST DENSITY USED	All
32	ERROR: FLOWING PRESSURE (PF) <= 0.0 OR > 40,000. PSIA	All
33	ERROR: FLOWING TEMPERATURE (TF) < -200 OR > 760 DEG F	All
35	ERROR: ANY COMPONENT MOLE FRACTION < 0.0 OR > 1.0	AGA8_ DetailMethod_ V3
36	ERROR: MOLE FRACTION FOR METHANE < 0.0 OR > 1.0 FOR NITROGEN < 0.0 OR > 1.0 FOR CARBON DIOXIDE < 0.0 OR > 1.0 FOR ETHANE < 0.0 OR > 1.0 FOR PROPANE < 0.0 OR > 0.12 FOR WATER < 0.0 OR > 0.10 FOR H2S < 0.0 OR > 1.0 FOR HYDROGEN < 0.0 OR > 1.0 FOR CARBON MONOXIDE < 0.0 OR > 0.03 FOR OXYGEN < 0.0 OR > 0.21 FOR BUTANES < 0.0 OR > 0.06 FOR PENTANES < 0.0 OR > 0.04 FOR HEXANES + < 0.0 OR > 0.10 FOR HELIUM < 0.0 OR > 0.03 FOR ARGON < 0.0 OR > 1.0	AGA8_ DetailMethod & AGA8_ DetailMethod_ V2
37	ERROR: REFERENCE TEMPERATURE < 32.0 OR > 77.0 DEG F	All
38	ERROR: REFERENCE PRESSURE < 13.0 OR > 16.0 PSIA	All
39	ERROR: SUM OF MOLE FRACTIONS < 0.98 OR > 1.020	All
42	WARNING: FLOWING PRESSURE (PF) < 0.0 OR > 1750. PSIA	All
43	WARNING: FLOWING TEMPERATURE (TF) < 17 OR > 143 DEG F	All
45	WARNING: ANY COMPONENT MOLE FRACTION OUTSIDE OF AGA REPORT NO. 8 RECOMMENDED RANGE	AGA8_ DetailMethod_

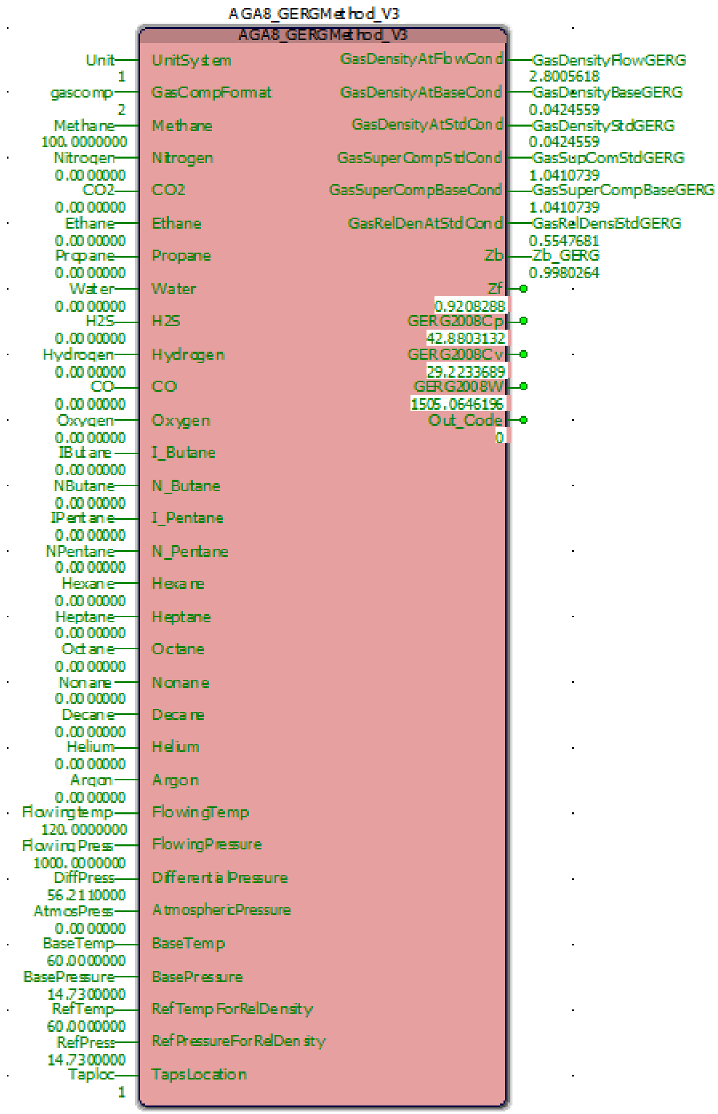
Out Code	Description	Apply to
		V3
46	WARNING: MOLE FRACTION FOR METHANE < 0.45 OR > 1.0 FOR NITROGEN < 0.0 OR > 0.5 FOR CARBON DIOXIDE < 0.0 OR > 0.3 FOR ETHANE < 0.0 OR > 0.1 FOR PROPANE < 0.0 OR > 0.04 FOR WATER < 0.0 OR >= 0.0005 FOR H2S < 0.0 OR > 0.0002 FOR HYDROGEN < 0.0 OR > 0.1 FOR CARBON MONOXIDE < 0.0 OR > 0.03 FOR OXYGEN < 0.0 OR > 0.0 FOR BUTANES < 0.0 OR > 0.01 FOR PENTANES < 0.0 OR >= 0.003 FOR HEXANES + < 0.0 OR >= 0.002 FOR HELIUM < 0.0 OR >= 0.002 FOR ARGON < 0.0 OR > 0.0	AGA8_ DetailMethod & AGA8_ DetailMethod_ V2
49	WARNING: SUM OF MOLE FRACTIONS < 0.9999 OR > 1.0001	All

AGA8_GERGMethod

Description

This AGA8_GERGMethod function block calculates Gas Compressibility, Density and Gas Super-compressibility at base, standard and flowing condition that is flowing temperature and pressure based on the input parameters defined below. Apart from this, it also calculating speed of sound in gas that can be used in health monitoring of Ultrasonic meters. It is used when all 21 gas composition elements are available to get more accurate densities. This description is applicable to following function blocks. This function block is based on AGA 8 (2017) upgrade and available only from R161.2 release.

Figure 3-1: AGA8_GERGMethod function block



Input

Input Parameter	Data types	Description
UnitSystem	INT	{1} for US unit system and {2} for Metric unit system
GasCompFormat	INT	Gas Composition Format: • {1} for Mole Fraction

Input Parameter	Data types	Description
		• {2} for Percentage **NOTE:** It is recommended to use 2 percentage as a default option.
Methane	LREAL	It can be in Mole Fraction or Percentage
Nitrogen	LREAL	It can be in Mole Fraction or Percentage
CO2	LREAL	It can be in Mole Fraction or Percentage
Ethane	LREAL	It can be in Mole Fraction or Percentage
Propane	LREAL	It can be in Mole Fraction or Percentage
Water	LREAL	It can be in Mole Fraction or Percentage
H2S	LREAL	It can be in Mole Fraction or Percentage
Hydrogen	LREAL	It can be in Mole Fraction or Percentage
CO	LREAL	It can be in Mole Fraction or Percentage
Oxygen	LREAL	It can be in Mole Fraction or Percentage
I_Butane	LREAL	It can be in Mole Fraction or Percentage
N_Butane	LREAL	It can be in Mole Fraction or Percentage
I_Pentane	LREAL	It can be in Mole Fraction or Percentage
N_Pentane	LREAL	It can be in Mole Fraction or Percentage
Hexane	LREAL	It can be in Mole Fraction or Percentage
Heptane	LREAL	It can be in Mole Fraction or Percentage
Octane	LREAL	It can be in Mole Fraction or Percentage
Nonane	LREAL	It can be in Mole Fraction or Percentage
Decane	LREAL	It can be in Mole Fraction or Percentage
Helium	LREAL	It can be in Mole Fraction or Percentage
Argon	LREAL	It can be in Mole Fraction or Percentage
FlowingTemp	LREAL	It is in FAHRENHEIT for US unit and Celcius for Metric unit.

Input Parameter	Data types	Description
FlowingPressure	LREAL	It is in PSIA for US unit and Kpa for Metric unit.
DifferentialPressure	LREAL	It is in INH2O for US unit and in Kpa for Metric unit. It is used to adjust flowing pressure when the tap location is DOWNSTREAM.
AtmosphericPressure	LREAL	It is in PSIA for US unit and Kpa for Metric unit. Atmospheric pressure is used to make Flowing Pressure absolute when flowing pressure is measured by a pressure gauge. If flowing pressure is already absolute then it can be left zero.
BaseTemp	LREAL	It is in FAHRENHEIT for US unit and Celcius for Metric unit. The recommended default is 60 ° Fahrenheit.
BasePressure	LREAL	It is in PSIA for US unit and Kpa for Metric unit. The recommended default is 14.73 PSIA.
RefTempForRelDensity	LREAL	It is in FAHRENHEIT for US unit and Celcius for Metric unit. The recommended default is 60 ° Fahrenheit for reference temperature.
RefPressureForRelDensity	LREAL	It is in PSIA for US unit and Kpa for Metric unit. The recommended default is 14.73 PSIA for reference pressure.
TapsLocation	INT	It is unit less number. 1 is for UPSTREAM location and 2 is for DOWNSTREAM location.

Output

Output Parameter	Data types	Description
GasDensityAtFlowCond	LREAL	It is Gas Density at flowing temperature and pressure. It will be in KG/M^3 for (Metric System) & LBM/FT^3 for (US System). It is an input to AGA3 Function Block.

Output Parameter	Data types	Description
GasDensityAtBaseCond	LREAL	It is Gas Density at base temperature and pressure. It will be in KG/M^3 for (Metric System) & LBM/FT^3 for (US System). It is an input to AGA3 Function Block.
GasDensityAtStdCond	LREAL	It is Gas Density at standard temperature and pressure. It will be in KG/M^3 for (Metric System) & LBM/FT^3 for (US System). It is an input to AGA3 Function Block.
GasSuperCompStdCond	LREAL	It is gas super-compressiblity factor at standard conditions. It is an input to AGA 7/9 function block and it is unitless.
GasSuperCompBaseCond	LREAL	It is gas super-compressiblity factor at standard conditions. It is an input to AGA 3 function block and it is unitless.
GasRelDenAtStdCond	LREAL	It is Gas Relative Density at standard temperature and pressure. It is an input to AGA3 Function Block. It is unitless.
Zb	LREAL	It is gas compressibility factor at base condition.
Zf	LREAL	It is gas compressibility factor at flowing condition.
GERG2008Cp	LREAL	Heat Capacity at Constant Pressure (J/mol K).
GERG2008Cv	LREAL	Heat Capacity at Constant Volume (J/mol K).
GERG2008W	LREAL	Speed of sound in gas being measured. Unit - ft/sec for US, meter/sec for Metric.
Out_Code	INT	This out parameter returns success or fail code.

Following are the error codes for AGA8_GERGMethod function block.

Out Code	Description
0	NO WARNING OR ERROR
1	ERROR: PRESSURE HAS A NEGATIVE DERIVATIVE DEFAULT GAS DENSITY USED

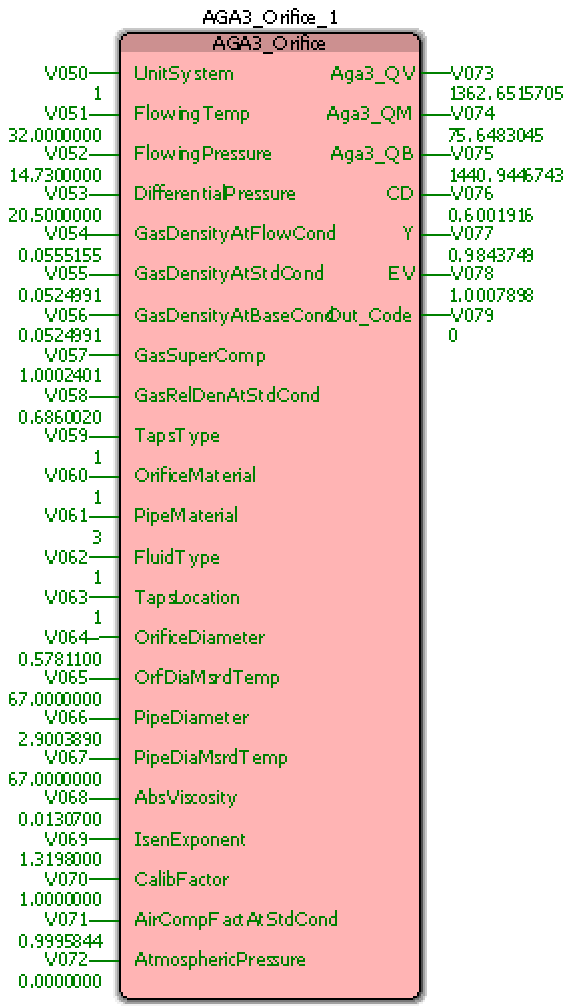
Out Code	Description
32	ERROR: FLOWING PRESSURE (PF) <= 0.0 OR > 40,000. PSIA
35	ERROR: ANY COMPONENT MOLE FRACTION < 0.0 OR > 1.0
86	WARNING: Flowing Pressure greater than 2017 AGA8 GERG-2008 Full Quality Range (10,150 PSIA)
87	WARNING: Flowing Pressure greater than 2017 AGA8 GERG-2008 Range (5075 PSIA)
88	WARNING: Flowing Temperature outside 2017 AGA8 GERG-2008 Full Quality Range (-352 F < TF < 800 F)
89	WARNING: Flowing Temperature outside 2017 AGA8 GERG-2008 Range (-298 F < TF < 350 F)
90	WARNING: A Component Mole % outside 2017 AGA8 GERG-2008 Intermediate Quality Range
91	WARNING: A Component Mole % outside 2017 AGA8 GERG-2008 Pipeline Quality Range

AGA3_Orifice

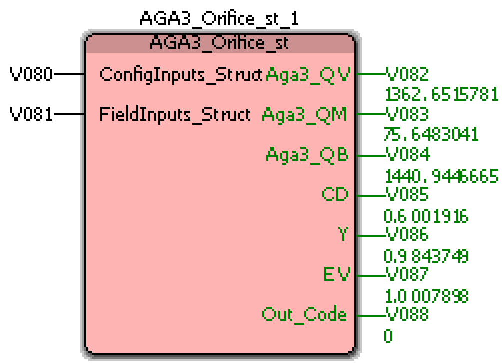
Description

This AGA3_Orifice function block calculates the volumetric flow-rate for an orifice meter using flange or pipe tap based on the input parameters defined below. It is used along with any one of the AGA8 function blocks since it requires densities @flowing, standard and base conditions as well as gas super compressibility and gas relative density @ standard condition coming out of AGA8 function block. This description is applicable to following function blocks

- AGA3_Orifice



- AGA3_Orifice_st (This function block is having input parameters in the form of structures to make function block organized and compact)



Input

Input Parameter	Data types	Description
UnitSystem	INT	{1} for US unit system and {2} for Metric unit system
FlowingTemp	LREAL	It is in FAHRENHEIT for US unit and Celcius for Metric unit.
FlowingPressure	LREAL	It is in PSIA for US unit and Kpa for Metric unit.
DifferentialPressure	LREAL	It is in INH2O for US unit and in Kpa for Metric unit.
GasDensityAtFlowCond	LREAL	It is Gas Density at flowing temperature and pressure. It is in KG/M^3 for (Metric System) & LBM/FT^3 for (US System). It is an output of AGA8.
GasDensityAtStdCond	LREAL	It is Gas Density at standard temperature and pressure. It is in KG/M^3 for (Metric System) & LBM/FT^3 for (US System).It is an output of AGA8.
GasDensityAtBaseCond	LREAL	It is Gas Density at base temperature and pressure. It is in KG/M^3 for (Metric System) & LBM/FT^3 for (US System).It is an output of AGA8.
GasSuperComp	LREAL	It is unit less number. It is an output of AGA8
GasRelDenAtStdCond	LREAL	It is unit less number. It is an output of AGA8

Input Parameter	Data types	Description
TapsType	INT	FLANGE=1 and PIPE=2
OrificeMaterial	INT	STAINLESS STEEL=1, MONEL=2, CARBON STEEL=3, STAINLESS_S_304=4 and STAINLESS_S_316=5
PipeMaterial	INT	STAINLESS STEEL=1, MONEL=2, CARBON STEEL=3, STAINLESS_S_304=4 and STAINLESS_S_316=5
FluidType	INT	COMPRESSIBLE FLUID =1 and NON-COMPRESSIBLE FLUID=2
TapsLocation	INT	UPSTREAM=1 and DOWNSTREAM=2
OrificeDiameter	LREAL	It is in inches for US unit system & in milimeter for Metric unit system.
OrfDiaMsrdTemp	LREAL	It is in FAHRENHEIT for US unit and Celcius for Metric unit.
PipeDiameter	LREAL	It is in inches for US unit system & in milimeter for Metric unit system.
PipeDiaMsrdTemp	LREAL	It is in FAHRENHEIT for US unit and Celcius for Metric unit.
AbsViscosity	LREAL	AGA3 Orifice method expects absolute viscosity in CENTIPOISE unit only for both US & Metric unit system. ABSOLUTE VISCOSITY OF FLUID FLOWING. (RECOMMENDED DEFAULT=0.010268 cP - PG 34 PART 4)
IsenExponent	LREAL	ISENTROPIC EXPONENT is unit less number. (RECOMMENDED DEFAULT=1.3 - PG 34 PART 4)
CalibFactor	LREAL	It is unit less number. Default value is 1.0
AirCompFactAtStdCond	LREAL	It is unit less number. COMPRESSIBILITY FACTOR OF AIR AT standard temperature and pressure. It is used for Pipe tap only.
AtmosphericPressure	LREAL	It is in PSIA for US unit and Kpa for Metric unit.

Input Parameter	Data types	Description
		If flowing pressure is already absolute, it can be left zero.

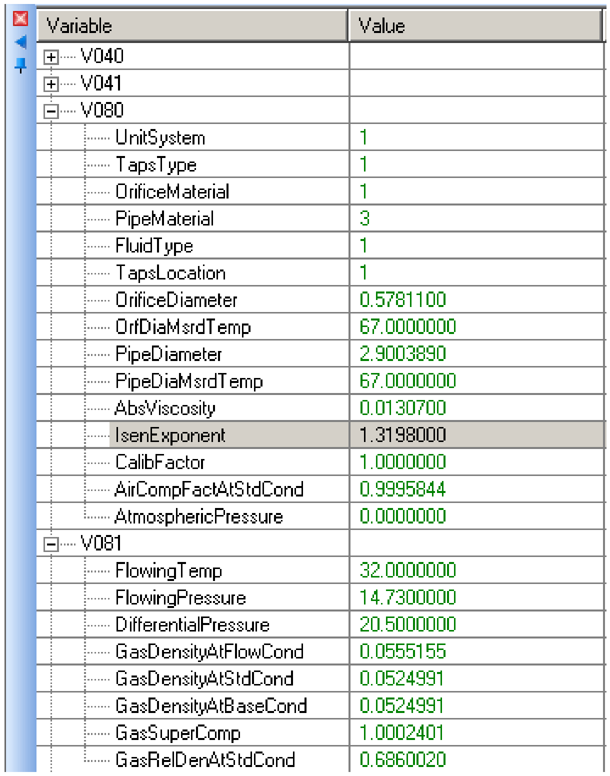
Output

Output Parameter	Data types	Description
Aga3_QV	LREAL	This is volume flow rate at standard (TS & PS) conditions. It is in Scf/Hr for US unit system and in cubic meter per hour in Metric unit system.
Aga3_QM	LREAL	This is mass flow rate. It is in Lbm/Hr for US unit system and Kg/Hr for Metric unit system.
Aga3_QB	LREAL	This is volumetric flow rate at base (TB & PB) conditions. It is in Scf/Hr for US unit system and in cubic meter per hour in Metric unit system.
CD	LREAL	It is Orifice plate coefficient of discharge (flange).
Y	LREAL	It is Expansion factor (flange and pipe).
EV	LREAL	It is Velocity of approach factor (flange).
Out_Code	INT	This out parameter returns success or fail code.

- AGA3_Orifice expects the input parameters to be either in US unit system or Metric unit system. Output parameters of the AGA8 function blocks become input parameters for this function block.
- The output parameters would be in the same unit as of inputs.

For AGA3_Orifice_st function block input structures are user defined data type. They are defined in Aga_Data Types under Data Types in IEC Programming Workspace.

Input Parameter	Data types
FieldInputs_Struct	AGA3_Orifice_FldInputs_STRUCT
ConfigInputs_Struct	AGA3_Config_Inputs_STRUCT



Following is the table that describes different out code for AGA3_ Orifice function blocks.

Out Code	Description	Apply to
0	SUCCESS, NO WARNING OR ERROR	All
51	ERROR: NTAPS WAS NOT 0, 1 OR 2	All
52	ERROR: FLOWING PRESSURE WAS <= 0.0 OR > 40000. PSIA	All
53	ERROR: FLOWING TEMPERATURE < -200. OR > 760. DEG F	All
54	ERROR: MATORF OR MATPIPE WAS NOT 0, 1, 2 OR 3	All
55	ERROR: ORIFICE DIAMETER WAS <= 0 OR => 100.0 INCHES	All
56	ERROR: PIPE DIAMETER WAS <= 0 OR => 100.0 INCHES	All
57	ERROR: FLOWING OR STANDARD DENSITY WAS <= 0.0 LBM/FT^3	All
58	ERROR: DIFFERENTIAL PRESSURE WAS <= 0.0 INCHES H2O	All
59	ERROR: GAS VISCOSITY WAS <= 0.005 OR > 0.5 CENTIPOISES	All
60	ERROR: ISENTROPIC EXPONENT <= 1.0 OR => 2.0	All

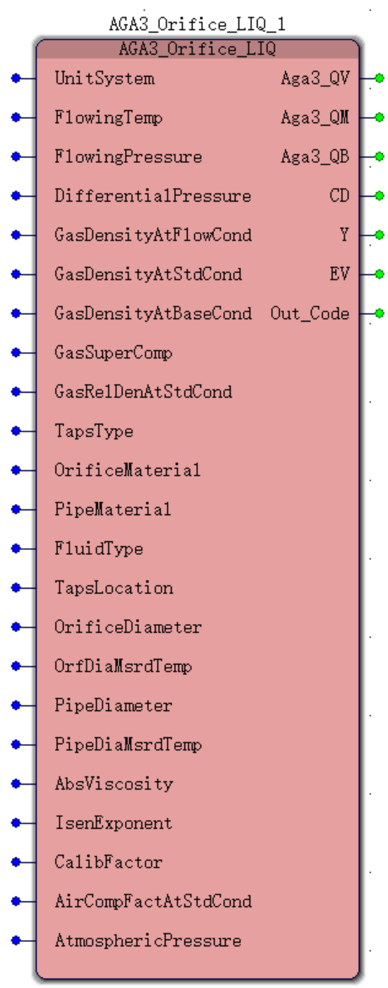
Out Code	Description	Apply to
61	ERROR: IFLUID WAS NOT 0, 1 OR 2	All
62	ERROR: STANDARD TEMPERATURE WAS NOT = 60.0 DEG F	All
63	ERROR: STANDARD PRESSURE WAS NOT = 14.73 PSIA	All
64	ERROR: TAP LOCATION WAS NOT 0, 1 OR 2 FOR NTAPS=2 (PIPE) OR TAP LOCATION WAS NOT 1 FOR NTAPS=1 (FLANGE)	All
65	ERROR: SUPERCOMPRESSIBILITY FACTOR WAS <= 0.0	All
66	ERROR: RELATIVE DENSITY AT STANDARD CONDITIONS WAS < 0.07 OR > 1.52	All
67	ERROR: CALIBRATION FACTOR WAS <= 0.0	All
68	ERROR: COMPRESSIBILITY FACTOR AT STANDARD CONDITIONS <= 0.0	All
69	ERROR: BETA RATIO (DO/DM) <= 0.0 OR => 1.0	All
70	ERROR: IF NTAPS = 1, GOF2015_OPTION NOT = 1 OR = 0	AGA3_ Orifice_ V3
71	ERROR: IF NTAPS = 2, GOF2015_OPTION NOT = 0	AGA3_ Orifice_ V3
72	ERROR: DIFFERENTIAL PRESSURE WAS GREATER THAN UPSTREAM STATIC PRESSURE	AGA3_ Orifice_ V3
75	WARNING: ORIFICE DIAMETER WAS <= 0.45 INCHES	All
76	WARNING: PIPE DIAMETER WAS <= 2.0 INCHES	All
79	WARNING: BETA RATIO (DO/DM) WAS <= 0.1 OR >= 0.75	All
80	WARNING: IF GOF2015_OPTION = 1, (HW)/(27.7072*(PF)) = OR > 0.25; IF GOF2015_OPTION = 0, (HW)/(27.707*(PF)) > 0.2	AGA3_ Orifice_ V3

AGA3_Orifice_LIQ

Description

This AGA3_Liquid Orifice function block calculates the volumetric flow-rate of Liquid for an orifice meter using flange or pipe tap based on the input parameters defined below. It is used along with any one of the API 11.1 function blocks since it requires densities @flowing, standard and base conditions. This standard is based on AGA 3 (2012) update and this block is available from R161.2 release.

Figure 3-2: AGA3_Orifice_LIQ function block



Input

Input Parameter	Data types	Description
UnitSystem	INT	{1} for US unit system and {2} for Metric unit system
FlowingTemp	LREAL	It is in FAHRENHEIT for US unit and Celcius for Metric unit.
FlowingPressure	LREAL	It is in PSIA for US unit and Kpa for Metric unit.
DifferentialPressure	LREAL	It is in INH2O for US unit and in Kpa for Metric unit.
GasDensityAtFlowCond	LREAL	It is Gas Density at flowing temperature and pressure. It is in KG/M^3 for (Metric System) & LBM/FT^3 for (US System). It is an output of AGA8.
GasDensityAtStdCond	LREAL	It is Gas Density at standard temperature and pressure. It is in KG/M^3 for (Metric System) & LBM/FT^3 for (US System).It is an output of AGA8.
GasDensityAtBaseCond	LREAL	It is Gas Density at base temperature and pressure. It is in KG/M^3 for (Metric System) & LBM/FT^3 for (US System).It is an output of AGA8.
GasSuperComp	LREAL	It is unit less number. It is an output of AGA8
GasRelDenAtStdCond	LREAL	It is unit less number. It is an output of AGA8
TapsType	INT	FLANGE=1 and PIPE=2
OrificeMaterial	INT	STAINLESS STEEL=1, MONEL=2, CARBON STEEL=3, STAINLESS_S_304=4 and STAINLESS_S_316=5
PipeMaterial	INT	STAINLESS STEEL=1, MONEL=2, CARBON STEEL=3, STAINLESS_S_304=4 and STAINLESS_S_316=5
FluidType	INT	COMPRESSIBLE FLUID =1 and NON-COMPRESSIBLE FLUID=2
TapsLocation	INT	UPSTREAM=1 and DOWNSTREAM=2
OrificeDiameter	LREAL	It is in inches for US unit system & in milimeter for

Input Parameter	Data types	Description
		Metric unit system.
OrfDiaMsrdTemp	LREAL	It is in FAHRENHEIT for US unit and Celcius for Metric unit.
PipeDiameter	LREAL	It is in inches for US unit system & in milimeter for Metric unit system.
PipeDiaMsrdTemp	LREAL	It is in FAHRENHEIT for US unit and Celcius for Metric unit.
AbsViscosity	LREAL	AGA3 Orifice method expects absolute viscosity in CENTIPOISE unit only for both US & Metric unit system. ABSOLUTE VISCOSITY OF FLUID FLOWING. (RECOMMENDED DEFAULT=0.010268 cP - PG 34 PART 4)
IsenExponent	LREAL	ISENTROPIC EXPONENT is unit less number. (RECOMMENDED DEFAULT=1.3 - PG 34 PART 4)
CalibFactor	LREAL	It is unit less number. Default value is 1.0
AirCompFactAtStdCond	LREAL	It is unit less number. COMPRESSIBILITY FACTOR OF AIR AT standard temperature and pressure. It is used for Pipe tap only.
AtmosphericPressure	LREAL	It is in PSIA for US unit and Kpa for Metric unit. If flowing pressure is already absolute, it can be left zero.

Output

Output Parameter	Data types	Description
Aga3_QV	LREAL	This is volume flow rate at standard (TS & PS) conditions. It is in Scf/Hr for US unit system and in cubic meter per hour in Metric unit system.
Aga3_QM	LREAL	This is mass flow rate. It is in Lbm/Hr for US unit system and Kg/Hr for Metric unit system.

Output Parameter	Data types	Description
Aga3_QB	LREAL	This is volumetric flow rate at base (TB & PB) conditions. It is in Scf/Hr for US unit system and in cubic meter per hour in Metric unit system.
CD	LREAL	It is Orifice plate coefficient of discharge (flange).
Y	LREAL	It is Expansion factor (flange and pipe).
EV	LREAL	It is Velocity of approach factor (flange).
Out_Code	INT	This out parameter returns success or fail code.

- AGA3_Orifice_LIQ expects the input parameters to be either in US unit system or Metric unit system. Output parameters from API 11.1 function blocks become input parameters for this function block.
- The output parameters would be in the same unit as of inputs.

Following is the table that describes different out code for AGA3_ Orifice_LIQ function blocks.

Out Code	Description
51	ERROR: NTAPS WAS NOT 0, 1 OR 2
52	ERROR: FLOWING PRESSURE WAS <= 0.0 OR > 40000. PSIA
53	ERROR: FLOWING TEMPERATURE < -200. OR > 760. DEG F
54	ERROR: MATORF OR MATPIPE WAS NOT 0, 1, 2 OR 3
55	ERROR: ORIFICE DIAMETER WAS <= 0 OR => 100.0 INCHES
56	ERROR: PIPE DIAMETER WAS <= 0 OR => 100.0 INCHES
57	ERROR: FLOWING OR STANDARD DENSITY WAS <= 0.0 LBM/FT^3
58	ERROR: DIFFERENTIAL PRESSURE WAS <= 0.0 INCHES H2O
60	ERROR: ISENTROPIC EXPONENT <= 1.0 OR => 2.0
61	ERROR: IFLUID WAS NOT 0, 1 OR 2
62	ERROR: STANDARD TEMPERATURE WAS NOT = 60.0 DEG F
63	ERROR: STANDARD PRESSURE WAS NOT = 14.73 PSIA
64	ERROR: TAP LOCATION WAS NOT 0, 1 OR 2 FOR NTAPS=2 (PIPE) OR TAP LOCATION WAS NOT 1 FOR NTAPS=1 (FLANGE)

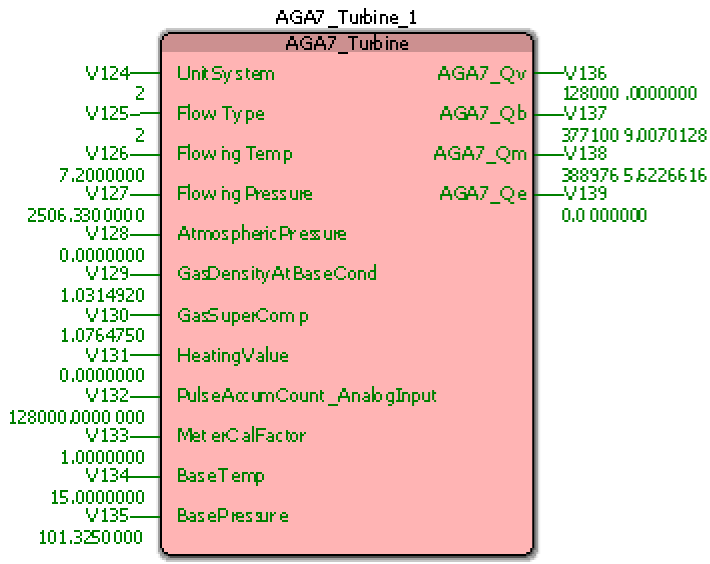
Out Code	Description
65	ERROR: SUPERCOMPRESSIBILITY FACTOR WAS <= 0.0
66	ERROR: RELATIVE DENSITY AT STANDARD CONDITIONS WAS < 0.07 OR > 1.52
67	ERROR: CALIBRATION FACTOR WAS <= 0.0
68	ERROR: COMPRESSIBILITY FACTOR AT STANDARD CONDITIONS <= 0.0
69	ERROR: BETA RATIO (DO/DM) <= 0.0 OR => 1.0
70	ERROR: IF NTAPS = 1, GOF2015_OPTION NOT = 1 OR = 0
71	ERROR: IF NTAPS = 2, GOF2015_OPTION NOT = 0
72	ERROR: DIFFERENTIAL PRESSURE WAS GREATER THAN UPSTREAM STATIC PRESSURE
80	WARNING: IF GOF2015_OPTION = 1, (HW)/(27.7072*(PF)) = OR > 0.25; IF GOF2015_OPTION = 0, (HW)/(27.707*(PF)) > 0.2

AGA7_Turbine and AGA9_Ultrasonic

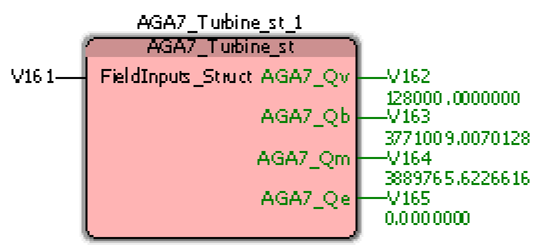
Description

AGA7_Turbine or AGA9_Ultrasonic function block corrects measured volume at flowing conditions read by either turbine or ultrasonic meter to volume at base conditions; based on the input parameters defined below. This description is applicable to following function blocks.

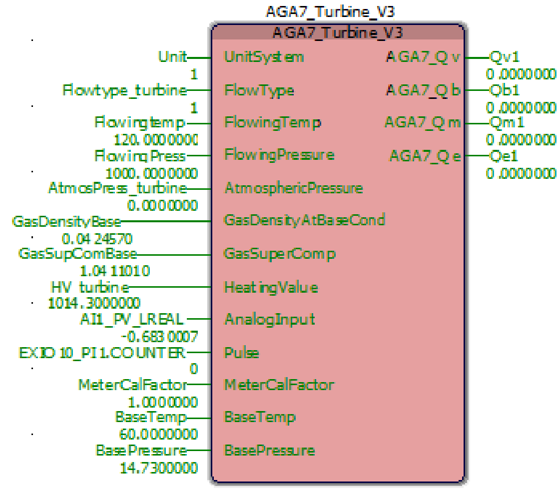
- AGA7_Turbine



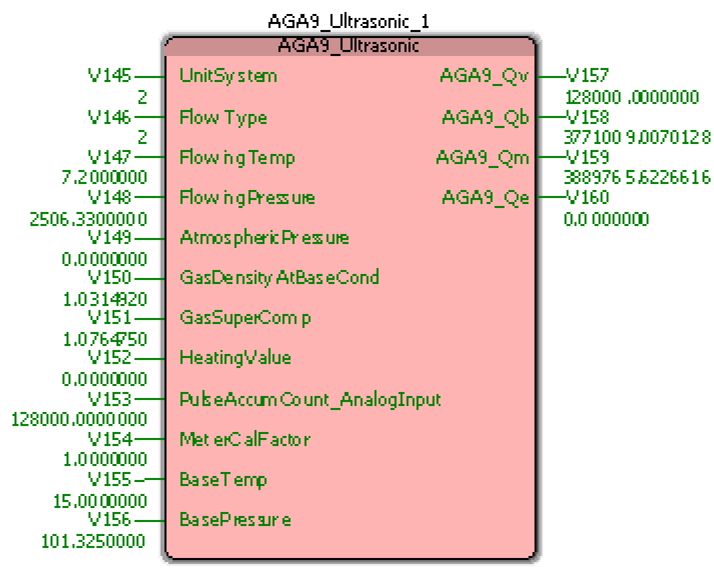
- AGA7_Turbine_st (This function block is having input parameters in the form of structures to make function block organized and compact)



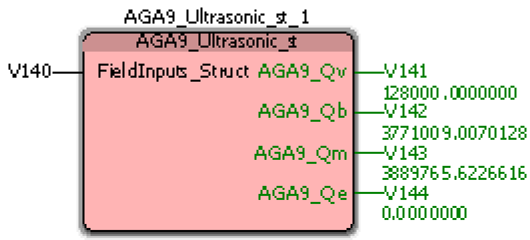
- AGA7_Turbine_V3



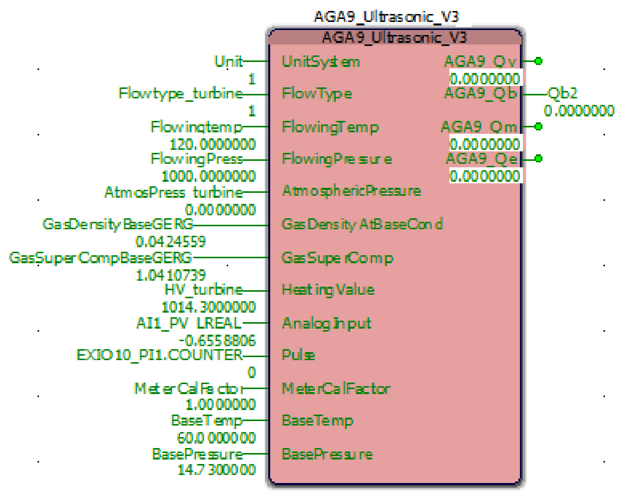
- AGA9_Ultrasonic



- AGA9_Ultrasonic_st (This function block is having input parameters in the form of structures to make function block organized and compact)



- AGA9_Ultrasonic_V3



TIP: AGA7_Turbine and AGA9_Ultrasonic both take unit system as one of their input parameters. The unit systems supported by these function blocks are US and Metric. These two function blocks are similar in all respect.

Input

Input Parameter	Data types	Description	Apply to
UnitSystem	INT	{1} US unit system and {2} Metric unit system	All
FlowType	INT	{1} Pulse Accumulated or {2} Analog Flow Rate	All
FlowingTemp	LREAL	If the unit system is US then it should be in FAHRENHEIT and if the unit system is Metric then it	All

Input Parameter	Data types	Description	Apply to
		should be in Celsius	
FlowingPressure	LREAL	If the unit system is US then it should be in PSIA and if the unit system is Metric then it should be in KPA.	All
AtmosphericPressure	LREAL	If the unit system is US then it should be in PSIA and if the unit system is Metric then it should be in KAR It is added in Flowing pressure to make it absolute pressure. If flowing pressure is already absolute then it can be left zero.	All
GasDensityAtBaseCond	LREAL	Gas density @ base condition should be in LBM/FT^3 (for US Unit System). For Metric unit system it should be in KG/M^3. It is an output parameter of AGA8function block.	All
GasSuperComp	LREAL	It is unit less number. It is gas super compressibility calculated by AGA8 function block. It is an output parameter of AGA8 function block.	All
HeatingValue	LREAL	For US unit system it should be in Btu/ft^3 and for Metric unit system it should be in MJ/m^3. Gas Heating value (usually from Gas Chromatograph or simply set as a constant). It is required when delta energy AGA7_Qe or AGA9_Qe needs to be determined as one of the outputs by the Function Block else it can be left/ set zero	All
PulseAccumCount_	LREAL	For Pulse Accumulated flow type	AGA7_

Input Parameter	Data types	Description	Apply to
AnalogInput		meter, it would be a high speed input pulse counter. And for Analog flow type meter, it would be an analog input value.	Turbine & AGA9_ Ultrasonic AGA7_ Turbine_V2 & AGA9_ Ultrasonic_ V2
AnalogInput	LREAL	Value of analog input if flow type is Analog. The value should be in lb^3/hr for US unit or m^3/hr for Metric unit.	AGA7_ Turbine_V3 & AGA9_ Ultrasonic_ V3
Pulse	UDINT	Pulse counter value	AGA7_ Turbine_V3 & AGA9_ Ultrasonic_ V3
MeterCalFactor	LREAL	This input parameter is a Meter Calibration factor which is a unit less value.	All
BaseTemp	LREAL	If the unit system is US then it must be in Deg Fahrenheit. If the unit system is Metric then it must be in Deg Celsius. The recommended default is 60 Deg F.	All
BasePressure	LREAL	If the unit system is US then it must be in PSIA. If the unit system is Metric then it must be in KPA. The recommended default is 14.73 PSIA.	All

Output

Following table describes output parameters for AGA7_Turbine function block.

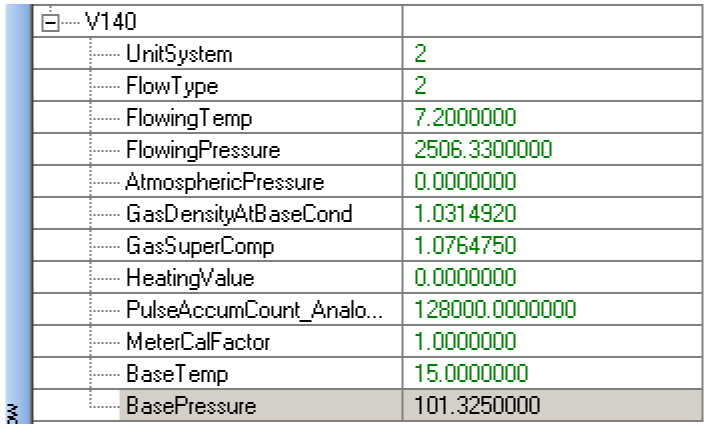
Output Parameter	Data types	Description
AGA7_Qv	LREAL	This is uncorrected volume @ Flowing conditions TF and PF. It's US unit is SCF/HR and Metric unit is M^3 /HR
AGA7_Qb	LREAL	This is Corrected volume at base conditions using compressibility from AGA8. It's US unit is SCF/HR and Metric unit is M^3 /HR
AGA7_Qm	LREAL	This is mass using base density (RHOB) from AGA8. Its US unit is LBM/HR and Metric unit is KG/HR.
AGA7_Qe	LREAL	This is energy flow using heating value. It's US unit is BTU/Hr (British thermal units) and Metric unit is GJ/hr (gigajoules per hour)

Following table describes output parameters for AGA9_Ultrasonic function block.

Output Parameter	Data types	Description
AGA9_Qv	LREAL	This is uncorrected volume @ Flowing conditions TF and PF. It's US unit is SCF/HR and Metric unit is M^3 /HR
AGA9_Qb	LREAL	This is Corrected volume at base conditions using compressibility from AGA8. It's US unit is SCF/HR and Metric unit is M^3 /HR
AGA9_Qm	LREAL	This is mass using base density (RHOB) from AGA8. Its US unit is LBM/HR and Metric unit is KG/HR.
AGA9_Qe	LREAL	This is energy flow using heating value. Its US unit is BTU/hr (British thermal units).

For AGA7_Turbine_st function block input structure is user defined data type. This is defined in Task_Info under Data Types in IEC Programming Workspace.

Input Parameter	Data types
FieldInputs_Struct	AGA7_9_Inputs_STRUCT



V140	
UnitSystem	2
FlowType	2
FlowingTemp	7.2000000
FlowingPressure	2506.3300000
AtmosphericPressure	0.0000000
GasDensityAtBaseCond	1.0314920
GasSuperComp	1.0764750
HeatingValue	0.0000000
PulseAccumCount_Analo...	128000.0000000
MeterCalFactor	1.0000000
BaseTemp	15.0000000
BasePressure	101.3250000

For AGA9_Ultrasonic_st function block input structure is user defined data type. This is defined in Aga_Data Types under Data Types in the IEC Programming Workspace.

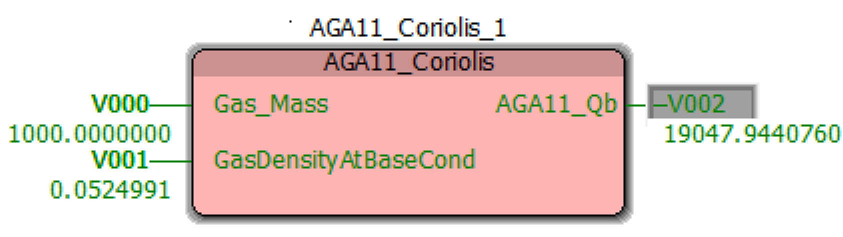
Input Parameter	Data types
FieldInputs_Struct	AGA7_9_Inputs_STRUCT

AGA11_Coriolis

Description

AGA11_Coriolis function block converts gas mass (absolute) to volume at base condition; based on the input parameters defined below. This description is applicable to following function block:

- AGA11_Coriolis



TIP: AGA11_Coriolis function block takes absolute gas mass measured by Coriolis meter and gas density at base conditions generally come out of AGA8 function block. If the gas mass is in US unit then base density from AGA8 should be in US unit. If gas mass is in Metric then base density should be in Metric unit system. Output volume at base condition would be in same unit as of inputs. For US unit output would be in SCF/Hr and for Metric it would be in M^3/Hr.

Input

Input Parameter	Data types	Description
Gas_Mass	LREAL	Gas mass should be in US unit or Metric unit. Gas Mass directly comes from Coriolis Meter.
GasDensityAtBaseCond	LREAL	Gas density @ base condition should be in LBM/FT^3 (for US Unit System). For Metric unit system it should be in KG/M^3.

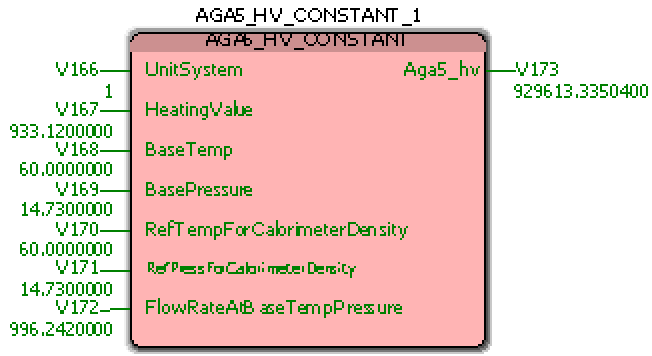
Output

Output Parameter	Data types	Description
AGA11_Qb	LREAL	This is volume @ base conditions. It's US unit is SCF/HR and Metric unit is M^3 /HR

AGA5_HV_CONSTANT

Description

AGA5_HV_CONSTANT function block calculates gas flow energy when we have gas heating value and volume at base condition. This function block can be used when gas heating value is directly available.



Input

Input Parameter	Data types	Description
Unit System	INT	It is unit less number. For US unit it is 1 and for Metric it is 2
Heating Value	LREAL	If using US {1} unit then heating value should be in US unit. If using Metric {2} unit then its value should be in Metric unit. Its US unit is BTU/FT^3. It's Metric unit is MJ/m^3
BaseTemp	LREAL	If the unit system is US then it must be in Deg Fahrenheit. If the unit system is Metric then it must be in Deg Celsius. The recommended default is 60 Deg F
BasePressure	LREAL	If the unit system is US then it must be in PSIA. If the unit system is Metric then it must be in KPA. The recommended default is 14.73 PSIA
RefTempForCalorimeterDensity	LREAL	It is the reference temperature for Calorimeter Density; If the unit system is US then it must be in Deg Fahrenheit. If the unit system is Metric then it must be in Deg Celsius. Recommended default value is 60.0 DEG

Input Parameter	Data types	Description
		F.
RefPressForCalorimeterDensity	LREAL	It is the reference pressure for Calorimeter Density; If the unit system is US then it must be in PSIA. If the unit system is Metric then it must be in BAR. Recommended default value is 14.73 PSIA.
Flow Rate @ Base Condition	LREAL	If using US {1} unit then Flow Rate value should be in US unit. If using Metric {2} unit then its value should be in Metric unit. It's US unit is FT^3/Hr. It's Metric unit is M^3/Hr.

Output

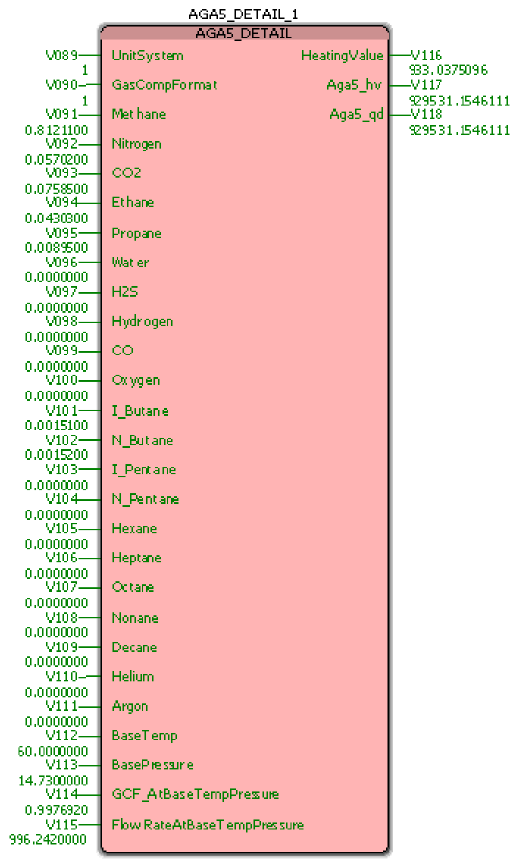
Output Parameter	Data types	Description
Aga5_hv	LREAL	This is gas flow energy at base condition. It’s US unit is BTU/Hr.

AGA5_DETAIL

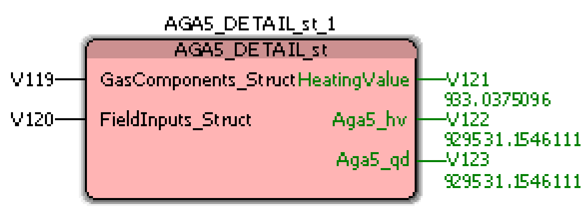
Description

AGA5_DETAIL function block calculates gas flow energy when all 21 gas elements are available. This description is applicable to following function block:

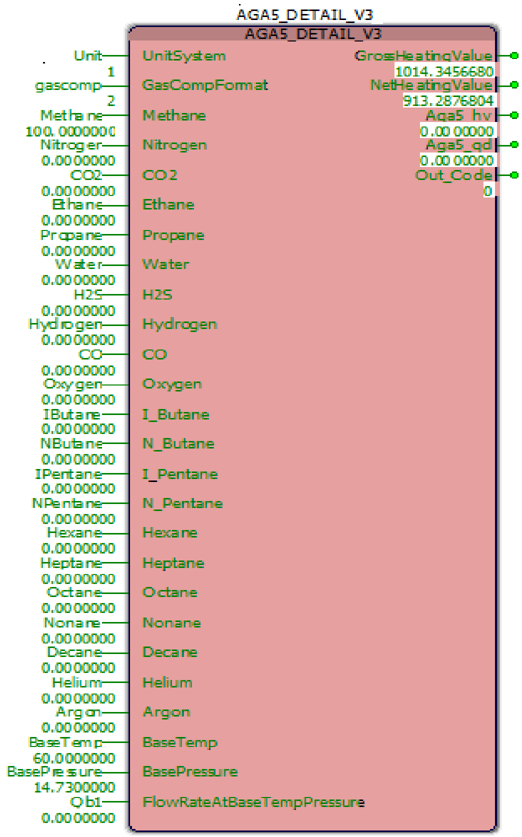
- AGA5_DETAIL



- AGA5_DETAIL_st



- AGA5_DETAIL_V3



This function block can be used to calculate gas energy flow or gas heating value when all 21 gas composition elements are available.

Input

Input Parameter	Data types	Description	Apply to
Unit System	INT	For US unit it is 1 and for Metric it is 2	All
GasCompFormat	INT	Gas Composition Format - {1} for Mole Fraction and {2} for Percentage **NOTE:** It is recommended to use 2 percentage as a	All

Input Parameter	Data types	Description	Apply to
		default option.	
Methane	LREAL	It can be in Mole Fraction or Percentage	All
Nitrogen	LREAL	It can be in Mole Fraction or Percentage	All
CO2	LREAL	It can be in Mole Fraction or Percentage	All
Ethane	LREAL	It can be in Mole Fraction or Percentage	All
Propane	LREAL	It can be in Mole Fraction or Percentage	All
Water	LREAL	It can be in Mole Fraction or Percentage	All
H2S	LREAL	It can be in Mole Fraction or Percentage	All
Hydrogen	LREAL	It can be in Mole Fraction or Percentage	All
CO	LREAL	It can be in Mole Fraction or Percentage	All
Oxygen	LREAL	It can be in Mole Fraction or Percentage	All
I_Butane	LREAL	It can be in Mole Fraction or Percentage	All
N_Butane	LREAL	It can be in Mole Fraction or Percentage	All
I_Pentane	LREAL	It can be in Mole Fraction or Percentage	All
N_Pentane	LREAL	It can be in Mole Fraction or Percentage	All
Hexane	LREAL	It can be in Mole Fraction or Percentage	All

Input Parameter	Data types	Description	Apply to
Heptane	LREAL	It can be in Mole Fraction or Percentage	All
Octane	LREAL	It can be in Mole Fraction or Percentage	All
Nonane	LREAL	It can be in Mole Fraction or Percentage	All
Decane	LREAL	It can be in Mole Fraction or Percentage	All
Helium	LREAL	It can be in Mole Fraction or Percentage	All
Argon	LREAL	It can be in Mole Fraction or Percentage	All
BaseTemp	LREAL	If the unit system is US then it must be in Deg Fahrenheit. If the unit system is Metric then it must be in Deg Celsius. The recommended default is 60 Deg F.	All
BasePressure	LREAL	If the unit system is US then it must be in PSIA. If the unit system is Metric then it must be in KPA. The recommended default is 14.73 PSIA.	All
GCF_AtBaseTempPressure	LREAL	It is unit less number. It is the output from AGA8 function block.	AGA5_Detail & AGA5_Detail_V2
FlowRateAtBaseTempPressure	LREAL	If using US {1} unit then Flow Rate value should be in US unit. If using Metric {2} unit then it's value should be in Metric unit. It's US unit is FT^3/Hr	All

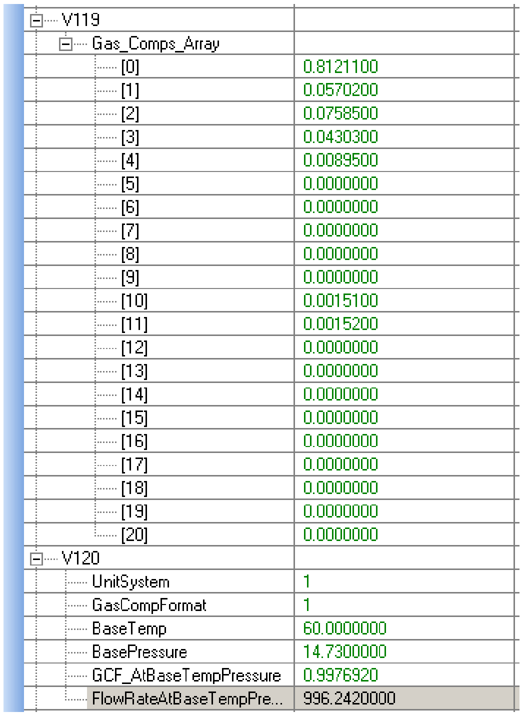
Input Parameter	Data types	Description	Apply to
		It's Metric unit is M^3/Hr.	

Output

Output Parameter	Data types	Description	Apply to
HeatingValue	LREAL	This is gas heating value in BTU/FT^3 for US unit.	AGA5_Detail & AGA5_DetaiL_V2
GrossHeatingValue	LREAL	This is gross heating value in BTU/FT^3 for US unit. MJ/M3 for Metric.	AGA5_DetaiL_V3
NetHeatingValue	LREAL	This is net heating value in BTU/FT^3 for US unit. MJ/M3 for Metric.	AGA5_DetaiL_V3
Aga5_hv	LREAL	This is gas flow energy at base condition. It's US unit is BTU/Hr.	All
Aga5_qd	LREAL	This is dry gas flow energy at base condition. It's US unit is BTU/Hr.	All
Outcode	INT	This out parameter returns success or fail code.	AGA5_DetaiL_V3

For AGA5_DETAIL_st function block input structures are user defined data type. They are defined in Aga_Data Types under Data Types in IEC Programming Workspace.

Input Parameter	Data types
GasComponents_Struct	DtlMtd_GasComps_STRUCT
FieldInputs_Struct	AGA5_DtlMtd_FldInputs_STRUCT



Following are the error codes for AGA5_Detail_V3 function block:

Out Code	Description	Apply to
1	ERROR: A COMPONENT MOLE FRACTION < 0.0 OR > 1.0	AGA5_DETAIL_V3
2	WARNING: SUM OF MOLE FRACTIONS < 0.9999 OR > 1.0001	AGA5_DETAIL_V3
3	WARNING: PRESSURE BASE (PB) <= 0.0 OR >= 16 PSIA	AGA5_DETAIL_V3
4	WARNING: TEMPERATURE BASE (TB) <= 32.0 OR >= 77.0 DEG F	AGA5_DETAIL_V3

CHAPTER

4 API 11.1

The following API 11.1 function blocks are available:

Function Block	Description
CRUDE_OIL_ALT_US	Calculation for Crude Oil with Alternet conditions- US Units
REFINED_PRODUCTS_ALT_US	Calculation for Refined Products with Alternet conditions- US Units
SPECIAL_PRODUCTS_ALT_US	Calculation for Special Products with Alternet conditions- US Units
LUBRICATING_OIL_ALT_US	Calculation for Lubricating Oil with Alternet conditions- US Units
CRUDE_OIL_OBS_US	Calculation for Crude Oil with Observed conditions- US Units
REFINED_PRODUCTS_OBS_US	Calculation for Refined Products with Observed conditions- US Units
SPECIAL_PRODUCTS_OBS_US	Calculation for Special Products with Observed conditions- US Units
LUBRICATING_OIL_OBS_US	Calculation for Lubricating Oil with Observed conditions- US Units
CRUDE_OIL_AO_US	Calculation for Crude Oil with Alternet & Observed conditions- US Units
REFINED_PRODUCTS_AO_US	Calculation for Refined Products with Alternet & Observed conditions- US Units
SPECIAL_PRODUCTS_AO_US	Calculation for Special Products with Alternet & Observed conditions- US Units
LUBRICATING_OIL_AO_US	Calculation for Lubricating Oil with Alternet & Observed conditions- US Units
CRUDE_OIL_ALT_ME	Calculation for Crude Oil with Alternet conditions- Metric Units
REFINED_PRODUCTS_ALT_ME	Calculation for Refined Products with Alternet conditions- Metric Units
SPECIAL_PRODUCTS_ALT_ME	Calculation for Special Products with Alternet conditions- Metric Units

Function Block	Description
LUBRICATING_OIL_ALT_ME	Calculation for Lubricating Oil with Alternet conditions- Metric Units
CRUDE_OIL_OBS_ME	Calculation for Crude Oil with Observed conditions- Metric Units
REFINED_PRODUCTS_OBS_ME	Calculation for Refined Products with Observed conditions- Metric Units
SPECIAL_PRODUCTS_OBS_ME	Calculation for Special Products with Observed conditions- Metric Units
LUBRICATING_OIL_OBS_ME	Calculation for Lubricating Oil with Observed conditions- Metric Units
CRUDE_OIL_AO_ME	Calculation for Crude Oil with Alternet & Observed conditions-Metric Units
REFINED_PRODUCTS_AO_ME	Calculation for Refined Products with Alternet & Observed conditions-Metric Units
SPECIAL_PRODUCTS_AO_ME	Calculation for Special Products with Alternet & Observed conditions-Metric Units
LUBRICATING_OIL_AO_ME	Calculation for Lubricating Oil with Alternet & Observed conditions-Metric Units

Supported commodities

- Crude oil
- Refined products
- Special products
- Lubricating oil

Supported Unit systems

- US
- Metric

Supported calculations

- API TYPE1- US unit system
- API TYPE2- US unit system
- API TYPE3- US unit system
- API TYPE4- Metric unit system
- API TYPE5- Metric unit system
- API TYPE6- Metric unit system

Output Error Codes

The following is table that describes different output error code generated by API function blocks.

Parameter	Description
0	No error, Calculations Successful
1	Error - Illegal arguments
2	Error - Memory allocation
3	Error - VCF out of range
4	Error - Non convergence
5	Error - Temperature out of range
6	Error - Density out of range
7	Error - Pressure out of range
8	Error - Alpha60 out of range
9	Error - Supercritical fluid
10	Error - No reference fluids
11	Error - No Solution

API TYPE1 Function Blocks

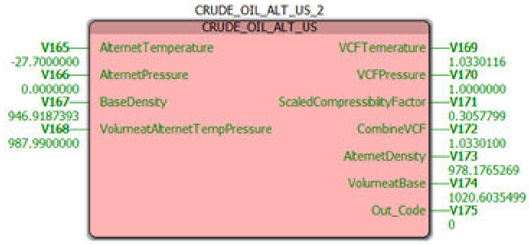
Description

These function blocks calculates the Volume Correction Factor (VCF) for correcting from the density at the base conditions (60°F and 0 psig) to alternate temperature and pressure conditions for crude-oil, refined products, special products and lubricating oil.

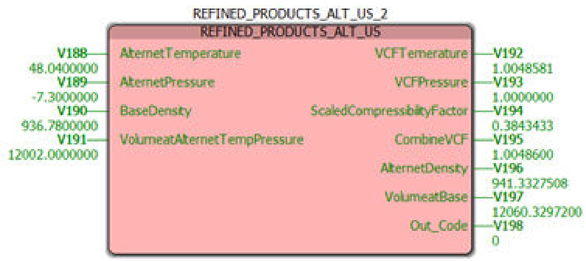
These function blocks are specific to US unit system.

API TYPE1 Function Blocks include the following function blocks:

- CRUDE_OIL_ALT_US

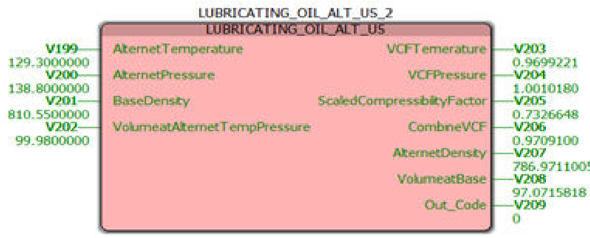


- REFINED_PRODUCTS_ALT_US



- SPECIAL_PRODUCTS_ALT_US

- LUBRICATING_OIL_ALT_US



These Function block expects the input values in the proper units (°F, psig, and kg/m³). If they are not in the proper units then appropriate unit conversion block should be used. The density values calculated by these function block are in the units of kg/m³. If these units do not match the original input units, then the output densities should be converted to that of the original input value's units appropriate unit conversion block.

Input

Input Parameter	Data types	Description
AlternateTemperature	LREAL	Value of alternate temperature in °F
AlternatePressure	LREAL	Value of alternate Pressure in kpa
BaseDensity	LREAL	Value of base density (kg/m3). If input density type is relative density or API Gravity then it must be converted into Density(kg/m3) using provided unit conversion blocks
VolumeatAlternetTempPressure	LREAL	No Conversion needed, Most of the time it is optional input

Output

Output Parameter	Data types	Description
VCFTemerature	LREAL	Volume correction factor due to temperature
VCFPressure	LREAL	Volume correction factor due to pressure
ScaledCompressibilityFactor	LREAL	Scaled compressibility factor

Output Parameter	Data types	Description
CombineVCF	LREAL	Combined volume correction factor due to temperature and pressure
AlternetDensity	LREAL	Density at alternate conditions
VolumeatBase	LREAL	Volume at base conditions
Out_Code	INT	This out parameter returns success or fail code.

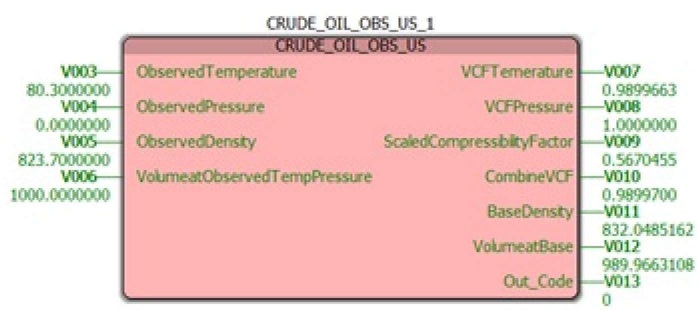
API TYPE2 Function Blocks

Description

These function blocks calculates the density at the base conditions (60°F and 0 psig) that is consistent with an observed density at its temperature and pressure condition. It has the flexibility of accepting a pre-calculated 60°F thermal expansion factor as per the commodity type of the liquid that is crude-oil, refined products, special products and lubricating oil. These function blocks are specific to US unit system.

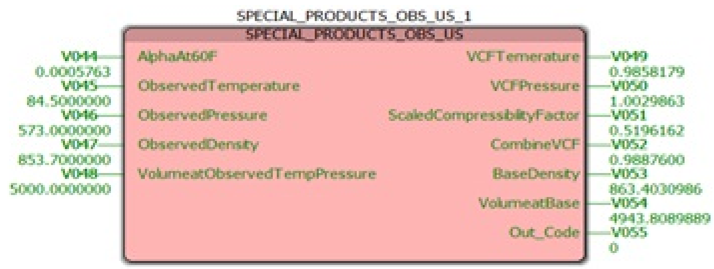
This description is applicable to following function blocks:

- CRUDE_OIL_OBS_US

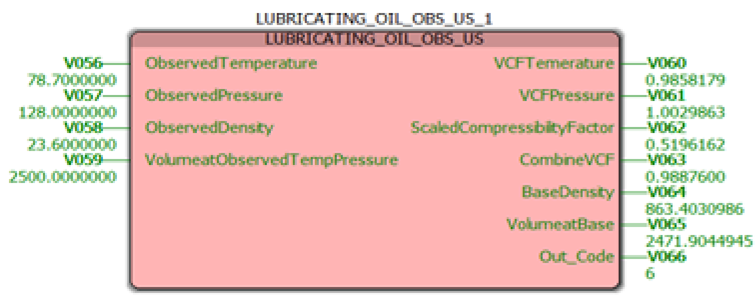


- REFINED_PRODUCTS_OBS_US

- SPECIAL_PRODUCTS_OBS_US



- LUBRICATING_OIL_OBS_US



These Function block expects the input values in the proper units (°F, Kpa, and kg/m³). If they are not in the proper units then appropriate unit conversion block should be used. The density values calculated by these function block are in the units of kg/m³. If these units do not match the original input units, then the output densities should be converted to that of the original input value's units appropriate unit conversion block.

Input

Input Parameter	Data types	Description
ObservedTemperature	LREAL	Value of observed temperature in °F
ObservedPressure	LREAL	Value of observed Pressure in kpa
ObservedDensity	LREAL	Value of observed density (kg/m3). If input density type is relative density or API Gravity then it must be converted into Density(kg/m3) using provided unit conversion blocks
AlphaAt60F	LREAL	Pre-calculated 60°F thermal expansion factor. This input parameter is only applicable for special products. For other commodity types this parameter is not present
VolumeatObservedTempPressure	LREAL	No Conversion needed, Most of the time it is optional input

Ouput

Output Parameter	Data types	Description
VCFTemerature	LREAL	Volume correction factor due to temperature
VCFPressure	LREAL	Volume correction factor due to pressure
ScaledCompressibilityFactor	LREAL	Scaled compressibility factor
CombineVCF	LREAL	Combined volume correction factor due to temperature and pressure
BaseDensity	LREAL	Density at Base conditions
VolumeatBase	LREAL	Volume at base conditions
Out_Code	INT	This out parameter returns success or fail code.

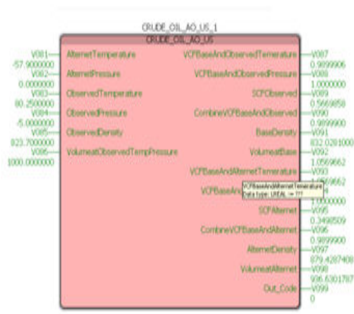
API TYPE3 Function Blocks

Description

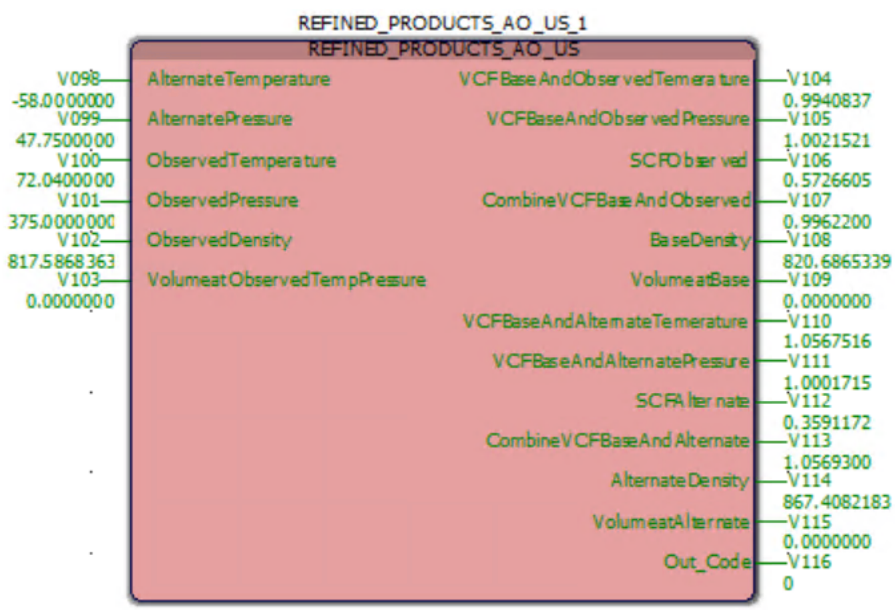
These function blocks combines those in TYPE1 and TYPE2. First, the density at the base conditions (60°F and 0 psig) consistent with an observed density is calculated. This base density is then corrected to the alternate temperature and pressure conditions as per commodity type of the liquid that is crude-oil, refined products, special products and lubricating oil. These function blocks are specific to US unit system.

This description is applicable to following function blocks:

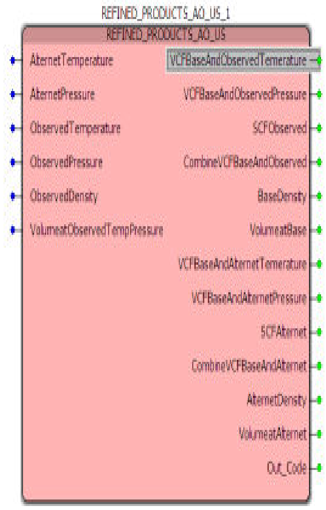
- CRUDE_OIL_AO_US



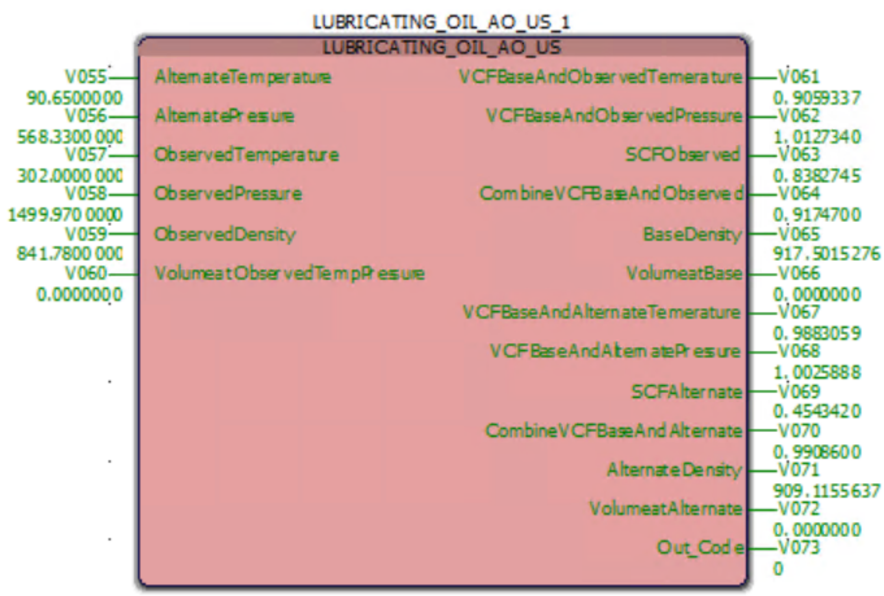
- REFINED_PRODUCTS_AO_US



- SPECIAL_PRODUCTS_AO_US



- LUBRICATING_OIL_AO_US



These Function block expects the input values in the proper units (°F, Kpa, and kg/m³). If they are not in the proper units then appropriate unit conversion block should be used. The density values calculated by these function block are in the units of kg/m³. If these units do not match the original input units, then the output densities should be converted to that of the original input value's units appropriate unit conversion block.

Input

Input Parameter	Data types	Description
AlternetTemperature	LREAL	Value of alternate temperature in °F
AlternetPressure	LREAL	Value of alternate Pressure in kpa
ObservedTemperature	LREAL	Value of observed temperature in °F
ObservedPressure	LREAL	Value of observed Pressure in kpa
ObservedDensity	LREAL	Value of observed density (kg/m3). If input density type is relative density or API Gravity then it must be converted into Density(kg/m3) using provided unit conversion blocks
AlphaAt60F	LREAL	Pre-calculated 60°F thermal expansion factor. This input parameter is only applicable for special products. For other commodity types this parameter is not present
VolumeatObservedTempPressure	LREAL	No Conversion needed, Most of the time it is optional input

Output

Output Parameter	Data types	Description
VCFBaseAndObservedTemerature	LREAL	Volume correction factor due to temperature between the base and observed temperatures
VCFBaseAndObservedPressure	LREAL	Volume correction factor due to pressure between the base and observed pressures at the observed temperature
SCFObserved	LREAL	Scaled compressibility factor at the observed temperature
CombineVCFBaseAndObserved	LREAL	Combined volume correction factor due to temperature and pressure between

Output Parameter	Data types	Description
		the base and observed conditions
BaseDensity	LREAL	Density at Base conditions
VolumeatBase	LREAL	Volume at base conditions
VCFBaseAndAlternetTemerature	LREAL	Volume correction factor due to temperature between the base and alternate temperatures
VCFBaseAndAlternetPressure	LREAL	Volume correction factor due to pressure between the base and alternate pressures at the alternate temperature
SCFAlternet	LREAL	Scaled compressibility factor at the alternate temperature
CombineVCFBaseAndAlternet	LREAL	Combined volume correction factor due to temperature and pressure between the base and alternate conditions
AlternetDensity	LREAL	Density at alternate conditions
VolumeatAlternet	LREAL	Volume at alternate conditions
Out_Code	INT	This out parameter returns success or fail code.

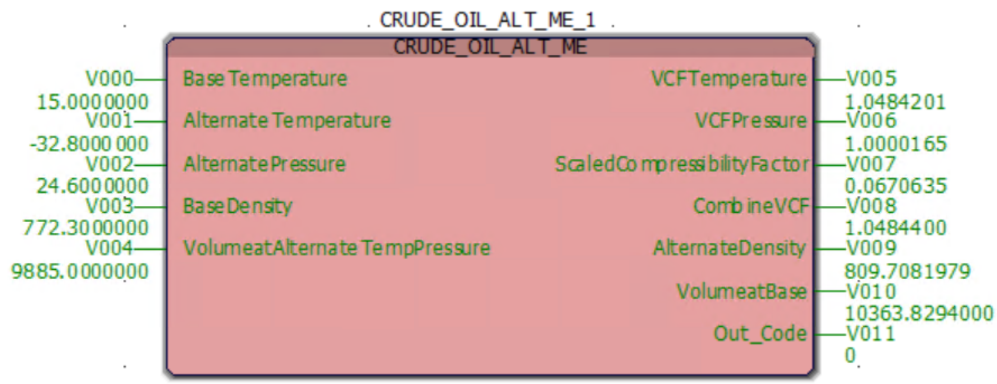
API TYPE4 Function Blocks

Description

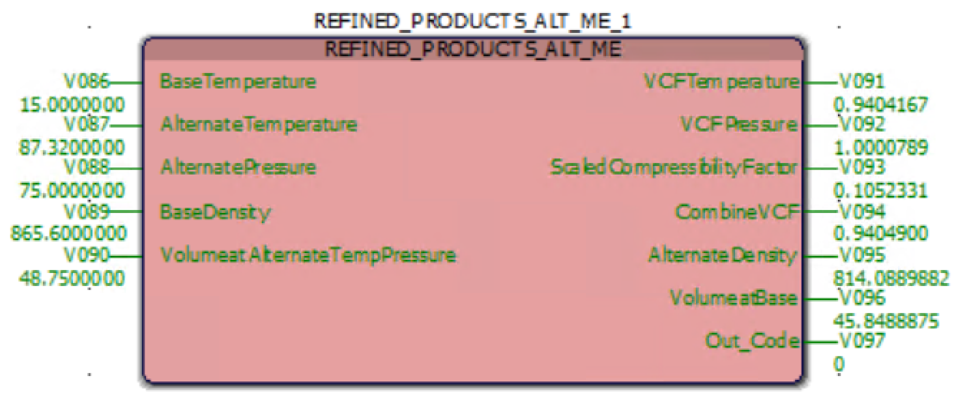
This procedure calculates the Volume Correction Factor (VCF) given the density at the metric base conditions (15°C or 20°C and 0 kPa (gauge)). The parameters used in these function blocks depends upon the commodity group to which the liquid belongs that is crude-oil, refined products, special products and lubricating oil. These function blocks are specific to Metric unit system.

This description is applicable to following function blocks:

- CRUDE_OIL_ALT_ME

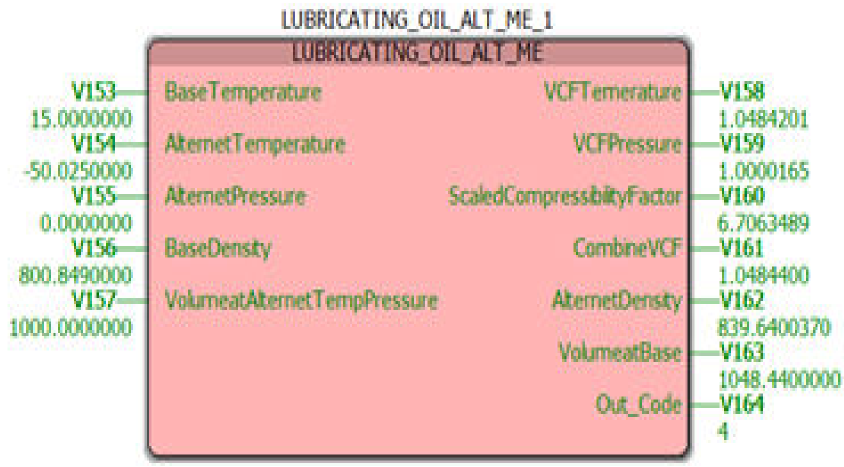


- REFINED_PRODUCTS_ALT_ME



- SPECIAL_PRODUCTS_ALT_ME

- LUBRICATING_OIL_ALT_ME



These Function block expects the input values in the proper units (°C, kPa). If they are not in the proper units then appropriate unit conversion block should be used. The density values calculated by these function block are in the units of kg/m³. If these units do not match the original input units, then the output densities should be converted to that of the original input value's units appropriate unit conversion block.

Input

Input Parameter	Data types	Description
BaseTemperature	LREAL	Value of Base temperature °C
AlternateTemperature	LREAL	Value of alternate temperature in °C
AlternatePressure	LREAL	Value of alternate Pressure in kpa
BaseDensity	LREAL	Value of base density (kg/m3). If input density type is relative density or API Gravity then it must be converted into Density(kg/m3) using provided unit conversion blocks
AlphaAt60F	LREAL	Pre-calculated 60°F thermal expansion factor. This input parameter is only applicable for special products. For other commodity types this parameter is not present

Input Parameter	Data types	Description
VolumeatAlternetTempPressure	LREAL	No Conversion needed, Most of the time it is optional input

Output

Output Parameter	Data types	Description
VCFTemerature	LREAL	Volume correction factor due to temperature
VCFPressure	LREAL	Volume correction factor due to pressure
ScaledCompressibilityFactor	LREAL	Scaled compressibility factor
CombineVCF	LREAL	Combined volume correction factor due to temperature and pressure
AlternetDensity	LREAL	Density at alternate conditions
VolumeatBase	LREAL	Volume at base conditions
Out_Code	INT	This out parameter returns success or fail code.

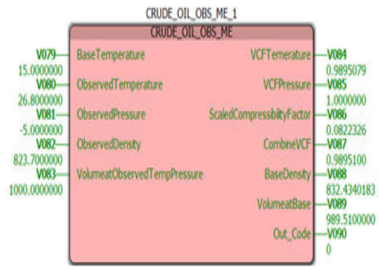
API TYPE5 Function Blocks

Description

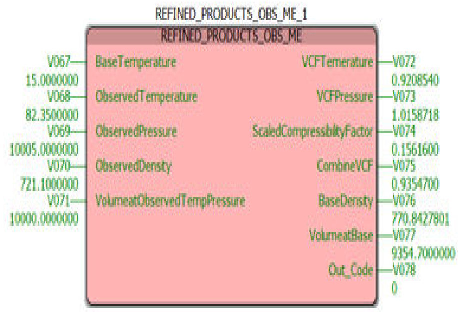
These function blocks calculates the density at the metric base conditions (15°C or 20°C and 0 kPa (gauge)) that is consistent with an observed density measured at the observed temperature and pressure conditions for crude-oil, refined products, special products and lubricating oil. These function blocks are specific to Metric unit system.

This description is applicable to following function blocks

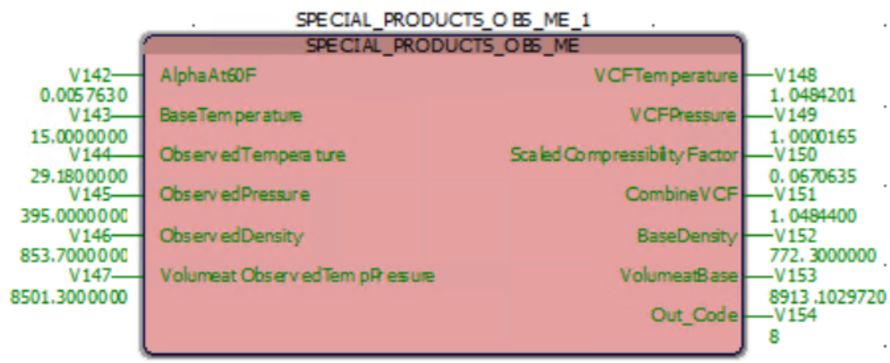
- CRUDE_OIL_OBS_ME



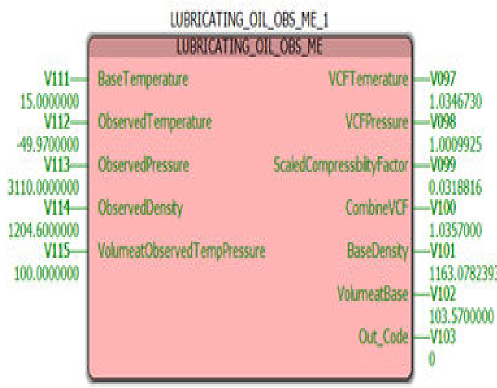
- REFINED_PRODUCTS_OBS_ME



- SPECIAL_PRODUCTS_OBS_ME



- LUBRICATING_OIL_OBS_ME



These Function block expects the input values in the proper units (°C, Kpa, and kg/m³). If they are not in the proper units then appropriate unit conversion block should be used. The density values calculated by these function block are in the units of kg/m³. If these units do not match the original input units, then the output densities should be converted to that of the original input value's units appropriate unit conversion block.

Input

Input Parameter	Data types	Description
BaseTemperature	LREAL	Value of Base temperature °C
ObservedTemperature	LREAL	Value of observed temperature in °C
ObservedPressure	LREAL	Value of observed Pressure in kpa
ObservedDensity	LREAL	Value of observed density (kg/m3). If input density type is relative density or API Gravity then it must be converted into Density(kg/m3) using provided unit conversion blocks
AlphaAt60F	LREAL	Pre-calculated 60°F thermal expansion factor. This input parameter is only applicable for special products. For other commodity types this parameter is not present
VolumeatObservedTempPressure	LREAL	No Conversion needed, Most of the time it is optional input

Output

Output Parameter	Data types	Description
VCFTemerature	LREAL	Volume correction factor due to temperature
VCFPressure	LREAL	Volume correction factor due to pressure
ScaledCompressibilityFactor	LREAL	Scaled compressibility factor
CombineVCF	LREAL	Combined volume correction factor due to temperature and pressure
BaseDensity	LREAL	Density at Base conditions
VolumeatBase	LREAL	Volume at base conditions
Out_Code	INT	This out parameter returns success or fail code.

API TYPE6 Function Blocks

Description

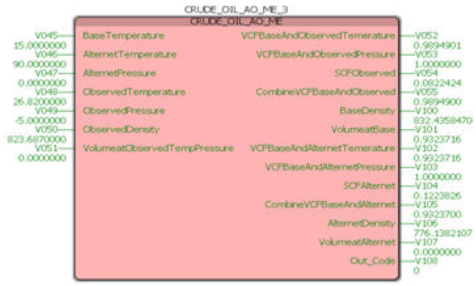
These function blocks combines those in TYPE4 and TYPE5. The density at conditions of 60ºF and 0 psig that is consistent with the observed density is first calculated. This density is then corrected to the alternate temperature and pressure conditions.

The corresponding density at the metric base temperature (15ºC or 20ºC) is also

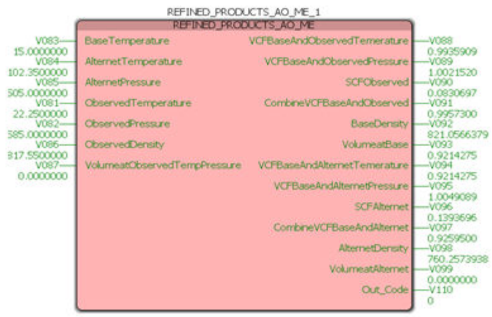
Calculated as per the commodity type of the liquid that is crude-oil, refined products, special products and lubricating oil. These function blocks are specific to Metric unit system.

This description is applicable to following function blocks

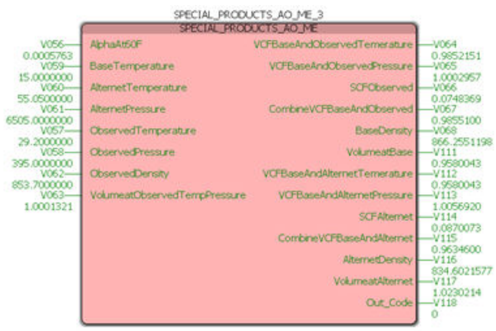
- CRUDE_OIL_AO_ME



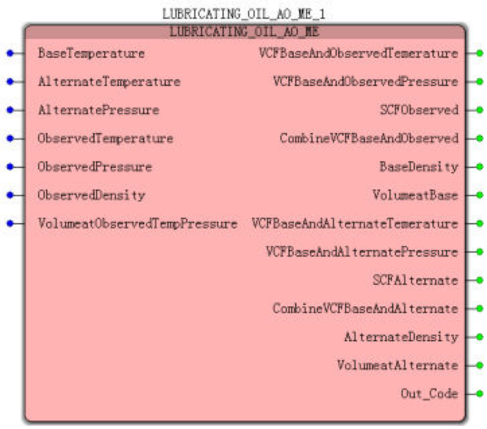
- REFINED_PRODUCTS_AO_ME



- SPECIAL_PRODUCTS_AO_ME



- LUBRICATING_OIL_AO_ME



These Function block expects the input values in the proper units (°C, Kpa, and kg/m³). If they are not in the proper units then appropriate unit conversion block should be used. The density values calculated by these function block are in the units of kg/m³. If these units do not match the original input units, then the output densities should be converted to that of the original input value’s units appropriate unit conversion block.

Input

Input Parameter	Data types	Description
BaseTemperature	LREAL	Value of Base temperature °C
AlternetTemperature	LREAL	Value of alternate temperature in °F
AlternetPressure	LREAL	Value of alternate Pressure in kpa
ObservedTemperature	LREAL	Value of observed temperature in °F
ObservedPressure	LREAL	Value of observed Pressure in kpa
ObservedDensity	LREAL	Value of observed density (kg/m3). If input density type is relative density or API Gravity then it must be converted into Density(kg/m3) using provided unit conversion blocks
AlphaAt60F	LREAL	Pre-calculated 60°F thermal expansion factor. This input parameter is only applicable for special products. For other commodity types this

Input Parameter	Data types	Description
		parameter is not present
VolumeatObservedTempPressure	LREAL	No Conversion needed, Most of the time it is optional input

Output

Output Parameter	Data types	Description
VCFBaseAndObservedTemerature	LREAL	Volume correction factor due to temperature between the base and observed temperatures
VCFBaseAndObservedPressure	LREAL	Volume correction factor due to pressure between the base and observed pressures at the observed temperature
SCFObserved	LREAL	Scaled compressibility factor at the observed temperature
CombineVCFBaseAndObserved	LREAL	Combined volume correction factor due to temperature and pressure between the base and observed conditions
BaseDensity	LREAL	Density at Base conditions
VolumeatBase	LREAL	Volume at base conditions
VCFBaseAndAlternetTemerature	LREAL	Volume correction factor due to temperature between the base and alternate temperatures
VCFBaseAndAlternetPressure	LREAL	Volume correction factor due to pressure between the base and alternate pressures at the alternate temperature
SCFAlternet	LREAL	Scaled compressibility factor at the alternate temperature

Output Parameter	Data types	Description
CombineVCFBaseAndAlternet	LREAL	Combined volume correction factor due to temperature and pressure between the base and alternate conditions
AlternetDensity	LREAL	Density at alternate conditions
VolumeatAlternet	LREAL	Volume at alternate conditions
Out_Code	INT	This out parameter returns success or fail code.

CHAPTER

5 API 21.1

The following libraries of API21.1 Function Blocks are supported:

Library	Description
API21_1Lib	The function block library provides support for creating flow measurement calculations for gas based on API 21.1 standard for Orifice, Turbine, Corolis and ultrasonic meters.
API21_1_V2	It is supported from R161.2 release. The function block library provides support for creating flow measurement calculations for gas based on API 21.1 standard for Orifice, Turbine, Corolis and ultrasonic meters with upgraded recent standards AGA3 (2012), AGA8 (2017) including GERG method and AGA5 (2009). This version also supports using override values in case of communication/out of range errors and also supports extended Quantity Transactions and Alarm and Events for effective audit trail.

The following API 21.1 meter run function blocks are available.

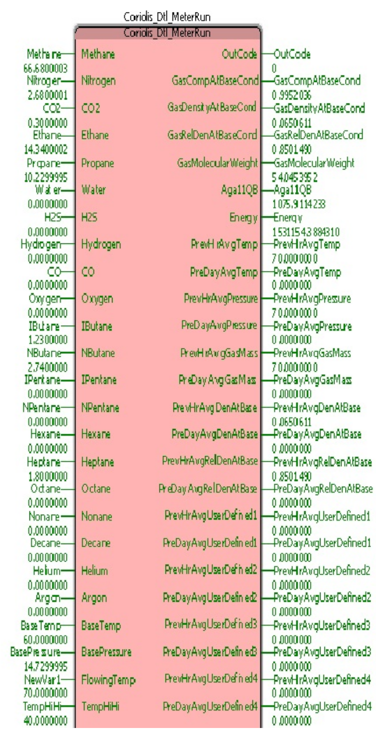
Function Block	Description
Orifice_Dtl_MeterRun	Orifice_Dtl_MeterRun calculates 1. Gas compressibility factor, density, relative density and molecular weight from AGA8 detailed method. 2. Volume flow rate at standard condition, mass flow rate and volume flow rate at base condition from AGA3. 3. Gas energy per hour 4. Hourly and daily Averages and Totals 5. Generates hourly & daily QTRs
Orifice_GM_MeterRun	Orifice_GM_MeterRun calculates 1. Gas compressibility factor, density, relative density and molecular weight from AGA8 gross method. 2. Volume flow rate at standard condition, mass flow rate and volume flow rate at base condition from AGA3.

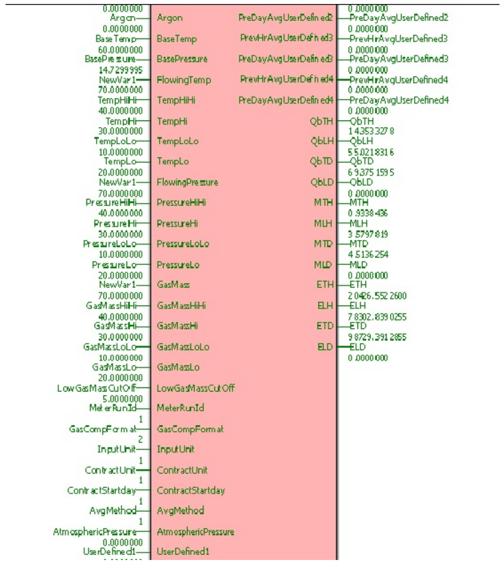
Function Block	Description
	3. Gas energy per hour 4. Hourly and daily Averages and Totals 5. Generates hourly & daily QTRs
Turbine_Dtl_MeterRun	Turbine_Dtl_MeterRun calculates 1. Gas compressibility factor, density, relative density and molecular weight from AGA8 detailed method. 2. Uncorrected volume flow rate at standard condition, mass flow rate and volume flow rate at base condition from AGA7. 3. Gas energy per hour 4. Hourly and daily Averages and Totals 5. Generates hourly & daily QTRs
Turbine_GM_MeterRun	Turbine_GM_MeterRun calculates 1. Gas compressibility factor, density, relative density and molecular weight from AGA8 gross method. 2. Uncorrected volume flow rate at standard condition, mass flow rate and volume flow rate at base condition from AGA7. 3. Gas energy per hour 4. Hourly and daily Averages and Totals 5. Generates hourly & daily QTRs
Coriolis_Dtl_MeterRun	Coriolis_Dtl_MeterRun calculates 1. Gas compressibility factor, density, relative density and molecular weight from AGA8 detailed method. 2. Volume flow rate at base condition from AGA11. 3. Gas energy per hour 4. Hourly and daily Averages and Totals 5. Generates hourly & daily QTRs
Coriolis_GM_MeterRun	Coriolis_GM_MeterRun calculates

Function Block	Description
	1. Gas compressibility factor, density, relative density and molecular weight from AGA8 gross method. 2. Volume flow rate at base condition from AGA11. 3. Gas energy per hour 4. Hourly and daily Averages and Totals 5. Generates hourly & daily QTRs

Orifice_Dtl_MeterRun Function Block

Here is an example for Orifice_Dtl_MeterRun:





Description

This Orifice_Dtl_MeterRun function block calculates gas compressibility factor, density, relative density and molecular weight from AGA8 detailed method, volume flow rate at standard condition, mass flow rate and volume flow rate at base condition from AGA3 and gas energy per hour from AGA5. It also calculates hourly and daily averages and totals. It generates hourly & daily QTRs and sends them to EFM application which logs them on the contoller's MRAM and flash memory. It also generates alarms when any of the process value crosses specified alarm limit. Orifice_Dtl_MeterRun expects the input parameters to be in US or Metric unit system. The exception is absolute viscosity of the gas that should be in centipoise in either unit system.

Input

Input Parameter	Data types	Description	Apply to
Methane	REAL	It could be in mole fraction or percentage.	All
Nitrogen	REAL	It could be in mole fraction or percentage.	All
CO2	REAL	It could be in mole fraction or percentage.	All

Input Parameter	Data types	Description	Apply to
Ethane	REAL	It could be in mole fraction or percentage.	All
Propane	REAL	It could be in mole fraction or percentage.	All
Water	REAL	It could be in mole fraction or percentage.	All
H2S	REAL	It could be in mole fraction or percentage.	All
Hydrogen	REAL	It could be in mole fraction or percentage.	All
CO	REAL	It could be in mole fraction or percentage.	All
Oxygen	REAL	It could be in mole fraction or percentage.	All
IButane	REAL	It could be in mole fraction or percentage.	All
NButane	REAL	It could be in mole fraction or percentage.	All
IPentane	REAL	It could be in mole fraction or percentage.	All
NPentane	REAL	It could be in mole fraction or percentage.	All
Hexane	REAL	It could be in mole fraction or percentage.	All
Heptane	REAL	It could be in mole fraction or percentage.	All
Octane	REAL	It could be in mole fraction or percentage.	All
Nonane	REAL	It could be in mole fraction or percentage.	All
Decane	REAL	It could be in mole fraction or percentage.	All

Input Parameter	Data types	Description	Apply to
Helium	REAL	It could be in mole fraction or percentage.	All
Argon	REAL	It could be in mole fraction or percentage.	All
DetailMethod	INT	Selection for Detail method: 1- Detail Method 2- GERG Method. GERG Method is only applicable for V2 function block.	Orifice_DtL_MeterRun_V2
BaseTemp	REAL	Base temperature should be in Fahrenheit for US unit system and in Celcius for Metric unit system. The recommended default is 60 Deg F.	All
BasePressure	REAL	Base pressure should be in Psia for US unit system and in Kpa for Metric unit system. The recommended default is 14.73 Psia.	All
FlowingTemp	REAL	Flowing temperature should be in Fahrenheit for US unit system and in Celcius for Metric unit system.	All
TempHiHi	REAL	This is the HiHi limit for flowing temperature. It should be either in Fahrenheit or Celcius.	All
TempHi	REAL	This is the Hi limit for flowing temperature. It should be either in Fahrenheit or Celcius.	All
TempLoLo	REAL	This is the LoLo limit for flowing temperature. It should be either in Fahrenheit or Celcius.	All
TempLo	REAL	This is the Lo limit for flowing temperature. It should be either in Fahrenheit or Celcius.	All
FlowTempIOSelection	INT	IO selection for meter temperature. The value should be {1} for Live or {2) for Keypad value.	Orifice_DtL_MeterRun_V2

Input Parameter	Data types	Description	Apply to
FlowTempStsStatus	USINT	Analog input channel status for meter temperature. The value should be {0} for Good or any positive integer for bad status.	Orifice_DtL_MeterRun_V2
FlowTempKeypadVal	REAL	Keypad value for meter temperature. The value that should be used when the meter temperature status is bad.	Orifice_DtL_MeterRun_V2
FlowingPressure	REAL	Flowing pressure should be in Psia for US unit system and in Kpa for Metric unit system.	All
PressureHiHi	REAL	This is the HiHi limit for flowing pressure. It should be either in Psia or Kpa.	All
PressureHi	REAL	This is the Hi limit for flowing pressure. It should be either in Psia or Kpa.	All
PressureLoLo	REAL	This is the LoLo limit for flowing pressure. It should be either in Psia or Kpa.	All
PressureLo	REAL	This is the Lo limit for flowing pressure. It should be either in Psia or Kpa.	All
FlowPresIOSelection	INT	IO selection for meter pressure. The value should be {1} for Live or {2) for Keypad value.	Orifice_DtL_MeterRun_V2
FlowPresStsStatus	USINT	Analog input channel status for meter pressure. The value should be {0} for Good or any positive integer for Bad status.	Orifice_DtL_MeterRun_V2
FlowPresKeypadVal	REAL	Analog input channel status for meter pressure. The value should be {0} for Good or any positive integer for Bad status.	Orifice_DtL_MeterRun_V2
DifferentialPressure	REAL	Differential Pressure should be in Inches of H2O for US unit system	All

Input Parameter	Data types	Description	Apply to
		and in Kpa for Metric unit system.	
DPHiHi	REAL	This is the HiHi limit for differential pressure. It should be either in Inches of H2O or Kpa.	All
DPHi	REAL	This is the Hi limit for differential pressure. It should be either in Inches of H2O or Kpa.	All
DPLoLo	REAL	This is the LoLo limit for differential pressure. It should be either in Inches of H2O or Kpa.	All
DPLo	REAL	This is the Lo limit for differential pressure. It should be either in Inches of H2O or Kpa.	All
LowDPCutOff	REAL	This is the low differential pressure cut off limit. It should be either in Inches of H2O or Kpa. This limit decides no flow condition.	All
MeterRunId	INT	This is an integer number that represents a configured meter run identifier.	All
GasCompFormat	INT	This parameter is for the gas composition format. It should be either mole fraction {1} or percentage {2}. **NOTE:** It is recommended to use 2 percentage as a default option.	All
InputUnit	INT	This parameter is for all the inputs of meter run function block. It should be either US {1} or Metric {2}.	All
ContractUnit	INT	This parameter is for all the outputs of meter run function block. It should be either US {1} or Metric {2}.	All

Input Parameter	Data types	Description	Apply to
ContractStartday	INT	This parameter represents the start of gas QTR day. Its value should be from 0 to 23.	All
AvgMethod	INT	This parameter is for averaging method to be used for averaging. As of now, it only supports value {1} that is for time weighted linear average.	All
MaintMode	INT	Parameter to Start or Stop the Maintenance Mode. The value should be either {0} Maintenance End or {1} for Maintenance Start. When Maintenance mode is started, an independent totalizer will be started and accumulate all the flow until the maintenance mode is stopped. During this period, non-resettable totals, hourly/daily/batch totals will be not incremented.	Orifice_DtL_MeterRun_V2
AGA8Version	INT	Selection for AGA 8 algorithm selection: 1- AGA 8 (1994) 2 - AGA 8 (2017)	Orifice_DtL_MeterRun_V2
TapsType	INT	Flange=1 and Pipe=2	All
OrificeMaterial	INT	STAINLESS STEEL=1, MONEL=2, CARBON STEEL=3, STAINLESS_S_304=4 and STAINLESS_S_316=5	All
PipeMaterial	INT	STAINLESS STEEL=1, MONEL=2, CARBON STEEL=3, STAINLESS_S_304=4 and STAINLESS_S_316=5	All
FluidType	INT	Compressible Fluid =1 and Non-Compressible Fluid=2	All
TapsLocation	INT	Upstream=1 and Downstream=2	All
OrificeDiameter	REAL	This parameter is the for orifice plate	All

Input Parameter	Data types	Description	Apply to
		diameter. It should be either in inches for US unit system or in millimeter for Metric unit system.	
OrfDiaMsrdTemp	REAL	This parameter represents the temperature at which orifice plate diameter is measured. It should be either in Fahrenheit for US unit system and in Celcius for Metric unit system.	All
PipeDiameter	REAL	This parameter is the for pipe diameter. It should be either in inches for US unit system or in millimeter for Metric unit system.	All
PipeDiaMsrdTemp	REAL	This parameter represents the temperature at which pipe diameter is measured. It should be either in Fahrenheit for US unit system and in Celcius for Metric unit system.	All
AbsViscosity	REAL	This parameter represents the absolute viscosity of the gas in Centipoise. In either unit system, it must be in centipoise only. (Recommended default=0.010268 cP - pg 34 part 4)	All
IsenExponent	REAL	This parameter isentropic exponent is a unit less number. (Recommended default=1.3 - pg 34 part 4)	All
AGA3Version	INT	Selection for AGA 3 algorithm selection: 1- AGA 3 (1992) 2- AGA 3 (2012)	Orifice_DtL_MeterRun_V2
AtmosphericPressure	REAL	Atmospheric pressure should be in Psia for US unit system and in Kpa for Metric unit system.	All

Input Parameter	Data types	Description	Apply to
		Atmospheric pressure is used to make Flowing pressure absolute when flowing pressure is measured by a pressure gauge. If flowing pressure is already absolute then it can be left zero.	
UserDefined1	REAL	This parameter is an optional one, if user wants to average some parameter then user can use it, it will be averaged and logged in the QTR.	All
UserDefined2	REAL	This parameter is an optional one, if user wants to average some parameter then user can use it, it will be averaged and logged in the QTR.	All
UserDefined3	REAL	This parameter is an optional one, if user wants to average some parameter then user can use it, it will be averaged and logged in the QTR.	All
UserDefined4	REAL	This parameter is an optional one, if user wants to average some parameter then user can use it, it will be averaged and logged in the QTR.	All

Output

Output Parameter	Data types	Description	Apply to
Out_Code	INT	This out parameter returns success or fail code.	All
GasCompAtBaseCond	LREAL	This parameter is gas compressibility factor at base condition. It is calculated in AGA8 Detailed method. It is unit less.	All
GasDensityAtBaseCond	LREAL	This parameter is gas density at base condition. It is calculated through AGA8 Detailed method. It	All

Output Parameter	Data types	Description	Apply to
		is in lbm/ft^3 for US unit system and in kg/m^3 for Metric unit system.	
GasRelDenAtBaseCond	LREAL	This parameter is gas relative density at base condition. It is calculated through AGA8 Detailed method. It is unit less.	All
GasMolecularWeight	LREAL	This parameter is gas molecular weight at base condition. It is calculated through AGA8 Detailed method. It is in lbm for US unit system and in kg for Metric unit system.	All
Aga3QV	LREAL	This parameter is volume flow rate at flowing condition. It is calculated through AGA3 method. It is in ft^3/hr for US unit system and in m^3/hr for Metric unit system.	All
Aga3QM	LREAL	This parameter is gas mass flow rate. It is calculated through AGA3 method. It is in lbm/hr for US unit system and in kg/hr for Metric unit system.	All
Aga3QB	LREAL	This parameter is volume flow rate at base condition. It is calculated through AGA3 method. It is in ft^3/hr for US unit system and in m^3/hr for Metric unit system.	All
Energy	LREAL	This parameter is gas energy per hour. It is calculated through AGA5 method. It is in Btu/hr for US unit system and in MJ/hr for Metric unit system.	All
PrevHrAvgTemp	LREAL	This parameter is previous hour average for temperature.	All
PreDayAvgTemp	LREAL	This parameter is previous day	All

Output Parameter	Data types	Description	Apply to
		average for temperature.	
PrevHrAvgPressure	LREAL	This parameter is previous hour average for pressure.	All
PreDayAvgPressure	LREAL	This parameter is previous day average for pressure.	All
PrevHrAvgDP	LREAL	This parameter is previous hour average for differential pressure.	All
PreDayAvgDP	LREAL	This parameter is previous day average for differential pressure.	All
PrevHrAvgDenAtBase	LREAL	This parameter is previous hour average for density at base condition.	All
PreDayAvgDenAtBase	LREAL	This parameter is previous day average for density at base condition.	All
PrevHrAvgRelDenAtBase	LREAL	This parameter is previous hour average for relative density at base condition.	All
PreDayAvgRelDenAtBase	LREAL	This parameter is previous day average for relative density at base condition.	All
PrevHrAvgFlowExt	LREAL	This parameter is previous hour average for flow extension.	All
PreDayAvgFlowExt	LREAL	This parameter is previous day average for flow extension.	All
PrevHrAvgUserDefined1	LREAL	This parameter is previous hour average for user defined parameter1.	All
PreDayAvgUserDefined1	LREAL	This parameter is previous day average for user defined parameter1.	All
PrevHrAvgUserDefined2	LREAL	This parameter is previous hour average for user defined parameter2.	All

Output Parameter	Data types	Description	Apply to
PreDayAvgUserDefined2	LREAL	This parameter is previous day average for user defined parameter2.	All
PrevHrAvgUserDefined3	LREAL	This parameter is previous hour average for user defined parameter3.	All
PreDayAvgUserDefined3	LREAL	This parameter is previous day average for user defined parameter3.	All
PrevHrAvgUserDefined4	LREAL	This parameter is previous hour average for user defined parameter4.	All
PreDayAvgUserDefined4	LREAL	This parameter is previous day average for user defined parameter4.	All
QbTH	LREAL	This parameter is volume flow rate at base condition total for this hour.	All
QbLH	LREAL	This parameter is volume flow rate at base condition total for last hour.	All
QbTD	LREAL	This parameter is volume flow rate at base condition total for this day.	All
QbLD	LREAL	This parameter is volume flow rate at base condition total for last day.	All
MTH	LREAL	This parameter is mass flow rate total for this hour.	All
MLH	LREAL	This parameter is mass flow rate total for last hour.	All
MTD	LREAL	This parameter is mass flow rate total for this day.	All
MLD	LREAL	This parameter is mass flow rate total for last day.	All

Output Parameter	Data types	Description	Apply to
ETH	LREAL	This parameter is energy total for this hour.	All
ELH	LREAL	This parameter is energy total for last hour.	All
ETD	LREAL	This parameter is energy total for this day.	All
ELD	LREAL	This parameter is energy total for last day.	All
QbNR	LREAL	Non-Resettable or Cumulative total for volume at Base. Unit - ft3/hr for US, m3/hr for Metric.	Orifice_DtL_MeterRun_V2
MNR	LREAL	Non-Resettable or Cumulative total for Mass. Unit - lbm/hr for US, kg/hr for Metric.	Orifice_DtL_MeterRun_V2
ENR	LREAL	Non-Resettable or Cumulative total for Energy. Unit -Btu/hr for US, MJ/hr for Metric.	Orifice_DtL_MeterRun_V2
QbRollover	INT	Rollover flag for volume at base condition non-resettable total. The value {1} indicates Rollover otherwise {0}. The value for roll-over is 999,999,999. The rollover flag will be on 5 times interval time.	Orifice_DtL_MeterRun_V2
MRollover	INT	Rollover flag for Non-Resettable Mass total. The value {1} indicates Rollover otherwise {0}. The value for roll-over is 999,999,999. The rollover flag will be on 5 times interval time.	Orifice_DtL_MeterRun_V2
ERollover	INT	Rollover flag for Non-Resettable Energy total. The value {1} indicates Rollover otherwise {0}. The value for roll-over is 999,999,999. The rollover flag will be on 5 times interval time.	Orifice_DtL_MeterRun_V2

Output Parameter	Data types	Description	Apply to
QbMaint	LREAL	Volume at Base in Maintenance mode.	Orifice_DtL_MeterRun_V2
MMaint	LREAL	Mass in Maintenance mode.	Orifice_DtL_MeterRun_V2
EMaint	LREAL	Energy at Base in Maintenance mode.	Orifice_DtL_MeterRun_V2
GERG2008CV	LREAL	Heat Capacity at Constant Volume (J/mol K).	Orifice_DtL_MeterRun_V2
GERG2008CP	LREAL	Heat Capacity at Constant Pressure (J/mol K).	Orifice_DtL_MeterRun_V2
GERG2008W	LREAL	Speed of sound in gas being measured. Unit - ft/sec for US, meter/sec for Metric.	Orifice_DtL_MeterRun_V2

NOTE: The above outputs including averages and totals would be in the contract unit. The QTR generated by this function block contains following fields. Datetime; Mass flow rate (total) Flowtime; Volume flow rate at base condition (total); Energy per hour (total); Average Temperature; Average Pressure; Average Differential pressure; Average Density at base condition; Average Relative Density at base condition; Average Flow Extension; Average User Defined 1 (optional); Average User Defined 2 (optional); Average User Defined 3 (optional); Average User Defined 4 (optional).

Following are the error codes for the above meter run function block.

Out Code	Description	Apply to
0	SUCCESS	All
1[1]	ERROR: PRESSURE HAS A NEGATIVE DERIVATIVE DEFAULT GAS DENSITY USED	All

Out Code	Description	Apply to
	ERROR: A COMPONENT MOLE FRACTION < 0.0 OR > 1.0	Orifice_DtL_MeterRun_V2
2[1]	WARNING: DENSITY IN BRAKET EXCEEDS MAXIMUM DEFAULT PROCEEDURE USED	All
	WARNING: SUM OF MOLE FRACTIONS < 0.9999 OR > 1.0001	Orifice_DtL_MeterRun_V2
3[1]	ERROR: MAXIMUM ITERATIONS EXCEEDED IN BRAKET DEFAULT DENSITY USED	All
	WARNING: PRESSURE BASE (PB) <= 0.0 OR >= 16 PSIA	Orifice_DtL_MeterRun_V2
4[1]	ERROR: MAXIMUM ITERATIONS IN DDETAIL EXCEEDED LAST DENSITY USED	All
	WARNING: TEMPERATURE BASE (TB) <= 32.0 OR >= 77.0 DEG F	Orifice_DtL_MeterRun_V2
32	ERROR: FLOWING PRESSURE (PF) <= 0.0 OR > 40,000. PSIA	All
33	ERROR: FLOWING TEMPERATURE (TF) < -200 OR > 760 DEG F	All
36	ERROR: MOLE FRACTION FOR METHANE < 0.0 OR > 1.0 FOR NITROGEN < 0.0 OR > 1.0 FOR CARBON DIOXIDE < 0.0 OR > 1.0 FOR ETHANE < 0.0 OR > 1.0 FOR PROPANE < 0.0 OR > 0.12 FOR WATER < 0.0 OR > 0.10 FOR H2S < 0.0 OR > 1.0 FOR HYDROGEN < 0.0 OR > 1.0 FOR CARBON MONOXIDE < 0.0 OR > 0.03 FOR OXYGEN < 0.0 OR > 0.21 FOR BUTANES < 0.0 OR > 0.06 FOR PENTANES < 0.0 OR > 0.04 FOR HEXANES + < 0.0 OR > 0.10 FOR HELIUM < 0.0 OR > 0.03 FOR ARGON < 0.0 OR > 1.0	Orifice_DtL_MeterRun

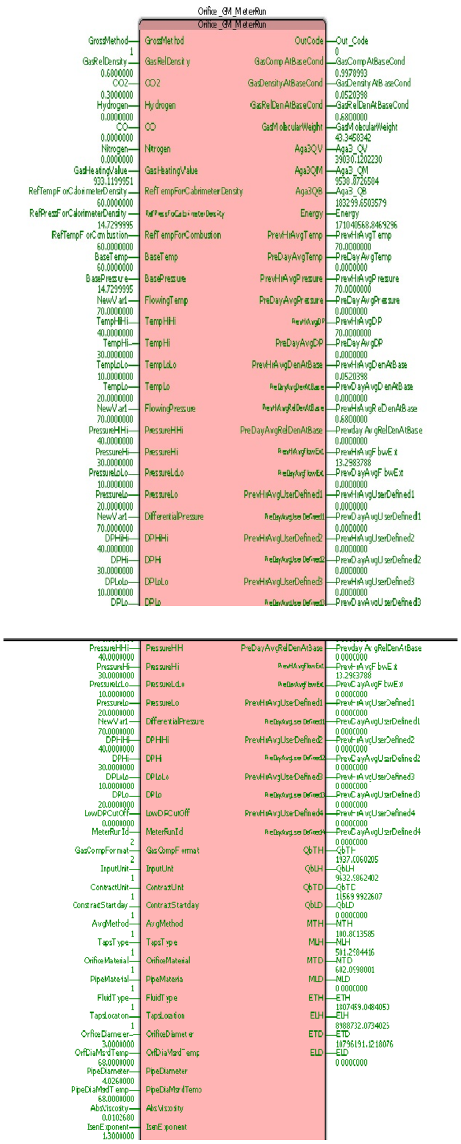
Out Code	Description	Apply to
37	ERROR: REFERENCE TEMPERATURE < 32.0 OR > 77.0 DEG F	All
38	ERROR: REFERENCE PRESSURE < 12.9 OR > 16.01 PSIA	All
39	ERROR: SUM OF MOLE FRACTIONS < 0.98 OR > 1.020	All
42	WARNING: FLOWING PRESSURE (PF) < 0.0 OR > 1750. PSIA	All
43	WARNING: FLOWING TEMPERATURE (TF) < 17 OR > 143 DEG F	All
45	WARNING: ANY COMPONENT MOLE FRACTION OUTSIDE OF AGA REPORT NO. 8 RECOMMENDED RANGE	Orifice_DtL_MeterRun_V2
46	WARNING: MOLE FRACTION FOR METHANE < 0.45 OR > 1.0 FOR NITROGEN < 0.0 OR > 0.5 FOR CARBON DIOXIDE < 0.0 OR > 0.3 FOR ETHANE < 0.0 OR > 0.1 FOR PROPANE < 0.0 OR > 0.04 FOR WATER < 0.0 OR >= 0.0005 FOR H2S < 0.0 OR > 0.0002 FOR HYDROGEN < 0.0 OR > 0.1 FOR CARBON MONOXIDE < 0.0 OR > 0.03 FOR OXYGEN < 0.0 OR > 0.0 FOR BUTANES < 0.0 OR > 0.01 FOR PENTANES < 0.0 OR >= 0.003 FOR HEXANES + < 0.0 OR >= 0.002 FOR HELIUM < 0.0 OR >= 0.002 FOR ARGON < 0.0 OR > 0.0	Orifice_DtL_MeterRun
49	WARNING: SUM OF MOLE FRACTIONS < 0.9999 OR > 1.0001	All
51	ERROR: NTAPS WAS NOT 0, 1 OR 2	All
52	ERROR: FLOWING PRESSURE WAS <= 0.0 OR > 40000. PSIA	All
53	ERROR: FLOWING TEMPERATURE < -200. OR > 760. DEG F	All
54	ERROR: MATORF OR MATPIPE WAS NOT 0, 1, 2 OR 3	All
55	ERROR: ORIFICE DIAMETER WAS <= 0 OR => 100.0 INCHES	All
56	ERROR: PIPE DIAMETER WAS <= 0 OR => 100.0 INCHES	All
57	ERROR: FLOWING OR STANDARD DENSITY WAS <= 0.0 LBM/FT^3	All
58	ERROR: DIFFERENTIAL PRESSURE WAS <= 0.0 INCHES H2O	All

Out Code	Description	Apply to
59	ERROR: GAS VISCOSITY WAS <= 0.005 OR > 0.5 CENTIPOISES	All
60	ERROR: ISENTROPIC EXPONENT <= 1.0 OR => 2.0	All
61	ERROR: IFLUID WAS NOT 0, 1 OR 2	All
62	ERROR: STANDARD TEMPERATURE WAS NOT = 60.0 DEG F	All
63	ERROR: STANDARD PRESSURE WAS NOT = 14.73 PSIA	All
64	ERROR: TAP LOCATION WAS NOT 0, 1 OR 2 FOR NTAPS=2 (PIPE) OR TAP LOCATION WAS NOT 1 FOR NTAPS=1 (FLANGE)	All
65	ERROR: SUPERCOMPRESSIBILITY FACTOR WAS <= 0.0	All
66	ERROR: RELATIVE DENSITY AT STANDARD CONDITIONS WAS < 0.07 OR > 1.52	All
67	ERROR: CALIBRATION FACTOR WAS <= 0.0	All
68	ERROR: COMPRESSIBILITY FACTOR AT STANDARD CONDITIONS <= 0.0	All
69	ERROR: BETA RATIO (DO/DM) <= 0.0 OR => 1.0	All
70	ERROR: IF NTAPS = 1, GOF2015_OPTION NOT = 1 OR = 0	Orifice_DtL_MeterRun_V2
71	ERROR: IF NTAPS = 2, GOF2015_OPTION NOT = 0	Orifice_DtL_MeterRun_V2
72	ERROR: DIFFERENTIAL PRESSURE WAS GREATER THAN UPSTREAM STATIC PRESSURE	Orifice_DtL_MeterRun_V2
75	WARNING: ORIFICE DIAMETER WAS <= 0.45 INCHES	All
76	WARNING: PIPE DIAMETER WAS <= 2.0 INCHES	All
79	WARNING: BETA RATIO (DO/DM) WAS <= 0.1 OR >= 0.75	All
80	WARNING: IF GOF2015_OPTION = 1, (HW)/(27.7072*(PF)) = OR > 0.25; IF GOF2015_OPTION = 0, (HW)/(27.707*(PF)) > 0.2	Orifice_DtL_MeterRun_V2
86	WARNING: Flowing Pressure greater than 2017 AGA8 GERG-	Orifice_DtL_

Out Code	Description	Apply to
	2008 Full Quality Range (10,150 PSIA)	MeterRun_ V2
87	WARNING: Flowing Pressure greater than 2017 AGA8 GERG-2008 Range (5075 PSIA)	Orifice_DtL_ MeterRun_ V2
88	WARNING: Flowing Temperature outside 2017 AGA8 GERG-2008 Full Quality Range (-352 F < TF < 800 F)	Orifice_DtL_ MeterRun_ V2
89	WARNING: Flowing Temperature outside 2017 AGA8 GERG-2008 Range (-298 F < TF < 350 F)	Orifice_DtL_ MeterRun_ V2
90	WARNING: A Component Mole % outside 2017 AGA8 GERG-2008 Intermediate Quality Range	Orifice_DtL_ MeterRun_ V2
91	WARNING: A Component Mole % outside 2017 AGA8 GERG-2008 Pipeline Quality Range	Orifice_DtL_ MeterRun_ V2
NOTE 1: Error codes 1~4 are common between AGA 8 and AGA 5. You must take caution and analyze when these specific out codes appear to determine the source.		

Orifice_GM_MeterRun Function Block

Here is an example for Orifice_GM_MeterRun:



Description

This Orifice_GM_MeterRun function block calculates gas compressibility factor, density, relative density and molecular weight from AGA8 gross method, volume flow rate at standard condition, mass flow rate and volume flow rate at base condition from AGA3 and gas energy per hour from AGA5. It also calculates hourly and daily averages and totals. It generates hourly & daily QTRs and sends them to EFM application which logs them on the contoller's MRAM and flash memory. It also generates alarms when any of the process value crosses specified alarm limit. Orifice_GM_MeterRun expects the input parameters to be in US or Metric unit system. The exception is absolute viscosity of the gas that should be in centipoise in either unit system.

