



**Heinzmann GmbH & Co. KG**  
**Engine & Turbine Controls**

Am Haselbach 1  
D-79677 Schönau (Schwarzwald)  
Germany

Telefon +49 7673 8208-0  
Telefax +49 7673 8208-188  
E-Mail [info@heinzmann.com](mailto:info@heinzmann.com)  
www.heinzmann.com

USt-IdNr.: DE145551926

**HEINZMANN®**  
**Digitale Elektronische Drehzahlregler**

**CANopen Implementation**



 <p><b>Achtung</b></p>	<p>Vor Installation, Inbetriebnahme und Wartung sind die entsprechenden Handbücher im ganzen durchzulesen.</p> <p>Alle Anweisungen die die Anlage und die Sicherheit betreffen, müssen unbedingt befolgt werden.</p>
 <p><b>Gefahr</b></p>	<p>Nichtbefolgen der Anweisung kann zu Personen- und/oder Sachschäden führen.</p> <p>HEINZMANN übernimmt keine Haftung für Schäden, die durch Nichtbefolgen von Anweisungen entstehen.</p>
 <p><b>Achtung! Hochspannung</b></p>  <p><b>Gefahr</b></p>	<p><b>Vor der Installation ist folgendes zu beachten:</b></p> <p>Vor Beginn einer Installation an der Anlage, ist diese spannungsfrei zu schalten!</p> <p>Kabelabschirmung und Stromversorgungsanschlüsse entsprechend der <i>Europäischen Richtlinie bezüglich EMV</i> verwenden.</p> <p>Überprüfung der Funktion vorhandener Schutz und Überwachungssysteme.</p>
 <p><b>Gefahr</b></p>	<p><b>Um Schäden an Anlage und Personen zu vermeiden, müssen folgende Überwachungs- und Schutzsysteme vorhanden sein:</b></p> <p>vom Drehzahlregler unabhängiger Überdrehzahlschutz</p> <p>Übertemperaturschutz</p> <p>HEINZMANN übernimmt keine Haftung für Schäden, die durch fehlenden oder unzureichenden Überdrehzahlschutz entstehen.</p> <p><b>Bei Generatoranlagen zusätzlich:</b></p> <p>Überstromschutz</p> <p>Schutz vor Fehlsynchronisation bei zu großer Frequenz-, Spannungs-, oder Phasendifferenz</p> <p>Rückleistungsschutz</p>
	<p><b>Ursachen für Überdrehzahl können sein:</b></p> <p>Ausfall der Spannungsversorgung</p> <p>Ausfall des Stellgerätes, des Kontrollgerätes oder dessen Zusatzgeräte</p> <p>Schwergängigkeit- und Festklemmen des Gestänges</p>



**Achtung**

Die Beispiele, Daten und alle übrigen Informationen in diesem Handbuch dienen ausschließlich dem Zweck der Unterweisung und sollten für keine spezielle Anwendung eingesetzt werden, ohne dass der Anwender unabhängige Tests und Überprüfungen durchgeführt hat.



**Gefahr**

Unabhängige Tests und Überprüfungen sind von besonderer Bedeutung bei allen Anwendungen, bei denen ein fehlerhaftes Funktionieren zu Personen- oder Sachschäden führen kann.

**HEINZMANN** übernimmt keine Garantie, weder ausdrücklich noch stillschweigend, dass die Beispiele, Daten oder sonstigen Informationen in diesem Handbuch fehlerfrei sind, Industriestandards entsprechen oder den Bedürfnissen irgendeiner besonderen Anwendung genügen.

**HEINZMANN** lehnt ausdrücklich die stillschweigende Garantie für die Marktfähigkeit oder die Eignung für einen speziellen Zweck ab, auch für den Fall, dass **HEINZMANN** auf einen speziellen Zweck aufmerksam gemacht wurde oder dass im Handbuch auf einen speziellen Zweck hingewiesen wird.

**HEINZMANN** lehnt jede Haftung für mittelbare und unmittelbare Schäden sowie für Begleit- und Folgeschäden ab, die sich aus irgendeiner Verwendung der in diesem Handbuch enthaltenen Beispiele, Daten oder sonstigen Informationen ergeben.

**HEINZMANN** übernimmt keine Gewähr für die Konzeption und Planung der technischen Gesamtanlage. Dies ist Sache des Betreibers bzw. deren Planer und Fachingenieure. Es liegt auch in deren Verantwortungsbereich zu überprüfen, ob die Leistungen unserer Geräte dem angestrebten Zweck genügen. Der Betreiber ist auch für eine ordnungsgemäße Inbetriebnahme der Gesamtanlage verantwortlich.

## Versionsinformation

<b>Version</b>	<b>Beschreibung der Änderung</b>	<b>Datum</b>	<b>Bearbeiter</b>
1.00	erstellt	29.08.06	Sz



## Inhaltsverzeichnis

	Seite
<b>1 Einführung</b> .....	<b>1</b>
<b>2 Allgemeine CAN-Parameter</b> .....	<b>2</b>
<b>3 Telegramm-Identifizier</b> .....	<b>3</b>
<b>4 Netzwerk-Management</b> .....	<b>4</b>
<b>5 Servicedaten-Objekt</b> .....	<b>5</b>
<b>6 Emergency-Objekt</b> .....	<b>6</b>
<b>7 Empfangstelegramme</b> .....	<b>7</b>
7.1 Telegrammaufbau .....	7
7.2 Datenempfang.....	8
7.2.1 Schalterfunktionen .....	8
7.2.2 Sensoren.....	11
<b>8 Sendetelegramme</b> .....	<b>13</b>
8.1 Übertragungsarten .....	13
8.2 Sendewerte .....	14
8.2.1 Zusammenfassung von Sende-Bits .....	14
<b>9 Lebenszeichenüberwachung</b> .....	<b>15</b>
9.1 Heartbeat-Überwachung.....	15
9.2 Node/Life Guarding.....	15
<b>10 Parameterbeschreibung</b> .....	<b>16</b>
10.1 Wertebereich von Sensoren.....	16
10.2 Wertebereich von Mess- und Anzeigewerten.....	17
10.3 Übersichtstabelle .....	18
10.4 Parameter .....	19
10.5 Messwerte.....	21
10.6 Funktionen.....	23
10.7 Felder.....	23



## 1 Einführung

CAN-Protokolle können auf Anfrage in alle HEINZMANN-Steuergeräte implementiert werden, die zumindest einen CAN-Controller besitzen. In diesem Dokument wird nicht das CANopen-Protokoll selbst, sondern seine Implementierung in die Firmware beschrieben. Grundlage für die Software-Implementierung ist das Dokument „CANopen Application Layer and Communication Profile“, CiA Draft Standard 301 Version 4.01 vom 1. Juni 2000.

Das HEINZMANN-Steuergerät arbeitet als Slave im 11 Bit CANopen Pre-defined Master-Slave Connection Set.

Vier RPDOs und vier TPDOs erhalten im Standarddatensatz die vordefinierten Identifier, die aber verändert werden können. Die eigene Knotennummer wird automatisch in diese Identifier integriert, so dass sie nicht bei jeder Anpassung der Knotennummer verändert werden müssen. Weitere 12 TPDOs können auf Anfrage implementiert werden. Für die TPDOs werden alle vordefinierten Übertragungsarten unterstützt. Die RPDOs werden asynchron erwartet. Bei regelmäßiger Sendung durch die Gegenseite kann der Empfang überwacht werden.

Heartbeat Producer, Heartbeat Consumer und Node/Life Guarding wurden beide nach Standard implementiert. Es kann wahlweise eine der beiden Überwachungsarten aktiviert werden.

Das Emergency-Objekt ist eingebunden und sendet sowohl das Auftreten aller Fehler, die im Steuergerät erkannt werden können als auch ihre Behebung. Das EMCY-Objekt kann separat aktiviert werden.

Es ist genau ein Servicedaten-Objekt SDO realisiert. Die implementierten Services werden in *↑5 Servicedaten-Objekt* beschrieben.

Zu jeder CANopen-Implementierung wird ein EDS-File geliefert mit der Beschreibung der Implementierung. Hierbei ist zu beachten, dass die Implementierungsparameter vom Kunden über DcDesk 2000 verändert werden können, insbesondere die Belegung der PDOs. Das EDS-File muss folglich eigenverantwortlich angepasst werden.

