

Altivar Machine 320

Variable Speed Drives for Synchronous and Asynchronous Motors

ATV Logic Manual

04/2016



The information provided in this documentation contains general descriptions and/or technical characteristics of the performance of the products contained herein. This documentation is not intended as a substitute for and is not to be used for determining suitability or reliability of these products for specific user applications. It is the duty of any such user or integrator to perform the appropriate and complete risk analysis, evaluation and testing of the products with respect to the relevant specific application or use thereof. Neither Schneider Electric nor any of its affiliates or subsidiaries shall be responsible or liable for misuse of the information contained herein. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

No part of this document may be reproduced in any form or by any means, electronic or mechanical, including photocopying, without express written permission of Schneider Electric.

All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to help ensure compliance with documented system data, only the manufacturer should perform repairs to components.

When devices are used for applications with technical safety requirements, the relevant instructions must be followed.

Failure to use Schneider Electric software or approved software with our hardware products may result in injury, harm, or improper operating results.

Failure to observe this information can result in injury or equipment damage.

© 2016 Schneider Electric. All Rights Reserved.

Table of Contents



	Safety Information	5
	About the Book	9
Chapter 1	Introduction to the Function Blocks	11
	Global Overview of Function Blocks and ATV Logic	12
	Creating an FBD Application Using Function Blocks	15
	Function Block Configuration	16
	Display Options	18
Chapter 2	Menu Bar	19
	Tree View	20
	Common Edition Commands	21
	Execution Order Commands	21
	Check the Application	22
	Manipulating FBD Objects	23
	Preferences	24
	Configure	24
	Function BlocksBlock Password	24
Chapter 3	Function Blocks Language Elements	25
3.1	Special Input Functions	26
	NUM (analog)	27
	NUM (register)	27
	TRUE	27
	FALSE	27
3.2	Logic Functions	28
	Introduction	29
	NOT Function	29
	The AND Function	30
	The OR Function	31
	The NAND Function	32
	The NOR Function	33
	The XOR (eXclusive OR) Function	34
	The A/C TIMER (Timer) Function	35
	The ADD/SUB Function	37
	The BOOLEAN Function	38
	The CAN Function (Word to Bit Conversion)	39
	The CNA Function (Bit to Word Conversion)	40
	The COMPARE (Comparison of two values)	41
	The EDGE Function	42
	The GAIN Function	43
	The MUL/DIV Arithmetic Function	44
	The MUX Function (Multiplexing)	45
	The BIT READ Function	46
	The READ Ana Param Function	46
	The READ Reg Param Function	47
	The RS Memory Function	47
	The SHIFT/ROLL Function	48
	The TRIGGER function (Schmitt Trigger)	49
	The COUNT Up/Down Counter Function	50
	The BIT WRITE Function (WriteBitParam)	51

	The WRITE Ana Param function	52
	The WRITE Reg Param function	53
	The STU Function (16-bit analog TO 16-bit register)	53
	The UTS Function (16-Bit Register Input to 16-Bit Analog Output)	54
Chapter 4	Description of ATV320 Parameters	55
	Function Block Status	56
	Function Block Command	57
	Function Block Fault	58
	Function Block Parameters	58
Chapter 5	Communication Parameters	59
	ADL Containers	60
	FB SYSTEM WORDS	61
Chapter 6	Application Example	63
	Windowpane Application Example	63



Important Information

NOTICE

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, service, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.



The addition of this symbol to a “Danger” or “Warning” safety label indicates that an electrical hazard exists which will result in personal injury if the instructions are not followed.



This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

DANGER

DANGER indicates a hazardous situation which, if not avoided, **will result in** death or serious injury.

WARNING

WARNING indicates a hazardous situation which, if not avoided, **could result in** death or serious injury.

CAUTION

CAUTION indicates a hazardous situation which, if not avoided, **could result in** minor or moderate injury.

NOTICE

NOTICE is used to address practices not related to physical injury.

PLEASE NOTE

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and its installation, and has received safety training to recognize and avoid the hazards involved.

Qualification Of Personnel

Only appropriately trained persons who are familiar with and understand the contents of this manual and all other pertinent product documentation are authorized to work on and with this product. In addition, these persons must have received safety training to recognize and avoid hazards involved. These persons must have sufficient technical training, knowledge and experience and be able to foresee and detect potential hazards that may be caused by using the product, by changing the settings and by the mechanical, electrical and electronic equipment of the entire system in which the product is used. All persons working on and with the product must be fully familiar with all applicable standards, directives, and accident prevention regulations when performing such work.

Intended Use

This product is a drive for three-phase synchronous and asynchronous motors and intended for industrial use according to this manual. The product may only be used in compliance with all applicable safety regulations and directives, the specified requirements and the technical data. Prior to using the product, you must perform a risk assessment in view of the planned application. Based on the results, the appropriate safety measures must be implemented. Since the product is used as a component in an entire system, you must ensure the safety of persons by means of the design of this entire system (for example, machine design). Any use other than the use explicitly permitted is prohibited and can result in hazards. Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel.

BEFORE YOU BEGIN

Do not use this product on machinery lacking effective point-of-operation guarding. Lack of effective point-of-operation guarding on a machine can result in serious injury to the operator of that machine.

 WARNING
UNGUARDED EQUIPMENT
<ul style="list-style-type: none">• Do not use this software and related automation equipment on equipment which does not have point-of-operation protection.• Do not reach into machinery during operation.
Failure to follow these instructions can result in death, serious injury, or equipment damage.

This automation equipment and related software is used to control a variety of industrial processes. The type or model of automation equipment suitable for each application will vary depending on factors such as the control function required, degree of protection required, production methods, unusual conditions, government regulations, etc. In some applications, more than one processor may be required, as when backup redundancy is needed.

Only you, the user, machine builder or system integrator can be aware of all the conditions and factors present during setup, operation, and maintenance of the machine and, therefore, can determine the automation equipment and the related safeties and interlocks which can be properly used. When selecting automation and control equipment and related software for a particular application, you should refer to the applicable local and national standards and regulations. The National Safety Council's Accident Prevention Manual (nationally recognized in the United States of America) also provides much useful information.

In some applications, such as packaging machinery, additional operator protection such as point-of-operation guarding must be provided. This is necessary if the operator's hands and other parts of the body are free to enter the pinch points or other hazardous areas and serious injury can occur. Software products alone cannot protect an operator from injury. For this reason the software cannot be substituted for or take the place of point-of-operation protection.

Ensure that appropriate safeties and mechanical/electrical interlocks related to point-of-operation protection have been installed and are operational before placing the equipment into service. All interlocks and safeties related to point-of-operation protection must be coordinated with the related automation equipment and software programming.

NOTE: Coordination of safeties and mechanical/electrical interlocks for point-of-operation protection is outside the scope of the Function Block Library, System User Guide, or other implementation referenced in this documentation.

START-UP AND TEST

Before using electrical control and automation equipment for regular operation after installation, the system should be given a start-up test by qualified personnel to verify correct operation of the equipment. It is important that arrangements for such a check be made and that enough time is allowed to perform complete and satisfactory testing.

WARNING

EQUIPMENT OPERATION HAZARD

- Verify that all installation and set up procedures have been completed.
- Before operational tests are performed, remove all blocks or other temporary holding means used for shipment from all component devices.
- Remove tools, meters, and debris from equipment.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Follow all start-up tests recommended in the equipment documentation. Store all equipment documentation for future references.

Software testing must be done in both simulated and real environments.

Verify that the completed system is free from all short circuits and temporary grounds that are not installed according to local regulations (according to the National Electrical Code in the U.S.A, for instance). If high-potential voltage testing is necessary, follow recommendations in equipment documentation to prevent accidental equipment damage.

Before energizing equipment:

- Remove tools, meters, and debris from equipment.
- Close the equipment enclosure door.
- Remove all temporary grounds from incoming power lines.
- Perform all start-up tests recommended by the manufacturer.

OPERATION AND ADJUSTMENTS

The following precautions are from the NEMA Standards Publication ICS 7.1-1995 (English version prevails):

- Regardless of the care exercised in the design and manufacture of equipment or in the selection and ratings of components, there are hazards that can be encountered if such equipment is improperly operated.
- It is sometimes possible to misadjust the equipment and thus produce unsatisfactory or unsafe operation. Always use the manufacturer's instructions as a guide for functional adjustments. Personnel who have access to these adjustments should be familiar with the equipment manufacturer's instructions and the machinery used with the electrical equipment.
- Only those operational adjustments actually required by the operator should be accessible to the operator. Access to other controls should be restricted to prevent unauthorized changes in operating characteristics.

⚠ WARNING

LOSS OF CONTROL

- The designer of any control scheme must consider the potential failure modes of control paths and, for critical control functions, provide a means to achieve a safe state during and after a path failure. Examples of critical control functions are emergency stop, overtravel stop, power outage and restart.
- Separate or redundant control paths must be provided for critical control functions.
- System control paths may include communication links. Consideration must be given to the implications of unanticipated transmission delays or failures of the link.
- Observe all accident prevention regulations and local safety guidelines (1).
- Each implementation of the product must be individually and thoroughly tested for proper operation before being placed into service.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

(1) For USA: Additional information, refer to NEMA ICS 1.1 (latest edition), Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control and to NEMA ICS 7.1 (latest edition), Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems.

About the Book



At a Glance

Document Scope

The purpose of the document is to describe how to use the ATV logic function for ATV320

The ATV logic functions are only accessible inside the Altivar Machine 320 DTM.

FDT/DTM (Field Device Tool / Device Type Manager) is a new technology chosen by several companies in automation.

To install the Altivar Machine 320 DTM, you can download and install our FDT: SoMove from www.schneider-electric.com. It is including the Altivar Machine 320 DTM.

The content of this manual is also accessible through the ATV320 DTM online help.

Validity Note

The technical characteristics of the devices described in this document also appear online. To access this information online:

Step	Action
1	Go to the Schneider Electric home page www.schneider-electric.com .
2	In the Search box type the reference of a product or the name of a product range. <ul style="list-style-type: none">• Do not include blank spaces in the reference or product range.• To get information on grouping similar modules, use asterisks (*).
3	If you entered a reference, go to the Product Datasheets search results and click on the reference that interests you. If you entered the name of a product range, go to the Product Ranges search results and click on the product range that interests you.
4	If more than one reference appears in the Products search results, click on the reference that interests you.
5	Depending on the size of your screen, you may need to scroll down to see the data sheet.
6	To save or print a data sheet as a .pdf file, click Download XXX product datasheet .

