X20(c)DS1119

1 General information

This module is a multifunction digital signal processor module. It can be used extremely flexibly for a wide variety of tasks involving digital signal processing or digital signal generation. Two primary example applications include controlling stepper output stages with pulse and direction signals or using as encoder emulation. In this application, for example, frequency inverters or servo axes with the speed follow function can follow a real or virtual master axis.

- 3 digital 5 V channels, configurable as inputs or outputs
- 2 digital 24 V input channels
- 1 universal counter pair (2 event counters, AB counters or up/down counters)
- Linear motion generator (A/B, direction/frequency) with one reference pulse
- SSI absolute encoder
- NetTime timestamp: Input data, target position, position change, edge change, counter change

NetTime timestamp

An additional major feature is the module's integrated timestamp function. It allows counter ramps curves to be generated virtually independently of bus cycle times during encoder emulation, for example. Only the target counter value and moment when it should be reached are transferred. The module automatically generates the corresponding counter values at the appropriate time, precisely in microsecond resolution and independently of the bus clock.

2 Coated modules

Coated modules are X20 modules with a protective coating for the electronics component. This coating protects X20c modules from condensation and corrosive gases.
The modules' electronics are fully compatible with the corresponding X20 modules.

For simplification purposes, only images and module IDs of uncoated modules are used in this data sheet.

The coating has been certified according to the following standards:

- Condensation: BMW GS 95011-4, 2x 1 cycle
- Corrosive gas: EN 60068-2-60, method 4, exposure 21 days

2.1 Starting temperature

The starting temperature describes the minimum permissible ambient temperature in a voltage-free state at the time the coated module is switched on. This is permitted to be as low as -40°C. During operation, the conditions as specified in the technical data continue to apply.

Information:

It is important to absolutely ensure that there is no forced cooling by air currents in the closed control cabinet, e.g. due to the use of a fan or ventilation slots.

3 Order data

Order number	Short description	Figure
	Digital signal processing and preparation	
X20DS1119	X20 multifunction digital signal processor, 3 digital channels 5 V (symmetrical) configurable as inputs or outputs, 2 digital input channels 24 V (asymmetrical), max. 2 event counters, 1 universal counter pair as AB counter or up/down counter, linear movement generator (A/B, direction/frequency) with 1 reference pulse, 1 SSI absolute encoder, NetTime function	
X20cDS1119	X20 multifunction digital signal processor, coated, 3 digital channels 5 V (symmetrical) configurable as inputs or outputs, 2 digital input channels 24 V (asymmetrical), max. 2 event counters, 1 universal counter pair as AB counter or up/down counter, linear movement generator (A/B, direction/frequency) with 1 reference pulse, 1 SSI absolute encoder, NetTime function	
	Required accessories	
	Bus modules	
X20BM11	X20 bus module, 24 VDC keyed, internal I/O supply continuous	
X20BM15	X20 bus module, with node number switch, 24 VDC keyed, internal I/O power supply connected through	
X20cBM11	X20 bus module, coated, 24 VDC keyed, internal I/O supply continuous	
	Terminal blocks	
X20TB12	X20 terminal block, 12-pin, 24 VDC keyed	

Table 1: X20DS1119, X20cDS1119 - Order data

4 Technical data

Order number	X20DS1119	X20cDS1119
Short description		
I/O module	3 digital 5 V (symmetrical) channels configurable as inputs or outputs, 2 digital 24 V (asymmetrical) input channels, 1 universal counter pair (2 event counters, AB counter or up/down counter), linear motion generator (A/B, direction/frequency) with 1 reference pulse, SSI absolute encoder, relative or absolute moments of input edges with microsecond resolution, time-triggered I/O, I/O oversampling	
General information		
B&R ID code	0xA067	0xE20D
Status indicators	I/O function per channel, operating state, module status	
Diagnostics		
Module run/error	Yes, using LED status indicator and software	
Inputs/Outputs	Yes, using LED status indicator	
Power consumption		
Bus	0.01 W	
Internal I/O	1.5 W	
Additional power dissipation caused by actuators (resistive) [W]	-	
Type of signal lines	Shielded lines must be used for all signal lines.	
Certifications		
CE	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 GL	Temperature: **B** (0 - 55°C) Humidity: **B** (up to 100%) Vibration: **B** (4 g) EMC: **B** (bridge and open deck)	
LR	ENV1	
KR	Yes	
ABS	Yes	
EAC	Yes	
KC	Yes	-
Linear motion generator		
Quantity	1	
Encoder outputs	5 V, symmetrical (A/B, direction/frequency)	
Counter size	16/32-bit	
SSI absolute encoder		
Quantity	1	
Counter size	Up to 32-bit depending on encoder	
Max. transfer rate	1 Mbit/s	

Table 2: X20DS1119, X20cDS1119 - Technical data

Order number	X20DS1119	X20cDS1119
Encoder signal	5 V, symmetrical	
Encoder power supply		
5 VDC	±5%, module-internal, max. 300 mA	
24 VDC	Module-internal, max. 300 mA	
Digital inputs 5 VDC		
Quantity	Up to 3, configuration as input or output using software	
Nominal voltage	5 VDC differential signal, EiA RS485 standard	
Input characteristics per EN 61131-2	Type 1	
Input frequency	600 kHz	
Common-mode range	-7 V ≤ V_{CM} ≤ +12 V	
Insulation voltage between encoder and bus	500 V_{eff}	
Overload characteristics of encoder power supply	Short-circuit proof, overload-proof	
Input filter		
Hardware	≤200 ns	
Software	-	
Additional functions	SSI absolute encoder, universal counter pair	
Digital inputs 24 VDC		
Quantity	2	
Nominal voltage	24 VDC	
Input characteristics per EN 61131-2	Type 1	
Input circuit	Sink	
Input voltage	24 VDC -15% / +20%	
Input filter		
Hardware	≤2 µs	
Software	-	
Input current at 24 VDC	Approx. 3.4 mA	
Input resistance	Approx. 7.19 kΩ	
Input frequency	100 kHz	
Switching threshold		
Low	<5 VDC	
High	>15 VDC	
Insulation voltage between channel and bus	500 V_{eff}	
Additional functions	Latch function for universal counter pair	
Universal counter pair		
Quantity	1	
Operating modes	2x event counter, up/down counter, AB counter	
Encoder inputs	5 V, symmetrical	
Counter size	16/32-bit	
Input frequency	Max. 600 kHz	
Evaluation		
AB counter	4x [1)]	
Event counters	2x	
Up/Down counter	2x	
Encoder power supply		
5 VDC	±5%, module-internal, max. 300 mA	
24 VDC	Module-internal, max. 300 mA	
Digital outputs 5 VDC		
Quantity	Up to 3, configuration as input or output using software	
Type	5 VDC differential signal, EiA RS485 standard	
Output circuit	Sink and/or source	
Output protection	Short-circuit protection	
Variant	Push/Pull/Push-Pull	
Nominal voltage	5 VDC	
Output current	Max. 65 mA	
Diagnostic status	Output is readable.	
Switching frequency	Max. 500 kHz	
Insulation voltage between channel and bus	500 V_{eff}	
Switching voltage	5 VDC differential signal, EiA RS485 standard	
Additional functions	SSI absolute encoder, linear motion generator	
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	

Table 2: X20DS1119, X20cDS1119 - Technical data

Order number	X20DS1119	X20cDS1119
Ambient conditions		
Temperature		
Operation		
Horizontal mounting orientation	-25 to 60°C	
Vertical mounting orientation	-25 to 50°C	
Derating	-	
Starting temperature	-	Yes, -40°C
Storage	-40 to 85°C	
Transport	-40 to 85°C	
Relative humidity		
Operation	5 to 95%, non-condensing	Up to 100%, condensing
Storage	5 to 95%, non-condensing	
Transport	5 to 95%, non-condensing	
Mechanical properties		
Note	Order 1x terminal block X20TB12 separately. Order 1x bus module X20BM11 separately.	Order 1x terminal block X20TB12 separately. Order 1x bus module X20cBM11 separately.
Pitch	12.5$^{+0.2}$ mm	

Table 2: X20DS1119, X20cDS1119 - Technical data

1) Problems may occur during evaluation due to the system at an input frequency greater than 500 kHz. For additional information, see section "Counter".

5 LED status indicators

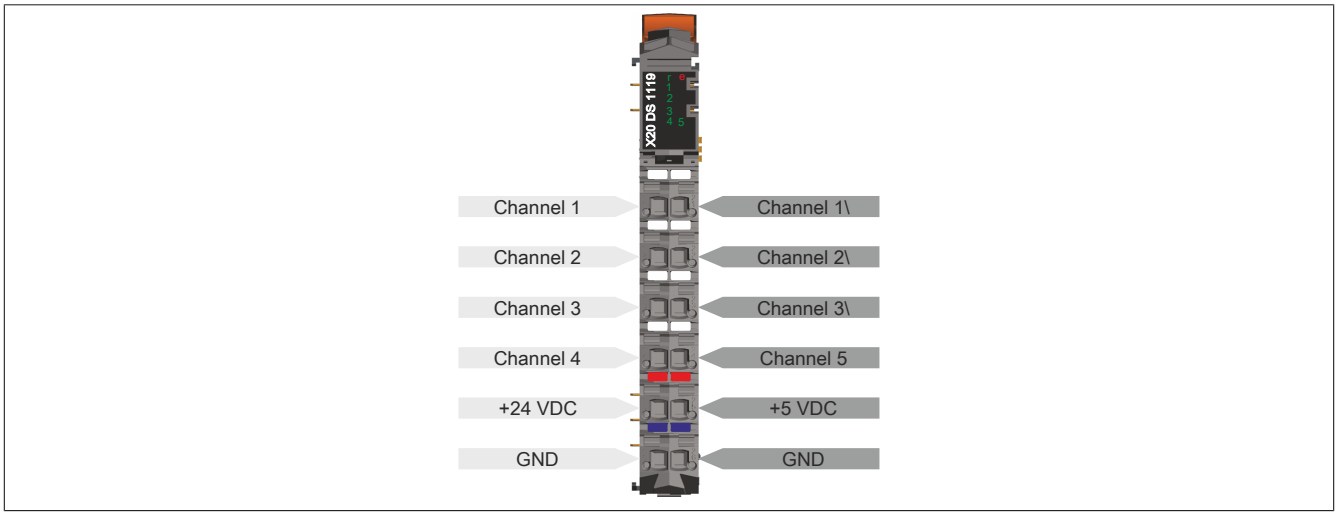
For a description of the various operating modes, see section "Additional information - Diagnostic LEDs" in the X20 system user's manual.

Figure	LED	Color	Status	Description
	r	Green	Off	No power to module
			Single flash	Mode RESET
			Double flash	Mode BOOT (during firmware update)[1]
			Blinking	Mode PREOPERATIONAL
			On	Mode RUN
	e	Red	Off	Module not supplied with power or everything OK
			Single flash	I/O error. Possible causes: • SSI error[2]
			Double flash	System error. Possible causes: • Motion function error[3] • I/O oversampling error[4] • Edge detection error[4]
			Triple flash	I/O error and system error occur together
			On	Error or reset state
	1 - 8	Green		State of the corresponding digital signal

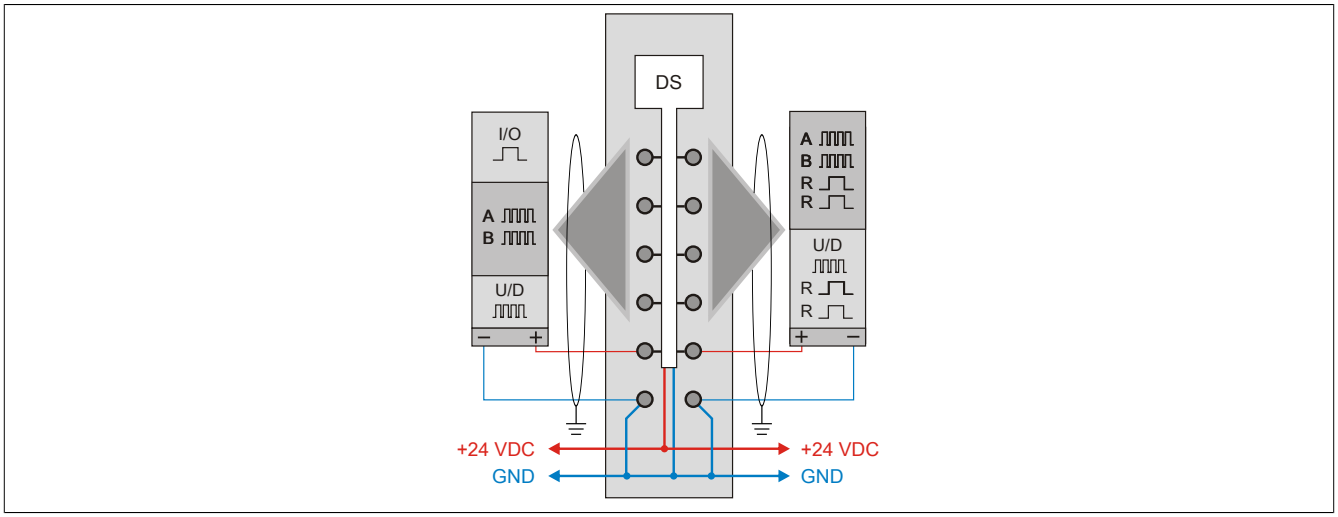
1) Depending on the configuration, a firmware update can take up to several minutes.
2) See register "Error state - SSI" on page 13 for the exact error description.
3) See register "Error state - Motion functions" on page 13 for the exact error description.
4) See register "Error state - Output data and edge detection" on page 12 for the exact error description.

6 Pinout

Shielded cables must be used for all signal lines.

