X20(c)AP31xx

Data sheet
2.70 (June 2025)

Translation of the original documentation

Publishing information

B&R Industrial Automation GmbH

B&R Strasse 1

5142 Eggelsberg

Austria

Telephone: +43 7748 6586-0

Fax: +43 7748 6586-26

office@br-automation.com

Disclaimer

1339366519268-2.70

1 General information

1.1 Other applicable documents

For additional and supplementary information, see the following documents.

Other applicable documents

Document name	Title
MAX20	X20 System user's manual

1.2 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

1.3 Order data

Order number	Short description	Figure
	Analog input modules	
X20AP3111	X20 energy metering module, 3 analog inputs, 480 VAC, 50/60 Hz, 4 analog inputs, 20 mA AC, calculates effective, reactive and apparent power/energy, calculates RMS values, 240 V keyed, NetTime function	
X20AP3121	X20 energy metering module, 3 analog inputs, 480 VAC, 50/60 Hz, 4 analog inputs, 1 A AC, calculates effective, reactive and apparent power/energy, calculates RMS values, 240 V keyed, NetTime function	
X20AP3131	X20 energy metering module, 3 analog inputs, 480 VAC, 50/60 Hz, 4 analog inputs, 5 A AC, calculates effective, reactive and apparent power/energy, calculates RMS values, 240 V keyed, NetTime function	
X20AP3161	X20 energy metering module, 3 analog inputs, 480 VAC, 50/60 Hz, 4 analog inputs, 333 mV AC, calculates effective, reactive and apparent power/energy, calculates RMS values, 240 V keyed, NetTime function	
X20AP3171	X20 energy metering module, 3 analog inputs, 480 VAC, 50/60 Hz, 4 analog inputs, Rogowski adjustable (µV/A), max. 52 mV, calculates effective, reactive and apparent power/energy, calculates RMS values, 240 V keyed, NetTime function	
X20AP3122	X20 energy metering module, 3 analog inputs, 480 VAC, 50/60 Hz, 4 analog inputs, 1 A AC, groundable, calculates effective, reactive and apparent power/energy, calculates RMS values, 240 V keyed, NetTime function	
X20AP3132	X20 energy metering module, 3 analog inputs, 480 VAC, 50/60 Hz, 4 analog inputs, 5 A AC, groundable, calculates effective, reactive and apparent power/energy, calculates RMS values, 240 V keyed, NetTime function	
X20cAP3121	X20 energy metering module, coated, 3 analog inputs, 480 VAC, 50/60 Hz, 4 analog inputs, 1 A AC, calculates effective, reactive and apparent power/energy, calculates RMS values, 240 V keyed, NetTime function	
X20cAP3131	X20 energy metering module, coated, 3 analog inputs, 480 VAC, 50/60 Hz, 4 analog inputs, 5 A AC, calculates effective, reactive and apparent power/energy, calculates RMS values, 240 V keyed, NetTime function	
	Required accessories	
	Bus modules	
X20BM32	X20 bus module, for double-width modules, 240 VAC keyed, internal I/O power supply connected through	
X20cBM32	X20 bus module, coated, for double-width modules, 240 VAC keyed, internal I/O power supply connected through	
	Terminal blocks	
X20TB32	X20 terminal block, 12-pin, 240 VAC keyed	

Table 1: X20AP3111, X20AP3121, X20AP3131, X20AP3161, X20AP3171, X20AP3122, X20AP3132, X20cAP3121, X20cAP3131 - Order data

1.4 Module description

The modules are used for power monitoring and energy management of a machine. Examples of application areas:

- Multiphase energy measurement for class 0.5S or class 1 for
  - 3-phase, 4-wire applications with neutral conductor (with/without grounding)
  - 3-phase, 3-wire applications (with/without grounding)
  - 2-phase networks with grounded phase B connection
- Single-phase measurement by disabling unnecessary inputs
- Network analysis according to harmonic content
- Signal trace via 8 kHz recording of the 3 voltage and 4 current channels with FIFO buffer

Functions

- Energy management
- Measurement function
- Fourier analysis
- Monitoring functions
- NetTime Technology
- Flatstream communication

Energy management

The modules' integrated functions not only map the machine's current power requirements in detail, they also serve to record the consumption of the machine or system. All relevant data is available to the user in the process image.

Measuring functions

The modules provide the following options for processing measured values:

- Temperature coefficient of the internal reference of 6 ppm/°C
- Energy registers for active, reactive and apparent energy, separate for forward and reverse, fundamental wave and harmonics
- Threshold register for status signal generation and enabling power and energy measurement
- Recording the THD harmonic component
- Discrete Fourier transform (DFT) up to the 31st harmonic component per phase for voltage and current

Monitoring functions

The module monitors numerous parameters and provides status signals for e.g. voltage dip, power failure, phase sequence, energy flow reversal and neutral conductor current monitoring.

Fourier analysis

The module calculates the harmonic component up to the 31st harmonic for voltage and current as well as the total harmonic distortion of each phase.

NetTime timestamp of the measurement

Not only is the measured value important for many applications, but also the exact time when the measurement takes place. The module is equipped with a NetTime timestamp function for this that supplies a timestamp for the recorded measurement with microsecond accuracy.

Flatstream communication

"Flatstream" was designed for X2X and POWERLINK networks and allows data transfer to be adapted to individual demands. This allows data to be transferred more efficiently than with standard cyclic polling.

2 Safety guidelines

General

Information:

If the maximum voltage value of 655 V is displayed, it is necessary to check whether the input measurement range has been exceeded.

Caution!

All current inputs must be double-insulated or reinforced-insulated.

Attention !

Toutes les entrées de courant doivent avoir une double isolation ou une isolation renforcée.

X20AP31x2

X20AP31x2 modules with current transformers that are groundable meet the requirement from the median voltage guideline that states that a current transformer connection must always be grounded in systems that exceed a certain system voltage.

Caution!

Only the connection marked "ILxb" is permitted to be grounded.

Attention !

Seule la connexion marquée « ILxb » peut être mise à la terre.

Caution!

The use of modules with current transformers that groundable is only permitted in systems with grounded median voltage. Operating these modules without connected grounding is not permitted.

Attention !

L'utilisation de modules avec des transformateurs de courant pouvant être mis à la terre n'est autorisée que dans les systèmes avec tension médiane mise à la terre. Il n'est pas autorisé de faire fonctionner ces modules sans connexion à la terre.

Information:

Because these modules do not feature a neutral conductor connection, the ground potential on the current transformer connections forms the central reference point. (See "Input circuit diagram" on page 16.)

Danger!

In consideration of a possible fault, such as insulation failure, additional insulating measures must be taken in the module beyond the basic insulation between the voltage and current inputs.
To avoid electric shock, the wiring to the module must ensure adequate insulation. The dielectric strength of the cable insulation MUST be designed for the level of the phase voltage.

Danger !

En cas de problème éventuel, tel qu'un défaut d'isolation, des mesures d'isolation supplémentaires doivent être prises dans le module en plus de l'isolation de base entre les entrées de tension et de courant.
Pour éviter les électrocutions, le câblage du module doit être suffisamment isolé. La résistance diélectrique de l'isolation du câble DOIT être conçue pour le niveau de la tension de phase.

3 Technical description

3.1 Technical data

3.1.1 X20AP3111, X20(c)AP3121 and X20(c)AP3131

Order number	X20AP3111	X20AP3121	X20cAP3121	X20AP3131	X20cAP3131
Short description					
I/O module	3-phase power and energy metering module for current transformers				
General information					
B&R ID code	0xC9DA	0xC9DB	0xE214	0xC9DC	0xEB55
Status indicators	I/O function per channel, operating state, module status				
Diagnostics					
Module run/error	Yes, using LED status indicator and software				
Inputs	Yes, using LED status indicator and software				
Power consumption					
Bus	0.85 W (Rev. <D0) 0.50 W (Rev. =D0) 0.45 W (Rev. >D0)		0.85 W (Rev. <C0) 0.50 W (Rev. =C0) 0.45 W (Rev. >C0)	0.85 W (Rev. <E0) 0.50 W (Rev. =E0) 0.45 W (Rev. >E0)	
Internal I/O	-				
Additional module power dissipation [W]	40 mW [1]	2 W [1]			
Insulation voltages					
Inputs - Bus / I/O power supply	Tested at 5500 VDC, 1 min				
Inputs - Ground	Tested at 5500 VDC, 1 min				
Bus / I/O power supply - Ground	Tested at 510 VAC, 1 min				
Certifications					
CE	Yes				
UKCA	Yes				
ATEX	Zone 2, II 3G Ex nA nC IIA T5 Gc IP20, Ta (see X20 user's manual) FTZÚ 09 ATEX 0083X				
UL	cULus E115267 Industrial control equipment				
HazLoc	cCSAus 244665 Process control equipment for hazardous locations Class I, Division 2, Groups ABCD, T5		-	cCSAus 244665 Process control equipment for hazardous locations Class I, Division 2, Groups ABCD, T5	
DNV	Temperature: **B** (0 to 55°C) Humidity: **B** (up to 100%) Vibration: **B** (4 g) EMC: **B** (bridge and open deck)				
LR	ENV1				
KR	Yes				
ABS	Yes				
BV	**EC33B** Temperature: 5 - 55°C Vibration: 4 g EMC: Bridge and open deck				
EAC	Yes		-	Yes	-
Voltage inputs					
Number of phases	3				
Input impedance	1.68 MΩ				
Nominal voltage U_N					
Between phases	Max. 480 VAC [2]				
Phase to N	Max. 277 VAC				
Max. display value	655 VAC				
Resolution	10 mV, with voltage connected directly				
Rated frequency	50 and 60 Hz				
Measurable frequency					
Measurement range	45 to 65 Hz				
Resolution	0.01 Hz				
Current inputs					
Quantity	4 AC inputs				
Nominal current I_N					
Secondary	20 mA	1 A		5 A	
Primary	65 A directly configurable, larger values through conversion in the application [3]				
Max. overload current	20 x I_N for 0.5 s	8 x I_N for 0.5 s			
Max. measurement current	20 mA	1 A		5 A	

Table 2: X20AP3111, X20AP3121, X20cAP3121, X20AP3131, X20cAP3131 - Technical data

Order number	X20AP3111	X20AP3121	X20cAP3121	X20AP3131	X20cAP3131
Resolution	1 mA, based on the primary current [3]				
Load	25 Ω	500 mΩ		20 mΩ	
Measurement accuracy [4]					
U_{RMS}	±0.65% [5]				
I_{RMS}	±0.65% [6]	±0.65% [7]		±1.65% (Rev. <E0) ±0.70% (Rev. ≥E0) [8]	±0.70% [8]
Effective, reactive and apparent power	±0.80% [9]	±0.80% [10]		±1.80% (Rev. <E0) ±0.85% (Rev. ≥E0) [11]	±0.85% [11]
Frequency, power factor and phase angle	±0.50% [12]				
Active energy per phase and total [13]	±0.40% [9]	±0.40% [10]		±1.40% (Rev. <E0) ±0.45% (Rev. ≥E0) [11]	±0.45% [11]
Active energy of fundamental frequency per phase and total [13]	±0.50% [9]	±0.50% [10]		±1.50% (Rev. <E0) ±0.55% (Rev. ≥E0) [11]	±0.55% [11]
Active energy of harmonics per phase and total [13]	±0.80% [9]	±0.80% [10]		±1.80% (Rev. <E0) ±0.85% (Rev. ≥E0) [11]	±0.85% [11]
Reactive energy per phase and total [14]	±0.50% [9]	±0.50% [10]		±1.50% (Rev. <E0) ±0.55% (Rev. ≥E0) [11]	±0.55% [11]
Apparent energy					
Per phase and arithmetic total	±0.50% [9]	±0.50% [10]		±1.50% (Rev. <E0) ±0.55% (Rev. ≥E0) [11]	±0.55% [11]
Vector sum	±0.80% [9]	±0.80% [10]		±1.80% (Rev. <E0) ±0.85% (Rev. ≥E0) [11]	±0.85% [11]
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	Not permitted				
Degree of protection per EN 60529	IP20				
Ambient conditions					
Temperature					
Operation					
Horizontal mounting orientation	-25 to 60°C				
Vertical mounting orientation	-25 to 50°C				
Derating	-	See section "Derating".			
Storage	-40 to 85°C				
Transport	-40 to 85°C				
Relative humidity					
Operation	5 to 95%, non-condensing		Up to 100%, condensing	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 X20TB32 separately. Order 1x bus module X20BM32 separately.		Order 1x terminal block X20TB32 separately. Order 1x bus module X20cBM32 separately.	Order 1x terminal block X20TB32 separately. Order 1x bus module X20BM32 separately.	Order 1x terminal block X20TB32 separately. Order 1x bus module X20cBM32 separately.
Pitch	$25^{+0.2}$ mm				

Table 2: X20AP3111, X20AP3121, X20cAP3121, X20AP3131, X20cAP3131 - Technical data

1) Power dissipation of current measurement shunts
2) The design of the module allows 480 VAC to be applied to the terminal block.
3) For measuring higher current values, see section "Current transformer - Pinout".
4) Based on the current measured value.
   The actual error value percentage may be larger due to the digital display.
5) With drift of 25 ppm/K
6) With drift of 50 ppm/K
7) With drift of 35 ppm/K
8) With drift of 225 ppm/K (Rev. < E0) or 100 ppm/K (Rev. ≥ E0)
9) With drift of 75 ppm/K
10) With drift of 60 ppm/K
11) With drift of 250 ppm/K (Rev. < E0) or 125 ppm/K (Rev. ≥ E0)
12) In power systems with approximately sinusoidal voltage starting at 10 VAC.
13) At power factor cos ϕ = 1, 0.5L and 0.8C
14) At reactive power factor sin ϕ = 1, 0.5L and 0.8C

3.1.2 X20AP3122 and X20AP3132

Order number	X20AP3122	X20AP3132
Short description		
I/O module	3-phase power and energy metering module for current transformers, groundable on one side	
General information		
B&R ID code	0xE7BF	0xE7C0
Status indicators	I/O function per channel, operating state, module status	
Diagnostics		
Module run/error	Yes, using LED status indicator and software	
Inputs	Yes, using LED status indicator and software	
Power consumption		
Bus	0.85 W (Rev. <C0) 0.50 W (Rev. =C0) 0.45 W (Rev. >C0)	
Internal I/O	-	
Additional module power dissipation [W]	2 W [1]	
Insulation voltages		
Voltage inputs - Current inputs	Tested at 1300 VAC, 1 min	
Inputs - Bus / I/O power supply	Tested at 5500 VDC, 1 min	
Inputs - Ground	Tested at 5500 VDC, 1 min	
Bus / I/O power supply - Ground	Tested at 510 VAC, 1 min	
Certifications		
CE	Yes	
UKCA	Yes	
ATEX	Zone 2, II 3G Ex nA nC IIA T5 Gc IP20, Ta (see X20 user's manual) FTZÚ 09 ATEX 0083X	
UL	cULus E115267 Industrial control equipment	
EAC	Yes	
Voltage inputs		
Number of phases	3	
Input impedance	1.68 MΩ	
Measurement category	CAT II	
Nominal voltage U_N		
Between phases	Max. 480 VAC [2]	
Phase to N	Max. 277 VAC	
Max. display value	655 VAC	
Resolution	10 mV, with voltage connected directly	
Rated frequency	50 and 60 Hz	
Measurable frequency		
Measurement range	45 to 65 Hz	
Resolution	0.01 Hz	
Current inputs		
Quantity	4 AC inputs	
Measurement category	CAT II	
Nominal current I_N		
Secondary	1 A	5 A
Primary	65 A directly configurable, larger values through conversion in the application [3]	
Max. overload current	8 x I_N for 0.5 s	
Max. measurement current	1 A	5 A
Resolution	1 mA, based on the primary current [3]	
Load	250 mΩ	20 mΩ
Measurement accuracy [4]		
U_{RMS}	±0.65% [5]	
I_{RMS}	±0.65% [6]	±0.65% [5]
Effective, reactive and apparent power	±0.80% [7]	±0.80% [8]
Frequency, power factor and phase angle	±0.50% [9]	
Active energy per phase and total [10]	±0.40% [7]	±0.40% [8]
Active energy of fundamental frequency per phase and total [10]	±0.50% [7]	±0.50% [8]
Active energy of harmonics per phase and total [10]	±0.80% [11]	±0.80% [8]
Reactive energy per phase and total [12]	±0.50% [7]	±0.50% [8]
Apparent energy		
Per phase and arithmetic total	±0.50% [7]	±0.50% [8]
Vector sum	±0.80% [7]	±0.80% [8]
Electrical properties		
Electrical isolation	Channel isolated from bus Channel not isolated from channel	

Table 3: X20AP3122, X20AP3132 - Technical data

Order number	X20AP3122	X20AP3132
Operating conditions		
Mounting orientation		
Horizontal	Yes	
Vertical	Yes	
Installation elevation above sea level		
0 to 2000 m	No limitation	
>2000 m	Not permitted	
Degree of protection per EN 60529	IP20	
Ambient conditions		
Temperature		
Operation		
Horizontal mounting orientation	-25 to 60°C	
Vertical mounting orientation	-25 to 50°C	
Derating	-	
Storage	-40 to 85°C	
Transport	-40 to 85°C	
Relative humidity		
Operation	5 to 95%, non-condensing	
Storage	5 to 95%, non-condensing	
Transport	5 to 95%, non-condensing	
Mechanical properties		
Note	Order 1x terminal block X20TB32 separately. Order 1x bus module X20BM32 separately.	
Pitch	$25^{+0.2}$ mm	

Table 3: X20AP3122, X20AP3132 - Technical data

1) Power dissipation of current measurement shunts
2) The design of the module allows 480 VAC to be applied to the terminal block.
3) For measuring higher current values, see section "Current transformer - Pinout".
4) Based on the current measured value.
   The actual error value percentage may be larger due to the digital display.
5) With drift of 25 ppm/K
6) With drift of 100 ppm/K
7) With drift of 125 ppm/K
8) With drift of 50 ppm/K
9) In power systems with approximately sinusoidal voltage starting at 10 VAC.
10) At power factor cos ϕ = 1, 0.5L and 0.8C
11) With drift of 125 ppm/K (Rev. < D0) or 40 ppm/K (Rev. ≥ D0)
12) At reactive power factor sin ϕ = 1, 0.5L and 0.8C

3.1.3 X20AP3161 and X20AP3171

Order number	X20AP3161	X20AP3171
Short description		
I/O module	3-phase power and energy metering module for current/voltage transformers	3-phase power and energy metering module for Rogowski current transformers
General information		
B&R ID code	0xE17B	0xE7C1
Status indicators	I/O function per channel, operating state, module status	
Diagnostics		
Module run/error	Yes, using LED status indicator and software	
Inputs	Yes, using LED status indicator and software	
Power consumption		
Bus	0.85 W (Rev. <D0) 0.50 W (Rev. =D0) 0.45 W (Rev. >D0)	0.85 W (Rev. <C0) 0.50 W (Rev. =C0) 0.45 W (Rev. >C0)
Internal I/O	-	
Additional module power dissipation [W]	- [1)]	
Insulation voltages		
Current inputs / Neutral conductor - Ground	-	Tested at 2300 VAC, 1 min
Voltage inputs / Neutral conductor - Ground	-	Tested at 3700 VAC, 1 min
Current inputs / Neutral conductor - Bus / I/O power supply	-	Tested at 2300 VAC, 1 min
Voltage inputs / Neutral conductor - Bus / I/O power supply	-	Tested at 3700 VAC, 1 min
Inputs - Bus / I/O power supply	Tested at 5500 VDC, 1 min	
Inputs - Ground	Tested at 5500 VDC, 1 min	
Bus / I/O power supply - Ground	Tested at 510 VAC, 1 min	
Certifications		
CE	Yes	
UKCA	Yes	
ATEX	Zone 2, II 3G Ex nA nC IIA T5 Gc IP20, Ta (see X20 user's manual) FTZÚ 09 ATEX 0083X	
UL	cULus E115267 Industrial control equipment	
HazLoc	cCSAus 244665 Process control equipment for hazardous locations Class I, Division 2, Groups ABCD, T5	-
DNV	Temperature: **B** (0 to 55°C) Humidity: **B** (up to 100%) Vibration: **B** (4 g) EMC: **B** (bridge and open deck)	-
LR	ENV1	-
KR	Yes	-
ABS	Yes	-
BV	**EC33B** Temperature: 5 - 55°C Vibration: 4 g EMC: Bridge and open deck	-
EAC	Yes	
Voltage inputs		
Number of phases	3	
Input impedance	1.68 MΩ	
Measurement category	-	CAT II
Nominal voltage U_N		
Between phases	Max. 480 VAC [2)]	
Phase to N	Max. 277 VAC	
Max. display value	655 VAC	
Resolution	10 mV, with voltage connected directly	
Rated frequency	50 and 60 Hz	
Measurable frequency		
Measurement range	45 to 65 Hz	
Resolution	0.01 Hz	
Current inputs		
Quantity	4 AC inputs	
Measurement category	-	CAT II
Nominal voltage (secondary)	333 mV	Configurable in µV/A
Nominal current (primary)	65 A directly configurable, larger values through conversion in the application [3)]	
Max. overload current	-	
Max. measurement voltage	333 mV	52 mV
Resolution	1 mA, based on the primary current [3)]	
Load	-	
Measurement accuracy [4)]		
U_{RMS}	±0.65% [5)]	

Table 4: X20AP3161, X20AP3171 - Technical data

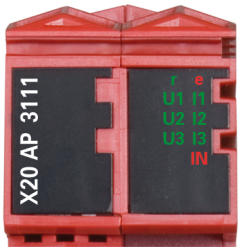
Order number	X20AP3161	X20AP3171
I_{RMS}	±0.65%	±0.85% 6)
Effective, reactive and apparent power	±0.80% 5)	±1.00%
Frequency, power factor and phase angle	±0.50% 7)	
Active energy per phase and total 8)	±0.40% 5)	±0.60% 5)
Active energy of fundamental frequency per phase and total 8)	±0.50% 5)	±0.70% 5)
Active energy of harmonics per phase and total 8)	±0.80% 5)	±1.00% 5)
Reactive energy per phase and total 9)	±0.50% 5)	±0.70% 5)
Apparent energy		
Per phase and arithmetic total	±0.50% 5)	±0.70% 5)
Vector sum	±0.80% 5)	±1.00% 5)
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	Not permitted	
Degree of protection per EN 60529	IP20	
Ambient conditions		
Temperature		
Operation		
Horizontal mounting orientation	-25 to 60°C	
Vertical mounting orientation	-25 to 50°C	
Derating	-	
Storage	-40 to 85°C	
Transport	-40 to 85°C	
Relative humidity		
Operation	5 to 95%, non-condensing	
Storage	5 to 95%, non-condensing	
Transport	5 to 95%, non-condensing	
Mechanical properties		
Note	Order 1x terminal block X20TB32 separately. Order 1x bus module X20BM32 separately.	Order 1x terminal block X20TB32 separately. Order 1x bus module X20BM32 separately.
Pitch	$25^{+0.2}$ mm	

Table 4: X20AP3161, X20AP3171 - Technical data

1) Shunts are external in the current transformer.
2) The design of the module allows 480 VAC to be applied to the terminal block.
3) For measuring higher current values, see section "Current transformer - Pinout".
4) Based on the current measured value.
   The actual error value percentage may be larger due to the digital display.
5) With drift of 25 ppm/K
6) At URogowski > 1 mVRMS
7) In power systems with approximately sinusoidal voltage starting at 10 VAC.
8) At power factor cos ϕ = 1, 0.5L and 0.8C
9) At reactive power factor sin ϕ = 1, 0.5L and 0.8C

3.2 LED status indicators

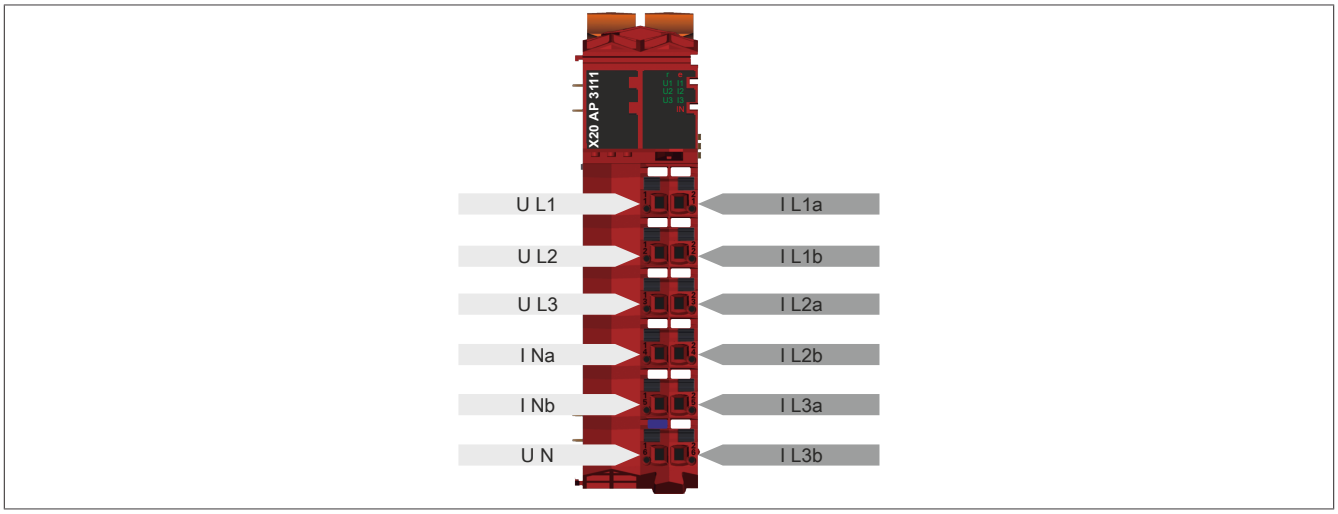
For a description of the various operating modes, see section "Additional information - Diagnostic LEDs" in the X20 system user's manual.

Figure	LED	Color	Status	Description
	**Operating state**			
	r	Green	Off	No power to module
			Single flash	UNLINK mode
			Double flash	BOOT mode (during firmware update)[1)]
			Blinking quickly	SYNC mode
			Blinking slowly	PREOPERATIONAL mode
			On	RUN mode
	Module status			
	e	Red	Off	No power to module or everything OK
			On	Error or reset status
	Analog input voltage			
	U1 - U3	Green/Yellow	Off	Display disabled or U_{Eff} < threshold value "Failure"
		Green	Blinking	Phase sequence is correct and U_{Eff} < threshold value "Warning"
			On	Phase sequence is correct and U_{Eff} > threshold value "Warning"
		Yellow	Blinking	Phase sequence is incorrect and U_{Eff} < threshold value "Warning"
			On	Phase sequence is incorrect and U_{Eff} > threshold value "Warning"
	Analog input current			
	I1 - I3	Green/Yellow	Off	Display disabled or I_{Eff} < threshold value "Display"
		Green	On	Active power positive
		Yellow	On	Active power negative
	Analog input neutral current			
	IN		Off	Neutral current < Threshold value
		Green	On	Neutral current > Threshold value "Failure", within the tolerance of the calculated total current
		Red	On	Neutral current > Threshold value "Failure", outside the tolerance of the calculated total current

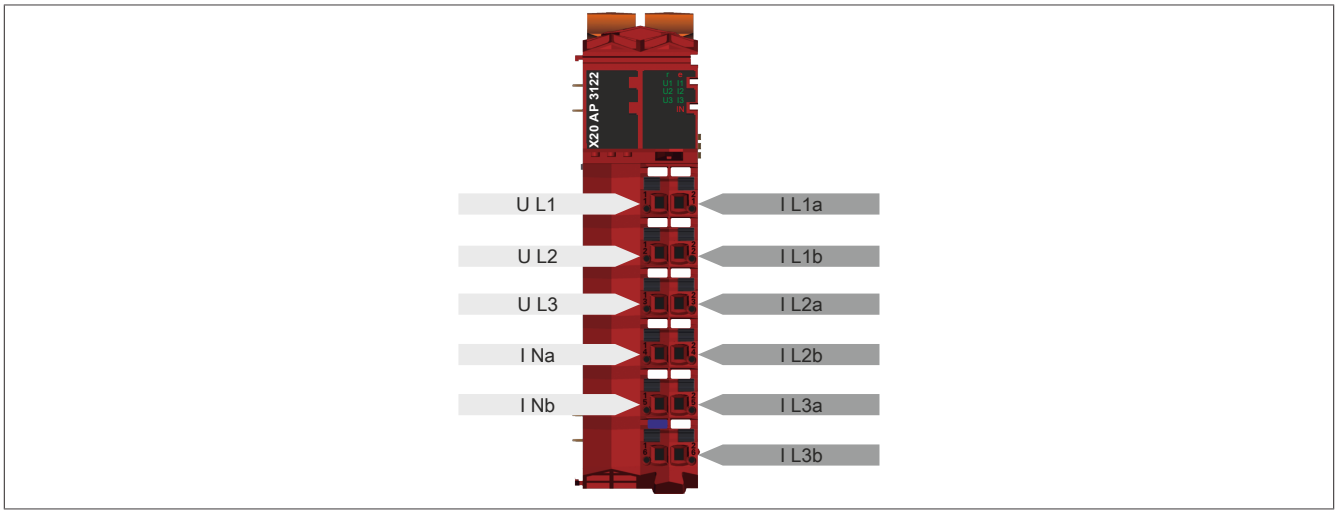
1) Depending on the configuration, a firmware update can take up to several minutes.

3.3 Pinout

X20AP31x1



X20AP31x2



Danger!

Risk of electric shock!

The terminal block is only permitted to conduct voltage when it is connected. It is not permitted to be disconnected or connected while voltage is applied or have voltage applied to it while it is removed under any circumstances!

This module is not permitted to be the last module connected on the X2X Link network. At least one subsequent X20ZF dummy module must provide protection against contact.

Danger !

Risque d'électrocution !

Le connecteur ne peut conduire la tension que lorsqu'il est connecté. Il est interdit de le déconnecter ou de le connecter lorsqu'une tension est appliquée ou lorsqu'une tension lui est appliquée lorsqu'il est retiré dans n'importe quelle circonstance !

Il est interdit que ce module soit le dernier module connecté sur le réseau X2X Link. Au moins un module factice X20ZF ultérieur doit assurer la protection contre les contacts.

Shielding

Shielded cables must be used for the current channels of module X20AP3171 in order to maintain the specified accuracy. Cabling can either take place using one cable per channel or a multiple twisted pair cable for multiple channels.

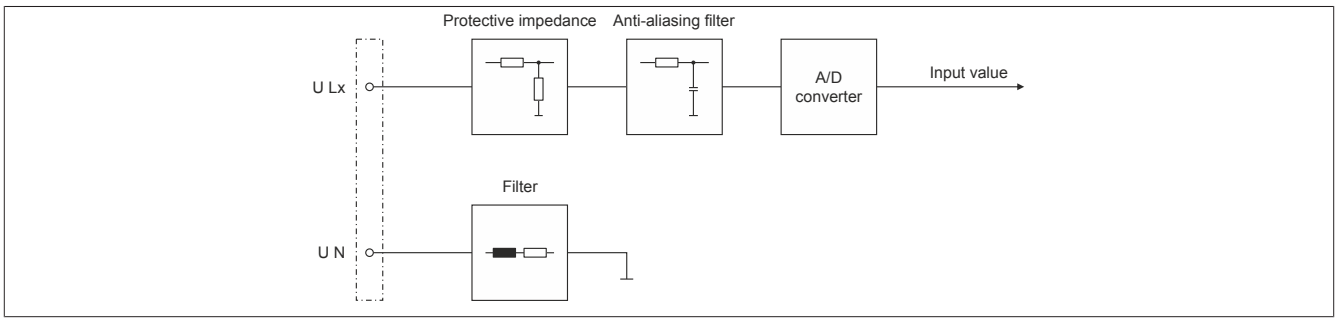
Information:

Shielded cables must be grounded on both sides.