The characteristics that are presented in this manual should be the same as those characteristics that appear online. In line with our policy of constant improvement, we may revise content over time to improve clarity and accuracy. If you see a difference between the manual and online information, use the online information as your reference.

Related Documents

Use your tablet or your PC to quickly access detailed and comprehensive information on all our products on www.schneider-electric.com

The internet site provides the information you need for products and solutions

- The whole catalog for detailed characteristics and selection guides
- The CAD files to help design your installation, available in over 20 different file formats
- All software and firmware to maintain your installation up to date
- A large quantity of White Papers, Environment documents, Application solutions, Specifications... to gain a better understanding of our electrical systems and equipment or automation
- And finally all the User Guides related to your drive, listed below:

Title of Documentation	Reference Number
Altivar 320 Getting Started	NVE21763 (English) , NVE21771 (French) , NVE21772 (German) , NVE21773 (Spanish) , NVE21774 (Italian) , NVE21776 (Chinese)
Altivar 320 Getting Started Annex (SCCR)	NVE21777 (English)
Altivar 320 Installation Manual	NVE41289 (English) , NVE41290 (French) , NVE41291 (German) , NVE41292 (Spanish) , NVE41293 (Italian) , NVE41294 (Chinese)
Altivar 320 Programming manual	NVE41295 (English) , NVE41296 (French) , NVE41297 (German) , NVE41298 (Spanish) , NVE41299 (Italian) , NVE41300 (Chinese)
Altivar 320 Modbus Serial Link manual	NVE41308 (English)
Altivar 320 Ethernet IP/Modbus TCP manual	NVE41313 (English)
Altivar 320 PROFIBUS DP manual (VW3A3607)	NVE41310 (English)
Altivar 320 DeviceNet manual (VW3A3609)	NVE41314 (English)
Altivar 320 CANopen manual (VW3A3608, 618, 628)	NVE41309 (English)
Altivar 320 POWERLINK Manual - VW3A3619	NVE41312 (English)
Altivar 320 EtherCAT manual - VW3A3601	NVE41315 (English)
Altivar 320 Communication Parameters	NVE41316 (English)
Altivar 320 Safety Functions manual	XXX (English), XXX (French), XXX (German), XXX (Spanish), XXX (Italian), XXX (Chinese)

You can download these technical publications and other technical information from our website at <http://www.schneider-electric.com/ww/en/download>

Terminology

The technical terms, terminology, and the corresponding descriptions in this manual normally use the terms or definitions in the relevant standards.

In the area of drive systems this includes, but is not limited to, terms such as **error**, **error message**, **failure**, **fault**, **fault reset**, **protection**, **safe state**, **safety function**, **warning**, **warning message**, and so on.

Among others, these standards include:

- IEC 61800 series: Adjustable speed electrical power drive systems
- IEC 61508 Ed.2 series: Functional safety of electrical/electronic/programmable electronic safety-related
- EN 954-1 Safety of machinery - Safety related parts of control systems
- EN ISO 13849-1 & 2 Safety of machinery - Safety related parts of control systems.
- IEC 61158 series: Industrial communication networks - Fieldbus specifications
- IEC 61784 series: Industrial communication networks - Profiles
- IEC 60204-1: Safety of machinery - Electrical equipment of machines – Part 1: General requirements

In addition, the term **zone of operation** is used in conjunction with the description of specific hazards, and is defined as it is for a **hazard zone** or **danger zone** in the EC Machinery Directive (2006/42/EC) and in ISO 12100-1.

Also see the glossary at the end of this manual.

Chapter 1

Introduction to the Function Blocks

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Global Overview of Function Blocks and ATV Logic	12
Creating an FBD Application Using Function Blocks	15
Function Block Configuration	16
Display Options	18

Global Overview of Function Blocks and ATV Logic

Description

Programmable logic functions allow you to write your own application for the ATV320 variable speed drive.

Programmable logic functions interact with:

- The drive core control
- The I/O, the communication functions
- The HMI via M0xx (local or external graphic keypad)

It is based on an FBD instruction set: Boolean and transfer operations, arithmetic operations, timers, counters, etc. ATV Logic is seen by the drive as a terminal.

The program in the edit window corresponds to:

- The compiled program
- The program transferred into the ATV320

To create an FBD program, insert various function blocks in the diagram, and link them together.

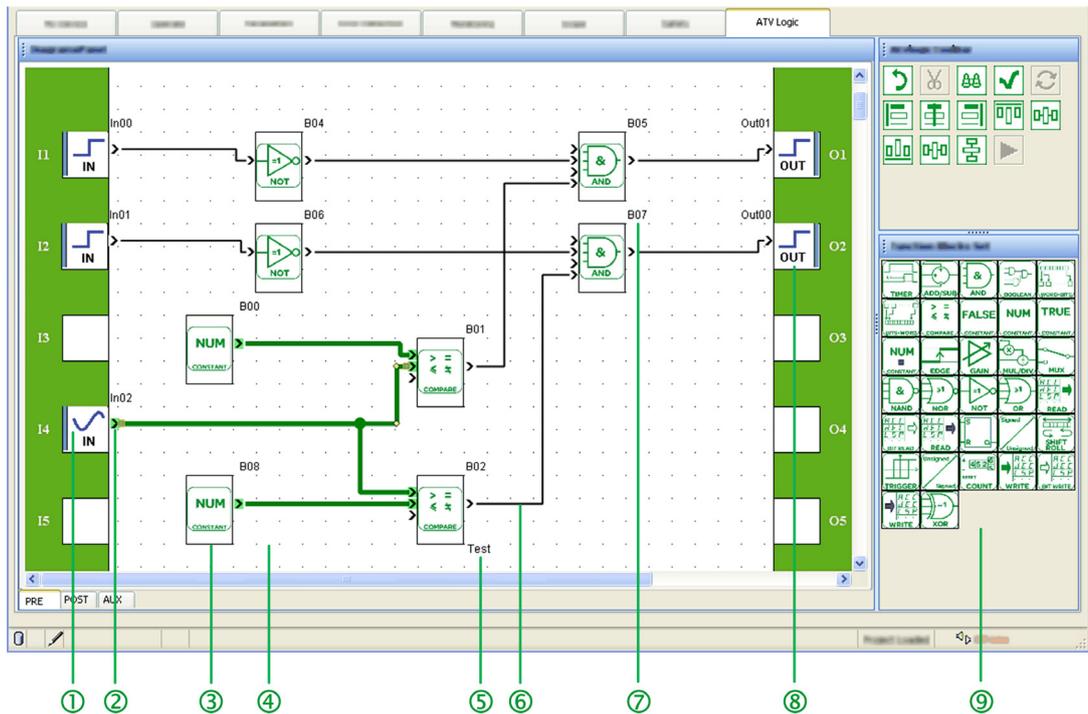
All types of function block can be placed on the sheet, except for inputs and outputs.

The ATV Logic tab of SoMove DTM consists of a workspace with 3 layers that can be selected via tabs (PRE, POST, AUX)

- The diagram, where the functions that make up the program are inserted
- The Input pins on the left of the diagram where the inputs are positioned
- The Output pins on the right of the diagram where the outputs are positioned

Each area contains the FBD edit panel itself and configurable ports, symbolically represented on both sides of the edit panel by terminal pins.

ATV Logic panel



Legend:

- ① : Virtual input pin, ② : Connector, ③ : Function block unit, ④ : Diagram, ⑤ : Comments, ⑥ , Connection between blocks, ⑦ : Function block number, ⑧ : Virtual output pin, ⑨ : Tool panels: Edit, Toolbox

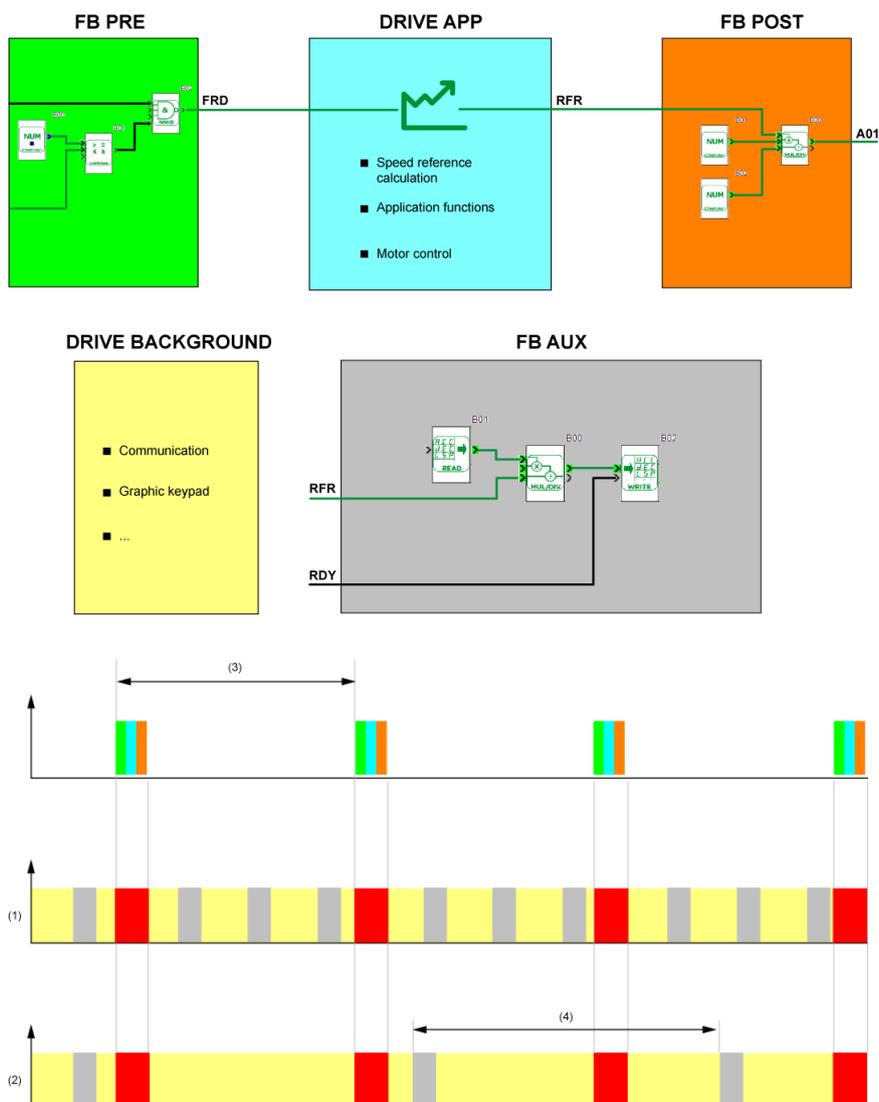
Interaction of the Function Block Tasks with the Drive Tasks

ATV Logic provides graphic programming based on predefined function blocks.

