

Modbus^(R) für BELIMO Ringbus

Modbus-Kommunikation mit der
BELIMO Ringbus-Lösung

INHALT

Inhalt	2
1. Einführung Ringbus	3
1.1. Ringbus-Topologie	6
1.2. Modul / Einheitentypen	7
2. Einführung in Modbus.....	8
2.1. Adressierung	9
3. Modbus mit Ringbus verwenden	10
3.1. Setup.....	10
3.2. Ringbus-Einheitenstatus auslesen	12
3.3. Befehle an Ringbus Module senden	24
Copyright	28
Rechtliche Hinweise	28

1. EINFÜHRUNG RINGBUS

Die BELIMO Ringbus-Lösung ist eine massgeschneiderte Lösung für die Überwachung und Steuerung von Brandschutz- und Entrauchungsklappen. Sie wird entweder als Standalone-System oder zusammen mit Brandschutzsystemen - z. B. Brandmeldeanlagen und ähnlichen Systemen - eingesetzt.

Die Hauptfunktionen des Ringbus-Systems sind:

- **Schnelle und benutzerfreundliche Systemkonfiguration:** Die Konfiguration des Systems erfolgt schnell und intuitiv. Die Konfigurations- und Steuerungsmatrix kann als Excel-Datei zur Verfügung gestellt und direkt auf die Ringbus-Kontroller geladen werden.

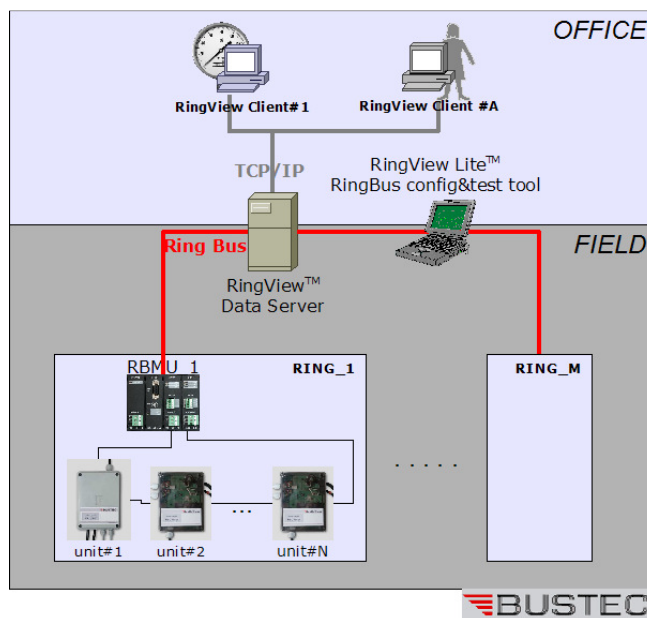


ABBILDUNG 1.1: RINGBUS SYSTEM-HIERARCHIE

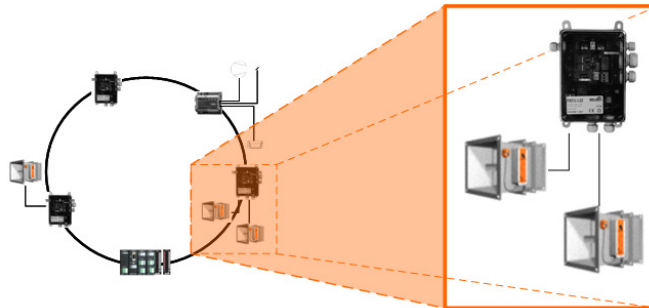
- **Verbesserte Systemanpassungen:** Das System verwendet standardisierte Kommunikationsprotokolle, kann schnell und problemlos konfiguriert werden und besitzt nur wenige Hierarchiestufen. Durch diese Eigenschaften bleibt das System bei erheblich reduzierten Kosten flexibel und benutzerfreundlich.

- **Basierend auf effizienten und zuverlässigen Technologien:**

Ringbus benötigt im Betrieb keine zusätzliche Spezialausrüstung, weder Hardware noch Software. Dadurch können herkömmliche PCs für die Einrichtung, den Betrieb, die Überwachung sowie für Wartung und Diagnose verwendet werden. Dies bedeutet, dass Wartung und Bedienung kostengünstig und überall verfügbar sind. Des Weiteren basieren die Ringbus Installations- und Konfigurationstools auf der plattformübergreifenden Java^(R)-Technologie und sind daher sowohl unter Windows^(R) als auch unter Linux^(R) lauffähig.

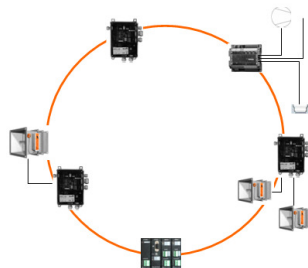
Grundbegriffe des Ringbus-Systems:

- Eine **Einheit / Modul** ist das kleinste Einzelement. Es steuert ein einzelnes Modul der Brandschutzanlage, z. B. eine Entrauchungsklappe.



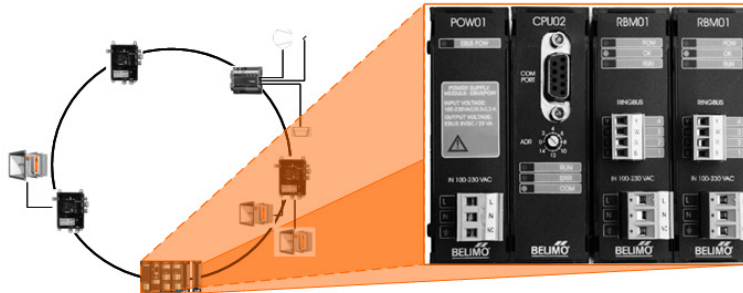
EINHEIT / MODUL

- Ein **Ring** besteht aus mehreren physikalisch miteinander verbundenen Einheiten. Jede Einheit kann zu einem einzelnen Ring gehören.



