

SYSTEMHANDBUCH  
Version 1.2

# **E/A Modul 515x**

zum Einsatz in CANopen – Netzwerken  
nach CiA DS301 **V3.0 (alt)** und DS401 V1.3

oder an Lenze-Antriebsreglern

# Inhaltsverzeichnis

---

<b>1. VERSIONS-HISTORIE</b>	<b>4</b>
<b>2. VORWORT UND ALLGEMEINES</b>	<b>4</b>
2.1 BESTIMMUNGSGEMÄSSE VERWENDUNG	6
<b>3. SICHERHEITSHINWEISE</b>	<b>6</b>
<b>4. KURZBESCHREIBUNG</b>	<b>7</b>
4.1 KOMPONENTEN	7
4.2 MECHANISCHE DATEN	7
4.3 ELEKTRISCHE DATEN EA515-8I/O-1A	7
4.3.1 Digitale Eingänge	7
4.3.2 Digitale Ausgänge	8
4.3.3 Analoge Eingänge	8
4.3.4 Analoge Ausgänge	8
4.3.5 Anschluß CAN-BUS	8
4.3.6 Anschluß Versorgungsspannung und I/O-Anschlüsse	9
4.4 GEHÄUSEDESIGN	9
4.5 SOFTWARE	10
<b>5. DIGITALE EIN/AUSGABEN</b>	<b>11</b>
5.1 MELDUNG DER DIGITALEN EINGÄNGE	11
5.2 DIGITALE AUSGÄNGE	12
5.2.1 Einschränkungen der Definition von E/A-Richtungen bei EA515-8I/O-1A	12
5.2.2 Rücklesen der digitalen Ausgänge	12
5.2.3 Empfangsüberwachung der digitalen Ausgänge (für Lenze-Modus)	13
5.2.4 Ausgangsstörungen	13
<b>6. ANALOGE EIN/AUSGABEN</b>	<b>14</b>
6.1 ZUGRIFF AUF DIE ANALOGWERTE VIA CAN	14
6.2 E/A-MODUL „EA 515 4 I-O1 1 A/D-D/A“	14
6.3 E/A-MODUL „EA 515 4 A/D“	15
6.4 E/A-MODUL „EA 515 4 D/A“	15
<b>7. SPEZIALFUNKTIONEN UND SOFTWARE-VARIANTEN</b>	<b>16</b>
7.1 IMPULSZÄHLER (VORLÄUFIG, 10/97)	16
7.2 ÜBERSICHT "SOFTWARE-VARIANTEN"	17
<b>8. INSTALLATION</b>	<b>19</b>
8.1 CAN-BUS-VERDRAHTUNG	19
<b>9. ANZEIGEELEMENTE</b>	<b>20</b>
9.1 LED - ANZEIGE - KOMBINATIONEN	21
<b>10. CAN-BUS</b>	<b>24</b>
10.1 CAN-BUS - TIMING	24
10.2 CAN-BUS-KANÄLE	25
10.2.1 Prozeßdatenkanäle	26
10.2.2 Parameterkanal	27
10.2.3 Interne Master-Slave-Kommunikation	28
<b>11. PARAMETRIERUNG</b>	<b>31</b>
11.1 PARAMETRIERUNG MIT DEM DIP-SCHALTER	31
11.1.1 Belegung des DIP-Schalters (im Normalzustand)	31

# Inhaltsverzeichnis

---

11.1.2 Einstellung der CAN-Baudrate :.....	32
11.1.3 Testmodus (nur für den Hersteller).....	33
11.1.4 Programmiermodus beim EA515-8I/O.....	34
11.1.5 Belegung des DIP-Schalters im Programmiermodus.....	35
11.2 PARAMETRIERUNG MIT DEM PARAMETERKANAL (SDO) .....	38
11.2.1 Zugriff auf E/A-Codestellen über den Parameterkanal.....	38
11.2.2 Hinweise zur Verwendung des Parameterkanals.....	40
<b>12. CANOPEN.....</b>	<b>41</b>
12.1 DAS CANOPEN-OBJEKTVERZEICHNIS.....	41
12.1.1 Kommunikationsprofil.....	42
12.1.2 Geräteprofil.....	43
12.1.3 Hersteller-spezifisches Profil.....	46
12.2 SERVICE-DATEN-OBJEKTE (SDOs).....	46
12.2.1 Aufbau der SDO - Telegramme.....	46
12.2.2 Nichtflüchtiges Speichern von Parametern.....	47
12.2.3 Wiederherstellen der Default-Einstellungen.....	48
12.3 NMT-BOOT-UP.....	49
12.3.1 Einfaches Boot-Up.....	49
12.3.2 Extended Boot-Up.....	49
12.4 PROZESSDATENOBJEKTE (PDOs).....	51
12.4.1 PDO - Mapping.....	52
12.4.2 PDO - Übertragungsarten.....	53
12.4.3 Zyklische asynchrone Übertragung.....	54
12.5 NODE GUARDING / LIFE GUARDING.....	55
12.6 DAS EMERGENCY-TELEGRAMM.....	56
<b>13. SCHNELLSTART.....</b>	<b>57</b>
13.1 KONFIGURATION FÜR DEN EINSATZ AM LENZE-ANTRIEBSREGLER.....	57
13.2 KONFIGURATION FÜR DEN EINSATZ IM CANOPEN-NETZWERK.....	58
13.3 VERWENDETE CAN-IDENTIFIER.....	59
<b>14. CODETABELLE.....</b>	<b>61</b>

## 1. Versions-Historie

Versionsnummer	Datum	Autor	Bemerkungen, Änderungsgrund
V0.5	19.06.1998	W.Büscher	Kurzbeschreibung des Analog-Erweiterungsmoduls "4* analog-In" aufgenommen.
V0.5	09.07.1998	W.Büscher	Zeichnung zur Lage des CAN-Anschlußsteckers aufgenommen.
V0.6	18.08.1998	W.Büscher	Beschreibung "Software-Varianten" und "Codestellen-Bereiche im EA515" aufgenommen, Codestellenverzeichnis "Impulszähler" ausgelagert (nicht mehr in Standard-Firmware enthalten).
V0.7	23.03.1999	W.Büscher	Kurzbeschreibung des Analog-Erweiterungsmoduls „4* analog-out“ aufgenommen.
V0.8	08.07.1999	W.Büscher	Gliederung verbessert, Verweise auf Zusatzhandbücher.
V0.9	26.07.2000	W.Büscher	Codestellen-Verzeichnis erweitert, Hinweise auf die geplante Umstellung von DS301 V3.0 auf DS301 V4.0 eingefügt. Warnung vor der Verwendung von „Nodeguarding“ und „extended NMT Bootup“ eingefügt.
V1.0	31.08.2000	W.Büscher	Hinweis zum normalerweise <b>nicht</b> „autonom“ erfolgendem dauerhaften Abspeichern von Parametern bei der Beschreibung von Objekt 0x1010 eingefügt.
V1.1	16.11.2000	H. Habermann	Neuer Verteiler für die Dokumentation
V1.2	23.07.2001	W. Büscher	Erweiterungen speziell für Lenze, Firmware-Variante „9374“ (neu!) + Sonderfirmware für „EA-515-4D/A“ mit PDO-Skalierung +/- 16383 statt +/- 32767
V1.3	12.11.2002	W.Büscher	Codestelle 14 umbenannt: Alt = „Sendezyklus der digitalen Eingänge“ Neu=„Zusätzlicher Sendezyklus für TPDO“
V1.4	2008-09-10	W. Büscher	Umstieg auf OpenOffice; ungültigen Verteiler ("Mart2") entfernt; Dokumentname nun offiziell mit Artikelnummer: art85021_Ea515_Software_Systemhandbuch (.odt/.pdf) . Export aus OO als <i>tagged PDF</i> mit Hyperlinks .

## 2. Vorwort und Allgemeines

Dieses Systemhandbuch ist als umfassende technische Information zu den E/A-Modulen der Serie „515“ vorgesehen.

Eine kundengerechte Bedienungsanleitung ohne „interne“ Funktionserläuterungen ist in Vorbereitung (Stand: 08.07.1999). Eine kurze Beschreibung zur Inbetriebnahme des Moduls findet sich im Kapitel „Schnellstart“.

Für Module mit erweiterten Funktionen (z.B. A/D-, D/A- Wandler, Impulszähler) sind „Systemhandbuch-Zusätze“ erhältlich, in denen die Besonderheiten der verschiedenen Module aufgeführt sind:

- Artikelnummer 85021 : Allgemeines Systemhandbuch zur E/A-515-Familie
- Artikelnummer 85022 : Zusatzinformationen für "E/A-515 4 A/D"
- Artikelnummer 85024 : Zusatzinformationen für "E/A-515 4 D/A"
- Artikelnummer 85029 : Zusatzinfo "SDO-Zugriffe ohne Ballast" (CANopen)

### 2.1 Bestimmungsgemäße Verwendung

Falls das in dieser Dokumentation beschriebene Modul für den Einsatz mit einem Lenze-Antriebsregler vorgesehen ist, sind grundsätzlich auch die Begriffsdefinitionen, Bestimmungen und Sicherheitshinweise aus den Handbüchern der verwendeten Lenze-Servo-Umrichter zu beachten.

Für Module mit erweiterten Funktionen sind die Hinweise im dazugehörigen „Systemhandbuch-Zusatz“ zu beachten.

## 3. Sicherheitshinweise

Der Betreiber bzw. sein Sicherheitsbeauftragter ist verpflichtet

- das Einhalten aller relevanten Vorschriften, Hinweise und Gesetze zu kontrollieren,
- zu gewährleisten, daß nur qualifiziertes Personal an und mit den Anlagenkomponenten arbeitet,
- zu gewährleisten, daß das Personal die Betriebsanleitung bei allen entsprechenden Arbeiten verfügbar hat und
- nichtqualifiziertem Personal das Arbeiten an und mit der Anlage zu untersagen.

