SoMachine Basic Example Guide

Altivar Control from Serial Modbus xSample_ATV_Modbus SL_M221.smbe

11/2015

www.schneider-electric.com

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.

Table of Contents

Safety Information

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.

About the Book

At a Glance

Document Scope

This document describes a SoMachine Basic example application that allows you to operate and control an Altivar variable speed drive through Modbus serial line communication. The example application is based on graphical SFC and communication function blocks.

Since the example described in this document is intended for learning purpose only, it must not be used directly on products that are part of a machine or process.

This document and its related SoMachine Basic project file focus on specific functions and function blocks of the Schneider libraries provided with SoMachine Basic, and on specific features available in SoMachine Basic if these features are related to these libraries. They are intended to help you develop, test, commission, and integrate applicative software of your own design on control systems.

The example is intended for new SoMachine Basic users who already have some degree of expertise in the design and programming of control systems.

Validity Note

This document has been updated for the release of SoMachine Basic V1.4.

Related Documents

Title of Documentation	Reference Number
ATV32 Quick Start	S1A41715
ATV32 Installation Manual	S1A28686
ATV32 Programming Manual	S1A28692
ATV32 Modbus Manual	S1A28698
ATV32 Communication Parameters	S1A44568
ATV32 Atex Manual	S1A45605
ATV32 Safety Manual	S1A45606
ATV32 Certification and Other Option Manuals	

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

Product Related Information

⚠ WARNING

LOSS OF CONTROL

- The designer of any control scheme must consider the potential failure modes of control paths and, for certain 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 and 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 this equipment 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 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" or their equivalent governing your particular location.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

- Only use software approved by Schneider Electric for use with this equipment.
- Update your application program every time you change the physical hardware configuration.

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

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Do not include the code from this example in your machine or process without thoroughly testing your entire application.

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

Chapter 1
Introduction

What Is in This Chapter?

This chapter contains the following topics:

Before You Begin

General

The products specified in this document have been tested under actual service conditions. Your specific application requirements may be different from those assumed for this and any related examples described herein. In that case, you have to adapt the information provided in this and other related documents to your particular needs. To do so, you need to consult the specific product documentation of the hardware and/or software components that you may add or substitute for any examples specified in this documentation. Pay particular attention and conform to any safety information, different electrical requirements, and normative standards that would apply to your adaptation.

⚠ WARNING

REGULATORY INCOMPATIBILITY

Ensure that all equipment applied and systems designed comply with all applicable local, regional, and national regulations and standards.

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

The use and application of the information contained herein require expertise in the design and programming of automated control systems. Only the user or integrator can be aware of all the conditions and factors present during installation and setup, operation, and maintenance of the machine or process, and can therefore determine the automation and associated equipment and the related safeties and interlocks which can be effectively and properly used. When selecting automation and control equipment, and any other related equipment or software, for a particular application, the user or integrator must also consider any applicable local, regional, or national standards and/or regulations.

Some of the major software functions and/or hardware components used in the proposed architectures and examples described in this document cannot be substituted without significantly compromising the performance of your application. Further, any such substitutions or alterations may completely invalidate any proposed architectures, descriptions, examples, instructions, wiring diagrams and/or compatibilities between the various hardware components and software functions specified herein and in related documentation.You must be aware of the consequences of any modifications, additions, or substitutions.

A residual risk, as defined by EN/ISO 12100-1, Article 5, remains if:

- it is necessary to modify the recommended logic and if the added or modified components are not properly integrated in the control circuit.
- you do not follow the required standards applicable to the operation of the machine, or if the adjustments to and the maintenance of the machine are not properly made (it is essential to strictly follow the prescribed machine maintenance schedule).
- the devices connected to any safety outputs do not have mechanically linked contacts.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Thoroughly read and understand any and all device manuals for the characteristics and properties of the devices employed before attempting to modify parameters that may alter those characteristics and properties.

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

Start-Up and Test

Before using electrical control and automation equipment after design and installation, the application and associated functional safety system must be subjected to a start-up test by qualified personnel to verify correct operation of the equipment. It is important that arrangements for such testing be made and that enough time is allowed to perform complete and satisfactory testing.

⚠ CAUTION

EQUIPMENT OPERATION HAZARD

- Verify that all installation and setup 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 injury or equipment damage.

Verify that the completed system, including the functional safety system, is free from all short circuits and grounds, except those grounds installed according to local regulations. If high-potential voltage testing is necessary, follow the recommendations in equipment documentation to help prevent injury or equipment damage.

Operations and Adjustments

General

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 installed and operated.

In some applications, such as packaging machinery, additional operator protection such as point-of-operation guarding must be provided. This is necessary if the hands and other parts of the body are free to enter the pinch points or other hazardous areas where 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.

