

# EcoStruxure Machine Expert

## FastSampling

## Library Guide

EIO0000004407.00

03/2021

# Legal Information

The Schneider Electric brand and any trademarks of Schneider Electric SE and its subsidiaries referred to in this guide are the property of Schneider Electric SE or its subsidiaries. All other brands may be trademarks of their respective owners.

This guide and its content are protected under applicable copyright laws and furnished for informational use only. No part of this guide may be reproduced or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), for any purpose, without the prior written permission of Schneider Electric.

Schneider Electric does not grant any right or license for commercial use of the guide or its content, except for a non-exclusive and personal license to consult it on an "as is" basis. Schneider Electric products and equipment should be installed, operated, serviced, and maintained only by qualified personnel.

As standards, specifications, and designs change from time to time, information contained in this guide may be subject to change without notice.

To the extent permitted by applicable law, no responsibility or liability is assumed by Schneider Electric and its subsidiaries for any errors or omissions in the informational content of this material or consequences arising out of or resulting from the use of the information contained herein.

© 2021 Schneider Electric. All rights reserved.

# Table of Contents

Safety Information .....	5
QUALIFICATION OF PERSONNEL .....	5
PROPER USE .....	6
Before You Begin .....	6
Start-up and Test .....	7
Operation and Adjustments .....	7
About the Book .....	8
General Information .....	13
Presentation of the Library .....	14
General Information .....	14
Common Inputs and Outputs .....	16
Behavior of Function Blocks with the Input <i>i_xEnable</i> .....	16
Data Unit Types .....	17
Enumerations .....	18
<i>ET_Result</i> .....	18
<i>ET_Parameter</i> .....	19
<i>ET_PositionTrend</i> .....	19
Program Organization Units (POU) .....	20
Function Blocks .....	21
Calling Function Blocks and Start Trigger .....	21
Function Blocks Using Dynamic Memory Allocation .....	22
<i>FB_SamplingDataRealTime</i> .....	23
<i>FB_SamplingDataAsynchron</i> .....	25
Functions .....	27
<i>FC_EtResultToString</i> .....	27
Index .....	29



# Safety Information

## Important Information

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.

<b>⚠ DANGER</b>
<b>DANGER</b> indicates a hazardous situation which, if not avoided, <b>will result in</b> death or serious injury.
<b>⚠ WARNING</b>
<b>WARNING</b> indicates a hazardous situation which, if not avoided, <b>could result in</b> death or serious injury.
<b>⚠ CAUTION</b>
<b>CAUTION</b> indicates a hazardous situation which, if not avoided, <b>could result in</b> minor or moderate injury.
<b>NOTICE</b>
<b>NOTICE</b> 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

A qualified person is one who has the following qualifications:

- Skills and knowledge related to the construction and operation of electrical equipment and the installation.
- Knowledge and experience in industrial control programming.
- Received safety-related training to recognize and avoid the hazards involved.

The qualified person must be able to detect possible hazards that may arise from parameterization, modifying parameter values and generally from mechanical,

electrical, or electronic equipment. The qualified person must be familiar with the standards, provisions, and regulations for the prevention of industrial accidents, which they must observe when designing and implementing the system.

## PROPER USE

This product is a library to be used together with the control systems and servo amplifiers intended solely for the purposes as described in the present documentation as applied in the industrial sector.

Always observe the applicable safety-related instructions, the specified conditions, and the technical data.

Perform a risk evaluation concerning the specific use before using the product. Take protective measures according to the result.

Since the product is used as a part of an overall system, you must ensure the safety of the personnel by means of the design of this overall system (for example, machine design).

Any other use is not intended and may be hazardous.

## 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**

- 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 are 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.

# About the Book

## Document Scope

This document describes the FastSampling library. The library provides function blocks to sample drive data in a 125  $\mu$ s time grid.

## Validity Note

This document has been created for the release of EcoStruxure™ Machine Expert V2.0.

The characteristics that are described in the present document, as well as those described in the documents included in the Related Documents section below, can be found online. To access the information online, go to the Schneider Electric home page [www.se.com/ww/en/download/](http://www.se.com/ww/en/download/).

The characteristics that are described in the present document 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 document and online information, use the online information as your reference.

## Related Documents

Document title	Reference
EcoStruxure Machine Expert, Functions and Libraries User Guide	EIO0000002829 (ENG); EIO0000002830 (FRE); EIO0000002831 (GER); EIO0000002832 (ITA); EIO0000002833 (SPA); EIO0000002834 (CHS)
EcoStruxure Machine Expert, Programming Guide	EIO0000002854 (ENG); EIO0000002855 (FRE); EIO0000002856 (GER); EIO0000002857 (ITA); EIO0000002858 (SPA); EIO0000002859 (CHS)
Cybersecurity Guidelines for Machine Solutions	EIO0000004242



## 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.<sup>1</sup>
- 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.**

<sup>1</sup> 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.

