User Manual Function Block AGA Report No. 3 1992 Version

Rev. A 12/2013

Doc. Code.: MU***

altus

www.altus.com.br

According to current legislation in force in Brazil, the Consumer Defense Code, we provide the following information regarding personal safety and installation by the client.

The industrial automation equipment built by Altus is strong and reliable due to the stringent quality control it is subjected to. However the electronic industrial control equipment (programmable controllers, numerical commands, etc.) can cause damage to the machines or processes through their controllers when there are defective components and programming or installation errors. This can even put human lives at risk.

The user should consider the possible consequences of these defects and should provide additional external installations for security so that, if necessary, the security of the system can be maintained especially during the initial installation and testing.

It is essential to completely read the manuals and/or about the technical characteristics of the product before its installation or use.

The equipments manufactured by Altus don’t cause direct environmental hazards and don’t produce any kind of pollution during its utilization. However, regarding the equipments discarded, it is important to stress that any electronic components inside these products may cause environmental impacts if discharged incorrectly. Therefore, it’s strongly recommended that the product is sent to recycle plants to be discarded with the proper waste treatment.

Altus guarantees that its equipment against genuine production defects for a period of twelve months starting at the shipping date. This guarantee is given in terms of factory maintenance, that is to say, the transportation costs of returning to factory will be borne by the client. The guarantee will be automatically suspended if there are modifications introduced to the equipment by personnel not authorized by Atlus. Altus are exempt from any responsibility with regard to repairs or replacement parts owing to faults created by outside influences, through inappropriate use, as well as the result of accidents or force maneuver.

Altus guarantees that its equipment works in accordance with the clear instructions contained in their manuals and/or the technical features, not guaranteeing the success of any particular type of application of the equipment.

Altus does not acknowledge any other guarantee, directly or implied, mainly when end customers are dealing with third-party suppliers The requests for additional information about the supply, equipment features and/or any other Altus services must be made in written form. Altus’s address can be found on the back cover. Altus is not responsible for supplying information about its equipment without formal request.

COPYRIGHTS

Table of Contents

1. Introduction

Function block AGA3_Flow computes mass and volume flows according part 4 of AGA Report No. 3, 1992 version.

This standard applies to steady-state mass flow conditions for fluids that, for all practical purposes, are considered to be clean, single phase, homogeneous, and Newtonian and have pipe Reynolds numbers of 4,000 or greater. All gases, most liquids, and most dense phase fluids associated with the petroleum, petrochemical, and natural gas industries are usually considered Newtonian fluids.

The standard demands that measurement be made with flange-tapped, concentric, square-edged orifice meters. These orifice meters must be built according part 2 of AGA Report No. 3.

Some limits must be observed for applying the standard:

- minimum bore diameter: 11.4 mm
- minimum tube diameter: 50 mm
- Beta ratio (ratio between bore and tube diameter) must be in the range 0.1 to 0.75.
- Pipe Reynolds number greater or equal 4000

The function block takes as inputs:

- Three analog input variables acquired from appropriate transmitters:
  - Temperature of actual condition
  - Absolute pressure of actual condition
  - Differential pressure
- Cutoff differential pressure (bellow this differential pressure, flows are set to zero)
- Position of absolute pressure tap: upstream or downstream
- Temperature of base condition
- Absolute upstream pressure of base condition
- Bore features:
  - Diameter at reference temperature
  - Reference temperature where diameter was measured
  - Coefficient of thermal expansion
- Tube features:
  - Diameter at reference temperature
  - Reference temperature where diameter was measured
  - Coefficient of thermal expansion

- Definition if densities are informed as inputs, or calculated by the function block:
    - For natural gas, there are two alternatives:
        - Densities may be informed as inputs. They can be calculated previously, for instance, using AGA-8 detailed method.
        - Densities may also be calculated by this function block, using subsequent inputs (ideal gas specific gravity and compressibilities). These compressibilities can be calculated previously by AGA-8 detailed or gross method.
    - For other fluids, densities must be informed as inputs.
- Density in actual condition informed as input. Optional for natural gas, required for other fluids.
- Density in base condition informed as input. Optional for natural gas, required for other fluids.
- Ideal gas specific gravity (also known as "ideal gas relative density") of natural gas. Required for natural gas if densities are not informed as inputs. Not required for other fluids. For users that want to use real gas specific gravity instead of ideal gas specific gravity, this user manual shows the equation used to convert real gas specific gravity in ideal gas specific gravity (see chapter **Real and Ideal Specific Gravity**).
- Compressibility of gas in actual condition. Required for natural gas if densities are not informed as inputs. Not required for other fluids.
- Compressibility of gas in base condition. Required for natural gas if densities are not informed as inputs. Not required for other fluids.
- Viscosity:
    - For natural gas, the standard recommends constant value 0.010268 centipoise, but the user can inform different values if necessary.
    - For other fluids, the appropriate value must be informed.
- Isentropic exponent:
    - For natural gas, the standard recommends constant value 1.3, but the user can inform different values if necessary.
    - For other fluids, the appropriate value must be informed.
    - The special constant value -1 must be informed for incompressible fluids.

The function block calculates the following outputs:

- Mass flow
- Volumetric flow at actual condition (actual temperature and actual absolute pressure)
- Volumetric flow at base condition (base temperature and base absolute pressure)
- Density at actual condition (actual temperature and actual absolute pressure)
- Density at base condition (base temperature and base absolute pressure)
- Warning and error flags

The function block is delivered within a library (AGA Report No 3 – 1992).

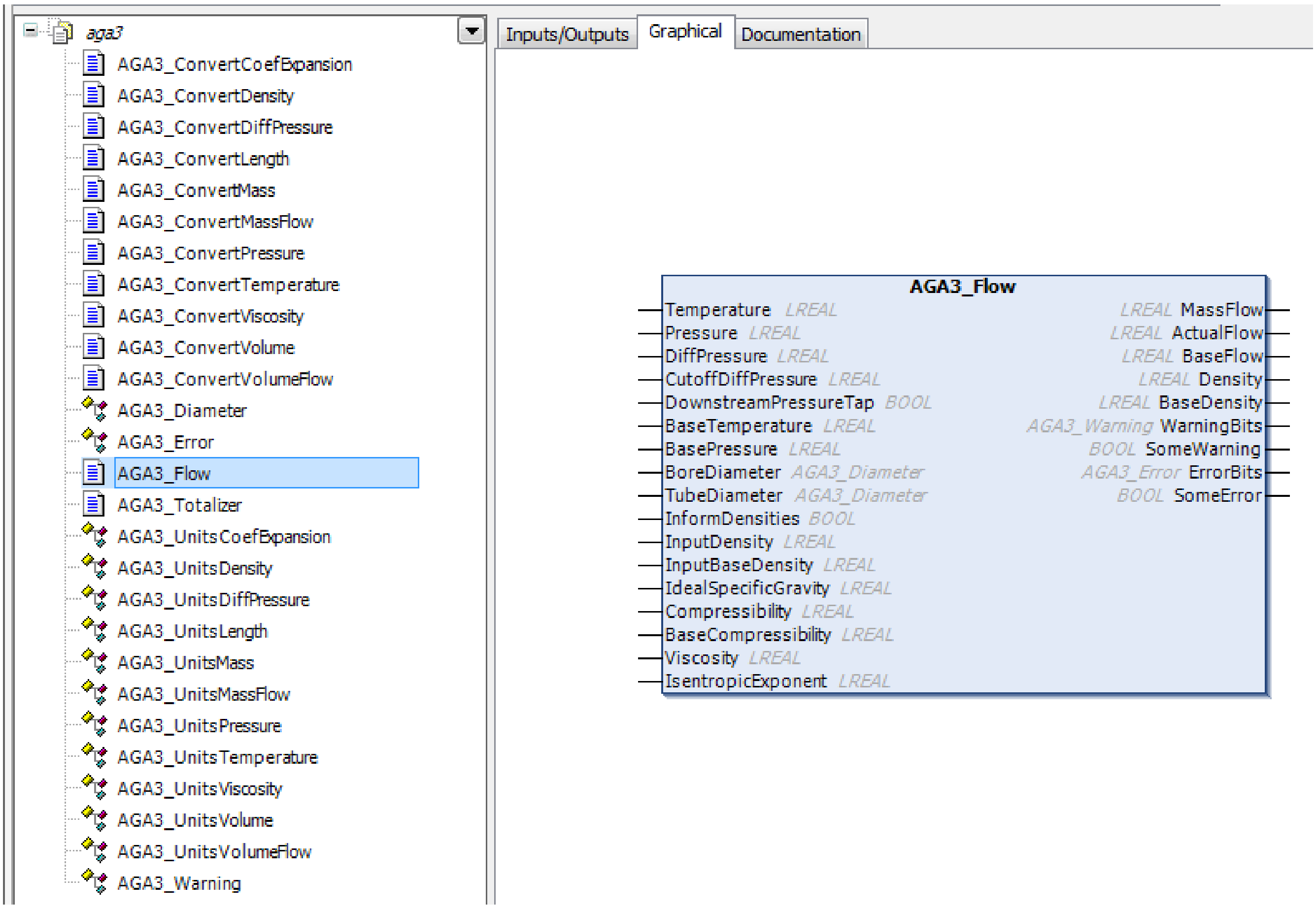

Figure 1-1. Components of library AGA Report No 8 – Detail Method

The main features of the function block and associated library are the following:

- Compatible with Nexto Series CPUs
- 64-bit floating point precision arithmetic
- Low processing time
- Low memory consumption

- Besides function block AGA3_Flow, the library also contains:
  - Auxiliary data structures used in AGA3_Flow function block (bore and tube diameter features, warnings and errors)
  - Function block AGA3_Totalizer, that can be used for calculating totalized mass or volume.
  - Auxiliary functions for converting units of:
    - Coefficient of thermal expansion
    - Density
    - Differential pressure
    - Length
    - Mass
    - Mass flow
    - Pressure
    - Temperature
    - Viscosity
    - Volume
    - Volume flow
  - Auxiliary enumerations defining the allowed units used in functions for converting units. The supported units are those used in Metric, SI, US (United States) and IP (inch-pound) systems, according table 4-5 of part 4 of AGA Report No. 3.

Documents Related to this Manual

In order to obtain additional information regarding the Nexto Series, other documents (manuals and technical features) besides this one, may be accessed. These documents are available in its last version on the site http://www.altus.com.br.

Each product has a document designed by Technical Features (CE), where the product features are described. Furthermore, the product may have Utilization Manuals (the manuals codes are listed in the CE).

For instance, the NX1001 module has the information for utilization features and purchasing on its CE. On another hand, the NX5001 has, besides the CE, a User Manual (MU).

