



**Motion Protocol CPR-CAN-V2  
für  
Closed Loop Motorcontroller**



---

## Inhaltsverzeichnis

<b>1</b>	<b>Dieses Dokument</b>	<b>2</b>
1.1	Kontakt . . . . .	2
1.2	Zielgruppe und Qualifikation . . . . .	3
1.3	Verwendete Symbole . . . . .	3
<b>2</b>	<b>Einleitung</b>	<b>4</b>
2.1	Kurzbeschreibung des Protokolls . . . . .	4
2.2	Ablauf der Kommunikation . . . . .	4
2.3	Kommunikationsparameter . . . . .	5
2.4	Konfigurationssoftware ModuleCtrl . . . . .	5
2.5	Demo-Software mit Sourcen . . . . .	5
<b>3</b>	<b>CAN-Nachrichten</b>	<b>7</b>
3.1	Bewegungsbefehle . . . . .	7
3.1.1	Position CMD 0x14 . . . . .	7
3.1.2	Velocity CMD 16bit 0x25 . . . . .	7
3.1.3	Torque CMD 0x16 . . . . .	8
3.2	Antwort auf Bewegungsbefehle . . . . .	9
3.3	Process CMD 0x01 . . . . .	11
3.3.1	Reset Error 0x06 . . . . .	11
3.3.2	Position auf 0 setzen 0x08 . . . . .	11
3.3.3	Enable Motor 0x09 . . . . .	12
3.3.4	Disable Motor 0x0A . . . . .	12
3.3.5	Referenzierung 0x0B . . . . .	13
3.3.6	Rotor Ausrichtung 0x0C . . . . .	14
3.3.7	Ping 0xCC . . . . .	14
3.3.8	Freischaltung EEPROM 0xCD . . . . .	15
3.4	Parameter V2 0x94 & 0x96 . . . . .	16
3.4.1	Parameter speichern 0x94 . . . . .	16
3.4.2	Parameter lesen 0x96 . . . . .	17
3.5	Besondere CAN-Nachrichten . . . . .	18
3.5.1	Startupnachricht . . . . .	18
3.5.2	Umgebungsparameter 0x12 . . . . .	18
3.5.3	Erweiterte Fehlernachrichten . . . . .	19
3.5.4	Encoder Rebel V3 Position . . . . .	22
3.5.5	Abtriebsencoder Rebel V2 Position . . . . .	22

---

## 1 Dieses Dokument

### 1.1 Kontakt

Commonplace Robotics GmbH  
Gewerbepark 9-11  
Im Innovationsforum  
D-49143 Bissendorf

Tel.: +49(0)5402 / 968929-0  
Fax: +49(0)5402 / 968929-9  
E-Mail: [info@commonplacrobotics.de](mailto:info@commonplacrobotics.de)

Internet: <https://cpr-robots.com>



## 1.2 Zielgruppe und Qualifikation

Diese Protokolldokumentation richtet sich an technisch geschulte Fachkräfte wie:

- Programmierer
- Entwicklungsingenieure

Erfahrungen mit folgenden Themen wird vorausgesetzt:

- Hardwarenahe Programmierung bspw. mit C++
- Kenntnis des CAN-Feldbusses
- ESD-Vorgehensweisen bei der Arbeit mit Elektronik

Diese Dokumentation ergänzt die Produktdokumentation der Motorcontroller. Die Kenntnis dieser Dokumentation wird vorausgesetzt.

## 1.3 Verwendete Symbole

Alle Hinweise in diesem Dokument folgen einer einheitlichen Form und sind gemäß nachfolgenden Klassen gegliedert.



**Der Hinweis WARNUNG macht den Leser auf mögliche gefährliche Situationen aufmerksam.**

Die Missachtung einer Warnung kann **möglicherweise** zu mittelschweren Verletzungen des Benutzers führen.

- Innerhalb einer Warnung beschreibt dies Möglichkeiten zur Vermeidung von Gefahren.



**Dieser Hinweis kennzeichnet mögliche Fehlbedienungen des Produktes.**

Die Missachtung dieses Hinweises kann die Funktionalität des Produktes einschränken.



In dieser Box befinden sich ergänzende Hinweise, sowie Tipps und Tricks.

## 2 Einleitung

### 2.1 Kurzbeschreibung des Protokolls

Das CPR-CAN-V2 Protokoll wird zur Kommunikation zwischen einer Steuerung und den Antriebsmodulen verwendet. Die Steuerung schickt über den CAN-Feldbus Positions-Sollwerte oder Geschwindigkeiten an die Motorcontroller, diese antworten mit ihrem aktuellen Status.

Die Berechnung von Beschleunigungsrampen und die Interpolation zwischen Zielpunkten erfolgt dabei in der Steuerung, nicht im Motorcontroller. Die Steuerung sendet in kurzen Zeitabständen von 10 bis 50 ms die Zielposition an die Motorcontroller.