Before you attempt to provide a solution (machine or process) for a specific application using the POUs found in the library, you must consider, conduct and complete best practices. These practices include, but are not limited to, risk analysis, functional safety, component compatibility, testing and system validation as they relate to this library.

### ▲ WARNING

#### IMPROPER USE OF PROGRAM ORGANIZATION UNITS

- Perform a safety-related analysis for the application and the devices installed.
- Ensure that the Program Organization Units (POUs) are compatible with the devices in the system and have no unintended effects on the proper functioning of the system.
- Use appropriate parameters, especially limit values, and observe machine wear and stop behavior.
- Verify that the sensors and actuators are compatible with the selected POUs.
- Thoroughly test all functions during verification and commissioning in all operation modes.
- Provide independent methods for critical control functions (emergency stop, conditions for limit values being exceeded, etc.) according to a safety-related analysis, respective rules, and regulations.

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

**⚠ 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.**

Incomplete file transfers, such as data files, application files and/or firmware files, may have serious consequences for your machine or controller. If you remove power, or if there is a power outage or communication interruption during a file transfer, your machine may become inoperative, or your application may attempt to operate on a corrupted data file. If an interruption occurs, reattempt the transfer. Be sure to include in your risk analysis the impact of corrupted data files.

**⚠ WARNING****UNINTENDED EQUIPMENT OPERATION, DATA LOSS, OR FILE CORRUPTION**

- Do not interrupt an ongoing data transfer.
- If the transfer is interrupted for any reason, re-initiate the transfer.
- Do not place your machine into service until the file transfer has completed successfully, unless you have accounted for corrupted files in your risk analysis and have taken appropriate steps to prevent any potentially serious consequences due to unsuccessful file transfers.

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

Care must be taken and provisions made for use of this library for machine control to avoid inadvertent consequences of commanded machine operation, state changes, or alteration of data memory or machine operating elements.

**⚠ WARNING****UNINTENDED EQUIPMENT OPERATION**

- Place operator devices of the control system near the machine or in a place where you have full view of the machine.
- Protect operator commands against unauthorized access.
- If remote control is a necessary design aspect of the application, ensure that there is a local, competent, and qualified observer present when operating from a remote location.
- Configure and install the Run/Stop input, if so equipped, or, other external means within the application, so that local control over the starting or stopping of the device can be maintained regardless of the remote commands sent to it.

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

## Terminology Derived from Standards

The technical terms, terminology, symbols and the corresponding descriptions in this manual, or that appear in or on the products themselves, are generally derived from the terms or definitions of international standards.

In the area of functional safety systems, drives and general automation, this may include, but is not limited to, terms such as *safety*, *safety function*, *safe state*, *fault*, *fault reset*, *malfunction*, *failure*, *error*, *error message*, *dangerous*, etc.

Among others, these standards include:

Standard	Description
IEC 61131-2:2007	Programmable controllers, part 2: Equipment requirements and tests.
ISO 13849-1:2015	Safety of machinery: Safety related parts of control systems. General principles for design.
EN 61496-1:2013	Safety of machinery: Electro-sensitive protective equipment. Part 1: General requirements and tests.
ISO 12100:2010	Safety of machinery - General principles for design - Risk assessment and risk reduction
EN 60204-1:2006	Safety of machinery - Electrical equipment of machines - Part 1: General requirements
ISO 14119:2013	Safety of machinery - Interlocking devices associated with guards - Principles for design and selection
ISO 13850:2015	Safety of machinery - Emergency stop - Principles for design
IEC 62061:2015	Safety of machinery - Functional safety of safety-related electrical, electronic, and electronic programmable control systems
IEC 61508-1:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: General requirements.
IEC 61508-2:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: Requirements for electrical/electronic/programmable electronic safety-related systems.
IEC 61508-3:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: Software requirements.
IEC 61784-3:2016	Industrial communication networks - Profiles - Part 3: Functional safety fieldbuses - General rules and profile definitions.
2006/42/EC	Machinery Directive
2014/30/EU	Electromagnetic Compatibility Directive
2014/35/EU	Low Voltage Directive

In addition, terms used in the present document may tangentially be used as they are derived from other standards such as:

Standard	Description
IEC 60034 series	Rotating electrical machines
IEC 61800 series	Adjustable speed electrical power drive systems
IEC 61158 series	Digital data communications for measurement and control – Fieldbus for use in industrial control systems

Finally, the term *zone of operation* may be used in conjunction with the description of specific hazards, and is defined as it is for a *hazard zone* or *danger zone* in the *Machinery Directive (2006/42/EC)* and *ISO 12100:2010*.