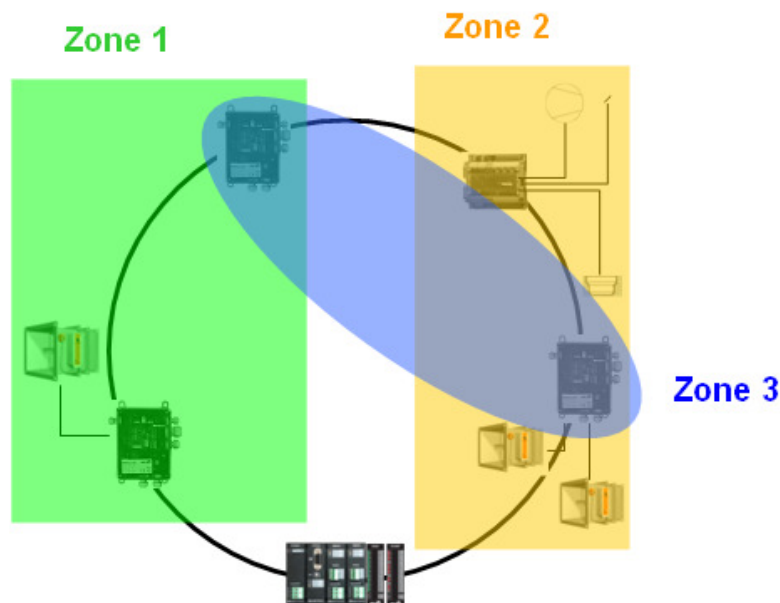
RING

- Der **Ringbus Master CPU (RBCPU)** ist der Master-Kontroller in einem Ring (Abb.1.2) und fungiert als Slave im Mid-Tier-Netzwerk, das aus weiteren RBCPUs und einem RingView SCADA-Server bestehen kann.



RBCPU

- Zone** ist die Kurzbezeichnung für „Sicherheitszone“. Dies ist ein Bereich des Einsatzortes, in dem sowohl Brand- als auch Rauchsicherheit gehandhabt werden. Die Topologie der Zone wird im Brandsicherheitskonzept des Bauprojekts bestimmt, wobei jedes Bauvorhaben gesondert behandelt wird. Jede Einheit kann Teil einer oder auch mehrerer Zonen sein.



ZONE

1.1. Ringbus-Topologie

Jede Ringbus-Installation besteht aus einem oder mehreren Ringen (Abb.1.3). Ein Ring besteht aus mehreren Modulen, die über einen gemeinsamen Kommunikationsbus kommunizieren. Alle Ringbus-Elemente sind physikalisch mit einem einzelnen Ring verbunden und werden hierdurch im System installiert. Alle zu einem Ring gehörenden Einheiten werden durch einen einzelnen Ringbus Master-Kontroller, dem Ringmaster, gesteuert.

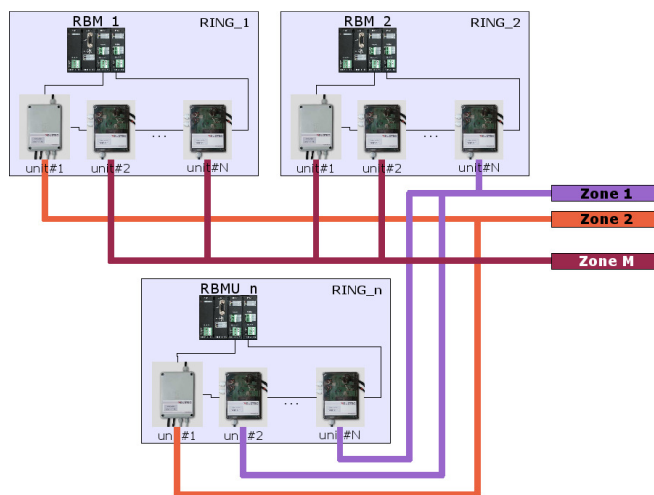


ABBILDUNG 1.3: RINGBUS SYSTEM-TOPOLOGIE

Die BELIMO Ringbus-Lösung ermöglicht den Anschluss von bis zu 500 Einheiten an einem einzigen Ring. Die optimale Anzahl hängt hierbei von der vor Ort verwendeten Systemtopologie, den Kabellängen, externen elektrischen und elektromagnetischen Interferenzen, dem Abstand zwischen benachbarten Einheiten im Ring und weiteren Faktoren ab. Mehrere Ringe können über ihre Ringmaster (RBCPUs) mit dem RingView-Server verbunden werden und somit ein einheitliches und ausgewogenes System bilden.

1.2. Modul / Einheitentypen

RingView unterscheidet zwischen verschiedenen Modultypen:

- **Brandschutzeinheit, RBFU 1.01 ST,**
steuert eine Brandschutzklappe
- **Doppel-Brandschutzeinheit, RBFU 1.05 ST,**
steuert zwei Brandschutzklappen unabhängig voneinander
- **Entrauchungseinheit, RBFU 2.01 ST,**
steuert eine Entrauchungsklappe
- **Feld-I/O-Einheit, RBFU 3.02,**
ein Mehrzweck-I/O-Modul, das bestimmte Signale von
Brandüberwachungs- und HLK-Anlagen überwacht und (digitale) Befehle
an diese sendet.
- **DIN/DOM, RBCPU 1.02, 1.03,**
sind direkt an die RBCPU angeschlossene digitale Ein-/Ausgabemodule,
die für verschiedene Zwecke eingesetzt werden können, etwa zur Anzeige
von Stati und Ereignissen lokaler HLK-Anlagen. Des Weiteren können
Geräte von Fremdherstellern, wie z. B. Datenlogger oder andere zu einer
HLK- oder Brandschutzanlage gehörenden Geräte, angesteuert werden.

