1. Product Description

The Nexto family represents a line of robust and versatile Programmable Logic Controllers (PLCs), designed to support applications ranging from small automation systems with critical timing requirements to complex medium- and high-performance systems. With a modular architecture, intelligent design, advanced diagnostic features, and compact dimensions, the PLCs of the Nexto family provide a powerful and scalable solution for a wide range of industrial automation challenges.

Mastertool X is a complete engineering environment that integrates programming, simulation, configuration, and debugging tools into a single platform. Developed with a focus on flexibility, integration, and ease of use, the software supports the five programming languages defined by the IEC 61131-3 standard: Structured Text (ST), Sequential Function Chart (SFC), Function Block Diagram (FBD), Ladder Diagram (LD), and Continuous Function Chart (CFC). The possibility of combining different languages within a single project provides better organization, code reuse, and development efficiency.

Mastertool X covers all stages of the lifecycle of an automation application: from the design of the system topology, through programming and realistic simulation of the control logic, to real-time diagnostics and monitoring of the application in operation. This allows the user to validate the application behavior even before transferring it to the physical system.

The software also incorporates robust security mechanisms, such as:

- **Intellectual Property Protection:** allows restricting access to projects or specific files through password protection, ensuring the confidentiality of the developed content.
- **Secure Login to the CP:** prevents unauthorized access to the application on the PLC by requiring authentication before executing critical commands such as program downloads, modifications, or variable forcing.

Field network configuration is simplified in an unprecedented way: Mastertool X eliminates the need for additional tools, allowing industrial protocols to be configured directly within the platform, reducing development time and facilitating device integration.

To further increase productivity, Mastertool X also provides additional features such as:

- **Printing Module:** generation of reports containing parameters and configurations of the modules used.
- **Logic Printing:** complete documentation of the application code.
- **Advanced Project Verification:** analysis of syntax, current consumption, assembly rules, and validation of parameters of Nexto family modules.
- **Real-Time Debugging:** detailed inspection of the control logic, with support for breakpoints, variable visualization, and step-by-step execution.

2. Technical Description

2.1. Mastertool X Versions

The Mastertool X is available in different licensing models, each with a set of features tailored to the specific needs of different types of applications.

- **Lite:** free version of the software, ideal for smaller applications, providing an accessible solution for simpler developments.
- **Advanced:** licensed version with full access to all engineering environment features, including support for large-scale applications and advanced resources such as half-cluster redundancy, recommended for critical and high-availability systems.

Each of these versions has specific characteristics, purposes, and functionalities for different use cases.

	Lite	Advanced (1 Year)	Advanced (Perpetual)
Free Version	✓	×	×
Perpetual License	✓	×	✓
Available Languages:	5	5	5
Structured Text (ST)	✓	✓	✓
Sequential Function Chart (SFC)	✓	✓	✓
Function Block Diagram (FBD)	✓	✓	✓
Ladder Diagram (LD)	✓	✓	✓
Continuous Function Chart (CFC)	✓	✓	✓
Bus Expansion Support	×	✓	✓
Ethernet Expansion Support	×	✓	✓
PROFIBUS Support	×	✓	✓
Half-Cluster Redundancy Support	×	✓	✓
CPU Redundancy Support	×	✓	✓
DNP3 Protocol Support	×	✓	✓
IEC 60870-5-104 Protocol Support	✓	✓	✓
IEC 61850 Protocol Support	×	✓	✓
Xtorm Ethernet Interface NIC Teaming Support	×	✓	✓

Table 1: Version Features

Notes:

Advanced License (1 Year): License valid for 365 days from the purchase date.

Continuous Function Chart (CFC): The CFC language has two editors. In the first, all functions are enumerated with a single execution order. In the second, the user can edit logical groups on individually enumerated pages – therefore it is called *Page Oriented*.