**NOTE:** The aforementioned standards may or may not apply to the specific products cited in the present documentation. For more information concerning the individual standards applicable to the products described herein, see the characteristics tables for those product references.



---

# General Information

## What's in This Part

Presentation of the Library .....	14
Common Inputs and Outputs.....	16

# Presentation of the Library

## What's in This Chapter

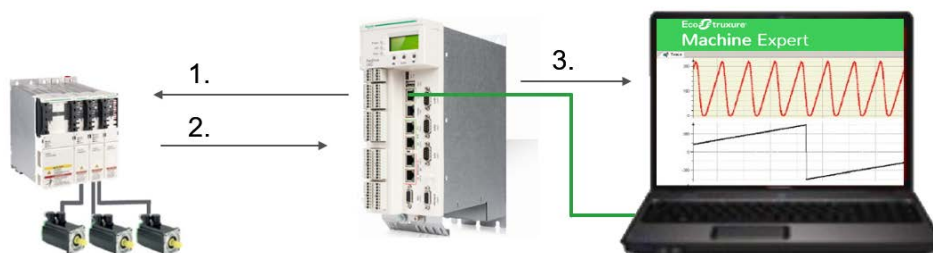
General Information..... 14

## General Information

### Library Overview

The FastSampling library provides objects for sampling data from Schneider Electric drive parameters with a higher resolution than provided by the Sercos cycle.

The figure illustrates the general data sampling process:



Process step	Description
1	Fast sampling parameters are configured in the controller and the sampling process is started.
2	Configured parameter data is retrieved from the drive.
3	The controller is stopping the sampling process, saves the sampled data to an ARRAY and provides them to EcoStruxure Machine Expert. They can be displayed in a individual IEC trace graph dedicated to fast sampling results.

## Data Sampling Methods

According to the performance provided by your system, retrieving sampled data from the drive can be performed in two different ways. A specific function block is provided for each sampling method:

Data Sampling Method	Function Block	Description
Asynchronous	<i>FB_SamplingDataAsynchron</i> , page 25	For systems providing restricted performance. Sampled data is buffered in the drive.
Real-time	<i>FB_SamplingDataRealTime</i> , page 23	For high-performance systems. Sampled data is transferred in a continuous stream by the Sercos real-time channel.

## Characteristics of the Library

The table indicates the characteristics of the library:

Characteristic	Value
Library title	FastSampling
Company	Schneider Electric
Category	<b>Application</b>
Component	<b>PacDrive</b>
Default namespace	<b>SE_FS</b>

Characteristic	Value
Language model attribute	Qualified-access-only (see EcoStruxure Machine Expert, Functions and Libraries User Guide)
Forward compatible library	Yes (see EcoStruxure Machine Expert, Functions and Libraries User Guide)

**NOTE:** For this library, qualified-access-only is set. The POU's, data structures, enumerations, and constants have to be accessed using the namespace of the library. The default namespace of the library is **SE\_FS**.

## Controller and Drive Platforms

The FastSampling library is supported by the following platforms:

- PacDrive LMC Eco and PacDrive LMC Pro/Pro2 motion controllers
- Lexium 62 Advanced and Lexium 62 Standard drives with hardware revision RS10 and higher

## General Considerations

**NOTE:** Schneider Electric adheres to industry best practices in the development and implementation of control systems. This includes a "Defense-in-Depth" approach to secure an Industrial Control System. This approach places the controllers behind one or more firewalls to restrict access to authorized personnel and protocols only.

<p><b>▲ WARNING</b></p> <p><b>UNAUTHENTICATED ACCESS AND SUBSEQUENT UNAUTHORIZED MACHINE OPERATION</b></p> <ul style="list-style-type: none"> <li>• Evaluate whether your environment or your machines are connected to your critical infrastructure and, if so, take appropriate steps in terms of prevention, based on Defense-in-Depth, before connecting the automation system to any network.</li> <li>• Limit the number of devices connected to a network to the minimum necessary.</li> <li>• Isolate your industrial network from other networks inside your company.</li> <li>• Protect any network against unintended access by using firewalls, VPN, or other, proven security measures.</li> <li>• Monitor activities within your systems.</li> <li>• Prevent subject devices from direct access or direct link by unauthorized parties or unauthenticated actions.</li> <li>• Prepare a recovery plan including backup of your system and process information.</li> </ul> <p><b>Failure to follow these instructions can result in death, serious injury, or equipment damage.</b></p>
--

For more information on organizational measures and rules covering access to infrastructures, refer to ISO/IEC 27000 series, Common Criteria for Information Technology Security Evaluation, ISO/IEC 15408, IEC 62351, ISA/IEC 62443, NIST Cybersecurity Framework, Information Security Forum - Standard of Good Practice for Information Security.