⚠ **WARNING**

UNINTENDED EQUIPMENT OPERATION

Do not use the equipment configured and programmed by this software in safety-critical machine functions, unless the equipment and software are otherwise designated as functional safety equipment and conforming to applicable regulations and standards.

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

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 examples and implementations suggested herein. It is sometimes possible to adjust the equipment incorrectly and this produces unsatisfactory or unsafe operation. Always use the manufacturer instructions as a guide to functional adjustments. Personnel who have access to these adjustments must be familiar with the equipment manufacturer instructions and the machinery used with the electrical equipment.

Only those operational adjustments required by the machine operator should be accessible to the operator. Access to other controls should be restricted to help prevent unauthorized changes in operating characteristics.

Chapter 2
Example Description

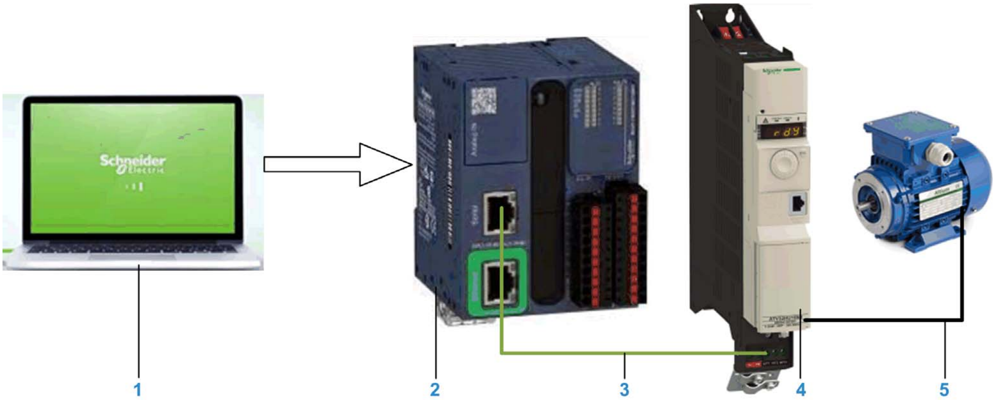
What Is in This Chapter?

This chapter contains the following topics:

Overview

Overview

This template example helps you to program the set of exchanges to perform with an Altivar (ATV32) drive to control asynchronous motors with a M221 Logic Controller.



1 PC with SoMachine Basic to program and monitor
2 M221 Logic Controller, master
3 Modbus serial line RTU 19200, 8, E
4 ATV32 Drive, slave
5 Power

The components in the example are:

- PC with SoMachine Basic V1.4 or later to program the M221 and monitor the application example.
- M221 Logic Controller to send commands to the ATV32 drive.
- Altivar drive matching with the motor (this example uses ATV32.)
- Asynchronous motor.

The template example is applied using an ATV32 drive, although another ATV reference may be used in its place. Refer to the particular ATV documentation to verify whether the Modbus registers used in this example are compatible with it and adapt the code used in the example, when and where necessary.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Install and operate this equipment according to the conditions described in the Environmental Characteristics found in the hardware guides for the equipment.

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

If you want to reproduce this example on your installation, you must have tested that the ATV32 drives your motor correctly.

Refer to the technical documentation *(see page 7)* that is available on *Schneider Electric website* to install and configure the ATV32 products.

The template example program is designed to be easily integrated into your application. You can find explanations on how to integrate the template example in the comments of the objects used in the template. An animation table is provided within the template example to assist you in the integration and use of it.

You can monitor the template in connected mode to:

- Initialize the ATV32 to drive the motor.
- Send the frequency setpoint (`LFR`) and activate the motor rotation (`CMD`).
- Monitor the ATV32 status (`ETA`) and the motor frequency (`RFR`).
- Monitor the communication status (count of invalid exchanges).

These data are designed for direct use in your application.

This example is a simple use of the ATV32. You can add more features by adapting the template to your requirements according to the ATV32 specifications.

For the M221 Logic Controller application, an ATV32 is a set of registers. These registers are stored into the ATV memory at dedicated addresses. When a register is modified, the ATV32 processes the modification and tries to perform the action. The ATV32 also stores its internal state in its status registers. These registers can be set or read directly by the ATV32 front panel or by the Modbus clients over the serial line. In this example, the M221 Logic Controller uses the Modbus serial line to modify the ATV registers. The set of registers of the ATV32 can be found in the document ATV32 Communication Parameters *(see page 7)*.

Mode of Operation

Notation

The graphic display terminal (to be ordered separately - reference VW3A1101) menus are shown in square brackets. Example: **[COMMUNICATION]**.

The integrated 7-segment graphic display terminal menus are shown in round brackets.

Example: (con-).

