1. Product Description

Automation of power applications is characterized by the use of rugged and reliable equipment and devices with high technology, capability to operate in hostile environments, where there are significant levels of electromagnetic interference and higher operating temperatures. This is the reality of applications in hydropower plants, power substations and wind farms, among others.

In this context, Hadron Xtorm Series is an innovative Remote Terminal Unit (RTU), perfect for applications in power generation, transmission and distribution. The Series has an ideal set of features with high performance for the different stages in the life cycle of any application, such as engineering, installation and commissioning, offering cost and risk reduction for every single stage. It also minimizes downtime and system maintenance when in operation. With intuitive and friendly interfaces, accurate and smart diagnostics, modern and robust design, as well as several innovative features, Hadron Xtorm Series surpasses the requirements of applications in this market.

The Series has a smart and versatile architecture, offering modularity in input and output (I/O) points, redundancy options, module hot swapping, high-speed communication protocols, such as IEC 61850, IEC 60870-5-104 and DNP3, logic implementation in accordance to IEC 61131-3 standard and time synchronism.

MasterTool Xtorm is a complete tool for programming, debugging, configuration and simulation of user applications. The software is based on the concept of integrated tool, providing flexibility and ease of use, allowing users the configuration of GOOSE messages, parameterizing of Datasets and IED configuration. Besides, it provides the user six programming editors which use the programming languages defined by IEC 61131-3 standard: Structured Text (ST), Sequential Function Chart (SFC), Function Block Diagram (FBD), Ladder Diagram (LD), Continuous Function Chart (CFC) and Continuous Function Chart (CFC) – Page Oriented. MasterTool Xtorm allows the usage of different languages within the same application, providing the user a powerful way of organizing his/her application and reutilizing programs developed in previous applications.

This product offers features for all development stages of an automation system, starting with a graphical analysis of the architecture's topology, going through a powerful programming environment that supports IEC 61131-3 languages and an environment for IEC 61850 protocol data configuration. It also provides a tool for realist simulations, where the user can verify the behavior of the application before executing it in a real system. Finally, it also has a complete interface for the visualization of diagnostics and status.

MasterTool Xtorm also offers two different application protection schemes and security features: Intellectual Property Protection and Secure RTU Login. Intellectual Property Protection is targeted to protect user's intellectual property, allowing the user to protect the complete project and files inside it by defining a password to access them. This means that these files will be available (both read and write operations) only after unlocking them with the correct password. Secure RTU Login provides a way to protect the user application from any unauthorized access. By enabling this feature, Hadron Xtorm Series' RTU will request a user password before performing any available command between MasterTool Xtorm and RTU.

In MasterTool Xtorm, the user doesn't need a special software to configure field networks, because it already complies this requirement through one single tool, thus reducing development time and simplifying the application.

Other important features are also available to increase user productivity such as: print module, which generates a printable report with specific module parameters and application configuration; logic print, which generates a printable report with all application codes; project verification, which helps the user verify different aspects while programming such as: program syntax, power supply current consumption, module positioning rules of Hadron Xtorm Series, module parameterization and configuration; real time debugging, which provides a way of verifying the functionality of the application step by step, verifying the content of variables and add/remove breakpoints while programming Hadron Xtorm Series' RTU.

2. Ordering Information

Free download of MasterTool Xtorm Lite software is available on Altus website: www.altus.com.br.

2.1. Product Code

The following code should be used to purchase the product:

Code	Description
HD8500	MasterTool Xtorm

Table 1: Product Code

Note:

HD8500: MasterTool Xtorm software is available in 2 different versions: Lite and Advanced. Consult the table below for further details.