In Function Blocks/ATV Logic, there are three types of task: PRE, POST and AUX.

The rule for using the task is:

- In PRE/POST tasks: Connection to Fast access parameters only
- In the AUX task: Connection to Slow and Fast access parameters



(1) Drive Background TASK < 1 ms (2) Drive Background TASK > 1 ms

(3) 2 ms (4) t > period duration AUX = FBE fault error TOAU

The FB PRE and POST tasks are executed in the **Drive Application Task** which is itself executed every 2 ms. The **Drive Application Task** duration is less than 2 ms. This leaves between 500 μs and 1 ms to carry out the **Drive Application Task**. However, the **Drive Background Task** may be interrupted each time the **Drive Application Task** has to execute.

The **Drive Background Task** may be loaded and therefore executed several times between two **Drive Application Tasks**. (See Drive Background TASK < 1 ms)

On the other hand, if the **Drive Background Task** is under a heavy load, it will be frequently interrupted by the **Drive Application Task**. (See Drive Background Task > 1 ms)

Since the AUX task execution time can be interrupted by the **Drive Application Task**, it is not possible to determine whether its duration is fixed or less than a predefined level.

However the Period duration parameter helps to ensure that the AUX task is refreshed sufficiently often for the application.

NOTE: If Run and Stop commands are managed in the AUX task, their activation can be delayed due to Pre and Post task priorities.

Groups of parameters

Tabs PRE and POST are priority to the tabs AUX. 4 groups of parameters are defining:

- **FastReadFunctionBLocks**
ETA, RFR, FRH, LCR, OTR, ETI, ULN, UOP, THD, OPR
THR1, THR2, THR3
LRS1, LRS2, LRS3, LRS4, LRS5, LRS6, LRS7, LRS8
IL1I, IL1R, OL1R, AI1C, AI2C, AI3C, AO1R, AO1C
RFRD, FRHD, SPD
M001, M002, M003, M004, M005, M006, M007, M008
S001, SLSS, STOS, SS1S
- **FastWriteFunctionBLocks**
M001, M002, M003, M004, M005, M006, M007, M008, S006
- **SlowReadFunctionBLocks**
HSP, LSP, BSP, ETA, RFR, FRH, LCR, OTR, ETI, ULN, UOP, THD, OPR
THR1, THR2, THR3, APH, RTH, PTH, UNT
LRS1, LRS2, LRS3, LRS4, LRS5, LRS6, LRS7, LRS8
IL1I, IL1R, OL1R, AI1C, AI2C, AI3C, AO1R, AO1C
RFRD, FRHD, ACC, DEC, AC2, DE2, INR, FRO
JPF, JF2, JF3, JFH
SP2, SP3, SP4, SP5, SP6, SP7, SP8
SP9, SP10, SP11, SP12, SP13, SP14, SP15, SP16
RPI, RP2, RP3, RP4, RPG, RIG, RDG, RPE, RPF, RPC, RPO, SPD
M001, M002, M003, M004, M005, M006, M007, M008
S001, SLSS, STOS, SS1S
- **SlowWriteFunctionBLocks**
HSP, LSP, BSP, RPR, ACC, DEC, AC2, DE2, INR, JPF
JF2, JF3, JFH, SP2, SP3, SP4, SP5, SP6, SP7, SP8
SP9, SP10, SP11, SP12, SP13, SP14, SP15, SP16, RPI, RP2
RP3, RP4, RPG, RIG, RDG, M001, M002, M003, M004, M005
M006, M007, M008, S006

Creating an FBD Application Using Function Blocks

Insertion of Function Blocks

To add a function block, left-click and drag the selected block to the desired location on the diagram.

Creation of a Virtual Port

To add a function block, drag the function block you wish to add to the diagram.



① : Virtual input pins

② : Virtual output pins

Configuration of Virtual Ports

The type of input/output pins can be configured by double-clicking on them.

- Logic input 1 assignment (IL01) to Logic input 10 assignment (IL10): Logic inputs
- Analog input 1 assignment (IA01) to Analog input 10 assignment (IA10): Analog inputs
- Logic output 1 assignment (OL01) to Logic output 10 assignment (OL10): Logic outputs
- Analog output 1 assignment (OA01) to Analog output 10 assignment (OA10): Analog outputs

OAXX must be in the range [0 - 8192] if used on OA1. Other values generate the OA1 max. value.

When you double-click on an input/output, only this port can be changed; the others are grayed-out.

Links Between Blocks

The following process describes how to link function blocks together:

Step	Action	Comment
1	Left -click on an output of a function block output.	Result: The mouse cursor is displayed as a cross.
2	Hold down the left button.	
3	With the button held down, move the cursor over a block output.	
4	Release the mouse button.	Result: The line is shown between the two linked blocks
5	Repeat steps 1 and 2 to link all the blocks.	

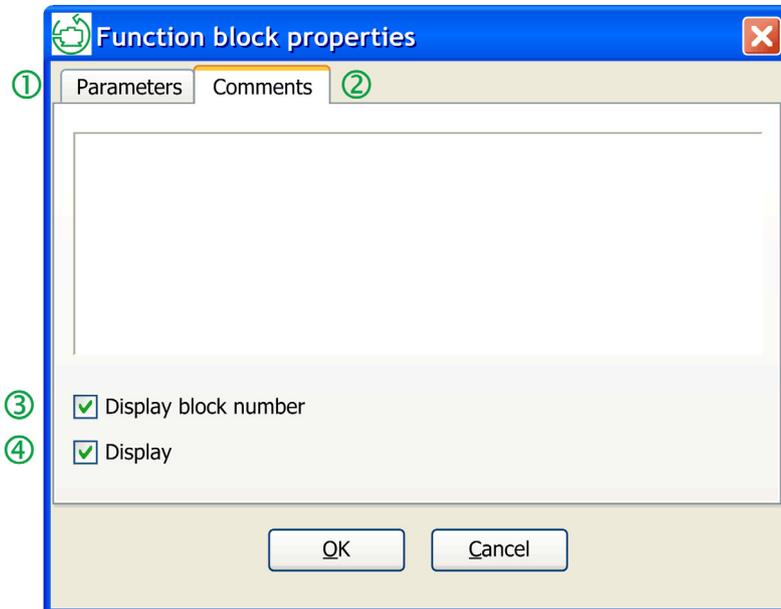
Function Block Configuration

The Configuration Window

Each function block has a configuration window. This window consists of 2 tabs:

- Comments for all function blocks
- Parameters that depend on the function block type

When you double-click on a block, a configuration window appears.



Legend:

① : **Parameters** tab: Most function blocks have a **Parameters** tab. In this tab, you need to set the function block's specific parameters. These parameters are described in detail for each of the blocks.

② : **Comments** tab: All the blocks can have an associated comment. These comments are displayed under the block in the diagram.

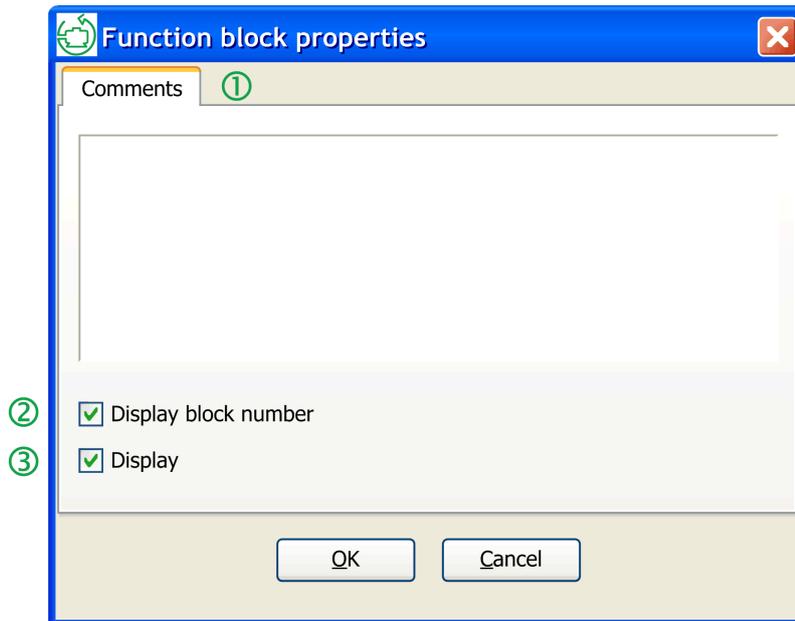
③ : **Display block number**: Displays the block number. This option is activated by default.

④ : **Display** comment: When a comment has been added to a function block, the comment is displayed if the **Display** comment box is checked. This option is activated by default.

Configuration: Virtual Input or Output

Each diagram input or output has a configuration window.

By double-clicking on the start point or end point, a configuration window appears.



Legend:

- ① : **Comments**tab: These comments are displayed under the block in the diagram.
- ② : **Display block number**: Displays the block number. This option is activated by default.
- ③ : **Display** comment: When a comment has been added to a function block, the comment is displayed if the **Display** comment box is checked. This option is activated by default.

Type of Link

Table with three columns

Link	Description	Possible value
	For a Boolean link, it is a Black line.	1 or 0
	For an analog link, it is a Green line.	- 32768 to 32767
	For a register link, it is a Blue line.	0 to 65535

Display Options

Function Block Set

To create an FBD program, various different functions are available in the function block set. Each picture in the function block set represents a function type.

When the mouse is moved over one of the function types, the name of the function appears.

Diagram Dimensions

It is possible to change the diagram dimensions.

Resize the diagram by clicking on the right-hand edge and dragging it to the left or right.

Chapter 2

Menu Bar

What Is in This Chapter?