## 4. Kurzbeschreibung

### 4.1 Komponenten

- EA515-8I/O: dezentrale E/A-Baugruppe für CAN-Bus mit acht frei als Ein- oder Ausgang verwendbaren Anschlußklemmen.
- Spannungsversorgung 24 V.
- Hutschienen-Montage
- Anschlußtechnik: Schraub- / Steck - Klemmen
- zwei grüne Zustandsanzeige-LEDs („Betrieb“(unten) und „Mode“)
- zwei rote Fehleranzeige-LEDs (Modul- und Ausgangsfehler)
- gelbe LEDs zur Anzeige der E/A-Signal-Anzeige
- Prozessor-Typ: SAB 80C515C-L
- CAN-Bus-Transceiver: PCA82C250T (SMD) mit galvanischer Trennung

### 4.2 mechanische Daten

- Gehäusotyp: UEGM-MSTB der Firma Phoenix Contact  
Maße: Höhe: 79 mm  
Tiefe: 90.5 mm (+10 mm für E/A - Anschlußklemme)  
Breite: 25 mm
- Position des DIP-Schalters an der oberen Gehäuseseite (im aufgeschnappten Zustand)
- Position des CAN-Bus-Steckverbinders an der unteren Gehäuseseite
- Position der grünen Betriebs-LED unten

### 4.3 elektrische Daten EA515-8I/O-1A

- Spannungsversorgung: 18 bis 30 Volt DC, typisch 24 V
- Stromaufnahme: typisch 30 mA, max. 80 mA (bei ausgeschalteten Ausgängen) (die Stromaufnahme von Modulen mit Anloggerweiterungen liegt höher, siehe Zusatzhandbuch)

#### 4.3.1 Digitale Eingänge

- keine galvanische Trennung
- Eingangswiderstand:  $3 \text{ k}\Omega \leq R_i \leq 4 \text{ k}\Omega$
- Eingangsspannungsbereich „low“: 0 .. 5 V
- Eingangsspannungsbereich „high“: 13 .. 30 V

## 4.3.2 Digitale Ausgänge

- keine galvanische Trennung
- Typ: „PNP“ (High-Side-Treiber)
- kurzschlußfest
- maximaler Ausgangsstrom: ca. 1 A (pro Kanal, Treiber-Spezifikation !)
- Erkennung von Kurzschluß / Überlast ab ca. 4 A (pro Kanal)
  
- Ausgangsspannung „low“ : über Eingangswiderstände 0V (bei offenem Ausgang)
- Ausgangsspannung „high“: >  $U_b - 1,8V$  (bei maximaler Last, Versorgung mit 24 V)
- der aktuelle Zustand der Ausgänge kann zurückgelesen werden
- gemeinsame Kurzschluß-Statusanzeige mit einer roten LED
- Anzeige über rote LED, wenn Summenstrom überschritten

## 4.3.3 Analoge Eingänge

- galvanische Trennung
- 12 Bit bzw. 11 Bit + Vorzeichen
- Eingangsspannung oder Ströme
  - +/-10 V, ±20 mA (je nach Modultyp)

## 4.3.4 Analoge Ausgänge

- galvanische Trennung
- 12 Bit bzw. 11 Bit + Vorzeichen
- kurzschlußfest
- Ausgangsspannung oder Ströme
  - +/-10 V, ±20 mA (je nach Modultyp)

## 4.3.5 Anschluß CAN-BUS

3-poliges COMBICON-Stecksystem (Stifte) der Fa. Phoenix ohne Kragen, Rastermaß 5,00mm Bezeichnung **MSTBA 2,5/3-G-5,00**

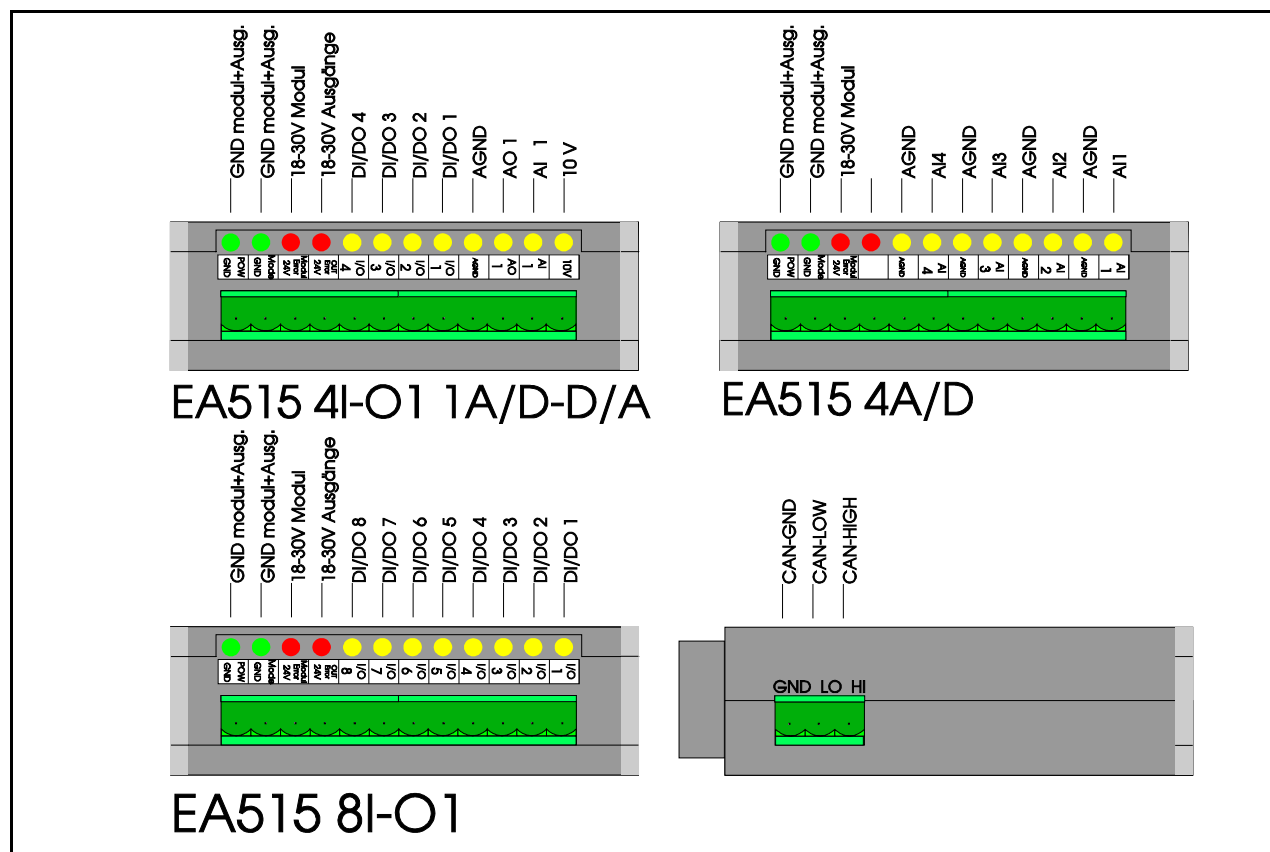
Pin1 : CAN-Ground

Pin2 : CAN-Low

Pin3 : CAN-High

Anschlußbelegung und Lage des CAN-Anschlußsteckers siehe 4.3.6 (rechts unten).

## 4.3.6 Anschluß Versorgungsspannung und I/O-Anschlüsse



\* „EA515 8I-O1“ == „EA515-8I/O-1A“ (Katalogbezeichnung),  
 klemmen-kompatibel zum „EA515-8O-1A“ und „EA515-8I“.

Die Spannungsversorgung der Ausgänge erfolgt über eine separate Anschlußklemme (Klemme 4 „Error OUT 24V“). Achtung wird die Baugruppe mit Eingängen und Ausgängen betrieben, werden die Ausgänge bei einem High-Signal an einem Eingang über diese Spannung versorgt.

## 4.4 Gehäusedesign

- Gehäusetyp: UEGM-MSTB der Fa. Phoenix Contact



## 4.5 Software

- CANopen mit speziellen Erweiterungen
- ein Empfangs- und ein Sende-PDO
- ein SDO (expedited protocol, max. 4 Nutzdatenbytes)
- Node- und Lifeguarding, Emergency Messages
- Kopplung von bis zu vier Modulen zu einem „logischen“ Modul mit 32 E/A-Leitungen (nur bei Modulen ohne A/D- und D/A-Wandler möglich)
- SYNC-Frame-Auswertung (mit Einschränkungen bei Modulkopplung)
- nur beim EA515-8I/O: Betriebsart „DS401“ oder „Lenze“ per DIP-Schalter wählbar

## 5. Digitale Ein/Ausgaben

Jeder digitale E/A-Anschluß wird per DIP-Schalter oder Konfigurations-Codestelle entweder als Ein- oder als Ausgang definiert.

Der Zustand der E/A-Anschlüsse wird jeweils mit einer gelben LED angezeigt.

### 5.1 Meldung der digitalen Eingänge

Die Übertragungsarten für die digitalen Eingänge können unterschieden werden:

- zyklisch, aber nicht synchron: Das Sende-Zeit-Intervall wird nur durch das E/A-Modul vorgegeben (im CANopen-Standard DS301 V3 noch nicht vorgesehen, für einfache Anwendungen aber sinnvoll !)
- azyklisch, synchron: Die Sendung erfolgt nach dem Empfang einer SYNC-Message (PDO - Übertragungstyp 0)
- zyklisch, synchron: Die Sendung erfolgt, nachdem jeweils eine bestimmte Anzahl SYNC-Message empfangen wurden (PDO-Übertragungstyp 1...240)
- ereignisgesteuert, asynchron: Die Sendung erfolgt, wenn sich der Zustand bestimmter digitaler Eingänge ändert (PDO-Übertragungstyp 255)

Wenn bis zu vier E/A-Module zu einem „logischen“ Modul zusammengefaßt sind, senden die E/A-Slave-Module ihre digitalen Eingangs-Informationen zunächst an den E/A-Master. Dieser faßt sie z.B. in einer 4-Byte-Message (PDO) für den Antriebsregler (bzw. für die übergeordnete Steuerung) zusammen und sendet sie anschließend über den Prozeßdatenkanal.

## 5.2 Digitale Ausgänge

Jeder Ausgangstreiber kann maximal 1 A liefern (Nennstrom). Beim Überschreiten von ca. 4 A pro Ausgang wird der entsprechende Ausgang sofort abgeschaltet und ein Fehler angezeigt.

Nach Werkseinstellung sind alle E/A-Anschlüsse nur als Eingang definiert.

Für das E/A-Modul Typ „515“ wird empfohlen, die Definition von Ein- und Ausgängen immer in „Zweiergruppen“ zu definieren, wie in Kapitel 5.2.1 beschrieben.

Um sie als Ausgang ansteuern zu können, muß erst per DIP-Schalter oder

Parameterkanal definiert werden, welcher Anschluß auch ein Ausgang sein soll (siehe Kapitel 11 ).

### 5.2.1 Einschränkungen der Definition von E/A-Richtungen bei EA515-8I/O-1A

Bei diesem E/A-Module mit 1-Ampere-Ausgängen verfügt der Ausgangstreiber für die 8 digitalen Ausgänge nur über 4 Statusleitungen, d.h. der Fehlerstatus von je zwei Ausgangsleitungen kann nicht unterschieden werden. Aus technischen Gründen muß der Fehlerstatus von als INPUT definierten Anschlüssen ignoriert werden (open-load detection des Treibers).

Wenn z.B. I/O-Anschluß 1 als digitaler Eingang und I/O-Anschluß 2 als digitaler Ausgang verwendet wird, dann können eventuell auftretende Störungen (z.B. Überlast) von I/O-Anschluß 2 nicht erkannt werden. Daher wird empfohlen, die E/A-Leitungen 1 und 2 , 3 und 4, 5 und 6 sowie 7 und 8 immer als „Zweiergruppen“ mit gleicher Richtung zu betreiben.

Beispiel für eine empfohlene Kombination:

1+2 = Eingänge, 3+4=Ausgänge, 5+6=Eingänge, 7+8=Ausgänge

Beispiel für eine nicht empfehlenswerte Kombination:

1=Ausgang, 2=Eingang, 3+4=Ausgänge, 5+6=Eingänge, 7+8=Ausgänge

In diesem Beispiel wird z.B. eine Überlastung von I/O-Anschluß 1 nicht erkannt, weil I/O-Anschluß 2 als Eingang definiert ist und die interne Statusleitung von I/O1+2 daher ignoriert werden muß.

Fehler an den Ausgängen 3,4,7,8 werden allerdings weiterhin erkannt und angezeigt.

### 5.2.2 Rücklesen der digitalen Ausgänge

Jede als Ausgang konfigurierte E/A-Leitung behält grundsätzlich auch ihre Funktion als digitaler Eingang, d.h. wird ein Ausgang geschaltet, so wird der aktuelle Zustand wie in 5.1 beschrieben übertragen.