2.2. MasterTool Xtorm Versions

The table below presents the differences between versions of MasterTool Xtorm:

	Lite	Advanced
Free version	Yes	No
Available languages :	5	5
Structured Text (ST)	Yes	Yes
Sequential Function Chart (SFC)	Yes	Yes
Function Block Diagram (FBD)	Yes	Yes
Ladder Diagram (LD)	Yes	Yes
Continuous Function Chart (CFC)	Yes	Yes
Available Hadron Xtorm Series CPUs:		
HX3040	Yes	Yes
Bus expansion support	No	Yes
Bus expansion redundancy support	No	Yes
SNTP synchronization support	Yes	Yes
DNP3 synchronization support	No	Yes
IRIG-B synchronization support	Yes	Yes
Event grouping support	No	Yes
Engineering conversion support	Yes	Yes
Alarm configuration support	Yes	Yes
MODBUS protocol support	Yes	Yes
DNP3 protocol support	No	Yes
IEC 60870-5-104 protocol support	No	Yes
IEC 61850 protocol support	No	Yes
Ethernet interface redundancy support	No	Yes
Power supply redundancy support	Yes	Yes
CPU redundancy support	No	Yes
Limitation of I/O points	Yes	No
Maximum quantity of I/O points	512	Unlimited

Table 2: MasterTool Xtorm Versions

Notes:

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

MODBUS protocol support: Hadron Xtorm Series architectures can use MODBUS RTU Master and Slave protocols in serial interfaces and MODBUS TCP and RTU over TCP Client and Server protocols in Ethernet interfaces.

DNP3 protocol support: Hadron Xtorm Series architectures use DNP3 protocol as supervisory protocol.

IEC 60870-5-104 protocol support: Hadron Xtorm Series architectures use IEC 60870-5-104 protocol as supervisory protocol and as a protocol for communication with electrical equipment.

IEC 61850 protocols support: Hadron Xtorm Series architectures use IEC 61850 protocol to communicate with electric utility equipment.

Maximum quantity of I/O points: In MasterTool Xtorm Advanced version, there's no limit to the quantity of I/O points, but it is limited to 100 modules. The quantity of points will be limited, however, by the available memory in the CPU.

3. Product Features

3.1. RTU Configuration

MasterTool Xtorm has a complete and user-friendly graphical interface that allows the configuration of all RTU features, such as: time synchronization, engineering conversion, alarms, event grouping, and more.

There are specific screens to configure communication protocols, which are symbolic variable oriented, that means it is possible to configure them using the project's existing tags, without the necessity of managing direct memory addresses (%I, %Q and %M).

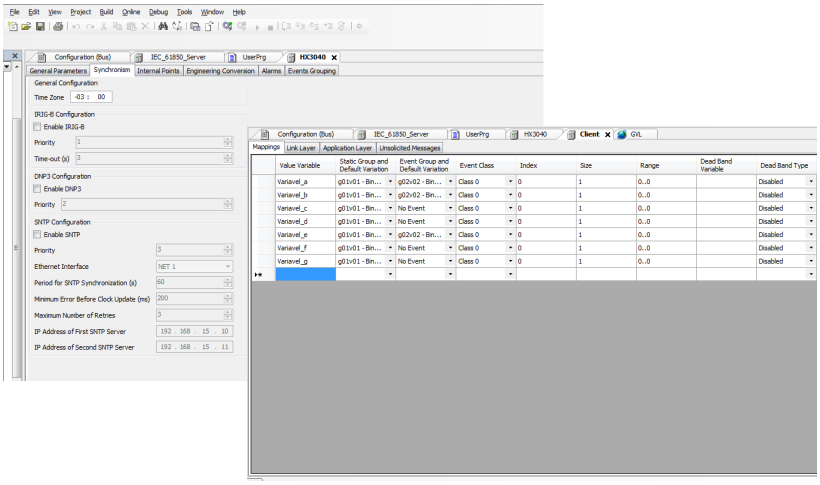

Figure 1: RTU Configuration

3.2. IEC 61850 Configurator

MasterTool Xtorm has an integrated IEC 61850 configurator, which makes the communication configuration process easier and faster. Through this configurator, it is possible to configure Logical Nodes, GOOSE messages and MMS Reports according to the IEC 61850 standard. It is also possible to import and export SCL files to configure other IEDs.

