X20AI2636

Data sheet
2.21 (June 2025)

Translation of the original documentation

Publishing information

B&R Industrial Automation GmbH

B&R Strasse 1

5142 Eggelsberg

Austria

Telephone: +43 7748 6586-0

Fax: +43 7748 6586-26

office@br-automation.com

Disclaimer

1282159634660-2.21

1 General information

1.1 Other applicable documents

For additional and supplementary information, see the following documents.

Other applicable documents

Document name	Title
MAX20	X20 System user's manual

1.2 Order data

Order number	Short description	Figure
	Analog input modules	
X20AI2636	X20 analog input module, 2 inputs, ±10 V or 0 to 20 mA, 16-bit converter resolution, configurable input filter, oversampling functions	
	Required accessories	
	Bus modules	
X20BM11	X20 bus module, 24 VDC keyed, internal I/O power supply connected through	
X20BM15	X20 bus module, with node number switch, 24 VDC keyed, internal I/O power supply connected through	
	Terminal blocks	
X20TB06	X20 terminal block, 6-pin, 24 VDC keyed	
X20TB12	X20 terminal block, 12-pin, 24 VDC keyed	

Table 1: X20AI2636 - Order data

1.3 Module description

The module is equipped with 2 inputs with 16-bit digital converter resolution. It is possible to select between the current and voltage signal using different terminals.

This module is designed for X20 6-pin terminal blocks. If needed (e.g. for logistical reasons), the 12-pin terminal block can also be used.

Functions:

- Physical values
- Logical values
- Oversampling
- Monitoring the inputs

Physical values

The conversion results of the analog inputs are scaled and filtered before being transferred to the higher-level system.

Logical values

The physical values can be further processed using mathematical functions and comparators. Another logical channel can also be used as a starting point for further processing for a logical function.

Oversampling

The input values of the enabled channels are stored in the module in a configurable interval independently of the X2X cycle. The memory depth is 16 analog values per physical and logical channel.

Monitoring the input signal

The input signal of the analog inputs is monitored for out-of-range, for upper and lower limit values and for filter errors.

2 Technical description

2.1 Technical data

Order number	X20AI2636
Short description	
I/O module	2 analog inputs ±10 V or 0 to 20 mA
General information	
B&R ID code	0xB3A7
Status indicators	I/O function per channel, operating state, module status
Diagnostics	
Module run/error	Yes, using LED status indicator and software
Inputs	Yes, using LED status indicator and software
Channel type	Yes, using software
Power consumption	
Bus	0.01 W
Internal I/O	1.2 W [1)]
Additional power dissipation caused by actuators (resistive) [W]	-
Certifications	
CE	Yes
UKCA	Yes
ATEX	Zone 2, II 3G Ex nA nC IIA T5 Gc IP20, Ta (see X20 user's manual) FTZÚ 09 ATEX 0083X
UL	cULus E115267 Industrial control equipment
HazLoc	cCSAus 244665 Process control equipment for hazardous locations Class I, Division 2, Groups ABCD, T5
DNV	Temperature: **B** (0 to 55°C) Humidity: **B** (up to 100%) Vibration: **B** (4 g) EMC: **B** (bridge and open deck)
CCS	Yes
LR	ENV1
KR	Yes
ABS	Yes
BV	**EC33B** Temperature: 5 - 55°C Vibration: 4 g EMC: Bridge and open deck
KC	Yes
Analog inputs	
Input	±10 V or 0 to 20 mA, via different terminal connections
Input type	Differential input
Digital converter resolution	
Voltage	±15-bit
Current	15-bit
Conversion time	40 µs for all inputs
Output format	INT
Output format	
Voltage	INT 0x8001 - 0x7FFF / 1 LSB = 0x0001 = 305.176 µV
Current	INT 0x0000 - 0x7FFF / 1 LSB = 0x0001 = 610.352 nA
Input impedance in signal range	
Voltage	20 MΩ
Current	-
Load	
Voltage	-
Current	<400 Ω
Input protection	Protection against wiring with supply voltage
Permissible input signal	
Voltage	Max. ±30 V
Current	Max. ±50 mA

Table 2: X20AI2636 - Technical data

Order number	X20AI2636
Output of digital value during overload	
Undershoot	
Voltage	0x8001
Current	0x0000
Overshoot	
Voltage	0x7FFF
Current	0x7FFF
Conversion procedure	SAR
Input filter	Hardware - Third-order low-pass filter / cutoff frequency 10 kHz
Max. error	
Voltage	
Gain	0.08% 2)
Offset	0.01% 3)
Current	
Gain	0.08% 2)
Offset	0.02% 4)
Max. gain drift	
Voltage	0.01%/°C 2)
Current	0.01%/°C 2)
Max. offset drift	
Voltage	0.001%/°C 3)
Current	0.002%/°C 4)
Common-mode rejection	
DC	70 dB
50 Hz	70 dB
Common-mode range	±12 V
Crosstalk between channels	<-70 dB
Nonlinearity	
Voltage	<0.01% 3)
Current	<0.015% 4)
Insulation voltage between channel and bus	500 V_{eff}
Electrical properties	
Electrical isolation	Channel isolated from bus Channel not isolated from channel
Operating conditions	
Mounting orientation	
Horizontal	Yes
Vertical	Yes
Installation elevation above sea level	
0 to 2000 m	No limitation
>2000 m	Reduction of ambient temperature by 0.5°C per 100 m
Degree of protection per EN 60529	IP20
Ambient conditions	
Temperature	
Operation	
Horizontal mounting orientation	-25 to 60°C
Vertical mounting orientation	-25 to 50°C
Derating	-
Storage	-40 to 85°C
Transport	-40 to 85°C
Relative humidity	
Operation	5 to 95%, non-condensing
Storage	5 to 95%, non-condensing
Transport	5 to 95%, non-condensing
Mechanical properties	
Note	Order 1x terminal block X20TB06 or X20TB12 separately. Order 1x bus module X20BM11 separately.
Pitch	$12.5^{+0.2}$ mm

Table 2: X20AI2636 - Technical data

1) To reduce power dissipation, B&R recommends bridging unused inputs on the terminal.
2) Based on the current measured value.
3) Based on the 20 V measurement range.
4) Based on the 20 mA measurement range.

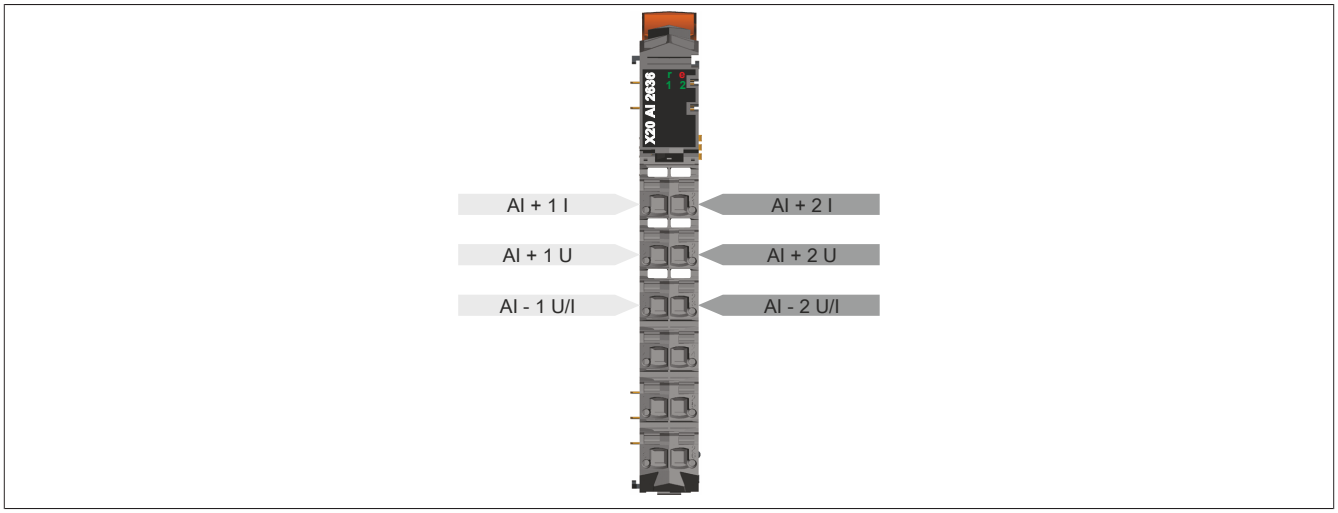
2.2 LED status indicators

For a description of the various operating modes, see section "Additional information - Diagnostic LEDs" in the X20 system user's manual.

Figure	LED	Color	Status	Description
	r	Green	Off	No power to module
			Single flash	RESET mode
			Double flash	BOOT mode (during firmware update)[1]
			Blinking	PREOPERATIONAL mode
			On	RUN mode
	e	Red	Off	No power to module or everything OK
			On	Error or reset status
			Double flash	System error: • Violation of the scan time • Synchronization error
	1 - 2	Green	Off	Open line[2] or sensor is disconnected
			Blinking	Channel error: Underflow, overflow or broken connection
			On	Analog/digital converter running, value OK

1) Depending on the configuration, a firmware update can take up to several minutes.
2) Open line detection only possible when measuring voltage.