### 5.2.3 Empfangsüberwachung der digitalen Ausgänge (für Lenze-Modus)

Die hier beschriebene Überwachung ist im Normalfall nur für den Einsatz im Lenze-Modus sinnvoll (oder wenn aus anderen Gründen kein Node-Guarding nach CANopen möglich ist):

Bei periodischer Übertragung der Telegramme für die digitalen Ausgänge („RX-PDO“) kann das E/A-Modul den Empfangszyklus überwachen.

Bleiben diese Telegramme für eine bestimmte Zeit aus (z.B. für 500ms), so schalten die betroffenen Module alle Ausgänge auf den vordefinierten Fehlerzustand (z.B. alle Ausgänge aus) und melden einen Fehler mit der roten LED. Das Modul bleibt aber im normalen Betriebszustand. Sobald wieder ein Telegramm für die digitalen Ausgänge empfangen wird, verschwindet die Fehleranzeige wieder.

Die Defaulteinstellung des entsprechenden Parameters (Codestelle 30, siehe Anhang) ist 0, d.h. bei Verwendung der Werkseinstellung findet keine Empfangsüberwachung statt.

### 5.2.4 Ausgangsstörungen

Bei Kurzschluß oder Überschreitung des maximal zulässigen Ausgangsstroms eines Kanals wird treiberintern nur der betroffene Ausgang abgeschaltet, die anderen Ausgänge werden davon zunächst nicht beeinflusst.

Der in EA-Modulen mit 1 Ampere Ausgangsstrom verwendete Ausgangstreiber signalisiert auch Fehler, wenn an seine Ausgänge statt einer nach Masse führenden Last eine externe Spannung angelegt wird. Aus diesem Grund sollte vermieden werden, an eine als Output definierte E/A-Leitung eine externe Spannung anzulegen !

Fehler an den Ausgängen werden mit der roten „Out Error“ - LED angezeigt. Außerdem wird ein Emergency-Telegramm nach dem CANopen-Standard gesendet.

## 6. Analoge Ein/Ausgaben

### 6.1 Zugriff auf die Analogwerte via CAN

Zum Einlesen und Ausgeben von Analogwerten dienen die CANopen-Objekte 0x6401 bzw. 0x6411 nach CiA DSP-401. Diese Objekte können auch im Zustand „pre-operational“ per SDO gelesen oder beschrieben werden.

Für den Normalbetrieb lassen sich die Objekte in die konfigurierbaren Prozeßdatenobjekte (PDO, vgl. Kapitel 10.2.1) einblenden. Bei der Werkseinstellung wird das PDO-Mapping bereits entsprechend eingestellt, wenn analoge E/A-Anschlüsse vorhanden sind.

### 6.2 E/A-Modul „EA 515 4 I-O1 1 A/D-D/A“

Dieses Modul bietet vier digitale Ein/Ausgänge mit einstellbarer Richtung, einen Analogeingang und einen Analogausgang mit je 12 Bit Auflösung. Beide Analoganschlüsse haben ein gemeinsames Massepotential, welches vom Massepotential der CPU und der digitalen Ein/Ausgänge galvanisch getrennt ist.

Der „manufacturer Device Name“ (Objekt 0x1008) dieses Moduls lautet „515u“ (bei Spannungs-Eingang) oder „515i“ (bei Strom-Eingang).

Dieses Modul arbeitet nicht mit „interner Kaskadierung“ (Kapitel 10.2.3).

Die folgenden Anschlußtypen sind derzeit möglich (Stand 10/97):

Analogeingang:

± 20 mA, Eingangswiderstand ca. 50 Ω

oder ± 10 V, Eingangswiderstand > 1 MΩ.

Die Umschaltung des Eingangstyps erfolgt mit einem DIP-Schalter auf der Erweiterungsplatine. Eine Umschaltung per Software ist nicht möglich (das CANopen-Objekt 0x6420 ist hier nur zur Identifikation des Eingangstyps vorhanden, ein Schreibzugriff hat keinen Effekt auf die Beschaltung des analogen Eingangs).

Analogausgang:

± 10 V.

(Strom-Ausgang in Vorbereitung.. Stand 10/97)

Weitere Informationen erhalten Sie im Systemhandbuch-Zusatz „**E/A-Modul 515: Analoge Ein- und Ausgänge**“.

## 6.3 E/A-Modul „EA 515 4 A/D“

Dieses Modul bietet vier analoge Eingänge, die wahlweise als Strom- oder Spannungseingang konfiguriert werden können (0..+10 V, -10V..+10V, 0..20 mA). Die Auflösung beträgt 12 Bit. Alle Analoganschlüsse haben ein gemeinsames Massepotential, welches vom Massepotential der CPU und der digitalen Ein/Ausgänge galvanisch getrennt ist. Dieses Modul arbeitet nicht mit „interner Kaskardierung“ (Kapitel 10.2.3).

Die folgenden Anschlußtypen sind derzeit möglich (Stand 10/97):

Analogeingang:

± 20 mA, Eingangswiderstand ca. 50 Ω  
oder ± 10 V, Eingangswiderstand > 200 kΩ.

Die Umschaltung des Eingangstyps erfolgt mit einem DIP-Schalter auf der Erweiterungsplatine. Eine Umschaltung per Software ist nicht möglich (das CANopen-Objekt 0x6420 ist hier nur zur Identifikation des Eingangstyps vorhanden, ein Schreibzugriff hat keinen Effekt auf die Beschaltung der analogen Eingänge).

Weitere Informationen erhalten Sie im Systemhandbuch-Zusatz „**E/A-Modul 515: Erweiterungsmodul mit 4 analogen Eingängen**“.

## 6.4 E/A-Modul „EA 515 4 D/A“

Dieses Modul bietet vier analoge Ausgänge, die grundsätzlich als bipolare Spannungsausgänge im Bereich –10 V bis +10 V arbeiten. Die Auflösung beträgt 12 Bit **plus** Vorzeichen. Alle Analoganschlüsse haben ein gemeinsames Massepotential, welches vom Massepotential der CPU und der digitalen Ein/Ausgänge galvanisch getrennt ist. Dieses Modul arbeitet nicht mit „interner Kaskardierung“ (Kapitel 10.2.3).

Weitere Informationen erhalten Sie im Systemhandbuch-Zusatz „**E/A-Modul 515: Erweiterungsmodul mit 4 analogen Ausgängen**“.

## 7. Spezialfunktionen und Software-Varianten

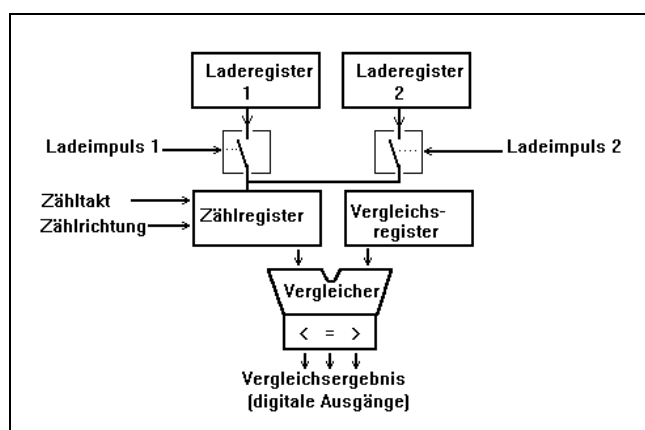
### 7.1 Impulszähler (vorläufig, 10/97)

Für besondere Anwendungen ist für die Hardware-Variante "EA515 8I-O1" einen per Software implementierter Aufwärts/Abwärts-Zähler verfügbar, der für einfache Zähl- und Positionieraufgaben verwendet werden kann.

(Siehe Beschreibung "Software-Varianten" in Kapitel 7.2 ).

Der Zählengang liegt fest auf Eingang 1. Zum Erkennung der Zählrichtung (up / down) dient Eingang 2. Bei jeder Flanke an Eingang 1 wird der Zustand von Eingang 2 abgefragt. Damit lassen sich z.B. Inkrementalgeber abfragen, solange die Frequenz an Eingang 1 unter 1 kHz liegt (unter bestimmten Bedingungen bis 4 kHz).

Ferner können digitale Eingänge als „Steuereingänge“ zum Laden bestimmter Werte in das Zählregister definiert werden. Damit kann z.B. der Zählerstand durch absolute „Wegmarken“ (Lichtschranke, Endlagenschalter o.ä.) synchronisiert werden.



Der aktuelle Zählerstand kann vom Modul ständig mit einem einstellbaren Wert verglichen werden und bestimmte Ausgangsleitungen in Abhängigkeit des Vergleichsergebnisses („kleiner, größer, gleich“) angesteuert werden.

Die Parameterierung des Impulszählers, z.B. Auswahl der verwendeten Signale erfolgt mit Codestelle C0256 (siehe Codeliste im Anhang). Bis auf den Zählerstand werden alle Einstellungen von

C0256 im EEPROM nichtflüchtig gespeichert.

Weitere Informationen erhalten Sie im Systemhandbuch-Zusatz „**E/A-Modul 515: Zählermodul**“.

## 7.2 Übersicht "Software-Varianten"

Derzeit sind die folgenden Software-Varianten für das im EPROM auf der Basisplatine verfügbar (Stand: 26.07.2000, wird laufend erweitert, bitte nachfragen !):

### "EA515-8IO-1AIN-1AOUT"

- einsetzbar in "EA515 8I-O 1A" und "EA515 1A/D-D/A"
- im "EA515 1A/D-D/A" als kalibrierbarer Analog-Eingang und Analog-Ausgang
- im "EA515 8I-O 1A" als digitales I/O-Modul betreibbar
- kein Impulszähler (zu knappe Systemressourcen)
- Projektname "EA5\_AD1\_"

### "EA515-4AIN"

- einsetzbar in "EA515 8I-O 1A" und "EA515 4A/D"
- individuell kalibrierbare Analog-Eingänge
- im "EA515 8I-O 1A" als digitales I/O-Modul betreibbar
- kein Impulszähler (zu knappe Systemressourcen)
- Projektname "EA5\_4AIN"

### "EA515-COUNTER"

- einsetzbar nur in "EA515 8I-O 1A"
- Software-Impulszähler
- auch als "normales" digitales I/O-Modul betreibbar
- kein Analog-Eingang und kein Analog-Ausgang ansteuerbar
- Projektname "EA5\_CNTR"

### "EA515-PID1"

- einsetzbar nur in "EA515 1A/D-D/A"
- enthält PID-Regler
- Istwert via Analogeingang
- Reglerausgang via Analogausgang
- Sollwertvorgabe und Parametrierung via CAN (SDO)
- auch als "normales" Analog-I/O-Modul betreibbar
- Projektname "EA5\_PID1"

### "EA515-SPS-BCD-Gateway" (in Vorbereitung, 17.8.98)

- einsetzbar nur in "EA515 8I-O"
- enthält Gateway zwischen SPS und CAN-Bus
- Daten von SPS an EA515 4-Bit-weise (BCD-codiert)
- Daten von EA515 an CAN-Gerät per SDO-Schreibauftrag
- Parametrierung derzeit (17.8.98) nur durch den Hersteller ("fest")
- Projektname "EA5\_BCDG", speziell für LENZE



## "EA515-4AOUT"

- einsetzbar in "EA515 8I-O 1A" und "EA515 4D/A"
- im "EA515 8I-O 1A" als digitales I/O-Modul betreibbar
- kein Impulszähler (zu knappe Systemressourcen)
- Projektname "EA5\_4AOUT" für Standard-Firmware
- Sondervariante mit Analog-PDO-Skalierung +-16383 für einen Anwender, der keine CANopen-konformen PDOs senden konnte (FW-Art. 11070).