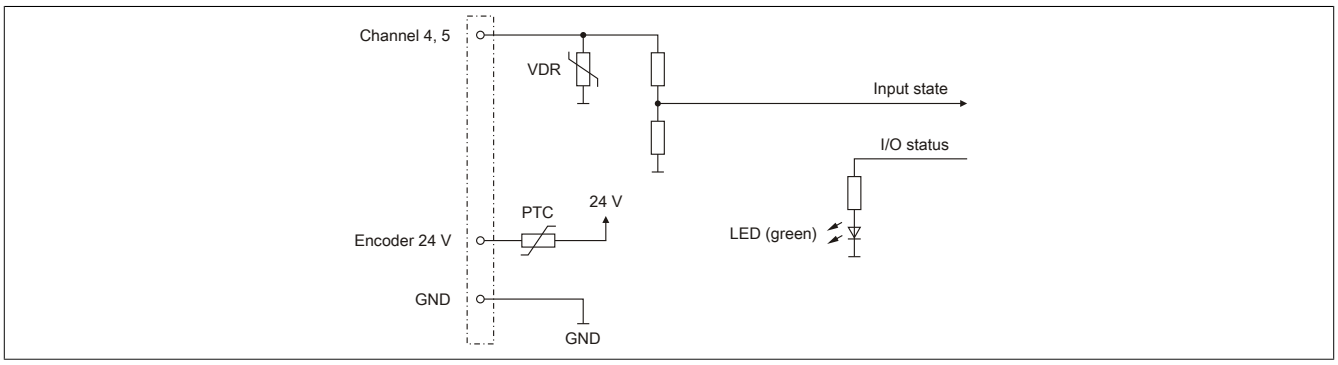


7 Connection example

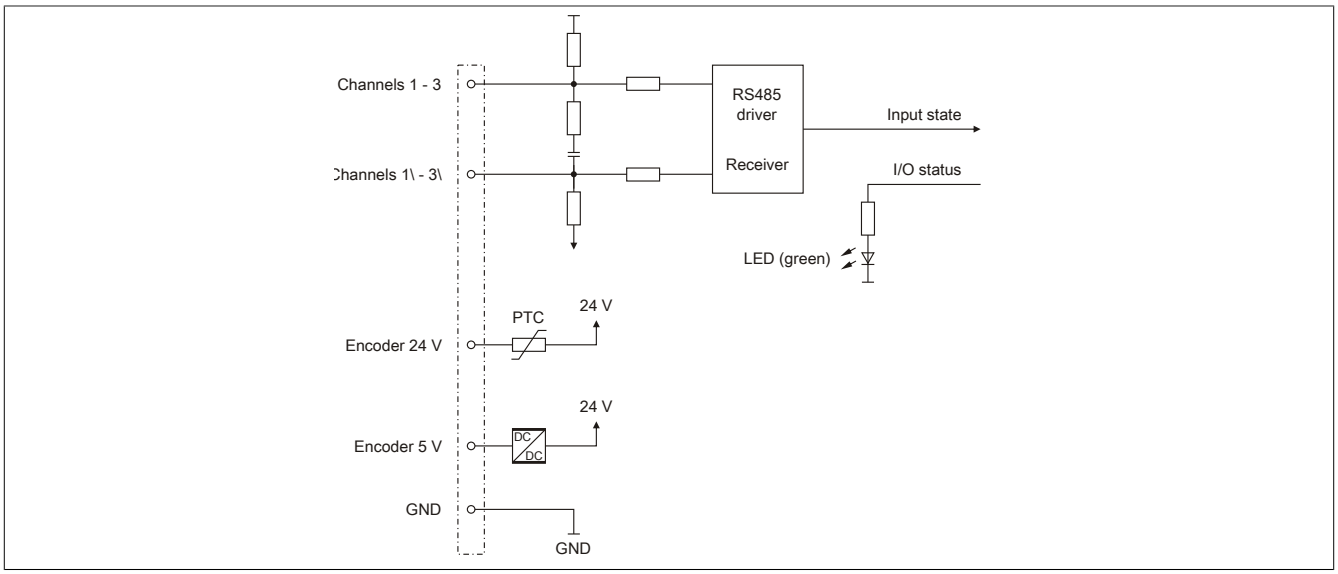


8 Input circuit diagram

Asymmetrical +24 VDC

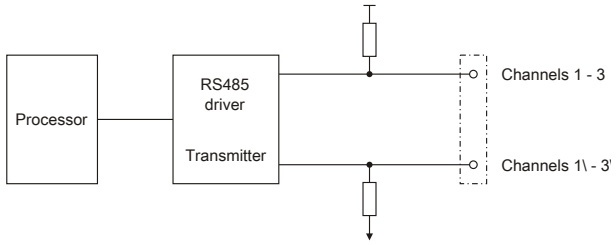


Symmetrical +5 VDC



9 Output circuit diagram

Symmetrical +5 VDC



10 Connection options

Digital input/output

Channel	Function
1	Input / Output (5 V symmetrical)
2	Input / Output (5 V symmetrical)
3	Input / Output (5 V symmetrical)
4	Input (24 V asymmetrical)
5	Input (24 V asymmetrical)

Wiring of the SSI absolute encoder

Channel	Function
1 (input)	Data
2 (output)	Clock

Wiring of the linear motion generator

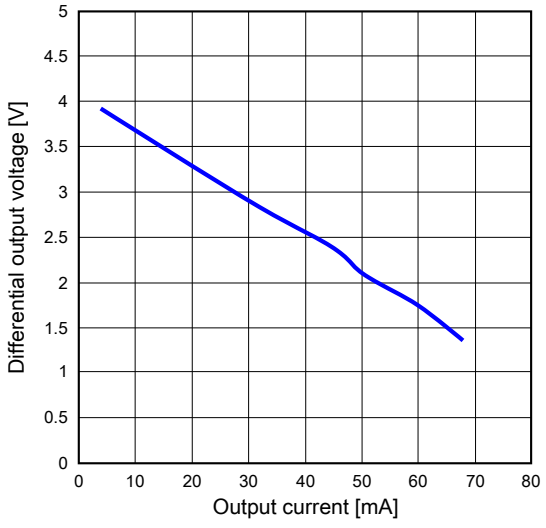
Channel	Up/Down	AB
1 (output)	Direction	A
2 (output)	Frequency	B
3 (output)	Reference	

Wiring of the universal counter pair

Channel	Edge counter	Up/Down counter	Incremental
1 (input)	Input 1	Direction	A
2 (input)	Input 2	Frequency	B
3 (input)		Latch input 1 (R)	
5 (input)		Latch input 2 (E)	

11 Differential output

The following diagram shows that the differential output voltage sinks when the output current rises.



12 Register description

12.1 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.

12.2 Function model 0 - Standard

Register	Name	Data type	Read		Write	
			Cyclic	Acyclic	Cyclic	Acyclic
Configuration - General						
513	CfO_SIframeGenID	USINT				•
Configuration - System timer						
642	CfO_SystemCycleTime	UINT				•
646	CfO_SystemCycleOffset	INT				•
650	CfO_SystemCyclePrescaler	UINT				•
Configuration - Physical I/Os						
769 + (N-1) * 2	CfO_PhyIOConfigCh0N (index N = 1 to 5)	USINT				•
Configuration - Direct I/O						
899	CfO_DirectIOClearMask0_7	USINT				•
903	CfO_DirectIOSetMask0_7	USINT				•
905	CfO_OutputUpdateCycle	USINT				•
Configuration - Oversampled I/O						
1025	CfO_OversampleMode	USINT				•
1027	CfO_OversampleSampleCycleID	USINT				•
1029	CfO_OversampleRelativeCycleID	USINT				•
1031	CfO_OversampleConsumeCycleID	USINT				•
1033	CfO_OversampleOutputBits	USINT				•
1035	CfO_OversampleInputBits	USINT				•
1037	CfO_OversampleOutputWindow	USINT				•
1039	CfO_OversampleInputWindow	USINT				•
1041 + (N*2)	CfO_OversampleConfigInputN (index N = 0 to 3)	USINT				•
1049 + (N*2)	CfO_OversampleConfigOutputN (index N = 0 to 3)	USINT				•
Configuration - Edge detection						
1537	CfO_EdgeDetectPollCycleID	USINT				•
1548	CfO_EdgeDetectEventEnable	UDINT				•
1665 + (N-1) * 16	CfO_EdgeDetectUnit0NMode (index N = 1 to 4)	USINT				•
1667 + (N-1) * 16	CfO_EdgeDetectUnit0NLeading (index N = 1 to 4)	USINT				•
1669 + (N-1) * 16	CfO_EdgeDetectUnit0NMaster (index N = 1 to 4)	USINT				•
1671 + (N-1) * 16	CfO_EdgeDetectUnit0NSlave (index N = 1 to 4)	USINT				•
Configuration - Motion functions						
4097	CfO_FifoSize	USINT				•
4099	CfO_Mode	SINT				•
4101	CfO_SpeedLimit	USINT				•
4103	CfO_FormatAdjust	USINT				•
4105	CfO_TimeStampRange	SINT				•
4107	CfO_PositionRange	SINT				•
4109	CfO_Reference0Range	SINT				•
4111	CfO_Reference1Range	SINT				•
4116	CfO_TimeStampDelay	DINT				•
4124	CfO_SpeedCycleTime_32bit	UDINT				•
4129	CfO_ResolPosition	SINT				•
4131	CfO_ResolSpeed	SINT				•
4220	CfO_AccelDataInit	UDINT				•
4260	CfO_Reference0Start	DINT				•
4268	CfO_Reference0StopMargin	DINT				•
4276	CfO_Reference1Start	DINT				•
4284	CfO_Reference1StopMargin	DINT				•
Configuration - SSI						
2049	CfO_CycleSelect	USINT				•
2051	CfO_PhysicalMode	USINT				•
2053	CfO_DataBits	USINT				•
2055	CfO_NullBits	USINT				•

Register	Name	Data type	Read		Write	
			Cyclic	Acyclic	Cyclic	Acyclic
Configuration - Universal counter						
6145	CfO_CounterCycleSelect	USINT				•
6147	CfO_CounterMode	USINT				•
6149	CfO_LatchMode	USINT				•
6151	CfO_LatchComparator	USINT				•
6153	CounterControl	USINT			•	
	CounterReset	Bit 0				
	LatchEnable	Bit 1				
Communication - General						
546	ProtocolError (16-bit)	USINT	•			
547	ProtocolError (8-bit)	UINT	•			
550	ProtocolSequenceViolation (16-bit)	UINT	•			
551	ProtocolSequenceViolation (8-bit)	USINT	•			
Communication - Error register						
257	Error state - Output data and edge detection	USINT	•			
	OutputControlError	Bit 4				
	OutputCopyError	Bit 5				
	EdgeDetectError	Bit 6				
259	Error state - SSI	USINT	•			
	SSICycleTimeViolation	Bit 0				
	SSIParityError	Bit 1				
261	Error state - Motion functions	USINT	•			
	MovFifoEmpty	Bit 0				
	MovFifoFull	Bit 1				
	MovTargetTimeViolation	Bit 2				
	MovMaxFrequencyViolation	Bit 3				
321	Acknowledging error messages - Output data and edge detection	USINT			•	
	QuitOutputControlError	Bit 4				
	QuitOutputCopyError	Bit 5				
	QuitEdgeDetectError	Bit 6				
323	Acknowledging error messages - SSI	USINT			•	
	SSIQuitCycleTimeViolation	Bit 0				
	SSIQuitParityError	Bit 1				
325	Acknowledging error messages - Motion functions	USINT			•	
	MovQuitFifoEmpty	Bit 0				
	MovQuitFifoFull	Bit 1				
	MovQuitTargetTimeViolation	Bit 2				
	MovQuitMaxFrequencyViolation	Bit 3				
Communication - System timer						
683	SDCLifeCount	SINT	•			
Communication - Direct I/O						
915	Output state	USINT			•	
	DigitalOutput03	Bit 2				
	DigitalOutput04	Bit 3				
	DigitalOutput07	Bit 6				
	DigitalOutput08	Bit 7				
927	Input state	USINT	•			
	DigitalInput01	Bit 0				
	...	...				
	DigitalInput08	Bit 7				
Communication - Oversampled I/O (output)						
1059	Oversampling configuration	USINT			•	
	OversampleEnable	Bit 0				
	OversampleOutputValidate	Bit 1				
1063	OversampleOutputCycle	USINT			•	
	OversampleSampleOffset	USINT			•	
1088 + N	OversampleOutput0NSample1_8 (index N = 1 to 4)	USINT			•	
1092 + N	OversampleOutput0NSample9_16 (index N = 1 to 4)	USINT			•	
1096 + N	OversampleOutput0NSample17_24 (index N = 1 to 4)	USINT			•	
1100 + N	OversampleOutput0NSample25_32 (index N = 1 to 4)	USINT			•	
1104 + N	OversampleOutput0NSample33_40 (index N = 1 to 4)	USINT			•	
1108 + N	OversampleOutput0NSample41_48 (index N = 1 to 4)	USINT			•	
1112 + N	OversampleOutput0NSample49_56 (index N = 1 to 4)	USINT			•	
1116 + N	OversampleOutput0NSample57_64 (index N = 1 to 4)	USINT			•	
Communication - Oversampled I/O (input)						
1074	OversampleInputTime	INT	•			
1079	OversampleInputCycle	USINT	•			
1120 + N	OversampleInput0NSample64_57 (index N = 1 to 4)	USINT	•			
1124 + N	OversampleInput0NSample56_49 (index N = 1 to 4)	USINT	•			
1128 + N	OversampleInput0NSample48_41 (index N = 1 to 4)	USINT	•			
1132 + N	OversampleInput0NSample40_33 (index N = 1 to 4)	USINT	•			
1136 + N	OversampleInput0NSample32_25 (index N = 1 to 4)	USINT	•			
1140 + N	OversampleInput0NSample24_17 (index N = 1 to 4)	USINT	•			

Register	Name	Data type	Read		Write	
			Cyclic	Acyclic	Cyclic	Acyclic
1144 + N	OversampleInput0NSample16_9 (index N = 1 to 4)	USINT	•			
1148 + N	OversampleInput0NSample8_1 (index N = 1 to 4)	USINT	•			
Communication - Edge detection						
1794 + (N-1) * 32	EdgeDetect0NMastercount (16-bit) (index N = 1 to 4)	INT	•			
1795 + (N-1) * 32	EdgeDetect0NMastercount (8-bit) (index N = 1 to 4)	SINT	•			
1798 + (N-1) * 32	EdgeDetect0NSlavecount (16-bit) (index N = 1 to 4)	INT	•			
1799 + (N-1) * 32	EdgeDetect0NSlavecount (8-bit) (index N = 1 to 4)	SINT	•			
1804 + (N-1) * 32	EdgeDetect0NDifference (32-bit) (index N = 1 to 4)	DINT	•			
1806 + (N-1) * 32	EdgeDetect0NDifference (16-bit) (index N = 1 to 4)	INT	•			
1812 + (N-1) * 32	EdgeDetect0NMastertime (32-bit) (index N = 1 to 4)	DINT	•			
1814 + (N-1) * 32	EdgeDetect0NMastertime (16-bit) (index N = 1 to 4)	INT	•			
1820 + (N-1) * 32	EdgeDetect0NSlavetime (32-bit) (index N = 1 to 4)	DINT	•			
1822 + (N-1) * 32	EdgeDetect0NSlavetime (16-bit) (index N = 1 to 4)	INT	•			
Communication - Motion functions						
4225	MovementControl	USINT			•	
	MovEnable - For position control	Bit 0				
	MovEnable - For speed control	Bit 1				
	MovReset - Movement reset (immediate stop)	Bit 7				
4244	MovTargetTime (32-bit)	DINT			•	
4246	MovTargetTime (16-bit)	INT			•	
4252	MovTargetPosition (32-bit)	DINT			•	
4254	MovTargetPosition (16-bit)	INT			•	
4260	MovReference1Start (32-bit)	DINT			•	
4262	MovReference1Start (16-bit)	INT			•	
4268	MovReference1StopMargin (32-bit)	DINT			•	
4270	MovReference1StopMargin (16-bit)	INT			•	
4276	MovReference2Start (32-bit)	DINT			•	
4278	MovReference2Start (16-bit)	INT			•	
4284	MovReference2StopMargin (32-bit)	DINT			•	
4286	MovReference2StopMargin (16-bit)	INT			•	
4212	MovSpeed (32-bit)	DINT			•	
4210	MovSpeed (16-bit)	INT			•	
4220	MovAcceleration (32-bit)	UDINT			•	
4218	MovAcceleration (16-bit)	UINT			•	
4292	MovTimeValid (32-bit)	DINT	•			
4294	MovTimeValid (16-bit)	INT	•			
4300	MovPosition (32-bit)	DINT	•			
4302	MovPosition (16-bit)	INT	•			
Communication - SSI						
2084	SSITimeValid (32-bit)	DINT	•			
2086	SSITimeValid (16-bit)	INT	•			
2092	SSITimeChanged (32-bit)	DINT	•			
2094	SSITimeChanged (16-bit)	INT	•			
2100	SSIPosition (32-bit)	(U)DINT	•			
2102	SSIPosition (16-bit)	UINT	•			
Communication - Universal counter						
6303	LatchCount	SINT	•			
6308	CounterTimeValid (32-bit)	DINT	•			
6310	CounterTimeValid (16-bit)	INT	•			
6324	Counter01TimeChanged (32-bit)	DINT	•			
6326	Counter01TimeChanged (16-bit)	INT	•			
6332	Counter02TimeChanged (32-bit)	DINT	•			
6334	Counter02TimeChanged (16-bit)	INT	•			
6340	CounterValue01 (32-bit)	DINT	•			
6342	CounterValue01 (16-bit)	INT	•			
6348	CounterValue02 (32-bit)	DINT	•			
6350	CounterValue02 (16-bit)	INT	•			
6356	CounterLatch01 (32-bit)	DINT	•			
6358	CounterLatch01 (16-bit)	INT	•			
6364	CounterLatch02 (32-bit)	DINT	•			
6366	CounterLatch02 (16-bit)	INT	•			
6372	CounterRel01 (32-bit)	DINT	•			
6374	CounterRel01 (16-bit)	INT	•			
6380	CounterRel02 (32-bit)	DINT	•			
6382	CounterRel02 (16-bit)	INT	•			

12.3 General

12.3.1 Use with Automation Studio

The module is supported via X2X Link and POWERLINK.

