Axioline F function module, 2 counter inputs, 2 incremental encoder inputs

User manual

PHOENIX CONTACT
INSPIRING INNOVATIONS

User manual

Axioline F function module, 2 counter inputs, 2 incremental encoder inputs

UM EN AXL F CNT2 INC2 1F, Revision 05 2019-05-15

This user manual is valid for:

Designation	Version	Order No.
Product designation	Hardware version 03 or later Firmware version 1.20 or later	2688093
AXL F CNT2 INC2 XC 1F	Hardware version 00 or later Firmware version 1.20 or later	2701239

8111_en_05

Table of contents

1 For your safety

Read this user manual carefully and keep it for future reference.

1.1 Labeling of warning notes

This symbol indicates hazards that could lead to personal injury.

There are three signal words indicating the severity of a potential injury.

DANGER
Indicates a hazard with a high risk level. If this hazardous situation is not avoided, it will result in death or serious injury.

WARNING
Indicates a hazard with a medium risk level. If this hazardous situation is not avoided, it could result in death or serious injury.

CAUTION
Indicates a hazard with a low risk level. If this hazardous situation is not avoided, it could result in minor or moderate injury.

This symbol together with the **NOTE** signal word alerts the reader to a situation which may cause damage or malfunction to the device, hardware/software, or surrounding property.

Here you will find additional information or detailed sources of information.

1.2 Qualification of users

The use of products described in this user manual is oriented exclusively to:

- Qualified electricians or persons instructed by them. The users must be familiar with the relevant safety concepts of automation technology as well as applicable standards and other regulations.
- Qualified application programmers and software engineers. The users must be familiar with the relevant safety concepts of automation technology as well as applicable standards and other regulations.

1.3 Product changes

Changes or modifications to hardware and software of the device are not permitted.

Incorrect operation or modifications to the device can endanger your safety or damage the device. Do not repair the device yourself. If the device is defective, please contact Phoenix Contact.

2 Function description of the module

Please refer to the module's data sheet for the technical data of the module including the terminal point assignment and meaning of the diagnostic and status indicators.
It can be downloaded at phoenixcontact.net/products.

2.1 General functions

AXL F CNT2 INC2 1F / AXL F CNT2 INC2 XC 1F

The module is designed for use within an Axioline F station. It is used to acquire digital input signals.

The module consists of two counter inputs (CNT) and two incremental encoder inputs (INC). In addition, it has two digital outputs (Out1/2) which can be controlled either over a CNT channel or an INC channel or, as an alternative, using process data.

The module has inputs which can be used to perform the following functions:

- Controlling the counting: gate inputs G1, G2,
- Changing the counting direction or stopping the counting process: direction/stop inputs Dir1, Dir2
- Latching specific values: latch inputs L1, L2
- Referencing: reference inputs Ref1, Ref2

As an option, these inputs can be used as digital inputs, for instance, to connect limit switches.

You can start up the module with the default parameterization without having to parameterize it. In this case the INC channels are deactivated.

But you can also parameterize the channels with the engineering tool or with the PDI object $0080_{hex.}$ The new parameterization is stored retentively. Thus, it will still be available after a voltage reset.

When the module is reparameterized, all channels will be stopped and then reparameterized. All channels will always be reinitialized, even if you have changed only one channel.

AXL F CNT2 INC2 XC 1F

Thanks to special engineering measures and tests, the module can be used under extreme ambient conditions (see module-specific data sheet).

2.2 Features of the channels

2.2.1 CNT channels

- Detecting digital input signals with an input frequency of up to 300 kHz (one channel wired) or 100 kHz (more than one channel wired)
- Counting the digital input signals with a 32-bit counter (up and down)
- Controlling the counting via a control input (gate) or starting and stopping the counting via process data
- Output control according to two limit values
- Single or periodic counting is possible
- Starting the counting process via the gate signal and stopping the counting via a separate input (direction/stop input)
- Specifying the counting direction via an external input or via process data
- Specifying a start value and an upper or lower limit is possible. The start value is loaded when the channel is restarted or reset.

2.2.2 INC channels

- Detecting digital signals of symmetrical or asymmetrical incremental encoders (5 V or 24 V) with an input frequency of up to 300 kHz (one channel wired) or 100 kHz (more than one channel wired)
- Position detection with a 32-bit counter
- Double or quadruple evaluation of encoder signals
- Evaluating linear or rotary axes (determining the direction of rotation or motion)
- Storing of up to two intermediate values via an external input signal (latch input Lx)
- Detecting an overflow or underflow with linear axes
- Output control according to two limit values
- Various methods to set a reference point
- Monitoring the increments by evaluating the Z signal of the encoder, in order to determine an error of the encoder
- Hardware monitoring of the encoders
  - Monitoring of the encoder supply
  - Open circuit detection
  - Detecting faulty electrical signals with symmetrical encoders

2.3 Maximum input frequencies

The maximum input frequency that can be used for operation depends on the wiring of the channels and whether the Z signal, in the case of a linear axis, is to be monitored via the firmware or not.

Table 2-1 Maximum input frequencies

Condition	Maximum frequency
One channel wired	300 kHz
More than one channel wired	100 kHz
Z signal monitoring via the firmware	100 kHz

2.4 Counter channel functions

The two counter channels (CNT) acquire and process digital input signals with a maximum input frequency of up to 300 kHz (one channel wired) or 100 kHz (more than one channel wired).

The two channels have the same structure and the same functions.

Each of the channels has a counter input (source), a control input (gate) and a direction/stop input (Dir/Stop).

The control and the direction/stop inputs can be used as digital inputs if they are not used directly by the channel. The status of the inputs is mapped to the process data, no matter whether they are used by the CNT channel or not.

There is an additional output available which can be controlled according to the parameterization (see Section “Output control” on page 22).

2.4.1 Counting functions

The digital input pulses at the counter input (S1/S2) are counted. Depending on the parameterization, the counter direction is either specified by the controller or by the direction input.

The following values must be defined during parameterization:

– Start value
  The start value is loaded when the channel is restarted or reset.
  The start value may be any value in the possible counting range of the 32-bit counter.
– Limit values: lower and upper limit
  The lower limit must always have a smaller value than the upper limit.
  During operation you can change the limit values online via the process data. These changes are not stored retentively.
  How to make changes via process data, see Section “OUT process data” on page 23, *SetNewLimitsx* bit.
– Type of counting: single or periodic
  Single counting means that the counter counts once from the start value to the upper or lower limit.
  Periodic counting means that the counter counts periodically between the upper and lower limit. Once a limit is reached, counting continues further from the other limit.
– Output behavior
  When one or both limits are reached, an output can be controlled according to the parameterization (see Section “Output control” on page 22).

2.4.2 Counting depending on the position of the start value

Example 1: Start value between the lower and upper limits

If the start values is between the lower and upper limits, the counter value is also between these limits.

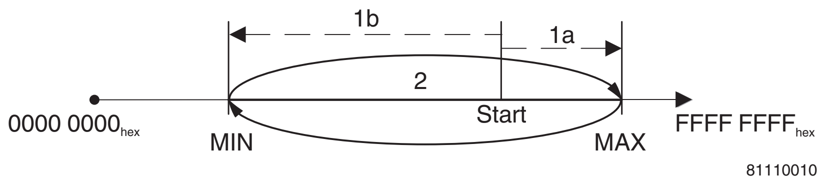

Figure 2-1 Counting with a start value between the limits

MIN	Lower limit
MAX	Upper limit
Start	Start value
1a	Positive counting direction
1b	Negative counting direction
2	Counting between the lower and upper limits

Example 2: Start value outside the lower and upper limits

If the start value is outside the lower and upper limits, it is possible to count in both directions. If one of the maximum limits (0000 0000_{hex} or FFFF $FFFF_{hex}$) is exceeded, the counting continues further from the other maximum limit (see example 2b). If the counter value is then between the lower and upper limits, the counting continues between these limits according to the parameterization and the counter status does not leave these limits.

Example 2a

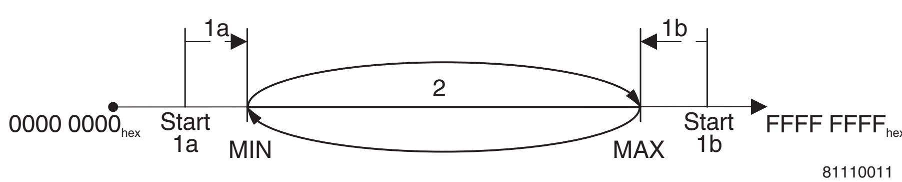

Figure 2-2 Counting with a start value outside the lower and upper limits without reaching the maximum limits

MIN	Lower limit
MAX	Upper limit
Start	Start value
1a	Start value below the lower limit; positive counting direction
1b	Start value above the upper limit; negative counting direction
2	Counting between the lower and upper limits

Example 2b

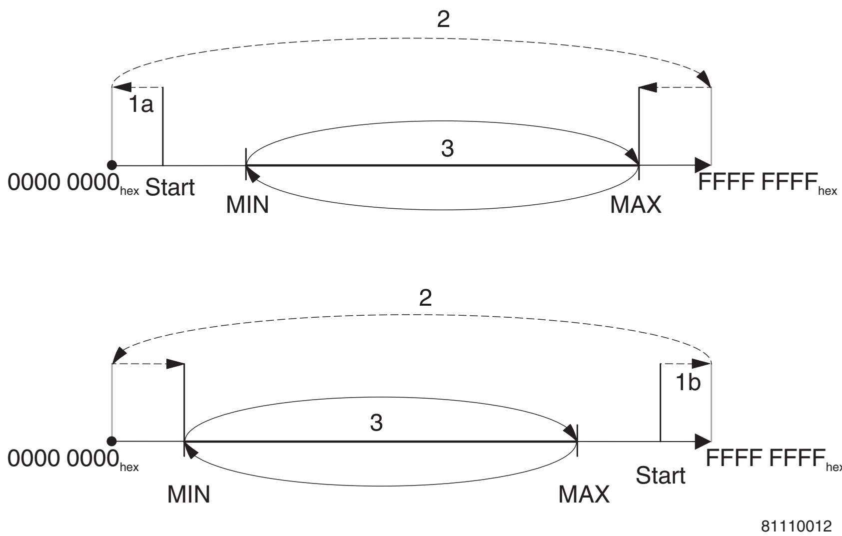

Figure 2-3 Counting with a start value outside the lower and upper limits and reaching the maximum limits

MIN	Lower limit
MAX	Upper limit
Start	Start value
1a	Start value below the lower limit; negative counting direction
1b	Start value above the upper limit; negative counting direction
2	Counting until the opposite maximum limit is reached
3	Counting between the lower and upper limits

2.4.3 Counting control

You can control the counting using process data bits, the gate input and the direction and stop input.