## "EA515-NNMT"

- einsetzbar in "EA515 8I-O 1A"
- kann (optional) **ohne Empfang eines NMT-„Start Node“-Telegramms** in den Zustand **Operational** schalten
- im "EA515 8I-O 1A" als digitales I/O-Modul betreibbar
- kein Impulszähler (zu knappe Systemressourcen)
- Projektname "EA5\_NNMT"

## "Lenze 9374"

- einsetzbar in "EA515 8I-O 1A"
- Firmware speziell für den Einsatz am LENZE-Antriebsregler 9300
- mit Master/Slave-Kaskadierung für bis zu 32 digitale E/A-Leitungen
- enthält einen zweiten SDO speziell für Lenze's Global Drive Control

Weitere Varianten folgen....

## 8. Installation

Generell sind bei der Verdrahtung sämtliche gültigen Richtlinien (DIN, VDE) bzgl. EMV-gerechtem Aufbau, Leitungsführung, Leiterquerschnitte, zu verwendende Materialien, Mindestabstände, Blitzschutz etc. zu beachten.

In diesem Abschnitt wird nur auf einige Besonderheiten genauer eingegangen.

### 8.1 CAN-Bus-Verdrahtung

Die CAN-Bus-Verkabelung erfolgt mit Leitungen nach ISO 11898. Die folgenden elektrischen Eigenschaften werden gefordert:

Parameter	Wert			Bemerkung
	min.	nom.	max.	
Impedanz (Z)	108 $\Omega$	120 $\Omega$	132 $\Omega$	Wellenwiderstand (HF), gemessen zwischen zwei Signalleitungen
spez. Widerstand (r)		70 m $\Omega$ /m		
spez. Leitungsverzögerung		5 ns / m		muß zu den Verzögerungen der Sende- und Empfangsschaltung addiert werden

Beide Enden des CAN-Busses müssen impedanzrichtig, d.h. mit einem induktivitätsarmen 120  $\Omega$  - Widerstand abgeschlossen werden. Die E/A-Module der „515“er-Serie enthalten im Normalfall keinen Abschlußwiderstand.

Die maximal mögliche Buslänge hängt von der Baudrate (Bit-Timing) ab. Dies wird in Kapitel 10.1 näher erläutert.

## 9. Anzeigeelemente

Zur Anzeige werden 8 E/A-Zustandsanzeige-LEDs, 2 Betriebsanzeige-LEDs und 2 Fehleranzeige-LEDs verwendet:

- Betriebs-Anzeige (grün)
- Mode-Anzeige (grün)
  
- Ausgangs-Fehleranzeige (rot)
- Modul-Fehleranzeige (rot)

Im Normalbetrieb sollten nur grüne LEDs leuchten. Sobald eine rote LED leuchtet, deutet dieses auf einen Fehler oder auf einen ganz besonderen Anlagenzustand (z.B. Programmiermodus) hin.

Beim Auftreten von Fehlern **während der Initialisierungsphase** werden die Betriebs-, Mode- und Modulfehler-LED „parallel“ zum Blinken gebracht, um diese Fehlerklasse klar von „normalen“ Fehlern (die im Betrieb auftreten) zu unterscheiden.

Ursachen für dieses Fehler sind z.B.:

- falsche Einstellung des DIP-Schalters, z.B. CAN-Adresse 0
- defektes EEPROM
- sonstige Fehler beim Initialisierungs-Selbsttest

Fehler, die **während des Normalbetriebs** auftreten, werden mit den roten Fehler-LEDs angezeigt.

Solange die Betriebs-LED nicht verlischt, ist der Fehler nicht so schwerwiegend. Zu dieser Fehlerklasse gehören z.B. :

- Ausbleiben der periodischen Messages vom Antriebsregler
- Zu hohe Fehlerrate auf dem CAN-Bus, z.B. durch fehlerhaften Busabschluß
- extreme Buslast, so daß Messages nicht abgesetzt werden können
- vorübergehende Überlastung einzelner Ausgänge

Bei schwerwiegenden Fehlern verlischt die Betriebs-LED. Dazu zählen z.B.:

- CAN-Bus-OFF
- Hardwarefehler

In diesen Fällen ist das Modul nicht mehr betriebsbereit, d.h. es akzeptiert keine Befehle über den CAN-Bus mehr. Es kann dann (wenn überhaupt) nur durch Aus- und Wiedereinschalten der Modulversorgung neu gestartet werden.

Die verschiedenen LED-Kombinationen werden im folgenden Kapitel aufgelistet.

## 9.1 LED - Anzeige - Kombinationen

Betriebs-Anzeige	Mode-Anzeige	Modul-Fehler	Ausgangs-Fehler	Funktion
ON	ON	OFF	OFF	Normalbetrieb
ON	OFF (kurzzeitig)	OFF	OFF	CAN-Message empfangen
BLINKEN	ON	OFF	X	Betriebzustand nicht „Operational“ (s.U.)
OFF	ON	BLINKEN	OFF	Programmier-Modus
X	X	OFF	BLINKEN	Fehler vom Ausgangstreiber
ON	ON	BLINKEN	X	CAN-Bus-Fehler, Protokollfehler etc. (s.U.) aber Modul noch in Betrieb
OFF	ON	BLINKEN	X	CAN-Bus-OFF, Modul nicht mehr in Betrieb
BLINKEN	BLINKEN	BLINKEN	OFF	Konfigurations- oder Selbsttest-Fehler

Blink - Code	Bedeutung	Störbehebung, Hinweise
<b>Blinken der Betriebsanzeige - LED (grün)</b>		
kurz-lang-lang-kurz	Preoperational, Modul als E/A-Master konfiguriert	sicherstellen daß CAN-Anschlüsse korrekt sind, „Start Node“ senden, Lenze-Modus:Antriebsregler muß als BUS-Master arbeiten
kurz-kurz-kurz	Preoperational, Modul als E/A-Slave konfiguriert	s.O.
gleichmäßiges, langsames Blinken	Preoperational, E/A-Master / Slave-Kaskardierung abgeschaltet	s.O.

# Systemhandbuch E/A Modul 515x

Aufschlüsselung *einiger* Blink-Codes für genauere Diagnosen :

Blink - Code	Bedeutung	Störbehebung, Hinweise
<b>Blinken der Modul-Fehler-LED (rot)</b>		
lang-kurz	<u>A</u> nalog-Fehler (z.B. Übersteuerung)	Klemmenverbindungen prüfen, Kontrollieren ob analoge Eingangssignale im erlaubten Bereich liegen
lang - kurz-kurz	<u>T</u> imeout durch RX-Monitor	CAN-Bus-Verbindung überprüfen, Empfangs-Überwachung des E/A- Moduls abschalten (Codennr. 30) Lenze-Modus: periodisches Senden von CAN-OUT3 des Antriebsreglers aktivieren
lang-kurz	<u>N</u> odeguard - Fehler (Nodeguard-RTRs vom NMT-Master bleiben aus)	CAN-Bus-Verbindung überprüfen, Lifetime-Factor nach DS203-2 erhöhen, (siehe Beschreibung „Extended NMT Bootup“)
lang-kurz-lang-kurz	<u>C</u> AN-Bus-Fehler oder CAN-Bus-OFF	Baudrate bei allen Busteilnehmern korrekt ? CAN-Bus impedanzrichtig abgeschlossen ? CAN-Bus korrekt abgeschirmt ?
kurz-kurz-kurz	<u>S</u> lave-Fehler (nur wenn „interne“ Master-Slave- Kaskadierung aktiv )	Das E/A-Master-Modul hat bei einem seiner E/A-Slave-Module einen Fehler festgestellt: CAN-Bus und Versorgungs- leitungen überprüfen
(sonstige Codes)	CAL-Error-Register nicht 0	CAL-Error-Register auslesen und weiter untersuchen (Objekt 1001H, siehe CANopen)
<b>gemeinsames Blinken von „Betrieb“, „Mode“und „Error“</b>		
3 * Blinken	Selbsttest-Fehler Typ 3	Spannungsversorgung überprüfen, evtl. auch Hardware-Defekt
4 * Blinken	EEPROM-Fehler	siehe Kapitel 11.1.4, evtl. auch Hardware-Defekt
5 * Blinken	Fehler bei DIP-Konfiguration	siehe Kapitel 11.1, z.B. CAN-Adresse > 0 einstellen

## 10. CAN-Bus

Dieses Kapitel beschreibt physikalische Eigenschaften des CAN-Busses und die wichtigsten in den E/A-Modulen implementierten Protokolle, die zum Einsatz des Moduls nötig sind.

Weitere Informationen über die höheren Protokollschichten sind den CAL/CANopen - Dokumentationen „CiA Draft Standard 201 ... 207“, „CiA Draft Standard 301“ und „CiA Draft Standard Proposal 401“ zu entnehmen. In diesen Unterlagen werden das Kommunikationsprofil(301) und das Geräteprofil(401) beschrieben. Wie weit die dort beschriebenen „Objekte“ in der E/A-Modul-Familie „515x“ implementiert sind, ist den Datenblättern der speziellen Module zu entnehmen. Die grundsätzlich implementierten Objekte werden in Kapitel 12.1 aufgelistet.

Die für den Datenverkehr verwendeten CAN-Identifizierer sind im Anhang, Kapitel 13.3 aufgeführt.

### 10.1 CAN-Bus - Timing

Das CAN-Bus-Timing ist nach den CiA-Empfehlungen eingestellt (gewisse Abweichungen sind hardwarebedingt). Der CAN-Controller wird je nach eingestellter CAN-Baudrate (vgl. Kap. 11.1.2) nach der folgenden Tabelle programmiert, außerdem gelten je nach Baudrate die angegebenen maximalen Buslängen (**ohne Optokoppler**):

Baudrate	Tseg 1	Tseg 2	Tsync	max. Buslänge (Kabel: 5ns/m)
1000 kBit / s	700 ns	200 ns	100 ns	10 m
500 kBit / s	1600 ns	300 ns	100 ns	80 m
250 kBit / s	3400 ns	400 ns	200 ns	250 m
125 kBit / s	6.5 µs	1 µs	500 ns	500 m
100 kBit / s	8 µs	1.5 µs	500 ns	700 m
50 kBit / s	16 µs	3 µs	1 µs	1000 m
20 kBit / s	40 µs	7.5 µs	2.5 µs	1500 m
10 kBit / s	80 µs	15 µs	5.0 µs	2000 m

Die Größen Tsync, Tseg1 und Tseg2 beziehen sich auf die CAN-Spezifikation des verwendeten CAN-Controllers, sie bezeichnen Zeitabschnitte innerhalb eines Bits aus dem CAN-Telegramm. Dabei ist:

Tsync: Dauer des Synchronisations-Segments

Tseg1: Die Summe aus „propagation time segment“ und „phase buffer segment 1“. Tseg 1 ist die Zeit zwischen dem Sync-Segment und dem Abtastzeitpunkt eines Bits („sample point“).

Tseg2: Die „restliche“ Zeit nach dem Abtastzeitpunkt („phase buffer segment 2“) bis zum Anfang des nächsten Bits.