The parameter names are displayed on the graphic display terminal in square brackets.

Example: **[Fallback speed]**.

ATV32 Configuration

The ATV32 communication configuration must be done before establishing the communication with the M221 Logic Controller. Modbus communication functions of the ATV32 can be accessed from its menu with the graphic display terminal.

Configure the drive as follows:

1. Before configuring the drive, set the parameters to factory setting (**[1.3 CONFIGURATION]** (conF) → **[FACTORY SETTINGS]** (FcS-)).
   - **[PARAMETER GROUP LIST]** (FrY-) = **[ALL]** (ALL).
   - **[Goto FACTORY SETTINGS]** (GFS) = enter.
2. Control the drive with a Modbus master (**[1.3 CONFIGURATION]** (conF) → **[FULL]** (FuLL) → **[COMMAND]** (CtL-)) to select Modbus as command channel active.
   - **[Ref.1 channel]** (Fr1-) = **[MODBUS]** (Mdb).
3. Select the Modbus address (**[1.3 CONFIGURATION]** (conF) → **[FULL]** (FuLL) → **[COMMUNICATION]** (CoM-) → **[MODBUS NETWORK]** (Md1-)).

The following table shows the set of possible configuration of Modbus address:

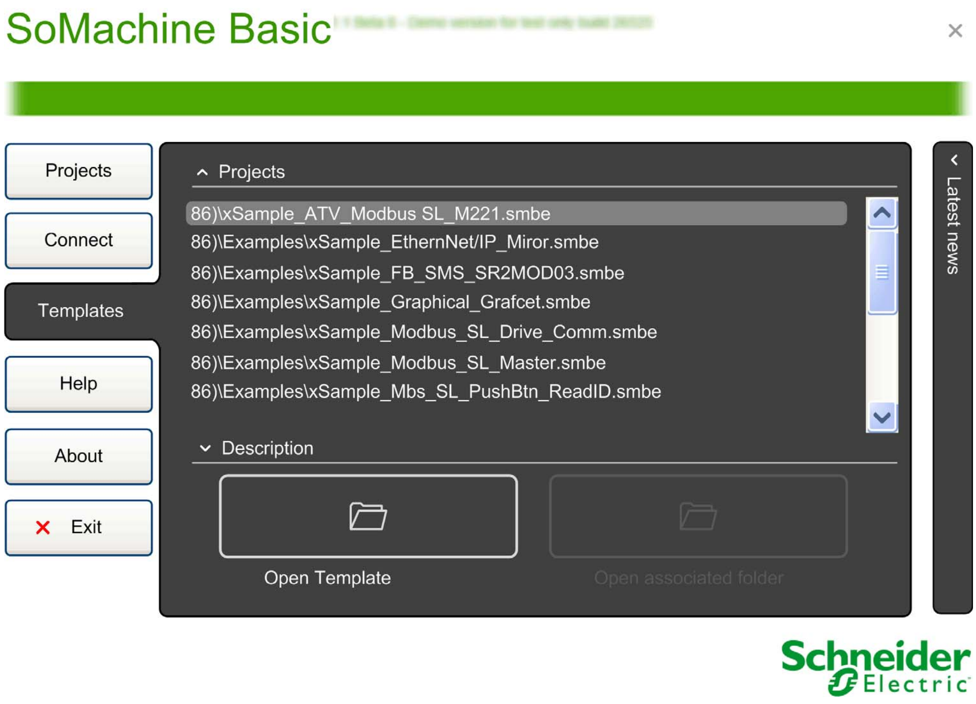
Parameter description	Range or listed values	Default	Possible value	Modbus address
[Modbus Address] (Add)	1 to 247 0:OFF (broadcast only)	**[OFF]** OFF	**[OFF; 1 to 247]** OFF; 1. . . 247	16#1771 = 6001
[Modbus baud rate] (tbr)	4.8 kbps 9.6 kbps 19.2 kbps 38.4 kbps	**[19.2 kbps]** 19. 2	**[4.8]** 4. 8 **[9.6]** 9. 6 **[19.2]** 19. 2 **[38.4]** 38. 4	16#1773 = 6003
[Modbus format] (tFO)	8 data bits, odd parity, 1 stop bit 8 data bits, even parity, 1 stop bit 8 data bits, no parity, 1 stop bit 8 data bits, no parity, 2 stop bits	**[8E1]** 8E1	**[8O1]** 8o1 **[8E1]** 8E1 **[8N1]** 8n1 **[8N2]** 8n2	16#1774 = 6004
[Modbus time out] (ttO)	Adjustable from 0.1 to 30 s	**[10s]** 10	**[0.1 to 30.0]** 0. 1. . . 30. 0	16#1775 = 6005

ATV32 configuration:

Step	Action
1	Configure your ATV32 with the following parameters: ● **[Modbus address]** (A d d) = 1 ● **[Modbus baud rate]** (t b r) =19.2 ● **[Modbus format]** (t F O) = 8E1 ● **[Modbus time out]** (t t O) = 10
2	Restart the ATV32 in order to take into account the Modbus parameters.