Der Betrieb ist damit vergleichbar mit dem CANopen-Betriebsmodus IPO oder CSP. Ein Betriebsmodus wie CANopen ProfilePosition wird nicht unterstützt.

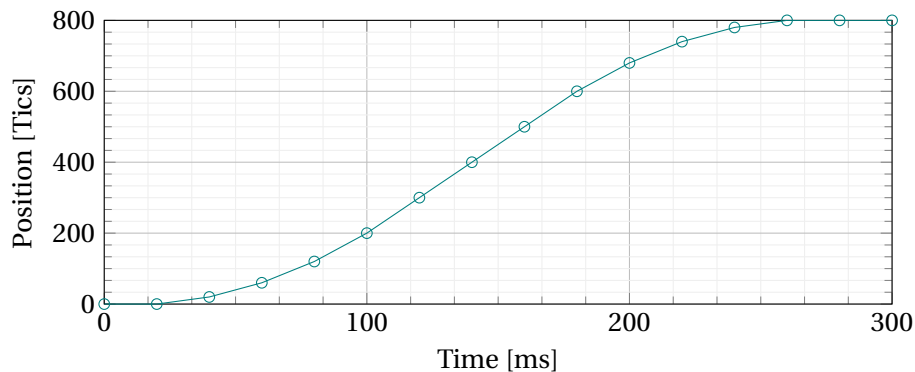
Zusätzlich zur Übermittlung der Positions- oder Geschwindigkeitsvorgaben können die Controller über das Protokoll parametrisiert werden.

### 2.2 Ablauf der Kommunikation

Dieses Beispiel soll den Ablauf der Kommunikation im Positionierbetrieb darstellen.

1. Starten des Motorcontrollers
2. Start einer zyklisch aufgerufenen Funktion die alle 20 ms eine Positionsvorgabe schickt (Kapitel 3.1.1)
3. Angleichen der Positionen auf Motorcontroller und Steuerungssoftware. Dazu gibt es zwei Möglichkeiten:
  - Reset der aktuellen Position auf 0 (Kapitel 3.3.2). Initialisierung der Sollposition in der Steuerungssoftware ebenfalls mit 0.  
Diese Variante ist für die ersten Tests empfehlenswert.
  - Auslesen der aktuellen Motorposition aus einer Antwortnachricht (Kapitel 3.2) und Übernahme dieser Position als aktuelle Sollposition in die Steuerungssoftware.  
Diese Variante wird für reale Anwendungen genutzt.
4. Fehler auf der Achse zurücksetzen. (Kapitel 3.3.1)
5. Achse aktivieren (Kapitel 3.3.3) so dass das Fehlerbyte in der Antwort 0x00 zeigt (Kapitel 3.2).
6. Langsames Verändern der Zielposition in der zyklisch gesendeten Positionsvorgabe. Die Achse folgt nun der Vorgabe.

Die folgende Grafik zeigt die Bewegung von 0 auf 800 Tics. Die Interpolation mit Beschleunigungs- und Bremsrampen erfolgt in der Steuerung, diese sendet die, durch Punkte markierten, Zielpositionen zyklisch zum Motorcontroller.



### 2.3 Kommunikationsparameter

CAN-Baudrate                      500kBaud  
Kommunikationsfrequenz    20 bis 100 hz

### 2.4 Konfigurationssoftware ModuleCtrl

Für eine einfache Ansteuerung aller Funktionen der Achsen steht dem Anwender die Konfigurationssoftware ModuleControl zur Verfügung. Diese unterstützt den Bootloader, sowie die Parameterschnittstelle.

[https://wiki.cpr-robots.com/index.php/Config\\_Software\\_ModuleCtrl](https://wiki.cpr-robots.com/index.php/Config_Software_ModuleCtrl)



### 2.5 Demo-Software mit Sourcen

Für einen schnellen Start bei der Entwicklung einer eigenen Ansteuerung steht auf GitHub ein Beispielprogramm zur Verfügung:

<https://github.com/CommonplaceRobotics/CANV2ProtocolDemoClient>



Enthalten ist ein VisualStudio2019-Projekt in CSharp das sich mit einem Motorcontroller verbindet und ihn ansteuert. Es sind alle Source-Dateien enthalten, diese können als Startpunkt für eine eigene Entwicklung genutzt werden.