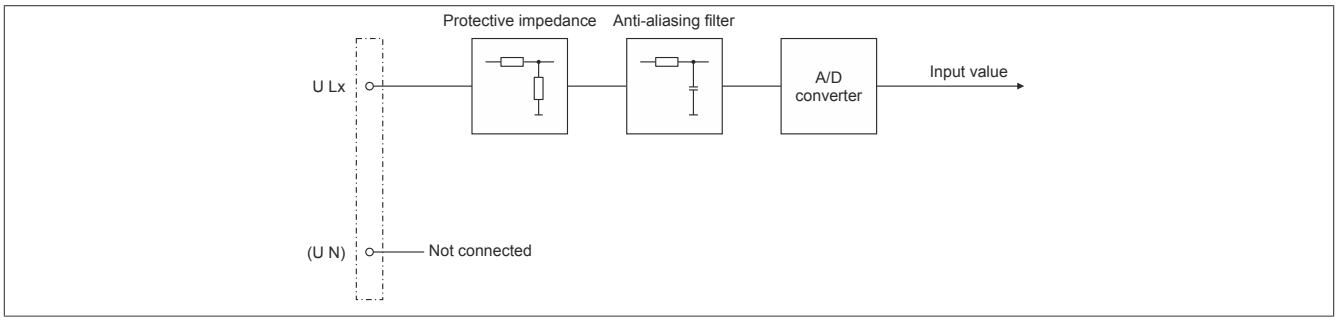
3.4 Input circuit diagram

AC voltage inputs

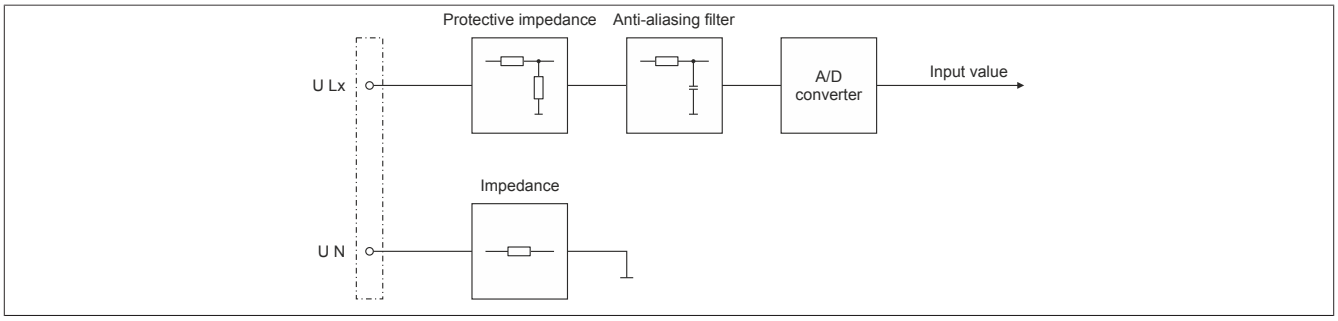
AP3111, AP3121, AP3131, AP3161



AP3122, AP3132

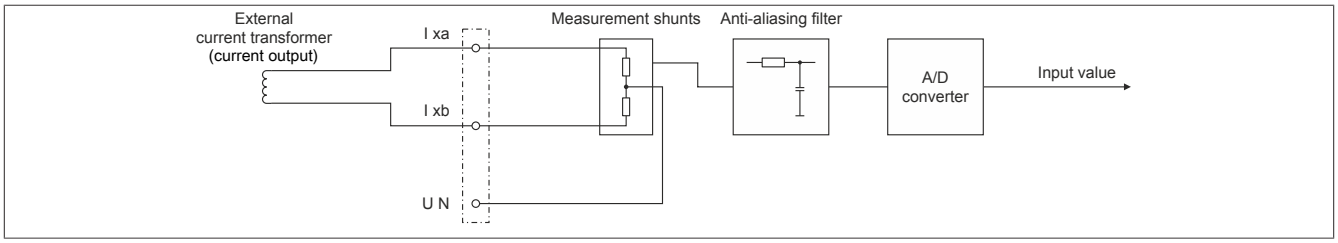


AP3171

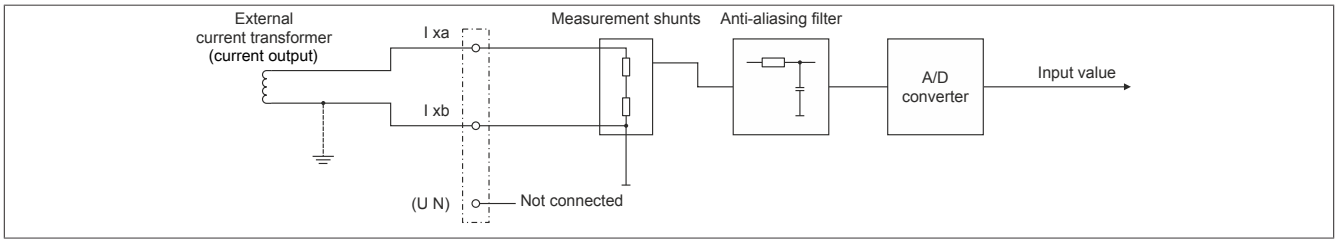


AC current inputs

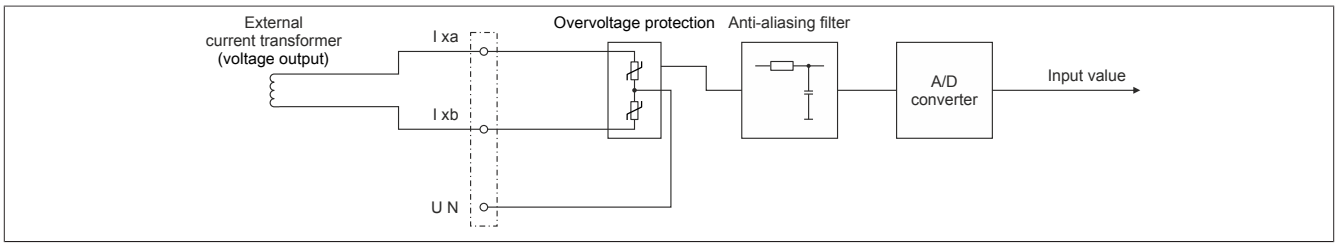
AP3111, AP3121, AP3131: (Current measurement)



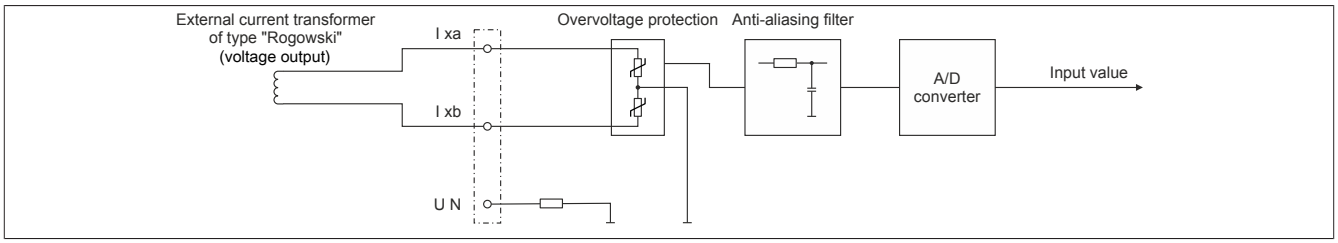
AP31x2: (Current measurement)



AP3161: (Voltage measurement)



AP3171: (Voltage measurement)



3.5 Typical connection examples for different network configurations

General information

There are many different network configurations around the world. This section will present a few typical connection examples.

Notice!

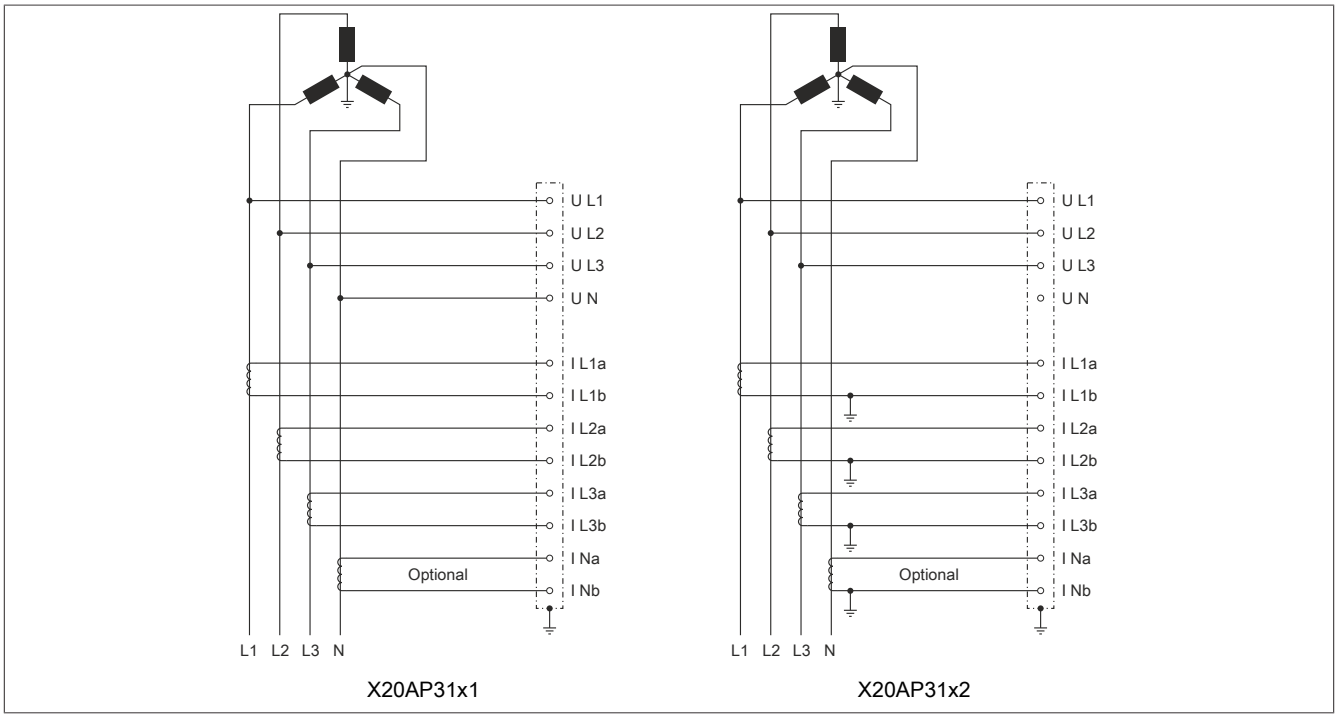
Modules X20AP31x2 are not permitted to be used with network configurations B, D and F due to the lack of grounding.

Avis !

Il n'est pas autorisé d'utiliser les modules X20AP31x2 avec les configurations de réseau B, D et F en raison de l'absence de mise à la terre.

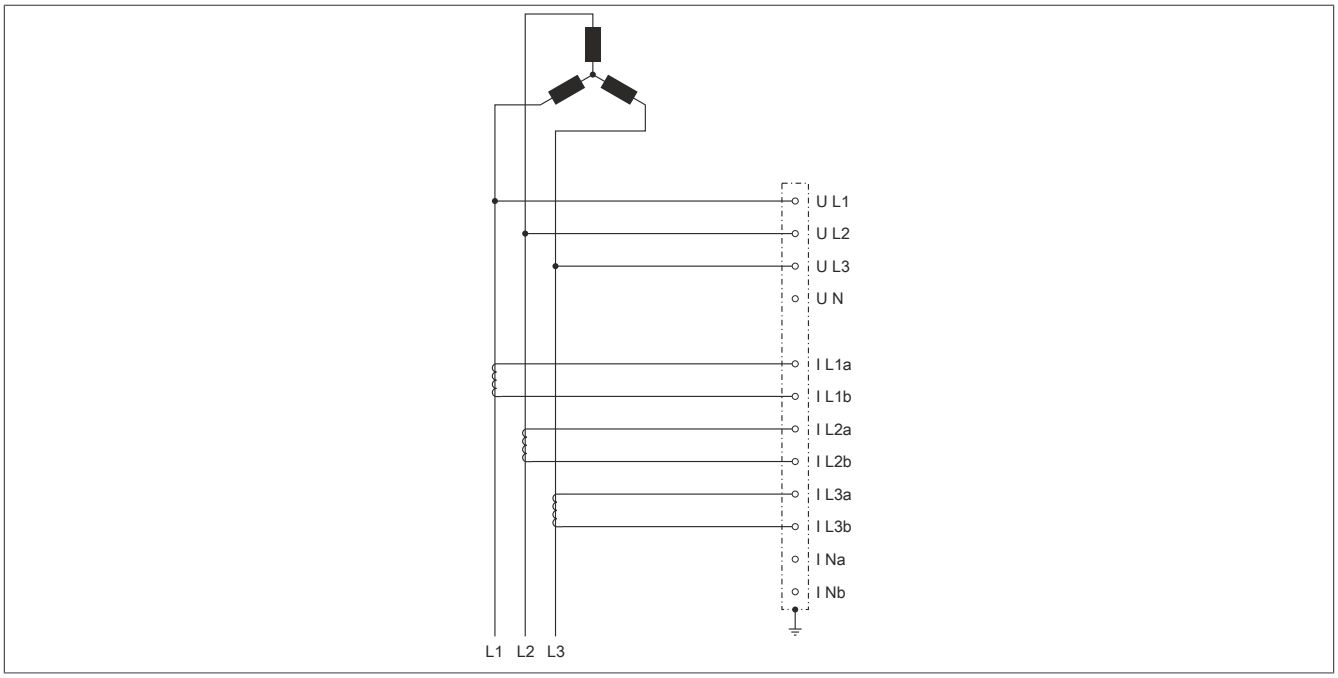
Connection example 1 - Network A

This example involves a 3-element, 3-phase, 4-line star measurement with grounded neutral conductor and optional fault current detection.



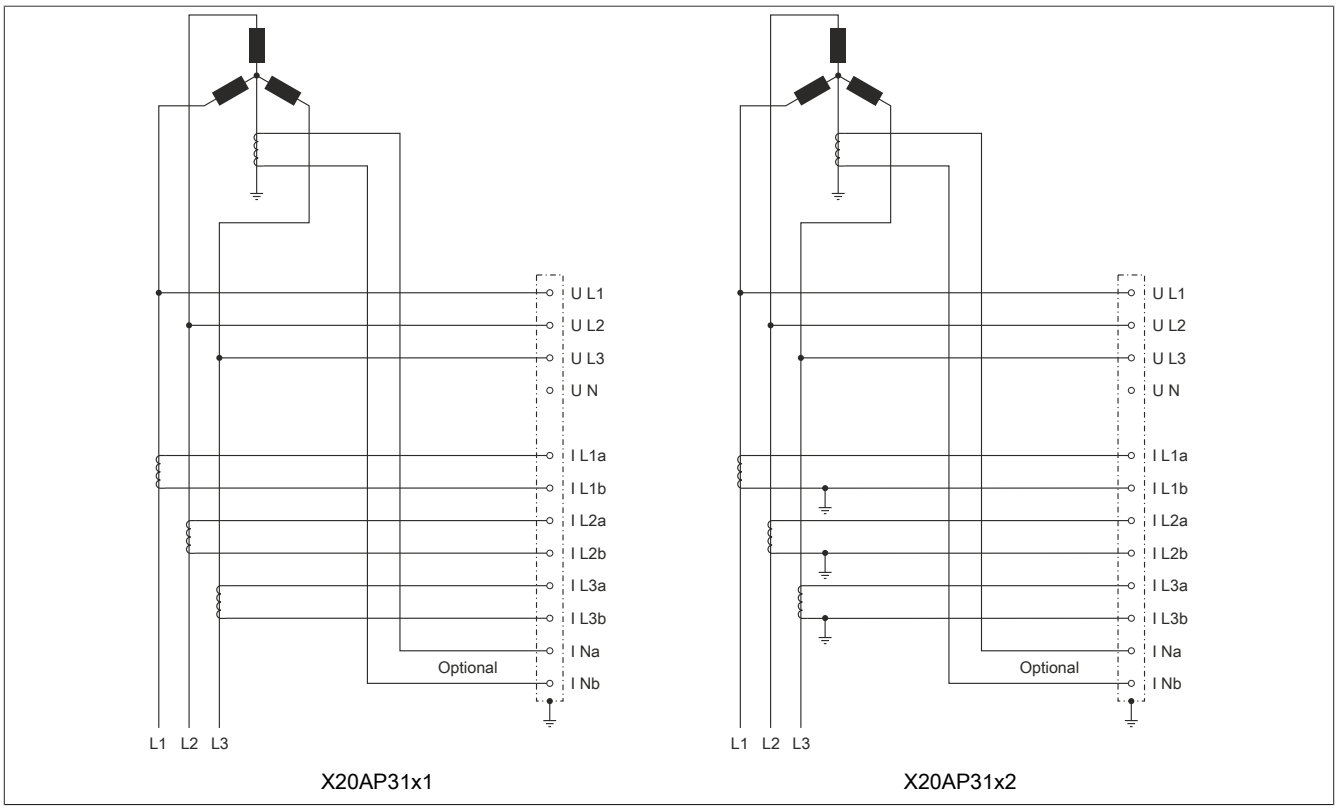
Connection example 2 - Network B

This example involves a 3-element, 3-phase, 3-line star measurement.



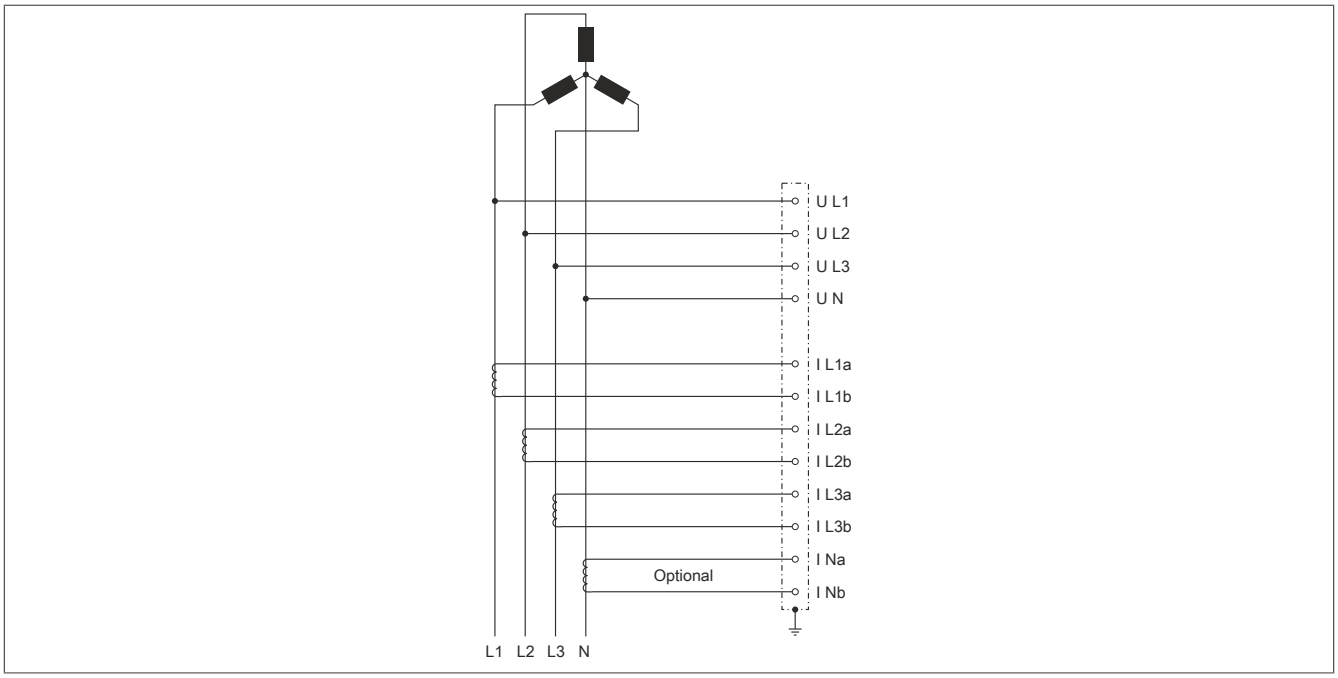
Connection example 3 - Network C

This example involves a 3-element, 3-phase, 3-line star measurement with grounded neutral conductor and optional fault current detection.



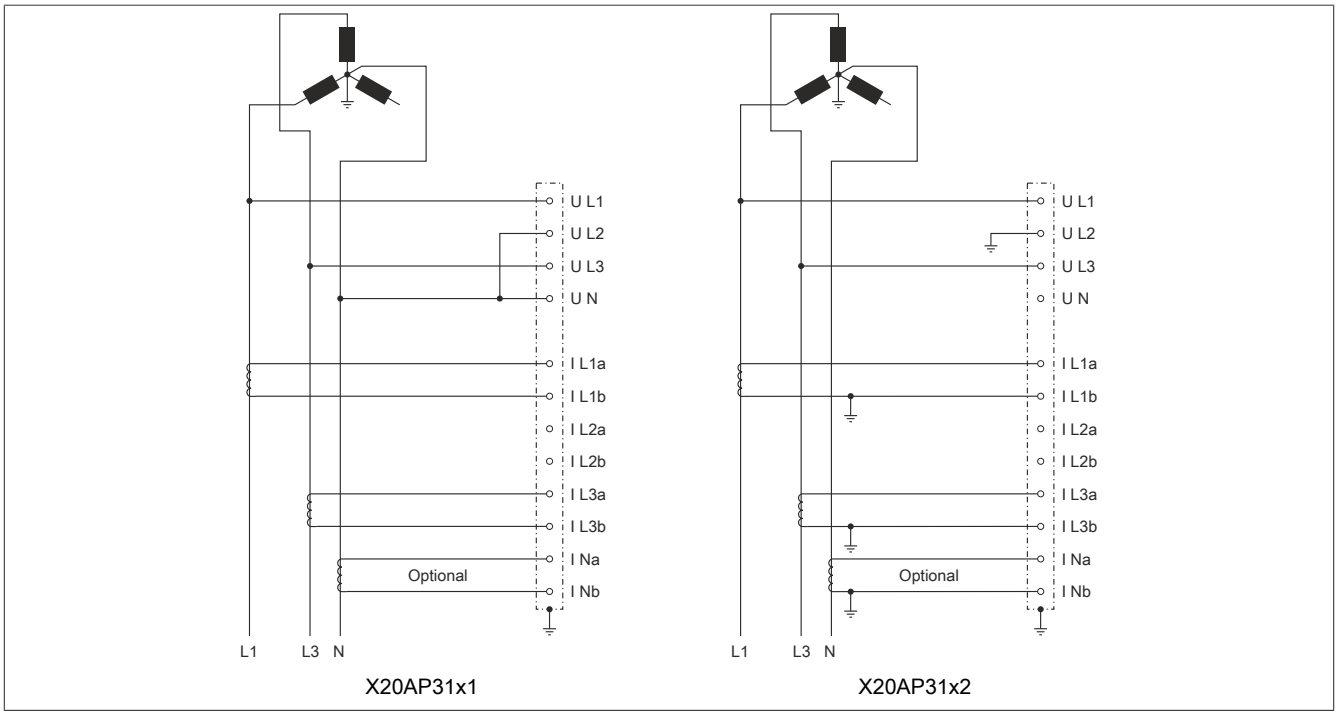
Connection example 4 - Network D

This example involves a 3-element, 3-phase, 4-line star measurement with optional fault current detection.



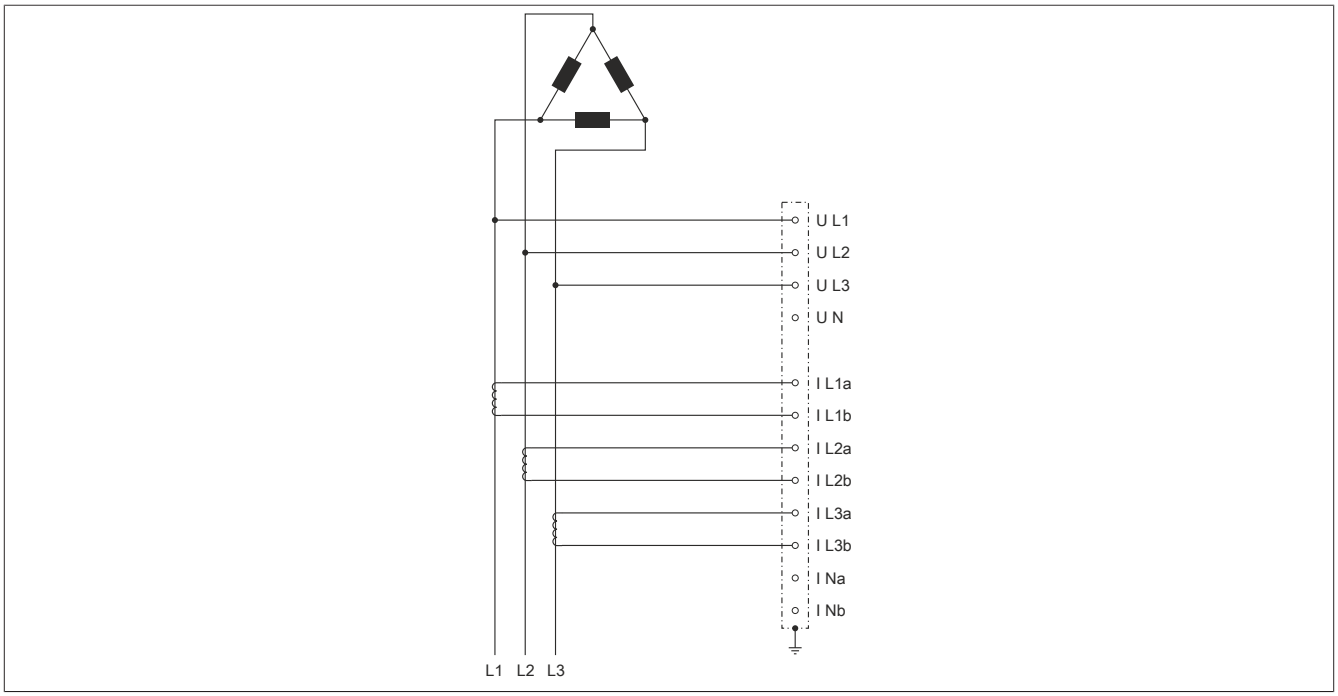
Connection example 5 - Network E

This example involves a 2-element, 2-phase, 3-line star measurement with grounded neutral conductor.



Connection example 6 - Network F

This example involves a 3-element, 3-phase, 3-line delta measurement.

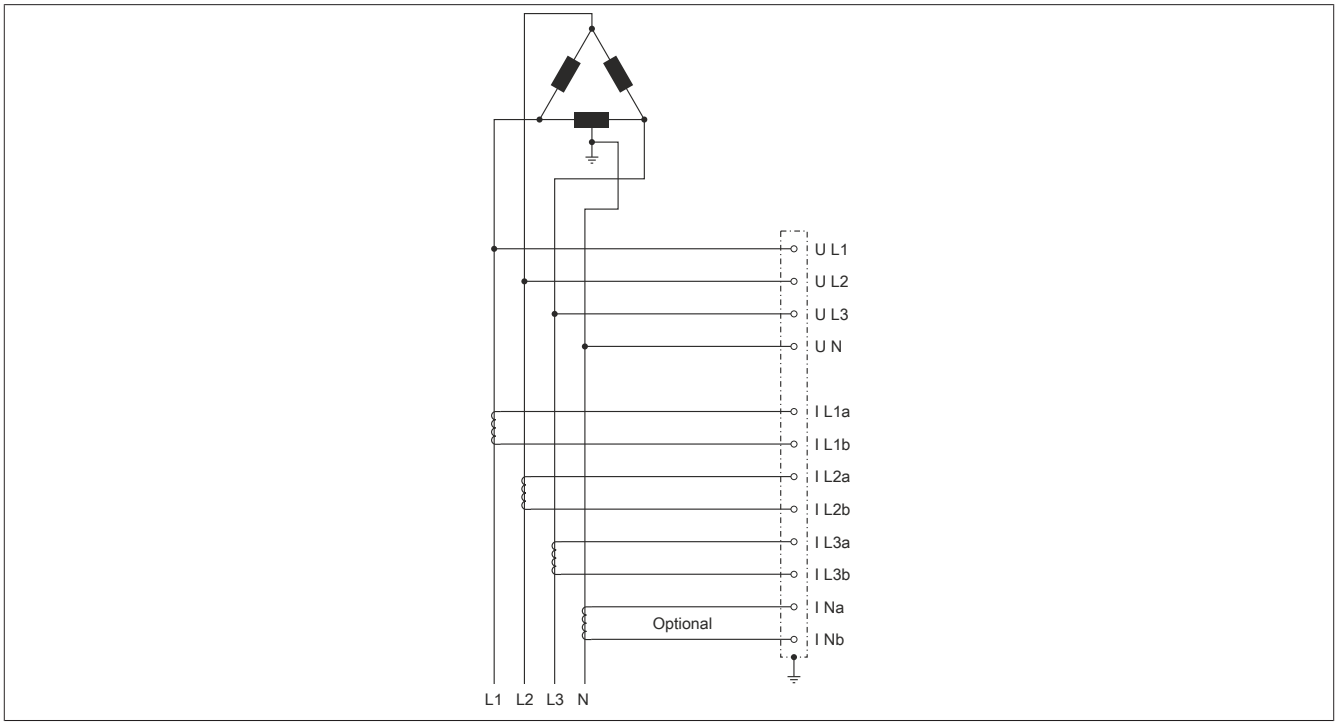


Connection example 7 - Network G

This example involves a 3-element, 3-phase, 4-line, delta measurement with grounded star point.

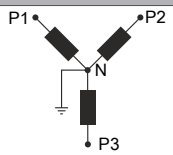
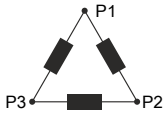
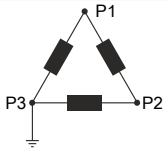
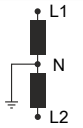
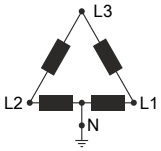
Information:

Exceeding the maximum voltage value specified in the data sheet is not permitted!



3.6 Permitted line-to-line nominal voltages (rated voltages)

The following table provides an overview of the maximum permitted line-to-line nominal voltage (rated voltage) depending on the used mains type and module.

Network configuration	Network type		Modules	Permissible rated voltage
1	3 phases 4 lines Grounded neutral conductor		All AP modules	480 V
2	3 phases 3 lines Not grounded		X20AP31x1	480 V
			X20AP31x2	Not permitted
3	3 phases 4 lines Grounded phase		X20AP31x1	480 V
			X20AP21x2	Not permitted
4	1 phase 2 lines Not grounded		X20AP31x1	480 V
			X20AP31x2	Not permitted
5	1 phase divided 3 lines Grounded neutral conductor		All AP modules	480 V
6	3 phases 4 lines Divided phase and grounded neutral conductor		All AP modules	240 V

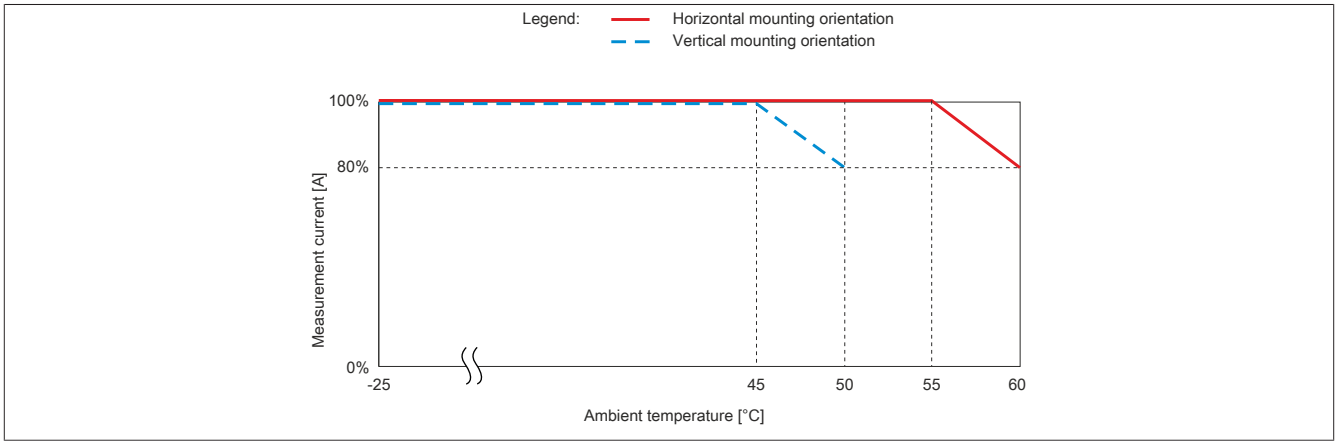
3.7 Derating

X20AP(c)3121, X20AP3131:

At high temperatures, the averaged measurement current is not permitted to exceed the percentage values of the diagram.

100% corresponds to 3x the nominal current of a channel. Averaging the measurement current is done in a time period of 10 minutes.

The derating listed below must be applied for the current:



4 Function description

4.1 Energy management

The modules' integrated functions not only map the machine's current power requirements in detail, they also serve to record the consumption of the machine or system. All relevant data is available to the user in the process image.

Due to the current and voltage measurement up to the 31st harmonic, the RMS values can be recorded much more precisely than is generally the case. This means that the modules not only handle impure sinusoidal curves well, they are also ideal for use in renewable energy generation. In the latter applications, for example, the precise measurement of the mains frequency with a 0.01 Hz resolution between 45 and 65 Hz is a huge advantage. The modules are generally suited for use in 1-phase, 2-phase or 3-phase networks.

4.2 Measurement function

The modules provide the following possibilities for measured value preparation:

- Temperature coefficient of internal reference of 6 ppm/°C
- Energy registers for active, reactive and apparent energy, separated for forward and backward, fundamental waves and harmonics
- Threshold register for status signal generation and activation of power and energy measurement
- Determining the THD harmonic component
- Discrete Fourier Transformation (DFT) up to 31st harmonic component per phase for voltage and current
- Status signals for voltage dip, loss of voltage, phase sequence, energy flow, neutral current monitor, harmonic component monitor

Measuring times

The measured values for RMS value, power, active power factor, phase angle and frequency are average values over 16 full waves; the update rate is ~3 Hz.

The following represents the measurement time over 16 full waves at the corresponding frequency:

50 Hz → 320 ms
60 Hz → 267 ms

4.2.1 Measurement ranges

Information	Description
Measurement range limiting	Due to the majority of registers consisting of 16-bit values (exception: energy registers, which are interpolated to 32-bit by the firmware), the measurement ranges are subject to limitations, e.g. voltage 650.00 Vrms and current 65,000 Arms (after accounting for the transfer factor of the current transformer).
Extended measurement ranges	Extended measurement ranges can be achieved with the software application by upscaling the measured values.

