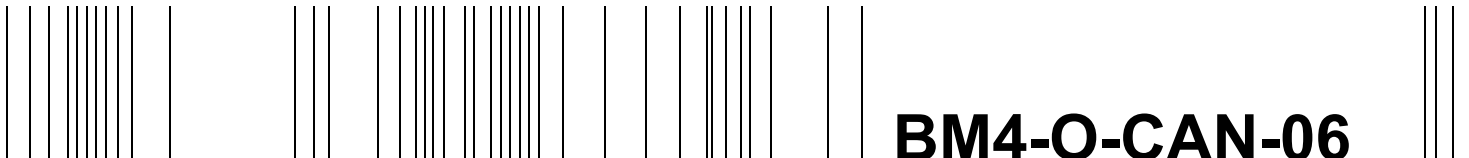


**be in motion be in motion**



**BM4-O-CAN-06**


**Optionsmodul CANsync für  
b maXX PLC  
Applikationshandbuch**

<b>D</b>	5.02066.01a
----------	-------------



Titel	Applikationshandbuch
Produkt	Optionsmodul CANsync für b maXX PLC BM4-O-CAN-06
Stand	28. Mai 2003
Copyright	<p>Dieses Applikationshandbuch darf vom Eigentümer ausschließlich für den internen Gebrauch in beliebiger Anzahl kopiert werden. Für andere Zwecke darf dieses Applikationshandbuch auch auszugsweise weder kopiert noch vervielfältigt werden.</p> <p>Verwertung und Mitteilung von Inhalten dieses Applikationshandbuches sind nicht gestattet.</p> <p>Bezeichnungen bzw. Unternehmenskennzeichen in diesem Applikationshandbuch können Marken sein, deren Benutzung durch Dritte für deren Zwecke die Rechte der Inhaber verletzen kann.</p>
Verbindlichkeit	<p>Dieses Applikationshandbuch ist Teil des Gerätes/der Maschine. Dieses Applikationshandbuch muss jederzeit für den Bediener zugänglich und in einem leserlichen Zustand sein. Bei Verkauf/Verlagerung des Gerätes/der Maschine muss dieses Applikationshandbuch vom Besitzer zusammen mit dem Gerät/der Maschine weitergegeben werden. Nach Verkauf des Gerätes/der Maschine sind dieses Original und sämtliche Kopien an den Käufer zu übergeben. Nach Entsorgung oder anderem Nutzungsende sind dieses Original und sämtliche Kopien zu vernichten.</p> <p>Mit der Übergabe des vorliegenden Applikationshandbuches werden entsprechende Applikationshandbücher mit einem früheren Stand außer Kraft gesetzt.</p> <p>Bitte beachten Sie, dass Angaben/Zahlen/Informationen <b>aktuelle Werte zum Druckdatum</b> sind. Zur Ausmessung, Berechnung und Kalkulationen sind diese Angaben <b>nicht rechtlich verbindlich</b>.</p> <p>Die Firma Baumüller Nürnberg Electronic GmbH &amp; Co. KG behält sich vor, im Rahmen der eigenen Weiterentwicklung der Produkte die technischen Daten und die Handhabung von Baumüller-Produkten zu ändern.</p> <p>Es kann jedoch keine Gewährleistung bezüglich der Fehlerfreiheit dieses Applikationshandbuches, soweit nicht in den Allgemeinen Verkaufs- und Lieferbedingungen anders beschrieben, übernommen werden.</p>
Hersteller	<p>Baumüller Nürnberg Electronic GmbH &amp; Co. KG Ostendstr. 80 - 90 90482 Nürnberg Deutschland Tel. +49 9 11 54 32 - 0 Fax: +49 9 11 54 32 - 1 30 www.baumueller.de</p>

---



Hiermit zeigen wir an, dass die Baumüller Nürnberg Electronic GmbH & Co. KG im Wege der Verschmelzung mit Wirkung zum 01.12.2004 in der Baumüller Nürnberg GmbH aufgegangen ist. Ihr zukünftiger Ansprechpartner ist damit die

**Baumüller Nürnberg GmbH, Ostendstrasse 80-90, 90482 Nürnberg.**

Please notify that with effect from 01-12-2004 Baumüller Nürnberg Electronic GmbH & Co. KG merged with Baumüller Nürnberg GmbH. Your future business partner will be

**Baumüller Nürnberg GmbH, Ostendstrasse 80-90, 90482 Nürnberg.**

Par la présente, nous vous signalons qu'en voie de la fusion, la Baumüller Nürnberg Electronic GmbH & Co. KG a été intégrée à la Baumüller Nürnberg GmbH avec effet au 1 décembre 2004. Votre interlocuteur sera par conséquent la

**Baumüller Nürnberg GmbH, Ostendstrasse 80 – 90, 90482 Nürnberg.**





<b>1</b>	<b>Einleitung</b>	<b>5</b>
1.1	Erste Schritte	5
1.2	Verwendete Begriffe	5
1.3	Voraussetzungen	6
<b>2</b>	<b>Grundlegende Sicherheitshinweise</b>	<b>7</b>
2.1	Gefahrenhinweise und Gebote	7
2.1.1	Struktur eines Gefahrenhinweises	8
2.1.2	Verwendete Gefahrenhinweise	9
2.1.2.1	Gefahrenhinweise vor Personenschaden	9
2.1.2.2	Gefahrenhinweise vor Sachschaden	10
2.1.2.3	Verwendete Gebotszeichen	10
2.2	Infozeichen	11
2.3	Rechtliche Hinweise	11
2.4	Bestimmungsgemäße Verwendung	11
2.5	Sachwidrige Verwendung	12
2.6	Schutzeinrichtungen	13
2.7	Ausbildung des Personals	13
2.8	Sicherheitsmaßnahmen im Normalbetrieb	13
2.9	Verpflichtung und Haftung	14
2.9.1	Gefahrenhinweise und Sicherheitshinweise beachten	14
2.9.2	Gefahren im Umgang mit diesem Modul	14
2.9.3	Gewährleistung und Haftung	14
<b>3</b>	<b>CANsync</b>	<b>15</b>
3.1	Allgemeines	15
3.1.1	Überblick	15
3.1.1.1	Prinzipieller Ablauf der Kommunikation	18
3.1.2	Hinweise zur Programmierung	25
3.1.3	Einstellungen und Anzeigen der Optionsmodule	26
3.1.3.1	Beschreibung der Dip-Schalter auf Optionsmodul CANsync-Slave für b maXX	26
3.1.4	Beschreibung der LEDs	28
3.1.4.1	Optionsmodul CANsync-Master	28
3.1.4.2	Optionsmodul CANsync-Slave für b maXX PLC	29
3.1.4.3	Einschalten und Initialisierung des Optionsmoduls CANsync-Slave für b maXX Regler	30
3.2	Detaillierte Informationen zum CANsync	32
3.2.0.1	Anlauf und Initialisierung	32
3.2.0.2	Zustand „synchronisiert“	33
3.2.0.3	Sollwert-Verlust bei Optionsmodul CANsync-Slave für Regler	33
3.2.1	Aufbau der Telegramme	33
3.2.1.1	Sollwertkanäle	34
3.2.1.2	Istwertkanäle	35
3.2.1.3	Kommandokanal & Antwortkanal	36
3.2.2	Registerstruktur und Funktion des Optionsmodul CANsync-Master für b maXX PLC	43
3.2.2.1	Erläuterung zur Deklaration der globalen Variablen	44
3.2.2.2	Allgemeine Register der CANsync-Anschaltung	48
3.2.2.3	Initialisierung	49
3.2.2.4	Sollwerte	52
3.2.2.5	Istwerte	56
3.2.2.6	Kommando- und Antwortkanal	62
3.2.2.7	CANsync-Slavestatus	71
3.2.3	Registerstruktur und Funktion des Optionsmoduls CANsync-Slave für b maXX PLC	71
3.2.3.1	Erläuterung zur Deklaration der globalen Variablen	72



3.2.3.2	Allgemeine Register der CANsync-Anschaltung . . . . .	77
3.2.3.3	Initialisierung . . . . .	78
3.2.3.4	Sollwerte . . . . .	81
3.2.3.5	Istwerte des CANsync-Slaves . . . . .	84
3.2.3.6	Istwerte anderer CANsync-Slaves . . . . .	88
3.2.3.7	Kommando- und Antwortkanal . . . . .	93
<b>4</b>	<b>CANsync-Funktionsbausteine . . . . .</b>	<b>99</b>
4.1	Funktionsbausteine für den synchronisierten CAN - Übersicht . . . . .	99
4.2	CANsync_BC_MA0 . . . . .	101
4.3	CANsync_BC_MA1 . . . . .	104
4.4	CANsync_BC_MA2 . . . . .	107
4.5	CANsync_BC_SL . . . . .	110
4.6	CANsync_COMM_CONTROL_MA . . . . .	113
4.7	CANsync_CONTROLWORD_MA . . . . .	117
4.8	CANsync_CONTROLWORD_SL . . . . .	120
4.9	CANsync_INIT . . . . .	122
4.10	CANsync_MODE_MA . . . . .	129
4.11	CANsync_MODE_SL . . . . .	133
4.12	CANsync_PAR_READ_MA . . . . .	137
4.13	CANsync_PAR_SL . . . . .	141
4.14	CANsync_PAR_WRITE_MA . . . . .	147
4.15	CANsync_PD_CFG_MA . . . . .	151
4.16	CANsync_PD_CFG_READ_MA . . . . .	155
4.17	CANsync_PD_CFG_READ_SL . . . . .	159
4.18	CANsync_PD_CFG_SL . . . . .	163
4.19	CANsync_PD_COMM_MA . . . . .	169
4.20	CANsync_PD_COMM_READ_MA . . . . .	175
4.21	CANsync_PD_COMM_READ_SL . . . . .	178
4.22	CANsync_PD_COMM_SL . . . . .	181
4.23	CANsync_SL_TYP_INIT . . . . .	186
4.24	CANsync_UPDOWNLOAD_MA . . . . .	188
4.25	CANsync_UPDOWNLOAD_SL . . . . .	194
	<b>Anhang A - Abkürzungen . . . . .</b>	<b>199</b>

# 1

## EINLEITUNG

Dieses Applikationshandbuch ist ein wichtiger Bestandteil ihres b maXX 4400 Gerätes; lesen Sie daher nicht zuletzt im Interesse Ihrer eigenen Sicherheit diese Dokumentation komplett durch.

In diesem Kapitel beschreiben wir die ersten Schritte.

### 1.1 Erste Schritte

---

- 1 Um CANsync-Master zu programmieren, benötigen Sie folgende Hardware:  
Grundgerät b maXX 4400,  
Optionsmodul b maXX PLC und  
Optionsmodul CANsync-Master.  
Die Hardware muss entsprechend der jeweiligen Betriebsanleitung installiert und betriebsbereit sein.
- 2 Um CANsync-Slave zu programmieren, benötigen Sie folgende Hardware:  
Grundgerät b maXX 4400,  
Optionsmodul b maXX PLC und  
Optionsmodul CANsync-Slave.  
Die Hardware muss entsprechend der jeweiligen Betriebsanleitung installiert und betriebsbereit sein.
- 3 Außerdem benötigen Sie folgende Software:  
PROPROG wt II zur Programmierung der b maXX PLC, des Optionsmoduls CANsync-Master und des Optionsmoduls CANsync-Slave für PLC.  
WinBASS II zur Parametrierung des Grundgeräts b maXX 4400 und des Optionsmoduls CANsync-Slave für Regler.

### 1.2 Verwendete Begriffe

---

Für das Baumüller-Produkt „BM4-O-CAN-06“ werden wir in dieser Dokumentation auch die Begriffe „Optionsmodul“, „Steckmodul“ oder „Optionsmodul CANsync-Master“ verwenden.

Für das Baumüller-Produkt „BM4-O-CAN-05“ werden wir in dieser Dokumentation auch die Begriffe „Optionsmodul“, „Steckmodul“ oder „Optionsmodul CANsync-Slave“ verwenden.

## 1.3 Voraussetzungen

---

Für das Baumüller-Produkt „BM4-O-PLC-01“ werden auch die Begriffe „b maXX PLC“ oder „BM4-O-PLC“ und für das Produkt „Grundgerät b maXX 4400“ wird der Begriff „b maXX“ verwendet.

Der Regler im Grundgerät wird auch „b maXX Regler“ genannt.

Eine Liste der verwendeten Abkürzungen finden Sie in [▶Anhang A - Abkürzungen◀](#) ab Seite 199.

## 1.3 Voraussetzungen

---

Dieses Handbuch baut auf das „Applikationshandbuch b maXX PLC“ auf und setzt die Kenntnis des Programmiertools PROPROG wt II und des zugehörigen Handbuchs voraus.



# GRUNDLEGENDE SICHERHEITS- HINWEISE

Jedes Baumüller-Steckmodul haben wir nach strengen Sicherheitsvorgaben konstruiert und gefertigt. Trotzdem kann die Arbeit mit dem Steckmodul für Sie gefährlich sein.

In diesem Kapitel beschreiben wir Gefahren, die bei der Arbeit mit dem Baumüller-Steckmodul auftreten können. Gefahren verdeutlichen wir mit Symbolen (Icons). Alle in dieser Dokumentation verwendeten Symbole werden wir auflisten und erklären.

Wie Sie sich vor den einzelnen Gefahren im konkreten Fall schützen können, können wir in diesem Kapitel nicht erklären. In diesem Kapitel geben wir ausschließlich allgemeine Schutzmaßnahmen. Die konkreten Schutzmaßnahmen werden wir in den nachfolgenden Kapiteln immer direkt nach dem Hinweis auf die Gefahr geben.

## 2.1 Gefahrenhinweise und Gebote

---



Gefahrenhinweise zeigen Ihnen Gefahren, die zu Verletzungen oder sogar zu Ihrem Tod führen können.

Beachten Sie immer die in dieser Dokumentation angegebenen Gefahrenhinweise.

---

Eine Gefahr teilen wir immer in eine der drei Gefahrenklassen ein. Jede Gefahrenklasse wird durch eines der folgenden Signalwörter gekennzeichnet:

**GEFAHR** (DANGER)

- erheblicher Sachschaden
- schwere Körperverletzung
- Tod - **wird** eintreffen

**WARNUNG** (WARNING)

- erheblicher Sachschaden
- schwere Körperverletzung
- Tod - **kann** eintreffen

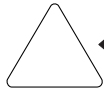
**VORSICHT** (CAUTION)

- Sachschaden
- leichte bis mittlere Körperverletzung - **kann** eintreffen

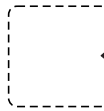
## 2.1 Gefahrenhinweise und Gebote

### 2.1.1 Struktur eines Gefahrenhinweises

Die nachfolgenden zwei Beispiele zeigen den prinzipiellen Aufbau eines Gefahrenhinweises. Ein Dreieck wird verwendet, wenn vor einer Gefahr für Lebewesen gewarnt wird. Fehlt das Dreieck, beziehen sich die Gefahrenhinweise ausschließlich auf Sachschäden.



Ein Dreieck zeigt, dass hier eine Gefahr für Lebewesen ist.  
Die Farbe der Umrandung zeigt, wie groß die Gefahr ist - je dunkler die Farbe, desto größer ist die Gefahr.



Das Icon im Viereck stellt die Gefahr dar.  
Die Farbe der Umrandung zeigt, wie groß die Gefahr ist - je dunkler die Farbe, desto größer ist die Gefahr.



Das Icon im Kreis stellt ein Gebot dar. Dieses Gebot muss der Anwender befolgen.  
(Der Kreis ist gestrichelt dargestellt, weil nicht bei jedem Gefahrenhinweis ein Gebot als Icon vorhanden ist.)



Der Kreis zeigt, dass eine Gefahr für Sachschaden existiert.



Das Icon im Viereck stellt die Gefahr dar.  
Die Farbe der Umrandung zeigt, wie groß die Gefahr ist - je dunkler die Farbe, desto größer ist die Gefahr. (Das Viereck ist gestrichelt dargestellt, weil nicht bei jedem Gefahrenhinweis die Gefahr als Icon dargestellt wird)

Der Text neben den Icons ist folgendermaßen aufgebaut:

#### **HIER STEHT DAS SIGNALWORT, WELCHES DEN GRAD DER GEFAHR ANZEIGT**




Hier schreiben wir, ob eine oder mehrere der untenstehenden Folgen eintreffen, wenn dieser Warnhinweis nicht beachtet wird.


- hier beschreiben wir die möglichen Folgen. Die schlimmste Folge steht ganz rechts.

*Hier beschreiben wir die Gefahr.*

Hier beschreiben wir, was Sie tun können, um die Gefahr zu vermeiden.


## 2.1.2 Verwendete Gefahrenhinweise

Steht vor einem Signalwort ein Gefahrzeichen:  oder  oder , dann bezieht sich der Sicherheitshinweis auf Personenschaden.

Steht vor einem Signalwort ein rundes Gefahrzeichen: , dann bezieht sich der Sicherheitshinweis auf Sachschaden.

### 2.1.2.1 Gefahrenhinweise vor Personenschaden

Zur optischen Unterscheidung verwenden wir für jede Klasse von Gefahrenhinweisen eine eigenen Umrandung für die dreieckigen Gefahrzeichen und die viereckigen Piktogramme.

Für die Gefahrenklasse **GEFAHR** (DANGER) verwenden wir das Gefahrzeichen . Folgende Gefahrenhinweise dieser Gefahrenklasse verwenden wir in dieser Dokumentation.

#### GEFAHR (DANGER)



Folgendes **wird eintreffen**, wenn Sie diesen Warnhinweis nicht beachten:

- schwere Körperverletzung
- Tod

*Die Gefahr ist: **Elektrizität**. Hier wird die Gefahr gegebenenfalls genauer beschrieben.*

Hier beschreiben wir, was Sie tun können, um die Gefahr zu vermeiden.



#### GEFAHR (DANGER)




Folgendes **wird eintreffen**, wenn Sie diesen Gefahrenhinweis nicht beachten:

- schwere Körperverletzung
- Tod

*Die Gefahr ist: **mechanische Einwirkung**. Hier wird die Gefahr gegebenenfalls genauer beschrieben.*

Hier beschreiben wir, was Sie tun können, um die Gefahr zu vermeiden.



Für die Gefahrenklasse **WARNUNG** (WARNING) verwenden wir das Gefahrzeichen . Folgende Gefahrenhinweise dieser Gefahrenklasse verwenden wir in dieser Dokumentation.

#### WARNUNG (WARNING)




Folgendes **kann eintreffen**, wenn Sie diesen Gefahrenhinweis nicht beachten:

- schwere Körperverletzung
- Tod

*Die Gefahr ist: **Elektrizität**. Hier wird die Gefahr gegebenenfalls genauer beschrieben.*

Hier beschreiben wir, was Sie tun können, um die Gefahr zu vermeiden.



Für die Gefahrenklasse **VORSICHT** (CAUTION) verwenden wir das Gefahrzeichen . Folgende Gefahrenhinweise dieser Gefahrenklasse verwenden wir in dieser Dokumentation.

## 2.1 Gefahrenhinweise und Gebote



### VORSICHT (CAUTION)

Folgendes **kann eintreffen**, wenn Sie diesen Gefahrenhinweis nicht beachten:

- leichte bis mittlere Körperverletzung

*Die Gefahr ist: **scharfe Kanten**. Hier wird die Gefahr gegebenenfalls genauer beschrieben.*

Hier beschreiben wir, was Sie tun können, um die Gefahr zu vermeiden.



### VORSICHT (CAUTION)

Folgendes **kann eintreffen**, wenn Sie diesen Warnhinweis nicht beachten:

- Umweltverschmutzung

*Die Gefahr ist: **unsachgemäße Entsorgung**. Hier wird die Gefahr gegebenenfalls genauer beschrieben.*

Hier beschreiben wir, was Sie tun können, um die Gefahr zu vermeiden.

### 2.1.2.2 Gefahrenhinweise vor Sachschaden

Steht vor einem Signalwort ein rundes Gefahrzeichen: ⓘ dann bezieht sich der Sicherheitshinweis auf Sachschaden.



### VORSICHT (CAUTION)

Folgendes **kann eintreffen**, wenn Sie diesen Gefahrenhinweis nicht beachten:

- Sachschaden

*Die Gefahr ist: **elektrostatische Entladung**. Hier wird die Gefahr gegebenenfalls genauer beschrieben.*

Hier beschreiben wir, was Sie tun können, um die Gefahr zu vermeiden.

### 2.1.2.3 Verwendete Gebotszeichen



Sicherheitshandschuhe tragen



Sicherheitsschuhe tragen

## 2.2 Infozeichen

---



### HINWEIS

Dieser Hinweis ist eine besonders wichtige Information.

---

## 2.3 Rechtliche Hinweise

---

Diese Dokumentation wendet sich an technisch qualifiziertes Personal, welches speziell ausgebildet ist und gründlich mit allen Warnungen und Instandhaltungsmaßnahmen vertraut ist.

Die Geräte sind nach dem Stand der Technik gefertigt und betriebssicher. Sie lassen sich gefahrlos installieren und in Betrieb setzen und funktionieren problemlos, wenn sichergestellt ist, dass die Hinweise der Dokumentation beachtet werden.

Der Benutzer trägt die Verantwortung für die Durchführung von Service und Inbetriebnahme gemäss den Sicherheitsvorschriften der geltenden Normen und allen anderen relevanten staatlichen oder örtlichen Vorschriften betreffend Leiterdimensionierung und Schutz, Erdung, Trennschalter, Überstromschutz usw.

Für Schäden, die bei der Montage oder beim Anschluss entstehen, haftet der Benutzer.

## 2.4 Bestimmungsgemäße Verwendung

---

Sie müssen das Steckmodul immer bestimmungsgemäß verwenden. Untenstehend haben wir einige wichtige Hinweise für Sie zusammengestellt. Die untenstehenden Hinweise sollen Ihnen ein Gefühl für die bestimmungsgemäße Verwendung des Steckmoduls geben. Mit den untenstehenden Hinweisen erheben wir keinen Anspruch auf Vollständigkeit - beachten Sie alle in dieser Betriebsanleitung gegebenen Hinweise.

- Sie dürfen das Steckmodul nur in Geräte der Reihe b maXX 4400 einbauen.
- Projektieren Sie die Anwendung so, dass Sie das Steckmodul immer innerhalb seiner Spezifikationen betreiben.
- Sorgen Sie dafür, dass ausschließlich qualifiziertes Personal mit diesem Steckmodul arbeitet.
- Montieren Sie das Steckmodul nur an dem/den vorgegebenen Steckplatz/Steckplätzen.
- Installieren Sie das Steckmodul so wie in es in dieser Dokumentation vorgegeben ist.
- Sorgen Sie dafür, dass die Anschlüsse immer den vorgegebenen Spezifikationen entsprechen.
- Betreiben Sie das Steckmodul nur, wenn es technisch einwandfrei ist.
- Betreiben Sie das Steckmodul immer in einer Umgebung, wie sie in den „Technischen Daten“ vorgeschrieben ist.
- Betreiben Sie das Steckmodul immer in serienmäßigem Zustand.  
Aus Sicherheitsgründen dürfen Sie das Steckmodul nicht umbauen.

- Beachten Sie alle diesbezüglichen Hinweise, falls Sie das Steckmodul lagern.

Sie verwenden das Steckmodul dann bestimmungsgemäß, wenn Sie alle Hinweise und Informationen dieser Betriebsanleitung beachten.

### 2.5 Sachwidrige Verwendung

---

Im Folgenden listen wir einige Beispiele sachwidriger Verwendung auf. Die untenstehenden Hinweise sollen Ihnen ein Gefühl dafür geben, was eine sachwidrige Verwendung des Steckmoduls ist. Wir können aber nicht alle erdenklichen sachwidrigen Verwendungen hier auflisten. Alle Verwendungen, bei denen die Hinweise dieser Dokumentation missachtet werden, sind sachwidrig und somit verboten, insbesondere in folgenden Fällen:

- Sie haben das Steckmodul in andere Geräte als die Reihe b maXX 4400 eingebaut.
- Sie haben Hinweise dieser Betriebsanleitung missachtet.
- Sie haben das Steckmodul nicht bestimmungsgemäß verwendet.
- Sie haben das Steckmodul
  - unsachgemäß montiert,
  - unsachgemäß angeschlossen,
  - unsachgemäß in Betrieb genommen,
  - unsachgemäß bedient,
  - von nicht bzw. nicht ausreichend qualifiziertem Personal montieren, anschließen, in Betrieb nehmen und betreiben lassen,
  - überlastet,
- betrieben
  - mit defekten Sicherheitseinrichtungen,
  - mit nicht ordnungsgemäß angebrachten bzw. ohne Sicherheitsvorrichtungen,
  - mit nicht funktionsfähigen Sicherheits- und Schutzvorrichtungen
  - außerhalb der vorgeschriebenen Umgebungsbedingungen
- Sie haben das Steckmodul umgebaut, ohne dass dies schriftlich von der Firma Baumüller Nürnberg Electronic GmbH & Co. KG genehmigt wurde.
- Sie haben die Anweisungen bezüglich Wartung in den Komponentenbeschreibungen nicht beachtet.
- Sie haben das Steckmodul unsachgemäß mit Produkten anderer Hersteller kombiniert.
- Sie haben das Antriebssystem mit fehlerhaften und/oder fehlerhaft dokumentierten Produkten anderer Hersteller kombiniert.
- Ihre selbsterstellte Software der PLC enthält Programmierfehler, die zu einer Fehlfunktion führen.

Die „Allgemeinen Verkaufs- und Lieferbedingungen“ Version 1.1 vom 15.02.2002 bzw. die jeweils neueste Version der Firma Baumüller Nürnberg Electronic GmbH & Co. KG gelten grundsätzlich. Diese stehen Ihnen spätestens seit Vertragsabschluss zur Verfügung.

## 2.6 Schutzeinrichtungen

Während des Transports werden die Steckmodule durch ihre Verpackung geschützt. Entnehmen Sie das Steckmodul erst unmittelbar vor der Montage der Transportverpackung.

Die Abdeckhaube des Reglerteils der b maXX Geräte schützt in Schutzklasse IP20 die Steckmodule vor Verschmutzung und Schäden durch statische Entladungen bei Berührungen. Stecken Sie daher nach erfolgter Montage des Steckmoduls die Abdeckhaube wieder auf.

## 2.7 Ausbildung des Personals



### WARNUNG (WARNING)

Folgendes **kann eintreffen**, wenn Sie diesen Gefahrenhinweis nicht beachten:

- schwere Körperverletzung
- Tod

Geräte der Firma Baumüller Nürnberg Electronic GmbH & Co. KG dürfen ausschließlich von qualifiziertem Personal montiert, installiert, betrieben und gewartet werden.

Qualifiziertes Personal (Fachkräfte) wird folgendermaßen definiert:

### Qualifiziertes Personal

Von der Firma Baumüller Nürnberg Electronic GmbH & Co. KG autorisierte Elektro-Ingenieure und Elektro-Fachkräfte des Kunden oder Dritter, die Installation und Inbetriebnahme von Baumüller-Antriebssystemen erlernt haben und berechtigt sind, Stromkreise und Geräte gemäß den Standards der Sicherheitstechnik in Betrieb zu nehmen, zu erden und zu kennzeichnen.

Qualifiziertes Personal verfügt über eine Ausbildung oder Unterweisung gemäß den örtlich jeweils gültigen Standards der Sicherheitstechnik in Pflege und Gebrauch angemessener Sicherheitsausrüstung.

### Anforderungen an das Bedienungs-personal

Die Bedienung des Antriebssystems darf nur von Personen durchgeführt werden, die dafür ausgebildet, eingewiesen und befugt sind.

Störungsbeseitigung, Instandhaltung, Reinigung, Wartung und Austausch dürfen nur durch geschultes oder eingewiesenes Personal durchgeführt werden. Diese Personen müssen die Betriebsanleitung kennen und danach handeln.

Inbetriebnahme und Einweisung dürfen nur vom qualifizierten Personal durchgeführt werden.

## 2.8 Sicherheitsmaßnahmen im Normalbetrieb

- Beachten Sie am Aufstellort des Gerätes die gültigen Sicherheitsbestimmungen für die Anlage, in die dieses Gerät eingebaut ist.
- Versehen Sie das Gerät mit zusätzlichen Überwachungs- und Schutzeinrichtungen, falls Sicherheitsbestimmungen dies fordern.
- Beachten Sie die Sicherheitsmaßnahmen für das Gerät, in das das Steckmodul eingebaut ist.

### 2.9 Verpflichtung und Haftung

---

Damit Sie sicherheitsgerecht mit diesen Optionsmodulen arbeiten können, müssen Sie die Gefahrenhinweise und Sicherheitshinweise dieser Dokumentation kennen und beachten.

#### 2.9.1 Gefahrenhinweise und Sicherheitshinweise beachten

---

Wir verwenden in dieser Betriebsanleitung optisch einheitliche Sicherheitshinweise, die sie vor Personen- und Sachschäden bewahren sollen.



##### **WARNUNG** (WARNING)

Folgendes **kann eintreffen**, wenn Sie diesen Gefahrenhinweis nicht beachten:

- schwere Körperverletzung
- Tod

Alle Personen, die an und mit Geräten der Reihe b maXX arbeiten, müssen bei ihren Arbeiten diese Betriebsanleitung verfügbar haben und die hierin enthaltenen Anweisungen und Hinweise - insbesondere die Sicherheitshinweise - beachten.

Außerdem müssen alle Personen, die an diesem Gerät arbeiten, zusätzlich alle Regeln und Vorschriften, die am Einsatzort gelten, kennen und beachten.

#### 2.9.2 Gefahren im Umgang mit diesem Modul

---

Die Optionsmodule CANsync-Master und CANsync-Slave wurden nach dem Stand der Technik und unter Einhaltung der geltenden Richtlinien und Normen entwickelt und gefertigt. Dennoch können bei der Verwendung Gefahren entstehen. Eine Übersicht möglicher Gefahren finden Sie im Kapitel [►Grundlegende Sicherheitshinweise◄](#) ab Seite 7 . Weiterhin warnen wir Sie vor der akuten Gefahr an der entsprechenden Stelle in dieser Dokumentation.

#### 2.9.3 Gewährleistung und Haftung

---

Alle Angaben in dieser Dokumentation sind unverbindliche Kundeninformationen, unterliegen einer ständigen Weiterentwicklung und werden laufend durch unseren permanenten Änderungsdienst aktualisiert.

Gewährleistungs- und Haftungsansprüche gegen die Firma Baumüller Nürnberg Electronic GmbH & Co. KG sind ausgeschlossen, wenn insbesondere eine oder mehrere der von uns in [►Sachwidrige Verwendung◄](#) ab Seite 12 oder unten aufgeführten Ursachen den Schaden bewirkt hat/haben:

- Eintritt eines Katastrophenfalls durch Fremdkörperwirkung bzw. höhere Gewalt



# CANSYNC

## HINWEIS



In diesem Kapitel genannte Funktionsbausteine befinden sich in den Bibliotheken SYSTEM1\_PLC01\_20bd00 (oder höher), SYSTEM2\_PLC01\_20bd00 (oder höher) und CANSync\_PLC01\_20bd00 (oder höher).

In diesem Kapitel genannten Datentypen sind in der Bibliothek BM\_TYPES\_20bd03 (oder höher) definiert.

Zur Programmierung des CANSync unter PROPROG wt II werden diese Bibliotheken in ein Projekt eingebunden.

## 3.1 Allgemeines

### 3.1.1 Überblick

Der CANSync-Feldbus ist eine Entwicklung der Baumüller Nürnberg Electronic GmbH & Co. KG. Das Ziel, mechanische Königswellen durch eine elektronische Leitachse zu ersetzen, wurde dadurch erreicht, dass allen angeschlossenen Antrieben (⇒ CANSync-Slaves) der Leitachswert zum selben Zeitpunkt zur Verfügung gestellt wird (zeitsynchrone Übertragung).

Physikalische Basis stellt der CAN-Bus dar. Dieser wurde erweitert durch die Hinzufügung eines Synchronisations-Signals (SYNC-Signal). Das SYNC-Signal wird auf zwei zusätzlichen Adern im CAN-Kabel übertragen. Das SYNC-Signal dient zur Hardwaresynchronisation des CANSync-Masters mit allen am CANSync-Bus befindlichen CANSync-Slaves. Dadurch ist es im Gegensatz zum CAN-Bus möglich, die Telegramme zu definierten Zeitpunkten zu senden und zu empfangen. Es wird ein garantierter, hoher Datendurchsatz erzielt, der zudem einen festen Zeitbezug auf dem CANSync-Bus besitzt.

Der CANSync-Bus ist ein Master-Slave-Bus mit einem CANSync-Master und bis zu 32 CANSync-Slaves. Zur Unterscheidung der CANSync-Slaves wird jedem CANSync-Slave eine Slavenummer zugeordnet. Die Slavenummer wird über Dip-Schaltereinstellung festgelegt (siehe Kapitel „Dip-Schalter“ in ▶ Betriebsanleitung CANSync-Slave für b maXX ◀).

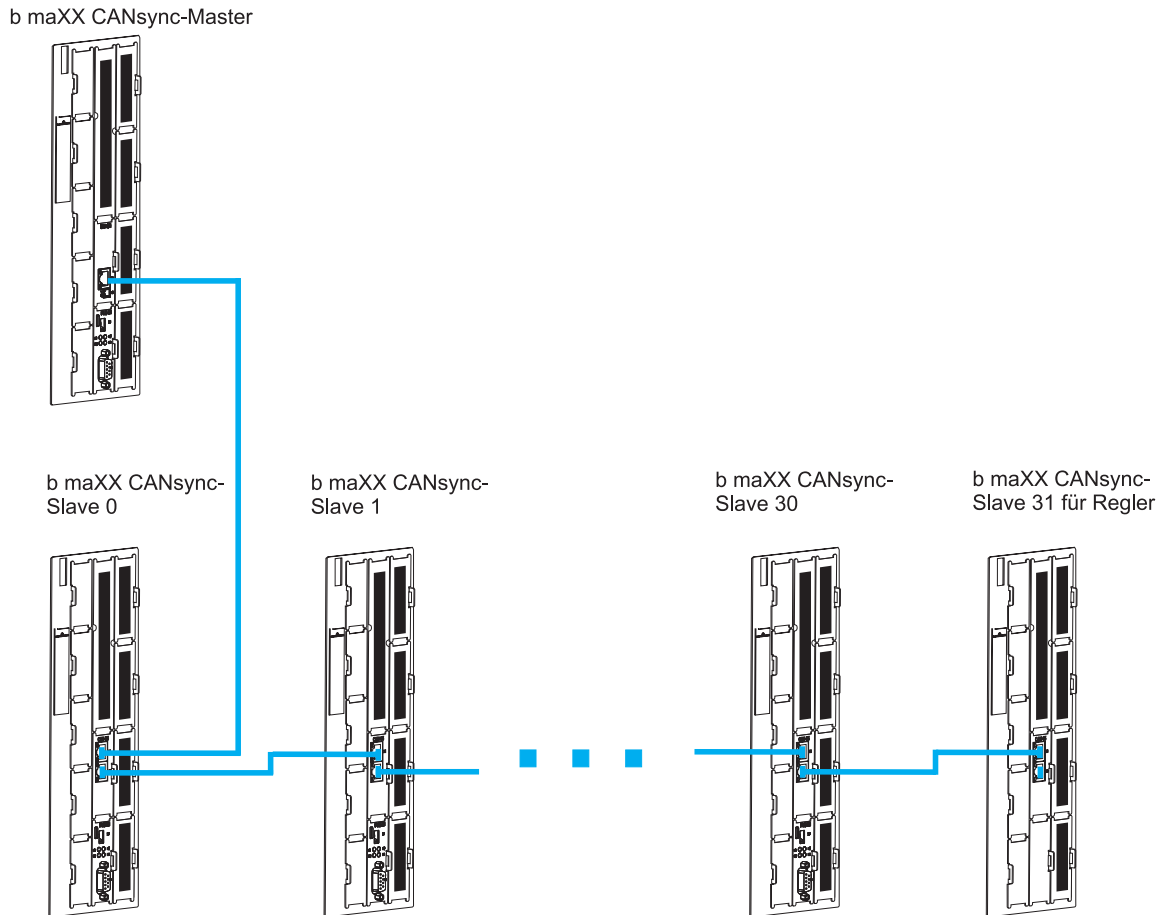


Abbildung 1: CANsync-Bus, CANsync-Master mit CANsync-Slaves 0, 1, ..., 30,31

Das CANsync-Synchronisationssignal (SYNC-Signal)

- ist ein spezifiziertes Hardwaresignal,
- wird auf der CANsync-Anschaltung des CANsync-Masters generiert,
- wird über zwei zusätzliche Leitungen im CANsync-Bus übertragen,

Je nach Betriebsart und Übertragungsgeschwindigkeit (Baudrate) auf dem CANsync-Bus wird das SYNC-Signal in einem bestimmten Raster, der CANsync-Zykluszeit (Zeit zwischen zwei fallenden Flanken des SYNC-Signals), generiert.

Baudrate	CANsync-Zykluszeit
1 MBit/s	1 ms
500 kBit/s	2 ms
250 kBit/s	4 ms
125 kBit/s	8 ms

Der Telegrammverkehr des CANsync-Busses findet in einer festgelegten Reihenfolge statt, so dass sich die einzelnen Telegramme immer in definierten Zeitfenstern befinden, diese werden auch Kanäle genannt. Folgende Kanäle sind im CANsync definiert und werden vom CANsync-Master gesendet:

- Sollwert-Telegramme, WRC1 und WRC2 (**W**rite**C**hannel)
- Broadcast-Telegramme, CC (**C**ommand**C**hannel)
- Parameter-Telegramme, CC
- Up-Download-Telegramme, CC

Die CANsync-Slaves müssen ihre Antworten dem Zeitschema des CANsync-Masters anpassen, ihnen stehen folgende Kanäle zur Verfügung:

- Istwert-Telegramme, RDC1 und RDC2 (**R**ead**C**hannel)
- Parameter-Antwort-Telegramme, RC (**R**esponse**C**hannel)
- Up-/Download-Antwort-Telegramme, RC

#### HINWEIS



WRC1 und WRC2 werden nachfolgend auch Sollwertkanal 1 und Sollwertkanal 2 genannt.  
RDC1 und RDC2 werden nachfolgend auch Istwertkanal 1 und Istwertkanal 2 genannt.  
CC wird nachfolgend auch Kommandokanal genannt.  
RC wird nachfolgend auch Antwortkanal genannt.

Das folgende Beispiel zeigt das Zeitschema der jeweiligen Telegramme mit der Übertragungsgeschwindigkeit 1 Mbit/s (CANsync-Intervall mit der CANsync-Zykluszeit 1 ms):

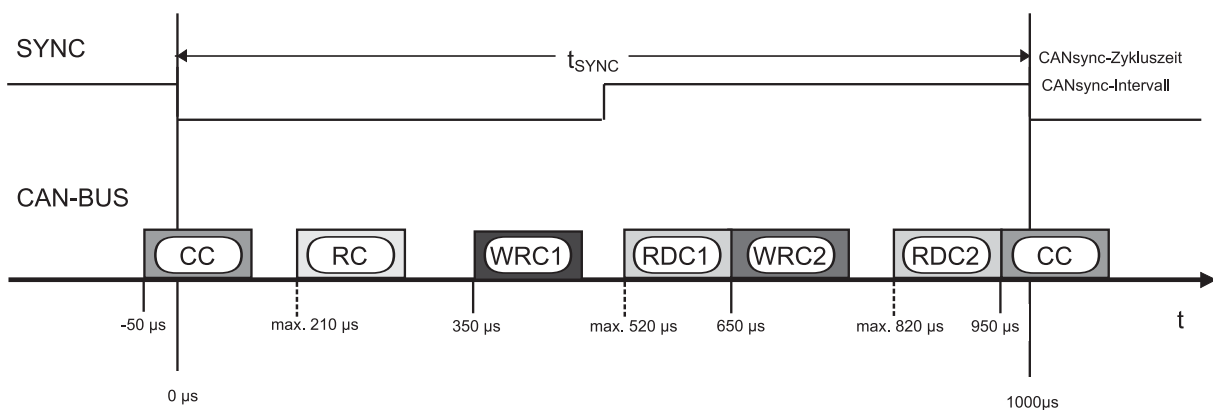


Abbildung 2: Zeitschema des CANsync-Intervalls mit der Übertragungsgeschwindigkeit 1 Mbit/s

Die Zeitpunkte für die Kanäle sind in Abhängigkeit von der Baudrate in der nachfolgenden Tabelle angegeben:

Tabelle: Zuordnung Übertragungsgeschwindigkeit (Baudrate), CANsync-Zykluszeit  $t_{\text{SYNC}}$  in ms und maximale Buslänge in m. Die angegebenen Zeiten sind Maximalwerte.

Baudrate	CANsync-Zykluszeit in $\mu\text{s}$ $t_{\text{SYNC}}$	$t_{\text{WRC1}}$ in $\mu\text{s}$	$t_{\text{RDC1}}$ in $\mu\text{s}$	$t_{\text{WRC2}}$ in $\mu\text{s}$	$t_{\text{RDC2}}$ in $\mu\text{s}$	$t_{\text{CC}}$ in $\mu\text{s}$	$t_{\text{RC}}$ in $\mu\text{s}$	max. CANsync-Buslänge
1 MBit/s	1000	350	520	650	820	950	210	34 m
500 kBit/s	2000	730	1060	1320	1630	1890	460	134 m
250 kBit/s	4000	1430	2100	2620	3290	3810	900	300 m
125 kBit/s	8000	3020	4310	5350	6640	7680	1920	600 m

### 3.1.1.1 Prinzipieller Ablauf der Kommunikation

#### Prozessdatenkommunikation

Für die Prozessdatenkommunikation werden die Sollwertkanäle WRC1 und WRC2 (WriteChannel), die Istwertkanäle RDC1 und RDC2 (ReadChannel) sowie der Kommandokanal CC (CommandChannel) verwendet. In jedem CANsync-Intervall werden sie in der vorgegebenen Reihenfolge gesendet.

Die Sollwert-Telegramme in den Kanälen WRC1 und WRC2 werden vom Master verschickt. Sie besitzen eine fest definierte Länge an Nutzdaten von jeweils 64 Bit  $\approx$  8 Byte  $\approx$  4 Worte!

Sie werden von allen Slaves/Teilnehmern empfangen, dabei entscheidet jeder Slave eigenständig, aufgrund seiner eingestellten Konfiguration ( $\rightarrow$  Mapping), welche Daten für ihn Sollwerte darstellen. Getroffen wird diese Einstellung für

- das Optionsmodul CANsync-Slave für b maXX **PLC** (BM4-O-CAN-05; Dip-Schalter 9 = ON) in dem zu programmierenden Mapping im Applikationsprogramm auf dem Optionsmodul b maXX PLC mit den FBs CANsync\_PD\_CFG\_SL und CANsync\_PD\_CFG\_READ\_SL
- das Optionsmodul CANsync-Slave für b maXX **Regler** (BM4-O-CAN-05; Dip-Schalter 9 = OFF) in WinBASS II auf den Seiten „BACI“ und „Optionsmodul G, H - Konfiguration“.

Als zusätzliche Information wird dem Sollwert-Telegramm hinzugefügt, welcher Slave in dem selben CANsync-Intervall sein bereits vorbereitetes Istwert-Telegramm an den Master zurücksenden soll. So ist es möglich, die Istwert-Telegramme von maximal zwei Slaves pro CANsync-Intervall zu erhalten, da zwei Istwert-Telegramme zur Verfügung stehen (in den Kanälen RDC1 und RDC2). Auch diese Telegramme besitzen die Nutzdatengröße von jeweils 64 Bit  $\approx$  8 Byte  $\approx$  4 Worte.

Eine automatische Abfrage aller Slaves nacheinander ist ebenfalls programmierbar.

In der Regel wird nur ein CANsync-Slave pro CANsync-Intervall auf seine Istwerte abgefragt, so dass die Istwert-Telegramme in den Kanälen RDC1 und RDC2 von einem Slave stammen.



## HINWEIS

Bei maximaler Buskonfiguration mit 32 Slaves benötigt man mindestens 16 CANsync-Intervalle um alle aktuellen Istwerte im Master zur Verfügung zu haben, wenn pro CANsync-Intervall zwei Slaves ihre Istwerte an den Master zurückmelden!

Beim Optionsmodul CANsync-Slave für b maXX Regler gehört zu den Prozessdaten das Steuerwort und das Statuswort. Während das Statuswort wie ein gewöhnlicher Istwert in die Istwert-Telegramme 1 oder 2, je nach eingestelltem Mapping, eingetragen wird, muss das Steuerwort gesondert behandelt werden. Es ist als Broadcastkommando definiert, welches im CC gesendet wird.

## Mapping

Im folgenden wird das Prinzip des Mapping am Beispiel der CANsync-Master-Anschaltung beschrieben. Die Einstellung des Mapping bei den CANsync-Slaves erfolgt für:

- das Optionsmodul CANsync-Slave für b maXX **PLC** (BM4-O-CAN-05; Dip-Schalter 9 = ON) im Applikationsprogramm auf der b maXX PLC mit den FBs CANsync\_PD\_CFG\_SL und CANsync\_PD\_CFG\_READ\_SL
- das Optionsmodul CANsync-Slave für b maXX **Regler** (BM4-O-CAN-05; Dip-Schalter 9 = OFF) in WinBASS II auf der Seite „Optionsmodul G, H - Konfiguration“.

Jeweils 64 Bit stehen als Nutzdaten für die Sollwert-Telegramme 1 und 2 zur Verfügung. Damit ergeben sich für die Sollwertvorgabe durch den CANsync-Master folgende Möglichkeiten:

- 4 Wort-Sollwerte (je 16 Bit)
- 2 Doppelwort-Sollwerte (je 32 Bit)
- 1 Doppelwort-Sollwert (32 Bit) und 2 Wort-Sollwerte (je 16 Bit)

Die Sollwerte werden in ein Feld (Array) eingetragen, dass bei der Prozessdatenkommunikation am FB CANsync\_PD\_COMM\_MA angeschlossen wird. Es besteht aus 8 Einträgen mit jeweils 32 Bit Länge:

Sollwerte für das Sollwert-Telegramm 1	Sollwert 0	Highword	Sollwert 0	Lowword
	Sollwert 1	Highword	Sollwert 1	Lowword
	Sollwert 2	Highword	Sollwert 2	Lowword
	Sollwert 3	Highword	Sollwert 3	Lowword
Sollwerte für das Sollwert-Telegramm 2	Sollwert 4	Highword	Sollwert 4	Lowword
	Sollwert 5	Highword	Sollwert 5	Lowword
	Sollwert 6	Highword	Sollwert 6	Lowword
	Sollwert 7	Highword	Sollwert 7	Lowword

Diese Sollwerte werden im Kommunikations-RAM der CANsync-Anschaltung als Doppelworte zwischengespeichert. Aus diesem Grund ist es notwendig der CANsync-Anschal-

zung mitzuteilen, welche Sollwerte und welches Wort, High- oder Lowword, für das Sollwert-Telegramm zu verwenden sind. Zu treffen ist diese Einstellung am Funktionsbaustein CANsync\_PD\_CFG\_MA. Da sich diese Einstellung in der Regel nicht mehr ändert, ist sie während der Initialisierung des CANsync-Busses von Anwender vorzunehmen.

### HINWEIS



Doppelwort-Parameter im Optionsmodul CANsync-Slave für b maXX Regler (BM4-O-CAN-05; Dip-Schalter 9 = OFF) müssen in der Reihenfolge:

Erst Lowword, dann Highword übertragen werden (wie in den folgenden Beispielen angegeben).

### Beispiel:

Sie wollen einen Doppelwort-Sollwert als Sollwert 0 im Sollwert-Telegramm 1 (im WRC1) schreiben und zusätzlich zwei Wortsollwerte als Sollwert 2 und Sollwert 3:

Zu treffende Einstellung: a\_WRC1 = [ 0, 0, 2, 3]  
a\_HL\_WRC1 = [FALSE, TRUE, FALSE, FALSE]

Sollwerte für das Sollwert-Telegramm 1	Sollwert 0	Highword	Sollwert 0	Lowword
		----		----
		----	Sollwert 2	Lowword
		----	Sollwert 3	Lowword

Die verwendeten Arrays werden an die Eingänge des FB CANsync\_PD\_CFG\_MA angeschlossen. Das Array a\_WRC1 definiert die Positionen der zu übertragenen Sollwerte, Doppelwort auf Sollwert 0, Wortsollwerte auf Sollwert 2 und 3. Das Array a\_HL\_WRC1 gibt dem System an, ob das High- (TRUE) oder Lowword (FALSE) aus dem Kommunikations-RAM genommen werden soll.

Für das Sollwert-Telegramm 1 im SWK 1 ergibt sich folgende Belegung:

	Wort 3	Wort 2	Wort 1	Wort 0
SWK 1 / WRC 1	Sollwert 3 Lowword	Sollwert 2 Lowword	Sollwert 0 Highword	Sollwert 0 Lowword

Auch für die beiden Istwert-Telegramme 1 und 2 stehen jeweils 64 Bit Nutzdaten zur Verfügung. Es können so

4 Wort-Istwerte (je 16 Bit)  
2 Doppelwort-Istwerte (je 32 Bit)  
1 Doppelwort-Istwert (32 Bit) und 2 Wort-Istwerte (je 16 Bit)

übertragen werden.

Diese Istwerte werden ebenfalls im Kommunikations-RAM als Doppelworte abgespeichert, so dass es auch hier notwendig ist, eine Zuordnung zu treffen.

Istwerte aus dem Istwert-Telegramm 1	Istwert 0	Highword	Istwert 0	Lowword
	Istwert 1	Highword	Istwert 1	Lowword
	Istwert 2	Highword	Istwert 2	Lowword
	Istwert 3	Highword	Istwert 3	Lowword
Istwerte aus dem Istwert-Telegramm 2	Istwert 4	Highword	Istwert 4	Lowword
	Istwert 5	Highword	Istwert 5	Lowword
	Istwert 6	Highword	Istwert 6	Lowword
	Istwert 7	Highword	Istwert 7	Lowword

**Beispiel:**

Sie wollen aus dem Istwert-Telegramm 1 (im RDC1) einen Doppelwort-Istwert als Istwert 0 (Position im Telegramm Wort 0 und 1) und einen zweiten Doppelwort-Istwert (Wort 2 und 3) als Istwert 2 lesen:

Zu treffende Einstellung: a\_RDC1 = [ 0, 0, 2, 2]  
 a\_HL\_RDC1 = [FALSE, TRUE, FALSE, TRUE]

Istwerte aus dem Istwert-Telegramm 1	Istwert 0	Highword	Istwert 0	Lowword
	----		----	
	Istwert 2	Highword	Istwert 2	Lowword
	----		----	

Die verwendeten Arrays werden an die Eingänge des FB CANsync\_PD\_CFG\_READ\_MA angeschlossen. Das Array a\_RDC1 definiert die Positionen der zu übertragenen Istwerte, Doppelwort auf Istwert 0, Doppelwort auf Istwert 2. Das Array a\_HL\_RDC1 gibt dem System an, ob das High- (TRUE) oder Lowword (FALSE) in das Kommunikations-RAM geschrieben werden soll.

Für das Sollwert-Telegramm 1 im SWK 1 ergibt sich folgende Belegung:

	Wort 3	Wort 2	Wort 1	Wort 0
IWK 1 / RDC 1	Istwert 2 Highword	Istwert 2 Lowword	Istwert 0 Highword	Istwert 0 Lowword

Im CANsync-Master werden die Istwerte der einzelnen CANsync-Slaves in unterschiedliche Bereiche des Kommunikations-RAM geschrieben, jedem CANsync-Slave wird ein eigener Bereich zugewiesen. Dieses Feld besteht aus 32 (da maximale Teilnehmerzahl)

mal 8 (da 8 Istwerte) Einträgen, die jeweils 32 Bit (da Doppelwort-Istwerte) Daten beinhalten.

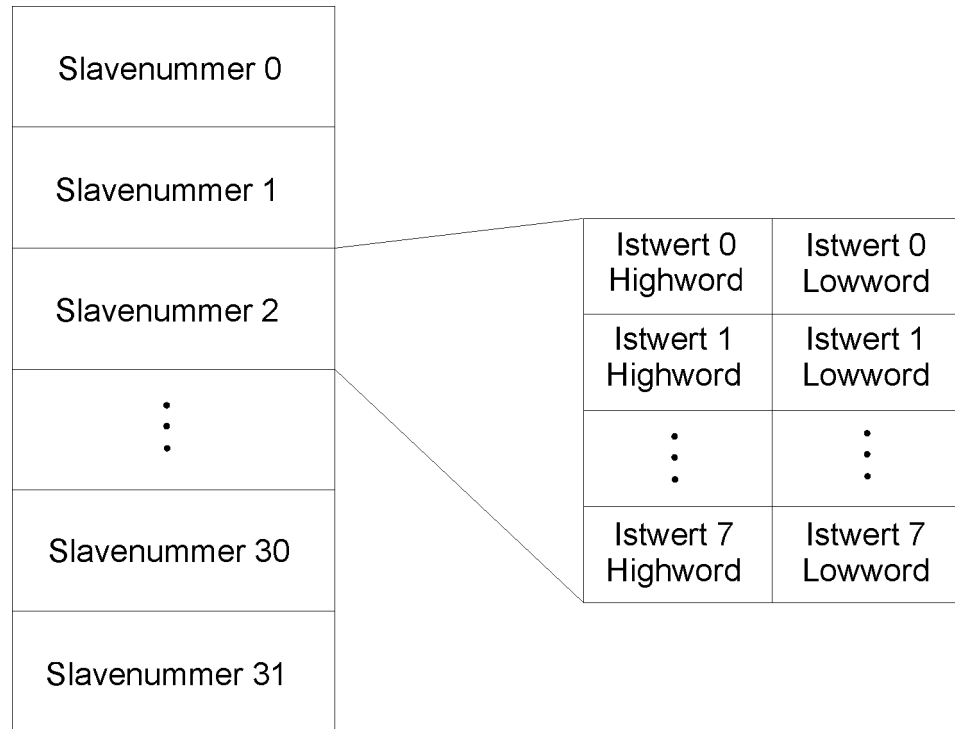


Abbildung 3: Istwerteintragung in das Kommunikations-RAM des CANsync-Masters der CANsync-Slaves, beispielhaft für Slave 2

Da in jedem CANsync-Istwert-Telegramm nur ein CANsync-Slave seine Istwerte und eventuell sein Statuswort sendet, ist es für alle anderen CANsync-Slaves möglich, diese Telegramme mitzuhören. Dadurch entsteht die Möglichkeit, dass diese Istwerte als Sollwerte für einen oder mehrere andere CANsync-Slaves genutzt werden. Ist dieses gewünscht, so muss die Einstellung lediglich im Mapping getroffen werden.

### Zusammenfassung der Prozessdatenkommunikation

Der CANsync-Master sendet in jedem CANsync-Intervall das Sollwert-Telegramm 1 und 2 und fordert (in der Regel) mit den Sollwert-Telegrammen die Istwert-Telegramme 1 und 2 von einem CANsync-Slave an.

Alle CANsync-Slaves empfangen die Sollwert-Telegramme des CANsync-Masters und senden nach der entsprechenden Aufforderung ihre Istwert-Telegramme an den CANsync-Master. Die Einstellung, welche Sollwerte in den Sollwert-Telegrammen für den CANsync-Slave relevant sind, wird im Mapping getroffen. Ebenfalls wird dort definiert, welche Istwert-Konstellation in das Istwert-Telegramm einzutragen ist.

Alle CANsync-Slaves können die vom Master angeforderten Istwert-Telegramme der anderen CANsync-Slaves auswerten.



Die Konfiguration des Mappings erfolgt für:

- das Optionsmodul CANsync-Master für b maXX PLC (BM4-O-CAN-06):  
im Applikationsprogramm auf der b maXX PLC mit den FBs
 

CANsync_PD_CFG_MA (FB wird 1 mal eingesetzt)	Mapping der zu sendenden Sollwert-Telegramme 1 und 2 im b maXX CANsync-Master
CANsync_PD_CFG_READ_MA (FB wird je CANsync-Slave eingesetzt, max. 32 mal)	Mapping der empfangenen Istwert-Telegramme 1 und 2 eines CANsync-Slaves
  
- das Optionsmodul CANsync-Slave für b maXX **PLC** (BM4-O-CAN-05; Dip-Schalter 9 = ON):  
im Applikationsprogramm auf der b maXX PLC mit den FBs
 

CANsync_PD_CFG_SL (FB wird 1 mal eingesetzt)	Mapping der empfangenen Sollwert-Telegramme 1 und 2 und der zu sendenden Istwert-Telegramme 1 und 2 im CANsync-Slave für b maXX PLC
CANsync_PD_CFG_READ_SL (FB wird je weiteren CANsync-Slave eingesetzt, max. 31 mal)	Mapping der empfangenen Istwert-Telegramme 1 und 2 eines CANsync-Slaves im CANsync-Slave für b maXX PLC
  
- das Optionsmodul CANsync-Slave für b maXX **Regler** (BM4-O-CAN-05; Dip-Schalter 9 = OFF):  
in WinBASS II auf der Seite „Optionsmodul G, H - Konfiguration“ (siehe Betriebsanleitung „Optionsmodul CANsync-Slave“)

### Bedarfsdaten

Die Bedarfsdatenkommunikation findet im CANsync über den CommandChannel (CC) und den ResponseChannel (RC) statt. Dabei werden vom CANsync-Master im Kommandokanal (CC) Telegramme gesendet, welche Aktionen in einem oder mehreren CANsync-Slaves auslösen.

Es stehen mehrere Telegramme zur Verfügung:

- Broadcast-Telegramme
- Steuerwort-Telegramme (Sonderfall eines Broadcast-Telegramm)
- Parameter-Telegramme
- Up-/Download-Telegramme

Der CANsync-Slave sendet seine Antwort im Antwortkanal (RC), es kann sich um

- Parameter-Antwort-Telegramme
- Up-/Download-Antwort-Telegramme

handeln.



### HINWEIS

Der CANsync-Master kann in jedem CANsync-Intervall Kommando-Telegramme senden. Die CANsync-Slaves antworten nur nach Aufforderung durch den CANsync-Master.

Die unterschiedlichen Telegramme im Kommandokanal werden in einer durch Prioritäten vorgegebenen Reihenfolge abgearbeitet:

Tabelle: Prioritätenvergabe der Telegramme im Kommandokanal (CC)

Telegrammtyp	Priorität
Broadcast-Telegramm 0	höchste
Broadcast-Telegramm 1	↑
Broadcast-Telegramm 2	
Steuerwort-Telegramme	
Parameter-Telegramme	↓
Up-/Download-Telegramme	niedrigste

Die Konsequenz dieser Prioritäten ist, dass kein anderes Telegramm verschickt werden kann, wenn ein höherprioreres gesendet wird. Sendet man z. B. in jedem CANsync-Intervall das Steuerwort-Telegramm, so kann nie ein Parameter- oder Up-/Download-Telegramm verschickt werden!

In jedem CANsync-Intervall kann eines der folgenden Telegramme gesendet werden:

- ein Broadcast-Telegramm an **alle** CANsync-Slaves oder
- ein Steuerwort-Telegramm an **einen** CANsync-Slave oder
- ein Parameter-Telegramm an **einen** CANsync-Slave oder
- ein Up-/Download-Telegramm an **einen** CANsync-Slave

Die Steuerwort-, Parameter- oder Up-/Download-Telegramme werden, sofern kein Broadcast-Telegramm zu senden ist, in jedem CANsync-Intervall an einen CANsync-Slave gesendet.

Dabei kann im CANsync-Master eingestellt werden, ob an einen bestimmten CANsync-Slave ein Telegramm gesendet wird, oder ob er automatisch nacheinander an alle CANsync-Slaves ein Telegramm sendet.

Da nicht immer die Maximalkonfiguration an CANsync-Slaves (32 Stück ohne Cluster) vorhanden sein muss, kann dem CANsync-Master die maximale Slavenummer zur Steuerwort-, Parameter-, Up-/Download-Telegramm-Versendung sowie Istwert-Telegramm-Anforderung mitgeteilt werden. Sie wird an dem Funktionsbaustein CANsync\_COMM\_CONTROL\_MA oder CANsync\_PD\_COMM\_MA getroffen.

### 3.1.2 Hinweise zur Programmierung

#### Initialisierung

Wird der b maXX 4400 mit b maXX PLC (BM4-O-PLC-01) und dem Optionsmodul CANsync-Master (BM4-O-CAN-06) verwendet, muss die CANsync-Master-Anschaltung auf dem Optionsmodul CANsync-Master initialisiert werden.

Anschließend wird der aktive Betrieb auf der CANsync-Master-Anschaltung freigegeben.

Benötigte FBs und deren Reihenfolge bei der Initialisierung der CANsync-Master-Anschaltung auf dem Optionsmodul CANsync-Master:

```
CANsync_SL_TYP_INIT
CANsync_INIT
CANsync_COMM_CONTROL_MA
CANsync_PD_CFG_MA
CANsync_PD_CFG_READ_MA (je CANsync-Slave einmal)
INTR_SET (mit i_EVENT = 11) *)
CANsync_MODE_MA
```

\*) FB fällt weg, wenn am FB BACI\_INIT der Eingang i\_EVENT = 11 wird

Wird der b maXX 4400 mit b maXX PLC (BM4-O-PLC-01) und dem Optionsmodul CANsync-Slave (BM4-O-CAN-05; Dip-Schalter 9 = ON) verwendet, muss die CANsync-Slave-Anschaltung auf dem Optionsmodul CANsync-Slave initialisiert werden.

Anschließend wird der aktive Betrieb auf der CANsync-Slave-Anschaltung freigegeben.