M221 Configuration

Open the application (*xSample_ATV_Modbus SL_M221.smbe*) on the computer running SoMachine Basic.



SoMachine Basic

Projects

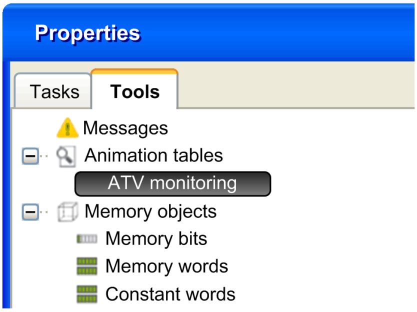
86)\xSample_ATV_Modbus SL_M221.smbe
86)\Examples\xSample_EthernNet/IP_Miror.smbe
86)\Examples\xSample_FB_SMS_SR2MOD03.smbe
86)\Examples\xSample_Graphical_Grafcet.smbe
86)\Examples\xSample_Modbus_SL_Drive_Comm.smbe
86)\Examples\xSample_Modbus_SL_Master.smbe
86)\Examples\xSample_Mbs_SL_PushBtn_ReadID.smbe

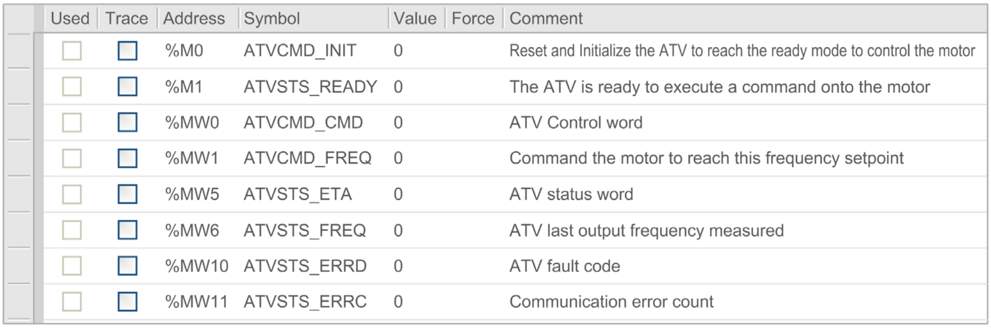
Description

Verify the parameters for serial line configuration:

Parameter	Value
Protocol	Modbus
Baud rate	19200
Parity	Even
Stop bits	1
Physical medium	RS-485
Transmission mode	RTU
Addressing	Master
Response time (x100 ms)	10
Time between frames (ms)	10

Download the application to the controller, open the animation table (**Tools** → **Animation tables** → **ATV monitoring**), then run the controller.





Used	Trace	Address	Symbol	Value	Force	Comment
□	□	%M0	ATVCMD_INIT	0		Reset and Initialize the ATV to reach the ready mode to control the motor
□	□	%M1	ATVSTS_READY	0		The ATV is ready to execute a command onto the motor
□	□	%MW0	ATVCMD_CMD	0		ATV Control word
□	□	%MW1	ATVCMD_FREQ	0		Command the motor to reach this frequency setpoint
□	□	%MW5	ATVSTS_ETA	0		ATV status word
□	□	%MW6	ATVSTS_FREQ	0		ATV last output frequency measured
□	□	%MW10	ATVSTS_ERRD	0		ATV fault code
□	□	%MW11	ATVSTS_ERRC	0		Communication error count

When the controller is in RUNNING state, requests are periodically sent to the ATV32 to get its `ETA` register value. Before initializing ATV32, verify that there is no Modbus communication interruption (`%MW11` value is stable).

If there is a communication interruption, verify again the hardware installation and the configuration parameters.

If the ATV32 is not in the READY state (`%M1` = 1), set the `%M0` to 1 to initialize the ATV32. The ATV32 `CMD` word (`%MW0`) and ATV32 `ETA` status word (`%MW5`) automatically change until the READY (`%M1` = 1) state is reached or the ATV32 ERRD status word (`%MW10`) is not 0.

If ATV32 is in ERROR state (`%MW10` <> 0), read the ATV32 documentation to identify the detected error. Use the INIT command (`%M0`) to reset the error and initialize the ATV32.

If `%M1` = 1, the ATV32 is ready to operate and the motor can be controlled. Start the motor by setting the frequency setpoint to the expected value (`%MW1`). The status word (`%MW6`) gives the frequency of the motor.