## 10.2 CAN-Bus-Kanäle

Für den Datenaustausch zwischen der Steuerung (z.B. Antriebsregler) und E/A-Baugruppen stehen die folgenden Kanäle zur Verfügung:

- 
- ein Prozeßdatenkanal (PDO) mit 8 bis 32 Bit, je nach Anzahl E/A-Slave-Modulen
- 
- ein Parameterkanal (SDO) zum Zugriff auf herstellerspezifische Codestellen

Je nach Erfordernissen können bis zu vier E/A-Modulen mit je 8 E/A-Anschlüssen via CAN-Bus zu einem „großen“ Modul mit 32 E/A-Anschlüssen zusammengefaßt werden. Derart zusammengefaßte Module werden in diesem Handbuch als „E/A-Master-Modul“ und „E/A-Slaves“ bezeichnet. Das E/A-Master-Modul ist nicht mit einem CANopen-Master zu verwechseln ! Für eine derartige Kopplung werden die folgenden Kanäle verwendet:

- ein interner Parameterkanal vom E/A-Master-Modul zu dessen E/A-Slave-Modulen
- maximal drei interne Parameter- und Prozeßdatenkanäle vom E/A-Slave-Modul zum E/A-Master-Modul

Vor dem Einsatz der internen Modul-Kopplung sollte geprüft werden, ob die dafür benötigten Identifier im CAN-

Netzwerk wirklich zur Verfügung stehen (sh. Kap. 13.3).



## 10.2.1 Prozeßdatenkanäle

Jedes als Master konfigurierte E/A-Modul ist über einen Prozeßdatenkanal (PDO = process data object) mit der Steuerung bzw. dem Antriebsregler verbunden.

Jeder Prozeßdatenkanal besteht aus Sicht des E/A-Moduls aus einem Empfangszweig (PDO-RX) und einem Sendezweig (PDO-TX).

Über den Prozeßdatenkanal empfängt das E/A-Modul Telegramme zum Schalten seiner digitalen Ausgänge und sendet den Zustand seiner digitalen Eingänge.

Im DS-401 - Modus kann der Inhalt der Prozeßdatentelegramme flexibel eingestellt werden. Näheres dazu im Kapitel 12.4.1, „PDO mapping“.

Im Lenze-Modus wird das PDO-Mapping bei der Defaulteinstellung so konfiguriert, daß die CAN-Telegramme den folgenden Aufbau haben:

### vom E/A-Modul im Lenze-Modus gesendetes Prozeßdaten-Telegramm (z.B. Adr. 2)

Identifizier	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7	Data 8
768+2 = 770	Eingang 1.. 8	Eingang 9..16	Eingang 17..24	Eingang 25..32	nicht benutzt (0)	nicht benutzt (0)	nicht benutzt (0)	nicht benutzt (0)

Dabei sind Eingang 1..8 die E/A-Anschlüsse des Master-Moduls, Eingang 9..16 befinden sich an Slave Nr. 1, Eingang 17.. 24 stammen von Slave Nr. 2 und Eingang 25..32 von Slave Nr. 3.

Eingang 1 wird in das niederwertige Bit von Daten 1 eingeblendet, Eingang 8 in das höchstwertige Bit usw.

Sind einige Slaves nicht vorhanden, liefern die entsprechenden Eingänge den Wert 0.

Die Initiative zum Senden der digitalen Eingangsinformationen geht im Lenze-Modus vom E/A-Modul aus, es sind ereignisgesteuerte und/oder zeitgesteuerte Sendungen möglich (siehe Kapitel 11, „Parametrierung“).

### vom E/A-Modul im Lenze-Modus empfangenes Prozeßdaten-Telegramm (z.B. Adr. 2)

Identifizier	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7	Data 8
767+2 = 769	Ausgang 1.. 8	Ausgang 9..16	Ausgang 17..24	Ausgang 25..32	nicht benutzt	nicht benutzt	nicht benutzt	nicht benutzt

Der Aufbau dieses Telegramms entspricht dem Aufbau des gesendeten Telegramms. Ein gesetztes Bit bewirkt, das der entsprechende Ausgang eingeschaltet wird (wenn er Ausgang definiert ist).

Versuche, nicht als Ausgang definierte Anschlüsse einzuschalten werden von den E/A-Modulen ignoriert.

## 10.2.2 Parameterkanal

Jedes E/A-Master-Modul kann über einen Parameterkanal konfiguriert werden. Wichtige Parameter werden vom E/A-Master-Modul auch an die E/A-Slaves weitergegeben. Eine genaue Beschreibung des Parameterkanals folgt in Kapitel 11.2.

Für Standard-Anwendungen ist die Verwendung des Parameterkanals nicht erforderlich, da die wichtigsten Parameter auch per DIP-Schalter eingestellt werden können.

Für den Parameter wird eine CANopen-kompatible SDO-Übertragung eingesetzt.

## 10.2.3 Interne Master-Slave-Kommunikation

Für die Kommunikation zwischen E/A-Master-Modul und dessen E/A-Slave-Modulen werden weitere CAN-Identifizier belegt, die dem Anhang zu entnehmen sind.

Für die CAN-Übertragung von den E/A-Slaves zum E/A-Master werden jeweils eigene Identifizier für jeden Slave verwendet (weil auf diesem Weg auch „schnelle“ Prozeßdaten übertragen werden müssen, die nicht kollidieren dürfen).

Der Aufbau der Telegramme für die Übertragung von den Slave-Modulen zum Master-Modul ist wie folgt:

Identifizier	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7	Data 8
896 ... + x	Kennung (s. U.)	Index (low)	Index (high)	Sub- Index	Nutz- daten	Nutz- daten	Nutz- daten	Nutz- daten

Als Kennung werden die folgenden Codes verwendet (in Anlehnung an den SDO-Kanal, der allerdings nichts mit dem hier beschriebenen „internen“ Parameterkanal zu tun hat):

Kennung	Bedeutung
0x00	Telegramm mit Prozessdaten und Status
0x40	Lesebefehl
0x2B	Schreibbefehl mit Daten (maximal 32 Bit)
0x4A	Antwort auf Lesebefehl mit Nutzdaten (maximal 32 Bit)
0x60	Quittierung fuer Schreibbefehl
0x80	Quittierung: "Fehler"

Für die CAN-Übertragung vom E/A-Master zu dessen E/A-Slaves wird nur ein Identifizier verwendet. Die Slave-Nummer steht in diesem Fall in den CAN-Nutzdaten, um Identifizier zu „sparen“. Auf diesem Weg werden Parameter und Prozeßdaten übertragen, die das E/A-Master-Modul nach der Analyse der PDO-Telegramme an seine E/A-Slaves weiterleitet.

Sollte die „interne“ Kommunikation von E/A-Master-Modul und E/A-Slave-Modul stören bzw. nicht benötigt sein, so kann sie auch durch eine spezielle Programmieroption mit dem DIP-Schalter abgeschaltet werden (siehe Kapitel 11.1.5.3 , Seite 35 ). Dies könnte nötig sein, wenn im CAN-Netzwerk keine Identifizier für die interne Master/Slave-Kommunikation zur Verfügung stehen.

Die als E/A-Slaves definierten Module überwachen die CAN-Verbindung zu „ihrem“ E/A-Master-Modul per „Timeout-Funktion“. Der E/A-Master sendet alle 500ms ein Telegramm an seine Slaves, die Slaves überwachen den Empfang und schalten beim Ausbleiben der Telegramme für mehr als ca. 600ms in den Fehlerzustand. Dazu gehört das Schalten der digitalen Ausgänge in den „sicheren“ Zustand.

Wird die Verbindung zwischen E/A-Master und E/A-Slave dann wiederhergestellt, läuft der Betrieb der Slaves wieder an (ohne Notwendigkeit eines Node-Resets oder ähnlichem).

## **Wichtiger Hinweis:**

Die hier beschriebene „Interne Master-Slave-Kommunikation“ ist nicht CANopen-konform, weil sie CAN-Identifizierer verwendet die laut PDO-Communication-Parameter des Moduls gar nicht existieren.

Sie wird in zukünftigen Software-Versionen (bei der Umstellung von DS301 V3.0 auf DS301 V4.0) nicht mehr enthalten sein !

Die Kaskadierung mehrerer E/A-Module zu einem „logischen“ Modul muß dann durch geschicktes PDO-Mapping ersetzt werden.

Die einzige Firmware-Variante die in Zukunft diese nicht CANopen-konforme Erweiterung enthält, ist die „Lenze 9374“-Firmware.



## 10.2.4

### 11. Parametrierung

Die Parametrierung des Moduls ist ohne externe Hilfsmittel möglich (Baudrate, CAN-Adressen).

Lediglich für die Programmierung der Ein- / Ausgänge und weiterer interner Parameter ist der Anschluß der Stromversorgung nötig.

#### 11.1 Parametrierung mit dem DIP-Schalter

##### 11.1.1 Belegung des DIP-Schalters (im Normalzustand):

Schaltnummer	Funktion	Bemerkungen
1	CAN - Adresse (Bit 0)	OFF=0, ON= 1
2	CAN - Adresse (Bit 1)	OFF=0, ON= +2
3	CAN - Adresse (Bit 2)	OFF=0, ON= +4
4	CAN - Adresse (Bit 3)	OFF=0, ON= +8
5	CAN - Adresse (Bit 4)	OFF=0, ON=+16
6	CAN - Adresse (Bit 5)	OFF=0, ON=+32
7 (*)	Master/Slave-Nummer (L) ODER CAN - Adresse (Bit 6)	OFF=0, ON= 1 OFF=0, ON=+64
8 (**)	Master/Slave-Nummer (H) ODER Reserve	OFF=0, ON= +2
9	CAN-Baudrate (Bit 0)	
10	CAN-Baudrate (Bit 1)	
11	CAN-Baudrate (Bit 2)	
12	Programmierschalter	Normalzustand „OFF“

Die CAN-Adresse ergibt sich aus der Addition der Zweierpotenzen (siehe Spalte „Bemerkungen“).

(\*) Die Belegung von Schalter 7 hängt davon ab, ob die E/A-Module mit „interner Master/Slave - Kaskardierung“ (Kap. 10.2.3) arbeiten sollen oder nicht. Die Einstellung ob mit oder ohne Kaskardierung gearbeitet wird, kann u.A. im Programmiermodus (Kap. 11.1.4) vorgenommen werden.

Bei „interner Kaskardierung“ sind daher nur 63 verschiedene Node-Identifizierer möglich, ohne „interne Kaskardierung“ dagegen 127.

(\*\*) Wird ohne „interne Master/Slave - Kaskardierung“ gearbeitet, so hat Schalter 8 derzeit noch keine Funktion. Er sollte in diesem Fall auf „OFF“ gesetzt werden, um nicht mit eventuellen zukünftigen Erweiterungen zu kollidieren.

## 11.1.2 Einstellung der CAN-Baudrate :

Schalter 9	Schalter 10	Schalter 11	CAN-Baudrate
OFF	OFF	OFF	1000 kBit / s
OFF	OFF	ON	500 kBit / s
OFF	ON	OFF	250 kBit / s
OFF	ON	ON	125 kBit / s
ON	OFF	OFF	100 kBit / s
ON	OFF	ON	50 kBit / s
ON	ON	OFF	20 kBit / s
ON	ON	ON	10 kBit / s

## 11.1.3 Testmodus (nur für den Hersteller)

Der Testmodus dient zum schnellen Test der E/A-Anschlüsse und zum Setzen der Werkseinstellung per DIP-Schalter. Aktuelle Informationen sind dem gesonderten Dokument „ea\_herst.doc“ zu entnehmen.