## 2 Allgemeine CAN-Parameter

Die Baudrate wird in 21750 *CanOp:Baudrate* eingetragen. Es sind nur die vier Werte 125 kBaud, 250 kBaud, 500 kBaud und 1000 kBaud gültig, bei jedem anderen Eintrag werden 250 kBaud verwendet.

Die eigene Knotennummer wird in 21751 *CanOp:MyNodeNo* übergeben, die des Masters/Partners im CANopen-Netzwerk in 21752 *CanOp:PartnerNodeNo*. Die entsprechenden Telegrammidentifizier werden automatisch mit diesen Knotennummern ergänzt, um die Systemparametrierung einfach zu gestalten.

Nach dem Start des Steuergerätes wird die Zeit 21753 *CanOp:TimeOutDelay* abgewartet, innerhalb der keine Timeout-Überwachung der RPDOs durchgeführt wird. Innerhalb dieser Zeit sollte sowohl das Steuergerät in den OperationalMode geschaltet als auch die vereinbarten RPDOs gesendet werden.



Hinweis

*Bei Veränderung der CANopen-Parameter sollten diese gespeichert und anschließend ein Steuergeräte-Reset durchgeführt werden.*

In 23759 *CanOp:RxIRCount* werden alle Empfangstelegramme gezählt. Dies dient ausschließlich der schnellen Überprüfung, ob der Bus läuft.

CAN-Bus-Fehler werden in 3070 *ErrCanBus/ErrCanBus1* bzw. 3072 *ErrCanBus2* angezeigt. Dieser Fehler signalisiert entweder den Bus-Off-Status des CAN-Controllers oder zeigt an, dass innerhalb von 100 s kein Telegramm auf den Bus gelegt werden konnte. Bei einem Busfehler geht das Steuergerät automatisch in den Pre-Operational-Zustand, wenn es vorher im Operational- oder Stopped-Zustand war.

Der Fehler 3071 *ErrCanComm/ErrCanComm1* bzw. 3073 *ErrCanComm2* signalisiert, dass der CAN-Bus in Ordnung ist, aber inhaltliche Fehler aufgetreten sind. Die wichtigsten inhaltlichen Fehler werden in 23757 *CanOp:ErrLifeSign* und 23758 *CanOp:ErrRPDODTimeOut* angezeigt. Der gesamte CANopen-Fehlerstatus wird zusammen mit dem Fehler 3071/3073 über das EMCY-Objekt übertragen.

Der CANopen-Fehlerstatus hat folgenden Aufbau:

- Bit 0: Partner-Lebenszeichen ist ausgefallen (23757 *CanOp:ErrLifeSign*)
- Bit 1: mindestens ein RPDO ist in Timeout (23758 *CanOp:ErrRPDODTimeOut*)
- Bit 2: Node-Reset nicht möglich weil Motor läuft
- Bit 3: Empfangspuffer-Überlauf, Empfangstelegramm kann nicht ausgewertet werden
- Bit 4: Sendepuffer-Überlauf, Sendetelegramm kann nicht gesendet werden
- Bit 5..7: Reserve, immer 0

### 3 Telegramm-Identifizier

Die Telegramm-Identifizier sind mit den Standardwerten vorbelegt, können aber über folgende Parameter verändert werden:

21760 <i>CanOp:ID_SYNCCons</i>	Identifizier SYNC-Objekt (80 hex)
21761 <i>CanOp:ID_EMCPProd</i>	Identifizier EMCY-Objekt (80 hex)
21762 <i>CanOp:ID_HBeatCons</i>	Identifizier Heartbeat-Consumer (700 hex)
21763 <i>CanOp:ID_HBeatProd</i>	Identifizier Heartbeat-Producer (700 hex)
21764 <i>CanOp:ID_ClientSDO</i>	Identifizier Client-SDO-Objekt (600 hex)
21765 <i>CanOp:ID_ServerSDO</i>	Identifizier Server-SDO-Objekt (580 hex)
21770 <i>CanOp:RPDOID()</i>	Identifizier der vier RPDOs (200, 300, 400, 500 hex)
21774 <i>CanOp:TPDOID()</i>	Identifizier der vier TPDOs, auf Anfrage erweiterbar auf 16 TPDOs (180, 280, 380, 480 hex, alle anderen 0)

Die Identifizier 21760 *CanOp:ID\_SYNCCons* und 21778 *CanOp:TPDOID(4)* bis 21789 *CanOp:TPDOID(15)* werden wie parametrisiert in das Steuergerät übernommen.

In die Identifizier der Sendeobjekte 21761 *CanOp:ID\_EMCPProd*, 21763 *CanOp:ID\_HBeatProd*, 21765 *CanOp:ID\_ServerSDO*, 21770 *CanOp:RPDOID(0)* bis 21773 *CanOp:RPDOID(3)* und 21774 *CanOp:TPDOID(0)* bis 21777 *CanOp:TPDOID(3)* wird in die unteren sieben Bit automatisch die eigene Knotennummer 21751 *CanOp:MyNodeNo* integriert, so dass sie bei wechselnder Knotennummer nicht verändert werden müssen. Datensätze können dadurch auf andere Steuergeräte kopiert werden und es ist nur die Knotennummer anzupassen.

Der Identifizier 21762 *CanOp:ID\_HBeatCons* wird mit der eigenen Knotennummer 21751 *CanOp:MyNodeNo* ergänzt, wenn das Node/Life Guarding aktiviert ist (23756 *CanOp:LifeGuarding* = 1), in allen anderen Fällen mit der Partnerknotennummer 21752 *CanOp:PartnerNodeNo*.

## 4 Netzwerk-Management

Der aktuelle Zustand des Steuergerätes im CANopen-Systems wird in folgenden Parametern angezeigt:

23750 <i>CanOp:Init</i> = 1	Initialisation
23751 <i>CanOp:PreOperational</i> = 1	Pre-Operational
23752 <i>CanOp:Operational</i> = 1	Operational
23753 <i>CanOp:Stopped</i> = 1	Stopped

Von diesen Werten kann jeweils nur einer aktiv sein.

Layer-Setting-Services werden nicht unterstützt. Baudrate und Knotennummer können über DcDesk 2000 verändert werden.

Die NMT-Kommandos 81 hex (ResetNode) und 82 hex (ResetComm) werden bei den Steuergeräten, die dies unterstützen, mit einem automatischen Steuergeräte-Reset beantwortet. Zur Sicherheit darf der Motor nicht laufen. In diesem Fall wird ein CANopen-Kommunikationsfehler erzeugt und über das EMCY-Telegramm gesendet  $\uparrow 6$  *Emergency-Objekt*.

Nach einem automatischen Reset geht das Steuergerät automatisch aus den Init-Zustand in den Pre-Operational-Zustand über und kann dann wieder angesprochen werden. Die Zeitdauer für diesen Vorgang ist unterschiedlich je nach Steuergerätetyp und kann bis zu 500 ms betragen.

## 5 Servicedaten-Objekt

Es ist genau ein Servicedatenobjekt SDO implementiert. Der Identifier des SDO-Empfangstelegramms ist in 21764 *CanOp:ID\_ClientSDO* zu parametrieren, der Identifier des SDO-Sendetelegramms in 21765 *CanOp:ID\_ServerSDO*. Diese Identifier werden intern automatisch mit der eigenen Knotennummer 21751 *CanOp:MyNodeNo* ergänzt, müssen also nicht angepasst werden, wenn sich diese ändert.

Segmentierter SDO-Transfer und Block-Transfers werden nicht unterstützt.

Die Objekte 1000, 1001, 1005, 100C, 100D, 1010, 1011, 1014, 1016, 1017, 1018, 1200, 1400-1403, 1600..1603, 1800..180F, 1A00..1A0F Hex sind realisiert.

Die Objekte 1002, 1003, 1006, 1007, 1008, 1009, 100A, 1012, 1013, 1015 und 1280 werden nicht unterstützt.

Folgende hersteller-spezifischen Objekte sind implementiert:

- 2000 Hex: Dieses Objekt erlaubt das separate Lesen und Beschreiben von Schalterfunktionen.
- 2001 Hex: Mit diesem Objekt können Sensorfehler gelesen und – wenn RPDO1 nicht aktiviert ist – auch beschrieben werden.
- 2002 Hex: Das Objekt erlaubt das Auslesen von Sensorwerten und – wenn das zugehörige RPDO nicht aktiviert ist – auch das Beschreiben von Sensorwerten.
- 2100..210F Hex: Diese Objekte erlauben das gezielte Auslesen einzelner Parameter der TPDOs, auch von Bitwerten.