Zum Ausführen des Codes und zum Testen eigener Anwendungen benötigen Sie folgende Komponenten:

1. Visual Studio 2019 oder später
2. PCAN-USB CAN to USB Interface von PEAK-System Technik GmbH

- 
3. Motorcontroller mit Motor
  4. Anschlusskabel je nach Motorcontroller



### 3 CAN-Nachrichten

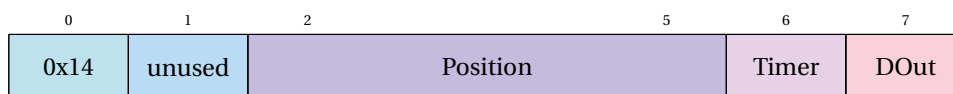
#### 3.1 Bewegungsbefehle

Bewegungsbefehle fallen unter die Kategorie der zyklischen Befehle. Diese Befehle müssen in einer festen Zeit immer wieder vom Bewegungsmaster an den Motorcontroller gesendet werden. Der Basi- stakt liegt bei 50ms. Hält der Bewegungsmaster diese Zeit nicht ein und überschreitet eine vorgege- bene Zeit, so geht der Motorcontroller in einen Fehlerzustand. Beim Wechseln zwischen zwei Bewe- gungsbefehlen unterbricht die Achse die Ausführung und stoppt die Bewegung. Erst nach Reset und Enable ist eine erneute Bewegung im neuen Modus möglich.

##### 3.1.1 Position CMD 0x14

Der Position CMD versetzt die Achse in den Positionniermodus. Dieser Modus ist vergleichbar mit dem CANopen-Betriebsmodus IPO oder CSP. Die Vorgabe der Position erfolgt in Encodertics. Der Motorcontroller versucht die Zielposition so schnell wie möglich zu erreichen, es wird keine Bahn mit Sollgeschwindigkeit und Beschleunigungsrampen berechnet. Liegt die Zielposition zu weit von der aktuellen Position entfernt kann zu unerwartetem Verhalten oder einer Abschaltung führen.

###### 1. Nachricht zum Motorcontroller



Position 32-Bit Position des Motors in [Tics]

Timer Wert zur Überprüfung der richtigen Reihenfolge

DOut Wenn der Motorcontroller über Ausgänge verfügt können diese hier gesetzt werden.

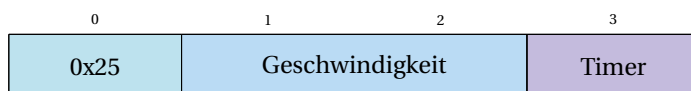
###### 2. Antwort vom Motorcontroller

Standardantwort CANV2 (Kapitel 3.2)

##### 3.1.2 Velocity CMD 16bit 0x25

Dieser Modus ist vergleichbar mit dem CANopen-Betriebsmodus ProfileVelocity.

###### 1. Nachricht zum Motorcontroller



Geschwindigkeit Vorgabe der Geschwindigkeit mit 16bit in [RPM] als int16.

---

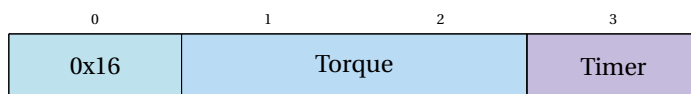
Timer                      Wert zur Überprüfung der richtigen Reihenfolge

2. Antwort vom Motorcontroller  
Standardantwort CANV2 (Kapitel 3.2)

### 3.1.3 Torque CMD 0x16

Dieser Modus ist vergleichbar mit dem CANopen-Betriebsmodus ProfileTorque.

1. Nachricht zum Motorcontroller



Torque    Vorgabe der Des Gewünschten Drehmoments in Engineeringunits zwischen  
             [-1024-1024]

Timer     Wert zur Überprüfung der richtigen Reihenfolge

2. Antwort vom Motorcontroller  
Standardantwort CANV2 (Kapitel 3.2)

### 3.2 Antwort auf Bewegungsbefehle

Die Motorcontroller antworten auf alle Bewegungsbefehle mit einer einheitlichen Antwort.

CAN-ID = BoardID + 1



errorCode Fehler Byte siehe unten für weitere Informationen

Position 32-Bit aktuelle Achsposition in [Tics]

Strom Aktueller RMS Strom in [mA]

DIn + Flags Die Eingänge der Motorcontroller, sowie die Flags sind unten erklärt

Der Motorcontroller sendet in jeder CANV2 Antwort ein Errorbyte mit. Diese kann verschiedene Fehlerzustände definieren.



TEMP	Übertemperatur	Die Temperatur des Motorcontrollers oder des Motors liegt über dem definierten Wert in den Parametern.
ESTOP	Not-Aus / keine Spannung	Die Spannung am Motorcontroller ist unter dem eingestellten Grenzwert. Die kann auf eine defekte Sicherung oder Not-Aus schließen.
MNE	Motor nicht aktiviert	Die Bewegung des Motors ist nicht freigegeben, dieser befindet sich nicht in Regelung.
COM	Kommunikations Ausfall	Der Motorcontroller benötigt in regelmäßigen CAN-Nachrichten. Ist der Abstand der Nachrichten zu groß oder die Nachrichten bleiben aus stoppt der Motorcontroller die Bewegung.
LAG	Schleppfehler	Der Motorcontroller überwacht den Schleppfehler, ist dieser größer als der eingestellte Wert in den Parameter, so stoppt der Motorcontroller die Bewegung.
ENC	Encoder Fehler	Der Motorcontroller hat einen Fehler des Encoders festgestellt. Fehler können sowohl durch den Motor oder den Abtriebsencoder ausgelöst werden.
OC	Überstrom	Der RMS Strom im Motorcontroller lag über dem zulässigen Wert in den Parametern.

