

X20CMR111

Datenblatt
1.12 (August 2025)



Impressum

B&R Industrial Automation GmbH

B&R Straße 1

5142 Eggelsberg

Österreich

Telefon: +43 7748 6586-0

Fax: +43 7748 6586-26

office@br-automation.com

Disclaimer

Alle Angaben entsprechen dem aktuellen Stand zum Zeitpunkt der Erstellung dieses Dokuments. Jederzeitige inhaltliche Änderungen dieses Dokuments ohne Ankündigung bleiben vorbehalten. B&R Industrial Automation GmbH haftet insbesondere für technische oder redaktionelle Fehler in diesem Dokument unbegrenzt nur (i) bei grobem Verschulden oder (ii) für schuldhaft zugefügte Personenschäden. Darüber hinaus ist die Haftung ausgeschlossen, soweit dies gesetzlich zulässig ist. Eine Haftung in den Fällen, in denen das Gesetz zwingend eine unbeschränkte Haftung vorsieht (wie z. B. die Produkthaftung), bleibt unberührt. Die Haftung für mittelbare Schäden, Folgeschäden, Betriebsunterbrechung, entgangenen Gewinn, Verlust von Informationen und Daten ist ausgeschlossen, insbesondere für Schäden, die direkt oder indirekt auf Lieferung, Leistung und Nutzung dieses Materials zurückzuführen sind.

B&R Industrial Automation GmbH weist darauf hin, dass die in diesem Dokument verwendeten Hard- und Softwarebezeichnungen und Markennamen der jeweiligen Firmen dem allgemeinen warenzeichen-, marken- oder patentrechtlichen Schutz unterliegen.

Hard- und Software von Drittanbietern, auf die in diesem Dokument verwiesen wird, unterliegt ausschließlich den jeweiligen Nutzungsbedingungen dieser Drittanbieter. B&R Industrial Automation GmbH übernimmt hierfür keine Haftung. Allfällige Empfehlungen von B&R Industrial Automation GmbH sind nicht Vertragsinhalt, sondern lediglich unverbindliche Hinweise, ohne dass dafür eine Haftung übernommen wird. Beim Einsatz der Hard- und Software von Drittanbietern sind ergänzend die relevanten Anwenderdokumentationen dieser Drittanbieter heranzuziehen und insbesondere die dort enthaltenen Sicherheitshinweise und technischen Spezifikationen zu beachten. Die Kompatibilität der in diesem Dokument dargestellten Produkte von B&R Industrial Automation GmbH mit Hard- und Software von Drittanbietern ist nicht Vertragsinhalt, es sei denn, dies wurde im Einzelfall gesondert vereinbart; insoweit ist die Gewährleistung für eine solche Kompatibilität jedenfalls ausgeschlossen und hat der Kunde die Kompatibilität in eigener Verantwortung vorab zu prüfen.

1493835589504-1.12

1 Allgemeines

1.1 Mitgeltende Dokumente

Weiterführende und ergänzende Informationen sind den folgenden gelisteten Dokumenten zu entnehmen.

Mitgeltende Dokumente

Dokumentname	Titel
MAX20	X20 System Anwenderhandbuch

1.2 Bestelldaten


Bestellnummer	Kurzbeschreibung	Abbildung
	Sonstige Funktionen	
X20CMR111	X20 Cabinet Monitoring Modul, integr. Temperatur-, Feuchtigkeits- und Beschleunigungssensor, Betriebsdatenaufzeichnung, 2 Eingänge für externe PT1000, 2 digitale Eingänge 24 V, 1 digitaler Ausgang 24 V, 0,5 A, 512 kByte Flash für Anwenderdaten, integrierter Technology Guard	
	Erforderliches Zubehör	
	Busmodule	
X20BM11	X20 Busmodul, 24 VDC codiert, interne I/O-Versorgung durchverbunden	
X20BM15	X20 Busmodul, mit Knotennummernschalter, 24 VDC codiert, interne I/O-Versorgung durchverbunden	
	Feldklemmen	
X20TB12	X20 Feldklemme, 12-polig, 24 VDC codiert	

Tabelle 1: X20CMR111 - Bestelldaten

1.3 Modulbeschreibung

Das Modul ist für die Messung von Umgebungsbedingungen im Schaltschrank sowie für die Aufzeichnung von Betriebsstunden und Einschaltzyklen vorgesehen. Zusätzlich bietet das Modul die Möglichkeit Anwenderdaten direkt am Modul zu speichern, verfügt es über einen Technology Guard sowie Ein-/Ausgänge und ist Blackout-fähig.

Funktionen:

- [Umgebungsbedingungen messen und auswerten](#)
- [Betriebsdaten aufzeichnen](#)
- [Interner Modulspeicher für Anwenderdaten](#)
- [Technology Guarding](#)
- [Konfiguration der Eingänge](#)
- ["Blackout-Modus"](#)
- ["Die Flatstream-Kommunikation"](#)

Umgebungsbedingungen messen und auswerten

Die Umgebungsbedingungen werden vom Modul laufend ausgewertet. Intern wird die Zeitdauer gespeichert, in welcher die einzelnen Parameter innerhalb bestimmter Bereiche liegen. Damit kann ermittelt werden, wie lange sich das System z. B. in einem bestimmten Temperaturbereich befunden hat. Die vom Modul aufgezeichneten Histogramme können vom Anwender ausgelesen werden.

Interner Modulspeicher für Anwenderdaten

Über einen 512 kByte großen, nichtflüchtigen Anwenderspeicher (Flash) können Daten aus der Applikation direkt am Modul gespeichert und auch wieder vom Modul gelesen werden. Die Daten bleiben somit am Modul erhalten, unabhängig von einem Neustart des Moduls bzw. der Steuerung oder davon, dass das Modul z. B. in eine andere Maschine oder Anlage gesteckt wird. Die Datenerhaltung erfolgt wartungsfrei - ohne Batterien.



Information:

Es ist zu beachten, dass der interne Modulspeicher im Funktionsmodell Bus Controller nicht zur Verfügung steht!

Technology Guarding

Mit dem integrierten Technology Guard besteht die Möglichkeit den Lizenzschutz direkt über das Modul anstelle des USB-Dongles zu realisieren. Es können dazu die gleichen Funktionen wie beim USB-Dongle verwendet werden.



Information:

- Um die Funktionen des Technology Guards nutzen zu können, ist eine Automation Runtime Version ab C4.44 erforderlich.
- Es ist zu beachten, dass der Technology Guard im Funktionsmodell Bus Controller nicht zur Verfügung steht!
- Das Modul ist nicht für Controller-Redundanz-Anwendungen geeignet.

Ein-/Ausgänge

Das Modul verfügt über 2 Eingänge für PT1000-Temperaturfühler, 2 digitale Eingänge 24 VDC und 1 digitalen Ausgang 24 VDC.

Blackout-Modus

Durch den integrierten Blackout-Modus ist die Modulfunktion auch bei einem Ausfall des Netzwerks weiter gegeben.

Flatstream-Kommunikation

Der "Flatstream" wurde für X2X und POWERLINK-Netzwerke konzipiert und ermöglicht einen individuell angepassten Datentransfer. Damit kann die Übertragung der Daten effizienter gestaltet werden als bei der zyklischen Standardabfrage.

2 Technische Beschreibung

2.1 Technische Daten

Bestellnummer	X20CMR111
Kurzbeschreibung	
I/O-Modul	Messung von Umgebungsbedingungen: Temperatur im Modul, Luftfeuchtigkeit, Vibration, Schock, Lage, Betriebsstunden, Einschaltzyklen 2 digitale Eingänge, 1 digitaler Ausgang, 2 PT1000-Temperatureingänge, Technology Guarding
Allgemeines	
B&R ID-Code	0xF1AE
Statusanzeigen	I/O-Funktion pro Kanal, Überlast, Speicherzugriff, Tech Guarding, Betriebszustand, Modulstatus
Diagnose	
Modul Run/Error	Ja, per Status-LED und SW-Status
Ausgang	Ja, per Status-LED und SW-Status (Ausgangsfehlerstatus)
Blackout-Modus	
Gültigkeitsbereich	Modul
Funktion	Modulfunktion
Standalone-Modus	Nein
Leistungsaufnahme	
Bus	0,4 W
I/O-intern	0,1 W
Zusätzliche Verlustleistung durch Aktoren (ohmsch) [W]	+0,075
Anwenderspeicher	
Typ	Flashspeicher 512 kByte
Sektoren	8 Sektoren zu je 64 kByte
Datenerhaltung	20 Jahre bei 55°C
garantierte Löscho-/Schreibzyklen	100.000 pro Sektor
Error Correction Coding (ECC)	Nein
Schreibschutz	Nein
Technology Guard	
Datenerhaltung	10 Jahre bei Zimmertemperatur
garantierte Löscho-/Schreibzyklen	Bis zu 10 Millionen
Anwenderspeicher	241 Byte kundenspezifische Daten
Codierung	128-Bit AES, SHA-256, 2048-Bit RSA, 224-Bit ECC
Zulassungen	
CE	Ja
UKCA	Ja
UL	cULus E115267 Industrial Control Equipment
DNV	Temperature: B (0 to 55 °C) Humidity: B (up to 100%) Vibration: B (4 g) EMC: B (bridge and open deck)
CCS	Ja
LR	ENV1
KR	Ja
ABS	Ja
BV	EC33B Temperature: 5 - 55 °C Vibration: 4 g EMC: Bridge and open deck
Digitale Eingänge	
Anzahl	2
Nennspannung	24 VDC
Eingangsspannung	24 VDC -15% / +20%
Eingangsstrom bei 24 VDC	typ. 3,75 mA
Eingangsbeschaltung	Sink
Eingangsfiler	
Hardware	≤100 µs
Software	Default 1 ms, zwischen 0 und 25 ms in 0,2 ms Schritten einstellbar
Anschlussstechnik	1-Leitertechnik
Eingangswiderstand	typ. 6,4 kΩ
Schaltsschwellen	
Low	<5 VDC
High	>15 VDC
Isolationsspannung zwischen Kanal und Bus	500 V _{eff}

Tabelle 2: X20CMR111 - Technische Daten

Technische Beschreibung

Bestellnummer	X20CMR111
Temperatureingänge Widerstandsmessung	
Anzahl	2
Eingang	Widerstandsmessung für 2-Leitertechnik
Digitale Wandlerauflösung	12 Bit
Filterzeit	Konfigurierbar
Wandlungszeit	1 ms für alle Eingänge
Ausgabeformat	INT
Fühler	
PT1000	-40 bis 125°C
Eingangsfiler	Mittelwertbildung mit Clipping-Funktion
Linearisierungsmethode	Intern
Zulässiges Eingangssignal	Kurzzeitig ±15 V
max. Gain-Drift	0,0003 %/°C
max. Offset-Drift	0,06 %/°C
Nichtlinearität	<0,010%
Auflösung Temperaturfühler	
PT1000	1 LSB = 0,1°C
Überwachung Temperaturmessung	
Bereichsunterschreitung	0x8001
Bereichsüberschreitung	0x7FFF
Drahtbruch	0x7FFF
allgemeiner Fehler	0x8000
offene Eingänge	0x7FFF
Temperatur- und Luftfeuchtesensor	
Sensorposition	Modulintern
Abtastrate	1 s
Temperaturmessung	
Messbereich	-25 bis 125°C
Auflösung	0,1 °C/LSB
max. Fehler	±0,3°C
Luftfeuchtemessung	
Messbereich	5 bis 95%
Auflösung	1 %/LSB
max. Fehler	±2% bei 10 bis 80% Luftfeuchtigkeit ±3% bei <10 und >80% Luftfeuchtigkeit
Beschleunigungs- und Winkelsensor	
Sensorposition	Modulintern
Abtastrate	typ. 10 ms
Beschleunigungsmessung	
Messbereich	±16 g
Auflösung	0,488 mg/LSB
Linearität über Temperatur	±1% bei -40 bis 85°C
Offset-Kalibrierungsfehler	±40 mg
Drehgeschwindigkeitsmessung	
Messbereich	±2000 dps ¹⁾
Auflösung	±70 mdps pro LSB
Linearität über Temperatur	±1,5% bei -40 bis 85°C
Offset-Kalibrierungsfehler	±10 dps
Digitale Ausgänge	
Anzahl	1
Ausführung	FET Plus-schaltend
Nennspannung	24 VDC
Schaltspannung	24 VDC -15% / +20%
Ausgangsnennstrom	0,5 A
Anschlusstechnik	1- oder 2-Leitertechnik
Ausgangsbeschaltung	Source
Diagnosestatus	Ausgangsüberwachung mit Verzögerung 10 ms
Leckstrom bei abgeschaltetem Ausgang	120 µA
R _{DS(on)}	300 mΩ
Schaltverzögerung	
0 → 1	<300 µs
1 → 0	<300 µs
Schaltfrequenz	
ohmsche Last	max. 100 Hz
induktive Last	Siehe Abschnitt "Schalten induktiver Lasten"
Isolationsspannung zwischen Kanal und Bus	500 V _{eff}
Schutzmaßnahmen	
Kurzschlussfest	Ja
Überlastfest	Kurzzeitige Überlast
Elektrische Eigenschaften	
Potenzialtrennung	Analoger Kanal zu Bus nicht getrennt Digitaler Kanal zu Bus getrennt Digitaler Kanal zu digitaler Kanal nicht getrennt

Tabelle 2: X20CMR111 - Technische Daten

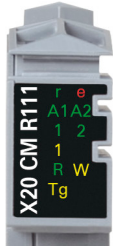
Bestellnummer	X20CMR111
Einsatzbedingungen	
Einbaulage	
waagrecht	Ja
senkrecht	Ja
Aufstellungshöhe über NN (Meeresspiegel)	
0 bis 2000 m	Keine Einschränkung
>2000 m	Reduktion der Umgebungstemperatur um 0,5°C pro 100 m
Schutzart nach EN 60529	IP20
Umgebungsbedingungen	
Temperatur	
Betrieb	
waagrechte Einbaulage	-25 bis 60°C
senkrechte Einbaulage	-25 bis 50°C
Derating	-
Lagerung	-40 bis 85°C
Transport	-40 bis 85°C
Luftfeuchtigkeit	
Betrieb	5 bis 95%, nicht kondensierend
Lagerung	5 bis 95%, nicht kondensierend
Transport	5 bis 95%, nicht kondensierend
Mechanische Eigenschaften	
Anmerkung	Feldklemme 1x X20TB12 gesondert bestellen Busmodul 1x X20BM11 gesondert bestellen
Rastermaß	12,5 ^{+0,2} mm

Tabelle 2: X20CMR111 - Technische Daten

1) dps: degrees per second (°/s)

2.2 Status-LEDs

Für die Beschreibung der verschiedenen Betriebsmodi siehe X20 System Anwenderhandbuch, Abschnitt "Zusätzliche Informationen - Diagnose-LEDs".

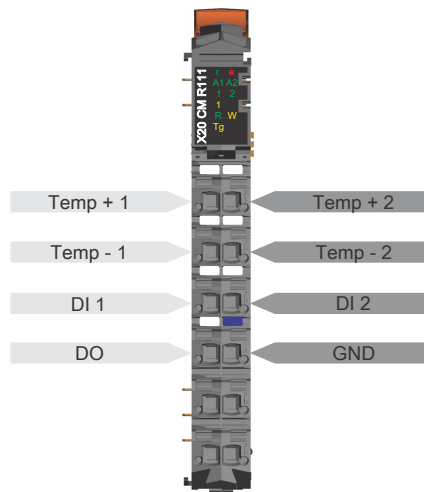
Abbildung	LED	Farbe	Status	Beschreibung
	r	Grün	Aus	Modul nicht versorgt
			Single Flash	Modus RESET
			Double Flash	Blackout-Modus aktiv
			Blinkend	Modus PREOPERATIONAL
			Ein	Modus RUN
	e	Rot	Aus	Modul nicht versorgt oder alles in Ordnung
	e + r	Rot ein / grüner Single Flash		Firmware ist ungültig
	A1 - A2	Grün	Aus	Kanal ist ausgeschaltet
			Blinkend	Drahtbruch oder Über-/Unterlauf des Eingangssignals
			Ein	Der Analog-/Digitalwandler läuft, Wert ist in Ordnung
	1 - 2	Grün		Eingangszustand des korrespondierenden digitalen Eingangs
	1	Gelb		Ausgangszustand des korrespondierenden digitalen Ausgangs
	R	Grün	Aus	Es werden keine Daten aus dem internen Speicher ausgelesen
			Ein	Das Modul liest Daten aus dem internen Speicher aus
	W	Gelb	Aus	Es werden keine Daten in den internen Speicher geschrieben
			Ein	Das Modul schreibt Daten in den internen Speicher
	Tg	Gelb	Aus	Es wird nicht auf den Technology Guard zugegriffen
			Ein	Das Modul greift auf den Technology Guard zu

2.3 Anschlussbelegung

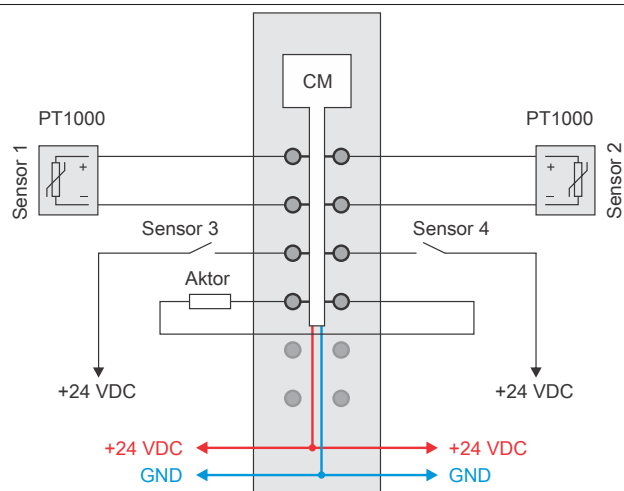


Information:

Die maximale Kabellänge für den externen Temperaturfühler ist 20 m.

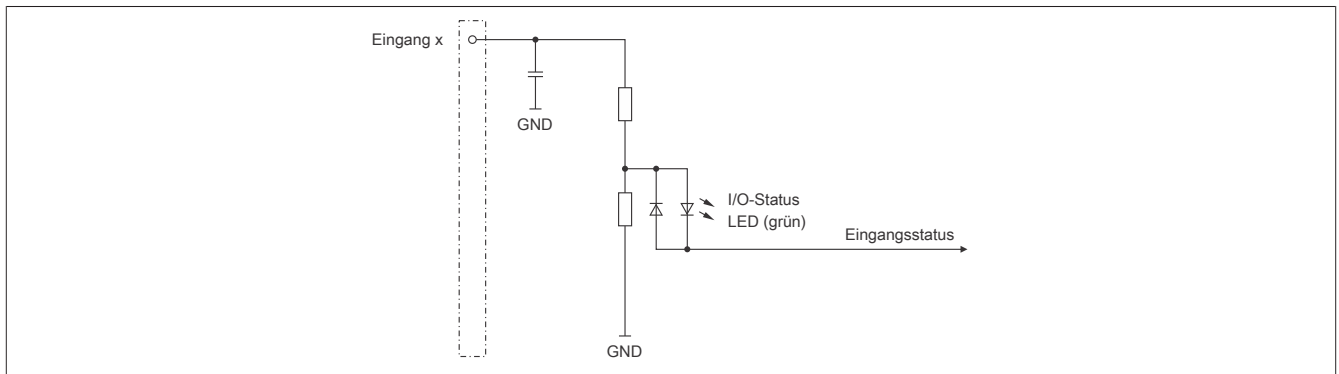


2.4 Anschlussbeispiel

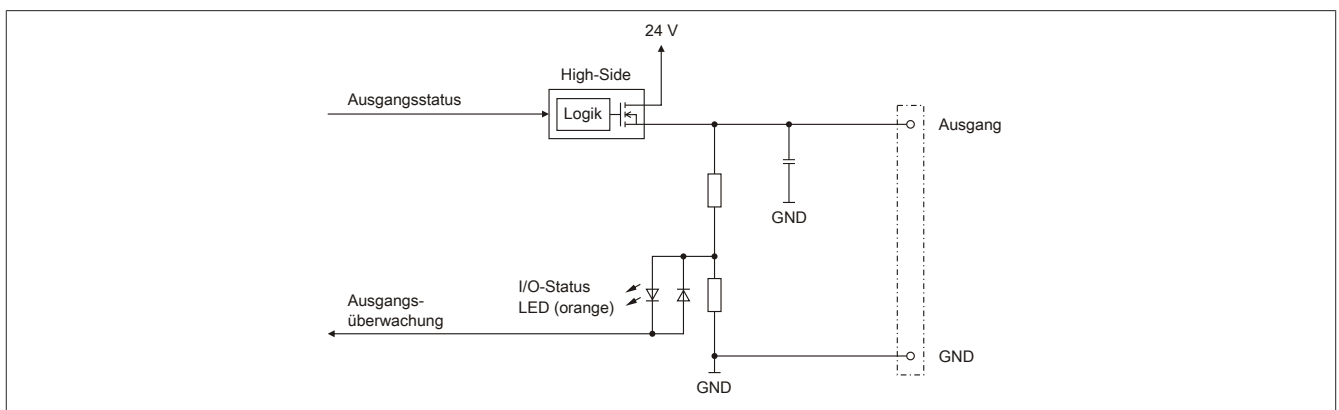


2.5 Ein-/Ausgangsschema

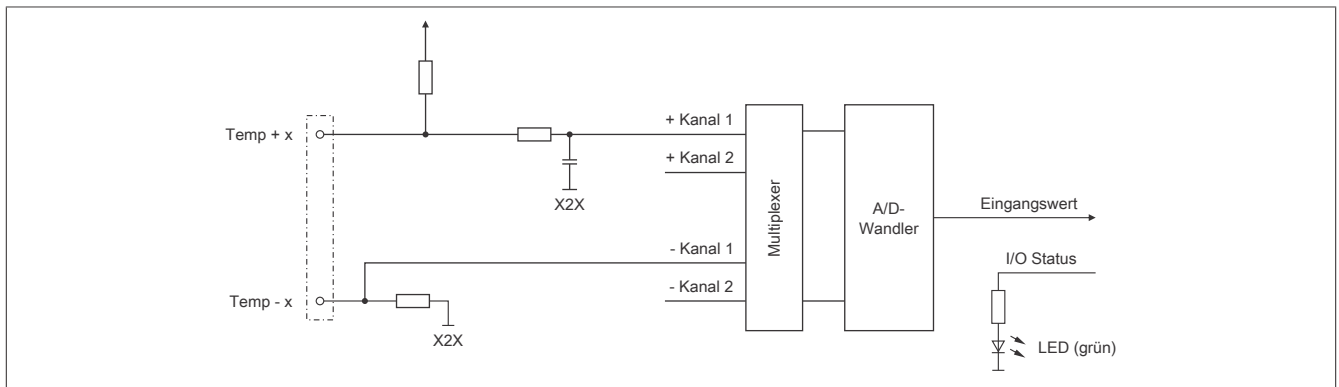
2.5.1 Digitale Eingänge



2.5.2 Digitaler Ausgang

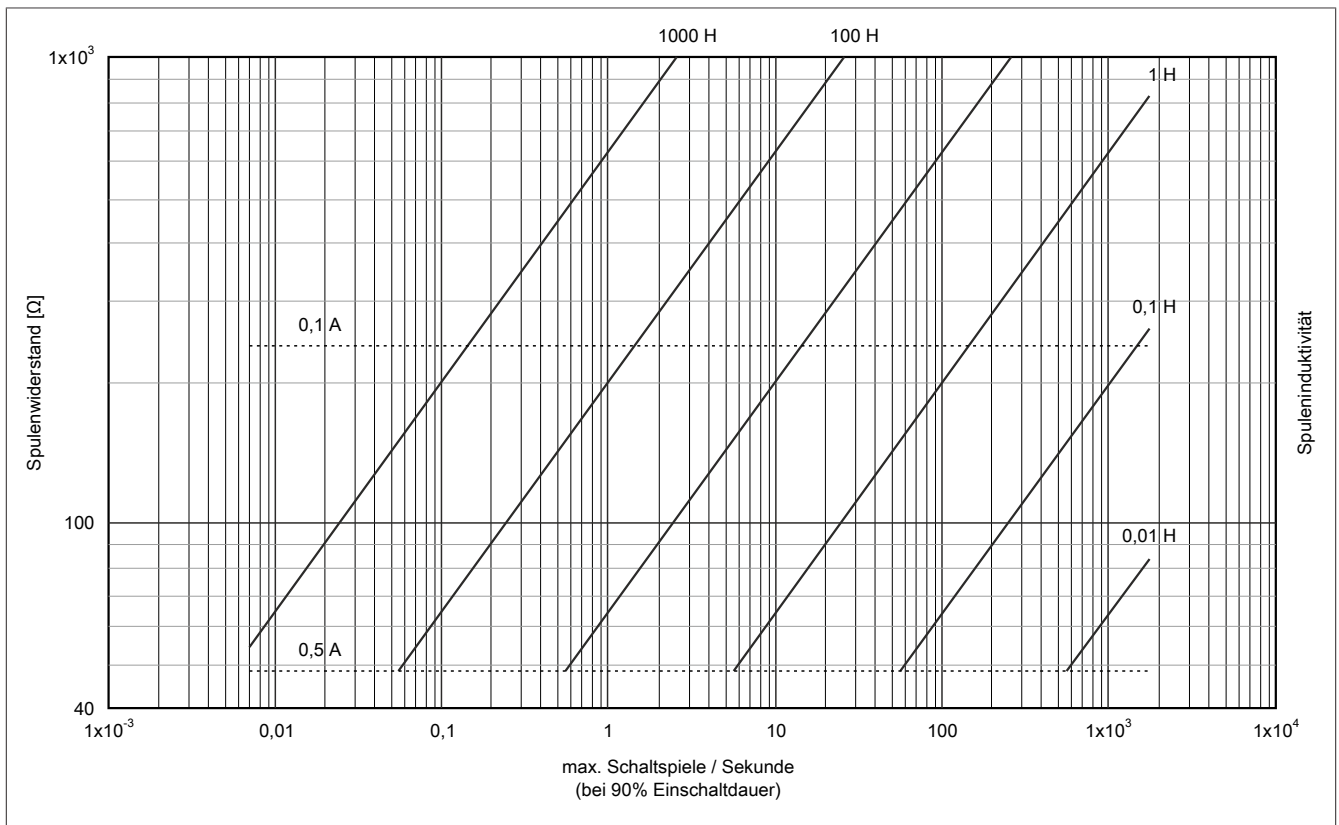


2.5.3 PT1000-Temperatureingänge



2.6 Schalten induktiver Lasten

Umgebungstemperatur: 60°C



Information:

Bei Überschreiten der maximalen Schaltspiele pro Sekunde muss eine externe Freilaufdiode verwendet werden.