## 6 Emergency-Objekt

Der Identifier des EMCY-Sendetelegramms ist in 21761 *CanOp:ID\_EMCYProd* zu parametrieren. Er wird intern automatisch mit der eigenen Knotennummer 21751 *CanOp:MyNodeNo* ergänzt, müssen also nicht angepasst werden, wenn sich diese ändert. Das EMCY-Objekt wird nur dann gesendet, wenn es mit 25751 *CanOp:EMCYOn* = 1 aktiviert wurde.

Das Emergency-Objekt unterstützt die beiden Fehlercodes 00xx (Error Reset or No error) und 10xx (Generic error).

Im Manufacturer specific Error Field in den Datenbytes 3 bis 6 werden der von Heinzmann generierte Fehlercode und evtl. zusätzlich ein Fehlerstatus gesendet.

- Byte 3,4 HEINZMANN-Fehlercode 3001..3094, 13000..13099, 23000..23099
- Byte 5,6 HEINZMANN-Fehlerstatus, wenn dies vom Steuergerät unterstützt wird (z.B. DARDANOS III, DARDANOS IV und ARIADNE)  
 CANopen-Fehlerstatus bei anderen Steuergeräten, nur bei den Fehlern 3071 *ErrCanComm/ErrCanComm1* bzw. 3073 *ErrCanComm2*, je nachdem, an welchem CAN-Controller das CANopen-Protokoll gefahren wird  
 0 sonst

Die Beschreibung aller existierenden Fehlercodes und der Fehlerstati ist der zum Steuergerät bzw. zur Software-Version gehörenden Dokumentation zu entnehmen.

Der CANopen-Fehlerstatus hat folgenden Aufbau:

- Bit 0: Partner-Lebenszeichen ist ausgefallen (23757 *CanOp:ErrLifeSign*)
- Bit 1: mindestens ein RPDO ist in Timeout (23758 *CanOp:ErrRPDOTimeOut*)
- Bit 2: Node-Reset nicht möglich weil Motor läuft
- Bit 3: Empfangspuffer-Überlauf, Empfangstelegramm kann nicht ausgewertet werden
- Bit 4: Sendepuffer-Überlauf, Sendetelegramm kann nicht gesendet werden
- Bit 5..7: Reserve, immer 0

Zum Löschen der Steuergeräte-Fehler über CANopen kann die Schalterfunktion 2828 *SwitchErrorReset* verwendet werden, indem 24828 *ChanTypErrorReset* auf 4 gesetzt wird und 20828 *CommErrorReset* auf die zugehörige Bitnummer in RPDO1.

Mit dem Löschen von Fehlern über DcDesk 2000 bzw. die Schalterfunktion 2828 *SwitchErrorReset* wird auch der CANopen-Fehlerstatus gelöscht.

Es werden maximal vier Emergency-Objekt-Telegramme innerhalb von 10 bzw. 16 ms gesendet, wobei dieses Zeitraster steuergerätespezifisch, dann aber immer gleich ist.

## 7 Empfangstelegramme

Die vier RPDOs entsprechen dem Standard des 11 Bit CANopen Pre-defined Master-Slave Connection Sets.

Die vier zugehörigen Identifier sind ab 21770 *CanOp:RPDOID()* mit den Defaultwerten 200, 300, 400 und 500 hex vorbelegt. Die Knotennummer des Steuergerätes 21751 *CanOp:MyNodeNo* wird automatisch in die sieben unteren Bit der Identifier übernommen, so dass diese bei Änderung der Knotennummer nicht jedes Mal angepasst werden müssen.

Über die RPDOs können die aktuellen Werte von Sensoren und Schalterfunktionen an das Steuergerät übertragen werden. Das jeweilige RPDO wird allerdings nur dann erwartet, wenn mindestens eine Schalterfunktion oder ein Sensor zugewiesen ist und das Telegramm ab 25770 *CanOp:RPDO1On* = 1 generell aktiviert wurde. Die Empfangstelegrammlänge muss mindestens der erwarteten Länge entsprechen, die ab 23770 *RPDOTelLen()* ausgegeben wird.

RPDOs werden asynchron erwartet. Bei regelmäßiger Sendung durch den Master/Partner kann aber auch separat für jedes RPDO ab 29000 *RPDOEvtTim()* eine Zeit vorgegeben werden, auf die der Datenempfang überwacht werden soll.

Für RPDO1 wird in den Parametern ab 23760 *CanOp:SwitchMask()* byteweise angegeben, an welchen Bits Schalterfunktionen bzw. Sensorfehler erwartet werden.

Für RPDO2 bis RPDO4 erscheint in den beiden Bytes ab 23764 *CanOp:SensorMask()* eine Maske, in der für jeden Platz, an dem ein Sensor erwartet wird, ein Bit gesetzt ist. Diese beiden Masken dienen der Überprüfung der Parametrierung auf Client- und Server-Seite.

### 7.1 Telegrammaufbau

RPDO1: Bitmaske	23760 <i>CanOp:SwitchMask(0)</i>
Telegrammlänge	23770 <i>RPDOTelLen(0)</i>
Aktivierung	25770 <i>CanOp:RPDO1On</i>
Empfangsrate	29000 <i>RPDOEvtTim(0)</i>
Byte 0 Bit 0..7:	Wert der Schalterfunktionen 1..8
Byte 1 Bit 0..7:	Wert der Schalterfunktionen 9..16
Byte 2 Bit 0..7:	Wert der Schalterfunktionen 17..24
Byte 3 Bit 0..7:	Wert der Schalterfunktionen 25..32
Byte 5 Bit 0..7:	Wert der Sensorfehler 1..8
Byte 6 Bit 0..3:	Wert der Sensorfehler 9..12
RPDO2: Sensormaske	23764 <i>CanOp:SensorMask(0)</i> , Bit 0..3
Telegrammlänge	23771 <i>RPDOTelLen(1)</i>
Aktivierung	25771 <i>CanOp:RPDO2On</i>
Empfangsrate	29001 <i>RPDOEvtTim(1)</i>
Wort 0..3:	Wert von Sensor 1..4
zugehörige Sensorfehler in RPDO1, Byte 5, Bit 0..3	

RPDO3: Sensormaske	23764 <i>CanOp:SensorMask(0)</i> , Bit 4..7
Telegrammlänge	23772 <i>RPDOTelLen(2)</i>
Aktivierung	25772 <i>CanOp:RPDO3On</i>
Empfangsrate	29002 <i>RPDOEvtTim(2)</i>
Wort 0..3:	Wert von Sensor 5..8
	zugehörige Sensorfehler in RPDO1, Byte 5, Bit 4..7
RPDO4: Sensormaske	23765 <i>CanOp:SensorMask(1)</i> , Bit 0..3
Telegrammlänge	23773 <i>RPDOTelLen(3)</i>
Aktivierung	25773 <i>CanOp:RPDO4On</i>
Empfangsrate	29003 <i>RPDOEvtTim(3)</i>
Wort 0..3:	Wert von Sensor 9..12
	zugehörige Sensorfehler in RPDO1, Byte 6, Bit 0..3

## 7.2 Datenempfang

Über das Telegramm RPDO1 werden die aktuellen Werte für Schalterfunktionen und Fehlerkennungen der Sensoren empfangen. Die Telegramme RPDO2 bis RPDO4 enthalten aktuelle Sensorwerte. Welche Schalterfunktion über welches Bit von RPDO1 empfangen wird bzw. welcher Sensor über welches Wort von RPDO2 bis RPDO4, muss der Produzent des sendenden Moduls mitteilen bzw. muss mit ihm abgestimmt werden.

### 7.2.1 Schalterfunktionen

Alle im Steuergerät definierten Schalterfunktionen können entweder über einen eigenen Hardware-Eingang oder über das Telegramm RPDO1 oder über beide Wege empfangen werden. Dieser Empfangsweg muss dem Steuergerät mitgeteilt werden.

Um die im Telegramm RPDO1 empfangenen Schalterfunktionen im Steuergerät verwenden zu können, muss für den Kanaltyp in 24810 *ChanTyp...* bis 24849 *ChanTyp...* der Wert 4 eingetragen werden, für den Empfang ausschließlich über die eigene Hardware oder für nicht benutzte Schalterfunktionen muss der Kanaltyp auf 0 stehen.

Bei Verwendung des Kanaltyps 4 für CANopen ist im zugehörigen Parameter 20810 *Comm...* bis 20849 *Comm...* die Bitnummer im Telegramm RPDO1 anzugeben. Es können bis zu 32 verschiedene Schalterfunktionen über das Telegramm gesendet werden. Die Aufteilung ist dem Programmierer des CANopen-Masters überlassen.