# Common Inputs and Outputs

## What's in This Chapter

Behavior of Function Blocks with the Input *i\_xEnable* ..... 16

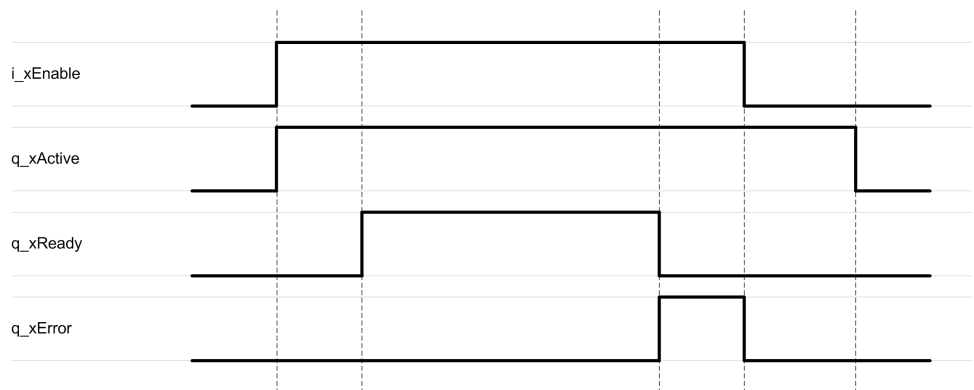
## Behavior of Function Blocks with the Input *i\_xEnable*

### General Information

By setting the input *i\_xEnable* to TRUE, the function block starts the enabling process. The function block continues initialization and the output *q\_xActive* is set to TRUE. Once the initialization is finished, the output *q\_xReady* is set to TRUE.

In case an error is detected, the output *q\_xError* remains TRUE until the function block is disabled.

### Example





---

# Data Unit Types

## What's in This Part

Enumerations .....	18
--------------------	----

# Enumerations

## What's in This Chapter

<i>ET_Result</i> .....	18
<i>ET_Parameter</i> .....	19
<i>ET_PositionTrend</i> .....	19

## ET\_Result

### Overview

Type:	Enumeration
Available as of:	V1.0.0.0

### Description

The enumeration *ET\_Result* specifies the possible values that indicate the result of operations executed by the POU's of this library.

### Enumeration Elements

Name	Value (UINT)	Description
<i>Disabled</i>	0	The function block is disabled.
<i>Ok</i>	1	The function block is being executed successfully.
<i>SercosReadError</i>	2	Unable to read the parameter of the drive.
<i>SercosWriteError</i>	3	Unable to write the parameter to the drive.
<i>TimeOut</i>	4	A timeout period has elapsed during execution.
<i>InputParameterInvalid</i>	5	A setting of an input variable is not valid. For further information, refer to the <i>q_sResultMsg</i> output.
<i>AsyncManagerError</i>	6	An exception has been detected by the internal asynchronous manager.
<i>NotReadyForSampling</i>	7	The drive configured for sampling data cannot execute the sampling process.
<i>BuildIDNError</i>	8	The Sercos parameter is not valid.
<i>DriveConditionInvalid</i>	9	The drive is not ready for sampling data.
<i>SercosNotInPhaseFour</i>	10	Sercos is not in phase 4. Thus, sampling data is not possible.
<i>SamplingIsBlocked</i>	11	Sampling cannot start because the drive is still busy with another sampling process.
<i>SamplingInDriveInError</i>	12	An error has been detected in the drive that is in sampling mode.
<i>RtpError</i>	13	An internal error has been detected.
<i>ControllerConditionInvalid</i>	14	The controller settings for fast sampling are not valid.
<i>SampledDataAvailable</i>	15	The data sampling process has been completed successfully. Collected data are available.
<i>SamplingData</i>	16	The data sampling process is being executed.
<i>IecTraceError</i>	17	An internal error has been detected.

## ET\_Parameter

### Overview

Type:	Enumeration
Available as of:	V1.0.0.0

### Description

The enumeration *ET\_Parameter* specifies the drive parameter that is sampled for a channel.