	Lite	Advanced
XF300-B	✓	✓
XF300	✓	✓
XF315	✓	✓
XF325	✓	✓
XF325-W	✓	✓

	Lite	Advanced
XP300	✓	✓
XP315	✓	✓
XP325	✓	✓
XP340	✓	✓
XP350	✓	✓
XP351	✓	✓
NX3008	✓	✓
NX3035	✗	✓
HX3040	✓	✓
NL717	✓	✓
CODESYS Control Win V3 - x64	✓	✓

Table 2: Supported Products

Note:

Support for CODESYS Control Win V3 - x64: CODESYS Control Win V3 x64 is supported directly by the CODESYS team. Available only for use in DEMO mode.

2.2. Data for Purchase

2.2.1. Included Items

The Mastertool X software is sold as a service, with the contract and the respective license sent in digital format to the customer. Get in touch with the Altus commercial department if physical media is required with the product.

2.2.2. Product Code

The following codes shall be used for product purchase:

Code	Description
MT9000 Lite	MT9000 Lite
MT9000/ADV-PL	MT9000 Advanced with perpetual license
MT9000/ADV-1YR	MT9000 Advanced with 1-year license

Table 3: Product Code

3. Product Features

3.1. IEC 61131-3 Programming Languages

Mastertool X provides all the editors defined by the IEC standard for application development: Structured Text (ST), Sequential Function Chart (SFC), Function Block Diagram (FBD), Ladder Diagram (LD), and Continuous Function Chart (CFC).

All editors have been specially developed to ensure optimal usability. Some examples:

- When using FBD or LD, the user can freely switch between these editors
- Elements of a language can be inserted directly or dragged from a toolbox into the editor
- Mastertool X offers an intelligent input assistant and the *Autocomplete* functionality
- Standard ST language constructs such as *IF* and *FOR* can be hidden or displayed in the textual editors
- Autocomplete for compound commands such as *IF .. END_IF* and *WHILE .. END_WHILE*
- Execution time monitoring for steps as well as diagnostic functionality
- Steps and transitions in the SFC editor and all elements in the CFC editor can be encapsulated in macros
- Automatic variable declaration
- Graphical table for variable declaration

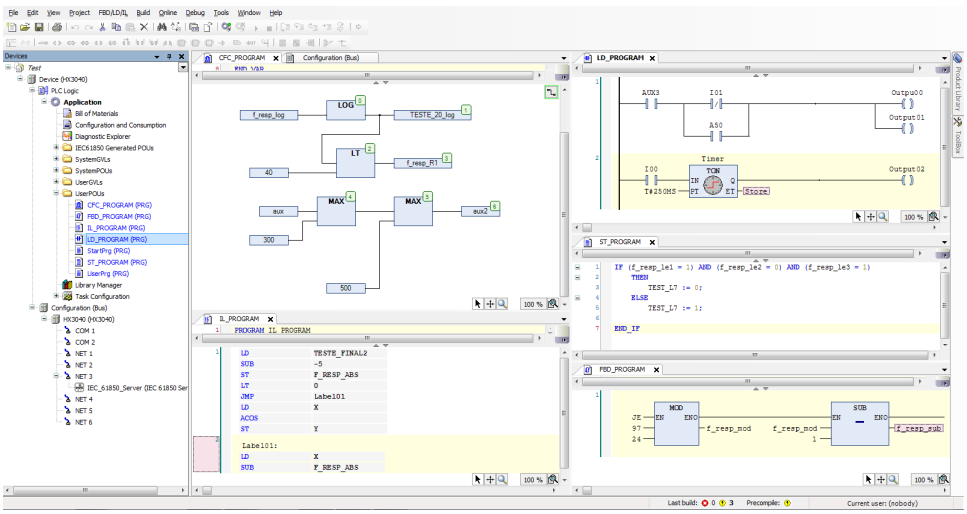

Figure 1: IEC 61131-3 Programming Languages

3.2. Editors for Project Configuration and Hardware Configuration

With the aid of specialized editors, a project can be easily configured in Mastertool X. The graphical tool provides a fast and comprehensive way to configure the system; the user only needs to drag the selected module from the product library and drop it onto the rack to add it to the application. Additionally, the user has a complete view of the application architecture, including the physical position and information of the modules.