X2X Link supports a maximum of 28 bytes of synchronous cyclic data per module. To optimize use and avoid needless data transfer, data points can be adjusted as needed in Automation Studio, i.e. unnecessary data points can be disabled, and the bit width of data points can be set.

12.3.2 Timestamp function

The timestamp function is based on synchronized timers. If a timestamp event occurs, the module immediately saves the current NetTime. After the respective data is transferred to the CPU, including this precise moment, the CPU can then evaluate the data using its own NetTime (or system time), if necessary.

Conversely, the CPU can predefine output events, apply a timestamp and transfer them to the module. The module then executes the predefined action at the precise moment defined by the CPU.

For more information about NetTime and timestamps, see "NetTime Technology" on page 45.

The resolution of the timestamp is up to 1/8 µs in both directions.

12.3.2.1 Synchronization jitter

Because the CPU – which specifies the X2X NetTime – and the module have different clocks, the module's internal X2X NetTime must be synchronized with the CPU's NetTime. Due to this synchronization, the module's internal X2X NetTime is corrected by a maximum of 1/8 µs per system cycle if necessary. This synchronization jitter becomes noticeable when using the NetTime with 1/8 µs resolution (max. ±1/8 µs).

If a 100% exact 1/8 µs resolution without jitter is required, then the "localtime 1/8 µs" must be used (see register "CfO_EdgeDetectUnitMode" on page 28).

12.4 General registers

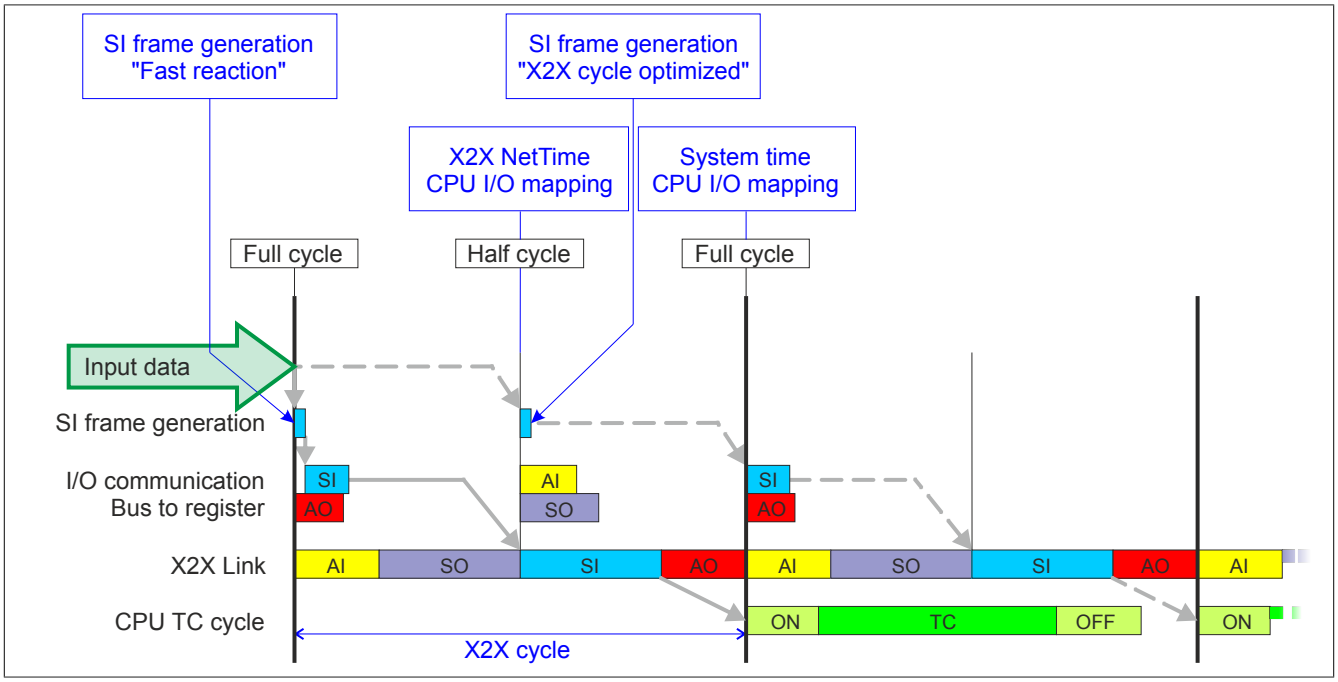
12.4.1 Defining the moment for generating synchronous input data

Name:
CfO_SIframeGenID
"SI frame generation" in the Automation Studio I/O configuration.

When the synchronous input data is generated for transfer is defined in this register. This has a decisive effect on the timing of the input data.

Setting "Fast reaction" causes the input data to be available one X2X cycle sooner in the CPU. However, this setting also has a negative effect on the minimum X2X cycle time.

Data type	Values	Information
USINT	9	X2X cycle optimized
	14	Fast reaction



12.4.2 Number of X2X protocol errors

Name:
ProtocolError

This register contains an error counter that specifies the number of X2X protocol errors. In the I/O configuration, parameter "Network information" can be used to help configure a data point for this register with a bit width of 8 or 16 bits in the I/O mapping.

Data type	Values	Information
USINT	0 to 255	Error counter (8-bit)
UINT	0 to 65535	Error counter (16-bit)

12.4.3 Number of X2X sequence violations

Name:
ProtocolSequenceViolation

This register contains an error counter that specifies the number of X2X sequence violations. In the I/O configuration, parameter "Network information" can be used to help configure a data point with a bit width of 8 or 16 bits in the I/O mapping.

Data type	Values	Information
USINT	0 to 255	Error counter (8-bit)
UINT	0 to 65535	Error counter (16-bit)

12.4.4 System clock counter for checking the validity of the data frame

Name:
SDCLifeCount

Counter that is incremented with each system timer cycle. "SDC information" in the Automation Studio I/O configuration can be used to enable this register in the I/O mapping as data point "SDCLifeCount".

The 8-bit counter register is needed for the SDC software package. It is incremented with the system clock to allow the SDC to check the validity of the data frame.

Data type	Values
SINT	-128 to 127

12.5 Error handling

If one of the functions detects an error, then an error bit is set in one of the error state registers. The application is now able to react to this and acknowledge the errors by setting a respective bit in the "Acknowledge error message" registers. This causes the bit to be reset in the error state register. If the error source persists, then the error bit is set again as soon as the error is detected again (i.e. resetting is not possible).

Error acknowledgment has no effect on the functionality of the module. The module resumes processing, automatically if possible, as soon as the error source is eliminated.

If an error occurs (not a warning), this is indicated by the red "e" LED on the module (double flash). This signal is automatically acknowledged as soon as the error source has been eliminated.

12.5.1 Error state - Output data and edge detection

Name:
OutputControlError
OutputCopyError
EdgeDetectError

Data output errors and cycle time setting errors are indicated in this register.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0 - 3	Reserved	-	
4	OutputControlError	0	No error
		1	The module did not receive new data in time when "Output control mode = Single", meaning that a bit that has already been output would have been output again by the output control buffer.
5	OutputCopyError	0	No error
		1	Oversampling output data could not be copied to the output control buffer (attempted to write to an address outside the oversample output window, for example).
6	EdgeDetectError	0	No error
		1	Edge detection cycle time violation: "EdgeDetectPollCycle" must be ≤255 µs. This error is occurs if the cycle set in register "CfO_EdgeDetectPollCycleID" on page 26 is >255 µs.
7	Reserved	-	

12.5.2 Error state - SSI

Name:
SSICycleTimeViolation
SSIParityError

SSI interface errors are indicated in this register.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0	SSICycleTimeViolation	0	No error
		1	An error occurred. Possible causes: • SSI transfer takes longer than the set "update cycle". • Monostable multivibrator testing is enabled, and the SSI data line does not take on the defined level at the end of the transfer.
1	SSIParityError	0	No error
		1	SSI parity error
2 - 7	Reserved	-	

12.5.3 Error state - Motion functions

Name:
MovFifoEmpty
MovFifoFull
MovTargetTimeViolation
MovMaxFrequencyViolation

Motion function errors are indicated in this register.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0	MovFifoEmpty	0	No error
		1	The position/timestamp FIFO buffer is empty.
1	MovFifoFull	0	No error
		1	The position/timestamp FIFO buffer has overshot the size set in register "FifoSize" on page 32.
2	MovTargetTimeViolation	0	No error
		1	This occurs if the moment set in register "MovTargetTime" on page 37 is already in the past.
3	MovMaxFrequencyViolation	0	No error
		1	The maximum output frequency setpoint has overshot the maximum frequency set in register "CfO_SpeedLimit" on page 33.
4 - 7	Reserved	-	

12.5.4 Acknowledging error messages - Output data and edge detection

Name:
QuitOutputControlError
QuitOutputCopyError
QuitEdgeDetectError

Error messages from register "Error state - Output data and edge detection" on page 12 can be acknowledged by setting the corresponding bits in this register.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0 - 3	Reserved	-	
4	QuitOutputControlError	0	No change
		1	Acknowledge error
5	QuitOutputCopyError	0	No change
		1	Acknowledge error
6	QuitEdgeDetectError	0	No change
		1	Acknowledge error
7	Reserved	-	

12.5.5 Acknowledging error messages - SSI

Name:
SSIQuitCycleTimeViolation
SSIQuitParityError

Error messages from register "Error state - SSI" on page 13 can be acknowledged by setting the corresponding bits in this register.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0	SSIQuitCycleTimeViolation	0	No change
		1	Acknowledge error
1	SSIQuitParityError	0	No change
		1	Acknowledge error
2 - 7	Reserved	-	

12.5.6 Acknowledging error messages - Motion functions

Name:
MovQuitFifoEmpty
MovQuitFifoFull
MovQuitTargetTimeViolation
MovQuitMaxFrequencyViolation

Error messages from register "Error state - Motion functions" on page 13 can be acknowledged by setting the corresponding bits in this register.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0	MovQuitFifoEmpty	0	No change
		1	Acknowledge error
1	MovQuitFifoFull	0	No change
		1	Acknowledge error
2	MovQuitTargetTimeViolation	0	No change
		1	Acknowledge error
3	MovQuitMaxFrequencyViolation	0	No change
		1	Acknowledge error
4 - 7	Reserved	-	

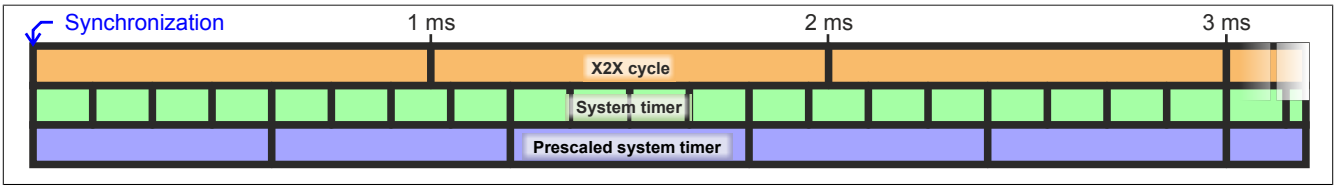
12.6 System timer

The module's individual functions all depend on a system timer. This internal "system cycle time" can be set from 25 to 255 µs. The functions can also be run using a configurable "prescaled system timer" to minimize the load on the module, thereby making it possible to use the shortest possible X2X cycle time.

The cycle of the "prescaled system timer" (and system timer) is referenced with the X2X Link as soon as the module has been started up and the X2X Link has been initialized. Since the system timer and the module's internal NetTime use the same clock, the two run synchronously from that point on. An X2X cycle time that is not a multiple of the system cycle time results in an offset, which can be calculated, however.

The following values apply to the following example:

X2X cycle	1 ms
System timer	150 µs
Prescaled system timer	4



12.6.1 Setting the cycle time of the system timer

Name:
CfO_SystemCycleTime
"Cycle time" in the Automation Studio I/O configuration.

The cycle time of the system timer can be set in steps of 1/8 µs in this register. The value entered in the Automation Studio I/O configuration is automatically multiplied by 8.

Information:

A setting <50 µs has a negative effect on the minimum X2X cycle time!

Data type	Values	Information
UINT	200 to 2047	System timer cycle time in steps of 1/8 µs (25 to 255.875 µs)

12.6.2 Offsetting the synchronization moment of the system cycle

Name:
CfO_SystemCycleOffset
"Cycle offset" in the Automation Studio I/O configuration.

The synchronization moment for the system cycle can be offset in steps of 1/8 µs in this register. The value entered in the Automation Studio I/O configuration is automatically multiplied by 8.

Data type	Values	Information
INT	-32768 to 32767	Cycle offset in steps of 1/8 µs (-4096 to 4095.875 µs)

12.6.3 Configuration of the cycle prescaler

Name:
CfO_SystemCyclePrescaler
"Cycle prescaler" in the Automation Studio I/O configuration.

The prescaler for setting the prescaled system timer can be configured in this register. The cycle time of the specified system timer is a product of the system timer multiple set in this register.

The "prescaled system timer" can be used as an alternative time source for the individual functions. This is useful if a function requires a very short system cycle. To reduce the load on the module in such a situation, other functions can be processed in a slow cycle.

Data type	Values	Information
UINT	2 to 128	Multiple of the system timer

12.7 Physical I/O configuration

12.7.1 Configuring the I/O channels

Name:
CfO_PhyIOConfigCh01 to CfO_PhyIOConfigCh05

Each physical I/O channel can be configured individually in these registers.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0	Push driver[1)]	0	Disabled
		1	Enabled, see Output signals.
1	Pull driver[1)]	0	Disabled
		1	Enabled, see Output signals.
2	Input inverted	0	Not inverted
		1	Inverted
3	Output inverted[1)]	0	Not inverted
		1	Inverted, see Output signals.
4 - 7	Output function[1)]	0 to 15	See Overview of output channel functions.

1) Only available for I/O channels 1 to 3

Overview of output channel functions

Values of bits 4 to 7	Output channel 1	Output channel 2	Output channel 3
0	Direct I/O	Direct I/O	Direct I/O
1		SSI clock output	
2	ABR emulation (A)	ABR emulation (B)	
3	Up/Down emulation (direction)	Up/Down emulation (frequency)	Emulation (reference) [1]
4 - 15	Reserved		

1) Applies to both ABR and up/down emulation of output channels 1 and 2

Output signals

For I/O channels 1 to 3, output signals can be operated in mode "Push", "Pull" and "Push-Pull". Output inversion is also available. This results in the following possibilities for the output signals of channels 1 to 3 and channels 1\ to 3\:

Signal to be output	Signal on output channel x or x\					
	Push[1]		Pull[2]		Push-Pull[3]	
	x	x\	x	x\	x	x\
0	Tristate	Tristate	0	1	0	1
1	1	0	Tristate	Tristate	1	0
0 (inverted)[4]	1	0	Tristate	Tristate	1	0
1 (inverted)[4]	Tristate	Tristate	0	1	0	1

1) Bit 0 = 1
2) Bit 1 = 1
3) Bits 0 and 1 = 1
4) Bit 3 = 1

12.8 Direct I/O

Direct I/O makes it possible to use the physical I/Os like normal I/Os. Additionally, the application can only set or reset I/Os (e.g. an output channel is set by the edge generator and manually reset by the application).