Betriebsfälle außerhalb des Diagrammbereichs sind nicht zulässig!

3 Funktionsbeschreibung

3.1 Umgebungsbedingungen messen und auswerten

Das Modul ist mit internen Sensoren bestückt um folgende Bedingungen zu erfassen:

Bedingung	Einheit	Abtastrate
Relative Luftfeuchtigkeit	[%]	1 s
Umgebungstemperatur	[°C]	1 s
Beschleunigung	[g]	typ. 10 ms
Rotation	[dps]	typ. 10 ms

3.1.1 Relative Luftfeuchtigkeit und Umgebungstemperatur

Da sich der Sensor für die relative Luftfeuchtigkeit und Umgebungstemperatur direkt im Modul befindet, hängen die gemessenen Werte von der Eigenerwärmung des Moduls und der Wärmestrahlung durch die benachbarten Module ab.

Der Einfluss der Erwärmung auf die gemessenen Werte kann durch den Anschluss eines externen Temperatursensors an einem der Temperatureingänge des Moduls umgangen werden. Der mit dem externen Temperatursensor gemessene Wert wird als Referenz verwendet. Mit ihm wird die relative Luftfeuchtigkeit an der Position des externen Temperatursensors mit Hilfe der Magnus-Formel berechnet.

$$\text{Sättigungsdampfdruck [Pa]} = 611,2 \cdot e^{\frac{17,62 \cdot \text{Temperatur}}{243,12 + \text{Temperatur}}}$$

$$\text{Absolute Luftfeuchtigkeit [g/m}^3\text{]} = \frac{\text{Sättigungsdampfdruck}}{461,52 \cdot (273,15 + \text{Temperatur})} \cdot 1000$$

$$\text{Luftfeuchtigkeit [g/m}^3\text{]} = \text{Absolute Luftfeuchtigkeit} \cdot \text{Relative Luftfeuchtigkeit}$$

$$\text{Relative Luftfeuchtigkeit [\%]} = \frac{\text{Luftfeuchtigkeit}}{\text{Absolute Luftfeuchtigkeit}} \cdot 100$$

Funktionsbeschreibung

Exemplarisches Beispiel

Im folgenden Beispiel wird die relative Luftfeuchtigkeit am Ort des externen Temperatursensors mit Hilfe der Magnus-Formel berechnet.

- Relative Luftfeuchtigkeit im Modul: 20%
- Umgebungstemperatur im Modul: 40°C
- Externer Temperatursensor: 35°C

Modul

$$\text{Sättigungsdampfdruck}_{\text{Modul}} = 611,2 \cdot e^{\frac{17,62 \cdot 40}{243,12 + 40}} = 7367,5 \text{ Pa}$$

$$\text{Absolute Luftfeuchtigkeit}_{\text{Modul}} = \frac{7367,5}{461,52 \cdot (273,15 + 40)} \cdot 1000 = 50,98 \text{ g/m}^3$$

$$\text{Luftfeuchtigkeit}_{\text{Modul}} = 50,98 \cdot 0,2 = 10,2 \text{ g/m}^3$$

Externer Temperatursensor

$$\text{Sättigungsdampfdruck}_{\text{ExtSensor}} = 611,2 \cdot e^{\frac{17,62 \cdot 35}{243,12 + 35}} = 5612,8 \text{ Pa}$$

$$\text{Absolute Luftfeuchtigkeit}_{\text{ExtSensor}} = \frac{5612,8}{461,52 \cdot (273,15 + 35)} \cdot 1000 = 39,47 \text{ g/m}^3$$

$$\text{Relative Luftfeuchtigkeit}_{\text{ExtSensor}} = \frac{10,2}{39,47} \cdot 100 = 25,84\%$$

Es ergibt sich somit in diesem konkreten Beispiel eine Abweichung der relativen Luftfeuchtigkeit zwischen dem gemessenen Wert im Modul und dem errechneten Wert beim externen Temperatursensor von ca. 6% Luftfeuchte.

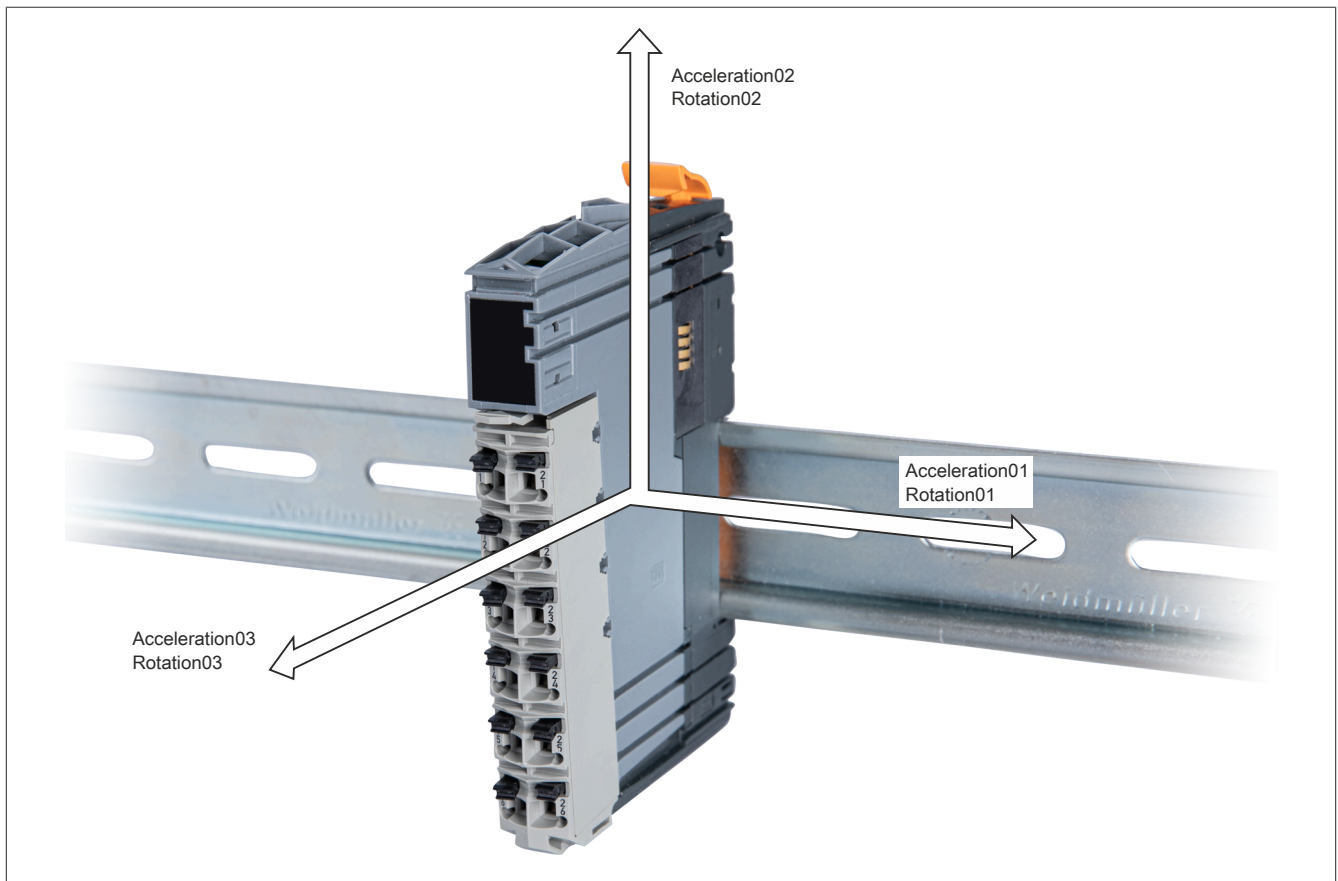
3.1.2 Beschleunigung und Rotation

Die vom internen Sensor gemessene Beschleunigung und Rotation werden als Rohwert geliefert. Die Umrechnung ist in der Applikation durchzuführen.

Gemessene Größe	Umrechnung
Beschleunigung	16 g = 32767 -16 g = -32768
Rotation	2000 dps = 32767 -2000 dps = -32768

Zuordnung der Achsen

Der Sensor ermittelt die Beschleunigungs- und Rotationswerte aller 3 Achsen. Im folgenden Bild ist die Zuordnung der Achsen im Bezug zum auf der Hutschiene montierten Modul ersichtlich.



3.1.3 Zusätzliche Informationen

Die Umgebungsbedingungen werden vom Modul erfasst und ausgewertet. Folgende Werte können ausgelesen werden:

- Kleinster aufgetretener Wert
- Größter aufgetretener Wert



Information:

Die Werte werden im modulinternen FRAM abgespeichert.

Bei Bedarf können die Werte zurückgesetzt werden.

Die Register sind im Abschnitt "[Zusätzliche Informationen](#)" auf Seite 58 beschrieben.

3.1.4 Histogramm für relative Luftfeuchtigkeit

Vom Modul wird ein Histogramm für die relative Luftfeuchtigkeit aufgezeichnet. Dazu wird der Messbereich für die relative Luftfeuchtigkeit in 10 Bereiche aufgeteilt:

Bereich	Relative Luftfeuchtigkeit	Register
1	0 bis <10%	RelHumHist01Entry RelHumHist01Time
2	10 bis <20%	RelHumHist02Entry RelHumHist02Time
3	20 bis <30%	RelHumHist03Entry RelHumHist03Time
4	30 bis <40%	RelHumHist04Entry RelHumHist04Time
5	40 bis <50%	RelHumHist05Entry RelHumHist05Time
6	50 bis <60%	RelHumHist06Entry RelHumHist06Time
7	60 bis <70%	RelHumHist07Entry RelHumHist07Time
8	70 bis <80%	RelHumHist08Entry RelHumHist08Time
9	80 bis <90%	RelHumHist09Entry RelHumHist09Time
10	90 bis 100%	RelHumHist10Entry RelHumHist10Time

Sobald die relative Luftfeuchtigkeit in einen der vordefinierten Bereiche eintritt, beginnt eine Verzögerungszeit von 3 s zu laufen. Nach Ablauf der Verzögerungszeit wird der Eintrittszähler um 1 erhöht und die Verweilzeit beginnt zu laufen. Mit der Verzögerungszeit wird verhindert, dass rund um den Übergangsbereich der Zähler ständig erhöht wird.



Information:

Die Werte werden im modulinternen FRAM abgespeichert.

Bei Bedarf können die Register zurückgesetzt werden.

Die Register sind im Abschnitt "[Relative Luftfeuchtigkeit](#)" auf Seite 60 beschrieben.

3.1.5 Histogramm für Umgebungstemperatur

Vom Modul wird ein Histogramm für die Umgebungstemperatur aufgezeichnet. Dazu wird der Messbereich für die Umgebungstemperatur in 12 Bereiche aufgeteilt:

Bereich	Umgebungstemperatur	Register
1	<-20°C	TempHist01Entry TempHist01Time
2	-20 bis <-10°C	TempHist02Entry TempHist02Time
3	-10 bis <0°C	TempHist03Entry TempHist03Time
4	0 bis <10°C	TempHist04Entry TempHist04Time
5	10 bis <20°C	TempHist05Entry TempHist05Time
6	20 bis <30°C	TempHist06Entry TempHist06Time
7	30 bis <40°C	TempHist07Entry TempHist07Time
8	40 bis <50°C	TempHist08Entry TempHist08Time
9	50 bis <60°C	TempHist09Entry TempHist09Time
10	60 bis <70°C	TempHist10Entry TempHist10Time
11	70 bis <80°C	TempHist11Entry TempHist11Time
12	≥80°C	TempHist12Entry TempHist12Time

Sobald die Umgebungstemperatur in einen der vordefinierten Bereiche eintritt, beginnt eine Verzögerungszeit von 3 s zu laufen. Nach Ablauf der Verzögerungszeit wird der Eintrittszähler um 1 erhöht und die Verweilzeit beginnt zu laufen. Mit der Verzögerungszeit wird verhindert, dass rund um den Übergangsbereich der Zähler ständig erhöht wird.



Information:

Die Werte werden im modulinternen FRAM abgespeichert.

Bei Bedarf können die Register zurückgesetzt werden.

Die Register sind im Abschnitt "[Umgebungstemperatur](#)" auf Seite 60 beschrieben.

3.1.6 Histogramm für Beschleunigung

Vom Modul wird ein Histogramm für die Beschleunigung aufgezeichnet. Dazu wird der Messbereich für die Beschleunigung in 8 Bereiche aufgeteilt:

Bereich	Beschleunigung	Register
1	-16 bis <-12 g	AccHist0N01Entry AccHist0N01Time
2	-12 bis <-8 g	AccHist0N02Entry AccHist0N02Time
3	-8 bis <-4 g	AccHist0N03Entry AccHist0N03Time
4	-4 bis <0 g	AccHist0N04Entry AccHist0N04Time
5	0 bis <4 g	AccHist0N05Entry AccHist0N05Time
6	4 bis <8 g	AccHist0N06Entry AccHist0N06Time
7	8 bis <12 g	AccHist0N07Entry AccHist0N07Time
8	12 bis 16 g	AccHist0N08Entry AccHist0N08Time

Legende: N = 1 bis 3

Sobald die Beschleunigung in einen der vordefinierten Bereiche eintritt, beginnt eine Verzögerungszeit von 3 s zu laufen. Nach Ablauf der Verzögerungszeit wird der Eintrittszähler um 1 erhöht und die Verweilzeit beginnt zu laufen. Mit der Verzögerungszeit wird verhindert, dass rund um den Übergangsbereich der Zähler ständig erhöht wird.



Information:

Die Werte werden im modulinternen FRAM abgespeichert.

Bei Bedarf können die Register zurückgesetzt werden.

Die Register sind im Abschnitt "[Beschleunigung](#)" auf Seite 60 beschrieben.

3.1.7 Histogramm für Rotation

Vom Modul wird ein Histogramm für die Rotation aufgezeichnet. Dazu wird der Messbereich für die Rotation in 8 Bereiche aufgeteilt:

Bereich	Rotation	Register
1	-2000 bis <-1500 dps	RotationHist0N01Entry RotationHist0N01Time
2	-1500 bis <-1000 dps	RotationHist0N02Entry RotationHist0N02Time
3	-1000 bis <-500 dps	RotationHist0N03Entry RotationHist0N03Time
4	-500 bis <0 dps	RotationHist0N04Entry RotationHist0N04Time
5	0 bis <500 dps	RotationHist0N05Entry RotationHist0N05Time
6	500 bis <1000 dps	RotationHist0N06Entry RotationHist0N06Time
7	1000 bis <1500 dps	RotationHist0N07Entry RotationHist0N07Time
8	1500 bis 2000 dps	RotationHist0N08Entry RotationHist0N08Time

Legende: N = 1 bis 3

Sobald die Rotation in einen der vordefinierten Bereiche eintritt, beginnt eine Verzögerungszeit von 3 s zu laufen. Nach Ablauf der Verzögerungszeit wird der Eintrittszähler um 1 erhöht und die Verweilzeit beginnt zu laufen. Mit der Verzögerungszeit wird verhindert, dass rund um den Übergangsbereich der Zähler ständig erhöht wird.



Information:

Die Werte werden im modulinternen FRAM abgespeichert.

Bei Bedarf können die Register zurückgesetzt werden.

Die Register sind im Abschnitt "[Rotation](#)" auf Seite 61 beschrieben.

3.2 Betriebsdaten aufzeichnen

Vom Modul werden folgende Betriebsdaten erfasst:

- Laufzeit mit aktiver Verbindung zum Netzwerkmaster
- Laufzeit ohne aktiver Verbindung zum Netzwerkmaster (Blackout-Modus)
- Gesamte Laufzeit
- Anzahl der Einschaltzyklen



Information:

Die Werte werden im modulinternen FRAM abgespeichert.

Bei Bedarf können die Betriebsdaten zurückgesetzt werden.

Die Register sind im Abschnitt "**Betriebsdaten**" auf Seite 58 beschrieben.

3.3 Interner Modulspeicher für Anwenderdaten

3.3.1 Allgemeines

Das Modul stellt 512 kByte nichtflüchtigen internen Flash-Speicher zur Verfügung, der über die Applikation verwendet werden kann. Es können Daten direkt am Modul gespeichert und auch wieder vom Modul gelesen werden. Dadurch ist z. B. die Ablage von Rezept- und Betriebsinformationen zur Maschine am Modul möglich.

3.3.2 Bedienung

Die Speicherschnittstelle des Moduls basiert auf der Flatstream-Kommunikation. Die Bedienung erfolgt über die Bibliothek "AsFltGen".



Information:

Weitere Informationen zur Bibliothek "AsFltGen" können der Automation Help entnommen werden.



Information:

Folgende Punkte sind zu beachten:

- Pro Lese- oder Schreibbefehl können maximal 256 Byte gelesen oder geschrieben werden. Die 256 Byte repräsentieren dabei eine Page. Wenn mehr als 256 Byte gelesen oder geschrieben werden müssen, muss eine sequentielle Abfolge der Befehle und ein Speichermanagement in der Applikation realisiert werden.
- Der Löschbefehl basiert auf Sektoren. Ein Sektor ist 64 kByte groß. Das entspricht 256 Pages. Der gesamte Sektor, in dem sich die angegebene Adresse befindet, wird gelöscht. Der Flash-Speicher ist in insgesamt 8 Sektoren gegliedert (8 x 64 kByte = 512 kByte).
- Um Daten zu überschreiben, muss der entsprechende Sektor zuerst gelöscht werden. Erst dann können die neuen Daten gespeichert werden.
- Die Speicheraufteilung kann beliebig erfolgen. Bei regelmäßigem Überschreiben von Daten sollte für diese ein eigener Sektor verwendet werden.

3.3.3 Befehle

3.3.3.1 Protokoll

Jedem Befehl ist ein Header vorangestellt. Abhängig vom Befehl folgen dem Header die Daten.




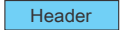
Funktionsbeschreibung

3.3.3.2 Header

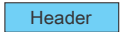

Jede Anforderung oder Antwort startet mit einem 16 Byte langen Header. Die folgenden Elemente müssen im Header definiert werden:

Element	Datentyp	Aktivität	Beschreibung
Code	USINT	Anforderung	Definiert den Befehl: "r" ... Daten lesen (ASCII Code 0x72) "w" ... Daten schreiben (ASCII Code 0x77) "e" ... Daten löschen (ASCII Code 0x65)
		Antwort	Der in der Anforderung enthaltene Befehlscode wird zurückgeschickt.
Fortlaufende Nummer	USINT	Anforderung	Frei verwendbar. Die fortlaufende Nummer ist z. B. dann wichtig, wenn mehr als 256 Byte gelesen oder geschrieben werden müssen. In diesem Fall muss der Anwender eine sequentielle Abfolge der Befehle und ein Speichermanagement in der Applikation durchführen.
		Antwort	Die in der Anforderung enthaltene Nummer wird zurückgeschickt.
Status	UINT	Anforderung	Nicht verwendet: Das Byte wird nicht ausgewertet.
		Antwort	Statusrückmeldung: 0x0000 ... Der Befehl wurde erfolgreich ausgeführt 0x8001 ... Ungültig: Allgemeiner Fehler 0x8002 ... Ungültige Adresse 0x8003 ... Ungültige Größe 0x8004 ... Flash busy 0x8006 ... Flash Timeout
Adresse	UDINT	Anforderung	Startadresse ab der die Daten gelesen oder geschrieben werden.
		Antwort	Die in der Anforderung enthaltene Startadresse wird zurückgeschickt.
Datengröße	UDINT	Anforderung	Datengröße der zu lesenden oder zu schreibenden Daten.
		Antwort	Die in der Anforderung enthaltene Datengröße wird zurückgeschickt.
Reserve	UDINT	Reserviert	


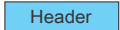
3.3.3.3 Daten schreiben

Aktion	Beschreibung
Anforderung	Um Daten auf dem Modul speichern zu können, muss der Header für die Kommunikation vorbereitet werden. Direkt an den Header werden die Daten angehängt. Header und Daten müssen dem Funktionsblock "fltWrite" als Sendepuffer übergeben werden. 
Antwort	Die Antwort des Moduls - der zurückgeschickte Header - wird mittels des Funktionsbausteins "fltRead" im Empfangspuffer abgelegt und kann von der Applikation ausgewertet werden. 

3.3.3.4 Daten lesen

Aktion	Beschreibung
Anforderung	Um Daten vom Modul lesen zu können, muss der Header für die Kommunikation vorbereitet werden. Der Header muss dem Funktionsbaustein "fltWrite" als Sendepuffer übergeben werden. 
Antwort	Die Antwort des Moduls - der zurückgeschickte Header und die Daten - werden mittels des Funktionsbausteins "fltRead" im Empfangspuffer abgelegt und können von der Applikation ausgewertet werden. 

3.3.3.5 Sektor löschen

Aktion	Beschreibung
Anforderung	Um einen Bereich im Flash des Moduls löschen zu können, muss der Header für die Kommunikation vorbereitet werden. Der gesamte 64 kByte große Sektor, in dem sich die angegebene Adresse befindet, wird gelöscht. Der Header muss dem Funktionsbaustein "fltWrite" als Sendepuffer übergeben werden. 
Antwort	Die Antwort des Moduls - der zurückgeschickte Header - wird mittels des Funktionsbausteins "fltRead" im Empfangspuffer abgelegt und kann von der Applikation ausgewertet werden. 

3.4 Technology Guarding

Das Modul ist mit einem Technology Guard ausgestattet. Durch den bereits am Modul integrierten Technology Guard ergeben sich folgende Vorteile:

- Für den Technology Guard ist keine USB-Schnittstelle notwendig
- Einsatz des Moduls, wenn keine USB-Schnittstelle mehr frei ist
- Einsatz des Moduls, wenn USB-Schnittstellen aus Sicherheitsgründen generell verboten sind

Das Technology Guarding über das Modul basiert auf dem gleichen Mechanismus wie beim USB-Dongle der an der Steuerung gesteckt wird. Die eigentliche Lizenzprüfung erfolgt im Automation Runtime auf dem Zielsystem. Das Automation Runtime erkennt eine Lizenzverletzung, wenn der Lizenzbedarf durch die auf dem Technology Guard vorhandenen Lizenzen nicht abgedeckt ist.

Der Technology Guard bietet folgende Möglichkeiten:

- 2 manipulationssichere Betriebsstundenzähler
- Speicherung mehrerer B&R Lizenzen
- Kundenspezifische Lizenzspeicherung möglich
- Kundenspezifische Datenspeicherung möglich



Information:

Die erweiterte anwenderspezifische Bedienung des Technology Guards erfolgt über die Bibliothek "AsGuard". Weitere Informationen zur Bibliothek "AsGuard" können der Automation Help entnommen werden.