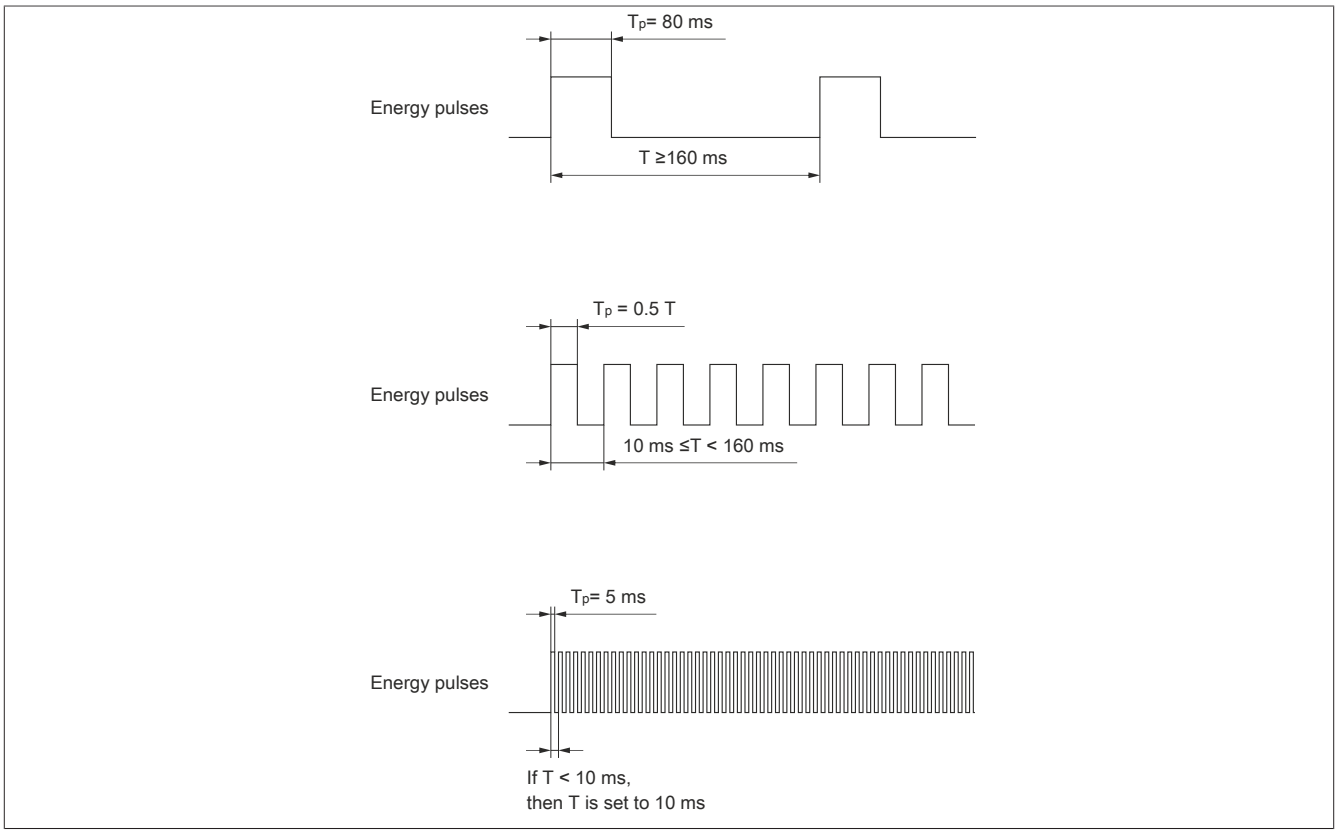
4.2.2 Energy measurement

The power measurement (energy measurement) is based on the integration of the measured values with a sampling rate of 1 MHz.

The gathered energy values are provided according to the set unit (1 Ws, 10 Ws, etc.) in the energy registers.

Automatic reading of the energy meter from the transformer must be enabled because valid values are only available after the transformer has been configured. It is possible to clear the energy registers or to configure them with a block of the register written in the software application.

The energy pulses are recorded at an interval of 200 µs. The length of the energy pulses varies with the resulting output rate.



Power line constants

The units of the collected energy values are defined by the power line constant.

Base value

10 increments in the energy register result in 1 energy pulse. Base value 0x4A81 7C80 = 1,250,000,000 corresponds to 360 energy pulses per kWh or 0.1 energy pulse per kWs. In the energy registers, this results in 1 kWs per digit.

Only the following units can be set. Other values are not permitted.

1 increment in the energy register corresponds to:	PLConstH	PLConstL
1 Ws	0x0013	0x12D0
10 Ws	0x00BE	0xBC20
100 Ws	0x0773	0x5940
1 kWs (bus controller default setting)	0x4A81	0x7C80
1 Wh	0x0010	0x0034
1 kWh	0x417B	0xCE6C

Information:

When 1 Wh and 1 kWh are set, the energy pulses on the register "StatusInput" on page 75 may not be used.

4.2.3 Power measurement

The modules measure effective, reactive and apparent power individually for each of the 3 phases and for all of them collectively. The power consumption of each phase is also recorded individually and in total. The modules also provide the effective values of the 3 phases for voltage and current. When measuring the current, the value of the current over the neutral conductor can also be recorded and monitored. The measurement of the mains frequency and the phase angles of the 3 phases (of current and voltage) complete the electrical power measurement data.

The phase outputs are calculated by the module and stored in the corresponding registers.

The total power ratings are equal to the sum of the phase power ratings. To prevent the number range from being exceeded, the value in the registers is equal to a fourth of the actual power. This value must be multiplied by 4 by the application.

The vector-based total apparent power (complex total apparent power) is calculated according to IEEE1459.

4.2.4 Power factor

The phase power factor is calculated by dividing the phase active power by the phase apparent power.

The total power factor is calculated by dividing the total active power by the total apparent power.

4.2.5 Neutral current

The neutral current can be measured or calculated. Both values are available.

The user can configure which one to use for displaying the status.

4.2.6 Phase angle

The phase angle is calculated based on the zero-crossing detection.

4.2.7 Frequency

Frequency measurement is based on Phase A. If A fails, then Phase C is used. If both A and C fail, then Phase B is used.

4.2.8 Temperature

The Chip-Junction temperature is measured approximately every 100 ms using the sensor integrated in the transformer.

4.2.9 THD+N - Sum of interference power of the harmonic (THD) + interference power of the noise (N)

The THD+N measurement is used to monitor the percentage of harmonics in the network.

If this percentage falls below 10%, then an accuracy of 0.01% can no longer be guaranteed.

This is calculated as follows: (SQR (RMS value$_{Total}$^2 - RMS value$_{FundamentalWave}$^2)) / RMS value$_{FundamentalWave}$

4.3 Fourier analysis

The harmonic component from the 2nd to the 31st harmonic is calculated for voltage and current and the THD (Total Harmonic Distortion) of each phase.

The DFT period (DFT = discrete Fourier Transformation) is 0.5 s. This corresponds to a resolution of 2 Hz. The input samples are recorded at a sampling rate of 8 kHz and can be optionally multiplied with a "Hann window" before being evaluated. This is initiated when requested by the application.

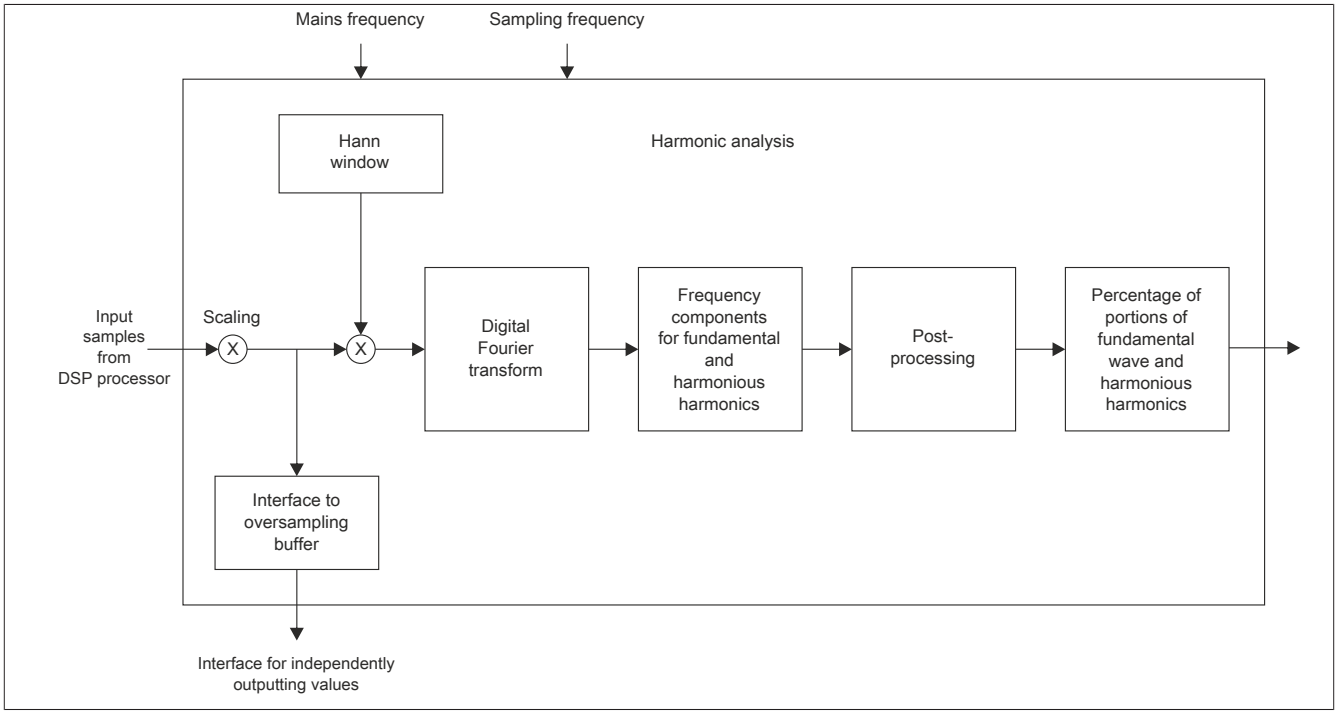


Figure 1: Diagram of Fourier analysis

Calculating the fundamental wave current

The fundamental wave current is calculated according to the following formula:

Standard calculation

$$\text{Fundamental wave}_{mA} = \text{Register value} * 3.2656 * \frac{\text{Ratio}_{Real}}{\text{Ratio}_{Configured}}$$

Inverted calculation

$$\text{Fundamental wave}_{mA} = \text{Register value} * 3.2656 * \frac{\text{Ratio}_{Configured}}{\text{Ratio}_{Real}}$$

Legend

Register value: The value of the "fundamental wave current register" on page 90.

$\text{Ratio}_{Configured}$: Configured rated value divided by 10.
See "Current transformer rating phase A/B/C/N" on page 93.

Ratio_{Real}: The real rated value depends on the AP module being used:

Module	Value
X20AP3111	25000
X20AP3121	500
X20AP3131	100
X20AP3161	500
X20AP3171	Default: 5000 Inverted: 1
All others	1

Calculating the fundamental wave voltage

The fundamental wave voltage is calculated according to the following formula:

$$\text{Fundamental wave}_{\text{Voltage}} = \text{Register value} * 3.2656 * 10^{-2}$$

Legend

Register value The value of the "fundamental wave voltage register" on page 90.

Information:

The registers are described in "Analog discrete Fourier transformation register (DFT)" on page 90.

4.4 Monitoring functions

Information:

The registers are described in "Status register" on page 75.

4.4.1 Zero-crossing detection

Zero-crossing detection can be configured for each phase for current or voltage and edge and forms the basis for frequency and angle measurements and subsequently also for active and reactive power calculations.

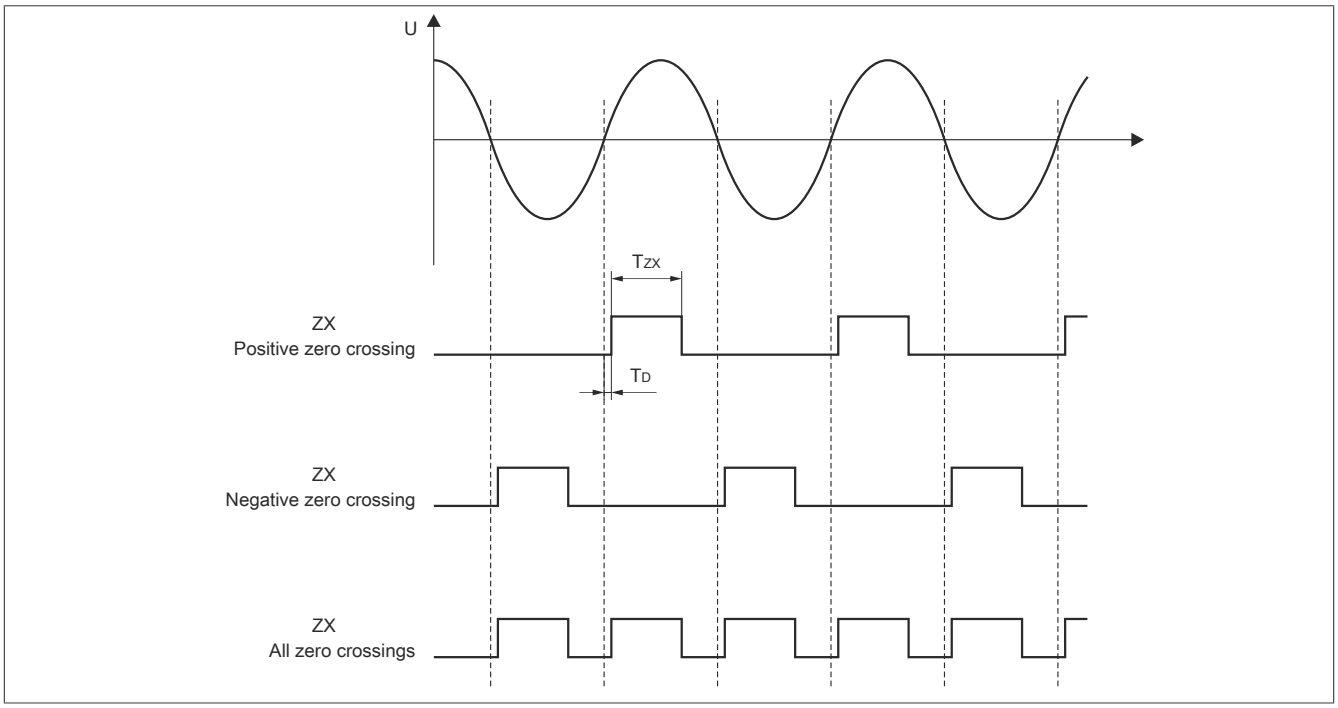


Figure 2: Timing diagram of zero-crossing detection per phase

Symbol	Description	Minimum	Typical	Maximum	Unit
T_{ZX}	Length of high signal		5		ms
T_D	Delay time		0.2	0.5	ms

4.4.2 Voltage dip or power failure detection

Event	Description
Voltage dip	The threshold for the voltage dip is typically set to 78% of the standard voltage (approx. 170 Vrms). The status flag is set if more than 3 8 kHz samples are below the threshold within 2 consecutive 11 ms windows.
Power failure	The threshold for the power failure is typically set to 10% of the standard voltage (approx. 22 Vrms). The status flag is set if more than 3 8 kHz samples are below the threshold within 2 consecutive 11 ms windows. If a power failure is detected, zero crossing detection for voltage and current is disabled for this phase.

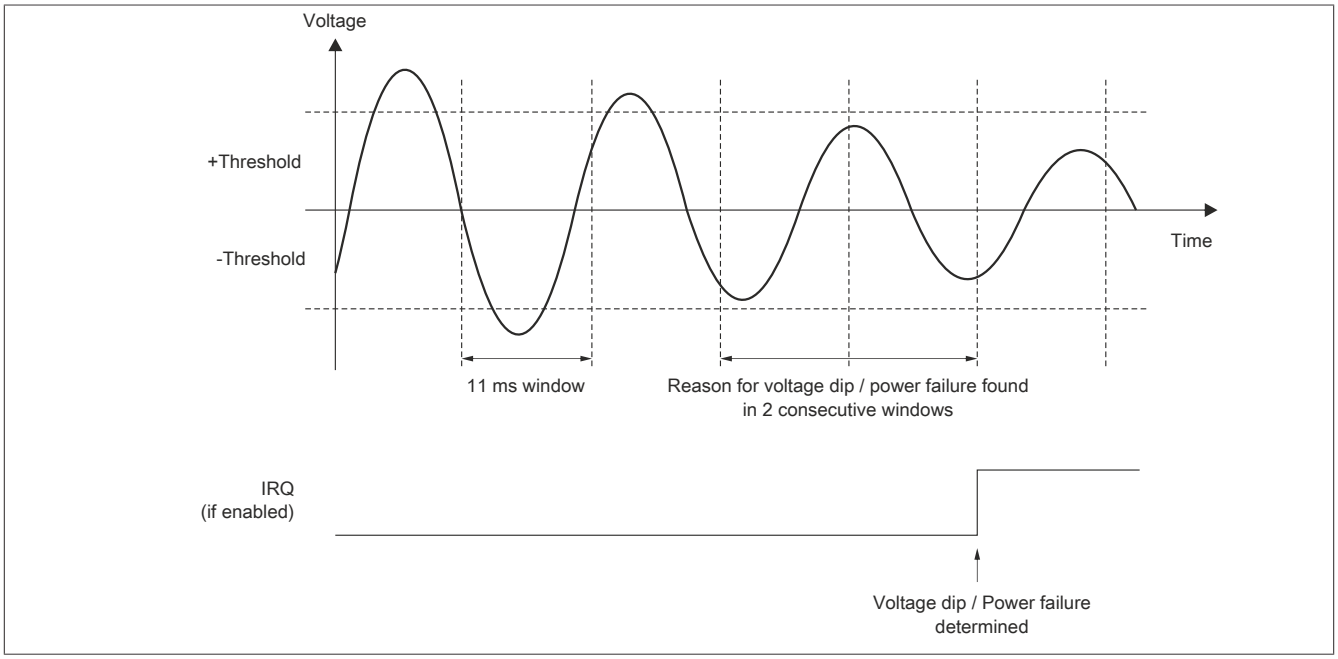


Figure 3: Timing diagram for detection of a voltage dip or power failure

4.4.3 Neutral current monitoring

Neutral current monitoring of the measured and the calculated value is done with separate threshold value registers and status flags.

4.4.4 Phase sequence monitoring

3 phase and 2 phase applications are handled differently:

Application	Description
3 phases	Zero cross-overs of voltage and current must follow the sequence Phase A before Phase B before Phase C
2 phases	Zero cross-overs of voltage and current must follow the sequence Phase A at least 180° before Phase C

4.5 Oversampling

The input values are measured with a configurable sampling cycle time and saved in the internal physical data buffer with timestamp. This data area can now be read out by means of configurable data length in cyclic data transfer.

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

With fast cycle times, however, it can happen that the set sampling cycle time is not sufficient for the sum of all physical and logical functions. If the physical sampling should remain unaffected, logical processing can be slowed down via a prescaler setting.

Information:

Due to the free configurability of the sampling cycle time on the module, there is no synchronicity to the X2X Link network, regardless of configuration as standard inputs or with the oversampling function.

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

Information:

The registers are described in "Oversampling buffer" on page 103.

4.5.1 Sample lines

A sample line contains the instantaneous values of currents (4 channels) and voltages (3 channels), a consecutive number and the NetTime at the moment of transfer from the converter. These values are recorded at intervals of a 125 µs multiplied by the prescaler.

Standardization to corresponding physical values must be performed by the user:

Voltage:	Vrms = (INT32)Vs * 4 / Sqrt(2)
Current:	Irms = (INT32)Is * 4 / Sqrt(2)

4.5.2 Frozen values

Sample time register
If the group of measured values is read from the power meter, they are assigned a NetTime. This NetTime can be used to detect when values freeze.

4.6 NetTime Technology

NetTime refers to the ability to precisely synchronize and transfer system times between individual components of the controller or network (controller, 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.

4.6.1 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.

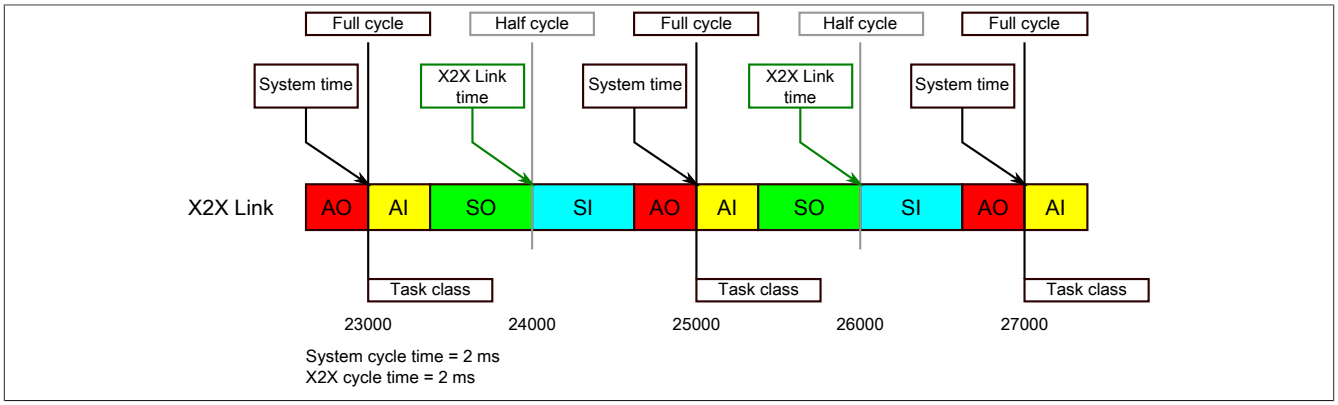
The initialization of the times is based on the system time during the startup of the X2X Link, the I/O modules or the POWERLINK interface.

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

4.6.1.1 Controller data points

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

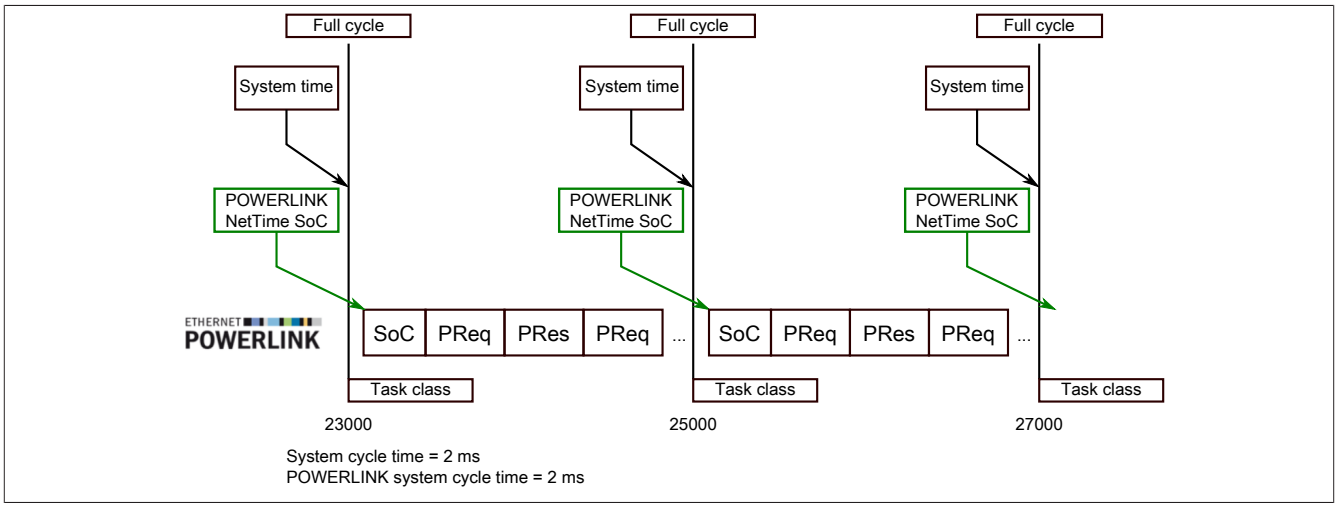
4.6.1.2 X2X Link - Reference time point



The reference time point 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 time point 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 time are compared at time 25000 in the task, then the system time returns the value 25000 and the X2X Link reference time returns the value 24000.

4.6.1.3 POWERLINK - Reference time point

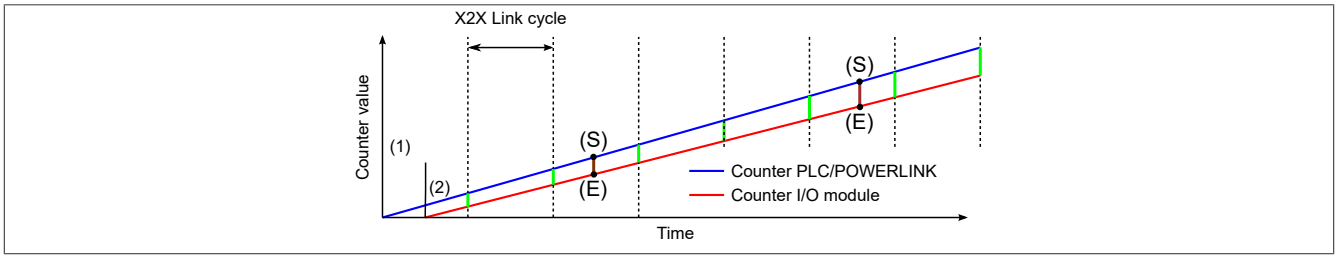


The POWERLINK reference time point is always calculated at the start of cycle (SoC) of the POWERLINK network. The SoC starts 20 µs after the system clock due to the system. This results in the following difference between the system time and the POWERLINK reference time:

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 time are compared at time 25000 in the task, then the system time returns the value 25000 and the POWERLINK reference time returns the value 23020.

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



At startup, the internal counters for the controller/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 controller or 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.

4.6.2 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 controller, including this precise moment, the controller can then evaluate the data using its own NetTime (or system time), if necessary.
For details, see the respective module documentation.

4.6.2.1 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.

4.6.2.2 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.

4.6.2.3 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.

4.7 Flatstream communication

4.7.1 Introduction

B&R offers an additional communication method for some modules. "Flatstream" was designed for X2X and POWERLINK networks and allows data transfer to be adapted to individual demands. Although this method is not 100% real-time capable, it still allows data transfer to be handled more efficiently than with standard cyclic polling.

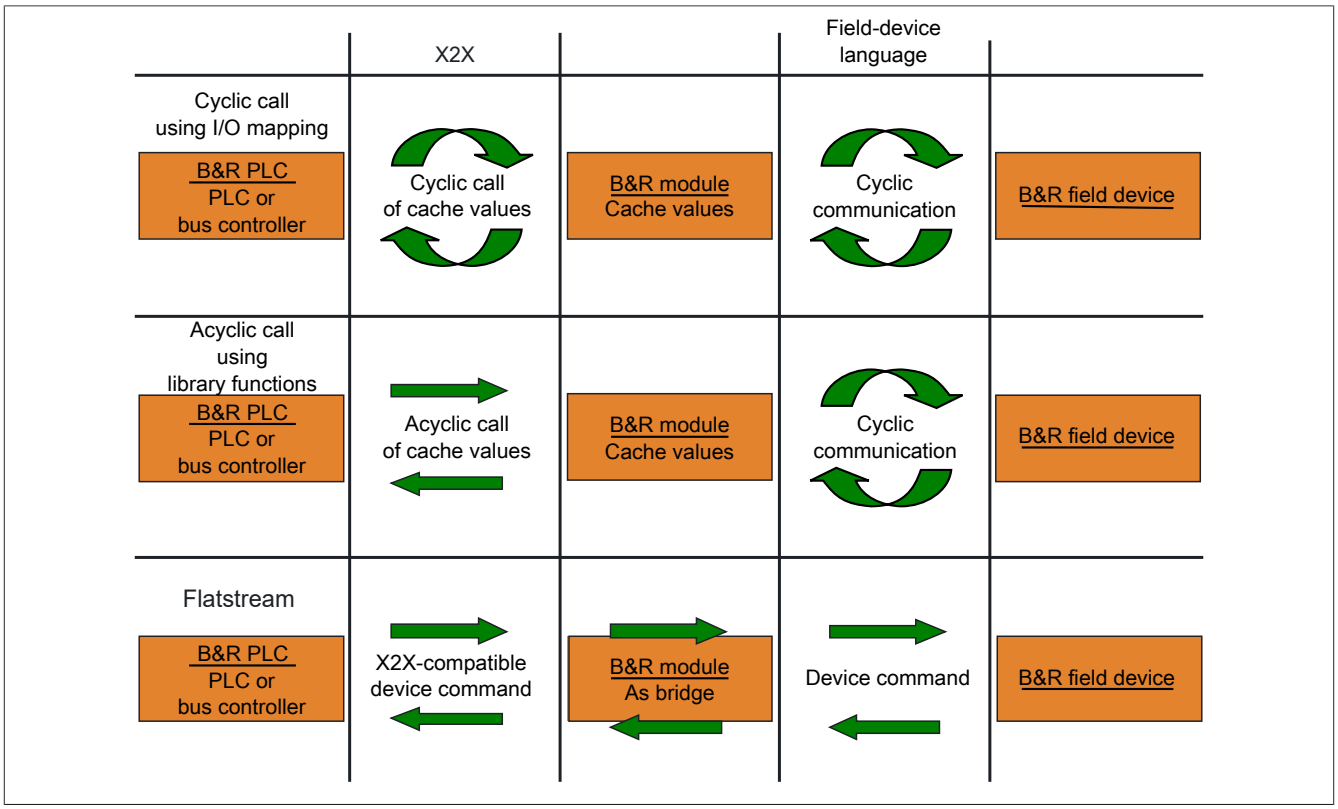


Figure 4: 3 types of communication

Flatstream extends cyclic and acyclic data queries. With Flatstream communication, the module acts as a bridge. The module is used to pass controller requests directly on to the field device.

4.7.2 Message, segment, sequence, MTU

The physical properties of the bus system limit the amount of data that can be transmitted during one bus cycle. With Flatstream communication, all messages are viewed as part of a continuous data stream. Long data streams must be broken down into several fragments that are sent one after the other. To understand how the receiver puts these fragments back together to get the original information, it is important to understand the difference between a message, a segment, a sequence and an MTU.

Message

A message refers to information exchanged between 2 communicating partner stations. The length of a message is not restricted by the Flatstream communication method. Nevertheless, module-specific limitations must be considered.

Segment (logical division of a message):

A segment has a finite size and can be understood as a section of a message. The number of segments per message is arbitrary. So that the recipient can correctly reassemble the transferred segments, each segment is preceded by a byte with additional information. This control byte contains information such as the length of a segment and whether the approaching segment completes the message. This makes it possible for the receiving station to interpret the incoming data stream correctly.

Sequence (how a segment must be arranged physically):

The maximum size of a sequence corresponds to the number of enabled Rx or Tx bytes (later: "MTU"). The transmitting station splits the transmit array into valid sequences. These sequences are then written successively to the MTU and transferred to the receiving station where they are lined up together again. The receiver stores the incoming sequences in a receive array, obtaining an image of the data stream in the process.
With Flatstream communication, the number of sequences sent are counted. Successfully transferred sequences must be acknowledged by the receiving station to ensure the integrity of the transfer.

MTU (Maximum Transmission Unit) - Physical transport:

MTU refers to the enabled USINT registers used with Flatstream. These registers can accept at least one sequence and transfer it to the receiving station. A separate MTU is defined for each direction of communication. OutputMTU defines the number of Flatstream Tx bytes, and InputMTU specifies the number of Flatstream Rx bytes. The MTUs are transported cyclically via the X2X Link network, increasing the load with each additional enabled USINT register.

Properties

Flatstream messages are not transferred cyclically or in 100% real time. Many bus cycles may be needed to transfer a particular message. Although the Rx and Tx registers are exchanged between the transmitter and the receiver cyclically, they are only processed further if explicitly accepted by register "InputSequence" or "OutputSequence".

Behavior in the event of an error (brief summary)

The protocol for X2X and POWERLINK networks specifies that the last valid values should be retained when disturbances occur. With conventional communication (cyclic/acyclic data queries), this type of error can generally be ignored.
In order for communication to also take place without errors using Flatstream, all of the sequences issued by the receiver must be acknowledged. If Forward functionality is not used, then subsequent communication is delayed for the length of the disturbance.
If Forward functionality is being used, the receiving station receives a transmission counter that is incremented twice. The receiver stops, i.e. it no longer returns any acknowledgments. The transmitting station uses SequenceAck to determine that the transfer was faulty and that all affected sequences must be repeated.

4.7.3 The Flatstream principle

Requirements

Before Flatstream can be used, the respective communication direction must be synchronized, i.e. both communication partners cyclically query the sequence counter on the remote station. This checks to see if there is new data that should be accepted.

Communication

If a communication partner wants to transmit a message to its remote station, it should first create a transmit array that corresponds to Flatstream conventions. This allows the Flatstream data to be organized very efficiently without having to block other important resources.