Input

Input Parameter	Data types	Description	Apply to
GrossMethod	INT	This parameter represents gross method number. It should be either {1} for gross method 1 and {2} for gross method 2.	All
GasRelDensity	REAL	This parameter is gas relative density at reference condition. It is unit less.	All
CO2	REAL	It could be in mole fraction or percentage.	All
Hydrogen	REAL	It could be in mole fraction or percentage.	All
CO	REAL	It could be in mole fraction or percentage.	All
Nitrogen	REAL	It could be in mole fraction or percentage. This parameter is only required for gross method 2, for gross method 1, it can be zero.	All
GasHeatingValue	REAL	This parameter is gas heating value. It is only required for gross method 1, for gross method 2, it can be zero. It is in Btu/ft^3 for US unit system and in MJ/m^3 for Metric unit system.	All

Input Parameter	Data types	Description	Apply to
RefTempForCalorimeterDensity	REAL	This parameter is reference temperature for calorimeter density. It should be in Fahrenheit for US unit system and in Celcius for Metric unit system. The recommended default is 60 Deg F.	All
RefPressForCalorimeterDensity	REAL	This parameter is reference pressure for calorimeter density. It should be in Psia for US unit system and in Kpa for Metric unit system. The recommended default is 14.73 Psia.	All
RefTempForCombustion	REAL	This parameter is reference temperature for combustion. It should be in Fahrenheit for US unit system and in Celcius for Metric unit system. The recommended default is 60 Deg F.	All
BaseTemp	REAL	Base temperature should be in Fahrenheit for US unit system and in Celcius for Metric unit system. The recommended default is 60 Deg F.	All
BasePressure	REAL	Base pressure should be in Psia for US unit system and in Kpa for Metric unit system. The recommended default is 14.73 Psia.	All
FlowingTemp	REAL	Flowing temperature should be in Fahrenheit for US unit system and in Celcius for Metric unit system.	All

Input Parameter	Data types	Description	Apply to
TempHiHi	REAL	This is the HiHi limit for flowing temperature. It should be either in Fahrenheit or Celcius.	All
TempHi	REAL	This is the Hi limit for flowing temperature. It should be either in Fahrenheit or Celcius.	All
TempLoLo	REAL	This is the LoLo limit for flowing temperature. It should be either in Fahrenheit or Celcius.	All
TempLo	REAL	This is the Lo limit for flowing temperature. It should be either in Fahrenheit or Celcius.	All
FlowTempIOSelection	INT	IO selection for meter temperature. The value should be {1} for Live or {2} for Keypad value.	Orifice_GM_ MeterRun_ V2
FlowTempStsStatus	USINT	Analog input channel status for meter temperature. The value should be {0} for Good or any positive integer for bad status.	Orifice_GM_ MeterRun_ V2
FlowTempKeypadVal	REAL	Keypad value for meter temperature. The value that should be used when the meter temperature status is bad.	Orifice_GM_ MeterRun_ V2
FlowingPressure	REAL	Flowing pressure should be in Psia for US unit system and in Kpa for Metric unit system.	All
PressureHiHi	REAL	This is the HiHi limit for flowing pressure. It should be either in Psia or Kpa.	All

Input Parameter	Data types	Description	Apply to
PressureHi	REAL	This is the Hi limit for flowing pressure. It should be either in Psia or Kpa.	All
PressureLoLo	REAL	This is the LoLo limit for flowing pressure. It should be either in Psia or Kpa.	All
PressureLo	REAL	This is the Lo limit for flowing pressure. It should be either in Psia or Kpa.	All
FlowPresIOSelection	INT	IO selection for meter pressure. The value should be {1} for Live or {2) for Keypad value.	Orifice_GM_MeterRun_V2
FlowPresStsStatus	USINT	Analog input channel status for meter pressure. The value should be {0} for Good or any positive integer for Bad status.	Orifice_GM_MeterRun_V2
FlowPresKeypadVal	REAL	Analog input channel status for meter pressure. The value should be {0} for Good or any positive integer for Bad status.	Orifice_GM_MeterRun_V2
DifferentialPressure	REAL	Differential Pressure should be in Inches of H2O for US unit system and in Kpa for Metric unit system.	All
DPHiHi	REAL	This is the HiHi limit for differential pressure. It should be either in Inches of H2O or Kpa.	All
DPHi	REAL	This is the Hi limit for differential pressure. It should be either in Inches of H2O or Kpa.	All
DPLoLo	REAL	This is the LoLo limit for	All

Input Parameter	Data types	Description	Apply to
		differential pressure. It should be either in Inches of H2O or Kpa.	
DPLo	REAL	This is the Lo limit for differential pressure. It should be either in Inches of H2O or Kpa.	All
LowDPCutOff	REAL	This is the low differential pressure cut off limit. It should be either in Inches of H2O or Kpa. This limit decides no flow condition.	All
MeterRunId	INT	This is an integer number that represents a configured meter run identifier.	All
GasCompFormat	INT	This parameter is for the gas composition format. It should be either mole fraction {1} or percentage {2}. **NOTE:** It is recommended to use 2 percentage as a default option.	All
InputUnit	INT	This parameter is for all the inputs of meter run function block. It should be either US {1} or Metric {2}.	All
ContractUnit	INT	This parameter is for all the outputs of meter run function block. It should be either US {1} or Metric {2}.	All
ContractStartday	INT	This parameter represents the start of gas QTR day. Its value should be from 0 to 23.	All

Input Parameter	Data types	Description	Apply to
AvgMethod	INT	This parameter is for averaging method to be used for averaging. As of now, it only supports value {1} that is for time weighted linear average.	All
MaintMode	INT	Parameter to Start or Stop the Maintenance Mode. The value should be either {0} Maintenance End or {1} for Maintenance Start. When Maintenance mode is started, an independent totalizer will be started and accumulate all the flow until the maintenance mode is stopped. During this period, non-resettable totals, hourly/daily/batch totals will be not incremented.	Orifice_GM_MeterRun_V2
AGA8Version	INT	Selection for AGA 8 algorithm selection: 1- AGA 8 (1994) 2 - AGA 8 (2017)	Orifice_GM_MeterRun_V2
TapsType	INT	Flange=1 and Pipe=2	All
OrificeMaterial	INT	STAINLESS STEEL=1, MONEL=2, CARBON STEEL=3, STAINLESS_S_304=4 and STAINLESS_S_316=5	All
PipeMaterial	INT	STAINLESS STEEL=1, MONEL=2, CARBON STEEL=3, STAINLESS_S_304=4 and STAINLESS_S_316=5	All
FluidType	INT	Compressible Fluid =1	All

Input Parameter	Data types	Description	Apply to
		and Non-Compressible Fluid=2	
TapsLocation	INT	Upstream=1 and Downstream=2	All
OrificeDiameter	REAL	This parameter is the for orifice plate diameter. It should be either in inches for US unit system or in millimeter for Metric unit system.	All
OrfDiaMsrdTemp	REAL	This parameter represents the temperature at which orifice plate diameter is measured. It should be either in Fahrenheit for US unit system and in Celcius for Metric unit system.	All
PipeDiameter	REAL	This parameter is the for pipe diameter. It should be either in inches for US unit system or in millimeter for Metric unit system.	All
PipeDiaMsrdTemp	REAL	This parameter represents the temperature at which pipe diameter is measured. It should be either in Fahrenheit for US unit system and in Celcius for Metric unit system.	All
AbsViscosity	REAL	This parameter represents the absolute viscosity of the gas in Centipoise. In either unit system, it must be in centipoise only. (Recommended default=0.010268 cP - pg 34 part 4)	All

Input Parameter	Data types	Description	Apply to
IsenExponent	REAL	This parameter isentropic exponent is a unit less number. (Recommended default=1.3 - pg 34 part 4)	All
AGA3Version	INT	Selection for AGA 3 algorithm selection: 1- AGA 3 (1992) 2- AGA 3 (2012)	Orifice_GM_MeterRun_V2
AtmosphericPressure	REAL	Atmospheric pressure should be in Psia for US unit system and in Kpa for Metric unit system. Atmospheric pressure is used to make Flowing pressure absolute when flowing pressure is measured by a pressure gauge. If flowing pressure is already absolute then it can be left zero.	All
UserDefined1	REAL	This parameter is an optional one, if user wants to average some parameter then user can use it, it will be averaged and logged in the QTR.	All
UserDefined2	REAL	This parameter is an optional one, if user wants to average some parameter then user can use it, it will be averaged and logged in the QTR.	All
UserDefined3	REAL	This parameter is an optional one, if user wants to average some parameter then user can use it, it will	All

Input Parameter	Data types	Description	Apply to
		be averaged and logged in the QTR.	
UserDefined4	REAL	This parameter is an optional one, if user wants to average some parameter then user can use it, it will be averaged and logged in the QTR.	All

Output

Output Parameter	Data types	Description	Apply to
Out_Code	INT	This out parameter returns success or fail code.	All
GasCompAtBaseCond	LREAL	This parameter is gas compressibility factor at base condition. It is calculated in AGA8 Detailed method. It is unit less.	All
GasDensityAtBaseCond	LREAL	This parameter is gas density at base condition. It is calculated through AGA8 Detailed method. It is in lbm/ft^3 for US unit system and in kg/m^3 for Metric unit system.	All
GasRelDenAtBaseCond	LREAL	This parameter is gas relative density at base condition. It is calculated through AGA8 Detailed method. It is unit less.	All
GasMolecularWeight	LREAL	This parameter is gas molecular weight at base condition. It is calculated through AGA8 Detailed method. It is in lbm for US unit system and in kg for Metric unit system.	All
Aga3QV	LREAL	This parameter is volume flow rate at flowing condition. It is	All

Output Parameter	Data types	Description	Apply to
		calculated through AGA3 method. It is in ft^3/hr for US unit system and in m^3/hr for Metric unit system.	
Aga3QM	LREAL	This parameter is gas mass flow rate. It is calculated through AGA3 method. It is in lbm/hr for US unit system and in kg/hr for Metric unit system.	All
Aga3QB	LREAL	This parameter is volume flow rate at base condition. It is calculated through AGA3 method. It is in ft^3/hr for US unit system and in m^3/hr for Metric unit system.	All
Energy	LREAL	This parameter is gas energy per hour. It is calculated through AGA5 method. It is in Btu/hr for US unit system and in MJ/hr for Metric unit system.	All
PrevHrAvgTemp	LREAL	This parameter is previous hour average for temperature.	All
PreDayAvgTemp	LREAL	This parameter is previous day average for temperature.	All
PrevHrAvgPressure	LREAL	This parameter is previous hour average for pressure.	All
PreDayAvgPressure	LREAL	This parameter is previous day average for pressure.	All
PrevHrAvgDP	LREAL	This parameter is previous hour average for differential pressure.	All
PreDayAvgDP	LREAL	This parameter is previous day average for differential pressure.	All
PrevHrAvgDenAtBase	LREAL	This parameter is previous hour average for density at base condition.	All

Output Parameter	Data types	Description	Apply to
PreDayAvgDenAtBase	LREAL	This parameter is previous day average for density at base condition.	All
PrevHrAvgRelDenAtBase	LREAL	This parameter is previous hour average for relative density at base condition.	All
PreDayAvgRelDenAtBase	LREAL	This parameter is previous day average for relative density at base condition.	All
PrevHrAvgFlowExt	LREAL	This parameter is previous hour average for flow extension.	All
PreDayAvgFlowExt	LREAL	This parameter is previous day average for flow extension.	All
PrevHrAvgUserDefined1	LREAL	This parameter is previous hour average for user defined parameter1.	All
PreDayAvgUserDefined1	LREAL	This parameter is previous day average for user defined parameter1.	All
PrevHrAvgUserDefined2	LREAL	This parameter is previous hour average for user defined parameter2.	All
PreDayAvgUserDefined2	LREAL	This parameter is previous day average for user defined parameter2.	All
PrevHrAvgUserDefined3	LREAL	This parameter is previous hour average for user defined parameter3.	All
PreDayAvgUserDefined3	LREAL	This parameter is previous day average for user defined parameter3.	All
PrevHrAvgUserDefined4	LREAL	This parameter is previous hour average for user defined parameter4.	All
PreDayAvgUserDefined4	LREAL	This parameter is previous day	All

Output Parameter	Data types	Description	Apply to
		average for user defined parameter4.	
QbTH	LREAL	This parameter is volume flow rate at base condition total for this hour.	All
QbLH	LREAL	This parameter is volume flow rate at base condition total for last hour.	All
QbTD	LREAL	This parameter is volume flow rate at base condition total for this day.	All
QbLD	LREAL	This parameter is volume flow rate at base condition total for last day.	All
MTH	LREAL	This parameter is mass flow rate total for this hour.	All
MLH	LREAL	This parameter is mass flow rate total for last hour.	All
MTD	LREAL	This parameter is mass flow rate total for this day.	All
MLD	LREAL	This parameter is mass flow rate total for last day.	All
ETH	LREAL	This parameter is energy total for this hour.	All
ELH	LREAL	This parameter is energy total for last hour.	All
ETD	LREAL	This parameter is energy total for this day.	All
ELD	LREAL	This parameter is energy total for last day.	All
QbNR	LREAL	Non-Resettable or Cumulative total for volume at Base. Unit - ft3/hr for US, m3/hr for Metric.	Orifice_GM_MeterRun_V2

Output Parameter	Data types	Description	Apply to
MNR	LREAL	Non-Resettable or Cumulative total for Mass. Unit - lbm/hr for US, kg/hr for Metric.	Orifice_GM_MeterRun_V2
ENR	LREAL	Non-Resettable or Cumulative total for Energy. Unit -Btu/hr for US, MJ/hr for Metric.	Orifice_GM_MeterRun_V2
QbRollover	INT	Rollover flag for volume at base condition non-resettable total. The value {1} indicates Rollover otherwise {0}. The value for roll-over is 999,999,999. The rollover flag will be on 5 times interval time.	Orifice_GM_MeterRun_V2
MRollover	INT	Rollover flag for Non-Resettable Mass total. The value {1} indicates Rollover otherwise {0}. The value for roll-over is 999,999,999. The rollover flag will be on 5 times interval time.	Orifice_GM_MeterRun_V2
ERollover	INT	Rollover flag for Non-Resettable Energy total. The value {1} indicates Rollover otherwise {0}. The value for roll-over is 999,999,999. The rollover flag will be on 5 times interval time.	Orifice_GM_MeterRun_V2
QbMaint	LREAL	Volume at Base in Maintenance mode.	Orifice_GM_MeterRun_V2
MMaint	LREAL	Mass in Maintenance mode.	Orifice_GM_MeterRun_V2
EMaint	LREAL	Energy at Base in Maintenance mode.	Orifice_GM_MeterRun_V2

NOTE: The above outputs including averages and totals would be in the contract unit. The QTR generated by this function block contains following fields. Datetime; Mass flow rate (total) Flowtime; Volume flow rate at base condition (total); Energy per hour (total); Average Temperature; Average Pressure; Average Differential pressure; Average Density at base condition; Average Relative Density at base condition; Average Flow Extension; Average User Defined 1 (optional); Average User Defined 2 (optional); Average User Defined 3 (optional); Average User Defined 4 (optional).

Following are the error codes for the above meter run function block.

Out Code	Description	Apply to
0	SUCCESS	All
5	ERROR: THE ROOT WAS NOT BOUNDED IN DGROSS	All
6	ERROR: NO CONVERGENCE IN DGROSS	All
7	ERROR: VIRGS SQURE ROOT NEGATIVE	All
8	ERROR: COMBINED VALUES OF GRGR, X[2] AND HV NOT CONSISTENT	All
9	ERROR: INVALID TERM IN VIRGS	All
11	ERROR: METHOD WAS NOT 1 OR 2	All
12	ERROR: FLOWING PRESSURE (PF) <= 0.0 OR > 1740.0 PSIA	All
13	ERROR: FLOWING TEMPERATURE (TF) < 14.0 OR > 149.0 DEG F	All
14	ERROR: HEATING VALUE (HV) < 477.0 OR > 1211.0 BTU/FT^3	All
15	ERROR: GAS RELATIVE DENSITY (GRGR) < 0.55 OR > 0.870	All
16	ERROR: MOLE FRACTION FOR N2 < 0.0 OR > 0.50 OR FOR CO2 < 0.0 OR > 0.30 OR FOR H2 < 0.0 OR > 0.10 OR FOR CO < 0.0 OR > 0.03	All
17	ERROR: REFERENCE TEMPERATURE < 32.0 OR > 77.0 DEG F	All
18	ERROR: REFERENCE PRESSURE < 13.0 OR > 16.0 PSIA	All
22	WARNING: FLOWING PRESSURE (PF) <= 0.0 OR > 1200.0 PSIA	Orifice_GM_

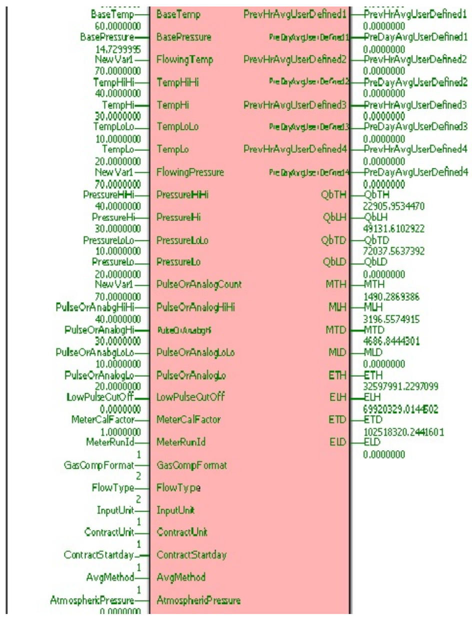
Out Code	Description	Apply to
		MeterRun
23	WARNING: FLOWING TEMPERATURE (TF) < 32.0 OR > 130.0 DEG F	Orifice_GM_ MeterRun
24	WARNING: HEATING VALUE (HV) < 805.0 OR > 1208.0 BTU/FT^3	Orifice_GM_ MeterRun
25	WARNING: GAS RELATIVE DENSITY (GRGR) < 0.55 OR > 0.800	Orifice_GM_ MeterRun
26	WARNING: MOLE FRACTION FOR N2 < 0.0 OR > 0.20 OR FOR CO2 < 0.0 OR > 0.20 OR FOR H2 < 0.0 OR > 0.0 OR FOR CO < 0.0 OR >0.0	Orifice_GM_ MeterRun
51	ERROR: NTAPS WAS NOT 0, 1 OR 2	All
54	ERROR: MATORF OR MATPIPE WAS NOT 0, 1, 2 OR 3	All
55	ERROR: ORIFICE DIAMETER WAS <= 0 OR => 100.0 INCHES	All
56	ERROR: PIPE DIAMETER WAS <= 0 OR => 100.0 INCHES	All
57	ERROR: FLOWING OR STANDARD DENSITY WAS <= 0.0 LBM/FT^3	All
58	ERROR: DIFFERENTIAL PRESSURE WAS <= 0.0 INCHES H2O	All
59	ERROR: GAS VISCOSITY WAS <= 0.005 OR > 0.5 CENTIPOISES	All
60	ERROR: ISENTROPIC EXPONENT <= 1.0 OR => 2.0	All
61	ERROR: IFLUID WAS NOT 0, 1 OR 2	All
62	ERROR: STANDARD TEMPERATURE WAS NOT = 60.0 DEG F	All
63	ERROR: STANDARD PRESSURE WAS NOT = 14.73 PSIA	All
64	ERROR: TAP LOCATION WAS NOT 0, 1 OR 2 FOR NTAPS=2 (PIPE) OR TAP LOCATION WAS NOT 1 FOR NTAPS=1 (FLANGE)	All
65	ERROR: SUPERCOMPRESSIBILITY FACTOR WAS <= 0.0	All
66	ERROR: RELATIVE DENSITY AT STANDARD CONDITIONS WAS < 0.07 OR > 1.52	All
67	ERROR: CALIBRATION FACTOR WAS <= 0.0	All
68	ERROR: COMPRESSIBILITY FACTOR AT STANDARD	All

Out Code	Description	Apply to
	CONDITIONS <= 0.0	
69	ERROR: BETA RATIO (DO/DM) <= 0.0 OR => 1.0	All
70	ERROR: IF NTAPS = 1, GOF2015_OPTION NOT = 1 OR = 0	Orifice_GM_MeterRun_V2
71	ERROR: IF NTAPS = 2, GOF2015_OPTION NOT = 0	Orifice_GM_MeterRun_V2
72	ERROR: DIFFERENTIAL PRESSURE WAS GREATER THAN UPSTREAM STATIC PRESSURE	Orifice_GM_MeterRun_V2
75	WARNING: ORIFICE DIAMETER WAS <= 0.45 INCHES	All
76	WARNING: PIPE DIAMETER WAS <= 2.0 INCHES	All
79	WARNING: BETA RATIO (DO/DM) WAS <= 0.1 OR >= 0.75	All
80	WARNING: IF GOF2015_OPTION = 1, (HW)/(27.7072*(PF)) = OR > 0.25; IF GOF2015_OPTION = 0, (HW)/(27.707*(PF)) > 0.2	Orifice_GM_MeterRun_V2
81	WARNING: FLOWING PRESSURE (PF) > 1500.0 PSIA AGA8 2017 RANGE 1	Orifice_GM_MeterRun_V2
82	WARNING: FLOWING TEMPERATURE (TF) < 17.01 OR > 143.0 DEG F AGA8 2017 RANGE 2 OR (TF) < 25.0 OR > 143.0 DEG F AGA8 2017 RANGE 1	Orifice_GM_MeterRun_V2
83	WARNING: HEATING VALUE (HV) < 665.0 OR > 1100.0 BTU/FT^3 AGA8 2017 RANGE 2 OR (HV) < 930.0 OR > 1040.0 BTU/FT^3 AGA8 2017 RANGE 1	Orifice_GM_MeterRun_V2
84	WARNING: GAS RELATIVE DENSITY (GRGR) < 0.554 OR > 0.801 AGA8 2017 RANGE 2 OR (GRGR) < 0.554 OR > 0.630 AGA8 RANGE 1	Orifice_GM_MeterRun_V2
85	WARNING: MOLE FRACTION FOR N2 > 0.20 AGA8 2017 RANGE 2 OR N2 > 0.07 AGA8 2017 RANGE 1 OR FOR CO2 > 0.25 AGA8 2017 RANGE 2 OR CO2 > 0.03 AGA8	Orifice_GM_MeterRun_V2

Out Code	Description	Apply to
	2017 RANGE 1 OR FOR H2 < 0.0 OR > 0.0 AGA8 2017 RANGE 1 AND 2 OR FOR CO < 0.0 OR > 0.0 AGA8 2017 RANGE 1 AND 2	

Turbine_Dtl_MeterRun Function Block

Here is an example for Turbine_Dtl_MeterRun:



Description

This Turbine_Dtl_MeterRun function block calculates gas compressibility factor, density, relative density and molecular weight from AGA8 detailed method, uncorrected flow, mass flow rate and volume flow rate at base condition from AGA7 and gas energy per hour from AGA5. It also calculates hourly and daily averages and totals. It generates hourly & daily QTRs and sends them to EFM application which logs them on he contoller's MRAM and flash memory. It also generates alarms when any of the process value crosses specified alarm limit.

Turbine_Dtl_MeterRun expects the input parameters to be in US or Metric unit system.

This description is also applicable to function block Ultrasonic_Dtl_MeterRun. Both Turbine and Ultrasonic meters are technically same.

Input

Input Parameter	Data types	Description	Apply to
Methane	REAL	It could be in mole fraction or percentage.	All
Nitrogen	REAL	It could be in mole fraction or	All

Input Parameter	Data types	Description	Apply to
		percentage.	
CO2	REAL	It could be in mole fraction or percentage.	All
Ethane	REAL	It could be in mole fraction or percentage.	All
Propane	REAL	It could be in mole fraction or percentage.	All
Water	REAL	It could be in mole fraction or percentage.	All
H2S	REAL	It could be in mole fraction or percentage.	All
Hydrogen	REAL	It could be in mole fraction or percentage.	All
CO	REAL	It could be in mole fraction or percentage.	All
Oxygen	REAL	It could be in mole fraction or percentage.	All
IButane	REAL	It could be in mole fraction or percentage.	All
NButane	REAL	It could be in mole fraction or percentage.	All
IPentane	REAL	It could be in mole fraction or percentage.	All
NPentane	REAL	It could be in mole fraction or percentage.	All
Hexane	REAL	It could be in mole fraction or percentage.	All
Heptane	REAL	It could be in mole fraction or percentage.	All
Octane	REAL	It could be in mole fraction or percentage.	All
Nonane	REAL	It could be in mole fraction or percentage.	All

Input Parameter	Data types	Description	Apply to
Decane	REAL	It could be in mole fraction or percentage.	All
Helium	REAL	It could be in mole fraction or percentage.	All
Argon	REAL	It could be in mole fraction or percentage.	All
DetailMethod	INT	Selection for Detail method: 1- Detail Method 2- GERG Method. GERG Method is only applicable for V2 function block.	Turbine_ Dtl_ MeterRun_ V2
BaseTemp	REAL	Base temperature should be in Fahrenheit for US unit system and in Celcius for Metric unit system. The recommended default is 60 Deg F.	All
BasePressure	REAL	Base pressure should be in Psia for US unit system and in Kpa for Metric unit system. The recommended default is 14.73 Psia.	All
FlowingTemp	REAL	Flowing temperature should be in Fahrenheit for US unit system and in Celcius for Metric unit system.	All
TempHiHi	REAL	This is the HiHi limit for flowing temperature. It should be either in Fahrenheit or Celcius.	All
TempHi	REAL	This is the Hi limit for flowing temperature. It should be either in Fahrenheit or Celcius.	All
TempLoLo	REAL	This is the LoLo limit for flowing temperature. It should be either in Fahrenheit or Celcius.	All
TempLo	REAL	This is the Lo limit for flowing temperature. It should be either in Fahrenheit or Celcius.	All

Input Parameter	Data types	Description	Apply to
FlowTempIOSelection	INT	IO selection for meter temperature. The value should be {1} for Live or {2} for Keypad value.	Turbine_ Dtl_ MeterRun_ V2
FlowTempStsStatus	USINT	Analog input channel status for meter temperature. The value should be {0} for Good or any positive integer for bad status.	Turbine_ Dtl_ MeterRun_ V2
FlowTempKeypadVal	REAL	Keypad value for meter temperature. The value that should be used when the meter temperature status is bad.	Turbine_ Dtl_ MeterRun_ V2
FlowingPressure	REAL	Flowing pressure should be in Psia for US unit system and in Kpa for Metric unit system.	All
PressureHiHi	REAL	This is the HiHi limit for flowing pressure. It should be either in Psia or Kpa.	All
PressureHi	REAL	This is the Hi limit for flowing pressure. It should be either in Psia or Kpa.	All
PressureLoLo	REAL	This is the LoLo limit for flowing pressure. It should be either in Psia or Kpa.	All
PressureLo	REAL	This is the Lo limit for flowing pressure. It should be either in Psia or Kpa.	All
PulseOrAnalogCount	REAL	For pulse input, it would be a number while for analog input, it should be in ft^3/hr for US unit system and in m^3/hr for Metric unit system.	Turbine_ Dtl_ MeterRun
PulseOrAnalogHiHi	REAL	This is the HiHi limit for pulse or analog input. For pulse input, it would be a number while for analog input, it should be in ft^3/hr for US unit system and in m^3/hr for Metric unit system.	Turbine_ Dtl_ MeterRun

Input Parameter	Data types	Description	Apply to
PulseOrAnalogHi	REAL	This is the Hi limit for pulse or analog input. For pulse input, it would be a number while for analog input, it should be in ft^3/hr for US unit system and in m^3/hr for Metric unit system.	Turbine_ Dtl_ MeterRun
PulseOrAnalogLoLo	REAL	This is the LoLo limit for pulse or analog input. For pulse input, it would be a number while for analog input, it should be in ft^3/hr for US unit system and in m^3/hr for Metric unit system.	Turbine_ Dtl_ MeterRun
PulseOrAnalogLo	REAL	This is the Lo limit for pulse or analog input. For pulse input, it would be a number while for analog input, it should be in ft^3/hr for US unit system and in m^3/hr for Metric unit system.	Turbine_ Dtl_ MeterRun
LowPulseCutOff	REAL	This is the low pulse cut off limit. For pulse input, it is a number. For analog input, it should be in ft^3/hr for US unit system and in m^3/hr for Metric unit system.	Turbine_ Dtl_ MeterRun
FlowPresIOSelection	INT	IO selection for meter pressure. The value should be {1} for Live or {2) for Keypad value.	Turbine_ Dtl_ MeterRun_ V2
FlowPresStsStatus	USINT	Analog input channel status for meter pressure. The value should be {0} for Good or any positive integer for Bad status.	Turbine_ Dtl_ MeterRun_ V2
FlowPresKeypadVal	REAL	Analog input channel status for meter pressure. The value should be {0} for Good or any positive integer for Bad status.	Turbine_ Dtl_ MeterRun_ V2
Analog	REAL	Value of analog input if flow type = analog. The value should be in ft3/hr	Turbine_ Dtl_

Input Parameter	Data types	Description	Apply to
		for US unit or m^3/hr for Metric unit.	MeterRun_V2
AnalogHiHi	REAL	This is the HiHi limit for analog input. The value should be in ft3/hr for US unit or m^3/hr for Metric unit.	Turbine_Dtl_MeterRun_V2
AnalogHi	REAL	This is the Hi limit for analog input. The value should be in ft3/hr for US unit or m^3/hr for Metric unit.	Turbine_Dtl_MeterRun_V2
AnalogLoLo	REAL	This is the LoLo limit for analog input. The value should be in ft3/hr for US unit or m^3/hr for Metric unit.	Turbine_Dtl_MeterRun_V2
AnalogLo	REAL	This is the Lo limit for analog input. The value should be in ft3/hr for US unit or m^3/hr for Metric unit.	Turbine_Dtl_MeterRun_V2
LowFlowCutOff	REAL	Low flow cutoff value checks the no flow condition in the calculations. If the flow is s less than this number, it will be considered as no flow condition. Unit is m3/hr for Metric unit, ft3/hr for US unit.	Turbine_Dtl_MeterRun_V2
Pulse	UDINT	Pulse counter value	Turbine_Dtl_MeterRun_V2
LowPulseCutOff	UINT	Low pulse cutoff value checks the no flow condition in the calculations. If the Pulse increment is less than this number, it will be considered as no flow condition.	Turbine_Dtl_MeterRun_V2
MeterCalFactor	REAL	It is same as Meter K Factor which converts pulse from flow meter into volume. The value should be in pulses/m3 or pulses/ft3. When the	All

Input Parameter	Data types	Description	Apply to
		flow type is analog, this is the correction factor to apply for volume calculation and the default value should be 1.0.	
MeterRunId	INT	This is an integer number that represents a configured meter run identifier.	All
GasCompFormat	INT	This parameter is for the gas composition format. It should be either mole fraction {1} or percentage {2}. **NOTE:** It is recommended to use 2 percentage as a default option.	All
FlowType	INT	This parameter represents the flow type, it should be either {1} Pulse Accumulated or {2} Analog Flow Rate.	All
InputUnit	INT	This parameter is for all the inputs of meter run function block. It should be either US {1} or Metric {2}.	All
ContractUnit	INT	This parameter is for all the outputs of meter run function block. It should be either US {1} or Metric {2}.	All
ContractStartday	INT	This parameter represents the start of gas QTR day. Its value should be from 0 to 23.	All
AvgMethod	INT	This parameter is for averaging method to be used for averaging. As of now, it only supports value {1} that is for time weighted linear average.	All
MaintMode	INT	Parameter to Start or Stop the Maintenance Mode. The value should be either {0} Maintenance End or {1} for Maintenance Start. When	Turbine_ Dtl_ MeterRun_ V2

Input Parameter	Data types	Description	Apply to
		Maintenance mode is started, an independent totalizer will be started and accumulate all the flow until the maintenance mode is stopped. During this period, non-resettable totals, hourly/daily/batch totals will be not incremented.	
AGA8Version	INT	Selection for AGA 8 algorithm selection: 1- AGA 8 (1994) 2 - AGA 8 (2017)	Turbine_ Dtl_ MeterRun_ V2
AtmosphericPressure	REAL	Atmospheric pressure should be in Psia for US unit system and in Kpa for Metric unit system. Atmospheric pressure is used to make Flowing pressure absolute when flowing pressure is measured by a pressure gauge. If flowing pressure is already absolute then it can be left zero.	All
UserDefined1	REAL	This parameter is an optional one, if user wants to average some parameter then user can use it, it will be averaged and logged in the QTR.	All
UserDefined2	REAL	This parameter is an optional one, if user wants to average some parameter then user can use it, it will be averaged and logged in the QTR.	All
UserDefined3	REAL	This parameter is an optional one, if user wants to average some parameter then user can use it, it will be averaged and logged in the QTR.	All
UserDefined4	REAL	This parameter is an optional one, if user wants to average some parameter then user can use it, it will be averaged and logged in the QTR.	All

Output

Output Parameter	Data types	Description	Apply to
Out_Code	INT	This out parameter returns success or fail code.	All
GasCompAtBaseCond	LREAL	This parameter is gas compressibility factor at base condition. It is calculated in AGA8 Detailed method. It is unit less.	All
GasDensityAtBaseCond	LREAL	This parameter is gas density at base condition. It is calculated through AGA8 Detailed method. It is in lbm/ft^3 for US unit system and in kg/m^3 for Metric unit system.	All
GasRelDenAtBaseCond	LREAL	This parameter is gas relative density at base condition. It is calculated through AGA8 Detailed method. It is unit less.	All
GasMolecularWeight	LREAL	This parameter is gas molecular weight at base condition. It is calculated through AGA8 Detailed method. It is in lbm for US unit system and in kg for Metric unit system.	All
UncorrectedFlow	LREAL	This parameter is uncorrected flow rate. It is in ft^3/hr for US unit system and in m^3/hr for Metric unit system in case FlowType= 2 (Analog). For FlowType = 1 (Pulse), this parameter represents instantaneous volume flow from the last time flow is calculated. Unit is ft3/task interval (US) or m3/task interval (Metric).	All
Aga7QM	LREAL	This parameter is gas mass flow rate. It is calculated through AGA7 method. It is in lbm/hr for US unit system and in kg/hr for Metric	All

Output Parameter	Data types	Description	Apply to
		unit system in case FlowType= 2 (Analog). For FlowType = 1 (Pulse), this parameter represents instantaneous mass flow from the last time mass flow is calculated. Unit is lbm/task interval (US) or kg/task interval (Metric).	
Aga7QB	LREAL	This parameter is volume flow rate at base condition. It is calculated through AGA7 method. It is in ft^3/hr for US unit system and in m^3/hr for Metric unit system in case FlowType= 2 (Analog). For FlowType = 1 (Pulse), this parameter represents instantaneous base volume flow from the last time flow is calculated. Unit is ft3/task interval (US) or m3/task interval (Metric).	All
Energy	LREAL	This parameter is gas energy per hour. It is calculated through AGA5 method. It is in BTU/hr for US unit system and in MJ/hr for Metric unit system in case FlowType= 2 (Analog). For FlowType = 1 (Pulse), this parameter represents instantaneous energy flow from the last time flow is calculated. Unit is BTU/task interval (US) or MJ/task interval (Metric).	All
PrevHrAvgTemp	LREAL	This parameter is previous hour average for temperature.	All
PreDayAvgTemp	LREAL	This parameter is previous day average for temperature.	All
PrevHrAvgPressure	LREAL	This parameter is previous hour average for pressure.	All
PreDayAvgPressure	LREAL	This parameter is previous day	All

Output Parameter	Data types	Description	Apply to
		average for pressure.	
PrevHrAvgPulse	LREAL	This parameter is previous hour average for analog input.	All
PreDayAvgPulse	LREAL	This parameter is previous day average for analog input.	All
PrevHrAvgDenAtBase	LREAL	This parameter is previous hour average for density at base condition.	All
PreDayAvgDenAtBase	LREAL	This parameter is previous day average for density at base condition.	All
PrevHrAvgRelDenAtBase	LREAL	This parameter is previous hour average for relative density at base condition.	All
PreDayAvgRelDenAtBase	LREAL	This parameter is previous day average for relative density at base condition.	All
PrevHrAvgUncorrFlow	LREAL	This parameter is previous hour average for uncorrected flow.	All
PreDayAvgUncorrFlow	LREAL	This parameter is previous day average for uncorrected flow.	All
PrevHrAvgUserDefined1	LREAL	This parameter is previous hour average for user defined parameter1.	All
PreDayAvgUserDefined1	LREAL	This parameter is previous day average for user defined parameter1.	All
PrevHrAvgUserDefined2	LREAL	This parameter is previous hour average for user defined parameter2.	All
PreDayAvgUserDefined2	LREAL	This parameter is previous day average for user defined parameter2.	All
PrevHrAvgUserDefined3	LREAL	This parameter is previous hour	All

Output Parameter	Data types	Description	Apply to
		average for user defined parameter3.	
PreDayAvgUserDefined3	LREAL	This parameter is previous day average for user defined parameter3.	All
PrevHrAvgUserDefined4	LREAL	This parameter is previous hour average for user defined parameter4.	All
PreDayAvgUserDefined4	LREAL	This parameter is previous day average for user defined parameter4.	All
QbTH	LREAL	This parameter is volume flow rate at base condition total for this hour.	All
QbLH	LREAL	This parameter is volume flow rate at base condition total for last hour.	All
QbTD	LREAL	This parameter is volume flow rate at base condition total for this day.	All
QbLD	LREAL	This parameter is volume flow rate at base condition total for last day.	All
MTH	LREAL	This parameter is mass flow rate total for this hour.	All
MLH	LREAL	This parameter is mass flow rate total for last hour.	All
MTD	LREAL	This parameter is mass flow rate total for this day.	All
MLD	LREAL	This parameter is mass flow rate total for last day.	All
ETH	LREAL	This parameter is energy total for this hour.	All
ELH	LREAL	This parameter is energy total for last hour.	All
ETD	LREAL	This parameter is energy total for	All

Output Parameter	Data types	Description	Apply to
		this day.	
ELD	LREAL	This parameter is energy total for last day.	All
QbNR	LREAL	Non-Resettable or Cumulative total for volume at Base. Unit - ft3/hr for US, m3/hr for Metric.	Turbine_ Dtl_ MeterRun_ V2
MNR	LREAL	Non-Resettable or Cumulative total for Mass. Unit - lbm/hr for US, kg/hr for Metric.	Turbine_ Dtl_ MeterRun_ V2
ENR	LREAL	Non-Resettable or Cumulative total for Energy. Unit -Btu/hr for US, MJ/hr for Metric.	Turbine_ Dtl_ MeterRun_ V2
QbRollover	INT	Rollover flag for volume at base condition non-resettable total. The value {1} indicates Rollover otherwise {0}. The value for roll-over is 999,999,999. The rollover flag will be on 5 times interval time.	Turbine_ Dtl_ MeterRun_ V2
MRollover	INT	Rollover flag for Non-Resettable Mass total. The value {1} indicates Rollover otherwise {0}. The value for roll-over is 999,999,999. The rollover flag will be on 5 times interval time.	Turbine_ Dtl_ MeterRun_ V2
ERollover	INT	Rollover flag for Non-Resettable Energy total. The value {1} indicates Rollover otherwise {0}. The value for roll-over is 999,999,999. The rollover flag will be on 5 times interval time.	Turbine_ Dtl_ MeterRun_ V2
QbMaint	LREAL	Volume at Base in Maintenance mode.	Turbine_ Dtl_

Output Parameter	Data types	Description	Apply to
			MeterRun_ V2
MMaint	LREAL	Mass in Maintenance mode.	Turbine_ Dtl_ MeterRun_ V2
EMaint	LREAL	Energy at Base in Maintenance mode.	Turbine_ Dtl_ MeterRun_ V2
GERG2008CV	LREAL	Heat Capacity at Constant Volume (J/mol K).	Turbine_ Dtl_ MeterRun_ V2
GERG2008CP	LREAL	Heat Capacity at Constant Pressure (J/mol K).	Turbine_ Dtl_ MeterRun_ V2
GERG2008W	LREAL	Speed of sound in gas being measured. Unit - ft/sec for US, meter/sec for Metric.	Turbine_ Dtl_ MeterRun_ V2

NOTE: The above outputs including averages and totals would be in the contract unit. The QTR generated by this function block contains following fields. Date; Time; Flow Time; Volume at Base; Mass; Energy; Temperature; Pressure; Pulse; Density; Uncorrected flow; Relative Density; Average User Defined 1 (optional); Average User Defined 2 (optional); Average User Defined 3 (optional); Average User Defined 4 (optional).

Following are the error codes for the above meter run function block.

Out Code	Description	Apply to
0	SUCCESS	All
1[1]	ERROR: PRESSURE HAS A NEGATIVE DERIVATIVE DEFAULT GAS DENSITY USED	All

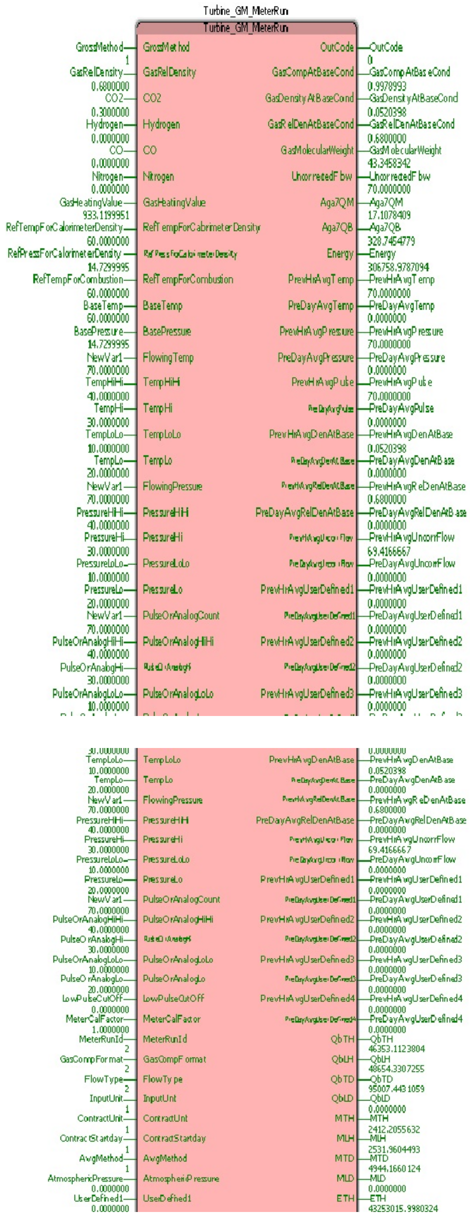
Out Code	Description	Apply to
	ERROR: A COMPONENT MOLE FRACTION < 0.0 OR > 1.0	Turbine_ Dtl_ MeterRun_ V2
2[1]	WARNING: DENSITY IN BRAKET EXCEEDS MAXIMUM DEFAULT PROCEEDURE USED	All
	WARNING: SUM OF MOLE FRACTIONS < 0.9999 OR > 1.0001	Turbine_ Dtl_ MeterRun_ V2
3[1]	ERROR: MAXIMUM ITERATIONS EXCEEDED IN BRAKET DEFAULT DENSITY USED	All
	WARNING: PRESSURE BASE (PB) <= 0.0 OR >= 16 PSIA	Turbine_ Dtl_ MeterRun_ V2
4[1]	ERROR: MAXIMUM ITERATIONS IN DDETAIL EXCEEDED LAST DENSITY USED	All
	WARNING: TEMPERATURE BASE (TB) <= 32.0 OR >= 77.0 DEG F	Turbine_ Dtl_ MeterRun_ V2
32	ERROR: FLOWING PRESSURE (PF) <= 0.0 OR > 40,000. PSIA	All
33	ERROR: FLOWING TEMPERATURE (TF) < -200 OR > 760 DEG F	All
36	ERROR: MOLE FRACTION FOR METHANE < 0.0 OR > 1.0 FOR NITROGEN < 0.0 OR > 1.0 FOR CARBON DIOXIDE < 0.0 OR > 1.0 FOR ETHANE < 0.0 OR > 1.0 FOR PROPANE < 0.0 OR > 0.12 FOR WATER < 0.0 OR > 0.10 FOR H2S < 0.0 OR > 1.0 FOR HYDROGEN < 0.0 OR > 1.0 FOR CARBON MONOXIDE < 0.0 OR > 0.03 FOR OXYGEN < 0.0 OR > 0.21 FOR BUTANES < 0.0 OR > 0.06	Turbine_ Dtl_ MeterRun

Out Code	Description	Apply to
	FOR PENTANES < 0.0 OR > 0.04 FOR HEXANES + < 0.0 OR > 0.10 FOR HELIUM < 0.0 OR > 0.03 FOR ARGON < 0.0 OR > 1.0	
37	ERROR: REFERENCE TEMPERATURE < 32.0 OR > 77.0 DEG F	All
38	ERROR: REFERENCE PRESSURE < 12.9 OR > 16.01 PSIA	All
39	ERROR: SUM OF MOLE FRACTIONS < 0.98 OR > 1.020	All
42	WARNING: FLOWING PRESSURE (PF) < 0.0 OR > 1750. PSIA	All
43	WARNING: FLOWING TEMPERATURE (TF) < 17 OR > 143 DEG F	All
45	WARNING: ANY COMPONENT MOLE FRACTION OUTSIDE OF AGA REPORT NO. 8 RECOMMENDED RANGE	Turbine_ Dtl_ MeterRun _ V2
46	WARNING: MOLE FRACTION FOR METHANE < 0.45 OR > 1.0 FOR NITROGEN < 0.0 OR > 0.5 FOR CARBON DIOXIDE < 0.0 OR > 0.3 FOR ETHANE < 0.0 OR > 0.1 FOR PROPANE < 0.0 OR > 0.04 FOR WATER < 0.0 OR >= 0.0005 FOR H2S < 0.0 OR > 0.0002 FOR HYDROGEN < 0.0 OR > 0.1 FOR CARBON MONOXIDE < 0.0 OR > 0.03 FOR OXYGEN < 0.0 OR > 0.0 FOR BUTANES < 0.0 OR > 0.01 FOR PENTANES < 0.0 OR >= 0.003 FOR HEXANES + < 0.0 OR >= 0.002 FOR HELIUM < 0.0 OR >= 0.002 FOR ARGON < 0.0 OR > 0.0	Turbine_ Dtl_ MeterRun
49	WARNING: SUM OF MOLE FRACTIONS < 0.9999 OR > 1.0001	All
86	WARNING: Flowing Pressure greater than 2017 AGA8 GERG-2008 Full Quality Range (10,150 PSIA)	Turbine_ Dtl_ MeterRun_ V2

Out Code	Description	Apply to
87	WARNING: Flowing Pressure greater than 2017 AGA8 GERG-2008 Range (5075 PSIA)	Turbine_Dtl_MeterRun_V2
88	WARNING: Flowing Temperature outside 2017 AGA8 GERG-2008 Full Quality Range (-352 F < TF < 800 F)	Turbine_Dtl_MeterRun_V2
89	WARNING: Flowing Temperature outside 2017 AGA8 GERG-2008 Range (-298 F < TF < 350 F)	Turbine_Dtl_MeterRun_V2
90	WARNING: A Component Mole % outside 2017 AGA8 GERG-2008 Intermediate Quality Range	Turbine_Dtl_MeterRun_V2
91	WARNING: A Component Mole % outside 2017 AGA8 GERG-2008 Pipeline Quality Range	Turbine_Dtl_MeterRun_V2
NOTE 1: Error codes 1~4 are common between AGA 8 and AGA 5. You must take caution and analyze when these specific out codes appear to determine the source.		

Turbine_GM_MeterRun Function Block

Here is an example for Turbine_GM_MeterRun:



Description

This Turbine_GM_MeterRun function block calculates gas compressibility factor, density, relative density and molecular weight from AGA8 gross method, uncorrected flow, mass flow rate and volume flow rate at base condition from AGA7 and gas energy per hour from AGA5. It also calculates hourly and daily averages and totals. It generates hourly & daily QTRs and sends them to EFM application which logs them on the controller’s MRAM and flash memory. It also generates alarms when any of the process value crosses specified alarm limit.

Turbine_GM_MeterRun expects the input parameters to be in US or Metric unit system.

This description is also applicable to function block Ultrasonic_GM_MeterRun. Both Turbine and Ultrasonic meters are technically same.

Input

Input Parameter	Data types	Description	Apply to
GrossMethod	INT	This parameter represents gross method number. It should be either {1} for gross method 1 and {2} for gross method 2.	All
GasRelDensity	REAL	This parameter is gas relative density at reference condition. It is unit less.	All
CO2	REAL	It could be in mole fraction or percentage.	All
Hydrogen	REAL	It could be in mole fraction or percentage.	All
CO	REAL	It could be in mole fraction or percentage.	All
Nitrogen	REAL	It could be in mole fraction or percentage. This parameter is only required for gross method 2, for gross method 1, it can be zero.	All
GasHeatingValue	REAL	This parameter is gas heating value. It is only required for gross method 1, for gross method 2, it can	All

Input Parameter	Data types	Description	Apply to
		be zero. It is in Btu/ft^3 for US unit system and in MJ/m^3 for Metric unit system.	
RefTempForCalorimeterDensity	REAL	This parameter is reference temperature for calorimeter density. It should be in Fahrenheit for US unit system and in Celcius for Metric unit system. The recommended default is 60 Deg F.	All
RefPressForCalorimeterDensity	REAL	This parameter is reference pressure for calorimeter density. It should be in Psia for US unit system and in Kpa for Metric unit system. The recommended default is 14.73 Psia.	All
RefTempForCombustion	REAL	This parameter is reference temperature for combustion. It should be in Fahrenheit for US unit system and in Celcius for Metric unit system. The recommended default is 60 Deg F.	All
BaseTemp	REAL	Base temperature should be in Fahrenheit for US unit system and in Celcius for Metric unit system. The recommended default is 60 Deg F.	All
BasePressure	REAL	Base pressure should be in Psia for US unit system and in Kpa for Metric unit system. The recommended default is 14.73 Psia.	All

Input Parameter	Data types	Description	Apply to
FlowingTemp	REAL	Flowing temperature should be in Fahrenheit for US unit system and in Celcius for Metric unit system.	All
TempHiHi	REAL	This is the HiHi limit for flowing temperature. It should be either in Fahrenheit or Celcius.	All
TempHi	REAL	This is the Hi limit for flowing temperature. It should be either in Fahrenheit or Celcius.	All
TempLoLo	REAL	This is the LoLo limit for flowing temperature. It should be either in Fahrenheit or Celcius.	All
TempLo	REAL	This is the Lo limit for flowing temperature. It should be either in Fahrenheit or Celcius.	All
FlowTempIOSelection	INT	IO selection for meter temperature. The value should be {1} for Live or {2} for Keypad value.	Turbine_ GM_ MeterRun_ V2
FlowTempStsStatus	USINT	Analog input channel status for meter temperature. The value should be {0} for Good or any positive integer for bad status.	Turbine_ GM_ MeterRun_ V2
FlowTempKeypadVal	REAL	Keypad value for meter temperature. The value that should be used when the meter temperature status is bad.	Turbine_ GM_ MeterRun_ V2
FlowingPressure	REAL	Flowing pressure should be in Psia for US unit system	All

Input Parameter	Data types	Description	Apply to
		and in Kpa for Metric unit system.	
PressureHiHi	REAL	This is the HiHi limit for flowing pressure. It should be either in Psia or Kpa.	All
PressureHi	REAL	This is the Hi limit for flowing pressure. It should be either in Psia or Kpa.	All
PressureLoLo	REAL	This is the LoLo limit for flowing pressure. It should be either in Psia or Kpa.	All
PressureLo	REAL	This is the Lo limit for flowing pressure. It should be either in Psia or Kpa.	All
PulseOrAnalogCount	REAL	For pulse input, it would be a number while for analog input, it should be in ft^3/hr for US unit system and in m^3/hr for Metric unit system.	Turbine_ GM_ MeterRun
PulseOrAnalogHiHi	REAL	This is the HiHi limit for pulse or analog input. For pulse input, it would be a number while for analog input, it should be in ft^3/hr for US unit system and in m^3/hr for Metric unit system.	Turbine_ GM_ MeterRun
PulseOrAnalogHi	REAL	This is the Hi limit for pulse or analog input. For pulse input, it would be a number while for analog input, it should be in ft^3/hr for US unit system and in m^3/hr for Metric unit system.	Turbine_ GM_ MeterRun
PulseOrAnalogLoLo	REAL	This is the LoLo limit for pulse or analog input. For	Turbine_ GM_

Input Parameter	Data types	Description	Apply to
		pulse input, it would be a number while for analog input, it should be in ft^3/hr for US unit system and in m^3/hr for Metric unit system.	MeterRun
PulseOrAnalogLo	REAL	This is the Lo limit for pulse or analog input. For pulse input, it would be a number while for analog input, it should be in ft^3/hr for US unit system and in m^3/hr for Metric unit system.	Turbine_ GM_ MeterRun
LowPulseCutOff	REAL	This is the low pulse cut off limit. For pulse input, it is a number. For analog input, it should be in ft^3/hr for US unit system and in m^3/hr for Metric unit system.	Turbine_ GM_ MeterRun
FlowPresIOSelection	INT	IO selection for meter pressure. The value should be {1} for Live or {2) for Keypad value.	Turbine_ GM_ MeterRun_ V2
FlowPresStsStatus	USINT	Analog input channel status for meter pressure. The value should be {0} for Good or any positive integer for Bad status.	Turbine_ GM_ MeterRun_ V2
FlowPresKeypadVal	REAL	Analog input channel status for meter pressure. The value should be {0} for Good or any positive integer for Bad status.	Turbine_ GM_ MeterRun_ V2
Analog	REAL	Value of analog input if flow type = analog. The value should be in ft3/hr for US unit or m^3/hr for Metric unit.	Turbine_ GM_ MeterRun_ V2

Input Parameter	Data types	Description	Apply to
AnalogHiHi	REAL	This is the HiHi limit for analog input. The value should be in ft3/hr for US unit or m^3/hr for Metric unit.	Turbine_ GM_ MeterRun_ V2
AnalogHi	REAL	This is the Hi limit for analog input. The value should be in ft3/hr for US unit or m^3/hr for Metric unit.	Turbine_ GM_ MeterRun_ V2
AnalogLoLo	REAL	This is the LoLo limit for analog input. The value should be in ft3/hr for US unit or m^3/hr for Metric unit.	Turbine_ GM_ MeterRun_ V2
AnalogLo	REAL	This is the Lo limit for analog input. The value should be in ft3/hr for US unit or m^3/hr for Metric unit.	Turbine_ GM_ MeterRun_ V2
LowFlowCutOff	REAL	Low flow cutoff value checks the no flow condition in the calculations. If the flow is s less than this number, it will be considered as no flow condition. Unit is m3/hr for Metric unit, ft3/hr for US unit.	Turbine_ GM_ MeterRun_ V2
Pulse	UDINT	Pulse counter value	Turbine_ GM_ MeterRun_ V2
LowPulseCutOff	UINT	Low pulse cutoff value checks the no flow condition in the calculations. If the Pulse increment is less than this	Turbine_ GM_ MeterRun_ V2

Input Parameter	Data types	Description	Apply to
		number, it will be considered as no flow condition.	
MeterCalFactor	REAL	It is same as Meter K Factor which converts pulse from flow meter into volume. The value should be in pulses/m3 or pulses/ft3. When the flow type is analog, this is the correction factor to apply for volume calculation and the default value should be 1.0.	All
MeterRunId	INT	This is an integer number that represents a configured meter run identifier.	All
GasCompFormat	INT	This parameter is for the gas composition format. It should be either mole fraction {1} or percentage {2}. **NOTE:** It is recommended to use 2 percentage as a default option.	All
FlowType	INT	This parameter represents the flow type, it should be either {1} Pulse Accumulated or {2} Analog Flow Rate.	All
InputUnit	INT	This parameter is for all the inputs of meter run function block. It should be either US {1} or Metric {2}.	All
ContractUnit	INT	This parameter is for all the	All

Input Parameter	Data types	Description	Apply to
		outputs of meter run function block. It should be either US {1} or Metric {2}.	
ContractStartday	INT	This parameter represents the start of gas QTR day. Its value should be from 0 to 23.	All
AvgMethod	INT	This parameter is for averaging method to be used for averaging. As of now, it only supports value {1} that is for time weighted linear average.	All
MaintMode	INT	Parameter to Start or Stop the Maintenance Mode. The value should be either {0} Maintenance End or {1} for Maintenance Start. When Maintenance mode is started, an independent totalizer will be started and accumulate all the flow until the maintenance mode is stopped. During this period, non-resettable totals, hourly/daily/batch totals will be not incremented.	Turbine_ GM_ MeterRun_ V2
AGA8Version	INT	Selection for AGA 8 algorithm selection: 1- AGA 8 (1994) 2 - AGA 8 (2017)	Turbine_ GM_ MeterRun_ V2
AtmosphericPressure	REAL	Atmospheric pressure should be in Psia for US unit system and in Kpa for Metric unit system. Atmospheric pressure is	All

Input Parameter	Data types	Description	Apply to
		used to make Flowing pressure absolute when flowing pressure is measured by a pressure gauge. If flowing pressure is already absolute then it can be left zero.	
UserDefined1	REAL	This parameter is an optional one, if user wants to average some parameter then user can use it, it will be averaged and logged in the QTR.	All
UserDefined2	REAL	This parameter is an optional one, if user wants to average some parameter then user can use it, it will be averaged and logged in the QTR.	All
UserDefined3	REAL	This parameter is an optional one, if user wants to average some parameter then user can use it, it will be averaged and logged in the QTR.	All
UserDefined4	REAL	This parameter is an optional one, if user wants to average some parameter then user can use it, it will be averaged and logged in the QTR.	All

Output

Output Parameter	Data types	Description	All
Out_Code	INT	This out parameter returns	All

Output Parameter	Data types	Description	All
		success or fail code.	
GasCompAtBaseCond	LREAL	This parameter is gas compressibility factor at base condition. It is calculated in AGA8 Detailed method. It is unit less.	All
GasDensityAtBaseCond	LREAL	This parameter is gas density at base condition. It is calculated through AGA8 Detailed method. It is in lbm/ft^3 for US unit system and in kg/m^3 for Metric unit system.	All
GasRelDenAtBaseCond	LREAL	This parameter is gas relative density at base condition. It is calculated through AGA8 Detailed method. It is unit less.	All
GasMolecularWeight	LREAL	This parameter is gas molecular weight at base condition. It is calculated through AGA8 Detailed method. It is in lbm for US unit system and in kg for Metric unit system.	All
UncorrectedFlow	LREAL	This parameter is uncorrected flow rate. It is in ft^3/hr for US unit system and in m^3/hr for Metric unit system in case FlowType= 2 (Analog). For FlowType = 1 (Pulse), this parameter represents instantaneous volume flow from the last time flow is calculated. Unit is ft3/task interval (US) or m3/task interval (Metric).	All
Aga7QM	LREAL	This parameter is gas mass flow rate. It is calculated	All

Output Parameter	Data types	Description	All
		through AGA7 method. It is in lbm/hr for US unit system and in kg/hr for Metric unit system in case FlowType= 2 (Analog). For FlowType = 1 (Pulse), this parameter represents instantaneous mass flow from the last time mass flow is calculated. Unit is lbm/task interval (US) or kg/task interval (Metric).	
Aga7QB	LREAL	This parameter is volume flow rate at base condition. It is calculated through AGA7 method. It is in ft^3/hr for US unit system and in m^3/hr for Metric unit system in case FlowType= 2 (Analog). For FlowType = 1 (Pulse), this parameter represents instantaneous base volume flow from the last time flow is calculated. Unit is ft3/task interval (US) or m3/task interval (Metric).	All
Energy	LREAL	This parameter is gas energy per hour. It is calculated through AGA5 method. It is in BTU/hr for US unit system and in MJ/hr for Metric unit system in case FlowType= 2 (Analog). For FlowType = 1 (Pulse), this parameter represents instantaneous energy flow from the last time flow is calculated. Unit is BTU/task interval (US) or MJ/task interval (Metric).	All
PrevHrAvgTemp	LREAL	This parameter is previous hour average for temperature.	All

Output Parameter	Data types	Description	All
PreDayAvgTemp	LREAL	This parameter is previous day average for temperature.	All
PrevHrAvgPressure	LREAL	This parameter is previous hour average for pressure.	All
PreDayAvgPressure	LREAL	This parameter is previous day average for pressure.	All
PrevHrAvgPulse	LREAL	This parameter is previous hour average for analog input.	All
PreDayAvgPulse	LREAL	This parameter is previous day average for analog input.	All
PrevHrAvgDenAtBase	LREAL	This parameter is previous hour average for density at base condition.	All
PreDayAvgDenAtBase	LREAL	This parameter is previous day average for density at base condition.	All
PrevHrAvgRelDenAtBase	LREAL	This parameter is previous hour average for relative density at base condition.	All
PreDayAvgRelDenAtBase	LREAL	This parameter is previous day average for relative density at base condition.	All
PrevHrAvgUncorrFlow	LREAL	This parameter is previous hour average for uncorrected flow.	All
PreDayAvgUncorrFlow	LREAL	This parameter is previous day average for uncorrected flow.	All
PrevHrAvgUserDefined1	LREAL	This parameter is previous hour average for user defined parameter1.	All
PreDayAvgUserDefined1	LREAL	This parameter is previous day average for user defined parameter1.	All
PrevHrAvgUserDefined2	LREAL	This parameter is previous hour average for user defined	All

Output Parameter	Data types	Description	All
		parameter2.	
PreDayAvgUserDefined2	LREAL	This parameter is previous day average for user defined parameter2.	All
PrevHrAvgUserDefined3	LREAL	This parameter is previous hour average for user defined parameter3.	All
PreDayAvgUserDefined3	LREAL	This parameter is previous day average for user defined parameter3.	All
PrevHrAvgUserDefined4	LREAL	This parameter is previous hour average for user defined parameter4.	All
PreDayAvgUserDefined4	LREAL	This parameter is previous day average for user defined parameter4.	All
QbTH	LREAL	This parameter is volume flow rate at base condition total for this hour.	All
QbLH	LREAL	This parameter is volume flow rate at base condition total for last hour.	All
QbTD	LREAL	This parameter is volume flow rate at base condition total for this day.	All
QbLD	LREAL	This parameter is volume flow rate at base condition total for last day.	All
MTH	LREAL	This parameter is mass flow rate total for this hour.	All
MLH	LREAL	This parameter is mass flow rate total for last hour.	All
MTD	LREAL	This parameter is mass flow rate total for this day.	All

Output Parameter	Data types	Description	All
MLD	LREAL	This parameter is mass flow rate total for last day.	All
ETH	LREAL	This parameter is energy total for this hour.	All
ELH	LREAL	This parameter is energy total for last hour.	All
ETD	LREAL	This parameter is energy total for this day.	All
ELD	LREAL	This parameter is energy total for last day.	All
QbNR	LREAL	Non-Resettable or Cumulative total for volume at Base. Unit - ft3/hr for US, m3/hr for Metric.	Turbine_GM_MeterRun_V2
MNR	LREAL	Non-Resettable or Cumulative total for Mass. Unit - lbm/hr for US, kg/hr for Metric.	Turbine_GM_MeterRun_V2
ENR	LREAL	Non-Resettable or Cumulative total for Energy. Unit -Btu/hr for US, MJ/hr for Metric.	Turbine_GM_MeterRun_V2
QbRollover	INT	Rollover flag for volume at base condition non-resettable total. The value {1} indicates Rollover otherwise {0}. The value for roll-over is 999,999,999. The rollover flag will be on 5 times interval time.	Turbine_GM_MeterRun_V2
MRollover	INT	Rollover flag for Non-Resettable Mass total. The value {1} indicates Rollover otherwise {0}. The value for roll-over is 999,999,999. The rollover flag will be on 5 times interval time.	Turbine_GM_MeterRun_V2

Output Parameter	Data types	Description	All
ERollover	INT	Rollover flag for Non-Resettable Energy total. The value {1} indicates Rollover otherwise {0}. The value for roll-over is 999,999,999. The rollover flag will be on 5 times interval time.	Turbine_ GM_ MeterRun_ V2
QbMaint	LREAL	Volume at Base in Maintenance mode.	Turbine_ GM_ MeterRun_ V2
MMaint	LREAL	Mass in Maintenance mode.	Turbine_ GM_ MeterRun_ V2
EMaint	LREAL	Energy at Base in Maintenance mode.	Turbine_ GM_ MeterRun_ V2

NOTE: The above outputs including averages and totals would be in the contract unit. The QTR generated by this function block contains following fields. Date; Time; Flow Time; Volume at Base; Mass; Energy; Temperature; Pressure; Pulse; Density; Uncorrected flow; Relative Density; Average User Defined 1 (optional); Average User Defined 2 (optional); Average User Defined 3 (optional); Average User Defined 4 (optional).

Following are the error codes for the above meter run function block.

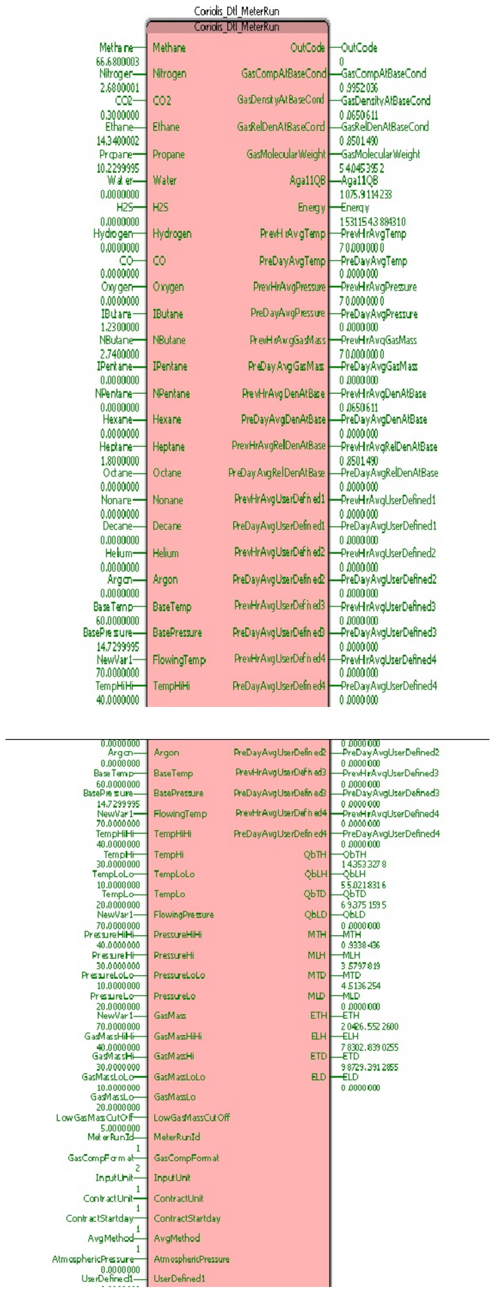
Out Code	Description	Apply to
0	SUCCESS	All
5	ERROR: THE ROOT WAS NOT BOUNDED IN DGROSS	All
6	ERROR: NO CONVERGENCE IN DGROSS	All
7	ERROR: VIRGS SQURE ROOT NEGATIVE	All

Out Code	Description	Apply to
8	ERROR: COMBINED VALUES OF GRGR, X[2] AND HV NOT CONSISTENT	All
9	ERROR: INVALID TERM IN VIRGS	All
11	ERROR: METHOD WAS NOT 1 OR 2	All
12	ERROR: FLOWING PRESSURE (PF) <= 0.0 OR > 1740.0 PSIA	All
13	ERROR: FLOWING TEMPERATURE (TF) < 14.0 OR > 149.0 DEG F	All
14	ERROR: HEATING VALUE (HV) < 477.0 OR > 1211.0 BTU/FT^3	All
15	ERROR: GAS RELATIVE DENSITY (GRGR) < 0.55 OR > 0.870	All
16	ERROR: MOLE FRACTION FOR N2 < 0.0 OR > 0.50 OR FOR CO2 < 0.0 OR > 0.30 OR FOR H2 < 0.0 OR > 0.10 OR FOR CO < 0.0 OR > 0.03	All
17	ERROR: REFERENCE TEMPERATURE < 32.0 OR > 77.0 DEG F	All
18	ERROR: REFERENCE PRESSURE < 13.0 OR > 16.0 PSIA	All
22	WARNING: FLOWING PRESSURE (PF) <= 0.0 OR > 1200.0 PSIA	Turbine_GM_MeterRun
23	WARNING: FLOWING TEMPERATURE (TF) < 32.0 OR > 130.0 DEG F	Turbine_GM_MeterRun
24	WARNING: HEATING VALUE (HV) < 805.0 OR > 1208.0 BTU/FT^3	Turbine_GM_MeterRun
25	WARNING: GAS RELATIVE DENSITY (GRGR) < 0.55 OR > 0.800	Turbine_GM_MeterRun
26	WARNING: MOLE FRACTION FOR N2 < 0.0 OR > 0.20 OR FOR CO2 < 0.0 OR > 0.20 OR FOR H2 < 0.0 OR > 0.0 OR FOR CO < 0.0 OR >0.0	Turbine_GM_MeterRun
81	WARNING: FLOWING PRESSURE (PF) > 1500.0 PSIA AGA8 2017 RANGE 1	Turbine_GM_

Out Code	Description	Apply to
		MeterRun_ V2
82	WARNING: FLOWING TEMPERATURE (TF) < 17.01 OR > 143.0 DEG F AGA8 2017 RANGE 2 OR (TF) < 25.0 OR > 143.0 DEG F AGA8 2017 RANGE 1	Turbine_ GM_ MeterRun_ V2
83	WARNING: HEATING VALUE (HV) < 665.0 OR > 1100.0 BTU/FT^3 AGA8 2017 RANGE 2 OR (HV) < 930.0 OR > 1040.0 BTU/FT^3 AGA8 2017 RANGE 1	Turbine_ GM_ MeterRun_ V2
84	WARNING: GAS RELATIVE DENSITY (GRGR) < 0.554 OR > 0.801 AGA8 2017 RANGE 2 OR (GRGR) < 0.554 OR > 0.630 AGA8 RANGE 1	Turbine_ GM_ MeterRun_ V2
85	WARNING: MOLE FRACTION FOR N2 > 0.20 AGA8 2017 RANGE 2 OR N2 > 0.07 AGA8 2017 RANGE 1 OR FOR CO2 > 0.25 AGA8 2017 RANGE 2 OR CO2 > 0.03 AGA8 2017 RANGE 1 OR FOR H2 < 0.0 OR > 0.0 AGA8 2017 RANGE 1 AND 2 OR FOR CO < 0.0 OR > 0.0 AGA8 2017 RANGE 1 AND 2	Turbine_ GM_ MeterRun_ V2

Coriolis_Dtl_MeterRun Function Block

Here is an example for Coriolis_Dtl_MeterRun:



Description

This Coriolis_Dtl_MeterRun function block calculates gas compressibility factor, density, relative density and molecular weight from AGA8 detailed method, volume flow rate at base condition from AGA11 and gas energy per hour from AGA5. It also calculates hourly and daily averages and totals. It generates hourly & daily QTRs and sends them to EFM application which logs them on he contoller s MRAM and flash memory. It also generates alarms when any of the process value crosses specified alarm limit.

Coriolis_Dtl_MeterRun expects the input parameters to be in US or Metric unit system.

Input

Input Parameter	Data types	Description	Apply to
Methane	REAL	It could be in mole fraction or percentage.	All
Nitrogen	REAL	It could be in mole fraction or percentage.	All
CO2	REAL	It could be in mole fraction or percentage.	All
Ethane	REAL	It could be in mole fraction or percentage.	All
Propane	REAL	It could be in mole fraction or percentage.	All
Water	REAL	It could be in mole fraction or percentage.	All
H2S	REAL	It could be in mole fraction or percentage.	All
Hydrogen	REAL	It could be in mole fraction or percentage.	All
CO	REAL	It could be in mole fraction or percentage.	All
Oxygen	REAL	It could be in mole fraction or percentage.	All

Input Parameter	Data types	Description	Apply to
IButane	REAL	It could be in mole fraction or percentage.	All
NButane	REAL	It could be in mole fraction or percentage.	All
IPentane	REAL	It could be in mole fraction or percentage.	All
NPentane	REAL	It could be in mole fraction or percentage.	All
Hexane	REAL	It could be in mole fraction or percentage.	All
Heptane	REAL	It could be in mole fraction or percentage.	All
Octane	REAL	It could be in mole fraction or percentage.	All
Nonane	REAL	It could be in mole fraction or percentage.	All
Decane	REAL	It could be in mole fraction or percentage.	All
Helium	REAL	It could be in mole fraction or percentage.	All
Argon	REAL	It could be in mole fraction or percentage.	All
DetailMethod	INT	Selection for Detail method: 1- Detail Method 2- GERG Method. GERG Method is only applicable for V2 function block.	Coriolis_Dtl_MeterRun_V2
BaseTemp	REAL	Base temperature should be in Fahrenheit for US unit system and in Celcius for Metric unit system. The recommended default is 60 Deg F.	All
BasePressure	REAL	Base pressure should be in Psia for US unit system and in Kpa for Metric unit system. The recommended	All

Input Parameter	Data types	Description	Apply to
		default is 14.73 Psia.	
FlowingTemp	REAL	Flowing temperature should be in Fahrenheit for US unit system and in Celcius for Metric unit system.	All
TempHiHi	REAL	This is the HiHi limit for flowing temperature. It should be either in Fahrenheit or Celcius.	All
TempHi	REAL	This is the Hi limit for flowing temperature. It should be either in Fahrenheit or Celcius.	All
TempLoLo	REAL	This is the LoLo limit for flowing temperature. It should be either in Fahrenheit or Celcius.	All
TempLo	REAL	This is the Lo limit for flowing temperature. It should be either in Fahrenheit or Celcius.	All
FlowTempIOSelection	INT	IO selection for meter temperature. The value should be {1} for Live or {2} for Keypad value.	Coriolis_Dtl_MeterRun_V2
FlowTempStsStatus	USINT	Analog input channel status for meter temperature. The value should be {0} for Good or any positive integer for bad status.	Coriolis_Dtl_MeterRun_V2
FlowTempKeypadVal	REAL	Keypad value for meter temperature. The value that should be used when the meter temperature status is bad.	Coriolis_Dtl_MeterRun_V2
FlowingPressure	REAL	Flowing pressure should be in Psia for US unit system and in Kpa for Metric unit system.	All
PressureHiHi	REAL	This is the HiHi limit for flowing pressure. It should be either in Psia or Kpa.	All
PressureHi	REAL	This is the Hi limit for flowing pressure. It should be either in Psia or Kpa.	All

Input Parameter	Data types	Description	Apply to
PressureLoLo	REAL	This is the LoLo limit for flowing pressure. It should be either in Psia or Kpa.	All
PressureLo	REAL	This is the Lo limit for flowing pressure. It should be either in Psia or Kpa.	All
GasMass	REAL	This parameter is gas mass that is directly measured from coriolis meter. It should be in lbm/hr for US unit system and in kg/hr for Metric unit system.	Coriolis_Dtl_MeterRun
GasMassHiHi	REAL	This is the HiHi limit for gas mass. It should be in lbm/hr for US unit system and in kg/hr for Metric unit system.	Coriolis_Dtl_MeterRun
GasMassHi	REAL	This is the Hi limit for gas mass. It should be in lbm/hr for US unit system and in kg/hr for Metric unit system.	Coriolis_Dtl_MeterRun
GasMassLoLo	REAL	This is the LoLo limit for gas mass. It should be in lbm/hr for US unit system and in kg/hr for Metric unit system.	Coriolis_Dtl_MeterRun
GasMassLo	REAL	This is the Lo limit for gas mass. It should be in lbm/hr for US unit system and in kg/hr for Metric unit system.	Coriolis_Dtl_MeterRun
LowGasMassCutOff	REAL	This is the low gas mass cut off limit. It should be in lbm/hr for US unit system and in kg/hr for Metric unit system.	Coriolis_Dtl_MeterRun
FlowPresIOSelection	INT	IO selection for meter pressure. The value should be {1} for Live or {2} for Keypad value.	Coriolis_Dtl_MeterRun_V2
FlowPresStsStatus	USINT	Analog input channel status for meter pressure. The value should be {0} for Good or any positive integer	Coriolis_Dtl_MeterRun_V2

Input Parameter	Data types	Description	Apply to
		for Bad status.	
FlowPresKeypadVal	REAL	Analog input channel status for meter pressure. The value should be {0} for Good or any positive integer for Bad status.	Coriolis_Dtl_MeterRun_V2
Analog	REAL	Value of analog input if flow type = analog. The value should be in lb/hr for US unit or kg/hr for Metric unit.	Coriolis_Dtl_MeterRun_V2
AnalogHiHi	REAL	This is the HiHi limit for analog input. The value should be in lb/hr for US unit or kg/hr for Metric unit.	Coriolis_Dtl_MeterRun_V2
AnalogHi	REAL	This is the Hi limit for analog input. The value should be in lb/hr for US unit or kg/hr for Metric unit.	Coriolis_Dtl_MeterRun_V2
AnalogLoLo	REAL	This is the LoLo limit for analog input. The value should be in lb/hr for US unit or kg/hr for Metric unit.	Coriolis_Dtl_MeterRun_V2
AnalogLo	REAL	This is the Lo limit for analog input. The value should be in lb/hr for US unit or kg/hr for Metric unit.	Coriolis_Dtl_MeterRun_V2
LowFlowCutOff	REAL	Low flow cutoff value checks the no flow condition in the calculations. If the flow is s less than this number, it will be considered as no flow condition. Unit is kg/hr for Metric unit, lb/hr for US unit.	Coriolis_Dtl_MeterRun_V2
FlowType	INT	This parameter represents the flow type, it should be either {1} Pulse Accumulated or {2} Analog Flow Rate.	Coriolis_Dtl_MeterRun_V2
Pulse	UDINT	Pulse counter value	Coriolis_Dtl_MeterRun_V2
LowPulseCutOff	UINT	Low pulse cutoff value checks the no flow condition in the calculations. If	Coriolis_Dtl_MeterRun_

Input Parameter	Data types	Description	Apply to
		the Pulse increment is less than this number, it will be considered as no flow condition.	V2
MeterCalFactor	REAL	It is same as Meter K factor which converts pulse from flow meter into mass. The value should be in pulses/kg (Metric) or pulses/lb (US). When the flow type is analog, this is the correction factor to apply for mass calculation and the default value should be 1.0.	Coriolis_DtL_MeterRun_V2
MeterRunId	INT	This is an integer number that represents a configured meter run identifier.	All
GasCompFormat	INT	This parameter is for the gas composition format. It should be either mole fraction {1} or percentage {2}. **NOTE:** It is recommended to use 2 percentage as a default option.	All
InputUnit	INT	This parameter is for all the inputs of meter run function block. It should be either US {1} or Metric {2}.	All
ContractUnit	INT	This parameter is for all the outputs of meter run function block. It should be either US {1} or Metric {2}.	All
ContractStartday	INT	This parameter represents the start of gas QTR day. Its value should be from 0 to 23.	All
AvgMethod	INT	This parameter is for averaging method to be used for averaging. As of now, it only supports value {1} that is for time weighted linear average.	All
MaintMode	INT	Parameter to Start or Stop the	Coriolis_DtL_

Input Parameter	Data types	Description	Apply to
		Maintenance Mode. The value should be either {0} Maintenance End or {1} for Maintenance Start. When Maintenance mode is started, an independent totalizer will be started and accumulate all the flow until the maintenance mode is stopped. During this period, non-resettable totals, hourly/daily/batch totals will be not incremented.	MeterRun_ V2
AGA8Version	INT	Selection for AGA 8 algorithm selection: 1- AGA 8 (1994) 2 - AGA 8 (2017)	Coriolis_Dtl_ MeterRun_ V2
AtmosphericPressure	REAL	Atmospheric pressure should be in Psia for US unit system and in Kpa for Metric unit system. Atmospheric pressure is used to make Flowing pressure absolute when flowing pressure is measured by a pressure gauge. If flowing pressure is already absolute then it can be left zero.	All
UserDefined1	REAL	This parameter is an optional one, if user wants to average some parameter then user can use it, it will be averaged and logged in the QTR.	All
UserDefined2	REAL	This parameter is an optional one, if user wants to average some parameter then user can use it, it will be averaged and logged in the QTR.	All
UserDefined3	REAL	This parameter is an optional one, if user wants to average some parameter then user can use it, it will be averaged and logged in the QTR.	All

Input Parameter	Data types	Description	Apply to
UserDefined4	REAL	This parameter is an optional one, if user wants to average some parameter then user can use it, it will be averaged and logged in the QTR.	All

Output

Output Parameter	Data types	Description	Apply to
Out_Code	INT	This out parameter returns success or fail code.	All
GasCompAtBaseCond	LREAL	This parameter is gas compressibility factor at base condition. It is calculated in AGA8 Detailed method. It is unit less.	All
GasDensityAtBaseCond	LREAL	This parameter is gas density at base condition. It is calculated through AGA8 Detailed method. It is in lbm/ft^3 for US unit system and in kg/m^3 for Metric unit system.	All
GasRelDenAtBaseCond	LREAL	This parameter is gas relative density at base condition. It is calculated through AGA8 Detailed method. It is unit less.	All
GasMolecularWeight	LREAL	This parameter is gas molecular weight at base condition. It is calculated through AGA8 Detailed method. It is in lbm for US unit system and in kg for Metric unit system.	All
Aga11QB	LREAL	This parameter is volume flow trate at base condition. It is calculated through AGA11 method. It is in ft^3/hr for US unit system and in m^3/hr for	All

Output Parameter	Data types	Description	Apply to
		Metric unit system in case FlowType= 2 (Analog). For FlowType = 1 (Pulse), this parameter represents instantaneous base volume flow from the last time flow is calculated. Unit is ft3/task interval (US) or m3/task interval (Metric).	
Energy	LREAL	This parameter is gas energy per hour. It is calculated through AGA5 method. It is in BTU/hr for US unit system and in MJ/hr for Metric unit system in case FlowType= 2 (Analog). For FlowType = 1 (Pulse), this parameter represents instantaneous energy flow from the last time flow is calculated. Unit is BTU/task interval (US) or MJ/task interval (Metric).	All
PrevHrAvgTemp	LREAL	This parameter is previous hour average for temperature.	All
PreDayAvgTemp	LREAL	This parameter is previous day average for temperature.	All
PrevHrAvgPressure	LREAL	This parameter is previous hour average for pressure.	All
PreDayAvgPressure	LREAL	This parameter is previous day average for pressure.	All
PrevHrAvgGasMass	LREAL	This parameter is previous hour average for gas mass.	All
PreDayAvgGasMass	LREAL	This parameter is previous day average for gas mass.	All
PrevHrAvgDenAtBase	LREAL	This parameter is previous hour average for density at base condition.	All

Output Parameter	Data types	Description	Apply to
PreDayAvgDenAtBase	LREAL	This parameter is previous day average for density at base condition.	All
PrevHrAvgRelDenAtBase	LREAL	This parameter is previous hour average for relative density at base condition.	All
PreDayAvgRelDenAtBase	LREAL	This parameter is previous day average for relative density at base condition.	All
PrevHrAvgUserDefined1	LREAL	This parameter is previous hour average for user defined parameter1.	All
PreDayAvgUserDefined1	LREAL	This parameter is previous day average for user defined parameter1.	All
PrevHrAvgUserDefined2	LREAL	This parameter is previous hour average for user defined parameter2.	All
PreDayAvgUserDefined2	LREAL	This parameter is previous day average for user defined parameter2.	All
PrevHrAvgUserDefined3	LREAL	This parameter is previous hour average for user defined parameter3.	All
PreDayAvgUserDefined3	LREAL	This parameter is previous day average for user defined parameter3.	All
PrevHrAvgUserDefined4	LREAL	This parameter is previous hour average for user defined parameter4.	All
PreDayAvgUserDefined4	LREAL	This parameter is previous day average for user defined parameter4.	All
QbTH	LREAL	This parameter is volume flow rate at base condition total for	All

Output Parameter	Data types	Description	Apply to
		this hour.	
QbLH	LREAL	This parameter is volume flow rate at base condition total for last hour.	All
QbTD	LREAL	This parameter is volume flow rate at base condition total for this day.	All
QbLD	LREAL	This parameter is volume flow rate at base condition total for last day.	All
MTH	LREAL	This parameter is mass flow rate total for this hour.	All
MLH	LREAL	This parameter is mass flow rate total for last hour.	All
MTD	LREAL	This parameter is mass flow rate total for this day.	All
MLD	LREAL	This parameter is mass flow rate total for last day.	All
ETH	LREAL	This parameter is energy total for this hour.	All
ELH	LREAL	This parameter is energy total for last hour.	All
ETD	LREAL	This parameter is energy total for this day.	All
ELD	LREAL	This parameter is energy total for last day.	All
QbNR	LREAL	Non-Resettable or Cumulative total for volume at Base. Unit - ft3/hr for US, m3/hr for Metric.	Coriolis_DtL_MeterRun_V2
MNR	LREAL	Non-Resettable or Cumulative total for Mass. Unit - lbm/hr for US, kg/hr for Metric.	Coriolis_DtL_MeterRun_V2
ENR	LREAL	Non-Resettable or Cumulative	Coriolis_DtL_

Output Parameter	Data types	Description	Apply to
		total for Energy. Unit -Btu/hr for US, MJ/hr for Metric.	MeterRun_ V2
QbRollover	INT	Rollover flag for volume at base condition non-resettable total. The value {1} indicates Rollover otherwise {0}. The value for roll-over is 999,999,999. The rollover flag will be on 5 times interval time.	Coriolis_Dtl_ MeterRun_ V2
MRollover	INT	Rollover flag for Non-Resettable Mass total. The value {1} indicates Rollover otherwise {0}. The value for roll-over is 999,999,999. The rollover flag will be on 5 times interval time.	Coriolis_Dtl_ MeterRun_ V2
ERollover	INT	Rollover flag for Non-Resettable Energy total. The value {1} indicates Rollover otherwise {0}. The value for roll-over is 999,999,999. The rollover flag will be on 5 times interval time.	Coriolis_Dtl_ MeterRun_ V2
QbMaint	LREAL	Volume at Base in Maintenance mode.	Coriolis_Dtl_ MeterRun_ V2
MMaint	LREAL	Mass in Maintenance mode.	Coriolis_Dtl_ MeterRun_ V2
EMaint	LREAL	Energy at Base in Maintenance mode.	Coriolis_Dtl_ MeterRun_ V2
GERG2008CV	LREAL	Heat Capacity at Constant Volume (J/mol K).	Coriolis_Dtl_ MeterRun_ V2
GERG2008CP	LREAL	Heat Capacity at Constant Pressure (J/mol K).	Coriolis_Dtl_ MeterRun_ V2

Output Parameter	Data types	Description	Apply to
GERG2008W	LREAL	Speed of sound in gas being measured. Unit - ft/sec for US, meter/sec for Metric.	Coriolis_DtL_MeterRun_V2

NOTE: The above outputs including averages and totals would be in the contract unit. The QTR generated by this function block contains following fields. Date; Time; Flow Time; Volume at Base; Mass; Energy; Temperature; Gas mass; Density; None; Relative Density; Average User Defined 1 (optional); Average User Defined 2 (optional); Average User Defined 3 (optional); Average User Defined 4 (optional).

Following are the error codes for the above meter run function block.

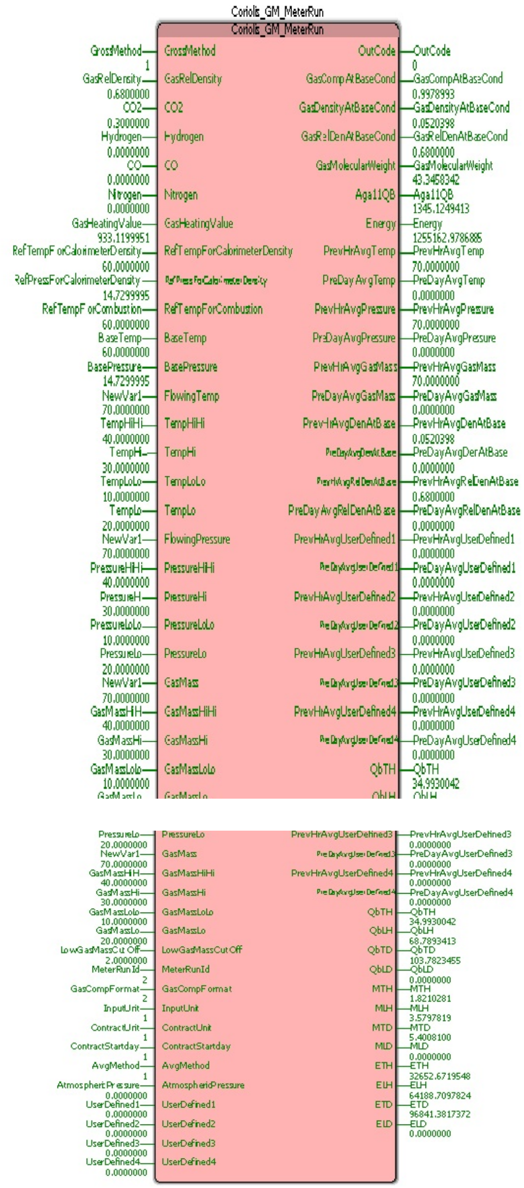
Out Code	Description	Apply to
0	SUCCESS	All
1[1]	ERROR: PRESSURE HAS A NEGATIVE DERIVATIVE DEFAULT GAS DENSITY USED	All
	ERROR: A COMPONENT MOLE FRACTION < 0.0 OR > 1.0	Coriolis_DtL_MeterRun _V2
2[1]	WARNING: DENSITY IN BRAKET EXCEEDS MAXIMUM DEFAULT PROCEEDURE USED	All
	WARNING: SUM OF MOLE FRACTIONS < 0.9999 OR > 1.0001	Coriolis_DtL_MeterRun _V2
3[1]	ERROR: MAXIMUM ITERATIONS EXCEEDED IN BRAKET DEFAULT DENSITY USED	All
	WARNING: PRESSURE BASE (PB) <= 0.0 OR >= 16 PSIA	Coriolis_DtL_MeterRun _V2
4[1]	ERROR: MAXIMUM ITERATIONS IN DDETAIL EXCEEDED LAST DENSITY USED	All
	WARNING: TEMPERATURE BASE (TB) <= 32.0 OR >= 77.0 DEG F	Coriolis_DtL_MeterRun _

Out Code	Description	Apply to
		V2
32	ERROR: FLOWING PRESSURE (PF) <= 0.0 OR > 40,000. PSIA	All
33	ERROR: FLOWING TEMPERATURE (TF) < -200 OR > 760 DEG F	All
36	ERROR: MOLE FRACTION FOR METHANE < 0.0 OR > 1.0 FOR NITROGEN < 0.0 OR > 1.0 FOR CARBON DIOXIDE < 0.0 OR > 1.0 FOR ETHANE < 0.0 OR > 1.0 FOR PROPANE < 0.0 OR > 0.12 FOR WATER < 0.0 OR > 0.10 FOR H2S < 0.0 OR > 1.0 FOR HYDROGEN < 0.0 OR > 1.0 FOR CARBON MONOXIDE < 0.0 OR > 0.03 FOR OXYGEN < 0.0 OR > 0.21 FOR BUTANES < 0.0 OR > 0.06 FOR PENTANES < 0.0 OR > 0.04 FOR HEXANES + < 0.0 OR > 0.10 FOR HELIUM < 0.0 OR > 0.03 FOR ARGON < 0.0 OR > 1.0	Coriolis_Dtl_ MeterRun
37	ERROR: REFERENCE TEMPERATURE < 32.0 OR > 77.0 DEG F	All
38	ERROR: REFERENCE PRESSURE < 12.9 OR > 16.01 PSIA	All
39	ERROR: SUM OF MOLE FRACTIONS < 0.98 OR > 1.020	All
42	WARNING: FLOWING PRESSURE (PF) < 0.0 OR > 1750. PSIA	All
43	WARNING: FLOWING TEMPERATURE (TF) < 17 OR > 143 DEG F	All
45	WARNING: ANY COMPONENT MOLE FRACTION OUTSIDE OF AGA REPORT NO. 8 RECOMMENDED RANGE	Coriolis_Dtl_ MeterRun _ V2
46	WARNING: MOLE FRACTION FOR METHANE < 0.45 OR > 1.0 FOR NITROGEN < 0.0 OR > 0.5 FOR CARBON DIOXIDE < 0.0 OR > 0.3 FOR ETHANE < 0.0 OR > 0.1 FOR PROPANE < 0.0 OR > 0.04 FOR WATER < 0.0 OR >= 0.0005 FOR H2S < 0.0 OR > 0.0002 FOR HYDROGEN < 0.0 OR > 0.1 FOR CARBON MONOXIDE < 0.0 OR > 0.03 FOR OXYGEN < 0.0 OR > 0.0	Coriolis_Dtl_ MeterRun

Out Code	Description	Apply to
	FOR BUTANES < 0.0 OR > 0.01 FOR PENTANES < 0.0 OR >= 0.003 FOR HEXANES + < 0.0 OR >= 0.002 FOR HELIUM < 0.0 OR >= 0.002 FOR ARGON < 0.0 OR > 0.0	
49	WARNING: SUM OF MOLE FRACTIONS < 0.9999 OR > 1.0001	All
86	WARNING: Flowing Pressure greater than 2017 AGA8 GERG-2008 Full Quality Range (10,150 PSIA)	Coriolis_Dtl_MeterRun_V2
87	WARNING: Flowing Pressure greater than 2017 AGA8 GERG-2008 Range (5075 PSIA)	Coriolis_Dtl_MeterRun_V2
88	WARNING: Flowing Temperature outside 2017 AGA8 GERG-2008 Full Quality Range (-352 F < TF < 800 F)	Coriolis_Dtl_MeterRun_V2
89	WARNING: Flowing Temperature outside 2017 AGA8 GERG-2008 Range (-298 F < TF < 350 F)	Coriolis_Dtl_MeterRun_V2
90	WARNING: A Component Mole % outside 2017 AGA8 GERG-2008 Intermediate Quality Range	Coriolis_Dtl_MeterRun_V2
91	WARNING: A Component Mole % outside 2017 AGA8 GERG-2008 Pipeline Quality Range	Coriolis_Dtl_MeterRun_V2
NOTE 1: Error codes 1~4 are common between AGA 8 and AGA 5. You must take caution and analyze when these specific out codes appear to determine the source.		

Coriolis_GM_MeterRun Function Block

Here is an example for Coriolis_GM_MeterRun:



Description

This Coriolis_GM_MeterRun function block calculates gas compressibility factor, density, relative density and molecular weight from AGA8 gross method, volume flow rate at base condition from AGA11 and gas energy per hour from AGA5. It also calculates hourly and daily averages and totals. It generates hourly & daily QTRs and sends them to EFM application which logs them on he contoller's MRAM and flash memory. It also generates alarms when any of the process value crosses specified alarm limit.

Coriolis_GM_MeterRun expects the input parameters to be in US or Metric unit system.

Input

Input Parameter	Data types	Description	Apply to
GrossMethod	INT	This parameter represents gross method number. It should be either {1} for gross method 1 and {2} for gross method 2.	All
GasRelDensity	REAL	This parameter is gas relative density at reference condition. It is unit less.	All
CO2	REAL	It could be in mole fraction or percentage.	All
Hydrogen	REAL	It could be in mole fraction or percentage.	All
CO	REAL	It could be in mole fraction or percentage.	All
Nitrogen	REAL	It could be in mole fraction or percentage. This parameter is only required for gross method 2, for gross method 1, it can be zero.	All
GasHeatingValue	REAL	This parameter is gas heating value. It is only required for gross method 1, for gross method 2, it can be zero. It is in Btu/ft^3 for US unit system and in MJ/m^3 for Metric unit system.	All

Input Parameter	Data types	Description	Apply to
RefTempForCalorimeterDensity	REAL	This parameter is reference temperature for calorimeter density. It should be in Fahrenheit for US unit system and in Celcius for Metric unit system. The recommended default is 60 Deg F.	All
RefPressForCalorimeterDensity	REAL	This parameter is reference pressure for calorimeter density. It should be in Psia for US unit system and in Kpa for Metric unit system. The recommended default is 14.73 Psia.	All
RefTempForCombustion	REAL	This parameter is reference temperature for combustion. It should be in Fahrenheit for US unit system and in Celcius for Metric unit system. The recommended default is 60 Deg F.	All
BaseTemp	REAL	Base temperature should be in Fahrenheit for US unit system and in Celcius for Metric unit system. The recommended default is 60 Deg F.	All
BasePressure	REAL	Base pressure should be in Psia for US unit system and in Kpa for Metric unit system. The recommended default is 14.73 Psia.	All
FlowingTemp	REAL	Flowing temperature should be in Fahrenheit for US unit system and in Celcius for Metric unit system.	All

Input Parameter	Data types	Description	Apply to
TempHiHi	REAL	This is the HiHi limit for flowing temperature. It should be either in Fahrenheit or Celcius.	All
TempHi	REAL	This is the Hi limit for flowing temperature. It should be either in Fahrenheit or Celcius.	All
TempLoLo	REAL	This is the LoLo limit for flowing temperature. It should be either in Fahrenheit or Celcius.	All
TempLo	REAL	This is the Lo limit for flowing temperature. It should be either in Fahrenheit or Celcius.	All
FlowTempIOSelection	INT	IO selection for meter temperature. The value should be {1} for Live or {2} for Keypad value.	Coriolis_ GM_ MeterRun_ V2
FlowTempStsStatus	USINT	Analog input channel status for meter temperature. The value should be {0} for Good or any positive integer for bad status.	Coriolis_ GM_ MeterRun_ V2
FlowTempKeypadVal	REAL	Keypad value for meter temperature. The value that should be used when the meter temperature status is bad.	Coriolis_ GM_ MeterRun_ V2
FlowingPressure	REAL	Flowing pressure should be in Psia for US unit system and in Kpa for Metric unit system.	All
PressureHiHi	REAL	This is the HiHi limit for	All

Input Parameter	Data types	Description	Apply to
		flowing pressure. It should be either in Psia or Kpa.	
PressureHi	REAL	This is the Hi limit for flowing pressure. It should be either in Psia or Kpa.	All
PressureLoLo	REAL	This is the LoLo limit for flowing pressure. It should be either in Psia or Kpa.	All
PressureLo	REAL	This is the Lo limit for flowing pressure. It should be either in Psia or Kpa.	All
GasMass	REAL	This parameter is gas mass that is directly measured from coriolis meter. It should be in lbm/hr for US unit system and in kg/hr for Metric unit system.	Coriolis_ GM_ MeterRun
GasMassHiHi	REAL	This is the HiHi limit for gas mass. It should be in lbm/hr for US unit system and in kg/hr for Metric unit system.	Coriolis_ GM_ MeterRun
GasMassHi	REAL	This is the Hi limit for gas mass. It should be in lbm/hr for US unit system and in kg/hr for Metric unit system.	Coriolis_ GM_ MeterRun
GasMassLoLo	REAL	This is the LoLo limit for gas mass. It should be in lbm/hr for US unit system and in kg/hr for Metric unit system.	Coriolis_ GM_ MeterRun
GasMassLo	REAL	This is the Lo limit for gas mass. It should be in lbm/hr for US unit system and in kg/hr for Metric unit system.	Coriolis_ GM_ MeterRun

Input Parameter	Data types	Description	Apply to
LowGasMassCutOff	REAL	This is the low gas mass cut off limit. It should be in lbm/hr for US unit system and in kg/hr for Metric unit system.	Coriolis_GM_MeterRun
FlowPresIOSelection	INT	IO selection for meter pressure. The value should be {1} for Live or {2) for Keypad value.	Coriolis_GM_MeterRun_V2
FlowPresStsStatus	USINT	Analog input channel status for meter pressure. The value should be {0} for Good or any positive integer for Bad status.	Coriolis_GM_MeterRun_V2
FlowPresKeypadVal	REAL	Analog input channel status for meter pressure. The value should be {0} for Good or any positive integer for Bad status.	Coriolis_GM_MeterRun_V2
Analog	REAL	Value of analog input if flow type = analog. The value should be in lb/hr for US unit or kg/hr for Metric unit.	Coriolis_GM_MeterRun_V2
AnalogHiHi	REAL	This is the HiHi limit for analog input. The value should be in lb/hr for US unit or kg/hr for Metric unit.	Coriolis_GM_MeterRun_V2
AnalogHi	REAL	This is the Hi limit for analog input. The value should be in lb/hr for US unit or kg/hr for Metric unit.	Coriolis_GM_MeterRun_V2
AnalogLoLo	REAL	This is the LoLo limit for analog input. The value should be in lb/hr for US	Coriolis_GM_MeterRun_

Input Parameter	Data types	Description	Apply to
		unit or kg/hr for Metric unit.	V2
AnalogLo	REAL	This is the Lo limit for analog input. The value should be in lb/hr for US unit or kg/hr for Metric unit.	Coriolis_ GM_ MeterRun_ V2
LowFlowCutOff	REAL	Low flow cutoff value checks the no flow condition in the calculations. If the flow is s less than this number, it will be considered as no flow condition. Unit is kg/hr for Metric unit, lb/hr for US unit.	Coriolis_ GM_ MeterRun_ V2
FlowType	INT	This parameter represents the flow type, it should be either {1} Pulse Accumulated or {2} Analog Flow Rate.	Coriolis_ GM_ MeterRun_ V2
Pulse	UDINT	Pulse counter value	Coriolis_ GM_ MeterRun_ V2
LowPulseCutOff	UINT	Low pulse cutoff value checks the no flow condition in the calculations. If the Pulse increment is less than this number, it will be considered as no flow condition.	Coriolis_ GM_ MeterRun_ V2
MeterCalFactor	REAL	It is same as Meter K factor which converts pulse from flow meter into mass. The value should be in pulses/kg (Metric) or	Coriolis_ GM_ MeterRun_ V2

Input Parameter	Data types	Description	Apply to
		pulses/lb (US). When the flow type is analog, this is the correction factor to apply for mass calculation and the default value should be 1.0.	
MeterRunId	INT	This is an integer number that represents a configured meter run identifier.	All
GasCompFormat	INT	This parameter is for the gas composition format. It should be either mole fraction {1} or percentage {2}. **NOTE:** It is recommended to use 2 percentage as a default option.	All
InputUnit	INT	This parameter is for all the inputs of meter run function block. It should be either US {1} or Metric {2}.	All
ContractUnit	INT	This parameter is for all the outputs of meter run function block. It should be either US {1} or Metric {2}.	All
ContractStartday	INT	This parameter represents the start of gas QTR day. Its value should be from 0 to 23.	All
AvgMethod	INT	This parameter is for averaging method to be used for averaging. As of now, it only supports value {1} that is for time weighted	All

Input Parameter	Data types	Description	Apply to
		linear average.	
MaintMode	INT	Parameter to Start or Stop the Maintenance Mode. The value should be either {0} Maintenance End or {1} for Maintenance Start. When Maintenance mode is started, an independent totalizer will be started and accumulate all the flow until the maintenance mode is stopped. During this period, non-resettable totals, hourly/daily/batch totals will be not incremented.	Coriolis_ GM_ MeterRun_ V2
AGA8Version	INT	Selection for AGA 8 algorithm selection: 1- AGA 8 (1994) 2 - AGA 8 (2017)	Coriolis_ GM_ MeterRun_ V2
AtmosphericPressure	REAL	Atmospheric pressure should be in Psia for US unit system and in Kpa for Metric unit system. Atmospheric pressure is used to make Flowing pressure absolute when flowing pressure is measured by a pressure gauge. If flowing pressure is already absolute then it can be left zero.	All
UserDefined1	REAL	This parameter is an optional one, if user wants to average some parameter then user can use it, it will be averaged and logged in	All

Input Parameter	Data types	Description	Apply to
		the QTR.	
UserDefined2	REAL	This parameter is an optional one, if user wants to average some parameter then user can use it, it will be averaged and logged in the QTR.	All
UserDefined3	REAL	This parameter is an optional one, if user wants to average some parameter then user can use it, it will be averaged and logged in the QTR.	All
UserDefined4	REAL	This parameter is an optional one, if user wants to average some parameter then user can use it, it will be averaged and logged in the QTR.	All

Output

Output Parameter	Data types	Description	Apply to
Out_Code	INT	This out parameter returns success or fail code.	All
GasCompAtBaseCond	LREAL	This parameter is gas compressibility factor at base condition. It is calculated in AGA8 Detailed method. It is unit less.	All
GasDensityAtBaseCond	LREAL	This parameter is gas density at base condition. It is calculated through AGA8 Detailed method. It is in lbm/ft^3 for US unit system and in kg/m^3 for Metric unit system.	All

Output Parameter	Data types	Description	Apply to
GasRelDenAtBaseCond	LREAL	This parameter is gas relative density at base condition. It is calculated through AGA8 Detailed method. It is unit less.	All
GasMolecularWeight	LREAL	This parameter is gas molecular weight at base condition. It is calculated through AGA8 Detailed method. It is in lbm for US unit system and in kg for Metric unit system.	All
Aga11QB	LREAL	This parameter is volume flow trate at base condition. It is calculated through AGA11 method. It is in ft^3/hr for US unit system and in m^3/hr for Metric unit system in case FlowType= 2 (Analog). For FlowType = 1 (Pulse), this parameter represents instantaneous base volume flow from the last time flow is calculated. Unit is ft3/task interval (US) or m3/task interval (Metric).	All
Energy	LREAL	This parameter is gas energy per hour. It is calculated through AGA5 method. It is in BTU/hr for US unit system and in MJ/hr for Metric unit system in case FlowType= 2 (Analog). For FlowType = 1 (Pulse), this parameter represents instantaneous energy flow from the last time flow is calculated. Unit is BTU/task interval (US) or MJ/task interval (Metric).	All
PrevHrAvgTemp	LREAL	This parameter is previous hour average for temperature.	All

Output Parameter	Data types	Description	Apply to
PreDayAvgTemp	LREAL	This parameter is previous day average for temperature.	All
PrevHrAvgPressure	LREAL	This parameter is previous hour average for pressure.	All
PreDayAvgPressure	LREAL	This parameter is previous day average for pressure.	All
PrevHrAvgGasMass	LREAL	This parameter is previous hour average for gas mass.	All
PreDayAvgGasMass	LREAL	This parameter is previous day average for gas mass.	All
PrevHrAvgDenAtBase	LREAL	This parameter is previous hour average for density at base condition.	All
PreDayAvgDenAtBase	LREAL	This parameter is previous day average for density at base condition.	All
PrevHrAvgRelDenAtBase	LREAL	This parameter is previous hour average for relative density at base condition.	All
PreDayAvgRelDenAtBase	LREAL	This parameter is previous day average for relative density at base condition.	All
PrevHrAvgUserDefined1	LREAL	This parameter is previous hour average for user defined parameter1.	All
PreDayAvgUserDefined1	LREAL	This parameter is previous day average for user defined parameter1.	All
PrevHrAvgUserDefined2	LREAL	This parameter is previous hour average for user defined parameter2.	All
PreDayAvgUserDefined2	LREAL	This parameter is previous day average for user defined parameter2.	All

Output Parameter	Data types	Description	Apply to
PrevHrAvgUserDefined3	LREAL	This parameter is previous hour average for user defined parameter3.	All
PreDayAvgUserDefined3	LREAL	This parameter is previous day average for user defined parameter3.	All
PrevHrAvgUserDefined4	LREAL	This parameter is previous hour average for user defined parameter4.	All
PreDayAvgUserDefined4	LREAL	This parameter is previous day average for user defined parameter4.	All
QbTH	LREAL	This parameter is volume flow rate at base condition total for this hour.	All
QbLH	LREAL	This parameter is volume flow rate at base condition total for last hour.	All
QbTD	LREAL	This parameter is volume flow rate at base condition total for this day.	All
QbLD	LREAL	This parameter is volume flow rate at base condition total for last day.	All
MTH	LREAL	This parameter is mass flow rate total for this hour.	All
MLH	LREAL	This parameter is mass flow rate total for last hour.	All
MTD	LREAL	This parameter is mass flow rate total for this day.	All
MLD	LREAL	This parameter is mass flow rate total for last day.	All
ETH	LREAL	This parameter is energy total for this hour.	All

Output Parameter	Data types	Description	Apply to
ELH	LREAL	This parameter is energy total for last hour.	All
ETD	LREAL	This parameter is energy total for this day.	All
ELD	LREAL	This parameter is energy total for last day.	All
QbNR	LREAL	Non-Resettable or Cumulative total for volume at Base. Unit - ft3/hr for US, m3/hr for Metric.	Coriolis_ GM_ MeterRun_ V2
MNR	LREAL	Non-Resettable or Cumulative total for Mass. Unit - lbm/hr for US, kg/hr for Metric.	Coriolis_ GM_ MeterRun_ V2
ENR	LREAL	Non-Resettable or Cumulative total for Energy. Unit -Btu/hr for US, MJ/hr for Metric.	Coriolis_ GM_ MeterRun_ V2
QbRollover	INT	Rollover flag for volume at base condition non-resettable total. The value {1} indicates Rollover otherwise {0}. The value for roll-over is 999,999,999. The rollover flag will be on 5 times interval time.	Coriolis_ GM_ MeterRun_ V2
MRollover	INT	Rollover flag for Non-Resettable Mass total. The value {1} indicates Rollover otherwise {0}. The value for roll-over is 999,999,999. The rollover flag will be on 5 times interval time.	Coriolis_ GM_ MeterRun_ V2
ERollover	INT	Rollover flag for Non-Resettable Energy total. The value {1} indicates Rollover otherwise {0}. The value for roll-over is 999,999,999. The rollover flag	Coriolis_ GM_ MeterRun_ V2

Output Parameter	Data types	Description	Apply to
		will be on 5 times interval time.	
QbMaint	LREAL	Volume at Base in Maintenance mode.	Coriolis_ GM_ MeterRun_ V2
MMaint	LREAL	Mass in Maintenance mode.	Coriolis_ GM_ MeterRun_ V2
EMaint	LREAL	Energy at Base in Maintenance mode.	Coriolis_ GM_ MeterRun_ V2

NOTE: The above outputs including averages and totals would be in the contract unit. The QTR generated by this function block contains following fields. Date; Time; Flow Time; Volume at Base; Mass; Energy; Temperature; Gas mass; Density; None; Relative Density; Average User Defined 1 (optional); Average User Defined 2 (optional); Average User Defined 3 (optional); Average User Defined 4 (optional).

Following are the error codes for the above meter run function block.

Out Code	Description	Apply to
0	SUCCESS	All
5	ERROR: THE ROOT WAS NOT BOUNDED IN DGROSS	All
6	ERROR: NO CONVERGENCE IN DGROSS	All
7	ERROR: VIRGS SQURE ROOT NEGATIVE	All
8	ERROR: COMBINED VALUES OF GRGR, X[2] AND HV NOT CONSISTENT	All
9	ERROR: INVALID TERM IN VIRGS	All
11	ERROR: METHOD WAS NOT 1 OR 2	All
12	ERROR: FLOWING PRESSURE (PF) <= 0.0 OR > 1740.0 PSIA	All
13	ERROR: FLOWING TEMPERATURE (TF) < 14.0 OR > 149.0 DEG F	All
14	ERROR: HEATING VALUE (HV) < 477.0 OR > 1211.0 BTU/FT^3	All
15	ERROR: GAS RELATIVE DENSITY (GRGR) < 0.55 OR > 0.870	All
16	ERROR: MOLE FRACTION FOR N2 < 0.0 OR > 0.50 OR FOR CO2 < 0.0 OR > 0.30 OR FOR H2 < 0.0 OR > 0.10 OR FOR CO < 0.0 OR > 0.03	All
17	ERROR: REFERENCE TEMPERATURE < 32.0 OR > 77.0 DEG F	All
18	ERROR: REFERENCE PRESSURE < 13.0 OR > 16.0 PSIA	All
22	WARNING: FLOWING PRESSURE (PF) <= 0.0 OR > 1200.0 PSIA	Coriolis_ GM_ MeterRun
23	WARNING: FLOWING TEMPERATURE (TF) < 32.0 OR > 130.0 DEG F	Coriolis_ GM_ MeterRun
24	WARNING: HEATING VALUE (HV) < 805.0 OR > 1208.0 BTU/FT^3	Coriolis_ GM_ MeterRun
25	WARNING: GAS RELATIVE DENSITY (GRGR) < 0.55 OR > 0.800	Coriolis_ GM_

Out Code	Description	Apply to
		MeterRun
26	WARNING: MOLE FRACTION FOR N2 < 0.0 OR > 0.20 OR FOR CO2 < 0.0 OR > 0.20 OR FOR H2 < 0.0 OR > 0.0 OR FOR CO < 0.0 OR >0.0	Coriolis_ GM_ MeterRun
81	WARNING: FLOWING PRESSURE (PF) > 1500.0 PSIA AGA8 2017 RANGE 1	Coriolis_ GM_ MeterRun_ V2
82	WARNING: FLOWING TEMPERATURE (TF) < 17.01 OR > 143.0 DEG F AGA8 2017 RANGE 2 OR (TF) < 25.0 OR > 143.0 DEG F AGA8 2017 RANGE 1	Coriolis_ GM_ MeterRun_ V2
83	WARNING: HEATING VALUE (HV) < 665.0 OR > 1100.0 BTU/FT^3 AGA8 2017 RANGE 2 OR (HV) < 930.0 OR > 1040.0 BTU/FT^3 AGA8 2017 RANGE 1	Coriolis_ GM_ MeterRun_ V2
84	WARNING: GAS RELATIVE DENSITY (GRGR) < 0.554 OR > 0.801 AGA8 2017 RANGE 2 OR (GRGR) < 0.554 OR > 0.630 AGA8 RANGE 1	Coriolis_ GM_ MeterRun_ V2
85	WARNING: MOLE FRACTION FOR N2 > 0.20 AGA8 2017 RANGE 2 OR N2 > 0.07 AGA8 2017 RANGE 1 OR FOR CO2 > 0.25 AGA8 2017 RANGE 2 OR CO2 > 0.03 AGA8 2017 RANGE 1 OR FOR H2 < 0.0 OR > 0.0 AGA8 2017 RANGE 1 AND 2 OR FOR CO < 0.0 OR > 0.0 AGA8 2017 RANGE 1 AND 2	Coriolis_ GM_ MeterRun_ V2

CHAPTER

6

API 21.2

The following libraries of API21.2 Function Blocks are supported:

Library	Description
API21_2	The function block library provides support for creating flow measurement calculations for various liquids based on API 21.2 standard for Turbine, Corolis, positive displacement and ultrasonic meters.
API21_2_V2	It is supported from R161.2 release. The function block library provides support for creating flow measurement calculations for Liquid based on API 21.2 standard for Turbine, Corolis, positive displacement and ultrasonic meters with upgraded support for low flow cutoff.

The following API 21.2 meter run function blocks are available:

Function Block	Description
Analog_AI_ Process	Function block to preprocess the data from the analog input channel of the ST103A device before the data is used with API21.2 function blocks. The general preprocessing includes scaling and analog input status determination.
Flowrate_Calc	Function block to compute the flow rate.
Liq_CrudeOil	This function block calculates volume correction factors for crude oil according to API 11.1. It takes input from flow meter in terms of Pulse/flow rate and calculate Gross Standard Volume, Net Standard Volume and Sediments and Water Volume and Mass. The function block supports base density from a offline densitometer, and live measured density from single/dual densitometer (fast loop). The block can take inputs either in US or Metric systems. The base temperature and pressure is 60 Deg F and 0 psig in US units. The base temperature is 15/20 Deg C and base pressure is 0 KPag in METRIC Unit. The function block has the capability to use keypad values when the live values from field devices are out of range or communication with devices is lost. The function block supports continuous operations and reporting. The function block also

Function Block	Description
	supports maintenance mode totalizer when the meter run is under maintenance mode. The function block can be configured to use with external ST103A or the native I/O of the ControlEdge RTU.
Liq_ LubricatingOil	This function block calculates volume correction factors for refined products like gasoline/diesel or lubricating oil according to API 11.1. It takes input from flow meter in terms of Pulse/flow rate and calculate Gross Standard Volume, and Mass. The function block supports base density from a offline densitometer and live measured density from single/dual densitometer (fast loop). The block can take inputs either in US or Metric systems. The base temperature and pressure is 60 Deg F and 0 psig in US units. The base temperature is 15/20 Deg C and base pressure is 0 KPag in METRIC units. The function block has the capability to use keypad values when the live values from field devices are out of range or communication with devices is lost. The function block supports continuous operations and reporting. The function block also supports maintenance mode totalizer when the meter run is under maintenance. The function block can be configured to use with external ST103A or the native I/O of the ControlEdge RTU.
Liq_NaturalGas	This function block calculates volume correction factors for Natural gas liquids (LNG/LPG) according to API 11.2.4, API 11.2.2/11.2.2M and API 11.2.5. It takes input from the flow meter in terms of Pulse/flow rate and calculate Gross Standard Volume, and Mass. The function block supports base density from a offline densitometer and live measured density from single/dual densitometer (fast loop). The block can take inputs either in US or Metric systems. The base temperature and pressure is 60 Deg F and 0 psig in US units. The base temperature is 15/20 Deg C and base pressure is 0 KPag in METRIC units. The function block has the capability to use keypad values when the live values from field devices are out of range or communication with devices is lost. The function block supports continuous operations and reporting. The function block also supports maintainence mode totalizer when meter run is under maintenance. In order to get accuracy as stated in API 11.2.4, the RoundingMethod needs to be set as enabled (1 as part of the

Function Block	Description
	configuration). The function block can be configured to use with external ST103A or the native I/O of the ControlEdge RTU.
Liq_ RefinedProducts	This function block calculates volume correction factors for refined products like gasoline or lubricating oil according to API 11.1. It takes input from flow meter in terms of Pulse/flow rate and calculate Gross Standard Volume, and Mass. The function block supports base density from a offline densitometer and live measured density from single/dual densitometer (fast loop). The block can take inputs either in US or Metric systems. The base temperature and pressure is 60 Deg F and 0 psig in US units. The base temperature is 15/20 Deg C and base pressure is 0 KPag in METRIC units. The function block has the capability to use keypad values when the live values from field devices are out of range or communication with devices is lost. The function block supports continuous operations and reporting. The function block also supports maintenance mode totalizer when the meter run is under maintenance. The function block can be configured to use with external ST103A or the native I/O of the ControlEdge RTU.
Liq_ SpecialProducts	This function block calculates volume correction factors for special products according to API 11.1. It takes input from flow meter in terms of Pulse/flow rate and calculate Gross Standard Volume, and Mass. The value of thermal expansion factor at 60 deg F needs to be provided. The function block supports base density from a offline densitometer and live measured density from single/dual densitometer (fast loop). The block can take inputs either in US or Metric systems. The base temperature and pressure is 60F and 0 psig in US units. The base temperature is 15/20 Deg C and base pressure is 0 KPag. The function block has the capability to use keypad values when the live values from field devices are out of range or communication with devices is lost. The function block supports continuous operations and reporting. The function block also supports maintenance mode totalizer when the meter run is under maintenance.

Function Block	Description
	The function block can be configured to use with external ST103A or the native I/O of the ControlEdge RTU.
Liquid_ StationTotalizer	The LiquidStationTotalizer function block calculates the meter station totals for the multiple streams connected to the Station. The Station totals are calculated by adding relevant individual totals from each stream and totalizing them to create Station totals. Station totalizer by default totalizes gross standard volume, net standard volume, mass and water and sediments volume. The station totalizer by default generates hourly and daily QTR similar to any other meter runs.
ST103A_Process	Function block to check the connection status of the ST103A device before the data is used with other API21.2 function blocks.
Volume_ Correction_FB	Function block to compute the corrected Volume for the given CTL and CPL.

Liq_CrudeOil, Liq_LubricatingOil, Liq_ NaturalGas, Liq_RefinedProducts and Liq_ SpecialProducts

Description

Function Block	Description
Liq_CrudeOil	This function block calculates volume correction factors for crude oil according to API 11.1. It takes input from flow meter in terms of Pulse/flow rate and calculate Gross Standard Volume, Net Standard Volume and Sediments and Water Volume and Mass. The function block supports base density from a offline densitometer, and live measured density from single/dual densitometer (fast loop). The block can take inputs either in US or Metric systems. The base temperature and pressure is 60 Deg F and 0 psig in US units. The base temperature is 15/20 Deg C and base pressure is 0 KPag in METRIC Unit. The function block has the capability to use keypad values when the live values from field devices are out of range or communication with devices is lost. The function block supports continuous operations and reporting. The function block also supports maintenance mode totalizer when the meter run is under

Function Block	Description
	maintenance mode. The function block can be configured to use with external ST103A or the native I/O of the ControlEdge RTU.
Liq_ RefinedProducts	This function block calculates volume correction factors for refined products like gasoline or lubricating oil according to API 11.1. It takes input from flow meter in terms of Pulse/flow rate and calculate Gross Standard Volume, and Mass. The function block supports base density from a offline densitometer and live measured density from single/dual densitometer (fast loop). The block can take inputs either in US or Metric systems. The base temperature and pressure is 60 Deg F and 0 psig in US units. The base temperature is 15/20 Deg C and base pressure is 0 KPag in METRIC units. The function block has the capability to use keypad values when the live values from field devices are out of range or communication with devices is lost. The function block supports continuous operations and reporting. The function block also supports maintenance mode totalizer when the meter run is under maintenance. The function block can be configured to use with external ST103A or the native I/O of the ControlEdge RTU.
Liq_ LubricatingOil	This function block calculates volume correction factors for refined products like gasoline/diesel or lubricating oil according to API 11.1. It takes input from flow meter in terms of Pulse/flow rate and calculate Gross Standard Volume, and Mass. The function block supports base density from a offline densitometer and live measured density from single/dual densitometer (fast loop). The block can take inputs either in US or Metric systems. The base temperature and pressure is 60 Deg F and 0 psig in US units. The base temperature is 15/20 Deg C and base pressure is 0 KPag in METRIC units. The function block has the capability to use keypad values when the live values from field devices are out of range or communication with devices is lost. The function block supports continuous operations and reporting. The function block also supports maintenance mode totalizer when the meter run is under maintenance.

Function Block	Description
	The function block can be configured to use with external ST103A or the native I/O of the ControlEdge RTU.
Liq_ SpecialProducts	This function block calculates volume correction factors for special products according to API 11.1. It takes input from flow meter in terms of Pulse/flow rate and calculate Gross Standard Volume, and Mass. The value of thermal expansion factor at 60 deg F needs to be provided. The function block supports base density from a offline densitometer and live measured density from single/dual densitometer (fast loop). The block can take inputs either in US or Metric systems. The base temperature and pressure is 60F and 0 psig in US units. The base temperature is 15/20 Deg C and base pressure is 0 KPag. The function block has the capability to use keypad values when the live values from field devices are out of range or communication with devices is lost. The function block supports continuous operations and reporting. The function block also supports maintenance mode totalizer when the meter run is under maintenance. The function block can be configured to use with external ST103A or the native I/O of the ControlEdge RTU.
Liq_NaturalGas	This function block calculates volume correction factors for Natural gas liquids (LNG/LPG) according to API 11.2.4, API 11.2.2/11.2.2M and API 11.2.5. It takes input from the flow meter in terms of Pulse/flow rate and calculate Gross Standard Volume, and Mass. The function block supports base density from a offline densitometer and live measured density from single/dual densitometer (fast loop). The block can take inputs either in US or Metric systems. The base temperature and pressure is 60 Deg F and 0 psig in US units. The base temperature is 15/20 Deg C and base pressure is 0 KPag in METRIC units. The function block has the capability to use keypad values when the live values from field devices are out of range or communication with devices is lost. The function block supports continuous operations and reporting. The function block also supports maintainence mode totalizer when meter run is under maintenance. In order to get accuracy as stated in API 11.2.4, the RoundingMethod needs to be set as enabled (1 as part of the configuration). The function block can be configured to use with external ST103A

Function Block	Description
	or the native I/O of the ControlEdge RTU.