Benötigte FBs und deren Reihenfolge bei der Initialisierung der CANsync-Slave-Anschaltung auf dem Optionsmodul CANsync-Slave:

```
CANsync_INIT
CANsync_PD_CFG_SL
CANsync_PD_CFG_READ_SL (je weiteren CANsync-Slave einmal)
INTR_SET (mit i_EVENT = 11) *)
CANsync_MODE_SL
```

\*) FB fällt weg, wenn am FB BACI\_INIT der Eingang i\_EVENT = 11 wird

#### Hinweise zur Prozessdatenkommunikation

Die FBs der CANsync-Prozessdatenkommunikation werden in einer POE platziert, die der CANsync Event-Task (Event 11 - SYNC-Signal 1 Optionsmodul) zugeordnet ist. Die FBs der Prozessdatenkommunikation müssen nach dem Aufruf der Event-Task sofort aufgerufen werden, damit Sollwerte und/oder Istwerte noch im gleichen Aufruf der Event-Task gesendet und/oder empfangen werden.

Wird der b maXX 4400 mit b maXX PLC (BM4-O-PLC-01) und dem Optionsmodul CANsync-Master (BM4-O-CAN-06) als CANsync-Master verwendet, muss der FB

```
CANsync_PD_COMM_MA
```

aufgerufen werden.

Wird der b maXX mit b maXX PLC (BM4-O-PLC-01) und dem Optionsmodul CANsync-Slave (BM4-O-CAN-05; Dip-Schalter 9 = ON) als CANsync-Slave für b maXX PLC verwendet, muss der FB

CANsync\_PD\_COMM\_SL

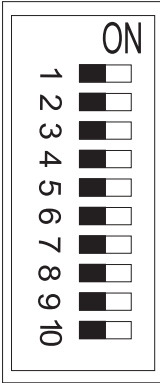
aufgerufen werden.

### 3.1.3 Einstellungen und Anzeigen der Optionsmodule

#### 3.1.3.1 Beschreibung der Dip-Schalter auf Optionsmoul CANsync-Slave für b maXX

Das Optionsmodul CANsync-Slave für b maXX PLC wird mittels Dip-Schalter vorkonfiguriert. Dabei werden eingestellt:

- CANsync-Slavenummer (Dip-Schalter 4 bis 8)
- Umschaltung zwischen CANsync-Slave für b maXX **PLC** und CANsync-Slave für b maXX **Regler** (Dip-Schalter 9)

	<p>Dip-Schalter: <b>3 2 1</b>                      reserviert, bzw. Baudrate, wenn Dip 9 = OFF</p>
	<p>Dip-Schalter: <b>8 7 6 5 4</b>                      CANsync-Slavenummer 0 0 0 0 0 0                      CANsync-Slavenummer 1 0 0 0 1 0                      CANsync-Slavenummer 2 .... 1 1 1 1 1                      CANsync-Slavenummer 31</p>
	<p>Dip-Schalter: <b>9</b> OFF                      CANsync-Slave für b maXX <b>Regler</b> ON                      CANsync-Slave für b maXX <b>PLC</b></p>
	<p>Dip-Schalter: <b>10</b>                      reserviert, muss auf OFF stehen</p>

## Beispiel 1: CANsync-Slave für b maXX PLC

	Dip-Schalter: <b>3 2 1</b> reserviert
	Dip-Schalter: <b>8 7 6 5 4</b> 0 0 1 1 0                CANsync-Slavenummer 6
	Dip-Schalter: <b>9</b> ON                         CANsync-Slave für b maXX PLC
	Dip-Schalter: <b>10</b> reserviert, muss auf OFF stehen

## Beispiel 2: CANsync-Slave für b maXX Regler

In dieser Verwendung des Optionsmoduls muss mit Dip-Schalter 1 bis 3 die Baudrate eingestellt werden (in der Abbildung 500 kBit/s):

	Dip-Schalter: <b>3 2 1</b> Baudrate 0 0 0                    reserviert 0 0 1                    reserviert 0 1 0                    reserviert 0 1 1                    125 kBit/s 1 0 0                    250 kBit/s 1 0 1                    500 kBit/s 1 1 0                    1 MBit/s 1 1 1                    reserviert
	Dip-Schalter: <b>8 7 6 5 4</b> 0 0 1 0 0                CANsync-Slavenummer 4
	Dip-Schalter: <b>9</b> OFF                        CANsync-Slave für b maXX Regler
	Dip-Schalter: <b>10</b> reserviert, muss auf OFF stehen

**HINWEIS**

Dip-Schalter 9 = ON:

Dieses Modul ist ein Optionsmodul CANsync-Slave für b maXX PLC

Dip-Schalter 9 = OFF:

Dieses Modul ist ein Optionsmodul CANsync-Slave für b maXX Regler

siehe auch ► Betriebsanleitung Optionsmodul CANsync-Slave für b maXX ◀, Abschnitt „Betrieb des Optionsmoduls CANsync-Slave für b maXX Regler“.

### 3.1.4 Beschreibung der LEDs

#### 3.1.4.1 Optionsmodul CANsync-Master

Die RJ45-Buchse X1 hat 2 LEDs (grün und rot), nachfolgend H1 und H2 genannt. Die LEDs haben während der Initialisierung und während des Betriebs unterschiedliche Bedeutungen.

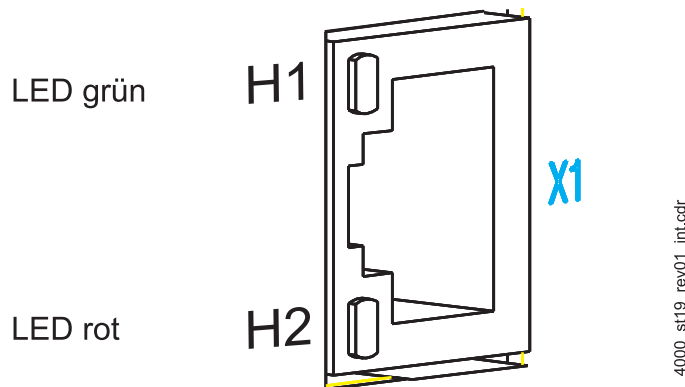


Abbildung 4: Vergrößerte Darstellung der LEDs an der Frontplatte

#### Einschalten und Initialisierung

Nach dem Einschalten blinken die LEDs nacheinander, zuerst die grüne, dann die rote, kurz auf.

Anschließend wird das Optionsmodul CANsync-Master initialisiert. Dabei wird folgendes Muster angezeigt:

Start Initialisierung	H1 ein, H2 aus
Initialisierung läuft	H1 und H2 ein
Ende Initialisierung	H1 ein, H2 aus
Initialisierung fertig	H1 und H2 aus

Damit ist die Initialisierung des Optionsmoduls CANsync-Master abgeschlossen.

Trat bei der Initialisierung ein Fehler auf, blinkt die LED H2.

Um die Fehlerursache zu beseitigen, siehe [▶ Betriebsanleitung Optionsmodul CANsync-Master \(BM4-O-CAN-06\)](#) [◀ Abschnitt Störungssuche und Störungsbeseitigung](#).

#### Betrieb des Optionsmoduls CANsync-Master

Nach der Initialisierung des Optionsmoduls CANsync-Master kann ein Applikationsprogramm auf der b maXX PLC die Konfigurierung des Optionsmoduls vornehmen.

Das Optionsmodul CANsync-Master zeigt jetzt mit H1 = ON (ein) an, dass es auf die Konfigurierung durch die PLC wartet. Beim Optionsmodul b maXX PLC wird diese Konfigurierung des Optionsmoduls CANsync-Master auch „Initialisierung der CANsync-Master-Anschaltung auf dem Optionsmodul CANsync-Master für b maXX PLC“ genannt.

Zur Konfigurierung siehe ▶ Betriebsanleitung b maXX PLC ◀ und ▶ Applikationshandbuch b maXX PLC ◀.

Nach der Konfigurierung des Optionsmoduls durch ein Applikationsprogramm auf der b maXX PLC haben die LEDs folgende Bedeutung:

H1 (grün) zeigt den Empfang und das Senden von Telegrammen auf dem CANsync-Bus an.

H2 (rot) ist normalerweise aus und blinkt nur im Fehlerfall.

Um die Fehlerursache zu beseitigen, siehe ▶ Betriebsanleitung Optionsmodul CANsync-Master (BM4-O-CAN-06) ◀ Abschnitt Störungssuche und Störungsbeseitigung.

### 3.1.4.2 Optionsmodul CANsync-Slave für b maXX PLC

Die RJ45-Buchsen X1 und X2 haben je 2 LEDs (grün und rot), nachfolgend H1 bis H4 genannt. Die LEDs haben während der Initialisierung und während des Betriebs des Optionsmoduls CANsync-Slave unterschiedliche Bedeutungen.

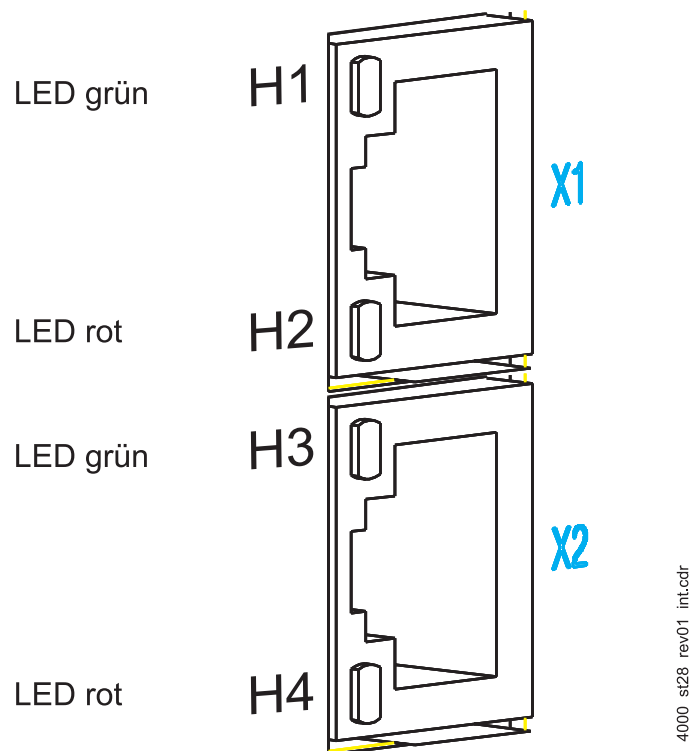


Abbildung 5: Vergrößerte Darstellung der LEDs an der Frontplatte

#### Einschalten und Initialisierung des Optionsmoduls CANsync-Slave für b maXX PLC

Nach dem Einschalten leuchten die LEDs nacheinander, in der Reihenfolge H1 (grün), H2 (rot), H3 (grün), H4 (rot), kurz auf.

Anschließend wird das Optionsmodul CANsync-Slave initialisiert. Dabei wird folgendes Muster angezeigt:

Start Initialisierung	H1 ein, H2 bis H4 aus
Ende Initialisierung	H3 ein, H1, H2 und H4 aus
Initialisierung fertig	H1 bis H4 aus

Damit ist die Basis-Initialisierung des Optionsmoduls CANsync-Slave abgeschlossen. Trat bei der Initialisierung ein Fehler auf, blinken die LEDs H2 und H4 synchron.

Um die Fehlerursache zu beseitigen, siehe ► Betriebsanleitung Optionsmodul CANsync-Slave (BM4-O-CAN-05) ◀, Abschnitt Störungssuche und Störungsbeseitigung.

### Betrieb des Optionsmoduls CANsync-Slave für b maXX PLC

Nach der Initialisierung des Optionsmoduls CANsync-Slave kann ein Applikationsprogramm auf der b maXX PLC die Konfigurierung des Optionsmoduls vornehmen.

Das Optionsmodul CANsync-Slave zeigt jetzt mit LED H1 = ON (ein) an, dass es auf die Konfigurierung durch die PLC wartet.

Beim Optionsmodul b maXX PLC wird diese Konfigurierung des Optionsmoduls CANsync-Slave auch „Initialisierung der CANsync-Slave-Anschaltung auf dem Optionsmodul CANsync-Slave für b maXX PLC“ genannt.

Weitere Informationen finden Sie im ► Applikationshandbuch b maXX PLC ◀.

Nach der Konfigurierung des Optionsmoduls durch ein Applikationsprogramm auf der b maXX PLC haben die LEDs folgende Bedeutung:

- H1 (grün) zeigt den Empfang und das Senden von Telegrammen auf dem CANsync-Bus an.
- H2 (rot) zeigt den Ausfall des Synchronisier-Signals auf dem CANsync-Bus an.
- H3 (grün) zeigt durch Blinken die Dauer eines SYNC-Intervals auf dem CANsync-Bus nach folgendem Muster an:

Muster	SYNC-Interval (Baudrate)
1-mal Blinken, dann Pause	1 ms (1 MBit/s)
2-mal Blinken, dann Pause	2 ms (500 kBit/s)
4-mal Blinken, dann Pause	4 ms (250 kBit/s)
8-mal Blinken, dann Pause	8 ms (125 kBit/s)

H4 (rot) ist normalerweise aus und blinkt nur im Fehlerfall.

Um die Fehlerursache zu beseitigen, siehe Betriebsanleitung Optionsmodul CANsync-Slave (BM4-O-CAN-05) Abschnitt Störungssuche und Störungsbeseitigung.

#### 3.1.4.3 Einschalten und Initialisierung des Optionsmoduls CANsync-Slave für b maXX Regler

Nach dem Einschalten leuchten die LEDs nacheinander, in der Reihenfolge H1 (grün), H2 (rot), H3 (grün), H4 (rot), kurz auf.

Anschließend wird das Optionsmodul CANsync-Slave initialisiert. Dabei wird folgendes Muster angezeigt:



Start Initialisierung	H1 ein, H2 bis H4 aus
Ende Initialisierung	H3 ein, H1, H2 und H4 aus
Initialisierung fertig	H1 bis H4 aus

Damit ist die Basis-Initialisierung des Optionsmoduls CANsync-Slave abgeschlossen. Trat bei der Initialisierung ein Fehler auf, blinkt die LED H2.

Um die Fehlerursache zu beseitigen, siehe ► Betriebsanleitung Optionsmodul CANsync-Slave (BM4-O-CAN-05)◄, Abschnitt Störungssuche und Störungsbeseitigung.

### Betrieb des Optionsmoduls CANsync-Slave für b maXX Regler

Nach der Initialisierung des Optionsmoduls CANsync-Slave wird das Optionsmodul entsprechend der in WinBASS II für das Optionsmodul eingestellten Parameter konfiguriert.

Nach der Konfigurierung des Optionsmoduls haben die LEDs folgende Bedeutung:

- H1 (grün) zeigt den Empfang und das Senden von Telegrammen auf dem CANsync-Bus an.
- H2 (rot) zeigt den Ausfall des Synchronisier-Signals auf dem CANsync-Bus an.
- H3 (grün) zeigt durch Blinken die Dauer eines SYNC-Intervals auf dem CANsync-Bus nach folgendem Muster an:

Muster	SYNC-Interval (Baudrate)
1-mal Blinken, dann Pause	1 ms (1 MBit/s)
2-mal Blinken, dann Pause	2 ms (500 kBit/s)
4-mal Blinken, dann Pause	4 ms (250 kBit/s)
8-mal Blinken, dann Pause	8 ms (125 kBit/s)

H4 (rot) ist normalerweise aus und blinkt nur im Fehlerfall.

Um die Fehlerursache zu beseitigen, siehe ► Betriebsanleitung Optionsmodul CANsync-Slave (BM4-O-CAN-05)◄, Abschnitt Störungssuche und Störungsbeseitigung.

H2 (rot) und H4 (rot) blinken im Wechsel:  
über die Dip-Schalter 1 bis 3 wurde eine Baudrate eingestellt, die vom Optionsmodul nicht unterstützt wird.

H2 (rot) und H4 (rot) ein:  
das Optionsmodul ist konfiguriert und wartet auf das Starten des CANsync-Busses durch den CANsync-Master.

### 3.2 Detaillierte Informationen zum CANsync



#### HINWEIS

Dieses Kapitel enthält detaillierte Informationen zum CANsync. Diese Informationen werden nicht benötigt, wenn zur Programmierung die Funktionsbausteine der Bibliothek CANsync\_PLCC01\_20bd00 oder höher verwendet werden.

$t_{RSPTO}$  - Response Timeout: Zeit innerhalb der der CANsync-Slave während der Initialisierung eine Antwort senden muss.

Baudrate	$t_{RSPTO}$
1 MBit/s	300 $\mu$ s
500 kBit/s	600 $\mu$ s
250 kBit/s	1100 $\mu$ s
125 kBit/s	2100 $\mu$ s



#### HINWEIS

Alle folgenden Zeitangaben beziehen sich auf eine Baudrate von 1 MBit/s.

#### 3.2.0.1 Anlauf und Initialisierung

Das Anlaufverhalten gliedert sich in die folgenden Schritte:

##### ⇒ Initialisierung mit FB CANsync\_INIT

Nachdem der FB CANsync\_INIT erfolgreich durchlaufen wurde gibt der CANsync-Master das SYNC-Signal im Zeitraster von 1 ms und das Aktionskommando „SYNC-Modus“ mit dem Dateninhalt: „SYNC-Betrieb einschalten“ an alle CANsync-Slaves aus. Die CANsync-Slaves beginnen ihre Regelungs-Task auf das SYNC-Signal zu synchronisieren.

Der CANsync-Master fordert von jedem CANsync-Slave, den er am CANsync-Bus erwartet über das Parameter-Kommando „Parameter Lesen“ dessen Statuswort an.

Der CANsync-Slave muss innerhalb von  $t_{RSPTO}$  mit der Parameter-Antwort antworten. Der CANsync-Master überwacht, ob der CANsync-Slave innerhalb dieser Zeit antwortet.

Der CANsync-Master gibt während der Synchronisierung noch keine Sollwerte aus.

### ⇒ Übergang in den aktiven Betrieb

Mit dem FB CANsync\_Mode (Eingang x\_CANsync\_RUN = TRUE) startet man direkt den aktiven Betrieb. Es werden dann auch Sollwerte und Istwerte übertragen.

Es ist dabei zu beachten, dass die CANsync-Slaves einige Zeit (Sekunden) benötigen, um sich auf das SYNC-Signal zu synchronisieren. Das Statuswort des Optionsmoduls CANsync-Slave für b maXX Regler (BM4-O-CAN-05; Dip-Schalter 9 = OFF) enthält die Information (Bit 15 = TRUE), ob er synchronisiert ist.

Wenn die Übertragung synchroner Lagesollwerte gewünscht wird, darf das Applikationsprogramm die Maschine erst starten, wenn alle CANsync-Slaves synchronisiert sind (= Zustand "synchronisiert"). Andere Sollwerte, bzw. andere Betriebsarten sind auch schon davor zulässig.

### 3.2.0.2 Zustand „synchronisiert“

---

Im Zustand „synchronisiert“ laufen alle Antriebe (CANsync-Master, CANsync-Slaves) im gleichen CANsync-Intervall, d. h. alle CANsync-Slaves bearbeiten gleichzeitig ihre Regelungs-Task und übernehmen die Sollwerte zeitgleich.

Der CANsync-Master sendet seine Aufträge in definierter Reihenfolge mit zugeordneten Zeitfenstern. Die CANsync-Slaves müssen ihre Antworten diesem Schema anpassen (siehe [►Überblick◄](#) ab Seite 15 und [►Abbildung 2◄](#) auf Seite 17).

Der CANsync-Master muss bis spätestens 350 µs nach dem SYNC-Signal das nächste Sollwert-Telegramm 1 senden.

### 3.2.0.3 Sollwert-Verlust bei Optionsmodul CANsync-Slave für Regler

---

Wird in einem CANsync-Intervall kein neuer Sollwert empfangen, führt der CANsync-Slave eine Extrapolation mit dem zuletzt empfangenen Sollwert durch.

### 3.2.1 Aufbau der Telegramme

---

#### ⇒ CANsync-Telegramme

Die Telegrammlängen sind variabel (0 bis 8 Datenbytes). Sie ergeben sich aus der vorangegangenen Auflistung und hängen bei manchen Telegrammen auch von dem jeweiligen Betriebszustand ab.

#### ⇒ Datenformat

Die Daten werden in den Telegrammen im Intelformat (Lowbyte / Highbyte) abgelegt.

#### ⇒ Statuswort

Das Statuswort eines CANsync-Slaves gibt dessen Antriebszustand an. Im obersten Bit (15) muss der SYNC-Zustand angezeigt werden. Wenn das Bit gesetzt ist, ist der CANsync-Slave synchronisiert.

### 3.2.1.1 Sollwertkanäle

#### Sollwertkanal 1

Im Sollwertkanal 1 sendet der CANsync-Master das Sollwert-Telegramm 1.

Der CANsync-Master überträgt damit an alle CANsync-Slaves einen oder mehrere (max. 4) Sollwerte. Die Nummer NNNNNNN im Identifier gibt an, welcher CANsync-Slave im Anschluss an das Sollwert-Telegramm 1 sein Istwert-Telegramm 1 senden muss.

<IDENTIFIER><SOLLWERTE>		
<IDENTIFIER>	::= 0010NNNNNNN	Identifier Sollwertkanal 1
<SOLLWERTE>	::= W_SOLL   DW_SOLL   W_DW_SOLL	
<W_SOLL>	::= <W_SOLL_1>   <W_SOLL_1..2>   <W_SOLL_1..3>   <W_SOLL_1..4>	nur Wort-Sollwerte
<DW_SOLL>	::= <DW_SOLL_1>   <DW_SOLL_1..2>	nur DWort-Sollwerte
<W_DW_SOLL>	::= <DW_SOLL_1><W_SOLL_3>   <DW_SOLL_1><W_SOLL_3><W_SOLL_4>	Wort- und DWort-Sollwerte
<W_SOLL_1>	::= <Word>	Wort-Sollwert 1 = CAN-DB 0..1
<W_SOLL_2>	::= <Word>	Wort-Sollwert 2 = CAN-DB 2..3
<W_SOLL_3>	::= <Word>	Wort-Sollwert 3 = CAN-DB 4..5
<W_SOLL_4>	::= <Word>	Wort-Sollwert 4 = CAN-DB 6..7
<DW_SOLL_1>	::= <DWord>	DWort-Sollwert 1 = CAN-DB 0..3
<DW_SOLL_2>	::= <DWord>	DWort-Sollwert 2 = CAN-DB 4..7 CAN-DB: CAN-Daten-Byte

#### Sollwertkanal 2

Im Sollwertkanal 2 sendet der CANsync-Master das Sollwert-Telegramm 2.

Der CANsync-Master überträgt damit auf diesem Sollwertkanal bis zu 4 zusätzliche Sollwerte.

Der Aufbau und die Funktion entsprechen dem Sollwertkanal 1. Das Sollwert-Telegramm 2 unterscheidet sich im Identifier und der Zuordnung der Sollwerte vom Sollwert-Telegramm 1. Dem Sollwertkanal 2 sind die Wort-Sollwerte 5..8 und die Doppelwort-Sollwerte 3..4 zugeordnet. Die Nummer NNNNNNN im Identifier gibt die Slave-Nummer des CANsync-Slaves an, der mit dem Istwert-Telegramm 2 antwortet.

<IDENTIFIER>	::= 0011NNNNNNN	Identifier Sollwertkanal 2
--------------	-----------------	----------------------------

### 3.2.1.2 Istwertkanäle

#### Istwertkanal 1

Im Istwertkanal 1 sendet der CANsync-Slave das Istwert-Telegramm 1.

Der Identifier der Sollwert-Telegramm 1 gibt an, welcher CANsync-Slave sein Istwert-Telegramm 1 als direkte Antwort auf das Sollwert-Telegramm 1 schicken darf.

<IDENTIFIER><ISTWERTE>		
<IDENTIFIER>	::= 0110NNNNNNN	NNNNNNN = CANsync-Slave-Nummer
<ISTWERTE>	::= W_IST   DW_IST   W_DW_IST	
<W_IST>	::= <W_IST_1>   <W_IST_1..2>   <W_IST_1..3>   <W_IST_1..4>	nur Wort-Istwerte
<DW_IST>	::= <DW_IST_1>   <DW_IST_1..2>	nur DWort-Istwerte
<W_DW_IST>	::= <DW_IST_1><W_IST_3>   <DW_IST_1><W_IST_3><W_IST_4>	Wort- und DWort-Istwerte
<W_IST_1>	::= <Word>	Wort-Istwert 1 = CAN-DB 0..1
<W_IST_2>	::= <Word>	Wort-Istwert 2 = CAN-DB 2..3
<W_IST_3>	::= <Word>	Wort-Istwert 3 = CAN-DB 4..5
<W_IST_4>	::= <Word>	Wort-Istwert 4 = CAN-DB 6..7
<DW_IST_1>	::= <DWord>	DWort-Istwert 1 = CAN-DB 0..3
<DW_IST_2>	::= <DWord>	DWort-Istwert 2 = CAN-DB 4..7
		CAN-DB: CAN-Daten-Byte

#### Istwertkanal 2

Im Istwertkanal 2 sendet der CANsync-Slave das Istwert-Telegramm 2.

Der Aufbau und die Funktion des Telegramms entsprechen dem Istwert-Telegramm 1. Es sind maximal 4 weitere Istwerte übertragbar. Das Istwert-Telegramm 2 unterscheidet sich im Identifier und der Zuordnung der Istwerte. Dem Istwertkanal 2 sind die Wort-Istwerte 5..8 und die Doppelwort-Istwerte 3..4 zugeordnet.

<IDENTIFIER>	::= 0111NNNNNNN	NNNNNNN = CANsync-Slave-Nummer
--------------	-----------------	--------------------------------

### 3.2.1.3 Kommandokanal & Antwortkanal

Der Kommandokanal und der zugehörige Antwortkanal bestehen funktional aus drei Gruppen von Telegrammen, wobei in einem CANsync-Intervall immer nur je ein Kommando/Antwort vorkommen kann.

- ⇒ Aktionskommandos dienen der Initialisierung und Steuerung der CANsync-Slaves und werden an einen oder mehrere CANsync-Slaves gesendet, ohne dass vom CANsync-Master eine Antwort erwartet wird.
- ⇒ Parameterkommandos werden zum Lesen oder Schreiben eines Parameters verwendet und sind immer an einen CANsync-Slave gerichtet. Der CANsync-Master erwartet immer eine Antwort.
- ⇒ Up/Download-Kommandos dienen der Übertragung größerer Datenmengen (Programmcode, Datensatz) und sind immer an einen CANsync-Slave gerichtet. Der CANsync-Master erwartet immer eine Antwort.

Parameter- und Up-/Download-Kommandos werden mit dem gleichen Identifier gesendet.

#### Aktionskommando

Ein Aktionskommando wird vom CANsync-Master an einen einzelnen CANsync-Slave oder eine Gruppe von CANsync-Slaves gesendet. Die Auswahl erfolgt über eine Bitleiste (SLAVE\_GROUP) in der jedem CANsync-Slave ein Bit zugeordnet ist. Wenn das Bit gesetzt ist, muss der zugeordnete CANsync-Slave dieses Kommando ausführen. In einem Broadcast-Kommando an alle CANsync-Slaves sind alle Bits in der Bitleiste gesetzt.

Die unterschiedlichen Kommandos unterscheiden sich im Datenbyte COMMAND. In Abhängigkeit vom Kommando folgen unterschiedlich viele weitere Datenbytes, die Daten zum Kommando enthalten.

<IDENTIFIER><SLAVE_GROUP><COMAND><DATA>		
<IDENTIFIER>	::= 00000010000	Identifier Aktionskommando
<SLAVE_GROUP>	::= <DWord>	Slave-Bits 0..30 = CAN-DB 0..3
<COMMAND>	::= <Byte> 1 = Steuerwort Schreiben	= CAN-DB 4
<DATA>	::= <DATA_1>	vom Kommando abhängige Daten
<DATA_1>	::= <res><Wert>	
<res>	::= <Byte>	CAN-DB 5
<Wert>	::= <Word>	Steuerwort = CAN-DB 6..7 CAN-DB: CAN-Daten-Byte

#### Parameter Lesen

Mit dem Parameter Lesen Kommando, fordert der CANsync-Master einen Parameter des CANsync-Slaves zum Lesen an. Der CANsync-Slave erkennt an der Telegrammlänge (= 4 Datenbytes), dass es sich um ein Parameter-Lese-Kommando handelt. Der CANsync-Slave muss die Elementauswahl nicht unterstützen, er antwortet dann immer mit dem aktuellen Datenwert.

Der CANsync-Slave muss innerhalb der vorgegeben Antwortzeit  $t_{RSPT0}$  antworten. Falls er bis dahin noch nicht den Auftrag beenden kann, antwortet er mit der Parameter-Antwort, in der die Parameternummer des Auftrags eingetragen ist und das BUSY-Bit gesetzt ist. Wenn der CANsync-Master das nächste Mal das Parameter-Lese-Kommando auf den Parameter wiederholt und der CANsync-Slave den Auftrag inzwischen bearbeitet hat, dann antwortet er mit den angeforderten Daten und das BUSY-Bit ist auf Null gesetzt.

Falls der Leseauftrag nicht bearbeitet werden kann oder ein Fehler auftritt, setzt der CANsync-Slave das ERR-Bit und gibt in den Datenbytes einen Fehlercode an.

Die Parameternummer kann von 0 bis 4095 gehen.

Auftrag:

<IDENTIFIER><CONTROL><PARA_NUM_L><SUB-ADRESSE>		
<IDENTIFIER>	::=	1010NNNNNNN
		NNNNNNN = CANsync-Slave-Nummer
<CONTROL>	::=	<P><ELEMENT><PARA_NUM_H>
<P>	::=	<Bit7>
		CAN-DB 0
		0 = Kennung: Parameter - Kommando
<ELEMENT>	::=	<Bit6..4>
		Elementauswahl des Parameters
<PARA_NUM_H>	::=	<Bit3..0>
<PARA_NUM_L>	::=	<Byte>
		Bit11..8 der Parameter-Nr.
		Bit7..0 der Parameter-Nr. = CAN-DB 1
<SUB-ADRESSE>	::=	<Word>
		Sub-Slave-Adresse = CAN-DB 2..3

Antwort:

<IDENTIFIER><STATUS><PARA_NUM_L><DATA><SUB-ADRESSE>		
<IDENTIFIER>	::=	1011NNNNNNN
		NNNNNNN = CANsync-Slave-Nummer
<STATUS>	::=	<P><BUSY><ERR> <FREI><PARA_NUM_H>
<P>	::=	<Bit7>
		CAN-DB 0
		0 = Kennung: Parameter - Antwort
<BUSY>	::=	<Bit6>
		0 = Antwort gültig, 1 = Auftrag wird bearbeitet
<ERR>	::=	<Bit5>
<FREI>	::=	<Bit4>
		0 = kein Fehler, 1 = Fehler frei
<PARA_NUM_H>	::=	<Bit3..0>
<PARA_NUM_L>	::=	<Byte>
		Bit11..8 der Parameter-Nr.
		Bit7..0 der Parameter-Nr. = CAN-DB 1
<DATA>	::=	<2_BYTE>   <4_BYTE>   <ERR_CODE>
<2_BYTE>	::=	<Word>
		Wort-Parameter = CAN-DB 2..3
<4_BYTE>	::=	<Dword>
		DWort-Parameter = CAN-DB 2..5
<ERR_CODE>	::=	<Word>
		2 Byte Fehlercode = CAN-DB 2..3
<SUB-ADRESSE>	::=	<Word>
		Sub-Slave-Adresse = CAN-DB 4..5 / 6..7
		CAN-DB: CAN-Daten-Byte

### Parameter Schreiben

Mit dem Parameter Schreiben Kommando, schreibt der CANsync-Master einen Parameter in einem CANsync-Slave. Der CANsync-Slave erkennt an der Telegrammlänge (= 6 oder 8 Datenbytes), ob es sich um ein Wort oder einen Doppelwort-Parameter handelt. Beim Schreiben ist zur Zeit nur das Element 7, der Parameterwert zulässig.

Der CANsync-Slave muss innerhalb der vorgegeben Antwortzeit  $t_{RSPT0}$  antworten. Falls er bis dahin noch nicht den Auftrag beenden kann, antwortet er mit der Parameter-Antwort, in der die Parameternummer des Auftrags eingetragen ist und das BUSY-Bit gesetzt ist. Wenn der CANsync-Master das nächste Mal das Parameter-Schreiben-Kommando auf den Parameter wiederholt und der CANsync-Slave den Auftrag inzwischen bearbeitet hat, dann antwortet er mit der Parameternummer und das BUSY-Bit ist auf Null gesetzt.

Falls der Schreibauftrag nicht bearbeitet werden kann oder ein Fehler auftritt, setzt der CANsync-Slave das ERR-Bit und gibt in den Datenbytes einen Fehlercode an.

Die Parameternummer kann von 0 bis 4095 gehen.

Auftrag:

<IDENTIFIER><CONTROL><PARA_NUM_L><DATA><SUB-ADRESSE>			
<IDENTIFIER>	::=	1010NNNNNNN	NNNNNNN = CANsync-Slave- Nummer
<CONTROL>	::=	<P><ELEMENT><PARA_NUM_H>	CAN-DB 0
<P>	::=	<Bit7>	0 = Kennung: Parameter- Kommando
<ELEMENT>	::=	<Bit6..4>	Elementauswahl des Parameters
<PARA_NUM_H>	::=	<Bit3..0>	Bit11..8 der Parameter-Nr.
<PARA_NUM_L>	::=	<Byte>	Bit7..0 der Parameter-Nr. = CAN-DB 1
<DATA>	::=	<2_BYTE>   <4_BYTE>	
<2_BYTE>	::=	<Word>	Wort-Parameter = CAN-DB 2..3
<4_BYTE>	::=	<Dword>	DWort-Parameter = CAN-DB 2..5
<SUB-ADRESSE>	::=	<Word>	Sub-Slave-Adresse = CAN-DB 4..5/6..7

Antwort:

<IDENTIFIER><STATUS><PARA_NUM_L><DATA><SUB-ADRESSE>			
<IDENTIFIER>	::=	1011NNNNNNN	NNNNNNN = CANsync-Slave- Nummer
<STATUS>	::=	<P><BUSY><ERR> <FREI> <PARA_NUM_H>	CAN-DB 0
<P>	::=	<Bit7>	0 = Kennung: Parameter- Antwort
<BUSY>	::=	<Bit6>	0 = Auftrag fertig, 1 = Auftrag wird bearbeitet
<ERR>	::=	<Bit5>	0 = kein Fehler, 1 = Fehler noch nicht belegt
<FREI>	::=	<Bit4>	
<PARA_NUM_H>	::=	<Bit3..0>	Bit11..8 der Parameter-Nr.
<PARA_NUM_L>	::=	<Byte>	Bit7..0 der Parameter-Nr. = CAN-DB 1
<DATA>	::=	<0_BYTE>   <ERR_CODE>	
<0_BYTE>	::=	keine Daten-Bytes wenn fehlerfrei	



<ERR_CODE>	::=	<Word>	2 Byte Fehlercode = CAN-DB 2..3
<SUB-ADRESSE>	::=	<Word>	Sub-Slave-Adresse = CAN-DB 2..3/4..5 CAN-DB: CAN-Daten-Byte

### Beginn eines Up- oder Downloads

Mit der Upload- oder Download-Übertragung können größere zusammenhängende Datenbereiche vom CANsync-Master zum CANsync-Slave oder umgekehrt übertragen werden.

Die Konfiguration der Übertragung erfolgt mit einem Initialisierungs-Telegramm.

Der CANsync-Slave muss innerhalb der vorgegeben Antwortzeit  $t_{RSPTO}$  antworten. Falls er bis dahin noch nicht den Auftrag beenden kann, antwortet er mit der Up/Download-Antwort, in der das BUSY-Bit gesetzt ist. Wenn der CANsync-Master das nächste Mal das Up/Download-Telegramm wiederholt und der CANsync-Slave den Auftrag inzwischen bearbeitet hat, dann antwortet er mit Antwort, in der das BUSY-Bit auf Null gesetzt ist.

Falls der Upload oder Download-Auftrag nicht bearbeitet werden kann oder ein Fehler auftritt, setzt der CANsync-Slave das ERR-Bit und gibt in den Datenbytes einen Fehlercode an.

Die Startadresse ist eine Doppelwort-Adresse. Die maximale Upload oder Downloadlänge ist 4096 Bytes. Größere Datenbereiche müssen über mehrere Up/Download-Initialisierungen übertragen werden. Als Option kann eine Sub-Slave-Adresse angegeben werden. Diese Adresse gibt an, dass der folgenden Up/Download sich nicht direkt auf den angesprochenen CANsync-Slave bezieht, sondern dass die Up/Download-Telegramme an einen Sub-Slave weitergereicht werden. Diese Sub-Slave-Adresse bleibt bis zum Ende des Up/Downloads gültig. Für den nächsten Up/Download-Auftrag muss die Adresse wieder neu angegeben werden. Wenn die Sub-Adresse gleich Null ist, wird direkt der CANsync-Slave angesprochen und kein Sub-Slave.

Auftrag:

<IDENTIFIER><CONTROL><OFFSET_L><ADRESSE><SUB-ADRESSE>		
<IDENTIFIER>	::=	1010NNNNNNN NNNNNNN = CANsync-Slave- Nummer
<CONTROL>	::=	<L><U/D><MODE><OFFSET_H> CAN-DB 0
<L>	::=	<Bit7> 1 = Kennung: Up/Download- Auftrag
<U/D>	::=	<Bit6> 0 = Upload, 1 = Download
<MODE>	::=	<Bit5..4> 01 = Initialisierung
<OFFSET_H>	::=	<Bit3..0> Blocklänge in Bytes Bit 11..8 <sup>1)</sup>
<OFFSET_L>	::=	<Byte> Blocklänge in Bytes Bit 7..0 = CAN-DB 1 <sup>1)</sup>
<ADRESSE>	::=	<Dword> absolute Startadresse = CAN-DB 2..5
<SUB-ADRESSE>	::=	<Word> Sub-Slave-Adresse = CAN-DB 6..7

<sup>1)</sup> Bei einem Upload-Auftrag kann die Länge statt durch den CANsync-Master auch durch den CANsync-Slave festgelegt werden. In diesem Fall wird die Länge in der Antwort in <OFFSET\_H> und <OFFSET\_L> angegeben. Ansonsten stehen diese Werte in der Antwort auf Null.

Antwort:

<IDENTIFIER><STATUS><OFFSET_L><DATA>			
<IDENTIFIER>	::=	1011NNNNNNN	NNNNNNN = CANsync-Slave- Nummer
<STATUS>	::=	<L><BUSY><ERR><FREI><OFFSET_H>	CAN-DB 0
<L>	::=	<Bit7>	1 = Kennung: Up/Download- Antwort
<BUSY>	::=	<Bit6>	0 = Auftrag fertig, 1 = Auftrag wird bearbeitet
<ERR>	::=	<Bit5>	0 = kein Fehler, 1 = Fehler
<FREI>	::=	<Bit4>	noch nicht belegt
<OFFSET_H>	::=	<Bit3..0>	Blocklänge in Bytes Bit 11..8 <sup>1)</sup>
<OFFSET_L>	::=	<Byte>	Blocklänge in Bytes Bit 7..0 = CAN-DB 1 <sup>1)</sup>
<DATA>	::=	<0_BYTE>   <ERR_CODE>	
<0_BYTE>	::=	keine Daten-Bytes, wenn fehlerfrei	
<ERR_CODE>	::=	<Word>	Bit15..0 des Fehlercodes = CAN-DB 2..3
			CAN-DB: CAN-Daten-Byte

### Laufender Upload & Ende eines Uploads

Der Uploadvorgang besteht aus aufeinanderfolgenden Upload-Telegrammen in denen der CANsync-Master von der in der Initialisierung eingestellten Startadresse ab, aufeinanderfolgende Datenblöcke anfordert. Die Offsetadresse steigt fortlaufend als Byteadresse. Mit jedem Telegramm werden 6 Bytes Nutzdaten übertragen. Das heißt das erste Telegramm beginnt mit der Offsetadresse 0, das zweite fordert die Daten mit der Offsetadresse 6 an, usw.

Der CANsync-Slave muss innerhalb der vorgegeben Antwortzeit  $t_{RSPT0}$  antworten. Falls er bis dahin noch nicht den Auftrag beenden kann, antwortet er mit der Upload-Antwort, in der die Offsetadresse des Auftrags eingetragen ist und das BUSY-Bit gesetzt ist. Wenn der CANsync-Master das nächste Mal das Upload-Telegramm wiederholt und der CANsync-Slave den Auftrag inzwischen bearbeitet hat, dann antwortet er mit dem angeforderten Datenblock und das BUSY-Bit ist auf Null gesetzt.

Im letzten Telegramm ist MODE auf 11 gesetzt. Der CANsync-Slave überprüft, ob er auch am Ende des eingestellten Datenbereich angekommen ist und sendet den letzten Datenblock immer mit 6 Datenbytes. Wenn der zuladende Speicherbereich nicht mehr so viele Daten umfasst wird mit irrelevanten Daten aufgefüllt. Falls der CANsync-Slave nicht am Ende angekommen ist, antwortet er mit gesetztem ERR-Bit und einem Fehlercode.

Auch wenn der Upload-Auftrag nicht bearbeitet werden kann, oder der CANsync-Slave eine Lücke in den angeforderten Offsetadressen feststellt, setzt der CANsync-Slave das ERR-Bit und gibt in den Datenbytes einen Fehlercode an. Der CANsync-Master kann gegebenenfalls das ausgefallene Telegramm wiederholen oder er bricht den Upload ab, in dem er MODE auf 01 setzt und als Basisadresse und Blocklänge 0 einträgt.

Auftrag:

<IDENTIFIER><CONTROL><OFFSET_L>			
<IDENTIFIER>	::=	1010NNNNNNN	NNNNNNN = CANsync-Slave- Nummer
<CONTROL>	::=	<UD><U/D><MODE><OFFSET_H>	CAN-DB 0

<UD>	::=	<Bit7>	1 = Kennung: Up/Download-Auftrag
<U/D>	::=	<Bit6>	0 = Upload
<MODE>	::=	<Bit5..4>	10 = Mittelstück 11 = letzter Block
<OFFSET_H>	::=	<Bit3..0>	Bit11..8 der Offset-Adresse
<OFFSET_L>	::=	<Byte>	Bit7..0 der Offset-Adresse = CAN-DB 1
Antwort:			
<IDENTIFIER><STATUS><OFFSET_L><DATA>			
<IDENTIFIER>	::=	1011NNNNNN	NNNNNN = CANsync-Slave-Nummer
<STATUS>	::=	<UD><BUSY><ERR><FREI><OFFSET_H>	CAN-DB 0
<UD>	::=	<Bit7>	1 = Kennung: Up/Download-Antwort
<BUSY>	::=	<Bit6>	0 = Auftrag fertig, 1 = Auftrag wird bearbeitet
<ERR>	::=	<Bit5>	0 = kein Fehler, 1 = Fehler
<FREI>	::=	<Bit4>	noch nicht belegt
<OFFSET_H>	::=	<Bit3..0>	Bit11..8 der Offset-Adresse
<OFFSET_L>	::=	<Byte>	Bit7..0 der Offset-Adresse = CAN-DB 1
<DATA>	::=	<DATEN>   <ERR_CODE>	
<DATEN>	::=	<Word><Word><Word>	6 Byte Daten = CAN-DB 2..7
<ERR_CODE>	::=	<Word>	Fehlercode = CAN-DB 2..3 CAN-DB: CAN-Daten-Byte

### Laufender Download & Ende eines Downloads

Der Downloadvorgang besteht aus aufeinanderfolgenden Download-Telegrammen in denen der CANsync-Master von der in der Initialisierung eingestellten Startadresse ab, aufeinanderfolgende Datenblöcke sendet. Die Offsetadresse steigt fortlaufend als Byteadresse. Das heißt das erste Telegramm beginnt mit der Offsetadresse 0, das zweite sendet die Daten mit der Offsetadresse 6 an, usw.

Der CANsync-Slave muss innerhalb der vorgegeben Antwortzeit  $t_{RSP\ TO}$  antworten. Falls er bis dahin noch nicht den Auftrag beenden kann, antwortet er mit der Download-Antwort, in der die Offsetadresse des Auftrags eingetragen ist und das BUSY-Bit gesetzt ist. Wenn der CANsync-Master das nächste Mal das Download-Telegramm wiederholt und der CANsync-Slave den Auftrag inzwischen bearbeitet hat, dann antwortet er mit dem Antwort-Telegramm, in dem das BUSY-Bit auf Null gesetzt ist.

Im letzten Telegramm ist MODE auf 11 gesetzt und es enthält auch 6 Datenbytes. Der CANsync-Slave darf jedoch nur die Datenbytes übernehmen, die der vorher eingestellten Downloadlänge entsprechen. Falls der CANsync-Slave noch nicht am Ende angekommen ist, antwortet er mit gesetztem ERR-Bit und einem Fehlercode.

Auch wenn der Download-Auftrag nicht bearbeitet werden kann, oder der CANsync-Slave eine Lücke in den gesendeten Offsetadressen feststellt, setzt der CANsync-Slave das ERR-Bit und gibt in den Datenbytes einen Fehlercode an. Der CANsync-Master kann gegebenenfalls das ausgefallene Telegramm wiederholen oder er bricht den Download ab, in dem er MODE auf 01 setzt und als Basisadresse und Blocklänge 0 einträgt.

## 3.2 Detaillierte Informationen zum CANsync

Auftrag:

<IDENTIFIER><CONTROL><OFFSET_L><DATA>		
<IDENTIFIER>	::= 1010NNNNNNN	NNNNNNN = CANsync-Slave-Nummer
<CONTROL>	::= <UD><U/D><MODE><OFFSET_H>	CAN-DB 0
<UD>	::= <Bit7>	1 = Kennung: Up/Download-Auftrag
<U/D>	::= <Bit6>	1 = Download
<MODE>	::= <Bit5..4>	10 = Mittelstück 11 = letzter Block
<OFFSET_H>	::= <Bit3..0>	Bit11..8 der Offset-Adresse
<OFFSET_L>	::= <Byte>	Bit7..0 der Offset-Adresse = CAN-DB 1
<DATA>	::= <Word><Word><Word>	6 Byte Nutzdaten = CAN-DB 2..7

Antwort:

<IDENTIFIER><STATUS><OFFSET_L><DATA>		
<IDENTIFIER>	::= 1011NNNNNNN	NNNNNNN = CANsync-Slave-Nummer
<STATUS>	::= <UD><BUSY><ERR><FREI><ERR_CODE_H>	CAN-DB 0
<UD>	::= <Bit7>	1 = Kennung: Up/Download
<BUSY>	::= <Bit6>	0 = Auftrag fertig, 1 = Auftrag wird bearbeitet
<ERR>	::= <Bit5>	0 = kein Fehler, 1 = Fehler noch nicht belegt
<FREI>	::= <Bit4>	
<OFFSET_H>	::= <Bit3..0>	Bit11..8 der Offset-Adresse
<OFFSET_L>	::= <Byte>	Bit7..0 der Offset-Adresse = CAN-DB 1
<DATA>	::= <0_BYTE>   <ERR_CODE>	
<0_BYTE>	::= keine Daten-Bytes, wenn fehlerfrei	Bit15..0 des Fehlercodes = CAN-DB 2..3
<ERR_CODE>	::= <Word>	CAN-DB: CAN-Daten-Byte

### 3.2.2 Registerstruktur und Funktion des Optionsmodul CANsync-Master für b maXX PLC

Im folgenden wird die Registerstruktur des Kommunikations-RAM im Optionsmodul CANsync-Master für b maXX PLC erläutert.

Um im PROPROG wt II Projekt auf die Register des Kommunikations-RAM zugreifen zu können, sind Datentypen definiert, die die Registerstruktur abbilden. Mit diesen Datentypen werden Variablen deklariert, die auf die Basisadresse der CANsync-Anschaltung gelegt werden.

Anschließend ist es möglich über die Strukturelemente der deklarierten Variablen auf die Register des Kommunikations-RAM zuzugreifen.

Bei der Initialisierung der CANsync-Master-Anschaltung haben die Register im Kommunikations-RAM eine andere Bedeutung als nach der Initialisierung, im zyklischen Betrieb.

Deshalb gibt es für die Initialisierung die Struktur

CANsync\_PLC\_INIT\_BMSTRUCT

und für den zyklischen Betrieb die Struktur

CANsync\_PLC\_MA\_BMSTRUCT

Diese Strukturen sind ab der Bibliothek BM\_TYPES\_20bd03 definiert. Nachdem die Bibliothek BM\_TYPES\_20bd03 im Projekt eingebunden ist, stehen die Datentypen zur Verfügung.

Diese Strukturen enthalten

- 8-Bit-Elemente,
- 16-Bit-Elemente,
- 32-Bit-Elemente,
- Strukturen aus den o.g. Elementen
- Felder (ARRAY) und Strukturen aus den o.g. Elementen und Strukturen

Den in einer Struktur verwendeten Datentypen (8-, 16-, 32-Bit-Elemente, Strukturen und Feldern) wurden Kurzbezeichnungen vorangefügt. Dies dient der Übersichtlichkeit bei der Verwendung der Strukturen in der Programmierung.

Datentyp	Kurzbezeichnung	Anzahl der Bits
BYTE	b	8
WORD	w	16
DWORD (double word)	d	32
SINT (short integer)	si	8
DINT (double integer)	di	32
USINT (unsigned short integer)	us	8
UINT (unsigned integer)	u	16
UDINT (unsigned double integer)	ud	32
STRUCT	<u>_</u> (underline)	-
ARRAY	a	-

## 3.2 Detaillierte Informationen zum CANsync

Weitere, nicht in den Strukturen verwendete Datentypen sind:

Datentyp	Kurzbezeichnung	Anzahl der Bits
BOOL (bit)	x	1
TIME	t	-

### 3.2.2.1 Erläuterung zur Deklaration der globalen Variablen

Für die Initialisierung sind im Template "BM4-O-PLC01" bereits globale Variablen vom Datentyp CANsync\_PLC\_INIT\_BMSTRUCT für jeden Optionsmodul-Steckplatz deklariert.

Das sind die globalen Variablen

```
_CANsyncMaster_Init_Slot_G AT %MB3.2001792 :  
                                CANsync_PLC_INIT_BMSTRUCT;  
_CANsyncMaster_Init_Slot_H AT %MB3.3001792 :  
                                CANsync_PLC_INIT_BMSTRUCT;  
_CANsyncMaster_Init_Slot_J AT %MB3.4001792 :  
                                CANsync_PLC_INIT_BMSTRUCT;  
_CANsyncMaster_Init_Slot_K AT %MB3.5001792 :  
                                CANsync_PLC_INIT_BMSTRUCT;  
_CANsyncMaster_Init_Slot_L AT %MB3.6001792 :  
                                CANsync_PLC_INIT_BMSTRUCT;  
_CANsyncMaster_Init_Slot_M AT %MB3.7001792 :  
                                CANsync_PLC_INIT_BMSTRUCT;
```

Sollten diese Variablen nicht mehr im PROPROGRAMM vorhanden sein, legen Sie, je nach Steckplatz (Slot G bis M), die globale Variable `_CANsyncMaster_Init_Slot_G` (bis `_CANsyncMaster_Init_Slot_M`) vom Datentyp CANsync\_PLC\_INIT\_BMSTRUCT an.

Diese Variable muß bei der Deklaration der globalen Variablen auf die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul gelegt werden.

Die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Master (BM4-O-CAN-06) in Abhängigkeit vom Steckplatz ist:

Steckplatz (Slot)	Basisadresse der CANsync-Anschaltung
G	%MB3.2001792
H	%MB3.3001792
J	%MB3.4001792
K	%MB3.5001792
L	%MB3.6001792
M	%MB3.7001792

Beispiel:

CANsync-Anschaltung auf Optionsmodul CANsync-Master im Steckplatz G des b maXX

```
_CANsyncMaster_Init_Slot_G AT %MB3.2001792 :
                                CANsync_PLC_INIT_BMSTRUCT;
```

dabei ist:

<code>_CANsyncMaster_Init_Slot_G</code>	der Variablenname mit der Datentypkurzbezeichnung "_" für Struct
<code>CANsync_PLC_INIT_BMSTRUCT</code>	der Datentyp
<code>%MB3.2001792</code>	die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Master im Steckplatz G des b maXX

Für den zyklischen Betrieb sind im Template "BM4-O-PLC01" bereits globale Variablen vom Datentyp `CANsync_PLC_MA_BMSTRUCT` für jeden Optionsmodul-Steckplatz deklariert.

Das sind die globalen Variablen

```
_CANsyncMaster_Ctrl_Slot_G AT %MB3.2001792 :
                                CANsync_PLC_MA_BMSTRUCT;
_CANsyncMaster_Ctrl_Slot_H AT %MB3.3001792 :
                                CANsync_PLC_MA_BMSTRUCT;
_CANsyncMaster_Ctrl_Slot_J AT %MB3.4001792 :
                                CANsync_PLC_MA_BMSTRUCT;
_CANsyncMaster_Ctrl_Slot_K AT %MB3.5001792 :
                                CANsync_PLC_MA_BMSTRUCT;
_CANsyncMaster_Ctrl_Slot_L AT %MB3.6001792 :
                                CANsync_PLC_MA_BMSTRUCT;
_CANsyncMaster_Ctrl_Slot_M AT %MB3.7001792 :
                                CANsync_PLC_MA_BMSTRUCT;
```

Sollten diese Variablen nicht mehr im PROLOG wt II Projekt vorhanden sein, legen Sie, je nach Steckplatz (Slot G bis M), die globale Variable `_CANsyncMaster_Ctrl_Slot_G` (bis `_CANsyncMaster_Ctrl_Slot_M`) vom Datentyp `CANsync_PLC_MA_BMSTRUCT` an.

Diese Variable muß bei der Deklaration der globalen Variablen auf die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul gelegt werden.

Die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Master (BM4-O-CAN-06) in Abhängigkeit vom Steckplatz ist:

Steckplatz (Slot)	Basisadresse der CANsync-Anschaltung
G	%MB3.2001792
H	%MB3.3001792
J	%MB3.4001792

## 3.2 Detaillierte Informationen zum CANsync

Steckplatz (Slot)	Basisadresse der CANsync-Anschaltung
K	%MB3.5001792
L	%MB3.6001792
M	%MB3.7001792

Beispiel:

CANsync-Anschaltung auf Optionsmodul CANsync-Master im Steckplatz G des b maXX

```
_CANsyncMaster_Ctrl_Slot_G AT %MB3.2001792 :  
                                CANsync_PLC_MA_BMSTRUCT;
```

dabei ist:

`_CANsyncMaster_Ctrl_Slot_G` der Variablenname mit der Datentypkurzbezeichnung „\_“ für Struct  
`CANsync_PLC_MA_BMSTRUCT` der Datentyp  
`%MB3.2001792` die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Master im Steckplatz G des b maXX



### HINWEIS

In den nachfolgenden Tabellen wird der Variablenname durch \* ersetzt. Weiterhin wird davon ausgegangen, dass das Optionsmodul CANsync-Master für b maXX PLC im Steckplatz G gesteckt ist.

Auf das Register `*.w_CPU_CONTROL` greift man demzufolge über

```
_CANsyncMaster_Init_Slot_G.w_CPU_CONTROL zu,  
auf *.w_OPTION_STATUS greift man über
```

```
_CANsyncMaster_Init_Slot_G.w_OPTION_STATUS zu.
```

Dabei ist:

`_CANsyncMaster_Init_Slot_G` der Variablenname mit der Datentypkurzbezeichnung „\_“ für STRUCT  
`w_CPU_CONTROL` das Steuerregister der CANsync-Anschaltung mit der Datentypkurzbezeichnung „w“ für WORD

Die Register `*.w_CPU_CONTROL` und `*.w_OPTION_STATUS` können auch über die Struktur für den zyklischen Betrieb angesprochen werden. Der Zugriff ist dann über

```
_CANsyncMaster_Ctrl_Slot_G.w_CPU_CONTROL und  
_CANsyncMaster_Ctrl_Slot_G.w_OPTION_STATUS  
möglich.
```



Dabei ist

<code>_CANsyncMaster_Ctrl_Slot_G</code>	der Variablenname mit der Datentypkurzbezeichnung „_“ für STRUCT
<code>w_OPTION_STATUS</code>	das Statusregister der CANsync-Anschaltung mit der Datentypkurzbezeichnung „w“ für WORD

Beispiel für den Zugriff auf ein Element eines Feldes, das in der Struktur verwendet wird:

laut Tabelle: `*.a_WR_VALUE[3]`

Zugriff: `_CANsyncMaster_Ctrl_Slot_G.a_WR_VALUE[3]`

Dabei ist

<code>_CANsyncMaster_Ctrl_Slot_G</code>	der Variablenname mit der Datentypkurzbezeichnung „_“ für STRUCT
<code>a_WR_VALUE[3]</code>	das Register für den Sollwert 3 mit der Datentypkurzbezeichnung „a“ für ARRAY. Der Datentyp der Elemente des Feldes (der Sollwerte) wird der entsprechenden Tabelle und der Beschreibung entnommen.

Beispiel für den Zugriff auf ein Element eines zweidimensionalen Feldes, das in der Struktur verwendet wird:

laut Tabelle: `*.a_RD_VALUE[5][7]`

Zugriff: `_CANsyncMaster_Ctrl_Slot_G.a_RD_VALUE[5][7]`

Dabei ist

<code>_CANsyncMaster_Ctrl_Slot_G</code>	der Variablenname mit der Datentypkurzbezeichnung „_“ für STRUCT
<code>a_RD_VALUE[5][7]</code>	das Register für den Istwert 7 des CANsync-Slaves 5 mit der Datentypkurzbezeichnung „a“ für ARRAY. Der Datentyp der Elemente des Feldes (der Sollwerte) wird der entsprechenden Tabelle und der Beschreibung entnommen.

Beispiel für den Zugriff auf ein Element einer (Sub-) Struktur, die selbst Element eines Feldes ist, das in der Struktur verwendet wird:

laut Tabelle: `*.a_CFG_RDC_WORD[31].b_CFG_RDC2_WORD3`

Zugriff:

`_CANsyncMaster_Ctrl_Slot_G.a_CFG_RDC_WORD[31].b_CFG_RDC2_WORD3`

Dabei ist

<code>_CANsyncMaster_Ctrl_Slot_G</code>	der Variablenname mit der Datentypkurzbezeichnung „_“ für STRUCT
---	--

## 3.2 Detaillierte Informationen zum CANsync

a_CFG_RDC_WORD[31]	das Feld mit den Konfigurationsdaten für das Mapping der Worte der Istwert-Telegramme des CANsync-Slaves 31 mit der Datentypkurzbezeichnung „a“ für ARRAY
b_CFG_RDC2_WORD3	das Register für die Konfigurationsdaten für das Mapping des 3. Wortes des Istwert-Telegramms 2 (des CANsync-Slaves 31) mit der Datentypkurzbezeichnung „b“ für BYTE

Beispiel für den Zugriff auf ein Element einer (Sub-) Struktur, die in der Struktur verwendet wird:

laut Tabelle: `*._CFG_WRC_WORD.b_CFG_WRC1_WORD0`

Zugriff:

`_CANsyncMaster_Ctrl_Slot_G._CFG_WRC_WORD.b_CFG_WRC1_WORD0`

Dabei ist

<code>_CANsyncMaster_Ctrl_Slot_G</code>	der Variablenname mit der Datentypkurzbezeichnung „_“ für STRUCT
<code>_CFG_WRC_WORD</code>	die Struktur mit den Konfigurationsdaten für das Mapping der Worte der Sollwert-Telegramme mit der Datentypkurzbezeichnung „_“ für STRUCT
<code>b_CFG_WRC1_WORD0</code>	das Register für die Konfigurationsdaten für das Mapping des 0. Wortes des Sollwert-Telegramms 1 mit der Datentypkurzbezeichnung „b“ für BYTE

### 3.2.2.2 Allgemeine Register der CANsync-Anschaltung

Register	Inhalt
*.w_CANsync_STATUS	CANsync-Status
*.i_SW1_NR	Karten-Softwarenummer
*.i_SW1_RELEASE	Softwarestand inkompatibel und kompatibel

#### CANsync-Status

Bei jedem Durchlaufen des CANsync-Prozessorzyklusses wird der CANsync-Status auf \*.w\_CANsync\_STATUS ausgegeben.

Bedeutung:

Bit-Nr.	Bedeutung (Bit = TRUE)
0	reserviert

Bit-Nr.	Bedeutung (Bit = TRUE)
1	Overrun: eine CANsync-Nachricht konnte nicht empfangen werden
2	CANsync-Sendepuffer ist frei
3	CANsync-Sendeauftrag wurde erfolgreich durchgeführt
4	es wird gerade eine CANsync-Meldung empfangen
5	es wird gerade eine CANsync-Meldung gesendet
6	Fehler vorhanden (Warnung)
7	CANsync-Knoten ist deaktiviert (BUS-off)
8-15	reserviert

### Softwarenummer und Softwarestand

Im Register `*.i_SW1_NR` wird die Nummer der CANsync-Software auf dem Optionsmodul CANsync-Master angezeigt.

Im Register `*.i_SW1_RELEASE` wird der inkompatible und der kompatible Stand der CANsync-Software auf dem Optionsmodul CANsync-Master angezeigt.

### 3.2.2.3 Initialisierung

Für die Initialisierung sind im Template "BM4-O-PLC01" bereits globale Variablen vom Datentyp `CANsync_PLC_INIT_BMSTRUCT` für jeden Optionsmodul-Steckplatz deklariert.