2. EINFÜHRUNG IN MODBUS

Modbus ist ein 1979 von der Firma Modicon veröffentlichtes serielles Kommunikationsprotokoll, das für speicherprogrammierbare Steuerungen (SPS) entwickelt wurde. Heute ist es einer der Industriestandard für Kommunikationsprotokolle und ist das in der Industrie am häufigsten verwendete Protokoll, um elektronische Geräte miteinander zu verbinden. Die Hauptgründe, warum Modbus den Vorzug vor anderen Kommunikationsprotokollen erhielt, sind:

1. Es ist ein offenes Protokoll und lizenzkostenfrei.
2. Es kann innerhalb von Tagen, nicht Monaten, implementiert werden.
3. Es überträgt Rohdaten und Worte, ohne dass die Teilnehmer vielen Beschränkungen unterliegen.

Modbus ermöglicht die Kommunikation zwischen vielen verschiedenen Geräten innerhalb eines Netzwerks, z. B. eines Systems, das Temperatur und Luftfeuchtigkeit misst und die Ergebnisse an einen Computer kommuniziert. Modbus wird häufig eingesetzt, um einen Überwachungscomputer mit einer RTU (Remote Terminal Unit, Fernbedienungsterminal) in SCADA-Systemen zu verbinden. Das Modbus-Protokoll ist in zwei Versionen verfügbar: Modbus seriell und Modbus Ethernet.

Die beiden Modbus-Varianten unterscheiden sich in der Übertragung numerischer Daten und in einigen kleineren Protokolldetails. Modbus RTU überträgt Daten kompakt in Binärform. Modbus ASCII überträgt Daten in lesbarer Form und mit mehr Details. Beide Varianten verwenden die serielle Kommunikation. Das RTU-Format prüft die Befehle/Daten mittels einer CRC-Prüfsumme, wogegen das ASCII-Format eine LRC-Prüfsumme verwendet. Für die RTU-Variante konfigurierte Knoten kommunizieren nicht mit Knoten, die für die ASCII-Variante konfiguriert wurden. Modbus/TCP ähnelt Modbus RTU, aber es überträgt die Protokollpakete in TCP/IP-Datenpaketen.

Es existiert auch eine erweiterte Version, Modbus Plus (Modbus+ oder MB+), diese wurde von Modicon jedoch nicht offengelegt. Sie benötigt einen speziellen Co-Prozessor für eine HDLC-ähnliche Tokenrotation. Diese Variante verwendet Twisted-Pair-Kabel mit einer Übertragungsrate von 1 Mbit/s und enthält Trenntransformatoren an jedem Knoten, wodurch sie übergangs-/flankengesteuert ist und nicht über

Spannung/Stufe gesteuert wird. Für den Anschluss von Modbus Plus an den seriellen Port eines Computers werden spezielle Schnittstellen benötigt.

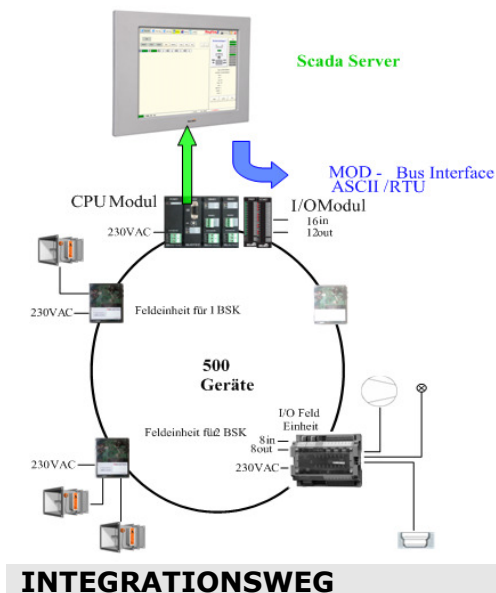
2.1. Adressierung

Jedem Gerät, das über Modbus kommunizieren soll, wird eine eigene Adresse zugewiesen. Jedes Gerät ist in der Lage, einen Modbus-Befehl abzuschicken, üblicherweise wird dies jedoch nur ein Master-Gerät tun. Ein Modbus-Befehl enthält die Modbus-Adresse des Empfängergerätes. Auch wenn andere Geräte diesen Befehl empfangen, wird nur das für den Empfang bestimmte Gerät darauf reagieren. Alle Modbus-Befehle enthalten Prüfinformationen, die sicherstellen, dass ein Befehl fehlerfrei empfangen wurde. Die grundlegenden Modbus-Befehle können eine RTU anweisen, einen Wert in einem ihrer Register zu ändern oder auch einen oder mehrere in ihrem Register enthaltenen Werte anfordern.

3. MODBUS MIT RINGBUS VERWENDEN

Die BELIMO Ringbus-Lösung verwendet die Variante Modbus ASCII, um mit übergeordneten Netzwerken zu kommunizieren. Das bedeutet, dass das zum Netzwerk gehörende Gerät (meist ein PC) als Master fungiert, wogegen die Ringbus-Master-Einheiten (RBCPUs) in der Kommunikation die Rolle eines Slaves innehaben. Die gesamte Steuerung der einzelnen Einheiten wird über zwei Funktionen realisiert, dem Auslesen des Status von Einheiten und dem Abschicken eines Befehls, den eine Einheit ausführen soll.

Hinweis: Es existieren viele freie und kommerzielle Tools für die Kommunikation mit Modbus, die zu Testzwecken heruntergeladen werden können. Eines dieser Tools, für das eine kostenlose Testversion heruntergeladen werden kann, ist das benutzerfreundliche Modbus *Poll* (<http://www.modbustools.com>)



3.1. Setup

Kommunikationsparameter:

- Baudrate: 115200 bps
- Anzahl der DATEN-Bits: 8
- Anzahl der Paritäts-Bits: Keine
- Anzahl der STOPP-Bits: 1
- Kommunikationsmodus: ASCII-Modus über serielle Schnittstelle (RS232)
- Flusssteuerung: Keine

Abb.3.1 zeigt die Einrichtung der Kommunikations-Verbindung mit dem Tool **Modbus Poll**.