### DRV Treiberfehler

Ein Treiberfehler kann verschiedene Ursachen haben. Eine mögliche Ursache ist die Überschreitung der maximalen Geschwindigkeit aus den Parametern. Bei den Closedloop Motorcontrollern tritt der Fehler zudem bei Problemen mit der initialen Rotorposition auf.

Das Bytefeld für die Eingänge, sowie die Referenzierungsflags der Motorcontroller ist in den Nibble unterschieden. Das Low-Nibble beinhaltet die Eingänge, das High-Nibble die Referenzierungsflags.

7	6	5	4	3	0
Flag referenziert	Flag aligned	unused	Flag axis ready	DIn	

Flag referenziert	Die Achse ist referenziert
Flag aligned	Das Rotorfeld ist ausgerichtet
Flag axis ready	Die Achse ist bereit für Bewegungen
DIn	Zustand der digitalen Eingänge am Board

### 3.3 Process CMD 0x01

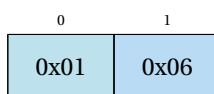
In diese Kategorie fallen Steueranweisungen, welche nicht zyklisch an das Board gesendet werden.

#### 3.3.1 Reset Error 0x06

Diese Nachricht setzt alle Fehler auf dem Board zurück.

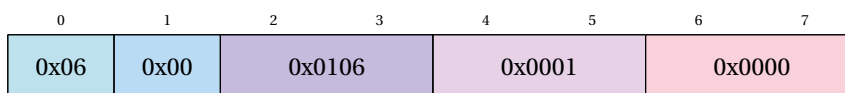
Je nach Boardversion verfügt das Board über irreversible Fehler, diese lassen sich nicht zurücksetzen. Ein Beispiel dafür ist ein Fehler in der Motorverdrahtung.

1. Nachricht zum Motorcontroller



2. Antwort vom Motorcontroller

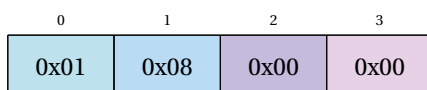
CAN-ID = BoardID + 2



#### 3.3.2 Position auf 0 setzen 0x08

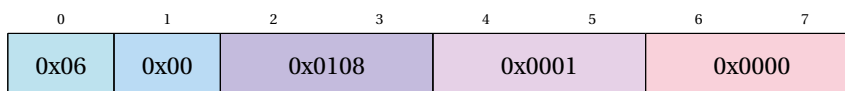
Diese Nachricht muss 2 mal innerhalb von 20 ms an den Controller gesendet werden. Dieser Befehl setzt die Aktuelle auf 0. Der Controller geht im Anschluss in einen Fehlerzustand.

1. 1. Nachricht zum Motorcontroller



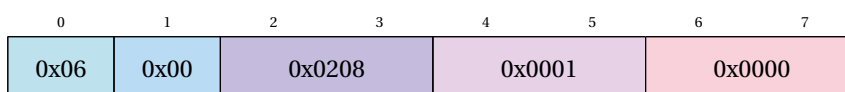
2. 1. Antwort vom Motorcontroller CMD erhalten

CAN-ID = BoardID + 2



3. 2. Antwort vom Motorcontroller CMD ok

Wird nur versendet wenn der Befehl ok ist. CAN-ID = BoardID + 2



## 4. 2. Nachricht zum Motorcontroller

0	1	2	3
0x01	0x08	0x00	0x00

## 5. 3. Antwort vom Motorcontroller CMD erhalten

CAN-ID = BoardID + 2

0	1	2	3	4	5	6	7
0x06	0x00	0x0108	0x0001	0x0000			

## 6. 4. Antwort vom Motorcontroller CMD ok

Wird nur versendet wenn der Befehl ok ist. Nach dieser Antwort hat der Motorcontroller die Position auf 0 gesetzt. CAN-ID = BoardID + 2

0	1	2	3	4	5	6	7
0x06	0x00	0x0208	0x0002	0x0000			

**3.3.3 Enable Motor 0x09**

Aktiviert den Motor. Nur möglich wenn kein Fehler anliegt.

## 1. Nachricht zum Motorcontroller

0	1
0x01	0x09

## 2. Antwort vom Motorcontroller

CAN-ID = BoardID + 2

0	1	2	3	4	5	6	7
0x06	0x00	0x0109	0x0001	0x0000			

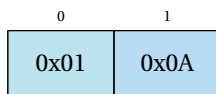
**3.3.4 Disable Motor 0x0A**

Motor deaktivieren. Die Leistungsendstufe schaltet den Strom zum Motor ab.

