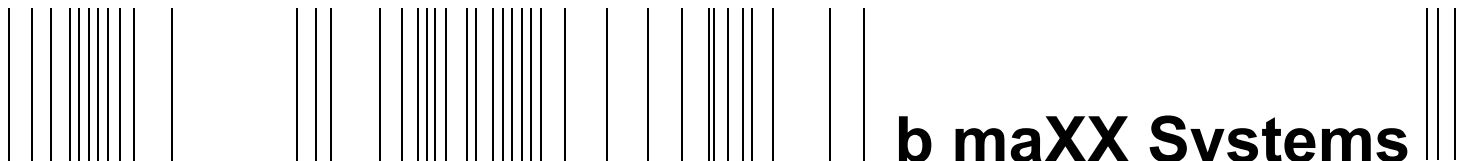


**be in motion be in motion**



**b maXX Systems**

**b maXX controller PLC**

**Applikationshandbuch**

**D**

5.04019.03



Titel	Applikationshandbuch
Produkt	<b>b maXX controller PLC</b> BMC-M-PLC-01/-02
Stand	15. Februar 2007
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 GmbH 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	Baumüller Nürnberg GmbH Ostendstr. 80 - 90 90482 Nürnberg Deutschland Tel. +49 9 11 54 32 - 0 Fax: +49 9 11 54 32 - 1 30 www.baumueller.de



<b>1</b>	<b>Einleitung</b>	<b>5</b>
1.1	Erste Schritte	5
1.2	Verwendete Begriffe	5
<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
2.2	Infozeichen	11
2.3	Rechtliche Hinweise	11
2.4	Bestimmungsgemäße Verwendung	11
2.5	Sachwidrige Verwendung	12
2.6	Schutzeinrichtungen	12
2.7	Ausbildung des Personals	13
2.8	Sicherheitsmaßnahmen im Normalbetrieb	13
2.9	Verpflichtung und Haftung	13
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>Inbetriebnahme</b>	<b>15</b>
3.1	Allgemeine Sicherheitsvorschriften	15
3.2	Anforderungen an das ausführende Personal	15
3.3	Beschreibung/Überprüfung der Sicherheits- und Überwachungseinrichtungen	16
3.4	Beschreibung und Überprüfung der Bedienungs- und Anzeigeelemente	16
3.4.1	LEDs zur Anzeige von Betriebszuständen der BMC-M-PLC-01	16
3.4.2	Schalter/Taster S1 zum Wechseln von Betriebszuständen der BMC-M-PLC-01	18
3.4.3	LEDs zur Anzeige von Betriebszuständen der BMC-M-PLC-02	19
3.4.4	Dreheschalter S1 der BMC-M-PLC-02	22
3.4.5	Schalter/Taster S2 zum Wechseln von Betriebszuständen der BMC-M-PLC-02	22
3.5	Ablauf der Inbetriebnahme	23
3.5.1	Einschalten der BMC-M-PLC-01	23
3.5.2	Testen der Funktion der BMC-M-PLC-01	24
3.5.3	Einschalten der BMC-M-PLC-02	25
3.5.4	Testen der Funktion der BMC-M-PLC-02	25
3.5.5	Bootprojekt senden	26
<b>4</b>	<b>Interne Kommunikationsschnittstelle zu den Systemkomponenten</b>	<b>31</b>
4.1	CBPB (Controller Based Parallel Bus)	31
4.2	I/O-Bus	31
4.3	Übertragungszeiten	32
<b>5</b>	<b>Betrieb</b>	<b>33</b>
5.1	Programmiersysteme PROPROG wt II und ProProg wt III	33
5.2	b maXX controller PLC Projekt	35
5.3	b maXX controller PLC Konfiguration	37
5.4	b maXX controller PLC Ressource	39
5.4.1	Kommunikation und Verbindung	40
5.4.2	Kontrolldialog für Ressourcen	44
5.4.3	Datenbereich	52
5.4.4	Die b maXX controller PLC Event-Tasks	56
5.5	b maXX controller PLC Anwenderbibliotheken	57
5.5.1	b maXX controller PLC Firmware	60
5.5.2	b maXX controller PLC Board Funktionen	61
5.5.3	Die b maXX controller PLC Datentypen	69



# Inhaltsverzeichnis

5.5.4	Die Standard Funktionsbaustein-Bibliotheken . . . . .	73
5.5.5	Die b maXX controller PLC Technologiebausteine . . . . .	74
5.5.6	Einfügen einer Anwenderbibliothek in ein Projekt . . . . .	75
5.6	CBPB-Systembeschreibung für die b maXX controller PLC . . . . .	77
5.6.1	Direkter Zugriff auf die CBPB-HW von der b maXX controller PLC aus . . . . .	77
5.6.2	Prozessdatenkommunikation zwischen b maXX controller PLC und Systemkomponenten, Synchronisierung auf eine gemeinsame CBPB-Hardware-Signalquelle . . . . .	79
5.6.3	Für die Applikation nutzbare Interruptquellen der b maXX controller PLC . . . . .	80
5.6.4	CBPB-Signalgenerierung über den Funktionsbaustein TIMER_A_INIT . . . . .	82
5.6.5	Funktionsbaustein "TIMER_A_INIT" . . . . .	83
5.6.6	Konfigurationsbeispiele . . . . .	94
5.6.7	Beispiel A: Einrichten eines CPU-Timer 1-Interrupts . . . . .	94
5.6.8	Beispiel B: Einrichten eines Board-Timer A Interrupts und Triggern des SYNC-Signal 195	95
5.6.9	Beispiel C: Realisierung einer einfachen seriellen RS485-Kommunikation mit Hilfe eines CPU-Timers . . . . .	99
5.6.10	Beschreibung der Bausteine für die serielle RS485-Kommunikation über die Schnittstelle X2 (9-pol. Sub-D-Buchse) auf der b maXX controller PLC . . . . .	101
5.6.11	Funktionsbaustein TER_INIT . . . . .	103
5.6.12	Funktionsbaustein TER_R . . . . .	108
5.6.13	Funktionsbaustein TER_S . . . . .	112
5.6.14	Funktionsbaustein T64_RSYN . . . . .	115
5.6.15	Funktionsbaustein T64_REC . . . . .	116
5.6.16	Funktionsbaustein TER_USS . . . . .	119
5.7	Codelaufzeiten . . . . .	126
5.7.1	Funktionsbaustein TIME_MEASURE_START . . . . .	128
5.7.2	Funktionsbaustein TIME_MEASURE_END . . . . .	129
5.8	Praktische Hinweise . . . . .	130
5.8.1	SPS-Fehler allgemein . . . . .	130
5.8.2	SPS-Fehler "Watchdog in Task 'x' überschritten!" . . . . .	130
5.8.3	SPS-Fehler "Globale Bereitmeldung (RST-Signal) fehlt! PLC gestoppt!" . . . . .	132
5.8.4	SPS-Fehler "Kein Zugriff auf den CX-Controller im Netzteil-Modul möglich!" . . . . .	132
5.8.5	SPS-Fehler "I/O-Bus Fehler aufgetreten! Fehler-Code:..." . . . . .	132
<b>6</b>	<b>Module (rechts der b maXX controller PLC) verwenden . . . . .</b>	<b>135</b>
6.1	ProMaster . . . . .	135
6.1.1	Anlegen eines ProProg wt III Projekts für die b maXX controller PLC . . . . .	136
6.1.2	Anlegen eines ProMaster Projektes mit b maXX controller PLC . . . . .	138
6.1.3	Konfigurierung des b maXX Systems mit ProMaster . . . . .	142
6.1.4	Konfigurierung der IOs mit ProMaster . . . . .	143
6.2	Modul Konfigurator ProModul . . . . .	150
6.2.1	Anlegen eines PROPROG wt II Projekts für die b maXX controller PLC . . . . .	151
6.2.2	Konfigurierung des b maXX Systems mit dem Modul Konfigurator . . . . .	152
6.3	Konfigurierung der Systemkomponenten / Module . . . . .	157
6.3.1	Digitale Eingänge / Digitale Ausgänge . . . . .	158
6.3.2	Analoge Eingänge / Analoge Ausgänge . . . . .	159
6.3.3	Alarm-Ausgang . . . . .	161
	<b>Anhang A - Abkürzungen . . . . .</b>	<b>163</b>
	<b>Revisionsübersicht . . . . .</b>	<b>165</b>
	<b>Index . . . . .</b>	<b>167</b>

# 1

## EINLEITUNG

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

Es wird vorausgesetzt, dass Sie die Betriebsanleitungen b maXX controller PLC und Netzteil für b maXX controller PLC verstanden und angewendet haben.

In diesem Kapitel beschreiben wir die ersten Schritte, die Sie nach Erhalt des Gerätes ausführen sollten. Wir definieren Begriffe, die in dieser Dokumentation durchgängig verwendet werden, und informieren Sie über Verpflichtungen, die beim Einsatz diese Gerätes beachtet werden müssen.

### 1.1 Erste Schritte

---

- ◆ Überprüfen Sie die Lieferung.
- ◆ Leiten Sie alle Unterlagen, die mit dem Modul geliefert wurden, an die entsprechenden Stellen in Ihrem Unternehmen weiter.
- ◆ Stellen Sie das geeignete Personal für Montage und Inbetriebnahme bereit.
- ◆ Übergeben Sie die Betriebsanleitung an das Personal und stellen Sie sicher, dass insbesondere die hier angegebenen Sicherheitshinweise verstanden und befolgt werden können.

### 1.2 Verwendete Begriffe

---

Für das Produkt b maXX controller PLC werden auch die Begriffe "PLC", "Modul" und "Modul b maXX controller PLC" verwendet. Weiterhin wird der Begriff „BMC-M-PLC-01“ bzw. „BMC-M-PLC-02“ verwendet.

Eine Liste der verwendeten Abkürzungen finden Sie in [▶Anhang A - Abkürzungen◀](#) ab Seite 163.

Für das Baumüller-Produkt "Netzteil für b maXX controller PLC" werden wir in dieser Dokumentation auch die Begriffe "Modul" und "Netzteil" verwenden.

Für die Kombination aus b maXX controller PLC und Netzteil für b maXX controller PLC (und gegebenenfalls weiteren Systemkomponenten/Modulen) wird auch der Begriff "b maXX System" verwendet.



# GRUNDLEGENDE SICHERHEITS- HINWEISE

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

In diesem Kapitel beschreiben wir Gefahren, die bei der Arbeit mit dem Baumüller-Modul 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

---



---

### **WARNUNG** (WARNING)

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

- schwere Körperverletzung
- Tod

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)

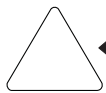
- erheblicher Sachschaden
- schwere Körperverletzung
- Tod - **kann** eintreffen

### VORSICHT (CAUTION)

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

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

- Sie dürfen das Modul nur an ein Netzteil für b maXX controller PLC anbauen.
- Projektieren Sie die Anwendung so, dass Sie das Modul immer innerhalb seiner Spezifikationen betreiben.
- Sorgen Sie dafür, dass ausschließlich qualifiziertes Personal mit diesem Modul arbeitet.
- Montieren Sie das Modul nur an das Netzteil für b maXX controller PLC.
- Installieren Sie das Modul so wie in es in dieser Dokumentation vorgegeben ist.
- Sorgen Sie dafür, dass die Anschlüsse immer den vorgegebenen Spezifikationen entsprechen.
- Betreiben Sie das Modul nur, wenn es technisch einwandfrei ist.
- Betreiben Sie das Modul immer in einer Umgebung, wie sie in den „Technischen Daten“ vorgeschrieben ist.
- Betreiben Sie das Modul immer in serienmäßigem Zustand.  
Aus Sicherheitsgründen dürfen Sie das Modul nicht umbauen.
- Beachten Sie alle diesbezüglichen Hinweise, falls Sie das Modul lagern.

Sie verwenden das Modul 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 Moduls 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 Modul an ein anderes Gerät/Modul als das Netzteil für b maXX controller PLC angebaut.
- Sie haben Hinweise dieser Betriebsanleitung missachtet.
- Sie haben das Modul nicht bestimmungsgemäß verwendet.
- Sie haben das Modul
  - 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 Modul umgebaut, ohne dass dies schriftlich von der Firma Baumüller Nürnberg GmbH genehmigt wurde.
- Sie haben die Anweisungen bezüglich Wartung in den Komponentenbeschreibungen nicht beachtet.
- Sie haben das Modul unsachgemäß mit Produkten anderer Hersteller kombiniert.
- Sie haben das b maXX System 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 GmbH gelten grundsätzlich. Diese stehen Ihnen spätestens seit Vertragsabschluss zur Verfügung.

### 2.6 Schutzeinrichtungen

---

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

Das Gehäuse des Moduls schützt in Schutzklasse IP20 das Modul vor Verschmutzung und Schäden durch statische Entladung bei Berührungen. Verwenden Sie niemals ein Modul mit beschädigtem Gehäuse.

## 2.7 Ausbildung des Personals



### WARNUNG (WARNING)

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

- schwere Körperverletzung
- Tod

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

Qualifiziertes Personal (Fachkräfte) wird folgendermaßen definiert:

#### Qualifiziertes Personal

Von der Firma Baumüller Nürnberg GmbH autorisierte Elektro-Ingenieure und Elektro-Fachkräfte des Kunden oder Dritter, die Installation und Inbetriebnahme von Baumüller-b maXX-Systemen 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 b maXX Systems 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 b maXX Systems die gültige Sicherheitsbestimmungen für die Anlage, in die dieses Gerät eingebaut ist.
- ▶ Versehen Sie das b maXX System mit zusätzlichen Überwachungs- und Schutzrichtungen, falls Sicherheitsbestimmungen dies fordern.
- ▶ Beachten Sie die Sicherheitsmaßnahmen für das b maXX System, in das das Modul eingebaut ist.

## 2.9 Verpflichtung und Haftung

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

## 2.9 Verpflichtung und Haftung

---

### 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

---

Das Modul wurde 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 und in der Betriebsanleitung.

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 GmbH 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 Fremdkörperereinwirkung bzw. höhere Gewalt

## INBETRIEBNAHME

In diesem Kapitel beschreiben wir, wie Sie das soeben montierte und installierte (siehe „Betriebsanleitung b maXX controller PLC“) Modul in Betrieb nehmen. Die Inbetriebnahme stellt sicher, dass das Modul richtig funktioniert.

Stellen Sie vor der Inbetriebnahme sicher, dass die folgenden Voraussetzungen erfüllt sind:

- 1 Modul ist korrekt montiert.
- 2 Modul ist korrekt installiert.
- 3 Alle Sicherheitsvorrichtungen sind in Betrieb gesetzt.
- 4 Das b maXX System ist einsatzbereit.

### 3.1 Allgemeine Sicherheitsvorschriften

---

► beachten Sie ► [Grundlegende Sicherheitshinweise](#) ◀ ab Seite 7.



#### GEFAHR (DANGER)

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

- schwere Körperverletzung
- Tod



Die Gefahr ist: **mechanische Einwirkung**. *Bei der Inbetriebnahme können sich Maschinenteile / Anlagenteile oder die gesamte Maschine / Anlage bewegen.*

Halten Sie genügend Abstand von sich bewegenden Maschinenteilen / Anlagenteilen bzw. von der sich bewegenden Maschine / Anlage. Beachten Sie, dass über die (an der b maXX controller PLC) angeschlossenen weiteren Module die Maschinenteile / Anlagenteile bzw. die Maschine / Anlage in Bewegung gesetzt werden können. Aktivieren Sie in jedem Fall deren Sicherheitseinrichtungen.

### 3.2 Anforderungen an das ausführende Personal

---

Die Arbeiten zur Inbetriebnahme dürfen nur von fachlich geschultem Personal, das insbesondere die Sicherheitsvorschriften und -hinweise versteht und befolgen kann, durchgeführt werden.

### 3.3 Beschreibung/Überprüfung der Sicherheits- und Überwachungseinrichtungen

Bevor Sie das Modul b maXX controller PLC in Betrieb nehmen, müssen Sie sicherstellen dass die +24 V DC Versorgung am Netzteil für b maXX controller PLC richtig angeschlossen ist und dass die +24 V DC Versorgung den Spezifikationen aus der Betriebsanleitung Netzteil für b maXX controller PLC entspricht. Erst nachdem Sie dieses überprüft und jenes sichergestellt haben, dürfen Sie mit der Inbetriebnahme fortfahren.

Beachten Sie bei der Inbetriebnahme, dass Sie die b maXX controller PLC nur zusammen mit dem Netzteil und gegebenenfalls weiteren Systemkomponenten in Betrieb nehmen können.

### 3.4 Beschreibung und Überprüfung der Bedienungs- und Anzeigeelemente

#### 3.4.1 LEDs zur Anzeige von Betriebszuständen der BMC-M-PLC-01

Das Modul b maXX controller PLC weist als Anzeigeelemente acht LEDs (vier grüne (H1, H3, H5, H7) und vier rote (H2, H4, H6, H8)) auf.

Die LEDs werden während der Initialisierung (Hochlaufphase) vom Betriebssystem der b maXX controller PLC verwendet.

Während des Betriebs (nach der Hochlaufphase des Betriebssystems) können die LEDs vom Anwender im Applikationsprogramm auf der b maXX controller PLC verwendet werden.

Alle Systemkomponenten im b maXX System müssen nach dem Einschalten der Versorgungsspannung einen gewissen internen Betriebszustand (globale Bereitmeldung) erreicht haben, bevor sie von der b maXX controller PLC angesprochen werden dürfen.

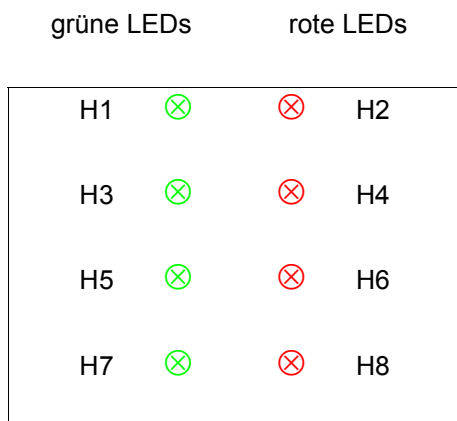


Abbildung 1: LEDs der b maXX controller PLC der BMC-M-PLC-01

#### 3.4.1.1 Einschalten und Initialisierung der BMC-M-PLC-01

- Nach dem Einschalten wartet das Modul b maXX controller PLC auf die globale Bereitmeldung folgender Module:
  - Modul Netzteil für b maXX controller PLC
  - Module die links der b maXX controller PLC gesteckt sind (z. B. Modul Ethernet mit CANopen-Master).



Diese Phase wird durch ein links rotierendes LED-Muster angezeigt. Dabei leuchten die vier oberen LEDs in der Reihenfolge H1 (grün) → H3 (grün) → H4 (rot) → H2 (rot) → H1 (grün) usw. im Wechsel von 500 ms auf.

Erfolgt die globale Bereitmeldung der Module nicht innerhalb von max. 8 s, wird H6 (rot) eingeschaltet.

Um die Fehlerursache zu beseitigen, siehe Betriebsanleitung b maXX controller PLC, Abschnitt Störungssuche und -beseitigung.

- Nach der globalen Bereitmeldung der Module initialisiert die b maXX controller PLC den I/O-Bus.

Diese Phase wird durch ein rechts rotierendes LED-Muster angezeigt. Dabei leuchten die vier oberen LEDs in der Reihenfolge H1 (grün) → H2 (rot) → H4 (rot) → H3 (grün) → H1 (grün) usw. im Wechsel von 500 ms auf.

Kann der I/O-Bus nicht innerhalb von max. 8 s initialisiert werden oder tritt ein Fehler bei der I/O-Bus Initialisierung auf, wird H8 (rot) eingeschaltet.

Die beiden soeben beschriebenen Abläufe können sehr schnell abgeschlossen sein, so dass man die zugehörigen Betriebsanzeigen nicht zwangsläufig beobachten muss.

Danach kann prinzipiell eine PROPROG-Kommunikation über die serielle RS232-Schnittstelle (X1) zwischen PC und der b maXX controller PLC erfolgen.

Ab jetzt ist auch die PROPROG-Kommunikation über TCP/IP möglich, wenn ein Modul mit Ethernet-Funktionalität (z. B. Modul Ethernet mit CANopen-Master) auf der linken Seite der b maXX controller PLC gesteckt ist und für die Kommunikation zur b maXX controller PLC konfiguriert wurde.

- Falls ein Bootprojekt vorhanden ist, wird jetzt das Bootprojekt geladen (Bootprojekt wird vom Flash gelesen, übersetzt und im SDRAM als ausführbarer Programmcode abgelegt). Das Laden des Bootprojektes wird durch ein schnelles Blinken der oberen beiden LEDs (H2 und H1) angezeigt.

#### HINWEIS



Falls die globale Bereitmeldung der Module fehlt (H6 (rot) ist eingeschaltet), kann kein Anwender-Programmcode ausgeführt werden; es wird eine PLC-Fehlermeldung erzeugt. Die PLC bleibt im Zustand „STOP“.

Die LEDs zeigen am Ende der Hochlaufphase folgende PLC-spezifische Betriebszustände an:

- Kein Projekt vorhanden, Zustand „POWER ON“:  
→ LED H3 (grün) und H4 (rot) leuchten.
- Projekt vorhanden, Zustand „STOP“:  
→ nur LED H4 (rot) leuchtet.
- Projekt vorhanden, Zustand „INIT“, die Steuerung befindet sich in der Kaltstart- bzw. Warmstartphase:  
→ nur LED H3 (grün) leuchtet.
- Projekt vorhanden, Zustand „RUN“:  
→ LED H1 (grün) und H3 (grün) leuchten.

## 3.4 Beschreibung und Überprüfung der Bedienungs- und Anzeigeelemente

- Globale Bereitmeldung fehlt:  
→ LED H6 (rot) leuchtet, PLC geht nicht in „RUN“.
- I/O-Bus lässt sich nicht initialisieren:  
→ LED H8 (rot) leuchtet.



### HINWEIS

Auch wenn die I/O-Bus Initialisierung nicht erfolgreich war (H8 (rot) ist eingeschaltet), wird das Bootprojekt geladen und der Anwender-Programmcode ausgeführt.

D. h. Sie können die b maXX controller PLC und die weiteren Module ohne die I/O-Bus Module verwenden.

### 3.4.1.2 Betrieb der BMC-M-PLC-01

Die LEDs zeigen am Ende der Hochlaufphase folgende PLC-spezifische Betriebszustände an:

- Kein Projekt vorhanden, Zustand „POWER ON“:  
→ LED H3 (grün) und H4 (rot) leuchten.
- Projekt vorhanden, Zustand „STOP“:  
→ nur LED H4 (rot) leuchtet.
- Projekt vorhanden, Zustand „INIT“, die Steuerung befindet sich in der Kaltstart- bzw. Warmstartphase:  
→ nur LED H3 (grün) leuchtet.
- Projekt vorhanden, Zustand „RUN“:  
→ LED H1 (grün) und H3 (grün) leuchten.
- Globale Bereitmeldung fehlt:  
→ LED H6 (rot) leuchtet, PLC geht nicht in „RUN“.
- I/O-Bus lässt sich nicht initialisieren:  
→ LED H8 (rot) leuchtet.

Im Zustand "RUN" können die acht LEDs vom Anwender frei programmiert werden. Zur Programmierung siehe "Applikationshandbuch b maXX controller PLC" im Kapitel "b maXX controller PLC Board Funktionen / Funktionsbaustein LED8"



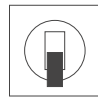
### HINWEIS

Auch wenn die I/O-Bus Initialisierung nicht erfolgreich war (H8 (rot) ist eingeschaltet), wird das Bootprojekt geladen und der Anwender-Programmcode ausgeführt.

D. h. Sie können die b maXX controller PLC und die weiteren Module ohne die I/O-Bus Module verwenden.

### 3.4.2 Schalter/Taster S1 zum Wechseln von Betriebszuständen der BMC-M-PLC-01

Das Modul b maXX controller PLC hat zum Ändern der Betriebszustände den Schalter/Taster S1.



Taster nach oben:	RESET
Schalter Mitte:	STOP
Schalter unten:	RUN

Abbildung 2: Schalter S1 auf Frontplatte BMC-M-PLC-01

### HINWEIS



Das Anwenderprojekt kann nur dann anlaufen, wenn der Schalter/Taster S1 in der unteren Stellung "RUN" steht und alle Module ihre globale Bereitmeldung an die b maXX controller PLC gemeldet haben (H6 (rot) ist ausgeschalten).

Mit dem Taster nach oben wird ein Reset für das Modul "b maXX controller PLC", für das Modul "Netzteil für b maXX controller PLC" und für die weiteren Systemkomponenten die links der b maXX controller PLC oder rechts des Netzteils gesteckt sind, durchgeführt.

### GEFAHR (DANGER)



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

- schwere Körperverletzung
- Tod



Die Gefahr ist: **mechanische Einwirkung**. *Bei der Inbetriebnahme der b maXX controller PLC und des an der b maXX controller PLC angeschlossenen Netzteils (sowie weiterer angeschlossener Systemkomponenten) kann durch ein komplett vorliegendes Applikationsprogramm die Maschine/Anlage bzw. Teile der Maschine/Anlage gestartet werden. Durch das nicht korrekte Einstellen einer Adresse am Netzteil (siehe > Betriebsanleitung Netzteil für b maXX controller PLC <) kann sich die Maschine/Anlage bzw. Teile der Maschine/Anlage unerwartet verhalten.*

Halten Sie genügend Abstand von sich bewegenden Maschinenteilen / Anlagenteilen bzw. von der sich bewegenden Maschine / Anlage. Beachten Sie, dass über die (an das Modul b maXX controller PLC) angeschlossenen weiteren Module die Maschinenteile / Anlagenteile bzw. die Maschine / Anlage in Bewegung gesetzt werden können. Aktivieren Sie in jedem Fall deren Sicherheitseinrichtungen.

### 3.4.3 LEDs zur Anzeige von Betriebszuständen der BMC-M-PLC-02

Das Modul b maXX controller PLC weist als Anzeigeelemente zwölf LEDs (fünf grüne (H1, H3, H5, H7, H11), fünf rote (H2, H4, H6, H8, H10), eine blaue (H9) und eine orange/grüne (H12)) auf.

Die LEDs werden während der Initialisierung (Hochlaufphase) vom Betriebssystem der b maXX controller PLC verwendet.

Während des Betriebs (nach der Hochlaufphase des Betriebssystems) können die LEDs vom Anwender im Applikationsprogramm auf der b maXX controller PLC verwendet werden.

Alle Systemkomponenten im b maXX System müssen nach dem Einschalten der Versorgungsspannung einen gewissen internen Betriebszustand (globale Bereitmeldung) erreicht haben, bevor sie von der b maXX controller PLC angesprochen werden dürfen.

### LEDs

H1 ⊗ grün	⊗ rot	⊗ grün	H4 ⊗ rot
H5 ⊗ grün	⊗ rot	⊗ grün	H8 ⊗ rot
H9 ⊗ blau	⊗ rot	⊗ grün	H12 ⊗/⊗ grün/ orange

Abbildung 3: LEDs der b maXX controller PLC der BMC-M-PLC-02

#### 3.4.3.1 Einschalten und Initialisierung der BMC-M-PLC-02

- Nach dem Einschalten erfolgt eine Basisinitialisierung der b maXX controller PLC, die für die weitere Initialisierung notwendig ist. Am Ende dieser Basisinitialisierung wird H9 (blau) eingeschaltet.
- Anschließend wartet das Modul b maXX controller PLC auf die globale Bereitmeldung folgender Module:
  - Modul Netzteil für b maXX controller PLC
  - Module die links der b maXX controller PLC gesteckt sind (z. B. Modul Ethernet mit CANopen-Master).

Diese Phase wird durch ein links rotierendes LED-Muster angezeigt. Dabei leuchten vier LEDs in der Reihenfolge H1 (grün) → H5 (grün) → H6 (rot) → H2 (rot) → H1 (grün) usw. im Wechsel von 500 ms auf. H9 (blau) ist wieder aus.

Erfolgt die globale Bereitmeldung der Module nicht innerhalb von max. 8 s, wird H4 (rot) eingeschaltet.

Um die Fehlerursache zu beseitigen, siehe Betriebsanleitung b maXX controller PLC, Abschnitt Störungssuche und -beseitigung.

- Nach der globalen Bereitmeldung der Module initialisiert die b maXX controller PLC den I/O-Bus.

Diese Phase wird durch ein rechts rotierendes LED-Muster angezeigt. Dabei leuchten vier LEDs in der Reihenfolge H1 (grün) → H2 (rot) → H6 (rot) → H5 (grün) → H1 (grün) usw. im Wechsel von 500 ms auf.

Kann der I/O-Bus nicht innerhalb von max. 8 s initialisiert werden oder tritt ein Fehler bei der I/O-Bus Initialisierung auf, wird H8 (rot) eingeschaltet.

Die beiden soeben beschriebenen Abläufe können sehr schnell abgeschlossen sein, so dass man die zugehörigen Betriebsanzeigen nicht zwangsläufig beobachten muss.

Danach kann prinzipiell eine PROPROG-Kommunikation über die serielle RS232-Schnittstelle (X1) zwischen PC und der b maXX controller PLC erfolgen.

Ab jetzt ist auch die PROPROG-Kommunikation über TCP/IP möglich, wenn ein Modul mit Ethernet-Funktionalität (z. B. Modul Ethernet mit CANopen-Master) auf der linken Seite der b maXX controller PLC gesteckt ist und für die Kommunikation zur b maXX controller PLC konfiguriert wurde.

- Falls ein Bootprojekt vorhanden ist, wird jetzt das Bootprojekt geladen (Bootprojekt wird vom Flash gelesen, übersetzt und im SDRAM als ausführbarer Programmcode abgelegt). Das Laden des Bootprojektes wird durch ein schnelles Blinken der LEDs H1 und H2 angezeigt.



#### HINWEIS

Falls die globale Bereitmeldung der Module fehlt (H4 (rot) ist eingeschaltet), kann **kein** Anwender-Programmcode ausgeführt werden; es wird eine PLC-Fehlermeldung erzeugt. Die PLC bleibt im Zustand „STOP“.

Die LEDs zeigen am Ende der Hochlaufphase folgende PLC-spezifische Betriebszustände an:

- Kein Projekt vorhanden, Zustand „POWER ON“:  
→ LED H5 (grün) und H6 (rot) leuchten.
- Projekt vorhanden, Zustand „STOP“:  
→ nur LED H6 (rot) leuchtet.
- Projekt vorhanden, Zustand „INIT“, die Steuerung befindet sich in der Kaltstart- bzw. Warmstartphase:  
→ nur LED H5 (grün) leuchtet.
- Projekt vorhanden, Zustand „RUN“:  
→ LED H1 (grün) und H5 (grün) leuchten.
- globale Bereitmeldung fehlt:  
→ LED H4 (rot) leuchtet, PLC geht **nicht** in „RUN“.
- I/O-Bus lässt sich nicht initialisieren:  
→ LED H8 (rot) leuchtet.



#### HINWEIS

Auch wenn die I/O-Bus Initialisierung nicht erfolgreich war (H8 (rot) ist eingeschaltet), wird das Bootprojekt geladen und der Anwender-Programmcode ausgeführt.

D. h. Sie können die b maXX controller PLC und die weiteren Module ohne die I/O-Bus Module verwenden.

### 3.4.3.2 Betrieb der BMC-M-PLC-02

Die LEDs zeigen am Ende der Hochlaufphase folgende PLC-spezifische Betriebszustände an:

- Kein Projekt vorhanden, Zustand „POWER ON“:  
→ LED H5 (grün) und H6 (rot) leuchten.

## 3.4 Beschreibung und Überprüfung der Bedienungs- und Anzeigeelemente

- Projekt vorhanden, Zustand „STOP“:  
→ nur LED H6 (rot) leuchtet.
- Projekt vorhanden, Zustand „INIT“, die Steuerung befindet sich in der Kaltstart- bzw. Warmstartphase:  
→ nur LED H5 (grün) leuchtet.
- Projekt vorhanden, Zustand „RUN“:  
→ LED H1 (grün) und H5 (grün) leuchten.
- globale Bereitmeldung fehlt:  
→ LED H4 (rot) leuchtet, PLC geht **nicht** in „RUN“.
- I/O-Bus lässt sich nicht initialisieren:  
→ LED H8 (rot) leuchtet.

Im Zustand "RUN" können die zwölf LEDs vom Anwender frei programmiert werden. Zur Programmierung siehe "Applikationshandbuch b maXX controller PLC" im Kapitel "b maXX controller PLC Board Funktionen / Funktionsbaustein LED12"



### HINWEIS

Auch wenn die I/O-Bus Initialisierung nicht erfolgreich war (H8 (rot) ist eingeschaltet), wird das Bootprojekt geladen und der Anwender-Programmcode ausgeführt.

D. h. Sie können die b maXX controller PLC und die weiteren Module ohne die I/O-Bus Module verwenden.

### 3.4.4 Drehschalter S1 der BMC-M-PLC-02

Der Drehschalter S1 der BMC-M-PLC-02 ist für zukünftige Entwicklungen vorgesehen und wird zur Zeit nicht verwendet.

### 3.4.5 Schalter/Taster S2 zum Wechseln von Betriebszuständen der BMC-M-PLC-02

Das Modul b maXX controller PLC hat zum Ändern der Betriebszustände den Schalter/Taster S2.



Taster nach oben:	RESET
Schalter Mitte:	STOP
Schalter unten:	RUN

Abbildung 4: Schalter S2 auf Modul b maXX controller PLC der BMC-M-PLC-02



### HINWEIS

Das Anwenderprojekt kann nur dann anlaufen, wenn der Schalter/Taster S2 in der unteren Stellung "RUN" steht und alle Module ihre globale Bereitmeldung an die b maXX controller PLC gemeldet haben (H4 (rot) ist ausgeschalten).

Mit dem Taster nach oben wird ein Reset nur für das Modul "b maXX controller PLC" und für das Modul "Netzteil für b maXX controller PLC" durchgeführt, nicht für die weiteren Systemkomponenten (d. h. es wird kein Reset für die Module die links der b maXX controller PLC oder rechts des Netzteils gesteckt sind durchgeführt).



### GEFAHR (DANGER)

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

- schwere Körperverletzung
- Tod

Die Gefahr ist: **mechanische Einwirkung**. *Bei der Inbetriebnahme der b maXX controller PLC und des an der b maXX controller PLC angeschlossenen Netzteils (sowie weiterer angeschlossener Systemkomponenten) kann durch ein komplett vorliegendes Applikationsprogramm die Maschine/Anlage bzw. Teile der Maschine/Anlage gestartet werden. Durch das nicht korrekte Einstellen einer Adresse am Netzteil (siehe ► Betriebsanleitung Netzteil für b maXX controller PLC ◄) kann sich die Maschine/Anlage bzw. Teile der Maschine/Anlage unerwartet verhalten.*

Halten Sie genügend Abstand von sich bewegenden Maschinenteilen / Anlagenteilen bzw. von der sich bewegenden Maschine / Anlage. Beachten Sie, dass über die (an das Modul b maXX controller PLC) angeschlossenen weiteren Module die Maschinenteile / Anlagenteile bzw. die Maschine / Anlage in Bewegung gesetzt werden können. Aktivieren Sie in jedem Fall deren Sicherheitseinrichtungen.

## 3.5 Ablauf der Inbetriebnahme

Die Inbetriebnahme gliedert sich in folgende Abschnitte:

- 1 Einschalten.
- 2 Testen der Funktion.

### 3.5.1 Einschalten der BMC-M-PLC-01

- Lesen und beachten Sie die ► [Allgemeine Sicherheitsvorschriften](#) ◄ ab Seite 15.
- Die Montage und Installation muss korrekt durchgeführt worden sein (siehe hierzu ► [Betriebsanleitung b maXX controller PLC BMC-M-PLC-01/-02](#) ◄).
- Stellen Sie den Schalter/Taster S1 an der b maXX controller PLC auf "STOP" (Mittelstellung).
- Schalten Sie die +24 V DC der Spannungsversorgung ein.



### HINWEIS

Das Modul b maXX controller PLC dürfen Sie nicht mit den anderen Systemkomponenten verbinden oder von den anderen Systemkomponenten trennen, wenn die +24 V DC der Spannungsversorgung des Netzteils eingeschaltet ist. Schalten Sie vorher die +24 V DC der Spannungsversorgung des Netzteils aus.

### 3.5.2 Testen der Funktion der BMC-M-PLC-01

- Nach dem Einschalten des b maXX Systems können zwei Zustände auftreten:
  - Kein Bootprojekt (= kein Anwenderprojekt auf der b maXX controller PLC) vorhanden:  
Die LED H2 (rot) leuchtet kurz auf und anschließend leuchten die LEDs H3 (grün) und H4 (rot) dauerhaft auf.  
Dies bedeutet, dass kein Projekt auf der b maXX controller PLC vorhanden ist.  
In diesem Zustand "POWER ON" wartet die b maXX controller PLC auf PROPROG-Kommunikation
  - Bootprojekt vorhanden:  
Beim Einschalten wird das Bootprojekt geladen. Dabei blinken die oberen LEDs (H1 und H2). Nach kurzer Zeit leuchtet die LED H4 (rot) auf. Die b maXX controller PLC befindet sich im Zustand "STOP".
- Solange der Schalter/Taster S1 an der b maXX controller PLC auf "STOP" (Mittelstellung) steht, kann ein vorhandenes Bootprojekt nicht anlaufen.  
Wenn Sie ein vorhandenes Bootprojekt starten wollen, indem Sie den Schalter S1 an der b maXX controller PLC auf "RUN" (untere Stellung) stellen, vergewissern Sie sich **vorher**, dass das für Ihre Anwendung **richtige** Bootprojekt für **diese** Anlage in **dieser** b maXX controller PLC eingespielt ist und dass die **richtige** Adresse am Netzteil (sofern Sie diese in Ihrem Projekt ausgewertet wird; siehe hierzu auch [►Direkter Zugriff auf die CBPB-HW von der b maXX controller PLC aus◄](#) ab Seite 77) eingestellt ist!

Weitere Informationen, wie Sie dies sicherstellen können entnehmen Sie diesem Applikationshandbuch b maXX controller PLC. Wie Sie z. B. das "Senden eines Bootprojekts" zur b maXX controller PLC durchführen können, entnehmen Sie [►Bootprojekt senden◄](#) ab Seite 26.



**GEFAHR (DANGER)**

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

- schwere Körperverletzung
- Tod

Die Gefahr ist: **mechanische Einwirkung**. *Ein auf der b maXX controller PLC vorhandenes Bootprojekt läuft an, wenn der Schalter/Taster S1 von "STOP" (Mittelstellung) auf "RUN" (untere Stellung) gestellt wird oder wenn beim Einschalten des b maXX Systems der Schalter/Taster S1 auf "RUN" gestellt ist. Das Bootprojekt kann so programmiert sein, dass die Maschine/Anlage bzw. Teile der Maschine/Anlage bewegt werden. Durch das nicht korrekte Einstellen einer Adresse am Netzteil (siehe > Betriebsanleitung Netzteil für b maXX controller PLC <) kann sich die Maschine/Anlage bzw. Teile der Maschine/Anlage unerwartet verhalten.*

Halten Sie genügend Abstand von sich bewegenden Maschinenteilen / Anlagenteilen bzw. von der sich bewegenden Maschine / Anlage. Beachten Sie, dass über die (an die b maXX controller PLC) angeschlossenen weiteren Module die Maschinenteile / Anlagenteile bzw. die Maschine / Anlage in Bewegung gesetzt werden können. Aktivieren Sie in jedem Fall deren Sicherheitseinrichtungen.

**3.5.3 Einschalten der BMC-M-PLC-02**

- Lesen und beachten Sie die [▶Allgemeine Sicherheitsvorschriften<](#) ab Seite 15.
- Die Montage und Installation muss korrekt durchgeführt worden sein (siehe hierzu [▶Betriebsanleitung b maXX controller PLC BMC-M-PLC-01/-02<](#)).
- Stellen Sie den Schalter/Taster S2 an der b maXX controller PLC auf "STOP" (Mittelstellung).
- Schalten Sie die +24 V DC der Spannungsversorgung ein.

**HINWEIS**

Das Modul b maXX controller PLC dürfen Sie nicht mit den anderen Systemkomponenten verbinden oder von den anderen Systemkomponenten trennen, wenn die +24 V DC der Spannungsversorgung des Netzteils eingeschaltet ist. Schalten Sie vorher die +24 V DC der Spannungsversorgung des Netzteils aus.

**3.5.4 Testen der Funktion der BMC-M-PLC-02**

- Nach dem Einschalten des b maXX Systems können zwei Zustände auftreten:
  - Kein Bootprojekt (= kein Anwenderprojekt auf der b maXX controller PLC) vorhanden:  
Die LED H2 (rot) leuchtet kurz auf und anschließend leuchten die LEDs H5 (grün) und H6 (rot) dauerhaft auf.  
Dies bedeutet, dass kein Projekt auf der b maXX controller PLC vorhanden ist.  
In diesem Zustand "POWER ON" wartet die b maXX controller PLC auf PROPROG-Kommunikation
  - Bootprojekt vorhanden:  
Beim Einschalten wird das Bootprojekt geladen. Dabei blinken die oberen LEDs.

## 3.5 Ablauf der Inbetriebnahme

Nach kurzer Zeit leuchtet die LED H6 (rot) auf. Die b maXX controller PLC befindet sich im Zustand "STOP".

- Solange der Schalter/Taster S2 an der b maXX controller PLC auf "STOP" (Mittelstellung) steht, kann ein vorhandenes Bootprojekt nicht anlaufen.  
Wenn Sie ein vorhandenes Bootprojekt starten wollen, indem Sie den Schalter S2 an der b maXX controller PLC auf "RUN" (untere Stellung) stellen, vergewissern Sie sich **vorher**, dass das für Ihre Anwendung **richtige** Bootprojekt für **diese** Anlage in **dieser** b maXX controller PLC eingespielt ist und dass die **richtige** Adresse am Netzteil (sofern Sie diese in Ihrem Projekt ausgewertet wird) eingestellt ist!

Weitere Informationen, wie Sie dies sicherstellen können entnehmen Sie diesem Applikationshandbuch b maXX controller PLC. Wie Sie z. B. das "Senden eines Bootprojekts" zur b maXX controller PLC durchführen können, entnehmen Sie [►Bootprojekt senden◄](#) ab Seite 26.

### 3.5.5 Bootprojekt senden

- 1 Verbinden Sie die serielle Schnittstelle Ihres PC mit der Sub-D-Buchse auf der **b maXX controller PLC (X1)**.
- 2 Starten Sie auf Ihrem PC die Bediensoftware PROPROG wt II (oder ProProg wt III) und wählen Sie ein Projekt für b maXX controller PLC aus. Dieses Projekt muss bereits erfolgreich kompiliert sein. Sie können das Projekt jetzt auch mit „Code\Projekt neu erzeugen“ erneut kompilieren.
- 3 Klicken Sie im Bedienprogramm auf „Ressourcekontrolle“ im Verzeichnis „Online“.

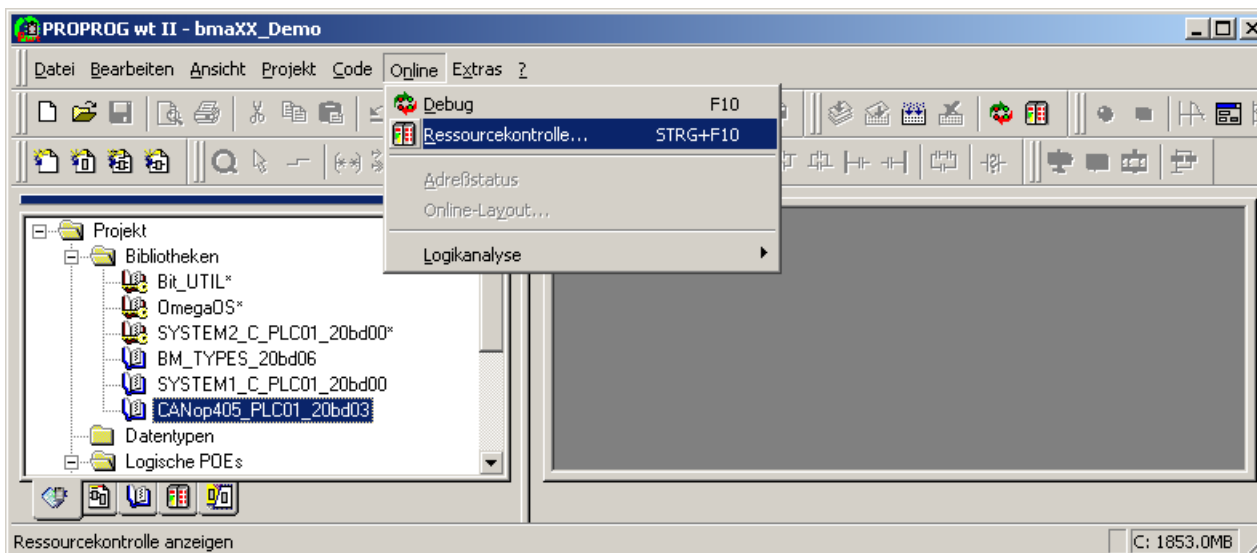


Abbildung 5: PROPROG wt II Bedienoberfläche

Es öffnet sich folgendes Fenster:



Abbildung 6: Ressourcekontrolle „Ein“

4 Drücken Sie den Button „Senden“. Es erscheint folgendes Bild:



Abbildung 7: Senden

5 Klicken Sie „Bootprojekt Senden“ an.

- Phase 1: Flash löschen:  
Ein eventuell vorhandenes Bootprojekt wird gelöscht. Am Bedienprogramm erscheint unten am Bildschirm ein weißer Balken. An der b maXX controller PLC-01 blinken die LEDs H1 bis H8 gleichzeitig. An der b maXX controller PLC-02 blinken die LEDs H1 bis H12 gleichzeitig.
- Phase 2: Projekt senden:  
Das Bootprojekt wird ins Flash übertragen. Der Balken unten am Bildschirm wird blau und zeigt den Fortschritt der Übertragung an. An der b maXX controller PLC-01 blinken die LEDs H1 (grün), H2 (rot), H3 (grün) und H4 (rot) während der Übertragung diagonal. An der b maXX controller PLC-02 blinken die LEDs H1 (grün), H2 (rot), H5 (grün) und H6 (rot) während der Übertragung diagonal.



### HINWEIS

Nachdem der Sendevorgang abgeschlossen wurde, sollte das b maXX System aus- und wieder eingeschaltet werden, damit die b maXX controller PLC die globale Bereitmeldung aller weiteren Module (Module die links der PLC gesteckt sind und Modul Netzteil für b maXX controller PLC) ordnungsgemäß erkennt.

Alternativ kann es jedoch auch genügen, nur das Projekt auf der b maXX controller PLC entweder über „Senden“ / „Bootprojekt Aktivieren“ und „Kaltstart“ oder über einen Reset an der b maXX controller PLC zu starten. Diese Vorgehensweise ist jedoch in hohem Maße applikationsabhängig und liegt somit im Verantwortungsbereich des Inbetriebnehmers.

Die Ausführung eines Projekts auf der b maXX controller PLC kann mit der Ressourcekontrolle gestoppt bzw. neu gestartet werden. Alternativ kann dies auch über den Schalter S1 (BMC-M-PLC-01) bzw. S2 (BMC-M-PLC-02) auf der b maXX controller PLC erreicht werden.



### HINWEIS

Der Inbetriebnehmer sollte sich bezüglich dem Gesamtverhalten seines b maXX Systems jedoch immer bewusst sein, dass im Zustand „Stop“ kein Applikationsprojekt ausgeführt wird, also auch keine Kommunikation mehr zu den Modulen stattfinden kann.



Abbildung 8: Ressourcekontrolle „Betrieb“

Wenn Sie den Button „Info“ drücken erhalten Sie das folgende Bild mit Informationen:

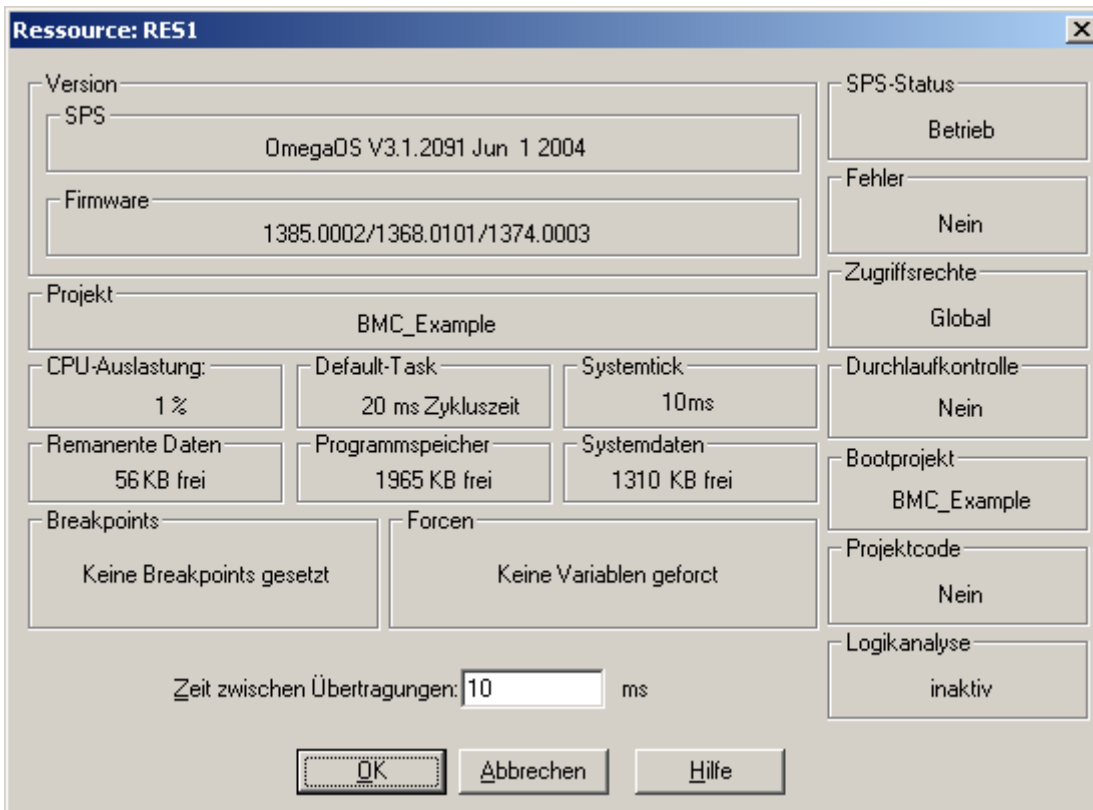


Abbildung 9: Informationsfenster Ressource



### HINWEIS

Bedeutung der angezeigten Versions-Strings im Info-Fenster:

SPS-String: "OmegaOS V3.1.2091 Jun 1 2004"  
 -> "OmegaOS": Produktbezeichnung  
 -> "V3.1.2091": OmegaOS Compiler Info  
 -> "Jun 1 2004": Datum, an dem die SPS-Version erzeugt wurde.

Firmware-String: "1385.0002/1368.0101/1374.0003"  
 -> "1385.0002": Version OmegaOS SW 6.1385.0002  
 -> "1368.0101": Version FPGA SW 6.1368.0101  
 -> "1374.0003": Version BOOT SW 6.1374.0003"

Im Rahmen der technischen Weiterentwicklung der Produkte können hier auch höhere Versionen mit jüngerem Datum installiert sein.

Quittieren Sie mit „OK“ und kehren Sie zur Ressourcekontrolle zurück (siehe [Abbildung 6](#) auf Seite 27).



# INTERNE KOMMUNIKATIONS- SCHNITTSTELLE ZU DEN SYSTEM- KOMPONENTEN

Die Kommunikation zu den Systemkomponenten (Modulen) die links der b maXX controller PLC gesteckt sind erfolgt über CBPB (Controller Based Parallel Bus)

Die Kommunikation zu den Systemkomponenten die rechts der b maXX controller PLC gesteckt sind erfolgt über den I/O-Bus.

## 4.1 CBPB (Controller Based Parallel Bus)

---

Der CBPB ist eine interne Kommunikationsschnittstelle für den Datenaustausch zwischen b maXX controller PLC und den Systemkomponenten die links der b maXX controller PLC gesteckt sind.

Die Zugriffe auf die Systemkomponenten (die links der b maXX controller PLC gesteckt sind) stellen sich für den Anwender als Zugriffe auf das DPRAM der jeweiligen Systemkomponente dar.

Eine Parametrierung der Schnittstelle ist nicht notwendig.

Anleitungen zur Programmierung und/oder Parametrierung der jeweiligen Systemkomponenten erhalten sie in den Applikationshandbüchern der jeweiligen Systemkomponenten und in der Online-Hilfe des b maXX controller Konfigurator.

## 4.2 I/O-Bus

---

Der I/O-Bus ist eine interne Kommunikationsschnittstelle für den Datenaustausch zwischen b maXX controller PLC und den Systemkomponenten / Modulen die rechts der b maXX controller PLC gesteckt sind.

Die Zugriffe auf die Systemkomponenten (die rechts der b maXX controller PLC gesteckt sind) stellen sich für den Anwender als Zugriffe auf die Ein- und Ausgänge der jeweiligen Systemkomponente dar.

Eine Parametrierung der Schnittstelle ist nicht notwendig.

### 4.3 Übertragungszeiten

Die Ein- und Ausgänge stehen als Merkerabbild in der PLC zur Verfügung. Standardmäßig erfolgt die Datenübertragung von und zum IO-Bus automatisch mit dem Zyklus des eingestellten Systemticks (siehe [b maXX controller PLC Ressource](#) ab Seite 39).

Wird in der IEC-Applikation eine Eingangsvariable auf eine Ausgangsvariable kopiert, ergibt sich eine maximale Übertragungszeit  $t_{\max}$ , bis sich eine Änderung des digitalen Eingangs am digitalen Ausgang auswirkt.

$$t_{\max} = t_{id} + 2 \cdot t_{st} + t_{cyc} + (t_{IO} + 10\%)$$

wobei	$t_{id}$	Eingangsverzögerung des digitalen Eingangs
	$t_{st}$	Systemtick
	$t_{cyc}$	Aufrufintervall mit dem der Kopierbefehl zyklisch aufgerufen wird
	$t_{IO}$	IO-Bus-Zykluszeit

Soll die Datenübertragung von und zum IO-Bus auf ein zyklisches Ereignis synchronisiert werden, ist dies mit Hilfe des Bausteins IO\_BUS\_CYCLE möglich. In diesem Fall wird die Datenübertragung zu dem Zeitpunkt gestartet, an dem der Baustein aufgerufen wird. Die maximale Übertragungszeit von einem digitalen Eingang zu einem digitalen Ausgang ist dann wie folgt:

$$t_{\max} = t_{id} + 3 \cdot t_{cyc} + (t_{IO} + 10\%)$$

wobei	$t_{id}$	Eingangsverzögerung des digitalen Eingangs
	$t_{cyc}$	Aufrufintervall mit dem der Kopierbefehl zyklisch aufgerufen wird
	$t_{IO}$	IO-Bus-Zykluszeit

Die IO-Bus-Zykluszeit (in us) für die bestehende Konfiguration des IO-Busses kann aus dem Merker (UINT) %MW3.110036 ausgelesen werden.