It is advised the following documents as additional information source:

Code	Description	Language
CE* CT*** CS*****	Function Block AGA Report No. 3 - 1992 – Technical Characteristics Function Block AGA Report No. 3 - 1992 – Caracteristicas Técnicas Function Block AGA Report No. 3 - 1992 – Especificaciones y Configuraciones	English Portuguese Spanish
MU* MU*** MU*****	Function Block AGA Report No. 3 - 1992 – User Manual Function Block AGA Report No. 3 - 1992 – Manual de Utilização Function Block AGA Report No. 3 - 1992 – Manual Del Usuario	English Portuguese Spanish
CE114000 CT114000 CS114000	Nexto Series – Technical Characteristics Série Nexto – Características Técnicas Serie Nexto – Especificaciones y Configuraciones	English Portuguese Spanish
CE114700 CT114700 CS114700	Nexto Series Backplane Racks Technical Characteristics Características Técnicas dos Bastidores da Série Nexto Características Técnicas de los Bastidores de la Serie	English Portuguese Spanish

	Nexto	
CE114900 **CT114900** **CS114900**	NX4010 Redundancy Link Module Technical Characteristics Características Técnicas do Módulo de Redundância NX4010 Características Técnicas del Módulo de Redundancia NX4010	English Portuguese Spanish
CE114902 **CT114902** **CS114902**	NX5001 PROFIBUS-DP Master Technical Characteristics Características Técnicas do Mestre PROFIBUS DP NX5001 Especificaciones y Configuraciones Maestro PROFIBUS-DP NX5001	English Portuguese Spanish
CE114903 **CT114903** **CS114903**	Ethernet Module NX5000 Technical Characteristics Características Técnicas do Módulo Ethernet NX5000 Especificaciones y Configuraciones Modulo Ethernet NX5000	English Portuguese Spanish
CT112500	Características Técnicas do Painel de Controle de Redundância PX2612	Portuguese
MU214600 **MU214000** **MU214300**	Nexto Series User Manual Manual de Utilização Série Nexto Manual del Usuario Serie Nexto	English Portuguese Spanish
MU214605 **MU214100** **MU214305**	Nexto Series CPUs User Manual Manual de Utilização UCPs Série Nexto Manual del Usuario UCPs Serie Nexto	English Portuguese Spanish
MU299609 **MU299048** **MU299800**	MasterTool IEC XE User Manual Manual de Utilização MasterTool IEC XE Manual del Usuario MasterTool IEC XE	English Portuguese Spanish
MP399609 **MP399048** **MP399800**	MasterTool IEC XE Programming Manual Manual de Programação MasterTool IEC XE Manual de Programación MasterTool IEC XE	English Portuguese Spanish
MU214601 **MU214001** **MU214301**	NX5001 PROFIBUS DP Master User Manual Manual de Utilização Mestre PROFIBUS DP NX5001 Manual del Usuario Maestro PROFIBUS DP NX5001	English Portuguese Spanish
MU219000 **MU209000**	Ponto Series Utilization Manual Manual de Utilização da Série Ponto	English Portuguese
MU209508	Manual de Utilização Cabeça PROFIBUS PO5063V1 e Cabeça Redundante PROFIBUS PO5063V5	Portuguese
MU219511 **MU209511**	PO5064 PROFIBUS Head and PO5065 Redundant PROFIBUS Head Utilization Manual Manual de Utilização Cabeça PROFIBUS PO5064 e Cabeça Redundante PROFIBUS PO5065	English Portuguese
MU209020	Manual de Utilização Rede HART sobre PROFIBUS	Portuguese

Table 1-1. Related Documents

Visual Inspection

Before resuming the installation process, it is advised to carefully visually inspect the equipments, verifying the existence of transport damage. Verify if all parts requested are in perfect shape. In case of damages, inform the transport company or Altus distributor closest to you.

> **CAUTION:**
> **Before taking the modules off the case, it is important to discharge any possible static energy accumulated in the body. Touch (with bare hands) any metallic grounded surface before handling the modules. Such procedure guaranties that the module static energy limits are not exceeded.**

It’s important to register each received equipment serial number, as well as software revisions, in case they exist. This information is necessary, in case the Altus Technical Support is contacted.

Technical Support

For Altus Technical Support contact in São Leopoldo, RS, call +55 51 3589-9500. For further information regarding the Altus Technical Support existent on other places, see http://www.altus.com.br or send an email to altus@altus.com.br.

If the equipment is already installed, you must have the following information at the moment of support requesting:

- The model of the used equipments and the installed system configuration
- The CPU serial number
- The equipment revision and the executive software version, written on the tag fixed on the product side
- CPU operation mode information, acquired through MasterTool IEC XE
- The application software content, acquired through MasterTool IEC XE
- Used program version

Warning Messages Used in this Manual

In this manual, the warning messages will be presented in the following formats and meanings:

DANGER:
Reports potential hazard that, if not detected, may be harmful to people, materials, environment and production.

CAUTION:
Reports configuration, application or installation details that must be taken into consideration to avoid any instance that may cause system failure and consequent impact.

ATTENTION:
Identifies configuration, application and installation details aimed at achieving maximum operational performance of the system.

2. Technical Description

This chapter presents the technical features of Library AGA Report No. 3 – 1992.

General Features

Feature	Description
Calculation Method	AGA Report No. 3, part4, 1992 version.
Applicable fluids	Fluids that, for all practical purposes, are considered to be clean, single phase, homogeneous, and Newtonian and have pipe Reynolds numbers of 4000 or greater. All gases, most liquids, and most dense phase fluids associated with the petroleum, petrochemical, and natural gas industries are usually considered Newtonian fluids.
Measurement device	Flange-tapped, concentric, square-edged orifice meters, built according part 2 of AGA Report No. 3.
AGA-3 1992 Standard Limits	Minimum bore diameter: 11.4 mm Minimum tube diameter: 50 mm Beta ratio: 0.1 to 0.75 Pipe Reynolds number: greater or equal 4000
Compatible CPUs	Nexto series: NX3030, NX3020, NX3010
Typical execution time (AGA3_Flow)	0.15 ms
Worst case execution time (AGA3_Flow)	0.3 ms
Data memory allocated for variables of each instance of function block	344 bytes
Units used by function block inputs and outputs (AGA3_Flow and AGA3_Totalizer)	Temperatures: ºC Pressures: bar Differential pressures: millibar Diameters: mm Coefficients of thermal expansion: mm/(mm * ºC) Densities: kg/m^3 Viscosity: cP Mass: kg Mass flow: kg/hr Volume: m^3 Volume flow: m^3/hr
Units supported by Coefficient of Thermal Expansion conversion function	mm/(mm * ºC), mm/(mm * ºK), in/(in * ºF), ft/(ft * ºF)
Units supported by Density conversion function	kg/m^3, lbm/ft^3
Units supported by Differential Pressure conversion function	Pa, psia, millibar
Units supported by Length (Diameter) conversion function	mm, m, in, ft
Units supported by Mass conversion function	kg/hr, kg/s, lbm/hr
Units supported by Pressure conversion function	Pa, psia, bar
Units supported by Temperature conversion function	ºC, ºK, ºF

Units supported by Viscosity conversion function	cP, lbm/(ft * s), Pa * s
Units supported by Volume conversion function	m^3, ft^3
Units supported by Volume Flow conversion function	m^3/hr, m^3/s, ft^3/hr

Table 2-1. General Features of Library AGA Report No. 3 – 1992

Compatibility with Other Products

Code	Description
MT8500	MasterTool IEC XE version 1.00 or onwards

Note:

Compatibility: There are some limitations related to the MasterTool IEC XE software version acquired, therefore it is recommended the user to see MasterTool IEC XE - MU299609 User Manual to verify specifically the desired applications compatibility with the programmer version.

Purchase Data

Integrant Items

The product package has the following items:

- Harkey with license (must be connected to an USB port of computer with Mastertool IEC XE).
- Library can be downloaded from ***
- ***???

Product Code

The following code must be used to purchase the product:

Code	Description
***	Function Block AGA Report No. 3 – 1992

Table 2-2. Product Code

Related Products

The following products must be purchased separately when necessary:

Code	Description
MT8500	MasterTool IEC XE
NX3010, NX3020 or NX3030	Nexto Series CPU

Table 2-3. Related Products

Notes:

MT8500: MaterTool IEC XE is available in three different versions: LITE, PROFESSIONAL and ADVANCED. For more details, please check MasterTool IEC XE User Manual - MU299609.

3. Programming

This section describes Library AGA Report No 3 – 1992, and presents some application examples.

This library contains the following components:

- The main function block that calculates mass and volume flows:
  - AGA3_Flow
- An additional function block that calculates totalized mass and volume flows:
  - AGA3_Totalizer
- Auxiliary data structures used by some inputs and outputs of function block AGA3_Flow:
  - AGA3_Diameter: features of bore and tube diameter (diameter, coefficient of thermal expansion, reference temperature).
  - AGA3_Warning: warning bits
  - AGA3_Error: error bits
- Units conversion functions:
  - AGA3_ConvertCoefExpansion: convert units of coefficient of thermal expansion
  - AGA3_ConvertDensity: convert units of density
  - AGA3_ConvertDiffPressure: convert units of differential pressure
  - AGA3_ConvertLength: convert units of length (diameter)
  - AGA3_ConvertMass: convert units of mass
  - AGA3_ConvertMassFlow: convert units of mass flow
  - AGA3_ConvertPressure: convert units of pressure
  - AGA3_ConvertTemperature: convert units of temperature
  - AGA3_ConvertViscosity: convert units of viscosity
  - AGA3_ConvertVolume: convert units of volume
  - AGA3_ConvertVolumeFlow: convert units of volume flow
- Enumerations with the allowed units, used in the units conversion functions:
  - AGA3_UnitsCoefExpansion: supported units of coefficient of thermal expansion
  - AGA3_UnitsDensity: supported units of density
  - AGA3_UnitsDiffPressure: supported units of differential pressure
  - AGA3_UnitsLength: supported units of length (diameter)
  - AGA3_UnitsMass: supported units of mass
  - AGA3_UnitsMassFlow: supported units of mass flow
  - AGA3_UnitsPressure: supported units of pressure
  - AGA3_UnitsTemperature: supported units of temperature
  - AGA3_UnitsViscosity: supported units of viscosity
  - AGA3_UnitsVolume: supported units of volume
  - AGA3_UnitsVolumeFlow: supported units of volume flow

The following figure shows these components in the Library Manager tab of Masertool IEC XE.