Information:

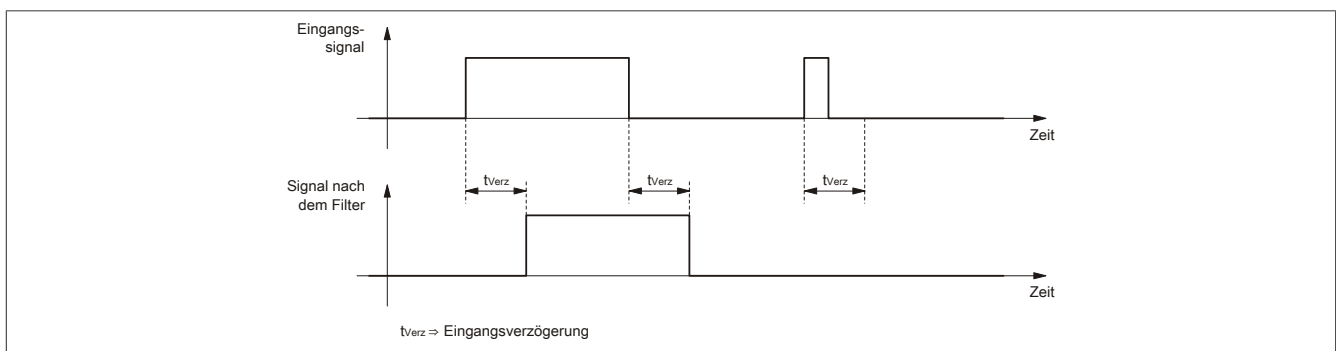
Um die Funktionen des Technology Guards nutzen zu können, ist eine Automation Runtime Version ab C4.44 erforderlich.

3.5 Digitale Eingänge

Das Modul ist mit 2 digitalen Eingängen ausgestattet. Dadurch kann z. B. über einen Türkontakt festgestellt werden, ob die Schaltschranktür gerade offen oder geschlossen ist.

3.5.1 Eingangsfilter

Für jeden Eingang ist ein Eingangsfilter vorhanden. Störimpulse, die kürzer als die Eingangsverzögerung sind, werden durch den Eingangsfilter unterdrückt.



Die Eingangsverzögerung kann in Schritten von 100 µs eingestellt werden. Da die Abtastung der Eingangssignale jedoch im Raster von 200 µs erfolgt, ist es sinnvoll Werte in 2er-Schritten einzugeben.

Werte	Filter
0	Kein Softwarefilter
2	0,2 ms
...	...
250	25 ms - höhere Werte werden auf diesen Wert begrenzt

**Information:**

Das Register ist im Abschnitt "**Digitale Eingangsfilter**" auf Seite 52 beschrieben.

3.5.2 Eingangszustand erfassen

Ungefiltert

Der Eingangszustand wird mit einem festen Versatz bezogen auf den Netzwerkzyklus erfasst und im selben Zyklus übertragen.

Gefiltert

Der gefilterte Zustand wird mit einem festen Versatz bezogen auf den Netzwerkzyklus erfasst und im selben Zyklus übertragen. Das Filtern erfolgt asynchron zum Netzwerk in einem Raster von 200 µs mit einem Netzwerk bedingten Jitter von bis zu 50 µs.

3.6 Analoge Eingänge

Mit Hilfe von 2 PT1000-Temperatureingängen kann die Temperatur an kritischen Stellen im Schaltschrank ermittelt werden.

3.6.1 Filterstufe

Mit der Filterstufe werden die Anzahl der Messungen für die Mittelwertbildung festgelegt. Die externen Temperaturfühler werden alle 1 ms gemessen. Das heißt, dass bei default 100 Messungen für die Mittelwertbildung alle 100 ms ein neuer Wert vorliegt ($100 \times 1 \text{ ms} = 100 \text{ ms}$).

**Information:**

Das Register ist im Abschnitt "**Filterstufe und Eingangsrampenbegrenzung**" auf Seite 53 beschrieben.

3.6.2 Eingangsrampenbegrenzung

Eine Eingangsrampenbegrenzung kann nur in Verbindung mit einer Filterung erfolgen. Wobei die Eingangsrampenbegrenzung vor der Filterung durchgeführt wird.

Es wird die Differenz der Eingangswertänderung auf Überschreitung der angegebenen Grenze überprüft. Im Falle einer Überschreitung ist der nachgeführte Eingangswert gleich dem alten Wert \pm dem Grenzwert. Der Eingangswert wird entsprechend dem Raster der Abtastung alle 1 ms nachgeführt.

Die Eingangsrampenbegrenzung eignet sich zur Unterdrückung von Störpulsen (Spikes).

**Information:**

Das Register ist im Abschnitt "**Filterstufe und Eingangsrampenbegrenzung**" auf Seite 53 beschrieben.

3.6.3 Analogwert begrenzen

Im Fehlerzustand wird der Analogwert per Standardeinstellung auf die unten angeführten Werte fixiert.

Fehlerzustand	Digitaler Wert bei Fehler
Drahtbruch	32767 (0x7FFF)
Oberer Grenzwert überschritten	32767 (0x7FFF)
Unterer Grenzwert unterschritten	-32767 (0x8001)
Ungültiger Wert	-32768 (0x8000) ¹⁾

1) Nach Abschalten des Kanals während des Betriebs oder bei deaktiviertem Kanal

3.6.4 Ausgabewert

Damit dem Anwender immer ein definierter Ausgabewert zur Verfügung steht, ist folgendes zu beachten:

- Bis zur ersten Wandlung wird 0x8000 ausgegeben.
- Wenn der Eingang nicht eingeschaltet ist, wird 0x8000 ausgegeben.

3.7 Blackout-Modus

Der Blackout-Modus ermöglicht es Anwendern, nach dem Ausfall von Teilen eines B&R Systems die Abarbeitung der Applikation in untergeordneten Teilsystemen aufrecht zu erhalten. Das B&R System bietet damit - unabhängig vom Einsatz von Redundanztechnologien - die Möglichkeit, auf systemkritische Situationen anwendungsspezifisch zu reagieren.

Der Einsatz Blackout-fähiger Module ist bei folgenden Anforderungen empfehlenswert:

- Exit-Routinen bei Systemausfall, z. B. um das Öffnen einer Presse bei Systemausfall zu ermöglichen.
- Halten bzw. kontrolliertes Setzen eines Ausgangs bei Systemausfall, z. B. automatisches Schließen von Zuflussventilen.
- Verzögerungssequenzen bei Systemausfall, z. B. Reduzieren von Motorgeschwindigkeiten vor dem Senden eines Stoppbefehls.

Bei entsprechender Parametrierung der Blackout-fähigen Module wird der Blackout-Modus ausgeführt, wenn die Netzwerkverbindung zum übergeordneten Controller bzw. zur Steuerung unterbrochen wird.

Sobald die Störung des Netzwerkes behoben wurde, wird der Blackout-Modus selbstständig von den Modulen beendet und stoßfrei mit dem Netzwerk synchronisiert.

Voraussetzungen zum Betrieb

Um den Blackout-Modus benützen zu können, müssen folgende Voraussetzungen erfüllt sein:

- Das verwendete Modul muss den Blackout-Modus unterstützen.
- Im Automation Studio muss der Parameter "Blackout mode" aktiviert sein.

3.7.1 Anwendungsbereiche

Durch den Einsatz von Blackout-fähigen Modulen kann ein Teil der Steuerung auch funktionsfähig bleiben, wenn die Netzwerk- oder X2X Link Verbindung zwischen den Modulen gestört wird.

3.7.1.1 Verlust der POWERLINK-Verbindung

Ausgangssituation

In einer Anwendung sind mehrere Stationen mittels Netzkabel mit der Steuerung verbunden. Durch einen Störfall wird die Datenübertragung zwischen der Steuerung und den Stationen unterbrochen.

Auswirkung

Nicht Blackout-fähige Module werden zurückgesetzt und im Standardverhalten betrieben.

Blackout-fähige Module zeigen folgendes Verhalten:

- Die programmierte Funktion wird weiter ausgeführt.
- Untergeordnete Netzwerke funktionieren weiterhin.
- Daten von der Steuerung werden mit "0" initialisiert.
- Das Modul fügt sich nach dem Beheben der Störung wieder stoßfrei in das übergeordnete Netzwerk ein.



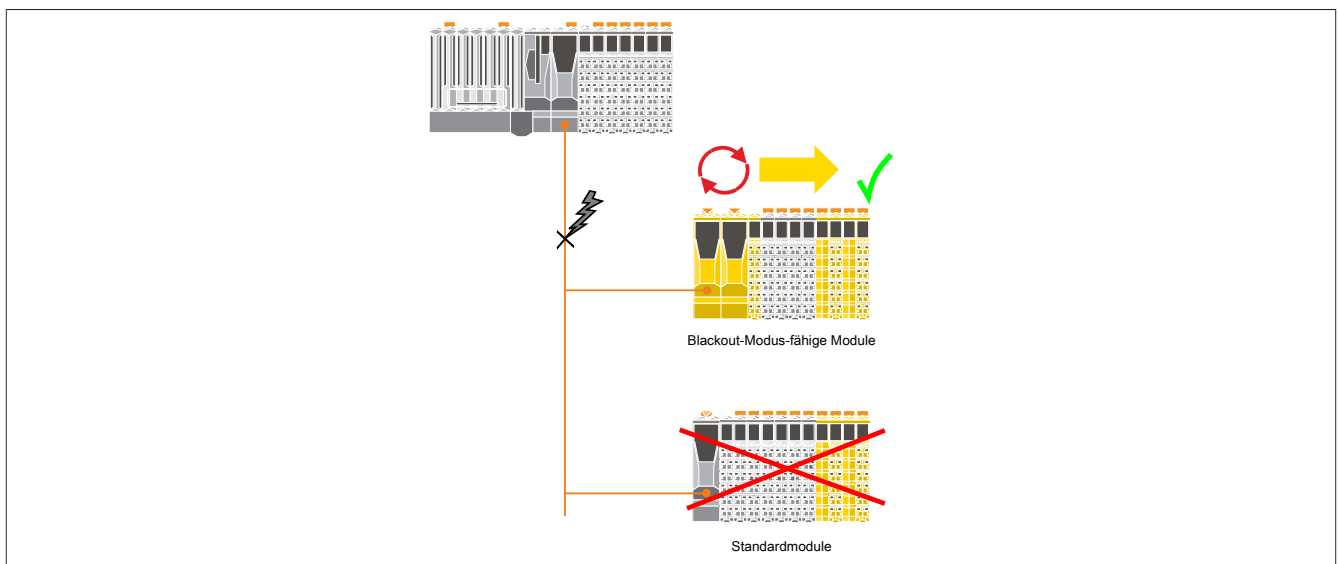
Warnung!

Der Blackout-Modus führt zu einer Initialisierung der Daten von der Steuerung mit "0". Wird der Blackout-Modus in Kombination mit "Ausgangsinvertierung" verwendet, kann dies zu einem ungewollten Setzen von Ausgängen führen.



Mise en garde !

Le mode « Blackout » entraîne l'initialisation des données du contrôleur avec « 0 ». Si le mode « Blackout » est utilisé en combinaison avec « l'inversion de sortie », il peut en résulter un réglage involontaire des sorties.



3.7.1.2 Verlust der X2X Link Verbindung

Ausgangssituation

In einer Anwendung sind Module mittels X2X Link Kabel mit dem Netzwerk verbunden. Durch einen Defekt des X2X Link Kabels wird die Datenübertragung zwischen der Steuerung und den Modulen unterbrochen.

Auswirkung

Nicht Blackout-fähige Module werden zurückgesetzt und im Standardverhalten betrieben.

Blackout-fähige Module zeigen folgendes Verhalten:

- Die programmierte Funktion wird weiter ausgeführt.
- Untergeordnete Netzwerke funktionieren weiterhin.
- Daten von der Steuerung werden mit "0" initialisiert.
- Das Modul fügt sich nach dem Beheben der Störung wieder stoßfrei in das übergeordnete Netzwerk ein.



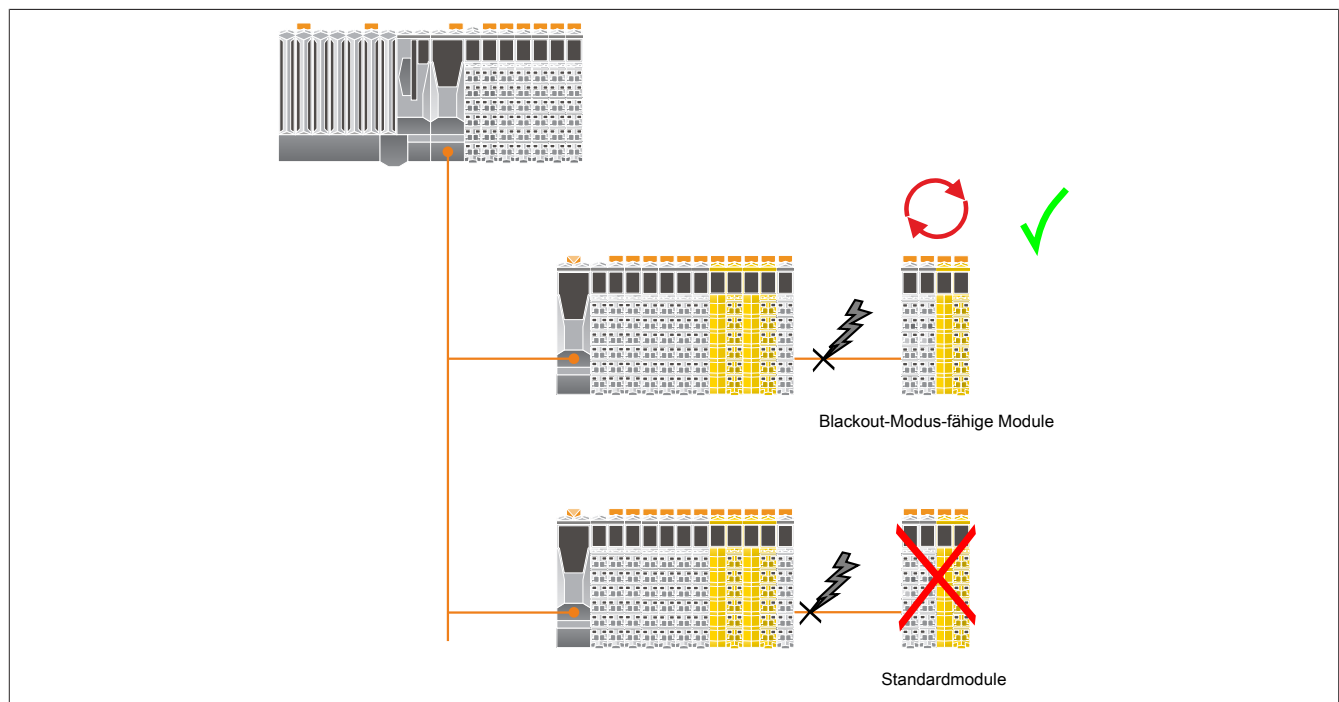
Warnung!

Der Blackout-Modus führt zu einer Initialisierung der Daten von der Steuerung mit "0". Wird der Blackout-Modus in Kombination mit "Ausgangsinvertierung" verwendet, kann dies zu einem ungewollten Setzen von Ausgängen führen.



Mise en garde !

Le mode « Blackout » entraîne l'initialisation des données du contrôleur avec « 0 ». Si le mode « Blackout » est utilisé en combinaison avec « l'inversion de sortie », il peut en résulter un réglage involontaire des sorties.



3.7.2 Programmierung des Blackout-Modus

Der Blackout-Modus kann von den Blackout-fähigen Modulen selbst nicht erkannt werden. Falls es in einer Applikation notwendig ist, ein spezielles Blackout-Verhalten zu programmieren, muss deshalb ein indirektes Verfahren gewählt werden.

Eine Möglichkeit ist, in der dem Blackout-fähigen Modul übergeordneten Steuerung einen Zähler zu implementieren und diesen zyklisch abzufragen. Der Blackout-Modus würde sich in diesem Fall durch einen sich nicht mehr ändernden Zählerwert oder durch einen Nullwert im Zähler bemerkbar machen.

Die Blackout-fähigen Module selbst lassen sich in 2 Kategorien einteilen:

- **Programmierbare Module**

Die Blackout-Funktion wird auf der Basis bestehender Funktionsbausteine programmiert, das heißt, es werden die bestehenden Technologien der Applikationsprogrammierung oder der reACTION Technology verwendet.

Die Blackout-Funktion wird dabei weitgehend unabhängig von anderen Systemkomponenten abgearbeitet.

- **Standardfunktionsmodule**

Diese Module sind nicht programmierbar, sondern behalten im Falle des Blackout-Modus ihr Standardverhalten bei.

3.8 Die Flatstream-Kommunikation

3.8.1 Einleitung

Für einige Module stellt B&R ein zusätzliches Kommunikationsverfahren bereit. Der "Flatstream" wurde für X2X und POWERLINK-Netzwerke konzipiert und ermöglicht einen individuell angepassten Datentransfer. Obwohl das Verfahren nicht unmittelbar echtzeitfähig ist, kann die Übertragung effizienter gestaltet werden als bei der zyklischen Standardabfrage.

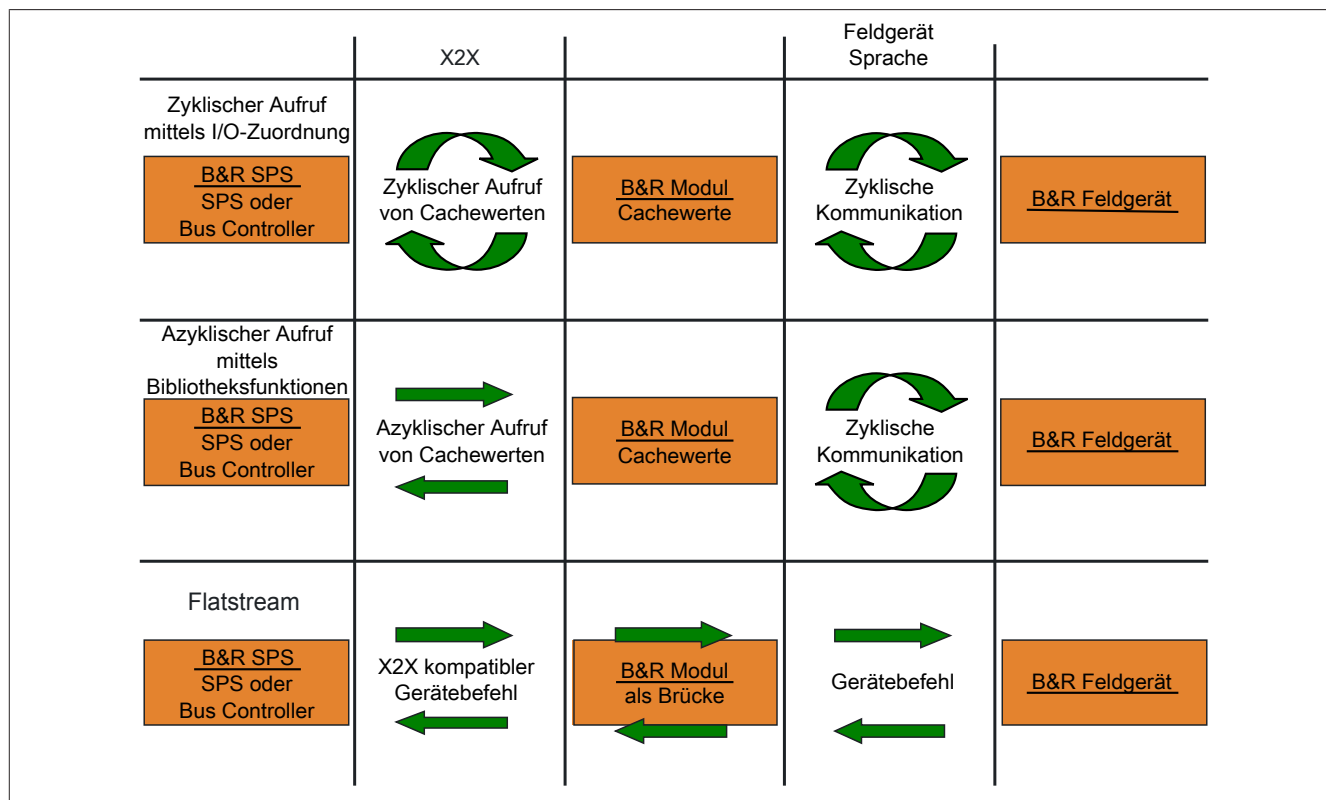


Abbildung 1: 3 Arten der Kommunikation

Durch den Flatstream wird die zyklische bzw. azyklische Abfrage ergänzt. Bei der Flatstream-Kommunikation fungiert das Modul als Bridge. Die Anfragen der Steuerung werden über das Modul direkt zum Feldgerät geleitet.

3.8.2 Nachricht, Segment, Sequenz, MTU

Die physikalischen Eigenschaften des Bussystems begrenzen die Datenmenge, die während eines Buszyklus übermittelt werden kann. Bei der Flatstream-Kommunikation werden alle Nachrichten als fortlaufender Datenstrom (engl. stream) betrachtet. Lange Datenströme müssen in mehrere Teile zerlegt und nacheinander versendet werden. Um zu verstehen, wie der Empfänger die ursprüngliche Information wieder zusammensetzt, werden die Begriffe Nachricht, Segment, Sequenz und MTU unterschieden.

Nachricht

Eine Nachricht ist eine Mitteilung, die zwischen 2 Kommunikationspartnern ausgetauscht werden soll. Die Länge einer solchen Mitteilung wird durch das Flatstream-Verfahren nicht begrenzt. Es müssen allerdings modulspezifische Beschränkungen beachtet werden.

Segment (logische Gliederung einer Nachricht)

Ein Segment ist endlich groß und kann als Abschnitt der Nachricht verstanden werden. Die Anzahl der Segmente pro Nachricht ist beliebig. Damit der Empfänger die übertragenen Segmente wieder korrekt zusammensetzen kann, geht jedem Segment ein Byte mit Zusatzinformationen voraus. Das sogenannte Control-byte enthält z. B. Informationen über die Länge eines Segments und ob das kommende Segment die Mitteilung vervollständigt. Auf diesem Weg wird der Empfänger in die Lage versetzt, den ankommenden Datenstrom korrekt zu interpretieren.

Sequenz (physikalisch notwendige Gliederung eines Segments)

Die maximale Größe einer Sequenz entspricht der Anzahl der aktivierten Rx- bzw. Tx-Bytes (später: "MTU"). Die sendende Station teilt das Sendearray in zulässige Sequenzen, die nacheinander in die MTU geschrieben, zum Empfänger übertragen und dort wieder aneinandergereiht werden. Der Empfänger legt die ankommenden Sequenzen in einem Empfangsarray ab und erhält somit ein Abbild des Datenstroms.

Bei der Flatstream-Kommunikation werden die abgesetzten Sequenzen gezählt. Erfolgreich übertragene Sequenzen müssen vom Empfänger bestätigt werden, um die Übertragung abzusichern.

MTU (Maximum Transmission Unit) - Physikalischer Transport

Die MTU des Flatstreams beschreibt die aktivierten USINT-Register für den Flatstream. Die Register können mindestens eine Sequenz aufnehmen und zum Empfänger übertragen. Für beide Kommunikationsrichtungen wird eine separate MTU vereinbart. Die OutputMTU definiert die Anzahl der Flatstream-Tx-Bytes und die InputMTU beschreibt die Anzahl der Flatstream-Rx-Bytes. Die MTUs werden zyklisch über den X2X Link transportiert, sodass die Auslastung mit jedem zusätzlich aktivierten USINT-Register steigt.

Eigenschaften

Flatstream-Nachrichten werden nicht zyklisch und nicht unmittelbar in Echtzeit übertragen. Zur Übertragung einer bestimmten Mitteilung werden individuell viele Buszyklen benötigt. Die Rx-/Tx-Register werden zwar zyklisch zwischen Sender und Empfänger ausgetauscht, aber erst weiterverarbeitet, wenn die Übernahme durch die Register "InputSequence" bzw. "OutputSequence" explizit angewiesen wird.

Verhalten im Fehlerfall (Kurzfassung)

Das Protokoll von X2X bzw. POWERLINK Netzwerken sieht vor, dass bei einer Störung die letzten gültigen Werte erhalten bleiben. Bei der herkömmlichen Kommunikation (zyklische/azyklische Abfrage) kann ein solcher Fehler in der Regel ignoriert werden.

Damit auch via Flatstream problemlos kommuniziert werden kann, müssen alle abgesetzten Sequenzen vom Empfänger bestätigt werden. Ohne die Nutzung des Forwards verzögert sich die weitere Kommunikation um die Dauer der Störung.

Falls der Forward genutzt wird, erhält die Empfängerstation einen doppelt inkrementierten Sendezähler. Der Empfänger stoppt, das heißt, er schickt keine Bestätigungen mehr zurück. Anhand des SequenceAck erkennt die Sendestation, dass die Übertragung fehlerhaft war und alle betroffenen Sequenzen wiederholt werden müssen.