Beispiele: IO-Bus mit 1\*DI8000 und 1\*DO8000

Der Kopierbefehl befindet sich in einem 4 ms Timerinterrupt.

1. Standard-Datenübertragung

$t_{id}$	3 ms	Eingangsverzögerung des digitalen Eingangs DI8000
$t_{st}$	10 ms	Systemtick (PLC-01)
$t_{cyc}$	4 ms	Aufrufintervall für den Kopierbefehl
$t_{IO}$	422 µs	IO-Bus-Zykluszeit (aus %MW3.110036)

$$t_{\max} = 3 \text{ ms} + 2 \cdot 10 \text{ ms} + 4 \text{ ms} + (0,422 \text{ ms} + 0,042 \text{ ms}) = 27,464 \text{ ms}$$

2. Datenübertragung mit dem Baustein IO\_BUS\_CYCLE im gleichen 4 ms Timerinterrupt wie der Kopierbefehl.

$t_{id}$	3 ms	Eingangsverzögerung des digitalen Eingangs DI8000
$t_{cyc}$	4 ms	Aufrufintervall für den Kopierbefehl
$t_{IO}$	422 µs	IO-Bus-Zykluszeit (aus %MW3.110036)

$$t_{\max} = 3 \text{ ms} + 3 \cdot 4 \text{ ms} + (0,422 \text{ ms} + 0,042 \text{ ms}) = 15,464 \text{ ms}$$



# BETRIEB

## 5.1 Programmiersysteme PROPROG wt II und ProProg wt III

---

PROPROG wt ist ein auf der Norm IEC 61131-3 basierendes Standard-Programmiersystem.

Das Programmiersystem verfügt über leistungsstarke Funktionen für die unterschiedlichen Entwicklungsphasen von SPS-Anwendungen wie:

- Editieren
- Kompilieren
- Debuggen
- Drucken

Das Programmiersystem PROPROG wt basiert auf einer modernen 32-Bit-Windows-Technologie und ermöglicht dem Benutzer eine einfache Bedienung durch Werkzeuge wie Zoom, Scrollen, spezielle Symbolleisten, Drag & Drop, einen Shortcut-Manager und andockbare Fenster.

Insbesondere ermöglicht das System die Bearbeitung mehrerer Konfigurationen und Ressourcen innerhalb eines Projektes sowie das Einbinden von Projektbibliotheken. Des Weiteren verfügt es über ein leistungsstarkes System zum Debuggen. Mit Hilfe des sehr einfach zu bedienenden Projektbaum-Editors können Projekte angezeigt und editiert werden. Die komplexe Struktur der Norm IEC 61131-3 wird so einfach und transparent wie möglich dargestellt. Dies ermöglicht dem Benutzer auf einfache Weise POEs, Datentypen, Bibliotheken und Konfigurationselemente in den Projektbaum einzufügen und zu editieren.

Das Programmiersystem PROPROG wt besteht aus einem SPS-unabhängigen Kern zum Programmieren in den verschiedenen IEC Programmiersprachen: dazu gehören die Textsprachen Strukturierter Text (ST) und Anweisungsliste (AWL) sowie die graphischen Sprachen Funktionsbaustein-Sprache (FBS), Kontaktplan (KOP) und Ablaufsprache (AS).

Zum Programmieren in den einzelnen Sprachen steht jeweils ein Editor-Assistent zur Verfügung, der ein schnelles und sehr einfaches Einfügen von vorbereiteten Schlüsselwörtern, Anweisungen, Operatoren, Funktionen und Funktionsbausteinen in den einzelnen Arbeitsblättern ermöglicht. Der Editor-Assistent kann auch zum Deklarieren von Variablen und Datentypen verwendet werden. Der unabhängige Kern des Systems wird durch spezielle Teile vervollständigt, die den verschiedenen SPSen angepasst sind.

Das neue, einfache Online-Handling und die 32-Bit-Simulation bieten Möglichkeiten zum Debuggen des Adressstatus und eine Multitask-Testumgebung in Echtzeit.

Ein einfach zu bedienendes Werkzeug für die Projektdokumentation ermöglicht das Drucken des Projektes in zeitsparender optimierter Form (mit weniger Papier) mit einem Seitenlayout, das vom Benutzer festgelegt werden kann.



### GEFAHR (DANGER)

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

- schwere Körperverletzung
- Tod

Die Gefahr ist: **mechanische Einwirkung**. *Ein auf der b maXX controller PLC-01 vorhandenes Bootprojekt läuft an, wenn der Schalter/Taster S1 von "STOP" (Mittelstellung) auf "RUN" (untere Stellung) gestellt wird oder wenn beim Einschalten des b maXX Systems der Schalter/Taster S1 auf "RUN" gestellt ist. Ein auf der b maXX controller PLC-02 vorhandenes Bootprojekt läuft an, wenn der Schalter/Taster S2 von "STOP" (Mittelstellung) auf "RUN" (untere Stellung) gestellt wird oder wenn beim Einschalten des b maXX Systems der Schalter/Taster S2 auf "RUN" gestellt ist. Das Bootprojekt kann so programmiert sein, dass die Maschine/Anlage bzw. Teile der Maschine/Anlage bewegt werden. Durch das nicht korrekte Einstellen einer Adresse am Netzteil (siehe „Betriebsanleitung Netzteil für b maXX controller PLC“) kann sich die Maschine/Anlage bzw. Teile der Maschine/Anlage unerwartet verhalten.*

Halten Sie genügend Abstand von sich bewegenden Maschinenteilen / Anlagenteilen bzw. von der sich bewegenden Maschine / Anlage. Beachten Sie, dass über die (an die b maXX controller PLC) angeschlossenen weiteren Module die Maschinenteile / Anlagenteile bzw. die Maschine / Anlage in Bewegung gesetzt werden können. Aktivieren Sie in jedem Fall deren Sicherheitseinrichtungen.



### GEFAHR (DANGER)

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

- schwere Körperverletzung
- Tod

Die Gefahr ist: **mechanische Einwirkung**. *Online-Änderungen werden automatisch an die SPS gesandt und sofort wirksam ohne vorher die Programmausführung auf der SPS zu stoppen.*

Halten Sie genügend Abstand von sich bewegenden Maschinenteilen / Anlagenteilen bzw. von der sich bewegenden Maschine / Anlage. Beachten Sie, dass über die (an die b maXX controller PLC) angeschlossenen weiteren Module die Maschinenteile / Anlagenteile bzw. die Maschine / Anlage in Bewegung gesetzt werden können. Aktivieren Sie in jedem Fall deren Sicherheitseinrichtungen.

**GEFAHR (DANGER)**

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

- schwere Körperverletzung
- Tod



Die Gefahr ist: **mechanische Einwirkung**. *Wenn Sie Variablen forcen oder überschreiben, während die SPS läuft, wird das SPS-Programm mit den geforcten oder überschriebenen Variablen ausgeführt.*

Halten Sie genügend Abstand von sich bewegenden Maschinenteilen / Anlagenteilen bzw. von der sich bewegenden Maschine / Anlage. Beachten Sie, dass über die (an die b maXX controller PLC) angeschlossenen weiteren Module die Maschinenteile / Anlagenteile bzw. die Maschine / Anlage in Bewegung gesetzt werden können. Aktivieren Sie in jedem Fall deren Sicherheitseinrichtungen.

**GEFAHR (DANGER)**

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

- schwere Körperverletzung
- Tod



Die Gefahr ist: **mechanische Einwirkung**. *Wenn Sie Breakpoints verwenden, während die SPS läuft, wird das SPS-Programm beim Auflaufen auf einen Breakpoint gestoppt.*

Halten Sie genügend Abstand von sich bewegenden Maschinenteilen / Anlagenteilen bzw. von der sich bewegenden Maschine / Anlage. Beachten Sie, dass über die (an die b maXX controller PLC) angeschlossenen weiteren Module die Maschinenteile / Anlagenteile bzw. die Maschine / Anlage in Bewegung gesetzt werden können. Aktivieren Sie in jedem Fall deren Sicherheitseinrichtungen.

## 5.2 b maXX controller PLC Projekt

Ein neues b maXX controller PLC Projekt wird unter PROPROG wt über das Menü "Datei\Neues Projekt" begonnen. Mit der „Vorlage für BMC\_M\_PLC01“ wird ein b maXX controller PLC Projekt geöffnet. Der Prozessortyp BMC\_M\_PLC01 in diesem Projekt ist in der Konfiguration voreingestellt.

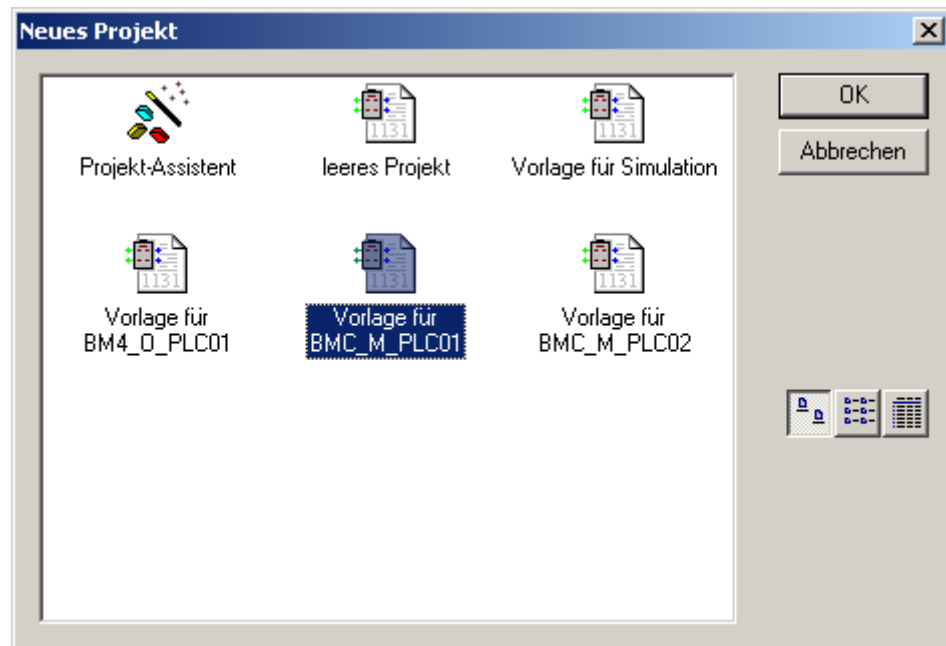


Abbildung 10: BMC\_M\_PLC01 Projektvorlage unter "Neues Projekt".

Mit der „Vorlage für BMC\_M\_PLC02“ wird ein b maXX controller PLC Projekt geöffnet. Der Prozessortyp BMC\_M\_PLC02 in diesem Projekt ist in der Konfiguration voreingestellt.

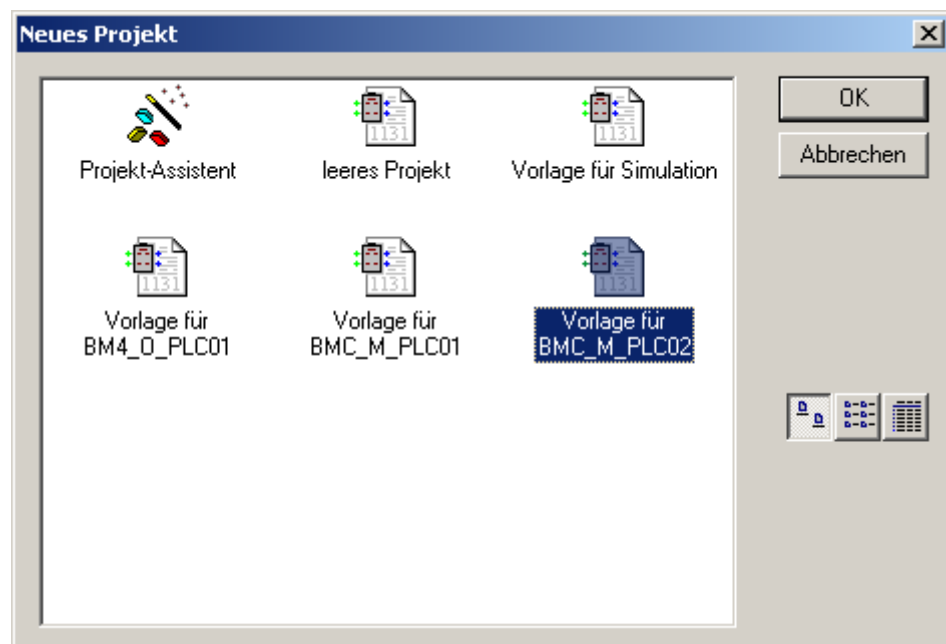


Abbildung 11: BMC\_M\_PLC02 Projektvorlage unter "Neues Projekt".

**HINWEIS**

Zur Programmierung der b maXX controller PLC BMC-M-PLC-02 wird das Programmiersystem ProProg wt III benötigt.

### 5.3 b maXX controller PLC Konfiguration

Als IEC 61131-3 Programmieroberfläche können mit PROPROG wt verschiedene Zielsysteme (CPUs) programmiert werden. Es können auch verschiedene Zielsysteme in einem Projekt programmiert werden. Die Erstellung eines Programms für das Zielsystem BMC\_M\_PLC01 erfolgt unter "Physical Hardware" mit der Ressource BMC\_M\_PLC01.

Über das Menü „Neues Projekt“ wird die Vorlage für BMC\_M\_PLC01 geöffnet. Die Konfiguration CONF1 hat unter Eigenschaften den SPS-Typ SH03\_30.

Die Erstellung eines Programms für das Zielsystem BMC\_M\_PLC02 erfolgt unter "Physical Hardware" mit der Ressource BMC\_M\_PLC02.

Über das Menü „Neues Projekt“ wird die Vorlage für BMC\_M\_PLC02 geöffnet. Die Konfiguration CONF1 hat unter Eigenschaften den SPS-Typ XPLC\_40.

Eine Konfiguration besteht aus mindestens einer Ressource. Die Ressource beinhaltet den spezifischen Datenbereich für die b maXX controller PLC, die Kommunikationsquelle, globale Variablenarbeitsblätter und die Tasks mit den Programmteilen.

## 5.3 b maXX controller PLC Konfiguration

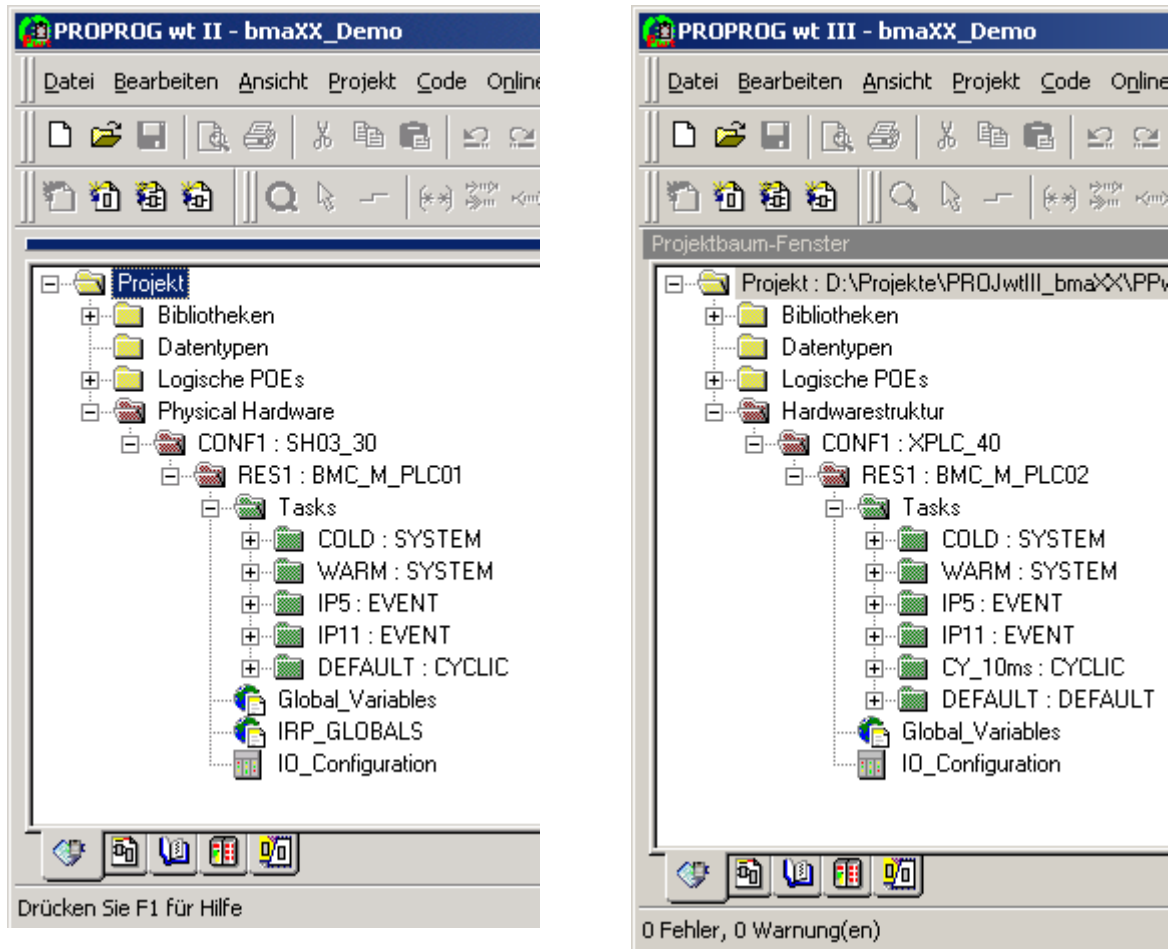


Abbildung 12: Einstellung unter "Physical Hardware" Eigenschaft einer b maXX controller PLC Konfiguration (links BMC-M-PLC-01, rechts BMC-M-PLC-02)

Übersicht der Einstellungen innerhalb der Physikalischen Hardware:

Projektvorlage	Konfiguration	Ressource
BMC_M_PLC01	SH03_30	BMC_M_PLC01
BMC_M_PLC02	XPLC_40	BMC_M_PLC02

Ein Projekt kann mehrere Konfigurationen mit jeweils mehreren Ressourcen enthalten.



### HINWEIS

Zur Programmierung der b maXX controller PLC BMC-M-PLC-02 wird das Programmiersystem ProProg wt III benötigt.

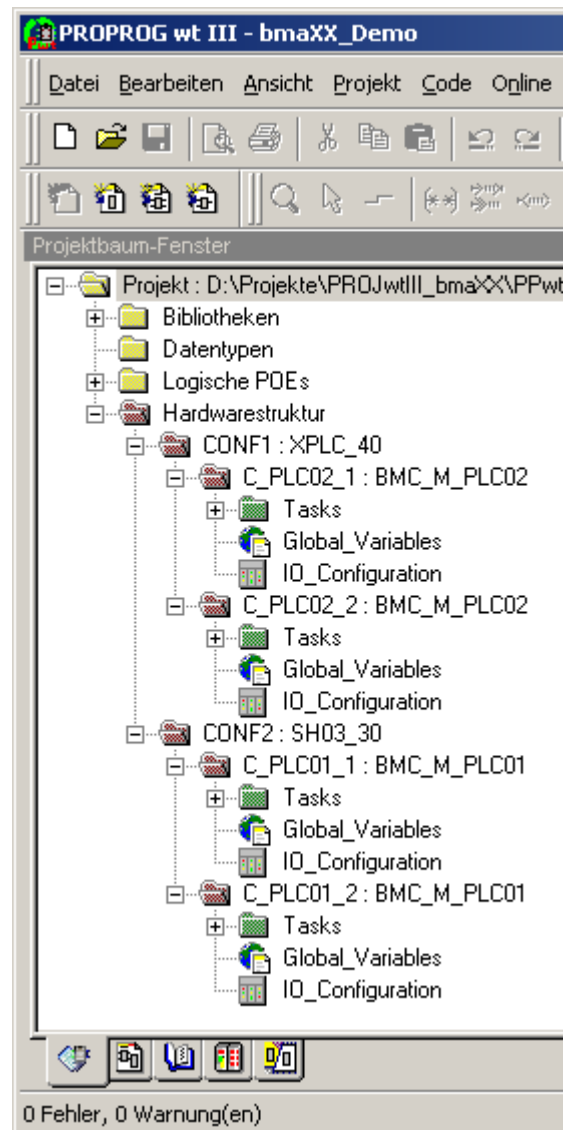


Abbildung 13: Projekt mit mehreren Konfigurationen und Ressourcen

## 5.4 b maXX controller PLC Ressource

Die Ressource beinhaltet die b maXX controller PLC spezifischen Einstellungen für ein Programm:

- Datenbereich
- Kommunikationsquelle
- Globale Variablenarbeitsblätter
- Unterschiedliche Tasks mit dem Programm
- Dokumentationsarbeitsblätter und I/O-Konfiguration  
(Die I/O-Konfiguration ist bereits fertig eingestellt und braucht nicht verändert werden.)

## 5.4 b maXX controller PLC Ressource

Einer Konfiguration können mehrere Ressourcen mit unterschiedlichen Ports zugeordnet werden. Somit ist eine komplette Applikation mit mehreren PLCs bzw. b maXX Systemen in einem Projekt umsetzbar.

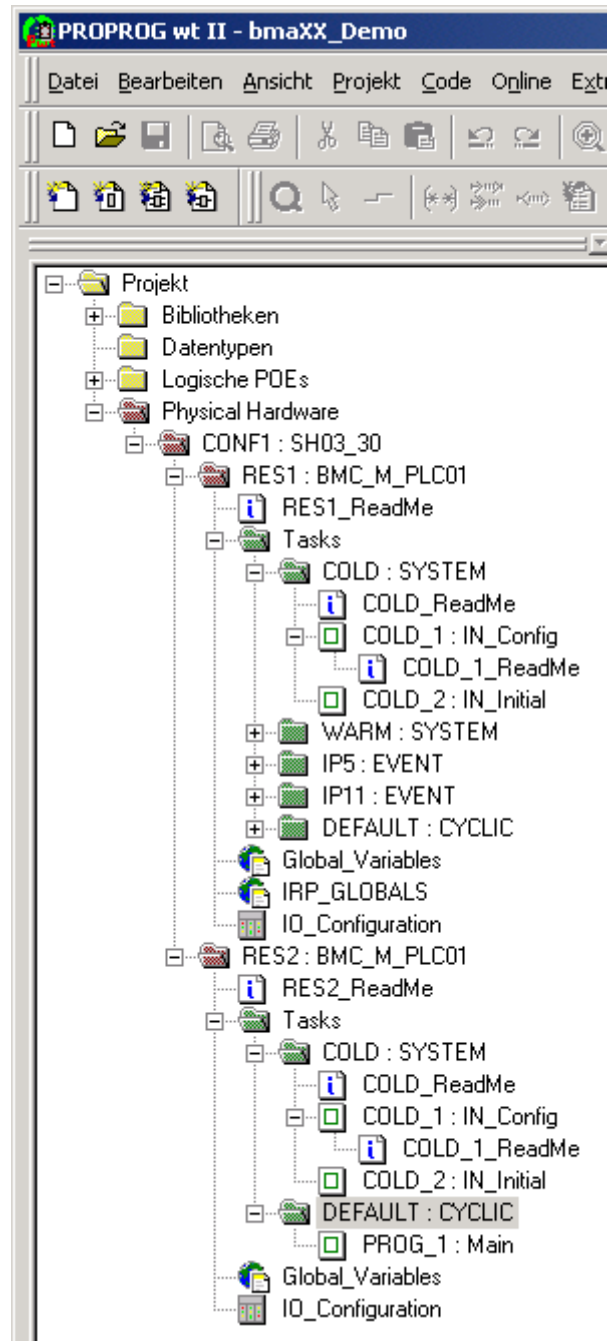


Abbildung 14: Beispiel einer b maXX controller PLC Konfiguration mit mehreren Ressourcen (BMC-M-PLC-01)

### 5.4.1 Kommunikation und Verbindung

Die Kommunikation zur Daten-Übertragung wird unter "Einstellung" im Kontextmenü der Ressource konfiguriert.



Die Kommunikation über RS232 wird, für den gewählten Port, wie folgt eingestellt:

- Baud: 38400
- Stoppbits: 1
- Datenbits: 8
- Parität: Keine
- Timeout: Default 2000 ms; Kommunikationsüberwachung während der Online-Darstellung.

Die Verbindung wird über den seriellen Port COM1 am PC aufgebaut:

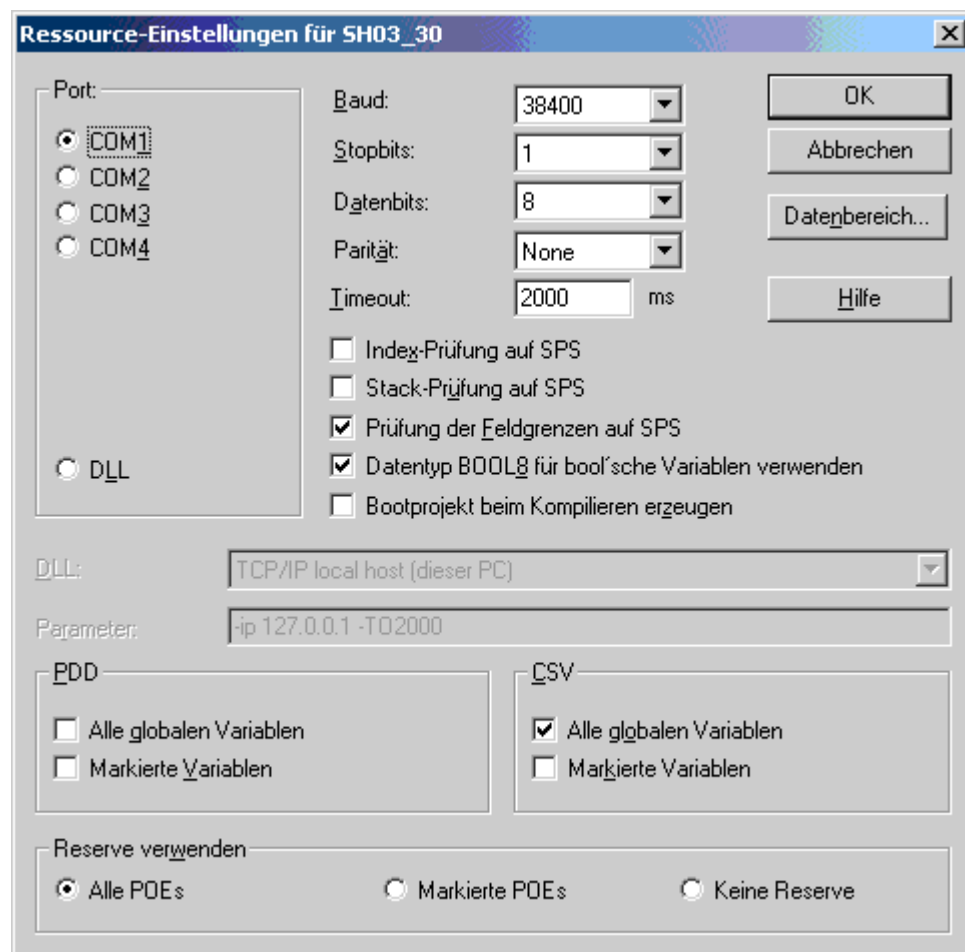


Abbildung 15: Ressource-Einstellung innerhalb einer b maXX controller PLC Konfiguration.

- "Index-Prüfung auf SPS": Deklarierte Feldgröße (Index) eines ARRAY wird zur Laufzeit kontrolliert. Achtung: Code-Ausführungszeit wird erhöht!
- "Stack-Prüfung auf SPS": Überprüfung auf Stacküberlauf zur Laufzeit. Mit der Programm-Task wird Stack-Speicher reserviert. Die Daten auf dem Stack erhöhen sich, wenn z. B. FB-Instanzen geschachtelt programmiert werden. Achtung: Code-Ausführungszeit wird erhöht!
- "Prüfung der Feldgrenzen auf SPS": Bei absoluter Adressierung wird geprüft, ob das Feld die parametrisierten Datenbereichsgrenzen überschreitet. Die Überprüfung erfolgt nur während des Übersetzungsvorganges auf dem PC. Die Code-Ausführungszeit wird nicht erhöht.

- "Datentyp BOOL8 für boolsche Variablen verwenden": 8-Bit Zugriff auf boolsche Variablen aktivieren. Für die b maXX controller PLC muss die Einstellung aktiviert sein.
- "Bootprojekt beim Kompilieren erzeugen": Bei aktivierter Funktion wird bereits beim Kompilieren des Projektes für jede Ressource das "bootfile.pro" erzeugt, und nicht erst bei Aktivieren der "Bootprojekt senden"-Funktion über die Ressourcenkontrolle.
- "PDD": Einstellungen zum Prozessdatenverzeichnis. (Siehe auch Programmierhandbuch PROPROG wt II / Online-Hilfesystem ProProg wt III)
- "CSV": Einstellungen zur Bereitstellung von Variablen für den OPC-Server. (Siehe auch Programmierhandbuch PROPROG wt II / Online-Hilfesystem ProProg wt III)
- "Reserve verwenden": Speicherreserve um Änderungen in Funktionsbausteinen und Funktionen mit der Funktion "Online-Änderung" durchführen zu können. (Siehe auch Programmierhandbuch PROPROG wt II / Online-Hilfesystem ProProg wt III)

Die Ressource-Einstellung kann für jede b maXX controller PLC - Ressource separat vorgenommen werden. Die serielle oder Ethernet-Kommunikationsquelle

- COM1
- COM2
- COM3
- COM4
- DLL

wird genutzt

- zur Ressourcenkontrolle.
- zur Online-Darstellung von Variablen und Strukturen im Watch-Fenster
- zum Senden des kompilierten Projekts.
- zum Debuggen.
- zur Verbindung zum OPC-Server.

### Einstellung der Ethernet-Kommunikationsquelle durch Auswahl bzw. Angabe der TCP/IP-Adresse

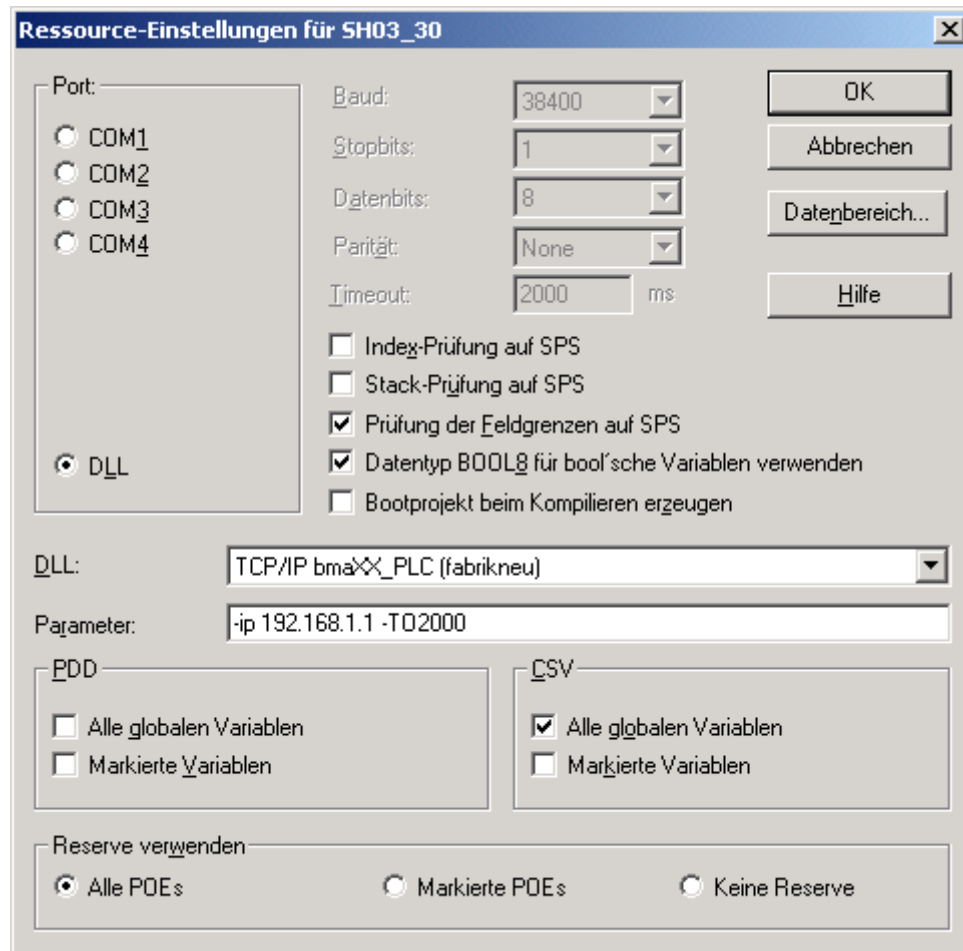


Abbildung 16: Einstellung der TCP/IP Adresse

Nach Umstellen des Ports auf "DLL" hat der Anwender über das DLL-Menü die Möglichkeit, die Anwendungsfälle "Soft-PLC" (=TCP/IP local Host (dieser PC)) und "b maXX PLC - Zugriff über Ethernet" (=TCP/IP bmaXX\_PLC (fabrikneu)) auszuwählen:

- **Soft-PLC:**  
Die voreingestellte TCP/IP-Adresse im Feld "Parameter" muss nicht geändert werden, wenn die Soft-PLC auf dem gleichen Rechner installiert ist. Bei Installation der Soft-PLC auf einem anderen Rechner muss der Anwender dessen entsprechende TCP/IP-Zieladresse (bzw. den entsprechenden Netzwerk-Namen) hier eintragen, um von PROPROG wt aus über TCP/IP auf die Soft-PLC zugreifen zu können.
- **b maXX controller PLC:**  
Um auf die b maXX controller PLC über Ethernet zugreifen zu können, muss im b maXX System zusätzlich zur b maXX controller PLC ein Ethernet-fähiges Modul (z. B. BMC-M-ETH-02) vorhanden sein. Die im Parameterfeld voreingestellte TCP/IP-Adresse "192.168.1.1" entspricht der IP-Adresse, die auf dem Ethernet-Modul bei Auslieferung vorinstalliert ist, so dass der Anwender eine erste Verbindung zum Modul für seine Grund-Initialisierung aufnehmen kann, in der er dann die Einstellung der TCP/IP-Adresse für sein TCP/IP-Netz an der Maschine vornehmen muss.

## 5.4 b maXX controller PLC Ressource

Die genaue Vorgehensweise für die Grund-Initialisierung ist den jeweiligen Betriebsanleitungen und Applikations-Handbüchern der eingesetzten Ethernet-Module zu entnehmen.

### 5.4.2 Kontrolldialog für Ressourcen

Mit dem Kontrolldialog für Ressourcen wird die Programmübermittlung zur b maXX controller PLC und der Betriebszustand der b maXX controller PLC eingestellt.

Sind mehrere Ressourcen in einem Projekt aktiv, muss die gewünschte Ressource ausgewählt werden.

Auswählen einer Ressource in PROPROG wt II:

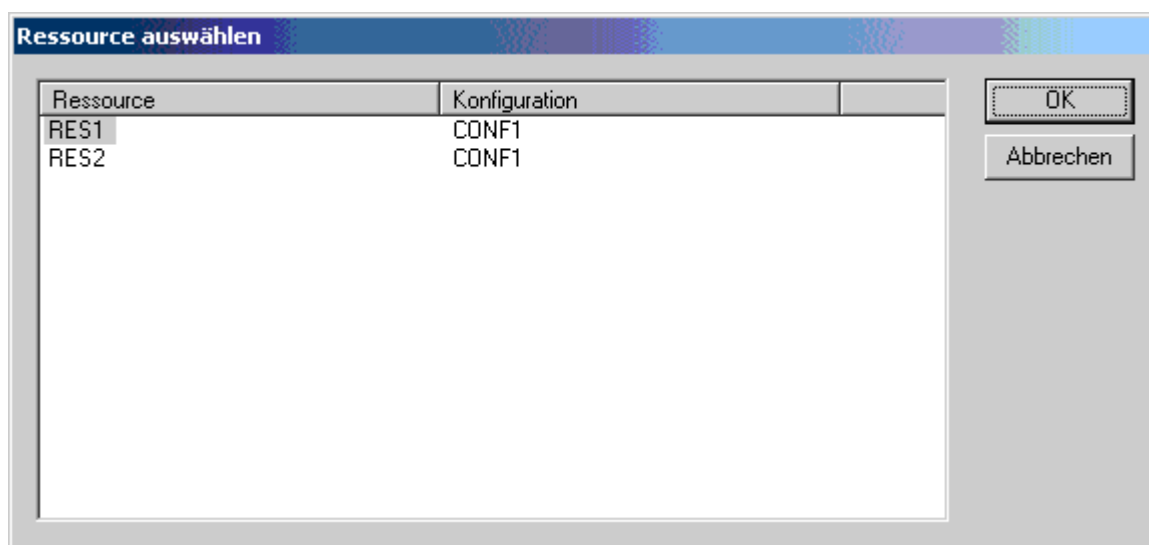


Abbildung 17: Auswahl der Ressource in PROPROG wt II - Kontrolldialog öffnen

Der Kontrolldialog wird geöffnet.

Auswählen einer Ressource in ProProg wt III:

Klicken Sie auf die gewünschte Ressource und drücken Sie anschließend auf den Button „Verbinden“.

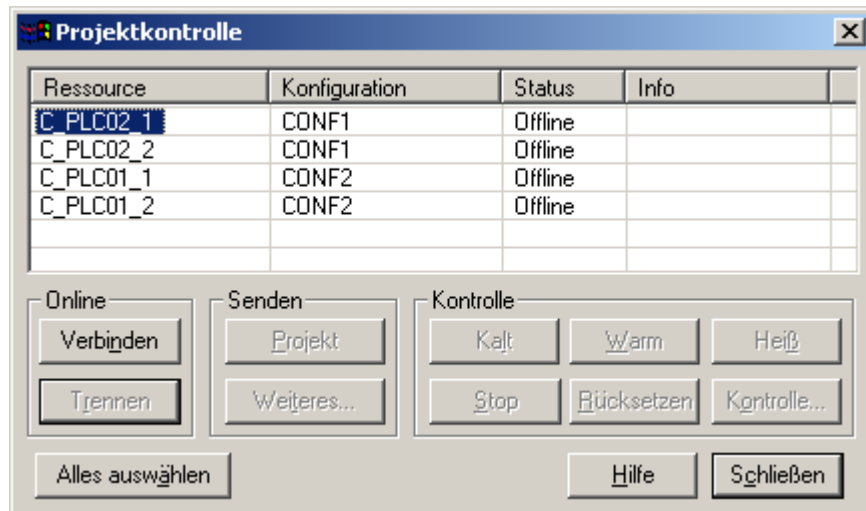


Abbildung 18: Auswahl der Ressource in ProProg wt III - Verbinden

Drücken Sie jetzt auf den Button „Kontrolle...“

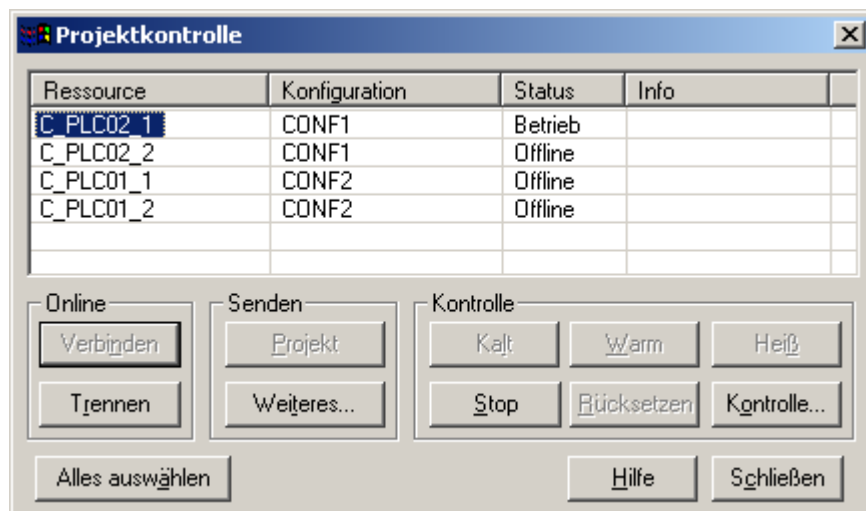


Abbildung 19: Auswahl der Ressource in ProProg wt III - Kontrolldialog öffnen

Der Kontrolldialog wird geöffnet.



Abbildung 20: Funktionen des Kontrolldialogs der ausgewählten Ressource im Zustand "RUN".

Mit "**Senden**" kann das kompilierte Projekt an das Zielsystem übermittelt werden.

Senden des Projektes zum Zielsystem.

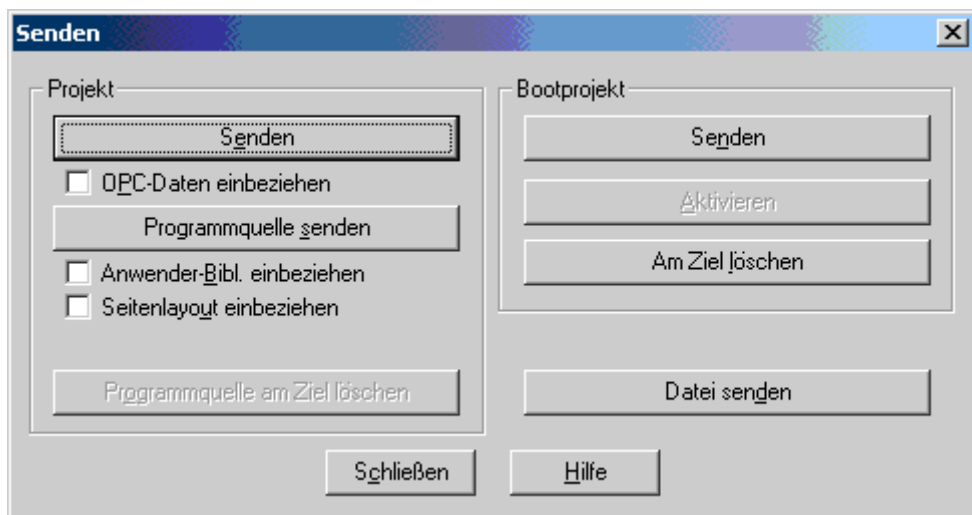


Abbildung 21: b maXX controller PLC Ressource Übertragung zum Flash oder RAM-Speicher.

Mit "**Bootprojekt / Senden**" wird das aktuelle Bootprojekt der Ressource gelöscht, das kompilierte Projekt als Bootprojekt gesendet und im b maXX controller PLC Flash-Speicher abgelegt. Mit "**Aktivieren**" wird das Projekt vom b maXX controller PLC Flash-Speicher in den RAM-Speicher geladen.

Mit "**Am Ziel löschen**" wird das Bootprojekt im Flash gelöscht.

Mit "**Projekt / Senden**" wird das kompilierte Projekt der Ressource direkt ins RAM gesendet und kann über den Kontrolldialog (Button „Kalt“) anschließend gestartet werden. Das Bootprojekt bleibt unverändert im b maXX controller PLC Flash-Speicher. Nach dem nächsten Hardware-Reset oder Steuerung Aus → Ein wird wieder das Bootprojekt geladen!

**HINWEIS**

Die Menüpunkte "Programmquelle senden", "Datei senden" und das Attribut "OPC-Daten einbeziehen" werden zur Zeit nicht unterstützt und dürfen nicht angewählt werden.

Will man das über „**Bootprojekt / Senden**“ ins Flash abgelegte Projekt aktivieren, geht man folgendermaßen vor:

- direkt Spannung „Aus/Ein“

oder

- über PROPROG wt:

Im Kontrolldialog das geladene Programm (RAM) über „**Stop**“ anhalten und über „**Reset**“ löschen. Der Betriebsstatus wechselt auf den Zustand „**Ein**“.



Abbildung 22: Ressourcenkontrolle Status „Ein“

Klicken Sie im „Senden“-Menü des Kontrolldialogs dann „Bootprojekt / Aktivieren“ an. Dadurch wird das Flash-Projekt auf der b maXX controller PLC im RAM installiert.

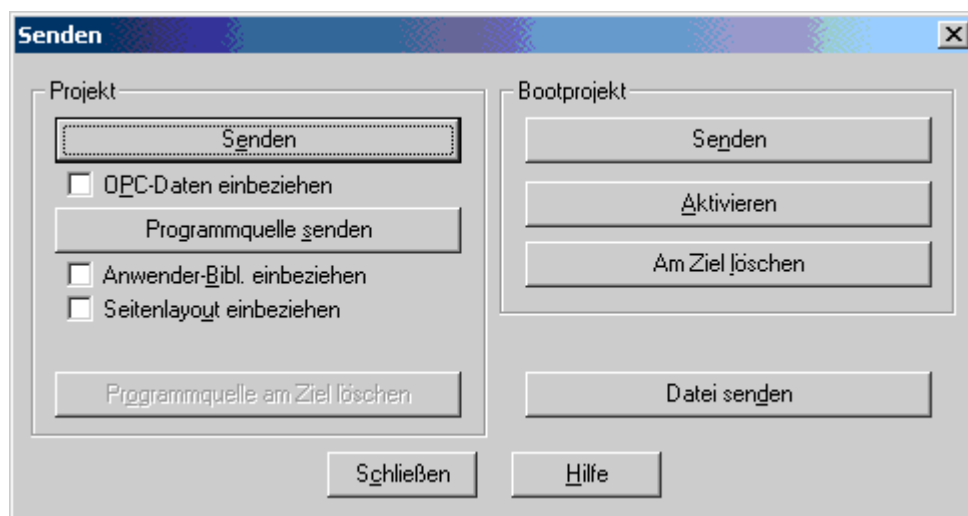


Abbildung 23: Kontrolldialog „Senden“

Dann können Sie das Programm über „Kalt“ im Kontrolldialog starten:



Abbildung 24: Ressourcenkontrolle Zustand "STOP".

### Beschreibung der Buttons in der b maXX controller PLC Ressourcenkontrolle

- Button „**Stop**“: Programmausführung wird angehalten, der PLC-Betriebsstatus wechselt von „Run“ nach „Stop“ (Der Status wird oben in der Ressourcenkontrolle angezeigt).
- Button „**Reset**“: Löscht das RAM-Projekt auf der b maXX controller PLC (nicht das im Flash abgespeicherte Bootprojekt!). Der Status wechselt von „Stop“ nach „Ein“.
- Button „**Senden**“: Ruft die Seite für die Programmübermittlung (b maXX controller PLC Ressource Übertragung zum RAM oder Flash) auf.
- Button „**Fehler**“: Hier können in der b maXX controller PLC anstehende Fehler- und Warnmeldungen ausgelesen werden, wenn der Button aktiviert ist. Durch Drücken des aktiven Fehler-Buttons werden die Fehlereinträge von der Steuerung abgefragt und im Fehler- bzw. Warnungsmeldefenster angezeigt.
- Button „**Schließen**“: Die b maXX Ressourcenkontrolle wird wieder geschlossen.
- Buttons „**Kalt/Warm/Heiß**“: Über diese PROPROG-Befehle kann die b maXX controller PLC manuell gestartet werden. Der Status wechselt von „Stop“ nach „Run“. Ein Start der PLC über die Buttons kann jedoch nur erfolgen, wenn der Schalter S1 auf dem Modul BMC-M-PLC-01 bzw. wenn der Schalter S2 auf dem Modul BMC-M-PLC-02 auf „RUN“ eingestellt ist. Sonst bleibt die Steuerung im Status „Stop“.

Die drei Startbuttons unterscheiden sich dabei wie folgt:

- Button „**Kalt**“: Die PLC durchläuft einmalig die PROPROG-System-Task „Kaltstart (SPG1)“, in der die Anwender-Programminitialisierung stattfindet, und wechselt dann in die zyklische Programmbearbeitung.  
Der Kaltstart ist dadurch gekennzeichnet, dass alle Variablen mit ihren Default-Werten initialisiert werden. Wurde im Projekt kein Default-Wert vom Anwender angegeben, wird ein Defaultwert „0“ (bzw. „FALSE“ für boolesche Variablen) gesetzt.
- Button „**Warm**“: Die PLC durchläuft einmalig die PROPROG-System-Task „Warmstart (SPG0)“, in der die Anwender-Programminitialisierung stattfindet, und wechselt dann in die zyklische Programmbearbeitung.  
Im Unterschied zum Kaltstart behalten die remanenten Merker ihre Werte, sie werden also nicht auf Default-Werte zurückgesetzt.  
Falls keine remanenten Merker auf der b maXX controller PLC vorhanden sind, und wenn der gleiche Anwender-Code sowohl im Kaltstart- als auch in der Warmstart-Task eingebunden ist, gibt es keine Unterschiede im Verhalten zwischen Kalt- und Warmstart.



- Button „**Heiß**“: Es wird keine Initialisierungstask durchlaufen, sondern direkt in die zyklische Programmbearbeitung gewechselt.

### HINWEIS



Ein Start der PLC über die Buttons (Kalt/Warm/Heiß-Start) bzw. nach Spannung Aus/Ein kann nur dann erfolgen, wenn der Schalter S1 auf dem Modul BMC-M-PLC-01 bzw. wenn der Schalter S2 auf dem Modul BMC-M-PLC-02 auf „RUN“ eingestellt ist. Sonst bleibt die Steuerung im Status „Stop“, und es wird kein Anwendercode ausgeführt.

Wird ein HW-Reset am Schalter S1 (bzw. S2 auf dem Modul BMC-M-PLC-02) ausgeführt, muss der Schalter S1 (bzw. S2 auf dem Modul BMC-M-PLC-02) anschließend ebenfalls auf "RUN" gestellt werden, damit der Anwendercode abgearbeitet werden kann.

Nach Senden eines Bootprojektes und anschließendem automatischen Programmstart (Spannung aus/ein bzw. HW-Reset am Schalter/Taster S1 des Moduls BMC-M-PLC-01 bzw. S2 des Moduls BMC-M-PLC-02) wird einmalig ein Kaltstart durchgeführt, in dem auch die remanenten Merker auf die vorgegebenen Default-Werte gesetzt werden. Die Information „Kaltstart wurde durchlaufen“ wird in den remanenten Bereich abgespeichert.

Bei jedem weiteren automatischen Wechsel in den Zustand „Run“ wird diese abgespeicherte Info ausgewertet und immer ein Warmstart ausgeführt, bei dem die remanenten Merker ihre Inhalte beibehalten.

Aktiviert der Anwender den manuellen Start-Button "Kalt" in der b maXX controller PLC Ressourcenkontrolle, wird das explizite Setzen der Default-Werte auch für die remanenten Merker erzwungen.

Bei Abspeichern von Bootprojekten ist das automatische Wiederanlauf-Verhalten davon abhängig, ob sich im remanenten Merkerbereich etwas geändert hat: Wird auf der Steuerung erkannt, dass im Vergleich zum Projekt-Download vorher neue remanente Merker angelegt bzw. gelöscht, oder der Name, der Datentyp oder die Reihenfolge von remanenten Merkern geändert wurden, wird daraufhin bei Wechsel in "Run" ebenfalls einmalig ein Kaltstart ausgeführt, um die Konsistenz der remanenten Merker gewährleisten zu können.

Das Ziel bei Projekten ist, remanente Daten auf der Steuerung auch nach einem Projekt-Download wenn möglich zu erhalten. Bei Änderung von Default-Werten alleine wird deswegen kein automatischer Kaltstart ausgeführt und muss gegebenenfalls vom Anwender selbst explizit durchgeführt werden.

### WARNUNG (WARNING)



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

- schwere Körperverletzung
- Tod



*Die Gefahr ist: **mechanische Einwirkung**. Mit dem Kontrolldialog kann die SPS-Betriebsart geändert werden. Dies kann Auswirkungen auf die Maschinenfunktion haben.*

Halten Sie genügend Abstand von sich bewegenden Maschinenteilen und aktivieren Sie in jedem Fall deren Sicherheitseinrichtungen.

- Button "**Hochladen**": Upload-Funktionen in PROPROG wt werden zur Zeit nicht unterstützt.
- Button "**Info**": Allgemeine Informationen zur b maXX controller PLC selbst und zum Projekt-Status auf der b maXX controller PLC (Beschreibung siehe unten).

- Button "**Hilfe**": Die allgemeine und spezifische SPS-Hilfe für die b maXX controller PLC werden aufgerufen.

### Beschreibung der "Info"-Seite "Ressource-Informationen"

Diese Seite wird über den „Info“ Button in der b maXX controller PLC Ressourcenkontrolle aufgerufen:

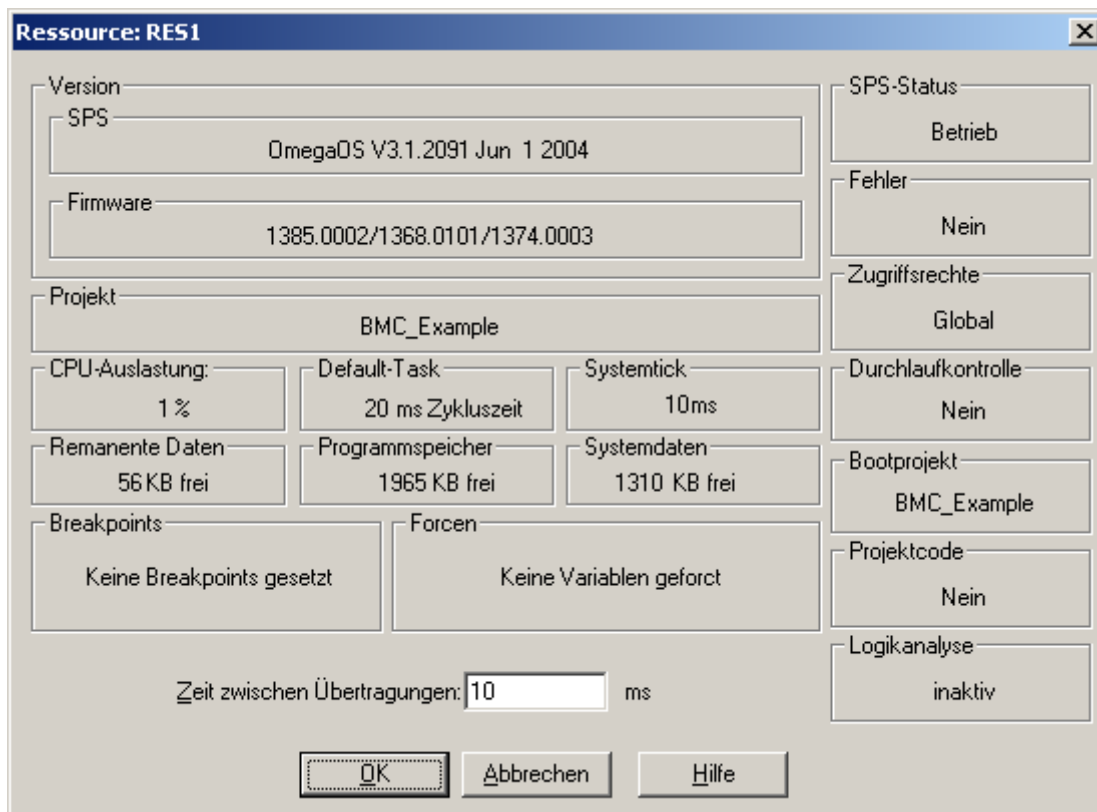


Abbildung 25: Informationsfenster Ressource

- Version: SPS "OmegaOS V3.1.2091 Jun 1 2004"
  - „OmegaOS“: Produktbezeichnung.
  - „V3.1.2091“: OmegaOS Compiler Info.
  - „Jun 1 2004“: Datum, an der die PLC-Version erzeugt wurde.
- Firmware: "1385.0002/1368.0101/1374.0003"
  - „1385.0002“: Version OmegaOS SW 6.1385.0002.
  - „1368.0101“: Version FPGA SW 6.1368.0101.
  - „1374.0003“: Version BOOT SW 6.1374.0003.
- Projekt: Name des aktiven Projekts (Die Darstellung des Projektnamens ist auf 11 Buchstaben begrenzt).
- CPU-Auslastung: Die Auslastung wird vom System automatisch zur Echtzeit ermittelt und wird hier in Prozent angezeigt.  
Es fließen u. a. die Laufzeiten der einzelnen zyklischen Tasks sowie der nicht über Watchdog-Zeiten überwachten Bypass-Tasks ein. Um wesentliche Betriebssystemfunktionen wie beispielsweise die Online-Kommunikation sicherzustellen, sollte eine Gesamt-Auslastung kleiner 80 % angestrebt werden. Bei 100 % Auslastung wird ein

System-Fehler generiert, der zum Stopp der b maXX controller PLC führt.