### Enumeration Elements

Name	Value (UDINT)	Description
<i>Disable</i>	0	No parameter of the drive is sampled on the channel.
<i>MechPosition</i>	1	Position parameter of the drive is sampled on the channel.
<i>MechVelocity</i>	2	Velocity parameter of the drive is sampled on the channel.
<i>Current</i>	3	Current parameter of the drive is sampled on the channel.
<i>FeedbackCurrent</i>	4	Feedback current parameter of the drive is sampled on the channel.

## ET\_PositionTrend

### Overview

Type:	Enumeration
Available as of:	V1.0.0.0

### Description

The enumeration *ET\_PositionTrend* specifies whether set positions of the axis are considered for sampling.

### Enumeration Elements

Name	Value (UDINT)	Description
<i>WithSetPos</i>	0	The set positions of the axis are considered for sampling. This results in a trace graph with setpos positions.
<i>Continuous</i>	1	The axis position value that is first sampled is calculated on actual mechanical position. This results in a continuous trace graph.

# Program Organization Units (POU)

## What's in This Part

Function Blocks.....	21
Functions.....	27

# Function Blocks

## What's in This Chapter

Calling Function Blocks and Start Trigger .....	21
Function Blocks Using Dynamic Memory Allocation .....	22
<i>FB_SamplingDataRealTime</i> .....	23
<i>FB_SamplingDataAsynchron</i> .....	25

## Calling Function Blocks and Start Trigger

### Calling the Function Blocks

Call the function blocks (*FB\_SamplingDataAsynchron* or *FB\_SamplingDataRealTime*) from the main task or a lower priority task.

### Calling the Start Trigger

Once the function block is enabled and ready for operation, the sampling process can be started by the property *xStartSampling* provided by the function block. This allows you to start the sampling process from a different task than the task where the function block is called. This can be an **External** task, if required by your application.

#### NOTE:

If you have configured a task of type **External** for triggering the property *xStartSampling*, consider that the event **MDT\_WRITE\_ACCESS** increases the delay for starting the sampling process by an additional Sercos cycle.

To reduce the delay to the minimum of 2 Sercos cycles + 250 µs, select an **External event** different than **MDT\_WRITE\_ACCESS**.

For further information about task configuration, refer to the section *Configuration of a Specific Task* in the EcoStruxure Machine Expert, Programming Guide.

### Behavior of the Property *xStartSampling*

Upon a rising edge at the property *xStartSampling*, the sampling process is initiated. The function block must be preconditioned and ready for operation and there must be no sampling process running.

You can stop a running sampling process by setting the input *i\_xEnable* of the function block to FALSE. Doing so, all present sampled data are deleted.

#### NOTE:

While the sampling process is in progress, an **Online Change** of the application cannot be performed. The sampling process is outsourced to a separate task. As long as this task is not completed, a requested online change is rejected. If this is the case, Logic Builder issues a message informing you that an online change cannot be performed.

### Displaying Sampled Data in a Trace Plot

To display sampled data in a trace plot in EcoStruxure Machine Expert, proceed as follows:

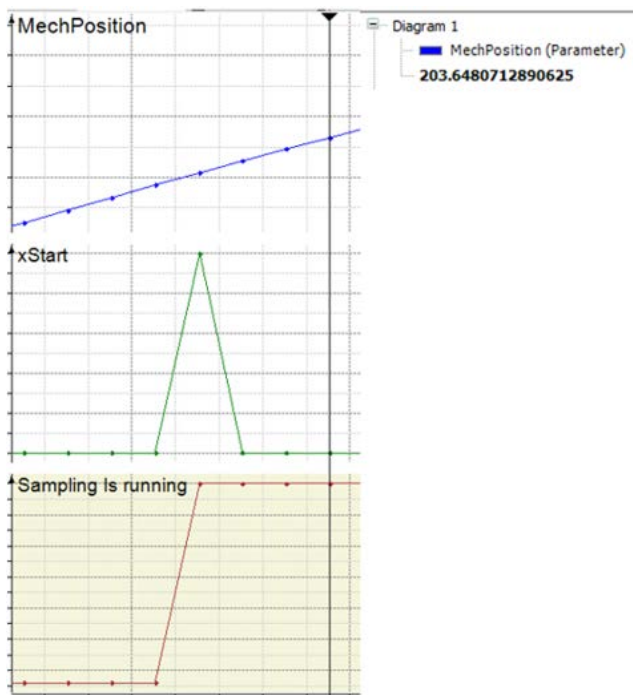
Step	Action
1	Right-click the node of the controller that is sampling data in the <b>Devices tree</b> and execute the command <b>Add Object &gt; DeviceTrace...</b> from the contextual menu.
2	Enter a name for the device trace and click <b>Add</b> .