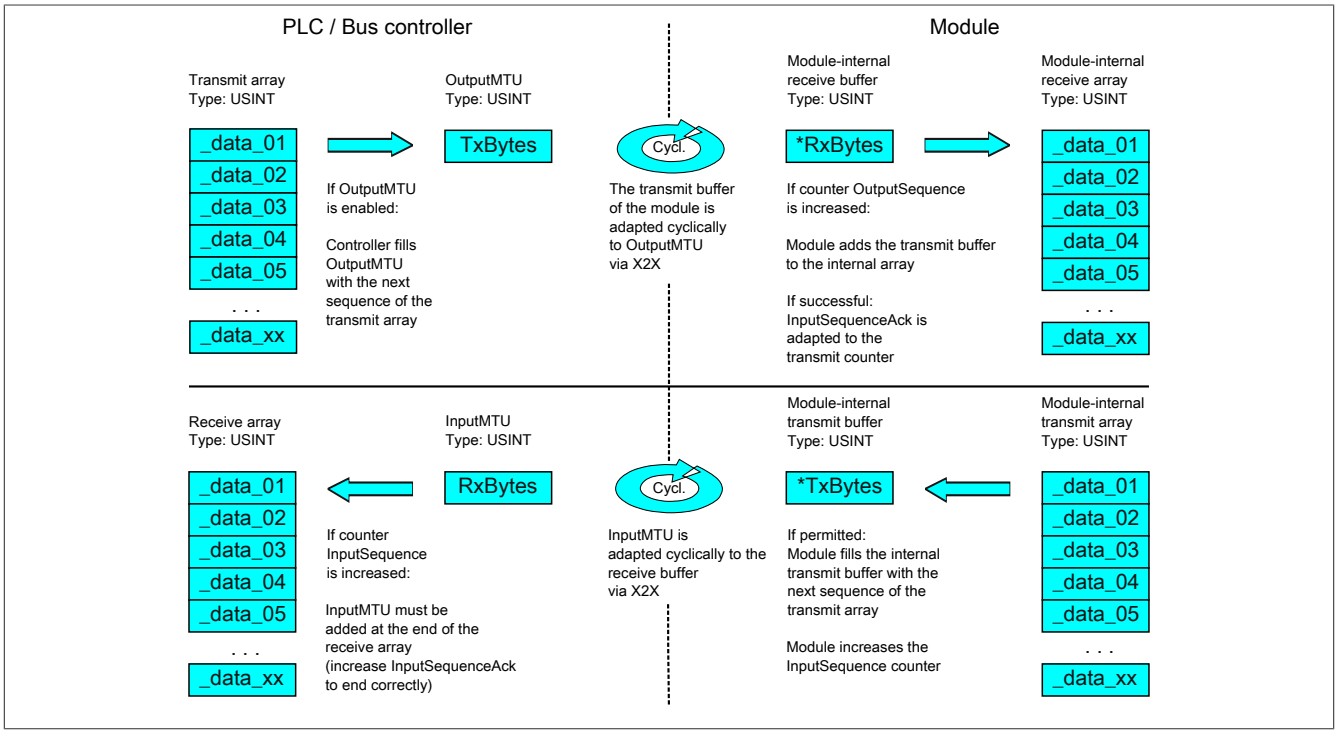


Figure 5: Flatstream communication

Procedure

The first thing that happens is that the message is broken into valid segments of up to 63 bytes, and the corresponding control bytes are created. The data is formed into a data stream made up of one control bytes per associated segment. This data stream can be written to the transmit array. The maximum size of each array element matches that of the enabled MTU so that one element corresponds to one sequence.
If the array has been completely created, the transmitter checks whether the MTU is permitted to be refilled. It then copies the first element of the array or the first sequence to the Tx byte registers. The MTU is transported to the receiver station via X2X Link and stored in the corresponding Rx byte registers. To signal that the data should be accepted by the receiver, the transmitter increases its SequenceCounter.
If the communication direction is synchronized, the remote station detects the incremented SequenceCounter. The current sequence is appended to the receive array and acknowledged by SequenceAck. This acknowledgment signals to the transmitter that the MTU can now be refilled.

If the transfer is successful, the data in the receive array will correspond 100% to the data in the transmit array. During the transfer, the receiving station must detect and evaluate the incoming control bytes. A separate receive array should be created for each message. This allows the receiver to immediately begin further processing of messages that are completely transferred.

4.7.4 Registers for Flatstream mode

5 registers are available for configuring Flatstream. The default configuration can be used to transmit small amounts of data relatively easily.

Information:

The controller communicates directly with the field device via registers "OutputSequence" and "InputSequence" as well as the enabled Tx and RxBytes bytes. For this reason, the user must have sufficient knowledge of the communication protocol being used on the field device.

4.7.4.1 Flatstream configuration

To use Flatstream, the program sequence must first be expanded. The cycle time of the Flatstream routines must be set to a multiple of the bus cycle. Other program routines should be implemented in Cyclic #1 to ensure data consistency.

At the absolute minimum, registers "InputMTU" and "OutputMTU" must be set. All other registers are filled in with default values at the beginning and can be used immediately. These registers are used for additional options, e.g. to transfer data in a more compact way or to increase the efficiency of the general procedure.

The Forward registers extend the functionality of the Flatstream protocol. This functionality is useful for substantially increasing the Flatstream data rate, but it also requires quite a bit of extra work when creating the program sequence.

Information:

In the rest of this description, the names "OutputMTU" and "InputMTU" do not refer to the registers names. Instead, they are used as synonyms for the currently enabled Tx or Rx bytes.

Information:

The registers are described in "Flatstream registers" on page 106.

Registers are described in section "Flatstream communication" in the respective data sheets.

4.7.4.2 Flatstream operation

When using Flatstream, the communication direction is very important. For transmitting data to a module (output direction), Tx bytes are used. For receiving data from a module (input direction), Rx bytes are used. Registers "OutputSequence" and "InputSequence" are used to control or secure communication, i.e. the transmitter uses them to give instructions to apply data and the receiver confirms a successfully transferred sequence.

Information:

The registers are described in "Flatstream registers" on page 106.

Registers are described in section "Flatstream communication" in the respective data sheets.

4.7.4.2.1 Format of input and output bytes

Name:

"Format of Flatstream" in Automation Studio

On some modules, this function can be used to set how the Flatstream input and output bytes (Tx or Rx bytes) are transferred.

- **Packed:** Data is transferred as an array.
- **Byte-by-byte:** Data is transferred as individual bytes.

4.7.4.2.2 Transporting payload data and control bytes

The Tx and Rx bytes are cyclic registers used to transport the payload data and the necessary control bytes. The number of active Tx and Rx bytes is taken from the configuration of registers "OutputMTU" and "InputMTU", respectively.
In the user program, only the Tx and Rx bytes from the controller can be used. The corresponding counterparts are located in the module and are not accessible to the user. For this reason, the names were chosen from the point of view of the controller.

- "T" - "Transmit" → Controller transmits data to the module.
- "R" - "Receive" → Controller receives data from the module.

4.7.4.2.2.1 Control bytes

In addition to the payload data, the Tx and Rx bytes also transfer the necessary control bytes. These control bytes contain additional information about the data stream so that the receiver can reconstruct the original message from the transferred segments.

Bit structure of a control byte

Bit	Name	Value	Information
0 - 5	SegmentLength	0 - 63	Size of the subsequent segment in bytes (default: Max. MTU size - 1)
6	nextCBPos	0	Next control byte at the beginning of the next MTU
		1	Next control byte directly after the end of the current segment
7	MessageEndBit	0	Message continues after the subsequent segment
		1	Message ended by the subsequent segment

SegmentLength

The segment length lets the receiver know the length of the coming segment. If the set segment length is insufficient for a message, then the information must be distributed over several segments. In these cases, the actual end of the message is detected using bit 7 (control byte).

Information:

The control byte is not included in the calculation to determine the segment length. The segment length is only derived from the bytes of payload data.

nextCBPos

This bit indicates the position where the next control byte is expected. This information is especially important when using option "MultiSegmentMTU".
When using Flatstream communication with MultiSegmentMTUs, the next control byte is no longer expected in the first Rx byte of the subsequent MTU, but transferred directly after the current segment.

MessageEndBit

"MessageEndBit" is set if the subsequent segment completes a message. The message has then been completely transferred and is ready for further processing.

Information:

In the output direction, this bit must also be set if one individual segment is enough to hold the entire message. The module will only process a message internally if this identifier is detected.
The size of the message being transferred can be calculated by adding all of the message's segment lengths together.

Flatstream formula for calculating message length:

Message [bytes] = Segment lengths (all CBs without ME) + Segment length (of the first CB with ME)	CB	Control byte
	ME	MessageEndBit

4.7.4.2.3 Communication status

The communication status is determined via registers "OutputSequence" and "InputSequence".

- OutputSequence contains information about the communication status of the controller. It is written by the controller and read by the module.
- InputSequence contains information about the communication status of the module. It is written by the module and should only be read by the controller.

4.7.4.2.3.1 Relationship between OutputSequence and InputSequence

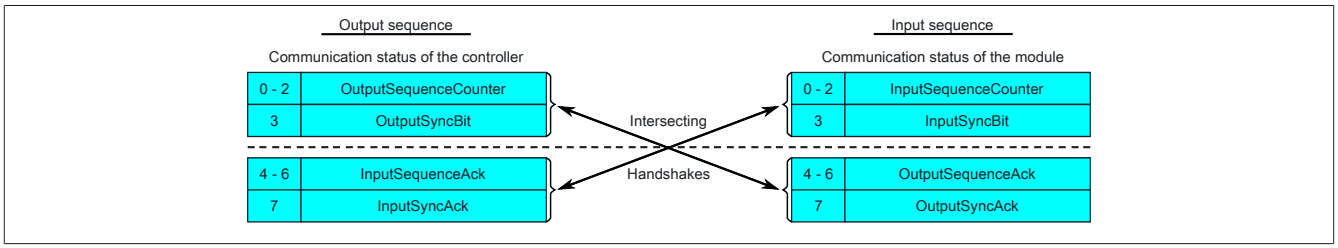


Figure 6: Relationship between OutputSequence and InputSequence

Registers OutputSequence and InputSequence are logically composed of 2 half-bytes. The low part indicates to the remote station whether a channel should be opened or whether data should be accepted. The high part is to acknowledge that the requested action was carried out.

SyncBit and SyncAck

If SyncBit and SyncAck are set in one communication direction, then the channel is considered "synchronized", i.e. it is possible to send messages in this direction. The status bit of the remote station must be checked cyclically. If SyncAck has been reset, then SyncBit on that station must be adjusted. Before new data can be transferred, the channel must be resynchronized.

SequenceCounter and SequenceAck

The communication partners cyclically check whether the low nibble on the remote station changes. When one of the communication partners finishes writing a new sequence to the MTU, it increments its SequenceCounter. The current sequence is then transmitted to the receiver, which acknowledges its receipt with SequenceAck. In this way, a "handshake" is initiated.

Information:

If communication is interrupted, segments from the unfinished message are discarded. All messages that were transferred completely are processed.

4.7.4.3 Synchronization

During synchronization, a communication channel is opened. It is important to make sure that a module is present and that the current value of SequenceCounter is stored on the station receiving the message. Flatstream can handle full-duplex communication. This means that both channels / communication directions can be handled separately. They must be synchronized independently so that simplex communication can theoretically be carried out as well.

Synchronization in the output direction (controller as the transmitter):

The corresponding synchronization bits (OutputSyncBit and OutputSyncAck) are reset. Because of this, Flatstream cannot be used at this point in time to transfer messages from the controller to the module.

Algorithm

1) The controller must write 000 to OutputSequenceCounter and reset OutputSyncBit. The controller must cyclically query the high nibble of register "InputSequence" (checks for 000 in OutputSequenceAck and 0 in OutputSyncAck).
The module does not accept the current contents of InputMTU since the channel is not yet synchronized. The module matches OutputSequenceAck and OutputSyncAck to the values of OutputSequenceCounter and OutputSyncBit.
2) If the controller registers the expected values in OutputSequenceAck and OutputSyncAck, it is permitted to increment OutputSequenceCounter. The controller continues cyclically querying the high nibble of register "OutputSequence" (checks for 001 in OutputSequenceAck and 0 in InputSyncAck).
The module does not accept the current contents of InputMTU since the channel is not yet synchronized. The module matches OutputSequenceAck and OutputSyncAck to the values of OutputSequenceCounter and OutputSyncBit.
3) If the controller registers the expected values in OutputSequenceAck and OutputSyncAck, it is permitted to increment OutputSequenceCounter. The controller continues cyclically querying the high nibble of register "OutputSequence" (checks for 001 in OutputSequenceAck and 1 in InputSyncAck). **Note:** Theoretically, data can be transferred from this point forward. However, it is still recommended to wait until the output direction is completely synchronized before transferring data.
The module sets OutputSyncAck.
The output direction is synchronized, and the controller can transmit data to the module.

Synchronization in the input direction (controller as the receiver):

The corresponding synchronization bits (InputSyncBit and InputSyncAck) are reset. Because of this, Flatstream cannot be used at this point in time to transfer messages from the module to the controller.

Algorithm

The module writes 000 to InputSequenceCounter and resets InputSyncBit. The module monitors the high nibble of register "OutputSequence" and expects 000 in InputSequenceAck and 0 in InputSyncAck.
1) The controller is not permitted to accept the current contents of InputMTU since the channel is not yet synchronized. The controller must match InputSequenceAck and InputSyncAck to the values of InputSequenceCounter and InputSyncBit.
If the module registers the expected values in InputSequenceAck and InputSyncAck, it increments InputSequenceCounter. The module monitors the high nibble of register "OutputSequence" and expects 001 in InputSequenceAck and 0 in InputSyncAck.
2) The controller is not permitted to accept the current contents of InputMTU since the channel is not yet synchronized. The controller must match InputSequenceAck and InputSyncAck to the values of InputSequenceCounter and InputSyncBit.
If the module registers the expected values in InputSequenceAck and InputSyncAck, it sets InputSyncBit. The module monitors the high nibble of register "OutputSequence" and expects 1 in InputSyncAck.
3) The controller is permitted to set InputSyncAck. **Note:** Theoretically, data could already be transferred in this cycle. If InputSyncBit is set and InputSequenceCounter has been increased by 1, the values in the enabled Rx bytes must be accepted and acknowledged (see also "Communication in the input direction").
The input direction is synchronized, and the module can transmit data to the controller.

4.7.4.4 Transmitting and receiving

If a channel is synchronized, then the remote station is ready to receive messages from the transmitter. Before the transmitter can send data, it must first create a transmit array in order to meet Flatstream requirements.

The transmitting station must also generate a control byte for each segment created. This control byte contains information about how the subsequent part of the data being transferred should be processed. The position of the next control byte in the data stream can vary. For this reason, it must be clearly defined at all times when a new control byte is being transmitted. The first control byte is always in the first byte of the first sequence. All subsequent positions are determined recursively.

Flatstream formula for calculating the position of the next control byte:

Position (of the next control byte) = Current position + 1 + Segment length

Example

3 autonomous messages (7 bytes, 2 bytes and 9 bytes) are being transmitted using an MTU with a width of 7 bytes. The rest of the configuration corresponds to the default settings.

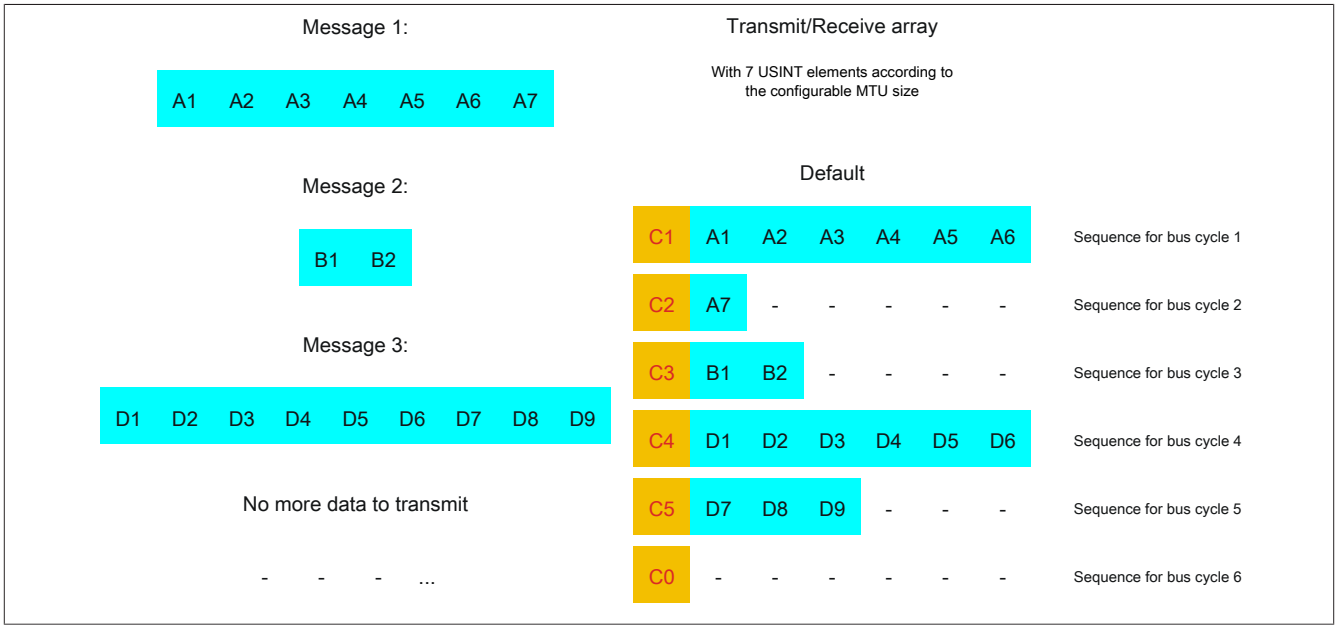


Figure 7: Transmit/Receive array (default)

The messages must first be split into segments. In the default configuration, it is important to ensure that each sequence can hold an entire segment, including the associated control byte. The sequence is limited to the size of the enable MTU. In other words, a segment must be at least 1 byte smaller than the MTU.

MTU = 7 bytes → Max. segment length = 6 bytes

- Message 1 (7 bytes)
  - ⇨ First segment = Control byte + 6 bytes of data
  - ⇨ Second segment = Control byte + 1 data byte
- Message 2 (2 bytes)
  - ⇨ First segment = Control byte + 2 bytes of data
- Message 3 (9 bytes)
  - ⇨ First segment = Control byte + 6 bytes of data
  - ⇨ Second segment = Control byte + 3 data bytes
- No more messages
  - ⇨ C0 control byte

A unique control byte must be generated for each segment. In addition, the C0 control byte is generated to keep communication on standby.

C0 (control byte 0)			C1 (control byte 1)			C2 (control byte 2)		
- SegmentLength (0)	=	0	- SegmentLength (6)	=	6	- SegmentLength (1)	=	1
- nextCBPos (0)	=	0	- nextCBPos (0)	=	0	- nextCBPos (0)	=	0
- MessageEndBit (0)	=	0	- MessageEndBit (0)	=	0	- MessageEndBit (1)	=	128
Control byte	Σ	0	Control byte	Σ	6	Control byte	Σ	129

Table 5: Flatstream determination of the control bytes for the default configuration example (part 1)

C3 (control byte 3)			C4 (control byte 4)			C5 (control byte 5)		
- SegmentLength (2)	=	2	- SegmentLength (6)	=	6	- SegmentLength (3)	=	3
- nextCBPos (0)	=	0	- nextCBPos (0)	=	0	- nextCBPos (0)	=	0
- MessageEndBit (1)	=	128	- MessageEndBit (0)	=	0	- MessageEndBit (1)	=	128
Control byte	Σ	130	Control byte	Σ	6	Control byte	Σ	131

Table 6: Flatstream determination of the control bytes for the default configuration example (part 2)

4.7.4.4.1 Transmitting data to a module (output)

When transmitting data, the transmit array must be generated in the application program. Sequences are then transferred one by one using Flatstream and received by the module.

Information:

Although all B&R modules with Flatstream communication always support the most compact transfers in the output direction, it is recommended to use the same design for the transfer arrays in both communication directions.

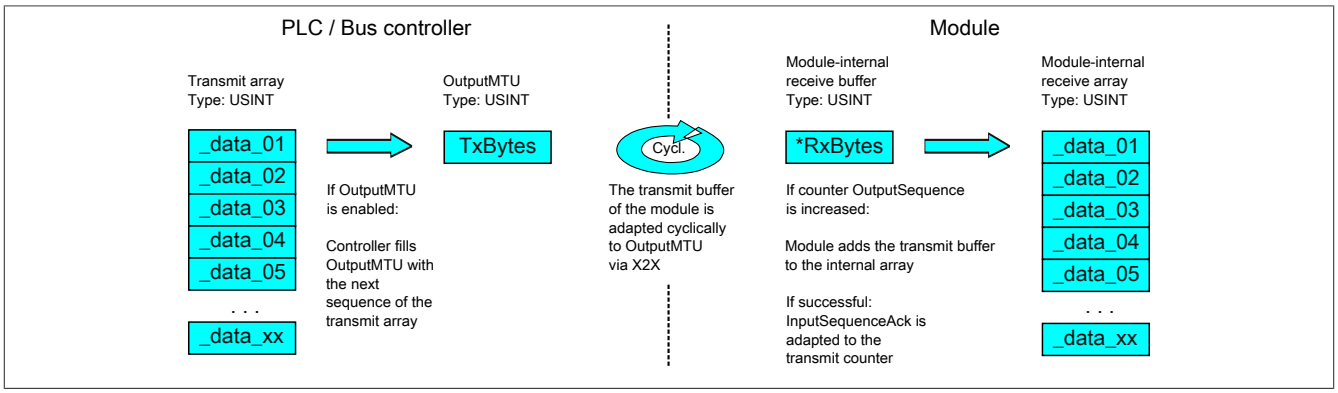


Figure 8: Flatstream communication (output)

Message smaller than OutputMTU

The length of the message is initially smaller than OutputMTU. In this case, one sequence would be sufficient to transfer the entire message and the necessary control byte.

Algorithm

Cyclic status query: - The module monitors OutputSequenceCounter.
0) Cyclic checks: - The controller must check OutputSyncAck. → If OutputSyncAck = 0: Reset OutputSyncBit and resynchronize the channel. - The controller must check whether OutputMTU is enabled. → If OutputSequenceCounter > InputSequenceAck: MTU is not enabled because the last sequence has not yet been acknowledged.
1) Preparation (create transmit array): - The controller must split up the message into valid segments and create the necessary control bytes. - The controller must add the segments and control bytes to the transmit array.
2) Transmit: - The controller transfers the current element of the transmit array to OutputMTU. → OutputMTU is transferred cyclically to the module's transmit buffer but not processed further. - The controller must increase OutputSequenceCounter.
Reaction: - The module accepts the bytes from the internal receive buffer and adds them to the internal receive array. - The module transmits acknowledgment and writes the value of OutputSequenceCounter to OutputSequenceAck.
3) Completion: - The controller must monitor OutputSequenceAck. → A sequence is only considered to have been transferred successfully if it has been acknowledged via OutputSequenceAck. In order to detect potential transfer errors in the last sequence as well, it is important to make sure that the length of the Completion phase is run through long enough. **Note:** To monitor communication times exactly, the task cycles that have passed since the last increase of OutputSequenceCounter should be counted. In this way, the number of previous bus cycles necessary for the transfer can be measured. If the monitoring counter exceeds a predefined threshold, then the sequence can be considered lost. (The relationship of bus to task cycle can be influenced by the user so that the threshold value must be determined individually.) - Subsequent sequences are only permitted to be transmitted in the next bus cycle after the completion check has been carried out successfully.

Message larger than OutputMTU

The transmit array, which must be created in the program sequence, consists of several elements. The user must arrange the control and data bytes correctly and transfer the array elements one after the other. The transfer algorithm remains the same and is repeated starting at the point Cyclic checks.

General flowchart

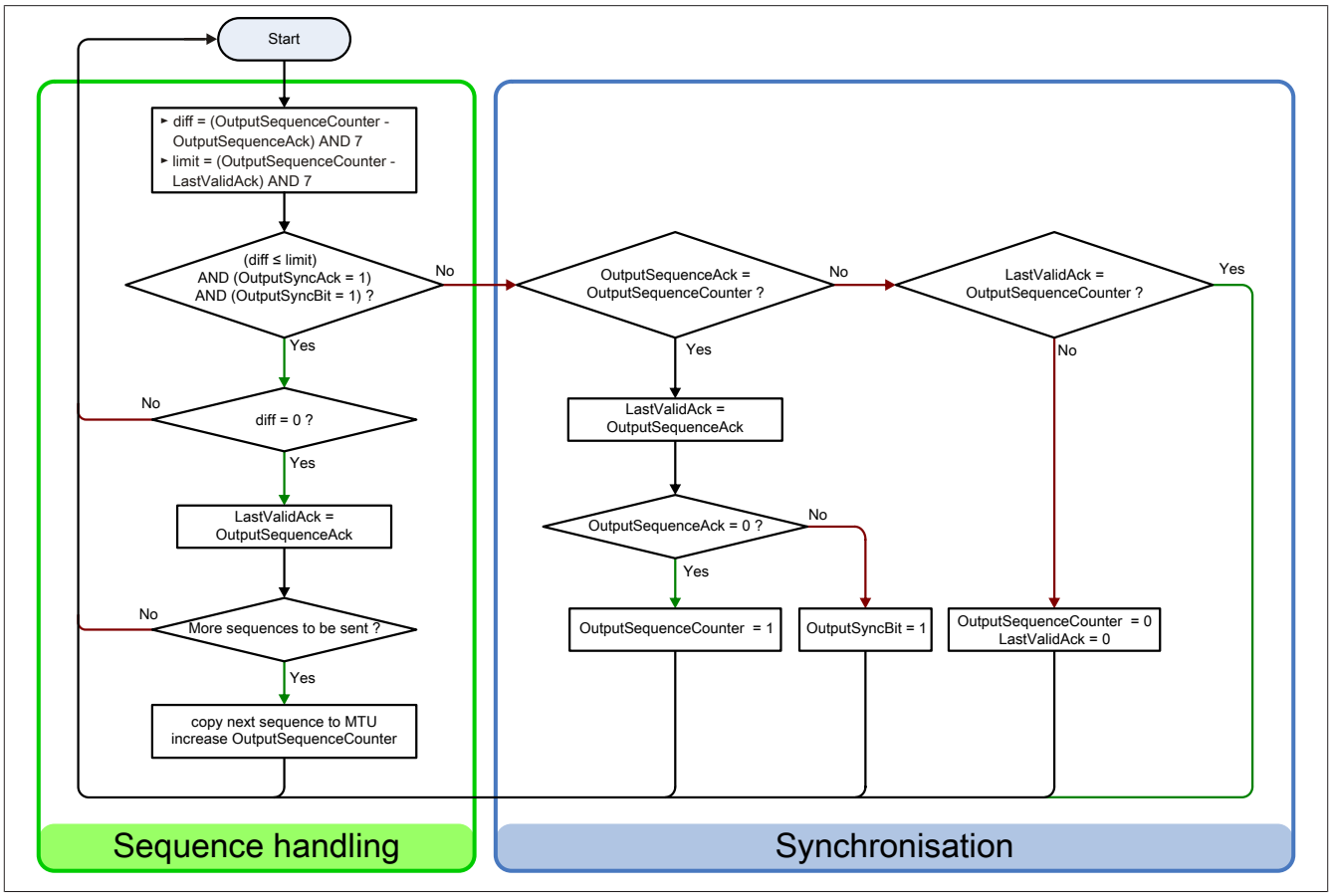


Figure 9: Flowchart for the output direction

4.7.4.4.2 Receiving data from a module (input)

When receiving data, the transmit array is generated by the module, transferred via Flatstream and must then be reproduced in the receive array. The structure of the incoming data stream can be set with the mode register. The algorithm for receiving the data remains unchanged in this regard.

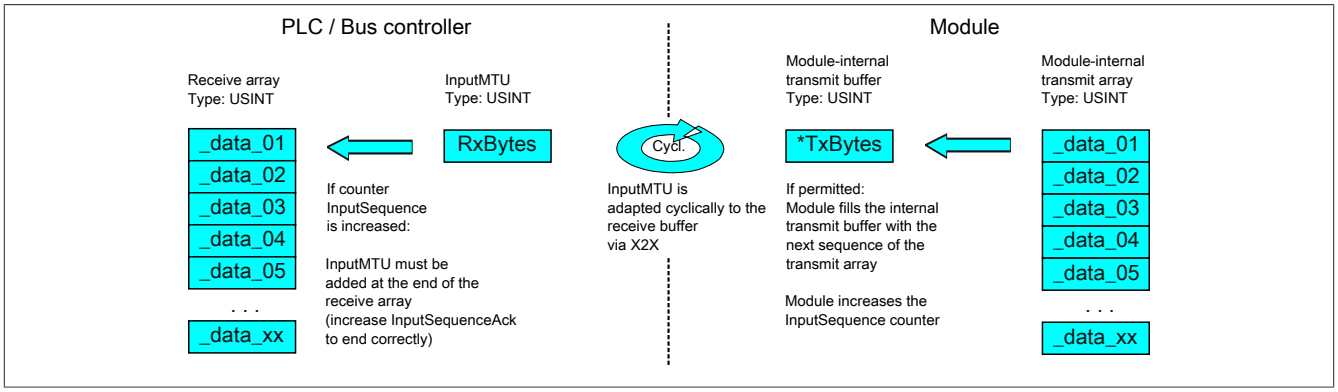


Figure 10: Flatstream communication (input)

Algorithm

0) Cyclic status query: **- The controller must monitor InputSequenceCounter.**
Cyclic checks: - The module checks InputSyncAck. - The module checks InputSequenceAck.
Preparation: - The module forms the segments and control bytes and creates the transmit array.
Action: - The module transfers the current element of the internal transmit array to the internal transmit buffer. - The module increases InputSequenceCounter.
1) Receiving (as soon as InputSequenceCounter is increased): **- The controller must apply data from InputMTU and append it to the end of the receive array.** **- The controller must match InputSequenceAck to InputSequenceCounter of the sequence currently being processed.**
Completion: - The module monitors InputSequenceAck. → A sequence is only considered to have been transferred successfully if it has been acknowledged via InputSequenceAck. - Subsequent sequences are only transmitted in the next bus cycle after the completion check has been carried out successfully.

General flowchart

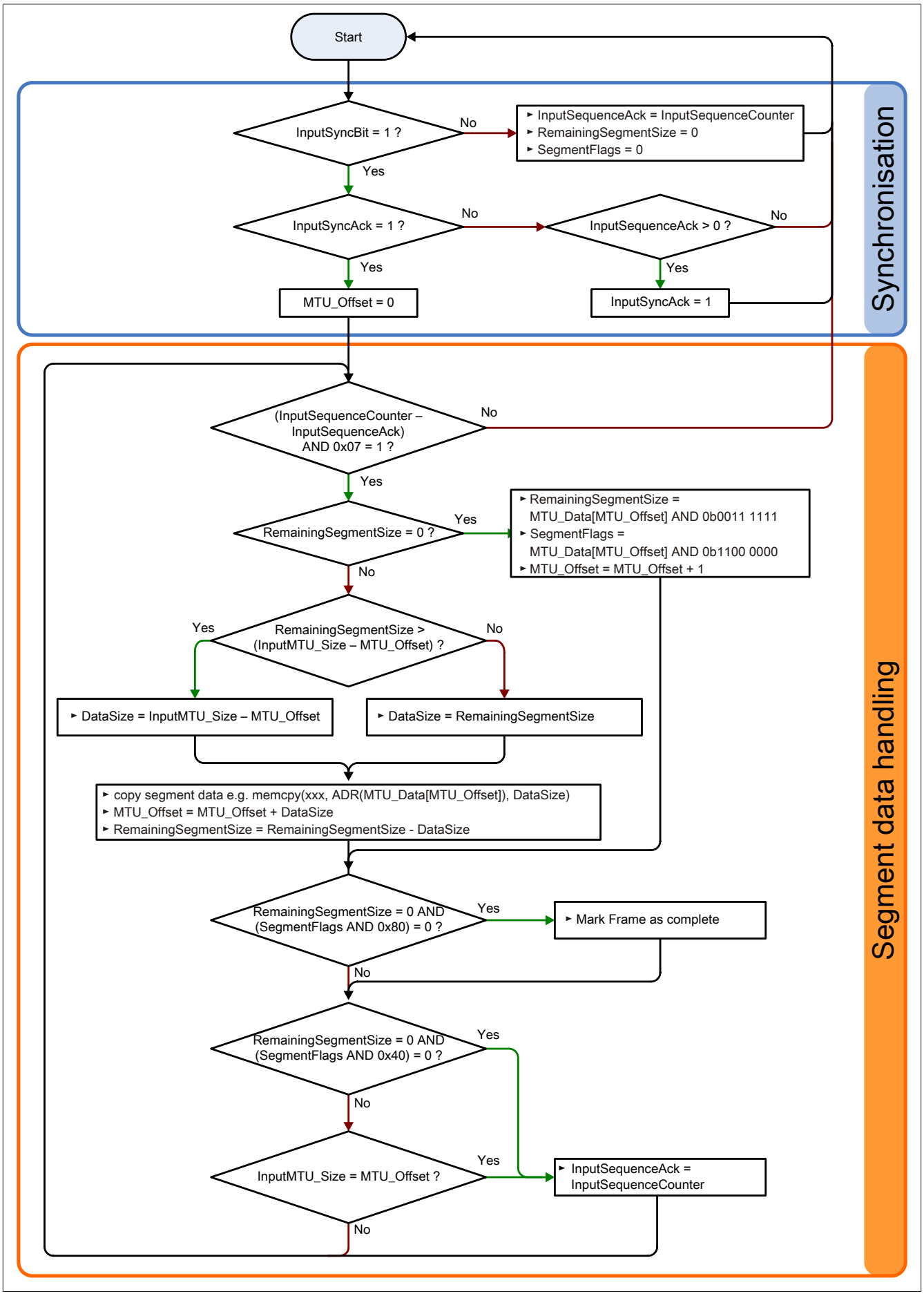


Figure 11: Flowchart for the input direction

4.7.4.4.3 Details

It is recommended to store transferred messages in separate receive arrays.

After a set MessageEndBit is transmitted, the subsequent segment should be added to the receive array. The message is then complete and can be passed on internally for further processing. A new/separate array should be created for the next message.

Information:

When transferring with MultiSegmentMTUs, it is possible for several small messages to be part of one sequence. In the program, it is important to make sure that a sufficient number of receive arrays can be managed. The acknowledge register is only permitted to be adjusted after the entire sequence has been applied.

If SequenceCounter is incremented by more than one counter, an error is present.

In this case, the receiver stops. All additional incoming sequences are ignored until the transmission with the correct SequenceCounter is retried. This response prevents the transmitter from receiving any more acknowledgments for transmitted sequences. The transmitter can identify the last successfully transferred sequence from the remote station's SequenceAck and continue the transfer from this point.

Information:

This situation is very unlikely when operating without "Forward" functionality.

Acknowledgments must be checked for validity.

If the receiver has successfully accepted a sequence, it must be acknowledged. The receiver takes on the value of SequenceCounter sent along with the transmission and matches SequenceAck to it. The transmitter reads SequenceAck and registers the successful transmission. If the transmitter acknowledges a sequence that has not yet been dispatched, then the transfer must be interrupted and the channel resynchronized. The synchronization bits are reset and the current/incomplete message is discarded. It must be sent again after the channel has been resynchronized.