Eine Auslastung von 100 % wird z. B. erreicht, wenn die Ausführung der Default-Task nach ca. 10 Sekunden noch nicht abgeschlossen werden konnte und die eingestellte Watchdogzeit für die Default-Task auf größer 10 Sekunden vom Anwender geändert wurde. (Bei Watchdog-Einstellungen kleiner 10 Sekunden in der Default-Task würde vorher die Watchdog-Überwachung einen System-Fehler auslösen und zurückmelden). Die augenblickliche Default-Taskdauer kann im Fenster rechts neben der CPU-Auslastung abgelesen werden.

Eine Überlastung kann vermieden werden, wenn man die entsprechenden Tasks in mehrere kürzere Einzel-Tasks mit größeren Aufrufintervallen teilt oder einzelne Programmteile (z. B. in den Bypass-Tasks) jeweils überspringt, um die Task-Laufzeiten zu verringern.

- Default-Task: Die augenblickliche Zykluszeit der Default-Task kann hier direkt in ms abgelesen werden.
- Systemtick: Diese Zeit gibt die kleinste Auflösung von Zyklus- bzw. Überwachungszeiten an, die das PLC-Laufzeitsystem verwalten kann.

Beispiel: Ob man das Intervall in einer Cyclic-Task mit „400 ms“ oder „409 ms“ angibt, ist „egal“: Bei einem Systemtick vom 10 ms wird immer „400 ms“ (also auf die Dauer eines Systemticks abgerundet) aktiviert.

- Remanente Daten:  
Bei der Anzeige "Remanente Daten" wird der verfügbare remanente Gesamt-Speicherbereich auf der PLC angezeigt. Da darin außer den remanenten Merkern noch zusätzliche remanente System-Informationen abgelegt werden (z. B. ob ein Kaltstart bei einem Projekt bereits durchlaufen wurde), ist der vom Anwender als remanente Merker nutzbare Bereich kleiner und kann den Datenbereichs-Einstellungen („Remanente Merker/Anfangsadresse Anwender“ und „Ende System“, siehe unten) entnommen werden.
- Programmspeicher: Anzeige des noch freien PLC-Programmspeichers des aktiven Projektes in KB.

Mit einem freien Programmspeicher von 2046 KB z. B. kann man:

- max. 400000 AWL-Zeilen (LD/ST-Anweisungen auf globale Variablen) bzw.
- typ. 120000 AWL-Zeilen (Typische AWL-Anweisungen auf Strukturen und Instanz-Variablen) programmieren.

- Systemdaten:  
Dynamischer Speicher (z. B. 1460 KB) für Debug- und Logic-Analyzer-Funktionen.
- Breakpoints: Hier wird angezeigt, ob auf der PLC Breakpoints gesetzt sind. Wenn dies der Fall ist, wird das Kontrollkästchen 'Breakpoints rücksetzen' angezeigt. Aktivieren Sie dieses Kontrollkästchen, wenn Sie alle gesetzten Breakpoints gleichzeitig rücksetzen möchten. (Breakpoints können über das Menü "Online/Debug" verwaltet werden).
- Forcen: Zeigt an, ob Variablen geforct wurden. Wenn dies der Fall ist, wird das Kontrollkästchen 'Forceliste rücksetzen' angezeigt. Aktivieren Sie dieses Kontrollkästchen, wenn Sie alle geforcten Variablen gleichzeitig rücksetzen möchten.
- Freier Speicher Systemdaten Laufzeitsystem (für Auslastung Oszilloskopfunktion).
- Zeit zwischen Übertragungen: Hier kann der Anwender eine Zeit angeben, nach der das Programmiersystem frühestens eine neue Kommunikation an die PLC startet, nachdem die Letzte abgeschlossen ist. Durch ein Heraufsetzen der Zeit durch den Anwender kann die allgemeine SPS-Belastung durch Kommunikationsaufgaben vermindert werden. Der Wert ist in ms angegeben.
- SPS-Status: Zeigt den augenblicklichen Betriebszustand (z. B. „Run“, „Stop“, etc) der b maXX controller PLC an.

- Fehler: Zeigt an, ob eine SPS-Fehlermeldung im Fehlerkatalog ansteht und über den Button „Fehler“ in der b maXX controller PLC Ressourcenkontrolle ausgelesen werden kann.
- Zugriffsrechte: Zeigt die Zugriffsrechte zum Senden und Debuggen auf der SPS an.
- Durchlaufkontrolle: Zeigt an, ob der Adressstatus mit Durchlaufkontrolle vom Anwender aktiviert wurde (Menüpunkt „Online/Adressstatus“).
- Bootprojekt: Name des (inaktiven) Bootprojekts auf der b maXX controller PLC. (Die Darstellung des Projektnamens ist auf 11 Buchstaben begrenzt.)  
Über die Seite für die Programmübermittlung (b maXX controller PLC Ressource Übertragung) kann durch den Menüpunkt „Aktivieren“ das Bootprojekt in das aktive RAM-Projekt übersetzt werden. (Geschieht immer automatisch bei einem HW-RESET oder Spannung aus/ein).
- Projektcode: Zeigt an, ob ein Source-Projekt (zwt) in der PLC vorhanden ist (wird aus Speicherplatzgründen zur Zeit nicht unterstützt).
- Logikanalyse: Zeigt an, ob die Logikanalyse-Aufzeichnung gerade aktiv ist.

### 5.4.3 Datenbereich

Der Datenbereich enthält die b maXX controller PLC spezifische Einstellung von physikalischen Adressbereichen in Form von Merkern, auf die der Anwender und der Übersetzer in PROPROGRAMM zugreifen können.

Die Einstellungen werden mit dem Projekt abgespeichert.

Das „Datenbereichs“-Formular kann im Projektbaum über die „Einstellungen/Datenbereich...“ (rechte Maustaste) in der Ressource „Physical Hardware/RES1“) erreicht werden:

Nicht remanente Merker	
Anfangsadresse Anwender:	0
Ende Anwender / Start System:	80000
Ende System (max 2097147):	2097147
Reserve pro POE:	500

Remanente Merker	
Anfangsadresse Anwender:	10000000
Ende Anwender / Start System:	10000000
Ende System (max 10057323):	10057323
Reserve pro POE:	10%

Autom. Deklaration von Anwenderspeicher in I/O-Konfiguration

Abbildung 26: Ressource b maXX controller PLC spezifischer Datenbereich ohne Anwenderbereich Remanente Merker.

Die Einteilung des Datenbereichs erfolgt dabei in zweierlei Hinsicht:

Erstens untergliedert sich der Datenbereich in „Nicht remanente Merker“ und „Remanente Merker“.

Zweitens wird in beiden Bereichen unterschieden, ob der Anwender oder der Übersetzer in PROPROG wt auf diese Merker zugreifen dürfen.

Die beiden auf dem Bild sichtbaren Einstellungen "Reserve pro POE" dienen der Compiler-Funktionalität „Online-Änderungen senden“ und reserviert Programmspeicher.

Begriffserklärung „Nicht remanente“ und „Remanente“ Merker:

Remanente Merker behalten im Gegensatz zu den nicht remanenten Merkern ihren Wert auch nach Wegnahme der Spannungsversorgung, so dass mit diesen Werten beim nächsten Wiederanlauf der PLC weitergearbeitet werden kann. Bevorzugt werden darin anlagenspezifische Daten abgelegt, die erst während des Betriebes ermittelt werden und nicht aus anderen Werten neu berechnet werden können.

Als Beispiel dafür sei ein Wickeldurchmesser genannt, der sich über die Geschwindigkeit und der Maschinenlaufzeit kontinuierlich verändert.

Begriffserklärung **Anwender-Merker** („Anfangsadresse-Anwender/ Ende Anwender“):

Will der Anwender in seinem Projekt direkt auf Merker zugreifen, können bei den nicht remanenten und remanenten Merkern dafür Bereiche vorab reserviert werden.

Man sollte jedoch darauf achten, dass die Anwenderbereiche nicht zu groß eingestellt werden, da sonst bei größeren Projekten u. U. zu wenig Systempeicher für den PRO-PROG-Übersetzer vorhanden ist.

Wieviele Bytes des Speichers vom Projekt für diese Merker verbraucht werden, ist unter Infos im Meldungsfenster ersichtlich. Vorher muss eine Projekt-Code-Erzeugung erfolgen, wobei in den Ressource-Einstellungen „Keine Reserve“ aktiviert sein muss.

Begriffserklärung **System-Merker** („Start System/Ende System“):

In PROPROG wt kann der Anwender symbolische Variablen definieren und in seinem Programmcode verwenden, ohne eigens Merker dafür anlegen zu müssen. Das wird beim Übersetzungsvorgang in PROPROG wt vom System automatisch erledigt, indem diese Variablen auf die Merker im Systembereich (= vom System vergebene Merker) abgebildet werden.

Da dies bei jeder Projekt-Code-Erzeugung erfolgt, können die Teilungsgrenzen auch im späteren Projektverlauf noch angepasst werden.

Wieviele Bytes des Speichers vom Projekt für diese Merker verbraucht werden, ist unter Infos im Meldungsfenster ersichtlich. Vorher muss eine Projekt-Code-Erzeugung erfolgen, wobei in den Ressource-Einstellungen „Keine Reserve“ aktiviert sein muss.

Der voreingestellte Standard-Anwenderbereich für die Vergabe absoluter Merker-Adressen innerhalb einer b maXX Ressource im nicht remanenten Bereich ist:

- %MB 0 - %MB 79999

Der restliche Bereich von %MB 80000 auf z. B. %MB 2097147 bleibt dann dem System vorbehalten für die automatische Umsetzung der nicht remanenten symbolischen Anwender-Variablen in diesen Merkerbereich.

Der Standard-Anwenderbereich für die Vergabe absoluter Merker-Adressen innerhalb einer b maXX Ressource im remanenten Bereich ist deaktiviert.

Der Anwender kann bei den remanenten Daten, entsprechend der jeweiligen Projektanforderung, die Teilgrenze "Ende Anwender/Start System" noch oben verschieben (z. B. von 10000000 auf 10020000).

Dies würde dann einem remanenten Merkerbereich %MB 10000000 - %MB 10019999 entsprechen, den der Anwender dann in seinem Projekt als absolute remanente Merker direkt ansprechen kann.

Der restliche Bereich von %MB 10020000 auf z. B. %MB 10057323 ist dann dem System vorbehalten für die automatische Umsetzung von remanenten symbolischen Anwender-Variablen in diesen Merkerbereich.

Werden absolute Merker-Adressen vom Anwender verwendet, die im Systembereich liegen, erfolgt beim Übersetzen des Projektcodes eine entsprechende Fehlermeldung.

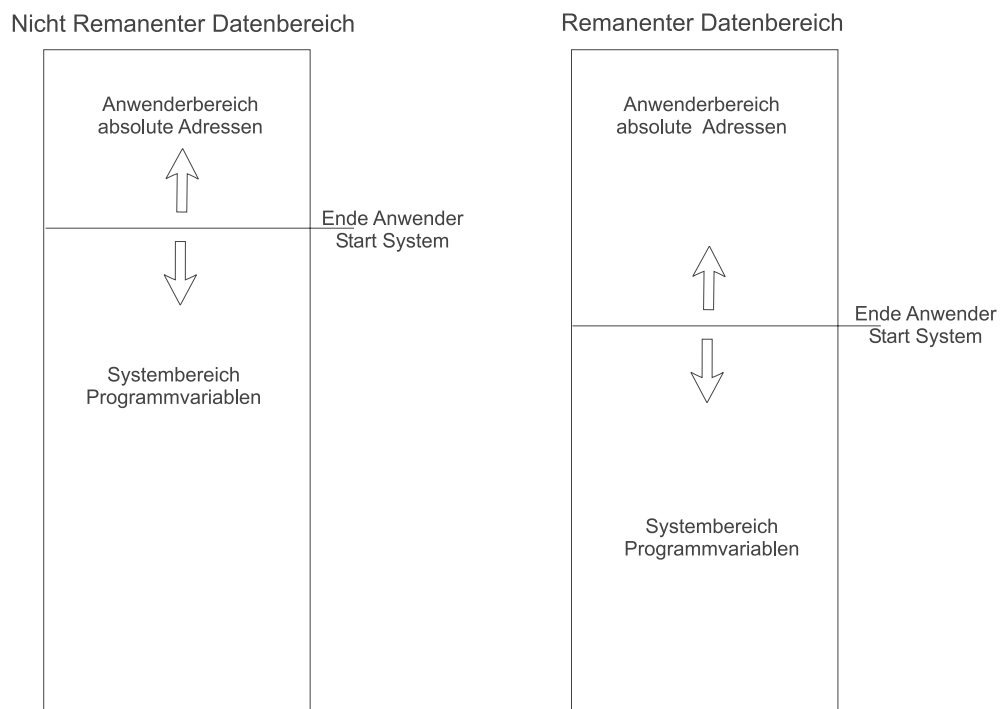


Abbildung 27: Aufteilung und Einstellung des b maXX controller PLC Datenbereichs

### Vergabe von absoluten Merker-Adressen im Programm

Eine absolute b maXX controller PLC-Adresse oder ein Variablenfeld mit einem Datentyp

- 16-Bit (WORD) kann nur für eine ohne Rest durch zwei zu dividierende Adresse und Null vergeben werden.
- 32-Bit (DWORD) kann nur für eine ohne Rest durch vier zu dividierende Adresse und Null vergeben werden.

Beispiel:

Eine Variable vom Datentyp DWORD soll im nicht remanenten Datenbereich deklariert werden.

PROPROG wt II:

```
d_abs_adr AT %MD12 : DWORD;      (* Symbolische Variable auf
absoluter Adresse *)
```

oder auch:

```
di_abs_adr AT %MD16 : DINT;      (* Symbolische Variable auf
absoluter Adresse *)
```

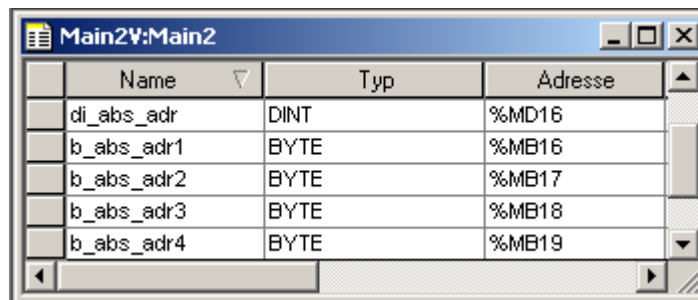
```
b_abs_adr1 AT %MB16 : BYTE;
```

```
b_abs_adr2 AT %MB17 : BYTE;
```

```
b_abs_adr3 AT %MB18 : BYTE;
```

```
b_abs_adr4 AT %MB19 : BYTE;
```

ProProg wt III:



Name	Typ	Adresse
di_abs_adr	DINT	%MD16
b_abs_adr1	BYTE	%MB16
b_abs_adr2	BYTE	%MB17
b_abs_adr3	BYTE	%MB18
b_abs_adr4	BYTE	%MB19

Abbildung 28: Variablendeklaration

Ein Vorteil bei der Verwendung von absoluten Merkern kann u. U. der schnelle und einfache Zugriff auf die einzelnen Variablenteile ohne zusätzliche Maskier- und Konvertierungsfunktionen sein:

Wird beispielsweise der Wert "DINT#16#98765432" auf die angelegte absolute Variable "di\_abs\_adr" geschrieben, kann der Variablen-Wert direkt als Einzel-Bytes weiterbearbeitet werden (INTEL-Format: b\_abs\_adr1 = BYTE#16#32, ..., b\_abs\_adr4 = BYTE#16#98).

Aber Vorsicht! Diese Art der Programmierung ist dann nicht mehr instanziiert, und die Sequenz könnte von einem Interrupt unterbrochen und von dort nochmal aufgerufen werden, so dass nach Beendigung des Interrupts mit falschen Werten weiter gearbeitet wird!

### 5.4.4 Die b maXX controller PLC Event-Tasks

Die b maXX controller PLC Event-Tasks dienen dem ereignisgesteuerten Programmaufruf (Interrupt).

Sie bestimmen durch ihre Art und Codelaufzeit das Echtzeitverhalten des b maXX Systems. Das Echtzeitverhalten des b maXX Systems ist nicht nur von der b maXX controller PLC, sondern auch von den weiteren Systemkomponenten abhängig.



#### HINWEIS

Die Daten der Systemkomponenten / Module rechts des Netzteils für b maXX controller PLC werden synchron zum Betriebssystem-Timer-Tick (alle 10 ms bei BMC-M-PLC-01 bzw. alle 2 ms bei BMC-M-PLC-02) der b maXX controller PLC aktualisiert.

Die Eigenschaft der Event-Task wird über die Ereignis-Nummer der Task zugewiesen:

Event	Bezeichnung	Level
0	CPU-Timer 1	14
1	reserviert	-
2	CPU-Timer 2	13
3, 4	reserviert	-
5	Timer A	14
6	Timer A	13
7, 8, 9, 10	reserviert	-
11	Sync-Signal 1 Optionsmodul	14
12	Sync-Signal 2 Optionsmodul	14



#### HINWEIS

Die reservierten Events, insbesondere Event 4 und 8, sind auf der b maXX controller PLC nicht implementiert.



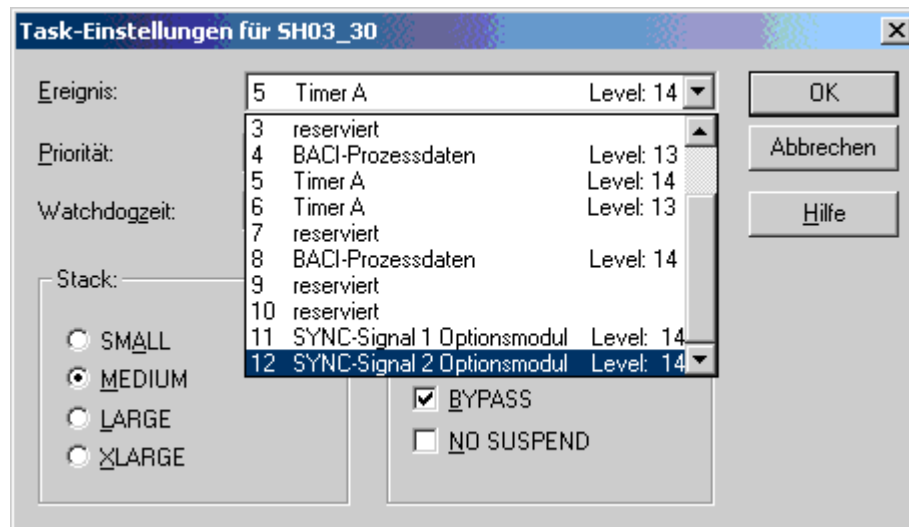


Abbildung 29: Die Ereignisse der b maXX controller PLC Event-Tasks (Ereignisse 4 und 8 sind auf der b maXX controller PLC nicht implementiert)

#### HINWEIS



Alle b maXX controller PLC Tasks mit dem Task-Typ „EVENT“ sind von der Ressource abhängig und benötigen das Attribut **"BYPASS"**. Initialisierung und Aufruf der deklarierten Event-Task sowie deren Priorität erfolgen über die Initialisierungs-FBs innerhalb des Programms. Daher sind Priorität, Watchdogzeit, "SAVE FPU" und "NO SUSPEND" bei Bypass-Event-Tasks ohne Bedeutung.

Ein höherer Interrupt-Level bedeutet höhere Priorität.

Der Stack ist bei mehreren Event-Tasks, die sich gegenseitig unterbrechen können, oder Funktionsbausteinen, die mehrfach geschachtelt sind, auf "Large" bzw. "XLarge" anzupassen.

## 5.5 b maXX controller PLC Anwenderbibliotheken

Die PROPROG wt Anwenderbibliotheken gliedern sich in Firmware und Anwenderbibliotheken, die hardwareunabhängig, b maXX gruppenabhängig oder ganz hardwareabhängig sein können.

Gruppenabhängige Bibliotheken können in mehreren Ressourcen der gleichen b maXX-Gruppe verwendet werden. Hardwareabhängige Bibliotheken können nur in Ressourcen des angegebenen Zielsystems verwendet werden. Die Hardwareabhängigkeit von b maXX controller PLC Bibliotheken ist mit **\*\_C\_PLC01\_\*** in der Bibliotheksbezeichnung gekennzeichnet.

PROPROG wt II: Die Version wird wie folgt angegeben: 20bd00.

- 20 ist der inkompatible Versionsstand
- 00 ist der kompatible Versionsstand.

Der Versionsstand wird bei Kompatibilität der Ein- und Ausgangsvariablen der Funktionsbausteine um eins erhöht, z. B. 20bd01, 20bd02, usw. Wird an einem FB der Bibliothek

z. B. ein Ein- oder Ausgang hinzugefügt, wird der Versionsstand vor dem "bd" um eins erhöht und nach dem "bd" auf Null gesetzt, z.B. 20bd03 auf 21bd00.

ProProg wt III: Die Version wird wie folgt angegeben: 30bd00.

- 30 ist der inkompatible Versionsstand
- 00 ist der kompatible Versionsstand.

Der Versionsstand wird bei Kompatibilität der Ein- und Ausgangsvariablen der Funktionsbausteine um eins erhöht, z. B. 30bd01, 30bd02, usw. Wird an einem FB der Bibliothek z. B. ein Ein- oder Ausgang hinzugefügt, wird der Versionsstand vor dem "bd" um eins erhöht und nach dem "bd" auf Null gesetzt, z.B. 30bd03 auf 31bd00.

Die Anwenderbibliotheken gliedern sich in:

- Firmware: b maXX controller PLC Board Funktionalität, z. B. Starten einer PROPROG wt Bypass Event-Task.
- Datentypen: b maXX controller PLC spezifische zusammengesetzte Datentypen und Felder, z. B. Registerstrukturen von Systemkomponenten wie CANopen-Master.
- Standard-FBs: Elementare FBs für Steuerungstechnik für Antriebe welche über ein Feldbussystem (z. B. CANopen) an die b maXX controller PLC angeschlossen sind.
- Technologiebausteine: Funktionalitäten für Antriebe welche über ein Feldbussystem (z. B. CANopen) an die b maXX controller PLC angeschlossen sind.

### Übersicht von PLC Bibliotheken die vollkommen hardwareunabhängig sind PROPROG wt II:

CD-ROM	Bibliothek	Bemerkung
Standard-Bibliotheken	BM_TYPES_20bd06	Baumüller Datentypen
	UNIVERSAL_20bd01	Basis-Funktionsbausteine
	MC_SYS_20bd01	Firmwarebausteine (werden von anderen Bibliotheken verwendet)

### ProProg wt III:

CD-ROM	Bibliothek	Bemerkung
Standard-Bibliotheken	BM_TYPES_30bd01	Baumüller Datentypen
	UNIVERSAL_30bd00	Basis-Funktionsbausteine
	MC_SYS_30bd00	Firmwarebausteine (werden von anderen Bibliotheken verwendet)

### Übersicht von PLC Bibliotheken die b maXX gruppenabhängig sind und somit sowohl von der b maXX controller PLC als auch von der b maXX drive PLC verwendet werden können (ab dem unten angegebenen Versionsstand) PROPROG wt II:

CD-ROM	Bibliothek	Bemerkung
TB Kurvenscheibe	CAM_PLC01_21bd00	FBs für Technologiebaustein "Kurvenscheibe"
TB Wickler	WINDER_PLC01_20bd01	FBs für Technologiebaustein "Wickler"
CANopen	CANopen_PLC01_20bd03	FBs für Modul BMC-M-ETH-02 bzw. BMC-M-CAN-04 (CANopen-Master) und BMC-M-CAN-03 (CANopen-Slave)
Ethernet	TCP_PLC01_21bd01	FBs für Modul BMC-M-ETH-01 bzw. BMC-M-ETH-02 (Ethernet)

**ProProg wt III:**

CD-ROM	Bibliothek	Bemerkung
TB Kurvenscheibe	CAM_PLC01_30bd00 CAM_PLC02_30bd00	FBs für Technologiebaustein "Kurvenscheibe"
TB Wickler	WINDER_PLC01_30bd00 WINDER_PLC02_30bd00	FBs für Technologiebaustein "Wickler"
CANopen	CANopen_PLC01_30bd00 CANopen_PLC02_30bd00	FBs für Modul BMC-M-ETH-02 bzw. BMC-M-CAN-04 (CANopen-Master) und BMC-M-CAN-03 (CANopen-Slave)
Ethernet	TCP_PLC01_30bd01 TCP_PLC02_30bd01	FBs für Modul BMC-M-ETH-01 bzw. BMC-M-ETH-02 (Ethernet)

**Übersicht von PLC Bibliotheken die hardwareabhängig sind und nur von der b maXX controller PLC verwendet werden können  
PROPROP wt II:**

CD-ROM	Bibliothek	Bemerkung
Standard-Bibliotheken	SYSTEM1_C_PLC01_20bd00	Funktionsbausteine für BMC-M-PLC-01
	SYSTEM2_C_PLC01_20bd00	Firmwarebausteine für BMC-M-PLC-01

**ProProg wt III:**

CD-ROM	Bibliothek	Bemerkung
Standard-Bibliotheken	SYSTEM1_C_PLC01_30bd00	Funktionsbausteine für BMC-M-PLC-01
	SYSTEM1_C_PLC02_30bd02	Funktionsbausteine für BMC-M-PLC-02
	SYSTEM2_C_PLC01_30bd00	Firmwarebausteine für BMC-M-PLC-01
	SYSTEM2_C_PLC02_30bd02	Firmwarebausteine für BMC-M-PLC-02

Der Verzeichnispfad für die Bibliotheken ist unter PROPROP wt, Optionen, Verzeichnisse anzugeben. Alle Bibliotheken werden im Projektbaum von PROPROP wt unter Bibliotheken eingefügt.

Zu jedem FB ist eine HTML-Hilfe aufrufbar, die eine Beschreibung der Ein- und Ausgänge bereitstellt (siehe Programmierhandbuch PROPROP wt II / ProProg wt III).

### 5.5.1 b maXX controller PLC Firmware

Die b maXX controller PLC Firmware besteht aus Funktionsbausteinen (FBs), die über Parameterübergabe mit Funktionen auf der b maXX controller PLC-CPU kommunizieren. Diese FBs sind nur ressourcenabhängig einsetzbar, d. h. vom Zielsystem abhängig.

PROPROG wt II:

Die BMC-M-PLC-01 Firmware ist mit der Bibliothek

**SYSTEM2\_C\_PLC01\_20bd00** (oder höher)

in ein Projekt einzufügen. Die Bibliothek beinhaltet folgenden Funktionsumfang:

- Bypass Event-Task starten, freiprogrammierbare LEDs an der b maXX controller PLC.
- P-, PI-Regler, 48-Bit Division über elektronisches Getriebe, Integrieren, Differenzieren.
- Funktionsbausteine zur Anschaltung an USS<sup>®</sup>- und 3964R<sup>®</sup>-Protokoll.



#### HINWEIS

Die b maXX controller PLC Firmware wird von einigen b maXX controller PLC Anwenderbibliotheken genutzt. Daher kann es erforderlich sein, mit einer b maXX controller PLC Anwenderbibliothek, die geforderte Firmware-Bibliothek SYSTEM2\_C\_PLC01\_20bd00 oder höher einzufügen (siehe Beschreibung der jeweiligen Anwenderbibliothek). Die jeweils aktuellste Version ist auf der PROPROG-Programm-CD integriert und wird bei der Installation von PROPROG wt II automatisch mitinstalliert.

Bei Auswahl der „Vorlage für BMC\_M\_PLC01“ unter „Datei\Neues Projekt“ ist die Bibliothek SYSTEM2\_C\_PLC01\_20bd00 (oder höher) automatisch angelegt. Die Standard-Anwenderbibliothek SYSTEM1\_C\_PLC01\_20bd00 (oder höher) wird im Regelfall ebenfalls in Projekten notwendig sein und wird auf einer eigenen CD ausgeliefert (siehe Übersicht in Abschnitt [►b maXX controller PLC Anwenderbibliotheken◀](#) auf Seite 57).

Dies gilt analog auch für ProProg wt III und die zugehörigen Bibliotheken.

ProProg wt III:

Die BMC-M-PLC-01 Firmware ist mit der Bibliothek

**SYSTEM2\_C\_PLC01\_30bd00** (oder höher)

in ein Projekt einzufügen. Die Bibliothek beinhaltet folgenden Funktionsumfang:

- Bypass Event-Task starten, freiprogrammierbare LEDs an der b maXX controller PLC.
- P-, PI-Regler, 48-Bit Division über elektronisches Getriebe, Integrieren, Differenzieren.
- Funktionsbausteine zur Anschaltung an USS<sup>®</sup>- und 3964R<sup>®</sup>-Protokoll.

Die BMC-M-PLC-02 Firmware ist mit der Bibliothek

**SYSTEM2\_C\_PLC02\_30bd02** (oder höher)

in ein Projekt einzufügen. Die Bibliothek beinhaltet folgenden Funktionsumfang:

- Bypass Event-Task starten, freiprogrammierbare LEDs an der b maXX controller PLC.

- P-, PI-Regler, 48-Bit Division über elektronisches Getriebe, Integrieren, Differenzieren.
- Funktionsbausteine zur Anschaltung an 3964R<sup>®</sup>-Protokoll.

## 5.5.2 b maXX controller PLC Board Funktionen

---

Die jeweilige Firmware-Bibliothek SYSTEM2\_xxx\_yybdzz enthält Funktionsbausteine (FBs) zur Kontrolle von Event Signalen für Interrupts und Board-LEDs. Um b maXX controller PLC Event-Tasks (Bypass) zu initialisieren und zu starten wird der Funktionsbaustein INTR\_SET benutzt. Um freiprogrammierbare LEDs zu nutzen wird der Funktionsbaustein LED8 (bei BMC-M-PLC-01) bzw. LED12 (bei BMC-M-PLC-02) benutzt.

Für die Optimierung von Code-Laufzeiten innerhalb von b maXX controller PLC Ressourcen werden die zwei folgenden FBs zur Code-Laufzeitmessung bereitgestellt.

PROPROG wt II: BMC-M-PLC-01:

Die FBs werden über die Anwenderbibliothek **SYSTEM1\_C\_PLC01\_20bd00** und höher eingefügt.

ProProg wt III: BMC-M-PLC-01:

Die FBs werden über die Anwenderbibliothek **SYSTEM1\_C\_PLC01\_30bd00** und höher eingefügt.

ProProg wt III: BMC-M-PLC-02:

Die FBs werden über die Anwenderbibliothek **SYSTEM1\_C\_PLC02\_30bd02** und höher eingefügt.

Der zu messende Code-Block innerhalb einer Task ist mit der Platzierung der FBs TIME\_MEASURE\_START und TIME\_MEASURE\_END einzugrenzen. Das Ergebnis der Laufzeit des gemessenen Code-Blocks wird am FB TIME\_MEASURE\_END als Zeitdifferenz in  $\mu\text{s}$  ausgegeben (siehe auch Online Beschreibung der FBs TIME\_MEASURE\_START und TIME\_MEASURE\_END). Mit diesen Bausteinen können Laufzeiten bis max. 10 ms (BMC-M-PLC-01) bzw. 15 min (BMC-M-PLC-02) ausgemessen werden.

### 5.5.2.1 Der Funktionsbaustein INTR\_SET

---

Der Funktionsbaustein INTR\_SET startet in einer Anlauf-Task eine Bypass Event-Task.

Die PROPROG wt Event-Task mit dem Programm muss auf das Ereignis und auf das Attribut "Bypass" gestellt werden.

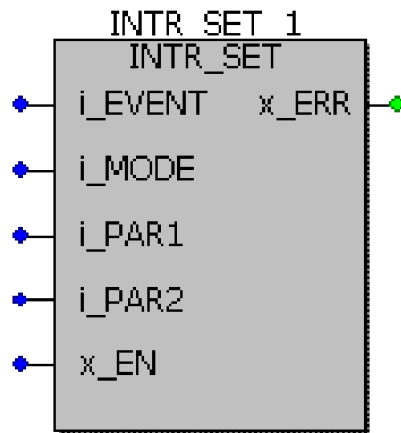


Abbildung 30: Bypass Event-Task über Funktionsbaustein INTR\_SET initialisieren und freigeben.

Parameter	Eingang	Wertebereich
i_EVENT	Interrupt-Hardware-Programmnummer	8 bit signed
i_MODE	Reserve	16 bit signed
i_PAR1	CPU-Timer(1,2)-Wert-Multiplikator zur Basis 50µs	16 bit signed
i_PAR2	Reserve	16 bit signed
x_EN	Sperren/Freigabe des Interrupts	1 bit

Parameter	Ausgang	Wertebereich
x_ERR	Fehler-Bit	1 bit

### Beschreibung:

Mit dem FB INTR\_SET kann der Anwender diverse systemeigene Interrupt-Quellen konfigurieren und aktivieren. Diese Interrupts können dann im Programm zur Event-Task Aktivierung benutzt werden.

Am Eingang i\_EVENT muss eine Ereignis Nummer angeschlossen werden. Diese Nummer spezifiziert die Interrupt-Quelle der Event-Task. Handelt es sich um einen CPU-Timer-Interrupt muss zusätzlich ein Faktor zur Zeitbasis 50 µs am Eingang i\_PAR1 angegeben werden.

Mit x\_EN = FALSE kann ein vorher aktivierter Interrupt gesperrt werden.

Verzeichnis der Ereignis Nummern für Event-Tasks:

Event	Bezeichnung	Level
0	CPU-Timer 1	14
2	CPU-Timer 2	13
5	Timer A	14
6	Timer A	13
11	Sync-Signal 1 Optionsmodul	14
12	Sync-Signal 2 Optionsmodul	14

Die Ereignis-Nummer am Eingang i\_Event muss identisch mit der Einstellung der Event-Task innerhalb der Ressource sein. Das Attribut "Bypass" muss aktiviert sein.



#### HINWEIS

Ein in der Priorität höherer Interrupt unterbricht einen niedrigeren. Für die Ereignisse 0 und 2, CPU-Timer 1 (oder 2) Interrupt, ist zusätzlich am Eingang i\_PAR1 der Faktor für die Zeitbasis 50 µs anzugeben.

Der höherpriore Timer "CPU-Timer 1" (i\_EVENT = 0) und der niederpriore Timer "CPU-Timer 2" (i\_EVENT = 2) können gleichzeitig verwendet werden.

Der "Board-Timer A" hohe Priorität (i\_EVENT = 5) und "Board-Timer A" niedrige Priorität (i\_EVENT = 6) sind nicht gleichzeitig verwendbar.

#### Beispiel 1:

Die Anwender-POE (z. B. mit dem Namen "Timer") soll mit dem höherprioren CPU-Timer 1 (i\_EVENT = 0) verknüpft werden und zyklisch mit 5 ms Intervall aufgerufen werden.

#### Umsetzung:

In der Kalt- und Warmstart-Task wird eine INI-POE aufgerufen, die das konfigurierte INTR\_SET enthält.

Für das Timer-Intervall gilt folgende Formel:

$$\text{Timer-Intervall} = i\_PAR1 \cdot 50 \mu\text{s}$$

Für das gewünschte Intervall von 5 ms ergibt sich:  $i\_PAR1 := (5000 \mu\text{s} / 50\mu\text{s}) = \text{INT}\#100$ .

Darstellung des konfigurierten und freigegebenen INTR\_SET:

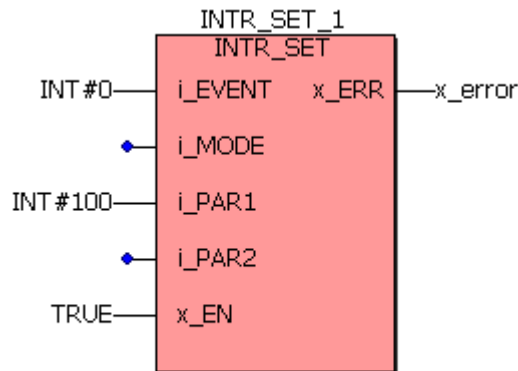


Abbildung 31: Funktionsbaustein INTR\_SET: Start eines CPU Timer 1-Interrupts mit dem Aufrufintervall 5 ms.

Damit wird der Interrupt während des SPS-Startvorgangs eingerichtet und aktiviert.

Die Anwender-POE "Timer" muss in der Gruppe Task innerhalb der BMC\_M\_PLC0x-Ressource als Task-Typ "EVENT" mit dem Ereignis "CPU\_Timer 1" mit der Einstell-Option "BYPASS" verknüpft werden:

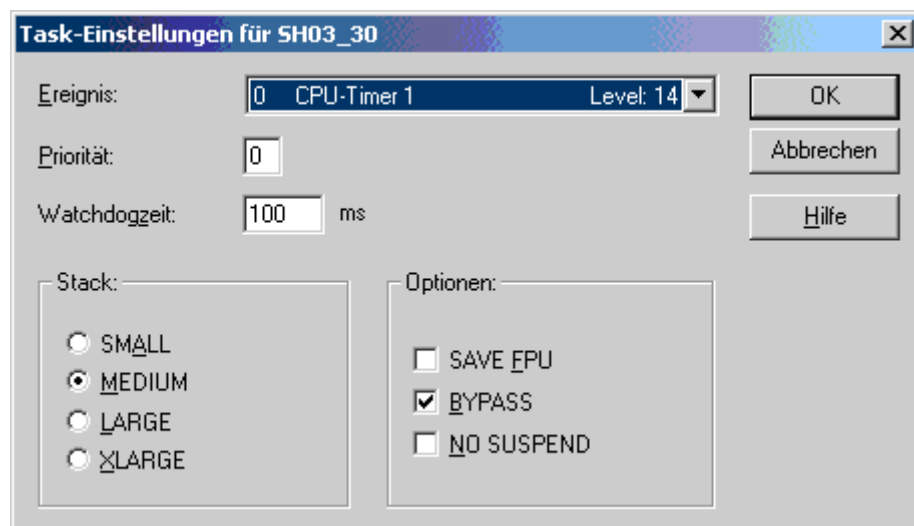


Abbildung 32: Verknüpfung mit dem Ereignis "CPU-Timer 1" als Bypass-Task.

### Beispiel 2:

Start des „Timer A“ -Interrupts mit 5 ms und Generierung des Triggersignals am CBPB-Signal-Ausgang "TRIGGER1":

Im Gegensatz zum CPU-Timer 1 (oder 2) kann der Board-Timer A auch als Triggersignal für Module (Systemkomponenten links von der PLC) genutzt werden, die die Triggersignale benötigen. Denn nur mit dem „Timer A“ kann sowohl ein zyklischer Interrupt ausgelöst werden als auch eine Triggerung erfolgen.

BMC-M-PLC-01:

Zunächst erfolgt die Einrichtung der Event-Task innerhalb der BMC\_M\_PLC01-Ressource über das Event „Timer A“ mit der Ereignisnummer 5 (oder 6).



Innerhalb der Kalt/Warmtask wird zunächst zweimal der FB „TIMER\_A\_INIT“ (siehe Standard Bibliothek „SYSTEM1\_C\_PLC01\_20bd00“ (PROPROG wt II) bzw. „SYSTEM1\_C\_PLC01\_30bd00“ (ProProg wt III) oder höher) und dann der FB „INTR\_SET“ aufgerufen.

- Mit dem 1. Aufruf von „TIMER\_A\_INIT“ wird der Timer für 5 ms eingerichtet.
- Mit dem 2. Aufruf wird das generierte 5 ms-Signal auf den CBPB-Ausgang „TRIGGER1“ gelegt, das dann von den Optionsmodulen verarbeitet werden kann.
- Mit dem Aufruf „INTR\_SET“ wird dann der Timer A - Interrupt eingerichtet und aktiviert, so dass der Anwender seine POE synchron zum generierten 5 ms-Triggersignal ablaufen lassen kann.

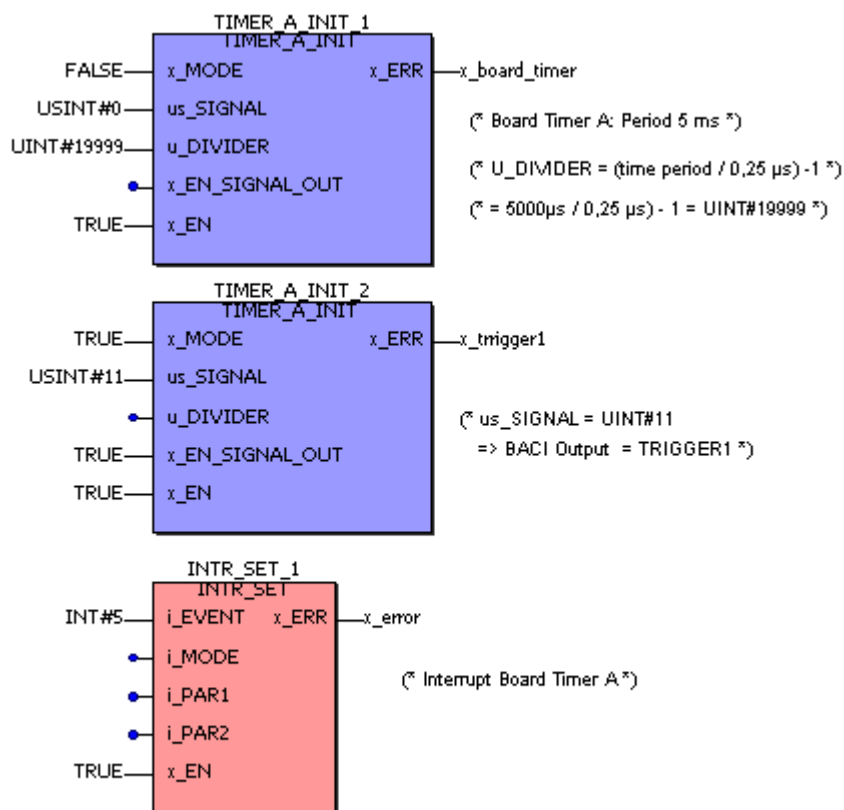


Abbildung 33: Start der "Timer A" Bypass Event-Task mit der Periode 5 ms und gleichzeitige Nutzung von "Timer A" als Triggersignal für Systemkomponenten (die links der b maXX controller PLC gesteckt sind) auf BMC-M-PLC-01.

#### BMC-M-PLC-02:

Zunächst erfolgt die Einrichtung der Event-Task innerhalb der BMC\_M\_PLC02-Ressource über das Event „Timer A“ mit der Ereignisnummer 5 (oder 6).

Innerhalb der Kalt/Warmtask wird zunächst zweimal der FB „TIMER\_A\_INIT“ (siehe Standard Bibliothek „SYSTEM1\_C\_PLC02\_30bd02“ oder höher) und dann der FB „INTR\_SET“ aufgerufen.

- Mit dem 1. Aufruf von „TIMER\_A\_INIT“ wird der Timer für 5 ms eingerichtet.
- Mit dem 2. Aufruf wird das generierte 5 ms-Signal auf den CBPB-Ausgang „TRIGGER1“ gelegt, das dann von den Optionsmodulen verarbeitet werden kann.

- Mit dem Aufruf „INTR\_SET“ wird dann der Timer A - Interrupt eingerichtet und aktiviert, so dass der Anwender seine POE synchron zum generierten 5 ms-Triggersignal ablaufen lassen kann.

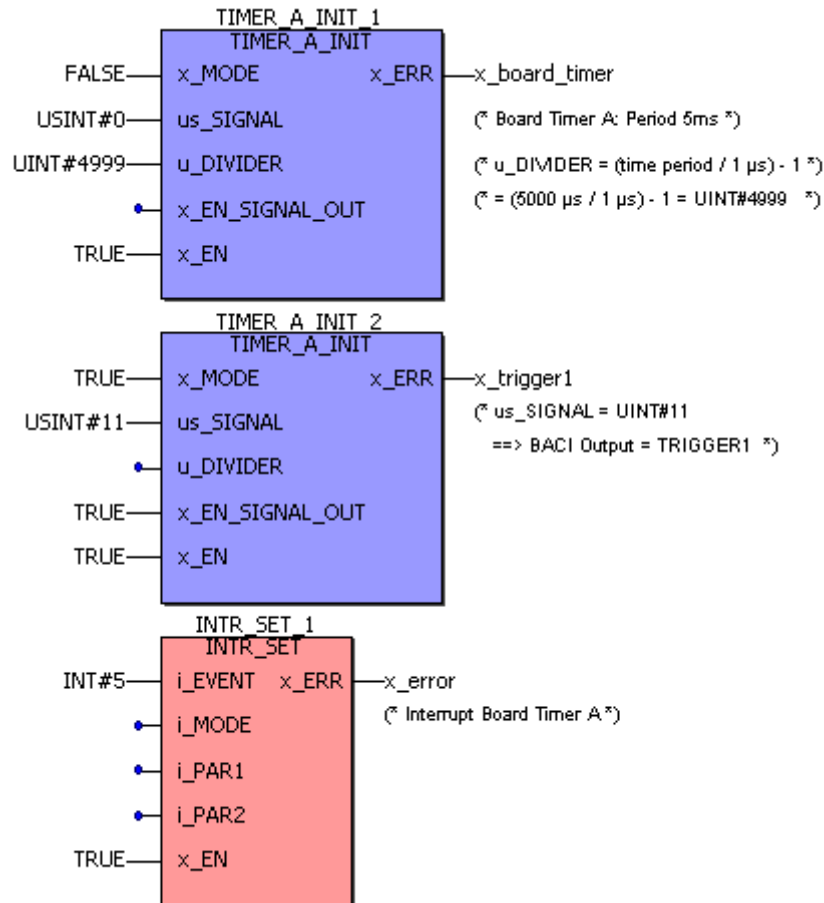


Abbildung 34: Start der "Timer A" Bypass Event-Task mit der Periode 5 ms und gleichzeitige Nutzung von "Timer A" als Triggersignal für Systemkomponenten (die links der b maXX controller PLC gesteckt sind) auf BMC-M-PLC-02.

### 5.5.2.2 Der Funktionsbaustein LED8 für BMC-M-PLC-01

Über den Funktionsbaustein LED8 aus der Firmware-Bibliothek „SYSTEM2\_C\_PLC01\_20bd00“ (PROPROG wt II) bzw. „SYSTEM2\_C\_PLC01\_30bd00“ (ProProg wt III) oder höher, sind die LEDs auf der b maXX controller PLC programmierbar.

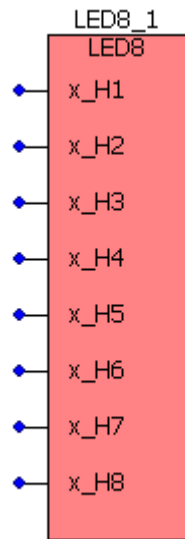


Abbildung 35: Funktionsbaustein LED8 für BMC-M-PLC-01

Parameter	Eingang	Datentyp
x_H1	LED H1	BOOL
x_H2	LED H2	BOOL
x_H3	LED H3	BOOL
x_H4	LED H4	BOOL
x_H5	LED H5	BOOL
x_H6	LED H6	BOOL
x_H7	LED H7	BOOL
x_H8	LED H8	BOOL

**Beschreibung:**

Die linken LEDs der b maXX controller PLC BMC-M-PLC-01 leuchten grün, die rechten LEDs leuchten rot.

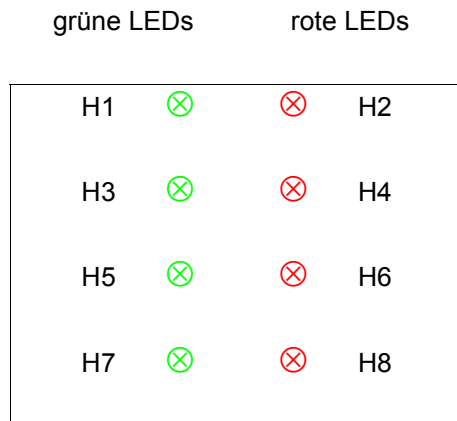


Abbildung 36: LEDs der b maXX controller PLC (BMC-M-PLC-01)

### 5.5.2.3 Der Funktionsbaustein LED12 für BMC-M-PLC-02

Über den Funktionsbaustein LED12 aus der Firmware-Bibliothek „SYSTEM2\_C\_PLC02\_30bd02“ oder höher, sind die LEDs auf der b maXX controller PLC programmierbar.

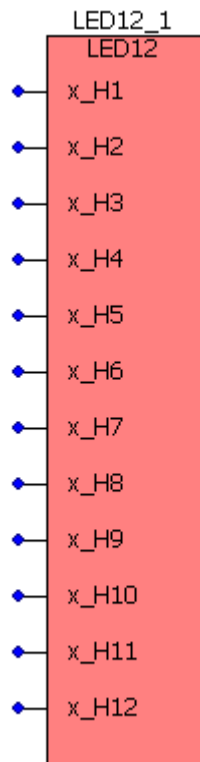


Abbildung 37: Funktionsbaustein LED12 für BMC-M-PLC-02

Parameter	Eingang	Datentyp
x_H1	LED H1	BOOL
x_H2	LED H2	BOOL
x_H3	LED H3	BOOL
x_H4	LED H4	BOOL
x_H5	LED H5	BOOL
x_H6	LED H6	BOOL
x_H7	LED H7	BOOL
x_H8	LED H8	BOOL
x_H9	LED H9	BOOL
x_H10	LED H10	BOOL
x_H11	LED H11	BOOL
x_H12	LED H12	BOOL

**Beschreibung:**

Das Modul b maXX controller PLC weist als Anzeigeelemente zwölf LEDs (fünf grüne (H1, H3, H5, H7, H11), fünf rote (H2, H4, H6, H8, H10), eine blaue (H9) und eine orange/ grüne (H12)) auf.

## LEDs

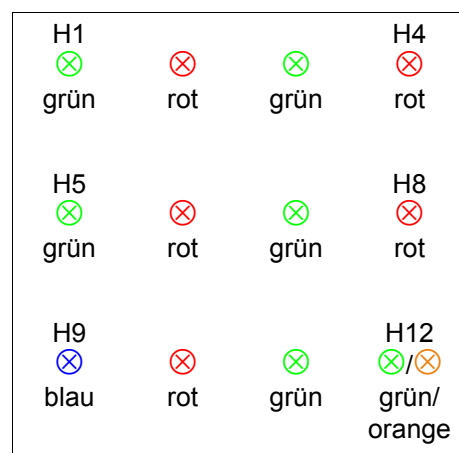


Abbildung 38: LEDs der b maXX controller PLC (BMC-M-PLC-02)

**5.5.3 Die b maXX controller PLC Datentypen**

In einem PROPROG wt Projekt erfolgt der Datenaustausch zwischen der b maXX controller PLC und den Systemkomponenten an der b maXX controller PLC über Variablen vom entsprechenden Datentyp. Die Anwenderbibliothek BM\_TYPES\_20bd06 (PROPROG wt II) bzw. BM\_TYPES\_30bd01 (ProProg wt III) oder höher stellt bereits eine Vielzahl von vorgefertigten Datentypen (Strukturen und Arrays) zur Verfügung. Die-

se strukturierten Datentypen werden in der Regel in den ebenfalls vorgefertigten Standard- und Technologie-Bibliotheken verwendet, um somit auf sehr einfache Weise mit den Systemkomponenten kommunizieren zu können.

Dazu sind in den Templates BMC\_M\_PLC01 ("Datei/Neues Projekt.../Vorlage für BMC\_M\_PLC01") bzw. BMC\_M\_PLC02 ("Datei/Neues Projekt.../Vorlage für BMC\_M\_PLC02") bereits die wichtigsten Variablen für die verschiedenen Funktionen und Steckplätze der einzelnen Systemkomponenten vordefiniert worden.

Der Anwender braucht somit nur noch die steckplatzabhängige Variable aus dem Template für sein gestecktes Modul auswählen und an die entsprechenden Ein- und Ausgänge der eingesetzten Funktionsbausteine (die in den verschiedenen Bibliotheken zu den Systemkomponenten zur Verfügung stehen) anschließen.

In den Bibliotheken wird an den relevanten Stellen auf die eingesetzten Datentypen der BM\_TYPES\_20bd06 (PROPROG wt II) bzw. BM\_TYPES\_30bd01 (ProProg wt III) oder höher verwiesen.