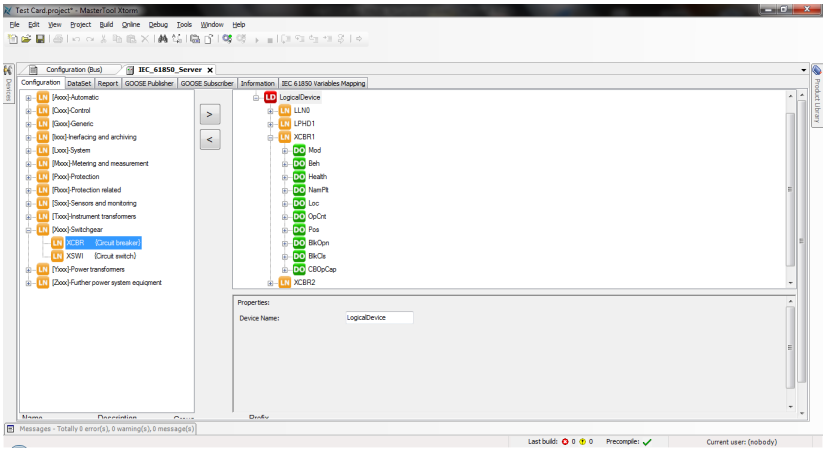

Figure 2: IEC 61850 Configurator

3.3. Integration of IEC 61850 with the IEC 61131-3 programming environment

For each logical node (LN) associated to the IED configuration system, a corresponding function block is declared automatically in the MasterTool Xtorm environment, therefore the user can access and monitor any data attribute through the IEC

61131-3 programming languages available in MasterTool Xtorm.

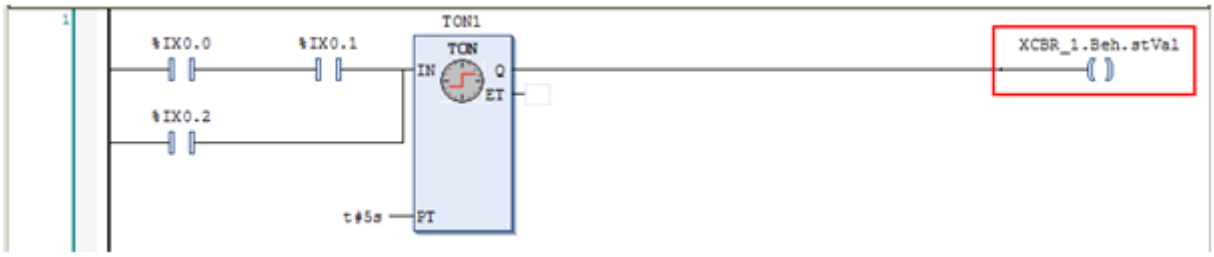

Figure 3: Integration of IEC 61850 with the IEC 61131-3 programming environment

3.4. IEC 61131-3 Programming Languages

MasterTool Xtorm offers all editors defined in 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 were specially designed to ensure optimal handling. Some examples:

- When working in FBD or LD you can freely switch between these editors
- Language elements can either be entered directly or dragged into the editor from a tool box
- MasterTool Xtorm offers an intelligent input assistant and an extended “Autocomplete” functionality
- Standard language constructs of ST language, such as *IF* and *FOR*, can be folded and unfolded in the text editor
- Auto declare for constructs such as *IF ... END_IF* and *WHILE . . . END_WHILE*
- Time monitoring mechanism for performed steps as well as the diagnostics functionality
- Steps and transitions in the SFC editor and all elements in the CFC editor can be encapsulated in macros
- Automatic declaration of variables
- Graphic table for declaration of variables