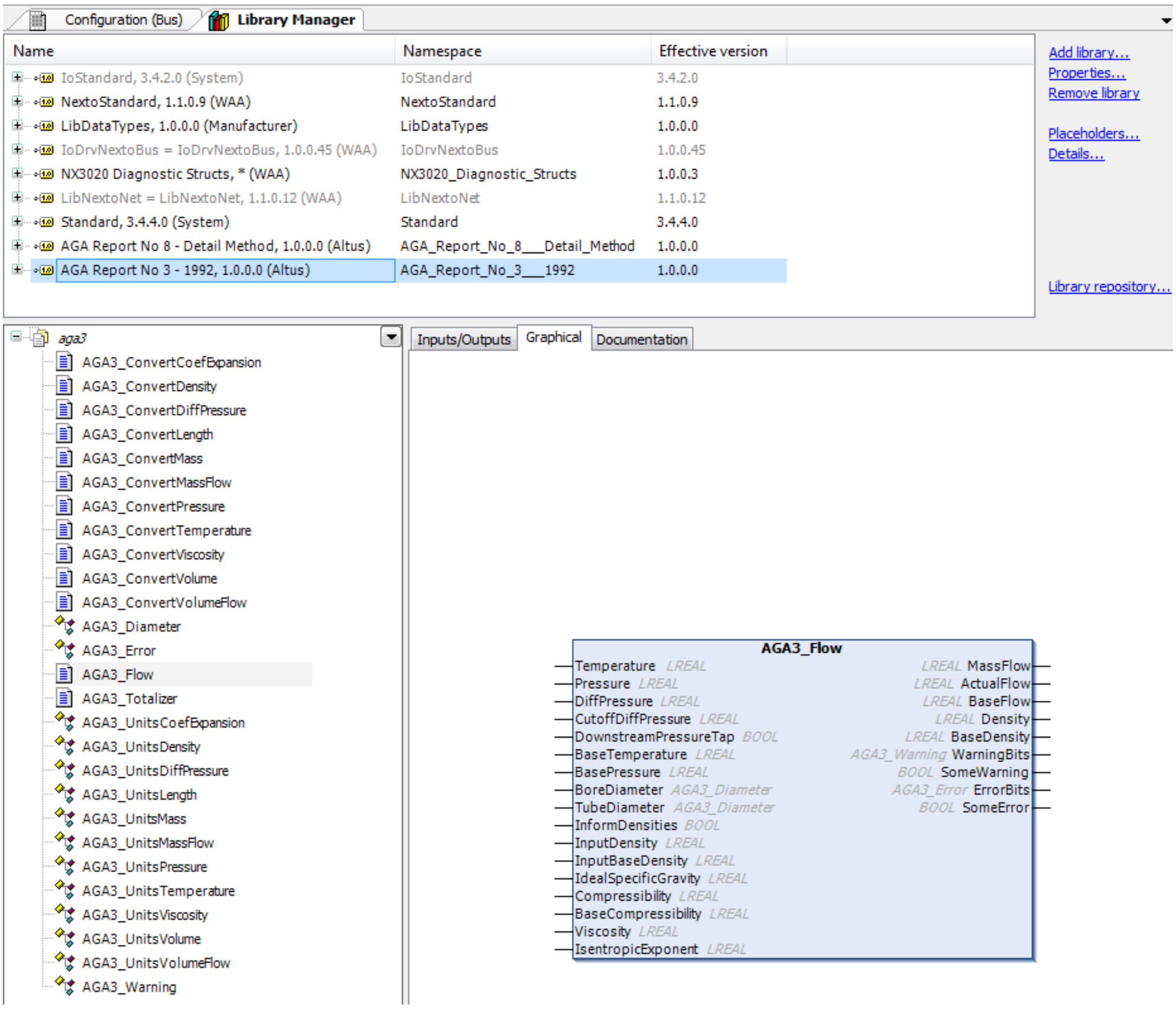

Figure 3-1. Components of Library AGA Report No 3 – 1992

> NOTE:
> Refer *Mastertool IEC XE User Manual* to learn how libraries can be installed in the library repository, and how they can be added in application projects.

AGA3_Flow Function Block

This function block is the main component of the library. The following figure shows inputs (left side of the box) and outputs (right side of the box) of this function block.

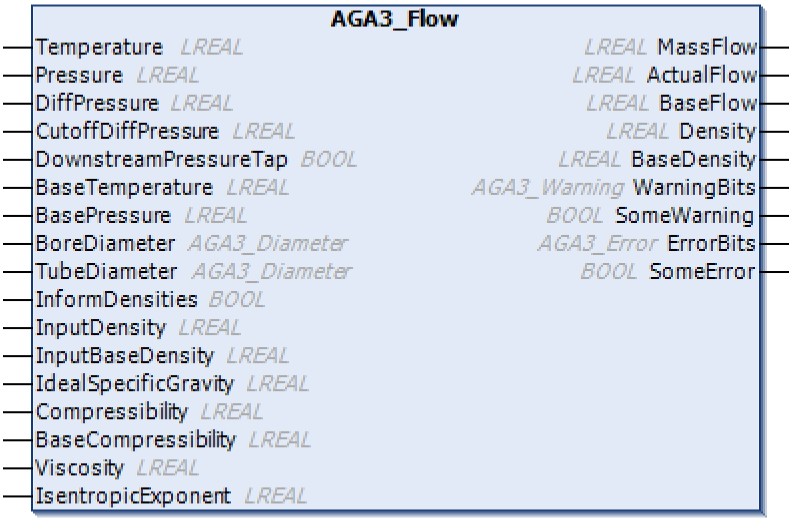

Figure 3-2. Graphical representation of function block AGA3_Flow

The types for each input and output are also shown in italic.

Some of these types are standard:

- LREAL: 64-bit floating point
- BOOL: binary logic bit

The other types are defined in data structures (AGA3_Diameter, AGA3_Warning, AGA3_Error). These data structures will be described in the next sub-section.

The following table describes the inputs with more details.

Input parameters	Type	Description
Temperature	LREAL	Temperature of actual condition in degrees Celsius (ºC).
Pressure	LREAL	Absolute pressure of actual condition in bar (bar)
DiffPressure	LREAL	Differential pressure in millibar (millibar)
CutoffDiffPressure	LREAL	Cutoff differential pressure in millibar (millibar). If DiffPressure is smaller than CutoffDiffPressure, the calculated outputs (mass and volume flows, and densities) are set to zero.
DownstreamPressureTap	BOOL	If TRUE, the function block assumes that the pressure tap is installed downstream. If FALSE, it assumes the pressure tap is installed upstream.
BaseTemperature	LREAL	Temperature of base condition in degrees Celsius (ºC). See **Note 1** for additional details.
BasePressure	LREAL	Absolute upstream pressure of base condition in bar (bar). See **Note 1** for additional details.
BoreDiameter	AGA3_Diameter	This data structure describes bore properties (diameter, reference temperature and coefficient of thermal expansion).
TubeDiameter	AGA3_Diameter	This data structure describes tube properties (diameter, reference temperature and coefficient of thermal

		expansion).
InformDensities	BOOL	This boolean flag informs that densities at actual condition and base condition are taken as inputs, rather than calculated by function block. These inputs are InputDensity (actual condition) and InputBaseDensity (base condition). See **Note 2** for additional details.
InputDensity	LREAL	This parameter informs the density at actual condition (kg/m^3), when parameter InformDensities = TRUE. See **Note 2** for additional details.
InputBaseDensity	LREAL	This parameter informs the density at base condition (kg/m^3), when parameter InformDensities = TRUE. See **Note 2** for additional details.
IdealSpecificGravity	LREAL	This parameter informs the natural gas ideal gas specific gravity (also known as ideal gas relative density), when parameter InformDensities = FALSE. See **Note 2** for additional details. For users that wants to use real gas specific gravity instead of ideal gas specific gravity, see chapter **Real and Ideal Specific Gravity**.
Compressibility	LREAL	This parameter informs the natural gas compressibility at actual condition, when parameter InformDensities = FALSE. See **Note 2** for additional details.
BaseCompressibility	LREAL	This parameter informs the natural gas compressibility at base condition, when parameter InformDensities = FALSE. See **Note 2** for additional details.
Viscosity	LREAL	This parameter informs the fluid viscosity. See **Note 3** for additional details.
IsentropicExponent	LREAL	This parameter informs the isentropic exponent. See **Note 4** for additional details.

Table 3-1. Input Parameters

Note 1:

Temperature and absolute upstream pressure of base condition may change in different applications (e.g.: customers or countries). For instance, in United States of America the base temperature is 60 ^{o}F (15.555 ^{o}C) and base pressure is 14.73 psia (1.0156 bar). However, in other countries, different base temperature and base pressure may be used.

Note 2:

For natural gas, there are two alternatives for calculating densities at actual and base conditions:

- They can be informed through input parameters (InputDensity and InputBaseDensity), with InformDensities = TRUE. In this situation, typically the AGA-8 Detailed Method is employed to calculate the densities.
- They can be calculated by AGA3_Flow function block, using inputs IdealSpecificGravity, Compressibility and BaseCompressibility, with InformDensities = FALSE. In this situation, typically the AGA-8 Gross or Detailed Method is employed to calculate the compressibilities.

For other fluids, the user must inform densities through input parameters (InputDensity and InputBaseDensity), with InformDensities = TRUE.

Note 3:

For natural gas, the AGA-3 standard recommends a constant value of 0.010268 cP for the viscosity. However, the user can set a different value if required.

For other fluids, the user must set an appropriate value for the viscosity.

Note 4:

For natural gas, the AGA-3 standard recommends a constant value of 1.3 for the isentropic exponent. However, the user can set a different value if required.

For other fluids, the user must set an appropriate value.

If the fluid is incompressible, the constant value -1 must be used. In reality, any value smaller or equal to zero will denote an incompressible fluid.

The following table describes the outputs with more details.

Output parameters	Type	Description
MassFlow	LREAL	Mass flow (kg/hr).
ActualFlow	LREAL	Volume flow at actual condition (m^3/hr).
BaseFlow	LREAL	Volume flow at base condition (m^3/hr).
Density	LREAL	Density at actual condition (kg/m^3).
BaseDensity	LREAL	Density at base condition (kg/m^3).
WarningBits	AGA3_Warning	Set of bits indicating warnings. Warnings don't prevent execution of function block, but indicate lack of precision in calculation.
SomeWarning	BOOL	Indicates that at least one warning bit is set in output parameter WarningBits. This may indicate some lack of precision in calculations.
ErrorBits	AGA3_Error	Set of bits indicating errors. Errors prevent the complete execution of function block.
SomeError	BOOL	Indicates that at least one error bit is set in output parameter ErrorBits. Therefore, the function block was not completely executed.

Table 3-2. Output Parameters

Data Structures Defined for AGA3_Flow Function Block

Three data structures (customized types) are defined for usage in function block AGA3_Flow.

AGA3_Diameter

This type is used for input parameters BoreDiameter and TubeDiameter. The following figure details the data structure.

STRUCT AGA3_Diameter

Name	Type	Comment
Diameter	LREAL	average diameter (mm)
RefTemp	LREAL	reference temperature during diameter measurement (degrees Celsius)
Alfa	LREAL	coefficient of thermal expansion (mm/(mm * oC))

Figure 3-3. Data Structure AGA3_Diameter

The data structure contains 3 fields related to the diameter:

- Diameter: informs the measured diameter (mm)
- RefTemp: informs the temperature (ºC) used to measure the diameter
- Alfa: informs the coefficient of thermal expansion (mm/(mm * ºC))

AGA3_Warning

This type is used for output parameter WarningBits. The data structure contains several warning bits (each warning bit has type BOOL).

Note that warning bits don't prevent the execution of function block. However, they indicate lack of precision in calculations. Ideally, all they should be cleared. If any warning bit is set in this data structure, the function block AGA3_Flow will also activate the SomeWarning output.

The following figure details the data structure.

STRUCT AGA3_Warning

Name	Type	Comment
CutoffDiffPress	BOOL	Differential pressure bellow the cutoff limit
LowReynolds	BOOL	Reynolds number bellow 4000 during calculation of Coefficient of Discharge

Figure 3-4. Data Structure AGA3_Warning

CutoffDiffPress indicates that input DiffPressure is smaller than input CutoffDiffPressure. In this case, the flow and density outputs (MassFlow, ActualFlow, BaseFlow, Density and BaseDensity) are set to zero.

LowReynolds indicates that the pipe Reynolds number, computed internally by the function block, is smaller than 4000. Therefore, the uncertanties may be bigger than those expected by AGA-3 standard.

AGA3_Error

This type is used for output parameter ErrorBits. The data structure contains several error bits (each error bit has type BOOL).

Note that error bits indicate that function block was not completely executed. If any error bit is set in this data structure, the function block AGA3 will also activate the SomeError output.

The following figure details the data structure.

STRUCT AGA3_Error

Name	Type	Comment
ExceededIterCD	BOOL	Maximum number of iterations reached in calculation of Coefficient of Dischar...
SmallBore	BOOL	Bore diameter smaller than 11.4 mm
SmallTube	BOOL	Tube diameter smaller than 50 mm
BadBeta	BOOL	Beta factor out of range 0.1 ... 0.75
LowTemperature	BOOL	Temperature <= -273.15
LowBaseTemperature	BOOL	BaseTemperature <= -273.15
LowBoreRefTemperature	BOOL	Reference temperature for bore diameter <= -273.15
LowTubeRefTemperature	BOOL	Reference temperature for tube diameter <= -273.15
LowBoreAlfa	BOOL	Alfa for bore diameter <= 0
LowTubeAlfa	BOOL	Alfa for tube diameter <= 0
LowPressure	BOOL	Pressure <= 0
LowBasePressure	BOOL	BasePressure <= 0
LowCutoffDiffPressure	BOOL	CutoffDiffPressure < 0
LowInputDensity	BOOL	InputDensity <= 0 and InformDensities
LowInputBaseDensity	BOOL	InputBaseDensity <= 0 and InformDensities
LowSpecificGravity	BOOL	SpecificGravity <= 0 and not InformDensities
LowCompressibility	BOOL	Compressibility <=0 and not InformDensities
LowBaseCompressibility	BOOL	BaseCompressibility <=0 and not InformDensities
LowViscosity	BOOL	Viscosity <= 0

Figure 3-5. Data Structure AGA3_Error

ExceededIterCD indicates that the iterative algorithm used in calculation of coefficient of discharged was aborted after 20 iterations. In normal situations, less than 10 iterations should be enough.