2.3 Pinout

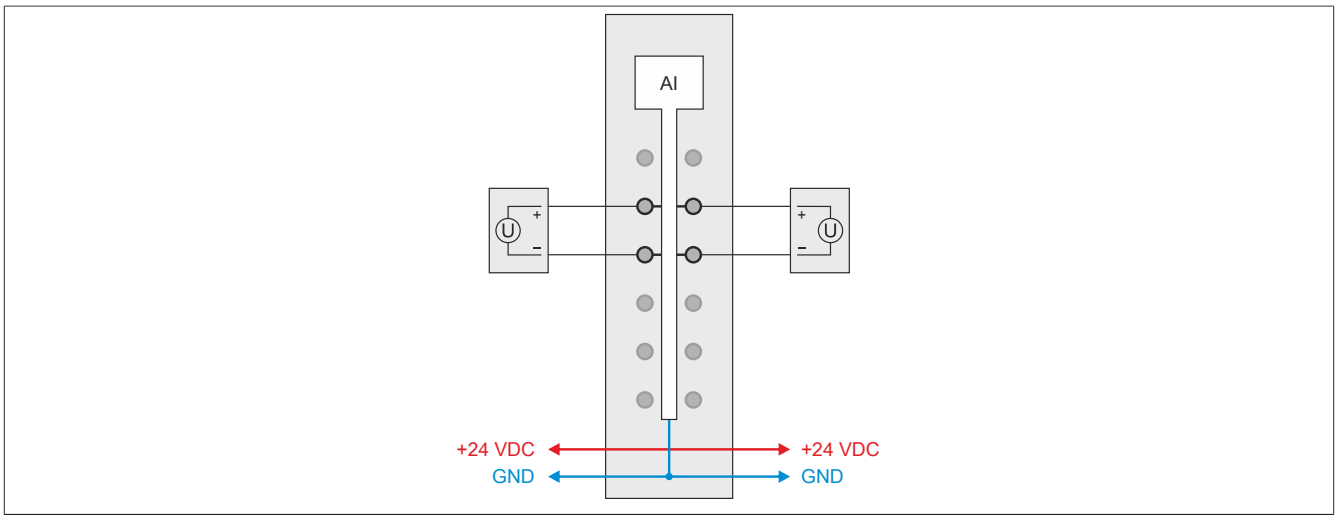


2.4 Connection example

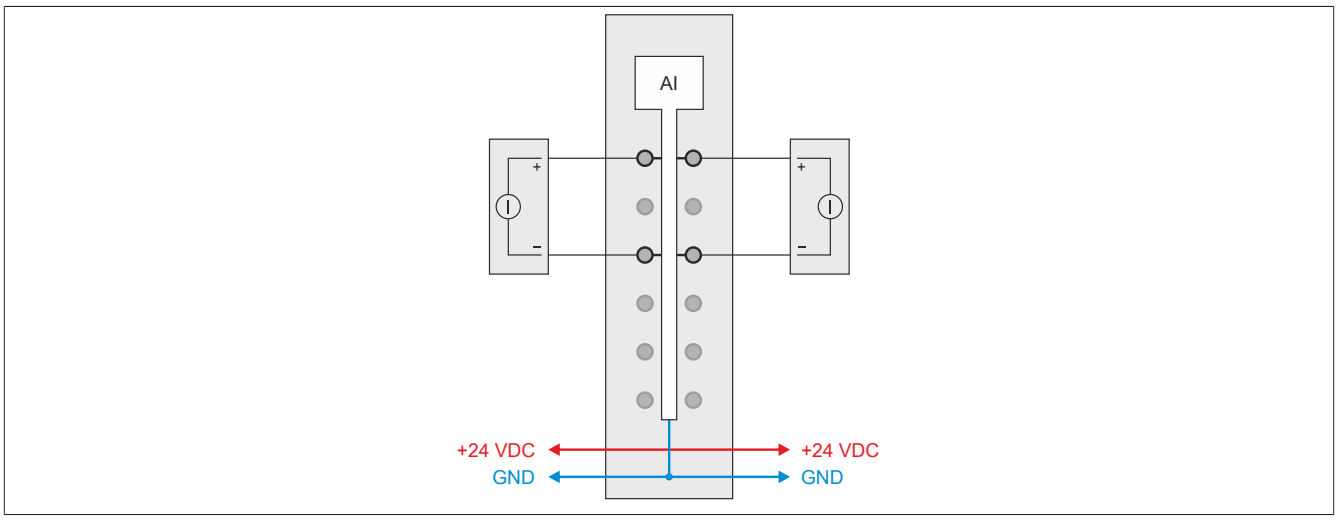
To prevent disturbances, the following modules must be separated by at least one module:

- Bus receiver X20BR9300
- Supply module X20PS3300/X20PS3310
- Supply module X20PS9400/X20PS9402
- Power supply module X20PS9600/X20PS9602
- Controller

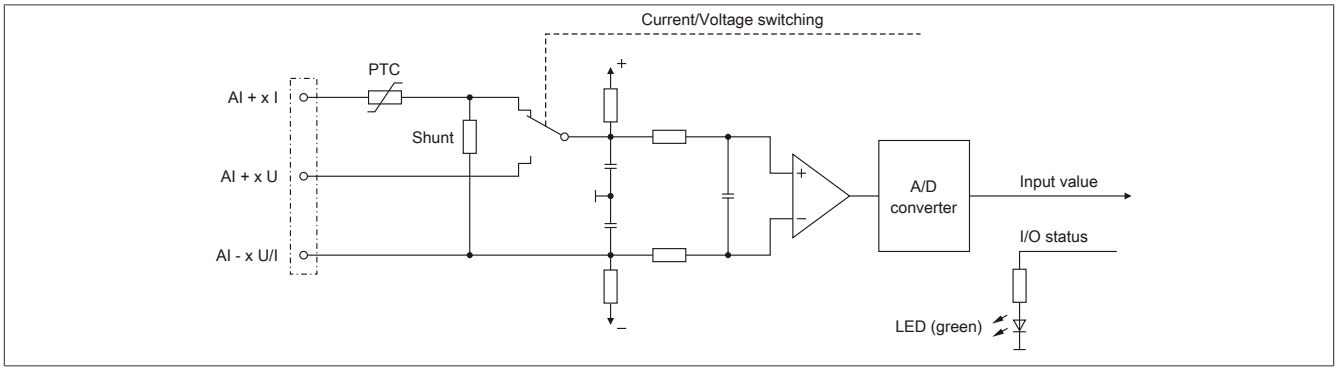
Voltage measurement



Current measurement



2.5 Input circuit diagram



3 Function description

3.1 Operating modes

3.1.1 Standard - Oversampling

The input values are recorded with a configurable sampling cycle time and saved with timestamp to the internal physical data buffer. This data range can then be read out in the cyclic data transfer using a configurable data length.

The recording and transmission system for the logical channels is identical to that for the physical channels. The functions of the logical channels are also executed in the configured sampling cycle time and saved with timestamp to the logical data buffer. The values can also be read out from here using configurable cyclic data points.

The defined sampling cycle time may not be sufficient for the sum of all physical and logical functions if using fast X2X Link cycle times, however. If influencing the physical sampling is not permitted, then a prescaler can be used to slow down the logical processing.

Information:

The ability to adjust the sampling cycle time as needed on the module means there is basically no synchronization with X2X Link, regardless of whether standard inputs or an oversampling function is configured.

If synchronization is required, then the configured sampling cycle time must be a multiple of the X2X Link cycle time!

3.1.2 Bus controller

The module is operated as a normal analog input module in this mode. The input values are measured with a configurable sampling cycle time and saved in the internal physical data buffer with timestamp. Only the latest value is transferred in the next possible bus cycle.

It is possible to assign a logical function directly to each input channel, however. The analog data on the bus controller is mapped using the calculation options of the logical channels and configured automatically, see "Operation in the standard function model" on page 16.

Function model "Bus controller" has the following limitations compared to function model Standard:

- No oversampling function since consistency is not possible due to the small data range when operating on CAN-based bus controllers
- The sampling cycle time is set to 100 µs.
- No timestamp function
- A selection of logical functions is available with which the physical values can already be processed on the module:
  - Physical value output (standard)
  - Addition of two channels with scaling
  - Integral addition of two channels with scaling
  - Multiplication of two channels with scaling
  - Integral multiplication of two channels with scaling

3.2 Oversampling

3.2.1 Analog oversampling

With analog oversampling, the enabled channels are stored in the module in a configurable interval independently of the X2X cycle. The memory depth is 16 analog values per physical and logical channel.

These samplings are numbered from 1 to 16 in the registers. The conversions or calculations of the individual channels with the same number (i.e. sample line 1 to 16, e.g. PhysCh01Sample**10**, PhysCh02Sample**10**) originate from the same sampling cycle or logical calculation cycle and therefore have the same timestamp.

The timestamp refers to the newest data value, i.e. always to sample line 1. If a timestamp for older data points is needed, it needs to be back-calculated in the application using the sampling cycle time configured on the module. The prescaler must also be taken into account for logical channels.