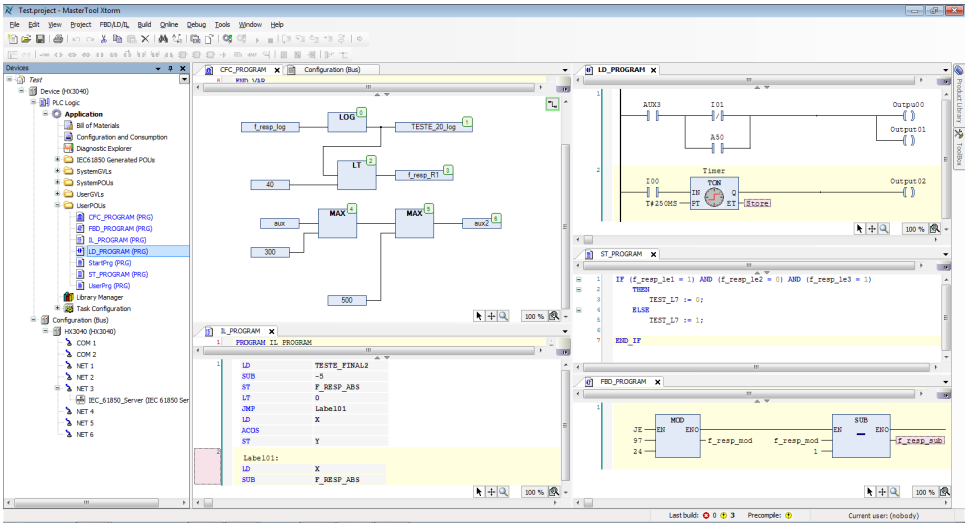

Figure 4: IEC 61131-3 Programming Languages

3.5. Editors for Project Configuration and Hardware Configuration

With the aid of special editors, a project can be easily configured in MasterTool Xtorm. The graphical tool allows a fast and comprehensive way to configure the system where the user just needs to drag from library products the selected module and drop it in the backplane to add them to the application. Additionally, the user has the complete visualization of the application architecture with the physical position and module information.

The configuration of standard communication protocols, such as DNP3, IEC104 and MODBUS, are integrated in the programming tool. This feature enables the user to set all configuration parameters in a single place, avoiding the need of switching between different software tools.

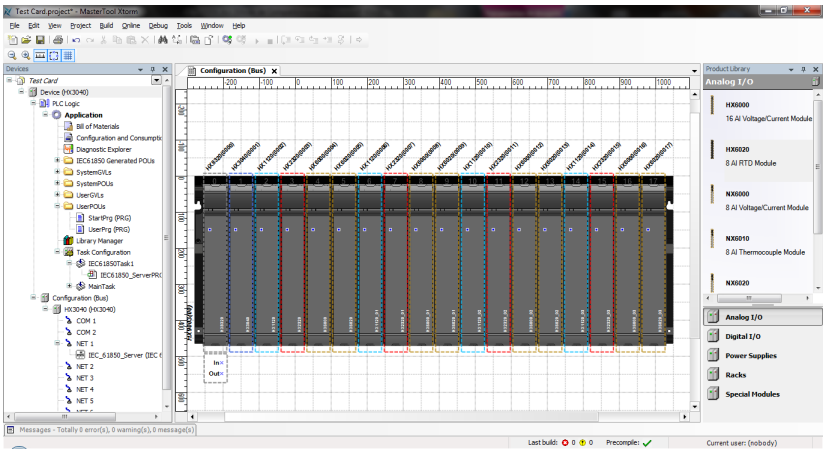

Figure 5: Editors for Project Configuration and Hardware Configuration

3.6. Object Oriented Programming

MasterTool Xtorm offers object-oriented programming with the known advantages from modern high-level languages such as JAVA or C++, such as the use of classes, interfaces, methods, inheritance and polymorphism. The IEC function blocks are seamlessly extended and the extensions are available to all engineering aspects. Object-oriented programming offers great advantages to the user, for example, when reusing existing parts of the application or when working on one application with several developers.

3.7. Online, Debugging and Commissioning Features

The code generated from the application is downloaded onto the target device with a single mouse click. Once MasterTool Xtorm is online, it offers many important functions for fast and efficient debugging, testing and commissioning.

The values of declared variables are displayed directly in the program code. These values can be changed or forced without difficulty. By setting breakpoints and then stepping through the code line by line, errors can easily be detected. Breakpoints in MasterTool Xtorm can be assigned to certain conditions to achieve even more precision in the debugging process. Using Run to Cursor feature, the user can follow the execution of the application through a complete cycle.

If the application code is modified, it is recompiled, and then loaded again without stopping the controller. Changes to several POUs (Program Organization Unit), variables or data types are also possible. This functionality is called online change. Shorter development cycles and a faster production process lead to reduced costs and increased competitiveness.