SmallBore indicates a bore diameter smaller than 11.4 mm at reference temperature (bellow the range accepted by AGA-3 standard).

SmallTube indicates a tube diameter smaller than 50 mm at reference temperature (bellow the range accepted by AGA-3 standard).

BadBeta indicate a that rate between bore and tube diameter is out of the range accepted by AGA-3 standard (0.1 to 0.75).

LowTemperature indicates a temperature at actual condition smaller or equal than 0 ºK (-273.15 ºC).

LowBaseTemperature indicates a temperature at base condition smaller or equal than 0 ºK (-273.15 ºC).

LowBoreRefTemperature indicates a reference temperature for bore diameter measurement smaller or equal than 0 ºK (-273.15 ºC).

LowTubeRefTemperature indicates a reference temperature for tube diameter measurement smaller or equal than 0 ºK (-273.15 ºC).

LowBoreAlfa indicates a coefficient of thermal expansion for bore smaller or equal to zero.

LowTubeAlfa indicates a coefficient of thermal expansion for tube smaller or equal to zero.

LowPressure indicates an absolute pressure at actual condition smaller or equal to zero.

LowBasePressure indicates an absolute upstream pressure at base condition smaller or equal to zero.

LowCutoffDiffPressure indicates that cutoff differential pressure is smaller or equal to zero.

LowInputDensity indicates that informed density for actual condition is smaller or equal to zero. This error is not set when input parameter InformDensities is FALSE.

LowInputBaseDensity indicates that informed density for base condition is smaller or equal to zero. This error is not set when input parameter InformDensities is FALSE.

LowSpecificGravity indicates that ideal gas specific gravity is smaller or equal to zero. This error is not set when input parameter InformDensities is TRUE.

LowCompressibility indicates that compressibility at actual condition is smaller or equal to zero. This error is not set when input parameter InformDensities is TRUE.

LowBaseCompressibility indicates that compressibility at base condition is smaller or equal to zero. This error is not set when input parameter InformDensities is TRUE.

LowViscosity indicates that viscosity is smaller or equal to zero.

Converting Units of Coefficient of Thermal Expansion

The library contains a function to convert coefficient of thermal expansion (AGA3_ConvertCoefExpansion) and an enumeration that lists the supported units (AGA3_UnitsCoefExpansion).

The following figure shows enumeration AGA3_UnitsCoefExpansion.

ENUM AGA3_UnitsCoefExpansion

Name	Type	Comment
AGA3_COEF_EXP_IN_F	INT	Coefficient of expansion in in/in-oF (US)
AGA3_COEF_EXP_FT_F	INT	Coefficient of expansion in ft/ft-oF (IP)
AGA3_COEF_EXP_MM_C	INT	Coefficient of expansion in mm/mm-oC (metric)
AGA3_COEF_EXP_M_K	INT	

Figure 3-6. Enumeration AGA3_UnitsCoefExpansion

The following figure shows the graphic representation of function AGA3_CoefExpansion.

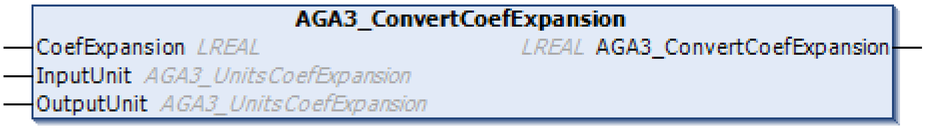

Figure 3-7. Graphical representation of function AGA3_ConvertCoefExpansion

The input parameter CoefExpansion (type LREAL) contains the coefficient of thermal expansion to be converted.

The input parameter InputUnit (type AGA3_UnitsCoefExpansion) defines the unit of coefficient of thermal expansion to be converted.

The input parameter OutputUnit (type AGA3_UnitsCoefExpansion) defines the unit of converted coefficient of thermal expansion.

The output parameter AGA3_ConvertCoefExpansion (type LREAL) returns the converted coefficient of thermal expansion.

Converting Units of Density

The library contains a function to convert density (AGA3_ConvertDensity) and an enumeration that lists the supported units (AGA3_UnitsDensity).

The following figure shows enumeration AGA3_UnitsDensity.

ENUM AGA3_UnitsDensity

Name	Type	Comment
AGA3_DENSITY_KG_M3	INT	Density in kg/m^3 (metric, SI)
AGA3_DENSITY_LBM_FT3	INT	

Figure 3-8. Enumeration AGA3_UnitsDensity

The following figure shows the graphic representation of function AGA3_ConvertDensity.

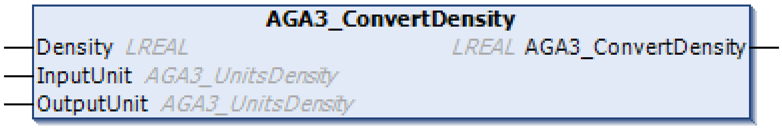

Figure 3-9. Graphical representation of function AGA3_ConvertDensity

The input parameter Density (type LREAL) contains the density to be converted.

The input parameter InputUnit (type AGA3_UnitsDensity) defines the unit of density to be converted.

The input parameter OutputUnit (type AGA3_UnitsDensity) defines the unit of converted density.

The output parameter AGA3_ConvertDensity (type LREAL) returns the converted density.

Converting Units of Differential Pressure

The library contains a function to convert differential pressure (AGA3_ConvertDiffPressure) and an enumeration that lists the supported units (AGA3_UnitsDiffPressure).

The following figure shows enumeration AGA3_UnitsDiffPressure.

ENUM AGA3_UnitsDiffPressure

Name	Type	Comment
AGA3_DIFF_PRESSURE_PA	INT	Differential pressure in Pascal (SI)
AGA3_DIFF_PRESSURE_MBAR	INT	Differential pressure in millibar (Metric)
AGA3_DIFF_PRESSURE_INH2O	INT	

Figure 3-10. Enumeration AGA3_UnitsDiffPressure

The following figure shows the graphic representation of function AGA3_ConvertDiffPressure.

Figure 3-11. Graphical representation of function AGA3_ConvertDiffPressure

The input parameter DiffPressure (type LREAL) contains the differential pressure to be converted.

The input parameter InputUnit (type AGA3_UnitsDiffPressure) defines the unit of differential pressure to be converted.

The input parameter OutputUnit (type AGA3_UnitsDiffPressure) defines the unit of converted differential pressure.

The output parameter AGA3_ConvertDiffPressure (type LREAL) returns the converted differential pressure.

Converting Units of Length

The library contains a function to convert length (AGA3_ConvertLength) and an enumeration that lists the supported units (AGA3_UnitsLength).

The following figure shows enumeration AGA3_UnitsLength.

ENUM AGA3_UnitsLength

Name	Type	Comment
AGA3_LENGTH_MM	INT	Length in mm (metric)
AGA3_LENGTH_M	INT	Length in m (SI)
AGA3_LENGTH_IN	INT	Length in inches (US)
AGA3_LENGTH_FT	INT	

Figure 3-12. Enumeration AGA3_UnitsLength

The following figure shows the graphic representation of function AGA3_ConvertLength.

Figure 3-13. Graphical representation of function AGA3_ConvertLength

The input parameter Length (type LREAL) contains the length to be converted.

The input parameter InputUnit (type AGA3_UnitsLength) defines the unit of length to be converted.

The input parameter OutputUnit (type AGA3_UnitsLength) defines the unit of converted length.

The output parameter AGA3_ConvertLength (type LREAL) returns the converted length.

Converting Units of Mass

The library contains a function to convert mass (AGA3_ConvertMass) and an enumeration that lists the supported units (AGA3_UnitsMass).

The following figure shows enumeration AGA3_UnitsMass.

ENUM AGA3_UnitsMass

Name	Type	Comment
AGA3_MASS_KG	INT	Mass in kg (metric, SI)
AGA3_MASS_LBM	INT	

Figure 3-14. Enumeration AGA3_UnitsMass

The following figure shows the graphic representation of function AGA3_ConvertMass.

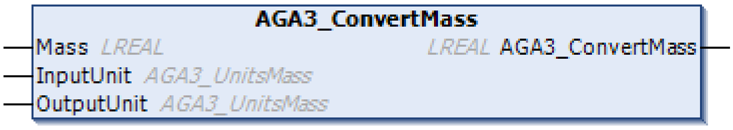

Figure 3-15. Graphical representation of function AGA3_ConvertMass

The input parameter Mass (type LREAL) contains the mass to be converted.

The input parameter InputUnit (type AGA3_UnitsMass) defines the unit of mass to be converted.

The input parameter OutputUnit (type AGA3_UnitsMass) defines the unit of converted mass.

The output parameter AGA3_ConvertMass (type LREAL) returns the converted mass.

Converting Units of Mass Flow

The library contains a function to convert mass flow (AGA3_ConvertMassFlow) and an enumeration that lists the supported units (AGA3_UnitsMassFlow).

The following figure shows enumeration AGA3_UnitsMassFlow.

ENUM AGA3_UnitsMassFlow

Name	Type	Comment
AGA3_MASSFLOW_KGHR	INT	Mass Flow in kg/hr (metric)
AGA3_MASSFLOW_KGS	INT	Mass Flow in kg/s (SI)
AGA3_MASSFLOW_LBMHR	INT	

Figure 3-16. Enumeration AGA3_UnitsMassFlow

The following figure shows the graphic representation of function AGA3_ConvertMassFlow.

Figure 3-17. Graphical representation of function AGA3_ConvertMassFlow

The input parameter MassFlow (type LREAL) contains the mass flow to be converted.

The input parameter InputUnit (type AGA3_UnitsMass) defines the unit of mass flow to be converted.

The input parameter OutputUnit (type AGA3_UnitsMass) defines the unit of converted mass flow.

The output parameter AGA3_ConvertMassFlow (type LREAL) returns the converted mass flow.

Converting Units of Pressure

The library contains a function to convert pressure (AGA3_ConvertPressure) and an enumeration that lists the supported units (AGA3_UnitsPressure).

The following figure shows enumeration AGA3_UnitsPressure.

ENUM AGA3_UnitsPressure

Name	Type	Comment
AGA3_PRESSURE_PA	INT	Pressure in Pascal (SI)
AGA3_PRESSURE_PSIA	INT	Pressure in psia (US, IP)
AGA3_PRESSURE_BAR	INT	

Figure 3-18. Enumeration AGA3_UnitsPressure

The following figure shows the graphic representation of function AGA3_ConvertPressure.

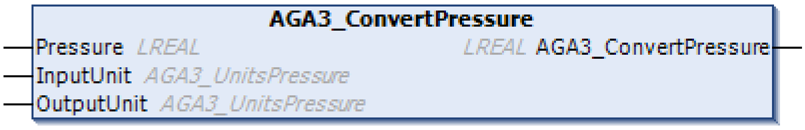

Figure 3-19. Graphical representation of function AGA3_ConvertPressure

The input parameter Pressure (type LREAL) contains the pressure to be converted.

The input parameter InputUnit (type AGA3_UnitsPressure) defines the unit of pressure to be converted.

The input parameter OutputUnit (type AGA3_UnitsPressure) defines the unit of converted pressure.