Bits in the process data

The *StopCNTx*, *StartCNTx* and *DirCNTx* control bits are available for controlling the counting by the controller.

The bit for stopping the counter is dominant in the *StopCNTx* and *StartCNTx* control bits. If the counting is stopped, only the counter is stopped. In this case, the current counter status is not cleared. The *ResetCNTx* reset bit must be set, if the counter status is to be reset for the current parameterization. See also Section “OUT process data” on page 23.

Gate input

If you use the gate input, the module waits for a valid gate condition after a successful parameterization. The counter is started after the gate condition is fulfilled. See also Section “Functions of the gate inputs (G1, G2)” on page 13.

Direction/stop input

If you use the direction/stop input, you can change the counting direction or stop the counting process depending on the parameterization. See also Section “Functions of the direction/stop inputs (Dir1, Dir2)” on page 14.

2.4.4 Functions of the gate inputs (G1, G2)

You can use the gate input to control the counting of the source input signals by an external signal.

The counting functions with regard to the limits described above are maintained.

You can parameterize the gate input with the engineering tool or object 0080_{hex} as follows:

Table 2-2 Gate functions

Name	Meaning
Off	The gate input has no function. In this case you can use the gate input as a normal digital input. The status is mapped to the IN process data word 0 in bit 2 *GateCNT2* or bit 10 *GateCNT1.* In this case the counter can be controlled via the process data (*StartCNTx* and *StopCNTx* bits).
Counting at high level	The counting process is started with a high level at the gate input and stopped at a low level. The current counter status is maintained.
Counting at low level	The counting process is started with a low level at the gate input and stopped at a high level. The current counter status is maintained.
Start on rising edge	The counting process is started with the first detected rising edge of the gate signal. It cannot be stopped by the gate signal. If the counter is stopped either by the controller or the stop input, it will be restarted on the next rising edge.
Start on falling edge	The counting process is started with the first detected falling edge of the gate signal. It cannot be stopped by the gate signal. If the counter is stopped either by the controller or the stop input, it will be restarted on the next falling edge.
Counting at high level and resetting the counter on a rising edge	The counting process is started on a rising edge and stopped on the next falling edge. The counter is reset on the next rising edge of the gate signal. Depending on the current counting direction, the upper (counting downwards) or the lower (counting upwards) limit is loaded into the counter as the counter status.
Counting at low level and resetting the counter on a falling edge	The counting process is started on a falling edge and stopped on the next rising edge. The counter is reset on the next falling edge of the gate signal. Depending on the current counting direction, the upper (counting downwards) or the lower (counting upwards) limit is loaded into the counter as the counter status.

If you control the counting process via the gate inputs, you can no longer control it via the process data (*StartCNTx* and *StopCNTx* bits).

2.4.5 Functions of the direction/stop inputs (Dir1, Dir2)

Depending on the parameterization you can use the direction/stop input to change the direction of counting or to stop the counting process.

You can parameterize the input with the engineering tool or with object 0080_{hex} as follows:

Table 2-3 Functions of the direction/stop input

Name	Description
Off	The direction/stop input has no function. In this case you can use the direction/stop input as a normal digital input. The status is mapped to the IN process data word 0 in bit 1 *DirCNT2* or bit 9 *DirCNT1.* In this case the counter can be controlled via the process data (*StartCNTx* and *StopCNTx* bits).
Changing the direction on a rising edge (starting direction is upwards)	Counting starts with the positive counting direction. The counting direction is changed on a rising edge of the signal.
Changing the direction on a rising edge (starting direction is downwards)	Counting starts with the negative counting direction. The counting direction is changed on a rising edge of the signal.
Changing the direction on a falling edge (starting direction is upwards)	Counting starts with the positive counting direction. The counting direction is changed on a falling edge of the signal.
Changing the direction on a falling edge (starting direction is downwards)	Counting starts with the negative counting direction. The counting direction is changed on a falling edge of the signal.
Counting upwards at high level	When the direction signal is on a high level the counter counts upwards, and when it is on a low level it counts downwards.
Counting upwards at low level	When the direction signal is on a low level the counter counts upwards, and when it is on a high level it counts downwards.
Stop on rising edge	The counter is stopped on the rising edge of the input signal.
Stop on falling edge	The counter is stopped on the falling edge of the input signal.

If you control the counting direction via the direction/stop inputs, you can no longer change it via the process data (*DirCNTx* bit).

If an input has been parameterized as a stop input, then the counter can no longer be stopped via the process data (*StopCNTx* bit).

2.5 Functions of the incremental encoder channels

The two incremental encoder channels (INC) are used to evaluate signals of symmetrical or asymmetrical incremental encoders with a maximum input frequency of 300 kHz (one channel wired) or 100 kHz (more than one channel wired).

Each of the channels has a 32-bit counter to detect a position.

Each channel has its own digital input for referencing (Ref1, Ref2).

An external signal can be used to store signal values specifically. For this, each INC channel has a latch input (L1, L2).

You can use the latch inputs (Lx) and the reference inputs (Refx) as digital inputs, since the status of these inputs is mapped to the process data, no matter whether they are used as latch or reference inputs or not.

2.5.1 Position detection

Specify the following parameters when your parameterize the channels (engineering tool or object 0080_{hex}):

- How the two inputs signals A and B are scanned (*BasicConfig* variable*, evaluation*)
  - Double: the positive edges of A and B are detected
  - Quadruple: both edges of both signals are detected

  In this way you can vary the accuracy and the maximum path that can be detected.
- Type of axis (*BasicConfig* variable*, axis type*)
  - Rotary axis
    With rotary axes positioning takes place between zero and the modulo value that can be parameterized (object, *ModuloVal* variable). In a positive direction of rotation the position is set to zero after the modulo value-1, and in negative direction of rotation after the value 0 to the modulo value-1. There is no position value overflow in this mode. The modulo value can be a position value from $0000\ 0001_{hex}$ to $FFFF\ FFFF_{hex}$.
  - Linear axis
    With linear axes positioning takes place without limits and the position value increments or decrements between $0000\ 0000_{hex}$ and the maximum value $FFFF\ FFFF_{hex}$. An error is reported to the controller if there is a position value overflow (see Section “Exceeding the position value with linear axes” on page 16).
- Conversion factor (*ConvFact* variable*)*

  This factor can be used to specify values which are parameterized or returned in relation to a measuring unit, for example.
  Relate all default values to this conversion factor.
- Two limit values (*1LimitValOut*, *2LimitValOut* variables)

  An output is controlled according to the parameterization when the limit value is reached (see Section “Output control” on page 22).
  For rotary axes both limit values must not be larger than the modulo value.

The direction of travel or motion is determined by the phase shift of the two signals A and B, and the position is incremented or decremented accordingly. The direction of travel or motion cannot be manipulated by the controller.

2.5.2 Exceeding the position value with linear axes

After axis referencing has been completed, a warning is signaled to the controller when the minimum position (0) is fallen below or the maximum position (2^{32}-1) is exceeded (8910_{hex}, 8920_{hex}; see Section “Diagnostic state (0018_{hex}: DiagState)” on page 32).

The position is still being detected and the *ErrorINCx* bit of the respective channel is set. Should the direction of travel be changed after an underflow and the position value again exceeds the underflow limit, the error bit will be reset and a removed error is reported. The same is true for an overflow.

As long as the axis has not been referenced, there will be no indication that the value has fallen below the minimum position or exceeded the maximum position.

2.5.3 Encoder monitoring

Monitoring by firmware

Monitoring of the encoder is only possible if the following conditions are met:

– Linear axis
– Encoder with Z signal
– Only **one** channel is operated, i.e. no further INC channel and no CNT channel.
– The maximum input frequency is 100 kHz.

During the encoder monitoring the module checks the correct encoder functioning using the Z signal. It is also checked whether the detected encoder increment value between two Z pulses corresponds to the default value or whether a Z occurs again after the parameterized encoder increment value.

For the encoder monitoring you have to enter the encoder increment value (number of increments per revolution).

The monitoring of the encoder increment value has a tolerance of ±3 pulses. This tolerance is caused by software runtimes.

An error is indicated by:

– The red “Encoder error (03/07)” LED of the channel is ON.
– The red “I/O error (E)” LED of the module is ON.
– The *EncSurvINCx* bit of the channel is set.
– Diagnostic message to the controller (7305_{hex}; see Section “Diagnostic state (0018_{hex}: DiagState)” on page 32).