The trace is a very useful tool when the user wants to record data or even trigger events for testing or commissioning purposes. This stored data, which is completely integrated in MasterTool Xtorm, can also be used to visualize application data.

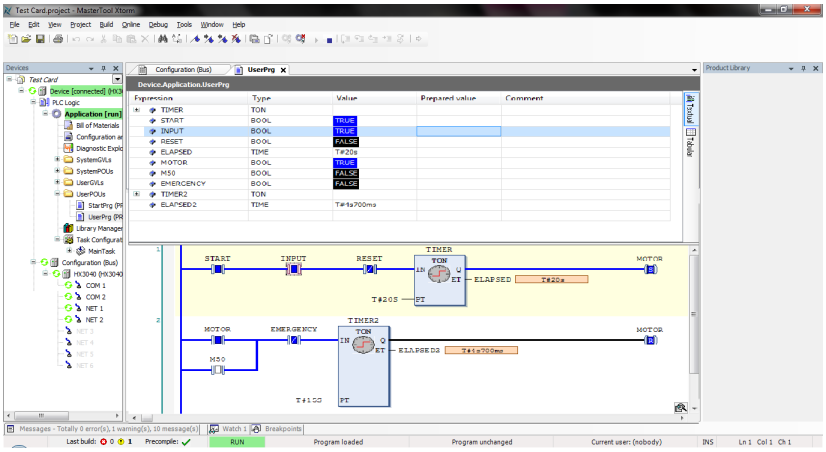

Figure 6: Online, Debugging and Commissioning Features

3.8. Simulation

The simulation tool allows the user to evaluate and test logic and algorithms. This feature enables the development and test of user applications without the need of a connected controller. It is also interesting for training, documentation and test cases evaluation. Since it is a simulator, some limitations may apply, such as not being able to test the communication interfaces, and consequently the application will be completely tested just when loaded onto the controller.

3.9. User Documentation & Help Files

Since programming and configuring a RTU according to IEC 61131-3 and IEC 61850 standards is a complex task, MasterTool Xtorm offers an extensive help page with many hints and descriptions in order to guide and serve as a first knowledge and troubleshooting base while designing the logic codes or using any software features. Besides, this help is available in different languages according to installation options.

As part of user documentation, MasterTool Xtorm can print out user application documents, such as bill of materials (BOM), POUs, bus configuration, tag and description, among other options.

3.10. Advanced Diagnostics

One of the key innovative features of Hadron Xtorm Series is the extensive support of diagnostics. This idea comes from requirements of large and complex applications, where the correct use of such information is important for maintenance, troubleshooting and to predict potential issues. This feature is also present in MasterTool Xtorm where user can access the complete diagnostics structures via watch windows and Diagnostics Explorer, when connected to an RTU.

3.11. Function Block Libraries

MasterTool Xtorm has several function block libraries ready to be used. Functions such as process variable handling, mathematic functions and timers are some examples of functions that are brought together with MasterTool Xtorm. Many libraries defined by IEC 61850 standard are also available, to be used in electric utility plants.

3.12. Docking View

The Docking View technology allows the user to customize MasterTool Xtorm environment to meet personal needs. This feature provides a user friendly interface to maximize the experience with the software tool.

3.13. Languages

MasterTool Xtorm is available in Portuguese and English languages. After installed the user interface assumes the language of the computer's operational system. The language can be changed after the installation in the Tools menu without needing to reinstall.

3.14. Minimum and Recommended Requirements for Installation and Operation

The following table shows the minimum and recommended requirements for the installation and operation of MasterTool Xtorm:

	MasterTool Xtorm
Platform	PC with Windows XP® (32 bits), Windows Vista® (32 bits) or Windows 7® (32 bits or 64 bits), Windows 8.1® (64 bits)
Processor	Intel Core 2 Duo 1,66 GHz (minimum)
Disk Usage	1 Gbyte (minimum), 2 Gbytes (recommended)
RAM	2 Gbytes (minimum), 8 Gbytes (recommended)
Resolution	1024 x 768 (recommended)
Language	Any language

Table 3: Minimum and Recommended Requirements for Installation and Operation

Notes:

Platform and RAM: Although there are computers with more than 3 Gbytes of RAM and 32 bits operational systems in the market, this entire memory could only be accessed by a 64 bits operational system. Therefore it is recommended to use a 64 bits operational system to run MasterTool Xtorm

Requirements: As a rule, PCs that fill the minimum requirements can be used for non-redundant applications. Redundant applications should use PCs that have at least the recommended settings.