The output parameter AGA3_ConvertPressure (type LREAL) returns the converted pressure.

Converting Units of Temperature

The library contains a function to convert temperature (AGA3_ConvertTemperature) and an enumeration that lists the supported units (AGA3_UnitsTemperature).

The following figure shows enumeration AGA3_UnitsTemperature.

ENUM AGA3_UnitsTemperature

Name	Type	Comment
AGA3_TEMPERATURE_KELVIN	INT	Temperature in Kelvin (SI)
AGA3_TEMPERATURE_CELSIUS	INT	Temperature in Celsius (metric)
AGA3_TEMPERATURE_FARENHEIT	INT	

Figure 3-20. Enumeration AGA3_UnitsTemperature

The following figure shows the graphic representation of function AGA3_ConvertTemperature.

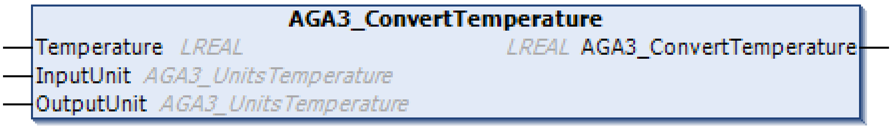

Figure 3-21. Graphical representation of function AGA3_ConvertTemperature

The input parameter Temperature (type LREAL) contains the temperature to be converted.

The input parameter InputUnit (type AGA3_UnitsTemperature) defines the unit of temperature to be converted.

The input parameter OutputUnit (type AGA3_UnitsTemperature) defines the unit of converted temperature.

The output parameter AGA3_ConvertTemperature (type LREAL) returns the converted temperature.

Converting Units of Viscosity

The library contains a function to convert viscosity (AGA3_ConvertViscosity) and an enumeration that lists the supported units (AGA3_UnitsViscosity).

The following figure shows enumeration AGA3_UnitsViscosity.

ENUM AGA3_UnitsViscosity

Name	Type	Comment
AGA3_VISCOSITY_CP	INT	Viscosity in centipoise (US, metric)
AGA3_PRESSURE_LBM_FTS	INT	Viscosity in lbm/ft-s (IP)
AGA3_PRESSURE_PA_S	INT	

Figure 3-22. Enumeration AGA3_UnitsViscosity

The following figure shows the graphic representation of function AGA3_ConvertViscosity.

Figure 3-23. Graphical representation of function AGA3_ConvertViscosity

The input parameter Viscosity (type LREAL) contains the viscosity to be converted.

The input parameter InputUnit (type AGA3_UnitsViscosity) defines the unit of viscosity to be converted.

The input parameter OutputUnit (type AGA3_UnitsViscosity) defines the unit of converted viscosity.

The output parameter AGA3_ConvertViscosity (type LREAL) returns the converted viscosity.

Converting Units of Volume

The library contains a function to convert volume (AGA3_ConvertVolume) and an enumeration that lists the supported units (AGA3_UnitsVolume).

The following figure shows enumeration AGA3_UnitsVolume.

ENUM AGA3_UnitsVolume

Name	Type	Comment
AGA3_VOLUME_M3	INT	Volume in m^3 (metric, SI)
AGA3_VOLUME_FT3	INT	

Figure 3-24. Enumeration AGA3_UnitsVolume

The following figure shows the graphic representation of function AGA3_ConvertVolume.

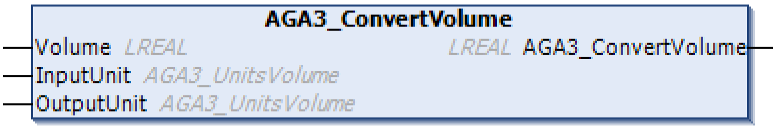

Figure 3-25. Graphical representation of function AGA3_ConvertVolume

The input parameter Volume (type LREAL) contains the volume to be converted.

The input parameter InputUnit (type AGA3_UnitsVolume) defines the unit of volume to be converted.

The input parameter OutputUnit (type AGA3_UnitsVolume) defines the unit of converted volume.

The output parameter AGA3_ConvertVolume (type LREAL) returns the converted volume.

Converting Units of Volume Flow

The library contains a function to convert volume flow (AGA3_ConvertVolumeFlow) and an enumeration that lists the supported units (AGA3_UnitsVolumeFlow).

The following figure shows enumeration AGA3_UnitsVolumeFlow.

ENUM AGA3_UnitsVolumeFlow

Name	Type	Comment
AGA3_VOLUMEFLOW_M3HR	INT	Volume Flow in m^3/hr (metric)
AGA3_VOLUMEFLOW_M3S	INT	Volume Flow in m^3/s (SI)
AGA3_VOLUMEFLOW_FT3HR	INT	

Figure 3-26. Enumeration AGA3_UnitsVolumeFlow

The following figure shows the graphic representation of function AGA3_ConvertVolumeFlow.

Figure 3-27. Graphical representation of function AGA3_ConvertVolumeFlow

The input parameter VolumeFlow (type LREAL) contains the volume flow to be converted.

The input parameter InputUnit (type AGA3_UnitsVolumeFlow) defines the unit of volume flow to be converted.

The input parameter OutputUnit (type AGA3_UnitsVolumeFlow) defines the unit of converted volume flow.

The output parameter AGA3_ConvertVolumeFlow (type LREAL) returns the converted volume flow.

Application Examples

In this section, three application examples will be presented, exploring the usage of function blocks AGA3_Flow and AGA3_Totalizer, and of unit conversion functions.

Example 1 shows an application for natural gas where densities at actual and base conditions are informed as inputs for function block AGA3_Flow. In this example, the densities are calculated previously using AGA-8 Detailed Method.

Example 1 also shows examples of utilization of AGA3_Totalizer function block, for mass and base volume totalization.

Example 1 works with metric units, so there is no need to use unit conversion functions.

Example 2 shows an application for natural gas where densities at actual and base conditions are calculated internally by function block AGA3_Flow. In this example, the user must provide the following input information:

- Ideal gas specific gravity
- Compressibilities at actual and base conditions. These compressibilities could be calculated, for instance, using AGA-8 gross method.

Example 2 also shows examples of utilization of AGA3_Totalizer function block, for mass and base volume totalization.

Example 2 works with US units, so it can be used as an example of usage of unit conversion functions between US and metric units.

Example 3 shows an application for an incrompressible fluid, like a liquid fluid.

Example 3 works with metric units, so there is no need to use unit conversion functions.

Application Example 1 - Natural Gas with Informed Densities

The following example, developed in ST language, contains a single instance of function block AGA3_Flow, and two instances of function block AGA3_Totalizer.

> NOTE:
> Several instances of function blocks AGA3_Flow and AGA3_Totalizer can exist in the same application.

In this example, the fluid is assumed to be natural gas, and the densities will be informed using input parameters InputDensity and InputBaseDensity.

The application example also calls an instance of function block AGA8Detail (AGA-8 Detailed Method), used for calculating the densities that will be informed as inputs to function block AGA3_Flow.

> NOTE:
> Function block AGA8Detail is another product, and has its own user manual. This user manual doesn’t intend to describe the AGA8Detail function block.

The next figure shows the variables declared in the application.

Configuration (Bus) | MainPrg

```
1  PROGRAM MainPrg
2  VAR
3      AGA8_DensityCalc : AGA8Detail;          // Instance of function block AGA8Detail, for calculating densities
4      AGA3_FlowCalc : AGA3_Flow;              // Instance of function block AGA3_Flow, for flow calculation
5      AGA3_TotalMass : AGA3_Totalizer;        // Instance of function block AGA3_Totalizer, for totalization of mass flow
6      AGA3_TotalBaseVolume : AGA3_Totalizer;  // Instance of function block AGA3_Totalizer, for totalization of volume flow at base conditions
7
8      FirstCycle : BOOL := TRUE;              // Flag for indicating the first cycle of application example
9
10     // Common inputs for AGA3_Flow and AGA8Detail function blocks
11     Temperature_C : LREAL;                  // Temperature of actual condition in Celsius
12     Pressure_bar : LREAL;                   // Absolute pressure of actual condition in bar
13     BaseTemperature_C : LREAL;              // Temperature of base condition in Celsius
14     BasePressure_bar : LREAL;               // Absolute upstream pressure of base condition in bar
15
16     // Specific inputs for AGA8Detail function block
17     GasComposition : AGA8Detail_GasComposition; // Mole % of gas components
18
19     // Outputs calculated by AGA8Detail function block (some taken as input by AGA3_Flow function block)
20     Density_kgm3 : LREAL;                   // Density at actual condition in kg/m3
21     BaseDensity_kgm3 : LREAL;               // Density at base condition in kg/m3
22     Uncertainty : LREAL;                    // Percentage of uncertainty according figure 1 of AGA Report No. 8
23     TotalPercentGas : LREAL;                // Total percentage of gas composition (should be 100%)
24     AGA8_WarningBits : AGA8Detail_Warning;  // Input parameters out of normal ranges and other real time warnings
25     AGA8_SomeWarning : BOOL;                // At least one warning bit is set in structure WarningBits
26     AGA8_ErrorBits : AGA8Detail_Error;      // Input parameters out of expanded ranges
27     AGA8_SomeError : BOOL;                  // At least one error bit is set in structure ErrorBits, FB not being executed
28
29     // Specific inputs for AGA3_Flow function block
30     DiffPressure_mbar : LREAL;              // Differential pressure in millibar
31     BoreDiameter_Metric : AGA3_Diameter;    // Bore diameter and complementary information in metric units
32     TubeDiameter_Metric : AGA3_Diameter;    // Tube diameter and complementary information in metric units
33
34     // Outputs calculated by AGA3_Flow function block (some taken as input by AGA3_Totalizer function blocks)
35     MassFlow_kghr : LREAL;                  // Mass flow in kg/hr
36     ActualFlow_m3hr : LREAL;                // Actual volume flow in m3/hr
37     BaseFlow_m3hr : LREAL;                  // Base volume flow in m3/hr
38     AGA3_WarningBits : AGA3_Warning;        // Warning bits
39     AGA3_SomeWarning : BOOL;                // Indicates at least one warning bit is set
40     AGA3_ErrorBits : AGA3_Error;            // Error bits
41     AGA3_SomeError : BOOL;                  // Indicates at leas on error bit is set
42
43     // Specific inputs for AGA3_Totalizer function blocks
44     ResetTotals : BOOL;                     // Command for resetting totals
45     PresetTotals : BOOL;                    // Command for presetting totals
46     PresetMass_kg : LREAL;                  // Preset value for mass in kg
47     PresetBaseVolume_m3 : LREAL;            // Preset value for volume at base condition in m3
48
49     // Outputs calculated by AGA3_Totalizer function blocks
50     TotalMass_kg : LREAL;                   // Total mass in kg
51     TotalBaseVolume_m3 : LREAL;             // Total volume at base condition in m3
52 END_VAR
```

Figure 3-28. Application Example 1 - Variables

In line 2, an instance of AGA8Detail function block is declared.

In line 3, an instance of AGA3_Flow function block is declared.

In lines 4 and 5, two instances of AGA3_Totalizer function block are declared, one for totalizing mass, other for totalizing volume at base condition.

In line 8, boolean flag FirstCycle is declared to control some initializations.

Between lines 11 and 14, some input parameters common for AGA3_Flow and AGA8Detail are declared.

In line 11, the actual temperature in Celsius (Temp_C) is declared.

In line 12, the actual absolute pressure in bar (Presure_bar) is declared.

In line 13, the base temperature in Celsius (BaseTemp_C) is declared.

In line 14, the base pressure in bar (BaseTemp_bar) is declared.

In line 17, the gas composition for AGA8Detail instance is declared.