Input

Input Parameter	Data types	Description	Apply to
MeterRunID	Integer	Identifier for the configured Meter Run. Possible values 1 to 12 (Redundant), 1 to 4 (Non-Redundant)	All
MeterType	Integer	Type of the flow meter: • 1 for Turbine meter or • 2 for Positive Displacement Meter • 3 for Ultrasonic Meter or • 4 for Coriolis Meter • 5 for Station Totalizer	All
InputUnit	Integer	Unit for all the input parameters. The value should be 1 for US or 2 for Metric.	All
ContractUnit	Integer	Unit for all the ouput parameters. The value should be 1 for US or 2 for Metric.	All
CorMtrAsDensiMtr	Integer	Option to set whether the Coriolis meter acts as Densitometer. The values should be either 1 for Yes or 0 for No. This parameter is applicable only for Corolis meter type.	All

Input Parameter	Data types	Description	Apply to
ThermalExpansionFactor	REAL	Thermal expansion factor at 60°F. This input parameter is only applicable for Special Products. For other commodity types this parameter is not present.	Liq_ SpecialProducts
DensiMtrAvailable	Integer	Option to set whether the measured desity from live Denstiometer is available or offline base desity will be used . The value should be either 1 for Yes or 0 for No.	All
DensiMtrCount	Integer	The number of Densitometers available. The value should be either 1 for Single or 2 for Dual densitometer.	All
DensiTemp	REAL	Parameter to set the Densitometer temperature. The value should be in Fahrenheit for US unit and in Celcius for Metric unit.	All
DensiTempLoLo	REAL	This is the LoLo limit for Densitometer temperature. The value should be in Fahrenheit for US unit and in Celcius for Metric unit.	All
DensiTempLo	REAL	This is the Lo limit for Densitometer temperature. The value should be in Fahrenheit for US unit and in Celcius for Metric unit.	All
DensiTempHi	REAL	This is the Hi limit for Densitometer temperature. The value should be in Fahrenheit for US unit and in Celcius for Metric unit.	All

Input Parameter	Data types	Description	Apply to
DensiTempHiHi	REAL	This is the HiHi limit for Densitometer temperature. The value should be in Fahrenheit for US unit and in Celcius for Metric unit.	All
DensiTempIOSelection	Integer	I/O selection for Densitometer Temperature. The value should be 1 for Live or 2 for Keypad value.	All
DensiTempStsStatus	USINT	Analog input channel status for Densitometer Temperature. The value should be 0 for good or any positive integer for bad status.	All
DensiTempKeypadVal	REAL	Keypad value for Densitometer temperature. The value that should be used when the Densitometer temperature status is bad.	All
DensiPressure	REAL	Parameter to set the Densitometer pressure. The value should be in Psig for US unit and in Kpag for Metric unit.	All
DensiPressureLoLo	REAL	This is the LoLo limit for Densitometer pressure. The value should be in Psig for US unit and in Kpag for Metric unit.	All
DensiPressureLo	REAL	This is the Lo limit for Densitometer pressure. The value should be in Psig for US unit and in Kpag for Metric unit.	All

Input Parameter	Data types	Description	Apply to
DensiPressureHi	REAL	This is the Hi limit for Densitometer pressure. The value should be in Psig for US unit and in Kpag for Metric unit.	All
DensiPressureHiHi	REAL	This is the HiHi limit for Densitometer pressure.The value should be in Psig for US unit and in Kpag for Metric unit.	All
DensiTempIOSelection	Integer	I/O selection for Densitometer pressure. The value should be 1 for Live or 2 for Keypad value.	All
DensiPressStsStatus	USINT	Analog input channel status for Densitometer Pressure. The value should be 0 for good or any positive integer for bad status.	All
DensiPressKeypadVal	REAL	Keypad value for Densitometer pressure. The value that should be used when the Densitometer pressure status is not good.	All
Densi1MeasuredDensity	REAL	Density value of the Densitometer 1. The value should be in lb/ft^3 for US unit and kg/m^3 for Metric unit.	All
Densi2MeasuredDensity	REAL	Density value of the Densitometer 2. The value should be in lb/ft^3 for US unit and kg/m^3 for Metric unit. This value is applicable only for Dual Densitometer	All
MeasuredDensityIOSel	Interger	I/O selection for Density. The value should be 1 for	All

Input Parameter	Data types	Description	Apply to
		Live or 2 for Keypad value.	
Densi1Status	USINT	Status of Densitometer 1 . The value should be 0 for good status or any positive integer for bad status.	All
Densi2Status	USINT	Status of Densitometer 2. The value should be 0 for good status or any positive integer for bad status.	All
PreferredDensiMtr	Interger	Parameter to select the preferred Densitometer. The value should be 1 for Densitometer 1 and 2 for Densitometer 2.	All
DensiKeypadValue	REAL	Keypad value for Densitometer measured density. The value should be in lb/ft^3 for US unit and kg/m^3 for Metric unit. This value should be a base density value. When on-line live densitometer status is bad, base density will be used a keypad value.	All
BaseDensity	REAL	Density at Base conditions. The value should be in lb/ft^3 for US unit and kg/m^3 for Metric unit.	All
BaseTemperature	REAL	Temperature at Base conditions. The value should be in Fahrenheit for US unit and in Celcius for Metric unit. The recommended default is 60 Deg F for US and 15/20 Deg C for Metric.	All

Input Parameter	Data types	Description	Apply to
MeterFactor	REAL	Meter K factor to convert pulse form flow meter into volume. The value should be in pulse/ft^3 for US unit in pulse/m^3 for Metric unit. When the flow type is equal to analog, this is the correction factor to apply for volume calculation and default value should be 1.0.	All
RoundingMethod	Integer	Option to enable or disable the rounding of output parameter values. The value should be 0 for Disabled and 1 for Enabled.	All
Temp	REAL	Parameter to set the meter temperature. The value should be in Fahrenheit for US unit and in Celcius for Metric unit.	All
TempLoLo	REAL	This is the LoLo limit for meter temperature. The value should be in Fahrenheit for US unit and in Celcius for Metric unit.	All
TempLo	REAL	This is the Lo limit for meter temperature. The value should be in Fahrenheit for US unit and in Celcius for Metric unit.	All
TempHi	REAL	This is the Hi limit for meter temperature. The value should be in Fahrenheit for US unit and in Celcius for Metric unit.	All
TempHiHi	REAL	This is the HiHi limit for meter temperature. The value should be in	All

Input Parameter	Data types	Description	Apply to
		Fahrenheit for US unit and in Celcius for Metric unit.	
TempIOSelection	Integer	I/O selection for meter temperature. The value should ne 1 for Live or 2 for Keypad value.	All
TempStsStatus	USINT	Analog input channel status for meter temperature. The value should be 0 for good or any positive integer for bad status.	All
TempKeypadVal	REAL	Keypad value for meter temperature. The value that should be used when the meter temperature status is bad.	All
Pressure	REAL	Parameter to set the meter pressure. The value should be in Psig for US unit and in Kpag for Metric unit.	All
PressureLoLo	REAL	This is the LoLo limit for meter pressure. The value should be in Psig for US unit and in Kpag for Metric unit.	All
PressureLo	REAL	This is the LoLo limit for meter pressure. The value should be in Psig for US unit and in Kpag for Metric unit.	All
PressureHi	REAL	This is the Hi limit for meter pressure.The value should be in Psig for US unit and in Kpag for Metric unit.	All
PressureHiHi	REAL	This is the HiHi limit for meter pressure. The value	All

Input Parameter	Data types	Description	Apply to
		should be in Psig for US unit and in Kpag for Metric unit.	
PressureIOSelection	Integer	I/O selection for meter pressure. The value should ne 1 for Live or 2 for Keypad value.	All
PressureStsStatus	USINT	Analog input channel status for meter pressure. The value should be 0 for good status or any positive integer for bad status.	All
PressureKeypadVal	REAL	Keypad value for meter pressure. The value that should be used when the meter pressure status is bad.	All
PulseOrAnalogInput	Integer	Parameter to set the flow type. The value should be 0 for Pulse and 1 for Analog.	All
IOType	Interger	Parameter to set the I/O type for Pulse input. The value should be 1 for ST103A (3rd party hardware) and 2 for Native I/O.	All
LowPulseCutoff	UINT	Low pulse cutoff value checks the no flow condition in the calculations. If the Pulse increment is less than this number, it will be considered as no flow condition.	All
Pulse	UDINT	The pulse counter value.	All
STMsgId	UDINT	Message ID from the ST103A device. This will be	All

Input Parameter	Data types	Description	Apply to
		used as heartbeat to know whether ST103A is live and communicating with ControlEdge RTU. MessageID will increment 2 in 1 seconds.	
STWDWaitTime	UINT	Maximum wait time in seconds for the ST103A to restore the connection after connection failure with ControlEdge RTU. Beyond this limit, pulse increments will not be utliized in volume calculations. The status for hourly/daily/Batch totals will be set to 1, i.e. the totals are suspicious.	All
MaxPulseIncrement	UDINT	Maximum pulse increment limit. If the pulse increment is beyond this limit, an alarm will be generated.	All
Analog	REAL	Value of analog input if flow type is qual to Analog. The value should be in lb^3/hr for US unit or m^3/hr for Metric unit.	All
AnalogLoLo	REAL	This is the LoLo limit for analog input. The value should be in lb^3/hr for US unit or m^3/hr for Metric unit.	All
AnalogLo	REAL	This is the Lo limit for analog input. The value should be in lb^3/hr for US unit or m^3/hr for Metric unit	All
AnalogHi	REAL	This is the Hi limit for	All

Input Parameter	Data types	Description	Apply to
		analog input. The value should be in lb^3/hr for US unit or m^3/hr for Metric unit.	
AnalogHiHi	REAL	This is the HiHi limit for analog input. The value should be in lb^3/hr for US unit or m^3/hr for Metric unit.	All
AveragingVariable	Integer	Parameter to set the variable used for flow weighted averaging. The value should be 0 for Gross Volume or 1 for Mass. For Liquid EFM, the averaging is based on flow weighted.	All
ContractHour	Integer	Parameter represents the time of daily QTR generation, as the day roll over to next day. Its value should be from 0 to 23. This has to be set according to the contract.	All
OperationType	Integer	Parameter to set the type of Operation. The value should be 1 for Continuous or 2 for Batch. When the operation type is equal to Batch, by default batch reporting and hourly reporting within batch will be configured automatically. When the mode is continuous, hourly and daily reports will be configured by default.	All
MaintMode	Integer	Parameter to Start or Stop the Maintenance Mode. The value should be either 0 for	All

Input Parameter	Data types	Description	Apply to
		Maintenance end or 1 for Maintenance Start. When Maintenance mode is started, an independent totalizer will be started and accumulate all the flow until the maintenance mode is stopped. During this period, non-resettable totals, hourly/daily/batch totals will be not incremented.	
BatchIdentifier	Integer	Identifier for a Batch operation. This number will be incremented for every batch.	All
RetroKfactor	Integer	Parameter to set for recalculation of totals if proving happens during a batch process and totals are adjusted with new Meter K factor. By default this parameter is set enabled.	All
WaterAndSedMeasAvbl	Integer	Parameter to set whether the Water and Sediments is available for the Crude Oil. The value should be 1 if available otherwise 0.	Liq_CrudeOil
WaterAndSediments	REAL	Parameter to set the percentage of Water and Sediments present in the Crude Oil.	Liq_CrudeOil
WaterAndSedIOSel	Integer	I/O selection for Water and Sediments value. The value should be 1 for Live or 2 for Keypad value.	Liq_CrudeOil
WaterAndSedStatus	USINT	Analog input channel status for Water and	Liq_CrudeOil

Input Parameter	Data types	Description	Apply to
		Sediments.The value should be 0 for good status or any positive integer for bad status.	
WaterAndSedKeypad	REAL	Keypad value in percentage for Water and Sediments. The value that should be used when the Water and Sediments status is bad.	Liq_CrudeOil
CPLCalcType	Integer	CPL Calculation Type. The value should be 1 for None or 2 for API21.2 or 3 for API21.2M.	Liq_NaturalGas
ConverCriteria	REAL	IP2 Convergence limit. Default value is 0.001.	Liq_NaturalGas
MaxIterations	Integer	IP2 Max loop limit. Default value is 50.	Liq_NaturalGas
IterationMethod	Integer	Main calculation method. The value should be either 1 for ASTM or 2 for IP2	Liq_NaturalGas
VapourPrInput	REAL	Parameter to set the user observed Vapour Pressure Input.	Liq_NaturalGas
VapourPrCalcMethod	Integer	Vapour Pressure calculation options. The value should be 1 for None, 2 for User Observed or 3 for API 11.2.5.	Liq_NaturalGas
Lowflowcutoff	REAL	Low flow cutoff value checks the no flow condition in the calculations. If the flow is s less than this number, it will be considered as no flow condition. Unit is m3/hr for Metric unit, ft3/hr for US unit.	All liquid types function blocks in API21.2_V2 library.

Output

Input Parameter	Data types	Description	Apply to
Out_Code	Integer	This out parameter returns success or fail code. The value will be 0 for success or -1 for Exception or +ve value for Error codes.	All
BaseDensityComputed	LREAL	Density at Base conditions. The value will be in lb/ft^3 for US unit and in kg/m^3 for Metric unit.	All
MeterDensity	LREAL	Density at measurement or metering conditions. The value will be in lb/ft^3 for US unit and in kg/m^3 for Metric unit.	All
ObservedDensity	LREAL	Observed Density or measured density. The value will be in lb/ft^3 for US unit and in kg/m^3 for Metric unit.	All
NetStdVolume	LREAL	This parameter is Net Standard volume increment. It is in ft^3/sec for US unit system and in m^3/sec for Metric unit system.	All
GrossStdVolume	LREAL	This parameter is Gross Standard volume increment. It is in ft^3/sec for US unit system and in m^3/sec for Metric unit system.	All
SedAndWaterVolume	LREAL	This parameter is Sediment And Water volume increment. It is in ft^3/sec for US unit system and in m^3/sec for Metric unit system.	All
CTL	LREAL	Correction factor for effects of temperature on the liquid.	All
CPL	LREAL	Correction factor for effects of pressure on the liquid.	All
Mass	LREAL	This parameter is mass increment. It is in lb/sec for US unit system and in kg/sec for Metric unit system.	All
AvgCTLLH	LREAL	Average CTL for the previous hour.	All

Input Parameter	Data types	Description	Apply to
AvgCTLLDOrBat	LREAL	Average CTL for the previous day or Batch.	All
AvgCPLLH	LREAL	Average CPL for the previous hour.	All
AvgCPLLDOrBat	LREAL	Average CTL for the previous day or Batch.	All
AvgObsDenLH	LREAL	Average Observed Density for the previous hour.	All
AvgObsDenLDOrBat	LREAL	Average Observed Density for the previous day or Batch.	All
AvgTempLH	LREAL	Average Temperature for the previous hour.	All
AvgTempLDOrBat	LREAL	Average Temperature for the previous day or Batch.	All
AvgPressLH	LREAL	Average Pressure for the previous hour.	All
AvgPressLDOrBat	LREAL	Average Pressure for the previous day or Batch.	All
AvgBaseDenLH	LREAL	Average Base Density for the previous hour.	All
AvgBaseDenLDOrBat	LREAL	Average Base Density for the previous day or Batch.	All
SwVTH	LREAL	Sediments And Water volume total for this hour.	All
SwVLH	LREAL	Sediment And Water volume total for last hour.	All
SwVTDOrBat	LREAL	Sediment And Water volume total for this day or Batch.	All
SwVLDOrBat	LREAL	Sediment And Water volume total for last day or Batch.	All
NSVTH	LREAL	Net Standard volume total for this hour.	All
NSVLH	LREAL	Net Standard volume total for last hour.	All
NSVTDOrBat	LREAL	Net Standard volume total for this day or Batch	All

Input Parameter	Data types	Description	Apply to
NSVLDOrBat	LREAL	Net Standard volume total for last day or Batch	All
MassTH	LREAL	Mass total for this hour.	All
MassLH	LREAL	Mass total for last hour.	All
MassTDOrBat	LREAL	Mass total for this day or batch.	All
MassLDOrBat	LREAL	Mass total for last day or batch.	All
GSVTH	LREAL	Gross Standard volume total for this hour.	All
GSVLH	LREAL	Gross Standard volume total for last hour.	All
GSVTDOrBat	LREAL	Gross Standard volume total for this day or Batch.	All
GSVLDOrBat	LREAL	Gross Standard volume total for last day or Batch.	All
SwVMaint	LREAL	Sediments And Water volume total for this hour in Maintenance mode.	All
NSVMaint	LREAL	Net Standard volume total in Maintenance mode.	All
MassMaint	LREAL	Mass total in Maintenance mode.	All
GSVMaint	LREAL	Gross Standard volume total in Maintenance mode.	All
SwVNR	LREAL	Non-Resettable Sediment And Water volume total.	All
NSVNR	LREAL	Non-Resettable Net Standard volume total.	All
MassNR	LREAL	Non-Resettable Mass total.	All
GSVNR	LREAL	Non-Resettable Gross Standard volume total.	All
SwVNRRollover	Integer	Rollover flag for Non-Resettable Sediments And Water volume total. The value 1 indicates Rollover otherwise 0.	All

Input Parameter	Data types	Description	Apply to
		The value for roll-over is 999,999,999. The rollover flag will be on 5 times interval time.	
NSVNRRollover	Integer	Rollover flag for Non-Resettable Net Standard volume total. The value 1 indicates Rollover otherwise 0. The value for roll-over is 999,999,999. The rollover flag will be on 5 times interval time.	All
MassNRRollover	Integer	Rollover flag for Non-ResettableMass total. The value 1 indicates Rollover otherwise 0. The value for roll-over is 999,999,999. The rollover flag will be on 5 times interval time.	All
GSVNRRollover	Integer	Rollover flag for Non-ResettableGross Standard volume total. The value 1 indicates Rollover otherwise 0. The value for roll-over is 999,999,999. The rollover flag will be on 5 times interval time.	All
StatusTH	Integer	Status of the period total for the this hour. The value will be 0 for good status or 1 for bad status.	All
StatusTD	Integer	Status of the period total for the this day. The value will be 0 for good status or 1 for bad status.	All
STIOReset	Integer	This parameter indicates whether the ST103A is restarted or not. The value will be 0 for false or 1 for true. The reset flag will be on for 5 times of interval time.	All

Input and Output

Input Parameter	Data types	Description	Apply to
BatchMode	Any	This parameter to set the Batch mode. The value should be 0 for Batch End and 1 for Batch Start.	All

> **NOTE:** If the flow type Input is Analog, operation type Batch is not supported.

Data type

See the following data types for detailed information:

Data types	Details
UINT2	TYPE UINT2: ARRAY [1..2] OF UINT; (*to support Modbus write*) END_TYPE
REAL5	TYPE REAL5: ARRAY [1..7] OF REAL; (*to support AI*) END_TYPE
UINT1	TYPE UINT1: ARRAY [1..1] OF UDINT; (*to support Pulse A*) END_TYPE
UDINT2	TYPE UDINT2: ARRAY [1..3] OF UDINT; (*to support Pulse E*) END_TYPE
REAL2	TYPE REAL2: ARRAY [1..2] OF REAL; (*to support Freq*) END_TYPE
PULSE_INPUT_TYPE_DUAL	TYPE PULSE_INPUT_TYPE_DUAL : STRUCT GOOD_PULSE : UDINT; (* Good Pulse*)

Data types	Details
	END_STRUCT; END_TYPE
FREQUENCY_INPUT_TYPE	TYPE FREQUENCY_INPUT_TYPE : STRUCT STS : USINT; (*Status*) PV : REAL; (* Frequency *) END_STRUCT; END_TYPE
ANALOG_INPUT_TYPE _TP	TYPE ANALOG_INPUT_TYPE _TP : STRUCT STS : USINT; (*Status*) PV : REAL; (* Value *) EUHI:REAL(* Range Hi *) EULO: REAL (*Range Low *) EUHIEX:REAL (*Range Hi Extended *) EULOEX:REAL (*Range Low Extended *) END_STRUCT; END_TYPE
BATCH_TOTALISE_STRUCT	TYPE BATCH_TOTALISE_STRUCT : STRUCT BATCHMODE :INT; END_STRUCT; END_TYPE

Error Codes

See the following table for error codes for the function blocks Liq_CrudeOil , Liq_RefinedProducts,Liq_LubricatingOil and Liq_SpecialProducts :

Error Code	Description
0	No error, Calculations Successful
1	Error - Illegal arguments
2	Error - Memory allocation
3	Error - VCF out of range
4	Error - Non convergence
5	Error - Temperature out of range
6	Error - Density out of range
7	Error - Pressure out of range
8	Error - Alpha60 out of range
9	Error - Supercritical fluid
10	Error - No reference fluids
11	Error - No Solution

See the following table for error codes for the function block LIQ_Naturalgas:

Error Code	Description
1	Density input is out of range (all calculations)
2	Temperature input is out of range (all calculations)
3	Pressure input is out of range (all calculations)
4	Calculation combination is invalid (all calculations)
14	API.11.2.4: Alpha error
15	API.11.2.4: Interpolation variable error
16	API.11.2.4: TC error
17	API.11.2.4: TRX error

Error Code	Description
18	API.11.2.4: H2 error
19	API.11.2.4: Saturated density error
20	API.11.2.4: Interpolation factor error
21	API.11.2.4: Step 4-5 error
22	API.11.2.4: Fluid 2 relative density low error
23	API.11.2.4: Step 6 TC2_TC1 error
24	API.11.2.4: RD X < Lower Limit
25	API.11.2.4: RD 60 Mid error
26	API.11.2.4: Step 9 Phi error
27	API.11.2.4: Step 9 A error
28	API.11.2.4: Step 9 B error
29	API.11.2.4: Step 9 RD 60 Trial error
30	API.11.2.4: Iteration Fail error
31	API.11.2.4: CTL range error
32	API.11.2.4: T60 Step 6 density error
101	API.11.2.4: Density conversion error
102	API.11.2.4: Rounding error
103	API.11.2.4: Reserved
104	API.11.2.4: CTL range error
105	API.11.2.4: CPL range error
106	API.11.2.4: Reserved
107	API.11.2.4: Reserved
108	API.11.2.4: Calculated density range error
109	API.11.2.4: Density units conversion error
110	API.11.2.4: Pressure units conversion error
111	API.11.2.4: CTPL range error
211	API 1122 and API1122M: TR > Max error

Error Code	Description
212	API 1122 and API1122M: Factor error
301	Ch.11.2.5: relative density out of range
302	Ch.11.2.5: Temperature out of range

LiquidStationTotalizer

Description

The LiquidStationTotalizer function block calculates the meter station totals for the multiple streams connected to the Station. The Station totals are calculated by adding relevant individual totals from each stream and totalizing them to create Station totals. Station totalizer by default totalizes gross standard volume, net standard volume, mass and water and sediments volume. The station totalizer by default generates hourly and daily QTR similar to any other meter runs.

> **ATTENTION:** Station totalizer can be configured for totalizing 2 to 4 meter runs. The maximum quantity that can be added to station totalizer is 4 and minimum is 2. When configuring the station totalizer, meter runs of same liquid type, operation type (period/batch) and contract hours should be considered. This function block will not do any error handling if the above conditions are not met.

Input

Input Parameter	Data Type	Description
MeterRunID	INT	Identifier for the configured Meter Run.
MeterType	INT	Type of the meter. The value should be {5} for Station Totalizer.
ContractHour	INT	This value represents the hour on which a day roll over for reporting. Its value should be from 0 to 23.
SwVTH1	LREAL	Sediments And Water volume total for this hour for the first meter run.

Input Parameter	Data Type	Description
SwVTDOrBat1	LREAL	Sediment And Water volume total for this day or Batch for the first meter run.
SwVNR1	LREAL	Non-Resettable Sediment And Water volume total for the first meter run.
NSVTH1	LREAL	Net Standard volume total for this hour for the first meter run.
NSVTDOrBat1	LREAL	Net Standard volume total for this day or Batch for the first meter run.
NSVNR1	LREAL	Non-Resettable Net Standard volume total for the first meter run.
MassTH1	LREAL	Mass total for this hour for the first meter run.
MassTDOrBat1	LREAL	Mass total for this day or batch for the first meter run.
MassNR1	LREAL	Non-Resettable Mass total for the first meter run.
GSVTH1	LREAL	Gross Standard volume total for this hour for the first meter run.
GSVTDOrBat1	LREAL	Gross Standard volume total for this day or Batch for the first meter run.
GSVNR1	LREAL	Non-Resettable Gross Standard volume total for the first meter run.
StatusTH1	INT	Status of the calculation for this hour for the first meter run. The value will be {0} for Status good or {1} for bad status.
StatusTD1	INT	Status of the calculation for the this day for the first meter run. The value will be {0} for Status good or {1} for bad status.
SwVTH2	LREAL	Sediments And Water volume total for this hour for the second meter run.
SwVTDOrBat2	LREAL	Sediment And Water volume total for this day or Batch for the second meter run.
SwVNR2	LREAL	Non-Resettable Sediment And Water volume total for the second meter run.
NSVTH2	LREAL	Net Standard volume total for this hour for the second meter run.

Input Parameter	Data Type	Description
NSVTDOrBat2	LREAL	Net Standard volume total for this day or Batch for the second meter run.
NSVNR2	LREAL	Non-Resettable Net Standard volume total for the second meter run.
MassTH2	LREAL	Mass total for this hour for the second meter run.
MassTDOrBat2	LREAL	Mass total for this day or batch for the second meter run.
MassNR2	LREAL	Non-Resettable Mass total for the second meter run.
GSVTH2	LREAL	Gross Standard volume total for this hour for the second meter run.
GSVTDOrBat2	LREAL	Gross Standard volume total for this day or Batch for the second meter run.
GSVNR2	LREAL	Non-Resettable Gross Standard volume total for the second meter run.
StatusTH2	INT	Status of the calculation for this hour for the second meter run. The value will be {0} for Status good or {1} for bad status.
StatusTD2	INT	Status of the calculation for the this day for the second meter run. The value will be {0} for Status good or {1} for bad status.
SwVTH3	LREAL	Sediments And Water volume total for this hour for the third meter run.
SwVTDOrBat3	LREAL	Sediment And Water volume total for this day or Batch for the third meter run.
SwVNR3	LREAL	Non-Resettable Sediment And Water volume total for the third meter run.
NSVTH3	LREAL	Net Standard volume total for this hour for the third meter run.
NSVTDOrBat3	LREAL	Net Standard volume total for this day or Batch for the third meter run.
NSVNR3	LREAL	Non-Resettable Net Standard volume total for the third meter run.

Input Parameter	Data Type	Description
MassTH3	LREAL	Mass total for this hour for the third meter run.
MassTDOrBat3	LREAL	Mass total for this day or batch for the third meter run.
MassNR3	LREAL	Non-Resettable Mass total for the third meter run.
GSVTH3	LREAL	Gross Standard volume total for this hour for the third meter run.
GSVTDOrBat3	LREAL	Gross Standard volume total for this day or Batch for the third meter run.
GSVNR3	LREAL	Non-Resettable Gross Standard volume total for the third meter run.
StatusTH3	INT	Status of the calculation for this hour for the third meter run. The value will be {0} for Status good or {1} for bad status.
StatusTD3	INT	Status of the calculation for the this day for the third meter run. The value will be {0} for Status good or {1} for bad status.
SwVTH4	LREAL	Sediments And Water volume total for this hour for the fourth meter run.
SwVTDOrBat4	LREAL	Sediment And Water volume total for this day or Batch for the fourth meter run.
SwVNR4	LREAL	Non-Resettable Sediment And Water volume total for the fourth meter run.
NSVTH4	LREAL	Net Standard volume total for this hour for the fourth meter run.
NSVTDOrBat4	LREAL	Net Standard volume total for this day or Batch for the fourth meter run.
NSVNR4	LREAL	Non-Resettable Net Standard volume total for the fourth meter run.
MassTH4	LREAL	Mass total for this hour for the fourth meter run.
MassTDOrBat4	LREAL	Mass total for this day or batch for the fourth meter run.
MassNR4	LREAL	Non-Resettable Mass total for the fourth meter run.
GSVTH4	LREAL	Gross Standard volume total for this hour for the fourth meter run.

Input Parameter	Data Type	Description
GSVTDOrBat4	LREAL	Gross Standard volume total for this day or Batch for the fourth meter run.
GSVNR4	LREAL	Non-Resettable Gross Standard volume total for the fourth meter run.
StatusTH4	INT	Status of the calculation for this hour for the fourth meter run. The value will be {0} for Status good or {1} for bad status.
StatusTD4	INT	Status of the calculation for the this day for the fourth meter run. The value will be {0} for Status good or {1} for bad status.

Output

Output Parameter	Description	Description
Out_Code	INT	This out parameter returns success or fail code. The value wll be {0} for success or {1} for Exception or {+ve value} for Error codes.
SwVTH	LREAL	Sum of Sediments And Water volume total for this hour for meter runs configured for station total.
SwVTDOrBat	LREAL	Sum of Sediment And Water volume total for this day or Batch for station total.
SwVNR	LREAL	Sum of Non-Resettable Sediment And Water volume total for station total.
NSVTH	LREAL	Sum of Net Standard volume total for this hour for station total.
NSVTDOrBat	LREAL	Sum of Net Standard volume total for this day or Batch for station total.
NSVNR	LREAL	Sum of Non-Resettable Net Standard volume total for station total.
MassTH	LREAL	Sum of Mass total for this hour for station total.
MassTDOrBat	LREAL	Sum of Mass total for this day or batch for station total.

Output Parameter	Description	Description
MassNR	LREAL	Sum of Non-Resettable Mass total for station total.
GSVTH	LREAL	Sum of Gross Standard volume total for this hour for station total.
GSVTDOrBat	LREAL	Sum of Gross Standard volume total for this day or Batch for station total.
GSVNR	LREAL	Sum of Non-Resettable Gross Standard volume total for station total.
StatusTH	INT	Overall status of the calculation for this hour for station total. The value will be {0} for Status good or {1} for bad status.
StatusTD	INT	Overall status of the calculation for the this day for station total. The value will be {0} for Status good or {1} for bad status.

Analog_AI_Process

Description

Function block to preprocess the data from the analog input channel of the ST103A device before the data is used with API21.2 function blocks. The general preprocessing includes scaling and analog input status determination.

Input

Input Parameter	Data Type	Description
AI	REAL	Value read from analog input channel of ST103A device.
EUHi	REAL	This is the Hi limit for analog input in Engineering units.
EULo	REAL	This is the Lo limit for analog input in Engineering units.
EUHiHi	REAL	This is the HiHi limit for analog input in Engineering units.
EULoLo	REAL	This is the LoLo limit for analog input in Engineering units.
ST103Status	INT	Status of ST103A device. The value will be {0} for status good or {1} for communication with ST103A is lost.

Output

Output Parameter	Data Type	Description
PV	REAL	Process value of the specific analog input channel in scaled to engineering units.
Status	USINT	Status of the analog input channel. Possible values: 0-Channel is good. 1-Channel is offline. The communication with device is lost. 11-The value is higher than the extended high range value. 12- The value is higher than the high range value but lower than the extended high range value. 13- The value is lower than the low range value but higher than the extended low range value. 14- The value is lower than than the extended low range value.

TIP: This function block should not be used as a standalone function. It is internally used by API 21.2 function blocks.

ST103A_Process

Description

Function block to check the connection status of the ST103A device before the data is used with API21.2 function blocks.

Input

Input Parameter	Data Type	Description
STMsgId	UDINT	Message ID from ST103A device

Output

Output Parameter	Data Type	Description
Status	USINT	Status of the ST103A device Possible values: 0-Good status 1- Communication with ST103A device is lost.

> **TIP:** This function block should not be used as a standalone function. It is internally used by API 21.2 function blocks.

Volume_Correction_FB

Description

This function block computes the corrected Volume for the given CTL and CPL.

Input

Parameter	Data type	Description
CTL	LREAL	Correction factor for effects of temperature on the liquid.
CPL	LREAL	Correction factor for effects of pressure on the liquid.
MeteredVol	LREAL	Recorded metered volume by the master meter.

Output

Parameter	Data type	Description
CorrVol_AtCtlCpl	LREAL	The corrected volume at this CTL and CPL.

Flowrate_Calc

Description

Function block to compute the flow rate.

Input

Parameter	Data Type	Description
IncCalcVal	LREAL	The incremental calculated volume or mass to compute the flow rate.
ExecTime	INT	Execution time in seconds.

Output

Parameter	Data Type	Description
Flowrate	LREAL	Flow rate computed in m^3/hr.

CHAPTER

7 CRC

Description

This function block is used to calculate CRC-16.

Input

Parameter	Data type	Description
ENABLE	BOOL	Enable: If TRUE, the function block is enabled and workable.
INPUT	Array of USINT, UINT, UDINT, LINT, REAL or LREAL;	User defined data type. The size of the array depends on the number of the registers to read. The end user should define a data type as shown below: TYPE VariableName: array[1..LENGTH] of USINT/UINT/UDINT/LINT/REAL/LREAL; END_TYPE The end user can read the data of a specific register by using the suffix.
LENGTH	UINT	Maximum number of bytes to be calculated. Default = 0: The DATA parameter determines the length of the data to be calculated CRC. The maximum size is 1024 bytes.
START_VAL	UINT	Define the values that are used for initialization of a CRC value for common used calculation methods. A list of CRC start code: CRC-START-16: 0x0000 CRC-START-MODBUS: 0xFFFF

Parameter	Data type	Description
		CRC-START-XMODEM: 0x0000 CRC-START-CCITT-1D0F: 0x1D0F CRC-START-CCITT-FFFF: 0xFFFF (X.25, V.41, HDLC, Bluetooth, SD, many others; known as CRC-CCITT) CRC-START-KERMIT: 0x0000 CRC-START-SICK: 0x0000 CRC-START-DNP: 0x0000
POLYNOMIALS	UINT	Define the polynomials for some well known CRC calculations. A list of CRC polynomials: CRC-16: 0xA001 (Modbus use this) CRC-CCITT: 0x1021 CRC-DNP: 0xA6BC CRC-KERMIT: 0x8408 CRC-SICK: 0x8005

Output

Parameter	Data type	Description
CRC_HIGH	USINT	CTC value high position
CRC_LOW	USINT	CTC value low postion
ERR_FLAG	BOOL	It would be set true if there is an error.
GEN_ERR	USINT	0: Communication succeeded 1: The input parameter given to the function block is invalid.

CHAPTER

8 ETHERNETIP

limitation for using the function blocks:

- Up to ten IP addresses
- Read: 65 arrays; 400 single variables
- Write: 65 arrays; 135 single variables

The following function blocks and items are available:

Function Blocks	Short Description
ETHERNETIP_RD	It is used to read a variable value from a peer to peer controller through the tag name.
ETHERNETIP_WR	It is used to write a value to a peer to peer controller through the tag name.

Related Topics:

Error Code	See EtherNet/IP Function Block Error Codes for more information.

ETHERNETIP_RD

Description

This function block reads a variable value from a peer-to-peer controller by the tag name.

Input

Parameter	Data type	Description
ENABLE	BOOL	Enable: If it is set to TRUE, the function block is enabled.
TAG	STRING	The name of the variable that the function block will read from the target controller. **TIP:** Up to 80 characters can be obtained from TAG.

Parameter	Data type	Description
ELE_NUM	USINT	Number of elements for array type variable. • If it is a single or scalar variable, set this parameter as 1. • If it is a arrayed variable, set this parameter as more than 1.
ELE_ DATATYPE	USINT	Data type of the value that the function block will read: DATATYPE_BOOL (0x01) DATATYPE_SINT (0x02) DATATYPE_INT (0x03) DATATYPE_DINT (0x04) DATATYPE_USINT (0x05) DATATYPE_UINT (0x06) DATATYPE_UDINT (0x07) DATATYPE_REAL (0x08)
IP_ADDR	STRING	The IP address of the target controller or adapter which connects with PLC. **TIP:** Up to ten IP addresses can be added in one project.
SLOT	USINT	The slot number of the rack which inserted the target controller via an adapter.
SEND_ FLAG	BOOL	Set it as true and when RDY_FLAG is true, the function block will send the request to read. Before last communication is finished, even if it is set as true, the request won't be sent.

Output

Parameter	Data types	Description
RDY_FLAG	BOOL	True: last communication is finished. The function block is ready for the next communication. False: command request is being sent or received.
DONE	BOOL	It indicates that the response data is received successfully and usable.
ERR_FLAG	BOOL	True: there is an error. False: there is no error.
GEN_STS	USINT	General status and vendor's status
EXT_STS	UINT	Vendor's external status
GEN_ERR	USINT	General errors. See EtherNet/IP Function Block Error Codes for more information.

Input and Output

Parameter	Data types	Description
VALUE	ANY	Buffer for the data to be read (for read-output parameter) Value= ELE_NUM*size of (data type) See the follow table for the size of each data type: Tata type Size of the data type DATATYPE_BOOL (0x01) 1 byte DATATYPE_SINT (0x02) 1 byte DATATYPE_INT (0x03) 2 bytes DATATYPE_DINT (0x04) 4 bytes DATATYPE_USINT (0x05) 1 byte DATATYPE_UINT (0x06) 2 bytes DATATYPE_UDINT (0x07) 4 bytes

Parameter	Data types	Description
		DATATYPE_REAL (0x08) 4 bytes Maximum is 512 bytes.

See the following datatype of parameter Value for details:

DATA_TYPE

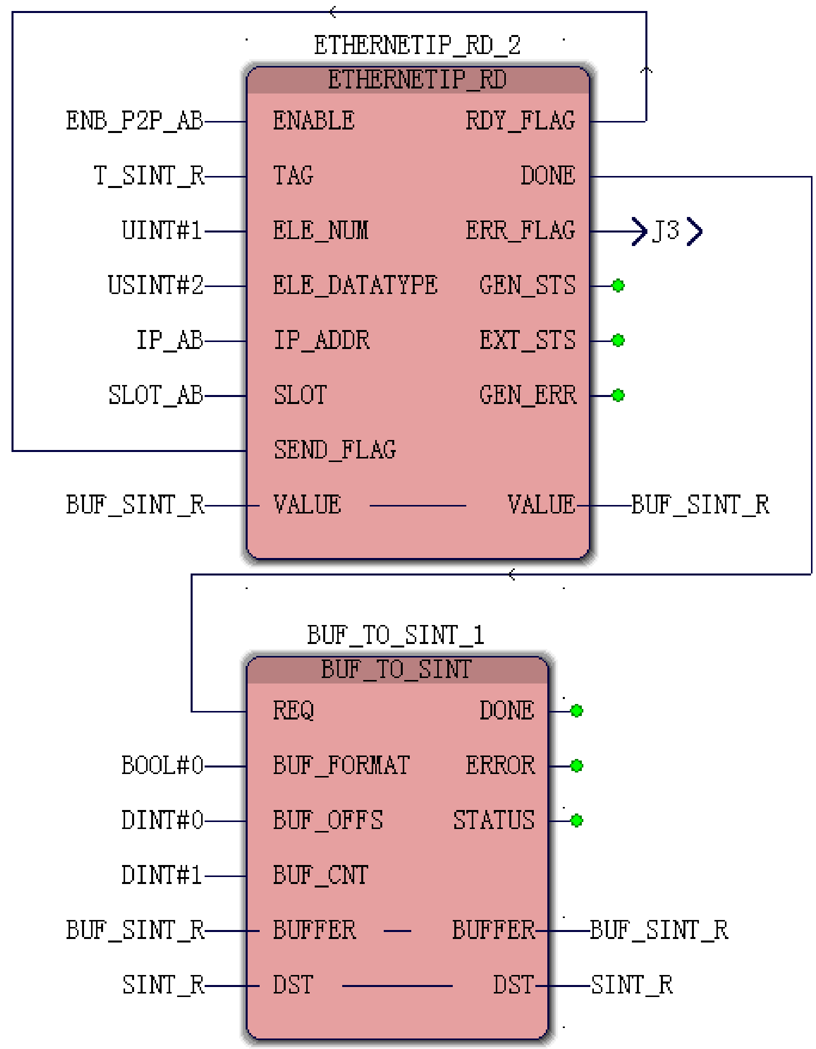
TYPE (* Array data type for tag data read/write *)

EIP_TAG_DATA: ARRAY[1..512] of BYTE;

END_TYPE

Programming Example

Use this function block to read Tag name to target Controller. At the same time, use Function block HW_BITS_TO_SINT to transfer the data type which can be used.



1. Double-click the pin-outs of the function block to assign variables. The **Variable Properties** dialog appears.

   Select the **Name**, **Data Type** and **Usage** from the list. for each of the Input pin.

2. Drag the pin from **SEND_FLAG** to **RDY_FLAG** and drag pin from **Done** to **REQ** of BUF_TO_SINT.

3. Click **OK**.

4. Click **Make** from the toolbar to compile the programs.

5. Click **Download** from the toolbar to download the compiled programs of HART to the controller.

ETHERNETIP_WR

Description

This function block writes a variable value of a peer-to-peer controller by the tag name.

Input

Parameter	Data types	Description
ENABLE	BOOL	Enable: If it is set to TRUE, the function block is enabled.
TAG	STRING	The name of the variable that the function block will write from of target controller. **TIP:** Up to 80 characters can be obtained from TAG.
ELE_NUM	USINT	Number of elements for array type variable. • If it is a single or scalar variable, set this parameter as 1. • If it is a arrayed variable, set this parameter as more than 1.
ELE_ DATATYPE	USINT	Data type of the value that the function block will write: DATATYPE_BOOL (0x01) DATATYPE_SINT (0x02) DATATYPE_INT (0x03) DATATYPE_DINT (0x04) DATATYPE_USINT (0x05) DATATYPE_UINT (0x06) DATATYPE_UDINT (0x07) DATATYPE_REAL (0x08)
IP_ADDR	STRING	The IP address of the target controller or adapter which connects with PLC.

Parameter	Data types	Description
SLOT	USINT	The slot number of the rack which inserted the target controller via an adapter.
SEND_ FLAG	BOOL	Set it as true and when RDY_FLAG is true, the function block will send the request to write. Before last communication is finished, even if it is set as true, the request won't be sent.

Output

Parameter	Data types	Description
RDY_FLAG	BOOL	True: last communication is finished. The function block is ready for the next communication. False: command request is being sent or received.
DONE	BOOL	It indicates that the response data is received successfully and usable.
ERR_FLAG	BOOL	True: there is an error. False: there is no error.
GEN_STS	USINT	General status and vendor's status
EXT_STS	UINT	Vendor's external status
GEN_ERR	USINT	General errors. See EtherNet/IP Function Block Error Codes for more information.

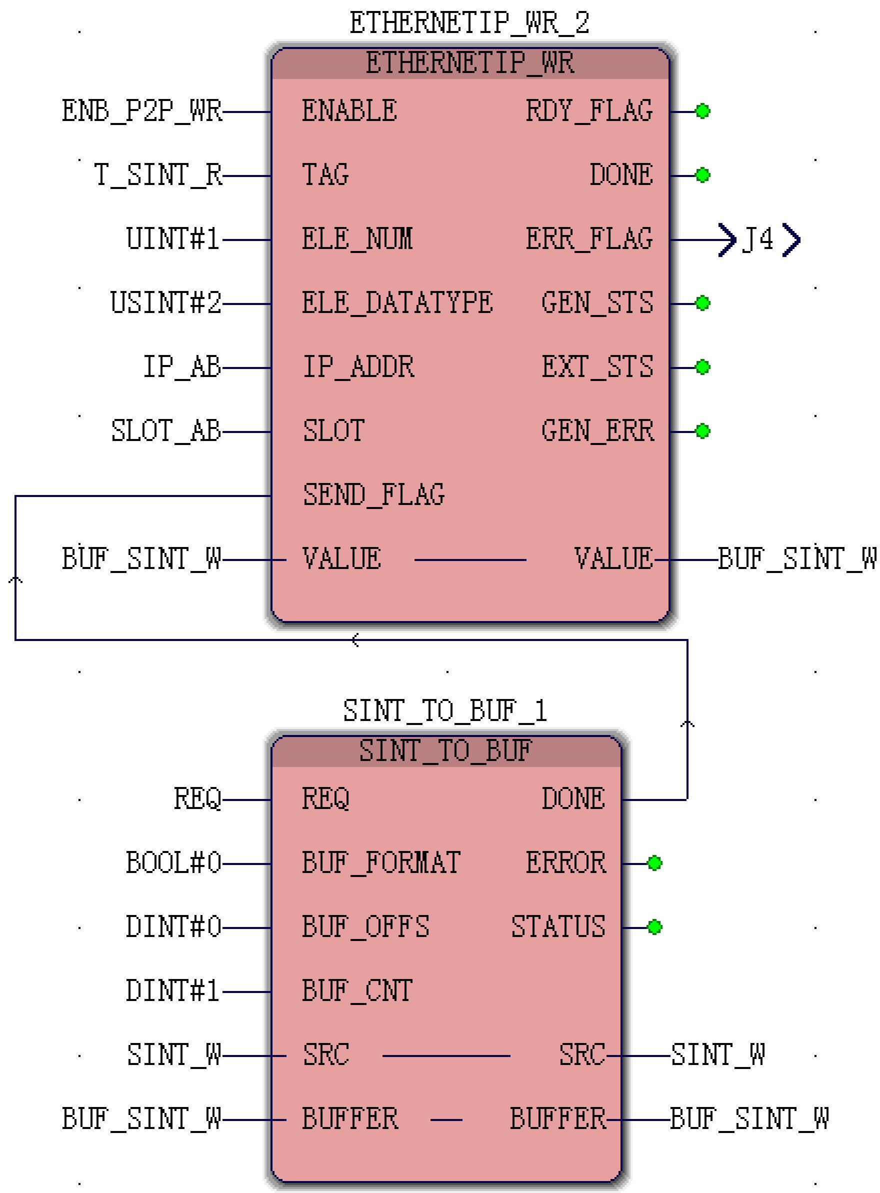
Input and Ouput

Parameter	Data types	Description
VALUE	ANY	Buffer for the data to write (for read-output parameter) Value= ELE_NUM*size of (data type) See the follow table for the size of each data type: Tata type Size of the data type

Parameter	Data types	Description
		DATATYPE_BOOL (0x01) 1 byte DATATYPE_SINT (0x02) 1 byte DATATYPE_INT (0x03) 2 bytes DATATYPE_DINT (0x04) 4 bytes DATATYPE_USINT (0x05) 1 byte DATATYPE_UINT (0x06) 2 bytes DATATYPE_UDINT (0x07) 4 bytes DATATYPE_REAL (0x08) 4 bytes Maximum is 512 bytes.

Programming Example

Use this function block to write Tag name to target Controller. At the same time, use Function block HW_SINT_TO_BUF to transfer the data type which can be used.



1. Double-click the pin-outs of the function block to assign variables. The **Variable Properties** dialog appears.

   Select the **Name**, **Data Type** and **Usage** from the list. for each of the Input pin.
2. Drag the pin from **SEND_FLAG** to **Done** of SINT_TO_BUF.
3. Click **OK**.
4. Click **Make** from the toolbar to compile the programs.

5. Click **Download** from the toolbar to download the compiled programs of HART to the controller.

EtherNet/IP Function Block Error Codes

Refer to the following table for EtherNet/IP function block error codes:

Error Code	Description
0	N/A
1	Input parameter is invalid.
2	Time out, no response is received.
3	Internal Process Commuication time out (eClr and EtherNet/IP process)
4	Invalid request
33	exceed maximum function block quantity
34	exceed maximum tag name length
35	exceed maximum data value size
36	Invalid data type
37	exceed maximum quantity of devices IP addresses

CHAPTER

9

HWFBLIB

The following HWFWLib function blocks are available:

Function Block	Description
HWAI	Analog input channel block used to initialize analog data type for use in control function blocks.
HWAI2PV	Analog input channel block used to convert from analog_input_ type to analog_type for regulatory control function blocks. This is a replacement for HWAI function block in R110 onwards.
HWAO	Analog output channel block used to connect regulatory control blocks such as HWPID or HWAUTOMAN to an analog output and provide back initialization to connected control block.
HWAUTOMAN	It is used to define user-specified gain and bias as well as a calculated bias to the output. Shall provide control initialization and override feedback processing.
HWCV2AO	Analog output channel block used to connect regulatory control blocks such as HWPID or HWAUTOMAN to an analog output and provide back initialization to connected control block.
HWDACA	Provides conversion from analog input data type to regulatory control data type and provides alarming, scaling, filtering and low cutoff processing. Typically this function block will be connected to an analog input channel. The PV output would be connected to regulatory control function blocks such as HWPID.
HWFANOUT	This function shall be used to provide one input and up to four initializable outputs. Shall allow separate gain and bias for each output. Typical use is for split range outputs. An AUTOMAN FB should be used between the output of the FANOUT and final Analog Output.
HWIOSTS	This function block is used to decode I/O channel status value and provide a I/O channel Bad flag.
HWMCC	It provides device control for a motor control to stop, forward (run) and reverse a motor. The block shall contain built-in structure for handling interlocks. The block has forward (or run) and reverse input indications with forward (run) and reverse outputs for control.

Function Block	Description
HWMLV	Device control for a fail last valve. The block contains built-in structure for handling interlocks. This block can handle single or dual limit switch position indication with dual pulsed outputs. The outputs shall be energised until valve reaches commanded state or maximum travel time is reached. Operation commands are pulsed.
HWMOV	It provides device control for a motor operated valve. The block shall contain built-in structure for handling interlocks. This block can handle single or dual limit switch position indication with dual pulsed outputs. The outputs shall be energised until valve reaches commanded state or maximum travel time is reached.
HWNOMINATION	This function block shall be provided with a daily nomination for each day of the week and shall calculate the desired flow setpoint to meet the nomination value. The setpoint is used to provide a remote setpoint to a flow control PID in cascade mode.
HWOVERSEL	This function shall be used to provide override select of either the maximum or minimum of up to four initializable inputs.
HWPI	This function block is connected to a pulse input channel and outputs a delta pulse count suitable for metering calculations such as AGA7/9.
HWPIACC	This function block is connected to a pulse input channel and outputs a delta pulse count suitable for metering calculations such as AGA7/9 and has a count accumulator that can be enabled and reset.
HWPID	It supports PI, PD, PID algorithms as function. Shall accept 2 analog inputs- process variable (PV) and set point (SP) and produces output calculated to reduce the difference between PV and SP.
HWRATIOCTL	It accepts the actual value of the controlled flow (X1), the actual value of the uncontrolled flow (X2) and the target ratio between the flows (SP), and shall calculate the target value of the controlled flow (OP) and the actual ratio between the flows (PV) as outputs.
HWRETAIN	This function block retains a global variable on a warm or cold start and after a reboot. This FB should be used for any user modified values such as tuning constants of PID or for accumulators on a totaliser FB. This function block requires that connected global variable is assigned an address.

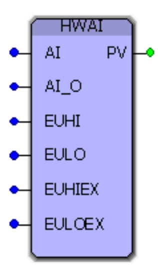
Function Block	Description
HWSDV	HWSDV provide device control for a solenoid operated shutdown valve. The block shall contain built-in structure for handling interlocks. This block can handle single or dual limit switch position indication with a single latched control output.
HWSLEWRATE	Slew Rate is the maximum rate of change required to drive the output from full OFF (0%-typically 0 mA or 4 mA) to full ON (100%-typically 20 mA). The block will convert this to a maximum change of the milliamp output per execution cycle of this block.
HWSPLITRNG	The Split Range function block is used in conjunction with the FANOUT function block. This block translates split range settings to gain and bias settings suitable for FANOUT.
HWTOTALIZER	It is used to accumulate flows. Shall periodically integrate or accumulate an input value to a totalised value and shall set status flags to indicate when accumulator value has reached the user specified target value.
HWTOT_LREAL_TO_REAL	This function block converts totaliser LREAL to Totaliser Real data.

HWAI

Description

Analog input channel block used to initialize analog data type for use in control function blocks.

> **ATTENTION:** This block is replaced by HWAI2PV or HWDACA from ControlEdge RTU R110 onwards.



Input

Parameter	Data type	Description
AI	REAL	Analog Input value from analog input channel
AI_O	BOOL	Analog Input channel Overrange status
EUHI	REAL	PV High Range (Only used to replicate range from the builder)
EULO	REAL	PV Low Range (Only used to replicate range from the builder)
EUHIEX	REAL	PV High Extended Range (Only used to replicate range from the builder)
EULOEX	REAL	PV Low Extended Range (Only used to replicate range from the builder)

Output

Parameter	Data type	Description
PV	Analog_ Type	Process variable analog data used to connect to control function block

HW_BITS_TO_BYTE

Description

This function block creates a byte from 8 individual bits.

Input

Parameter	Data type	Description
B(0-7)	BOOL	Bit (0-7) of a byte

Output

Parameter	Data type	Description
OUT	BYTE	Byte created by 8 individual bits

HW_BITS_TO_SINT

Description

This function block creates a SINT from 8 individual bits .

Input

Parameter	Data type	Description
B(0-7)	BOOL	Bit (0-7) of a sint

Output

Parameter	Data type	Description
OUT	SINT	SINT created by eight individual bits

HW_BITS_TO_USINT

Description

This function block creates a USINT from 8 individual bits .

Input

Parameter	Data type	Description
B (0-7)	BOOL	Bit (0-7) of a usint

Output

Parameter	Data type	Description
OUT	USINT	USINT created by eight individual bits

HW_BYTE_TO_BITS

Description

This function block extracts 8 individual bits from a byte.

Input

Parameter	Data type	Description
IN	BYTE	Raw value to be extracted

Output

Parameter	Data type	Description
B (0-7)	BOOL	Bit (0-7) of a byte

HW_BYTES_TO_DINT

Description

This function block creates a DINT from four individual bytes.

Input

Parameter	Data type	Description
B (0-3)	BYTE	Byte (0-3) of DINT

Output

Parameter	Data type	Description
OUT	DINT	DINT created by four individual bytes

HW_BYTES_OF_INT

Description

This function block creates an INT from two individual bytes.

Input

Parameter	Data type	Description
B (0-1)	BYTE	Byte (0-1) of INT

Output

Parameter	Data type	Description
OUT	INT	INT created by two individual bytes

HW_BYTES_OF_UDINT

Description

This function block creates a UDINT from four individual bytes.

Input

Parameter	Data type	Description
B (0-3)	BYTE	Byte (0-3) of UDINT

Output

Parameter	Data type	Description
OUT	UDINT	UDINT created by four individual bytes

HW_BYTES_OF_UINT

Description

This function block creates a uint from two individual bytes.

Input

Parameter	Data type	Description
B (0-1)	BYTE	Byte (0-1) of UINT

Output

Parameter	Data type	Description
OUT	UINT	UINT created by two individual bytes

HW_BYTES_TO_DWORD

Description

This function block creates a DWORD from four individual bytes.

Input

Parameter	Data type	Description
B (0-3)	BYTE	Byte (0-3) of DWORD

Output

Parameter	Data type	Description
OUT	DWORD	DWORD created by four individual bytes

HW_BYTES_TO_WORD

Description

This function block creates a WORD from two individual bytes.

Input

Parameter	Data type	Description
B (0-1)	BYTE	Byte (0-1) of WORD

Output

Parameter	Data type	Description
OUT	WORD	WORD created by four individual bytes

HW_BYTE_OF_DINT

Description

This function block extracts a single byte from DINT with the position.

Input

Parameter	Data type	Description
IN	DINT	Raw value to be extracted
N	BYTE	The position to be extracted, start from 0.

Output

Parameter	Data type	Description
OUT	BYTE	Extracted byte

HW_BYTE_OF_INT

Description

This function block extracts a single byte from int with the position.

Input

Parameter	Data type	Description
IN	INT	Raw value to be extracted
N	BYTE	The position to be extracted, start from 0.

Output

Parameter	Data type	Description
OUT	BYTE	Extracted byte

HW_BYTE_OF_DWORD

Description

This function block extracts a single byte from dword with the position.

Input

Parameter	Data type	Description
IN	DWORD	Raw value to be extracted
N	BYTE	The position to be extracted, start from 0.

Output

Parameter	Data type	Description
OUT	BYTE	Extracted byte

HW_BYTE_OF_UDINT

Description

This function block extracts a single byte from udint with the position.

Input

Parameter	Data type	Description
IN	UDINT	Raw value to be extracted
N	BYTE	The position to be extracted, start from 0.

Output

Parameter	Data type	Description
OUT	BYTE	Extracted byte

HW_BYTE_OF_UINT

Description

This function block extracts a single byte from uint with the position.

Input

Parameter	Data type	Description
IN	UINT	Raw value to be extracted
N	BYTE	The position to be extracted, start from 0.

Output

Parameter	Data type	Description
OUT	BYTE	Extracted byte

HW_BYTE_OF_WORD

Description

This function block extracts a single byte from word with the position.

Input

Parameter	Data type	Description
IN	WORD	Raw value to be extracted
N	BYTE	The position to be extracted, start from 0.

Output

Parameter	Data type	Description
OUT	BYTE	Extracted byte

HW_SINT_TO_BITS

Description

This function block extracts 8 bits from a sint.

Input

Parameter	Data type	Description
IN	SINT	Raw value to be extracted

Output

Parameter	Data type	Description
B(0-7)	BOOL	Bit (0-7) of a sint

HW_SINT_OF_DINT

Description

This function block extracts a single sint from dint with the position.

Input

Parameter	Data type	Description
IN	DINT	Raw value to be extracted
N	BYTE	The position to be extracted

Output

Parameter	Data type	Description
OUT	SINT	Extracted sint

HW_SINT_OF_DWORD

Description

This function block extracts a single sint from dword with the position.

Input

Parameter	Data type	Description
IN	DWORD	Raw value to be extracted
N	BYTE	The position to be extracted

Output

Parameter	Data type	Description
OUT	SINT	Extracted sint

HW_SINT_OF_INT

Description

This function block extracts a single sint from int with the position.

Input

Parameter	Data type	Description
IN	INT	Raw value to be extracted
N	BYTE	The position to be extracted

Output

Parameter	Data type	Description
OUT	SINT	Extracted sint

HW_SINT_OF_UDINT

Description

This function block extracts a single sint from udint with the position.

Input

Parameter	Data type	Description
IN	UDINT	Raw value to be extracted
N	BYTE	The position to be extracted

Output

Parameter	Data type	Description
OUT	SINT	Extracted sint

HW_SINT_OF_UINT

Description

This function block extracts a single int from uint with the position.

Input

Parameter	Data type	Description
IN	UINT	Raw value to be extracted
N	BYTE	The position to be extracted

Output

Parameter	Data type	Description
OUT	SINT	Extracted sint

HW_SINT_OF_WORD

Description

This function block extracts a single sint from word with the position.

Input

Parameter	Data type	Description
IN	WORD	Raw value to be extracted
N	BYTE	The position to be extracted

Output

Parameter	Data type	Description
OUT	SINT	Extracted sint

HW_SINTS_TO_DINT

Description

This function block creates a dint from four individual sints.

Input

Parameter	Data type	Description
B (0-3)	SINT	Byte (0-3) of DINT

Output

Parameter	Data type	Description
OUT	DINT	DINT created by four individual sints

HW_SINTS_TO_DWORD

Description

This function block creates a dword from four individual sints.

Input

Parameter	Data type	Description
B (0-3)	SINT	Byte (0-3) of DWORD

Output

Parameter	Data type	Description
OUT	DWORD	DWORD created by four individual sints

HW_SINTS_TO_INT

Description

This function block creates an int from two individual sints.

Input

Parameter	Data type	Description
B (0-1)	SINT	Byte (0-1) of INT

Output

Parameter	Data type	Description
OUT	INT	INT created by two individual sints

HW_SINTS_TO_UDINT

Description

This function block creates a udint from four individual sints.

Input

Parameter	Data type	Description
B (0-3)	SINT	Byte (0-3) of UDINT

Output

Parameter	Data type	Description
OUT	UDINT	UDINT created by four individual sints

HW_SINTS_TO_UINT

Description

This function block creates a uint from two individual sints.

Input

Parameter	Data type	Description
B (0-1)	SINT	Byte (0-1) of UINT

Output

Parameter	Data type	Description
OUT	UINT	UINT created by two individual sints

HW_SINTS_TO_WORD

Description

This function block creates a word from two individual sints.

Input

Parameter	Data type	Description
B (0-1)	SINT	Byte (0-1) of WORD

Output

Parameter	Data type	Description
OUT	WORD	WORD created by two individual sints

HW_USINT_OF_INT

Description

This function block extracts a single usint from int with the position.

Input

Parameter	Data type	Description
IN	INT	Raw value to be extracted
N	BYTE	The position to be extracted

Output

Parameter	Data type	Description
OUT	USINT	Extracted usint

HW_USINT_OF_UDINT

Description

This function block extracts a single usint from udint with the position.

Input

Parameter	Data type	Description
IN	UDINT	Raw value to be extracted
N	BYTE	The position to be extracted

Output

Parameter	Data type	Description
OUT	USINT	Extracted usint

HW_USINT_OF_DINT

Description

This function block extracts a single usint from dint with the position.

Input

Parameter	Data type	Description
IN	DINT	Raw value to be extracted
N	BYTE	The position to be extracted

Output

Parameter	Data type	Description
OUT	USINT	Extracted usint

HW_USINT_OF_DWORD

Description

This function block extracts a single usint from dword with the position.

Input

Parameter	Data type	Description
IN	DWORD	Raw value to be extracted
N	BYTE	The position to be extracted

Output

Parameter	Data type	Description
OUT	USINT	Extracted usint

HW_USINT_OF_UINT

Description

This function block extracts a single usint from uint with the position.

Input

Parameter	Data type	Description
IN	UINT	Raw value to be extracted
N	BYTE	The position to be extracted

Output

Parameter	Data type	Description
OUT	USINT	Extracted usint

HW_USINT_OF_WORD

Description

This function block extracts a single usint from word with the position.

Input

Parameter	Data type	Description
IN	WORD	Raw value to be extracted
N	BYTE	The position to be extracted

Output

Parameter	Data type	Description
OUT	USINT	Extracted usint

HW_USINT_TO_BITS

Description

This function block extracts 8 bits from a usint.

Input

Parameter	Data type	Description
IN	USINT	Raw value to be extracted

Output

Parameter	Data type	Description
B(0-7)	BOOL	Bit (0-7) of a usint

HW_USINTS_TO_DINT

Description

This function block creates a dint from four individual usints.

Input

Parameter	Data type	Description
B (0-3)	USINT	Byte (0-3) of DINT

Output

Parameter	Data type	Description
OUT	DINT	DINT created by four individual usints

HW_USINTS_TO_DWORD

Description

This function block creates a dword from four individual usints.

Input

Parameter	Data type	Description
B (0-3)	USINT	Byte (0-3) of DWORD

Output

Parameter	Data type	Description
OUT	DWORD	DWORD created by four individual usints

HW_USINTS_TO_INT

Description

This function block creates an int from two individual usints.

Input

Parameter	Data type	Description
B (0-1)	USINT	Byte (0-1) of INT

Output

Parameter	Data type	Description
OUT	INT	INT created by two individual usints

HW_USINTS_TO_UDINT

Description

This function block creates a udint from four individual usints.

Input

Parameter	Data type	Description
B (0-3)	USINT	Byte (0-3) of UDINT

Output

Parameter	Data type	Description
OUT	UDINT	UDINT created by four individual usints

HW_USINTS_TO_UINT

Description

This function block creates a uint from two individual usints.

Input

Parameter	Data type	Description
B (0-1)	USINT	Byte (0-1) of UINT

Output

Parameter	Data type	Description
OUT	UINT	UINT created by two individual usints

HW_USINTS_TO_WORD

Description

This function block creates a word from two individual usints.

Input

Parameter	Data type	Description
B (0-1)	USINT	Byte (0-1) of WORD

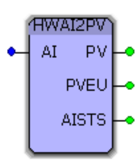
Output

Parameter	Data type	Description
OUT	WORD	WORD created by two individual usints

HWAI2PV

Description

Analog input channel block used to convert from analog_input_type to analog_type for regulatory control function blocks. This is a replacement for HWAI function block from ControlEdge RTU R110 onwards. If additional analog processing is required such as filtering, rescaling or analog alarming, the HWDACA function block can be used instead of HWAI2PV.



Input

Parameter	Data Type	Description
AI	Analog_Input_Type	Analog Input value from analog input channel

Output

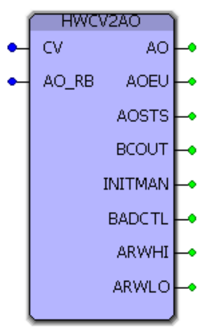
Parameter	Data Type	Description
PV	Analog_ Type	Process variable analog data used to connect to regulatory control function blocks such as HWPID.
PVEU	REAL	PV value from analog input.
AISTS	STRING	Analog Input Channel status message

HWAO

Description

Analog output channel block used to connect control blocks to an analog output and provide back initialization to connected control block.

> **ATTENTION:** This block is replaced by HWCV2AO from ControlEdge RTU R110 onwards.



Parameter	Data Type	Description
CV	Analog_Type	Control Analog data from Control Block such as HWPID or HWAUTOMAN
AO_RB	Analog_Output_ Readback_Type	Analog Output Readback connection.

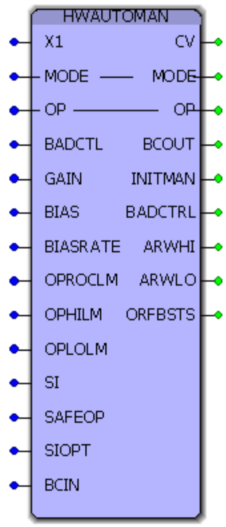
Ouput

Parameter	Data Type	Description
AO	Analog_ Output_ Type	Analog Output data to be connected to analog ouput channel
AOEU	REAL	Analog output value
AOSTS	STRING	Analog output channel status message
BCOUT	BackCalc_ Type	BackCalc information used to initialise upstream control block based on open wire and range exceeded flags from analog output channel.
INITMAN	BOOL	FB InitMan has been requested by downstream block
BADCTRL	BOOL	Bad Control Option is active
ARWHI	BOOL	FB is in high windup status
ARWLO	BOOL	FB is in low windup status

HWAUTOMAN

Description

It is used to define user-specified gain and bias as well as a calculated bias to the output. It provides control initialization and override feedback processing.



Input

Parameter	Data Type	Description
X1	Analog_Type	Process variable input. AI_Type contains value and quality flags
Mode	INT	Sets Mode. 0 - Manual OUT = OP 2 - Cascade OUT = GAIN * IN + BIAS + BIAS_FLOAT
OP	REAL	Manual Output
BADCTL	INT	Bad Control Option as per C200/C300 0. No Shed 1. Shed Hold 2. Shed Low 3. Shed High 4. Shed Safe OP
GAIN	REAL	OUT= GAIN * IN + BIAS
BIAS	REAL	OUT= GAIN * IN + BIAS
ROCLM	REAL	Maximum rate of change of Control Variable output in %/min

Parameter	Data Type	Description
		Default – 0 , no rate limiting
BIASRATE	REAL	Rate in %/min to reduce floating bias to eliminate bumps on transfer from Manual to Cascade. If BIASRATE = 0.0, no floating bias is applied to OP which may result in a bump.
OPROCLM	REAL	Maximum rate of change of Control Output in %/min Default – 0 , no rate limiting
OPHILM	REAL	Maximum Output
OPLOLM	REAL	Minimum Output Default – 0%
SI	BOOL	Safety Interlock. False – No shutdown (default) True – Shutdown using SIOPT
SAFEOP	REAL	Shutdown Control Variable Target value.
SIOPT	INT	Safety Option as per C200/C300 0. No Shed 1. Shed Hold 2. Shed Low 3. Shed High 4. Shed Safe OP
BCIN	BackCalc_Type	Back Calculation Input. This comes from Back Calculation Output of downstream block

Output

Parameter	Data type	Description
CV	Analog_ Type	Control Variable that is normally used to drive the analog output to a control device.
BCOUT	BackCalc_	Back Calculation Output. This goes to Back Calculation

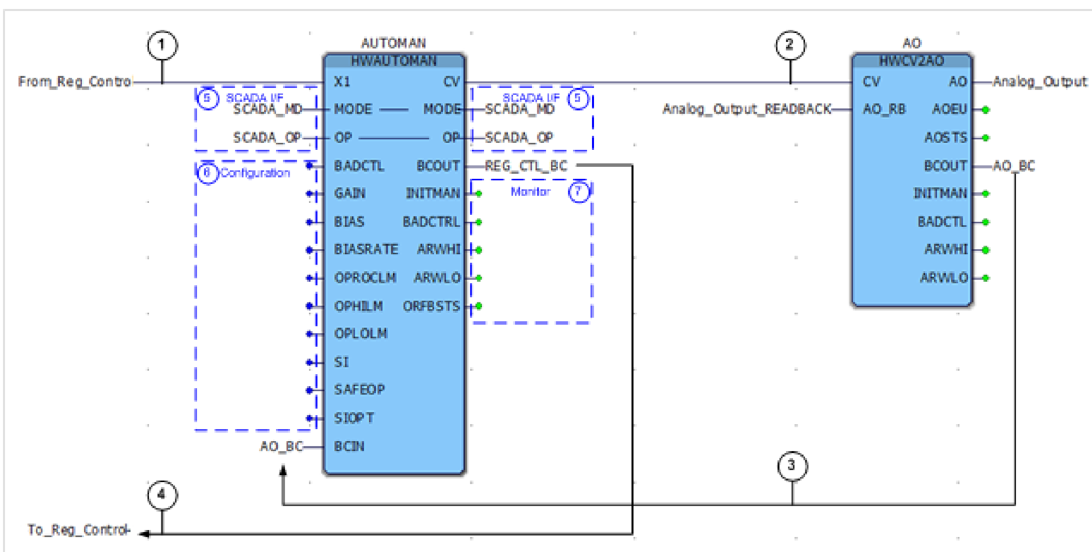
Parameter	Data type	Description
	Type	Input of upstream block
INITMAN	BOOL	FB InitMan has been requested by downstream block
BADCTRL	BOOL	Bad Control Option is active
ARWHI	BOOL	FB is in high windup status
ARWLO	BOOL	FB is in low windup status
ORFBSTS	BOOL	FB is using Override Feedback value from OVRSEL

The HWAUTOMAN function block is normally used for final control before an analog output where a complex control strategy is used such as override select or fanout/split range.

In the case of override select, the HWAUTOMAN provides a common point to control mode of final control output.

For fanout/split range control, the Gain and Bias provide a means to rescale control output. The Bias rate provides a means to smooth bumps when transitioning modes.

Below is an example of HWAUTOMAN used to connect a regulatory control strategy to an analog out.



The following describes the main connections in the figure above.

1. The input X1 is connected to the output of a control strategy. This connection will be an Analog_Type from a control variable output of another regulatory control function block like HWPID or HWOVRSEL. This connection carries value, range and status information.
2. The CV parameter connection is used to send output control data to the Analog Output Channel (HWCV2AO) block. The CV parameter is an Analog_Type as per the X1.
3. Use the BCOUT/BCIN connection to carry secondary data from the CV2AO block to the HWAUTOMAN block. Note that CV2AO requires the analog output read back connection to provide secondary data such as status of analog output channel. The secondary data in the BACKCALC_TYPE data connection between CV2AO BCOUT pin and the HWPID BCIN includes the following information.
   - Anti-Reset Windup Status (ARWHI, ARWLO): Indicates if the secondary's initialize input (which is this block's output) is at its high or low limit.
   - Initialization Request Flag (INITMAN): Used to request initialization. If the flag is set by CV2AO, the AUTOMAN block initializes itself
   - Initialization Value (INITVAL): Used for initialization when INITMAN true.
   - Override Status (ORFBSTS): If a block is in an override strategy, this flag indicates whether it is the selected strategy or not. If the block is in an unselected strategy, it uses Override Feedback Value (ORFBVAL) to initialize Control value; this is calculated to prevent “wind-up” if this AUTOMAN block is unselected.
4. Use the BCOUT/BCIN connection to carry secondary data from the HWAUTOMAN block to the upstream function block connected to X1 to prevent windup and to request initialisation when required.
5. The SCADA control interface for the function block is typically mapped to an analog point OP and MD parameters. The PV for this point can be linked to the OP of the HWAUTOMAN function block or it could be connected to the Analog Output Readback value to reflect the final analog output control value.

6. This group of parameters determines how the control variable, CV is calculated and how it will behave for bad control or safety interlock conditions. Please refer to following sections.
7. These pins can be used for monitoring control state of function block to see if it is in windup, initialisation or override conditions.

Mode Operation

The function block has modes Manual (MD=0) and Cascade (MD=2). There is no Auto mode.

In Manual Mode, the Control Variable will track the SCADA OP value entered by an operator.

In Cascade mode, the Control Variable will be calculated from input X1 as follows:

CV = GAIN * X1 + BIAS + BIAS_FLOAT

Where BIAS_FLOAT is calculated internally on mode transition from Manual to Cascade to ensure a bumpless transfer of CV. BIASRATE determines how fast the BIAS_FLOAT is reduced to zero. If no floating bias is required, BIASRATE should be set to zero.

Rate of Change of Output

The maximum rate of change of the control output can be set by OPROCLM. The units are defined in %/Minute. To disable rate of change limiting, set the value to zero. Rate limiting is not applied when mode is Manual.

Bad Control Options

The BADCTL option determines how the AUTOMAN block will behave if there is an error in X1 caused by any fault or configuration error in the Analog Input chain connected to the AUTOMAN block. Bad control is invoked if

- The X1 status flag is set by an upstream function block.
- The X1 value exceeds EUHIEX or EULOEX extended range
- The X1 value is NaN

If the output BADCTRL is true, bad control processing occurs based on the BADCTL option values shown below.

0. (default) No Shed – CV will stop calculating and hold last valid value. Mode will remain unchanged.

1. Shed Hold – CV will stop calculating and hold last valid value and Mode will shed to Manual.

2. Shed Low – CV will be set to 0% and Mode will shed to Manual.

3. Shed High – CV will be set to 100% and Mode will shed to Manual.

4. Shed Safe OP – CV will be set to value defined by SAFEOP and Mode will shed to Manual.

Safety Interlock Options

The safety interlock option (SIOPT) determines how the AUTOMAN block will behave if the Safety Interlock input (SI) is set to true.

The values of SIOPT are shown below.

0. (default) No Shed – CV will stop calculating and hold last valid value. Mode will remain unchanged.

1. Shed Hold – CV will stop calculating and hold last valid value and Mode will shed to Manual.

2. Shed Low – CV will be set to 0% and Mode will shed to Manual.

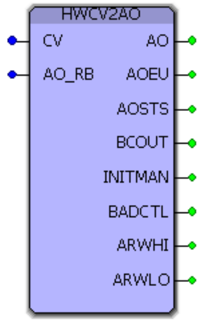
3. Shed High – CV will be set to 100% and Mode will shed to Manual.

4. Shed Safe OP – CV will be set to value defined by SAFEOP and Mode will shed to Manual.

HWCV2AO

Description

Analog output channel block used to connect regulatory control blocks such as HWPID or HWAUTOMAN to an analog output and provide back initialization to connected control block.



Input

Parameter	Data Type	Description
CV	Analog_Type	Control Analog data from Control Block such as HWPID or HWAUTOMAN
AO_RB	Analog_Output_Readback_Type	Analog Output Readback connection.

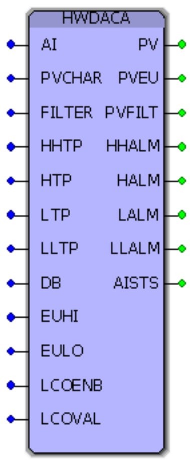
Output

Parameter	Data Type	Description
AO	Analog_Output_Type	Analog Output data to be connected to analog ouput channel
AOEU	REAL	Analog output value
AOSTS	STRING	Analog output channel status message
BCOUT	BackCalc_Type	BackCalc information used to initialise upstream control block based on open wire and range exceeded flags from analog output channel.
INITMAN	BOOL	FB InitMan has been requested by downstream block
BADCTRL	BOOL	Bad Control Option is active
ARWHI	BOOL	FB is in high windup status
ARWLO	BOOL	FB is in low windup status

HWDACA

Description

The Data Acquisition function block provides alarming, scaling, filtering and low cutoff processing. Typically, this function block will be connected to an analog input channel to provide these additional functions if required. The PV output can be connected to regulatory control function blocks such as HWPID as shown below.



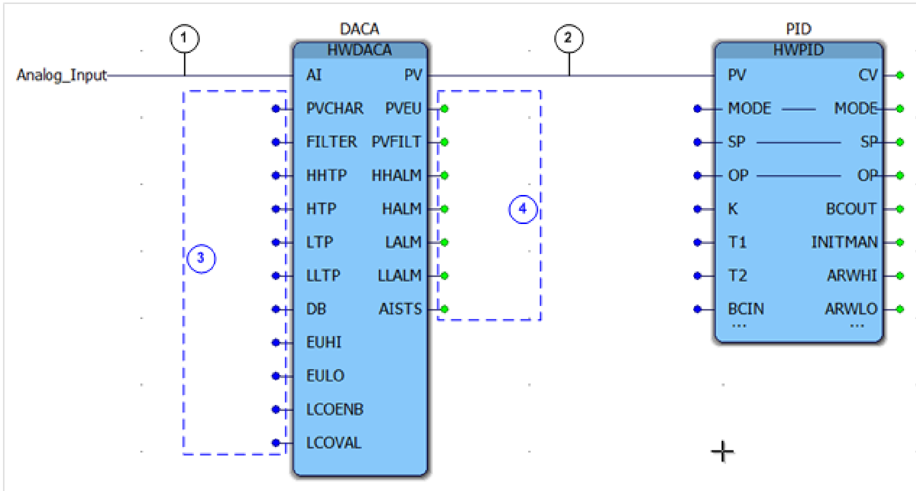
Input

Parameter	Data Type	Description
AI	Analog_Input_ Type	Analog input channel
PVCHAR	INT	PV Characterisation 0 – None (Scaling done by the builder ranges) 1 – Linear (Scaling done by EUHI and EULO on HWDACA FB) 2- Square Root (Square root scaling of input to range defined by EUHI and EULO on HWDACA FB)
FILT	REAL	First order filter time constant in minutes
HHTP	REAL	High High alarm trip point

Parameter	Data Type	Description
HTP	REAL	High alarm trip point
LTP	REAL	Low alarm trip point
LLTP	REAL	Low Low alarm trip point
DB	REAL	Alarm deadband in scaled engineering units
EUHI	REAL	Engineering Units High range. Used when PVCHAR = 1 or 2
EULO	REAL	Engineering Units Low range. Used when PVCHAR = 1 or 2
LCOENB	BOOL	Enable Low Cutoff Value.
LCOVAL	REAL	If LCOENB is true then if processed PV (PVFILT) is less than LCOVAL, PVFILT will be clamped at EULO if PVCHAR = 1 or 2, or the Builder Low range if PVCHAR = 0

Ouput

Parameter	Data Type	Description
PV	Analog_Data_Type	Analog_Data_Type output that is typically connected to a regulatory FB such as HWPID.
PVEU	REAL	Unfiltered PV engineering unit value.
PVFILT	REAL	Filtered PV. This is the PV value passed to downstream Function Blocks via PV. Note that if FILT=0.0, PVFILT = PVEU.
HHALM	BOOL	In High High Alarm
HALM	BOOL	In High Alarm
LALM	BOOL	In Low Alarm
LLALM	BOOL	In Low Low Alarm
AISTS	STRING	Analog Inputs Status message



The following describes the main connections in the figure above.

1. The input AI parameter is connected to an analog input channel of Analog_Input_Data type. This connection carries the analog input channel value, range and status information.
2. The PV output is connected to a regulatory control function block such as HWPID.
3. Configuration parameters are configured as required.
4. Additional outputs can be used for alarm indication and status information.

PVCHAR Characterisation

The characterisation determines how the analog input is scaled. PVCHAR has the following values

0. No scaling of the PV is done. The PV will be scaled per the analog input channel scaling configuration

1. Linear. The PV will be rescaled to the range defined by EUHI and EULO. This feature can be used where a template program is developed with all analog inputs scaled generically to 0-100% and engineering unit scaling is configured on the DACA function block.

2. Square Root Scaling. The PV will be rescaled to range defined by EUHI and EULO with a square root characterisation.

The parameter FILTER is used to set the time constant in minutes of a 1st order filter of the analog input value. The filter equation is

$PV = AI\ (1 - e^{(-FILTER/t)})$

Where

FILTER = Time Constant (minutes)

t = time (minutes)

For a step change in AI, after time

1 x FILTER Minutes : PV = 63.2 % AI

2 x FILTER Minutes : PV = 86.5 % AI

3 x FILTER Minutes : PV = 95.0 % AI

PV Alarming

Analog alarm trip points can be configured as follows

- HHTP – High High trip point in scaled engineering units
- HTP –High trip point in scaled engineering units
- LTP – Low trip point in scaled engineering units
- LLTP – Low Low trip point in scaled engineering units
- DB – Alarm Dead Band in scaled engineering units

These trip points will drive the corresponding alarm flag outputs to true when limits are exceeded.

If an alarm trip point is not required, it can be left unconnected or hidden.

Low Cut-Off

When Low cut off enable (LCOENB) is set to true, the PV will be clamped to 0.0 for analog input values less than low cut off value (LCOVAL). This feature is commonly used for differential pressure inputs for flow calculations to prevent negative values.

PV Values

The output PVEU reflects the unfiltered scaled value of the analog input. PVFILT is filtered value of PVEU. For a FILTER value of 0.0, these will be the same.

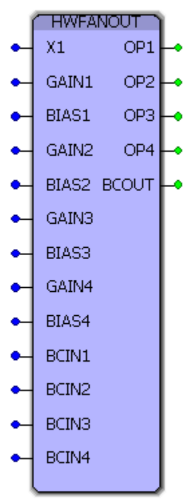
AI Status

The output AISTS will display a string description of the status of the analog input channel.

HWFANOUT

Description

This function shall be used to provide one input and up to four initializable outputs. Shall allow separate gain and bias for each output. Typical use is for split range outputs. An AUTOMAN FB should be used between the output of the FANOUT and final Analog Output.



Input

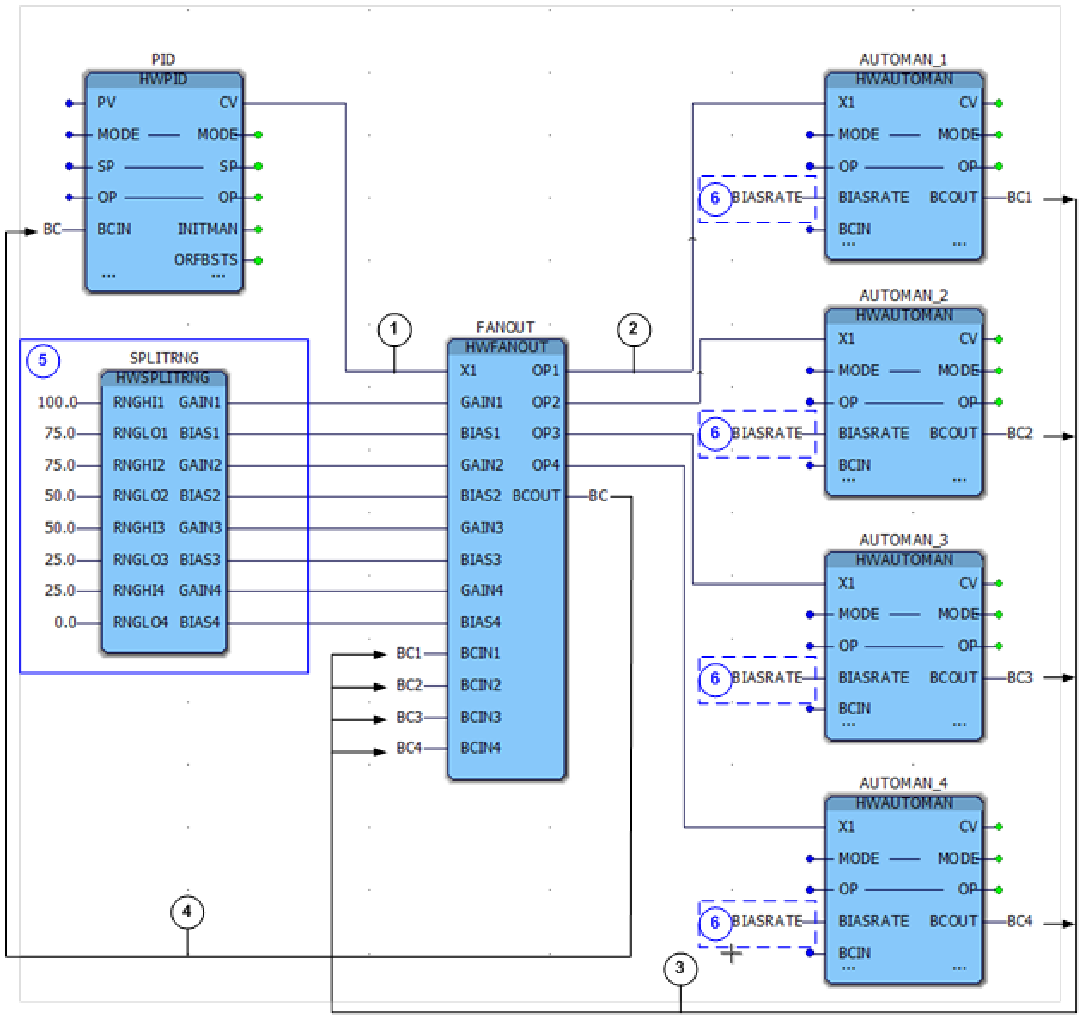
Parameter	Data Type	Description
X1	Analog_Type	Process variable input. AI_Type contains value and quality flags. Generally this is connected to the CV of a regulatory function block such as PID.
GAIN1 to 4	REAL	OUT= GAIN * IN + BIAS
BIAS1 to 4	REAL	OUT= GAIN * IN + BIAS
BCIN1 to 4	BackCalc_Type	Back Calculation Input. This comes from

Parameter	Data Type	Description
		Back Calculation Output of downstream block connected to each output.

Output

Parameter	Data Type	Description
OP11 to 4	Analog_ Type	Output that is normally used to drive the analog output to a control device.
BCOUT	BackCalc_ Type	Back Calculation Output. This goes to Back Calculation Input of upstream block.

The HWFANOUT function block is used to provide up to four separate control variable outputs from a single HWPID control variable input. Each output has separate gain and bias settings. A typical use is for split range valves. To simplify configuration in split range applications, a companion function block HWSPLITRNG, is also provided. An AUTOMAN FB should be used between each output of the HWFANOUT and final Analog Output for individual mode control.



The following describes the main connections in the figure above.

1. The connection to the input X1 will be an Analog_Type from a control variable output of another regulatory control function block HWPID. This connection carries value, range and status information.
2. The pins OP1 to OP4 are used to send output control data to the AUTOMAN blocks. The OP parameter is an Analog_Type as per the X1 input. The AUTOMAN function block provides mode control for each control output. The CV of the AUTOMAN function block is usually connected to an analog output via a CV2AO function block for the final control device. Unused outputs can be left disconnected and hidden.

3. Use the BCOUT/BCIN connection to carry secondary data from each HWAUTOMAN block to the HWFANOUT function block. This will in turn be passed to the HWPID function block to via connection (4) to complete the initialisation path from end to end to carry the following information. (Please refer to section on Initialisation)
   - Anti-Reset Windup Status (ARWHI, ARWLO): Indicates if the HWAUTOMAN is at its high or low limit.
   - Initialization Request Flag (INITMAN): Used to request initialization. If the flag is set by AUTOMAN, the selected PID block initializes itself
   - Initialization Value (INITVAL): Used for initialization when INITMAN true.
4. Use the BCOUT/BCOUT connection to carry secondary data from the HWAUTOMAN blocks to the PID block BCIN pin. Since there are up to 4 sets of secondary data from the HWAUTOMAN function blocks, there are some limitations on initialisation (Please refer to section on Initialisation).
5. The input pins GAIN1..4 and BIAS1..4 determine how the input X1 is scaled for each output according to the equation.
   - OP(i) = GAIN(i) * X1 + BIAS(i) Where i = 1 to 4

     > **NOTE:** The output OP(i) is always range limited to 0-100%.

   - To assist in calculation of GAIN and BIAS for split range control applications, the function block HWSPLITRNG can be used as indicated. This function block is discussed below.
6. The AUTOMAN function block feature, BIASRATE (%/Min) should be used to minimize the effects of bumps when mode is changed from Manual to Auto. (Please refer to section on Initialisation)

Mode Operation

The HWFANOUT function block has no operational parameters that need to be monitored or changed during runtime.

Mode control, manual output, rate of change and output limiting control is set by the AUTOMAN function block on each individual control output. These AUTOMAN function blocks can be interfaced to SCADA if required.

Initialisation and Windup

Since there can be up to 4 individual control outputs, it is not possible to back calculate a single set of secondary data for anti-reset windup, Initialisation request and initialisation value for the upstream PID function block back calculation under all scenarios. The upstream PID will only accept an initialisation request via the FANOUT when all connected AUTOMAN function blocks are requesting initialisation, usually by being put into Manual Mode. Under these conditions, the initialisation value will be computed from the last AUTOMAN function block set to Manual.

This means that when the AUTOMAN function blocks are placed back into Auto mode, there maybe a significant bump in control variable. The mitigate this bump, the BIASRATE setting on the AUTOMAN function block should be configured to provide a floating bias that will ensure the transition is smoothed as a ramp rather than a step change.

Split Range Companion Function Block (HWSPLITRNG)

To assist in simplifying configuration of split range control outputs, a helper function block HWSPITRNG is available to translate split ranges into the appropriate values of GAIN and BIAS. The function block is designed to simply connect directly as shown in the example. Each range defines the range of the common output from the PID that will be translated to full range (0-100%) for the corresponding output from the FANOUT. In the example given

- X1 in range 0-25% results in OP4 range 0-100%
- X1 in range 25-50% results in OP3 range 0-100%
- X1 in range 50-75% results in OP2 range 0-100%
- X1 in range 75-100% results in OP4 range 0-100%

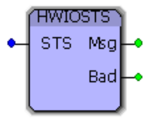
So, if X1 = 80%, OP4, OP3 and OP2 will be 100% and OP1 will be 20%

The ranges for each output can be overlapped to help with reduction of any dead band in transitioning between output control elements such as control valves.

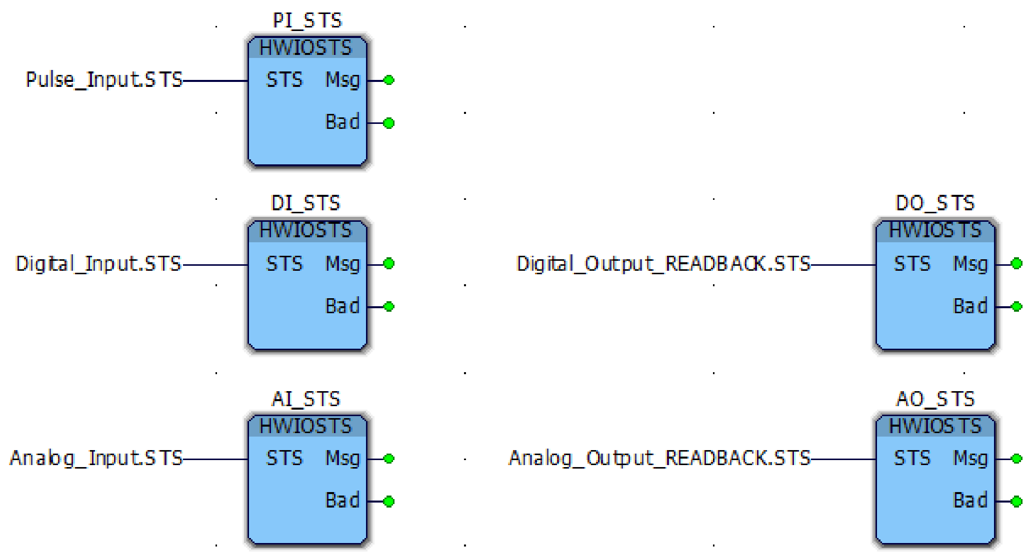
HWIOSTS

Description

This function block is used to decode I/O channel status value (STS member of I/O datatypes) and provide an I/O channel Bad flag for alarming and logic and a status message.



Below is an example for each I/O channel data type. Note that the channel status information is carried by the READBACK data for digital and analog outputs.



Input

Parameter	Data Type	Description
STS	USINT	This is connected to the STS member of any I/O data type.

Output

Parameter	Data Type	Description
Msg	STRING	Status Message STS BAD MSG 0 False Good 1 True Offline 11 True ORHIEX 12 False ORHI 13 False URLO 14 True URLOEX 15 False No Cal 16 True Open Wire 17 True Chn Bad 18 True Short Cct 19 True IO Hdw Err 20 True RB Test Fail
Bad	BOOL	Flag indicating I/O is Bad (See MSG above). This can be used in downstream logic to take appropriate action.

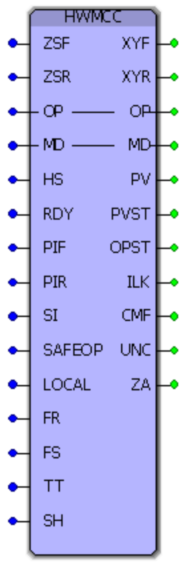
HWMCC

The HWFBLib contains a group of related device control function blocks for digital control of valves and motors as shown below.

- HWSDV – Control of solenoid operated values such as shutdown valves
- HWMOV – Control of motor operated valves
- HWMCC – Control of motors
- HWMLV – Control of main line valves

Description

The HWMCC function block is applicable to motor control. This function block can command a motor to run for single output or to command a motor in forward and reverse direction for dual output. The outputs of function block are latched. If a pulsed output to Motor controller is required, a rising edge trigger can be used before digital output.



Input

Parameter	Data Type	Description
ZSF	BOOL	Forward (or Run) indication from field. False – Not running True - Forward (Running)
ZSR	BOOL	Reverse indication from field. Not connected for simple Stop/Run applications. False – Not running True – Reverse indication
OP	INT	Accepts command from SCADA when MD is in Manual. When MD is Auto, OP tracks HS. OP Command States 0 – Stop

Parameter	Data Type	Description
		1 – Forward (Run) 2 – Reverse
MD	BOOL	Mode control. False – Manual – OP Can be commanded from SCADA OP True – Auto – Commands come from HS input. OP tracks HS.
HS	INT	Hand Switch command from logic to control motor. 0 – Stop 1 – Forward (Run) 2 – Reverse
RDY	BOOL	Motor controller Ready input to indicate control is available from MCC. Normally wired to DI from MCC.
PIF	BOOL	Forward (run) permissive. Must be true to permit forward (run) command. SI will override.
PIR	BOOL	Reverse permissive. Must be true to permit reverse command. SI will override.
SI	BOOL	Safety override interlock enforced if True
SAFEOP	INT	Safety override interlock command. 0 – Stop 1 – Forward (Run) 2 – Reverse
LOCAL	BOOL	Local = True. When in local OP commands will track the valve state. OP commands will not be accepted from SCADA or HS regardless of MD. Normally LOCAL is a digital input from MCC.
FR	BOOL	Forward/Reverse transition allowed. False – OP command must go to stop before direction can be changed True – OP can change direction

Parameter	Data Type	Description
FS	BOOL	If FR is True, FS controls whether a forced stop is performed as part of direction change False – Immediate reversal of direction True – Stop command issued between forward/reverse direction change
TT	TIME	Maximum operation time to control motor. This is used for command fail alarm.
SH	TIME	Stop hold time when using FR and FS options. Controls how long stop command is held between direction change commands.

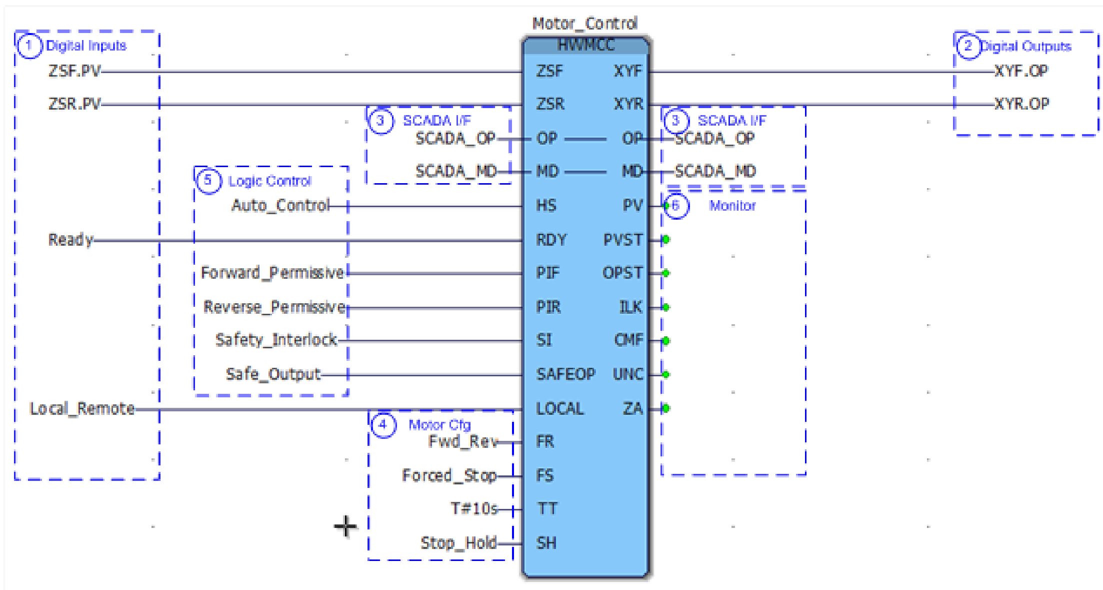
Output

Parameter	Data Type	Description
XYF	BOOL	Output command to Forward/Run DO. This output is latched
XYR	BOOL	Output command to Reverse DO. This output is latched
PVST	STRING	Description of valve state used for monitoring in IEC Programming Workspace debug mode. Note that if PV = CfgErr then the settings of INBET, CLOSE, OPEN, BAD are inconsistent, that is values are outside of range 0 to 3 and/or there are duplicate values.
PV	INT	Valve state as an integer 0 – Stop 1 – Run/Fwd 2 – Rev
OPST	STRING	Description of valve output command used for monitoring in IEC Programming Workspace debug mode.
ILK	BOOL	Interlock Override active
CMF	BOOL	Motor failed to match commanded state within TT time. This alarm is inhibited when in LOCAL

Parameter	Data Type	Description
UNC	BOOL	Uncommanded change of state alarm if motor state changes from commanded state. This alarm is inhibited when in LOCAL
ZA	BOOL	Common alarm.

Implementation Example

A typical example is shown below with the main configuration areas highlighted.



1. For a simple run/stop motor control, ZSF is connected to the motor run indicator. For Fwd/Rev control, the forward and reverse indications are connected to ZSF and ZSR respectively. Optional inputs for Local can be used where a Hand/Off/Auto switch is used and Motor Controller Ready input.
2. The function block main control outputs XYF (Run/Forward command) and optionally XYR (Reverse Command) are connected to digital outputs to drive the motor. These outputs will be latched. A rising edger trigger can be inserted before digital outputs if a pulsed output is required.

3. The SCADA control interface for the function block is mapped to a status point where
   a. ZIC and ZIO are used for PV indication of valve state
   b. OP is used for SCADA control of valve when MD (Mode) is manual. The OP states are
      - Stop (0)
      - Fwd/Run (1)
      - Rev (2).
   c. MD is used to control mode of function block. When MD is Manual (False), the OP is used to control valve operation. When MD is Auto (True), controller logic operates valve via the HS input and OP tracks HS.
4. This group of parameters determines how motor control is configured.
   a. Forward Reverse Allowed FR is set to true if switching OP between forward and reverse is allowed without first stopping the motor. If set to False, the motor control OP must be commanded to Stop before a direction change can be commanded.
   b. Forced Stop FS. (Only applicable if FR=True). If set to true, a stop will be performed before direction is reversed.
   c. Travel Time TT specifies time out period before command fail alarm is generated.
   d. Stop Hold ST. If FR and FS are true, this specifies how long a stop is held for before direction is changed.

5. These inputs are primarily driven by program logic to control valve.
   a. HS controls valve operation when MD is Auto.
   b. PIF and PIR are permissives which need to be True before a motor can be commanded to Run/Fwd or Rev. If these pins are not connected, permissives will be true by default.
   c. SI and SAFEOP are used for safety interlock operation. If SI is true, the motor will be commanded to the SAFEOP state of either:
      - Stop (0)
      - Fwd/Run (1)
      - Rev (2).
6. These pins can be used for monitoring operation of function block.
   a. PV is a numeric indicating state of valve where
      - PV = 0 (Stop)
      - PV = 1 (RUN/Fwd)
      - PV = 2 (Rev)
   b. PVST and OPST display descriptive state of motor state and command
   c. Alarm indications
      - ILK Interlock active
      - CMF Commanded state ofmoto not met within thimeout period
      - UNC Uncommanded alarm is active if valve state becomes different to commanded state
      - ZA Common alarm

Mode Operation

The function block has modes Manual (MD=False) and Auto (MD=True).

In Manual Mode, the outputs XYF/R tracks OP set from SCADA such that

OP = 0 (Stop), Then XYF = False, XYR = False

OP = 1 (Run/Fwd), Then XYF = True, XYR = False

OP = 2 (Rev), Then XYF = False, XYR =True

In Auto Mode, the output XYF/R tracks HS which is driven by program logic. When in Auto, OP will track HS so that Mode change from Auto to Manual is bumpless.

Local/Remote

This input is normally connected to Hand/Off/Auto switch. When the LOCAL input is true, OP will track the PV state of valve and control of valve will be via a local control panel. While in LOCAL, commands will not be accepted from SCADA OP or Logic controlled HS regardless of MD setting.

Ready Indication

A ready input from motor controller or logic is available to indicate if motor control is allowed.

Permissive and Safety Interlock.

A permissive is available for Run/Fwd commands. If the respective permissive is not true, then that command cannot be executed. If the permissive becomes false after command is issued, the command is unaffected.

A safety interlock input (SI) of True will command the motor to the state set by SAFEOP which is defined as per OP. The safety interlock will take the highest precedence in Auto or Manual and will override a permissive.

Forward/Reverse Configuration

The MCC function has options to determine if change in direction is allowed.

If FR = False, then the MCC must be commanded to stop before a change in direction can be commanded.

If FR = True, then the MCC can be commanded to change direction from Fwd to Rev or Rev to Fwd without stopping first. If this configuration is selected, a further setting, Forced Stop FS determines if MCC function block performs the Stop command before changing direction and Stop Time ST determines how long Stop state should be held before changing direction.

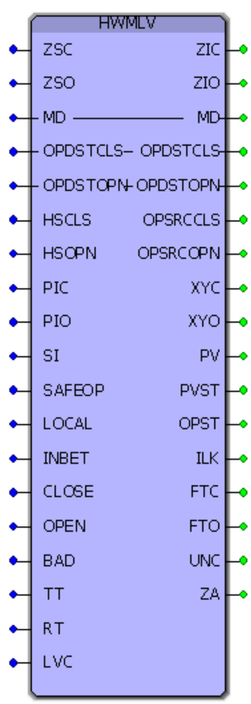
HWMLV

The HWFBLib contains a group of related device control function blocks for digital control of valves and motors as shown below.

- HWSDV – Control of solenoid operated values such as shutdown valves
- HWMOV – Control of motor operated valves
- HWMCC – Control of motors
- HWMLV – Control of main line valves

Description

The HWMLV function block is like the motor operated valves except that it is driven by a pulsed command from SCADA/Logic rather than a latched command. These devices are characterised by dual outputs to drive a valve open or closed. The output will be energised until the valve meets its command position. And optional extra seating time can be used to ensure valve is firmly seated after it meets the commanded position. These valves will fail in last commanded position.



Input

Parameter	Data Type	Description
ZSC	BOOL	Close Limit Switch Input from DI
ZSO	BOOL	Open Limit Switch Input from DI
MD	BOOL	Mode control. False – Manual – OP Can be commanded from SCADA OP True – Auto – Commands come from HS input. OP tracks HS.
OPDSTCLS	BOOL	Accepts pulsed command from SCADA to close valve when MD is in Manual. This will typically be used for the SCADA close command destination address.
OPDSTOPN	BOOL	Accepts pulsed command from SCADA to open valve when MD is in Manual. This will typically be used for the SCADA open command destination address.
HSCLS	BOOL	Hand Switch pulsed command from logic to close valve when MD is Auto
HSOPN	BOOL	Hand Switch pulsed command from logic to open valve when MD is Auto
PIC	BOOL	Close permissive. Must be true to permit close command. SI will override.
PIO	BOOL	Open permissive. Must be true to permit close command. SI will override.
SI	BOOL	Safety override interlock enforced if True
SAFEOP	BOOL	Safety override interlock command. False – Close True - Open
LOCAL	BOOL	Local = True. When in local OP commands will track the valve state. OP commands will not be accepted from SCADA or HS regardless of MD. Normally LOCAL is a digital input from device.
INBET	INT	Range 0 – 3, Value of in between or travel state. This value is calculated by evaluating ZSC + 2 x ZSO when valve in this state

Parameter	Data Type	Description
CLOSE	INT	Range 0 – 3, Value of Close state. This value is calculated by evaluating ZSC + 2 x ZSO when valve in this state
OPEN	INT	Range 0 – 3, Value of Open state. This value is calculated by evaluating ZSC + 2 x ZSO when valve in this state
BAD	INT	Range 0 – 3, Value of bad or inconsitent state. This value is calculated by evaluating ZSC + 2 x ZSO when valve in this state. This will usually be the remaining state after INBET, OPEN and CLOSE states have been determined.
TT	TIME	Maximum travel time to open or close valve. This is used for fail to open and fail to close alarms.
RT	TIME	Run on time once valve has reached commanded state. The run on time can be used to seal or seat valve.
LVC	BOOL	When set to TRUE, OPSRCCLS and OPSRCOPN will indicate last valid or successful command. When set to False, OPSRCCLS and OPSRCOPN will indicate last command issued.

Output

Parameter	Data Type	Description
ZIC, ZIO	BOOL	Normalised Close and Open limits with the following truth table ZIC ZIO State F F Inbet or Travel T F Closed F T Open T T Bad **NOTE:** SCADA can use these for PV for consistency across all valves or can address ZSC and ZSO based on implementers preferences.
OPSRCCLS	BOOL	Latched command feedback for Close operation. This is

Parameter	Data Type	Description
		typically used for the SCADA close command source address. This output behaviour is determined by setting of LVC (Last Valid Command).
OPSRCOPN	BOOL	Latched command feedback for Open operation. This is typically used for the SCADA open command source address. This output behaviour is determined by setting of LVC (Last Valid Command).
XYC	BOOL	Output command to Close DO. This output is energised for duration of close command until valve reaches close state plus run on time RT or travel time TT expires.
XYO	BOOL	Output command to Open DO. This output is energised for duration of open command until valve reaches open state plus run on time RT or travel time TT expires.
PV	INT	Valve state as an integer 0 – Travel 1 – Closed 2 – Open 3 - Bad
PVST	STRING	Description of valve state used for monitoring in IEC Programming Workspace debug mode. Note that if PV = CfgErr then the settings of INBET, CLOSE, OPEN, BAD are inconsistent, that is values are outside of range 0 to 3 and/or there are duplicate values.
OPST	STRING	Description of valve output command used for monitoring in IEC Programming Workspace debug mode.
ILK	BOOL	Interlock Override active
FTC	BOOL	Fail to close alarm raised if valve fails to close within TT. This alarm is inhibited when in LOCAL
FTO	BOOL	Fail to open alarm raised if valve fails to open within TT. This alarm is inhibited when in LOCAL
UNC	BOOL	Uncommanded change of state alarm if valve moves from commended state. This alarm is inhibited when in LOCAL

Parameter	Data Type	Description
ZA	BOOL	Common alarm.

> **NOTE:** Block can be used for single limit switch valve state indication by using only single limit connected to either ZSC or ZSO and configuring INBET, CLOSE, OPEN and BAD. For example, single limit switch indication

ZS – False Valve Open

ZS – True Valve Closed

Connect ZS to ZSC, leave ZSO unconnected. States will be

INBET 2 (don’t care)

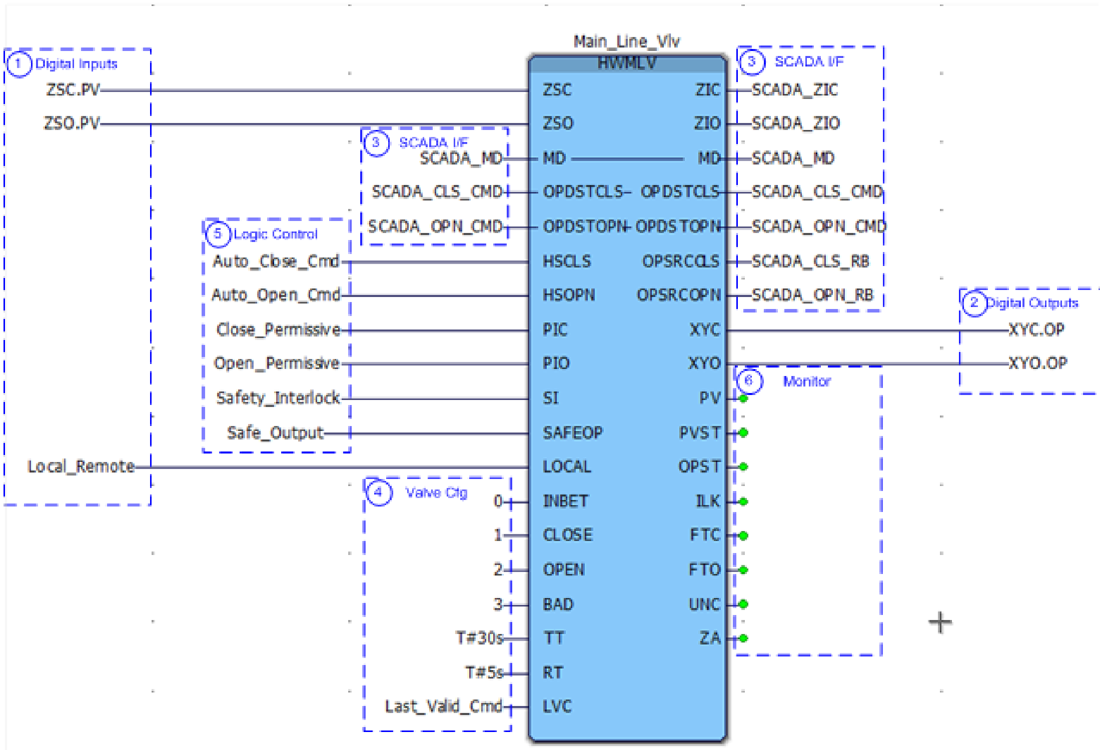
CLOSE 1

OPEN 0

BAD 3 (don’t care)

Implementation Example

A typical example is shown below with the main configuration areas highlighted.



1. Digital Inputs are connected to valve position feedback limit switches. Typically, there will be an open (ZSO) and a close (ZSC) limit switch. In some cases, only a single limit switch is provided. An optional input for Local can be used where a Hand/Off/Auto switch is used.

2. The function block main control outputs XYC (close command) and XYO (Open Command) are connected to digital outputs to drive the valve operating motor. These outputs will be energized for duration of time it takes valve to meet commanded state plus an option extra seating time.

3. The SCADA control interface for the function block is mapped to a status point where

   a. ZIC and ZIO are used for PV indication of valve state

   b. OP is used for SCADA control of valve when MD (Mode) is manual. The OP is configured for pulsed operation from SCADA such that OPDSTCLS/OPN is used for SCADA OP destination address and OPSRCCLS/OPN is used for SCADA OP source address to read back the command that was issued.

c. MD is used to control mode of function block. When MD is Manual (False), the OP is used to control valve operation. When MD is Auto (True), controller logic operates valve via the HSCLS/OPN inputs and OPSRCCLS/DST tracks the logic controlled commands.

4. This group of parameters determines how valve is configured.

   a. Limit mapping (INBET, CLOSE, OPEN, BAD) determine how limit switches (ZSC, ZSO) map to SCADA indication (ZIC, ZIO). Due to the different possible configurations of limit switch operation (Normally Open or Normally Closed) and orientations of operating cams (make/break at beginning or end of valve movement), this mapping allows a single point to configure limit switch behavior without affecting downstream SCADA and logic functions should a there be difference individual valves. Please refer to limit mapping table section.

   b. Travel Time TT specifies time out period before a travel time alarm is generated and commanded output is declared failed and output is de-energised.

   c. Run-On Time RT specifies the additional time output is energized after the valve meets its commanded position. This is used to seat the valve firmly in position.

   d. Last Valid Command LVC determines how command feedback is indicated in SCADA via OPSRCCLS/OPN feedback. If LVC is True, OPSRCCLS/OPN will indicate the last successful command. If LVC is False, OPSRCCLS/OPN will indicate last issued command regardless of whether it was successful or not.

5. These inputs are primarily driven by program logic to control valve.

   a. HSCLS and HSOPN are pulsed inputs derived from logic to control valve operation when MD is Auto.

   b. PIC and PIO are permissives which need to be True before a valve can be closed or opened. If these pins are not connected, permissives will be true by default.

   c. SI and SAFEOP are used for safety interlock operation. If SI is true, the valve will be commanded to the SAFEOP state of Close (False) or Open (True).

6. These pins can be used for monitoring operation of function block.
   a. PV is a numeric indicating state of valve where
      - PV = 0 (Travel)
      - PV = 1 (Close)
      - PV = 2 (Open)
      - PV = 3 (Bad)
   b. PVST and OPST display descriptive state of valve position and command
   c. Alarm indications
      - ILK Interlock active
      - FTC, FTO Fail to Close, Open if commanded state not met within travel timeout period.
      - UNC Uncommanded alarm is active if valve state becomes different to commanded state
      - ZA Common alarm

Limit Mapping

The limit mapping inputs INBET, CLOSE, OPEN and BAD provide a means to standardize valve indication ZIC, ZIO in SCADA and for all downstream logic operations. The standardized indication is based on positive logic as shown below.

ZIC	ZIO	Valve State
FALSE	FALSE	Travel (In-between)
TRUE	FALSE	Closed
FALSE	TRUE	Opened
TRUE	TRUE	Bad (Error)

The operation of the actual valve limit indications may vary from the above due to actual configuration of limits on valves. In many cases, these differences are discovered during commissioning. The limit table provides a single place to rationalize limits to above table so that any impacts to downstream configuration of SCADA and logic are not impacted during commissioning.

The mapping value for a state is calculated by the formula

ZSC + 2 x ZSO

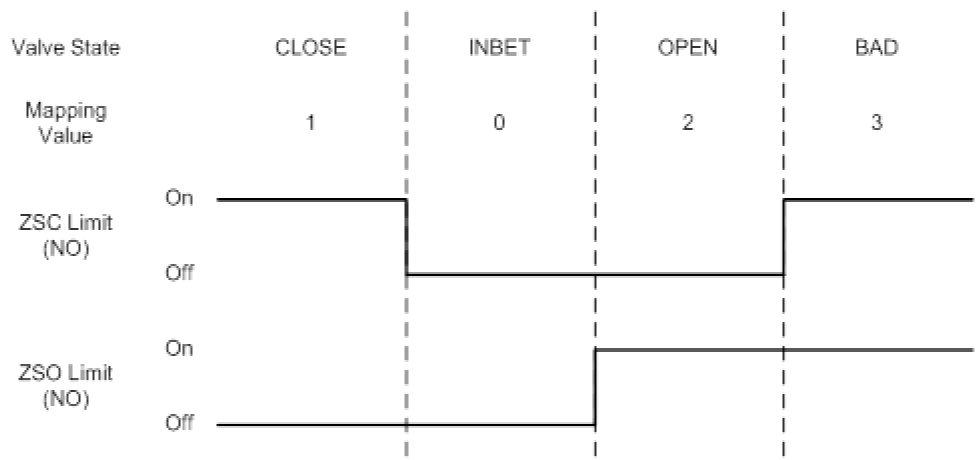
For example, if the valve is physically in the OPEN state and the open limit ZSO is On and the close limit ZSC is On then

OPEN = 1 + 2 x 1 = 3

Following are some examples typically encountered.

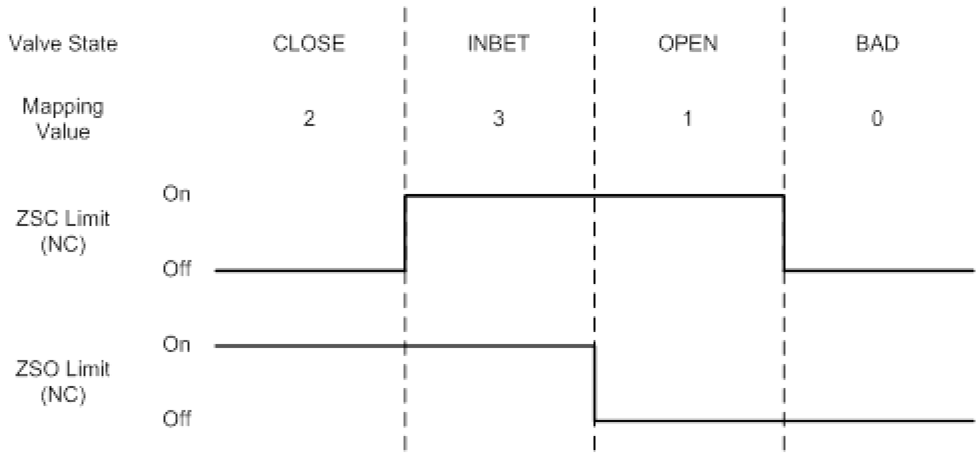
Valve Limit Switches Configured in Normally Open state.

This follows the ZIC, ZIO positive logic mapping. This is the default limit mapping of function block.



Valve Limit Switches Configured in Normally Closed state.

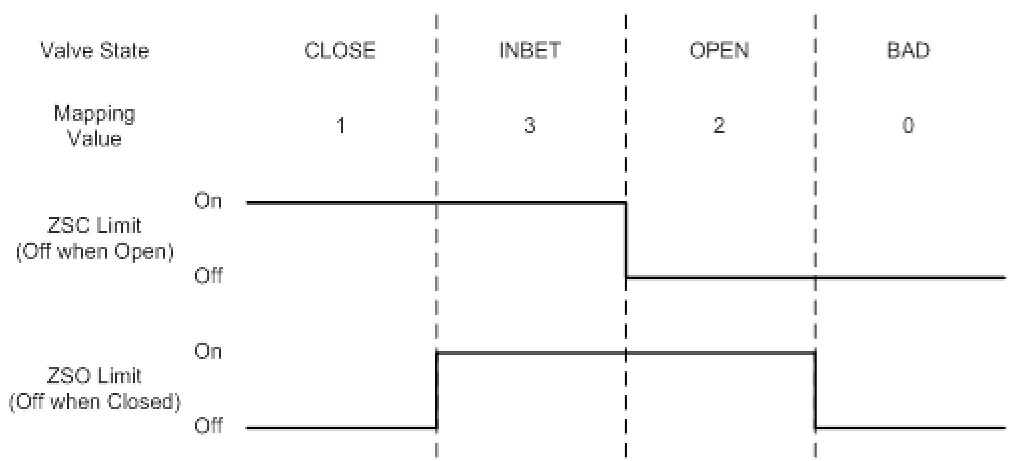
This arrangement is the reverse configuration (negative logic) and is sometimes used as it provides an error indication if field cables are cut (Both limits are off). The mapping values shown will convert ZIC and ZIO to follow positive logic.



Valve Limit Switches Configured in Complimentary Arrangement.

This arrangement has cams driving limit switches at end of movement so the ZSC limit is active unless valve fully opened and ZSO limit is active unless valve is fully closed. The advantage of this arrangement is that ZSO and ZSC use positive logic for Open and Close state but the BAD state is detected if ZSO and ZSC are both off (open circuit). This may be due to a failure of limit switches, links or fuses removed, field power lost or field cables damaged.

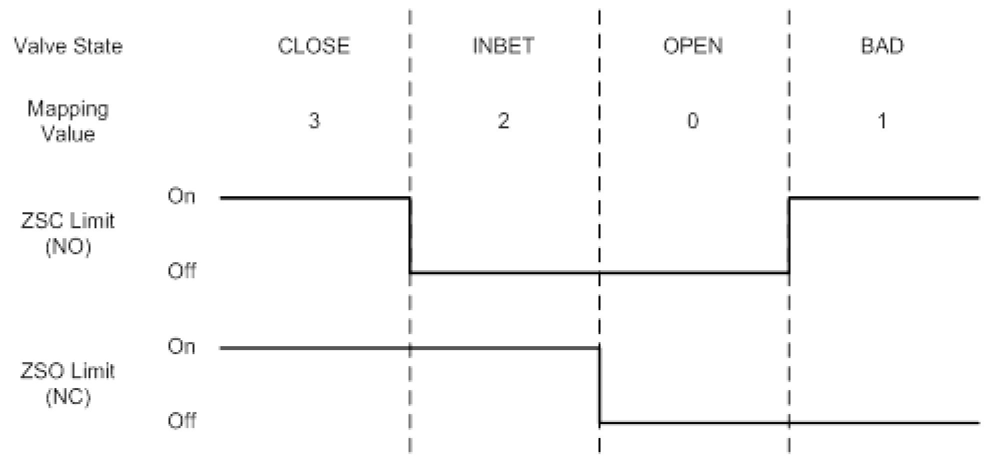
Below are the mapping values to be used to translate to the standard ZIC, ZIO



Valve Limit Switches Configured in Mixed State.

In this example, ZSC is configured as normally opened and ZSO is configured as normally closed. This is mainly an example to indicate flexibility to handle different arrangements that might arise.

Below are the mapping values to be used to translate to the standard ZIC, ZIO states.



Single Limit Switch

The function block can be used for single limit switch valve state indication by using only single limit connected to either ZSC or ZSO and configuring INBET, CLOSE, OPEN and BAD. For example, if there is only a single limit switch indication where :

ZS – False Valve Open

ZS – True Valve Closed

Connect ZS to ZSC, leave ZSO unconnected. The states will be:

INBET 2 (don’t care)

CLOSE 1

OPEN 0

BAD 3 (don’t care)

Mode Operation

The function block has modes Manual (MD=False) and Auto (MD=True).

In Manual Mode, the output XYC/XYO tracks OP set from SCADA.

In Auto Mode, the output XYC/XYO tracks HSCLS/HSOPN which is driven by program logic. When in Auto, OP will track HSCLS/HSOPN so that Mode change from Auto to Manual is bumpless.

Local/Remote

This input is normally connected to Hand/Off/Auto switch. When the LOCAL input is true, OP will track the PV state of valve and control of valve will be via a local control panel. While in LOCAL, commands will not be accepted from SCADA OP or Logic controlled HS regardless of MD setting.

Permissive and Safety Interlock.

A permissive is available for Open and Close commands. If the respective permissive is not true, then that command cannot be executed. If the permissive becomes false after command is issued, the command is unaffected.

A safety interlock input (SI) of True will command the valve to the state set by SAFEOP (False = Close, Open = True). The safety interlock will take the highest precedence in Auto or Manual and will override a permissive.

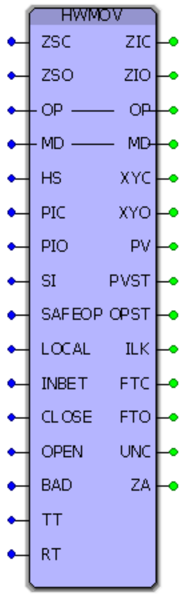
HWMOV

The HWFBLib contains a group of related device control function blocks for digital control of valves and motors as shown below.

- HWSDV – Control of solenoid operated values such as shutdown valves
- HWMOV – Control of motor operated valves
- HWMCC – Control of motors
- HWMLV – Control of main line valves

Description

The HWMOV function block is applicable to motor operated valves. These devices are characterize by dual outputs to drive a valve open or closed via motor operation. The output will be energized until the valve meets its command position. And optional extra seating time can be used to ensure valve is firmly seated after it meets the commanded position. These valves will fail in last commanded position.



Input

Parameter	Data Type	Description
ZSC	BOOL	Close Limit Switch Input from DI
ZSO	BOOL	Open Limit Switch Input from DI
OP	BOOL	Accepts command from SCADA when MD is in Manual. When MD is Auto, OP tracks HS. OP Command States False – Close True – Open
MD	BOOL	Mode control. False – Manual – OP Can be commanded from SCADA OP True – Auto – Commands come from HS input. OP tracks HS.
HS	BOOL	Hand Switch command from logic to open or close valve. False – Close command True – Open command

Parameter	Data Type	Description
PIC	BOOL	Close permissive. Must be true to permit close command. SI will override.
PIO	BOOL	Open permissive. Must be true to permit close command. SI will override.
SI	BOOL	Safety override interlock enforced if True
SAFEOP	BOOL	Safety override interlock command. False – Close True - Open
LOCAL	BOOL	Local = True. When in local OP commands will track the valve state. OP commands will not be accepted from SCADA or HS regardless of MD. Normally LOCAL is a digital input from device.
INBET	INT	Range 0 – 3, Value of in between or travel state. This value is calculated by evaluating ZSC + 2 x ZSO when valve in this state
CLOSE	INT	Range 0 – 3, Value of Close state. This value is calculated by evaluating ZSC + 2 x ZSO when valve in this state
OPEN	INT	Range 0 – 3, Value of Open state. This value is calculated by evaluating ZSC + 2 x ZSO when valve in this state
BAD	INT	Range 0 – 3, Value of bad or inconsitent state. This value is calculated by evaluating ZSC + 2 x ZSO when valve in this state. This will usually be the remaining state after INBET, OPEN and CLOSE states have been determined.
TT	TIME	Maximum travel time to open or close valve. This is used for fail to open and fail to close alarms.
RT	TIME	Run on time once valve has reached commanded state. The run on time can be used to seal or seat valve.

Output

Parameter	Data Type	Description
ZIC, ZIO	BOOL	Normalised Close and Open limits with the following truth table

Parameter	Data Type	Description
		ZIC ZIO State F F Inbet or Travel T F Closed F T Open T T Bad Note SCADA can use these for PV for consistency across all valves or can address ZSC and ZSO based on implementers preferences.
XYC	BOOL	Output command to Close DO. This output is energised for duration of close command until valve reaches close state plus run on time RT or travel time TT expires.
XYO	BOOL	Output command to Open DO. This output is energised for duration of open command until valve reaches open state plus run on time RT or travel time TT expires.
PV	INT	Valve state as an integer 0 – Travel 1 – Closed 2 – Open 3 - Bad
PVST	STRING	Description of valve state used for monitoring in IEC Programming Workspace debug mode. Note that if PV = CfgErr then the settings of INBET, CLOSE, OPEN, BAD are inconsistent, that is values are outside of range 0 to 3 and/or there are duplicate values.
OPST	STRING	Description of valve output command used for monitoring in IEC Programming Workspace debug mode.
ILK	BOOL	Interlock Override active
FTC	BOOL	Fail to close alarm raised if valve fails to close within TT. This alarm is inhibited when in LOCAL
FTO	BOOL	Fail to open alarm raised if valve fails to open within TT. This alarm is inhibited when in LOCAL
UNC	BOOL	Uncommanded change of state alarm if valve moves from

Parameter	Data Type	Description
		commended state. This alarm is inhibited when in LOCAL
ZA	BOOL	Common alarm.