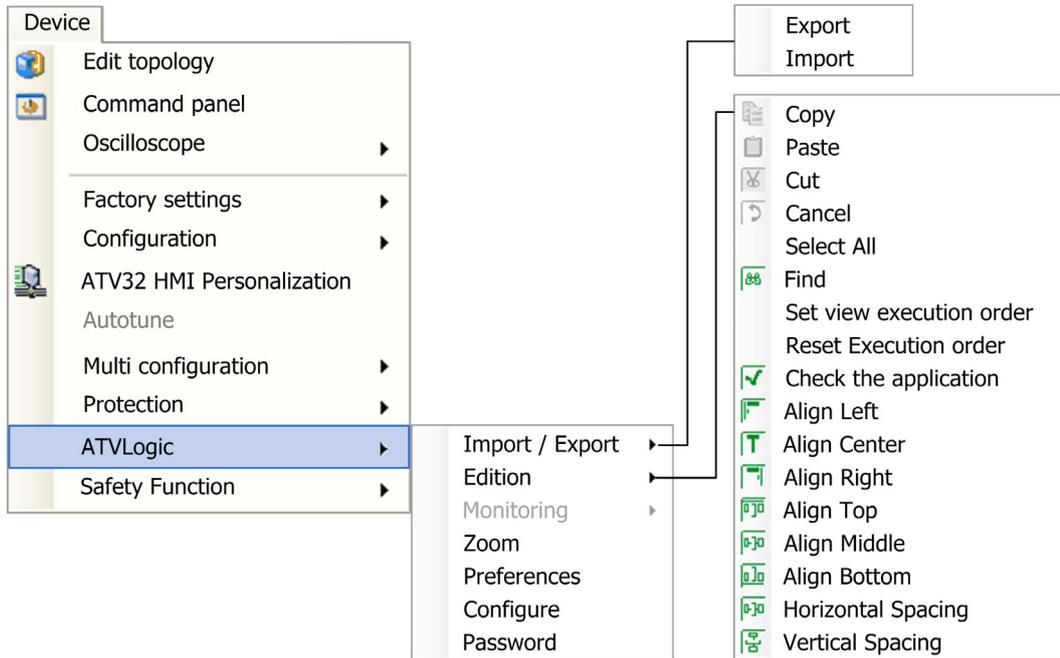
This chapter contains the following topics:

Topic	Page
Tree View	20
Common Edition Commands	21
Execution Order Commands	21
Check the Application	22
Manipulating FBD Objects	23
Preferences	24
Configure	24
Function BlocksBlock Password	24

Tree View

Menu Bar Overview

For Logic functions, several different options are available:



Common Edition Commands

Commands

- **Copy Paste Cut:** This allows you to copy, paste or cut one or more selected objects.
- **Cancel :** This allows you to cancel the last action performed.

Working with Blocks

- **How to select an isolated block:**
By left-clicking the mouse on an isolated function block.
- **How to select more objects:**
To select several adjacent objects, define a selection zone by drawing a frame around the objects to be selected.
To select several objects scattered about in the diagram, press the Ctrl key, then click on the objects to be selected while continuing to hold down the Ctrl key.
- **How to deselect one or more blocks:**
By clicking on any free space in the diagram or pressing the Escape key, all the blocks will be deselected.
To deselect one block in the selection, press the Ctrl key and click on the object you want to deselect.
- **Find:**
The find feature allows you to search in: comments, function blocks, block numbers.
The search process will search in the names and the comment associated with each FB.
Once searching has started, when a block meets the search criteria, it will be marked in blue.
To see any other blocks that meet the criteria, click again to find the next one.

Execution Order Commands

Set / View execution order

This feature determines the order of execution of the function blocks.

This action can reorganize the order of execution of function blocks and optimize cycle times.

This function can be accessed via the menu bar: **ATVLogic → Edition → Set view execution order.**

Reset execution order

This feature resets the order of execution of the function blocks.

This function can be accessed via the menu bar: **ATVLogic → Edition → Reset Execution order.**

Check the Application

Description

With the **Device** → **ATVLogic** → **Edition** → **Check the compilation** command, you launch the compilation (check) of the program.

The result of compilation is displayed in the compilation report window.

In this window, the following information appears:

- The result of the program check
- Used and available resources

NOTE: This check must be performed before the program can be downloaded.

Results Window Elements

The compiler calculates the volumes of resources used in the drive.

If the values calculated are greater than the available values, they appear in the compilation report window.

The compilation report window displays the various possible errors revealed during compilation.

List of errors that can occur during compilation

Unknown Error

Aux task duration too long

The virtual port ID of the block {0} in the {1} task is not allowed

Binary file not created. Error during the file creation

Error during the generation of the {0} task

Error during the Header generation

Error during the Signals generation

The Catalog isn't compatible with the Compiler version

Binary too large

Memory overflow

Virtual Input present in the {0} task, but no one were connected

Virtual Output present in the {0} task, but no one were connected

Error Pre and Post tasks are too long

Two or more Function Blocks have the same Execution Order in the {0} task

Internal error : {0}

Internal error : {0}, block {1}

List of warnings that can occur during compilation

Unknown Warning

The Function Block {0} in the {1} task has no Input port connected

A parameter of the Function Block {0} in the {1} task has no value

At least one of the output ports of the block {0} from the {1} task is not connected

The Virtual Input {0} is used in different tasks. The value could be modified during the other tasks execution

The Virtual Output {0} is used in different tasks. The value could be overridden during the other tasks execution

Manipulating FBD Objects

How to Align a Group of Objects

Select the objects to be aligned.

Result: All the selected objects are highlighted by small yellow squares at each corner of the block.

From the **atvlogicToolBar** toolbox, select:

Icon	Title	Description
	Align left	Aligns the group of blocks vertically to the left
	Align center	Aligns the group of blocks vertically to the center
	Align right	Aligns the group of blocks vertically to the right
	Align top	Aligns the group of blocks vertically to the top
	Align middle	Aligns the group of blocks vertically to the middle
	Align bottom	Aligns the group of blocks vertically to the bottom

How to Center a Group of Objects

Select the objects to center.

Result: All the selected objects are highlighted by small yellow squares at each corner of the block.

From the **atvlogicToolBar** toolbox, select:

Icon	Title	Description
	Horizontal spacing	Aligns the group of blocks vertically to the left
	Vertical spacing	Aligns the group of blocks vertically to the center

Preferences

Access

This function can be accessed via the menu bar: **Device** → **ATVLogic** → **Preferences**.

Commands

This feature edits:

- Diagrams
 - Display or hide the grid
 - Choose the size of spaces
 - Choose the zoom
- Functions

All function blocks can have an associated comment and block number.
You can choose to display:

 - All comments
 - All numbers

Configure

Access

This function can be accessed via the menu bar: **Device** → **ATVLogic** → **Configure**.

This window has two tabs: **Application Information** and **Tasks**.

Application Information tab

This tab lets you enter information relating to the program:

- The name of the Programmer
- The version
- The date
- Add a comment

Task tab

This tab specifies the maximum time allowed for the AUX task to be performed.

This is similar to a WATCHDOG action (control of the module cycle time).

Function Blocks Block Password

Access

This function can be accessed via the menu bar: **Device** → **ATVLogic** → **Password**.

Description

The purpose of the password is to help protect your program and prevent others from reading it.

The password can protect the programmer's intellectual property but in no way prevent the configuration being uploaded.

NOTE: To disable the password, enter 0000. If an incorrect password entered is three times, the program is deleted, and a blank program is available for editing.

Chapter 3

Function Blocks Language Elements

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
3.1	Special Input Functions	26
3.2	Logic Functions	28

Section 3.1

Special Input Functions

What Is in This Section?

This section contains the following topics:

Topic	Page
NUM (analog)	27
NUM (register)	27
TRUE	27
FALSE	27

NUM (analog)

Description



The numerical constant NUM is an analog integer with a value between -32768 and +32767. The value of the constant can be set in the Parameters window.

NUM (register)

Description



The numerical constant NUM is a register integer with a value between 0 and 65535. The value of the constant can be set in the Parameters window.

TRUE

Description



This constant can be used to set the function inputs to 1.

FALSE

Description



This constant can be used to set the function inputs to 0.

Section 3.2

Logic Functions

What Is in This Section?

This section contains the following topics:

Topic	Page
Introduction	29
NOT Function	29
The AND Function	30
The OR Function	31
The NAND Function	32
The NOR Function	33
The XOR (eXclusive OR) Function	34
The A/C TIMER (Timer) Function	35
The ADD/SUB Function	37
The BOOLEAN Function	38
The CAN Function (Word to Bit Conversion)	39
The CNA Function (Bit to Word Conversion)	40
The COMPARE (Comparison of two values)	41
The EDGE Function	42
The GAIN Function	43
The MUL/DIV Arithmetic Function	44
The MUX Function (Multiplexing)	45
The BIT READ Function	46
The READ Ana Param Function	46
The READ Reg Param Function	47
The RS Memory Function	47
The SHIFT/ROLL Function	48
The TRIGGER function (Schmitt Trigger)	49
The COUNT Up/Down Counter Function	50
The BIT WRITE Function (WriteBitParam)	51
The WRITE Ana Param function	52
The WRITE Reg Param function	53
The STU Function (16-bit analog TO 16-bit register)	53
The UTS Function (16-Bit Register Input to 16-Bit Analog Output)	54

Introduction

Logic Function

In FBD language, it is possible to use logic functions in the block diagrams.

These functions can be accessed from the Function block set.

If one or more inputs are not connected, their state is not taken into account when calculating the number of functions.

However, Enable inputs are always activated.

NOT Function

Description



If the input is inactive or not connected, the output is active.

If the input is active, the output is inactive.

- [IN]: 1 Boolean input
- [Q]: 1 Boolean output

Table with two columns

In	Q
0	1
1	0

The AND Function

Description



If all the inputs are active or not connected, the output is active. If at least one input is connected, the output is inactive.

- [IN1] to [IN4]: 4 Boolean inputs
- [Q]: 1 Boolean output

In1	In2	In3	In4	Q
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

The OR Function

Description



If at least one input is active, the output is active. If all the inputs are inactive or not connected, the output is inactive.

- [IN1] to [IN4]: 4 Boolean inputs
- [Q]: 1 Boolean output

In1	In2	In3	In4	Q
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

The NAND Function

Description



If at least one input is inactive, the output is active. If all the inputs are active or not connected, the output is inactive.

- [IN1] to [IN4]: 4 Boolean inputs
- [Q]: 1 Boolean output

In1	In2	In3	In4	Q
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

The NOR Function

Description



If all the inputs are inactive or not connected, the output is active. If at least one input is active, the output is inactive.

- [IN1] to [IN4]: 4 Boolean inputs
- [Q]: 1 Boolean output

In1	In2	In3	In4	Q
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

The XOR (eXclusive OR) Function

Description



If one input is inactive and the other input is active or not connected, the output is active. If both inputs are active or inactive, or not connected, the output is inactive.

- [IN1] and [IN2]: 2 Boolean inputs
- [Q]: 1 Boolean output

In1	In2	Q
0	0	0
0	1	1
1	0	1
1	1	0

The A/C TIMER (Timer) Function

Description



The Timer function is used to delay, prolong, and control actions over a predefined time.