12.8.1 Direct operation of the output channel - Reset

Name:
CfO_DirectIOClearMask0_7
"Direct operation of output channel 01" to "Direct operation of output channel 03" in the Automation Studio I/O configuration.

If the bit for the respective channel is set in this register, then the output is reset as soon as its direct I/O output channel (register "DigitalOutput" on page 17 or "DigitalOutput0x" in the Automation Studio I/O mapping) is reset.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0	Output channel 0	0	No change
		1	Reset channel
1	Output channel 1	0	No change
		1	Reset channel
2	Output channel 2	0	No change
		1	Reset channel
3 - 7	Reserved	-	

12.8.2 Direct operation of the output channel - Set

Name:
CfO_DirectIOSetMask0_7
"Direct operation of output channel 01" to "Direct operation of output channel 03" in the Automation Studio I/O configuration.

If the bit for the respective channel is set in this register, then the output is set as soon as its direct I/O output channel (register "DigitalOutput" on page 17 or "DigitalOutput0x" in the Automation Studio I/O mapping) is set.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0	Output channel 0	0	No change
		1	Set channel
1	Output channel 1	0	No change
		1	Set channel
2	Output channel 2	0	No change
		1	Set channel
3 - 7	Reserved	-	

12.8.3 Direct operation of the output channel - Moment of data output

Name:
CfO_OutputUpdateCycle

The moment when data is output is set with this register.

Data type	Values	Information
USINT	10	X2X cycle optimized (jitter-free)
	15	Fast reaction (with jitter)

12.8.4 Output state

Name:
DigitalOutput01 to DigitalOutput03

This register contains the bits for controlling the direct I/O output channels. Depending on the configuration of registers "CfO_DirectIOClearMask0_7" on page 16 and "CfO_DirectIOSetMask0_7" on page 17, the digital outputs are set to the status of the respective bit in this register.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0	DigitalOutput01	0 or 1	Output state of the channel
1	DigitalOutput02	0 or 1	Output state of the channel
2	DigitalOutput03	0 or 1	Output state of the channel
3 - 7	Reserved	-	

12.8.5 Input state

Name:
DigitalInput01 to DigitalInput05

The state of the digital input channels is contained in this register.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0	DigitalInput01	0 or 1	Input state of channel 1
1	DigitalInput02	0 or 1	Input state of channel 2
2	DigitalInput03	0 or 1	Input state of channel 3
3	Reserved	-	
4	DigitalInput04	0 or 1	Input state of channel 4
5	DigitalInput05	0 or 1	Input state of channel 5
6 - 7	Reserved	-	

12.9 Oversampled I/O

"Oversampled I/O" is based on input status buffers and output control buffers. The input data acquisition and output control occur in one sample cycle (one sample cycle corresponds to one bit in the buffer). The precise moment of an input buffer entry is indicated by its position in the buffer and the NetTime assigned to the buffer.

When "Output control mode = Single", every output buffer entry is marked as invalid once it has been executed. This ensures that the outputs are not supplied with invalid data. In this mode, the application needs to ensure that the module is always supplied with valid data.

When using "Output control mode = Continuous" the contents of the buffer are output again if the module is not supplied with new oversample output data.

12.9.1 Addressing the output control buffer

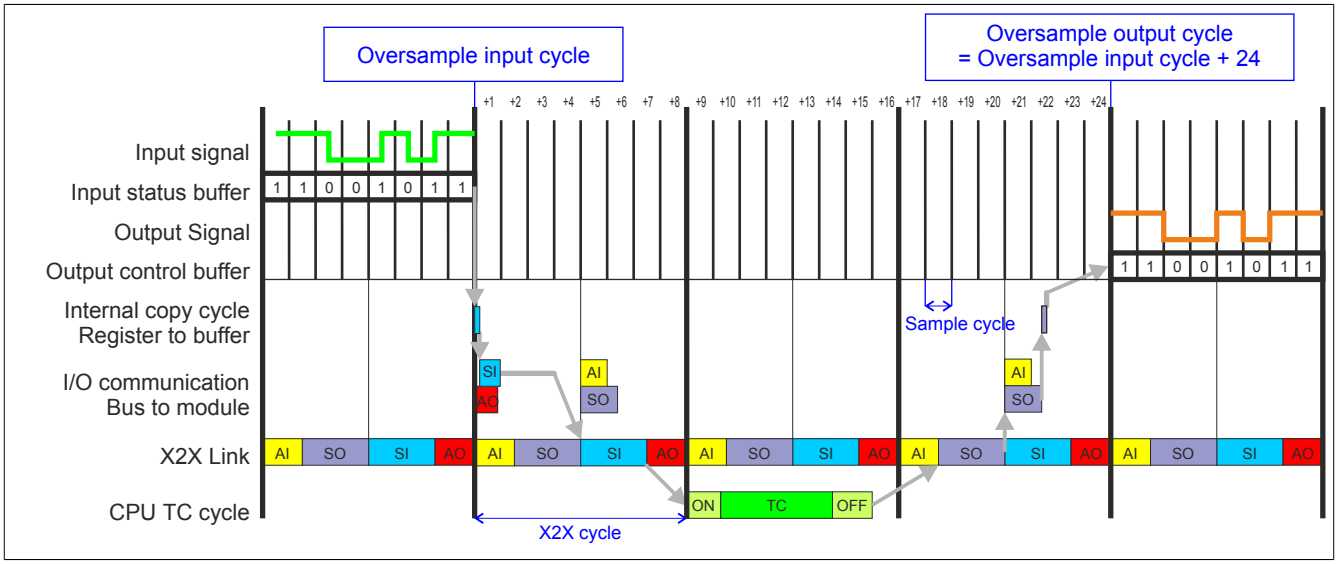
The module has one cyclic 256-bit output control buffer for each oversample channel. One bit is output from these buffers to the configured physical output channels in each "sample cycle". When new data is transferred to one of these buffers, the application must define where in the respective buffer the data should be written to. There are 2 possibilities available for this (absolute or relative "Output mode" in the Automation Studio I/O configuration).

12.9.1.1 Absolute addressing of the output control buffer

With absolute addressing, in each cycle where "OversampleOutputValidate = True", in addition to the oversample output sample data (in the "OversampleOutput0NSample" on page 25 registers) an address must also be transferred in register "OversampleOutputCycle" on page 24. This address defines where in the output control buffer the new data should be copied. In order to calculate this address, the contents of register "OversampleInputCycle" on page 25, which contains the address of the most recently output data, and the transfer time to the module must be taken into account. To help avoid incorrect addressing of the output control buffer, the buffer section that is capable of being written to can be limited using register "OversampleOutputWindow" on page 22. This window will always be shifted relative to the current sample address. An "OutputCopyError" will be triggered if an attempt is made to write to an address that is outside of this window.

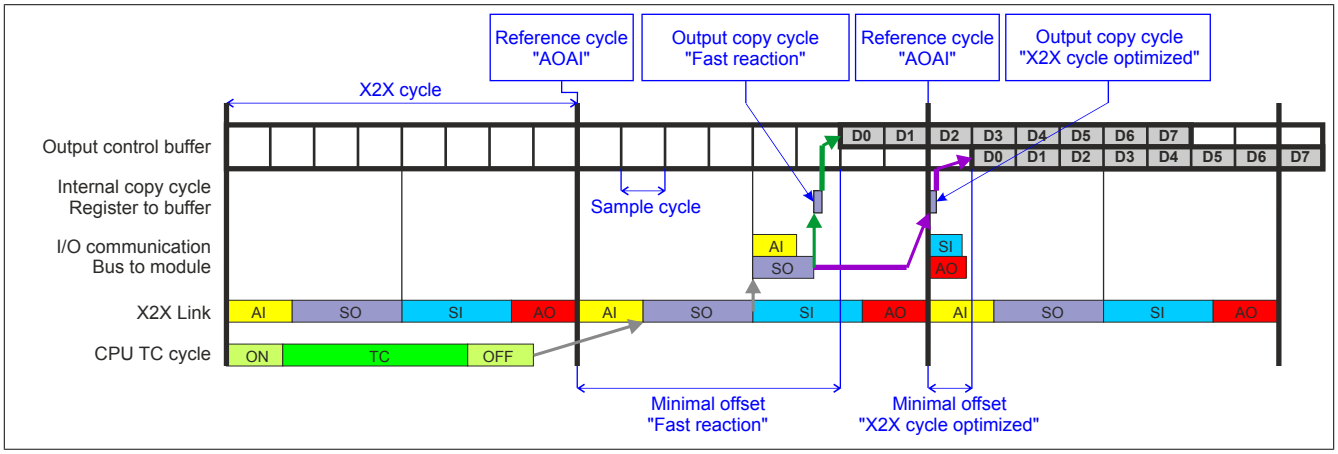
Example

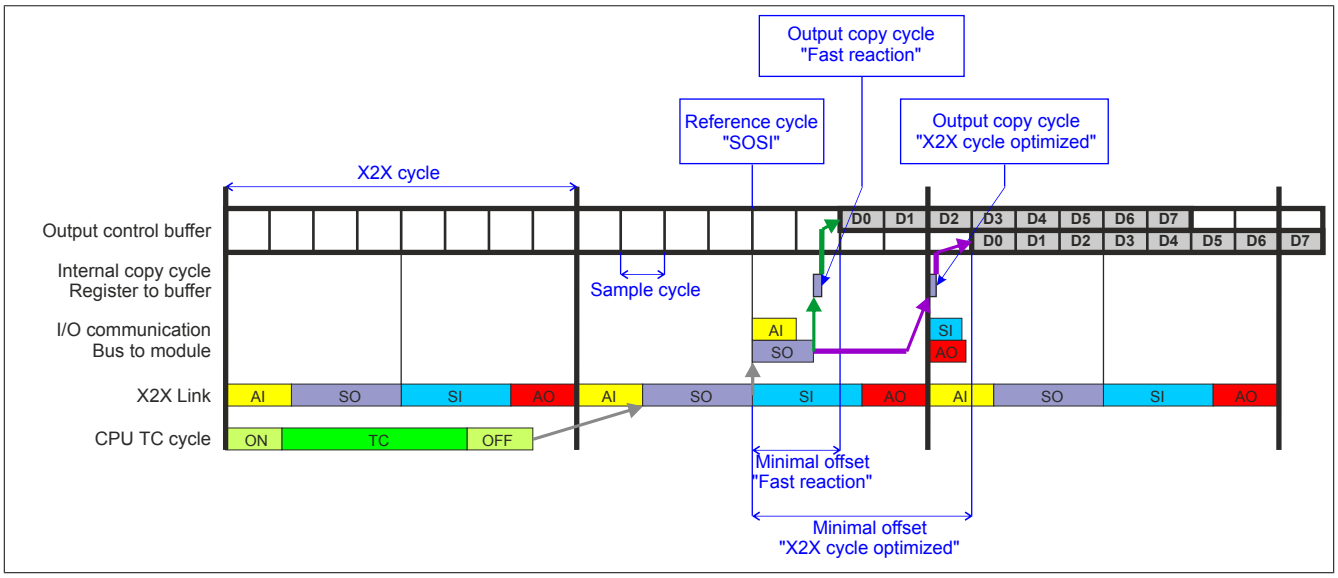
Timing characteristics from the oversample input cycle to the oversample output cycle in absolute output mode ("SI frame generation = Fast reaction", "Output copy cycle = Fast reaction", 8 samples per X2X cycle):



12.9.1.2 Relative addressing of the output control buffer

When "OversampleOutputValidate = True", then the oversample output sample data is automatically copied to an address relative to the last referenced address at the set output copy cycle moment. Register "OversampleSampleOffset" on page 24 serves as the offset. The new data cannot start being output immediately at the output copy cycle moment because it takes time to copy the data from the registers to the buffer. This means that an offset of 0 is not allowed. The relative output control buffer address + offset must point to an address within the "oversample output window". The oversample output window is always offset relative to the current sample address. An OutputCopyError is triggered if an attempt is made to write to an address that is outside of this window.





12.9.2 Configuring the output control buffers

Name:
CfO_OversampleMode
"Output mode" in the Automation Studio I/O configuration
"Output control mode" in the Automation Studio I/O configuration

The output control buffer can be configured globally for all channels in this register.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0	Addressing the output control buffer "Output mode"	0	Absolute addressing of the output control buffer
		1	Relative addressing of the output control buffer
1	Cyclic output control "Output control mode"	0	One-time Output control buffer entry is marked invalid after execution.
		1	Continuous Output control buffer entry is not changed.
2 - 7	Reserved	-	

Cyclic output control

If cyclic output control is enabled, then all data in the output control buffer is marked invalid as soon as it is output ("Output control mode = Single"). An OutputControlError is generated if the module does not receive data in time, thereby causing a situation in which a bit that has already been output would be output in the buffer again. In this type of error situation, the output takes on the "Output default state" configured in register "CfO_OversampleConfigOutput" on page 23.

If cyclic output control is disabled, then the data is output again if the output control buffer overflows ("Output control mode = Continuous").

Information:

All 256 bits of the output control buffer are always output.

12.9.3 Configuration of the source for the sample cycle

Name:
CfO_OversampleSampleCycleID
"Sample cycle" in the Automation Studio I/O configuration.

The source for the sample cycle is configured in this register. During each sample cycle, one bit from the output control buffers of the oversampled I/O channels is output to the configured physical output, and the status of the configured inputs is read into one bit of the respective input status buffer.

Data type	Values	Information
USINT	2	System timer The value set in register "CfO_SystemCycleTime" on page 15 is used as the sample cycle.
	3	Prescaled system timer The "prescaled system timer" is used as sample cycle.
	10	AOAI The sample cycle is clocked with the AOAI interrupt of the X2X cycle.
	14	SOSI The sample cycle is clocked with the SOSI interrupt of the X2X cycle.

12.9.4 Configuration of the source for the user interface reference cycle

Name:
CfO_OversampleRelativeCycleID
"Reference cycle" in the Automation Studio I/O configuration.

The source for the user interface reference cycle is configured in this register.

- The input data is referenced at the moment of the reference cycle. The referenced data is then copied to the "oversample input sample register" on page 26 at the moment of SI frame generation, taking into account the oversample input window.
- With relative addressing of the output control buffer, the new sample data is copied to an address relative to the output control buffer address current to the "reference cycle".
- The reference cycle is also used to reference the sample cycle and thus the output data production and input data acquisition (e.g. to the X2X cycle).

Data type	Values	Information
USINT	2	System timer The value set in register "CfO_SystemCycleTime" on page 15 is used as the reference cycle.
	3	Prescaled system timer The prescaled system timer is used as sample cycle.
	10	AOAI The sample cycle is referenced with the AOAI interrupt of the X2X cycle.
	14	SOSI The sample cycle is referenced with the SOSI interrupt of the X2X cycle.

12.9.5 Defining the moment for copying the data to the output control buffer

Name:
CfO_OversampleConsumeCycleID
"Output copy cycle" in the Automation Studio I/O configuration.

At the time of the output copy cycle, data is copied from the "OversampleOutput0NSample" on page 25 registers into the output control buffer.

When "Output copy cycle = Fast reaction", it is not possible to determine when the data is copied to the output control buffer in either of the two addressing modes. The copy cycles will experience a certain degree of jitter depending on the module load. However, this only affects the moment of the internal copy procedures and therefore the moment of the earliest possible output sample. This will not affect the quality of the output signal. However, "Output copy cycle = Fast reaction" also has a negative effect on the minimum X2X cycle time.