3.8.3 Prinzip des Flatstreams

Voraussetzung

Bevor der Flatstream genutzt werden kann, muss die jeweilige Kommunikationsrichtung synchronisiert sein, das heißt, beide Kommunikationspartner fragen zyklisch den SequenceCounter der Gegenstelle ab. Damit prüfen sie, ob neue Daten vorliegen, die übernommen werden müssen.

Kommunikation

Wenn ein Kommunikationspartner eine Nachricht an seine Gegenstelle senden will, sollte er zunächst ein Sendearray anlegen, das den Konventionen des Flatstreams entspricht. Auf diese Weise kann der Flatstream sehr effizient gestaltet werden, ohne wichtige Ressourcen zu blockieren.

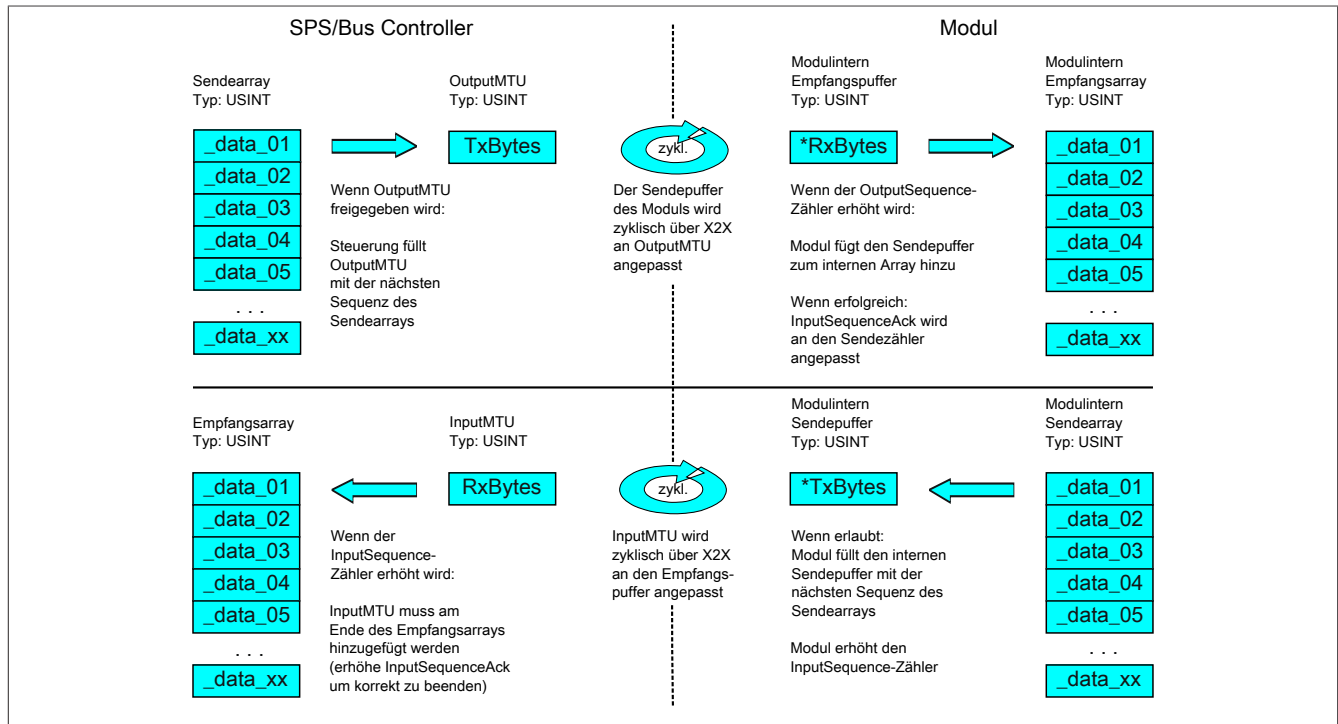


Abbildung 2: Kommunikation per Flatstream

Vorgehensweise

Als erstes wird die Nachricht in zulässige Segmente mit max. 63 Bytes aufgeteilt und die entsprechenden Controlbytes gebildet. Die Daten werden zu einem Datenstrom zusammengefügt, das heißt, je ein Controlbyte und das dazugehörige Segment im Wechsel. Dieser Datenstrom kann in das Sendearray geschrieben werden. Jedes Arrayelement ist dabei max. so groß, wie die freigegebene MTU, sodass ein Element einer Sequenz entspricht.

Wenn das Array vollständig angelegt ist, prüft der Sender, ob die MTU neu befüllt werden darf. Danach kopiert er das erste Element des Arrays bzw. die erste Sequenz auf die Tx-Byte-Register. Die MTU wird zyklisch über den X2X Link zur Empfängerstation transportiert und auf den korrespondierenden Rx-Byte-Registern abgelegt. Als Signal, dass die Daten vom Empfänger übernommen werden sollen, erhöht der Sender seinen SequenceCounter.

Wenn die Kommunikationsrichtung synchronisiert ist, erkennt die Gegenstelle den inkrementierten SequenceCounter. Die aktuelle Sequenz wird an das Empfangsarray angefügt und per SequenceAck bestätigt. Mit dieser Bestätigung wird dem Sender signalisiert, dass die MTU wieder neu befüllt werden kann.

Bei erfolgreicher Übertragung entsprechen die Daten im Empfangsarray exakt denen im Sendearray. Während der Übertragung muss die Empfangsstation die ankommenden Controlbytes erkennen und auswerten. Für jede Nachricht sollte ein separates Empfangsarray angelegt werden. Auf diese Weise kann der Empfänger vollständig übertragene Nachrichten sofort weiterverarbeiten.

3.8.4 Die Register für den Flatstream-Modus

Zur Konfiguration des Flatstreams sind 5 Register vorgesehen. Mit der Standardkonfiguration können geringe Datenmengen relativ einfach übermittelt werden.



Information:

Die Steuerung kommuniziert über die Register "OutputSequence" und "InputSequence" sowie den aktivierten Tx- bzw. RxBytes direkt mit dem Feldgerät. Deshalb benötigt der Anwender ausreichend Kenntnisse über das Kommunikationsprotokoll des Feldgerätes.

3.8.4.1 Konfiguration des Flatstreams

Um den Flatstream zu nutzen, muss der Programmablauf erweitert werden. Die Zykluszeit der Flatstream-Routinen muss auf ein Vielfaches des Buszyklus festgelegt werden. Die zusätzlichen Programmroutinen sollten in Cyclic #1 implementiert werden, um die Datenkonsistenz zu gewährleisten.

Bei der Minimalkonfiguration müssen die Register "InputMTU" und "OutputMTU" eingestellt werden. Alle anderen Register werden beim Start mit Standardwerten belegt und können sofort genutzt werden. Sie stellen zusätzliche Optionen bereit, um Daten kompakter zu übertragen bzw. den allgemeinen Ablauf hoch effizient zu gestalten.

Mit den Forward-Registern wird der Ablauf des Flatstream-Protokolls erweitert. Diese Funktion eignet sich, um die Datenrate des Flatstreams stark zu erhöhen, bedeutet aber erheblichen Mehraufwand bei der Erstellung des Programmablaufs.



Information:

In der weiteren Beschreibung stehen die Bezeichnungen "OutputMTU" und "InputMTU" nicht für die Registernamen, sondern als Synonym für die momentan aktivierten Tx- bzw. Rx-Bytes.



Information:

Die Register sind unter "[Flatstream-Register](#)" auf [Seite 61](#) beschrieben.

Die Register sind in den jeweiligen Datenblättern in Abschnitt "Die Flatstream-Kommunikation" beschrieben.

3.8.4.2 Bedienung des Flatstreams

Bei der Verwendung des Flatstreams ist die Kommunikationsrichtung von großer Bedeutung. Für das Senden von Daten an ein Modul (Output-Richtung) werden die Tx-Bytes genutzt. Für den Empfang von Daten eines Moduls (Input-Richtung) sind die Rx-Bytes vorgesehen.

Mit den Registern "OutputSequence" und "InputSequence" wird die Kommunikation gesteuert bzw. abgesichert, das heißt, der Sender gibt damit die Anweisung, Daten zu übernehmen und der Empfänger bestätigt eine erfolgreich übertragene Sequenz.



Information:

Die Register sind unter "[Flatstream-Register](#)" auf [Seite 61](#) beschrieben.

Die Register sind in den jeweiligen Datenblättern in Abschnitt "Die Flatstream-Kommunikation" beschrieben.

3.8.4.2.1 Format der Ein- und Ausgangsbytes

Name:

"Format des Flatstream" im Automation Studio

Bei einigen Modulen kann mit Hilfe dieser Funktion eingestellt werden, wie die Ein- und Ausgangsbytes des Flatstream (Tx- bzw. Rx-Bytes) übergeben werden.

- **gepackt:** Daten werden als ein Array übergeben
- **byteweise:** Daten werden als einzelne Bytes übergeben

3.8.4.2.2 Transport der Nutzdaten und der Controlbytes

Die Tx- bzw. Rx-Bytes sind zyklische Register, die zum Transport der Nutzdaten und der notwendigen Controlbytes dienen. Die Anzahl aktiver Tx- bzw. Rx-Bytes ergibt sich aus der Konfiguration der Register "OutputMTU" bzw. "InputMTU".

Im Programmablauf des Anwenders können nur die Tx- bzw. Rx-Bytes der Steuerung genutzt werden. Innerhalb des Moduls gibt es die entsprechenden Gegenstücke, welche für den Anwender nicht zugänglich sind. Aus diesem Grund wurden die Bezeichnungen aus Sicht der Steuerung gewählt.

- "T" - "transmit" → Steuerung sendet Daten an das Modul
- "R" - "receive" → Steuerung empfängt Daten vom Modul

3.8.4.2.2.1 Controlbytes

Neben den Nutzdaten übertragen die Tx- bzw. Rx-Bytes auch die sogenannten Controlbytes. Sie enthalten zusätzliche Informationen über den Datenstrom, damit der Empfänger die übertragenen Segmente wieder korrekt zur ursprünglichen Nachricht zusammensetzen kann.

Bitstruktur eines Controlbytes

Bit	Bezeichnung	Wert	Information
0 - 5	SegmentLength	0 - 63	Bytegröße des folgenden Segments (Standard: max. MTU-Größe - 1)
6	nextCBPos	0	Nächstes Controlbyte zu Beginn der nächsten MTU
		1	Nächstes Controlbyte direkt nach Ende des Segments
7	MessageEndBit	0	Nachricht wird nach dem folgenden Segment fortgesetzt
		1	Nachricht wird durch das folgende Segment beendet

SegmentLength

Die Segmentlänge kündigt dem Empfänger an, wie lang das kommende Segment ist. Wenn die eingestellte Segmentlänge für eine Nachricht nicht ausreicht, muss die Mitteilung auf mehrere Segmente verteilt werden. In diesen Fällen kann das tatsächliche Ende der Nachricht über Bit 7 (Controlbyte) erkannt werden.



Information:

Bei der Bestimmung der Segmentlänge wird das Controlbyte nicht mitgerechnet. Die Segmentlänge ergibt sich rein aus den Bytes der Nutzdaten.

nextCBPos

Mit diesem Bit wird angezeigt, an welcher Position das nächste Controlbyte zu erwarten ist. Diese Information ist vor allem bei Anwendung der Option "MultiSegmentMTU" wichtig.

Bei der Flatstream-Kommunikation mit MultiSegmentMTUs ist das nächste Controlbyte nicht mehr auf dem ersten Rx-Byte der darauffolgenden MTU zu erwarten, sondern wird direkt im Anschluss an das Segment übertragen.

MessageEndBit

Das "MessageEndBit" wird gesetzt, wenn das folgende Segment eine Nachricht abschließt. Die Mitteilung ist vollständig übertragen und kann weiterverarbeitet werden.



Information:

In Output-Richtung muss dieses Bit auch dann gesetzt werden, wenn ein einzelnes Segment ausreicht, um die vollständige Nachricht aufzunehmen. Das Modul verarbeitet eine Mitteilung intern nur, wenn diese Kennzeichnung vorgenommen wurde. Die Größe einer übertragenen Mitteilung lässt sich berechnen, wenn alle Segmentlängen der Nachricht addiert werden.

Flatstream-Formel zur Berechnung der Nachrichtenlänge:

Nachricht [Byte] = Segmentlängen (aller CBs ohne ME) + Segmentlänge (des ersten CB mit ME)	CB	Controlbyte
	ME	MessageEndBit

3.8.4.2.3 Kommunikationsstatus

Der Kommunikationsstatus wird über die Register "OutputSequence" und "InputSequence" ermittelt.

- OutputSequence enthält Informationen über den Kommunikationsstatus der Steuerung. Es wird von der Steuerung geschrieben und vom Modul gelesen.
- InputSequence enthält Informationen über den Kommunikationsstatus des Moduls. Es wird vom Modul geschrieben und sollte von der Steuerung nur gelesen werden.

3.8.4.2.3.1 Beziehung zwischen Output- und InputSequence

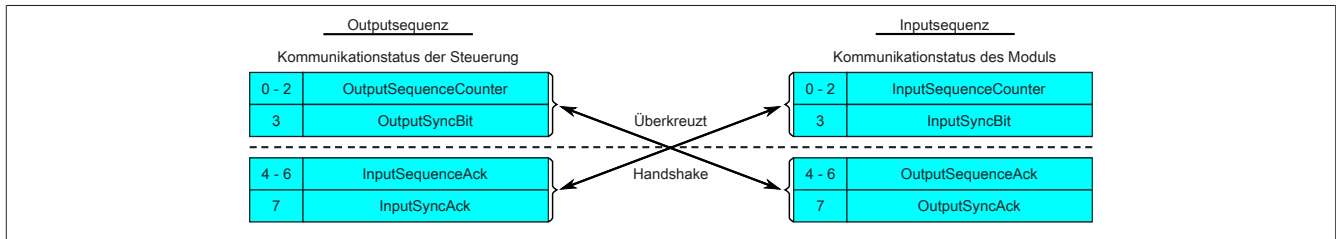


Abbildung 3: Zusammenhang zwischen Output- und InputSequence

Die Register OutputSequence und InputSequence sind logisch aus 2 Halb-Bytes aufgebaut. Über den Low-Teil wird der Gegenstelle signalisiert, ob ein Kanal geöffnet werden soll bzw. ob Daten übernommen werden können. Der High-Teil dient zur Bestätigung, wenn die geforderte Aktion erfolgreich ausgeführt wurde.

SyncBit und SyncAck

Wenn das SyncBit und das SyncAck einer Kommunikationsrichtung gesetzt sind, gilt der Kanal als "synchronisiert", das heißt, es können Nachrichten in diese Richtung versendet werden. Das Statusbit der Gegenstelle muss zyklisch überprüft werden. Falls das SyncAck zurückgesetzt wurde, muss das eigene SyncBit angepasst werden. Bevor neue Daten übertragen werden können, muss der Kanal resynchronisiert werden.

SequenceCounter und SequenceAck

Die Kommunikationspartner prüfen zyklisch, ob sich das Low-Nibble der Gegenstelle ändert. Wenn ein Kommunikationspartner eine neue Sequenz vollständig auf die MTU geschrieben hat, erhöht er seinen SequenceCounter. Daraufhin übernimmt der Empfänger die aktuelle Sequenz und bestätigt den erfolgreichen Empfang per SequenceAck. Auf diese Weise wird ein Handshake-Verfahren initiiert.



Information:

Bei einer Unterbrechung der Kommunikation werden Segmente von unvollständig übermittelten Mitteilungen verworfen. Alle fertig übertragenen Nachrichten werden bearbeitet.

3.8.4.3 Synchronisieren

Beim Synchronisieren wird ein Kommunikationskanal geöffnet. Es muss sichergestellt sein, dass ein Modul vorhanden und der aktuelle Wert des SequenceCounters beim Empfänger der Nachricht hinterlegt ist. Der Flatstream bietet die Möglichkeit Vollduplex zu kommunizieren. Beide Kanäle/Kommunikationsrichtungen können separat betrachtet werden. Sie müssen unabhängig voneinander synchronisiert werden, so dass theoretisch auch simplex kommuniziert werden könnte.

Synchronisation der Output-Richtung (Steuerung als Sender)

Die korrespondierenden Synchronisationsbits (OutputSyncBit und OutputSyncAck) sind zurückgesetzt. Aus diesem Grund können momentan keine Nachrichten von der Steuerung an das Modul per Flatstream übertragen werden.

Algorithmus

1) Steuerung muss 000 in OutputSequenceCounter schreiben und OutputSyncBit zurücksetzen. Steuerung muss High-Nibble des Registers "InputSequence" zyklisch abfragen (Prüfung ob 000 in OutputSequenceAck und 0 in OutputSyncAck). Modul übernimmt den aktuellen Inhalt der InputMTU nicht, weil der Kanal noch nicht synchronisiert ist. Modul gleicht OutputSequenceAck und OutputSyncAck an die Werte des OutputSequenceCounters bzw. des OutputSyncBits an.
2) Wenn die Steuerung die erwarteten Werte in OutputSequenceAck und OutputSyncAck registriert, darf sie den OutputSequenceCounter inkrementieren. Die Steuerung fragt das High-Nibble des Registers "OutputSequence" weiter zyklisch ab (Prüfung ob 001 in OutputSequenceAck und 0 in InputSyncAck). Modul übernimmt den aktuellen Inhalt der InputMTU nicht, weil der Kanal noch nicht synchronisiert ist. Modul gleicht OutputSequenceAck und OutputSyncAck an die Werte des OutputSequenceCounters bzw. des OutputSyncBits an.
3) Wenn die Steuerung die erwarteten Werte in OutputSequenceAck und OutputSyncAck registriert, darf sie das OutputSyncBit setzen. Die Steuerung fragt das High-Nibble des Registers "OutputSequence" weiter zyklisch ab (Prüfung ob 001 in OutputSequenceAck und 1 in InputSyncAck).
Hinweis: Theoretisch könnten ab diesem Moment Daten übertragen werden. Es wird allerdings empfohlen, erst dann Daten zu übertragen, wenn die Output-Richtung vollständig synchronisiert ist.
Modul setzt OutputSyncAck.
Output-Richtung synchronisiert, Steuerung kann Daten an Modul senden.

Synchronisation der Input-Richtung (Steuerung als Empfänger)

Die korrespondierenden Synchronisationsbits (InputSyncBit und InputSyncAck) sind zurückgesetzt. Aus diesem Grund können momentan keine Nachrichten vom Modul an die Steuerung per Flatstream übertragen werden.

Algorithmus

Modul schreibt 000 in InputSequenceCounter und setzt InputSyncBit zurück. Modul überwacht High-Nibble des Registers "OutputSequence" - erwartet 000 in InputSequenceAck bzw. 0 in InputSyncAck.
1) Steuerung darf den aktuellen Inhalt der InputMTU nicht übernehmen, weil der Kanal noch nicht synchronisiert ist. Steuerung muss InputSequenceAck und InputSyncAck an die Werte des InputSequenceCounters bzw. des InputSyncBits angleichen. Wenn das Modul die erwarteten Werte in InputSequenceAck und in InputSyncAck registriert, inkrementiert es den InputSequenceCounter. Modul überwacht High-Nibble des Registers "OutputSequence" - erwartet 001 in InputSequenceAck bzw. 0 in InputSyncAck.
2) Steuerung darf den aktuellen Inhalt der InputMTU nicht übernehmen, weil der Kanal noch nicht synchronisiert ist. Steuerung muss InputSequenceAck und InputSyncAck an die Werte des InputSequenceCounters bzw. des InputSyncBits angleichen. Wenn das Modul die erwarteten Werte in InputSequenceAck und in InputSyncAck registriert, setzt es das InputSyncBit. Modul überwacht High-Nibble des Registers "OutputSequence" - erwartet 1 in InputSyncAck.
3) Steuerung darf InputSyncAck setzen.
Hinweis: Theoretisch könnten bereits in diesem Zyklus Daten übertragen werden. Es gilt: Wenn das InputSyncBit gesetzt ist und der InputSequenceCounter um 1 erhöht wurde, müssen die Informationen der aktivierten Rx-Bytes übernommen und bestätigt werden (siehe dazu auch Kommunikation in Input-Richtung).
Input-Richtung synchronisiert, Modul kann Daten an Steuerung senden.

3.8.4.4 Senden und Empfangen

Wenn ein Kanal synchronisiert ist, gilt die Gegenstelle als empfangsbereit und der Sender kann Nachrichten verschicken. Bevor der Sender Daten absetzen kann, legt er das sogenannte Sendearray an, um den Anforderungen des Flatstreams gerecht zu werden.

Die sendende Station muss für jedes erstellte Segment ein individuelles Controlbyte generieren. Ein solches Controlbyte enthält Informationen, wie der nächste Teil der übertragenen Daten zu verarbeiten ist. Die Position des nächsten Controlbytes im Datenstrom kann variieren. Aus diesem Grund muss zu jedem Zeitpunkt eindeutig definiert sein, wann ein neues Controlbyte übermittelt wird. Das erste Controlbyte befindet sich immer auf dem ersten Byte der ersten Sequenz. Alle weiteren Positionen werden rekursiv mitgeteilt.

Flatstream-Formel zur Berechnung der Position des nächsten Controlbytes:

$$\text{Position (nächstes Controlbyte)} = \text{aktuelle Position} + 1 + \text{Segmentlänge}$$

Beispiel

Es werden 3 unabhängige Nachrichten (7 Bytes, 2 Bytes, 9 Bytes) über eine 7-Byte breite MTU übermittelt. Die sonstige Konfiguration entspricht den Standardeinstellungen.

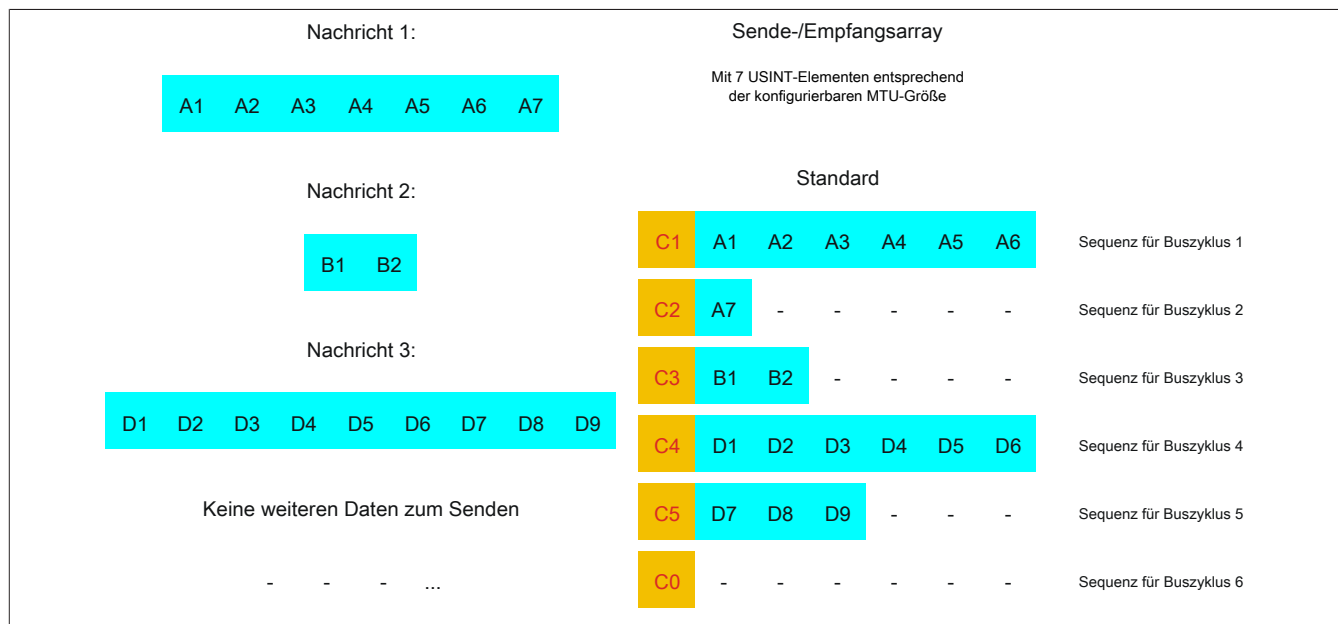


Abbildung 4: Sende-/Empfangsarray (Standard)

Funktionsbeschreibung

Zunächst müssen die Nachrichten in Segmente geteilt werden. Bei der Standardkonfiguration muss sichergestellt sein, dass jede Sequenz ein gesamtes Segment inklusive dem dazugehörigen Controlbyte aufnehmen kann. Die Sequenz ist auf die Größe der aktivierten MTU begrenzt, das heißt, ein Segment muss mindestens um 1 Byte kleiner sein als die aktivierte MTU.