The Timer has three functions:

- A function: Timer on-delay, or timer active
- C function: Timer off-delay, or timer idle
- A/C function: Combination of A and C functions

Inputs/Outputs

- [IN]: 1 Boolean input [Q]: 1 Boolean output

Configuration

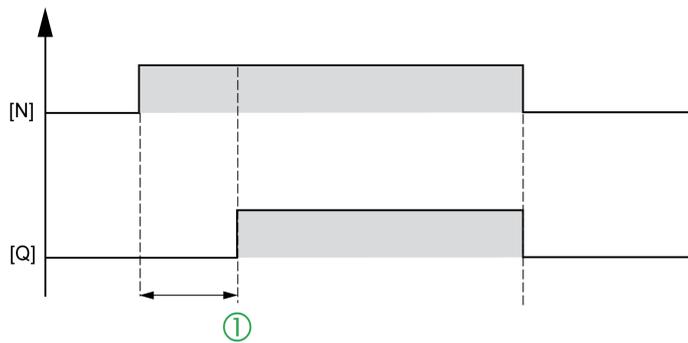
You can adjust the parameters from the Configuration window.

In the Parameters tab, you can adjust the value of the delays for each of the functions (A, C and A/C):

- ON delay for function A
- OFF delay for function C
- A combination of both the ON and OFF delays can be used to adjust function A/C.

A Function

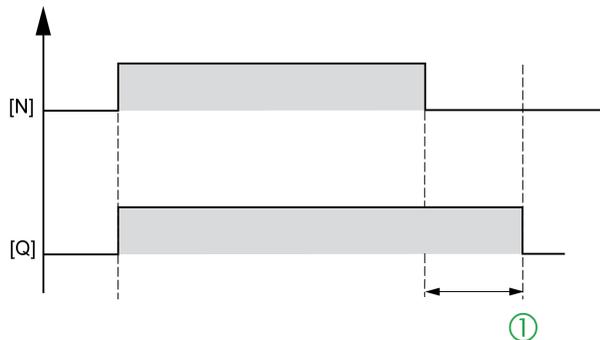
The following diagram shows the operation of the timer in function A.



① : ON delay

C Function

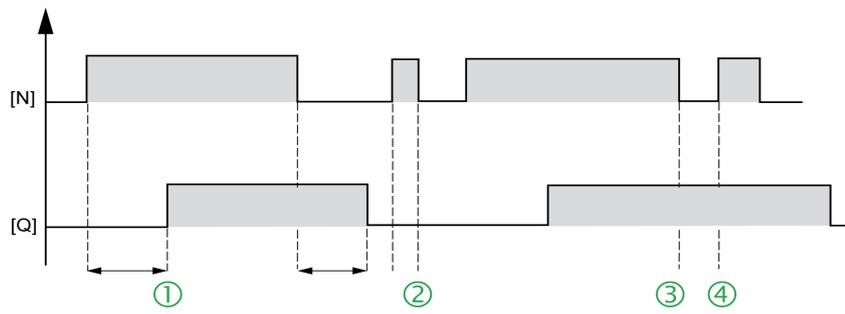
The following diagram shows the operation of the timer in function C.



① : OFF delay

A/C Function

The following diagram shows the operation of the timer in function A/C.



① : ON delay, ② : OFF delay, ③ : $t < \text{ON delay}$, ④ : $t < \text{OFF delay}$

The ADD/SUB Function

Description



The ADD-SUB arithmetic function is used to perform simple operations on integers:

- Addition
- Subtraction

Calculation formula

CALCULATION OUTPUT [Q]= [A] + [B] - [C]

Inputs/Outputs

Description of the inputs:

- [A]: An analog input
- [B]: An analog input
- [C]: An analog input

NOTE: If the inputs are not connected, they are set to 0. It is not possible to do mathematical operations on registers.

Description of the outputs:

- [Q]: Analog output: this is the value of the calculation formula output.
- [Overflow]: This output, which is Boolean type, indicates the presence of any saturation errors. This output is activated in the following cases.
- Result not included in the interval [-32768, +32767]

Example

Addition: Do not use Input [C].

Subtraction: Do not use either [A] or [B].

The BOOLEAN Function

Description



The BOOLEAN function gives the value of the output according to the combination of inputs.

The function has four inputs, and therefore 16 combinations. These combinations can be found in a truth table; for each of these, the output value can be adjusted. The number of configurable combinations depends on the number of inputs connected to the function.

Non-connected inputs are set to 0.

Inputs/Outputs

Description of the inputs:

- [IN1], [IN2], [IN3] and [IN4]: 4 Boolean inputs
- [OUT]: 1 Boolean output

Configuration

From the Configuration window, you can adjust the parameters.

In the parameters tab you can choose the ADL Container (LA0x) to be used

Having connected at least one input, configure the value of the output in the truth table, in the Parameters window.

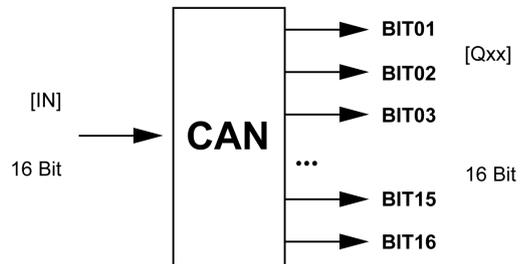
The output values can be 0 for the Inactive state, and 1 for the Active state.

The CAN Function (Word to Bit Conversion)

Description



The Word to Bit Conversion function breaks down a register type input (16-bit) into 16-bit outputs.



Inputs/Outputs

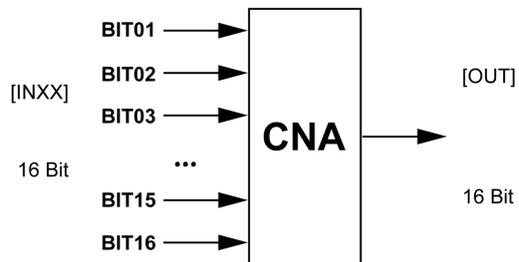
- **[IN]**: 1 register input
- **Q01 to Q16**: 16 Boolean outputs: Bit01 (least significant byte) ... Bit16 (most significant byte).

The CNA Function (Bit to Word Conversion)

Description



The Bit to Word Conversion function produces a register (16-bit) type output from 16-bit type inputs.



Inputs/Outputs

- [IN01] to [IN10]: 16 Boolean inputs: Bit01 (least significant byte) ... Bit16 (most significant byte)
- [OUT]: 1 register output

The COMPARE (Comparison of two values)

Description



This function is used to compare two analog values.

The output [Q] is active if the result of the comparison between [IN1] and [IN2] is true, and if the Enable functions input is active or not connected.

The output does not change state if the Enable functions input changes from Active to Inactive.

Inputs / Outputs

- [IN1]: 1 analog input
- [IN2]: 1 analog input
- [ENABLE]: 1 Boolean input
If the [IN1] or [IN2] input is not connected, the value is set to 0. Note: The function block output is not updated because if the function block output is set to 1 and the Enable input switches to 0, the output will remain at 1.
- [Q]: 1 Boolean output

Configuration

From the Configuration window, in the Parameters tab you can choose the following comparison operators:

Greater than

Greater than or equal to

Equal to

Different

Less than or equal to

Less than

The EDGE Function

Description



This function detects a signal falling or rising edge.

Inputs/Outputs

- Signal tested.
[Q] 1 Boolean Output.
- Pulse when the edge is detected.
[IN]: 1 Boolean input.

Configuration

From the Configuration window, adjust the Edge type in the Parameters tab:

- Falling
- Rising
- Twice

Edge_Type = rising edge

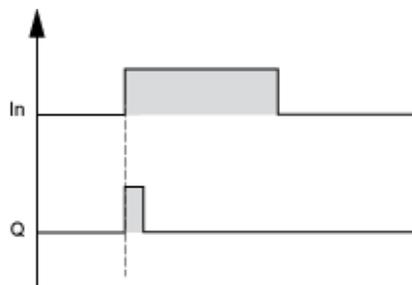


Table with four columns

Edge type		In (t)	In (t-1)	Q (t)
2 Q values possible	Falling	1	0	1
		0	0	0
		1	1	0
		0	1	0
	Rising	1	0	0
		0	0	0
		1	1	0
		0	1	1

The GAIN Function

Description



The Gain function allows analog values to be converted by changing the scale and the offset.

Gain calculation formula:

$$[Q] = (A / B) \times \text{INPUT} + [C].$$

Inputs Description

- **[IN]:** 1 analog input. This is an integer between -32768 and 32767.
- **[ENABLE]:** 1 Boolean input. This is the gain function input command.
The state of this input determines the block operation: if the Enable Function input is inactive, the Calculation output retains the last calculated value.

Outputs Description

- **[Q]:** 1 analog output. This is the output value of the gain function. This value depends upon the state of the Enable function input.
- **[OVERFLOW]:** 1 Boolean output. If it is:
 - Active: the Calculation output is equal to the result of the gain calculation formula.
 - Inactive: the Calculation output is equal to zero.

NOTE: The function block output is not updated because if the function block output is set to 1 and the Enable input switches to 0, the output will remain at 1.

Configuration

From the **Configuration** window, set in the **Parameters** tab:

A/B which corresponds to the gain applied by the function with:

- A: from -32768 to 32767
- B: from -32768 to -1 and from 1 to 32767 (4)
- C is the offset applied by the function, and is an integer between -32768 and 32767(5)

In addition, it is possible to define an operating range by setting limits for the function output:

- Lower limit: integers between -32768 and 32767
- Upper limit: integers between -32768 and 32767

The MUL/DIV Arithmetic Function

Description



The MUL/DIV arithmetic function is used to perform operations on integers:

- Multiplication
- Division

The MUL/DIV Calculation formula:

$$[Q] = [A] \times [B] / [C]$$

Description of Inputs

- **[A]**: 1 analog input
- **[B]**: 1 analog input
- **[C]**: 1 analog input

NOTE: If the INPUTS are not connected, they are set to 1. It is not possible to do mathematical operations on registers.

Description of Outputs

- **[Q]**: 1 analog output. This is the value of the calculation formula output.
- **[OVERFLOW]**: This output, which is Boolean type, indicates the presence of any saturation errors. This output is activated in the following cases:
 - - The consequence of the operations is a result that is not in the range [-32768, +32767]
 - The Error propagation input is active
 - Input 3 equals 0

Examples

- Multiplication: Do not use input **[C]**
- Division: Do not use either the **[A]** or **[B]** inputs

The MUX Function (Multiplexing)

Description



The Multiplexing function carries out dual input channel multiplexing on the output.