Step	Action
	<b>Result:</b> A new device trace node is created as a subnode of the selected controller and the <b>DeviceTrace</b> editor opens on the right-hand side.
3	Right-click in the <b>DeviceTrace</b> editor and execute the command <b>Online List</b> . <b>Result:</b> The <b>Online list</b> dialog box opens that displays the traces that are existing on the device.
4	Select the function block that is sampling data and click the <b>Upload</b> command. <b>Result:</b> The selected trace is uploaded from the runtime system.
5	Click <b>OK</b> to close the <b>Online list</b> dialog box. <b>Result:</b> The sampled data are displayed as trace plot in the <b>DeviceTrace</b> editor.

**NOTE:**

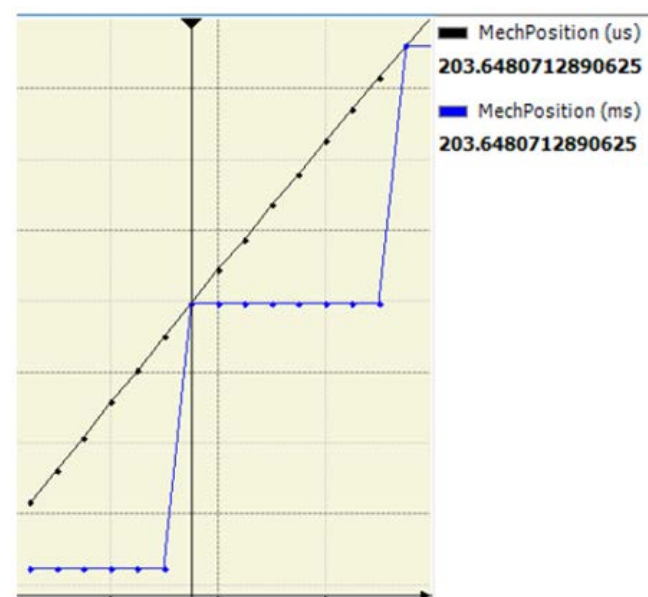
Based on the *MechPosition*, a delay of 2 Sercos cycles + 250  $\mu$ s is applied after the sampling process has been started.

The graphs illustrate the trace recording of *MechPosition* and the start of the sampling process:



- The green and the red graph indicate the start of the sampling process.
- The cursor cuts the blue graph at the value 203.648 units of the *MechPosition*. This value is recorded 3 Sercos cycles after the sampling process has been started.

The graphs illustrate a magnified section of the same *MechPosition* in a trace plot after the sampling process has been started:



- The black graph displays the sampled data in a 125  $\mu$ s time grid.
- The cursor cuts the blue graph at the same value 203.648 units of the *MechPosition*. The recording of data starts before 3 Sercos cycles are completed, exactly with a delay of 2 Sercos cycles + 250  $\mu$ s after the sampling process has been started.

## Function Blocks Using Dynamic Memory Allocation

### Overview

The function blocks *FB\_SamplingDataAsynchron* and *FB\_SamplingDataRealTime* allocate memory for the arrays of data sampled from the drives. To achieve this, activate the option **Use dynamic memory allocation** in the **View > Properties... > Application Build Options** tab in EcoStruxure

Machine Expert. Enter the **Maximum size of memory (bytes)** you want to reserve for the application.

For further information, refer to the *View Menu Commands — Properties...* section of the *Menu Commands Online Help*.

**NOTE:**

The maximum admissible value for the parameter **Maximum size of memory (bytes)** depends on the controller you are using, on the size of the present application and on other functionalities using this function. If there is not enough memory available, a diagnostic message is issued.

As long as the function blocks are disabled, memory is released and can be used elsewhere.

### Calculating the Memory Size

You can calculate the memory according to this example:

- Number of samples = 1000
- Number of channels = 4
- Datatype = LREAL (= 8 bytes)

The size of memory required for 1000 samples equals:

$$1000 \times 4 \times 8 = 32,000 \text{ bytes}$$

## FB\_SamplingDataRealTime

### Overview

Type:	Function block
Available as of:	V1.0.0.0



### Functional Description

The function block *FB\_SamplingDataRealTime* retrieves data from the drive in real-time. It must be called in a cyclic task. A continuous stream of data packages is transferred in 1 ms intervals. It contains data in a 125 μs time grid.



The number of samples to be recorded is limited to 80,000. Thus, the maximum sampling time is 10 s. Consider that the recording of 80,000 samples requires an ARRAY which is capable to hold 320,000 values of type LREAL (ARRAY [1..80000] OF ARRAY [0..3] OF LREAL). For the memory size, refer to Function Blocks Using Dynamic Memory Allocation, page 22.