Die vordefinierten Datentypen in BM\_TYPES\_20bd06 (PROPROG wt II) bzw. BM\_TYPES\_30bd01 (ProProg wt III) oder höher können und sollen selbstverständlich auch vom Anwender verwendet werden:

Das Arbeitsblatt der BM\_TYPES\_20bd06 (PROPROG wt II) bzw. BM\_TYPES\_30bd01 (ProProg wt III) ist im Projektbaum nicht direkt aufrufbar; die darin enthaltenen Datentypen können jedoch eingesehen werden, indem der Projektbaum-Editor über die an der Fußzeile des Editors angebrachten Reiter von "Projekt" auf "Bibliotheken" umgestellt und auf das Arbeitsblatt BM\_TYPES\_20bd06 (PROPROG wt II) bzw. BM\_TYPES\_30bd01 (ProProg wt III) doppelgeklickt wird:

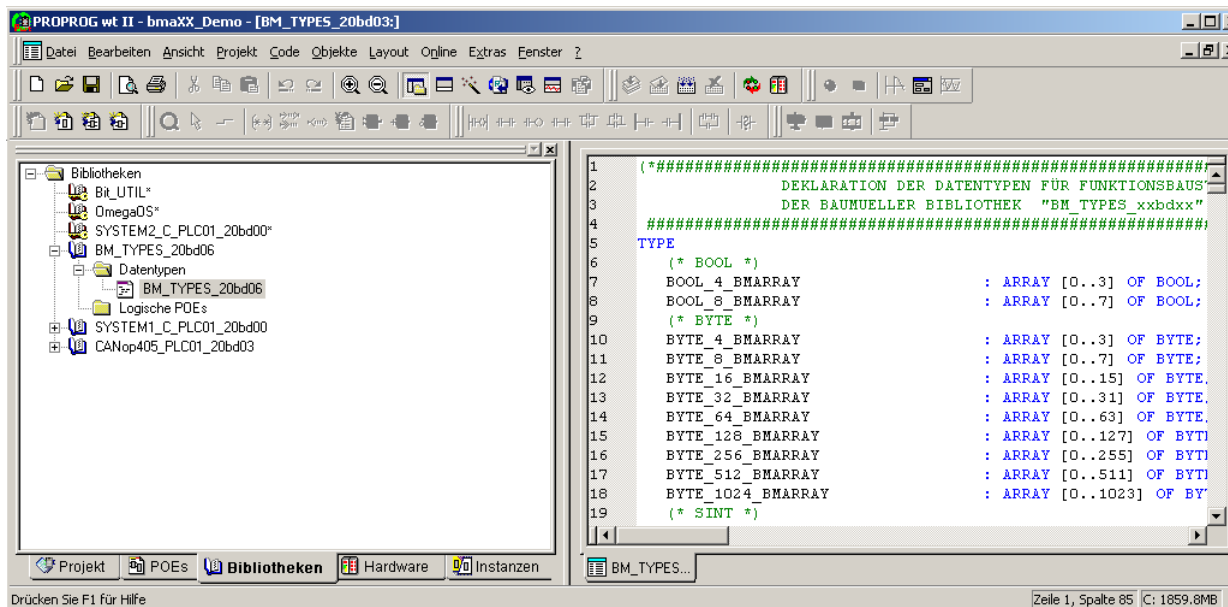


Abbildung 39: Einblick in die in Ihrem Projekt eingebundenen Baumüller-Datentypen "BM\_TYPES\_20bd06" in PROPROG wt II.

Auf diese Datentypen kann man dann bei der Vergabe von Variablen über den Variablen-Dialog zugreifen.

PROPROG wt II:

In PROPROG wt II klicken im Variablendialog auf den Button „Eigenschaften...“. Das Fenster „Automatische Variablen-Deklaration“ wird geöffnet:

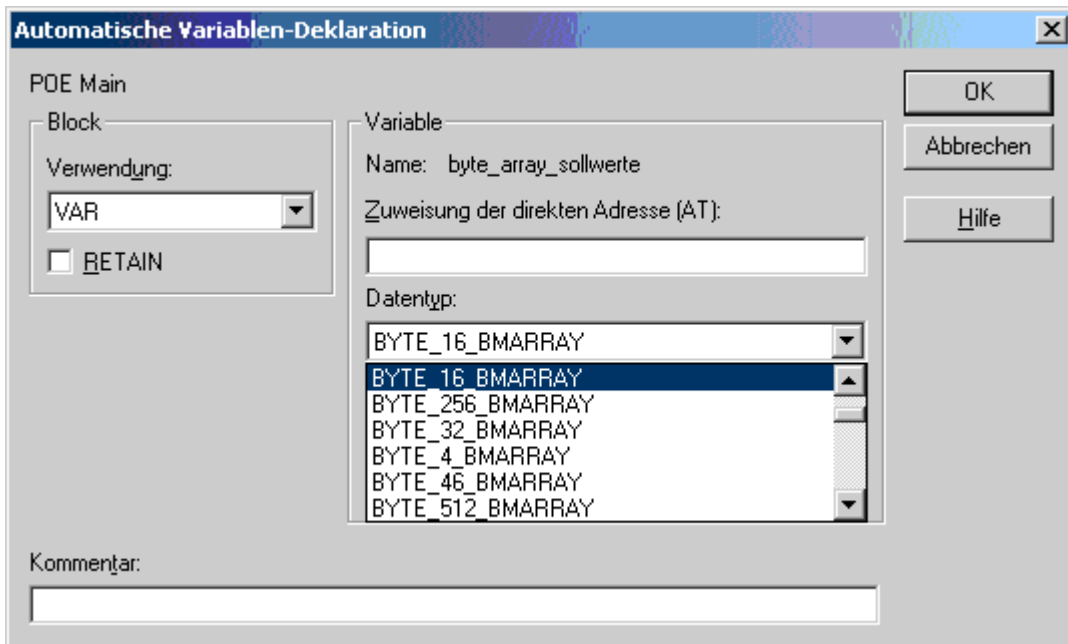


Abbildung 40: Automatische Variablen-Deklaration.

ProProg wt III:

In ProProg wt III ist die automatische Variablendeklaration im Variablendialog integriert.

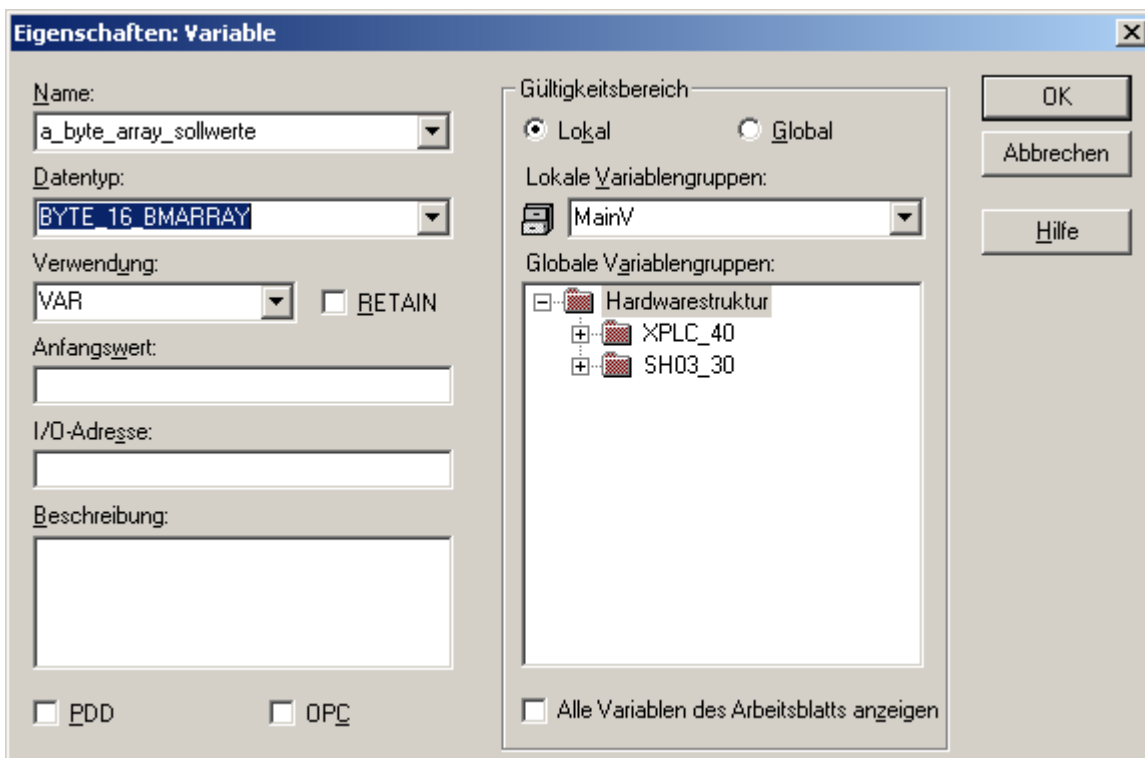


Abbildung 41: Variablendeklaration

Wenn Sie ein PROPROGRAMM Projekt erstellen und dazu das Template BMC\_M\_PLC01 bzw. BMC\_M\_PLC02 verwenden, sind im globalen Variablen-Arbeitsblatt bereits die Variablen für den Datenaustausch deklariert. Für die Module (Systemkomponenten links der PLC) finden Sie eine (manchmal auch zwei) Variablen pro Modul-Nummer.

### Beispiel:

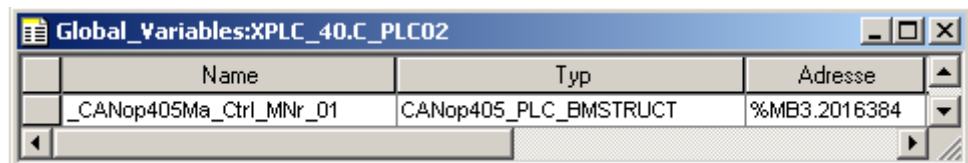
Die Systemkomponente Ethernet mit CANopen-Master (BMC-M-ETH-02) benötigt zur Initialisierung Einstellungen in verschiedenen Registern.

Mit der Systemkomponente mit Modul-Nummer 1 ergibt sich folgende (im Template bereits vorgefertigte) Variablendeklaration:

PROGRAMM II:

```
_CANop405Ma_Ctrl_MNr_01 AT %MB3.2016384 : CANop405_PLC_BMSTRUCT;
```

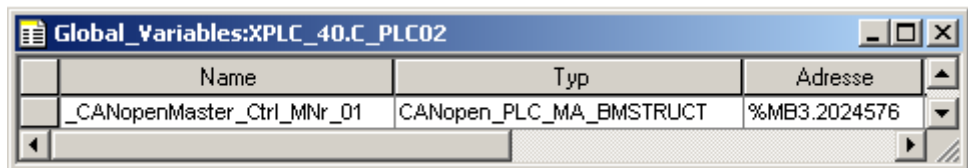
ProPROGRAMM III:



Name	Typ	Adresse
_CANop405Ma_Ctrl_MNr_01	CANop405_PLC_BMSTRUCT	%MB3.2016384

Abbildung 42: Variablendeklaration

ProPROGRAMM III mit ProMaster:



Name	Typ	Adresse
_CANopenMaster_Ctrl_MNr_01	CANopen_PLC_MA_BMSTRUCT	%MB3.2024576

Abbildung 43: Variablendeklaration mit ProMaster

Der komplette Registeraufbau der Struktur CANop405\_PLC\_BMSTRUCT ist mit seinen Elementen und verwendeten Datentypen in den BM\_TYPES\_20bd06 (PROGRAMM II) bzw. BM\_TYPES\_30bd01 (ProPROGRAMM III) oder höher hinterlegt (Auszug).



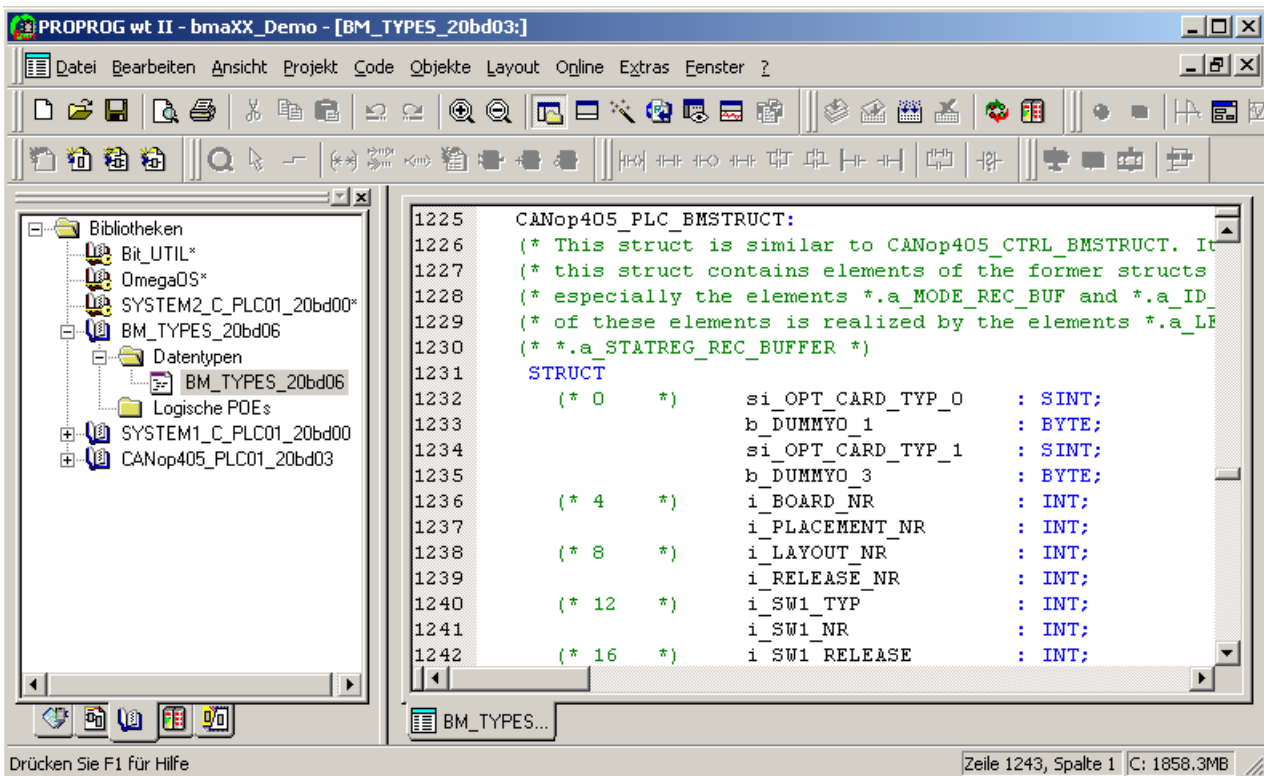


Abbildung 44: Auszug aus dem Registeraufbau der Struktur CANop405\_PLC\_BMSTRUCT

Die Elemente der Variable `_CANop405Ma_Ctrl_MNr_01` bzw. `_CANopenMaster_Ctrl_MNr_01` bilden jetzt die Register der Systemkomponente mit der Modul-Nummer 1 ab. Weitere Informationen zum Einsatz der Systemkomponente Ethernet mit CANopen-Master entnehmen Sie bitte der zugehörigen Betriebsanleitung und dem zugehörigen Applikationshandbuch.

#### 5.5.4 Die Standard Funktionsbaustein-Bibliotheken

Die Standardbibliotheken enthalten Funktionsbausteine (FBs) mit Grundfunktionalitäten einer antriebsnahen Programmierung und Projektierung des Echtzeitverhaltens.

Diese Funktionsbausteine finden sich in:

PROPROGRAMM II:

- **SYSTEM1\_C\_PLC01\_20bd00** (oder höher)
  - Funktionsbaustein für CBPB (Input, Timer, Output, Trigger)
  - USS<sup>®</sup>-Protokoll-Anschaltung
  - Code-Laufzeitmessung
- **SYSTEM2\_C\_PLC01\_20bd00** (oder höher)
  - b maXX controller PLC Firmware
  - Firmwarebaustein LED8
  - Firmwarebaustein INTR\_SET
  - 3964R<sup>®</sup>-Protokoll-Anschaltung

- Firmwarebausteine für Terminalkommunikation TER\_x
- **UNIVERSAL\_20bd01** oder höher (unabhängig von der Hardware)
  - Extrapolatoren, Rampengeneratoren, Positionsgeneratoren, Min-, Max- und Begrenzungs-Funktionsbausteine
  - Funktionsbaustein Virtuelle Leitachse TRAJECTORY\_GEN1
  - Antriebs-Zustand und -Kontrolle durch den Funktionsbaustein DRIVE1, z. B. für Antriebe welche über ein Feldbussystem wie CANopen an die b maXX controller PLC angeschlossen sind.

ProProg wt III:

- **SYSTEM1\_C\_PLC01\_30bd00** (oder höher; für BMC-M-PLC-01)
  - Funktionsbaustein für CBPB (Input, Timer, Output, Trigger)
  - USS<sup>®</sup>-Protokoll-Anschaltung
  - Code-Laufzeitmessung
- **SYSTEM2\_C\_PLC01\_30bd00** (oder höher; für BMC-M-PLC-01)
  - b maXX controller PLC Firmware
  - Firmwarebaustein LED8
  - Firmwarebaustein INTR\_SET
  - 3964R<sup>®</sup>-Protokoll-Anschaltung
  - Firmwarebausteine für Terminalkommunikation TER\_x
- **SYSTEM1\_C\_PLC02\_30bd02** (oder höher; für BMC-M-PLC-02)
  - Funktionsbaustein für CBPB (Input, Timer, Output, Trigger)
  - Code-Laufzeitmessung
- **SYSTEM2\_C\_PLC02\_30bd02** (oder höher; für BMC-M-PLC-02)
  - b maXX controller PLC Firmware
  - Firmwarebaustein LED12
  - Firmwarebaustein INTR\_SET
  - 3964R<sup>®</sup>-Protokoll-Anschaltung
  - Firmwarebausteine für Terminalkommunikation TER\_x
- **UNIVERSAL\_30bd00** oder höher (unabhängig von der Hardware)
  - Extrapolatoren, Rampengeneratoren, Positionsgeneratoren, Min-, Max- und Begrenzungs-Funktionsbausteine
  - Funktionsbaustein Virtuelle Leitachse TRAJECTORY\_GEN1
  - Antriebs-Zustand und -Kontrolle durch den Funktionsbaustein DRIVE1, z. B. für Antriebe welche über ein Feldbussystem wie CANopen an die b maXX controller PLC angeschlossen sind.

### 5.5.5 Die b maXX controller PLC Technologiebausteine

Die Standard-Anwenderbibliotheken lassen sich um komplette Antriebsfunktionalitäten, den Technologiebausteinen, erweitern. Diese sind:

- Technologiebaustein Kurvenscheibe:      PROPROG wt II: Anwenderbibliothek  
CAM\_PLC01\_21bd00 (oder höher)

ProProg wt III: Anwenderbibliothek

- Technologiebaustein Wickler:

CAM\_PLC01\_30bd00 (oder höher)  
CAM\_PLC02\_30bd00 (oder höher)  
PROPROG wt II: Anwenderbibliothek  
WINDER\_PLC01\_20bd01 (oder höher)

ProProg wt III: Anwenderbibliothek  
WINDER\_PLC01\_30bd01 (oder höher)  
WINDER\_PLC02\_30bd01 (oder höher)

Die Technologiebausteine bieten Antriebsfunktionalitäten, die durch die Beschaltung und mehrfache Instanziierung eine Vielzahl von Applikationslösungen bereitstellen.



#### HINWEIS

Alle Anwenderbibliotheken der Technologiebausteine benötigen zur Einbindung die Datentypen ab BM\_TYPES\_20bd06 (PROPROG wt II) bzw. BM\_TYPES\_30bd01 (ProProg wt III).

#### 5.5.6 Einfügen einer Anwenderbibliothek in ein Projekt

Anwenderbibliotheken werden in PROPROG wt im Projektbaum unter Bibliotheken eingefügt. Handelt es sich um Firmware, ist bei der Anwahl das Dateiformat .fwl einzustellen.

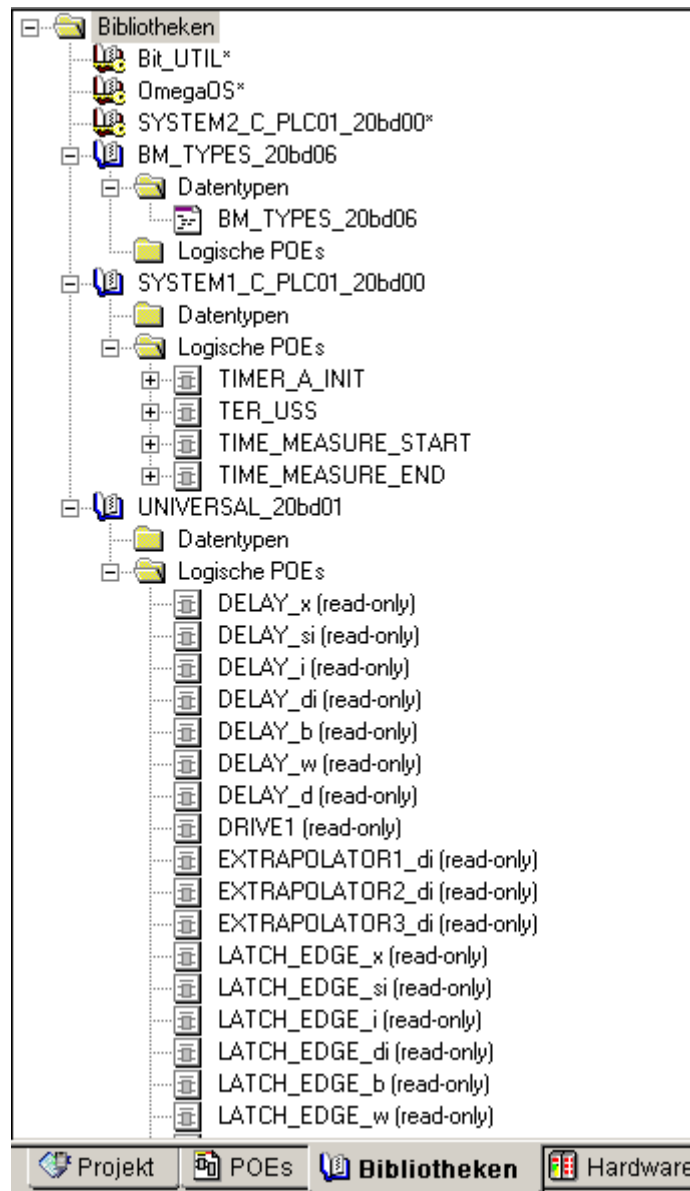


Abbildung 45: Standard-Auswahl von Anwenderbibliotheken, Firmware und Datentypen mit dem Filter Bibliotheken

### HINWEIS



Die Auslieferung einer Bibliothek für PROPROG wt erfolgt gepackt (ZWT-File). Das ZWT-File ist unter PROPROG wt zu entpacken. Beim Entpacken wird die schreibgeschützte Bibliothek automatisch im angegebenen PROPROG wt Bibliothekspfad (Menü Extras>Optionen) abgelegt. In PROPROG wt II erscheint ein "Untitled"-Projekt, welches ohne Speicherung zu schließen ist. In ProProg wt III wird ein Projekt mit dem gleichen Namen wie das ZWT-File angelegt. Dieses Projekt wird nicht weiter benötigt.

Die Firmware-Bibliotheken werden dabei automatisch in das Firmware-Bibliotheks-Verzeichnis von PROPROG wt entpackt.

## 5.6 CBPB-Systembeschreibung für die b maXX controller PLC

### 5.6.1 Direkter Zugriff auf die CBPB-HW von der b maXX controller PLC aus

Der CBPB ist ein Bussystem mit mehreren Steckplätzen, zwischen denen Daten ausgetauscht werden müssen. Jedes Modul ist dabei neben den CBPB-Signalen in der Regel über ein DPRAM (=Dual Ported RAM) an den gemeinsamen CBPB-BUS angeschlossen.

Es wird zwischen CBPB-Master (b maXX controller PLC) und CBPB-Slave unterschieden. Nur der CBPB-Master darf auf die Systemkomponenten über den CBPB-Bus zugreifen. Zugriffe von den Systemkomponenten untereinander oder auf den CBPB-Master sind nicht möglich. Jede einzelne Systemkomponente kann die vom CBPB-Master geschriebenen Daten über das eigene DPRAM abfragen und auswerten.

Übersicht der Modul-Nummern der Systemkomponenten im b maXX System:

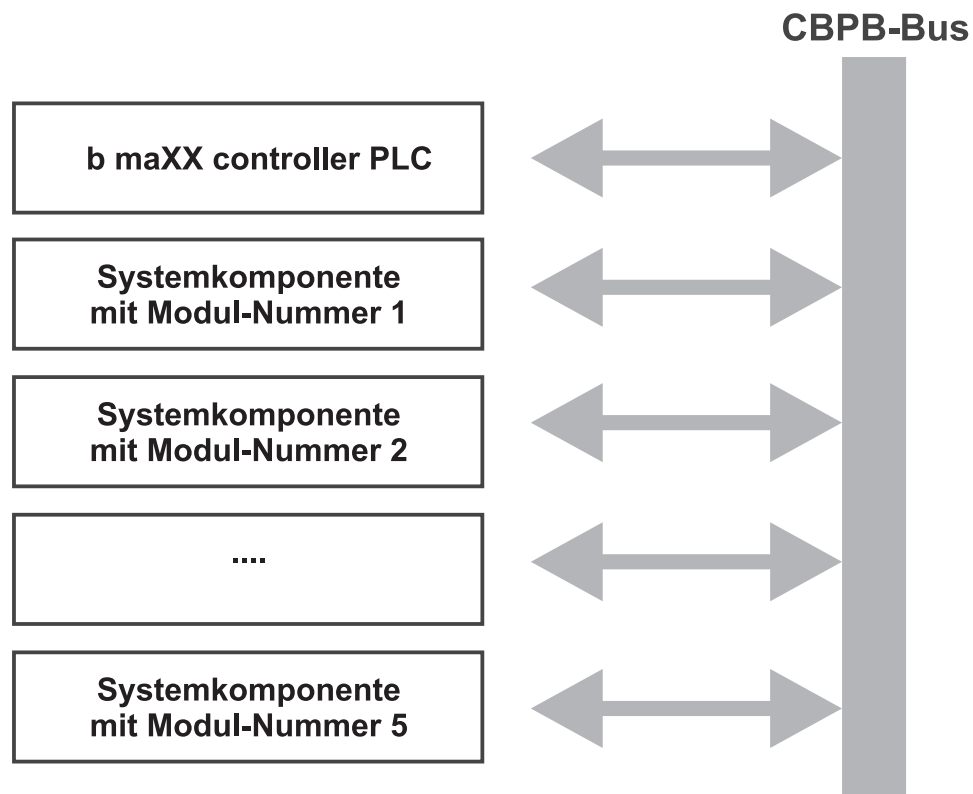


Abbildung 46: Übersicht der Adressen der Systemkomponenten

Aus Sicht der b maXX controller PLC können in der Applikation die Systemkomponenten mit der CBPB-Modul-Nummer 1 bis 5 angesprochen werden, um mit den Systemkomponenten zu kommunizieren.

#### HINWEIS



Sie können maximal 5 Systemkomponenten links der b maXX controller PLC stecken. Jede Systemkomponente erhält eine Modul-Nummer aus dem Bereich 1 bis 5. Die Nummernvergabe ist unabhängig von der Reihenfolge der Systemkomponenten.

Modul-Nummer	Zugeordneter IEC 61131-3 - Bereich (max.)
1	%MB 3.20000000 - %MB 3.20262143
2	%MB 3.30000000 - %MB 3.30262143
3	%MB 3.40000000 - %MB 3.40262143
4	%MB 3.50000000 - %MB 3.50262143
5	%MB 3.60000000 - %MB 3.60262143



### HINWEIS

Der Zugriff auf eine Systemkomponenten mit einer der oben genannten Modul-Nummern kann immer nur von der b maXX controller PLC aus erfolgen.

Eine Software-Struktur, die den Speicher- und Register-Aufbau einer Systemkomponente beschreibt und die in einer vorgefertigten Bibliothek verwendet wird, kann dann in der Applikation einer Variable zugewiesen werden, die der verwendeten Modul-Nummer zugeordnet ist.

Solche globalen Variablen sind bereits in der BMC\_M\_PLC01 - bzw. BMC\_M\_PLC02 - Vorlage in PROPROG wt im globalen Variablenblatt "Global\_Variables" entsprechend vordefiniert worden, so dass der Anwender diese je nach eingesetzten Systemkomponenten und der verwendeten Modul-Nummer nur noch an die eingesetzten Bibliotheksbausteine anschließen muss. In Zukunft werden diese Aufgaben vom Konfigurator automatisch übernommen, der in einer späteren Ausbauphase von PROPROG wt für die b maXX controller PLC integriert sein wird, so dass der Anwender dann mit den oben genannten IEC 61131-3-Bereichen nicht mehr in Berührung kommen wird.

Weitere Erläuterungen zu Strukturen und Funktionen der Systemkomponenten finden sich in den Betriebsanleitungen und Applikationshandbüchern der jeweiligen Module.

### Auslesen der Adresse (S1, S2, S3) des Netzteils für b maXX controller PLC

Die Systemkomponente „Netzteil für b maXX controller PLC“ weist zum Einstellen einer Adresse/Nummer die Drehschalter S1, S2 und S3 auf (siehe Betriebsanleitung Netzteil für b maXX controller PLC). Diese Adresse/Nummer kann vom Applikationsprogramm auf der b maXX controller PLC ausgelesen werden.

PROPROG wt II:

Das Auslesen der am Netzteil eingestellten Adresse S1/S2/S3 erfolgt über die bereits im Template BMC\_M\_PLC01 angelegte globale Variable w\_PSB\_S1\_S2\_S3.

```
w_PSB_S1_S2_S3 AT %MW3.170002 : WORD
```

ProProg wt III:

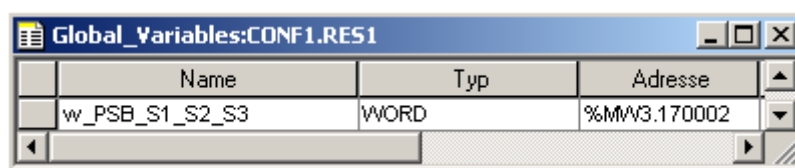


Abbildung 47: Auslesen der Adresse (S1, S2, S3) des Netzteils für b maXX controller PLC

Im Applikationsprogramm können Sie dann gegebenenfalls Teile Ihrer modularisierten Software in Abhängigkeit des Wertes von `w_PSB_S1_S2_S3` aktivieren oder deaktivieren.

Beispiel für die Verwendung der Netzteiladresse als b maXX System Adresse:

Ihr lokales b maXX System besteht aus einer b maXX controller PLC, dem Netzteil für b maXX controller PLC und weiteren Systemkomponenten/Modulen.

Die drei Netzteil-Drehschalter wollen sie nun wie folgt verwenden:

S1 soll die einzelne Maschine kennzeichnen (z. B. Wert „1“ für die Maschine 1), S2 das Maschinenmodul (z. B. „2“ für Maschinenmodul 2) und S3 das lokale b maXX System Nr. 3.

Folglich stellen Sie am Netzteil für die b maXX controller PLC die Drehschalter S1 auf 1, S2 auf 2 und S3 auf 3.

Auf der b maXX controller PLC ergibt sich dann für die globale Variable `w_PSB_S1_S2_S3` der Wert `WORD#16#0123`.

### 5.6.2 Prozessdatenkommunikation zwischen b maXX controller PLC und Systemkomponenten, Synchronisierung auf eine gemeinsame CBPB-Hardware-Signalquelle

Der Prozessdatenaustausch zwischen b maXX controller PLC und den Systemkomponenten (die links der b maXX controller PLC gesteckt sind) kann über ein CBPB-Hardware-Signal synchronisiert werden.

Auf der b maXX controller PLC wird eine EVENT-Task auf das entsprechende Ereignis (CBPB-Hardware-Signal) angelegt und in dieser Task findet dann die Prozessdatenkommunikation statt.

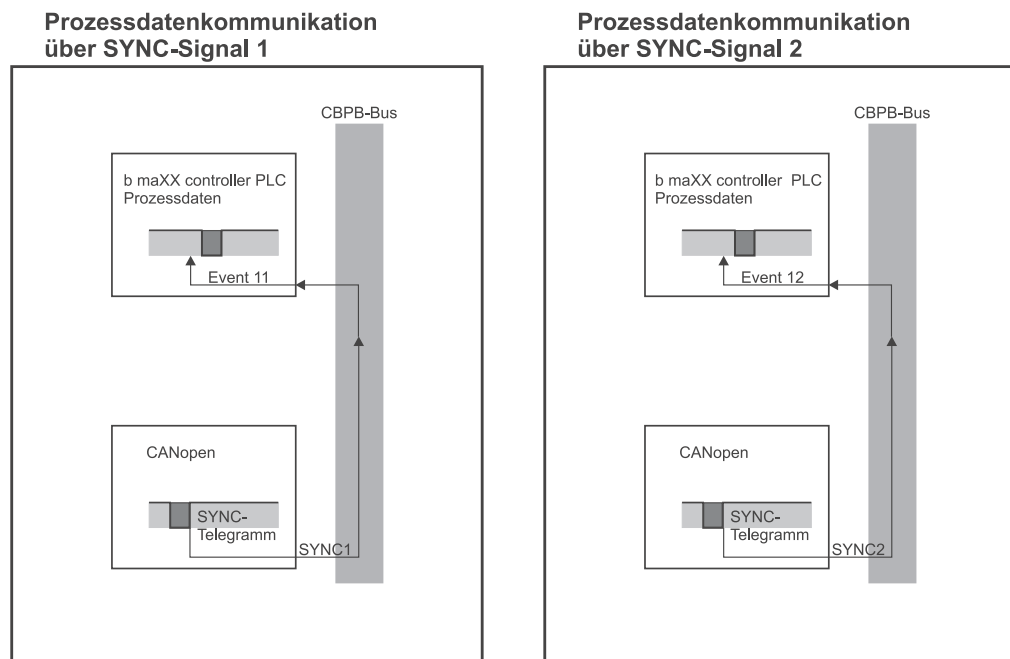


Abbildung 48: Prozessdatenkommunikation über SYNC1- bzw. SYNC2-Signal

### 5.6.3 Für die Applikation nutzbare Interruptquellen der b maXX controller PLC

Für die b maXX controller PLC können Sie Hardware-Signale vom CBPB-Bus und PLC-interne Quellen wie CPU-Timer 1 und 2 oder den Board Timer A als Interruptquellen verwenden, die über die Ereignis-Auswahl der BYPASS EVENT-Tasks mit spezifischen Programmteilen der Applikation verbunden werden können.



#### HINWEIS ZU EVENT-BYPASS-EVENT-TASKS

Wird in PROPROG wt eine Task vom Typ EVENT eingefügt, muss auch das Attribut "BYPASS" gesetzt werden.

Da EVENT-BYPASS-Tasks direkt durch das Ereignis aufgerufen und nicht vom Laufzeitsystem überwacht werden, sind die Einstellungen "Intervall" und "Watchdogzeit" in der Task-Verwaltung ohne Bedeutung und werden ignoriert.

Weiterhin dürfen in den Programmteilen, die mit einer EVENT-BYPASS-Task verknüpft ist, prinzipiell keine Breakpoints gesetzt und auch kein Adressstatus aktiviert werden.

Eine Interruptquelle muss während der Kalt-/Warmstart-Phase der b maXX controller PLC über den Funktionsbaustein INTR\_SET für die Ereignisse 0, 2, 5, 6, 11, 12 (siehe Standard-Bibliothek SYSTEM2\_C\_PLC01\_20bd00 (PROPROG wt II) bzw. SYSTEM2\_C\_PLC01\_30bd00 / SYSTEM2\_C\_PLC02\_30bd00 (ProProg wt III) oder höher) eingerichtet werden. Erst dann kann im SPS-Zustand "RUN" bei Auftreten des Ereignisses der in der EVENT-BYPASS-Task zugeordnete Programmteil aufgerufen werden.

Durch die Zuordnung von Ereignis zu Programmteil wird dieser Teil dann immer synchron und mit der gleichen Periodizität wie die Ereignis-Quelle abgearbeitet.

Darstellung aller EVENTS auf der b maXX controller PLC, die einen Interrupt auslösen können:



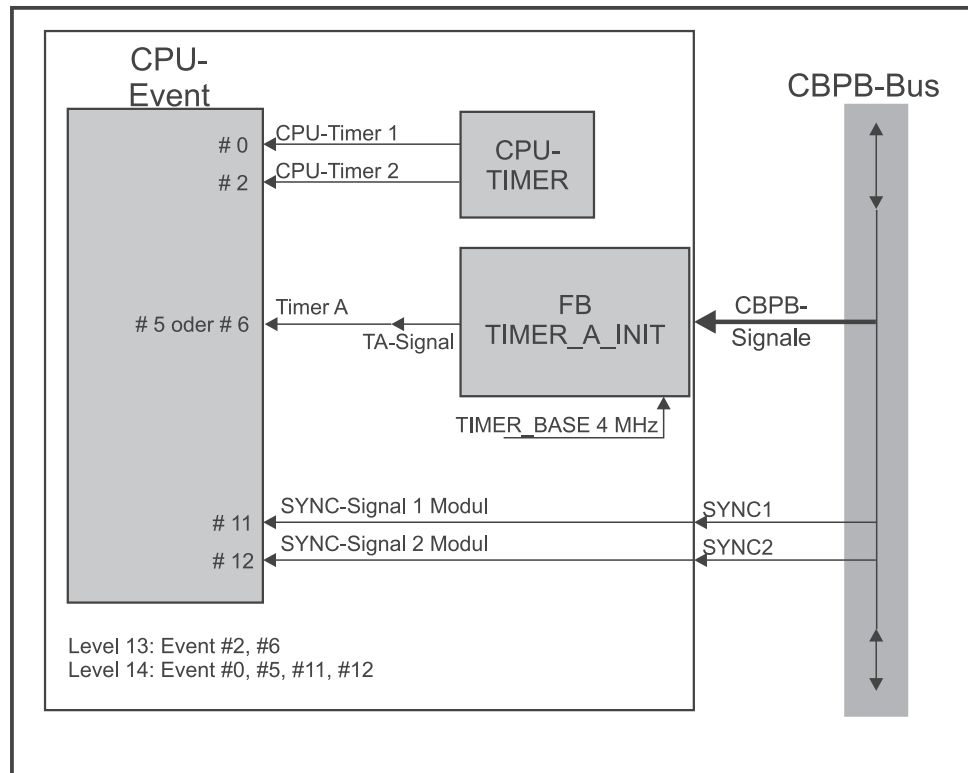


Abbildung 49: Darstellung aller EVENTS

Detail Konfigurationsmöglichkeiten für das EVENT Timer A durch Auswahl des CBPB-Signals über die INPUT-Funktion von Baustein "TIMER\_A\_INIT" und Aufschaltung auf das interne TA-Signal:

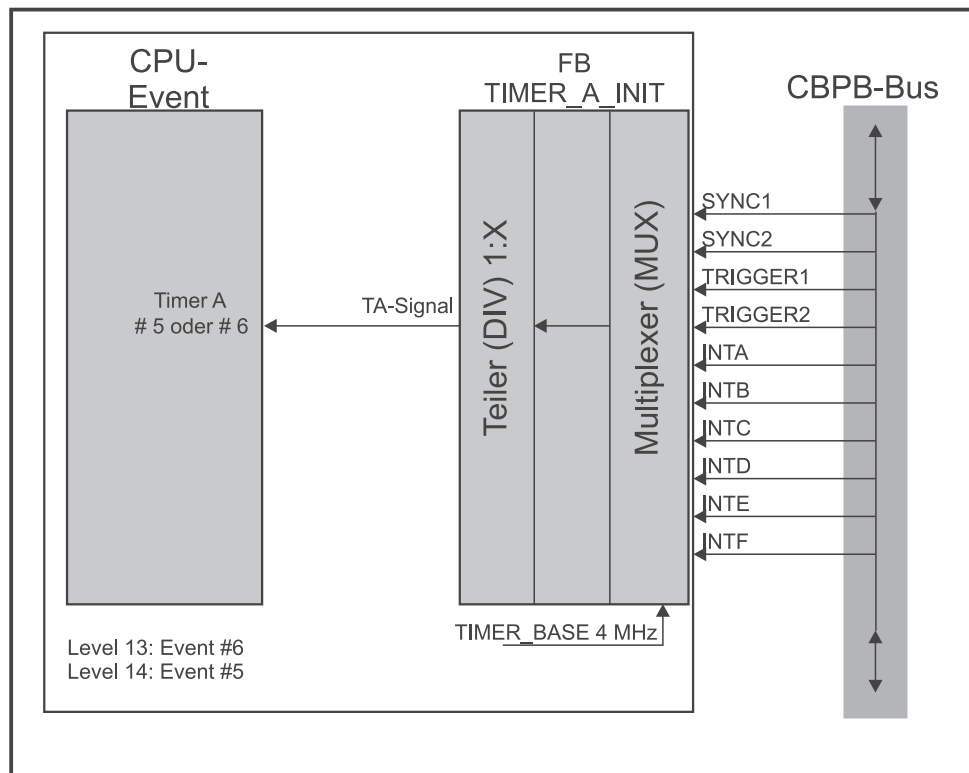


Abbildung 50: Detail Konfigurationsmöglichkeiten über die INPUT-Funktion von TIMER\_A\_INIT

### 5.6.4 CBPB-Signalgenerierung über den Funktionsbaustein TIMER\_A\_INIT

Die b maXX controller PLC kann auch für eine CBPB-Signalausgabe (b maXX controller PLC → CBPB) konfiguriert werden, wo sie dann z. B. als Triggerquelle für andere Systemkomponenten (die links der b maXX controller PLC gesteckt sind) fungieren kann.

Die Signalquelle wird dabei entweder auf der PLC selbst erzeugt oder stammt von einem anderen CBPB-Signal, das über die PLC umgeleitet und gegebenenfalls heruntergeteilt wird (siehe Beschreibung zu Funktionsbaustein "TIMER\_A\_INIT" der Bibliothek SYSTEM1\_C\_PLC01\_20bd00 (PROPROG wt II) bzw. SYSTEM1\_C\_PLC01\_30bd00 / SYSTEM1\_C\_PLC02\_30bd00 (ProProg wt III) oder höher).

Konfigurationsmöglichkeiten zur CBPB-Signalgenerierung durch Aufschalten des intern generierten TA-Signals auf die CBPB-Signalleitung über die OUTPUT-Funktion von Baustein "TIMER\_A\_INIT":

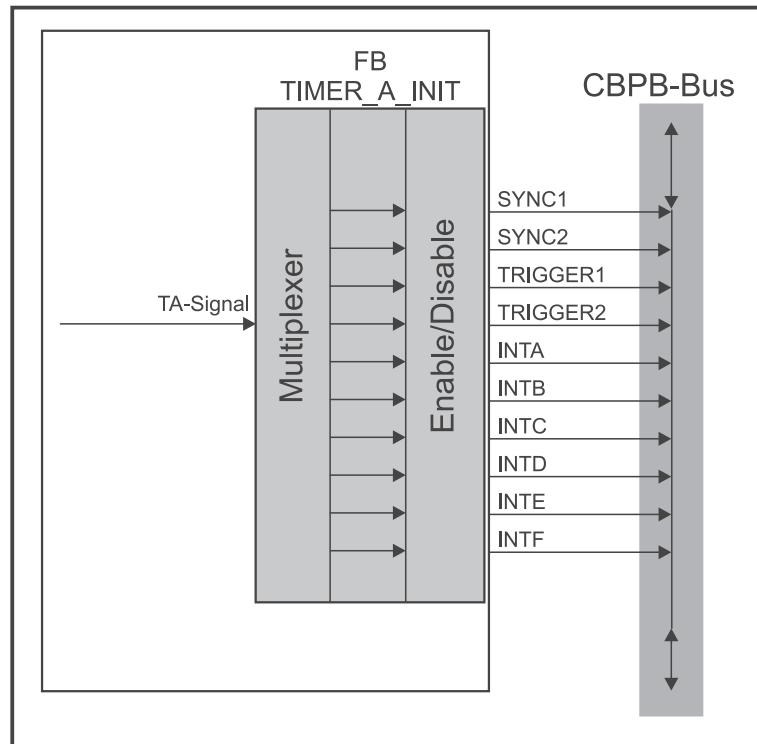


Abbildung 51: Konfigurationsmöglichkeiten zur CBPB-Signalgenerierung über die OUTPUT-Funktion von TIMER\_A\_INIT

Durch Mehrfachaufruf der OUTPUT-Funktion kann das interne TA-Signal auch gleichzeitig auf verschiedene CPBP-Signale durchgeschaltet werden.

Eine Umleitung von CPBP-Signalen über die PLC (bei Bedarf mit einer Signal-Teilung) kann man erreichen, indem man die INPUT-Funktion und die OUTPUT-Funktion von "TIMER\_A\_INIT" entsprechend miteinander kombiniert (siehe "TIMER\_A\_INIT"-Beschreibung).

## 5.6.5 Funktionsbaustein "TIMER\_A\_INIT"

### 5.6.5.1 Funktionsbaustein "TIMER\_A\_INIT" für BMC-M-PLC-01

Beschreibung des Funktionsbausteins "TIMER\_A\_INIT" der Bibliothek „SYSTEM1\_C\_PLC01\_20bd00“ (PROPROP wt II) bzw. „SYSTEM1\_C\_PLC01\_30bd00“ (ProProg wt III) oder höher.

Dieser Funktionsbaustein für CPBP besitzt mehrere Funktionen:

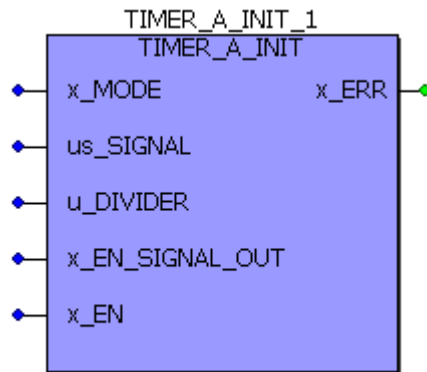


Abbildung 52: Funktionsbaustein TIMER\_A\_INIT

Funktion	Beschreibung
INPUT	Sie können ein CPBP-Signal auswählen, teilen und auf die interne TA-Leitung durchschalten sowie bei Bedarf damit einen TIMER_A_Interrupt auslösen.
TIMER	INPUT-Funktion: Durch Auswahl der Source-Quelle "4MHz" (= Zeitbasis Timer_A) und Angabe eines Teilerwertes kann nahezu jedes beliebige Periodensignal erzeugt werden. Bei Bedarf kann damit ein TIMER_A_Interrupt ausgelöst werden.
OUTPUT	Die interne TA-Leitung wird auf einen auswählbaren CPBP-Ausgang geschaltet.
TRIGGER	Kombiniert man die TIMER-Funktion mit der OUTPUT-Funktion, kann das erzeugte Periodensignal auf einen oder mehrere CPBP-Ausgänge geschaltet werden.
OUTPUT-Signal freigeben / sperren	CPBP-Ausgänge können selektiv gesperrt und wieder freigegeben werden.



### HINWEIS

zum Baustein TIMER\_A\_INIT:

Einsatz: Kann sowohl in der Initialisierungstask (Kalt/Warmstart-Task) als auch in der DE-Fault-Task oder in einer beliebigen Cyclic Task eingebunden werden.

Der Baustein kann dabei auch mehrere Male an verschiedenen Stellen platziert oder auch öfters hintereinander aufgerufen werden, um die entsprechende Einzelfunktion zu installieren.

Wird der Baustein im zyklischen Teil verwendet, muss nach Aktivierung der Funktion der Eingang x\_EN wieder auf FALSE zurückgestellt werden, damit der TIMER ablaufen bzw. der Teiler teilen kann!

Parameter Eingang	Datentyp	Beschreibung
x_MODE	BOOL	Auswahl der Grund-Funktion (Signal-Richtung)
us_SIGNAL	USINT	Signal-Quelle / Ziel
u_DIVIDER	UINT	Teiler-Wert
x_EN_SIGNAL_OUT	BOOL	Freigeben/Sperren des Ausgangssignals bei gewählter OUTPUT-Funktion
x_EN	BOOL	Freigabe des Bausteins

Parameter Ausgang	Datentyp	Beschreibung
x_ERR	BOOL	Fehlerbit

#### Eingang **x\_MODE**:

Über x\_MODE kann die grundlegende Funktion ausgewählt werden:

x\_MODE = FALSE (us\_SIGNAL = Signal-Quelle): Funktionen INPUT, TIMER. => Es wird ein internes TA-Signal generiert.

x\_MODE = TRUE (us\_SIGNAL = Signal-Ziel): OUTPUT, TRIGGER, OUTPUT-Signal freigeben/sperren. Wurde der CPBP-Ausgang freigegeben, wird das interne TA-Signal auf das angegebene CPBP-Signal-Ziel durchgeschaltet.

#### Eingang **us\_SIGNAL**:

Für x_MODE := FALSE: us_SIGNAL :=	Auswahl der Signal-Quelle:
USINT#0	4MHz
USINT#1 bis USINT#4	Nicht erlaubt
USINT#5	INTA
USINT#6	INTB
USINT#7	INTC
USINT#8	INTD
USINT#9	INTE
USINT#10	INTF
USINT#11	TRIGGER1
USINT#12	TRIGGER2
USINT#13	SYNC1
USINT#14	SYNC2
> 14	Nicht erlaubt

Für <b>x_MODE := TRUE:</b> <b>us_SIGNAL :=</b>	Auswahl des CBPB-Signal-Ziels:
USINT#0 - USINT#4	Nicht erlaubt
USINT#5	INTA
USINT#6	INTB
USINT#7	INTC
USINT#8	INTD
USINT#9	INTE
USINT#10	INTF
USINT#11	TRIGGER1
USINT#12	TRIGGER2
USINT#13	SYNC1
USINT#14	SYNC2
> 14	Nicht erlaubt

### Eingang **u\_DIVIDER**:

Wird nur bei der INPUT- und TIMER-Funktion ausgewertet ( $x\_MODE = FALSE$ ).

$u\_DIVIDER$  gibt an, in welchem Verhältnis das Eingangssignal geteilt wird. Dabei gilt folgende Teilungsformel:

$$\text{Internes TA-Signal} := \text{us\_SIGNAL} / (u\_DIVIDER + 1)$$

Bei  $u\_DIVIDER = 0$  wird somit das externe CBPB-Signal direkt 1:1 auf das interne TA-Signal durchgeschaltet, es erfolgt keine Teilung des Signals.

### Eingang **x\_EN\_SIGNAL\_OUT**:

Wird nur bei der OUTPUT- und TRIGGER-Funktion ausgewertet ( $x\_MODE = TRUE$ ).

Das interne TA-Signal wird auf den ausgewählten CBPB-Ausgang geschaltet ( $x\_EN\_SIGNAL\_OUT := TRUE$ ) bzw. wieder gesperrt ( $x\_EN\_SIGNAL\_OUT := FALSE$ ).

Durch mehrere Baustein-Aufrufe hintereinander und unterschiedlichen CBPB-Zielen kann das interne TA-Signal auf mehrere CBPB-Ausgänge ausgegeben werden.

### Eingang **x\_EN**:

Mit  $x\_EN = TRUE$  wird die jeweilige Funktion installiert. Bei  $x\_EN := FALSE$  erfolgt keine Bearbeitung des Bausteins. Wenn der Baustein zyklisch ausgeführt wird, sollte nach Aktivierung einer Einzel-Funktion der Eingang  $x\_EN$  wieder auf FALSE zurückgestellt werden, damit der TIMER ablaufen bzw. das CBPB-Eingangssignal geteilt werden kann.

### Ausgang **x\_ERR := TRUE**:

Auswahl des CBPB-Ziels wird nicht unterstützt!

**Erläuterung der Funktionen im Detail:****INPUT-Sonderfunktion: TIMER****(x\_MODE := FALSE und us\_SIGNAL := USINT#0):**

Als Timer-Basis sind 4 MHz fest vorgegeben. Formel für die Berechnung der Zeitperiode:

$$\text{Zeitperiode} = (u\_DIVIDER + 1) \times 0,25 \mu\text{s}.$$

Minimalwert für "u\_DIVIDER": UINT#999 =&gt; Zeit = 250 µs

Maximalwert für "u\_DIVIDER": UINT#65535 =&gt; Zeit = 16384 µs

Formel für die Berechnung von u\_DIVIDER bei gegebener Zeitperiode:

$$u\_DIVIDER = (\text{Zeitperiode} / 0,25 \mu\text{s}) - 1$$

Wenn der Timer über das intern erzeugte TA-Signal einen Interrupt auslösen soll, muss nach Aufruf der TIMER-Funktion über "TIMER\_A\_INIT" der System-Funktionsbaustein "INTR\_SET" (siehe FW-LIB SYSTEM2\_C\_PLC01\_20bd00 (PROPROG wt II) bzw. „SYSTEM2\_C\_PLC01\_30bd00“ (ProProg wt III) oder höher) ausgeführt werden mit den Parametern: i\_EVENT := INT#5 (bzw. i\_EVENT := INT#6) für TIMER A. (Die anderen Parameter i\_MODE, i\_PAR1 und i\_PAR2 werden ignoriert).

Dabei ist darauf zu achten, dass die über u\_DIVIDER festgelegte Zeitperiode nicht zu klein bzw. zu schnell gewählt wird, sonst kann der nächste zyklische Interrupt bereits anstehen, noch bevor der gerade Laufende beendet ist. Da die Interrupts als Bypass-Interrupts mit hoher Priorität eingerichtet sind, läuft nur noch dieser Interrupt, und die SPS geht offline (-> Die übrige Anwendung "reagiert nicht mehr").

Um dies zu vermeiden, wird ein Interruptintervall von mindestens 250 µs empfohlen. Dies entspricht einem "u\_DIVIDER"-Wert von >= UINT#999.

**Output-Sonderfunktion: TRIGGER**

Wenn man die TIMER-Funktion mit der OUTPUT-Funktion kombiniert, kann das erzeugte Periodensignal als Triggersignal auf einen oder mehrere CPBP-Ausgänge geschaltet werden. Dazu muss der Baustein „TIMER\_A\_INIT“ zweimal hintereinander in einer Kalt- bzw. Warmstart-POE aufgerufen werden:

**1. Aufruf: TIMER-Funktion:**

Als Timer-Basis sind 4 MHz fest vorgegeben. Wenn ein Teilerwert am Eingang "u\_DIVIDER" kleiner UINT#3 angeschlossen wird, wird ein dauerhaftes LOW-Signal am internen Timerausgang TA erzeugt. Wenn das interne TA-Signal über die OUTPUT-Funktion anschließend auf den CPBP ausgegeben werden soll (z. B. PLC-Funktion als CPBP-Trigger-Quelle), sollte zuvor bei der INPUT-Funktion mindestens ein Wert größer UINT#7 eingegeben werden, um das CPBP-BUS-Timing nicht zu verletzen.

Formel für die Berechnung der Zeitperiode:

$$\text{Zeitperiode} = (u\_DIVIDER + 1) \times 0,25 \mu\text{s}.$$

Minimalwert für "u\_DIVIDER": UINT#8 =&gt; Zeit = 2,25 µs

Maximalwert für "u\_DIVIDER": UINT#65535 =&gt; Zeit = 16384 µs

Formel für die Berechnung von u\_DIVIDER:

$$u\_DIVIDER = (\text{Zeitperiode} / 0,25 \mu\text{s}) - 1$$

Soll der Triggerimpuls auch einen Interrupt auslösen, wird ein Interruptintervall von mindestens 250  $\mu\text{s}$  empfohlen. Dies entspricht einen "u\_DIVIDER"-Wert von  $\geq$  UINT#999.