The configuration of standard communication protocols, such as DNP, IEC104, and MODBUS, is integrated into the programming tool. This feature allows the user to configure all parameters in a single place, without the need to switch between different tools.

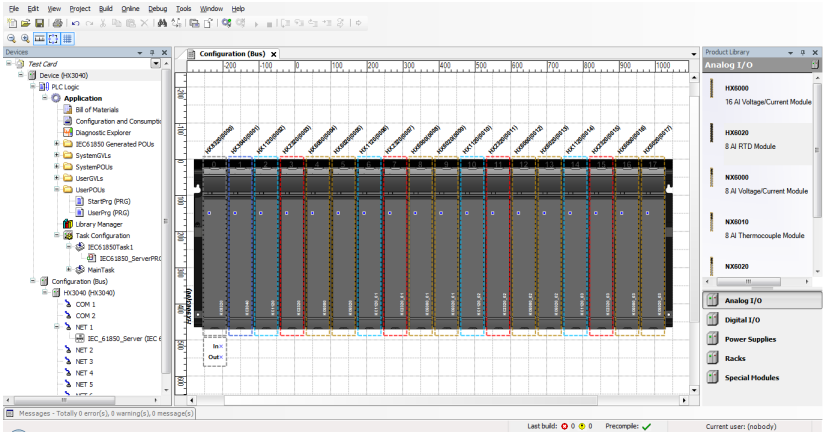

Figure 2: Editors for Project Configuration and Hardware Configuration

3.3. Object Oriented Programming

Mastertool X offers object-oriented programming with the well-known advantages of modern high-level languages such as Java or C++, including the use of classes, interfaces, methods, inheritance, and polymorphism. Functional blocks written in IEC can be extended, and these extensions are available across all aspects of engineering. Object-oriented programming provides significant benefits for the user, for example, by reusing existing parts of an application or enabling multiple developers to work on the same application simultaneously.

3.4. Online, Debugging and Commissioning Features

The code generated by the application is sent to the device with a simple mouse click. Once Mastertool X is online, it provides extensive information for quick and efficient debugging, testing, and commissioning.

The values of declared variables are displayed directly in the program code. These values can be modified or forced without difficulty. By configuring line-by-line breakpoints in the code, errors can be easily detected. Breakpoints in Mastertool X can be associated with specific conditions to achieve greater precision during debugging. Using the “Run to Cursor” option, the user can follow the execution of the application for a complete cycle.

If the application is modified, it is recompiled and then loaded again without stopping the control. Changes in multiple POUs (Program Organization Units), variables, or data types are also possible. This functionality is known as Online Change, which allows online modification. It enables shorter development cycles and increased productivity, resulting in cost reduction and enhanced competitiveness.

Tracing is another frequently used tool when the user wants to record data or events for testing or commissioning. The stored data is fully integrated into Mastertool X and can, of course, be used to visualize the application data.

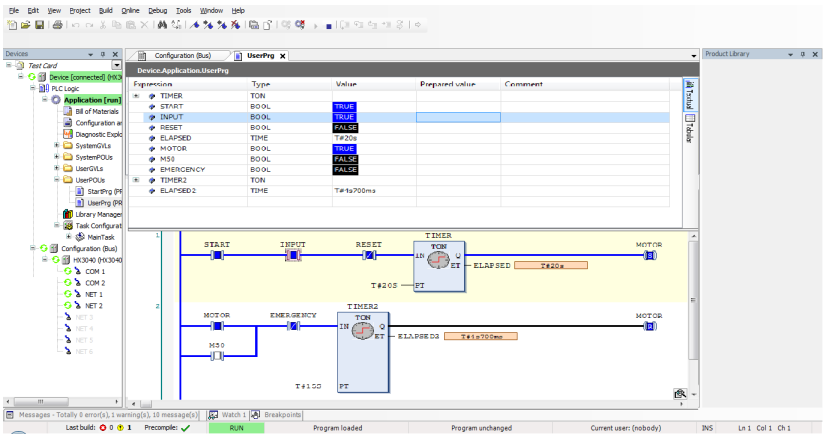

Figure 3: Online, Debugging and Commissioning Features

3.5. Simulation

The simulation tool available in Mastertool X allows the user to evaluate and test various logics and algorithms. This feature enables the development and testing of the user's application without the need to connect to a physical CPU. Simulation is also useful for training, documentation, and test case evaluation.