Between lines 20 e 27, output parameters of AGA8Detail are declared. Some of these output parameters will be used as input parameters for AGA3_Flow.

In line 20, the density at actual condition (Density_kgm3) is declared. This is also an input for AGA3_Flow.

In line 21, the density at base condition (BaseDensity_kgm3) is declared. This is also an input for AGA3_Flow.

Between lines 22 and 27, some outputs of AGA8Detail are declared (not scope of this manual).

Between lines 30 and 32, some inputs for AGA3_Flow are declared.

In line 30, differential pressure in millibar is declared (DiffPressure_mbar).

In line 31, bore diameter features are declared (BoreDiameter_Metric). It consists of three fields: diameter, reference temperature and coefficient of thermal expansion.

In line 32, tube diameter features are declared (TubeDiameter_Metric). It consists of three fields: diameter, reference temperature and coefficient of thermal expansion.

Between lines 35 and 41, output parameters of AGA3_Flow are declared.

In line 35, the mass flow in kg/hr is declared (MassFlow_kghr).

In line 36, the volume flow at actual condition in m^3/hr is declared (ActualFlow_m3hr).

In line 37, the volume flow at base condition in m^3/hr is declared (BaseFlow_m3hr).

In line 38, the warning bits data structure is declared (AGA3_WarningBits).

In line 39, the logic OR of warning bits is declared (AGA3_SomeWarning).

In line 40, the error bits data structure is declared (AGA3_ErrorBits).

In line 41, the logic OR of error bits is declared (AGA3_SomeError).

Between lines 44 and 47, input parameters of AGA3_Totalizer function blocks are declared.

In line 44, the command for resetting totals is declared (ResetTotals).

In line 45, the command for presetting totals is declared (PresetTotals).

In line 46, the preset value for mass is declared (PresetMass_kg).

In line 47, the preset value for volume at base condition is declared (PresetBaseVolume_m3).

Between lines 50 and 51, outputs of AGA3_Totalizer function blocks are declared.

In line 50, the totalized mass is declared (TotalMass_kg).

In line 51, the totalize volume at base condition is declared (TotalBaseVolume_m3).

The next figure shows the code for this application.

```
// Initializations at first cycle
IF FirstCycle THEN
    FirstCycle := FALSE;

    // initialize inputs in first cycle of this application example
    Temperature_C := 10;
    Pressure_bar := 41.01325;
    DiffPressure_mbar := 100;
    BaseTemperature_C := 15;
    BasePressure_bar := 1.01325;
    BoreDiameter_Metric.Alfa := 1.665E-5;
    BoreDiameter_Metric.RefTemp := 20;
    BoreDiameter_Metric.Diameter := 50.8;
    TubeDiameter_Metric.Alfa := 1.116E-5;
    TubeDiameter_Metric.RefTemp := 20;
    TubeDiameter_Metric.Diameter := 102.26;
    GasComposition.Methane := 90.6724;
    GasComposition.Nitrogen := 3.1284;
    GasComposition.CarbonDioxide := 0.4676;
    GasComposition.Ethane := 4.5279;
    GasComposition.Propane := 0.828;
    GasComposition.Water := 0;
    GasComposition.HydrogenSulphide := 0;
    GasComposition.Hydrogen := 0;
    GasComposition.CarbonMonoxide := 0;
    GasComposition.Oxygen := 0;
    GasComposition.isoButane := 0.1037;
    GasComposition.nButane := 0.1563;
    GasComposition.isoPentane := 0.0321;
    GasComposition.nPentane := 0.0443;
    GasComposition.nHexane := 0.0393;
    GasComposition.nHeptane := 0;
    GasComposition.nOctane := 0;
    GasComposition.nNonane := 0;
    GasComposition.nDecane := 0;
    GasComposition.Helium := 0;
    GasComposition.Argon := 0;
END_IF
```

```
40  // Call of function block instance for densities calculation
41  AGA8_DensityCalc(
42      Temperature:= Temperature_C,
43      Pressure:= Pressure_bar,
44      BaseTemperature:= BaseTemperature_C,
45      BasePressure:= BasePressure_bar,
46      GasComposition:= GasComposition,
47      Density=> Density_kgm3,
48      BaseDensity=> BaseDensity_kgm3,
49      Uncertainty=> Uncertainty,
50      TotalPercentGas=> TotalPercentGas,
51      WarningBits=> AGA8_WarningBits,
52      SomeWarning=> AGA8_SomeWarning,
53      ErrorBits=> AGA8_ErrorBits,
54      SomeError=> AGA8_SomeError);
55
56  // Call of function block instance for flow calculation
57  AGA3_FlowCalc(
58      Temperature:= Temperature_C,
59      Pressure:= Pressure_bar,
60      DiffPressure:= DiffPressure_mbar,
61      CutoffDiffPressure:= 0,
62      DownstreamPressureTap:= FALSE,
63      BaseTemperature:= BaseTemperature_C,
64      BasePressure:= BasePressure_bar,
65      BoreDiameter:= BoreDiameter_Metric,
66      TubeDiameter:= TubeDiameter_Metric,
67      InformDensities := TRUE,
68      InputDensity := Density_kgm3,
69      InputBaseDensity := BaseDensity_kgm3,
70      Viscosity:= 0.010268,
71      IsentropicExponent:= 1.3,
72      MassFlow=> MassFlow_kghr,
73      ActualFlow=> ActualFlow_m3hr,
74      BaseFlow=> BaseFlow_m3hr,
75      WarningBits=> AGA3_WarningBits,
76      SomeWarning=> AGA3_SomeWarning,
77      ErrorBits=> AGA3_ErrorBits,
78      SomeError=> AGA3_SomeError);
79
80  // Call of function block for mass totalization
81  AGA3_TotalMass(
82      InputFlow:= MassFlow_kghr,
83      CycleTime:= 0.1,
84      Reset:= ResetTotals,
85      PresetTotal:= PresetMass_kg,
86      Preset:= PresetTotals,
87      TotalizedFlow=> TotalMass_kg);
88
89  // Call of function block for base volume totalization
90  AGA3_TotalBaseVolume(
91      InputFlow:= BaseFlow_m3hr,
92      CycleTime:= 0.1,
93      Reset:= ResetTotals,
94      PresetTotal:= PresetBaseVolume_m3,
95      Preset:= PresetTotals,
96      TotalizedFlow=> TotalBaseVolume_m3);
97
98  // Cancel reset and preset commands
99  ResetTotals := FALSE;
100 PresetTotals := FALSE;
```

Figure 3-29. Application Example 1 - Code

Between lines 1 and 38, in the first cycle of application, the input variables are initialized. In real applications, the input variables should be read from instruments (temperature transmitters, pressure transmitters, gas chromatographs, etc). Also some fixed input parameters, like bore and tube diameters, are initialized in this code section.

Between lines 41 and 54, the instance of function block AGA8Detail is called.

Between lines 57 and 78, the instance of function block AGA3_Flow is called. Note that:

- InformDensities is TRUE, so the densities (InputDensity and InputBaseDensity) are supplied as inputs. These densisties are outputs of AGA8Detail function block, in this example.
- Input parameters IdealSpecificGravity, Compressibility and BaseCompressibility are supressed. This is possible because they are not used when InformDensities = TRUE.
- The viscosity is informed with the constant value suggested by AGA-3 standard for natural gas (0.010268 cP).
- The isentropic exponent is informed with the constant value suggested by AGA-3 standard for natural gas (1.3).

Between lines 81 and 87, an instance of AGA3_Totalizer is called for calculating the totalized mass.

Between lines 90 and 96, another instance of AGA3_Totalizer is called for calculating the totalized volume at base condition.

In lines 99 and 100, the reset and preset commands, used in AGA3_Totalizer instances, are canceled. They must last only one PLC cycle.

Application Example 2 - Natural Gas with Calculated Densities

The following example, developed in ST language, contains a single instance of function block AGA3_Flow, and two instances of function block AGA3_Totalizer. It also calls functions for converting units, supposing that the US system is being used.

NOTE:
Several instances of function blocks AGA3_Flow and AGA3_Totalizer can exist in the same application.

In this example, the fluid is assumed to be natural gas, and the densities will be calculated using input parameters IdealSpecificGravity (ideal gas relative density), Compressibility and BaseCompressibility. These compressibilities could have been calculated using AGA-8 Gross or Detailed method (the actual method will not be discussed in this example).

The next figure shows the variables declared in the application.

Configuration (Bus) | MainPrg

```
1  PROGRAM MainPrg
2  VAR
3      AGA3_FlowCalc : AGA3_Flow;                // Instance of function block for flow calculation
4      AGA3_Totalizer_Mass : AGA3_Totalizer;      // Instance of function block for mass totalization
5      AGA3_Totalizer_BaseVolume : AGA3_Totalizer; // Instance of function block for base volume totalization
6
7      FirstCycle : BOOL := TRUE;                 // Flag for indicating the first cycle of application example
8
9      // Inputs in US Units
10     Temperature_F : LREAL;                     // Temperature of actual condition in Farenheit
11     Pressure_psia : LREAL;                     // Absolute pressure of actual condition in psia
12     DiffPressure_inW : LREAL;                  // Differential pressure in inches of water at 60 oF
13     BaseTemperature_F : LREAL;                 // Temperature of base condition in Farenheit
14     BasePressure_psia : LREAL;                 // Absolute upstream pressure of base condition in psia
15     BoreDiameter_US : AGA3_Diameter;           // Bore diameter and complementary information in US units
16     TubeDiameter_US : AGA3_Diameter;           // Tube diameter and complementary information in US units
17
18     // Inputs in units required by AGA3_Flow function block
19     Temperature_C : LREAL;                     // Temperature of actual condition in Celsius
20     Pressure_bar : LREAL;                      // Absolute pressure of actual condition in bar
21     DiffPressure_mbar : LREAL;                 // Differential pressure in millibar
22     BaseTemperature_C : LREAL;                 // Temperature of base condition in Celsius
23     BasePressure_bar : LREAL;                  // Absolute upstream pressure of base condition in bar
24     BoreDiameter_Metric : AGA3_Diameter;       // Bore diameter and complementary information in metric units
25     TubeDiameter_Metric : AGA3_Diameter;       // Tube diameter and complementary information in metric units
26     IdealSpecificGravity : LREAL;              // Ideal gas specific gravity or ideal gas relative density
27     Compressibility : LREAL;                   // Compressibility at actual condition
28     BaseCompressibility : LREAL;               // Compressibility at base condition
29
30     // Inputs for AGA3_Totalizer function blocks
31     ResetTotals : BOOL;                        // Command for resetting totalizers
32     PresetTotals : BOOL;                       // Command for presetting totalizers
33     PresetMass_kg : LREAL;                     // Preset value for mass in kg
34     PresetBaseVolume_m3 : LREAL;               // Preset value for volume in m3
35
36     // Outputs in units calculated by AGA3_Flow function block
37     MassFlow_kghr : LREAL;                     // Mass flow in kg/hr
38     ActualFlow_m3hr : LREAL;                   // Actual volume flow in m3/hr
39     BaseFlow_m3hr : LREAL;                     // Base volume flow in m3/hr
40     Density_kgm3 : LREAL;                      // Actual density in kg/m3
41     BaseDensity_kgm3 : LREAL;                  // Base density in kg/m3
42     WarningBits : AGA3_Warning;                // Warning bits
43     SomeWarning : BOOL;                        // Indicates at least one warning bit is set
44     ErrorBits : AGA3_Error;                    // Error bits
45     SomeError : BOOL;                          // Indicates at least on error bit is set
46
47     // Outputs in units calculated by AGA3_Totalizer function blocks in metric system
48     TotalMass_kg : LREAL;                      // Total mass in kg
49     TotalBaseVolume_m3 : LREAL;                // Total base volume in m3
50
51     // Outputs in US Units
52     MassFlow_lbmhr : LREAL;                    // Mass flow in lbm/hr
53     ActualFlow_ft3hr : LREAL;                  // Actual volume flow in ft3/hr
54     BaseFlow_ft3hr : LREAL;                    // Base volume flow in ft3/hr
55     Density_lbmft3 : LREAL;                    // Actual density in lbm/ft3
56     BaseDensity_lbmft3 : LREAL;                // Base density in lbm/m3
57     TotalMass_lbm : LREAL;                     // Total mass in lbm
58     TotalBaseVolume_ft3 : LREAL;               // Total base volume in ft3
59 END_VAR
```

Figure 3-30. Application Example 2 - Variables

In line 3, an instance of AGA3_Flow function block is declared.