MTU = 7 Bytes → max. Segmentlänge 6 Bytes

- Nachricht 1 (7 Bytes)
 - ⇒ erstes Segment = Controlbyte + 6 Datenbytes
 - ⇒ zweites Segment = Controlbyte + 1 Datenbyte
- Nachricht 2 (2 Bytes)
 - ⇒ erstes Segment = Controlbyte + 2 Datenbytes
- Nachricht 3 (9 Bytes)
 - ⇒ erstes Segment = Controlbyte + 6 Datenbytes
 - ⇒ zweites Segment = Controlbyte + 3 Datenbytes
- Keine weiteren Nachrichten
 - ⇒ C0-Controlbyte

Für jedes gebildete Segment muss ein spezifisches Controlbyte generiert werden. Außerdem wird das Controlbyte C0 generiert, um die Kommunikation auf Standby halten zu können.

C0 (Controlbyte0)			C1 (Controlbyte1)			C2 (Controlbyte2)		
- 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
Controlbyte	Σ	0	Controlbyte	Σ	6	Controlbyte	Σ	129

Tabelle 3: Flatstream-Ermittlung der Controlbytes für Beispiel mit Standardkonfiguration (Teil 1)

C3 (Controlbyte3)			C4 (Controlbyte4)			C5 (Controlbyte5)		
- 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
Controlbyte	Σ	130	Controlbyte	Σ	6	Controlbyte	Σ	131

Tabelle 4: Flatstream-Ermittlung der Controlbytes für Beispiel mit Standardkonfiguration (Teil 2)

3.8.4.4.1 Senden von Daten an ein Modul (Output)

Beim Senden muss das Sendearray im Programmablauf generiert werden. Danach wird es Sequenz für Sequenz über den Flatstream übertragen und vom Modul empfangen.



Information:

Obwohl alle B&R Module mit Flatstream-Kommunikation stets die kompakteste Übertragung in Output-Richtung unterstützen wird empfohlen die Übertragungsarrays für beide Kommunikationsrichtungen gleichermaßen zu gestalten.

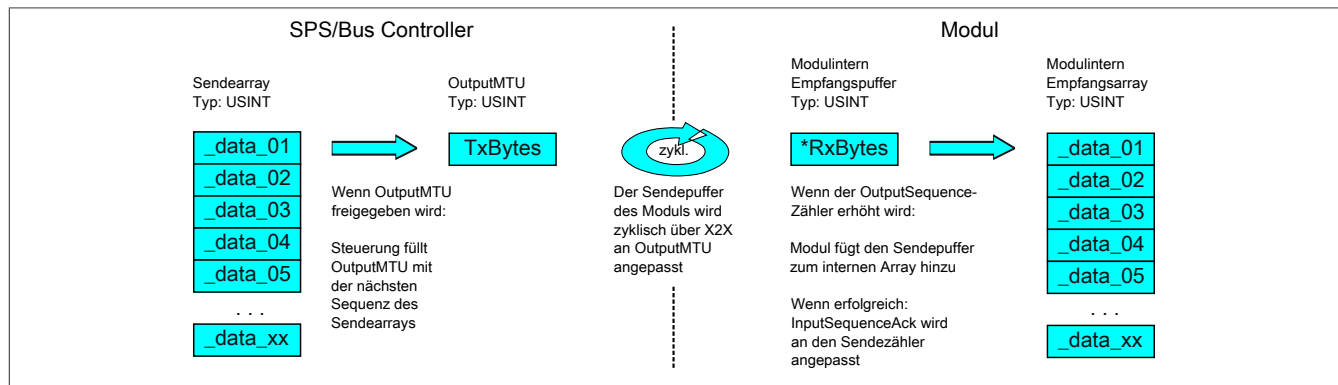


Abbildung 5: Kommunikation per Flatstream (Output)

Nachricht kleiner als OutputMTU

Die Länge der Nachricht sei zunächst kleiner als die OutputMTU. In diesem Fall würde eine Sequenz ausreichen, um die gesamte Nachricht und ein benötigtes Controlbyte zu übertragen.

Algorithmus

Zyklische Statusabfrage: - Modul überwacht OutputSequenceCounter
0) Zyklische Prüfungen: - Steuerung muss OutputSyncAck prüfen → falls OutputSyncAck = 0; OutputSyncBit zurücksetzen und Kanal resynchronisieren - Steuerung muss Freigabe der OutputMTU prüfen → falls OutputSequenceCounter > InputSequenceAck; MTU nicht freigegeben, weil letzte Sequenz noch nicht bestätigt
1) Vorbereitung (Sendearray anlegen): - Steuerung muss Nachricht auf zulässige Segmente aufteilen und entsprechende Controlbytes bilden - Steuerung muss Segmente und Controlbytes zu Sendearray zusammenfügen
2) Senden: - Steuerung überträgt das aktuelle Element des Sendearrays in die OutputMTU → OutputMTU wird zyklisch in den Sendepuffer des Moduls übertragen, aber noch nicht weiterverarbeitet - Steuerung muss OutputSequenceCounter erhöhen
Reaktion: - Modul übernimmt die Bytes des internen Empfangspuffers und fügt sie an das interne Empfangsarray an - Modul sendet Bestätigung; schreibt Wert des OutputSequenceCounters auf OutputSequenceAck
3) Abschluss: - Steuerung muss OutputSequenceAck überwachen → Eine Sequenz gilt erst dann als erfolgreich übertragen, wenn sie über das OutputSequenceAck bestätigt wurde. Um Übertragungsfehler auch bei der letzten Sequenz zu erkennen, muss sichergestellt werden, dass der Abschluss lange genug durchlaufen wird.
Hinweis: Für eine exakte Überwachung der Kommunikationszeiten sollten die Taskzyklen gezählt werden, die seit der letzten Erhöhung des OutputSequenceCounters vergangen sind. Auf diese Weise kann die Anzahl der Buszyklen abgeschätzt werden, die bislang zur Übertragung benötigt wurden. Übersteigt der Überwachungszähler eine vorgegebene Schwelle, kann die Sequenz als verloren betrachtet werden. (Das Verhältnis von Bus- und Taskzyklus kann vom Anwender beeinflusst werden, sodass der Schwellwert individuell zu ermitteln ist.) - Weitere Sequenzen dürfen erst nach erfolgreicher Abschlussprüfung im nächsten Buszyklus versendet werden.

Nachricht größer als OutputMTU

Das Sendearray, welches im Programmablauf erstellt werden muss, besteht aus mehreren Elementen. Der Anwender muss die Control- und Datenbytes korrekt anordnen und die Arrayelemente nacheinander übertragen. Der Übertragungsalgorithmus bleibt gleich und wird ab dem Punkt zyklische Prüfungen wiederholt durchlaufen.

Allgemeines Ablaufdiagramm

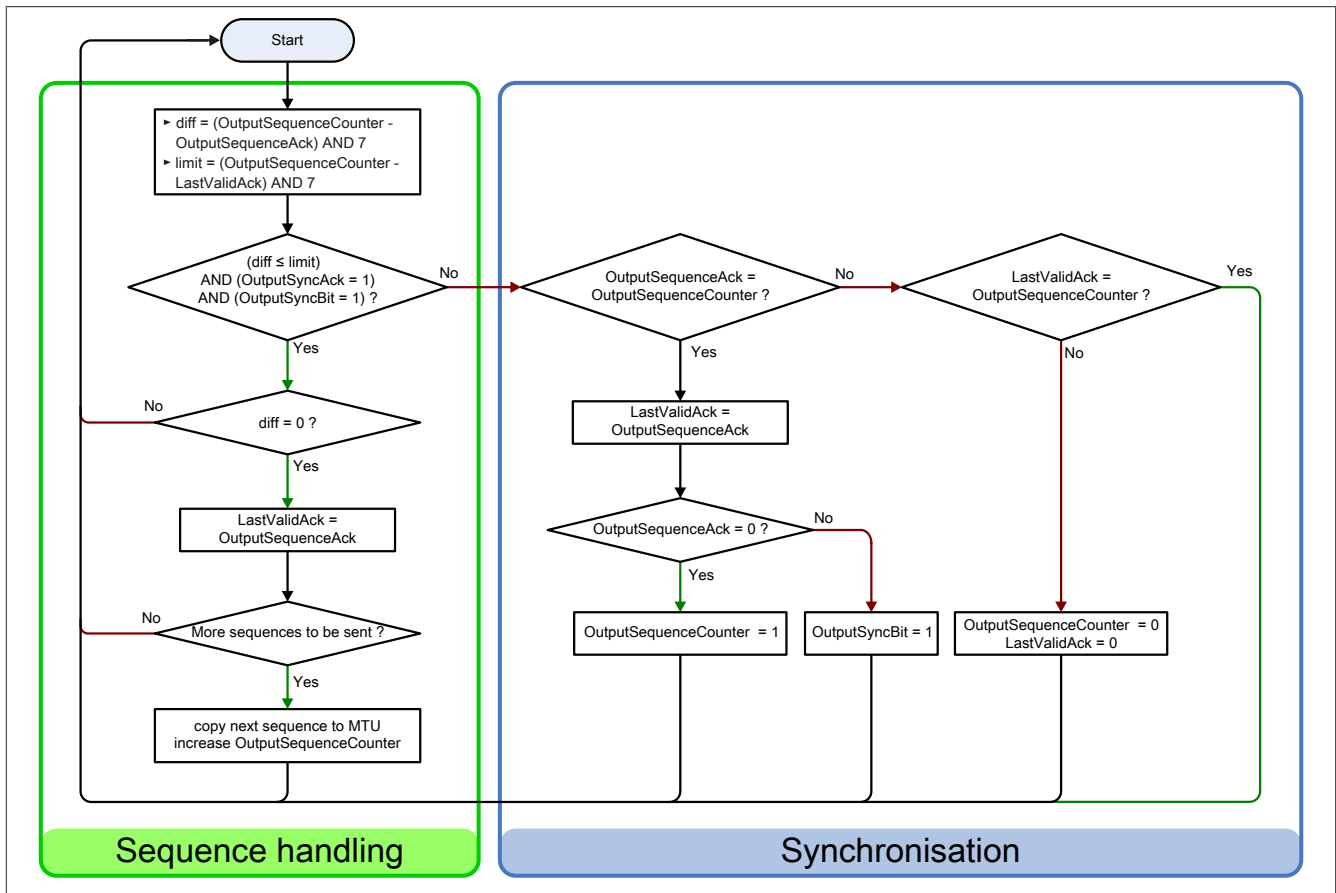


Abbildung 6: Ablaufdiagramm für Output-Richtung

3.8.4.4.2 Empfangen von Daten aus einem Modul (Input)

Beim Empfangen von Daten wird das Sendearray vom Modul generiert, über den Flatstream übertragen und muss auf dem Empfangsarray abgebildet werden. Die Struktur des ankommenden Datenstroms kann über das Modusregister eingestellt werden. Der Algorithmus zum Empfangen bleibt dabei aber unverändert.

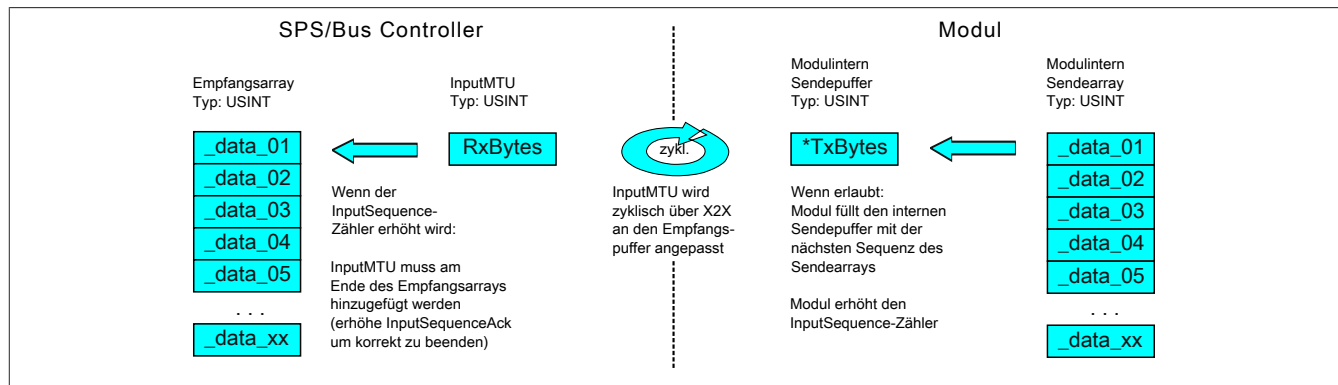


Abbildung 7: Kommunikation per Flatstream (Input)

Algorithmus

0) Zyklische Statusabfrage: - Steuerung muss InputSequenceCounter überwachen
Zyklische Prüfungen: - Modul prüft InputSyncAck - Modul prüft InputSequenceAck
Vorbereitung: - Modul bildet Segmente bzw. Controlbytes und legt Sendearray an
Aktion: - Modul überträgt das aktuelle Element des internen Sendearrays in den internen Sendepuffer - Modul erhöht InputSequenceCounter
1) Empfangen (sobald InputSequenceCounter erhöht): - Steuerung muss Daten aus InputMTU übernehmen und an das Ende des Empfangsarrays anfügen - Steuerung muss InputSequenceAck an InputSequenceCounter der aktuell verarbeiteten Sequenz angleichen
Abschluss: - Modul überwacht InputSequenceAck → Eine Sequenz gilt erst dann als erfolgreich übertragen, wenn sie über das InputSequenceAck bestätigt wurde. - Weitere Sequenzen werden erst nach erfolgreicher Abschlussprüfung im nächsten Buszyklus versendet.

Allgemeines Ablaufdiagramm

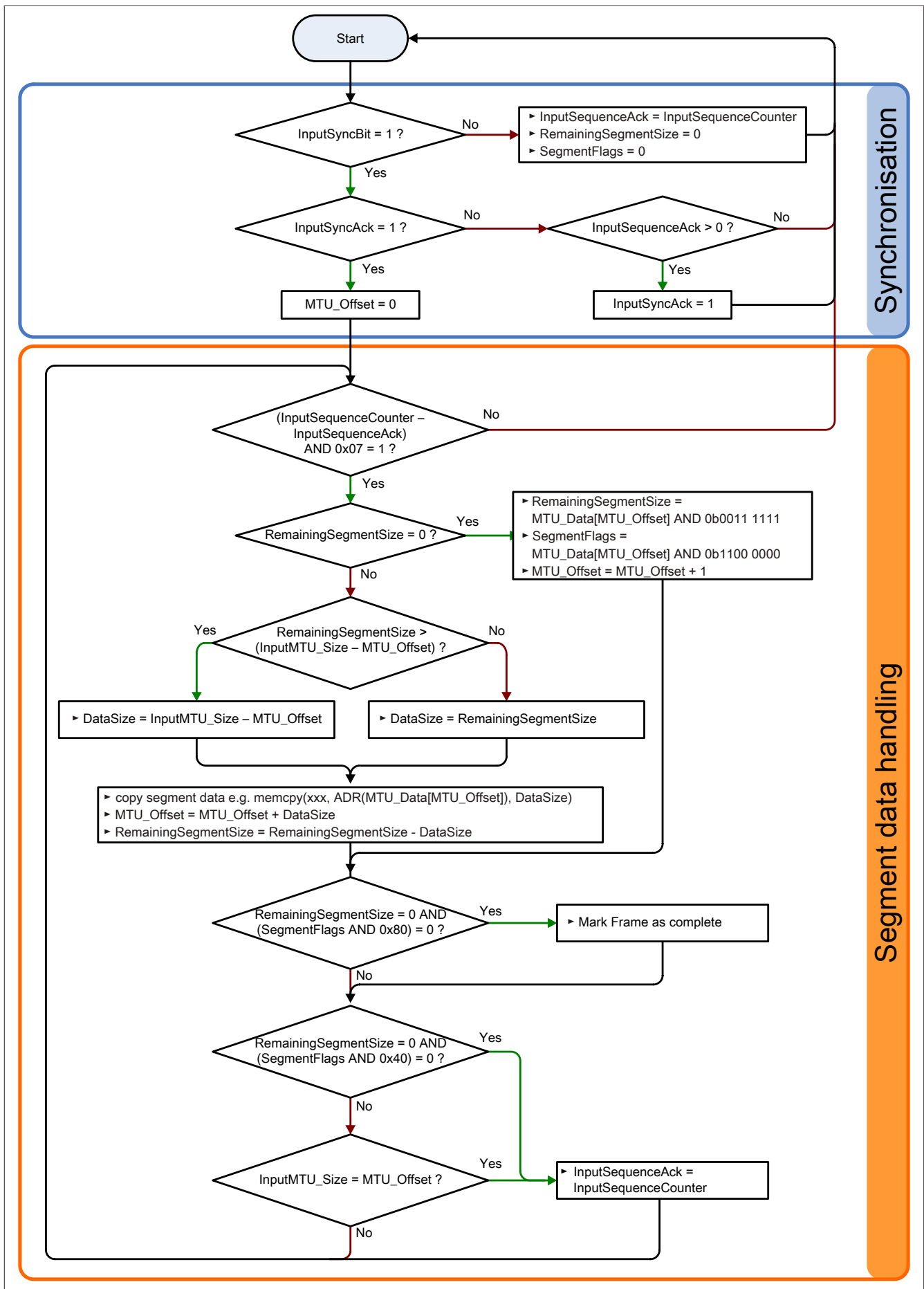


Abbildung 8: Ablaufdiagramm für Input-Richtung

3.8.4.4.3 Details

Es wird empfohlen die übertragenen Nachrichten in separate Empfangsarrays abzulegen

Nach der Übermittlung eines gesetzten MessageEndBits sollte das Folgesegment zum Empfangsarray hinzugefügt werden. Danach ist die Mitteilung vollständig und kann intern weiterverarbeitet werden. Für die nächste Nachricht sollte ein neues/separates Array angelegt werden.



Information:

Bei der Übertragung mit MultiSegmentMTUs können sich mehrere kurze Nachrichten in einer Sequenz befinden. Im Programmablauf muss sichergestellt sein, dass genügend Empfangsarrays verwaltet werden können. Das Acknowledge-Register darf erst nach Übernahme der gesamten Sequenz angepasst werden.

Wenn ein SequenceCounter um mehr als einen Zähler inkrementiert wird, liegt ein Fehler vor

In diesem Fall stoppt der Empfänger. Alle weiteren eintreffenden Sequenzen werden ignoriert, bis die Sendung mit dem korrekten SequenceCounter wiederholt wird. Durch diese Reaktion erhält der Sender keine Bestätigungen mehr für die abgesetzten Sequenzen. Über den SequenceAck der Gegenstelle kann der Sender die letzte erfolgreich übertragene Sequenz identifizieren und die Übertragung ab dieser Stelle fortsetzen.



Information:

Beim Betrieb ohne Forward ist diese Situation sehr unwahrscheinlich.

Bestätigungen müssen auf Gültigkeit geprüft werden

Wenn der Empfänger eine Sequenz erfolgreich übernommen hat, muss sie bestätigt werden. Dazu übernimmt der Empfänger den mitgesendeten Wert des SequenceCounters und gleicht den SequenceAck daran an. Der Absender liest das SequenceAck und registriert die erfolgreiche Übermittlung. Falls dem Absender eine Sequenz bestätigt wird, die noch nicht abgesendet wurde, muss die Übertragung unterbrochen und der Kanal resynchronisiert werden. Die Synchronisationsbits werden zurückgesetzt und die aktuelle/unvollständige Nachricht wird verworfen. Sie muss nach der Resynchronisierung des Kanals erneut versendet werden.

3.8.4.5 Flatstream-Modus

In Input-Richtung wird das Sende-Array automatisch generiert. Dem Anwender werden über den Flatstream-Modus mehrere Optionen zur Verfügung gestellt, um eine kompaktere Anordnung beim eintreffenden Datenstrom zu erlauben. Diese sind:

- Standard
- MultiSegmentMTU erlaubt
- Große Segmente erlaubt

Nach der Aktivierung muss der Programmablauf zur Auswertung entsprechend angepasst werden.



Information:

Alle B&R Module, die den Flatstream-Modus anbieten, unterstützen in Output-Richtung die Optionen "große Segmente" und "MultiSegmentMTU". Nur für die Input-Richtung muss die kompakte Übertragung explizit erlaubt werden.

Standard

Per Standard sind beide Optionen zur kompakten Übertragung in Input-Richtung deaktiviert.

1. Vom Modul werden nur Segmente gebildet, die mindestens ein Byte kleiner sind als die aktivierte MTU. Jede Sequenz beginnt mit einem Controlbyte, sodass der Datenstrom klar strukturiert ist und relativ einfach ausgewertet werden kann.
2. Weil die Länge einer Flatstream-Nachricht beliebig lang sein darf, füllt das letzte Segment der Mitteilung häufig nicht den gesamten Platz der MTU aus. Per Standard werden während eines solchen Übertragungszyklus die restlichen Bytes nicht verwendet.

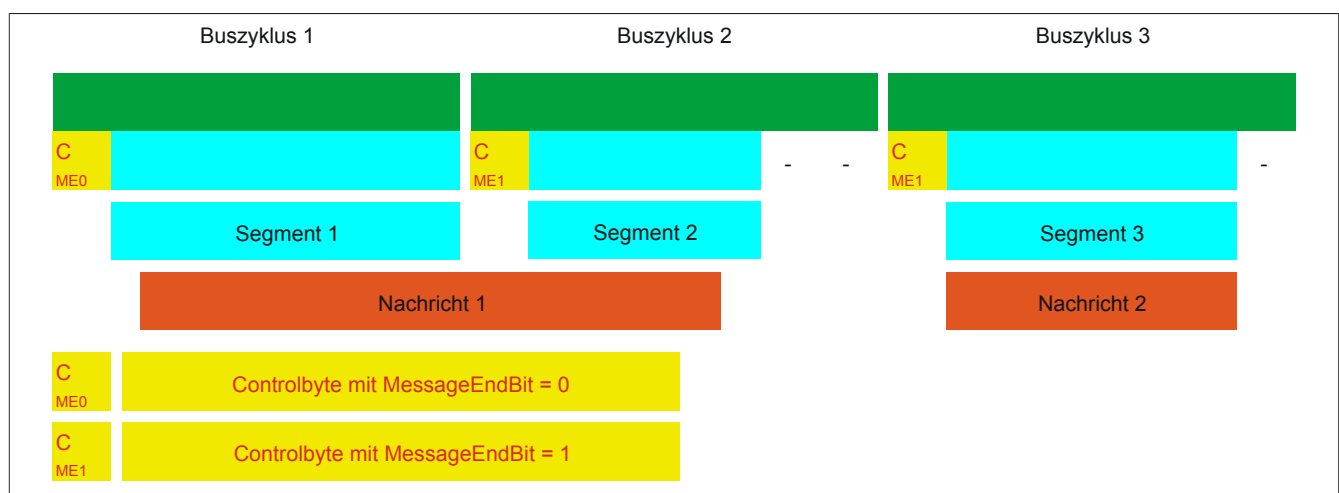


Abbildung 9: Anordnung von Nachrichten in der MTU (Standard)

MultiSegmentMTU erlaubt

Bei dieser Option wird die InputMTU vollständig befüllt (wenn genügend Daten anstehen). Die zuvor frei gebliebenen Rx-Bytes übertragen die nächsten Controlbytes bzw. deren Segmente. Auf diese Weise können die aktivierten Rx-Bytes effizienter genutzt werden.

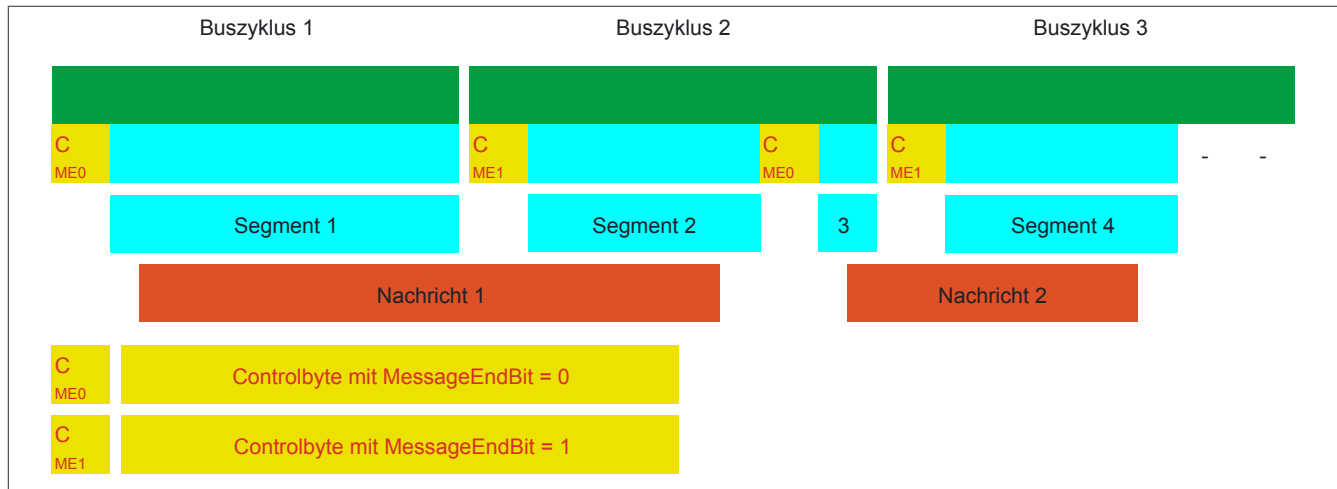


Abbildung 10: Anordnung von Nachrichten in der MTU (MultiSegmentMTU)

Große Segmente erlaubt

Bei der Übertragung sehr langer Mitteilungen bzw. bei der Aktivierung von nur wenigen Rx-Bytes müssen per Standard sehr viele Segmente gebildet werden. Das Bussystem wird stärker belastet als nötig, weil für jedes Segment ein zusätzliches Controlbyte erstellt und übertragen wird. Mit der Option "große Segmente" wird die Segmentlänge unabhängig von der InputMTU auf 63 Bytes begrenzt. Ein Segment darf sich über mehrere Sequenzen erstrecken, das heißt, es können auch reine Sequenzen ohne Controlbyte auftreten.