2. Aufruf: OUTPUT-Funktion (CPBP-Ziel = USINT#11 (Trigger 1)): siehe folgende Erläuterung:

### OUTPUT-Sonderfunktion: CPBP-Signal freigeben/sperrern

Falls das intern generierte TA-Signal an mehrere CPBP-Ausgängen gleichzeitig ausgegeben werden soll, kann der Baustein mehrere Male hintereinander mit dem jeweiligen CPBP-Ziel und "x\_MODE := TRUE" sowie "x\_EN\_CPBP\_OUT := TRUE" für die Treiberfreigabe aufgerufen werden. Mit "x\_EN\_SIGNAL\_OUT := FALSE" kann die Freigabe selektiv wieder gesperrt werden.

Internes TA-Signal soll einen Interrupt generieren:

Nach Parametrierung von "TIMER\_A\_INIT" muss der System-Funktionsbaustein "INTR\_SET" (siehe FW-LIB SYSTEM2\_C\_PLC01\_20bd00 (PROPROG wt II) bzw. „SYSTEM2\_C\_PLC01\_30bd00“ (ProProg wt III) oder höher) ausgeführt werden mit den Parametern: i\_EVENT := INT#5 (bzw. i\_EVENT := INT#6) für TIMER A. (Die anderen Parameter i\_MODE, i\_PAR1 und i\_PAR2 werden ignoriert).

Anwendungsbeispiel:

Das Eingangs-CPBP-Signal TRIGGER1 soll durch den Wert 10 geteilt und dann wieder auf die Ausgangs-CPBP-Leitung TRIGGER2 geschaltet werden; außerdem soll das geteilte Signal einen Interrupt auslösen.

1. Schritt: TRIGGER1 -> 1:10 -> internes TA-Signal:  
FB "TIMER\_A\_INIT" ausführen mit: x\_MODE: = FALSE; us\_SIGNAL := USINT#11;  
(Für Auswahl TRIGGER1) u\_DIVIDER := UINT#9; (Für Teilverhältnis 1:10)
2. Schritt: Internes TA-Signal -> TRIGGER2:  
FB "TIMER\_A\_INIT" ausführen mit: x\_MODE: = TRUE; us\_SIGNAL := USINT#12;  
(Für Auswahl TRIGGER2); x\_EN\_SIGNAL\_OUT := TRUE;
3. Schritt: FB "INTR\_SET" ausführen mit i\_EVENT := INT#5.  
=> Durch Anlegen einer EVENT-Bypass-Task mit dem Ereignis „5“ kann ein SPS-Programm synchron zum geteilten Signal ausgeführt werden.



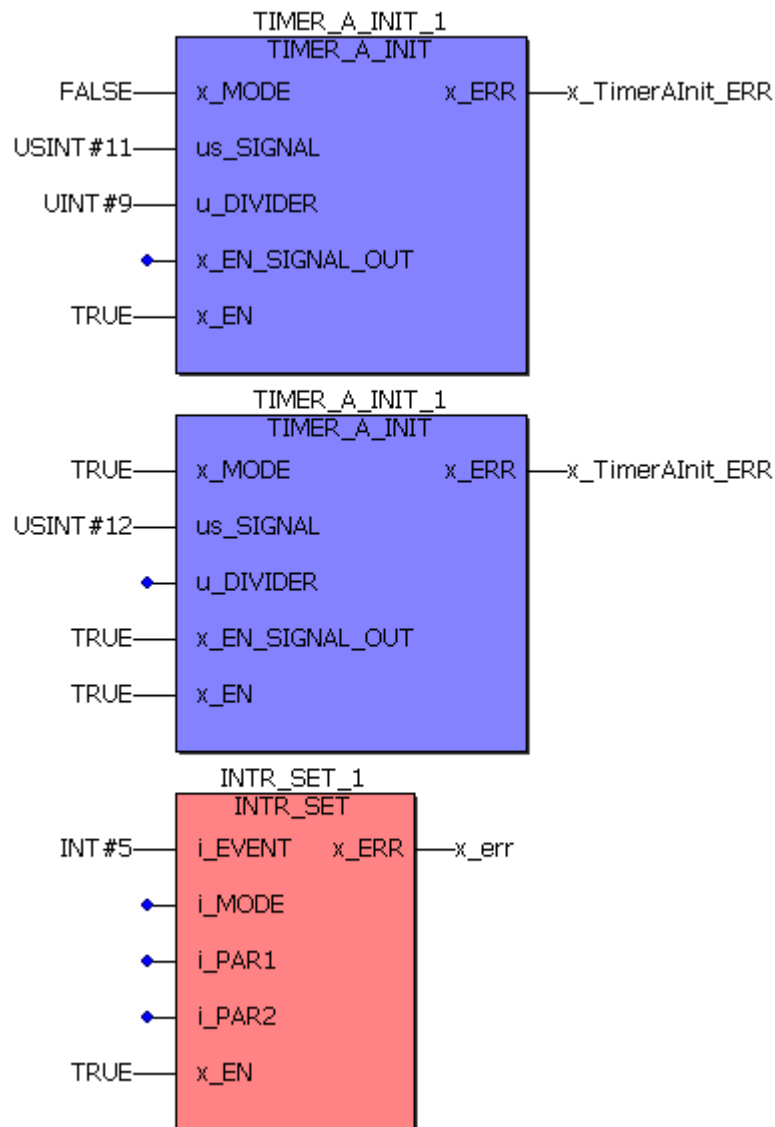


Abbildung 53: Das Signal TRIGGER1 löst jedes 10. mal das Signal TRIGGER2 aus und das Signal TRIGGER1 löst die Bypass Event-Task 5 aus

### 5.6.5.2 Funktionsbaustein "TIMER\_A\_INIT" für BMC-M-PLC-02

Beschreibung des Funktionsbausteins "TIMER\_A\_INIT" der Bibliothek „SYSTEM1\_C\_PLC02\_30bd02“ (ProProg wt III) oder höher.

Dieser Funktionsbaustein für CPBP besitzt mehrere Funktionen:

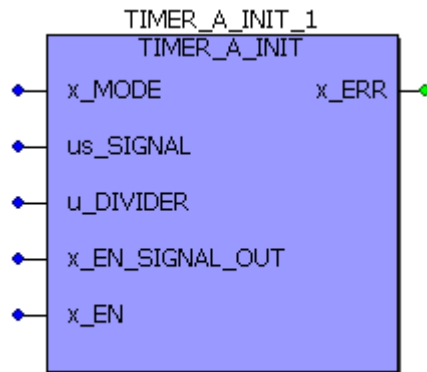


Abbildung 54: Funktionsbaustein TIMER\_A\_INIT

Funktion	Beschreibung
INPUT	Sie können ein CPBP-Signal auswählen, teilen und auf die interne TA-Leitung durchschalten sowie bei Bedarf damit einen TIMER_A_Interrupt auslösen.
TIMER	INPUT-Funktion: Durch Auswahl der Source-Quelle "4MHz" (= Zeitbasis Timer_A) und Angabe eines Teilerwertes kann nahezu jedes beliebige Periodensignal erzeugt werden. Bei Bedarf kann damit ein TIMER_A_Interrupt ausgelöst werden.
OUTPUT	Die interne TA-Leitung wird auf einen auswählbaren CPBP-Ausgang geschaltet.
TRIGGER	Kombiniert man die TIMER-Funktion mit der OUTPUT-Funktion, kann das erzeugte Periodensignal auf einen oder mehrere CPBP-Ausgänge geschaltet werden.
OUTPUT-Signal freigeben / sperren	CPBP-Ausgänge können selektiv gesperrt und wieder freigegeben werden.



### HINWEIS

zum Baustein TIMER\_A\_INIT:

Einsatz: Kann sowohl in der Initialisierungstask (Kalt/Warmstart-Task) als auch in der DE-Fault-Task oder in einer beliebigen Cyclic Task eingebunden werden.

Der Baustein kann dabei auch mehrere Male an verschiedenen Stellen platziert oder auch öfters hintereinander aufgerufen werden, um die entsprechende Einzelfunktion zu installieren.

Wird der Baustein im zyklischen Teil verwendet, muss nach Aktivierung der Funktion der Eingang x\_EN wieder auf FALSE zurückgestellt werden, damit der TIMER ablaufen bzw. der Teiler teilen kann!

Parameter Eingang	Datentyp	Beschreibung
x_MODE	BOOL	Auswahl der Grund-Funktion (Signal-Richtung)
us_SIGNAL	USINT	Signal-Quelle / Ziel
u_DIVIDER	UINT	Teiler-Wert
x_EN_SIGNAL_OUT	BOOL	Freigeben/Sperren des Ausgangssignals bei gewählter OUTPUT-Funktion
x_EN	BOOL	Freigabe des Bausteins

Parameter Ausgang	Datentyp	Beschreibung
x_ERR	BOOL	Fehlerbit

#### Eingang **x\_MODE**:

Über x\_MODE kann die grundlegende Funktion ausgewählt werden:

x\_MODE = FALSE (us\_SIGNAL = Signal-Quelle): Funktionen INPUT, TIMER. => Es wird ein internes TA-Signal generiert.

x\_MODE = TRUE (us\_SIGNAL = Signal-Ziel): OUTPUT, TRIGGER, OUTPUT-Signal freigeben/sperren. Wurde der CPBP-Ausgang freigegeben, wird das interne TA-Signal auf das angegebene CPBP-Signal-Ziel durchgeschaltet.

#### Eingang **us\_SIGNAL**:

Für x_MODE := FALSE: us_SIGNAL :=	Auswahl der Signal-Quelle:
USINT#0	1MHz
USINT#1 bis USINT#4	Nicht erlaubt
USINT#5	INTA
USINT#6	INTB
USINT#7	INTC
USINT#8	INTD
USINT#9	INTE
USINT#10	INTF
USINT#11	TRIGGER1
USINT#12	TRIGGER2
USINT#13	SYNC1
USINT#14	SYNC2
> 14	Nicht erlaubt

Für <b>x_MODE := TRUE:</b> <b>us_SIGNAL :=</b>	Auswahl des CBPB-Signal-Ziels:
USINT#0 - USINT#4	Nicht erlaubt
USINT#5	INTA
USINT#6	INTB
USINT#7	INTC
USINT#8	INTD
USINT#9	INTE
USINT#10	INTF
USINT#11	TRIGGER1
USINT#12	TRIGGER2
USINT#13	SYNC1
USINT#14	SYNC2
> 14	Nicht erlaubt

### Eingang **u\_DIVIDER**:

Wird nur bei der INPUT- und TIMER-Funktion ausgewertet ( $x\_MODE = FALSE$ ).

$u\_DIVIDER$  gibt an, in welchem Verhältnis das Eingangssignal geteilt wird. Dabei gilt folgende Teilungsformel:

$$\text{Internes TA-Signal} := \text{us\_SIGNAL} / (u\_DIVIDER + 1)$$

Bei  $u\_DIVIDER = 0$  wird somit das externe CBPB-Signal direkt 1:1 auf das interne TA-Signal durchgeschaltet, es erfolgt keine Teilung des Signals.

### Eingang **x\_EN\_SIGNAL\_OUT**:

Wird nur bei der OUTPUT- und TRIGGER-Funktion ausgewertet ( $x\_MODE = TRUE$ ).

Das interne TA-Signal wird auf den ausgewählten CBPB-Ausgang geschaltet ( $x\_EN\_SIGNAL\_OUT := TRUE$ ) bzw. wieder gesperrt ( $x\_EN\_SIGNAL\_OUT := FALSE$ ).

Durch mehrere Baustein-Aufrufe hintereinander und unterschiedlichen CBPB-Zielen kann das interne TA-Signal auf mehrere CBPB-Ausgänge ausgegeben werden.

### Eingang **x\_EN**:

Mit  $x\_EN = TRUE$  wird die jeweilige Funktion installiert. Bei  $x\_EN := FALSE$  erfolgt keine Bearbeitung des Bausteins. Wenn der Baustein zyklisch ausgeführt wird, sollte nach Aktivierung einer Einzel-Funktion der Eingang  $x\_EN$  wieder auf FALSE zurückgestellt werden, damit der TIMER ablaufen bzw. das CBPB-Eingangssignal geteilt werden kann.

### Ausgang **x\_ERR := TRUE**:

Auswahl des CBPB-Ziels wird nicht unterstützt!

**Erläuterung der Funktionen im Detail:****INPUT-Sonderfunktion: TIMER****(x\_MODE := FALSE und us\_SIGNAL := USINT#0):**

Als Timer-Basis ist 1 MHz fest vorgegeben. Formel für die Berechnung der Zeitperiode:

$$\text{Zeitperiode} = (u\_DIVIDER + 1) \times 1 \mu\text{s}.$$

Minimalwert für "u\_DIVIDER": UINT#249 => Zeit = 250  $\mu\text{s}$ Maximalwert für "u\_DIVIDER": UINT#65535 => Zeit = 65536  $\mu\text{s}$ 

Formel für die Berechnung von u\_DIVIDER bei gegebener Zeitperiode:

$$u\_DIVIDER = (\text{Zeitperiode} / 1 \mu\text{s}) - 1$$

Wenn der Timer über das intern erzeugte TA-Signal einen Interrupt auslösen soll, muss nach Aufruf der TIMER-Funktion über "TIMER\_A\_INIT" der System-Funktionsbaustein "INTR\_SET" (siehe FW-LIB SYSTEM2\_C\_PLC02\_30bd02 oder höher) ausgeführt werden mit den Parametern: i\_EVENT := INT#5 (bzw. i\_EVENT := INT#6) für TIMER A. (Die anderen Parameter i\_MODE, i\_PAR1 und i\_PAR2 werden ignoriert).

Dabei ist darauf zu achten, dass die über u\_DIVIDER festgelegte Zeitperiode nicht zu klein bzw. zu schnell gewählt wird, sonst kann der nächste zyklische Interrupt bereits anstehen, noch bevor der gerade Laufende beendet ist. Da die Interrupts als Bypass-Interrupts mit hoher Priorität eingerichtet sind, läuft nur noch dieser Interrupt, und die SPS geht offline (-> Die übrige Anwendung "reagiert nicht mehr").

Um dies zu vermeiden, wird ein Interruptintervall von mindestens 250  $\mu\text{s}$  empfohlen. Dies entspricht einem "u\_DIVIDER"-Wert von  $\geq$  UINT#249.

**Output-Sonderfunktion: TRIGGER**

Wenn man die TIMER-Funktion mit der OUTPUT-Funktion kombiniert, kann das erzeugte Periodensignal als Triggersignal auf einen oder mehrere CPBP-Ausgänge geschaltet werden. Dazu muss der Baustein „TIMER\_A\_INIT“ zweimal hintereinander in einer Kalt- bzw. Warmstart-POE aufgerufen werden:

**1. Aufruf: TIMER-Funktion:**

Als Timer-Basis ist 1 MHz fest vorgegeben. Wenn ein Teilerwert am Eingang "u\_DIVIDER" kleiner UINT#1 angeschlossen wird, wird ein dauerhaftes LOW-Signal am internen Timerausgang TA erzeugt. Wenn das interne TA-Signal über die OUTPUT-Funktion anschließend auf den CPBP ausgegeben werden soll (z. B. PLC-Funktion als CPBP-Trigger-Quelle), sollte zuvor bei der INPUT-Funktion mindestens ein Wert größer UINT#1 eingegeben werden, um das CPBP-BUS-Timing nicht zu verletzen.

Formel für die Berechnung der Zeitperiode:

$$\text{Zeitperiode} = (u\_DIVIDER + 1) \times 1 \mu\text{s}.$$

Minimalwert für "u\_DIVIDER": UINT#2 => Zeit = 3  $\mu\text{s}$ Maximalwert für "u\_DIVIDER": UINT#65535 => Zeit = 65536  $\mu\text{s}$ 

Formel für die Berechnung von u\_DIVIDER:

$$u\_DIVIDER = (\text{Zeitperiode} / 1 \mu\text{s}) - 1$$

Soll der Triggerimpuls auch einen Interrupt auslösen, wird ein Interruptintervall von mindestens 250 µs empfohlen. Dies entspricht einen "u\_DIVIDER"-Wert von  $\geq$  UINT#249.

2. Aufruf: OUTPUT-Funktion (CPBP-Ziel = USINT#11 (Trigger 1)): siehe folgende Erläuterung:

### OUTPUT-Sonderfunktion: CPBP-Signal freigeben/sperrern

Falls das intern generierte TA-Signal an mehrere CPBP-Ausgängen gleichzeitig ausgegeben werden soll, kann der Baustein mehrere Male hintereinander mit dem jeweiligen CPBP-Ziel und "x\_MODE := TRUE" sowie "x\_EN\_CPBP\_OUT := TRUE" für die Treiberfreigabe aufgerufen werden. Mit "x\_EN\_SIGNAL\_OUT := FALSE" kann die Freigabe selektiv wieder gesperrt werden.

Internes TA-Signal soll einen Interrupt generieren:

Nach Parametrierung von "TIMER\_A\_INIT" muss der System-Funktionsbaustein "INTR\_SET" (siehe FW-LIB SYSTEM2\_C\_PLC02\_30bd02 oder höher) ausgeführt werden mit den Parametern: i\_EVENT := INT#5 (bzw. i\_EVENT := INT#6) für TIMER A. (Die anderen Parameter i\_MODE, i\_PAR1 und i\_PAR2 werden ignoriert).

Siehe [►Anwendungsbeispiel:◄](#) auf Seite 88.

### 5.6.6 Konfigurationsbeispiele

---

Je nach verwendeten Systemkomponenten und Synchronisierungsanforderungen werden verschiedene Konfigurationen für die Prozessdatenkommunikation zwischen der b maXX controller PLC und den Systemkomponenten, die Verwaltung von Interruptquellen sowie die Bereitstellung und Verarbeitung von Trigger- und SYNC-Signalen benötigt.

Für die Mehrzahl der Anwendungsfälle sind die Beispiele A bis C verwendbar (siehe die folgenden Kapitel [►Beispiel A: Einrichten eines CPU-Timer 1-Interrupts◄](#) ab Seite 94, [►Beispiel B: Einrichten eines Board-Timer A Interrupts und Triggern des SYNC-Signal 1◄](#) ab Seite 95 und [►Beispiel C: Realisierung einer einfachen seriellen RS485-Kommunikation mit Hilfe eines CPU-Timers◄](#) ab Seite 99).

### 5.6.7 Beispiel A: Einrichten eines CPU-Timer 1-Interrupts

---

Für die b maXX controller PLC soll ein CPU-Timer 1 Interrupt (Level 14) mit der Periode 20 ms erzeugt werden, zu dem auf der b maXX controller PLC ein SPS-Programmteil ausgeführt wird.

FB INTR\_SET ausführen mit: i\_EVENT := INT#0; i\_MODE := INT#0; i\_PAR1 := INT#400; x\_EN := TRUE.

Der Bausteinaufruf erfolgt einmalig während der Initialisierungsphase (Kaltstart-/Warmstart-Task).

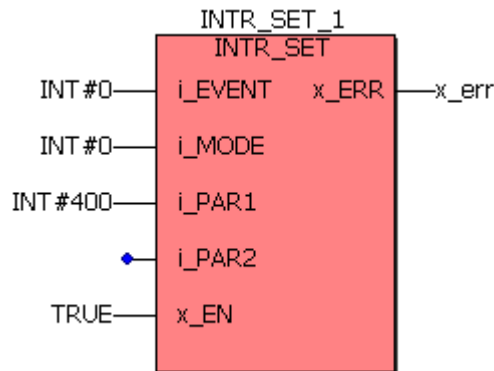


Abbildung 55: Der CPU-Timer 1 löst alle 20 ms die Bypass Event-Task 0 aus

Der Anwender-Programmcode wird in einer Event-Task, die dem Ereignis "0 CPU-Timer 1 Level 14" zugeordnet ist, abgearbeitet.

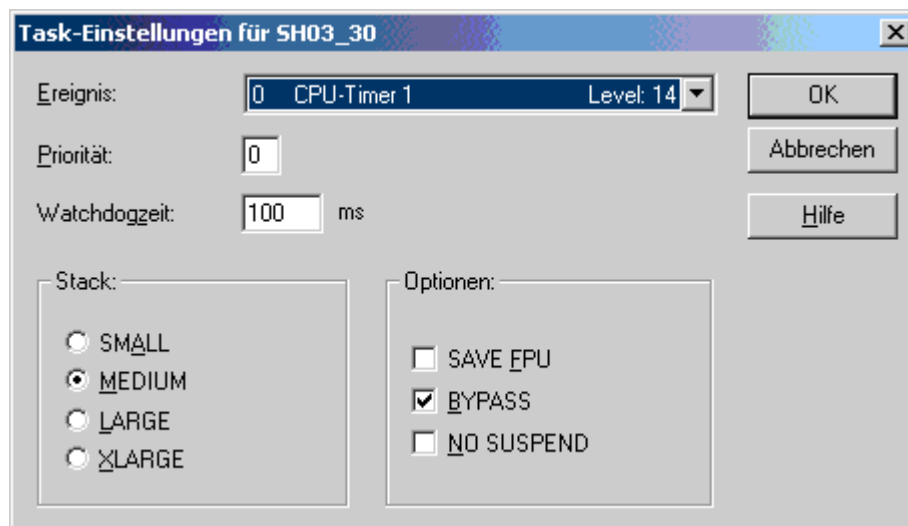


Abbildung 56: Einstellungen für die Bypass Event-Task auf das Ereignis 0 CPU-Timer 1 Level 14

## 5.6.8 Beispiel B: Einrichten eines Board-Timer A Interrupts und Triggern des SYNC-Signal 1

Bei Einrichten des Board-Timer A Interrupts wird zwischen BMC-M-PLC-01 und BMC-M-PLC-02 unterschieden.

### 5.6.8.1 Board-Timer A Interrupt auf BMC-M-PLC-01

Die b maXX controller PLC soll auf der SYNC-Signal 1 Leitung ein Triggersignal mit 2 ms Periode erzeugen. Synchron zum generierten Triggersignal soll ein SPS-Programmteil ausgeführt werden (mit dem Interrupt-Level 14).

1. Schritt: Internes TA-Signal erzeugen über die Timer-Funktion mit der Zeitperiode 2 ms:  
 FB "TIMER\_A\_INIT" ausführen mit: x\_MODE := FALSE; us\_SIGNAL := USINT#0; (für  
 Timer-Funktion); u\_DIVIDER = (2000 µs / 0,25 µs) -1 = UINT#7999; x\_EN := TRUE;
2. Schritt: TA-Signal als Triggersignal an SYNC1 ausgeben:  
 FB "TIMER\_A\_INIT" ausführen mit: x\_MODE := TRUE; us\_SIGNAL := USINT#13;  
 (Für Ziel-Auswahl SYNC1); x\_EN\_SIGNAL\_OUT := TRUE; x\_EN := TRUE;
3. Schritt: FB "INTR\_SET" ausführen mit i\_EVENT := INT#5.  
 => Durch Anlegen einer EVENT-Bypass-Task mit dem Ereignis „5“ kann ein SPS-Pro-  
 gramm synchron zum Triggersignal ausgeführt werden.

Die Bausteinaufrufe erfolgen einmalig während der Initialisierungsphase (Kaltstart-/  
 Warmstart-Task).

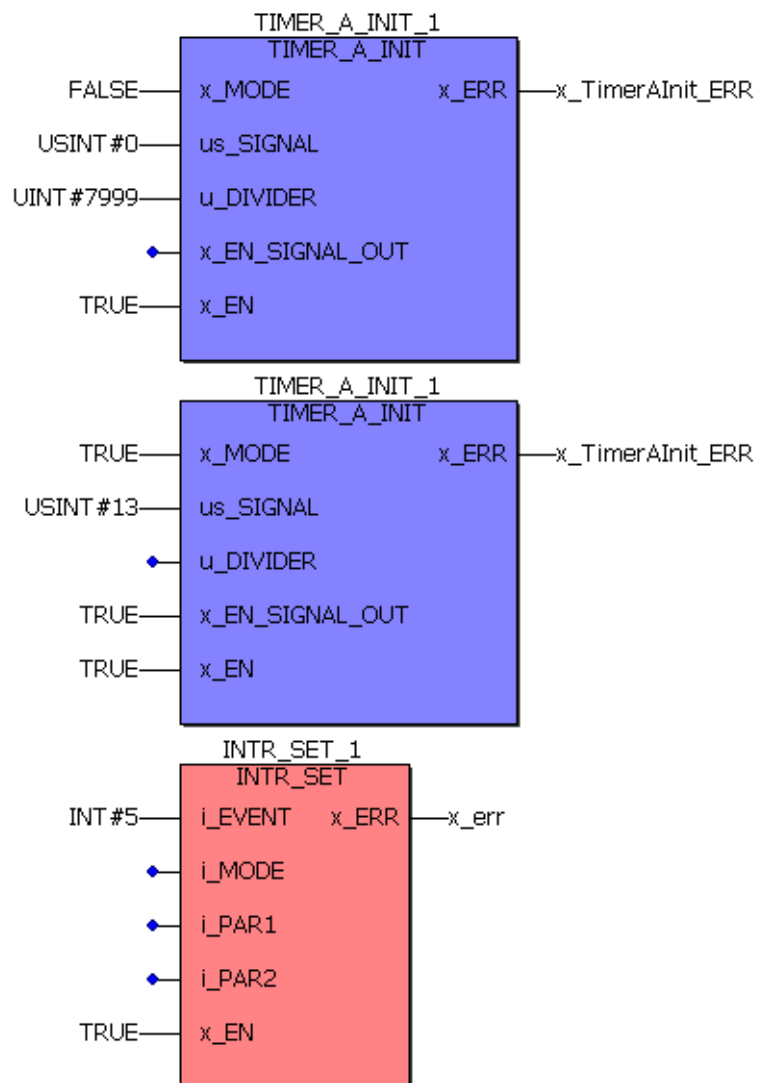


Abbildung 57: Das Signal Board-Timer A löst alle 2 ms das SYNC-Signal 1 aus und das Signal Board-Timer A löst die Bypass Event-Task 5 aus

Der Programmcode wird in einer Event-Task, die dem Ereignis "5 Timer A Level 14" zugeordnet ist, abgearbeitet.



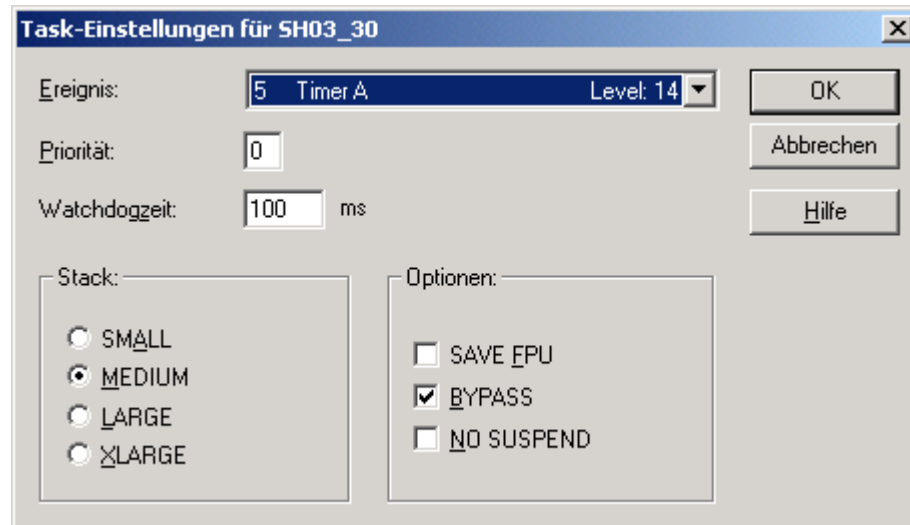


Abbildung 58: Einstellungen für die Bypass Event-Task auf das Ereignis 5 Timer A Level 14

### 5.6.8.2 Board-Timer A Interrupt auf BMC-M-PLC-02

Die b maXX controller PLC soll auf der SYNC-Signal 1 Leitung ein Triggersignal mit 2 ms Periode erzeugen. Synchron zum generierten Triggersignal soll ein SPS-Programmteil ausgeführt werden (mit dem Interrupt-Level 14).

1. Schritt: Internes TA-Signal erzeugen über die Timer-Funktion mit der Zeitperiode 2 ms: FB "TIMER\_A\_INIT" ausführen mit: `x_MODE := FALSE; us_SIGNAL := USINT#0;` (für Timer-Funktion); `u_DIVIDER = (2000 µs / 1 µs) - 1 = UINT#1999; x_EN := TRUE;`
2. Schritt: TA-Signal als Triggersignal an SYNC1 ausgeben: FB "TIMER\_A\_INIT" ausführen mit: `x_MODE := TRUE; us_SIGNAL := USINT#13;` (Für Ziel-Auswahl SYNC1); `x_EN_SIGNAL_OUT := TRUE; x_EN := TRUE;`
3. Schritt: FB "INTR\_SET" ausführen mit `i_EVENT := INT#5.`  
=> Durch Anlegen einer EVENT-Bypass-Task mit dem Ereignis „5“ kann ein SPS-Programm synchron zum Triggersignal ausgeführt werden.

Die Bausteinaufrufe erfolgen einmalig während der Initialisierungsphase (Kaltstart-/Warmstart-Task).

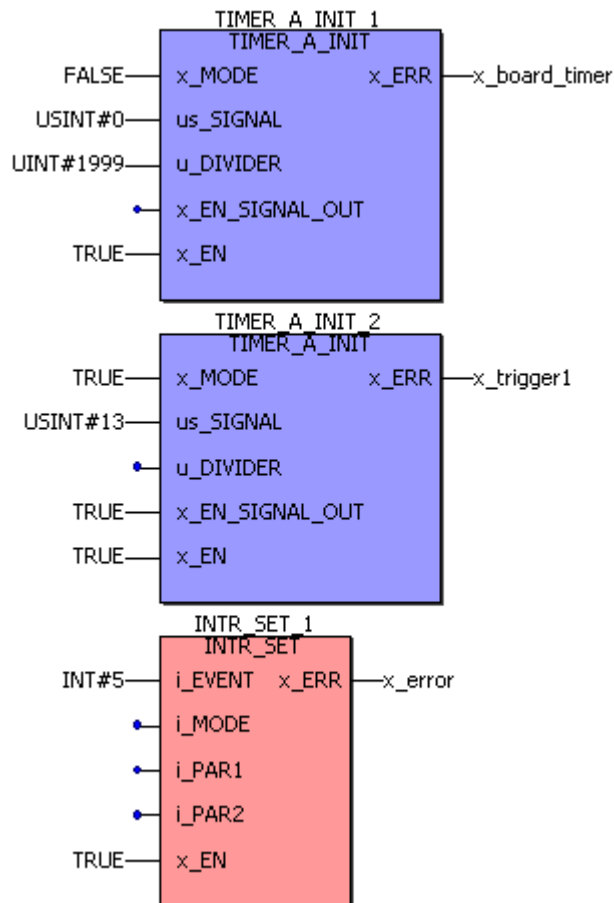


Abbildung 59: Das Signal Board-Timer A löst alle 2 ms das SYNC-Signal 1 aus und das Signal Board-Timer A löst die Bypass Event-Task 5 aus

Der Programmcode wird in einer Event-Task, die dem Ereignis "5 Timer A Level 14" zugeordnet ist, abgearbeitet.

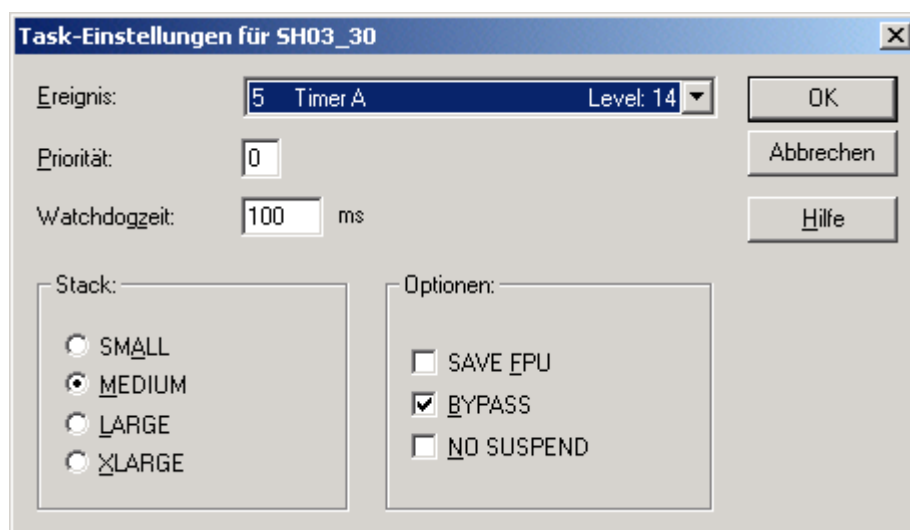


Abbildung 60: Einstellungen für die Bypass Event-Task auf das Ereignis 5 Timer A Level 14

### 5.6.9 Beispiel C: Realisierung einer einfachen seriellen RS485-Kommunikation mit Hilfe eines CPU-Timers

Einrichten einer Timer-Event-Task für den zyklischen Aufruf, in der eine serielle RS485-Kommunikation (über die 9-pol. Sub-D-Buchse auf der b maXX controller PLC Schnittstelle X2) bearbeitet werden soll. Die Funktionsbausteine zur Kommunikation werden in einer POE platziert, die dieser Task zugeordnet sind. (Details zur Anschlussbelegung und Kabel der X2-Schnittstelle siehe Betriebsanleitung b maXX controller PLC).

Kalt-/Warmstart-Task:

```

UINT #0—terminal_init[0]  (* protocol type: interface to ASCII-protocol *)
UINT #0—terminal_init[1]  (* protocol mode: RS485-Pull-Up-resistor on (activ)
                           RS485-terminator-resistors off (inactiv) *)
UINT #0—terminal_init[2]  (* bidirectional communication activated *)
UINT #10—terminal_init[3] (* baudrate: 38400 Bd *)
UINT #2—terminal_init[4]  (* parity: even *)
UINT #2—terminal_init[5]  (* 8 databits *)
UINT #1—terminal_init[6]  (* 1 Stopbit *)

```

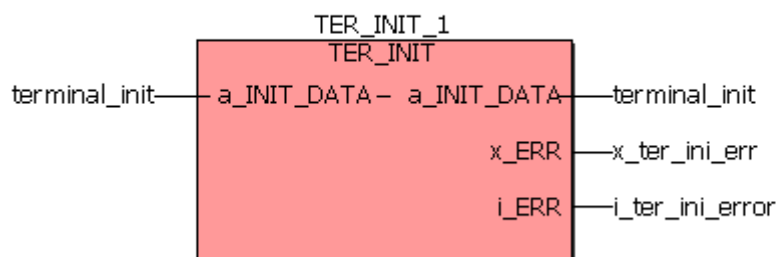


Abbildung 61: Initialisierung der seriellen Schnittstelle

Der Bausteinaufruf von "TER\_INIT" erfolgt einmalig während der Initialisierungsphase (Kalt/Warmstart-Task). (Details zu "TER\_INIT" und den seriellen Schnittstellenparametern siehe [Beschreibung der Bausteine für die serielle RS485-Kommunikation über die Schnittstelle X2 \(9-pol. Sub-D-Buchse\) auf der b maXX controller PLC](#) ab Seite 101 oder Online-Beschreibung in "SYSTEM1\_C\_PLC01\_20bd00 (PROPROG wt II) bzw. SYSTEM1\_C\_PLC01\_30bd00 / SYSTEM1\_C\_PLC02\_30bd02 (ProProg wt III) oder höher").

(\* Event #2: CPU-Timer 2 Level 13;  
i\_PAR1=#200 => 10 mSec Interrupt intervall time \*)

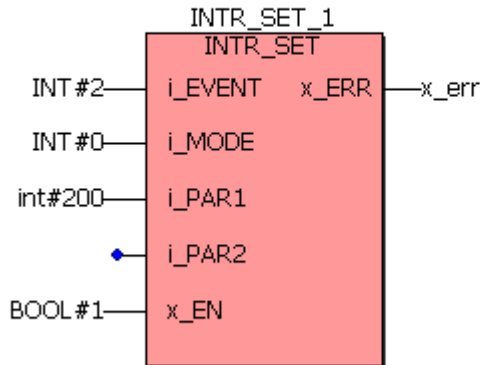


Abbildung 62: Einrichten eines 10 ms Bypass-Interrupts Level 13, in der die Sende- und Empfangsbausteine bearbeitet werden sollen:

Der Bausteinaufruf von TER\_INIT und INTR\_SET erfolgt einmalig während der Initialisierungsphase (Kalt/Warmstart-Task).

Serielle Kommunikation in der BYPASS-Task CPU-Timer 2:

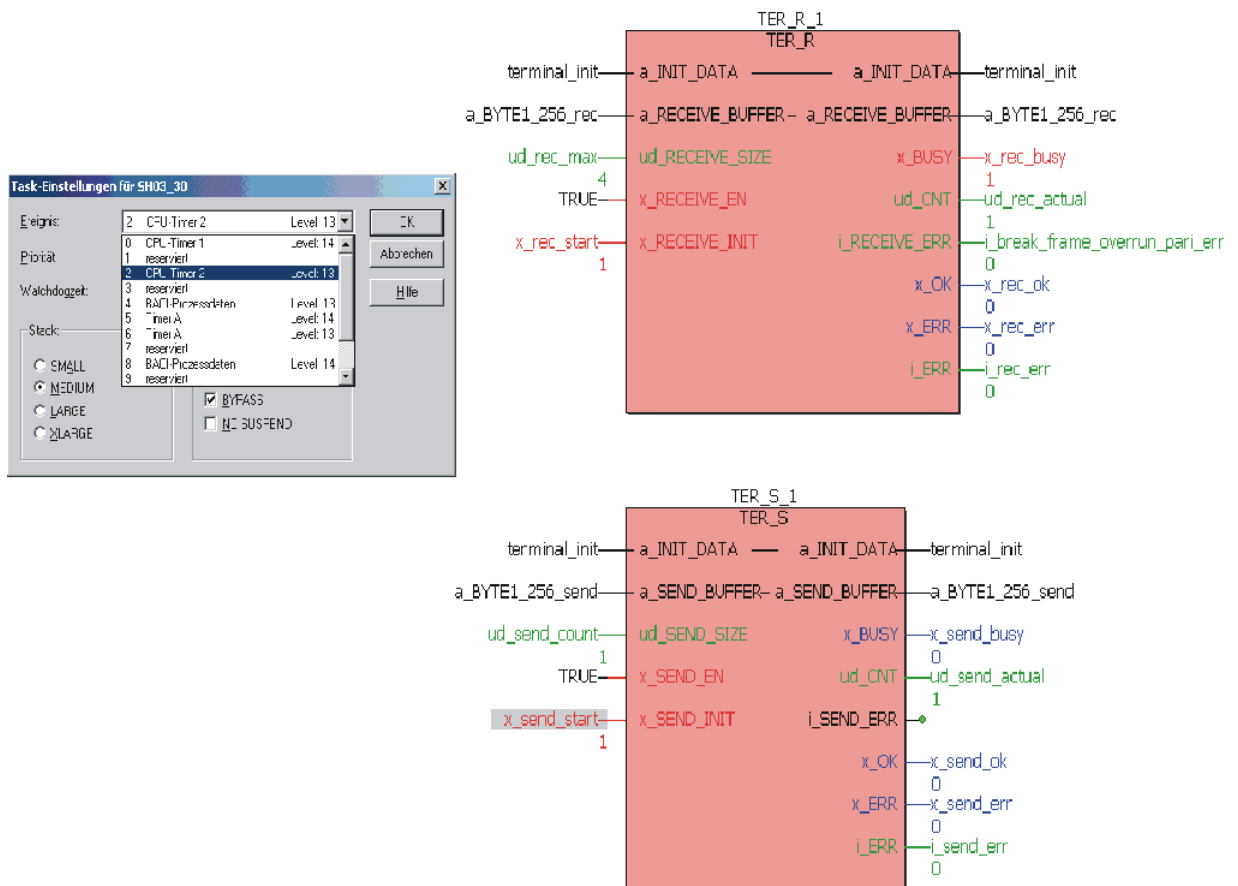


Abbildung 63: Einstellungen für die Bypass Event-Task auf das Ereignis 2 CPU-Timer 2 Level 13 und Aufruf der Sende- und Empfangsbausteine

### Kurzbeschreibung

(Details zu den FW-Bausteinen siehe [Beschreibung der Bausteine für die serielle RS485-Kommunikation über die Schnittstelle X2 \(9-pol. Sub-D-Buchse\) auf der b maXX controller PLC](#) ab Seite 101 oder Online-Beschreibungen in "SYSTEM1\_C\_PLC01\_20bd00" (PROPROG wt II) bzw. SYSTEM1\_C\_PLC01\_30bd00 / SYSTEM1\_C\_PLC02\_30bd02 (ProProg wt III) oder höher):

Empfangsbaustein "TER\_R":

Über eine steigende Flanke an x\_RECEIVE\_INIT wird der Empfangsauftrag aktiviert. Bis nicht alle Zeichen empfangen worden sind (Anzahl an "ud\_RECEIVE\_SIZE"), bleibt x\_BUSY auf TRUE, und am Ausgang "ud\_CNT" werden die bisher empfangenen und im Array "a\_BYTE1\_256\_rec" abgelegten Zeichen angezeigt.

Sendebaustein "TER\_S":

Über eine steigende Flanke an x\_SEND\_INIT wird der Sendeauftrag aktiviert: Alle "ud\_SEND\_SIZE" Zeichen im Sendebuffer-Array "a\_BYTE1\_256\_send" werden gesendet. Sobald der Sendeauftrag komplett ausgeführt worden ist, wird x\_BUSY zurückgesetzt und x\_OK gesetzt.

### 5.6.10 Beschreibung der Bausteine für die serielle RS485-Kommunikation über die Schnittstelle X2 (9-pol. Sub-D-Buchse) auf der b maXX controller PLC

Details zur Anschlussbelegung und Kabel der X2-Schnittstelle siehe Betriebsanleitung "b maXX controller PLC".



#### VORSICHT (CAUTION)

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

- Sachschaden

*Die Gefahr ist: **Kurzschluss** in PLC oder Beschädigung der am Pin 2 der Schnittstelle X2 angeschlossenen Teilnehmer.*

Die + 5V an Pin 2 der Schnittstelle X2 an der BMC-M-PLC-01 Sub-D-Buchse bzw. an X2 der BMC-M-PLC-02 Sub-D-Buchse sind nur für die Versorgung von externen Baumüller-eigenen RS485/RS232-Umsetzern vorgesehen und dürfen nicht kurzgeschlossen oder im Ring miteinander durchverbunden werden.



### HINWEIS

zur RS485-Schnittstelle

Die RS485-Schnittstelle X2 ist galvanisch getrennt und optisch entkoppelt.

Die RS485 ist eine mögliche Erweiterung der RS422 auf Mehrsenderbetrieb (= Kommunikation erfolgt wahlweise mit mehreren Sendern je Übertragungsweg, wobei zeitgleich immer nur max. ein Sender je Übertragungsweg aktiv sein darf). Deswegen muss bei einer RS485 der Sende-Treiber prinzipiell abschaltbar sein. Funktionell schließt eine RS485 auch die RS422 mit ein.

Üblicherweise werden die differentiellen Einzelleitungen miteinander verdreht, womit eingekoppelte elektromagnetische Störungen auf beide Einzelleitungen gleichmäßig wirken. Durch die Differenzauswertung am Empfangsbaustein können diese Störungen dann unterdrückt werden (Gleichtaktunterdrückung). Die Verdrehung steigert somit die Übertragungsqualität.

Empfehlung: Zusätzlich zur Zwei/Vierdrahtleitung kann auch die Masse-Leitung mitgeführt werden, womit beide Treiberbausteine (Sende-Treiber auf der einen Seite und Empfangs-Treiber auf der anderen Seite der Verbindung) das gleiche absolute Bezugspotential besitzen. Dadurch werden Überschreitungen des Gleichtakteingangsspannungsbereiches der Empfangs-Treiber vermieden.

Die Masse-Leitung darf nirgends auf der gesamten Verbindungsstrecke mit dem Schirm verbunden werden!

- Konfigurationsmöglichkeit bei der b maXX controller PLC über die Software - Verwendung der auf der PLC integrierten Pull-Up-, Pull-Down- und Abschlusswiderstände:

Beim Einsatz differentieller Schnittstellen, deren Sende-Treiber abgeschaltet werden können, besteht oftmals die Notwendigkeit, Pull-Up/Down-Widerstände und/oder Abschlusswiderstände den Verbindungskabeln zuzufügen.

Durch den Einsatz von Pull-Up/Down-Widerständen soll sichergestellt werden, dass an einer differentiellen Empfangseinheit stets ein definiertes Potential anliegt, auch wenn gerade keine Zeichen übertragen werden und die zugeordnete Sendeeinheit des gegenüberliegenden Partners seine Treiber hochohmig schaltet (protokollabhängig).

Fehlen solche Widerstände, so kann die Leitung floaten mit der Folge, dass sporadische Übertragungsfehler auftreten wie "Break Error", "Framing Error", "Parity Error" etc.

Besonders bei Verwendung von längeren Leitungen und höheren Übertragungsraten können die Kabelwellenwiderstände nicht mehr vernachlässigt werden. Deshalb werden die differentiellen Leitungen an den beiden Bus-Enden der Übertragungsstrecke dann mit Abschlusswiderständen abgeschlossen. Fehlen diese, können Reflexionen auftreten, welche zu Empfangsfehlern führen. Die Verwendung von Busabschlüssen verbessert somit die Übertragungsqualität.

Bei netzartigen Strukturen mit mehreren Teilnehmern auf einer Leitung sollte man jedoch darauf achten, dass auf dieser Leitung nicht mehrere Pull-Up/Down-Widerstände und nicht zuviele Abschlusswiderstände (höchstens die beiden an den Bus-Enden) parallel liegen.

Um die umständliche Prozedur "Pull-Up/Down- und Abschlusswiderstände direkt in den Stecker oder in das Kabel integrieren" zu verhindern, sind Pull-Up/Down- und Ab-

schlusswiderstände in der b maXX controller PLC bereits vorhanden und können jeweils getrennt voneinander über die Applikations-Software zu- und abgeschaltet werden.

Die genaue Vorgehensweise dazu kann der FB-Beschreibung "TER\_INIT" in der PROPROG wt II Firmware-Bibliothek „SYSTEM2\_C\_PLC01\_20bd00“ bzw. der ProProg wt III Firmware-Bibliothek SYSTEM2\_C\_PLC01\_30bd00 oder SYSTEM2\_C\_PLC02\_30bd02 oder höher (siehe [>Funktionsbaustein TER\\_INIT<](#) ab Seite 103) bzw. deren Online-Beschreibung entnommen werden.

- Konfigurationsmöglichkeit bei der b maXX controller PLC - Zweidraht / Vierdraht

Unterschied Zweidraht/Vierdraht:

Über eine Vierdrahtleitung (+ Masseleitung empfohlen) können Zeichen voll duplex gesendet werden. Voll duplex bedeutet immer, dass Zeichen über getrennte Kanäle zeitgleich in beiden Richtungen gesendet und empfangen werden können. Das 3964R<sup>®</sup>-Protokoll ist z. B. ein solches Voll duplex-Protokoll.

Im Unterschied dazu gibt es auch Protokolle im Halbduplexverfahren, die ihre Send- und Empfangsaufträge über eine gemeinsame Zweidrahtleitung (+ Masseleitung empfohlen) abwickeln können. Bei einer Zweidrahtleitung sind an den beiden Enden des Verbindungskabels die TxD+ mit der RxD+-Leitung kurzgeschlossen, sowie die TxD- mit der RxD- Leitung. Dadurch brauchen im Gegensatz zum Vierdraht zwei Einzelleitungen weniger verlegt werden (= Materialersparnis).

Das Fehlen von den beiden Einzelleitungen muss aber durch einen exakten Protokollablauf "kompensiert" werden:

Das Protokoll muss zu jedem Zeitpunkt sicherstellen, dass Zeichen nie in beiden Richtungen gleichzeitig übertragen werden und dass immer nur maximal eine Sendeeinheit aktiviert ist und dabei die andere gegenüberliegende Sendeeinheit dann abgeschaltet (hochohmig) ist.

Hierzu gibt es die folgende Konfigurationsmöglichkeit bei der b maXX controller PLC:

Es kann zwischen Zweidraht- und Vierdrahtmodus ausgewählt werden.

Unterschied: Im Zweidraht-Modus erfolgt das Abschalten der Sendeeinheit, nachdem alle Zeichen gesendet wurden. Zu beachten ist dabei, dass durch das Abschalten die Verbindungsleitungen hochohmig werden, falls keine Pull-Up/Down-Widerstände programmiert wurden.

Die genaue Vorgehensweise für die Einstellmöglichkeiten kann der FB-Beschreibung "TER\_INIT" der PROPROG wt II Standard-Bibliothek "SYSTEM2\_C\_PLC01\_20bd00" bzw. der ProProg wt III Firmware-Bibliothek SYSTEM2\_C\_PLC01\_30bd00 oder SYSTEM2\_C\_PLC02\_30bd02 oder höher bzw. deren Online-Beschreibung entnommen werden.

### 5.6.11 Funktionsbaustein TER\_INIT

Der Firmware-FB TER\_INIT initialisiert die Terminal-Schnittstelle (serielle Schnittstelle) der b maXX controller PLC mit den notwendigen Schnittstellen-Parametern wie Baudrate, Parity, usw.

Der FB TER\_INIT ist in der FW-Bibliothek „SYSTEM2\_C\_PLC\_20bd00“ (PROProg wt II) bzw. „SYSTEM2\_C\_PLC01\_30bd00 / SYSTEM2\_C\_PLC02\_30bd02“ (ProProg wt III) oder höher enthalten.

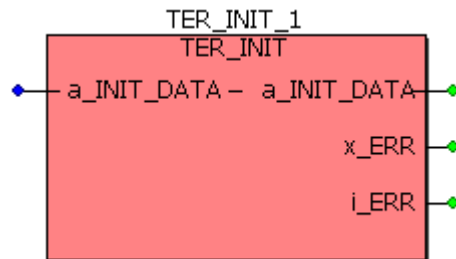


Abbildung 64: Funktionsbaustein TER\_INIT

Parameter Eingang	Datentyp	Beschreibung
a_INIT_DATA	UINT_256_BMARRAY	Initialisierungs-Array für die Terminalschnittstelle

Parameter Ausgang	Datentyp	Beschreibung
a_INIT_DATA	UINT_256_BMARRAY	Initialisierungs-Array für die Terminalschnittstelle
x_ERR	BOOL	Fehlerbit
i_ERR	INT	Fehlerwort 0, 1, 2, 3, 4



### HINWEIS

Es müssen alle Eingänge belegt werden, da sonst kein definierter Wert übergeben wird!

### Typdefinition:

```
UINT_256_BMARRAY : ARRAY [0..255] OF UINT
```

### Beschreibung:

Bevor eine Kommunikation über die Terminal-Schnittstelle (serielle Schnittstelle) der b maXX controller PLC erfolgen kann, muss diese parametrisiert werden. Dies erfolgt über den Anschluss a\_INIT\_DATA. Die dort angeschlossene Variable enthält die notwendigen Schnittstellen-Parameter wie Baudrate, Parity usw... Die eigentliche Kommunikation erfolgt über entsprechende Kommunikations-FBs (siehe unten).

Da die Terminal-Schnittstelle nur einmal initialisiert wird, muss der Aufruf von TER\_INIT in der Kalt- und in der Warmstart-Task erfolgen.



Eingang **a\_INIT\_DATA**:

Am Eingang **a\_INIT\_DATA** wird eine Variable vom Typ **UINT\_256\_BMARRAY** angeschlossen. Die einzelnen Einträge von **a\_INIT\_DATA** haben folgende Bedeutung:

<b>a_INIT_DATA[Index]</b>	<b>Bedeutung</b>
a_INIT_DATA[0]	Protokoll 0: beliebiges ASCII-Protokoll 1: Anschaltung an Prozedur 3964R <sup>®</sup> auf Datenbausteine
a_INIT_DATA[1]	Betriebsart des Protokolls 0: nur RS485-Pull-Up-Widerstände sind aktiv (on) 1: nur RS485-Abschlusswiderstand ist aktiv (on) 2: RS485-Pull-Up-Widerstände und RS485-Abschlusswiderstand sind aktiv (on) 3: RS485-Pull-Up-Widerstände und RS485-Abschlusswiderstand sind nicht aktiv (off)
a_INIT_DATA[2]	Betriebsart RS485 0: Vierdraht-Betrieb RS485 1: Zweidraht-Betrieb RS485 <sup>1)</sup>
a_INIT_DATA[3]	Baudrate 0: 50 Bit/s <sup>1)</sup> 1: 110 Bit/s <sup>1)</sup> 2: 150 Bit/s <sup>1)</sup> 3: 300 Bit/s <sup>1)</sup> 4: 600 Bit/s <sup>1)</sup> 5: 1200 Bit/s <sup>1)</sup> 6: 2400 Bit/s <sup>1)</sup> 7: 4800 Bit/s 8: 9600 Bit/s 9: 19200 Bit/s 10: 38400 Bit/s
a_INIT_DATA[4]	Parity 0: kein Parity-Bit 1: ungerade (odd) 2: gerade (even)
a_INIT_DATA[5]	Zeichen-Länge 0: Reserviert 1: 7 Bit je Zeichen 2: 8 Bit je Zeichen
a_INIT_DATA[6]	Stopp-Bit 0: 1 Stopp-Bit 1: 1 Stopp-Bit 2: 1 Stopp-Bit 3: 2 Stopp-Bit
a_INIT_DATA[7]	Reserviert
a_INIT_DATA[8]	max. Anzahl der Kommunikationsversuche je Telegramm <sup>2)</sup>
a_INIT_DATA[9]	Priorität <sup>2)</sup> 0: niedrige Priorität 1: hohe Priorität

a_INIT_DATA[Index]	Bedeutung
a_INIT_DATA[10]	Zeichenverzugszeit (ZVZ) in ms <sup>2)</sup>
a_INIT_DATA[11]	Quittungsverzugszeit (QVZ) in ms <sup>2)</sup>
a_INIT_DATA[12]	Blockverzugszeit (BVZ) in ms <sup>2)</sup>
a_INIT_DATA[13]	Reserviert
...	...
a_INIT_DATA[255]	Reserviert

1) nicht für Anschaltung an Prozedur 3964R<sup>®</sup> auf Datenbausteine

2) nur für Anschaltung an Prozedur 3964R<sup>®</sup> auf Datenbausteine

Die Prozedur 3964R<sup>®</sup> ist ein eingetragenes Warenzeichen der Firma Siemens AG.