Obviously, as it is a simulator, the tool has some limitations, such as not allowing the testing of communication interfaces. Consequently, the application can only be fully tested once it is loaded onto the controller.

3.6. User Documentation & Help Files

Since programming a PLC according to the languages defined by the IEC 61131-3 standard is a complex task, Mastertool X provides a complete set of help files with tips and descriptions intended to guide the user and serve as an initial knowledge base for troubleshooting while developing logic or using some feature of the tool. In addition, the help files are available in different languages according to the installation options.

As part of the user documentation, Mastertool X allows printing application documents such as the bill of materials (BOM), POUs, bus configuration, tags and descriptions, among other options.

3.7. Advanced Diagnostics

One of the main innovations of the Nexto Series is its high diagnostic capability. This concept is a requirement for the development of complex and large-scale applications, where the proper use of such information is necessary for maintenance, troubleshooting, and the prediction of potential failures. The *Advanced Diagnostics* are also present in Mastertool X, where the user can access diagnostic structures through monitoring windows when connected to a CPU.

3.8. Function Block Libraries

Mastertool X provides numerous ready-to-use function block libraries. Functions such as process variable handling, mathematical functions, and timers are examples of functions that are available together with Mastertool X.

3.9. Docking View

The *Docking View* technology allows the user to customize the Mastertool X environment so that it meets their personal needs. This feature provides a user-friendly interface aimed at maximizing the user's experience with the tool.

3.10. Languages

Mastertool X software is available in a few languages. After the installation, the interface assumes the language of Computer Operating System. The language can be changed after installation without the need for resettlement.

3.11. Compatibility with Other Product

Versions of Mastertool X are not compatible with all versions of controllers. To know which version is compatible, the technical datasheet document of each controller should be consulted.

3.12. Minimum and Recommended Requirements

Mastertool X presents as minimum and recommended requirements for its installation and utilization the following specs:

	Mastertool X
Platform	PC with operating system: Windows 10® (64-bit) or Windows 11® (64-bit)
Processor	2.5 GHz
Disk Space	12 Gbytes
RAM	8 Gbytes, 16 Gbytes (recommended)
Resolution	1024 x 768
Language	Any language

Table 4: Minimum and Recommended Requirements for Installation and Operation

Note:

Requirements: As a rule, PCs that meet the minimum requirements can be used for simple and non-redundant applications. Complex and/or redundant applications should use PCs that meet at least the recommended configurations.

4. Installation

The information required for the installation of Mastertool X can be found in *Mastertool X User Manual*.

5. Programming

Mastertool X allows programming using five different programming languages, four of which are defined by the IEC 61131-3 standard, along with one additional language.

5.1. Ladder Diagram (LD)

The *Ladder Diagram* (LD) programming language is a graphical language based on electrical diagrams that represent interconnected contacts and coils, highlighting the flow of energization between the elements. It is used to describe the behavior of programs, function blocks, and functions, as well as steps, actions, and transitions in the SFC language.

This language basically consists of a technique that uses relay logic drawings. These diagrams were already used to document relay cabinets even before the existence of PLCs. The basic elements are normally open and normally closed contacts and relay coils. For proper operation, the elements must be connected in such a way that they link a vertical bar on the left, representing an energized bus, to the bar on the right, representing ground. Due to their shape, these diagrams were called ladder.