In den Parametern ab 23760 *CanOp:SwitchMask* wird byteweise angegeben, an welchen Bits Schalterfunktionen bzw. Sensorfehler erwartet werden. Dies dient der Überprüfung der Parametrierung auf Client- und Server-Seite.

Soll eine über CANopen empfangene Schalterfunktion zusätzlich direkt verkabelt werden, dann ist außerdem im Parameter 810 *Funct...* bis 849 *Funct...* die Nummer des ver-

wendeten Digitaleingangs anzugeben. Wird dieser Parameter aber auf 0 gesetzt, dann wird die Schalterfunktion nur über CAN empfangen.

Bei Verwendung von Kanaltyp 0 (nur eigene Hardware) ist im zugehörigen Parameter 810 *Funct...* bis 849 *Funct...* die Nummer des verwendeten Digitaleingangs anzugeben. Eine Eingangsnummer 0 ist gleichbedeutend mit „nicht benutzt“.

8xx *Funct...* = DI-Nr.      <> 0: (redundante) Verkabelung, 0: nicht verkabelt

208xx *Comm...* = Bitnr.      Bitnummer im Telegramm RPDO1 (0, 1..32)

248xx *ChanTyp...* = 4      Schalter wird über CANopen empfangen

Die Bitnummer zählt byteweise, d.h. das erste Datenbyte im Telegramm enthält die Bits 1..8 (LSB...MSB), das zweite die Bits 9..16 (LSB...MSB) usw. Eine Bitnummer 0 ist gleichbedeutend mit „nicht benutzt“. Solch eine Schalterfunktion hat von der Kommunikationsseite her immer den Wert 0.

Die Schalterfunktion wird aktiviert, wenn sie von mindestens einer der beiden Quellen aktiviert wird: Digitaleingang ODER RPDO1.

Mit dem Wert „1“ in RPDO1 wird eine Schalterfunktion aktiviert, mit „0“ deaktiviert. Bei Schalterfunktionen, die als Umschalter dienen, wird mit „1“ die Bedeutung links vom „Or“ im Namen und mit „0“ die Bedeutung rechts vom „Or“ aktiviert. Zum Beispiel wird durch die Schalterfunktion 2827 *SwitchSetpoint2Or1* mit Sendung einer „1“ der Sollwertgeber 2 aktiviert und bei Übertragung einer „0“ der Sollwertgeber 1.

Aus Sicherheitsgründen muss über ein Kommunikationsmodul eine Funktion bewusst aktiviert werden. Deshalb können die Schalterfunktionen über Kommunikationsmodule im Gegensatz zu den Digitaleingängen nur high-aktiv, d.h. aktiv beim Empfang einer „1“, ausgeführt sein. Wenn die Verbindung zum Kommunikationsmodul unterbrochen ist, dann hat die Schalterfunktion automatisch den Wert 0.

### 7.2.1.1 Fehler bei Konfiguration oder CAN-Empfang von Schalterfunktionen

Wird für Schalterfunktionen ab 24810 *ChanTypEngineStop* der Wert 4 für CANopen gesetzt, ohne dass das Protokoll über 25750 *CanOpenOn* = 1 aktiviert ist, dann werden alle diese Schalterfunktionen auf den CAN-Wert Null zurückgesetzt und gleichzeitig ein Konfigurationsfehler 3000 *ConfigurationError* ausgegeben.

Liegt ein CAN-Fehler an, sei es durch einen Busfehler oder ein Timeout des Telegramms RPDO1, dann werden ebenfalls alle Schalterfunktionen, die über CANopen belegt werden sollen, in ihrem CAN-Wert auf Null zurückgesetzt. Wird das Telegramm wieder empfangen, so werden die Schalterfunktionen auch wieder über CAN ermittelt.

### 7.2.1.2 Schalterfunktion Motorstop

Im Falle eines CAN-Fehlers werden die über CAN ermittelten Schalter gelöscht bzw. auf Null zurückgesetzt. Soll in diesem Fall ein zuvor über CAN anliegendes „Motorstop-Signal“ in jedem Fall zu einem Motorstop führen, so muss Parameter 4810 *StopImpulseOrSwitch* auf 1 gesetzt werden. Mit diesem Parameter kann konfiguriert werden, ob ein externer Stoppbefehl nur während der Zeit wirkt, in der der Befehl explizit anliegt, oder ob ein Impuls ausreicht, ihn zu aktivieren bis der Motor steht.

4810 *StopImpulseOrSwitch* = 1                      Motorstop nur aktiv, wenn der Stoppbefehl explizit anliegt

4810 *StopImpulseOrSwitch* = 0                      durch einmaligen Schaltimpuls ist die Motorstop-Anforderung aktiv bis der Motor steht.



Hinweis

*Aus Sicherheitsgründen empfiehlt HEINZMANN, den Motorstop grundsätzlich direkt zu verkabeln, unabhängig davon, ob er auch über ein Kommunikationsmodul empfangen wird.*

### 7.2.1.3 Wert einer Schalterfunktion

Bei Ein- bzw. Ausschaltern entspricht der Name der Bedeutung *Ein*. Der Zustand 1 der Schalterfunktion definiert immer *Ein* und der Zustand 0 steht für *Aus*. Bei Umschaltern, die immer den Text „Or“ im Namen enthalten, ist der Teil links vom „Or“ aktiv, wenn die Schalterfunktion den Wert 1 hat und der Teil rechts vom „Or“, wenn die Schalterfunktion den Wert 0 hat.

Wenn in die aktuelle Firmware kein Kommunikationsmodul eingebunden ist, dann bestimmt ausschließlich der Digitaleingang den Wert der Schalterfunktion. Die Parameter ab 20810 *CommEngineStop* und 24810 *ChanTypEngineStop* existieren nicht.

Wenn aber ein Kommunikationsmodul zu berücksichtigen ist, dann kann jede Schalterfunktionen entweder über einen Digitaleingang oder über das Kommunikationsmodul oder sogar über beide Wege empfangen werden.

#### 1. Empfang nur über Digitaleingang

Der Parameter ab 24810 *ChanTypEngineStop* muss auf 0 gesetzt werden.

Ist der Parameter ab 810 *FunctEngineStop* = 0, dann hat die Schalterfunktion immer den Wert 0, sonst den aktuellen Wert des Digitaleingangs (evtl. mit umgekehrter Aktivität).

#### 2. Empfang nur über Kommunikationsmodul

Der Parameter ab 810 *FunctEngineStop* muss 0 sein und der zugehörige Parameter ab 24810 *ChanTypEngineStop*  $\geq 3$ .

Ist der Parameter ab 20810 *CommEngineStop* = 0, dann hat die Schalterfunktion immer den Wert 0, sonst den aktuellen Wert aus dem Empfangstelegramm. Ist die Verbindung zum Kommunikationsmodul unterbrochen, dann hat die Schalterfunktion den Wert 0.

### 3. Empfang über Digitaleingang und Kommunikationsmodul

Der Parameter ab 810 *FunctEngineStop* ist ungleich 0, derjenige ab 20810 *CommEngineStop* > 0 und der ab 24810 *ChanTypEngineStop* >= 3.

Der aktuelle Wert vom Digitaleingang (evtl. invertiert) und der des Kommunikationsmoduls werden ODER-verknüpft. Die Schalterfunktion ist also nur dann 0, wenn beide Quellen den Wert 0 vorgeben, und 1, wenn mindestens eine Quelle den Wert 1 vorgibt. Wenn die Verbindung zum Kommunikationsmodul unterbrochen ist, dann hat die Schalterfunktion über diesen Weg den Wert 0. In dem Fall entscheidet der Digitaleingang allein über den Gesamtwert.



Hinweis

*HEINZMANN empfiehlt, Umschalter – also Schalterfunktionen, die zwischen zwei Funktionen umschalten (Or im Parameternamen) - niemals über beide Wege empfangen zu lassen.*

## 7.2.2 Sensoren

Jeder im Steuergerät definierte Sensor kann entweder über einen eigenen Hardware-Eingang oder über die Telegramme RPDO2 bis RPDO4 empfangen werden. Dieser eindeutige Empfangsweg muss dem Steuergerät mitgeteilt werden.