4. Installation

The necessary information about MasterTool Xtorm's installation can be found at Hadron Xtorm Utilization Manual MU223600.

5. Programming

MasterTool Xtorm allows programming by making use of five different programming languages, five of them being defined by the IEC 61131-3 standard plus one additional language.

5.1. Ladder Diagram (LD)

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

The language is basically a technique that uses the logic design by using relays. Such diagrams were already used to document relay panels even before the existence of RTUs. The basic elements are usually open and closed contacts, and relay coils. For the operation, the elements must be connected in such a way that they link a vertical bar on the left, which represents a powered bus, with the right bar, which represents the ground. These diagrams were called ladder due to their format.

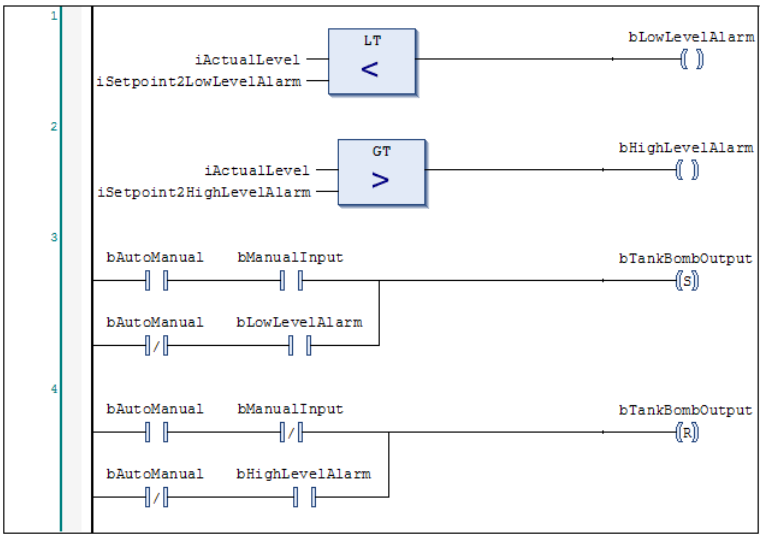

Figure 7: Ladder Diagram (LD)

5.2. Structured Text (ST)

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

The ST has common commands in structured languages such as conditional testing and selection commands, as well as distinct types of repeating loops. It also has specific commands for mathematical and logical operations without the need of function calls.

It is vital to have some knowledge of this language's syntax, since variable declarations are made through it, even when the programmer's graphic features are used. For users that do not wish to use ST language, MasterTool Xtorm offers a table for variables graphical declaration and also the auto-declaration option.