Template Adaptation

Basic Adaptation

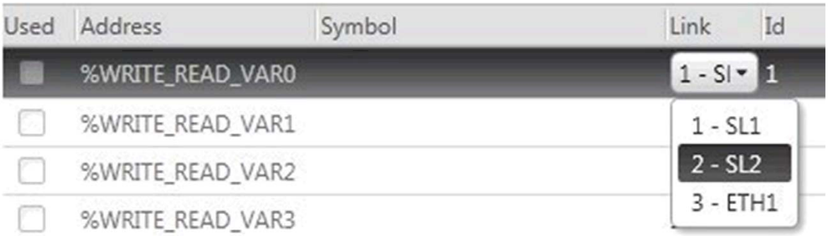
Communication settings can be modified, as long as:

- They are set on both ATV32 drive and M221 Logic Controller.
- The ATV32 drive is rebooted after the modification.

For example, increasing the communication speed could improve the response time of the ATV32. It is also possible to change the slave address of the ATV32.

If your M221 Logic Controller is equipped with two serial lines, you can use SL2 instead of SL1.

Configure the second serial line in SoMachine Basic configuration screen, and change the serial line selector of the function block:

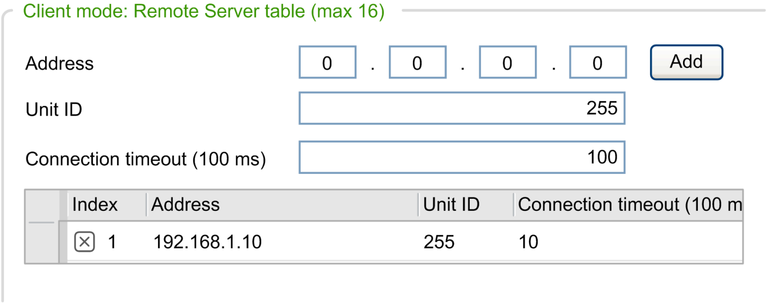


Used	Address	Symbol	Link	Id
■	%WRITE_READ_VAR0		1 - Sl ▾	1
□	%WRITE_READ_VAR1			
□	%WRITE_READ_VAR2			
□	%WRITE_READ_VAR3			

If your ATV32 is equipped with an Ethernet port, you can communicate with the Ethernet port of the M221 Logic Controller.

Configure the ATV32 with the Ethernet port and set its Ethernet parameters (for example, @IP=192.168.1.10).

- Set an address in the **ETH1 Modbus TCP configuration** screen:



Client mode: Remote Server table (max 16)

Address 0 . 0 . 0 . 0

Unit ID 255

Connection timeout (100 ms) 100

Index	Address	Unit ID	Connection timeout (100 m
☒ 1	192.168.1.10	255	10

- Change the `WRITE_READ_VAR` function blocks link to **[3 - ETH1]**.
- Connect the Ethernet port to the ATV32.
- Download the application in the controller and run it.

Adding Features to Your Application

Manage ATV32 Quick Stop Feature

The quick stop feature uses the bit 2 of the command register `CMD` of the ATV32. To manage this feature in your application code, choose an internal bit that is not used by your application.

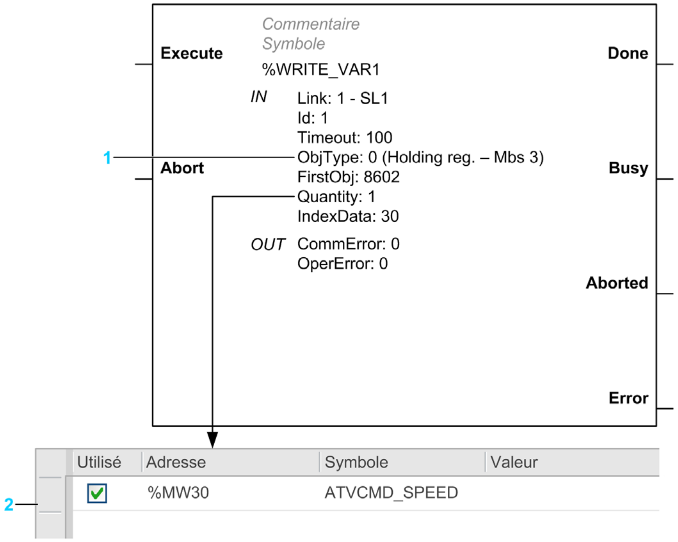
Management example if `%M2` is available:

1. Your program must copy this bit to `%MW0:X2` before the `WRITE_READ_VAR` block execution.
2. Your program must reset `%M2` when:
   - The exchange is not finished.
     NOTE: If a communication error occurs, a retry should be performed.
   - Controller goes in RUNNING state (`%S13`).
   - ATV32 becomes operational.
   - INIT occurs (`%S0` or `%S1`).
3. `%M2` can be set to 1 through animation table or through the program when required.