4.7.4.5 Flatstream mode

In the input direction, the transmit array is generated automatically. Flatstream mode offers several options to the user that allow an incoming data stream to have a more compact arrangement. These include:

- Standard
- MultiSegmentMTU allowed
- Large segments allowed:

Once enabled, the program code for evaluation must be adapted accordingly.

Information:

All B&R modules that offer Flatstream mode support options "Large segments" and "MultiSegmentMTU" in the output direction. Compact transfer must be explicitly allowed only in the input direction.

Standard

By default, both options relating to compact transfer in the input direction are disabled.

1. The module only forms segments that are at least one byte smaller than the enabled MTU. Each sequence begins with a control byte so that the data stream is clearly structured and relatively easy to evaluate.
2. Since a Flatstream message is permitted to be any length, the last segment of the message frequently does not fill up all of the MTU's space. By default, the remaining bytes during this type of transfer cycle are not used.

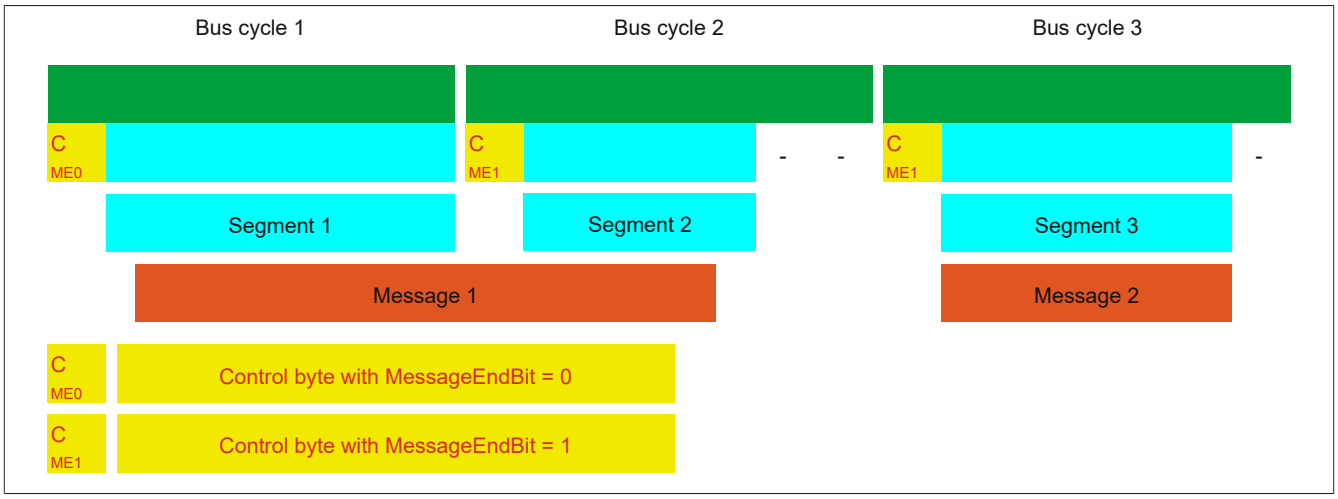


Figure 12: Message arrangement in the MTU (default)

MultiSegmentMTU allowed

With this option, InputMTU is completely filled (if enough data is pending). The previously unfilled Rx bytes transfer the next control bytes and their segments. This allows the enabled Rx bytes to be used more efficiently.

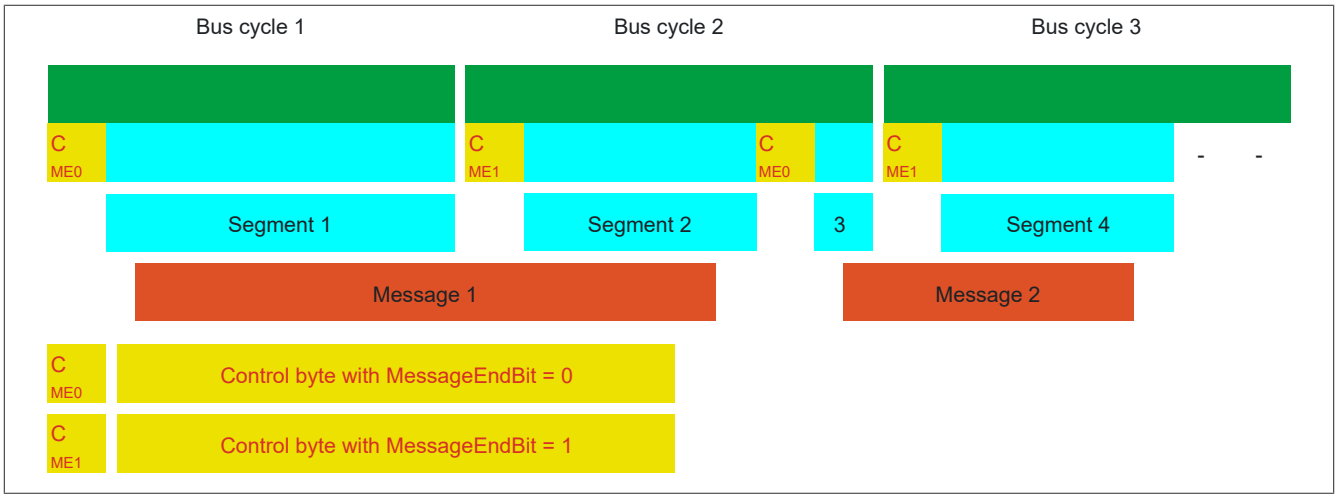


Figure 13: Arrangement of messages in the MTU (MultiSegmentMTU)

Large segments allowed:

When transferring very long messages or when enabling only very few Rx bytes, then a great many segments must be created by default. The bus system is more stressed than necessary since an additional control byte must be created and transferred for each segment. With option "Large segments", the segment length is limited to 63 bytes independently of InputMTU. One segment is permitted to stretch across several sequences, i.e. it is possible for "pure" sequences to occur without a control byte.

Information:

It is still possible to split up a message into several segments, however. If this option is used and messages with more than 63 bytes occur, for example, then messages can still be split up among several segments.

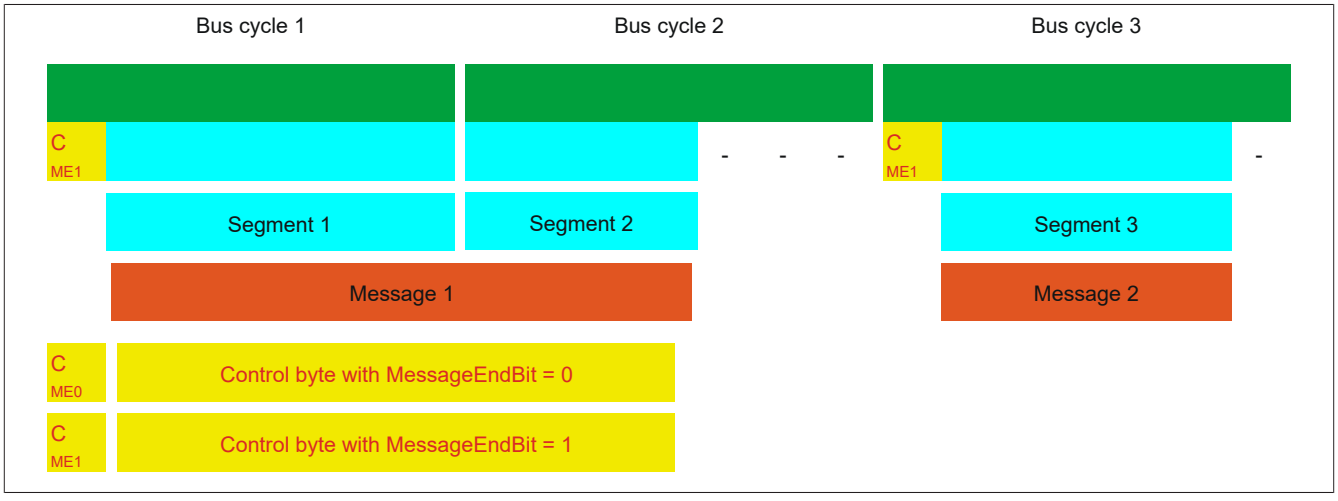


Figure 14: Arrangement of messages in the MTU (large segments)

Using both options

Using both options at the same time is also permitted.

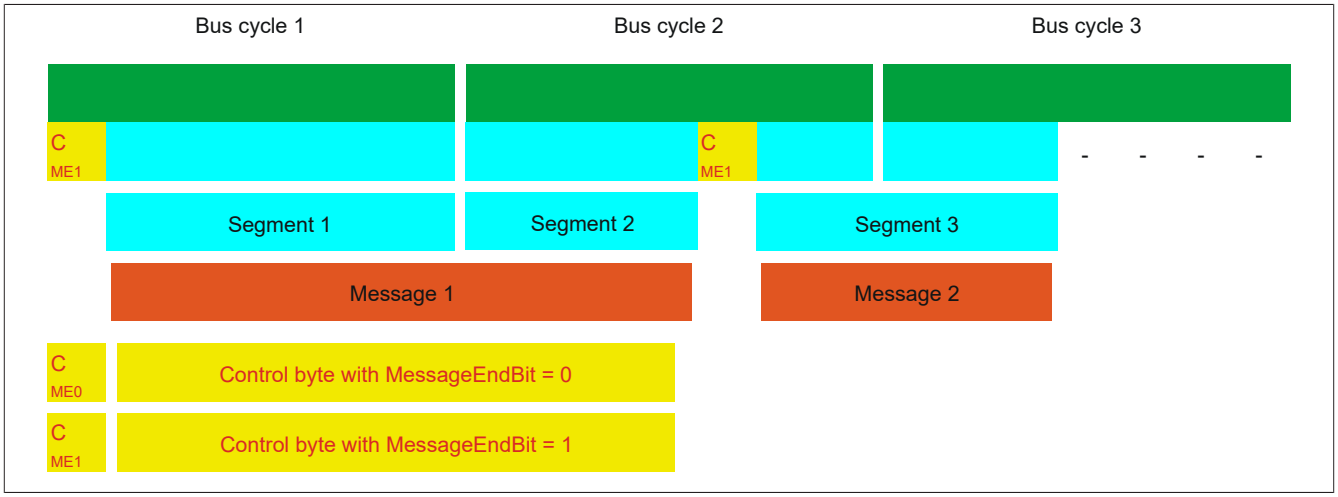


Figure 15: Arrangement of messages in the MTU (large segments and MultiSegmentMTU)

4.7.4.6 Adjusting the Flatstream

If the way messages are structured is changed, then the way data in the transmit/receive array is arranged is also different. The following changes apply to the example given earlier.

MultiSegmentMTU

If MultiSegmentMTUs are allowed, then "open positions" in an MTU can be used. These "open positions" occur if the last segment in a message does not fully use the entire MTU. MultiSegmentMTUs allow these bits to be used to transfer the subsequent control bytes and segments. In the program sequence, the "nextCBPos" bit in the control byte is set so that the receiver can correctly identify the next control byte.

Example

3 autonomous messages (7 bytes, 2 bytes and 9 bytes) are being transmitted using an MTU with a width of 7 bytes. The configuration allows the transfer of MultiSegmentMTUs.

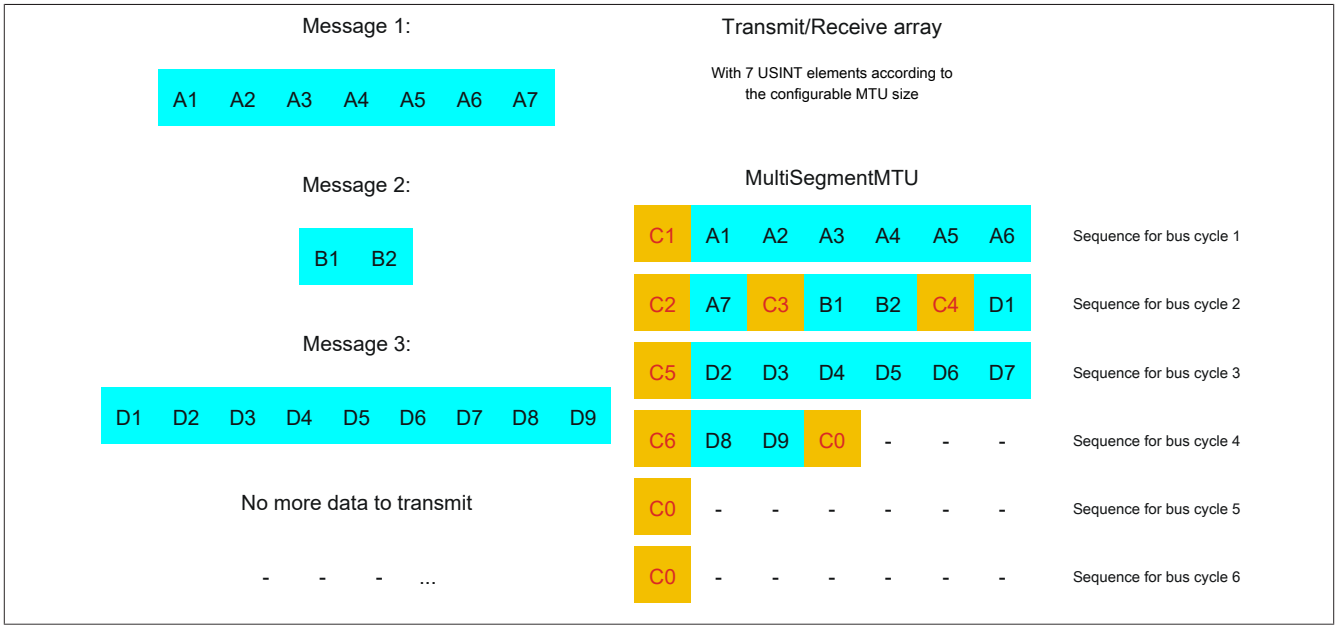


Figure 16: Transmit/Receive array (MultiSegmentMTU)

The messages must first be split into segments. As in the default configuration, it is important for each sequence to begin with a control byte. The free bits in the MTU at the end of a message are filled with data from the following message, however. With this option, the "nextCBPos" bit is always set if payload data is transferred after the control byte.

MTU = 7 bytes → Max. segment length = 6 bytes

- Message 1 (7 bytes)
  - ⇨ First segment = Control byte + 6 bytes of data (MTU full)
  - ⇨ Second segment = Control byte + 1 byte of data (MTU still has 5 open bytes)
- Message 2 (2 bytes)
  - ⇨ First segment = Control byte + 2 bytes of data (MTU still has 2 open bytes)
- Message 3 (9 bytes)
  - ⇨ First segment = Control byte + 1 byte of data (MTU full)
  - ⇨ Second segment = Control byte + 6 bytes of data (MTU full)
  - ⇨ Third segment = Control byte + 2 bytes of data (MTU still has 4 open bytes)
- No more messages
  - ⇨ C0 control byte

A unique control byte must be generated for each segment. In addition, the C0 control byte is generated to keep communication on standby.

C1 (control byte 1)			C2 (control byte 2)			C3 (control byte 3)		
- SegmentLength (6)	=	6	- SegmentLength (1)	=	1	- SegmentLength (2)	=	2
- nextCBPos (1)	=	64	- nextCBPos (1)	=	64	- nextCBPos (1)	=	64
- MessageEndBit (0)	=	0	- MessageEndBit (1)	=	128	- MessageEndBit (1)	=	128
Control byte	Σ	70	Control byte	Σ	193	Control byte	Σ	194

Table 7: Flatstream determination of the control bytes for the MultiSegmentMTU example (part 1)

Warning!

The second sequence is only permitted to be acknowledged via SequenceAck if it has been completely processed. In this example, there are 3 different segments within the second sequence, i.e. the program must include enough receive arrays to handle this situation.

Mise en garde !

La deuxième séquence ne peut être acquittée via SequenceAck que si elle a été entièrement traitée. Dans cet exemple, il y a 3 segments différents dans la deuxième séquence, c'est-à-dire que le programme doit inclure suffisamment de tableaux de réception pour gérer cette situation.

C4 (control byte 4)			C5 (control byte 5)			C6 (control byte 6)		
- SegmentLength (1)	=	1	- SegmentLength (6)	=	6	- SegmentLength (2)	=	2
- nextCBPos (6)	=	6	- nextCBPos (1)	=	64	- nextCBPos (1)	=	64
- MessageEndBit (0)	=	0	- MessageEndBit (1)	=	0	- MessageEndBit (1)	=	128
Control byte	Σ	7	Control byte	Σ	70	Control byte	Σ	194

Table 8: Flatstream determination of the control bytes for the MultiSegmentMTU example (part 2)

Large segments

Segments are limited to a maximum of 63 bytes. This means they can be larger than the active MTU. These large segments are divided among several sequences when transferred. It is possible for sequences to be completely filled with payload data and not have a control byte.

Information:

It is still possible to subdivide a message into several segments so that the size of a data packet does not also have to be limited to 63 bytes.

Example

3 autonomous messages (7 bytes, 2 bytes and 9 bytes) are being transmitted using an MTU with a width of 7 bytes. The configuration allows the transfer of large segments.

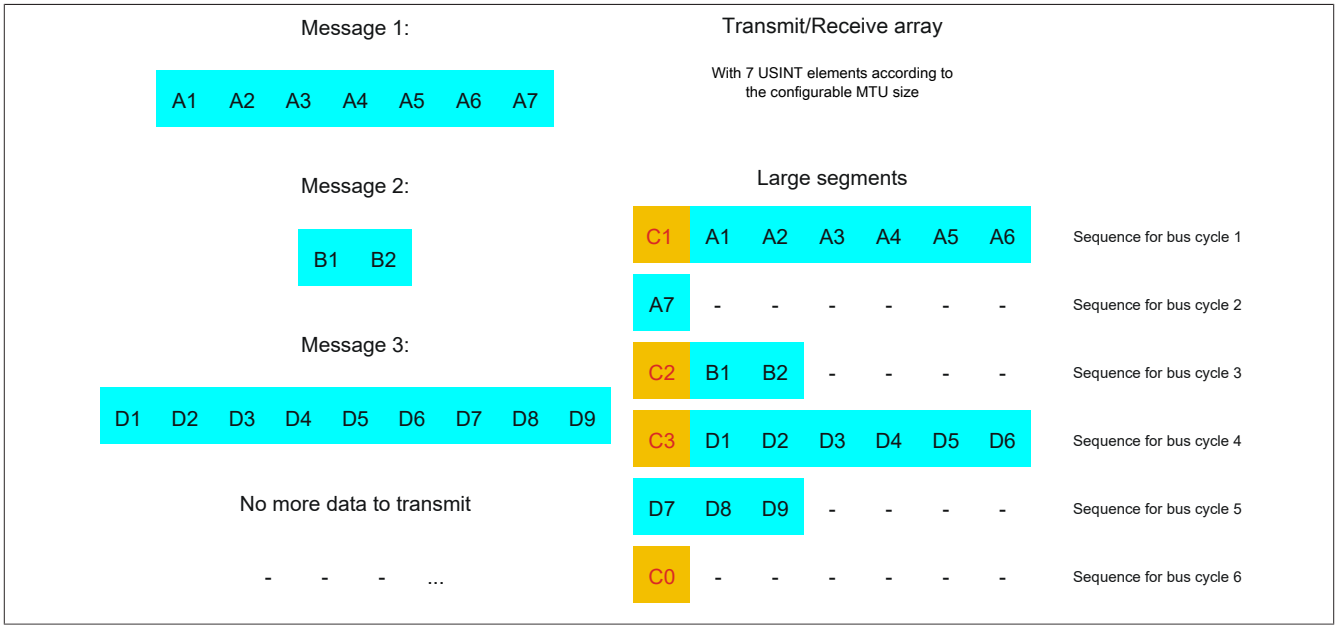


Figure 17: Transmit/receive array (large segments)

The messages must first be split into segments. The ability to form large segments means that messages are split up less frequently, which results in fewer control bytes generated.

Large segments allowed → Max. segment length = 63 bytes

- Message 1 (7 bytes)
  - ⇨ First segment = Control byte + 7 bytes of data
- Message 2 (2 bytes)
  - ⇨ First segment = Control byte + 2 bytes of data
- Message 3 (9 bytes)
  - ⇨ First segment = Control byte + 9 bytes of data
- No more messages
  - ⇨ C0 control byte

A unique control byte must be generated for each segment. In addition, the C0 control byte is generated to keep communication on standby.

C1 (control byte 1)			C2 (control byte 2)			C3 (control byte 3)		
- SegmentLength (7)	=	7	- SegmentLength (2)	=	2	- SegmentLength (9)	=	9
- nextCBPos (0)	=	0	- nextCBPos (0)	=	0	- nextCBPos (0)	=	0
- MessageEndBit (1)	=	128	- MessageEndBit (1)	=	128	- MessageEndBit (1)	=	128
Control byte	Σ	135	Control byte	Σ	130	Control byte	Σ	137

Table 9: Flatstream determination of the control bytes for the large segment example

Large segments and MultiSegmentMTU

Example

3 autonomous messages (7 bytes, 2 bytes and 9 bytes) are being transmitted using an MTU with a width of 7 bytes. The configuration allows transfer of large segments as well as MultiSegmentMTUs.

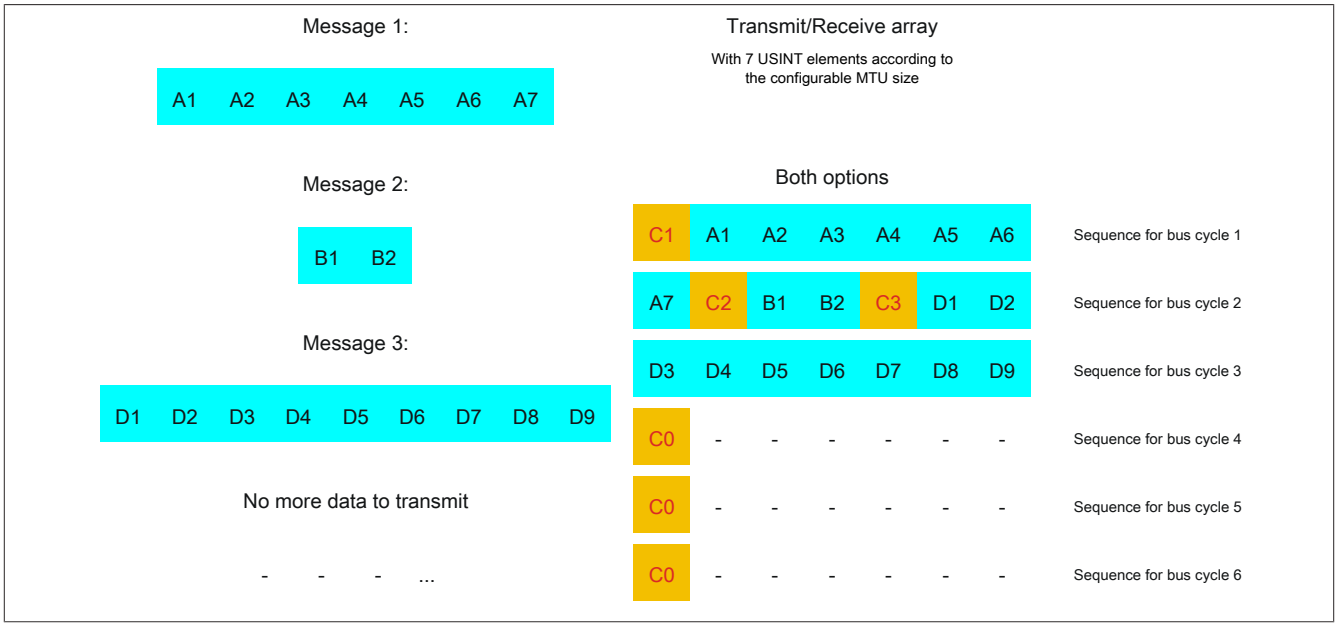


Figure 18: Transmit/Receive array (large segments and MultiSegmentMTU)

The messages must first be split into segments. If the last segment of a message does not completely fill the MTU, it is permitted to be used for other data in the data stream. Bit "nextCBPos" must always be set if the control byte belongs to a segment with payload data.
The ability to form large segments means that messages are split up less frequently, which results in fewer control bytes generated. Control bytes are generated in the same way as with option "Large segments".

Large segments allowed → Max. segment length = 63 bytes

- Message 1 (7 bytes)
  - ⇨ First segment = Control byte + 7 bytes of data
- Message 2 (2 bytes)
  - ⇨ First segment = Control byte + 2 bytes of data
- Message 3 (9 bytes)
  - ⇨ First segment = Control byte + 9 bytes of data
- No more messages
  - ⇨ C0 control byte

A unique control byte must be generated for each segment. In addition, the C0 control byte is generated to keep communication on standby.

C1 (control byte 1)			C2 (control byte 2)			C3 (control byte 3)		
- SegmentLength (7)	=	7	- SegmentLength (2)	=	2	- SegmentLength (9)	=	9
- nextCBPos (0)	=	0	- nextCBPos (0)	=	0	- nextCBPos (0)	=	0
- MessageEndBit (1)	=	128	- MessageEndBit (1)	=	128	- MessageEndBit (1)	=	128
Control byte	Σ	135	Control byte	Σ	130	Control byte	Σ	137

Table 10: Flatstream determination of the control bytes for the large segment and MultiSegmentMTU example

4.7.5 Example of function "Forward" with X2X Link

Function "Forward" is a method that can be used to substantially increase the Flatstream data rate. The basic principle is also used in other technical areas such as "pipelining" for microprocessors.

4.7.5.1 Function principle

X2X Link communication cycles through 5 different steps to transfer a Flatstream sequence. At least 5 bus cycles are therefore required to successfully transfer the sequence.

	Step I	Step II	Step III	Step IV	Step V
Actions	Transfer sequence from transmit array, increase Sequence-Counter	Cyclic synchronization of MTU and module buffer	Append sequence to receive array, adjust SequenceAck	Cyclic synchronization MTU and module buffer	Check SequenceAck
Resource	Transmitter (task to transmit)	Bus system (direction 1)	Recipients (task to receive)	Bus system (direction 2)	Transmitter (task for Ack checking)

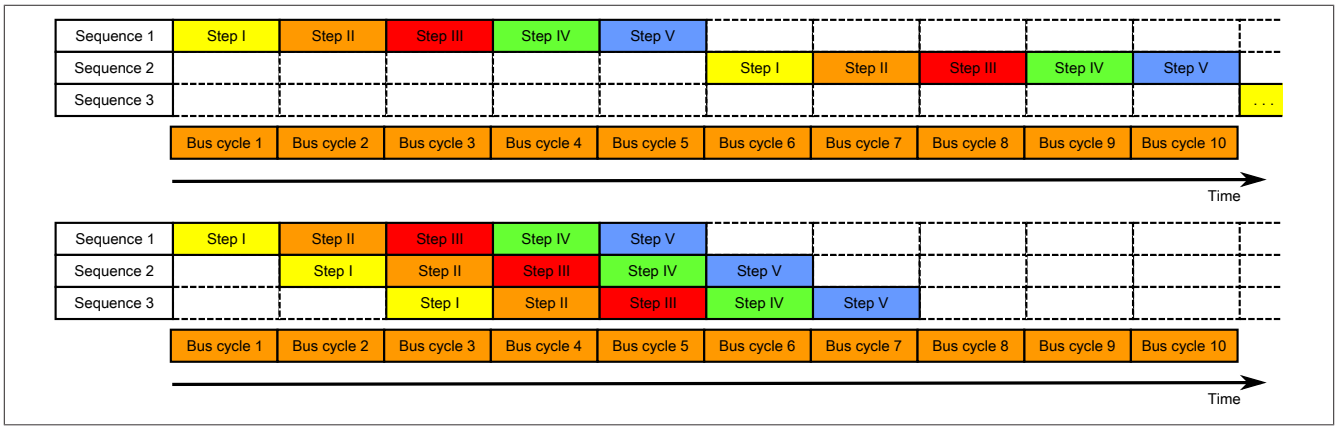


Figure 19: Comparison of transfer without/with Forward

Each of the 5 steps (tasks) requires different resources. If Forward functionality is not used, the sequences are executed one after the other. Each resource is then only active if it is needed for the current sub-action. With Forward, a resource that has executed its task can already be used for the next message. The condition for enabling the MTU is changed to allow for this. Sequences are then passed to the MTU according to the timing. The transmitting station no longer waits for an acknowledgment from SequenceAck, which means that the available bandwidth can be used much more efficiently.
In the most ideal situation, all resources are working during each bus cycle. The receiver must still acknowledge every sequence received. Only when SequenceAck has been changed and checked by the transmitter is the sequence considered as having been transferred successfully.

4.7.5.2 Configuration

The Forward function must only be enabled for the input direction. Flatstream modules have been optimized in such a way that they support this function. In the output direction, the Forward function can be used as soon as the size of OutputMTU is specified.

Information:

The registers are described in "Flatstream registers" on page 106.

Registers are described in section "Flatstream communication" in the respective data sheets.

4.7.5.2.1 Delay time

The delay time is specified in microseconds. This is the amount of time the module must wait after sending a sequence until it is permitted to write new data to the MTU in the following bus cycle. The program routine for receiving sequences from a module can therefore be run in a task class whose cycle time is slower than the bus cycle.

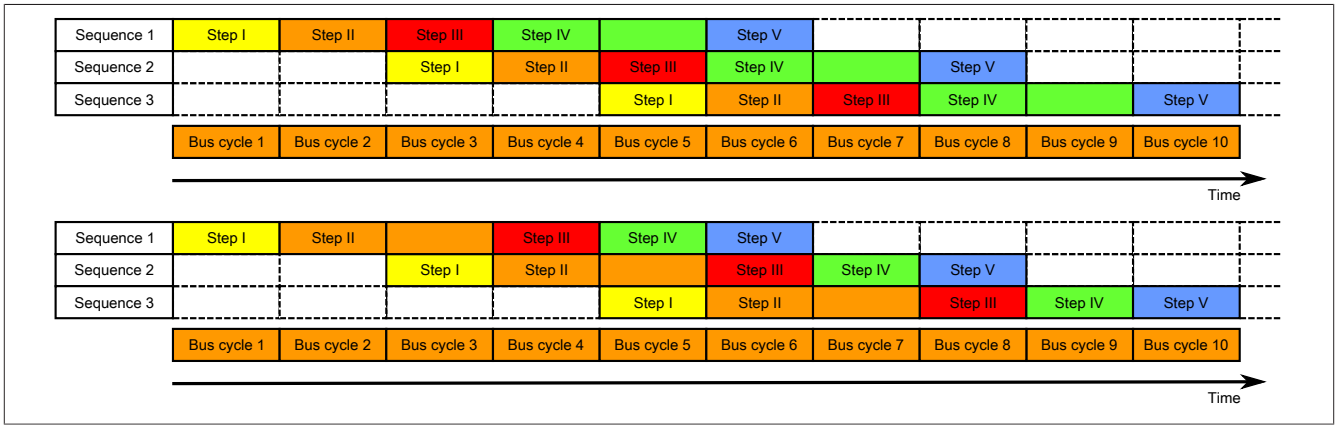


Figure 20: Effect of ForwardDelay when using Flatstream communication with the Forward function

In the program, it is important to make sure that the controller is processing all of the incoming InputSequences and InputMTUs. The ForwardDelay value causes delayed acknowledgment in the output direction and delayed reception in the input direction. In this way, the controller has more time to process the incoming InputSequence or InputMTU.

4.7.5.3 Transmitting and receiving with Forward

The basic algorithm for transmitting and receiving data remains the same. With the Forward function, up to 7 unacknowledged sequences can be transmitted. Sequences can be transmitted without having to wait for the previous message to be acknowledged. Since the delay between writing and response is eliminated, a considerable amount of additional data can be transferred in the same time window.

Algorithm for transmitting

Cyclic status query: - The module monitors OutputSequenceCounter.
0) Cyclic checks: - The controller must check OutputSyncAck. → If OutputSyncAck = 0: Reset OutputSyncBit and resynchronize the channel. - The controller must check whether OutputMTU is enabled. → If OutputSequenceCounter > OutputSequenceAck + 7, then it is not enabled because the last sequence has not yet been acknowledged.
1) Preparation (create transmit array): - The controller must split up the message into valid segments and create the necessary control bytes. - The controller must add the segments and control bytes to the transmit array.
2) Transmit: - The controller must transfer the current part of the transmit array to OutputMTU. - The controller must increase OutputSequenceCounter for the sequence to be accepted by the module. - The controller is then permitted to transmit in the next bus cycle if the MTU has been enabled.
The module responds since OutputSequenceCounter > OutputSequenceAck: - The module accepts data from the internal receive buffer and appends it to the end of the internal receive array. - The module is acknowledged and the currently received value of OutputSequenceCounter is transferred to OutputSequenceAck. - The module queries the status cyclically again.
3) Completion (acknowledgment): - The controller must check OutputSequenceAck cyclically. → A sequence is only considered to have been transferred successfully if it has been acknowledged via OutputSequenceAck. In order to detect potential transfer errors in the last sequence as well, it is important to make sure that the algorithm is run through long enough. **Note:** To monitor communication times exactly, the task cycles that have passed since the last increase of OutputSequenceCounter should be counted. In this way, the number of previous bus cycles necessary for the transfer can be measured. If the monitoring counter exceeds a predefined threshold, then the sequence can be considered lost (the relationship of bus to task cycle can be influenced by the user so that the threshold value must be determined individually).

Algorithm for receiving

0) Cyclic status query: - The controller must monitor InputSequenceCounter.
Cyclic checks: - The module checks InputSyncAck. - The module checks if InputMTU for enabling. → Enabling criteria: InputSequenceCounter > InputSequenceAck + Forward
Preparation: - The module forms the control bytes / segments and creates the transmit array.
Action: - The module transfers the current part of the transmit array to the receive buffer. - The module increases InputSequenceCounter. - The module waits for a new bus cycle after time from ForwardDelay has expired. - The module repeats the action if InputMTU is enabled.
1) Receiving (InputSequenceCounter > InputSequenceAck): - The controller must apply data from InputMTU and append it to the end of the receive array. - The controller must match InputSequenceAck to InputSequenceCounter of the sequence currently being processed.
Completion: - The module monitors InputSequenceAck. → A sequence is only considered to have been transferred successfully if it has been acknowledged via InputSequenceAck.