Das sind die globalen Variablen

```

_CANsyncMaster_Init_Slot_G  AT  %MB3.2001792  :
                               CANsync_PLC_INIT_BMSTRUCT;
_CANsyncMaster_Init_Slot_H  AT  %MB3.3001792  :
                               CANsync_PLC_INIT_BMSTRUCT;
_CANsyncMaster_Init_Slot_J  AT  %MB3.4001792  :
                               CANsync_PLC_INIT_BMSTRUCT;
_CANsyncMaster_Init_Slot_K  AT  %MB3.5001792  :
                               CANsync_PLC_INIT_BMSTRUCT;
_CANsyncMaster_Init_Slot_L  AT  %MB3.6001792  :
                               CANsync_PLC_INIT_BMSTRUCT;
_CANsyncMaster_Init_Slot_M  AT  %MB3.7001792  :
                               CANsync_PLC_INIT_BMSTRUCT;

```

Sollten diese Variablen nicht mehr im PROPROGRAM II Projekt vorhanden sein, legen Sie, je nach Steckplatz (Slot G bis M), die globale Variable `_CANsyncMaster_Init_Slot_G` (bis `_CANsyncMaster_Init_Slot_M`) vom Datentyp `CANsync_PLC_INIT_BMSTRUCT` an.

Diese Variable muß bei der Deklaration der globalen Variablen auf die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul gelegt werden.

## 3.2 Detaillierte Informationen zum CANsync

Die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Master (BM4-O-CAN-06) in Abhängigkeit vom Steckplatz ist:

Steckplatz (Slot)	Basisadresse der CANsync-Anschaltung
G	%MB3.2001792
H	%MB3.3001792
J	%MB3.4001792
K	%MB3.5001792
L	%MB3.6001792
M	%MB3.7001792

Beispiel:

CANsync-Anschaltung auf Optionsmodul CANsync-Master im Steckplatz G des b maXX

```
_CANsyncMaster_Init_Slot_G AT %MB3.2001792 :
                                CANsync_PLC_INIT_BMSTRUCT;
```

dabei ist:

<code>_CANsyncMaster_Init_Slot_G</code>	der Variablenname mit der Datentypkurzbezeichnung "_" für Struct
<code>CANsync_PLC_INIT_BMSTRUCT</code>	der Datentyp
<code>%MB3.2001792</code>	die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Master im Steckplatz G des b maXX

Zusätzlich konfiguriert man für den Synchron-Betrieb:

Bedeutung		Register	Wert
Baudrate	z. B.: 1 MBit/s	*.b_BT_0 *.b_BT_1	16#40 16#34
CANsync- Intervall	z. B.: 1 ms	*.b_TIME_PATTERN	16#01
Acceptance Code	alle Telegramme	*.b_AC	16#FF
Acceptance Mask		*.b_AM	16#FF
Output-Control	16#FA	*.b_OUTPUT_CONTROL	16#FA
Clock-Divider	16#07	*.b_CLOCK_DIVIDER	16#07
Slave/Master	Master	*.b_MA_SL_MODE	16#00
Slave-Typen	CANsync-Slave mit Slavenummer x nicht vorhanden / vorhanden	*.a_SL_TYP[x]	16#00 / 16#01

Für jede am CANsync-Bus vorhandene CANsync-Slave-Anschaltung (Slavenummer über Dip-Schalter eingestellt) gibt man den CANsync-Slavetyp an. Zur Zeit gibt es nur den CANsync-Slavetyp 16#01. Wenn der Wert auf 16#00 gestellt wird, bedeutet dieses,

dass kein CANsync-Slave mit dieser Slavenummer vorhanden ist, bzw. vom CANsync-Master erwartet wird.

Die Betriebsarteneinstellung erfolgt über das Register `*.w_CPU_CONTROL`. Die Anzeige der gerade aktiven Betriebsart erfolgt im Register `*.w_OPTION_STATUS`. Auch nach dem erfolgreichen Starten einer Betriebsart kann die Betriebsart wieder geändert werden.

Register	Inhalt
<code>*.w_CPU_CONTROL</code>	Steuerregister CANsync-Anschaltung
<code>*.w_OPTION_STATUS</code>	Statusregister CANsync-Anschaltung

(\* entspricht zum Beispiel bei der Initialisierung `_CANsyncMaster_Init_Slot_G`, nach der Initialisierung zum Beispiel `_CANsyncMaster_Ctrl_Slot_G`)

Steuerregister CANsync- Anschaltung	Bedeutung
16#0000	Neuanlauf
16#0001	Handshake
16#0002	Initialisierungsdaten übernehmen
16#0012	reserviert
16#0013	reserviert
16#0020	Synchron-Betrieb starten
16#0040	Aktiven Betrieb freigeben
16#0080	(Bit 7 = TRUE) Reset CAN-Controller

Statusregister CANsync- Anschaltung	Bedeutung
16#0001	Anlauf
16#0002	Warten auf Initialisierungsdaten übernehmen
16#0003	Warten auf Start
16#0011	reserviert
16#0012	reserviert
16#0013	<i>Synchron-Betrieb CANsync-Slave wird eingerichtet</i>
16#0020	<i>Synchron-Betrieb CANsync-Slave ist aktiv</i>
16#0041	reserviert
16#0042	reserviert
16#0043	Synchron-Betrieb CANsync-Master wird eingerichtet
16#0080	Synchron-Betrieb CANsync-Master ist aktiv

Die Initialisierung wird mit den Kommandos 16#0000, 16#0001 und 16#0002 auf `*.w_CPU_CONTROL` durchgeführt. Damit wird der Einrichtbetrieb gestartet. Als nächstes wird der Slavestatus der initialisierten CANsync-Slaves gemeldet (siehe [►Kommando- und Antwortkanal◄](#) ab Seite 62). Wenn alle CANsync-Slaves sich gemel-

## 3.2 Detaillierte Informationen zum CANsync

det haben und den Zustand „synchronisiert“ haben, muss der aktive Betrieb freigegeben werden. Dies geschieht durch Setzen des Bit 6 in \*.w\_CPU\_CONTROL (\*.w\_CPU\_CONTROL = 16#0040).

Die Freigabe darf auch dann erfolgen, wenn sich nicht alle CANsync-Slaves gemeldet haben, dies aber von der Applikation verwaltet werden kann.

Wenn man das Bit 7 von \*.w\_CPU\_CONTROL setzt (\*.w\_CPU\_CONTROL = 16#0080), wird der CAN-Controller zurückgesetzt und das Bit wieder gelöscht. Dadurch kann der BUS-OFF-Zustand des CAN-Controllers zurückgesetzt werden und wieder CANsync-Telegramme gesendet und empfangen werden. Die Anzeige über den BUS-OFF-Zustand erfolgt in \*.w\_CANsync\_STATUS (siehe [Allgemeine Register der CANsync-Anschaltung](#) ab Seite 48).

In der folgenden Tabelle sind die Register angegeben, die in der Initialisierung bedient werden.

Register	Inhalt
*.b_MA_SL_MODE	Betriebsart: Master / Slave (SYNC-OUT / SYNC-IN)
*.b_AC	Acceptance Code des CANsync-Controllers
*.b_AM	Acceptance Mask des CANsync-Controllers
*.b_BT_0	Bit-Timing-Register 0 des CANsync-Controllers
*.b_BT_1	Bit-Timing-Register 1 des CANsync-Controllers
*.b_OUTPUT_CONTROL	Output-Control-Register des CANsync-Controllers
*.b_CLOCK_DIVIDER	Clock-Divider des CANsync-Controllers
*.b_TIME_PATTERN	CANsync-Intervall in ms
*.a_SL_TYP[0]	CANsync-Slave-Typ 0
*.a_SL_TYP[1]	CANsync-Slave-Typ 1
...	...
*.a_SL_TYP[31]	CANsync-Slave-Typ 31

(\* entspricht zum Beispiel bei der Initialisierung der Struktur `_CANsyncMaster_Init_Slot_G`)

### 3.2.2.4 Sollwerte

Das Senden der Sollwerte erfolgt im CANsync Event-Task.

Register	Inhalt
*.b_CTRLREG_WRC1	Kontrollregister Sollwertkanal 1
*.b_CTRLREG_WRC2	Kontrollregister Sollwertkanal 2
*.b_CTRLREG_WRC3	reserviert
*.b_CTRLREG_WRC4	reserviert
*.b_CTRLREG_WRC5	reserviert
*.b_CTRLREG_WRC6	reserviert
*.b_CTRLREG_WRC7	reserviert
*.b_CTRLREG_WRC8	reserviert

(\* entspricht nach der Initialisierung zum Beispiel der Variablen `_CANsyncMaster_Ctrl_Slot_G`)

Im Kontrollregister wird mit 16#05 gekennzeichnet, dass neue Sollwerte für das jeweilige Sollwert-Telegramm (auch Sollwertkanal (SWK oder WRC)) eingetragen wurden und das Telegramm gesendet wird. Die CANsync-Anschaltung quittiert den Befehl mit 16#04.

Die Konfiguration der Sollwert-Telegramme erfolgt in folgenden Registern.

Register	Inhalt
*_CFG_WRC_WORD.b_CFG_WRC1_WORD0	Konfigurierung Sollwert-Telegramm 1 Wort 0
*_CFG_WRC_WORD.b_CFG_WRC1_WORD1	Konfigurierung Sollwert-Telegramm 1 Wort 1
*_CFG_WRC_WORD.b_CFG_WRC1_WORD2	Konfigurierung Sollwert-Telegramm 1 Wort 2
*_CFG_WRC_WORD.b_CFG_WRC1_WORD3	Konfigurierung Sollwert-Telegramm 1 Wort 3
*_CFG_WRC_WORD.b_CFG_WRC2_WORD0	Konfigurierung Sollwert-Telegramm 2 Wort 0
*_CFG_WRC_WORD.b_CFG_WRC2_WORD1	Konfigurierung Sollwert-Telegramm 2 Wort 1
*_CFG_WRC_WORD.b_CFG_WRC2_WORD2	Konfigurierung Sollwert-Telegramm 2 Wort 2
*_CFG_WRC_WORD.b_CFG_WRC2_WORD3	Konfigurierung Sollwert-Telegramm 2 Wort 3
*_CFG_WRC_WORD.b_CFG_WRC3_WORD0	reserviert
*_CFG_WRC_WORD.b_CFG_WRC3_WORD1	reserviert
*_CFG_WRC_WORD.b_CFG_WRC3_WORD2	reserviert
*_CFG_WRC_WORD.b_CFG_WRC3_WORD3	reserviert
...	...
*_CFG_WRC_WORD.b_CFG_WRC8_WORD0	reserviert
*_CFG_WRC_WORD.b_CFG_WRC8_WORD1	reserviert
*_CFG_WRC_WORD.b_CFG_WRC8_WORD2	reserviert
*_CFG_WRC_WORD.b_CFG_WRC8_WORD3	reserviert

(\* entspricht zum Beispiel \_CANsyncMaster\_Ctrl\_Slot\_G)

Mit der Konfiguration gibt man an, welcher Sollwert an welcher Stelle (...\_WORD0, ..., ...\_WORD3) in ein Sollwerttelegramm (...\_WRC1\_..., ...\_WRC2\_...) eingetragen wird.

#### Bedeutung

Bit 0	Belegt Wenn = 1, wird dieses Wort des Telegramms verwendet
Bit 1	Highword/Lowword Wenn = 1, wird das Highword des Sollwertes eingetragen
Bit 2	Sollwertnummer Nummer des Sollwertes 0 bis 31
Bit 3	
Bit 4	
Bit 5	
Bit 6	
Bit 7	reserviert, muss auf Null gesetzt werden

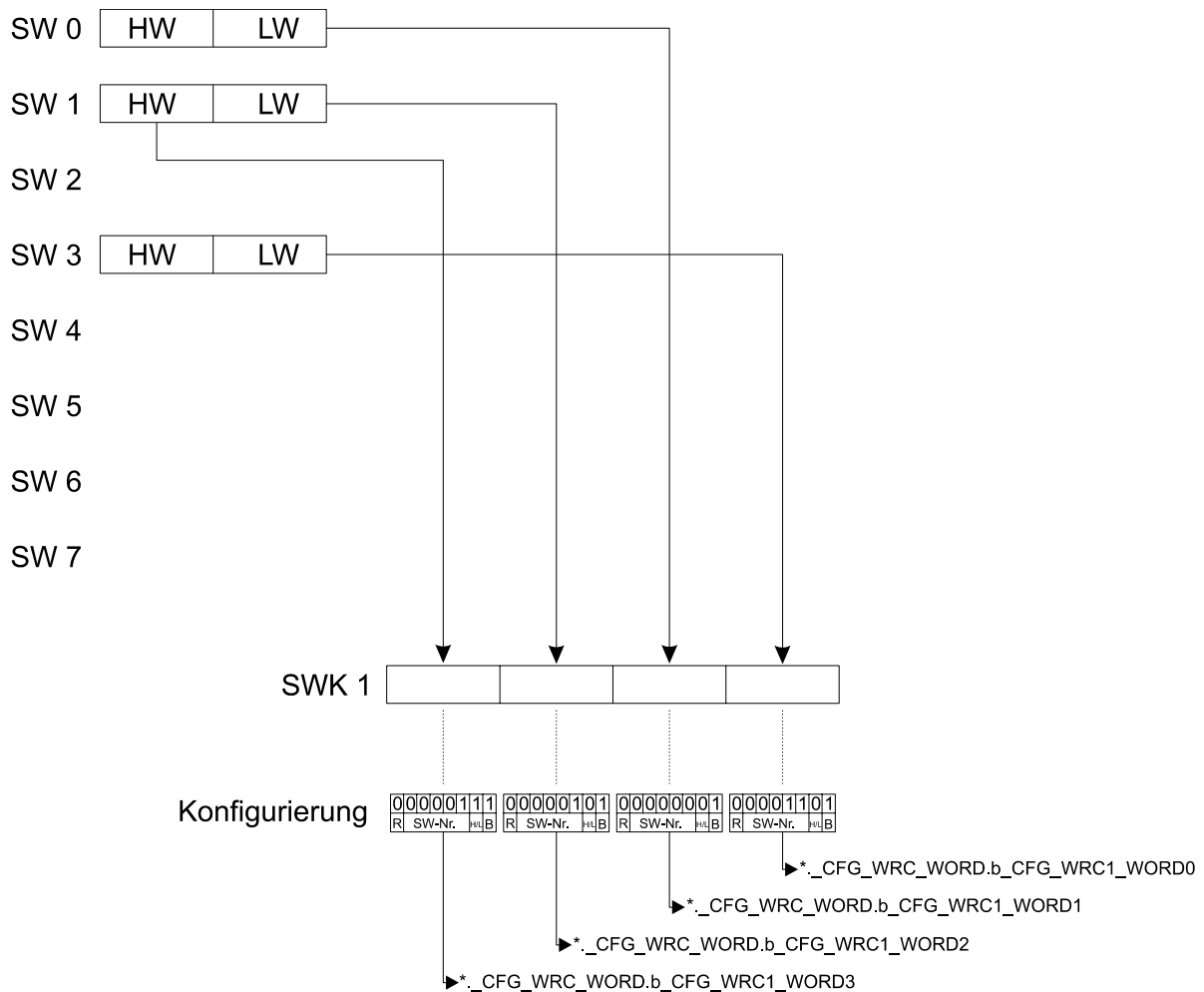
## 3.2 Detaillierte Informationen zum CANsync



### HINWEIS

Ein Wort der Nachricht wird nur dann nicht verwendet, wenn Sollwertnummer = 0, Highword/ Lowword = 0 und Belegt = 0 gesetzt wird.

### Beispiel:



\*\_CFG\_WRC\_WORD.b\_CFG\_WRC1\_WORD0 = 16#0D

0	0	0	0	1	1	0	1
R	SW-Nr.					H/L	B

(\* entspricht zum Beispiel \_CANsyncMaster\_Ctrl\_Slot\_G)

Mit dieser Einstellung wird das Lowword des Sollwertes 3 im Sollwert-Telegramm 1 im Wort 0 eingetragen.



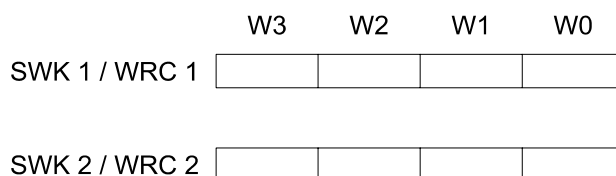
Register	Inhalt
*.a_WR_VALUE[0]	Sollwert 0
*.a_WR_VALUE[1]	Sollwert 1
*.a_WR_VALUE[2]	Sollwert 2
*.a_WR_VALUE[3]	Sollwert 3
*.a_WR_VALUE[4]	Sollwert 4
*.a_WR_VALUE[5]	Sollwert 5
*.a_WR_VALUE[6]	Sollwert 6
*.a_WR_VALUE[7]	Sollwert 7
*.a_WR_VALUE[8]	reserviert
...	...
*.a_WR_VALUE[31]	reserviert

(\* entspricht zum Beispiel \_CANsyncMaster\_Ctrl\_Slot\_G)

Die Sollwerte können Wort- oder Doppelwort-Sollwerte sein.

### Erläuterung zur Benutzung der Sollwertkanäle

Im Synchron-Betrieb stehen zwei Sollwertkanäle (Kanal 1 und 2; auch WRC1 und WRC2) zur Verfügung. In den Sollwertkanälen 1 und 2 werden die Sollwert-Telegramme 1 und 2 gesendet.



Beide Sollwert-Telegramme bestehen aus jeweils vier Worten (W0 bis W3). Nach der CAN-Initialisierung muss zumindest einmal angegeben werden, wie die Belegung dieser Worte ist. Dies kann im Initialisierungsprogramm erfolgen. Dazu trägt man im Bereich `*._CFG_WRC_WORD.b_CFG_WRC1_WORD0` bis `*._CFG_WRC_WORD.b_CFG_WRC2_WORD3` für jedes Wort der Sollwert-Telegramme ein, welcher Sollwert an dieser Stelle übertragen wird. Die Sollwertnummern sind im Bereich von 0 bis 7 gültig. Für Doppelwort-Sollwerte müssen zwei Worte im Telegramm verwendet werden.

Die Konfiguration kann auch während des aktiven Betriebs geändert werden. Die Änderung wird spätestens im nächsten CANsync-Intervall übernommen. Zu Beginn des CANsync-Intervalls werden die Konfigurationsdaten von der CANsync-Anschaltung gelesen.

Die Sollwertgenerierung wird im Synchron-Betrieb in der CANsync Event-Task erfolgen. Von der fallenden Flanke des SYNC-Signals bis zur Ausführung des Applikationsprogramms in der CANsync Event-Task vergehen ca. 70 µs. Die Sollwerte für die Sollwert-Telegramme müssen bis 200 µs nach dem Start der CANsync Event-Task im Kommunikations-RAM als Sollwert (\*.a\_WR\_VALUE[0] bis \*.a\_WR\_VALUE[7]) eingetragen werden, weil dann die CANsync-Anschaltung die Generierung der Sollwert-Telegramme beginnt. Als Kennzeichnung, dass neue Sollwerte eingetragen wurden, muss im entsprechenden Kontrollregister (\*.b\_CTRLREG\_WRC1 oder \*.b\_CTRLREG\_WRC2) 16#05 eingetragen werden. Dies ist die Freigabe für die CANsync-Anschaltung, dass die Sollwerte gelesen werden dürfen und die Sollwert-Telegramme erzeugt werden. Falls bis

## 3.2 Detaillierte Informationen zum CANsync

200 µs nach dem Start der CANsync Event-Task diese Freigabe nicht erfolgt, fallen in diesem CANsync-Intervall die Sollwert-Telegramme aus. Als Quittung für die Erzeugung der Sollwert-Telegramme wird in den Kontrollregistern 16#04 von der CANsync-Anschaltung eingetragen. Dadurch kann der Anwender überprüfen, ob die Sollwertgenerierung rechtzeitig fertig geworden ist oder nicht. Falls die Sollwertgenerierung länger dauert, muss die Sollwertgenerierung immer für den nächsten Start der CANsync Event-Task erfolgen und zu Beginn der neuen CANsync Event-Task müssen die vorberechneten Sollwerte nur noch an die entsprechenden Stellen im Kommunikations-RAM kopiert werden.

Mit dem Sollwert-Telegramm wird das Istwert-Telegramm eines CANsync-Slaves angefordert. Die Nummer der CANsync-Slave-Anschaltung wird im Kontrollregister Istwertanforderung (\*.b\_CTRLREG\_RD\_ORDER\_RDC1 oder \*.b\_CTRLREG\_RD\_ORDER\_RDC2) eingetragen (weitere Beschreibung siehe [Istwerte](#) ab Seite 56). Das Sollwert-Telegramm 1 fordert das Istwert-Telegramm 1 an und das Sollwert-Telegramm fordert das Istwert-Telegramm 2 an.

### 3.2.2.5 Istwerte

Die Istwert-Telegramme werden durch eine bestimmte Information im Identifier der Sollwert-Telegramme angefordert (siehe [Sollwertkanal 1](#) auf Seite 34).

Register	Inhalt
*.b_MAX_SL_NR	maximale Slavenummer (Bus-Adresse des CANsync-Slaves)

(\* entspricht zum Beispiel \_CANsyncMaster\_Ctrl\_Slot\_G)

Sie gibt die größte Slavenummer des CANsync-Slaves für die automatische Istwertanforderung (siehe \*.b\_CTRLREG\_RD\_ORDER\_RDC1 ff.) an.

Register	Inhalt
*.b_CTRLREG_RD_ORDER_RDC1	Kontrollregister Istwertanforderung Istwertkanal 1
*.b_CTRLREG_RD_ORDER_RDC2	Kontrollregister Istwertanforderung Istwertkanal 2
*.b_CTRLREG_RD_ORDER_RDC3	reserviert
*.b_CTRLREG_RD_ORDER_RDC4	reserviert

(\* entspricht zum Beispiel \_CANsyncMaster\_CtrlSlot\_G)

Im Kontrollregister kann die Slavenummer des CANsync-Slaves angegeben werden, der im entsprechenden Istwert-Telegramm seine Istwerte melden soll. Wenn im Kontrollregister 16#80 eingegeben wird, wird die Slavenummer in jedem Zyklus um eins erhöht, bis die maximale Slavenummer (\*.b\_MAX\_SL\_NR) erreicht ist. Dann wird wieder mit der Slavenummer 0 gestartet.

Register	Inhalt
*.b_STATREG_RDC1	Statusregister Istwertkanal 1
*.b_STATREG_RDC2	Statusregister Istwertkanal 2
*.b_STATREG_RDC3	reserviert
*.b_STATREG_RDC4	reserviert

(\* entspricht zum Beispiel \_CANsyncMaster\_Ctrl\_Slot\_G)

Im Statusregister wird die Slavenummer des CANsync-Slaves eingetragen, von dem ein Istwert-Telegramm empfangen wurde.

Register	Inhalt
*.a_STATREG_RDC[0].b_STATREG_RDC1	Istwertquittung Istwertkanal 1 Slave 0
*.a_STATREG_RDC[0].b_STATREG_RDC2	Istwertquittung Istwertkanal 2 Slave 0
*.a_STATREG_RDC[0].b_STATREG_RDC3	reserviert
*.a_STATREG_RDC[0].b_STATREG_RDC4	reserviert
*.a_STATREG_RDC[1].b_STATREG_RDC1	Istwertquittung Istwertkanal 1 Slave 1
*.a_STATREG_RDC[1].b_STATREG_RDC2	Istwertquittung Istwertkanal 2 Slave 1
*.a_STATREG_RDC[1].b_STATREG_RDC3	reserviert
*.a_STATREG_RDC[1].b_STATREG_RDC4	reserviert
*.a_STATREG_RDC[2].b_STATREG_RDC1	Istwertquittung Istwertkanal 1 Slave 2
*.a_STATREG_RDC[2].b_STATREG_RDC2	Istwertquittung Istwertkanal 2 Slave 2
*.a_STATREG_RDC[2].b_STATREG_RDC3	reserviert
*.a_STATREG_RDC[2].b_STATREG_RDC4	reserviert
...	...
*.a_STATREG_RDC[31].b_STATREG_RDC1	Istwertquittung Istwertkanal 1 Slave 31
*.a_STATREG_RDC[31].b_STATREG_RDC2	Istwertquittung Istwertkanal 2 Slave 31
*.a_STATREG_RDC[31].b_STATREG_RDC3	reserviert
*.a_STATREG_RDC[31].b_STATREG_RDC4	reserviert

(\* entspricht zum Beispiel \_CANsyncMaster\_Ctrl\_Slot\_G)

Wenn das entsprechende Istwert-Telegramm für einen CANsync-Slave angekommen ist, wird eine 16#02 in das Istwertquittungsregister eingetragen.

Die Konfigurierung der Istwert-Telegramme erfolgt in folgenden Registern.

Register	Inhalt
*.a_CFG_RDC_WORD[0].b_CFG_RDC1_WORD0	Konfigurierung Istwert-Telegramm 1 Wort 0 Slave 0
*.a_CFG_RDC_WORD[0].b_CFG_RDC1_WORD1	Konfigurierung Istwert-Telegramm 1 Wort 1 Slave 0
*.a_CFG_RDC_WORD[0].b_CFG_RDC1_WORD2	Konfigurierung Istwert-Telegramm 1 Wort 2 Slave 0
*.a_CFG_RDC_WORD[0].b_CFG_RDC1_WORD3	Konfigurierung Istwert-Telegramm 1 Wort 3 Slave 0
*.a_CFG_RDC_WORD[0].b_CFG_RDC2_WORD0	Konfigurierung Istwert-Telegramm 2 Wort 0 Slave 0
*.a_CFG_RDC_WORD[0].b_CFG_RDC2_WORD1	Konfigurierung Istwert-Telegramm 2 Wort 1 Slave 0
*.a_CFG_RDC_WORD[0].b_CFG_RDC2_WORD2	Konfigurierung Istwert-Telegramm 2 Wort 2 Slave 0
*.a_CFG_RDC_WORD[0].b_CFG_RDC2_WORD3	Konfigurierung Istwert-Telegramm 2 Wort 3 Slave 0
*.a_CFG_RDC_WORD[0].b_CFG_RDC3_WORD0	reserviert
*.a_CFG_RDC_WORD[0].b_CFG_RDC3_WORD1	reserviert
*.a_CFG_RDC_WORD[0].b_CFG_RDC3_WORD2	reserviert
*.a_CFG_RDC_WORD[0].b_CFG_RDC3_WORD3	reserviert
*.a_CFG_RDC_WORD[0].b_CFG_RDC4_WORD0	reserviert
*.a_CFG_RDC_WORD[0].b_CFG_RDC4_WORD1	reserviert
*.a_CFG_RDC_WORD[0].b_CFG_RDC4_WORD2	reserviert
*.a_CFG_RDC_WORD[0].b_CFG_RDC4_WORD3	reserviert
*.a_CFG_RDC_WORD[1].b_CFG_RDC1_WORD0	Konfigurierung Istwert-Telegramm 1 Wort 0 Slave 1

## 3.2 Detaillierte Informationen zum CANsync

Register	Inhalt
*.a_CFG_RDC_WORD[1].b_CFG_RDC1_WORD1	Konfigurierung Istwert-Telegramm 1 Wort 1 Slave 1
*.a_CFG_RDC_WORD[1].b_CFG_RDC1_WORD2	Konfigurierung Istwert-Telegramm 1 Wort 2 Slave 1
*.a_CFG_RDC_WORD[1].b_CFG_RDC1_WORD3	Konfigurierung Istwert-Telegramm 1 Wort 3 Slave 1
*.a_CFG_RDC_WORD[1].b_CFG_RDC2_WORD0	Konfigurierung Istwert-Telegramm 2 Wort 0 Slave 1
*.a_CFG_RDC_WORD[1].b_CFG_RDC2_WORD1	Konfigurierung Istwert-Telegramm 2 Wort 1 Slave 1
*.a_CFG_RDC_WORD[1].b_CFG_RDC2_WORD2	Konfigurierung Istwert-Telegramm 2 Wort 2 Slave 1
*.a_CFG_RDC_WORD[1].b_CFG_RDC2_WORD3	Konfigurierung Istwert-Telegramm 2 Wort 3 Slave 1
*.a_CFG_RDC_WORD[1].b_CFG_RDC3_WORD0	reserviert
*.a_CFG_RDC_WORD[1].b_CFG_RDC3_WORD1	reserviert
*.a_CFG_RDC_WORD[1].b_CFG_RDC3_WORD2	reserviert
*.a_CFG_RDC_WORD[1].b_CFG_RDC3_WORD3	reserviert
*.a_CFG_RDC_WORD[1].b_CFG_RDC4_WORD0	reserviert
*.a_CFG_RDC_WORD[1].b_CFG_RDC4_WORD1	reserviert
*.a_CFG_RDC_WORD[1].b_CFG_RDC4_WORD2	reserviert
*.a_CFG_RDC_WORD[1].b_CFG_RDC4_WORD3	reserviert
...	...
*.a_CFG_RDC_WORD[31].b_CFG_RDC1_WORD0	Konfigurierung Istwert-Telegramm 1 Wort 0 Slave 31
*.a_CFG_RDC_WORD[31].b_CFG_RDC1_WORD1	Konfigurierung Istwert-Telegramm 1 Wort 1 Slave 31
*.a_CFG_RDC_WORD[31].b_CFG_RDC1_WORD2	Konfigurierung Istwert-Telegramm 1 Wort 2 Slave 31
*.a_CFG_RDC_WORD[31].b_CFG_RDC1_WORD3	Konfigurierung Istwert-Telegramm 1 Wort 3 Slave 31
*.a_CFG_RDC_WORD[31].b_CFG_RDC2_WORD0	Konfigurierung Istwert-Telegramm 2 Wort 0 Slave 31
*.a_CFG_RDC_WORD[31].b_CFG_RDC2_WORD1	Konfigurierung Istwert-Telegramm 2 Wort 1 Slave 31
*.a_CFG_RDC_WORD[31].b_CFG_RDC2_WORD2	Konfigurierung Istwert-Telegramm 2 Wort 2 Slave 31
*.a_CFG_RDC_WORD[31].b_CFG_RDC2_WORD3	Konfigurierung Istwert-Telegramm 2 Wort 3 Slave 31
*.a_CFG_RDC_WORD[31].b_CFG_RDC3_WORD0	reserviert
*.a_CFG_RDC_WORD[31].b_CFG_RDC3_WORD1	reserviert
*.a_CFG_RDC_WORD[31].b_CFG_RDC3_WORD2	reserviert
*.a_CFG_RDC_WORD[31].b_CFG_RDC3_WORD3	reserviert
*.a_CFG_RDC_WORD[31].b_CFG_RDC4_WORD0	reserviert
*.a_CFG_RDC_WORD[31].b_CFG_RDC4_WORD1	reserviert
*.a_CFG_RDC_WORD[31].b_CFG_RDC4_WORD2	reserviert
*.a_CFG_RDC_WORD[31].b_CFG_RDC4_WORD3	reserviert

(\* entspricht zum Beispiel \_CANsyncMaster\_Ctrl\_Slot\_G)

Mit der Konfigurierung gibt man an, welcher Istwert an welcher Stelle im Istwert-Telegramm übertragen wird.

### Bedeutung

Bit 0	Belegt Wenn = 1, wird dieses Wort des Telegramms verwendet
Bit 1	Highword/Lowword Wenn = 1, wird das Highword des Istwertes gelesen

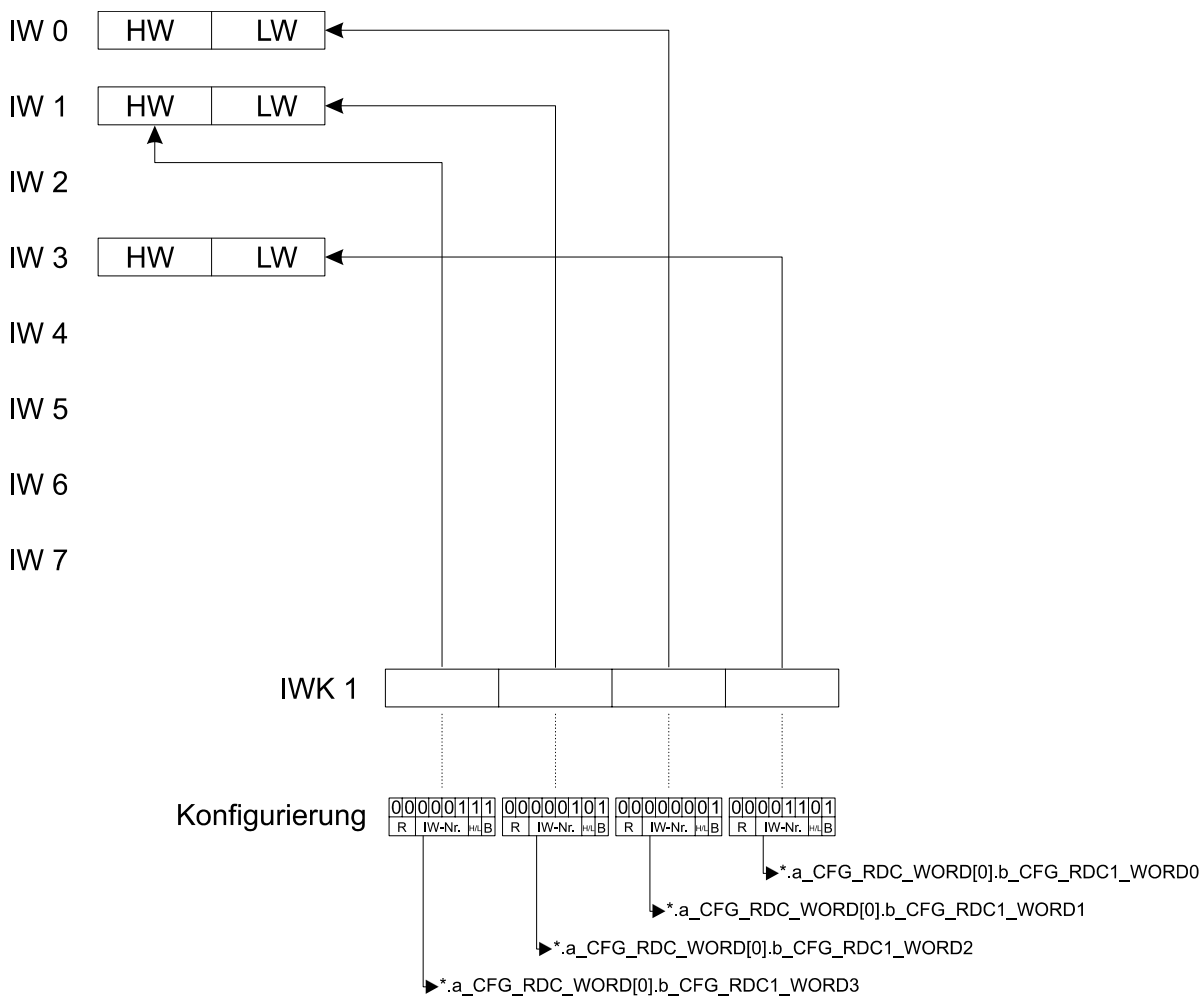
Bit 2	<b>Istwertnummer</b> Nummer des Istwertes 0 bis 15
Bit 3	
Bit 4	
Bit 5	
Bit 6	reserviert
Bit 7	



**HINWEIS**

Ein Wort des Telegramms wird nur dann nicht verwendet, wenn Istwertnummer = 0, Highword/Lowword = 0 und Belegt = 0 gesetzt wird.

**Beispiel:**



## 3.2 Detaillierte Informationen zum CANsync

\*a\_CFG\_RDC\_WORD[0].b\_CFG\_RDC1\_WORD0 = 16#0D

0	0	0	0	1	1	0	1
R		IW-Nr.				H/L	B

(\* entspricht zum Beispiel \_CANsyncMaster\_Ctrl\_Slot\_G)

Damit wird aus dem Istwert-Telegramm 1 vom CANsync-Slave 0 das Wort 0 gelesen und in das Lowword des Istwertes 3 geschrieben.

Register	Inhalt
*a_RD_BMARRAY[0][0]	Istwert 0 Slave 0
*a_RD_BMARRAY[0][1]	Istwert 1 Slave 0
*a_RD_BMARRAY[0][2]	Istwert 2 Slave 0
*a_RD_BMARRAY[0][3]	Istwert 3 Slave 0
*a_RD_BMARRAY[0][4]	Istwert 4 Slave 0
*a_RD_BMARRAY[0][5]	Istwert 5 Slave 0
*a_RD_BMARRAY[0][6]	Istwert 6 Slave 0
*a_RD_BMARRAY[0][7]	Istwert 7 Slave 0
*a_RD_BMARRAY[0][8]	reserviert
...	...
*a_RD_BMARRAY[0][15]	reserviert
*a_RD_BMARRAY[1][0]	Istwert 0 Slave 1
*a_RD_BMARRAY[1][1]	Istwert 1 Slave 1
*a_RD_BMARRAY[1][2]	Istwert 2 Slave 1
*a_RD_BMARRAY[1][3]	Istwert 3 Slave 1
*a_RD_BMARRAY[1][4]	Istwert 4 Slave 1
*a_RD_BMARRAY[1][5]	Istwert 5 Slave 1
*a_RD_BMARRAY[1][6]	Istwert 6 Slave 1
*a_RD_BMARRAY[1][7]	Istwert 7 Slave 1
*a_RD_BMARRAY[1][8]	reserviert
...	...
*a_RD_BMARRAY[1][15]	reserviert
...	...
*a_RD_BMARRAY[31][0]	Istwert 0 Slave 31
*a_RD_BMARRAY[31][1]	Istwert 1 Slave 31
*a_RD_BMARRAY[31][2]	Istwert 2 Slave 31
*a_RD_BMARRAY[31][3]	Istwert 3 Slave 31
*a_RD_BMARRAY[31][4]	Istwert 4 Slave 31
*a_RD_BMARRAY[31][5]	Istwert 5 Slave 31
*a_RD_BMARRAY[31][6]	Istwert 6 Slave 31
*a_RD_BMARRAY[31][7]	Istwert 7 Slave 31
*a_RD_BMARRAY[31][8]	reserviert
...	...
*a_RD_BMARRAY[31][15]	reserviert

(\* entspricht zum Beispiel \_CANsyncMaster\_Ctrl\_Slot\_G)

Die Istwerte können Wort- oder Doppelwort-Istwerte sein.



### HINWEIS

Bei der Anzeige von `_CANsyncMaster_Ctrl_Slot_G` im PROPROG wt II Watch-Fenster kann zwischen den eckigen Klammern ein Punkt stehen.

### Erläuterung zur Benutzung der Istwertkanäle

Im Synchron-Betrieb stehen zwei Istwertkanäle (Kanal 1 und 2; auch RDC1 und RDC2) zur Verfügung. In den Istwertkanälen 1 und 2 werden die Istwert-Telegramme 1 und 2 gesendet.

	W3	W2	W1	W0
IWK 1 / RDC 1				
IWK 2 / RDC 2				

Beide Istwert-Telegramme bestehen aus jeweils vier Worten (W0 bis W3). Nach der CANsync-Initialisierung muss zumindest einmal angegeben werden, wie die Belegung dieser Worte ist. Dies kann im Initialisierungsprogramm erfolgen.

Dazu trägt man im Bereich `*.a_CFG_RDC_WORD[0].b_CFG_RDC1_WORD0` bis `*.a_CFG_RDC_WORD[31].b_CFG_RDC2_WORD3` für jedes belegte Wort der Istwert-Telegramme (jedes CANsync-Slaves) ein, welcher Istwert an dieser Stelle übertragen wird. Istwertnummern sind im Bereich von 0 bis 7 gültig. Für Doppelwort-Istwerte müssen zwei Worte im Telegramm verwendet werden.

Die Konfigurierung kann auch während des aktiven Betriebs geändert werden. Die Änderung wird spätestens im nächsten CANsync-Intervall übernommen. (Die Konfigurationsdaten werden von der CANsync-Anschaltung gelesen, wenn das entsprechende Istwert-Telegramm empfangen wird.)

Die Istwert-Telegramm-Auswertung erfolgt zu Beginn der CANsync Event-Task. Dazu gibt es zwei Auswertungsmethoden: Bei der ersten Methode fragt man die Statusregister der Istwertkanäle ab (`*.b_STATREG_RDC1` oder `*.b_STATREG_RDC2`). In diesen Registern wird die Slavenummer des CANsync-Slaves eingetragen, von dem das entsprechende Istwert-Telegramm empfangen wurde. Dann kann man die Istwerte aus den Registern (`*.a_RD_BMARRAY[0][0]` bis `*.a_RD_BMARRAY[31][7]`) dieses CANsync-Slaves auslesen und das Statusregister auf z. B. `16#FF` setzen, um die nächste Eintragung richtig erkennen zu können.

Bei der zweiten Methode fragt man direkt die Istwertquittung (`*.a_STATREG_RDC[0].b_STATREG_RDC1` bis `*.a_STATREG_RDC[31].b_STATREG_RDC2`) für einen CANsync-Slave ab. In diesem Quittungsregister wird der Empfang eines Istwert-Telegramms mit `16#02` gekennzeichnet. Dann kann man die Istwerte aus den Registern (`*.a_RD_BMARRAY[0][0]` bis `*.a_RD_BMARRAY[31][7]`) dieses CANsync-Slaves auslesen. Auch hier muss man das Statusregister anschließend mit einem anderen Wert beschreiben, um das Wiedereintragen der Quittung zu erkennen.

Die Zuordnung welche Istwerte gelesen werden dürfen, wenn ein Istwert-Telegramm empfangen wurde, muss entsprechend der Istwertkonfigurierung im Applikationsprogramm erfolgen.

Die Anforderung an einen CANsync-Slave, sein Istwert-Telegramm zu senden, erfolgt über das Sollwert-Telegramm.

Die Nummer des CANsync-Slaves wird im Kontrollregister Istwertanforderung (\*b\_CTRLREG\_RD\_ORDER\_RDC1 und \*b\_CTRLREG\_RD\_ORDER\_RDC2) eingetragen. Das Sollwert-Telegramm 1 fordert das Istwert-Telegramm 1 an, usw.

Die Anforderungsnummer für das Istwert-Telegramm 1 und Istwert-Telegramm 2, usw. können gleich sein oder auch verschieden, d. h. Istwert-Telegramm 1 kann von CANsync-Slave x und Istwert-Telegramm 2 kann von CANsync-Slave y angefordert werden.

Für die Anforderungsnummer des CANsync-Slaves gibt es zwei Möglichkeiten: Bei der ersten Möglichkeit gibt man direkt die Slavenummer im Kontrollregister an. Solange dieses Register nicht verändert wird, wird in jedem CANsync-Intervall der gleiche CANsync-Slave angesprochen.

Wenn man als Anforderungsnummer 16#80 einträgt, wird die Slavenummer von der CANsync-Anschaltung automatisch in jedem CANsync-Intervall um eins erhöht, bis die maximale Slavenummer (\*b\_MAX\_SL\_NR) erreicht ist. Dann beginnt die Abfrage wieder mit Slavenummer 0. Damit ist eine automatische Anforderung der Istwert-Telegramme aller vorhandenen CANsync-Slaves möglich. Wenn die Slavenummern der CANsync-Slaves mit Lücken vergeben werden, wird zwar auch eine Istwertanforderung für die nicht vorhandenen CANsync-Slaves erzeugt, dies führt jedoch zu keinen Funktionsproblemen.

### 3.2.2.6 Kommando- und Antwortkanal

#### Konfigurierung Kommandokanal

Die Konfigurierung der Kommandokanalbelegung erfolgt in folgenden Registern:

Register	Inhalt
*b_SL_NR_CONTROLWORD	Slavenummer des CANsync-Slaves für Steuerwortkommando
*b_SL_NR_PARAMETER	Slavenummer des CANsync-Slaves für Parameterkommando
*b_SL_NR_UPDOWN	Slavenummer des CANsync-Slaves für Up-/Downloadkommando

(\* entspricht zum Beispiel \_CANsyncMaster\_Ctrl\_Slot\_G)

#### Erläuterung zur Konfigurierung des Kommandokanals

Über die Konfigurierungsregister \*b\_SL\_NR\_CONTROLWORD, \*b\_SL\_NR\_PARAMETER und \*b\_SL\_NR\_UPDOWN kann die Benutzung des Kommandokanals eingestellt werden. Die Normaleinstellung ist, dass alle drei Register auf 16#80 eingestellt werden. Dann werden die Slavenummern der CANsync-Slaves, für die gegebenenfalls Kommando-Telegramme erzeugt werden, zyklisch erhöht. Die maximale Slavenummer ist die gleiche wie für die Istwert-Anforderung (\*b\_MAX\_SL\_NR).

Alternativ kann auch in jedem Register eine explizite Vorgabe der Slavenummer erfolgen. Dann wird nur geprüft, ob für diesen CANsync-Slave das entsprechende Kommando erzeugt werden soll.

In einem CANsync-Intervall kann nur ein Kommando gesendet werden. Das folgende Schema gibt an, welche Bereiche in welcher Reihenfolge im Synchron-Betrieb abgefragt werden, ob ein Auftrag für die Erzeugung eines Kommandos eingetragen ist. Die verschiedenen Kommandos sind in den nächsten Abschnitten erläutert.



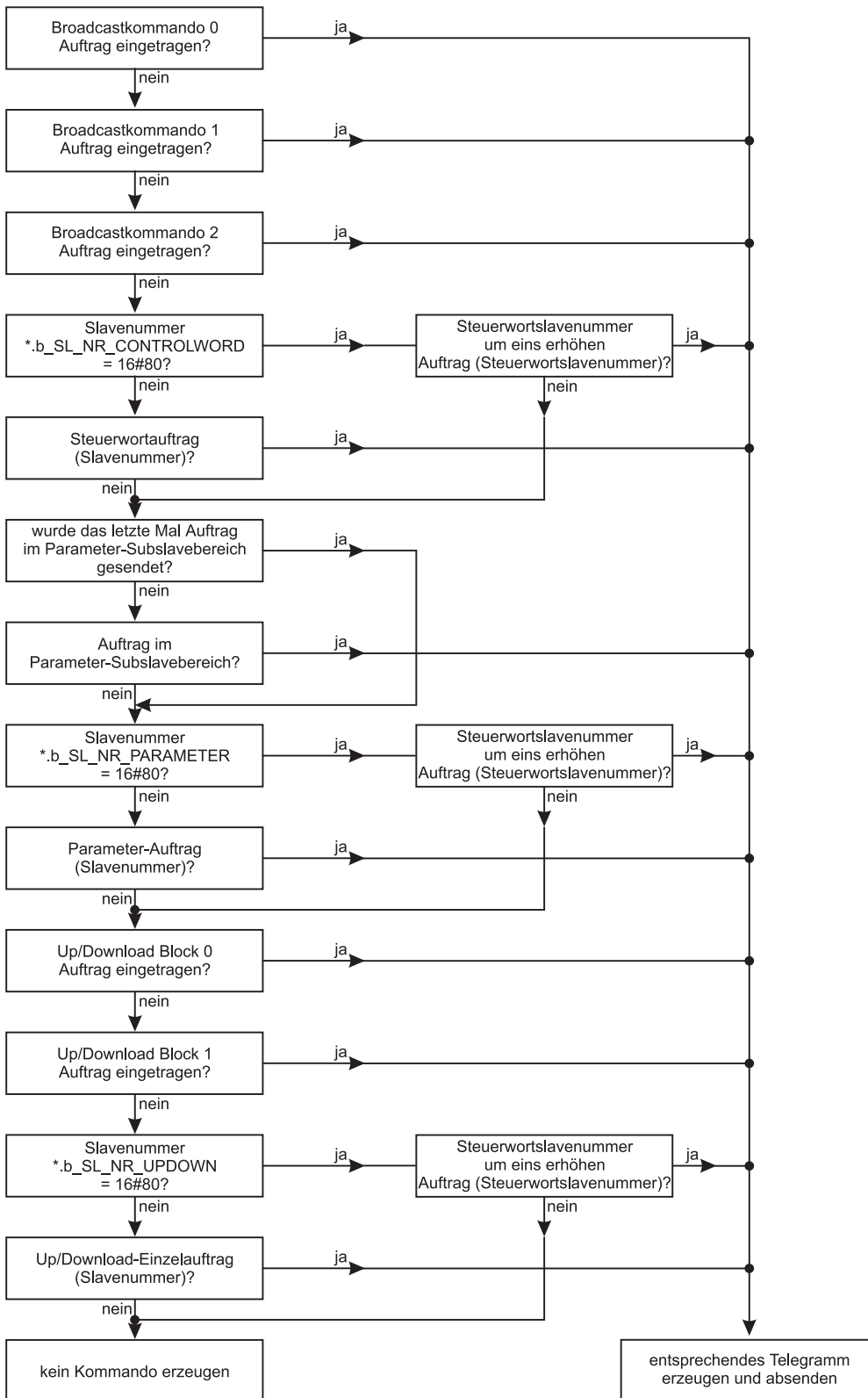


Abbildung 6: Ablauf und Priorität von Aufträgen (\* entspricht zum Beispiel \_CANsyncMaster\_Ctrl\_Slot\_G)

Aus dem Schema kann man erkennen, welche Aufträge vorrangig behandelt werden (z. B. Broadcast-Kommandos), welche Aufträge andere Aufträge blockieren können und nach welcher Anzahl von CANsync-Intervallen spätestens ein Auftrag bearbeitet wird. Das Broadcastkommando 0 hat die höchste Priorität, es wird sofort in diesem CANsync-Intervall gesendet. Nur wenn in diesem Bereich kein Auftrag eingetragen ist, wird der nächste Bereich abgeprüft. Die Vorgeschichte wird nicht berücksichtigt. Wenn also in jedem CANsync-Intervall ein Auftrag im Broadcastbereich 0 eingetragen ist, wird kein anderes Kommando mehr gesendet.

### Broadcastkommando

Broadcastkommandos können mit den folgenden Registern gesendet werden:

Register	Inhalt
*.b_CTRL_REG_BC0	Kontrollregister Broadcastkommando 0
*.d_SL_MASK_BC0	Bitleiste Broadcastkommando 0
*.b_CMD_BC0	Kommandobyte Broadcastkommando 0
*.b_DATA_BYTE_BC0	Datenbyte 0 Broadcastkommando 0 (DB)
*.w_DATA_WORD_BC0	Datenwort 1 Broadcastkommando 0 (DW)

(\* entspricht zum Beispiel \_CANsyncMaster\_Ctrl\_Slot\_G)

Register	Inhalt
*.b_CTRL_REG_BC1	Kontrollregister Broadcastkommando 1
*.d_SL_MASK_BC1	Bitleiste Broadcastkommando 1
*.b_CMD_BC1	Kommandobyte Broadcastkommando 1
*.b_DATA_BYTE_BC1	Datenbyte 0 Broadcastkommando 1 (DB)
*.w_DATA_WORD_BC1	Datenwort 1 Broadcastkommando 1 (DW)

(\* entspricht zum Beispiel \_CANsyncMaster\_Ctrl\_Slot\_G)

Register	Inhalt
*.b_CTRL_REG_BC2	Kontrollregister Broadcastkommando 2
*.d_SL_MASK_BC2	Bitleiste Broadcastkommando 2
*.b_CMD_BC2	Kommandobyte Broadcastkommando 2
*.b_DATA_BYTE_BC2	Datenbyte 0 Broadcastkommando 2 (DB)
*.w_DATA_WORD_BC2	Datenwort 1 Broadcastkommando 2 (DW)

(\* entspricht zum Beispiel \_CANsyncMaster\_Ctrl\_Slot\_G)

In der Bitleiste \*.d\_SL\_MASK\_BC0, \*.d\_SL\_MASK\_BC1, \*.d\_SL\_MASK\_BC2 wird angegeben für welche CANsync-Slaves das Aktionskommando bestimmt ist. Jedem CANsync-Slave ist ein Bit zugeordnet (Bit 0 = Slave 0; Bit 1 = Slave 1; ... Bit 31 = Slave 31).

Wenn dieses Bit = TRUE ist, wird der zugeordnete CANsync-Slave das Kommando ausführen.

Zu einem Befehl können noch ein Datenbyte (\*.b\_DATA\_BYTE...) und ein Datenwort (\*.w\_DATA\_WORD...) übertragen werden.

Um ein Kommando abzuschicken, muss in das Kontrollregister 16#05 eingetragen werden. Als Bestätigung für das Senden wird dann von der CANsync-Anschaltung 16#04 gemeldet.

#### Liste der Kommandos

Kommandobyte	Kommando
16#01	Steuerwort DB = nicht benutzt DW = Steuerwort
16#02 - 16#FF	reserviert

#### Steuerwortkommando

Das Steuerwortkommando wird mit den folgenden Registern gesendet.

Register	Inhalt
*.a_STEUREG_CONTROLWORD[0]	Kontrollregister Steuerwort für Slave 0
*.a_STEUREG_CONTROLWORD[1]	Kontrollregister Steuerwort für Slave 1
*.a_STEUREG_CONTROLWORD[2]	Kontrollregister Steuerwort für Slave 2
*.a_STEUREG_CONTROLWORD[3]	Kontrollregister Steuerwort für Slave 3
...	...
*.a_STEUREG_CONTROLWORD[31]	Kontrollregister Steuerwort für Slave 31

(\* entspricht zum Beispiel \_CANsyncMaster\_Ctrl\_Slot\_G)

Register	Inhalt
*.a_CONTROLWORD_SL[0]	Steuerwort für Slave 0
*.a_CONTROLWORD_SL[1]	Steuerwort für Slave 1
*.a_CONTROLWORD_SL[2]	Steuerwort für Slave 2
*.a_CONTROLWORD_SL[3]	Steuerwort für Slave 3
...	...
*.a_CONTROLWORD_SL[31]	Steuerwort für Slave 31

(\* entspricht zum Beispiel \_CANsyncMaster\_Ctrl\_Slot\_G)

Um an einen CANsync-Slave ein Steuerwort zu senden, muss das Steuerwort in den Bereich ab \*.a\_CONTROLWORD\_SL[0] eingetragen werden und anschließend auf das entsprechende Kontrollregister ab \*.a\_STEUREG\_CONTROLWORD[0] 16#05 eingetragen werden. Wenn das Steuerwort gesendet wurde wird als Quittung 16#04 gemeldet.

Beispiel: Steuerwort an CANsync-Slave 3 senden:

1. Steuerwort in \*.a\_CONTROLWORD\_SL[3] eintragen.
2. In Kontrollregister \*.a\_STEUREG\_CONTROLWORD[3] 16#05 eintragen.

Quittung nach dem Senden: 16#04 auf \*.a\_STEUREG\_CONTROLWORD[3].

## 3.2 Detaillierte Informationen zum CANsync

Auf dem CANsync-Bus wird das Steuerwortkommando in ein Aktionskommando umgesetzt, bei dem in der Maske (Bitleiste) nur das Bit für den einen CANsync-Slave gesetzt ist, das Kommando byte auf 16#01 gesetzt ist und als Datenwort (DW) das Steuerwort eingetragen ist.

### Parameterkommando

Parameter können mit den folgenden Registern geschrieben und gelesen werden:

Register	Inhalt
*.a_CTRLREG_PAR_CMD_SL[0].b_STEUREG_PAR_CMD	Steuerregister Parameterkommando 0
*.a_CTRLREG_PAR_CMD_SL[0].b_STATREG_PAR_CMD	Statusregister Parameterkommando 0
*.a_CTRLREG_PAR_CMD_SL[1].b_STEUREG_PAR_CMD	Steuerregister Parameterkommando 1
*.a_CTRLREG_PAR_CMD_SL[1].b_STATREG_PAR_CMD	Statusregister Parameterkommando 1
...	...
*.a_CTRLREG_PAR_CMD_SL[31].b_STEUREG_PAR_CMD	Steuerregister Parameterkommando 31
*.a_CTRLREG_PAR_CMD_SL[31].b_STATREG_PAR_CMD	Statusregister Parameterkommando 31

(\* entspricht zum Beispiel \_CANsyncMaster\_Ctrl\_Slot\_G)

Bedeutung des Steuer- und des Statusregisters:

Bit 0	<b>Aktiv</b> muss auf 1 gesetzt werden, um Befehl zu starten wird auf 0 gesetzt, wenn Antwort empfangen wurde
Bit 1	<b>Read</b> =1: Parameter lesen
Bit 2	<b>Write</b> =1: Parameter schreiben
Bit 3	<b>Send</b> =1: Anzeige, dass Befehl gesendet wurde
Bit 4	<b>Format</b> 0: Parameter hat Format Wort 1: Parameter hat Format Doppelwort
Bit 5	<b>Fehleranzeige</b> 1: Fehler aufgetreten 0: kein Fehler
Bit 6	<b>Busy</b> = 1 CANsync-Slave bearbeitet Auftrag, Daten noch nicht gültig
Bit 7	reserviert

Register	Inhalt
*.a_DATA_PARAMETER[0].d_PAR_VALUE	Daten CANsync-Slave 0
*.a_DATA_PARAMETER[0].w_PAR_NR	Parameter Nummer 0
*.a_DATA_PARAMETER[0].w_SUBSL_ADR	reserviert

Register	Inhalt
*.a_DATA_PARAMETER[1].d_PAR_VALUE *.a_DATA_PARAMETER[1].w_PAR_NR *.a_DATA_PARAMETER[1].w_SUBSL_ADR	Daten CANsync-Slave 1 Parameternummer 1 reserviert
...	...
*.a_DATA_PARAMETER[31].d_PAR_VALUE *.a_DATA_PARAMETER[31].w_PAR_NR *.a_DATA_PARAMETER[31].w_SUBSL_ADR	Daten CANsync-Slave 31 Parameternummer 31 reserviert

(\* entspricht zum Beispiel \_CANsyncMaster\_Ctrl\_Slot\_G)

Unter „Daten CANsync Slave x“ wird der Parameterwert eingetragen bzw. empfangen. Dieser Wert kann ein Wort oder Doppelwort sein.

Die Parameternummer wählt einen Parameter in dem angesprochenen CANsync-Slave aus. Die Bedeutung des Parameters muss aus der Beschreibung des CANsync-Slaves entnommen werden.

#### Ablauf eines Parameterzugriffs

Die Parameterkommandos können sowohl im CANsync Event-Task als auch im restlichen Programm verwendet werden. Zur Sicherung der Datenkonsistenz erfolgt die Auftragserteilung und die Auswertung der Rückantwort über das Steuerregister und das Statusregister. Um eine konfliktfreie Durchführung des Auftrags zu gewährleisten, muss vom Applikationsprogramm aus immer zuerst das Steuerregister und dann das Statusregister gelesen werden und wenn ein neuer Wert geschrieben wird, immer zuerst das Statusregister und dann das Steuerregister gesetzt werden. (Für die CANsync-Anschaltung ist das Steuerregister das entscheidende Register.)

Die folgende Erläuterung bezieht sich in den Adressen auf den CANsync-Slave 0.

Für einen Parameter-Schreibauftrag wird der **Datenwert** in \*.a\_DATA\_PARAMETER[0].d\_PAR\_VALUE (Doppelwort) und die **Parameternummer** in \*.a\_DATA\_PARAMETER[0].w\_PAR\_NR eingetragen.

Anschließend trägt man im Statusregister

\*.a\_CTRLREG\_PAR\_CMD\_SL[0].b\_STATREG\_PAR\_CMD und im Steuerregister

\*.a\_CTRLREG\_PAR\_CMD\_SL[0].b\_STEUREG\_PAR\_CMD den Wert 16#05 ein, wenn ein Wortparameter beschrieben wird, bzw. 16#15, wenn ein Doppelwort-Parameter beschrieben wird.

Wenn der Auftrag (entsprechend der [Konfigurierung Kommandokanal](#) ab Seite 62) von der CAN-Anschaltung übernommen wurde, wird als Bestätigung das Bit 3 (Send) im Steuerregister und im Statusregister gesetzt.

Wenn der CANsync-Slave den Auftrag akzeptiert, aber noch nicht beendet hat, wird das Bit 6 (Busy) gesetzt.

Falls im CANsync-Slave bei der Ausführung des Auftrags ein Fehler auftritt, wird das Bit 5 (Fehleranzeige) gesetzt, das Bit 6 (Busy) und das Bit 0 (Aktiv) wird gelöscht, dann ist die Fehlernummer im Datenbereich \*.a\_DATA\_PARAMETER[0].d\_PAR\_VALUE zu lesen.

Wenn der CANsync-Slave den Auftrag beendet hat, wird das Bit 6 (Busy) und das Bit 0 (Aktiv) gelöscht (im Byte ergibt sich 16#04 für Wort schreiben, bzw. 16#14 für Doppelwort schreiben).

Die Auswertung im Applikationsprogramm kann sich darauf beschränken, das Bit 0 abzufragen, solange es gesetzt ist, wird der Auftrag noch bearbeitet, wenn es zurückgesetzt wird, ist der Auftrag beendet.

Es kann sein, dass das Bit 6 (Busy) nicht gesetzt wird, wenn der CANsync-Slave den Auftrag sofort beantworten kann.

Der Parameter-Leseauftrag läuft dazu ähnlich ab, nur mit dem Unterschied, dass der Befehl 16#03 für Wort lesen und 16#13 für Doppelwort lesen lautet und dass die Daten nicht vor dem Auftrag eingetragen werden, sondern nachdem das Bit 0 (Aktiv) gelöscht wurde, zum Auslesen zur Verfügung stehen. Die Quittung im Byte gesehen lautet dann 16#02 für Wortzugriff und 16#12 für Doppelwortzugriff. Das Format wird aus der Antwort des CANsync-Slaves übernommen.