As soon as the Z signal error is no longer detected, the LEDs as well as the process data bit are reset and a removed error is reported.

Monitoring by hardware

The encoders are usually monitored by hardware. The hardware monitors the encoder supply, detects open circuits, and faulty electrical signals with symmetrical encoders.

An error is indicated by:

– The red “Encoder error (03/07)” LED of the channel is ON.
– The red “I/O error (E)” LED of the module is ON.
– The *EncSurvINCx* bit of the channel is set.
– Diagnostic message to the controller (8600_{hex}; see Section “Diagnostic state (0018_{hex}: DiagState)” on page 32).

2.5.4 Referencing

In order to have a defined reference value for position detection, it is possible to set a reference point. The parameterized reference value is entered as a position value in the IN process data at this point. The value range for the reference point is 0000 0000_{hex} to FFFF $FFFF_{hex}$. For each channel there is one input for connecting a reference switch (Ref1, Ref2) available in order to determine the reference point.

You can set the reference point in the engineering tool or in object 0080_{hex} as follows:

Table 2-4 Referencing functions

Function	Meaning	Example
Direct setting of the reference point	In this mode, the reference point is set immediately in the process data by the corresponding bit (*StateRefINx*). The only condition for the direct setting is that the drive is at rest at the reference position.	
Referencing to the rising edge of the reference signal in the positive direction of motion	The reference value is taken over as position value, when a rising edge has been detected at the reference input and the direction of motion is positive.	Figure 2-4
Referencing to the falling edge of the reference signal in the positive direction of motion	The reference value is taken over as position value, when a falling edge has been detected at the reference input and the direction of motion is positive.	
Referencing to the rising edge of the reference signal in the negative direction of motion	The reference value is taken over as position value, when a rising edge has been detected at the reference input and the direction of motion is negative.	
Referencing to the falling edge of the reference signal in the negative direction of motion	The reference value is taken over as position value, when a falling edge has been detected at the reference input and the direction of motion is negative.	
Referencing to the rising edge of the reference signal in the positive direction of motion with Z	The reference point is set on the rising edge of the Z signal, after the rising edge of the reference point signal has been detected and the direction of motion is positive. You can specify an offset for detecting a Z signal (see below).	Figure 2-5 Figure 2-6
Referencing to the falling edge of the reference signal in the positive direction of motion with Z	The reference point is set on the rising edge of the Z signal, after the falling edge of the reference point signal has been detected and the direction of motion is positive. You can specify an offset for detecting a Z signal (see below).	
Referencing to the rising edge of the reference signal in the negative direction of motion with Z	The reference point is set on the rising edge of the Z signal, after the rising edge of the reference point signal has been detected and the direction of motion is negative. You can specify an offset for detecting a Z signal (see below).	

Table 2-4 Referencing functions

Function	Meaning	Example
Referencing to the falling edge of the reference signal in the negative direction of motion with Z	The reference point is set on the rising edge of the Z signal, after the falling edge of the reference point signal has been detected and the direction of motion is negative. You can specify an offset for detecting a Z signal (see below).	
Direct referencing on Z	In this mode the reference point is set directly on a rising edge of the Z signal. **Do not change the direction of rotation or motion during the reference run! It is recommended that you do not stop the axis during the reference run.** You can specify a gear ratio for rotary axes with object 0080_{hex}, *TransmFact* variable, and determine to which Z signal the referencing is to take place after the start (see Figure 2-7 on page 20).	Figure 2-7

> You can use input Ref1/Ref2 as normal digital input if you have selected another referencing function than “Direct setting of the reference point”. The status is mapped to IN process data word 1 in bit 6 *StateRefIN2* or bit 14 *StateRefIN1*.

Offset

When you use referencing to Z, you can specify an offset for detecting the Z signal using the engineering tool or object 0080_{hex}, variable *OffsetZRef.*

This offset indicates after which number of increments the Z signal detection is enabled. The current distance between reference signal and Z is transferred with the process data.

> The distance between the selected reference edge and the Z signal is shown as long as the *EnableRefx* bit is set.

Referencing

> If a certain direction of motion is specified in the function for setting the reference point, no reference point will be set with an opposite direction of motion even if the reference signal occurs.

Key for the following diagrams:

Ref	Reference signal
P	Positive direction of motion
Z	Z signals
R	Reference point

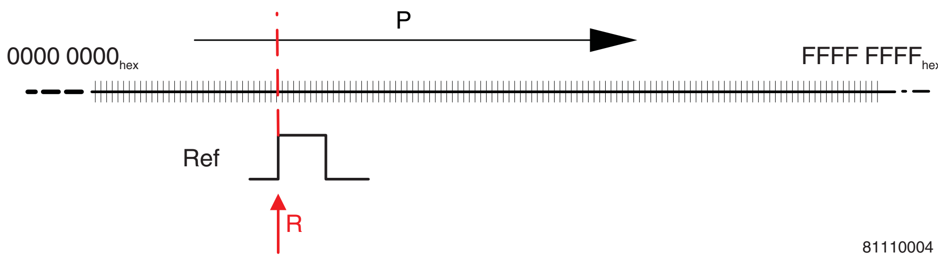

Figure 2-4 Referencing to the rising edge in positive direction

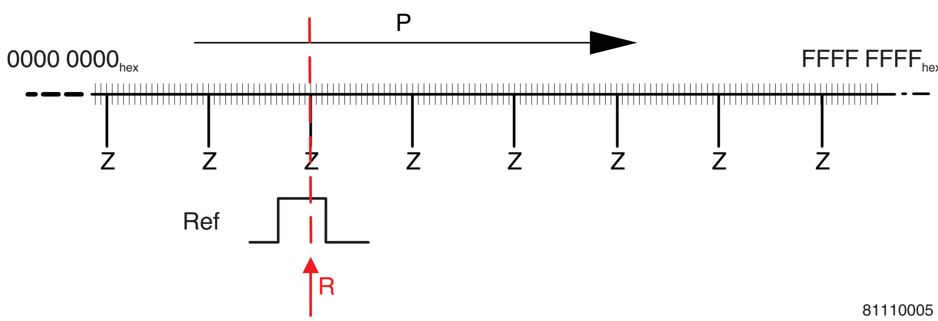

Figure 2-5 Referencing to the rising edge of the reference signal in positive direction of motion with Z

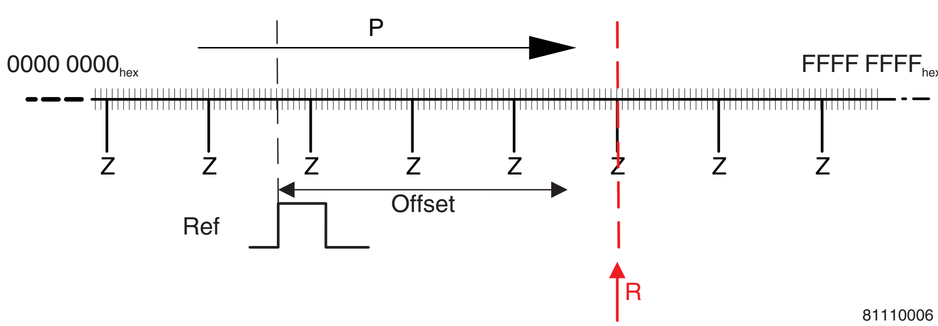

Figure 2-6 Referencing to the rising edge of the reference signal in positive direction with Z - with offset

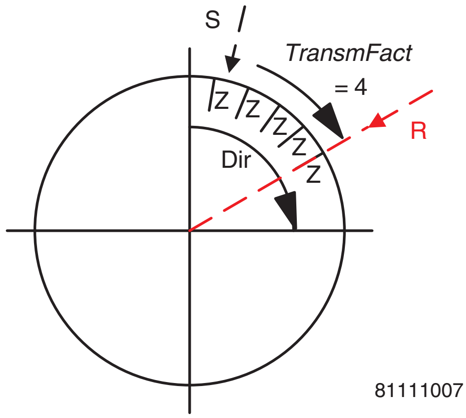

Figure 2-7 Referencing to Z with gear ratio

Key for Figure 2-7:

Dir	Direction of rotation
Z	Z signals
S	Starting point
TransmFact	Parameterized gear ratio
R	Reference point

When you parameterize the channel, parameterize the reference method and the parameter values with the engineering tool or object 0080_{hex}. Start referencing using the process data. Referencing can be started again at any time.

2.5.5 Latch function

The latch function temporarily stores the current position value by a signal at the latch input (L1, L2). The current latch value is transmitted to the controller.

You can influence the latch functions with the engineering tool or object 0080_{hex} as follows:

Table 2-5 Latch functions

Name	Meaning
Off	The latch functions are deactivated. In this case you can use the latch input as a normal digital input. The status is mapped to the IN process data word 1 in bit 3 *LatchINC2* or bit 11 *LatchINC1*.
Latch on rising edge	The current position value is saved on the rising edge of the latch signal.
Latch on falling edge	The current position value is saved on the falling edge of the latch signal.
Latch on both edges	The current position value is saved on the rising edge as well as on the falling edge of the latch signal.

In the process data, only the last latch value is shown regardless of the edge. All latch values can be read with the PDI object “Latch values INC” (see Section “Latch values of INC object (0091_{hex}: LatchValuesINCIN)” on page 45).

2.6 Output control

A common output is available for one CNT channel and one INC channel each. CNT1 and INC1 as well as CNT2 and INC2 share one output.