When using the setting "Output copy cycle = X2X cycle optimized", be aware that the sample data cannot start being output immediately at the "Output copy cycle" time due to the internal copy cycle to the output control buffers.

Data type	Values	Information
USINT	10	X2X cycle optimized The output data is copied to the output control buffer with the AOAI interrupt of the X2X cycle.
	15	Fast reaction The output data is copied to the output control buffer immediately after being received.

12.9.6 Number of output bits to be transferred

Name:
CfO_OversampleOutputBits
"User interface size" in the Automation Studio I/O configuration.

Specifies how many bits are transferred from the "OversampleOutput0NSample" on page 25 registers to the output control buffers at the output copy cycle moment.

Data type	Values	Information
USINT	1 to 64	Output bits

12.9.7 Number of input bits to be transferred

Name:
CfO_OversampleInputBits
"User interface size" in the Automation Studio I/O configuration.

Specifies how many bits are transferred from the input status buffer to the "OversampleInput0NSample" on page 26 registers during SI frame generation.

Data type	Values	Information
USINT	1 to 64	Input bits

12.9.8 Write area in the output control buffer

Name:
CfO_OversampleOutputWindow
"Output control mode" in the Automation Studio I/O configuration.

Defines the area in the output control buffer to which data is permitted to be written. The window is always offset relative to the current sample position. (a value of 128, for example, means that the 128 bits following the current sample cycle can be written to). An OutputCopyError is triggered if an attempt is made to write output sample data to a location outside of this window.
In Automation Studio, the value for this register is set to 128 bits with "Output control mode = Single" and to 255 bits with "Output control mode = Continuous".

Data type	Values	Information
USINT	0 to 255	Output window

12.9.9 Defining the moment for referencing input data

Name:
CfO_OversampleInputWindow
"Input mode" in the Automation Studio I/O configuration.

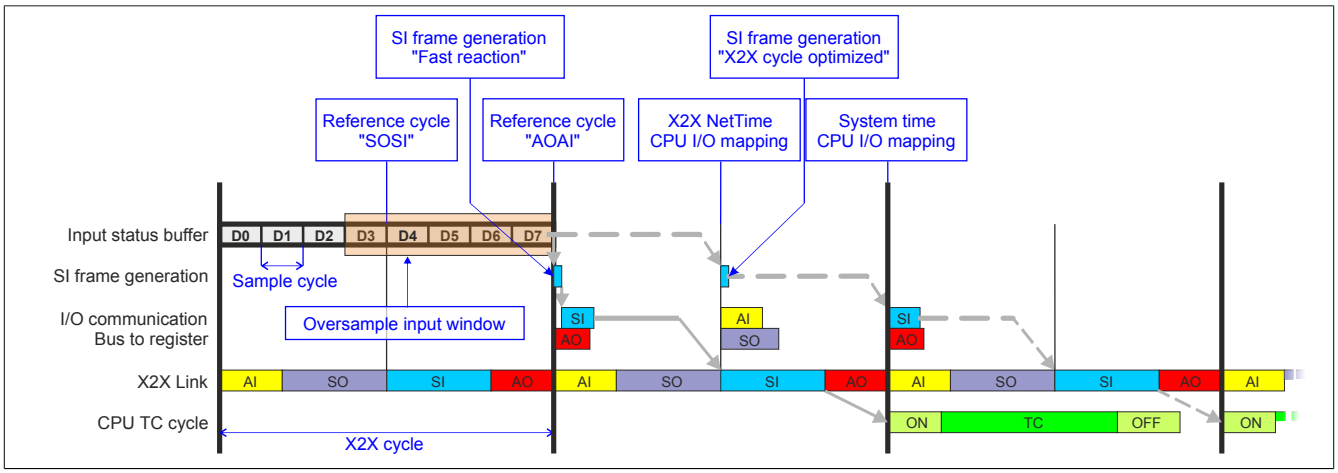
The "oversample input window" defines when the input data is referenced. It is located chronologically before SI frame generation. If the reference moment ("reference cycle" on page 21) is within this window, then the referenced data from the input status buffer is copied to register "OversampleInput0NSample" on page 26. If the moment at which the reference occurs is outside the "oversample input window" then the data that is most recent at the moment of "SI frame generation" is copied from the input status buffer to register "OversampleInput0NSample" on page 26.
This register is limited internally to the value from register "CfO_OversampleInputBits" on page 21.

Information:

As a result, the oversample input time and oversample input cycle are set either at the reference time or at the moment of "SI frame generation".

In Automation Studio, the value for this register is set to 63 with "Input mode = Referenced values" and to 0 with "Input mode = Most recent values".

Data type	Values	Information
USINT	0 to 63	Input window



12.9.10 Configuring the outputs of the oversampling channels

Name:
CfO_OversampleConfigOutput
"Oversample I/O 01 → Output" to "Oversample I/O 04 → Output" in the Automation Studio I/O configuration
"Oversample I/O 01 → Output control" to "Oversample I/O 04 → Output control" in the Automation Studio I/O configuration
"Oversample I/O 01 → Output default value" to "Oversample I/O 04 → Output default value" in der Automation Studio I/O configuration

This register helps configure the outputs of the individual oversample channels.

The "Output default state" bits define which level the respective output takes on before oversampling is started. In addition, the output is set to the defined "Output default state" in the event of an error.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0 - 3	Number of the physical output channel "Oversample I/O 0x → Output"	0	Output channel 1
		1	Output channel 2
		2	Output channel 3
4	Output: Clear "Oversample I/O 0x → Output control"	0	Output cannot be reset by the oversample channel.
		1	Output can be reset by the oversample channel.
5	Output: Set "Oversample I/O 0x → Output control"	0	Output cannot be set by the oversample channel.
		1	Output can be set by the oversample channel.
6	Default output state: Clear "Oversample I/O 0x → Output default value"	0	Output not cleared by default
		1	Output cleared by default
7	Default output state: Set "Oversample I/O 0x → Output default value"	0	Output not set by default
		1	Output set by default

12.9.11 Assigning between the physical input channel and oversample I/O input

Name:
CfO_OversampleConfigInput
"Oversample I/O 01 → Input" to "Oversample I/O 04 → Input" in the Automation Studio I/O configuration.

Which physical input channel an oversample I/O input should be linked to is defined in this register.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0 - 3	Number of the physical input channel	0	Input channel 1
		1	Input channel 2
		2	Input channel 3
		3	Reserved
		4	Input channel 4
		5	Input channel 5
4 - 7	Reserved	-	

12.9.12 Oversampling configuration

Name:
OversampleEnable
OversampleOutputValidate

The oversampling and copy process for the output buffer can be configured in this register.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0	OversampleEnable	0	Disables oversampling (with the next reference cycle)
		1	Enables oversampling (with the next reference cycle)
1	OversampleOutputValidate	0	Disable the copy procedure to the output control buffer.
		1	Enables the copy procedure to the output control buffer. • Used to synchronize the oversampling procedure at startup. • This makes it possible to prevent new data from being transferred to the "OversampleOutput0NSample" on page 25 registers in each X2X cycle.
2 - 7	Reserved	-	

12.9.13 Address of the new output sampling data in the output control buffer

Name:
OversampleOutputCycle

When absolute addressing of the output control buffer is being used, this register specifies the address from which the new output sample data should be copied to the output control buffer.

Data type	Values	Information
USINT	0 to 255	Address of the output control buffer

12.9.14 Offset of new output sample data

Name:
OversampleSampleOffset

When relative addressing of the output control buffer is being used, this register serves as the offset for the new output sample data. (Sample address at the time of the reference cycle + Offset = Address to which the new output sample data is copied in the output control buffer).

Data type	Values	Information
USINT	0 to 255	Offset of output sample data

12.9.15 Oversample output sample data

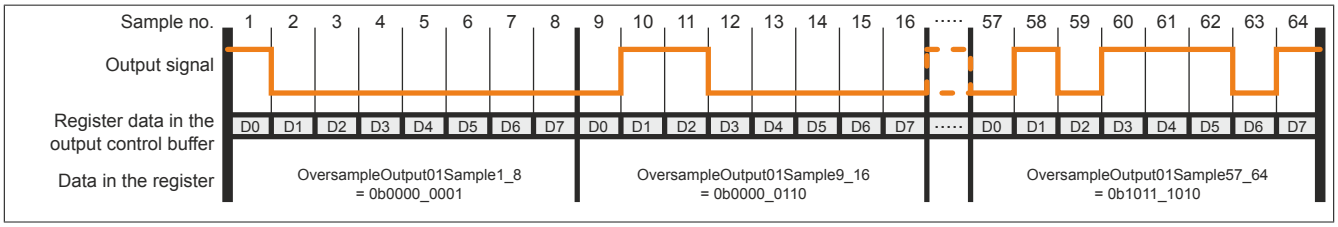
Name:
OversampleOutput01Sample1_8 to OversampleOutput04Sample1_8
OversampleOutput01Sample9_16 to OversampleOutput04Sample9_16
OversampleOutput01Sample17_24 to OversampleOutput04Sample17_24
OversampleOutput01Sample25_32 to OversampleOutput04Sample25_32
OversampleOutput01Sample33_40 to OversampleOutput04Sample33_40
OversampleOutput01Sample41_48 to OversampleOutput04Sample41_48
OversampleOutput01Sample49_56 to OversampleOutput04Sample49_56
OversampleOutput01Sample57_64 to OversampleOutput04Sample57_64

Contains the oversample output sample data. Up to 64 samples (8 bytes) for each oversample I/O channel can be synchronously transferred with a X2X cycle. This data is copied to the specified address (absolute or relative) in the output control buffer at the set output copy cycle. 1 bit of this data is then output during each "sample cycle" to the physical output that is assigned to the oversample I/O channel.
Bit 0 of "OversampleOutputSample1_8" is copied to the output control buffer first, meaning that it is the first bit that is output. "OversampleOutputSample57_64" bit 7 is the last bit to be output.

Data type	Values	Information
USINT	0 to 255	Output sample data

Example

Assignment of "OversampleOutputSample" register data to output signal



12.9.16 X2X NetTime of the input data

Name:
OversampleInputTime

This register contains the 2 low-order bytes of the X2X NetTime from the moment at which the oversample input data was referenced. This provides an easy way to accurately calculate the moment of each individual input sample.

For more information about NetTime and timestamps, see "NetTime Technology" on page 45.

Data type	Values	Information
INT	-32768 to 32767	X2X NetTime of the input data in microseconds

12.9.17 Input status buffer address of the input sample data

Name:
OversampleInputCycle

This register contains the input status buffer address of the input sample data.
In addition, the value in this register can be used to reference an absolute addressing of the output control buffer.

Data type	Values	Information
USINT	0 to 255	Input status buffer address

12.9.18 Input sample data

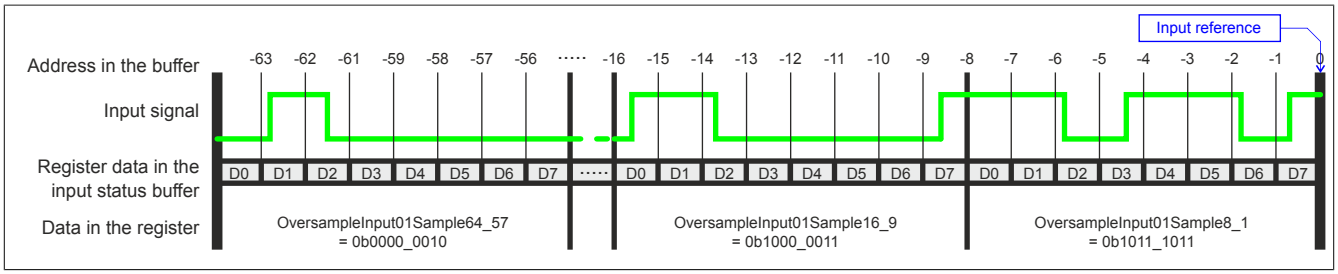
Name:
OversampleInput01Sample8_1 to OversampleInput04Sample8_1
OversampleInput01Sample16_9 to OversampleInput04Sample16_9
OversampleInput01Sample24_17 to OversampleInput04Sample24_17
OversampleInput01Sample32_25 to OversampleInput04Sample32_25
OversampleInput01Sample40_33 to OversampleInput04Sample40_33
OversampleInput01Sample48_41 to OversampleInput04Sample48_41
OversampleInput01Sample56_49 to OversampleInput04Sample56_49
OversampleInput01Sample64_57 to OversampleInput04Sample64_57

The data of the 4 oversample input status buffers are copied to this register at the moment of SI frame generation. A maximum of 64 samples (8 bytes) per oversample I/O channel can be synchronously retrieved from the oversample input status buffer with each X2X cycle.
The most recent input sample bit is stored in "OversampleInputSample8_1" bit 7. The oldest input sample is stored in "OversampleInputSample64_57" bit 0.

Data type	Values	Information
USINT	0 to 255	Input sample data

Example

Input signal and resulting data in "OversampleInputSample"



12.10 Edge detection

The module's edge detection function allows edges to be measured with microsecond precision. The concept is based on a maximum of 4 units. One master and one slave edge can be configured for each unit.

At the moment of each master edge, the NetTime of the master edge and the NetTime of a previous slave edge (if present) are logged. A "master counter" and a "slave counter" can always be used to determine how many edges have been detected since the last X2X cycle.

12.10.1 Configuring the source for the polling cycle

Name:
CfO_EdgeDetectPollCycleID
"Polling cycle" in the Automation Studio I/O configuration.

The source for the polling cycle can be configured in this register.

> **Information:**
>
> **The polling cycle must be ≤255 µs. If the configured cycle >255 µs, EdgeDetectError occurs.**

Data type	Values	Information
USINT	2	System timer The time set in register "CfO_SystemCycleTime" on page 15 is used for the polling cycle.
	3	Prescaled system timer The time set in register "CfO_SystemCyclePrescaler" on page 15 is used for the polling cycle.

12.10.2 Edge detection mode

Name:
CfO_EdgeDetectEventEnable
"Edge detection mode" in the Automation Studio I/O configuration.

The bits in this register define on which edges of the individual input channels an interrupt should be triggered for edge detection.

In the Automation Studio I/O configuration, this register is initialized with 0x00000000 when "Edge detection mode = Polling" and with 0xFFFFFFFF when "Edge detection mode = Event-triggered".

In mode "Event-triggered", the NetTime of each edge is recorded as an interrupt immediately when the edge occurs. However, an extremely large amount of interrupts within a short amount of time can prevent the module from being able to process any other operations in time!.

In mode "Polling", only the NetTime of the first edge that occurs within a polling cycle is recorded. This ensures that the module is not overloaded by too many edges.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0	Physical input 1	0	No interrupt triggered on falling edge
		1	Interrupt triggered on falling edge
1	Physical input 2	0	No interrupt triggered on falling edge
		1	Interrupt triggered on falling edge
2	Physical input 3	0	No interrupt triggered on falling edge
		1	Interrupt triggered on falling edge
3	Reserved	-	
4	Physical input 4	0	No interrupt triggered on falling edge
		1	Interrupt triggered on falling edge
5	Physical input 5	0	No interrupt triggered on falling edge
		1	Interrupt triggered on falling edge
6 - 15	Reserved	-	
16	Physical input 1	0	No interrupt triggered on rising edge
		1	Interrupt triggered on rising edge
17	Physical input 2	0	No interrupt triggered on rising edge
		1	Interrupt triggered on rising edge
18	Physical input 3	0	No interrupt triggered on rising edge
		1	Interrupt triggered on rising edge
19	Reserved	-	
20	Physical input 4	0	No interrupt triggered on rising edge
		1	Interrupt triggered on rising edge
21	Physical input 5	0	No interrupt triggered on rising edge
		1	Interrupt triggered on rising edge
22 - 31	Reserved	-	

12.10.3 Setting the time base, slave edge and master edge

Name:
CfO_EdgeDetectUnit01Mode to CfO_EdgeDetectUnit04Mode
"Time base" in the Automation Studio I/O configuration
"Slave edge" in the Automation Studio I/O configuration
"Master edge" in the Automation Studio I/O configuration

When using a time base with 1/8 µs resolution, keep in mind that the timestamps produced also have a resolution of exactly 1/8 µs. The respective conversions must be made for calculating in combination with the CPU system time or X2X NetTime.