After the function block is enabled and is ready for operation, the sampling process is started upon a rising edge at the *xStartSampling* property. The property can be located in a different task than the task where the function block is called.

To allow real-time sampling of data, perform the following settings in the configuration of the drive:

1. Double-click the drive node to open the drive editor.
2. Select the **Sercos Cyclic Data Exchange** tab.
3. Select the option **Enable expert settings**.
4. Click the plus sign (**Add a parameter**) from the **Acknowledge Telegram (Slave -> Master)** section on the right-hand side of the drive editor.

**Result:** The **Select parameters for cyclic exchange** dialog box opens.

5. Select the **FastSamplingChannels** row and click **OK**.

**Result:** A **FastSamplingChannels** row is added to the **Acknowledge Telegram (Slave -> Master)** section.

## Interface

Input	Data type	Description
<i>i_xEnable</i>	BOOL	Activation and initialization of the function block.  Refer to Behavior of Function Blocks with the Input <i>i_xEnable</i> , page 16.
<i>i_aetParameter</i>	ARRAY [0...3] OF <i>ET_Parameter</i>	Specify the parameters to be sampled.
<i>i_etPositionTrend</i>	<i>ET_PositionTrend</i> , page 19	Specifies whether set positions of the axis are considered for sampling.
<i>i_udiNumberOfSamples</i>	UDINT	Specifies the number of samples to be recorded.  Value range: 0...80,000  The samples are saved to an internal array in allocated memory. For further information, refer to Function Blocks Using Dynamic Memory Allocation, page 22.
<i>i_xUseTracePlot</i>	BOOL	If TRUE, the sampled data is plotted to an IEC trace graph, also refer to Displaying Sampled Data in a Trace Plot, page 21.
<i>i_ifDrive</i>	<i>SystemConfigurationIf.Drive</i>	Specifies the axis from which data is sampled.

Input/Output	Data type	Description
<i>iq_aalrSampledData</i>	ARRAY [*] OF ARRAY [0..3] OF LREAL	Array for storing the sampled data. The meaning and the order of the parameters per data record corresponds to the specified parameter at <i>i_aetParameter</i> . This IN_OUT parameter provides an array with variable length, the amount of data records you want to store must correspond to the size of the array assigned.

Output	Data type	Description
<i>q_xActive</i>	BOOL	If this output is set to TRUE, the function block is active.
<i>q_xReady</i>	BOOL	If this output is set to TRUE, the activation was successful.
<i>q_xError</i>	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to <i>q_etResult</i> and <i>q_sResultMsg</i> .
<i>q_xSamplingDone</i>	BOOL	If this output is set to TRUE, the sampling process is finished and the recorded data are provided in the array at <i>iq_aalrSampledData</i> .



Output	Data type	Description
<i>q_etResult</i>	<i>ET_Result</i>	Provides diagnostic and status information as a numeric value.
<i>q_sResultMsg</i>	STRING [80]	Provides additional diagnostic and status information as a text message.

Property	Data type	Access	Description
<i>xStartSampling</i>	BOOL	Read/write	Input that initiates the start of the sampling process.  The input can be called in a fast task. For further information, refer to Behavior of the Property <i>xStartSampling</i> , page 21.

**NOTE:**

While the sampling process is in progress, an **Online Change** of the application cannot be performed. The sampling process is outsourced to a separate task. As long as this task is not completed, a requested online change is rejected. If this is the case, Logic Builder issues a message informing you that an online change cannot be performed.

## FB\_SamplingDataAsynchron

### Overview

Type:	Function block
Available as of:	V1.0.0.0



### Functional Description

The function block *FB\_SamplingDataAsynchron* retrieves data from the drive in asynchronous mode. It must be called in a cyclic task. The duration of the sampling process is fixed to 100 ms, accordingly it provides 800 data records. The sampled data are stored on the drive. When the sampling process is finished, the application is retrieving the single packages from the drive in a certain interval. After the packages have been retrieved, the function block indicates that the sampling process is done and provides the recorded data in the array at *iq\_aalrSampledData*.



The *xStartSampling* property is additionally provided to start the sampling process. For a quick start of the sampling process, it can be called in a fast task.

No additional configuration is necessary to use asynchronous sampling.