When you assign an output to both channels simultaneously, the parameterization will not be accepted nor transferred.

When you parameterize the channel, parameterize the output with the engineering tool or object 0080_{hex} and the *OutputConfig* variable. Set the output mode and the limits for the trigger. The following options are available:

Limits for triggering:

– Trigger to lower limit (1st limit)
– Trigger to upper limit (2nd limit)
– Trigger to both limits

Output mode

The output can be controlled as follows:

Table 2-6 Output mode

Value	Meaning
Off	The output is not used by the corresponding channel.
Output set to high (start state is low)	The output is set when the limit value is reached and remains high.
Output set to low (start state is high)	The output is reset when the limit value is reached and remains low.
Toggling with start state low	The output is toggled each time a limit is reached. The output is low when counting starts.
Toggling with start state high	The output is toggled each time a limit is reached. The output is high when counting starts.
High pulse with a length that can be parameterized	A high pulse is generated when a limit is reached. The pulse length can be parameterized. The length can be set between 1 ms and 1 s in steps of 1 ms.
Low pulse with a length that can be parameterized	A low pulse is generated when a limit is reached. The pulse length can be parameterized. The length can be set between 1 ms and 1 s in steps of 1 ms.
Controlling the output directly by the master	The output is set or reset depending on the corresponding process data bit.

The type of how the output is controlled applies to both limits. It is not possible to parameterize a separate response of the output for each limit.

The outputs are switched off (outputs set to low) when the logic fails (e.g., no voltage) or an application error occurs (e.g., failure of the higher-level bus system).

3 Data transmission via process data

The controller sends control bits to the module via the process data and receives the status bits as well as counter and latch values.

3.1 OUT process data

The OUT process data words of the controller occupy 14 words. The parameterization is specified with the OUT process data words. The words as structured as follows:

Word 0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	SetNew Limits 1	Set OUT1	Reset CNT1	Dir CNT1	Stop CNT1	Start CNT1	Res.	Res.	SetNew Limits 2	Set OUT2	Reset CNT2	Dir CNT2	Stop CNT2	Start CNT2

Word 1															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Enable Ref INC1	Set Ref INC1	Res.	Res.	Res.	Res.	Res.	Res.	Enable Ref INC2	Set Ref INC2

Word 2
High word lower limit CNT1
Word 3
Low word lower limit CNT1
Word 4
High word upper limit CNT1
Word 5
Low word upper limit CNT1
Word 6
High word lower limit CNT2
Word 7
Low word lower limit CNT2
Word 8
High word upper limit CNT2
Word 9
Low word upper limit CNT2
Word 10 ... 13
Reserved

Table 3-1 Meaning of the bits in output word 0

Bit for channel		Designation	Meaning	
1	2			
8	0	StartCNTx	= 1:	CNT channel x is started (only if bit 9/1 = 0; stop bit is dominant) The bit is only relevant if no gate function is parameterized: Object 0080_{hex}, BasicConfig, Gate mode = 0000_{bin}
9	1	StopCNTx	= 1:	CNT channel x is stopped This bit is only relevant should the gate input or the stop input be parameterized in such a way that the counter is not stopped by these functions: Object 0080_{hex}, BasicConfig Gate mode = 0000_{bin}, 0011_{bin} or 0100_{bin} Direction/stop input mode = 0000_{bin} ... 0100_{bin}
10	2	DirCNTx	 = 1: = 0:	Counting direction of CNT channel x Up Down This bit is only relevant if no specific direction has been specified for the channel: Object 0080_{hex}, BasicConfig, Direction/stop input mode = 0000_{bin}, 0101_{bin} or 0110_{bin}
11	3	ResetCNTs	= 1:	The counting value of the channel is reset and the starting value is loaded again
12	4	SetOUTx	= 1: = 0:	Output x is set (high) Output x is reset (low) You can manually set or reset the output only if the output is parameterized accordingly: Object 0080_{hex}, OutputConfig, Output mode = 0111_{bin}
13	5	SetNewLimitsx	0->1:	The limits specified in the process data for channel x are transferred
14, 15	6, 7	Res.		Reserved Set reserved bits to 0.

Table 3-2 Meaning of the bits in output word 1

Bit for channel		Designation	Meaning	
1	2			
8	0	SetRefx	0->1:	The reference point of channel x is set immediately You can set the reference point only if the type of referencing is parameterized accordingly: Object 0080_{hex}, BasicConfig, Type of referencing = 0000_{bin}
9	1	EnableRefx	0->1:	The parameterized referencing of channel x is started Object 0080_{hex}, BasicConfig, Type of referencing > 0000_{bin}
10 ... 15	2 ... 7	Res.		Reserved Set reserved bits to 0.

The limits of the two CNT channels can be changed online with the words 2 to 9.

With a positive edge of the *SetNewLimitsx* bit, the module tries to transfer the limit values. The following conditions must be met:

- The lower limit is smaller than the upper limit.
- The current counter value is within the new limits.

If the limit values are valid, they will be transferred and are valid after they have been transferred. The corresponding *LimitConfCNTx* bit is set in the IN process data. This bit will be shown as long as the *SetNewLimitsx* is set, at least for one cycle.

The *ErrorCNTx* bit is set if the limits are not valid, and the new limits will not be transferred. The counting process will continue with the current limits. The *ErrorCNTx* bit is set as long as the *SetNewLimitsx* bit is set (if only this error has occurred).

3.2 IN process data

The IN process data words occupy 14 words in the controller. Input process data contains general states of the module, states of the channels, and the corresponding counter and latch values. The words are structured as follows:

Word 0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Error CNT1	Res.	Limit Conf CNT1	OUT1	Dir/ Stop Input CNT1	Gate CNT1	Dir CNT1	Run/ Stop CNT1	Error CNT2	Res.	Limit Conf CNT2	OUT2	Dir/ Stop Input CNT2	Gate CNT2	Dir CNT2	Run/ Stop CNT2

Word 1															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Error INC1	State Ref IN1	Ref Compl INC1	Run Ref1	Latch INC1	Enc Surv INC1	Dir INC1	Halt INC1	Error INC2	State Ref IN2	Ref Compl INC2	Run Ref2	Latch INC2	Enc Surv INC2	Dir INC2	Halt INC2

Word 2
High word counter status CNT1
Word 3
Low word counter status CNT1
Word 4
High word counter status CNT2
Word 5
Low word counter status CNT2
Word 6
High word position INC1
Word 7
Low word position INC1
Word 8

High word latch value INC1
Word 9
Low word latch value INC1
Word 10
High word position INC2
Word 11
Low word position INC2
Word 12
High word latch value INC2
Word 13
Low word latch value INC2

Table 3-3 Meaning of the bits in the input word 0

Bit for channel		**Designation**	**Meaning**	
1	**2**			
8	0	Run/StopCNTx	= 1: = 0:	Counter of channel x running Counter of channel x not running
9	1	DirCNTx	 = 1: = 0:	Current counting direction of the CNT x Up Down
10	2	GateCNTx	 = 1: = 0:	Status of the gate input x High Low
11	3	Dir/StopInputCNTx	 = 1: = 0:	Current status of the Dir/Stop input x High Low
12	4	OUTx	 = 1: = 0:	Current status of output x High Low
13	5	LimitConfCNTx	= 1:	New CNT limit values of channel x have been transferred successfully. The bit is set as long as the corresponding *SetNewLimitsx* bit is set, at least for one transmission cycle.
14	6	Res.		Reserved Set reserved bits to 0.
15	7	ErrorCNTx	= 1: = 0:	An error has occurred on channel x. If the bit is not set when the limits are reparameterized, the process data of the channel is invalid. No error

Table 3-4 Meaning of the bits in input word 1

Bit for channel		Designation	Meaning
1	2		
8	0	Run/StopINCx	Status of the encoder = 1: Encoder is rotating = 0: Encoder is not rotating
9	1	DirINCx	Direction of rotation of the encoder (counting direction) = 1: Up = 0: Down
10	2	EncSurvINCx	Status of encoder monitoring via firmware at INC x = 1: Error = 0: No error The bit is only valid when the encoder monitoring of the channel was enabled: Object 0080_{hex}, BasicConfig, Encoder monitoring = 1_{bin}
11	3	LatchINCx	Status of latch input x = 1: High = 0: Low
12	4	RunRefINCx	Status of referencing = 1: Referencing in progress = 0: Referencing not in progress The bit is only valid when referencing was started; in normal operation the bit is = 0.
13	5	RefComplINCx	Acknowledgment of referencing = 1: Referencing completed = 0: Referencing not completed The bit is only valid when referencing was started; in normal operation the bit is = 0.
14	6	StateRefINx	Status of reference input Refx; = 1: Reference input Refx set = 0: Reference input Refx not set
15	7	ErrorINCx	= 1: An error has occurred on channel x. The process data is no longer valid. = 0: No error

The words 2 to 13 show the current counter, position, and latch values.

The position and latch values are transmitted with reference to the gear ratio as 32-bit integer values. Possible decimal places are not taken into consideration.

For the referencing type “Referencing to the xth Z after a reference edge”, the position values of the INC channels show the distance of the Z signal to the selected edge of the reference input, as long as the *EnableRefx* bit is set.

4 Parameter, diagnostics and information

PDI = Parameter, Diagnostics and Information

Parameter and diagnostic data as well as other information is transmitted via the PDI channel.

For information on PDI, please refer to the UM EN AXL F SYS INST user manual.

The standard objects and manufacturer-specific application objects stored in the module are described in the following sections.