Um die in den Telegrammen RPDO2 bis RPDO4 empfangenen Sensorwerte im Steuergerät verwenden zu können, muss für den Kanaltyp ab 4900 *ChanTypSetpoint1Ext* der Wert 4 eingetragen werden, für den Empfang über einen Analogeingang auf der eigenen Hardware muss der Kanaltyp auf 0 stehen und für den Empfang über einen PWM-Eingang auf eigener Hardware ist der Kanaltyp 1 zu verwenden.

Im zugehörigen Parameter ab 900 *AssignIn\_Setp1Ext* ist die Nummer des Eingangskanals anzugeben. Eine Kanalnummer 0 ist gleichbedeutend mit „nicht benutzt“.

49xx *ChanTyp...* = 4                      Sensor wird über CANopen empfangen

9xx *AssignIn\_...* = Kanalnr.          Kanalnummer (0, 1..12)

Es können bis zu 12 verschiedene Sensoren über die Telegramme RPDO2 bis RPDO4 empfangen werden. Die Aufteilung ist dem Programmierer des CANopen-Masters überlassen. Die Kanalnummern in den Telegrammen zählen wortweise, d.h. das erste Wort im Telegramm RPDO2 definiert den Kanal 1, das zweite Wort den Kanal 2 usw. Das vierte Wort des Telegramms RPDO4 hat die Kanalnummer 12.

### Parametrierbeispiel

Der aktuelle Ladedruck und die Kühlmitteltemperatur sollen alle 50 ms über die Worte 1 und 2 des Telegramms RPDO2 empfangen werden.

Nummer	Parameter	Wert	Einheit
904	<i>AssignIn_BoostPressure</i>	1	
907	<i>AssignIn_CoolantTemp</i>	2	
4904	<i>ChanType_BoostPress</i>	4	
4907	<i>ChanType_CoolantTemp</i>	4	
25750	<i>CanOpenOn</i>	1	
25771	<i>CanOp:RPDO2On</i>	1	
29001	<i>RPDOEvtTim(1)</i>	0,05	s

### 7.2.2.1 Fehler bei Konfiguration oder CAN-Empfang von Sensoren

Wird für Sensoren 49xx *ChanTyp...* = 4 gesetzt, ohne dass CANopen über 25750 *CanOpenOn* = 1 aktiviert ist, dann werden alle diese Sensorwerte intern auf den Wert Null zurückgesetzt und gleichzeitig ein Konfigurationsfehler 3000 *ConfigurationError* ausgegeben.

Liegt ein CAN-Fehler an, sei es durch einen Busfehler oder ein Timeout eines Empfangstelegramms, dann werden alle zugehörigen Sensoren, die über CANopen belegt werden sollen, auf den internen Wert Null zurückgesetzt. Kommt das Telegramm wieder, so werden die Sensorwerte auch wieder über CAN bereitgestellt.



Hinweis

*Der interne Wert Null bedeutet je nach Sensortyp einen anderen externen physikalischen Wert ↑ 10.1 Wertebereich von Sensoren.*

Der tatsächliche Wert von Sensoren ist bei einem Fehler aber abhängig von den Einstellungen ab 5000 *SubstOrLastSetp1Ext* und ab 5040 *HoldOrResetSetp1Ext*.

500x *SubstOrLast...* = 1                      Substitutionswert 1000 *Subst...* wird verwendet

500x *SubstOrLast...* = 0                      letzter gültiger Wert wird verwendet

Der durch den CAN-Fehler entstandene Sensorfehler wird nach Rückkehr des CAN-Signals je nach Parametrierung in 504x *HoldOrReset..* entweder bis zu einem Fehlerlöschen beibehalten oder bei Verschwinden des Fehlers selbst wieder gelöscht.

504x *HoldOrReset...* = 1                      Sensorfehler wird bis zum Fehlerlöschen gehalten

504x *HoldOrReset...* = 0                      Sensorfehler löscht sich bei Verschwinden der Fehlerursache von selbst

## 8 Sendetelegramme

Je nach Applikationsanforderung können bis zu 16 TPDOs definiert werden. Für jedes TPDO kann der Identifier in 21774 *CanOp:TPDOID()* parametrisiert werden.

Die ersten vier TPDOs entsprechen immer dem Standard des 11 Bit CANopen Pre-defined Master-Slave Connection-Sets. Diese vier Identifier sind mit den Defaultwerten 180, 280, 380 und 480 hex vorbelegt. Die Knotennummer des Steuergerätes 21751 *CanOp:MyNodeNo* wird bei diesen Telegrammen automatisch in die sieben unteren Bit der Identifier übernommen, so dass diese bei Änderung der Knotennummer nicht jedes Mal angepasst werden müssen. Die Identifier der TPDOs 5 bis 16 werden unverändert übernommen.

Es werden nur die TPDOs gesendet, die mit 25774 *CanOp:TPDO1On* = 1 aktiviert wurden.

### 8.1 Übertragungsarten

Es werden die folgenden Übertragungsarten unterstützt, die für jedes TPDO separat in 29004 *CanOp:TPDOTxType()* definiert werden können. Wenn andere Werte angegeben sind, wird das zugehörige Telegramm nicht gesendet.

- 0 Synchron azyklisch  
Sendung nach Empfang eines SYNC-Signals, aber nur bei Änderung mindestens eines Sendewertes zwischen zwei SYNC-Signalen
- 1..240 Synchron zyklisch  
Sendung nach Empfang des x-ten SYNC-Signals, x = [1,240]
- 252 Synchron, RTR only  
bei Empfang des SYNC-Telegramms werden die Daten gelatcht, Sendung erst mit RTR
- 253 Asynchron, RTR only  
Sendung bei Anforderung über RTR-Message
- 254 Asynchron, event manufacturer specific  
Sendung nach einer vordefinierten Zeit ab 29020 *CanOp:TPDOEvtTim()*, aber nur bei Änderung mindestens eines Wertes über den vordefinierten Hysteresewert hinaus und nicht häufiger als ab 29036 *CanOp:TPDOInhTim()* angegeben ist. Die Hysterese kann für jeden Sendeparameter separat ab 29116 *CanOp:TPDOxHyst()* definiert werden.  
  
Sendung auch bei Anforderung über RTR-Message

Der Empfang des SYNC-Objektes wird automatisch freigegeben, wenn eine synchrone Übertragung gefordert ist, ansonsten aber zur Minimierung der Systembelastung unterdrückt.

## 8.2 Sendewerte

Die zu sendenden Werte können für jedes TPDO separat definiert werden, indem ihre Parameternummern ab 29052 *TPDOxAssign()* eingetragen werden. Es können alle Parameter gesendet werden, die einen Level nicht höher als 4 haben. Die jeweiligen Parameternummern sind fortlaufend in die Felder einzutragen. Mit der ersten Null oder einem nicht existierenden bzw. nicht zugelassenen Parameter (zu hoher Level) endet das Übertragungsfeld. Die daraus vom Steuergerät ermittelte Telegrammlänge wird ab 23774 *CanOp:TPDOTelLen()* angezeigt. Ein TPDO wird nur dann gesendet, wenn die Telegrammlänge ungleich Null ist und das Telegramm über die Parameter ab 25774 *CanOp:TPDOIO* eingeschaltet wurde.

### 8.2.1 Zusammenfassung von Sende-Bits

Jeder Parameter wird als Wort übertragen, auch wenn er nur Byte- oder Bitgröße hat. Um den verfügbaren Platz besser nutzen zu können, können Bitparameter – also solche Parameter, die nur den Wert 0 oder 1 annehmen können – komprimiert werden. Dazu wird das Feld 29900 *BitCollParamSet()* zur Verfügung gestellt. Hier können beliebig Parameternummern von Bitparametern eingetragen werden. Die jeweils aktuellen Werte werden dann an gleicher Position (Feldindex = Bitnummer) in 23720 *BitCollection()* angezeigt. Diese Parameternummern 23720 ff können wiederum in die Parameter ab 29052 *TPDOxAssign()* eingetragen werden, um Bits komprimiert übertragen zu können. Standardmäßig enthält 29900 *BitCollParamSet()* 32 Elemente, daraus können also zwei Sendeworte gebildet werden. An nicht belegten Stellen wird eine 0 übertragen.

## 9 Lebenszeichenüberwachung

HEINZMANN-Steuergeräte erlauben sowohl die Heartbeat-Überwachung als auch das Node/Life Guarding-Verfahren. Für beide Verfahren werden die gleichen Identifier 21762 *CanOp:ID\_HbeatCons* und 21763 *CanOp:ID\_HBeatProd* verwendet, wobei der Consumer-Identifier unterschiedlich ergänzt wird.