Description of Inputs

- **[SEL]**: 1 Boolean input. This input is used to choose the input channel to be applied to the output.
- **[IN1]**: 1 analog input. This is the multiplexer input [IN1].
- **[IN2]**: 1 analog input. This is the multiplexer input [IN2]

NOTE: If channels **[IN1]** or **[IN2]** are not connected, they are set to 0.

Description of Outputs

- **[Q]**: 1 analog input. This is the multiplexer output **[Q]**.

This value depends upon the state of the Command input **[SEL]**.

If...	Then ...
If the Command input is inactive	the Output corresponds to [IN1] .
If the Command input is active	the Output corresponds to [IN2] .

NOTE: The function block output is not updated because if the function block output is set to 1 and the Enable input switches to 0, the output will remain at 1.

The BIT READ Function

Description



The **BIT READ** function block is used for reading one bit of the parameters.

To...	Use ...	Comment
read one bit of the drive parameters	the ADL containers to link the drive parameters.	See description of the [ADL CONTAINERS] (<i>F R d -</i>)
read a communication protocol parameter bit	the M00x parameters to store data.	See description of the [FB PARAMETERS] (<i>F b P -</i>)

Inputs / Outputs Description

- **[ENABLE]**: 1 Boolean input
- **[Q]**: 1 Boolean Readbit output

Configuration

From the **Configuration** window, you can adjust the parameters.

In the **Parameters** tab you can choose the **ADL Container** (LA0x) that will be used.

The READ Ana Param Function

Description



The **Read Ana Param** function block is used to read a value.

To...	Use ...	Comment
Read the drive parameters	the ADL containers to link the drive parameters.	See description of the [ADL CONTAINERS] (<i>F R d -</i>)
Read communication protocol parameters	the M00x parameters to store data. Store values for reuse by the function blocks and communication protocol.	See description of the [FB PARAMETERS] (<i>F b P -</i>)

Inputs / Outputs

- 1 Boolean input
- 1 analog output

NOTE: The function block output is not updated because if the function block output is set to 1 and the Enable input switches to 0, the output will remain at 1.

Configuration

From the **Configuration** window, you can adjust the parameters.

In the **Parameters** tab you can choose the ADL Container (LA0x) to be used.

The READ Reg Param Function

Description



The **Read Reg Param** function block is used to read a value.

To...	Use ...	Comment
Read the drive parameters	the ADL containers to link the drive parameters.	See description of the [ADL CONTAINERS] (<i>F R d -</i>)
Read communication protocol parameters	the M00x parameters to store data. Store values for reuse by the function blocks and communication protocol.	See description of the [FB PARAMETERS] (<i>F b P -</i>)

Inputs / Outputs

- 1 Boolean input
- 1 register output

Configuration

From the **Configuration** window, you can adjust the parameters.

In the **Parameters** tab you can choose the ADL Container (LA0x) to be used.

The RS Memory Function

Description



The **SET RESET** function works as follows:

- Activating the **SET** input activates the output, which remains in this state even if the **SET** input is then deactivated
- Activating the **RESET** input deactivates the output

Non-connected inputs are set to Inactive.

Inputs / Outputs Description

- **[SET]**: 1 Boolean input
- **[RESET]**: 1 Boolean input
- **[Q]**: 1 Boolean output

Configuration

From the **Configuration** window, you can change settings in the **Parameters** tab.

If both inputs are active, the state of the output depends on how the function is configured:

- The **Q** output is active if the **SET** has priority option is configured
- The **Q** output is inactive if the **RESET** has priority option is configured

The SHIFT/ROLL Function

Description



ROLL moves bits in the selected direction and replaces new empty bits with bits ejected from the register. It is a circular register. The **CARRY** output contains the level (0/1) of the last bit moved.

The function is used to shift or roll the [IN] value of a fixed number of bits to the left/right.

At each cycle if the **ENABLE** signal is set, the logic block will perform a shift/roll to the left/right to the Number Bit request concerning the parameter configuration.

SHIFT moves bits in the selected direction and replaces empty bits with 0.

ROLL moves bits in the selected direction and replaces new empty bits with bits ejected from the register. It is a circular register. The **CARRY** output contains the level (0/1) of the last bit moved.

NOTE: It is impossible to shift records with analog values.

Inputs Description

- [IN]: 1 register input
- [ENABLE]: 1 Boolean function input

Outputs Description

- [Q]: 1 register output
- [CARRY]: 1 Boolean output. The carry output tells you the value of the last bit shifted.
NOTE: The function block output is not updated because if the function block output is set to 1 and the Enable input switches to 0, the output will remain at 1.

Configuration

From the **Configuration** window, you can change settings in the **Parameters** tab

- The mode: Shift or Roll.
- The direction: Left or right.
- The number of bits.

The TRIGGER function (Schmitt Trigger)

Description



The Schmitt Trigger function allows an analog value to be monitored relative to two thresholds.

The output changes state if:

- The **[IN]** value is less than the minimum value
- The **[IN]** value is greater than the maximum value

If the **[IN]** input is between the two, the **[Q]** output does not change state.

Two setpoints - On to Off and Off to On - can be set as the minimum or maximum value. This involves reverse operation of the function. These two operations are shown in the diagrams.

If the **[ENABLE]** input is inactive, the **[Q]** output remains inactive. The **[Q]** output does not change state if the **[ENABLE]** input changes from Active to Inactive.

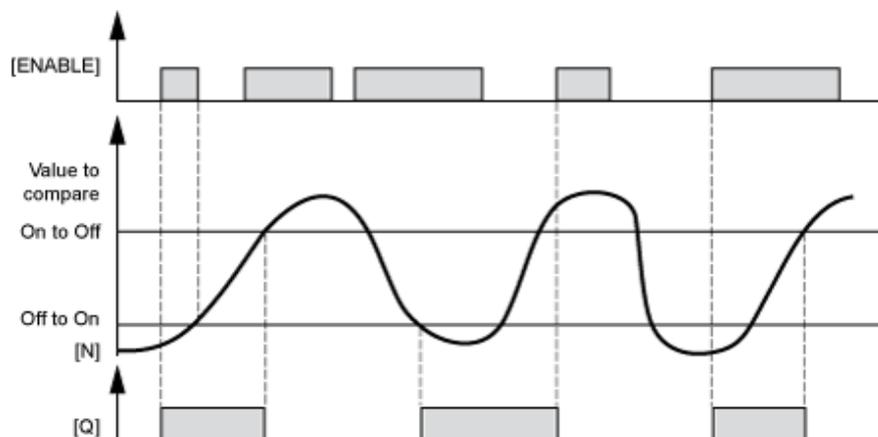
Inputs/Outputs

- **[IN]**: 1 analog setpoint input
- **[ENABLE]**: 1 Boolean input
- **[Q]**: 1 Boolean output

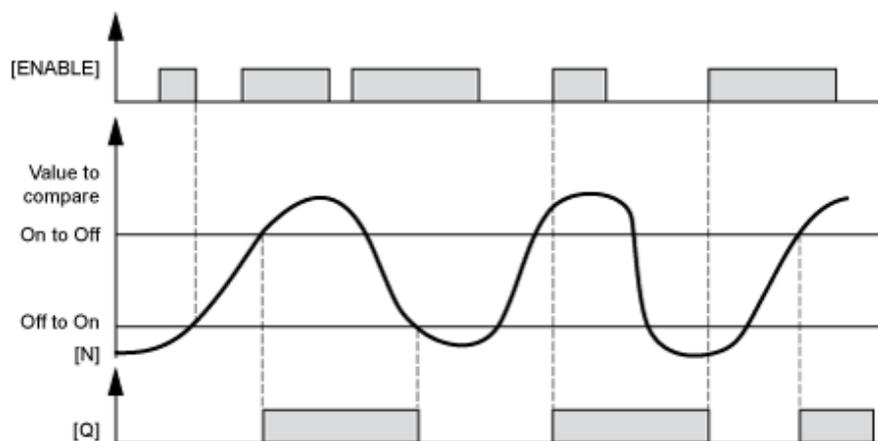
NOTE: The function block output is not updated because if it is set to 1 and the Enable input changes to 0, the output will remain at 1.

Operating Diagrams

The figure below shows the possible output states when the On to Off setpoint is higher than the Off to On setpoint.



The figure below shows the possible output states when the Off to On setpoint is higher than the On to Off setpoint.



The COUNT Up/Down Counter Function

Description



The Up/Down Counter function is used to up/down count from a preset value resulting from a calculation outside the function.

Inputs Description

- **[UP]**: 1 Boolean upcounter input
- **[DOWN]**: 1 Boolean downcounter input
- **[RESET]**: 1 Boolean input
- **[LOAD]**: 1 Boolean input
- **[PV]**: 1 analog Preset value input
- **[ENABLE]**: 1 input

Activation of the **[LOAD]** input to 1 is used to change the counter with the value available at the **[PV]** input.

The **[PV]** input can be connected to a NUM constant, to an analog input, or to any other kind of function block output that delivers an analog type value.

A rising edge on the:

- Upcounter input: Increments the counter
- Downcounter input: Decrements the counter

Inputs Description

- **[Q]**: 1 Boolean output
- **[CURRENT]**: The current counter value

NOTE: The function block output is not updated because if it is set to 1 and the Enable input changes to 0, the output will remain at 1.

Available Functions

Several functions are available:

- Upcounting and forcing the counter to 0 on initialization
- Upcounting and forcing the counter to 0 on initialization and when the count value is reached
- Downcounting and forcing the counter to the preset value on initialization
- Downcounting and forcing the counter to the preset value on initialization and when 0 is reached

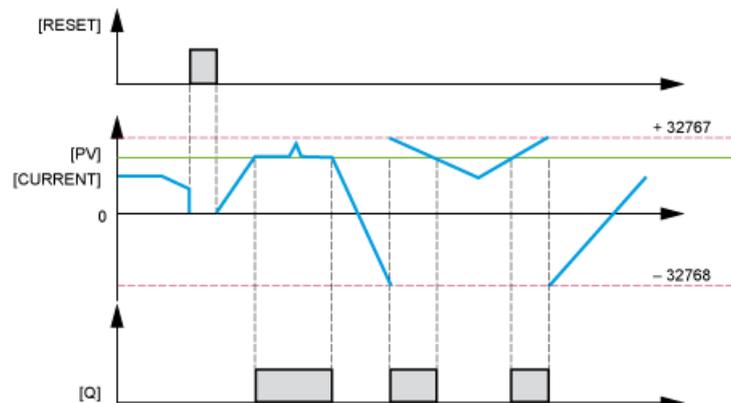
State of the Output

A1: When the count value is reached, the **[Q]** output changes to 1 and remains at 1 for as long as the count value is greater than or equal to the **[PV]** Preset value.