The following applies to all tables below:

Table 4-1 Explanation of object codes and data types

Data type	Meaning
Var	Single, simple variable
Array	String of simple variables of the same data type
Record	String of simple variables of different data types or of the same data type with different lengths
Visible string	Byte string with ASCII characters that can be printed only, terminated with 00_{hex}
Octet string	Byte string with any contents
Unsigned 8	Value without sign, only positive values from 00_{hex} ... FF_{hex}
Unsigned 16	Value without sign, only positive values from 0000_{hex} ... $FFFF_{hex}$
Unsigned 32	Value without sign, only positive values from $0000\ 0000_{hex}$ ... $FFFF\ FFFF_{hex}$

Table 4-2 Abbreviations in the table headers

Abbrevia-tion	Meaning
N	Number of elements
L	Length of an element in bytes
R	Read
W	Write

4.1 Standard objects

Please refer to the basic profile for comprehensive information.

Every visible string is terminated with a zero terminator (00_{hex}). The length of a visible string element is therefore one byte larger than the amount of user data.

4.1.1 Objects for identification (device rating plate)

Table 4-3 Objects for identification

Index [hex]	Object name	Object code	Data type	N	L	Rights	Meaning	Contents
Manufacturer								
0001	VendorName	Var	Visible string	1	16	R	Vendor name	Phoenix Contact
0002	VendorID	Var	Visible string	1	7	R	Vendor ID	00A045
0003	VendorText	Var	Visible string	1	49	R	Vendor text	Components and systems for industrial automation
0012	VendorURL	Var	Visible string	1	30	R	Vendor URL	http://www.phoenixcontact.com
Module - general								
0004	DeviceFamily	Var	Visible string	1	20	R	Device range	I/O function module
0006	Product-Family	Var	Visible string	1	33	R	Product range	AXL F CNT2 INC2 1F: Axioline - High speed I/O System
					9			AXL F CNT2 INC2 XC 1F: AXL F XC
000E	CommProfile	Var	Visible string	1	4	R	Communication profile	633
000F	DeviceProfile	Var	Visible string	1	5	R	Device profile	0010
0011	Profile-Version	Record	Visible string	2	11; 20	R	Profile version	2011-12-07; Basis - Profil V2.0
003A	Version-Count	Array	Unsigned 16	4	2	R	Version counter	0005 0000 0000 0000
Module - special								
0005	Capabilities	Array	Visible string	1	8	R	Properties	Nothing
0007	Product-Name	Var	Visible string	1	19	R	Product name	AXL F CNT2 INC2 1F
					22			AXL F CNT2 INC2 XC 1F
0008	Serial-Number	Var	Visible string	1	11	R	Serial number	E.g., 5555555555

Table 4-3 Objects for identification

Index [hex]	Object name	Object code	Data type	N	L	Rights	Meaning	Contents
0009	ProductText	Var	Visible string	1	47	R	Product text	2 counter inputs, 2 incremental encoder inputs
000A	Order-Number	Var	Visible string	1	8	R	Order number	AXL F CNT2 INC2 1F: 2688093
								AXL F CNT2 INC2 XC 1F: 2701239
000B	Hardware-Version	Record	Visible string	2	11; 3	R	Hardware version	E.g., 2010-06-21; 01
000C	Firmware-Version	Record	Visible string	2	11; 17	R	Firmware version	E.g., 0000-00-00, FW version V1.00
000D	PCH Version	Record	Visible string	2		R	Parameter channel version	E.g., 2010-01-08; V1.00
0037	DeviceType	Array	Octet string	1	8	R	Module identifica-tion	00 08 00 1C 00 00 00 F1
Use of the device								
0014	Location	Var	Visible string	1	59	R/W	Installation location	Can be filled out by the user.
0015	Equipmen-tIdent	Var	Visible string	1	59	R/W	Equipment identifier	Can be filled out by the user.
0016	Appl-DeviceAddr	Var	Unsigned 16	1	2	R/W	Application device address	Can be filled out by the user.

4.1.2 Object for multilingual capacity

Tabelle 4-4 Objekte zur Diagnose

Index [hex]	Object name	Object type	Data type	A	L	Rights	Meaning	Contents
0017	Language	Record	Visible String	2	6; 8	R	Language	en-us; English

4.1.3 Object descriptions

Table 4-5 Object descriptions

Index [hex]	Object name	Object code	Data type	N	L	Rights	Meaning
003B	PDIN_Descr	Record		12	48	R	Description of the IN process data structure; This object is only important for tools and is therefore not described in more detail here.
003C	PDOUT_Descr	Record		12	48	R	Description of the OUT process data structure; This object is only important for tools and is therefore not described in more detail here.

4.1.4 Objects for diagnostics

Table 4-6 Objects for diagnostics

Index [hex]	Object name	Object code	Data type	N	L	Rights	Meaning/contents
0018	DiagState	Record		6	22	R	Diagnostic state; see page 32
0019	ResetDiag	Var	Unsigned 8	1	1	W	Reset diagnostics; see page 33

4.1.4.1 Diagnostic state (0018$_{hex}$: DiagState)

This object is used for a structured message of an error.

You can access this object only via subindex 0 and read the complete object.

Table 4-7 Diagnostic state

0018$_{hex}$: DiagState (Read)					
Subindex	**Data type**	**Length in bytes**	**Meaning**	**Contents**	
0	Record	22	Diagnostic state	Complete diagnostics information	
1	Unsigned 16	2	Error number	0 ... 65535$_{dec}$ Consecutive error number since the last reset or the error memory reset	
2	Unsigned 8	1	Priority	00$_{hex}$	No error
				01$_{hex}$	Error still pending
				02$_{hex}$	Warning still pending
				81$_{hex}$	Error eliminated
				82$_{hex}$	Warning eliminated
3	Unsigned 8	1	Channel	00$_{hex}$	No error
				01$_{hex}$	Error of channel CNT1
				02$_{hex}$	Error of channel CNT2
				03$_{hex}$	Error of channel INC1
				04$_{hex}$	Error of channel INC2
				FF$_{hex}$	Error not related to a specific channel
4	Unsigned 16	2	Error code	See table below	
5	Unsigned 8	1	More follows	00 (not supported)	
6	Visible string	15	Text (14 characters)	See table below	

Error code and related text:

Table 4-8 Error code and related text for diagnostic messages with “Error” priority

Error	Error code	Text	Remark	Priority	Channel
No error	0000$_{hex}$		Status OK	01$_{hex}$	FF$_{hex}$
Short circuit or overload of the 24 V supply	5112$_{hex}$	24V Supply No x	24 V supply no. x (x = 1 or 2)	01$_{hex}$	FF$_{hex}$
Short circuit or overload of the 5 V supply	5113$_{hex}$	5V Supply No x	5 V supply no. x (x = 1 or 2)	01$_{hex}$	FF$_{hex}$
Overload or short circuit at the output	3300$_{hex}$	Output x	Output x (x = 1 or 2)	01$_{hex}$	FF$_{hex}$
Encoder error	7305$_{hex}$	Malfunction of encoder	Z signal error of the encoder Line loss during an encoder rotation	01$_{hex}$	03$_{hex}$ 04$_{hex}$

Table 4-8 Error code and related text for diagnostic messages with “Error” priority

Error	Error code	Text	Remark	Priority	Chan-nel
Input error of the incremental encoder	8600_{hex}	Encoder input signal error	One of the following errors has occurred: – Faulty input signal at the encoder input – Short circuit – No encoder connected	01_{hex}	03_{hex} 04_{hex}
Counter overflow for the INC channels	8910_{hex}	Counter overflow	Counter overflow at the counter of the corresponding INC channel	02_{hex}	03_{hex} 04_{hex}
Counter underflow for the INC channels	8920_{hex}	Counter underflow	Counter underflow at the counter of the corresponding INC channel	02_{hex}	03_{hex} 04_{hex}
No valid parameterization found in the module’s memory	6300_{hex}	ParameterSet not ok	See “6300_{hex}: No valid parameterization found in the module’s memory”	01_{hex}	FF_{hex}

6300_{hex}: No valid parameterization found in the module’s memory

This error indicates that no valid parameterization has been found.

This error occurs during a restart of the module when the parameterization is to be read from the memory and no valid parameterization has been found.

You can reset the error with object 0019_{hex}. You can parameterize the channels with object 0080_{hex}. The module functions until the next voltage reset occurs. However, it cannot be guaranteed that the parameterization is stored. We recommend to replace the module.

Resetting the diagnostic counter and the diagnostic status

You can delete the diagnostic counter and the diagnostic status with object 0019_{hex}.

4.1.4.2 Reset diagnostics (0019_{hex}: ResetDiag)

You can delete the diagnostics memory and acknowledge the diagnostic messages with this object.

Table 4-9 Reset diagnostics

0019_{hex}: ResetDiag (Write)					
Subindex	**Data type**	**Length in bytes**	**Meaning**	**Contents**	
0	Unsigned 8	1	Reset diagnostics	00_{hex}	Permits diagnostic messages
				02_{hex}	Deletes and acknowledges all pending diagnostic messages that have not been read out
				05_{hex}	Deletes and acknowledges the current diagnostic message
				06_{hex}	Deletes and acknowledges the complete diagnostics and does not permit any new diagnostic messages
				Other	Reserved

4.1.5 Objects for process data management

Table 4-10 Objects for process data management

Index [hex]	Object name	Object code	Data type	N	L	Rights	Assignment
0025	PDIN	Var	Octet string	1	28	R	IN process data: xxxx...; See Section 4.1.5.1
0026	PDOUT	Var	Octet string	1	28	R/W	OUT process data: xxxx...; See Section 4.1.5.2
0027	GetExRight	Var	Unsigned 8	1	1	R/W	
001D	Password	Var	Octet string		40, max.	W	Password; 5474902010

4.1.5.1 IN process data (0025$_{hex}$: PDIN)

You can read the IN process data of the module with this object. The structure corresponds to the representation in Section “IN process data” on page 25.