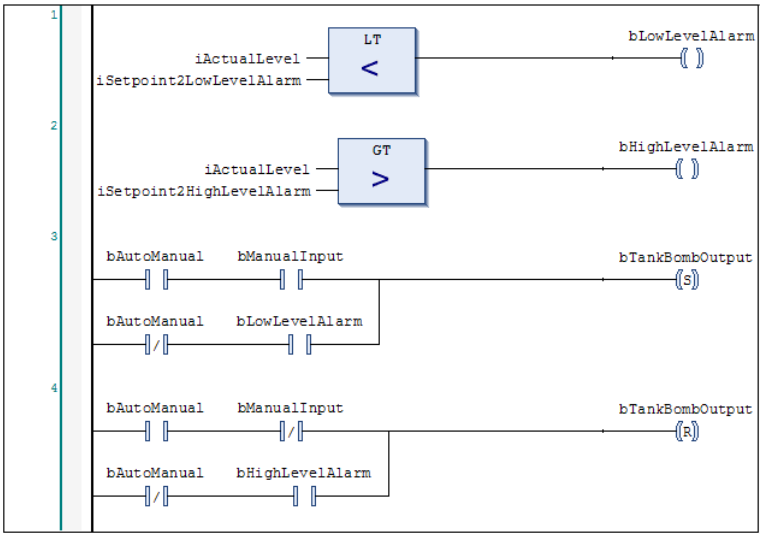

Figure 4: Ladder Diagram (LD)

5.2. Structured Text (ST)

The programming language *Structured Text* (ST) is a textual language, as its name indicates, of high-level syntax similar to Pascal (ISO 7185). It was specifically developed for industrial control and is used to describe the behavior of programs, function blocks, functions, as well as steps, actions, and transitions in the SFC language. The language is flexible and easy to learn for software developers in general.

ST includes common commands found in structured languages, such as conditional tests, selection commands, and various types of loops. It also includes specific commands for logical and mathematical operations without the need for function calls.

It is essential to have some knowledge of this language's syntax, as variable declarations are made through it, even when using the programmer's graphical resources. For users who do not wish to use ST, Mastertool X provides a table for graphical variable declaration or the option for automatic declaration.

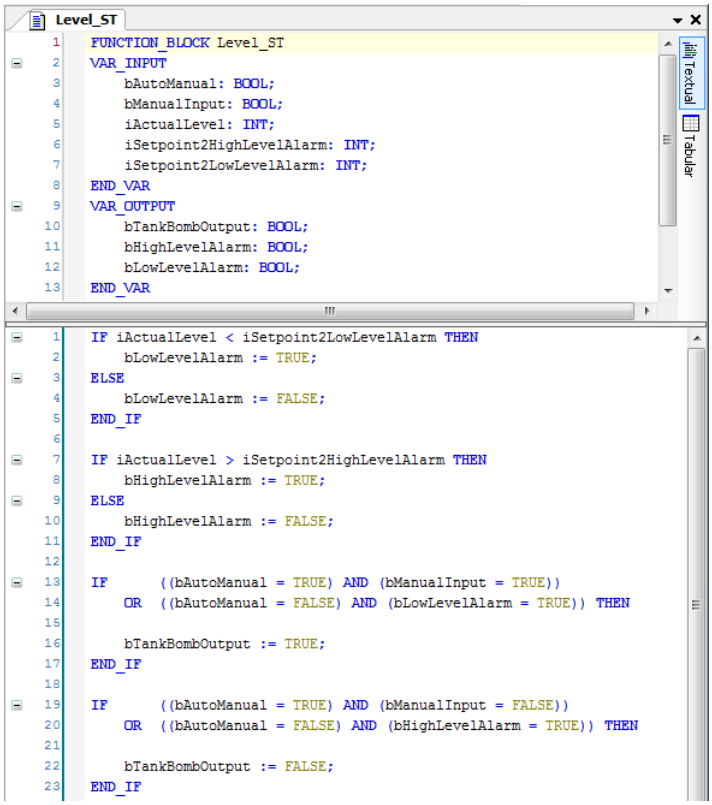

Figure 5: Structured Text (ST)

5.3. Function Block Diagram (FBD)

The *Function Block Diagram* (FBD) programming language is a graphical language based on circuit diagrams that represents interconnected blocks, highlighting the flow of signals between elements. It is used to describe the behavior of programs, function blocks, and functions, as well as steps, actions, and transitions in the SFC language.