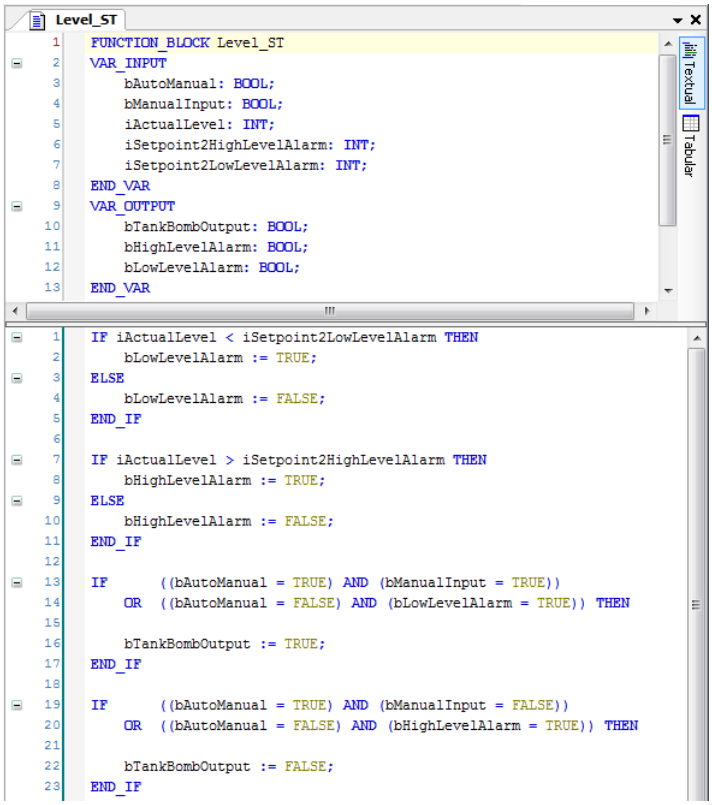

Figure 8: Structured Text (ST)

5.3. Function Block Diagram (FBD)

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

The concept of a block is any element with inputs that performs a specific processing and then writes the results of the operations to the outputs. The blocks are of two distinct types: function blocks and functions. These two types of POUs differ in that the function blocks keep the value of the local variables between the block calls, which means they must be instantiated and can run state machines, in different execution cycles. The functions perform their functionality and when the execution is finished there is no information left but the result written to the output. In general, although the functions are native of the system, they can also be implemented by the user.

An important characteristic of this language is that the processing runs left-right and top-bottom in the diagram.

The function blocks may also, as well as the functions, be used together with the Ladder diagram, increasing the opportunity of programming with this language.

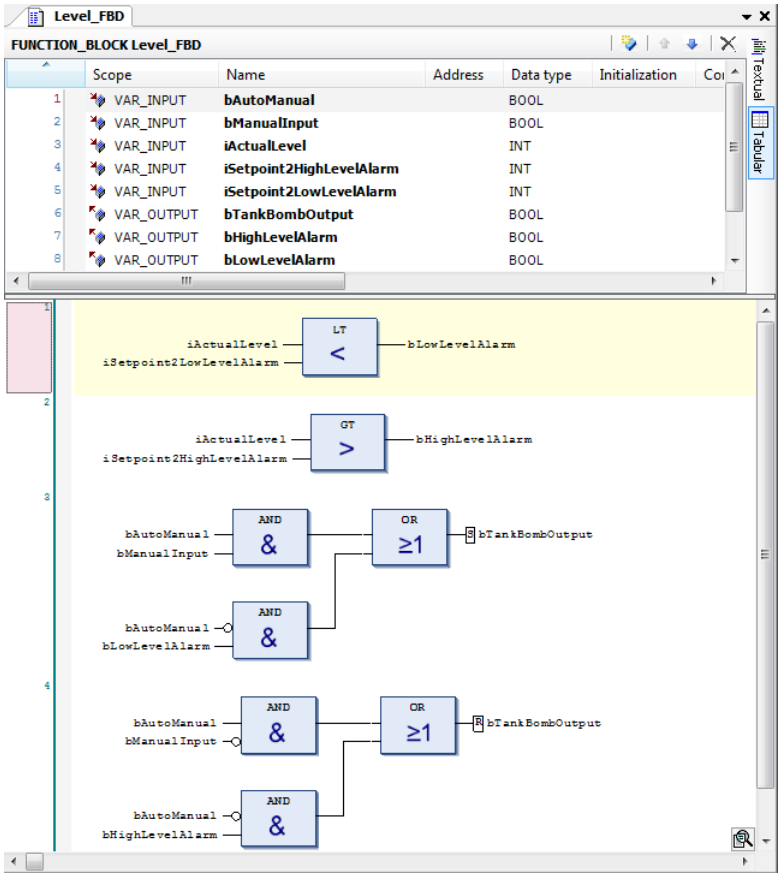

Figure 9: Function Block Diagram (FBD)

5.4. Sequential Function Chart (SFC)

The programming language Sequential Function Chart (SFC) is a graphical language based on techniques to describe sequential behavior. The European standard for this type of behavior is described in IEC 848 and is based on Petri Nets. The IEC 61131 standard introduced changes to IEC 848 in order to adjust the SFC to the other standard languages.