Details/Background

1. Illegal SequenceCounter size (counter offset)
   Error situation: MTU not enabled
   If the difference between SequenceCounter and SequenceAck during transmission is larger than permitted, a transfer error occurs. In this case, all unacknowledged sequences must be repeated with the old SequenceCounter value.
2. Checking an acknowledgment
   After an acknowledgment has been received, a check must verify whether the acknowledged sequence has been transmitted and had not yet been unacknowledged. If a sequence is acknowledged multiple times, a severe error occurs. The channel must be closed and resynchronized (same behavior as when not using Forward).

   Information:

   In exceptional cases, the module can increment OutputSequenceAck by more than 1 when using Forward.
   An error does not occur in this case. The controller is permitted to consider all sequences up to the one being acknowledged as having been transferred successfully.
3. Transmit and receive arrays
   The Forward function has no effect on the structure of the transmit and receive arrays. They are created and must be evaluated in the same way.

4.7.5.4 Errors when using Forward

In industrial environments, it is often the case that many different devices from various manufacturers are being used side by side. The electrical and/or electromagnetic properties of these technical devices can sometimes cause them to interfere with one another. These kinds of situations can be reproduced and protected against in laboratory conditions only to a certain point.

Precautions have been taken for transfer via X2X Link in case such interference should occur. For example, if an invalid checksum occurs, the I/O system will ignore the data from this bus cycle and the receiver receives the last valid data once more. With conventional (cyclic) data points, this error can often be ignored. In the following cycle, the same data point is again retrieved, adjusted and transferred.

Using Forward functionality with Flatstream communication makes this situation more complex. The receiver receives the old data again in this situation as well, i.e. the previous values for SequenceAck/SequenceCounter and the old MTU.

Loss of acknowledgment (SequenceAck)

If a SequenceAck value is lost, then the MTU was already transferred properly. For this reason, the receiver is permitted to continue processing with the next sequence. The SequenceAck is aligned with the associated SequenceCounter and sent back to the transmitter. Checking the incoming acknowledgments shows that all sequences up to the last one acknowledged have been transferred successfully (see sequences 1 and 2 in the image).

Loss of transmission (SequenceCounter, MTU):

If a bus cycle drops out and causes the value of SequenceCounter and/or the filled MTU to be lost, then no data reaches the receiver. At this point, the transmission routine is not yet affected by the error. The time-controlled MTU is released again and can be rewritten to.
The receiver receives SequenceCounter values that have been incremented several times. For the receive array to be put together correctly, the receiver is only permitted to process transmissions whose SequenceCounter has been increased by one. The incoming sequences must be ignored, i.e. the receiver stops and no longer transmits back any acknowledgments.
If the maximum number of unacknowledged sequences has been sent and no acknowledgments are returned, the transmitter must repeat the affected SequenceCounter and associated MTUs (see sequence 3 and 4 in the image).

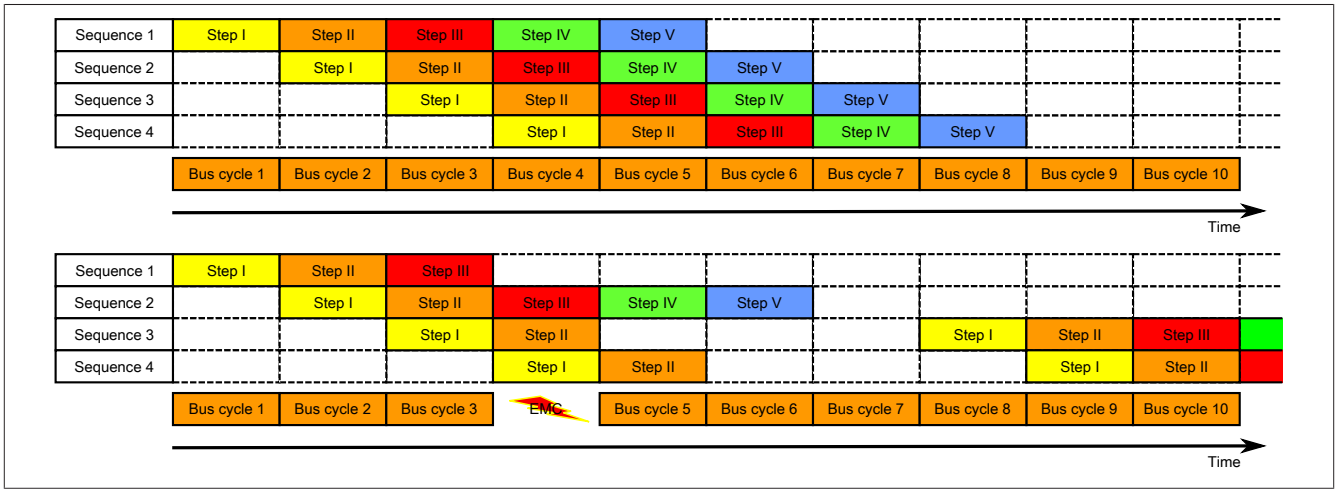


Figure 21: Effect of a lost bus cycle

Loss of acknowledgment

In sequence 1, the acknowledgment is lost due to disturbance. Sequences 1 and 2 are therefore acknowledged in Step V of sequence 2.

Loss of transmission

In sequence 3, the entire transmission is lost due to disturbance. The receiver stops and no longer sends back any acknowledgments.
The transmitting station continues transmitting until it has issued the maximum permissible number of unacknowledged transmissions.
5 bus cycles later at the earliest (depending on the configuration), it begins resending the unsuccessfully sent transmissions.

4.8 Flatstream communication with function blocks

As an additional option for Flatstream communication, communication with the module can be easily carried out with the "AsFltGen" library.

The library function blocks handle all incoming tasks with Flatstream mode, such as forwarding, sequencing, generation and evaluation of control bytes.

5 Commissioning

5.1 Using the module on the bus controller

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

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

5.1.1 CAN I/O bus controller

The module occupies 3 analog logical slots with CAN I/O.

5.2 Usage after the X20IF1091-1

If this module is operated after X2X Link module X20IF1091-1, delays may occur during the Flatstream transfer. For detailed information, see section "Data transfer on the Flatstream" in X20IF1091-1.

5.3 Current transformers

Since the current inputs are not floating, a current transformer is required for each current channel used. The current transformer is a transducer that delivers a secondary signal proportional to the primary current. This secondary signal is measured by the module. The maximum directly configurable primary current is 65 A. Values greater than 65 A can also be measured by implementing a conversion in the software application (see explanation and example provided below).

The maximum secondary signal depends on the module:

Module	Secondary current/voltage
X20AP3111	20 mA
X20AP3121	1 A
X20AP3122	1 A
X20AP3131	5 A
X20AP3132	5 A
X20AP3161	333 mV
X20AP3171	Configurable, maximum 52 mV

The rated transformation ratio is calculated using the following formula:

X20AP3111 - X20AP3121 - X20AP3131 - X20AP3122 - X20AP3132	$\text{Rated transformation ratio } K_n = \frac{\text{Primary nominal current}}{\text{Secondary nominal current}}$
X20AP3161	No transformation; the maximum primary current corresponds to the 333 mV.
X20AP3171	Direct input of µV/A

A smaller transformation ratio should be defined for measuring higher primary currents. The values calculated by the module must be converted in the application according to the real rated transformation ratio that must be defined.

Examples

Information:

The same factor must be used for all power ratings and energy values when making the conversion.

All AP modules except for AP3171

Currents of up to 100 A flow on the primary side. A current transformer with a rated transformation ratio of 100 to 1 A or a measurement range of 100 A is used. A rated transformation ratio of 50 to 1 A or a measurement range of 50 A is defined in the module to match the current transformer. If the primary current calculated by the module is 40 A, then the actual value will be calculated as follows:

Actual primary current = 40 A * 100 / 50 = 80 A
Actual resolution = 1 mA * 100 / 50 = 2 mA

X20AP3171

Because the primary current can be up to 300 A, a Rogowski coil with a primary current range of up to 500 A is used. This exhibits a transformation ratio of 100 µV/A.
Since the module can only supply values up to 65000, however, only a maximum of 65 A could be displayed or measured with it. The transformation ratio of the phases must therefore additionally be set.

The unit of the phase is 0.1 µV/A; the transformation ratio must therefore be converted. For example, the value 5000 * 0.1 µV corresponds to 500 µV/A.

Example
The transformation ratio of the Rogowski coil is 100 µV/A.
The transformation ratio of the phase is set to value 5000.
The module returns a value of 8155 mA.

The calculation is made according to this formula:

$$\text{Value} = \frac{\text{Transformation ratio}_{\text{Phase}} * \text{Unit}_{\text{Phase}}}{\text{Transformation ratio}_{\text{Rogowski}}} * \text{Measured value}$$

This results in this value for our example:

$$\frac{5000 * 0.1\,\mu V/A}{100\mu V/A} * 8155 = 40.775\ A$$

Maximum primary current that can be measured by the module:

$$\text{Primary current}_{\text{Maximum}} = \frac{8000_{\text{Phase}} * \text{Unit}_{\text{Phase}}}{\text{Transformation ratio}_{\text{Rogowski}}} * 65000$$

Caution!

To prevent damage to the module, ensure that the current inputs are potential-free. Therefore, one current transformer must be connected for each current input used.

If other devices are connected to this secondary circuit, they must be galvanically isolated.

X20AP31x1:

The current inputs on the module are not galvanically isolated, so the secondary circuit between the converter and the module is not permitted to be grounded. Grounding or any other conductive connection between the converters will falsify the measurement and display current values that are too low!

X20AP31x2:

Since the "ILxb" connections on the current inputs are all at the same electric potential, the transformers on the "Lxb" side must be grounded for these modules.

Attention !

Pour éviter d'endommager le module, s'assurer que les entrées de courant sont libres de potentiel. Pour ce faire, un transformateur de courant doit être connecté pour chaque entrée de courant utilisée.

Si d'autres appareils sont connectés à ce circuit secondaire, ils doivent posséder une isolation galvanique.

X20AP31x1 :

Les entrées de courant du module n'ayant pas d'isolation galvanique, il n'est pas autorisé de mettre à la terre le circuit secondaire entre le convertisseur et le module. La mise à la terre ou toute autre connexion conductrice entre les convertisseurs faussera la mesure et affichera des valeurs de courant trop faibles !

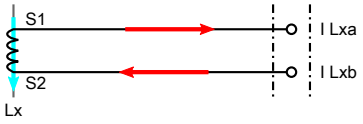
X20AP31x2 :

Étant donné que les connexions « ILxb » sur les entrées de courant ont toutes le même potentiel électrique, les transformateurs côté « Lxb » doivent être mis à la terre pour ces modules.

5.3.1 Connecting current transformers

In order to be able to properly calculate values, it is important for the current transformer phases to be connected correctly (i.e. direction of the current flow).

- Output on the transformer (S1) to the respective ILxa input on the module.
- Input on the transformer (S2) to the respective ILxb output on the module.



5.4 Voltage transformer

Voltage transformers are not provided in the configuration by default (e.g. by setting the transformation ratio).

However, voltage transformers can be used if higher voltages need to be measured than are specified under nominal voltages in the technical data.

In addition, as with current value correction, the rated transformation between primary and secondary current must be calculated and applied (see "Current transformers" on page 58)

Information:

The same factor must be used for all voltage values, power ratings and energy values when making the transformation.

5.5 Configuration registers

The configuration and calibration registers are each composed of blocks and employ a checksum feature to highlight undesired changes. In order to apply this register to the transformer, the respective transfer register must be changed after the data is transferred to the module (incrementing, bit toggling, etc.). The start value of the transfer register is 0 after startup.

5.5.1 User calibration

5.5.1.1 Current and voltage values

The following procedure must be followed to correctly calculate the gain and offset:

- Read out the specified values:
  The values of the registers described in section "A/D converter RMS value synchronization – read" on page 98 must be read at the start of calibration. This is the only way to ensure that the gain and offset can be calculated correctly.
  The values contained in the registers correspond to $value_{alt}$ in the calculation formulas for gain and offset (see "A/D converter RMS value synchronization checksum 3" on page 100).
- Calculate and write new values:
  See "A/D converter RMS value synchronization checksum 3" on page 100.
  - Voltage RMS value gain of phase A/B/C/N:
    $Value_{new} = Value_{old}$ * Correction factor, determined with $U = U_n$
  - Current RMS gain of phase A/B/C/N:
    $Value_{new} = Value_{old}$ * Correction factor, determined with $I = I_n$
  - Voltage RMS value offset of phase A/B/C:
    Negated value of the corresponding RMS value register at U = 0.
  - Current RMS value offset of phase A/B/C/N:
    Negated value of the corresponding RMS value register at I = 0.
- Apply the specified values by writing to register "Cs3Update" on page 94. Only when the value in register "Cs3UpdateFB" on page 94 corresponds to the value of <Cs3Update> are all the specified values applied.

5.5.1.2 Power values

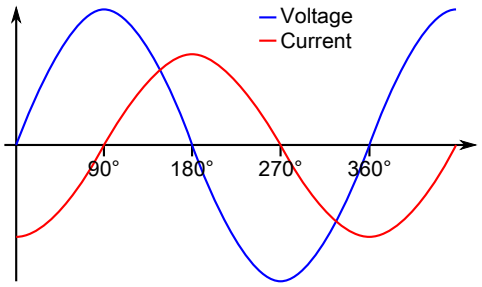
The following procedure must be followed to correctly calculate the power angle corrections:

1) Determine the values.
2) Write value 0xFFFF to register "Cs1Update" on page 94.
3) Read register "Cs1UpdateFB" on page 94 until value 0xFFFF is returned.
4) Write the determined values to registers "PhiA_W, PhiB_W, PhiC_W" on page 101.

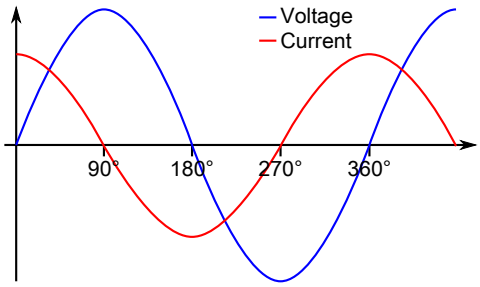
   Description of bits 0 to 9

   The maximum correction value 0x3FF = 1023 dec. corresponds to a time of 0.49951 ms.
   With a 50 Hz mains, this corresponds to a change of 8.99 degrees.
   With a 60 Hz mains, this corresponds to a change of 10.79 degrees.

   Schematic diagram of inductive load: Voltage leads the current



   Schematic representation of capacitive load: Current leads the voltage



   Description of bit 15

0	Delay affects current channel	
	Effect with inductive load	Reducing of the angle between I and U, increasing the power factor
	Effect with capacitive load	Increasing the angle between U and I, reducing the power factor
1	Delay affects voltage channel	
	Effect with inductive load	Reducing the angle between U and I, increasing the power factor
	Effect with capacitive load	Increasing the angle between I and U, reducing the power factor

5) Write value 0x0001 to register Cs1Update.
6) Read register Cs1UpdateFB until value 0x0001 is returned.

Information:

The power factor correction registers are NOT nonvolatile, and the procedure must be repeated at each startup or on each positive edge of bit ModulOK.

5.5.2 Saving the user configuration

The following procedure has to be complied with to apply the new values in a configuration change.

1) Writing update register
   - CfgUpdate = 0xFFFF
   - Cs0Update = 0xFFFF
2) Writing the desired configuration register
3) Writing update register
   - CfgUpdate = 0x1
   - Cs0Update = 0x1

5.6 Interface for transferring process variable mapping

Due to the amount of potential cyclic input data and the limitation to 30 byte cyclic X2X data, the extended Flatstream Interface (DPS = Data Point Stream) has been defined as the mechanism for transferring the process variables. DPS is based on the Flat Streaming Interface for serial interface modules. The Flat Streaming Interface was expanded to include the block number as the first byte of the user data frame and implements the termination of a frame (data image of the channel) with a zero segment.

The data blocks are re-transferred if a read request is triggered after a transfer has been completed. A block number can be sent via the DPS to set a different block or transfer the entire image (default: block number 0).

It should be possible to adapt the DPS interface to the available buffer size. However, the higher-level fieldbus must be taken into account when doing so (e.g. CAN 8 byte object, InputMTU size 7). The block number is added to the front of the actual payload data as a means to differentiate the blocks.

```
#define ADC_BLK_ALL         0   // struct ADC_REG
#define ADC_BLK_STATUS      1   // long NetTimeReg + struct ADC_REG_STATUS
#define ADC_BLK_RMS         2   // struct ADC_REG_RMS
#define ADC_BLK_POWER       3   // struct ADC_REG_POWER
#define ADC_BLK_THD_ANGLE   4   // struct THD_ANGLE
#define ADC_BLK_ENERGY      5   // long NetTimeEnergy + struct ADC_REG_ENERGY
#define ADC_BLK_DFT         6   // long NetTimeDft + struct ADC_REG_DFT
#define ADC_BLK_CFGACT      7   // struct ADC_REG_CFGACT
#define ADC_BLK_ENVREG      8   // struct ENV_STATUS
```

Information:

- **Consistency of the data is only provided for the individual variables because the data is transferred from the A/D converter asynchronously to the conversion.**
- **Make sure that the byte sequence of the register is in accordance with the Little Endian model (Intel format).**

The NetTime timestamps are always updated after the blocks have been generated when a new alternating buffer is provided.

5.6.1 Data block structures

5.6.1.1 ADC_REG

```
typedef struct ADC_REG     ADC_REG;
struct ADC_REG
{
 long              NetTimeReg;  // Time of Section copy to Buffer
 ADC_REG_STATUS    Status;      // Status registers
 ADC_REG_RMS       Rms;         // RMS Registers
 ADC_REG_POWER     Power;       // Power Registers
 ADC_REG_THD_ANGLE ThdAngle;    // THD + Angle Registers

// Regular Energy Registers
 long            NetTimeEnergy; // Time of Section copy to Buffer
 ADC_REG_ENERGY  Energy;        // Energy Registers

 long            NetTimeDft;    // Time of Section copy to Buffer
 ADC_REG_DFT     Dft;           // DFT Registers
// Read Back selected CFG Registers
  ADC_REG_CFGACT  CfgAct;       // Config read back
// Read Back Environment Registers
  ENV_STATUS      EnvReg;
};
```

5.6.1.2 ADC_REG_STATUS

```
typedef struct ADC_REG_STATUS  ADC_REG_STATUS;
struct ADC_REG_STATUS
{
  unsigned short SysStatus0;   // SysStatus 0
  unsigned short SysStatus1;   // SysStatus 1
  unsigned short EnStatus0;    // SysStatus 2
  unsigned short EnStatus1;    // SysStatus 3
};
```

5.6.1.3 ADC_REG_RMS

```
typedef struct ADC_REG_RMS  ADC_REG_RMS;
struct ADC_REG_RMS
{
  unsigned short IrmsN1;   // N Line Sampled current RMS
  unsigned short UrmsA;    // phase A voltage RMS
  unsigned short UrmsB;    // phase B voltage RMS
  unsigned short UrmsC;    // phase C voltage RMS
  unsigned short IrmsN0;   // N Line calculated current RMS
  unsigned short IrmsA;    // phase A current RMS
  unsigned short IrmsB;    // phase B current RMS
  unsigned short IrmsC;    // phase C current RMS
};
```

5.6.1.4 ADC_REG_POWER

```
typedef struct ADC_REG_POWER  ADC_REG_POWER;
struct ADC_REG_POWER
{
  unsigned short SVmeanTLSB;  // LSB of (Vector Sum) Total Apparent Power
  unsigned short SVmeanT;     // (Vector Sum) Total Apparent Power

  // Power and Power Factor Register
  signed short PmeanT;   // Total Active Power
  signed short PmeanA;   // Phase A Active Power
  signed short PmeanB;   // Phase B Active Power
  signed short PmeanC;   // Phase C Active Power
  signed short QmeanT;   // Total Reactive Power
  signed short QmeanA;   // Phase A Reactive Power
  signed short QmeanB;   // Phase B Reactive Power
  signed short QmeanC;   // Phase C Reactive Power
  signed short SmeanT;   // (Arithmetic Sum) Total apparent power
  signed short SmeanA;   // phase A apparent power
  signed short SmeanB;   // phase B apparent power
  signed short SmeanC;   // phase C apparent power
  signed short PFmeanT;  // Total power factor
  signed short PFmeanA;  // phase A power factor
  signed short PFmeanB;  // phase A power factor
  signed short PFmeanC;  // phase A power factor

  // Fundamental/ Harmonic Power and Voltage/ Current RMS Registers
  signed short PmeanTF;  // Total active fundamental power
  signed short PmeanAF;  // phase A active fundamental power
  signed short PmeanBF;  // phase B active fundamental power
  signed short PmeanCF;  // phase C active fundamental power
  signed short PmeanTH;  // Total active harmonic power
  signed short PmeanAH;  // phase A active harmonic power
  signed short PmeanBH;  // phase B active harmonic power
  signed short PmeanCH;  // phase C active harmonic power
};
```

5.6.1.5 ADC_REG_THD_ANGLE

```
typedef struct ADC_REG_THD_ANGLE  ADC_REG_THD_ANGLE;
struct ADC_REG_THD_ANGLE
{
  // THD+N, Frequency, Angle and Temperature Registers
  unsigned short THDNUA;    // phase A voltage THD+N
  unsigned short THDNUB;    // phase B voltage THD+N
  unsigned short THDNUC;    // phase C voltage THD+N
  unsigned short THDNIA;    // phase A current THD+N
  unsigned short THDNIB;    // phase B current THD+N
  unsigned short THDNIC;    // phase C current THD+N
  unsigned short Freq;      // Frequency
  signed short     PAngleA;   // phase A mean phase angle
  signed short     PAngleB;   // phase B mean phase angle
  signed short     PAngleC;   // phase C mean phase angle
  signed short     Temp;      // Measured temperature
  signed short     UangleA;   // phase A voltage phase angle
  signed short     UangleB;   // phase B voltage phase angle
  signed short     UangleC;   // phase C voltage phase angle
};
```

5.6.1.6 ADC_REG_ENERGY

```
typedef struct ADC_REG_ENERGY  ADC_REG_ENERGY;
struct ADC_REG_ENERGY
{
  unsigned long APenergyT;   // Total Forward Active Energy
  unsigned long APenergyA;   // Phase A Forward Active Energy
  unsigned long APenergyB;   // Phase B Forward Active Energy
  unsigned long APenergyC;   // Phase C Forward Active Energy
  unsigned long ANenergyT;   // Total Reverse Active Energy
  unsigned long ANenergyA;   // Phase A Reverse Active Energy
  unsigned long ANenergyB;   // Phase B Reverse Active Energy
  unsigned long ANenergyC;   // Phase C Reverse Active Energy
  unsigned long RPenergyT;   // Total Forward Reactive Energy
  unsigned long RPenergyA;   // Phase A Forward Reactive Energy
  unsigned long RPenergyB;   // Phase B Forward Reactive Energy
  unsigned long RPenergyC;   // Phase C Forward Reactive Energy
  unsigned long RNenergyT;   // Total Reverse Reactive Energy
  unsigned long RNenergyA;   // Phase A Reverse Reactive Energy
  unsigned long RNenergyB;   // Phase B Reverse Reactive Energy
  unsigned long RNenergyC;   // Phase C Reverse Reactive Energy
  unsigned long SAenergyT;   // (Arithmetic Sum) Total Apparent Energy
  unsigned long SenergyA;    // Phase A Apparent Energy
  unsigned long SenergyB;    // Phase B Apparent Energy
  unsigned long SenergyC;    // Phase C Apparent Energy
  unsigned long SVenergyT;   // (Vector Sum) Total Apparent Energy

  // Fundamental / Harmonic Energy Register
  unsigned long APenergyTF;  // Total Forward Active Fundamental Energy
  unsigned long APenergyAF;  // Phase A Forward Active Fundamental Energy
  unsigned long APenergyBF;  // Phase B Forward Active Fundamental Energy
  unsigned long APenergyCF;  // Phase C Forward Active Fundamental Energy
  unsigned long ANenergyTF;  // Total Reverse Active Fundamental Energy
  unsigned long ANenergyAF;  // Phase A Reverse Active Fundamental Energy
  unsigned long ANenergyBF;  // Phase B Reverse Active Fundamental Energy
  unsigned long ANenergyCF;  // Phase C Reverse Active Fundamental Energy
  unsigned long APenergyTH;  // Total Forward Active Harmonic Energy
  unsigned long APenergyAH;  // Phase A Forward Active Harmonic Energy
  unsigned long APenergyBH;  // Phase B Forward Active Harmonic Energy
  unsigned long APenergyCH;  // Phase C Forward Active Harmonic Energy
  unsigned long ANenergyTH;  // Total Reverse Active Harmonic Energy
  unsigned long ANenergyAH;  // Phase A Reverse Active Harmonic Energy
  unsigned long ANenergyBH;  // Phase B Reverse Active Harmonic Energy
  unsigned long ANenergyCH;  // Phase C Reverse Active Harmonic Energy

  signed long AEnergyT;   // Total Active Energy
  signed long REnergyT;   // Total Reactive Energy
};
```

5.6.1.7 ADC_REG_DFT

```
typedef struct ADC_REG_DFT  ADC_REG_DFT;
struct ADC_REG_DFT
{
  // Arithmetic ratio, 2 bits integer and 14 bits fractional;
  // That is: Harmonic Ratio (%) = Register Value / 163.84
  unsigned short DftAI[32]; // phase A, Current, Harmonic Ratio for 2nd to 32nd
                            // order componentand Total Harmonic DistortionRatio
  unsigned short DftBI[32]; // phase B, Current, Harmonic Ratio for 2nd to 32nd
                            // order componentand Total Harmonic DistortionRatio
  unsigned short DftCI[32]; // phase C, Current, Harmonic Ratio for 2nd to 32nd
                            // order componentand Total Harmonic DistortionRatio
  unsigned short DftAV[32]; // phase A, Voltage, Harmonic Ratio for 2nd to 32nd
                            // order componentand Total Harmonic DistortionRatio
  unsigned short DftBV[32]; // phase B, Voltage, Harmonic Ratio for 2nd to 32nd
                            // order componentand Total Harmonic DistortionRatio
  unsigned short DftCV[32]; // phase C, Voltage, Harmonic Ratio for 2nd to 32nd
                            // order componentand Total Harmonic DistortionRatio

  unsigned short DftAI_Fund;
  unsigned short DftAV_Fund;
  unsigned short DftBI_Fund;
  unsigned short DftBV_Fund;
  unsigned short DftCI_Fund;
  unsigned short DftCV_Fund;
};
```

5.6.1.8 ADC_REG_CFACT

```
// Except of configuration registers used by APROL, readable only by FS-IF
// and with register numbers of registers with the same names.

typedef struct ADC_REG_CFGACTADC_REG_CFGACT;
struct ADC_REG_CFGACT
{
  unsigned short ChanControl;
  unsigned short IDispTh;
  unsigned short I_RatioA;
  unsigned short I_RatioB;
  unsigned short I_RatioC;
  unsigned short I_RatioN;
  unsigned short ZXConfig;
  unsigned short SagTh;
  unsigned short PhaseLoseTh;
  unsigned short INWarnTh0;
  unsigned short INWarnTh1;
  unsigned short THDNUTh;
  unsigned short THDNITh;
  unsigned short MeteringMode;
  unsigned short PLconstL;
  unsigned short PLconstH;
};
```

5.6.1.9 ENV_STATUS

```
// Environment Variables

typedef struct ENV_STATUSENV_STATUS;
struct ENV_STATUS
{
  unsigned long  ulUpTime;
  unsigned long  ulUpCnt;
  signed short   ssMinTemp;
  signed short   ssMaxTemp;
  unsigned long  ulRes[13];      // reserved
};
```

6 UL certificate information

To install the module(s) according to the UL standard, the following rules must be observed.

Information:

- **Use copper conductors only. Minimum temperature rating of the cable to be connected to the field wiring terminals: 76°C, xxx - xxx AWG.**
- **All models are intended to be used in a final safety enclosure that must conform with requirements for protection against the spread of fire and have adequate rigidity per UL 61010-1 and UL 61010-2-201.**
- **Repairs can only be made by B&R.**

Information:

- **Remark regarding UL61010 certification: For use with UL-Listed Energy-Monitoring Current Transformers only.**
- **The current transformers are not permitted to be installed in equipment where they exceed 75 percent of the wiring space of any cross-sectional area within the equipment**
- **Restrict installation of current transformer in an area where it would block ventilation openings.**
- **Restrict installation of current transformer in an area of breaker arc venting.**
- **Not suitable for Class 2 wiring methods.**
- **Not intended for connection to Class 2 equipment.**
- **Secure current transformer and route conductors so that they do not directly contact live terminals or bus.**
- **For use with listed energy-monitoring current transformers.**
- **Associated leads of the current transformers must be maintained within the same overall enclosure.**
- **Unless the current transformers and its leads have been evaluated for REINFORCED INSULATION, a statement to segregate or insulate the leads from different circuits must be provided.**
- **The current transformers are intended for installation within the same enclosure as the equipment. These are not permitted to be installed within switchgear and panel boards.**

Danger!

- **To reduce the risk of electric shock, always open or disconnect circuit from the power distribution system (or service) of the building before installing or servicing current-sensing transformers.**

7 Register description

7.1 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.