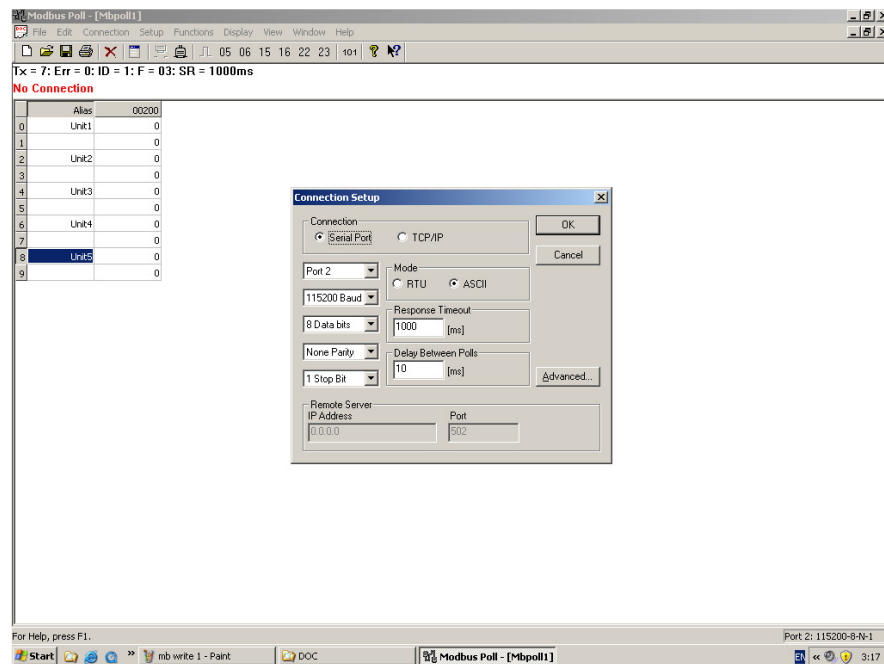


ABBILDUNG 3.1: SETUP MODBUS-KOMMUNIKATION FÜR RINGBUS

3.2. Ringbus-Einheitenstatus auslesen

Die Statusregister aller Einheiten werden im Kommunikationsbereich der RBCPU-Adressregister gesetzt. Unabhängig von ihrem Typ besitzt jede Einheit zwei **"Words"** für verschiedene Stati und Werte. Der Modbus-Funktionstyp ist **03** (*"Read Holding Registers"*). Ringeinheiten beanspruchen den Adressraum von Adresse 200 (0x00C8) an aufwärts, rechts beginnend mit der "höchstwertigen" Ringeinheit (mit Ringadresse 01).

Hinweis: Bitte beachten Sie, dass jede Einheit zwei Worte des Adressraumes für die Speicherung der Statusdaten beansprucht. Daher wird Einheit Nr. 1 die Adresse 200..201 zugewiesen, Einheit Nr. 2 die Adresse 202..203, Einheit Nr. 3 die Adresse 206..207 usw.

Im Detail bedeutet dies:

Ring 1 (Slave Adr. 1) startet bei Adresse 200, für jeden Ring werden 2000 Register reserviert.

Die Stati mehrerer Einheiten können mit einem einzigen Befehl abgefragt werden. Die Ringbus-Antwort auf die Anfrage enthält zwei **"Words"** (der Status-Bits) für jede Einheit.

Die Befehlsparameter sind wie folgt:

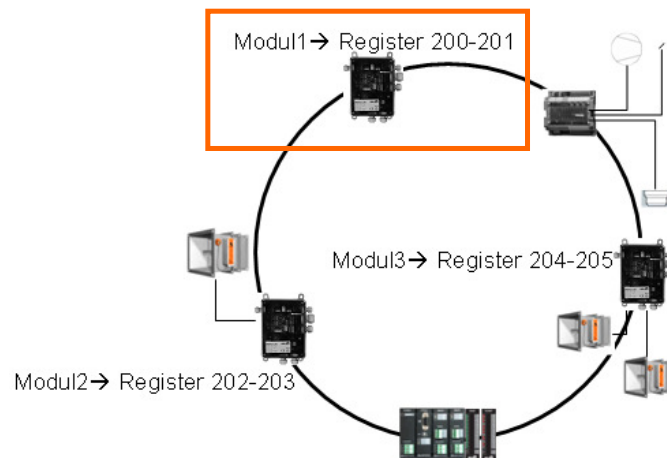
- Modbus-Befehlstyp: 03
- Startadresse: 200 **Für Ring 1 (Slave Adr.1)**
- Startadresse: 200 **Für Ring 2 (Slave Adr.2)**
- Startadresse: 200 **Für Ring 3 (Slave Adr.3)**
- Datenlänge pro Nachricht: 2 **"Words"**

Hinweis: Aufgrund von Beschränkungen des Modbus-Protokolls hinsichtlich des Nachrichtenformates (Nachrichtenlänge) kann eine einzelne Abfrage nur die Stati von maximal 20 Einheiten (40 **"Words"**) enthalten.

Beispiel 1: Auslesen des Status der Einheit mit Ringadresse 01:

Anfrage: (0x) 3A 01 03 00C8 0002 <LRC><CR><LF>

Antwort: (0x) 3A 01 03 04 <U01HI> <U01LO> <LRC><CR><LF>



Beispiel 2: Auslesen der Stati von Einheiten mit den Adressen 05..07
(insgesamt 3 Einheiten):

Anfrage: (0x) 3A 01 03 00D0 0006 <LRC><CR><LF>

Antwort: (0x) 3A 01 03 0C <U05HI> <U05LO> <U06HI>
<U06LO> <U07HI> <U07LO> <LRC><CR><LF>

In diesem Fall ist <LRC> die Prüfsumme der Längsparitätsprüfung (LRC, Longitudinal Redundancy Check), die gemäss der Modbus-Protokollspezifikation berechnet wird.