At 0: If the transitions on the downcounter input change the count value back to a value less than the Preset value

Activation of the **[Reset]** or **[LOAD]** inputs resets the counter.

When the **[Reset]** input is set to 1, the **[Q]** output remains at state 0. When the **[Reset]** input changes to 0, the up/down counting operation is restarted from zero.



The BIT WRITE Function (WriteBitParam)

Description



The **WriteBitParam** function block is used to write one bit of the parameters.

To write one bit of the drive parameters:

You must use the ADL containers to link the drive parameters (see description of **[ADL CONTAINERS]** (*F R d -*)).

⚠ WARNING

PARAMETER MODIFICATIONS WITHOUT EFFECT

While parameters are saved to the EEPROM, no other modifications to parameters can be saved since write access to the EEPROM is unavailable during this process.

Verify that write access to the EEPROM is possible before modifying further parameters (bit 0 of the internal status register ETI must be 0).

Failure to follow these instructions can result in death, serious injury, or equipment damage.

To store or reuse a value: The M00x parameters are easy to use to store data. You can store values for reuse by the functions blocks and communication protocol (see description of **[FB PARAMETERS]** (*F b P -*)).

Inputs/Outputs

The WriteBitParam function uses:

- **[IN]**: 1 Boolean WriteBit input
- **[ENABLE]**: 1 Boolean input
NOTE: The function block output is not updated because if it is set to 1 and the Enable input changes to 0, the output will remain at 1.

Configuration

You can adjust the parameters from the **Configuration** window:

In the **Parameters** tab, you can select the **ADL** Container (LA0x) to be used.

The WRITE Ana Param function

Description



The **WRITE Ana Param** function block is used to write one bit of the parameters.

To write drive parameters:

You must use the **ADL** containers to link the drive parameters (see description of **[ADL CONTAINERS]** (*F R d -*)).

⚠ WARNING

PARAMETER MODIFICATIONS WITHOUT EFFECT

While parameters are saved to the EEPROM, no other modifications to parameters can be saved since write access to the EEPROM is unavailable during this process.

Verify that write access to the EEPROM is possible before modifying further parameters (bit 0 of the internal status register ETI must be 0).

Failure to follow these instructions can result in death, serious injury, or equipment damage.

To store or reuse a value: The M00x parameters are easy to use to store data. You can store values for reuse by the functions blocks and communication protocol (see description of **[FB PARAMETERS]** (*F b P -*)).

Inputs/Outputs

The WriteBitParam function uses:

- 1 analog input
- 1 Boolean input

NOTE: The function block output is not updated because if it is set to 1 and the Enable input changes to 0, the output will remain at 1.

Configuration

You can adjust the parameters from the **Configuration** window:

In the **Parameters** tab, you can select the **ADL** Container (LA0x) to be used.

The WRITE Reg Param function

Description



The **WRITE Reg Param** function block is used to write one bit of the parameters.

To write drive parameters:

You must use the **ADL** containers to link the drive parameters (see description of **[ADL CONTAINERS]** (*F R d -*)).

⚠ WARNING

PARAMETER MODIFICATIONS WITHOUT EFFECT

While parameters are saved to the EEPROM, no other modifications to parameters can be saved since write access to the EEPROM is unavailable during this process.

Verify that write access to the EEPROM is possible before modifying further parameters (bit 0 of the internal status register ETI must be 0).

Failure to follow these instructions can result in death, serious injury, or equipment damage.

To store or reuse a value: The M00x parameters are easy to use to store data. You can store values for reuse by the functions blocks and communication protocol (see description of **[FB PARAMETERS]** (*F b P -*)).

Inputs/Outputs

The WriteBitParam function uses:

- 1 register input
- 1 Boolean input

NOTE: The function block output is not updated because if it is set to 1 and the Enable input changes to 0, the output will remain at 1.

Configuration

You can adjust the parameters from the **Configuration** window:

In the **Parameters** tab, you can select the **ADL** Container (LA0x) to be used.

The STU Function (16-bit analog TO 16-bit register)

Description



The STU function block is used to convert an analog signal to a register signal by limiting.

Analog input range: -32768 to +32767.

Register output range: 0 to 32767.

Inputs/Outputs

- **[IN]:** 1 analog input
- **[Q]:** 1 register output
- **[OVERFLOW]:** 1 Overflow output
[OVERFLOW] is a Boolean output. If the analog input **[IN]** value is negative, **[OVERFLOW]** = 1. If the analog input **[IN]** value is positive, **[OVERFLOW]** = 0.

The UTS Function (16-Bit Register Input to 16-Bit Analog Output)

Description



The UTS function block is used to convert a 16-bit register input to a 16-bit analog output (with limiting).

Register input range: 0 to 65535.

Register output range: -32768 to +32767.

Inputs/Outputs

- **[IN]**: 1 register input
- **[Q]**: 1 analog output
- **[OVERFLOW]**: 1 Overflow output

The **[OVERFLOW]** output is of Boolean type. This output is active if the register input value exceeds 32767; the analog output value will be set to 32767.



Chapter 4

Description of ATV320 Parameters

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Function Block Status	56
Function Block Command	57
Function Block Fault	58
Function Block Parameters	58

Function Block Status

Description

This indicates the function block status in the HMI.

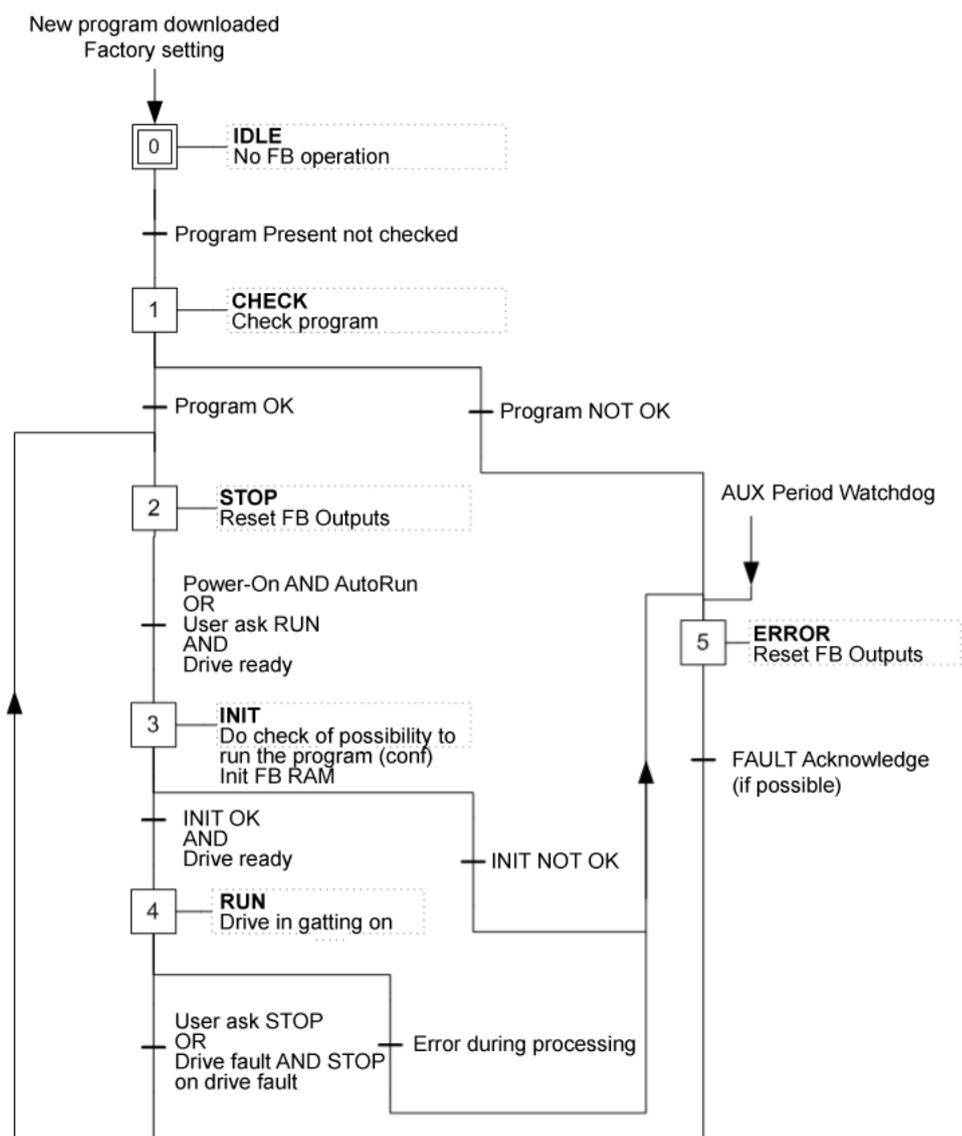
Example of display of function block status in the graphic display terminal:

OFF light: A valid function block program is in the ATV320 in stop mode.

ON light: A valid function block program is in the ATV320 in run mode. The drive is considered as being in run mode and the configuration parameters cannot be modified.

Operation

FBSst	Function blocks status
IDLE	No binary file in the target, the FB is waiting for a download
CHEC	Check the program downloaded.
STOP	The FB is stopped.
INIT	Initialization of FB program in function of the configuration, perform also a reset of FB RAM.
RUN	The FB is running.
ERR	An internal error occurs. The FB is in default mode.



Function Block Command

Description

Appears in the HMI menu to command the function block to switch automatically from Run to STOP.

WARNING

LOSS OF CONTROL

If the **[FB start mode]** (*F b r Π*) or **[FB Command]** (*F b C d*) parameters are modified by the communication network, the communication status must be monitored with Bit 11, 12 and 13 of **[FB SYSTEM WORD 06]** (*S D D 2*).

Failure to follow these instructions can result in death, serious injury, or equipment damage.