The concept of blocks represents any element that has inputs, performs a specific processing, and then writes the result of the operations to the outputs. Blocks are divided into two distinct types: function blocks and functions. These two types of POUs differ in that function blocks retain the value of local variables between block calls; that is, they must be instantiated and can execute state machines across several execution cycles. Functions, on the other hand, execute and, once execution is finished, do not retain any information, only writing the result to the output. In general, functions are native to the system, but they can also be implemented by the user.

An important characteristic of this language is that processing is executed from left to right and from top to bottom in the diagram.

Function blocks can also, like functions, be used together with the *Ladder Diagram*, increasing the programming possibilities with this language.

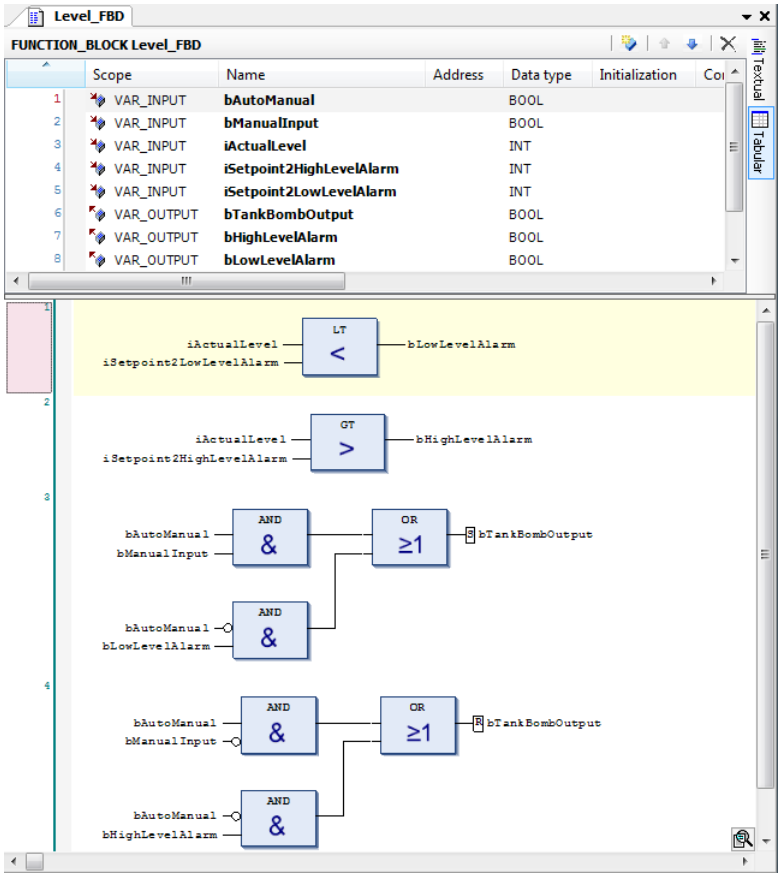

Figure 6: Function Block Diagram (FBD)

5.4. Sequential Function Chart (SFC)

The programming language *Sequential Function Chart* (SFC) is a graphical language based on techniques for describing sequential behavior. The European standard for this type of behavior is defined in IEC 848 and is based on Petri Nets. The IEC 61131-3 standard introduced modifications to IEC 848 in order to adapt SFC to the other languages defined by the standard.

Therefore, this language is used to describe the sequential behavior of a system, structure programs, describe the low-level operation of a sequential process, define the basis of a batch process, represent data communication, and model event-driven systems such as state machines.

The language consists of several steps connected by vertical lines, where each step represents a state in which the program remains until the transition condition described on the connection line between steps is satisfied.

The flow is top-down, but branches for return can also exist. The actions executed in each step can be performed continuously or be event-oriented, such as entering or leaving a state.