Hinweis: Weitere Informationen zum Modbus-Kommunikationsprotokoll (ASCII) erhalten Sie in Dokumenten zur Protokollspezifikation unter <http://www.modbus.com>.

Spezielles: Bei DIN und DOM Modulen wird, bei der Adressierung, auf die installierte Reihenfolge geachtet. Wenn nun ein DIN oder DOM Modul in "Slot 3" steckt, so hat dies die Adresse 32. "Slot 4" → 33 und so weiter. Es können maximal 14 DIN oder Dom Module verwendet werden. (Abb. 3.3)



ABBILDUNG 3.3: REGISTER ADRESSE DIN UND DOM MODULE

TABELLE 3.1: BESCHREIBUNG DER STATUS ZONE, BRANDSCHUTZMODUL

NAME	TYP	ADRESSE (WORT.TYP)	BESCHREIBUNG
bus_pow	Bit	0.1	Stromversorgung Bus (0 = stromlos)
ext_pow	Bit	0.2	Externe Stromversorgung
pushbutton	Bit	0.3	Status der Drucktaste
	Bit	0.4	(ungenutzt)
	Bit	0.5	(ungenutzt)
	Bit	0.6	(ungenutzt)
	Bit	0.7	(ungenutzt)
	Bit	0.8	(ungenutzt)
open	Bit	0.9	Klappe geöffnet
close	Bit	0.10	Klappe geschlossen
m_on	Bit	0.11	Motor ist AN
m_off	Bit	0.12	Motor ist AUS
err1	Bit	0.13	Timeout-Fehler
err2	Bit	0.14	Beide Endschalter EIN Fehler
bae	Bit	0.15	BAE Alarm
	Bit	0.16	(ungenutzt)
	Byte	1.0	(ungenutzt)
com_err	Byte	1.1	Zähler für Kommunikationsfehler

TABELLE 3.2: BESCHREIBUNG DER STATUS ZONE, DOPPEL-BRANDSCHUTZMODUL

NAME	TYP	ADRESSE (WORT.TYP)	BESCHREIBUNG
bus_pow	Bit	0.1	Anzeige Bus-Stromversorgung
ext_pow	Bit	0.2	Anzeige externe Stromversorgung
pushbutton	Bit	0.3	Status der Drucktaste Modul
	Bit	0.4	(ungenutzt)
	Bit	0.5	(ungenutzt)
	Bit	0.6	(ungenutzt)
	Bit	0.7	(ungenutzt)
	Bit	0.8	(ungenutzt)
open_1	Bit	0.9	Klappe 1 geöffnet
close_1	Bit	0.10	Klappe 1 geschlossen
m_on_1	Bit	0.11	Motor 1 ist AN
m_off_1	Bit	0.12	Motor 1 ist AUS
err1_1	Bit	0.13	Timeout-Fehler, Antrieb 1
err2_1	Bit	0.14	Beide Endschalter EIN Fehler, Antrieb 1
bae_1	Bit	0.15	BAE Alarm, Antrieb 1
	Bit	0.16	(ungenutzt)
open_2	Bit	1.1	Klappe 2 geöffnet
close_2	Bit	1.2	Klappe 2 geschlossen
m_on_2	Bit	1.3	Motor 2 ist AN
m_off_2	Bit	1.4	Motor 2 ist AUS

**TABELLE 3.2: BESCHREIBUNG DER STATUS ZONE,
DOPPEL-BRANDSCHUTZMODUL**

err1_2	Bit	1.5	Timeout-Fehler, Antrieb 2
err2_2	Bit	1.6	Beide Endschalter EIN Fehler, Antrieb 2
bae_2	Bit	1.7	BAE Alarm, Antrieb 2
	Bit	1.8	(ungenutzt)
com_err	Byte	1.1	Zähler für Kommunikationsfehler

TABELLE 3.3: BESCHREIBUNG DER STATUS ZONE, ENTRAUCHUNGSMODUL

NAME	TYP	ADRESSE (WORT.TYP)	BESCHREIBUNG
bus_pow	Bit	0.1	Anzeige Bus-Stromversorgung
ext_pow	Bit	0.2	Anzeige externe Stromversorgung
pushbutton	Bit	0.3	Status der Drucktaste Modul
	Bit	0.4	(ungenutzt)
	Bit	0.5	(ungenutzt)
	Bit	0.6	(ungenutzt)
	Bit	0.7	(ungenutzt)
	Bit	0.8	(ungenutzt)
open	Bit	0.9	Klappe geöffnet
close	Bit	0.10	Klappe geschlossen
m_on	Bit	0.11	Motor ist AN
m_off	Bit	0.12	Motor ist AUS
err1	Bit	0.13	Timeout-Fehler
err2	Bit	0.14	Beide Endschalter EIN Fehler
bae	Bit	0.15	BAE Alarm
	Bit	0.16	(ungenutzt)
	Byte	1.0	(ungenutzt)
com_err	Byte	1.1	Zähler für Kommunikationsfehler

TABELLE 3.4: BESCHREIBUNG DER STATUS ZONE, FIO-MODUL

NAME	TYP	ADRESSE (WORT.TYP)	BESCHREIBUNG
bus_pow	Bit	0.1	Anzeige Bus-Stromversorgung
ext_pow	Bit	0.2	Anzeige externe Stromversorgung
pushbutton	Bit	0.3	Status der Drucktaste Modul
	Bit	0.4	(ungenutzt)
	Bit	0.5	(ungenutzt)
	Bit	0.6	(ungenutzt)
	Bit	0.7	(ungenutzt)
	Bit	0.8	(ungenutzt)
in1	Bit	0.9	Digitaler Eingang Nr. 1
in2	Bit	0.10	Digitaler Eingang Nr. 2
in3	Bit	0.11	Digitaler Eingang Nr. 3
in4	Bit	0.12	Digitaler Eingang Nr. 4
in5	Bit	0.13	Digitaler Eingang Nr. 5
in6	Bit	0.14	Digitaler Eingang Nr. 6
in7	Bit	0.15	Digitaler Eingang Nr. 7
in8	Bit	0.16	Digitaler Eingang Nr. 8
out1	Bit	1.1	Digitaler Ausgang Nr. 1
out2	Bit	1.2	Digitaler Ausgang Nr. 2
out3	Bit	1.3	Digitaler Ausgang Nr. 3
out4	Bit	1.4	Digitaler Ausgang Nr. 4
out5	Bit	1.5	Digitaler Ausgang Nr. 5