	<i>F b C d</i> [FB Command]	<i>F b r Π</i> [FB start mode]	FB Behaviour	New <i>F b S t</i> [FB stop mode]
On PowerOn	STOP	No	STOP	STOP
	STOP	Yes	AutoRun	RUN
	STOP	Llx	STOPRUN on rise edge of LI	STOPRUN on rise edge of LI
On change of <i>F b r u</i>	STOP	XX → No	STOP	STOP
	STOP	XX → Yes	No impact, only take into account on next PowerOn	STOP
	STOP	XX → Llx	STOPRUN on rise edge of LI	STOPRUN on rise edge of LI
On Start with <i>F b C d</i>	STOP → START	No	RUN	START
	STOP → START	Yes	RUN	START
	STOP → START	Llx	STOPRUN on rise edge of LI	STOPRUN on rise edge of LI
On Stop with <i>F b C d</i>	START → STOP	XX	STOP	STOP
End of download/Factory setting	STOP	No	STOP	STOP
	STOP	Yes	STOP	STOP
	STOP	Llx	STOPRUN on rise edge of LI	STOPRUN on rise edge of LI
On LI fall edge	START	Llx	STOP	STOP

FB execution has the same effect as motor rotation, the configuration is locked and transfers are prohibited.

Function Block Fault

Description

If the drive is in *F b E* fault mode, see the *F b F t* [**FB fault**] parameter.

Code	Name/Description
<i>F b F t</i>	[FB fault]
<i>n o</i>	[No]: No fault detected
<i>i n t</i>	[Internal]: Internal fault detected
<i>b i n</i>	[Binary file]: Binary fault detected
<i>i n P</i>	[Intern para.]: Internal parameter fault detected
<i>P A r</i>	[Para. RW]: Parameter access fault detected
<i>C A L</i>	[Calculation]: Calculation fault detected
<i>t o A U x</i>	[TO AUX]: TimeOut AUX task
<i>t o P P</i>	[TO synch]: TimeOut in PRE/POST task
<i>A d L</i>	[Bad ADLC]: ADLC with bad parameter
<i>i n</i>	[Input assign.]: Input not configured

Errors do not persist, switching ON and OFF resets the FB fault (as it was detected again).

Function Block Parameters

Description

The 8 Function block parameters (M001 to M008) can store values, which can be used to configure or monitored the application. Because, the M00x are in reading and writing on the drive.

NOTE: The M001 to M004 are saved in EEprom but the M005 to M008 are written in RAM.

Chapter 5

Communication Parameters

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
ADL Containers	60
FB SYSTEM WORDS	61

ADL Containers

Description

ADL Containers are linked to 6 function blocks

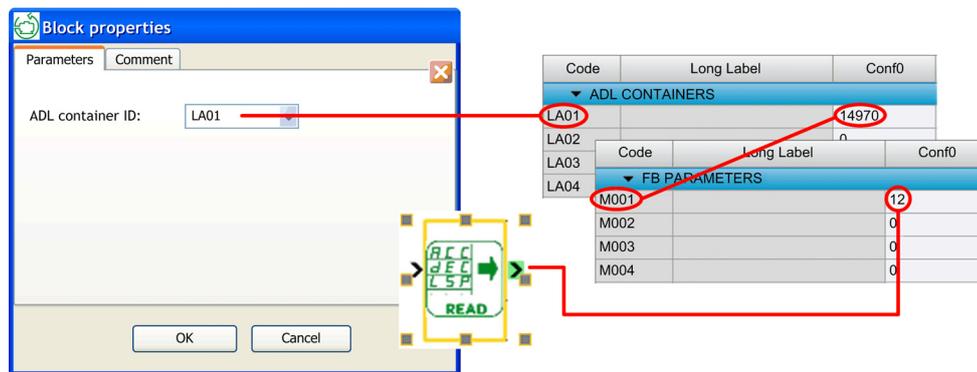
- Read Ana Param
- Read Reg Param
- Write Ana Param
- Write Reg Param
- ReadBitParam
- WriteBitParam

The drive has 8 ADL containers (LA01 to LA08).

These LA0x contain the address of the target parameters.

This configuration is used during graphic editing of the program in Function Blocks.

There are 8 ADL containers used to connect %MW, %SW, or other drive parameters (see programming manual in Help menu).



Depending on which task in the Read/Write Ana/Reg Param logic block is implemented, it will allow connection to Fast or Slow parameters.

The rule is:

- In PRE/POST tasks: Connection to Fast access parameters only
- In the AUX task: Connection to Slow and Fast access parameters

PRE and POST tabs have priority over AUX tabs.

FB SYSTEM WORDS

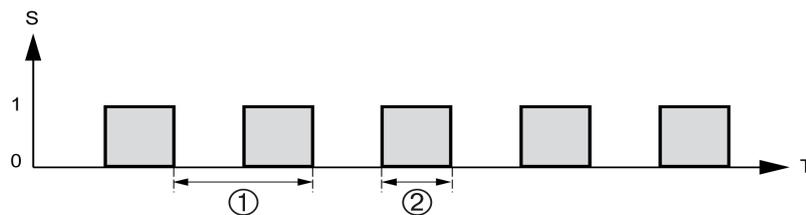
FB SYSTEM WORD 01 Overview

[FB SYSTEM WORD 01] (SDDI)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TB1MIN:Clock	TB1SEC:Clock	TB100MS: Clock	Reserved	Reserved	Reserved	Reserved	Reserved
bit: period1min	bit: period1s,	bit: period 100ms,					
duty cycle50%	duty cycle50%	duty cycle 50%					
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reserved	Reserved	1RSTSCANRUN: FB is executing its 1rst cycle since last RUN. Used only in PRE and POST	Reserved	Reserved	Reserved	Reserved	Reserved

Information about Bits 5, 6 and 7

An internal timer controls the bit change of status.

NOTE: Bits 5, 6 and 7 should be used in the PRE and POST tasks. As in the AUX task, repeatability is not guaranteed (See Global overview of Function Blocks/ATV Logic).



Information about Bit 13

The transition from STOP to RUN mode (even after a cold start) is postponed by setting the %S13 system bit to 1. This bit is reset to 0 at the end of the first cycle of the POST task in run mode.

FB SYSTEM WORD 02 Overview

[FB SYSTEM WORD 02] (SDDP)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reserved	Reserved	ModbusCommunicationloss	CANopenCommunicationloss	Communication loss withoption card	Reserved	Reserved	Reserved

Information about Bits 13, 12 and 11

Detection is based on loss of communication, not on lack of communication. There must therefore be an initial communication to enable detection.

FB SYSTEM WORD 06 Overview

[FB SYSTEM WORD 06] (5 0 0 5)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reserved	USRMWSAVE: Users variable save request	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved

Information about Bit 14

The %S006.14 bit is the %S94 bit on Modicon M340 and Premium.

WARNING

PARAMETER MODIFICATIONS WITHOUT EFFECT

Setting bit 14 of [FB SYSTEM WORD 06] parameter to 1 starts saving of parameters to the EEPROM. While parameters are saved to the EEPROM, no other modifications to parameters can be saved since write access to the EEPROM is unavailable during this process.

Verify that write access to the EEPROM is possible before modifying further parameters (bit 0 of the internal status register ETI must be 0).

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Chapter 6

Application Example

Windowpane Application Example

Overview

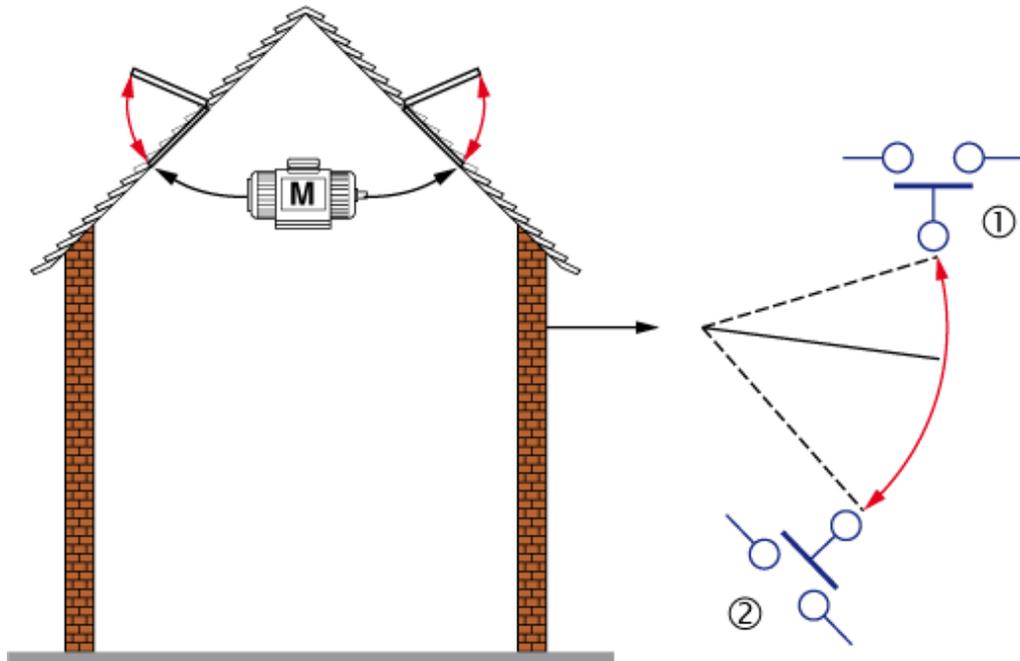
This example describes how the glass panes in a greenhouse can be managed automatically.

Specifications

The owner of a greenhouse wishes to manage opening and closing of the ventilation panes located on the greenhouse roof.

The greenhouse has two panes to provide ventilation.

The opening of these panes is controlled by a motor and two sensors that indicate whether the panes are open or closed:



During the day, the panes open to ventilate the structure when the temperature reaches 25°C. If the temperature falls below 20°C, the panes must close again.

Input/Output Tables

Real input	Virtual input
IL01	LI3
IL02	LI4
IA01	AI3
Real output	Virtual output
OL01	FRD
OL02	RRS

Virtual Input Pin Creation

Double-click on the inputs/outputs, then assign a data type using the **Virtual Port Configuration** window.

The detailed procedure is explained in the [Creating an FBD Application Using Function Blocks \(see page 15\)](#) section.

Program Creation

Drag and drop function blocks from the Function Block Set window. Set the required links between inputs, function blocks and outputs.

The detailed procedure is explained in the Creating an FBD Application Using Function Blocks ([see page 15](#)) and Function Block Configuration ([see page 16](#)) sections.

To optimize the cycle time you can adjust the program order of execution.

Check the Application

The detailed procedure is explained in the Check the Application ([see page 22](#)) section.

Inputs, Outputs and ADL Container Configuration

The detailed procedure is explained in the ADL Containers ([see page 60](#)) section.

Once the program and parameter assignment is complete the configuration can be downloaded.

After downloading, to execute the program, you should set the **[FB Command]** *F b C d* parameter to *S t r t* for testing.

But for independent applications use the **[FB start mode]** *F b r n* parameter (see the Programming Manual in the Help menu).