Therefore, this language is used to describe the sequential behavior of a system, build program structure, describe the low level of a sequential process, describe the foundations of a batch process, represent data communication and modeling systems which are event-oriented such as state machines.

The language is composed of multiple steps connected by vertical lines, being each step a state in which the program remains until the transition condition described in the connection line between steps is not satisfied.

The flow is from top to bottom, and it is possible to have a branch to return. The actions performed on each step can be continuous or event-oriented such as input or output of the state.

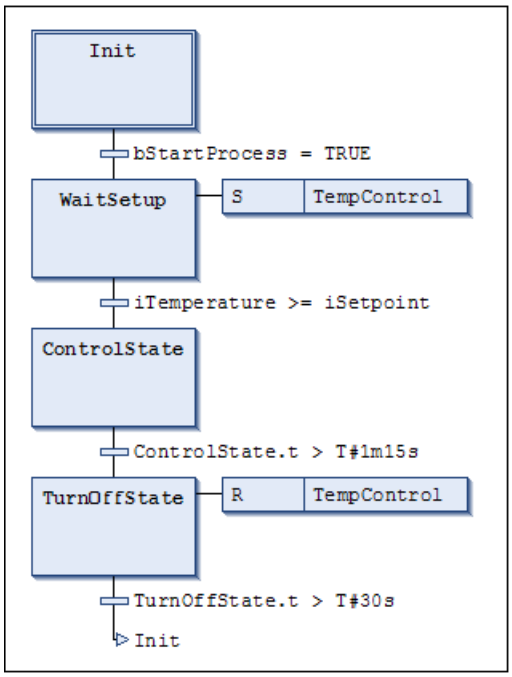

Figure 10: Sequential Function Chart (SFC)

5.5. Continuous Function Chart (CFC)

The Continuous Function Chart (CFC) programming language is a graphical language that is not described by the IEC 61131-3 standard, but is complementary to it. It resembles the FBD language, however, when a block is inserted into it, the same must be numbered. This numbering serves to indicate the diagram execution sequence, which facilitates the development and understanding of the diagram, solving this existing problem in the diagram described by the standard.

The CFC – Page Oriented – presents the same features of the regular CFC, but it separates the logics in pages, making the debug and hierarchy of the logic easier.

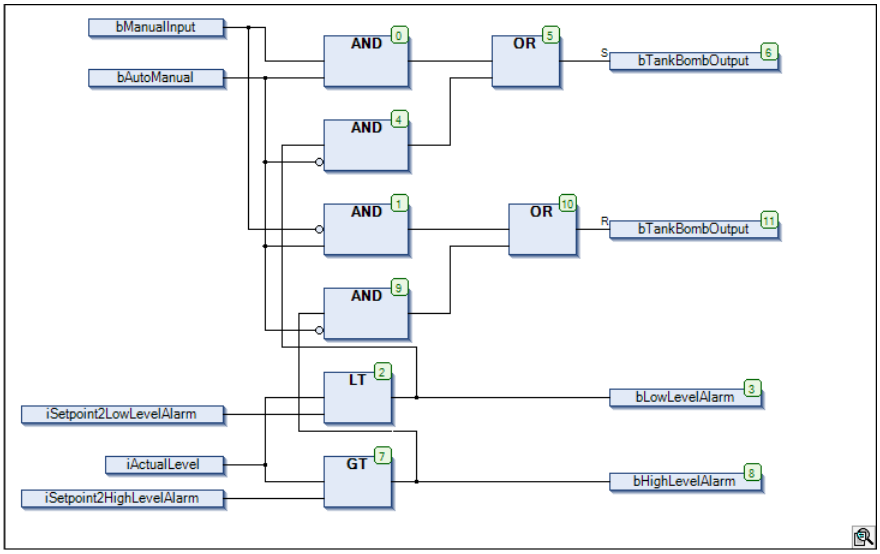

Figure 11: Continuous Function Chart (CFC)

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
MP399609	MasterTool IEC XE Programming Manual	English
MP399048	Manual de Programação MasterTool IEC XE	Portuguese
MU223600	Hadron Xtorm Utilization Manual	English
MU223000	Manual de Utilização Hadron Xtorm	Portuguese

Table 4: Related Documents