Use ATV32 Speed Command Instead of Frequency Set Point

The speed command is not mapped with the `CMD` register. The following changes have to be programmed:

- Add the `WRITE_VAR` function block to write the register speed.
- Optional: remove the `FREQUENCY` of the `WRITE_READ_VAR` function block.



%WRITE_VAR1

IN
Link: 1 - SL1
Id: 1
Timeout: 100
ObjType: 0 (Holding reg. – Mbs 3)
FirstObj: 8602
Quantity: 1
IndexData: 30

OUT
CommError: 0
OperError: 0

Utilisé	Adresse	Symbole	Valeur
☑	%MW30	ATVCMD_SPEED	

1 Write the Modbus address 8602 of the ATV32, corresponding to the speed setpoint word `LFRD`.
2 `%MW30`: Speed setpoint of the ATV32 (address 8602).

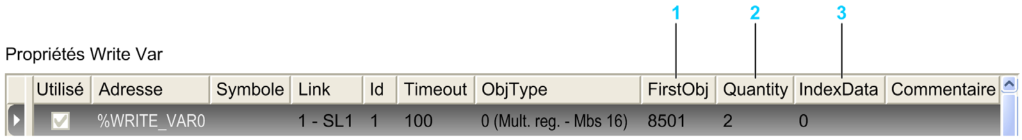
The response time of the ATV32 is impacted when the second exchange is performed.

Use Ramp or Other ATV32 Features

The ATV32 embeds various features that can be used in applications. For example, an acceleration ramp can be customized.

To configure such parameters for an application, add a new WRITE_VAR function block to the application. This block has the same parameters as the cyclic exchange block (WRITE_READ_VAR) except the address of the register and the size of the exchange. The size is generally 1 for one register to write. You can find the address value in the ATV32 parameter list. For example, the RPT (ramp type) register is mapped at address 9004 and the ACC (acceleration) is mapped at address 9001.

The WRITE_VAR block should be executed one time after the address value changes and before the execution of WRITE_READ_VAR block.



Propriétés Write Var

	Utilisé	Adresse	Symbole	Link	Id	Timeout	ObjType	FirstObj	Quantity	IndexData	Commentaire
▶	☑	%WRITE_VAR0		1 - SL1	1	100	0 (Mult. reg. - Mbs 16)	8501	2	0	

1 Modbus address of first object to write
2 Number of words to write
3 First address of the word table where the values to write are stored (%MW)

Monitor Several ATV32 Concurrently

If there are several ATV32 drives on the same serial line, verify the responsiveness required by your application. Use the Ethernet port in this case. The code has to be duplicated for each drive, and the data used must have its own objects assigned. Duplicate the function blocks used and adapt their parameters for each ATV32 address.

Appendices

Appendix A
Appendices

What Is in This Chapter?

This chapter contains the following topics:

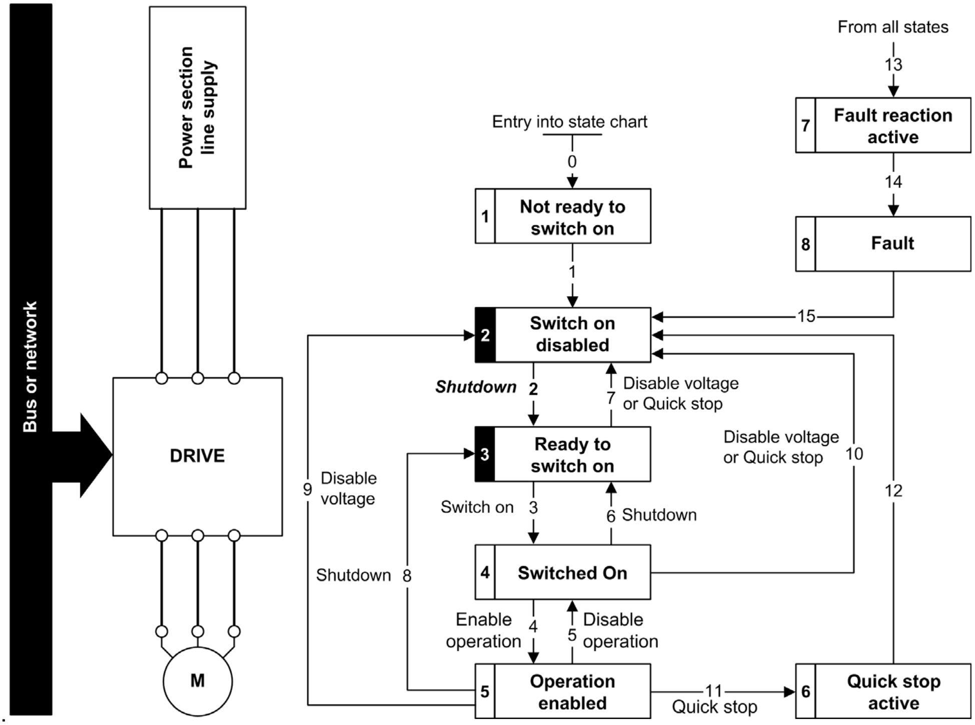
Sequence for a Drive Powered by the Power Section Line Supply