Calculation example

Sample line	Calculation	
1	Timestamp	**Newest value**
2	Timestamp - Sampling cycle time	
3	Timestamp - 2 * Sampling cycle time	
4	Timestamp - 3 * Sampling cycle time	
...	...	
10	Timestamp - 9 * Sampling cycle time	
...	...	
16	Timestamp - 15 * Sampling cycle time	**Oldest value**

How the buffer is organized can be seen from this. This is not a FIFO buffer but a static buffer that the values are pushed through. Sample line 1 always contains the newest values, the next line the second newest, all the way up to sample line 16, which contains the oldest values.

The sample counter is a circular counter, with the number of new sample lines derived from the value of the last transfer cycle.

Example

A difference of 3 to the last transfer cycle means:

The data in sample line 1 and all subsequent data from the previous transfer cycle is now shifted in the current cycle beginning with sample line 4. Sample lines 1 through 3 contain the new values for further processing by the application. Sample lines 14 through 16 from the last transfer cycle are no longer in the buffer.

3.2.2 Comparator oversampling

With comparator oversampling, the results of the enabled channels are stored in the module in a configurable interval independently of the X2X cycle. The memory depth is 16 bits per logical channel.

These samplings, i.e. the result bits, are numbered consecutively from 1 to 8 and 9 to 16 for the two registers. The results of the individual channels with the same number (i.e. sample line 1 to 16, e.g. LogicCh01Sample**16_9**, LogicCh01Sample**8_1** for channel 1) originate from the same sampling cycle or logical calculation cycle and therefore have the same timestamp.

The timestamp refers to the latest data value, so always to sample line 1, i.e. bit 0 in register "LogicCh01Sample8_1". If a timestamp for the older comparator results is needed, this must be recalculated in the application using the sampling cycle time set on the module. The prescaler setting must also be taken into account.

Calculation example

Sample line	(register name)	Calculation	
1	(LogicCh01Sample8_1 bit 0)	Timestamp	**Newest value**
2	(LogicCh01Sample8_1 bit 1)	Timestamp - Sampling cycle time	
3	(LogicCh01Sample8_1 bit 2)	Timestamp - 2 * Sampling cycle time	
4	(LogicCh01Sample8_1 bit 3)	Timestamp - 3 * Sampling cycle time	
...			
10	(LogicCh01Sample16_9 bit 1)	Timestamp - 9 * Sampling cycle time	
...			
16	(LogicCh01Sample16_9 bit 7)	Timestamp - 15 * Sampling cycle time	**Oldest value**

How the buffer is organized can be seen from this. This is not a FIFO buffer but a static buffer that the values are pushed through. Sample line 1 always contains the newest values, the next line the second newest, all the way up to sample line 16, which contains the oldest values.

The sample counter is a circular counter, with the number of new sample lines derived from the value of the last transfer cycle.

Example

A difference of 3 to the last transfer cycle means:

The comparator result in sample line 1 and all subsequent data from the previous transfer cycle is now shifted in the current cycle beginning with sample line 4. Sample lines 1 through 3 contain the new bit values for further processing by the application. Sample lines 14 through 16 from the last transfer cycle are no longer in the buffer.

3.2.3 Priorities and values

Priority of logical oversampling

- Low priority setting
  Preparation of the logical and physical buffer does not run in the same context. If the calculation time in logical oversampling is longer than the set sampling cycle time, this setting and a prescaler >1 can be used to split the logical processing over several sampling cycle times. The sample lines of the physical and logical oversampling therefore do not automatically have the same recording or calculation time. If the prescaler is configured incorrectly, logical oversampling cannot be processed successfully.
- High priority setting
  The logical and physical buffers are prepared in the same context. The sample lines of the physical and logical oversampling have the same recording or calculation time. All configured functions must be able to be executed in the set sampling cycle time; otherwise, a cycle time violation will occur and the configuration must be changed accordingly. The setting of the logical prescaler has no influence here; only the data volume in logical oversampling is limited.

Current or referenced values for logical or physical oversampling

In a busy system, jitter can occur in the sampling cycle on the module even with synchronous cycle time settings due to the necessary processing of the functions (X2X Link operation, logical and physical oversampling). This results in a different number of sample lines in the same time frames. For this reason, more samples should be configured in the cyclic image as well than is mathematically necessary.

- Current values setting
  The sample lines are transferred to the higher-level system as quickly as possible, with more or fewer new sample lines possibly occurring.
- Referenced values setting
  With this setting, jitter is minimized and a constant number of new sample lines per cycle is achieved with the optimal setting. With regard to response time, however, there may be delays of several sampling cycle times.

Information:

The register is described in "Logical oversampling and data acquisition" on page 26.

3.2.4 Data transfer

The analog conversion rate / sampling cycle time can be considerably faster than the X2X Link cycle. Saved analog or comparator data can be transferred to the higher-level system synchronously and consistently.

In terms of the application, it must be ensured that the ratio of cyclic data points, the sampling cycle time on the module and the transfer time is sufficient to read out all new data points in the higher-level system.

The sample counter can be used to check how many data values are actually new since the last transfer cycle. If the counter difference to the previous cycle is larger than the number of existing cyclic data points, then values have been overlooked and the system needs to be adjusted.

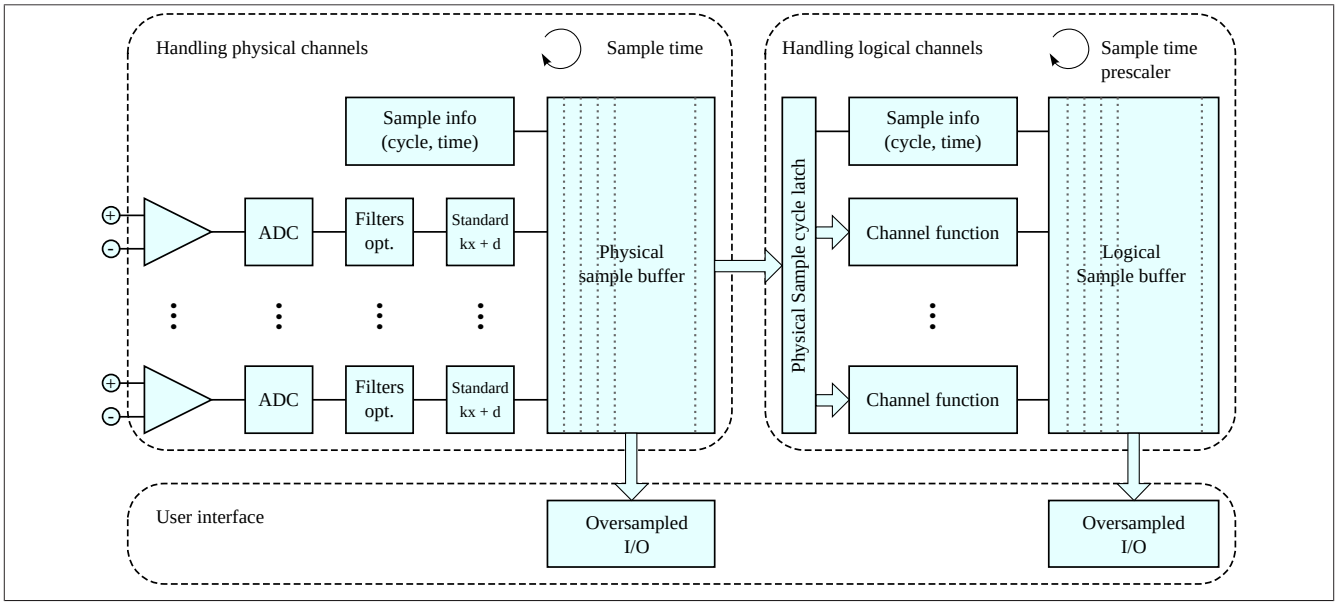
The general guideline is that a cyclic data point should be configured more than is actually required computing-wise.

Example with synchronous settings

- Sampling cycle time = 50 µs
- X2X Link cycle time = 500 µs

Samples 1 to 10 of a channel are possible to calculate in this example. Sample 11 should also be configured as a cyclic data point, however.
The reason for this is the possible jitter in the module caused by interruptions, e.g. from the X2X Link transfer. For the current cycle, this can mean that only 9 new values are available and that 11 values will have to be transferred in the next cycle.

For logical comparator functions, this problem doesn't exist since the maximum number is always transferred in the cycle data range.



3.3 Physical values

The conversion results are scaled and filtered before being transferred to the higher-level system. No further processing takes place.

Information:

The registers are described in "Physical configuration" on page 29.

3.3.1 Physical sampling

This module has a data buffer with 16 entries for each of the physical input channels. This buffer is processed according to the configured sampling cycle time.

A maximum of only 30 bytes is available for cyclic transfer on the X2X bus, however. Minus the status and sample counter, this allows only a selection of 14 samples (with a 16-bit data width) from the physical and logical buffer to be transferred.
Data loss can therefore occur with an imprecise selection and configuration.

Example

Displaying continuous sample lines.

- Sampling cycle time = 100 µs
- X2X cycle time = 500 µs

Sample line 1	PhysCh0xSample1
Sample line 2	PhysCh0xSample2
Sample line 3	PhysCh0xSample3
Sample line 4	PhysCh0xSample4
Sample line 5	PhysCh0xSample5
Sample line 6	PhysCh0xSample6
Difference SampleCount = 1	New value in sample line 1
Difference SampleCount = 2	New values in sample line 1 and sample line 2
...	
Difference SampleCount = 5	New values in sample line 1 to sample line 5

Information:

It is important to note that the sample counter refers to the update of the sample lines in the data buffer and not to the number of values transferred cyclically.