> **NOTE:** Block can be used for single limit switch valve state indication by using only single limit connected to either ZSC or ZSO and configuring INBET, CLOSE, OPEN and BAD. For example, single limit switch indication

ZS – False Valve Open

ZS – True Valve Closed

Connect ZS to ZSC, leave ZSO unconnected. States will be

INBET 2 (don’t care)

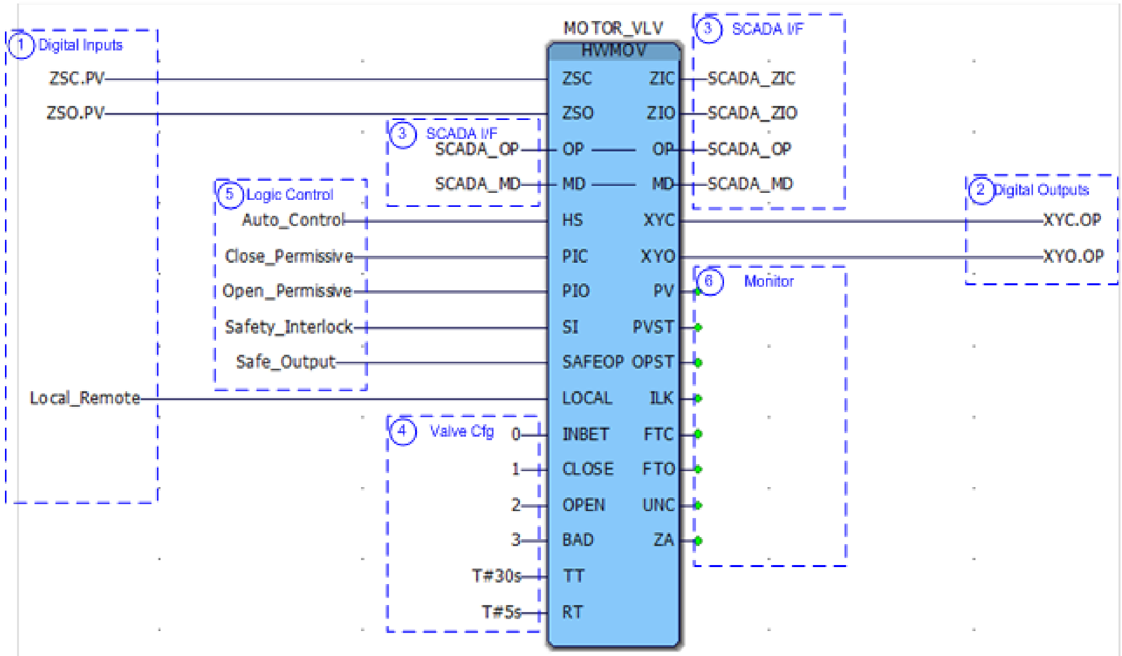
CLOSE 1

OPEN 0

BAD 3 (don’t care)

Implementation Example

A typical example is shown below with the main configuration areas highlighted.



1. Digital Inputs are connected to valve position feedback limit switches. Typically, there will be an open (ZSO) and a close (ZSC) limit switch. In some cases, only a single limit switch is provided. An optional input for Local can be used where a Hand/Off/Auto switch is used.

2. The function block main control outputs XYC (close command) and XYO (Open Command) are connected to digital outputs to drive the valve operating motor. These outputs will be energized for duration of time it takes valve to meet commanded state plus an option extra seating time.

3. The SCADA control interface for the function block is mapped to a status point where

   a. ZIC and ZIO are used for PV indication of valve state

   b. OP is used for SCADA control of valve when MD (Mode) is manual. The OP states are Close (False) and Open (True).

   c. MD is used to control mode of function block. When MD is Manual (False), the OP is used to control valve operation. When MD is Auto (True), controller logic operates valve via the HS input and OP tracks HS.

4. This group of parameters determines how valve is configured.
   a. Limit mapping (INBET, CLOSE, OPEN, BAD) determine how limit switches (ZSC, ZSO) map to SCADA indication (ZIC, ZIO). Due to the different possible configurations of limit switch operation (Normally Open or Normally Closed) and orientations of operating cams (make/break at beginning or end of valve movement), this mapping allows a single point to configure limit switch behavior without affecting downstream SCADA and logic functions should a there be difference individual valves. Please refer to limit mapping table section.
   b. Travel Time TT specifies time out period before a travel time alarm is generated and commanded output is declared failed and output is de-energised.
   c. Run-On Time RT specifies the additional time output is energized after the valve meets its commanded position. This is used to seat the valve firmly in position.
5. These inputs are primarily driven by program logic to control valve.
   a. HS controls valve operation when MD is Auto
   b. PIC and PIO are permissives which need to be True before a valve can be closed or opened. If these pins are not connected, permissives will be true by default.
   c. SI and SAFEOP are used for safety interlock operation. If SI is true, the valve will be commanded to the SAFEOP state of Close (False) or Open (True).
6. These pins can be used for monitoring operation of function block.
   a. PV is a numeric indicating state of valve where
      - PV = 0 (Travel)
      - PV = 1 (Close)
      - PV = 2 (Open)
      - PV = 3 (Bad)
   b. PVST and OPST display descriptive state of valve position and command

c. Alarm indications

- ILK Interlock active
- FTC, FTO Fail to Close, Open if commanded state not met within travel timeout period.
- UNC Uncommanded alarm is active if valve state becomes different to commanded state
- ZA Common alarm

Limit Mapping

The limit mapping inputs INBET, CLOSE, OPEN and BAD provide a means to standardize valve indication ZIC, ZIO in SCADA and for all downstream logic operations. The standardized indication is based on positive logic as shown below.

ZIC	ZIO	Valve State
FALSE	FALSE	Travel (In-between)
TRUE	FALSE	Closed
FALSE	TRUE	Opened
TRUE	TRUE	Bad (Error)

The operation of the actual valve limit indications may vary from the above due to actual configuration of limits on valves. In many cases, these differences are discovered during commissioning. The limit table provides a single place to rationalize limits to above table so that any impacts to downstream configuration of SCADA and logic are not impacted during commissioning.

The mapping value for a state is calculated by the formula

ZSC + 2 x ZSO

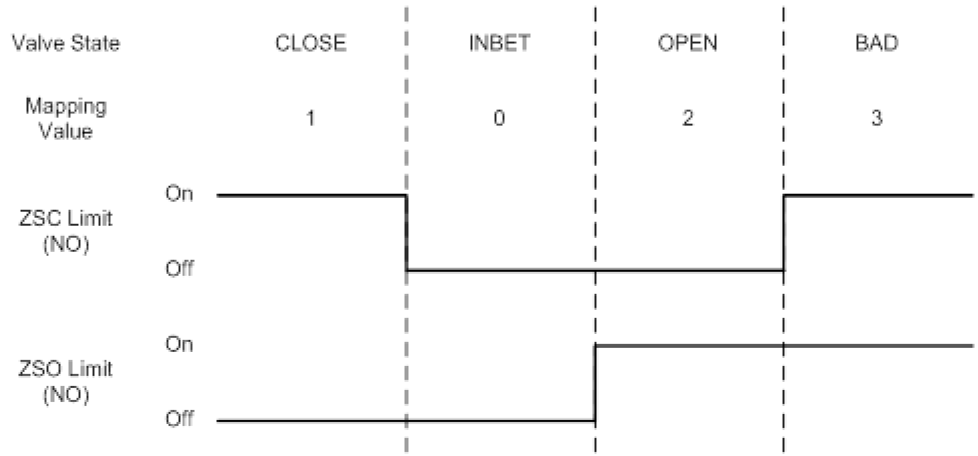
For example, if the valve is physically in the OPEN state and the open limit ZSO is On and the close limit ZSC is On then

OPEN = 1 + 2 x 1 = 3

Following are some examples typically encountered.

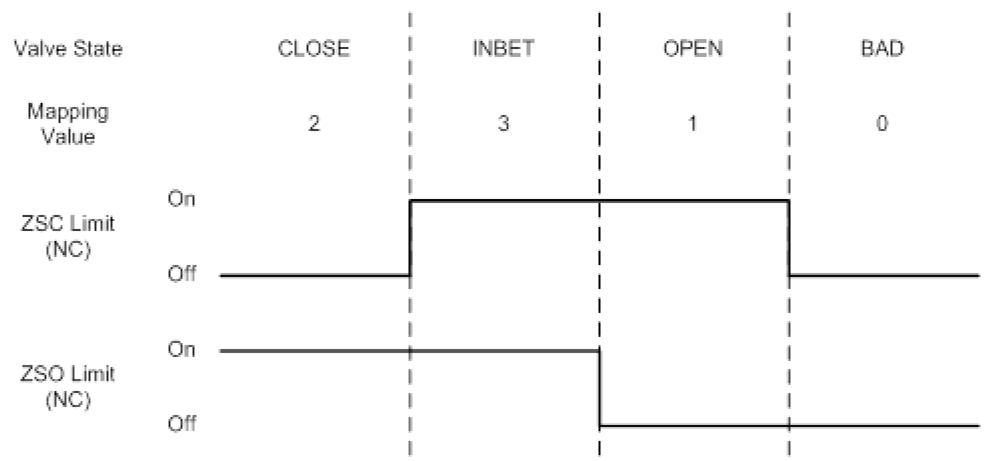
Valve Limit Switches Configured in Normally Open state.

This follows the ZIC, ZIO positive logic mapping. This is the default limit mapping of function block.



Valve Limit Switches Configured in Normally Closed state.

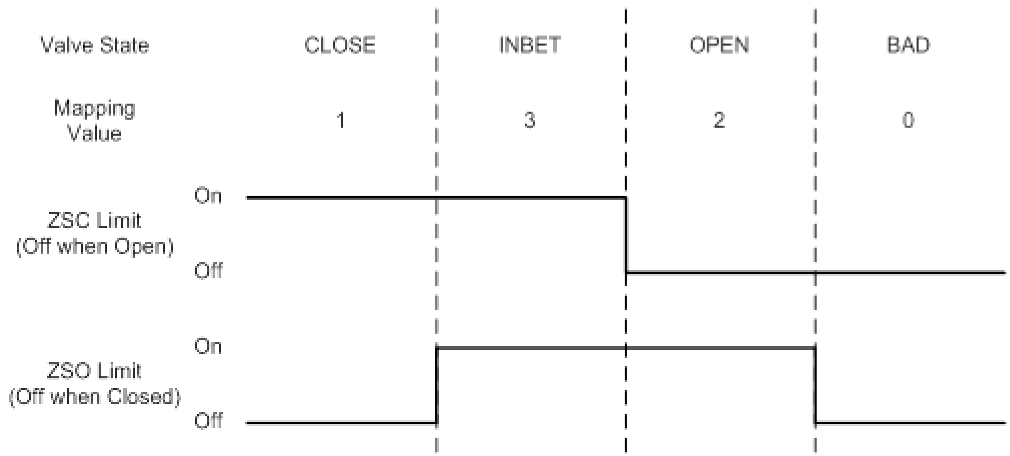
This arrangement is the reverse configuration (negative logic) and is sometimes used as it provides an error indication if field cables are cut (Both limits are off). The mapping values shown will convert ZIC and ZIO to follow positive logic.



Valve Limit Switches Configured in Complimentary Arrangement.

This arrangement has cams driving limit switches at end of movement so the ZSC limit is active unless valve fully opened and ZSO limit is active unless valve is fully closed. The advantage of this arrangement is that ZSO and ZSC use positive logic for Open and Close state but the BAD state is detected if ZSO and ZSC are both off (open circuit). This may be due to a failure of limit switches, links or fuses removed, field power lost or field cables damaged.

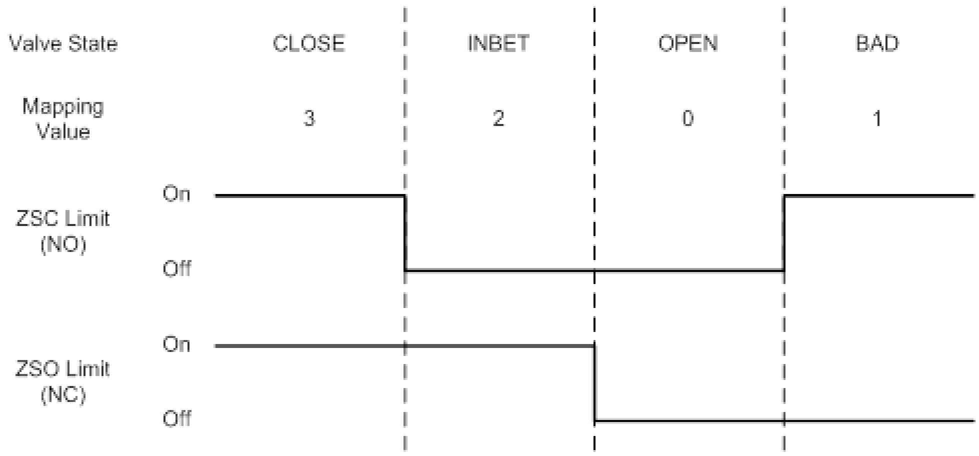
Below are the mapping values to be used to translate to the standard ZIC, ZIO



Valve Limit Switches Configured in Mixed State.

In this example, ZSC is configured as normally opened and ZSO is configured as normally closed. This is mainly an example to indicate flexibility to handle different arrangements that might arise.

Below are the mapping values to be used to translate to the standard ZIC, ZIO states.



Single Limit Switch

The function block can be used for single limit switch valve state indication by using only single limit connected to either ZSC or ZSO and configuring INBET, CLOSE, OPEN and BAD. For example, if there is only a single limit switch indication where

ZS – False Valve Open

ZS – True Valve Closed

Connect ZS to ZSC, leave ZSO unconnected. The states will be:

INBET 2 (don’t care)

CLOSE 1

OPEN 0

BAD 3 (don’t care)

Mode Operation

The function block has modes Manual (MD=False) and Auto (MD=True).

In Manual Mode, the output XY tracks OP set from SCADA.

In Auto Mode, the output XY tracks HS which is driven by program logic.

When in Auto, OP will track HS so that Mode change from Auto to Manual is bumpless.

Local/Remote

This input is normally connected to Hand/Off/Auto switch. When the LOCAL input is true, OP will track the PV state of valve and control of valve will be via a local control panel. While in LOCAL, commands will not be accepted from SCADA OP or Logic controlled HS regardless of MD setting.

Permissive and Safety Interlock.

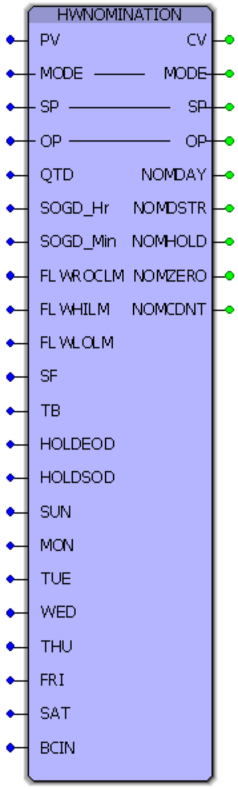
A permissive is available for Open and Close commands. If the respective permissive is not true, then that command cannot be executed. If the permissive becomes false after command is issued, the command is unaffected.

A safety interlock input (SI) of True will command the valve to the state set by SAFEOP (False = Close, Open = True). The safety interlock will take the highest precedence in Auto or Manual and will override a permissive.

HWNOMINATION

Description

TThis function block is used for nomination control where a fixed amount of product is delivered over a day. The nomination function block continually calculates a set point based on remaining nomination, remaining time in the day and amount already delivered as calculated from a totalizer function block. The calculated set point is used to provide a remote set point to a flow control PID in cascade mode. The nomination function block can be configured with a week of nomination values. The figure bellows shows how the nomination function block is integrated with a totalizer and PID controller to provide a complete nomination control solution.



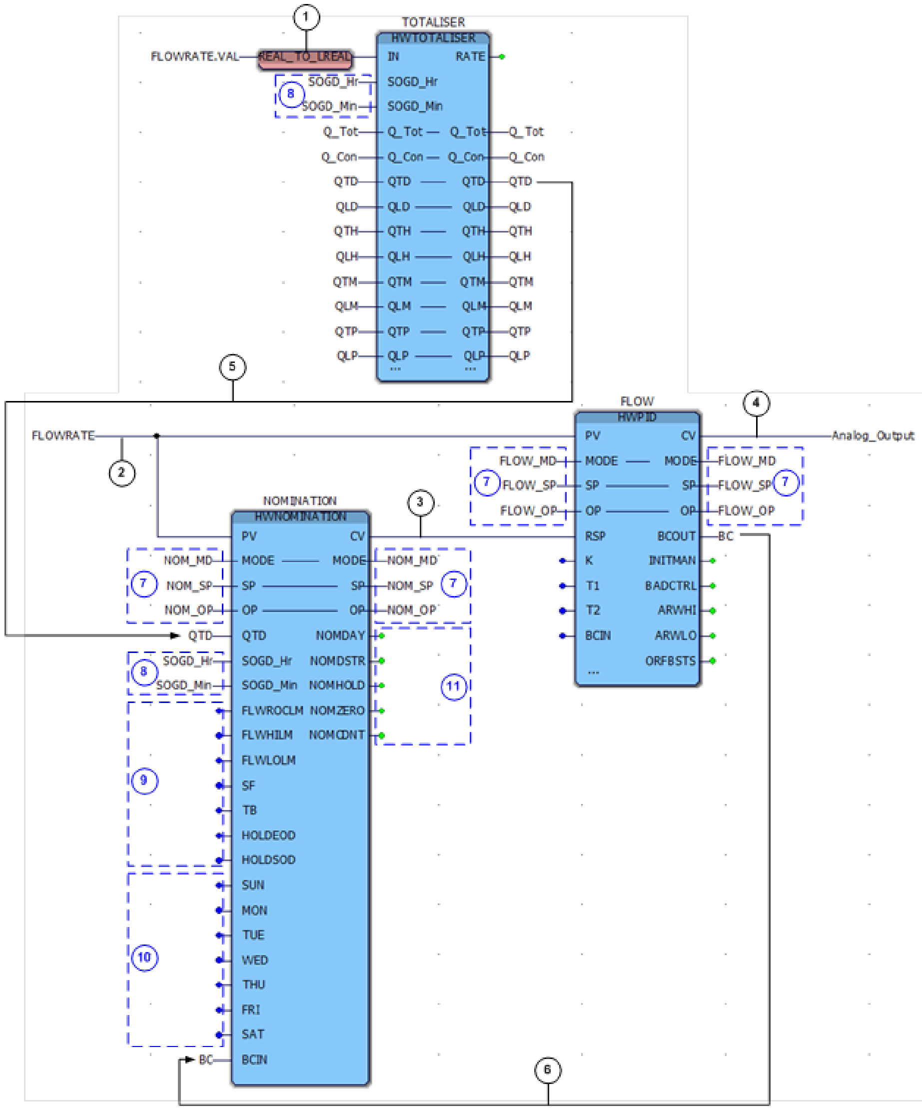
Input

Parameter	Data Type	Description
PV	Analog_Type	Flow Process variable input. AI_Type contains value and quality flags
Mode	INT	Sets Mode. 0- Manual – OP can be set manually 1- Fixed Automatic – Nomination SP can be set 2- Scheduled Automatic – Nomination SP read from daily nomination values.
SP	REAL	Nomination Setpoint. From SCADA.
OP	REAL	Calculated Flow Rate output.
QTD	REAL	Accumulation in current gas day. This should be in the same engineering units as nomination values.

Parameter	Data Type	Description
SOD_Hr	INT	Contract Start of Day Hour
SOD_Mn	INT	Contract Start of Day Minute
FLW ROCLM	REAL	Maximum rate of change of Control Variable output in EU/min Default – 0 , no rate limiting
FLWHILM	REAL	Maximum Flow Rate Limit in EU
FLWLOLM	REAL	Minimum Flow Rate Limit in EU
SF	REAL	Scale factor between flow units and nomination Units. For example if nomination is in energy and flow is in volumetric units, Scale Factor would be the Heating Value.
TB	INT	Time Base of flow rate units 0 – Seconds 1 – Minutes 2 – Hours 3 – Days
HOLDEOD	DINT	Time period to hold SP at end of gas day in seconds. This is to prevent OP swinging wildly as time remaining approaches 0.
HOLDSOD	DINT	Time period to hold OP at beginning of gas day in seconds. This is to prevent OP swing wildly if accumulation today rolls over just after start of gas day. This may happen if daily accumulation is read from a separate flow computer. Note that if the Nomination is 0.0, this hold will be overridden and OP will immediately go to 0.0.
SUN..SAT	REAL	Daily nomination value for each day of the week.
BCIN	BackCalc_Type	Back Calculation Input. This comes from Back Calculation Output of downstream block

Output

Parameter	Data Type	Description
CV	Analog_Type	Control Variable that is normally used to drive the analog output to a control device or as a remote SP to a PID block for flow control.
NOMDAY	INT	Current Nomination Day index (0-Sun..6-Sat)
NOMDSTR	STRING	Current Nomination Day
NOMHOLD	BOOL	OP currently in HOLD
NOMZERO	BOOL	Nomination is 0.0. This can be used to force a control valve to 0.0 if desired.
NOMCDNT	DINT	Number of seconds remaining in contract day



The following describes the main connections in the figure above.

1. The input is a flow rate either from an analog input or from a flow calculation such as AGA for gas or API for hydrocarbon liquids. The Totaliser function block is used to totalize the flow to provide the amount of flow accumulated in the current day. The totalizer can also be used for all other totals. (See HWTOTALISER help for further details). Note that the flowrate value needs to be converted to a LREAL data type. To ensure maximum accuracy of the nomination controller, the Totaliser should be executed before the Nomination function block.

2. The same flow rate input used by the Totaliser provides the PV for the Nomination and the flow control PID function blocks.

3. The Nomination control variable is connected to the flow control PID remote set point to provide a flow control cascade set point.
4. The control variable from the PID function block is connected to the flow control element, usually via an analog output to a control valve.
5. The totalizer QTD (Flow in current day) is connected to the Nomination function block input QTD.
6. The BCOUT of the PID is connected to the BCIN of the nomination function to provide initialisation data to the Nomination function block for bump less operation in cascade connection.
7. The Mode, Set Point and Output are typically interfaced to analog SCADA points to provide SCADA monitoring and control of the nomination and flow controllers. It should be noted that the output (OP) of the nomination controller is a flow rate in configured engineering units whereas the output of the PID controller is 0-100%
8. The Start of day (SOGD_Hr and SOGD_Min) are connected to the Totaliser and Nomination function blocks to ensure they both use the same start of day for nomination calculations.
9. This group of parameters is used to configure the Nomination calculation parameters. These are described below
10. The group of parameters defines the nomination values to be used for each day of the week.
11. This group of outputs can be used to monitor the nomination control status. These are described below.

Modes of Operation

The Nomination function block has three modes of operation.

- Manual Mode. In this mode, the flow rate can be set manually by the output (OP) of the nomination control
- Fixed Nomination Mode. In this mode, the Nomination can be set in the set point (SP) of the nomination control.

- Scheduled Nomination Mode. In this mode, the nomination set point (SP) is automatically updated at the start of the current day (defined by SOGD_Hr and SOGD_Min) with the corresponding nomination day value defined by SUN, MON, TUE, WED, THU, FRI and SAT.

Nomination Calculation Configuration Parameters

The nomination flow set point is calculated as follows

CV = (SP – QTD)/R_Day

Where

- CV = Calculated Flow set point
- SP = Daily Nomination
- QTD = Totalised flow in current day
- R_Day = Amount of time left in current day

This calculation can become very unstable towards the end of the day as the denominator and numerator both approach zero. Several calculation parameters are required to ensure stable operation at the end of the nomination day.

Calculated Flow Rate Limits

- FLWROCLM defines the maximum rate of change of the flow setpoint in engineering units/min. If set to zero, no rate limiting applies. This setting is useful to ensure a smooth transition of the flow set point when the nomination changes at the start of the nomination day.
- FLWHILM and FLWLOLM define the maximum and minimum calculated flow rate values

Nomination Day Rollover

In addition to flow rate limits, the following time settings are used to prevent bumps when the nomination day rolls over.

- HOLDEOD – Hold period at end of day (seconds). This defines the period where flow set point calculation is stopped and frozen to prevent issues caused by denominator of flow calculation approaching zero at the end of the day. If this value is set too small, some instability in the flow set point may occur just before end of nomination day. If it is set too large, the actual flow nomination may not equal the desired nomination. Typically, this value can be set to 300 seconds to provide a good compromise between stability and accuracy.

- HOLDSOD – Hold period at start of day (seconds). This setting is usually only applicable to applications where the totalising of QTD is done in a separate flow computer. This setting will hold the flow set point calculation for a short period at the beginning of the nomination day to prevent any bumps that may be caused by time sync issues between the RTU2020 and external flow computer rollover of QTD. For example, if the flow computer time is lagging the RTU2020 time, when the new nomination day starts, the QTD will still be set at value for previous day. Therefore, it will appear that nomination has been met and calculated flow set point will go to zero. Typically, this value is set to cover expected time drift and scan time update of external flow computer values. Normally a value of 60 seconds is sufficient.

Nomination Flow Units

These parameters determine calculated flow rate units.

- Scale Factor (SF) Scale factor between flow rate units and nomination units.
- Time Base (TB) sets the time base used for calculated flow rate units. TB can have the following values

  0. Seconds (Default)
  1. Minutes
  2. Hours
  3. Days

For example, if the nomination values are in ksm3 and flow rate units are in sm3/hr the following values will be used

SF = 1000.0

TB = 2

Nomination Status

The nomination function block has several outputs for monitoring the status of calculation.

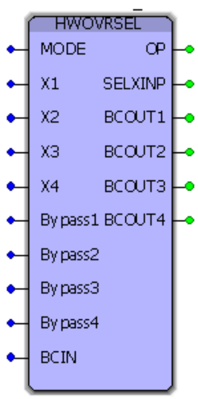
- NOMDAY – Integer to identify currently used nomination day value (0-SUN…6-SAT)
- NOMDSTR – String representation of current nomination day

- NONHOLD – Flag indicating that nomination calculation is in hold state due to HOLDEOD and/or HOLDSOD time settings.
- NOMCDNT – Seconds remaining in current nomination day.
- NOMZERO – Flag that is set if current nomination is zero. This flag can be used to force flow control output to zero if a zero nomination is scheduled.

HWOVERSEL

Description

This function shall be used to provide override select of either the maximum or minimum of up to four initializable inputs.



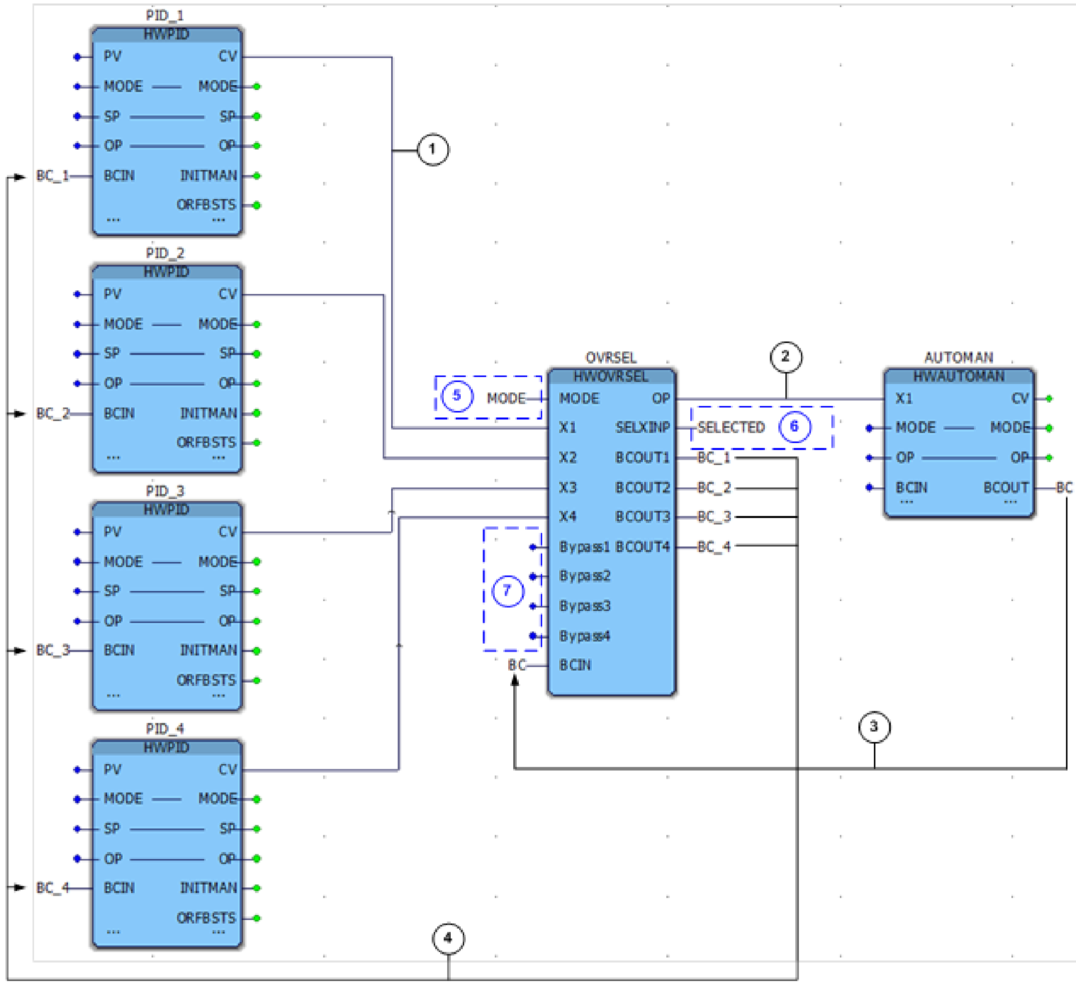
Input

Parameter	Data Type	Description
MODE	BOOL	False – OUT is the Minimum of IN1 to 4 True – OUT is Maximum of IN1 to 4
X1 to 4	Analog_Type	Process variable input. AI_Type contains value and quality flags. Generally this is connected to the CV of a regulatory FB such as PID.
Bypass1 to 4	BOOL	Bypass (ignore) input from override select
BCIN	BackCalc_Type	Back Calculation Input. This comes from Back Calculation Output of downstream block connected to output

Output

Parameter	Data Type	Description
OP	Analog_Type	Output that is normally used to drive the analog output to a control device which is the Minimum or Maximum of inputs.To enable mode control of final control element, an AUTOMAN block or PID block should be inserted beween HWOVRDSEL and final control output.
SELINP	INT	Selected Input
BCOUT1 to 4	BackCalc_Type	Back Calculation Output. This goes to Back Calculation Input of upstream block connected to each input.

This function block is used to provide override select of either the maximum or minimum of up to four control inputs from HWPID function blocks. This function block is used when control with restraints is required, for example, a flow control loop with a pressure override. An example is shown below. The PID and AUTOMAN function blocks have pins hidden for clarity. As a minimum, the HWORDSEL should have two or more HWPID connected to the inputs and a HWAUTOMAN connected to the output for overall mode and manual output control.



The following describes the main connections in the figure above.

1. The inputs X1 to X4 are connected to the control variable outputs of the PID function blocks using an Analog_Type connection to pass control variable and status information. If less than 4 inputs are used, simply leave unused inputs disconnected (or hidden).
2. The OP parameter connection is used to send selected output control data to the AUTOMAN block. The OP parameter is an Analog_Type as per the X1-X4 inputs. The AUTOMAN function block provides overall mode control for override select. The CV of the AUTOMAN function block is usually connected to an analog output via a CV2AO function block for the final control device.

3. Use the BCOUT/BCIN connection to carry secondary data from the HWAUTOMAN block to the HWOVRDSEL function block. This will in turn be passed to the selected HWPID function block to complete the initialisation path from end to end to carry the following information.
   - Anti-Reset Windup Status (ARWHI, ARWLO): Indicates if the HWAUTOMAN is at its high or low limit.
   - Initialization Request Flag (INITMAN): Used to request initialization. If the flag is set by AUTOMAN, the selected PID block initializes itself
   - Initialization Value (INITVAL): Used for initialization when INITMAN true.
4. Use the BCOUT1..BCOUT4 connections to carry secondary data from the HWAUTOMAN block to the respective PID block BCIN pin. In addition to the initialisation data described in point 3 above, the HWORDSEL function block sets the following data .

   Override Status (ORFBSTS): This flag indicates whether this PID input is the selected control strategy. If the block is in an unselected strategy, it uses Override Feedback Value (ORFBVAL) to initialize Control value; this is calculated to prevent “wind-up” if this PID block input is unselected.
5. The MODE determines whether the Minimum (MODE=False) or Maximum (MODE=True) of X1..X4 is selected for the output OP.
6. The SELECTED output indicates which input is currently selected for control. This can be indicated on SCADA.
7. These pins can be used to bypass any of the inputs. When a bypass is active, the associated PID input will be set to INITMAN and its control variable will track the selected output.

Operation

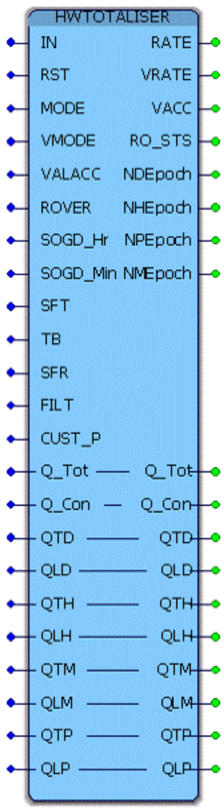
Only Bypass control is available for the OVRDSEL function block. When a Bypass input is set to True, the PID input associated with the bypass will be set to INITMAN and its Control Variable will track the selected OP of the OVRDSEL function block until the Bypass input is returned to False.

Overall Mode control, manual output, rate of change and output limiting control is set by the AUTOMAN function block. In a typical override strategy, the AUTOMAN function block is interfaced to SCADA. Additionally, the OVRDSEL output pin, SELXINP can be used to display which PID control is active.

HWTOTALISER

Description

The HWTOTALISER is used to accumulate totals by periodically integrating or accumulating an input value to a totalised value and daily, hourly, monthly and custom period subtotals.



Input

Parameter	Deta Type	Description
IN	REAL	Input value. For accumulation modes, this will be a delta value derived from a pulse input type device, for example

Parameter	Deta Type	Description
		AGA7. For integration modes, this will be rate value that is integrated over time. **NOTE:** For best results, the HWTOTALISER must be executed every 1 second or less
RST	BOOL	Resets accumulation on a rising edge.
MODE	INT	Totaliser Mode 0 – Accumulation Mode – if value is NaN, use zero for totalisation 1 – Accumulation Mode – if value is NaN, use last good delta for totalisation 10 – Integration Mode – if value is NaN, use zero for totalisation 11 – Integration Mode – if value is NaN, use last good value for totalisation
VMODE	BOOL	When VMODE transitions to True, Accumulation will continue with value of IN at that time. This allows meter validation to be performed while accumulation continues.
VALACC	BOOL	While true when in VMODE is true, validation accumulation will occur
ROVER	LREAL	Rollover Value for Totaliser output Q_Tot. If not connected, no rollover will be applied.
SOD_Hr	INT	Contract Start of Day Hour
SOD_Mn	INT	Contract Start of Day Minute
SFT	LREAL	Accumulator Scale Factor (for example if IN is in litres and accumulation is in kilolitres, SF = 0.001)
TB	INT	Time Base of input rate for integration modes or time base for rate output in accumulation modes 0 – Seconds 1 – Minutes 2 – Hours

Parameter	Deta Type	Description
		3 – Days
SFR	REAL	Rate Multiplier when in accumulation modes (for example if IN is in litres and rate is in kilolitres/hr, SFR = 0.001)
FILT	REAL	Rate filter time constant in seconds. Filtering is based on a first order filter function that behaves like an RC filter rather than an averaging filter.
CUST_P	INT	Custom totalisation period in minutes

Output

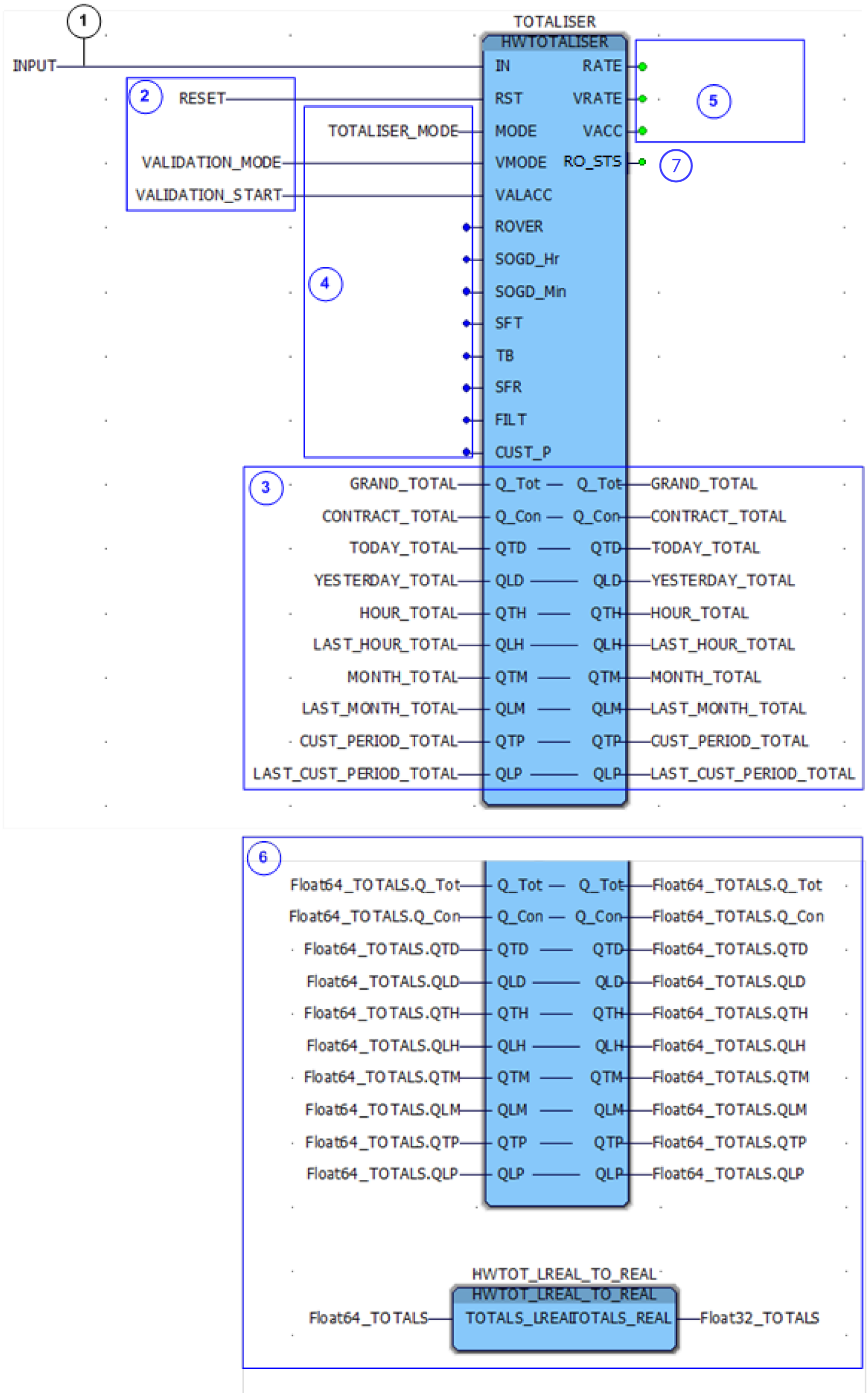
Parameter	Data Type	Description
RATE	REAL	Rate of change of totaliser when in accumulation modes as defined by TB, SFR and FILT. This is used for determining say flow rate from a pulse input turbine meter. When in integration modes, RATE = IN.
VRATE	REAL	Rate of change of totaliser when in validation mode.
VACC	REAL	Accumulation while in validation mode
RO_STS	WORD	A Rollover status output (hidden by default) has been added for monitoring. The status word is broken into 4 nibbles Least Significant Nibble - Day Rollover Status 2nd Nibble - Hour Rollover Status 3rd Nibble - Month Rollover Status Most Significant Nibble - Custom Period Rollover Status **The rollover statuses are** 0 - Normal Rollover 1 - Missed Rollover - rollover forced on power up 2 - Missed 2 or more rollovers - rollover forced and last period total zeroed on power up C - 12(DEC) - Start of Day Time Changed - Initialise Boundary Epoch times

Parameter	Data Type	Description
		D - 13(DEC) - Significant negative time shift - Initialise Boundary Epoch times E - 14(DEC) - Warm start has occurred F - 15(DEC) - Cold start has occurred - Initialise Boundary Epoch times mode
NDEpoch	DINT	Epoch secs of next day boundary (hidden by default)
NHEpoch	DINT	Epoch secs of next hour boundary (hidden by default)
NPEpoch	DINT	Epoch secs of next custom period boundary (hidden by default)
NMEpoch	DINT	Epoch secs of next month boundary (hidden by default)
Q_Tot	LREAL	Total non resetting value
Q_Con	LREAL	Contract snapshot of Q_Tot perfomed at start of day defined by SOD_Hr and SOD_Mn
QTD	LREAL	Accumulation in this contract day
QLD	LREAL	Accumulation in last contract day
QTH	LREAL	Accumulation in this contract hour
QLH	LREAL	Accumulation in last contract hour
QTM	LREAL	Accumulation in this contract month
QLM	LREAL	Accumulation in last contract month
QTP	LREAL	Accumulation in this contract custom period defined by CUST_P
QLP	LREAL	Accumulation in last contract custom period defined by CUST_P

General Notes:

- All internal calculations use 64-bit floats and the Kahan summation technique to minimize accumulation errors.
- Integration modes use trapezoidal rule.
- Filtering uses a simple first order filter and is only applicable to accumulation modes when using pulse input type applications.

The HWTOTALISER function block is used to accumulate totals by periodically integrating or accumulating an input value to totalised values of hourly, daily, monthly and user defined periods. The example below shows the main configuration groups.



1
2
3
4
5
6
7
TOTALISER
HWTOTALISER
INPUT
RESET
TOTALISER_MODE
VALIDATION_MODE
VALIDATION_START
IN
RST
MODE
VMODE
VALACC
ROVER
SOGD_Hr
SOGD_Min
SFT
TB
SFR
FILT
CUST_P
RATE
VRATE
VACC
RO_STS
GRAND_TOTAL
CONTRACT_TOTAL
TODAY_TOTAL
YESTERDAY_TOTAL
HOUR_TOTAL
LAST_HOUR_TOTAL
MONTH_TOTAL
LAST_MONTH_TOTAL
CUST_PERIOD_TOTAL
LAST_CUST_PERIOD_TOTAL
Q_Tot
Q_Con
QTD
QLD
QTH
QLH
QTM
QLM
QTP
QLP
Float64_TOTALS.Q_Tot
Float64_TOTALS.Q_Con
Float64_TOTALS.QTD
Float64_TOTALS.QLD
Float64_TOTALS.QTH
Float64_TOTALS.QLH
Float64_TOTALS.QTM
Float64_TOTALS.QLM
Float64_TOTALS.QTP
Float64_TOTALS.QLP
HWTOT_LREAL_TO_REAL
Float64_TOTALS
TOTALS_LREAL
TOTALS_REAL
Float32_TOTALS

The following describes the main connections in the figure above.

1. The input value which is to be totalized. This data must be converted to a long real (LREAL) datatype.
2. Optional control inputs which can be used to reset totals or to accumulate validation data for meter validation.
3. Variables to hold the various period totals. Note that all pins must be configured. The totals data is stored as LREAL (64-bit Floating Point). For most SCADA interfaces, these will need to be converted to REAL (32-bit Floating Point) before mapping to SCADA mapping.
4. Totalizer mode and configuration options. The totalizer can operate in four different modes.

   0. Accumulation Mode where IN value is added to totals – if IN value is NaN, use zero instead

   1. Accumulation Mode where IN value is added to totals – if IN value is NaN, use last good IN value.

   10. Integration Mode where IN is integrated with respect to time – if IN value is NaN, use zero instead

   11. Integration Mode where IN is integrated with respect to time – if IN value is NaN, use last good IN value
5. Optional monitoring values for calculated rate when in Accumulation Mode 0 or 1 and Validation Accumulation results for all modes.
6. Alternative method of storing totals in the user defined variable type Totaliser_Data_LREAL. The function block HWTOT_LREAL_TO_REAL is used to convert to 32-Bit Floating Point data type Totaliser_Data_REAL for use in SCADA where 64-bit Floats are not supported. The advantage of this technique is that only a single variable is required to store all total data.
7. A Rollover status output (hidden by default), this can be useful to investigate the behaviour of RTU. This output is not required for configuration and thus won’t affect existing configurations.

Modes of Operation

The mode of operation will be determined by the physical type of measurement represented by the IN value. If the IN value represents a flow or rate value in Units/Time, integration modes 10 and 11 are applicable. If the IN value represents a counter type value such as meter pulses or motor starts, then accumulation modes 0 or 1 are applicable.

Total Reset

On a, rising edge of RST, all totaliser totals will be reset to zero. Normally this function is only used in engineering mode within control environment but it can be made available to SCADA if required.

Validation Mode

These inputs can be used for meter validation functions. If this isn’t required, these pins can be hidden.

When VMODE transitions to True, the current value of IN will be frozen and totalisation will continue with this value. While VMODE is true, changes to IN will not affect running totalisation and it will be directed to the validation total VACC and validation calculated rate VRATE.

While VMODE is true, when VALACC transitions to true, the validation total output VACC will be reset to zero and validation totalisation will start.

At the completion of validation VMODE and VALACC are set to False. This will unfreeze IN value and totals will track live value.

The implementation of how these two flags are used will depend on your validation procedures. For examples, in some applications, validation cannot be performed until live flow has reduced to zero. These details can be handled with external logic.

Configuration Parameters

The TOTALISER function block configuration parameters determine how totals are calculated.

Rollover

This value represents the maximum value that a total can reach before it is rolled over back to zero. This simulates the operation of a counter or odometer. If left as zero or not connected, the total will keep accumulating to the maximum value of a 64-bit float. If you can read the 64-Bit Float value by SCADA this will not present a problem.

However, if the total needs to be converted to a 32-bit Float, consideration of a suitable Rollover value is required to prevent the totals losing resolution. Typically, if you require totals to have a resolution of 1 unit, the rollover value should be set to 1000000 for 32-bit floats.

Start of Day

SOGD_Hr and SOGD_Min specify the start of day for period total rollover boundaries. For example, if SOGD_Hr = 8 and SOGD_Min = 30, then at 8:30 AM, the total for current day will be copied to yesterday and current day total will be reset to start totalizing for new day.

For hourly values, the hour rollover will occur at 30 minutes past the hour.

Scale Factor Total

Scale Factor Total (SFT) determines the scale factor to be used for totals. For example, if your input IN is measuring litres and you want totals to be in kilolitres, SFT should be set to 0.001. If not connected, the default value is 1.0.

Time Base

Time base (TB) sets the time base used for input rate in integration modes 10 and 11 or the time base for RATE calculations in accumulation modes 0 and 1. TB can have the following values

0. Seconds (Default)

1. Minutes

2. Hours

3. Days

Scale Factor for Rate

Scale factor for Rate (SFR) determines the scale factor for calculated RATE when using accumulation modes 0 and 1. For example, if the IN value represents 1 pulse/litre and you wish to calculate kilolitres/hour, SFR would be set to 0.001 and time base TB would be set to Hours (2). The default value if not used is 1.0.

Filter for Rate Calculation

When calculating RATE in accumulation modes 0 or 1, depending on the number of pulses counted per execution cycle, significant variation in RATE can be seen from cycle to cycle. The FILT parameter can be used to apply a filter time constant in seconds to help smooth out variations.

Custom Total Period

The parameter CUST_P can be used for defining a custom totalisation period in minutes. For example, if you wish to have totals by shift, CUST_P would be set to 480 minutes (8 hours). The start of a custom period is defined by the start of day settings SOGD_Hr and SOGD_Min. The value of CUST_P should divide into a 24-hour period evenly thus the largest valid value is 720 minutes (12 hours) and the smallest valid value is 1 minute.

Totals

The totaliser function block provides a series of totals for current period and the last period as follows.

- Q_Tot – Running non-resetting total not affected by time periods. This total will accumulate until it reaches the ROLLOVER value if configured or until the RST is triggered.
- Q_Con – Contract total. This is a snapshot of Q_Tot at the start of the day defined by SOGD_Hr and SOGD_Min. This is equivalent to a meter read at the beginning of each day. This value is useful for reconciling billing information should there be an extended communications outage to SCADA since estimated quantities during outage can be re-aligned to actual values using Q_Con once communications are restored.
- QTH, QLH – Totals for current and last hour defined by start of day minute boundary SOGD_Min
- QTD, QLD – Totals for current day and yesterday defined by start of day at SOGD_Hr and SOGD_Min
- QTM, QLM – Totals for current calendar month and last calendar month based on the boundary defined by SOGD_Hr and SOGD_Min
- QTP, QLP – Totals for current and last custom periods as deined by SOGD_Hr, SOGD_Min and CUST_P

Totaliser Algorithms

All internal calculations of the totaliser function block use 64-bit floating point. To minimise accumulation errors caused by adding small numbers to large numbers, the Kahan summation technique is used.

http://en.wikipedia.org/wiki/Kahan_summation_algorithm

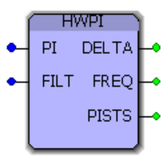
Integration modes use the trapezoidal technique.

http://en.wikipedia.org/wiki/Trapezoidal_rule

HWPI

Description

This function block is connected to a pulse input channel and outputs a delta pulse count suitable for metering calculations such as AGA7/9.



Input

Parameter	Data Type	Description
PI	Pulse_ Input_Type	Connected to a pulse input channel
FILT	REAL	First order filter time constant in minutes for smoothing calculated frequency output FREQ.

Output

Parameter	Data Type	Description
DELTA	LREAL	Delta counts since last execution. This output can be connected to meter calculations such as AGA7/9
FREQ	REAL	Calculated pulse frequency in Hz. This value is

Parameter	Data Type	Description
		smoothed using FILT setting
PISTS	STRING	Pulse input channel status message

If users want to calculate a higher accuracy frequency using the pulse input channel of ControlEdge 2020 onboard I/O or Expansion I/O, a frequency estimation program should be configured and run at the highest possible fastest task (20ms).

> **NOTE:** It also applies to HWPIACC function block.

To configure the program

1. Create a POU with program option with ST, and add the below frequency estimation code.

   > **NOTE:** Global variables in **bold** must be created manually first and name them as desired.

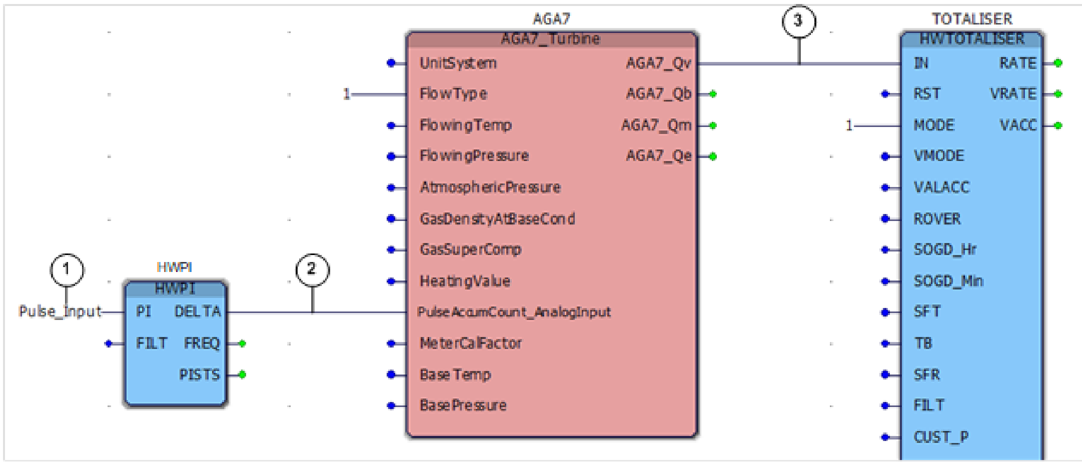
   > **NOTE:** "PI_1" is the variable name assigned with specific pulse input channel on "Configure I/O" page.

   > **NOTE:** "HWPI_1" is an instance of HWPI function block.

   ```
   COUNT := COUNT + UDINT#1;
   IF COUNT >= UDINT#50 THEN
   HWPI_1(PI:= PI_1,FILT:= filt);
   delta:=HWPI_1.DELTA;
   freq:=HWPI_1.FREQ;
   Pists:=HWPI_1.PISTS;
   COUNT := UDINT#0;
   END_IF;
   ```

2. Associate the POU to the fastest cyclic task (20ms). The "freq" parameter provides the frequency estimation.

Below is a typical example of how the HWPI function is used for a metering application.



The following describes the main connections in the figure above.

1. The input (PI) is connected to a pulse input channel to receive raw pulses.
2. The output (DELTA) will be the number of pulses counted since the program was last executed. This value is a LREAL type. This can be scaled appropriately. In the example, an AGA7 calculation is used to convert pulses into engineering units.
3. Typically, the scaled delta pulses would be connected to a Totaliser function block in accumulation mode to total the scaled value.

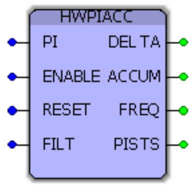
Frequency Calculation

In addition to providing the number of pulses counted in the last execution cycle, the pulse rate or frequency is calculated in Hertz. If the number of pulses sampled in an execution cycle is low, the FREQ output can vary significantly from cycle to cycle. The FILT parameter can be used to apply a filter to FREQ output to help smooth the variations.

HWPIACC

Description

This function block is an extension of the HWPI function block that includes a local pulse accumulator register (ACCUM). This function block is connected to a pulse input channel and outputs a delta pulse count suitable for metering calculations such as AGA7/9 and has a count accumulator that can be enabled and reset.



Input

Paramter	Data Type	Description
PI	Pulse_ Input_Type	Connected to a pulse input channel
ENABLE	BOOL	Enables accumulation and delta/freq calculation when TRUE.
RESET	BOOL	Resets ACCUM on a rising edge.
FILT	REAL	First order filter time constant in minutes for smoothing calculated frequency output FREQ.

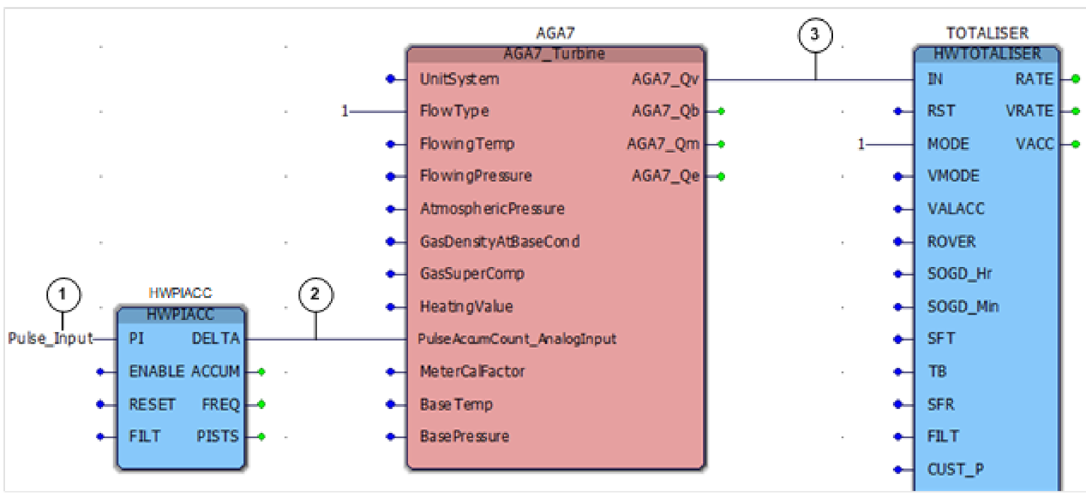
Output

Parameter	Data Type	Description
DELTA	LREAL	Delta counts since last execution. This output can be connected to meter calculations such as AGA7/9.
ACCUM	UDINT	Accumulates pulses from pulse input when ENABLE is TRUE. ACCUM range is 0- 4294967295
FREQ	REAL	Calculated pulse frequency in Hz. This value is smoothed

Parameter	Data Type	Description
		using FILT setting
PISTS	STRING	Pulse input channel status message

> **NOTE:** If users want to calculate a higher accuracy frequency using the pulse input channel of ControlEdge 2020 onboard I/O or Expansion I/O, a frequency estimation program should be configured and run at the highest possible fastest task (20ms). For how to configure the program, see the content in HWPI function block.

Below is a typical example of how the HWPIACC function is used for a metering application.



The following describes the main connections in the figure above.

1. The input (PI) is connected to a pulse input channel to receive raw pulses.
2. The output (DELTA) will be the number of pulses counted since the program was last executed. This value is a LREAL type. This can be scaled appropriately. In the example, an AGA7 calculation is used to convert pulses into engineering units.
3. Typically, the scaled delta pulses would be connected to a Totaliser function block in accumulation mode to total the scaled value.

Frequency Calculation

In addition to providing the number of pulses counted in the last execution cycle, the pulse rate or frequency is calculated in Hertz. If the number of pulses sampled in an execution cycle is low, the FREQ output can vary significantly from cycle to cycle. The FILT parameter can be used to apply a filter to FREQ output to help smooth the variations.

Pulse Accumulation

The function block includes a local pulse accumulator. This operates independently from the from the DELTA used for external accumulation. The inputs ENABLE and RESET can be used to enable and reset pulse accumulation stored in the output ACCUM. This output can be used for operations such as calibration or validation where a fixed number of pulses are injected into pulse input.

HWPID

Description

The HWPID block is a regulatory control block that operates as a proportional-integral-derivative (PID) controller. It supports the Ideal form of calculating the PID terms. The Ideal form is often called the digital-computer version of the PID controller. This is the same form used in the C300/C200 controller.

It supports PI, PD, PID algorithms as function. It accepts two analog inputs- process variable (PV) and set point (SP) and produces output calculated to reduce the difference between PV and SP.

The HWPID block has two principle inputs - a process variable (PV) and a set point (SP). The difference between PV and SP is the error and this block calculates a control output (CV) that should drive the error to zero.

The following equations are supported:

- Proportional, Integral, and Derivative (PID) on the error
- Proportional and Integral (PI) on the error and Derivative (D) on changes in PV

- Integral (I) on the error and Proportional and Derivative (PD) on changes in PV
- Integral (I) only

The mode (Mode), set point (SP) and output (OP) are normally mapped to SCADA for monitoring and control. The HWPID block may be used in a single control loop or with multiple PIDs in a cascade strategy using the Remote Set Point (RSP) to receive the CV from another PID block.

Input

Parameter	Data Type	Description
PV	Analog_ Type	Process variable input. AI_Type contains value and quality flags. This usually comes from a HWDACA block.
Mode	INT	Sets Mode or logic. 0- Manual 1- Automatic 2- Cascade
SP	REAL	Setpoint From SCADA or logic.
OP	REAL	Manual Output from SCADA or logic.
RSP	Analog_ Type	Remote setpoint from an upstream PID block’s CV for SCADA operation.
CTLACTN	BOOL	False – Forward Acting (Default) True – Reverse Acting
CTLEQN	INT	Control Equation 0. EQN A Proportional, integral, derivative act on error (PV-SP) 1. EQN B Proportional, integral act on error (PV-SP), derivative acts on PV changes 2. EQN C Integral acts on error (PV-SP), proportional, derivative acts on PV changes 3. EQN D Integral control only on error (PV-SP)
BADCTL	INT	Bad Control Option as per C200/C300

Parameter	Data Type	Description
		0. No Shed 1. Shed Hold 2. Shed Low 3. Shed High 4. Shed Safe OP
SPTRACK	BOOL	False – No Tracking (Default) True – SP tracks PV in manual mode.
K	REAL	Proportional Gain
T1	REAL	Integral Time – Minutes
T2	REAL	Derivative Time – Minutes
DB	REAL	Control Error (PV-SP) Deadband
SPHILM	REAL	Maximum Setpoint Limit in EU
SPLOLM	REAL	Minimum Setpoint Limit in EU
OPROCLM	REAL	Maximum rate of change of output in %/min Default – 0 , no rate limiting
OPHILM	REAL	Maximum output Default – 100%
OPLOLM	REAL	Minimum output Default – 0%
SI	BOOL	Safety Interlock. False – No shutdown (default) True – Shutdown using SIOPT
SAFEOP	REAL	Shutdown Control Variable Target value.
SIOPT	INT	Safety Option as per C200/C300 0. No Shed 1. Shed Hold

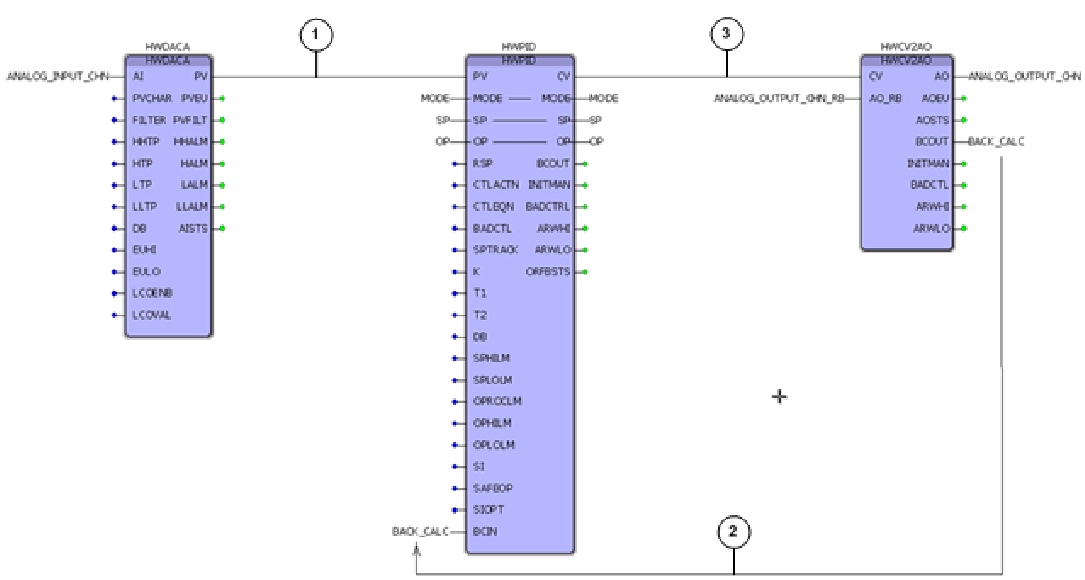
Parameter	Data Type	Description
		2. Shed Low 3. Shed High 4. Shed Safe OP
BCIN	BackCalc_ Type	Back Calculation Input. This comes from Back Calculation Output of downstream block

Output

Parameter	Data Type	Description
CV	Analog_Type	Control Variable that is normally used to drive the analog output to a control device.
BCOUT	BackCalc_Type	Back Calculation Output. This goes to Back Calculation Input of upstream block
INITMAN	BOOL	FB InitMan has been requested by downstream block
BADCTRL	BOOL	Bad Control Option is active
ARWHI	BOOL	FB is in high windup status
ARWLO	BOOL	FB is in low windup status
ORFBSTS	BOOL	FB is using Override Feedback value from OVRSEL
ARWNETHI	BOOL	Antireset Windup sent to upstream block via BCOUT
ARWNETLO	BOOL	Antireset Windup sent to upstream block via BCOUT
CV_CEE	REAL	Internal PID calculated variable before OP limiting. This can be used as a diagnostic.
DELCV	REAL	Calculated change in control variable
DeLCp	REAL	Calculated change of Proportional Term
DelCi	REAL	Calculated change of Integral Term
DelCd	REAL	Calculated change of Derivative Term

Simple Loop

The following figure shows a simple single loop controller using a HWDACA block to connect the PV to an analog input channel and a HWCV2AO block to connect the control output CV to an analog output channel. If you do not need any alarming or filtering, the HWAI2PV function block can be used instead of the HWDACA function block.



The following describes the main connections in the figure above.

1. Use the PV parameter connection to carry data from the analog input to the HWPID block. PV is an Analog_Type which carries the PV value, PV status and PV range information.
2. Use the BCOUT/BCIN connection to carry secondary data from the CV2AO block to the HWPID block. Note that CV2AO requires the analog output read back connection to provide secondary data. The secondary data in the BACKCALC_TYPE data connection between CV2AO BCOUT pin and the HWPID BCIN includes the following information.
    a. Anti-Reset Windup Status (ARWHI, ARWLO): Indicates if the secondary's initialize input (which is this block's output) is at its high or low limit.
    b. Initialization Request Flag (INITMAN): Used to request initialization. If the flag is set by CV2AO, the PID block initializes itself

   c. Initialization Value (INITVAL): Used for initialization when INITMAN true.

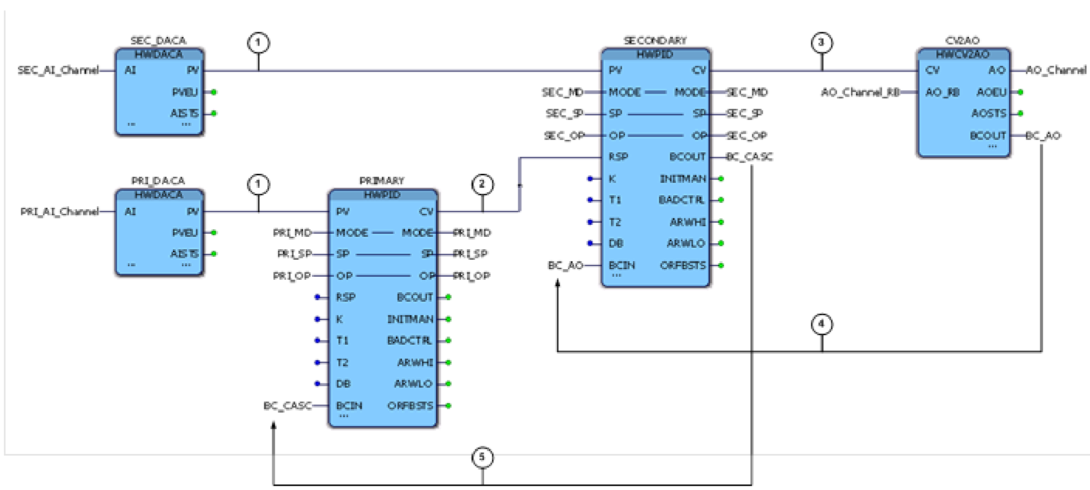
   d. Override Status (ORFBSTS): If a block is in an override strategy, this flag indicates whether it is the selected strategy or not. If the block is in an unselected strategy, it uses Override Feedback Value (ORFBVAL) to initialize Control value; this is calculated to prevent “wind-up” if this PID block is unselected.

   e. Cascade Flag: Indicates that secondary block has the Remote Set Point connected in a cascade strategy.

3. Use the CV parameter connection to send output data to the Analog Output Channel (CV2AO) block. The CV parameter is an Analog_Type as per the PV.

Cascade Loop

The following figure shows two PID controllers being used for simple cascade. In this example, optional pins are hidden to help declutter the view. If you do not need any alarming or filtering, the HWAI2PV function block can be used instead of the HWDACA function block.



The following table describes the main connections in the figure above.

1. Use the PV parameter connection to carry data from the analog inputs to the Primary and Secondary PID blocks.

2. Use the CV parameter connection on Primary PID to send output data to Secondary PID Remote Set Point (RSP).

3. Use the CV parameter connection on Secondary PID to send output to analog output channel.
4. Connect BCOUT/BCIN between CV2AO and Secondary PID to propagate initialisation data from Analog Output channel to the Secondary PID.
5. Connect BCOUT/BCIN between Secondary and Primary PID to propagate initialisation data from Secondary to Primary PID.

Operating modes and mode handling-PID Block

The PID block operates in the following modes:

- MAN (MANual) Mode = 0. OP may be set by the operator or other program logic; PV and SP are ignored - if a primary exists, it goes to the initialized state INITMAN
- AUTO (AUTOmatic) Mode = 1. SP may be set by the operator or other program logic. If a primary exists, it goes to the initialized state.
- CAS (CAScade) Mode = 2. If mode is CAScade, SP is pulled from a primary via RSP.

Ranges and limits-PID Block

- PV EUHI and EULO are contained in the PV Analog_Data type and are defined by either the Analog Input configuration or the HWDACA if PVCHAR is 1 (Linear) or 2 (Square Root).
  - EUHI and EULO define the full range of PV in engineering units.
  - EUHI represents the 100% of full scale value.
  - EULO represents the 0% of full scale value.
  - EUHI and EULO also define the engineering unit range of SP
  - PV and SP are assumed to have the same range.
- The PID block assumes PV is within EUHI and EULO - it applies no range check - however, PV typically comes from a data acquisition (DACA) block which applies its own limit and range check. If the PV goes outside of extended ranges EUHIEX and EULOEX as defined in analog channel, the PVSTS is set to initiate Bad Control options.

- SPHILM and SPLOLM define set point operating limits in engineering units. The operator is prevented from storing a set point value that is outside these limits; if the primary or a user program attempts to store a value outside of the limits, the PID block clamps it to the appropriate limit and sets the primary's windup status.
- OPHILM and OPLOLM define control output operating limits in percentage.
  - The PID algorithm clamps the calculated control variable to these limits and sets the primary's windup status if limits are reached.
  - The maximum control output range is assumed to be 0-100%.
  - An operator can set a value outside of OPHILM and OPLOLM limits when in MANUAL mode but only within the 0-100 % range.

Direct or reverse control-PID Block

A PID block may be configured for direct-control action or reverse-control action. Changing the control action effectively changes the sign of the gain.

- With direct-control action, an increase in the error (PV - SP) increases the PID output (CV).
- With reverse-control action, an increase in the error (PV - SP) decreases the PID output (CV).

For example, if SP = 50% and PV = 51%, then the error is 1%.

- With direct-control action, if PVP changes to 52%, the error increases causing CV to increase.
- With reverse-control action, if PVP changes to 52%, the error increases causing CV to decrease.

PID equations

The PID block provides four different equations for calculating the PID - the CTLEQN parameter is used to specify the desired equation.

- Equation A (CTLEQN = 0) - all three terms (Proportional, Integral, Derivative) act on the error (PV - SP) as follows:

$$\mathrm{CV} = \mathrm{K} * L^{-1}\left[\left(1 + \frac{1}{T1_S} + \frac{T2_S}{1 + a * T2_S}\right) * (PVP_S - SPP_S)\right]$$

- Equation B (CTLEQN = 1) - the proportional and integral terms act on the error (PV - SP) and the derivative term acts on changes in PV as follows:

$$\mathrm{CV} = \mathrm{K} * L^{-1}\left[\left(1 + \frac{1}{T1_S} + \frac{T2_S}{1 + a * T2_S}\right) * PVP_S - \left(1 + \frac{1}{T1_S}\right) * SPP_S\right]$$

- This equation is used to eliminate derivative spikes in the control action because of quick changes in SP.
- Equation C (CTLEQN = 2) - the integral term acts on the error (PV - SP) and the proportional and derivative terms act on changes in PV as follows:

$$\mathrm{CV} = \mathrm{K} * L^{-1}\left[\left(1 + \frac{1}{T1_S} + \frac{T2_S}{1 + a * T2_S}\right) * PVP_S - \left(\frac{1}{T1_S}\right) * SPP_S\right]$$

- This equation provides the smoothest and slowest response to SP changes.
- Equation D (CTLEQN = 3)- integral control only as follows:

$$\mathrm{CV} = L^{-1}\left[\frac{1}{T1_S} * (PVP_S - SPP)\right]$$

Where:

CV output of PID (Equations A, B, C, D) in percent

K gain (proportional term)

L^{-1} inverse of the LaPlace transform

PV process input value in engineering units

PVP PV in percent

a 1/16 fixed rate amplitude

s La Place operator

SP set point value in engineering units

SPP SP in percent

T1 integral time constant in minutes

T2 derivative time constant in minutes

To reduce frequent control variable changes to the final control element, the dead band setting (DB) can be used so that PID calculation is only performed if the error between PV and SP is greater than the DB setting defined in engineering units.

Tuning Constant Change Considerations

You cannot undo a change in OP due to a change in a tuning constant in an online control loop by simply changing the constant back to its original value. The output (OP) does not jump back to its original prior value just because you return the constant to its prior value. In this case, you must put the loop in MANUAL mode and set the output (OP) to the desired value before returning the loop to AUTO mode.

Rate of Change of Output

The maximum rate of change of the control output can be set by OPROCLM. The units are defined in %/Minute. To disable rate of change limiting, set the value to zero. Rate limiting is not applied when PID mode is Manual.

Windup handling-PID block

When a windup condition is reached, the PID block stops calculating the integral term, but continues to calculate the proportional and derivative term.

A windup condition exists if:

- PID block has a secondary and the secondary is in windup.
- PID block's output exceeds one of the user-specified output limits (OPHILM, OPLOLM).

Windup processing

The PID block maintains anti-reset windup status for its output (ARWHI and ARWLO) and each of its initializable inputs (ARWNETHI and ARWNETLO). The following table lists the possible values for ARWHI/LO and ARWNETHI/LO parameters.

If the Value is	Then, the Associated Parameter
False	is free to move in either direction
ARWHI or ARWNETHI is True	is at its high limit and may only be lowered
ARWLO or ARWNETLO is True	is at its low limit and may only be raised
Both HI and LO are True	may not move in either direction

Manual Mode Interaction

When the MODE of a PID block is changed to Manual (MAN), the block sets its windup status (ARWNETHI/LO) to True. This means that every block upstream in a cascade strategy will set its windup status (ARWNETHI/LO and ARWHI/LO) to True.

ARWHI/LO computation

The ARWHI/LO indicates if the output (OP) can be raised or lowered. The PID function blocks use ARWHI/LO to restrict integral control. When either ARWHI or ARWLO is true, the PID block stops integral control in the windup direction. Integral control continues in the other direction, as does proportional and derivative control. But, windup status has no impact on proportional and derivative control.

If a function block has a secondary, it fetches the secondary's windup status via BCOUT/BCIN connection and recomputes its ARWHI/LO. The conditions within the function block, such as output being at its high limit, also affect the ARWHI/LO. The ARWHI/LO is computed as follows, assuming the block has only one output or that it is not a FANOUT block.

If Any of the floowing are ture	Then, ARWHI/LO equals
A secondary exists and its windup state equals Hi and Lo	HI and LO = True
This block is in initialization (INITMAN = On).	
A secondary exists and it is requesting this block to initialize.	
A secondary exists and its windup state equals Hi.	HI = True
This block's output is at its high limit OPHILM	

If Any of the floowing are ture	Then, ARWHI/LO equals
A secondary exists and its windup state equals Lo.	LO = True
This block's output is at its low limit OPLOLM.	

ARWNETHI/LO computation

The only limiting anti-reset windup status ever does is to stop integral action in one or both directions on PID blocks. For any other regulatory control type block, ARWNETHI/LO is not used for any kind of limiting. The ARWNETHI/LO is computed as follows, assuming the block has only one output or that it is not a FANOUT block.

If Any of the floowing are ture	Then, ARWNET equals
The ARWHI and ARWLO are True	ARWNETHI = True, ARWNETLO = True
This block is in Manual mode (MODE = MAN).	
ARWHI equals True (Pid function blocks have a configurable Control Action option (CTLACTN). If CTLACTN = True (Reverse) , ARWNETHI/LO will track ARWHI/LO; but if CTLACTN = False (Direct) , ARWNETHI/LO will be the opposite of ARWHI/LO	ARWNETHI = True
The input from the primary is at a high SP limit SPHILM	
This block's output has reached its positive rate-of-change limit defined by OPROCLM	
ARWLO equals True (Pid function blocks have a configurable Control Action option (CTLACTN). If CTLACTN = True (Reverse) , ARWNETHI/LO will track ARWHI/LO; but if CTLACTN = False (Direct) , ARWNETHI/LO will be the opposite of ARWHI/LO).	ARWNETLO = True
The input from the primary is at a low SP limit SPLOLM	
This block's output has reached its negative rate-of-change limit defined by OPROCLM	

Override feedback processing-PID block

If the PID block is in a cascade strategy with a downstream OVRDSEL (Override Selector) block, it receives override feedback data. The data consists of an override status and override feedback value. The status ORFBSTS indicates if this block is in the selected or unselected strategy (as determined by the OVRDSEL block).

When the override status changes from False to True, the PID block initialises its CV value to OVRDVAL from secondary via BCOUT/BCIN connection before calculating P, I and D contributions to CV for current execution.

Bad Control Options

The BADCTL option determines how the PID block will behave if there is an error in the PV caused by any fault or configuration error in the Analog Input chain connected to the PID block. Bad control is invoked if

- The Analog Input encounters a critical error such as an open circuit detection. This can be diagnosed on the HWDACA block using the AISTS message.
- The PV value exceeds EUHIEX or EULOEX extended range
- The PV value is NaN
- The PV range is zero
- The OP range is zero

If the output BADCTRL is true, bad control processing occurs based on the BADCTL option values shown below.

0. (default) No Shed – CV will stop calculating and hold last valid value. Mode will remain unchanged.

1. Shed Hold – CV will stop calculating and hold last valid value and Mode will shed to Manual.

2. Shed Low – CV will be set to 0% and Mode will shed to Manual.

3. Shed High – CV will be set to 100% and Mode will shed to Manual.

4. Shed Safe OP – CV will be set to value defined by SAFEOP and Mode will shed to Manual.

Safety Interlock Options

The safety interlock option (SIOPT) determines how the PID block will behave if the Safety Interlock input (SI) is set to true.

The values of SIOPT are shown below.

0. (default) No Shed – CV will stop calculating and hold last valid value. Mode will remain unchanged.

1. Shed Hold – CV will stop calculating and hold last valid value and Mode will shed to Manual.

2. Shed Low – CV will be set to 0% and Mode will shed to Manual.

3. Shed High – CV will be set to 100% and Mode will shed to Manual.

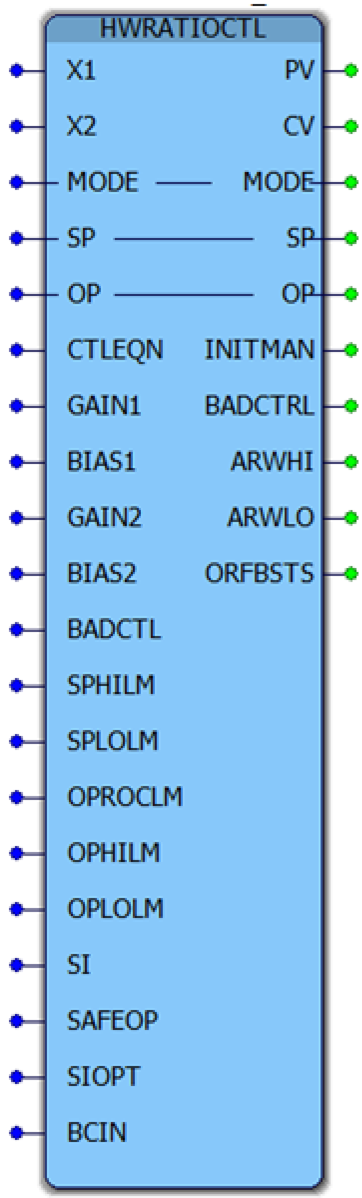
4. Shed Safe OP – CV will be set to value defined by SAFEOP and Mode will shed to Manual.

HWRATIOCTL

Description

This function block calculates a target flow rate to maintain a constant ratio between the controlled flow rate (X1) and an uncontrolled flow rate (X2). A typical application for this function block is continuous dosing control. The calculated flow rate is normally connected to a flow control PID as a cascade set point as shown below.

> **ATTENTION:** Versions prior to HWRATIOCTL V00.6 are limited to a controlled flow range of 0-100.



Input

Parameter	Data Type	Description
X1	Analog_Type	Controlled flow variable input. AI_Type contains value and quality flags
X2	Analog_Type	Uncontrolled flow variable input. AI_Type contains value and quality flags

Parameter	Data Type	Description
Mode	BOOL	Sets Mode. False - Manual True - Automatic
SP	REAL	Setpoint. From SCADA.
OP	REAL	Manual Output
CTLEQN	INT	Ratio Equation 0. PV = X1 / X2 1. PV = X2 / X1 2. PV = X1 / (X1 + X2) 3. PV = X2 / (X1 + X2)
GAIN1	REAL	Gain applied to X1
BIAS1	REAL	Bias applied to X1
GAIN2	REAL	Gain applied to X2
BIAS2	REAL	Bias applied to X2
BADCTL	INT	Bad Control Option as per C200/C300 0. No Shed 1. Shed Hold 2. Shed Low 3. Shed High 4. Shed Safe OP
SPHILM	REAL	Maximum Setpoint Limit in EU
SPLOLM	REAL	Minimum Setpoint Limit in EU
OPROCLM	REAL	Maximum rate of change of Control Output in %/min Default – 0 , no rate limiting
OPHILM	REAL	Maximum Output Default – 100%

Parameter	Data Type	Description
OPLOLM	REAL	Minimum Output Default – 0%
SI	BOOL	Safety Interlock. False – No shutdown (default) True – Shutdown using SIOPT
SAFEOP	REAL	Shutdown Control Variable Target value.
SIOPT	INT	Safety Option as per C200/C300 0. No Shed 1. Shed Hold 2. Shed Low 3. Shed High 4. Shed Safe OP
BCIN	BackCalc_Type	Back Calculation Input. This comes from Back Calculation Output of downstream block

Output

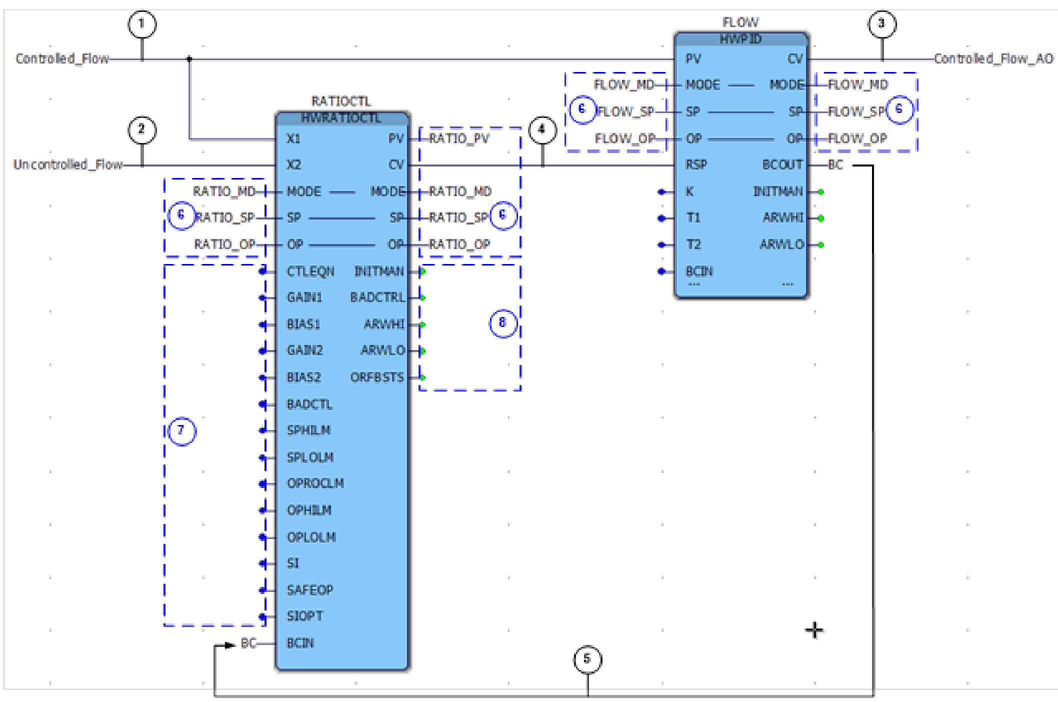
Parameter	Data Type	Description
PV	REAL	Calculated ratio after applying X1 and X2 gain and bias according to Equation selected.
CV	Analog_Type	Control Variable that is normally used to drive the analog output to a control device or as a remote SP to a PID block
INITMAN	BOOL	FB InitMan has been requested by downstream block
BADCTRL	BOOL	Bad Control Option is active
ARWHI	BOOL	FB is in high windup status
ARWLO	BOOL	FB is in low windup status
ORFBSTS	BOOL	FB is using Override Feedback value from

Parameter	Data Type	Description
		OVRSEL

Usage Notes

A typical application for this function block is continuous dosing control. The calculated flow rate is normally connected to a flow control PID as a cascade set point as shown below.

ATTENTION: HWRATIOCTL function blocks prior to V00.6 are limited to a controlled flow range of 0-100.



The following describes the main connections in the figure above.

1. The controlled flow input is a flow rate either from an analog input or from a flow calculation such as AGA for gas or API for hydrocarbon liquids. This is the process feedback from the flow we can control such as a dosing flow rate. This is connected to RATIOCTL (X1) and the flow control PID (PV).

2. The uncontrolled flow input is a flow rate either from an analog input or from a flow calculation such as AGA for gas or API for hydrocarbon liquids. This is the process indication from the flow we are adding controlled flow dosing to.
3. The flow control PID is connected to controlled flow control element via an analog output.
4. The calculated ratio control variable from the RATIOCTL function block is connected to the flow control PID remote set point for cascade mode.
5. The BCOUT of the PID is connected to the BCIN of the RATIOCTL function block to provide initialisation data for cascade connection.
6. The Mode, Set Point and Output are typically interfaced to analog SCADA points to provide SCADA monitoring and control of the RATIOCTL and flow controllers.
7. This group of parameters is used to configure the ratio control parameters. These are described below.
8. This group of outputs can be used to monitor the ratio control for windup and initialize status.

Modes of Operation

The ration control function block has two modes of operation.

- Manual Mode (MODE=False). In this mode, the controlled flow rate can be set manually via SCADA or logic. The ratio set point will be back calculated based on the entered OP and the current value of X2 to ensure bumpless transfer to Automatic mode.
- Automatic Mode (MODE=True). In this mode, the set point can be set via SCADA or logic. The OP is calculated for controlled flow rate using the selected equation, desired ratio set point and the uncontrolled flow rate.

Control Equation CTLEQN and Scaling

There are four different ratio control equations. The control equations are based on the scaled flows after the gain and bias are applied to the measured flows X1 and X2 such that.

X1_SCALED = X1 * GAIN1 + BIAS1

X2_SCALED = X2 * GAIN2 + BIAS2

The default values of GAIN1,2 are 1.0 and BIAS1,2 are 0.0. For each equation, there are two calculations. The ratio of the scaled measured flow values which is output as PV and the calculated value of controlled flow X1 to achieve the desired ratio set point SP. These are summarised below for the following values of CTLEQN

CTLEQN = 0

- PV = X1_SCALED / X2_SCALED
- CV = (X2_SCALED * SP – BIAS1) / GAIN1

CTLEQN = 1

- PV = X2_SCALED / X1_SCALED
- CV = (X2_SCALED / SP – BIAS1) / GAIN1

CTLEQN = 2

- PV = X1_SCALED / (X1_SCALED + X2_SCALED)
- CV = (SP * X2_SCALED + (SP – 1.0) * BIAS1) / ((1.0 – SP) * GAIN1)

CTLEQN = 3

- PV = X2_SCALED / (X1_SCALED + X2_SCALED)
- CV = (X2_SCALED / SP – X2_SCALED – BIAS1) / GAIN1

Should any of the denominator values become 0.0 for the PV equations, the value of PV will be set to NaN.

Should any of the denominator values become 0.0 for the CV equations, CV will not be updated and Bad Control processing will be invoked.

Set Point Limits

SPHILM and SPLOLM can be configured to limit the range of the entered ratio set point.

Rate of Change of Output and Output Limits

The maximum rate of change of the control output can be set by OPROCLM. The units are defined in %/Minute. This is useful to smooth bumps in the calculated output when changes are made to the set point as there is no time component in ratio calculation to stop step changes in output

To disable rate of change limiting, set the value to zero. Rate limiting is not applied when mode is Manual.

OPHILM and OPLOLM can be configured to limit the range of the calculated variable output in Auto mode. The output range is always a maximum of 0-100% in Manual mode.

Bad Control Options

The BADCTL option determines how the RATIOCTL block will behave if there is an error in X1 or X2 caused by any fault or configuration error in the Analog Input chain connected to the RATIOCTL block. Bad control is invoked if

- The X1 or X2 status flag is set by an upstream function block.
- The X1 or X2 value is NaN.

Bad control will also be invoked for the following configuration errors.

- The X1 Gain is set to zero which causes a divide by zero condition.
- The ratio SP is set to a value which causes a divide by zero condition

If the output BADCTRL is true, bad control processing is in effect based on the BADCTL option values shown below.

0. (default) No Shed – CV will stop calculating and hold last valid value. Mode will remain unchanged.

1. Shed Hold – CV will stop calculating and hold last valid value and Mode will shed to Manual.

2. Shed Low – CV will be set to 0% and Mode will shed to Manual.

3. Shed High – CV will be set to 100% and Mode will shed to Manual.

4. Shed Safe OP – CV will be set to value defined by SAFEOP and Mode will shed to Manual.

Safety Interlock Options

The safety interlock option (SIOPT) determines how the RATIOCTL block will behave if the Safety Interlock input (SI) is set to true.

The values of SIOPT are shown below.

0. (default) No Shed – CV will stop calculating and hold last valid value. Mode will remain unchanged.

1. Shed Hold – CV will stop calculating and hold last valid value and Mode will shed to Manual.

2. Shed Low – CV will be set to 0% and Mode will shed to Manual.

3. Shed High – CV will be set to 100% and Mode will shed to Manual.

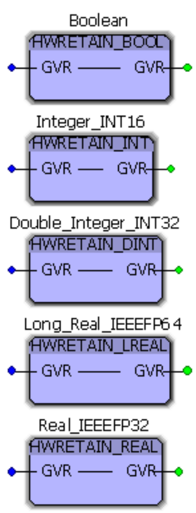
4. Shed Safe OP – CV will be set to value defined by SAFEOP and Mode will shed to Manual.

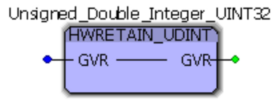
HWRETAIN

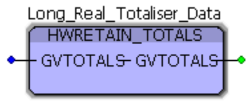
Description

ATTENTION: This function block requires the Global Variable attached to have an address defined. In R100, this was done automatically, however since R110 Global Variables no longer have an address assigned. The “RETAIN” check box on the Global Variable should be used instead of this function block.

This function block retains a global variable on a warm or cold start and after a reboot. This FB should be used for any user modified values such as tuning constants of PID or for accumulators on a totaliser FB. This function block requires that connected global variable is assigned an address.







In addition, a retain function block that handles the Totaliser_Data_LREAL Type is included. This allows simpler setup of totaliser function block.

A conversion FB to convert totaliser data from LREAL to REAL is included to enable easy conversion to data type suitable for SCADA mapping.

Input and Output

Parameter	Data Type	Description
GVR	By FB	Global variable to be retained

HWSDV

The HWFBLib contains a group of related device control function blocks for digital control of valves and motors as shown below.

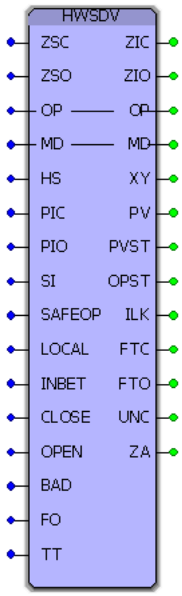
- HWSDV – Control of solenoid operated values such as shutdown valves
- HWMOV – Control of motor operated valves
- HWMCC – Control of motors
- HWMLV – Control of main line valves

Description

The HWSDV function block is applicable to solenoid operated valves such as shutdown valves. These devices are characterised by a latched digital output to an operating solenoid. The action of the de-energised solenoid is specified by the failure mode of the valve.

- FO (Fail Open) indicates that an energised solenoid will close valve and de-energising will open valve.

- FC (Fail Close) indicates that an energised solenoid will open valve and de-energising will close valve.
- FL (Fail Last) indicates that valve will fail in last commanded state. The HWSDV function block is not applicable for this application and either the HWMOV or HWMLV function blocks should be used.



Input

Parameter	Data Type	Description
ZSC	BOOL	Close Limit Switch Input from DI
ZSO	BOOL	Open Limit Switch Input from DI
OP	BOOL	Accepts command from SCADA when MD is in Manual. When MD is Auto, OP tracks HS. OP Command States False – Close True – Open
MD	BOOL	Mode control. False – Manual – OP Can be commanded from SCADA OP True – Auto – Commands come from HS input. OP tracks HS.

Parameter	Data Type	Description
HS	BOOL	Hand Switch command from logic to open or close valve. False – Close command True – Open command
PIC	BOOL	Close permissive. Must be true to permit close command. SI will override.
PIO	BOOL	Open permissive. Must be true to permit close command. SI will override.
SI	BOOL	Safety override interlock enforced if True
SAFEOP	BOOL	Safety override interlock command. False – Close True - Open
LOCAL	BOOL	Local = True. When in local OP commands will track the valve state. OP commands will not be accepted from SCADA or HS regardless of MD. Normally LOCAL is a digital input from device.
INBET	INT	Range 0 – 3, Value of in between or travel state. This value is calculated by evaluating ZSC + 2 x ZSO when valve in this state
CLOSE	INT	Range 0 – 3, Value of Close state. This value is calculated by evaluating ZSC + 2 x ZSO when valve in this state
OPEN	INT	Range 0 – 3, Value of Open state. This value is calculated by evaluating ZSC + 2 x ZSO when valve in this state
BAD	INT	Range 0 – 3, Value of bad or inconsitent state. This value is calculated by evaluating ZSC + 2 x ZSO when valve in this state. This will usually be the remaining state after INBET, OPEN and CLOSE states have been determined.
FO	BOOL	Failure mode. • If FO = False, this is a Fail Close valve meaning that output XY is energised to Open, de-energised to Close. • If FO = True, this is a Fail Open valve meaning that output XY is energised to Close, de-energised to Open.

Parameter	Data Type	Description
TT	TIME	Maximum travel time to open or close valve. This is used for fail to open and fail to close alarms.

Output

Parameter	Data Type	Description
ZIC, ZIO	BOOL	Normalised Close and Open limits with the following truth table ZIC ZIO State F F Inbet or Travel T F Closed F T Open T T Bad Note SCADA can use these for PV for consistency across all valves or can address ZSC and ZSO based on implementers preferences.
XY	BOOL	Output command to DO to control solenoid valve based on FO value. See FO parameter
PV	INT	Valve state as an integer 0 – Travel 1 – Closed 2 – Open 3 - Bad
PVST	STRING	Description of valve state used for monitoring in IEC Programming Workspace debug mode. Note that if PV = CfgErr then the settings of INBET, CLOSE, OPEN, BAD are inconsistent, that is values are outside of range 0 to 3 and/or there are duplicate values.
OPST	STRING	Description of valve output command used for monitoring in IEC Programming Workspace debug mode.

Parameter	Data Type	Description
ILK	BOOL	Interlock Override active
FTC	BOOL	Fail to close alarm raised if valve fails to close within TT. This alarm is inhibited when in LOCAL
FTO	BOOL	Fail to open alarm raised if valve fails to open within TT. This alarm is inhibited when in LOCAL
UNC	BOOL	Uncommanded change of state alarm if valve moves from commended state. This alarm is inhibited when in LOCAL
ZA	BOOL	Common alarm.

NOTE: Block can be used for single limit switch valve state indication by using only single limit connected to either ZSC or ZSO and configuring INBET, CLOSE, OPEN and BAD. For example, single limit switch indication

ZS – False Valve Open

ZS – True Valve Closed

Connect ZS to ZSC, leave ZSO unconnected. States will be

INBET 2 (don’t care)

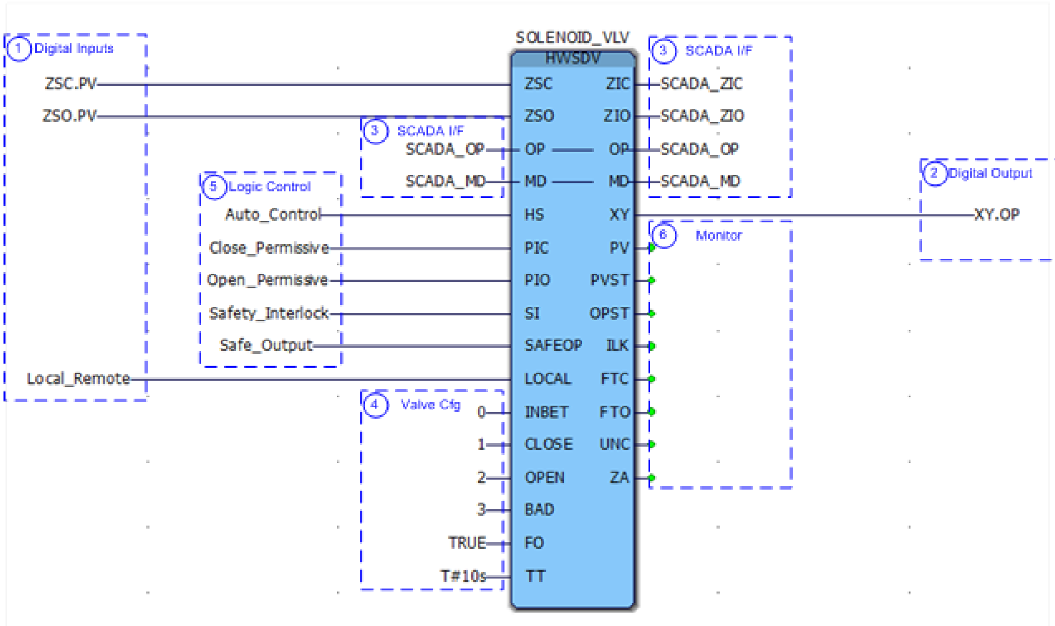
CLOSE 1

OPEN 0

BAD 3 (don’t care)

Implementation Example

A typical example is shown below with the main configuration areas highlighted.



1. Digital Inputs are connected to valve position feedback limit switches. Typically, there will be an open (ZSO) and a close (ZSC) limit switch. In some cases, only a single limit switch is provided. An optional input for Local can be used where a Hand/Off/Auto switch is used.
2. The function block main control output XY is connected to a digital output to drive the valve operating solenoid.
3. The SCADA control interface for the function block is mapped to a status point where .
   a. ZIC and ZIO are used for PV indication of valve state
   b. OP is used for SCADA control of valve when MD (Mode) is manual. The OP states are Close (False) and Open (True).
   c. MD is used to control mode of function block. When MD is Manual (False), the OP is used to control valve operation. When MD is Auto (True), controller logic operates valve via the HS input and OP tracks HS.
4. This group of parameters determines how valve is configured.
   a. Limit mapping (INBET, CLOSE, OPEN, BAD) determine how limit switches (ZSC, ZSO) map to SCADA indication (ZIC, ZIO). Due to the different possible configurations of limit switch operation (Normally Open or Normally Closed) and orientations of operating cams (make/break at beginning or end of valve movement), this mapping allows a single point to

configure limit switch behavior without affecting downstream SCADA and logic functions should a there be difference individual valves. Please refer to limit mapping table section.

b. Fail Mode of valve FO determines operation of valve command XY as follows:

FO value	Description	XY = False CMD	XY = True CMD
False	Fail Close Valve	Close	Open
True	Fail Open Valve	Open	Close

c. Travel Time TT specifies time out period before a travel time alarm is generated. If no travel time alarm is required, leave this unconfigured.

5. These inputs are primarily driven by program logic to control valve.

a. HS controls valve operation when MD is Auto.

b. PIC and PIO are permissives which need to be True before a valve can be closed or opened. If these pins are not connected, permissives will be true by default.

c. SI and SAFEOP are used for safety interlock operation. If SI is true, the valve will be commanded to the SAFEOP state of Close (False) or Open (True).

6. These pins can be used for monitoring operation of function block.

a. PV is a numeric indicating state of valve where

- PV = 0 (Travel)
- PV = 1 (Close)
- PV = 2 (Open)
- PV = 3 (Bad)

b. PVST and OPST display descriptive state of valve position and command

c. Alarm indications

- ILK Interlock active
- FTC, FTO Fail to Close, Open if commanded state not met within travel timeout period.
- UNC Uncommanded alarm is active if valve state becomes different to commanded state
- ZA Common alarm

Limit Mapping

The limit mapping inputs INBET, CLOSE, OPEN and BAD provide a means to standardize valve indication ZIC, ZIO in SCADA and for all downstream logic operations. The standardized indication is based on positive logic as shown below.

ZIC	ZIO	Valve State
FALSE	FALSE	Travel (In-between)
TRUE	FALSE	Closed
FALSE	TRUE	Opened
TRUE	TRUE	Bad (Error)

The operation of the actual valve limit indications may vary from the above due to actual configuration of limits on valves. In many cases, these differences are discovered during commissioning. The limit table provides a single place to rationalize limits to above table so that any impacts to downstream configuration of SCADA and logic are not impacted during commissioning.

The mapping value for a state is calculated by the formula

ZSC + 2 x ZSO

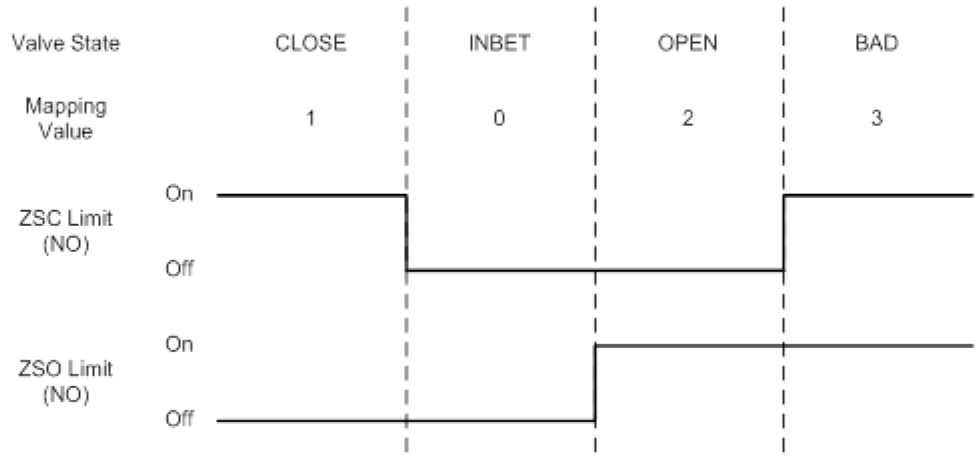
For example, if the valve is physically in the OPEN state and the open limit ZSO is On and the close limit ZSC is On then

OPEN = 1 + 2 x 1 = 3

Following are some examples typically encountered.

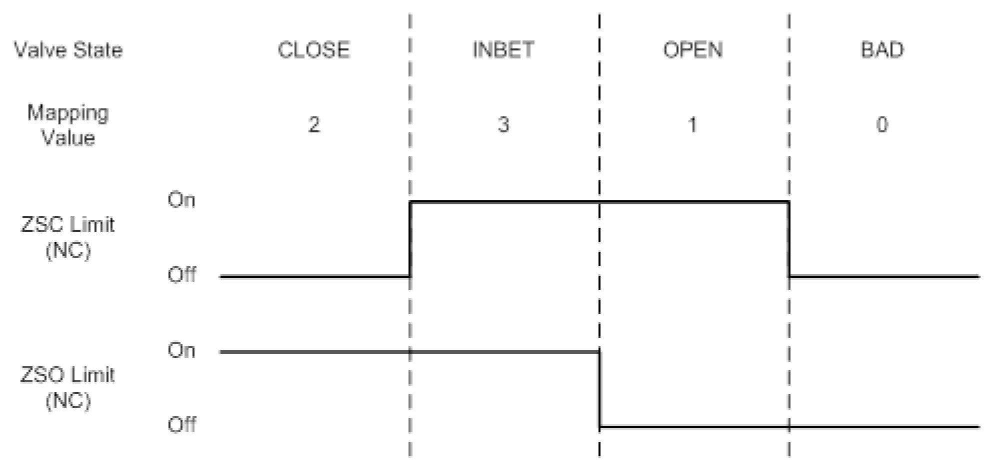
Valve Limit Switches Configured in Normally Open state.

This follows the ZIC, ZIO positive logic mapping. This is the default limit mapping of function block.



Valve Limit Switches Configured in Normally Closed state.

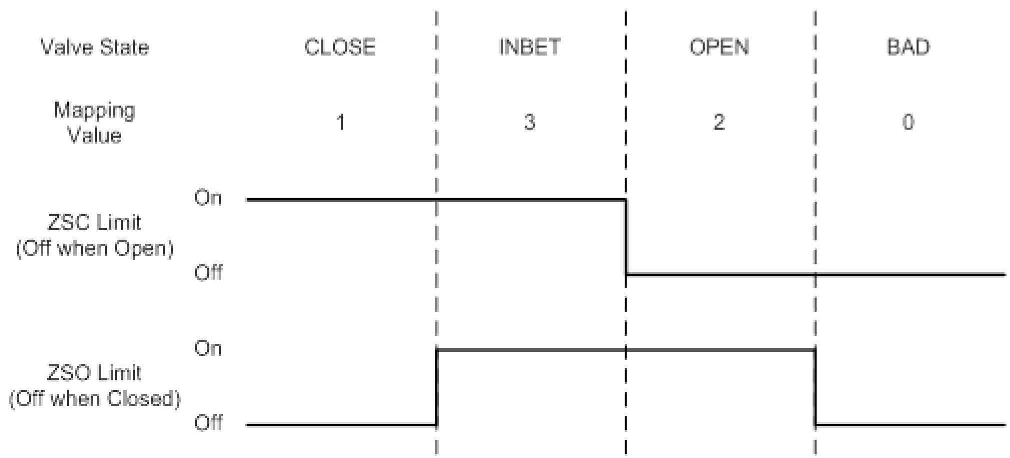
This arrangement is the reverse configuration (negative logic) and is sometimes used as it provides an error indication if field cables are cut (Both limits are off). The mapping values shown will convert ZIC and ZIO to follow positive logic.



Valve Limit Switches Configured in Complimentary Arrangement.

This arrangement has cams driving limit switches at end of movement so the ZSC limit is active unless valve fully opened and ZSO limit is active unless valve is fully closed. The advantage of this arrangement is that ZSO and ZSC use positive logic for Open and Close state but the BAD state is detected if ZSO and ZSC are both off (open circuit). This may be due to a failure of limit switches, links or fuses removed, field power lost or field cables damaged.

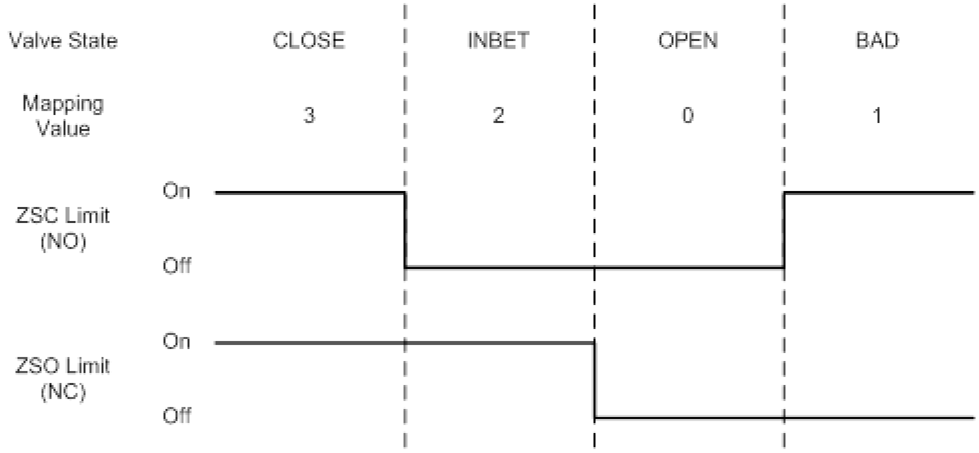
Below are the mapping values to be used to translate to the standard ZIC, ZIO



Valve Limit Switches Configured in Mixed State.

In this example, ZSC is configured as normally opened and ZSO is configured as normally closed. This is mainly an example to indicate flexibility to handle different arrangements that might arise.

Below are the mapping values to be used to translate to the standard ZIC, ZIO states.



Single Limit Switch

The function block can be used for single limit switch valve state indication by using only single limit connected to either ZSC or ZSO and configuring INBET, CLOSE, OPEN and BAD. For example, if there is only a single limit switch indication where

ZS – False Valve Open

ZS – True Valve Closed

Connect ZS to ZSC, leave ZSO unconnected. The states will be:

INBET 2 (don’t care)

CLOSE 1

OPEN 0

BAD 3 (don’t care)

Mode Operation

The function block has modes Manual (MD=False) and Auto (MD=True).

In Manual Mode, the output XY tracks OP set from SCADA.

In Auto Mode, the output XY tracks HS which is driven by program logic.

When in Auto, OP will track HS so that Mode change from Auto to Manual is bumpless.

Local/Remote

This input is normally connected to Hand/Off/Auto switch. When the LOCAL input is true, OP will track the PV state of valve and control of valve will be via a local control panel. While in LOCAL, commands will not be accepted from SCADA OP or Logic controlled HS regardless of MD setting.

Permissive and Safety Interlock.

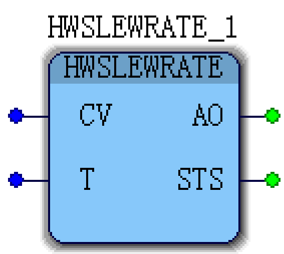
A permissive is available for Open and Close commands. If the respective permissive is not true, then that command cannot be executed. If the permissive becomes false after command is issued, the command is unaffected.

A safety interlock input (SI) of True will command the valve to the state set by SAFEOP (False = Close, Open = True). The safety interlock will take the highest precedence in Auto or Manual and will override a permissive.

HWSLEWRATE

Description

Slew Rate is the maximum rate of change required to drive the output from full OFF (0%-typically 0 mA or 4 mA) to full ON (100%-typically 20 mA). The block will convert this to a maximum change of the milliamp output per execution cycle of this block.



Input

Input	Data Type	Description
CV	Analog_Type	Control Analog data from Control Block such as HWPID or HWAUTOMAN

Input	Data Type	Description
T	Time	It is the maximum rate of change required to drive the output from full OFF (0%-typically 0 mA or 4 mA) to full ON (100%-typically 20 mA). Range from 0.00 to 99.00 s 0.00 indicates inactive state.

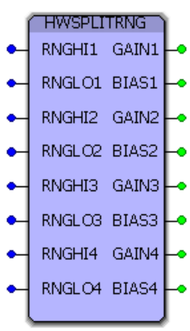
Output

Output	Data Type	Description
AO	REAL	Slew Rate result output
STS	BOOL	Analog output channel status True: With error False: Normal with no error

HWSPLITRNG

Description

The Split Range function block is used in conjunction with the FANOUT function block. This block translates split range settings to gain and bias settings suitable for FANOUT. Refer to help for the HWFANOUT function block for a typical application example.



Input

Parameter	Data type	Description
RNGHI1..4	REAL	Low range value of input to fanout that corresponds to 0% of corresponding OP of fanout.
RNGLO1..4	REAL	High range value of input to fanout that corresponds to 100% of corresponding OP of fanout.

Output

Parameter	Data type	Description
GAIN1..4	REAL	Connected to corresponding GAIN inputs of Fanout
BIAS1..4	REAL	Connected to corresponding BIAS inputs of Fanout

HWTOT_LREAL_TO_REAL

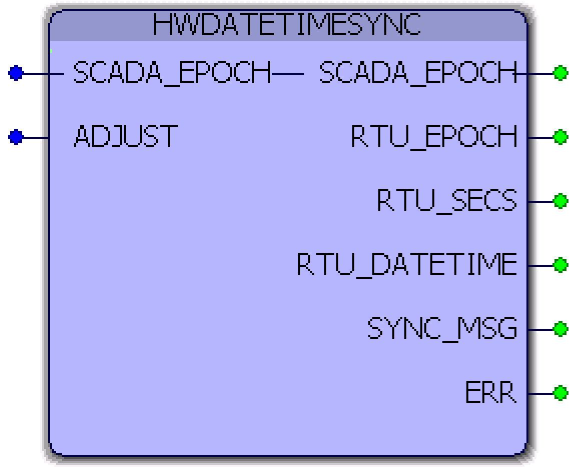
Description

This function block converts totaliser LREAL to Totaliser Real data.

HWDATETIMESYNC

Description

Provides ability to sync controllers Date/Time to SCADA when a SNTP time source is not available or accessible. For example, serial radio connected RTUs using Modbus RTU Protocol. It should be noted that this is not high precision time synchronisation. Typical accuracies will be with 1-2 seconds depending a communications network latency.



Input

Input Parameter	Data types	Description
SCADA_EPOCH	DINT	Epoch Time (secs since 1/1/1970 00:00:00) written from SCADA. If value is non-zero, value will be converted to set RTU Date and Time. This value will be reset to zero on completion ready for next time sync.
ADJUST	DINT	Optional value in seconds to add to SCADA_EPOCH time to allow for communications delay in SCADA connection.

Usage notes

The HWDATETIMESYNC FB requires Epoch time to be written from SCADA. For Honeywell Experion SCADA systems, this value can be read directly from the following System file using a Database Reference point parameter source address.

File	8
Record	1
Word	51
Format	INT4 (Signed 32 bit Integer)

This value can then be written via Modbus to the variable connected to the SCADA_EPOCH pin. For the point parameter used to write value to the RTU, you should only define the Destination address with no scan period or Source address. This prevents control fail alarms when value is zeroed by function block and prevents unnecessary scanning. Typically, the value might be written on a daily point schedule or on demand.

For other SCADA systems, refer to documentation to source EPOCH time.

Output

Output Parameter	Data types	Description
RTU_EPOCH	DINT	Current RTU Epoch time in seconds since 1/1/1970 00:00:00.
RTU_SECS	REAL	Current RTU seconds in day (secs since Midnight).
RTU_DATETIME	STRING	Current RTU Date/Time String.
SYNC_MSG	STRING	Date/Time of last Sync or Sync Error Message if unsuccessful.
ERR	BOOL	Date/Time Sync Error Flag.

Usage notes

To time sync, a push button or scheduled point control can be used to copy the Experion EPOCH time to the RTU.

The most reliable and efficient way to do a time sync over radio networks is as:

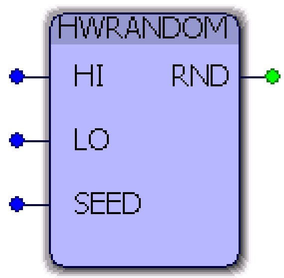
- Date/Time set with a single write of a 32-bit integer. Thus, no timing issues are encountered as opposed to writing Date/Time using multiple writes of each Date/Time field.
- The Point Parameter Write to the RTU must specify a destination address only with no source address or scan period. Thus, this doesn’t add any additional scanning load.

It should be noted that this technique is only suitable for time synching accuracy of a few seconds which is typically satisfactory for applications using serial radios.

HWRANDOM

Description

This function generates a uniformly distributed random number between user defined HI and LO value each execution cycle using a **linear congruential generator** algorithm. This function is useful for simulations and control development testing.



Input

Input Parameter	Data types	Description
HI, LO	REAL	Specifies the range over which the random number will be generated.
SEED	UINT	(OPTIONAL – PIN hidden by default) Algorithm SEED value. Specifies a starting value for algorithm so that the same sequence of random numbers is generated after a warm/cold start. If SEED is not used or set to ZERO, the SEED value will be taken from current value of GetMicroTickCount which will effectively result in a different sequence of random numbers after a restart.

Output

Output Parameter	Data types	Description
RND	REAL	A uniformly distributed random number between HI and LO values.

Usage notes

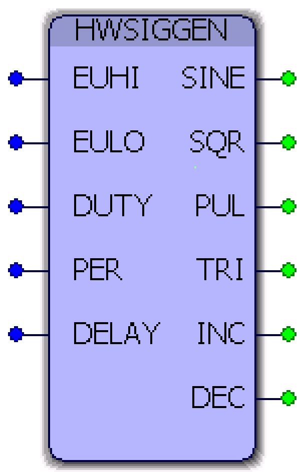
If the user want to generate a random integer, use a type conversion such as REAL_TO_INT and a range of Integer LO – 0.5 to Integer HI + 0.5. For example, to generate a dice roll of 1 to 6, a range of 0.5 to 6.5 should be used and then RND output is converted to an integer.

For more information about **Linear Congruential Generator** algorithm, See https://en.wikipedia.org/wiki/Linear_congruential_generator

HWSIGGEN

Description

Generates common wave forms based on controlles clock as time base. This function block is useful for generating changing values in simulations. By combining HWSIGGEN and HWRANDOM, it is possible to generate a wide range of periodic wave forms with random noise. You can also use multiple HWSIGGEN FBs and use the DELAY to phase shift waveforms.



Input

Input Parameter	Data types	Description
EUHI,EULO	REAL	Specifies the range over which the wave form will change.
DUTY	REAL	Fraction (0.0 to 1.0) or PER for Pulse = EUHI value. The remainder will be at EULO value.

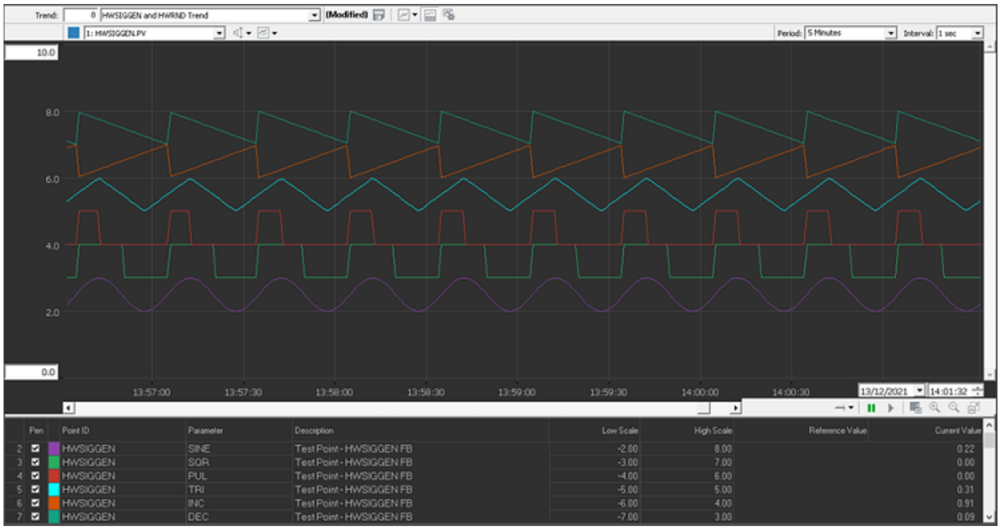
Input Parameter	Data types	Description
PER	REAL	Wave form cycle period in seconds. Value should be > 0.0. It is possible to set to less than one second.
DELAY	REAL	Wave form delay in seconds. Typically, this is only used if you want to have two or more HWSIGGEN FBs with a phase delay between wave form outputs.

Output

Output Parameter	Data types	Description
SINE	REAL	Sine Wave
SQR	REAL	Square Wave
PUL	REAL	Pulse Wave with duty cycle defined by DUTY.
TRI	REAL	Triangle Wave
INC	REAL	Increasing Ramp
DEC	REAL	Decreasing Ramp

Usage notes

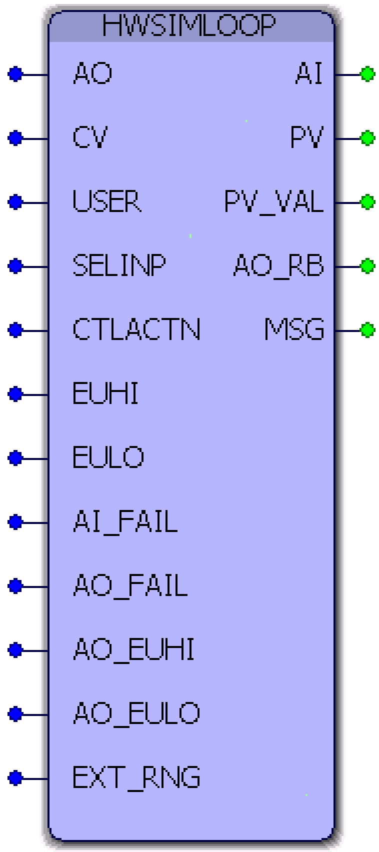
A Example of outputs for configuration with EULO = 0.0, EUHI=1.0, DUTY=0.25 and PER = 30 sec. Trend curves are offset to show each more clearly.



HWSIMLOOP

Description

This function block provides a means to Loop an Analog Output Channel back to an Analog Input Channel. This can be useful for testing process control strategies when I/O is not present or when using a virtual controller.



Input

Input Parameter	Data types	Description
AO	ANALOG_ OUTPUT_TYPE	Normally connected to final analog output channel of a control loop.
CV	ANALOG_TYPE	Normally connected to final CV of a control strategy.
USER	REAL	User define value for AI or PV outputs.
SELINP	INT	Select Input to use. 0 – AO 1 – CV 2 – USER
CTLACTN	BOOL	Control Action should match the control action

Input Parameter	Data types	Description
		(CTLACTN) of the connected HWPID block. When set to TRUE, an increase in AO or CV input will cause an increase in output AI or PV. When set to FALSE, an increase in AO or CV input will cause a decrease in output AI or PV.
EUHI	REAL	Set High Range of AI or PV output.
EULO	REAL	Set Low Range of AI or PV output.
AI_FAIL	BOOL	When True, simulates effect of disconnecting analog input to cause an OPEN WIRE fault.
AO_FAIL	BOOL	When True, simulates effect of disconnecting analog output to cause an OPEN WIRE fault.
AO_EUHI	REAL	Hidden by Default. This is intended for Future Use. Currently this should be left at default value of 100.0.
AO_EULO	REAL	Hidden by Default. This is intended for Future Use. Currently this should be left at default value of 0.0.
EXT_RNG	REAL	Hidden by Default. This specifies extended range outside EUHI-EULO as a percentage. The default is 10%.

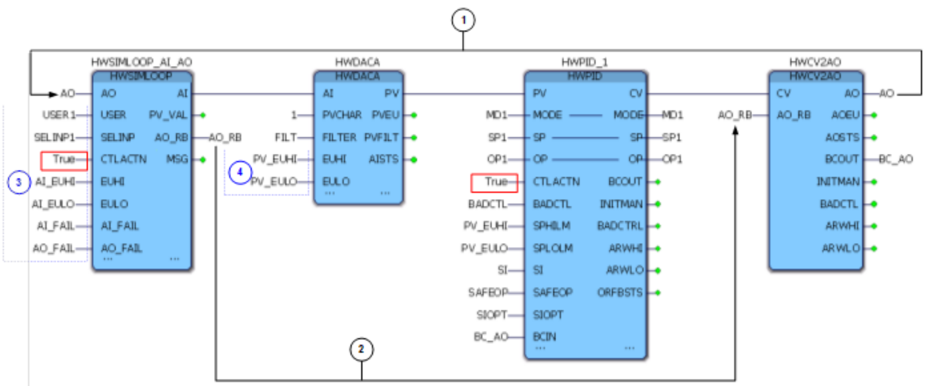
Output

Output Parameter	Data types	Description
AI	ANALOG_INPUT_TYPE	Normally connected to an analog input on a HWDACA FB.
PV	ANALOG_TYPE	Normally connected to a PV input on regulatory control FB such as HWPID.
PV_VAL	REAL	Current PV Value in AI or PV output. Typically used for monitoring only.
AO_RB	ANALOG_OUTPUT_READBACK_TYPE	Normally connected to AO_RB on a HWCV2AO FB.
MSG	STRING	Informational Message.

Usage notes

This function block can be used for testing control loops in a virtual RTU or a real RTU with no connected I/O. The loop back variable for testing/simulation only needs to be a local variable. Once testing is completed, the HWSIMLOOP FB can be removed, and real I/O variables are connected to control loop.

Below is an example for simulating from Analog Input to Analog Output.