### a) ASCII-Protokoll

Die Kommunikation mit dem Protokoll "beliebiges ASCII-Protokoll" erfolgt über die FBs TER\_S und TER\_R. Der Anwender kann damit n Zeichen an einen Kommunikations-Partner senden und n Zeichen von einem Kommunikations-Partner empfangen.

Die FBs TER\_S und TER\_R sind in der FW-Bibliothek „SYSTEM2\_C\_PLC01\_20bd00“ (PROPROG wt II) bzw. „SYSTEM2\_C\_PLC01\_30bd00 / SYSTEM2\_C\_PLC02\_30bd02“ (ProProg wt III) oder höher enthalten.

Typische Initialisierungswerte für das ASCII-Protokoll sind:

a_INIT_DATA[Index]	Bedeutung
a_INIT_DATA[0]	Protokoll 0: beliebiges ASCII-Protokoll
a_INIT_DATA[1]	Betriebsart des Protokolls 0: nur RS485-Pull-Up-Widerstände sind aktiv (on) 1: nur RS485-Abschlusswiderstand ist aktiv (on) 2: RS485-Pull-Up-Widerstände und RS485-Abschlusswiderstand sind aktiv (on) 3: RS485-Pull-Up-Widerstände und RS485-Abschlusswiderstand sind nicht aktiv (off)
a_INIT_DATA[2]	Betriebsart RS485 0: Vierdraht-Betrieb RS485 1: Zweidraht-Betrieb RS485
a_INIT_DATA[3]	Baudrate 0: 50 Bit/s 1: 110 Bit/s 2: 150 Bit/s 3: 300 Bit/s 4: 600 Bit/s 5: 1200 Bit/s 6: 2400 Bit/s 7: 4800 Bit/s 8: 9600 Bit/s 9: 19200 Bit/s 10: 38400 Bit/s

a_INIT_DATA[Index]	Bedeutung
a_INIT_DATA[4]	Parity 0: kein Parity-Bit 1: ungerade (odd) 2: gerade (even)
a_INIT_DATA[5]	Zeichen-Länge 1: 7 Bit je Zeichen 2: 8 Bit je Zeichen
a_INIT_DATA[6]	Stopp-Bit 0: 1 Stopp-Bit 1: 1 Stopp-Bit 2: 1 Stopp-Bit 3: 2 Stopp-Bit
a_INIT_DATA[7]	Reserviert 0
...	...
a_INIT_DATA[255]	Reserviert 0

#### b) Anschaltung an Prozedur 3964R®

Die Anschaltung an die Prozedur 3964R® auf Datenbausteine erfolgt über die Firmware-FBs T64\_RSYN und T64\_REC (diese FBs sind in der FW-Bibliothek „SYSTEM2\_C\_PLC01\_20bd00“ (PROPROG wt II) bzw. „SYSTEM2\_C\_PLC01\_30bd00 / SYSTEM2\_C\_PLC02\_30bd02“ (ProProg wt III) oder höher enthalten).

Typische Initialisierungswerte für die Anschaltung an die Prozedur 3964R® auf Datenbausteine sind:

a_INIT_DATA[Index]	Bedeutung
a_INIT_DATA[0]	Protokoll 1: Anschaltung an Prozedur 3964R® auf Datenbausteine
a_INIT_DATA[1]	Betriebsart des Protokolls 0: nur RS485-Pull-Up-Widerstände sind aktiv (on)
a_INIT_DATA[2]	Betriebsart RS485 0: Vierdraht-Betrieb RS485
a_INIT_DATA[3]	Baudrate (Baud=Bit/s) 7: 4800 Bit/s 8: 9600 Bit/s 9: 19200 Bit/s
a_INIT_DATA[4]	Parity 2: gerade (even)
a_INIT_DATA[5]	Zeichen-Länge 2: 8 Bit je Zeichen
a_INIT_DATA[6]	Stopp-Bit 1: 1 Stopp-Bit

a_INIT_DATA[Index]	Bedeutung
a_INIT_DATA[7]	Reserviert 0
a_INIT_DATA[8]	max. Anzahl der Kommunikationsversuche je Telegramm 2: max. 2 Kommunikationsversuche je Telegramm
a_INIT_DATA[9]	Priorität 0: niedrige Priorität
a_INIT_DATA[10]	Zeichenverzugszeit (ZVZ) in ms 100
a_INIT_DATA[11]	Quittungsverzugszeit (QVZ) in ms 500
a_INIT_DATA[12]	Blockverzugszeit (BVZ) in ms 2000
a_INIT_DATA[13]	Reserviert 0
...	...
a_INIT_DATA[255]	Reserviert 0

Fehlerauswertung:

Wird eine falsche Parametrierung festgestellt, wird das Fehlerbit x\_ERR gesetzt und der Fehler über die Fehlernummer i\_ERR genauer spezifiziert:

i_ERR	Fehler serielle Schnittstelle
0	kein Fehler
1 bis 3	Reserviert
4	eingestellte Baudrate nicht möglich

### 5.6.12 Funktionsbaustein TER\_R

Der Firmware-FB TER\_R empfängt Zeichen über die Terminal-Schnittstelle (serielle Schnittstelle) der b maXX controller PLC und legt die Zeichen in einem Array ab.

Der FB TER\_R ist in der FW-Bibliothek „SYSTEM2\_C\_PLC01\_20bd00“ (PROPROG wt II) bzw. „SYSTEM2\_C\_PLC01\_30bd00 / SYSTEM2\_C\_PLC02\_30bd02“ (ProProg wt III) oder höher enthalten.

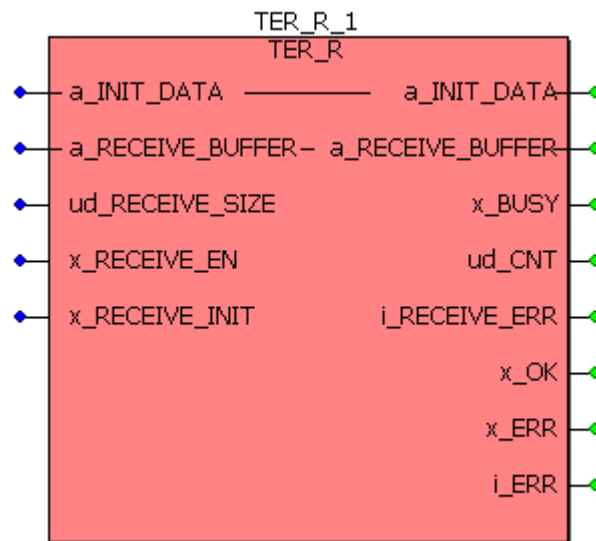


Abbildung 65: Funktionsbaustein TER\_R

Parameter Eingang	Datentyp	Beschreibung
a_INIT_DATA	UINT_256_BMARRAY	Initialisierungs-Array für die Terminalschnittstelle
a_RECEIVE_BUFFER	ANY <sup>1)</sup>	Array in das eingelesen wird
ud_RECEIVE_SIZE	UDINT	Anzahl der Bytes, die empfangen werden soll
x_RECEIVE_EN	BOOL	Schnittstellen-Empfangs-Interrupt sperren/freigeben
x_RECEIVE_INIT	BOOL	Empfangs-Auftrag absetzen

Parameter Ausgang	Datentyp	Beschreibung
a_INIT_DATA	UINT_256_BMARRAY	Initialisierungs-Array für die Terminalschnittstelle
a_RECEIVE_BUFFER	ANY <sup>1)</sup>	Array in das eingelesen wird
x_BUSY	BOOL	zeigt an, ob Auftrag aktiv ist
ud_CNT	UDINT	Anzahl der bereits empfangenen Zeichen
i_RECEIVE_ERR	INT	Anzeige serieller Empfangs-Fehler
x_OK	BOOL	zeigt erfolgreich aktivierten Auftrag an
x_ERR	BOOL	Fehlerbit
i_ERR	INT	Fehlernummer

<sup>1)</sup> Es ist ein Array von der Größe 256 Byte anzuschließen. Es eignen sich z. B. der Datentyp BYTE\_256\_BMARRAY (: ARRAY [0..255] OF BYTE) oder der Datentyp DINT\_64\_BMARRAY (: ARRAY [0..63] OF DINT) aus der Bibliothek BM\_TYPES\_20bd06 (PROPROP wt II) bzw. BM\_TYPES\_30bd01 (ProProg wt III) und höher.



### HINWEIS

Es müssen alle Eingänge belegt werden, da sonst kein definierter Wert übergeben wird!

#### Typdefinition:

```
UINT_256_BMARRAY : ARRAY [0..255] OF UINT
```

#### Beschreibung:

Der Firmware-FB TER\_R muss zyklisch im Anwenderprogramm aufgerufen werden.

Für die Kommunikation mit den Firmware-FBs TER\_S und TER\_R sind die hardware-spezifischen Schnittstellen-Parameter am Firmware-FB TER\_INIT einzustellen (siehe Beschreibung Firmware-FB TER\_INIT).

Mit dem Protokoll "beliebiges ASCII-Protokoll" kann der Anwender mit dem Firmware-FB TER\_S n Zeichen an einen Kommunikations-Partner senden. Mit dem Firmware-FB TER\_R kann der Anwender n Zeichen von einem Kommunikations-Partner empfangen.

#### Ein-/Ausgang a\_INIT\_DATA:

An a\_INIT\_DATA wird das parametrierte Initialisierungs-Array für die Terminal-Schnittstelle angeschlossen (siehe Beschreibung Firmware-FB TER\_INIT). D. h. an a\_INIT\_DATA ist die gleiche globale Variable anzuschließen wie am Firmware-FB TER\_INIT.

#### Ein-/Ausgang a\_RECEIVE\_BUFFER:

An a\_RECEIVE\_BUFFER wird ein Array (256 Byte, siehe oben) angeschlossen. Die empfangenen (und im systeminternen Terminal-Schnittstellen-Empfangsbuffer eingetragenen) Daten/Zeichen werden abgeholt und in diesem Array abgelegt.

#### Eingang ud\_RECEIVE\_SIZE:

Am Eingang ud\_SEND\_SIZE wird die Anzahl der Zeichen angegeben, die vom systeminternen Empfangsbuffer der Terminal-Schnittstelle in das Array am Ein-/Ausgang a\_RECEIVE\_BUFFER abgelegt werden.

#### Eingang x\_RECEIVE\_EN:

Der Terminal-Schnittstellen-Empfangs-Interrupt kann bei Aktivierung eines Auftrags (steigende Flanke an x\_RECEIVE\_INIT) entweder freigegeben (x\_RECEIVE\_EN = TRUE) oder gesperrt (x\_RECEIVE\_EN = FALSE) werden. Ist x\_RECEIVE\_EN = FALSE werden im systeminternen Terminal-Schnittstellen-Empfangsbuffer alle bisher empfangenen Zeichen (unabhängig vom Zustand des Eingangs x\_RECEIVE\_INIT) gelöscht. Bei gesperrtem Interrupt erfolgt kein Zeichenempfang in dem systeminternen Empfangsbuffer mehr.

**Eingang x\_RECEIVE\_INIT:**

Durch eine steigende Flanke am Eingang x\_RECEIVE\_INIT wird ein neuer Auftrag zum Empfang von "ud\_RECEIVE\_SIZE" Zeichen (aus dem systeminternen Terminal-Schnittstellen-Empfangsbuffer) aktiviert.

**Ausgang x\_BUSY:**

Der Ausgang x\_BUSY ist TRUE solange ein Auftrag ausgeführt wird. Wurden alle Zeichen aus dem systeminternen Terminal-Schnittstellen-Empfangsbuffer abgeholt und in das Array am Ein-Ausgang a\_RECEIVE\_BUFFER eingetragen, ist der Auftrag erfolgreich ausgeführt, und x\_BUSY wird FALSE.

**Ausgang ud\_CNT:**

Am Ausgang ud\_CNT wird die Anzahl der bis zu diesem Zeitpunkt (aus dem Terminal-Schnittstellen-Empfangsbuffer) empfangenen Zeichen angezeigt.

**Ausgang x\_OK:**

Der Ausgang x\_OK zeigt mit x\_OK = TRUE einen Zyklus lang an, ob ein neuer Auftrag (steigende Flanke am Eingang x\_RECEIVE\_INIT) übernommen wurde. Ein Auftrag wird auch dann übernommen, wenn gerade ein Empfangsauftrag aktiv (x\_BUSY = TRUE) ist.

D. h. der laufende Auftrag wird sofort abgebrochen und der neue aktiviert.

**Fehlerauswertung:**

Der Ausgang i\_RECEIVE\_ERR zeigt Empfangs-Fehler an.

i_RECEIVE_ERR	Fehler serielle Schnittstelle
0	kein Fehler
1	BREAK-Fehler
2	FRAME-Fehler
3	PARITY-Fehler
4	OVERRUN-Fehler
5	Interner Empfangsfehler RxRDY

**Ausgänge x\_ERR, i\_ERR:**

Wird eine falsche Parametrierung festgestellt, wird der Ausgang x\_ERR = TRUE gesetzt und der Fehler über die Fehlernummer (Ausgang i\_ERR) genauer spezifiziert:

i_ERR	Fehler serielle Schnittstelle
0	kein Fehler
1 bis 2	Reserviert
3	Terminal-Schnittstelle wurde nicht initialisiert. Bitte überprüfen Sie den Aufruf von TER_INIT in der Kalt- und in der Warmstart-Task sowie alle a_INIT_DATA-Anschlüsse

### 5.6.13 Funktionsbaustein TER\_S

Der Firmware-FB TER\_S sendet Zeichen aus einem Array über die Terminal-Schnittstelle (serielle Schnittstelle) der b maXX controller PLC an einen Kommunikations-Partner.

Der FB TER\_S ist in der FW-Bibliothek „SYSTEM2\_C\_PLC01\_20bd00“ (PROPROG wt II) bzw. „SYSTEM2\_C\_PLC01\_30bd00 / SYSTEM2\_C\_PLC02\_30bd02“ (ProProg wt III) oder höher enthalten.

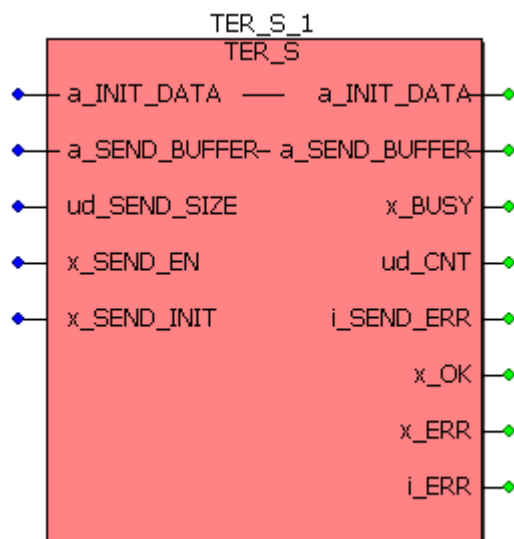


Abbildung 66: Funktionsbaustein TER\_S

Parameter Eingang	Datentyp	Beschreibung
a_INIT_DATA	UINT_256_BMARRAY	Initialisierungs-Array für die Terminalschnittstelle
a_SEND_BUFFER	ANY <sup>1)</sup>	Array aus dem gesendet wird
ud_SEND_SIZE	UDINT	Anzahl der Bytes, die gesendet werden soll
x_SEND_EN	BOOL	Schnittstellen-Sende-Interrupt sperren/freigeben
x_SEND_INIT	BOOL	Sende-Auftrag absetzen

Parameter Ausgang	Datentyp	Beschreibung
a_INIT_DATA	UINT_256_BMARRAY	Initialisierungs-Array für die Terminalschnittstelle
a_SEND_BUFFER	ANY <sup>1)</sup>	Array aus dem gesendet wird
x_BUSY	BOOL	zeigt an, ob Auftrag aktiv ist
ud_CNT	UDINT	Anzahl der bereits gesendeten Zeichen
i_SEND_ERR	INT	Anzeige serieller Sende-Fehler (zur Zeit nicht benutzt, also immer 0)
x_OK	BOOL	zeigt erfolgreich aktivierten Auftrag an



Parameter Ausgang	Datentyp	Beschreibung
x_ERR	BOOL	Fehlerbit
i_ERR	INT	Fehlernummer

- <sup>1)</sup> Es ist ein Array von der Größe 256 Byte anzuschließen. Es eignen sich z. B. der Datentyp BYTE\_256\_BMARRAY (: ARRAY [0..255] OF BYTE) oder der Datentyp DINT\_64\_BMARRAY (: ARRAY [0..63] OF DINT) aus der Bibliothek BM\_TYPES\_20bd06 (PROPROG wt II) bzw. BM\_TYPES\_30bd01 (ProProg wt III) und höher.



#### HINWEIS

Es müssen alle Eingänge belegt werden, da sonst kein definierter Wert übergeben wird!

#### Typdefinition:

```
UINT_256_BMARRAY : ARRAY [0..255] OF UINT
```

#### Beschreibung:

Der Firmware-FB TER\_S muss zyklisch im Anwenderprogramm aufgerufen werden.

Für die Kommunikation mit den Firmware-FBs TER\_S und TER\_R sind die hardware-spezifischen Schnittstellen-Parameter am Firmware-FB TER\_INIT einzustellen (siehe Beschreibung Firmware-FB TER\_INIT).

Mit dem Protokoll "beliebiges ASCII-Protokoll" kann der Anwender mit dem Firmware-FB TER\_S n Zeichen an einen Kommunikations-Partner senden. Mit dem Firmware-FB TER\_R kann der Anwender n Zeichen von einem Kommunikations-Partner empfangen.

#### Ein-/Ausgang a\_INIT\_DATA:

An a\_INIT\_DATA wird das parametrierte Initialisierungs-Array für die Terminal-Schnittstelle angeschlossen (siehe Beschreibung Firmware-FB TER\_INIT). D. h. an a\_INIT\_DATA ist die gleiche globale Variable anzuschließen wie am Firmware-FB TER\_INIT.

#### Ein-/Ausgang a\_SEND\_BUFFER:

An a\_SEND\_BUFFER wird ein Array (256 Byte, siehe oben) angeschlossen. In diesem Array werden vom Anwender die zu sendenden Daten/Zeichen abgelegt.

#### Eingang ud\_SEND\_SIZE:

Die Anzahl der zu sendenden Zeichen wird an ud\_SEND\_SIZE angegeben.

#### Eingang x\_SEND\_EN:

Der Schnittstellen-Sende-Interrupt kann bei Aktivierung eines Auftrags (steigende Flanke an x\_SEND\_INIT) entweder freigegeben (x\_SEND\_EN = TRUE) oder gesperrt (x\_SEND\_EN = FALSE) werden. Bei gesperrtem Interrupt erfolgt keine Bearbeitung ei-

nes gerade laufenden oder neu ausgelösten Sendeauftrags mehr. Wenn die Betriebsart RS485 auf "Zweidraht" eingestellt ist (siehe TER\_INIT), wird die RS485-Sende-Schnittstelle zusätzlich physikalisch gesperrt (= hochohmiger Zustand des Sende-Ausgangs für Mehr-Punkt-Verbindungen).

### Eingang **x\_SEND\_INIT**:

Durch eine steigende Flanke am Eingang **x\_SEND\_INIT** wird ein neuer Auftrag zum Senden von "**ud\_SEND\_SIZE**" Zeichen aktiviert. Die RS485-Sende-Schnittstelle wird physikalisch freigegeben.

### Ausgang **x\_BUSY**:

Der Ausgang **x\_BUSY** ist TRUE solange ein Auftrag ausgeführt wird. Wurden alle Zeichen aus dem Array **a\_SEND\_BUFFER** gesendet, ist der Auftrag erfolgreich ausgeführt, und **x\_BUSY** wird FALSE.

### Ausgang **ud\_CNT**:

Am Ausgang **ud\_CNT** wird die Anzahl der bis zu diesem Zeitpunkt gesendeten Zeichen angezeigt.

### Ausgang **i\_SEND\_ERR**:

Dieser Ausgang zeigt mit **i\_SEND\_ERR = INT#0** an, dass kein Sendefehler aufgetreten ist. Andere Werte von **i\_SEND\_ERR** sind reserviert.

### Ausgang **x\_OK**:

Der Ausgang **x\_OK** zeigt mit **x\_OK = TRUE** einen Zyklus lang an, ob ein neuer Auftrag (steigende Flanke am Eingang **x\_SEND\_INIT**) übernommen wurde. Ein Auftrag kann erst dann übernommen werden, wenn der zuvor laufende Sendevorgang bereits vollkommen abgeschlossen ist (**x\_BUSY = FALSE**) oder vom Anwender aktiv abgebrochen wurde (**x\_SEND\_EN = FALSE**). Solange der Ausgang **x\_BUSY = TRUE** ist, werden Sendeaufträge ignoriert (**x\_OK** bleibt FALSE).

### Fehlerauswertung:

Wird eine falsche Parametrierung festgestellt, wird der Ausgang **x\_ERR = TRUE** gesetzt und der Fehler über die Fehlernummer am Ausgang **i\_ERR** genauer spezifiziert:

<b>i_ERR</b>	<b>Fehler serielle Schnittstelle</b>
0	kein Fehler
1 bis 2	Reserviert
3	Terminal-Schnittstelle wurde nicht initialisiert. Bitte überprüfen Sie den Aufruf von TER_INIT in der Kalt- und in der Warmstart-Task sowie alle a_INIT_DATA-Anschlüsse

### 5.6.14 Funktionsbaustein T64\_RSYN

Der Firmware-FB T64\_RSYN bildet einen Datenbaustein einer Anschaltung an die Prozedur 3964R<sup>®</sup> auf ein Array ab. Die eigentliche Kommunikationsabwicklung erfolgt über den Firmware-FB T64\_REC.

Der FB T64\_RSYN ist in der FW-Bibliothek „SYSTEM2\_C\_PLC01\_20bd00“ (PROPROG wt II) bzw. „SYSTEM2\_C\_PLC01\_30bd00 / SYSTEM2\_C\_PLC02\_30bd02“ (ProProg wt III) oder höher enthalten.

Die Prozedur 3964R<sup>®</sup> ist ein eingetragenes Warenzeichen der Firma Siemens AG.

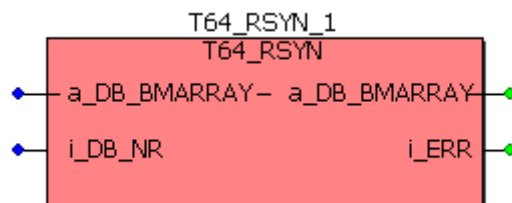


Abbildung 67: Funktionsbaustein T64\_RSYN

Parameter Eingang	Datentyp	Beschreibung
a_DB_BMARRAY	ANY <sup>1)</sup>	Array, das für die Kommunikation mit der Anschaltung an die Prozedur 3964R <sup>®</sup> auf Datenbausteine freigegeben werden soll.
i_DB_NR	INT	Nummer des Datenbausteins, mit der das Array vom Kommunikations-Partner angesprochen werden soll

Parameter Ausgang	Datentyp	Beschreibung
a_DB_BMARRAY	ANY <sup>1)</sup>	Array, das für die Kommunikation mit der Anschaltung an die Prozedur 3964R <sup>®</sup> auf Datenbausteine freigegeben werden soll.
i_ERR	INT	Fehlernummer

<sup>1)</sup> Es ist ein Array von der Größe 256 Byte anzuschließen. Es eignen sich z. B. der Datentyp SINT\_256\_BMARRAY (: ARRAY [0..255] OF SINT) oder der Datentyp DINT\_64\_BMARRAY (: ARRAY [0..63] OF DINT) aus der Bibliothek BM\_TYPES\_20bd06 (PROPROG wt II) bzw. BM\_TYPES\_30bd01 (ProProg wt III) und höher.

#### HINWEIS



Es müssen alle Eingänge belegt werden, da sonst kein definierter Wert übergeben wird!

### Beschreibung:

Für die Anschaltung an die Prozedur 3964R<sup>®</sup> über Datenbausteine müssen in der Kalt- und in der Warmstart-Task einmalig die Zuordnungen der unterschiedlichen Datenbaustein-Nummern zu den unterschiedlichen Arrays getroffen werden.

Es sind bis zu 20 Zuordnungen möglich, d. h. der Firmware-FB T64\_RSYN kann höchstens 20 mal aufgerufen werden. Mit jedem Aufruf wird die angeschlossene Nummer und das zugeordnete Array in eine system-intern verwaltete Tabelle eingetragen; im zyklischen Programmteil genügt dann der Aufruf des Firmware-FB T64\_REC, mit dem dann über diese interne Tabelle die eigentliche Kommunikation abgewickelt wird.

### Ein-/Ausgang a\_DB\_BMARRAY:

An a\_DB\_BMARRAY wird eine Variable vom Datentyp ARRAY mit der Größe 256 Byte angeschlossen (siehe oben).

### Eingang i\_DB\_NR:

Mit i\_DB\_NR wird dem angeschlossenen Array eine entsprechende Datenbaustein-Nummer zugeordnet.

### Fehlerauswertung:

i_ERR	Fehlerbeschreibung
0	kein Fehler
1	Es wurden zuviele Zuordnungen getroffen (maximal 20 möglich).
2	Fehlender Eintrag von a_DB_BMARRAY.
3	Kein Array-Typ an a_DB_BMARRAY angeschlossen.

### 5.6.15 Funktionsbaustein T64\_REC

Über den Firmware-FB T64\_REC können Daten zwischen der Terminal-Schnittstelle (serielle Schnittstelle) an der b maXX controller PLC und einem weiteren Teilnehmer ausgetauscht werden.

Der FB T64\_REC ist in der FW-Bibliothek „SYSTEM2\_C\_PLC01\_20bd00“ (PROPROG wt II) bzw. „SYSTEM2\_C\_PLC01\_30bd00 / SYSTEM2\_C\_PLC02\_30bd02“ (ProProg wt III) oder höher enthalten.

Als Übertragungsprotokoll wird die Anschaltung an die Prozedur 3964R<sup>®</sup> eingesetzt mit der Vereinfachung, dass der Datentransfer nur über Datenbausteine erfolgen kann und keine Folgetelegramme unterstützt werden.

Die Prozedur 3964R<sup>®</sup> ist ein eingetragenes Warenzeichen der Firma Siemens AG.

Die Terminal-Schnittstelle an der b maXX controller PLC dient dabei immer als Slave-Anschaltung an die Prozedur 3964R<sup>®</sup> auf Datenbausteine.

Für die Anschaltung an die Prozedur 3964R<sup>®</sup> auf Datenbausteine müssen vor der Verwendung von T64\_REC die entsprechenden Zuordnungen zwischen den von der Kom-

munikation verwendeten Datenbaustein-Nummern und den einzelnen Array-Speicherbereichen getroffen worden sein (siehe Firmware-FB T64\_RSYN).

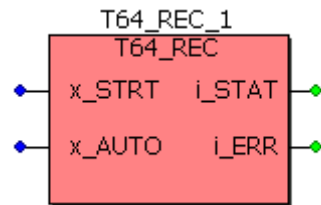


Abbildung 68: Funktionsbaustein T64\_REC

Parameter Eingang	Datentyp	Beschreibung
x_STRT	BOOL	Freigabe für genau einen Kommunikationszyklus bei steigender Flanke
x_AUTO	BOOL	Wenn TRUE erfolgen alle folgenden Kommunikationszyklen automatisch

Parameter Ausgang	Datentyp	Beschreibung
i_STAT	INT	Status-Meldung (-1, 0, 1, 2, 3, 4)
i_ERR	INT	Fehlernummer



#### HINWEIS

Es müssen alle Eingänge belegt werden, da sonst kein definierter Wert übergeben wird!

#### Beschreibung:

Die Kommunikation findet zwischen zwei Teilnehmern statt (Punkt-Zu-Punkt-Verbindung). Der angeschlossene Partner ist dabei immer der Master, von dem der Kommunikationsaufbau ausgeht. Die b maXX controller PLC Anschaltung an die Prozedur 3964R<sup>®</sup> auf Datenbausteine stellt den reagierenden Slave dar.

Für die Übertragung mit RS485-Schnittstellen an der b maXX controller PLC ist eine Vierdrahtleitung (plus Bezugsmasse) Voraussetzung, da die Daten bidirektional übertragen werden müssen. Außerdem ist darauf zu achten, dass die differentiellen Leitungen mit entsprechenden Pull-Up-Widerständen abgeschlossen sind (siehe Firmware-FB TER\_INIT).

Der Firmware-FB T64\_REC sollte in regelmäßigen Zeitabständen (Richtwert:  $\geq 10$  ms) aufgerufen werden. Daher ist es sinnvoll, den Aufruf in einer zyklischen Task zu implementieren oder über eine Timer-Event-Task abzuwickeln, falls die Programmlaufzeit der zyklischen Task zu groß ist.

Für die Initialisierung der Terminal-Schnittstelle werden die Firmware-FBs TER\_INIT und T64\_RSYN in der Kalt- und in der Warmstart-Task eingesetzt (siehe Firmware-FBs TER\_INIT und T64\_RSYN).

### Eingang **x\_STRT**:

Mit einer steigenden Flanke am Eingang **x\_STRT** des Firmware-FB T64\_REC wird die Kommunikation einmalig freigegeben. Die Übertragung zum Partner wird jedoch nach dem ersten Kommunikationszyklus wieder gesperrt. Diese Funktion kann zu Testzwecken benutzt werden, um genau einen Kommunikationszyklus zu bearbeiten.

### Eingang **x\_AUTO**:

Wenn hier TRUE angeschlossen wird, werden auch alle nachfolgenden Kommunikationsaufträge vom Partner bearbeitet. Auch dann, wenn ein Fehler am Ausgang **i\_STAT** (= -1) gemeldet wurde.

Üblicherweise sollte der Anwender nach dem ersten Aufruf vom Firmware-FB T64\_REC diesen Eingang fest auf TRUE setzen, damit alle Kommunikationsanforderungen vom Partner an die Steuerung bearbeitet werden können.

### Ausgang **i\_STAT**:

Nach Start der Kommunikation mit **x\_AUTO** = TRUE und einer steigenden Flanke am Eingang **x\_STRT** wird die Kommunikation freigegeben. Sobald der Ausgang **i\_STAT** = 2 angezeigt, können vom Kommunikations-Partner ankommende Lese- und Schreibaufträge bearbeitet werden. Der Kommunikations-Partner kann dann direkt auf die Array-Inhalte über die zugeordneten Datenbaustein-Nummern zugreifen (siehe Firmware-FB T64\_RSYN).

Wurde ein Auftrag vom Kommunikations-Partner korrekt empfangen, wechselt der Ausgang **i\_STAT** von 2 auf 3 und der Auftrag wird von der b maXX controller PLC bearbeitet. Beim nächsten Aufruf des Firmware-FB T64\_REC wird die Rückantwort zum Kommunikations-Partner gesendet und der Ausgang **i\_STAT** wechselt von 3 auf 4. Nach Abschluss des Sendevorgangs wird wieder selbstständig (also ohne notwendige Flanke am Eingang **x\_STRT**) der Ausgang **i\_STAT** = 2 gesetzt, so dass ein neuer Auftrag vom Kommunikations-Partner an die b maXX controller PLC bearbeitet werden kann. Tritt während der Kommunikation ein Empfangsfehler auf, wird am Ausgang **i\_STAT** der Fehlerzustand "-1" angezeigt.

<b>i_STAT</b>	<b>interner Zustand der Kommunikation der Anschaltung an Prozedur 3964R® auf Datenbausteine</b>
0	Firmware-FB T64_REC ist nicht aktiv oder Initialisierung nicht durchlaufen
1	RECEIVE-Funktion wurde angestoßen
2	Ab jetzt können Aufträge vom Kommunikations-Partner empfangen werden
3	Telegramm erhalten, Auftrag wird ausgeführt; "Antwort zurückschicken" einleiten
4	Warten, bis Antwort abgeschickt: Dann über Zustand 1 in Zustand 2 wechseln
-1	Fehler aufgetreten! Fehlerursache in <b>i_ERR</b> näher spezifiziert

Fehlerauswertung:

Tritt während der Kommunikation ein Empfangsfehler auf, wird am Ausgang i\_STAT der Fehlerzustand "-1" angezeigt, wobei die genaue Fehlerursache am Ausgang i\_ERR angezeigt wird. Beim nächsten Aufruf des Firmware-FB T64\_RSYN wird wieder selbstständig in den Empfangsmodus i\_STAT = 2 gewechselt.

i_ERR	Fehlernummer
0	kein Fehler
1	Reserviert
2	Terminal-Schnittstelle nicht initialisiert (siehe Firmware-FB TER_INIT)
3	BREAK-Fehler Terminal-Schnittstelle
4	FRAME-Fehler Terminal-Schnittstelle
5	PARITY-Fehler Terminal-Schnittstelle
6	OVERRUN-Fehler Terminal-Schnittstelle
7 bis 15	Reserviert
16	max. Anzahl der Kommunikationsversuche erreicht
17	falsche ENDE-Kennung nach Blockübertragung erhalten
18	Zeichenverzugszeit (ZVZ) überschritten
19	Blockverzugszeit (BVZ) überschritten
20	falsches Zeichen als Verbindungszeichen (kein STX)
21 bis 49	Reserviert
50	empfangene Telegrammlänge des Kommunikations-Partners zu kurz (< 10 Zeichen)
51	Auftrag nicht implementiert (weder "SEND" noch "FETCH"), oder vom Kommunikations-Partner angeforderter Quell-/Zielbereich wird nicht unterstützt
52	Zugriffsbereich vom Anwender nicht freigegeben (siehe Firmware-FB T64_RSYN)
ab 53	Reserviert

### 5.6.16 Funktionsbaustein TER\_USS

Mit dem FB TER\_USS wird eine Kommunikation mit Anschaltung an das USS-Protokoll<sup>®</sup> realisiert. Die Verbindung erfolgt über die Terminal-Schnittstelle (serielle RS485 X2-Schnittstelle) der b maXX controller PLC BMC-M-PLC-01.

#### HINWEIS



Software-Anschaltung an USS-Protokoll<sup>®</sup> nur für nur für BMC-M-PLC-01 verfügbar.

Die b maXX controller PLC ist der alleinige Master und kann mit allen Slaves, die das USS-Protokoll<sup>®</sup> verarbeiten können, kommunizieren.



### HINWEIS

Die genaue Funktionsweise, der Telegrammaufbau und die Bedeutung von Parametern des USS-Protokolls<sup>®</sup> ist in der Beschreibung "Baumotronic Kommunikations-Software" nachzulesen.



### HINWEIS

Der FB TER\_USS ist in der Standard-Bibliothek „SYSTEM1\_C\_PLC01\_20bd00“ (PROPROG wt II) bzw. „SYSTEM1\_C\_PLC01\_30bd00“ (ProProg wt III) oder höher enthalten und verwendet den Firmware-FB USS\_SR aus der Firmwarebibliothek SYSTEM2\_C\_PLC01\_20bd00 (PROPROG wt II) bzw. SYSTEM2\_C\_PLC01\_30bd00 (ProProg wt III) oder höher. In dieser Bibliothek befindet sich auch der Firmware-FB TER\_INIT, welcher zur Initialisierung der Terminal-Schnittstelle benötigt wird. Zudem werden Datentypen aus der Bibliothek BM\_TYPES\_20bd06 (PROPROG wt II) bzw. BM\_TYPES\_30bd01 (ProProg wt III) oder höher verwendet.

Das USS-Protokoll<sup>®</sup> ist ein eingetragenes Warenzeichen der Firma Siemens AG.

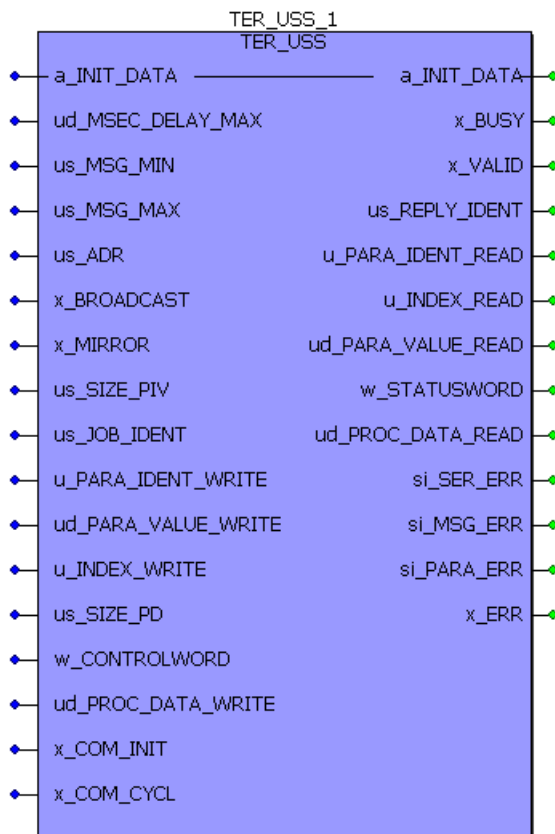


Abbildung 69: Funktionsbaustein TER\_USS



Parameter Eingang	Datentyp	Beschreibung
a_INIT_DATA	UINT_256_BMARRAY	Array mit den Schnittstellenparametern
ud_MSEC_DELAY_MAX	UDINT	maximale Verzugszeit für zu empfangende Zeichen
us_MSG_MIN	USINT	minimale Anzahl der Komm-Versuche
us_MSG_MAX	USINT	maximale Anzahl der Komm-Versuche
us_ADR	USINT 0 bis 31	Adresse des USS-Teilnehmers
x_BROADCAST	BOOL	Broadcast-Telegramm an alle Teilnehmer
x_MIRROR	BOOL	Spiegeltelegramm
us_SIZE_PIV	USINT 0, 3, 4	PKW-Anzahl
us_JOB_IDENT	USINT 0 bis 4	Auftrags-Kennung
u_PARA_IDENT_WRITE	UINT	Parameter-Nummer
ud_PARA_VALUE_WRITE	UDINT	Parameterwert schreiben
u_INDEX_WRITE	UINT	Index-Nummer schreiben
us_SIZE_PD	USINT 0 bis 3	PZD-Anzahl
w_CONTROLWORD	WORD	Steuerwort Prozessdaten
ud_PROC_DATA_WRITE	UDINT	Sollwert Prozessdaten schreiben
x_COM_INIT	BOOL	Startbit für Telegramm (steigende Flanke)
x_COM_CYCL	BOOL	Anschluss für zyklische Datenübertragung

Parameter Ausgang	Datentyp	Beschreibung
a_INIT_DATA	UINT_256_BMARRAY	Array mit den Schnittstellenparametern
x_BUSY	BOOL	Anzeigebit für laufende Kommunikation
x_VALID	BOOL	Anzeigebit für Empfangsdaten gültig
us_REPLY_IDENT	USINT	empfangene Auftrags-Kennung
u_PARA_IDENT_READ	UINT	empfangene Parameter-Nummer
u_INDEX_READ	UINT	Index-Nummer lesen
ud_PARA_VALUE_READ	UDINT	empfangener Parameterwert
w_STATUSWORD	WORD	empfangenes Statuswort Prozessdaten
ud_PROC_DATA_READ	UDINT	empfangener Istwert Prozessdaten
si_SER_ERR	SINT	Fehlernummer serielle Schnittstelle
si_MSG_ERR	SINT	Fehlernummer Telegramm

Parameter Ausgang	Datentyp	Beschreibung
si_PARA_ERR	SINT	Fehlernummer Auftrag
x_ERR	BOOL	globales Fehlerbit



### HINWEIS

Es müssen alle Eingänge belegt werden, da sonst kein definierter Wert übergeben wird!

### Beschreibung:

Der FB TER\_USS sollte zyklisch (Richtwert:  $\geq 10$  ms) aufgerufen werden. Daher ist es sinnvoll, den Aufruf des FB TER\_USS in einer zyklischen Task zu implementieren oder über eine Timer-Event-Task abzuwickeln, falls die Programmlaufzeit der zyklischen Task durch darin befindliche andere Programmteile zu hoch oder zu unregelmäßig ist.

Über die Anbindung an das USS-Protokoll<sup>®</sup> können Bedarfsdaten über den cPKW-Bereich zwischen dem Master (b maXX controller PLC) und der daran angeschlossenen USS-Slaves ausgetauscht werden.

Die Prozessdatenübertragung ("schneller Soll/Istwert-Kanal") erfolgt im USS-Protokoll<sup>®</sup> über den cPZD-Bereich.

An a\_INIT\_DATA muss das gleiche Array (global deklariert) wie am Firmware-FB TER\_INIT angeschlossen werden, da während der Kommunikation auf dieses Array zugegriffen wird.

Bei einer steigenden Flanke am Eingang x\_COM\_INIT wird die USS-Kommunikation gestartet. Sollen automatisch neue Kommunikations-Zyklen angestoßen werden, muss an dem Eingang x\_COM\_CYCL = TRUE eingestellt sein.

Der Ausgang x\_BUSY bleibt solange gesetzt, bis eine Master-Slave-Kommunikation zum Teilnehmer vollständig abgeschlossen ist. Der Abschluss einer Kommunikation wird durch x\_VALID = TRUE angezeigt. Dieser bleibt nach fallender x\_BUSY-Flanke einen Programm-Zyklus lang TRUE.

Am Eingang ud\_MSEC\_DELAY\_MAX wird die maximale Timeoutzeit in ms angeschlossen, bis wann eine Master-Slave-Kommunikation abgeschlossen sein muss.

Wird diese Zeit nach Anstoßen der Kommunikation überschritten, erfolgt eine "Timeout"-Fehlermeldung. Üblicherweise sollte dort ein Wert von ca. 220 ms angeschlossen werden.

Falls kein Fehler während der Kommunikation aufgetreten ist, werden bei x\_VALID = TRUE die Ausgänge mit den empfangenen Werten (w\_STATUSWORD, ud\_PROC\_DATA\_READ, us\_REPLY\_IDENT, u\_PARA\_IDENT\_READ, u\_INDEX\_READ, ud\_PARA\_VALUE\_READ) beschrieben.

Vor der Parameterausgabe erfolgt eine Überprüfung der gesendeten und empfangenen Parameternummern u\_PARA\_IDENT\_WRITE und u\_PARA\_IDENT\_READ.

Dabei wird das Telegramm genau us\_MSG\_MIN-mal angefordert, bis eine Parameterausgabe erfolgt.

Bei einem Fehler (z. B. "Timeout") wird der Telegramm-Aufbau `us_MSG_MAX`-mal versucht, bevor der Ausgang `x_VALID = TRUE` wird und ein Fehler angezeigt wird. Je nachdem, was für ein Fehler aufgetreten ist, können folgende Ausgänge ausgewertet werden:

Parameter- und Empfangs-Fehler, Timeout-Fehler:

Das `x_ERR`-Bit und die Fehlernummern (`si_SER_ERR`, `si_MSG_ERR`, `si_PARA_ERR`) werden ausgegeben, die anderen Ausgänge behalten ihren vorherigen Instanzwert!

Auftrags-Fehler:

Das Bit `x_ERR`, `si_SER_ERR`, `si_MSG_ERR`-Zustand, die Auftrags-Fehlernummer `si_PARA_ERR`, `us_REPLY_IDENT`, `u_PARA_IDENT_READ` werden neu beschrieben, `u_INDEX_READ` und `ud_PARA_VALUE_READ` behalten ihren vorherigen Instanzwert!

Die Prozessdaten (`w_STATUSWORD` und `ud_PROC_DATA_READ`) werden bei jedem geglückten Telegramm-Aufbau ausgegeben, auch wenn ein sonstiger Parameter-Auftragsfehler aufgetreten ist. Die Prozessdaten-Ausgabe ist deswegen von `us_MSG_MIN` unabhängig!

Die Eingänge `us_MSG_MIN` und `us_MSG_MAX` legen fest, wie oft eine Kommunikation mindestens (`us_MSG_MIN`) und wie oft höchstens (`us_MSG_MAX`, z. B. bei Timeout-Meldung) stattfinden darf, bevor eine entsprechende Reaktion am Ausgang durch die vom Slave zurückgelieferten Daten oder eine Fehlermeldung erfolgt.

Normalerweise sollte der Anwender folgende Initialisierungswerte verwenden:

```
us_MSG_MIN = 1;
us_MSG_MAX = 2;
```

Die Organisation des Inhalts im Telegrammaufbau des USS-Protokoll<sup>®</sup> wird über die PKW- und PZD-Anzahl festgelegt. Diese Einstellungen müssen bei allen Teilnehmern im USS-Ring identisch sein, damit der Inhalt des ausgetauschten Telegramms richtig interpretiert werden kann.

<code>us_SIZE_PIV</code>	<b>PKW-Anzahl: Aufteilung der Bedarfsdaten im USS-Protokoll</b>
4	Doppelwort-Parameter (32 Bit-Typen)
3	Wort-Parameter (16 Bit-Typen)
0	Telegramm enthält keine Bedarfsdaten!

<code>us_SIZE_PD</code>	<b>PZD-Anzahl: Aufteilung der Prozessdaten im USS-Protokoll</b>
3	Steuer/Statuswort und Doppelwort-Soll/Istwerte (32 Bit-Typen)
2	Steuer/Statuswort und Wort-Soll/Istwerte (16 Bit-Typen)
1	nur Steuer/Statuswort
0	Telegramm enthält keine Prozess-Daten!

<b>us_JOB_IDENT</b>	<b>Auftragskennung (Master -&gt; Slave)</b>
0	kein Auftrag
1	Parameterwert lesen (Antwort in ud_PARA_VALUE_READ)
2	Parameterwert ud_PARA_VALUE_WRITE schreiben vom Typ Wort (16 Bit)
3	Parameterwert ud_PARA_VALUE_WRITE schreiben vom Typ Doppelwort (32 Bit)
4	Ein Element aus der Parameter-Beschreibung lesen (welches, steht in u_INDEX_WRITE)
5 bis ...	(reserviert)

<b>us_REPLY_IDENT</b>	<b>Antwortkennung (Slave -&gt; Master)</b>
0	kein Auftrag
1	Parameter-Wort (16 Bit) wurde geschrieben (us_JOB_IDENT = 2) bzw. gelesen (us_JOB_IDENT = 1) (in ud_PARA_VALUE_READ steht der übertragene Wert)
2	Parameter-Doppelwort (32 Bit) wurde geschrieben (us_JOB_IDENT = 3) bzw. gelesen (us_JOB_IDENT = 1) (in ud_PARA_VALUE_READ steht der übertragene Wert)
3	Element aus der Parameter-Beschreibung wurde gelesen (Antwort steht in ud_PARA_VALUE_READ)
4 bis 6	(reserviert)
7	Parameter-Fehler: Auftrag nicht ausführbar! (Fehlerkennung siehe si_PARA_ERR)
8 bis ...	(reserviert)

### Fehlerauswertung:

<b>si_SER_ERR</b>	<b>Fehler serielle Schnittstelle</b>
0	kein Fehler
1	(reserviert)
2	Falsches Serial-Initialisierungsarray a_INIT_DATA angeschlossen
3	Schnittstelle nicht initialisiert
4 bis 9	(reserviert)
10	Break beim Empfangen
11	Frame-Fehler beim Empfangen
12	Parity-Fehler beim Empfangen
13	Overrun-Fehler beim Empfangen

si_MSG_ERR	Telegramm-Fehler beim Senden/Empfangen
0	kein Fehler
1	Der am FB angeschlossene us_SIZE_PIV-Bereich wird nicht unterstützt
2	Der am FB angeschlossene us_SIZE_PD-Bereich wird nicht unterstützt
3	us_MSG_MIN > us_MSG_MAX oder us_MSG_MIN bzw. us_MSG_MAX = 0
4	an us_ADR ungültige Adresse angeschlossen (nur 0..31 zulässig)
5	gesendete und empfangene Adresse unterschiedlich
6	gesendete und empfangene Parameternummer u_PARA_IDENT_WRITE / u_PARA_IDENT_READ unterschiedlich!
7	us_SIZE_PIV-Bereich falsch
8	us_SIZE_PD-Bereich falsch
9	STX-Zeichen nicht empfangen!
10	empfangene und erwartete Längenangabe (us_SIZE_PIV/us_SIZE_PD) unterschiedlich!
11	BCC-Checksummen-Fehler
12	Timeout-Fehler beim Empfangen

si_PARA_ERR	Parameter-Fehler
0	kein Fehler
-1	Parameter-Nummer wird nicht unterstützt (unzulässige u_PARA_IDENT_WRITE)
1	Parameter nicht änderbar
2	MIN/MAX-Begrenzung
3	fehlerhafter Index u_INDEX_WRITE
4	kein Array-Typ
5	falscher Datentyp
6	kein Setzen erlaubt
7	Beschreibungselement nicht änderbar
8 bis 100	(reserviert)
101	Unbestimmter Fehler
102	Dienst nicht implementiert
103	Parameterformat zu groß für PKW-Bereich
104	PBE-Element nicht vorhanden

## 5.7 Codelaufzeiten

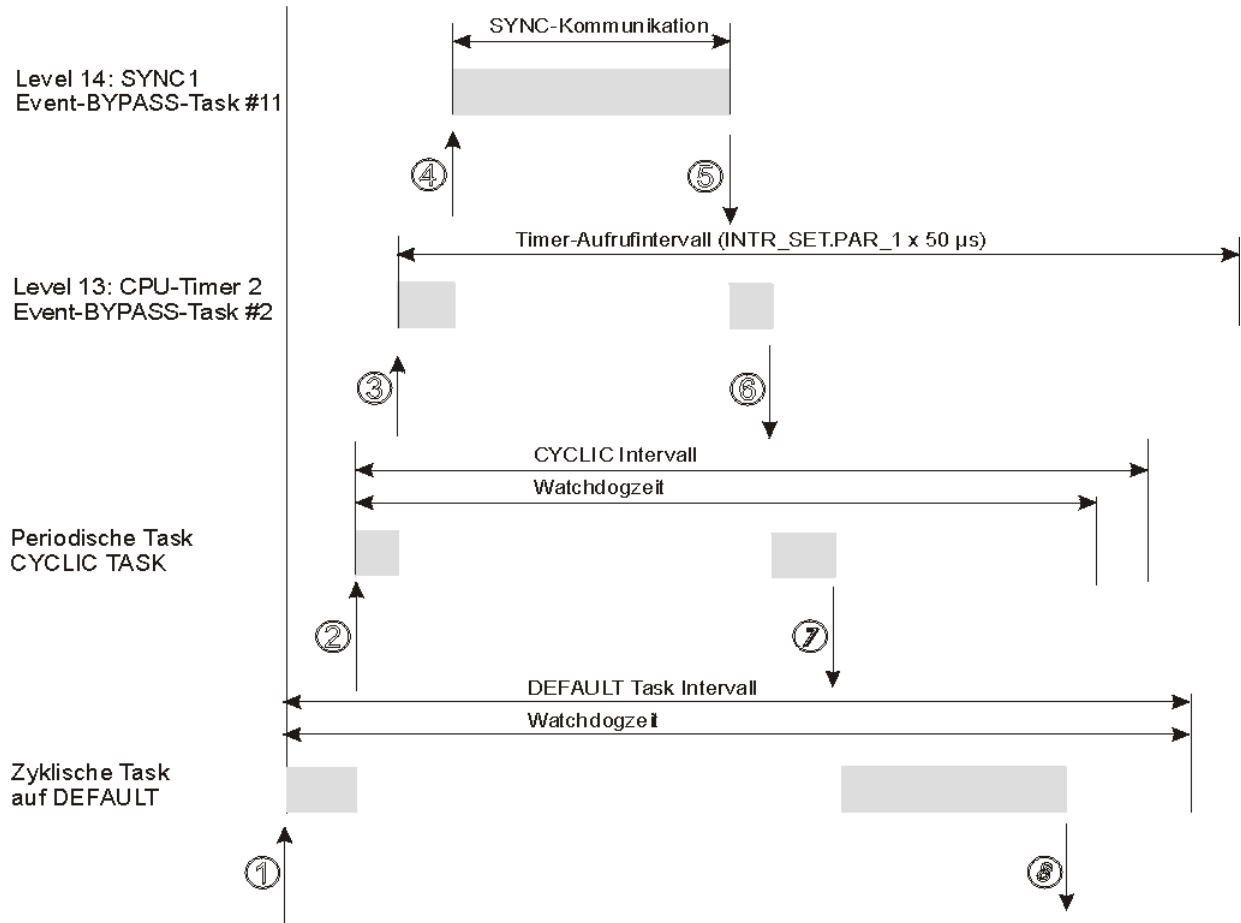


Abbildung 70: Aufteilung von Codelaufzeiten auf die Ressource BMC\_M\_PLC01



## HINWEIS

Alle Events müssen als Tasks mit der Eigenschaft "BYPASS" angelegt werden.

Eine höherpriorige Event-Task unterbricht eine niederpriorige Event-Task, daher sollten Timer-Event-Tasks, die zeitgleich zu synchronen Bussystemen laufen könnten, immer mit dem niedrigeren Level 13 eingesetzt werden, damit sie unterbrochen werden können.

Bei zyklischen Tasks, die keine BYPASS-Tasks sind, muss in den "Einstellungen" der Task die zyklische Abtastzeit (= Einstellung "Intervall") und eine Überwachungszeit (= Einstellung "Watchdogzeit") vom Anwender angegeben werden. Diese zyklischen Tasks sowie die Default-Task werden hinsichtlich der Watchdogzeit während des Betriebes durch das Laufzeitsystem überwacht.

Dabei ist darauf zu achten, dass die eingestellte Watchdogzeit nicht nur die reine Laufzeit dieser Task selbst berücksichtigt, sondern auch die Summe der maximalen Einzel-Laufzeiten enthält, die diese Task unterbrechen können.

Ist die Überwachungszeit zu klein eingestellt, wird der SPS-Fehler "Watchdog in Task 'x' überschritten!" zurückgemeldet, um den Anwender auf die aufgetretene System-Überlast hinzuweisen, und die Steuerung geht gleichzeitig in den sicheren Zustand "STOP".

Die Watchdog-Zeiten in BYPASS-Tasks werden ignoriert.

### Weitere Einfluss-Faktoren auf die Watchdog-Zeiten in den zyklischen Nicht-Bypass-Tasks und der Default-Task:

Für die Überwachungszeiten sollte man zusätzlich eine gewisse Reserve einkalkulieren für sonstige im Hintergrund ablaufende Systemfunktionen, wie z. B. Task-Aufrufschalen sowie Funktionen, die für die Online-Darstellungen von Variablen in den Watch-Listen und/oder in den globalen Variablen-Arbeitsblättern aufgerufen werden und die Systemlast erhöhen.

Außerdem können bestimmte Überwachungsfunktionen vom Anwender eingeschaltet sein, die während der Laufzeit aktiv sind:

Dazu gehören die Einstellungen in der Ressource für "Index-Prüfung auf SPS" und "Stack Prüfung auf SPS". Diese Einstellungen können eventuell nach Abschluss einer Inbetriebnahme wieder zurückgesetzt werden, um Laufzeit-Ressourcen zu sparen.