Table 4-11 IN process data

0025$_{hex}$: PDIN (Read)			
Subindex	**Data type**	**Length in bytes**	**Meaning**
0	Octet string	28	IN process data

4.1.5.2 OUT process data (0026$_{hex}$: PDOUT)

You can read and write the OUT process data of the module with this object. The structure corresponds to the representation in Section “OUT process data” on page 23.

Table 4-12 OUT process data

0026$_{hex}$: PDOUT (Read/Write)			
Subindex	**Data type**	**Length in bytes**	**Meaning**
0	Octet string	28	OUT process data

4.1.5.3 Get exclusive rights (0027$_{hex}$: GetExRight)

You can request exclusive rights for writing the OUT process data via the PDI channel with this object. Following a positive confirmation, the data is no longer updated via the process data channel, but is only available via the PDI channel. The OUT process data is changed with the OUT process data object (see also Section "Controlling the process data via the PDI channel" on page 36).

NOTE: Disregarding the following information may result in malfunction

– The exclusive rights are reset each time a connection is aborted or the bus is reset.
– This action may have serious consequences for the connected process. This is why this object is password-protected.

Table 4-13 Get exclusive rights

0027$_{hex}$: GetExRight (Read, write)					
Subindex	**Data type**	**Length in bytes**	**Meaning**	**Contents**	
0	Unsigned 8	1	Get exclusive rights	00$_{hex}$	Transmit OUT process data via the process data channel
				01$_{hex}$	Transmit OUT process data via the PDI channel
				Other	Reserved

4.1.5.4 Password (001D$_{hex}$: Password)

You can transmit a password with this object. For example, you can enable the writing of OUT process data via the PDI channel (see also Section "Controlling the process data via the PDI channel" on page 36).

The password for this module is "5474902010".

Table 4-14 Password

001D$_{hex}$: Password (Write)			
Subindex	**Data type**	**Length in bytes**	**Meaning**
0	Octet string	40, max.	Password

4.1.6 Controlling the process data via the PDI channel

You can always read IN process data with the *Read IN process data* (0025_{hex}) object. The data structure corresponds to the process data structure (see Section “Data transmission via process data” on page 23).

You can always read OUT process data with the *Read OUT process data* (0026_{hex}) object. The data structure corresponds to the process data structure (see Section “Data transmission via process data” on page 23).

To overwrite the OUT process data of the module via the PDI channel, you have to enable this option first. To do this, proceed as follows:

- Enter the correct password (“5474902010”) into the *Password* ($001D_{hex}$) object.
- You can now enable the process data via the PDI channel. To do this, write the value 01_{hex} to the *Get exclusive rights* (0027_{hex}) object.

Then you can access the OUT process data. Process data which is sent via the process data channel to the module is no longer taken into consideration.

You can disable write access to the process data by writing the value 00_{hex} to the *Get exclusive rights* object. Process data via the process data channel becomes valid again.

> Before you can access the *Get exclusive rights* (0027_{hex}) object, you have to enter the valid password in the *Password* ($001D_{hex}$) object, even if the write access is returned via PDI. This prevents unintentional writing to the *Get exclusive rights* (0027_{hex}) object.
>
> The password to enable the *Get exclusive rights* (0027_{hex}) object is “5474902010”.

4.1.7 Objects for device management

Table 4-15 Objects for device management

Index [hex]	Object name	Object code	Data type	N	L	Rights	Assignment
002D	ResetParam	Var	Unsigned 8	1	1	W	Reset parameterization

4.1.7.1 Reset parameterization (002D$_{hex}$: ResetParam)

You can reset various parameterizations using this object.

Table 4-16 Reset parameterization

002D$_{hex}$: ResetParam (Write)					
Subindex	**Data type**	**Length in bytes**	**Meaning**	**Contents**	
0	Unsigned 8	1	Reset parameterization	00$_{hex}$	No action
				01$_{hex}$	Module is reset to the default settings
				Other	Reserved

4.2 Application objects

These objects can be used to specify device-specific parameters or to read device-specific data.

Table 4-17 Application objects

Index [hex]	Object name	Object code	Data type	N	L	Rights	Assignment
0080	ParamTable	Record		4	44	R/W	Parameter table; see page 38
0090	LimitValuesCNT	Array	Unsigned 32	10	4	R	CNT limit values; see page 40
0091	LatchValuesINCIN	Array	Unsigned 32	6	4	R	INC latch values; see page 45

In the hardware configurator (e.g., PC Worx, Step 7) the parameters can be selected via a pull-down menu. If you do not use a hardware configurator, parameterize the module over PDI objects.

4.2.1 Parameter table (0080_{hex}: ParaTable)

Parameterize the module using this object.

Parameterize the complete module when you access via subindex 0.
Parameterize a single channel when you access via the subindexes 1 to 4.

The written parameter record is stored in the module and reloaded from the memory when the module is powered up.

The current states and counter values are reset with the parameterization and the current channel configuration is reloaded. This is also true when only a single channel is parameterized via the subindexes.

NOTE: Changes made via process data are overwritten during parameterization.
If you have made changes via process data (e.g., new limits), these changes are no longer available after parameterization.

Table 4-18 Parameter table

0080_{hex}: ParamTable (Read, write)				
Subindex	**Data type**	**Length in bytes**	**Meaning**	**Contents**
0	Record	176	Parameterization of the entire module	See subindexes
1	Record	44	Parameterization of CNT1	See Table "Parameterization of CNTx" on page 39
2	Record	44	Parameterization of CNT2	
3	Record	44	Parameterization of INC1	See Table "Parameterization of INCx" on page 41
4	Record	44	Parameterization of INC2	

Set all reserved bits to 0 in the parameter table.

Table 4-19 Parameterization of CNTx

Variable	Data type	Bit	Meaning	Values		Default
Basic Config	Unsigned 16	0	Type of counting	1_{bin}	Periodic	1_{bin}
				0_{bin}	One-time	
		1 ... 3	Reserved	000_{bin}		000_{bin}
		4 ... 7	Gate mode	0000_{bin}	Gate off	0000_{bin}
				0001_{bin}	Counting at high level	
				0010_{bin}	Counting at low level	
				0011_{bin}	Start on rising edge	
				0100_{bin}	Start on falling edge	
				0101_{bin}	Counting at high level and resetting the counter value on a rising edge	
				0110_{bin}	Counting at low level and resetting the counter value on a falling edge	
				Other	Reserved	
		8 ... 11	Direction/stop input mode	0000_{bin}	No function	0000_{bin}
				0001_{bin}	Changing the direction on a rising edge (starting direction is upwards)	
				0010_{bin}	Changing the direction on a rising edge (starting direction is downwards)	
				0011_{bin}	Changing the direction on a falling edge (starting direction is upwards)	
				0100_{bin}	Changing the direction on a falling edge (starting direction is downwards)	
				0101_{bin}	Counting up at high level	
				0110_{bin}	Counting up at low level	
				0111_{bin}	Stop on rising edge	
				1000_{bin}	Stop on falling edge	
				Other	Reserved	
		12 ... 15	Reserved			0000_{bin}
Reserved	Unsigned 16	0 ... 15	Reserved			0000_{hex}

Table 4-19 Parameterization of CNTx

Variable	Data type	Bit	Meaning	Values		Default
Output Config	Unsigned 16	0 ... 3	Output mode (see also Section "Output mode" on page 22)	0000_{bin}	Output deactivated for this channel	0000_{bin}
				0001_{bin}	High (start state is low)	
				0010_{bin}	Low (start state is high)	
				0011_{bin}	Toggle (start state is low)	
				0100_{bin}	Toggle (start state is high)	
				0101_{bin}	High pulse	
				0110_{bin}	Low pulse	
				0111_{bin}	Output is set by the controller	
		4 ... 5	When reaching the corresponding limit, the output is set (see also Section "Output control" on page 22)	00_{bin}	Output not used	00_{bin}
				01_{bin}	Lower limit	
				10_{bin}	Upper limit	
				11_{bin}	Both limits	
		6 ... 15	Reserved			0000000000_{bin}
Output PulseTime	Unsigned 16		Pulse duration at the output for the high or low output mode in ms	1_{dec} ... 1000_{dec}		1_{dec}
StartVal	Unsigned 32		Start value for the channel	0_{dec} ... 4294967295_{dec}		0_{dec}
Lower Limit	Unsigned 32		Lower limit for the channel	0_{dec} ... 4294967295_{dec}		0_{dec}
Upper Limit	Unsigned 32		Upper limit for the channel	0_{dec} ... 4294967295_{dec}		4294967295_{dec}
Reserved	Unsigned 32		Reserved	0_{dec}		
Reserved	Unsigned 32		Reserved	0_{dec}		
Reserved	Unsigned 32		Reserved	0_{dec}		
Reserved	Unsigned 32		Reserved	0_{dec}		
Reserved	Unsigned 32		Reserved	0_{dec}		
Reserved	Unsigned 32		Reserved	0_{dec}		