### Fehlernummer des CANsync-Slaves

Wert	Bedeutung
16#0000	Kein Fehler aufgetreten
16#FFFF	Fehler aufgetreten
16#FFFE	Wert kleiner Minimalwert
16#FFFD	Wert größer Maximalwert
16#FFFC	Element nicht veränderbar
16#FFFB	Element nicht vorhanden
16#FFFA	Daten nicht verfügbar (z. B. in Bearbeitung)
16#FFF9	Fehler beim Datenformat

### Up/Download Blockbereich

Upload- und Download-Aufträge können mit den folgenden Registern gesteuert werden:

Register	Inhalt
*.b_STEUREG_UPDOWNBLK1	Steuerregister Up/down-Block 1
*.b_STATREG_UPDOWNBLK1	Statusregister Up/down-Block 1
*.b_SL_NR_UPDOWNBLK1	CANsync-Slavenummer Up/down-Block 1
*.w_ERR_NR_UPDOWNBLK1	Fehlernummer Up/down-Block 1
*.w_SUBSL_NR_UPDOWNBLK1	Subslavenummer Up/down-Block 1
*.d_BASE_ADR_UPDOWNBLK1	Basisadresse Up/down-Block 1
*.w_LENGTH_UPDOWNBLK1	Länge in Bytes Up/down-Block 1
*.w_COUNTER_UPDOWNBLK1	Zähler Up/down-Block 1
*.a_DATA_UPDOWNBLK1[0 bis 74]	Datenblock Up/down-Block 1 (75 Doppelworte)

(\* entspricht zum Beispiel \_CANsyncMaster\_Ctrl\_Slot\_G)

### Bedeutung des Steuer- und des Statusregisters

Bit 0	<b>Aktiv</b> muss auf 1 gesetzt werden, um Befehl zu starten wird auf 0 gesetzt, wenn Befehl abgearbeitet wird										
Bit 1	<b>Send</b> = 1: Anzeige, dass Befehl gesendet wurde										
Bit 2	<b>Mode</b> <table border="0"> <tr> <td>Bit3</td> <td>Bit2</td> </tr> <tr> <td>0</td> <td>0: reserviert</td> </tr> <tr> <td>0</td> <td>1: Initialisierung</td> </tr> <tr> <td>1</td> <td>0: laufender Up/Download</td> </tr> <tr> <td>1</td> <td>1: Blockende</td> </tr> </table>	Bit3	Bit2	0	0: reserviert	0	1: Initialisierung	1	0: laufender Up/Download	1	1: Blockende
Bit3		Bit2									
0	0: reserviert										
0	1: Initialisierung										
1	0: laufender Up/Download										
1	1: Blockende										
Bit 3											
Bit 4	<b>Up/Download</b> 0: Upload 1: Download										
Bit 5	<b>Fehleranzeige</b> 1: Fehler aufgetreten 0: kein Fehler										
Bit 6	<b>Busy</b> = 1: CANsync-Slave bearbeitet Auftrag, Daten noch nicht gültig										
Bit 7	<b>Reset</b> Abbruch eines Auftrags										

### Ablauf eines Up/Download-Auftrags im Block 1

Die Up/Downloadkommandos können sowohl im CANsync Event-Task als auch im restlichen Programm verwendet werden. Zur Sicherung der Datenkonsistenz erfolgt die Auftragserteilung und die Auswertung der Rückantwort über das Steuerregister und das Statusregister. Um eine konfliktfreie Durchführung des Auftrags zu gewährleisten, muss vom Applikationsprogramm aus immer zuerst das Steuerregister und dann das Statusregister gelesen werden und wenn ein neuer Wert geschrieben wird, immer zuerst das Statusregister und dann das Steuerregister gesetzt werden. (Für die CANsync-Anschaltung ist das Steuerregister das entscheidende Register.)

Für einen Download wird zuerst der Datenblockbereich \*.a\_DATA\_UPDOWNBLK1[0] bis \*.a\_DATA\_UPDOWNBLK1[74] mit den zu übertragenden Daten gefüllt. Die maximale Blocklänge sind 300 Bytes (75 Doppelworte), die mit einem Downloadauftrag gesendet werden können.

Die Nummer des CANsync-Slaves, der den Download empfangen soll wird im \*.b\_SL\_NR\_UPDOWNBLK1 und die Basisadresse im \*.d\_BASE\_ADR\_UPDOWNBLK1 eingetragen.

Die Basisadressen 16#0000\_0000 bis 16#0000\_00FF sind für Aufträge des Betriebssystems reserviert.

Anschließend trägt man im Statusregister \*.b\_STATREG\_UPDOWNBLK1 und im Steuerregister \*.b\_STEUREG\_UPDOWNBLK1 den Wert 16#15 ein.

Die CANsync-Anschaltung wird nun versuchen den gesamten Block zum CANsync-Slave zu übertragen. Die aufeinanderfolgenden Modi werden automatisch generiert. Der Fortschritt kann im Byte-Zähler im \*.w\_COUNTER\_UPDOWNBLK1 abgelesen werden.

Wenn der Auftrag (siehe [►Konfigurierung Kommandokanal◄](#) ab Seite 62) von der CANsync-Anschaltung übernommen wurde, wird als Bestätigung das Bit 1 (Send) im Steuerregister und im Statusregister gesetzt.

Wenn der CANsync-Slave den Auftrag akzeptiert, aber noch nicht beendet hat, wird das Bit 6 (Busy) gesetzt.

Falls im CANsync-Slave bei der Ausführung des Auftrags ein Fehler auftritt, wird das Bit 5 (Fehleranzeige) gesetzt, das Bit 6 (Busy) und das Bit 0 (Aktiv) wird gelöscht, dann ist die Fehlernummer im \*.b\_SL\_NR\_UPDOWNBLK1 zu lesen.

Wenn der CANsync-Slave den Auftrag beendet hat, wird das Bit 6 (Busy) und das Bit 0 (Aktiv) gelöscht (im Steuer- und Statusregister ergibt sich 16#1C).

Die Auswertung im Applikationsprogramm kann sich darauf beschränken, das Bit 0 (Aktiv) abzufragen. Solange es gesetzt ist, wird der Auftrag noch bearbeitet, wenn es zurückgesetzt wird, ist der Auftrag beendet.

Es kann sein, dass das Bit 6 (Busy) nicht gesetzt wird, wenn der CANsync-Slave den einzelnen Modi des Auftrags sofort beantworten kann.

Der Upload-Auftrag läuft dazu ähnlich ab, nur mit dem Unterschied, dass der Befehl 16#05 lautet und dass die Daten nicht vor dem Auftrag eingetragen werden, sondern nachdem das Bit 0 (Aktiv) gelöscht wurde, zum Auslesen zur Verfügung stehen. Die Quitting im Steuer- und Statusregister gesehen lautet dann 16#0C.

Der Abbruch eines Up/Download-Auftrags ist möglich, indem man das Bit 7 (Reset) auf TRUE setzt. An den CANsync-Slave wird dann ein Upload-Initialisierungstelegramm mit Adresse = 0 und Länge = 0 gesendet.

### Fehlernummern Up/Download

Fehlernummer	Bedeutung
16#0001	CANsync-Slave quittiert falsche Blocknummer
16#0002	eingetragenen Länge > 300 Bytes
...	
16#0100	CANsync-Slave erwartet Block mit Nummer, die im Zähler eingetragen ist
16#0101	CANsync-Slave erwartet Block-Ende
16#0102	CANsync-Slave erwartet noch nicht Block-Ende
16#0103	CANsync-Slave bricht Up/Download ab
16#0104	Up/Download nicht möglich
16#0105	Basisadresse nicht erlaubt
16#0106	reserviert
16#0107	Blocklänge > maximale Blocklänge des CANsync-Slaves
16#0108	Telegramm-Modedefehler (in dieser Phase nicht erlaubter Mode)



### 3.2.2.7 CANsync-Slavestatus

Register	Inhalt
*.a_STAT_SL[0]	CANsync-Slavestatus 0
*.a_STAT_SL[1]	CANsync-Slavestatus 1
*.a_STAT_SL[2]	CANsync-Slavestatus 2
...	...
*.a_STAT_SL[31]	CANsync-Slavestatus 31

(\* entspricht zum Beispiel \_CANsyncMaster\_Ctrl\_Slot\_G)

Bedeutung CANsync-Slavestatus:

Bit 0	<b>Response</b> = 1: CANsync-Slave meldet sich
Bit 1	<b>Synchron</b> = 1: CANsync-Slave ist synchronisiert
Bit 2	reserviert
Bit 3	
Bit 4	
Bit 5	
Bit 6	
Bit 7	

Im Einrichtbetrieb (siehe [► Initialisierung](#) ab Seite 49) wird der Slavestatus von allen initialisierten CANsync-Slaves abgefragt und hier eingetragen. Das Bit 0 (Response) zeigt an, dass der CANsync-Slave sich am CANsync-Bus gemeldet hat. Das Bit 1 (Synchron) zeigt an, dass der CANsync-Slave synchronisiert ist und der Synchronbetrieb gestartet werden kann.

### 3.2.3 Registerstruktur und Funktion des Optionsmoduls CANsync-Slave für b maXX PLC

Im folgenden wird die Registerstruktur des Kommunikations-RAM im Optionsmodul CANsync-Slave für b maXX PLC erläutert.

Um im PROPROG wt II Projekt auf die Register des Kommunikations-RAM zugreifen zu können, sind Datentypen definiert, die die Registerstruktur abbilden. Mit diesen Datentypen werden Variablen deklariert, die auf die Basisadresse der CANsync-Anschaltung gelegt werden.

Anschließend ist es möglich über die Strukturelemente der deklarierten Variablen auf die Register des Kommunikations-RAM zuzugreifen.

Bei der Initialisierung der CANsync-Slave-Anschaltung haben die Register im Kommunikations-RAM eine andere Bedeutung als nach der Initialisierung, im zyklischen Betrieb.

Deshalb gibt es für die Initialisierung die Struktur

## 3.2 Detaillierte Informationen zum CANsync

CANsync\_PLC\_INIT\_BMSTRUCT

und für den zyklischen Betrieb die Struktur

CANsync\_PLC\_SL\_BMSTRUCT

Diese Strukturen sind ab der Bibliothek BM\_TYPES\_20bd03 definiert. Nachdem die Bibliothek BM\_TYPES\_20bd03 im Projekt eingebunden ist, stehen die Datentypen zur Verfügung.

Diese Strukturen enthalten

- 8-Bit-Elemente,
- 16-Bit-Elemente,
- 32-Bit-Elemente,
- Strukturen aus den o.g. Elementen
- Felder (ARRAY) und Strukturen aus den o.g. Elementen und Strukturen

Den in einer Struktur verwendeten Datentypen (8-, 16-, 32-Bit-Elemente, Strukturen und Feldern) wurden Kurzbezeichnungen vorangefügt. Dies dient der Übersichtlichkeit bei der Verwendung der Strukturen in der Programmierung.

Datentyp	Kurzbezeichnung	Anzahl der Bits
BYTE	b	8
WORD	w	16
DWORD (double word)	d	32
SINT (short integer)	si	8
DINT (double integer)	di	32
USINT (unsigned short integer)	us	8
UINT (unsigned integer)	u	16
UDINT (unsigned double integer)	ud	32
STRUCT	_ (underline)	-
ARRAY	a	-

Weitere, nicht in den Strukturen verwendete Datentypen sind:

Datentyp	Kurzbezeichnung	Anzahl der Bits
BOOL (bit)	x	1
TIME	t	-

### 3.2.3.1 Erläuterung zur Deklaration der globalen Variablen

Für die Initialisierung sind im Template "BM4-O-PLC01" bereits globale Variablen vom Datentyp CANsync\_PLC\_INIT\_BMSTRUCT für jeden Optionsmodul-Steckplatz deklariert.

Das sind die globalen Variablen

```

_CANsyncSlave_Init_Slot_G AT %MB3.2001792 :
                                CANsync_PLC_INIT_BMSTRUCT;
_CANsyncSlave_Init_Slot_H AT %MB3.3001792 :
                                CANsync_PLC_INIT_BMSTRUCT;
_CANsyncSlave_Init_Slot_J AT %MB3.4001792 :
                                CANsync_PLC_INIT_BMSTRUCT;
_CANsyncSlave_Init_Slot_K AT %MB3.5001792 :
                                CANsync_PLC_INIT_BMSTRUCT;
_CANsyncSlave_Init_Slot_L AT %MB3.6001792 :
                                CANsync_PLC_INIT_BMSTRUCT;
_CANsyncSlave_Init_Slot_M AT %MB3.7001792 :
                                CANsync_PLC_INIT_BMSTRUCT;

```

Sollten diese Variablen nicht mehr im PROPROGRAMM II Projekt vorhanden sein, legen Sie, je nach Steckplatz (Slot G bis M), die globale Variable `_CANsyncSlave_Init_Slot_G` (bis `_CANsyncSlave_Init_Slot_M`) vom Datentyp `CANsync_PLC_INIT_BMSTRUCT` an.

Diese Variable muß bei der Deklaration der globalen Variablen auf die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul gelegt werden.

Die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Slave für b maXX PLC (BM4-O-CAN-05, Dip-Schalter 9 = ON) in Abhängigkeit vom Steckplatz ist:

Steckplatz (Slot)	Basisadresse der CANsync-Anschaltung
G	%MB3.2001792
H	%MB3.3001792
J	%MB3.4001792
K	%MB3.5001792
L	%MB3.6001792
M	%MB3.7001792

Beispiel:

CANsync-Anschaltung auf Optionsmodul CANsync-Slave für b maXX PLC im Steckplatz G des b maXX

```

_CANsyncSlave_Init_Slot_G AT %MB3.2001792 :
                                CANsync_PLC_INIT_BMSTRUCT;

```

dabei ist:

<code>_CANsyncSlave_Init_Slot_G</code>	der Variablenname mit der Datentypkurzbezeichnung "_" für Struct
<code>CANsync_PLC_INIT_BMSTRUCT</code>	der Datentyp
<code>%MB3.2001792</code>	die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Slave für b maXX PLC im Steckplatz G des b maXX

## 3.2 Detaillierte Informationen zum CANsync

Für den zyklischen Betrieb sind im Template "BM4-O-PLC01" bereits globale Variablen vom Datentyp CANsync\_PLC\_SL\_BMSTRUCT für jeden Optionsmodul-Steckplatz deklariert.

Das sind die globalen Variablen

```
_CANsyncSlave_Ctrl_Slot_G AT %MB3.2001792 :
                                CANsync_PLC_SL_BMSTRUCT;
_CANsyncSlave_Ctrl_Slot_H AT %MB3.3001792 :
                                CANsync_PLC_SL_BMSTRUCT;
_CANsyncSlave_Ctrl_Slot_J AT %MB3.4001792 :
                                CANsync_PLC_SL_BMSTRUCT;
_CANsyncSlave_Ctrl_Slot_K AT %MB3.5001792 :
                                CANsync_PLC_SL_BMSTRUCT;
_CANsyncSlave_Ctrl_Slot_L AT %MB3.6001792 :
                                CANsync_PLC_SL_BMSTRUCT;
_CANsyncSlave_Ctrl_Slot_M AT %MB3.7001792 :
                                CANsync_PLC_SL_BMSTRUCT;
```

Sollten diese Variablen nicht mehr im PROPROG wt II Projekt vorhanden sein, legen Sie, je nach Steckplatz (Slot G bis M), die globale Variable `_CANsyncSlave_Ctrl_Slot_G` (bis `_CANsyncSlave_Ctrl_Slot_M`) vom Datentyp `CANsync_PLC_SL_BMSTRUCT` an.

Diese Variable muß bei der Deklaration der globalen Variablen auf die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul gelegt werden.

Die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Slave für b maXX PLC (BM4-O-CAN-05, Dip-Schalter 9 = ON) in Abhängigkeit vom Steckplatz ist:

Steckplatz (Slot)	Basisadresse der CANsync-Anschaltung
G	%MB3.2001792
H	%MB3.3001792
J	%MB3.4001792
K	%MB3.5001792
L	%MB3.6001792
M	%MB3.7001792

Beispiel:

CANsync-Anschaltung auf Optionsmodul CANsync-Slave im Steckplatz G des b maXX

```
_CANsyncSlave_Ctrl_Slot_G AT %MB3.2001792 :
                                CANsync_PLC_SL_BMSTRUCT;
```

dabei ist:

`_CANsyncSlave_Ctrl_Slot_G` der Variablenname mit der Datentypkurzbezeichnung "\_" für Struct

`CANsync_PLC_SL_BMSTRUCT` der Datentyp

MB3.2001792

die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Slave im Steckplatz G des b maXX



#### HINWEIS

In den nachfolgenden Tabellen wird der Variablenname durch \* ersetzt. Weiterhin wird davon ausgegangen, dass das Optionsmodul CANsync-Slave für b maXX PLC im Steckplatz G gesteckt ist.

Auf das Register `*.w_CPU_CONTROL` greift man demzufolge über `_CANsyncSlave_Init_Slot_G.w_CPU_CONTROL` zu, auf `*.w_OPTION_STATUS` greift man über `_CANsyncSlave_Init_Slot_G.w_OPTION_STATUS` zu.

Dabei ist:

<code>_CANsyncSlave_Init_Slot_G</code>	der Variablenname mit der Datentypkurzbezeichnung „_“ für STRUCT
<code>w_CPU_CONTROL</code>	das Steuerregister der CANsync-Anschaltung mit der Datentypkurzbezeichnung „w“ für WORD

Die Register `*.w_CPU_CONTROL` und `*.w_OPTION_STATUS` können auch über die Struktur für den zyklischen Betrieb angesprochen werden. Der Zugriff ist dann über

`_CANsyncSlave_Ctrl_Slot_G.w_CPU_CONTROL` und `_CANsyncSlave_Ctrl_Slot_G.w_OPTION_STATUS` möglich.

Dabei ist

<code>_CANsyncSlave_Ctrl_Slot_G</code>	der Variablenname mit der Datentypkurzbezeichnung „_“ für STRUCT
<code>w_OPTION_STATUS</code>	das Statusregister der CANsync-Anschaltung mit der Datentypkurzbezeichnung „w“ für WORD

Beispiel für den Zugriff auf ein Element eines Feldes, das in der Struktur verwendet wird: laut Tabelle: `*.a_WR_VALUE_RECEIVE[3]`

Zugriff: `_CANsyncSlave_Ctrl_Slot_G.a_WR_VALUE_RECEIVE[3]`

Dabei ist

<code>_CANsyncSlave_Ctrl_Slot_G</code>	der Variablenname mit der Datentypkurzbezeichnung „_“ für STRUCT
--	--

`a_WR_VALUE_RECEIVE[3]`

das Register für den Sollwert 3 mit der Datentypkurzbezeichnung „a“ für ARRAY. Der Datentyp der Elemente des Feldes (der Sollwerte) wird der entsprechenden Tabelle und der Beschreibung entnommen.

Beispiel für den Zugriff auf ein Element eines zweidimensionalen Feldes, das in der Struktur verwendet wird:

laut Tabelle: `*.a_RD_VALUE[5][7]`

Zugriff: `_CANsyncSlave_Ctrl_Slot_G.a_RD_VALUE[5][7]`

Dabei ist

`_CANsyncSlave_Ctrl_Slot_G`

der Variablenname mit der Datentypkurzbezeichnung „\_“ für STRUCT

`a_RD_VALUE[5][7]`

das Register für den Istwert 7 des CANsync-Slaves 5 mit der Datentypkurzbezeichnung „a“ für ARRAY. Der Datentyp der Elemente des Feldes (der Sollwerte) wird der entsprechenden Tabelle und der Beschreibung entnommen.

Beispiel für den Zugriff auf ein Element einer (Sub-) Struktur, die selbst Element eines Feldes ist, das in der Struktur verwendet wird:

laut Tabelle: `*.a_CFG_RDC_WORD[31].b_CFG_RDC2_WORD3`

Zugriff:

`_CANsyncSlave_Ctrl_Slot_G.a_CFG_RDC_WORD[31].b_CFG_RDC2_WORD3`

Dabei ist

`_CANsyncSlave_Ctrl_Slot_G`

der Variablenname mit der Datentypkurzbezeichnung „\_“ für STRUCT

`a_CFG_RDC_WORD[31]`

das Feld mit den Konfigurationsdaten für das Mapping der Worte der Istwert-Telegramme des CANsync-Slaves 31 mit der Datentypkurzbezeichnung „a“ für ARRAY

`b_CFG_RDC2_WORD3`

das Register für die Konfigurationsdaten für das Mapping des 3. Wortes des Istwert-Telegramms 2 (des CANsync-Slaves 31) mit der Datentypkurzbezeichnung „b“ für BYTE

Beispiel für den Zugriff auf ein Element einer (Sub-) Struktur, die in der Struktur verwendet wird:

laut Tabelle: `*._CFG_WRC_WORD.b_CFG_WRC1_WORD0`

Zugriff: `_CANsyncSlave_Ctrl_Slot_G._CFG_WRC_WORD.b_CFG_WRC1_WORD0`

Dabei ist

<code>_CANsyncSlave_Ctrl_Slot_G</code>	der Variablenname mit der Datentypkurzbezeichnung „_“ für STRUCT
<code>_CFG_WRC_WORD</code>	die Struktur mit den Konfigurationsdaten für das Mapping der Worte der Sollwert-Telegramme mit der Datentypkurzbezeichnung „_“ für STRUCT
<code>b_CFG_WRC1_WORD0</code>	das Register für die Konfigurationsdaten für das Mapping des 0. Wortes des Sollwert-Telegramms 1 mit der Datentypkurzbezeichnung „b“ für BYTE

### 3.2.3.2 Allgemeine Register der CANsync-Anschaltung

Register	Inhalt
<code>*.w_CANsync_STATUS</code>	CANsync-Status
<code>*.w_SLAVE_NUMBER</code>	über Dip-Schalter eingestellte Slave-Nummer
<code>*.i_SW1_NR</code>	Karten-Softwarenummer
<code>*.i_SW1_RELEASE</code>	Softwarestand inkompatibel und kompatibel

#### CANsync-Status

Bei jedem Durchlaufen des CANsync-Prozessorzyklusses wird der CANsync-Status auf `*.w_CANsync_STATUS` ausgegeben.

Bedeutung:

Bit-Nr.	Bedeutung (Bit = TRUE)
0	reserviert
1	Overrun: eine CANsync-Nachricht konnte nicht empfangen werden
2	CANsync-Sendepuffer ist frei
3	CANsync-Sendeauftrag wurde erfolgreich durchgeführt
4	es wird gerade eine CANsync-Meldung empfangen
5	es wird gerade eine CANsync-Meldung gesendet
6	Fehler vorhanden (Warnung)
7	CANsync-Knoten ist deaktiviert (BUS-off)
8-15	reserviert

#### Slave-Nummer

Im Register `*.w_SLAVE_NUMBER` wird die über die Dip-Schalter (siehe ▶ Betriebsanleitung Optionsmodul CANsync-Slave für b maXX ◀) eingestellte Slave-Nummer angezeigt.

### Softwarenummer und Softwarestand

Im Register \*.i\_SW1\_NR wird die Nummer der CANsync-Software auf dem Optionsmodul CANsync-Slave für b maXX PLC angezeigt.

Im Register \*.i\_SW1\_RELEASE wird der inkompatible und der kompatible Stand der CANsync-Software auf dem Optionsmodul CANsync-Slave für b maXX PLC angezeigt.

### 3.2.3.3 Initialisierung

Für die Initialisierung sind im Template "BM4-O-PLC01" bereits globale Variablen vom Datentyp CANsync\_PLC\_INIT\_BMSTRUCT für jeden Optionsmodul-Steckplatz deklariert.

Das sind die globalen Variablen

```
_CANsyncSlave_Init_Slot_G  AT  %MB3.2001792  :  
                             CANsync_PLC_INIT_BMSTRUCT;  
_CANsyncSlave_Init_Slot_H  AT  %MB3.3001792  :  
                             CANsync_PLC_INIT_BMSTRUCT;  
_CANsyncSlave_Init_Slot_J  AT  %MB3.4001792  :  
                             CANsync_PLC_INIT_BMSTRUCT;  
_CANsyncSlave_Init_Slot_K  AT  %MB3.5001792  :  
                             CANsync_PLC_INIT_BMSTRUCT;  
_CANsyncSlave_Init_Slot_L  AT  %MB3.6001792  :  
                             CANsync_PLC_INIT_BMSTRUCT;  
_CANsyncSlave_Init_Slot_M  AT  %MB3.7001792  :  
                             CANsync_PLC_INIT_BMSTRUCT;
```

Sollten diese Variablen nicht mehr im PROPROGRAMM vorhanden sein, legen Sie, je nach Steckplatz (Slot G bis M), die globale Variable `_CANsyncSlave_Init_Slot_G` (bis `_CANsyncSlave_Init_Slot_M`) vom Datentyp `CANsync_PLC_INIT_BMSTRUCT` an.

Diese Variable muß bei der Deklaration der globalen Variablen auf die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul gelegt werden.

Die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Slave für b maXX PLC (BM4-O-CAN-05, Dip-Schalter 9 = ON) in Abhängigkeit vom Steckplatz ist:

Steckplatz (Slot)	Basisadresse der CANsync-Anschaltung
G	%MB3.2001792
H	%MB3.3001792
J	%MB3.4001792
K	%MB3.5001792
L	%MB3.6001792
M	%MB3.7001792



Beispiel:

CANsync-Anschaltung auf Optionsmodul CANsync-Slave für b maXX PLC im Steckplatz G des b maXX

```
_CANsyncSlave_Init_Slot_G AT %MB3.2001792 :
                                CANsync_PLC_INIT_BMSTRUCT;
```

dabei ist:

`_CANsyncSlave_Init_Slot_G` der Variablenname mit der Datentypkurzbezeichnung "\_" für Struct

`CANsync_PLC_INIT_BMSTRUCT` der Datentyp

`%MB3.2001792` die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Slave für b maXX PLC im Steckplatz G des b maXX

Zusätzlich konfiguriert man für den Synchron-Betrieb:

Bedeutung		Register	Wert
Baudrate	z. B.: 1 MBit/s	*.b_BT_0 *.b_BT_1	16#40 16#34
CANsync- Intervall	z. B.: 1 ms	*.b_TIME_PATTERN	16#01
Acceptance Code	alle Telegramme	*.b_AC	16#FF
Acceptance Mask		*.b_AM	16#FF
Output-Control	16#FA	*.b_OUTPUT_CONTROL	16#FA
Clock-Divider	16#07	*.b_CLOCK_DIVIDER	16#07
Slave/Master	Slave	*.b_MA_SL_MODE	16#01

Die Betriebsarteneinstellung erfolgt über das Register \*.w\_CPU\_CONTROL. Die Anzeige der gerade aktiven Betriebsart erfolgt im Register \*.w\_OPTION\_STATUS. Auch nach dem erfolgreichen Starten einer Betriebsart kann die Betriebsart wieder geändert werden.

Register	Inhalt
*.w_CPU_CONTROL	Steuerregister CANsync-Anschaltung
*.w_OPTION_STATUS	Statusregister CANsync-Anschaltung

(\* entspricht zum Beispiel bei der Initialisierung `_CANsyncSlave_Init_Slot_G`, nach der Initialisierung zum Beispiel `_CANsyncSlave_Ctrl_Slot_G`)

Steuerregister CANsync-Anschaltung	Bedeutung
16#0000	Neuanlauf
16#0001	Test Handshake
16#0002	Initialisierungsdaten übernehmen
16#0012	reserviert

## 3.2 Detaillierte Informationen zum CANsync

Steuerregister CANsync- Anschaltung	Bedeutung
16#0013	reserviert
16#0020	Synchron-Betrieb starten
16#0040	Aktiven Betrieb freigeben
16#0080	(Bit 7 = TRUE) Reset CAN-Controller

Statusregister CANsync- Anschaltung	Bedeutung
16#0001	Anlauf
16#0002	Warten auf Initialisierungsdaten übernehmen
16#0003	Warten auf Start
16#0011	reserviert
16#0012	reserviert
16#0013	Synchron-Betrieb CANsync-Slave wird eingerichtet
16#0020	Synchron-Betrieb CANsync-Slave ist aktiv
16#0041	reserviert
16#0042	reserviert
16#0043	<i>Synchron-Betrieb CANsync-Master wird eingerichtet</i>
16#0080	<i>Synchron-Betrieb CANsync-Master ist aktiv</i>

Die Initialisierung wird mit den Kommandos 16#0000, 16#0001 und 16#0002 auf \*.w\_CPU\_CONTROL durchgeführt. Damit wird der Einrichtbetrieb gestartet. Anschließend kann der aktive Betrieb freigegeben werden. Dies geschieht durch Setzen des Bit 6 in \*.w\_CPU\_CONTROL (\*.w\_CPU\_CONTROL = 16#0040).

Wenn man das Bit 7 von \*.w\_CPU\_CONTROL setzt, wird der CANsync-Controller zurückgesetzt und das Bit wieder gelöscht. Dadurch kann der BUS-OFF-Zustand des CANsync-Controllers zurückgesetzt werden und wieder CANsync-Telegramme gesendet und empfangen werden. Die Anzeige über den BUS-OFF-Zustand erfolgt in \*.w\_CANsync\_STATUS (siehe [Allgemeine Register der CANsync-Anschaltung](#) ab Seite 77).

In der folgenden Tabelle sind die Register angegeben, die in der Initialisierung bedient werden.

Register	Inhalt
*.b_MA_SL_MODE	Betriebsart: Master / Slave (SYNC-OUT / SYNC-IN)
*.b_AC	Acceptance Code des CANsync-Controllers
*.b_AM	Acceptance Mask des CANsync-Controllers
*.b_BT_0	Bit-Timing-Register 0 des CANsync-Controllers
*.b_BT_1	Bit-Timing-Register 1 des CANsync-Controllers
*.b_OUTPUT_CONTROL	Output-Control-Register des CANsync-Controllers
*.b_CLOCK_DIVIDER	Clock-Divider des CANsync-Controllers
*.b_TIME_PATTERN	CANsync-Intervall in ms

(\* entspricht zum Beispiel bei der Initialisierung der Struktur \_CANsyncSlave\_Init\_Slot\_G;)

### 3.2.3.4 Sollwerte

Das Empfangen der Sollwerte erfolgt in der CANsync Event-Task.

Register	Inhalt
*.b_CTRLREG_WRC1	Statusregister Sollwertkanal 1
*.b_CTRLREG_WRC2	Statusregister Sollwertkanal 2
*.b_CTRLREG_WRC3	reserviert
*.b_CTRLREG_WRC4	reserviert

(\* entspricht zum Beispiel \_CANsyncSlave\_Ctrl\_Slot\_G)

Im Statusregister der Sollwertkanäle wird mit 16#02 gemeldet, dass das jeweilige Sollwert-Telegramm empfangen wurde.

Die Konfigurierung der Sollwert-Telegramme erfolgt mit den folgenden Registern.

Register	Inhalt
*._CFG_WRC_WORD.b_CFG_WRC1_WORD0	Konfigurierung Sollwert-Telegramm 1 Wort 0
*._CFG_WRC_WORD.b_CFG_WRC1_WORD1	Konfigurierung Sollwert-Telegramm 1 Wort 1
*._CFG_WRC_WORD.b_CFG_WRC1_WORD2	Konfigurierung Sollwert-Telegramm 1 Wort 2
*._CFG_WRC_WORD.b_CFG_WRC1_WORD3	Konfigurierung Sollwert-Telegramm 1 Wort 3
*._CFG_WRC_WORD.b_CFG_WRC2_WORD0	Konfigurierung Sollwert-Telegramm 2 Wort 0
*._CFG_WRC_WORD.b_CFG_WRC2_WORD1	Konfigurierung Sollwert-Telegramm 2 Wort 1
*._CFG_WRC_WORD.b_CFG_WRC2_WORD2	Konfigurierung Sollwert-Telegramm 2 Wort 2
*._CFG_WRC_WORD.b_CFG_WRC2_WORD3	Konfigurierung Sollwert-Telegramm 2 Wort 3
*._CFG_WRC_WORD.b_CFG_WRC3_WORD0	reserviert
*._CFG_WRC_WORD.b_CFG_WRC3_WORD1	reserviert
*._CFG_WRC_WORD.b_CFG_WRC3_WORD2	reserviert
*._CFG_WRC_WORD.b_CFG_WRC3_WORD3	reserviert
*._CFG_WRC_WORD.b_CFG_WRC4_WORD0	reserviert
*._CFG_WRC_WORD.b_CFG_WRC4_WORD1	reserviert
*._CFG_WRC_WORD.b_CFG_WRC4_WORD2	reserviert
*._CFG_WRC_WORD.b_CFG_WRC4_WORD3	reserviert

(\* entspricht zum Beispiel \_CANsyncSlave\_Ctrl\_Slot\_G)

Mit der Konfigurierung gibt man an, welcher Sollwert an welcher Stelle (...\_WORD0, ..., ...\_WORD3) im Sollwert-Telegramm (...\_WRC1\_..., ...\_WRC2\_...) gelesen wird.

#### Bedeutung

Bit 0	<b>Belegt</b> Wenn = 1, wird dieses Wort des Telegramms verwendet
Bit 1	<b>Highword/Lowword</b> Wenn = 1, wird das Highword des Sollwertes eingetragen

## 3.2 Detaillierte Informationen zum CANsync

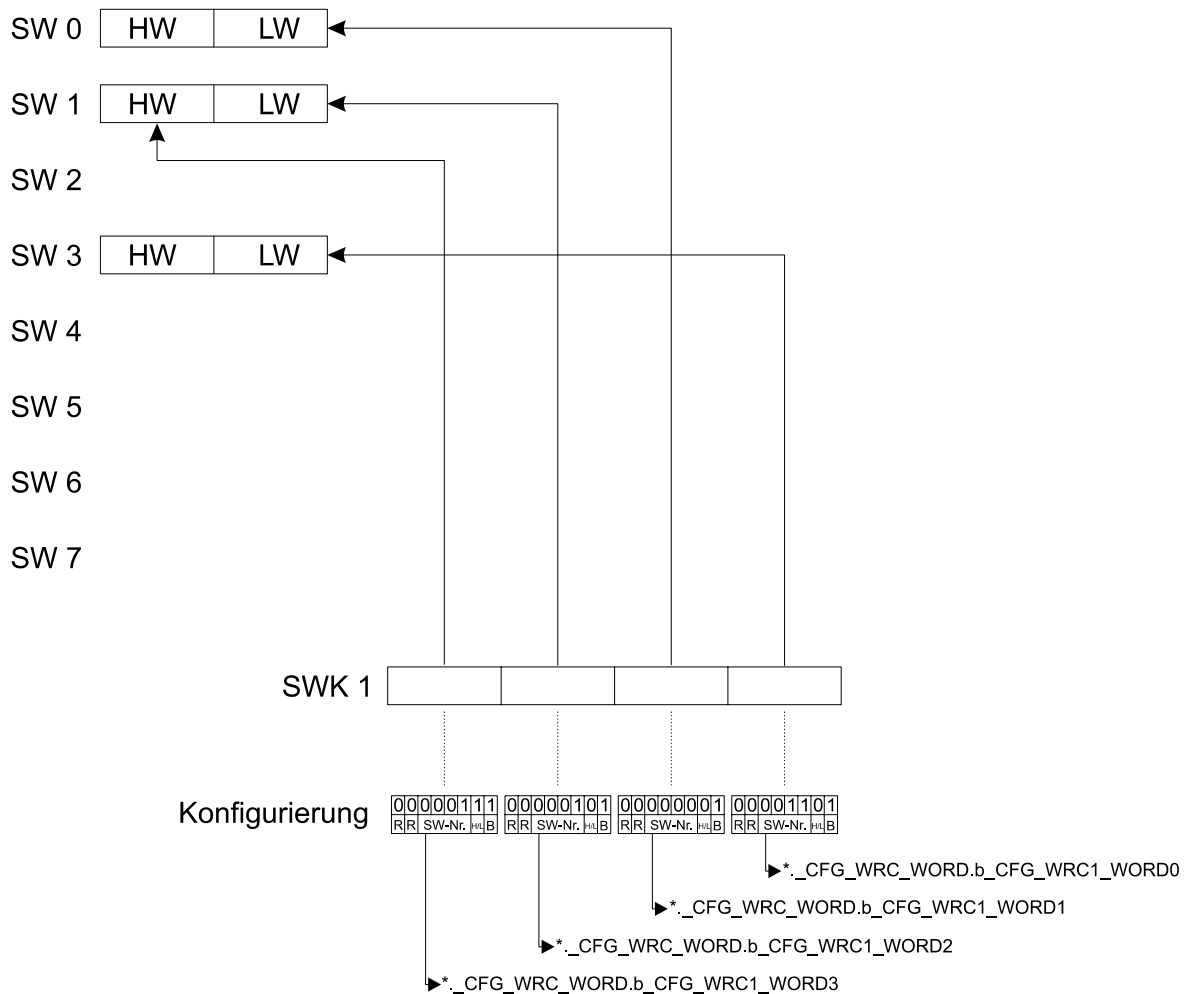
Bit 2	<b>Sollwertnummer</b> Nummer des Sollwertes 0 bis 15
Bit 3	
Bit 4	
Bit 5	
Bit 6	reserviert
Bit 7	



### HINWEIS

Ein Wort des Telegramms wird nur dann nicht verwendet, wenn Sollwertnummer = 0, Highword/Lowword = 0 und Belegt = 0 gesetzt wird.

### Beispiel:



\*\_CFG\_WRC\_WORD.b\_CFG\_WRC1\_WORD0 = 16#0D

0	0	0	0	1	1	0	1
R	R	SW-Nr.				H/L	B

(\* entspricht zum Beispiel \_CANsyncSlave\_Ctrl\_Slot\_G)

Mit dieser Einstellung wird das Wort 0 des Sollwert-Telegramms 1 als Lowword des Sollwertes 3 eingetragen.

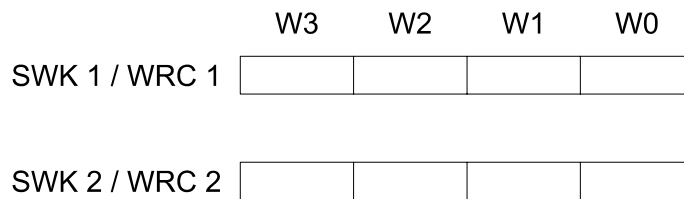
Register	Inhalt
*.a_WR_VALUE_RECEIVE[0]	Sollwert 0
*.a_WR_VALUE_RECEIVE[1]	Sollwert 1
*.a_WR_VALUE_RECEIVE[2]	Sollwert 2
*.a_WR_VALUE_RECEIVE[3]	Sollwert 3
*.a_WR_VALUE_RECEIVE[4]	Sollwert 4
*.a_WR_VALUE_RECEIVE[5]	Sollwert 5
*.a_WR_VALUE_RECEIVE[6]	Sollwert 6
*.a_WR_VALUE_RECEIVE[7]	Sollwert 7
*.a_WR_VALUE_RECEIVE[8]	reserviert
...	...
*.d_WR_VALUE_RECEIVE[15]	reserviert

(\* entspricht zum Beispiel \_CANsyncSlave\_Ctrl\_Slot\_G)

Die Sollwerte können Wort- oder Doppelwort-Sollwerte sein.

### Erläuterung zur Benutzung der Sollwertkanäle

Im Synchron-Betrieb stehen zwei Sollwertkanäle (Kanal 1 und 2; auch WRC1 und WRC2) zur Verfügung. In den Sollwertkanälen 1 und 2 werden die Sollwert-Telegramme 1 und 2 des CANsync-Masters empfangen.



Beide Sollwert-Telegramme bestehen aus jeweils vier Worten (W0 bis W3). Nach der CANsync-Initialisierung muss zumindest einmal angegeben werden, wie die Belegung dieser Worte ist.

Dies kann im Initialisierungsprogramm erfolgen. Dazu trägt man im Bereich \*\_CFG\_WRC\_WORD.b\_CFG\_WRC1\_WORD0 bis \*\_CFG\_WRC\_WORD.b\_CFG\_WRC2\_WORD3 für jedes Wort der Sollwert-Telegramme ein, welcher Sollwert an dieser Stelle empfangen wird. Sollwertnummern sind im Bereich von 0 bis 7 gültig. Für Doppelwort-Sollwerte müssen zwei Worte im Telegramm verwendet werden.

Die Konfigurierung kann auch während des aktiven Betriebs geändert werden. Die Änderung wird spätestens im nächsten CANsync-Intervall übernommen. Zu Beginn des CANsync-Intervalls werden die Konfigurierungsdaten von der CANsync-Anschaltung gelesen.

## 3.2 Detaillierte Informationen zum CANsync

Die Sollwert-Telegramm-Auswertung erfolgt zu Beginn des CANsync-Intervalls. Dazu fragt man die Statusregister (Kontrollregister) der Sollwertkanäle ab (\*b\_CTRLREG\_WRC1 bis \*b\_CTRLREG\_WRC2). In diesen Registern wird 16#02 eingetragen, wenn das entsprechende Sollwert-Telegramm empfangen wurde. Dann kann man die Sollwerte aus den Registern \*a\_WR\_VALUE\_RECEIVE[0] bis \*a\_WR\_VALUE\_RECEIVE[7] auslesen und das Statusregister auf z. B. 16#00 setzen, um die nächste Eintragung richtig erkennen zu können.

Die Zuordnung welche Sollwerte gelesen werden dürfen, wenn ein Sollwert-Telegramm empfangen wurde, muss entsprechend der Sollwertkonfigurierung im Applikationsprogramm erfolgen.

Falls ein Sollwert-Telegramm ausfällt, das z. B. einen Lagesollwert von der virtuellen Leitachse enthält, muss im Applikationsprogramm eine Extrapolation ausgehend vom letzten Lagesollwert erfolgen. Fällt der Sollwert mehrere Male hintereinander aus, muss eine Fehlerreaktion angestoßen werden.

### 3.2.3.5 Istwerte des CANsync-Slaves

Das Senden der Istwerte erfolgt in der CANsync Event-Task.

Register	Inhalt
*b_CTRLREG_RD_RDC1	Kontrollregister Istwertkanal 1
*b_CTRLREG_RD_RDC2	Kontrollregister Istwertkanal 2
*b_CTRLREG_RD_RDC3	reserviert
*b_CTRLREG_RD_RDC4	reserviert

(\* entspricht zum Beispiel \_CANsyncSlave\_Ctrl\_Slot\_G)

Im Kontrollregister wird mit 16#05 gekennzeichnet, dass neue Istwerte für das jeweilige Istwert-Telegramm eingetragen wurden und das Telegramm gesendet wird. Die CANsync-Anschaltung quittiert den Befehl mit 16#04.

Die Konfigurierung der Istwert-Telegramme erfolgt mit den folgenden Registern.

Register	Inhalt
*_CFG_RDC_WORD.b_CFG_RDC1_WORD0	Konfigurierung Istwert-Telegramm 1 Wort 0
*_CFG_RDC_WORD.b_CFG_RDC1_WORD1	Konfigurierung Istwert-Telegramm 1 Wort 1
*_CFG_RDC_WORD.b_CFG_RDC1_WORD2	Konfigurierung Istwert-Telegramm 1 Wort 2
*_CFG_RDC_WORD.b_CFG_RDC1_WORD3	Konfigurierung Istwert-Telegramm 1 Wort 3
*_CFG_RDC_WORD.b_CFG_RDC2_WORD0	Konfigurierung Istwert-Telegramm 2 Wort 0
*_CFG_RDC_WORD.b_CFG_RDC2_WORD1	Konfigurierung Istwert-Telegramm 2 Wort 1
*_CFG_RDC_WORD.b_CFG_RDC2_WORD2	Konfigurierung Istwert-Telegramm 2 Wort 2
*_CFG_RDC_WORD.b_CFG_RDC2_WORD3	Konfigurierung Istwert-Telegramm 2 Wort 3
*_CFG_RDC_WORD.b_CFG_RDC3_WORD0	reserviert
*_CFG_RDC_WORD.b_CFG_RDC3_WORD1	reserviert
*_CFG_RDC_WORD.b_CFG_RDC3_WORD2	reserviert
*_CFG_RDC_WORD.b_CFG_RDC3_WORD3	reserviert
*_CFG_RDC_WORD.b_CFG_RDC4_WORD0	reserviert
*_CFG_RDC_WORD.b_CFG_RDC4_WORD1	reserviert

Register	Inhalt
*._CFG_RDC_WORD.b_CFG_RDC4_WORD2	reserviert
*._CFG_RDC_WORD.b_CFG_RDC4_WORD3	reserviert

(\* entspricht zum Beispiel \_CANsyncSlave\_Ctrl\_Slot\_G)

Mit der Konfigurierung gibt man an, welcher Istwert an welcher Stelle im Istwert-Telegramm eingetragen wird.

Bedeutung:

Bit 0	<b>Belegt</b> Wenn = 1, wird dieses Wort des Telegramms verwendet
Bit 1	<b>Highword/Lowword</b> Wenn = 1, wird das Highword des Istwertes eingetragen
Bit 2	<b>Istwertnummer</b> Nummer des Istwertes 0 bis 15
Bit 3	
Bit 4	
Bit 5	
Bit 6	reserviert
Bit 7	

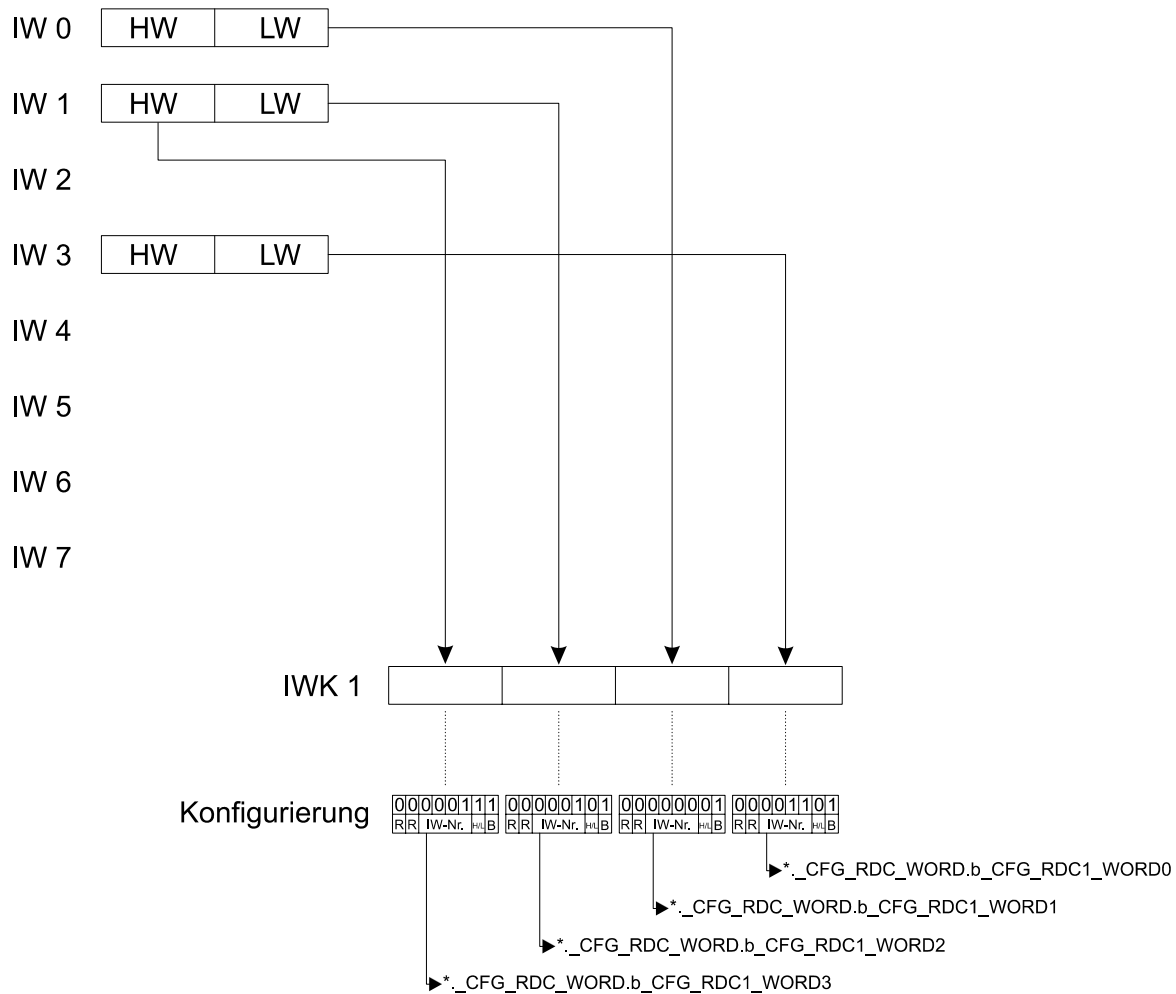
#### HINWEIS



Ein Wort des Telegramms wird nur dann nicht verwendet, wenn Istwertnummer = 0, Highword/Lowword = 0 und Belegt = 0 gesetzt wird.

## 3.2 Detaillierte Informationen zum CANsync

Beispiel:



\*\_CFG\_RDC\_WORD.b\_CFG\_RDC1\_WORD0 = 16#0D

0	0	0	0	1	1	0	1
R	R	IW-Nr.		H/L		B	

(\* entspricht zum Beispiel \_CANsyncSlave\_Ctrl\_Slot\_G)

Mit dieser Einstellung wird das Lowword des Istwertes 3 in das Istwert-Telegramm 1 im Wort 0 eingetragen.

Register	Inhalt
*.a_RD_VALUE_SEND[0]	Istwert 0
*.a_RD_VALUE_SEND[1]	Istwert 1
*.a_RD_VALUE_SEND[2]	Istwert 2
*.a_RD_VALUE_SEND[3]	Istwert 3
*.a_RD_VALUE_SEND[4]	Istwert 4
*.a_RD_VALUE_SEND[5]	Istwert 5
*.a_RD_VALUE_SEND[6]	Istwert 6
*.a_RD_VALUE_SEND[7]	Istwert 7



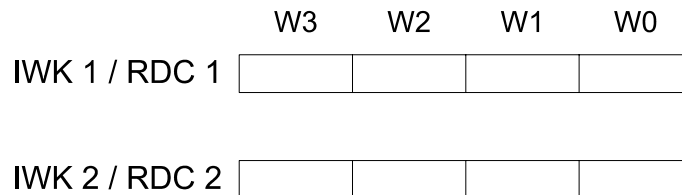
Register	Inhalt
*.a_RD_VALUE_SEND[8]	reserviert
...	...
*.a_RD_VALUE_SEND[15]	reserviert

(\* entspricht zum Beispiel \_CANsyncSlave\_Ctrl\_Slot\_G)

Die Istwerte können Wort- oder Doppelwort-Istwerte sein.

### Erläuterung zur Benutzung der Istwertkanäle

Im Synchron-Betrieb stehen zwei Istwertkanäle (Kanal 1 und 2; auch RDC1 und RDC2) zur Verfügung. In den Istwertkanälen 1 und 2 werden die Istwert-Telegramme 1 und 2 dieses CANsync-Slaves an den CANsync-Master gesendet.



Beide Istwert-Telegramme bestehen aus jeweils vier Worten (W0 bis W3). Nach der CANsync-Initialisierung muss zumindest einmal angegeben werden, wie die Belegung dieser Worte ist.

Dies kann im Initialisierungsprogramm erfolgen. Dazu trägt man im Bereich  
 \*.CFG\_RDC\_WORD.b\_CFG\_RDC1\_WORD0 bis  
 \*.CFG\_RDC\_WORD.b\_CFG\_RDC2\_WORD3 für jedes Wort der Istwert-Telegramme ein, welcher Istwert an dieser Stelle übertragen wird. Istwertnummern sind im Bereich von 0 bis 7 gültig. Für Doppelwort-Istwerte müssen zwei Worte im Telegramm verwendet werden.

Die Konfigurierung kann auch während des aktiven Betriebs geändert werden. Die Änderung wird spätestens im nächsten CANsync-Intervall übernommen. Die Konfigurationsdaten von der CANsync-Anschaltung werden gelesen, wenn das entsprechende Istwert-Telegramm empfangen wird.

Der CANsync-Slave kann nicht von sich aus das Senden seiner Istwerte auslösen, sondern der CANsync-Master fordert über den Identifier der Sollwert-Telegramme die Istwerte eines CANsync-Slaves an.

Damit immer die aktuellen Istwerte gesendet werden, muss die Istwerteintragung zu Beginn der CANsync Event-Task erfolgen. Von der fallenden Flanke des SYNC-Signals bis zur Ausführung des Applikationsprogramms in der CANsync Event-Task vergehen ca. 70 µs. Die Istwerte für das Istwert-Telegramm 1 müssen bis ca. 200 µs nach dem Start der CANsync Event-Task in die Register \*.a\_RD\_VALUE\_SEND[0] bis \*.a\_RD\_VALUE\_SEND[7] eingetragen werden, weil dann die CANsync-Anschaltung die Vorbereitung der Istwert-Telegramme beginnt. Als Kennzeichnung, dass neue Istwerte eingetragen wurden, muss im entsprechenden Kontrollregister des Istwertkanals \*.b\_CTRLREG\_RD\_RDC1 bis \*.b\_CTRLREG\_RD\_RDC2 16#05 eingetragen werden. Dies ist die Freigabe für die CANsync-Anschaltung, dass die Istwerte gelesen werden dürfen und das Istwert-Telegramm erzeugt wird. Falls bis 200 µs nach dem Start der CANsync Event-Task diese Freigabe nicht erfolgt, fällt in diesem CANsync-Intervall das Istwert-Telegramm 1 aus. Als Quittung für die Erzeugung des Istwert-Telegramms wird

## 3.2 Detaillierte Informationen zum CANsync

im Kontrollregister 16#04 von der CANsync-Anschaltung eingetragen. Dadurch kann der Anwender überprüfen, ob die Istwerteintragung rechtzeitig fertig geworden ist oder nicht. Falls die Istwertgenerierung länger dauert, muss die Istwertgenerierung immer für das nächste CANsync-Intervall erfolgen und zu Beginn des neuen CANsync-Intervalls müssen die vorberechneten Istwerte nur noch an die entsprechenden Register kopiert werden.

Der Zeitpunkt bis zu dem die Istwerte für das Istwert-Telegramm 2 eingetragen werden müssen, ist 450 µs nach Start der CANsync Event-Task. Die Signalisierung im Kontrollregister erfolgt wie für das Istwert-Telegramm 1.

### 3.2.3.6 Istwerte anderer CANsync-Slaves

Das Empfangen der Istwerte der anderen CANsync-Slaves erfolgt in der CANsync Event-Task.

Register	Inhalt
*.b_STATREG_RDC1	Statusregister Istwertkanal 1
*.b_STATREG_RDC2	Statusregister Istwertkanal 2
*.b_STATREG_RDC3	reserviert
*.b_STATREG_RDC4	reserviert

(\* entspricht zum Beispiel \_CANsyncSlave\_Ctrl\_Slot\_G)

Im Statusregister wird die Slavenummer der CANsync-Slaves eingetragen, von dem ein Istwert-Telegramm empfangen wurde.

Register	Inhalt
*.a_STATREG_RDC[0].b_STATREG_RDC1	Istwertquittung Istwertkanal 1 Slave 0
*.a_STATREG_RDC[0].b_STATREG_RDC2	Istwertquittung Istwertkanal 2 Slave 0
*.a_STATREG_RDC[0].b_STATREG_RDC3	reserviert
*.a_STATREG_RDC[0].b_STATREG_RDC4	reserviert
*.a_STATREG_RDC[1].b_STATREG_RDC1	Istwertquittung Istwertkanal 1 Slave 1
*.a_STATREG_RDC[1].b_STATREG_RDC2	Istwertquittung Istwertkanal 2 Slave 1
*.a_STATREG_RDC[1].b_STATREG_RDC3	reserviert
*.a_STATREG_RDC[1].b_STATREG_RDC4	reserviert
...	...
*.a_STATREG_RDC[31].b_STATREG_RDC1	Istwertquittung Istwertkanal 1 Slave 31
*.a_STATREG_RDC[31].b_STATREG_RDC2	Istwertquittung Istwertkanal 2 Slave 31
*.a_STATREG_RDC[31].b_STATREG_RDC3	reserviert
*.a_STATREG_RDC[31].b_STATREG_RDC4	reserviert

(\* entspricht zum Beispiel \_CANsyncSlave\_Ctrl\_Slot\_G)

Wenn das entsprechende Istwert-Telegramm für einen CANsync-Slave angekommen ist, wird 16#02 in das Istwertquittungsregister eingetragen.

Die Konfigurierung der Istwert-Telegramme der anderen CANsync-Slaves erfolgt mit den folgenden Registern.

Register	Inhalt
*.a_CFG_RDC_WORD[0].b_CFG_RDC1_WORD0	Konfigurierung Istwert-Telegramm 1 Wort 0 Slave 0
*.a_CFG_RDC_WORD[0].b_CFG_RDC1_WORD1	Konfigurierung Istwert-Telegramm 1 Wort 1 Slave 0
*.a_CFG_RDC_WORD[0].b_CFG_RDC1_WORD2	Konfigurierung Istwert-Telegramm 1 Wort 2 Slave 0
*.a_CFG_RDC_WORD[0].b_CFG_RDC1_WORD3	Konfigurierung Istwert-Telegramm 1 Wort 3 Slave 0
*.a_CFG_RDC_WORD[0].b_CFG_RDC2_WORD0	Konfigurierung Istwert-Telegramm 2 Wort 0 Slave 0
*.a_CFG_RDC_WORD[0].b_CFG_RDC2_WORD1	Konfigurierung Istwert-Telegramm 2 Wort 1 Slave 0
*.a_CFG_RDC_WORD[0].b_CFG_RDC2_WORD2	Konfigurierung Istwert-Telegramm 2 Wort 2 Slave 0
*.a_CFG_RDC_WORD[0].b_CFG_RDC2_WORD3	Konfigurierung Istwert-Telegramm 2 Wort 3 Slave 0
*.a_CFG_RDC_WORD[0].b_CFG_RDC3_WORD0	reserviert
*.a_CFG_RDC_WORD[0].b_CFG_RDC3_WORD1	reserviert
*.a_CFG_RDC_WORD[0].b_CFG_RDC3_WORD2	reserviert
*.a_CFG_RDC_WORD[0].b_CFG_RDC3_WORD3	reserviert
*.a_CFG_RDC_WORD[0].b_CFG_RDC4_WORD0	reserviert
*.a_CFG_RDC_WORD[0].b_CFG_RDC4_WORD1	reserviert
*.a_CFG_RDC_WORD[0].b_CFG_RDC4_WORD2	reserviert
*.a_CFG_RDC_WORD[0].b_CFG_RDC4_WORD3	reserviert
*.a_CFG_RDC_WORD[1].b_CFG_RDC1_WORD0	Konfigurierung Istwert-Telegramm 1 Wort 0 Slave 1
*.a_CFG_RDC_WORD[1].b_CFG_RDC1_WORD1	Konfigurierung Istwert-Telegramm 1 Wort 1 Slave 1
*.a_CFG_RDC_WORD[1].b_CFG_RDC1_WORD2	Konfigurierung Istwert-Telegramm 1 Wort 2 Slave 1
*.a_CFG_RDC_WORD[1].b_CFG_RDC1_WORD3	Konfigurierung Istwert-Telegramm 1 Wort 3 Slave 1
*.a_CFG_RDC_WORD[1].b_CFG_RDC2_WORD0	Konfigurierung Istwert-Telegramm 2 Wort 0 Slave 1
*.a_CFG_RDC_WORD[1].b_CFG_RDC2_WORD1	Konfigurierung Istwert-Telegramm 2 Wort 1 Slave 1
*.a_CFG_RDC_WORD[1].b_CFG_RDC2_WORD2	Konfigurierung Istwert-Telegramm 2 Wort 2 Slave 1
*.a_CFG_RDC_WORD[1].b_CFG_RDC2_WORD3	Konfigurierung Istwert-Telegramm 2 Wort 3 Slave 1
...	...
*.a_CFG_RDC_WORD[31].b_CFG_RDC1_WORD0	Konfigurierung Istwert-Telegramm 1 Wort 0 Slave 31
*.a_CFG_RDC_WORD[31].b_CFG_RDC1_WORD1	Konfigurierung Istwert-Telegramm 1 Wort 1 Slave 31
*.a_CFG_RDC_WORD[31].b_CFG_RDC1_WORD2	Konfigurierung Istwert-Telegramm 1 Wort 2 Slave 31
*.a_CFG_RDC_WORD[31].b_CFG_RDC1_WORD3	Konfigurierung Istwert-Telegramm 1 Wort 3 Slave 31
*.a_CFG_RDC_WORD[31].b_CFG_RDC2_WORD0	Konfigurierung Istwert-Telegramm 2 Wort 0 Slave 31
*.a_CFG_RDC_WORD[31].b_CFG_RDC2_WORD1	Konfigurierung Istwert-Telegramm 2 Wort 1 Slave 31
*.a_CFG_RDC_WORD[31].b_CFG_RDC2_WORD2	Konfigurierung Istwert-Telegramm 2 Wort 2 Slave 31
*.a_CFG_RDC_WORD[31].b_CFG_RDC2_WORD3	Konfigurierung Istwert-Telegramm 2 Wort 3 Slave 31
*.a_CFG_RDC_WORD[31].b_CFG_RDC3_WORD0	reserviert
*.a_CFG_RDC_WORD[31].b_CFG_RDC3_WORD1	reserviert
*.a_CFG_RDC_WORD[31].b_CFG_RDC3_WORD2	reserviert
*.a_CFG_RDC_WORD[31].b_CFG_RDC3_WORD3	reserviert
*.a_CFG_RDC_WORD[31].b_CFG_RDC4_WORD0	reserviert
*.a_CFG_RDC_WORD[31].b_CFG_RDC4_WORD1	reserviert
*.a_CFG_RDC_WORD[31].b_CFG_RDC4_WORD2	reserviert
*.a_CFG_RDC_WORD[31].b_CFG_RDC4_WORD3	reserviert

(\* entspricht zum Beispiel \_CANsyncSlave\_Ctrl\_Slot\_G)

Mit der Konfigurierung gibt man an, welcher Istwert an welcher Stelle im Istwert-Telegramm (des anderen CANsync-Slaves) übertragen wird.