TABELLE 3.4: BESCHREIBUNG DER STATUS ZONE, FIO-MODUL

out6	Bit	1.6	Digitaler Ausgang Nr. 6
out7	Bit	1.7	Digitaler Ausgang Nr. 7
out8	Bit	1.8	Digitaler Ausgang Nr. 8
com_err	Byte	1.1	Zähler für Kommunikationsfehler

TABELLE 3.5: BESCHREIBUNG DER DIN MODULE

NAME	TYP	ADRESSE (WORT.TYP)	BESCHREIBUNG
Input 1	Bit	0.1	Digitaler Eingangskontakt 1
Input 2	Bit	0.2	Digitaler Eingangskontakt 2
Input 3	Bit	0.3	Digitaler Eingangskontakt 3
Input 4	Bit	0.4	Digitaler Eingangskontakt 4
Input 5	Bit	0.5	Digitaler Eingangskontakt 5
Input 6	Bit	0.6	Digitaler Eingangskontakt 6
Input 7	Bit	0.7	Digitaler Eingangskontakt 7
Input 8	Bit	0.8	Digitaler Eingangskontakt 8
Input 9	Bit	0.9	Digitaler Eingangskontakt 9
Input 10	Bit	0.10	Digitaler Eingangskontakt 10
Input 11	Bit	0.11	Digitaler Eingangskontakt 11
Input 12	Bit	0.12	Digitaler Eingangskontakt 12
Input 13	Bit	0.13	Digitaler Eingangskontakt 13
Input 14	Bit	0.14	Digitaler Eingangskontakt 14
Input 15	Bit	0.15	Digitaler Eingangskontakt 15
Input 16	Bit	0.16	Digitaler Eingangskontakt 16

TABELLE 3.6: BESCHREIBUNG DER DOM MODULE

NAME	TYP	ADRESSE (WORT.TYP)	BESCHREIBUNG
Output 1	Bit	0.1	Relaiskontakt 1
Output 2	Bit	0.2	Relaiskontakt 2
Output 3	Bit	0.3	Relaiskontakt 3
Output 4	Bit	0.4	Relaiskontakt 4
Output 5	Bit	0.5	Relaiskontakt 5
Output 6	Bit	0.6	Relaiskontakt 6
Output 7	Bit	0.7	Relaiskontakt 7
Output 8	Bit	0.8	Relaiskontakt 8
Output 9	Bit	0.9	Relaiskontakt 9
Output 10	Bit	0.10	Relaiskontakt 10
Output 11	Bit	0.11	Relaiskontakt 11
Output 12	Bit	0.12	Relaiskontakt 12
	Bit	0.13	(ungenutzt)
	Bit	0.14	(ungenutzt)
	Bit	0.15	(ungenutzt)
	Bit	0.16	(ungenutzt)

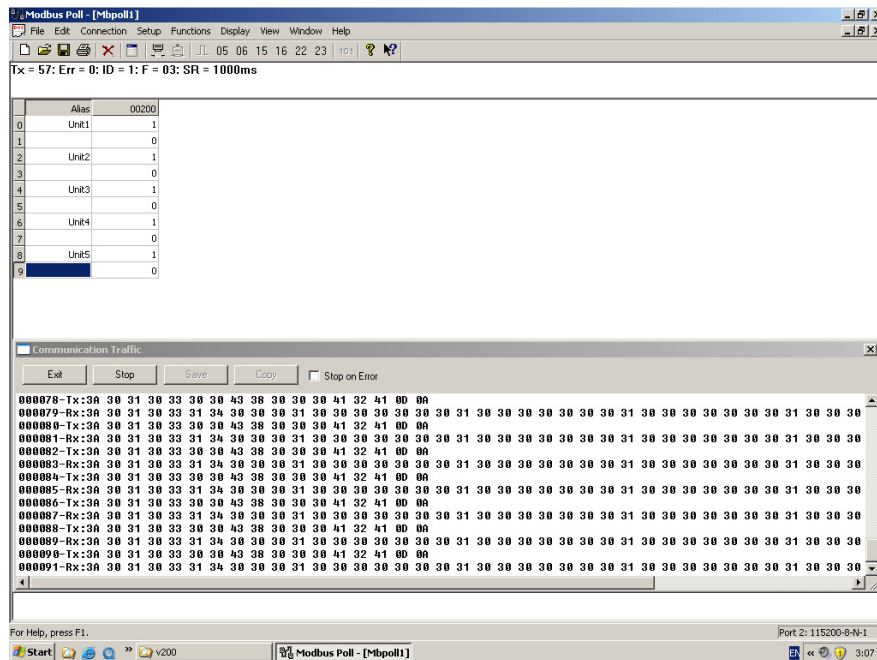


ABBILDUNG 3.4: STATI VON RINGBUS-EINHEITEN AUSLESEN

3.3. Befehle an Ringbus Module senden

Befehle können an Ringbus Module gesendet werden, indem "**Data Words**" in die Kommunikationsbereich der RBCPU's geschrieben werden, beginnend bei Adresse 100 (0x0064). Zu diesem Zweck wird der Modbus-Funktionscode Nr. 16 (Write Multiple Registers) verwendet.

Vorsicht:

Beim Beschreiben der Register / Ansteuern der Module über ein übergeordnetes System, müssen die Länderspezifischen Normen eingehalten werden.