Für die BYPASS-Tasks lässt sich die tatsächliche Systembelastung während des Betriebes unter einer "worst case"-Betrachtung durch Einsatz der "Time\_Measure"- Bausteine aus der Bibliothek "SYSTEM1\_C\_PLC01\_20bd00" (PROPROG wt II) bzw. „SYSTEM1\_C\_PLC01\_30bd00“ / „SYSTEM1\_C\_PLC02\_30bd02“ (ProProg wt III) oder höher exakt ermitteln. Mit den FBs lassen sich Laufzeiten von max. 10 ms mit einer sehr hohen Auflösung erfassen.

Wenn in den niederpriorigen zyklischen Tasks größere Zeitabstände ausgemessen werden müssen, kann in einem BYPASS-Interrupt, der mit einer festen Zeitbasis läuft, ein kontinuierlicher globaler Zählwert gebildet werden, der dann direkt in den zyklischen Tasks über Differenzbildung des Zählwertes ausgewertet werden kann. (Wie man einen zyklischen CPU-Timer als BYPASS-Task einrichtet, siehe Beschreibung zu FB "INTR\_SET" in der FW-Lib "SYSTEM2\_C\_PLC01\_20bd00" (PROPROG wt II) bzw. „SYSTEM2\_C\_PLC01\_30bd00“ / „SYSTEM2\_C\_PLC02\_30bd02“ (ProProg wt III) ).

Ein BYPASS-Event kann über den FB "INTR\_SET" der FW-Bibliothek "SYSTEM2\_C\_PLC01\_20bd00" (PROPROG wt II) bzw. „SYSTEM2\_C\_PLC01\_30bd00“ / „SYSTEM2\_C\_PLC02\_30bd02“ (ProProg wt III) oder höher mit x\_EN = FALSE auch wieder während der Laufzeit gesperrt werden.

### Vermeidung einer System-Überlastung:

Bei Watchdog-Überschreitungen müssen die Einstellungen (Intervall bzw. Watchdogzeit) nach oben gesetzt werden und/oder die Laufzeiten in den BYPASS-Interrupts minimiert werden, entweder durch Programm-Aufteilung im Interrupt selbst (= abwechselnde Programmteil-Abläufe je Interrupt) oder durch Verlagerung in niederpriorie Nicht-BYPASS-Tasks mit größeren Aufruf-Intervallen bzw. durch Platzierung von Programmteilen direkt in die Default-Task.

Soll in zyklischen Nicht-Bypass-Tasks keine Laufzeit-Überwachung stattfinden, kann auch eine Watchdogzeit eingestellt werden, die größer als das für diese Task eingestellte Intervall ist.

### 5.7.1 Funktionsbaustein TIME\_MEASURE\_START

Diesen Funktionsbaustein für TIME\_MEASURE können Sie zusammen mit dem Funktionsbaustein TIME\_MEASURE\_END verwenden, um Code-Laufzeiten und Aufrufzeiten für Tasks zu ermitteln.

Der Funktionsbaustein TIME\_MEASURE\_START ist in der Standard-Bibliothek SYSTEM1\_C\_PLC01\_20bd00 (PROPROG wt II) bzw. SYSTEM1\_C\_PLC01\_30bd00 / SYSTEM1\_C\_PLC02\_30bd02 (ProProg wt III) oder höher enthalten.

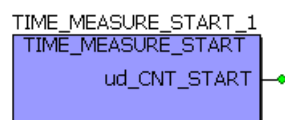


Abbildung 71: Funktionsbaustein TIME\_MEASURE\_START

Ausgangsparameter	Datentyp	Beschreibung
ud_CNT_START	UDINT	aktueller Systemtimer-Wert in Systemeinheiten

### Beschreibung

Der FB TIME\_MEASURE\_START wird zusammen mit dem FB TIME\_MEASURE\_END für Zeitmessungen auf der b maXX controller PLC eingesetzt. Ermittelt wird die Laufzeit des Programmcodes, der sich zwischen dem Aufruf des FB TIME\_MEASURE\_START (Beginn der Zeitmessung) und dem Aufruf des FB TIME\_MEASURE\_END (Ende der Zeitmessung) befindet.

Der FB TIME\_MEASURE\_START liest von einem Systemtimer den aktuellen Wert in Systemeinheiten und gibt diesen an ud\_CNT\_START aus. Der FB



TIME\_MEASURE\_END liest wiederum den aktuellen Systemtimer-Wert in Systemeinheiten und bildet die Differenz zum vom FB TIME\_MEASURE\_START gelesenen Wert. An beiden FBs muss daher an ud\_CNT\_START die gleiche Variable angeschlossen werden.



#### HINWEIS

Eine korrekte Zeitmessung ist nur möglich, wenn die beiden FBs in einer BYPASS-Event-Task mit dem Level 14 eingesetzt werden. Es können Laufzeiten bis max. 10 ms (BMC-M-PLC-01) bzw. 15 min (BMC-M-PLC-02) ermittelt werden. Bei Programmcode, der mehr als 10 ms (BMC-M-PLC-01) bzw. 15 min (BMC-M-PLC-02) Laufzeit hat, werden keine korrekten Werte geliefert. Werden die Zeitmess-Bausteine in einer BYPASS-Event-Task mit dem Level 13 eingesetzt und von einem BYPASS-Event mit dem Level 14 unterbrochen, wird diese Task komplett mitgemessen!

### 5.7.2 Funktionsbaustein TIME\_MEASURE\_END

Diesen Funktionsbaustein für TIME\_MEASURE können Sie zusammen mit dem Funktionsbaustein TIME\_MEASURE\_START verwenden, um Code-Laufzeiten und Aufrufzeiten für Tasks zu ermitteln.

Der Funktionsbaustein TIME\_MEASURE\_END ist in der Standard-Bibliothek SYSTEM1\_C\_PLC01\_20bd00 (PROPROG wt II) bzw. SYSTEM1\_C\_PLC01\_30bd00 / SYSTEM1\_C\_PLC02\_30bd02 (ProProg wt III) oder höher enthalten.

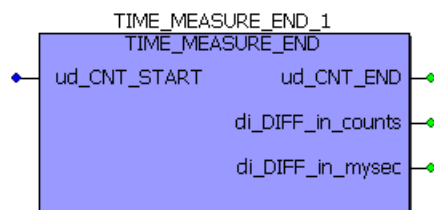


Abbildung 72: Funktionsbaustein TIME\_MEASURE\_END

Eingangsparameter	Datentyp	Beschreibung
ud_CNT_START	UDINT	Systemtimer-Wert von FB TIME_MEASURE_START

Ausgangsparameter	Datentyp	Beschreibung
ud_CNT_END	UDINT	aktueller Systemtimer- Wert in Systemeinheiten
di_DIFF_in_counts	DINT	Differenz zwischen ud_CNT_END und ud_CNT_START
di_DIFF_in_mysec	DINT	Zeitdifferenz in ms

### Beschreibung

Der FB TIME\_MEASURE\_START wird zusammen mit dem FB TIME\_MEASURE\_END für Zeitmessungen auf der b maXX controller PLC eingesetzt. Ermittelt wird die Laufzeit des Programmcodes, der sich zwischen dem Aufruf des FB TIME\_MEASURE\_START (Beginn der Zeitmessung) und dem Aufruf des FB TIME\_MEASURE\_END (Ende der Zeitmessung) befindet.

Der FB TIME\_MEASURE\_START liest von einem Systemtimer den aktuellen Wert in Systemeinheiten und gibt diesen an ud\_CNT\_START aus. Der FB TIME\_MEASURE\_END liest wiederum den aktuellen Systemtimer-Wert in Systemeinheiten und bildet die Differenz zum vom FB TIME\_MEASURE\_START gelesenen Wert. An beiden FBs muss daher an ud\_CNT\_START die gleiche Variable angeschlossen werden.

Die Zeitdifferenz wird an di\_DIFF\_in\_mysec ausgegeben, an di\_DIFF\_in\_counts steht die Differenz der Systemtimer-Werte in Systemeinheiten zur Verfügung. Der vom FB TIME\_MEASURE\_END ausgelesene Systemtimer-Wert kann an ud\_CNT\_END abgelesen werden.

Um die Aufrufzeit einer Task zu ermitteln, werden die beiden FBs an den Beginn einer POE gesetzt, wobei der FB TIME\_MEASURE\_END vor dem FB TIME\_MEASURE\_START zu platzieren ist. Die Beschaltung mit Variablen erfolgt genauso wie bei einer Code-Laufzeitmessung. An di\_DIFF\_in\_mysec kann dann die Aufrufzeit abgelesen werden.



### HINWEIS

Eine korrekte Zeitmessung ist nur möglich, wenn die beiden FBs in einer BYPASS-Event-Task mit dem Level 14 eingesetzt werden. Es können Laufzeiten bis max. 10 ms (BMC-M-PLC-01) bzw. 15 min (BMC-M-PLC-02) ermittelt werden. Bei Programmcode, der mehr als 10 ms (BMC-M-PLC-01) bzw. 15 min (BMC-M-PLC-02) Laufzeit hat, werden keine korrekten Werte geliefert. Werden die Zeitmess-Bausteine in einer BYPASS-Event-Task mit dem Level 13 eingesetzt und von einem BYPASS-Event mit dem Level 14 unterbrochen, wird diese Task komplett mitgemessen!

## 5.8 Praktische Hinweise

### 5.8.1 SPS-Fehler allgemein

Eine Liste der SPS-Fehler finden Sie in der Online-Hilfe von PROPROG wt unter:  
? \ "Hilfe zur SPS" → Register "Inhalt" → Punkt "SPS-spezifische Hilfe" - "Fehlerkatalog".

### 5.8.2 SPS-Fehler "Watchdog in Task 'x' überschritten!"

Die SPS-Fehlermeldung ist ein vom Laufzeitsystem generierter Fehler, um den Anwender auf folgende System-Überlastung hinzuweisen:

Bei zyklischen Tasks, die keine BYPASS-Tasks sind, muss in den "Einstellungen" der Task die zyklische Abtastzeit (= Einstellung "Intervall") und eine Überwachungszeit

(= Einstellung "Watchdogzeit") vom Anwender angegeben werden. Diese Tasks werden während des Betriebes durch das Laufzeitsystem überwacht. Wenn nun eine Verletzung der gewählten Einstellungen durch das Laufzeitsystem festgestellt wird, geht die SPS in den Zustand "Stop" und meldet den Fehler "Watchdog in Task 'x' überschritten!" zurück.

#### **Mögliche Gründe für eine Verletzung:**

Werden BYPASS-Tasks vom Anwender eingesetzt, die sehr hoch ausgelastet sind (z. B. reine Interrupt-Laufzeit von 1,5 ms bei 2 ms Abtastzeit) und sich womöglich noch gegenseitig unterbrechen (z. B. eine weitere BYPASS-Timer-Task mit 300 ms reiner Laufzeit), stellt dies eine zusätzliche Belastung der CPU-Rechenzeit dar, die unter einer "worst case"-Laufzeitbetrachtung zu diesen oben genannten SPS-Fehlern in einer niedrigeren, vom Betriebssystem überwachten zyklischen Task (= Nicht-BYPASS-Task) führen kann.

#### **Womit man "rechnen" sollte:**

Bei der Einrichtung der Überwachungszeit muss man also nicht nur die reine Laufzeit der einzurichtenden Task alleine berücksichtigen, sondern auch die Maximal-Laufzeiten von höherpriorigen Tasks und BYPASS-Tasks hinzuaddieren sowie eine gewisse Reserve berücksichtigen für sonstige ablaufende Systemfunktionen, z. B. für die im Hintergrund ablaufenden Task-Aufrufschalen sowie den Online-Darstellungen von Variablen in den Watch-Listen und/oder in den globalen Variablen-Arbeitsblättern. Auch sollte man nicht vergessen, dass bestimmte Überwachungsfunktionen vom Anwender eingeschaltet sein können, die während der Laufzeit aktiv sind (z. B. Die Einstellungen in der Ressource für "Index-Prüfung auf SPS" oder "Stack Prüfung auf SPS") und die eventuell nach Abschluss einer Inbetriebnahme wieder zurückgesetzt werden können, um Ressourcen zu sparen.

Für die BYPASS-Tasks lässt sich die tatsächliche Systembelastung während des Betriebes unter einer "worst case"-Betrachtung durch Einsatz der "Time\_Measure"- Bausteine aus der Bibliothek "SYSTEM1\_C\_PLC01\_20bd00" (PROPROG wt II) bzw. SYSTEM1\_C\_PLC01\_30bd00 / SYSTEM1\_C\_PLC02\_30bd02 (ProProg wt III) oder höher exakt ermitteln. Mit den FBs lassen sich Laufzeiten von max. 10 ms mit einer sehr hohen Auflösung erfassen.

Wenn in den niedrigeren zyklischen Tasks größere Zeitabstände ausgemessen werden müssen, kann in einem BYPASS-Interrupt, der mit einer festen Zeitbasis läuft, ein kontinuierlicher globaler Zählwert gebildet werden, der dann direkt in den zyklischen Tasks über Differenzbildung des Zählwertes ausgewertet werden kann. (Wie man einen zyklischen CPU-Timer als BYPASS-Task einrichtet, siehe Beschreibung zu FB "INTR\_SET" in der FW-Lib "SYSTEM2\_C\_PLC01\_20bd00" (PROPROG wt II) bzw. SYSTEM2\_C\_PLC01\_30bd00 / SYSTEM2\_C\_PLC02\_30bd02 (ProProg wt III) ).

#### **Vermeidung einer Überlastung:**

Bei Watchdog-Überschreitungen müssen entweder die Einstellungen (Intervall bzw. Watchdogzeit) nach oben gesetzt werden und/oder die Laufzeiten in den BYPASS-Interrupts minimiert werden, entweder durch Programm-Aufteilung im Interrupt selbst (= abwechselnde Programmteil-Abläufe je Interrupt) oder durch Verlagerung in niedrigeren Nicht-BYPASS-Tasks mit größeren Aufruf-Intervallen bzw. durch Platzierung von Programmteilen direkt in die Default-Task.

### 5.8.3 SPS-Fehler "Globale Bereitmeldung (RST-Signal) fehlt! PLC gestoppt!"

---

Das Modul b maXX controller PLC erhielt nicht die globale Bereitmeldung folgender Module.

- Modul Netzteil für b maXX controller PLC
- Module die links der b maXX controller PLC gesteckt sind (z. B. Modul Ethernet mit CANopen-Master)).

Der Grund dafür kann sein:

- Es liegt ein Defekt oder ein Anschlussfehler eines Moduls vor.

→ Tritt der Fehler auch dann auf, wenn nur die b maXX controller PLC und ihr Netzteil verwendet werden?

Die Montage und Demontage von Modulen darf nur im spannungslosen Zustand erfolgen. Zur Montage und Demontage siehe die jeweilige Betriebsanleitung der Module.

Ja: Überprüfen Sie die Kontakte zwischen b maXX controller PLC und Netzteil auf umgebogene Pins.

Tauschen Sie gegebenenfalls die b maXX controller PLC oder das Netzteil.

Nein: Montieren Sie nacheinander die anderen Module und deren Anschlüsse bis sie das fehlerhafte Modul ermitteln können.

### 5.8.4 SPS-Fehler "Kein Zugriff auf den CX-Controller im Netzteil-Modul möglich!"

---

Die b maXX PLC kann nicht mit dem Netzteil kommunizieren.

Der Grund dafür kann sein:

- Es liegt ein Defekt oder ein Anschlussfehler vor.

→ Überprüfen Sie die Kontakte zwischen b maXX controller PLC und Netzteil auf umgebogene Pins.

Tauschen Sie gegebenenfalls das Netzteil oder die b maXX controller PLC, falls der Fehler weiterhin erzeugt wird.

### 5.8.5 SPS-Fehler "I/O-Bus Fehler aufgetreten! Fehler-Code:..."

---

Der Fehlercode wird als 0xAABB angegeben. Die Bedeutung von AA und BB entnehmen Sie bitte der Tabelle:

Fehlercode		Bedeutung	Abhilfe
AA	BB		
01	00	I/O-Bus: Mapping Fehler (falsche Offsets 0xFDC/0xFDE)	Die Module einer Gruppe (Digital Input, Digital Output, Analog In-/Output) nutzen mehr als 512 Byte im zugehörigen Datenbereich (siehe Applikationshandbuch b maXX controller PLC)
03	00	I/O-Bus: Kommandofehler	Ein I/O-Modul reagiert fehlerhaft auf ein I/O-Bus Kommando: <b>1</b> Das I/O-Modul ist defekt. Tauschen Sie das I/O-Modul aus. <b>2</b> Es ist kein I/O-Modul gesteckt. Stecken Sie ein I/O-Modul <b>3</b> Es ist nur das End-Modul gesteckt. Stecken Sie ein I/O-Modul
04	n	I/O-Bus: Bruchstelle hinter I/O-Modul n	Kontrollieren Sie das I/O-Modul hinter dem n-ten I/O-Modul und tauschen Sie es gegebenenfalls
05	n	I/O-Bus: Fehler bei Kommunikation mit I/O-Modul n	Kontrollieren Sie das n-te I/O-Modul und tauschen Sie es gegebenenfalls
14	n	I/O-Bus: Bus-Störung	Kontrollieren Sie das n-te I/O-Modul und tauschen Sie es gegebenenfalls
15	n	I/O-Bus: Bus-Störung	Kontrollieren Sie das n-te I/O-Modul und tauschen Sie es gegebenenfalls



# MODULE (RECHTS DER b maXX CONTROLLER PLC) VERWENDEN

Die IO Module rechts der b maXX controller PLC können sowohl in ProProg wt III als auch in PROPROG wt II automatisch konfiguriert werden.

In ProProg wt III steht dafür das Engineering Framework [►ProMaster◄](#) ab Seite 135 zur Verfügung.

In PROPROG wt II steht dafür der [►Modul Konfigurator ProModul◄](#) ab Seite 150 zur Verfügung.

## 6.1 ProMaster

---

Mit dem Engineering Framework ProMaster werden Systemkomponenten/Module, die links und rechts der b maXX controller PLC stecken, konfiguriert.

Nachfolgend wird beschrieben, wie die Module, die rechts der b maXX controller PLC stecken, konfiguriert werden.

ProMaster legt für die Module im ProProg wt III Projekt-Kommunikationsvariablen an, über die die Eingänge dieser Module gelesen bzw. die Ausgänge dieser Module geschrieben werden können.

---

### HINWEIS



Die Verwendung von ProMaster setzt ProProg wt III voraus.

---

### Durchzuführende Schritte

Voraussetzung	Die Montage, der Anschluss und die Inbetriebnahme von <ul style="list-style-type: none"><li>• b maXX controller PLC</li><li>• Netzteil für b maXX controller PLC</li><li>• Systemkomponenten die links der b maXX controller PLC gesteckt sind (z. B. CANopen-Master)</li></ul>
---------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

- Systemkomponenten die rechts der b maXX controller PLC gesteckt sind (z. B. digitale und/oder analoge I/O-Module, Endmodul)

müssen erfolgreich abgeschlossen sein.

Damit über die b maXX controller PLC die Systemkomponenten angesprochen und Daten ausgetauscht werden können, müssen folgende Schritte abgearbeitet werden.

- Anlegen eines ProProg wt III Projekts für die b maXX controller PLC
- Anlegen eines ProMaster Projekts mit b maXX controller PLC
- Konfigurierung des b maXX Systems mit ProMaster und Verbinden des ProMaster Projekts mit dem ProProg wt III Projekt
- Konfigurierung der IOs mit ProMaster

**Ergebnis:** Variablenliste für die Initialisierung und/oder den Betrieb dieser Systemkomponenten. Anschließend erfolgt die Programmierung des Applikationsteils im ProProg wt III Projekt für die b maXX controller PLC. Anleitungen für die Programmierung der Systemkomponenten, die rechts der b maXX controller PLC gesteckt sind, entnehmen Sie bitte den jeweiligen Applikationshandbüchern.

**Beispiel:** Dieses Kapitel wird von einem Beispiel für b maXX controller PLC (BMC-M-PLC-02) begleitet. Im Beispiel erstellen wir eine Konfiguration mit (von links nach rechts) folgenden Systemkomponenten/Module rechts der b maXX controller PLC:

Netzteil für b maXX controller PLC

- 8 digitale Ausgänge +24V DC (DO8000)
- 8 digitale Eingänge +24V DC (DI8000)
- 2 analoge Ausgänge 0-10V (AO2010)
- 2 analoge Eingänge 0-10V (AI2010)
- Endmodul (EK0000)

### 6.1.1 Anlegen eines ProProg wt III Projekts für die b maXX controller PLC

---

#### 6.1.1.1 Allgemein

---

Falls Sie im Abschnitt [►b maXX controller PLC Projekt◄](#) ab Seite 35 noch kein eigenes Projekt für Ihre Applikation angelegt haben, erstellen sie das Projekt "Example\_C\_PLC02" bitte mit der Vorlage "BMC\_M\_PLC02". Sie benötigen dazu ein ProProg wt III ab Version 1.1 Build 78 und ProMaster ab Version 1.1.5.0. Die Versionsnummer von ProProg wt III finden Sie auf der Hülle der Installations CD von ProProg wt III oder in ProProg wt III selbst im Menüpunkt ? \ Info.

Prüfen Sie ob die Bibliotheken

- BM\_TYPES\_30bd01 (oder höher)
- SYSTEM2\_C\_PLC02\_30bd02 (oder höher)

sowie gegebenenfalls die jeweiligen Bibliotheken für Ihre Systemkomponenten in Ihrem ProProg wt III Projekt vorhanden sind.

Sollte dies nicht der Fall sein, binden Sie diese Bibliotheken bitte in Ihr Projekt ein. Sie enthalten wichtigen Datentypen und Funktionsbausteine für Ihre Systemkomponenten.





## HINWEIS

Sie können auch ein Projekt "Example\_C\_PLC01" mit der Ressource BMC\_M\_PLC01 anlegen. In diesem Fall verwenden Sie das Template-Projekt BMC\_M\_PLC01 und die Bibliotheken BM\_TYPES\_30bd01 (oder höher) und SYSTEM2\_C\_PLC01\_30bd00 (oder höher)

### 6.1.1.2 Beispiel: Anlegen des Projekts "Example\_C\_PLC02"

Es wurde das Projekt "Example\_C\_PLC02" mit der Vorlage "BMC\_M\_PLC02" angelegt. Die Bibliotheken

- BM\_TYPES\_30bd01
- SYSTEM2\_C\_PLC02\_30bd02

sind eingebunden.

Im Beispiel werden keine Systemkomponenten/Module, die links der b maXX controller PLC gesteckt sind, verwendet, so dass auch keine entsprechenden Bibliotheken eingebunden werden müssen.

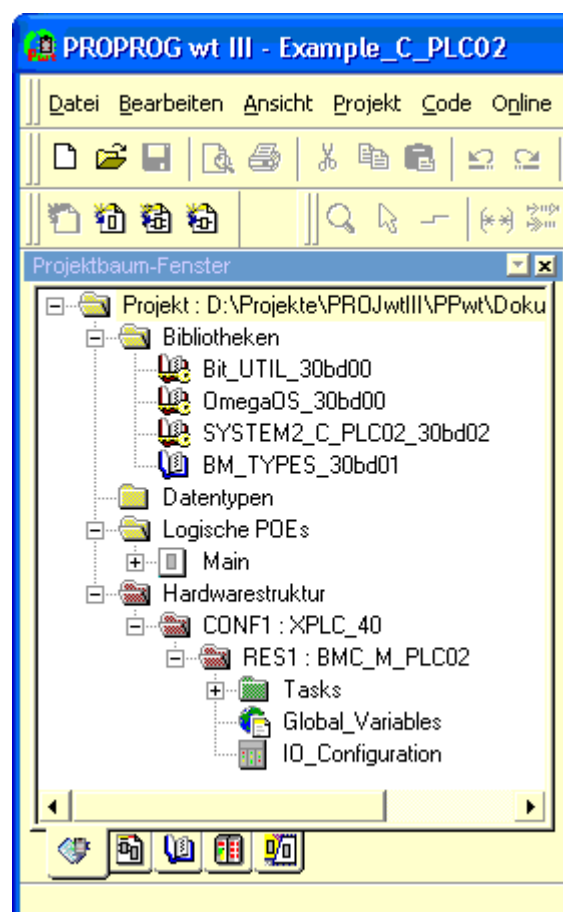


Abbildung 73: Beispiel: Anlegen des Projekts "Example\_C\_PLC02"

### 6.1.2 Anlegen eines ProMaster Projektes mit b maXX controller PLC

In diesem Abschnitt wird ein ProMaster Projekt mit b maXX controller PLC angelegt  
Öffnen Sie ProMaster:

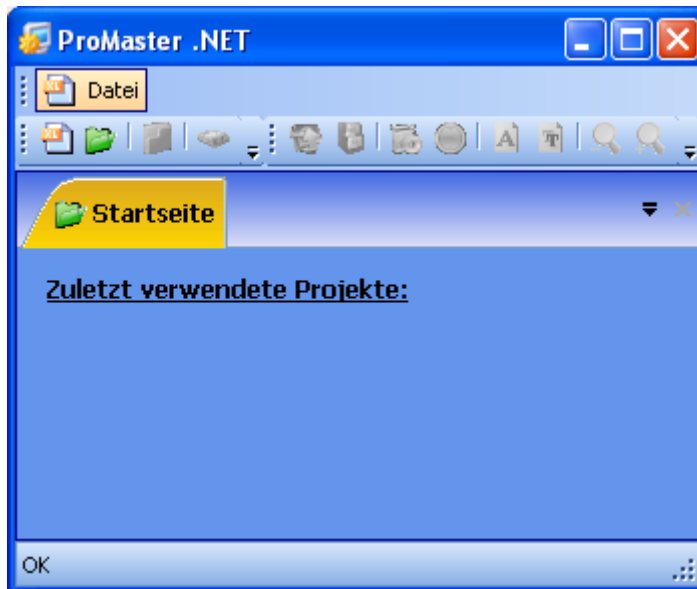


Abbildung 74: ProMaster ohne Projekte

Legen Sie ein neues ProMaster-Projekt an, indem Sie über Menü "Datei\NeuesProjekt" das Fenster "Projekt Einstellungen" öffnen.

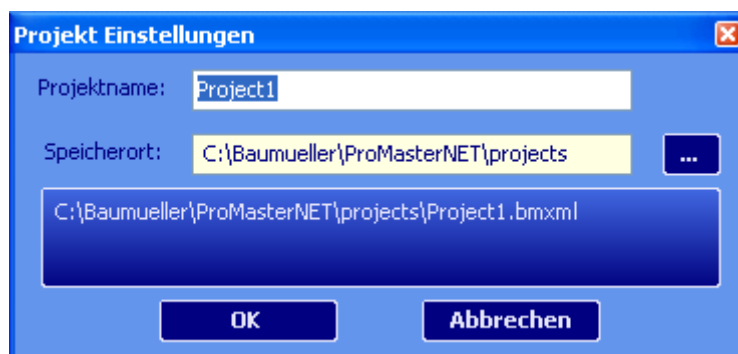


Abbildung 75: ProMaster - Projektname

Geben Sie den Namen (in der Edit-Box) und den Speicherort (über die Pfadauswahl) des Projektes an. Für unser Beispielprojekt verwenden wir den Namen "Example\_C\_PLC02.bmxml".

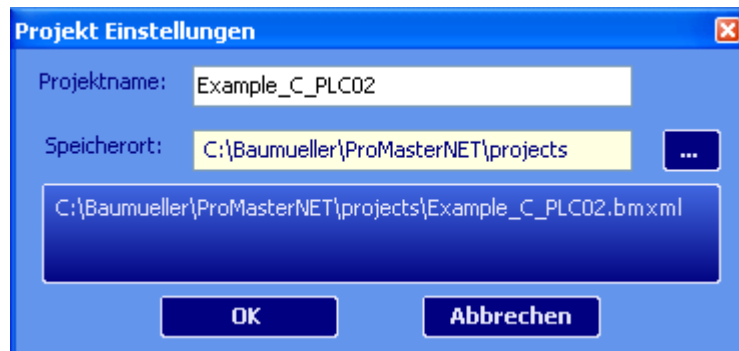


Abbildung 76: ProMaster - Projektname vergeben

Übernehmen Sie die Einstellungen aus dem Fenster "Projekt Einstellungen" durch Klicken auf den Button "OK". ProMaster wechselt jetzt in die Feldbus-Ansicht.

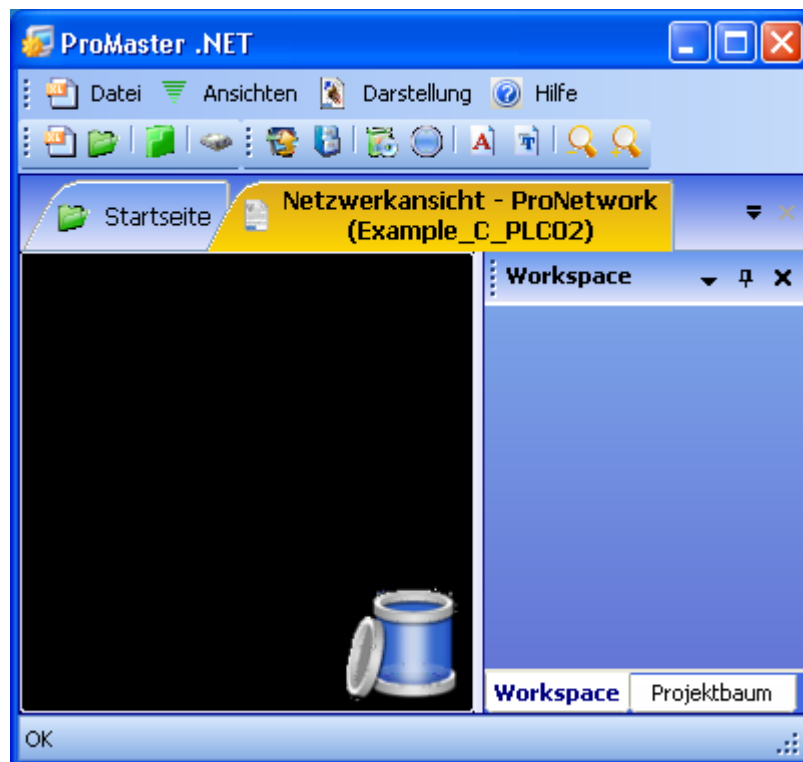


Abbildung 77: ProMaster - Projektname vergeben

Öffnen Sie jetzt über das Menü "Ansichten\Katalog" den Baumüller Katalog. Der Baumüller Katalog enthält die Geräte, Bussystem und Komponenten, die Baumüller Ihnen per Default zur Verfügung stellt.

Für unser Beispiel verwenden wir das Gerät

- "b maXX controller PLC CANopen-Master" (aus der Gruppe "b maXX controller PLC System")

Da wir unser Beispiel ohne CANopen-Master im b maXX controller PLC System erstellen, löschen wir den CANopen-Master aus unserem b maXX controller PLC System in-

dem wir mit im Kontextmenü auf dem Bus CANopen-Master (die gelbe Linie unter der b maXX controller PLC wird dabei rot) "Entfernen" anwählen.

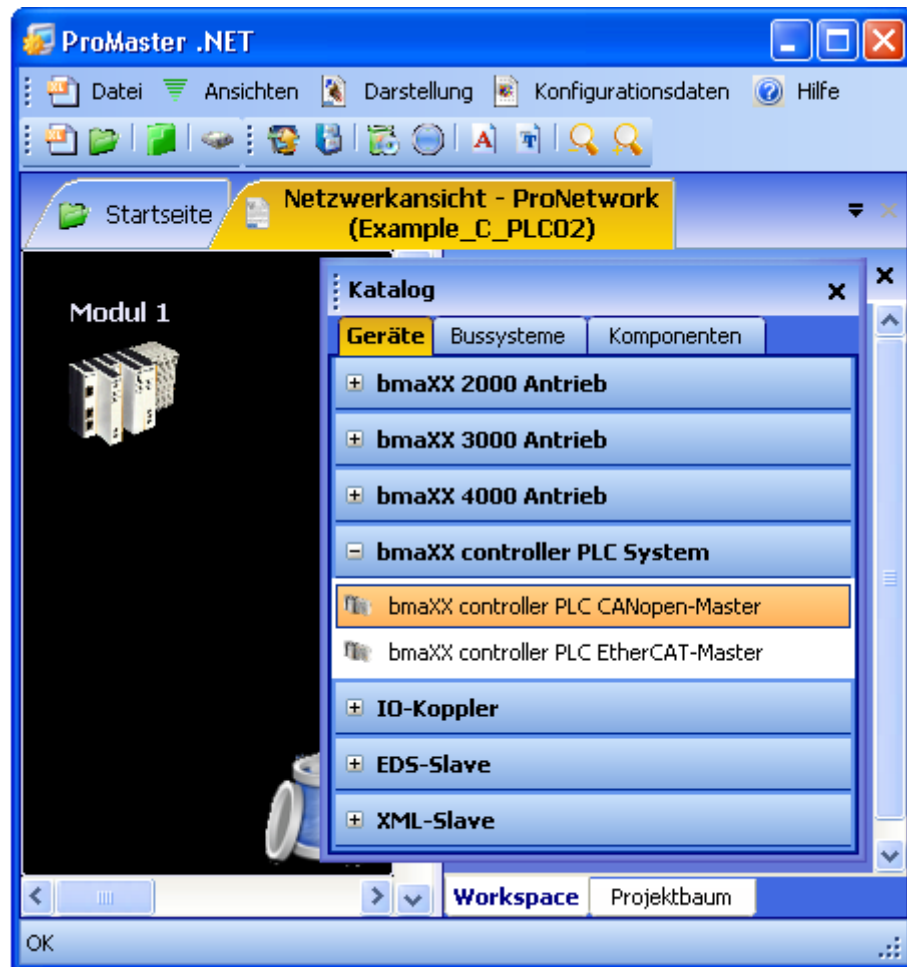


Abbildung 78: ProMaster - Maschinenkonfiguration angelegt

Im Bild sehen Sie jetzt die b maXX controller PLC. Der Übersichtlichkeit halber wird jetzt das Fenster "Katalog" geschlossen.

Der Name des Moduls wird geändert indem man auf das Modul klickt und dann über das Kontextmenü und "Eigenschaften" das Eigenschaften-Fenster des Moduls öffnet.

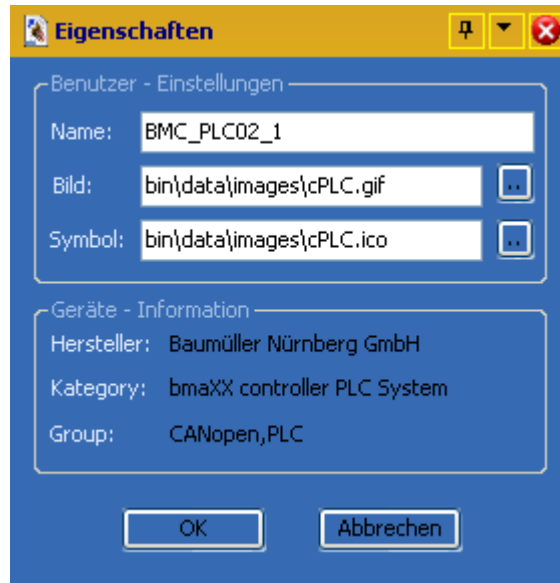


Abbildung 79: ProMaster - Modul-Eigenschaften

In unserem Beispiel geben wir der b maXX controller PLC den Namen BMC\_PLC02\_1. In der ProMaster Netzwerk-Ansicht und im Projektbaum steht nun der neue Name.

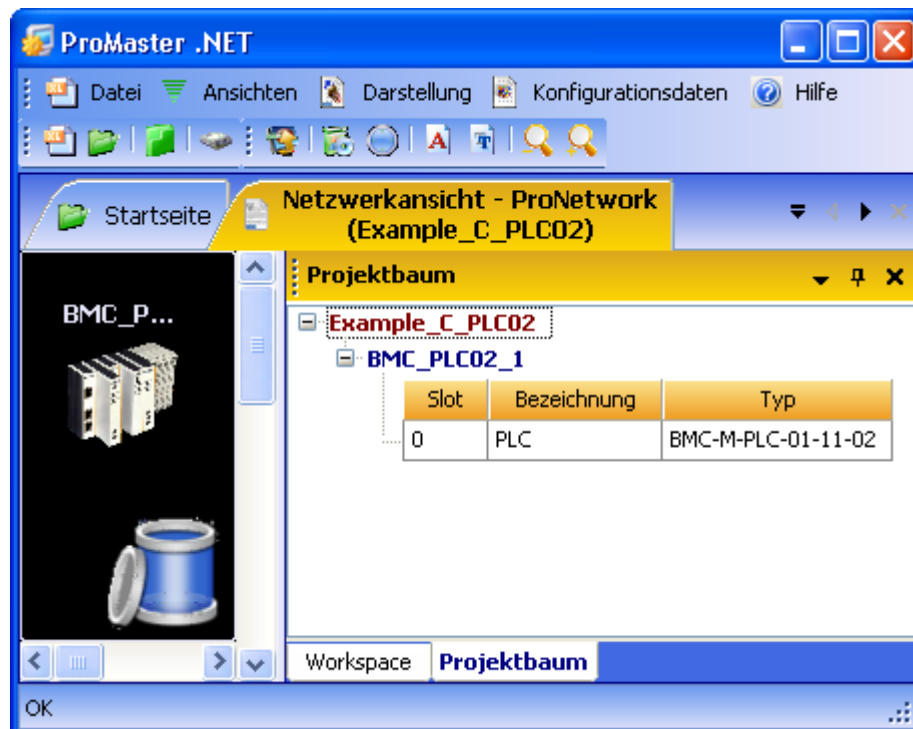


Abbildung 80: ProMaster - Maschinenkonfiguration - Module umbenannt

Die Kommunikationseinstellungen für die Kommunikation von ProMaster zur b maXX controller PLC können Sie einstellen in dem Sie auf das Gerät „BMC\_PLC\_1“ klicken und über das Kontextmenü „Kommunikations-Einstellungen“ auswählen. Speichern Sie anschließend das Projekt über Menü „Datei\Projekt speichern“.

### 6.1.3 Konfigurierung des b maXX Systems mit ProMaster

Jetzt muss noch das ProProg wt III Projekt "Example\_C\_PLC02.mwt" in das ProMaster Projekt "Example\_C\_PLC02.bmxml" eingebunden werden.

Öffnen Sie im ProMaster Projekt in der Netzwerk-Ansicht für unsere b maXX controller PLC das Fenster "PLC-Konfiguration" in dem Sie auf das Gerät "BMC\_PLC02\_1" klicken und über das Kontextmenü

"Konfigurationsdaten\PLC (Modulnummer 0)\PLC - Konfiguration" anwählen und dort das Register "IEC" auswählen.

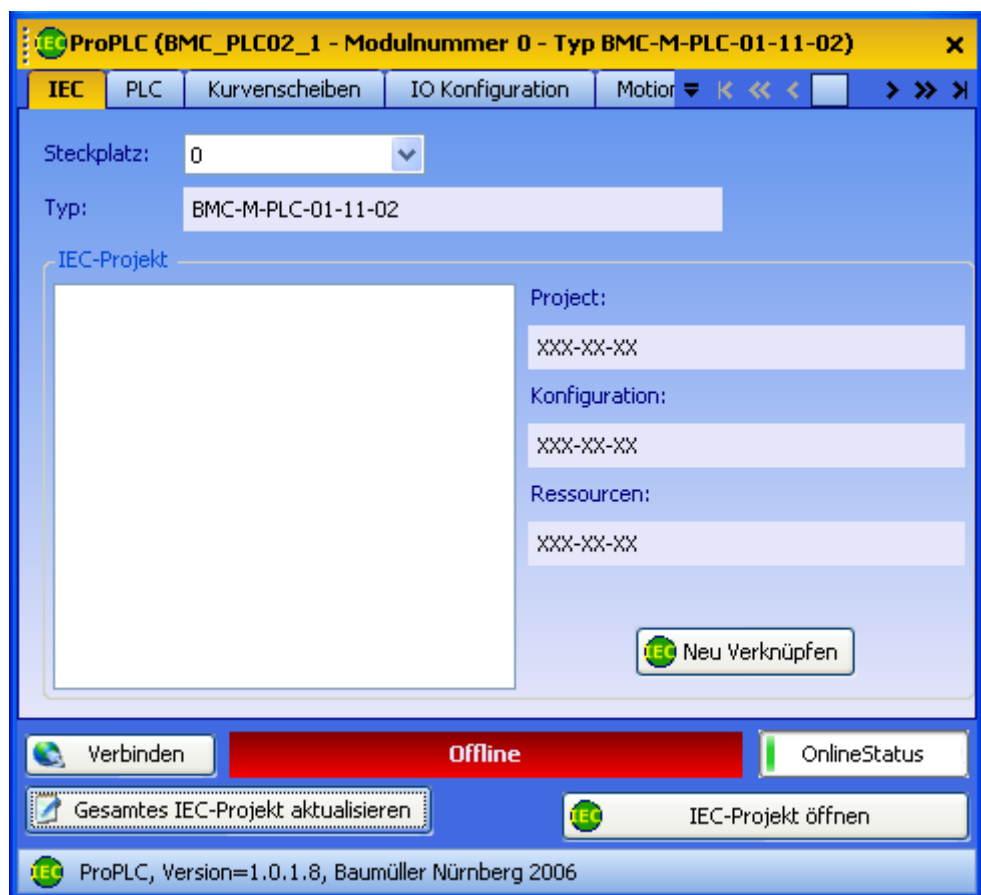


Abbildung 81: Register „IEC“



#### HINWEIS

Mit ProMaster können nur ProProg wt III Projekte geöffnet werden. Falls Sie ein bestehendes PROPROG wt II Projekt verwenden wollen, müssen Sie dieses zuerst mit ProProg wt III öffnen (und dabei konvertieren) und können es danach in ProMaster öffnen und verwenden.

Beachten Sie, dass dabei Ihre PROPROG wt II-Bibliotheken ebenfalls konvertiert werden!

Klicken Sie auf den Button "Neu Verknüpfen" und wählen Sie unser Beispiel IEC Projekt "Example\_C\_PLC02.mwt", welches wir in Kapitel [▶Anlegen eines ProProg wt III Projekts für die b maXX controller PLC](#) ab Seite 136 angelegt haben, aus.

Im Register „IEC“ wird das eben ausgewählte Projekt angezeigt.

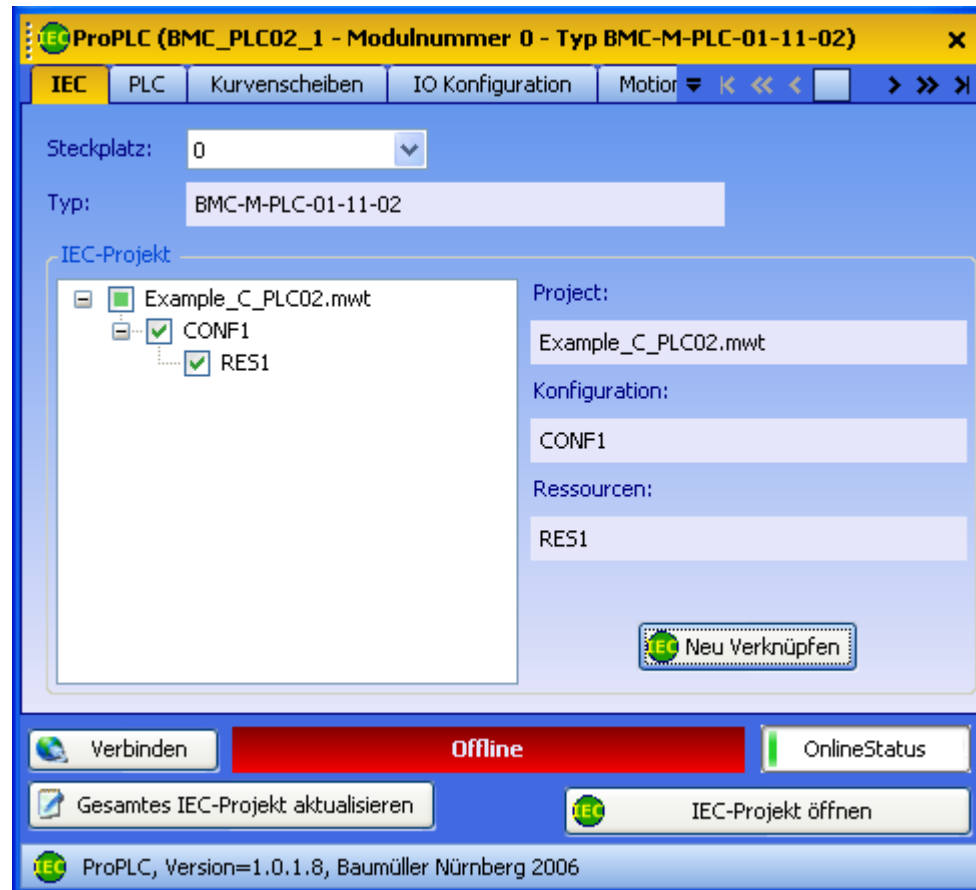


Abbildung 82: Register „IEC“

Speichern Sie die bisherigen Einstellungen in ProMaster über Menü "Datei\Projektspeichern".

## 6.1.4 Konfigurierung der IOs mit ProMaster

In diesem Abschnitt erläutern wir, wie eine IO-Konfiguration für die b maXX controller PLC erstellt und mit ProMaster verbunden wird.

Öffnen Sie im ProMaster Projekt in der Netzwerk-Ansicht für unsere b maXX controller PLC die Geräte-Ansicht in dem Sie auf das Gerät "BMC\_PLC02\_1" klicken und über das Kontextmenü

"Konfigurationsdaten\PLC (Modulnummer 0)\IO - Konfiguration" anwählen.

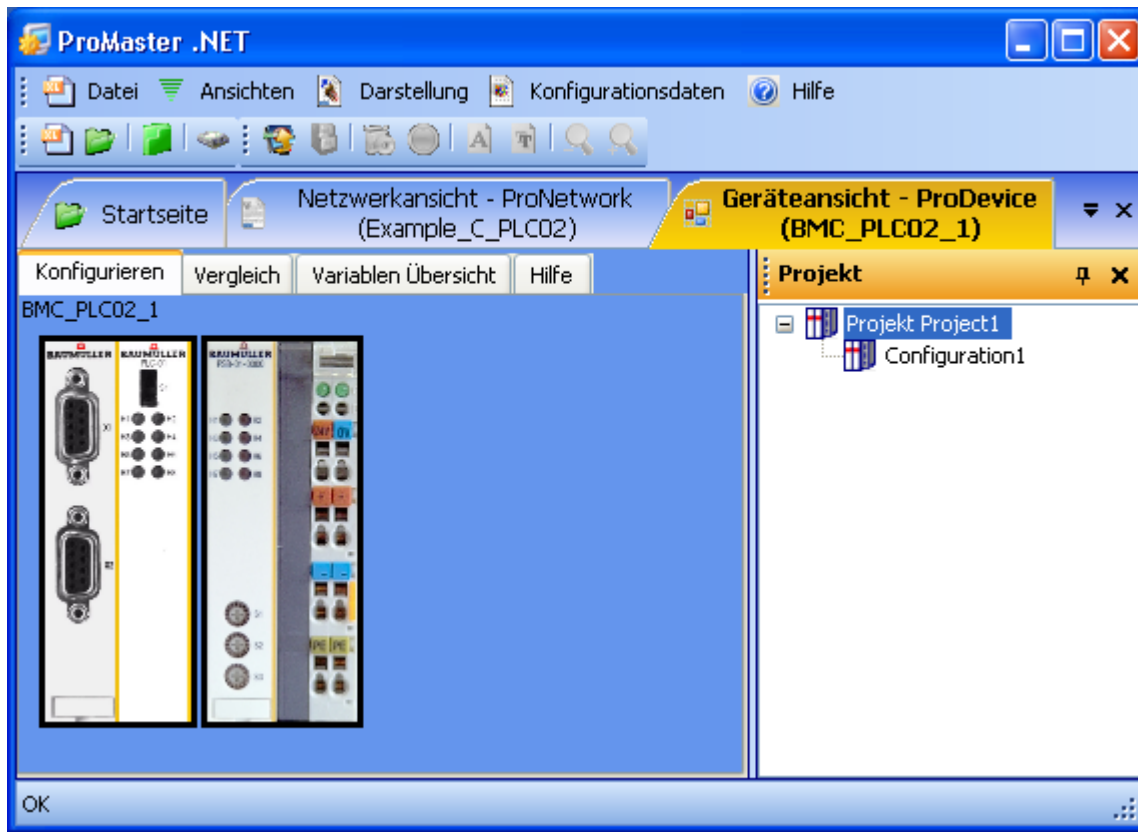


Abbildung 83: ProMaster - IO Konfiguration 1

Öffnen Sie jetzt über das Menü "Ansichten\Katalog" den Baumüller Katalog. Wählen Sie das Register IO-Module an.

Für unser Beispiel verwenden wir die IO Module:

- DO8000 (8 digitale Ausgänge, aus der Gruppe "Digitale IO-Module")
- DI8000 (8 digitale Eingänge, aus der Gruppe "Digitale IO-Module")
- AO2010 (2 analoge Ausgänge, aus der Gruppe "Analoge IO-Module")
- AI2010 (2 analoge Eingänge, aus der Gruppe "Analoge IO-Module")

Platzieren Sie jetzt per Drag&Drop das IO Modul DO8000 (rechts vom Netzteil für b maXX controller PLC). Die Endklemme wird automatisch rechts vom IO Modul platziert. Ziehen Sie anschließend das IO Modul DI8000 zwischen "DO8000" und Endklemme, dann AO2010 zwischen DI8000 und Endklemme und dann AI2010 zwischen AO2010 und Endklemme.



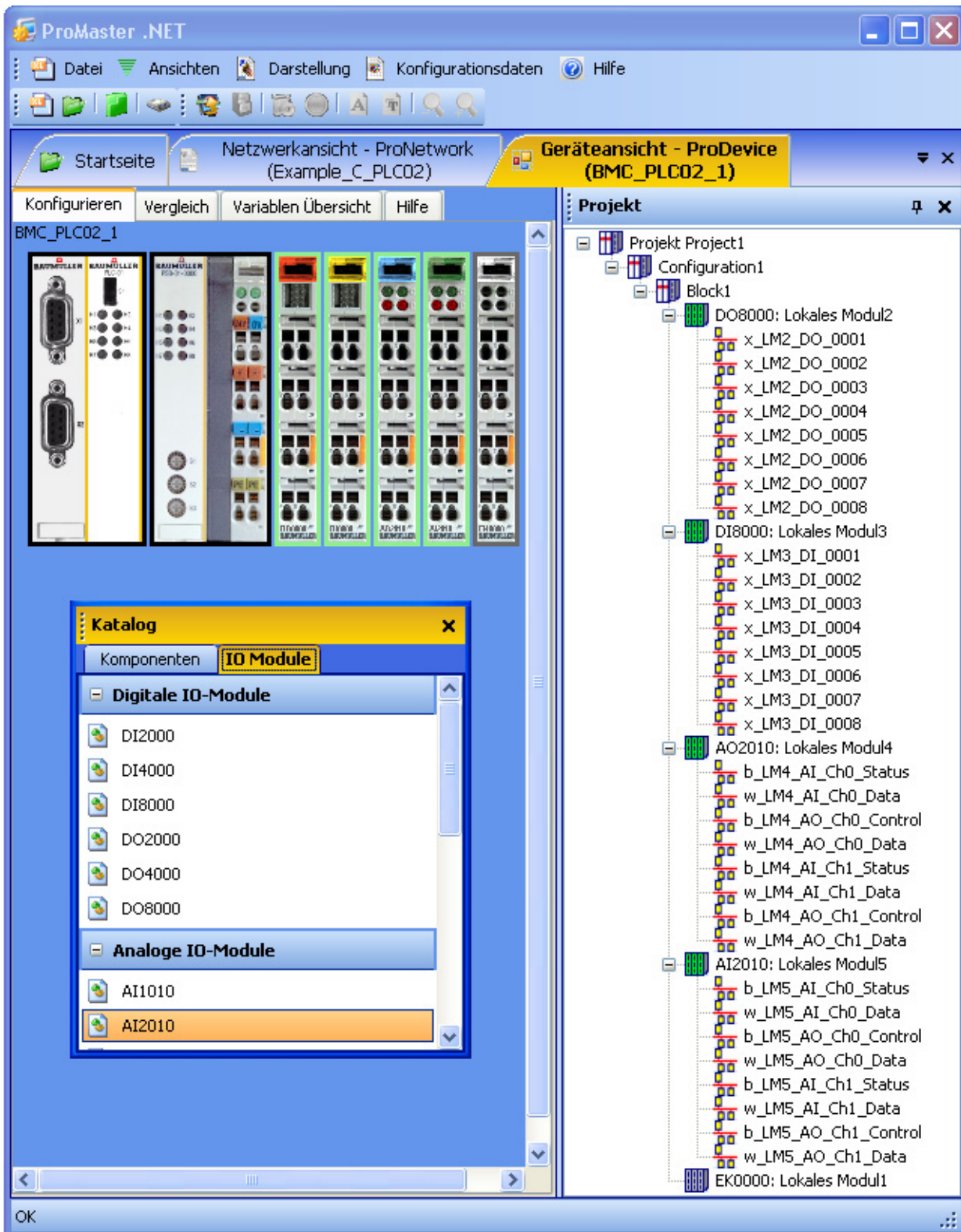


Abbildung 84: ProMaster - IO Konfiguration 2

In „Konfigurationen“ sehen Sie die IO-Module und ihre einzelnen Ein- bzw. Ausgänge. Die IEC Variablennamen der einzelnen Ein- und Ausgänge können sowohl in „Konfigurationen“ als auch im Register „IEC - Variablen“ der „Geräte-Ansicht“ geändert werden.

Der Übersichtlichkeit halber wird jetzt das Fenster „Katalog“ geschlossen.

Wählen Sie jetzt in der „Geräte-Ansicht“ das Register „Vergleich“ an und übernehmen Sie die eben zusammengestellte IO-Konfiguration in das IEC Projekt für ProProg wt III. Verwenden Sie dazu die Pfeile.