**kein sicherer Stopp**

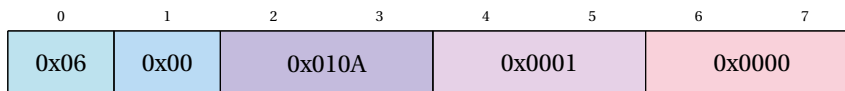
Diese Funktion stellt keinen sicheren Stopp dar. Es ist möglich, dass der Motor weiter mit Spannung versorgt wird.

## 1. Nachricht zum Motorcontroller



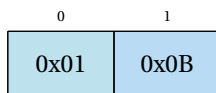
## 2. Antwort vom Motorcontroller

CAN-ID = BoardID + 2

**3.3.5 Referenzierung 0x0B**

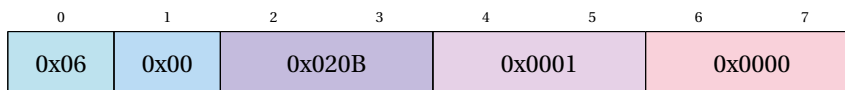
Anforderung der Referenzierung. Die Nachricht muss mehrfach an den Motorcontroller gesendet werden.

## 1. 1. Nachricht zum Motorcontroller

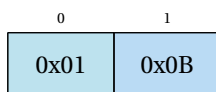


## 2. 1. Antwort vom Motorcontroller

CAN-ID = BoardID + 2

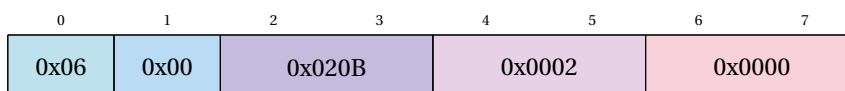


## 3. 2. Nachricht zum Motorcontroller



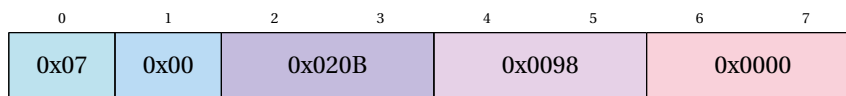
## 4. 2. Antwort vom Motorcontroller

CAN-ID = BoardID + 2



## 5. Fehler vom Motorcontroller Referenzierung aktiv. Die Referenzierung der Achse ist bereits aktiv.

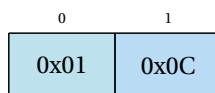
CAN-ID = BoardID + 2



### 3.3.6 Rotor Ausrichtung 0x0C

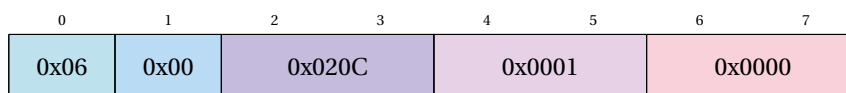
Für den Betrieb von Closedloop Motorcontrollern ist es notwendig, dass diese die initiale Rotorausrichtung kennen. Diese kann durch nachfolgenden Befehl neu eingemessen werden. Dieser Befehl ist ebenfalls 2 mal zusenden.

1. Nachricht zum Motorcontroller

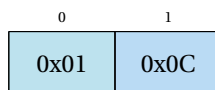


2. 1. Antwort vom Motorcontroller

CAN-ID = BoardID + 2

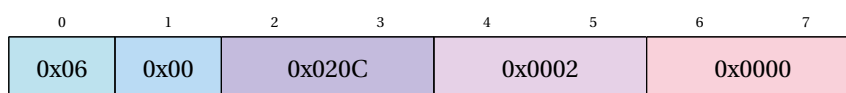


3. 2. Nachricht zum Motorcontroller



4. 2. Antwort vom Motorcontroller

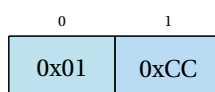
CAN-ID = BoardID + 2



### 3.3.7 Ping 0xCC

Der Controller antwortet auf diese Nachricht mit der Startupnachricht.

1. Nachricht zum Motorcontroller



2. Antwort vom Motorcontroller

Die versendete CAN-Nachricht entspricht der Startupnachricht. (Kapitel 3.5.1)



### 3.3.8 Freischaltung EEPROM 0xCD

Bevor Parameter auf dem Motorcontroller gespeichert werden können muss mit dieser Nachricht der EEPROM freigeschaltet werden. Der Motorcontroller antwortet auf diese Nachricht nicht.

0	1
0x01	0xCD

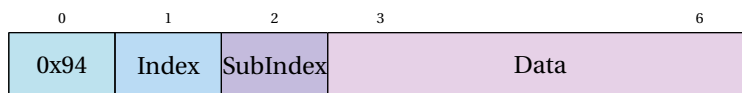
### 3.4 Parameter V2 0x94 & 0x96

Für die Closedloop Motorcontroller ist die folgende Parameterschnittstelle vorhanden. Diese greift auf eine Struktur aus Index und Subindex zurück.