Display every second sample line to bridge a longer recording duration:

- Sampling cycle time = 100 µs
- X2X cycle time = 1000 µs

Sample line 1	PhysCh0xSample1
Sample line 3	PhysCh0xSample3
Sample line 5	PhysCh0xSample5
Sample line 7	PhysCh0xSample7
Sample line 9	PhysCh0xSample9
Sample line 11	PhysCh0xSample11
Difference SampleCount = 1	New value in sample line 1
Difference SampleCount = 3	New values in sample line 1 and sample line 3
...	
Difference SampleCount = 5	New values in sample line 1 to sample line 5
...	
Difference SampleCount = 9	New values in sample line 1 to sample line 9

3.3.2 Input filter

This module is equipped with an individually configurable input filter for each channel. The following filters can be selected:

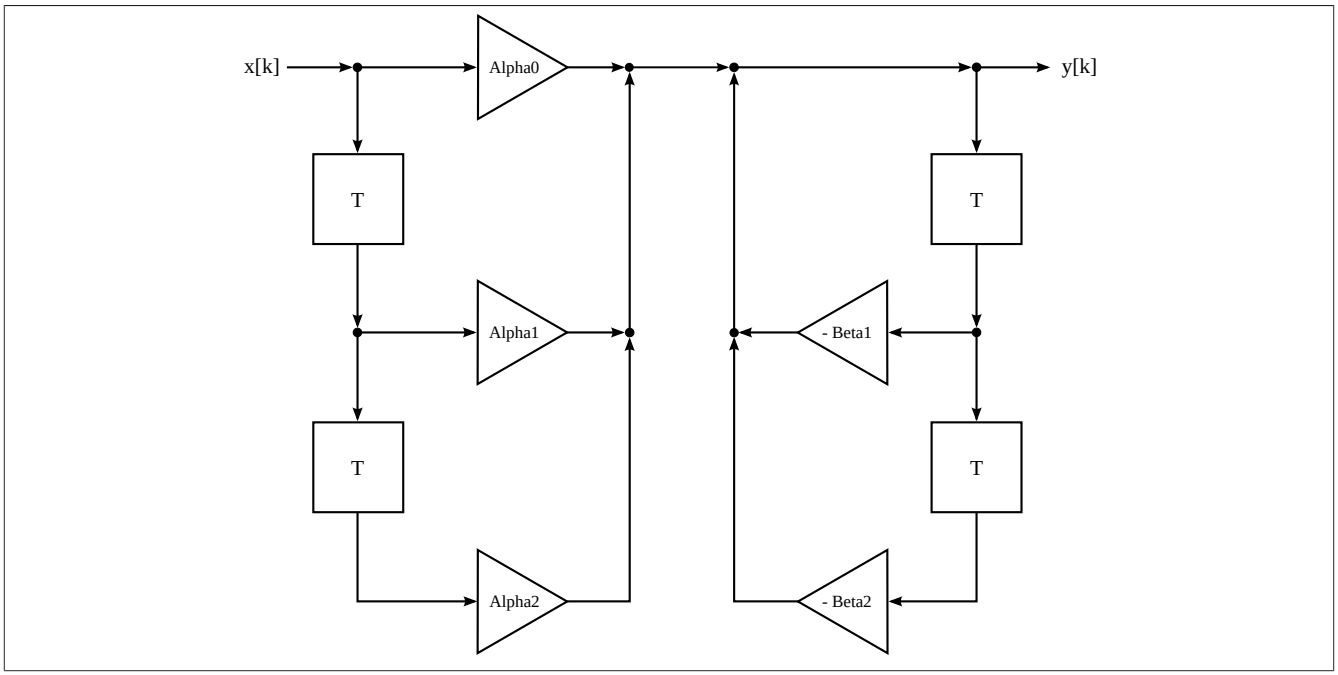
- 1st-order low pass
- 2nd-order low pass
- 2nd-order IIR

The cutoff frequency for the first-order and second-order low-pass filters is configurable. Coefficients Alpha0, Alpha1, Alpha2, Beta1 and Beta2 must be configured for the IIR filter.

Image as a z-transfer function

The second-order z-transfer function is specified in coefficient form (denominator polynomial Beta1, Beta2 and numerator polynomial Alpha0, Alpha1, Alpha2). The transfer method is calculated with the sampling cycle time.

$$S(Z) = \frac{a(Z)}{b(Z)} = \frac{Alpha0 + Alpha1 * Z^{-1} + Alpha2 * Z^{-2}}{1 + Beta1 * Z^{-1} + Beta2 * Z^{-2}}$$



Information:

The registers are described in "Filtering" on page 27.

3.3.3 Scaling

The analog input channels are calibrated and scaled when delivered (gain = k, offset = d).

User-defined scaling (gain = ku, offset = du) is also available. The calculation is optimized by combining the factors.

Normalization calculation:

nom = k * RawValue + d

k = k * ku

d = k * d + du

The values calculated here are limited to 16 bits.

Information:

The registers are described in "Scaling" on page 27.

3.4 Logical values

The physical values can be further processed using mathematical functions and comparators. Another logical channel can also be used as a starting point for further processing for a logical function.

Information:

The registers are described in "Logical configuration" on page 29.

3.4.1 Logical sampling

The module has a data buffer with 16 entries for each of the 6 logical channels. This buffer is processed according to the configured sampling cycle time. In addition, it's also possible to adjust the logical execution cycle using a prescaler for the sampling cycle time.

A maximum of only 30 bytes is available for cyclic transfer on the X2X bus, however. Minus the status and sample counter, this allows only a selection of 14 samples (with a 16-bit data width) from the physical and logical buffer to be transferred. For the logical channels, it is also possible to configure a 32-bit data width. Data loss can therefore occur with an imprecise selection and configuration.

Example

Displaying continuous sample lines.

- Sampling cycle time = 100 µs
- X2X cycle time = 500 µs

Sample line 1	LogicCh0xSample1
Sample line 2	LogicCh0xSample2
Sample line 3	LogicCh0xSample3
Sample line 4	LogicCh0xSample4
Sample line 5	LogicCh0xSample5
Sample line 6	LogicCh0xSample6
Difference SampleCount = 1	New value in sample line 1
Difference SampleCount = 2	New values in sample line 1 and sample line 2
...	
Difference SampleCount = 5	New values in sample line 1 to sample line 5

Information:

It is important to note that the sample counter refers to the update of the sample lines in the data buffer and not to the number of values transferred cyclically.

Display every second sample line to bridge a longer recording duration:

- Sampling cycle time = 100 µs
- X2X cycle time = 1000 µs

Sample line 1	LogicCh0xSample1
Sample line 3	LogicCh0xSample3
Sample line 5	LogicCh0xSample5
Sample line 7	LogicCh0xSample7
Sample line 9	LogicCh0xSample9
Sample line 11	LogicCh0xSample11
Difference SampleCount = 1	New value in sample line 1
Difference SampleCount = 3	New values in sample line 1 and sample line 3
...	
Difference SampleCount = 5	New values in sample line 1 to sample line 5
...	
Difference SampleCount = 9	New values in sample line 1 to sample line 9

3.4.2 Operation in the standard function model

6 logical channels are available on the module. Each channel can be configured with one of the following functions:

- "Addition of two channels with scaling" on page 17
- "Integral addition of two channels with scaling" on page 18
- "Multiplication of two channels with scaling" on page 19
- "Integral multiplication of two channels with scaling" on page 20
- "Comparator function of two channels" on page 20
- "Hysteresis comparator of one channel " on page 20

The sources to be used for each logical channel are selected using register "CfO_LogCh0NSource0x". The additionally required function parameters are configured in the "CfO_LogCh0NFuncPar0x" registers. "N" stands for the logical channel to be used; "x" stands for the source or function 0 or 1.

The following links can be made:

- Addition: Result = (Source0 * FunctionParameter0) + (Source1 * FunctionParameter1)
- Integral of the addition: Result = Σ(Source0 * FunctionParameter0) + (Source1 * FunctionParameter1)
- Multiplication: Result = Source0 * Source1 * FunctionParameter0
- Integral of the multiplication: Result = Σ(Source0 * Source1 * FunctionParameter0)
- Channel comparator: Result = Comparison of Source0 with Source1
- Hysteresis comparator: Result = Comparison of Source0 with (Lower threshold value = FunctionParameter0) and (Upper threshold value = FunctionParameter1)
- Physical value display: Result = (Source0 * 1) + (Source1 * 0)

With logical oversampling, 32-bit data points are available in addition to the 16-bit data points due to the possible calculation results. Which ones are used can be selected via the Automation Studio I/O configuration or the data point mapping.

If there is no need for 32-bit data points or this would result in a large restriction in the number of data points, scaling can be used to limit the number range to 16 bits.

The buffer depth for the digital comparator is also able to handle 16 results. Since these are Boolean results, these 16 bits are compressed into 2-byte data points and transferred that way.

3.4.2.1 Addition

This function can be used to determine the sum or difference of two channels. Only negative scaling of a channel must be configured for calculating the difference.