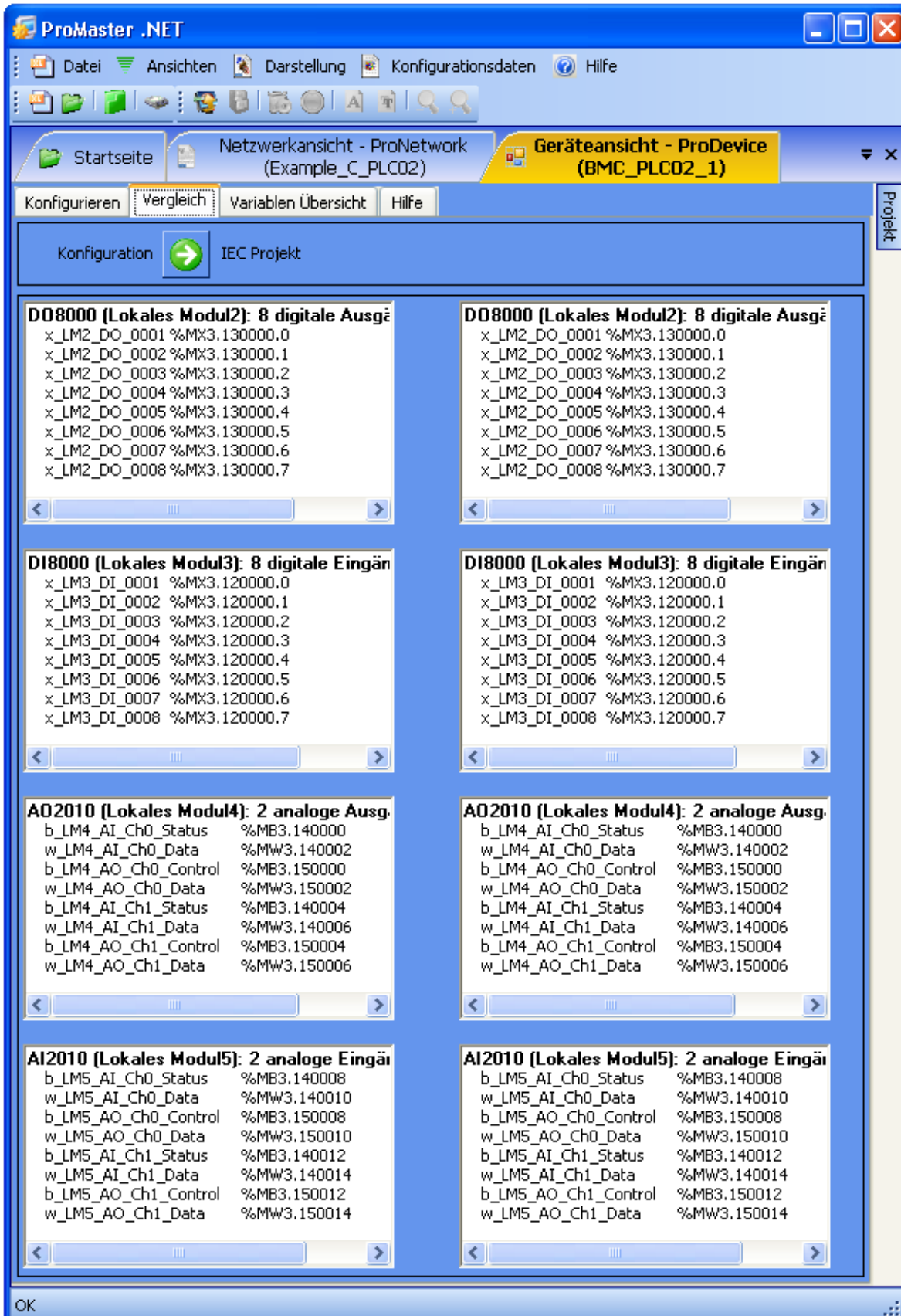


Abbildung 85: ProMaster - IO Konfiguration Vergleich

Wechseln Sie jetzt wieder zum Fenster „PLC Konfiguration“. Falls Sie es geschlossen hatten, öffnen Sie es indem Sie auf das Gerät „BMC\_PLC\_1“ klicken und über das Kontextmenü „Konfigurationsdaten\PLC (Slot 0)\PLC - Konfiguration“ auswählen.

Im Register „IO Konfiguration“ werden unter anderem Name, Adresse und Kommentar der Variablen im IEC Projekt dargestellt.

Das Kürzel „\_LM\_“ in den Default IO-Variablennamen steht für „Local Modul“.

Die IEC-Variablen werden in das IEC-Projekt (globales Variablenarbeitsblatt „Global\_Variables“) geschrieben, wenn in „PLC - Konfiguration“ im Register „IO Konfiguration“ der Button „IO-Variablen aktualisieren“ betätigt wird.

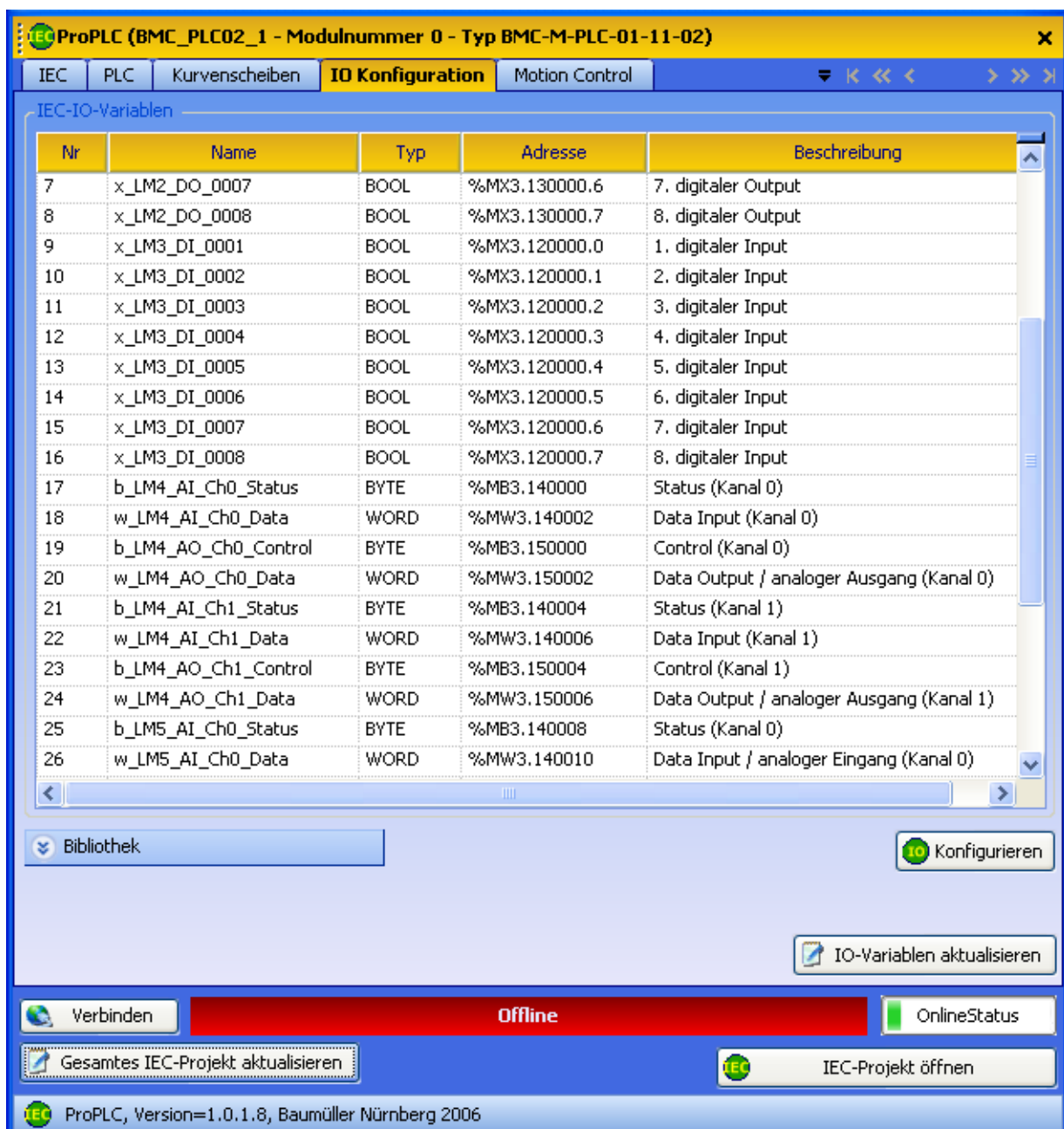


Abbildung 86: ProMaster - PLC Konfiguration - Register IO mit DO, DI, AI und AO

Klicken Sie jetzt im Fenster "PLC Konfiguration" auf den Button "Gesamtes IEC-Projekt aktualisieren". Die Daten für die b maXX controller PLC werden jetzt erzeugt. Dieser Vorgang kann etwas länger dauern, da unter anderem ProProg wt III geöffnet wird und im IEC-Projekt die konfigurierten IEC-Variablen in das globale Variablenarbeitsblatt geschrieben werden.

ProMaster hat jetzt die Variablen für die IOs in das globale Variablen Arbeitsblatt "Global\_Variables" in die Gruppe "IOVariables" geschrieben. Diese Variablen stehen jetzt für die Programmierung im ProProg wt III Projekt zur Verfügung.

Name	Typ	Verwendung	Beschreibung	Adresse
<b>IOVariables</b>				
x_LM2_DO_0001	BOOL	VAR_GLOBAL	1. digitaler Output	%MX3.130000.0
x_LM2_DO_0002	BOOL	VAR_GLOBAL	2. digitaler Output	%MX3.130000.1
x_LM2_DO_0003	BOOL	VAR_GLOBAL	3. digitaler Output	%MX3.130000.2
x_LM2_DO_0004	BOOL	VAR_GLOBAL	4. digitaler Output	%MX3.130000.3
x_LM2_DO_0005	BOOL	VAR_GLOBAL	5. digitaler Output	%MX3.130000.4
x_LM2_DO_0006	BOOL	VAR_GLOBAL	6. digitaler Output	%MX3.130000.5
x_LM2_DO_0007	BOOL	VAR_GLOBAL	7. digitaler Output	%MX3.130000.6
x_LM2_DO_0008	BOOL	VAR_GLOBAL	8. digitaler Output	%MX3.130000.7
x_LM3_DI_0001	BOOL	VAR_GLOBAL	1. digitaler Input	%MX3.120000.0
x_LM3_DI_0002	BOOL	VAR_GLOBAL	2. digitaler Input	%MX3.120000.1
x_LM3_DI_0003	BOOL	VAR_GLOBAL	3. digitaler Input	%MX3.120000.2
x_LM3_DI_0004	BOOL	VAR_GLOBAL	4. digitaler Input	%MX3.120000.3
x_LM3_DI_0005	BOOL	VAR_GLOBAL	5. digitaler Input	%MX3.120000.4
x_LM3_DI_0006	BOOL	VAR_GLOBAL	6. digitaler Input	%MX3.120000.5
x_LM3_DI_0007	BOOL	VAR_GLOBAL	7. digitaler Input	%MX3.120000.6
x_LM3_DI_0008	BOOL	VAR_GLOBAL	8. digitaler Input	%MX3.120000.7
b_LM4_AI_Ch0_Status	BYTE	VAR_GLOBAL	Status (Kanal 0)	%MB3.140000
w_LM4_AI_Ch0_Data	WO...	VAR_GLOBAL	Data Input (Kanal 0)	%MW3.140002
b_LM4_AO_Ch0_Control	BYTE	VAR_GLOBAL	Control (Kanal 0)	%MB3.150000
w_LM4_AO_Ch0_Data	WO...	VAR_GLOBAL	Data Output / analoger Ausgang (Kanal 0)	%MW3.150002
b_LM4_AI_Ch1_Status	BYTE	VAR_GLOBAL	Status (Kanal 1)	%MB3.140004
w_LM4_AI_Ch1_Data	WO...	VAR_GLOBAL	Data Input (Kanal 1)	%MW3.140006
b_LM4_AO_Ch1_Control	BYTE	VAR_GLOBAL	Control (Kanal 1)	%MB3.150004
w_LM4_AO_Ch1_Data	WO...	VAR_GLOBAL	Data Output / analoger Ausgang (Kanal 1)	%MW3.150006
b_LM5_AI_Ch0_Status	BYTE	VAR_GLOBAL	Status (Kanal 0)	%MB3.140008
w_LM5_AI_Ch0_Data	WO...	VAR_GLOBAL	Data Input / analoger Eingang (Kanal 0)	%MW3.140010
b_LM5_AO_Ch0_Control	BYTE	VAR_GLOBAL	Control (Kanal 0)	%MB3.150008
w_LM5_AO_Ch0_Data	WO...	VAR_GLOBAL	Data Output (Kanal 0)	%MW3.150010
b_LM5_AI_Ch1_Status	BYTE	VAR_GLOBAL	Status (Kanal 1)	%MB3.140012
w_LM5_AI_Ch1_Data	WO...	VAR_GLOBAL	Data Input / analoger Eingang (Kanal 1)	%MW3.140014
b_LM5_AO_Ch1_Control	BYTE	VAR_GLOBAL	Control (Kanal 1)	%MB3.150012
w_LM5_AO_Ch1_Data	WO...	VAR_GLOBAL	Data Output (Kanal 1)	%MW3.150014

Abbildung 87: Globales Variablenarbeitsblatt mit den Daten von ProMaster

Kompilieren Sie jetzt das IEC-Projekt in ProProg wt III über das ProProg wt III Menü „CodeProjekt neu erzeugen“.

Führen Sie jetzt den Download des IEC-Projektes auf die b maXX controller PLC durch (ProProg wt III - Menü „OnlineProjektkontrolle...“ → „Senden“ → Bootprojekt „Senden“).

Führen Sie jetzt an der b maXX controller PLC einen Reset aus und schalten Sie danach die PLC in den Zustand „RUN“.

Jetzt können Sie über das IEC Projekt in ProProg wt III die Ausgänge schreiben und die Eingänge lesen.

### 6.2 Modul Konfigurator ProModul

---

Mit dem Modul Konfigurator werden Systemkomponenten/Module, die rechts der b maXX controller PLC stecken (z. B. I/O-Module), konfiguriert.

Für die Module werden Kommunikationsvariablen angelegt, über die die Eingänge dieser Module gelesen bzw. die Ausgänge dieser Module geschrieben/gesetzt werden können.

#### Durchzuführende Schritte

Voraussetzung

Die Montage, der Anschluss und die Inbetriebnahme von

- b maXX controller PLC
- Netzteil für b maXX controller PLC
- Systemkomponenten die links der b maXX controller PLC gesteckt sind (z. B. CANopen-Master)
- Systemkomponenten die rechts der b maXX controller PLC gesteckt sind (z. B. digitale und/oder analoge I/O-Module, Endmodul)

müssen erfolgreich abgeschlossen sein.

Damit über die b maXX controller PLC die Systemkomponenten angesprochen und Daten ausgetauscht werden können, müssen folgende Schritte abgearbeitet werden.

- Anlegen eines PROPROG wt II Projekts für die b maXX controller PLC
- Konfigurierung des b maXX Systems mit dem Modul Konfigurator
- Konfigurierung der Systemkomponenten/Module die rechts der b maXX controller PLC gesteckt sind.

Ergebnis:

Variablenliste für die Initialisierung und/oder den Betrieb dieser Systemkomponenten

Anschließend erfolgt die Programmierung des Applikationsteils im PROPROG wt II Projekt für die b maXX controller PLC. Anleitungen für die Programmierung der Systemkomponenten, die rechts der b maXX controller PLC gesteckt sind, entnehmen Sie bitte den jeweiligen Applikationshandbüchern.

Beispiel

Dieses Kapitel wird von einem Beispiel begleitet. Im Beispiel erstellen wir eine Konfiguration mit (von links nach rechts) folgenden Systemkomponenten/Modulen rechts der b maXX controller PLC:

- Netzteil für b maXX controller PLC
- 2 digitale Eingänge +24V DC (DI2000)
- 2 digitale Ausgänge +24V DC (DO2000)
- 2 analoge Eingänge 0-10V (AI2010)
- 2 analoge Ausgänge 0-10V (AO2010)
- Endmodul (EK0000)

## 6.2.1 Anlegen eines PROPROG wt II Projekts für die b maXX controller PLC

### 6.2.1.1 Allgemein

Falls Sie im Abschnitt [b maXX controller PLC Projekt](#) ab Seite 35 noch kein eigenes Projekt für Ihre Applikation angelegt haben, erstellen Sie das Projekt bitte mit der Vorlage BMC\_M\_PLC01 mit Konfigurator. Sie benötigen dazu ein PROPROG wt II Version 3.1 Build 274 und den Modul Konfigurator für PROPROG wt II Version 1.0 (oder höher). Die Versionsnummer von PROPROG wt II finden Sie auf der Hülle der Installations CD von PROPROG wt II oder in PROPROG wt II selbst im Menüpunkt ? \ Info.

Prüfen Sie ob die Bibliotheken

BM\_TYPES\_20bd06 (oder höher)

SYSTEM2\_C\_PLC01\_20bd00 (oder höher)

sowie gegebenenfalls die jeweiligen Bibliotheken für Ihre Systemkomponenten in Ihrem PROPROG wt II - Projekt vorhanden sind.

Sollte dies nicht der Fall sein, binden Sie diese Bibliotheken bitte in Ihr Projekt ein. Sie enthalten wichtige Datentypen und Funktionsbausteine für Ihre Systemkomponenten.

### 6.2.1.2 Beispiel: Anlegen des Projekts "Modul\_Example"

Es wurde das Beispiel-Projekt „Modul\_Example“ mit der Vorlage „BMC\_M\_PLC01 mit Konfigurator“ angelegt. Die Bibliotheken

BM\_TYPES\_20bd06

SYSTEM2\_C\_PLC01\_20bd00

sind eingebunden.

Im Beispiel werden keine Systemkomponenten/Module, die links der b maXX controller PLC gesteckt sind, verwendet, so dass auch keine entsprechenden Bibliotheken eingebunden werden müssen.

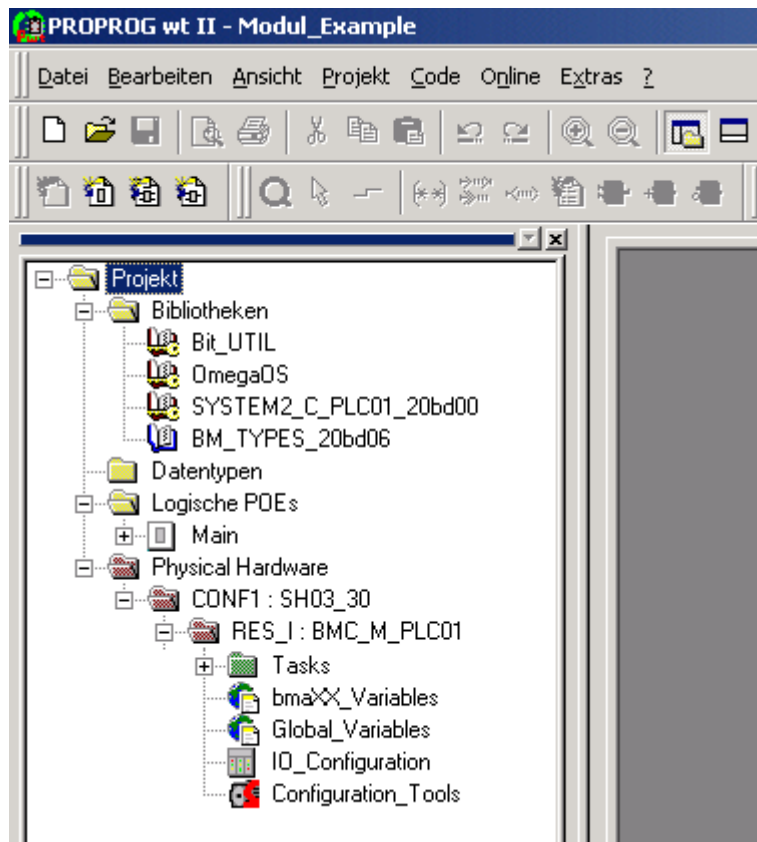


Abbildung 88: Beispiel: Anlegen des Projekts Modul\_Example

### 6.2.2 Konfigurierung des b maXX Systems mit dem Modul Konfigurator

Im PROPROPAG wt II Projekt, das Sie (unter [►Anlegen eines PROPROPAG wt II Projekts für die b maXX controller PLC◄](#) ab Seite 151) bereits angelegt haben, finden Sie den Button für die Konfiguratoren (Configuration\_Tools). Nach dem Doppelklick auf diesen Button öffnet sich das Konfiguratoren-Fenster, unter anderem mit dem Modul Konfigurator (BMC\_Configuration).



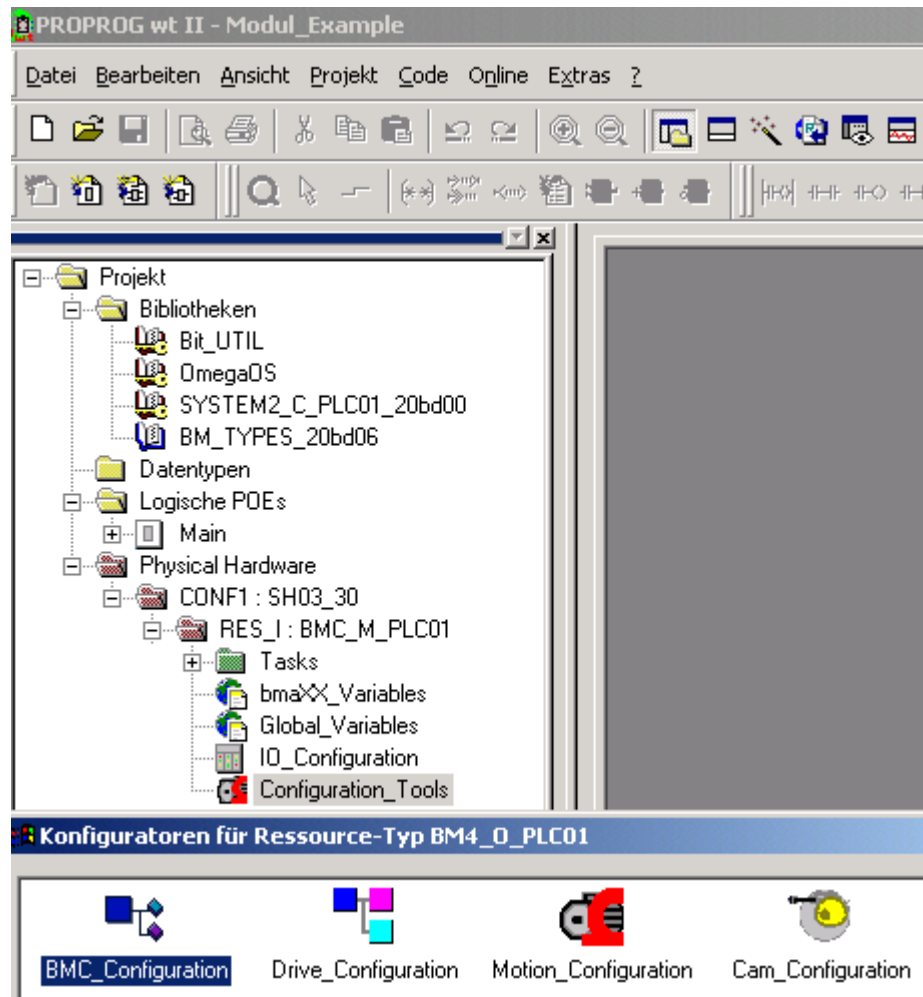


Abbildung 89: Öffnen des Modul Konfigurators

Bevor Sie den Modul Konfigurator öffnen, kompilieren Sie bitte Ihr PROPROG wt II Projekt (Menü: Code\Projekt neu erzeugen).

Klicken Sie jetzt auf den Button zum Öffnen des Modul Konfigurators.

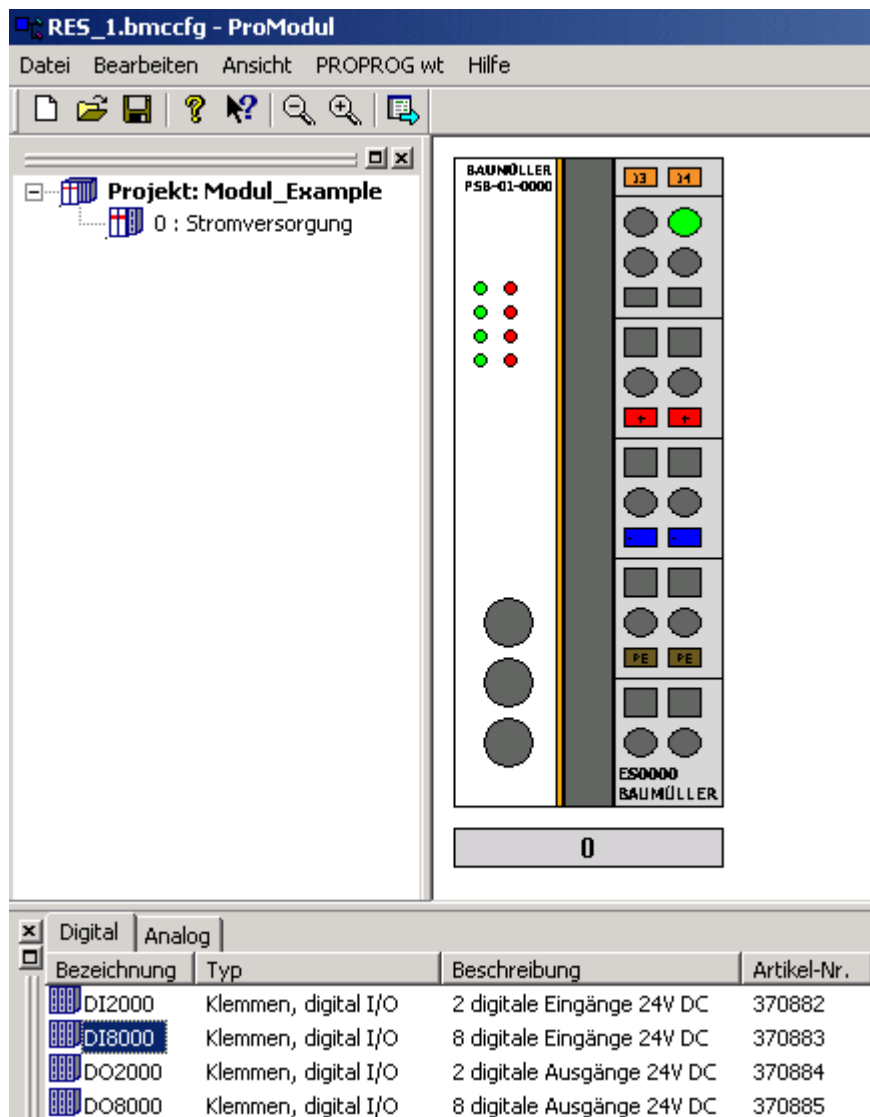


Abbildung 90: Modul Konfigurator

Im Projektbaum des Modul Konfigurators ist folgende Konfiguration eingestellt:

- Netzteil für b maXX controller PLC (Stromversorgung)

Diese Konfiguration stellt die Ausgangskonfiguration dar. Für digitale/analoge Ein- und/oder Ausgänge müssen entsprechende Module (rechts) zugefügt werden.

In den folgenden Abschnitten erläutern wir wie Sie einzelne Module ändern, hinzufügen oder auch löschen können.

Zum Einfügen einer weiteren Systemkomponente gehen Sie wie folgt vor:

- Wählen Sie im Fenster „Katalog“ das Register Digital an.
- Wählen Sie aus dem Fenster "Katalog" Ihr Modul, z. B. 2 digitale Eingänge 24 V DC (DI2000) aus.
- Ziehen Sie dieses Modul per Drag&Drop rechts neben das Netzteil.

Damit wird das neue Modul übernommen und in die Grafik eingepasst.

Falls Sie weitere Module verwenden wollen, wiederholen Sie diese Schritte.

Sie können maximal 64 Module verwenden.

Für unser Beispiel fügen Sie nun weitere Module hinzu:

2 digitale Ausgänge +24 V DC (DO2000),

2 analoge Eingänge 0-10 V (AI2010)

2 analoge Ausgänge 0-10 V (AO2010)

Die Konfiguration der Systemkomponenten/Module erfolgt über die entsprechenden Konfiguratoren, die Sie über das Kontextmenü und den Menüpunkt "Konfigurieren" öffnen. Anleitungen hierzu erhalten Sie aus den jeweiligen Betriebsanleitungen der Systemkomponenten/Module und der Online-Hilfe des Konfigurators.

Das b maXX System aus unserem Beispiel sieht jetzt wie folgt aus:

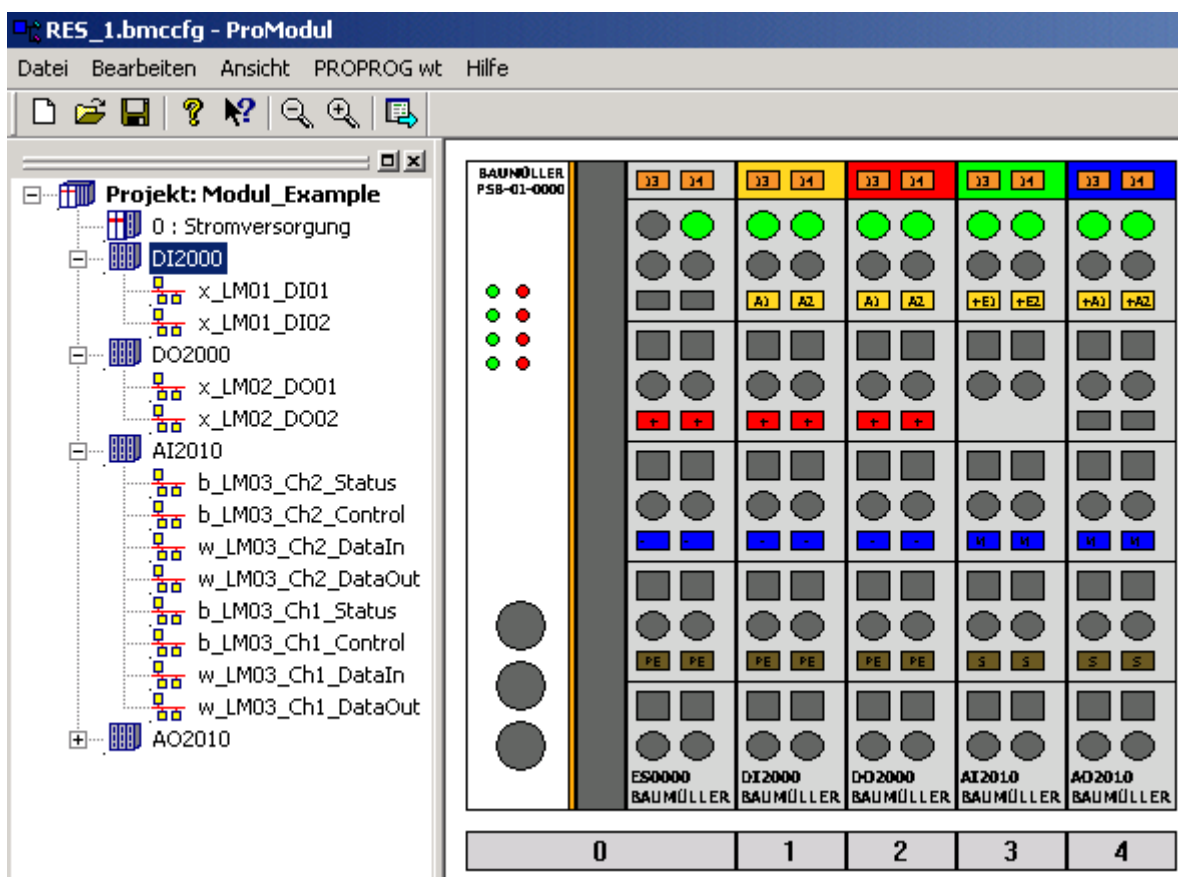


Abbildung 91: Beispiel - b maXX System im Konfigurator (ohne Endmodul)

Den Variablen der Module werden Default-Namen vergeben. Sie können diese Namen ändern, indem Sie über das Kontextmenü der einzelnen Module den Punkt „Konfigurieren“ anwählen.

Im Beispiel wird der digitale Eingang 1 vom Modul DI2000 auf den Variablennamen x\_MyDigitalInput\_0 geändert.

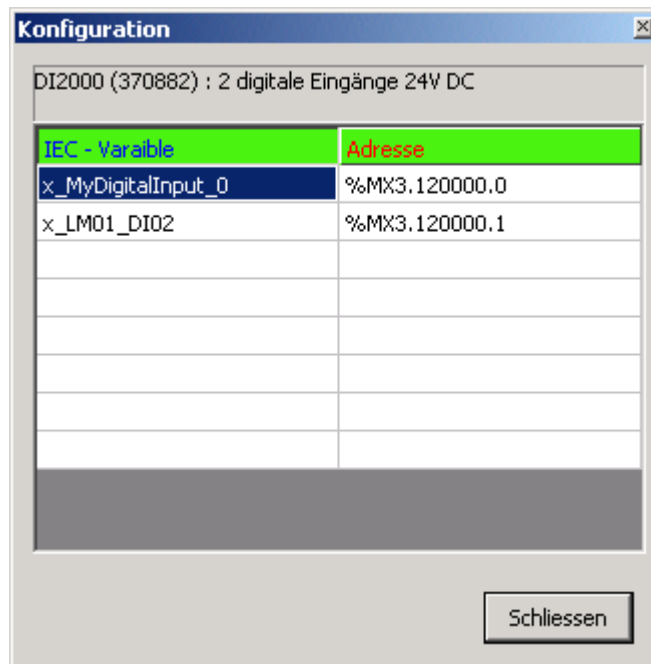


Abbildung 92: Ändern eines Variablennamens im Modul Konfigurator

Im PROPROG wt II Projekt verwenden Sie dann diesen Variablennamen um den Zustand des Digitalen Eingang 1 vom Modul DI2000 auszuwerten.

Das Fenster „Konfiguration“ wird mit „Schließen“ beendet.

Zur Übernahme der Konfigurator-Daten (in das globale Variablen-Arbeitsblatt „bmaXX\_Variables“ im PROPROG wt II Projekt) wählen Sie den Menüpunkt *PRO-PROGwt \ Daten aktualisieren* aus und bestätigen Sie die Sicherheitsabfrage mit ja, ebenso die Quittiermeldung.

Schließen Sie den Modul Konfigurator mit *Datei \ Beenden*. Falls Sie die Sicherheitsabfrage mit „Ja“ bestätigen, wird die eingestellte Konfiguration lokal auf der Festplatte des PC gespeichert.

```

VAR_GLOBAL
  (*LM-1::DI2000 (370882) 2 digitale Eingänge 24V DC *)
x_MyDigitalInput_0 AT %MX3.120000.0::BOOL;
x_LM01_DIO2 AT %MX3.120000.1::BOOL;
  (*LM-2::DO2000 (370884) 2 digitale Ausgänge 24V DC *)
x_LM02_DOO1 AT %MX3.130000.0::BOOL;
x_LM02_DOO2 AT %MX3.130000.1::BOOL;
  (*LM-3::AI2010 (370886) 2 analoge Eingänge 0-10V *)
b_LM03_Ch2_Status AT %MB3.140000::BYTE;
b_LM03_Ch2_Control AT %MB3.150000::BYTE;
w_LM03_Ch2_DataIn AT %MW3.140002::WORD;
w_LM03_Ch2_DataOut AT %MW3.150002::WORD;
b_LM03_Ch1_Status AT %MB3.140004::BYTE;
b_LM03_Ch1_Control AT %MB3.150004::BYTE;
w_LM03_Ch1_DataIn AT %MW3.140006::WORD;
w_LM03_Ch1_DataOut AT %MW3.150006::WORD;
  (*LM-4::AO2010 (370890) 2 analoge Ausgänge 0-10V *)
b_LM04_Ch2_Status AT %MB3.140008::BYTE;
b_LM04_Ch2_Control AT %MB3.150008::BYTE;
w_LM04_Ch2_DataIn AT %MW3.140010::WORD;
w_LM04_Ch2_DataOut AT %MW3.150010::WORD;
b_LM04_Ch1_Status AT %MB3.140012::BYTE;
b_LM04_Ch1_Control AT %MB3.150012::BYTE;
w_LM04_Ch1_DataIn AT %MW3.140014::WORD;
w_LM04_Ch1_DataOut AT %MW3.150014::WORD;

END_VAR

```

Abbildung 93: Beispiel - Arbeitsblatt bmaXX\_Variables

Im PROPROG wt II Projekt kann jetzt auf die Ein- bzw. Ausgänge der Systemkomponenten / Module zugegriffen werden.

### 6.3 Konfigurierung der Systemkomponenten / Module

Es können maximal 64 Systemkomponenten und eine Endklemme rechts der b maXX controller PLC (direkt, d. h. ohne Verlängerungsmodule) gesteckt sein. Falls IO Bus Verlängerungsmodule verwendet werden, muss das Verlängerungsmodul mitgezählt werden und im ersten Block entfällt die Endklemme.

Es können maximal 64 Systemkomponenten und eine Endklemme an IO-Bus Verlängerungsmodulen (direkt, d. h. ohne weitere Verlängerungsmodule) gesteckt sein. Falls weitere IO-Bus Verlängerungsmodule verwendet werden muss das weitere Verlängerungsmodul mitgezählt werden und im jeweiligen Block entfällt die Endklemme.

Es können maximal 255 Module (über Verlängerungsmodule) an einer b maXX controller PLC gesteckt sein.

Bei der Zusammenstellung der Module sollte man jedoch darauf achten, dass aus Performance-Gründen alle digitalen Module als Gruppe und alle analogen Module als Gruppe angeordnet sind.

Anleitungen zur Verwendung der Systemkomponenten/Module und ihrer Kommunikationsvariablen erhalten Sie in den folgenden Kapitel und in der Online-Hilfe vom Enginee-

ring Framework ProMaster (in Verbindung mit ProProg wt III) bzw. vom Modul Konfigurator ProModul (in Verbindung mit PROPROG wt II).

### 6.3.1 Digitale Eingänge / Digitale Ausgänge

Für die digitalen Eingänge und die digitalen Ausgänge steht jeweils ein Datenbereich von 512 Bytes zur Verfügung. Bei der Modul-Konfiguration muss die maximale Anzahl von insgesamt 64 Modulen beachtet werden. D. h. es können in einem System ohne Verlängerungsmodule

maximal 64 Module \* 8 = 512 digitale Eingänge, oder alternativ

maximal 64 Module \* 8 = 512 digitale Ausgänge

bzw. mit Verlängerungsmodulen

maximal 255 Module \* 8 = 2040 digitale Eingänge, oder alternativ

maximal 255 Module \* 8 = 2040 digitale Ausgänge

verarbeitet werden.

Eine Reihenfolge bei der Anordnung der digitalen Module untereinander muss hinsichtlich der Performance nicht beachtet werden.

In der b maXX controller PLC wird jeder digitale Eingang und jeder digitale Ausgang durch eine Variable vom Datentyp BOOL repräsentiert. Die Namen der Variablen können Sie im Modul Konfigurator angeben, alternativ verwenden Sie die vom Modul Konfigurator automatisch erzeugten Namen.

#### Digitale Eingänge / Digital Input - (DI)

Für die digitalen Eingänge steht ein Datenbereich von 512 Bytes zur Verfügung. Die Anzahl der Module ist je System auf 64 begrenzt. D. h. es können in einem System ohne Verlängerungsmodule

maximal 64 Module \* 8 = 512 digitale Eingänge

bzw. mit Verlängerungsmodulen

maximal 255 Module \* 8 = 2040 digitale Eingänge

verarbeitet werden.

Die Erfassung der digitalen Eingänge erfolgt über die Systemkomponenten "Digitaler Eingang / Digital Input" (im weiteren DI-Module genannt). Sie können zwischen verschiedenen digitalen Eingangsmodulen wählen, z. B. Module mit 2 digitalen Eingängen oder Module mit 8 digitalen Eingängen.

#### Digitale Ausgänge / Digital Output - (DO)

Für die digitalen Ausgänge steht ein Datenbereich von 512 Bytes zur Verfügung. Die Anzahl der Module ist je System auf 64 begrenzt. D. h. es können in einem System ohne Verlängerungsmodule

maximal 64 Module \* 8 = 512 digitale Ausgänge

bzw. mit Verlängerungsmodulen

maximal 255 Module \* 8 = 2040 digitale Ausgänge

verarbeitet werden.

Das Setzen der digitalen Ausgänge erfolgt über die Systemkomponenten "Digitaler Ausgang / Digital Output" (im weiteren DO-Module genannt). Sie können zwischen verschiedenen digitalen Ausgangsmodulen wählen, z. B. Module mit 2 digitalen Ausgängen oder Module mit 8 digitalen Ausgängen.

### 6.3.2 Analoge Eingänge / Analoge Ausgänge

Für die analogen Eingänge und die analogen Ausgänge steht jeweils ein Datenbereich von 512 Bytes zur Verfügung.

Dabei ist jedoch zu beachten, dass jeder Analogkanal gleichviel Speicher sowohl im Ausgangsbereich als auch im Eingangsbereich belegt. Außerdem muss bei der Modul-Konfiguration die maximale Anzahl von insgesamt 64 Modulen beachtet werden.

Eine Reihenfolge bei der Anordnung der analogen Module untereinander muss hinsichtlich der Performance nicht beachtet werden.

In der b maXX controller PLC wird jeder analoge Eingang und jeder analoge Ausgang durch zwei Variablen vom Datentyp WORD repräsentiert. Eine der Variablen enthält den Wert (für den analogen Eingang bzw. für den analogen Ausgang), die andere Variable ist für spezielle Kommunikationsaufgaben (Status/Control) reserviert. Wie bereits erwähnt ist zu beachten, dass beide Worte sowohl im Eingangsbereich als auch im Ausgangsbereich abgebildet werden.

Die Namen der Variablen für den Wert (des jeweiligen analogen Eingangs bzw. analogen Ausgangs) können Sie im Modul Konfigurator angeben, alternativ verwenden Sie die vom Modul Konfigurator automatisch erzeugten Namen.

#### Beispiel 1:

32 analoge Ausgangsmodule á 4 analoge Ausgänge:

Für jeden analogen Ausgang wird ein Speicherbereich von 2 Worten (= 4 Byte) benötigt, d. h. für diese Konfiguration werden 512 Byte im Datenbereich belegt.

$$32 * 4 \text{ analoge Ausgänge} * 4 \text{ Byte} = 512 \text{ Byte}$$

Zusätzlich könnten Sie in das System noch bis zu 32 digitale Module (ohne IO-Bus Verlängerung) bzw. 223 digitale Module (mit IO-Bus Verlängerung) integrieren.

#### Beispiel 2:

32 analoge Eingangsmodule á 4 analoge Eingänge.

$$32 * 4 \text{ analoge Eingänge} * 4 \text{ Byte} = 512 \text{ Byte},$$

Zusätzlich könnten Sie in das System noch bis zu 32 digitale Module (ohne IO-Bus Verlängerung) bzw. 223 digitale Module (mit IO-Bus Verlängerung) integrieren.

### Beispiel 3:

Kombination aus z. B.

16 analogen Ausgangsmodulen á 4 analoge Ausgänge und

16 analogen Eingangsmodulen á 4 analoge Eingänge

$16 * 4 \text{ analoge Ausgänge} * 4 \text{ Byte}$

$+ 16 * 4 \text{ analoge Eingänge} * 4 \text{ Byte}$

$= 512 \text{ Byte.}$

Sie können jetzt kein weiteres Analogmodul mehr einfügen.

Zusätzlich könnten Sie in das System jedoch noch bis zu 32 (ohne IO-Bus Verlängerung) bzw. 223 digitale Module (mit IO-Bus Verlängerung) digitale Module integrieren.

### Beispiel 4:

62 analoge Ausgangsmodule á 2 analoge Ausgänge und 1 analoges Ausgangsmodul á 4 analoge Ausgänge:

Für jeden analogen Ausgang wird ein Speicherbereich von 2 Worten (= 4 Byte) benötigt, d. h. für diese Konfiguration werden 512 Byte im Datenbereich belegt.

$[(62 * 2) + (1 * 4)] \text{ analoge Ausgänge} * 4 \text{ Byte} = 512 \text{ Byte}$

Zusätzlich könnten Sie in das System noch 1 digitales Modul (ohne IO-Bus Verlängerung) bzw. 192 digitale Module (mit IO-Bus Verlängerung) integrieren.

### Beispiel 5:

62 analoge Eingangsmodule á 2 analoge Eingänge und 1 analoges Eingangsmodul á 4 analoge Eingänge:

$[(62 * 2) + (1 * 4)] \text{ analoge Eingänge} * 4 \text{ Byte} = 512 \text{ Byte}$

Zusätzlich könnten Sie in das System noch 1 digitales Modul (ohne IO-Bus Verlängerung) bzw. 192 digitale Module (mit IO-Bus Verlängerung) integrieren.

### Beispiel 6:

Kombination aus z. B.

16 analogen Ausgangsmodulen á 4 analoge Ausgänge,

30 analogen Eingangsmodulen á 2 analoge Eingänge und

1 analogen Eingangsmodul á 4 analoge Eingänge.

$16 \text{ analoge Ausgänge} * 4 \text{ Byte}$

$+ [(30 * 2) + (1 * 4)] \text{ analoge Eingänge} * 4 \text{ Byte}$

$= 512 \text{ Byte}$

Sie können jetzt kein weiteres Analogmodul mehr verwenden.

Zusätzlich könnten Sie in das System jedoch noch bis zu 17 digitale Module (ohne IO-Bus Verlängerung) bzw. 224 digitale Module (mit IO-Bus Verlängerung) integrieren.



### Analoge Eingänge / Analog Input - (AI)

Die Erfassung der analogen Eingänge erfolgt über die Systemkomponenten „Analoger Eingang / Analog Input“ (im weiteren AI-Module genannt). Sie können zwischen verschiedenen analogen Eingangsmodulen wählen, z. B. Module mit 2 analogen Eingängen oder Module mit 4 analogen Eingängen.

In unserem Beispiel können Sie die analogen Eingänge der Systemkomponente AI2010 über die Variablen

```
w_LM03_Ch1_DataIn    und    w_LM03_Ch2_DataIn
```

lesen. Bei einer Eingangsspannung von +2,5 V DC am Eingang erhalten Sie in der entsprechenden Variablen den Wert WORD#16#1FFF.

### Analoge Ausgänge / Analog Output - (AO)

Das Setzen der analogen Ausgänge erfolgt über die Systemkomponenten „Analoger Ausgang / Analog Output“ (im weiteren AO-Module genannt). Sie können zwischen verschiedenen analogen Ausgangsmodulen wählen, z. B. Module mit 2 analogen Ausgängen oder Module mit 4 analogen Ausgängen.

In unserem Beispiel können Sie die analogen Ausgänge der Systemkomponente AO2010 über die Variablen

```
w_LM04_Ch1_DataOut   und   w_LM04_Ch2_DataOut
```

schreiben. Sie erreichen am jeweiligen Ausgang eine Ausgangsspannung von +5 V DC indem Sie auf die entsprechende Variable den Wert WORD#16#3FFF schreiben.

### 6.3.3 Alarm-Ausgang

Ist für die Erkennung der ordnungsgemäßen Funktion der SPS ein Hardware-Ausgang notwendig, so kann dies mit dem Ausgang eines digitalen Ausgangsmoduls realisiert werden. Für den Zustand des ordnungsgemäßen Betriebs ist der Ausgang auf 1 zu setzen. Beim Auftreten eines Fehlers muss der Ausgang auf 0 gesetzt werden. Wenn die Ausgangsklemme länger als 100 ms kein Signal über den IO-Bus bekommt, schaltet sie ebenfalls ihre Ausgänge auf 0.





## ANHANG A - ABKÜRZUNGEN

<b>CAN</b>	Controller Area Network	<b>USS®</b>	Warenzeichen Siemens, universelle serielle Schnittstelle
<b>CBPB</b>	Controller Based Parallel Bus		
<b>CPU</b>	Central Processing Unit		
<b>EMV</b>	Elektromagnetische Verträglichkeit		
<b>DPRAM</b>	Dual Ported RAM		
<b>DRAM</b>	Dynamic RAM		
<b>EN</b>	Europäische Norm		
<b>ESD</b>	Electrostatic sensitive device (elektrostatisch gefährdetes Bauteil, EGB)		
<b>EXT, ext</b>	Extern		
<b>I/O</b>	Input/Output, Eingang und Ausgang		
<b>I/O-Bus</b>	Bus für die Eingangs- und Ausgangsmodule (Input module, Output module; I/O)		
<b>LED</b>	Leuchtdiode		
<b>NOVRAM</b>	<b>Non-volatile RAM</b>		
<b>OPC</b>	OLE for Process Control (OLE: Object Linking and Embedding)		
<b>PLC</b>	Process loop control, Speicher programmierbare Steuerung, SPS		
<b>PROPROG wt II</b>	Tool zur Programmierung der b maXX PLC (BMC-M-PLC-01)		
<b>ProProg wt III</b>	Tool zur Programmierung der b maXX PLC (BMC-M-PLC-01 und BMC-M-PLC-02)		
<b>RAM</b>	Random access memory		
<b>RISC</b>	Reduced Instruction Set Computers		
<b>SDRAM</b>	Synchronized Dynamic RAM		
<b>SW</b>	Software		





## Revisionsübersicht

Version	Stand	Änderungen
5.04019.02	07.07.2006	Erweiterung für BMC-M-PLC-02
5.04019.03	15.02.2007	Kap. 4.3 und Kap. 6.3.3 hinzu



## Revisionsübersicht

---



# Index

<b>A</b>		<b>F</b>	
Adresse Netzteil auslesen	78	Fachkraft	13
Alarm-Ausgang	161	Firmware-Bibliothek	57
Anwenderbibliothek	57	Funktionsbaustein	60
Einfügen in Projekt	75	INTR_SET	61
Anwender-Merker	53	LED12	68
Anzeigeelemente	16, 19, 69	LED8	66
Arbeitsblatt	70	T64_REC	116
ASCII-Protokoll	106	T64_RSYN	115
<b>B</b>		TER_INIT	103
b maXX controller PLC	5	TER_R	108
Datentypen	69	TER_S	112
Firmware	60	TER_USS	119
Baumüller	7	TIME_MEASURE_END	61, 129
Begriffe		TIME_MEASURE_START	61, 128
Definition	5	TIMER_A_INIT	82
Bibliotheken		TIMER_A_INIT für BMC-M-PLC-01	83
Übersicht	58	TIMER_A_INIT für BMC-M-PLC-02	89
BMC-M-PLC-01	5	Funktionstest	
Betrieb	18	BMC-M-PLC-01	24
Initialisierung	16	BMC-M-PLC-02	25
LEDs	16	<b>G</b>	
Schalter/Taster S1	18	Gewährleistung und Haftung	14
Testen der Funktion	24	Globale Variablenarbeitsblätter	39
BMC-M-PLC-02	5	Grundfunktionalität	73
Betrieb	21	<b>I</b>	
Initialisierung	20	I/O-Bus	31, 133
LEDs	19	I/O-Konfiguration	39
Schalter/Taster S2	22	Interruptquellen	80
Testen der Funktion	25	INTR_SET	61
Board-Funktion	61	<b>K</b>	
Bootprojekt senden	26	Kommunikation über RS232	41
<b>C</b>		Kommunikationsquelle	39
CAM_PLC01_21bd00	74	Kontrolldialog für Ressourcen	44
CAM_PLC01_30bd00	75	<b>L</b>	
CAM_PLC02_30bd00	75	LED12	68
CBPB	31, 77	LED8	66
CBPB-Hardware-Signal	79	LED-Anzeige	16, 19, 69
<b>D</b>		Level	
Datenbereich	39, 52	Interrupt	56, 63
Datentyp	69	<b>M</b>	
Datenübertragung	32	Merker-Adresse	
Dokumentationsarbeitsblätter	39	absolut	55
<b>E</b>		Modul Konfigurator	152
Echtzeitverhalten	56	<b>N</b>	
Eigenschaft		Netzteil	
Event-Task	56	Adresse auslesen	78
Einschalten	23	Neues Projekt	35
Event-Task	56		



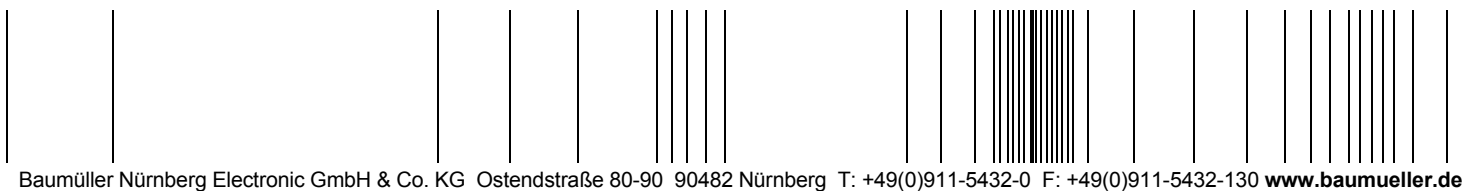
## Stichwortverzeichnis

<b>P</b>			
Personal	13	TER_INIT	103
qualifiziert	13	TER_R	108
Programmiersprachen	33	TER_S	112
Projekt		TER_USS	119
neu, öffnen	35	TIME_MEASURE_END	61, 129
Projekt anlegen		TIME_MEASURE_START	61, 128
mit ProMaster	138	TIMER_A_INIT	81, 82
mit ProModul	151	TIMER_A_INIT für BMC-M-PLC-01	83
Projekt zum Zielsystem senden	46	TIMER_A_INIT für BMC-M-PLC-02	89
Projektvorlage	35		
ProMaster	135	<b>U</b>	
ProModul	150	Übertragungszeit	32
PROPROG wt	33	UNIVERSAL_20bd01	74
PROPROG wt II Bibliothek	76	UNIVERSAL_30bd00	74
Prozessdatenkommunikation	79		
		<b>V</b>	
<b>Q</b>		Verpflichtung und Haftung	13
Qualifiziertes Personal	13	Verzeichnispfad für Bibliotheken	59
<b>R</b>		<b>W</b>	
Remanente Daten	51	Watchdog	50, 127, 128, 131
Remanente Merker	53	WINDER_PLC01_20bd01	75
Ressource	37, 39	WINDER_PLC01_30bd01	75
Einstellungen	40	WINDER_PLC02_30bd01	75
Ressource BMC_M_PLC01	37		
Ressource BMC_M_PLC02	37	<b>Z</b>	
Ressourcen		Zweidraht / Vierdraht	103
Kontrolldialog	44	ZWT-File	76
RS485-Schnittstelle			
Hinweis zur	102		
RUN	17, 18, 21, 22		
<b>S</b>			
Schalter/Taster	22		
SPS-Fehler	130		
Standardbibliothek	73		
STOP	17, 18, 21, 22		
SYSTEM1_C_PLC01_20bd00	61, 73		
SYSTEM1_C_PLC01_30bd00	61, 74		
SYSTEM1_C_PLC02_30bd00	61, 74		
SYSTEM2_C_PLC01_20bd00	60, 73		
SYSTEM2_C_PLC01_30bd00	60, 74		
SYSTEM2_C_PLC02_30bd02	60, 74		
Systemdaten	51		
System-Merker	53		
<b>T</b>			
T64_REC	116		
T64_RSYN	115		
Technologiebaustein			
Kurvenscheibe	74		
Wickler	75		
Template	70		





**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.