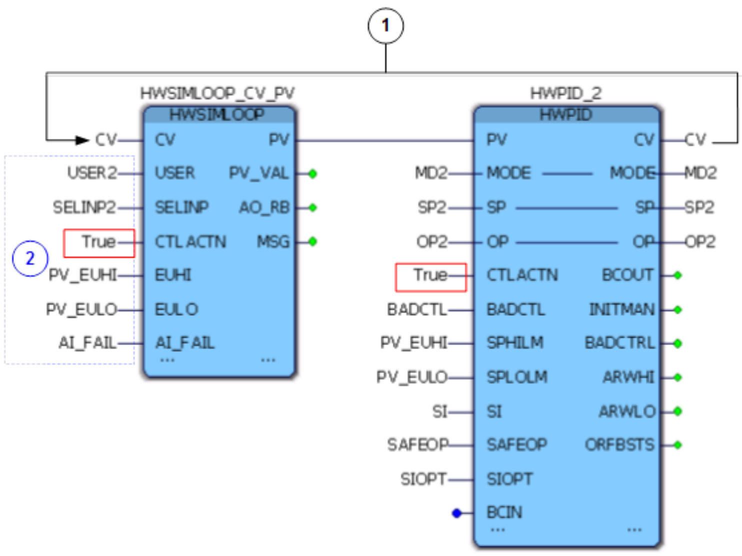
The following describes the main connections numbered in the figure above.

1. The AO ouput on HWCV2AO is connected to AO input on the HWSIMLOOP to complete a control loop. Note that you could just as easily connect HWSIMLOOP directly to the HWCV2AO via AO and then loop back AI to the HWDACA FB.
2. The AO_RB is connected between the HWSIMLOOP and HWCV2AO. This is mainly used for simulating analog output channel failures.
3. These are configuration settings for HWSIMLOOP. Note that for this configuration, the EUHI and EULO should correspond to HI and LO range configured in Analog Inputs.
4. These ranges will depend on PVCHAR option. In above example engineering units are ranged in the HWDACA FB which maybe different to Analog Input Range. For example, Analog Inputs are ranged generically as 0-100% and then HWDACA FB is used for engineering unit ranges for example 0-1000 kPa. In this example, AI_EULO = 0.0, AI_EUHI = 100.0, PV_EULO = 0.0 and PV_EUHI = 1000.0. If PVCHAR was set to 0, then only AI_EULO and AI_EUHI

are applicable as engineering unit range is configured in Analog Inputs.

> **ATTENTION:** The CTLACTN setting on HWLOOPSIM must be the same as CTLACTN on HWPID for correct operation of loop control.

Below is an example for simulating PV to CV loop. This is useful for quickly simulating a control loop for prototyping control strategies.



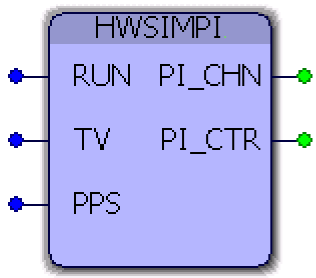
1. The CV output on HWPID is connected to CV input on the HWSIMLOOP to complete a control loop. Note that you could just as easily connect HWSIMLOOP directly to the HWPID via CV and then loop back PV to the HWPID FB.
2. These are configuration settings for HWSIMLOOP. Note that for this configuration, the EUHI and EULO should correspond to PV range.

> **ATTENTION:** The CTLACTN setting on HWLOOPSIM must be the same as CTLACTN on HWPID for correct operation of loop control.

HWSIMPI

Description

This function block simulates a pulse input channel. This can be used to simulate a pulse input device such as a turbine meter for development and testing of metering configuration.



Input

Input Parameter	Data types	Description
RUN	BOOL	When TRUE, pulses will be generated according to settings specified by TV and PPS inputs.
TV	UDINT	If TV=0, the function block will generate pulses continuously while RUN is True. If TV is set to a non-zero value, the function block will increment the counter by this number of pulse when RUN transitions from False to True.
PPS	UINT	Sets the pulse rate in Pulses per Second.

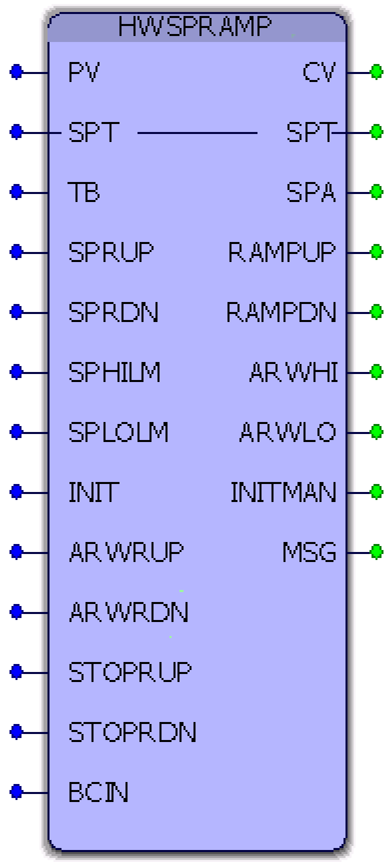
Output

Output Parameter	Data types	Description
PI_CHN	PULSE_ INPUT_TYPE	This can be connected to function blocks expecting a pulse input type such as HWPI or HWPIACC.
PI_CTR	UDINT	Pulse Counter Value.

HWSPRAMP

Description

This function block generates a ramping setpoint that can be connected to a HWPID as a cascade input. This leverages the bumpless mode transfer to ensure smooth transition and tracking between PID AUTO mode when setpoint is fixed and CASC when setpoint is controlled by HWSPRAMP.



Input

Input Parameter	Data types	Description
PV	ANALOG_TYPE	Connected to process variable.
SPT	REAL	Set point target to ramp to. Once ramp SP reaches SPT, ramping stops.
TB	INT	Time Base for SP Ramp Rate.

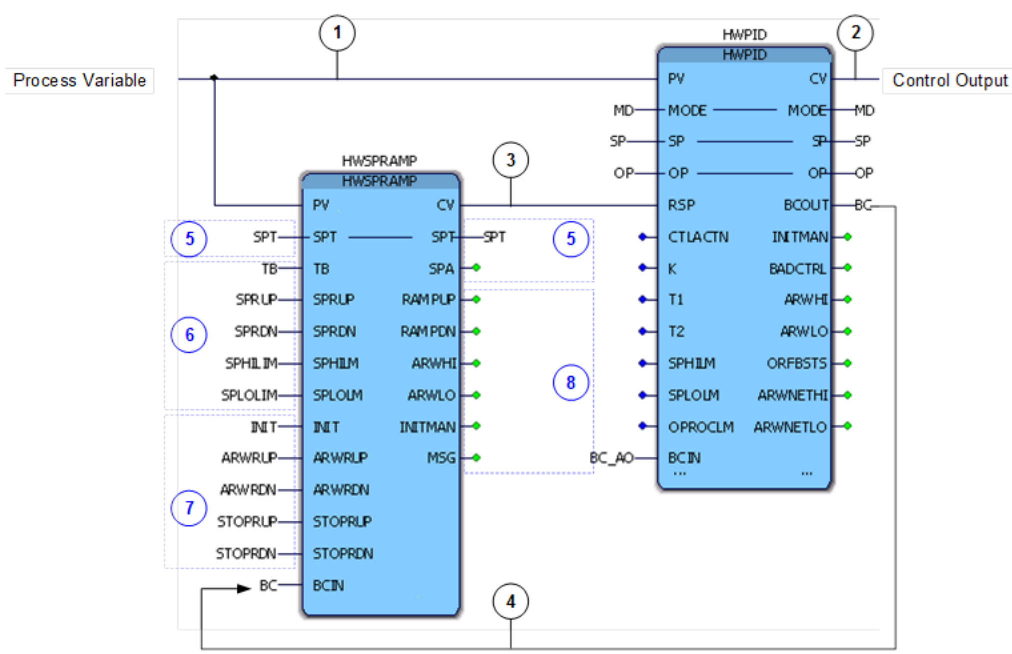
<table>
<tr><th>Input Parameter</th><th>Data types</th><th>Description</th></tr>
<tr><td></td><td></td><td><table><tr><td>0</td><td>Sec</td></tr><tr><td>1</td><td>Min</td></tr><tr><td>2</td><td>Hour</td></tr><tr><td>3</td><td>Day</td></tr></table></td></tr>
<tr><td>SPRUP</td><td>REAL</td><td>Setpoint Ramp Rate Up. Always a positive value. Setting to 0.0 disables ramp up, that is if SPT is increased SP will immediately track SPT.</td></tr>
<tr><td>SPRDN</td><td>REAL</td><td>Setpoint Ramp Rate Down. Always a positive value. Setting to 0.0 disables ramp down, that is if SPT is decreased SP will immediately track SPT.</td></tr>
<tr><td>SPHILM</td><td>REAL</td><td>Setpoint Target High Limit.</td></tr>
<tr><td>SPLOLM</td><td>REAL</td><td>Setpoint Target Low Limit.</td></tr>
<tr><td>INIT</td><td>BOOL</td><td>Enable Back Initialisation. (Optional - By default, this value is TRUE and hidden). In some applications, for example ramping a gap control setpoint, it maybe desirable to ignore back initialisation</td></tr>
<tr><td>ARWRUP</td><td>BOOL</td><td>Enable Anti Reset Windup of SP ramp up. (Optional - By default, this value is FALSE and hidden). When enabled, the set point ramp up rate will be limited by downstream windup status.</td></tr>
<tr><td>ARWRDN</td><td>BOOL</td><td>Enable Anti Reset Windup of SP ramp down. (Optional - By default, this value is FALSE and hidden). When enabled, the set point ramp rate down will be limited by downstream windup status.</td></tr>
<tr><td>STOPRUP</td><td>BOOL</td><td>Stop SP ramp up. (Optional - By default, this value is FALSE and hidden). When set to TRUE, ramp up will stop until set to false.</td></tr>
<tr><td>STOPRDN</td><td>BOOL</td><td>Stop SP ramp down. (Optional - By default, this value is FALSE and hidden). When set to TRUE, ramp down will stop until set to false.</td></tr>
<tr><td>BCIN</td><td>BACK_CALC</td><td>Back Calculation from downstream PID FB.</td></tr>
</table>

Output

Output Parameter	Data types	Description
CV	ANALOG_TYPE	Calculated SP, this is typically connected to downstream PID RSP pin.
SPA	REAL	Active Setpoint Value. Typically used for monitoring only.
RAMPUP	BOOL	Indicates when ramping up.
RAMPDN	BOOL	Indicates when ramping down.
ARWHI	BOOL	Indicates High Windup (if ARW set TRUE).
ARWLO	BOOL	Indicates Low Windup (if ARW set TRUE).
INITMAN	BOOL	Indicates InitMan active (if BINIT set to TRUE).
MSG	STRING	Informational Message of current ramping status.

Setpoint Ramp Example

The following figure shows a simple PID loop with setpoint ramping control.



The following describes the main connections in the figure above.

1. Process Variable is an Analog_Type which carries the PV value, PV status and PV range information to both function blocks.
2. The HWPID control output CV is normally connected to an analog output or another regulatory control function block.
3. The HWSPRAMP control output CV is connected to the HWPID RSP (Remote Setpoint) pin much the same as a cascade control strategy where the primary is the HWSPRAMP and secondary is the HWPID.
4. Use the BCOUT/BCIN connection to carry secondary data from the HWPID block to the HWSPRAMP block. The secondary data in the BACKCALC_TYPE data connection between HWPID BCOUT pin and the HWSPRAMP BCIN includes the following information.

   a. Anti-Reset Windup Status (ARWHI, ARWLO): Indicates if the secondary's initialize input (which is this block's output) is at its high or low limit.
   b. Initialization Request Flag (INITMAN): Used to request initialization. If the flag is set by CV2AO, the PID block initializes itself.
   c. Initialization Value (INITVAL): Used for initialization when INITMAN true.
   d. Cascade Flag: Indicates that secondary block has the Remote Set Point connected in a cascade strategy.
5. Setpoint Target (SPT) is the desired setpoint you wish to ramp to. This value is usually entered from SCADA. SPA is current active setpoint that is being sent to HWPID when HWPID is in Cascade mode. When the HWPID is in AUTO or MAN, SPA will track the HWPID SP to ensure bumpless transfer when SP Ramp is engaged using Cascade mode.
6. HWSPRAMP Configuration items
   a. TB determines ramp rate time base. For example, if you wish to enter SP ramp rate in SP_Units/Day, TB will be set to 3.
   b. SPRUP/SPRDN sets ramp up and ramp down rates using TB as rate time base. If either of these values is set to 0.0, then that ramp direction is disabled and output setpoint follows

entered setpoint target.

c. SPHILIM, SPLOLIM can be used to limit setpoint target value.

7. HWSPRAMP Configuration items (Optional – Hidden by default)

   a. Set INIT to true for INITMAN processing so that SP will track HWPID SP when HWPID Mode is in AUTO or MAN. There are some use cases where you may not want INITMAN processing. For example, if using HWSPRAMP to ramp control limits on a gap controller.

   b. ARWRUP/ARWRDN. Anti-Reset Windup. Enable these inputs if you want ramp rate to be limited by secondary (HWPID) windup status. These can be set independently for Ramp and Ramp Down. If set to false, the ramp rate will not be affected by windup status.

   c. STOPRUP/STOPRDN. These inputs can be used to stop ramping in either direction using external logic.

8. Flags to monitor ramping status.

CHAPTER

10 FUNCLIB

The following function blocks are available:

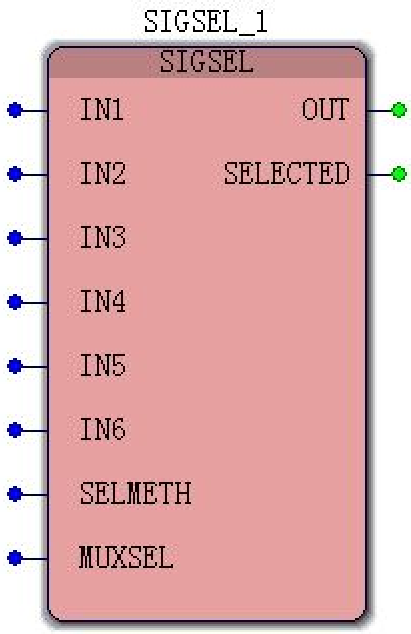
Function Blocks	Short Description
ANNUC	Accepts boolean inputs and shall provide one alarm output in case of abnormal input
GAINOFF	Provides linear characterization.
GENLIN	Provides a linearized PV (in engineering units) for a sensor with nonlinear characteristics and characterization option for Linear or Square Root conversion on the input, if required.
PULSE	Provides a maximum time limit pulse, minimum time limit pulse and fixed time limit pulse output each time when the input transitions from OFF to ON.
SIGSEL	Accepts as many as four input signals and select minimum value or maximum value or median input or calculate the average of the inputs or select an input based on the value of an external control signal.

You can still find information about structure variables at Structured Variables

SIGSEL

Description

The Signal Selector function block accepts as many as four input signals and shall be able to select minimum value or maximum value or median input or calculate the average of the inputs or select an input based on the value of an external control signal.



Input

Parameter	Data type	Description
IN1..IN6	REAL	Up to 6 Input values
SELMETH	INT	Selection Method. 0- Average (Default) 1- Minimum 2- Maximum 3- Median 4- MUX
MUXSEL	INT	Input to select when SELMETH = 4 (MUX)

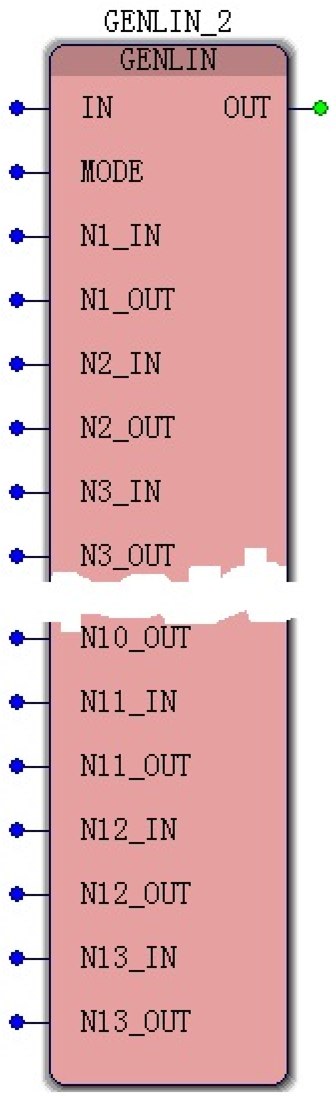
Output

Parameter	Data type	Description
OUT	REAL	Select value based on selection method.
SELECTED	INT	Input Selected (only valid for Max, Min and MUX modes)

GENLIN

Description

Provides a linearized PV (in engineering units) for a sensor with nonlinear characteristics. Shall provide characterization option for Linear or Square Root conversion on the input, if required.



Input

Parameter	Data type	Description
IN	REAL	Input value
MODE	INT	Sets conversion mode. Square Root (Default) Gen_Lin
N1_IN to N13_IN	REAL	Node 1 to 13 input value
N1_OUT to N13_OUT	REAL	Node 1 to 13 output value

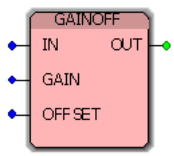
Output

Parameter	Data type	Description
OUT	REAL	Converted input value

GAINOFF

Description

Provides linear characterization.



Input

Parameter	Data type	Description
IN	REAL	Input value
GAIN	REAL	Gain value
OFFSET	REAL	Offset value

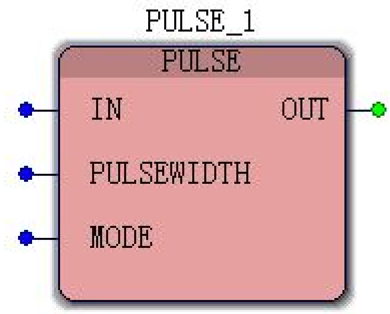
Output

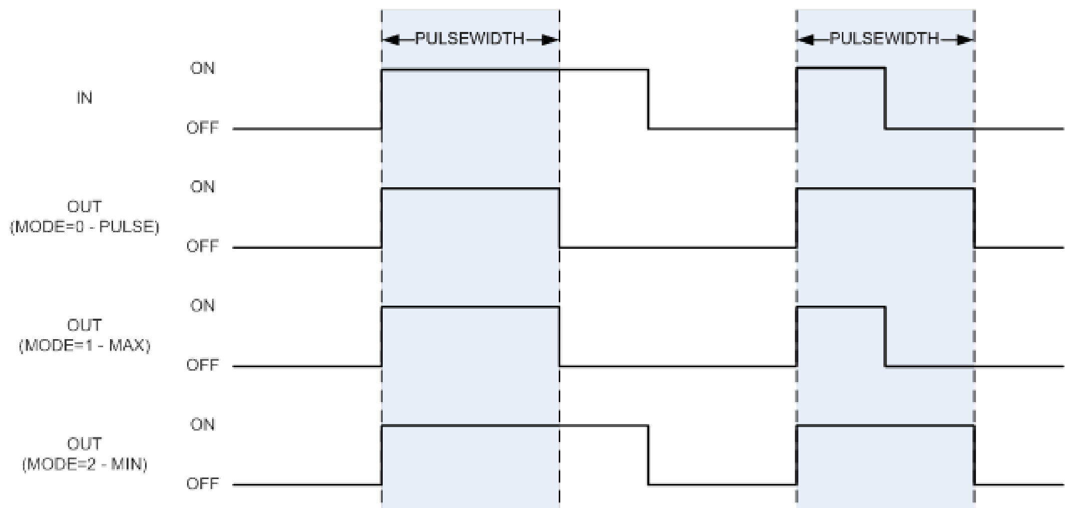
Parameter	Data type	Description
OUT	REAL	OUT = GAIN x IN + OFFSET

PULSE

Description

Provides a maximum time limit pulse, minimum time limit pulse and fixed time limit pulse output each time when the input transitions from OFF to ON.





Input

Parameter	Data type	Description
IN	BOOL	Logic Input
PULSEWIDTH	REAL	Pulse width in seconds
MODE	INT	Mode of pulse generation 0. PULSE (default) Generates a fixed pulse defined by PULSEWIDTH when a rising edge on IN occurs. 1. MAX –If the input (IN) pulse time is less than or equal to the specified PULSEWIDTH time, IN is assumed to equal one output (OUT) pulse. If the IN pulse time is greater than the specified PULSEWIDTH time, OUT pulse terminates at end of specified PULSEWIDTH time.

Parameter	Data type	Description
		2. MIN -If the input (IN) pulse time is less than or equal to the specified PULSEWIDTH time, output (OUT) pulse width equals the specified PULSEWIDTH time. If the IN pulse time is greater than the specified PULSEWIDTH time, OUT pulse width tracks IN pulse time, so OUT pulse exceeds specified PULSEWIDTH time.

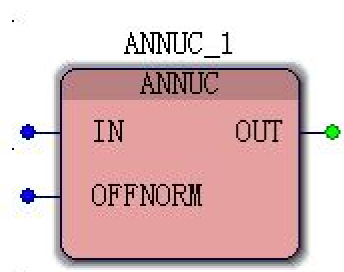
Output

Parameter	Data type	Description
OUT	BOOL	Output Pulse

ANNUC-Alarm Annunciator

Description

The Annunciator block function accepts Boolean inputs and shall provide one alarm output in case of abnormal input.



Input

Parameter	Data type	Description
IN	BOOL	Logic Input
OFFNORM	BOOL	Off Normal State

Output

Parameter	Data type	Description
OUT	BOOL	Set true if IN is equal to Off Normal state.

Structured Variables

This section details common structured variables.

Analog_Type

```
TYPE
ANALOG_TYPE :
STRUCT
    VAL      :   REAL; (* Value *)
    STS      :   BOOL; (* Out of Range*)
    EUHI     :   REAL; (* Range Hi *)
    EULO     :   REAL; (* Range Low *)
    EUHIEX   :   REAL; (* Range Hi Extended *)
    EULOEX   :   REAL; (* Range Low Extended *)
END_STRUCT;
END_TYPE
```

BackCalc_Type

```
TYPE
BACKCALC_TYPE   :
STRUCT
    INITMAN   :   BOOL; (* Initialise Manual Flag*)
    ORFBSTS   :   BOOL; (* Use Override FB Value *)
    BADCTL    :   BOOL; (* Bad Control Flag *)
    ARWHI     :   BOOL; (* Hi Windup Status *)
    ARWLO     :   BOOL; (* Lo Windup Status *)
    CASCADE   :   BOOL; (* Downstream Cascade Present *)
```

```
    INITVAL    :    REAL; (* Initialisation Value *)
    ORFBVAL    :    REAL; (* Override Feedback Value from OVRSEL *)
END_STRUCT;
END_TYPE
```

DI_Type (HOLD)

```
TYPE
 Analog_Type :
  STRUCT
    VALU     : REAL;    (* Value *)
   BAD       : BOOL;    (* Bad Value Flag *)
   EUHI     : REAL;    (* Range High *)
   EULO      : REAL;    (* Range Low *)
  END_STRUCT;
END_TYPE
```

DO_Type (HOLD)

```
TYPE
 Analog_Type :
  STRUCT
    VALU     : REAL;    (* Value *)
   BAD       : BOOL;    (* Bad Value Flag *)
   EUHI     : REAL;    (* Range High *)
   EULO      : REAL;    (* Range Low *)
  END_STRUCT;
END_TYPE
```

PI_Type (HOLD)

```
TYPE
 Analog_Type :
```

```
  STRUCT
     VALU      : REAL;     (* Value *)
    BAD        : BOOL;     (* Bad Value Flag *)
    EUHI      : REAL;     (* Range High *)
    EULO       : REAL;     (* Range Low *)
  END_STRUCT;
END_TYPE
```

CHAPTER

11 HART

From R150, two libraries of HART Function Blocks are supported:

Library	Releases applied
HART	RTU2020 R101, R110, R111, and ControlEdge RTU 140, R150
HART_V2	ControlEdge PLC R150 and ControlEdge RTU R150

The following HART function blocks are available:

Function Blocks	Short Description
HART_CMD3	Read dynamic variables.
HART_CMD48	Read additional device status.
HART CMDx	The HART CMDx function block supports all HART commands.

HART_CMD3

Description

Reads up to four predefined Dynamic Variables.

The Response Data is truncated after the last Dynamic Variables supported by each Device Type. For a given Device Type the number of Response Data bytes must be fixed. In other words, a Device type may not return PV, SV and TV in one operating mode and later(in a different operating) only return PV and SV.

Input

Parameter	Data type	Description
ENABLE	BOOL	Enable: If TRUE, the HART function block is enabled and workable.
RACK	USINT	Rack number: 0: local rack; 1~255: remote rack.

Parameter	Data type	Description
		TIP: This pin is only required for ControlEdge 900 Platform, and it is not applicable for ControlEdge 2020 Platform.
IOM	USINT	I/O module number: 0: built in I/O; 1~255: remote I/O
CHN	USINT	Channel Number 1~255, currently the valid data is 1~10 for build in I/O,AI 1~8, AO 9~10
SEND_ FLAG	BOOL	If SEND_FLAG is true and RDY_FLAG is true, HART_CMD3 would send the request. RDY_FLAG is TRUE means last communication is finished. Before last communication is finished, even if SEND_FLAG is true the request won't be sent.

Output

Parameter	Data type	Description
RDY_FLAG	BOOL	True: last communication is finished. FB is ready for the next communication. False: command request is being sent or received.
PV	REAL	Primary variable
PV_UC	USINT	Unit code of PV
SV	REAL	Secondary variable
SV_UC	USINT	Unit code of SV
TV	REAL	Tertiary variable
TV_UC	USINT	Unit code of TV
QV	REAL	Quaternary variable
QV_UC	USINT	Unit code of QV
GEN_DEV_STATUS	Array [1..8] of BOOL (user defined data type)	The output is valid If the PROTOCOL_ERR is less than 0x80 (it means the response message doesn't indicate a communication error). Bit 8: field device malfunction Bit 7: configuration has changed Bit 6: cold start(device has reset /power cycled) Bit 5: True: More status is available, command 48 can be sent to read the status. Bit 4: loop current fixed Bit 3: loop current saturated(PV out of limits) Bit 2: non-primary variable out of limits Bit 1: primary variable out of limits **TIP:** True (Logical –1) at a particular bit position indicates the described condition exists. Off is normal no error.

<table>
<tr><th>Parameter</th><th>Data type</th><th>Description</th></tr>
<tr><td>DONE</td><td>BOOL</td><td>It indicates that the response data is received successfully and usable.</td></tr>
<tr><td>ERR_FLAG</td><td>BOOL</td><td>It would be set true if there is an error</td></tr>
<tr><td>PROTOCOL_ERR</td><td>USINT</td><td>the response code received from HART device:
Bit 8 = true:
Bit 7 Parity error
Bit 6 Overrun error
Bit 5 Framing error
Bit 4 Checksum error
Bit 3 Always 0 (reserved)
Bit 2 buffer overflow
Bit 1 Always 0 (undefined)
TIP: True (Logical –1) at a particular bit position indicates the described condition exists. Off is normal no error.
Bit 8 = false:
0= No command-specific error
1= (undefined)
2= Invalid selection
3= Passed parameter to large
4= Passed parameter to small
5= Too few data bytes received
6= Device-specific command error
7= In write-protect mode
8-15= Command Specific (see command)
16= Access restricted</td></tr>
</table>

Parameter	Data type	Description
		17-127= Command Specific (see command) 32= Device is busy 64= Command not implemented
GEN_ERR	USINT	0: success 1= the input parameter given to the function block is invalid 2 = response timeout 3= internal error. IPC timeout or no response is received from HART server within a period of time(5 seconds) 17 = invalid I/O card (the I/O card is not configured in the system, or none of the I/O channels of this I/O card is HART-enabled) 18 = invalid I/O channel, the channel is HART-disabled or not exists 19 = device is offline. 20 = invalid I/O rack (the I/O rack is not configured in the system, or none of the I/O channels on the I/O cards of this I/O rack is HART-enabled)

HART_CMD48

Description

This command must be implemented by all HART devices.

Returns device status information not included in the Response Code or Device Status Byte. Peform Self Test. Responses Bytes 0-5 and 14-24 may contain Device-Specific Status information. Extended Device Status, Device Operationg Mode and Standarized Status 0-3 contain commonly used status information.

In addition, this command contains status information regarding Analog Channel 1 through Analog Channel 8. Bits in Analog Channel Saturated are set when the electrical limits established by the Field Device are exceeded for the corresponding Analog Channel. Bits in Analog Channel Fixed are set when the corresponding Analog Channel is directly or indirectly being manually controlled. In both of these data items the least Significant Bit (i.e.,Bit 0) refers to the Analog Channel (i.e. the Secondary Variable) and the Most Significant Bit refers to the 8th Analog Channel (if available in the Field Device).

Input

Parameter	Data type	Description
EN	BOOL	Enable: If TRUE, the HART FB is enabled and workable.
RACK	USINT	Rack number: 0: local rack; 1~255: remote rack. **TIP:** This pin is only required for ControlEdge PLC, and it is not applicable for ControlEdge RTU..
IOM	USINT	I/O module number: 0: built in I/O; 1~255: remote I/O
CHN	USINT	Channel Number 1~255, currently the valid data is 1~10 for build in I/O, AI 1~8, AO 9~10
SEND_ FLAG	BOOL	If SEND_FLAG is true and RDY_FLAG is true, HART_CMD48 would send the request. RDY_FLAG is TRUE means last communication is finished. Before last communication is finished, even if SEND_FLAG is true the request won't be sent.

Output

Command 48 response data can be maximum 25 bytes. Each byte is bitwise meaningful. To make it easy to access these bits, user-defined data type—ARRAY [1..n] of BOOL is used instead of Byte type.

To save the number of output pins a user defined data structure is created as follows:

TYPE

BIT8: ARRAY [1..8] of BOOL;

```
BIT48:   ARRAY [1..48] of BOOL;
BIT88:   ARRAY [1..88] of BOOL;
HART_CMD48_DEV_INFO:
            STRUCT
DEV_SPEC_STATUS_0: BIT48;
EXT_DEV_STATUS: BIT8;
DEV_OPER_MODE: BIT8;
STD_STATUS_0: BIT8;
STD_STATUS_1: BIT8;
ANALOG_CHN_SATURATED: BIT8;
STD_STATUS_2: BIT8;
STD_STATUS_3: BIT8;
ANALOG_CHN_FIXED: BIT8;
DEV_SPEC_STATUS_1: BIT88;
            END_STRUCT;
END_TYPE
```

The way to access a specific bit is to use the suffix, e.g. the fifth bit of STD_STATUS_0 is obtained by using HART_CMD48_DEV_INFO. STD_ STATUS_0 [5].

TIP: The data structure would be provided by Honeywell and is not allowed to be modified by the end-user. Any modification might leads to the corruption of the controller. There are 80 more reserved bits, or in other words, 10 more reserved bytes defined in the structure of HART_CMD48_DEV_INFO. Because some types of devices have more response bytes than the latest HART specification. So 10 more bytes are defined to support potential long response data.

Parameter	Data type	Description
RDY_FLAG	BOOL	True: last communication is finished. FB is ready for the next communication.

Parameter	Data type	Description
		false: command request is being sent or received
HART_CMD48_ DEV_INFO.DEV_ SPEC_STATUS_0	Array [1..48] of BOOL	Device specific status (refer to appropriate device-specific document for detailed information)
HART_CMD48_ DEV_INFO.EXT_ DEV_STATUS	Array [1..8]of BOOL	Extended device status: **Code** | **Map** | **Description** 0x01: N maintenance required 0x02: S device variable Alert 0x04: F critical power failure 0x08: N failure 0x10: N out of specification 0x20: N function check
HART_CMD48_ DEV_INFO.DEV_ OPER_MODE	Array [1..8]of BOOL	Device operation mode (reserved)
HART_CMD48_ DEV_INFO.STD_ STATUS_0	Array [1..8]of BOOL	Standardized status 0: **Code** | **Map** | **Description** 0x01: C device variable simulation active 0x02: F non-volatile memory defect 0x04: F volatile memory defect 0x08: F watchdog reset executed 0x10: S power supply conditions out of range 0x20: S Environmental conditions out of range 0x40: F electronic defect 0x80: N device configuration locked
HART_CMD48_ DEV_INFO.STD_	Array [1..8]of	Standardized status 1: **Code** | **Map** | **Description**

Parameter	Data type	Description
STATUS_1	BOOL	0x01: N status simulation active 0x02: C discrete variable simulation active 0x04: N event notification overflow
HART_CMD48_ DEV_ INFO.ANALOG_ CHN_SATURATED	Array [1..8]of BOOL	Analog channel saturated: **Code | Map | Description** 0x01: S analog channel 1 0x02: S analog channel 2 0x04: S analog channel 3 0x08: S analog channel 4
HART_CMD48_ DEV_INFO.STD_ STATUS_2	Array [1..8]of BOOL	Standardized status 2: **Code | Map | Description** 0x01: N sub-device list changed 0x02: M duplicate master detected 0x04: M sub-device mismatch 0x08: N sub-device with duplicate IDs found 0x10 S stale data notice
HART_CMD48_ DEV_INFO.STD_ STATUS_3	Array [1..8]of BOOL	Standardized status 3: **Code | Map | Description** 0x01: M capacity denied 0x02: N reserved 0x04: N bandwidth allocation pending 0x08: N block transfer pending 0x10: F radio failure
HART_CMD48_ DEV_ INFO.ANALOG_	Array [1..8]of BOOL	Analog channel fixed: **Code | Map | Description**

Parameter	Data type	Description
CHN_FIXED		0x01: C analog channel 1 0x02: C analog channel 2 0x04: C analog channel 3 0x08: C analog channel 4
HART_CMD48_ DEV_INFO.DEV_ SPEC_STATUS_1	Array [1..88]of BOOL	Device specific status (refer to appropriate device-specific document for detailed information)
GEN_DEV_STATUS	Array [1..8] of BOOL	The output is valid If the PROTOCOL_ERR is less than 0x80(it means the response message doesn't indicate a communication error). Bit 8: field device malfunction Bit 7: configuration has changed Bit 6: cold start(device has reset /power cycled) Bit 5: True: More status is available, command 48 can be sent to read the status. Bit 4: loop current fixed Bit 3: loop current saturated(PV out of limits) Bit 2: non-primary variable out of limits Bit 1: primary variable out of limits **TIP:** True (Logical –1) at a particular bit position indicates the described condition exists. Off is normal no error.
DONE	BOOL	It Indicates that the response data is received successfully and usable
ERR_FLAG	BOOL	It would be set true if there is an error
PROTOCOL_ERR	USINT	the response code received from HART device:

Parameter	Data type	Description
		Bit 8 = true: Bit 7 Parity error Bit 6 Overrun error Bit 5 Framing error Bit 4 Checksum error Bit 3 Always 0 (reserved) Bit 2 buffer overflow Bit 1 Always 0 (undefined) **TIP:** True (Logical –1) at a particular bit position indicates the described condition exists. Off is normal no error. Bit 8 = false: 0= No command-specific error 1= (undefined) 2= Invalid selection 3= Passed parameter to large 4= Passed parameter to small 5= Too few data bytes received 6= Device-specific command error 7= In write-protect mode 8-15= Command Specific (see command) 16= Access restricted 17-127= Command Specific (see command) 32= Device is busy 64= Command not implemented

Parameter	Data type	Description
GEN_ERR	USINT	0: Communication succeeded. 1: The input parameter is invalid. 2: Response timeout. 3: RTU internal time out (IPC timeout), 17: invalid I/O card (the I/O card is not configured in the system, or none of the I/O channels of this I/O card is HART-enabled). 18: invalid I/O channel, the channel is HART-disabled or not exists 19: device is offline. 20 = invalid I/O rack (the I/O rack is not configured in the system, or none of the I/O channels on the I/O cards of this I/O rack is HART-enabled)

TIP: As the response data length is device dependent, those pins with no data received would be set to 0.

HART_CMDx

Description

It supports all HART commands with the command number no more than 255, except Command 6 and commands relevant to “Burst”. The end user needs to create a HART command request message for the pin “IN”. If a command response is received, it would be put in the pin “OUT” and the end user needs to parse it.

Input

Parameter	Data type	Description
ENABLE	BOOL	Enable: If TRUE, the HART FB is enabled and workable.

Parameter	Data type	Description
RACK	USINT	Rack number: 0: local rack; 1~255: remote rack.(This pin is only available for ControlEdge PLC, it should be not configured for RTU)
IOM	USINT	I/O module number: • For ControlEdge RTU: 0: built in I/O; 1~30: expansion I/O; • For ControlEdge PLC: 1~12
CHN	USINT	Channel number : 1~255; • For ControlEdge PLC UIO, currently the valid channel number is 1~16 for AI or AO • For ControlEdge RTU, currently the valid channel number is 1~10 with 1~8 as AI and 9~10 for AO.
SEND_ FLAG	BOOL	If SEND_FLAG is true and RDY_FLAG is true, the FB would send the request. RDY_FLAG is TRUE means last communication is finished. Before last communication is finished, even if SEND_FLAG is true the request won't be sent.
CMD	USINT	HART command
IN	HART_ CMDx_IN (ARRAY [1..255] of BYTE)	User provides the "data" segment of the frame. **TIP:** User should be responsible for the validity of the "data".
IN_SIZE	USINT	The number of bytes contained in the "IN" buffer, which is also the "Byte Count" segment of the frame.

Output

Parameter	Data type	Description
RDY_FLAG	BOOL	True: last communication is finished. FB is ready for the next communication. False: command request is being sent or received
DONE	BOOL	It Indicates that the response data is received successfully and usable

Parameter	Data type	Description
ERR_FLAG	BOOL	It would be set to True if there is an error.
PROTOCOL_ERR	USINT	The response code received from the HART device: Bit 8 = True: Bit 7 Parity error Bit 6 Overrun error Bit 5 Framing error Bit 4 Checksum error Bit 3 Always 0 (reserved) Bit 2 buffer overflow Bit 1 Always 0 (undefined) **TIP:** True (Logical –1) at a particular bit position indicates the described condition exists. Off is normal no error. Bit 8 = false: 0= No command-specific error 1= (undefined) 2= Invalid selection 3= Passed parameter to large 4= Passed parameter to small 5= Too few data bytes received 6= Device-specific command error 7= In write-protect mode 8-15= Command Specific (see command) 16= Access restricted 17-127= Command Specific (see command) 32= Device is busy

Parameter	Data type	Description
		64= Command not implemented
GEN_ERR	USINT	0: Communication succeeded. 1: The input parameter is invalid. 2: Response timeout. 3: Controller internal time out (IPC timeout). 17: Invalid I/O card (the I/O card is not configured in the system, or none of the I/O channels of this I/O card is HART-enabled). 18: Invalid I/O channel, the channel is HART-disabled or not exists. 19: Device is offline. 20 = invalid I/O rack (the I/O rack is not configured in the system, or none of the I/O channels on the I/O cards of this I/O rack is HART-enabled)
GEN_DEV_ STATUS	HART_ GEN_ DEV_ STATUS (Array [1..8] of BOOL)	The output is valid If the PROTOCOL_ERR is less than 0x80 (it means the response message doesn't indicate a communication error). Bit 8: Field device malfunction Bit 7: Configuration has changed Bit 6: cold start(device has reset /power cycled) Bit 5: True: More status is available, command 48 can be sent to read the status. Bit 4: Loop current fixed Bit 3: Loop current saturated(PV out of limits) Bit 2: Non-primary variable out of limits Bit 1: Primary variable out of limits **TIP:** True(Logical –1) at a particular bit position indicates the described condition exists. Off is normal no error.

Parameter	Data type	Description
OUT	HART_ CMDx_ OUT (ARRAY [1..255] of BYTE)	The “data” segment of the frame returned by the device except for the first two bytes. The data segment received from HART device is broken up into three parts: PROTOCOL_ERR (the first byte of the data segment), GEN_DEV_STATUS (the second byte of the data segment) and OUT (the rest bytes of the data segment). **TIP:** User should be responsible for parsing the “data”. And the function block doesn’t know what it contains.
OUT_SIZE	USINT	The number of bytes contained in the “OUT” buffer

TIP: As the response data length is device dependent, those pins with no data received would be set to 0.

To save the data of Pin “IN” and Pin “OUT”, user defined data structures are created as follows:

TYPE

HART_CMDx_IN: ARRAY[1..255] of USINT;

END_TYPE

TYPE

HART_CMDx_OUT: ARRAY[1..255] of USINT;

END_TYPE

Example

Use HART_CMDx and assign CMDx as Command 1 to read the Primary Variable from HART device.

See the following table for Hart Command 1 Specification:

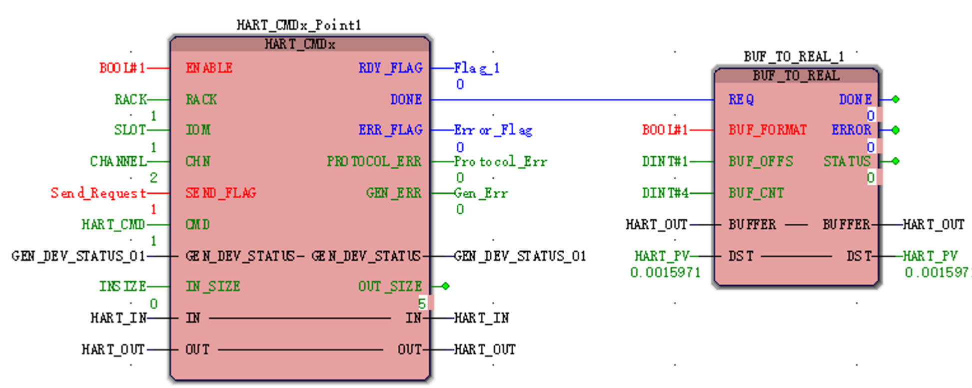
Request Data Byte-Input:

Input Data Bytes	Format	Description
None	NA	NA

Request Data Byte-Output:

Output Data Bytes	Format	Description
0	Enum	Primary Variable Units
1-4	Float	Primary Variable

1. In HART_CMDx, specify the rack address to RACK, slot number of the module to IOM, and the channel number to CHN for the connected HART device.
2. Specify the HART command (CMD) which the FB is used. In this example, assign Command 1.
3. Assign the number of bytes of the request data to IN_SIZE, which is 0 in this example, and the number of bytes of the response data will be shown in OUT_SIZE after the response from the device is received.
4. Assign variables with specific data types to GEN_DEV_STATUS, IN and OUT, which is mandatory. Then apply the Request Data Bytes to the variable assigned to the IN. It is None for Request Data Bytes in this example, so no need to apply.
5. Apply BUF_TO_REAL function block to split message and obtian Primary Variable: Assign BUF_OFFS for byte address in the buffer as DINT#1; Assign BUF_CNT for number of bytes to be coppied for the Primary Variable as DINT#4-float; assign the same variable for OUT of HART CMDx to BUFFER.
6. Connect REQ of BUF_TO_REAL to DONE of HART_CMDx.
7. Make ENABLE of HART_CMDx as True to read Primary Variable from HART Device.



And DST displays Primary Variable.

CHAPTER

12 UNITCONVERSIONLIB

The following Unit Conversion function blocks are available.

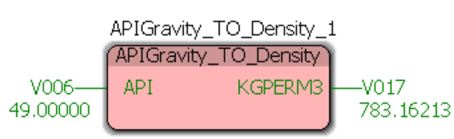
Function Blocks	Short Description
FAHRENHEIT_TO_KELVIN	FAHRENHEIT_TO_KELVIN function block converts temperature from Fahrenheit to Kelvin.
CELCIUS_TO_KELVIN	CELCIUS_TO_KELVIN function block converts temperature from Celcius to Kelvin.
FAHRENHEIT_TO_RANKINE	FAHRENHEIT_TO_RANKINE function block converts temperature from Fahrenheit to Rankine.
CELCIUS_TO_RANKINE	CELCIUS_TO_RANKINE function block converts temperature Celcius to Rankine.
PSIA_TO_MPA	PSIA_TO_MPA function block converts pressure from Psia to Mpa (mega pascal).
BAR_TO_MPA	BAR_TO_MPA function block converts pressure from Bar to Mpa.
BAR_TO_PSIA	BAR_TO_PSIA function block converts pressure from Bar to Psia.
INH2O_TO_MPA	INH2O_TO_MPA function block converts differential pressure from INH2O (inches of water) to Mpa.
MILIBAR_TO_MPA	MILIBAR_TO_MPA function block converts differential pressure from Milibar to Mpa.
MILIBAR_TO_INH2O	MILIBAR_TO_INH2O function block converts differential pressure from Milibar to INH2O.
HEATING_VALUE_US_TO_SI	HEATING_VALUE_US_TO_SI function block converts gas heating value from US unit (Btu/ft^3) to SI unit (MJ/m^3).
DENSITY_SI_TO_US	DENSITY_SI_TO_US function block converts density from SI unit (KG/M^3) to US unit (LBM/FT^3).
DIAMETER_MM_TO_INCHE	DIAMETER_MM_TO_INCHE function block converts diameter from millimeter to inches.
FLOWRATE_US_TO_METRIC	FLOWRATE_US_TO_METRIC function block converts flow rate from US unit system to Metric unit system.
MASS_FLORATE_US_TO_MET	MASS_FLORATE_US_TO_MET function block converts mass flow rate from US unit system to Metric unit system.

Function Blocks	Short Description
VISCO_US_TO_CENTIPOISE	VISCO_US_TO_CENTIPOISE function block converts viscosity from US unit to Centipoise.
CELCIUS_TO_FAHRENHEIT	CELCIUS_TO_FAHRENHEIT function block converts temperature from Celcius to Fahrenheit.
KPA_TO_PSIG	KPA_TO_PSIG function block converts pressure from Kpa (Kilo Pascal) to PSIG.
BAR_TO_PSIG	BAR_TO_PSIG function block converts pressure from Bar to PSIG.
APIGravity_TO_Density	APIGravity_TO_Density function block converts density from API Gravity to Density (KG/M^3).
Density_TO_APIGravity	Density_TO_APIGravity function block converts density from Density (KG/M^3) to API Gravity.
THERMAL_EXPAN_CEL_TO_FEH	THERMAL_EXPAN_CEL_TO_FEH function block converts thermal expansion from Celcius to Fahrenheit.
THERMAL_EXPAN_FAH_TO_CEL	THERMAL_EXPAN_CEL_TO_FEH function block converts thermal expansion from Fahrenheit to Celcius.
RELATIVE_DENSITY_TO_KGPE	RELATIVE_DENSITY_TO_KGPE function block converts relative density to density (KG/M^3).
KGPERM_TO_REL_DENSITY	KGPERM_TO_REL_DENSITY function block converts density to relative density.

APIGravity_TO_Density

Description

APIGravity_TO_Density function block converts API gravity to Density of liquid.



Input

Input Parameter	Data types	Description
API	LREAL	Input value of API Gravity

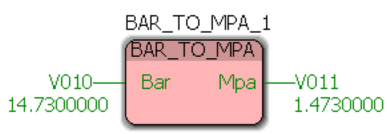
Output

Output Parameter	Data types	Description
KGPERM3	LREAL	Output value of density. It is in Kg/cubic meter.

BAR_TO_MPA

Description

AR_TO_MPA function block converts pressure from BAR to Mpa.



BAR is Metric unit of pressure.

MPA is SI unit of pressure. It is mega pascal.

Input

Input Parameter	Data types	Description
Bar	LREAL	Input pressure in Bar.

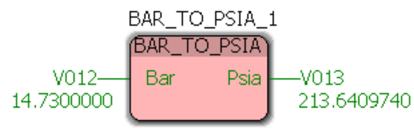
Output

Output Parameter	Data types	Description
Mpa	LREAL	Output pressure in mega pascal

BAR_TO_PSIA

Description

BAR_TO_PSIA function block converts pressure from BAR to PSIA.



BAR is Metric unit of pressure.

PSIA is US unit of pressure. It is pressure per square inches absolute.

Input

Input Parameter	Data types	Description
Bar	LREAL	Input pressure in Bar.

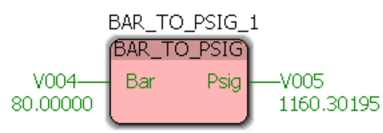
Output

Output Parameter	Data types	Description
Psia	LREAL	Output pressure in Psia.

BAR_TO_PSIG

Description

BAR_TO_PSIG function block converts pressure from BAR to PSIG.



BAR is Metric unit of pressure.

PSIG is US unit of pressure. It is pressure per square inches by gauge.

Input

Input Parameter	Data types	Description
Bar	LREAL	Input pressure in Bar.

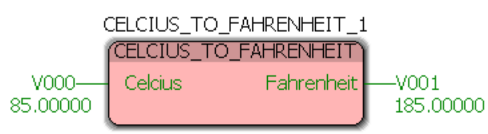
Output

Output Parameter	Data types	Description
Psig	LREAL	Output pressure in Psig.

CELCIUS_TO_FAHRENHEIT

Description

ELCIUS_TO_FAHRENHEIT function block converts temperature from Celcius to Fahrenheit.



Input

Input Parameter	Data types	Description
Celcius	LREAL	Input temperature in Celcius

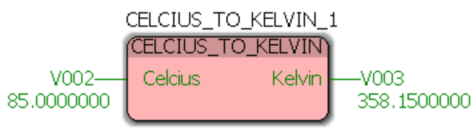
Output

Output Parameter	Data types	Description
Fahrenheit	LREAL	Output temperature in Fahrenheit

CELCIUS_TO_KELVIN

Description

CELCIUS_TO_KELVIN function block converts temperature from Celcius to Kelvin.



Input

Input Parameter	Data types	Description
Celcius	LREAL	Input temperature in Celcius

Output

Output Parameter	Data types	Description
Kelvin	LREAL	Output temperature in Kelvin

CELCIUS_TO_RANKINE

Description

CELCIUS_TO_RANKINE function block converts temperature from Celcius to Rankine.

Input

Input Parameter	Data types	Description
Celcius	LREAL	Input temperature in Celcius

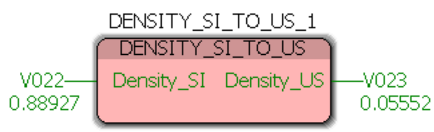
Output

Output Parameter	Data types	Description
Rankine	LREAL	Output temperature in Rankine

DENSITY_SI_TO_US

Description

ENSITY_SI_TO_US function block converts gas density from SI unit system to US unit system.



SI unit of the gas density is KG per cubic meter.

US unit of the gas density is LBM (pounds) per cubic feet.

Input

Input Parameter	Data types	Description
Density_SI	LREAL	Input value of gas density in SI unit.

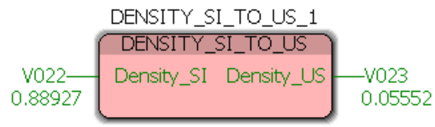
Output

Output Parameter	Data types	Description
Density_US	LREAL	Output value of gas density in US unit.

Density_TO_APIGravity

Description

Density_TO_APIGravity function block converts density of liquid to API gravity.



Input

Input Parameter	Data types	Description
KGPERM3	LREAL	Input value of density. It is in Kg/cubic meter.

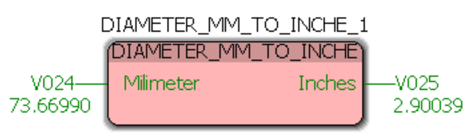
Output

Output Parameter	Data types	Description
API	LREAL	Output value of API Gravity.

DIAMETER_MM_TO_INCHE

Description

DIAMETER_MM_TO_INCHE function block converts diameter from millimeter to inches.



Millimeter is Metric unit of length.

An inch is the US unit of length.

Input

Input Parameter	Data types	Description
Milimeter	LREAL	Input value of diameter in Metric unit system.

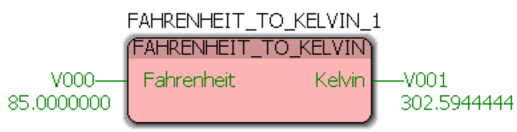
Output

Output Parameter	Data types	Description
Inches	LREAL	Output value of diameter in US unit system.

FAHRENHEIT_TO_KELVIN

Description

FAHRENHEIT_TO_KELVIN function block converts temperature from Fahrenheit to Kelvin.



Input

Input Parameter	Data types	Description
Fahrenheit	LREAL	Input temperature in Fahrenheit

Output

Output Parameter	Data types	Description
Kelvin	LREAL	Output temperature in Kelvin

FAHRENHEIT_TO_RANKINE

Description

FAHRENHEIT_TO_RANKINE function block converts temperature from Fahrenheit to Rankine.

Input

Input Parameter	Data types	Description
Fahrenheit	LREAL	Input temperature in Fahrenheit

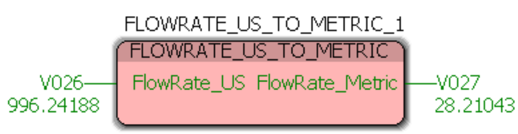
Output

Output Parameter	Data types	Description
Rankine	LREAL	Output temperature in Rankine

FLOWRATE_US_TO_METRIC

Description

FLOWRATE_US_TO_METRIC function block converts volumetric flow rate from US unit to Metric unit.



US unit of flow rate is cubic feet per hour.

Metric unit of flow rate is cubic meter per hour.

Input

Input Parameter	Data types	Description
FlowRate_US	LREAL	Input value of flow rate in US unit system.

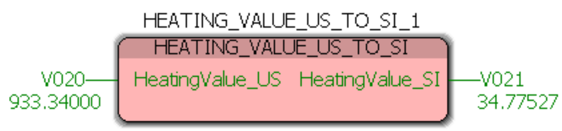
Output

Output Parameter	Data types	Description
FlowRate_Metric	LREAL	Output value of flow rate in Metric unit system.

HEATING_VALUE_US_TO_SI

Description

HEATING_VALUE_US_TO_SI function block converts gas heating value from US unit system to SI unit system.



US unit of the gas heating is BTU per cubic feet.

SI unit of the gas heating value is Mega joules per cubic meter.

Input

Input Parameter	Data types	Description
HeatingValue_US	LREAL	Input value of gas heating value in US unit.

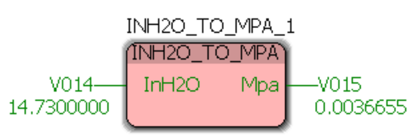
Output

Output Parameter	Data types	Description
HeatingValue_SI	LREAL	Output value of gas heating value in SI unit.

INH2O_TO_MPA

Description

INH2O_TO_MPA function block converts differential pressure from INH2O to MPA.



INH2O (Inches of water) is US unit to measure differential pressure of orifice meter.

MPA is SI unit of pressure. It is mega pascal.

Input

Input Parameter	Data types	Description
InH2O	LREAL	Input differential pressure in InH2O.

Output

Output Parameter	Data types	Description
Mpa	LREAL	Output differential pressure in Mpa.

KGPERM_TO_REL_DENSITY

Description

GPERM_TO_REL_DENSITY function block converts density of the liquid to relative density of liquid.

Input

Input Parameter	Data types	Description
Density	LREAL	Input value of density. It is in kg/m^3.

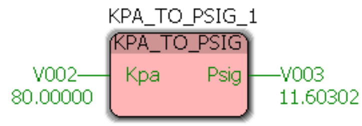
Output

Output Parameter	Data types	Description
RelativeDensity	LREAL	Output value of relative density.

KPA_TO_PSIG

Description

KPA_TO_PSIG function block converts pressure from KPA to PSIG.



KPA is SI unit of pressure.

PSIG is US unit of pressure. It is pressure per square inches by gauge.

Input

Input Parameter	Data types	Description
Kpa	LREAL	Input pressure in Kpa.

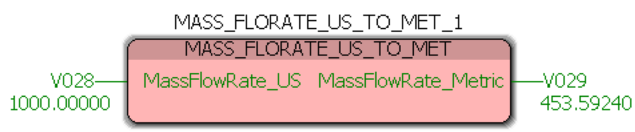
Output

Output Parameter	Data types	Description
Psig	LREAL	Output pressure in Psig.

MASS_FLORATE_US_TO_MET

Description

MASS_FLORATE_US_TO_MET function block converts mass flow rate from US unit to Metric unit.



US unit of mass flow rate is LBM (pounds) per hour.

Metric unit of mass flow rate is KG (Kilograms) per hour.

Input

Input Parameter	Data types	Description
MassFlowRate_US	LREAL	Input value of mass flow rate in US unit system.

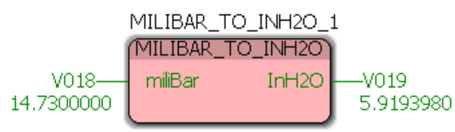
Output

Output Parameter	Data types	Description
MassFlowRate_Metric	LREAL	Output value of mass flow rate in Metric unit system.

MILIBAR_TO_INH2O

Description

MILIBAR_TO_MPA function block converts differential pressure from Milibar to INH2O.



Milibar is Metric unit to measure differential pressure of orifice meter.

INH2O (Inches of water) is US unit to measure differential pressure of orifice meter.

Input

Input Parameter	Data types	Description
miliBar	LREAL	Input differential pressure in Milibar.

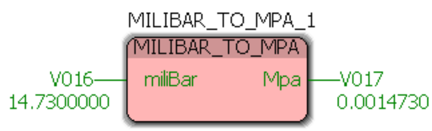
Output

Output Parameter	Data types	Description
InH2O	LREAL	Output differential pressure in Inches of water.

MILIBAR_TO_MPA

Description

MILIBAR_TO_MPA function block converts differential pressure from Milibar to MPA.



Milibar is Metric unit to measure differential pressure of orifice meter.

MPA is SI unit of pressure. It is mega pascal.

Input

Input Parameter	Data types	Description
miliBar	LREAL	Input differential pressure in Milibar.

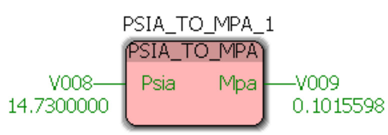
Output

Output Parameter	Data types	Description
Mpa	LREAL	Output differential pressure in Mpa.

PSIA_TO_MPA

Description

PSIA_TO_MPA function block converts pressure from PSIA to Mpa.



PSIA is US unit of pressure. It is pressure per square inches absolute.

MPA is SI unit of pressure. It is mega pascal.

Input

Input Parameter	Data types	Description
Psia	LREAL	Input pressure in Psia.

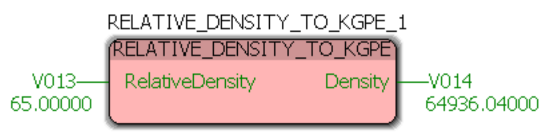
Output

Output Parameter	Data types	Description
Mpa	LREAL	Output pressure in mega pascal

RELATIVE_DENSITY_TO_KGPE

Description

Type topic text here. RELATIVE_DENSITY_TO_KGPE function block converts relative density of liquid to density of the liquid.



Input

Input Parameter	Data types	Description
RelativeDensity	LREAL	Input value of relative density.

Output

Output Parameter	Data types	Description
Density	LREAL	Output value of density. It is in kg/m^3.

THERMAL_EXPAN_CEL_TO_FEH

Description

HERMAL_EXPAN_CEL_TO_FEH function block converts the thermal expansion factor from degree Celcius to degree Fahrenheit.

Input

Input Parameter	Data types	Description
TE_C	LREAL	Input value of thermal expansion factor in °C-1.

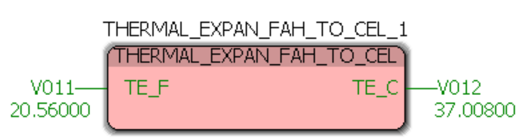
Output

Output Parameter	Data types	Description
TE_F	LREAL	Output value of thermal expansion factor in °F-1.

THERMAL_EXPAN_FAH_TO_CEL

Description

THERMAL_EXPAN_FAH_TO_CEL function block converts the thermal expansion factor from degree Fahrenheit to degree Celcius.



Input

Input Parameter	Data types	Description
TE_F	LREAL	Input value of thermal expansion factor in °F-1.

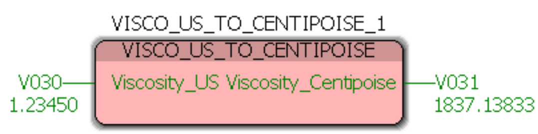
Output

Output Parameter	Data types	Description
TE_C	LREAL	Output value of thermal expansion factor in °C-1.

VISCO_US_TO_CENTIPOISE

Description

VISCO_US_TO_CENTIPOISE function block converts viscosity from US unit to Centipoise.



US unit of viscosity is pound per foot-sec.

Input

Input Parameter	Data types	Description
Viscosity_US	LREAL	Input value of viscosity in US unit system.

Output

Output Parameter	Data types	Description
Viscosity_Centipoise	LREAL	Output value of viscosity in Centipoise.

CHAPTER

13

UTILITYLIB

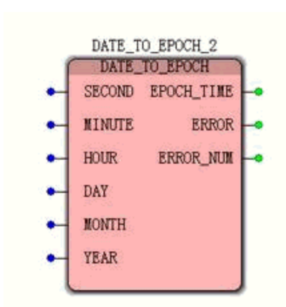
The following Utility function blocks are available:

Function block	Short description
Set RTC	Set the controller Real Time Clock by a provided timestamp value.
Get RTC	Read out the current time and date from the real-time clock and presents them as the parameters.
EPOCH_TO_DATE	Converts the EPOCH time to local timestamp value.
DATE_TO_EPOCH	Converts local timestamp value to the EPOCH time.
GetMicroTickCount	Returns tick count in microseconds.
SafeMove	Guarantee the consistence of the value.

DATE_TO_EPOCH

Description

This function Libraries converts local timestamp value to the EPOCH time.



INPUT

SECOND: Second

MINUTES: Minute

HOUR: Hour

DAY: Day

MONTH: Month

YEAR: year number, valid range: 0~37 (2000-2037)

OUTPUT

EPOCH_TIME: (also known as POSIX time or Unix time) is a system for describing instants in time, defined as the number of seconds that have elapsed since 00:00:00 Coordinated Universal Time (UTC), Thursday, 1 January 1970, not counting leap seconds.

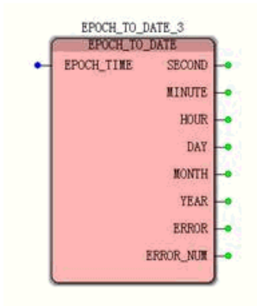
Error Message

ERROR_NUM: Error code	Description
0	no error
1	wrong second, valid range: 0 to 59
2	wrong minute, valid range: 0 to 59
3	wrong hour, valid range: 0 to 23
4	wrong day, valid range: 1 to 31
5	wrong month, valid range: 1 to 12
6	wrong year, valid range: 0 to 37
8	Fail to read RTC
9	Fail to write RTC
10	Fail to convert between date and EPOCH time

EPOCH_TO_DATE

Description

This function Libraries converts the EPOCH time to local timestamp value.



INPUT

EPOCH_TIME: (also known as POSIX time or Unix time) is a system for describing instants in time, defined as the number of seconds that have elapsed since 00:00:00 Coordinated Universal Time (UTC), Thursday, 1 January 1970, not counting leap seconds

OUTPUT

SECOND: Second

MINUTE: Minute

HOUR: Hour

DAY: Day

MONTH: Month

YEAR: year number, range from 0 to 37 (2000~2037), or from 70 to 99 (1970~1999)

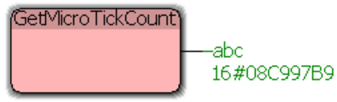
ERROR: True or False

ERROR_NUM: Error code 10, fail to convert between date and EPOCH time or EPOCH_TIME is negative, which is invalid.

GetMicroTickCount

Description

GetMicroTickCount function returns tick count in microseconds. The Output type is UDINT. That means the tick count would roll over every 4294 seconds.



Input

N/A

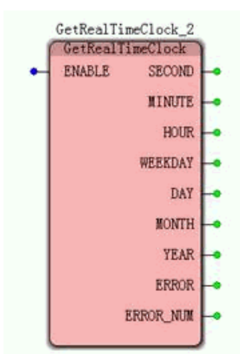
Output

Micro Tick Count (UDINT)

Get Real Time Clock

Description

If the function block "GetRealTimeClock" is called, it reads out the current time and date from the real-time clock and presents them as the parameters described below. An enable is not required, the block provides the values as soon as it is called.



> **NOTE:** Get RTC - This function block shall return the controller Real Time Clock (RTC) as a UTC or GMT timestamp value.

Input

ENABLE: Enable for accepting the applied values

Output

SECOND: Second

MINUTE: Minute

HOUR: Hour

WEEKDAY: Weekday, range from 0 to 6 (0=Sunday)

DAY: Day

MONTH: Month

YEAR: current year number (2-figures)

EPOCH_TIME: (also known as POSIX time or Unix time) is a system for describing instants in time, defined as the number of seconds that have elapsed since 00:00:00 Coordinated Universal Time (UTC), Thursday, 1 January 1970, not counting leap seconds.

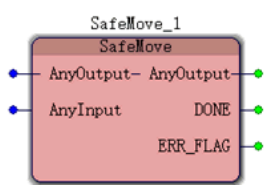
ERROR: True or False

ERROR_NUM: Error code

SafeMove

Description

If a global variable is used in multiple tasks, in order to guarantee the consistence of the value, this function block is used.



Input and Output

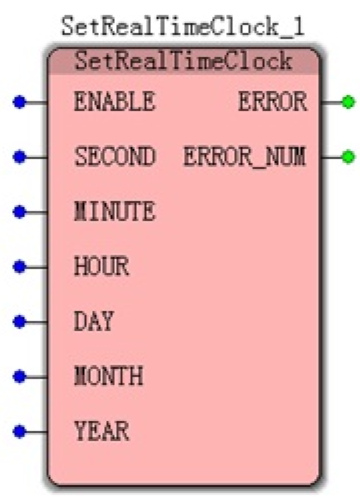
Pin	Type	Description
AnyInput	Any type	The source of the SafeMove function block
AnyOutput	Any type	The destination of the SafeMove function block with same length as AnyInput

Pin	Type	Description
DONE	BOOL	Indicate whether the move operation is executed successfully. Set true if move successfully from AnyInput to AnyOutput
ERR_FLAG	BOOL	SafeMove will check the length of AnyInput and AnyOutput. If they don't equal, this flag will report true.

Set Real Time Clock

Description

This function block shall set the controller Real Time Clock by a provided local timestamp value.



Input

ENABLE: Enable for accepting the applied values, rising edge.

SECOND: Second

MINUTES: Minute

HOUR: Hour

DAY: Day

MONTH: Month

YEAR: year number, valid range from 0 to 37(2000 to 2037)

Output

ERROR: Error message

ERROR_NUM: Error code

Error Code	Description
0	No error
1	Wrong second, valid range: 0 to 59
2	Wrong minute, valid range: 0 to 59
3	Wrong hour, valid range: 0 to 23
4	Wrong day, valid range: 1 to 31
5	Wrong month, valid range: 1 to 12
6	Wrong year, valid range: 1990 to 2099
7	Fail to read RTC
8	Fail to write RTC

CHAPTER

14 APINGLLIB

The following API NGL function block is available:

Function block	Short description
API NGL Block	API NGL Block calculates: Base density using API Chapter 11, Section 2, Part 4 in conjunction with either API Chapter 11.2.2 or Chapter 11.2.2M. Calculates standard density using API Chapter 11, Section 2, Part 4 in conjunction with either API Chapter 11.2.2 or Chapter 11.2.2M. Calculation of vapor pressure using API Chapter 11, Section 2, Part 5.

API NGL Function Block

This function block does calculations using following standards.

- API Manual of Petroleum Measurement Standards, Chapter 11, Section 2, Part 4 is an international standard covering temperature volume correction for NGLs and LPGs.
- API Manual of Petroleum Measurement Standards, Chapter 11, Section 2, Part 5 is an international standard covering vapor pressure correlation for commercial NGLs.
- API Manual of Petroleum Measurement Standards, Chapter 11.2.2 is an international standard covering compressibility factors for Hydrocarbons from relative density and temperature (Fahrenheit).
- API Manual of Petroleum Measurement Standards, Chapter 11.2.2M is an international standard covering compressibility factors for Hydrocarbons from density (Kg/m3) and temperature (Centigrade).

Information:

The basic function of API NGL block when set for line to base operation is to calculate standard density and associated volume correction factor from an observed density, temperature and pressure with an option to either calculate a vapor pressure or use an operator entered value.

The basic function of API NGL block when set for base to line operation is to calculate meter density and associated volume correction factor from an observed density, temperature and pressure with an option to either calculate a vapor pressure or use an operator entered value.

API NGL block solves either a line to base or base to line correction but not both.

It is possible, however, to connect the resulting standard density from a line to base block to the input of a base to line block.

Input

Input Parameter	Data types	Description
APITables	INT	This can be from one of the following standards: 0 - T23E (line to base (60 'F) from observed relative density) 1 - T24E (base to line from standard relative density (60 'F)) 2 - T53E (line to base (15 'C) from observed Kg/m3) 3 - T54E (base to line to from standard Kg/m3(15 'C)) 4 - T59E (line to base (20 'C) from observed Kg/m3) 5 - T60E (base to line to from Kg/m3 (20 'C))
CPLCalcType	INT	CPL Calculation Type. This can be from one of the following options: 0 – None (No CPL calculation is performed)

Input Parameter	Data types	Description
		2 - API1122 (CPL calculated from standard) relative density (60 'F) and observed temperature) 4 - API1122M
ConverCriteria	LREAL	IP2 Convergence limit. Reserved for future use. Set to 0.001
MaxIterations	INT	IP2 Max loop limit. Reserved for future use. Preset to Set to 50.
DensityInput	LREAL	Density Input
DensityUnits	INT	This can be from one of the following options: 0 - Kg/m3 1 - RD 2 - Degrees API
IterationMethod	INT	Main calculation method such as ASTM and IP2 0 - ASTM 1 - IP2
PressureInput	LREAL	Pressure Input.
PressureUnits	INT	0 - PSIA 1 - PSIG 2 - Kpa 3 - Bar 4 - BarG
EquilibPressureInput	LREAL	Operator Entered or Observed Vapor Pressure Input
PECalcType	INT	Vapor Pressure Options. The vapor pressure option can be from one of the following values:

Input Parameter	Data types	Description
		0 - None (Vapor pressure assumed to be zero.) 1 - Use Observed (The operator entered value is used) 2 - API1125 (Vapor pressure calculated from standard relative density at 60 'F and observed Temperature.)
ReferanceTemperature	LREAL	Reference Temperature. This is only used by T59E and T60E. The usual value is 20. 0 degC.
Rounding	INT	0 – Rounding Disabled 2 – Rounding Enabled When enabled, the function block follows the rounding standards specified by the calculations. The API 11.2.4 temperature correction calculation only specifies rounding for the inputs and final results i.e. no interim variables require rounding.
TemperatureInput	LREAL	Observed Temperature Input
TemperatureUnits	INT	0 – Deg F 1 – Deg C

Output

Output Parameter	Data types	Description
CPL	LREAL	Correction factor for effects of pressure on the liquid
CTL	LREAL	Correction factor for effects of temperature on the liquid
CTPL	LREAL	Correction factor for effects of temperature and pressure on the liquid
AFactor	LREAL	CPL calculation interim result
BFactor	LREAL	CPL calculation interim result

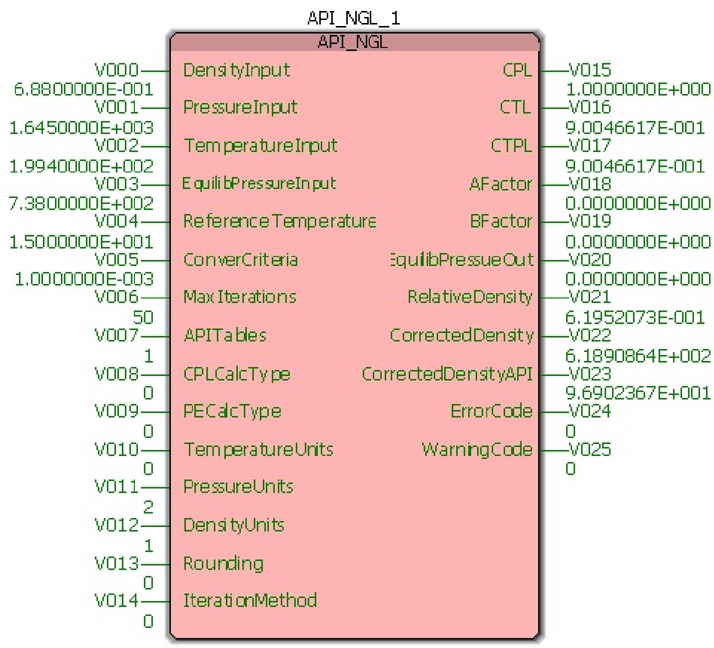
Output Parameter	Data types	Description
EquilibriumPressureOut	LREAL	Calculated (or used) Vapor Pressure
Relative Density	LREAL	Relative Density Corrected
CorrectedDensity	LREAL	Calculated Density in kg/m3 units
CorrectedDensityAPI	LREAL	Calculated Density in degrees API
ErrorCode	INT	Critical error code
WarningCode	INT	Warning code

Operating Limits:

The API 11 Chapter 11 methods are derived from correlations of density versus physical conditions of liquids. As such they are only valid over certain operating ranges. To allow flexibility, especially with the IP2 iteration method where compressibility and temperature correction calculation results interact, the limits are set as follows:

T23E	RD from 0.21 to 0.74. Temperature from -50.8 'F to 199.4 'F
T24E	RD from 0.35 to 0.688. Temperature from -50.8 'F to 199.4 'F
T53E	Density from 210 to 739 kg/m3. Temperature from -46 'C to 93 'C
T54E	Density from 351.7 to 687.8 kg/m3. Temperature from -46 'C to 93 'C
T59E	Density from 351.7 to 687.8 kg/m3. Temperature from -46 'C to 93 'C
T60E	Density from 331.7 to 683.6 kg/m3. Temperature from -46 'C to 93 'C
API1122	RD from 0.2 to 0.75. Temperature from -50.8 'F to 140 'F. Pressure from 0 to 2200 psi

API1122M	Density from 200 to 750 kg/m3. Temperature from -50.8 'F to 140 'F. Pressure from 0 to 2200 psi.
API1125	RD from 0.2 to 0.75. Temperature from -50 'F to 140 'F



Critical Error Codes

Code	Description
1	Density value is invalid
2	Temperature value is invalid
3	Pressure value is invalid
4	Vapor Pressure value is invalid

Warning codes

Code	Description
1	Density input is out of range (all calculations)
2	Temperature input is out of range (all calculations)
3	Pressure input is out of range (all calculations)
4	Calculation combination is invalid (all calculations)
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	API.11.2.4: Alpha error
15	API.11.2.4: Interpolation variable error
16	API.11.2.4: TC error
17	API.11.2.4: TRX error
18	API.11.2.4: H2 error
19	API.11.2.4: Saturated density error
20	API.11.2.4: Interpolation factor error
21	API.11.2.4: Step 4-5 error
22	API.11.2.4: Fluid 2 relative density low error
23	API.11.2.4: Step 6 TC2_TC1 error
24	API.11.2.4: RD X < Lower Limit
25	API.11.2.4: RD 60 Mid error
26	API.11.2.4: Step 9 Phi error

Code	Description
27	API.11.2.4: Step 9 A error
28	API.11.2.4: Step 9 B error
29	API.11.2.4: Step 9 RD 60 Trial error
30	API.11.2.4: Iteration Fail error
31	API.11.2.4: CTL range error
32	API.11.2.4: T60 Step 6 density error
101	API.11.2.4: Density conversion error
102	API.11.2.4: Rounding error
103	API.11.2.4: Reserved
104	API.11.2.4: CTL range error
105	API.11.2.4: CPL range error
106	API.11.2.4: Reserved
107	API.11.2.4: Reserved
108	API.11.2.4: Calculated density range error
109	API.11.2.4: Density units conversion error
110	API.11.2.4: Pressure units conversion error
111	API.11.2.4: CTPL range error
211	API 1122 and API1122M: TR > Max error
212	API 1122 and API1122M: Factor error
301	Ch.11.2.5: relative density out of range
302	Ch.11.2.5: Temperature out of range

Invalid generally means one of the following:

- The input block pin is not connected.
- The input value is NaN.
- The input value is out of range.

If critical errors occur, all key output parameters are forced to NaN.

CHAPTER

15

ISO5167DUALLIB

The following ISO 5167 Dual function block is available:

Function Block	Short Description
See ISO 5167Dual for more information.	ISO 5167 Dual function block calculates: Mass flow to the 1991, 1997 and 2003 versions of ISO 5167.Calorific value on a superior and inferior basis Gross volume flow, standard volume flow and energy flow. Fully-recovered downstream pressure. Calculation of upstream density from a downstream measurement (see section 11.2). Each density measurement input can be configured upstream or downstream independently. Calculation of upstream temperature from a downstream measurement (see section 11.1)

ISO 5167Dual

ISO 5167 is an international standard covering the measurement of fluid flow by means of pressure differential devices such as orifice plates and venturis. When some parameters are known, ISO 5167 allows other variables to be calculated. The most common usage is to calculate mass flow rate from differential pressure, static pressure and density. ISO 5167 is widely used in most areas of the world except North America.

The basic function of the ISO 5167 block is to calculate mass flow rate from primary element DP and other required inputs. This block supports the 1991, 1997 and 2003 versions of the ISO 5167 standard. These versions differ in small but significant ways.

- The basic functions supported are listed below
- Calculation of mass flow to the 1991, 1997 and 2003 versions of ISO 5167.

- Calculation of gross volume flow, standard volume flow and energy flow.
- Calculation of fully-recovered downstream pressure (see section 9).
- Dual density inputs with automatic fail-over and deviation checking (see section 5.4).
- Calculation of upstream density from a downstream measurement (see section 11.2). Each density measurement input can be configured upstream or downstream independently.
- Calculation of upstream temperature from a downstream measurement (see section 11.1).
- Temperature compensation of primary element and pipe.
- Gauge or absolute static pressure transmitters located upstream or downstream.
- Automatic selection of DP from up to three DP transmitters (see section 0).
- Orifice plates with all three tapping types (corner, D and D/2 and flange).
- Classical ventures of all three construction types: as-cast, machined and rough-welded.
- Externally calculated viscosity and isentropic exponent or constant values.
- Incompressible fluids (liquids) or compressible ones (gases).
- UK DTI limits on beta and Reynolds No for fiscal purposes.

Input

Input Parameter	Data types	Description	Key Configuration Parameter
CalorificValue	LREAL	Calorific value in MJ/Sm3	
ISO5167Version	INT	ISO5167 Version of the computation: 0 = version 1991; 1 = version 1997;	Yes

Input Parameter	Data types	Description	Key Configuration Parameter
		2 = version 2003	
FluidType	INT	Fluid type selection: 0 = Compressible; 1 = Uncompressible	Yes
DensityFromBlockPin	BOOL	Density configuration: 1: RHOTP from block pin; 0: constant value	Yes
ConstantDensity	LREAL	Constant Density. Value to be provided when DensityFromBlockPin is selected as 0	Yes
DensityInput1	LREAL	Density Input 1	
DensityInput2	LREAL	Density Input 2	
NoOfDensityInputs	INT	Number of density inputs 0-1 densitometer, 1-2 densitometer, 2-NA[1]	Yes
DensityInputSelection	INT	Density input selection 0 - Auto 1 - 1st Densitometer , 2-2nd Densitometer, 3- NA	Yes
DensityMeasurPosition1	INT	Density measurement position for input 1 0 = Upstream, 1 = Downstream, 2- NA	Yes

[1]Not Applicable

Input Parameter	Data types	Description	Key Configuration Parameter
DensityMeasurPosition2	INT	Density measurement position for input 2 0 = Upstream, 1 = Downstream, 2- NA	Yes
DensityInputComparDB	LREAL	Density inputs 1 and 2 comparison deadband. This input should be between 0 and 10	Yes
DensityInputComparTimeDelay	INT	Density inputs 1 and 2 comparison time delay. This input should be between 0 and 300	Yes
RhoInputSelStatus1	BOOL	1st Densitometer input status	Yes
RhoInputSelStatus2	BOOL	2nd Densitometer input status	Yes
ViscosityFromBlockPin	BOOL	Viscosity of the fluid 1: VISCOSITY from block pin; 0: constant value	Yes
ViscosityOfFluid	LREAL	Viscosity of the fluid, if ViscosityFromBlockPin is selected as constant value.	Yes
ISEN_EXPFromBlockPin	BOOL	Isentropic Exponent 1: Isentropic Exponent from block pin; 0: constant value	Yes
IsentropicExponent	LREAL	Isentropic Exponent if ISEN_ EXPFromBlockPIN is selected as constant	

Input Parameter	Data types	Description	Key Configuration Parameter
		value.	
MassFlowUnit	INT	Mass flow units: 0 = kg/sec; 1 = kg/min; 2 = kg/hour; 3 = tonne/min; 4 = tonne/hour	Yes
MassFlowScaling	LREAL	Mass flow scaling factor. This value should be configured as > 0.0	Yes
QVComputation	BOOL	Carry out computation for Volume Flow or not. 1 = Enable; 0 = Disable	Yes
VolumeFlowUnit	INT	Volume flow units: 0 = m3/sec; 1 = m3/min; 2 = m3/hour; 4 = km3/hour	Yes
VolumeFlowScaling	LREAL	Volume flow scaling factor. This value should be configured as > 0.0	Yes
QSComputation	BOOL	Carry out Standard Volume Flow Computation or not. 1 = Enable; 0 = Disable	Yes

Input Parameter	Data types	Description	Key Configuration Parameter
StdVolumeFlowUnit	INT	Standard volume units: 0 = Sm3/sec; 1 = Sm3/min; 2 = Sm3/hour; 4 = kSm3/hour	Yes
StdVolumeFlowScaling	LREAL	Standard volume flow scaling factor. This value should be configured as > 0.0	Yes
QHComputation	BOOL	Carry out Energy Flow Computation or not. 1 = Enable; 0 = Disable	Yes
EnergyFlowUnit	INT	Energy flow units: 0 = KJ/sec; 1 = MJ/sec; 2 = MJ/min; 3 = MJ/hour; 4 = GJ/hour	Yes
EnergyFlowScaling	LREAL	Energy flow scaling factor. This value should be configured as > 0.0	Yes
InitialCValue	LREAL	Initial C value. Default value is 0.6	Yes
MaxItrations	INT	Maximum number of iterations. The value should be between 6 and 12.	Yes
PrecisionLimit	LREAL	Precision Limits. The	Yes

Input Parameter	Data types	Description	Key Configuration Parameter
		value should be between 0.000000001 and 0.000001.	
FiscalMetering	BOOL	Fiscal Metering: 1 = Yes; 0 =No	Yes
PrmiaryElementType	INT	Primary element type: 0 = Orifice Plate; 1 = Classical Venturi	Yes
OrificeTapType	INT	Orifice plate tap type: 0 = Corner; 1 = Flange; 2 = D&D/2	Yes
VenturiType	INT	Venturi meter type: 0 = As-Cast; 1 = Machined; 2 = Roughwelded	Yes
AllowanceForExp	BOOL	Allowance for expansion: 1 = Yes; 0 =No	Yes
PipeRefTemperature	LREAL	Pipe reference temperature (deg C). The value should be between 0 and 50 Deg C.	Yes
PipeCoefficient	LREAL	Pipe coefficient of expansion (mm/mm/deg C). The	Yes

Input Parameter	Data types	Description	Key Configuration Parameter
		value should be between 0.000005 and 0.00005.	
ElementRefTemperature	LREAL	Primary element reference temperature (deg C). The value should be between 0 and 50 Deg C.	Yes
ElementCoefficient	LREAL	Primary element coefficient of expansion (mm/mm/deg C). The value should be between 0.000005 and 0.00005.	Yes
PipeReferenceBore	LREAL	Pipe reference bore (mm) or pipe diameter. The value should be between 16.67 and 1200.	Yes
ElementReferenceBore	LREAL	Primary element reference bore (mm) or primary element diameter. The value should be between 12.5 and 1000.	Yes
PermLossA	LREAL	Venturi permanent pressure loss(%DP) for coefficients A. The value should be between 0 and 5.	Yes
PermLossB	LREAL	Venturi permanent pressure loss(%DP) for coefficients B.	Yes
StaticPressMeasurementPos	INT	static pressure measurement position (0 = Upstream, 1 =	Yes

Input Parameter	Data types	Description	Key Configuration Parameter
		Downstream)	
StaticPressUnit	INT	Static Pressure units: 0 = KPa; 1 = MPa; 2 = bar	Yes
StaticPressBasis	INT	Static Pressure Base: 0 = Gauge; 1 = Absolute	Yes
AtmosphericPress	LREAL	Atmospheric Pressure	Yes
AtmosphericPressUnit	INT	Atmospheric pressure measurement units: 0 = KPa abs; 1 = MPa abs; 2 = bara.	Yes
DiffPressUnit	INT	Deferential pressure measurement units: 0 = KPa; 1 = MPa; 2 = bar; 3 = mbar.	Yes
TempMeasurePosition	INT	temperature measurement position. (0 = Upstream, 1= Downstream)	Yes
DiffPressureTxNumber	INT	No. of differential pressure transmitters: 1 = 1 transmitter;	Yes

Input Parameter	Data types	Description	Key Configuration Parameter
		2 = 2 transmitter; 3 = 3 transmitter.	
HiLimDP1	LREAL	Hight limit value of transition 1-2. The value should between 50 and 95.	Yes
HiLimDP2	LREAL	Hight limit value of transition 2-3. The value should between 50 and 95.	Yes
DeadbandValueDP1	LREAL	Deadband value of transition 1-2. The value should between 0 and 10.	Yes
DeadbandValueDP2	LREAL	Deadband value of transition 2-3. The value should between 0 and 10.	Yes
DiffPressInput1	LREAL	Diff Pressure Transmitter input 1	
DiffPressInput2	LREAL	Diff Pressure Transmitter input 2	
DiffPressInput3	LREAL	Diff Pressure Transmitter input 3	
DiffPressureStatus1	BOOL	Diff Pressure Transmitter 1 status; 0=OK, 1=fault	Yes
DiffPressureStatus2	BOOL	Diff Pressure Transmitter 2 status; 0=OK, 1=fault	Yes
DiffPressureStatus3	BOOL	Diff Pressure Transmitter 3 status; 0=OK, 1=fault	Yes
DPPVEUHI1	LREAL	DP transmitter x EUHI	

Input Parameter	Data types	Description	Key Configuration Parameter
DPPVEUHI2	LREAL	DP transmitter x EUHI	
StaticPressure	LREAL	Static Pressure	
StandardDensity	LREAL	Standard density	
Temperature	LREAL	Temperature	
CalorificValue	LREAL	Calorific Value	

Output

Output Parameter	Data types	Description
BetaRatio	LREAL	Beta ratio (d/D) at flowing conditions
ElementActualBore	LREAL	Corrected bore/throat size
CValue	LREAL	Coefficient of discharge
SelectedDiffPressure	LREAL	Selected differential pressure
DiffTxInuse	INT	In-use DP transmitter
ExpFactor	LREAL	Expansibility factor
Pressure1Abs	LREAL	Upstream absolute pressure in Pa
Pressure1Guage	LREAL	Upstream gauge pressure
Pressure3Abs	LREAL	Fully recovered downstream absolute pressure
Pressure3Guage	LREAL	Fully recovered downstream gauge pressure
PipeActBore	LREAL	Corrected pipe size
Qh	LREAL	Energy flow
Qm	LREAL	Mass flow
Qs	LREAL	Standard volume flow
Qv	LREAL	Volume flow
Red	LREAL	Reynolds Number

Output Parameter	Data types	Description
RHO1	LREAL	In-use Upstream density
RHO1_1	LREAL	Upstream density derived from RHOTP1
RHO1_2	LREAL	Upstream density derived from RHOTP2
Temperature1	LREAL	Upstream temperature
VelApproachFactor	LREAL	Velocity of Approach factor
NumberIterations	LREAL	Number of Iterations for the last scan
ErrorCode	INT	Critical Error Code
WarningCode	INT	Warning Code
JT_COEFF	LREAL	Joule-Thomson coefficient in K/bar. If Version < 2003 Or Fluid is incompressible, value = NaN.

In ISO 5167 block, there is DP, static pressure, temperature and density exposed as input pin (or constant density from the configuration form) and measured form field. There are other additional inputs as well. Based on these inputs, the mass flow, volume flow, standard volume flow and energy flow of the fluid can be figured out via ISO5167_DUAL function block.

Input parameters range:

Input Parameter	Min Value	Max Value
PipeReferenceBore	16.67	1200
ElementReferenceBore	12.5	1000
DensityInputComparTimeDelay	0	300
HiLimDP1	50.0	95.0
HiLimDP2	50.0	95.0
PipeRefTemperature	0	50
ElementRefTemperature	0	50

Input Parameter	Min Value	Max Value
MaxItrations	6	12
DeadbandValueDP1	0	10.0
DeadbandValueDP2	0	10.0
DensityInputComparDB	0	10
IsentropicExponent	1.0	5.0
InitialCValue	0.58	0.62
PermLossA	0	5.0
PermLossB	0	5.0
PipeCoefficient	0.000005	0.00005
ElementCoefficient	0.000005	0.00005
PrecisionLimit	0.000000001	0.000001

Error and Warning list

Critical Error Codes

Code	Description
1	Static pressure value is invalid (if a compressible fluid is selected).
2	Differential pressure value is invalid.
3	If Temp Comp is enabled, temperature value is invalid.
4	Density value is invalid.
5	Viscosity value is invalid.
6	Isentropic exponent value is invalid (if a compressible fluid is selected).
7	Iteration failed to converge.
8	Multiple DP transmitter configuration is invalid.
9	Pipe bore is invalid.
10	Beta is invalid.
11	P2 is invalid (if a compressible fluid is selected).

Code	Description
20	If standard volume is enabled, standard density is invalid.
21	If energy flow is enabled, CV by volume is invalid.

- Error 30 will appear if any of the parameter in "Key configuration parameter" marked as "Yes" in Inputs table is not configured or wrongly configured.
- If critical errors 1 to 11 and 30 occur, Qm and all derived values are set to NaN. If critical error 20 occurs, QS and QH are set to NaN. If critical error 21 occurs, QH is set to NaN.

Warning codes

Code	Description
1	For a compressible fluid, P2/P1 ratio is too low.
2	Element bore is too small.
3	Pipe size is out of range for an orifice plate.
4	Pipe size is out of range for a venturi.
5	Orifice beta ratio is outside fiscal limits.
6	Orifice beta ratio is outside limits.
7	Venturi beta ratio is outside limits.
8	Orifice plate is outside Reynolds No limits.
9	Orifice plate is above fiscal Reynolds No limit.
10	Venturi is outside Reynolds No limits.
11	For dual density inputs, input 1 is invalid.
12	For dual density inputs, input 2 is invalid.
13	For dual density inputs, the deviation between the inputs is greater than the deadband.

CHAPTER

16 ISO5167DUALJTLIB

The following ISO 5167 Dual JT function block is available:

Function block	Short description
ISO 5167 DUAL JT	ISO 5167 Dual JT Block calculates 1. Mass flow to the 1991, 1997 and 2003 versions of ISO 5167.Calorific value on a superior and inferior basis 2. Gross volume flow, standard volume flow and energy flow. 3. Fully-recovered downstream pressure. 4. Calculation of upstream density from a downstream measurement (see section 11.2). Each density measurement input can be configured upstream or downstream independently. 5. Calculation of upstream temperature from a downstream measurement (see section 11.1)

ISO 5167 DUAL JT

ISO 5167 DUAL JT is an international standard covering the measurement of fluid flow by means of pressure differential devices such as orifice plates and venturis. When some parameters are known, ISO 5167 allows other variables to be calculated. The most common usage is to calculate mass flow rate from differential pressure, static pressure and density. ISO 5167 is widely used in most areas of the world except North America.

Information:

The basic function of the ISO 5167 block is to calculate mass flow rate from primary element DP and other required inputs. This block supports the 1991, 1997 and 2003 versions of the ISO 5167 standard. These versions differ in small but significant ways.

The basic functions supported are listed below

- Calculation of mass flow to the 1991, 1997 and 2003 versions of ISO 5167.

- Calculation of gross volume flow, standard volume flow and energy flow.
- Calculation of fully-recovered downstream pressure (see section 9).
- Dual density inputs with automatic fail-over and deviation checking (see section 5.4).
- Calculation of upstream density from a downstream measurement (see section 11.2). Each density measurement input can be configured upstream or downstream independently.
- Calculation of upstream temperature from a downstream measurement (see section 11.1).
- Temperature compensation of primary element and pipe.
- Gauge or absolute static pressure transmitters located upstream or downstream.
- Automatic selection of DP from up to three DP transmitters (see section 0).
- Orifice plates with all three tapping types (corner, D and D/2 and flange).
- Classical ventures of all three construction types: as-cast, machined and rough-welded.
- Externally calculated viscosity and isentropic exponent or constant values.
- Incompressible fluids (liquids) or compressible ones (gases).
- UK DTI limits on beta and Reynolds No for fiscal purposes.

Input

Input Parameter	Data types	Description	Key Configuration Parameter
ISO5167Version	INT	ISO5167 Version of the computation: 0 = version 1991; 1 = version 1997; 2 = version 2003	Yes

Input Parameter	Data types	Description	Key Configuration Parameter
FluidType	INT	Fluid type selection: 0 = Compressible; 1 = Uncompressible	Yes
DensityFromBlockPin	BOOL	Density configuration: 1: RHOTP from block pin; 0: constant value	Yes
ConstantDensity	LREAL	Constant Density. Value to be provided when DensityFromBlockPin is selected as 0	Yes
DensityInput1	LREAL	Density Input 1	
DensityInput2	LREAL	Density Input 2	
NoOfDensityInputs	INT	Number of density inputs 0 = 1 densitometer, 1 = 2 densitometer, 2 = NA[1]	Yes
DensityInputSelection	INT	Density input selection 0 - Auto 1 - 1st Densitometer , 2-2nd Densitometer, 3- NA	Yes
DensityMeasurPosition1	INT	Density measurement position for input 1 0 = Upstream, 1 = Downstream, 2- NA	Yes
DensityMeasurPosition2	INT	Density measurement position for input 2 0 = Upstream, 1 = Downstream, 2- NA	Yes

[1]Not Applicable

Input Parameter	Data types	Description	Key Configuration Parameter
DensityInputComparDB	LREAL	Density inputs 1 and 2 comparison deadband. This input should be between 0 and 10	Yes
DensityInputComparTimeDelay	INT	Density inputs 1 and 2 comparison time delay. This input should be between 0 and 300	Yes
RhoInputSelStatus1	BOOL	1st Densitometer input status	Yes
RhoInputSelStatus2	BOOL	2nd Densitometer input status	Yes
ZP1T1	LREAL	Compressibility at P1, T1 line conditions.	
ZP2T3	LREAL	Compressibility at P2, T3 line conditions.	
ViscosityFromBlockPin	BOOL	Viscosity of the fluid 1: VISCOSITY from block pin; 0: constant value	Yes
ViscosityOfFluid	LREAL	Viscosity of the fluid, if ViscosityFromBlockPin is selected as constant value.	Yes
ISEN_EXPFromBlockPin	BOOL	Isentropic Exponent 1: Isentropic Exponent from block pin; 0: constant value	Yes
IsentropicExponent	LREAL	Isentropic Exponent if ISEN_EXPFromBlockPIN is selected as constant value.	

Input Parameter	Data types	Description	Key Configuration Parameter
MassFlowUnit	INT	Mass flow units: 0 = kg/sec; 1 = kg/min; 2 = kg/hour; 3 = tonne/min; 4 = tonne/hour	Yes
MassFlowScaling	LREAL	Mass flow scaling factor. This value should be configured as > 0.0	Yes
QVComputation	BOOL	Carry out computation for Volume Flow or not. 1 = Enable; 0 = Disable	Yes
VolumeFlowUnit	INT	Volume flow units: 0 = m3/sec; 1 = m3/min; 2 = m3/hour; 4 = km3/hour	Yes
VolumeFlowScaling	LREAL	Volume flow scaling factor. This value should be configured as > 0.0	Yes
QSComputation	BOOL	Carry out Standard Volume Flow Computation or not. 1 = Enable; 0 = Disable	Yes
StdVolumeFlowUnit	INT	Standard volume units:	Yes

Input Parameter	Data types	Description	Key Configuration Parameter
		0 = Sm3/sec; 1 = Sm3/min; 2 = Sm3/hour; 4 = kSm3/hour	
StdVolumeFlowScaling	LREAL	Standard volume flow scaling factor. This value should be configured as > 0.0	Yes
QHComputation	BOOL	Carry out Energy Flow Computation or not. 1 = Enable; 0 = Disable	Yes
EnergyFlowUnit	INT	Energy flow units: 0 = KJ/sec; 1 = MJ/sec; 2 = MJ/min; 3 = MJ/hour; 4 = GJ/hour	Yes
EnergyFlowScaling	LREAL	Energy flow scaling factor. This value should be configured as > 0.0	Yes
InitialCValue	LREAL	Initial C value. Default value is 0.6	Yes
MaxItrations	INT	Maximum number of iterations. The value should be between 6 and 12.	Yes
PrecisionLimit	LREAL	Precision Limits. The value should be between 0.000000001 and	Yes

Input Parameter	Data types	Description	Key Configuration Parameter
		0.000001.	
FiscalMetering	BOOL	Fiscal Metering: 1 = Yes; 0 =No	Yes
PrmiaryElementType	INT	Primary element type: 0 = Orifice Plate; 1 = Classical Venturi	Yes
OrificeTapType	INT	Orifice plate tap type: 0 = Corner; 1 = Flange; 2 = D&D/2	Yes
VenturiType	INT	Venturi meter type: 0 = As-Cast; 1 = Machined; 2 = Roughwelded	Yes
AllowanceForExp	BOOL	Allowance for expansion: 1 = Yes; 0 =No	Yes
PipeRefTemperature	LREAL	Pipe reference temperature (deg C). The value should be between 0 and 50 Deg C.	Yes
PipeCoefficient	LREAL	Pipe coefficient of expansion (mm/mm/deg C). The value should be between 0.000005 and 0.00005.	Yes

Input Parameter	Data types	Description	Key Configuration Parameter
ElementRefTemperature	LREAL	Primary element reference temperature (deg C). The value should be between 0 and 50 Deg C.	Yes
ElementCoefficient	LREAL	Primary element coefficient of expansion (mm/mm/deg C). The value should be between 0.000005 and 0.00005.	Yes
PipeReferenceBore	LREAL	Pipe reference bore (mm) or pipe diameter. The value should be between 16.67 and 1200.	Yes
ElementReferenceBore	LREAL	Primary element reference bore (mm) or primary element diameter. The value should be between 12.5 and 1000.	Yes
PermLossA	LREAL	Venturi permanent pressure loss(%DP) for coefficients A. The value should be between 0 and 5.	Yes
PermLossB	LREAL	Venturi permanent pressure loss(%DP) for coefficients B.	Yes
StaticPressMeasurementPos	INT	Static pressure measurement position. (0 = Upstream, 1 = Downstream)	Yes
StaticPressUnit	INT	Static Pressure units: 0 = KPa;	Yes

Input Parameter	Data types	Description	Key Configuration Parameter
		1 = MPa; 2 = bar	
StaticPressBasis	INT	Static Pressure Base: 0 = Gauge; 1 = Absolute	Yes
AtmosphericPress	LREAL	Atmospheric Pressure	Yes
AtmosphericPressUnit	INT	Atmospheric pressure measurement units: 0 = KPa abs; 1 = MPa abs; 2 = bara.	Yes
DiffPressUnit	INT	Deferential pressure measurement units: 0 = KPa; 1 = MPa; 2 = bar; 3 = mbar.	Yes
TempMeasurePosition	INT	temperature measurement position. (0 = Upstream, 1= Downstream)	Yes
DiffPressureTxNumber	INT	No. of differential pressure transmitters: 1 = 1 transmitter; 2 = 2 transmitter; 3 = 3 transmitter.	Yes

Input Parameter	Data types	Description	Key Configuration Parameter
HiLimDP1	LREAL	Hight limit value of transition 1-2. The value should between 50 and 95.	Yes
HiLimDP2	LREAL	Hight limit value of transition 2-3. The value should between 50 and 95.	Yes
DeadbandValueDP1	LREAL	Deadband value of transition 1-2. The value should between 0 and 10.	Yes
DeadbandValueDP2	LREAL	Deadband value of transition 2-3. The value should between 0 and 10.	Yes
DiffPressInput1	LREAL	Diff Pressure Transmitter input 1	
DiffPressInput2	LREAL	Diff Pressure Transmitter input 2	
DiffPressInput3	LREAL	Diff Pressure Transmitter input 3	
DiffPressureStatus1	BOOL	Diff Pressure Transmitter 1 status; 0=OK, 1=fault	Yes
DiffPressureStatus2	BOOL	Diff Pressure Transmitter 2 status; 0=OK, 1=fault	Yes
DiffPressureStatus3	BOOL	Diff Pressure Transmitter 3 status; 0=OK, 1=fault	Yes
DPPVEUHI1	LREAL	DP transmitter x EUHI	
DPPVEUHI2	LREAL	DP transmitter x EUHI	

Input Parameter	Data types	Description	Key Configuration Parameter
StaticPressure	LREAL	Static Pressure	
StandardDensity	LREAL	Standard density	
Temperature	LREAL	Temperature	
CalorificValue	LREAL	Calorific Value	

Output

Output Parameter	Data types	Description
BetaRatio	LREAL	Beta ratio (d/D) at flowing conditions
ElementActualBore	LREAL	Corrected bore/throat size
CValue	LREAL	Coefficient of discharge
SelectedDiffPressure	LREAL	Selected differential pressure
DiffTxInuse	INT	In-use DP transmitter
ExpFactor	LREAL	Expansibility factor
Pressure1Abs	LREAL	Upstream absolute pressure in Pa
Pressure1Guage	LREAL	Upstream gauge pressure
Pressure3Abs	LREAL	Fully recovered downstream absolute pressure
Pressure3Guage	LREAL	Fully recovered downstream gauge pressure
PipeActBore	LREAL	Corrected pipe size
Qh	LREAL	Energy flow
Qm	LREAL	Mass flow
Qs	LREAL	Standard volume flow
Qv	LREAL	Volume flow
Red	LREAL	Reynolds Number
RHO1	LREAL	In-use Upstream density
RHO1_1	LREAL	Upstream density derived from RHOTP1

Output Parameter	Data types	Description
RHO1_2	LREAL	Upstream density derived from RHOTP2
Temperature1	LREAL	Upstream temperature
VelApproachFactor	LREAL	Velocity of Approach factor
NumberIterations	LREAL	Number of Iterations for the last scan
ErrorCode	INT	Critical Error Code
WarningCode	INT	Warning Code
JT_COEFF	LREAL	Joule-Thomson coefficient in K/bar. If Version < 2003 Or Fluid is incompressible, value = NaN.

Information:

Compared with ISO5167_DUAL, ISO5167_DUAL_JT function block have another two extra pin inputs, ZP1T1 and ZP2T3, based on these two inputs, ISO5167_DUAL_JT adopts different algorithm to calculate RHO1_1 and RHO1_2.

In addition, ISO5167_DUAL_JT calculate T1 based on Joule-Thomson coefficient, finally flow rate can be figured out via ISO5167_DUAL_JT Function Block.

Input parameters range:

Input Parameter	Min Value	Max Value
PipeRefBore?	16.67	1200
PipeActBore?	12.5	1000
DensityInputComparTimeDelay	0	300
HiLimDP1	50.0	95.0
HiLimDP2	50.0	95.0
PipeRefTemperature	0	50
ElementRefTemperature	0	50
MaxItrations	6	12
DeadbandValueDP1	0	10.0
DeadbandValueDP2	0	10.0
DensityInputComparDB	0	10

Input Parameter	Min Value	Max Value
IsentropicExponent	1.0	5.0
InitialCValue	0.58	0.62
PermLossA	0	5.0
PermLossB	0	5.0
PipeCoefficient	0.000005	0.00005
ElementCoefficient	0.000005	0.00005
PrecisionLimit	0.000000001	0.000001

Error and Warning list

Critical Error Codes

Code	Description
1	Static pressure value is invalid (if a compressible fluid is selected.
2	Differential pressure value is invalid.
3	If Temp Comp is enabled, temperature value is invalid.
4	Density value is invalid.
5	Viscosity value is invalid.
6	Isentropic exponent value is invalid (if a compressible fluid is selected).
7	Iteration failed to converge.
8	Multiple DP transmitter configuration is invalid.
9	Pipe bore is invalid.
10	Beta is invalid.
11	P2 is invalid (if a compressible fluid is selected).
12	ZP1T1 is invalid**.
13	ZP2T3 is invalid**.
20	If standard volume is enabled, standard density is invalid.
21	If energy flow is enabled, CV by volume is invalid.
30	Configuration parameter invalid.

- Error 30 will appear if any of the parameter in "Key configuration parameter" marked as "Yes" in Inputs table is not configured or wrongly configured.
- If critical errors 1 to 11 and 30 occur, Qm and all derived values are set to NaN. If critical error 20 occurs, QS and QH are set to NaN. If critical error 21 occurs, QH is set to NaN.
- ** ZP1T1 errors and ZP2T3 errors are only relevant when ISO 5167:2003 is used.
- That is when the ISO5167_DUAL_JT function block is used, the fluid is compressible, the density position is downstream and the 2003 version of ISO5167 is used.

Warning codes

Code	Description
1	For a compressible fluid, P2/P1 ratio is too low.
2	Element bore is too small.
3	Pipe size is out of range for an orifice plate.
4	Pipe size is out of range for a venturi.
5	Orifice beta ratio is outside fiscal limits.
6	Orifice beta ratio is outside limits.
7	Venturi beta ratio is outside limits.
8	Orifice plate is outside Reynolds No limits.
9	Orifice plate is above fiscal Reynolds No limit.
10	Venturi is outside Reynolds No limits.
11	For dual density inputs, input 1 is invalid.
12	For dual density inputs, input 2 is invalid.
13	For dual density inputs, the deviation between the inputs is greater than the deadband.

CHAPTER

17 ISO6976LIB

The following ISO 6976 function block is available:

Function block	Short description
See ISO 6976 for more information.	ISO 6976 Block calculates Calorific value on a molar, mass and volumetric basis. Calorific value on a superior and inferior basis Calculation of values on an ideal and a real basis. Standard density and compressibility at the 15 deg C and 1.01325 bara conditions regardless of the chosen combustion/metering

ISO 6976

ISO 6976:1995 is an international standard covering the calculation for natural gas of calorific value, density, relative density and Wobbe Index from its composition. ISO 6976 is widely used in most areas of the world except North America.

Description

The basic function of this block is to calculate the following values using gas composition and the ISO 6976 definitive methods:

- Molar mass
- Ideal relative density
- Real relative density at 1.01325 bara and selected metering temperature.
- Ideal density at 1.01325 bara and selected metering temperature.
- Ideal density at 1.01325 bara and 15 deg C.
- Real density at 1.01325 bara and selected metering temperature.
- Real density at 1.01325 bara and 15 deg C.
- Compressibility at 1.01325 bara and 15 deg C.

- Compressibility at 1.01325 bara and metering temperature.
- CV on a molar basis – superior
- CV on a molar basis – inferior
- CV on a mass basis – superior
- CV on a mass basis – inferior
- Ideal CV on a volumetric basis – superior
- Ideal CV on a volumetric basis – inferior
- Real CV on a volumetric basis – superior
- Real CV on a volumetric basis – inferior
- Ideal Wobbe Index
- Real Wobbe Index

This function block does not support:

- Versions of ISO 6976 earlier than 1995.
- Alternative calculation methods as defined in ISO 6976.
- Normalization of gas composition.
- Calculation of line density. This is not supported by ISO 6976. If this value is required, AGA 8 Detailed must be used.

Explanation of Calorific Value (CV) Basis

CV can be calculated on a molar basis, a mass basis or a volumetric basis. In order to calculate CV on a mass basis or a volumetric basis, it is first necessary to calculate it on a molar basis. Also, in order to calculate Wobbe Index, it is necessary to calculate CV on a volumetric, superior basis.

Thus, it is always necessary to calculate CV on a molar, superior basis, CV on a volumetric, ideal, superior basis and CV on a volumetric, real, superior basis. Calculation of CV on a mass basis or on any form of inferior basis is optional and will only be done when those values are required.

CV on all 6 bases are exposed as outputs. Where the CV is not calculated, the value will be set to NaN. In most cases, only the CV on a volumetric, real, superior basis will be exposed and connected.

For CV on a molar basis or a mass basis, the ideal and real values are the same.

> **TIP:** Generally gas composition will be normalized prior to connecting it to the ISO 6976 Function block. This means that fractions are adjusted such that they sum to 1.0. Depending on how the gas analysis is done and to what extent on- line gas chromatographs are used, the normalization methods vary. Normalization is not provided by the function block and is beyond the scope of this document.

ISO6976 Components

ISO 6976 defines 58 components and provides complete or partial constant data for them. This function block input shows the name of the real components as seen in the below table. The mol fractions of the non-exposed components and indexes should be forced to 0.0.

Input

Input Parameter	Data types	Description
Methane	LREAL	Input mol fractions or mol percentage
Ethane	LREAL	Input mol fractions or mol percentage
Propane	LREAL	Input mol fractions or mol percentage
n_Butane	LREAL	Input mol fractions or mol percentage
i_Butane	LREAL	Input mol fractions or mol percentage
n_Pentane	LREAL	Input mol fractions or mol percentage
i_Pentane	LREAL	Input mol fractions or mol percentage
neo_Pentane	LREAL	Input mol fractions or mol percentage
n_Hexane	LREAL	Input mol fractions or mol percentage
Methylpentane_2	LREAL	Input mol fractions or mol percentage
Methylpentane_3	LREAL	Input mol fractions or mol percentage
Dimethylbutane_2_2	LREAL	Input mol fractions or mol percentage
Dimethylbutane_2_3	LREAL	Input mol fractions or mol percentage
n_Heptane	LREAL	Input mol fractions or mol percentage

Input Parameter	Data types	Description
n_Octane	LREAL	Input mol fractions or mol percentage
n_Nonane	LREAL	Input mol fractions or mol percentage
n_Decane	LREAL	Input mol fractions or mol percentage
Ethylene	LREAL	Input mol fractions or mol percentage
Propylene	LREAL	Input mol fractions or mol percentage
Butene1	LREAL	Input mol fractions or mol percentage
cis_2_Butene	LREAL	Input mol fractions or mol percentage
trans_2_Butene	LREAL	Input mol fractions or mol percentage
2-Methylpropene	LREAL	Input mol fractions or mol percentage
Pentene_1	LREAL	Input mol fractions or mol percentage
Propadiene	LREAL	Input mol fractions or mol percentage
Butadiene_1_2	LREAL	Input mol fractions or mol percentage
Butadiene_1_3	LREAL	Input mol fractions or mol percentage
Acetylene	LREAL	Input mol fractions or mol percentage
Cyclopentane	LREAL	Input mol fractions or mol percentage
Methylcyclopentane	LREAL	Input mol fractions or mol percentage
Ethylcyclopentane	LREAL	Input mol fractions or mol percentage
Cyclohexane	LREAL	Input mol fractions or mol percentage
Methylcyclohexane	LREAL	Input mol fractions or mol percentage
Ethylcyclohexane	LREAL	Input mol fractions or mol percentage
Benzene	LREAL	Input mol fractions or mol percentage
Toluene	LREAL	Input mol fractions or mol percentage
Ethylbenzene	LREAL	Input mol fractions or mol percentage
o_Xylene	LREAL	Input mol fractions or mol percentage
Methano	LREAL	Input mol fractions or mol percentage
Methanethiol	LREAL	Input mol fractions or mol percentage
Hydrogen	LREAL	Input mol fractions or mol percentage

Input Parameter	Data types	Description
Water	LREAL	Input mol fractions or mol percentage
Hydrogensulphide	LREAL	Input mol fractions or mol percentage
Ammonia	LREAL	Input mol fractions or mol percentage
Hydrogencyanide	LREAL	Input mol fractions or mol percentage
Carbonmonoxide	LREAL	Input mol fractions or mol percentage
Carbonyldisulphide	LREAL	Input mol fractions or mol percentage
Carbondisulphide	LREAL	Input mol fractions or mol percentage
Helium	LREAL	Input mol fractions or mol percentage
Neon	LREAL	Input mol fractions or mol percentage
Argon	LREAL	Input mol fractions or mol percentage
Nitrogen	LREAL	Input mol fractions or mol percentage
Oxygen	LREAL	Input mol fractions or mol percentage
Carbondioxide	LREAL	Input mol fractions or mol percentage
Sulphurdioxide	LREAL	Input mol fractions or mol percentage
Dinitrogenmonoxide	LREAL	Input mol fractions or mol percentage
Krypton	LREAL	Input mol fractions or mol percentage
Xenon	LREAL	Input mol fractions or mol percentage
MeteringTemperature	INT	The possible combinations are: 0 = 0/0, 1 = 15/0 ,2 = 25/0 ,3 = 15/15,4 = 20/22 25 = 25/20
InferiorCV_Values	INT	If inferior values are required, this needs to be set to 1
DensityScalingFactor	LREAL	The function block only calculates density in units of kg/Sm3. However, it is possible to use it for alternative metric units or non- metric units by scaling the output using the scale factor. For instance from, say, kg/Sm3 to lbs/Scuf. The scale factor must be set to 1.0 for no scaling

Input Parameter	Data types	Description
		but must be numeric and greater than zero.
CV_ MolarScalingFactor	LREAL	The function block only calculates CV on a molar basis in units of KJ/mol. However, it is possible to use it for alternative metric units or non- metric units by scaling the output using the scale factor. For instance from, say, KJ/mol to BTU/mol. The scale factor must be set to 1.0 for no scaling but must be numeric and greater than zero.
Input Basis	INT	Input compositions can either be in mol fraction terms or mol percentage terms. Mol fractions must sum to 1.0 and mol percentages must sum to 100.0. 0 = fraction , 1 = percent.
CV_onMassBasisMode	INT	If mass based CV is required, this must be set to 1.
CV_MassScalingFactor	LREAL	The function block only calculates CV on a mass basis in units of MJ/kg. However, it is possible to use it for alternative metric units or non- metric units by scaling the output using the scale factor. For instance from, say, MJ/kg to BTU/lb. The scale factor must be set to 1.0 for no scaling but must be numeric and greater than zero.
CV_ VolumeScalingFactor	LREAL	The function block only calculates CV on a volumetric basis in units of MJ/Sm3. However, it is possible to use it for alternative metric units or non-metric units by scaling the output using the scale factor. For instance from, say, MJ/Sm3 to BTU/scuf. The scale factor must be set to 1.0 for no scaling but must be numeric and greater than zero.

Output

Output Parameter	Data types	Description
ErrorCode	INT	Critical error code

Output Parameter	Data types	Description
CV_MolarBasisSuperior	LREAL	Calorific value on a molar basis, superior (KJ/mol)
CV_MolarBasisInferior	LREAL	Calorific value on a molar basis, inferior (KJ/mol)
CV_MassBasisSuperior	LREAL	Calorific value on a mass basis,superior (MJ/kg)
CV_MassBasisInferior	LREAL	Calorific value on a mass basis,inferior (MJ/kg)
CVIdeal_VolBasisSuperior	LREAL	Ideal calorific value on a volumetric basis, superior (MJ/Sm3)
CVReal_VolBasisSuperior	LREAL	Real calorific value on a volumetric basis, superior (MJ/Sm3)
CVIdeal_VolBasisInferior	LREAL	Ideal calorific value on a volumetric basis, inferior (MJ/Sm3)
CVReal_VolBasisInferior	LREAL	Real calorific value on a volumetric basis, inferior (MJ/Sm3)
MolarMass	LREAL	Molar mass
ComponentCount	INT	Number of components
RelativeDensityIdeal	LREAL	Relative density - ideal
RelativeDensityReal	LREAL	Relative density - real
ReferenceDensityIdeal	LREAL	Density at metering conditions – ideal (kg/Sm3)
ReferenceDensityReal	LREAL	Density at metering conditions – real (kg/Sm3)
StandardDensityIdeal	LREAL	Standard density - ideal (kg/Sm3)
StandardDensityReal	LREAL	Standard density - real (kg/Sm3)
SumOfComponentFracs	LREAL	Sum of component fractions
WobbeIdxIdeal	LREAL	Wobbe Index - ideal
WobbeIdxReal	LREAL	Wobbe Index - real
WarningCode	INT	Warning code
CompressAtRefCond	LREAL	Compressibility at metering conditions
CompressAtStdCond	LREAL	Compressibility at standard conditions

Information

ISO 6976 defines 58 components and provides complete or partial constant data for them.

This data is used in the calculation of molar mass, CV etc. Each component has an index number (1-58) and this index will be used to refer to the particular component. These same index numbers are also used to point at elements in data arrays. The relationship of indexes and component names is as follows:

Index	Component Name
1	Methane
2	Ethane
3	Propane
4	n-Butane
5	i-Butane
6	n-Pentane
7	i-Pentane
8	neo-Pentane
9	n-Hexane
10	2-Methylpentane
11	3-Methylpentane
12	2,2-Dimethylbutane
13	2,3 Dimethylbutane
14	n-Heptane
15	n-Octane
16	n-Nonane
17	n-Decane
18	Ethylene
19	Propylene
20	1-Butene
21	cis-2-Butene

Index	Component Name
22	trans-2-Butene
23	2-Methylpropene
24	1-Pentene
25	Propadiene
26	1,2 Butadiene
27	1,3 Butadiene
28	Acetylene
29	Cyclopentane
30	Methylcyclopentane
31	Ethylcyclopentane
32	Cyclohexane
33	Methylcyclohexane
34	Ethylcyclohexane
35	Benzene
36	Toluene
37	Ethylbenzene
38	o-Xylene
39	Methano
40	Methanethiol
41	Hydrogen
42	Water
43	Hydrogen sulphide
44	Ammonia
45	Hydrogen cyanide
46	Carbon monoxide
47	Carbonyl disulphide
48	Carbon disulphide

Index	Component Name
49	Helium
50	Neon
51	Argon
52	Nitrogen
53	Oxygen
54	Carbon dioxide
55	Sulphur dioxide
56	Dinitrogen monoxide
57	Krypton
58	Xenon

Error and Warning list

Critical Error Codes

Code	Description
1	An input component is invalid.
2	Inputs do not sum to 1.0 +/- 0.001.
3	Input configuration is invalid.
4	Absolute temperature is zero.
5	Molar mass is zero.
6	Reference compressibility factor is zero.
7	Standard compressibility factor is zero.
8	Relative gas density real is zero.
9	Relative gas density ideal is zero.

Invalid generally means one of the following:

- The input value is NaN.
- The input value is out of range.

Warning codes

Code	Description
1	Inputs do not sum to 1.0 +/- 0.0001.
2	Ethane mol fraction is > 0.15.
3	Water mol fraction is > 0.00005.
4	Nitrogen mol fraction is > 0.3.
5	Carbon dioxide mol fraction is > 0.15.
6	General component mol fraction is > 0.05.
7	Methane mol fraction is < 0.5.

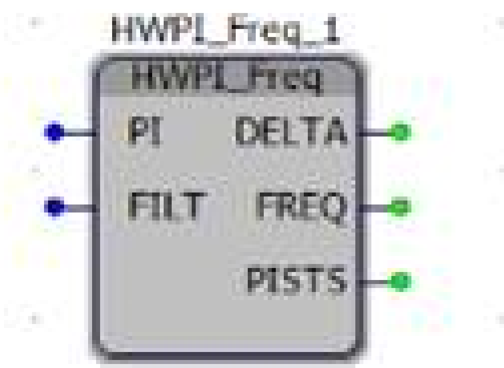
CHAPTER

18 HWPI_FREQ

Description

This function block is connected to a pulse input channel's counter and outputs frequency and pulse delta count (at 1 sec interval). The first order filter is used to filter variations and help to calculate frequency.

NOTE: The POU containing this block must be configured at the 50mS cycle task. The FREQ can be used to calculate the instantaneous flow rate of the meter.



Input

Input Parameter	Data types	Description
PI	UDINT	Counter of a pulse input channel.
FILT	REAL	First order filter time constant in minutes for smoothing calculated frequency output FREQ. Recommended value is 0.05 to 0.1.

Output

Output Parameter	Data types	Description
DELTA	LREAL	Delta counts in last 1 sec
FREQ	REAL	Calculated pulse frequency in Hz.
PISTS	STRING	Pulse input channel status message.

CHAPTER

19 MODBUS MASTER

The following Modbus function blocks are available:

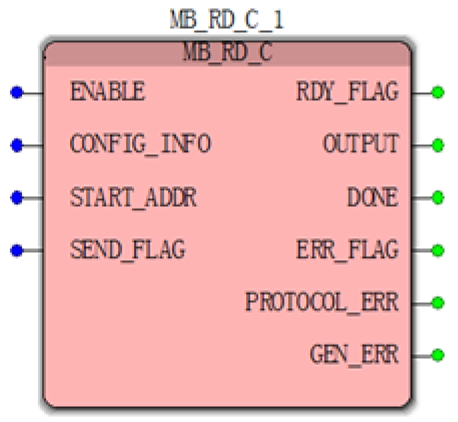
Function Blocks	Short Description
Read Multiple Coils	It is used to read multiple coils.
Read Multiple discrete Inputs	It is used to read multiple discrete inputs.
Read Multiple Holding Registers	It is used to read multiple holding registers.
Read Multiple Input Registers	It is used to read multiple input registers.
Read Single Coil	It is used to read a single coil.
Read Single Discrete Input	It is used to read single discrete input.
Read Single Holding Register	It is used to read a single holding register.
Read Single Input Register	It is used to read single input register.
Write Multiple Coils	It is used to write multiple coils.
Write Multiple Holding Registers	It is used to write multiple holding registers.
Write Single Coil	It is used to write a single coil.
Write Single Holding Register	It is used to write single holding register.

With these function blocks, you can read and write single coil, multiple coils, single discrete input, multiple discrete inputs, single input register, multiple input registers, single holding register, etc., as per Modbus protocol.

Related topics:

- Description of CONFIG_INFO
- Description of Input and Output Data Type
- Modbus Protocol Error Codes
- Endian Mode

Read Single Coil



Description

It is used to read a single coil.

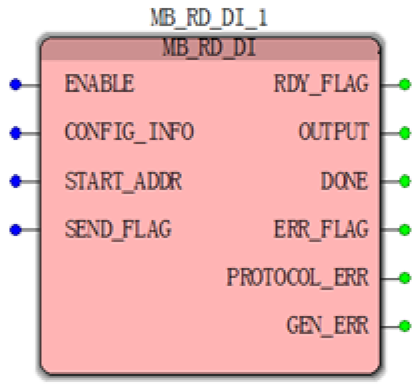
Input

Parameter	Data type	Description
ENABLE	BOOL	Enable: If TRUE, the FB is enabled and workable.
CONFIG_ INFO	User defined data type	This is a structure provided by Honeywell. Modbus related information is included. See Description of CONFIG_INFO for more information.
START_ ADDR	UINT	The first Modbus register address to read. Function code is not included in the address.
SEND_ FLAG	BOOL	If SEND_FLAG is true and RDY_FLAG is true, function blocks will send the request. RDY_FLAG is TRUE means last communication is finished. Before the last communication is finished, even if the SEND_FLAG is true, the request won't be sent.

Output

Parameter	Data type	Description
RDY_FLAG	BOOL	True: last communication is finished. FB is ready for the next communication. False: command request is being sent or received.
OUTPUT	BOOL	Output: 1: true, 0: OFF
DONE	BOOL	Indicates that the response is received from responder device.
ERR_FLG	BOOL	Will be set to TRUE if there is either a general error or a protocol error.
PROTOCOL_ERR	USINT	Error numbers defined by Modbus protocol. See Modbus Protocol Error Codes for more information.
GEN_ERR	USINT	General error code: 0: Communication succeeded. 1: The input parameter is invalid. 2: Response timeout 3: Controller internal time out (IPC timeout). 4: Invalid request

Read Single Discrete Input



Description

It is used to read single discrete input.

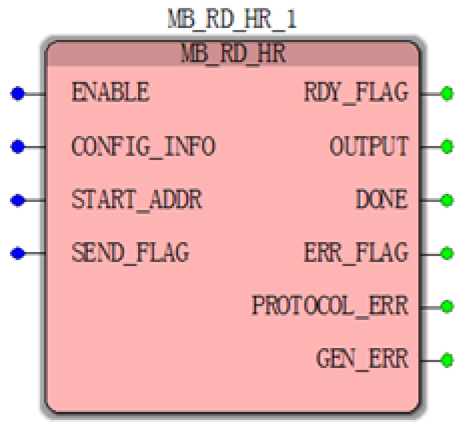
Input

Parameter	Data type	Description
ENABLE	BOOL	Enable: If TRUE, the FB is enabled and workable.
CONFIG_ INFO	User defined data type	This is a structure provided by Honeywell. Modbus related information is included. See Description of CONFIG_INFO for more information.
START_ ADDR	UINT	The first Modbus register address to read. Function code is not included in the address.
SEND_ FLAG	BOOL	If SEND_FLAG is true and RDY_FLAG is true, function blocks will send the request. RDY_FLAG is TRUE means last communication is finished. Before last communication is finished, even if SEND_FLAG is true the request won't be sent.