## 3.2 Detaillierte Informationen zum CANsync

---

Bedeutung:

Bit 0	<b>Belegt</b> Wenn = 1, wird dieses Wort des Telegramms verwendet
Bit 1	<b>Highword/Lowword</b> Wenn = 1, wird das Highword des Istwertes eingetragen
Bit 2	<b>Istwertnummer</b> Nummer des Istwertes 0 bis 7
Bit 3	
Bit 4	
Bit 5	
Bit 6	reserviert
Bit 7	

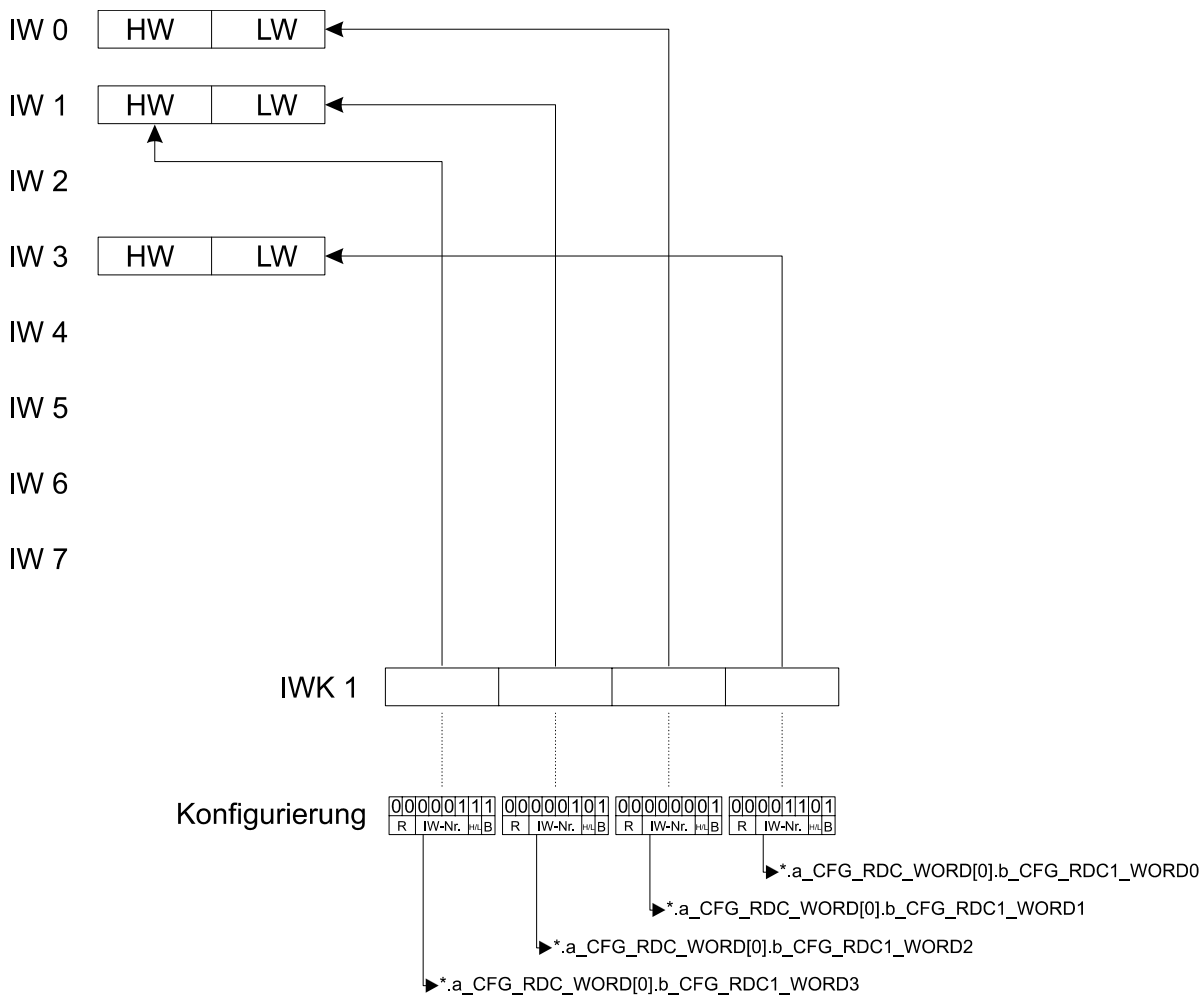


### HINWEIS

Ein Wort des Telegramms wird nur dann nicht verwendet, wenn Istwertnummer = 0, Highword/Lowword = 0 und Belegt = 0 gesetzt wird.

---

**Beispiel:**



\*\_a\_CFG\_RDC\_WORD[0].b\_CFG\_RDC1\_WORD0 = 16#0D 

0	0	0	0	1	1	0	1
R		IW-Nr.			H/L		B

(\* entspricht zum Beispiel \_CANsyncSlave\_Ctrl\_Slot\_G)

Damit wird aus dem Istwert-Telegramm 1 des CANsync-Slaves 0 das Wort 0 gelesen und in das Lowword des Istwertes 3 geschrieben.

Register	Inhalt
*_a_RD_BMARRAY[0][0]	Istwert 0 Slave 0
*_a_RD_BMARRAY[0][1]	Istwert 1 Slave 0
*_a_RD_BMARRAY[0][2]	Istwert 2 Slave 0
*_a_RD_BMARRAY[0][3]	Istwert 3 Slave 0
*_a_RD_BMARRAY[0][4]	Istwert 4 Slave 0
*_a_RD_BMARRAY[0][5]	Istwert 5 Slave 0
*_a_RD_BMARRAY[0][6]	Istwert 6 Slave 0
*_a_RD_BMARRAY[0][7]	Istwert 7 Slave 0
*_a_RD_BMARRAY[0][8]	reserviert

## 3.2 Detaillierte Informationen zum CANsync

Register	Inhalt
...	...
*.a_RD_BMARRAY[0][15]	reserviert
*.a_RD_BMARRAY[1][0]	Istwert 0 Slave 1
*.a_RD_BMARRAY[1][1]	Istwert 1 Slave 1
*.a_RD_BMARRAY[1][2]	Istwert 2 Slave 1
*.a_RD_BMARRAY[1][3]	Istwert 3 Slave 1
*.a_RD_BMARRAY[1][4]	Istwert 4 Slave 1
*.a_RD_BMARRAY[1][5]	Istwert 5 Slave 1
*.a_RD_BMARRAY[1][6]	Istwert 6 Slave 1
*.a_RD_BMARRAY[1][7]	Istwert 7 Slave 1
*.a_RD_BMARRAY[1][8]	reserviert
...	...
*.a_RD_BMARRAY[31][0]	Istwert 0 Slave 31
*.a_RD_BMARRAY[31][1]	Istwert 1 Slave 31
*.a_RD_BMARRAY[31][2]	Istwert 2 Slave 31
*.a_RD_BMARRAY[31][3]	Istwert 3 Slave 31
*.a_RD_BMARRAY[31][4]	Istwert 4 Slave 31
*.a_RD_BMARRAY[31][5]	Istwert 5 Slave 31
*.a_RD_BMARRAY[31][6]	Istwert 6 Slave 31
*.a_RD_BMARRAY[31][7]	Istwert 7 Slave 31
*.a_RD_BMARRAY[31][8]	reserviert
...	...
*.a_RD_BMARRAY[31][15]	reserviert

(\* entspricht zum Beispiel \_CANsyncSlave\_Ctrl\_Slot\_G)

Die Istwerte können Wort- oder Doppelwort-Istwerte sein.



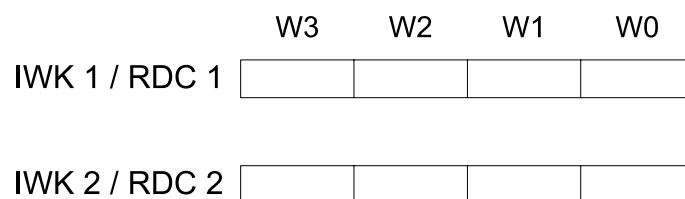
### HINWEIS

Bei der Anzeige von \_CANsyncSlave\_Ctrl\_Slot\_G im PROPROG wt II Watch-Fenster kann zwischen den eckigen Klammern ein Punkt stehen.

### Erläuterung zur Benutzung der Istwertkanäle

Ein CANsync-Slave kann auch alle Istwert-Telegramme der anderen CANsync-Slaves auswerten. Die Anforderung ist jedoch nur vom CANsync-Master aus möglich.

Im Synchron-Betrieb stehen zwei Istwertkanäle (Kanal 1 und 2; auch RDC1 und RDC2) zur Verfügung. In den Istwertkanälen 1 und 2 werden die Istwert-Telegramme 1 und 2 gesendet.



Beide Istwert-Telegramme bestehen aus jeweils vier Worten (W0 bis W3). Nach der CANsync-Initialisierung muss zumindest einmal angegeben werden, wie die Belegung dieser Worte ist. Dies kann im Initialisierungsprogramm erfolgen.

Dazu trägt man im Bereich `*.a_CFG_RDC_WORD[0].b_CFG_RDC1_WORD0` bis `*.a_CFG_RDC_WORD[31].b_CFG_RDC2_WORD3` für jedes belegte Wort der Istwert-Telegramme (jedes CANsync-Slaves) ein, welcher Istwert der anderen CANsync-Slaves an dieser Stelle übertragen wird. Istwertnummern sind im Bereich von 0 bis 7 gültig. Für Doppelwort-Istwerte müssen zwei Worte im Telegramm verwendet werden.

Die Konfiguration kann auch während des aktiven Betriebs geändert werden. Die Änderung wird spätestens im nächsten CANsync-Intervall übernommen. Die Konfigurationsdaten werden von der CANsync-Anschaltung gelesen, wenn das entsprechende Istwert-Telegramm empfangen wird.

Die Istwert-Telegramm-Auswertung erfolgt zu Beginn der CANsync Event-Task. Dazu gibt es zwei Auswertungsmethoden: Bei der ersten Methode fragt man die Statusregister der Istwertkanäle ab (`*.b_STATREG_RDC1` oder `*.b_STATREG_RDC2`). In diesen Registern wird die Slavenummer des CANsync-Slaves eingetragen, von dem das entsprechende Istwert-Telegramm empfangen wurde. Dann kann man die Istwerte aus den Registern (`*.a_RD_BMARRAY[0][0]` bis `*.a_RD_BMARRAY[31][7]`) dieses CANsync-Slaves auslesen und das Statusregister auf z. B. `16#FF` setzen, um die nächste Eintragung richtig erkennen zu können.

Bei der zweiten Methode fragt man direkt die Istwertquittung (`*.a_STATREG_RDC[0].b_STATREG_RDC1` bis `*.a_STATREG_RDC[31].b_STATREG_RDC2`) für einen CANsync-Slave (dessen Istwert-Telegramme mitgelesen werden) ab. In diesem Quittungsregister wird der Empfang eines Istwert-Telegramms mit `16#02` gekennzeichnet. Dann kann man die Istwerte aus den Registern (`*.a_RD_BMARRAY[0][0]` bis `*.a_RD_BMARRAY[31][7]`) dieses CANsync-Slaves auslesen. Auch hier muss man das Statusregister anschließend mit einem anderen Wert beschreiben, um das Wiedereintragen der Quittung zu erkennen.

Die Zuordnung welche Istwerte gelesen werden dürfen, wenn ein Istwert-Telegramm empfangen wurde, muss entsprechend der Istwertkonfiguration im Applikationsprogramm erfolgen.

Die Anforderung an einen CANsync-Slave, sein Istwert-Telegramm zu senden, erfolgt über das Sollwert-Telegramm des CANsync-Masters. Die Einstellung kann nur am CANsync-Master erfolgen.

### 3.2.3.7 Kommando- und Antwortkanal

#### Aktionskommando

Aktionskommandos werden in den folgenden Registern gemeldet.

Register	Inhalt
<code>*.b_STATREG_AKT_CMD</code>	Statusregister Aktionskommando
<code>*.b_AKT_CMD</code>	Aktionskommando
<code>*.b_DATA_BYTE_AKT_CMD</code>	Datenbyte 0 (DB)
<code>*.w_DATAWORD_AKT_CMD</code>	Datenwort 1 (DW)
<code>*.b_STATREG_CONTROLWORD</code>	Statusregister Steuerwort
<code>*.w_CONTROLWORD</code>	Steuerwort

## 3.2 Detaillierte Informationen zum CANsync

Register	Inhalt
*.b_STATREG_SYNC_MODUS	Statusregister SYNC-Modus
*.b_SYNC_MODUS	SYNC-Modus

(\* entspricht zum Beispiel \_CANsyncSlave\_Ctrl\_Slot\_G)

Das Aktionskommando, das als Broadcastkommando vom CANsync-Master gesendet wird, wird nur gemeldet, wenn der CANsync-Slave angesprochen wird, d. h. das entsprechende Bit in der Bitleiste gesetzt ist (Siehe [▶Registerstruktur und Funktion des Optionsmodul CANsync-Master für b maXX PLC◀](#) ab Seite 43).

Wenn das Aktionskommando ein Steuerwortkommando (Kommandobyte = 16#01) ist, wird das Datenwort des Kommandos im \*.w\_CONTROLWORD als Steuerwort und im \*.b\_STATREG\_CONTROLWORD als Empfangsanzeige 16#02 eingetragen.

Bei alle anderen Aktionskommandos werden die Daten in \*.b\_DATA\_BYTE\_AKT\_CMD und \*.w\_DATAWORD\_AKT\_CMD eingetragen. Die Quittung (Empfangsanzeige 16#02) steht dann im \*.b\_STATREG\_AKT\_CMD. Da in jedem CANsync-Intervall ein neues Aktionskommando übertragen werden kann, sollte das Statusregister des Aktionskommandos in jeder CANsync Event-Task ausgewertet werden.

### Parameterkommando

Parameterkommandos werden in folgenden Registern gemeldet.

Register	Inhalt
*.b_STEUREG_PAR_CMD	Steuerregister Parameterkommando
*.b_STATREG_PAR_CMD	Statusregister Parameterkommando
*.w_ERR_NR_PAR_CMD	Fehlernummer Parameterkommando
*.d_DATA_PAR_CMD	Daten Parameterkommando
*.w_PAR_NR	Parameter Nummer

(\* entspricht zum Beispiel \_CANsyncSlave\_Ctrl\_Slot\_G)

### Bedeutung der Steuer- und Statusregister

Bit 0	<b>Aktiv</b> wird auf 1 gesetzt, wenn Auftrag empfangen wurde muss auf 0 gesetzt werden, wenn Auftrag bearbeitet wurde
Bit 1	<b>Read</b> = 1: Parameter schreiben
Bit 2	<b>Write</b> = 1: Parameter lesen
Bit 3	<b>Change</b> = 1: Anzeige, dass Parameterkommando vom CANsync-Master geändert wurde, muss mit Reset quittiert werden
Bit 4	<b>Format</b> 0: Wortparameter 1: Doppelwort-Parameter



Bit 5	<b>Fehleranzeige</b> muss auf 1 gesetzt werden, wenn der Auftrag einen Fehler erzeugt
Bit 6	<b>Busy</b> = 1: Antwort ist noch nicht gesendet = 0: Antwort wurde an CANsync-Master gesendet
Bit 7	<b>Reset</b> = 1 bestätigt, dass die Kommandoänderung am Bit Change gesehen wurde und das alte Kommando nicht weiter bearbeitet wird.

### Ablauf eines Parameterzugriffs

Die Bearbeitung von Parameterkommandos kann sowohl in der CANsync Event-Task als auch im restlichen Programm erfolgen. Zur Sicherung der Datenkonsistenz erfolgt die Auftragserteilung und die Auswertung der Rückantwort über das Steuerregister und das Statusregister. Um eine konfliktfreie Durchführung des Auftrags zu gewährleisten, muss vom Applikationsprogramm aus immer zuerst das Steuerregister und dann das Statusregister gelesen werden und wenn ein neuer Wert geschrieben wird, immer zuerst das Statusregister und dann das Steuerregister gesetzt werden. (Für die CANsync-Anschaltung ist das Steuerregister das entscheidende Register.)

Wenn ein Parameter-Schreibauftrag empfangen wurde, steht in `*.d_DATA_PAR_CMD` der zu schreibende Wert und in `*.w_PAR_NR` die Parameternummer. Im Steuerregister (`*.b_STEUREG_PAR_CMD`) und im Statusregister (`*.b_STATREG_PAR_CMD`) wird mit `16#45` ein Wortschreibkommando und mit `16#55` ein Doppelwort-Schreibkommando gemeldet.

Wenn vom Applikationsprogramm der geforderte Parameter fehlerfrei beschrieben wurde, muss dies im Statusregister und im Steuerregister durch das Löschen des Bit 0 (Aktiv) angezeigt werden.

Sobald das Antwort-Telegramm zum CANsync-Master gesendet wurde, wird das Bit 6 (Busy) von der CANsync-Anschaltung gelöscht.

Wenn das Applikationsprogramm den Schreibzugriff nicht ausführen kann, muss eine Fehlernummer in `*.w_ERR_NR_PAR_CMD` eingetragen werden und in den Registern das Bit 5 (Fehleranzeige) gesetzt werden und das Bit 0 (Aktiv) gelöscht werden. Dann wird an den CANsync-Master ein Antwort-Telegramm mit der eingetragenen Fehlernummer gesendet.

Sobald das Antwort-Telegramm zum CANsync-Master gesendet wurde, wird das Bit 6 (Busy) von der CANsync-Anschaltung gelöscht.

Der Parameter-Leseauftrag läuft dazu ähnlich ab, nur mit dem Unterschied, dass der Befehl unabhängig vom Format `16#43` lautet und dass zur Antwort die Daten in `*.d_DATA_PAR_CMD` eingetragen werden müssen. Vor der Quittierung durch das Löschen des Bit 0 (Aktiv) muss das tatsächliche Format des Parameters im Bit 4 (Format) eingetragen werden, damit die Rückantwort an den CANsync-Master richtig erzeugt werden kann.

Wenn der CANsync-Master das Parameterkommando ändert (Parameternummer) wird dies durch das gesetzte Bit 3 (Change) angezeigt. Dann muss der alte Befehl nicht mehr beantwortet werden. Das Applikationsprogramm muss die Änderung erkennen und dies durch das Setzen des Bit 7 (Reset) quittieren. Dann wird das aktuelle Parameterkommando eingetragen.

### Fehlercode

Wert	Bedeutung
16#0000	Kein Fehler aufgetreten
16#FFFF	Fehler aufgetreten
16#FFFE	Wert kleiner Minimalwert
16#FFFD	Wert größer Maximalwert
16#FFFC	Element nicht veränderbar
16#FFFB	Element nicht vorhanden
16#FFFA	Daten nicht verfügbar (z. B. in Bearbeitung)
16#FFF9	Fehler beim Datenformat

### Up/Download-Kommando

Upload- und Download-Kommandos werden in den folgenden Registern angezeigt.

Register	Inhalt
*.b_STEUREG_UPDOWNBLK1	Steuerregister Up/Download Block 1
*.b_STATREG_UPDOWNBLK1	Statusregister Up/Download Block 1
*.b_EN_UPDOWNBLK1	Freigabe Block 1
*.w_ERR_NR_UPDOWNBLK1	Fehlernummer Up/Download Block 1
*.d_BASE_ADR_UPDOWNBLK1	Basisadresse Up/Download Block 1
...	...
*.w_LENGTH_UPDOWNBLK1	Länge in Bytes Up/Download Block 1
*.w_COUNTER_UPDOWNBLK1	Zähler Up/Download Block 1
*.a_DATA_UPDOWNBLK1[0 bis 74]	Datenblock Up/Download Block 1

(\* entspricht zum Beispiel \_CANsyncSlave\_Ctrl\_Slot\_G)

### Bedeutung der Steuer- und Statusregister

Bit 0	<b>Aktiv</b> wird auf 1 gesetzt, wenn Auftrag empfangen wurde muss auf 0 gesetzt werden, wenn Auftrag bearbeitet wurde															
Bit 1	<b>Change</b> = 1: Anzeige, dass Parameterkommando vom CANsync-Master geändert wurde, muss mit Reset quittiert werden															
Bit 2	<b>Mode</b> <table border="0"> <tr> <td>Bit3</td> <td>Bit2</td> <td></td> </tr> <tr> <td>0</td> <td>0:</td> <td>reserviert</td> </tr> <tr> <td>0</td> <td>1:</td> <td>Initialisierung</td> </tr> <tr> <td>1</td> <td>0:</td> <td>laufender Up/Download</td> </tr> <tr> <td>1</td> <td>1:</td> <td>Blockende</td> </tr> </table>	Bit3	Bit2		0	0:	reserviert	0	1:	Initialisierung	1	0:	laufender Up/Download	1	1:	Blockende
Bit3		Bit2														
0	0:	reserviert														
0	1:	Initialisierung														
1	0:	laufender Up/Download														
1	1:	Blockende														
Bit 3																
Bit 4	<b>Up/Download</b> = 0: Upload = 1: Download															

Bit 5	<b>Fehleranzeige</b> muss auf 1 gesetzt werden, wenn der Auftrag einen Fehler erzeugt
Bit 6	<b>Busy</b> = 1: Antwort ist noch nicht gesendet = 0: Antwort wurde an CANsync-Master gesendet
Bit 7	<b>Reset</b> = 1 bestätigt, dass die Kommandoänderung am Bit Change gesehen wurde und das alte Kommando nicht weiter bearbeitet wird.

### Ablauf eines Up/Download-Auftrags im Block 1

Die Auswertung von Up/Downloadkommandos können sowohl in der CANsync Event-Task als auch im restlichen Programm verwendet werden. Zur Sicherung der Datenkonsistenz erfolgt die Auftragserteilung und die Auswertung der Rückantwort über das Steuerregister und das Statusregister. Um eine konfliktfreie Durchführung des Auftrags zu gewährleisten, muss vom Applikationsprogramm aus immer zuerst das Steuerregister und dann das Statusregister gelesen werden und wenn ein neuer Wert geschrieben wird, immer zuerst das Statusregister und dann das Steuerregister gesetzt werden. (Für die CANsync-Anschaltung ist das Steuerregister das entscheidende Register.)

Wenn ein Download gemeldet wird, wird die Basisadresse im `*.d_BASE_ADR_UPDOWNBLK1` und die Blocklänge im `*.w_LENGTH_UPDOWNBLK1` eingetragen.

Die Basisadressen `16#00000000` bis `16#000000FF` sind für Aufträge des Betriebssystems reserviert.

Im Statusregister `*.b_STATREG_UPDOWNBLK1` und im Steuerregister `*.b_STEUREG_UPDOWNBLK1` wird der Wert `16#55` gemeldet. Wenn der Download erlaubt ist, muss das Applikationsprogramm das Bit 0 (Aktiv) löschen. Dann wird der gesamte Block vom CANsync-Master empfangen (der Fortschritt kann im Byte-Zähler in `*.w_COUNTER_UPDOWNBLK1` abgelesen werden) und am Ende in den Registern `16#5D` gemeldet. Wenn die Daten aus dem Bereich `*.a_DATA_UPDOWNBLK1[0]` bis `*.a_DATA_UPDOWNBLK1[74]` übernommen wurden, muss man das Bit 0 (Aktiv) löschen. Als Anzeige, dass das Antwort-Telegramm zum CANsync-Master gesendet wurde, wird das Bit 6 (Busy) gelöscht.

Falls im CANsync-Slave bei der Ausführung des Auftrags ein Fehler auftritt, muss die Fehlernummer in `*.w_ERR_NR_UPDOWNBLK1` eingetragen werden, das Bit 5 (Fehleranzeige) gesetzt und das Bit 0 (Aktiv) gelöscht. Dann wird an den CANsync-Master ein entsprechendes Fehlertelegramm gesendet und damit der Download abgebrochen.

Der Upload-Auftrag läuft dazu ähnlich ab, nur mit dem Unterschied, dass der Befehl `16#05` lautet und dass die Daten vor der Quittierung des Initialisierungs-Telegramms in den Blockbereich eingetragen werden müssen.

Wenn der CANsync-Master das Up/Downloadkommando ändert wird dies durch das gesetzte Bit 1 (Change) angezeigt. Dann muss der alte Befehl abgebrochen werden. Das Applikationsprogramm muss die Änderung erkennen und dies durch das Setzen des Bit 7 (Reset) quittieren. Dann wird das aktuelle Up/Downloadkommando eingetragen.

Die Fehlernummern, die vom CANsync-Slave eingetragen werden können, sind größer `16#00FF`.

## 3.2 Detaillierte Informationen zum CANsync

---

Fehlernummern Up/Download, die die CANsync-Slave-Anschaltung an den CANsync-Master sendet; Anzeige nur in CANsync-Master:

Fehlernummer	Bedeutung
16#0001	CANsync-Slave quittiert falsche Blocknummer
16#0002	eingetragenen Länge > 300 Bytes
16#0100	CANsync-Slave erwartet Block mit Nummer, im Zähler eingetragen ist
16#0101	CANsync-Slave erwartet Block-Ende
16#0102	CANsync-Slave erwartet noch nicht Block-Ende
16#0103	CANsync-Slave bricht Up/Download ab
16#0104	Up/Download nicht möglich
16#0105	Basisadresse nicht erlaubt
16#0106	reserviert
16#0107	Blocklänge > maximale Blocklänge des CANsync-Slaves
16#0108	Telegramm-Modedefehler (in dieser Phase nicht erlaubter Mode)

## CANSYNC-FUNKTIONSBAUSTEINE

In diesem Kapitel werden die einzelnen Funktionsbausteine der CANSync-Bibliothek beschrieben. Diese Beschreibung finden Sie auch in der Online-Hilfe des jeweiligen Funktionsbausteins.

### 4.1 Funktionsbausteine für den synchronisierten CAN - Übersicht

Zusätzlich zu den Standardfunktionen können Sie herstellerdefinierte Funktionen verwenden, wenn Sie herstellerdefinierte Bibliotheken in einem Projekt angemeldet haben.

Anmerkung: Das Anmelden von Bibliotheken ist in der allgemeinen Hilfe beschrieben.

Folgende Funktionsbausteine für synchronisierten CAN sind verfügbar:

Funktion	Kurzbeschreibung
CANSync_BC_MA0	sendet Broadcastkommando des CANSync-Masters in Sendebereich 0 (höchste Priorität)
CANSync_BC_MA1	sendet Broadcastkommando des CANSync-Masters in Sendebereich 1 (mittlere Priorität)
CANSync_BC_MA2	sendet Broadcastkommando des CANSync-Masters in Sendebereich 2 (niedrige Priorität)
CANSync_BC_SL	empfängt Broadcastkommandos des CANSync-Masters
CANSync_COMM_CONTROL_MA	konfiguriert die Benutzung des Kommandokanals einer CANSync-Master-Anschaltung
CANSync_CONTROLWORD_MA	sendet Steuerwortkommandos des CANSync-Masters
CANSync_CONTROLWORD_SL	empfängt Steuerwort des CANSync-Masters in einer CANSync-Slave-Anschaltung
CANSync_INIT	initialisiert eine CANSync-Anschaltung
CANSync_MODE_MA	stellt Betriebsart einer CANSync-Master-Anschaltung ein
CANSync_MODE_SL	stellt Betriebsart einer CANSync-Slave-Anschaltung ein
CANSync_PAR_READ_MA	CANSync-Master fordert über CANSync einen Parameterwert vom CANSync-Slave an

## 4.1 Funktionsbausteine für den synchronisierten CAN - Übersicht

<b>Funktion</b>	<b>Kurzbeschreibung</b>
CANsync_PAR_SL	erkennt Parameteranforderung oder übertragenen Parameter
CANsync_PAR_WRITE_MA	CANsync-Master sendet über CANsync einen Parameterwert an CANsync-Slave
CANsync_PD_CFG_MA	konfiguriert Belegung der Sollwertkanäle der CAN-Anschaltung für einen CANsync-Master
CANsync_PD_CFG_READ_MA	konfiguriert Belegung der Istwertkanäle der CAN-Anschaltung für einen CANsync-Master
CANsync_PD_CFG_READ_SL	konfiguriert Belegung der Istwertkanäle der CAN-Anschaltung für einen CANsync-Slave
CANsync_PD_CFG_SL	konfiguriert Belegung der Sollwert- und der Istwertkanäle der CAN-Anschaltung für einen CANsync-Slave
CANsync_PD_COMM_MA	kopiert die Prozessdaten (Sollwerte und Istwerte der CANsync-Anschaltung für einen CANsync-Master
CANsync_PD_COMM_READ_MA	kopiert in einer CANsync Anschaltung (Master) die Prozessdaten-Istwerte von einem CANsync-Slave
CANsync_PD_COMM_READ_SL	kopiert in einer CANsync-Anschaltung (Slave) die Prozessdaten-Istwerte von einem CANsync-Slave
CANsync_PD_COMM_SL	kopiert die Prozessdaten (Soll- und Istwerte) der CANsync-Anschaltung für einen CANsync-Slave
CANsync_SL_TYP_INIT	CANsync Slave-Typen (Initialisierung)
CANsync_UPDOWNLOAD_MA	CANsync Updownload Master
CANsync_UPDOWNLOAD_SL	CANsync Updownload Slave

## 4.2 CANsync\_BC\_MA0

### Beschreibung

Diesen Funktionsbaustein für CANsync können Sie verwenden, um ein Broadcastkommando des CANsync-Masters im Sendebereich 0 (höchste Priorität) zu senden.

### HINWEIS



Der FB CANsync\_BC\_MA0 verwendet die Bibliothek BM\_TYPES\_20bd03 oder höher.

Parameter Eingang	Datentyp	Beschreibung
_BASE	CANsync_PLC_MA_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung
us_BC_CMD_NR	USINT	Kommandonummer des Broadcastkommandos
si_BC_BYTE	SINT	Datenbyte des Broadcastkommandos
i_BC_WORD	INT	Datenwort des Broadcastkommandos
d_SL_MASK	DWORD	Bitleiste an welche CANsync-Slaves das Broadcastkommando geschickt werden soll
x_EN	BOOL	Freigabe

Parameter Ausgang	Datentyp	Beschreibung
_BASE	CANsync_PLC_MA_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung
x_OK	BOOL	OK-Bit

Es gibt beim CANsync 3 Sendebereiche für die Broadcast-Telegramme mit den Broadcastkommandos (Sendebereich 0 bis 2). In einem CANsync-Intervall kann nur ein Broadcast-Telegramm gesendet werden. Die höchste Priorität hat der Sendebereich 0. Der FB CANsync\_BC\_MA0 nutzt den Sendebereich 0 zum Senden des Broadcast-Telegramms.

Ein-/Ausgang \_BASE:

Am Ein-/Ausgang \_BASE wird, entsprechend dem Steckplatz (Slot G bis M) des Optionsmoduls, die globale Variable `_CANsyncMaster_Ctrl_Slot_G` (bis `_CANsyncMaster_Ctrl_Slot_M`) angeschlossen.

Ist diese globale Variable nicht im Projekt vorhanden, legen Sie, je nach Steckplatz (Slot G bis M), die globale Variable `_CANsyncMaster_Ctrl_Slot_G` (bis `_CANsyncMaster_Ctrl_Slot_M`) vom Datentyp `CANsync_PLC_MA_BMSTRUCT` an.

Diese Variable muß über die Deklaration der globalen Variablen auf die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul gelegt werden.

Die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Master (BM4-O-CAN-06) in Abhängigkeit vom Steckplatz ist:

Steckplatz (Slot)	Basisadresse der CANsync-Anschaltung
G	%MB3.2001792
H	%MB3.3001792
J	%MB3.4001792
K	%MB3.5001792
L	%MB3.6001792
M	%MB3.7001792

Beispiel:

CANsync-Anschaltung auf Optionsmodul CANsync-Master im Steckplatz G des b maXX

```
_CANsyncMaster_Ctrl_Slot_G AT %MB3.2001792 :
                                CANsync_PLC_MA_BMSTRUCT;
```

dabei ist:

<code>_CANsyncMaster_Ctrl_Slot_G</code>	der Variablenname mit der Datentypkurzbezeichnung "_" für Struct
<code>CANsync_PLC_MA_BMSTRUCT</code>	der Datentyp
<code>%MB3.2001792</code>	die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Master im Steckplatz G des b maXX

Eingang `us_BC_CMD_NR`:

Am Eingang `us_BC_CMD_NR` wird die Kommandonummer des Broadcastkommandos angegeben.

Eingänge `si_BC_BYTE`, `i_BC_WORD`:

An den Eingängen `si_BC_BYTE` und `i_BC_WORD` sind je nach Kommandonummer die zugehörigen Daten anzuschließen (siehe weiter unten Liste der Broadcastkommandos).

Eingang `d_SL_MASK`:

Am Eingang `d_SL_MASK` wird angegeben, welche CANsync-Slaves das Kommando empfangen sollen. Dazu muss für jeden CANsync-Slave die seiner Slavenummer entsprechende Bitnummer auf TRUE gesetzt werden. Für den CANsync-Slave mit der Slavenummer 0 wird das Bit 0 von `d_SL_MASK` auf TRUE gesetzt, für den CANsync-Slave mit der Slavenummer 1 wird das Bit 1 von `d_SL_MASK` auf TRUE gesetzt usw. Wenn alle CANsync-Slaves angesprochen werden sollen, muss `d_SL_MASK = 16#FFFFFFF` gesetzt werden.



Eingang x\_EN:

Wenn der Eingang x\_EN auf TRUE gesetzt wird, wird das Broadcastkommando zum Senden eingetragen. Die CANsync-Anschaltung sendet das Broadcastkommando dann im Broadcast-Telegramm an die ausgewählten CANsync-Slaves (Eingang d\_SL\_MASK). Solange x\_EN auf TRUE bleibt, wird bei jedem Aufruf des FB das Broadcastkommando zum Senden eingetragen. Damit würde jedoch der Kommandokanal des CANsync dauernd belegt. Deshalb sollte der Eingang x\_EN nur für ein CANsync-Intervall auf TRUE gesetzt sein, um das Broadcastkommando einmal abzuschicken.

Ausgang x\_OK:

Der Ausgang x\_OK zeigt mit TRUE an, dass das letzte Broadcast-Telegramm gesendet wurde. Der Ausgang x\_OK ist FALSE wenn kein Broadcast-Telegramm gesendet wurde.

Liste der Broadcastkommandos:

Kommando- nummer	Bedeutung
1	Steuerwort si_BC_BYTE: nicht benutzt i_BC_WORD: Steuerwort
2 bis 127	reserviert
128 – 255	stehen dem Anwender zur Verfügung

### 4.3 CANsync\_BC\_MA1

#### Beschreibung

Diesen Funktionsbaustein für CANsync können Sie verwenden, um ein Broadcastkommando des CANsync-Masters im Sendebereich 1 (mittlere Priorität) zu senden.

#### HINWEIS



Der FB CANsync\_BC\_MA1 verwendet die Bibliothek BM\_TYPES\_20bd03 oder höher.

Parameter Eingang	Datentyp	Beschreibung
_BASE	CANsync_PLC_MA_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung
us_BC_CMD_NR	USINT	Kommandonummer des Broadcastkommandos
si_BC_BYTE	SINT	Datenbyte des Broadcastkommandos
i_BC_WORD	INT	Datenwort des Broadcastkommandos
d_SL_MASK	DWORD	Bitleiste an welche CANsync-Slaves das Broadcastkommando geschickt werden soll
x_EN	BOOL	Freigabe

Parameter Ausgang	Datentyp	Beschreibung
_BASE	CANsync_PLC_MA_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung
x_OK	BOOL	OK-Bit

Es gibt beim CANsync 3 Sendebereiche für die Broadcast-Telegramme mit den Broadcastkommandos (Sendebereich 0 bis 2). In einem CANsync-Intervall kann nur ein Broadcast-Telegramm gesendet werden. Die mittlere Priorität hat der Sendebereich 1. Der FB CANsync\_BC\_MA1 nutzt den Sendebereich 1 zum Senden des Broadcast-Telegramms.

Ein-/Ausgang \_BASE:

Am Ein-/Ausgang \_BASE wird, entsprechend dem Steckplatz (Slot G bis M) des Optionsmoduls, die globale Variable `_CANsyncMaster_Ctrl_Slot_G` (bis `_CANsyncMaster_Ctrl_Slot_M`) angeschlossen.

Ist diese globale Variable nicht im Projekt vorhanden, legen Sie, je nach Steckplatz (Slot G bis M), die globale Variable `_CANsyncMaster_Ctrl_Slot_G` (bis `_CANsyncMaster_Ctrl_Slot_M`) vom Datentyp `CANsync_PLC_MA_BMSTRUCT` an.

Diese Variable muß über die Deklaration der globalen Variablen auf die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul gelegt werden.

Die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Master (BM4-O-CAN-06) in Abhängigkeit vom Steckplatz ist:

Steckplatz (Slot)	Basisadresse der CANsync-Anschaltung
G	%MB3.2001792
H	%MB3.3001792
J	%MB3.4001792
K	%MB3.5001792
L	%MB3.6001792
M	%MB3.7001792

Beispiel:

CANsync-Anschaltung auf Optionsmodul CANsync-Master im Steckplatz G des b maXX

```
_CANsyncMaster_Ctrl_Slot_G AT %MB3.2001792 :
                                CANsync_PLC_MA_BMSTRUCT;
```

dabei ist:

- `_CANsyncMaster_Ctrl_Slot_G` der Variablenname mit der Datentypkurzbezeichnung "\_" für Struct
- `CANsync_PLC_MA_BMSTRUCT` der Datentyp
- `%MB3.2001792` die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Master im Steckplatz G des b maXX

Eingang `us_BC_CMD_NR`:

Am Eingang `us_BC_CMD_NR` wird die Kommandonummer des Broadcastkommandos angegeben.

Eingänge `si_BC_BYTE`, `i_BC_WORD`:

An den Eingängen `si_BC_BYTE` und `i_BC_WORD` sind je nach Kommandonummer die zugehörigen Daten anzuschließen (siehe weiter unten Liste der Broadcastkommandos).

Eingang `d_SL_MASK`:

Am Eingang `d_SL_MASK` wird angegeben, welche CANsync-Slaves das Kommando empfangen sollen. Dazu muss für jeden CANsync-Slave die seiner Slavenummer entsprechende Bitnummer auf TRUE gesetzt werden. Für den CANsync-Slave mit der Slavenummer 0 wird das Bit 0 von `d_SL_MASK` auf TRUE gesetzt, für den CANsync-Slave mit der Slavenummer 1 wird das Bit 1 von `d_SL_MASK` auf TRUE gesetzt usw. Wenn alle CANsync-Slaves angesprochen werden sollen, muss `d_SL_MASK = 16#FFFFFFFF` gesetzt werden.

Eingang x\_EN:

Wenn der Eingang x\_EN auf TRUE gesetzt wird, wird das Broadcastkommando zum Senden eingetragen. Die CANsync-Anschaltung sendet das Broadcastkommando dann im Broadcast-Telegramm an die ausgewählten CANsync-Slaves (Eingang d\_SL\_MASK). Solange x\_EN auf TRUE bleibt, wird bei jedem Aufruf des FB das Broadcastkommando zum Senden eingetragen. Damit würde jedoch der Kommandokanal des CANsync dauernd belegt. Deshalb sollte der Eingang x\_EN nur für ein CANsync-Intervall auf TRUE gesetzt sein, um das Broadcastkommando einmal abzuschicken.

Ausgang x\_OK:

Der Ausgang x\_OK zeigt mit TRUE an, dass das letzte Broadcast-Telegramm gesendet wurde. Der Ausgang x\_OK ist FALSE wenn kein Broadcast-Telegramm gesendet wurde.

Liste der Broadcastkommandos:

Kommando- nummer	Bedeutung
1	Steuerwort si_BC_BYTE: nicht benutzt i_BC_WORD: Steuerwort
2 bis 127	reserviert
128 bis 255	stehen dem Anwender zur Verfügung

## 4.4 CANsync\_BC\_MA2

### Beschreibung

Diesen Funktionsbaustein für CANsync können Sie verwenden, um ein Broadcastkommando des CANsync-Masters im Sendebereich 2 (niedrigste Priorität) zu senden.



### HINWEIS

Der FB CANsync\_BC\_MA2 verwendet die Bibliothek BM\_TYPES\_20bd03 oder höher.

Parameter Eingang	Datentyp	Beschreibung
_BASE	CANsync_PLC_MA_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung
us_BC_CMD_NR	USINT	Kommandonummer des Broadcastkommandos
si_BC_BYTE	SINT	Datenbyte des Broadcastkommandos
i_BC_WORD	INT	Datenwort des Broadcastkommandos
d_SL_MASK	DWORD	Bitleiste an welche CANsync-Slaves das Broadcastkommando geschickt werden soll
x_EN	BOOL	Freigabe

Parameter Ausgang	Datentyp	Beschreibung
_BASE	CANsync_PLC_MA_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung
x_OK	BOOL	OK-Bit

Es gibt beim CANsync 3 Sendebereiche für die Broadcast-Telegramme mit den Broadcastkommandos (Sendebereich 0 bis 2). In einem CANsync-Intervall kann nur ein Broadcast-Telegramm gesendet werden. Die niedrigste Priorität hat der Sendebereich 2. Der FB CANsync\_BC\_MA2 nutzt den Sendebereich 2 zum Senden des Broadcast-Telegramms.

Ein-/Ausgang \_BASE:

Am Ein-/Ausgang \_BASE wird, entsprechend dem Steckplatz (Slot G bis M) des Optionsmoduls, die globale Variable `_CANsyncMaster_Ctrl_Slot_G` (bis `_CANsyncMaster_Ctrl_Slot_M`) angeschlossen.

Ist diese globale Variable nicht im Projekt vorhanden, legen Sie, je nach Steckplatz (Slot G bis M), die globale Variable `_CANsyncMaster_Ctrl_Slot_G` (bis `_CANsyncMaster_Ctrl_Slot_M`) vom Datentyp `CANsync_PLC_MA_BMSTRUCT` an.

Diese Variable muß über die Deklaration der globalen Variablen auf die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul gelegt werden.

Die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Master (BM4-O-CAN-06) in Abhängigkeit vom Steckplatz ist:

Steckplatz (Slot)	Basisadresse der CANsync-Anschaltung
G	%MB3.2001792
H	%MB3.3001792
J	%MB3.4001792
K	%MB3.5001792
L	%MB3.6001792
M	%MB3.7001792

Beispiel:

CANsync-Anschaltung auf Optionsmodul CANsync-Master im Steckplatz G des b maXX

```
_CANsyncMaster_Ctrl_Slot_G AT %MB3.2001792 :
                                CANsync_PLC_MA_BMSTRUCT;
```

dabei ist:

<code>_CANsyncMaster_Ctrl_Slot_G</code>	der Variablenname mit der Datentypkurzbezeichnung "_" für Struct
<code>CANsync_PLC_MA_BMSTRUCT</code>	der Datentyp
<code>%MB3.2001792</code>	die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Master im Steckplatz G des b maXX

Eingang `us_BC_CMD_NR`:

Am Eingang `us_BC_CMD_NR` wird die Kommandonummer des Broadcastkommandos angegeben.

Eingänge `si_BC_BYTE`, `i_BC_WORD`:

An den Eingängen `si_BC_BYTE` und `i_BC_WORD` sind je nach Kommandonummer die zugehörigen Daten anzuschließen (siehe weiter unten Liste der Broadcastkommandos).

Eingang `d_SL_MASK`:

Am Eingang `d_SL_MASK` wird angegeben, welche CANsync-Slaves das Kommando empfangen sollen. Dazu muss für jeden CANsync-Slave die seiner Slavenummer entsprechende Bitnummer auf TRUE gesetzt werden. Für den CANsync-Slave mit der Slavenummer 0 wird das Bit 0 von `d_SL_MASK` auf TRUE gesetzt, für den CANsync-Slave mit der Slavenummer 1 wird das Bit 1 von `d_SL_MASK` auf TRUE gesetzt usw. Wenn alle CANsync-Slaves angesprochen werden sollen, muss `d_SL_MASK = 16#FFFFFFF` gesetzt werden.

Eingang x\_EN:

Wenn der Eingang x\_EN auf TRUE gesetzt wird, wird das Broadcastkommando zum Senden eingetragen. Die CANsync-Anschaltung sendet das Broadcastkommando dann im Broadcast-Telegramm an die ausgewählten CANsync-Slaves (Eingang d\_SL\_MASK). Solange x\_EN auf TRUE bleibt, wird bei jedem Aufruf des FB das Broadcastkommando zum Senden eingetragen. Damit würde jedoch der Kommandokanal des CANsync dauernd belegt. Deshalb sollte der Eingang x\_EN nur für ein CANsync-Intervall auf TRUE gesetzt sein, um das Broadcastkommando einmal abzuschicken.

Ausgang x\_OK:

Der Ausgang x\_OK zeigt mit TRUE an, dass das letzte Broadcast-Telegramm gesendet wurde. Der Ausgang x\_OK ist FALSE wenn kein Broadcast-Telegramm gesendet wurde.

Liste der Broadcastkommandos:

Kommando- nummer	Bedeutung
1	Steuerwort si_BC_BYTE: nicht benutzt i_BC_WORD: Steuerwort
2 bis 127	reserviert
128 bis 255	stehen dem Anwender zur Verfügung

### 4.5 CANsync\_BC\_SL

#### Beschreibung

Diesen Funktionsbaustein für CANsync können Sie verwenden, um mit einem CANsync-Slave ein Broadcastkommando des CANsync-Masters zu empfangen.



#### HINWEIS

Der FB CANsync\_BC\_SL verwendet die Bibliothek BM\_TYPES\_20bd03 oder höher.

Parameter Eingang	Datentyp	Beschreibung
_BASE	CANsync_PLC_SL_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung

Parameter Ausgang	Datentyp	Beschreibung
_BASE	CANsync_PLC_SL_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung
us_BC_CMD_NR	USINT	Kommandonummer des Broadcastkommandos
si_BC_BYTE	SINT	Datenbyte des Broadcastkommandos
i_BC_WORD	INT	Datenwort des Broadcastkommandos
si_BC_RECEIVED	SINT 0, 2	Anzeige, dass ein Kommando empfangen wurde

Dieser FB zeigt an si\_BC\_RECEIVED an, dass ein Broadcastkommando des CANsync-Masters empfangen wurde und gibt die Kommandonummer (us\_BC\_CMD\_NR) sowie den Inhalt des Broadcastkommandos (si\_BC\_BYTE, i\_BC\_WORD) aus.

Ein-/Ausgang \_BASE:

Am Ein-/Ausgang \_BASE wird, entsprechend dem Steckplatz (Slot G bis M) des Optionsmoduls, die globale Variable \_CANsyncSlave\_Ctrl\_Slot\_G (bis \_CANsyncSlave\_Ctrl\_Slot\_M) angeschlossen.

Ist diese globale Variable nicht im Projekt vorhanden, legen Sie, je nach Steckplatz (Slot G bis M), die globale Variable \_CANsyncSlave\_Ctrl\_Slot\_G (bis \_CANsyncSlave\_Ctrl\_Slot\_M) vom Datentyp CANsync\_PLC\_SL\_BMSTRUCT an.

Diese Variable muß über die Deklaration der globalen Variablen auf die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul gelegt werden.

Die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Slave für b maXX PLC (BM4-O-CAN-05, Dip-Schalter 9 = ON) in Abhängigkeit vom Steckplatz ist:



Steckplatz (Slot)	Basisadresse der CANsync-Anschaltung
G	%MB3.2001792
H	%MB3.3001792
J	%MB3.4001792
K	%MB3.5001792
L	%MB3.6001792
M	%MB3.7001792

Beispiel:

CANsync-Anschaltung auf Optionsmodul CANsync-Slave im Steckplatz G des b maXX

```
_CANsyncSlave_Ctrl_Slot_G AT %MB3.2001792 :
                                CANsync_PLC_SL_BMSTRUCT;
```

dabei ist:

<code>_CANsyncSlave_Ctrl_Slot_G</code>	der Variablenname mit der Datentypkurzbezeichnung "_" für Struct
<code>CANsync_PLC_SL_BMSTRUCT</code>	der Datentyp
<code>%MB3.2001792</code>	die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Slave im Steckplatz G des b maXX

Ausgang `us_BC_CMD_NR`:

Am Ausgang `us_BC_CMD_NR` wird die Kommandonummer des Broadcastkommandos ausgegeben.

Ausgänge `si_BC_BYTE`, `i_BC_WORD`:

An den Ausgängen `si_BC_BYTE` und `i_BC_WORD` sind je nach Kommandonummer die zugehörigen Daten zu lesen.

Ausgang `si_BC_RECEIVED`:

Am Ausgang `si_BC_RECEIVED` wird angezeigt, ob ein Broadcast-Telegramm (mit einem Broadcastkommando) empfangen wurde. Dann zeigt `si_BC_RECEIVED` eine 2 an. Andernfalls zeigt `si_BC_RECEIVED` eine 0 an. Nur wenn das Broadcast-Telegramm empfangen wurde, werden die zugehörigen Daten (`us_BC_CMD_NR`, `si_BC_BYTE`, `i_BC_WORD`) ausgelesen. Ansonsten werden die alten Werte weiter angezeigt.

Liste der Broadcastkommandos:

<b>Kommando- nummer</b>	<b>Bedeutung</b>
1	Steuerwort si_BC_BYTE: nicht benutzt i_BC_WORD: Steuerwort
2 bis 127	reserviert
128 bis 255	stehen dem Anwender zur Verfügung

## 4.6 CANsync\_COMM\_CONTROL\_MA

### Beschreibung

Diesen Funktionsbaustein für CANsync können Sie verwenden, um die Benutzung des Kommandokanals einer CANsync-Anschaltung zu konfigurieren.



### HINWEIS

Der FB CANsync\_COMM\_CONTROL\_MA verwendet die Bibliothek BM\_TYPES\_20bd03 oder höher.

Parameter Eingang	Datentyp	Beschreibung
_BASE	CANsync_PLC_MA_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung
si_SL_NR_CTRL	SINT -128, 0 bis 31	Slavenummer des CANsync-Slaves mit Steuerwort-Senden-Auftrag
x_EN_CTRL	BOOL	Freigabe für si_SL_NR_CTRL
si_SL_NR_PAR	SINT -128, 0 bis 31	Slavenummer des CANsync-Slaves mit Parameter-Auftrag
x_EN_PAR	BOOL	Freigabe für si_SL_NR_PAR
si_SL_NR_UDL	SINT -128, 0 bis 31	Slavenummer des CANsync-Slaves mit Up/Download-Auftrag
x_EN_UDL	BOOL	Freigabe für si_SL_NR_UDL
si_MAX_SL_NR <sup>a)</sup>	SINT -1, 0 bis 31	maximale Slavenummer für automatische Erhöhung <sup>a)</sup>

a) Dieser Eingang entspricht dem Eingang si\_MAX\_SL\_NR am FB CANsync\_PD\_COMM\_MA als maximale Slavenummer für automatische Istwert-Telegramm-Anforderung.

Parameter Ausgang	Datentyp	Beschreibung
_BASE	CANsync_PLC_MA_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung

Mit diesem FB wird die Slavenummer des CANsync-Slaves angegeben, für den abgefragt wird ob Steuerwort-Senden-Aufträge (si\_SL\_NR\_CTRL, x\_EN\_CTRL), Parameter-Aufträge (si\_SL\_NR\_PAR, x\_EN\_PAR) und Up-/Download-Aufträge (si\_SL\_NR\_UDL, x\_EN\_UDL) vorhanden sind.



### HINWEIS

Der CANsync-Master kann in jedem CANsync-Intervall ein Kommando-Telegramm senden.

Die unterschiedlichen Telegramme im Kommandokanal werden in einer durch Prioritäten vorgegebenen Reihenfolge abgearbeitet:

Telegrammtyp	Priorität
Broadcast-Telegramm 0	höchste
Broadcast-Telegramm 1	↑
Broadcast-Telegramm 2	
Steuerwort-Telegramme	
Parameter-Telegramme	↓
Up-/Download-Telegramme	niedrigste

Die Konsequenz dieser Prioritäten ist, dass kein anderes Telegramm verschickt werden kann, wenn ein höherprioritäres gesendet wird. Sendet man z. B. in jedem CANsync-Intervall das Steuerwort-Telegramm, so kann nie ein Parameter-Telegramm oder Up-/Download-Telegramm verschickt werden!

Ist die Abfrage nach vorhanden Aufträgen zu mehreren CANsync-Slaves notwendig, wird die CANsync-Anschaltung so eingestellt:

In jedem CANsync-Intervall wird die Slavenummer des CANsync-Slaves (für den abgefragt wird ob Aufträge vorhanden sind) automatisch um eins erhöht. Diese Erhöhung wird bis zur `si_MAX_SL_NR` durchgeführt. Danach wird wieder mit der Abfrage für den CANsync-Slave mit der Slavenummer 0 begonnen usw.

Ein-/Ausgang `_BASE`:

Am Ein-/Ausgang `_BASE` wird, entsprechend dem Steckplatz (Slot G bis M) des Optionsmoduls, die globale Variable `_CANsyncMaster_Ctrl_Slot_G` (bis `_CANsyncMaster_Ctrl_Slot_M`) angeschlossen.

Ist diese globale Variable nicht im Projekt vorhanden, legen Sie, je nach Steckplatz (Slot G bis M), die globale Variable `_CANsyncMaster_Ctrl_Slot_G` (bis `_CANsyncMaster_Ctrl_Slot_M`) vom Datentyp `CANsync_PLC_MA_BMSTRUCT` an.

Diese Variable muß über die Deklaration der globalen Variablen auf die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul gelegt werden.

Die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Master (BM4-O-CAN-06) in Abhängigkeit vom Steckplatz ist:

Steckplatz (Slot)	Basisadresse der CANsync-Anschaltung
G	%MB3.2001792
H	%MB3.3001792
J	%MB3.4001792
K	%MB3.5001792
L	%MB3.6001792
M	%MB3.7001792

Beispiel:

CANsync-Anschaltung auf Optionsmodul CANsync-Master im Steckplatz G des b maXX

```
_CANsyncMaster_Ctrl_Slot_G AT %MB3.2001792 :
                                CANsync_PLC_MA_BMSTRUCT;
```

dabei ist:

`_CANsyncMaster_Ctrl_Slot_G` der Variablenname mit der Datentypkurzbezeichnung "\_" für Struct

`CANsync_PLC_MA_BMSTRUCT` der Datentyp

`%MB3.2001792` die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Master im Steckplatz G des b maXX

Eingänge `si_SL_NR_CTRL`, `x_EN_CTRL`:

Am Eingang `si_SL_NR_CTRL` wird die Slavenummer des CANsync-Slaves angegeben, für den abgefragt werden soll, ob ein Steuerwort-Senden-Auftrag vorhanden ist (FB `CANsync_CONTROLWORD_MA`).

Mit `si_SL_NR_CTRL = -128` wird eingestellt, dass in jedem CANsync-Intervall die Slavenummer des CANsync-Slaves (für den abgefragt werden soll, ob ein Steuerwort-Senden-Auftrag vorhanden ist) automatisch um eins erhöht wird, bis `si_MAX_SL_NR` erreicht ist. Danach wird wieder mit der Abfrage für den CANsync-Slave mit der Slavenummer 0 begonnen usw.

Voreinstellung ist `si_SL_NR_CTRL = -128`, d. h. automatisches Erhöhen bis `si_MAX_SL_NR`.

Die Einstellung an `si_SL_NR_CTRL` (auch wenn nicht belegt) wird nur übernommen, wenn `x_EN_CTRL = TRUE` gesetzt ist. Wenn `x_EN_CTRL = FALSE` ist, wird dieser Teil der Kommandokanal-Konfigurierung nicht verändert.

Eingänge `si_SL_NR_PAR`, `x_EN_PAR`:

Am Eingang `si_SL_NR_PAR` wird die Slavenummer des CANsync-Slaves angegeben, für den abgefragt werden soll, ob ein Parameter-Auftrag vorhanden ist (FB `CANsync_PAR_WRITE_MA` oder `CANsync_PAR_READ_MA`).

Mit `si_SL_NR_PAR = -128` wird eingestellt, dass in jedem CANsync-Intervall die Slavenummer des CANsync-Slaves (für den abgefragt werden soll, ob ein Parameter-Auftrag vorhanden ist) automatisch um eins erhöht wird, bis `si_MAX_SL_NR` erreicht ist. Danach wird wieder mit der Abfrage für den CANsync-Slave mit der Slavenummer 0 begonnen usw.

Voreinstellung ist `si_SL_NR_PAR = -128`, d. h. automatisches Erhöhen bis `si_MAX_SL_NR`.

Die Einstellung an `si_SL_NR_PAR` (auch wenn nicht belegt) wird nur übernommen, wenn `x_EN_PAR = TRUE` gesetzt ist. Wenn `x_EN_PAR = FALSE` ist, wird dieser Teil der Kommandokanal-Konfigurierung nicht verändert.

Eingänge si\_SL\_NR\_UDL, x\_EN\_UDL:

Am Eingang si\_SL\_NR\_UDL wird die Slavenummer des CANsync-Slaves angegeben, für den abgefragt werden soll, ob ein Up-/Download-Auftrag vorhanden ist (FB CANsync\_UPDOWNLOAD\_MA).

Mit si\_SL\_NR\_UDL = -128 wird eingestellt, dass in jedem CANsync-Intervall die Slavenummer des CANsync-Slaves (für den abgefragt werden soll, ob ein Up-/Download-Auftrag vorhanden ist) automatisch um eins erhöht wird, bis si\_MAX\_SL\_NR erreicht ist. Danach wird wieder mit der Abfrage für den CANsync-Slave mit der Slavenummer 0 begonnen usw.

Voreinstellung ist si\_SL\_NR\_UDL = -128, d. h. automatisches Erhöhen bis si\_MAX\_SL\_NR.

Die Einstellung an si\_SL\_NR\_UDL (auch wenn nicht belegt) wird nur übernommen, wenn x\_EN\_PAR = TRUE gesetzt ist. Wenn x\_EN\_UDL = FALSE ist, wird dieser Teil der Kommandokanal-Konfigurierung nicht verändert.

Eingang si\_MAX\_SL\_NR:

Die höchste Slavenummer eines CANsync-Slaves wird am Eingang si\_MAX\_SL\_NR angegeben. Bis zu dieser Slavenummer wird abgefragt ob Steuerwort-Senden-Aufträge, Parameter-Aufträge und Up-/Download-Aufträge vorhanden sind. Dazu werden dann si\_SL\_NR\_CTRL, si\_SL\_NR\_PAR, si\_SL\_UDL nicht belegt und si\_EN\_CTRL, si\_EN\_PAR und si\_EN\_UDL auf TRUE gesetzt.

Ist si\_MAX\_SL\_NR = -1, bleibt auf der CANsync-Anschaltung der Wert unverändert. Voreinstellung ist si\_MAX\_SL\_NR = -1, d. h. auf der CANsync-Anschaltung bleibt der Wert unverändert.



### HINWEIS

Dieser Eingang entspricht dem Eingang si\_MAX\_SL\_NR am FB CANsync\_PD\_COMM\_MA. D. h. mit dem Eingang si\_MAX\_SL\_NR am FB CANsync\_COMM\_CONTROL\_MA **oder** dem Eingang si\_MAX\_SL\_NR am FB CANsync\_PD\_COMM\_MA wird die höchste Slavenummer eines CANsync-Slaves angegeben.

Es darf nur einer von beiden Eingängen verwendet werden!

---

## 4.7 CANsync\_CONTROLWORD\_MA

### Beschreibung

Diesen Funktionsbaustein für CANsync können Sie verwenden, um ein Steuerwortkommando des CANsync-Masters an einen CANsync-Slave zu senden.



### HINWEIS

Für den Einsatz dieses FBs ist es notwendig, für den Kommandokanal (CC) die Steuerwortkommandos freizugeben.

Diese Freigabe erfolgt mit dem FB CANsync\_COMM\_CONTROL\_MA.

Der FB CANsync\_CONTROLWORD\_MA verwendet die Bibliothek BM\_TYPES\_20bd03 oder höher.

Parameter Eingang	Datentyp	Beschreibung
_BASE	CANsync_PLC_MA_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung
si_SL_NR_CTRL	SINT 0 bis 31	Slavenummer des CANsync-Slaves, an den das Steuerwort gesendet wird
w_CONTROLWORD	WORD	Steuerwort
x_EN	BOOL	Freigabe

Parameter Ausgang	Datentyp	Beschreibung
_BASE	CANsync_PLC_MA_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung
x_OK	BOOL	OK-Bit

Das Steuerwort (w\_CONTROLWORD), die Freigabe zum Senden (x\_EN) und die Slavenummer des CANsync-Slaves an den das Steuerwort gesendet wird (si\_SL\_NR\_CTRL), wird der CANsync-Anschaltung übergeben. Nachdem die CANsync-Anschaltung den Auftrag erkannt hat (siehe FB CANsync\_COMM\_CONTROL\_MA) wird das Steuerwort an den CANsync-Slave mit einem Steuerwort-Telegramm gesendet und das Senden am Ausgang x\_OK quittiert.

Ein-/Ausgang \_BASE:

Am Ein-/Ausgang \_BASE wird, entsprechend dem Steckplatz (Slot G bis M) des Optionsmoduls, die globale Variable `_CANsyncMaster_Ctrl_Slot_G` (bis `_CANsyncMaster_Ctrl_Slot_M`) angeschlossen.

Ist diese globale Variable nicht im Projekt vorhanden, legen Sie, je nach Steckplatz (Slot G bis M), die globale Variable `_CANsyncMaster_Ctrl_Slot_G` (bis `_CANsyncMaster_Ctrl_Slot_M`) vom Datentyp `CANsync_PLC_MA_BMSTRUCT` an.

Diese Variable muß über die Deklaration der globalen Variablen auf die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul gelegt werden.