Um den Testmodus zu aktivieren, müssen vor Einschalten der Versorgungsspannung alle DIP-Schalter auf „ON“ geschaltet werden und dann (max. 2 Sekunden nach Power-On) einer der Schalter 8..12 wieder auf „OFF“ gestellt werden. Das E/A-Modul schaltet daraufhin in den Test-Modus um, der nur durch Abschalten der Versorgungsspannung wieder verlassen werden kann. Außerdem werden **alle EEPROM-Zellen auf die Defaultwerte gesetzt**.

Beim Setzen der Defaultwerte werden auch die „nicht veränderbaren“ Parameter „EAM\_model“ und „EAM\_options“ ins EEPROM programmiert. Diese Parameter definieren eventuell installierte Zusatzhardware (siehe unten aufgeführte Tabelle).

Im Testmodus wird mit den letzten beiden DIP-Schaltern definiert, welcher Test durchgeführt werden soll:

S11 ON, S12 OFF:

„Lauflicht“-Test aller LEDs und Zurücklesen der digitalen Eingänge (direkt nach Test-Aktivierung). Bei Rücklese-Fehlern wird die rote Modul-Fehler-LED zusätzlich angesteuert.

S11 OFF, S12 OFF: reserviert

S11 ON, S12 ON: reserviert

S11 OFF, S12 ON: „DIP“-Test: Schalter 1.. 10 werden per Software auf die Ausgänge 1.. 8 und die beiden Fehler-LEDs gekoppelt.

Bisher implementierte Hersteller-Default-Einstellungen (während Power-ON alle DIP-Schalter ON, dann nach ca 2 Sekunden...):

Schalter 12 OFF: „Lenze“-Modus, kein A/D-Wandler installiert

Schalter 11 OFF: „DS401“, kein A/D-Wandler installiert

Schalter 10 OFF: „DS401“, 1 Kanal A/D + D/A, Spannungskopplung (+- 10 V)

Schalter 9 OFF: „DS401“, 1 Kanal A/D + D/A, Stromkopplung (+-20 mA)

Schalter 8 OFF: „DS401“, Eingang 1 und 2 als Counter-Inputs (C0256)



## 11.1.4 Programmiermodus beim EA515-8I/O

Dieses Kapitel gilt nur für das „EA515-8i/O“, **nicht für Module mit analogen Ein/Ausgängen**.

Das Modul befindet sich im „Normalbetrieb“, solange der Programmierschalter (S12) in der Stellung „OFF“ steht. Durch Umschalten dieses Schalters nach „ON“ wird das Modul unabhängig vom aktuellen Betriebsmodus in den Modus „Programmieren“ gebracht.

Ist der Programmierschalter schon während Power-ON auf „ON“, so wird eine Werkseinstellung durchgeführt mit der alle eventuell gespeicherten „Sonderkonfigurationen“ überschrieben werden. Wird der Programmierschalter dagegen erst nach Power-ON auf „ON“ geschaltet, so bleiben die nur via SDO einstellbaren Parameter erhalten.

Der Programmiermodus wird folgendermaßen signalisiert:

- Verlöschen der grünen Betriebs-LED
- grüne Mode-LED permanent an
- rote Modul-Fehler-LED blinkt langsam, solange der Programmierschalter auf „ON“

Im Programmiermodus sind alle Ausgänge sicherheitshalber abgeschaltet. In dieser Phase können die Schalter S1 ... S9 entsprechend der folgenden Kapitel (11.1.5 und 11.1.5.2) eingestellt werden. Jede Schalterbetätigung wird durch ein kurzes Aufblitzen der grünen Betriebs-LED bestätigt.

Erst beim Umschalten des Programmierschalters von „ON“ nach „OFF“ wird die eingestellte Konfiguration (Definition von Ein- und Ausgängen) in einen nichtflüchtigen Speicher übernommen.

Nach erfolgreicher Programmierung wird der Blink-Takt der roten Modul-Fehler-LED kürzer. Der Programmiermodus kann aus Sicherheitsgründen nur durch Abschalten der Versorgungsspannung verlassen werden.

Vor dem Wiedereinschalten der Versorgungsspannung müssen die DIP-Schalter für die Adreßeinstellung und CAN-Baudrate (S1..S11) wieder auf die aktuellen Werte gesetzt werden !

Alle im Programmiermodus mit dem DIP-Schalter einstellbaren Parameter sind auch via SDO einstellbar (siehe Kapitel 11.2). Dazu muß allerdings ein geeignetes Konfigurationstool zur Verfügung stehen.

## 11.1.5 Belegung des DIP-Schalters im Programmiermodus<sup>1</sup>

Schaltnummer	Funktion	Bemerkungen
1	Portrichtung I/O 1	OFF = Eingang,
2	Portrichtung I/O 2	ON = Ausgang
3	Portrichtung I/O 3	..
4	Portrichtung I/O 4	
5	Portrichtung I/O 5	
6	Portrichtung I/O 6	
7	Portrichtung I/O 7	
8	Portrichtung I/O 8	
9	CAN-ID-Basis , Software-Modell	OFF = Lenze-Modus ON = DS401-Modus
10	interne E/A-Kaskardierung  (E/A-M/S-Verbund)	OFF = Kaskardierung ON = keine Kaskard.
11	„Reserve“	sollte „OFF“ bleiben !
12	Programmierschalter	Normalzustand „OFF“

### 11.1.5.1 Programmierung der Ein-/ Ausgabefunktionen

Im Programmiermodus haben die Schalter 1 bis 8 des DIP-Schalters die folgenden Sonderfunktionen:

Der Schalterzustand „ON“ definiert die entsprechende E/A-Leitung als Ausgang, der Zustand „OFF“ definiert einen Eingang.

Beim E/A-Modul „515“ sind die Hinweise in Kapitel 5.2 zu beachten !

### 11.1.5.2 Programmierung der CAN-Identifizierung und des Software-Modells

Im Programmiermodus hat Schalter 9 die folgende Bedeutung:

Schalter 9 „OFF“:

Es werden die in Kapitel 13.3 aufgeführten CAN-Identifizierung („Lenze-Modus“) verwendet.

Schalter 9 „ON“:

Es werden *zunächst* CAN-Identifizierung aus dem „Predefined Connection Set“ verwendet („DS 401“-Modus, siehe Anhang).

---

<sup>1</sup> Nicht bei Software für Analog-Modul "EA515 4A/D" und „EA515 4D/A“. Siehe Systemhandbuch-Zusatz.

### 11.1.5.3 Unterdrücken des E/A-Master/Slave-Verbundes

Ist der Verbund von E/A-Master mit drei E/A-Slaves nicht erwünscht, so kann diese Funktionalität auch unterbunden werden. Dazu dient Schalter 10 im Programmiermodus:

Schalter 10 „OFF“:

Verbund von E/A-Master-Modul mit maximal drei Slave-Modulen möglich (dies ist der Normalfall).

Schalter 10 „ON“:

kein E/A-Master-Slave-Verbund möglich (Sonderfall, nur nötig wenn der in Kapitel 10.2.3 beschriebene „interne“ Verkehr zwischen E/A-Master und E/A-Slaves stört).

## 11.1.5.4 Beispiele für Programmierung mit dem DIP-Schalter beim EA515-8I/O

### Lenze-Modus mit allen I/O-Pins als Ausgängen :

- Spannungsversorgung aus
- DIP-Schalter 1 bis 8 „ON“
- DIP-Schalter 9 bis 11 „OFF“
- DIP-Schalter 12 „ON“
- Spannungsversorgung an: rote LED blinkt langsam
- DIP-Schalter 12 „OFF“: rote LED „blitzt“ langsam (Programmierung fertig)
- Spannungsversorgung wieder aus
- vor dem nächsten Einschalten:  
Gerätenummer und Adressen wieder einstellen (siehe Seite 30) !

### DS401-Modus ohne interne E/A-Kaskardierung, alle I/O-Pins als Eingänge :

- Spannungsversorgung aus
- DIP-Schalter 1 bis 8 „OFF“
- DIP-Schalter 9 „OFF“
- DIP-Schalter 10 „OFF“
- DIP-Schalter 11 „OFF“
- DIP-Schalter 12 „ON“
- Spannungsversorgung an: rote LED blinkt langsam
- DIP-Schalter 12 „OFF“: rote LED „blitzt“ langsam (Programmierung fertig)
- Spannungsversorgung wieder aus
- vor dem nächsten Einschalten:  
Gerätenummer und Adressen wieder einstellen (siehe Seite 30) !

## 11.2 Parametrierung mit dem Parameterkanal (SDO)

Mit Hilfe eines PC's mit CAN-Interface und Konfigurationstool oder eines anderen Leitsystems können die wichtigsten internen Parameter des E/A-Moduls konfiguriert werden. Dazu steht in jedem als Master konfigurierten E/A-Modul ein SDO-Kanal zur Verfügung, der im Kapitel „CANopen“ genauer beschrieben wird.

### 11.2.1 Zugriff auf E/A-Codestellen über den Parameterkanal

Das E/A-Modul verwendet eigene „Codestellen“ zum Zugriff auf interne Parameter, die der Auflistung im Anhang zu entnehmen sind. Der Subindex wird innerhalb dieses „herstellerspezifischen Profils“ (CANopen-Bezeichnung) nicht verwendet, das entsprechende Telegrammfeld sollte auf „0“ gesetzt werden.

#### 11.2.1.1 Zeitliches Verhalten bei SDO-Schreibzugriffen

Schreibzugriffe auf E/A-Module ohne „interne Master/Slave - Kopplung“ werden innerhalb von 30 ms bestätigt, wenn *mit* „automatischer Speicherung“ gearbeitet wird.

Wesentlich schneller erfolgt die Bestätigung eines „normalen“ Schreibzugriffs, wenn *ohne* automatische Speicherung gearbeitet wird. In diesem Fall erfolgt die Bestätigung eines Schreibzugriffes innerhalb von 10ms. Sollen die Parameter auch nach Abschalten der Versorgungsspannung erhalten werden, müssen die Parameter durch einen speziellen „save“-Befehl (an CANopen-Objekt 1010h abgespeichert werden. Die Ausführungszeit des „save“-Befehls ist dann von der Anzahl geänderter Speicherzellen im EEPROM abhängig. Im ungünstigsten Fall benötigt das verwendete EEPROM bis zu 20ms pro Schreibzugriff, derart lange Wartezeiten konnten im Test allerdings nie beobachtet werden.

Genauere Informationen zum dauerhaften Abspeichern von Parametern im EEPROM siehe Kapitel 12.2.2 .

Besonders lange (bis zu 100ms) kann ein SDO-Schreibzugriff dauern, wenn mit „interner Master/Slave - Kopplung“ gearbeitet wird und Codestellen beschrieben werden, die das E/A-Master-Modul erst an seine E/A-Slaves weitergeben muß. Die dabei intern ablaufende Prozedur wird im nächsten Kapitel beschrieben.

## 11.2.1.2 Interner Ablauf bei SDO-Schreibzugriffen

Das folgende Beispiel soll den „kompliziertesten“ internen Ablauf bei SDO-Schreibzugriffen mit „interner Master/Slave - Kopplung“ verdeutlichen:

### Beispiel 1: Definition von 32 Ein-/Ausgängen über den Parameterkanal

Alle E/A-Leitungen des Master-Moduls und der drei daran angehängten Slave-Module sollen als Ausgänge definiert werden. Das Master-Modul hat die Geräteadresse 2.