In addition, synchronization jitter also plays a role when using "Time base = Nettime resolution 1/8 usec" (see "Synchronization jitter" on page 10). This means that exactly identical input edges can cause slight differences in the results. If 100% exact 1/8 µs resolution is required, then "Local resolution 1/8 usec" must be used.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0 - 1	"Time base"	0	Local time 1/8 µs (Automation Studio: Local resolution 1/8 usec)
		1	Local time 1 µs (Automation Studio: Local resolution 1 usec)
		2	NetTime 1/8 µs (Automation Studio: Nettime resolution 1/8 usec)
		3	NetTime 1 µs (Automation Studio: Nettime resolution 1 usec)
2 - 5	Reserved	-	
6	"Slave edge"	0	Disabled
		1	Enabled
7	"Master edge"	0	Disabled
		1	Enabled

12.10.4 Load position of the slave time from the FIFO buffer

Name:
CfO_EdgeDetectUnit01Leading to CfO_EdgeDetectUnit04Leading
"Slave leading" in the Automation Studio I/O configuration.

When a slave edge occurs, the current NetTime is always saved within the module. A FIFO buffer is provided inside the module that always stores the last 256 slave stamps (even when a master edge occurs).

This value defines the position from which the slave time should be retrieved from the FIFO buffer when a master edge occurs. This can be used to measure average periodic signals over multiple cycles.

Data type	Values	Information
USINT	0 to 255	Position in the FIFO buffer for slave edges

12.10.5 Source of the master edge per edge detection unit

Name:
CfO_EdgeDetectUnit01Master to CfO_EdgeDetectUnit01Master
"Master edge" in the Automation Studio I/O configuration.

Defines the source of the master edge for the respective "edge detection unit".

Data type	Values	Information
USINT	0	Rising edge on physical input 1
	1	Rising edge on physical input 2
	2	Rising edge on physical input 3
	4	Rising edge on physical input 4
	5	Rising edge on physical input 5
	16	Falling edge on physical input 1
	17	Falling edge on physical input 2
	18	Falling edge on physical input 3
	20	Falling edge on physical input 4
	21	Falling edge on physical input 5

12.10.6 Source of the slave edge per edge detection unit

Name:
CfO_EdgeDetectUnit01Slave to CfO_EdgeDetectUnit04Slave
"Slave edge" in the Automation Studio I/O configuration.

Defines the source of the slave edge for the respective "edge detection unit".

Data type	Values	Information
USINT	0	Rising edge on physical input 1
	1	Rising edge on physical input 2
	2	Rising edge on physical input 3
	4	Rising edge on physical input 4
	5	Rising edge on physical input 5
	16	Falling edge on physical input 1
	17	Falling edge on physical input 2
	18	Falling edge on physical input 3
	20	Falling edge on physical input 4
	21	Falling edge on physical input 5

12.10.7 Number of detected master edges

Name:
EdgeDetect01Mastercount to EdgeDetect04Mastercount

Detected master edges are counted in this register.

Data type	Values	Information
SINT	-128 to 127	Number of detected master edges (8-bit)
INT	-32768 to 32767	Number of detected master edges (16-bit)

12.10.8 Number of detected slave edges

Name:
EdgeDetect01Slavecount to EdgeDetect04Slavecount

Continuously counts the detected slave edges. The contents of this register are only updated on a master edge. This counter can detect if several slave edges occur before a master edge.

Data type	Values	Information
SINT	-128 to 127	Number of detected slave edges (8-bit)
INT	-32768 to 32767	Number of detected slave edges (16-bit)

12.10.9 Difference between master and slave edge

Name:
EdgeDetect01Difference to EdgeDetect04Difference

This register contains the time difference between a master edge and the last slave edge addressed by "Slave leading" on page 28.

Data type	Values	Information
INT	-32768 to 32767	Slave edge / Master edge time difference (16-bit)
DINT	-2,147,483,648 to 2,147,483,647	Slave edge / Master edge time difference (32-bit)

12.10.10 NetTime when a master edge occurs

Name:
EdgeDetect01Mastertime to EdgeDetect04Mastertime

The exact NetTime is copied to this register when a master edge occurs.

For more information about NetTime and timestamps, see "NetTime Technology" on page 45.

Data type	Values	Information
INT	-32768 to 32767	NetTime master edge in microseconds (16-bit)
DINT	-2,147,483,648 to 2,147,483,647	NetTime master edge in microseconds (32-bit)

12.10.11 NetTime when a slave edge occurs

Name:
EdgeDetect01Slavetime to EdgeDetect04Slavetime

When a master edge occurs, the exact NetTime of any slave edge that may have occurred prior to the master edge and addressed by "Slave leading" on page 28 is copied to this register. If multiple slave edges occur before a master edge, then only the NetTime of the last edge that was not ignored by "Slave leading" is stored. The occurrence of multiple edges can be determined by register "EdgeDetectSlavecount" on page 29.

For more information about NetTime and timestamps, see "NetTime Technology" on page 45.

Data type	Values	Information
INT	-32768 to 32767	NetTime slave edge in microseconds (16-bit)
DINT	-2,147,483,648 to 2,147,483,647	NetTime slave edge in microseconds (32-bit)

12.11 Motion functions

Encoder emulation can be used to generate up/down counters (direction/frequency) and ABR encoder signals. The following conditions must be met to achieve an exact match of the position of the module with the remote station:

- Up/Down counter: The remote station must evaluate both rising and falling edges.
- ABR encoder: The remote station must employ 4x evaluation.

The motion function can be operated in 2 different operating modes:

- "Mode "Position control"" on page 30
- "Mode "Speed control"" on page 31

Minimizing jitter

Depending on the configuration of the module, unfavorable system-related jitter times can result in every motion function. In order to increase the smooth running of the motor, however, the flank switching times and thus the unfavorable jitter can be minimized using register "CfO_ResolPosition" on page 35.

12.11.1 Mode "Position control"

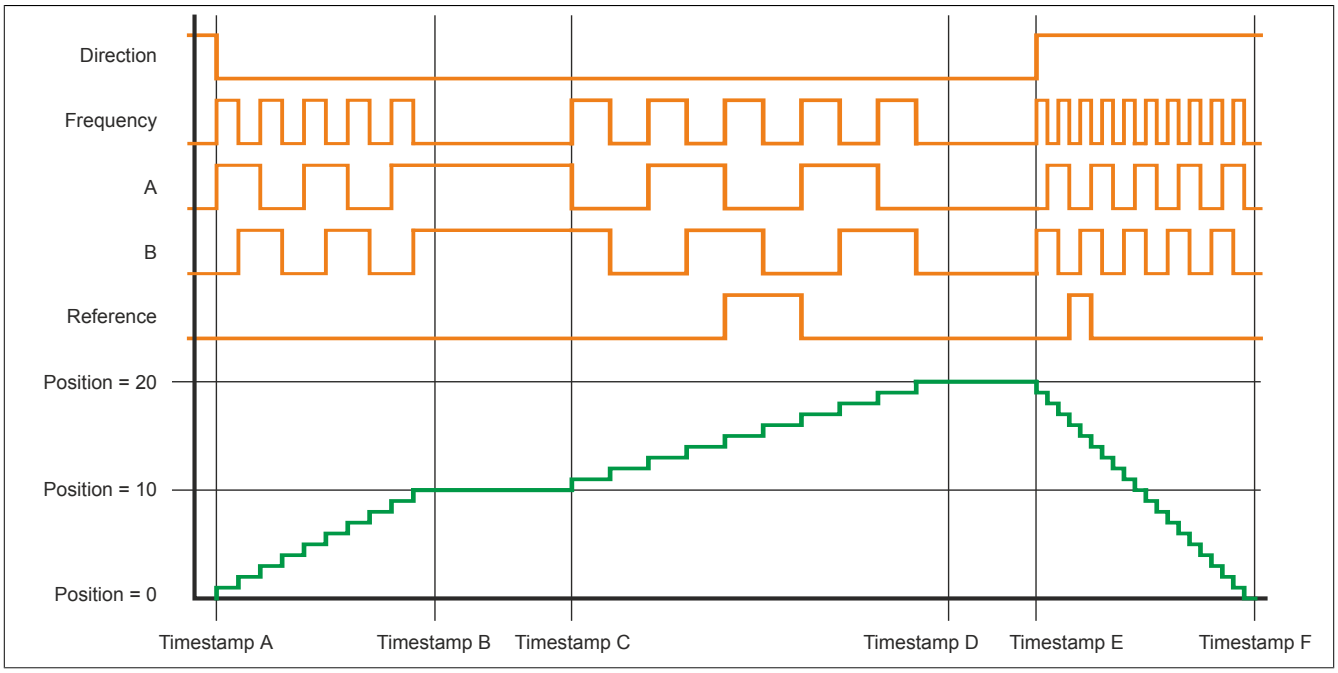
Each time register "MovTargetTime" on page 37 changes, a new position setpoint is transferred from register "MovPosition" on page 37 to the FIFO buffer. The time/position data in the FIFO buffer is then processed in such a manner that the positions are always reached at the moment of the respective timestamps. This means that the module internally ensures that the positions are reached by the set timestamps (number/frequency of the pulses is calculated automatically). The timestamps can be based on the X2X NetTime, the system time of the CPU or register "MovCurrentTime" on page 38. Timestamps that are set in a manner that does not allow the required position change to be reached before the timestamp (output frequency of the pulse would exceed "CfO_SpeedLimit" on page 33) cause a MovMaxFrequencyViolation error.

For more information about NetTime and timestamps, see "NetTime Technology" on page 45.

Selected values for the example "Timing of movement":

Timestamp A = MovTimeValid + 40,000	Position for timestamp A = 0
Timestamp B = Timestamp A + 40,000	Position for timestamp B = 10
Timestamp C = Timestamp B + 25,000	Position for timestamp C = 10
Timestamp D = Timestamp C + 70,000	Position for timestamp D = 20
Timestamp E = Timestamp D + 15,000	Position for timestamp E = 20
Timestamp F = Timestamp E + 40,000	Position for timestamp F = 0

Configuration: Reference pulse = Starting and end position, Starting position = 15, End position = 17



12.11.2 Mode "Speed control"

In mode "Speed control", the application only specifies the speed setpoint. The module returns the current position in register "MovPosition (32-bit)" on page 38.

The default setting (resolSpeed = 24) is designed in such a way that a value of 16,777,216 (0x01000000) in register "MovSpeed" on page 38 results in exactly one increment per "control period".

First, an internal speed value must be calculated:

$$vIntern = vOut * 2^{resolPos}$$

This results in the following correlation for a 32-bit speed specification (data format of the speed values = 32-bit):

$$MovSpeed = vIntern * 2^{resolSpeed} * period$$

Atypically to other registers, when writing to register "MovSpeed (16-bit)", the 2 higher-order bytes of "MovSpeed (32-bit)" are written. This results in the following correlation for direct calculation with "MovSpeed (16-bit)".

$$MovSpeed = \frac{vIntern * 2^{resolSpeed} * period}{2^{16}}$$

Variable	Description	Unit
MovSpeed	Value for register "MovSpeed" (16- or 32-bit)	
vIntern	Internally calculated speed value.	Inc/s
vOut	Desired output speed. Each edge (rising or falling) counts as an increment.	Inc/s
resolPos	Configured value of register "CfO_ResolPosition" on page 35	
resolSpeed	Configured value of register "CfO_ResolSpeed" on page 36	Bits
period	Configured value of register "CfO_SpeedCycleTime_32Bit" on page 35 **Information:** **Must be set in microseconds in Automation Studio. The calculation is performed in s, however.**	s

12.11.3 Performing a movement in mode "Position control"

Several things must be kept in mind when operating the module in order to perform a movement without errors and avoid error messages.

Information:

The specified time/position pairs are not "movement commands", but position data that is continuously processed by the module.

- To allow the module to calculate movement pulses, the first time/position data pair (t, x) is interpreted as the home position. In this case, "t" represents the starting moment and "x" the current position. A movement is not yet performed.
- As long as bit 0 "MovEnable - For position control" on page 37 is set to "1", time/position data pairs must be continuously transmitted to the module. As soon as the last data pair has been processed and the module does not find another data pair in the FIFO buffer, error message MovFifoEmpty is sent (see "Error state - Motion functions" on page 13). In addition, error message MovTargetTimeViolation occurs because no "future moment" for another movement was found.
- To enable a standstill, the time/position data pairs must be specified with an unchanged position but future moments in time.
- Ending the movement with bit 0 = "0" "MovEnable - For position control" on page 37
  This only stops filling the FIFO buffer and subsequently suppresses error message MovFifoEmpty. All entries in the FIFO buffer are still processed. The last specified position is applied as the reference position. As soon as bit 0 = "1" again, all movements are performed relative to this position.
- Ending the movement with bit 7 = "1" "MovReset - Movement reset (immediate stop)" on page 37
  This stops the movement immediately. No more pulses are output. To restart the movement, bit 7 must be set to "0" and bit 0 must be set to "0" for a short time and then back to "1".

12.11.4 FIFO buffer size

Name:
FifoSize
"Number of FIFO buffer entries" in the Automation Studio I/O configuration.

Determines the size of the FIFO buffer for "MovTargetTime" on page 37 and "MovTargetPosition" on page 37. One timestamp and one position that should be reached by the timestamp can be transferred to the FIFO buffer per X2X cycle.

Data type	Values	Information
USINT	0	FIFO buffer disabled
	3	8 entries (2^3)
	4	16 entries (2^4)
	5	32 entries (2^5)
	6	64 entries (2^6)
	7	128 entries (2^7)
	8	256 entries (2^8)

12.11.5 Motion function mode

Name:
CfO_Mode

The mode of the motion functions can be configured in this register.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0	Must be enabled when working without timestamps. Enabled in Automation Studio if: • "Movement = Speed control" • "Movement = Position control" and "Data format / Mode of the preset time value = Local time"	0	Disabled
		1	Enabled
1	If this function is enabled, repositioning is triggered as soon as the value in register "MovPosition" on page 37 changes. Enabled in Automation Studio if: • "Movement = Position control" and "Data format / Mode of the preset time value = Local time"	0	No position control (speed control)
		1	Position control enabled (position control)
2	Reference mode "Configuration of reference pulse #1" in the Automation Studio I/O configuration.	0	Starting/End position
		1	Starting position and span
3 - 7	Reserved	-	

12.11.6 Maximum output frequency

Name:
CfO_SpeedLimit
"Max. movement frequency" in the Automation Studio I/O configuration.

Configures the maximum permissible output frequency and the maximum internal computing frequency. The higher internal computing frequencies of 2, 4, 8, 16, 32 and 64 MHz can only be achieved by configuring n bits as decimal places (see register "CfO_ResolPosition" on page 35).