### 9.1 Heartbeat-Überwachung

Wenn der Parameter 21754 *CanOp:HbeatConsTime* einen Wert ungleich Null enthält, wird das Lebenszeichen des Masters/Partners auf diesen Zeitabstand überwacht. Dies wird mit 23754 *CanOp:HBeatConsumer* = 1 angezeigt.

Wenn 21755 *CanOp:HbeatProdTime* ungleich Null ist, wird das eigene Lebenszeichen in dieser Senderate übertragen. 23755 *CanOp:HBeatProducer* = 1 zeigt dieses an.

Der Identifier des Heartbeat-Consumers ist in 21762 *CanOp:ID\_HbeatCons* zu parametrieren, er wird automatisch mit der Partner-Knotennummer 21752 *CanOp:PartnerNodeNo* erweitert, wenn die Lebenszeichen-Überwachung wie oben beschrieben aktiviert ist. Der Identifier des Heartbeat-Producers 21763 *CanOp:ID\_HbeatProd* wird hingegen automatisch mit der eigenen Knotennummer 21751 *CanOp:MyNodeNo* beaufschlagt.

### 9.2 Node/Life Guarding

Sind beide Werte für die Heartbeat-Überwachung mit Null belegt, dann entscheiden 21756 *CanOp:GuardingTime* und 21757 *CanOp:LifeTimeFactor*, ob die Funktion Node/Life Guarding aktiviert werden soll. Dies erfolgt nur dann, wenn diese beiden Parameter ungleich Null sind.

Bei aktiviertem Node-Guarding wird 21762 *CanOp:ID\_HbeatCons* mit der eigenen Knotennummer 21751 *CanOp:MyNodeNo* ergänzt. In 23756 *CanOp:LifeGuarding* wird angezeigt, ob das Node/Life Guarding-Verfahren aktiviert wurde.

## 10 Parameterbeschreibung

### 10.1 Wertebereich von Sensoren

Sensoren werden sowohl vom Steuergerät zum CANopen-Partner als auch in der anderen Richtung grundsätzlich im internen Wertebereich des Steuergerätes übertragen. Die Zuordnung des internen zum genutzten Wertebereich wird in den folgenden Tabellen aufgelistet. Es ist zu beachten, dass der genutzte Wertebereich bei einigen Parametern selbst wieder parametrierbar ist.

Sensor		Wertebereich				
		maximal		genutzt		intern
Nr.	Anzeigewert		Einheit	Nr.	Referenzparameter	
2900	SetpointIExtern	0,0..100,0	%		0,0 100,0	0 65535
2901	SetpointIExtern	0,0..100,0	%		0,0 100,0	0 65535
2902	LoadControlInput	0,0..100,0	%		0,0 100,0	0 65535
2903	SyncInput	0,0..100,0	%		0,0 100,0	0 65535
2904	BoostPressure	0,00..5,00	bar	982 983	BoostPressSensorLow BoostPressSensorHigh	0 65535
2905	OilPressure	0,00..20,00	bar	980 981	OilPressSensorLow OilPressSensorHigh	0 65535
2906	AmbientPressure	0..2000	mbar	984 985	AmbPressSensorLow AmbPressSensorHigh	0 65535
2907	CoolantTemp	-100,0..1000,0	°C		-100,0 1000,0	0 65535
2908	ChargeAirTemp	-100,0..1000,0	°C		-100,0 1000,0	0 65535
2909	OilTemp	-100,0..1000,0	°C		-100,0 1000,0	0 65535
2910	FuelTemp	-100,0..1000,0	°C		-100,0 1000,0	0 65535
2911	ExhaustTemp	-100,0..1000,0	°C		-100,0 1000,0	0 65535
2914	SlideExcitReduction	0,0..100,0	%		0,0 100,0	0 65535
2915	SlideSpeedReduction	0,0..4000,0	1/Min	987	0,0 SpeedRedSensorHigh	0 65535
2916	CoolantPressure	0,00..5,00	bar	978 979	CoolPressSensorLow CoolPressSensorHigh	0 65535
2917	AsymmetricLoad	0,0..100,0	%		0,0 100,0	0 65535
2918	MeasuredPower	0,0..100,0	%		0,0 100,0	0 65535

		0,0..2500,0	kW	992	MeasPowerSensorLow	0
				993	MeasPowerSensorHigh	65535
2919	PowerSetpoint	0,0..100,0	%		0,0 100,0	0 65535
		0,0..2500,0	kW	994 995	PowerSetpSensorLow PowerSetpSensorHigh	0 65535

## 10.2 Wertebereich von Mess- und Anzeigewerten

Nr.	Anzeigewert	Externer Wertebereich	Einheit	Interner Wertebereich
2000	Speed	0,0..4000,0	1/Min	0..65535
2031	SpeedSetp	0,0..4000,0		0..65535
2350	FuelQuantity	0,0..100,0 0.. 500,0	% mm <sup>3</sup> /Hub	0..65535
2300	ActPos	0,0..100,0	%	0..65535

weitere Werte auf Anfrage

### 10.3 Übersichtstabelle

In der folgenden Tabelle sind nur die für das CANopen-Protokoll relevanten Parameter in den einzelnen Parametergruppen nebeneinander aufgeführt.

Nr	Parameter	Nr	Messwerte	Nr	Funktionen	Nr	Kurven
810 ...	FunctEngineStop ...	2810...	SwitchEngineStop ...				
900 ...	AssignIn_Setp1Ext ...	2900...	Setpoint1Extern ...	4900...	ChanTypSetp1Ext ...		
		3070	ErrCanBus				
		3071	ErrCanComm				
20810.	CommEngineStop ...			24810.	ChanTypeEngineStop ..		
		23720	BitCollection(0)				
21750	CanOp:Baudrate	23750	CanOp:Init	25750	CanOpenOn		
21751	CanOp:MyNodeNo	23751	CanOp:PreOperational	25751	CanOp:EMCYOn		
21752	CanOp:PartnerNodeNo	23752	CanOp:Operational				
21753	CanOp:TimeOutDelay	23753	CanOp:Stopped				
21754	CanOp:HBeatConsTime	23754	CanOp:HBeatConsumer				
21755	CanOp:HBeatProdTime	23755	CanOp:HBeatProducer				
21756	CanOp:GuardingTime	23756	CanOp:LifeGuarding				
21757	CanOp:LifeTimeFactor	23757	CanOp:ErrLifeSign				
		23758	CanOp:ErrRPDOTimeOut				
		23759	CanOp:RxIRCount				
21760	CanOp:ID_SYNCCons	23760	CanOp:SwitchMask(0)				
21761	CanOp:ID_EMCYProd						
21762	CanOp:ID_HBeatCons						
21763	CanOp:ID_HBeatProd						
21764	CanOp:ID_ClientSDO	23764	CanOp:SensorMask(0)				
21765	CanOp:ID_ServerSDO						
21770	CanOp:RPDOID(0)	23770	CanOp:RPDOTelLen(0)	25770	CanOp:RPDO1On		
				25771	CanOp:RPDO2On		
				25772	CanOp:RPDO3On		
				25773	CanOp:RPDO4On		
21774	CanOp:TPDOID(0)	23774	CanOp:TPDOTelLen(0)	25774	CanOp:TPDO1On		
				25775	CanOp:TPDO2On		
				25776	CanOp:TPDO3On		
				25777	CanOp:TPDO4On		
				25778	CanOp:TPDO5On		
				25779	CanOp:TPDO6On		
				25780	CanOp:TPDO7On		
				25781	CanOp:TPDO8On		
				25782	CanOp:TPDO9On		
				25783	CanOp:TPDO10On		
				25784	CanOp:TPDO11On		
				25785	CanOp:TPDO12On		
				25786	CanOp:TPDO13On		
				25787	CanOp:TPDO14On		
				25788	CanOp:TPDO15On		
				25789	CanOp:TPDO16On		
						29000	CanOp:RPDOEvtTim()
						29004	CanOp:TPDOTxType()
						29020	CanOp:TPDOEvtTim()
						29036	CanOp:TPDOInhTim()
						29052	CanOp:TPDO1Assign()
						29056	CanOp:TPDO2Assign()
						29060	CanOp:TPDO3Assign()
						29064	CanOp:TPDO4Assign()
						29068	CanOp:TPDO5Assign()