Die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Master (BM4-O-CAN-06) in Abhängigkeit vom Steckplatz ist:

Steckplatz (Slot)	Basisadresse der CANsync-Anschaltung
G	%MB3.2001792
H	%MB3.3001792
J	%MB3.4001792
K	%MB3.5001792
L	%MB3.6001792
M	%MB3.7001792

Beispiel:

CANsync-Anschaltung auf Optionsmodul CANsync-Master im Steckplatz G des b maXX

```
_CANsyncMaster_Ctrl_Slot_G AT %MB3.2001792 :
                                CANsync_PLC_MA_BMSTRUCT;
```

dabei ist:

<code>_CANsyncMaster_Ctrl_Slot_G</code>	der Variablenname mit der Datentypkurzbezeichnung "_" für Struct
<code>CANsync_PLC_MA_BMSTRUCT</code>	der Datentyp
<code>%MB3.2001792</code>	die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Master im Steckplatz G des b maXX

Eingang `w_CONTROLWORD`:

An dem Eingang `w_CONTROLWORD` wird das zu sendende Steuerwort angeschlossen.

Eingang `x_EN`:

Wenn der Eingang `x_EN` auf TRUE gesetzt wird, wird das Steuerwort zum Senden eingetragen. Die CANsync-Anschaltung sendet das Steuerwort dann im Steuerwort-Telegramm an den CANsync-Slave `si_SL_NR_CTRL`. Solange `x_EN` auf TRUE bleibt, wird bei jedem Aufruf des FB das Steuerwort zum Senden eingetragen. Dies wird bei jedem Aufruf des FBs neu durchgeführt, solange `x_EN` auf TRUE bleibt. Damit würde jedoch der Kommandokanal des CANsync dauernd belegt. Deshalb sollte der Eingang `x_EN` nur für ein CANsync-Intervall auf TRUE gesetzt sein, um das Steuerwort einmal abzuschicken.

Ausgang `x_OK`:

Der Ausgang `x_OK` zeigt mit TRUE an, dass das letzte Steuerwort-Telegramm gesendet wurde. Der Ausgang `x_OK` ist FALSE wenn kein Steuerwort-Telegramm gesendet wurde.



Ursache für das nicht Senden eines Steuerwort-Telegramm kann die Belegung des Kommandokanals mit höherpriorigen Nachrichten sein (Broadcast-Kommandos; siehe [▶Betriebsanleitung Optionsmodul CANsync-Master für b maXX PLC](#), Abschnitt Kommandokanal). Weiterhin kann die Ursache die fehlende Freigabe des Kommandokanals für Steuerwortkommandos sein (siehe FB [▶CANsync\\_COMM\\_CONTROL\\_MA](#) ab Seite 113).

### 4.8 CANsync\_CONTROLWORD\_SL

#### Beschreibung

Diesen Funktionsbaustein für CANsync können Sie verwenden, um ein Steuerwort des CANsync-Masters in einer CANsync-Slave-Anschaltung zu empfangen.



#### HINWEIS

Der FB CANsync\_CONTROLWORD\_SL verwendet die Bibliothek BM\_TYPES\_20bd03 oder höher.

Parameter Eingang	Datentyp	Beschreibung
_BASE	CANsync_PLC_SL_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung

Parameter Ausgang	Datentyp	Beschreibung
_BASE	CANsync_PLC_SL_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung
w_CONTROLWORD	WORD	Steuerwort
si_RECEIVED	SINT 0, 2	Anzeige, dass ein Steuerwort empfangen wurde

Empfängt der CANsync-Slave ein Steuerwort-Telegramm vom CANsync-Master gibt der FB CANsync\_CONTROLWORD\_SL das empfangene Steuerwort am Ausgang w\_CONTROLWORD aus.

Ein-/Ausgang \_BASE:

Am Ein-/Ausgang \_BASE wird, entsprechend dem Steckplatz (Slot G bis M) des Optionsmoduls, die globale Variable \_CANsyncSlave\_Ctrl\_Slot\_G (bis \_CANsyncSlave\_Ctrl\_Slot\_M) angeschlossen.

Ist diese globale Variable nicht im Projekt vorhanden, legen Sie, je nach Steckplatz (Slot G bis M), die globale Variable \_CANsyncSlave\_Ctrl\_Slot\_G (bis \_CANsyncSlave\_Ctrl\_Slot\_M) vom Datentyp CANsync\_PLC\_SL\_BMSTRUCT an.

Diese Variable muß über die Deklaration der globalen Variablen auf die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul gelegt werden.

Die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Slave für b maXX PLC (BM4-O-CAN-05, Dip-Schalter 9 = ON) in Abhängigkeit vom Steckplatz ist:

Steckplatz (Slot)	Basisadresse der CANsync-Anschaltung
G	%MB3.2001792
H	%MB3.3001792
J	%MB3.4001792
K	%MB3.5001792
L	%MB3.6001792
M	%MB3.7001792

Beispiel:

CANsync-Anschaltung auf Optionsmodul CANsync-Slave im Steckplatz G des b maXX

```
_CANsyncSlave_Ctrl_Slot_G AT %MB3.2001792 :
                                CANsync_PLC_SL_BMSTRUCT;
```

dabei ist:

<code>_CANsyncSlave_Ctrl_Slot_G</code>	der Variablenname mit der Datentypkurzbezeichnung "_" für Struct
<code>CANsync_PLC_SL_BMSTRUCT</code>	der Datentyp
<code>%MB3.2001792</code>	die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Slave im Steckplatz G des b maXX

Ausgang `w_CONTROLWORD`:

Am Ausgang `w_CONTROLWORD` wird das empfangene Steuerwort ausgegeben.

Ausgang `si_RECEIVED`:

Am Ausgang `si_RECEIVED` wird angezeigt, ob ein Steuerwort-Telegramm empfangen wurde. Dann zeigt der Ausgang eine 2 an. Andernfalls wird am Ausgang `si_RECEIVED` eine 0 angezeigt.

Das Steuerwort-Telegramm ist ein Spezialfall des Broadcast-Telegramms (mit Broadcastkommandonummer 1).

### 4.9 CANsync\_INIT

#### Beschreibung

Diesen Funktionsbaustein für CANsync können Sie verwenden, um eine CANsync-Anschaltung (CANsync-Master oder CANsync-Slave) zu initialisieren.



#### HINWEIS

Der FB CANsync\_INIT verwendet die Bibliothek BM\_TYPES\_20bd03 oder höher.

Parameter Eingang	Datentyp	Beschreibung
_BASE	CANsync_PLC_INIT_BMSTRUCT	Initialisierungsdaten für die CANsync-Anschaltung
x_SL	BOOL	Auswahl CANsync-Slave / CANsync-Master
x_SYNC_IN	BOOL	Konfigurierung SYNC-Signal empfangen / senden
x_SYNC_MODE	BOOL	Synchron-Betrieb einrichten
x_ASYNC_MODE0	BOOL	reserviert
x_ASYNC_MODE1	BOOL	reserviert
a_SL_TYP	BYTE_32_BMARRAY	Initialisierungsdaten „Slave-Typen“
b_ACCEPT_CODE	BYTE	Acceptance Code
b_ACCEPT_MASK	BYTE	Acceptance Mask
b_BIT_TIMING0	BYTE	Bus-Timing 0
b_BIT_TIMING1	BYTE	Bus-Timing 1
us_BAUDRATE	USINT 3, 4, 5, 6	Baudrate
us_SYNC_INTERVAL	USINT 8, 4, 2, 1	CANsync-Intervall (Zyklusschema) in ms

Parameter Ausgang	Datentyp	Beschreibung
_BASE	CANsync_PLC_INIT_BMSTRUCT	Initialisierungsdaten für die CANsync-Anschaltung
w_ERR	WORD	Fehlerwort
x_ERR	BOOL	Fehlerbit
x_OK	BOOL	OK-Bit

Der FB CANsync\_INIT bietet mehrere Konfigurierungsmöglichkeiten für die Initialisierung einer CANsync-Anschaltung. Der FB wird verwendet, wenn eine CANsync-Master-Anschaltung initialisiert wird oder wenn eine CANsync-Slave-Anschaltung initialisiert wird.

Ein-/Ausgang \_BASE:

Bei der **Initialisierung der CANsync-Anschaltung auf dem Optionsmodul CANsync-Master für b maXX PLC** wird am Ein-/Ausgang \_BASE, entsprechend dem Steckplatz (Slot G bis M) des Optionsmoduls, die globale Variable \_CANsyncMaster\_Init\_Slot\_G (bis \_CANsyncMaster\_Init\_Slot\_M) angeschlossen.

Ist diese globale Variable nicht im Projekt vorhanden, legen Sie, je nach Steckplatz (Slot G bis M), die globale Variable \_CANsyncMaster\_Init\_Slot\_G (bis \_CANsyncMaster\_Init\_Slot\_M) vom Datentyp CANsync\_PLC\_INIT\_BMSTRUCT an.

Diese Variable muß über die Deklaration der globalen Variablen auf die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul gelegt werden.

Die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Master (BM4-O-CAN-06) in Abhängigkeit vom Steckplatz ist:

Steckplatz (Slot)	Basisadresse der CANsync-Anschaltung
G	%MB3.2001792
H	%MB3.3001792
J	%MB3.4001792
K	%MB3.5001792
L	%MB3.6001792
M	%MB3.7001792

Beispiel:

CANsync-Anschaltung auf Optionsmodul CANsync-Master im Steckplatz G des b maXX

```
_CANsyncMaster_Init_Slot_G AT %MB3.2001792 :
    CANsync_PLC_INIT_BMSTRUCT;
```

dabei ist:

<code>_CANsyncMaster_Init_Slot_G</code>	der Variablenname mit der Datentypkurzbezeichnung "_" für Struct
<code>CANsync_PLC_INIT_BMSTRUCT</code>	der Datentyp
<code>%MB3.2001792</code>	die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Master im Steckplatz G des b maXX

Bei der **Initialisierung der CANsync-Anschaltung auf dem Optionsmodul CANsync-Slave für b maXX PLC** wird am Ein-/Ausgang \_BASE, entsprechend dem Steckplatz (Slot G bis M) des Optionsmoduls, die globale Variable

`_CANsyncSlave_Init_Slot_G` (bis `_CANsyncSlave_Init_Slot_M`) angeschlossen.

Ist diese globale Variable nicht im Projekt vorhanden, legen Sie, je nach Steckplatz (Slot G bis M), die globale Variable `_CANsyncSlave_Init_Slot_G` (bis `_CANsyncSlave_Init_Slot_M`) vom Datentyp `CANsync_PLC_INIT_BMSTRUCT` an.

Diese Variable muß über die Deklaration der globalen Variablen auf die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul gelegt werden.

Die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Slave für b maXX PLC (BM4-O-CAN-05, Dip-Schalter 9 = ON) in Abhängigkeit vom Steckplatz ist:

Steckplatz (Slot)	Basisadresse der CANsync-Anschaltung
G	%MB3.2001792
H	%MB3.3001792
J	%MB3.4001792
K	%MB3.5001792
L	%MB3.6001792
M	%MB3.7001792

Beispiel:

CANsync-Anschaltung auf Optionsmodul CANsync-Slave im Steckplatz G des b maXX

```
_CANsyncSlave_Init_Slot_G AT %MB3.2001792 :
                                CANsync_PLC_INIT_BMSTRUCT;
```

dabei ist:

<code>_CANsyncSlave_Init_Slot_G</code>	der Variablenname mit der Datentypkurzbezeichnung "_" für Struct
<code>CANsync_PLC_INIT_BMSTRUCT</code>	der Datentyp
<code>%MB3.2001792</code>	die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Slave für b maXX PLC im Steckplatz G des b maXX

Eingang `x_SL`:

Am Eingang `x_SL` wird ausgewählt, ob die CANsync-Anschaltung als CANsync-Slave-Anschaltung oder als CANsync-Master-Anschaltung initialisiert wird.

Mit `x_SL = FALSE` wird die CANsync-Anschaltung als CANsync-Master-Anschaltung initialisiert, mit `x_SL = TRUE` wird die CANsync-Anschaltung als CANsync-Slave-Anschaltung initialisiert.

Eingang `x_SYNC_IN`:

Dieser Eingang wird nur belegt, wenn die CANsync-Anschaltung als CANsync-Master-Anschaltung initialisiert wird.

Eine CANsync-Master-Anschaltung kann ein eigenes SYNC-Signal generieren, es kann aber auch eingestellt werden, dass die CANsync-Master-Anschaltung ein externes SYNC-Signal (z. B. von einer CANsync-Slave-Anschaltung im selben Gerät) als SYNC-Signal verwendet.

Wird nur die CANsync-Master-Anschaltung betrieben, muss die CANsync-Master-Anschaltung ein eigenes SYNC-Signal generieren und `x_SYNC_IN` bleibt FALSE.

Mit `x_SYNC_IN = FALSE` generiert die CANsync-Master-Anschaltung ein eigenes SYNC-Signal, mit `x_SYNC_IN = TRUE` übernimmt die CANsync-Master-Anschaltung ein externes SYNC-Signal (z. B. das der CANsync-Slave-Anschaltung). Hierzu sind ggf. weitere Einstellungen erforderlich, die Sie bitte den jeweiligen Technischen Beschreibungen entnehmen.

Voreinstellung ist `x_SYNC_IN = FALSE`, d. h. wenn `x_SYNC_IN` nicht belegt ist, generiert die CANsync-Master-Anschaltung ein eigenes SYNC-Signal.

Eingänge `x_SYNC_MODE`, `x_ASYNC_MODE0`, `x_ASYNC_MODE1`:

An diesen drei Eingängen wird die Betriebsart der CANsync-Anschaltung eingestellt. Es darf nur einer der drei Eingänge gleich TRUE sein.

Mit `x_SYNC_MODE = TRUE` wird der Synchron-Betrieb eingestellt.

Die Eingänge `x_ASYNC_MODE0` und `x_ASYNC_MODE1` sind reserviert und werden nicht belegt.

Erläuterung der Betriebsart siehe [► Allgemeines ◀](#) ab Seite 15.

Wenn alle drei Eingänge gleich FALSE sind, werden nur die Initialisierungsdaten an die CANsync-Anschaltung übergeben und keine Betriebsart eingestellt. Die Betriebsart kann dann mit dem FB `CANsync_MODE_MA` (CANsync-Master-Anschaltung) oder `CANsync_MODE_SL` (CANsync-Slave-Anschaltung) eingestellt werden.

#### HINWEIS



Die Betriebsart wird mit dem FB `CANsync_MODE_MA` bzw. `CANsync_MODE_SL` freigegeben.

Eingang `a_SL_TYP`:

Dieser Eingang wird nur belegt, wenn die CANsync-Anschaltung als CANsync-Master-Anschaltung initialisiert wird. Hier wird angegeben welche CANsync-Slaves am CANsync-Bus angeschlossen sind. Diese Angabe kann auch mit dem FB `CANsync_SL_TYP_INIT` erfolgen.

Am Eingang `a_SL_TYP` wird eine Variable vom Datentyp `BYTE_32_BMARRAY` angeschlossen. Der Datentyp `BYTE_32_BMARRAY` ist ein Feld von 32 Einträgen des Datentyps `Byte`:

```
BYTE_32_BMARRAY : ARRAY [0..31] OF BYTE;
```

Beispiel:

```
a_Slave_Typen : BYTE_32_BMARRAY;
```

dabei ist:

`a_Slave_Typen` der Variablenname mit der Datentypkurzbezeichnung "a" für Array

`BYTE_32_BMARRAY` der Datentyp.

In den einzelnen Einträgen des Feldes wird der Slave-Typ der CANsync-Slaves am CANsync-Bus angegeben. Im Eintrag [0] steht der Slave-Typ des CANsync-Slaves mit der Slavenummer 0, im Eintrag [1] steht der Slave-Typ des CANsync-Slaves mit der Slavenummer 1 usw.

Eine 0 im Eintrag [x] bedeutet, dass kein CANsync-Slave mit der Slavenummer x am CANsync-Bus vorhanden ist.

Ein Wert  $\neq 0$  im Eintrag [x] bedeutet, dass ein CANsync-Slave mit der Slavenummer x am CANsync-Bus vorhanden ist.

Bedeutung der Slave-Typen

Slave-Typ	Bedeutung
0	kein CANsync-Slave mit der Slavenummer x vorhanden
1	b maXX mit Optionsmodul CANsync-Slave mit der Slavenummer x vorhanden
2 bis 255	Reserviert

Eingänge `b_ACCEPT_MASK`, `b_ACCEPT_CODE`:

An den Eingängen `b_ACCEPT_MASK` und `b_ACCEPT_CODE` kann der Akzeptanzfilter der CANsync-Anschaltung eingestellt werden. Werden die Eingänge nicht belegt, ergeben sich die Voreinstellungen `b_ACCEPT_MASK = 16#FF` und `b_ACCEPT_CODE = 16#FF`, d. h. alle Objekte werden berücksichtigt.

Andere Einstellungen werden beim CANsync nicht benötigt.

Diese Eingänge sind aus Kompatibilitätsgründen vorhanden.

Eingänge `b_BIT_TIMING0`, `b_BIT_TIMING1`, `us_BAUDRATE`, `us_SYNC_INTERVAL`:

Am Eingang `us_BAUDRATE` wird die Baudrate für den CANsync-Bus eingestellt. Es darf maximal die Baudrate eingestellt werden, die alle Teilnehmer am CANsync-Bus "verstehen".

### HINWEIS



Einschränkungen der Baudrate entnehmen Sie bitte der jeweiligen Technischen Beschreibung.

Für vier verschiedene Baudrates ist das Bus-Timing berechnet und wird vom FB CANsync\_INIT der CANsync-Anschaltung übergeben.

Baudrate	<code>us_BAUDRATE</code>	<code>b_BIT_TIMING0</code>	<code>b_BIT_TIMING1</code>
125 kBit/s	3	16#47	16#34
250 kBit/s	4	16#43	16#34



500 kBit/s	5	16#41	16#34
1 MBit/s	6	16#40	16#34

Ist der Wert `us_BAUDRATE` kleiner 3 oder größer 6, wird die Baudrate auf 125 kBit/s eingestellt und im Fehlerwort `w_ERR` das Bit 1 auf TRUE gesetzt.

An den Eingängen `b_BIT_TIMING0` und `b_BIT_TIMING1` kann das Bus-Timing der CANsync-Anschaltung individuell eingestellt werden. Die Werte hierfür entnehmen Sie bitte der jeweiligen Technischen Beschreibung.

Die Einstellungen dieser Eingänge werden übernommen, wenn der Eingang `us_BAUDRATE = 0` ist.

Die Einstellungen dieser Eingänge werden ignoriert, wenn der Eingang `us_BAUDRATE` mit einem Wert von 3 bis 6 belegt ist.

Voreinstellung ist `us_BAUDRATE = 0`, d. h. wenn `us_BAUDRATE` nicht belegt ist, werden die Einstellungen von `b_BIT_TIMING0` und `b_BIT_TIMING1` übernommen.

## HINWEIS



Die Werte an den Eingängen `b_BIT_TIMING0` und `b_BIT_TIMING1` werden nur übernommen, wenn der Eingang `us_BAUDRATE = 0` oder nicht belegt ist.

Am Eingang `us_SYNC_INTERVAL` wird die Dauer des CANsync-Intervalls, die CANsync-Zykluszeit, in ms angegeben.

Zusammen mit dem Eingang `us_BAUDRATE` sind die folgenden Kombinationen zulässig:

CANsync-Baudrate und CANsync-Intervalldauer	<code>us_BAUDRATE</code>	<code>us_SYNC_INTERVAL</code>
1 MBit/s und 1 ms	6	1
500 kBit/s und 2 ms	5	2
250 kBit/s und 4 ms	4	4
125 kBit/s und 8 ms	3	8

Ausgang `x_OK`:

Der Ausgang `x_OK` wird auf TRUE gesetzt, wenn die CANsync-Anschaltung erfolgreich initialisiert wurde. Der Ausgang `x_OK` bleibt FALSE, wenn die CANsync-Anschaltung nicht initialisiert wurde oder bei der Initialisierung ein Fehler aufgetreten ist.

Ausgänge `x_ERR`, `w_ERR`:

Falls ein Fehler auftritt wird das Fehlerbit `x_ERR` auf TRUE gesetzt und das Fehlerwort `w_ERR` ausgegeben. Der Ausgang `x_OK` bleibt dann FALSE.

Fehlerwort w\_ERR:

Bit-Nr.	Fehler
0	Timeout beim Handshake mit der CANsync-Anschaltung
1	Eingabefehler bei b_BIT_TIMING0, b_BIT_TIMING1 oder us_BAUDRATE
2 bis 10	Reserviert
11	Initialisierung der CANsync-Anschaltung nicht abgeschlossen
12 bis 15	Reserviert

## 4.10 CANsync\_MODE\_MA

### Beschreibung

Diesen Funktionsbaustein für CANsync können Sie verwenden, um die Betriebsart einer CANsync-Anschaltung einzustellen.



### HINWEIS

Der FB CANsync\_MODE\_MA verwendet die Bibliothek BM\_TYPES\_20bd03 oder höher.

Parameter Eingang	Datentyp	Beschreibung
_BASE	CANsync_PLC_MA_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung
x_RESET_SOFTWARE	BOOL	Software-Reset
x_SET_INIT_DATA	BOOL	Initialisierungsdaten übernehmen
x_CANsync_RUN	BOOL	Aktiven Betrieb freigeben
x_RESET_CANsync_CONTROLLER	BOOL	Reset CANsync-Controller (Bus-Off-Reset)
x_SYNC_MODE	BOOL	Synchron-Betrieb einrichten
x_ASYNC_MODE0	BOOL	reserviert
x_ASYNC_MODE1	BOOL	reserviert

Parameter Ausgang	Datentyp	Beschreibung
_BASE	CANsync_PLC_MA_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung
x_HS_ACTIV	BOOL	Handshake aktiv
x_INIT_POSSIBLE	BOOL	Initialisierung möglich
x_WAIT	BOOL	Warten auf Befehl zur Einstellung der Betriebsart
x_PREPARE_ACTIV	BOOL	Einstellung der Betriebsart aktiv
x_CANsync_ACTIV	BOOL	Betrieb aktiv
x_SYNC_MODE_ACTIV	BOOL	Synchron-Betrieb
x_ASYNC_MODE_ACTIV	BOOL	Asynchron-Betrieb
x_SL	BOOL	Slave-Betrieb
x_MA	BOOL	Master-Betrieb

Der FB CANsync\_MODE\_MA ermöglicht die Einstellung der Betriebsarten auf der CANsync-Anschaltung. Die Eingänge entsprechen Befehlen. Die Ausgänge zeigen den aktuellen Ist-Zustand an. Die Signale sind jeweils aktiv, wenn sie TRUE sind. Wenn alle Eingänge FALSE sind, wird kein Befehl ausgeführt, es bleibt dann der letzte Zustand aktiv.



### HINWEIS

Der FB CANsync\_MODE\_MA wartet nicht auf die Statusmeldung der CANsync-Anschaltung. D. h. wenn der FB in der Kalt- und Warmstart-Task aufgerufen wird, kann es passieren, dass die Ausgänge nicht gesetzt sind.

Falls eine Anzeige des Status benötigt wird, muss der FB noch einmal aufgerufen werden. Die Eingänge müssen dann auf FALSE gesetzt sein. An den Ausgängen wird dann der Status der CANsync-Anschaltung angezeigt.

Einsatz in Kalt- und Warmstart-Task:

Das Starten einer Betriebsart ist möglich. Dazu wird eine Betriebsart eingestellt (z. B. für Synchron-Betrieb wird x\_SYNC\_MODE auf TRUE gesetzt).

Die eingestellte Betriebsart wird mit x\_CANsync\_RUN = TRUE gestartet.

Einsatz im zyklischen Programm:

Die CANsync-Anschaltung kann neu initialisiert werden. Dazu setzt man die CANsync-Anschaltung mit x\_RESET\_SOFTWARE = TRUE zurück (FB CANsync\_MODE\_MA).

Anschließend wird eine neue Initialisierung der CANsync-Anschaltung durchgeführt (FB CANsync\_INIT), eine Betriebsart eingestellt und freigegeben (FB CANsync\_MODE\_MA).

Ein-/Ausgang \_BASE:

Am Ein-/Ausgang \_BASE wird, entsprechend dem Steckplatz (Slot G bis M) des Optionsmoduls, die globale Variable \_CANsyncMaster\_Ctrl\_Slot\_G (bis \_CANsyncMaster\_Ctrl\_Slot\_M) angeschlossen.

Ist diese globale Variable nicht im Projekt vorhanden, legen Sie, je nach Steckplatz (Slot G bis M), die globale Variable \_CANsyncMaster\_Ctrl\_Slot\_G (bis \_CANsyncMaster\_Ctrl\_Slot\_M) vom Datentyp CANsync\_PLC\_MA\_BMSTRUCT an.

Diese Variable muß über die Deklaration der globalen Variablen auf die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul gelegt werden.

Die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Master (BM4-O-CAN-06) in Abhängigkeit vom Steckplatz ist:

Steckplatz (Slot)	Basisadresse der CANsync-Anschaltung
G	%MB3.2001792
H	%MB3.3001792
J	%MB3.4001792

Steckplatz (Slot)	Basisadresse der CANsync-Anschaltung
K	%MB3.5001792
L	%MB3.6001792
M	%MB3.7001792

Beispiel:

CANsync-Anschaltung auf Optionsmodul CANsync-Master im Steckplatz G des b maXX

```
_CANsyncMaster_Ctrl_Slot_G AT %MB3.2001792 :
                                CANsync_PLC_MA_BMSTRUCT;
```

dabei ist:

`_CANsyncMaster_Ctrl_Slot_G` der Variablenname mit der Datentypkurzbezeichnung "\_" für Struct

`CANsync_PLC_MA_BMSTRUCT` der Datentyp

`%MB3.2001792` die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Master im Steckplatz G des b maXX

Eingang `x_RESET_CANsync_CONTROLLER`:

Mit `x_RESET_CANsync_CONTROLLER = TRUE` setzt man den CANsync-Controller zurück. Dadurch verlässt der CANsync-Controller den Bus-Off Zustand und kann wieder am CANsync-Bus aktiv werden.

Eingang `x_SET_INIT_DATA`:

Mit `x_SET_INIT_DATA = TRUE` übernimmt die CANsync-Anschaltung neue Initialisierungsdaten.

Dieser Eingang wird nur bei expliziter Programmierung der Initialisierung ohne den FB `CANsync_INIT` benötigt.

Eingang `x_CANsync_RUN`:

Mit `x_CANsync_RUN = TRUE` aktiviert man die unter `x_SYNC_MODE`, `x_ASYNC_MODE0` oder `x_ASYNC_MODE1` eingestellte Betriebsart.

Eingänge `x_SYNC_MODE`, `x_ASYNC_MODE0`, `x_ASYNC_MODE1`:

An diesen drei Eingängen wird die Betriebsart der CANsync-Anschaltung eingestellt. Es darf nur einer der drei Eingänge gleich `TRUE` sein.

Mit `x_SYNC_MODE = TRUE` wird der Synchron-Betrieb eingestellt.

Die Eingänge `x_ASYNC_MODE0` und `x_ASYNC_MODE1` sind reserviert und bleiben gleich `FALSE`.

Nachdem die Betriebsart eingestellt ist, wird mit `x_CANsync_RUN = TRUE` der aktive Betrieb freigegeben.

An den Ausgängen bekommt man mit TRUE eine entsprechende Rückmeldung für die Befehle. Anderenfalls sind die Ausgänge FALSE.

Ausgang x\_HS\_ACTIV:

Der Ausgang x\_HS\_ACTIV zeigt mit TRUE an, dass der Handshakebetrieb aktiv ist. Dies wird vom FB CANsync\_INIT verwendet.

Ausgang x\_INIT\_POSSIBLE:

Der Ausgang x\_INIT\_POSSIBLE zeigt mit TRUE an, dass die CANsync-Anschaltung im Initialisierungszustand ist. Sie kann dann neue Initialisierungsdaten oder den Befehl zur Einstellung der Betriebsart empfangen.

Ausgang x\_WAIT:

Der Ausgang x\_WAIT zeigt mit TRUE an, dass die CANsync-Anschaltung die Initialisierungsdaten übernommen hat und auf den Befehl zur Einstellung der Betriebsart wartet.

Ausgang x\_PREPARE\_ACTIV:

Der Ausgang x\_PREPARE\_ACTIV zeigt mit TRUE an, dass eine Betriebsart eingerichtet wird.

Ausgang x\_CANsync\_ACTIV:

Der Ausgang x\_CANsync\_ACTIV zeigt mit TRUE an, dass eine Betriebsart aktiv ist.

Ausgang x\_SYNC\_MODE\_ACTIV:

Der Ausgang x\_SYNC\_MODE\_ACTIV zeigt mit TRUE an, dass Synchronbetrieb eingestellt wurde.

Ausgang x\_ASYNC\_MODE\_ACTIV:

Der Ausgang x\_ASYNC\_MODE\_ACTIV zeigt mit TRUE an, dass Asynchronbetrieb (Mode 0 oder Mode 1) eingestellt wurde.

Ausgang x\_SL:

Der Ausgang x\_SL zeigt mit TRUE an, dass die CANsync-Anschaltung als Slave konfiguriert wurde.

Ausgang x\_MA:

Der Ausgang x\_MA zeigt mit TRUE an, dass die CANsync-Anschaltung als Master konfiguriert wurde.

Es können auch Kombinationen von Ausgängen gesetzt sein. Wenn z. B. x\_CANsync\_ACTIV, x\_SYNC\_MODE\_ACTIV und x\_MA auf TRUE gesetzt ist, heißt das, dass die CANsync-Anschaltung als CANsync-Master im aktiven Synchron-Betrieb ist.

## 4.11 CANsync\_MODE\_SL

### Beschreibung

Diesen Funktionsbaustein für CANsync können Sie verwenden, um die Betriebsart einer CANsync-Anschaltung einzustellen.



### HINWEIS

Der FB CANsync\_MODE\_SL verwendet die Bibliothek BM\_TYPES\_20bd03 oder höher.

Parameter Eingang	Datentyp	Beschreibung
_BASE	CANsync_PLC_SL_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung
x_RESET_SOFTWARE	BOOL	Software-Reset
x_SET_INIT_DATA	BOOL	Initialisierungsdaten übernehmen
x_CANsync_RUN	BOOL	Aktiven Betrieb freigeben
x_RESET_CANsync_CONTROLLER	BOOL	Reset CANsync-Controller (Bus-Off-Reset)
x_SYNC_MODE	BOOL	Synchron-Betrieb einrichten
x_ASYNC_MODE0	BOOL	reserviert
x_ASYNC_MODE1	BOOL	reserviert

Parameter Ausgang	Datentyp	Beschreibung
_BASE	CANsync_PLC_SL_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung
x_HS_ACTIV	BOOL	Handshake aktiv
x_INIT_POSSIBLE	BOOL	Initialisierung möglich
x_WAIT	BOOL	Warten auf Befehl zur Einstellung der Betriebsart
x_PREPARE_ACTIV	BOOL	Einstellung der Betriebsart aktiv
x_CANsync_ACTIV	BOOL	Betrieb aktiv
x_SYNC_MODE_ACTIV	BOOL	Synchron-Betrieb
x_ASYNC_MODE_ACTIV	BOOL	Asynchron-Betrieb
x_SL	BOOL	Slave-Betrieb
x_MA	BOOL	Master-Betrieb

Der FB CANsync\_MODE\_SL ermöglicht die Einstellung der Betriebsarten auf der CANsync-Anschaltung. Die Eingänge entsprechen Befehlen. Die Ausgänge zeigen den aktuellen Ist-Zustand an. Die Signale sind jeweils aktiv, wenn sie TRUE sind. Wenn alle Eingänge FALSE sind, wird kein Befehl ausgeführt, es bleibt dann der letzte Zustand aktiv.



### HINWEIS

Der FB CANsync\_MODE\_SL wartet nicht auf die Statusmeldung der CANsync-Anschaltung. D. h. wenn der FB in der Kalt- und Warmstart-Task aufgerufen wird, kann es passieren, dass die Ausgänge nicht gesetzt sind.

Falls eine Anzeige des Status benötigt wird, muss der FB noch einmal aufgerufen werden. Die Eingänge müssen dann auf FALSE gesetzt sein. An den Ausgängen wird dann der Status der CANsync-Anschaltung angezeigt.

Einsatz in Kalt- und Warmstart-Task:

Das Starten einer Betriebsart ist möglich. Dazu wird eine Betriebsart eingestellt (z. B. für Synchron-Betrieb wird x\_SYNC\_MODE auf TRUE gesetzt).

Die eingestellte Betriebsart wird mit x\_CANsync\_RUN = TRUE gestartet.

Einsatz im zyklischen Programm:

Die CANsync-Anschaltung kann neu initialisiert werden. Dazu setzt man die CANsync-Anschaltung mit x\_RESET\_SOFTWARE = TRUE zurück (FB CANsync\_MODE\_SL).

Anschließend wird eine neue Initialisierung der CANsync-Anschaltung durchgeführt (FB CANsync\_INIT) und eine Betriebsart eingestellt und freigegeben (FB CANsync\_MODE\_SL).

Ein-/Ausgang \_BASE:

Am Ein-/Ausgang \_BASE wird, entsprechend dem Steckplatz (Slot G bis M) des Optionsmoduls, die globale Variable \_CANsyncSlave\_Ctrl\_Slot\_G (bis \_CANsyncSlave\_Ctrl\_Slot\_M) angeschlossen.

Ist diese globale Variable nicht im Projekt vorhanden, legen Sie, je nach Steckplatz (Slot G bis M), die globale Variable \_CANsyncSlave\_Ctrl\_Slot\_G (bis \_CANsyncSlave\_Ctrl\_Slot\_M) vom Datentyp CANsync\_PLC\_SL\_BMSTRUCT an.

Diese Variable muß über die Deklaration der globalen Variablen auf die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul gelegt werden.

Die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Slave für b maXX PLC (BM4-O-CAN-05, Dip-Schalter 9 = ON) in Abhängigkeit vom Steckplatz ist:

Steckplatz (Slot)	Basisadresse der CANsync-Anschaltung
G	%MB3.2001792
H	%MB3.3001792
J	%MB3.4001792



Steckplatz (Slot)	Basisadresse der CANsync-Anschaltung
K	%MB3.5001792
L	%MB3.6001792
M	%MB3.7001792

Beispiel:

CANsync-Anschaltung auf Optionsmodul CANsync-Slave im Steckplatz G des b maXX

```
_CANsyncSlave_Ctrl_Slot_G AT %MB3.2001792 :
                                CANsync_PLC_SL_BMSTRUCT;
```

dabei ist:

<code>_CANsyncSlave_Ctrl_Slot_G</code>	der Variablenname mit der Datentypkurzbezeichnung "_" für Struct
<code>CANsync_PLC_SL_BMSTRUCT</code>	der Datentyp
<code>%MB3.2001792</code>	die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Slave im Steckplatz G des b maXX

Eingang `x_RESET_CANsync_CONTROLLER`:

Mit `x_RESET_CANsync_CONTROLLER = TRUE` setzt man den CANsync-Controller zurück. Dadurch verlässt der CANsync-Controller den Bus-Off Zustand und kann wieder am CANsync-Bus aktiv werden.

Eingang `x_SET_INIT_DATA`:

Mit `x_SET_INIT_DATA = TRUE` übernimmt die CANsync-Anschaltung neue Initialisierungsdaten. Dieser Eingang wird nur bei expliziter Programmierung der Initialisierung ohne den FB `CANsync_INIT` benötigt.

Eingang `x_CANsync_RUN`:

Mit `x_CANsync_RUN = TRUE` aktiviert man die unter `x_SYNC_MODE`, `x_ASYNC_MODE0` oder `x_ASYNC_MODE1` eingestellte Betriebsart.

Eingänge `x_SYNC_MODE`, `x_ASYNC_MODE0`, `x_ASYNC_MODE1`:

An diesen drei Eingängen wird die Betriebsart der CANsync-Anschaltung eingestellt. Es darf nur einer der drei Eingänge gleich `TRUE` sein.

Mit `x_SYNC_MODE = TRUE` wird der Synchron-Betrieb eingestellt.

Die Eingänge `x_ASYNC_MODE0` und `x_ASYNC_MODE1` sind reserviert und bleiben gleich `FALSE`.

Nachdem die Betriebsart eingestellt ist, wird mit `x_CANsync_RUN = TRUE` der aktive Betrieb freigegeben.

An den Ausgängen bekommt man mit `TRUE` eine entsprechende Rückmeldung für die Befehle. Anderenfalls sind die Ausgänge `FALSE`.

### Ausgang x\_HS\_ACTIV:

Der Ausgang x\_HS\_ACTIV zeigt mit TRUE an, dass der Handshakebetrieb aktiv ist. Dies wird vom FB CANsync\_INIT verwendet.

### Ausgang x\_INIT\_POSSIBLE:

Der Ausgang x\_INIT\_POSSIBLE zeigt mit TRUE an, dass die CANsync-Anschaltung im Initialisierungszustand ist. Sie kann dann neue Initialisierungsdaten oder den Befehl zur Einstellung der Betriebsart empfangen.

### Ausgang x\_WAIT:

Der Ausgang x\_WAIT zeigt mit TRUE an, dass die CANsync-Anschaltung die Initialisierungsdaten übernommen hat und auf den Befehl zur Einstellung der Betriebsart wartet.

### Ausgang x\_PREPARE\_ACTIV:

Der Ausgang x\_PREPARE\_ACTIV zeigt mit TRUE an, dass eine Betriebsart eingerichtet wird.

### Ausgang x\_CANsync\_ACTIV:

Der Ausgang x\_CANsync\_ACTIV zeigt mit TRUE an, dass eine Betriebsart aktiv ist.

### Ausgang x\_SYNC\_MODE\_ACTIV:

Der Ausgang x\_SYNC\_MODE\_ACTIV zeigt mit TRUE an, dass Synchronbetrieb eingestellt wurde.

### Ausgang x\_ASYNC\_MODE\_ACTIV:

Der Ausgang x\_ASYNC\_MODE\_ACTIV zeigt mit TRUE an, dass Asynchronbetrieb (Mode 0 oder Mode 1) eingestellt wurde.

### Ausgang x\_SL:

Der Ausgang x\_SL zeigt mit TRUE an, dass die CANsync-Anschaltung als Slave konfiguriert wurde.

### Ausgang x\_MA:

Der Ausgang x\_MA zeigt mit TRUE an, dass die CANsync-Anschaltung als Master konfiguriert wurde.

Es können auch Kombinationen von Ausgängen gesetzt sein. Wenn z. B. x\_CANsync\_ACTIV, x\_SYNC\_MODE\_ACTIV und x\_SL auf TRUE gesetzt ist, heißt das, dass die CANsync-Anschaltung als CANsync-Slave im aktiven Synchron-Betrieb ist.

## 4.12 CANsync\_PAR\_READ\_MA

### Beschreibung

Diesen Funktionsbaustein für CANsync können Sie verwenden, um vom CANsync-Master über den CANsync einen Parameterwert vom CANsync-Slave anzufordern.



### HINWEIS

Dieser FB kann mehrfach instanziiert werden, wenn jeweils verschiedene CANsync-Slaves angesprochen werden.

Für den Einsatz dieses FBs ist es notwendig, für den Kommandokanal (CC) die Parameterkommandos freizugeben.

Diese Freigabe erfolgt mit dem FB CANsync\_COMM\_CONTROL\_MA.

Der FB CANsync\_PAR\_READ\_MA verwendet die Bibliothek BM\_TYPES\_20bd03 oder höher.

Parameter Eingang	Datentyp	Beschreibung
_BASE	CANsync_PLC_MA_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung
si_SL_NR	SINT 0 bis 31	Slavenummer des CANsync-Slaves, an den der Parameter-Lesen-Auftrag gerichtet ist
u_PAR_NR	UINT	Parameter-Nummer
x_PAR_FORMAT	BOOL	Parameter-Format
i_SUB_SL	INT 0 bis 31	Sub-Slave-Adresse (reserviert)
t_TIME	TIME	Überwachungszeit
x_EN	BOOL	Freigabe
x_RESET	BOOL	Reset

Parameter Ausgang	Datentyp	Beschreibung
_BASE	CANsync_PLC_MA_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung
ud_PAR_VALUE	UDINT	gelesener Parameter-Wert
x_PAR_FORMAT_READ	BOOL	gelesenes Parameter-Format
x_BUSY	BOOL	Kommunikation ist aktiv
b_ERR	BYTE	Fehlerbyte
i_ERR	INT	Fehlernummer
x_ERR	BOOL	Fehlerbit
x_OK	BOOL	OK-Bit

Der FB CANsync\_PAR\_READ\_MA übergibt mit den Werten der Eingänge u\_PAR\_NR, x\_PAR\_FORMAT und i\_SUB\_SL einen Parameter-Lesen-Auftrag an den CANsync-Slave mit der Slavenummer si\_SL\_NR. Der CANsync-Slave bearbeitet den Parameter-Lesen-Auftrag und gibt das Ergebnis der Kommunikation zurück. Der gelesene Parameter-Wert wird am Ausgang ud\_PAR\_VALUE ausgegeben.

Ein-/Ausgang \_BASE:

Am Ein-/Ausgang \_BASE wird, entsprechend dem Steckplatz (Slot G bis M) des Optionsmoduls, die globale Variable `_CANsyncMaster_Ctrl_Slot_G` (bis `_CANsyncMaster_Ctrl_Slot_M`) angeschlossen.

Ist diese globale Variable nicht im Projekt vorhanden, legen Sie, je nach Steckplatz (Slot G bis M), die globale Variable `_CANsyncMaster_Ctrl_Slot_G` (bis `_CANsyncMaster_Ctrl_Slot_M`) vom Datentyp `CANsync_PLC_MA_BMSTRUCT` an.

Diese Variable muß über die Deklaration der globalen Variablen auf die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul gelegt werden.

Die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Master (BM4-O-CAN-06) in Abhängigkeit vom Steckplatz ist:

Steckplatz (Slot)	Basisadresse der CANsync-Anschaltung
G	%MB3.2001792
H	%MB3.3001792
J	%MB3.4001792
K	%MB3.5001792
L	%MB3.6001792
M	%MB3.7001792

Beispiel:

CANsync-Anschaltung auf Optionsmodul CANsync-Master im Steckplatz G des b maXX

```
_CANsyncMaster_Ctrl_Slot_G AT %MB3.2001792 :
                                CANsync_PLC_MA_BMSTRUCT;
```

dabei ist:

`_CANsyncMaster_Ctrl_Slot_G` der Variablenname mit der Datentypkurzbezeichnung "\_" für Struct

`CANsync_PLC_MA_BMSTRUCT` der Datentyp

`%MB3.2001792` die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Master im Steckplatz G des b maXX

Eingang si\_SL\_NR:

Am Eingang si\_SL\_NR wird die Slavenummer des CANsync-Slaves am CANsync-Bus angegeben, von dem der Parameter-Wert gelesen wird.

**Eingang u\_PAR\_NR:**

Die Parameter-Nummer für den Parameter-Lesen-Auftrag wird am Eingang u\_PAR\_NR angegeben.

**Eingang x\_PAR\_FORMAT:**

Das Format des angeforderten Parameter-Werts wird am Eingang x\_PAR\_FORMAT eingestellt. x\_PAR\_FORMAT = FALSE bedeutet Format Wort, x\_PAR\_FORMAT = TRUE bedeutet Format Doppelwort.

**Eingang i\_SUB\_SL:**

Dieser Eingang ist reserviert und wird nicht belegt.

**Eingang t\_TIME:**

Am Eingang t\_TIME wird die Überwachungszeit angegeben, innerhalb der der Parameter-Lesen-Auftrag ausgeführt sein soll. Wird der Parameter-Lesen-Auftrag nicht innerhalb der Überwachungszeit beendet, wird im Fehlerbyte b\_ERR das Bit 1 auf TRUE gesetzt.

Voreinstellung für t\_TIME ist 3 s.

Ursache für das nicht vollständige Abarbeiten eines Parameter-Lesen-Auftrags kann die Belegung des Kommandokanals mit höherprioreren Nachrichten sein (Broadcastkommandos, Steuerwort-Senden-Aufträge; siehe Technische Beschreibung Optionsmodul CANsync-Master für b maXX PLC, Abschnitt Kommandokanal). Weiterhin kann die Ursache die fehlende Freigabe des Kommandokanals für Parameter-Aufträge sein (siehe FB CANsync\_COMM\_CONTROL\_MA).

**Eingang x\_EN:**

Die Kommunikation wird mit x\_EN = TRUE gestartet. Der Eingang x\_EN darf erst wieder auf FALSE gesetzt werden, wenn der Ausgang x\_BUSY nach Abschluss der Kommunikation auf FALSE fällt. Anderenfalls wird von einem bewussten Abbruch der Kommunikation ausgegangen und der FB muss zurückgesetzt werden (x\_RESET = TRUE).

**Eingang x\_RESET:**

Der FB CANsync\_PAR\_READ\_MA wird mit x\_RESET = TRUE zurückgesetzt. Dies ist z. B. nach einem Abbruch der Kommunikation (durch x\_EN = FALSE) oder nach einer Fehlermeldung notwendig. Anschließend muss x\_RESET wieder auf FALSE gesetzt werden.

**Ausgang ud\_PAR\_VALUE:**

Der gelesene Parameter-Wert wird am Ausgang ud\_PAR\_VALUE zur Verfügung gestellt.

**Ausgang x\_BUSY:**

Der Ausgang x\_BUSY zeigt mit TRUE an, dass die Kommunikation aktiv ist.

Ausgang x\_OK:

Der Ausgang x\_OK wird auf TRUE gesetzt, wenn der Parameter-Lesen-Auftrag korrekt ausgeführt wurde. Der Ausgang x\_OK ist FALSE wenn kein Parameter-Lesen-Auftrag ausgeführt wurde oder der Auftrag nicht korrekt ausgeführt wurde.

Ausgänge x\_ERR, b\_ERR, i\_ERR:

Falls ein Fehler auftritt, wird das Fehlerbit x\_ERR auf TRUE gesetzt und das Fehlerbyte b\_ERR ausgegeben. Der Ausgang x\_OK bleibt dann FALSE.

Wenn vom CANsync-Slave eine Fehlernummer gemeldet wird, wird am Ausgang i\_ERR eine Fehlernummer ausgegeben. Der Inhalt der Fehlernummer wird durch die Applikation im CANsync-Slave bestimmt.

Fehlerbyte b\_ERR:

Bit-Nr.	Fehler
0	Kommunikationsfehler, Fehlernummer steht in i_ERR
1	Timeout
2	Format Doppelwort (TRUE) am Eingang x_PAR_FORMAT erwartet
3	Format Wort (FALSE) am Eingang x_PAR_FORMAT erwartet
4	Ungültige Slavenummer des CANsync-Slaves am CANsync-Bus
5 - 7	Reserviert

### 4.13 CANsync\_PAR\_SL

#### Beschreibung

Diesen Funktionsbaustein für CANsync können Sie verwenden, um einen Parameter-Lesen-Auftrag oder einen Parameter-Schreiben-Auftrag zu erkennen. Der FB ist für den Einsatz mit den BACI-Bedarfsdaten-FBs BACI\_PAR\_READ und BACI\_PAR\_WRITE geeignet.

#### HINWEIS



Der FB CANsync\_PAR\_SL verwendet die Bibliothek BM\_TYPES\_20bd03 oder höher.

Parameter Eingang	Datentyp	Beschreibung
_BASE	CANsync_PLC_SL_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung
ud_PAR_VALUE_READ	UDINT	Parameter-Wert (Lesen)
x_PAR_FORMAT_READ	BOOL	Parameter-Format (Lesen)
i_ERR	INT	Fehlernummer (Applikation)
x_ERR_IN	BOOL	Fehlerbit (Applikation)
x_OK_IN	BOOL	OK-Bit (Applikation)
x_EN	BOOL	Freigabe
x_RESET	BOOL	Reset

Parameter Ausgang	Datentyp	Beschreibung
_BASE	CANsync_PLC_SL_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung
x_RESET_JOB	BOOL	Reset (Auftragswechsel)
x_READ	BOOL	Parameter-Lesen-Auftrag
x_WRITE	BOOL	Parameter-Schreiben-Auftrag
u_PAR_NR	UINT	Parameter-Nummer
x_PAR_FORMAT_WRITE	BOOL	Parameter-Format (Schreiben)
ud_PAR_VALUE_WRITE	UDINT	Parameter-Wert (Schreiben)
x_ACTIV	BOOL	Auftrag gemeldet, warten auf Ergebnis der Applikation
x_BUSY	BOOL	Kommunikation ist aktiv
x_OK	BOOL	OK-Bit

Der FB CANsync\_PAR\_SL erkennt einen Parameter-Lesen-Auftrag oder einen Parameter-Schreiben-Auftrag und stellt die jeweiligen Daten an den Ausgängen zur Verfügung. Die Applikation bearbeitet die Aufträge und übergibt die Ergebnisse dem FB über die Eingänge. Der FB meldet dann die Ergebnisse an die CANsync-Slave-Anschaltung und zeigt an dass die Ergebnisse an den CANsync-Master gesendet wurden.

Bei einem Parameter-Lesen-Auftrag wird mit `x_READ = TRUE` der Auftrag signalisiert und die Parameter-Nummer an `u_PAR_NR` ausgegeben. Von der Applikation wird der Parameter-Wert an `ud_PAR_VALUE_READ` und das Parameter-Format an `x_PAR_FORMAT_READ` erwartet. Mit `x_OK_IN = TRUE` werden Parameter-Wert und Parameter-Format übernommen und an den CANsync-Master gesendet. Wird von der Applikation ein Fehler gemeldet, kann eine Fehlernummer an `i_ERR` angeschlossen und `x_ERR_IN = TRUE` gesetzt werden. In diesem Fall wird die Fehlernummer an den CANsync-Master gesendet.

Bei einem Parameter-Schreiben-Auftrag wird mit `x_WRITE = TRUE` der Auftrag signalisiert, die Parameter-Nummer an `u_PAR_NR`, das Parameter-Format an `x_PAR_FORMAT_WRITE` und der Parameter-Wert an `ud_PAR_VALUE_WRITE` ausgegeben. Von der Applikation wird das Ergebnis der Kommunikation erwartet. Mit `x_OK_IN = TRUE` wird dem CANsync-Master gemeldet, dass der Parameter-Schreiben-Auftrag erfolgreich ausgeführt wurde. Wird von der Applikation ein Fehler gemeldet, kann eine Fehlernummer an `i_ERR` angeschlossen und `x_ERR_IN = TRUE` gesetzt werden. In diesem Fall wird die Fehlernummer an den CANsync-Master gesendet.



### HINWEIS

In der b maXX PLC mit Optionsmodul CANsync-Slave können Parameter-Lesen-Aufträge und Parameter-Schreiben-Aufträge an den Regler weitergeleitet werden. Hierfür werden die FBs der BACI-Bedarfsdatenkommunikation verwendet.

Im folgenden Abschnitt ist bei einigen Ein- und Ausgängen in Klammern der jeweilige Ein- oder Ausgang der FBs der BACI-Bedarfsdatenkommunikation angegeben.

Ein-/Ausgang `_BASE`:

Am Ein-/Ausgang `_BASE` wird, entsprechend dem Steckplatz (Slot G bis M) des Optionsmoduls, die globale Variable `_CANsyncSlave_Ctrl_Slot_G` (bis `_CANsyncSlave_Ctrl_Slot_M`) angeschlossen.

Ist diese globale Variable nicht im Projekt vorhanden, legen Sie, je nach Steckplatz (Slot G bis M), die globale Variable `_CANsyncSlave_Ctrl_Slot_G` (bis `_CANsyncSlave_Ctrl_Slot_M`) vom Datentyp `CANsync_PLC_SL_BMSTRUCT` an.

Diese Variable muß über die Deklaration der globalen Variablen auf die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul gelegt werden.

Die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Slave für b maXX PLC (BM4-O-CAN-05, Dip-Schalter 9 = ON) in Abhängigkeit vom Steckplatz ist:

Steckplatz (Slot)	Basisadresse der CANsync-Anschaltung
G	%MB3.2001792
H	%MB3.3001792
J	%MB3.4001792



Steckplatz (Slot)	Basisadresse der CANsync-Anschaltung
K	%MB3.5001792
L	%MB3.6001792
M	%MB3.7001792

Beispiel:

CANsync-Anschaltung auf Optionsmodul CANsync-Slave im Steckplatz G des b maXX

```
_CANsyncSlave_Ctrl_Slot_G AT %MB3.2001792 :
                                CANsync_PLC_SL_BMSTRUCT;
```

dabei ist:

<code>_CANsyncSlave_Ctrl_Slot_G</code>	der Variablenname mit der Datentypkurzbezeichnung "_" für Struct
<code>CANsync_PLC_SL_BMSTRUCT</code>	der Datentyp
<code>%MB3.2001792</code>	die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Slave im Steckplatz G des b maXX

Eingang `ud_PAR_VALUE_READ`:

An `ud_PAR_VALUE_READ` wird bei einem Parameter-Lesen-Auftrag der Parameter-Wert erwartet. Diesen Parameter-Wert stellt das Applikationsprogramm zur Verfügung. (Der Eingang `ud_PAR_VALUE_READ` kann über den FB `DINT_TO_UDINT` mit FB `BACI_PAR_READ`, Ausgang `di_PAR_VALUE` verbunden sein.)

Eingang `x_PAR_FORMAT_READ`:

An `x_PAR_FORMAT_READ` wird bei einem Parameter-Lesen-Auftrag das Parameter-Format erwartet. Mit `x_PAR_FORMAT_READ = FALSE` wird angegeben, dass der Parameter-Wert an `ud_PAR_VALUE_READ` vom Format Wort (16 Bit) ist, mit `x_PAR_FORMAT_READ = TRUE` wird angegeben, dass der Parameter-Wert an `ud_PAR_VALUE_READ` vom Format Doppelwort (32 Bit) ist.

Das Parameter-Format stellt das Applikationsprogramm zur Verfügung. (Der Eingang `x_PAR_FORMAT_READ` kann mit FB `BACI_PAR_READ`, Ausgang `x_PAR_FORMAT` verbunden sein.)

Eingänge `x_ERR_IN`, `i_ERR`:

Falls das Applikationsprogramm den Parameter-Auftrag des CANsync-Masters nicht bearbeiten oder erfüllen kann, kann an `i_ERR` eine Fehlernummer angegeben und `x_ERR_IN = TRUE` gesetzt werden. In diesem Fall wird die Fehlernummer an den CANsync-Master gesendet.

Die Fehlernummer `i_ERR` und das Fehlerbit `x_ERR_IN` stellt das Applikationsprogramm zur Verfügung. (Der Eingang `x_ERR_IN` kann mit FB `BACI_PAR_WRITE`, Ausgang `x_ERR` und/oder FB `BACI_PAR_READ`, Ausgang `x_ERR` verbunden sein. Der Eingang `i_ERR` kann mit FB `BACI_PAR_WRITE`, Ausgang `i_ERR_COMM` und/oder FB

BACI\_PAR\_READ, Ausgang i\_ERR\_COMM verbunden sein. Siehe ►Abbildung 7◄ auf Seite 145: Nr.1)

Eingang x\_OK\_IN:

Hat das Applikationsprogramm den Parameter-Auftrag des CANsync-Master erfüllt, wird der Eingang x\_OK\_IN = TRUE gesetzt.

Bei einem Parameter-Lesen-Auftrag wird an ud\_PAR\_VALUE\_READ der gelesene Parameter-Wert und an x\_PAR\_FORMAT\_READ das Format des gelesenen Parameters erwartet. Bei einem Parameter-Schreiben-Auftrag wird kein weiterer Wert erwartet.

Das OK-Bit muss vom Applikationsprogramm zur Verfügung gestellt werden. (Der Eingang x\_OK kann mit FB BACI\_PAR\_WRITE, Ausgang x\_OK und/oder FB BACI\_PAR\_READ, Ausgang x\_OK verbunden sein. Siehe ►Abbildung 7◄ auf Seite 145: Nr.2)

Eingang x\_EN:

Der FB CANsync\_PAR\_SL wird mit x\_EN = TRUE aktiviert. Nur wenn der FB aktiviert ist, werden Parameter-Aufträge gemeldet und Antworten an den CANsync-Master gesendet.

Falls der FB CANsync\_PAR\_SL deaktiviert wird (x\_EN = FALSE), muss abgewartet werden bis der letzte Parameter-Auftrag bearbeitet und an den CANsync-Master gesendet wurde (x\_BUSY = FALSE). Anderenfalls wird von einem bewussten Abbruch der Kommunikation ausgegangen und der FB muss dann mit x\_RESET = TRUE zurückgesetzt werden.

Eingang x\_RESET:

Mit x\_RESET = TRUE kann der FB zurückgesetzt werden. Dies ist z. B. nach einem Abbruch der Kommunikation (durch x\_EN = FALSE) oder nach einer Fehlermeldung notwendig. Anschließend muss x\_RESET wieder auf FALSE gesetzt werden.

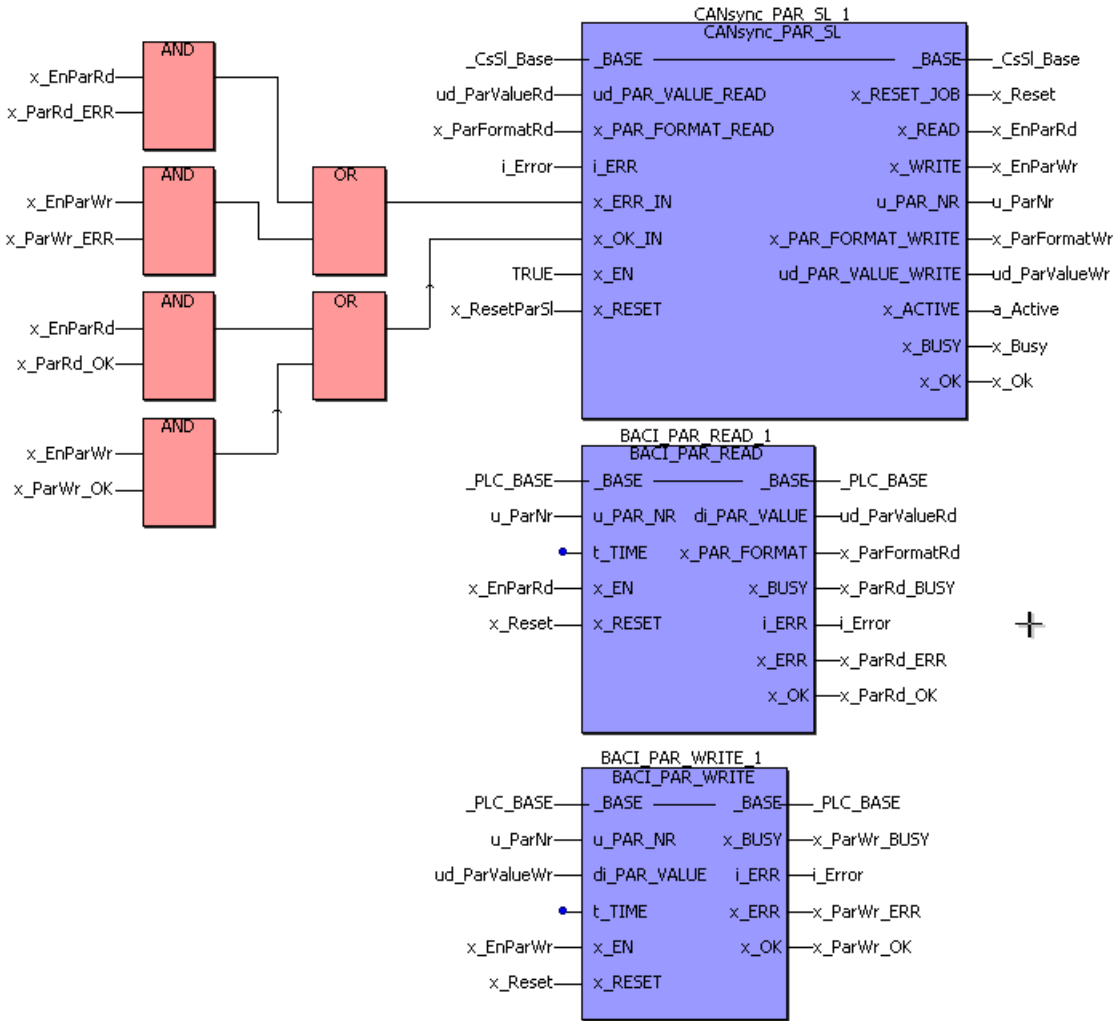


Abbildung 7: Beispiel für die Verwendung des FBs CANsync\_PAR\_SL in Verbindung mit den FBs BACI\_PAR\_READ und BACI\_PAR\_WRITE

**Ausgang x\_RESET\_JOB:**

Der CANSync-Master kann einen Parameter-Auftrag abbrechen. In diesem Fall wird der Ausgang  $x\_RESET\_JOB$  auf TRUE gesetzt. Der Ausgang  $x\_RESET\_JOB$  wird wieder auf FALSE gesetzt, wenn ein neuer Parameter-Auftrag vom CANSync-Master gestartet wird. (Der Ausgang  $x\_RESET\_JOB$  kann mit FB BACI\_PAR\_WRITE, Eingang  $x\_RESET$  und/oder FB BACI\_PAR\_READ, Eingang  $x\_RESET$  verbunden sein.)

**Ausgang x\_READ:**

Der Ausgang  $x\_READ$  wird auf TRUE gesetzt, wenn ein Parameter-Lesen-Auftrag ansteht. (Der Ausgang  $x\_READ$  kann mit FB BACI\_PAR\_READ, Eingang  $x\_EN$  verbunden sein).

Wenn kein Parameter-Lesen-Auftrag ansteht wird der Ausgang  $x\_READ$  auf FALSE gesetzt.

### Ausgang x\_WRITE:

Der Ausgang x\_WRITE wird auf TRUE gesetzt, wenn ein Parameter-Schreiben-Auftrag ansteht. (Der Ausgang x\_WRITE kann mit FB BACI\_PAR\_WRITE, Eingang x\_EN verbunden sein). Wenn kein Parameter-Schreiben-Auftrag ansteht wird der Ausgang x\_WRITE auf FALSE gesetzt.