Im Anhang findet sich die dafür zuständige Codestelle 10 (Portrichtung).

Daraus ergibt sich das folgende vom Leitsystem zu sendende CAN- (SDO-)Telegramm:

Identifizier	Befehls - code	Index ( low )	Index (high)	Sub-index	Data 1	Data 2	Data 3	Data 4
1536 + 2 = 1538	(write) 0x23	0xFF-10 = 0xF5	0x5F - 0 = 0x5F	0x00	(Master) 0xFF	(Slave1) 0xFF	(Slave2) 0xFF	(Slave3) 0xFF

Dieses Telegramm wird nur vom E/A-Master-Modul mit der Geräteadresse 2 empfangen. Weil dieses Telegramm aber auch Daten für die an den Master angeschlossenen Slaves enthält, sendet der Master daraufhin an „seine“ Slaves entsprechende Telegramme über den internen Master->Slave-Kanal (Identifizier 1664+2) und wartet die Antwort der Slaves ab.

(Für den Fall, daß nicht alle drei Slaves vorhanden sind, bricht der Master das Warten nach einer bestimmten Zeit (50ms) ab, meldet aber keinen Fehler !).

Die Slaves speichern ihre individuellen Portrichtungs-Definitionen im nichtflüchtigen Speicher ab und bestätigen dies nach erfolgtem Speichern über die Slave->Master-Kanäle.

Anschließend speichert auch der Master seine eigene „neue“ Portrichtungs-Definition im nichtflüchtigen Speicher. Auch dieser Vorgang braucht eine gewisse Zeit (max. 20 ms). Erst nach Abschluß der Programmierung, meldet das Master-E/A-Modul die Bestätigung an das Leitsystem:

Identifizier	Befehls - code	Index ( low )	Index (high)	Sub-index	Data 1	Data 2	Data 3	Data 4
1408 + 2 = 1410	(w.resp) 0x23	0xFF-10 = 0xF5	0x5F - 0 = 0x5F	0x00	0x00	0x00	0x00	0x00

Die tatsächliche Reaktionsdauer kann nur abgeschätzt werden, weil sie z.B. von folgenden Faktoren abhängt:

- CAN-Bus-Baudrate
- CAN-Bus-Last und Anteil höher priorisierter CAN-Telegramme
- Belastung der CPUs in den E/A-Modulen durch andere Aufgaben, z.B. schnelle Zustandswechsel an den digitalen Eingängen.

## 11.2.2 Hinweise zur Verwendung des Parameterkanals

Generell ist bei der Verwendung der SDOs zu beachten:

- SDOs dienen zur Konfiguration der E/A-Module. Sie sind nicht dafür vorgesehen, im laufenden („Normal“-) Betrieb ständig interne Parameter zu ändern. Dies würde auch auf Kosten der Lebensdauer der zur Speicherung verwendeten EEPROMs gehen !
- Beim Austausch eines per SDO konfigurierten Moduls bzw. Einsatz des Moduls an einem anderen Ort sollte sicherheitshalber die Konfiguration wieder auf die Werkseinstellung gesetzt werden. Dies gilt insbesondere für die oben demonstrierte Änderung der E/A-Richtungen.
- Bei Parametern, die per DIP-Schalter *und* per SDO konfiguriert werden können, gilt grundsätzlich die zuletzt durchgeführte Einstellung.
- Umstellungen von Identifier-“Basisadressen“ durch Schreibzugriffe auf Codestellen ab 50 werden erst nach Neu-Initialisierung der Module z.B. durch Netzwerk-Management oder Abschalten der Versorgungsspannung wirksam. Die Möglichkeit des „Umdefinierens“ der CAN-ID-Basisadressen sollte nur verwendet werden, wenn die Einstellmöglichkeit mit Hilfe des DIP-Schalters nicht ausreicht !
- Der Zugriff auf die Codestellen ist nicht vom Zustand des Moduls (Operational, Preoperational) abhängig.
- Beim Zusammenschalten von mehreren Modulen zu einem „logischen“ Modul sollten die E/A-Slave-Module auch bei der Parametrierung des E/A-Master-Moduls angeschlossen sein, weil das E/A-Master-Modul einige Parameter bei SDO-Zugriffen über den internen Parameterkanal mit seinen E/A-Slave-Modulen austauscht (z.B. die Portrichtungs-Definitionen).
- Nur nach dem Auftreten schwerwiegender Fehler ist kein Zugriff per SDO mehr möglich.
- Das automatische Speichern der Parameter im EEPROM bei jedem Schreibzugriff kann per Zugriff auf eine hersteller-spezifische Codestelle abgeschaltet werden. Dies kann in Objekt 1010h im CANopen-Kommunikationsprofil abgefragt werden. Nach der Werkseinstellung via DIP-Schalter ist das „autonome“ Abspeichern der Parameter allerdings aktiv, d.h. es braucht kein „save“-Befehl gesendet zu werden.

## 12. CANopen

Dieses Kapitel enthält einige Grundlagen zum Thema „CANopen“ und die wichtigsten Informationen über die in der E/A-Modul-Familie 515 implementierten Funktionen.

Eine komplette CANopen-Beschreibung läge außerhalb des Rahmens dieses Handbuchs.

Der interessierte Leser sei auf die nicht ganz einfache CAL / CANopen - Dokumentation verwiesen !

### **Wichtiger Hinweis:**

**Das hier beschriebene „CANopen“ bezieht sich auf die mittlerweile veralteten Standards DS301 V3.0 und DS401 V1.4.**

**MKT wird voraussichtlich im 2. Halbjahr 2000 auf DS301 V4.0 und DS401 V1.4 umstellen, wodurch nicht nur neue Funktionalitäten dazukommen sondern auch einige alte Funktionalitäten entfallen bzw. ersetzt werden !**

**(Sommer 2001: die Umstellung erfolgte bislang nur für das Modul „EA515-4A/D“)**

### 12.1 Das CANopen-Objektverzeichnis

Das Objektverzeichnis ist im wesentlichen eine (sortierte) Gruppierung von Objekten, auf die über das Netzwerk zugegriffen werden kann. Jedes Objekt in diesem Verzeichnis wird mit einem 16-Bit-Index adressiert. In den Objektverzeichnissen wird der Index in hexadezimaler Form angegeben.

Der Index kann 16-Bit-Parameter nach der CanOpen-Spezifikation (CiA-Draft DS301, DS401) oder ein herstellerspezifischer Code sein. Anhand der höherwertigen Bits des Index wird festgelegt, zu welcher Objektklasse der Parameter gehört.

Zum Objektverzeichnis gehören unter anderem:

Index (hex)	Objekt	Beispiel
0001 ... 009F	Definition von Datentypen	
1000 ... 1FFF	Commuikation Profile Area	1001 = Fehler-Register
2000 ... 5FFF	Manufacturer Specific Profile Area	5FF5 = Portrichtungen
6000 ... 9FFF	Standardised Device Profile Area	(hier : nach DS401) 6000 = Zustand von 8 Eingängen
A000 ... FFFF	reserviert	



## 12.1.1 Kommunikationsprofil

(Communication Profile Area)

Innerhalb dieses Abschnittes sind folgende Objekte implementiert (Stand 8/97) :

Index (hex)	Name	Subindizes	Typ	Zugriff
1000	Device Type	-	Unsigned32	ro
1001	Error Register	-	Unsigned8	ro
1003	Predef'd Error.. (Error History)	variabel	Unsigned32	R, W (nur Subindex 0)
1004	Number of PDOs	0,1,2	Unsigned32	ro
1005 (***)	COB-ID of Sync Message	-	Unsigned32	rw (*)
1008	Manufacturer's Device Name	-	Visible String	ro
1009	Manufacturer's Hardware Version	-	Visible String	ro
100A	Manufacturer's Software Version	-	Visible String	ro
100B (**)	Node-ID	-	Unsigned32	ro (=DIP-switch)
100C	Guard Time	-	Unsigned16	rw (*)
100D	Life Time Factor	-	Unsigned8	rw (*)
100E (**)	Node Guarding ID	-	Unsigned32	rw (*)
1010	Store Parameters	0,1	Unsigned32	rw
1011	Restore Default Parameters	0,1	Unsigned32	rw
1400	Receive PDO Communication Parameter	0,1,2	PDOCommPa r	rw
1600	Reveive PDO Mapping Parameter	0, 1..max.8	PDOMapping	rw
1800	Transmit PDO Communication Parameter	0,1,2	PDOCommPa r	rw
1A00	Transmit PDO Mapping Parameter	0, 1.. max.8	PDOMapping	rw

(\*) diese Parameter **sollten** nur mit Hilfe des „Extended NMT Bootups“ gesetzt werden.

(\*\*) Dieses Objekt ist in DS301 V4.0 nicht mehr vorhanden und wird bei der Umstellung der Firmware von DS301 V3.0 auf DS301 V4.0 voraussichtlich entfallen.

(\*\*\*) Die Funktion dieses Objekts wird sich (zumindest teilweise) bei der Umstellung von DS301 V3.0 auf DS301 V4.0 ändern und könnte zu Problemen führen.

## 12.1.2 Geräteprofil

(Standardised Device Profile Area)

Die hier implementierten Objekte basieren auf dem „CiA Draft Standard Proposal 301“ Version 3.0 und „DS401 V1.4“. Einige Objekte, z.B. die 16- und 32-Bit-Zugriffs-Objekte sind nur verfügbar wenn mehrere E/A-515-Module mit je 8 E/A-Anschlüssen zu einem „Verbund“ gekoppelt sind.

## 12.1.2.1 Digitale Ein- und Ausgänge

Index (hex)	Name	Subindizes	Typ	Zugriff
6000	Read State [of] 8 Input Lines	0,1 .. max.4	Unsigned8	ro
6006	Input Interrupt Mask [for] 8 Input Lines, any change	0, 1.. max.4	Unsigned8	rw
6020	Read State [of] 1 Input Line	0, 1.. max.32	Unsigned8, Boolean	ro
6100	Read State [of] 16 Input Lines	0, [1, 2]	Unsigned16	ro
6120	Read State [of] 32 Input Lines	0, max. 1	Unsigned32	ro
6200	Write State [for] 8 Output Lines	0, 1..max.4	Unsigned8	rw
6206	Fault Mode [for] 8 Output Lines	0, 1..max 4	Unsigned8	rw
6207	Fault State [for] 8 Output Lines	0, 1..max 4	Unsigned8	rw
6220	Write State [for] 1 Ouput Line	0, 1.. max.32	Unsigned8, Boolean	rw
6250	Fault Mode [for] 1 Output Line	0, 1..max.32	Unsigned8, Boolean	rw
6260	Fault State [for] 1 Output Line	0, 1..max.32	Unsigned8, Boolean	rw
6300	Write State [for] 16 Output Lines	0, 1..max.2	Unsigned8, Unsigned16	rw
6306	Fault Mode [for] 16 Output Lines	0, 1..max 2	Unsigned8, Unsigned16	rw
6307	Fault State [for] 16 Output Lines	0, 1..max 4	Unsigned8, Unsigned16	rw
6320	Write State [for] 32 Output Lines	0, max.1	Unsigned8, Unsigned32	rw
6326	Fault Mode [for] 32 Output Lines	0, max 1	Unsigned8, Unsigned32	rw
6327	Fault State [for] 32 Output Lines	0, max 1	Unsigned8, Unsigned32	rw

## 12.1.2.2 Analoge Ein- und Ausgänge

(nur vorhanden wenn ein Analog-Erweiterungsmodul installiert ist)

Index (hex)	Name	Subindizes	Typ	Zugriff
6401	Reads value of the input channel (not converted)	0 1 [... 4]	Unsigned 8 Unsigned16	ro
6411	Writes value of the output channel (not converted)	0 1 [... 4]	Unsigned8 Unsigned 16	rw
6420	Set Analogue Input Range (hier: nur lesbar)	0 1 [... 4]	Unsigned8 Unsigned 16	rw
6421	Determines which events cause an interrupt for a specific channel	0 1 [... 4]	Unsigned 8 Unsigned 8	rw
6423	Globally enable/disable Interrupt	0	Boolean	rw
6424	When enabled, interrupt triggered when analogue input rises above this value (not converted)	0 1 [... 4]	Unsigned8 Unsigned 32	rw
6425	When enabled, interrupt triggered when analogue input falls below this value (not converted)	0 1 [... 4]	Unsigned8 Unsigned 32	rw
6426	When enabled, interrupt triggered when analogue changes by more than thisvalue from previous reading (rising or falling) (not converted)	0 1 [...4]	Unsigned8, Unsigned 32	rw
6443	Output Fault Mode	0,1	Unsigned8	rw
6444	Default Output Fault value (unconverted)	0 1	Unsigned 8 Unsigned 32	rw

## 12.1.3 Hersteller-spezifisches Profil

(Manufacturer Specific Profile Area)

Wichtig ist der „Manufacturer Specific Profile“ - Bereich von 0x2000-0x5FFF, in der spezielle Systemparameter gelesen und gesetzt werden können.