Information:

Die Möglichkeit eine Nachricht auf mehrere Segmente aufzuteilen bleibt erhalten, das heißt, wird diese Option genutzt und treten Nachrichten mit mehr als 63 Bytes auf, kann die Mitteilung weiterhin auf mehrere Segmente verteilt werden.

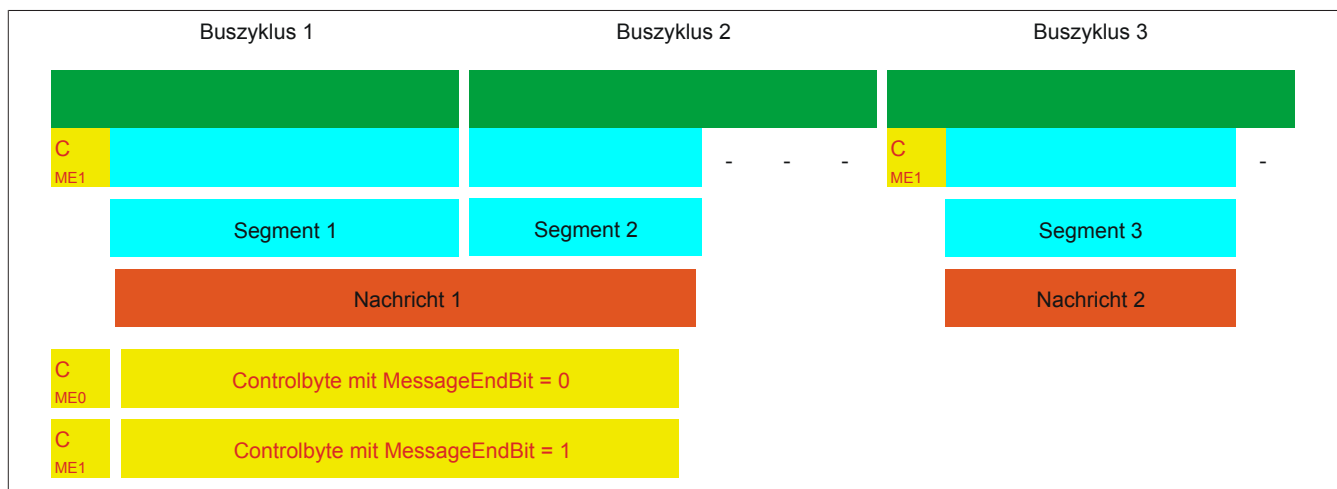


Abbildung 11: Anordnung von Nachrichten in der MTU (große Segmente)

Funktionsbeschreibung

Anwendung beider Optionen

Die beiden Optionen dürfen auch gleichzeitig angewendet werden.

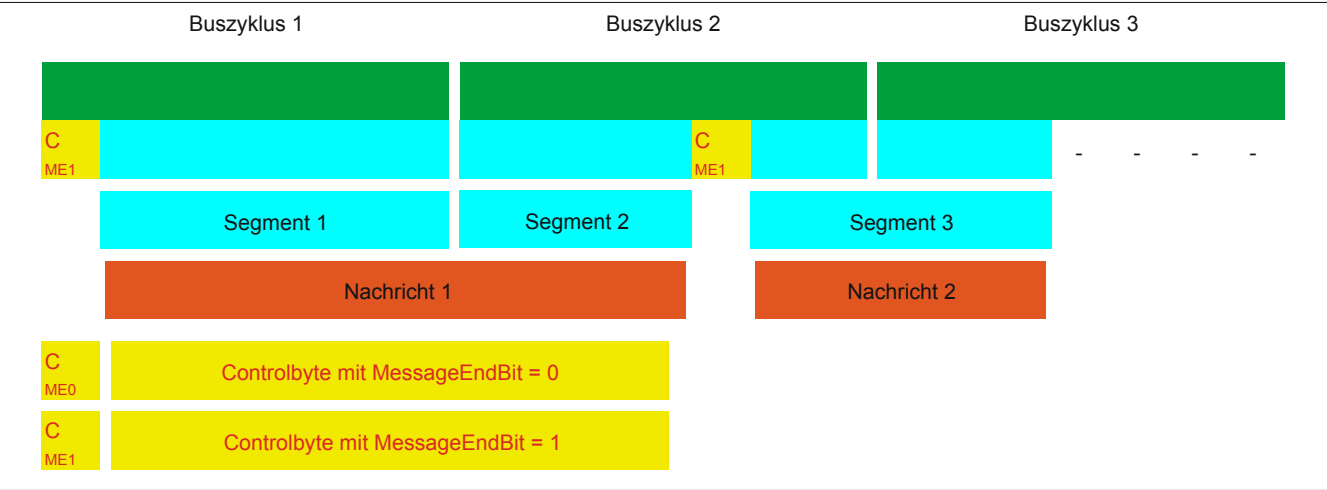


Abbildung 12: Anordnung von Nachrichten in der MTU (große Segmente und MultiSegmentMTU)

3.8.4.6 Anpassung des Flatstreams

Wenn die Strukturierung der Nachrichten verändert wurde, verändert sich auch die Anordnung der Daten im Sende-/Empfangsarray. Für das eingangs genannte Beispiel ergeben sich die folgenden Änderungen.

MultiSegmentMTU

Wenn MultiSegmentMTUs erlaubt sind, können "freie Stellen" in einer MTU genutzt werden. Diese "freien Stellen" entstehen, wenn das letzte Segment einer Nachricht nicht die gesamte MTU ausnutzt. MultiSegmentMTUs ermöglichen die Verwendung dieser Bits, um die folgenden Controlbytes bzw. Segmente zu übertragen. Im Programmablauf wird das "nextCBPos"-Bit innerhalb des Controlbytes gesetzt, damit der Empfänger das nächste Controlbyte korrekt identifizieren kann.

Beispiel

Es werden 3 unabhängige Nachrichten (7 Bytes, 2 Bytes, 9 Bytes) über eine 7-Byte breite MTU übermittelt. Die Konfiguration erlaubt die Übertragung von MultiSegmentMTUs.

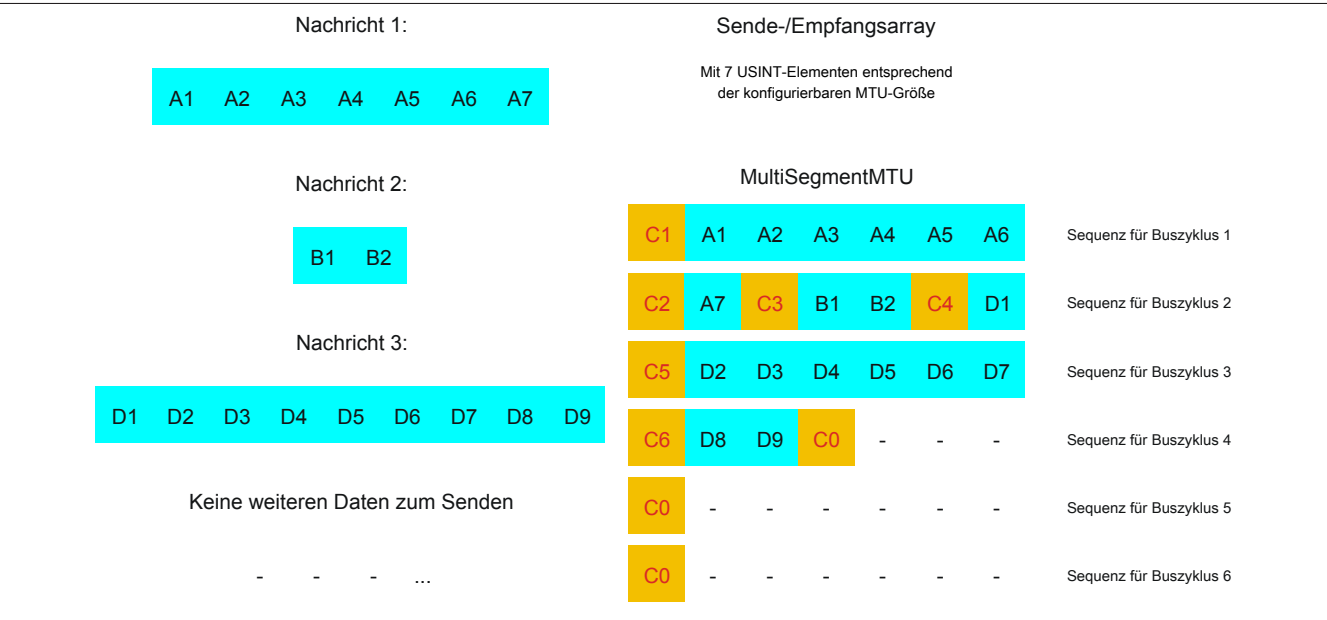


Abbildung 13: Sende-/Empfangsarray (MultiSegmentMTU)

Zunächst müssen die Nachrichten in Segmente geteilt werden. Wie in der Standardkonfiguration muss sichergestellt sein, dass jede Sequenz mit einem Controlbyte beginnt. Die freien Bits in der MTU am Ende einer Nachricht, werden allerdings mit Daten der Folgenachricht aufgefüllt. Bei dieser Option wird das Bit "nextCBPos" immer gesetzt, wenn im Anschluss an das Controlbyte Nutzdaten übertragen werden.

MTU = 7 Bytes → max. Segmentlänge 6 Bytes

- Nachricht 1 (7 Bytes)
 - ⇒ erstes Segment = Controlbyte + 6 Datenbytes (MTU voll)
 - ⇒ zweites Segment = Controlbyte + 1 Datenbyte (MTU noch 5 leere Bytes)
- Nachricht 2 (2 Bytes)
 - ⇒ erstes Segment = Controlbyte + 2 Datenbytes (MTU noch 2 leere Bytes)
- Nachricht 3 (9 Bytes)
 - ⇒ erstes Segment = Controlbyte + 1 Datenbyte (MTU voll)
 - ⇒ zweites Segment = Controlbyte + 6 Datenbytes (MTU voll)
 - ⇒ drittes Segment = Controlbyte + 2 Datenbytes (MTU noch 4 leere Bytes)
- Keine weiteren Nachrichten
 - ⇒ C0-Controlbyte

Für jedes gebildete Segment muss ein spezifisches Controlbyte generiert werden. Außerdem wird das Controlbyte C0 generiert, um die Kommunikation auf Standby halten zu können.

C1 (Controlbyte1)			C2 (Controlbyte2)			C3 (Controlbyte3)		
- 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
Controlbyte	Σ	70	Controlbyte	Σ	193	Controlbyte	Σ	194

Tabelle 5: Flatstream-Ermittlung der Controlbytes für Beispiel mit MultiSegmentMTU (Teil 1)



Warnung!

Die zweite Sequenz darf erst über den SequenceAck bestätigt werden, wenn sie vollständig verarbeitet wurde. Im Beispiel befinden sich 3 verschiedene Segmente innerhalb der zweiten Sequenz, das heißt, im Programmablauf müssen ausreichend Empfänger-Arrays gehandhabt werden können.



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 (Controlbyte4)			C5 (Controlbyte5)			C6 (Controlbyte6)		
- 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
Controlbyte	Σ	7	Controlbyte	Σ	70	Controlbyte	Σ	194

Tabelle 6: Flatstream-Ermittlung der Controlbytes für Beispiel mit MultiSegmentMTU (Teil 2)

Große Segmente

Die Segmente werden auf maximal 63 Bytes begrenzt. Damit können sie größer sein als die aktive MTU. Diese großen Segmente werden bei der Übertragung auf mehrere Sequenzen aufgeteilt. Es können Sequenzen ohne Controlbyte auftreten, die vollständig mit Nutzdaten befüllt sind.



Information:

Um die Größe eines Datenpakets nicht ebenfalls auf 63 Bytes zu begrenzen, bleibt die Möglichkeit erhalten, eine Nachricht in mehrere Segmente zu untergliedern.

Beispiel

Es werden 3 unabhängige Nachrichten (7 Bytes, 2 Bytes, 9 Bytes) über eine 7-Byte breite MTU übermittelt. Die Konfiguration erlaubt die Übertragung von großen Segmenten.

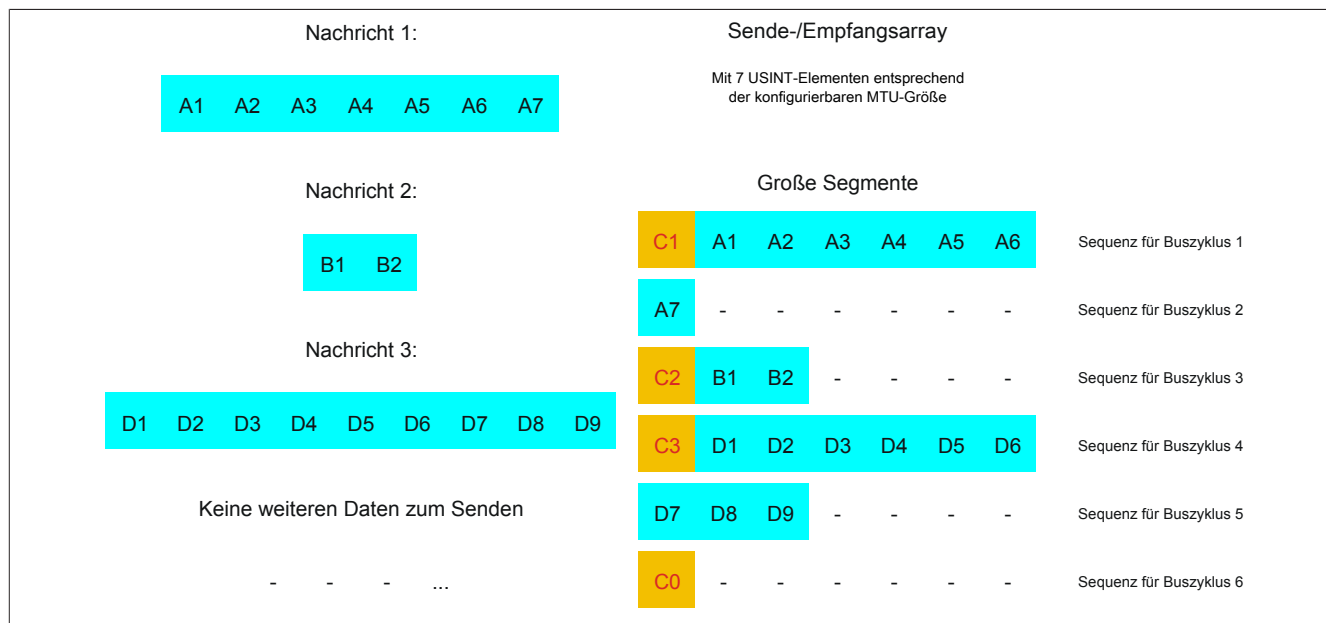


Abbildung 14: Sende-/Empfangsarray (große Segmente)

Zunächst müssen die Nachrichten in Segmente geteilt werden. Durch die Möglichkeit große Segmente zu bilden, müssen Nachrichten seltener geteilt werden, sodass weniger Controlbytes generiert werden müssen.

Große Segmente erlaubt → max. Segmentlänge 63 Bytes

- Nachricht 1 (7 Bytes)
 - ⇒ erstes Segment = Controlbyte + 7 Datenbytes
- Nachricht 2 (2 Bytes)
 - ⇒ erstes Segment = Controlbyte + 2 Datenbytes
- Nachricht 3 (9 Bytes)
 - ⇒ erstes Segment = Controlbyte + 9 Datenbytes
- Keine weiteren Nachrichten
 - ⇒ C0-Controlbyte

Für jedes gebildete Segment muss ein spezifisches Controlbyte generiert werden. Außerdem wird das Controlbyte C0 generiert, um die Kommunikation auf Standby halten zu können.

C1 (Controlbyte1)			C2 (Controlbyte2)			C3 (Controlbyte3)		
- 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
Controlbyte	Σ	135	Controlbyte	Σ	130	Controlbyte	Σ	137

Tabelle 7: Flatstream-Ermittlung der Controlbytes für Beispiel mit großen Segmenten

Große Segmente und MultiSegmentMTU

Beispiel

Es werden 3 unabhängige Nachrichten (7 Bytes, 2 Bytes, 9 Bytes) über eine 7-Byte breite MTU übermittelt. Die Konfiguration erlaubt sowohl die Übertragung von MultiSegmentMTUs als auch von großen Segmenten.

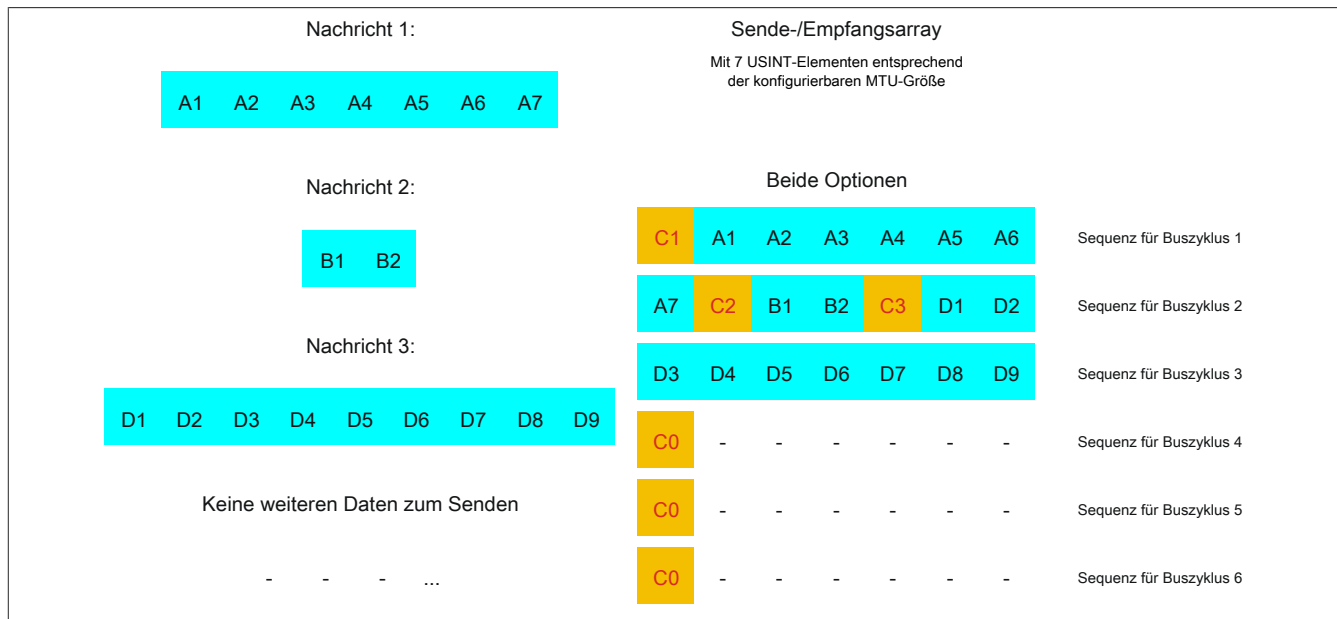


Abbildung 15: Sende-/Empfangsarray (große Segmente und MultiSegmentMTU)

Zunächst müssen die Nachrichten in Segmente geteilt werden. Wenn das letzte Segment einer Nachricht die MTU nicht komplett befüllt, darf sie für weitere Daten aus dem Datenstrom verwendet werden. Das Bit "nextCBPos" muss immer gesetzt werden, wenn das Controlbyte zu einem Segment mit Nutzdaten gehört. Durch die Möglichkeit große Segmente zu bilden, müssen Nachrichten seltener geteilt werden, sodass weniger Controlbytes generiert werden müssen. Die Generierung der Controlbytes erfolgt auf die gleiche Weise, wie bei der Option "große Segmente".

Große Segmente erlaubt → max. Segmentlänge 63 Bytes

- Nachricht 1 (7 Bytes)
 - ⇒ erstes Segment = Controlbyte + 7 Datenbytes
- Nachricht 2 (2 Bytes)
 - ⇒ erstes Segment = Controlbyte + 2 Datenbytes
- Nachricht 3 (9 Bytes)
 - ⇒ erstes Segment = Controlbyte + 9 Datenbytes
- Keine weiteren Nachrichten
 - ⇒ C0-Controlbyte

Für jedes gebildete Segment muss ein spezifisches Controlbyte generiert werden. Außerdem wird das Controlbyte C0 generiert, um die Kommunikation auf Standby halten zu können.

C1 (Controlbyte1)			C2 (Controlbyte2)			C3 (Controlbyte3)		
- 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
Controlbyte	Σ	135	Controlbyte	Σ	130	Controlbyte	Σ	137

Tabelle 8: Flatstream-Ermittlung der Controlbytes für Beispiel mit großen Segmenten und MultiSegmentMTU

3.8.5 Die "Forward"-Funktion am Beispiel des X2X Link

Bei der "Forward"-Funktion handelt es sich um eine Methode, die Datenrate des Flatstreams deutlich zu erhöhen. Das grundsätzliche Prinzip wird auch in anderen technischen Bereichen angewandt, z. B. beim "Pipelining" für Mikroprozessoren.

3.8.5.1 Das Funktionsprinzip

Bei der Kommunikation mittels X2X Link werden 5 Teilschritte durchlaufen, um eine Flatstream-Sequenz zu übertragen. Eine erfolgreiche Sequenzübertragung benötigt deshalb mindestens 5 Buszyklen.

	Schritt I	Schritt II	Schritt III	Schritt IV	Schritt V
Aktionen	Sequenz aus Sendearray übertragen, SequenceCounter erhöhen	Zyklischer Abgleich MTU und Modulpuffer	Sequenz an Empfangsarray fügen, SequenceAck anpassen	Zyklischer Abgleich MTU und Modulpuffer	Prüfung des SequenceAck
Ressource	Sender (Task zum Versenden)	Bussystem (Richtung 1)	Empfänger (Task zum Empfangen)	Bussystem (Richtung 2)	Sender (Task zur Ack-Prüfung)

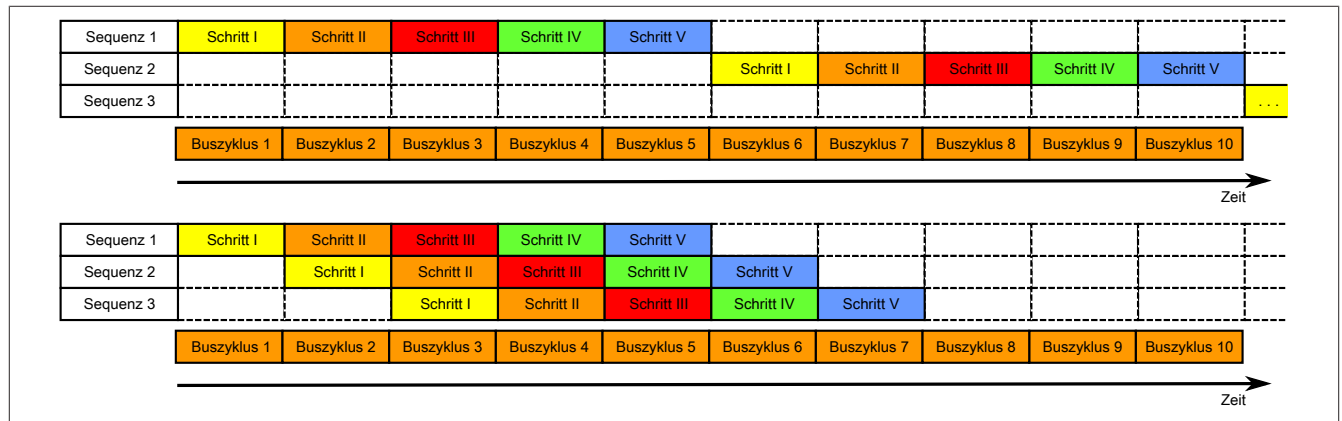


Abbildung 16: Vergleich Übertragung ohne bzw. mit Forward

Jeder der 5 Schritte (Tasks) beansprucht unterschiedliche Ressourcen. Ohne die Verwendung des Forward werden die Sequenzen nacheinander abgearbeitet. Jede Ressource ist nur dann aktiv, wenn sie für die aktuelle Teilaktion benötigt wird.