Calculation

Sample line = (Channel 1 * Scaling 1) + (Channel 2 * Scaling 2)

The addition calculation is performed internally with 32 bits in 16.16 format; the data from the source channels is evaluated as integers (applied to the high word), with decimal places possible due to scaling. When displayed as a logical 32-bit result, these decimal places are visible. When displayed as a 16-bit value, only the integral high word is used.

Example

Channel 1 = 2000
Channel 2 = 1000
Both scalings = 1

Results

3000.x = (2000.x * 1.0) + (1000.x * 1.0)
32-bit representation = 196608000 = 0xBB80000
16-bit representation = 3000 = 0xBB8

Information:

The maximum value for channels 1 and 2 can only be 32767; otherwise, an additional overflow occurs. If values greater than 32767 are possible, the range of values must be limited with scaling.

3.4.2.2 Integral of addition

This function can be used in the application to establish the average value of the channels or to calculate the deviation/difference between two channels over n samples. In each cycle, addition of the channels is carried out first; the summed result is then saved with the previous value in the current sample line. Depending on the result data type used (16-bit or 32-bit), the calculation overflows sooner or later after n samples due to continuous integration. Due to the signed result value, it must be ensured by the application that number n of samples is chosen small enough so that the integral calculation is less than half the range of values. If this is done, determining the average value can be carried out despite an overflow.

Calculation

Sample line result = Integral ((Channel 1 * Scaling 1) + (Channel 2 * Scaling 2))

The addition calculation is performed internally with 32 bits in 16.16 format; the data from the source channels is evaluated as integers (applied to the high word), with decimal places possible due to scaling. When displayed as a logical 32-bit result, these decimal places are visible. When displayed as a 16-bit value, only the integral high word is used.

Example

Channel 1 = 2000
Channel 2 = 1000
Both scalings = 1

Results

3000.x = (2000.x * 1.0) + (1000.x * 1.0)
32-bit representation = 196608000 = 0xBB80000.
16-bit representation = 3000 = 0xBB8

The average value can now be calculated as follows:

n = Number of samples / sample line
$Value_x$ = Value from sample line x → Newer value
$Value_{(x-n)}$ = Value from sample line x-n → Older value, n samplings back

Average value = ($Value_x$ - $Value_{(x-n)}$) / n

Information:

The maximum value for channels 1 and 2 can only be 32767; otherwise, an additional overflow occurs. If values greater than 32767 are possible, the range of values must be limited with scaling.

3.4.2.3 Multiplication

This function can be used to calculate the current effective power P = U * I.

Calculation

Sample line = Channel 1 * Channel 2 * Scaling

Multiplication is calculated internally as a 32-bit value; the 16-bit data from the source channels is passed to the low word. When displayed as a logical 32-bit value, the entire result is visible (no multiplication overflow possible when scaling ≤1). When displayed as a 16-bit value, only the high word is used. Though there is a loss of precision, the 16-bit values allow more data points to be transferred.

Example

Channel 1 = 2000
Channel 2 = 1000
Scaling = 1

Results

2000000 = (2000 * 1000 * 1.0)
32-bit representation = 2000000 = 0x1E8480
16-bit representation = 30 = 0x1E

Information:

If higher accuracy is required for the 16-bit value, scaling can be used to shift the bit value in steps of 2^n (... *128, * 256, ...). It is important to ensure that the input values of the source channels are limited here as well; otherwise, there will be an overflow in the multiplication.

3.4.2.4 Integral of multiplication

This function can be used to calculate the average of the active power in the application. In each cycle, multiplication of the channels is carried out first; the summed result is then saved with the previous value in the current sample line. Depending on the result data type used (16-bit or 32-bit), the calculation overflows sooner or later after n samples due to continuous integration. Due to the signed result value, it must be ensured by the application that number n of samples is chosen small enough so that the integral calculation is less than half the range of values. If this is done, determining the average value can be carried out despite an overflow.

Calculation

Sample line = Integral (Channel 1 * Channel 2 * Scaling)

Multiplication is calculated internally as a 32-bit value; the 16-bit data from the source channels is passed to the low word. When displayed as a logical 32-bit value, the entire result is visible (no multiplication overflow possible when scaling ≤1). When displayed as a 16-bit value, only the high word is used. Though there is a loss of precision, the 16-bit values allow more data points to be transferred.

Example

Channel 1 = 2000
Channel 2 = 1000
Scaling = 1

Results

2000000 = (2000 * 1000 * 1.0)
32-bit representation = 2000000 = 0x1E8480
16-bit representation = 30 = 0x1E

The average value can now be calculated as follows:

n = Number of samples / sample line
$Value_x$ = Value from sample line x ⟶ Newer value
$Value_{(x-n)}$ = Value from sample line x-n ⟶ Older value, n samplings back

Average value = ($Value_x$ - $Value_{(x-n)}$) / n

Information:

If higher accuracy is required for the 16-bit value, scaling can be used to shift the bit value in steps of 2^n (... *128, * 256, ...). It is important to ensure that the input values of the source channels are limited here as well; otherwise, there will be an overflow in the multiplication.

3.4.2.5 Channel comparator

This function can be used to compare channel values. The following applies:

- Channel 1 > Channel 2 = 1
- Channel 1 < Channel 2 = 0
- Channel 1 = Channel 2 = State before values are the same

Calculation

Sample line (bit) = Comparison (channel value 1 with channel value 2)

3.4.2.6 Hysteresis comparator

This function can be used to monitor channel limit violations. The following applies:

- Channel > Upper threshold value = 1
- Channel < Lower threshold value = 0
- Channel within threshold = Value before occurrence

Calculation

Sample line (bit) = Comparison (channel value with lower threshold value) and (channel value with upper threshold value))

3.4.3 Operation in the bus controller function model

When used on the bus controller, there are 4 logical functions available for each of the analog input channels in addition to the physical value output. Each channel can be configured with one of the following functions:

- "Output of physical values" on page 21 (default setting)
- "Addition of two channels with scaling" on page 17
- "Integral addition of two channels with scaling" on page 18
- "Multiplication of two channels with scaling" on page 19
- "Integral multiplication of two channels with scaling" on page 20
- "Comparator function of two channels" on page 20
- "Hysteresis comparator of one channel " on page 20

In contrast to the standard function model, oversampling and the two digital comparators are not supported. As a result, there is only one newly generated value per channel in each update cycle. Another difference is that there are only 4 logical calculation channels instead of 6.

The logical functions addition, integral of addition, multiplication and integral of multiplication do not differ from the standard function model in their configuration and function when operating on the bus controller.

3.4.3.1 Value display in bus controller operating mode

The physical value display in the bus controller function model is initialized automatically and represents a special form of the logical function "Addition" with defined scaling factors.

Calculation

Result = Channel value
Formula used for addition: Result = (Channel value 1 * 1) + (Channel value 2 * 0)

Information:

In this function model, only the 2 physical input channels are available, and the scaling factors have fixed values.

3.5 Monitoring the inputs

The module's inputs are monitored. The standard and extended error messages must be enabled individually for each channel.

The following areas are monitored:

- **Out of range:** The analog input signal is outside the specified operating range.
- **Filter error:** The set filter theorem cannot be calculated (parameter error).
- **Underflow:** The input signal is less than the lower limit value.
- **Overflow:** The input signal is greater than the upper limit value.

Value	Information
0	No error occurred
1	Error occurred

The composite errors of the channels are derived from the individual extended error states, e.g. underflow, overflow of the input range on the analog value. The error state of the oversampling results from a cycle time violation of the set sampling cycle time. All configured physical and logical oversampling functions must be able to be carried out in the configured sampling cycle time; otherwise, an error message will be displayed.

Any error messages that occur must be acknowledged by setting the corresponding bits.

Information:

The registers are described in "Error monitoring" on page 33.

4 Commissioning

4.1 Using the module on the bus controller

Function model 254 "Bus controller" is used by default only by non-configurable bus controllers. All other bus controllers can use other registers and functions depending on the fieldbus used.

For detailed information, see section "Additional information - Using I/O modules on the bus controller" in the X20 user's manual (version 3.50 or later).

4.1.1 CAN I/O bus controller

The module occupies 1 analog logical slot on CAN I/O.

5 Register description

5.1 General data points

In addition to the registers described in the register description, the module has additional general data points. These are not module-specific but contain general information such as serial number and hardware variant.

General data points are described in section "Additional information - General data points" in the X20 System user's manual.