Output

Parameter	Data type	Description
RDY_FLAG	BOOL	True: last communication is finished. FB is ready for the next communication. False: command request is being sent or received.
OUTPUT	Array of BOOL	User defined data type: array of BOOL. The size of the array should be equal to the number of the registers to read.
OUTPUT	BOOL	Output: 1: true, 0: OFF
DONE	BOOL	Indicates that the response is received from a responder device.
ERR_FLG	BOOL	Will be set to TRUE if there is either a general error or a protocol error.
PROTOCOL_ERR	USINT	Error numbers defined by Modbus protocol. See Modbus Protocol Error Codes for more information.
GEN_ERR	USINT	General error code: 0: Communication succeeded. 1: The input parameter is invalid. 2: Response timeout 3: Controller internal time out (IPC timeout). 4: Invalid request

Read Single Holding Register



Description

It is used to read a single holding register.

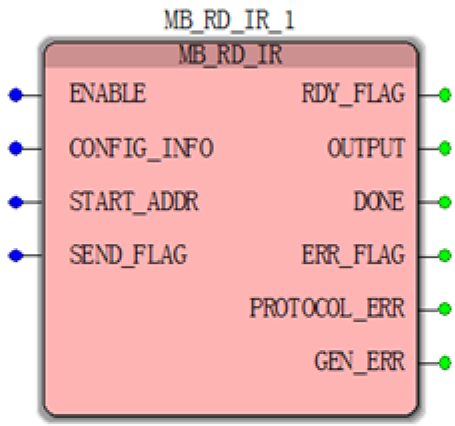
Input

Parameter	Data type	Description
ENABLE	BOOL	Enable: If TRUE, the FB is enabled and workable.
CONFIG_ INFO	User defined data type	This is a structure provided by Honeywell. Modbus related information is included. See Description of CONFIG_INFO for more information.
START_ ADDR	UINT	The Modbus register address to read. Function code is not included in the address.
SEND_ FLAG	BOOL	If SEND_FLAG is true and RDY_FLAG is true, function blocks would send the request. RDY_FLAG is TRUE means the last communication is finished. Before the last communication is finished, even if the SEND_FLAG is true the request won't be sent.

Output

Parameter	Data type	Description
RDY_FLAG	BOOL	True: last communication is finished. FB is ready for the next communication. False: command request is being sent or received.
OUTPUT	UINT	16 bit data read from the START_ADDR
DONE	BOOL	Indicates that the response is received from responder device.
ERR_FLG	BOOL	Will be set true if there is either a general error or a protocol error.
PROTOCOL_ERR	USINT	Error numbers defined by Modbus protocol. See Modbus Protocol Error Codes for more information.
GEN_ERR	USINT	General error code: 0: Communication succeeded. 1: The input parameter is invalid. 2: Response timeout 3: Controller internal time out (IPC timeout). 4: Invalid request

Read Single Input Register



Description

It is used to read single input register.

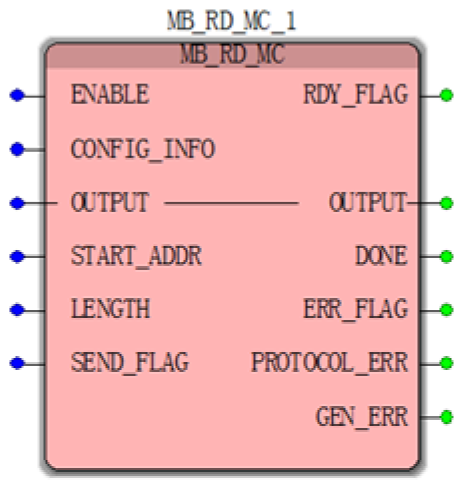
Input

Parameter	Data type	Description
ENABLE	BOOL	Enable: If TRUE, the FB is enabled and workable.
CONFIG_ INFO	User defined data type	This is a structure provided by Honeywell. Modbus related information is included. See Description of CONFIG_INFO for more information.
START_ ADDR	UINT	The Modbus register address to read. Function code is not included in the address.
SEND_ FLAG	BOOL	If SEND_FLAG is true and RDY_FLAG is true, function blocks would send the request. RDY_FLAG is TRUE means the last communication is finished. Before last communication is finished, even if SEND_FLAG is true, the request won't be sent.

Output

Parameter	Data type	Description
RDY_FLAG	BOOL	True: last communication is finished. FB is ready for the next communication. False: command request is being sent or received.
OUTPUT	UINT	16bit Data read from the START_ADDR
DONE	BOOL	Indicates that the response is received from responder device.
ERR_FLG	BOOL	Will be set to TRUE if there is either a general error or a protocol error.
PROTOCOL_ERR	USINT	Error numbers defined by Modbus protocol. See Modbus Protocol Error Codes for more information.
GEN_ERR	USINT	General error code: 0: Communication succeeded. 1: The input parameter is invalid. 2: Response timeout 3: Controller internal time out(IPC timeout). 4: Invalid request

Read Multiple Coils



Description

It is used to read multiple coils.

Input

Parameter	Data type	Description
ENABLE	BOOL	Enable: If TRUE, the FB is enabled and workable.
CONFIG_ INFO	User defined data type	This is a structure provided by Honeywell. Modbus related information is included. See Description of CONFIG_INFO for more information.
START_ ADDR	UINT	The first Modbus register address to read. Function code is not included in the address.
LENGTH	UINT	The number of registers to read, ranging from 1 to 2000.
SEND_ FLAG	BOOL	If SEND_FLAG is true and RDY_FLAG is true, function blocks will send the request. RDY_FLAG is TRUE means the last communication finished. Before the last communication is finished, even if SEND_FLAG is true, the request won't be sent.

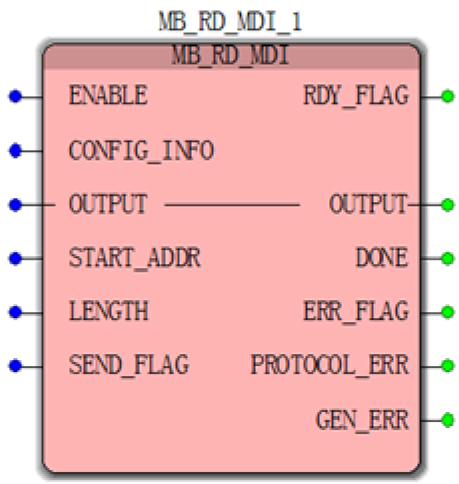
Output

Parameter	Data type	Description
RDY_FLAG	BOOL	True: last communication is finished. FB is ready for the next communication. False: command request is being sent or received.

OUTPUT	Array of BOOL	User defined data type: array of bool. The size of the array should be equal to the number of the registers to read. Define a data type as shown below:
		
		`TYPE`
		`Variable Name: array[1..LENGTH] of BOOL;`
		`END_TYPE`

DONE	BOOL	Indicates that the response is received from a responder device.
ERR_FLG	BOOL	Will be set to TRUE if there is either a general error or a protocol error.
PROTOCOL_ERR	USINT	Error numbers defined by Modbus protocol. See Modbus Protocol Error Codes for more information.
GEN_ERR	USINT	General error code: 0: Communication succeeded. 1: The input parameter is invalid. 2: Response timeout 3: Controller internal time out (IPC timeout). 4: Invalid request

Read Multiple Discrete Inputs



Description

It is used to read multiple discrete inputs.

Input

Parameter	Data type	Description
ENABLE	BOOL	Enable: If TRUE, the FB is enabled and workable.
CONFIG_ INFO	User defined data type	This is a structure provided by Honeywell. Modbus related information is included. See Description of CONFIG_INFO for more information.
START_ ADDR	UINT	The first Modbus register address to read. Function code is not included in the address.
LENGTH	UINT	The number of registers to read, ranging from 1 to 2000.
SEND_ FLAG	BOOL	If SEND_FLAG is true and RDY_FLAG is true, function blocks will send the request. RDY_FLAG is TRUE means the last communication is finished. Before the last communication is finished, even if SEND_FLAG is true, the request won't be sent.

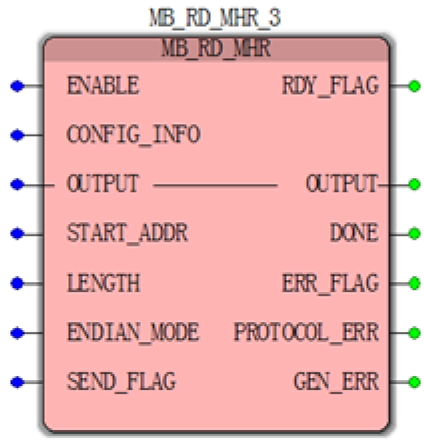
Output

Parameter	Data type	Description
RDY_FLAG	BOOL	True: last communication is finished. FB is ready for the next communication. False: command request is being sent or received.

OUTPUT	Array of BOOL	User defined data type: array of bool. The size of the array should be equal to the number of the registers to read. Define a data type as shown here:
		
		`TYPE`
		`          Variable Name: array[1..LENGTH] of BOOL;`
		`END_TYPE`

DONE	BOOL	Indicates that the response is received from a responder device.
ERR_FLG	BOOL	Will be set to TRUE if there is either a general error or a protocol error.
PROTOCOL_ERR	USINT	Error numbers defined by Modbus protocol. See Modbus Protocol Error Codes for more information.
GEN_ERR	USINT	General error code: 0: Communication succeeded. 1: The input parameter is invalid. 2: Response timeout 3: Controller internal time out (IPC timeout). 4: Invalid request

Read Multiple Holding Registers



Description

It is used to read multiple holding registers.

Input

Parameter	Data type	Description
ENABLE	BOOL	Enable: If TRUE, the FB is enabled and workable.
CONFIG_INFO	User defined data type	This is a structure provided by Honeywell. Modbus related information is included. See Description of CONFIG_INFO for more information.
START_ADDR	UINT	The first Modbus register address to read. Function code is not included in the address.
LENGTH	UINT	The number of registers to read, ranging from 1 to 125.
ENDIAN_MODE	USINT	Endian mode is required for reading/writing 32bit and 64 bit variables. As Modbus always use big Endian to transceive data, there is no need to set the Endian mode for 16-bit data. 1: little Endian mode for 32 bit data 2: byte-swapped little Endian mode for 32 bit data

Parameter	Data type	Description
		3: big Endian mode for 32 bit data 4: byte-swapped big Endian mode for 32 bit data 5: little Endian mode for 64 bit data 6: byte-swapped little Endian mode for 64 bit data 7: big Endian mode for 64 bit data 8: byte-swapped big Endian mode for 64 bit data See Endian Mode for more information.
SEND_FLAG	BOOL	If SEND_FLAG is true and RDY_FLAG is true, function blocks would send the request. RDY_FLAG is TRUE means the last communication is finished. Before the last communication is finished, even if the SEND_FLAG is true, the request won't be sent.

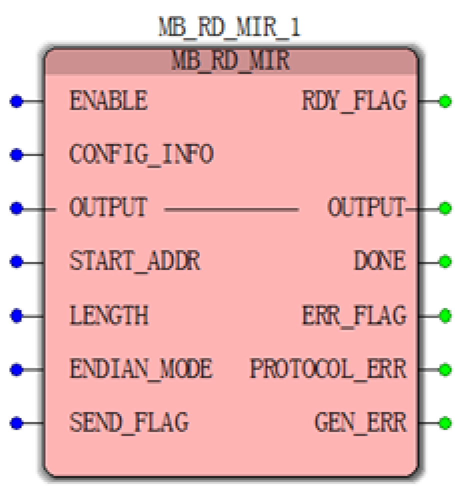
Output

Parameter	Data type	Description
RDY_FLAG	BOOL	True: last communication is finished. FB is ready for the next communication. False: command request is being sent or received.

OUTPUT	Array of INT, UINT, DINT, UDINT, LINT, REAL or LREAL;	User defined data type. The size of the array should be equal to the number of the registers to read multiplied by the register size.
		
		The end user should define a data type as shown here:
		
		`TYPE`
		`    Variable Name: array[1..LENGTH] of INT/UINT/DINT/UDINT/LINT/REAL/LREAL;`
		`END_TYPE`
		
		The end user can read the data of a specific register by using the suffix.
		
		TIP: This block supports reading data from a Modbus

Parameter	Data type	Description
		responder configured with non-standard register sizes (For example: 32-bit or 64-bit registers).
DONE	BOOL	Indicates that the response is received from responder device.
ERR_FLG	BOOL	Will be set to TRUE if there is either a general error or a protocol error.
PROTOCOL_ERR	USINT	Error numbers defined by Modbus protocol. See Modbus Protocol Error Codes for more information.
GEN_ERR	USINT	General error code: 0: Communication succeeded. 1: The input parameter is invalid. 2: Response timeout 3: Controller internal time out (IPC timeout). 4: Invalid request

Read Multiple Input Registers



Description

It is used to read multiple input registers.

Input

Parameter	Data type	Description
ENABLE	BOOL	Enable: If TRUE, the FB is enabled and workable.
CONFIG_ INFO	User defined data type	This is a structure provided by Honeywell. Modbus related information is included. See Description of CONFIG_INFO for more information.
START_ ADDR	UINT	The first Modbus register address to read. Function code is not included in the address.
LENGTH	UINT	The number of registers to read, ranging from 1 to 125.
ENDIAN_ MODE	USINT	Endian mode is required for reading/writing 32bit and 64 bit variables. As Modbus always use big Endian to transceive data, there is no need to set the Endian mode for 16-bit data. 1: little Endian mode for 32 bit data 2: byte-swapped little Endian mode for 32 bit data 3: big Endian mode for 32 bit data 4: byte-swapped big Endian mode for 32 bit data 5: little Endian mode for 64 bit data 6: byte-swapped little Endian mode for 64 bit data 7: big Endian mode for 64 bit data 8: byte-swapped big Endian mode for 64 bit data See Endian Mode for more information.
SEND_ FLAG	BOOL	If SEND_FLAG is true and RDY_FLAG is true, function blocks would send the request. RDY_FLAG is TRUE means the last communication is finished. Before the last communication is finished, even if the SEND_FLAG is true, the request won't be sent.

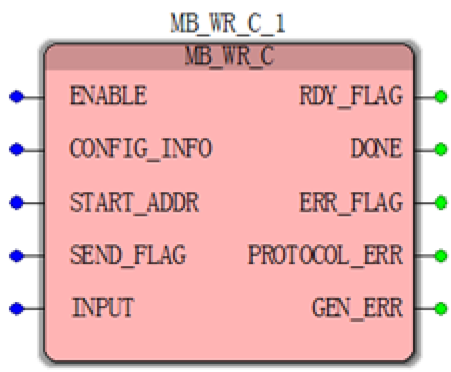
Output

Parameter	Data type	Description
RDY_FLAG	BOOL	True: last communication is finished. FB is ready for the next communication. False: command request is being sent or received.

OUTPUT	INT, UINT, DINT, UDINT, LINT, REAL or LREAL;	User defined data type. The size of the array should be equal to the number of the registers to read multiplied by the register size.
		
		The end user should define a data type as shown here:
		
		`TYPE`
		`          array[1..LENGTH] of`
		`INT/UINT/DINT/UDINT/LINT/REAL/LREAL;`
		`END_TYPE`
		
		The end user can read the data of a specific register by using the suffix.
		
		TIP: This block supports reading data from a Modbus responder configured with non-standard register sizes (For example: 32-bit or 64-bit registers).

DONE	BOOL	Indicates that the response is received from responder device.
ERR_FLG	BOOL	Will be set to TRUE if there is either a general error or a protocol error.
PROTOCOL_ERR	USINT	Error numbers defined by Modbus protocol. See Modbus Protocol Error Codes for more information.
GEN_ERR	USINT	General error code: 0: Communication succeeded. 1: The input parameter is invalid. 2: Response timeout 3: Controller internal time out (IPC timeout). 4: Invalid request

Write Single Coil



Description

It is used to write a single coil.

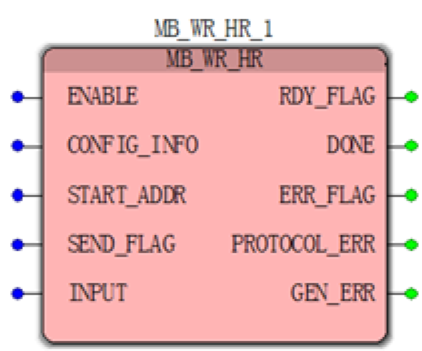
Input

Parameter	Data type	Description
ENABLE	BOOL	Enable: If TRUE, the FB is enabled and workable.
CONFIG_ INFO	User defined data type	This is a structure provided by Honeywell. Modbus related information is included. See Description of CONFIG_INFO for more information.
START_ ADDR	UINT	The Modbus register address to read. Function code is not included in the address.
SEND_ FLAG	BOOL	If SEND_FLAG is true and RDY_FLAG is true, function blocks would send the request. RDY_FLAG is TRUE means the last communication is finished. Before the last communication is finished, even if the SEND_FLAG is true, the request won't be sent.
INPUT	BOOL	1: ON 0: OFF

Output

Parameter	Data type	Description
RDY_FLAG	BOOL	True: last communication is finished. FB is ready for the next communication. False: command request is being sent or received.
DONE	BOOL	Indicates that the response is received from responder device.
ERR_FLG	BOOL	Will be set to TRUE if there is either a general error or a protocol error.
PROTOCOL_ERR	USINT	Error numbers defined by Modbus protocol. See Modbus Protocol Error Codes for more information.
GEN_ERR	USINT	General error code: 0: Communication succeeded. 1: The input parameter is invalid. 2: Response timeout 3: Controller internal time out (IPC timeout). 4: Invalid request

Write Single Holding Register



Description

It is used to write single holding register.

Input

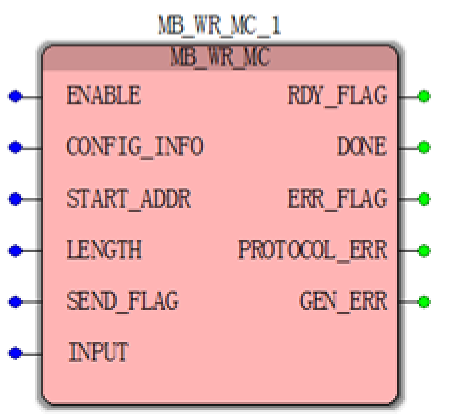
Parameter	Data type	Description
ENABLE	BOOL	Enable: If TRUE, the FB is enabled and workable.
CONFIG_INFO	User defined data type	This is a structure provided by Honeywell. Modbus related information is included. See Description of CONFIG_INFO for more information.
START_ADDR	UINT	The Modbus register address to read. Function code is not included in the address.
SEND_FLAG	BOOL	If SEND_FLAG is true and RDY_FLAG is true, function blocks would send the request. RDY_FLAG is TRUE means the last communication is finished. Before the last communication is finished, even if the SEND_FLAG is true, the request won't be sent.
INPUT	UINT	16 bit input data of START_ADDR register

Output

Parameter	Data type	Description
RDY_FLAG	BOOL	True: last communication is finished. FB is ready for the next communication. False: command request is being sent or received.
DONE	BOOL	Indicates that the response is received from responder device.
ERR_FLG	BOOL	Will be set to TRUE if there is either a general error or a protocol error.
PROTOCOL_ERR	USINT	Error numbers defined by Modbus protocol. See Modbus Protocol Error Codes for more information.

Parameter	Data type	Description
GEN_ERR	USINT	General error code: 0: Communication succeeded. 1: The input parameter is invalid. 2: Response timeout 3: Controller internal time out (IPC timeout). 4: Invalid request

Write Multiple Coils



Description

It is used to write multiple coils.

Input

Parameter	Data type	Description
ENABLE	BOOL	Enable: If TRUE, the FB is enabled and workable.
CONFIG_ INFO	User defined data	This is a structure provided by Honeywell. Modbus related information is included. See Description of CONFIG_INFO for more information.

Parameter	Data type	Description
	type	
START_ADDR	UINT	The first Modbus register address to read. Function code is not included in the address.
LENGTH	UINT	The number of registers to write, ranging from 1 to 1968.
SEND_FLAG	BOOL	If SEND_FLAG is TRUE and RDY_FLAG is true, function blocks would send the request. RDY_FLAG is TRUE means the last communication is finished. Before the last communication is finished, even if the SEND_FLAG is true, the request won't be sent.

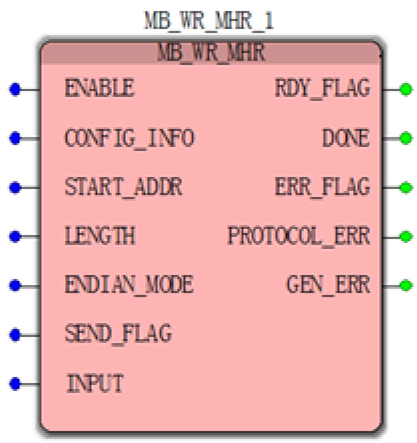
INPUT	Array of BOOL	User defined data type: array of bool. The size of the array should be equal to the number of the registers to read. The end user should define a data type as shown here:
		`TYPE`
		`Variable Name: array[1..LENGTH] of BOOL;`
		`END_TYPE`
		Use the suffix to set the status of a specific register.

Output

Parameter	Data type	Description
RDY_FLAG	BOOL	True: last communication is finished. FB is ready for the next communication. False: command request is being sent or received.
DONE	BOOL	Indicates that the response is received from responder device.
ERR_FLG	BOOL	Will be set to TRUE if there is either a general error or a protocol error.
PROTOCOL_ERR	USINT	Error numbers defined by Modbus protocol. See Modbus Protocol Error Codes for more information.
GEN_ERR	USINT	General error code:

Parameter	Data type	Description
		0: Communication succeeded. 1: The input parameter is invalid. 2: Response timeout 3: Controller internal time out (IPC timeout). 4: Invalid request

Write Multiple Holding Registers



Description

It is used to write multiple holding registers.

Input

Parameter	Data type	Description
ENABLE	BOOL	Enable: If TRUE, the function block is enabled and workable.
CONFIG_INFO	User defin	This is a structure provided by Honeywell. Modbus related information is included. See Description of CONFIG_INFO for more information.

Parameter	Data type	Description
	ed data type	
START_ADDR	UINT	The first Modbus register address to read. Function code is not included in the address.
LENGTH	UINT	The number of registers to write, ranging from 1 to 123.
ENDIAN_MODE	USINT	Endian mode is required for reading/writing 32bit and 64 bit variables. As Modbus always use big Endian to transceive data, there is no need to set the Endian mode for 16-bit data. 1: little Endian mode for 32 bit data 2: byte-swapped little Endian mode for 32 bit data 3: big Endian mode for 32 bit data 4: byte-swapped big Endian mode for 32 bit data 5: little Endian mode for 64 bit data 6: byte-swapped little Endian mode for 64 bit data 7: big Endian mode for 64 bit data 8: byte-swapped big Endian mode for 64 bit data See Endian Mode for more information.
SEND_FLAG	BOOL	If SEND_FLAG is true and RDY_FLAG is true, function blocks would send the request. RDY_FLAG is TRUE means the last communication is finished. Before the last communication is finished, even if the SEND_FLAG is true, the request won't be sent.

INPUT	Array of INT, UINT, DINT, UDINT, LINT, REA	User defined data type. The size of the array depends on the number of the registers to write:
		Size of (array) * size of (element of array) / size of (UINT) = LENGTH.
		The end user should define a data type as shown here:
		`TYPE`
		`    Variable Name: array[1..LENGTH] of INT/UINT/DINT/UDINT/LINT/REAL/LREAL;`

Parameter	Data type	Description
	L, or LREAL	END_TYPE Use the suffix to read the data of a specific register.

Output

Parameter	Data type	Description
RDY_FLAG	BOOL	True: last communication is finished. The function block is ready for the next communication. False: command request is being sent or received.
DONE	BOOL	Indicates that the response is received from responder device.
ERR_FLG	BOOL	Will be set to TRUE if there is either a general error or a protocol error.
PROTOCOL_ERR	USINT	Error numbers defined by Modbus protocol. See Modbus Protocol Error Codes for more information.
GEN_ERR	USINT	General error code: 0: Communication succeeded. 1: The input parameter is invalid. 2: Response timeout 3: Controller internal time out (IPC timeout). 4: Invalid request

Description of CONFIG_INFO

The CONFIG_INFO pin defined in the function blocks is to input all the configuration information for the Modbus master.

There are three types of communication between Modbus master and Modbus responder: serial communication of ControlEdge 2020 controllers using RS232 or RS485, Ethernet communication and serial communication of ControlEdge 900 Controllers. Accordingly three types of data structures are defined for CONFIG_INFO.

- For serial communication of ControlEdge 2020 controllers, the data structure is defined as:

```
TYPE
        MB_CONFIG_INFO_COM:
        STRUCT
                    MB_[1]RESPONDER_ID: USINT;
                    PORT_NUM:      USINT;
                    RETRIES:          USINT;
                    TIMEOUT:          UDINT;
        END_STRUCT;
END_TYPE
```

- For Ethernet communication, the data structure is defined as:

```
TYPE
        MB_CONFIG_INFO_ETH:
        STRUCT
                    MB_RESPONDER_ID:
USINT;
                    PORT_NUM:
USINT;
                    RETRIES:
USINT;
                    TIMEOUT:
UDINT;
                    TCP_PORT_NUM:      UINT;
                    IP_ADDR:
STRING;
        END_STRUCT;
END_TYPE
```

- For serial communication of ControlEdge 900 Controllers, the data structure is defined as:

```
TYPE
        MB_CONFIG_INFO_ECOM:
```

[1]Adaption of new inclusive terminologies.

```
        STRUCT
                MB_RESPONDER_ID: USINT;
                PORT_NUM:         USINT;
                RETRIES:          USINT;
                TIMEOUT:          UDINT;
                RACK_NUM:         UDINT;
                SLOT_NUM:         UDINT;
        END_STRUCT;
END_TYPE
```

See the following table for the parameter descriptions:

Parameter	Data type	Description
MB_ [1]RESPONDER_ ID	USINT	Modbus responder ID: valid arrange: 1~247.
PORT_NUM	USINT	The physical interface of serial port: 1. RS232 port 1 2. RS232 port 2 3. RS485 port 1 4. RS485 port 2 5. reserved 6. reserved The physical interface of Ethernet port: 1. Ethernet port 1 2. Ethernet port 2 3. reserved 4. reserved
RETRIES	USINT	Retry times before it is failed.
TIMEOUT	UDINT	Timeout unit: millisecond. The minimal timeout is 500 ms. If the end-user gives a number less than 500, the FB would send the default

[1]Adaption of new inclusive terminologies.

Parameter	Data type	Description
		timeout value instead.
TCP_PORT_ NUM	UINT	TCP/IP port number of the Modbus responder device
IP_ADDR	STRING	The IP address of the Modbus responder device. Example: ‘192.168.0.100’
RACK_NUM	UDINT	The rack number of the serial port: • 0 for local CPM, • 1 to 99 for remote EPM
SLOT_NUM	UDINT	The slot number of the serial port, 1 to 12 are available

Description of Input and Output Data Type

Modbus supports reading and writing multiple consecutive registers. In these cases, the input or output is defined as an array.

- For reading and writing coils and discrete inputs, array of BOOL is defined.

  Set or retrieve the data value by using the suffix. For example: there are 10 coils to read, the output array COIL_OUT can be defined as array [1...10] of BOOL, reading the status of the fifth register could be COIL_OUT [5].

- For reading and writing input registers and holding registers, multiple array types can be defined: INT, UINT, DINT, UDINT, REAL, LREAL or LINT.

  Set or retrieve the data value by using the suffix. For example: there are 3 LREAL variables , or in other words, 12 holding registers to read, the output array LREAL_OUT can be defined as array[1..3] of LREAL, reading the value of the second register could be LREAL_OUT[2]. In this case, the Endian mode is involved.

Modbus Protocol Error Codes

Refer to the following table for Modbus Protocol Error Codes:

Error Code	Item	Description
0	success	N/A
65	I/O error	The underlaying I/O system reported an error.
69	Connection broken	Signals that the TCP/IP connection is closed by the remote peer or broken.
129	checksum error	N/A
130	invalid frame error	Signals that a received frame does not correspond either by structure or content to the specification or does not match a previously sent query frame. A poor data link typically causes this error.
131	Invalid reply error	Signals that a received reply does not correspond to the specification
132	reply timeout error	Signals that a fieldbus data transfer timed out. This can occur if the responder device does not reply in time or does not reply at all. A wrong unit address will also cause this error. On some occasions, this exception is also produced if the characters received don't constitute a complete frame.
133	send timeout error	Signals that a fieldbus data send timed out. This can only occur if the handshake lines are not properly set.
134	Invalid responder[1] ID	Signals that a fieldbus data is not for me.
161	illegal function response	Signals that an illegal Function exception response was received. This exception response is sent by a responder device instead of a normal response message if a master sent a Modbus function not supported by the responder device.
162	illegal address response	Signals that an illegal Data Address exception response was received. This exception response is sent by a responder device instead of a normal response message if a master queried an invalid or non-existing data address.
163	illegal value response	Signals that an illegal Value exception response was received. This exception response is sent by a responder device instead of a normal response message if a master sent a data value

[1]Adaption of new inclusive terminologies.

Error Code	Item	Description
		that is not an allowed value for the responder device.
164	failure response	Signals that a Responder Device Failure exception response (code 04) was received. This exception response is sent by a responder device instead of a normal response message if an unrecoverable error occurred while processing the requested action. This response is also sent if the request would generate a response whose size exceeds the allowable data size.
165	Acknowledge	Responder has accepted request and is processing it, but a long duration of time is required. This response is returned to prevent a timeout error from occurring in the master. Master can next issue a Poll Program Complete message to determine whether processing is completed.
166	Responder Device Busy	Responder is engaged in processing a long-duration command. Master should retry later.
167	Negative Acknowledge	Responder cannot perform the programming functions. Master should request diagnostic or error information from responder.
168	Memory Parity Error	Responder detected a parity error in memory. Master can retry the request, but service may be required on the responder device.
170	Gateway Path Unavailable	Specialized use in conjunction with gateways, indicates that the gateway was unable to allocate and an internal communication path from the input port to the output port for processing the request. Usually means that the gateway is misconfigured or overloaded.
171	Gateway Target Device Failed to Respond	Specialized use in conjunction with gateways, indicates that no response was obtained from the target device. Usually mean that the device is not present on the network.

Endian Mode

Modbus protocol supports 16bit data only. If there are 32bit or 64bit variables, 2 or 4 consecutive registers should be used to read the data value. In these cases, the Endian mode may be involved due to the different Endian modes in Modbus responder devices.

See the following table for the concept of Endian modes used in Modbus function blocks:

Endian mode	Description
Little endian	Lower registers contain lower bits and higher registers contain higher bits. The order is on a register basis. Inside each register, the more significant byte is always at the first place as defined by the Modbus protocol.
Big endian	Lower registers contain higher bits and higher registers contain lower bits. The order is on a register basis. Inside each register, the more significant byte is always at the first place as defined by the Modbus protocol.
Byte-swapped	The two bytes inside each register would be swapped.

See the following table for the valid Endian modes:

Valid Endian mode	Description
1	little Endian mode for 32 bit data
2	byte-swapped little Endian mode for 32 bit data
3	big Endian mode for 32 bit data
4	byte-swapped big Endian mode for 32 bit data
5	little Endian mode for 64 bit data
6	byte-swapped little Endian mode for 64 bit data
7	big Endian mode for 64 bit data
8	byte-swapped big Endian mode for 64 bit data

CHAPTER

20 USER DEFINED PROTOCOL

The following user defined protocol function blocks are available:

Function Block	Short Description
COM_RECV	This function block is used to received user defined data from the target device.
COM_SEND	This function block is used to send user defined data to the target device.

Related topics:

Topic	Short Description
User Defined Protocol Error Codes	See User Defined Protocol Error Codes for more information.

COM_SEND

This function block is used to send user defined data to the target device.

Input

Parameter	Data type	Description
ENABLE	BOOL	Enable: If TRUE, the function block is enabled and workable.
RACK	USINT	Rack number of expanded communication module. Default as 0. • 0: Local rack • 1~99: Remote rack. **TIP:** This pin is only required for ControlEdge 900 Platform, and it is not applicable for ControlEdge 2020 Platform.

Parameter	Data type	Description
PORT	USINT	The physical interface of serial port: 1. RS232 port 1 2. RS232 port 2 3. RS485 port 1 4. RS485 port 2 5. reserved 6. reserved
IOM	USINT	Module number of expanded communication module. Default as 0. For Control Edge RTU: • 0: Controller; • 1~30: Expanded communication module; For Control Edge PLC: 1 to 12 are available.
SEND_ FLAG	BOOL	If SEND_FLAG is true and RDY_FLAG is true, the function block would send the request. RDY_FLAG is TRUE means last communication is finished. Before last communication is finished, even if SEND_FLAG is true the request won't be sent.
DATA	Array of USINT,UINT, UDINT, LINT, REAL or LREAL;	User defined data type. The size of the array depends on the number of the registers to read. the end user should define a data type as shown below: TYPE VariableName: array[1..LENGTH] of UINT/USINT/UDINT/LINT/REAL/LREAL; END_TYPE The end user can read the data of a specific register by using the suffix.
LENGTH	UINT	Maximum number of bytes to be sent. The DATA parameter determines the length of the data to be sent. Default = 0; The maximum number is 1024 bytes.

Output

Parameter	Data type	Description
RDY_FLAG	BOOL	True: last communication is finished.The Function Block is ready for the next communication. False: The command request is being sent or received.
DONE	BOOL	It Indicates that the response is receivedfrom the device.
ERR_FLAG	BOOL	It would be set true if there is an error.
PROTOCOL_ERR	USINT	Error numbers defined by serial protocol 0: success For other errors, see User Defined Protocol Error Codes for more information.
GEN_ERR	USINT	0: Communication succeeded For other errors, see User Defined Protocol Error Codes for more information.

COM_RECV

This function block is used to receive user defined data from the target device.

Input

Parameter	Data type	Description
ENABLE	BOOL	Enable: If TRUE, the function block is enabled and workable.
RACK	USINT	Rack number: 0: local rack; 1~99: remote rack. **TIP:** This pin is only required for ControlEdge 900 Platform, and it is not applicable for ControlEdge 2020 Platform.

Parameter	Data type	Description
IOM	USINT	Module number of expanded communication module. Default as 0. For ControlEdge RTU: • 0: Controller; • 1~30: Expanded communication module; For ControlEdge PLC: 1 to 12 are available.
PORT	USINT	The physical interface of serial port: 1. RS232 port 1 2. RS232 port 2 3. RS485 port 1 4. RS485 port 2 5. reserved 6. reserved
MAXLENGTH	UINT	Used to define the size of receiving buffer. The maximum size is 1024 bytes.

Output

Parameter	Data type	Description
RDY_FLAG	BOOL	True: last communication is finished.The Function Block is ready for the next communication. False: The command request is being sent or received.
DONE	BOOL	It Indicates that the response is received from the device.
ERR_FLAG	BOOL	It would be set true if there is an error.
PROTOCOL_ ERR	USINT	Error numbers defined by serial protocol 0: success For other errors, see User Defined Protocol Error Codes for more information.
GEN_ERR	USINT	0: Communication succeeded For other errors, see User Defined Protocol Error Codes for more information.
LENGTH	UINT	Maximum number of bytes to be received. The DATA parameter determines the length of the data to be received. Default = 0; The maximum number is 1024 bytes.

Input and Output

Parameter	Data type	Description
DATA	Array of USINT,UINT, UDINT, LINT, REAL or LREAL;	User defined data type. The size of the array depends on the number of the registers to read. the end user should define a data type as shown below: TYPE VariableName: array[1..LENGTH] of UINT/USINT/UDINT/LINT/REAL/LREAL; END_TYPE The end user can read the data of a specific register by using the suffix.

User Defined Protocol Error Codes

Refer to the following table for Defined Protocol Error Codes:

GEN_ERR:

Error Code	Description
1	Input parameter is invalid.
2	Time out no response received
3	Request time out
4	Invalid request
5	Invalid module or module offline

Protocol Errors:

Error Code	Description
03	time out, no response from the device
04	Service version mismatch
05	The port is used by another function block.
06	internal error
07	Connection invalid, the target port is not bound to user defined protocol.

CHAPTER

21

OPC UA

The following OPC UA function blocks are available:

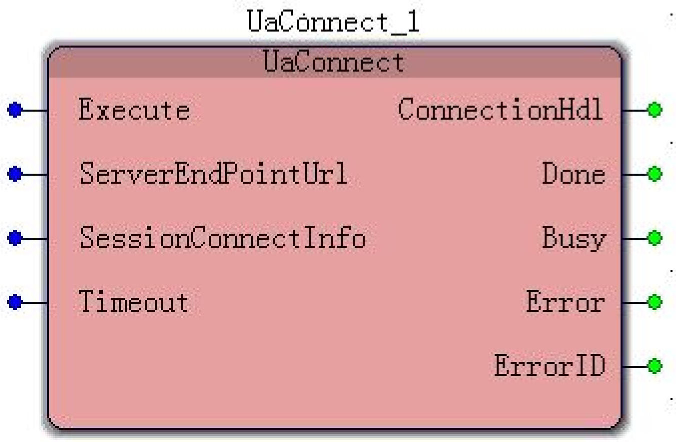
Function Blocks	Short Description
See UaConnect for more information.	This Function Block is used to create a (optional secure) transport connection and an OPC-UA session. The connection shall be terminated by calling the UA_ Disconnect after establishing the connection.
See UaDisconnect for more information.	This Function Block is used to close a transport connection and an OPC-UA session.
See UaMethodCall for more information.	This Function Block is used to call a method routine.
See UaMethodReleaseHandle for more information.	This Function Block is used to release the method handle.
See UaMethodGetHandle for more information.	This Function Block is used to get the method handle for a method call.
See UaNamespaceGetIndex for more information.	This Function Block is used to get the namespace-index of a namespace-URI.
See UaNodeGetHandle for more information.	This Function Block is used to get the node handle.
See UaNodeGetHandleList for more information.	This Function Block is used to get node handles for multiple nodes.
See UaNodeReleaseHandle for more information.	This Function Block is used to release the node handle.
See UaNodeReleaseHandleList for more information.	This Function Block is used to release a set of node handles.
See UaRead for more information.	This Function Block is used to read the value of a single node.
See UaReadList for more information.	This Function Block is used to read values of multiple nodes using a list of node handles.
See UaTranslatePath for	This Function Block is used to get the node parameters

Function Blocks	Short Description
more information.	of a node using path of the node.
See UaTranslatePaths for more information.	This Function Block is used to get the node parameters of a node using path of the node.
See UaWrite for more information.	This Function Block is used to write a value to a single node.
See UaWriteList for more information.	This Function Block is used to write values to multiple nodes using a list of node handles
See UA_MonitoredItemAdd for more information.	This Function Block is used to add handle that values are updated by subscription.
UAMonitoredItemRemove	This Function Block is used to remove a handle from a subscription.
See UASubscriptionCreate for more information.	This Function Block is used to create a subscription.
See UA_SubscriptionDelete for more information.	This Function Block is used to delete a subscription.
See UASubscriptionOperate for more information.	This Function Block is designed to be optionally called - even cyclically- to check if the variables have been published and to check and modify publishing parameters (enable / interval).

Related Topics:

The_Block_Diagram
OPCUA_Function_Block__Data_Type_Reference
OPCUA_Function_Block_Error_Code_Reference

UaConnect



Description

This Function Block is used to create a (optional secure) transport connection and an OPC-UA session. The connection shall be terminated by calling the UA_Disconnect after establishing the connection.

Input

Parameter	Data type	Description
Excute	BOOL	On rising edge connection is started.
ServerEndPointUrl	STRING	URL
SessionConnectInfo	STRUCT	See the information below
Timeout	TIME	Maximum time to establish the connection. **TIP:** If the time to establish the connection takes longer than the Timeout, the Error output variable will be set TRUE and ErrorID will be set to error code 0x800A0000 (OpcUa_BadTimeout).

SessionConnectInfo

UASessionConnectInfo	DataType	Description
SessionName	STRING	Defines the name of the session assigned by the client. The name is shown in the diagnostics information of the server. In case of empty string the server will generate a session name.
ApplicationName	STRING	Defines the readable name of the OPC UA client application. The string can be empty.
SecurityMsgMode	UASecurityMsgMode	See UASecurityMsgMode section below.
SecurityPolicy	UASecurityPolicy	See UASecurityPolicy section below.
CertificateStore	STRING	Defines the location of the certificate store used for the application certificates and trust lists. The structure of the certificate store is vendor specific. In case of empty string the default certificate store is used.
ClientCertificateName	STRING	Defines the name of the client certificate and private key in the certificate store. In case of empty string the default client application certificate is used. Implementation note: The ApplicationURI will be extracted from the certificate.
ServerUri	STRING	Defines the URI of the server.
CheckServerCertificate	BOOL	Flag indicating if the server certificate should be checked with the trust list of the client application.
TransportProfile	UATransportProfile	See UATransportProfile section below.
UserIdentityToken	UAUserIdentityToken	See UAUserIdentityToken section

UASessionConnectInfo	DataType	Description
		below.
VendorSpecificParameter	STRING	Vendor may define specific parameters. e.g. In case multiple clients are available, client instance can be defined with this parameter. The string can be empty.
SessionTimeout	TIME	Defines how long the session will survive when there is no connection.
MonitorConnection	TIME	Defines the interval time to check the connection.
LocaleIDs	ARRAY [1..5] OF STRING[6]	OPC-UA Part3 / Chapter 8.4: <language>[-<country/region>] where <language> is a two letter ISO639 code for language, <country /region> is the three letter ISO3166 code for the country/region. Sample: en-US, zh-CHS

Output

Parameter	Data type	Description
ConnectionHdl	DWORD	Connection handle – is valid until UA_Disconnect is called.
Done	BOOL	Signals a connection has been initially established.
Busy	BOOL	The FB is not finished and new output values are to be expected.
Error	BOOL	Signals that an error has occurred within the FB.
ErrorID	DWORD	Error code.

UASecurityMsgMode

Value	UASecurityMsgMode	Description
0	UASecurityMsgMode_ BestAvailable	Best available message security mode to the UA server. The client receives the available message security from the server and selects the best. This could also result in level “none security”.
1	UASecurityMsgMode_ None	No security is applied.
2	UASecurityMsgMode_ Sign	All messages are signed but not encrypted.
3	UASecurityMsgMode_ SignEncrypt	All messages are signed and encrypted.

UASecurityPolicy.

Value	UASecurityPolicy	Description
0	UASecurityPolicy_ BestAvailable	Provides the best available security connection to the UA server. The client receives the available policies from the server and selects the best. This can also result in level “none security”.
1	UASecurityPolicy_ None	See OPC UA Part 7 Chapter SecurityPolicy-None.
2	UASecurityPolicy_ Basic128Rsa15	See OPC UA Part 7 Chapter SecurityPolicy-Basic128Rsa15
3	UASecurityPolicy_ Basic256	See OPC UA Part 7 Chapter Securitypolicy-Basic256
4	UASecurityPolicy_ Basic256Sha256	See OPC UA Part 7 Chapter Securitypolicy-Basic256Sha256

UATransportProfile

Value	UATransportProfile	Description
1	UATP_UATcp	See OPC UA Part 7 Chapter UA-TCP UA-SC UA Binary
2	UATP_WSHttpBinary	See OPC UA Part 7 Chapter SOAP-HTTP WS-SC UA Binary

Value	UATransportProfile	Description
3	UATP_ WSHttpXmlOrBinary	See OPC UA Part 7 Chapter SOAP-HTTP WS-SC UA XML-UA Binary
4	UATP_WSHttpXml	See OPC UA Part 7 Chapter SOAP-HTTP WS-SC UA XML

UAUserIdentityToken

<table>
<tr><th>UAUserIdentityToken</th><th>DataType</th><th>Description</th></tr>
<tr><td>UserIdentityToken Type</td><td>UAUserIdentityToken Type</td><td>Defines the identity Token to authenticate a user during the creation of a Session.
UAUserIdentityTokenType:
<table>
<tr><th>Value</th><th>UAUserIdentityToken Type</th><th>Description</th></tr>
<tr><td>0</td><td>UAUITT_Anonymous</td><td>See OPC UA Part 7 Chapter User Token – Anonymous Facet</td></tr>
<tr><td>1</td><td>UAUITT_Username</td><td>See OPC UA Part 7 Chapter User Token – User Name Password Server Facet</td></tr>
<tr><td>2</td><td>UAUITT_x509</td><td>See OPC UA Part 7 Chapter User Token –</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>UAUserIdentityToken</th><th>DataType</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>Value</th><th>UAUserIdentityToken Type</th><th>Description</th></tr>
<tr><td></td><td></td><td>X509 Certificate Server Facet</td></tr>
<tr><td>3</td><td>UAUITT_IssuedToken</td><td>See OPC UA Part 7 Chapter User Token – Issued Token Server Facet (Not supported yet)</td></tr>
</table>
</td></tr>
<tr><td>TokenParam1</td><td>STRING</td><td>In case of TokenType “Anonymous” the Param1 will not be evaluated.

In case of TokenType “Username” the Param1 contains the user name.

In case of TokenType “x509” the Param1 contains the location of the certificate store.</td></tr>
<tr><td>TokenParam2</td><td>STRING</td><td>In case of TokenType “Anonymous” the Param2 will not be evaluated.

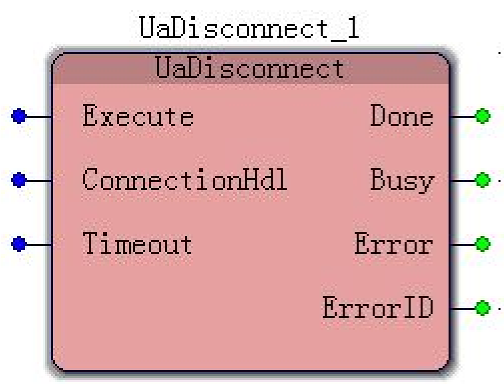
In case of TokenType “Username” the Param2 contains the user password.

In case of TokenType “x509” the Param2 contains the certificate name.</td></tr>
</table>

UserIdentityToken and LocaleIDs have pre-defined types and can be found in **OpcUa_DataTypes** type library. See OPC UA DataType Reference for more information.

TIP: Currently "SecurityMsgMode" must be set to "UASecurityMsgMode_None" (1), "SecurityPolicy" must be set to "UASecurityPolicy_None" (1) and "UserIdentityToken" must be set to "UAUITT_Anonymous" (0). Due to these settings, "CertifcateStore", "ClientCertificateName" and "CheckServercertificate" are ignored. “TransportProfile” must be set to "UATP_UATcp" (1). “MonitorConnection” is currently ignored and is set internally at 5 seconds. “LocalIDs” is defined as an array of 5 elements however, only the first element in the array is used at this time.

UaDisconnect



Description

This Function Block is used to close a transport connection and an OPC-UA session.

Input

Parameter	Data type	Description
Execute	BOOL	On rising edge connection is terminated.
ConnectionHdl	DWORD	Connection handle of connection to be closed.
Timeout	TIME	Maximum time to close the connection. If the time to

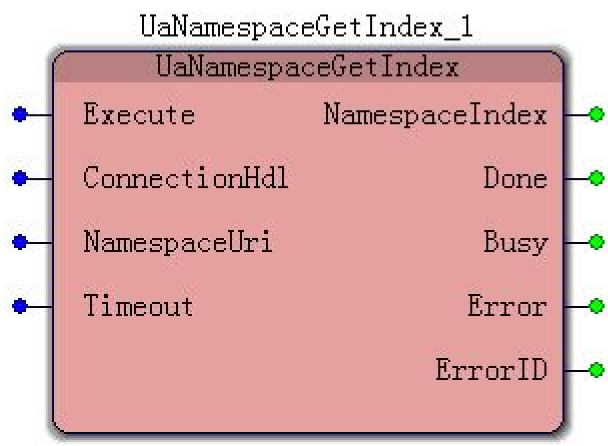
Parameter	Data type	Description
		close the connection takes longer than the Timeout, the Error output variable will be set TRUE and ErrorID will be set to error code 0x800A0000 (OpcUa_BadTimeout)"

Output

Parameter	Data type	Description
Done	BOOL	The function block has completed its task.
Busy	BOOL	The function block is not finished and new output values are to be expected.
Error	BOOL	Signals that an error has occurred within the function block.
ErrorID	DWORD	Error code.

TIP: Calling UA_Disconnect (even in case of timeout or error) will release the ConnectionHdl, all node-handles and MonitoredItems.

UaNamespaceGetIndex



Description

This Function Block is used to get the namespace-index of a namespace-URI.

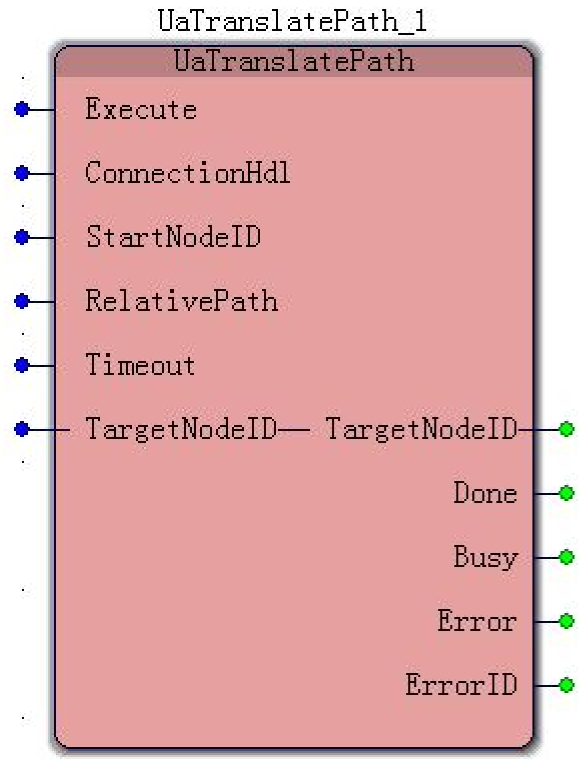
Input

Parameter	Data type	Description
ConnectionHdl	DWORD	Connection handle.
NamespaceUri	STRING	Namespace URI.
Timeout	TIME	Maximum time to response. If the response takes longer than the Timeout, the Error output variable will be set TRUE and ErrorID will be set to error code 0x800A0000 (OpcUa_BadTimeout)"

Output

Parameter	Data type	Description
Done	BOOL	The function block has completed its task.
Busy	BOOL	The function block is not finished and new output values are to be expected.
Error	BOOL	Signals that an error has occurred within the function block.
ErrorID	DWORD	Error code.

UaTranslatePath



Description

This Function Block is used to get the node parameters of a node using path of the node.

Input

Parameter	Data type	Description
Execute	BOOL	The function block performs its task on rising edge on this input.
ConnectionHdl	DWORD	Connection handle.
StartNodeID	STRUCT	See UANodeID. Structure UANodeID with node parameters below for starting node.
RelativePath	STRING	Path of the Target node; BNF of RelativePath is defined in the OPC UA specification Part 4.

Parameter	Data type	Description
Timeout	TIME	Maximum time to response. If the response takes longer than the Timeout, the Error output variable will be set TRUE and ErrorID will be set to error code 0x800A0000 (OpcUa_BadTimeout).

Output

Parameter	Data type	Description
TargetNodeID	STRUCT	See UANodeID. Structure UANodeID below with node parameters. For target node mentioned by RelativePath at the input of this function block.
Done	BOOL	The function block has completed its task.
Busy	BOOL	The function block is not finished and new output values are to be expected.
Error	BOOL	Signals that an error has occurred within the function block.
ErrorID	DWORD	Error code.

UANodeID

<table>
<tr><th>UANodeID</th><th>DataType</th><th>Description</th></tr>
<tr><td>NamespaceIndex</td><td colspan="2">UINT</td></tr>
<tr><td>Identifier</td><td>STRING</td><td>In case of IdentifierType GUID the format is like 00000316-0000-0000-C000-000001000046

In case of IdentifierType Opaque string has to be base 64 encoded byte string.</td></tr>
<tr><td>IdentifierType</td><td>UAIdentifierType</td><td><table>
<tr><td>1</td><td>UAIdentifierType_String</td><td rowspan="3">see OPC UA Part 3 or Part 6</td></tr>
<tr><td>2</td><td>UAIdentifierType_Numeric</td></tr>
<tr><td>3</td><td>UAIdentifierType_GUID</td></tr>
</table></td></tr>
</table>

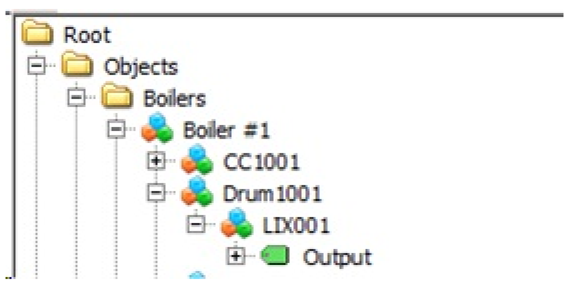
<table>
<tr><th>UANodeID</th><th>DataType</th><th>Description</th></tr>
<tr><td></td><td></td><td><table><tr><td>4</td><td>UAIdentifierType_ Opaque</td><td></td></tr></table></td></tr>
</table>

“RelativePath” is of type “string255” which is simply type “string” with a maximum length of 255 characters. The following rule applies to the format of parameter “RelativePath” :

The RelativePath string is constructed as follows (BNF notation):

```
<relative-path> ::= <reference-type> <browse-name>
[relative-path]

<reference-type> ::= '/'

<browse-name> ::= <namespace-index>':'<name>

<namespace-index> ::= <digit> [<digit>]

<digit> ::= '0' | '1' | '2' | '3' | '4' | '5' | '6' |
'7' | '8' | '9'
```

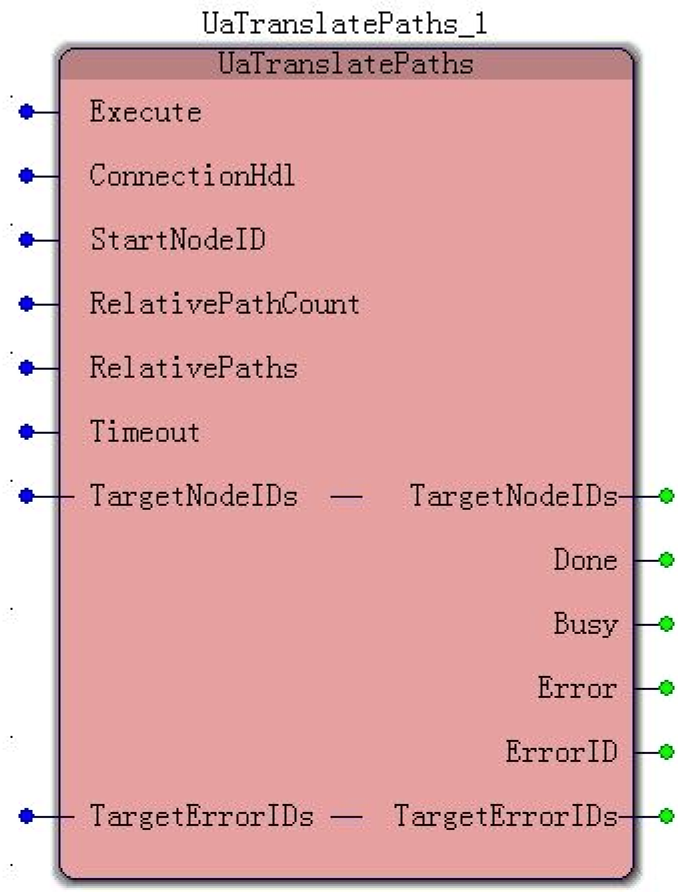
For example, assume a data variable "Output" exists in an address space as shown below:



The relative path to node "Output" from starting node "Boiler #1" would be: **"/4:Drum1001/4:LIX001/4:Output"**. Assume that the naming authority responsible for components "Drum1001", "LIX001" and "Output" is located in the server's namespace table at index 4.See inserted text above which explains where the '4' comes from.

For background on the purpose of this function block refer to the OPC UA specification (OPC UA Part 4 – Services).

UaTranslatePaths



Description

This Function Block is used to get the node parameters of a node using path of the node.

Input

Parameter	Data type	Description
Execute	BOOL	The function block performs its task on rising edge on this input.
ConnectionHdl	DWORD	Connection handle.
StartNodeID	STRUCT	See UANodeID section beolw. Structure UANodeID with

Parameter	Data type	Description
		node parameters for starting node.
RelativePaths	Array of STRING	Paths of the Target nodes; BNF of RelativePath is defined in the OPC UA specification Part 4.
Timeout	TIME	Maximum time to response. If the response takes longer than the Timeout, the Error output variable will be set TRUE and ErrorID will be set to error code 0x800A0000 (OpcUa_BadTimeout)

Output

Parameter	Data type	Description
TargetNodeID	STRUCT	See UANodeID section below. Structure UANodeID with node parameters. For target node mentioned by RelativePath at the input of this FB.
Done	BOOL	The function block has completed its task.
Busy	BOOL	The function block is not finished and new output values are to be expected.
Error	BOOL	Signals that an error has occurred within the function block.
ErrorID	DWORD	Error code.
TargetErrorIDs	ARRAY OF DWORD	Array of DWORD. Contains an error code corresponding to each element in the RelativePaths array. Max length of the array is defined by the vendor and shall be the same length as the RelativePaths array length.

UANodeID

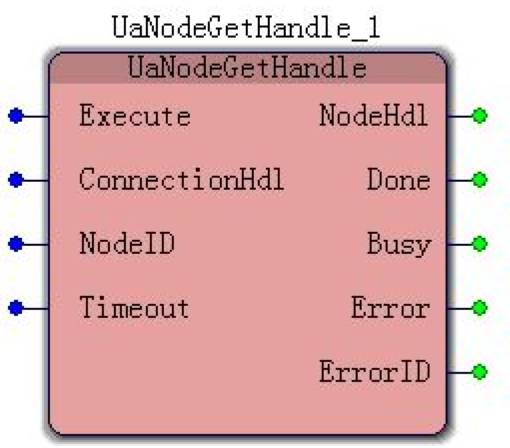
<table>
<tr><th>UANodeID</th><th>DataType</th><th>Description</th></tr>
<tr><td>NamespaceIndex</td><td colspan="2">UINT</td></tr>
<tr><td>Identifier</td><td>STRING</td><td>In case of IdentifierType GUID the format is like 00000316-0000-0000-C000-000001000046

In case of IdentifierType Opaque string has to be base 64 encoded byte string.</td></tr>
<tr><td>IdentifierType</td><td>UAIdentifierType</td><td><table><tr><td>1</td><td>UAIdentifierType_String</td><td>see OPC UA Part 3 or Part 6</td></tr></table></td></tr>
</table>

<table>
<tr><th>UANodeID</th><th>DataType</th><th>Description</th></tr>
<tr><td></td><td></td><td><table><tr><td>2</td><td>UAIdentifierType_
Numeric</td><td></td></tr><tr><td>3</td><td>UAIdentifierType_
GUID</td><td></td></tr><tr><td>4</td><td>UAIdentifierType_
Opaque</td><td></td></tr></table></td></tr>
</table>

TIP: “RelativePaths” is of type “string255List”, pre-defined in **OpcUa_DataTypes** type library. OPC UA DataType Reference

UaNodeGetHandle



Description

This Function Block is used to get the node handle.

Input

Parameter	Data type	Description
Execute	BOOL	The function block performs its task on rising edge on this input.

Parameter	Data type	Description
ConnectionHdl	DWORD	Connection handle.
NodeID	STRUCT	See UANodeID section below.
Timeout	TIME	Time to response. If the response takes longer than the Timeout, the Error output variable will be set TRUE and ErrorID will be set to error code 0x800A0000 (OpcUa_BadTimeout)"

Output

Parameter	Data type	Description
NodeHdl	DWORD	Node handle.
Done	BOOL	The function block has completed its task.
Busy	BOOL	The function block is not finished and new output values are to be expected.
Error	BOOL	Signals that an error has occurred within the function block.
ErrorID	DWORD	Error code.

UANodeID

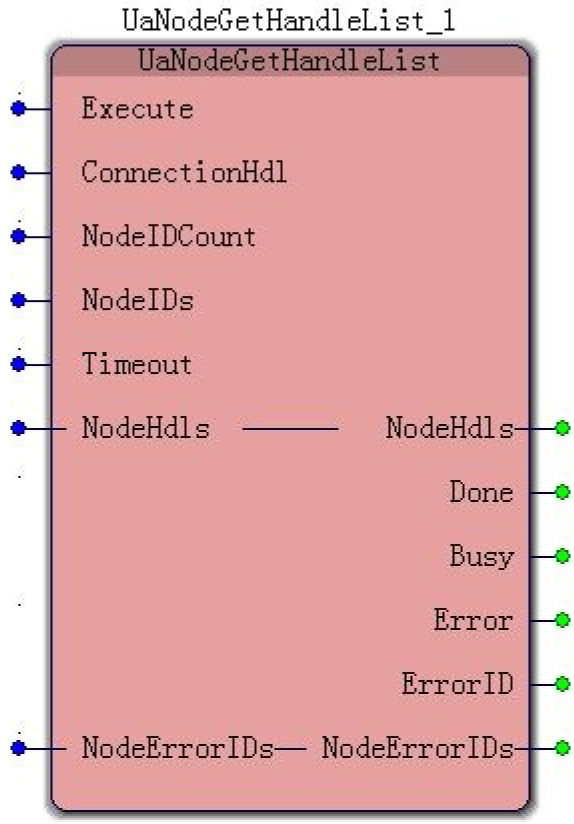
<table>
<tr><th>UANodeID</th><th>DataType</th><th>Description</th></tr>
<tr><td>NamespaceIndex</td><td colspan="2">UINT</td></tr>
<tr><td>Identifier</td><td>STRING</td><td>In case of IdentifierType GUID the format is like 00000316-0000-0000-C000-000001000046

In case of IdentifierType Opaque string has to be base 64 encoded byte string.</td></tr>
<tr><td>IdentifierType</td><td>UAIdentifierType</td><td><table><tr><td>1</td><td>UAIdentifierType_String</td><td rowspan="3">see OPC UA Part 3 or Part 6</td></tr><tr><td>2</td><td>UAIdentifierType_Numeric</td></tr><tr><td>3</td><td>UAIdentifierType_GUID</td></tr></table></td></tr>
</table>

UANodeID	DataType	Description
		4 | UAIdentifierType_ Opaque |

NOTE: The NodeHdl is a reference to the internal management object for the node in the client. But the client shall also register the node at the server ("RegisterNode"). This enables the UA-server to optimize the communication. The scope of the NodeHdl is the connection. So a NodeHdl is unique for a connection but could be equal to a NodeHdl of another connection. Parameter "NodeID" has a pre-defined type and can be found in **OpcUa_DataTypes** type library. Individual structure fields are described . (note that type "string255", also defined in OpcUa_DataTypes and is simply a string data type where the maximum string length is 255 characters). See OPC UA DataType Reference for more information.

UaNodeGetHandleList



Description

This Function Block is used to get node handles for multiple nodes.

Input

Parameter	Data type	Description
Execute	BOOL	The function block performs its task on rising edge on this input.
ConnectionHdl	DWORD	Connection handle.
NodeIDCount	UINT	Number of NodeIDs in Array of NodeIDs. The maximum value for this input variable is 20.
NodeIDs	ARRAY OF STRUCT	See UANodeID section below. Max length of array is to be defined by the vendor. Array length of NodeIDs and NodeHdls must be same.

Parameter	Data type	Description
Timeout	TIME	Maximum time to response. If the response takes longer than the Timeout, the Error output variable will be set TRUE and ErrorID will be set to error code 0x800A0000 (OpcUa_BadTimeout).

Parameter “NodeIDs” has a pre-defined type, “UaNodeIDList” which can be found in **OpcUa_DataTypes** type library. See OPC UA DataType Reference for details. Individual NodeID structure fields are described below.

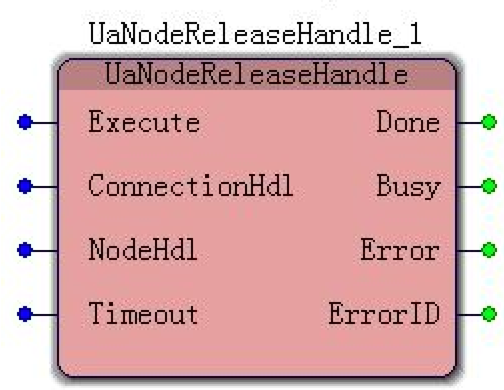
Output

Parameter	Data type	Description
NodeHdls	ARRAY OF DWORD	Array of Node Handles. Max length of array is to be defined by the vendor. Array length of NodeIDs and NodeHdls must be same.
Done	BOOL	The function block has completed its task.
Busy	BOOL	The function block is not finished and new output values are to be expected.
Error	BOOL	Signals that an error has occurred within the function block. Set to true if either ErrorID or any of the NodeErrorIDs indicates an error.
ErrorID	DWORD	Error code.
NodeErrorIDs	ARRAY OF DWORD	Array of NodeErrorIDs. Contains an error code for each valid element of the NodeIds array. Max length of array is to be defined by the vendor and shall be same size like the NodesIDs array length.

UANodeID

UANodeID	DataType	Description
NamespaceIndex	UINT	
Identifier	STRING	In case of IdentifierType GUID the format is like 00000316-0000-0000-C000-000001000046 In case of IdentifierType Opaque string has to be base 64 encoded byte string.
IdentifierType	UAIdentifierType	1 — UAIdentifierType_String 2 — UAIdentifierType_Numeric 3 — UAIdentifierType_GUID 4 — UAIdentifierType_Opaque see *OPC UA Part 3 or Part 6*

UaNodeReleaseHandle



Description

This Function Block is used to release the node handle.

Input

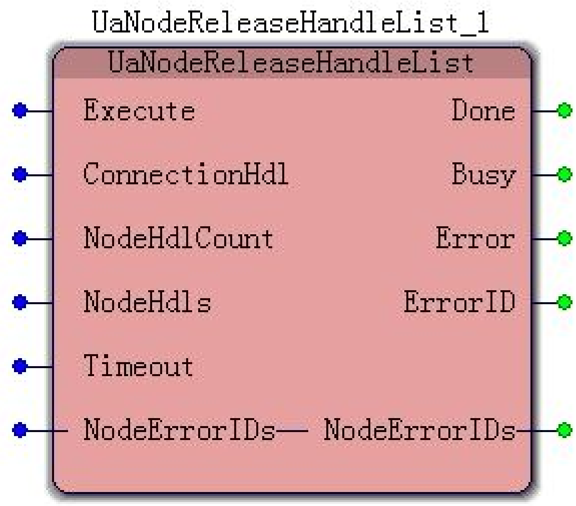
Parameter	Data type	Description
Execute	BOOL	The function block performs its task on rising edge on this input.
ConnectionHdl	DWORD	Connection handle.
NodeHdl	DWORD	Node handle to be released.
Timeout	TIME	Maximum time to response. If the response takes longer than the Timeout, the Error output variable will be set TRUE and ErrorID will be set to error code 0x800A0000 (OpcUa_BadTimeout).

Output

Parameter	Data type	Description
Done	BOOL	The function block has completed its task.
Busy	BOOL	The function block is not finished and new output values are to be expected.
Error	BOOL	Signals that an error has occurred within the function block.
ErrorID	DWORD	Error code.

TIP: After calling UA_NodeReleaseHandle the NodeHdl will be invalid.

UaNodeReleaseHandleList



Description

This Function Block is used to release a set of node handles.

Input

Parameter	Data type	Description
Execute	BOOL	The function block performs its task on rising edge on this input.
ConnectionHdl	DWORD	Connection handle.
NodeHdlCount	UINT	Number of Nodes in NodeHdls Array. The maximum value for this input variable is 20.
NodeHdls	ARRAY OF DWORD	Array of Node handles to be released. Max length of array is to be defined by the vendor. NULL is not a valid handle. **TIP:** "NodeHdls" has a pre-defined type "UaDWORDList" which can be found in OpcUa_DataTypes type library.
Timeout	TIME	Maximum time to response. If the response takes longer

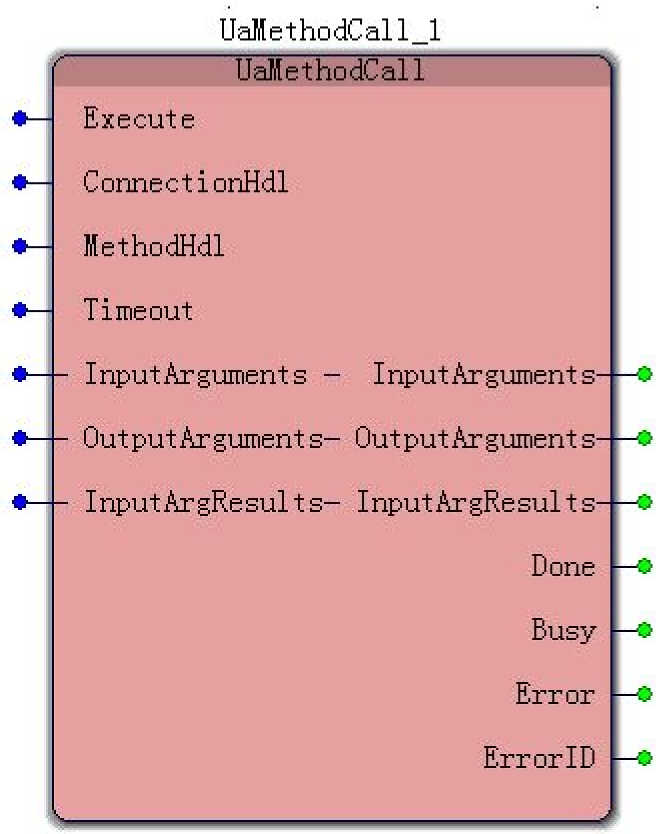
Parameter	Data type	Description
		than the Timeout, the Error output variable will be set TRUE and ErrorID will be set to error code 0x800A0000 (OpcUa_BadTimeout).

Output

Parameter	Data type	Description
Done	BOOL	The function block has completed its task.
Busy	BOOL	The function block is not finished and new output values are to be expected.
Error	BOOL	Signals that an error has occurred within the function block. Set to true if either ErrorID or any of the NodeErrorIDs indicates an error.
ErrorID	DWORD	Error code.
NodeErrorIDs	ARRAY OF DWORD	Array of DWORD. Contains an error code for each valid element of the NodeHdls array. Max length of array is to be defined by the vendor and shall be same size like the NodeHdls array length. **TIP:** "NodeErrorIDs" has a pre-defined type "UaDWORDList" which can be found in OpcUa_ DataTypes type library.

TIP: After calling UA_NodeReleaseHandleList the NodeHdls will be invalid.

UaMethodCall



Description

This Function Block is used to call a method routine.

Input

Parameter	Data type	Description
Execute	BOOL	The function block performs its task on rising edge on this input.
ConnectionHdl	DWORD	Connection handle.
MethodHdl	DWORD	Method handle.

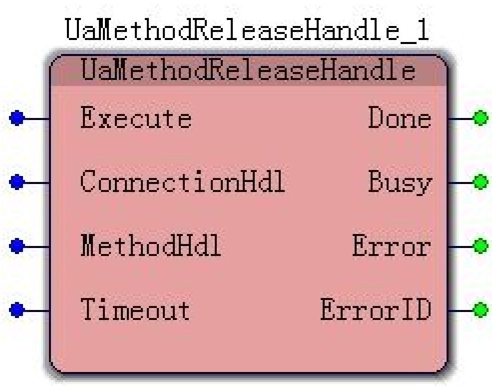
Parameter	Data type	Description
Timeout	TIME	Maximum time to response. If the response takes longer than the Timeout, the Error output variable will be set TRUE and ErrorID will be set to error code 0x800A0000 (OpcUa_BadTimeout).

Output

Parameter	Data type	Description
Busy	BOOL	The function block is not finished and new output values are to be expected.
Error	BOOL	Signals that an error has occurred within the function block.
ErrorID	DWORD	Error code.

TIP: InputArguments and OutputArguments have a pre-defined type, “UAVariantList” which can be found in **OpcUa_DataTypes** type library. The UA_MethodCall function block has one additional parameter, inputArgResults which has a pre-defined type “UaDWORDList” and can be found in **OpcUa_DataTypes** type library. See OPC UA DataType Reference for more information.

UaMethodReleaseHandle



Description

This Function Block is used to release the method handle.

Input

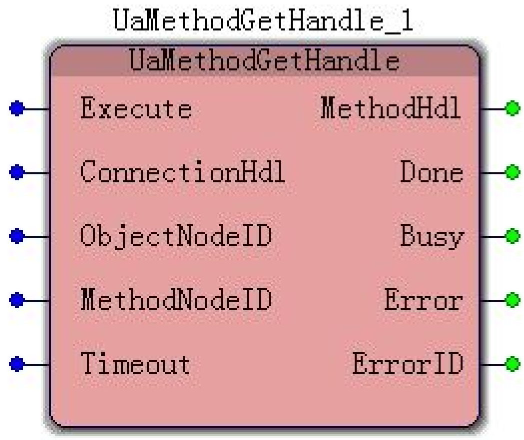
Parameter	Data type	Description
ConnectionHdl	DWORD	Connection handle.
MethodHdl	DWORD	Method handle to be released.
Timeout	TIME	Maximum time to response. If the response takes longer than the Timeout, the Error output variable will be set TRUE and ErrorID will be set to error code 0x800A0000 (OpcUa_BadTimeout).

Output

Parameter	Data type	Description
Busy	BOOL	The function block is not finished and new output values are to be expected.
Error	BOOL	Signals that an error has occurred within the function block.
ErrorID	DWORD	Error code

TIP: After calling UA_MethodReleaseHandle the MethodHdl will be invalid.

UaMethodGetHandle



Description

This Function Block is used to get the method handle for a method call.

Input

Parameter	Data type	Description
ConnectionHdl	DWORD	Connection handle.
ObjectNodeID	STRUCT	See UANodeID section below.
MethodNodeID	STRUCT	See UANodeID section below.
Timeout	TIME	Maximum time to response. If the response takes longer than the Timeout, the Error output variable will be set TRUE and ErrorID will be set to error code 0x800A0000 (OpcUa_BadTimeout).

Output

Parameter	Data type	Description
MethodHdl	DWORD	Method handle.

Parameter	Data type	Description
Done	BOOL	The function block has completed its task.
Busy	BOOL	The function block is not finished and new output values are to be expected.
Error	BOOL	Signals that an error has occurred within the function block.
ErrorID	DWORD	Error code.

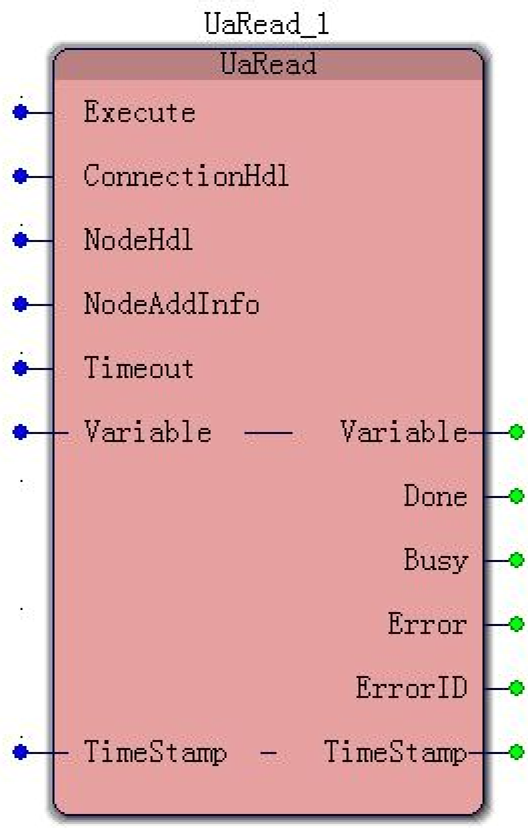
UANodeID

<table>
<tr><th>UANodeID</th><th>DataType</th><th>Description</th></tr>
<tr><td>NamespaceIndex</td><td>UINT</td><td></td></tr>
<tr><td>Identifier</td><td>STRING</td><td>In case of IdentifierType GUID the format is like 00000316-0000-0000-C000-000001000046

In case of IdentifierType Opaque string has to be base 64 encoded byte string.</td></tr>
<tr><td>IdentifierType</td><td>UAIdentifierType</td><td><table>
<tr><td>1</td><td>UAIdentifierType_String</td><td rowspan="4">see OPC UA Part 3 or Part 6</td></tr>
<tr><td>2</td><td>UAIdentifierType_ Numeric</td></tr>
<tr><td>3</td><td>UAIdentifierType_GUID</td></tr>
<tr><td>4</td><td>UAIdentifierType_ Opaque</td></tr>
</table></td></tr>
</table>

> **TIP:** ObjectNodeID and MethodNodeID parameters have a pre-defined type, “UANodeID” which can be found in **OpcUa_ DataTypes** type library.

UaRead



Description

This Function Block is used to read the value of a single node.

Input

Parameter	Data type	Description
Execute	BOOL	On rising edge node information will be read.
ConnectionHdl	DWORD	Connection handle.
NodeHdl	DWORD	Node handle.
NodeAddInfo	STRUCT	See UANodeAdditionalInfo. Specifies the attribute and IndexRange below.
Timeout	TIME	Maximum time to response. If the response takes longer than the Timeout, the Error output variable will be set

Parameter	Data type	Description
		TRUE and ErrorID will be set to error code 0x800A0000 (OpcUa_BadTimeout)

UANodeAdditionalInfo

<table>
<tr><th>ANodeAdditionalInfo</th><th>DataType</th><th>Description</th></tr>
<tr><td>AttributeID</td><td>UAAttributeID</td><td>Selects the attribute to be accessed. The default AttributeID is eUAAI_Value (13).

The value of a variable.</td></tr>
<tr><td>IndexRangeCount</td><td>UINT</td><td>Count of valid IndexRange specified. Vendorspecific.</td></tr>
<tr><td>IndexRange</td><td>ARRAY OF UAIndexRange</td><td>
<table>
<tr><th>UAIndexRange</th><th>DataType</th><th>Description</th></tr>
<tr><td>StartIndex</td><td>UINT</td><td>Start index</td></tr>
<tr><td>EndIndex</td><td>UINT</td><td>End index</td></tr>
</table>
TIP:
IndexRange can be defined as follows:

For each Dimension:

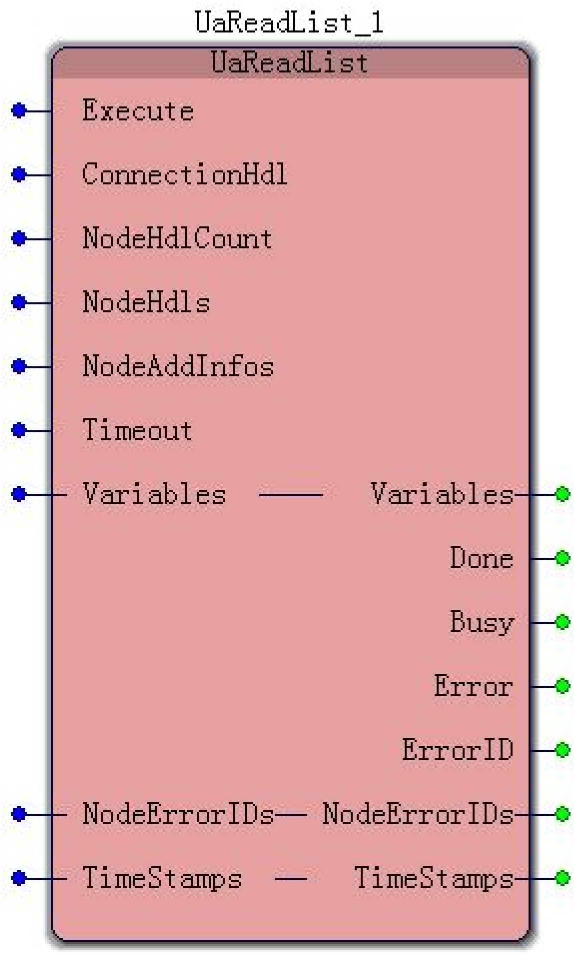
1. Start and EndIndex are to be assigned.
2. StartIndex must be smaller than EndIndex.
3. To access all the elements in a Dimension it’s a must to assign StartIndex and EndIndex depending on the number of total Elements in the Dimension.
4. A single element in a Dimension can be selected by specifying the same StartIndex and EndIndex.
</td></tr>
</table>

TIP: "NodeAddInfo" has a pre-defined type which can be found in **OpcUa_DataTypes** type library. Parameter "Variable" has a pre-defined type "UADataValue" which can be found in **OpcUa_DataTypes** type library. Embedded fields UAVariant and UADateTime also have pre-defined types which can be found in **OpcUa_DataTypes type library**. See OPC UA DataType Reference for more information.

Output

Parameter	Data type	Description
Done	BOOL	The function block has completed its task.
Busy	BOOL	The function block is not finished and new output values are to be expected.
Error	BOOL	Signals that an error has occurred within the function block.
ErrorID	DWORD	Error code.
TimeStamp	DT	TimeStamp.

UaReadList



Description

This Function Block is used to read values of multiple nodes using a list of node handles.

Input

Parameter	Data type	Description
Execute	BOOL	On rising edge node information will be read.
ConnectionHdl	DWORD	Connection handle.
NodeHdlCount	UINT	Number of valid elements in the array to read. The

Parameter	Data type	Description
		maximum value for this input variable is 20.
NodeHdls	ARRAY OF DWORD	Array of Node Handles. Max length of array is to be defined by the vendor and shall be same size like the Variables array length. **TIP:** "NodeHdls" has a pre-defined type "UaDWORDList" which can be found in OpcUa_ DataTypes type library.
NodeAddInfos	ARRAY OF STRUCT	See UANodeAdditionalInfo section below. Array of UANodeAdditionalInfo. Specifies the attribute and IndexRange. Max length of array is to be defined by the vendor and shall be same size like the Variables array length.
Timeout	TIME	Maximum time to response. If the response takes longer than the Timeout, the Error output variable will be set TRUE and ErrorID will be set to error code 0x800A0000 (OpcUa_BadTimeout).

Output

Parameter	Data type	Description
Done	BOOL	The function block has completed its task.
Busy	BOOL	The function block is not finished and new output values are to be expected.
Error	BOOL	Signals that an error has occurred within the function block. Set to true if either ErrorID or any of the NodeErrorIDs indicates an error.
ErrorID	DWORD	Error code for the OPC UA service call.
NodeErrorIDs	ARRAY OF DWORD	Array of DWORD. Contains an error code for each valid element of the Variables array. Max length of array is to be defined by the vendor and shall be same size like the Variables array length. **TIP:** "NodeErrorIDs" has a pre-defined type "UaDWORDList" which can be found in OpcUa_

Parameter	Data type	Description
		DataTypes type library.
TimeStamps	ARRAY OF DT	Contains a TimeStamp for each valid element of the Variables array. Max length of array is to be defined by the vendor and shall be same size like the Variables array length. **TIP:** "TimeStamps" has a pre-defined type "UaDateTimeList" which can be found in OpcUa_DataTypes type library.

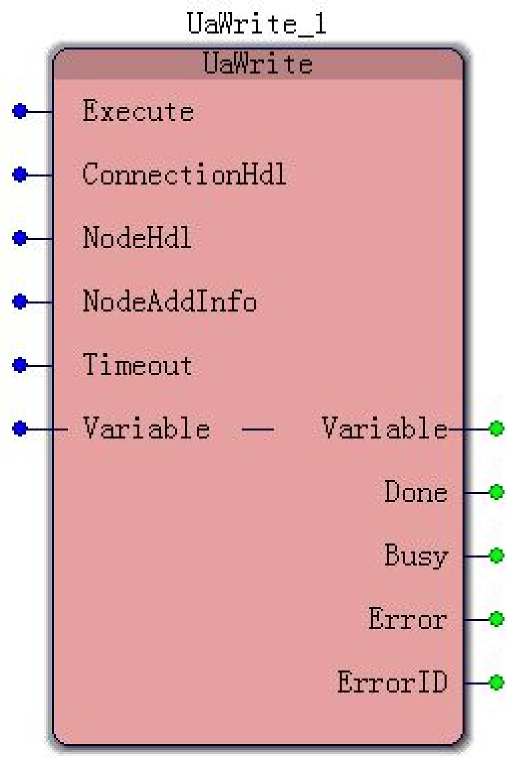
UANodeAdditionalInfo

ANodeAdditionalInfo	DataType	Description
AttributeID	UAAttributeID	Selects the attribute to be accessed. The default AttributeID is eUAAI_Value (13). The value of a variable.
IndexRangeCount	UINT	Count of valid IndexRange specified. Vendorspecific.
IndexRange	ARRAY OF UAIndexRange	UAIndexRange / DataType / Description: StartIndex / UINT / Start index EndIndex / UINT / End index **TIP:** IndexRange can be defined as follows: For each Dimension: 1. Start and EndIndex are to be assigned. 2. StartIndex must be smaller than EndIndex. 3. To access all the elements in a

ANodeAdditionalInfo	DataType	Description
		UAIndexRange | DataType | Description Dimension **it's a must to assign** StartIndex and EndIndex depending on the number of total Elements in the Dimension. 4. A single element in a Dimension can be selected by specifying the same StartIndex and EndIndex.

TIP: Parameter "NodeAddInfos" has a pre-defined type, "UaNodeAddInfoList" which can be found in **OpcUa_DataTypes** type library. Parameter "Variables" has a pre-defined type, "UADataValueList" which can be found in **OpcUa_DataTypes** type library. See OPC UA DataType Reference for more information.

UaWrite



Description

This Function Block is used to write a value to a single node.

Input

Parameter	Data type	Description
Execute	BOOL	On rising edge node information will be written.
ConnectionHdl	DWORD	Connection handle.
NodeHdl	DWORD	Node handle.
NodeAddInfo	STRUCT	See UANodeAdditionalInfo. Specifies the attribute and IndexRange below.
Timeout	TIME	Maximum time to response. If the response takes longer than the Timeout, the Error output variable will be set TRUE and ErrorID will be set to error code 0x800A0000

Parameter	Data type	Description
		(OpcUa_BadTimeout).

Output

Parameter	Data type	Description
Done	BOOL	The function block has completed its task.
Busy	BOOL	The function block is not finished and new output values are to be expected.
Error	BOOL	Signals that an error has occurred within the function block.
ErrorID	DWORD	Error code.

UANodeAdditionalInfo

<table>
<tr><th>NodeAdditionalInfo</th><th>DataType</th><th>Description</th></tr>
<tr><td>AttributeID</td><td>UAAttributeID</td><td>Selects the attribute to be accessed. The default AttributeID is eUAAI_Value (13).

The value of a variable.</td></tr>
<tr><td>IndexRangeCount</td><td>UINT</td><td>Count of valid IndexRange specified. Vendorspecific.</td></tr>
<tr><td>IndexRange</td><td>ARRAY OF UAIndexRange</td><td><table>
<tr><th>UAIndexRange</th><th>DataType</th><th>Description</th></tr>
<tr><td>StartIndex</td><td>UINT</td><td>Start index</td></tr>
<tr><td>EndIndex</td><td>UINT</td><td>End index</td></tr>
</table>
TIP:
IndexRange can be defined as follows:

For each Dimension:

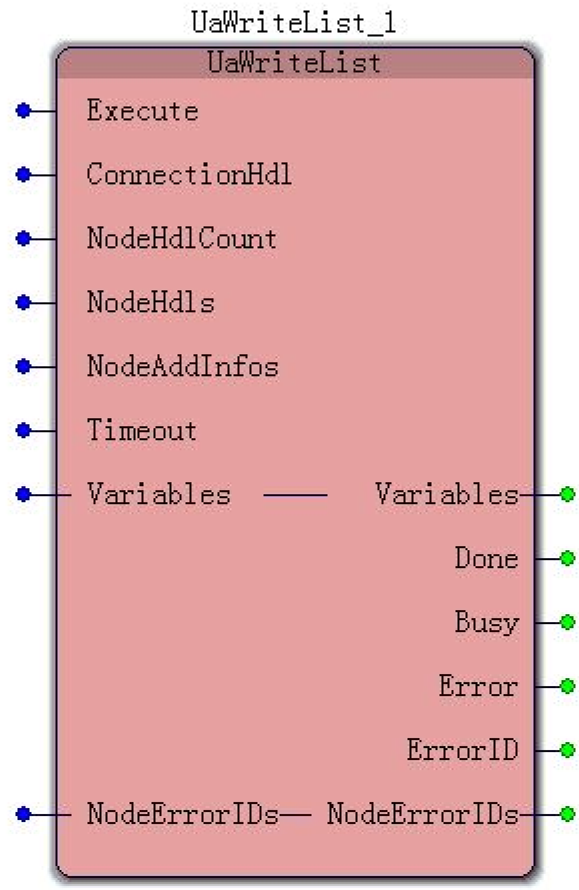
1. Start and EndIndex are to be assigned.

2. StartIndex must be smaller than EndIndex.</td></tr>
</table>

NodeAdditionalInfo	DataType	Description
		UAIndexRange / DataType / Description: 3. To access all the elements in a Dimension **it's a must to assign** StartIndex and EndIndex depending on the number of total Elements in the Dimension. 4. A single element in a Dimension can be selected by specifying the same StartIndex and EndIndex.

TIP: "NodeAddInfo" has a pre-defined type, "UaNodeAdditionalInfo" which can be found in OpcUa_DataTypes type library (refer to UA_Read above). Parameter "Variable" has a pre-defined type, "UADataValue" which can be found in OpcUa_DataTypes type library. See OPC UA DataType Reference for more information.

UaWriteList



Description

This Function Block is used to write values to multiple nodes using a list of node handles.

Input

Parameter	Data type	Description
Execute	BOOL	On rising edge node values will be written.
ConnectionHdl	DWORD	Connection handle.
NodeHdlCount	UINT	Number of valid elements in the array to write. The maximum value for this input variable is 20.

Parameter	Data type	Description
NodeHdls	ARRAY OF DWORD	Array of Node Handles. Max length of array is to be defined by the vendor and shall be same size like the Variables array length. **TIP:** "NodeHdls" has a pre-defined type "UaDWORDList" which can be found in OpcUa_DataTypes type library.
NodeAddInfos	ARRAY OF STRUCT	See UANodeAdditionalInfo section below. Array of UANodeAdditionalInfo. Specifies the attribute and IndexRange. Max length of array is to be defined by the vendor and shall be same size like the Variables array length.
Timeout	TIME	Maximum time to response. If the response takes longer than the Timeout, the Error output variable will be set TRUE and ErrorID will be set to error code 0x800A0000 (OpcUa_BadTimeout).

Output

Parameter	Data type	Description
Done	BOOL	The function block has completed its task.
Busy	BOOL	The function block is not finished and new output values are to be expected.
Error	BOOL	Signals that an error has occurred within the function block. Set to true if either ErrorID or any of the NodeErrorIDs indicates an error.
ErrorID	DWORD	Error code for the OPC UA service call.
NodeErrorIDs	ARRAY OF DWORD	Array of DWORD. Contains an error code for each valid element of the Variables array. Max length of array is to be defined by the vendor and shall be same size like the Variables array length. **TIP:** "NodeErrorIDs" has a pre-defined type

Parameter	Data type	Description
		"UaDWORDList" which can be found in OpcUa_ DataTypes type library.

UANodeAdditionalInfo

<table>
<tr><th>ANodeAdditionalInfo</th><th>DataType</th><th>Description</th></tr>
<tr><td>AttributeID</td><td>UAAttributeID</td><td>Selects the attribute to be accessed. The default AttributeID is eUAAI_Value (13).

The value of a variable.</td></tr>
<tr><td>IndexRangeCount</td><td>UINT</td><td>Count of valid IndexRange specified. Vendorspecific.</td></tr>
<tr><td>IndexRange</td><td>ARRAY OF UAIndexRange</td><td>
<table>
<tr><th>UAIndexRange</th><th>DataType</th><th>Description</th></tr>
<tr><td>StartIndex</td><td>UINT</td><td>Start index</td></tr>
<tr><td>EndIndex</td><td>UINT</td><td>End index</td></tr>
</table>
TIP:
IndexRange can be defined as follows:

For each Dimension:

1. Start and EndIndex are to be assigned.
2. StartIndex must be smaller than EndIndex.
3. To access all the elements in a Dimension it's a must to assign StartIndex and EndIndex depending on the number of total Elements in the Dimension.
4. A single element in a Dimension can be selected by specifying the same StartIndex
</td></tr>
</table>

ANodeAdditionalInfo	DataType	Description
		UAIndexRange / DataType / Description: and EndIndex.

TIP: "NodeAddInfos" has a pre-defined type, "UaNodeAddInfoList" which can be found in OpcUa_DataTypes type library (refer to UA_ReadList above). Parameter "Variables" has a pre-defined type "UADataValueList" which can be found in OpcUa_DataTypes type library. See OPC UA DataType Reference for more information.

UA_MonitoredItemAdd

This Function Block can be used to add handle that values are updated by subscription.

Input

Parameter	Data type	Description
Execute	BOOL	On rising edge monitored item will be added to a subscription.
SubscriptionHdl	DWORD	Subscription handle.
NodeHdl	DWORD	Node handle.
NodeAddInfo	DWORD	See 3.2.8 UANodeAdditionalInfo. Specifies the attribute and IndexRange.
Timeout	TIME	Time to response

Output

Parameter	Data type	Description
MonitoredItemHdl	DWORD	Monitored item handle.

Parameter	Data type	Description
Done	BOOL	The function block has completed its task.
Busy	BOOL	The function block is not finished and new output values are to be expected.
Error	BOOL	Signals that an error has occurred within the function block.
ErrorID	DWORD	Error code.

Input and Output

Parameter	Data type	Description
DataChangeNotification	See its corresponding type in OPCUA_Function_Block__Data_Type_Reference.	Please note the discussion on the two ways to retrieve the latest subscribed values within the PLC in the PLCopen specification section 2.2. This implementation utilizes method “PLC_sync”. This means that although data change notifications are “pushed” from the server, it is still necessary for the Monitored Item function block to execute in order to retrieve the value. When the block's execute flag is first set and the rising edge is detected, the Monitored item will be added to the subscription. Thereafter, each subsequent rising edge will copy the latest pushed value for the monitored item into the DataChangeNotification output parameter. Additionally, note that the DataValues field of the UADataChangeNotification structure below is an array of UaDataValue. The reason for this is that depending on the settings of the subscriptions PublishingInterval and the monitored items SamplingInterval, multiple values for a single monitored item may be available. For example, if the SamplingInterval is set to one second

Parameter	Data type	Description
		and the PublishingInterval is set to five seconds then on average five values will be pushed from the server per notification. Refer to OPC UA Specification part 4: Services for additional detail on this behavior.
MonitoringSettings	STRUCT	See 3.2.4 UAMonitoringSettings

NOTE: VAR_IN_OUT: „Variable" as would provide best type save solution for users: The client firmware is able to map the UA memory layout to the controller layout. The firmware client can receive the type definition from the UA-Server. Workaround would be to provide a byte array as "Variable" and the firmware client just provide the blob (UA memory layout – so called "raw data") into that byte array. "Variable" could be the name of the variable so the internal firmware can get address, length, data type of variable.

TIP: Parameter MonitoringSettings is both input and output parameter. The reason for this is due to field SamplingInterval which is a negotiated value similar to PublishingInterval for blocks UA_SubscriptionCreate and UA_SubscriptionOperate.A requested value of zero is a signal to the server to select the fastest practical rate.

UAMonitoredItemRemove

This Function Block can be used to remove a handle from a subscription.

Input

Parameter	Data type	Description
Execute	BOOL	On rising edge monitored item will be added to a subscription.
SubscriptionHdl	DWORD	Subscription handle.

Parameter	Data type	Description
Timeout	TIME	Time to response

Output

Parameter	Data type	Description
Done	BOOL	The function block has completed its task.
Busy	BOOL	The function block is not finished and new output values are to be expected.
Error	BOOL	Signals that an error has occurred within the function block.
ErrorID	DWORD	Error code.

> **NOTE:** Parameter SubscriptionHdl is appears as an input but is not shown in specification. This represents the handle of the subscription to which this monitored item was added and is an omission in the specification.

UASubscriptionCreate

This Function Block can be used to create a subscription.

Input

Parameter	Data type	Description
Execute	BOOL	On rising edge subscription will be created.
ConnectionHdl	DWORD	Connection handle.
PublishingEnable	BOOL	Activate the publishing.
Priority	BYTE	Priority of the Subscription in the server relative to the other Subscriptions created by this client.
Timeout	TIME	Maximum time to response.

Output

Parameter	Data type	Description
SubscriptionHdl	DWORD	Subscription handle.
Done	BOOL	The function block has completed its task.
Busy	BOOL	The function block is not finished and new output values are to be expected.
Error	BOOL	Signals that an error has occurred within the function block.
ErrorID	DWORD	Error code.

Input and Output

Parameter	Data type	Description
PublishingInterval	TIME	Publishing interval (can be changed by the Server revised publishing interval).

NOTE: PublishingInterval is both an input and an output. This is because of the concept that the publishing interval is a negotiated parameter in OPC UA.The block begins execution with a “requested” publishing interval. If the server is unable to support this interval then it will respond with an alternate value. If the requested value is 0 or negative, the server will revise with the fastest supported publishing interval. Also, the negotiated value for this parameter returned in the response is used as the default sampling interval for Monitored Items assigned to this Subscription (see UA_MonitoredItemAdd below). For more information on publishing interval negotiation, refer to OPC UA specification part 4: Services.

UA_SubscriptionDelete

This Function Block can be used to delete a subscription.

Input

Parameter	Data type	Description
Variable	Type	Description
Execute	BOOL	On rising edge subscription will be created.
SubscriptionHdl	DWORD	Subscription handle.
Timeout	TIME	Maximum time to response.

Output

Parameter	Data type	Description
Variable	Type	Description
Done	BOOL	The function block has completed its task.
Busy	BOOL	The function block is not finished and new output values are to be expected.
Error	BOOL	Signals that an error has occurred within the function block.
ErrorID	DWORD	Error code.

UASubscriptionOperate

This Function Block is designed to be optionally called -even cyclically- to check if the variables have been published and to check and modify publishing parameters (enable / interval).

Input

Parameter	Data type	Description
Execute	BOOL	On rising edge subscription will be created.
SubscriptionHdl	DWORD	Subscription handle.
PublishingEnable	BOOL	Activate the publishing.
Priority	BYTE	Priority of the Subscription in the server relative to the other Subscriptions created by this client.

Parameter	Data type	Description
Timeout	TIME	Time to response.

Output

Parameter	Data type	Description
Published	BOOL	Indicates, that variables have been published since the previous call.
Done	BOOL	FB has completed its task.
Busy	BOOL	The FB is not finished and new output values are to be expected.
Error	BOOL	Signals that an error has occurred within the FB.
ErrorID	DWORD	Error code.

Input and Output

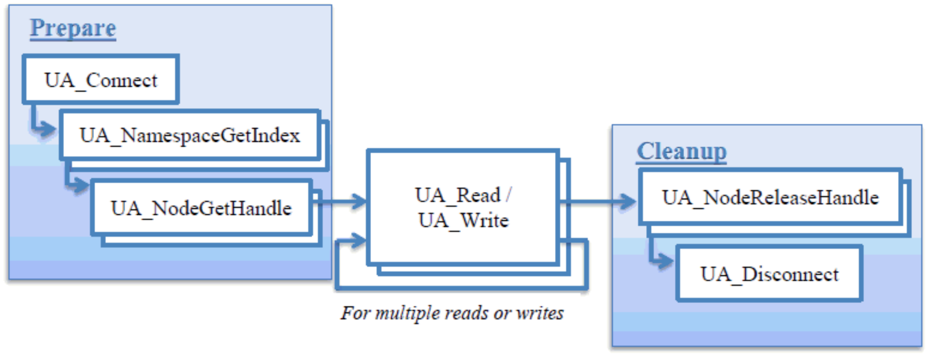
Parameter	Data type	Description
Variable	Type	Description
PublishingInterval	TIME	Publishing interval (can be changed by the Server revised publishing interval).

The Block Diagram

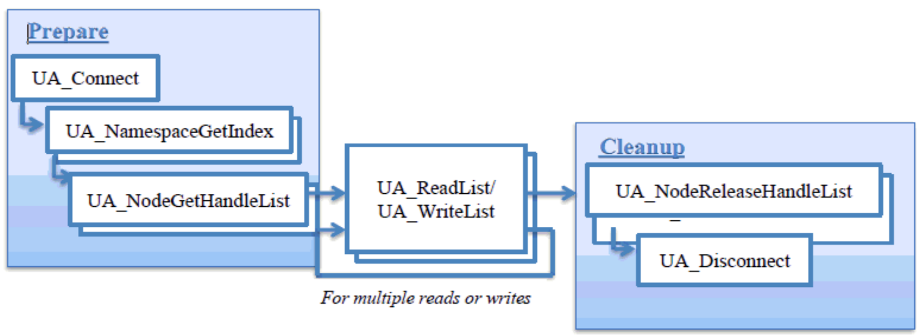
In order to perform an operation like UA_Read, UA_Write, UA_ReadList, UA_WriteList or UA_MethodCall following sequence of calls is required:

Read and Write

UA_Connect is to be performed once for each connection. The UA_NamespaceGetIndex is to be performed once for each namespace. The NodeHdl for a specific node is to be retrieved once. Read and write can be performed as frequent as necessary and permitted by the system. Once the purpose is served release the node handle not required anymore: use UA_ReleaseNodeHdl. Connection handle must be released using UA_Disconnect.



In addition to access the single elements there are function blocks which handle lists:



A list is handled as an array of the related base type (e.g. UANodeID or UANodeAdditionalInfo). Additionally there is a length which holds the number of elements in the array. Although several arrays can be connected to the function block (e.g. node handles and variables in case of UA_ReadList) there is only one length because all arrays have the same number of elements to be processed.

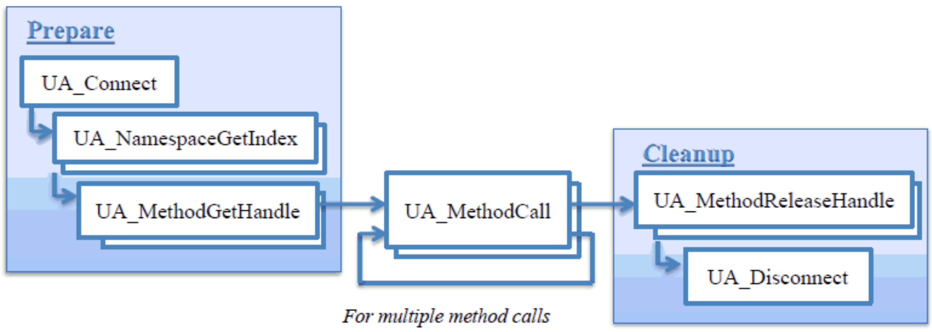
Please note that UA_NodeGetHandleList may not be able to resolve all input UANodeIDs. For such an unresolvable node the function block writes a value 0 (indicating an invalid handle) into the corresponding element of the output array. This output array can be used unchanged for subsequent calls to function blocks UA_ReadList, UA_WriteList and UA_NodeReleaseHandleList which do not perform any operation on nodeIds with the value 0.

Calling Methods

The appropriate sequence for calling methods is shown below. A valid method handle is necessary to call a method.

Successful call of UA_MethodGetHandle will deliver a valid MethodHdl.

Please release the method handle before you disconnect.



OPC UA DataType Reference

OpcUa_DataTypes

string255

TYPE

string255:STRING(255);

END_TYPE

string255List

TYPE

string255List : ARRAY [1..20] OF string255;

END_TYPE

UaUserIdentityToken

TYPE

UaUserIdentityToken :

STRUCT

UAUserIdentityTokenType: INT;

TokenParam1: STRING;

```
        TokenParam2:  STRING;
    END_STRUCT;
END_TYPE

UaLocaleIdList
TYPE
    UaLocaleIdList : ARRAY [1..5] OF STRING;
END_TYPE

UASessionConnectInfo
TYPE
    UASessionConnectInfo :
    STRUCT
        SessionName   :      STRING;
        ApplicationName: STRING;
        SecurityMsgMode: INT;
        SecurityPolicy:      INT;
        CertificateStore: STRING;
        ClientCertificateName: STRING;
        ServerUri: STRING;
        CheckServerCertificate: BOOL;
        TransportProfile: INT;
        UserIdentityToken: UaUserIdentityToken;
        VendorSpecificParameter: STRING;
        SessionTimeout: TIME;
        MonitorConnection: TIME;
        LocaleIDs: UaLocaleIdList;
    END_STRUCT;
END_TYPE
```

UANodeID

```
TYPE
    UANodeID     :
    STRUCT
        NamespaceIndex     :    UINT;
        IdentifierType     :    INT;
        Identifier         :    string255;
    END_STRUCT;
END_TYPE
```

UaNodeIDList

```
TYPE
    UaNodeIDList : ARRAY [1..20] OF UANodeID;
END_TYPE
```

UAIndexRange

```
TYPE
    UAIndexRange  :
    STRUCT
        StartIndex   :    UINT;
        EndIndex     :    UINT;
    END_STRUCT;
END_TYPE
```

UAIndexRangeList

```
TYPE
    UAIndexRangeList : ARRAY [1..20] OF UAIndexRange;
END_TYPE
```

UANodeAdditionalInfo

```
TYPE
    UANodeAdditionalInfo :
    STRUCT
        AttributeID   :     INT;
        IndexRangeCount     :     UINT;
        IndexRangeList : UAIndexRangeList;
    END_STRUCT;
END_TYPE
```

UaNodeAddInfoList

```
TYPE
    UaNodeAddInfoList : ARRAY [1..20] OF UANodeAdditionalInfo;
END_TYPE
```

UAVariant

```
TYPE
    UAVariant     :
    STRUCT
        VariantType   :     INT;
        value_bool    :     BOOL; (* VariantType = 1 *)
        value_sbyte : SINT; (* VariantType = 2 *)
        value_byte    :     USINT; (* VariantType = 3 *)
        value_int16 : INT; (* VariantType = 4 *)
        value_uint16: UINT; (* VariantType = 5 *)
        value_int32 : DINT; (* VariantType = 6 *)
        value_uint32: UDINT; (* VariantType = 7 *)
        value_int64 : LINT; (* VariantType = 8 *)
(*          value_uint64: ULINT;*) (* ULINT Not Yet Available *)
        value_real    :     REAL; (* VariantType = 10 *)
        value_lReal :   LREAL; (* VariantType = 11 *)
```

```
        value_string: string255; (* VariantType = 12 *)
    END_STRUCT;
END_TYPE

UAVariantList
TYPE
    UAVariantList : ARRAY [1..20] OF UAVariant;
END_TYPE

UADateTime
TYPE
    UADateTime    :
    STRUCT
        LowDateTime   :     UDINT;
        HighDateTime  :     UDINT;
    END_STRUCT;
END_TYPE

UaDateTimeList
TYPE
    UaDateTimeList : ARRAY [1..20] OF UADateTime;
END_TYPE

UADataValue
TYPE
    UADataValue   :
    STRUCT
        Value  :      UAVariant;
        StatusCode : UDINT;
        SourceTimeStamp : UADateTime;
```

```
        ServerTimeStamp : UADateTime;
    END_STRUCT;
END_TYPE

UaDataValueList
TYPE
    UaDataValueList : ARRAY [1..20] OF UADataValue;
END_TYPE

UaDWORDList
TYPE
    UaDWORDList : ARRAY [1..20] OF DWORD;
END_TYPE

UAMonitoringSettings
TYPE
UAMonitoringSettings :
STRUCT
DeadbandType : INT;
Deadband : REAL;
SamplingInterval : TIME;
END_STRUCT;
END_TYPE
```

UADataChangeNotification

```
TYPE
UADataChangeNotification :
STRUCT
SubscriptionHdl : UDINT;
NodeHdl : DWORD;
```

```
Count : UINT;
DataValues : UADataValueList;
END_STRUCT;
END_TYPE
```

OPC UA Error Code Reference

See the following table for OPC UA function block error codes definition:

Error Code	Symbolic ID	Description
16#00000000	success	NA
16#00000001	FB_GEN_ERR_INPUT_PARA_INVALID	The input parameter is invalid.
16#00000002	FB_GEN_ERR_RCV_RSP_TIME_OUT	Time out and no response data is received.
16#00000003	FB_GEN_ERR_INTERNAL_TIME_OUT	IPC is time out.
16#00000004	FB_GEN_ERR_INVALID_REQUEST	The request is invalid.
0x00000000	OpcUa_Good	The operation was successful.
0x80000000	OpcUa_Bad	The operation was unsuccessful but no specific reason is known.
0x80010000	OpcUa_BadUnexpectedError	An unexpected error occurred.
0x80020000	OpcUa_BadInternalError	An internal error occurred as a result of a programming or configuration error.
0x80030000	OpcUa_BadOutOfMemory	Not enough memory to complete the operation.
0x80040000	OpcUa_BadResourceUnavailable	An operating system resource is not

Error Code	Symbolic ID	Description
		available.
0x80050000	OpcUa_BadCommunicationError	A low level communication error occurred.
0x80060000	OpcUa_BadEncodingError	Encoding halted because of invalid data in the objects being serialized.
0x80070000	OpcUa_BadDecodingError	Decoding halted because of invalid data in the stream.
0x80080000	OpcUa_BadEncodingLimitsExceeded	The message encoding/decoding limits imposed by the stack have been exceeded.
0x80B80000	OpcUa_BadRequestTooLarge	The request message size exceeds limits set by the server.
0x80B90000	OpcUa_BadResponseTooLarge	The response message size exceeds limits set by the client.
0x80090000	OpcUa_BadUnknownResponse	An unrecognized response was received from the server.
0x800A0000	OpcUa_BadTimeout	The operation timed out.
0x800B0000	OpcUa_BadServiceUnsupported	The server does not support the requested service.
0x800C0000	OpcUa_BadShutdown	The operation was cancelled because the application is shutting down.
0x800D0000	OpcUa_BadServerNotConnected	The operation could not complete because

Error Code	Symbolic ID	Description
		the client is not connected to the server.
0x800E0000	OpcUa_BadServerHalted	The server has stopped and cannot process any requests.
0x800F0000	OpcUa_BadNothingToDo	There was nothing to do because the client passed a list of operations with no elements.
0x80100000	OpcUa_BadTooManyOperations	The request could not be processed because it specified too many operations.
0x80DB0000	OpcUa_BadTooManyMonitoredItems	The request could not be processed because there are too many monitored items in the subscription.
0x80110000	OpcUa_BadDataTypeIdUnknown	The extension object cannot be (de)serialized because the data type id is not recognized.
0x80120000	OpcUa_BadCertificateInvalid	The certificate provided as a parameter is not valid.
0x80130000	OpcUa_BadSecurityChecksFailed	An error occurred verifying security.
0x80140000	OpcUa_BadCertificateTimeInvalid	The Certificate has expired or is not yet valid.
0x80150000	OpcUa_BadCertificateIssuerTimeInvalid	An Issuer Certificate has expired or is not yet valid.

Error Code	Symbolic ID	Description
0x80160000	OpcUa_BadCertificateHostNameInvalid	The HostName used to connect to a Server does not match a HostName in the Certificate.
d 0x80170000	OpcUa_BadCertificateUriInvali	The URI specified in the ApplicationDescription does not match the URI in the Certificate.
0x80180000	OpcUa_BadCertificateUseNotAllowed	The Certificate may not be used for the requested operation.
0x80190000	OpcUa_BadCertificateIssuerUseNotAllowed	The Issuer Certificate may not be used for the requested operation.
0x801A0000	OpcUa_BadCertificateUntrusted	The Certificate is not trusted.
0x801B0000	OpcUa_BadCertificateRevocationUnknown	It was not possible to determine if the Certificate has been revoked.
0x801C0000	OpcUa_ BadCertificateIssuerRevocationUnknown	It was not possible to determine if the Issuer Certificate has been revoked.
0x801D0000	OpcUa_BadCertificateRevoked	The Certificate has been revoked.
0x801E0000	OpcUa_BadCertificateIssuerRevoked	The Issuer Certificate has been revoked.
0x801F0000	OpcUa_BadUserAccessDenied	User does not have permission to perform the requested operation.
0x80200000	OpcUa_BadIdentityTokenInvalid	The user identity token is not valid.

Error Code	Symbolic ID	Description
0x80210000	OpcUa_BadIdentityTokenRejected	The user identity token is valid but the server has rejected it.
0x80220000	OpcUa_BadSecureChannelIdInvalid	The specified secure channel is no longer valid.
0x80230000	OpcUa_BadInvalidTimestamp	The timestamp is outside the range allowed by the server.
0x80240000	OpcUa_BadNonceInvalid	The nonce does appear to be not a random value or it is not the correct length.
0x80250000	OpcUa_BadSessionIdInvalid	The session id is not valid.
0x80260000	OpcUa_BadSessionClosed	The session was closed by the client.
0x80270000	OpcUa_BadSessionNotActivated	The session cannot be used because ActivateSession has not been called.
0x80280000	OpcUa_BadSubscriptionIdInvalid	The subscription id is not valid.
0x802A0000	OpcUa_BadRequestHeaderInvalid	The header for the request is missing or invalid.
0x802B0000	OpcUa_BadTimestampsToReturnInvalid	The timestamps to return parameter is invalid.
0x802C0000	OpcUa_BadRequestCancelledByClient	The request was cancelled by the client.
0x002D0000	OpcUa_GoodSubscriptionTransferred	The subscription was transferred to another session
0x002E0000	OpcUa_GoodCompletesAsynchronously	The processing will

Error Code	Symbolic ID	Description
		complete asynchronously.
0x002F0000	OpcUa_GoodOverload	Sampling has slowed down due to resource limitations.
0x00300000	OpcUa_GoodClamped	The value written was accepted but was clamped
0x80310000	OpcUa_BadNoCommunication	Communication with the data source is d, but not established, and there is no last known value available.
0x80320000	OpcUa_BadWaitingForInitialData	Waiting for the server to obtain values from the underlying data source.
0x80330000	OpcUa_BadNodeIdInvalid	The syntax of the node id is not valid.
0x80340000	OpcUa_BadNodeIdUnknown	The node id refers to a node that does not exist in the server address space.
0x80350000	OpcUa_BadAttributeIdInvalid	The attribute is not supported for the specified Node.
0x80360000	OpcUa_BadIndexRangeInvalid	The syntax of the index range parameter is invalid.
0x80370000	OpcUa_BadIndexRangeNoData	No data exists within the range of indexes specified.
0x80380000	OpcUa_BadDataEncodingInvalid	The data encoding is invalid.
0x80390000	OpcUa_BadDataEncodingUnsupported	The server does not support the requested

Error Code	Symbolic ID	Description
		data encoding for the node.
0x803A0000	OpcUa_BadNotReadable	The access level does not allow reading or subscribing to the Node.
0x803B0000	OpcUa_BadNotWritable	The access level does not allow writing to the Node.
0x803C0000	OpcUa_BadOutOfRange	The value was out of range.
0x803D0000	OpcUa_BadNotSupported	The requested operation is not supported.
0x803E0000	OpcUa_BadNotFound	A requested item was not found or a search operation ended without success.
0x803F0000	OpcUa_BadObjectDeleted	The object cannot be used because it has been deleted.
0x80400000	OpcUa_BadNotImplemented	Requested operation is not implemented.
0x80410000	OpcUa_BadMonitoringModeInvalid	The monitoring mode is invalid.
0x80420000	OpcUa_BadMonitoredItemIdInvalid	The monitoring item id does not refer to a valid monitored item.
0x80430000	OpcUa_BadMonitoredItemFilterInvalid	The monitored item filter parameter is not valid.
0x80440000	OpcUa_ BadMonitoredItemFilterUnsupported	The server does not support the requested monitored item filter.
0x80450000	OpcUa_BadFilterNotAllowed	A monitoring filter

Error Code	Symbolic ID	Description
		cannot be used in combination with the attribute specified.
0x80460000	OpcUa_BadStructureMissing	A mandatory structured parameter was missing or null.
0x80470000	OpcUa_BadEventFilterInvalid	The event filter is not valid.
0x80480000	OpcUa_BadContentFilterInvalid	The content filter is not valid.
0x80C10000	OpcUa_BadFilterOperatorInvalid	An unregognized operator was provided in a filter.
0x80C20000	OpcUa_BadFilterOperatorUnsupported	A valid operator was provided, but the server does not provide support for this filter operator.
0x80C30000	OpcUa_BadFilterOperandCountMismatch	The number of operands provided for the filter operator was less then expected for the operand provided.
0x80490000	OpcUa_BadFilterOperandInvalid	The operand used in a content filter is not valid.
0x80C40000	OpcUa_BadFilterElementInvalid	The referenced element is not a valid element in the content filter.
0x80C50000	OpcUa_BadFilterLiteralInvalid	The referenced literal is not a valid value.
0x804A0000	OpcUa_BadContinuationPointInvalid	The continuation point provide is longer valid.
0x804B0000	OpcUa_BadNoContinuationPoints	The operation could not be processed

Error Code	Symbolic ID	Description
		because all continuation points have been allocated.
0x804C0000	OpcUa_BadReferenceTypeIdInvalid	The operation could not be processed because all continuation points have been allocated.
0x804D0000	OpcUa_BadBrowseDirectionInvalid	The browse direction is not valid.
0x804E0000	OpcUa_BadNodeNotInView	The node is not part of the view.
0x804F0000	OpcUa_BadServerUriInvalid	The ServerUri is not a valid URI.
0x80500000	OpcUa_BadServerNameMissing	No ServerName was specified.
0x80510000	OpcUa_BadDiscoveryUrlMissing	No DiscoveryUrl was specified.
0x80520000	OpcUa_BadSempahoreFileMissing	The semaphore file specified by the client is not valid.
0x80530000	OpcUa_BadRequestTypeInvalid	The security token request type is not valid.
0x80540000	OpcUa_BadSecurityModeRejected	The security mode does not meet the requirements set by the Server.
0x80550000	OpcUa_BadSecurityPolicyRejected	The security policy does not meet the requirements set by the Server.
0x80560000	OpcUa_BadTooManySessions	The maximum number of sessions has been reached.

Error Code	Symbolic ID	Description
0x80570000	OpcUa_BadUserSignatureInvalid	The user token signature is missing or invalid.
0x80580000	OpcUa_BadApplicationSignatureInvalid	The signature generated with the client certificate is missing or invalid.
0x80590000	OpcUa_BadNoValidCertificates	The client did not provide at least one software certificate that is valid and meets the profile requirements for the server.
0x80C60000	OpcUa_BadIdentityChangeNotSupported	The Server does not support changing the user identity assigned to the session.
0x805A0000	OpcUa_BadRequestCancelledByRequest	The request was cancelled by the client with the Cancel service.
0x805B0000	OpcUa_BadParentNodeIdInvalid	The parent node id does not to refer to a valid node.
0x805C0000	OpcUa_BadReferenceNotAllowed	The reference could not be created because it violates constraints imposed by the data model.
0x805D0000	OpcUa_BadNodeIdRejected	The requested node id was reject because it was either invalid or server does not allow node ids to be specified by the client.
0x805E0000	OpcUa_BadNodeIdExists	The requested node id is already used by

Error Code	Symbolic ID	Description
		another node.
0x805F0000	OpcUa_BadNodeClassInvalid	The node class is not valid.
0x80600000	OpcUa_BadBrowseNameInvalid	The browse name is invalid.
0x80610000	OpcUa_BadBrowseNameDuplicated	The browse name is not unique among nodes that share the same relationship with the parent.
0x80620000	OpcUa_BadNodeAttributesInvalid	The node attributes are not valid for the node class.
0x80630000	OpcUa_BadTypeDefinitionInvalid	The type definition node id does not reference an appropriate type node.
0x80640000	OpcUa_BadSourceNodeIdInvalid	The source node id does not reference a valid node.
0x80650000	OpcUa_BadTargetNodeIdInvalid	The target node id does not reference a valid node.
0x80660000	OpcUa_BadDuplicateReferenceNotAllowed	The reference type between the nodes is already d.
0x80670000	OpcUa_BadInvalidSelfReference	The server does not allow this type of self reference on this node.
0x80680000	OpcUa_BadReferenceLocalOnly	The reference type is not valid for a reference to a remote server.
0x80690000	OpcUa_BadNoDeleteRights	The server will not allow the node to be deleted.

Error Code	Symbolic ID	Description
0x40BC0000	OpcUa_UncertainReferenceNotDeleted	The server was not able to delete all target references.
0x806A0000	OpcUa_BadServerIndexInvalid	The server index is not valid.
0x806B0000	OpcUa_BadViewIdUnknown	The view id does not refer to a valid view node.
0x80C90000	OpcUa_BadViewTimestampInvalid	The view timestamp is not available or not supported.
0x80CA0000	OpcUa_BadViewParameterMismatch	The view parameters are not consistent with each other.
0x80CB0000	OpcUa_BadViewVersionInvalid	The view version is not available or not supported.
0x40C00000	OpcUa_UncertainNotAllNodesAvailable	The list of references may not be complete because the underlying system is not available.
0x00BA0000	OpcUa_GoodResultsMayBeIncomplete	The server should have followed a reference to a node in a remote server but did not. The result set may be incomplete.
0x80C80000	OpcUa_BadNotTypeDefinition	The provided Nodeid was not a type definition nodeid.
0x406C0000	OpcUa_UncertainReferenceOutOfServer	One of the references to follow in the relative path references to a node in the address space in another server.