In einzelnen Ländern ist das Beschreiben/ Ansteuern der Module nicht erlaubt !

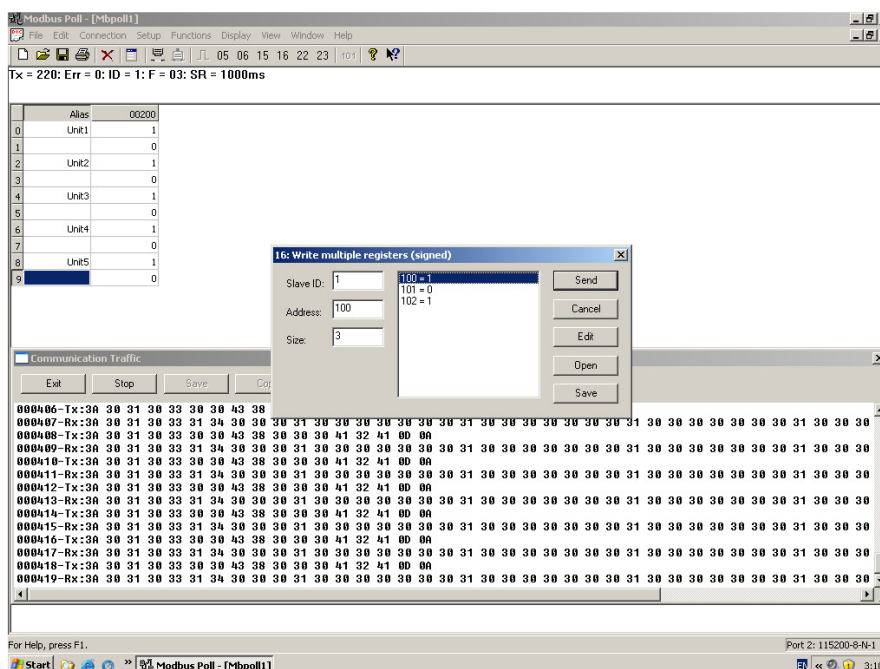


ABBILDUNG 3.5: BEFEHLE AN RINGBUS-EINHEITEN SENDEN

Hinweis: Es kann in einem einzelnen Modbus-Telegramm nur ein Befehl pro Einheit abgeschickt werden.

Die Befehlsparameter sind wie folgt:

- Modbus-Befehlstyp: 16 (0x10)
- Startadresse: 100 (0x0064)
- Nachrichtendatenlänge pro Einheit : 3 "**Data Words**"

Die Befehlssyntax wird in Tabelle 3.6 erläutert.

TABELLE 3.6: EINHEITENBEFEHLS-SYNTAX DES BEFEHLS "EXECUTE COMMAND"

ADRESSE	BESCHREIBUNG	WERT
100 (0x64)	Befehlsbezeichnung	1
101 (0x65)	Einheitenindex	0.. <i>max_unit_perRing</i> - 1
102	Befehlscode	(siehe folgende Tabelle)