Die Beschreibung der entsprechenden Parameter mit Index und Subindex sind in der Produktdokumentation der Motorcontroller zu finden.

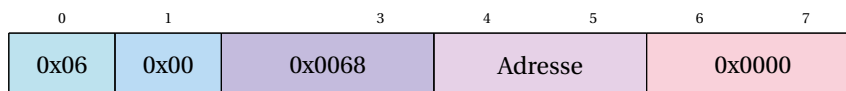
#### 3.4.1 Parameter speichern 0x94

1. Freischalten des EEPROM (Kapitel 3.3.8)
2. Nachricht zum Motorcontroller



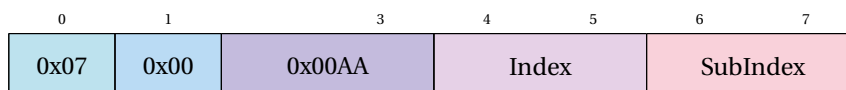
Index      Kategorie des Parameters  
 SubIndex    Parameter in der Kategorie  
 Data        32bit Wert der Daten

3. Nachricht vom Motorcontroller: Übertragung ok  
 CAN-ID = BoardID + 2



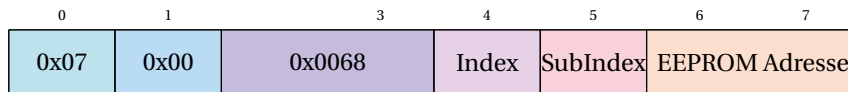
Adresse    Berechnete EEPROM Adresse

4. Fehler vom Motorcontroller: EEPROM nicht freigeschaltet  
 Der EEPROM wurde zuvor nicht freigeschaltet, siehe Kapitel 3.3.8  
 CAN-ID = BoardID + 2



Index      Übertragener Index  
 SubIndex   Übertragener SubIndex

5. Fehler vom Motorcontroller: Nicht gespeichert  
 Die Daten konnten nicht im EEPROM abgelegt werden.  
 CAN-ID = BoardID + 2



Index                      Übertragener Index  
 SubIndex                Übertragener SubIndex  
 EEPROM Adresse      Zu schreibende EEPROM Adresse

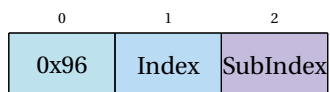
### 3.4.2 Parameter lesen 0x96



#### Unterbrechung der zyklischen Nachrichten

Für das Übertragen der Parameter müssen die zyklischen CAN-Nachrichten unterbrochen werden. Die Antwort zum Lesen der Parameter und die Standard Antwort (Kapitel 3.2) teilen sich die die Antwort-ID BoardID + 1.

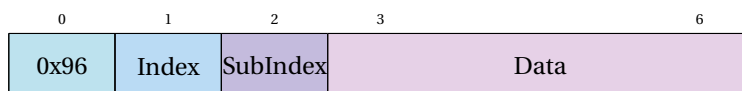
#### 1. Nachricht zum Motorcontroller



Index            Kategorie des Parameters  
 SubIndex      Parameter in der Kategorie

#### 2. Nachricht vom Motorcontroller: Übertragung ok

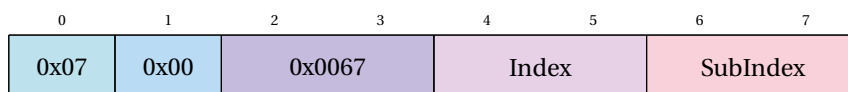
CAN-ID = BoardID + 1



Index            Übertragener Index  
 SubIndex      Übertragener SubIndex  
 Data            32bit Wert der Daten

#### 3. Fehler vom Motorcontroller: Fehler beim Lesen

CAN-ID = BoardId + 2



Index      Übertragener Index  
SubIndex   Übertragener SubIndex

### 3.5 Besondere CAN-Nachrichten

Diese Sektion beinhaltet die Beschreibung von besonderen CAN-Nachrichten, bspw. versenden die Motorcontroller zyklisch, ohne Nachfrage, verschiedene Zustandsinformationen.

#### 3.5.1 Startupnachricht

Beim Starten der Motorcontroller versenden diese eine definierte Nachricht. Diese Nachricht hilft den Motorcontroller zu identifizieren.