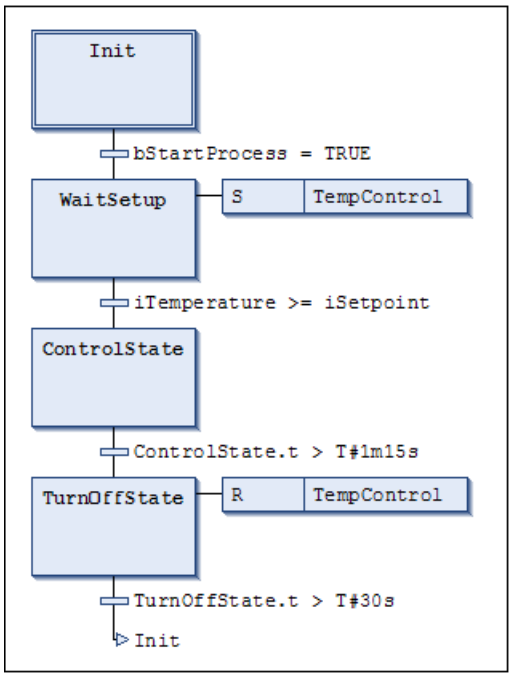

Figure 7: Sequential Function Chart (SFC)

5.5. Continuous Function Chart (CFC)

The *Continuous Function Chart* (CFC) programming language is a graphical language that is not defined by the IEC 61131-3 standard, but is complementary to it. It is similar to the FBD language; however, when a block is inserted, it must be numbered. This numbering is used to indicate the execution sequence of the diagram, facilitating the development and understanding of the diagram and solving this issue present in the diagram described by the standard. The *Continuous Function Chart – Page Oriented* has the same characteristics and programming approach as the usual CFC; however, it separates the logic into pages, facilitating debugging and the logical hierarchy.

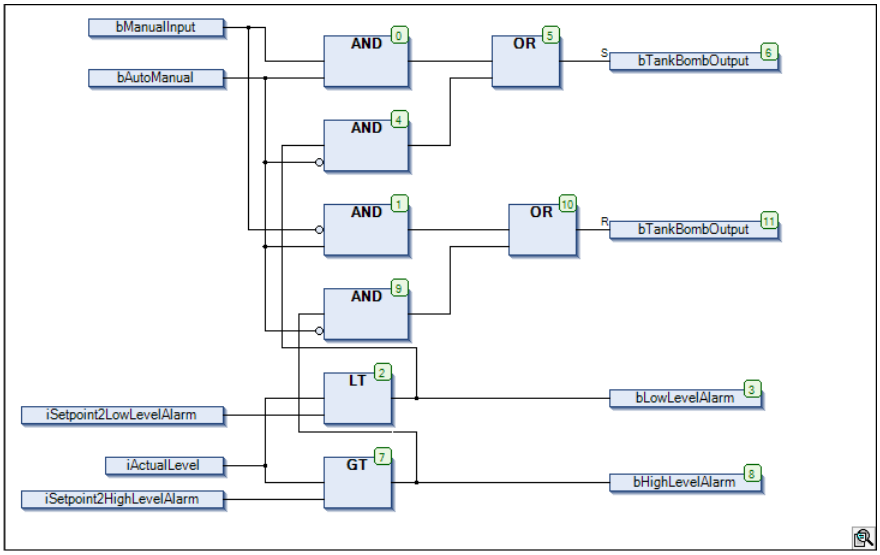

Figure 8: Continuous Function Chart (CFC)

6. Manuals

For further technical details, configuration, installation and programming, the table below should be consulted.

The table below is only a guide of some relevant documents that can be useful during the use, maintenance, and programming of this product.

Code	Description	Language
MU299611	Mastertool X User Manual	English
MU299049	Manual de Utilização Mastertool X	Portuguese
MP399609	IEC 61131 Programming Manual	English
MP399048	Manual de Programação IEC 61131	Portuguese
MU214600	Nexto Series User Manual	English
MU214000	Manual de Utilização Série Nexto	Portuguese
MU223600	Hadron Xtorm Utilization Manual	English
MU223000	Manual de Utilização Hadron Xtorm	Portuguese
MU216600	Nexto Xpress User Manual	English
MU216000	Manual de Utilização Nexto Xpress	Portuguese
MU218600	Nexto XF User Manual	English
MU218000	Manual de Utilização Nexto XF	Portuguese

Table 5: Related Documents