7.2 Function model 0 - Standard

Register	Name	Data type	Read		Write	
			Cyclic	Acyclic	Cyclic	Acyclic
Status register						
130	StatusInput	UINT	•			
	CntPulseActive	Bit 0				
	CntPulseApparent	Bit 1				
	CntPulseActiveFund	Bit 2				
	CntPulseActiveHarm	Bit 3				
	ZeroCrossA	Bit 4				
	ZeroCrossB	Bit 5				
	ZeroCrossC	Bit 6				
	RBTrigDFT	Bit 8				
	RBUpdateEnergy	Bit 9				
	RBClearEnergy	Bit 10				
	RBForceEnergy	Bit 11				
194	ControlOutput	UINT			•	
	TrigDFT	Bit 0				
	EnabEnergy	Bit 1				
	ClearEnergy	Bit 2				
	ForceEnergy	Bit 3				
266	SysStatus1	UINT	•			
270	SysStatus2	UINT	•			
274	SysStatus3	UINT	•			
278	SysStatus4	UINT	•			
265	SystemStatusSel01	USINT	•			
	SumStatusPhaseLoss	Bit 2				
	SumStatusPhaseWarning	Bit 3				
	ErrOrderPhasecurrent	Bit 6				
	ErrOrderPhaseVoltage	Bit 7				
271	SystemStatusSel02	USINT	•			
	SumStatusWarningTHDCurrent	Bit 2				
	SumStatusWarningTHDVoltage	Bit 3				
	ErrIrmsNCalc	Bit 6				
	ErrIrmsNMeas	Bit 7				
278	PhaseStatus	UINT	•			
	LossPhaseC	Bit 0				
	LossPhaseB	Bit 1				
	LossPhaseA	Bit 2				
	WarningPhaseC	Bit 4				
	WarningPhaseB	Bit 5				
	WarningPhaseA	Bit 6				
Analog RMS value registers						
290	IrmsN (measured)	UINT	•			
294	UrmsA	UINT	•			
298	UrmsB	UINT	•			
302	UrmsC	UINT	•			
306	IrmsNcalc (calculated)	UINT	•			
310	IrmsA	UINT	•			
314	IrmsB	UINT	•			
318	IrmsC	UINT	•			
Analog THD and angle registers						
538	Freq	UINT	•			
542	PAngleA	INT	•			
546	PAngleB	INT	•			

Register	Name	Data type	Read		Write	
			Cyclic	Acyclic	Cyclic	Acyclic
550	PAngleC	INT	•			
554	Temperature	INT	•			
558	UAngleA	INT	•			
562	UAngleB	INT	•			
564	UAngleC	INT	•			
Analog power registers						
778	PmeanT	INT	•			
782	PmeanA	INT	•			
786	PmeanB	INT	•			
790	PmeanC	INT	•			
794	QmeanT	INT	•			
798	QmeanA	INT	•			
802	QmeanB	INT	•			
806	QmeanC	INT	•			
810	SmeanT	INT	•			
814	SmeanA	INT	•			
818	SmeanB	INT	•			
822	SmeanC	INT	•			
826	PFmeanT	INT	•			
830	PFmeanA	INT	•			
834	PFmeanB	INT	•			
838	PFmeanC	INT	•			
Analog energy registers						
4108	APenergyT	UDINT	•			
4116	APenergyA	UDINT	•			
4124	APenergyB	UDINT	•			
4132	APenergyC	UDINT	•			
4140	ANenergyT	UDINT	•			
4148	ANenergyA	UDINT	•			
4156	ANenergyB	UDINT	•			
4164	ANenergyC	UDINT	•			
4172	RPenergyT	UDINT	•			
4180	RPenergyA	UDINT	•			
4188	RPenergyB	UDINT	•			
4196	RPenergyC	UDINT	•			
4204	RNenergyT	UDINT	•			
4212	RNenergyA	UDINT	•			
4220	RNenergyB	UDINT	•			
4228	RNenergyC	UDINT	•			
4236	SAenergyT	UDINT	•			
4244	SenergyA	UDINT	•			
4252	SenergyB	UDINT	•			
4260	SenergyC	UDINT	•			
4268	SVenergyT	UDINT	•			
4404	AEnergyT	DINT	•			
4412	REnergyT	DINT	•			
Module configuration						
1026	ChanControl	UINT				•
1030	IDispTh	UINT				•
1034	I_RatioA	UINT				•
1038	I_RatioB	UINT				•
1042	I_RatioC	UINT				•
1046	I_RatioN	UINT				•
Update request						
1050	CfgUpdate	UINT				•
1054	Cs0Update	UINT				•
1058	Cs1Update	UINT				•
1066	Cs3Update	UINT				•
1570	Cs1UpdateFB	UINT		•		
1578	Cs3UpdateFB	UINT		•		
A/D converter status configuration						
1090	ZXConfig	UINT				•
1094	SagTh	UINT				•
1098	PhaseLoseTh	UINT				•
1102	INWarnTh0	UINT				•
1106	INWarnTh1	UINT				•
1110	THDNUTh	UINT				•
1114	THDNITh	UINT				•
A/D converter measurement configuration checksum 0						
1154	PLconstH	UINT				•
1158	PLconstL	UINT				•
1162	MeteringMode	UINT				•

Register description

Register	Name	Data type	Read		Write	
			Cyclic	Acyclic	Cyclic	Acyclic
A/D converter power calibration checksum 1						
1246	PhiA_W	UINT				•
1254	PhiB_W	UINT				•
1262	PhiC_W	UINT				•
A/D converter RMS value synchronization checksum 3						
1346	UGainA_W	UINT				•
1350	IGainA_W	UINT				•
1354	UoffsetA_W	INT				•
1358	IoffsetA_W	INT				•
1362	UGainB_W	UINT				•
1366	IGainB_W	UINT				•
1370	UoffsetB_W	INT				•
1374	IoffsetB_W	INT				•
1378	UGainC_W	UINT				•
1382	IGainC_W	UINT				•
1386	UoffsetC_W	INT				•
1390	IoffsetC_W	INT				•
1394	IGainN_W	UINT				•
1398	IoffsetN_W	INT				•
A/D converter power calibration – read						
1758	PhiA_R	UINT		•		
1766	PhiB_R	UINT		•		
1774	PhiC_R	UINT		•		
A/D converter RMS value synchronization – read						
1858	UGainA_R	UINT		•		
1862	IGainA_R	UINT		•		
1866	UoffsetA_R	INT		•		
1870	IoffsetA_R	INT		•		
1874	UGainB_R	UINT		•		
1878	IGainB_R	UINT		•		
1882	UoffsetB_R	INT		•		
1886	IoffsetB_R	INT		•		
1890	UGainC_R	UINT		•		
1894	IGainC_R	UINT		•		
1898	UoffsetC_R	INT		•		
1902	IoffsetC_R	INT		•		
1906	IGainN_R	UINT		•		
1910	IoffsetN_R	INT		•		
Flatstream interface						
2049	OutputMTU	USINT				•
2051	InputMTU	USINT				•
2055	FlatstreamMode	USINT				•
2057	Forward	USINT				•
2059	ForwardDelay	USINT				•
2113	InputSequence	USINT	•			
2113 + 2*N	RxByteN (index N = 1 to 27)	USINT	•			
2177	OutputSequence	USINT			•	
2177 + 2*N	TxByteN (index N = 1 to 15)	USINT			•	
Force analog energy registers						
2316	Frc_APenergyT	UDINT				•
2324	Frc_APenergyA	UDINT				•
2332	Frc_APenergyB	UDINT				•
2340	Frc_APenergyC	UDINT				•
2348	Frc_ANenergyT	UDINT				•
2356	Frc_ANenergyA	UDINT				•
2364	Frc_ANenergyB	UDINT				•
2372	Frc_ANenergyC	UDINT				•
2380	Frc_RPenergyT	UDINT				•
2388	Frc_RPenergyA	UDINT				•
2396	Frc_RPenergyB	UDINT				•
2404	Frc_RPenergyC	UDINT				•
2412	Frc_RNenergyT	UDINT				•
2420	Frc_RNenergyA	UDINT				•
2428	Frc_RNenergyB	UDINT				•
2436	Frc_RNenergyC	UDINT				•
2444	Frc_SAenergyT	UDINT				•
2452	Frc_SenergyA	UDINT				•
2460	Frc_SenergyB	UDINT				•
2468	Frc_SenergyC	UDINT				•
2476	Frc_SVenergyT	UDINT				•
2484	Frc_APenergyTF	UDINT				•
2492	Frc_APenergyAF	UDINT				•
2500	Frc_APenergyBF	UDINT				•

Register	Name	Data type	Read		Write	
			Cyclic	Acyclic	Cyclic	Acyclic
2508	Frc_APenergyCF	UDINT				•
2516	Frc_ANenergyTF	UDINT				•
2524	Frc_ANenergyAF	UDINT				•
2532	Frc_ANenergyBF	UDINT				•
2540	Frc_ANenergyCF	UDINT				•
2548	Frc_APenergyTH	UDINT				•
2556	Frc_APenergyAH	UDINT				•
2564	Frc_APenergyBH	UDINT				•
2572	Frc_APenergyCH	UDINT				•
2580	Frc_ANenergyTH	UDINT				•
2588	Frc_ANenergyAH	UDINT				•
2596	Frc_ANenergyBH	UDINT				•
2604	Frc_ANenergyCH	UDINT				•
Oversampling buffer						
6146 + ((16-N)*40)	IactN_SampleN (Index N = 1 to 16)	INT	•			
6150 + ((16-N)*40)	IactA_SampleN (Index N = 1 to 16)	INT	•			
6154 + ((16-N)*40)	UactA_SampleN (Index N = 1 to 16)	INT	•			
6158 + ((16-N)*40)	IactB_SampleN (Index N = 1 to 16)	INT	•			
6162 + ((16-N)*40)	UactB_SampleN (Index N = 1 to 16)	INT	•			
6166 + ((16-N)*40)	IactC_SampleN (Index N = 1 to 16)	INT	•			
6170 + ((16-N)*40)	UactC_SampleN (Index N = 1 to 16)	INT	•			
6773	SampleCountN	SINT	•			
6774		INT				
6778	Timestamp	INT	•			
6780		DINT				
Environment variables						
15108	OnTime	UDINT		•		
15116	UpCounter	UDINT		•		
15122	MinTemp	INT		•		
15126	MaxTemp	INT		•		

7.3 Function model 254 - Bus controller

Register	Offset[1)]	Name	Data type	Read		Write	
				Cyclic	Acyclic	Cyclic	Acyclic
Status register							
130	0	StatusInput	UINT	•			
		CntPulseActive	Bit 0				
		CntPulseApparent	Bit 1				
		CntPulseActiveFund	Bit 2				
		CntPulseActiveHarm	Bit 3				
		ZeroCrossA	Bit 4				
		ZeroCrossB	Bit 5				
		ZeroCrossC	Bit 6				
		RBTrigDFT	Bit 8				
		RBUpdateEnergy	Bit 9				
		RBClearEnergy	Bit 10				
		RBForceEnergy	Bit 11				
194	0	ControlOutput	UINT			•	
		TrigDFT	Bit 0				
		EnabEnergy	Bit 1				
		ClearEnergy	Bit 2				
		ForceEnergy	Bit 3				
266	-	SysStatus1	UINT		•		
270	-	SysStatus2	UINT		•		
274	-	SysStatus3	UINT		•		
278	-	SysStatus4	UINT		•		
265	-	SystemStatusSel01	USINT		•		
		SumStatusPhaseLoss	Bit 2				
		SumStatusPhaseWarning	Bit 3				
		ErrOrderPhasecurrent	Bit 6				
		ErrOrderPhaseVoltage	Bit 7				
271	-	SystemStatusSel02	USINT		•		
		SumStatusWarningTHDCurrent	Bit 2				
		SumStatusWarningTHDVoltage	Bit 3				
		ErrIrmsNCalc	Bit 6				
		ErrIrmsNMeas	Bit 7				
278	-	PhaseStatus	UINT		•		
		LossPhaseC	Bit 0				
		LossPhaseB	Bit 1				
		LossPhaseA	Bit 2				
		WarningPhaseC	Bit 4				
		WarningPhaseB	Bit 5				
		WarningPhaseA	Bit 6				
Analog RMS value registers							
290	-	IrmsN (measured)	UINT		•		
294	-	UrmsA	UINT		•		
298	-	UrmsB	UINT		•		
302	-	UrmsC	UINT		•		
306	-	IrmsNcalc (calculated)	UINT		•		
310	-	IrmsA	UINT		•		
314	-	IrmsB	UINT		•		
318	-	IrmsC	UINT		•		
Analog THD and angle registers							
538	-	Freq	UINT		•		
542	-	PAngleA	INT		•		
546	-	PAngleB	INT		•		
550	-	PAngleC	INT		•		
554	-	Temperature	INT		•		
558	-	UAngleA	INT		•		
562	-	UAngleB	INT		•		
564	-	UAngleC	INT		•		
Analog power registers							
778	2	PmeanT	INT	•			
782	-	PmeanA	INT		•		
786	-	PmeanB	INT		•		
790	-	PmeanC	INT		•		
794	4	QmeanT	INT	•			
798	-	QmeanA	INT		•		
802	-	QmeanB	INT		•		
806	-	QmeanC	INT		•		
810	6	SmeanT	INT	•			
814	-	SmeanA	INT		•		
818	-	SmeanB	INT		•		
822	-	SmeanC	INT		•		
826	-	PFmeanT	INT		•		

Register	Offset[1)]	Name	Data type	Read		Write	
				Cyclic	Acyclic	Cyclic	Acyclic
830	-	PFmeanA	INT		•		
834	-	PFmeanB	INT		•		
838	-	PFmeanC	INT		•		
Analog energy registers							
4108	-	APenergyT	UDINT		•		
4116	-	APenergyA	UDINT		•		
4124	-	APenergyB	UDINT		•		
4132	-	APenergyC	UDINT		•		
4140	-	ANenergyT	UDINT		•		
4148	-	ANenergyA	UDINT		•		
4156	-	ANenergyB	UDINT		•		
4164	-	ANenergyC	UDINT		•		
4172	-	RPenergyT	UDINT		•		
4180	-	RPenergyA	UDINT		•		
4188	-	RPenergyB	UDINT		•		
4196	-	RPenergyC	UDINT		•		
4204	-	RNenergyT	UDINT		•		
4212	-	RNenergyA	UDINT		•		
4220	-	RNenergyB	UDINT		•		
4228	-	RNenergyC	UDINT		•		
4236	-	SAenergyT	UDINT		•		
4244	-	SenergyA	UDINT		•		
4252	-	SenergyB	UDINT		•		
4260	-	SenergyC	UDINT		•		
4268	-	SVenergyT	UDINT		•		
4404	8	AEnergyT	DINT	•			
4412	12	REnergyT	DINT	•			
Module configuration							
1026	-	ChanControl	UINT				•
1030	-	IDispTh	UINT				•
1034	-	I_RatioA	UINT				•
1038	-	I_RatioB	UINT				•
1042	-	I_RatioC	UINT				•
1046	-	I_RatioN	UINT				•
Update request							
1050	-	CfgUpdate	UINT				•
1054	-	Cs0Update	UINT				•
1058	-	Cs1Update	UINT				•
1066	-	Cs3Update	UINT				•
1570	-	Cs1UpdateFB	UINT		•		
1578	-	Cs3UpdateFB	UINT		•		
A/D converter status configuration							
1090	-	ZXConfig	UINT				•
1094	-	SagTh	UINT				•
1098	-	PhaseLoseTh	UINT				•
1102	-	INWarnTh0	UINT				•
1106	-	INWarnTh1	UINT				•
1110	-	THDNUTh	UINT				•
1114	-	THDNITh	UINT				•
A/D converter measurement configuration checksum 0							
1154	-	PLconstH	UINT				•
1158	-	PLconstL	UINT				•
1162	-	MeteringMode	UINT				•
A/D converter power calibration checksum 1							
1246	-	PhiA_W	UINT				•
1254	-	PhiB_W	UINT				•
1262	-	PhiC_W	UINT				•
A/D converter RMS value synchronization checksum 3							
1346	-	UGainA_W	UINT				•
1350	-	IGainA_W	UINT				•
1354	-	UoffsetA_W	INT				•
1358	-	IoffsetA_W	INT				•
1362	-	UGainB_W	UINT				•
1366	-	IGainB_W	UINT				•
1370	-	UoffsetB_W	INT				•
1374	-	IoffsetB_W	INT				•
1378	-	UGainC_W	UINT				•
1382	-	IGainC_W	UINT				•
1386	-	UoffsetC_W	INT				•
1390	-	IoffsetC_W	INT				•
1394	-	IGainN_W	UINT				•
1398	-	IoffsetN_W	INT				•

Register description

Register	Offset[1]	Name	Data type	Read		Write	
				Cyclic	Acyclic	Cyclic	Acyclic
A/D converter power calibration – read							
1758	-	PhiA_R	UINT		•		
1766	-	PhiB_R	UINT		•		
1774	-	PhiC_R	UINT		•		
A/D converter RMS value synchronization – read							
1858	-	UGainA_R	UINT		•		
1862	-	IGainA_R	UINT		•		
1866	-	UoffsetA_R	INT		•		
1870	-	IoffsetA_R	INT		•		
1874	-	UGainB_R	UINT		•		
1878	-	IGainB_R	UINT		•		
1882	-	UoffsetB_R	INT		•		
1886	-	IoffsetB_R	INT		•		
1890	-	UGainC_R	UINT		•		
1894	-	IGainC_R	UINT		•		
1898	-	UoffsetC_R	INT		•		
1902	-	IoffsetC_R	INT		•		
1906	-	IGainN_R	UINT		•		
1910	-	IoffsetN_R	INT		•		
Flatstream interface							
2049	-	OutputMTU	USINT				•
2051	-	InputMTU	USINT				•
2055	-	FlatstreamMode	USINT				•
2057	-	Forward	USINT				•
2059	-	ForwardDelay	USINT				•
2113	16	InputSequence	USINT	•			
2113 + 2*N	16 + N	RxByteN (index N = 1 to 7)	USINT	•			
2177	16	OutputSequence	USINT			•	
2177 + 2*N	16 + N	TxByteN (index N = 1 to 7)	USINT			•	
Force analog energy registers							
2316	-	Frc_APenergyT	UDINT				•
2324	-	Frc_APenergyA	UDINT				•
2332	-	Frc_APenergyB	UDINT				•
2340	-	Frc_APenergyC	UDINT				•
2348	-	Frc_ANenergyT	UDINT				•
2356	-	Frc_ANenergyA	UDINT				•
2364	-	Frc_ANenergyB	UDINT				•
2372	-	Frc_ANenergyC	UDINT				•
2380	-	Frc_RPenergyT	UDINT				•
2388	-	Frc_RPenergyA	UDINT				•
2396	-	Frc_RPenergyB	UDINT				•
2404	-	Frc_RPenergyC	UDINT				•
2412	-	Frc_RNenergyT	UDINT				•
2420	-	Frc_RNenergyA	UDINT				•
2428	-	Frc_RNenergyB	UDINT				•
2436	-	Frc_RNenergyC	UDINT				•
2444	-	Frc_SAenergyT	UDINT				•
2452	-	Frc_SenergyA	UDINT				•
2460	-	Frc_SenergyB	UDINT				•
2468	-	Frc_SenergyC	UDINT				•
2476	-	Frc_SVenergyT	UDINT				•
2484	-	Frc_APenergyTF	UDINT				•
2492	-	Frc_APenergyAF	UDINT				•
2500	-	Frc_APenergyBF	UDINT				•
2508	-	Frc_APenergyCF	UDINT				•
2516	-	Frc_ANenergyTF	UDINT				•
2524	-	Frc_ANenergyAF	UDINT				•
2532	-	Frc_ANenergyBF	UDINT				•
2540	-	Frc_ANenergyCF	UDINT				•
2548	-	Frc_APenergyTH	UDINT				•
2556	-	Frc_APenergyAH	UDINT				•
2564	-	Frc_APenergyBH	UDINT				•
2572	-	Frc_APenergyCH	UDINT				•
2580	-	Frc_ANenergyTH	UDINT				•
2588	-	Frc_ANenergyAH	UDINT				•
2596	-	Frc_ANenergyBH	UDINT				•
2604	-	Frc_ANenergyCH	UDINT				•
Environment variables							
15108	-	OnTime	UDINT		•		
15116	-	UpCounter	UDINT		•		
15122	-	MinTemp	INT		•		
15126	-	MaxTemp	INT		•		

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

7.4 Status register

7.4.1 Status signals and responses

Name:
StatusInput

The signals are recorded in 200 µs intervals. The energy pulse values in this register are not valid when 1 kWh and 1 Wh are set.

Data type	Value
UINT	See bit structure.

Bit structure:

Bit	Name	Value	Information
0	Energy pulse total active energy	0	Energy increase too low
		1	Energy threshold reached
1	Energy pulse total apparent energy Default: Arithmetic or vector sum (see register "MeteringMode" on page 98, bit 6)	0	Energy increase too low
		1	Energy threshold reached
2	Energy pulse total active energy, fundamental wave	0	Energy increase too low
		1	Energy threshold reached
3	Energy pulse total active energy, harmonics	0	Energy increase too low
		1	Energy threshold reached
4	ZX1 zero cross signal (ZCS) – Phase A	0	Zero crossing not detected
		1	Default: Pulse at rising edge of the zero cross signal of the voltage input, can be reconfigured via register "ZXConfig" on page 95
5	ZX2 zero cross signal (ZCS) – Phase B	0	Zero crossing not detected
		1	Default: Pulse at rising edge of the zero cross signal of the voltage input, can be reconfigured via register "ZXConfig" on page 95
6	ZX3 zero cross signal (ZCS) – Phase C	0	Zero crossing not detected
		1	Default: Pulse at rising edge of the zero cross signal of the voltage input, can be reconfigured via register "ZXConfig" on page 95
7	Reserved	0	
8	DFT response sent	x	If the state in the register "ControlOutput" on page 75 corresponds with the response, then the action is complete
9	Energy value update response sent	0	No update
		1	Update complete
10	Energy value response deleted	x	If the state in the register "ControlOutput" on page 75 corresponds with the response, then the action is complete
11	Energy value response set	x	If the state in the register "ControlOutput" on page 75 corresponds with the response, then the action is complete
12 - 15	Reserved	0	

7.4.2 Control signals

Name:
ControlOutput

Control signals are evaluated in a ~5 ms interval.

Data type	Value
UINT	See bit structure.

Bit structure:

Bit	Name	Value	Information
0	DFT analysis	0	Don't start
		1	Start[1)]
1	Automatically read energy values	0	Do not automatically read
		1	Automatically read
2	Clear energy values	0	Don't delete
		1	Clear[1)]
3	Set energy values	0	Don't start
		1	Start[1)]
4 - 15	Reserved	0	

1) If the state in the register "ControlOutput" on page 75 corresponds with the response, then the action is complete.

7.4.3 Read timestamp for I/O register (+0x0022 = 16-bit)

Name:
SampleTime01_32bit

NetTime timestamp for the readout time of the status, effective value and power registers.

For additional information about NetTime and timestamps, see "NetTime Technology" on page 30.

Data type	Value	Information
DINT	-2,147,483,647 to 2,147,483,647	NetTime timestamp in µs

7.4.4 A/D converter system status 1

Name:
SysStatus1

The register is read by the converter in a ~5 ms interval.

Data type	Value
UINT	See bit structure.

Bit structure:

Bit	Name	Value	Information
0 - 1	Reserved	0	
2	SumStatusPhaseLoss Voltage of one or more phases < failure threshold in the register "PhaseLoseTh" on page 96	0	Voltage within permitted range
		1	Voltage lower than the failure threshold
3	SumStatusPhaseWarning Voltage of one or more phases < warning threshold in the register "SagTh" on page 96	0	Voltage within permitted range
		1	Voltage lower than the warning threshold
4 - 5	Reserved	0	
6	ErrOrderPhasecurrent Error in the order of phase currents	0	No error
		1	Errors
7	ErrOrderPhaseVoltage Error in the order of phase voltages	0	No error
		1	Errors
8	CS3Err Checksum error in configuration block 3	0	No error
		1	Errors
9	Reserved	0	
10	CS2Err Checksum error in configuration block 2	0	No error
		1	Errors
11	Reserved	0	
12	CS1Err Checksum error in configuration block 1	0	No error
		1	Errors
13	Reserved	0	
14	CS0Err Checksum error in configuration block 0	0	No error
		1	Errors
15	Reserved	0	

7.4.5 A/D converter system status 2

Name:
SysStatus2

Data type	Value
UINT	See bit structure.

Bit structure:

Bit	Name	Value	Information
0	RevPchgC Direction of the active energy for phase C has changed	0	No change of direction
		1	Direction has changed
1	RevPchgB Direction of the active energy for phase B has changed	0	No change of direction
		1	Direction has changed
2	RevPchgA Direction of the active energy for phase A has changed	0	No change of direction
		1	Direction has changed
3	RevPchgT Direction of the active energy for the sum has changed	0	No change of direction
		1	Direction has changed
4	RevQchgC Direction of the reactive energy for phase C has changed	0	No change of direction
		1	Direction has changed
5	RevQchgB Direction of the reactive energy for phase B has changed	0	No change of direction
		1	Direction has changed
6	RevQchgA Direction of the reactive energy for phase A has changed	0	No change of direction
		1	Direction has changed
7	RevQchgT Direction of the reactive energy for the total has changed	0	No change of direction
		1	Direction has changed
8	Reserved	0	
9	DFTDone DFT analysis complete (temporary bit)	0	DFT analysis not complete
		1	DFT analysis complete
10	SumStatusWarningTHDCurrent THDIx value of one or more phases > warning threshold in register "THDNITh" on page 96	0	THDIx value within permitted range
		1	THDIx value higher than warning threshold
11	SumStatusWarningTHDVoltage THDUx value of one or more phases > warning threshold in register "THDNUTh" on page 96	0	THDUx value within permitted range
		1	THDUx value higher than warning threshold
12 - 13	Reserved	0	
14	ErrIrmsNCalc Calculated value of the neutral conductor > warning threshold in register "INWarnTh0" on page 96	0	Calculated value within permitted range
		1	Calculated value higher than warning threshold
15	ErrIrmsNMeas Measured value of the neutral conductor > warning threshold in register "INWarnTh1" on page 96	0	Measured value within permitted range
		1	Measured value higher than warning threshold

7.4.6 A/D converter system status 3

Name:
SysStatus3

Data type	Value
UINT	See bit structure.

Bit structure:

Bit	Name	Value	Information
0	CF1RevFlag Direction of energy pulses	0	Forward[1]
		1	Back[2]
1	CF2RevFlag Direction of energy pulses	0	Forward[1]
		1	Back[2]
2	CF3RevFlag Direction of energy pulses	0	Forward[1]
		1	Back[2]
3	CF4RevFlag Direction of energy pulses	0	Forward[1]
		1	Back[2]
4 - 11	Reserved	0	
12	TVSNoload Vector based total apparent power of all phases in "No load" state	0	Status with load
		1	State without load
13	TASNoload Total apparent power of all phases in "No load" state	0	Status with load
		1	State without load
14	TPNoload Total active power of all phases in "No load" state	0	Status with load
		1	State without load
15	TQNoload Total reactive power of all phases in "No load" state	0	Status with load
		1	State without load

1) Forward direction of energy pulses (positive sign of corresponding energy register)
2) Reverse direction of energy pulses (negative sign of corresponding energy register)

7.4.7 A/D converter system status 4

Name:
SysStatus4

Data type	Value
UINT	See bit structure.

Bit structure:

Bit	Name	Value	Information
0	LossPhaseC Voltage < failure threshold in the register "PhaseLoseTh" on page 96	0	Voltage within permitted range
		1	Voltage less than the failure threshold
1	LossPhaseB Voltage < failure threshold in the register "PhaseLoseTh" on page 96	0	Voltage within permitted range
		1	Voltage less than the failure threshold
2	LossPhaseA Voltage < failure threshold in the register "PhaseLoseTh" on page 96	0	Voltage within permitted range
		1	Voltage less than the failure threshold
3	Reserved	0	
4	WarningPhaseC Voltage < Warning threshold in register "SagTh" on page 96	0	Voltage within permitted range
		1	Voltage less than the warning threshold
5	WarningPhaseB Voltage < warning threshold in the register "SagTh" on page 96	0	Voltage within permitted range
		1	Voltage less than the warning threshold
6	WarningPhaseA Voltage < warning threshold in the register "SagTh" on page 96	0	Voltage within permitted range
		1	Voltage less than the warning threshold
7 - 15	Reserved	0	

7.4.8 Selection A/D converter system status 1

Name:
SystemStatusSel01

The most important bits of the "SysStatus1" on page 76 register are stored in this register.

Data type	Value
USINT	See bit structure.

Bit structure:

Bit	Name	Value	Information
0 - 1	Reserved	0	
2	SumStatusPhaseLoss Voltage of one or more phases < failure threshold in the register "PhaseLoseTh" on page 96	0	Voltage within permitted range
		1	Voltage lower than the failure threshold
3	SumStatusPhaseWarning Voltage of one or more phases < warning threshold in the register "SagTh" on page 96	0	Voltage within permitted range
		1	Voltage lower than the warning threshold
4 - 5	Reserved	0	
6	ErrOrderPhasecurrent Error in the order of phase currents	0	No error
		1	Errors
7	ErrOrderPhaseVoltage Error in the order of phase voltages	0	No error
		1	Errors

7.4.9 Selection A/D converter system status 2

Name:
SystemStatusSel02

The most important bits of the "SysStatus2" on page 77 register are stored in this register.

Data type	Value
USINT	See bit structure.

Bit structure:

Bit	Name	Value	Information
0 - 1	Reserved	0	
2	SumStatusWarningTHDCurrent THDIx value of one or more phases > warning threshold in register "THDNITh" on page 96	0	THDIx value within permitted range
		1	THDIx value higher than warning threshold
3	SumStatusWarningTHDVoltage THDUx value of one or more phases > warning threshold in register "THDNUTh" on page 96	0	THDUx value within permitted range
		1	THDUx value higher than warning threshold
4 - 5	Reserved	0	
6	ErrIrmsNCalc The calculated value of the neutral line > warning threshold in the register "INWarnTh0" on page 96	0	Calculated value within permitted range
		1	Calculated value higher than warning threshold
7	ErrIrmsNMeas Measured value of the neutral line > warning threshold in the register "INWarnTh1" on page 96	0	Measured value within permitted range
		1	Measured value higher than warning threshold

7.4.10 Phase status

Name:
PhaseStatus

This register corresponds to the "SysStatus4" on page 78 register. It contains the status of the 3 phases A, B and C.

Data type	Value
UINT	See bit structure.

Bit structure:

Bit	Name	Value	Information
0	LossPhaseC Voltage < failure threshold in the register "PhaseLoseTh" on page 96	0	Voltage within permitted range
		1	Voltage less than the failure threshold
1	LossPhaseB Voltage < failure threshold in the register "PhaseLoseTh" on page 96	0	Voltage within permitted range
		1	Voltage less than the failure threshold
2	LossPhaseA Voltage < failure threshold in the register "PhaseLoseTh" on page 96	0	Voltage within permitted range
		1	Voltage less than the failure threshold
3	Reserved	0	
4	WarningPhaseC Voltage < warning threshold in the register "SagTh" on page 96	0	Voltage within permitted range
		1	Voltage less than the warning threshold
5	WarningPhaseB Voltage < warning threshold in the register "SagTh" on page 96	0	Voltage within permitted range
		1	Voltage less than the warning threshold
6	WarningPhaseA Voltage < warning threshold in the register "SagTh" on page 96	0	Voltage within permitted range
		1	Voltage less than the warning threshold
7 - 15	Reserved	0	

7.5 Analog RMS value registers

7.5.1 Current RMS value neutral line measured

Name:
IrmsN

Measured value of the neutral current between the P and N connections on the current terminal, multiplied with the transfer factor of the transformer.

Data type	Value	Information
UINT	0 to 65535	Measured value 0.001 Arms

7.5.2 Voltage RMS value phase A/B/C

Name:
UrmsA
UrmsB
UrmsC

Measured value for N-terminal or virtual zero point.

Data type	Value	Information
UINT	0 to 65535	Measured value 0.01 Vrms

7.5.3 Current RMS value neutral line calculated

Name:
IrmsNcalc

Calculated value of neutral current derived from the other 3 phases.

Data type	Value	Information
UINT	0 to 65535	Measured value 0.001 Arms

7.5.4 Current RMS value phase A/B/C

Name:
IrmsA
IrmsB
IrmsC

Measured value of the phase current between the P and N connections on the current terminal, multiplied with the transfer factor of the transformer.

Data type	Value	Information
UINT	0 to 65535	Measured value 0.001 Arms

7.6 Analog total harmonic distortion (THD) and angle registers

7.6.1 THD and N value voltage phase A/B/C

Name:
THDNUA
THDNUB
THDNUC

Harmonic ratio = (SQR(RMS value$_{total}$^2 - RMS value$_{fundamental}$^2))/RMS value$_{fundamental}$

Data type	Value	Information
UINT	0 to 10000	Resolution 0.01%

7.6.2 THD and N value of current phase A/B/C

Name:
THDNIA
THDNIB
THDNIC

Harmonics ratio = (SQR(RMS value$_{total}$^2 - RMS value$_{fundamental}$^2))/RMS value$_{fundamental}$

Data type	Values	Information
UINT	0 to 10000	Resolution 0.01%

7.6.3 Fundamental frequency measured

Name:
Freq

Measured fundamental frequency of phases A, B and C.

Data type	Value	Information
UINT	0 to 10000	Resolution 0.01 Hz

7.6.4 Phase angle of power on phase A/B/C

Name:
PAngleA
PAngleB
PAngleC

Middle phase angle (power angle) of the current to the voltage based on the zero-crossing detection.

Data type	Value	Information
INT	-1800 to 1800	Resolution 0.1°

7.6.5 Temperature of the converter

Name:
Temperature

This register contains the internal temperature of the transformer component. The temperature is recorded in a 100 ms interval.

Data type	Value	Information
INT	-200 to 200	Resolution 1°C

7.6.6 Phase angle of voltage on phase A/B/C

Name:
UAngleA
UAngleB
UAngleC

The value for phase A is always 0. On the other phases, the angle corresponds with the offset to A. This is based on the zero-crossing detection.

Data type	Value	Information
INT	-1800 to 1800	Resolution 0.1°

7.7 Analog power registers

7.7.1 Vector sum of total apparent power decimal places

Name:
SVmeanTLSB

The value in this register corresponds to a quarter of the actual power and can be converted in the application according to the example below.

Conversion example

Actual vector sum of the total apparent power = ((REAL)SVmeanT * 4) + ((REAL)SVmeanTLSB * 4/256)

Data type	Value	Information
UINT	0 to 65535	Resolution unit/LSB corresponds to 4/265 VA.

7.7.2 Vector sum of the total apparent power MSW

Name:
SVmeanT

The value in the register equals a fourth of the actual power. The calculation is made in accordance with IEEE 1459.

This value must be multiplied by 4 by the application. Calculation formula for actual power:

Actual vector sum of the total apparent power MSW = register value * 4 (complex sum)

Data type	Value	Information
UINT	0 to 32767	Resolution 4 VA

7.7.3 Total active power

Name:
PmeanT

The value in the register equals a fourth of the actual power. The calculation can be performed in either absolute or arithmetic mode (see register "MeteringMode" on page 98 <Bit 3>). Each phase can be separately enabled for the power calculation (see register "MeteringMode" on page 98 <Bits 0, 1 and 2>).

This value must be multiplied by 4 by the application. Calculation formula for actual power:

Actual total active power = Register value * 4

Data type	Value	Information
INT	-32767 to 32767	Resolution 4 W

7.7.4 Active power on phase A/B/C

Name:
PmeanA
PmeanB
PmeanC

Active power on the phase. Each phase can be separately enabled for the power calculation (see register "MeteringMode" on page 98 <Bits 0, 1 and 2>).

Data type	Value	Information
INT	-32767 to 32767	Resolution 1 W

7.7.5 Total reactive power

Name:
QmeanT

The value in the register equals a fourth of the actual power. The calculation can be performed in either absolute or arithmetic mode (see register "MeteringMode" on page 98 <Bit 4>). Each phase can be separately enabled for the power calculation (see register "MeteringMode" on page 98 <Bits 0, 1 and 2>).

This value must be multiplied by 4 by the application. Calculation formula for actual power:

Actual total reactive power = Register value * 4

Data type	Value	Information
INT	-32767 to 32767	Resolution 4 var

7.7.6 Reactive power on phase A/B/C

Name:
QmeanA
QmeanB
QmeanC

Reactive power on the phase. Each phase can be separately enabled for the power calculation (see register "MeteringMode" on page 98 <Bits 0, 1 and 2>).

Data type	Value	Information
INT	-32767 to 32767	Resolution 1 var

7.7.7 Total apparent power

Name:
SmeanT

The value in the register equals a fourth of the actual power. The power is calculated in arithmetic mode. Each phase can be separately enabled for the power calculation (see register "MeteringMode" on page 98 <Bits 0, 1 and 2>).

This value must be multiplied by 4 by the application. Calculation formula for actual power:

Actual total apparent power = Register value * 4

Data type	Value	Information
INT	0 to 32767	Resolution 4 VA

7.7.8 Apparent power on phase A/B/C

Name:
SmeanA
SmeanB
SmeanC

Apparent power on the phase. Each phase can be separately enabled for the power calculation (see register "MeteringMode" on page 98 <Bits 0, 1 and 2>).

Data type	Value	Information
INT	0 to 32767	Resolution 1 VA

7.7.9 Total power factor

Name:
PFmeanT

Data type	Value	Information
INT	-1000 to 1000	Resolution 0.001

7.7.10 Power factor on phase A/B/C

Name:
PFmeanA
PFmeanB
PFmeanC

Data type	Value	Information
INT	-1000 to 1000	Resolution 0.001

7.7.11 Total active power of fundamental wave

Name:
PmeanTF

The value in the register equals a fourth of the actual power.

This value must be multiplied by 4 by the application. Calculation formula for actual power:

Actual total active power of fundamental wave = Register value * 4

Data type	Value	Information
INT	-32767 to 32767	Resolution 4 W

7.7.12 Fundamental wave active power on phase A/B/C

Name:
PmeanAF
PmeanBF
PmeanCF

Active power of fundamental wave on the phase.