5.2 Function model 0 - Standard

Register	Name	Data type	Read		Write	
			Cyclic	Acyclic	Cyclic	Acyclic
System configuration						
513	CfO_BaseConfig	USINT				•
15364	CfO_CycleTime	UDINT				•
15370	CfO_SyncOffset	UINT				•
15374	CfO_Prescaler	UINT				•
Error messages - Configuration						
385	CfO_ErrorID0007	USINT				•
389	CfO_ErrorID1017	USINT				•
Physical channel configuration						
8194 8450	CfO_ModeCh01 CfO_ModeCh02	UINT				•
8204 8460	CfO_UserGainCh01 CfO_UserGainCh02	DINT				•
8212 8468	CfO_UserOffsetCh01 CfO_UserOffsetCh02	DINT				•
8220 8476	CfO_Alpha0Ch01 CfO_Alpha0Ch02	DINT				•
8228 8484	CfO_Alpha1Ch01 CfO_Alpha1Ch02	DINT				•
8236 8492	CfO_Alpha2Ch01 CfO_Alpha2Ch02	DINT				•
8244 8500	CfO_Beta1Ch01 CfO_Beta1Ch02	DINT				•
8252 8508	CfO_Beta2Ch01 CfO_Beta2Ch02	DINT				•
8198 8454	CfO_CutOffFrequCh01 CfO_CutOffFrequCh02	UINT				•
Logical channel configuration						
10242 + (N-1) * 256	CfO_LogCh0NMode (index N = 1 to 6)	UINT				•
10245 + (N-1) * 256	CfO_LogCh0NSource00 (index N = 1 to 6)	USINT				•
10247 + (N-1) * 256	CfO_LogCh0NSource01 (index N = 1 to 6)	USINT				•
10260 + (N-1) * 256	CfO_LogCh0NFuncPar00 (index N = 1 to 6)	UDINT				•
10268 + (N-1) * 256	CfO_LogCh0NFuncPar01 (index N = 1 to 6)	UDINT				•
Analog inputs - Communication						
5062 5070	AnalogInput01 AnalogInput02	INT	•			
Error messages - Communication						
261	Composite error	USINT	•			
	Channel01Error	Bit 0				
	Channel02Error	Bit 1				
	PhysicalError	Bit 4				
	LogicalError	Bit 5				
325	Acknowledging standard errors	USINT			•	
	AckChannel01Error	Bit 0				
	AckChannel01Error	Bit 1				
	AckPhysicalError	Bit 4				
	AckLogicalError	Bit 5				

Register description

Register	Name	Data type	Read		Write	
			Cyclic	Acyclic	Cyclic	Acyclic
257	Extended channel error messages	USINT	•			
	Channel01OutOfRange	Bit 0				
	Channel01FilterError	Bit 1				
	Channel01Underflow	Bit 2				
	Channel01Overflow	Bit 3				
	Channel02OutOfRange	Bit 4				
	Channel02FilterError	Bit 5				
	Channel02Underflow	Bit 6				
	Channel02Overflow	Bit 7				
321	Acknowledging extended channel error messages	USINT			•	
	AckChannel01OutOfRange	Bit 0				
	AckChannel01FilterError	Bit 1				
	AckChannel01Underflow	Bit 2				
	AckChannel01Overflow	Bit 3				
	AckChannel02OutOfRange	Bit 4				
	AckChannel02FilterError	Bit 5				
	AckChannel02Underflow	Bit 6				
	AckChannel02Overflow	Bit 7				
Physical analog sample display						
4102 + (16-N) * 64	PhysCh01SampleN (index N = 1 to 16)	INT	•			
4110 + (16-N) * 64	PhysCh02SampleN (index N = 1 to 16)	INT	•			
5106	PhysTimestamp	INT	•			
5108	PhysTimestamp	DINT	•			
5113	PhysSampleCount	SINT	•			
5114	PhysSampleCount	INT	•			
Logical analog and digital sample display						
6148 + (16-N) * 64	LogicCh01SampleN (index N = 1 to 16) (32-bit)	DINT	•			
6150 + (16-N) * 64	LogicCh01SampleN (index N = 1 to 16) (16-bit)	INT	•			
6156 + (16-N) * 64	LogicCh02SampleN (index N = 1 to 16) (32-bit)	DINT	•			
6158 + (16-N) * 64	LogicCh02SampleN (index N = 1 to 16) (16-bit)	INT	•			
6164 + (16-N) * 64	LogicCh03SampleN (index N = 1 to 16) (32-bit)	DINT	•			
6166 + (16-N) * 64	LogicCh03SampleN (index N = 1 to 16) (16-bit)	INT	•			
6172 + (16-N) * 64	LogicCh04SampleN (index N = 1 to 16) (32-bit)	DINT	•			
6174 + (16-N) * 64	LogicCh04SampleN (index N = 1 to 16) (16-bit)	INT	•			
6180 + (16-N) * 64	LogicCh05SampleN (index N = 1 to 16) (32-bit)	DINT	•			
6182 + (N-16) * 64	LogicCh05SampleN (index N = 1 to 16) (16-bit)	INT	•			
6188 + (16-N) * 64	LogicCh06SampleN (index N = 1 to 16) (32-bit)	DINT	•			
6190 + (16-N) * 64	LogicCh06SampleN (index N = 1 to 16) (16-bit)	INT	•			
7109 + (N-1) * 8	LogicCh0NSample16_9 (index N = 1 to 5)	USINT	•			
7151	LogicCh06Sample16_9	USINT	•			
7111 + (N-1) * 8	LogicCh0NSample8_1 (index N = 1 to 5)	USINT	•			
7149	LogicCh06Sample8_1	USINT	•			
7154	LogicTimestamp	INT	•			
7156	LogicTimestamp	DINT	•			
7161	LogicSampleCount	SINT	•			
7162	LogicSampleCount	INT	•			

5.3 Function model 254 - Bus controller

Function model "Bus controller" has limitations compared to function model Standard. For details, see "Operating modes " on page 8.

Register	Offset[1)]	Name	Data type	Read		Write	
				Cyclic	Acyclic	Cyclic	Acyclic
System configuration							
513	-	CfO_BaseConfig	USINT				•
15364	-	CfO_CycleTime	UDINT				•
15370	-	CfO_SyncOffset	UINT				•
15374	-	CfO_Prescaler	UINT				•
Error messages - Configuration							
385	-	CfO_ErrorID0007	USINT				•
389	-	CfO_ErrorID1017	USINT				•
Physical channel configuration							
8194 8450	-	CfO_ModeCh01 CfO_ModeCh02	UINT				•
8204 8460	-	CfO_UserGainCh01 CfO_UserGainCh02	DINT				•
8212 8468	-	CfO_UserOffsetCh01 CfO_UserOffsetCh02	DINT				•
8220 8476	-	CfO_Alpha0Ch01 CfO_Alpha0Ch02	DINT				•
8228 8484	-	CfO_Alpha1Ch01 CfO_Alpha1Ch02	DINT				•
8236 8492	-	CfO_Alpha2Ch01 CfO_Alpha2Ch02	DINT				•
8244 8500	-	CfO_Beta1Ch01 CfO_Beta1Ch02	DINT				•
8252 8508	-	CfO_Beta2Ch01 CfO_Beta2Ch02	DINT				•
8198 8454	-	CfO_CutOffFrequCh01 CfO_CutOffFrequCh02	UINT				•
Logical channel configuration							
10242 + (N-1) * 256	-	CfO_LogCh0NMode (index N = 1 to 6)	UINT				•
10245 + (N-1) * 256	-	CfO_LogCh0NSource00 (index N = 1 to 6)	USINT				•
10247 + (N-1) * 256	-	CfO_LogCh0NSource01 (index N = 1 to 6)	USINT				•
10260 + (N-1) * 256	-	CfO_LogCh0NFuncPar00 (index N = 1 to 6)	UDINT				•
10268 + (N-1) * 256	-	CfO_LogCh0NFuncPar01 (index N = 1 to 6)	UDINT				•
Analog inputs - Communication							
5062 5070	0 2	AnalogInput01 AnalogInput02	INT	•			
Error messages - Communication							
261	-	Composite error	USINT		•		
		Channel01Error	Bit 0				
		Channel02Error	Bit 1				
		PhysicalError	Bit 4				
		LogicalError	Bit 5				
325	-	Acknowledging standard errors	USINT				•
		AckChannel01Error	Bit 0				
		AckChannel01Error	Bit 1				
		AckPhysicalError	Bit 4				
		AckLogicalError	Bit 5				
257	-	Extended channel error messages	USINT		•		
		Channel01OutOfRange	Bit 0				
		Channel01FilterError	Bit 1				
		Channel01Underflow	Bit 2				
		Channel01Overflow	Bit 3				
		Channel02OutOfRange	Bit 4				
		Channel02FilterError	Bit 5				
		Channel02Underflow	Bit 6				
		Channel02Overflow	Bit 7				

Register description

Register	Offset[1)]	Name	Data type	Read		Write	
				Cyclic	Acyclic	Cyclic	Acyclic
321	-	Acknowledging extended channel error messages	USINT				•
		AckChannel01OutOfRange	Bit 0				
		AckChannel01FilterError	Bit 1				
		AckChannel01Underflow	Bit 2				
		AckChannel01Overflow	Bit 3				
		AckChannel02OutOfRange	Bit 4				
		AckChannel02FilterError	Bit 5				
		AckChannel02Underflow	Bit 6				
		AckChannel02Overflow	Bit 7				

1) The offset specifies the position of the register within the CAN object.

5.4 Configuration

Configuration must take place in addition to using suitable terminals.

5.4.1 System configuration

The following registers are used to configure the module's system settings.

5.4.1.1 Logical oversampling and data acquisition

Name:
CfO_BaseConfig

This register can be used to configure settings relating to handling logical oversampling and data acquisition. For details, see "Priorities and values" on page 11.