### Ausgang u\_PAR\_NR:

Am Ausgang u\_PAR\_NR wird die Parameter-Nummer des Parameter-Auftrags ausgegeben. (Der Ausgang u\_PAR\_NR kann mit FB BACI\_PAR\_WRITE, Eingang u\_PAR\_NR oder FB BACI\_PAR\_READ, Eingang u\_PAR\_NR verbunden sein.)

### Ausgang x\_PAR\_FORMAT:

Am Ausgang x\_PAR\_FORMAT wird bei einem Parameter-Schreiben-Auftrag das Parameter-Format des Parameters u\_PAR\_NR ausgegeben.

x\_PAR\_FORMAT = FALSE bedeutet Format Wort (16 Bit), x\_PAR\_FORMAT = TRUE bedeutet Format Doppelwort (32 Bit). (Der Ausgang x\_PAR\_FORMAT hat für den FB BACI\_PAR\_WRITE keine Bedeutung, denn beim Parameter-Schreiben über die BACI auf den Regler wird das im Regler hinterlegte Parameter-Format automatisch erkannt.)

### Ausgang ud\_PAR\_VALUE:

Am Ausgang ud\_PAR\_VALUE wird bei einem Parameter-Schreiben-Auftrag der Parameter-Wert des Parameters u\_PAR\_NR ausgegeben. (Der Ausgang ud\_PAR\_VALUE kann über der FB UDINT\_TO\_DINT mit FB BACI\_PAR\_WRITE, Eingang di\_PAR\_VALUE verbunden sein.)

### Ausgang x\_ACTIV:

Der Ausgang x\_ACTIV zeigt mit TRUE an, dass der FB CANsync\_PAR\_SL während eines Parameter-Auftrags auf das Ergebnis des Parameter-Auftrags wartet (Eingang x\_OK\_IN oder x\_ERR\_IN). Anderenfalls ist der Ausgang x\_ACTIV auf FALSE gesetzt.

### Ausgang x\_BUSY:

Der Ausgang x\_BUSY zeigt mit TRUE an, dass der FB CANsync\_PAR\_SL auf das Ergebnis des Parameter-Schreiben oder -Lesen wartet **und** dass die Antwort an den CANsync-Master fertig ist, jedoch der CANsync-Master die Antwort noch nicht angefordert hat. Anderenfalls ist der Ausgang x\_BUSY auf FALSE gesetzt.

### Ausgang x\_OK:

Der Ausgang x\_OK wird auf TRUE gesetzt, wenn der CANsync-Master die Antwort abgeholt hat. Es ist dabei unerheblich ob es sich um die Fehler-Antwort oder die Antwort für das ordnungsgemäße Abarbeiten des Auftrags handelt. Der Ausgang x\_OK ist FALSE wenn noch kein Parameter-Auftrag ausgeführt wurde, der Parameter-Auftrag nicht ausgeführt wurde oder der CANsync-Master die Antwort nicht abgeholt hat.

#### 4.14 CANsync\_PAR\_WRITE\_MA

##### Beschreibung

Diesen Funktionsbaustein für CANsync können Sie verwenden, um vom CANsync-Master über den CANsync-Bus einen Parameterwert an den CANsync-Slave zu senden.



##### HINWEIS

Dieser FB kann mehrfach instanziiert werden, wenn jeweils verschiedene CANsync-Slaves angesprochen werden.

Für den Einsatz dieses FBs ist es notwendig, für den Kommandokanal (CC) die Parameterkommandos freizugeben.

Diese Freigabe erfolgt mit dem FB CANsync\_COMM\_CONTROL\_MA.

Der FB CANsync\_PAR\_WRITE\_MA verwendet die Bibliothek BM\_TYPES\_20bd03 oder höher.

Parameter Eingang	Datentyp	Beschreibung
_BASE	CANsync_PLC_MA_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung
si_SL_NR	SINT 0 bis 31	Slavenummer des CANsync-Slaves, an den der Parameter-Schreiben-Auftrag gerichtet ist
u_PAR_NR	UINT	Parameter-Nummer
x_PAR_FORMAT	BOOL	Parameter-Format
i_SUB_SL	INT 0 bis 31	Sub-Slave-Adresse (reserviert)
ud_PAR_VALUE	UDINT	zu schreibender Parameter-Wert
t_TIME	TIME	Überwachungszeit
x_EN	BOOL	Freigabe
x_RESET	BOOL	Reset

Parameter Ausgang	Datentyp	Beschreibung
_BASE	CANsync_PLC_MA_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung
x_BUSY	BOOL	Kommunikation ist aktiv
i_ERR	INT	Fehlernummer
b_ERR	BYTE	Fehlerbyte
x_ERR	BOOL	Fehlerbit
x_OK	BOOL	OK-Bit

Der FB CANsync\_PAR\_WRITE\_MA übergibt mit den Werten der Eingänge u\_PAR\_NR, x\_PAR\_FORMAT, i\_SUB\_SL und ud\_PAR\_VALUE einen Parameter-Schreiben-Auftrag an den CANsync-Slave mit der Slavenummer si\_SL\_NR. Der CANsync-Slave bearbeitet den Parameter-Schreiben-Auftrag und gibt das Ergebnis der Kommunikation zurück.

Ein-/Ausgang \_BASE:

Am Ein-/Ausgang \_BASE wird, entsprechend dem Steckplatz (Slot G bis M) des Optionsmoduls, die globale Variable \_CANsyncMaster\_Ctrl\_Slot\_G (bis \_CANsyncMaster\_Ctrl\_Slot\_M) angeschlossen.

Ist diese globale Variable nicht im Projekt vorhanden, legen Sie, je nach Steckplatz (Slot G bis M), die globale Variable \_CANsyncMaster\_Ctrl\_Slot\_G (bis \_CANsyncMaster\_Ctrl\_Slot\_M) vom Datentyp CANsync\_PLC\_MA\_BMSTRUCT an.

Diese Variable muß über die Deklaration der globalen Variablen auf die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul gelegt werden.

Die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Master (BM4-O-CAN-06) in Abhängigkeit vom Steckplatz ist:

Steckplatz (Slot)	Basisadresse der CANsync-Anschaltung
G	%MB3.2001792
H	%MB3.3001792
J	%MB3.4001792
K	%MB3.5001792
L	%MB3.6001792
M	%MB3.7001792

Beispiel:

CANsync-Anschaltung auf Optionsmodul CANsync-Master im Steckplatz G des b maXX

```
_CANsyncMaster_Ctrl_Slot_G AT %MB3.2001792 :
                                CANsync_PLC_MA_BMSTRUCT;
```

dabei ist:

\_CANsyncMaster\_Ctrl\_Slot\_G der Variablenname mit der Datentypkurzbezeichnung "\_" für Struct

CANsync\_PLC\_MA\_BMSTRUCT der Datentyp

%MB3.2001792 die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Master im Steckplatz G des b maXX

Eingang si\_SL\_NR:

Am Eingang si\_SL\_NR wird die Slavenummer des CANsync-Slaves am CANsync-Bus angegeben, an den der Parameter-Wert gesendet werden soll.

**Eingang u\_PAR\_NR:**

Die Parameter-Nummer für den Parameter-Schreiben-Auftrag wird am Eingang u\_PAR\_NR angegeben.

**Eingang x\_PAR\_FORMAT:**

Das Format des zu übertragenden Parameter-Werts wird am Eingang x\_PAR\_FORMAT eingestellt. x\_PAR\_FORMAT = FALSE bedeutet Format Wort, x\_PAR\_FORMAT = TRUE bedeutet Format Doppelwort.

**Eingang i\_SUB\_SL:**

Dieser Eingang ist reserviert und wird nicht belegt.

**Eingang ud\_PAR\_VALUE:**

Der zu übertragende Parameter-Wert wird am Eingang ud\_PAR\_VALUE angegeben.

**Eingang t\_TIME:**

Am Eingang t\_TIME wird die Überwachungszeit angegeben, innerhalb der der Parameter-Schreiben-Auftrag ausgeführt sein soll. Wird der Parameter-Schreiben-Auftrag nicht innerhalb der Überwachungszeit beendet, wird im Fehlerbyte b\_ERR das Bit 1 auf TRUE gesetzt.

Voreinstellung für t\_TIME ist 3 s.

Ursache für das nicht vollständige Abarbeiten eines Parameter-Schreiben-Auftrags kann die Belegung des Kommandokanals mit höherpriorären Nachrichten sein (Broadcastkommandos, Steuerwort-Senden-Aufträge; siehe ►Betriebsanleitung Optionsmodul CANsync-Master für b maXX PLC◄, Abschnitt Kommandokanal). Weiterhin kann die Ursache die fehlende Freigabe des Kommandokanals für Parameter-Aufträge sein (siehe FB ►CANsync\_COMM\_CONTROL\_MA◄ ab Seite 113).

**Eingang x\_EN:**

Die Kommunikation wird mit x\_EN = TRUE gestartet. Der Eingang x\_EN darf erst wieder auf FALSE gesetzt werden, wenn der Ausgang x\_BUSY nach Abschluss der Kommunikation auf FALSE fällt. Anderenfalls wird von einem bewussten Abbruch der Kommunikation ausgegangen und der FB muss zurückgesetzt werden (x\_RESET = TRUE).

**Eingang x\_RESET:**

Der FB CANsync\_PAR\_WRITE\_MA wird mit x\_RESET = TRUE zurückgesetzt. Dies ist z. B. nach einem Abbruch der Kommunikation (durch x\_EN = FALSE) oder nach einer Fehlermeldung notwendig. Anschließend muss x\_RESET wieder auf FALSE gesetzt werden.

**Ausgang x\_BUSY:**

Der Ausgang x\_BUSY zeigt mit TRUE an, dass die Kommunikation aktiv ist. Anderenfalls ist x\_BUSY = FALSE.

Ausgang x\_OK:

Der Ausgang x\_OK wird auf TRUE gesetzt, wenn der Parameter-Schreiben-Auftrag korrekt ausgeführt wurde. Der Ausgang x\_OK ist FALSE wenn kein Parameter-Schreiben-Auftrag ausgeführt wurde oder der Auftrag nicht korrekt ausgeführt wurde.

Ausgänge x\_ERR, b\_ERR, i\_ERR:

Falls ein Fehler auftritt, wird das Fehlerbit x\_ERR auf TRUE gesetzt und das Fehlerbyte b\_ERR ausgegeben. Der Ausgang x\_OK bleibt dann FALSE.

Wenn vom CANsync-Slave eine Fehlernummer gemeldet wird, wird am Ausgang i\_ERR eine Fehlernummer ausgegeben. Der Inhalt der Fehlernummer wird durch die Applikation im CANsync-Slave bestimmt.

Fehlerbyte b\_ERR:

Bit-Nr.	Fehler
0	Kommunikationsfehler, Fehlernummer steht in i_ERR
1	Timeout
2, 3	Reserviert
4	Ungültige Slavenummer des CANsync-Slaves am CANsync-Bus
5 - 7	Reserviert



## 4.15 CANsync\_PD\_CFG\_MA

### Beschreibung

Diesen Funktionsbaustein für CANsync können Sie verwenden, um die Belegung der Sollwert-Telegramme der CANsync-Anschaltung für einen CANsync-Master zu konfigurieren.



### HINWEIS

Der FB CANsync\_PD\_CFG\_MA verwendet die Bibliothek BM\_TYPES\_20bd03 oder höher.

Parameter Eingang	Datentyp	Beschreibung
_BASE	CANsync_PLC_MA_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung
a_WRC1	SINT_4_BMARRAY	Sollwertnummern für Sollwert-Telegramm 1
a_HL_WRC1	BOOL_4_BMARRAY	Zuordnung High- oder Lowword für Sollwert-Telegramm 1
a_WRC2	SINT_4_BMARRAY	Sollwertnummern für Sollwert-Telegramm 2
a_HL_WRC2	BOOL_4_BMARRAY	Zuordnung High- oder Lowword für Sollwert-Telegramm 2

Parameter Ausgang	Datentyp	Beschreibung
_BASE	CANsync_PLC_MA_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung

Mit dem FB CANsync\_PD\_CFG\_MA erfolgt:

- die Zuordnung von 8 Sollwerten á 32 Bit zu den Sollwert-Telegrammen 1 und 2 (Senden)

in einem CANsync-Master.

Die Applikation kann im CANsync-Master 8 Sollwerte á 32 Bit schreiben. Die Sollwerte haben die Sollwertnummern 0 bis 7.

Ein Sollwert setzt sich aus Lowword (Bit 0 bis 15) und Highword (Bit 16 bis 31) zusammen.

In den 2 Sollwert-Telegrammen können (jeweils) 4 \* 16 Bit Daten an CANsync-Slaves gesendet werden. Das sind 4 Worte mit den Wortnummern 0 bis 3.

Mit dem FB CANsync\_PD\_CFG\_MA wird festgelegt, welche Daten in die 4 Worte (Wort 0 bis 3) des Sollwert-Telegramm 1 und welche Daten in die 4 Worte (Wort 0 bis 3) des Sollwert-Telegramm 2 eingetragen werden.

Die Daten für Sollwert-Telegramm 1 können aus den Sollwerten 0 bis 3 gewählt werden, die Daten für Sollwert-Telegramm 2 können aus den Sollwerten 4 bis 7 gewählt werden.

Jedem Wort in einem Sollwert-Telegramm kann ein Low- oder Highword eines Sollwerts zugeordnet werden. Bei der Übertragung eines 32 Bit-Sollwerts werden 2 Worte im Sollwert-Telegramm benötigt.

Ein-/Ausgang `_BASE`:

Am Ein-/Ausgang `_BASE` wird, entsprechend dem Steckplatz (Slot G bis M) des Optionsmoduls, die globale Variable `_CANsyncMaster_Ctrl_Slot_G` (bis `_CANsyncMaster_Ctrl_Slot_M`) angeschlossen.

Ist diese globale Variable nicht im Projekt vorhanden, legen Sie, je nach Steckplatz (Slot G bis M), die globale Variable `_CANsyncMaster_Ctrl_Slot_G` (bis `_CANsyncMaster_Ctrl_Slot_M`) vom Datentyp `CANsync_PLC_MA_BMSTRUCT` an.

Diese Variable muß über die Deklaration der globalen Variablen auf die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul gelegt werden.

Die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Master (BM4-O-CAN-06) in Abhängigkeit vom Steckplatz ist:

Steckplatz (Slot)	Basisadresse der CANsync-Anschaltung
G	%MB3.2001792
H	%MB3.3001792
J	%MB3.4001792
K	%MB3.5001792
L	%MB3.6001792
M	%MB3.7001792

Beispiel:

CANsync-Anschaltung auf Optionsmodul CANsync-Master im Steckplatz G des b maXX

```
_CANsyncMaster_Ctrl_Slot_G AT %MB3.2001792 :
                                CANsync_PLC_MA_BMSTRUCT;
```

dabei ist:

`_CANsyncMaster_Ctrl_Slot_G` der Variablenname mit der Datentypkurzbezeichnung "\_" für Struct

`CANsync_PLC_MA_BMSTRUCT` der Datentyp

`%MB3.2001792` die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Master im Steckplatz G des b maXX

Eingang `a_WRC1`:

Die Zuordnung

"Sollwert (0 bis 3)" -> "Wort im Sollwert-Telegramm 1" erfolgt in `a_WRC1`.

$a\_WRC1[\text{Wortnummer}] := \text{Sollwertnummer}$

Eingang  $a\_HL\_WRC1$ :

Die Zuordnung

"Low- oder Highword des ausgewählten Sollwerts" -> "Wort im Sollwert-Telegramm 1" erfolgt in  $a\_HL\_WRC1$ .

$a\_HL\_WRC1[\text{Wortnummer}] := \text{FALSE}$  (wenn Lowword)

$a\_HL\_WRC1[\text{Wortnummer}] := \text{TRUE}$  (wenn Highword)

Beispiel:

Wortnummer im Sollwert-Telegramm 1	ausgewählter Sollwert	Anschluss an Eingang $a\_WRC1$	Anschluss an Eingang $a\_HL\_WRC1$
0	Sollwert 1 Lowword	$a\_WRC1[0] = \text{SINT\#1}$	$a\_HL\_WRC1[0] = \text{FALSE}$
1	Sollwert 1 Highword	$a\_WRC1[1] = \text{SINT\#1}$	$a\_HL\_WRC1[1] = \text{TRUE}$
2	Sollwert 0 Word	$a\_WRC1[2] = \text{SINT\#0}$	$a\_HL\_WRC1[2] = \text{FALSE}$ oder offen
3	Sollwert 2 Word	$a\_WRC1[3] = \text{SINT\#2}$	$a\_HL\_WRC1[3] = \text{FALSE}$ oder offen

Eingang  $a\_WRC2$ :

Die Zuordnung

"Sollwert (4 bis 7)" -> "Wort im Sollwert-Telegramm 2" erfolgt in  $a\_WRC2$ .

$a\_WRC2[\text{Wortnummer}] := \text{Sollwertnummer}$

Eingang  $a\_HL\_WRC2$ :

Die Zuordnung

"Low- oder Highword des ausgewählten Sollwerts" -> "Wort im Sollwert-Telegramm 2" erfolgt in  $a\_HL\_WRC2$ .

$a\_HL\_WRC2[\text{Wortnummer}] := \text{FALSE}$  (wenn Lowword)

$a\_HL\_WRC2[\text{Wortnummer}] := \text{TRUE}$  (wenn Highword)

Wenn einem Wort im Sollwert-Telegramm 1 oder 2 kein Sollwert zugeordnet werden soll, gibt man am entsprechenden Eintrag in  $a\_WRC1$  bzw.  $a\_WRC2$  als Sollwertnummer eine -1 an.

Die entsprechende Einstellung in  $a\_HL\_WRC1$  bzw.  $a\_HL\_WRC2$  ist in diesem Fall ohne Bedeutung.

$a\_WRC1[\text{Wortnummer}] := \text{SINT\#-1}$

$a\_WRC2[\text{Wortnummer}] := \text{SINT\#-1}$

Beispiel:

Dem Wort 1 des Sollwert-Telegramm 2 soll kein Sollwert zugeordnet werden.

Wortnummer im Sollwert-Telegramm 2	ausgewählter Sollwert	Anschluss an Eingang a_WRC2	Anschluss an Eingang a_HL_WRC2
1	kein	a_WRC2[1] = SINT# -1	a_HL_WRC2[1] ohne Bedeutung

Im CANsync-Slave findet eine Zuordnung der Worte der Sollwert-Telegramme zu Sollwerten statt. Diese Zuordnung erfolgt standardmäßig analog zum CANsync-Master. Eine Überprüfung zwischen der Zuordnung im CANsync-Master und im CANsync-Slave findet nicht statt, weil es auch sinnvolle Anwendungen für abweichende Zuordnungen gibt.

4.16 CANsync\_PD\_CFG\_READ\_MA

**Beschreibung**

Diesen Funktionsbaustein für CANsync können Sie verwenden, um die Belegung der Istwert-Telegramme der CANsync-Slaves in der CANsync-Anschaltung für einen CANsync-Master zu konfigurieren.



**HINWEIS**

Der FB CANsync\_PD\_CFG\_READ\_MA verwendet die Bibliothek BM\_TYPES\_20bd03 oder höher.

Parameter Eingang	Datentyp	Beschreibung
_BASE	CANsync_PLC_MA_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung
si_SL_NR	SINT 0 bis 31	Slavenummer des CANsync-Slaves, von dem Istwerte empfangen werden
a_RDC1	SINT_4_BMARRAY	Istwertnummern für Istwert-Telegramm 1
a_HL_RDC1	BOOL_4_BMARRAY	Zuordnung High- oder Lowword für Istwert-Telegramm 1
a_RDC2	SINT_4_BMARRAY	Istwertnummern für Istwert-Telegramm 2
a_HL_RDC2	BOOL_4_BMARRAY	Zuordnung High- oder Lowword für Istwert-Telegramm 2

Parameter Ausgang	Datentyp	Beschreibung
_BASE	CANsync_PLC_MA_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung

Mit dem FB CANsync\_PD\_CFG\_READ\_MA erfolgt:

- die Zuordnung der Istwert-Telegramme 1 und 2 eines CANsync-Slaves (Empfang) zu 8 Istwerten á 32 Bit

in einem CANsync-Master.

Die Applikation kann im CANsync-Master 8 Istwerte á 32 Bit jedes CANsync-Slaves lesen.

Die CANsync-Slaves haben die Slavenummern 0 bis 31.

Die Istwerte haben die Istwertnummern 0 bis 7.

Ein Istwert setzt sich aus Lowword (Bit 0 bis 15) und Highword (Bit 16 bis 31) zusammen.

Im CANsync-Master können je CANsync-Slave 2 Istwert-Telegramme mit (jeweils) 4 \* 16 Bit Daten empfangen werden. Das sind (jeweils) 4 Worte mit den Wortnummern 0 bis 3.

Mit dem FB CANsync\_PD\_CFG\_READ\_MA wird für einen CANsync-Slave festgelegt, welchen Istwerten die Daten aus den 4 Worten (Wort 0 bis 3) des Istwert-Telegramm 1

und welchen Istwerten die Daten aus den 4 Worten (Wort 0 bis 3) des Istwert-Telegramm 2 des CANsync-Slaves zugeordnet werden.

Die Daten aus dem Istwert-Telegramm 1 können den Istwerten 0 bis 3 zugeordnet werden, die Daten aus dem Istwert-Telegramm 2 können den Istwerten 4 bis 7 zugeordnet werden.

Jedes Wort in einem Istwert-Telegramm kann nur einem Low- oder Highword eines Istwerts zugeordnet werden.

Bei der Übertragung eines 32 Bit-Istwerts werden 2 Worte im Istwert-Telegramm benötigt (ein Wort aus dem Istwert-Telegramm wird dem Lowword eines Istwerts zugeordnet, ein anderes Wort dieses Istwert-Telegramms wird dem Highword dieses Istwerts zugeordnet).

Ein-/Ausgang `_BASE`:

Am Ein-/Ausgang `_BASE` wird, entsprechend dem Steckplatz (Slot G bis M) des Optionsmoduls, die globale Variable `_CANsyncMaster_Ctrl_Slot_G` (bis `_CANsyncMaster_Ctrl_Slot_M`) angeschlossen.

Ist diese globale Variable nicht im Projekt vorhanden, legen Sie, je nach Steckplatz (Slot G bis M), die globale Variable `_CANsyncMaster_Ctrl_Slot_G` (bis `_CANsyncMaster_Ctrl_Slot_M`) vom Datentyp `CANsync_PLC_MA_BMSTRUCT` an.

Diese Variable muß über die Deklaration der globalen Variablen auf die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul gelegt werden.

Die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Master (BM4-O-CAN-06) in Abhängigkeit vom Steckplatz ist:

Steckplatz (Slot)	Basisadresse der CANsync-Anschaltung
G	%MB3.2001792
H	%MB3.3001792
J	%MB3.4001792
K	%MB3.5001792
L	%MB3.6001792
M	%MB3.7001792

Beispiel:

CANsync-Anschaltung auf Optionsmodul CANsync-Master im Steckplatz G des b maXX

```
_CANsyncMaster_Ctrl_Slot_G AT %MB3.2001792 :
                                CANsync_PLC_MA_BMSTRUCT;
```

dabei ist:

`_CANsyncMaster_Ctrl_Slot_G` der Variablenname mit der Datentypkurzbezeichnung "\_" für Struct

`CANsync_PLC_MA_BMSTRUCT` der Datentyp

§MB3.2001792

die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Master im Steckplatz G des b maXX

Eingang si\_SL\_NR:

Am Eingang si\_SL\_NR wird die Slavenummer des CANsync-Slaves am CANsync-Bus angegeben, dessen Istwert-Telegramme konfiguriert werden.

Eingang a\_RDC1:

Die Zuordnung

"Wort im Istwert-Telegramm 1" -> "Istwert (0 bis 3)" erfolgt in a\_RDC1.

a\_RDC1[Wortnummer] := Istwertnummer

Eingang a\_HL\_RDC1:

Die Zuordnung

"Wort im Istwert-Telegramm 1" -> "Low- oder Highword des ausgewählten Istwerts" erfolgt in a\_HL\_RDC1.

a\_HL\_RDC1[Wortnummer] := FALSE (wenn Lowword)

a\_HL\_RDC1[Wortnummer] := TRUE (wenn Highword)

Beispiel:

Wortnummer im Istwert-Telegramm 1	ausgewählter Istwert	Anschluss an Eingang a_RDC1	Anschluss an Eingang a_HL_RDC1
0	Istwert 1 Lowword	a_RDC1[0] = SINT#1	a_HL_RDC1[0] = FALSE
1	Istwert 1 Highword	a_RDC1[1] = SINT#1	a_HL_RDC1[1] = TRUE
2	Istwert 0 Word	a_RDC1[2] = SINT#0	a_HL_RDC1[2] = FALSE
3	Istwert 2 Word	a_RDC1[3] = SINT#2	a_HL_RDC1[3] = FALSE

Eingang a\_RDC2:

Die Zuordnung

"Wort im Istwert-Telegramm 2" -> "Istwert (4 bis 7)" erfolgt in a\_RDC2.

a\_RDC2[Wortnummer] := Istwertnummer

Eingang a\_HL\_RDC2:

Die Zuordnung

"Wort im Istwert-Telegramm 2" -> "Low- oder Highword des ausgewählten Istwerts" erfolgt in a\_HL\_RDC2.

a\_HL\_RDC2[Wortnummer] := FALSE (wenn Lowword)

a\_HL\_RDC2[Wortnummer] := TRUE (wenn Highword)

Wenn einem Istwert kein Wort aus dem Istwert-Telegramm 1 oder 2 zugeordnet werden soll, gibt man am entsprechenden Eintrag in a\_RDC1 bzw. a\_RDC2 als Istwertnummer eine -1 an. Die entsprechende Einstellung in a\_HL\_RDC1 bzw. a\_HL\_RDC2 ist in diesem Fall ohne Bedeutung.

a\_RDC1[Wortnummer] := SINT# -1

a\_RDC2[Wortnummer] := SINT# -1

Beispiel:

Das Wort 1 des Istwert-Telegramm 2 soll keinem Istwert zugeordnet werden.

Wortnummer im Istwert-Telegramm 2	ausgewählter Istwert	Anschluss an Eingang a_RDC2	Anschluss an Eingang a_HL_RDC2
1	kein	a_RDC2[1] = SINT# -1	a_HL_RDC2[1] ohne Bedeutung

Im CANsync-Slave findet eine Zuordnung von Istwerten zu den Worten der Istwert-Telegramme statt. Diese Zuordnung erfolgt standardmäßig analog zum CANsync-Master. Eine Überprüfung zwischen der Zuordnung im CANsync-Master und im CANsync-Slave findet nicht statt, weil es auch sinnvolle Anwendungen für abweichende Zuordnungen gibt.



4.17 CANsync\_PD\_CFG\_READ\_SL

**Beschreibung**

Diesen Funktionsbaustein für CANsync können Sie verwenden, um die Belegung der Istwert-Telegramme der CANsync-Slaves in der CANsync-Anschaltung für einen CANsync-Slave zu konfigurieren.



**HINWEIS**

Der FB CANsync\_PD\_CFG\_READ\_SL verwendet die Bibliothek BM\_TYPES\_20bd03 oder höher.

Parameter Eingang	Datentyp	Beschreibung
_BASE	CANsync_PLG_SL_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung
si_SL_NR	SINT 0 bis 31	Slavenummer des CANsync-Slaves, von dem Istwerte empfangen werden
a_RDC1	SINT_4_BMARRAY	Istwertnummern für Istwert-Telegramm 1
a_HL_RDC1	BOOL_4_BMARRAY	Zuordnung High- oder Lowword für Istwert-Telegramm 1
a_RDC2	SINT_4_BMARRAY	Istwertnummern für Istwert-Telegramm 2
a_HL_RDC2	BOOL_4_BMARRAY	Zuordnung High- oder Lowword für Istwert-Telegramm 2

Parameter Ausgang	Datentyp	Beschreibung
_BASE	CANsync_PLG_SL_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung

Mit dem FB CANsync\_PD\_CFG\_READ\_SL erfolgt:

- die Zuordnung der Istwert-Telegramme 1 und 2 eines anderen CANsync-Slaves (Empfang) zu 8 Istwerten á 32 Bit

in einem CANsync-Slave.

Jeder CANsync-Slave kann die Istwert-Telegramme der anderen CANsync-Slaves mitlesen und die Daten in den Istwert-Telegrammen verwenden.

Die Applikation kann im CANsync-Slave 8 Istwerte á 32 Bit jedes anderen CANsync-Slaves am CANsync-Bus lesen.

Die CANsync-Slaves haben die Slavenummern 0 bis 31.

Die Istwerte haben die Istwertnummern 0 bis 7.

Ein Istwert setzt sich aus Lowword (Bit 0 bis 15) und Highword (Bit 16 bis 31) zusammen.

Im CANsync-Slave können die 2 Istwert-Telegramme jedes anderen CANsync-Slaves mit je 4 \* 16 Bit Daten empfangen werden. Das sind je 4 Worte mit den Wortnummern 0 bis 3.

Mit dem FB CANsync\_PD\_CFG\_READ\_SL wird für einen anderen CANsync-Slave festgelegt, welchen Istwerten die Daten aus den 4 Worten (Wort 0 bis 3) des Istwert-Telegramm 1 und welchen Istwerten die Daten aus den 4 Worten (Wort 0 bis 3) des Istwert-Telegramm 2 des anderen CANsync-Slaves zugeordnet werden.

Die Daten aus dem Istwert-Telegramm 1 können den Istwerten 0 bis 3 zugeordnet werden, die Daten aus dem Istwert-Telegramm 2 können den Istwerten 4 bis 7 zugeordnet werden.

Jedes Wort in einem Istwert-Telegramm kann nur einem Low- oder Highword eines Istwerts zugeordnet werden.

Bei der Übertragung eines 32 Bit-Istwerts werden 2 Worte im Istwert-Telegramm benötigt (ein Wort aus dem Istwert-Telegramm wird dem Lowword eines Istwerts zugeordnet, ein anderes Wort dieses Istwert-Telegramms wird dem Highword dieses Istwerts zugeordnet).

Ein-/Ausgang \_BASE:

Am Ein-/Ausgang \_BASE wird, entsprechend dem Steckplatz (Slot G bis M) des Optionsmoduls, die globale Variable `_CANsyncSlave_Ctrl_Slot_G` (bis `_CANsyncSlave_Ctrl_Slot_M`) angeschlossen.

Ist diese globale Variable nicht im Projekt vorhanden, legen Sie, je nach Steckplatz (Slot G bis M), die globale Variable `_CANsyncSlave_Ctrl_Slot_G` (bis `_CANsyncSlave_Ctrl_Slot_M`) vom Datentyp `CANsync_PLC_SL_BMSTRUCT` an.

Diese Variable muß über die Deklaration der globalen Variablen auf die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul gelegt werden.

Die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Slave für b maXX PLC (BM4-O-CAN-05, Dip-Schalter 9 = ON) in Abhängigkeit vom Steckplatz ist:

Steckplatz (Slot)	Basisadresse der CANsync-Anschaltung
G	%MB3.2001792
H	%MB3.3001792
J	%MB3.4001792
K	%MB3.5001792
L	%MB3.6001792
M	%MB3.7001792

Beispiel:

CANsync-Anschaltung auf Optionsmodul CANsync-Slave im Steckplatz G des b maXX

```
_CANsyncSlave_Ctrl_Slot_G AT %MB3.2001792 :
                                CANsync_PLC_SL_BMSTRUCT;
```

dabei ist:

<code>_CANsyncSlave_Ctrl_Slot_G</code>	der Variablenname mit der Datentypkurzbezeichnung "_" für Struct
<code>CANsync_PLC_SL_BMSTRUCT</code>	der Datentyp
<code>%MB3.2001792</code>	die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Slave im Steckplatz G des b maXX

Eingang `si_SL_NR`:

Am Eingang `si_SL_NR` wird die Slavenummer des CANsync-Slaves am CANsync-Bus angegeben, dessen Istwert-Telegramme konfiguriert werden.

Eingang `a_RDC1`:

Die Zuordnung

"Wort im Istwert-Telegramm 1" -> "Istwert (0 bis 3)" erfolgt in `a_RDC1`.  
`a_RDC1[Wortnummer] := Istwertnummer`

Eingang `a_HL_RDC1`:

Die Zuordnung

"Wort im Istwert-Telegramm 1" -> "Low- oder Highword des ausgewählten Istwerts" erfolgt in `a_HL_RDC1`.  
`a_HL_RDC1[Wortnummer] := FALSE` (wenn Lowword)  
`a_HL_RDC1[Wortnummer] := TRUE` (wenn Highword)

Beispiel:

Wortnummer im Istwert-Telegramm 1	ausgewählter Istwert	Anschluss an Eingang <code>a_RDC1</code>	Anschluss an Eingang <code>a_HL_RDC1</code>
0	Istwert 1 Lowword	<code>a_RDC1[0] = SINT#1</code>	<code>a_HL_RDC1[0] = FALSE</code>
1	Istwert 1 Highword	<code>a_RDC1[1] = SINT#1</code>	<code>a_HL_RDC1[1] = TRUE</code>
2	Istwert 0 Word	<code>a_RDC1[2] = SINT#0</code>	<code>a_HL_RDC1[2] = FALSE</code>
3	Istwert 2 Word	<code>a_RDC1[3] = SINT#2</code>	<code>a_HL_RDC1[3] = FALSE</code>

Eingang `a_RDC2`:

Die Zuordnung

"Wort im Istwert-Telegramm 2" -> "Istwert (4 bis 7)" erfolgt in `a_RDC2`.  
`a_RDC2[Wortnummer] := Istwertnummer`

Eingang a\_HL\_RDC2:

Die Zuordnung

"Wort im Istwert-Telegramm 2" -> "Low- oder Highword des ausgewählten Istwerts" erfolgt in a\_HL\_RDC2.

a\_HL\_RDC2[Wortnummer] := FALSE (wenn Lowword)

a\_HL\_RDC2[Wortnummer] := TRUE (wenn Highword)

Wenn einem Istwert kein Wort aus dem Istwert-Telegramm 1 oder 2 zugeordnet werden soll, gibt man am entsprechenden Eintrag in a\_RDC1 bzw. a\_RDC2 als Istwertnummer eine -1 an. Die entsprechende Einstellung in a\_HL\_RDC1 bzw. a\_HL\_RDC2 ist in diesem Fall ohne Bedeutung.

a\_RDC1[Wortnummer] := SINT# -1

a\_RDC2[Wortnummer] := SINT# -1

Beispiel:

Das Wort 1 des Istwert-Telegramm 2 soll keinem Istwert zugeordnet werden.

Wortnummer im Istwert-Telegramm 2	ausgewählter Istwert	Anschluss an Eingang a_RDC2	Anschluss an Eingang a_HL_RDC2
1	kein	a_RDC2[1] = SINT# -1	a_HL_RDC2[1] ohne Bedeutung

Im anderen CANsync-Slave findet eine Zuordnung von Istwerten zu den Worten der Istwert-Telegramme statt. Diese Zuordnung erfolgt standardmäßig analog zum (empfangenden) CANsync-Slave. Eine Überprüfung zwischen der Zuordnung im (empfangenden) CANsync-Slave und im anderen CANsync-Slave findet nicht statt, weil es auch sinnvolle Anwendungen für abweichende Zuordnungen gibt.

## 4.18 CANsync\_PD\_CFG\_SL

### Beschreibung

Diesen Funktionsbaustein für CANsync können Sie verwenden, um die Belegung der Sollwert- und der Istwert-Telegramme der CANsync-Anschaltung für einen CANsync-Slave zu konfigurieren.



### HINWEIS

Der FB CANsync\_PD\_CFG\_SL verwendet die Bibliothek BM\_TYPES\_20bd03 oder höher.

Parameter Eingang	Datentyp	Beschreibung
_BASE	CANsync_PLC_SL_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung
a_WRC1	SINT_4_BMARRAY	Sollwertnummern für Sollwert-Telegramm 1
a_HL_WRC1	BOOL_4_BMARRAY	Zuordnung High- oder Lowword für Sollwert-Telegramm 1
a_WRC2	SINT_4_BMARRAY	Sollwertnummern für Sollwert-Telegramm 2
a_HL_WRC2	BOOL_4_BMARRAY	Zuordnung High- oder Lowword für Sollwert-Telegramm 2
a_RDC1	SINT_4_BMARRAY	Istwertnummern für Istwert-Telegramm 1
a_HL_RDC1	BOOL_4_BMARRAY	Zuordnung High- oder Lowword für Istwert-Telegramm 1
a_RDC2	SINT_4_BMARRAY	Istwertnummern für Istwert-Telegramm 2
a_HL_RDC2	BOOL_4_BMARRAY	Zuordnung High- oder Lowword für Istwert-Telegramm 2

Parameter Ausgang	Datentyp	Beschreibung
_BASE	CANsync_PLC_SL_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung

Mit dem FB CANsync\_PD\_CFG\_SL erfolgt:

- die Zuordnung der Sollwert-Telegramme 1 und 2 (Empfang) zu 8 Sollwerten á 32 Bit

und

- die Zuordnung von 8 Istwerten á 32 Bit zu den Istwert-Telegrammen 1 und 2 (Senden)

in einem CANsync-Slave.

Sollwerte empfangen:

Die Applikation kann im CANsync-Slave 8 Sollwerte á 32 Bit lesen. Die Sollwerte haben die Sollwertnummern 0 bis 7.

Ein Sollwert setzt sich aus Lowword (Bit 0 bis 15) und Highword (Bit 16 bis 31) zusammen.

Im CANsync-Slave können 2 Sollwert-Telegramme mit je 4 \* 16 Bit Daten empfangen werden. Das sind je 4 Worte mit den Wortnummern 0 bis 3.

Mit dem FB CANsync\_PD\_CFG\_SL wird festgelegt, welchen Sollwerten die Daten aus den 4 Worten (Wort 0 bis 3) des Sollwert-Telegramm 1 und welchen Sollwerten die Daten aus den 4 Worten (Wort 0 bis 3) des Sollwert-Telegramm 2 zugeordnet werden.

Die Daten aus dem Sollwert-Telegramm 1 können den Sollwerten 0 bis 3 zugeordnet werden, die Daten aus dem Sollwert-Telegramm 2 können den Sollwerten 4 bis 7 zugeordnet werden.

Jedes Wort in einem Sollwert-Telegramm kann nur einem Low- oder Highword eines Sollwerts zugeordnet werden.

Bei der Übertragung eines 32 Bit-Sollwerts werden 2 Worte im Sollwert-Telegramm benötigt (ein Wort aus dem Sollwert-Telegramm wird dem Lowword eines Sollwerts zugeordnet, ein anderes Wort dieses Sollwert-Telegramms wird dem Highword dieses Sollwerts zugeordnet).

Istwerte senden:

Weiterhin kann die Applikation im CANsync-Slave 8 Istwerte á 32 Bit schreiben. Die Istwerte haben die Istwertnummern 0 bis 7.

Ein Istwert setzt sich aus Lowword (Bit 0 bis 15) und Highword (Bit 16 bis 31) zusammen.

In den 2 Istwert-Telegrammen können je 4 \* 16 Bit Daten an den CANsync-Master gesendet werden. Das sind 4 Worte mit den Wortnummern 0 bis 3.

Mit dem FB CANsync\_PD\_CFG\_SL wird festgelegt, welche Daten in die 4 Worte (Wort 0 bis 3) des Istwert-Telegramm 1 und welche Daten in die 4 Worte (Wort 0 bis 3) des Istwert-Telegramm 2 eingetragen werden.

Die Daten für Istwert-Telegramm 1 können aus den Istwerten 0 bis 3 gewählt werden, die Daten für Istwert-Telegramm 2 können aus den Istwerten 4 bis 7 gewählt werden.

Jedem Wort in einem Istwert-Telegramm kann ein Low- oder Highword eines Istwerts zugeordnet werden. Bei der Übertragung eines 32 Bit-Istwerts werden 2 Worte im Istwert-Telegramm benötigt.

Ein-/Ausgang \_BASE:

Am Ein-/Ausgang \_BASE wird, entsprechend dem Steckplatz (Slot G bis M) des Optionsmoduls, die globale Variable `_CANsyncSlave_Ctrl_Slot_G` (bis `_CANsyncSlave_Ctrl_Slot_M`) angeschlossen.

Ist diese globale Variable nicht im Projekt vorhanden, legen Sie, je nach Steckplatz (Slot G bis M), die globale Variable `_CANsyncSlave_Ctrl_Slot_G` (bis `_CANsyncSlave_Ctrl_Slot_M`) vom Datentyp `CANsync_PLC_SL_BMSTRUCT` an.

Diese Variable muß über die Deklaration der globalen Variablen auf die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul gelegt werden.

Die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Slave für b maXX PLC (BM4-O-CAN-05, Dip-Schalter 9 = ON) in Abhängigkeit vom Steckplatz ist:

Steckplatz (Slot)	Basisadresse der CANsync-Anschaltung
G	%MB3.2001792
H	%MB3.3001792
J	%MB3.4001792
K	%MB3.5001792
L	%MB3.6001792
M	%MB3.7001792

Beispiel:

CANsync-Anschaltung auf Optionsmodul CANsync-Slave im Steckplatz G des b maXX

```
_CANsyncSlave_Ctrl_Slot_G AT %MB3.2001792 :
                                CANsync_PLC_SL_BMSTRUCT;
```

dabei ist:

<code>_CANsyncSlave_Ctrl_Slot_G</code>	der Variablenname mit der Datentypkurzbezeichnung "_" für Struct
<code>CANsync_PLC_SL_BMSTRUCT</code>	der Datentyp
<code>%MB3.2001792</code>	die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Slave im Steckplatz G des b maXX

Eingang a\_WRC1:

Die Zuordnung

"Wort im Sollwert-Telegramm 1" -> "Sollwert (0 bis 3)" erfolgt in a\_WRC1.

`a_WRC1[Wortnummer] := Sollwertnummer`

Eingang a\_HL\_WRC1:

Die Zuordnung

"Wort im Sollwert-Telegramm 1" -> "Low- oder Highword des ausgewählten Sollwerts" erfolgt in a\_HL\_WRC1.

`a_HL_WRC1[Wortnummer] := FALSE` (wenn Lowword)

`a_HL_WRC1[Wortnummer] := TRUE` (wenn Highword)

Beispiel:

Wortnummer im Sollwert-Telegramm 1	ausgewählter Sollwert	Anschluss an Eingang a_WRC1	Anschluss an Eingang a_HL_WRC1
0	Sollwert 1 Lowword	a_WRC1[0] = SINT#1	a_HL_WRC1[0] = FALSE
1	Sollwert 1 Highword	a_WRC1[1] = SINT#1	a_HL_WRC1[1] = TRUE
2	Sollwert 0 Word	a_WRC1[2] = SINT#0	a_HL_WRC1[2] = FALSE
3	Sollwert 2 Word	a_WRC1[3] = SINT#2	a_HL_WRC1[3] = FALSE

Eingang a\_WRC2:

Die Zuordnung

"Wort im Sollwert-Telegramm 2" -> "Sollwert (4 bis 7)" erfolgt in a\_WRC2.

a\_WRC2[Wortnummer] := Sollwertnummer

Eingang a\_HL\_WRC2:

Die Zuordnung

"Wort im Sollwert-Telegramm 2" -> "Low- oder Highword des ausgewählten Sollwerts" erfolgt in a\_HL\_WRC2.

a\_HL\_WRC2[Wortnummer] := FALSE (wenn Lowword)

a\_HL\_WRC2[Wortnummer] := TRUE (wenn Highword)

Wenn einem Sollwert kein Wort aus dem Sollwert-Telegramm 1 oder 2 zugeordnet werden soll, gibt man am entsprechenden Eintrag in a\_WRC1 bzw. a\_WRC2 als Sollwertnummer eine -1 an.

Die entsprechende Einstellung in a\_HL\_WRC1 bzw. a\_HL\_WRC2 ist in diesem Fall ohne Bedeutung.

a\_WRC1[Wortnummer] := SINT# -1

a\_WRC2[Wortnummer] := SINT# -1

Beispiel:

Das Wort 1 des Sollwert-Telegramm 2 soll keinem Sollwert zugeordnet werden.

Wortnummer im Sollwert-Telegramm 2	ausgewählter Sollwert	Anschluss an Eingang a_WRC2	Anschluss an Eingang a_HL_WRC2
1	kein	a_WRC2[1] = SINT# -1	a_HL_WRC2[1] ohne Bedeutung

Im CANsync-Master findet eine Zuordnung von Sollwerten zu den Worten der Sollwert-Telegramme statt. Diese Zuordnung erfolgt standardmäßig analog zum CANsync-Slave. Eine Überprüfung zwischen der Zuordnung im CANsync-Master und im CANsync-Slave findet nicht statt, weil es auch sinnvolle Anwendungen für abweichende Zuordnungen gibt.



Eingang a\_RDC1:

Die Zuordnung

"Istwert (0 bis 3)" -> "Wort im Istwert-Telegramm 1" erfolgt in a\_RDC1.

a\_RDC1[Wortnummer] := Istwertnummer

Eingang a\_HL\_RDC1:

Die Zuordnung

"Low- oder Highword des ausgewählten Istwerts" -> "Wort im Istwert-Telegramm 1" erfolgt in a\_HL\_RDC1.

a\_HL\_RDC1[Wortnummer] := FALSE (wenn Lowword)

a\_HL\_RDC1[Wortnummer] := TRUE (wenn Highword)

Beispiel:

Wortnummer im Istwert-Telegramm 1	ausgewählter Istwert	Anschluss an Eingang a_RDC1	Anschluss an Eingang a_HL_RDC1
0	Istwert 1 Lowword	a_RDC1[0] = SINT#1	a_HL_RDC1[0] = FALSE
1	Istwert 1 Highword	a_RDC1[1] = SINT#1	a_HL_RDC1[1] = TRUE
2	Istwert 0 Word	a_RDC1[2] = SINT#0	a_HL_RDC1[2] = FALSE
3	Istwert 2 Word	a_RDC1[3] = SINT#2	a_HL_RDC1[3] = FALSE

Eingang a\_RDC2:

Die Zuordnung

"Istwert (4 bis 7)" -> "Wort im Istwert-Telegramm 2" erfolgt in a\_RDC2.

a\_RDC2[Wortnummer] := Istwertnummer

Eingang a\_HL\_RDC2:

Die Zuordnung

"Low- oder Highword des ausgewählten Istwerts" -> "Wort im Istwert-Telegramm 2" erfolgt in a\_HL\_RDC2.

a\_HL\_RDC2[Wortnummer] := FALSE (wenn Lowword)

a\_HL\_RDC2[Wortnummer] := TRUE (wenn Highword)

Wenn einem Wort im Istwert-Telegramm 1 oder 2 kein Istwert zugeordnet werden soll, gibt man am entsprechenden Eintrag in a\_RDC1 bzw. a\_RDC2 als Istwertnummer eine -1 an. Die entsprechende Einstellung in a\_HL\_RDC1 bzw. a\_HL\_RDC2 ist in diesem Fall ohne Bedeutung.

a\_RDC1[Wortnummer] := SINT# -1

a\_RDC2[Wortnummer] := SINT# -1

Beispiel:

Dem Wort 1 des Istwert-Telegramm 2 soll kein Istwert zugeordnet werden.

Wortnummer im Istwert-Telegramm 2	ausgewählter Istwert	Anschluss an Eingang a_RDC2	Anschluss an Eingang a_HL_RDC2
1	kein	a_RDC2[1] = SINT# -1	a_HL_RDC2[1] ohne Bedeutung

Im CANsync-Master findet eine Zuordnung der Worte der Istwert-Telegramme zu Istwerten statt. Diese Zuordnung erfolgt standardmäßig analog zum CANsync-Slave. Eine Überprüfung zwischen der Zuordnung im CANsync-Master und im CANsync-Slave findet nicht statt, weil es auch sinnvolle Anwendungen für abweichende Zuordnungen gibt.

4.19 CANsync\_PD\_COMM\_MA

**Beschreibung**

Diesen Funktionsbaustein für CANsync können Sie verwenden, um die Prozessdatenkommunikation (Sollwerte und Istwerte) der CANsync-Master-Anschaltung durchzuführen.

**HINWEIS**



Der FB CANsync\_PD\_COMM\_MA verwendet die Bibliothek BM\_TYPES\_20bd03 oder höher.

Parameter Eingang	Datentyp	Beschreibung
_BASE	CANsync_PLC_MA_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung
a_RD_ARRAY	CANsync_RD_BMARRAY	Ausgabearray für Istwerte der CANsync-Slaves
a_WR_VALUES_SEND	DINT_8_BMARRAY	Sollwerte, die gesendet werden sollen
si_WRC1_SEND	SINT 5	Kommando, dass Sollwert-Telegramm 1 gesendet werden soll
si_WRC2_SEND	SINT 5	Kommando, dass Sollwert-Telegramm 2 gesendet werden soll
si_RD_SL_NR1_RECEIVE	SINT -1, 0 bis 31	Slavenummer des CANsync-Slaves, von dem Istwert-Telegramm 1 angefordert werden soll
si_RD_SL_NR2_RECEIVE	SINT -1, 0 bis 31	Slavenummer des CANsync-Slaves, von dem Istwert-Telegramm 2 angefordert werden soll
si_MAX_SL_NR	SINT 0 bis 31	maximale Slavenummer für automatische Erhöhung <sup>a</sup>
x_COPY_TO_RD_ARRAY	BOOL	Angabe, ob Istwerte in das RD_ARRAY kopiert werden sollen

a) Dieser Eingang entspricht dem Eingang si\_MAX\_SL\_NR am FB CANsync\_COMM\_CONTROL\_MA als maximale Slavenummer für automatische Abfrage nach Steuerwort-Senden-Aufträgen, Parameter-Aufträgen und/oder Up-/Download-Aufträgen.

Parameter Ausgang	Datentyp	Beschreibung
_BASE	CANsync_PLC_MA_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung
a_RD_ARRAY	CANsync_RD_BMARRAY	Ausgabearray für Istwerte der CANsync-Slaves

Parameter Ausgang	Datentyp	Beschreibung
si_RD_SL_NR1_RECEIVED	SINT -1, 0 bis 31	Anzeige der Slavenummer des CANsync-Slaves, von dem Istwert-Telegramm 1 empfangen wurde
si_RD_SL_NR2_RECEIVED	SINT -1, 0 bis 31	Anzeige der Slavenummer des CANsync-Slaves, von dem Istwert-Telegramm 2 empfangen wurde

Mit diesem FB werden an die CANsync-Anschaltung Sollwerte (`a_WR_VALUES_SEND`) übergeben, die mittels der Sollwert-Telegramme an die CANsync-Slaves gesendet werden <sup>1)</sup>. Weiterhin werden von den CANsync-Slaves die Istwert-Telegramme angefordert <sup>1)</sup> und die Istwerte (in `a_RD_ARRAY`) ausgegeben.

In jedem CANsync-Intervall werden die Sollwert-Telegramme 1 und 2 an die CANsync-Slaves gesendet.

In jedem CANsync-Intervall wird das Istwert-Telegramm 1 vom CANsync-Slave mit der Slavenummer `si_RD_SL_NR1_RECEIVE` angefordert und in jedem CANsync-Intervall wird das Istwert-Telegramm 2 vom CANsync-Slave mit der Slavenummer `si_RD_SL_NR2_RECEIVE` angefordert.

Die Anforderung der Istwert-Telegramme der CANsync-Slaves läuft automatisch wie folgt ab:

In jedem CANsync-Intervall wird die Slavenummer des CANsync-Slaves (von dem Istwert-Telegramm 1 und 2 angefordert werden) automatisch um eins erhöht. Diese Erhöhung wird bis zur `si_MAX_SL_NR` durchgeführt. Danach wird wieder mit der Abfrage für den CANsync-Slave mit der Slavenummer 0 begonnen usw. (`si_RD_SL_NR1_RECEIVE` und `si_RD_SL_NR2_RECEIVE` werden dann nicht belegt).

Ein-/Ausgang `_BASE`:

Am Ein-/Ausgang `_BASE` wird, entsprechend dem Steckplatz (Slot G bis M) des Optionsmoduls, die globale Variable `_CANsyncMaster_Ctrl_Slot_G` (bis `_CANsyncMaster_Ctrl_Slot_M`) angeschlossen.

Ist diese globale Variable nicht im Projekt vorhanden, legen Sie, je nach Steckplatz (Slot G bis M), die globale Variable `_CANsyncMaster_Ctrl_Slot_G` (bis `_CANsyncMaster_Ctrl_Slot_M`) vom Datentyp `CANsync_PLC_MA_BMSTRUCT` an.

Diese Variable muß über die Deklaration der globalen Variablen auf die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul gelegt werden.

Die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Master (BM4-O-CAN-06) in Abhängigkeit vom Steckplatz ist:

1. sofern eine entsprechende Konfigurierung für die Sollwert-Telegramme 1 und 2 mit dem FB `CANsync_PD_CFG_MA` bzw. die Konfigurierung für die Istwert-Telegramme 1 und 2 von jedem CANsync-Slave mit dem FB `CANsync_PD_CFG_READ_MA` erfolgte.

Steckplatz (Slot)	Basisadresse der CANsync-Anschaltung
G	%MB3.2001792
H	%MB3.3001792
J	%MB3.4001792
K	%MB3.5001792
L	%MB3.6001792
M	%MB3.7001792

Beispiel:

CANsync-Anschaltung auf Optionsmodul CANsync-Master im Steckplatz G des b maXX

```
_CANsyncMaster_Ctrl_Slot_G AT %MB3.2001792 :
                                CANsync_PLC_MA_BMSTRUCT;
```

dabei ist:

<code>_CANsyncMaster_Ctrl_Slot_G</code>	der Variablenname mit der Datentypkurzbezeichnung "_" für Struct
<code>CANsync_PLC_MA_BMSTRUCT</code>	der Datentyp
<code>%MB3.2001792</code>	die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Master im Steckplatz G des b maXX

Ein-/Ausgang `a_RD_ARRAY` (Istwerte 0 bis 7 der CANsync-Slaves 0 bis 31):

An `a_RD_ARRAY` wird eine Variable vom Datentyp `CANsync_RD_BMARRAY` angeschlossen. Der Datentyp `CANsync_RD_BMARRAY` ist ein 2-dimensionales Feld von 32 (CANsync-Slaves) á 16 Istwerten <sup>1)</sup>.

D. h. der Datentyp `CANsync_RD_BMARRAY` ist ein Feld von 32 Einträgen des Datentyps `DINT_16_BMARRAY`. Der Datentyp `DINT_16_BMARRAY` ist ein Feld von 16 Einträgen des Datentyps Double Integer:

```
DINT_16_BMARRAY      : ARRAY [0..15] OF DINT;
CANsync_RD_BMARRAY  : ARRAY [0..31] OF DINT_16_BMARRAY
```

Beispiel:

```
a_Istwerte          : CANsync_RD_BMARRAY;
```

dabei ist:

<code>a_Istwerte</code>	der Variablenname mit der Datentypkurzbezeichnung "a" für Array
<code>CANsync_RD_BMARRAY</code>	der Datentyp

1. zur Zeit werden die Istwerte 0 bis 3 (Istwert-Telegramm 1) und die Istwerte 4 bis 7 (Istwert-Telegramm 2) unterstützt.

Der Zugriff auf die einzelnen Istwerte erfolgt nach dem Muster:

Variablenname[Slavenummer des CANsync-Slaves][Nummer des Istwerts]



### HINWEIS

Zwischen "Variablenname" und den eckigen Klammern sowie zwischen den eckigen Klammern steht kein Punkt.

Beispiel: Die Variable `di_Istwert_21_6` wird mit dem Istwert 6 des CANsync-Slaves mit der Slavenummer 21 beschrieben (in Structured Text (ST)):

```
di_Istwert_21_6 := a_Istwerte[21][6];
```

Eingänge `a_WR_VALUES_SEND`, `si_WRC1_SEND` und `si_WRC2_SEND`:

An dem Eingang `a_WR_VALUES_SEND` wird eine Variable vom Datentyp `DINT_8_BMARRAY` angeschlossen. Der Datentyp `DINT_8_BMARRAY` ist ein Feld mit 8 Einträgen des Datentyps Double Integer:

```
DINT_8_BMARRAY : ARRAY [0..7] OF DINT;
```

Beispiel:

```
a_Sollwerte      : DINT_8_BMARRAY;
```

dabei ist:

`a_Sollwerte`                      der Variablenname mit der Datentypkurzbezeichnung "a" für Array

`DINT_8_BMARRAY`                    der Datentyp

Die Sollwerte 0 bis 7 werden dann z. B. in den Feld-Elementen `a_Sollwerte[0]` bis `a_Sollwerte[7]` erwartet.

Am Eingang `si_WRC1_SEND` wird angegeben wann die Sollwerte 0 bis 3 gültig sind. Dann kann das Sollwert-Telegramm 1 gesendet werden. Der Wert 5 gibt an, dass die Sollwerte 0 bis 3 gültig sind, während jeder andere Wert angibt, dass die Sollwerte 0 bis 3 nicht gültig sind.

Am Eingang `si_WRC2_SEND` wird angegeben wann die Sollwerte 4 bis 7 gültig sind. Dann kann das Sollwert-Telegramm 2 gesendet werden. Der Wert 5 gibt an, dass die Sollwerte 4 bis 7 gültig sind, während jeder andere Wert angibt, dass die Sollwerte 4 bis 7 nicht gültig sind.



### HINWEIS

Im Sollwert-Telegramm 1 werden die Sollwerte 0 bis 3 gesendet, im Sollwert-Telegramm 2 werden die Sollwerte 4 bis 7 gesendet. Lücken bei den Sollwertnummern sind zulässig.

Eingänge `si_RD_SL_NR1_RECEIVE`, `si_RD_SL_NR2_RECEIVE` und `si_MAX_SL_NR`:

Am Eingang `si_RD_SL_NR1_RECEIVE` wird die Slavenummer des CANsync-Slaves angegeben, von dem das Istwert-Telegramm 1 angefordert wird.

Zum automatischen Anfordern des Istwert-Telegramm 1 der CANsync-Slaves wird `si_RD_SL_NR1_RECEIVE` nicht belegt (oder gleich -128 gesetzt) und die höchste Slavenummer an `si_MAX_SL_NR` angegeben.

Am Eingang `si_RD_SL_NR2_RECEIVE` wird die Slavenummer des CANsync-Slaves angegeben, von dem das Istwert-Telegramm 2 angefordert wird.

Zum automatischen Anfordern des Istwert-Telegramm 2 der CANsync-Slaves wird `si_RD_SL_NR2_RECEIVE` nicht belegt (oder gleich -128 gesetzt) und die höchste Slavenummer an `si_MAX_SL_NR` angegeben.

Dann wird in jedem CANsync-Intervall die Slavenummer des CANsync-Slaves (von dem Istwert-Telegramm 1 und/oder 2 automatisch angefordert werden) um eins erhöht, bis die Nummer am Eingang `si_MAX_SL_NR` erreicht ist.

Danach wird wieder mit der Abfrage für den CANsync-Slave mit der Slavenummer 0 begonnen usw.

Die explizite Vorgabe und die automatische Erhöhung kann für die Istwert-Telegramme 1 und 2 auch gemischt werden.

Die höchste Slavenummer eines CANsync-Slaves von dem Istwert-Telegramme angefordert werden (Eingang `si_MAX_SL_NR`) wird auch für die Bedarfsdatenkommunikation (Steuerwort, Parameter, Up-/Download) verwendet.

Ist `si_MAX_SL_NR = -1`, bleibt auf der CANsync-Anschaltung der Wert unverändert.

Voreinstellung ist `si_MAX_SL_NR = -1`, d. h. auf der CANsync-Anschaltung bleibt der Wert unverändert.

#### HINWEIS



Dieser Eingang entspricht dem Eingang `si_MAX_SL_NR` am FB CANsync\_COMM\_CONTROL\_MA. D. h. mit dem Eingang `si_MAX_SL_NR` am FB CANsync\_COMM\_CONTROL\_MA **oder** dem Eingang `si_MAX_SL_NR` am FB CANsync\_PD\_COMM\_MA wird die höchste Slavenummer eines CANsync-Slaves angegeben.

Es darf nur einer von beiden Eingängen verwendet werden!

Eingang `x_COPY_RD_ARRAY`:

Am Eingang `x_COPY_TO_RD_ARRAY` wird mit TRUE angegeben dass die empfangenen Istwerte in das zweidimensionale Feld an `a_RD_ARRAY` eingetragen werden.

Wird an `x_COPY_TO_RD_ARRAY` FALSE angegeben, oder ist `x_COPY_TO_RD_ARRAY` nicht belegt, werden die empfangenen Istwerte nicht in das zweidimensionale Feld an `a_RD_ARRAY` eingetragen.

Ausgänge `si_RD_SL_NR1_RECEIVED`, `si_RD_SL_NR2_RECEIVED`:

Am Ausgang `si_RD_SL_NR1_RECEIVED` wird die Slavenummer des CANsync-Slaves angezeigt von dem im letzten CANsync-Intervall das Istwert-Telegramm 1 empfangen

wurde. Wenn in einem CANsync-Intervall kein Istwert-Telegramm 1 empfangen wurde, wird -128 an si\_RD\_SL\_NR1\_RECEIVED angezeigt.

Am Ausgang si\_RD\_SL\_NR2\_RECEIVED wird die Slavenummer des CANsync-Slaves angezeigt von dem im letzten CANsync-Intervall das Istwert-Telegramm 2 empfangen wurde. Wenn in einem CANsync-Intervall kein Istwert-Telegramm 2 empfangen wurde, wird -128 an si\_RD\_SL\_NR2\_RECEIVED angezeigt.