## Interface

Input	Data type	Description
<i>i_xEnable</i>	BOOL	Activation and initialization of the function block. Refer to Behavior of Function Blocks with the Input <i>i_xEnable</i> , page 16.
<i>i_aetParameter</i>	ARRAY [0...3] OF <i>ET_Parameter</i>	Specify the parameters to be sampled.
<i>i_etPositionTrend</i>	<i>ET_PositionTrend</i> , page 19	Specifies whether set positions of the axis are considered for sampling.
<i>i_xUseTracePlot</i>	BOOL	If TRUE, the sampled data is plotted to an IEC trace graph, also refer to Displaying Sampled Data in a Trace Plot, page 21.
<i>i_ifDrive</i>	<i>SystemConfigurationItf.IF_Drive</i>	Specifies the axis from which data is sampled.

Input/Output	Data type	Description
<i>iq_aalrSampledData</i>	ARRAY [0...799] OF ARRAY [0...3] OF REAL	Array for storing the sampled data. The meaning and the order of the parameters per data record correspond to the specified parameter at <i>i_aetParameter</i> .

Output	Data type	Description
<i>q_xActive</i>	BOOL	If this output is set to TRUE, the function block is active.
<i>q_xReady</i>	BOOL	If this output is set to TRUE, the activation was successful.
<i>q_xError</i>	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to <i>q_etResult</i> and <i>q_sResultMsg</i> .
<i>q_xSamplingDone</i>	BOOL	If this output is set to TRUE, the sampling process is finished and the recorded data are provided in the array at <i>iq_aalrSampledData</i> .
<i>q_etResult</i>	<i>ET_Result</i>	Provides diagnostic and status information as a numeric value.
<i>q_sResultMsg</i>	STRING [80]	Provides additional diagnostic and status information as a text message.

Property	Data type	Access	Description
<i>xStartSampling</i>	BOOL	Read/write	Input that initiates the start of the sampling process.  The input can be called in a fast task. For further information, refer to Behavior of the Property <i>xStartSampling</i> , page 21.

### NOTE:

While the sampling process is in progress, an **Online Change** of the application cannot be performed. The sampling process is outsourced to a separate task. As long as this task is not completed, a requested online change is rejected. If this is the case, Logic Builder issues a message informing you that an online change cannot be performed.

# Functions

## What's in This Chapter

*FC\_EtResultToString* .....27

## *FC\_EtResultToString*

### Overview

Type:	Function
Available as of:	V1.0.0.0
Inherits from:	-
Implements:	-

### Task

Convert an enumeration element of type *ET\_Result* to a variable of type STRING.

### Functional Description

Using the function *FC\_EtResultToString*, you can convert an enumeration element of type *ET\_Result* to a variable of type STRING.

### Interface

Input	Data type	Description
<i>i_etVar</i>	<i>ET_Result</i>	Enumeration to be converted.

### Return Value

Data type	Description
STRING[80]	The <i>ET_Result</i> converted to text.



# Index

## C

common inputs and outputs	
behavior of function blocks with the input <i>i_</i>	
<i>xEnable</i> .....	16

## E

<i>ET_Parameter</i> .....	19
<i>Current</i> .....	19
<i>Disable</i> .....	19
<i>FeedbackCurrent</i> .....	19
<i>MechPosition</i> .....	19
<i>MechVelocity</i> .....	19
<i>ET_PositionTrend</i> .....	19
<i>Continuous</i> .....	19
<i>WithSetPos</i> .....	19
<i>ET_Result</i> .....	18
<i>AsyncManagerError</i> .....	18
<i>BuildIDNError</i> .....	18
<i>ControllerConditionInvalid</i> .....	18
<i>Disabled</i> .....	18
<i>DriveConditionInvalid</i> .....	18
<i>IecTraceError</i> .....	18
<i>InputParameterInvalid</i> .....	18
<i>NotReadyForSampling</i> .....	18
<i>Ok</i> .....	18
<i>RtpError</i> .....	18
<i>SampledDataAvailable</i> .....	18
<i>SamplingData</i> .....	18
<i>SamplingInDriveInError</i> .....	18
<i>SamplingIsBlocked</i> .....	18
<i>SercosNotInPhaseFour</i> .....	18
<i>SercosReadError</i> .....	18
<i>SercosWriteError</i> .....	18
<i>TimeOut</i> .....	18

## F

<i>FB_SamplingDataAsynchron</i> .....	25
<i>FB_SamplingDataRealTime</i> .....	23
<i>FC_EtResultToString</i> .....	27

Schneider Electric  
35 rue Joseph Monier  
92500 Rueil Malmaison  
France

+ 33 (0) 1 41 29 70 00

[www.se.com](http://www.se.com)

As standards, specifications, and design change from time to time,  
please ask for confirmation of the information given in this publication.

© 2021 – Schneider Electric. All rights reserved.

EIO0000004407.00