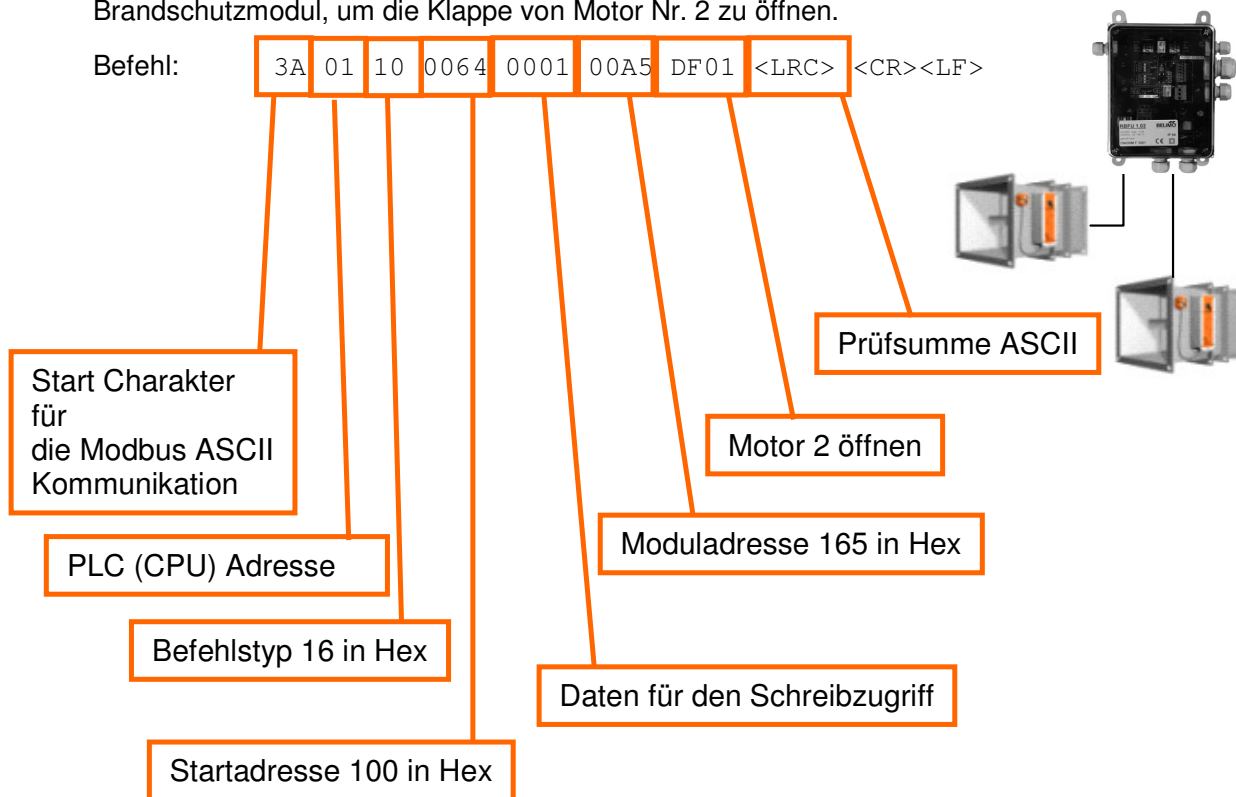
TABELLE 3.7: BEFEHLSCODES

MODULTYP	WERT	BESCHREIBUNG
Brandschutzmodul	0x0001	öffnen
	0x0002	schliessen
	0x0003	testen
Entrauchungsmodul	0x0001	öffnen
	0x0002	schliessen
	0x0003	testen
Doppel- Brandschutzmodul	0xFD01	Motor 1 öffnen
	0xFD02	Motor 1 schliessen
	0xFD03	Motor 1 testen
	0xDF01	Motor 2 öffnen
	0xDF02	Motor 2 schliessen
	0xDF03	Motor 2 testen
Feld-I/O-Modul (nur Ausgänge)	0x00mm	<i>mm</i> = Bi(R)aske für Ausgänge 0x01 = Ausgang 1, 0x02 = Ausgang 2, 0x04 = Ausgang 3, ... 0x80 = Ausgang 8;

Beispiel: Senden eines Befehls an das Modul mit der Adresse 165 (0x00A5), ein Doppel-Brandschutzmodul, um die Klappe von Motor Nr. 2 zu öffnen.

Befehl:

3A 01 10 0064 0001 00A5 DF01 <LRC> <CR><LF>



Copyright

Ringbus, **RingView** und **RingView Lite** sind eingetragene Marken der *Bustec GmbH*. **Windows** ist eine eingetragene Marke der Firma *Microsoft*. **Linux** ist eine eingetragene Marke von *Linus Torvalds*. **Java** ist eine eingetragene Marke der Firma *Sun Microsystems*. **Modbus** ist eine eingetragene Marke von *Modbus-IDA*. Alle anderen in diesem Dokument erwähnten Markenzeichen sind Eigentum der jeweiligen Inhaber.

Rechtliche Hinweise

Dieses Dokument unterliegt dem Copyright © 2009, BELIMO Automation. Alle Rechte vorbehalten. Jegliche Art der Vervielfältigung ist nur zum Zwecke der Datensicherung gestattet.

DIESES DOKUMENT UND ZUKÜNFTIGE VERSIONEN DIESES DOKUMENTES WERDEN UNTER FOLGENDEN BEDINGUNGEN ZUR VERFÜGUNG GESTELLT:

1. DIESES DOKUMENT WIRD "WIE VORLIEGEND" GELIEFERT, OHNE GARANTIE IRGENDWEIER ART, SOWOHL AUSDRÜCKLICH GENANNT ALS AUCH ANGEDEUTET. DIES BEZIEHT SICH AUCH OHNE EINSCHRÄNKUNG AUF GARANTIE, DASS DIESES DOKUMENT ODER VERÄNDERTE FASSUNGEN DIESES DOKUMENTS FREI VON HANDELSDEFEKTE, FÜR EINEN BESTIMMTEN ZWECK GEEIGNET IST ODER DASS ES KEINE RECHTE DRITTER VERLETZT. DAS VOLLE RISIKO HINSICHTLICH QUALITÄT, GENAUIGKEIT UND LEISTUNG DES DOKUMENTS ODER VERÄNDERTEN FASSUNGEN DES DOKUMENTS LIEGT BEI IHNEN. SOLLTE EIN DOKUMENT ODER EINE VERÄNDERTE FASSUNG DAVON FEHLER IRGENDWEIER ART BEIHALTEN, TRAGEN SIE (NICHT DER URSPRUNGSUTOR, DER AUTOR ODER EIN MITWIRKENDER) DIE KOSTEN FÜR NOTWENDIGE DIENSTLEISTUNGEN, REPARATUREN ODER FEHLERKORREKTUREN. DIESER HAFTUNGSAUSSCHLUSS IST EIN ESSENZIELLER TEIL DIESER LIZENZ. DIE VERWENDUNG EINES

DOKUMENTS ODER EINER VERÄNDERTEN VERSION DES DOKUMENTS IST NICHT GESTATTET AUSSER UNTER BEACHTUNG DIESES HAFTUNGSAUSSCHLUSSES UND

2. UNTER KEINEN UMSTÄNDEN UND AUF BASIS KEINER RECHTSGRUNDLAGE, UNABHÄNGIG DAVON, OB DURCH UNERLAUBTE HANDLUNGEN (EINSCHLIESSLICH FAHRLÄSSIGKEIT), VERTRAG ODER ANDERWEITIG, KANN DER AUTOR, URSPRUNGSAUTOR, EIN MITWIRKENDER ODER EIN VERTRIEBSPARTNER DIESES DOKUMENTS ODER EINER VERÄNDERTEN FASSUNG DES DOKUMENTS ODER EIN ZULIEFERER EINER DIESER PARTEIEN, HAFTBAR GEMACHT WERDEN FÜR DIREKTE, INDIREKTE, SPEZIELLE, VERSEHENTLICHE ODER FOLGESCHÄDEN JEDLICHER ART, EINSCHLIESSLICH UND OHNE EINSCHRÄNKUNGEN SCHÄDEN DURCH VERLUST VON KULANZ, ARBEITSAUSFALL, COMPUTERVERSAGEN ODER COMPUTERFEHLFUNKTIONEN ODER ALLE ANDEREN SCHÄDEN ODER VERLUSTE, DIE SICH AUS ODER IN VERBINDUNG MIT DER VERWENDUNG DES DOKUMENTS UND VERÄNDERTER FASSUNGEN DES DOKUMENTS ERGEBEN, AUCH WENN DIE OBEN GENANNTE PARTeien ÜBER DIE MÖGLICHKEIT SOLCHER SCHÄDEN INFORMIERT WAREN.