## 4.20 CANsync\_PD\_COMM\_READ\_MA



### HINWEIS

Der FB CANsync\_PD\_COMM\_READ\_MA ist aus Kompatibilitätsgründen vorhanden und sollte in neuen Projekten nicht eingesetzt werden. Die Istwerte aller CANsync-Slaves werden mit dem FB CANsync\_PD\_COMM\_MA in a\_RD\_BMARRAY ausgegeben.

### Beschreibung

Diesen Funktionsbaustein für CANsync können Sie verwenden, um in einer CANsync-Master-Anschaltung die Prozessdaten-Istwerte von einem CANsync-Slave auszugeben.



### HINWEIS

Der FB CANsync\_PD\_COMM\_READ\_MA verwendet die Bibliothek BM\_TYPES\_20bd03 oder höher.

Parameter Eingang	Datentyp	Beschreibung
_BASE	CANsync_PLC_MA_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung
si_SL_NR	SINT 0 bis 31	Slavenummer des CANsync-Slaves, von dem Istwerte gelesen werden

Parameter Ausgang	Datentyp	Beschreibung
_BASE	CANsync_PLC_MA_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung
a_RD_VALUES	DINT_8_BMARRAY	Ausgabearray für Istwerte die vom CANsync-Slave empfangen wurden
si_RD_SL_NR1_RECEIVED	SINT	Anzeige dass Istwert-Telegramm 1 empfangen wurde
si_RD_SL_NR2_RECEIVED	SINT	Anzeige dass Istwert-Telegramm 2 empfangen wurde

Mit diesem FB werden die Istwerte eines CANsync-Slaves ausgegeben (a\_RD\_VALUES). Die Anforderung der Istwert-Telegramme des CANsync-Slaves muss durch den FB CANsync\_PD\_COMM\_MA erfolgen.

Der Einsatz des FB CANsync\_PD\_COMM\_READ\_MA ist nur sinnvoll, wenn am FB CANsync\_PD\_COMM\_MA die Istwerte nicht ausgegeben werden, d. h. wenn FB CANsync\_PD\_COMM\_MA, Eingang x\_COPY\_TO\_RD\_ARRAY = FALSE ist.

Ein-/Ausgang `_BASE`:

Am Ein-/Ausgang `_BASE` wird, entsprechend dem Steckplatz (Slot G bis M) des Optionsmoduls, die globale Variable `_CANsyncMaster_Ctrl_Slot_G` (bis `_CANsyncMaster_Ctrl_Slot_M`) angeschlossen.

Ist diese globale Variable nicht im Projekt vorhanden, legen Sie, je nach Steckplatz (Slot G bis M), die globale Variable `_CANsyncMaster_Ctrl_Slot_G` (bis `_CANsyncMaster_Ctrl_Slot_M`) vom Datentyp `CANsync_PLC_MA_BMSTRUCT` an.

Diese Variable muß über die Deklaration der globalen Variablen auf die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul gelegt werden.

Die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Master (BM4-O-CAN-06) in Abhängigkeit vom Steckplatz ist:

Steckplatz (Slot)	Basisadresse der CANsync-Anschaltung
G	%MB3.2001792
H	%MB3.3001792
J	%MB3.4001792
K	%MB3.5001792
L	%MB3.6001792
M	%MB3.7001792

Beispiel:

CANsync-Anschaltung auf Optionsmodul CANsync-Master im Steckplatz G des b maXX

```
_CANsyncMaster_Ctrl_Slot_G AT %MB3.2001792 :
                                CANsync_PLC_MA_BMSTRUCT;
```

dabei ist:

`_CANsyncMaster_Ctrl_Slot_G` der Variablenname mit der Datentypkurzbezeichnung "\_" für Struct

`CANsync_PLC_MA_BMSTRUCT` der Datentyp

`%MB3.2001792` die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Master im Steckplatz G des b maXX

Eingang `si_SL_NR`:

Am Eingang `si_SL_NR` wird die Slavenummer des CANsync-Slaves am CANsync-Bus angegeben, dessen Istwerte ausgegeben werden.

Ausgang `a_RD_VALUES` (Istwerte 0 bis 7 des CANsync-Slaves `si_SL_NR`):

An `a_RD_VALUES` wird eine Variable vom Datentyp `DINT_8_BMARRAY` angeschlossen. Der Datentyp `DINT_8_BMARRAY` ist ein Feld von 8 Einträgen des Datentyps `Double Integer`:

```
DINT_8_BMARRAY : ARRAY [0..7] OF DINT;
```

Beispiel:

```
a_Istwerte_3 : DINT_8_BMARRAY;
```

dabei ist:

a_Istwerte_3	der Variablenname mit der Datentypkurzbezeichnung "a" für Array
DINT_8_BMARRAY	der Datentyp

Der Zugriff auf die einzelnen Istwerte erfolgt nach dem Muster:

Variablenname[Nummer des Istwerts]

Beispiel: Die Variable `di_Istwert_3_6` wird mit dem Istwert 6 des CANsync-Slaves mit der Slavenummer 3 beschrieben (in Structured Text (ST)):

```
di_Istwert_3_6 := a_Istwerte_3[6];
```



#### HINWEIS

Zur Zeit werden die Istwerte 0 bis 3 (Istwert-Telegramm 1) und die Istwerte 4 bis 7 (Istwert-Telegramm 2) unterstützt.

Ausgänge `si_RD_SL_NR1_RECEIVED`, `si_RD_SL_NR2_RECEIVED`:

Am Ausgang `si_RD_SL_NR1_RECEIVED` wird mit 2 angezeigt, ob das Istwert-Telegramm 1 empfangen wurde. Wenn das Istwert-Telegramm 1 nicht empfangen wurde, ist `si_RD_SL_NR1_RECEIVED = 0`. Nur wenn das Istwert-Telegramm 1 empfangen wurde, werden die neuen Istwerte 0 bis 3 an `a_RD_VALUES` ausgegeben.

Am Ausgang `si_RD_SL_NR2_RECEIVED` wird mit 2 angezeigt, ob das Istwert-Telegramm 2 empfangen wurde. Wenn das Istwert-Telegramm 2 nicht empfangen wurde, ist `si_RD_SL_NR2_RECEIVED = 0`. Nur wenn das Istwert-Telegramm 2 empfangen wurde, werden die neuen Istwerte 4 bis 7 an `a_RD_VALUES` ausgegeben.

### 4.21 CANsync\_PD\_COMM\_READ\_SL



#### HINWEIS

Der FB CANsync\_PD\_COMM\_READ\_SL ist aus Kompatibilitätsgründen vorhanden und sollte in neuen Projekten nicht eingesetzt werden. Die mitgelesenen Istwerte der anderen CANsync-Slaves werden mit dem FB CANsync\_PD\_COMM\_SL in a\_RD\_BMARRAY ausgegeben.

#### Beschreibung

Diesen Funktionsbaustein für CANsync können Sie verwenden, um in einer CANsync-Slave-Anschaltung die Prozessdaten-Istwerte von einem CANsync-Slave auszugeben.



#### HINWEIS

Der FB CANsync\_PD\_COMM\_READ\_SL verwendet die Bibliothek BM\_TYPES\_20bd03 oder höher.

Parameter Eingang	Datentyp	Beschreibung
_BASE	CANsync_PLC_SL_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung
si_SL_NR	SINT 0 bis 31	Slavenummer des CANsync-Slaves, von dem Istwerte mitgelesen werden

Parameter Ausgang	Datentyp	Beschreibung
_BASE	CANsync_PLC_SL_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung
a_RD_VALUES	DINT_8_BMARRAY	Ausgabearray für Istwerte die vom CANsync-Slave empfangen wurden
si_RD_SL_NR1_RECEIVED	SINT	Anzeige dass Istwert-Telegramm 1 empfangen wurde
si_RD_SL_NR2_RECEIVED	SINT	Anzeige dass Istwert-Telegramm 2 empfangen wurde

Mit diesem FB werden die mitgelesenen Istwerte eines anderen CANsync-Slaves ausgegeben (a\_RD\_VALUES). Die Anforderung der Istwert-Telegramme des anderen CANsync-Slaves muss durch den CANsync-Master mit dem FB CANsync\_PD\_COMM\_MA erfolgen.

Der Einsatz des FB CANsync\_PD\_COMM\_READ\_SL ist nur sinnvoll, wenn am FB CANsync\_PD\_COMM\_SL die Istwerte nicht ausgegeben werden, d. h. wenn FB CANsync\_PD\_COMM\_SL, Eingang x\_COPY\_TO\_RD\_ARRAY = FALSE ist.

Ein-/Ausgang `_BASE`:

Am Ein-/Ausgang `_BASE` wird, entsprechend dem Steckplatz (Slot G bis M) des Optionsmoduls, die globale Variable `_CANsyncSlave_Ctrl_Slot_G` (bis `_CANsyncSlave_Ctrl_Slot_M`) angeschlossen.

Ist diese globale Variable nicht im Projekt vorhanden, legen Sie, je nach Steckplatz (Slot G bis M), die globale Variable `_CANsyncSlave_Ctrl_Slot_G` (bis `_CANsyncSlave_Ctrl_Slot_M`) vom Datentyp `CANsync_PLC_SL_BMSTRUCT` an.

Diese Variable muß über die Deklaration der globalen Variablen auf die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul gelegt werden.

Die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Slave für b maXX PLC (BM4-O-CAN-05, Dip-Schalter 9 = ON) in Abhängigkeit vom Steckplatz ist:

Steckplatz (Slot)	Basisadresse der CANsync-Anschaltung
G	%MB3.2001792
H	%MB3.3001792
J	%MB3.4001792
K	%MB3.5001792
L	%MB3.6001792
M	%MB3.7001792

Beispiel:

CANsync-Anschaltung auf Optionsmodul CANsync-Slave im Steckplatz G des b maXX

```
_CANsyncSlave_Ctrl_Slot_G AT %MB3.2001792 :
                                CANsync_PLC_SL_BMSTRUCT;
```

dabei ist:

<code>_CANsyncSlave_Ctrl_Slot_G</code>	der Variablenname mit der Datentypkurzbezeichnung "_" für Struct
<code>CANsync_PLC_SL_BMSTRUCT</code>	der Datentyp
<code>%MB3.2001792</code>	die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Slave im Steckplatz G des b maXX

Eingang `si_SL_NR`:

Am Eingang `si_SL_NR` wird die Slavenummer des CANsync-Slaves am CANsync-Bus angegeben, dessen mitgelesene Istwerte ausgegeben werden.

Ausgang `a_RD_VALUES` (Istwerte 0 bis 7 des CANsync-Slaves `si_SL_NR`):

An `a_RD_VALUES` wird eine Variable vom Datentyp `DINT_8_BMARRAY` angeschlossen. Der Datentyp `DINT_8_BMARRAY` ist ein Feld von 8 Einträgen des Datentyps Double Integer:

```
DINT_8_BMARRAY : ARRAY [0..7] OF DINT;
```

Beispiel:

```
a_Istwerte_3 : DINT_8_BMARRAY;
```

dabei ist:

a_Istwerte_3	der Variablenname mit der Datentypkurzbezeichnung "a" für Array
DINT_8_BMARRAY	der Datentyp

Der Zugriff auf die einzelnen Istwerte erfolgt nach dem Muster:

Variablenname[Nummer des Istwerts]

Beispiel: Die Variable `di_Istwert_22_6` wird mit dem Istwert 6 des CANsync-Slaves mit der Slavenummer 22 beschrieben (in Structured Text (ST)):

```
di_Istwert_22_6 := a_Istwerte_22[6];
```



### HINWEIS

Zur Zeit werden die Istwerte 0 bis 3 (Istwert-Telegramm 1) und die Istwerte 4 bis 7 (Istwert-Telegramm 2) unterstützt.

Ausgänge `si_RD_SL_NR1_RECEIVED`, `si_RD_SL_NR2_RECEIVED`:

Am Ausgang `si_RD_SL_NR1_RECEIVED` wird mit 2 angezeigt, ob das Istwert-Telegramm 1 empfangen wurde. Wenn das Istwert-Telegramm 1 nicht empfangen wurde, ist `si_RD_SL_NR1_RECEIVED = 0`. Nur wenn das Istwert-Telegramm 1 empfangen wurde, werden die neuen Istwerte 0 bis 3 an `a_RD_VALUES` ausgegeben.

Am Ausgang `si_RD_SL_NR2_RECEIVED` wird mit 2 angezeigt, ob das Istwert-Telegramm 2 empfangen wurde. Wenn das Istwert-Telegramm 2 nicht empfangen wurde, ist `si_RD_SL_NR2_RECEIVED = 0`. Nur wenn das Istwert-Telegramm 2 empfangen wurde, werden die neuen Istwerte 4 bis 7 an `a_RD_VALUES` ausgegeben.

## 4.22 CANsync\_PD\_COMM\_SL

### Beschreibung

Diesen Funktionsbaustein für CANsync können Sie verwenden, um die Prozessdatenkommunikation (Sollwerte und Istwerte) der CANsync-Slave-Anschaltung durchzuführen.



### HINWEIS

Der FB CANsync\_PD\_COMM\_SL verwendet die Bibliothek BM\_TYPES\_20bd03 oder höher.

Parameter Eingang	Datentyp	Beschreibung
_BASE	CANsync_PLC_SL_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung
a_RD_ARRAY	CANsync_RD_BMARRAY	Ausgabearray für Istwerte der CANsync-Slaves
a_RD_VALUES_SEND	DINT_8_BMARRAY	Istwerte, die an den CANsync-Master gesendet werden
si_RDC1_SEND	SINT 5	Kommando, dass Istwert-Telegramm 1 gesendet werden soll
si_RDC2_SEND	SINT 5	Kommando, dass Istwert-Telegramm 2 gesendet werden soll
x_COPY_TO_RD_ARRAY	BOOL	Angabe, ob empfangene Istwerte in das RD_ARRAY kopiert werden sollen

Parameter Ausgang	Datentyp	Beschreibung
_BASE	CANsync_PLC_SL_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung
a_RD_ARRAY	CANsync_RD_BMARRAY	Ausgabearray für Istwerte der CANsync-Slaves
a_WR_VALUES_RECEIVED	DINT_8_BMARRAY	Sollwerte, die vom CANsync-Master empfangen wurden
si_WRC1_RECEIVED	SINT	Anzeige, dass Sollwert-Telegramm 1 empfangen wurde
si_WRC2_RECEIVED	SINT	Anzeige, dass Sollwert-Telegramm 2 empfangen wurde
si_RD_SL_NR1_RECEIVED	SINT	Anzeige Slavenummer des CANsync-Slaves, von dem Istwert-Telegramm 1 empfangen wurde

Parameter Ausgang	Datentyp	Beschreibung
si_RD_SL_NR2_RECEIVED	SINT	Anzeige Slavenummer des CANsync-Slaves, von dem Istwert-Telegramm 2 empfangen wurde

Mit diesem FB werden an die CANsync-Anschaltung Istwerte (a\_RD\_VALUES\_SEND) übergeben, die mittels der Istwert-Telegramme an den CANsync-Master gesendet werden<sup>1)</sup>. Weiterhin werden von den anderen CANsync-Slaves am CANsync-Bus die Istwert-Telegramme mitgelesen<sup>1)</sup> und die Istwerte (in a\_RD\_ARRAY) ausgegeben.

In jedem CANsync-Intervall werden die Sollwert-Telegramme 1 und 2 an die CANsync-Slaves gesendet und von diesem FB an a\_WR\_VALUES\_RECEIVED ausgegeben.

In jedem CANsync-Intervall fordert der CANsync-Master von einem CANsync-Slave das Istwert-Telegramm 1 an. Nur wenn der CANsync-Master von diesem CANsync-Slave das Istwert-Telegramm 1 anfordert, wird das Istwert-Telegramm 1 gesendet.

In jedem CANsync-Intervall fordert der CANsync-Master von einem CANsync-Slave das Istwert-Telegramm 2 an. Nur wenn der CANsync-Master von diesem CANsync-Slave das Istwert-Telegramm 2 anfordert, wird das Istwert-Telegramm 2 gesendet.

Dieser FB kann die Istwert-Telegramme 1 und 2 der anderen CANsync-Slaves mitlesen und auswerten wenn der CANsync-Master diese Istwert-Telegramme anfordert.



### HINWEIS

Ein CANsync-Slave kann Istwert-Telegramme anderer CANsync-Slaves auswerten aber nicht anfordern!

Ein-/Ausgang \_BASE:

Am Ein-/Ausgang \_BASE wird, entsprechend dem Steckplatz (Slot G bis M) des Optionsmoduls, die globale Variable \_CANsyncSlave\_Ctrl\_Slot\_G (bis \_CANsyncSlave\_Ctrl\_Slot\_M) angeschlossen.

Ist diese globale Variable nicht im Projekt vorhanden, legen Sie, je nach Steckplatz (Slot G bis M), die globale Variable \_CANsyncSlave\_Ctrl\_Slot\_G (bis \_CANsyncSlave\_Ctrl\_Slot\_M) vom Datentyp CANsync\_PLC\_SL\_BMSTRUCT an.

Diese Variable muß über die Deklaration der globalen Variablen auf die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul gelegt werden.

Die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Slave für b maXX PLC (BM4-O-CAN-05, Dip-Schalter 9 = ON) in Abhängigkeit vom Steckplatz ist:

1. sofern eine entsprechende Konfiguration für die Sollwert-Telegramme 1 und 2 mit dem FB CANsync\_PD\_CFG\_SL bzw. die Konfiguration für die Istwert-Telegramme 1 und 2 von jedem anderen (außer diesem) CANsync-Slave mit dem FB CANsync\_PD\_CFG\_READ\_SL erfolgte.



Steckplatz (Slot)	Basisadresse der CANsync-Anschaltung
G	%MB3.2001792
H	%MB3.3001792
J	%MB3.4001792
K	%MB3.5001792
L	%MB3.6001792
M	%MB3.7001792

Beispiel:

CANsync-Anschaltung auf Optionsmodul CANsync-Slave im Steckplatz G des b maXX

```
_CANsyncSlave_Ctrl_Slot_G AT %MB3.2001792 :
                                CANsync_PLC_SL_BMSTRUCT;
```

dabei ist:

<code>_CANsyncSlave_Ctrl_Slot_G</code>	der Variablenname mit der Datentypkurzbezeichnung "_" für Struct
<code>CANsync_PLC_SL_BMSTRUCT</code>	der Datentyp
<code>%MB3.2001792</code>	die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Slave im Steckplatz G des b maXX

Ein-/Ausgang `a_RD_ARRAY` (Istwerte 0 bis 7 der (anderen) CANsync-Slaves 0 bis 31):

An `a_RD_ARRAY` wird eine Variable vom Datentyp `CANsync_RD_BMARRAY` angeschlossen.

Der Datentyp `CANsync_RD_BMARRAY` ist ein 2-dimensionales Feld von 32 (CANsync-Slaves) á 16 Istwerten <sup>1</sup>).

D. h. der Datentyp `CANsync_RD_BMARRAY` ist ein Feld von 32 Einträgen des Datentyps `DINT_16_BMARRAY`. Der Datentyp `DINT_16_BMARRAY` ist ein Feld von 16 Einträgen des Datentyps Double Integer:

```
DINT_16_BMARRAY : ARRAY [0..15] OF DINT;
CANsync_RD_BMARAY : ARRAY [0..31] OF DINT_16_BMARRAY
```

Beispiel:

```
a_Istwerte : CANsync_RD_BMARRAY;
```

dabei ist:

<code>a_Istwerte</code>	der Variablenname mit der Datentypkurzbezeichnung "a" für Array
<code>CANsync_RD_BMARRAY</code>	der Datentyp

1. zur Zeit werden die Istwerte 0 bis 3 (Istwert-Telegramm 1) und die Istwerte 4 bis 7 (Istwert-Telegramm 2) unterstützt.

Der Zugriff auf die einzelnen Istwerte erfolgt nach dem Muster:

Variablenname[Slavenummer des CANsync-Slaves][Nummer des Istwerts]



### HINWEIS

Zwischen "Variablenname" und den eckigen Klammern sowie zwischen den eckigen Klammern steht kein Punkt.

Beispiel: Die Variable `di_Istwert_21_6` wird mit dem Istwert 6 des CANsync-Slaves mit der Slavenummer 21 beschrieben (in Structured Text (ST)):

```
di_Istwert_21_6 := a_Istwerte[21][6];
```



### HINWEIS

Die mitgelesenen Istwerte der anderen CANsync-Slaves werden in `a_RD_ARRAY` eingetragen, jedoch nicht die Istwerte dieses CANsync-Slaves.

Eingänge `a_RD_VALUES_SEND`, `si_RDC1_SEND` und `si_RDC2_SEND`:

An dem Eingang `a_RD_VALUES_SEND` wird eine Variable vom Datentyp `DINT_8_BMARRAY` angeschlossen. Der Datentyp `DINT_8_BMARRAY` ist ein Feld mit 8 Einträgen des Datentyps Double Integer:

```
DINT_8_BMARRAY : ARRAY [0..7] OF DINT;
```

Beispiel:

```
a_Istwerte_Senden : DINT_8_BMARRAY;
```

dabei ist:

`a_Istwerte_Senden`            der Variablenname mit der Datentypkurzbezeichnung "a" für Array

`DINT_8_BMARRAY`            der Datentyp

Die Istwerte 0 bis 7 werden in die Einträge 0 bis 7 (z. B. `a_Istwerte_Senden[0]` bis `a_Istwerte_Senden[7]`) eingetragen.

Am Eingang `si_RDC1_SEND` wird angegeben wann die Istwerte 0 bis 3 gültig sind. Dann kann das Istwert-Telegramm 1 gesendet werden, sofern der CANsync-Master es anfordert. Der Wert 5 gibt an, dass die Istwerte 0 bis 3 gültig sind, während jeder andere Wert angibt, dass die Istwerte 0 bis 3 nicht gültig sind.

Am Eingang `si_RDC2_SEND` wird angegeben wann die Sollwerte 4 bis 7 gültig sind. Dann kann das Istwert-Telegramm 2 gesendet werden, sofern der CANsync-Master es anfordert. Der Wert 5 gibt an, dass die Istwerte 4 bis 7 gültig sind, während jeder andere Wert angibt, dass die Istwerte 4 bis 7 nicht gültig sind.



### HINWEIS

Im Istwert-Telegramm 1 werden die Istwerte 0 bis 3 gesendet, im Istwert-Telegramm 2 werden die Istwerte 4 bis 7 gesendet. Lücken bei den Istwertnummern sind zulässig.

Eingang `x_COPY_RD_ARRAY`:

Am Eingang `x_COPY_TO_RD_ARRAY` wird mit `TRUE` angegeben dass die mitgelesenen Istwerte in das zweidimensionale Feld an `a_RD_ARRAY` eingetragen werden.

Wird an `x_COPY_TO_RD_ARRAY` `FALSE` angegeben, oder ist `x_COPY_TO_RD_ARRAY` nicht belegt, werden die mitgelesenen Istwerte nicht in das zweidimensionale Feld an `a_RD_ARRAY` eingetragen.

Ausgänge `a_WR_VALUES_RECEIVED`, `si_WRC1_RECEIVED`, `si_WRC2_RECEIVED`:

An dem Ausgang `a_WR_VALUES_RECEIVED` wird eine Variable vom Datentyp `DINT_8_BMARRAY` angeschlossen. Der Datentyp `DINT_8_BMARRAY` ist ein Feld mit 8 Einträgen des Datentyps `Double Integer`:

```
DINT_8_BMARRAY : ARRAY [0..7] OF DINT;
```

Beispiel:

```
a_Sollwerte      : DINT_8_BMARRAY;
```

dabei ist:

`a_Sollwerte`                      der Variablenname mit der Datentypkurzbezeichnung "a" für Array

`DINT_8_BMARRAY`                    der Datentyp

Die vom CANsync-Master empfangenen Sollwerte 0 bis 7 werden dann z. B. in die Feld-Elemente `a_Sollwerte[0]` bis `a_Sollwerte[7]` eingetragen. Die Sollwerte können vom Format Wort oder Doppelwort sein.

Am Ausgang `si_WRC1_RECEIVED` wird mit 2 angezeigt, ob das Sollwert-Telegramm 1 empfangen wurde. Wenn das Sollwert-Telegramm 1 nicht empfangen wurde, ist `si_WRC1_RECEIVED = 0`. Nur wenn das Sollwert-Telegramm 1 empfangen wurde, werden die neuen Sollwerte 0 bis 3 an `a_WR_VALUES_RECEIVED` ausgegeben.

Am Ausgang `si_WRC2_RECEIVED` wird mit 2 angezeigt, ob das Sollwert-Telegramm 2 empfangen wurde. Wenn das Sollwert-Telegramm 2 nicht empfangen wurde, ist `si_WRC2_RECEIVED = 0`. Nur wenn das Sollwert-Telegramm 2 empfangen wurde, werden die neuen Sollwerte 4 bis 7 an `a_WR_VALUES_RECEIVED` ausgegeben.

#### HINWEIS



Im Sollwert-Telegramm 1 werden die Sollwerte 0 bis 3 empfangen, im Sollwert-Telegramm 2 werden die Sollwerte 4 bis 7 empfangen. Lücken bei den Sollwertnummern sind zulässig.

Ausgänge `si_RD_SL_NR1_RECEIVED`, `si_RD_SL_NR2_RECEIVED`:

Am Ausgang `si_RD_SL_NR1_RECEIVED` wird die Slavenummer des CANsync-Slaves angezeigt von dem im letzten CANsync-Intervall das Istwert-Telegramm 1 mitgelesen wurde. Wenn in einem CANsync-Intervall kein Istwert-Telegramm 1 mitgelesen wurde, wird -128 an `si_RD_SL_NR1_RECEIVED` angezeigt.

Am Ausgang `si_RD_SL_NR2_RECEIVED` wird die Slavenummer des CANsync-Slaves angezeigt von dem im letzten CANsync-Intervall das Istwert-Telegramm 2 mitgelesen wurde. Wenn in einem CANsync-Intervall kein Istwert-Telegramm 2 mitgelesen wurde, wird -128 an `si_RD_SL_NR2_RECEIVED` angezeigt.

### 4.23 CANsync\_SL\_TYP\_INIT

#### Beschreibung

Diesen Funktionsbaustein für CANsync können Sie verwenden, um die Slave-Typen für die CANsync-Initialisierung anzugeben.



#### HINWEIS

Dieser FB wird zusammen mit dem FB CANsync\_INIT eingesetzt.

Der FB CANsync\_SL\_TYP\_INIT verwendet die Bibliothek BM\_TYPES\_20bd03 oder höher.

Parameter Eingang	Datentyp	Beschreibung
us_SL_TYP0	USINT	Slave-Typ 0
us_SL_TYP1	USINT	Slave-Typ 1
us_SL_TYP2	USINT	Slave-Typ 2
...	...	...
us_SL_TYP31	USINT	Slave-Typ 31

Parameter Ausgang	Datentyp	Beschreibung
a_SL_TYP	BYTE_32_BMARRAY	Initialisierungsdaten "Slave-Typen"

Mit diesem FB werden die Slave-Typen der CANsync-Slaves am CANsync-Bus angegeben. Die Daten werden in ein Feld (Ausgang a\_SL\_TYP) eingetragen. Dieses Feld wird am Eingang a\_SL\_TYP des FB CANsync\_INIT angeschlossen.

Eingänge us\_SL\_TYP0 bis us\_SL\_TYP31:

Für den CANsync-Slave mit der Slavenummer 0 wird der Slave-Typ am Eingang us\_SL\_TYP0 angegeben.

- us\_SL\_TYP0 = 0 bedeutet kein CANsync-Slave mit der Slavenummer 0 vorhanden
- us\_SL\_TYP0 = 1 bedeutet CANsync-Slave mit der Slavenummer 0 vorhanden

Für den CANsync-Slave mit der Slavenummer 1 wird der Slave-Typ am Eingang us\_SL\_TYP1 angegeben.

- us\_SL\_TYP1 = 0 bedeutet kein CANsync-Slave mit der Slavenummer 1 vorhanden
- us\_SL\_TYP1 = 1 bedeutet CANsync-Slave mit der Slavenummer 1 vorhanden

usw...

Für den CANsync-Slave mit der Slavenummer 31 wird der Slave-Typ am Eingang us\_SL\_TYP31 angegeben.

- us\_SL\_TYP31 = 0 bedeutet kein CANsync-Slave mit der Slavenummer 31 vorhanden
- us\_SL\_TYP31 = 1 bedeutet CANsync-Slave mit der Slavenummer 31 vorhanden

Bedeutung der Slave-Typen:

Slave-Typ	Bedeutung
0	Kein CANsync-Slave mit der Slavenummer x vorhanden
1	b maXX mit Optionsmodul CANsync-Slave mit der Slavenummer x vorhanden
2 - 255	reserviert

Ausgang a\_SL\_TYP:

Am Ausgang a\_SL\_TYP wird eine Variable vom Datentyp BYTE\_32\_BMARRAY angeschlossen. Der Datentyp BYTE\_32\_BMARRAY ist ein Feld von 32 Einträgen des Datentyps Byte:

```
BYTE_32_BMARRAY : ARRAY [0..31] OF BYTE;
```

Beispiel:

```
a_Slave_Typen : BYTE_32_BMARRAY;
```

dabei ist:

a\_Slave\_Typen                    der Variablenname mit der Datentypkurzbezeichnung "a" für Array

BYTE\_32\_BMARRAY                der Datentyp

In den einzelnen Einträgen des Feldes steht der Slave-Typ der CANsync-Slaves am CANsync-Bus. Im Eintrag [0] steht der Slave-Typ des CANsync-Slaves mit der Slavenummer 0, im Eintrag [1] steht der Slave-Typ des CANsync-Slaves mit der Slavenummer 1 usw.

Eine 0 im Eintrag [x] bedeutet, dass kein CANsync-Slave mit der Slavenummer x am CANsync-Bus vorhanden ist.

Ein Wert  $\neq$  0 im Eintrag [x] bedeutet, dass ein CANsync-Slave mit der Slavenummer x am CANsync-Bus vorhanden ist.



#### HINWEIS

Diese Variable wird am Eingang a\_SL\_TYP des FB CANsync\_INIT angeschlossen.

### 4.24 CANsync\_UPDOWNLOAD\_MA

#### Beschreibung

Diesen Funktionsbaustein für CANsync können Sie verwenden, um einen Upload oder einen Download im Block 1 - Bereich (siehe [▶Up/Download Blockbereich◀](#) ab Seite 68) durchzuführen. Er ist speziell für den Einsatz mit dem FB CANsync\_UPDOWNLOAD\_SL im CANsync-Slave konzipiert.



#### HINWEIS

Für den Einsatz dieses FBs ist es notwendig, für den Kommandokanal (CC) die Upload- und Downloadkommandos freizugeben.

Diese Freigabe erfolgt mit dem FB CANsync\_COMM\_CONTROL\_MA.

Der FB CANsync\_UPDOWNLOAD\_MA verwendet die Bibliothek BM\_TYPES\_20bd03 oder höher.

Parameter Eingang	Datentyp	Beschreibung
_BASE	CANsync_PLC_MA_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung
a_DOWNLOAD	CANsync_UPDOWN_BMARRAY	Downloadwerte
a_UPLOAD	CANsync_UPDOWN_BMARRAY	Uploadwerte
si_SL_NR	SINT 0 bis 31	Slavenummer des CANsync-Slaves, an den der Download- oder Upload-Auftrag gerichtet ist
x_UPDOWN	BOOL	Up- oder Download
u_LENGTH	UINT 1 bis 2048	Länge des Updownblocks
us_MAX_NR_OF_COMM	USINT	max. Anzahl der Kommunikationsversuche
t_TIME	TIME	Überwachungszeit
x_EN	BOOL	Freigabe
x_RESET	BOOL	Reset

Parameter Ausgang	Datentyp	Beschreibung
_BASE	CANsync_PLC_MA_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung
a_DOWNLOAD	CANsync_UPDOWN_BMARRAY	Downloadwerte
a_UPLOAD	CANsync_UPDOWN_BMARRAY	Uploadwerte
x_BUSY	BOOL	Kommunikation ist aktiv
w_ERR_SL	WORD	Fehlerwort (CANsync-Slave -> CANsync-Master)

Parameter Ausgang	Datentyp	Beschreibung
x_ERR_SL	BOOL	Fehlerbit (Sammel-Fehlerbit von w_ERR_SL)
b_ERR	BYTE	Fehlerbyte
x_ERR	BOOL	Fehlerbit (Sammel-Fehlerbit von b_ERR)
x_OK	BOOL	OK-Bit

Der FB CANsync\_UPDOWNLOAD\_MA führt einen Upload oder einen Download im Block 1 - Bereich durch. Der FB CANsync\_UPDOWNLOAD\_MA ist speziell für den Einsatz mit dem FB CANsync\_UPDOWNLOAD\_SL im CANsync-Slave konzipiert.

Der FB CANsync\_UPDOWN\_MA übergibt Download-Daten an den CANsync-Slave mit der Slavenummer si\_SL\_NR.

Der Download-Auftrag besteht aus u\_LENGTH Werten aus dem Feld a\_DOWNLOAD.

(Der FB CANsync\_UPDOWN\_MA führt einen Download-Auftrag an den CANsync-Slave in mehreren Download-Telegramm-Blöcken aus. Je Download-Telegramm-Block werden maximal 75 Werte aus dem Feld a\_DOWNLOAD an den CANsync-Slave gesendet.)

Der CANsync-Slave bearbeitet den Download-Auftrag und gibt das Ergebnis der Kommunikation zurück.

Der FB CANsync\_UPDOWN\_MA fordert Upload-Daten von dem CANsync-Slave mit der Slavenummer si\_SL\_NR an.

Es werden u\_LENGTH Werte vom CANsync-Slave angefordert.

(Der FB CANsync\_UPDOWN\_MA führt einen Upload-Auftrag an den CANsync-Slave in mehreren Upload-Telegramm-Blöcken aus. Je Upload-Telegramm-Block werden maximal 75 Werte vom CANsync-Slave empfangen.)

Der CANsync-Slave bearbeitet den Upload-Auftrag und gibt das Ergebnis der Kommunikation zurück. Die Werte werden an a\_UPLOAD ausgegeben.

Ein-/Ausgang \_BASE:

Am Ein-/Ausgang \_BASE wird, entsprechend dem Steckplatz (Slot G bis M) des Optionsmoduls, die globale Variable `_CANsyncMaster_Ctrl_Slot_G` (bis `_CANsyncMaster_Ctrl_Slot_M`) angeschlossen.

Ist diese globale Variable nicht im Projekt vorhanden, legen Sie, je nach Steckplatz (Slot G bis M), die globale Variable `_CANsyncMaster_Ctrl_Slot_G` (bis `_CANsyncMaster_Ctrl_Slot_M`) vom Datentyp `CANsync_PLC_MA_BMSTRUCT` an.

Diese Variable muß über die Deklaration der globalen Variablen auf die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul gelegt werden.

Die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Master (BM4-O-CAN-06) in Abhängigkeit vom Steckplatz ist:

Steckplatz (Slot)	Basisadresse der CANsync-Anschaltung
G	%MB3.2001792
H	%MB3.3001792
J	%MB3.4001792
K	%MB3.5001792
L	%MB3.6001792
M	%MB3.7001792

Beispiel:

CANsync-Anschaltung auf Optionsmodul CANsync-Master im Steckplatz G des b maXX

```

_CANsyncMaster_Ctrl_Slot_G AT %MB3.2001792 :
                                CANsync_PLC_MA_BMSTRUCT;

```

dabei ist:

<code>_CANsyncMaster_Ctrl_Slot_G</code>	der Variablenname mit der Datentypkurzbezeichnung "_" für Struct
<code>CANsync_PLC_MA_BMSTRUCT</code>	der Datentyp
<code>%MB3.2001792</code>	die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Master im Steckplatz G des b maXX

Ein-/Ausgang a\_DOWNLOAD:

An a\_DOWNLOAD wird eine Variable vom Datentyp CANsync\_UPDOWN\_BMARRAY angeschlossen. Der Datentyp CANsync\_UPDOWN\_BMARRAY ist ein Feld von 2048 Einträgen des Datentyps Double Integer.

```

CANsync_UPDOWN_BMARRAY : ARRAY [0..2047] OF DINT;

```

Beispiel:

```

a_Downloadwerte : CANsync_UPDOWN_BMARRAY;

```

dabei ist:

<code>a_Downloadwerte</code>	der Variablenname mit der Datentypkurzbezeichnung "a" für Array
<code>CANsync_UPDOWN_BMARRAY</code>	der Datentyp

Die Daten für den Download werden dann z. B. in den Feld-Elementen `a_Downloadwerte[0]` bis `a_Downloadwerte[2047]` erwartet.

Ein-/Ausgang a\_UPLOAD:

An a\_DOWNLOAD wird eine Variable vom Datentyp CANsync\_UPDOWN\_BMARRAY angeschlossen. Der Datentyp CANsync\_UPDOWN\_BMARRAY ist ein Feld von 2048 Einträgen des Datentyps Double Integer.

```

CANsync_UPDOWN_BMARRAY : ARRAY [0..2047] OF DINT;

```



Beispiel:

```
a_Uploadwerte : CANsync_UPDOWN_BMARRAY;
```

dabei ist:

a\_Uploadwerte                      der Variablenname mit der Datentypkurzbezeichnung "a" für Array

CANsync\_UPDOWN\_BMARRAY        der Datentyp

Beim Upload werden die Daten dann z. B. in den Feld-Elementen a\_Uploadwerte[0] bis a\_Uploadwerte[2047] ausgegeben.

Eingang si\_SL\_NR:

Am Eingang si\_SL\_NR wird die Slavenummer des CANsync-Slaves am CANsync-Bus angegeben, von dem ein Upload oder zu dem ein Download durchgeführt werden soll.

#### HINWEIS



Der Up-/Download zu diesem CANsync-Slave muss über den FB CANsync\_COMM\_CONTROL\_MA freigegeben werden (siehe Beschreibung FB CANsync\_COMM\_CONTROL\_MA).

Eingang x\_UPDOWN:

Am Eingang x\_UPDOWN wird mit x\_UPDOWN = FALSE ein Upload-Auftrag und mit x\_UPDOWN = TRUE ein Download-Auftrag eingestellt.

Eingang u\_LENGTH:

Am Eingang u\_LENGTH wird die Länge des zu übertragenden Up-/Download-Bereiches angegeben. Es werden maximal 2048 Werte á 32 Bit übertragen. Wenn an u\_LENGTH eine 0 angegeben ist, wird im Fehlerbyte b\_ERR das Bit1 auf TRUE gesetzt.

Die Daten werden Blockweise in Telegramm-Blöcken übertragen. Ein Telegramm-Block ist 75 Werte á 32 Bit groß. D. h. bei einem Download- oder Upload-Auftrag über mehr als 75 Werte á 32 Bit werden entsprechend mehrere Telegramm-Blöcke gesendet.

Eingang us\_MAX\_NR\_OF\_COMM:

Am Eingang us\_MAX\_NR\_OF\_COMM kann angegeben werden, wie oft ein Block des Up-Download-Bereiches wiederholt werden soll, wenn a) die Überwachungszeit (des CANsync-Masters) abgelaufen und b) vom CANsync-Slave kein Fehler gemeldet wurde (Voreinstellung ist us\_MAX\_NR\_OF\_COMM = 1). Die Timeout-Meldung (siehe Eingang t\_TIME) erfolgt dann erst, nachdem die Zeit us\_MAX\_NR\_OF\_COMM \* t\_TIME abgelaufen ist und bis dahin keine Antwort vom CANsync-Slave vorliegt.

Eingang t\_TIME:

Die Überwachungszeit wird am Eingang t\_TIME angegeben. Wird der Download-Auftrag oder der Upload-Auftrag nicht innerhalb der Überwachungszeit vollständig abgearbeitet, wird im Fehlerbyte das Bit 2 gesetzt (siehe auch Eingang us\_MAX\_NR\_OF\_COMM).

Ursache für das nicht vollständige Abarbeiten eines Download- oder Upload-Auftrags kann die Belegung des Kommandokanals mit höherpriorigen Nachrichten sein (Broadcast-kommandos, Steuerwort-Senden-Aufträge, Parameter-Aufträge; siehe [►Bedarfsdaten◄](#) ab Seite 23). Weiterhin kann die Ursache die fehlende Freigabe des Kommandokanals für Up-/Download-Aufträge sein (siehe FB CANsync\_COMM\_CONTROL\_MA).

Eingang x\_EN:

Mit x\_EN = TRUE wird der FB CANsync\_UPDOWNLOAD\_MA freigegeben d. h. Download- oder Upload-Aufträge können an den CANsync-Slave gesendet werden.

Falls x\_EN auf FALSE gesetzt wird, bevor der Download- oder Upload-Auftrag abgeschlossen ist, wird von einem bewussten Abbruch des Auftrags ausgegangen. Der FB CANsync\_UPDOWNLOAD\_MA muss anschließend mit x\_RESET = TRUE zurückgesetzt werden, um einen neuen Download- oder Upload-Auftrag über die angegebene Länge (Eingang u\_LENGTH) zu starten.

Eingang x\_RESET:

Mit x\_RESET gleich TRUE kann der FB CANsync\_UPDOWNLOAD\_MA zurückgesetzt werden. Dies ist z. B. nach einem Abbruch des Download- oder Upload-Auftrags (durch x\_EN = FALSE) oder nach einer Fehlermeldung notwendig. Anschließend muss x\_RESET wieder auf FALSE gesetzt werden.

Ausgang x\_BUSY:

Der Ausgang x\_BUSY zeigt mit TRUE an, dass der FB CANsync\_UPDOWNLOAD\_MA einen Auftrag abarbeitet.

Ausgang x\_OK:

Der Ausgang x\_OK zeigt mit TRUE die erfolgreiche Ausführung des Download- oder Upload-Auftrags an. Der Ausgang x\_OK ist FALSE wenn kein Download- oder Upload-Auftrag ausgeführt oder ein Auftrag nicht korrekt ausgeführt wurde.

Ausgänge x\_ERR\_SL, w\_ERR\_SL:

Falls bei der Ausführung des Download- oder Upload-Auftrags im CANsync-Slave ein Fehler auftritt, wird das Fehlerbit x\_ERR\_SL auf TRUE gesetzt und das Fehlerwort w\_ERR\_SL ausgegeben (siehe unten). Der Ausgang x\_OK bleibt dann FALSE.

Ausgänge x\_ERR, b\_ERR:

Falls bei der Ausführung des Download- oder Upload-Auftrags ein Fehler auftritt, wird das Fehlerbit x\_ERR auf TRUE gesetzt und das Fehlerbyte b\_ERR ausgegeben (siehe unten). Der Ausgang x\_OK bleibt dann FALSE.

Fehlerbyte b\_ERR:

Bit-Nr.	Bedeutung
---------	-----------

0	CANsync-Slave meldet Fehler bei der Bearbeitung des Up-/Download-Auftrags
1	Länge des zu übertragenden Up-/Download-Bereiches gleich 0 (Eingang u_LENGTH)
2	Timeout
3 bis 7	Reserviert

Fehlerwort w\_ERR\_SL:

w_ERR_SL	Bedeutung
16#0000	Reserviert
16#0001	CANsync-Slave quittiert falsche Blocknummer
16#0002	eingetragene Übertragungsblock-Länge > 300 Byte / 75 Doppelworte
16#0003 bis 16#00FF	Reserviert
16#0100	CANsync-Slave erwartet Übertragungsblock mit der Nummer, die im Zähler eingetragen ist
16#0101	CANsync-Slave erwartet Übertragungsblock-Ende
16#0102	CANsync-Slave erwartet noch nicht Übertragungsblock-Ende
16#0103	CANsync-Slave bricht Auftrag ab
16#0104	Auftrag nicht möglich
16#0105	Basisadresse nicht erlaubt
16#0106	Reserviert
16#0107	Up-/Download-Bereich CANsync-Master > Up-/Download-Bereich CANsync-Slave
16#0108	Telegramm-Modedefehler (in dieser Phase nicht erlaubter Mode)
16#0109 bis 16#FFFF	Reserviert

### 4.25 CANsync\_UPDOWNLOAD\_SL

#### Beschreibung

Diesen Funktionsbaustein für CANsync können Sie verwenden, um je nach Auftrag vom CANsync-Master einen Upload oder einen Download im Block 1 - Bereich (siehe [▶Up/Download-Kommando](#) ab Seite 96) durchzuführen. Er ist speziell für den Einsatz mit dem FB CANsync\_UPDOWNLOAD\_MA im CANsync-Master konzipiert.



#### HINWEIS

Der FB CANsync\_UPDOWNLOAD\_SL verwendet die Bibliothek BM\_TYPES\_20bd03 oder höher.

Parameter Eingang	Datentyp	Beschreibung
_BASE	CANsync_PLC_SL_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung
a_DOWNLOAD	CANsync_UPDOWN_BMARRAY	Downloadwerte
a_UPLOAD	CANsync_UPDOWN_BMARRAY	Uploadwerte
t_TIME	TIME	Überwachungszeit
x_EN	BOOL	Freigabe
x_RESET	BOOL	Reset

Parameter Ausgang	Datentyp	Beschreibung
_BASE	CANsync_PLC_SL_BMSTRUCT	Betriebsdaten für die CANsync-Anschaltung
a_DOWNLOAD	CANsync_UPDOWN_BMARRAY	Downloadwerte
a_UPLOAD	CANsync_UPDOWN_BMARRAY	Uploadwerte
x_UPDOWN	BOOL	Anzeige Up- oder Download
x_ORDER_ACTIV	BOOL	Anzeige Auftrag steht an
u_LENGTH	UINT	Länge des übertragenen Blocks
x_BUSY	BOOL	Anzeige FB ist aktiv
w_ERR	WORD	Fehlerwort
x_ERR	BOOL	Fehlerbit
x_OK	BOOL	OK-Bit

Der FB CANsync\_UPDOWNLOAD\_SL führt einen Download-Auftrag oder einen Upload-Auftrag des CANsync-Masters aus. Downloadwerte (a\_DOWNLOAD) werden ausgegeben, Uploadwerte (a\_UPLOAD) werden an den CANsync-Master gesendet.

Die Daten werden in mehreren Telegramm-Blöcken übertragen. Je Telegramm-Block werden maximal 75 Werte á 32 Bit übertragen.

Der FB CANsync\_UPDOWNLOAD\_SL ist speziell für den Einsatz mit dem FB CANsync\_UPDOWNLOAD\_MA im CANsync-Master konzipiert.

Ein-/Ausgang \_BASE:

Am Ein-/Ausgang \_BASE wird, entsprechend dem Steckplatz (Slot G bis M) des Optionsmoduls, die globale Variable \_CANsyncSlave\_Ctrl\_Slot\_G (bis \_CANsyncSlave\_Ctrl\_Slot\_M) angeschlossen.

Ist diese globale Variable nicht im Projekt vorhanden, legen Sie, je nach Steckplatz (Slot G bis M), die globale Variable \_CANsyncSlave\_Ctrl\_Slot\_G (bis \_CANsyncSlave\_Ctrl\_Slot\_M) vom Datentyp CANsync\_PLC\_SL\_BMSTRUCT an.

Diese Variable muß über die Deklaration der globalen Variablen auf die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul gelegt werden.

Die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Slave für b maXX PLC (BM4-O-CAN-05, Dip-Schalter 9 = ON) in Abhängigkeit vom Steckplatz ist:

Steckplatz (Slot)	Basisadresse der CANsync-Anschaltung
G	%MB3.2001792
H	%MB3.3001792
J	%MB3.4001792
K	%MB3.5001792
L	%MB3.6001792
M	%MB3.7001792

Beispiel:

CANsync-Anschaltung auf Optionsmodul CANsync-Slave im Steckplatz G des b maXX

```
_CANsyncSlave_Ctrl_Slot_G AT %MB3.2001792 :
                                CANsync_PLC_SL_BMSTRUCT;
```

dabei ist:

- \_CANsyncSlave\_Ctrl\_Slot\_G      der Variablenname mit der Datentypkurzbezeichnung "\_" für Struct
- CANsync\_PLC\_SL\_BMSTRUCT      der Datentyp
- %MB3.2001792                  die Basisadresse der CANsync-Anschaltung auf dem Optionsmodul CANsync-Slave im Steckplatz G des b maXX

Ein-/Ausgang a\_DOWNLOAD:

An a\_DOWNLOAD wird eine Variable vom Datentyp CANsync\_UPDOWN\_BMARRAY angeschlossen. Der Datentyp CANsync\_UPDOWN\_BMARRAY ist ein Feld von 2048 Einträgen des Datentyps Double Integer.

```
CANsync_UPDOWN_BMARRAY                  : ARRAY [0..2047] OF DINT;
```

Beispiel:

```
a_Downloadwerte : CANsync_UPDOWN_BMARRAY;
```

dabei ist:

a\_Downloadwerte                    der Variablenname mit der Datentypkurzbezeichnung "a" für Array

CANsync\_UPDOWN\_BMARRAY            der Datentyp

Beim Download werden die Daten dann z. B. in den Feld-Elementen a\_Downloadwerte[0] bis a\_Downloadwerte[2047] ausgegeben.

Ein-/Ausgang a\_UPLOAD:

An a\_DOWNLOAD wird eine Variable vom Datentyp CANsync\_UPDOWN\_BMARRAY angeschlossen. Der Datentyp CANsync\_UPDOWN\_BMARRAY ist ein Feld von 2048 Einträgen des Datentyps Double Integer.

```
CANsync_UPDOWN_BMARRAY            : ARRAY [0..2047] OF DINT;
```

Beispiel:

```
a_Uploadwerte : CANsync_UPDOWN_BMARRAY;
```

dabei ist:

a\_Uploadwerte                    der Variablenname mit der Datentypkurzbezeichnung "a" für Array

CANsync\_UPDOWN\_BMARRAY            der Datentyp

Die Daten für den Upload werden dann z. B. in den Feld-Elementen a\_Uploadwerte[0] bis a\_Uploadwerte[2047] erwartet.

Eingang t\_TIME:

Die Überwachungszeit wird am Eingang t\_TIME angegeben. Wird der Download-Auftrag oder der Upload-Auftrag nicht innerhalb der Überwachungszeit vollständig abgearbeitet, wird im Fehlerwort das Bit 0 gesetzt. Wird der Eingang t\_TIME nicht belegt, ergibt sich eine Voreinstellung der Überwachungszeit von 30 s.

Ursache für das nicht vollständige Abarbeiten eines Download- oder Upload-Auftrags kann die Belegung des Kommandokanals mit höherpriorigen Nachrichten sein (Broadcastkommandos, Steuerwort-Senden-Aufträge, Parameter-Aufträge; siehe [►Bedarfsdaten◄](#) ab Seite 23). Weiterhin kann die Ursache die fehlende Freigabe des Kommandokanals für Up-/Download-Aufträge sein (siehe FB [►CANsync\\_COMM\\_CONTROL\\_MA◄](#) ab Seite 113).

Eingang x\_EN:

Mit x\_EN = TRUE wird der FB CANsync\_UPDOWNLOAD\_SL freigegeben, d. h. Download- oder Upload-Aufträge können bearbeitet werden.

Falls x\_EN auf FALSE gesetzt wird, bevor der Download- oder Upload-Auftrag abgeschlossen ist, wird von einem bewussten Abbruch ausgegangen. Der FB CANsync\_UPDOWNLOAD\_SL muss anschließend mit x\_RESET = TRUE zurückgesetzt werden, um einen neuen Download- oder Upload-Auftrag ausführen zu können.

**Eingang x\_RESET:**

Mit `x_RESET = TRUE` kann der FB `CANsync_UPDOWNLOAD_SL` zurückgesetzt werden. Dies ist z. B. nach einem Abbruch des Download- oder Upload-Auftrags (durch `x_EN = FALSE`) oder nach einer Fehlermeldung notwendig. Anschließend muss `x_RESET` wieder auf `FALSE` gesetzt werden.

**Ausgänge x\_UPDOWN, x\_ORDER\_ACTIV:**

Der Ausgang `x_ORDER_ACTIV` zeigt mit `TRUE` an, dass ein Auftrag vorliegt. Die Art des Auftrags wird durch den Ausgang `x_UPDOWN` angezeigt. Bei einem Upload-Auftrag ist `x_UPDOWN = FALSE`, bei einem Download-Auftrag ist `x_UPDOWN = TRUE`.

**Ausgang u\_LENGTH:**

Der Ausgang `u_LENGTH` zeigt die Anzahl der Werte á 32 Bit des übertragenen Telegramm-Blocks an. Ein Telegramm-Block ist maximal 75 Werte á 32 Bit groß. Bei einem Download- oder Upload-Auftrag über mehr als 75 Werte á 32 Bit werden entsprechend mehrere Telegramm-Blöcke übertragen und an `u_LENGTH` die Anzahl der Werte á 32 Bit des aktuell übertragenen Telegramm-Blocks angezeigt.

**Ausgang x\_BUSY:**

Der Ausgang `x_BUSY` zeigt mit `TRUE` an, dass der FB `CANsync_UPDOWNLOAD_SL` einen Download- oder Upload-Auftrag bearbeitet.

**Ausgang x\_OK:**

Der Ausgang `x_OK` zeigt mit `TRUE` die erfolgreiche Ausführung des Download- oder Upload-Auftrags an. Der Ausgang `x_OK` ist `FALSE` wenn kein Download- oder Upload-Auftrag ausgeführt wurde oder ein Auftrag nicht korrekt ausgeführt wurde.

**Der Ausgänge x\_ERR, w\_ERR:**

Falls ein Fehler auftritt, wird das Fehlerbit `x_ERR` auf `TRUE` gesetzt und das Fehlerwort `w_ERR` ausgegeben. Der Ausgang `x_OK` bleibt dann `FALSE`.

**Fehlerwort w\_ERR:**

Bit-Nr.	Bedeutung
0	Timeout
1 bis 15	reserviert







## ANHANG A - ABKÜRZUNGEN

<b>BACI</b>	Baumüller Component Interface	<b>RAM</b>	Random Access Memory
<b>BUB</b>	Ballast-Einheit	<b>RC</b>	Antwortkanal (Response Channel)
<b>BUC</b>	Baumüller Ein-/Rückspeise-Einheit	<b>RDC</b>	Read Channel (Istwertkanal, IWK)
<b>BUG</b>	Baumüller Umrichter Grund-Einspeise-Einheit	<b>SWK</b>	Sollwertkanal (Write Channel, WRC)
<b>BUM</b>	Baumüller Einzel-Leistungs-Einheit	<b>t<sub>RSPTO</sub></b>	Response Timeout
<b>BUS</b>	Baumüller Leistungs-Modul	<b>UL</b>	Underwriter Laboratories Inc.
<b>CAN</b>	Controller Area Network	<b>USS</b>	Funktionsmodul USS-Protokoll
<b>CANsync</b>	synchronisierter CAN	<b>USS®</b>	Warenzeichen Siemens, universelle serielle Schnittstelle
<b>CC</b>	Kommandokanal (Command Channel)	<b>VDE</b>	Verband deutscher Elektrotechniker
<b>CPU</b>	Central Processing Unit	<b>WRC</b>	Write Channel (Sollwertkanal, SWK)
<b>DB</b>	Datenbyte		
<b>DC</b>	Gleichstrom		
<b>DP-RAM</b>	Dual-Port RAM		
<b>DIN</b>	Deutsches Institut für Normung e.V.		
<b>EMV</b>	Elektromagnetische Verträglichkeit		
<b>EN</b>	Europäische Norm		
<b>EPROM</b>	Erasable Programmable Read Only Memory		
<b>ESD</b>	Electrostatic sensitive devices		
<b>FB</b>	Funktionsbaustein		
<b>I/O</b>	Input/Output, Eingang und Ausgang		
<b>ISO</b>	International Standard Organisation		
<b>IWK</b>	Istwertkanal (Read Channel, RDC)		
<b>LED</b>	Leuchtdiode		
<b>16#</b>	Präfix für Hexadezimalzahl		





# Index

<b>A</b>	
Acceptance Code	50, 79
Acceptance Mask	50, 79
Aktionskommando	36
Empfangen	93
Anlaufverhalten	32
Antwortkanal	36
<b>B</b>	
Baudrate	50, 79
Baumüller	7
Begriffe	
Definition	5
Betriebsart einstellen	129, 133
BM4-O-CAN-06	5
Broadcastkommando	64
Empfangen	110
Senden	101, 104, 107
<b>C</b>	
CANsync	
Bedarfsdatenkommunikation	23
Initialisierung	25
Mapping	19
Prozeßdatenkommunikation	18, 22
Prozessdatenkommunikation	25
CANsync-Anschaltung initialisieren	122
CANsync-Buslänge	18
CANsync-Intervall	17
CANsync-Status	48, 77
CANsync-Synchronisationssignal	16
CANsync-Zykluszeit	17
<b>D</b>	
Datenformat	33
Download	39
Auftrag	68
Ende	41
Download-Kommando	
Empfangen	96
Downloadvorgang	41
<b>E</b>	
Einleitung	5
Erste Schritte	5
<b>F</b>	
Fachkraft	13
FB	
Betriebsart einstellen	129, 133
Broadcastkommando empfangen	110
Broadcastkommando senden	101, 104, 107
CANsync-Anschaltung initialisieren	122
Istwert-Telegramme konfigurieren	155, 159
Kommandokanal konfigurieren	113
Parameter-Auftrag erkennen	141
Parameterwert an Slave senden	147
Parameterwert vom Slave anfordern	137
Prozeßdatenkommunikation	169, 181
Slave-Typen für CANsync-Initialisierung angeben	186
Sollwert-Telegramme konfigurieren	151
Steuerwort empfangen	120
Steuerwortkommando senden	117
Up/Download durchführen	188, 194
Funktionsbausteine CANsync	
Übersicht	99
<b>G</b>	
Gewährleistung und Haftung	14
<b>I</b>	
Initialisierung	44, 51
Initialisierungsablauf	80
Initialisierungs-Telegramm	39
Istwert	
von anderen CANsync-Slaves empfangen	88
Istwertanforderung	56
Istwertkanal	61, 92
Benutzung	87, 92
Istwert-Telegramm	35, 61, 93
Istwert-Telegramme konfigurieren	155, 159
<b>K</b>	
Kommandokanal	36
Kommandokanal konfigurieren	113
Konfigurierung	
Kommandokanal	62
Sollwert-Telegramm	53
Konfigurierungsregister	62
Kontrollregister	56
<b>M</b>	
Mapping	19
Master	
Sendezeitpunkt	33
<b>O</b>	
Optionsmodul	5
Optionsmodul CANsync-Master	5
<b>P</b>	
Parameter Lesen	
Kommando	36
Parameter Schreiben	



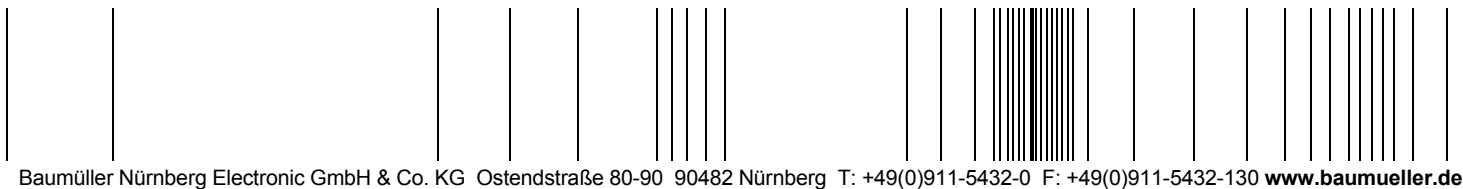
## Stichwortverzeichnis

---

Kommando	38	<b>V</b>	
Parameter-Auftrag erkennen	141	Verpflichtung und Haftung	14
Parameterkommando	36, 66	<b>Z</b>	
Bearbeiten	95	Zustand „synchronisiert“	33
Empfangen	94	zyklischen Betrieb	45
Parameterwert an Slave senden	147		
Parameterwert vom Slave anfordern	137		
Parameterzugriff			
Ablauf	95		
Parameterzugriff, Ablauf	67		
Personal	13		
qualifiziert	13		
Prozeßdatenkommunikation	169, 181		
<b>Q</b>			
Qualifiziertes Personal	13		
<b>R</b>			
Register	52		
Response Timeout	32		
<b>S</b>			
Sicherheitshinweise	7		
Slave-Typen für CANsync-Initialisierung angeben	186		
Sollwert			
Empfangen	81		
Sollwerte			
Senden	52		
Sollwertkanal	55		
Benutzung	83		
Sollwert-Telegramm	34, 55		
Sollwert-Telegramme konfigurieren	151		
Statuswort	33		
Steckmodul	5		
Steuerwort empfangen	120		
Steuerwortkommando	65		
Steuerwortkommando senden	117		
Struktur	43, 72		
SYNC-Signal	16		
<b>U</b>			
Übersicht			
CANsync Funktionsbausteine	99		
Up/Download durchführen	188, 194		
Up/Download-Auftrag			
Ablauf	97		
Up/Download-Kommando	36		
Upload	39		
Auftrag	68		
Ende	40		
Upload-Antwort	40		
Upload-Kommando			
Empfangen	96		
Uploadvorgang	40		



**be in motion**



Alle Angaben in dieser Betriebsanleitung sind unverbindliche Kundeninformationen, unterliegen einer ständigen Weiterentwicklung und werden fortlaufend durch unseren permanenten Änderungsdienst aktualisiert. Bitte beachten Sie, dass Angaben/Zahlen/Informationen aktuelle Werte zum Druckdatum sind.  
Zur Ausmessung, Berechnung und Kalkulationen sind diese Angaben nicht rechtlich verbindlich. Bevor Sie in dieser Betriebsanleitung aufgeführte Informationen zur Grundlage eigener Berechnungen und/oder Verwendungen machen, informieren Sie sich bitte, ob Sie den aktuellsten Stand der Informationen besitzen.  
Eine Haftung für die Richtigkeit der Informationen wird daher nicht übernommen.