Data type	Values	Bus controller default setting
USINT	See the bit structure.	49

Bit structure:

Bit	Description	Value	Information
0	"Display configuration for logical values active/inactive" in the Automation Studio I/O configuration	0	Inactive
		1	Active (bus controller default setting)
1	"Logical handling priority" in the Automation Studio I/O configuration	0	Low (bus controller default setting)
		1	High
2 - 3	Reserved	-	
4	"Physical input mode" in the Automation Studio I/O configuration	0	Newest value
		1	Referenced value (reference = prescaled system timer) (bus controller default setting)
5	"Logical input mode" in the Automation Studio I/O configuration	0	Newest value
		1	Referenced value (reference = prescaled system timer) (bus controller default setting)
6 - 7	Reserved	-	

5.4.1.2 Sampling cycle time

Name:
CfO_CycleTime
"Physical sample time" in the Automation Studio I/O configuration.

This register is used to set the sampling cycle time on the module. The format is a 16.16-bit unsigned 4-byte value, where the high word is the microseconds integer and the low word is the decimal places. The decimal places allow a more precise adjustment to the X2X cycle time. The absolute resolution is 1 µs.

Input value = Time in µs * 65536 data type

Data type	Value	Information
UDINT	2,621,440 to 2,147,483,647	40 µs to 32 ms sampling cycle time. Bus controller default setting: 6,553,600 = 100 µs

5.4.1.3 Prescaler of the logical channel processing time

Name:
CfO_Prescaler

This register contains the prescaler for configuring the logical channel processing time. The actual logical cycle time will be calculated from the multiple of the sampling cycle time that is defined here. If a very short sampling cycle time is required for physical samples, then the module load can be reduced using the second time base for the logical samples.

Data type	Value	Information
UINT	1 to 10	Multiples of the physical sampling cycle for logical processing Bus controller default setting: 2

5.4.1.4 Synchronization offset

Name:
CfO_SyncOffset
"Synchronization offset" in the Automation Studio I/O configuration.

The system cycle can be offset in 1 µs steps in this register.

Data type	Value	Information
UINT	-32,768 to 32,767	Synchronization offset in µs. Bus controller default setting: 0

5.4.2 Scaling

The analog input channels are calibrated and scaled when delivered. User-defined scaling is also available.

5.4.2.1 Gain

Name:
CfO_UserGainCh01 to CfO_UserGainCh02
"Configuration channel 0x / gain" in the Automation Studio I/O configuration

These registers are used to set the gain for the corresponding channel. The format is a 16.16-bit signed 4-byte value, where the high word is the integer and the low word is the decimal places.

Input value = Gain ku * 65536

Value 65535 corresponds to a gain of 1.

Data type	Value	Information
DINT	-2,147,483,648 to 2,147,483,647	Gain. Bus controller default setting: 65535

5.4.2.2 Offset

Name:
CfO_UserOffsetCh01 to CfO_UserOffsetCh02
"Configuration channel 0x / offset" in the Automation Studio I/O configuration

These registers are used to set the offset for the corresponding channel. The format is a 16.16-bit signed 4-byte value, where the high word is the integer and the low word is the decimal places.

Input value = Offset du * 65536

Value 65536 corresponds to an offset of 1.

Data type	Value	Information
DINT	-2,147,483,648 to 2,147,483,647	Offset. Bus controller default setting: 0

5.4.3 Filtering

The module is equipped with an individually configurable input filter for each individual channel.

5.4.3.1 Coefficients

Name:
CfO_Alpha0Ch01 to CfO_Alpha0Ch02
CfO_Alpha1Ch01 to CfO_Alpha1Ch02
CfO_Alpha2Ch01 to CfO_Alpha2Ch02
CfO_Beta1Ch01 to CfO_Beta1Ch02
CfO_Beta1Ch01 to CfO_Beta1Ch02

These registers are used to set the coefficients for the IIR filter.

Data type	Values	
DINT	-2,147,483,648 to 2,147,483,647	IIR filter coefficient. Bus controller default setting: 0

5.4.3.2 Cutoff frequency

Name:
CfO_CutOffFrequCh01 to CfO_CutOffFrequCh02

These registers are used to configure the limit frequency in hertz for a 1st- or 2nd-order low pass for the corresponding channel.

Data type	Value	Information
UINT	0 to 65535	Cutoff frequency for 1st- or 2nd-order low pass [Hz]. Bus controller default setting: 1000

5.4.4 Physical configuration

The conversion results are scaled and filtered before being transferred to the higher-level system. To do this, the operating mode must be set for each channel.

5.4.4.1 Operating mode

Name:
CfO_ModeCh01 to CfO_ModeCh02

The operating mode for each physical channel can be configured in this register.

Data type	Values	Bus controller default setting
UINT	See the bit structure.	256

Bit structure:

Bit	Description	Value	Information
0 - 2	Connection configuration This value must be set the same for each register!	000	Voltage signal (bus controller default setting)
		111	Current signal
3 - 7	Reserved	0	
8 - 10	Operating mode	000	Channel disabled
		001	No filtering (bus controller default setting)
		010	2nd-order IIR (configurableAlpha and Beta coefficients)
		011	1st-order low pass (configurable limit frequency)
		100	2nd-order low pass (configurable limit frequency)
		101 to 111	Reserved
11 - 15	Reserved	0	

5.4.5 Logical configuration

The physical values can be further processed using mathematical functions and comparators. Various settings must be made for this.

5.4.5.1 Operating mode

Name:
CfO_LogCh01Mode to CfO_LogCh06Mode
"Logical configuration channel 0x / Addition" in the Automation Studio I/O configuration.
"Logical configuration channel 0x / Integral of addition" in the Automation Studio I/O configuration.
"Logical configuration channel 0x / Multiplication" in the Automation Studio I/O configuration.
"Logical configuration channel 0x / Integral of multiplication" in the Automation Studio I/O configuration.
"Logical configuration channel 0x / Channel comparator" in the Automation Studio I/O configuration.
"Logical configuration channel 0x / Hysteresis comparator" in the Automation Studio I/O configuration.
"Logical configuration channel 0x / Physical value display" in the Automation Studio I/O configuration.

The operating mode for each logical channel can be configured in this register.

The sources to be used for each logical channel are selected using the "CfO_LogCh0NSource0x" on page 30 registers. The additionally required function parameters are configured in the "CfO_LogCh0NFuncPar0x" on page 30 registers. "N" stands for the logical channel to be used; "x" stands for the source or function 0 or 1.

Data type	Value	Information
UINT	0	Channel switched off. Bus controller default setting: Channel 3 to 6
	256	Addition or physical value display[1]. Bus controller default setting: Channel 1 to 2
	257	Integral of addition
	512	Multiplication
	513	Integral of multiplication
	768	Channel comparator
	1024	Hysteresis comparator

1) Only registers CfO_LogCh01Mode to CfO_LogCh02Mode are used for physical value display.

5.4.5.2 Source registers

Name:
CfO_LogCh01Source00 to CfO_LogCh06Source00
CfO_LogCh01Source01 to CfO_LogCh06Source01

These registers can be used to select the source registers for the operating mode of the logical channel configured in the register "CfO_LogCh0NMode" on page 29.
In the name, "Source00" stands for source register 0; "Source01" stands for source register 1.

In Value display in bus controller operating mode mode, the same channel number is written to both source registers.

Data type	Value	Information
USINT	0	Physical channel 01. Bus controller default setting[1)]
	1	Physical channel 02. Bus controller default setting[1)]
	8	Logical channel 01[1)]
	...	...
	13	Logical channel 06

1) **Values**

Channel 1: 0
Channel 2: 1
Channels 3 to 6: 0

2) Logical channels cannot be used in the bus controller function model.

5.4.5.3 Additional function parameters

Name:
CfO_LogCh01FuncPar00 to CfO_LogCh06FuncPar00
CfO_LogCh01FuncPar01 to CfO_LogCh06FuncPar01

These registers can be used to store additional function parameters for the operating modes of the logical channel set in register "CfO_LogCh0NMode" on page 29.

The meaning of the function parameter varies depending on the operating mode.

Operating mode	Parameter 1	Parameter 2
(Integral of) addition	Scaling factor	Scaling factor
(Integral of) multiplication	Scaling factor	-
Channel comparator	-	-
Hysteresis comparator	Upper threshold value	Lower threshold value
Output of physical values	Fixed scaling factor = 65536	Defined scaling factor = 0

Value 65536 corresponds to scaling or a threshold value of 1.

Data type	Value	Information
UDINT	0 to 4,294,967,295	Scaling factor or threshold value. Bus controller default setting: Register "...FuncPar00" Channels 1 to 4 65536 Channels 5 to 6 0 Register "...FuncPar01" All 0

5.5 Communication - General

The analog inputs of the module convert the current or voltage values with a resolution of 16 bits.

5.5.1 Analog inputs

Name:
AnalogInput01 to AnalogInput02

This module can be configured and operated as a normal analog input module without logical auxiliary functions. The physical values from the last sampling cycle are used as input values in this case.

Analog input values are displayed as signed 16-bit values depending on the configured operating mode.

Data type	Value	Information
INT	-32,768 to 32,767	Voltage signal ±10 VDC
	0 to 32,767	Current signal 0 to 20 mA

Information:

It is important to note that the oversampling function is not available in function model "Bus controller" due to the amount of data and lack of consistency!

5.5.2 Physical sampling

The module has a data buffer with 16 entries for each of the physical input channels. This buffer is processed with the set sampling cycle time.