Data type	Values	Max. increment frequency	Max. frequency for frequency output channel	Max. frequency for A/B output channel
USINT	253	64 MHz	500 kHz	250 kHz
	254	32 MHz		
	255	16 MHz		
	0	8 MHz		
	1	4 MHz		
	2	2 MHz		
	3	1 MHz (default)		
	4	500 kHz	250 kHz	125 kHz
	5	250 kHz	125 kHz	62.5 kHz
	6	125 kHz	62.5 kHz	31.25 kHz

Information:

In Mode "Position control", increment frequencies 16, 32 and 64 MHz are not permitted to be used when 29-bit timestamps are set (see register "CfO_TimeStampRange" on page 34) due to an internal range overrun.

12.11.7 Number of absolute bits that can be output

Name:
CfO_FormatAdjust

The number of bits that can be output absolutely on the signal output are determined in this register (e.g. for a direction/frequency signal, the least significant bit can be output directly on the frequency output; for an AB signal, 2 bits are possible).

Data type	Values	Information
USINT	1 to 2	Number of absolute bits (default value in Automation Studio = 1)

12.11.8 Width of the transferred timestamp data

Name:
CfO_TimeStampRange
"Data format / Mode of the preset time value" in the Automation Studio I/O configuration.

The width of the transferred timestamp data in the module is configured in this register.

Information:

Since the module works internally with 1/8 µs resolution, timestamp data is processed internally with a maximum width of 29 bits.

Data type	Values	Information
SINT	16	16-bit timestamp (selection "16-bit" in the Automation Studio I/O configuration)
	24	24-bit timestamp (selection "Local time" or movement "Speed control" in the Automation Studio I/O configuration)
	29	29-bit timestamp (selection "29-bit" in the Automation Studio I/O configuration)

12.11.9 Number of bits for position control

Name:
CfO_PositionsRange
"Target position range" in the Automation Studio I/O configuration.

The number of bits for position control are configured in this register. "PositionRange" must be reduced if the motion function should follow the absolute value of a 12-bit SSI encoder, for example. In this case, the bit width of the movement position must also be limited to the number of bits of the encoder; otherwise, the movement position would not be overrun if the encoder overflows. In this case, the module would attempt (in the opposite direction) to reach the position of an encoder that had just overflow.

Example

The 12-bit SSI encoder overflows from 2047 to -2048. The module would generate 4096 negative increments if more than 12 bits were set for "CfO_PositionRange" in order to reach position -2048 from the position 2047.

Information:

If the 16-bit value of register "MovPosition" on page 38 is used, then the number of bits of the position must also be limited to ≤16; otherwise, this would also result in incorrect overflow behavior.

Data type	Values	Information
SINT	8 to 32	Number of bits for position control

12.11.10 Number of bits for reference position comparison

Name:
CfO_ReferenceRange
"Reference range" in the Automation Studio I/O configuration.

The number of bits used for the reference position comparison are configured in this register. This makes it possible to generate a reference pulse every 2^n increments.

Information:

The number of bits set in this register is not permitted to exceed the number of bits of registers "MovReferenceStart" on page 36 and "MovReferenceStopMargin" on page 37.

Data type	Values	Information
SINT	4 to 32	Number of bits for position comparison

12.11.11 Timestamp delay

Name:
CfO_TimeStampDelay
"Default time delay" in the Automation Studio I/O configuration.

All timestamps are delayed by the value set in this register in microseconds.

Information:

When setting to "Local time" in register "CfO_TimeStampRange" on page 34, a value at least 2x the X2X cycle time in microseconds must be entered here.

Data type	Values	Information
DINT	0 to 1000000	Timestamp delay in microseconds

12.11.12 Control period for mode "Speed control"

Name:
CfO_SpeedCycleTime_32Bit
"Control period" in the Automation Studio I/O configuration.

The control period for mode "Speed control" can be configured in 1/8 µs steps in this register.

Information:

The value set in the Automation Studio I/O configuration under "Control period" is automatically multiplied by 8 and then used as CfO_SpeedCycleTime_32bit.

Data type	Values	Information
UDINT	400 to 40000	Control period for mode "Speed control"

12.11.13 Minimizing jitter for the position

Name:
CfO_ResolPosition
"Position resolution" in the Automation Studio I/O configuration.

This register contains the number of bits as decimal place for jitter reduction. Internally, the module adds 2^n (n = number of decimal places) to the frequency, which results in edge switching times with a higher resolution. The output switching frequency is not increased from a hardware perspective, but the edge moment is more precise.

Data type	Values	Information
SINT	0	Default, no decimal places
	1 to 14	Selection of bits as decimal places

Information:

It is important to note that each configured decimal place also limits the maximum number range by that number of bits.

For example: 0 decimal places → Maximum position range = 29-bit

3 decimal places → Maximum position range = 26-bit

It is also important to note that parameter "CfO_SpeedLimit" on page 33 must be adjusted for these higher computing frequencies according to the number of configured decimal places.

12.11.14 Minimizing jitter for the speed

Name:
CfO_ResolSpeed
"Speed resolution" in the Automation Studio I/O configuration.

This register contains the number of bits for minimizing the jitter of the speed value as decimal places. Within the module, a 2^n (n = number of decimal places) higher frequency is calculated, resulting in speed values with higher resolution.

A 16-bit or 32-bit speed value is basically configured in the Automation Studio I/O configuration due to the bit limitation. Since the internal calculation is always based on a 32-bit configuration, offset 16 must always be added to the desired decimal places for 16-bit configurations.

Data type	Values	Information
SINT	0 to 31	Selection of bits as decimal places. Default value in Automation Studio = 24

Information:

It is important to note that each configured decimal place also limits the maximum number range by that number of bits.

12.11.15 Acceleration value

Name:
CfO_AccelDataInit
MovAcceleration
"Acceleration value" in the Automation Studio I/O configuration.

The acceleration value in increments per control period² is contained in this register.

- 32-bit: 16777216 (0x01000000) corresponds to 1 increment per control period²
- 16-bit: 256 (0x0100) corresponds to 1 increment per control period²

Data type	Values	Information
UINT	0 to 65535	Acceleration value (16-bit)
UDINT	0 to 4,294,967,296	Acceleration value (32-bit)

12.11.16 Starting position of the reference pulses

Name:
CfO_Reference0Start
MovReferenceStart
"Starting position" in the Automation Studio I/O configuration.

The starting position for the reference pulses is contained in these registers.

If the direction is positive, the output (R) is set when the starting position is reached. In the negative direction, the output is reset as soon as the starting position is undershot.

Data type	Values	Information
INT	-32768 to 32767	Starting position (16-bit)
DINT	-2,147,483,648 to 2,147,483,647	Starting position (32-bit)

12.11.17 End position or range of the reference pulse output

Name:
CfO_Reference0StopMargin
MovReferenceStopMargin
"End position or range" in the Automation Studio I/O configuration.

The end position or range in which the reference pulse is output can be configured in these registers.

If setting "Reference mode = Starting/End position" is used in register "CfO_Mode" on page 33, the output (R) is reset when the end position is reached if the direction is positive. In the negative direction, the output is set as soon as the end position is undershot.

When using "Reference mode = Starting position and span", the contents of this register are added to the starting position and the resulting sum is used as the end position.

Data type	Values	Information
INT	-32768 to 32767	End position (16-bit)
DINT	-2,147,483,648 to 2,147,483,647	End position (32-bit)

12.11.18 Enabling position and speed control

Name:
MovEnable
MovEnable
MovReset

This register can be used to enable position and speed control.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0	MovEnable - For position control	0	Position control disabled
		1	Position control enabled
1	MovEnable - For speed control	0	Speed control disabled
		1	Speed control enabled
2 - 6	Reserved	-	
7	MovReset - Movement reset (immediate stop)	0	No function
		1	Reset active

12.11.19 Timestamp data of the target position

Name:
MovTargetTime

The timestamp data is contained in this register. Each time this register changes, the new position data ("MovTargetPosition" on page 37) and timestamp data are transferred to the FIFO buffer. If bit 1 for speed control "MovEnable = True", the module calculates the output speed (frequency) so that "MovTargetPosition" is reached based on "MovTargetTime".

Data type	Values	Information
INT	-32768 to 32767	Timestamp in microseconds (16-bit)
DINT	-2,147,483,648 to 2,147,483,647	Timestamp in microseconds (32-bit)

Information:

Only 29 bits are processed internally by this register.

12.11.20 Data of the target position

Name:
MovTargetPosition

The position data is contained in this register.

Data type	Values	Information
INT	-32768 to 32767	Position (16-bit)
DINT	-2,147,483,648 to 2,147,483,647	Position (32-bit)

12.11.21 Velocity setpoint

Name:
MovSpeed

The speed setpoint for mode "Speed control" in increments per control period is contained in this register.

- 32-bit: 16,777,216 (0x01000000) corresponds to 1 increment per control period
- 16-bit: 256 (0x0100) corresponds to 1 increment per control period

Data type	Values	Information
INT	-32768 to 32767	Speed setpoint (16-bit)
DINT	-2,147,483,648 to 2,147,483,647	Speed setpoint (32-bit)

12.11.22 NetTime of the current position

Name:
MovTimeValid

The NetTime of the current position is contained in this register.

For more information about NetTime and timestamps, see "NetTime Technology" on page 45.

Data type	Values	Information
INT	-32768 to 32767	NetTime of the current position in microseconds (16-bit)
DINT	-2,147,483,648 to 2,147,483,647	NetTime of the current position in microseconds (32-bit)

12.11.23 Current position

Name:
MovPosition

The current position is contained in this register.

Data type	Values	Information
INT	-32768 to 32767	Current position (16-bit)
DINT	-2,147,483,648 to 2,147,483,647	Current position (32-bit)

12.12 Synchronous Serial Interface

Synchronous Serial Interface makes it possible to receive data from SSI absolute encoders.

2 cables are required for data exchange:

SSI clock:	Generated by the module on output 2 (if configured)
SSI data:	With each clock pulse, one data bit is transferred from the encoder to the module (input 1 can be used as the SSI input).

12.12.1 Procedure for SSI transfer

On the first edge on the SSI clock, a monostable multivibrator is triggered in the encoder and the value currently present in parallel is latched to the shift register (the low level of the monostable multivibrator prevents the transfer of additional values to the shift register during data transfer).

On the next edge, the most significant bit is transferred to the module.

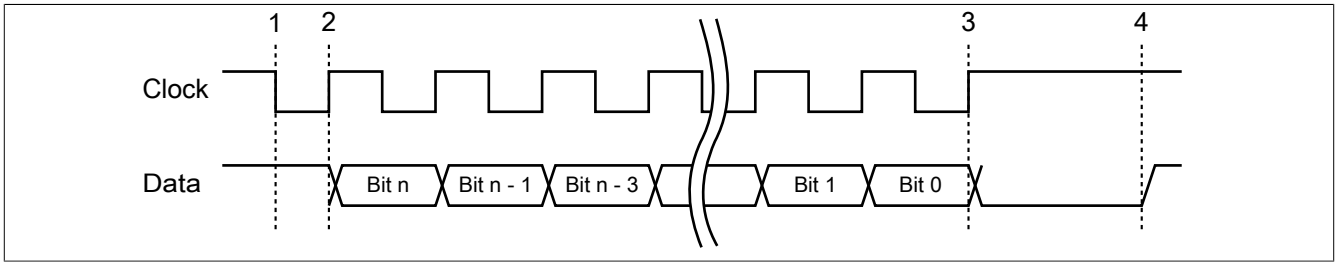
With each additional clock pulse, the next least significant bit is transferred. The clocks constantly retrigger the monostable multivibrator, which prevents its output from accepting new data.

If the number of data bits set in register "CfO_DataBits" on page 40 have been received, the clock sequence is ended.

The monostable multivibrator is no longer triggered; after a time dependent on the encoder, the output of the monostable multivibrator accepts the output level again, thus allowing parallel data to be transferred to the encoder's shift register.

With monostable multivibrator testing, the data line is queried for the configured level before starting a new transfer. This makes it possible to ensure that the monostable multivibrator has really returned before a new transfer is started.

Transferring using Synchronous Serial Interface



Processing the measured value

1) Start bit ... The measured value is saved.
2) Output of the first data bit
3) All data bits are transferred; the monostable multivibrator time starts to run.
4) The monostable multivibrator returns to its initial state; a new transfer can be started.

12.12.2 Update cycle - Starting SSI transfer

Name:
CfO_CycleSelect
"Update cycle" in the Automation Studio I/O configuration.

SSI transfer is started at the update cycle. The clock sequence is generated on the SSI clock output. The first edge of the clock signal triggers the monoflop in the encoder and latches the current position. At the same time, the current NetTime is also logged in register "SSITimeValid" on page 40. As soon as all bits have been transferred via the SSI, the position is passed on with the next "SIframeGenCycle" via the X2X Link. A SSICycleTimeViolation error is reported if the SSI transfer is not completed within the SSI update cycle (e.g. system timer as update cycle). The SSI transfer is still fully completed and then started again with the next update cycle.

Data type	Values	Information
USINT	2	System timer
	3	Prescaled system timer
	10	AOAI
	14	SOSI

12.12.3 Configuring the SSI interface

Name:
CfO_PhysicalMode
"Parity bit" in the Automation Studio I/O configuration
"Monostable multivibrator testing" in the Automation Studio I/O configuration
"Data format" in the Automation Studio I/O configuration
"Clock frequency" in the Automation Studio I/O configuration

The SSI interface is configured in this register.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0 - 1	"Parity bit"[1)]	00	Disabled
		01	Even parity
		10	Odd parity
		11	Ignore (the parity bit is transferred but not evaluated)
2 - 3	"Monostable multivibrator testing"[2)]	00	Disabled
		01	Low level (the data signal is checked for a low level after the monostable multivibrator has returned to its stable state)
		10	High level (data signal is checked for high level after the monostable multivibrator has returned to its stable state)
		11	Ignore (the necessary clock is triggered but not evaluated)
4	"Data format"	0	Encoder with binary data output
		1	Encoder with Gray code. The position data is converted into binary format by the module.
5	Reserved	-	
6 - 7	"Clock frequency"	00	1 MHz
		01	500 kHz
		10	250 kHz
		11	125 kHz

1) If the parity bit is not correct, SSIParityError is generated and the position data is not applied to register "SSIPosition" on page 41.
2) As long as the data signal has not reached the level defined for monostable multivibrator testing after the transfer, no new SSI transfer is started. This will trigger error SSICycleTimeViolation.

12.12.4 Valid number of SSI data bits

Name:
CfO_DataBits
"Valid number of SSI data bits" in the Automation Studio I/O configuration.

Determines the number of valid data bits to be transferred via SSI. The valid data bits are used for "SSIPosition" on page 41.

Data type	Values	Information
USINT	1 to 32	Number of valid data bits

12.12.5 Number of leading zero bits

Name:
CfO_NullBits
"Number of leading zero bits" in the Automation Studio I/O configuration.

The number of leading zero bits can be configured in this register. The leading zero bits may be required before the valid data bits.

Data type	Values	Information
USINT	0 to 31	Number of leading zero bits

12.12.6 NetTime of the current position

Name:
SSITimeValid

The NetTime of the current position is contained in this register.

For more information about NetTime and timestamps, see "NetTime Technology" on page 45.

Data type	Values	Information
INT	-32768 to 32767	NetTime of the current position in microseconds (16-bit)
DINT	-2,147,483,648 to 2,147,483,647	NetTime of the current position in microseconds (32-bit)

12.12.7 NetTime of the last position change

Name:
SSITimeChanged

The NetTime at which the last position change took place is contained in this register.

For more information about NetTime and timestamps, see "NetTime Technology" on page 45.

Data type	Values	Information
INT	-32768 to 32767	NetTime of the last position change in microseconds (16-bit)
DINT	-2,147,483,648 to 2,147,483,647	NetTime of the last position change in microseconds (32-bit)

12.12.8 Current position

Name:
SSIPosition

The current position transferred via the SSI interface is contained in this register.