Table 4-20 Parameterization of INCx

Variable	Data type	Bit	Meaning	Values		Default
Basic Config	Unsigned 16	0	Activate channel	0_{bin}	Channel inactive	0_{bin}
				1_{bin}	Channel active Activate the channel only if an encoder is connected, otherwise you will get the error message 8600_{hex} (see Table 4-8 on page 32).	
		1	Axis type	0_{bin}	Linear	0_{bin}
				1_{bin}	Round	
		2	Encoder type	0_{bin}	Symmetrical	0_{bin}
				1_{bin}	Asymmetrical	
		3	Reserved			0_{bin}
		4	Encoder with Z signal	0_{bin}	Without Z signal	0_{bin}
				1_{bin}	With Z signal	
		5	Encoder monitoring	0_{bin}	Off	0_{bin}
				1_{bin}	On	
		6 ... 7	Latch input mode	00_{bin}	Off	00_{bin}
				01_{bin}	Positive edge	
				10_{bin}	Negative edge	
				11_{bin}	Both edges	
		8 ... 9	Evaluation	00_{bin}	Reserved	11_{bin}
				01_{bin}	Reserved	
				10_{bin}	Double	
				11_{bin}	Quadruple	

Table 4-20 Parameterization of INCx

Variable	Data type	Bit	Meaning	Values		Default
		12 ... 15	Type of referencing (see also Section "Referencing" on page 17)	0000_{bin}	Direct setting of the reference point by the controller	0000_{bin}
				0001_{bin}	Referencing to the rising edge of the Ref signal in positive direction	
				0010_{bin}	Referencing to the rising edge of the Ref signal in negative direction	
				0011_{bin}	Referencing to the falling edge of the Ref signal in positive direction	
				0100_{bin}	Referencing to the falling edge of the Ref signal in negative direction	
				0101_{bin}	Referencing to the rising edge of the Ref signal in positive direction with Z	
				0110_{bin}	Referencing to the rising edge of the Ref signal in negative direction with Z	
				0111_{bin}	Referencing to the falling edge of the Ref signal in positive direction with Z	
				1000_{bin}	Referencing to the falling edge of the Ref signal in negative direction with Z	
				1001_{bin}	Referencing to Z	
				Other	Reserved	
Reserved	Unsigned 16	0 ... 15	Reserved			0000_{hex}
Output Config	Unsigned 16	0 ... 3	Output mode (see also Section "Output mode" on page 22)	0000_{bin}	Output deactivated for this channel	0000_{bin}
				0001_{bin}	High (start state is low)	
				0010_{bin}	Low (start state is high)	
				0011_{bin}	Toggle (start state is low)	
				0100_{bin}	Toggle (start state is high)	
				0101_{bin}	High pulse	
				0110_{bin}	Low pulse	
				0111_{bin}	Output is set by the controller	
		4 ... 5	When reaching the corresponding limit, the output is set (see also Section "Output control" on page 22)	00_{bin}	Output not used	00_{bin}
				01_{bin}	Lower limit	
				10_{bin}	Upper limit	
				11_{bin}	Both limits	
		6 ... 15	Reserved			0000000000_{bin}

Table 4-20 Parameterization of INCx

Variable	Data type	Bit	Meaning	Values	Default
Output-PulseTime	Unsigned 16		Pulse duration at the output for the high or low output mode in ms	1_{dec} ... 1003_{dec}	1_{dec}
ConvFact	Unsigned 32	0 ... 15	Gear ratio denominator (see also Section "Gear ratio" on page 50)	1_{dec} ... 65535_{dec}	1_{dec}
		16 ... 31	Gear ratio numerator (see also Section "Gear ratio" on page 50)	1_{dec} ... 65535_{dec}	1_{dec}
ModuloVal	Unsigned 32		Modulo value for rotary axes	1_{dec} ... 4294967295_{dec}	1_{dec}
EncIncVal	Unsigned 32		Encoder increment value	1_{dec} ...1073741823_{dec}	1_{dec}
1Limit ValOut	Unsigned 32		1st limit value for switching the output For rotary axes make sure: 1LimitValOut ≤ ModuloVal	0_{dec} ... 4294967295_{dec}	0_{dec}
2Limit ValOut	Unsigned 32		2nd limit value for switching the output For rotary axes make sure: 2LimitValOut ≤ ModuloVal	0_{dec} ... 4294967295_{dec}	4294967295_{dec}
Reference Val	Unsigned 32		Reference value	0_{dec} ... 4294967295_{dec}	0_{dec}
OffsetZRef	Unsigned 32		Offset for detecting a Z signal (see also Section "Offset" on page 18)	0_{dec} ... 4294967295_{dec}	0_{dec}
Transm Fact	Unsigned 32		Gear ratio for rotary axes	1_{dec} ... 4294967295_{dec}	1_{dec}
Res	Unsigned 32		Reserved	0_{dec}	0_{dec}

4.2.2 CNT limit values (0090_{hex}: LimitValuesCNT)

The limit values of the CNT channels can be read with this object. You can read both the current limits (may have changed via process data) as well as the limits stored in the parameterization (see object 0080_{hex}).

Read all limit values when you access via subindex 0.
Read the selected value when you access via subindexes 1 to 10.

Table 4-21 CNT limit values

0090_{hex}: LimitValuesCNT (Read)			
Subindex	**Data type**	**Length in bytes**	**Meaning/contents**
0	Array of Unsigned 32	40	Read the limit values of CNT channels
1	Unsigned 32	4	Current lower limit of CNT1
2	Unsigned 32	4	Current upper limit of CNT1
3	Unsigned 32	4	Lower limit of CNT1 parameterized with PDI
4	Unsigned 32	4	Upper limit of CNT1 parameterized with PDI
5	Unsigned 32	4	Reserved
6	Unsigned 32	4	Current lower limit of CNT2
7	Unsigned 32	4	Current upper limit of CNT2
8	Unsigned 32	4	Lower limit of CNT2 parameterized with PDI
9	Unsigned 32	4	Upper limit of CNT2 parameterized with PDI
10	Unsigned 32	4	Reserved

4.2.3 Latch values of INC object (0091_{hex}: LatchValuesINCIN)

You can read the individual latch values of the INC channels with this object.

Read all latch values when you access via subindex 0.
Read the selected value when you access via subindexes 1 to 6.

Table 4-22 INC latch values

0091_{hex}: LatchValuesINCIN (Read)			
Subindex	**Data type**	**Length in bytes**	**Meaning/contents**
0	Array of Unsigned 32	24	Read all latch values of the INC channels
1	Unsigned32	4	Latch value on rising edge of the INC1 channel
2	Unsigned 32	4	Latch value on falling edge of the INC1 channel
3	Unsigned32	4	Reserved
4	Unsigned 32	4	Latch value on rising edge of the INC2 channel
5	Unsigned32	4	Latch value on falling edge of the INC2 channel
6	Unsigned32	4	Reserved

The values are transmitted with reference to the gear ratio. The values are transmitted as 32-bit integer values. Decimal places are not considered.

5 Example: Counting goods

Example description

In the following example, sets of 1000 screws are to be packed in a cardboard box.

The control input (2) is set when a flap of the funnel is opened. The rising edge of this control input activates the counter input automatically.

The sensor (3) detects and counts the screws via the counter input. Each screw that falls out of the funnel into the box initiates a pulse at the counter input. Once 1000 screws have passed the sensor (3), the switching output (1) is set for a period of 1 s.

The flap can be closed by setting the switching output (1).

Now a new box can be placed under the funnel and the process can be repeated.

Example structure

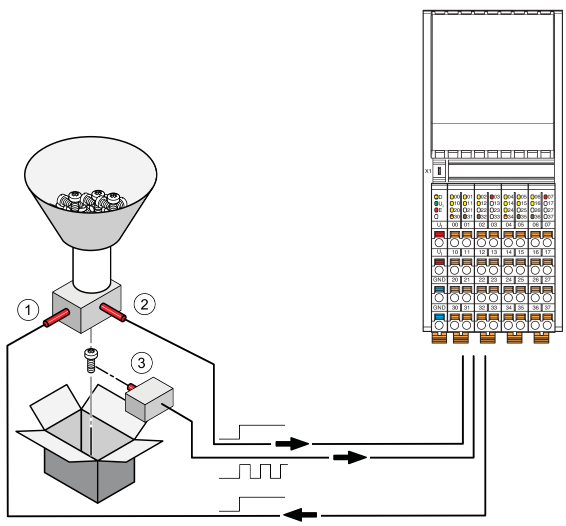

Figure 5-1 Example system for counting goods

Table 5-1 Key for Figure 5-1

No.	Sensor/actuator	Associated input/output	Terminal point when using counter 1
1	Valve (flap control)	Output 1 (OUT 1)	30 (Out1)
2	Sensor for valve position	Control input 1 (Gate1)	10 (G1)
3	Sensor (for detecting the screws, e.g. photoelectric barrier)	Counter input 1 (Source1)	00 (S1)

Parameterization of the module

Only counter channel 1 (CNT1) is used for this example. Parameterization of the other channels can be left unchanged. The parameterization can either be done with a corresponding hardware configurator or with a PDI write access to object 0080_{hex} subindex 1_{dec} during operation.

> Object 0080_{hex} must be completely written to when a subindex cannot be addressed via the higher-level fieldbus system. All channels are reparameterized.