5.5.2.1 Physical data buffer

Name:
PhysCh01Sample1 to PhysCh01Sample16
PhysCh02Sample1 to PhysCh02Sample16

These registers are the physical buffer registers of the analog channels. 16 registers are available for each channel. Sample 1 is the newest value; sample 16 is the oldest.

Data type	Value	Information
INT	-32,768 to 32,767	Voltage signal ±10 VDC
	0 to 32,767	Current signal 0 to 20 mA

5.5.2.2 Physical sample counter

Name:
PhysSampleCount

This register is an integer counter that is increased as soon as the module has saved a new physical sample line. The number of new sample lines is calculated from the difference to the previous cycle.

Data type	Value
SINT	-128 to 127
INT	-32,768 to 32,767

5.5.2.3 Physical timestamp

Name:
PhysTimestamp

This register returns the timestamp of the values currently being determined as signed values in µs. This data point is the timestamp of the physical sample line 1.

Data type	Values
INT	-32768 to 32767
DINT	-2147483648 to 2147483647

5.5.3 Logical sampling

The physical values can be further processed using mathematical functions and comparators. The module has a data buffer with 16 entries for each of the 6 logical channels.

5.5.3.1 Logical data buffer

Name:
LogicCh01Sample1 to LogicCh01Sample16
...
LogicCh06Sample1 to LogicCh06Sample16

These registers are the buffer registers of the logical input channels. 16 registers are available for each channel. Sample 1 is the newest value; sample 16 is the oldest.

The calculated values are displayed as signed 16-bit or 32-bit values depending on the register used.

Data type	Values
INT	-32768 to 32767
DINT	-2147483648 to 2147483647

5.5.3.2 Results 9-16 of the comparator comparison

Name:
LogicCh01Sample16_9 to LogicCh06Sample16_9

The results of samples 9 to 16 of the logical digital comparator of the logical channels are contained in these registers. Each of these bits corresponds to a sample line, with sample 9 the newest and sample 16 the oldest comparator comparison. The results of samples 1 to 8 are contained in register "LogicCHSample8_1" on page 32.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0	Comparator result	x	Sample 9
...	...		
7	Comparator result	x	Sample 16

5.5.3.3 Results 1-8 of the comparator comparison

Name:
LogicCh01Sample8_1 to LogicCh06Sample8_1

These registers are used to represent the results of samples 1 to 8 of the logical digital comparator for the logical channels. Each of these bits corresponds to a sample line, with sample 1 the newest and Sample 8 the oldest comparator comparison. The results of samples 9 to 16 are represented in register "LogicSample16_9" on page 32.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0	Comparator result	x	Sample 1
...	...		
7	Comparator result	x	Sample 8

5.5.3.4 Logical sample counter

Name:
LogicSampleCount

This register is an integer counter that is increased as soon as the module has saved a new logical sample line. The number of new sample lines is calculated from the difference to the previous cycle.

Data type	Value
SINT	-128 to 127
INT	-32,768 to 32,767

5.5.3.5 Logical timestamp

Name:
LogicTimestamp

This register provides the timestamp of the currently determined values as a signed 2-byte or 4-byte value in microseconds. This data point is the timestamp of logical sample line 1.

Data type	Values
INT	-32768 to 32767
DINT	-2147483648 to 2147483647

5.6 Error monitoring

The registers for displaying and acknowledging errors are transferred either cyclically or acyclically depending on the function model.

5.6.1 Enabling standard error messages

Name:
CfO_ErrorID1017
Automatic enabling by the Automation Studio I/O configuration.

This register can be used to enable the standard error messages. The composite errors of the channels are derived from the individual extended error states, e.g. underflow, overflow of the input range on the analog value. The error status of the oversampling results from a cycle time violation of the set sampling cycle time.

Data type	Values	Bus controller default setting
USINT	See the bit structure.	63

Bit structure:

Bit	Description	Value	Information
0	Composite errors on channel 01	0	Error generation disabled
		1	Error generation enabled (bus controller default setting)
1	Composite errors on channel 02	0	Error generation disabled
		1	Error generation enabled (bus controller default setting)
2 - 3	Reserved	0	
4	Physical sample error status	0	Error generation disabled
		1	Error generation enabled (bus controller default setting)
5	Logical sample error status	0	Error generation disabled
		1	Error generation enabled (bus controller default setting)
6 - 7	Reserved	0	

5.6.2 Enabling extended error messages

Name:
CfO_ErrorID0007
Automatic enabling in the Automation Studio I/O configuration by selecting "Extended error status information" and channel activation.

This register can be used to enable the extended error messages for analog channels 1 and 2.

Data type	Values	Bus controller default setting
USINT	See the bit structure.	0

Bit structure:

Bit	Description	Value	Information
0	Channel 1: Range exceeded violation (pos.)	0	Error generation disabled (bus controller default setting)
		1	Range exceeded violation (pos.) enabled
1	Channel 1: Filter error	0	Error generation disabled (bus controller default setting)
		1	Filter error enabled
2	Channel 1: Underflow	0	Error generation disabled (bus controller default setting)
		1	Underflow enabled
3	Channel 1: Overrun	0	Error generation disabled (bus controller default setting)
		1	Overflow enabled
4	Channel 2: Range exceeded violation (pos.)	0	Error generation disabled (bus controller default setting)
		1	Range exceeded violation (pos.) enabled
5	Channel 2: Filter error	0	Error generation disabled (bus controller default setting)
		1	Filter error enabled
6	Channel 2: Underflow	0	Error generation disabled (bus controller default setting)
		1	Underflow enabled
7	Channel 2: Overrun	0	Error generation disabled (bus controller default setting)
		1	Overflow enabled

5.6.3 Composite error

Name:
Channel01Error to Channel02Error
PhysicalError
LogicalError

Composite errors are mapped to this register.

All configured physical and logical oversampling functions must be able to be carried out in the configured sampling cycle time; otherwise, these error messages will be displayed. The system can be further adjusted with settings for the processing priority and the prescaler for logical oversampling.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0	Channel01Error	0	No error
		1	Composite errors on channel 1
1	Channel02Error	0	No error
		1	Composite errors on channel 2
2 - 3	Reserved	0	
4	PhysicalError	0	No error
		1	Physical sample error status, sampling cycle time too short
5	LogicalError	0	No error
		1	Logical sample error status, sampling cycle time too short or prescaler configured too low
6 - 7	Reserved	0	

5.6.4 Acknowledging standard errors

Name:
AckChannel01Error to AckChannel02Error
AckPhysicalError
AckLogicalError

In this register, error messages from register "Composite error" on page 34 can be acknowledged by setting the corresponding bits.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0	AckChannel01Error	0	No change
		1	Acknowledge error
1	AckChannel02Error	0	No change
		1	Acknowledge error
2 - 3	Reserved	0	
4	AckPhysicalError	0	No change
		1	Acknowledge error
5	AckLogicalError	0	No change
		1	Acknowledge error
6 - 7	Reserved	0	

5.6.5 Extended channel error messages

Name:
Channel01OutOfRange to Channel02OutOfRange
Channel01FilterError to Channel02FilterError
Channel01Underflow to Channel02Underflow
Channel01Overflow to Channel02Overflow

The error states of input channels 1 and 2 are represented in these registers.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0	Channel01OutOfRange	0	No error
		1	Range exceeded violation (pos.) occurred
1	Channel01FilterError	0	No error
		1	Filter error occurred
2	Channel01Underflow	0	No error
		1	Underflow occurred
3	Channel01Overflow	0	No error
		1	Overflow occurred
4	Channel02OutOfRange	0	No error
		1	Range exceeded violation (pos.) occurred
5	Channel02FilterError	0	No error
		1	Filter error occurred
6	Channel02Underflow	0	No error
		1	Underflow occurred
7	Channel02Overflow	0	No error
		1	Overflow occurred

5.6.6 Acknowledging extended channel error messages

Name:
AckChannel01OutOfRange to AckChannel02OutOfRange
AckChannel01FilterError to AckChannel02FilterError
AckChannel01Underflow to AckChannel02Underflow
AckChannel01Overflow to AckChannel02Overflow

These registers can be used to acknowledge the error messages from the "ExtendedChannelErrorMessages" on page 35 registers by setting the corresponding bit.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0	AckChannel01OutOfRange	0	No change
		1	Acknowledge error
1	AckChannel01FilterError	0	No change
		1	Acknowledge error
2	AckChannel01Underflow	0	No change
		1	Acknowledge error
3	AckChannel01Overflow	0	No change
		1	Acknowledge error
4	AckChannel02OutOfRange	0	No change
		1	Acknowledge error
5	AckChannel02FilterError	0	No change
		1	Acknowledge error
6	AckChannel02Underflow	0	No change
		1	Acknowledge error
7	AckChannel02Overflow	0	No change
		1	Acknowledge error

5.7 Minimum cycle time

The minimum cycle time specifies how far the bus cycle can be reduced without communication errors occurring. It is important to note that very fast cycles reduce the idle time available for handling monitoring, diagnostics and acyclic commands.

Minimum cycle time
200 µs

5.8 Minimum I/O update time

There is no limitation or dependency on the bus cycle time.

The I/O update time is defined using the "Sampling time" register. The fastest possible sampling time depends on the number of channels to be converted and the configuration.