Data type	Values	Information
INT	-32768 to 32767	Current position (16-bit)
UDINT	0 to 4,294,967,295	Current position (32-bit)
DINT	-2,147,483,648 to 2,147,483,647	

12.13 Counter

The universal counter pair can be used in 3 different modes. Signals up to 600 kHz are reliably detected, depending on the system timer. Up to 4 latch inputs can be configured in all modes. If required, the enabled latch inputs are negated and linked to a latch condition with a logical AND operator. If the latch condition is met, the current counter value is saved to a separate register.

Inputs

Depending on the function model, the physical inputs are permanently configured for the counter.

Mode	Input 1	Input 2	Input 3	Input 4
Edge counter	Counter input for counter 1 Latch input 1	Counter input for counter 2 Latch input 2	- Latch input 3	- Latch input 4
Up/Down counter	Counting direction Latch input 1	Counter frequency Latch input 2	- Latch input 3	- Latch input 4
Incremental encoders	A Latch input 1	B Latch input 2	- Latch input 3	- Latch input 4

Counter evaluation

Counter edges are evaluated at 2 or 4 times the input frequency depending on the counter mode. At an input frequency of 200 kHz, for example, 400000 counting edges are thus acquired for the up/down counter and incremental encoder, but 800000 for AB counters.

The maximum input frequency of 600 kHz is based on a default cycle time of 100 µs. Due to possible jitter, however, counting errors can occur already starting from 2 million counting edges, which corresponds to an input frequency of 500 kHz for the AB counters, due to the system and circuitry. The system cycle time can be reduced accordingly to avoid this.

Latch function

As latch inputs, inputs 1 to 4 can each be queried for a high or low level.

With "Latch mode = Continuous", the counters are latched once as soon as "LatchEnable = True" and the configured latch condition is met. When the latch condition is met again, the counter content is latched again (i.e. a latch event is triggered on every rising edge of the output of the AND operator of all latch inputs).

With "Latch mode = Single", the counters are latched once as soon as "LatchEnable = True" and the configured latch condition is met. When the latch condition is met again, the counter content is not automatically recopied. Another latch event can only be processed after "LatchEnable = False" and another "LatchEnable = True".

12.13.1 Update cycle for counter values

Name:
CfO_CounterCycleSelect
"Update cycle" in the Automation Studio I/O configuration.

The update cycle for the counter values is configured in this register.

Information:

The maximum counting frequency depends on this cycle. The module can process a maximum of 200 increments (edges) within one counter cycle.

Data type	Values	Information
USINT	2	System timer
	3	Prescaled system timer
	10	AOAI moment from X2X cycle
	14	SOSI moment from X2X cycle

12.13.2 Counter mode

Name:
CfO_CounterMode
"Counter mode" in the Automation Studio I/O configuration.

The counter mode is configured in this register.

Data type	Values	Information
USINT	0	Edge counter The two counters serve as edge counters in this mode. The counter input of counter 1 is permanently connected to input 1; the counter input of the second counter is permanently connected to input 2. Both rising and falling edges are counted.
	2	Up/Down counter The up/down counter works according to the direction/frequency principle. Input 1 sets the counting direction (LOW = positive, HIGH = negative); input 2 is used as the counter frequency input. Both rising and falling edges on the counter frequency input are counted.
	3	Incremental encoder (AB counter) When configured as an AB counter, input 1 is used as the A channel and input 2 is used as the B channel. All edges are evaluated (4x evaluation).

12.13.3 Latch mode

Name:
CfO_LatchMode
"Latch mode" in the Automation Studio I/O configuration.

The latch mode is configured in this register.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0	LatchMode	0	One-time
		1	Continuous
1 - 7	Reserved	-	

12.13.4 Latch comparators for counter inputs

Name:
CfO_LatchComparator
"Latch level channel 01" to "Latch level channel 04" in the Automation Studio I/O configuration.

The latch comparators for the counter inputs are configured in this register.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0	Comparison level for the latch comparator on input 1	0	Low
		1	High
...		...	
3	Comparison level for the latch comparator on input 4	0	Low
		1	High
4	Enables the latch comparator on input 1	0	Disabled
		1	Enabled
...		...	
7	Enables the latch comparator on input 4	0	Disabled
		1	Enabled

12.13.5 Clearing counter values and enabling latching

Name:
CounterReset
LatchEnable

Counter values can be deleted or the latch enabled using this register.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0	CounterReset	0	No action
		1	Delete counter
1	LatchEnable	0	Disabled
		1	Enabled
2 - 7	Reserved	-	

12.13.6 Counter for latch events

Name:
LatchCount

Latch events that occur are counted in this register. This counter can be used to detect whether a new value has been latched, for example.

Data type	Values	Information
SINT	-128 to 127	Latch counter

12.13.7 NetTime of the current counter value

Name:
CounterTimeValid

The X2X NetTime of the current counter value is contained in this register.

For more information about NetTime and timestamps, see "NetTime Technology" on page 45.

Data type	Values	Information
INT	-32768 to 32767	NetTime of the current counter value in microseconds (16-bit)
DINT	-2,147,483,648 to 2,147,483,647	NetTime of the current counter value in microseconds (32-bit)

12.13.8 NetTime of the last counter value change

Name:
Counter01TimeChanged to Counter02TimeChanged

The NetTime at which the last change of the respective counter took place is contained in this register.

For more information about NetTime and timestamps, see "NetTime Technology" on page 45.

Data type	Values	Information
INT	-32768 to 32767	NetTime of the last change of the respective counter in microseconds (16-bit)
DINT	-2,147,483,648 to 2,147,483,647	NetTime of the last change of the respective counter in microseconds (32-bit)

12.13.9 Current counter value

Name:
CounterValue01 to CounterValue02

The current counter value of the respective counter is contained in this register.

Data type	Values	Information
INT	-32768 to 32767	Counter value of the respective counter (16-bit)
DINT	-2,147,483,648 to 2,147,483,647	Counter value of the respective counter (32-bit)

12.13.10 Latch counter

Name:
CounterLatch01 to CounterLatch02

As soon as the latch conditions set in register "CfO_LatchComparator" on page 43 are met, the content of the relevant "CounterValue register" on page 44 is copied to this register.

Data type	Values	Information
INT	-32768 to 32767	Latch counter (16-bit)
DINT	-2,147,483,648 to 2,147,483,647	Latch counter (32-bit)

12.13.11 Counter value relative to the last latch

Name:
CounterRel01 to CounterRel02

In this register, the counter value of the respective counter is calculated relative to the last latch of the respective counter.

Data type	Values	Information
INT	-32768 to 32767	Counter value relative to the last latch (16-bit)
DINT	-2,147,483,648 to 2,147,483,647	Counter value relative to the last latch (32-bit)

12.14 NetTime Technology

NetTime refers to the ability to precisely synchronize and transfer system times between individual components of the controller or network (CPU, I/O modules, X2X Link, POWERLINK, etc.).

This allows the moment that events occur to be determined system-wide with microsecond precision. Upcoming events can also be executed precisely at a specified moment.

12.14.1 Time information

Various time information is available in the controller or on the network:

- System time (on the PLC, Automation PC, etc.)
- X2X Link time (for each X2X Link network)
- POWERLINK time (for each POWERLINK network)
- Time data points of I/O modules

The NetTime is based on 32-bit counters, which are increased with microsecond resolution. The sign of the time information changes after 35 min, 47 s, 483 ms and 648 µs; an overflow occurs after 71 min, 34 s, 967 ms and 296 µs.

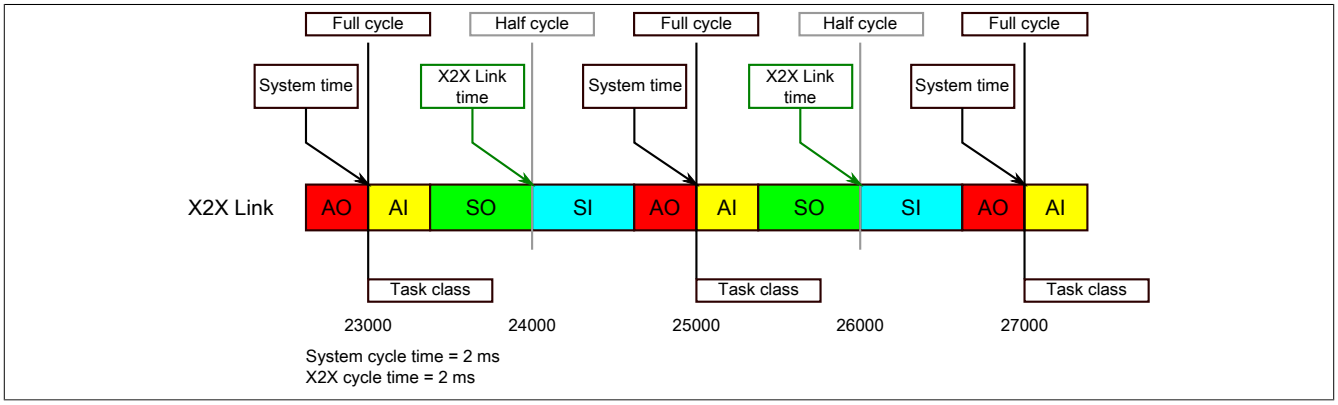
The initialization of the times is based on the system time during the startup of the X2X Link, the I/O modules or the POWERLINK interface.

Current time information in the application can also be determined via library AsIOTime.

12.14.1.1 PLC/Controller data points

The NetTime I/O data points of the PLC or the controller are latched to each system clock and made available.

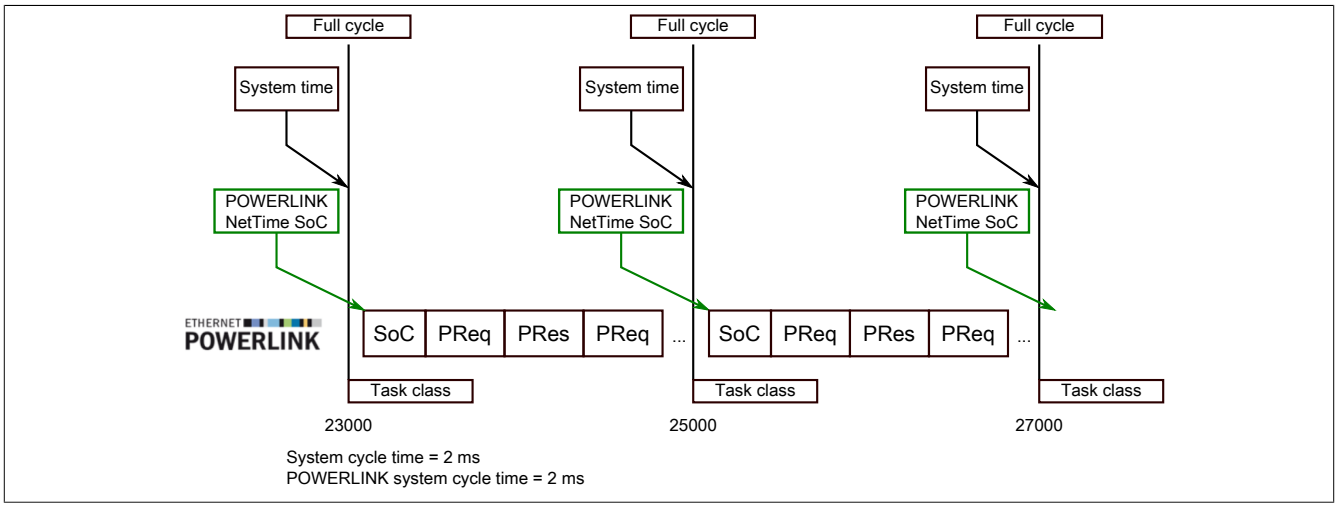
12.14.1.2 X2X Link reference moment



The reference moment on the X2X Link network is always calculated at the half cycle of the X2X Link cycle. This results in a difference between the system time and the X2X Link reference moment when the reference time is read out.

In the example above, this results in a difference of 1 ms, i.e. if the system time and X2X Link reference moment are compared at time 25000 in the task, then the system time returns the value 25000 and the X2X Link reference moment returns the value 24000.

12.14.1.3 POWERLINK reference moment

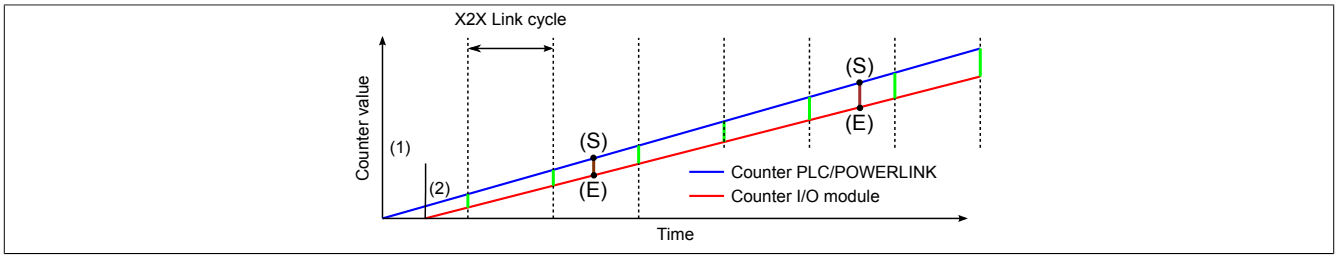


The reference moment on the POWERLINK network is always calculated at the start of cycle (SoC) of the POWERLINK network. The SoC starts 20 µs after the system tick. This results in the following difference between the system time and the POWERLINK reference time:

POWERLINK reference time = System time - POWERLINK cycle time + 20 µs.

In the example above, this means a difference of 1980 µs, i.e. if the system time and POWERLINK reference moment are compared at time 25000 in the task, then the system time returns the value 25000 and the POWERLINK reference moment returns the value 23020.

12.14.1.4 Synchronization of system time/POWERLINK time and I/O module



At startup, the internal counters for the PLC/POWERLINK (1) and the I/O module (2) start at different times and increase the values with microsecond resolution.

At the beginning of each X2X Link cycle, the PLC or the POWERLINK network sends time information to the I/O module. The I/O module compares this time information with the module's internal time and forms a difference (green line) between the two times and stores it.

When a NetTime event (E) occurs, the internal module time is read out and corrected with the stored difference value (brown line). This means that the exact system moment (S) of an event can always be determined, even if the counters are not absolutely synchronous.

Note

The deviation from the clock signal is strongly exaggerated in the picture as a red line.

12.14.2 Timestamp functions

NetTime-capable modules provide various timestamp functions depending on the scope of functions. If a timestamp event occurs, the module immediately saves the current NetTime. After the respective data is transferred to the CPU, including this precise moment, the CPU can then evaluate the data using its own NetTime (or system time), if necessary.

12.14.2.1 Time-based inputs

NetTime Technology can be used to determine the exact moment of a rising edge at an input. The rising and falling edges can also be detected and the duration between 2 events can be determined.

> **Information:**
>
> **The determined moment always lies in the past.**

12.14.2.2 Time-based outputs

NetTime Technology can be used to specify the exact moment of a rising edge on an output. The rising and falling edges can also be specified and a pulse pattern generated from them.

> **Information:**
>
> **The specified time must always be in the future, and the set X2X Link cycle time must be taken into account for the definition of the moment.**

12.14.2.3 Time-based measurements

NetTime Technology can be used to determine the exact moment of a measurement that has taken place. Both the starting and end moment of the measurement can be transmitted.

12.15 Minimum X2X cycle time

The minimum X2X cycle time is strongly dependent on the configured functions and the resulting load on the module. Setting "Fast reaction" and a very short system cycle (<50 µs) generally have a negative effect on the minimum X2X cycle time. This can result in error behavior with short X2X cycle times.