Error Code	Symbolic ID	Description
0x806D0000	OpcUa_BadTooManyMatches	The requested operation has too many matches to return.
0x806E0000	OpcUa_BadQueryTooComplex	The requested operation requires too many resources in the server.
0x806F0000	OpcUa_BadNoMatch	The requested operation has no match to return.
0x80700000	OpcUa_BadMaxAgeInvalid	The max age parameter is invalid.
0x80710000	OpcUa_BadHistoryOperationInvalid	The history details parameter is not valid.
0x80720000	OpcUa_BadHistoryOperationUnsupported	The server does not support the requested operation.
0x80BD0000	OpcUa_BadInvalidTimestampArgument	The d timestamp to return was invalid.
0x80730000	OpcUa_BadWriteNotSupported	The server not does support writing the combination of value, status and timestamps provided.
0x80740000	OpcUa_BadTypeMismatch	The value supplied for the attribute is not of the same type as the attribute's value.
0x80750000	OpcUa_BadMethodInvalid	The method id does not refer to a method for the specified object.
0x80760000	OpcUa_BadArgumentsMissing	The client did not specify all of the input arguments for the

Error Code	Symbolic ID	Description
		method.
0x80770000	OpcUa_BadTooManySubscriptions	The server has reached its maximum number of subscriptions.
0x80780000	OpcUa_BadTooManyPublishRequests	The server has reached the maximum number of queued publish requests.
0x80790000	OpcUa_BadNoSubscription	There is no subscription available for this session.
0x807A0000	OpcUa_BadSequenceNumberUnknown	The sequence number is unknown to the server.
0x807B0000	OpcUa_BadMessageNotAvailable	The requested notification message is no longer available.
0x807C0000	OpcUa_BadInsufficientClientProfile	The Client of the current Session does not support one or more Profiles that are necessary for the Subscription.
0x80BF0000	OpcUa_BadStateNotActive	The sub-state machine is not currently active.
0x807D0000	OpcUa_BadTcpServerTooBusy	The server cannot process the request because it is too busy.
0x807E0000	OpcUa_BadTcpMessageTypeInvalid	The type of the message specified in the header invalid.
0x807F0000	OpcUa_BadTcpSecureChannelUnknown	The SecureChannelId and/or TokenId are not currently in use.
0x80800000	OpcUa_BadTcpMessageTooLarge	The size of the message specified in

Error Code	Symbolic ID	Description
		the header is too large.
0x80810000	OpcUa_BadTcpNotEnoughResources	There are not enough resources to process the request.
0x80820000	OpcUa_BadTcpInternalError	An internal error occurred.
0x80830000	OpcUa_BadTcpEndpointUrlInvalid	The Server does not recognize the QueryString specified.
0x80840000	OpcUa_BadRequestInterrupted	The request could not be sent because of a network interruption.
0x80850000	OpcUa_BadRequestTimeout	Timeout occurred while processing the request.
0x80860000	OpcUa_BadSecureChannelClosed	The secure channel has been closed.
0x80870000	OpcUa_BadSecureChannelTokenUnknown	The token has expired or is not recognized.
0x80880000	OpcUa_BadSequenceNumberInvalid	The sequence number is not valid.
0x80BE0000	OpcUa_BadProtocolVersionUnsupported	The applications do not have compatible protocol versions.
0x80890000	OpcUa_BadConfigurationError	There is a problem with the configuration that affects the usefulness of the value.
0x808A0000	OpcUa_BadNotConnected	The variable should receive its value from another variable, but has never been configured to do so.
0x808B0000	OpcUa_BadDeviceFailure	There has been a

Error Code	Symbolic ID	Description
		failure in the device/data source that generates the value that has affected the value.
0x808C0000	OpcUa_BadSensorFailure	There has been a failure in the sensor from which the value is derived by the device/data source.
0x808D0000	OpcUa_BadOutOfService	The source of the data is not operational.
0x808E0000	OpcUa_BadDeadbandFilterInvalid	The deadband filter is not valid.
0x408F0000	OpcUa_ UncertainNoCommunicationLastUsableValue	Communication to the data source has failed. The variable value is the last value that had a good quality.
0x40900000	OpcUa_UncertainLastUsableValue	Whatever was updating this value has stopped doing so.
0x40910000	OpcUa_UncertainSubstituteValue	The value is an operational value that was manually overwritten.
0x40920000	OpcUa_UncertainInitialValue	The value is an initial value for a variable that normally receives its value from another variable.
0x40930000	OpcUa_UncertainSensorNotAccurate	The value is at one of the sensor limits.
0x40940000	OpcUa_ UncertainEngineeringUnitsExceeded	The value is outside of the range of values d for this parameter.
0x40950000	OpcUa_UncertainSubNormal	The value is derived

Error Code	Symbolic ID	Description
		from multiple sources and has less than the required number of Good sources.
0x00960000	OpcUa_GoodLocalOverride	The value has been overridden.
0x80970000	OpcUa_BadRefreshInProgress	This Condition refresh failed, a Condition refresh operation is already in progress.
0x80980000	OpcUa_BadConditionAlreadyDisabled	This condition has already been disabled.
0x80CC0000	OpcUa_BadConditionAlreadyEnabled	This condition has already been enabled.
0x80990000	OpcUa_BadConditionDisabled	Property not available, this condition is disabled.
0x809A0000	OpcUa_BadEventIdUnknown	The specified event id is not recognized.
0x80BB0000	OpcUa_BadEventNotAcknowledgeable	The event cannot be acknowledged.
0x80CD0000	OpcUa_BadDialogNotActive	The dialog condition is not active.
0x80CE0000	OpcUa_BadDialogResponseInvalid	The response is not valid for the dialog.
0x80CF0000	OpcUa_BadConditionBranchAlreadyAcked	The condition branch has already been acknowledged.
0x80D00000	OpcUa_ BadConditionBranchAlreadyConfirmed	The condition branch has already been confirmed.
0x80D10000	OpcUa_BadConditionAlreadyShelved	The condition has already been shelved.
0x80D20000	OpcUa_BadConditionNotShelved	The condition is not currently shelved.

Error Code	Symbolic ID	Description
0x80D30000	OpcUa_BadShelvingTimeOutOfRange	The shelving time not within an acceptable range.
0x809B0000	OpcUa_BadNoData	No data exists for the requested time range or event filter.
0x80D70000	OpcUa_BadBoundNotFound	No data found to provide upper or lower bound value.
0x80D80000	OpcUa_BadBoundNotSupported	The server cannot retrieve a bound for the variable.
0x809D0000	OpcUa_BadDataLost	Data is missing due to collection started/stopped/lost.
0x809E0000	OpcUa_BadDataUnavailable	Expected data is unavailable for the requested time range due to an un-mounted volume, an off-line archive or tape, or similar reason for temporary unavailability.
0x809F0000	OpcUa_BadEntryExists	The data or event was not successfully inserted because a matching entry exists.
0x80A00000	OpcUa_BadNoEntryExists	The data or event was not successfully updated because no matching entry exists.
0x80A10000	OpcUa_BadTimestampNotSupported	The client requested history using a timestamp format the server does not support (i.e requested ServerTimestamp

Error Code	Symbolic ID	Description
		when server only supports SourceTimestamp).
0x00A20000	OpcUa_GoodEntryInserted	The data or event was successfully inserted into the historical database.
0x00A30000	OpcUa_GoodEntryReplaced	The data or event field was successfully replaced in the historical database.
0x40A40000	OpcUa_UncertainDataSubNormal	The value is derived from multiple values and has less than the required number of Good values.
0x00A50000	OpcUa_GoodNoData	No data exists for the requested time range or event filter.
0x00A60000	OpcUa_GoodMoreData	The data or event field was successfully replaced in the historical database.
0x80D40000	OpcUa_BadAggregateListMismatch	The requested number of Aggregates does not match the requested number of NodeIds.
0x80D50000	OpcUa_BadAggregateNotSupported	The requested Aggregate is not support by the server.
0x80D60000	OpcUa_BadAggregateInvalidInputs	The aggregate value could not be derived due to invalid data inputs.
0x80DA0000	OpcUa_BadAggregateConfigurationRejected	The aggregate configuration is not

Error Code	Symbolic ID	Description
		valid for specified node.
0x00D90000	OpcUa_GoodDataIgnored	The request pecifies fields which are not valid for the EventType or cannot be saved by the historian.
0x00A70000	OpcUa_GoodCommunicationEvent	The communication layer has raised an event.
0x00A80000	OpcUa_GoodShutdownEvent	The system is shutting down.
0x00A90000	OpcUa_GoodCallAgain	The operation is not finished and needs to be called again.
0x00AA0000	OpcUa_GoodNonCriticalTimeout	A non-critical timeout occurred.
0x80AB0000	OpcUa_BadInvalidArgument	One or more arguments are invalid.
0x80AC0000	OpcUa_BadConnectionRejected	Could not establish a network connection to remote server.
0x80AD0000	OpcUa_BadDisconnect	The server has disconnected from the client.
0x80AE0000	OpcUa_BadConnectionClosed	The network connection has been closed.
0x80AF0000	OpcUa_BadInvalidState	The operation cannot be completed because the object is closed, uninitialized or in some other invalid state.
0x80B00000	OpcUa_BadEndOfStream	Cannot move beyond end of the stream.

Error Code	Symbolic ID	Description
0x80B10000	OpcUa_BadNoDataAvailable	No data is currently available for reading from a non-blocking stream.
0x80B20000	OpcUa_BadWaitingForResponse	The asynchronous operation is waiting for a response.
0x80B30000	OpcUa_BadOperationAbandoned	The asynchronous operation was abandoned by the caller.
0x80B40000	OpcUa_BadExpectedStreamToBlock	The stream did not return all data requested (possibly because it is a non-blocking stream).
0x80B50000	OpcUa_BadWouldBlock	Non blocking behaviour is required and the operation would block.
0x80B60000	OpcUa_BadSyntaxError	A value had an invalid syntax.
0x81000000	OpcUa_StartOfStackStatusCodes	Begin of status codes internal to the stack.
0x81010000	OpcUa_BadSignatureInvalid	The message signature is invalid.
0x81040000	OpcUa_BadExtensibleParameterInvalid	The extensible parameter provided is not a valid for the service.
0x81050000	OpcUa_ BadExtensibleParameterUnsupported	The extensible parameter provided is valid but the server does not support it.
0x81060000	OpcUa_BadHostUnknown	The hostname could not be resolved.

Error Code	Symbolic ID	Description
0x81070000	OpcUa_BadTooManyPosts	Too many posts were made to a semaphore.
0x81080000	OpcUa_BadSecurityConfig	The security configuration is not valid.
0x81090000	OpcUa_BadFileNotFound	Invalid file name specified.
0x810A0000	OpcUa_BadContinue	Accept bad result and continue anyway.
0x810B0000	OpcUa_BadHttpMethodNotAllowed	Accept bad result and continue anyway.
0x810C0000	OpcUa_BadFileExists	File exists.

CHAPTER

22 HONUAFBHELPERS

HonUaCallMethod

VAR_INPUT

Parameter	Data type	Description
ConnectionHandle	DWORD	Connection handle obtained from Connection block (e.g., “Connect_SecurityNone” above)
NodeIdentifierObject	UaNodeID	Node ID of the object node whose method is to be called by this block
NodeIdentifierMethod	UaNodeID	Node ID of the method node to be called by this block
InputArguments	UAVariantList	Input arguments for this method. Note that some methods may not require any input arguments.
Done	BOOL	Flag indicating that the method call has completed. This flag will be reset FALSE the next time ExecuteCall is set TRUE.

VAR_OUTPUT

Parameter	Data type	Description
ErrorID	DWORD	Error ID if any, returned by the server when attempting to invoke the Call service.
Error	BOOL	If set, signals that an error occurred when attempting to invoke the Call service
OutputArguments	UAVariantList	Output arguments returned by this method. Note that some methods may not return output arguments
InputArgResults	UaDWORDList	Status code associated with each argument in the InputArguments.

VAR_IN_OUT

Parameter	Data type	Description
ExecuteCall	BOOL	When set TRUE, invokes the method call. Upon completion of 1 method call attempt (successful or unsuccessful) will automatically reset to FALSE.

HonUaConnectSecurityNone

VAR_INPUT

Parameter	Data type	Description
ServerEndpointURL	STRING	e.g., "opc.tcp://192.168.1.30:51210/UA/SampleServer"
SessionName	STRING	Each time Connect executes a new session is created on the server. This name will be associated with that session

VAR_OUTPUT

Parameter	Data type	Description
ConnectionHandle	DWORD	The handle associated with this connection. Handle is valid until Disconnect or DisconnectAll are set.
ErrorID	DWORD	Error ID if any, returned by the server
Error	BOOL	If set, signals that an error occurred when attempting to connect

VAR_IN_OUT

Parameter	Data type	Description
Connect	BOOL	When set TRUE and if ConnectionHandle is zero, initiates a new connection. Upon completion of 1 connection attempt

Parameter	Data type	Description
		(successful or unsuccessful) will automatically reset to FALSE
Disconnect	BOOL	When set TRUE initiates a disconnect of the current ConnectionHandle (as indicated by ConnectionHandle). Upon completion of 1 disconnect attempt (successful or unsuccessful) will automatically reset to FALSE.

HonUaHandleDetector

VAR_INPUT

Parameter	Data type	Description
Enable	BOOL	When set TRUE enables the block functionality. When set FALSE disables the block functionality.
DWORDIn	DWORD	When Enable is set TRUE, the block will monitor DWORDIn for change to 0. If this occurs then SignalOut will be set TRUE.

VAR_OUTPUT

Parameter	Data type	Description
SignalOut	BOOL	See DWORDIn above.

HonUaManageSubscription

VAR_INPUT

Parameter	Data type	Description
ConnectionHandle	DWORD	Connection handle obtained from Connection block (e.g., “Connect_SecurityNone” above)
PublishingInterval	TIME	The publishing interval that should be applied to the subscription.

VAR_OUTPUT

Parameter	Data type	Description
SubscriptionHdl	DWORD	Subscription Handle generated after successful execution of the block where CreateSubscription is set TRUE.
SubscriptionEnabled	BOOL	A flag indicating that the subscription is currently enabled.
ErrorID	DWORD	Error ID if any, returned by the server when attempting to invoke the subscription or monitored item service.
Error	BOOL	If set, signals that an error occurred when attempting to invoke the subscription or monitored item service.

VAR_IN_OUT

Parameter	Data type	Description
CreateSubscription	BOOL	Set to TRUE to create a new subscription. Successful execution will result in non-zero SubscriptionHdl.
DeleteSubscription	BOOL	Set to TRUE to delete an existing subscription. SubscriptionHdl will be set to zero.
EnableSubscription	BOOL	Set the subscription enabled.
DisableSubscription	BOOL	Set the subscription disabled.

HonUaReadNode

VAR_INPUT

Parameter	Data type	Description
ConnectionHandle	DWORD	Connection handle obtained from Connection block (e.g., “Connect_SecurityNone” above)
NodeIdRead	UaNodeID	Node ID whose data value is to be read
IsArray	BOOL	Flag indicating whether or not the NodeIdRead data value is an array

Parameter	Data type	Description
ArrayIndex	UINT	If IsArray is TRUE then this identifies the array index to read.

VAR_OUTPUT

Parameter	Data type	Description
DataStatus	UDINT	Status code associated with the DataValueOut
DataValueOut	UAVariant	Value of the node (attribute 13)
TimeStamp	UADateTime	Source timestamp associated with DataValueOut
ErrorID	DWORD	Error ID if any, returned by the server when attempting to invoke the Read service.
Error	BOOL	If set, signals that an error occurred when attempting to invoke the Read service
ReadEnabled	BOOL	When set, indicates that block is enabled and read service will be called with each task cycle.

VAR_IN_OUT

Parameter	Data type	Description
EnableRead	BOOL	When set TRUE, enables this read block. Read service will be called with each task cycle. See ReadEnabled above to verify that block is enabled.
DisableRead	BOOL	When set TRUE, disables this read block. Read service will not be called with each task cycle. See ReadEnabled above to verify that block is disabled.

HonUaReadNodeList

VAR_INPUT

Parameter	Data type	Description
ConnectionHandle	DWORD	Connection handle obtained from Connection

Parameter	Data type	Description
		block (e.g., “Connect_SecurityNone” above)
NodeIdCount	UINT	The number of Node IDs in NodeIdReadList
NodeIdReadList	UaNodeIDList	Node identifiers of the nodes whose values are to be read by this block (max 20 identifiers).
IsArray	BOOL	Flag indicating whether or not the NodeIdReadList data values are arrays
ArrayIndices	UINTList	If IsArray is TRUE then this identifies the array index for each data value of NodeIdReadList to read. NodeIdReadList and ArrayIndices must contain the same number of elements.

VAR_OUTPUT

Parameter	Data type	Description
ErrorID	DWORD	Error ID if any, returned by the server when attempting to invoke the Read service.
Error	BOOL	If set, signals that an error occurred when attempting to invoke the Read service
ReadEnabled	BOOL	When set, indicates that block is enabled and the Read service will be called with each task cycle.
DataStatusList	UaDWORDList	Status code associated with corresponding value of the DataValueOutList
DataValueOutList	UAVariantList	Value of each node (attribute 13)
TimeStampList	UaDateTimeList	Source Timestamp associated with corresponding value of the DataValueOutList
NodeErrorIdList	UaDWORDList	Error ID associated with corresponding value of the DataValueOutList. Note that ErrorID above will be set if any element of this list has a status other than good.

VAR_IN_OUT

Parameter	Data type	Description
EnableReadList	BOOL	When set TRUE, enables this read block. Read service will be called with each task cycle. See ReadEnabled above to verify that block is enabled.
DisableReadList	BOOL	When set TRUE, disables this read block. Read service will not be called with each task cycle. See ReadEnabled above to verify that block is disabled.

HonUaStateDetector

VAR_INPUT

Parameter	Data type	Description
Enable	BOOL	When set TRUE enables the block functionality. When set FALSE disables the block functionality.
BOOLIn	BOOL	When Enable is set TRUE, the block will monitor BOOLIn for change to FALSE. If this occurs then SignalOut will be set TRUE.

VAR_OUTPUT

Parameter	Data type	Description
SignalOut	BOOL	See BOOLIn above

HonUaSubscribeNode

VAR_INPUT

Parameter	Data type	Description
ConnectionHandle	DWORD	Connection handle obtained from Connection block (e.g., “Connect_SecurityNone” above)
NodeIdSubscribe	UaNodeID	The NodeId of the data variable node which will be

Parameter	Data type	Description
		added as monitored item to the subscription.
IsArray	BOOL	Flag indicating whether or not the NodeIdSubscribe data value is an array.

VAR_OUTPUT

Parameter	Data type	Description
ErrorID	DWORD	Error ID if any, returned by the server when attempting to invoke the subscription or monitored item service.
Error	BOOL	If set, signals that an error occurred when attempting to invoke the subscription or monitored item service.
SubscriptionEnabled	BOOL	A flag indicating that the subscription is currently enabled.
DataChangeNotification	UaDataChangeNotification	Notifications for the subscribed node. A notification will occur when the value or the status of the variable changes.

VAR_IN_OUT

Parameter	Data type	Description
EnableSubscription	BOOL	Set the subscription enabled.
DisableSubscription	BOOL	Set the subscription disabled.

HonUaTranslatePathList

VAR_INPUT

Parameter	Data type	Description
ConnectionHandle	DWORD	Connection handle obtained from Connection

Parameter	Data type	Description
		block (e.g., “Connect_SecurityNone” above)
NodeIdStartNode	UaNodeID	The RelativePathList is evaluated using this node as a starting point
RelativePathList	String255List	Relative paths to the target nodes using NodeIdStartNode as a starting point.
NamespaceUri	STRING	If the substitution token ‘#’ is inserted into the relative paths in RelativePathList, then the block will first acquire the index of this Uri from the server’s namespace table then substitute that index at each ‘#’. For example, if a string in the RelativePathList is "/#:Drum1001/#:LIX001/#:Output" and NameSpaceUri "http://opcfoundation.org/sampleserver" is located at index 4 of the server's Namespace table, then the block will modify the string to "/4:Drum1001/4:LIX001/4:Output" prior to pass to the server for evaluation. NamespaceUri may be set to empty string if no substitution token is supplied in any Relative Path.

VAR_OUTPUT

Parameter	Data type	Description
ErrorID	DWORD	Error ID if any, returned by the server when attempting to invoke the Call service.
Error	BOOL	If set, signals that an error occurred when attempting to invoke the Call service
Done	BOOL	Flag indicating that the function block execution has completed. This flag will be reset FALSE the next time ExecuteTranslate is set TRUE.
NodeIdOutCount	UINT	Number of NodeIDs returned
NodeIdOutList	UaNodeIDList	Node IDs corresponding to the relative paths in RelativePathList
NodeErrorIdList	UaDWORDList	Error ID associated with translating the corresponding relative path to a Node ID. Note that ErrorID above will be set if any element of this list has a status other than good.

VAR_IN_OUT

Parameter	Data type	Description
ExecuteTranslate	BOOL	When set TRUE, initiates the relative path to NodeID translation. Upon completion of 1 such attempt (successful or unsuccessful) will automatically reset to FALSE.

HonUaVariantToString

VAR_INPUT

Parameter	Data type	Description
VariantIn	UAVariant	Variant value (i.e., as returned from the function block “ReadNode”)

VAR_OUTPUT

Parameter	Data type	Description
StringOut	STRING	String representation of VariantIn

HonUaWriteNode

VAR_INPUT

Parameter	Data type	Description
ConnectionHandle	DWORD	Connection handle obtained from Connection block (e.g., “Connect_SecurityNone” above)
NodeIdWrite	UaNodeID	Node ID whose data value is to be written
IsArray	BOOL	Flag indicating whether or not the NodeIdWrite data value is an array
ArrayIndex	UINT	If IsArray is TRUE then this identifies the array index to write.
DataValue	UAVariant	Value to be written (attribute 13)

VAR_OUTPUT

Parameter	Data type	Description
ErrorID	DWORD	Error ID if any, returned by the server when attempting to invoke the Write service.
Error	BOOL	If set, signals that an error occurred when attempting to invoke the Write service
WriteEnabled	BOOL	When set, indicates that block is enabled and write service will be called with each task cycle.

VAR_IN_OUT

Parameter	Data type	Description
EnableWrite	BOOL	When set TRUE, enables this write block. Write service will be called with each task cycle. See WriteEnabled above to verify that block is enabled.
DisableWrite	BOOL	When set TRUE, disables this write block. Write service will not be called with each task cycle. See WriteEnabled above to verify that block is disabled.

HonUaWriteNodeList

VAR_INPUT

Parameter	Data type	Description
ConnectionHandle	DWORD	Connection handle obtained from Connection block (e.g., “Connect_SecurityNone” above)
NodeIdCount	UINT	The number of Node IDs in NodeIdWriteList
NodeIdWriteList	UaNodeIDList	Node identifiers of the nodes whose values are to be written by this block (max 20 identifiers).
IsArray	BOOL	Flag indicating whether or not the NodeIdWriteList data values are arrays
ArrayIndices	UINTList	If IsArray is TRUE then this identifies the array

Parameter	Data type	Description
		index for each data value of NodeIdWriteList to read. NodeIdWriteList and ArrayIndices must contain the same number of elements.
DataValueList	UAVariantList	Values to be written (attribute 13).

VAR_OUTPUT

Parameter	Data type	Description
ErrorID	DWORD	Error ID if any, returned by the server when attempting to invoke the Write service.
Error	BOOL	If set, signals that an error occurred when attempting to invoke the Write service
WriteEnabled	BOOL	When set, indicates that block is enabled and the Write service will be called with each task cycle.
NodeErrorIdList	UaDWORDList	Error ID associated with corresponding value of the DataValueList when attempting to write the value. Note that ErrorID above will be set if any element of this list has a status other than good.

VAR_IN_OUT

Parameter	Data type	Description
EnableWriteList	BOOL	When set TRUE, enables this write block. Write service will be called with each task cycle. See WriteEnabled above to verify that block is enabled.
DisableWriteList	BOOL	When set TRUE, disables this write block. Write service will not be called with each task cycle. See WriteEnabled above to verify that block is disabled.

CHAPTER

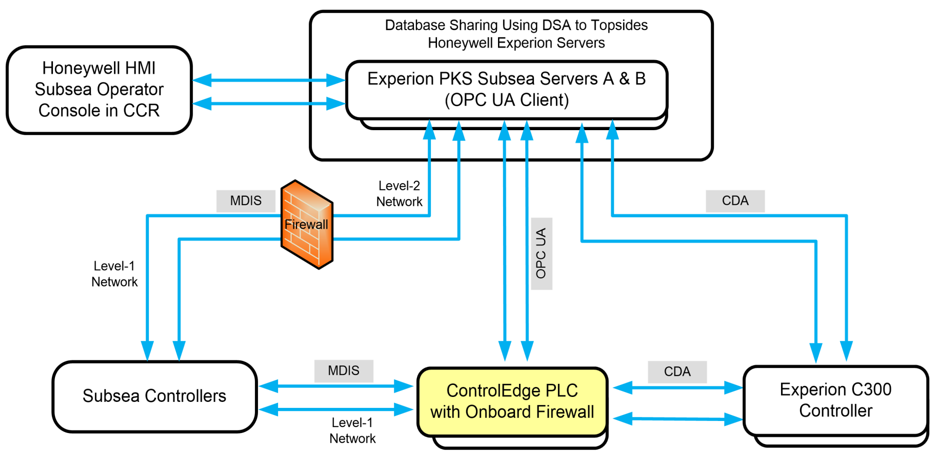
23

MDIS

The MDIS library has a set of OPC UA function blocks representing all the MDIS OPC UA object types as defined in the MDIS OPC UA Companion Specification V1.2. The MDIS OPC UA Object function blocks are used to obtain data from MDIS OPC UA compliant Servers. For each MDIS object type, the specification identifies a set of data variables as well as method definitions. The MDIS function block library incorporates the data variables into each block as function block parameters or 'pins'. Separate method function blocks are provided for each of the methods defined in the specification.

Below is an example architecture with Experion and C300s. Note that ControlEdge PLC with MDIS can be used independent of C300s.

Figure 23-1: Example Experion architecture with MDIS support



The following MDIS function blocks are available:

Function Blocks	Short Description
See MDISDiscreteInstrObj for more information.	This function block is used to create an instance of a Discrete Instrument object.
See MDISDigitalInstrObj for more information.	This function block is used to create an instance of a Digital Instrument object.
See MDISInstrObj for more information.	This function block is used to create an instance of a Instrument object.

Function Blocks	Short Description
See MDISChokeObj for more information.	This function block is used to create an instance of a Choke object.
See MDISValveObj for more information.	This function block is used to create an instance of a Valve object.
See MDISObjEnableDisable for more information.	This function block is used to invoke the EnableDisable method on an object.
See MDISDiscrtInstrWriteVal for more information.	This function block is used to change the value of the 'State' variable on a Discrete Instrument object by invoking the WriteValue Method.
See MDISDigInstrWriteState for more information.	This function block is used to change the value of the 'State' variable on a Digital Instrument object by invoking the WriteState Method.
See MDISInstrWriteValue for more information.	This function block is used to change the value of the 'ProcessVariable' on an Instrument by invoking the WriteValue Method.
See MDISChokeMove for more information.	This function block is used to adjust the opening on a Choke by invoking the Move method.
See MDISChokeStep for more information.	This function block is used to adjust the opening on a Choke by invoking the Step method.
See MDISChokeAbort for more information.	This function block is used to cancel any active Choke Move or Step command by invoking the Abort method.
See MDISChokeSetCalcPos for more information.	This function block is used to overwrite the CalculatedPosition on a Choke by invoking the SetCalculatedPosition method.
See MDISValveMove for more information.	This function block is used to open or close a Valve by invoking the Move method.

There are five MDIS object function blocks. They are Choke, Valve, Instrument, Digital Instrument, and Discrete Instrument. In addition, there are several associated function blocks that enable method invocation on the five MDIS object function blocks. The method invocation function blocks and associated object function blocks are listed below.

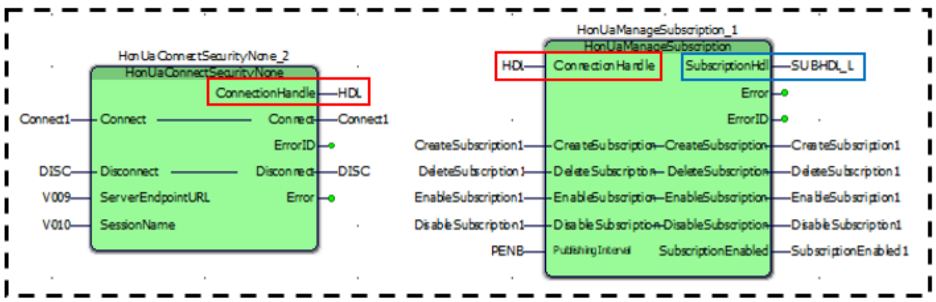
MDIS Object	Object Function Block	Associated Method(s)	Method Function Block
Instrument	MDISInstrObj	Enable-Disable Write Instrument Value	MDISObjEnableDisable MDISInstrWriteValue
Digital Instrument	MDISDigitalInstrObj	Enable-Disable Write Digital Instrument State	MDISObjEnableDisable MDISDigInstrWriteState
Discrete Instrument	MDISDiscreteInstrObj	Enable-Disable Write Discrete Instrument Value	MDISObjEnableDisable MDISDiscrtInstrWriteVal
Valve	MDISValveObj	Enable-Disable Move	MDISObjEnableDisable MDISValveMove
Choke	MDISChokeObj	Enable-Disable Move Step Set Calculated Position Abort	MDISObjEnableDisable MDISChokeMove MDISChokeStep MDISChokeSetCalcPos MDISChokeAbort

All MDIS Object function blocks require a subscription and all MDIS Method Invocation blocks require a connection. Connections and subscriptions are created using function blocks from the OPC UA function block library or optionally from the OPC UA “Helper Block” library. MDIS object and Method Invocation blocks can share a common connection and subscription. Multiple connections are required in the case of multiple target OPC UA servers. Multiple subscriptions may be required depending on project implementation strategies. For example, a subset of MDIS object blocks may have a data freshness requirement of 500ms, others 1000ms and still others 2000ms. In this case, one technique would be to create three subscriptions each with a different publishing interval (i.e., 500ms, 1000ms and 2000ms). The MDIS blocks which require 500ms freshness would then be assigned the 500ms subscription and so on.

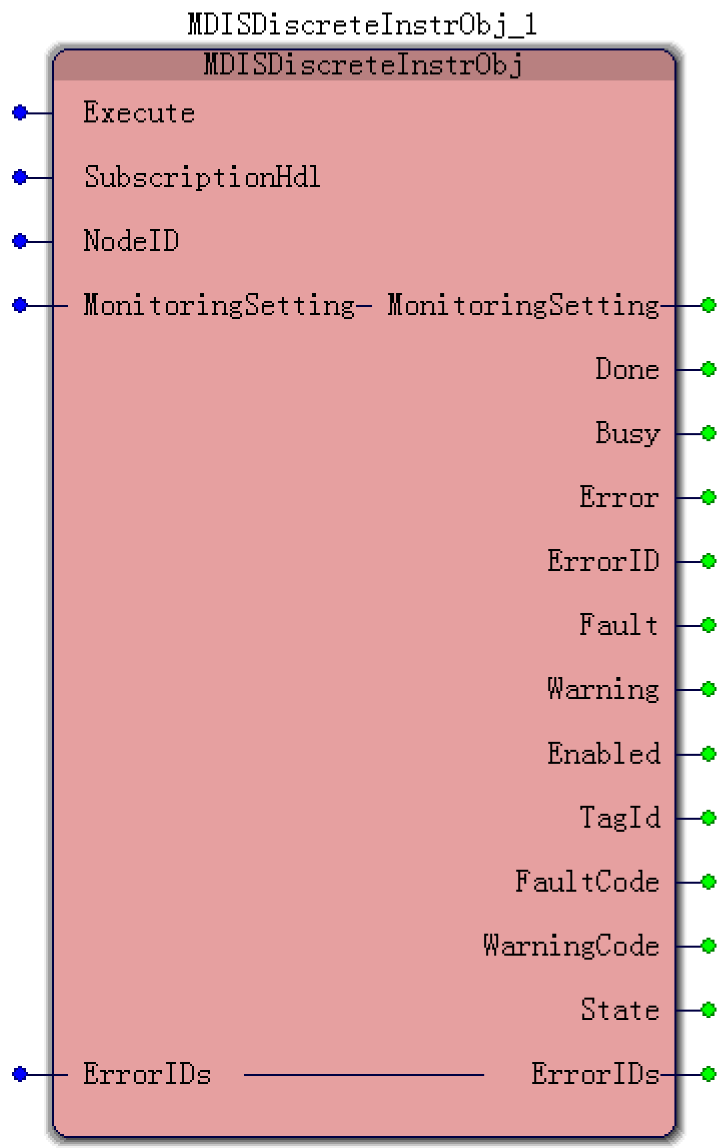
Several examples follow showing possible representations of each MDIS object within a POU. While each of the MDIS objects shown are accompanied by all associated method blocks, this is not required. For example, instruments, including digital and discrete instruments, are often read-only therefore, no write value/state function block would be needed. As another example, the MDIS specification states that the Choke object's Step method is optional therefore some MDIS server vendors may not include this functionality.

Common Connection block and Subscription block

As noted above, MDIS object blocks must be associated with a subscription and MDIS method blocks must be associated with a connection. The example POU below is included for context and shows a possible configuration which yields a connection and a subscription. The connection and subscription "handles" which result from the execution of these blocks are highlighted and referenced in the subsequent MDIS examples.



MDISDiscreteInstrObj



Description

This function block is used to create an instance of a Discrete Instrument object.

Input

Parameter	Data type	Description
Excute	BOOL	When the execute flag is first set and the rising edge is detected, all the defined data variables for the object are added to the subscription. Thereafter, each subsequent rising edge will copy the latest pushed value for all the variables into the output pin representing the variable.
SubscriptionHdl	DWORD	Subscription handle
NodeID	STRUCT	NodeID of Discrete Instrument object. (See UANodeID for STRUCT description)

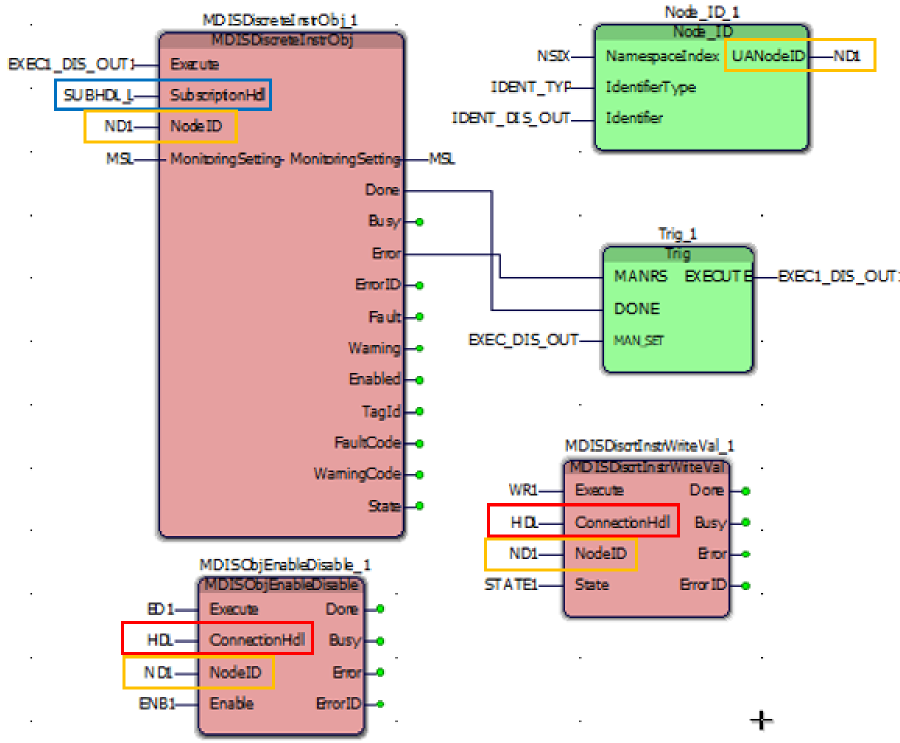
Output

Parameter	Data type	Description
Done	BOOL	The function block has completed its task.
Busy	BOOL	The function block is not finished and is waiting for data value updates.
Error	BOOL	Signals that an error has occurred within the function block. Set to true when ErrorID indicates an error.
ErrorID	DWORD	Error ID if any, returned by the server when monitoring the object.
Fault	BOOL	The status of the object, true if any fault exists.
Warning	BOOL	The status of the object, true if any warning exists.
Enabled	BOOL	Set as true when object is enabled.
TagId	STRING	Unique equipment identifier for the object.
FaultCode	DWORD	Vendor specific fault code. Zero indicates no fault.
WarningCode	DWORD	Vendor specific warning code. Zero indicates no fault.
State	DWORD	State of the Discrete Instrument object.

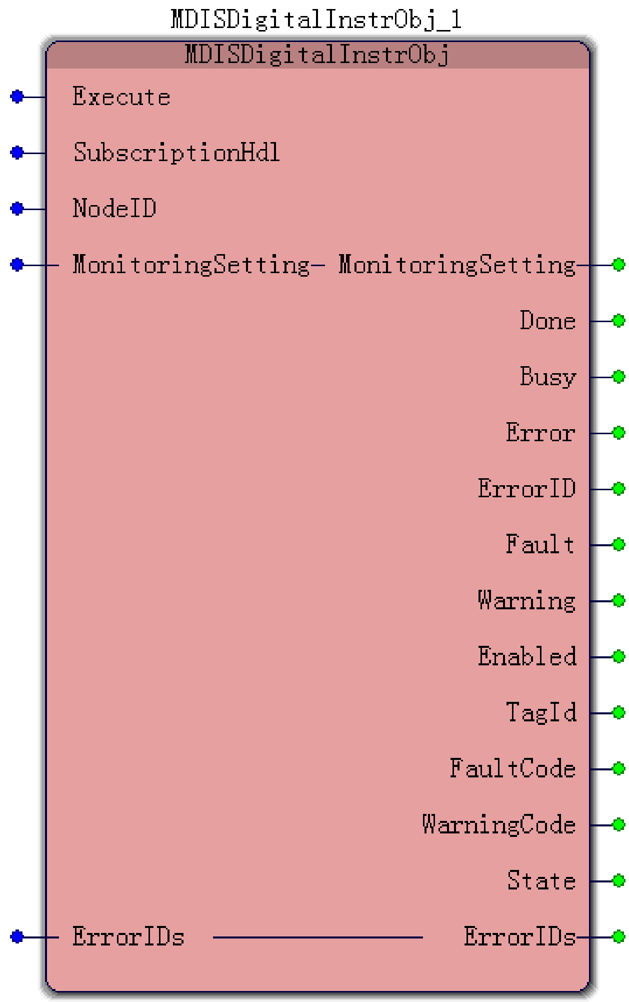
Input and Output

<table>
<tr><th>Parameter</th><th>Data type</th><th>Description</th></tr>
<tr><td>MonitoringSettings</td><td>STRUCT</td><td>See OPC UA DataType Reference for more information.</td></tr>
<tr><td>ErrorIDs</td><td>ARRAY OF DWORD</td><td>Array of DWORD. Contains an error code for each variable listed below:
<table>
<tr><th>Index</th><th>Varaible name</th><th>Mandatory/Optional</th></tr>
<tr><td>1</td><td>Fault</td><td>Mandatory</td></tr>
<tr><td>2</td><td>Warning</td><td>Optional</td></tr>
<tr><td>3</td><td>Enabled</td><td>Optional</td></tr>
<tr><td>4</td><td>TagId</td><td>Optional</td></tr>
<tr><td>5</td><td>FaultCode</td><td>Optional</td></tr>
<tr><td>6</td><td>WarningCode</td><td>Optional</td></tr>
<tr><td>7</td><td>State</td><td>Mandatory</td></tr>
</table>
NOTE: "ErrorIDs" has a pre-defined type "UaDWORDList" which can be found in OpcUa_DataTypes type library.</td></tr>
</table>

Implementation Example



MDISDigitalInstrObj



Description

This function block is used to create an instance of a Digital Instrument object.

Input

Parameter	Data type	Description
Excute	BOOL	When the execute flag is first set and the rising edge is detected, all the defined data variables for the object are added to the subscription. Thereafter, each subsequent

Parameter	Data type	Description
		rising edge will copy the latest pushed value for all the variables into the output pin representing the variable.
SubscriptionHdl	DWORD	Subscription handle
NodeID	STRUCT	NodeID of Digital Instrument object. (See UANodeID for STRUCT description)

Output

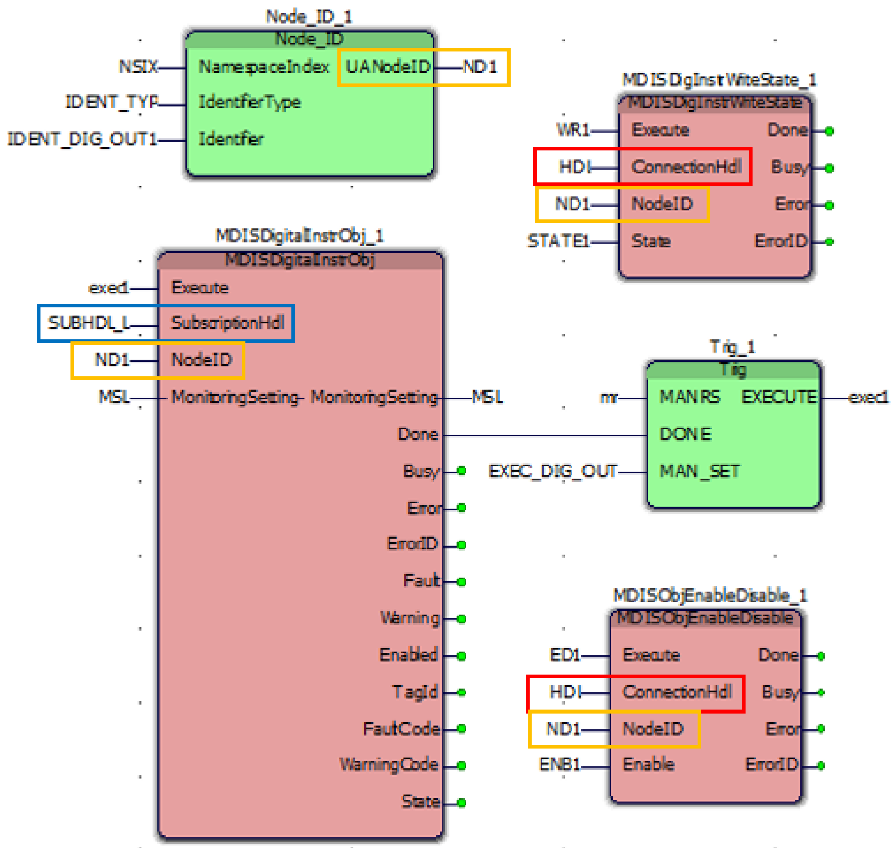
Parameter	Data type	Description
Done	BOOL	The function block has completed its task.
Busy	BOOL	The function block is not finished and is waiting for data value updates.
Error	BOOL	Signals that an error has occurred within the function block. Set to true when ErrorID indicates an error.
ErrorID	DWORD	Error ID if any, returned by the server when monitoring the object.
Fault	BOOL	The status of the object, true if any fault exists.
Warning	BOOL	The status of the object, true if any warning exists.
Enabled	BOOL	Set as true when object is enabled.
TagId	STRING	Unique equipment identifier for the object.
FaultCode	DWORD	Vendor specific fault code. Zero indicates no fault.
WarningCode	DWORD	Vendor specific warning code. Zero indicates no fault.
State	BOOL	State of the Digital Instrument object.

Input and Output

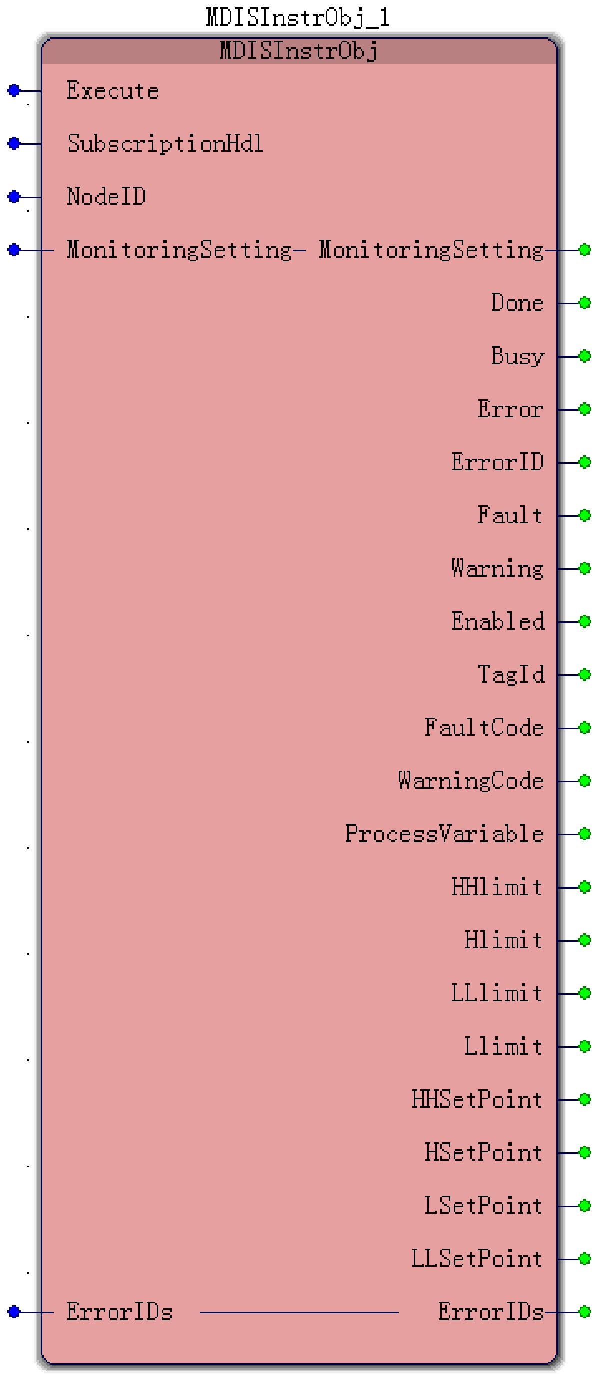
Parameter	Data type	Description
MonitoringSettings	STRUCT	See OPC UA DataType Reference for more information.
ErrorIDs	ARRAY OF DWORD	Array of DWORD. Contains an error code for each variable listed below:

<table>
<tr><th>Parameter</th><th>Data type</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>Index</th><th>Varaible name</th><th>Mandatory/Optional</th></tr>
<tr><td>1</td><td>Fault</td><td>Mandatory</td></tr>
<tr><td>2</td><td>Warning</td><td>Optional</td></tr>
<tr><td>3</td><td>Enabled</td><td>Optional</td></tr>
<tr><td>4</td><td>TagId</td><td>Optional</td></tr>
<tr><td>5</td><td>FaultCode</td><td>Optional</td></tr>
<tr><td>6</td><td>WarningCode</td><td>Optional</td></tr>
<tr><td>7</td><td>State</td><td>Mandatory</td></tr>
</table>
NOTE: "ErrorIDs" has a pre-defined type "UaDWORDList" which can be found in OpcUa_DataTypes type library.
</td></tr>
</table>

Implementation Example



MDISInstrObj



Description

This function block is used to create an instance of a Instrument object.

Input

Parameter	Data type	Description
Excute	BOOL	When the execute flag is first set and the rising edge is detected, all the defined data variables for the object are added to the subscription. Thereafter, each subsequent rising edge will copy the latest pushed value for all the variables into the output pin representing the variable.
SubscriptionHdl	DWORD	Subscription handle
NodeID	STRUCT	NodeID of Instrument object. (See UANodeID for STRUCT description)

Output

Parameter	Data type	Description
Done	BOOL	The function block has completed its task.
Busy	BOOL	The function block is not finished and is waiting for data value updates.
Error	BOOL	Signals that an error has occurred within the function block. Set to true when ErrorID indicates an error.
ErrorID	DWORD	Error ID if any, returned by the server when monitoring the object.
Fault	BOOL	The status of the object, true if any fault exists.
Warning	BOOL	The status of the object, true if any warning exists.
Enabled	BOOL	Set as true when object is enabled.
TagId	STRING	Unique equipment identifier for the object.
FaultCode	DWORD	Vendor specific fault code. Zero indicates no fault.
WarningCode	DWORD	Vendor specific warning code. Zero indicates no fault.
ProcessVariable	REAL	Value of the Instrument.

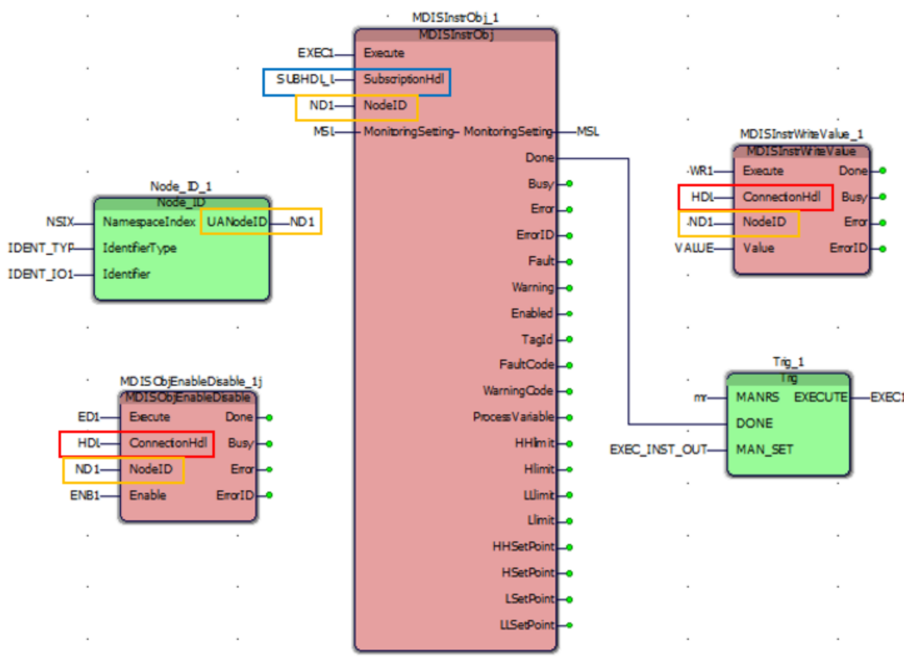
Parameter	Data type	Description
HHlimit	BOOL	HH state of the Instrument.
Hlimit	BOOL	H state of the Instrument.
Llimit	BOOL	L state of the Instrument.
LLlimit	BOOL	LL state of the Instrument.
HHSetPoint	REAL	HHSetPoint value
HSetPoint	REAL	HSetPoint value
LSetPoint	REAL	LSetPoint value
LLSetPoint	REAL	LLSetPoint value

Input and Output

<table>
<tr><th>Parameter</th><th>Data type</th><th>Description</th></tr>
<tr><td>MonitoringSettings</td><td>STRUCT</td><td>See OPC UA DataType Reference for more information.</td></tr>
<tr><td>ErrorIDs</td><td>ARRAY OF DWORD</td><td>Array of DWORD. Contains an error code for each variable listed below:
<table>
<tr><th>Index</th><th>Variable name</th><th>Mandatory/Optional</th></tr>
<tr><td>1</td><td>Fault</td><td>Mandatory</td></tr>
<tr><td>2</td><td>Warning</td><td>Optional</td></tr>
<tr><td>3</td><td>Enabled</td><td>Optional</td></tr>
<tr><td>4</td><td>TagId</td><td>Optional</td></tr>
<tr><td>5</td><td>FaultCode</td><td>Optional</td></tr>
<tr><td>6</td><td>WarningCode</td><td>Optional</td></tr>
<tr><td>7</td><td>ProcessVariable</td><td>Mandatory</td></tr>
<tr><td>8</td><td>HHlimit</td><td>Optional</td></tr>
<tr><td>9</td><td>Hlimit</td><td>Optional</td></tr>
<tr><td>10</td><td>Llimit</td><td>Optional</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Parameter</th><th>Data type</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>Index</th><th>Variable name</th><th>Mandatory/Optional</th></tr>
<tr><td>11</td><td>LLlimit</td><td>Optional</td></tr>
<tr><td>12</td><td>HHSetPoint</td><td>Optional</td></tr>
<tr><td>13</td><td>HSetPoint</td><td>Optional</td></tr>
<tr><td>14</td><td>LSetPoint</td><td>Optional</td></tr>
<tr><td>15</td><td>LLSetPoint</td><td>Optional</td></tr>
</table>
NOTE: "ErrorIDs" has a pre-defined type "UaDWORDList" which can be found in OpcUa_DataTypes type library.
</td></tr>
</table>

Implementation Example



MDISChokeObj

Description

This function block is used to create an instance of a Choke object.

Input

Parameter	Data type	Description
Excute	BOOL	When the execute flag is first set and the rising edge is detected, all the defined data variables for the object are added to the subscription. Thereafter, each subsequent rising edge will copy the latest pushed value for all the variables into the output pin representing the variable.
SubscriptionHdl	DWORD	Subscription handle
NodeID	STRUCT	NodeID of Choke object. (See UANodeID for STRUCT description)

Output

Parameter	Data type	Description
Done	BOOL	The function block has completed its task.
Busy	BOOL	The function block is not finished and is waiting for data value updates.
Error	BOOL	Signals that an error has occurred within the function block. Set to true when ErrorID indicates an error.
ErrorID	DWORD	Error ID if any, returned by the server when monitoring the object.
Fault	BOOL	The status of the object, true if any fault exists.
Warning	BOOL	The status of the object, true if any warning exists.
Enabled	BOOL	Set as true when object is enabled.
TagId	STRING	Unique equipment identifier for the object.
FaultCode	DWORD	Vendor specific fault code. Zero indicates no fault.
WarningCode	DWORD	Vendor specific warning code. Zero indicates no fault.
CalculatedPosition	REAL	A floating-point number that represents the estimated percent open of the choke.

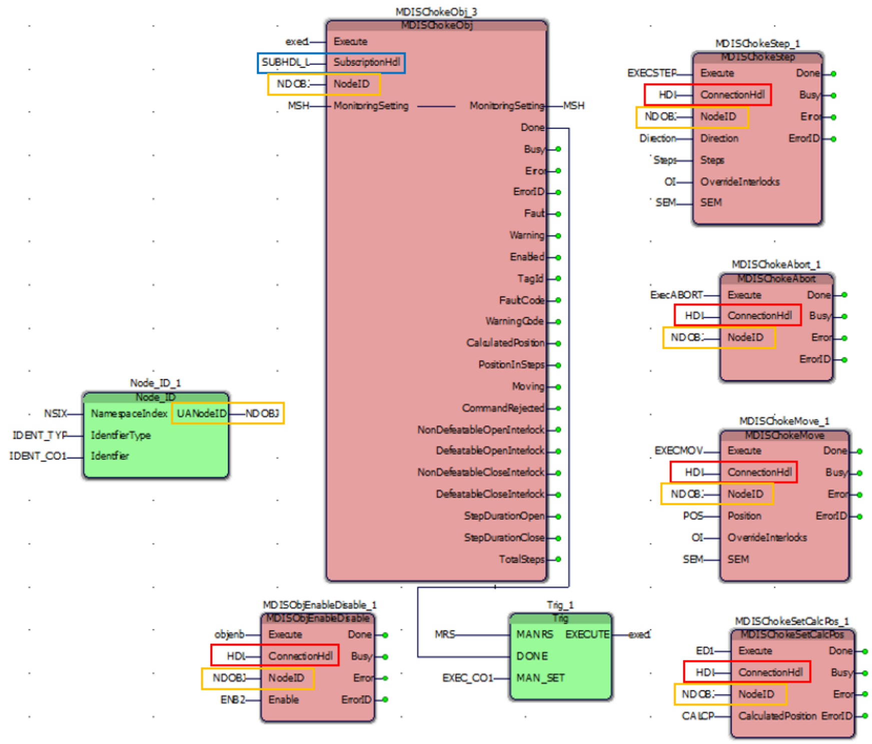
Parameter	Data type	Description
PositionInSteps	INT	An int16 that represents position in steps for the choke.
Moving	DINT	An enumeration indicating the confirmed operation of the choke. Possible status is 1 (Moving) and 2(Stopped).
CommandRejected	BOOL	A flag that, if set to True, indicates that the choke has rejected the last command issued to it.
NonDefeatableOpenInterlock	BOOL	If set to TRUE, open choke command is interlocked and cannot be overridden.
DefeatableOpenInterlock	BOOL	If set to TRUE, open choke command is interlocked and can be overridden.
NonDefeatableCloseInterlock	BOOL	If set to TRUE, close choke command is interlocked and cannot be overridden.
DefeatableCloseInterlock	BOOL	If set to TRUE, close choke command is interlocked and can be overridden.
StepDurationOpen	LREAL	This is the time in milliseconds for the choke to open one step.
StepDurationClose	LREAL	This is the time in milliseconds for the choke to close one step.
TotalSteps	WORD	Max steps of a choke.

Input and Output

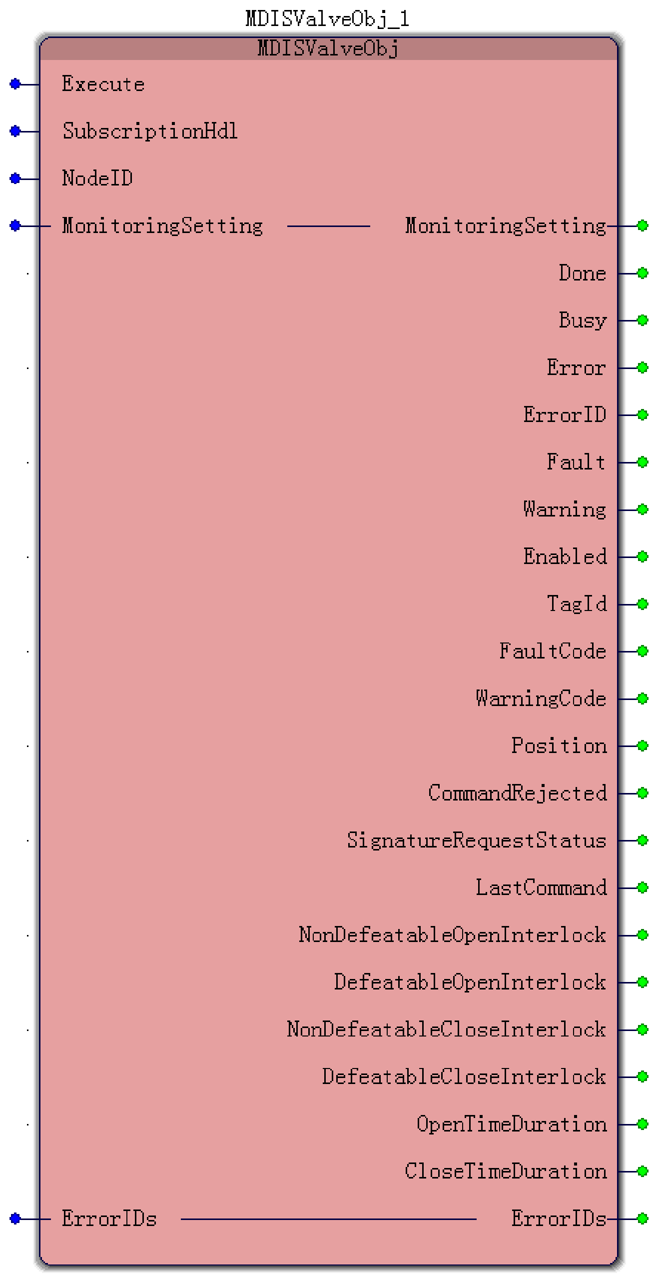
Parameter	Data type	Description
MonitoringSettings	STRUCT	See OPC UA DataType Reference for more information.
ErrorIDs	ARRAY OF DWORD	Array of DWORD. Contains an error code for each variable listed below:

<table>
<tr><th>Parameter</th><th>Data type</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>Index</th><th>Variable name</th><th>Mandatory/Optional</th></tr>
<tr><td>1</td><td>Fault</td><td>Mandatory</td></tr>
<tr><td>2</td><td>Warning</td><td>Optional</td></tr>
<tr><td>3</td><td>Enabled</td><td>Optional</td></tr>
<tr><td>4</td><td>TagId</td><td>Optional</td></tr>
<tr><td>5</td><td>FaultCode</td><td>Optional</td></tr>
<tr><td>6</td><td>WarningCode</td><td>Optional</td></tr>
<tr><td>7</td><td>CalculatedPosition</td><td>Mandatory</td></tr>
<tr><td>8</td><td>PositionInSteps</td><td>Optional</td></tr>
<tr><td>9</td><td>Moving</td><td>Mandatory</td></tr>
<tr><td>10</td><td>CommandRejected</td><td>Optional</td></tr>
<tr><td>11</td><td>NonDefeatableOpenInterlock</td><td>Optional</td></tr>
<tr><td>12</td><td>DefeatableOpenInterlock</td><td>Optional</td></tr>
<tr><td>13</td><td>NonDefeatableCloseInterlock</td><td>Optional</td></tr>
<tr><td>14</td><td>DefeatableCloseInterlock</td><td>Optional</td></tr>
<tr><td>15</td><td>StepDurationOpen</td><td>Optional</td></tr>
<tr><td>16</td><td>StepDurationClose</td><td>Optional</td></tr>
<tr><td>17</td><td>TotalSteps</td><td>Optional</td></tr>
</table>
NOTE: "ErrorIDs" has a pre-defined type "UaDWORDList" which can be found in OpcUa_DataTypes type library.
</td></tr>
</table>

Implementation Example



MDISValveObj



Description

This function block is used to create an instance of a Valve object.

Input

Parameter	Data type	Description
Excute	BOOL	When the execute flag is first set and the rising edge is detected, all the defined data variables for the object are added to the subscription. Thereafter, each subsequent rising edge will copy the latest pushed value for all the variables into the output pin representing the variable.
SubscriptionHdl	DWORD	Subscription handle
NodeID	STRUCT	NodeID of Valve object. (See UANodeID for STRUCT description)

Output

Parameter	Data type	Description
Done	BOOL	The function block has completed its task.
Busy	BOOL	The function block is not finished and is waiting for data value updates.
Error	BOOL	Signals that an error has occurred within the function block. Set to true when ErrorID indicates an error.
ErrorID	DWORD	Error ID if any, returned by the server when monitoring the object.
Fault	BOOL	The status of the object, true if any fault exists.
Warning	BOOL	The status of the object, true if any warning exists.
Enabled	BOOL	Set as true when object is enabled.
TagId	STRING	Unique equipment identifier for the object.
FaultCode	DWORD	Vendor specific fault code. Zero indicates no fault.
WarningCode	DWORD	Vendor specific warning code. Zero indicates no fault.
Position	DINT	Current position of the valve.

<table>
<tr><th>Parameter</th><th>Data type</th><th>Description</th></tr>
<tr><td></td><td></td><td><table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>The Valve is Closed.</td></tr><tr><td>2</td><td>The Valve is Open.</td></tr><tr><td>4</td><td>The Valve is Moving.</td></tr><tr><td>8</td><td>The Valve is in an unknown state. This value can be used when a subsea vendor does not have any last command information and does not know the state of the valve.</td></tr></table></td></tr>
<tr><td>CommandRejected</td><td>BOOL</td><td>A flag that, if set to True, indicates that the valve has rejected the last command issued to it.</td></tr>
<tr><td>SignatureRequestStatus</td><td>DINT</td><td>Status of the current signature request.</td></tr>
<tr><td>LastCommand</td><td>DINT</td><td>Last command sent to the equipment.<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Valve Close command</td></tr><tr><td>2</td><td>Valve Open command</td></tr><tr><td>4</td><td>No known command has been sent to the valve. The initial setting on start-up of a server.</td></tr></table></td></tr>
<tr><td>NonDefeatableOpenInterlock</td><td>BOOL</td><td>If set to TRUE, open valve command is interlocked and cannot be overridden.</td></tr>
<tr><td>DefeatableOpenInterlock</td><td>BOOL</td><td>If set to TRUE, open valve command is interlocked and can be overridden.</td></tr>
<tr><td>NonDefeatableCloseInterlock</td><td>BOOL</td><td>If set to TRUE, close valve command is interlocked and cannot be overridden.</td></tr>
<tr><td>DefeatableCloseInterlock</td><td>BOOL</td><td>If set to TRUE, close valve command is interlocked and can be overridden.</td></tr>
<tr><td>OpenTimeDuration</td><td>LREAL</td><td>This is the estimated time in milliseconds to travel to open position.</td></tr>
</table>

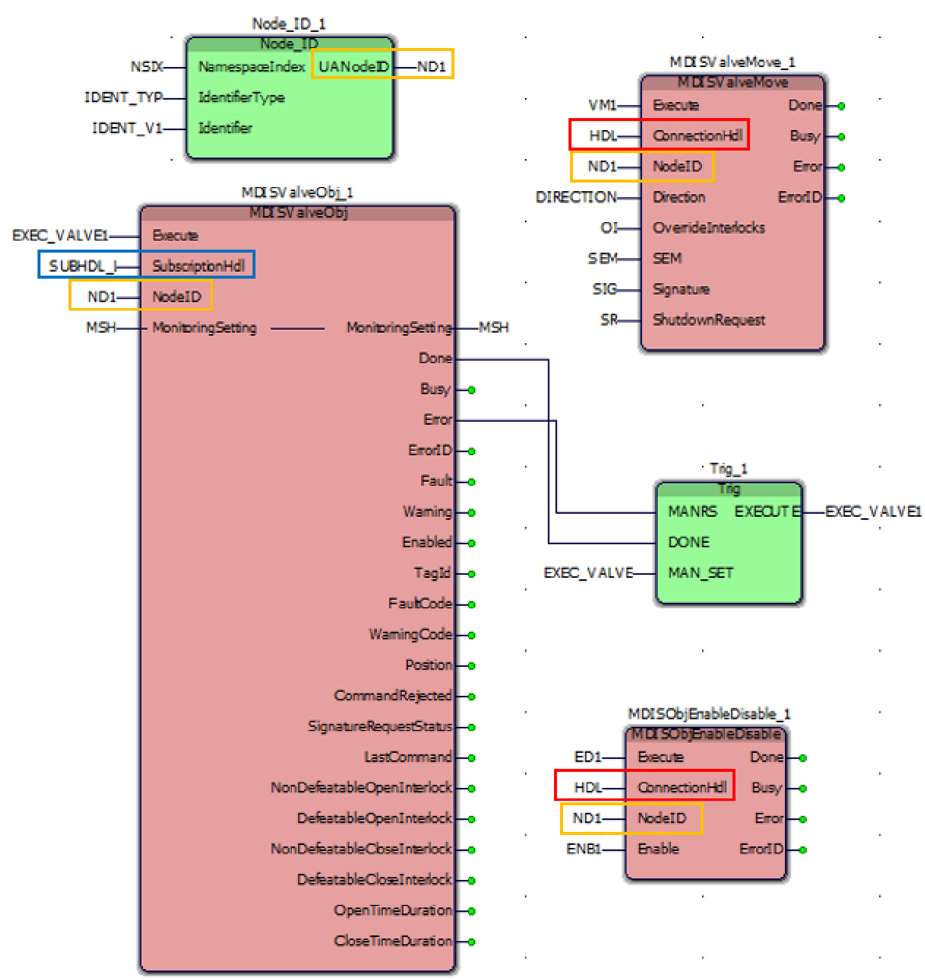
Parameter	Data type	Description
CloseTimeDuration	LREAL	This is the estimated time in milliseconds to travel to close position.

Input and Output

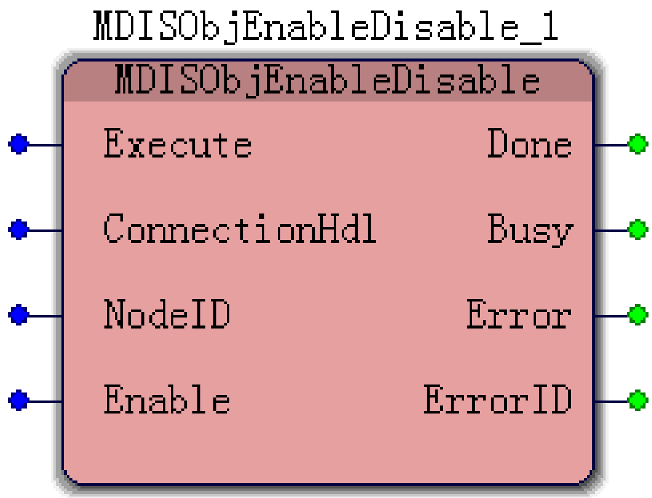
<table>
<tr><th>Parameter</th><th>Data type</th><th>Description</th></tr>
<tr><td>MonitoringSettings</td><td>STRUCT</td><td>See OPC UA DataType Reference for more information.</td></tr>
<tr><td>ErrorIDs</td><td>ARRAY OF DWORD</td><td>Array of DWORD. Contains an error code for each variable listed below:
<table>
<tr><th>Index</th><th>Variable name</th><th>Mandatory/Optional</th></tr>
<tr><td>1</td><td>Fault</td><td>Mandatory</td></tr>
<tr><td>2</td><td>Warning</td><td>Optional</td></tr>
<tr><td>3</td><td>Enabled</td><td>Optional</td></tr>
<tr><td>4</td><td>TagId</td><td>Optional</td></tr>
<tr><td>5</td><td>FaultCode</td><td>Optional</td></tr>
<tr><td>6</td><td>WarningCode</td><td>Optional</td></tr>
<tr><td>7</td><td>Position</td><td>Mandatory</td></tr>
<tr><td>8</td><td>CommandRejected</td><td>Optional</td></tr>
<tr><td>9</td><td>SignatureRequestStatus</td><td>Optional</td></tr>
<tr><td>10</td><td>LastCommand</td><td>Optional</td></tr>
<tr><td>11</td><td>NonDefeatableOpenInterlock</td><td>Optional</td></tr>
<tr><td>12</td><td>DefeatableOpenInterlock</td><td>Optional</td></tr>
<tr><td>13</td><td>NonDefeatableCloseInterlock</td><td>Optional</td></tr>
<tr><td>14</td><td>DefeatableCloseInterlock</td><td>Optional</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Parameter</th><th>Data type</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>Index</th><th>Variable name</th><th>Mandatory/Optional</th></tr>
<tr><td>15</td><td>OpenTimeDuration</td><td>Optional</td></tr>
<tr><td>16</td><td>CloseTimeDuration</td><td>Optional</td></tr>
</table>
NOTE: "ErrorIDs" has a pre-defined type "UaDWORDList" which can be found in OpcUa_DataTypes type library.
</td></tr>
</table>

Implementation Example



MDISObjEnableDisable



Description

This function block is used to invoke the EnableDisable method on an object. An instance of this function block must be added for every object that should be enabled or disabled. The function block accepts the NodeID of the object to be enabled (or disabled) as input.

Input

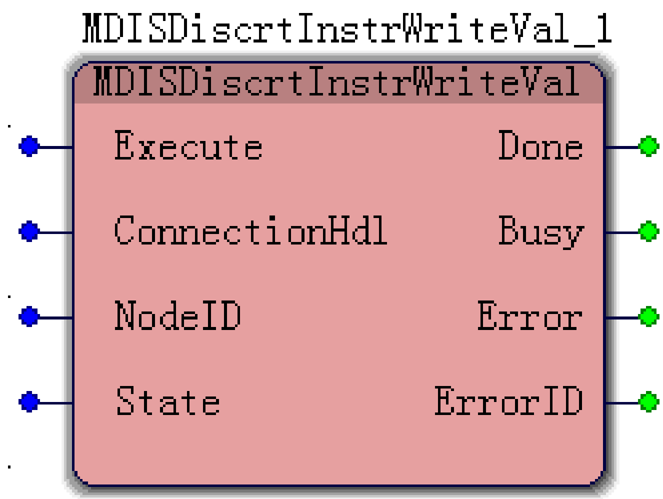
Parameter	Data type	Description
Excute	BOOL	The function block performs its task on rising edge.
ConnectionHdl	DWORD	Connection handle obtained by calling UaConnect
NodeID	STRUCT	NodeID of MDIS object to enable or disable. (See UANodeID for STRUCT description)
Enable	BOOL	Set to true to Enable

Output

Parameter	Data type	Description
Done	BOOL	The function block has completed its task.

Parameter	Data type	Description
Busy	BOOL	The function block is not finished.
Error	BOOL	Signals that an error has occurred. Set to true when ErrorID indicates an error.
ErrorID	DWORD	Error ID if any, returned by the server when EnableDisable method is called.

MDISDiscrtInstrWriteVal



Description

This function block is used to change the value of the 'State' variable on a Discrete Instrument object by invoking the WriteValue Method.

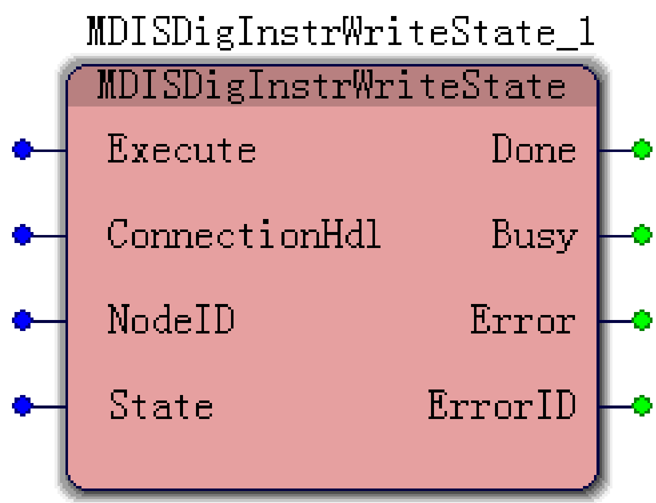
Input

Parameter	Data type	Description
Excute	BOOL	The function block performs its task on rising edge.
ConnectionHdl	DWORD	Connection handle obtained by calling UaConnect
NodeID	STRUCT	NodeID of Discrete Instrument object. (See UANodeID for STRUCT description)
State	DWORD	Value to write to the 'State' variable

Output

Parameter	Data type	Description
Done	BOOL	The function block has completed its task.
Busy	BOOL	The function block is not finished.
Error	BOOL	Signals that an error has occurred. Set to true when ErrorID indicates an error.
ErrorID	DWORD	Error ID if any, returned by the server when WriteValue method is called.

MDISDigInstrWriteState



Description

This function block is used to change the value of the 'State' variable on a Digital Instrument object by invoking the WriteState Method.

Input

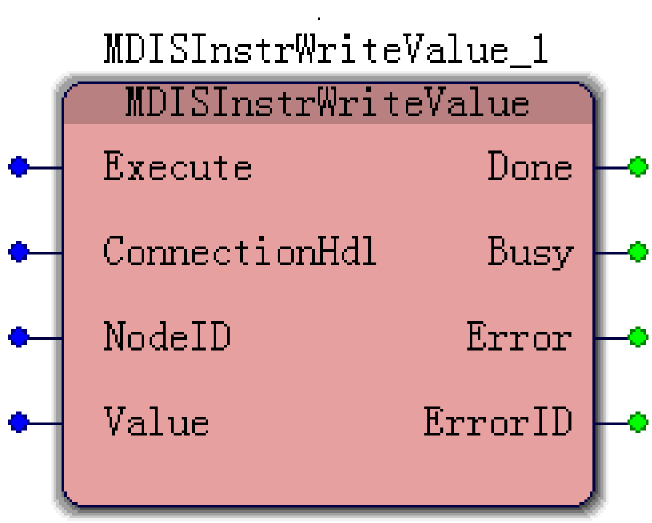
Parameter	Data type	Description
Excute	BOOL	The function block performs its task on rising edge.

Parameter	Data type	Description
ConnectionHdl	DWORD	Connection handle obtained by calling UaConnect
NodeID	STRUCT	NodeID of Digital Instrument object. (See UANodeID for STRUCT description)
State	BOOL	Value to write to the 'State' variable

Output

Parameter	Data type	Description
Done	BOOL	The function block has completed its task.
Busy	BOOL	The function block is not finished.
Error	BOOL	Signals that an error has occurred. Set to true when ErrorID indicates an error.
ErrorID	DWORD	Error ID if any, returned by the server when WriteState method is called.

MDISInstrWriteValue



Description

This function block is used to change the value of the 'ProcessVariable' on an Instrument by invoking the WriteValue Method.

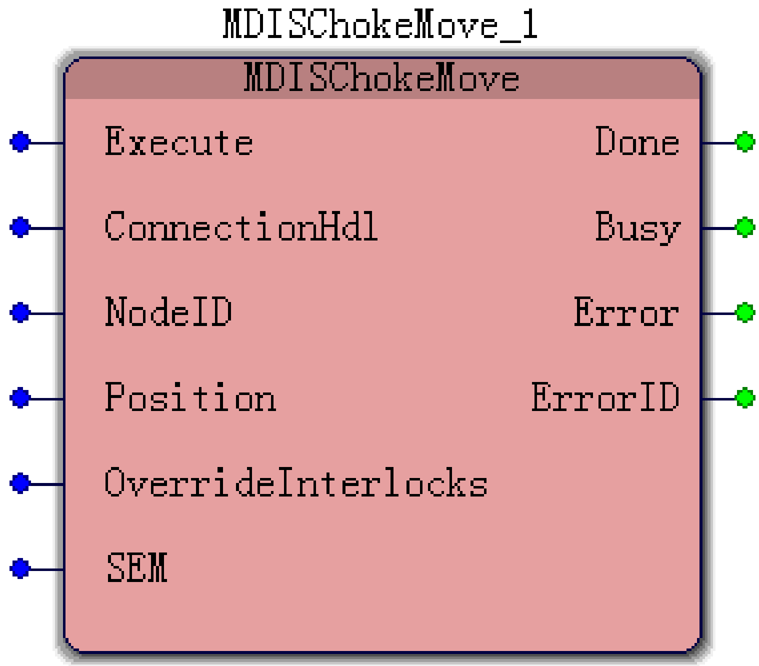
Input

Parameter	Data type	Description
Excute	BOOL	The function block performs its task on rising edge.
ConnectionHdl	DWORD	Connection handle obtained by calling UaConnect
NodeID	STRUCT	NodeID of Instrument object. (See UANodeID for STRUCT description)
Value	REAL	Value to write to the 'ProcessVariable'

Output

Parameter	Data type	Description
Done	BOOL	The function block has completed its task.
Busy	BOOL	The function block is not finished.
Error	BOOL	Signals that an error has occurred. Set to true when ErrorID indicates an error.
ErrorID	DWORD	Error ID if any, returned by the server when WriteValue method is called.

MDISChokeMove



Description

This function block is used to adjust the opening on a Choke by invoking the Move method.

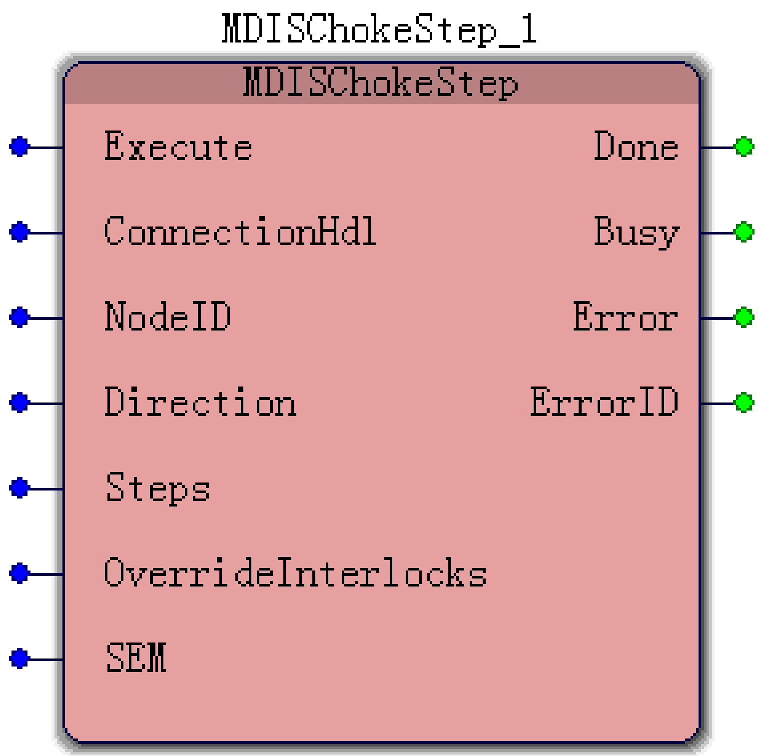
Input

Parameter	Data type	Description
Excute	BOOL	The function block performs its task on rising edge.
ConnectionHdl	DWORD	Connection handle obtained by calling UaConnect
NodeID	STRUCT	NodeID of the Choke object. (See UANodeID for STRUCT description)
Position	REAL	A number indicating the percent by which to move the choke.
OverrideInterlocks	BOOL	If set to 'True', overrides any defeatable interlocks
SEM	DINT	SEM to which command is sent. 1(SEM_A), 2(SEM_B), 4(AUTO)

Output

Parameter	Data type	Description
Done	BOOL	The function block has completed its task.
Busy	BOOL	The function block is not finished.
Error	BOOL	Signals that an error has occurred. Set to true when ErrorID indicates an error.
ErrorID	DWORD	Error ID if any, returned by the server when Move method is called.

MDISChokeStep



Description

This function block is used to adjust the opening on a Choke by invoking the Step method.

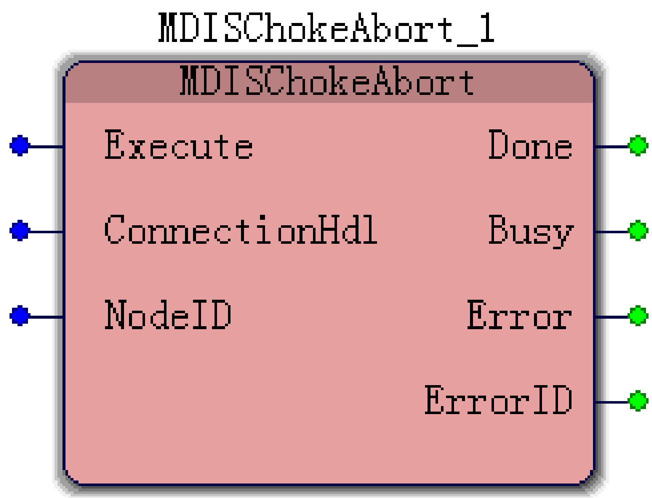
Input

Parameter	Data type	Description
Excute	BOOL	The function block performs its task on rising edge.
ConnectionHdl	DWORD	Connection handle obtained by calling UaConnect
NodeID	STRUCT	NodeID of the Choke object. (See UANodeID for STRUCT description)
Direction	DINT	Open or close command: 1 (Close), 2(Open)
Steps	UINT	Number of steps to open or close the choke
OverrideInterlocks	BOOL	If set to ‘True’, overrides any defeatable interlocks
SEM	DINT	SEM to which command is sent. 1(SEM_A), 2(SEM_B), 4(AUTO)

Output

Parameter	Data type	Description
Done	BOOL	The function block has completed its task.
Busy	BOOL	The function block is not finished.
Error	BOOL	Signals that an error has occurred. Set to true when ErrorID indicates an error.
ErrorID	DWORD	Error ID if any, returned by the server when Step method is called.

MDISChokeAbort



Description

This function block is used to cancel any active Choke Move or Step command by invoking the Abort method.

Input

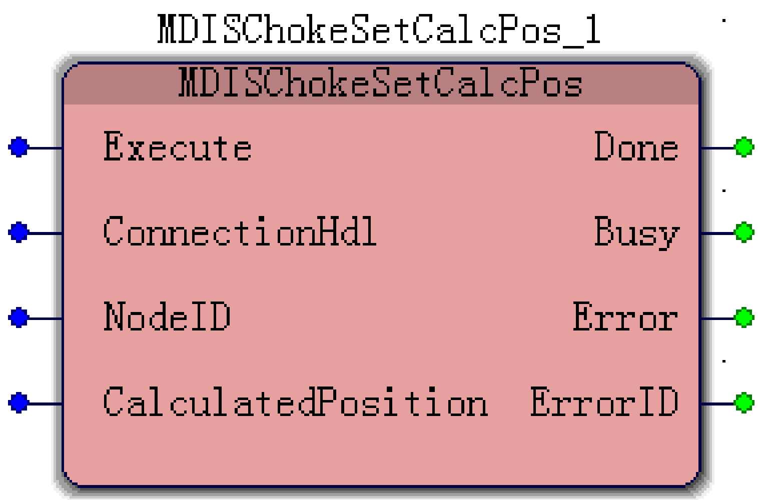
Parameter	Data type	Description
Excute	BOOL	The function block performs its task on rising edge.
ConnectionHdl	DWORD	Connection handle obtained by calling UaConnect
NodeID	STRUCT	NodeID of the Choke object. (See UANodeID for STRUCT description)

Output

Parameter	Data type	Description
Done	BOOL	The function block has completed its task.
Busy	BOOL	The function block is not finished.
Error	BOOL	Signals that an error has occurred. Set to true when ErrorID indicates an error.

Parameter	Data type	Description
ErrorID	DWORD	Error ID if any, returned by the server when Abort method is called.

MDISChokeSetCalcPos



Description

This function block is used to overwrite the CalculatedPosition on a Choke by invoking the SetCalculatedPosition method.

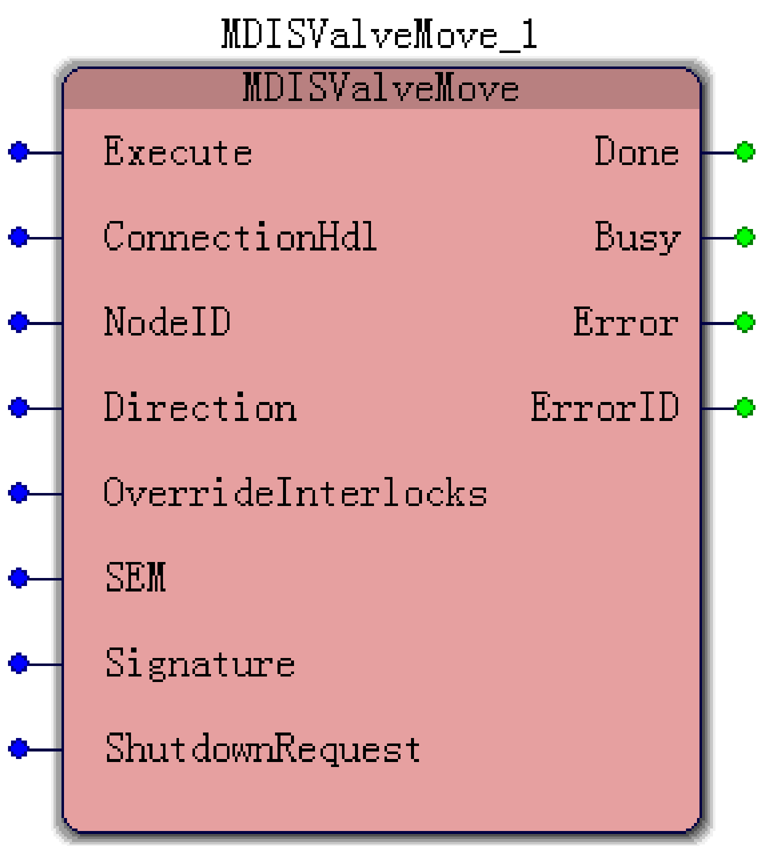
Input

Parameter	Data type	Description
Excute	BOOL	The function block performs its task on rising edge.
ConnectionHdl	DWORD	Connection handle obtained by calling UaConnect
NodeID	STRUCT	NodeID of the Choke object. (See UANodeID for STRUCT description)
CalculatedPosition	REAL	Value to write to ‘CalculatedPosition’ variable.

Output

Parameter	Data type	Description
Done	BOOL	The function block has completed its task.
Busy	BOOL	The function block is not finished.
Error	BOOL	Signals that an error has occurred. Set to true when ErrorID indicates an error.
ErrorID	DWORD	Error ID if any, returned by the server when SetCalculatedPosition method is called.

MDISValveMove



Description

This function block is used to open or close a Valve by invoking the Move method.

Input

Parameter	Data type	Description
Excute	BOOL	The function block performs its task on rising edge.
ConnectionHdl	DWORD	Connection handle obtained by calling UaConnect
NodeID	STRUCT	NodeID of the Valve object. (See UANodeID for STRUCT description)
Direction	DINT	Open or close command. 1 (Close), 2(Open).
OverrideInterlocks	BOOL	If set to 'True', overrides any defeatable interlocks
SEM	DINT	SEM to which command is sent. 1(SEM_A), 2(SEM_B), 4(AUTO)
Signature	BOOL	Boolean indicating if a profile /signature should be generated by this move command request.
ShutdownRequest	BOOL	Boolean indicates that this command is part of a shutdown sequence.

Output

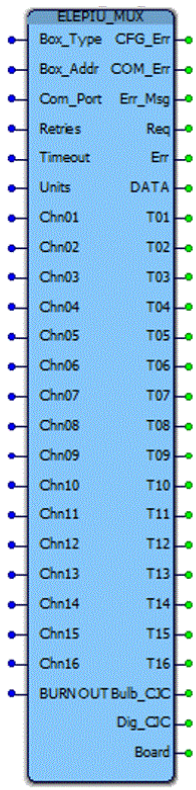
Parameter	Data type	Description
Done	BOOL	The function block has completed its task.
Busy	BOOL	The function block is not finished.
Error	BOOL	Signals that an error has occurred. Set to true when ErrorID indicates an error.
ErrorID	DWORD	Error ID if any, returned by the server when Move method is called.

CHAPTER 24

ELEPIU_MUX

Description

This function block is used to connect to the ELEPIU MUX board and provides the temperatures in a data structure for SCADA or PCDI connections.



Input

Parameter	Data Type	Description
Box_Type	INT	Specifies the type of MUX box: 0 - Not Used 1 - Therocouple 2 - RTD

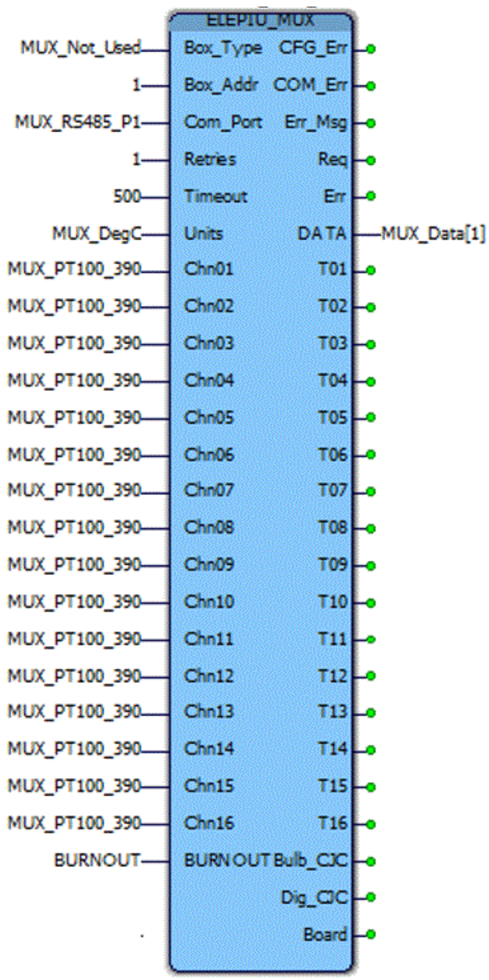
Parameter	Data Type	Description
Box_Addr	INT	Address of the MUX box. Valid values: 1-16
Com_Port	INT	RS485 Communication Port to use. Use 1.
Retries	INT	Number of retries when communication fails.
Timeout	INT	Time out value before a retry. Value is in milliseconds.
Units	INT	Specifies the temperature units: 0 - DegC 1 - DegF 2 - DegK 3 - DegR
Chn01..Chn16	INT	Temperature Element Type: 1 - Type B 2 - Type E 3 - Type J 4 - Type K 5 - Type N 6 - Type R 7 - Type S 8 - Type T 10 - PT390 RTD 11 - PT385 RTD
BURNOUT	REAL	Burnout Value Default is 850.0

Output

Parameter	Data Type	Description
CFG_Err	BOOL	MUX Function Block is not configured correctly.
COM_Err	BOOL	Communication error to MUX board has occurred.
Err_Msg	STRING	Plain text error message
Req	UDINT	Total number of requests made to MUX board
Err	UDINT	Total number of errors
DATA	MUX_ DATA	Data structure used as a rollup for interface to supervisory systems
T01..T16	REAL	Channel Temperatures
Bulb_CJC	REAL	Temperature value of the thermocouple termination board CJC resistor
Dig_CJC	REAL	Onboard CJC
Board	REAL	MUX Board Type read from MUX Board: 2 - Thermocouple 4 - RTD

Example Usage

The following is a completed block. Parameters are defined in global variables. The Modbus Mapping file which maps to MUX_DATA has been provided in the library path. Additionally an example project has been provided as a starting point with 16 MUX boards built.


ELEPIU_MUX
MUX_Not_Used
1
MUX_RS485_P1
1
500
MUX_DegC
MUX_PT100_390
BURNOUT
Box_Type
Box_Addr
Com_Port
Retries
Timeout
Units
Chn01
Chn02
Chn03
Chn04
Chn05
Chn06
Chn07
Chn08
Chn09
Chn10
Chn11
Chn12
Chn13
Chn14
Chn15
Chn16
BURNOUT
CFG_Err
COM_Err
Err_Msg
Req
Err
DATA
MUX_Data[1]
T01
T02
T03
T04
T05
T06
T07
T08
T09
T10
T11
T12
T13
T14
T15
T16
Bulb_CJC
Dig_CJC
Board

CHAPTER

25

DNP3 MASTER

The following DNP3 Master function blocks are available:

Function Blocks	Short Description
DNP3_RD	It is used to read single or multiple DNP3 point(s).
DNP3_WR	It is used to write single or multiple DNP3 point(s).

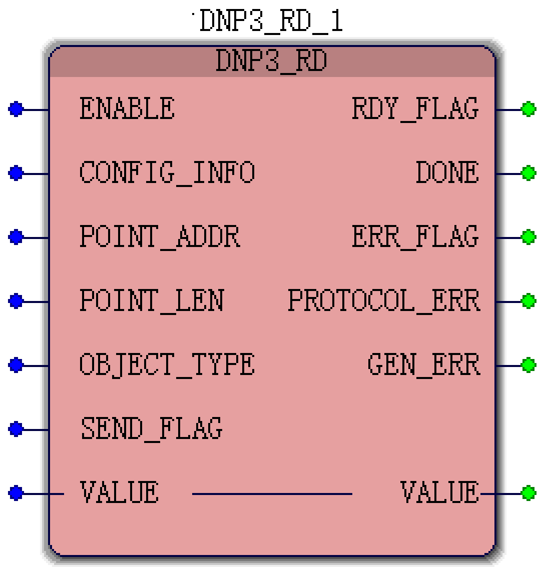
With these function blocks, you can read and write the following types of DNP3 points:

- Single-bit Binary Input
- Double-bit Binary Input
- Binary Output
- Analog Input
- Analog Output
- Counter
- Octet String

Related topics:

- Description of CONFIG_INFO
- Description of Input and Output Data Type
- DNP3 Master Protocol Error Codes

DNP3_RD



Description

It is used to read the following types of DNP3 points from outstation.

- Single-bit Binary Input
- Double-bit Binary Input
- Binary Output
- Analog Input
- Analog Output
- Counter
- Octet String

Input

Parameter	Data type	Description
ENABLE	BOOL	Enable: If TRUE, the FB is enabled and workable.
CONFIG_ INFO	DNP3_ CONFIG_ INFO	This is a structure provided by Honeywell. DNP3 Master related information is included. See Description of CONFIG_ INFO for more information.

Parameter	Data type	Description
POINT_ ADDR	UINT	The start point address you want to read from outstation.
POINT_LEN	UINT	The length of the points you want to read from outstation. The maximum length is 100 points.

OBJECT_ TYPE	USINT	DNP3 data object you want to read from outstation.
		This parameter can be set to the following values:
		kDnp3BinaryInput = 0;
		kDnp3BinaryOutputStatus = 1;
		kDnp3AnalogInput16 = 2;
		kDnp3AnalogInput16_NoFlag = 3;
		kDnp3AnalogOutput16Status = 4;
		kDnp3AnalogInput32 = 5;
		kDnp3AnalogInput32_NoFlag = 6;
		kDnp3AnalogOutput32Status = 7;
		kDnp3AnalogInputFloat = 8;
		kDnp3AnalogOutputFloatStatus = 9;
		kDnp3OctetStringRD = 10;
		kDnp3DoubleBitBinaryInput = 11;
		kDnp3Counter16 = 12;
		kDnp3Counter16_NoFlag = 13;
		kDnp3Counter32 = 14;
		kDnp3Counter32_NoFlag = 15;
		kDnp3FrozenCounter16 = 16;
		kDnp3FrozenCounter16_NoFlag = 17;
		kDnp3FrozenCounter32 = 18;
		kDnp3FrozenCounter32_NoFlag = 19;

Parameter	Data type	Description
SEND_ FLAG	BOOL	If SEND_FLAG is true and RDY_FLAG is true, function blocks will send the request. RDY_FLAG is TRUE means last communication is finished. Before last communication is finished, even if SEND_ FLAG is true the request won't be sent.

Output

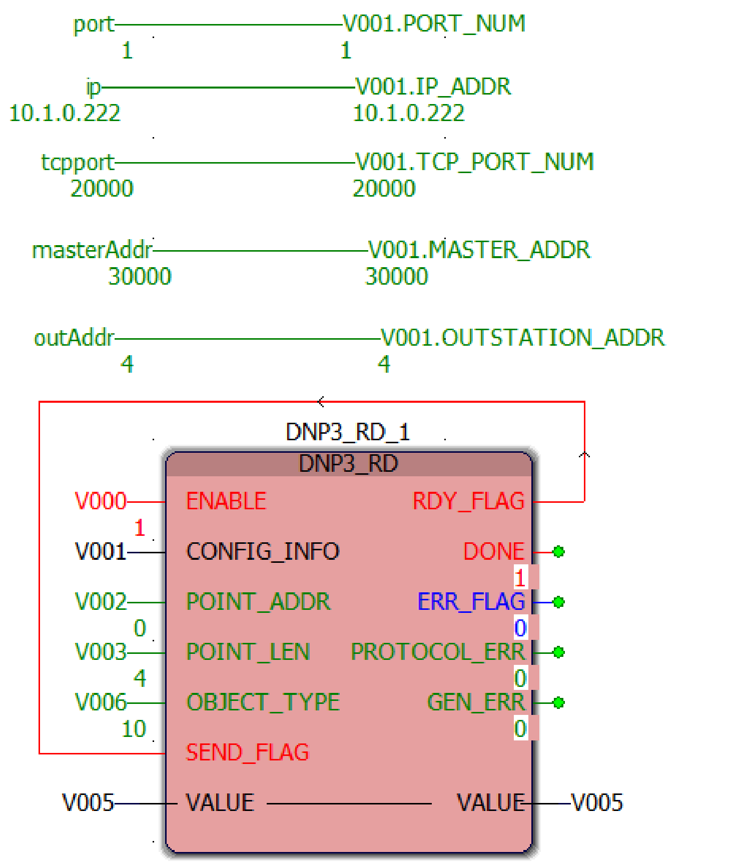
Parameter	Data type	Description
RDY_FLAG	BOOL	True: last communication is finished. FB is ready for the next communication. False: command request is being sent or received.
DONE	BOOL	Indicates that the response is received from responder device.
ERR_FLG	BOOL	Will be set to TRUE if there is either a general error or a protocol error.
PROTOCOL_ ERR	USINT	Error numbers defined by DNP3 Master protocol. See DNP3 Master Protocol Error Codes for more information.
GEN_ERR	USINT	General error code: 0: Communication succeeded. 1: The input parameter is invalid. 2: Response timeout 3: Controller internal time out (IPC timeout). 4: Invalid request

Input and Output

Parameter	Data type	Description
VALUE	DNP3_ DATA	Buffer for the data to be read (for read-output parameter)

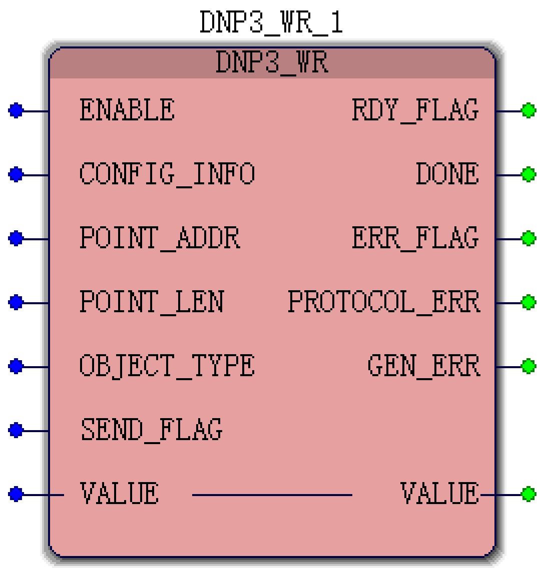
Parameter	Data type	Description
		Buffer size = POINT_LEN*size of (data type) , maximum 512 bytes for this buffer. See the follow size of each data type: Dnp3BinaryInput (0) 1 byte Dnp3BinaryOutputStatus (1) 1 byte Dnp3AnalogInput16 (2) 2 bytes Dnp3AnalogInput16_NoFlag (3) 2 bytes Dnp3AnalogOutput16Status (4) 2 bytes Dnp3AnalogInput32 (5) 4 bytes Dnp3AnalogInput32_NoFlag (6) 4 bytes Dnp3AnalogOutput32Status (7) 4 bytes Dnp3AnalogInputFloat (8) 4 bytes Dnp3AnalogOutputFloatStatus (9) 4 bytes Dnp3OctetString (10) 1 byte Dnp3DoubleBitBinaryInput (11) 1 byte Dnp3Counter16 (12) 2 bytes Dnp3Counter16_NoFlag (13) 2 bytes Dnp3Counter32 (14) 4 bytes Dnp3Counter32_NoFlag (15) 4 bytes Dnp3FrozenCounter16 (16) 2 bytes Dnp3FrozenCounter16_NoFlag (17) 2 bytes Dnp3FrozenCounter32 (18) 4 bytes Dnp3FrozenCounter32_NoFlag (19) 4 bytes

Example



port
1
V001.PORT_NUM
1
ip
10.1.0.222
V001.IP_ADDR
10.1.0.222
tcpport
20000
V001.TCP_PORT_NUM
20000
masterAddr
30000
V001.MASTER_ADDR
30000
outAddr
4
V001.OUTSTATION_ADDR
4
DNP3_RD_1
DNP3_RD
V000
1
ENABLE
RDY_FLAG
V001
CONFIG_INFO
DONE
1
V002
0
POINT_ADDR
ERR_FLAG
0
V003
4
POINT_LEN
PROTOCOL_ERR
0
V006
10
OBJECT_TYPE
GEN_ERR
0
SEND_FLAG
V005
VALUE
VALUE
V005

DNP3_WR



Description

It is used to write the following types of DNP3 points from outstation.

- Single-bit Binary Input
- Double-bit Binary Input
- Binary Output
- Analog Input
- Analog Output
- Counter
- Octet String

Input

Parameter	Data type	Description
ENABLE	BOOL	Enable: If TRUE, the FB is enabled and workable.
CONFIG_INFO	DNP3_CONFIG_INFO	This is a structure provided by Honeywell. DNP3 Master related information is included. See Description of CONFIG_INFO for more information.

Parameter	Data type	Description
POINT_ ADDR	UINT	The start point address you want to write to outstation.
POINT_LEN	UINT	The length of the points you want to write to outstation. The maximum length is 100 points. **NOTE:** The maximum number of objects allowed in a single control request on external outstation side must be considered. If the number on the outstation side is less than 100, the "POINT_LEN" cannot exceed the number of the outstation.

OBJECT_ TYPE	USINT	DNP3 data object you want to write to outstation.
		
		This parameter can be set to the following values:
		
		kDnp3OctetStringWR = 20;
		
		kDnp3CROB_SelOp = 21;
		
		kDnp3CROB_DirOp = 22;
		
		kDnp3CROB_DONA = 23;
		
		kDnp3AnalogOutput16_SelOp = 24;
		
		kDnp3AnalogOutput16_DirOp = 25;
		
		kDnp3AnalogOutput16_DONA = 26;
		
		kDnp3AnalogOutput32_SelOp = 27;
		
		kDnp3AnalogOutput32_DirOp = 28;
		
		kDnp3AnalogOutput32_DONA = 29;
		
		kDnp3AnalogOutputFloat_SelOp = 30;
		
		kDnp3AnalogOutputFloat_DirOp = 31;
		
		kDnp3AnalogOutputFloat_DONA = 32;

SEND_ FLAG	BOOL	If SEND_FLAG is true and RDY_FLAG is true, function blocks will send the request. RDY_FLAG is TRUE means last communication is finished. Before last communication is finished, even if SEND_ FLAG is true the request won’t be sent.

Output

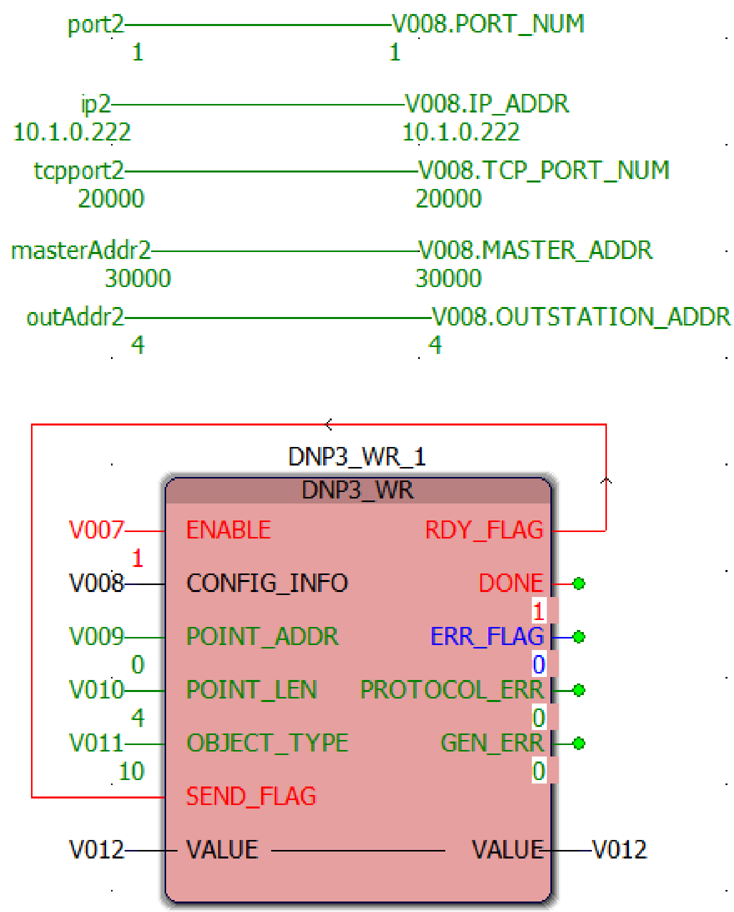
Parameter	Data type	Description
RDY_FLAG	BOOL	True: last communication is finished. FB is ready for the next communication. False: command request is being sent or received.
DONE	BOOL	Indicates that the response is received from responder device.
ERR_FLG	BOOL	Will be set to TRUE if there is either a general error or a protocol error.
PROTOCOL_ERR	USINT	Error numbers defined by DNP3 Master protocol. See DNP3 Master Protocol Error Codes for more information.
GEN_ERR	USINT	General error code: 0: Communication succeeded. 1: The input parameter is invalid. 2: Response timeout 3: Controller internal time out (IPC timeout). 4: Invalid request

Input and Output

Parameter	Data type	Description
VALUE	DNP3_DATA	Buffer for the data to be read (for read-output parameter) Buffer size = POINT_LEN*size of (data type) , maximum 512 bytes for this buffer. See the follow size of each data type: Dnp3OctetStringWR (20) 1 byte Dnp3CROB_SelOp (21) 1 byte

Parameter	Data type	Description
		Dnp3CROB_DirOp (22) 1 byte Dnp3CROB_DONA (23) 1 byte Dnp3AnalogOutput16_SelOp (24) 2 bytes Dnp3AnalogOutput16_DirOp (25) 2 bytes Dnp3AnalogOutput16_DONA (26) 2 bytes Dnp3AnalogOutput32_SelOp (27) 4 bytes Dnp3AnalogOutput32_DirOp (28) 4 bytes Dnp3AnalogOutput32_DONA (29) 4 bytes Dnp3AnalogOutputFloat_SelOp (30) 4 bytes Dnp3AnalogOutputFloat_DirOp (31) 4 bytes Dnp3AnalogOutputFloat_DONA (32) 4 bytes

Example



Description of CONFIG_INFO

The CONFIG_INFO pin defined in the function blocks is to input all the configuration information for the DNP3 Master.

- For Ethernet communication of ControlEdge 2020 controllers, the data structure is defined as:

```
TYPE
          DNP3_CONFIG_INFO:
          STRUCT
                   PORT_NUM:        UDINT;
```

```
                    TCP_PORT_NUM:   UDINT;
                    MASTER_ADDR:    UDINT;
                    OUTSTATION_ADDR:UDINT;
                    IP_ADDR:        STRING;
                END_STRUCT;


                (* Array data type for data read/write *)
                DNP3_DATA: ARRAY[1..512] of BYTE;
        END_TYPE
```

See the following table for the parameter descriptions:

Parameter	Data type	Description
PORT_NUM	UDINT	The physical interface of Ethernet port: 1. Ethernet port 1 2. Ethernet port 2
TCP_PORT_NUM	UDINT	TCP/IP port number of the DNP3 Master device
MASTER_ADDR	UDINT	The address of the DNP3 master
OUTSTATION_ADDR	UDINT	The address of the DNP3 outstation
IP_ADDR	STRING	The IP address of the DNP3 outstation device. Example: ‘192.168.0.100’

Description of Input and Output Data Type

See the following datatype of parameter Value for details:

DNP3_DATA

TYPE (* Array data type for data read/write *)

DNP3_DATA: ARRAY[1..512] of BYTE;

END_TYPE

DNP3 Master Protocol Error Codes

Refer to the following table for DNP3 Master Protocol Error Codes:

Error Code	Item	Description
0	SUCCESS	This indicates the request has completed successfully.
1	INTERMEDIATE	This indicates a response was received but the requested command is not yet complete. This could mean the response is part of a multi-fragment response and did not have the FINAL bit set. Or this could be a request such as a select operate that requires multiple requests and responses.
2	FAILURE	This indicates that the transmission of the request failed.
3	MISMATCH	The response to a select or an execute did not echo the request.
4	STATUSCODE	The response to a select or an execute echoed the request, except the status code was different indicating a failure.
5	IIN	The response to the request had IIN bits set indicating the command failed.
6	TIMEOUT	This indicates that the request has timed out. This could either be an incremental timeout indicating we received no link layer frame from the device in the specified time, or an application response timeout indicating this particular request did not complete in the specified time.
7	CANCELED	This indicates either that the user asked that the request be canceled by calling dnpchnl_cancel Fragment or that a second duplicate request has been made and therefore this first one is canceled.

CHAPTER

26 ENERGY CONTROL

The following Energy Control function blocks are available:

Function Block	Description
PeakShaver	The PeakShaver function block controls the charging and discharging of batteries which is determined by: ■ A configurable algorithm based controller depending on the Shave level and Battery State of charge, load and generation. ■ Secondary controller which allows charging/ discharging to be commanded depending on logic running externally.
VARControl	The VARControl (Volt Ampere Reactive power control) function block monitors the grid voltage, active power and reactive power to or from the grid. This function block configures the PCS to control reactive power flow to support reactive loads.
FrequencyRegulation	The FrequencyRegulation function block regulates the grid frequency when there is a positive or negative frequency drift. This is done by using battery as load (and thus draw power from the grid and charge the battery) or generator of power (discharge the battery) respectively.
RampRateControl	The RampRateControl function block monitors and controls the grid power ramp rate and maintains the grid power ramp rate within specified limits using battery storage. This function block calculates current grid power ramp rate based on sampling period and compares it with the grid code ramp rate values. If grid power ramping up/ down is faster than the specified ramp rate limits then a reference power is calculated to compensate the increase/ decrease in grid power ramp rate thereby maintaining ramp rate at grid connection point in compliance to grid code.
CapacityFirming	The CapacityFirming function block takes renewable power generation as an input and generates an active power reference for Power Conversion System (PCS). This compensates for changes in power output due to the

Function Block	Description
	intermittency in power generation.
CapacitySmoothing	The CapacitySmoothing function block absorbs short term or high frequency variations in the output power delivered to the grid. The battery is either charged/discharged depending on a power reference computed by the smoothing algorithm.
RampRateLimiter	The RampRateLimiter function block ramps output value when there is change in input value. The block provides an option to configure separate rates for up and down ramping. This function block applies Ramp Rate limiting on the output provided by the Summer block in a controlled scheme in which: 1. More than one function block of the Energy Control Library can be used. OR 2. Individual function block of the Energy Control Library can be used.
Summer	The Summer function block calculates an output value from four inputs that are summed, scaled and biased. Typically, the Summer function block takes inputs from different function blocks provided by the Energy Control Library and applies a net effect on the output determined by different algorithms.
ECAutoman	The ECAutoman function block transfers the reference power to the PCS. It can transfer both active and reactive power. This function block is typically used as the last element in the Energy Control scheme and its output is provided which can be connected to a PCS. ECAutoman function block also allows choosing between automatic or manual control mode of operation.
PowerShare	The PowerShare function block shares an output power reference up to four different outputs. The sharing mode can be configured on the function block which allows equal/ unequal sharing. This function block will be applied when there are more than one Power Conversion System (PCS) or Battery Management System (BMS) being controlled by the energy controller.
Dynamic containment	The Dynamic Containment is a response service that controls frequency within the statutory range for a sudden demand or generation loss. It is a post-fault frequency management service and delivers a quick response proportional to the

Function Block	Description
	frequency deviation. This block provides modes for a low-frequency response, a high-frequency response or both. This service is also necessary in an electrical grid system with low inertia that experience large loss.
Firm Frequency Response	The Firm Frequency Response (FFR) is a service that utility requires in the response to a change in system frequency. This change in active power could be either from its initial state or a baseline. The FFR can provide both dynamic and non-dynamic response to the changes in system frequency. The key difference between Non-dynamic and Dynamic is that, when Non-dynamic response is triggered, it sustains at same level irrespective of any further frequency change until it meets sustain time.
Battery Dispatch Scheduler	The Battery Dispatch Scheduler is used to schedule charge and discharge power of the battery in a given day and duration. The battery dispatch scheduler can be configured with fixed number of recurring or non-recurring schedules(i.e 10 schedules) based on mode and setpoint. The output power reference is calculated based on the time interval, mode and setpoint. The power and energy units are in engineering units.

PeakShaver

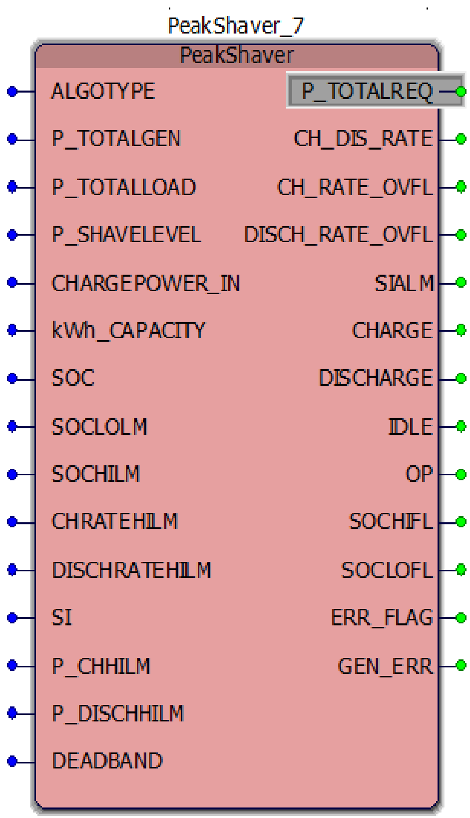
Peak shaving is a process of leveling out peak power demand in electricity used by industrial and commercial power consumers.

The PeakShaver function block allows to control charging and discharging of batteries which is determined by:

- A configurable algorithm based controller depending on a Shave level and Battery State of charge.
- Secondary controller which allows charging/ discharging to be commanded depending on logic running externally.

Using the peak shaving algorithms, power consumption could be shifted to battery storage for a period of time to avoid a spike in consumption over a defined threshold (referred to as Shave Level in this function block).

Rule based peak shaving algorithm (ALGOTYPE 2) is meant for shaving facility peaks in a Behind the meter Commercial and Industrial facility or at Front of the meter Renewable or Hybrid Power developer facility. Shave level is a configurable parameter that allows user to configure the shave power level above which power drawn from the grid in case of a behind the meter facility is shaved using Battery Energy Storage. In case of a front of the meter facility, it is the power exported to the grid that is sourced from Battery Energy Storage System to make up for shortfall in power generation within the power developer facility. Shaving Grid co-incident peaks require secondary peak shaving mode to be invoked and let the grid peaks be predicted by software running on an external controller.



Input

Parameter Name	Description	Data Type
ALGOTYPE	0 – Pass through 1 – Conditional Pass Through	USINT

Parameter Name	Description	Data Type
	2 – Load Management The default value is 0.	
P_TOTALGEN	Total Generation Input. The default value is 0.0.	LREAL
P_TOTALLOAD	Total Load Input. The default value is 0.0.	LREAL
P_SHAVELEVEL	Shave Level Input. The default value is 0.0.	LREAL
CHARGEPOWER_IN	Power to Charge (-ve) /Discharge (+ve) Input. The default value is 0.	LREAL
kWh_CAPACITY	Battery capacity in kWh. The default value is 0.0.	REAL
SOC	State Of charge expressed as %. The default value is 0.0.	REAL
SOCLOLM	Minimum State of charge %. The default value is 0.	REAL
SOCHILM	Maximum State of charge %. The default value is 100.	REAL
CHRATEHILM	Maximum Charge Rate. The default value is 1.0.	REAL
DISCHRATEHILM	Maximum Discharge Rate. The default value is 1.	REAL
SI	Safety Interlock . The default value is OFF.	BOOL
P_CHHILM	Max charging power. Must be < 0 as Charge Power is provided as a negative value. The default value is 0.0	LREAL
P_DISCHHILM	Max discharging power. Must be > 0 as Discharge Power is provided as a positive value. The default value is 0.0	LREAL
DEADBAND	User configurable Dead band range to be applied to the ALGOTYPE 2. The default value is 0.0.	REAL

Output

Parameter Name	Description	Data Type
P_TOTALREQ	P_TOTALREQ = P_TOTALLOAD - P_TOTALGEN - P_ SHAVELEVEL Used to determine charge or discharge.	LREAL
CH_DIS_RATE	Charge/Discharge Rate. The default value is 0.	REAL

Parameter Name	Description	Data Type
CH_RATE_OVFL	Charge Rate Overflow Flag. The default value is False.	BOOL
DISCH_RATE_OVFL	DisCharge Rate Overflow Flag. The default value is FALSE.	BOOL
SIALM	Safety Interlock Flag. The default value is False.	BOOL
CHARGE	Battery Charging. The default value is False.	BOOL
DISCHARGE	Battery Discharging. The default value is False.	BOOL
IDLE	Batery Idle. The default value is False.	BOOL
OP	Power to Charge/Discharge Output. The default value is 0.	LREAL
ERR_FLAG	Error Flag. The default value is False.	BOOL
GEN_ERR	Error Code. The default value is 0.	USINT
SOCHIFL	SOC High limit reached flag. The default value is False.	BOOL
SOCLOFL	SOC Low limit reached flag. The default value is False.	BOOL

GEN_ERR	Description
0	No Error
1	Invalid ALGOTYPE
2	Invalid SOC Limit
3	Invalid Charge Discharge Limit
4	Invalid Max Charge Rate
5	Invalid Max Discharge Rate
6	Invalid Battery Capacity
7	Invalid Dead band

Detailed Description

The PeakShaver function block provides 3 modes of operation configured by ALGOTYPE parameter.

When ALGOTYPE is 1 (Conditional Pass through) or 0 (Pass through):

The block does not perform any automatic computations and instead accepts inputs from another primary controller. The primary controller could be:

- SCADA based controller
- Another PLC
- An optimization algorithm running on the same PLC

When ALGOTYPE is 0 (Pass through):

The function block accepts CHARGEPOWER_IN as the input and provides the same output in OP as is without applying any other constraints except SI. When SI is asserted, the output is changed to 0.

When ALGOTYPE is 1 (Conditional Pass through):

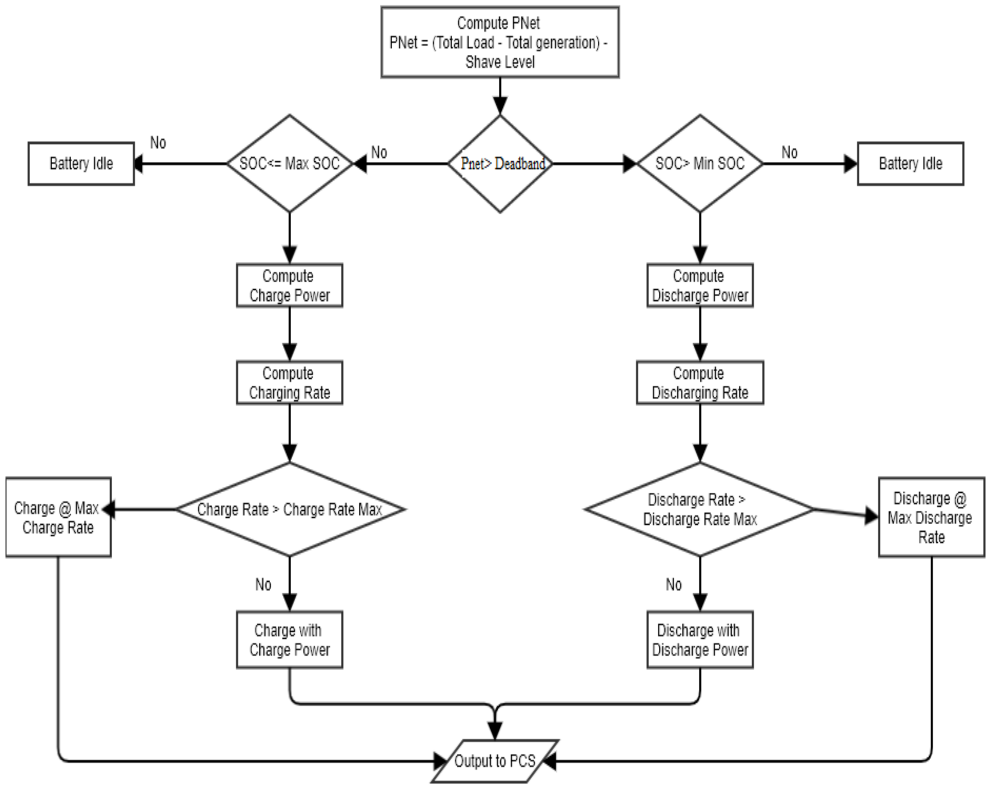
The function block accepts CHARGEPOWER_IN as the input and provides the same output in OP after applying the following constraints:

- SOC of the battery is read as an input and the OP is supplied only when SOC is in limits defined by SOCLOLM and SOCHILM.
- The CHARGEPOWER_IN is passed to the output only when the output power is within the range defined by P_CHHILM and P_ DISCHHILM.
- SI is OFF. The Safety Interlock could be asserted by other logic/ function blocks and indicate to the PeakShaver Block that the output must be driven to 0.

When ALGOTYPE is 2 (Load Management):

The function block considers Total Generation, Total Load and Shave Levels as the inputs and produces a charging or discharging power based on the following flowchart.

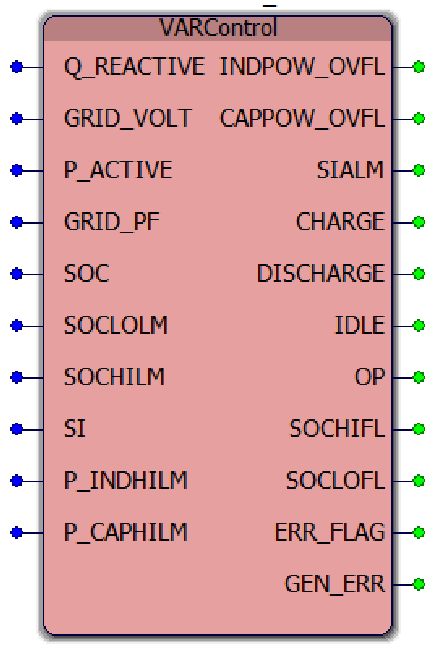
Essentially, battery will be charged when the loading is lower than a configured threshold level and discharged when loading exceeds the defined threshold level.



NOTE: The above flowchart represents one iteration of the algorithm.