CAN-ID = BoardID + 2

0	4	5	6	7
0x01020304				Restart cause
				HW- Ident
				Version

Restart cause    Bitfeld für den Auslöser des Neustart

HW-Ident        Identifikation des Boards

Version         Major und Minor Version der Firmware

Für die Identifikations des Boards sind folgende Werte implementiert:

DIN-RAIL CL BLDC	0x50	Closedloop BLDC-Motorcontroller für Hutschienenmontage mit CPR Rückwandbus
DIN-RAIL CL STEPPER	0x51	Closedloop Schrittmotorcontroller für Hutschienenmontage mit CPR Rückwandbus
ON-AXIS CL BLDC	0x52	Closedloop integrierter Motorcontroller für die Montage in einer Achse
DIN-RAIL CL DC	0x53	Closedloop DC Motorcontroller für Hutschienenmontage mit CPR Rückwandbus
dsPIC33EP128GM Bootloader	0x89	Bootloader für Motorcontroller dsPIC33
dsPIC33EP256GM Bootloader	0x8A	Bootloader für Motorcontroller dsPIC33
dsPIC33EP128MC Bootloader	0x8B	Bootloader für Motorcontroller dsPIC33
dsPIC33EP256MC Bootloader	0x8C	Bootloader für Motorcontroller dsPIC33

#### 3.5.2 Umgebungsparameter 0x12

Der Motorcontroller versendet einmal pro Sekunde eine Nachricht mit der aktuellen Spannung, sowie der Motor- und Boardtemperatur.

CAN-ID = BoardID + 3

0	1	2	3	4	5	6	7
0x1200		Voltage		Temp Motor		Temp Board	

Voltage      Aktuelle Spannung am Motorcontroller[mV]

Temp Motor    Motortemperatur [m°C]

Temp Board    Temperatur Motorcontroller [m°C]

### 3.5.3 Erweiterte Fehlerinformationen

Der Motorcontroller versendet einmal pro Sekunde eine Nachricht mit detaillierten Fehlerinformationen. Für eine Auswertung der einzelnen Bytes nutzen Sie bitte die Konfigurationssoftware ModuleControl.

CAN-ID = BoardID + 2

0	1	2	3	4	5	6	7
0xE0	Motor Error	ADC Error	Enc V2 Error	Control Error	Enc V3 Gear	Enc V3 Motor	reserved

Motor Error    Byte mit Fehlern des Motors

ADC Error      Byte mit Fehlern des AD-Wandlers

Enc V2 Error    Byte mit Rebel V2 Abtriebsencoder Fehlern

Control Error    Byte mit Steuerungsfehlern

Enc V3 Gear     Byte mit Rebel V3 Abtriebsencoder Fehlern

Enc V3 Motor    Byte mit Rebel V3 Antriebssencoder Fehlern

Auch die Rebel Base versendet einmal pro Sekunde eine Nachricht mit detaillierten Fehlerinformationen. Für eine Auswertung der einzelnen Bytes nutzen Sie bitte die Konfigurationssoftware ModuleControl.

CAN-ID = 2

0	1	2	3	7
0xE1	Power Error	Voltage Error	reserved	

Power Error    Byte mit Fehlern der Stromversorgung der Rebel Subsysteme

Voltage Error    Byte mit erkannten Spannungsabweichungen vom erwarteten Bereich

#### 3.5.3.1 Motor Error

Die Motorcontroller überwachen verschiedene Parameter des Motors. Für die Überwachung der Ströme sind zwei Werte entscheidend, der RMS-Wert gemäß den Parametern und ein festprogrammierter Wert für einzelne Phasen. In den regulären Nachrichten tauchen beide Fehler als OC auf.

7	3	2	1	0
reserved			OC RMS	OC Phase
				Motor n.c.

Motor n.c. Motor nicht angeschlossen, aktuell nicht verwendet

OC Phase Überstrom in einer Phase

OC RMS Überstrom in der RMS Messung

### 3.5.3.2 ADC Error

Da der richtige Offset unerlässlich für den Betrieb der Motorcontroller ist, prüft dieser vorher ob es hier zu einem defekt gekommen ist.

7	1	0
reserved		ADC Offset

ADC Offset Fehlerhafter Offset in der Strommessung

### 3.5.3.3 Control Error

Durch die verschiedenen Ansteuerungsmethodiken der Closedloop Motorcontroller kann es im Bereich der Motorsteuerung ebenfalls zu Fehlern kommen. Diese führen zu einem Drivererror in der Steuerung und sind in diesem Byte genauer aufgeschlüsselt.

7	4	3	2	1	0
reserved		No free Rotation	Low Align	Parameter fault	VEL High

VEL High Geschwindigkeit höher als erlaubt

Parameter fault Es konnten nicht alle Parameter gelesen werden

Low Allign Strom zum Ausrichten wurde nicht erreicht

No free Rotation Die Achse lässt sich nach dem Ausrichten nicht frei bewegen

### 3.5.3.4 Rebel Base Power Error

Dieses Byte ist für spezielle Fehler der Stromversorgung der einzelnen Subsysteme des Rebels gewidmet. Fehler in dieser Nachricht weisen häufig auf einen Kurzschluss oder elektrisches Problem des Rebels hin.