In line 4, an instance of AGA3_Totalizer function block is declared, for calculating the totalized mass.

In line 5, an instance of AGA3_Totalizer function block is declared, for calculating the totalized volume at base condition.

In line 7, boolean flag FirstCycle is declared to control some initializations.

Between lines 10 and 16, input variables in US unit system are declared.

In line 10, the actual temperature in Farenheit (Temperature_F) is declared.

In line 11, the actual absolute pressure in psia (Pressure_psia) is declared.

In line 12, the differential pressure in inches of water at 60 ºF (DiffPressure_inW) is declared.

In line 13, the base temperature in Farenheit (BaseTemperature_F) is declared.

In line 14, the base absolute upstream pressure in psia (BasePressure_psia) is declared.

In line 15, the data structure with diameter properties of bore in US units (BoreDiameter_US) is declared.

In line 16, the data structure with diameter properties of tube in US units (TubeDiameter_US) is declared.

Between lines 19 and 28, inputs for AGA3_Flow function block are declared, in the suitable metric units.

In line 19, the actual temperature in Celsius (Temperature_C) is declared.

In line 20, the actual absolute pressure in bar (Pressure_bar) is declared.

In line 21, the differential pressure in millibar (DiffPressure_mbar) is declared.

In line 22, the base temperature in Celsius (BaseTemperature_C) is declared.

In line 23, the base absolute upstream pressure in bar (BasePressure_bar) is declared.

In line 24, the data structure with diameter properties of bore in metric units (BoreDiameter_Metric) is declared.

In line 25, the data structure with diameter properties of tube in metric units (TubeDiameter_ Metric) is declared.

In line 26, the ideal gas specific gravity or ideal gas relative density (IdealSpecificGravity) is declared.

In line 27, the compressibility at actual condition (Compressibility) is declared.

In line 28, the compressibility at base condition (BaseCompressibility) is declared.

Between lines 37 and 45, outputs of AGA3_Flow function block are declared.

In line 37, the mass flow in kg/hr (MassFlow_kghr) is declared.

In line 38, the volume flow at actual conditon in m^3/hr (ActualFlow_m3hr) is declared.

In line 39, the volume flow at base conditon in m^3/hr (BaseFlow_m3hr) is declared.

In line 40, the density at actual condition in kg/m^3 (Density_kgm3) is declared.

In line 41, the density at base condition in kg/m^3 (BaseDensity_kgm3) is declared.

In line 42, the warning bits data structure is declared (WarningBits).

In line 43, the logic OR of warning bits is declared (SomeWarning).

In line 44, the error bits data structure is declared (ErrorBits).

In line 45, the logic OR of error bits is declared (SomeError).

Between lines 48 and 49, outputs of AGA3_Totalizer function blocks are declared.

In line 48, the total mass in kg is declared (TotalMass_kg).

In line 49, the total volume flow at base condition in m^3 is declared (TotalBaseVolume_m3).

Between lines 52 and 58, outputs converted to US units are declared.

In line 52, mass flow in lbm/hr is declared (MassFlow_lbmhr).

In line 53, volume flow at actual condition in ft^3/hr is declared (ActualFlow_ft3hr).

In line 54, volume flow at base condition in ft^3/hr is declared (BaseFlow_ft3hr).

In line 55, density at actual condition in lbm/ft^3 is declared (Density_lbmft3).

In line 56, density at base condition in lbm/ft^3 is declared (BaseDensity_lbmft3).

In line 57, total mass in lbm is declared (TotalMass_lbm).

In line 58, total volume at base condition in ft^3 is declared (TotalBaseVolume_ft3).

The next figure shows the code for this application.

```
IF FirstCycle THEN
    FirstCycle := FALSE;
    // initialize inputs in first cycle of this application example
    Temperature_F := 50;
    DiffPressure_inW := 40.1864652;
    Pressure_psia := 594.84692499;
    BaseTemperature_F := 59;
    BasePressure_psia := 14.69594940;
    BoreDiameter_US.Alfa := 9.250000E-06;
    BoreDiameter_US.RefTemp := 68;
    BoreDiameter_US.Diameter := 2;
    TubeDiameter_US.Alfa := 6.200000E-06;
    TubeDiameter_US.RefTemp := 68;
    TubeDiameter_US.Diameter := 4.02598425;
    IdealSpecificGravity := 0.607527;
    Compressibility := 0.904351;
    BaseCompressibility := 0.997761;
    ResetTotals := FALSE;
    PresetTotals := FALSE;
END_IF

// Units conversion for inputs, from US to metric
Temperature_C := AGA3_ConvertTemperature(Temperature_F, AGA3_TEMPERATURE_FARENHEIT, AGA3_TEMPERATURE_CELSIUS);
DiffPressure_mbar := AGA3_ConvertDiffPressure(DiffPressure_inW, AGA3_DIFF_PRESSURE_INH2O, AGA3_DIFF_PRESSURE_MBAR);
Pressure_bar := AGA3_ConvertPressure(Pressure_psia, AGA3_PRESSURE_PSIA, AGA3_PRESSURE_BAR);
BaseTemperature_C := AGA3_ConvertTemperature(BaseTemperature_F, AGA3_TEMPERATURE_FARENHEIT, AGA3_TEMPERATURE_CELSIUS);
BasePressure_bar := AGA3_ConvertPressure(BasePressure_psia, AGA3_PRESSURE_PSIA, AGA3_PRESSURE_BAR);
BoreDiameter_Metric.Alfa := AGA3_ConvertCoefExpansion(BoreDiameter_US.Alfa, AGA3_COEF_EXP_IN_F, AGA3_COEF_EXP_MM_C);
BoreDiameter_Metric.Diameter := AGA3_ConvertLength(BoreDiameter_US.Diameter, AGA3_LENGTH_IN, AGA3_LENGTH_MM);
BoreDiameter_Metric.RefTemp := AGA3_ConvertTemperature(BoreDiameter_US.RefTemp, AGA3_TEMPERATURE_FARENHEIT, AGA3_TEMPERATURE_CELSIUS);
TubeDiameter_Metric.Alfa := AGA3_ConvertCoefExpansion(TubeDiameter_US.Alfa, AGA3_COEF_EXP_IN_F, AGA3_COEF_EXP_MM_C);
TubeDiameter_Metric.Diameter := AGA3_ConvertLength(TubeDiameter_US.Diameter, AGA3_LENGTH_IN, AGA3_LENGTH_MM);
TubeDiameter_Metric.RefTemp := AGA3_ConvertTemperature(TubeDiameter_US.RefTemp, AGA3_TEMPERATURE_FARENHEIT, AGA3_TEMPERATURE_CELSIUS);

// Call of function block instance for flow calculation
AGA3_FlowCalc(
    Temperature:= Temperature_C,
    Pressure:= Pressure_bar,
    DiffPressure:= DiffPressure_mbar,
    CutoffDiffPressure:= 0,
    DownstreamPressureTap:= FALSE,
    BaseTemperature:= BaseTemperature_C,
    BasePressure:= BasePressure_bar,
    BoreDiameter:= BoreDiameter_Metric,
    TubeDiameter:= TubeDiameter_Metric,
    InformDensities := FALSE,
    IdealSpecificGravity:= IdealSpecificGravity,
    Compressibility:= Compressibility,
    BaseCompressibility:= BaseCompressibility,
    Viscosity:= 0.010268,
    IsentropicExponent:= 1.3,
    MassFlow=> MassFlow_kghr,
    ActualFlow=> ActualFlow_m3hr,
    BaseFlow=> BaseFlow_m3hr,
    Density=> Density_kgm3,
    BaseDensity=> BaseDensity_kgm3,
    WarningBits=> WarningBits,
    SomeWarning=> SomeWarning,
    ErrorBits=> ErrorBits,
    SomeError=> SomeError);

//  Call of function block instance for mass totalization
AGA3_Totalizer_Mass(
    InputFlow:= MassFlow_kghr,
    CycleTime:= 0.1,
    Reset:= ResetTotals,
    PresetTotal:= PresetMass_kg,
    Preset:= PresetTotals,
    TotalizedFlow=> TotalMass_kg);

//  Call of function block instance for base volume totalization
AGA3_Totalizer_BaseVolume(
    InputFlow:= BaseFlow_m3hr,
    CycleTime:= 0.1,
    Reset:= ResetTotals,
    PresetTotal:= PresetBaseVolume_m3,
    Preset:= PresetTotals,
    TotalizedFlow=> TotalBaseVolume_m3);

// Reset commands for totalization after calling function blocks (must last only one PLC cycle)
ResetTotals := FALSE;
PresetTotals := FALSE;

// Units conversion for outputs, from metric to US
MassFlow_lbmhr := AGA3_ConvertMassFlow(MassFlow_kghr, AGA3_MASSFLOW_KGHR, AGA3_MASSFLOW_LBMHR);
ActualFlow_ft3hr := AGA3_ConvertVolumeFlow(ActualFlow_m3hr, AGA3_VOLUMEFLOW_M3HR, AGA3_VOLUMEFLOW_FT3HR);
BaseFlow_ft3hr := AGA3_ConvertVolumeFlow(BaseFlow_m3hr, AGA3_VOLUMEFLOW_M3HR, AGA3_VOLUMEFLOW_FT3HR);
Density_lbmft3 := AGA3_ConvertDensity(Density_kgm3, AGA3_DENSITY_KG_M3, AGA3_DENSITY_LBM_FT3);
BaseDensity_lbmft3 := AGA3_ConvertDensity(BaseDensity_kgm3, AGA3_DENSITY_KG_M3, AGA3_DENSITY_LBM_FT3);
TotalMass_lbm := AGA3_ConvertMass(TotalMass_kg, AGA3_MASS_KG, AGA3_MASS_LBM);
TotalBaseVolume_ft3 := AGA3_ConvertVolume(TotalBaseVolume_m3, AGA3_VOLUME_M3, AGA3_VOLUME_FT3);
```

Figure 3-31. Application Example 2 - Code

Between lines 1 and 20, in the first cycle of application, the input variables are initialized. In real applications, the input variables should be read from instruments (temperature transmitters, pressure transmitters, etc). Also some fixed input parameters, like bore and tube diameters, are initialized in this code section. The ideal gas specific gravity and compressibilities are also initialized in this section, but in real applications they should be input by the user.

Between lines 23 and 33, several functions for converting units are used. They convert units of inputs in US system to metric system, as required by function block AGA3_Flow.

Between lines 36 and 60, the instance of function block AGA3_Flow is called. Note that:

- InformDensities is FALSE, so the densities (InputDensity and InputBaseDensity) are not supplied as inputs, and were suppressed from the input parameter list.
- Input parameters IdealSpecificGravity, Compressibility and BaseCompressibility were provided as inputs, because InformDensities is FALSE.
- The viscosity is informed with the constant value suggested by AGA-3 standard for natural gas (0.010268 cP).
- The isentropic exponent is informed with the constant value suggested by AGA-3 standard for natural gas (1.3).