Presentation

The SoMachine Basic example implements the initialization steps required by the ATV32. Both the power and control sections are powered by the power section line supply. If power is supplied to the control section, it needs to be supplied to the power section as well.

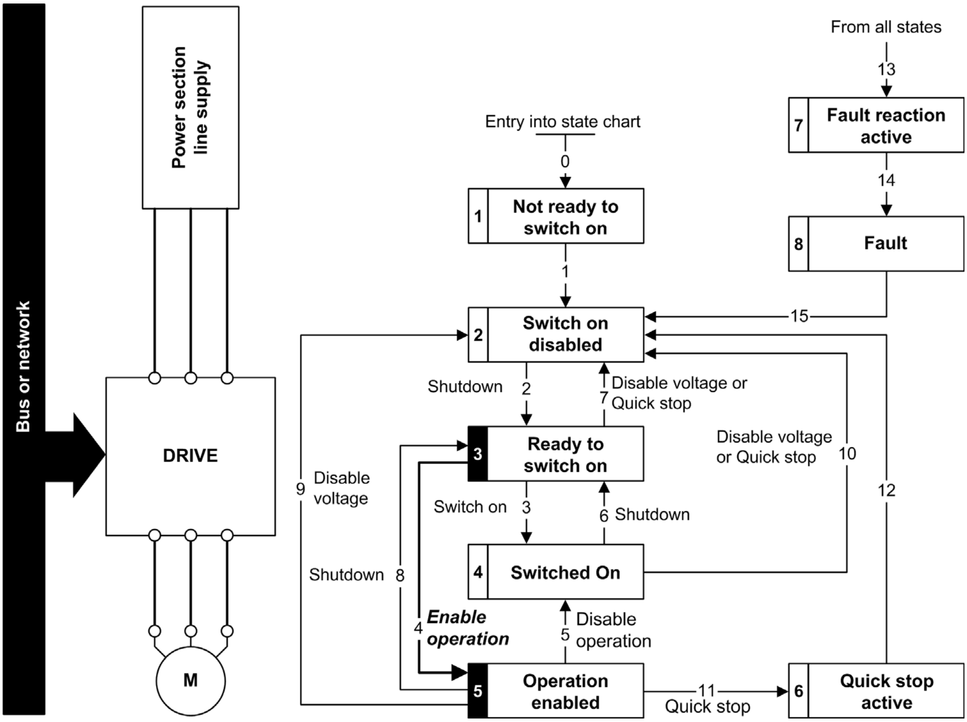
The sequence is as follows:

1. Set `%M0`.
2. Send **[2 - Shutdown]** command



3. Verify that the drive is in the **[3 - Ready to switch on]** state.

4. Send the **[4 - Enable operation]** command.



The motor can now be controlled (%M1 = 1).