VARControl

The Volt Ampere Reactive Control (VARControl) function block monitors the grid voltage, active power and reactive power to or from grid. This control function block configures the PCS to control reactive power flow to support reactive loads



Input

Parameter Name	Description	Data Type
SI	Safety Interlock. The default value is False.	BOOL
GRID_VOLT	Grid Voltage. This is an input from PCS. The default value is 0.0.	LREAL
P_ACTIVE	Active Power to/ from grid. Grid Voltage. This is an input from PCS. The default value is 0.0.	LREAL
Q_REACTIVE	Reactive Power to/ from Grid. The default value is 0.0.	LREAL
GRID_PF	Power Factor. The default value is 0.0.	LREAL
SOC	State of Charge expressed as %. This is an input from Battery Management System. The default value is 0.0.	REAL
SOCLOLM	Minimum battery SoC threshold limit in %. The default value is 0.0.	REAL
SOCHILM	Maximum battery SoC threshold limit in %. The default value is 100.0	REAL

Parameter Name	Description	Data Type
P_INDHILM	Inductive Power Upper Limit; Negative Value since inductive power is absorbed into the Battery. The default value is -1.0. This parameter should be configured with a non zero negative value for normal operation of the function block.	LREAL
P_CAPHILM	Capacitive Power Upper Limit; Positive value since capacitive power is dispatched from the Battery. The default value is 1.0. This parameter should be configured with a non zero positive value for normal operation of the function block.	LREAL

Output

Parameter Name	Description	Data Type
OP	Output Power Reference. The default value is 0.0.	LREAL
INDPOW_OVFL	Inductive Power Overflow. The default value is False.	BOOL
CAPPOW_OVFL	Capacitive Power Overflow. The default value is False.	BOOL
SIALM	Safety Interlock Flag. The default value is False.	BOOL
CHARGE	Battery Charging State. The default value is False.	BOOL
DISCHARGE	Battery Discharging State. The default value is False.	BOOL
IDLE	Battery Idle State. The default value is False.	BOOL
SOCHIFL	SOC High Limit Reached Flag. The default value is False.	BOOL
SOCLOFL	SOC Low Limit Reached Flag. The default value is False.	BOOL
ERR_FLAG	Error Flag. The default value is False. The default value is False.	BOOL

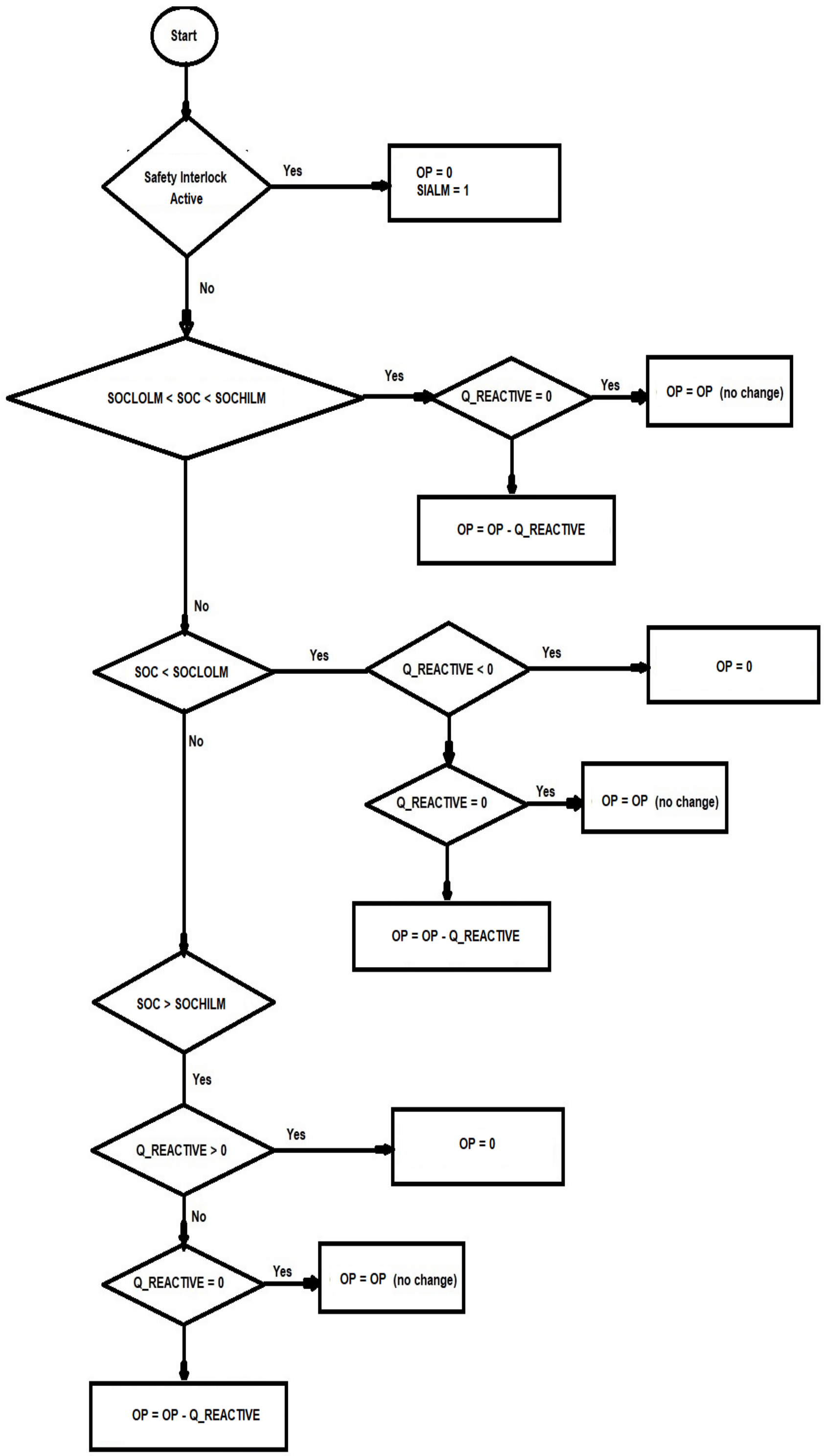
Parameter Name	Description	Data Type
GEN_ERR	Error Code: 0 : No Error 1 : SOC Low Limit Reached 2 : SOC High Limit Reached 3 : Grid Voltage Out of Range 4 : Grid Active Power Out of Range 5 : Grid Reactive Power Out of Range 6 : Grid Power Factor Out of Range 100 : Max Warnings 101 : Invalid SOC Limit The default value is 0.	USINT

Detailed description

Reactive Power function block generates Reactive Power reference (OP) for Power Conversion system (PCS), configuration for normal reactive power reference or grid support reactive power reference, ac voltage controller gain, and ac voltage controller integration time.

Interlocks for Volt-Ampere Reactive Power /Power Factor Control function block

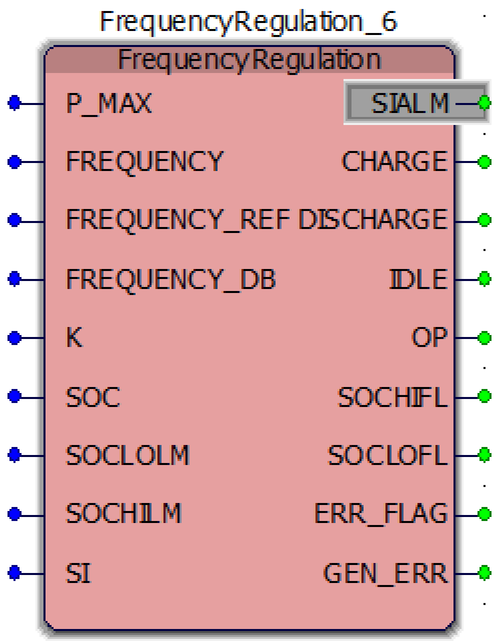
The function block logic configures reactive power reference for PCS and is complimentary to PCS curves for reactive power such as Q(U) curve, Q(P) curve or Cos ϕ(P) curve for reactive power control. An active power reserve is also configured. The algorithm limits energy dispatch by discharging of battery to SOCLOLM which is a configurable lower limit for SOC and charging of battery to SOCHILM , which is a configurable upper limit for SOC. Reactive power that is either input or output is also limited to the maximum reactive power. These limits can be made configurable from Energy control as well.


Start
Safety Interlock Active
Yes
OP = 0
SIALM = 1
No
SOCLOLM < SOC < SOCHILM
Yes
Q_REACTIVE = 0
Yes
OP = OP (no change)
OP = OP - Q_REACTIVE
No
SOC < SOCLOLM
Yes
Q_REACTIVE < 0
Yes
OP = 0
No
Q_REACTIVE = 0
Yes
OP = OP (no change)
OP = OP - Q_REACTIVE
SOC > SOCHILM
Yes
Q_REACTIVE > 0
Yes
OP = 0
No
Q_REACTIVE = 0
Yes
OP = OP (no change)
OP = OP - Q_REACTIVE

NOTE: The above flowchart represents one iteration of the algorithm.

FrequencyRegulation

The FrequencyRegulation function block regulates the grid frequency when there is a positive or negative frequency drift. This is done by using battery as load (and thus draw power from the grid and charge the battery) or generator of power (discharge the battery) respectively.



Input

Parameter Name	Description	Data Type
P_MAX	User configurable. Maximum Power at any point of time that a PCS can deal with . The default value is 0.0.	REAL
FREQUENCY	Grid Frequency. Input from PCS. The default value is 0.0	REAL
FREQUENCY_REF	Nominal Frequency. User configurable. The default value is 0.0.	USINT

Parameter Name	Description	Data Type
SOC	Battery State of Charge. The default value is 0.0	REAL
SOCLOLM	Minimum battery SoC threshold limit in %. The default value is 0.0.	REAL
SOCHILM	Maximum battery SoC threshold limit in %The default value is 100.0.	REAL
SI	External Input to inhibit Algorithm Execution. The default value is 0.0.	BOOLEAN
K	Ramping Constant. User configurable. The default value is 0.45.	REAL
FREQUENCY_DB	Frequency Dead band to be applied on Frequency Reference. This is user configurable. The default value is 0.05.	REAL

NOTE:

- As per standards, Frequency Reference can be either 50Hz or 60Hz.
- The Frequency Deadband and Ramping constant must be always be associated together. For Example:

Deadband (Hz)	Ramping constant (k)
50 ± 0.05	0.45
50 ± 0.015	0.485

- The frequency Deadband and Ramping constant are user configurable. The user must calculate the appropriate value for Ramping constant based on the chosen Deadband.

  The Ramping constant is df/dp, or rate of change of frequency with power.

  This can be calculated using (f-max – f- Deadband) / Normalized Power.

  Normalized Power = Instantaneous Power / Pmax

Output

Parameter Name	Description	Data Type
OP	Output Power Reference. The default value is 0.0.	LREAL
ERR_FLAG	Error Flag – Active when the block detects any errors. The default value is False.	BOOL
GEN_ERR	Error Code. The default value is 0.	USINT
CHARGE	Flag to Indicate negative OP (Pref). Battery is instructed to charge. The default value is False.	BOOL
DISCHARGE	Flag to Indicate positive OP (Pref). Battery is instructed to Discharge. The default value is False.	BOOL
IDLE	Flag to Indicate Battery is Idle. SOC Moderation can be applied by another block only if this is ON. The default value is False.	BOOL
SIALM	SAFETY Interlock Alarm. The default value is False.	BOOL
SOCHIFL	Flag to indicate SOC >=SOCHILM. The default value is False.	BOOL
SOCLOFL	Flag to indicate SOC <=SOCLOLM. The default value is False.	BOOL

GEN_ERR	Description	Details
0	kFRNoError	No errors
1	kFRInvalidSOC	To flag when SOC < 0 or SOC >100
2	kFRInvalidSOCLimit	To flag when SOC Limits are < 0 or >100
3	kFRInvalidk	To flag if K is set to 0.0
4	kFRInvalidDB.	To flag if Negative Deadband is configured

Detailed description

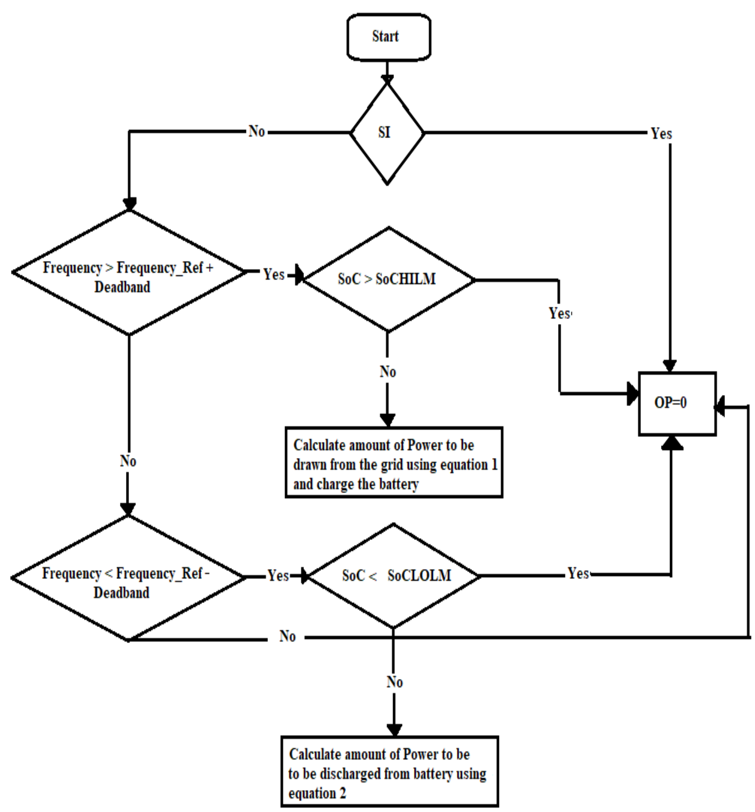
In the FrequencyRegulation function block, the changes in supply and demand for electricity can have a major effect on the frequency of the grid. For instance, if there is more demand for electricity than there is supply, then frequency will fall (Negative Frequency Drift), or if there is more supply, frequency will rise (Positive Frequency Drift).

The FrequencyRegulation algorithm limits the discharge from battery to SOCLOLM, which is a configurable lower limit for SOC and charge from battery to SOCHILM, which is a configurable upper limit for SOC.

The FrequencyRegulation function block uses the following equations:

Equation 1: `OP = Minimum (Pmax * ((Frequency - (Frequency_Ref + Frequency_DB))/k), P Max)`

Equation 2: `OP = Maximum (Pmax * ((Frequency- (Frequency_Ref - Frequency_DB))/k), -Pmax)`



> **NOTE:** The above flowchart represents one iteration of the algorithm.

The margin for error is very small. This block functions around +/- 0.05 or 0.015 of the nominal frequency.

The FrequencyRegulation function block has the following scenarios:

1. **Positive/ Negative Frequency Drift**

   Positive Frequency Drift

   If the measured grid frequency goes beyond the Nominal Frequency + 0.05, it implies that the supply has exceeded the demand and thus additional power needs to be drawn from the grid and must be used to charge the battery. If the SOC is within the operational limits, the FrequencyRegulation function block will calculate the amount of power to be drawn based on the formula described earlier, and passes this information to PCS. Charging Flag is set. If the battery SOC is above the upper threshold limit, battery cannot support Frequency Regulation as it cannot be charged further. Thus, OP will be set to 0.

   Negative Frequency Drift

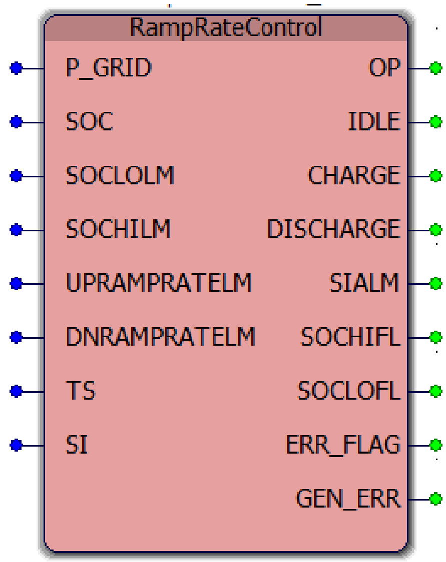
   If the measured grid frequency goes below the Nominal Frequency - 0.05, it implies that the demand has exceeded the supply and thus the battery needs to discharge power. If the SOC is within the operational limits, the FrequencyRegulation function block will calculate the amount of power to be drawn based on the formula described earlier, and passes this information to PCS. Discharge Flag is set. If the battery SOC is below the lower threshold limit, battery cannot support Frequency regulation as it cannot be discharged further. Thus, OP will be set to 0.

2. **Safety Interlock**

   Safety Interlock is considered as an input to the algorithm. This could be asserted by other logic/ function blocks. When this Flag is ON, it implies that the Algorithm should not take part in frequency regulation and drive an OP of 0.

RampRateControl

The RampRateControl function block monitors and controls the grid power ramp rate and maintains the grid power ramp rate within specified limits using battery storage. This function block calculates current grid power ramp rate based on sampling period and compares it with the grid code ramp rate values. If grid power ramping up/ down is faster than the specified ramp rate limits then a reference power is calculated to compensate the increase/ decrease in grid power ramp rate thereby maintaining ramp rate at grid connection point in compliance to grid code



Input

Parameter Name	Description	Data Type
P_GRID	Measured total grid power at grid connection point. The default value is 0.0.	LREAL
SOC	Measured battery SOC in %. The default value is 0.0.	REAL
SOCHILM	Maximum battery SOC threshold limit in %. The default value is 100.0.	REAL
SOCLOLM	Minimum battery SOC threshold limit in % . The default value is 0.0.	REAL
UPRAMPRATELM	Up ramp rate limit at grid connection point per minute as specified by grid code. The default value is 0.0.	REAL
DNRAMPRATELM	Down ramp rate limit at grid connection point per minute as specified by grid code. The default value is 0.0.	REAL

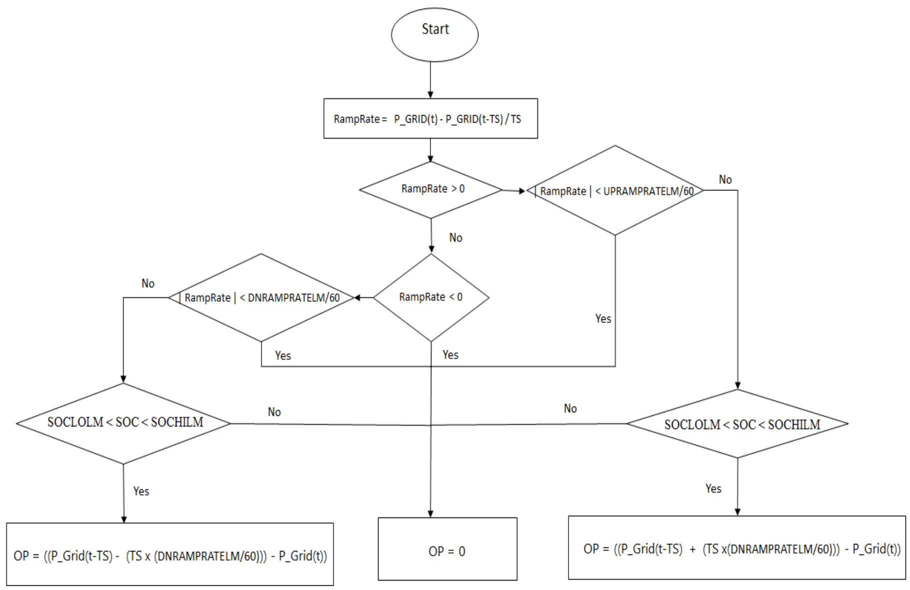
Parameter Name	Description	Data Type
TS	Grid power sampling time in seconds. The default value is 1.0	REAL
SI	Safety interlock. The default value is OFF.	BOOL

Output

Parameter Name	Description	Data Type
OP	Reactive Power Reference. The default value is 0.0.	LREAL
IDLE	Indicates idle OP power reference. The default value is ON.	BOOL
CHARGE	Indicates charge OP power reference. Indicates idle OP power reference. The default value is OFF.	BOOL
DISCHARGE	Indicates discharge OP power reference. The default value is OFF.	BOOL
SIALM	Safety Interlock Flag. The default value is OFF.	BOOL
ERR_FLAG	Error Flag. The default value is OFF.	BOOL
GEN_ERR	Error Code Valid Error code for RRC blocks are 0 : No Error 1 : Sample time less than the cycle execution time 2 : Invalid Sampling time 3 : Invalid SOC Limits 4 : Invalid Up rate limit 5 : Invalid Down rate limit The default value is 0.	USINT
SOCHIFL	Flag to indicate SOC >= SOCHILM. The default value is OFF.	BOOL

Parameter Name	Description	Data Type
SOCLOFL	Flag to indicate SOC <= SOCLOLM. The default value is OFF.	BOOL

Detailed Description



NOTE: The above flowchart represents one iteration of the algorithm.

RampRateControl function block calculates the current grid power (P_GRID) per second based on sampling period and compares it with the grid code ramp rate values (per minute).

In case the grid power ramping up/down does not exceed the specified up/down rate, then the power reference is set to zero.

In case the grid power ramping up/down is faster than the specified ramp rate limits , then the RampRateControl function block calculates the required reference power to compensate the increase/ decrease in grid power ramp rate, thereby maintaining ramp rate at grid connection point in compliance to grid code.

In case safety interlock (SI) is active then reference power is set to zero.

The RampRateControl function block is used to maintain grid power ramp-rate in compliance with grid code ramp-rate limits.

Rate of change of power at grid connection point must be controlled and limited to a maximum defined by grid code. Rapid changes in power imported or exported at grid connection point can be due to a new generator that is turned on, such as a PV inverter or any other generator. It could also be due to a generator that is down or turned off. Energy Control coordinated by RR control algorithms are often applied for mitigating these power fluctuations to the grid. These algorithms generate a power reference to the PCS that opposes the PV fluctuations. This reduces the PV fluctuations to an acceptable value and helps in maintaining the ramp rate at grid connection point in compliance to grid code.

Two main scenarios under which RampRateControl function block may act to limit the grid power rate are:

- **PV Inverter turned on during the day or if there is a large change in PV inverter output**

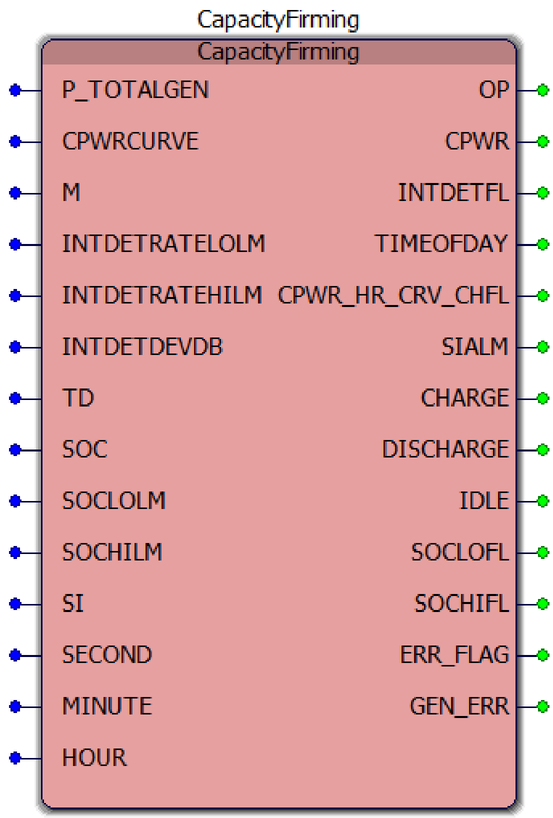
  In case PV inverter is turned on and PV ramps up too fast, then the grid power at grid connection point may exceed the specified RR limit. Thus, to maintain the grid RR limit as per grid code, the grid RR must be calculated every second (TS=1) and compared with the grid code RR limit. In case the current upward RR exceeds the specified upward RR limit then a power reference is calculated to compensate the rapid increase in PV power and charge battery power. This maintains the RR limit at grid connection point as per the grid code.

- **PV Inverter turned off or PV inverter output ramp down too fast during evening time**

  In case PV ramps down too fast, then the grid power at grid connection point may exceed the specified RR limit. Thus, to maintain the grid RR limit as per grid code, grid RR must be calculated every second (TS=1) and compared with the grid code RR limit. In case current downward RR exceeds the specified downward RR limit, then a power reference is calculated to compensate rapid decrease in PV power and discharge battery power. This maintain the RR limit at grid connection point as per the grid code.

CapacityFirming

The CapacityFirming function block takes renewable power generation as an input and generates an active power reference for Power Conversion System (PCS). This compensates for changes in power output due to the intermittency in power generation.



Input

Parameter Name	Description	Data Type
SOC	State Of charge expressed as %. The default value is 0.0.	REAL
SOCLOLM	Minimum battery SOC threshold limit in %. The default value is 0.0	REAL
SOCHILM	Maximum battery SOC threshold limit in %. The default value is 100.0.	REAL
P_TOTALGEN	Total renewable power generation in Engineering Units (EU). The default value is 0.0.	LREAL
CPWRCURVE	Array of characteristic power curve in Engineering Units (EU) for a day with 25 samples. One segment in the curve for each hour of the day. The values between two successive samples is extrapolated based on a	ANY

Parameter Name	Description	Data Type
	straight line connecting the two points on the power-time graph specified by the curve. Each sample in the curve must be of type 'REAL' (4 byte real value).	
M	Multiplication factor applied to values obtained from "Characteristic power curve" to obtain “Optimum power reference” curve. It can be used to control the degree/ extent of firming obtained by the algorithm. The default value is 1.0.	REAL
INTDETRATEHILM	Upper limit of the rate of change of difference between instantaneous power and calculated optimum power reference. The units are in EU/ minute. The default value is 1.0.	REAL
INTDETRATELOLM	Lower limit of the rate of change of difference between instantaneous power and calculated optimum power reference. The units are in EU/ minute. The default value is 0.0.	REAL
INTDETDEVDB	Dead band value used for intermittency detection. The units are in EU. The default value is 1.0.	REAL
TD	Intermittency off-delay timer in minutes. The default value is 5.0.	REAL
HOUR	Time of current day maintained as the number of current hour. The default value is 0.	USINT
MINUTE	Minute in the current hour maintained as a number. The default value is 0.	USINT
SECOND	Second in the current minute maintained as a number. The default value is 0.	USINT
SI	Safety Interlock. The default value is False.	BOOL

Output

Parameter Name	Description	Data Type
OP	Output power reference to charge or discharge	LREAL

Parameter Name	Description	Data Type
	the battery. When intermittency is present, it is the deviation between the instantaneous renewable power output and the value determined from the characteristic power curve. The Units are in Engineering units (EU). The default value is 0.0.	
CPWR	The instantaneous value of power as determined from the characteristic power curve. The value is extrapolated from the “Characteristic power curve” between two consecutive samples using linear extrapolation. The Units are in Engineering units (EU). The default value is 0.0.	LREAL
INTDETFL	Flag used to track the intermittency state as detected by the block. The default value is False.	BOOL
TIMEOFDAY	Time of current day maintained as the number of hours. The default value is 0.0.	REAL
CPWR_HR_CRV_CHFL	The parameter is used to track if the "Characteristic power curve" samples for the current hour have changed. The value of the parameter remains true until the current hour elapses. The default value is False.	BOOL
SIALM	Safety Interlock condition status flag. The default value is False.	BOOL
CHARGE	The parameter when true indicates that the battery is being charged. The default value is False.	BOOL
DISCHARGE	The parameter when true indicates that the battery is being discharged. The default value is False.	BOOL
IDLE	The parameter when true indicates that the battery is idle. The default value is True.	BOOL
SOCLOFL	The parameter when true indicates that the block is in a wind-up condition. The block was supposed to perform Capacity firming action but is unable to due to the battery SOC in a wind-up state because of being at the low limit. The default value is false.	BOOL

Parameter Name	Description	Data Type
SOCHIFL	The parameter when true indicates that the block is in a wind-up condition. The block was supposed to perform Capacity firming action but is unable to due to the battery SOC in a wind-up state because of being at the high limit. The default value is False.	BOOL
ERR_FLAG	The parameter when true indicates that the block is in an erroneous state. The default value is False.	BOOL
GEN_ERR	The parameter indicates the error condition (non zero value) of the block when the block is in an erroneous state and cannot continue normal operation. The valid set of values are: 0 – No error 1 - The Intermittency detection rate related configuration parameters (INTDETRATELOLM, INTDETRATEHILM) have invalid values. 2 - The SOC related configuration parameters (SOCLOLM, SOCHILM) have invalid values. 3 - Number of samples defining the "Characteristic power curve" are more than 25. 4 - Number of samples defining the "Characteristic power curve" are less than 25. The default value is 0-no error.	USINT

Detailed Description

CapacityFirming function block primarily addresses the problem of intermittent power generation from renewable energy source. The active power reference generated by the function block is an input to the PCS that compensates for the intermittency in generated power (For example, in the case of solar power generation, during the long duration cloud covers or during large power swings at noon when the solar output power is at its peak).

The block receives “Characteristic power curve” as an input which defines the power curve profile for a complete day. The input is received as a set of twenty-five samples that defines the characteristic power as a function on a particular time of the day. Two successive samples define a characteristic power curve profile for a given hour of day. The values between two successive samples is extrapolated based on a straight line connecting the two points on the power-time graph specified by the curve.

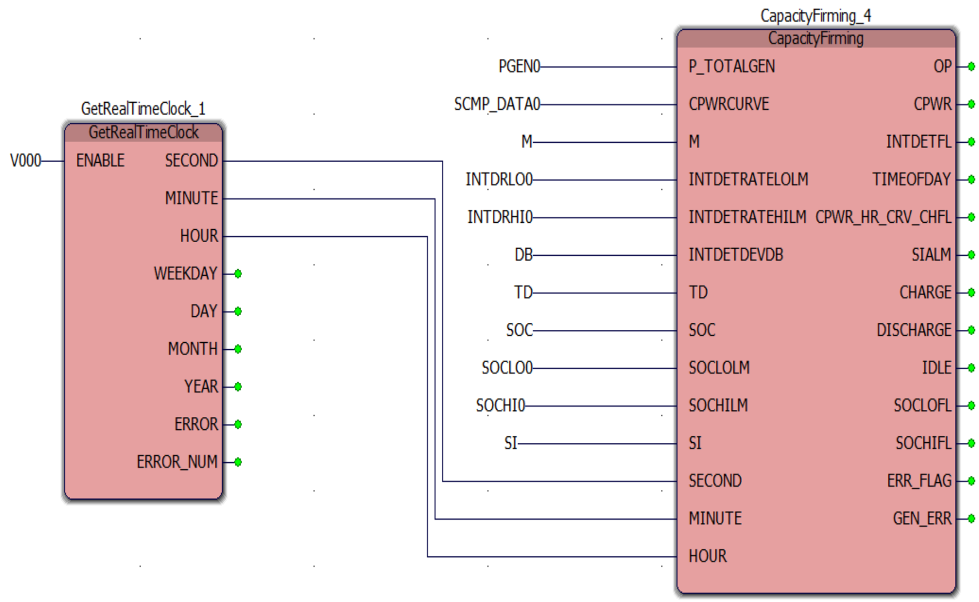
During the periods of intermittent power generation, the output is maintained at an optimum level as determined by the “Characteristic power curve”.

The parameter “CPWRCURVE” must be connected to an arrayed variable with twenty-five REAL values. The block will produce incorrect results if it is connected to an array data type other than REAL.

The following data type can be used to instantiate an arrayed variable to be used with “CPWRCURVE”:

```
 TYPE

(* Array of reals *)

 SCMP_DATA: ARRAY[0..24] of REAL;

END_TYPE
```

The block calculates CPWR which is the instantaneous value of power as determined from the characteristic power curve. The value is extrapolated from the “Characteristic power curve” between two consecutive samples using linear extrapolation.



The block tracks changes to the variable connected to “CPWRCURVE” for changes to the values used to extrapolate the values for the current hour. The block immediately acts on the changes to the values for the current hour and uses the updated values to calculate the value of CPWR. When the block detects changes to “CPWRCURVE” values for the current hour it also sets the parameter “CPWR_HR_CRV_CHFL” to true which remains true until the current hour elapses.

This essentially means that the block functions on the current value of the curve defined by the parameter “CPWR_HR_CRV_CHFL”. The value of parameter “CPWR_HR_CRV_CHFL” can be changed anytime there is a need for the block to operate on updated values of characteristic power curve. The Safety Interlock can be asserted if there is a need to bring the block output to a safe value before modifying the characteristic power curve.

When and how often to change the Characteristic power curve would depend on the particular use case for which the block is being deployed. For example, it could depend on the seasonal variations in weather patterns that affect the power generation from renewable sources, or it could be dependent on other factors that affect external optimization logic.

The output of the block is a function of time of day and the block receives the current time of day as an input through the parameters HOUR, MINUTE and SECOND. These Input parameters can be connected to corresponding parameters of “GetRealTimeClock” function block from the “utilitylib” library.

Block Algorithm

The block calculates the instantaneous value of characteristic power *($P_{CP}(t)$)* for the current time from the “Characteristic power curve” as described above.

$P_{CP}(t)$ is then multiplied by the Multiplication factor M to generate optimum power reference ($P_{OPR}(t)$).

$P_{OPR}(t) = M \times P_{CP}(t)$ (1)

Multiplication factor M can be used to control the degree/ extent of firming obtained by the algorithm.

Intermittency detection

Intermittency detection allows the battery management system to remain idle during times when renewable output power is smooth and does not require any conditioning.

The intermittency detection algorithm tracks the rate of change of the difference *(Pc(t))* between the instantaneous value of renewable power output and the optimum power reference.

Pc(t) = P_TOTALGEN - P_{OPR}(t).................. (2)

The calculated values of *Pc(t)* is then rate limited to generate *($P_{CF}(t)$)* which maintains the maximum rate of change within predefined limits (*INTDETRATELOLM, INTDETRATEHILM*).

$$P_{CF}(t) = \begin{cases} P_c(t), \ if \ INTDETRATELOLM < \Delta P_c(t)/\Delta t < INTDETRATEHILM \\ INTDETRATEHILM \times \Delta t + P_c(t - \Delta t), \ if \ \Delta P_c(t)/\Delta t > INTDETRATEHILM \\ INTDETRATELOLM \times \Delta t + P_c(t - \Delta t), \ if \ \Delta P_c(t)/\Delta t < INTDETRATELOLM \end{cases} \quad (3)$$

$P_{CF}(t)$ is then subtracted from *Pc(t)* to obtain *D(t)*.

If the value of D(t) violates a dead band (INTDETDEVDB), intermittency in power output is then assumed to be present and firming is commenced. Capacity firming continues till value of D(t) remains within the dead band for a period of TD minutes.

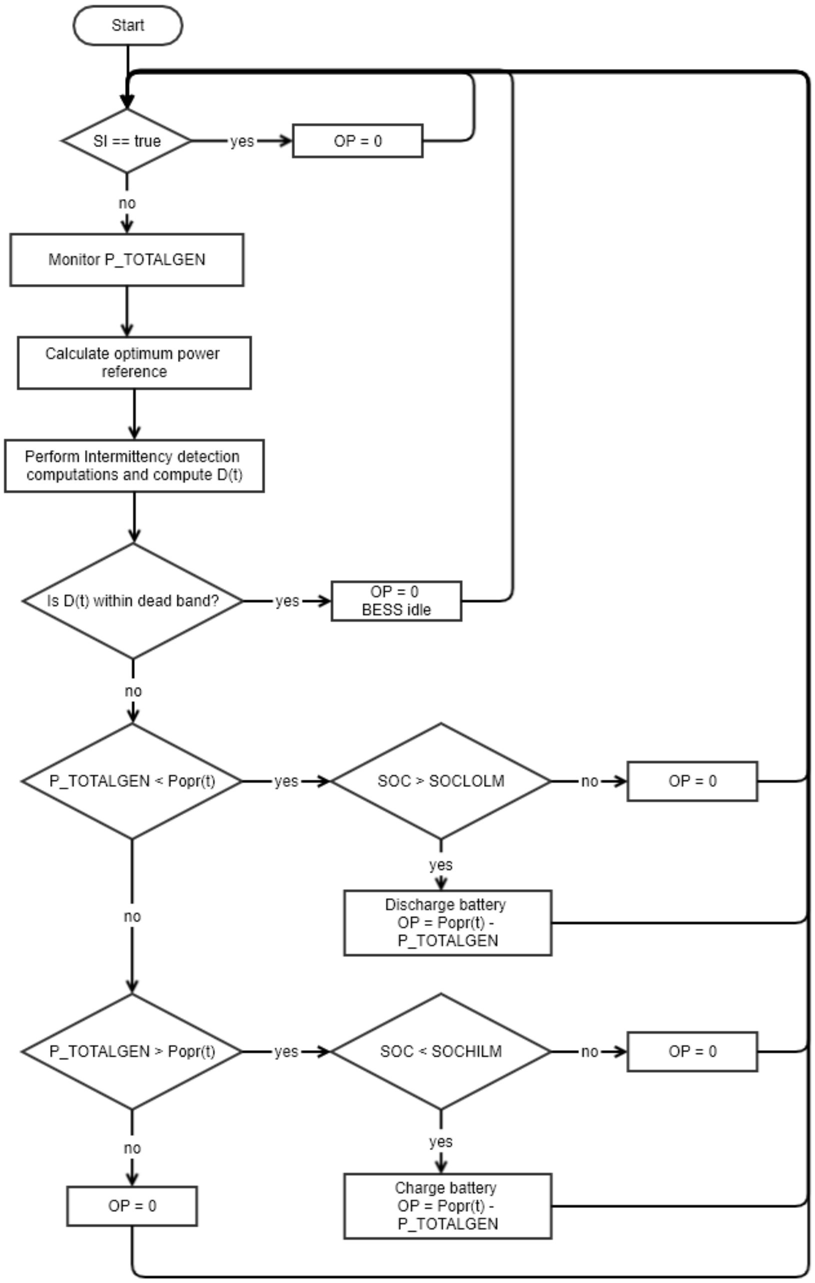
If the intermittency in renewable output power is present, the output (OP) of the block is set to an appropriate value determined by the error and battery state of charge, otherwise the output is set to zero.

If the total generation is less than the optimum power reference, then the battery is discharged if the battery SOC is in a state to support the action. If the total generation is more than the optimum power reference the battery is charged if the battery SOC is in a state to support the action.

$$OP(t) = \begin{cases} P_{OPR}(t) - P\_TOTALGEN, & \text{if intermittency is detected and the battery SOC is not in a wind-up state with respect to the direction of firming action.} \quad (5) \\ 0, & \text{if intermittency is detected and the battery is in a wind-up state with respect to the direction firming of action.} \\ 0, & \text{if intermittency is not detected.} \end{cases}$$

If the Safety Interlock is asserted by external logic/function block, the block drives the output to the safe value of 0.

The following flowchart depicts the operation on the CapacityFirming function block.



The CapacityFirming function block has the following scenarios:

1. **Output Power Firming based on committed power level**

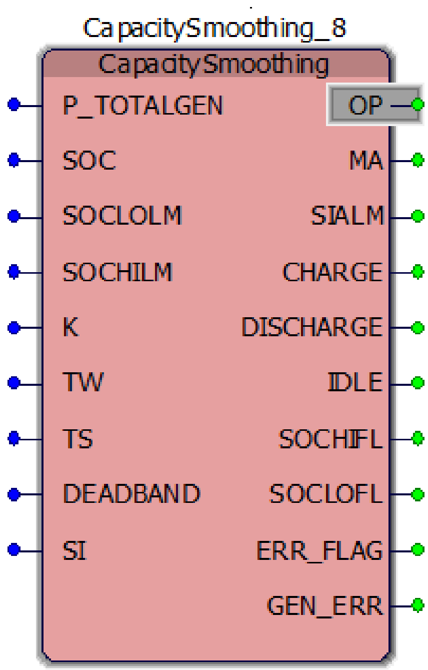
   The characteristic power curve that is an input to the block can be derived based on the committed power levels in the agreement between the power generator and power distributor/ user.

2. **Output Power Firming based on optimization logic**

The characteristic power curve input can be fed from an optimization logic running within or outside the PLC that can derive the input power curve based on historical data or another optimization logic.

CapacitySmoothing

The CapacitySmoothing function block absorbs short term or high frequency variations in the output power delivered to the grid. The battery is either charged/discharged depending on a power reference error correction computed by the smoothing algorithm.



Input

Parameter Name	Description	Data Type
P_TOTALGEN	Total Generation input. The default value is 0.0.	LREAL
K	Smoothing Error Gain. The default value is 1.0.	REAL
TW	Moving Average Time Window in seconds. The default value is 3600.	UDINT
TS	Sampling Time of P_TOTALGEN in seconds. The default value is 1. It is recommended to use	UDINT

Parameter Name	Description	Data Type
	TS as 1 second.	
SI	Safety Interlock - Inhibit Automatic Control. The default value is False.	BOOL
DEADBAND	Smoothing Error Dead Band. The default value is 0.0.	REAL
SOC	State Of charge expressed as %. The default value is 0.0	REAL
SOCLOLM	Minimum battery SOC threshold limit in %. The default value is 0.0.	REAL
SOCHILM	Maximum battery SOC threshold limit in %. The default value is 100.0.	REAL

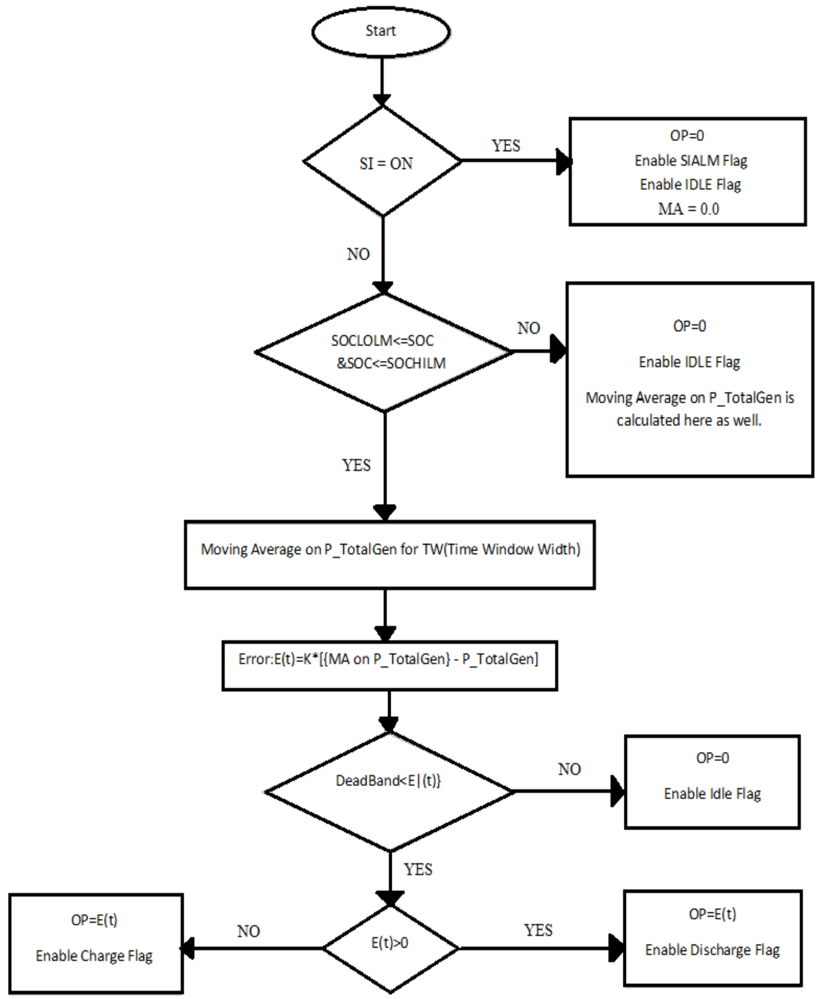
Output

Parameter Name	Description	Data Type
OP	Output Power Reference. The default value is 0.0.	LREAL
MA	Moving Average Value of Total Generation input. This parameter is an interim calculation step and should not be used in the PLC program. The default value is 0.0.	LREAL
SIALM	Safety Interlock Alarm. The default value is False.	BOOL
CHARGE	TRUE if Battery is charging. The default value is False.	BOOL
DISCHARGE	TRUE if Battery is discharging. The default value is False.	BOOL
IDLE	TRUE if Battery is neither charging/ discharging. The default value is False.	BOOL
SOCHIFL	Flag to indicate SOC >=SOCHILM. The default value is False.	BOOL
SOCLOFL	Flag to indicate SOC <=SOCLOLM. The default value is False.	BOOL
ERR_FLAG	Error Flag. The default value is False.	BOOL
GEN_ERR	Error ID. The default value is 0.	USINT

GEN_ERR	Description	Explanation
0	CSNoError	No Errors.
1	CSInvalidSOCLimit	Checks SOCLOLM/SOCHILM beyond 100 or below 0.0 or SOCLOLM>SOCHILM enables this error.
2	CSInvalidTW	Checks Invalid Time Window Value Configured. TW always >2. TW as 2 sec is taken here as Minimum Sampling Rate is 1 sec.
3	CSInvalidTS	Checks that Sampling Rate in seconds is at-least 1 sec or greater.
4	CSTWBelowSamplingTime	Checks that Sampling Time Window in sec is always greater than Sampling Rate.
5	CSTSLessThanExecTime	Error condition where TS is less than Execution cycle of PLC.

Detailed Description

The CapacitySmoothing algorithm is designed to reduce the variability of Total Renewable Power Generation. This function block considers the Total Renewable generation, SOC (within limits) and Smoothing Error Gain (K) as the inputs. It produces a Renewable output smoothed out which is the charging or discharging power passed to the battery based on the following flowchart.



NOTE: The above flowchart represents one iteration of the algorithm.

The Total Renewable Power Generation is considered as an input to the CapacitySmoothing function block. This block calculates the error between the MA (Moving Average) Value on P_TOTALGEN and the instantaneous P_TOTALGEN Value. The MA is calculated on the last TW (Time Window Width) duration which is in seconds. For obtaining MA, samples are collected at TS (Sampling Rate in seconds). Hence total number of samples collected is TW/TS. Cumulative MA is maintained till TW time duration. Once TW is reached, the last MA value keeps updating itself with each new sample. The error [MA on P_TOTALGEN – P_TOTALGEN] is multiplied by K.

In DeadBand Function, the same error (obtained from above, absolute error) is checked against the DeadBand width that is specified. Here, the absolute value of DeadBand is considered. The DeadBand Function output is 0 i.e there is no change or Output power is the error, greater than absolute value of Deadband

Essentially, battery is charged when the OP is negative Power and discharged when Output (OP) is positive power or Idle otherwise.

The function block provides an output in OP after applying the following constraints: -

- The computed Smoothing Error is compared such that it is within the Deadband.
- SI is OFF. The SI can be asserted by other logic/ function blocks and indicate to the Capacity Smoothing block that the output must be driven to 0.
- In-order to handle failure scenario, bring the failure indicative value and provide it to the SI parameter of the CapacitySmoothing block. When SI will be ON, CapacitySmoothing block OP and MA both will drop to 0 immediately and IDLE Flag and SIALM will be ON.
- SOC of the battery is read as an input, compared within SOC limits. The MA on Total Renewable Generation power is calculated despite the SOC is within limits or not. The calculated MA value can be read from MA parameter on the block.
- This function block needs to smooth the MA over a defined TW. The input collection rate is defined as TS. (default value or minimum sampling rate is 1 second.). MA smoothens the output based on last TW/TS samples collected.

The CapacitySmoothing function block can be used in the following scenarios:

1. **Output Power Smoothing**

   The CapacitySmoothing function block can be used to smooth out the Power Variation in Renewable source. (For example, PV power varies throughout the day due to cloud cover).

2. **Renewable Power, Load variability and Area Control Error (ACE) Smoothing**

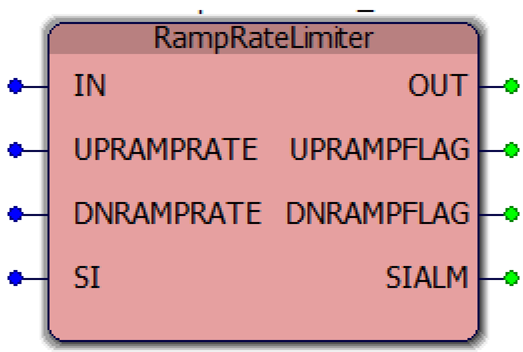
   Apart from Smoothing Renewable Source Generation variability, this function block can also be used to smooth out Load Variability, ACE or a combination of the three by using multiple instances of CapacitySmoothing function block.

RampRateLimiter

The RampRateLimiter function block ramps the output value, when there is change in the input value according to the configured up and down ramp rates.

The RampRateLimiter function block applies ramp rate limiting on the output provided by the Summer block in a controlled scheme in which:

- More than one function block of the Energy Control Library can be used. OR
- Individual function block of the Energy Control Library can be used.



Input

Parameter Name	Description	Data Type
IN	Input value. The default value is NAN.	LREAL
UPRAMPRATE	Upward ramp rate per minute. The default value is 0.0.	REAL
DNRAMPRATE	Downward ramp rate per minute. The default value is 0.0.	REAL
SI	Safety interlock. The default value is 0.0.	BOOL

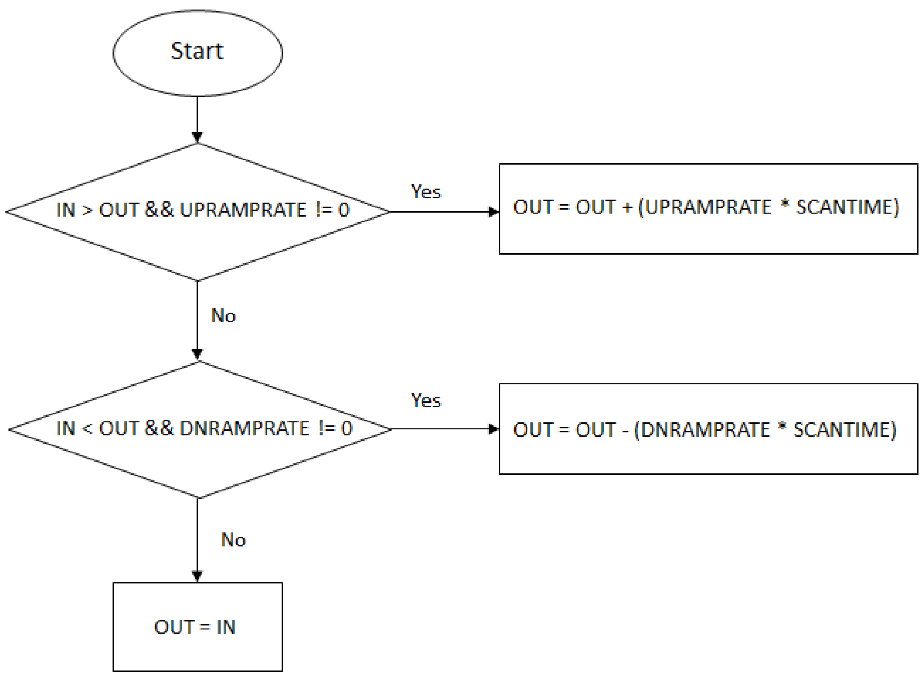
Ouput

Parameter Name	Description	Data Type
OUT	Ramp output. The default value is NAN	LREAL

Parameter Name	Description	Data Type
UPRAMPFLAG	0- OFF 1- ON. The default value is OFF.	BOOL
DNRAMPFLAG	0- OFF 1- ON. The default value is OFF.	BOOL
SIALM	Safety interlock flag. The default value is OFF.	BOOL

Detailed Description

The following flowchart depicts the operation on the RampRateLimiter function block:



NOTE: The above flowchart represents one iteration of the algorithm.

Ramp Rate formula :

UpRampRate

OUT = OUT + UPRAMPRATE * SCANTIME

DownRampRate

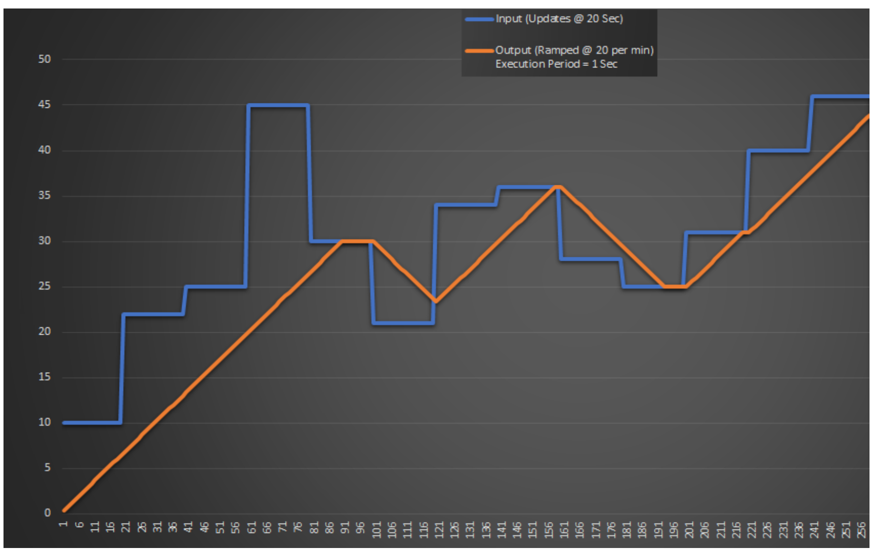
OUT = OUT – DNRAMPRATE * SCANTIME

> **NOTE:** If ramp rates are not specified as 0 or NaN, then no ramping will be applied.

For Example,

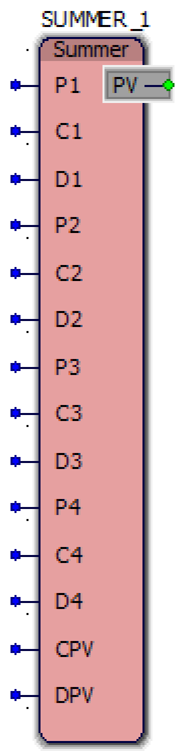
- Input sample: {10, 22, 25, 45, 30, 21, 34, 36, 28, 25, 31, 40, 46}
- Input changes every 12 sec
- Execution period is 1 sec
- Up/Down Ramp rate is 60 per minute

The output of this example is shown in the following graph:



Summer

The Summer block sums four inputs and calculates an output value that can be scaled and biased. Through configuration, you can define a scale factor and bias value for each input.



Input

Parameter Name	Description	Data Type
P1, P2, P3,P4	Input Values P1 to P4. The default value is 0.0.	LREAL
C1,C2,C3,C4	Scaling Factor (C1-C4) for associated block input P1-P4. The default value is 1.0.	REAL
D1,D2,D3,D4	Bias (D1-D4) for associated block input P1 -P4. The default value is 0.0.	REAL
CPV	Overall Scaling Factor for PV. The default value is 1.0.	REAL
DPV	Overall Bias for PV. The default value is 0.0.	REAL

Output

Parameter Name	Description	Data Type
PV	Process Output Value. The default value is 0.0.	64 bit Real Number (LREAL)

Detailed Description

The Summer block uses the following equation to calculate the Process Output Value (PV) value based on four configured inputs.

```
PV = CPV * { ((C1 * P1) + D1) + ((C2 * P2) + D2) + ((C3 * P3) + D3)
+ ((C4 * P4) + D4) } + DPV
```

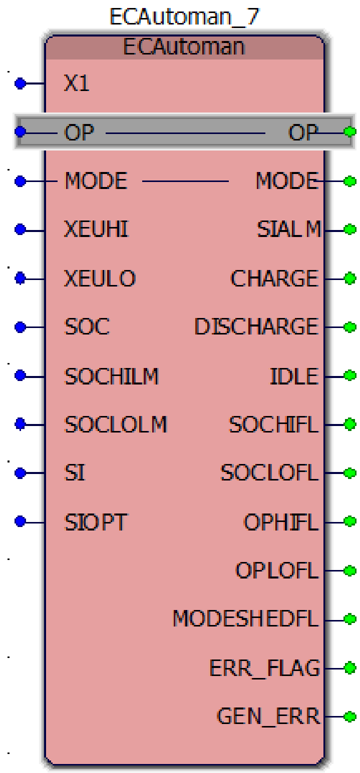
The Summer block receives input values from other function blocks. It evaluates four inputs P1, P2, P3 and P4. It derives value for PV on its calculation of the inputs and the configuration entries for the overall PV scale factor (CPV) , overall PV bias value (DPV) parameters, per input specific scale factor (C[i]) and per input specific Bias factor per (D[i]).

NOTE: To add more than four inputs, multiple function blocks can be stacked together.

Typically, the Summer function block takes inputs from different functions function blocks provided by the Energy Control Library and applies a net effect on the output determined by different algorithms.

ECAutoman

The ECAutoman function block transfers the Reference Power to the PCS. It can transfer both active and reactive power. This function block is typically used as the last element in the Energy Control scheme and its output is provided to a PCS. ECAutoman function block also allows choosing between automatic or manual control mode of operation.



Input

Parameter Name	Description	Data Type
X1	Power Reference. Setpoint to PCS. The default value is 0.0.	LREAL
XEUHI	Engineering Unit High Limit. The default value is 100.0.	LREAL
XEULO	Engineering Unit Low Limit. The default value is -100.0.	LREAL
SOC	State Of charge expressed as %. Input from Battery Management System. The default value is 0.0.	REAL
SOCHILM	Maximum SOC %. The default value is 100.0.	REAL
SOCLOLM	Minimum SOC %. The default value is 0.0.	REAL

Parameter Name	Description	Data Type
SI	Safety Interlock. The default value is False.	BOOL
SIOPT	SHED Mode Option for SI. The default value is 0.	USINT

Input and Output

Parameter Name	Description	Data Type
OP	Output Value. Setpoint to PCS. The default value is 0.0.	ANY (LREAL)
MODE	Operating Mode of the block. The default value is 0.	ANY (USINT)

Output

Parameter Name	Description	Data Type
SIALM	Safety Interlock Alarm Flag. The default value is False.	BOOL
CHARGE	State of Battery is charging. The default value is False.	BOOL
DISCHARGE	State of Battery is Discharging. The default value is False.	BOOL
IDLE	State of Battery is Idle. The default value is False.	BOOL
SOCHIFL	SOC High limit reached flag. The default value is False.	BOOL
SOCLOFL	SOC Low limit reached flag. The default value is False.	BOOL
OPHIFL	OP High limit reached flag. The default value is False.	BOOL
MODESHEDFL	Mode Shed flag. The default value is False.	BOOL
ERR_FL	Error Flag. The default value is False.	BOOL
GEN_ERR	General Error ID. The default value is 0.	USINT

Error ID	Description
0	Indicates no configuration Error.

Error ID	Description
1	Indicates Invalid EU Ranges.
2	Indicates Invalid SOC Limits.
3	Indicates if either MODE (>1) or SIOPT (>1) is out of range.

Detailed Description

The following functions that can be configured and achieved using ECAutoman function block.

1. For OP and MODE, the datatype is ANY, but the connected input datatype should be LREAL for OP and USINT for MODE. If user tries to connect other than the LREAL for OP or USINT for MODE, the function block will not work.
2. For any invalid configuration ERR_FL is set to TRUE and the Error ID value is set to GEN_ERR parameter.

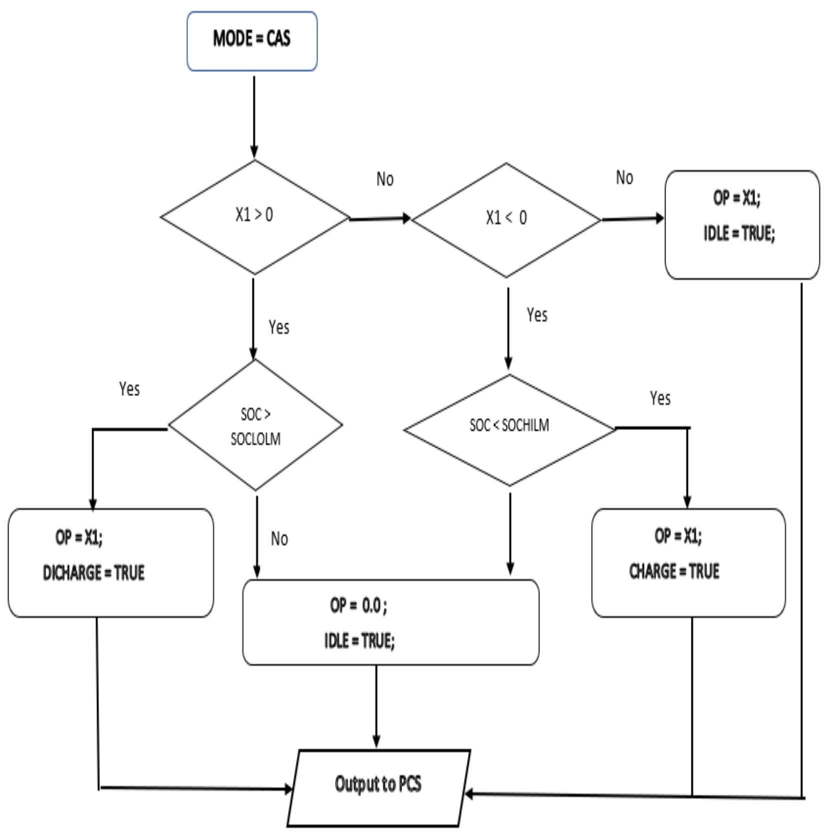
Error ID	Description
GEN_ERR 0	Indicates no configuration Error.
GEN_ERR 1	Indicates Invalid EU Ranges.
GEN_ERR 2	Indicates Invalid SOC Limits.
GEN_ERR 3	Indicates if either MODE (>1) or SIOPT (>1) is out of range.

3. ECAutoman function block operates in 2 Modes such as CAS and MAN. In CAS MODE the function block receives the Power Reference Input from Upstream block and transfers the same to PCS. When the value exceeds its configured Engineering Unit Limits, the value gets clamped to XEUHI or XEULO which ever limit is crossed.
4. When a Safety Interlock is configured, and when that is active the Output (OP) will be set to Zero and SIALM is reported irrespective of the SIOPT configured. If SIOPT is configured as SHEDSAFE then the function block’s MODE would shed to MAN and MODESHEDFL is set to TRUE. Thus, the user needs to take corrective action and bring it to CAS mode again.
5. User can set or change the output only in MAN mode.

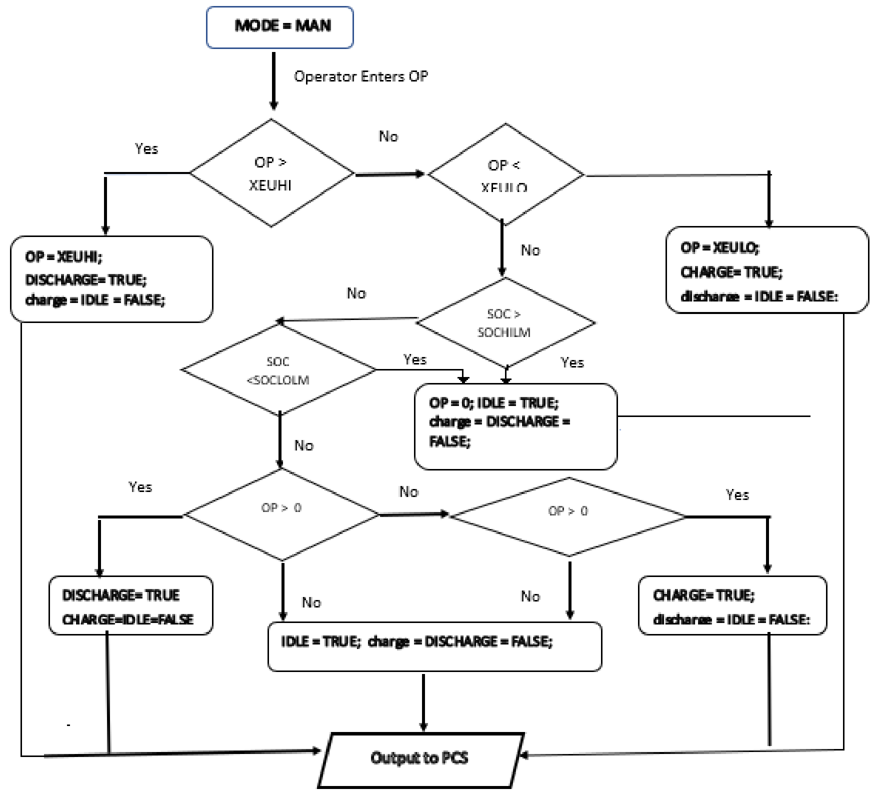
6. In CAS mode, the SOC of the battery is validated against its threshold limits and then the Setpoint(X1) is set to Output. If SOC exceeds its limits, then then the OP is set to zero.
   a. If SOC is at SOCHILM, and if the OP < 0 (charging) then OP would be set to zero.
   b. If SOC is at SOCLOLM, and if the OP > 0 (discharging) then OP would be set to zero.
7. In MAN mode, the OP set by the user is also validated against the configured EURANGES and SOC limits. If OP is not with in these limits, then OP is clamped when the set value exceeds the limits. If SOC reaches it limits, then it gets clamped to zero.
8. The state of the battery either charging, discharging or idle condition is set based on the OP value of CHARGE, DISCHARGE or IDLE Parameters.

Control Logic

1. Mode is in CAS



2. Manual Control: MODE is in MAN

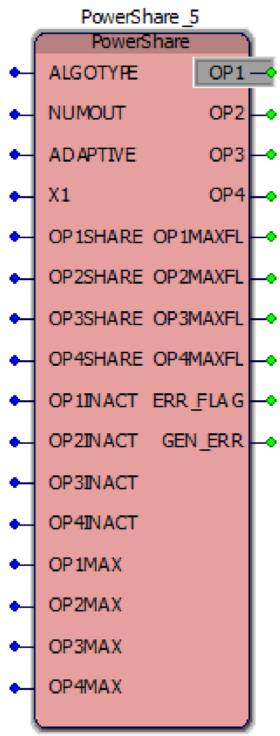


NOTE: The above flowcharts represents one iteration of the algorithm.

PowerShare

The PowerShare function block allows an output power reference to be shared up to four different outputs. The sharing mode can be configured on the function block which allows equal/ unequal sharing.

This function block will be applied when there are more than one PCS/BMS being controlled by the energy controller.



Input

Parameter Name	Description	Data Type
X1	Input Power Reference. The default value is 0.	LREAL
ALGOTYPE	0 = Equal, i.e. IN/NUMOUT 1 = Unequal, i.e., Based on OUT1..4SHARE. The default value is 1.	USINT
NUMOUT	Number of Outputs. The default value is 2.	USINT
ADAPTIVE	Adaptive Logic: When selected, the output is adjusted based on Algo Type and Number of Inputs available. Equal - if NUMOUT changes from x to y, the y outputs will be provided with IN/y as the output power reference. Unequal - if NUMOUT changes from x to y, the y outputs will be provided their % share + share of missing output. The default value is False.	BOOL
OP1SHARE	Output 1 Power Share %. The default value is 0.	REAL
OP2SHARE	Output 2 Power Share %. The default value is 0.	REAL

Parameter Name	Description	Data Type
OP3SHARE	Output 3 Power Share %. The default value is 0.	REAL
OP4SHARE	Output 4 Power Share %. The default value is 0.	REAL
OP1INACT	Output 1 InActive - Input pin to mark if Output is inactive. The values are not pushed in that case. The default value is False.	BOOL
OP2INACT	Output 2 InActive - Input pin to mark if Output is inactive. The values are not pushed in that case. The default value is False.	BOOL
OP3INACT	Output 3 InActive - Input pin to mark if Output is inactive. The values are not pushed in that case. The default value is False.	BOOL
OP4INACT	Output 4 InActive - Input pin to mark if Output is inactive. The values are not pushed in that case. The default value is False.	BOOL
OP1MAX	Maximum OP1 Value. The default value is LREAL Max.	LREAL
OP2MAX	Maximum OP2 Value. The default value is LREAL Max.	LREAL
OP3MAX	Maximum OP3 Value. The default value is LREAL Max.	LREAL
OP4MAX	Maximum OP4 Value.The default value is LREAL Max.	LREAL

Output

Parameter Name	Description	Data Type
OP1	Output 1 Power Reference - Actual Power Reference Output. The default value is 0.	LREAL
OP2	Output 2 Power Reference - Actual Power Reference Output. The default value is 0.	LREAL
OP3	Output 3 Power Reference - Actual Power Reference Output. The default value is 0.	LREAL
OP4	Output 4 Power Reference - Actual Power Reference	LREAL

Parameter Name	Description	Data Type
	Output. The default value is 0.	
OP1MAXFL	Maximum OP1 Value Flag. The default value is OFF.	BOOL
OP2MAXFL	Maximum OP2 Value Flag. The default value is OFF.	BOOL
OP3MAXFL	Maximum OP3 Value Flag. The default value is OFF.	BOOL
OP4MAXFL	Maximum OP4 Value Flag. The default value is OFF.	BOOL
ERR_FLAG	Error Flag – Active when the block detects any errors. The default value is False.	BOOL
GEN_ERR	Error Id – Number denoting the specific error detected by the block. The default value is 0.	USINT

Error ID	Description
0	No Error
1	Invalid NUMOUT
2	Invalid ALGOTYPE
3	Invalid OP1...4SHARE

Detailed Description

PowerShare function block can be used to provide one input and four outputs. Typically, this is used for splitting output power reference to more than 1 connected PCS.

The number of outputs in use is configured by the NUMOUT parameter.

The function block can be configured for splitting outputs using the ALGOTYPE parameter with the following options:

1. 0 – Equal: When this option is selected, the input value is divided equally between the NUMOUT outputs.
2. 1- Unequal: When this option is selected, the input value is divided based on the percentage share configured for each output. The percentage share per output is configured by OP1...4SHARE parameters.

This function block can detect when an output is not available to be supplied with a value. This information is supplied to the Function Block using the OP1...4INACT parameters. When any of these parameters is active, the Function block would continue working based on ALGOTYPE and the configuration of the ADAPTIVE parameter.

When ADAPTIVE is ON and one or more parameters out of OP1...4INACT are active, the function block does the following:

- ALGOTYPE=0 (Equal)
  - If the number of outputs that are inactive = N (>=1) then the OUT parameters are computed by dividing IN into NUMOUT-N equal parts.
- ALGOTYPE=1 (Unequal)
  - The % share of output which cannot be passed to the downstream block/ output will be added to other available outputs. The available outputs will increase their shares to fully accommodate the remaining Power reference.

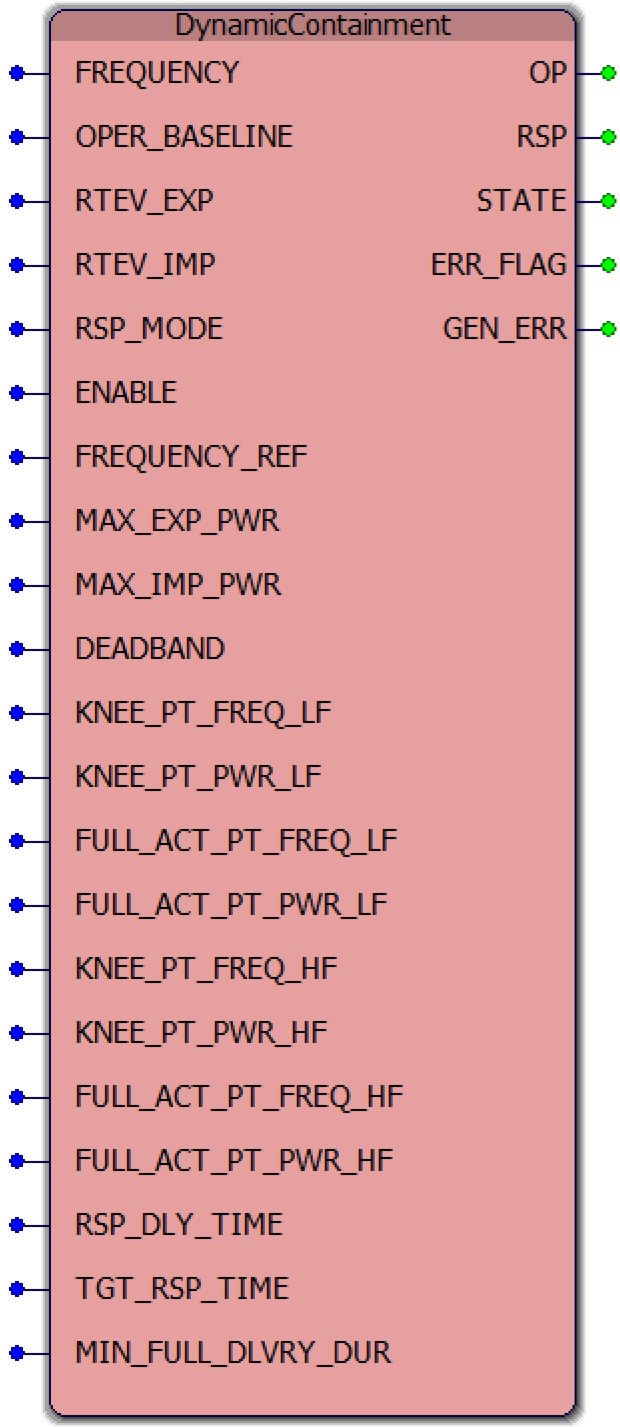
> **NOTE:** Value of an individual output will always be clamped at the respective maximum value configured using OP1...4MAX and the respective flags are made high to indicate to the user.

ERR_FLAG will be ON whenever the block detects an internal error.

GEN_ERR will specify the error numbers as per the following table.

Dynamic containment

The Dynamic Containment is a response service that controls frequency within the statutory range for a sudden demand or generation loss. It is a post-fault frequency management service and delivers a quick response proportional to the frequency deviation. This block provides modes for a low-frequency response, a high-frequency response or both. This service is also necessary in an electrical grid system with low inertia that experience large loss.


DynamicContainment
FREQUENCY
OPER_BASELINE
RTEV_EXP
RTEV_IMP
RSP_MODE
ENABLE
FREQUENCY_REF
MAX_EXP_PWR
MAX_IMP_PWR
DEADBAND
KNEE_PT_FREQ_LF
KNEE_PT_PWR_LF
FULL_ACT_PT_FREQ_LF
FULL_ACT_PT_PWR_LF
KNEE_PT_FREQ_HF
KNEE_PT_PWR_HF
FULL_ACT_PT_FREQ_HF
FULL_ACT_PT_PWR_HF
RSP_DLY_TIME
TGT_RSP_TIME
MIN_FULL_DLVRY_DUR
OP
RSP
STATE
ERR_FLAG
GEN_ERR

Input

Parameter Name	Description	Data Type	Range	Default Value
FREQUENCY_ REF	Grid nominal frequency in Hertz (Hz).	REAL	{50.0, 60.0}	50.0 Hz
FREQUENCY	Real time grid frequency in Hertz (Hz).	LREAL	Non negative range of LREAL data type.	0.0 Hz
OPER_BASELINE	Real time baseline active power output of the system. The units are in engineering units (EU). This corresponds to the baseline power excluding any active dynamic containment response.	LREAL	Range of LREAL data type.	0.0
MAX_EXP_PWR	Maximum active power export capacity of the system. The units are in engineering Units (EU). This corresponds to the contracted volume in the dynamic containment service specification.	LREAL	Non negative range of LREAL data type.	0.0
MAX_IMP_PWR	Maximum active power import capacity of the system. The units are in engineering Units (EU). This corresponds to the contracted volume in the dynamic containment service specification.	LREAL	Non positive range of LREAL data type.	0.0
RSP_MODE	Mode in which the blocks operates. The valid modes are low-frequency response, high-frequency response or both. The valid set of values are as	USINT	{0, 1, 2}	0

Parameter Name	Description	Data Type	Range	Default Value
	follows: • 0 - Both low and high frequency response mode. • 1 - Low frequency response mode. • 2 - High frequency response mode.			
ENABLE	Enable or disable block processing response. The valid set of values are: • false – disabled block frequency response. • true - enabled block frequency response.	BOOL	false, true	false
RTEV_EXP	Real time export energy volume available in the system for Dynamic containment response. The units are in Engineering Units (EU). The units of energy must be consistent with the units of power. The export Response Energy Volume (REV) is calculated by multiplying MAX_EXP_PWR with MIN_FULL_DLVRY_DUR and used directly in computations.	LREAL	Non negative range of LREAL data type.	0.0
RTEV_IMP	Real time import energy volume available in the system for dynamic containment response. The units are in Engineering Units (EU). The units of energy must be consistent with the units of power. The import Response Energy Volume (REV) is	LREAL	Non negative range of LREAL data type.	0.0

Parameter Name	Description	Data Type	Range	Default Value
	calculated by multiplying MAX_IMP_PWR with MIN_FULL_DLVRY_DUR and used directly in computations.			
DEADBAND	Frequency dead band value (as a deviation from nominal system frequency) within which the block does not respond to frequency deviation. This applies to both low frequency and high frequency response. The units are in Hertz.	LREAL	Non negative range of LREAL data type.	0.015 Hz
KNEE_PT_FREQ_LF	Small power delivery knee point frequency component for low frequency response as a deviation from nominal system frequency. The units are in Hertz.	LREAL	Non positive range of LREAL data type.	-0.2 Hz
KNEE_PT_PWR_LF	Small power delivery knee point power component for low frequency response. The units are in percentage of maximum active power export capacity (MAX_EXP_PWR) of the system (% of EU).	LREAL	Non negative range of LREAL data type.	5 %
FULL_ACT_PT_FREQ_LF	Full activation point frequency component for low frequency response as a deviation from nominal system frequency. The units are in Hertz.	LREAL	Non positive range of LREAL data type.	-0.5 Hz
FULL_ACT_PT_PWR_LF	Full activation point power component for low frequency response. The units are in percentage of maximum active power export capacity (MAX_EXP_PWR) of the system (% of	LREAL	Non negative range of LREAL data type.	100 %

Parameter Name	Description	Data Type	Range	Default Value
	EU).			
KNEE_PT_FREQ_HF	Small power delivery knee point frequency component for high frequency response as a deviation from nominal system frequency. The units are in Hertz.	LREAL	Non negative range of LREAL data type.	0.2 Hz
KNEE_PT_PWR_HF	Small power delivery knee point power component for high frequency response. The units are in percentage of maximum active power import capacity (MAX_IMP_PWR) of the system (% of EU).	LREAL	Non positive range of LREAL data type.	-5 %
FULL_ACT_PT_FREQ_HF	Full activation point frequency component for high frequency response as a deviation from nominal system frequency. The units are in Hertz.	LREAL	Non negative range of LREAL data type.	0.5 Hz
FULL_ACT_PT_PWR_HF	Full activation point power component for high frequency response. The units are in percentage of maximum active power import capacity (MAX_IMP_PWR) of the system (% of EU).	LREAL	Non positive range of LREAL data type.	-100 %
RSP_DLY_TIME	Delay in initiation of the response following a deviation in frequency beyond the dead band. It is calculated from the time the deviation is detected. The units are in seconds.	LREAL	Non negative range of LREAL data type.	0.2 s
TGT_RSP_TIME	Time after which a full calculated response is delivered following a deviation in frequency beyond the dead	LREAL	Non negative range of LREAL data	0.7 s

Parameter Name	Description	Data Type	Range	Default Value
	band. It is calculated from the time the deviation is detected. The units are in seconds.		type.	
MIN_FULL_DLVRY_DUR	Minimum full response delivery duration in hours. This is used to calculate Response energy volume (REV) as per the Dynamic containment service specification.	REAL	Non negative range of REAL data type.	0.25 hours

Output

Parameter Name	Description	Data Type	Range	Default Value
OP	Output active power is the set point for the unit participating in Dynamic containment service. It is the sum of OPER_BASELINE and RSP. The Units are in Engineering units (EU).	LREAL	Range of LREAL data type.	0.0
RSP	The dynamic containment response value determined by the block as per the configured response characteristics. When the dynamic containment response is inactive, the value of the parameter is 0.0. The Units are in Engineering units (EU).	LREAL	Range of LREAL data type.	0.0
STATE	The parameter specifies the state of the block. The valid set of values are: • 0 - Frequency response inactive, • 1 – Frequency response active, • 4 - System unable to respond. Frequency deviation is outside the dead band and a response is required. However, the system did not have energy available to be able to respond.	USINT	{0, 1, 4}	0

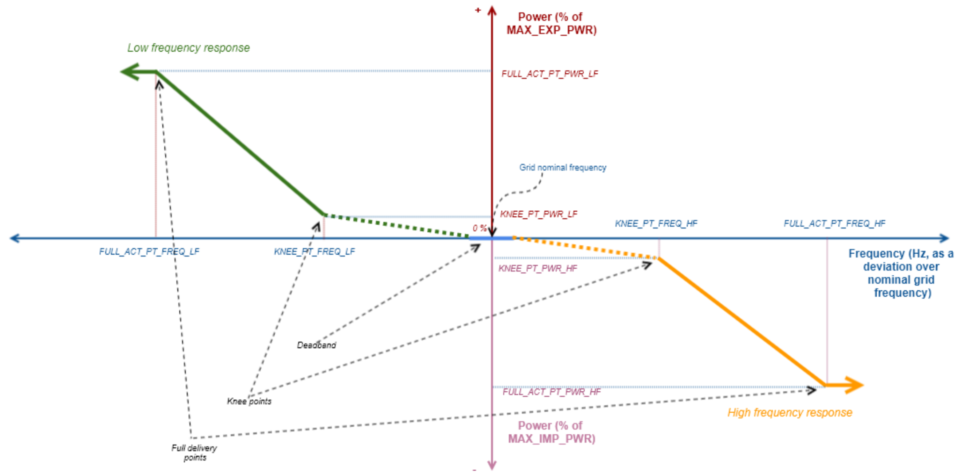
Parameter Name	Description	Data Type	Range	Default Value
	If the mode of the block is set to provide both low and high frequency response and the energy availability in the system is sufficient only for response in one direction, the block can still operate in that direction in a degraded mode. Also, the system can also respond even if the system does not have full Response Energy Volume (REV).			
ERR_FLAG	The parameter when true indicates that the block is in an erroneous state and cannot continue normal operations.	BOOL	false, true	false
GEN_ERR	The parameter indicates the error condition (non zero value) of the block when the block is in an erroneous state and cannot continue normal operations. The valid set of values are: • 0 - No error. • 130 - Invalid grid nominal frequency specified. • 131 - Invalid block response mode specified. • 132 - Invalid value of dead band specified. • 133 - Invalid low frequency response knee point configuration. • 134 - Invalid low frequency response full activation point configuration. • 135 - Invalid high frequency response knee point configuration. • 136 - Invalid high frequency	USINT		0 – No error

Parameter Name	Description	Data Type	Range	Default Value
	response full activation point configuration. • 137 - Invalid response time configuration. • 138 - Invalid minimum full response delivery duration specified. • 139 - Invalid maximum export/import power for Dynamic containment response specified.			

Detailed Description

The Dynamic containment block offers a fast-acting frequency response service post detection of frequency deviation in the system. The block can provide both low frequency and high frequency response in the event of frequency deviation. The block receives the response characteristics as a set of configuration parameters. The block processing is enabled only when the parameter ENABLE is set to true.

> **NOTE:** If the input parameters related to configuration are changed, those would take effect only after disabling and enabling the block (while the dynamic containment response is not active). This applies to parameters FREQUENCY_REF, MAX_EXP_PWR, MAX_IMP_PWR, MIN_FULL_DLVRY_DUR, RSP_DLY_TIME, TGT_RSP_TIME and DEADBAND.



Dynamic containment response profile
(This is over the baseline power requirement.)

The following table describes the service specification of the Dynamic containment block.

Sl No	Service Specification	Details
1	Dead band delivery	0% (within a frequency deviation of +/- DEADBAND over grid nominal frequency.)
2	Small low frequency linear delivery	Small linear delivery proportional to the frequency deviation (up to KNEE_PT_PWR_LF) within the frequency deviation of (KNEE_PT_FREQ_LF, - DEADBAND).
3	Low frequency linear delivery	Linear delivery proportional to the frequency deviation (up to FULL_ACT_PT_PWR_LF) within the frequency deviation of (FULL_ACT_PT_FREQ_LF, KNEE_PT_FREQ_LF).
4	Low frequency full delivery	Full delivery of FULL_ACT_PT_PWR_LF beyond a frequency deviation of FULL_ACT_PT_FREQ_LF.
5	Small high frequency linear delivery	Small linear delivery proportional to the frequency deviation (up to KNEE_PT_PWR_HF) within the frequency deviation of (DEADBAND, KNEE_PT_FREQ_HF).
6	High frequency linear delivery	Linear delivery proportional to the

Sl No	Service Specification	Details
		frequency deviation (up to FULL_ACT_PT_PWR_HF) within the frequency deviation of (KNEE_PT_FREQ_HF, FULL_ACT_PT_FREQ_HF).
7	High frequency full delivery	Full delivery of FULL_ACT_PT_PWR_HF beyond a frequency deviation of FULL_ACT_PT_FREQ_HF.

NOTE: As per the Dynamic containment service requirement, the system output should monotonically progress to the required response for a step change in frequency. The block RSP parameter does satisfy the requirement. It is up to the user to configure the system such that the system satisfies the requirement as well.

As per the Dynamic containment service requirement, the system at any time must be in a state to sustain full delivery response for a period of MIN_FULL_DLVRY_DUR hours. The system must have sufficient energy store/sink capacity depending on the mode of operation. The block receives the current state of energy availability through the parameters RTEV_EXP and RTEV_IMP. If the mode of the block is set to provide both low and high frequency response and the energy availability of the system is sufficient only for response in one direction, the block can still operate in that direction in a degraded mode. Also, the system can also respond even if the system does not have full Response Energy Volume (REV). The parameter STATE provides the state information of the operating block.

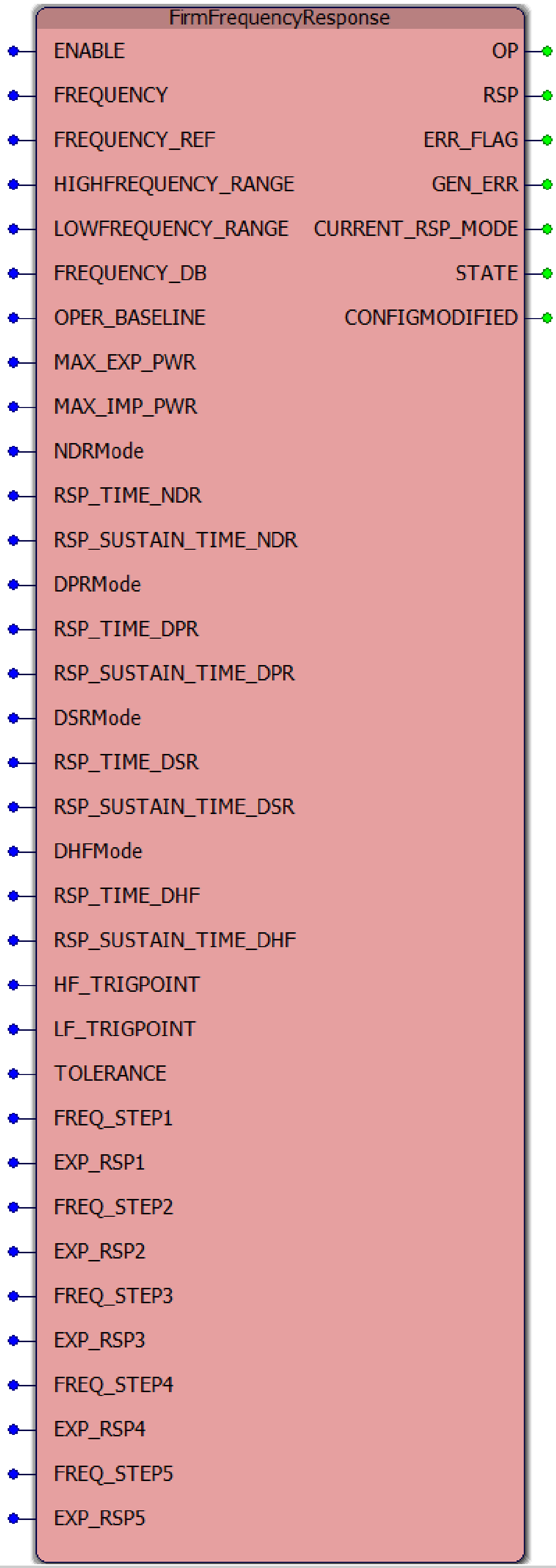
When a frequency deviation is detected by the block, the block provides the response in RSP_DLY_TIME seconds after the detection of frequency deviation and provides the required response with TGT_RSP_TIME seconds after the detection of frequency deviation. Subsequently, the block keeps responding to the deviation in frequency with a value that is proportional to the real time frequency deviation as per the Dynamic containment response profile.

The response is sustained until the frequency reaches the nominal frequency. When determining whether the nominal frequency has been reached, the sensitivity of frequency measurement is taken into account. Sensitivity of Frequency measurement equipment of 0.005 Hertz is assumed in the calculations. When the system reaches the nominal frequency (subject to the Sensitivity of Frequency measurement equipment of 0.005 Hz) the previous response delivery is considered to be complete; any subsequent deviation will begin a new response that will be ramped up to the target value (subject to the delivery times of RSP_DLY_TIME and TGT_RSP_TIME as mentioned above).

> **NOTE:** As per the Dynamic containment service requirement, delay in response from the system following a frequency deviation should be between 0.2 and 0.55 seconds. Also, the required response must be delivered within 0.5 second to 1 second following a frequency deviation. The user must configure the values of parameters RSP_DLY_TIME and TGT_RSP_TIME so that the system as a whole adheres to these specifications.

Firm Frequency Response

The Firm Frequency Response (FFR) is a service that utility requires in the response to a change in system frequency. This change in active power could be either from its initial state or a baseline. The FFR can provide both dynamic and non-dynamic response to the changes in system frequency: The key difference between Non-dynamic and Dynamic is that, when Non-dynamic response is triggered, it sustains at same level irrespective of any further frequency change until it meets sustain time.


FirmFrequencyResponse
ENABLE
FREQUENCY
FREQUENCY_REF
HIGHFREQUENCY_RANGE
LOWFREQUENCY_RANGE
FREQUENCY_DB
OPER_BASELINE
MAX_EXP_PWR
MAX_IMP_PWR
NDRMode
RSP_TIME_NDR
RSP_SUSTAIN_TIME_NDR
DPRMode
RSP_TIME_DPR
RSP_SUSTAIN_TIME_DPR
DSRMode
RSP_TIME_DSR
RSP_SUSTAIN_TIME_DSR
DHFMode
RSP_TIME_DHF
RSP_SUSTAIN_TIME_DHF
HF_TRIGPOINT
LF_TRIGPOINT
TOLERANCE
FREQ_STEP1
EXP_RSP1
FREQ_STEP2
EXP_RSP2
FREQ_STEP3
EXP_RSP3
FREQ_STEP4
EXP_RSP4
FREQ_STEP5
EXP_RSP5
OP
RSP
ERR_FLAG
GEN_ERR
CURRENT_RSP_MODE
STATE
CONFIGMODIFIED

Input (Common to both Static and Dynamic Response)

Parameter Name	Description	Data Type	Default Value
FREQUENCY_REF	Nominal frequency in Hz. User configurable.	Real	50.0
FREQUENCY	Real time grid frequency in Hertz.	LReal	0.0
MAX_EXP_PWR	Contracted maximum active power export capacity of the system. (Max discharging power)	LReal	0.0
MAX_IMP_PWR	Contracted active power import capacity of the system. (Max charging power)	LReal	0.0
ENABLE	External input to inhibit algorithm execution. The valid set of values are as follows: • false – block frequency response processing disabled. • true - block frequency response processing enabled **NOTE:** If Battery Energy Storage System is used to provide the FFR service, then additional (External) logic can stop response to output if any unit specific constraint is met. This should be used to enable or disable the service using ENABLE parameter.	Bool	Off
NDR Mode	Non-Dynamic Response.	Bool	Off
RSP_TIME	Response Time. This indicates the time by which a full response should be reached.	USINT	0s
RSP_SUSTAIN_TIME	Response sustain period or delivery timescale. This indicates the amount of time the response must be sustained. It is represented in seconds.	UINT	0s
DPR Mode	Dynamic Primary Response Mode	Bool	Off
DSR Mode	Dynamic Secondary Response Mode	Bool	Off

Parameter Name	Description	Data Type	Default Value
DHFR Mode	Dynamic High Frequency Response Mode	Bool	Off
OPER_BASELINE	Real time baseline active power output of the system. The units are in engineering units (EU).This corresponds to the baseline power excluding any active dynamic containment response.	LREAL	0.0
RSP_TIME_Sec	Response time for DSR mode when running in combined mode configuration. This indicates the time by which a full response is reached.	USINT	0s
RSP_SUSTAIN_ TIME_Sec	Response sustain period for DSR mode when running in combined mode configuration. This indicates the time by which a full response is reached.	UINT	0s

Output (Common to both Static and Dynamic Response)

Parameter Name	Description	Data Type	Default Value
RSP	Active power reference delivered as a response to a change in system frequency.	LREAL	0.0
OP	Output active power is the set point for the unit participating in the Firm Frequency Response service. The Units are in Engineering units (EU).	LREAL	0.0
STATE	The parameter specifies the state of the block. • 0 - Service inactive. • 1 – Service Active • 2 – Service Denied • 3 - Service Failed • 4 - Service Complete	Enum	Service inactive

Parameter Name	Description	Data Type	Default Value
ERR_FLAG	The parameter when true indicates that the block is in an erroneous state and cannot continue normal operations.	BOOL	False
GEN_ERR	The parameter indicates the error condition, The valid set of values are as follows: • 140 - Invalid Nominal Frequency is configured. • 141 - No Response Mode configured. • 142 - Invalid Response Mode combination • 143 - Invalid High Trigger Frequency • 144 - Invalid Low Trigger Frequency • 145 - Invalid Response Time configured • 146 - Invalid Response sustain time configured • 147 - Invalid Frequency Deadband • 148 - Invalid Frequency Step, • 149 - Invalid Expected Response.	USINT	0
CURRENT_RSP_MODE	The mode of services are as follows:. • 0 - None • 1 - Non-Dynamic Response • 2 - Dynamic Primary Response • 3 - Dynamic Secondary Response • 4 - Dynamic High Frequency Response. Indicates which mode the user has chosen and the block that responds, if the frequency conditions are met.	USINT	0
CONFIGMODIFIED	A flag that indicates one or more configuration parameters that was modified when the block responded.	Bool	False

Parameter Name	Description	Data Type	Default Value
	This will be cleared only when the enable parameter toggles from OFF→ ON, i.e. when the changes to the configuration parameters are consumed.		

Input Parameters (Specific to Static Response)

Parameter Name	Description	Data Type	Default Value
HF_TRIGPOINT	High Frequency Trigger Point.	Real	0.0
LF_TRIGPOINT	Low Frequency Trigger Point.	Real	0.0
TOLERANCE	Indicates the permitted tolerance (±0.01Hz) from the trigger frequency when the response should start.	Real	0.0

Input Parameters (Specific to Dynamic Response)

Parameter Name	Description	Data Type	Default Value
FREQ_STEP	Frequency Deviation (Hz) from the nominal frequency. There are five parameters for this. The values must form a linear curve.	Real	0.0
EXP_RSP	Expected Response corresponding to a particular frequency step defined above. (Percentage of maximum). There are five parameters for this. The values must form a linear curve.	USINT	0.0
FREQUENCY_DB	Frequency dead band to be applied on frequency reference. This is user configurable.	Real	0.015 Hz

NOTE: An example of how Freq_Step and Exp_Rsp can be configured:

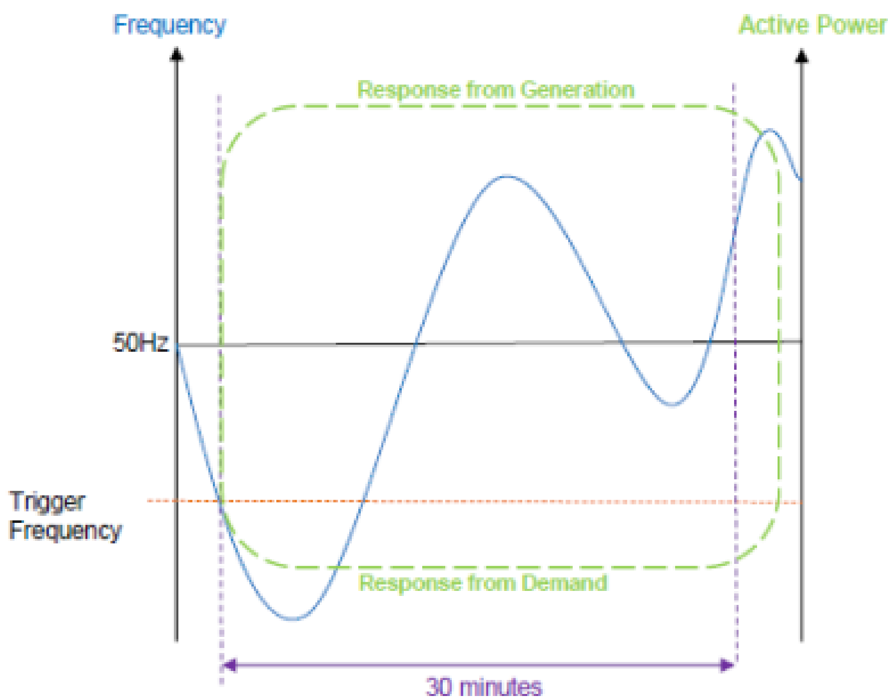
Frequency Deviation (Hz)	Expected Response (Percentage of maximum)
0.1Hz	20
0.2Hz	40
0.3Hz	60
0.4Hz	80
0.5Hz	100

50.1	20%
49.9	20%
50.2	40%
49.8	40%
50.3	60%
49.7	60%
50.4	80%
49.6	80%
50.5	100%
49.5	100%

Detailed description

Non-dynamic frequency response:

The block processing is enabled only when the parameter ENABLE is set to true. When the block is operating in this service mode, it begins to respond to the frequency change once the current frequency goes beyond the trigger frequencies on either direction. The response begins at the configured trigger frequency and within the permitted tolerance. It is ensured that the full response is reached by the configured response time.



Example:

If Tolerance = 0.01Hz, HTF = 50.3, Rsp_Time = 30s and Rsp_Sustain_Period = 30min (1770s), then the response starts when the frequency goes beyond 50.29 (HTF - Tolerance) and ramped up linearly until a full response is reached by 30s. This response will then be sustained in the same direction for 30min.

> **NOTE:** OP = OPER_BASELINE+RSP.

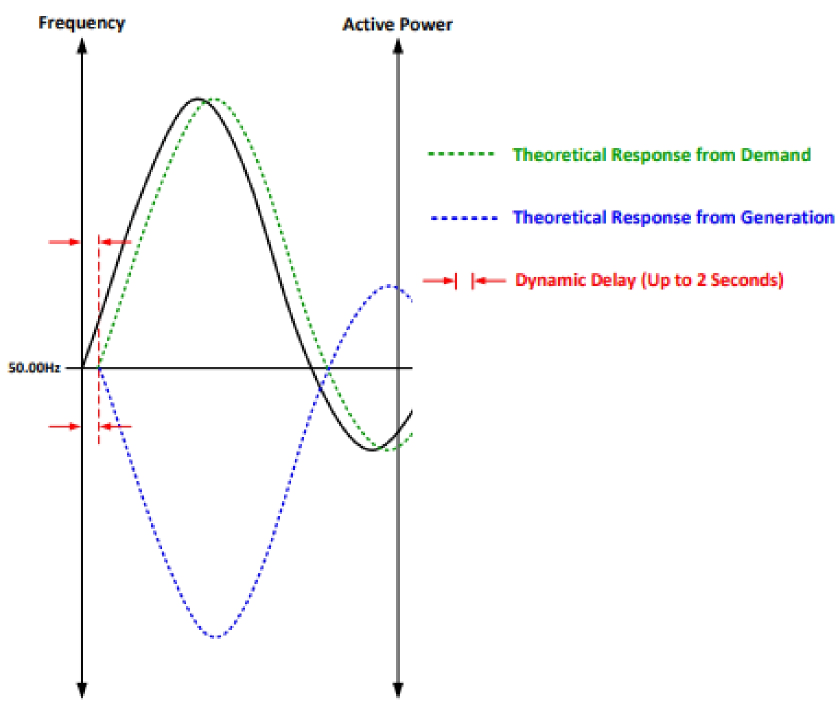
Dynamic frequency response:

The response begins when the frequency deviates beyond the configured Deadband and ensures that full response is reached by the configured response time. The active power changes progressively as the frequency changes. In dynamic response, the change in active power is proportional to the change in frequency as shown previous figure.

> **NOTE:** When the response to first deviation in frequency are ramped up and achieved, the subsequent responses to frequency deviation are achieved within a Max Response Delay of 2sec. This needs to be handled while engineering the POU.

The Variants of Dynamic Response are:

1. Low Frequency : Acts on the frequency deviation which is on the lower side of the nominal frequency
   a. Primary Response Mode: Ensures quick response and sustains for a short duration. For Example, it ensures the response is given within 10s and lasts for 30s.
   b. Secondary Response Mode : Slower response and sustains for a longer duration. For Example, full response is given within 30s and lasts for 30min.
2. High Frequency : Acts on the frequency deviation which is on the higher side of the nominal frequency. It ensures quick response and sustains for a longer duration. For Example, it ensures a full response is given within 10s and lasts for 30min.



Combination of Modes: Scenario where both Low Frequency and High frequency response are supported.

- Case #1 : If DPR and DSR are selected.
  When the first deviation in frequency is on the lower side, then the response will be based on the response time and sustain period of DPR. Beyond the sustain period of DPR, a deviation in frequency on the lower side is responded in DSR mode i.e. Sustain Period of DSR starts to apply. In this case, the response time of DSR does not come into picture as the initial ramping has already happened. Any deviation on the higher frequency side are not accounted for and if such a scenario occurs, RSP is set to 0.
- Case #2 : If DPR and DHFR are selected.
  When the first deviation in frequency is on the lower side, then the response is based on the response time and sustain period of DPR. Beyond the sustain period of DPR, a deviation in frequency on the lower side are not supported and the RSP is set to 0. Any subsequent deviation on the Higher Frequency side are accounted for and sustained for the configured DHFR Sustain Period. The response time of DHFR does not come into picture as the initial ramping has already happened. When the first deviation in frequency is on the higher side, then the response is based on the response time and sustain period of DHFR. Any subsequent deviation on the lower frequency side are accounted for and sustained for the configured DPR sustain period. The response time of DPR does not come into picture as the initial ramping has already happened.
- Case #3 : If DSR and DHFR are selected.
  Same behavior as above with DSR time configurations.
- Case #4 : If all three Modes are selected
  When the first deviation in frequency is on the lower side, then the response is based on the response time and sustain period of DPR. Beyond the sustain period of DPR, a deviation in frequency on the lower side is responded in DSR mode i.e. sustain period of DSR starts to apply. In this case, response time of DSR does not come into picture as the initial ramping has already happened. The subsequent deviation in the high frequency direction is attended in the DHF response mode. Here response time of DHF does not come into picture as the initial ramping has already happened. The responses is vice-versa if the first deviation is in the direction of high frequency. In either case, the configured sustain period of DHF is considered as the delivery timescale.

Modifying Configuration Parameter

The changes to configuration parameters like Response Time, Response Sustain Time, Trigger Frequencies, Frequency Steps , Frequency Reference are validated and consumed only when the Enable Parameter is toggled from OFF - ON.

Block Operation with Configured Frequency Range

Example configuration : HIGHFREQUENCY_RANGE = 60Hz and LOWFREQUENCY_RANGE = 40Hz.

- If the Frequency goes beyond this range, the RSP is set to 0.
- State is set to ‘Service Denied’ or ‘Service Failed’ depending on when the erroneous frequency is encountered.
- Since the block enters these states only under undesirable conditions, recovery from this are only via user intervention.

Thus the block state remains in Service Denied or Failed state until the Enable Parameter is toggled.

Details

Case #1 : If the service starts with a Frequency that is beyond the configured range. Eg , 39.0Hz or 61Hz.

- In this case, the RSP is set to 0 and State to Service Denied.

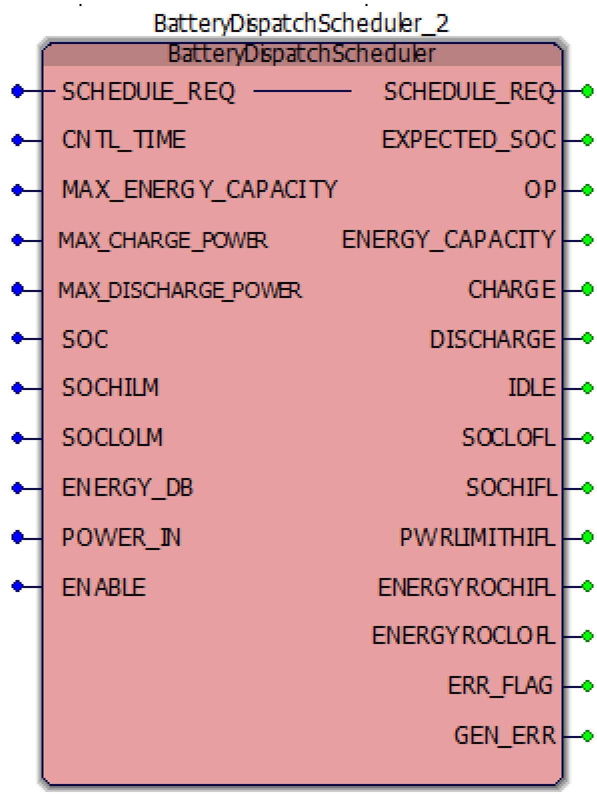
Case #2 : Block currently responds to a Frequency deviation, and eventually the frequency goes beyond the set range.

- In this case, the RSP is set to 0 and State to Service Failed.

Once this situation is encountered, the block does not respond to any further changes in Frequency even if it comes back to normal range.

Battery Dispatch Scheduler

The battery dispatch scheduler is used to schedule charge and discharge power of the battery in a given day and duration. The battery dispatch scheduler can be configured with fixed number of recurring or non-recurring schedules(i.e 10 schedules) based on mode and setpoint. The output power reference is calculated based on the time interval, mode and setpoint. The power and energy units are in engineering units.



Input Parameters

Parameter Name	Description	Datatype	Range	Default Value
SCHEDULE_REQ [10]	Schedule request structure. Configures maximum 10 schedules. Refer to section "Input attributes" in section "Schedule request structure", .	CUSTOMTYPE Refer to section "Schedule Request and Schedule Array Structure Datatype" .		
CNTL_TIME	Controller (PLC/RTU) current time and date.	CUSTOMTYPE • HOUR - USINT	• HOUR: 0 to 23	0

Parameter Name	Description	Datatype	Range	Default Value
		• MINUTE - USINT • SECOND - USINT • DAY - USINT • MONTH - USINT • YEAR - USINT Refer to section "Time and Date Custom Datatype"	• MINUTE: 0 to 59 • SECOND: 0 to 59 • DAY: 1 to 31 • MONTH: 1 to 12 • YEAR (YY): Last 2 digits of a year. Valid range: 0~255 (2000-2255)	
MAX_ENERGY_CAPACITY	Battery maximum energy capacity	REAL	Non negative range of REAL data type	0
MAX_CHARGE_POWER	Battery maximum charging power	REAL	Non negative range of REAL data type	0
MAX_DISCHARGE_POWER	Battery maximum discharging power	REAL	Non negative range of REAL data type	0
SOC	Battery current state of charge(%)	REAL	0-100	0
SOCHILM	Battery SOC high limit	REAL	0-100	100
SOCLOLM	Battery SOC low limit	REAL	0-100	0
ENERGYDB	Acceptable energy difference before raising low or high rate	REAL	Non negative range of REAL data	0

Parameter Name	Description	Datatype	Range	Default Value
	of change of energy flag based on EXPECTED_SOC change and SOC as an input. Flags to monitor are ENERGYROCHIFL or ENERGYROCLOFL. Applicable to schedule MODE Target SOC, Charge Power, and Discharge Power.		type	
POWER_IN	Available power for charging/discharging from battery. POWER_IN = PLoad - PGeneration This input is considered only when Schedule Mode is in following mode: • Charge with available power • Discharge on power deficit • Charge or discharge based on load generation balancing Refer MODE & SETPOINT table	LREAL		0
ENABLE	Enable or disable schedule execution.	BOOL	{0,1}	0

Output Parameters

Parameter Name	Description	Datatype	Range	Default Value
SCHEDULE_REQ [10]	Schedule request structure output parameters. Support maximum 10 schedules. Refer section "Output attributes" in the section "Schedule Request Structure (SCHEDULE_REQ)".	CUSTOMTYPE Refer to section "Schedule Request and Schedule Array Structure Datatype" .		
EXPECTED_SOC	Expected SOC. It is calculated based on RUNTIMESECELAPSED and DURATION of schedule. Applicable to schedule MODE Target SOC, Charge Power, and Discharge Power.	REAL	0-100	0
OP	Output power reference	LREAL		0
ENERGY_CAPACITY	Battery current energy capacity	REAL	Non negative range of REAL data type	0
CHARGE	Battery Charging State.	BOOL	{0,1}	0
DISCHARGE	Battery Discharging State.	BOOL	{0,1}	0
IDLE	Battery Idle State.	BOOL	{0,1}	0
SOCLOFL	SOC low flag	BOOL	{0,1}	0
SOCHIFL	SOC high flag	BOOL	{0,1}	0
PWRLIMITHIFL	The flag is set when the calculated charge & discharge power limits are higher than the	BOOL	{0,1}	0

Parameter Name	Description	Datatype	Range	Default Value
	MAX_CHARGE_POWER and MAX_DISCHARGE_ POWER respectively. Power limit high flag, applicable in "Target SOC" Mode.			
ENERGYROCHIFL	High energy rate of change flag. This flag is set when SOC is updating faster than the EXPECTED_SOC.	BOOL	{0,1}	0
ENERGYROCLOFL	Low energy rate of change flag. This flag is set when SOC is updating slower than the EXPECTED_SOC.	BOOL	{0,1}	0
ERR_FLAG	When the parameter is true, it indicates that the block is in an erroneous state and cannot continue normal operations.	BOOL	{0,1}	0
GEN_ERR	The parameter indicates the error condition. The valid set of values are listed below: • 0 - No errors • 150 - Invalid energy capacity • 151 - Invalid charge power • 152 - Invalid discharge power • 153 - Invalid current input date and time structure size	UINT		0

Parameter Name	Description	Datatype	Range	Default Value
	• 154 - Invalid schedule request structure size • 155 - Invalid time • 156 - Invalid date			

MODE and SETPOINT

The following table describes MODE and SETPOINT value of schedule request.

MODE Value	MODE	Description	SETPOINT Range
0	None	Schedule is not configured.	0
1	Target SOC	In SOC mode, If current SOC is less than set point then charge the battery and if current SOC is more than set point then discharge the battery. The output depends upon energy change and duration of schedule.	SOCLOLM-SOCHILM
2	Charge Power	Charge the battery using the set point as the charge power.	0-MAX_CHARGE_POWER
3	Discharge Power	Discharge the battery using the set point as the discharge power.	0-MAX_DISCHARGE_POWER
4	Charge with available power	This mode charges the battery with the excess power available. i.e Generation > Load and Power_IN is negative. For any deficit in power, the OP will be zero. It charges the battery with available power (POWER_IN) not exceeding the specified SP charge rate.	0-MAX_CHARGE_POWER

MODE Value	MODE	Description	SETPOINT Range
5	Discharge on power deficit	This mode discharges the battery with the deficit power . i.e Generation< Load and Power_IN is positive. For any excess power, the OP will be zero. It discharges the battery with power (POWER_IN) not exceeding the specified SP discharge rate.	0-MAX_ DISCHARGE_ POWER
6	Charge discharge based on load generation balancing	Charge the battery with available power not exceeding the specified SP charge rate when there is excess power. And discharges the battery to meet the deficit power not exceeding the specified SP.	0-MAX_CHARGE_ POWER 0-MAX_ DISCHARGE_ POWER

Schedule Request Structure (SCHEDULE_REQ)

Input Attributes

Attribute Name	Description	Datatype	Range	Default value
MODE	Mode specifies the criteria for charging or discharging battery. For more information, refer Schedule request MODE and SETPOINT value table	USINT	{0,1,2,3,4,5,6}	0
SETPOINT	Set point value based on MODE selection. It	REAL	Based on MODE value	0.0

Attribute Name	Description	Datatype	Range	Default value
	specifies the power required to charge or discharge the battery. For more information, refer Schedule request MODE and SETPOINT value table			
STARTTIME	Start time for charging or discharging the battery. STARTIME is compared with CNTL_ TIME (PLC time) for schedule to start. In start time "SECOND" is an optional configuration.	CUSTOMTYPE TIME_VAL • HOUR-USINT • MINUTE-USINT • SECOND -USINT Refer to section "Time and Date Custom Datatype"	• HOUR: 0 to 23 • MINUTE: 0 to 59 • SECOND: 0 to 59	0:0:0
DURATION	Schedule duration in hours.	REAL	0-23	0
SCHDATE	Schedule start date.	CUSTOMTYPE • DATE_VAL • DAY -USINT • MONTH -USINT • YEAR-	• DAY: 1 to 31 • MONTH: 1 to 12 • YEAR (YY): Last 2 digits of a year.	1/1/2021

Attribute Name	Description	Datatype	Range	Default value
		USINT Refer to section "Time and Date Custom Datatype".	Valid range: 0~255 (2000-2255)	
NUMSCHDAYS	Specify number of days the schedule will run. • 0 - Runs everyday (repeat schedule) • 1-7 - Runs on the specified days	USINT	{0,1,2,3,4,5,6,7}	0

Output Attributes

Attribute Name	Description	Datatype	Range	Default value
STATE	Schedule state • 0- Not Configured. (Schedule not configured, MODE is None) • 1-Queued (Active schedule request) • 2-Running, (Only one schedule request will be active) • 3-Completed, (Inactive schedule,	USINT	{0,1,2,3,4}	0

Attribute Name	Description	Datatype	Range	Default value
	Schedule date expired) • 4-Disabled, (Inactive schedule, Schedule requests with error or ENABLE parameter is false)			
NUMSCHDAYSLEFT	Number of schedule days left.	UDINT	Non negative range of UDINT data type	0
RUNTIMEHRELAPSED	Schedule execution time elapsed in hours	REAL	Non negative range of REAL data type	0
ENERGYCHANGE	Expected amount of energy change based on MODE, SETPOINT and schedule time interval. Calculated during start of schedule.	REAL		0
ENERGYATEND	Expected amount of energy at end of the schedule.	REAL	Non negative range of REAL data type	0
SOCATSTART	Battery SOC during start of schedule	REAL	0-100	0
SOCATEND	Battery SOC during end of schedule	REAL	0-100	0
ERROR	Error occurred during	UINT		0

Attribute Name	Description	Datatype	Range	Default value
	schedule configuration or execution. The valid set of values are listed below: • 0 - No errors • 150 -Invalid Energy Capacity • 153 - Invalid control time • 154 - Invalid schedule • 155 - Invalid time • 156 - Invalid date • 157 - Invalid mode • 158 - Invalid setpoint • 159 - Invalid duration • 160 - Invalid days • 161 - Invalid schedule conflict			

Time and Date Custom Datatype

Define the following Time, Date and TimeDate datatypes structure in ControlEdge Builder.

```
TYPE
        DATE_VAL:
         (* Date structure size 3 bytes *)
         STRUCT
                DAY                  :          USINT;
                MONTH                :          USINT;
                YEAR                 :          USINT;
         END_STRUCT;
END_TYPE
```

```
TYPE
        TIME_VAL:
         (* Time structure size 3 bytes *)
         STRUCT
                HOUR                :                USINT;
                MINUTE              :                USINT;
                SECOND              :                USINT;
         END_STRUCT;
END_TYPE

TYPE
        TIMEDATE:
         (* Date and Time structure size 6 bytes *)
         STRUCT
                TIMEVAL             :                TIME_VAL;
                DATEVAL             :                DATE_VAL;
         END_STRUCT;
END_TYPE
```

Schedule Request and Schedule Array Structure Datatype

Define the following SCHEDULE_REQ and SCHEDULE_ARRAY structure datatypes in ControlEdge Builder.

```
TYPE
        SCHEDULE_REQ:
         (* Schedule request structure size 40 bytes *)
         STRUCT
                ERROR               :                USINT;
                MODE                :                USINT;
                STATE               :                USINT;
                SETPOINT            :                REAL;
                DURATION            :                REAL;
                STARTTIME           :                TIME_VAL;
                SCHDATE             :                DATE_VAL;
                NUMDAYS             :                USINT;
                NUMDAYSLEFT         :                USINT;
                RUNTIMEHRELAPSED    :                REAL;
                ENERGYCHANGE        :                REAL;
                ENERGYATEND         :                REAL;
                SOCATSTART          :                REAL;
                SOCATEND            :                REAL;
         END_STRUCT;
END_TYPE


TYPE
```

```
        (* Array data type for SCHEDULE_REQ *)
            SCHEDULE_ARRAY: ARRAY[1..10] of SCHEDULE_REQ;
END_TYPE
```

Configuring a schedule

The Battery dispatch scheduler supports 10 schedule requests (SCHEDULE_REQ). To configure a new schedule, set MODE other than “None” (0), refer to “MODE and SETPOINT" table and specify schedule request input attributes as described in the schedule request input attributes table. A schedule request with STATE as “Queued” is an active schedule and it gets executed whenever the date and time reaches the specified values and block ENABLE is ON. Schedule states set during a schedule configuration are as follows:

- If MODE is zero then the schedule is considered as not configured and schedule request STATE is set to “Not Configured”. Schedule after "Running" state changes to "Queued" for next day.
- If input attribute values of the schedule request are valid and there is no schedule conflict with another configured schedule, then schedule request STATE is set to “Queued”.
- If input attribute values of schedule request are invalid, then schedule request STATE is set to “Disabled”. Correct the attribute values as indicated by schedule error.
- If input attribute values of schedule request are valid but schedule time (STARTTIME, DURATION) and date (SCHDATE, NUMSCHDAYS) is conflicting with another configured schedule, then schedule request STATE is set to Disabled. Correct the date and time attributes to clear schedule conflict errors.

Re-configuring a schedule

A schedule request attribute can be modified to “Queued” or “Disabled” STATE. In any STATE, if MODE is modified other than zero then the current schedule STATE resets.

> **NOTE:** When schedule request attributes other than MODE is modified in the running state then the effect takes place during next execution period and if it is modified in the complete state, there is no effect.

Clearing a configured schedule

A configured schedule is cleared by setting schedule MODE to zero. The MODE can be set to zero in any schedule STATE such as, "Disable", "Queued", "Running" or "Completed". Once the MODE is set to zero then the schedule STATE changes to "Not Configured".

Recurrence schedule

A schedule is configured to run either everyday or on a given number of days (1 to 7 days). This can be configured using schedule request “NUMSCHDAYS” attribute.

- If NUMSCHDAYS is specified as zero, then schedule executes everyday from start date. The STATE remains in "Queued" and it never sets to "Completed"
- If NUMSCHDAYS is specified between 1 to 7, then schedule executes number of days specified from start date. The STATE remains in "Queued" and it sets to "Completed" on the end date.
- If NUMSCHDAYSLEFT is not zero or the schedule STATE is set to “Completed”, the schedule request output attributes values shows the last execution details, and it resets during next schedule execution. For Example:
  - If NUMSCHDAY is 1 and start date is 11th Nov 2021, then the schedule executes on 11th Nov 2021 and is marked as “Completed”.
  - If NUMSCHDAY is 4 and start date is 11th Nov 2021, then the schedule executes on 11th Nov till 14th Nov 2021 and is marked as “Completed”.

Guidance on using Energy Control Function Block

Output from individual function block vs overall Output

- Individual function blocks produce output power reference as per their algorithms. The Summer Function Block produces the Power reference that is provided to the PCS. This output is further conditioned by using a RampRateLimiter.

- If only one PCS is provided with the output of the control scheme, the output of the ECAutoman block can be mapped in a Display.
- If more than 1 PCS is in use, then the overall output must be determined from the output of RamRateLimiter.
- Alternately, an ECAutoman can be introduced at the output of RampRateLimiter.

Individual PCS Control mode vs Composite Mode

- ECAutoman represents the final control element connecting to 1 PCS.
- If only 1 PCS is provided with the output of the control scheme, the Control Mode can be defined using the MODE parameter of the ECAUTOMAN Block.
- If more than one PCS is in use, then the overall MODE must be applied at the output of All ECAutoman or must be programmed by using a SEL block to select Control vs Manual output.
- Alternately, an ECAutoman can be introduced at the output of RampRateLimiter and the Control Mode can be Changed before the value is split between multiple PCS.

Charge/ Discharge/ Idle status

- ECAutoman represents the final control element connecting to a PCS.
- Charge/ Discharge/ Idle flag of the ECAutoman represents the command being sent to the PCS.
- If more than one PCS is in use, then the overall Charge/ Discharge/ Idle status can be inferred at the Output of the RampRateLimiter as follows.
    - OP > 0.0 = Discharge
    - OP < 0.0 = Charge
    - OP = 0.0 = Idle
- Alternately, an ECAutoman can be introduced at the output of RampRateLimiter and the Charge/ Discharge/ Idle status can be readily used.

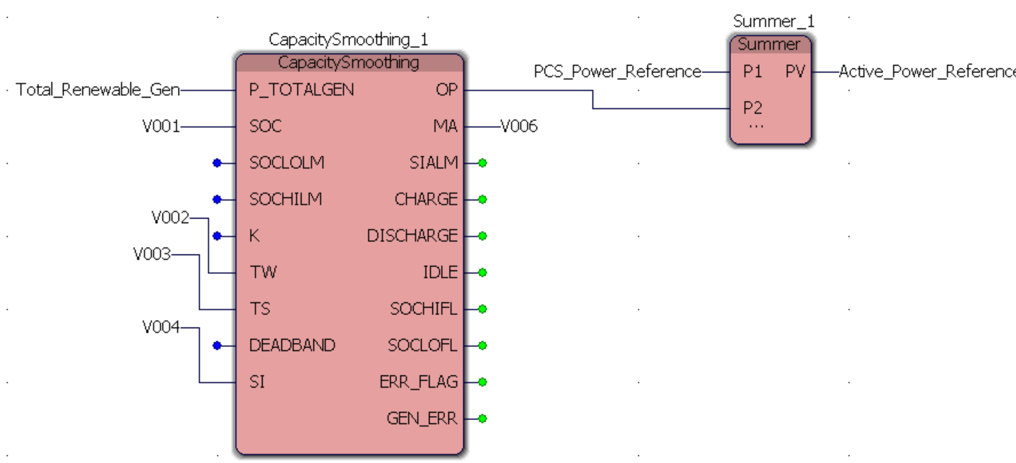
Blocks which produce output error correction on Active Power Reference

The function blocks namely CapacitySmoothing, FrequencyRegulation, RampRateControl and CapacityFirming produce the error correction on the active power reference based on their individual algorithms.

In scenarios where these blocks are used independently, active power reference must be added to the output obtained from these function blocks to obtain the active Power Reference for PCS. The Summer function block can be used to Sum the output of these function blocks to an active power reference which can be obtained by one of the following methods:

- If PeakShaver function block is in use, its OP can be added to the error correction OP produced by the above mentioned blocks to produce an active power reference to be supplied to the PCS.
- Current active power reference of the PCS can be read and the error correction be applied to it using a Summer block.
- Active power reference is being provided by an external source, the value provided by the external source could be summed to the error correction OP produced by the above mentioned blocks.

A typical scenario, where to the OP, which is the error correction obtained from CapacitySmoothing FB, PCS power reference needs to be summed to obtain the active power reference.



Similar approach can be followed for other function block whose output (OP) is the error correction on active power reference.

NOTICES

Trademarks

Experion® is a registered trademark of Honeywell International, Inc.

ControlEdge™ is a trademark of Honeywell International, Inc.

OneWireless™ is a trademark of Honeywell International, Inc.

Other trademarks

Microsoft and SQL Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Trademarks that appear in this document are used only to the benefit of the trademark owner, with no intention of trademark infringement.

Third-party licenses

This product may contain or be derived from materials, including software, of third parties. The third party materials may be subject to licenses, notices, restrictions, and obligations imposed by the licensor. The licenses, notices, restrictions and obligations, if any, may be found in the materials accompanying the product, in the documents or files accompanying such third party materials, in a file named third_party_licenses on the media containing the product, or at http://www.honeywell.com/en-us/privacy-statement.

Documentation feedback

You can find the most up-to-date documents in the Support section of the Honeywell Process Solutions website at: https://process.honeywell.com/us/en/support/product-documents-downloads

If you have comments about Honeywell Process Solutions documentation, send your feedback to: hpsdocs@honeywell.com

Use this email address to provide feedback, or to report errors and omissions in the documentation. For immediate help with a technical problem, contact HPS Technical Support through your local Customer Contact Center, or by raising a support request on the Honeywell Process Solutions Support website.

How to report a security vulnerability

For the purpose of submission, a security vulnerability is defined as a software defect or weakness that can be exploited to reduce the operational or security capabilities of the software.

Honeywell investigates all reports of security vulnerabilities affecting Honeywell products and services.

To report a potential security vulnerability against any Honeywell product, please follow the instructions at:

https://www.honeywell.com/en-us/product-security.

Support

For support, contact your local Honeywell Process Solutions Customer Contact Center (CCC). To find your local CCC visit the website, https://process.honeywell.com/us/en/contact-us.

Training classes

Honeywell holds technical training classes that are taught by process control systems experts. For more information about these classes, contact your Honeywell representative, or see http://www.automationcollege.com.