Beim Forward kann die Ressource, welche ihre Aufgabe abgearbeitet hat, bereits für die nächste Nachricht genutzt werden. Dazu wird die Bedingung zur MTU-Freigabe verändert. Die Sequenzen werden zeitgesteuert auf die MTU gelegt. Die Sendestation wartet nicht mehr auf die Bestätigung durch das SequenceAck und nutzt auf diese Weise die gegebene Bandbreite effizienter.

Im Idealfall arbeiten alle Ressourcen während jedes Buszyklus. Der Empfänger muss weiterhin jede erhaltene Sequenz bestätigen. Erst wenn das SequenceAck angepasst und vom Absender geprüft wurde, gilt die Sequenz als erfolgreich übertragen.

3.8.5.2 Konfiguration

Die Forward-Funktion muss nur für die Input-Richtung freigeschaltet werden. Die Flatstream-Module wurden dahingehend optimiert, diese Funktion unterstützen zu können. In Output-Richtung kann die Forward-Funktion genutzt werden, sobald die Größe der OutputMTU vorgegeben ist.



Information:

Die Register sind unter **"Flatstream-Register"** auf [Seite 61](#) beschrieben.

Die Register sind in den jeweiligen Datenblättern in Abschnitt **"Die Flatstream-Kommunikation"** beschrieben.

3.8.5.2.1 Verzögerungszeit

Die Verzögerungszeit wird in μs vorgegeben. Das Modul muss nach dem Versand einer Sequenz diese Zeit abwarten, bevor es im darauf folgenden Buszyklus neue Daten in die MTU schreiben darf. Die Programmroutine zum Empfang von Sequenzen aus einem Modul kann somit auch in einer Taskklasse betrieben werden deren Zykluszeit langsamer ist als der Buszyklus.

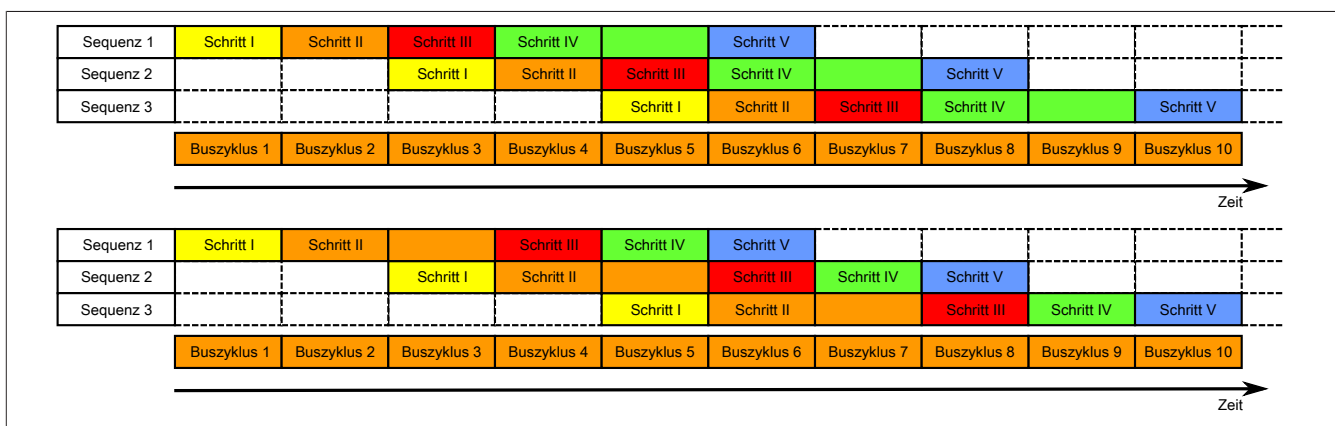


Abbildung 17: Auswirkung des ForwardDelay bei der Flatstream-Kommunikation mit Forward

Im Programmablauf muss sichergestellt werden, dass die Steuerung alle eintreffenden InputSequences bzw. InputMTUs verarbeitet. Der ForwardDelay-Wert bewirkt in Output-Richtung eine verzögerte Bestätigung und in Input-Richtung einen verzögerten Empfang. Auf diese Weise hat die Steuerung länger Zeit die eintreffende InputSequence bzw. InputMTU zu verarbeiten.

3.8.5.3 Senden und Empfangen mit Forward

Der grundsätzliche Algorithmus zum Senden bzw. Empfangen von Daten bleibt gleich. Durch den Forward können bis zu 7 unbestätigte Sequenzen abgesetzt werden. Sequenzen können gesendet werden, ohne die Bestätigung der vorangegangenen Nachricht abzuwarten. Da die Wartezeit zwischen Schreiben und Rückmeldung entfällt, können im gleichen Zeitraum erheblich mehr Daten übertragen werden.

Algorithmus zum Senden

Zyklische Statusabfrage: - Modul überwacht OutputSequenceCounter
0) Zyklische Prüfungen: - Steuerung muss OutputSyncAck prüfen → falls OutputSyncAck = 0; OutputSyncBit zurücksetzen und Kanal resynchronisieren - Steuerung muss Freigabe der OutputMTU prüfen → falls OutputSequenceCounter > OutputSequenceAck + 7, in diesem Fall nicht freigegeben, weil letzte Sequenz noch nicht quittiert
1) Vorbereitung (Sendearray anlegen): - Steuerung muss Nachricht auf zulässige Segmente aufteilen und entsprechende Controlbytes bilden - Steuerung muss Segmente und Controlbytes zu Sendearray zusammenfügen
2) Senden: - Steuerung muss aktuellen Teil des Sendearrays in die OutputMTU übertragen - Steuerung muss OutputSequenceCounter erhöhen, damit Sequenz vom Modul übernommen wird - Steuerung darf im nächsten Buszyklus erneut senden, falls MTU freigegeben ist
Reaktion des Moduls, weil OutputSequenceCounter > OutputSequenceAck: - Modul übernimmt Daten aus internem Empfangspuffer und fügt sie am Ende des internen Empfangsarrays an - Modul quittiert; aktuell empfangener Wert des OutputSequenceCounters auf OutputSequenceAck übertragen - Modul fragt Status wieder zyklisch ab
3) Abschluss (Bestätigung): - Steuerung muss OutputSequenceAck zyklisch überprüfen → Eine Sequenz gilt erst dann als erfolgreich übertragen, wenn sie über das OutputSequenceAck bestätigt wurde. Um Übertragungsfehler auch bei der letzten Sequenz zu erkennen, muss sichergestellt werden, dass der Algorithmus lange genug durchlaufen wird.
Hinweis: Für eine exakte Überwachung der Kommunikationszeiten sollten die Taskzyklen gezählt werden, die seit der letzten Erhöhung des OutputSequenceCounters vergangen sind. Auf diese Weise kann die Anzahl der Buszyklen abgeschätzt werden, die bislang zur Übertragung benötigt wurden. Übersteigt der Überwachungszähler eine vorgegebene Schwelle, kann die Sequenz als verloren betrachtet werden (das Verhältnis von Bus- und Taskzyklus kann vom Anwender beeinflusst werden, sodass der Schwellwert individuell zu ermitteln ist).

Algorithmus zum Empfangen

0) Zyklische Statusabfrage: - Steuerung muss InputSequenceCounter überwachen
Zyklische Prüfungen: - Modul prüft InputSyncAck - Modul prüft InputMTU auf Freigabe → Freigabekriterium: InputSequenceCounter > InputSequenceAck + Forward
Vorbereitung: - Modul bildet Controlbytes/Segmente und legt Sendearray an
Aktion: - Modul überträgt aktuellen Teil des Sendearrays in den Empfangspuffer - Modul erhöht InputSequenceCounter - Modul wartet auf neuen Buszyklus, nachdem Zeit aus ForwardDelay abgelaufen ist - Modul wiederholt Aktion, falls InputMTU freigegeben ist
1) Empfangen (InputSequenceCounter > InputSequenceAck): - Steuerung muss Daten aus InputMTU übernehmen und an das Ende des Empfangsarrays anfügen - Steuerung muss InputSequenceAck an InputSequenceCounter der aktuell verarbeiteten Sequenz angleichen
Abschluss: - Modul überwacht InputSequenceAck → Eine Sequenz gilt erst dann als erfolgreich übertragen, wenn sie über das InputSequenceAck bestätigt wurde.

Details/Hintergründe

1. SequenceCounter unzulässig groß (Zählerversatz)

Fehlersituation: MTU nicht freigegeben

Wenn beim Senden der Unterschied zwischen SequenceCounter und SequenceAck größer wird, als es erlaubt ist, liegt ein Übertragungsfehler vor. In diesem Fall müssen alle unbestätigten Sequenzen mit dem alten Wert des SequenceCounters wiederholt werden.

2. Prüfung einer Bestätigung

Nach dem Empfang einer Bestätigung muss geprüft werden, ob die bestätigte Sequenz abgesendet wurde und bisher unbestätigt war. Falls eine Sequenz mehrfach bestätigt wird, liegt ein schwerwiegender Fehler vor. Der Kanal muss geschlossen und resynchronisiert werden (gleiches Verhalten wie ohne Forward).



Information:

In Ausnahmefällen kann das Modul bei der Verwendung des Forward den OutputSequenceAck um mehr als 1 erhöhen.

In diesem Fall liegt kein Fehler vor. Die Steuerung darf alle Sequenzen bis zur Bestätigten als erfolgreich übertragen betrachten.

3. Sende- und Empfangsarrays

Der Forward beeinflusst die Struktur des Sende- und Empfangsarrays nicht. Sie werden auf dieselbe Weise gebildet bzw. müssen auf dieselbe Weise ausgewertet werden.

3.8.5.4 Fehlerfall bei Verwendung des Forward

Im industriellen Umfeld werden in der Regel viele verschiedene Geräte unterschiedlicher Hersteller nebeneinander genutzt. Technische Geräte können sich gegenseitig durch ungewollte elektrische oder elektromagnetische Effekte störend beeinflussen. Unter Laborbedingungen können diese Situationen nur bis zu einem bestimmten Punkt nachempfunden und abgesichert werden.

Für die Übertragung per X2X Link wurden Vorkehrungen getroffen, falls es zu derartigen Beeinflussungen kommen sollte. Tritt beim Datentransfer z. B. eine unzulässige Prüfsumme auf, ignoriert das I/O-System die Daten dieses Buszyklus und der Empfänger erhält die letzten gültigen Daten erneut. Bei den herkömmlichen (zyklischen) Datenpunkten kann dieser Fehler oft ignoriert werden. Im darauffolgenden Zyklus wird der gleiche Datenpunkt wieder abgerufen, angepasst und übertragen.

Bei der Flatstream-Kommunikation mit aktiviertem Forward ist die Situation komplexer. Auch hier erhält der Empfänger ein weiteres mal die alten Daten, das heißt, die vorherigen Werte für SequenceAck/SequenceCounter und die alte MTU.

Ausfall einer Bestätigung (SequenceAck)

Wenn durch den Ausfall ein SequenceAck-Wert verloren geht, wurde die MTU bereits korrekt übertragen. Aus diesem Grund darf die nächste Sequenz vom Empfänger weiterverarbeitet werden. Der SequenceAck wird wieder an den mitgelieferten SequenceCounter angepasst und zum Absender zurückgeschickt. Für die Prüfung der eingehenden Bestätigungen folgt daraus, dass alle Sequenzen bis zur zuletzt Bestätigten erfolgreich übertragen sind (siehe Bild Sequenz 1, 2).

Ausfall einer Sendung (SequenceCounter, MTU)

Wenn durch den Ausfall eines Buszyklus der SequenceCounter-Wert bzw. die befüllte MTU verloren geht, kommen beim Empfänger keine Daten an. Zu diesem Zeitpunkt wirkt sich der Fehler noch nicht auf die Routine zum Absenden aus. Die zeitgesteuerte MTU wird wieder freigegeben und kann neu beschrieben werden.

Der Empfänger erhält SequenceCounter-Werte, die mehrfach inkrementiert sind. Damit das Empfangsarray korrekt zusammengestellt wird, darf der Empfänger nur Sendungen verarbeiten, die einen um eins erhöhten SequenceCounter besitzen. Die eintreffenden Sequenzen müssen ignoriert werden, das heißt, der Empfänger stoppt und gibt keine neuen Bestätigungen zurück.

Wenn die maximale Anzahl an unbestätigten Sequenzen abgesendet wurde und keine Bestätigungen zurück kommen, muss der Sender die betroffenen SequenceCounter und die dazugehörigen MTUs wiederholen (siehe Bild Sequenzen 3 und 4).

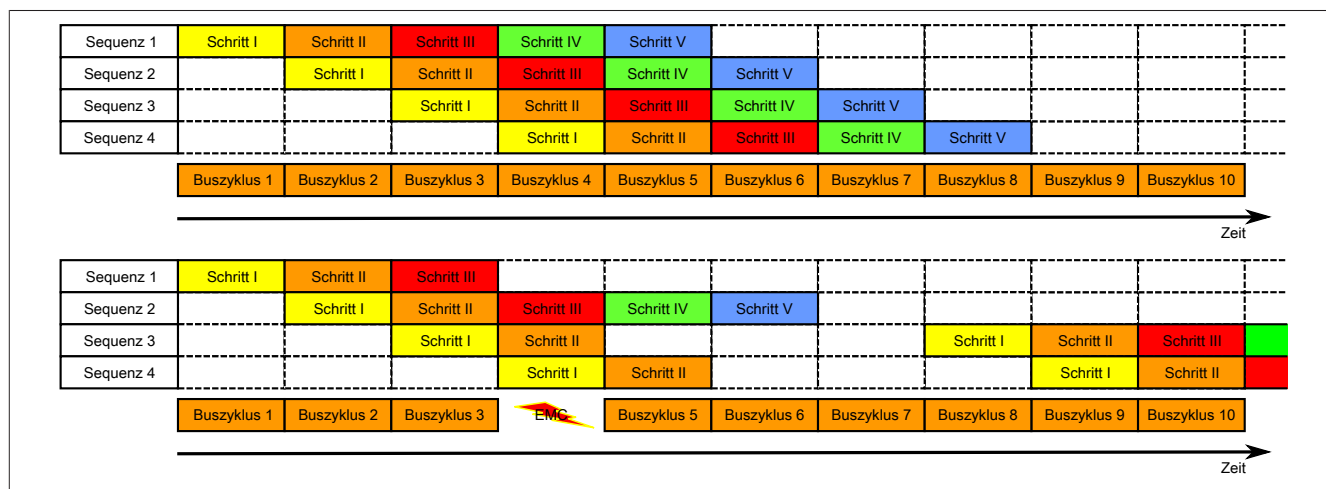


Abbildung 18: Auswirkung eines ausgefallenen Buszyklus

Ausfall der Bestätigung

Bei Sequenz 1 ging aufgrund der Störung die Bestätigung verloren. Im Schritt V der Sequenz 2 werden deshalb die Sequenzen 1 und 2 bestätigt.

Ausfall einer Sendung

Bei Sequenz 3 ging aufgrund der Störung die gesamte Sendung verloren. Der Empfänger stoppt und gibt keine Bestätigungen mehr zurück.

Der Sender sendet zunächst weiter, bis er die max. erlaubte Anzahl an unbestätigten Sendungen abgesetzt hat.

Je nach Konfiguration beginnt er frühestens 5 Buszyklen später, die vergeblich abgesendeten Sendungen zu wiederholen.

4 Inbetriebnahme

4.1 Verwendung des Moduls am Bus Controller

Das Funktionsmodell 254 "Bus Controller" wird defaultmäßig nur von nicht konfigurierbaren Bus Controllern verwendet. Alle anderen Bus Controller können, abhängig vom verwendeten Feldbus, andere Register und Funktionen verwenden.

Für Detailinformationen siehe X20 Anwenderhandbuch (ab Version 3.50), Abschnitt "Zusätzliche Informationen - Verwendung von I/O-Modulen am Bus Controller".

4.1.1 CAN-I/O Bus Controller

Das Modul belegt an CAN-I/O 4 analoge logische Steckplätze.

5 Registerbeschreibung

5.1 Allgemeine Datenpunkte

Neben den in der Registerbeschreibung beschriebenen Registern verfügt das Modul über zusätzliche allgemeine Datenpunkte. Diese sind nicht modulspezifisch, sondern enthalten allgemeine Informationen wie z. B. Seriennummer und Hardware-Variante.

Die allgemeinen Datenpunkte sind im X20 System Anwenderhandbuch, Abschnitt "Zusätzliche Informationen - Allgemeine Datenpunkte" beschrieben.

5.2 Funktionsmodell 0 - Standard

Register	Bezeichnung	Datentyp	Lesen		Schreiben	
			Zyklisch	Azyklisch	Zyklisch	Azyklisch
Modul - Konfiguration						
794	CfO_DiFilter	UINT				•
798	CfO_AnFilter	UINT				•
802	CfO_AnEnable	UINT				•
Modul - Steuerung						
129	Schaltzustand des digitalen Ausgangs	USINT			•	
	DigitalOutput01	Bit 0				
134	Zusätzliche Informationen und Datenpunkt Histogramme zurücksetzen	UINT			•	
	ClrStatistics_OperatingData	Bit 0				
	ClrStatistics_RelHumidity	Bit 1				
	ClrStatistics_Temperature	Bit 2				
	ClrStatistics_Acceleration01	Bit 8				
	ClrStatistics_Acceleration02	Bit 9				
	ClrStatistics_Acceleration03	Bit 10				
	ClrStatistics_Rotation01	Bit 11				
	ClrStatistics_Rotation02	Bit 12				
	ClrStatistics_Rotation03	Bit 13				
Modul - Kommunikation						
41	Status digitale Eingänge, digitaler Ausgang und I/O-Versorgung	USINT	•			
	DigitalInput01	Bit 0				
	DigitalInput02	Bit 1				
	StateDigitalOutput01	Bit 4				
	PowerSupply	Bit 7				
46	ModuleState01	UINT	•			
Messwerte						
2	RelHumidity	INT	•			
6	Temperature	INT	•			
10	Acceleration01	INT	•			
14	Acceleration02	INT	•			
18	Acceleration03	INT	•			
22	Rotation01	INT	•			
26	Rotation02	INT	•			
30	Rotation03	INT	•			
34	TempExt01	INT	•			
38	TempExt02	INT	•			
Zusätzliche Informationen						
4100	OnTimeConnected	UDINT	•			
4108	OnTimeDisconnected	UDINT	•			
4116	OnTimeCombined	UDINT	•			
4124	PowerCycles	UDINT	•			
4134	RelHumidityMin	INT	•			
4138	RelHumidityMax	INT	•			
4150	TemperatureMin	INT	•			
4154	TemperatureMax	INT	•			
4166 + N*16	Acceleration0NMin (Index N = 1 bis 3)	INT	•			
4170 + N*16	Acceleration0NMax (Index N = 1 bis 3)	INT	•			
4198 + N*16	Rotation0NMin (Index N = 1 bis 3)	INT	•			
4202 + N*16	Rotation0NMax (Index N = 1 bis 3)	INT	•			

Register	Bezeichnung	Datentyp	Lesen		Schreiben	
			Zyklisch	Azyklisch	Zyklisch	Azyklisch
Datenpunkt Histogramme						
4244 + N*16	RelHumHist0NEntry (Index N = 1 bis 10)	UDINT	•			
4252 + N*16	RelHumHist0NTime (Index N = 1 bis 10)	UDINT	•			
4404 + N*16	TempHist0NEntry (Index N = 1 bis 12)	UDINT	•			
4412 + N*16	TempHist0NTime (Index N = 1 bis 12)	UDINT	•			
4596 + N*16	AccHist010NEntry (Index N = 1 bis 8)	UDINT	•			
4604 + N*16	AccHist010NTime (Index N = 1 bis 8)	UDINT	•			
4724 + N*16	AccHist020NEntry (Index N = 1 bis 8)	UDINT	•			
4732 + N*16	AccHist020NTime (Index N = 1 bis 8)	UDINT	•			
4852 + N*16	AccHist030NEntry (Index N = 1 bis 8)	UDINT	•			
4860 + N*16	AccHist030NTime (Index N = 1 bis 8)	UDINT	•			
4980 + N*16	RotationHist010NEntry (Index N = 1 bis 8)	UDINT	•			
4988 + N*16	RotationHist010NTime (Index N = 1 bis 8)	UDINT	•			
5108 + N*16	RotationHist020NEntry (Index N = 1 bis 8)	UDINT	•			
5116 + N*16	RotationHist020NTime (Index N = 1 bis 8)	UDINT	•			
5236 + N*16	RotationHist030NEntry (Index N = 1 bis 8)	UDINT	•			
5244 + N*16	RotationHist030NTime (Index N = 1 bis 8)	UDINT	•			
Flatstream - Konfiguration (Zugriff auf internen Flash-Speicher)						
513	OutputMTU	USINT				•
515	InputMTU	USINT				•
517	FlatstreamMode	USINT				•
519	Forward	USINT				•
522	ForwardDelay	UINT				•
Flatstream - Kommunikation (Zugriff auf internen Flash-Speicher)						
577	InputSequence	USINT	•			
577 + N*2	RxByteN (Index N = 1 bis 27)	USINT	•			
641	OutputSequence	USINT			•	
641 + N*2	TxByteN (Index N = 1 bis 15)	USINT			•	

5.3 Funktionsmodell 254 - Bus Controller

Register	Offset ¹⁾	Bezeichnung	Datentyp	Lesen		Schreiben	
				Zyklisch	Azyklisch	Zyklisch	Azyklisch
Modul - Konfiguration							
794	-	CfO_DiFilter	UINT				•
798	-	CfO_AnFilter	UINT				•
802	-	CfO_AnEnable	UINT				•
Modul - Steuerung							
129	2	Schaltzustand des digitalen Ausgangs	USINT			•	
		DigitalOutput01	Bit 0				
134	-	Zusätzliche Informationen und Datenpunkt Histogramme zurücksetzen	UINT				•
		ClrStatistics_OperatingData	Bit 0				
		ClrStatistics_RelHumidity	Bit 1				
		ClrStatistics_Temperature	Bit 2				
		ClrStatistics_Acceleration01	Bit 8				
		ClrStatistics_Acceleration02	Bit 9				
		ClrStatistics_Acceleration03	Bit 10				
		ClrStatistics_Rotation01	Bit 11				
		ClrStatistics_Rotation02	Bit 12				
		ClrStatistics_Rotation03	Bit 13				
Modul - Kommunikation							
41	20	Status digitale Eingänge, digitaler Ausgang und I/O-Versorgung	USINT	•			
		DigitalInput01	Bit 0				
		DigitalInput02	Bit 1				
		StateDigitalOutput01	Bit 4				
		PowerSupply	Bit 7				
46	22	ModuleState01	UINT	•			
Messwerte							
2	0	RelHumidity	INT	•			
6	2	Temperature	INT	•			
10	4	Acceleration01	INT	•			
14	6	Acceleration02	INT	•			
18	8	Acceleration03	INT	•			
22	10	Rotation01	INT	•			
26	12	Rotation02	INT	•			
30	14	Rotation03	INT	•			
34	16	TempExt01	INT	•			
38	18	TempExt02	INT	•			

Registerbeschreibung

Register	Offset ¹⁾	Bezeichnung	Datentyp	Lesen		Schreiben	
				Zyklisch	Azyklisch	Zyklisch	Azyklisch
Zusätzliche Informationen							
4100	-	OnTimeConnected	UDINT		•		
4108	-	OnTimeDisconnected	UDINT		•		
4116	-	OnTimeCombined	UDINT		•		
4124	-	PowerCycles	UDINT		•		
4134	-	RelHumidityMin	INT		•		
4138	-	RelHumidityMax	INT		•		
4150	-	TemperatureMin	INT		•		
4154	-	TemperatureMax	INT		•		
4166 + N*16	-	Acceleration0NMin (Index N = 1 bis 3)	INT		•		
4170 + N*16	-	Acceleration0NMax (Index N = 1 bis 3)	INT		•		
4198 + N*16	-	Rotation0NMin (Index N = 1 bis 3)	INT		•		
4202 + N*16	-	Rotation0NMax (Index N = 1 bis 3)	INT		•		
Datenpunkt Histogramme							
4244 + N*16	-	RelHumHist0NEntry (Index N = 1 bis 10)	UDINT		•		
4252 + N*16	-	RelHumHist0NTime (Index N = 1 bis 10)	UDINT		•		
4404 + N*16	-	TempHist0NEntry (Index N = 1 bis 12)	UDINT		•		
4412 + N*16	-	TempHist0NTime (Index N = 1 bis 12)	UDINT		•		
4596 + N*16	-	AccHist010NEntry (Index N = 1 bis 8)	UDINT		•		
4604 + N*16	-	AccHist010NTime (Index N = 1 bis 8)	UDINT		•		
4724 + N*16	-	AccHist020NEntry (Index N = 1 bis 8)	UDINT		•		
4732 + N*16	-	AccHist020NTime (Index N = 1 bis 8)	UDINT		•		
4852 + N*16	-	AccHist030NEntry (Index N = 1 bis 8)	UDINT		•		
4860 + N*16	-	AccHist030NTime (Index N = 1 bis 8)	UDINT		•		
4980 + N*16	-	RotationHist010NEntry (Index N = 1 bis 8)	UDINT		•		
4988 + N*16	-	RotationHist010NTime (Index N = 1 bis 8)	UDINT		•		
5108 + N*16	-	RotationHist020NEntry (Index N = 1 bis 8)	UDINT		•		
5116 + N*16	-	RotationHist020NTime (Index N = 1 bis 8)	UDINT		•		
5236 + N*16	-	RotationHist030NEntry (Index N = 1 bis 8)	UDINT		•		
5244 + N*16	-	RotationHist030NTime (Index N = 1 bis 8)	UDINT		•		

1) Der Offset gibt an, wo das Register im CAN-Objekt angeordnet ist.