Data type	Value	Information
INT	-32767 to 32767	Resolution 1 W

7.7.13 Total active power of harmonics

Name:
PmeanTH

The value in the register equals a fourth of the actual power.

This value must be multiplied by 4 by the application. Calculation formula for actual power:

Actual total active power of harmonics = Register value * 4

Data type	Value	Information
INT	-32767 to 32767	Resolution 4 W

7.7.14 Harmonics active power on phase A/B/C

Name:
PmeanAH
PmeanBH
PmeanCH

Active power of harmonics on the phase.

Data type	Value	Information
INT	-32767 to 32767	Resolution 1 W

7.8 Analog energy registers

The following information applies to all analog energy registers in this section with the exception of register "SampleTime02_32bit":

Information:

- **The register is updated automatically after being enabled, see register "ControlOutput" on page 75 <bit 1>**
- **The register is deleted upon request, see register "ControlOutput" on page 75 <bit 2>**
- **The register is set upon request, see register "ControlOutput" on page 75 <bit 3>**
- **For information about energy unit; see register "Power line constants" on page 97**

7.8.1 Read timestamp for energy register (+0x0022 = 16-bit)

Name:
SampleTime02_32bit

NetTime timestamp for the readout time of the energy registers.

For additional information about NetTime and timestamps, see "NetTime Technology" on page 30.

Data type	Value	Information
DINT	-2,147,483,647 to 2,147,483,647	NetTime timestamp in µs

7.8.2 Forward total active energy

Name:
APenergyT

Total active power in the forward direction.

See also "Information" on page 85.

Data type	Value	Information
UDINT	0 to 4294967295	Resolution corresponding to setting in register "Power line constants" on page 97.

7.8.3 Forward active energy on phase A/B/C

Name:
APenergyA
APenergyB
APenergyC

Active energy in forward direction of the phase.

See also "Information" on page 85.

Data type	Value	Information
UDINT	0 to 4294967295	Resolution corresponding to setting in register "Power line constants" on page 97.

7.8.4 Reverse total active energy

Name:
ANenergyT

Total active energy in reverse direction.

See also "Information" on page 85.

Data type	Value	Information
UDINT	0 to 4294967295	Resolution corresponding to setting in register "Power line constants" on page 97.

7.8.5 Reverse active energy on phase A/B/C

Name:
ANenergyA
ANenergyB
ANenergyC

Active energy in reverse direction of the phase.

See also "Information" on page 85.

Data type	Value	Information
UDINT	0 to 4294967295	Resolution corresponding to setting in register "Power line constants" on page 97.

7.8.6 Forward total reactive energy

Name:
RPenergyT

Total reactive energy in forward direction.

See also "Information" on page 85.

Data type	Value	Information
UDINT	0 to 4294967295	Resolution corresponding to setting in register "Power line constants" on page 97.

7.8.7 Forward reactive energy on phase A/B/C

Name:
RPenergyA
RPenergyB
RPenergyC

Reactive energy in forward direction of the phase.

See also "Information" on page 85.

Data type	Value	Information
UDINT	0 to 4294967295	Resolution corresponding to setting in register "Power line constants" on page 97.

7.8.8 Reverse total reactive energy

Name:
RNenergyT

Total reactive energy in reverse direction.

See also "Information" on page 85.

Data type	Value	Information
UDINT	0 to 4294967295	Resolution corresponding to setting in register "Power line constants" on page 97.

7.8.9 Reverse reactive energy of the phase A/B/C

Name:
RNenergyA
RNenergyB
RNenergyC

Reactive energy in reverse direction of the phase.

See also "Information" on page 85.

Data type	Value	Information
UDINT	0 to 4294967295	Resolution corresponding to setting in register "Power line constants" on page 97.

7.8.10 Arithmetic total apparent energy

Name:
SAenergyT

See also "Information" on page 85.

Data type	Value	Information
UDINT	0 to 4294967295	Resolution corresponding to setting in register "Power line constants" on page 97.

7.8.11 Apparent energy on phase A/B/C

Name:
SenergyA
SenergyB
SenergyC

See also "Information" on page 85.

Data type	Value	Information
UDINT	0 to 4294967295	Resolution corresponding to setting in register "Power line constants" on page 97.

7.8.12 Vectorized total apparent energy

Name:
SVenergyT

See also "Information" on page 85.

Data type	Value	Information
UDINT	0 to 4294967295	Resolution corresponding to setting in register "Power line constants" on page 97.

7.8.13 Forward fundamental wave total active energy

Name:
APenergyTF

Fundamental wave of total active energy in forward direction.

See also "Information" on page 85.

Data type	Value	Information
UDINT	0 to 4294967295	Resolution corresponding to setting in register "Power line constants" on page 97.

7.8.14 Forward fundamental wave active energy on phase A/B/C

Name:
APenergyAF
APenergyBF
APenergyCF

Fundamental wave of active energy in forward direction of the phase.

See also "Information" on page 85.

Data type	Value	Information
UDINT	0 to 4294967295	Resolution corresponding to setting in register "Power line constants" on page 97.

7.8.15 Reverse fundamental wave total active energy

Name:
ANenergyTF

Fundamental wave of total active energy in reverse direction.

See also "Information" on page 85.

Data type	Value	Information
UDINT	0 to 4294967295	Resolution corresponding to setting in register "Power line constants" on page 97.

7.8.16 Reverse fundamental wave active energy on phase A/B/C

Name:
ANenergyAF
ANenergyBF
ANenergyCF

Fundamental wave of active energy in reverse direction of the phase.

See also "Information" on page 85.

Data type	Value	Information
UDINT	0 to 4294967295	Resolution corresponding to setting in register "Power line constants" on page 97.

7.8.17 Forward harmonics total active energy

Name:
APenergyTH

Harmonics of total active energy in forward direction.

See also "Information" on page 85.

Data type	Value	Information
UDINT	0 to 4294967295	Resolution corresponding to setting in register "Power line constants" on page 97.

7.8.18 Forward harmonics active energy on phase A/B/C

Name:
APenergyAH
APenergyBH
APenergyCH

Harmonics of active energy in forward direction of the phase.

See also "Information" on page 85.

Data type	Value	Information
UDINT	0 to 4294967295	Resolution corresponding to setting in register "Power line constants" on page 97.

7.8.19 Reverse harmonics total active energy

Name:
ANenergyTH

Harmonics of total active energy in forward direction.

See also "Information" on page 85.

Data type	Value	Information
UDINT	0 to 4294967295	Resolution corresponding to setting in register "Power line constants" on page 97.

7.8.20 Reverse harmonics active energy on phase A/B/C

Name:
ANenergyAH
ANenergyBH
ANenergyCH

Harmonics of active energy in reverse direction of the phase.

See also "Information" on page 85.

Data type	Value	Information
UDINT	0 to 4294967295	Resolution corresponding to setting in register "Power line constants" on page 97.

7.8.21 Total active energy combined

Name:
AEnergyT

Total active energy in forward and backward direction.

Internal calculation formula for the total active energy:

AEnergyT = (DINT)(APenergyT - ANenergyT) Calculation overflows are ignored

See also "Information" on page 85.

Data type	Value	Information
DINT	-2,147,483,647 to 2,147,483,647	Resolution corresponding to setting in register "Power line constants" on page 97.

7.8.22 Total reactive energy combined

Name:
REnergyT

Total reactive power in the forward and reverse directions.

Internal calculation formula for the total reactive energy:

REnergyT = (DINT)(RPenergyT - RNenergyT) ... Calculated overflows are not processed.

See also "Information" on page 85.

Data type	Value	Information
DINT	-2,147,483,647 to 2,147,483,647	Resolution corresponding to setting in register "Power line constants" on page 97.

7.9 Analog discrete Fourier transformation register (DFT)

7.9.1 Read timestamp for DFT register (+0x0022 = 16-bit)

Name:
SampleTime03_32bit

NetTime timestamp for the readout time of the DFT registers.

Data type	Value	Information
DINT	-2,147,483,647 to 2,147,483,647	NetTime timestamp in µs

7.9.2 Harmonic distortion register (HD) current I and voltage V for phases A/B/C

Name:
DftAI0 to DftAI30
DftAV0 to DftAV30
DftBI0 to DftBI30
DftBV0 to DftBV30
DftCI0 to DftCI30
DftCV0 to DftCV30

Ratio of 2nd to 32nd order harmonic wave components.

Conversion from % = register value / 163.84

Data type	Value	Information
UINT	0 to 32767	Ratio of frequency component

7.9.3 THD register current I and voltage V for phases A/B/C

Name:
DftAI31
DftAV31
DftBI31
DftBV31
DftCI31
DftCV31

Ratio of total harmonic distortion.

Conversion from % = register value / 163.84

Data type	Value	Information
UINT	0 to 32767	Total harmonic distortion

7.9.4 Fundamental wave current on phase A/B/C

Name:
DftAI_Fund
DftBI_Fund
DftCI_Fund

Calculation of the fundamental wave current. For details, see "Calculating the fundamental wave current " on page 26.

Data type	Values	Information
UINT	0 to 32767	Fundamental wave current in mA

7.9.5 Fundamental wave voltage on phase A/B/C

Name:
DftAV_Fund
DftBV_Fund
DftCV_Fund

Calculation of the fundamental wave voltage. For details, see "Calculating the fundamental wave voltage" on page 27.

Data type	Values	Information
UINT	0 to 32767	Fundamental value voltage in volts

7.10 Environment variables

7.10.1 Operating time

Name:
ulUpTime

Total operating time of the module.

Data type	Value	Information
UDINT	0 to 4,294,967,295	Operating time in seconds

7.10.2 Switch-on and reset counter

Name:
ulUpCount

Switch-on and reset counter of the module.

Data type	Value	Information
UDINT	0 to 4,294,967,295	Switch-on and reset counter

7.10.3 Minimum operating temperature

Name:
ssMinTemp

Lowest measured module temperature since the last time it was started.

Data type	Value	Information
INT	-200 to 200	Resolution 1°C

7.10.4 Maximum operating temperature

Name:
ssMaxTemp

Highest measured module temperature since the last time it was started.

Data type	Value	Information
INT	-200 to 200	Resolution 1°C

7.11 Module configuration

7.11.1 Mode register

Name:
ChanControl

Data type	Value	Bus controller default setting
UINT	See bit structure.	15

Bit structure:

Bit	Name	Value	Information
0	Channel status LED for phase A	0	Off
		1	On (bus controller default setting)
1	Channel status LED for phase B	0	Off
		1	On (bus controller default setting)
2	Channel status LED for phase C	0	Off
		1	On (bus controller default setting)
3	Reserved	0	
4	Neutral current monitor and status LED	0	Off (bus controller default setting)
		1	On
5	Neutral current status derived from the calculated or measured value	0	Derived from the calculated value (bus controller default setting)
		1	Derived from the measured value
6	Conversion of energy register to Wh and kWh[2)]	0	Disabled (1 Ws, 10 Ws, 100 Ws, 1 kWs) (bus controller default setting)
		1	Enabled (1 Wh and 1 kWh)
7	Display current values despite power failure[3)]	0	Off[3)] (bus controller default setting)
		1	On
8 - 15	Oversampling with prescaler	0	Disabled (bus controller default setting)
		1 - 255	Enabled Sample cycle time as a multiple of 125 µs; only in the "Oversampling" function model

1) When 1 Wh and 1 kWh are set, the energy pulses on the register "Status signals and responses" on page 75 may not be used.
2) When a power failure occurs, all current values are held at 0 by default.
3) According to the power failure status of the individual phases, the following values are held at 0 by default.

- Mains frequency, phase angle, power factor
- Effective voltage and current values
- Active, reactive and apparent power values

7.11.2 Analog minimum current for active current channel LEDs

Name:
IDispTh

The display threshold defines the effective value of the current above which the phase current status LED is switched on. The default values are module-dependent and should be adapted to the maximum primary current.
Suggestion: 1% of the maximum value

Data type	Value	Information
UINT	1 to 65000	RMS indicator threshold in mA. Bus controller default setting:

Module	Indicator threshold
X20AP3111	200 mA
X20AP3121/22	500 mA
X20AP3131/32	500 mA
X20AP3161	500 mA
X20AP3171	500 mA

7.11.3 Current transformer rating phase A/B/C/N

Name:
I_RatioA
I_RatioB
I_RatioC
I_RatioN

The following current transformer measurements are applied in these registers. The permissible values are module-dependent (0.1 resolution).

- **X20AP3111, 3121/22 and 3131/32:** The measured current is multiplied by the current transformation ratio.
- **X20AP3161:** The maximum primary current of the transformer is configured.
- **X20AP3171:** The current transformation ratio of the Rogowski coil is entered. This is the voltage in µV that the coil provides at 10 A primary current (0.1 µV/A).

Data type	Value	Information
UINT	x	Current transformer measurement. Bus controller default setting:

Module	Rating
X20AP3111	Transformation ratio: 10 to 32500. Bus controller default setting: 25000
X20AP3121/22	Transformation ratio: 10 to 650. Bus controller default setting: 500
X20AP3131/32	Transformation ratio: 10 to 130. Bus controller default setting: 100
X20AP3161	Measurement range: 50 to 650. Bus controller default setting: 500
X20AP3171	Current transformation ratio: 2600 to 8000. Bus controller default setting: 5000

Information:

The maximum resulting current must not exceed the value of 65000 mA.

7.12 Update requests

7.12.1 Update request status configuration register

Name:
CfgUpdate

The registers in section "A/D converter status configuration" on page 95 are only applied after changing this register. Writing with 0xFFFF only resets this register without applying the values.

Data type	Value	Information
UINT	0 to 65535	Update request. Bus controller default setting: 65535

7.12.2 Update request A/D converter Cs0, Cs1 and Cs3 register

Name:
Cs0Update
Cs1Update
Cs3Update

The registers of the respective section are only applied after changing the corresponding CsxUpdate register. These include:

- Cs0Update: 3 registers in section "A/D converter measurement configuration checksum 0" on page 97
- Cs1Update: 3 registers in "A/D converter power calibration checksum 1" on page 101
- Cs3Update: 14 registers in section "A/D converter RMS value synchronization checksum 3" on page 100

Writing with 0xFFFF only resets the affected register without applying the value.

Data type	Value	Information
UINT	0 to 65535	Update request. Bus controller default setting: 65535

7.12.3 Reading update request A/D converter Cs1 and Cs3 register

Name:
Cs1UpdateFB
Cs3UpdateFB

The A/D converter configuration registers in the sections "A/D converter status configuration" on page 95 and "A/D converter measurement configuration checksum 0" on page 97 are only transferred to the feedback buffer after transfer to the A/D converter is complete.

Data type	Value	Information
UINT	0 to 65535	

7.13 A/D converter status configuration

Changes in the registers in this section are only applied after an update request in register "CfgUpdate" on page 93.

7.13.1 A/D converter hardware signal pinout

Name:
ZXConfig

Data type	Value	Bus controller default setting
UINT	See bit structure.	0x4400

Bit structure:

Bit	Name	Value	Information
0	Zero cross signals	0	Enabled (bus controller default setting)
		1	Disabled
1 - 2	ZX20Con Trigger zero crossing	00	Positive zero crossing (bus controller default setting)
		01	Negative zero crossing
		10	Both zero cross-overs
		11	No zero crossing
3 - 4	ZX1Con Trigger zero crossing	00	Positive zero crossing (bus controller default setting)
		01	Negative zero crossing
		10	Both zero cross-overs
		11	No zero crossing
5 - 6	ZX2Con Trigger zero crossing	00	Positive zero crossing (bus controller default setting)
		01	Negative zero crossing
		10	Both zero cross-overs
		11	No zero crossing
7 - 9	ZX0Src Signal source for ZX0 hardware signal	000	Voltage A (bus controller default setting)
		001	Voltage B
		010	Voltage C
		011	Fix 0
		100	Current A
		101	Current B
		110	Current C
		111	Fix 0
10 - 12	ZX1Src Signal source for ZX1 hardware signal	000	Voltage A
		001	Voltage B (bus controller default setting)
		010	Voltage C
		011	Fix 0
		100	Current A
		101	Current B
		110	Current C
		111	Fix 0
13 - 15	ZX2Src Signal source for ZX2 hardware signal	000	Voltage A
		001	Voltage B
		010	Voltage C (bus controller default setting)
		011	Fix 0
		100	Current A
		101	Current B
		110	Current C
		111	Fix 0

7.13.2 Voltage warning threshold

Name:
SagTh

This register defines an RMS voltage value for monitoring the voltage warning signals.

Data type	Value	Information
UINT	5000 to 50000	Resolution 0.01 V. Bus controller default setting: 12368

7.13.3 Power failure threshold

Name:
PhaseLoseTh

This register defines an RMS voltage value for monitoring the power failure signals.

Data type	Value	Information
UINT	1000 to 6000	Resolution 0.01 V. Bus controller default setting: 2420

7.13.4 Warning threshold for the calculated neutral current

Name:
INWarnTh0

Current value for monitoring the calculated neutral line current.

Data type	Value	Information
UINT	0 to 65000	Resolution 0.001 A. Bus controller default setting: 50

7.13.5 Warning threshold for the measured neutral current.

Name:
INWarnTh1

Current value for monitoring the measured neutral line current.

Data type	Value	Information
UINT	0 to 65000	Resolution 0.001 A. Bus controller default setting: 50

7.13.6 Warning threshold for voltage THD overshoot

Name:
THDNUTh

Percentage value defining warning threshold for THD ratio.

Data type	Value	Information
UINT	0 to 10000	Resolution 0.01%. Bus controller default setting: 1000

7.13.7 Warning threshold for current THD overshoot

Name:
THDNITh

Percentage value defining warning threshold for THD ratio.

Data type	Value	Information
UINT	0 to 10000	Resolution 0.01%. Bus controller default setting: 1000

7.14 A/D converter measurement configuration checksum 0

Changes in the registers in this section are only applied after an update request in register "Cs0Update" on page 94.

7.14.1 Power line constants

Name:
PLconstH
PLconstL

Base value for power line constant.

10 increments in the energy register result in 1 energy pulse. Base value 0x4A81 7C80 = 1,250,000,000 corresponds to 360 energy pulses per kWh or 0.1 energy pulses per kWs. In the energy registers, this results in 1 kWs per position.

The two registers can be set to the following values. Other values are not allowed

Data type	PLConstH	PLConstL	1 increment in the energy register corresponds to:
UINT	0x0013	0x12D0	1 Ws[1)]
	0x00BE	0xBC20	10 Ws[1)]
	0x0773	0x5940	100 Ws[1)]
	0x4A81	0x7C80	1 kWs[1)] (bus controller default setting)
	0x0010	0x0034	1 Wh[2)]
	0x417B	0xCE6C	1 kWh[2)]

1) Register "ChanControl" on page 92, bit 6 = 0
2) Register "ChanControl" on page 92, bit 6 = 1

Information:

When 1 Wh and 1 kWh are set, the energy pulses on the register "StatusInput" on page 75 may not be used.

7.14.2 Analog A/D converter measurement setting 1

Name:
MeteringMode

Data type	Value	Bus controller default setting
UINT	See bit structure.	135

Bit structure:

Bit	Name	Value	Information
0	Enables phase C for adding the power and energy values together	0	Not released
		1	Enabled (bus controller default setting)
1	Enables phase B for adding the power and energy values together	0	Not released
		1	Enabled (bus controller default setting)
2	Enables phase A for adding the power and energy values together	0	Not released
		1	Enabled (bus controller default setting)
3	Calculation method for adding active power and active energy	0	Arithmetic sum (bus controller default setting)
		1	Absolute sum
4	Calculation method for adding reactive power and reactive energy	0	Arithmetic sum (bus controller default setting)
		1	Absolute sum
5	Reserved	0	
6	Selects apparent energy for Energypulse2-Source	0	Arithmetic sum (bus controller default setting)
		1	Vector sum
7	Energypulse2-Source	0	Apparent energy
		1	Reactive energy (bus controller default setting)
8	Measuring configuration	0	3P4W (bus controller default setting)
		1	3P3W
9	Resolution of energy register	0	Must be 0!
10	Integrator for DIDT current transformer	0	Off (bus controller default setting)
		1	On
11	High-pass filter	0	On (bus controller default setting)
		1	Off
12	Basis frequency	0	50 Hz (bus controller default setting)
		1	60 Hz
13	Phase assignment	0	I1 to phase A and I3 to phase C (bus controller default setting)
		1	I1 to Phase C and I3 to Phase A
14 - 15	Reserved	0	

Comments regarding measurement configurations:

Measuring configuration	Note
3P4W	Monitors the phasing of voltages and currents: Phase A before phase B before phase C
3P3W	Measuring configuration: Phase A and phase C, N connection bridges to phase B or open
	Measurement: e.g. the 2 phases A and C and the 2 corresponding currents are measured, phase B disabled
	Monitors the phasing of voltages and currents: Phase difference between A and C >180°

7.15 A/D converter RMS value synchronization – read

7.15.1 Voltage RMS value gain phase A/B/C

Name:
UGainA_R
UGainB_R
UGainC_R

Data type	Value
UINT	0 to 65535

7.15.2 Current RMS value gain phase A/B/C/N

Name:
IGainA_R
IGainB_R
IGainC_R
IGainN_R

Data type	Value
UINT	0 to 65535

7.15.3 Voltage RMS value offset phase A/B/C

Name:
UoffsetA_R
UoffsetB_R
UoffsetC_R

Data type	Value
INT	-32767 to 32767

7.15.4 Current RMS value offset phase A/B/C/N

Name:
IoffsetA_R
IoffsetB_R
IoffsetC_R
IoffsetN_R

Data type	Value
INT	-32767 to 32767

7.16 A/D converter RMS value synchronization checksum 3

Changes in the registers in this section are only applied after an update request in register "Cs3Update" on page 94.

7.16.1 Voltage RMS value gain phase A/B/C

Name:
UGainA_W
UGainB_W
UGainC_W

The resulting gain is calculated using the following formula:

$Value_{new} = Value_{old} * correction\ factor$, determined when $U = U_n$

Data type	Value	Information
UINT	0 to 65535	Voltage RMS value gain, phase-based. Bus controller default setting: 26400

7.16.2 Current RMS value gain phase A/B/C/N

Name:
IGainA_W
IGainB_W
IGainC_W
IGainN_W

The resulting gain is calculated using the following formula:

$Value_{new} = Value_{old} * correction\ factor$, determined when $I = I_n$

Data type	Value	Information
UINT	0 to 65535	Current RMS value gain, phase-based. Bus controller default setting: X20AP3111, X20AP312x: 31248 X20AP313x: 38704 X20AP3161: 23339 X20AP3171: 16653

7.16.3 Voltage RMS value offset phase A/B/C

Name:
UoffsetA_W
UoffsetB_W
UoffsetC_W

Corresponds to the negated value of the corresponding RMS value register when U = 0.

Data type	Value	Information
INT	-32767 to 32767	RMS value voltage offset, phase-based. Bus controller default setting: 0

7.16.4 Current RMS value offset phase A/B/C/N

Name:
IoffsetA_W
IoffsetB_W
IoffsetC_W
IoffsetN_W

Corresponds to the negated value of the corresponding RMS value register when I = 0.

Data type	Value	Information
INT	-32767 to 32767	RMS value current offset, phase-based. Bus controller default setting: 0

7.17 User calibration of power values

7.17.1 A/D converter power angle correction phase A/B/C

Name:
PhiA_R
PhiB_R
PhiC_R

These registers can be used to read out the configured values at runtime, but are not nonvolatile and have the value 0 after the system is started.

Data type	Values
UINT	See the bit structure.

Bit structure:

Bit	Name	Value	Information
0 - 9	Delay time for energy phase angle correction	x	The clock base is 2.048 MHz. Maximum 0.499 ms
10 - 14	Reserved	0	
15	Delay times	0	Effect on current channel
		1	Effect on voltage channel

7.17.2 A/D converter power calibration checksum 1

Name:
PhiA_W
PhiB_W
PhiC_W

These registers can be used to correct phase shifts at runtime. This can be necessary if the transformers used distort the phase shift.

Changes in these registers are only applied after an update request in register "Cs1Update" on page 94.

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

Bit structure:

Bit	Name	Value	Information
0 - 9	Delay time for energy phase angle correction	0 to 1023	See description of "bits 0 to 9" on page 62. Bus controller default setting: 0
10 - 14	Reserved	0	
15	Delay times	0 or 1	See description of "bit 15" on page 62.

7.18 Force analog energy registers

The registers are described under "Analog energy registers" on page 85. A corresponding comparison attached:

Force registers	Read registers
Frc_APenergyT Frc_APenergyTF Frc_APenergyTH	"APenergyT"
Frc_APenergyA Frc_APenergyAF Frc_APenergyAH	"APenergyA"
Frc_APenergyB Frc_APenergyBF Frc_APenergyBH	"APenergyB"
Frc_APenergyC Frc_APenergyCF Frc_APenergyCH	"APenergyC"
Frc_ANenergyT Frc_ANenergyTF Frc_ANenergyTH	"ANenergyT"
Frc_ANenergyA Frc_ANenergyAF Frc_ANenergyAH	"ANenergyA"
Frc_ANenergyB Frc_ANenergyBF Frc_ANenergyBH	"ANenergyB"
Frc_ANenergyC Frc_ANenergyCF Frc_ANenergyCH	"ANenergyC"
Frc_RPenergyT	"RPenergyT"
Frc_RPenergyA	"RPenergyA"
Frc_RPenergyB	"RPenergyB"
Frc_RPenergyC	"RPenergyC"
Frc_RNenergyT	"RNenergyT"
Frc_RNenergyA	"RNenergyA"
Frc_RNenergyB	"RNenergyB"
Frc_RNenergyC	"RNenergyC"
Frc_SAenergyT	"SAenergyT"
Frc_SenergyA	"SenergyA"
Frc_SenergyB	"SenergyB"
Frc_SenergyC	"SenergyC"
Frc_SVenergyT	"SVenergyT"

After a module replacement, these registers can be used to set the energy meters to a specific value. The registers are transferred to the current values by triggering via register "ControlOutput" on page 75, bit 3.

Data type	Value	Information
UDINT	0 to 4294967295	Bus controller default setting: 0

7.19 Oversampling buffer

7.19.1 Sample - Neutral current

Name:
IactN_Sample1 to IactN_Sample16

Current value of the neutral current.
The value of these registers must be converted by the application: See "Sample lines" on page 29.

Data type	Value	Information
INT	-32767 to 32767	Resolution 0.001 A

7.19.2 Sample - Current on phase A

Name:
IactA_Sample1 to IactA_Sample16

Current value of the current of phase A.
The value of these registers must be converted by the application: See "Sample lines" on page 29.

Data type	Value	Information
INT	-32767 to 32767	Resolution 0.001 A

7.19.3 Sample - Voltage on phase A

Name:
UactA_Sample1 to UactA_Sample16

Current value of the voltage of phase A.
The value of these registers must be converted by the application: See "Sample lines" on page 29.

Data type	Value	Information
INT	-32767 to 32767	Resolution 0.01 V

7.19.4 Sample - Current on phase B

Name:
IactB_Sample1 to IactB_Sample16

Current value of the current of phase B.
The value of these registers must be converted by the application: See "Sample lines" on page 29.

Data type	Value	Information
INT	-32767 to 32767	Resolution 0.001 A

7.19.5 Sample - Voltage of phase B

Name:
UactB_Sample1 to UactB_Sample16

Current value of the voltage of phase B.
The value of these registers must be converted by the application: See "Sample lines" on page 29.

Data type	Values	Information
INT	-32767 to 32767	Resolution 0.01 V

7.19.6 Sample - Current on phase C

Name:
IactC_Sample1 to IactC_Sample16

Current value of the voltage of phase C.
The value of these registers must be converted by the application: See "Sample lines" on page 29.

Data type	Value	Information
INT	-32767 to 32767	Resolution 0.001 A

7.19.7 Sample - Voltage on phase C

Name:
UactC_Sample1 to UactC_Sample16

Current value of the voltage of phase C.
The value of these registers must be converted by the application: See "Sample lines" on page 29.

Data type	Value	Information
INT	-32767 to 32767	Resolution 0.01 V

7.19.8 Sample number

Name:
SampleCount1 to Samplecount16

Sample line number, ascending, cyclic.
Number of new sample lines since last readout.

Data type	Value
SINT	-127 to 127
INT	-32767 to 32767

7.19.9 Sample time

Name:
Timestamp

NetTime timestamp of sample line 1.
The older sample lines must be recalculated with 125 µs each.

For additional information about NetTime and timestamps, see "NetTime Technology" on page 30.

Data type	Value
INT	-32767 to 32767
DINT	-2,147,483,647 to 2,147,483,647

7.20 Environment variables

7.20.1 Operating time in seconds

Name:
OnTime

The operating time since startup is saved in seconds in this register.

Data type	Value
UDINT	0 to 4294967295

7.20.2 Startup counter

Name:
UpCounter

The number of restarts since startup is saved in this register.

Data type	Value
UDINT	0 to 4294967295

7.20.3 Minimum operating temperature

Name:
MinTemp

The lowest transformer temperature [°C] since startup is saved in this register.

Data type	Value	Information
INT	-200 to 200	Resolution 1°C

7.20.4 Maximum operating temperature

Name:
MaxTemp

The highest transformer temperature [°C] since startup is saved in this register.

Data type	Value	Information
INT	-200 to 200	Resolution 1°C

7.21 Flatstream registers

At the absolute minimum, registers "InputMTU" and "OutputMTU" must be set. All other registers are filled in with default values at the beginning and can be used immediately. These registers are used for additional options, e.g. to transfer data in a more compact way or to increase the efficiency of the general procedure.

Information:

For detailed information about Flatstream, see Flatstream communication.

7.21.1 Number of enabled Tx and Rx bytes

Name:
OutputMTU
InputMTU

These registers define the number of enabled Tx or Rx bytes and thus also the maximum size of a sequence. The user must consider that the more bytes made available also means a higher load on the bus system.

Data type	Values
USINT	See the register overview.

7.21.2 Transporting payload data and control bytes

Name:
TxByte1 to TxByteN
RxByte1 to RxByteN

(The value the number N is different depending on the bus controller model used.)

The Tx and Rx bytes are cyclic registers used to transport the payload data and the necessary control bytes. The number of active Tx and Rx bytes is taken from the configuration of registers "OutputMTU" and "InputMTU", respectively.

- "T" - "Transmit" → Controller transmits data to the module.
- "R" - "Receive" → Controller receives data from the module.

Data type	Values
USINT	0 to 255

7.21.3 Communication status of the controller

Name:
OutputSequence

This register contains information about the communication status of the controller. It is written by the controller and read by the module.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Name	Value	Information
0 - 2	OutputSequenceCounter	0 - 7	Counter for the sequences issued in the output direction
3	OutputSyncBit	0	Output direction (disable)
		1	Output direction (enable)
4 - 6	InputSequenceAck	0 - 7	Mirrors InputSequenceCounter
7	InputSyncAck	0	Input direction not ready (disabled)
		1	Input direction ready (enabled)

OutputSequenceCounter

The OutputSequenceCounter is a continuous counter of sequences that have been issued by the controller. The controller uses OutputSequenceCounter to direct the module to accept a sequence (the output direction must be synchronized when this happens).

OutputSyncBit

The controller uses OutputSyncBit to attempt to synchronize the output channel.

InputSequenceAck

InputSequenceAck is used for acknowledgment. The value of InputSequenceCounter is mirrored if the controller has received a sequence successfully.

InputSyncAck

The InputSyncAck bit acknowledges the synchronization of the input channel for the module. This indicates that the controller is ready to receive data.

7.21.4 Communication status of the module

Name:
InputSequence

This register contains information about the communication status of the module. It is written by the module and should only be read by the controller.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Name	Value	Information
0 - 2	InputSequenceCounter	0 - 7	Counter for sequences issued in the input direction
3	InputSyncBit	0	Not ready (disabled)
		1	Ready (enabled)
4 - 6	OutputSequenceAck	0 - 7	Mirrors OutputSequenceCounter
7	OutputSyncAck	0	Not ready (disabled)
		1	Ready (enabled)

InputSequenceCounter

The InputSequenceCounter is a continuous counter of sequences that have been issued by the module. The module uses InputSequenceCounter to direct the controller to accept a sequence (the input direction must be synchronized when this happens).

InputSyncBit

The module uses InputSyncBit to attempt to synchronize the input channel.

OutputSequenceAck

OutputSequenceAck is used for acknowledgment. The value of OutputSequenceCounter is mirrored if the module has received a sequence successfully.

OutputSyncAck

The OutputSyncAck bit acknowledges the synchronization of the output channel for the controller. This indicates that the module is ready to receive data.

7.21.5 Flatstream mode

Name:
FlatstreamMode

A more compact arrangement can be achieved with the incoming data stream using this register.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Name	Value	Information
0	MultiSegmentMTU	0	Not allowed (default)
		1	Permitted
1	Large segments	0	Not allowed (default)
		1	Permitted
2 - 7	Reserved		

7.21.6 Number of unacknowledged sequences

Name:
Forward

With register "Forward", the user specifies how many unacknowledged sequences the module is permitted to transmit.

Recommendation:
X2X Link: Max. 5
POWERLINK: Max. 7

Data type	Values
USINT	1 to 7 Default: 1

7.21.7 Delay time

Name:
ForwardDelay

This register is used to specify the delay time in microseconds.

Data type	Values
UINT	0 to 65535 [µs] Default: 0

7.22 Minimum cycle time

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

Minimum cycle time
200 µs

7.23 Minimum I/O update time

The minimum I/O update time specifies how far the bus cycle can be reduced so that an I/O update is performed in each cycle.

Minimum I/O update time	
Voltage and current sampling rate for calculation of RMS value, power and energy	1 MHz
Derived values: RMS value, power, energy, power factor, phase angle, frequency (mean values over 16 full waves)	Approx. 3 Hz
FFT on request (sample rate: 8 kHz)	2 Hz