Nr	Parameter	Nr Messwerte	Nr Funktionen	Nr Kurven
				29072 CanOp:TPDO6Assign()
				29076 CanOp:TPDO7Assign()
				29080 CanOp:TPDO8Assign()
				29084 CanOp:TPDO9Assign()
				29088 CanOp:TPDO10Assign()
				29092 CanOp:TPDO11Assign()
				29096 CanOp:TPDO12Assign()
				29100 CanOp:TPDO13Assign()
				29104 CanOp:TPDO14Assign()
				29108 CanOp:TPDO15Assign()
				29112 CanOp:TPDO16Assign()
				29116 CanOp:TPDO1Hyst()
				29120 CanOp:TPDO2Hyst()
				29124 CanOp:TPDO3Hyst()
				29128 CanOp:TPDO4Hyst()
				29132 CanOp:TPDO5Hyst()
				29136 CanOp:TPDO6Hyst()
				29140 CanOp:TPDO7Hyst()
				29144 CanOp:TPDO8Hyst()
				29148 CanOp:TPDO9Hyst()
				29152 CanOp:TPDO10Hyst()
				29156 CanOp:TPDO11Hyst()
				29160 CanOp:TPDO12Hyst()
				29164 CanOp:TPDO13Hyst()
				29168 CanOp:TPDO14Hyst()
				29172 CanOp:TPDO15Hyst()
				29176 CanOp:TPDO16Hyst()
				29900 BitCollParamSet()

In den folgenden Tabellen sind nur die für das CANopen-Protokoll relevanten Parameter und ihre Bedeutung aufgeführt. Für weitere Parameter des Steuergerätes wird auf die zugehörige Basis-Information verwiesen.

Bei Feldern ist jeweils nur der erste Feldparameter angegeben, wobei die Parameternummer mit der Kennung „ff“ (und folgende) gekennzeichnet ist.

## 10.4 Parameter

Nr.	Name	Bedeutung	
<b>810</b>	<b>Funct_...</b>		
<b>ff</b>	Level:	6	Zuweisung des Digitaleingangs, an dem der Wert
	Bereich:	-8..8	für die angegebene Schalterfunktion erwartet
	Seite(n):	8, 10	wird
<b>900</b>	<b>AssignIn_...</b>		
<b>ff</b>	Level:	6	Zuweisung der Kanalnummer, an der der Wert
	Bereich:	0..12	für den angegebenen Sensor erwartet wird
	Seite(n):	11	
<b>21750</b>	<b>CanOp:Baudrate</b>		
	Level:	4	CAN-Baudrate
	Bereich:	125..1000 kBaud	
	Seite(n):	2	

Nr.	Name	Bedeutung
<b>21751</b>	<b>CanOp:MyNodeNo</b> Level: 4 Bereich: 1..127 Seite(n): 2, 5, 7, 13, 15	Eigene Knotennummer im CAN-Netzwerk
<b>21752</b>	<b>CanOp:PartnerNodeNo</b> Level: 4 Bereich: 0..127 Seite(n): 2, 15	Knotennummer des Masters/Partners im CAN-Netzwerk
<b>21753</b>	<b>CanOp:TimeOutDelay</b> Level: 4 Bereich: 0..100 s Seite(n): 2	Verzögerungszeit nach Einschalten des Steuergeräts
<b>21754</b>	<b>CanOp:HBeatConsTime</b> Level: 4 Bereich: 0..50 s Seite(n): 15	Heartbeat-Empfangsrate
<b>21755</b>	<b>CanOp:HBeatProdTime</b> Level: 4 Bereich: 0..50 s Seite(n): 15	Heartbeat-Senderate
<b>21756</b>	<b>CanOp:GuardingTime</b> Level: 4 Bereich: 0..50 s Seite(n): 15	NodeGuarding-Überwachungszeit
<b>21757</b>	<b>CanOp:LifeTimeFactor</b> Level: 4 Bereich: 0..255 Seite(n): 15	Faktor für die NodeGuarding-Überwachungszeit
<b>21760</b>	<b>CanOp:ID_SYNCCons</b> Level: 4 Bereich: 0000..07FF Hex Seite(n): 3	Identifizier des SYNC Empfangstelegramms Standard: 80 Hex
<b>21761</b>	<b>CanOp:ID_EM CYProd</b> Level: 4 Bereich: 0000..07FF Hex Seite(n): 3, 6	Identifizier des EMCY Sendetelegramms Standard: 80 Hex
<b>21762</b>	<b>CanOp:ID_HBeatCons</b> Level: 4 Bereich: 0000..07FF Hex Seite(n): 3, 15	Identifizier des Heartbeat Empfangstelegramms Standard: 700 Hex + Partner-Knotennummer (wird automatisch addiert) <u>oder</u> Identifizier des NodeGuarding Empfangstelegramms Standard: 700 Hex + eigene Knotennummer (wird automatisch addiert)
<b>21763</b>	<b>CanOp:ID_HBeatProd</b> Level: 4 Bereich: 0000..07FF Hex Seite(n): 3, 15	Identifizier des Heartbeat Sendetelegramms Standard: 700 Hex + eigene Knotennummer (wird automatisch addiert)
<b>21764</b>	<b>CanOp:ID_ClientSDO</b> Level: 4 Bereich: 0000..07FF Hex	Identifizier des SDO Empfangstelegramms Standard: 600 Hex + Partner-Knotennummer

Nr.	Name	Bedeutung	
	Seite(n):	3, 5	(wird automatisch addiert)
<b>21765</b>	<b>CanOp:ID_ServerSDO</b>		
	Level:	4	Identifiziert das SDO Sendetelegramm
	Bereich:	0000..07FF Hex	Standard: 580 Hex + eigene Knotennummer
	Seite(n):	3, 5	(wird automatisch addiert)
<b>21770</b>	<b>CanOp:RPDOID(0)</b>		
<b>ff</b>	Level:	4	Identifiziert die max. 4 RPDO
	Bereich:	0000..07FF Hex	Standard: 200, 300, 400, 500 hex + eigene Knotennummer (wird automatisch addiert)
	Seite(n):	3, 7	
<b>21774</b>	<b>CanOp:TPDOID(0)</b>		
<b>ff</b>	Level:	4	Identifiziert die max. 16 TPDO
	Bereich:	0000..07FF Hex	Standard der ersten vier TPDOs: 180, 280, 380, 480 hex + eigene Knotennummer (wird automatisch addiert)
	Seite(n):	3, 13	TPDO 5 bis 16 sind frei belegbar
<b>20810</b>	<b>Comm...</b>		
<b>ff</b>	Level:	6	Zuweisung einer Bitnummer in den ersten 4
	Bereich:	0..32	Byte von RPDO1, an der der Wert für die angegebene Schalterfunktion übergeben wird
	Seite(n):	8, 10	

## 10.5 Messwerte

Nr.	Name	Bedeutung	
<b>3070</b>	<b>ErrCanBus/ErrCanBus1</b>		
	Level:	1	CAN Busfehler, CAN-Controller 1
	Bereich:	0..1	
	Seite(n):	2	
<b>3071</b>	<b>ErrCanComm/ErrCanComm1</b>		
	Level:	1	CAN Kommunikationsfehler, CAN-Controller
	Bereich:	0..1	1
	Seite(n):	2, 6	
<b>3072</b>	<b>ErrCanBus2</b>		
	Level:	1	CAN Busfehler, CAN-Controller 2
	Bereich:	0..1	
	Seite(n):	2	
<b>3073</b>	<b>ErrCanComm2</b>		
	Level:	1	CAN Kommunikationsfehler, CAN-Controller
	Bereich:	0..1	2
	Seite(n):	2, 6	
<b>23720</b>	<b>BitCollection(0)</b>		
<b>ff</b>	Level:	1	Zusammenfassung von Einzelbits zur komprimierten Übertragung in TPDOs
	Bereich:	0000..FFFF Hex	Siehe 29900 <i>BitCollParamSet(0)</i>
	Seite(n):	14	
<b>23750</b>	<b>CanOp:Init</b>		
	Level:	1	1: CANopen im Init-Zustand
	Bereich:	0..1	
	Seite(n):	4	
<b>23751</b>	<b>CanOp:PreOperational</b>		
	Level:	1	1: CANopen im Preoperational-Zustand
	Bereich:	0..1	
	Seite(n):	4	