Between lines 63 and 70, an instance of AGA3_Totalizer is called for calculating the totalized mass.

Between lines 72 and 78, another instance of AGA3_Totalizer is called for calculating the totalized volume at base condition.

In lines 81 and 82, the reset and preset commands, used in AGA3_Totalizer instances, are canceled. They must last only one PLC cycle.

Between lines 85 and 91, several functions for converting units are used. They convert units of outputs in metric system (produced by function blocks AGA3_Flow and AGA3_Totalizer) to US system.

Application Example 3 – Incompressible Fluid

The following example, developed in ST language, contains a single instance of function block AGA3_Flow.

> NOTE:
> Several instances of function blocks AGA3_Flow and AGA3_Totalizer can exist in the same application.

In this example, the fluid is assumed to be an incompressible fluid, like a liquid.

The next figure shows the variables declared in the application.

Configuration (Bus) | MainPrg

```
1  PROGRAM MainPrg
2  VAR
3      AGA3_FlowCalc : AGA3_Flow;              // Instance of function block for flow calculation
4
5      FirstCycle : BOOL := TRUE;              // Flag for indicating the first cycle of application example
6
7      // Inputs in units required by AGA3_Flow function block
8      Temperature_C : LREAL;                  // Temperature of actual condition in Celsius
9      Pressure_bar : LREAL;                   // Absolute pressure of actual condition in bar
10     DiffPressure_mbar : LREAL;              // Differential pressure in millibar
11     BaseTemperature_C : LREAL;              // Temperature of base condition in Celsius
12     BasePressure_bar : LREAL;               // Absolute upstream pressure of base condition in bar
13     BoreDiameter_Metric : AGA3_Diameter;    // Bore diameter and complementary information in metric units
14     TubeDiameter_Metric : AGA3_Diameter;    // Tube diameter and complementary information in metric units
15     InputDensity_kgm3 : LREAL;              // Input value of density at actual condition (kg/m3)
16     InputBaseDensity_kgm3 : LREAL;          // Input value of density at base condition (kg/m3)
17     Viscosity_cP : LREAL;                   // Viscosity in centipoise (same in US and metric)
18
19     // Outputs in units calculated by AGA3_Flow function block
20     MassFlow_kghr : LREAL;                  // Mass flow in kg/hr
21     ActualFlow_m3hr : LREAL;                // Actual volume flow in m3/hr
22     BaseFlow_m3hr : LREAL;                  // Base volume flow in m3/hr
23     WarningBits : AGA3_Warning;             // Warning bits
24     SomeWarning : BOOL;                     // Indicates at least one warning bit is set
25     ErrorBits : AGA3_Error;                 // Error bits
26     SomeError : BOOL;                       // Indicates at leas on error bit is set
27 END_VAR
```

Figure 3-32. Application Example 3 - Variables

In line 3, an instance of AGA3_Flow function block is declared.

In line 5, boolean flag FirstCycle is declared to control some initializations.

Between lines 8 and 17, inputS for AGA3_Flow function block are declared, in the suitable metric units.

In line 8, the actual temperature in Celsius (Temperature_C) is declared.

In line 9, the actual absolute pressure in bar (Pressure_bar) is declared.

In line 10, the differential pressure in millibar (DiffPressure_mbar) is declared.

In line 11, the base temperature in Celsius (BaseTemperature_C) is declared.

In line 12, the base absolute upstream pressure in bar (BasePressure_bar) is declared.

In line 13, the data structure with diameter properties of bore in metric units (BoreDiameter_Metric) is declared.

In line 14, the data structure with diameter properties of tube in metric units (TubeDiameter_ Metric) is declared.

In line 15, the density at actual condition (InputDensity_kgm3) is declared.

In line 16, the density at base condition (InputBaseDensity_kgm3) is declared.

In line 17, the viscosity (Viscosity_cP) is declared.

Between lines 20 and 26, outputs of AGA3_Flow function block are declared.

In line 20, the mass flow in kg/hr (MassFlow_kghr) is declared.

In line 21, the volume flow at actual conditon in m^3/hr (ActualFlow_m3hr) is declared.

In line 22, the volume flow at base conditon in m^3/hr (BaseFlow_m3hr) is declared.

In line 23, the warning bits data structure is declared (WarningBits).

In line 24, the logic OR of warning bits is declared (SomeWarning).

In line 25, the error bits data structure is declared (ErrorBits).

In line 26, the logic OR of error bits is declared (SomeError).

The next figure shows the code for this application.

```
1  IF FirstCycle THEN
2      FirstCycle := FALSE;
3      // initialize inputs in first cycle of this application example
4      Temperature_C := -17.78;
5      DiffPressure_mbar := 22.38;
6      Pressure_bar := 1.01325;
7      BaseTemperature_C := 15;
8      BasePressure_bar:= 1.01325;
9      BoreDiameter_Metric.Alfa := 1.665E-05;
10     BoreDiameter_Metric.RefTemp := 20;
11     BoreDiameter_Metric.Diameter := 40.481;
12     TubeDiameter_Metric.Alfa := 1.116E-05;
13     TubeDiameter_Metric.RefTemp := 20;
14     TubeDiameter_Metric.Diameter := 202.729;
15     InputDensity_kgm3 := 932.26;
16     InputBaseDensity_kgm3 := 910.83;
17     Viscosity_cP := 1.865E3;
18 END_IF
19
20 // Call of function block instance for flow calculation
21 AGA3_FlowCalc(
22     Temperature:= Temperature_C,
23     Pressure:= Pressure_bar,
24     DiffPressure:= DiffPressure_mbar,
25     CutoffDiffPressure:= 0,
26     DownstreamPressureTap:= FALSE,
27     BaseTemperature:= BaseTemperature_C,
28     BasePressure:= BasePressure_bar,
29     BoreDiameter:= BoreDiameter_Metric,
30     TubeDiameter:= TubeDiameter_Metric,
31     InformDensities:= TRUE,
32     InputDensity:= InputDensity_kgm3,
33     InputBaseDensity:= InputBaseDensity_kgm3,
34     Viscosity:= Viscosity_cP,
35     IsentropicExponent:= -1,
36     MassFlow=> MassFlow_kghr,
37     ActualFlow=> ActualFlow_m3hr,
38     BaseFlow=> BaseFlow_m3hr,
39     WarningBits=> WarningBits,
40     SomeWarning=> SomeWarning,
41     ErrorBits=> ErrorBits,
42     SomeError=> SomeError);
```

Figure 3-33. Application Example 3 - Code

Between lines 1 and 18, in the first cycle of application, the input variables are initialized. In real applications, the input variables should be read from instruments (temperature transmitters, pressure transmitters, etc). Also some fixed input parameters, like bore and tube diameters, are initialized in this code section.

Between lines 21 and 42, the instance of function block AGA3_Flow is called. Note that:

- InformDensities is TRUE, so the densities (InputDensity and InputBaseDensity) are supplied as inputs. This is necessary for any fluid different from natural gas.

- Input parameters IdealSpecificGravity, Compressibility and BaseCompressibility were supressed, because InformDensities is FALSE.
- The isentropic exponent is informed with the constant value -1, indicating that the fluid is incompressible. If the fluid was compressible, the value should be greater than zero.

4. Real and Ideal Specific Gravity

The AGA3_Flow function block, in some circumstances, requires the ideal specific gravity of natural gas as input (for instance, see application example 2 in previous chapter).

If the user knows the real gas specific gravity instead, he can use the following equation to convert it into ideal gas specific gravity:

Gi = Gr * [Pm_air * (Tm_gas + 273.15) * Zm_gas] / [Pm_gas * (Tm_air + 273.15) * Zm_air]

where:

- Gi = ideal gas specific gravity
- Gr = real gas specific gravity
- Pm_air = measured air pressure (bar)
- Pm_gas = measured gas pressure (bar)
- Tm_air = measured air pressure (°C)
- Tm_gas = measured gas pressure (°C)
- Zm_air = air compressibility, at Tm_air and Pm_air
- Zm_gas = air compressibility, at Tm_gas and Pm_gas

The air and gas compressibilities must be calculated using some appropriate method, like AGA-8 detailed or gross method.

5. Glossary

General Glossary

Algorithm Finite sequence of well defined instructions, for problem solution.

Bit Basic information unit which can assume state 0 or 1.

Byte Information unit composed by 8 bits.

Programmable controller Also called PLC. Equipment which executes a control under the applicative program command. It’s composed by a CPU, a power supply and a I/O structure.

CP See Programmable controller.

Database Data base.

Default Pre defined value for a variable, used in case there’s no definition.

Diagnostic Procedure used to detect and isolate failures. It’s also the data group used for such determination, which serves for problem analysis and correction.

Download Program or configuration load in the PLC.

I/O See Input/output

Input/output Also called I/O. Data I/O devices of a system. In case of PLCs, typically correspond to digital or analog inputs or outputs modules which monitor or activate the controlled device.

Hardkey Connector normally connected to the parallel interface of a PC in order to avoid the execution of software illegal copies.

Hardware Physical equipments used in data processing where the programs (software) are executed.

IEC 61131 Generic standard for operation and utilization of PLCs. Old IEC 1131.

Interface Device which adapts electrically and/or logically the signal transference between two pieces of equipment.

Interruption High priority attending event which temporarily stops the program execution and detour for a specific attending routine.

kbytes Memory quantity unit. Means 1024 bytes (if 1kbyte).

Programming language A group of rules and conventions used for a program creation.

Logic Graphic matrix where are inserted the language instructions of a relay diagram which compose a applicative program. A group of logics organized in sequence form a program module.

MasterTool Identifies the Altus software for PC, executable only in Windows®, which allows the development of applicative for the Ponto Series CPUs, AL-2000, AL-3000 e Quark. Throughout this manual, this software is referenced by its acronym or as MasterTool programming.

Menu Set of options available and displayed by a program on video and that can be selected by the user to activate or perform a certain task.

Module (referencing hardware) Basic element of a complete system that has well defined functions. Normally the system is connected by connectors and can be easily replaced.

Module (referencing software) Part of an application program capable of performing a specific function. It can be run independently or in conjunction with other modules, exchanging information via parameter passing.

I/O Modules Module belonging to the inputs and outputs subsystem.

PLC Acronym for programmable logic controller.

Start up Procedure for final clearance of the control system when the programs of all stations and remote CPUs are run together, having been developed and verified individually.

Applicative Program It’s the program loaded into a PLC, which determines the operation of a machine or process.

Executive Program Operating system from a programmable controller. Controls the basic functions of the controller and the execution of applicative programs.

RAM Acronym for random access memory. It's where all the memory addresses can be accessed directly at random and at the same speed. It is volatile, thus, its contents are lost when the device is powered down, unless you have a battery for retaining values.

Software Computer programs, procedures and rules related to the operation of a data processing system.

I/O Subsystem Set of analog or digital I/O modules and interfaces of a programmable controller.

Tag Name associated with a variable or a logic that allows a brief identification of its contents.

CPU Abbreviation for central processing unit. Controls the information flow, interprets and executes program instructions and monitors the devices in the system.

Upload PLC configuration or program reading.

Revisões deste Manual

Esta página não fará parte da versão enviada ao usuário (arquivo .PDF). Ela serve somente para o uso da Altus. Preencha o formulário a seguir para manter o histórico das alterações correspondentes a cada revisão do manual.

Revisão: A	Data: 16/12/2013
Aprovação: Fernando Trein	
Autor: Osmar Brune	

Observações:

- *Versão Inicial*
- *Revisor Gerencial Sr.Fernando Trein*
- *Revisor Técnico Sr. Daniel Salazar/ Matheus Webler*