5. Send a reference (%MW1) not equal to 0.

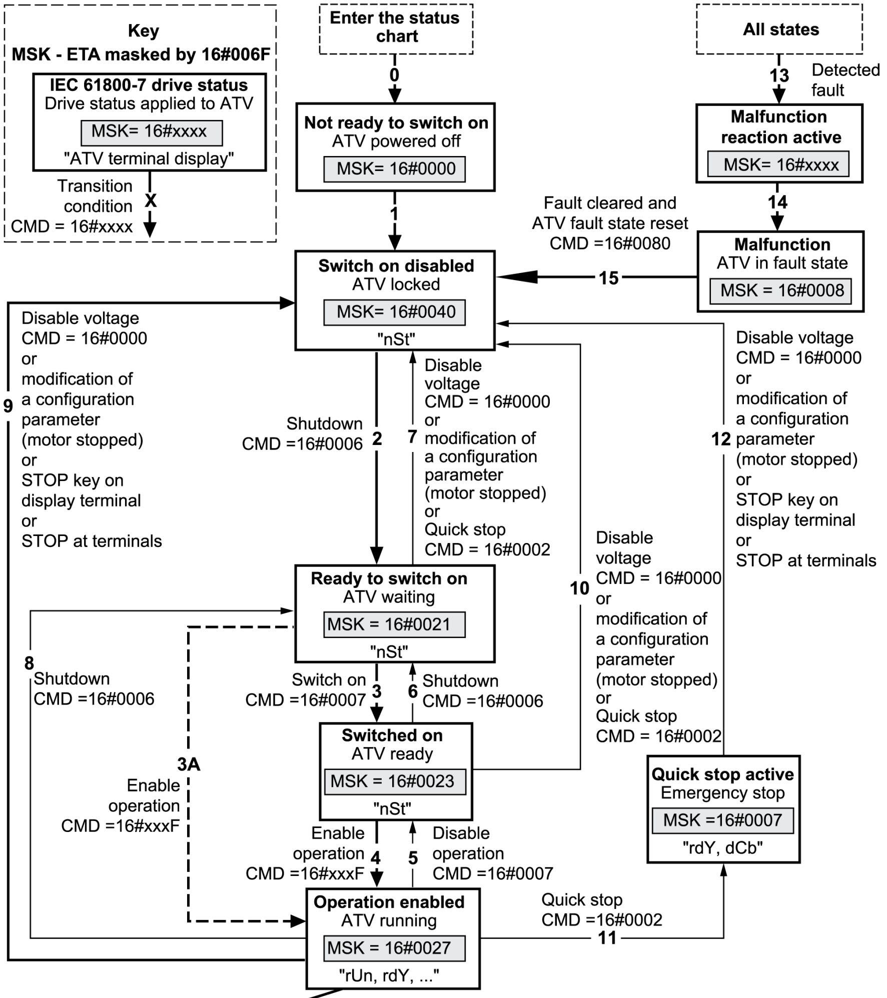
Common ATV32 Commands and Status

Common ATV32 Commands (CMD)

Command	CMD (hex)
[Shutdown]	0006
[Switch on]	0007
[Enable operation]	000F
[Disable operation]	0007
[Disable voltage]	0000
[Quick stop]	0002
[Fault reset]	0080

Refer to ATV32 documentation for other commands.

The following status chart includes the commands:


Key
MSK - ETA masked by 16#006F
IEC 61800-7 drive status
Drive status applied to ATV
MSK= 16#xxxx
"ATV terminal display"
Transition condition
X
CMD = 16#xxxx
Enter the status chart
0
Not ready to switch on
ATV powered off
MSK= 16#0000
1
All states
13
Detected fault
Malfunction reaction active
MSK= 16#xxxx
14
Malfunction
ATV in fault state
MSK = 16#0008
Fault cleared and ATV fault state reset CMD =16#0080
15
Switch on disabled
ATV locked
MSK= 16#0040
"nSt"
9
Disable voltage CMD = 16#0000 or modification of a configuration parameter (motor stopped) or STOP key on display terminal or STOP at terminals
Shutdown CMD =16#0006
2
7
Disable voltage CMD = 16#0000 or modification of a configuration parameter (motor stopped) or Quick stop CMD = 16#0002
12
Disable voltage CMD = 16#0000 or modification of a configuration parameter (motor stopped) or STOP key on display terminal or STOP at terminals
10
Disable voltage CMD = 16#0000 or modification of a configuration parameter (motor stopped) or Quick stop CMD = 16#0002
Ready to switch on
ATV waiting
MSK = 16#0021
"nSt"
8
Shutdown CMD =16#0006
3A
Enable operation CMD =16#xxxF
Switch on CMD =16#0007
3
6
Shutdown CMD =16#0006
Switched on
ATV ready
MSK = 16#0023
"nSt"
Enable operation CMD =16#xxxF
4
5
Disable operation CMD =16#0007
Quick stop active
Emergency stop
MSK =16#0007
"rdY, dCb"
Operation enabled
ATV running
MSK = 16#0027
"rUn, rdY, ..."
Quick stop CMD =16#0002
11

Examples:	Examples (default configuration):
ETA = 16#0627 : Normal stop or Forward operation, speed reached	CMD = 16#000F : Forward operation
ETA = 16#8627 : Reverse operation, speed reached	CMD = 16#080F : Reverse operation
ETA = 16#0227 : Forward operation, ACC or DEC	CMD = 16#100F : Stop (configured by "Stt")
ETA = 16#8227 : Reverse operation, ACC or DEC	CMD = 16#200F : DC injection stop
	CMD = 16#400F : Fast stop

Exiting the "Operation enabled" status via a "Disable voltage" (9) or "Shutdown" (8) command causes a freewheel stop.

Common ATV32 Status (ETA)

ETA (hex)	ATV32 status
xx40 or xx50	Switch on disabled
xx21or xx31	Ready to switch on
xx33	Switched on
xx37	Operation enabled
xx17	Quick stop active
xxx8 or xx28	Error detected

Refer to ATV32 documentation for other values.