Nr.	Name	Bedeutung	
<b>23752</b>	<b>CanOp:Operational</b>		
	Level:	1	1: CANopen im Operational-Zustand
	Bereich:	0..1	
	Seite(n):	4	
<b>23753</b>	<b>CanOp:Stopped</b>		
	Level:	1	1: CANopen im Stopped-Zustand
	Bereich:	0..1	
	Seite(n):	4	
<b>23754</b>	<b>CanOp:HBeatConsumer</b>		
	Level:	1	1: Heartbeat Consumer ist aktiviert
	Bereich:	0..1	
	Seite(n):	15	
<b>23755</b>	<b>CanOp:HBeatProducer</b>		
	Level:	1	1: Heartbeat Producer ist aktiviert
	Bereich:	0..1	
	Seite(n):	15	
<b>23756</b>	<b>CanOp:LifeGuarding</b>		
	Level:	1	1: NodeGuarding ist aktiviert
	Bereich:	0..1	
	Seite(n):	3, 15	
<b>23757</b>	<b>CanOp:ErrLifeSign</b>		
	Level:	1	1: Lebenszeichenfehler liegt an (Heartbeat Consumer oder NodeGuarding)
	Bereich:	0..1	
	Seite(n):	2, 6	
<b>23758</b>	<b>CanOp:ErrRPDOTimeout</b>		
	Level:	1	1: mindestens ein RPDO ist im Timeout
	Bereich:	0..1	
	Seite(n):	2, 6	
<b>23759</b>	<b>CanOp:RxIRCount</b>		
	Level:	1	Zähler der Receive-Telegramme
	Bereich:	0..65535	
	Seite(n):	2	
<b>23760</b>	<b>CanOp:SwitchMask(0)</b>		
<b>ff</b>	Level:	4	Angabe der Bits, die in RPDO1 durch zugewiesene Schalterfunktionen belegt sind
	Bereich:	00..FF Hex	
	Seite(n):	7	
<b>23764</b>	<b>CanOp:SensorMask(0)</b>		
<b>ff</b>	Level:	4	Angabe der Sensoren, die RPDO2 bis RPDO4 zugewiesen wurden
	Bereich:	00..FF Hex	
	Seite(n):	7	
<b>23770</b>	<b>CanOp:RPDOTelLen(0)</b>		
<b>ff</b>	Level:	4	Ermittelte Minimallänge der bis zu 4 RPDOs
	Bereich:	0..8	
	Seite(n):	7	
<b>23774</b>	<b>CanOp:TPDOTelLen(0)</b>		
<b>ff</b>	Level:	4	Ermittelte Länge der bis zu 16 TPDOs
	Bereich:	0..8	
	Seite(n):	14	

## 10.6 Funktionen

Nr.	Name	Bedeutung	
<b>4900</b> <b>ff</b>	<b>ChanTyp...</b> Level: Bereich: Seite(n):	6 0..4 11	Zuweisung des Kanaltyps, von dem der Wert für den angegebenen Sensor erwartet wird 0 = Analog-Eingang 1 = PWM-Eingang 4 = CANopen (siehe 900 <i>AssignIn...</i> )
<b>24810</b> <b>ff</b>	<b>ChanTyp...</b> Level: Bereich: Seite(n):	6 0..4 8, 10	Zuweisung des Protokolltyps, von dem der Wert für die angegebene Schalterfunktion erwartet wird 0 = nur über Digitaleingänge (810 <i>Funct...</i> ) 4 = auch über CANopen (20810 <i>Comm...</i> )
<b>25750</b>	<b>CanOpenOn</b> Level: Bereich: Seite(n):	4 0..1 9, 12	Aktivierung des CANopen-Protokolls
<b>25751</b>	<b>CanOp:EMCYOn</b> Level: Bereich: Seite(n):	4 0..1 6	Aktivierung des EMCY-Sendetelegramms
<b>25770</b> <b>ff</b>	<b>CanOp:RPDOxOn</b> Level: Bereich: Seite(n):	4 0..1 7	Aktivierung der bis zu 4 RPDOs
<b>25774</b> <b>ff</b>	<b>CanOp:TPDOxOn</b> Level: Bereich: Seite(n):	4 0..1 13	Aktivierung der bis zu 16 TPDOs

## 10.7 Felder

Nr.	Name	Bedeutung	
<b>29000</b> <b>ff</b>	<b>CanOp:RPDOEvtTim(0)</b> Level: Bereich: Seite(n):	4 0..50 s 7	Event-Time der bis zu 4 RPDOs Index 0..3 = RPDO1..RPDO4
<b>29004</b> <b>ff</b>	<b>CanOp:TPDOTxType(0)</b> Level: Bereich: Seite(n):	4 0..255 13	Transmit-Typen der bis zu 16 TPDOs bis auf die Werte 241..251 und 255, die ignoriert werden, sind alle Transmit-Typen möglich Index 0..15 = TPDO1..TPDO16
<b>29020</b> <b>ff</b>	<b>CanOp:TPDOEvtTim(0)</b> Level: Bereich: Seite(n):	4 0..50 s 13	Event-Time der bis zu 16 TPDOs, nur bei Transmit-Type 254 Index 0..15 = TPDO1..TPDO16
<b>29036</b>	<b>CanOp:TPDOInhTim(0)</b>		

Nr.	Name		Bedeutung
<b>ff</b>	Level:	4	Inhibit-Time der bis zu 16 TPDOs, nur bei
	Bereich:	0..50 s	Transmit-Type 254
	Seite(n):	13	Index 0..15 = TPDO1..TPDO16
<b>29052</b>	<b>CanOp:TPDOxAssign(0)</b>		
<b>ff</b>	Level:	4	Zuweisung von Parameternummern zu den je-
	Bereich:	0..29999	weils max. 4 Sendeparametern der max. 16
	Seite(n):	14	TPDOs
			x = [1,16]
			Index 0..3 = Wort 1..4
<b>29116</b>	<b>CanOp:TPDOxHyst(0)</b>		
<b>ff</b>	Level:	4	Zuweisung von Hysteresewerten zu den jeweils
	Bereich:	0..100 %	max. 4 Sendeparametern der max. 16 TPDOs, nur
	Seite(n):	13	bei Transmit-Type 254
			x = [1,16]
			Index 0..3 = Wort 1..4
<b>29900</b>	<b>BitCollParamSet(0)</b>		
<b>ff</b>	Level:	4	Zusammenfassung von Parameternummern, die
	Bereich:	-29999..29999	Bitwerten entsprechen
	Seite(n):	14	für Wortübertragung in TPDO1..TPD16
			Siehe 23720 <i>BitCollection(0)</i>

## 11 Bestellung von Druckschriften

Unsere Druckschriften können in geringem Umfang kostenlos angefordert werden.

Bestellen Sie die notwendigen Druckschriften über unsere Drehzahlregler bei der nächsten [HEINZMANN Filiale/Vertretung](#).

Bitte vergl. Sie auch die Liste unserer Vertretungen in der Welt (Klick auf „HEINZMANN Filiale/Vertretung“).

### Bitte geben Sie folgende Informationen an:

- Ihren Namen,
- Name und Adresse Ihres Unternehmens (legen Sie einfach Ihre Visitenkarte bei),
- Adresse, an die wir die Druckschriften senden sollen (falls abweichend von oben),
- die Nummer und den Titel der gewünschten Druckschrift,
- oder die technischen Angaben Ihres HEINZMANN- Gerätes,
- die Anzahl der gewünschten Druckschriften.

Für die Bestellung einer oder mehrerer Druckschriften können Sie direkt die beiliegende Fax-Vorlage benutzen.

Mittlerweile sind auch die meisten Druckschriften im PDF-Format erhältlich. Diese können auf Wunsch per E-Mail verschickt werden.

Wir würden uns sehr freuen, Ihre Kommentare zu unseren Druckschriften zu erhalten.

Bitte senden Sie Ihre Meinung darüber an:

### **HEINZMANN GmbH & Co. KG**

Service Abteilung

Am Haselbach 1

D-79677 Schönau

Germany

# Fax Antwort

## Bestellung von HEINZMANN-Druckschriften

**Fax-Hotline +49 7673 / 8208-194**

Bitte senden Sie mir folgende Druckschriften:

Stückzahl	Druckschrift-Nummer	Bezeichnung

Bitte senden Sie mir Ihre neuesten Prospekte über

( ) die HEINZMANN Analogregler. Anwendung: .....

( ) die HEINZMANN Digitalregler. Anwendung: .....

Firma .....

Ansprechpartner .....

Abt./Funktion .....

Straße..... PLZ/Ort .....

Telefon. .... Fax .....

E-Mail.....

Branche.....

Datum .....