Parameterize the CNT channel 1 (CNT1) for this example as follows:

Table 5-2 Parameterization for the example

Parameters	Value	Explanation
BasicConfig -> Type of counting	1_{bin} = periodic count	Periodic counting is parameterized since it is a recurring counting process.
BasicConfig -> Gate mode	0101_{bin} = Counting at high level and reset of the counter status on a rising edge	Counting is started on the rising edge of the control input (2) (flap open) and stopped on the falling edge (flap closed). Any number of boxes can be filled since a new counting process is started each time the flap is opened.
BasicConfig-> Direction/stop input mode	0000_{bin} = no function	
OutputConfig-> Output mode	0101_{bin} = high pulse	When the limit is reached, a positive pulse is to be generated at the output.
OutputConfig-> Switching behavior of the output	10_{bin} = switching with upper limit	
OutputPulseTime	1000_{dec}	1000 ms pulse length at the output
Start value (StartVal)	0_{dec}	Counting starts with 0 screws.
Lower limit	0_{dec}	Counting starts with 0 screws.
Upper limit	1000_{dec}	The upper counting limit is 1000 screws.

In the example, the counting direction is controlled by process data. Set the *DirCNT1* direction bit in the OUT process data constantly to 1 (positive counting direction).

The counting starts as soon as the flap opens and a 1 (set) is detected at control input (2).

The counting stops as soon as the flap closes and a 0 (not set) is detected at control input (2).

Reading the counter states

The current counter state, the counting direction and the status of the switching output (1) and the control input (2) are mapped to the IN process data.

Words 2 and 3	= Current counter state
OUT1 bit	= Switching output (1) state
GateCNT1 bit	= Control input (2) state
DirCNT1 bit	= Current counting direction
Run/StopCNT1 bit	= Counter state

Addition

If the number of screws per box is to be changed during the current counting process, the counting limit (upper limit) can be changed during runtime via the process data.

Please make sure at the time of the modification that the current counter state is not higher than the new counting limit. Otherwise, an error message will be generated.

To change the maximum number of screws, proceed as follows:

- Write the new upper limit into the OUT process data words 4 and 5 (high word upper limit CNT1 and low word upper limit CNT1).
  The OUT process data words 2 and 3 (high word lower limit CNT1 and low word lower limit CNT1) are not changed (0_{dec}).
- The new limits are taken over on a rising edge of the *SetNewLimitsCNT1* bit in the OUT process data. The state must be maintained for this bit until the take-over is indicated by the *LimitConfCNT1* bit in the IN process data. The new limits are valid when this bit is set.

> The limits set via process data are transient, that means after a restart the parameters set in the hardware configuration or transferred via PDI are again valid.

A Appendix: Axis types and gear ratios

A 1 Axis types

This module can be used to detect the position with linear or rotary axes. The desired axis type must be defined during parameterization. Depending on the axis type, there are differences during parameterization and operation.

Linear axis

A linear axis is an axis with a limited path.

The maximum path is the permitted position range of 0 to $2^{32} - 1$ increments.

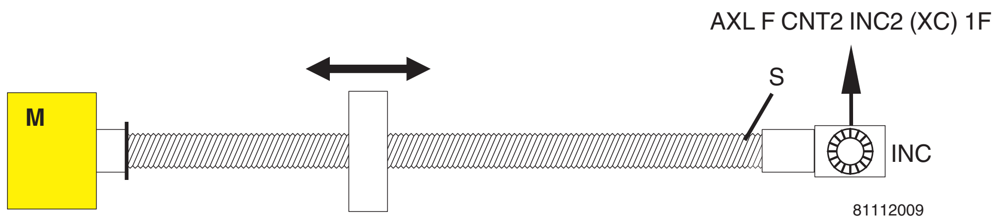

Figure A-1 Example of a linear axis

M	Motor
S	Spindle
AXL F CNT2 INC2 (XC) 1F	Connection to the AXL F CNT2 INC2 1F / AXL F CNT2 INC2 XC 1F module
INC	Incremental encoder

Rotary axis

A rotary axis is an axis with an unlimited path (rotary table, continuous conveyor belt). For a rotary axis, the path starts and ends at the same physical point on the axis. Hardware and software limit switches are not considered.

For rotary axes, the path (increments) for a revolution is referred to as a modulo value. You specify the modulo value with the engineering tool or object 0080_{hex}, variable *ModuloVal*.

If rotary axis mode is active, the drive must be freely rotatable. If this is not possible, the axis will not meet the requirement for operation as a rotary axis and must, therefore, be parameterized as a linear axis.

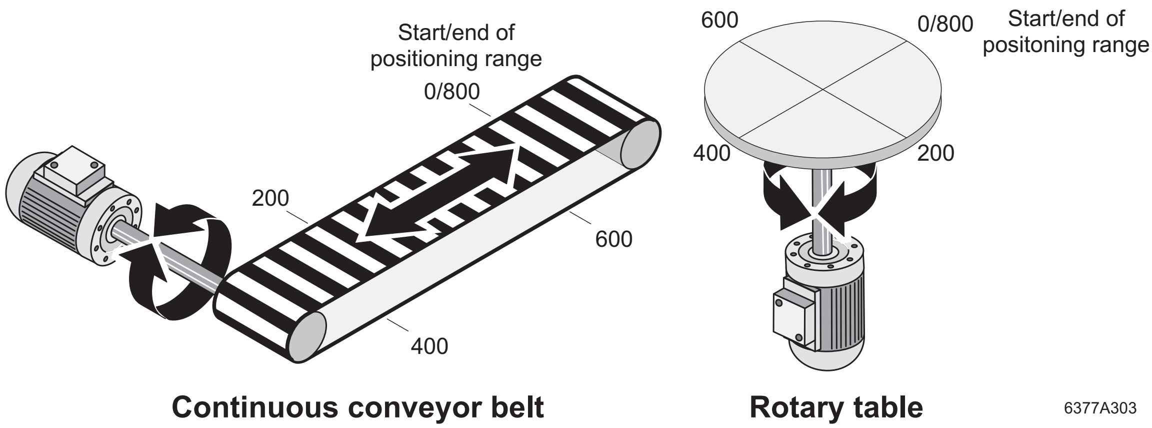

Figure A-2 Examples of rotary axes

In Figure A-2 the modulo value is specified as 800. This value has the same position on the axis as the value 0. The actual value range is indicated by the module as 0 to 799. The value 800 is not indicated.

A 2 Gear ratio

You can use object 0080_{hex}, variable *ConfFact* to specify the gear ratio.

The gear ratio determines how many units of a measurement unit correspond to a certain number of increments.

The gear ratio can be used for linear and rotary axes.

Example

You want to specify the position in centimeters rather than increments. You know that the entire path is 200 centimeters long and that this represents 4000 increments.

Therefore:

Gear ratio numerator	GRN = 200 (cm)	(1 to 1023)
Gear ratio denominator	GRD = 4000	(1 to 2^{16} - 1)

GRN/GRD = 200/4000 = 1/20 (cm)

All other values must now be specified in centimeters:
- Limit values
- Reference point
- Offset

The module will also indicate all values in centimeters.

B Appendix: Revision history

Tabelle 7-1 Revision history

Revision	Date	Contents	
01	2011-02-08	First publication	
02	2013-06-20	Hardware verison 03 or later, Firmware version 1.20 or later	
03	2014-09-04	Addition: AXL F CNT2 INC2 XC 1F	
		Section 1.1	Addition: Data for AXL F CNT2 INC2 XC 1F
		Table 3-3	Change: Order of the objects Addition: Data for AXL F CNT2 INC2 XC 1F
		Section 3.1.2	New; Removed from table 3-3
		Section 3.1.4.2	Correction: read -> read and write
		Section 4	Reading the counter states Correction: control output (1) -> switching output (1)
04	2015-11-25	Renaming AXL CNT 2/INC 2 -> AXL F CNT2 INC2 1F	
05	2019-05-15	Table 4-18	Correction of max values for StartVal, LowerLimit, UpperLimit
		Table 4-20	Correction of max values for ConvFact, ModuloVal, EncIncVal, 1LimitValOut, 2LimitValOut, ReferenceVal, OffsetZRef, TransmFact

Please observe the following notes

General terms and conditions of use for technical documentation

Phoenix Contact reserves the right to alter, correct, and/or improve the technical documentation and the products described in the technical documentation at its own discretion and without giving prior notice, insofar as this is reasonable for the user. The same applies to any technical changes that serve the purpose of technical progress.

The receipt of technical documentation (in particular user documentation) does not constitute any further duty on the part of Phoenix Contact to furnish information on modifications to products and/or technical documentation. You are responsible to verify the suitability and intended use of the products in your specific application, in particular with regard to observing the applicable standards and regulations. All information made available in the technical data is supplied without any accompanying guarantee, whether expressly mentioned, implied or tacitly assumed.

In general, the provisions of the current standard Terms and Conditions of Phoenix Contact apply exclusively, in particular as concerns any warranty liability.

Phoenix Contact reserves the right to register its own intellectual property rights for the product identifications of Phoenix Contact products that are used here. Registration of such intellectual property rights by third parties is prohibited.

Other product identifications may be afforded legal protection, even where they may not be indicated as such.

How to contact us

Internet

Up-to-date information on Phoenix Contact products and our Terms and Conditions can be found on the Internet at:
phoenixcontact.com

Make sure you always use the latest documentation.
It can be downloaded at:
phoenixcontact.net/products

Subsidiaries

If there are any problems that cannot be solved using the documentation, please contact your Phoenix Contact subsidiary.
Subsidiary contact information is available at phoenixcontact.com.

Published by

PHOENIX CONTACT GmbH & Co. KG
Flachsmarktstraße 8
32825 Blomberg
GERMANY

PHOENIX CONTACT Development and Manufacturing, Inc.
586 Fulling Mill Road
Middletown, PA 17057
USA

Should you have any suggestions or recommendations for improvement of the contents and layout of our manuals, please send your comments to:
tecdoc@phoenixcontact.com