7	4	3	2	1	0
reserved		Logic Bus Fault	Motor Bus Fault	System Bus Fault	VIN Fault

VIN Fault Eingangsversorgung fehlerhaft

System Bus Fault	24V Versorgung für Lüfter und DIO Kanäle fehlerhaft
Motor Bus Fault	24V Versorgung für Motoren fehlerhaft
Logic Bus Fault	5V Versorgung für Motorcontroller und Steuerung fehlerhaft

### 3.5.3.5 Rebel Base Low Voltage Error

Dieses Byte ist den gemessenen Spannungen der Rebel Base Subsysteme gewidmet. Wenn Spannungen von erwarteten Werten abweichen, werden diese als fehlerhaft markiert. Fehler in dieser Nachricht weisen häufig auf einen Kurzschluss oder elektrisches Problem des Rebels hin.

7	4	3	2	1	0
reserved		Logic Voltage	Motor Voltage	System Voltage	VIN Voltage

VIN Voltage	Eingangsversorgung nicht innerhalb erwartetem Bereich (21,1V-26,0V)
System Voltage	24V Versorgung für Lüfter und DIO Kanäle nicht innerhalb erwartetem Bereich (22,0V-24,0V)
Motor Voltage	24V Versorgung für Motoren nicht innerhalb erwartetem Bereich (23,0V-26,0V)
Logic Voltage	5V Versorgung nicht innerhalb erwartetem Bereich (4,7V-5,6V)

### 3.5.3.6 Rebel V3 Encoder Error

Dieses Byte ist für spezielle Fehler der beiden Absolut Encoder im Rebel V3 gewidmet. Fehler in dieser Nachricht weisen häufig auf einen Defekt des Encoder oder auf eine fehlerhafte Kalibrierung hin. Der antriebs- und abtriebseitige Encoder haben ihr eigenes Fehler Byte. Der Abtrieb ist im Byte 5 und der Antrieb im Byte 6 der Extended Error Nachricht der Motorcontroller zu finden

7	6	5	4	3	2	1	0
Other	Not Calibrated	Position Error	Checksum Error	I2C Error	Master Analog	Nonius Analog	COM Error

COM Error	Kommunikation mit dem Encoder nicht möglich oder fehlerhaft (invertiert)
Nonius Analog	Analog Fehler des Nonius Tracks
Master Analog	Analog Fehler des Master Tracks
I2C Error	Kommunikation mit EEPROM nicht möglich oder fehlerhaft
Checksum Error	Checksum stimmt nicht mit EEPROM Daten überein
Position Error	Absolut Position stimmt nicht mit gezählter Position überein
Not Calibrated	Kalibrierung der Achse wurde gelöscht oder nicht durchgeführt
Other	Absolut Position während Referenzierung inkonsistent oder anomales Fehlerregister gesetzt

### 3.5.4 Encoder Rebel V3 Position

Der Igus Rebel V3 ist mit On-Axis Motorcontrollern ausgestattet, die Achsen besitzen einen Abtriebs und Antriebsencoder. Sind die Encoder Parametrierung sendet der Motorcontroller einmal pro Sekunde die Die Antriebs und Abtriebsposition.

CAN-ID = BoardID + 2

0	1	2	3	4	5	6	7
0xEF	Encoder Identifier	reserved		Encoder Position			

Encoder Identifier    Identifier des Abtriebs oder Antriebsencoders (Eins für Antrieb, zwei für Abtrieb)

Encoder Position    Position des Abtriebs in  $\frac{Grad}{100}$

#### 3.5.4.1 Rebel V2 Abtriebsencoder Error

Dieses Byte ist für spezielle Fehler des Abtriebsencoders im Rebel V2 gewidmet. Fehler in dieser Nachricht weisen häufig auf einen Defekt des Encoder oder auf eine fehlerhafte Kalibrierung hin.

7	6	5	4	3	2	1	0
Contact B	Contact A	reserved		Calib. corrupt	Calib. logic	Out of Range	COM Error

COM Error    Prüfnummer nicht richtig empfangen

Out of Range    Gemessener Wert nicht im Wertebereich  $[-180^\circ - 180^\circ]$

Calib. logic    Encoder meldet Fehler in der Kalibrierung

Calib. corrupt    Encoder meldet fehlende Werte in der Kalibrierung

Contact A    Schleifkontakt Spur A abgehoben

Contact B    Schleifkontakt Spur B abgehoben

### 3.5.5 Abtriebsencoder Rebel V2 Position

Der Igus Rebel V2 ist mit On-Axis Motorcontrollern ausgestattet, die Achsen besitzen einen Abtriebsencoder. Ist der Abtriebsencoder Parametrierung sendet der Motorcontroller einmal pro Sekunde die Abtriebsposition.

CAN-ID = BoardID + 2

0	1	2	3	4	5	6	7
0xEF	0x0000	0x7E	Encoder Position				

0x7E    Erkennung des Encoders der Firma Hatron

Encoder Position    Position des Abtriebs in  $\frac{Grad}{100}$