5.4 Konfiguration

5.4.1 Digitale Eingangsfilter

Name:

CfO_DiFilter

In diesem Register kann der Filterwert für alle digitalen Eingänge parametrierbar werden.

Datentyp	Werte	Filter
USINT	0	Kein Softwarefilter (Bus Controller Default)
	2	0,2 ms

	250	25 ms - höhere Werte werden auf diesen Wert begrenzt

5.4.2 Filterstufe und Eingangsrampenbegrenzung

Name:
CfO_AnFilter

In diesem Register werden die Filterstufe und die Eingangsrampenbegrenzung des Eingangsfilters der Analogeingänge eingestellt.

Datentyp	Werte
UINT	Siehe Bitstruktur

Bitstruktur:

Bit	Beschreibung	Wert	Information
0 - 7	Filterstufe	0	Filter ausgeschaltet
		1	1 Messung
		2	2 Messungen
		3	3 Messungen
		:	:
		100	100 Messungen: Default
		:	:
8 - 10	Eingangsrampenbegrenzung	255	255 Messungen
		000	Der Eingangswert wird ohne Begrenzung übernommen
		001	Grenzwert = 0x3FFF (16383)
		010	Grenzwert = 0x1FFF (8191)
		011	Grenzwert = 0x0FFF (4095)
		100	Grenzwert = 0x07FF (2047)
		101	Grenzwert = 0x03FF (1023)
		110	Grenzwert = 0x01FF (511)
11 - 15	Reserviert	111	Grenzwert = 0x00FF (255): Default
		0	

5.4.3 Analogeingang aktivieren

Name:
CfO_AnEnable

Mit diesem Register werden die Analogeingänge aktiviert.

Datentyp	Werte
UINT	Siehe Bitstruktur

Bitstruktur:

Bit	Bezeichnung	Wert	Information
0	Kanal 1	0	Nicht aktiviert
		1	Aktiviert
1 - 7	Reserviert	0	
8	Kanal 2	0	Nicht aktiviert
		1	Aktiviert
9 - 15	Reserviert	0	

5.5 Steuerung

5.5.1 Schaltzustand des digitalen Ausgangs

Name:

DigitalOutput01

Mit diesem Register wird der digitale Ausgang gesteuert.

Datentyp	Werte
USINT	Siehe Bitstruktur

Bitstruktur:

Bit	Bezeichnung	Wert	Information
0	DigitalOutput01	0	Ausgang rücksetzen
		1	Ausgang setzen
1 - 7	Reserviert	0	

5.5.2 Zusätzliche Informationen und Datenpunkt Histogramme zurücksetzen

Name:

ClrStatistics_OperatingData

ClrStatistics_RelHumidity

ClrStatistics_Temperature

ClrStatistics_Acceleration01 bis ClrStatistics_Acceleration03

ClrStatistics_Rotation01 bis ClrStatistics_Rotation03

Durch Setzen des jeweiligen Bits im Register werden Betriebsdaten, Informationen und Histogramme zurückgesetzt. Vorgehensweise:

- Das Bit zum Rücksetzen der gewünschten Daten setzen
- Das Bit muss so lange gesetzt bleiben, bis die Register zurückgesetzt wurden
- Sobald der Anwender festgestellt hat, dass die Daten zurückgesetzt wurden, kann das Bit zum Zurücksetzen der Daten gelöscht werden
- Wenn das Bit zum Rücksetzen der Daten nicht gelöscht wird, werden die Daten dauerhaft auf 0 gesetzt



Information:

Es kann bis zu 1 s dauern, bis der Löschvorgang der Daten ausgeführt wird.

Datentyp	Werte
UINT	Siehe Bitstruktur

Bitstruktur:

Bit	Bezeichnung	Wert	Information
0	ClrStatistics_OperatingData Betriebsdaten zurücksetzen	0	Nicht zurücksetzen
		1	Zurücksetzen
1	ClrStatistics_RelHumidity Informationen und Histogramme für relative Luftfeuchtigkeit zurücksetzen	0	Nicht zurücksetzen
		1	Zurücksetzen
2	ClrStatistics_Temperature Informationen und Histogramme für Umgebungstemperatur zurücksetzen	0	Nicht zurücksetzen
		1	Zurücksetzen
3 - 7	Reserviert	0	
8	ClrStatistics_Acceleration01 Informationen und Histogramme für Acceleration01 zurücksetzen	0	Nicht zurücksetzen
		1	Zurücksetzen
9	ClrStatistics_Acceleration02 Informationen und Histogramme für Acceleration02 zurücksetzen	0	Nicht zurücksetzen
		1	Zurücksetzen
10	ClrStatistics_Acceleration03 Informationen und Histogramme für Acceleration03 zurücksetzen	0	Nicht zurücksetzen
		1	Zurücksetzen
11	ClrStatistics_Rotation01 Informationen und Histogramme für Rotation01 zurücksetzen	0	Nicht zurücksetzen
		1	Zurücksetzen
12	ClrStatistics_Rotation02 Informationen und Histogramme für Rotation02 zurücksetzen	0	Nicht zurücksetzen
		1	Zurücksetzen
13	ClrStatistics_Rotation03 Informationen und Histogramme für Rotation03 zurücksetzen	0	Nicht zurücksetzen
		1	Zurücksetzen
14 - 15	Reserviert	0	

5.6 Kommunikation

5.6.1 Status digitale Eingänge, digitaler Ausgang und I/O-Versorgung

Name:

DigitalInput01

DigitalInput02

StateDigitalOutput01

PowerSupply

In diesem Register wird der Zustand der digitalen Eingänge, des digitalen Ausgangs und der I/O-Versorgungsspannung angezeigt.

Datentyp	Werte
USINT	Siehe Bitstruktur

Bitstruktur:

Bit	Bezeichnung	Wert	Information
0	DigitalInput01	0 oder 1	Eingangszustand Digitaleingang 1
1	DigitalInput02	0 oder 1	Eingangszustand Digitaleingang 2
2 - 3	Reserviert	-	
4	StateDigitalOutput01	0 oder 1	Ausgangszustand Digitalausgang 1
5 - 6	Reserviert	-	
7	PowerSupply Zustand der I/O-Versorgungsspannung	0	I/O-Versorgung im erlaubten Bereich: 24 VDC -15% / +20%
		1	I/O-Versorgung außerhalb vom erlaubten Bereich

5.6.2 Modulstatus

Name:

ModuleState01

Statusregister zur Überwachung der analogen Eingänge.

Datentyp	Werte
UINT	Siehe Bitstruktur

Bitstruktur:

Bit	Bezeichnung	Wert	Information
0 - 1	Kanal 1	00	Kein Fehler
		01	Bereichsunterschreitung
		10	Bereichsüberschreitung
		11	Drahtbruch
2 - 3	Kanal 2	00	Kein Fehler
		01	Bereichsunterschreitung
		10	Bereichsüberschreitung
		11	Drahtbruch
4 - 7	Reserviert	-	

5.7 Messwerte

5.7.1 Relative Luftfeuchtigkeit

Name:

RelHumidity

Ein interner Sensor misst die relative Luftfeuchtigkeit in der Umgebung.

Datentyp	Werte	Information
INT	0 bis 100	Relative Luftfeuchtigkeit [%], Auflösung 1%

5.7.2 Umgebungstemperatur

Name:

Temperature

Ein interner Sensor misst die Umgebungstemperatur.

Datentyp	Werte	Information
INT	-250 bis 1250	Umgebungstemperatur [°C], Auflösung 0,1°C

5.7.3 Beschleunigung

Name:

Acceleration01 bis Acceleration03

Ein interner Sensor misst die Beschleunigung.

Datentyp	Werte	Information
INT	-32768 bis 32767	Beschleunigung als Rohwert. Die Umrechnung ist in der Applikation durchzuführen: <ul style="list-style-type: none"> 16 g = 32767 -16 g = -32768

5.7.4 Rotation

Name:

Rotation01 bis Rotation03

Ein interner Sensor misst die Rotation.

Datentyp	Werte	Information
INT	-32768 bis 32767	Rotation als Rohwert. Die Umrechnung ist in der Applikation durchzuführen: <ul style="list-style-type: none"> 2000 dps = 32767 -2000 dps = -32768

5.7.5 Analoge Eingänge

Name:

TempExt01 bis TempExt02

In diesem Register werden die analogen Eingangswerte abgebildet. Der Eingangsfilter und die Eingangsrampenbegrenzung wird mit dem Register "[CfO_AnFilter](#)" auf [Seite 53](#) eingestellt.

Datentyp	Digitaler Wert	Eingangssignal
INT	-400 bis 1250 (Auflösung 0,1°C)	Fühlertyp PT1000, Temperaturmessung -40,0 bis 125,0°C

5.8 Zusätzliche Informationen



Information:

Folgende Punkte sind zu beachten:

- Das Speichern der vom Modul erfassten Daten erfolgt in einem Raster von 10 s.
- Beim Rücksetzen der Werte kann es bis zu 1 s dauern, bis der Löschvorgang ausgeführt wird (siehe Register "[ClrStatistics](#)" auf Seite 54).

5.8.1 Betriebsdaten

Name:

OnTimeConnected
OnTimeDisconnected
OnTimeCombined
PowerCycles

In diesen Registern werden die jeweiligen Betriebsdaten ausgegeben. Bei Bedarf können die Werte mit dem Register "[ClrStatistics](#)" auf Seite 54 zurückgesetzt werden.

Register	Datentyp	Werte	Information
OnTimeConnected	UDINT	0 bis 4.294.967.295	Laufzeit, während der das Modul aktiv mit dem Netzwerkmaster verbunden war [s], Auflösung 1 s
OnTimeDisconnected	UDINT	0 bis 4.294.967.295	Laufzeit, während der das Modul nicht aktiv mit dem Netzwerkmaster verbunden war [s] (Blackout-Modus), Auflösung 1 s
OnTimeCombined	UDINT	0 bis 4.294.967.295	Gesamte Laufzeit des Moduls [s], Auflösung 1 s
PowerCycles	UDINT	0 bis 4.294.967.295	Anzahl der Einschaltzyklen

5.8.2 Relative Luftfeuchtigkeit

Name:

RelHumidityMin
RelHumidityMax

In diesen Registern werden Informationen zur relativen Luftfeuchtigkeit ausgegeben. Der Abtastintervall beträgt 1 s. Bei Bedarf können die Werte mit dem Register "[ClrStatistics](#)" auf Seite 54 zurückgesetzt werden.

Register	Datentyp	Werte	Information
RelHumidityMin	INT	0 bis 100	Kleinsten aufgetretener Wert [%], Auflösung 1%
RelHumidityMax	INT	0 bis 100	Größter aufgetretener Wert [%], Auflösung 1%

5.8.3 Umgebungstemperatur

Name:

TemperatureMin
TemperatureMax

In diesen Registern werden Informationen zur Umgebungstemperatur ausgegeben. Der Abtastintervall beträgt 1 s. Bei Bedarf können die Werte mit dem Register "[ClrStatistics](#)" auf Seite 54 zurückgesetzt werden.

Register	Datentyp	Werte	Information
TemperatureMin	INT	-250 bis 1250	Kleinsten aufgetretener Wert [°C], Auflösung 0,1°C
TemperatureMax	INT	-250 bis 1250	Größter aufgetretener Wert [°C], Auflösung 0,1°C

5.8.4 Beschleunigung

Name:

Acceleration01Min bis Acceleration03Min

Acceleration01Max bis Acceleration03Max

In diesen Registern werden Informationen zur Beschleunigung ausgegeben. Der Abtastintervall beträgt typ. 10 ms. Bei Bedarf können die Werte mit dem Register "[ClrStatistics](#)" auf [Seite 54](#) zurückgesetzt werden.

Die Beschleunigung wird als Rohwert geliefert. Die Umrechnung ist in der Applikation durchzuführen.

Register	Datentyp	Werte	Information
Acceleration01Min bis Acceleration03Min	INT	-32768 bis 32767	Kleinster aufgetretener Wert
Acceleration01Max bis Acceleration03Max	INT	-32768 bis 32767	Größter aufgetretener Wert

5.8.5 Rotation

Name:

Rotation01Min bis Rotation03Min

Rotation01Max bis Rotation03Max

In diesen Registern werden Informationen zur Rotation ausgegeben. Der Abtastintervall beträgt typ. 10 ms. Bei Bedarf können die Werte mit dem Register "[ClrStatistics](#)" auf [Seite 54](#) zurückgesetzt werden.

Die Rotation wird als Rohwert geliefert. Die Umrechnung ist in der Applikation durchzuführen.

Register	Datentyp	Werte	Information
Rotation01Min bis Rotation03Min	INT	-32768 bis 32767	Kleinster aufgetretener Wert
Rotation01Max bis Rotation03Max	INT	-32768 bis 32767	Größter aufgetretener Wert

5.9 Datenpunkt Histogramme

5.9.1 Relative Luftfeuchtigkeit

Name:

RelHumHist01Entry bis RelHumHist10Entry

RelHumHist01Time bis RelHumHist10Time

In diesen Registern werden die vom Modul aufgezeichneten Histogrammdaten für die relative Luftfeuchtigkeit ausgegeben. Bei Bedarf können die Werte mit dem Register "[ClrStatistics](#)" auf Seite 54 zurückgesetzt werden.

Register	Datentyp	Werte	Information
RelHumHist01Entry bis RelHumHist10Entry	UDINT	0 bis 4.294.967.295	Eintrittszähler für den entsprechenden Bereich
RelHumHist01Time bis RelHumHist10Time	UDINT	0 bis 4.294.967.295	Verweilzeit im entsprechenden Bereich [s], Auflösung 1 s

5.9.2 Umgebungstemperatur

Name:

TempHist01Entry bis TempHist12Entry

TempHist01Time bis TempHist12Time

In diesen Registern werden die vom Modul aufgezeichneten Histogrammdaten für die Umgebungstemperatur ausgegeben. Bei Bedarf können die Werte mit dem Register "[ClrStatistics](#)" auf Seite 54 zurückgesetzt werden.

Register	Datentyp	Werte	Information
TempHist01Entry bis TempHist12Entry	UDINT	0 bis 4.294.967.295	Eintrittszähler für den entsprechenden Bereich
TempHist01Time bis TempHist12Time	UDINT	0 bis 4.294.967.295	Verweilzeit im entsprechenden Bereich [s], Auflösung 1 s

5.9.3 Beschleunigung

Name:

AccHist0101Entry bis AccHist0108Entry

AccHist0101Time bis AccHist0108Time

AccHist0201Entry bis AccHist0208Entry

AccHist0201Time bis AccHist0208Time

AccHist0301Entry bis AccHist0308Entry

AccHist0301Time bis AccHist0308Time

In diesen Registern werden die vom Modul aufgezeichneten Histogrammdaten für die Beschleunigung ausgegeben. Bei Bedarf können die Werte mit dem Register "[ClrStatistics](#)" auf Seite 54 zurückgesetzt werden.

Register	Datentyp	Werte	Information
AccHist0N01Entry bis AccHist0N08Entry	UDINT	0 bis 4.294.967.295	Eintrittszähler für den entsprechenden Bereich
AccHist0N01Time bis AccHist0N08Time	UDINT	0 bis 4.294.967.295	Verweilzeit im entsprechenden Bereich [s], Auflösung 1 s

Legende: N = 1 bis 3

5.9.4 Rotation

Name:

RotationHist0101Entry bis RotationHist0108Entry
 RotationHist0101Time bis RotationHist0108Time
 RotationHist0201Entry bis RotationHist0208Entry
 RotationHist0201Time bis RotationHist0208Time
 RotationHist0301Entry bis RotationHist0308Entry
 RotationHist0301Time bis RotationHist0308Time

In diesen Registern werden die vom Modul aufgezeichneten Histogrammdaten für die Rotation ausgegeben. Bei Bedarf können die Werte mit dem Register "[ClrStatistics](#)" auf Seite 54 zurückgesetzt werden.

Register	Datentyp	Werte	Information
RotationHist0N01Entry bis RotationHist0N08Entry	UDINT	0 bis 4.294.967.295	Eintrittszähler für den entsprechenden Bereich
RotationHist0N01Time bis RotationHist0N08Time	UDINT	0 bis 4.294.967.295	Verweilzeit im entsprechenden Bereich [s], Auflösung 1 s

Legende: N = 1 bis 3

5.10 Flatstream-Register

Bei der Minimalkonfiguration müssen die Register "InputMTU" und "OutputMTU" eingestellt werden. Alle anderen Register werden beim Start mit Standardwerten belegt und können sofort genutzt werden. Sie stellen zusätzliche Optionen bereit, um Daten kompakter zu übertragen bzw. den allgemeinen Ablauf hoch effizient zu gestalten.



Information:

Für detaillierte Informationen zum Flatstream siehe [Die Flatstream-Kommunikation](#).

5.10.1 Anzahl der aktivierten Tx- bzw. Rx-Bytes

Name:

OutputMTU
 InputMTU

Diese Register definieren die Anzahl der aktivierten Tx- bzw. Rx-Bytes und somit auch die maximale Größe einer Sequenz. Der Anwender muss beachten, dass mehr freigegebene Bytes auch eine stärkere Belastung für das Bussystem bedeuten.

Datentyp	Werte
USINT	Siehe Registerübersicht

5.10.2 Transport der Nutzdaten und der Controlbytes

Name:

TxByte1 bis TxByteN
 RxByte1 bis RxByteN

(Die Größe der Zahl N ist je nach verwendetem Bus Controller Modell unterschiedlich.)

Die Tx- bzw. Rx-Bytes sind zyklische Register, die zum Transport der Nutzdaten und der notwendigen Controlbytes dienen. Die Anzahl aktiver Tx- bzw. Rx-Bytes ergibt sich aus der Konfiguration der Register "[OutputMTU](#)" bzw. "[InputMTU](#)".

- "T" - "transmit" → Steuerung sendet Daten an das Modul
- "R" - "receive" → Steuerung empfängt Daten vom Modul

Datentyp	Werte
USINT	0 bis 255

5.10.3 Kommunikationsstatus der Steuerung

Name:

OutputSequence

Dieses Register enthält Informationen über den Kommunikationsstatus der Steuerung. Es wird von der Steuerung geschrieben und vom Modul gelesen.

Datentyp	Werte
USINT	Siehe Bitstruktur

Bitstruktur:

Bit	Bezeichnung	Wert	Information
0 - 2	OutputSequenceCounter	0 - 7	Zähler der in Output abgesetzten Sequenzen
3	OutputSyncBit	0	Output-Richtung deaktiviert (disable)
		1	Output-Richtung aktiviert (enable)
4 - 6	InputSequenceAck	0 - 7	Spiegel des InputSequenceCounters
7	InputSyncAck	0	Input-Richtung nicht bereit (disable)
		1	Input-Richtung bereit (enable)

OutputSequenceCounter

Der OutputSequenceCounter ist ein umlaufender Zähler der Sequenzen, die von der Steuerung abgeschickt wurden. Über den OutputSequenceCounter weist die Steuerung das Modul an, eine Sequenz zu übernehmen (zu diesem Zeitpunkt muss die Output-Richtung synchronisiert sein).

OutputSyncBit

Mit dem OutputSyncBit versucht die Steuerung den Output-Kanal zu synchronisieren.

InputSequenceAck

Der InputSequenceAck dient zur Bestätigung. Der Wert des InputSequenceCounters wird darin gespiegelt, wenn die Steuerung eine Sequenz erfolgreich empfangen hat.

InputSyncAck

Das Bit InputSyncAck bestätigt dem Modul die Synchronität des Input-Kanals. Die Steuerung zeigt damit an, dass sie bereit ist, Daten zu empfangen.

5.10.4 Kommunikationsstatus des Moduls

Name:

InputSequence

Dieses Register enthält Informationen über den Kommunikationsstatus des Moduls. Es wird vom Modul geschrieben und sollte von der Steuerung nur gelesen werden.

Datentyp	Werte
USINT	Siehe Bitstruktur

Bitstruktur:

Bit	Bezeichnung	Wert	Information
0 - 2	InputSequenceCounter	0 - 7	Zähler der in Input abgesetzten Sequenzen
3	InputSyncBit	0	Nicht bereit (disable)
		1	Bereit (enable)
4 - 6	OutputSequenceAck	0 - 7	Spiegel des OutputSequenceCounters
7	OutputSyncAck	0	Nicht bereit (disable)
		1	Bereit (enable)

InputSequenceCounter

Der InputSequenceCounter ist ein umlaufender Zähler der Sequenzen, die vom Modul abgeschickt wurden. Über den InputSequenceCounter weist das Modul die Steuerung an, eine Sequenz zu übernehmen (zu diesem Zeitpunkt muss die Input-Richtung synchronisiert sein).

InputSyncBit

Mit dem InputSyncBit versucht das Modul den Input-Kanal zu synchronisieren.

OutputSequenceAck

Der OutputSequenceAck dient zur Bestätigung. Der Wert des OutputSequenceCounters wird darin gespiegelt, wenn das Modul eine Sequenz erfolgreich empfangen hat.

OutputSyncAck

Das Bit OutputSyncAck bestätigt der Steuerung die Synchronität des Output-Kanals. Das Modul zeigt damit an, dass es bereit ist, Daten zu empfangen.

5.10.5 Flatstream Modus

Name:

FlatstreamMode

Mit Hilfe dieses Registers kann eine kompaktere Anordnung beim eintreffenden Datenstrom erreicht werden.

Datentyp	Werte
USINT	Siehe Bitstruktur

Bitstruktur:

Bit	Bezeichnung	Wert	Information
0	MultiSegmentMTU	0	Nicht erlaubt (Standard)
		1	Erlaubt
1	Große Segmente	0	Nicht erlaubt (Standard)
		1	Erlaubt
2 - 7	Reserviert		

5.10.6 Anzahl der unbestätigten Sequenzen

Name:

Forward

Über das Register "Forward" stellt der Anwender ein, wie viele unbestätigte Sequenzen das Modul abschicken darf.

Empfehlung:

X2X Link: max. 5

POWERLINK: max. 7

Datentyp	Werte
USINT	1 bis 7 Standard: 1

5.10.7 Verzögerungszeit

Name:

ForwardDelay

Mit diesem Register wird die Verzögerungszeit in μs vorgegeben.

Datentyp	Werte
UINT	0 bis 65535 [μs] Standard: 0

5.11 Minimale Zykluszeit

Die minimale Zykluszeit gibt an, bis zu welcher Zeit der Buszyklus heruntergefahren werden kann, ohne dass Kommunikationsfehler auftreten. Es ist zu beachten, dass durch sehr schnelle Zyklen die Restzeit zur Behandlung der Überwachungen, Diagnosen und azyklischen Befehle verringert wird.

Minimale Zykluszeit
200 μs

5.12 Minimale IO-Updatezeit

Die minimale I/O-Updatezeit gibt an, bis zu welcher Zeit der Buszyklus heruntergefahren werden kann, so dass in jedem Zyklus ein I/O-Update erfolgt.

Minimale IO-Updatezeit	
Temperatur und relative Luftfeuchtigkeit	1 s
Beschleunigung und Rotation	typ. 10 ms
Digitale Eingänge	100 μs ohne Filterung 200 μs mit Filterung
Digitaler Ausgang	Entspricht der minimalen Zykluszeit
Analoge Eingänge	1 ms
Zustand der I/O-Versorgungsspannung	<10 ms
Anwenderflash Flatstream-Kommunikation	<10 ms