Im E/A-Modul „515“ berechnet sich der Index nach der Formel

$$\text{CANopen-Objekt-Index} = 0x5FFF - \langle \text{Codestelle des E/A-Moduls} \rangle$$

Die "Codestellen" des E/A-Moduls sind im Anhang, Kapitel 14 aufgeführt.

Das System der Codestellen (statt CANopen-Objekt-Indizes) hat historische Gründe, die hier nicht genauer erläutert werden sollen.

## 12.2 Service-Daten-Objekte (SDOs)

Service-Daten-Objekte dienen zum Zugriff auf das Objektverzeichnis eines Gerätes.

Ein SDO stellt daher einen Kanal zum Zugriff auf die Parameter des Gerätes dar.

Der Zugriff über diesen Kanal ist beim E/A-Modul 515 im Zustand „Operational“ und „Preoperational“ möglich. Die Hinweise aus Kapitel 11.2.2 sind zu beachten !

### 12.2.1 Aufbau der SDO - Telegramme

11 Bit ID

8 Byte Nutzdaten

	Befehls - code	Index ( low )	Index (high)	Sub- index	Data 1	Data 2	Data 3	Data 4
--	-------------------	------------------	-----------------	---------------	--------	--------	--------	--------

Der Befehlscode setzt sich unter anderem aus dem Command Specifier und der Länge zusammen. Häufig benötigte Kombinationen sind z.B.:

0x40 = 64 dez: Read Request, d.h. ein Parameter soll gelesen werden

0x23 = 35 dez: Write Request mit 32 Bit Daten, d.h. ein Parameter soll gesetzt werden

Das maximal 4 Byte lange Datenfeld ist grundsätzlich nach der Regel „niederwertiges Byte zuerst, höherwertiges Byte zuletzt“ aufgebaut. Dabei steht das niederwertige Byte **immer** in „Data 1“, bei 16-Bit-Werten steht das höchstwertige Byte (Bits 8..15) in „Data 2“, und bei 32-Bit-Werten steht das MSB (Bits 24..31) in „Data 4“.

Das E/A-Modul antwortet auf **jedes** empfangene Telegramm mit einem Antworttelegramm.

Das Antworttelegramm kann folgende Befehlscodes enthalten:

0x42 = 66 dez: Read Response, dieses Telegramm enthält den gewünschten Parameter

0x60 = 96 dez: Write Response, d.h. ein Parameter wurde erfolgreich gesetzt

0x80 = 128 dez: Error Response, d.h. das E/A-Modul meldet einen Kommunikationsfehler.

Weitere Informationen sind den CiA-Spezifikationen oder einer noch zu verfassenden „Programmierer-Information“ zu entnehmen (Stand: 8/97).

## 12.2.2 Nichtflüchtiges Speichern von Parametern

Es ist beim EA515 möglich, die Parameter bei jedem Schreibvorgang in einem nichtflüchtigen Speicher abgelegt, ohne daß dazu ein spezieller „save“-Befehl an das Modul gesendet werden muß. Die Antwort auf ein SDO-Telegramm erfolgt dann erst nach erfolgter Aktualisierung des nichtflüchtigen Speichers. Diese Art des Speicherns ist beim Umprogrammieren mehrerer Parameter sehr langwierig, weil jedesmal die Prüfsumme im EEPROM neu berechnet und aktualisiert werden muss

Im Normalfall erfolgt das Speichern daher nur „auf Kommando“. Dazu wird ein standardisiertes Objekt nach „CANopen“ verwendet (Objekt 1010h). Mit Objekt 1010h kann leider nicht zwischen „automatischem Speichern“ und „Speichern auf Kommando“ umgeschaltet werden, es kann daran nur abgelesen werden ob das Modul „automatisch“ speichert oder nicht<sup>1</sup>. Außerdem dient Objekt 1010h zum Auslösen der „SAVE“-Funktion (siehe Unten).

Der Inhalt von Objekt 1010h zeigt den aktuellen Speicher-Modus an:

- Ist Bit 1 von Objekt 1010h, Subindex 1 gesetzt, so speichert das E/A-Modul die Konfiguration nach jeder Änderung via SDO-Kanal automatisch.
- Ist Bit 1 von Objekt 1010h, Subindex 1 nicht gesetzt, so speichert das E/A-Modul die Konfiguration nicht „automatisch“, sondern muß durch Schreiben der Zeichenkette „save“ (73h 61h 76h 65h, Reihenfolge aus CAN-Telegramm) in das Objekt 1010h, Subindex 1 extra dazu veranlaßt werden.

Zum dauerhaften Abspeichern der Konfiguration nach CANopen senden Sie die folgende Sequenz per SDO-Kanal (hier für Knoten-Nummer „2“):

Identifizier	Befehls- code	Index ( low )	Index (high)	Sub- index	Data 1	Data 2	Data 3	Data 4
1408 + 2 = 1410	(w.req.) 0x22	0x10	0x10	0x01	„s“= 0x76	„a“= 0x61	„v“= 0x76	„e“= 0x65

Dieses Telegramm wird nach erfolgreicher Speicherung der Parameter vom Modul mit einem „Write-Response“-Telegramm (Befehlscode 0x60) bestätigt.

<sup>1</sup> bei „alten“ Varianten des EA515's wie „EA515-8 I/O“ war die Default-Einstellung „automatisch Speichern“. Sie sollten allerdings zur Sicherheit davon ausgehen, daß sich dies in Zukunft ändern wird !

Anmerkung: Der CANopen-Standard sieht keine Möglichkeit zum „Umschalten“ des Speicher-Modus vor. Die Umschaltmöglichkeit mit Codestelle 17 wurde für eine größtmögliche Flexibilität der E/A-Module vorgesehen.  
Wenn bei jedem Hochfahren eines Netzwerkes viele Schreibzugriffe per SDO durchgeführt werden, ist es ratsam, die „automatische“ Speicherung durch Null-Setzen von Codestelle 17 abzuschalten und das Aktualisieren des EEPROMS nach der Initialisierung „auf einmal“ durchzuführen. Dadurch wird dem E/A-Modul unter anderem das ständige Neu-Berechnen der EEPROM-Prüfsumme erspart, und die vielen SDO-Schreibzugriffe werden erheblich schneller bestätigt (vgl. Kapitel 11.2.1.1).  
Das verwendete EEPROM hat eine Lebensdauer von 1 Mio. Lösch/Schreib-Zyklen.

## 12.2.3 Wiederherstellen der Default-Einstellungen

Um die Konfiguration eines nicht „fabrikfrischen“ Moduls wieder auf den Ursprungszustand zu setzen, bietet sich das CANopen-Objekt 0x1011 an („Restore default parameters“).

Im einfachsten Fall setzen sie **alle** Parameter wieder auf Defaultwerte, indem Sie folgendes CAN-Telegramm per SDO an das Modul schicken (hier für Knoten-Nr. „2“):

Identifizier	Befehls-code	Index ( low )	Index (high)	Sub-index	Data 1	Data 2	Data 3	Data 4
--------------	--------------	---------------	--------------	-----------	--------	--------	--------	--------

1408 + 2 = 1410	(w.req.) 0x22	0x11	0x10	0x01	„l“= 0x6C	„o“= 0x6F	„a“= 0x61	„d“= 0x64
--------------------	------------------	------	------	------	--------------	--------------	--------------	--------------

Dieses Telegramm wird nach erfolgreichem Wiederherstellen der „default“-Parameter vom Modul mit einem „Write-Response“-Telegramm (Befehlscode 0x60) bestätigt.

Eine genauere Beschreibung der Möglichkeiten, die Ihnen dieses Objekt bietet, entnehmen Sie bitte der CANopen-Dokumentation CiA DS301, Beschreibung von Objekt 0x1011 .

## 12.3 NMT-Boot-up

### 12.3.1 Einfaches Boot-Up

Die E/A-Module der „515“er-Serie können mit dem im CiA-Draft Standard 301 in Kapitel 8.3 beschriebenen „Minimum Capability Device“ - Boot-up initialisiert werden. Im einfachsten Fall reicht nach dem Einschalten ein Telegramm zum Umschalten vom Zustand „Pre-Operational“ in den Zustand „Operational“. Dazu reicht es im einfachsten Fall, an den CAN-Identifizier 0000h das 2-Byte-Telegramm „01 00“ zu senden (=Start Remote Node).

### 12.3.2 Extended Boot-Up

#### **Hinweis:**

**Das Extended Boot-Up wird bei der Umstellung von DS301 V3.0 auf DS301 V4.0 vermutlich *entfallen*, weil auch das Nodeguarding entfallen wird ! Sie sollten auf das Node-Guarding daher verzichten...**

Zusätzlich ist auch das „Extended Boot-up“ mit Nodeguarding-Aktivierung möglich.

Der NMT-Master kann dazu die auf der folgenden Seite skizzierte Sequenz an den NMT-Slave senden.

( hier: Modul-ID = 1, COB-IDs dezimal, CAN-Daten hexadezimal ):

Die in den CAN-Telegrammen eingetragenen Parameter haben die folgende Bedeutung:

Node-ID: „Knoten-Identifizier“, beim EA515 ist das die per DIP-Schalter eingestellte „CAN-Adresse“ im Bereich 1..63 (ohne „E/A-Kaskardierung“ 1 .. 127).

req.Guard Time: vom Slave angeforderte Überwachungszeit in Millisekunden. Das EA515 fordert keine Überwachungszeit an, der Eintrag ist daher 0.

req.Life Time Factor: vom Slave angeforderter „life time factor“. Wird durch den „assigned life time factor“ überschrieben.

node class+flags: gibt die Fähigkeiten des E/A-Moduls als NMT-Slave an. Beim EA515 „2“. In Bit 7 dieses Bytes könnte der NMT-Slave einen „Konfigurations-Download“ anfordern. Diese Möglichkeit besteht beim EA515 nicht, Bit 7 ist daher immer „0“.

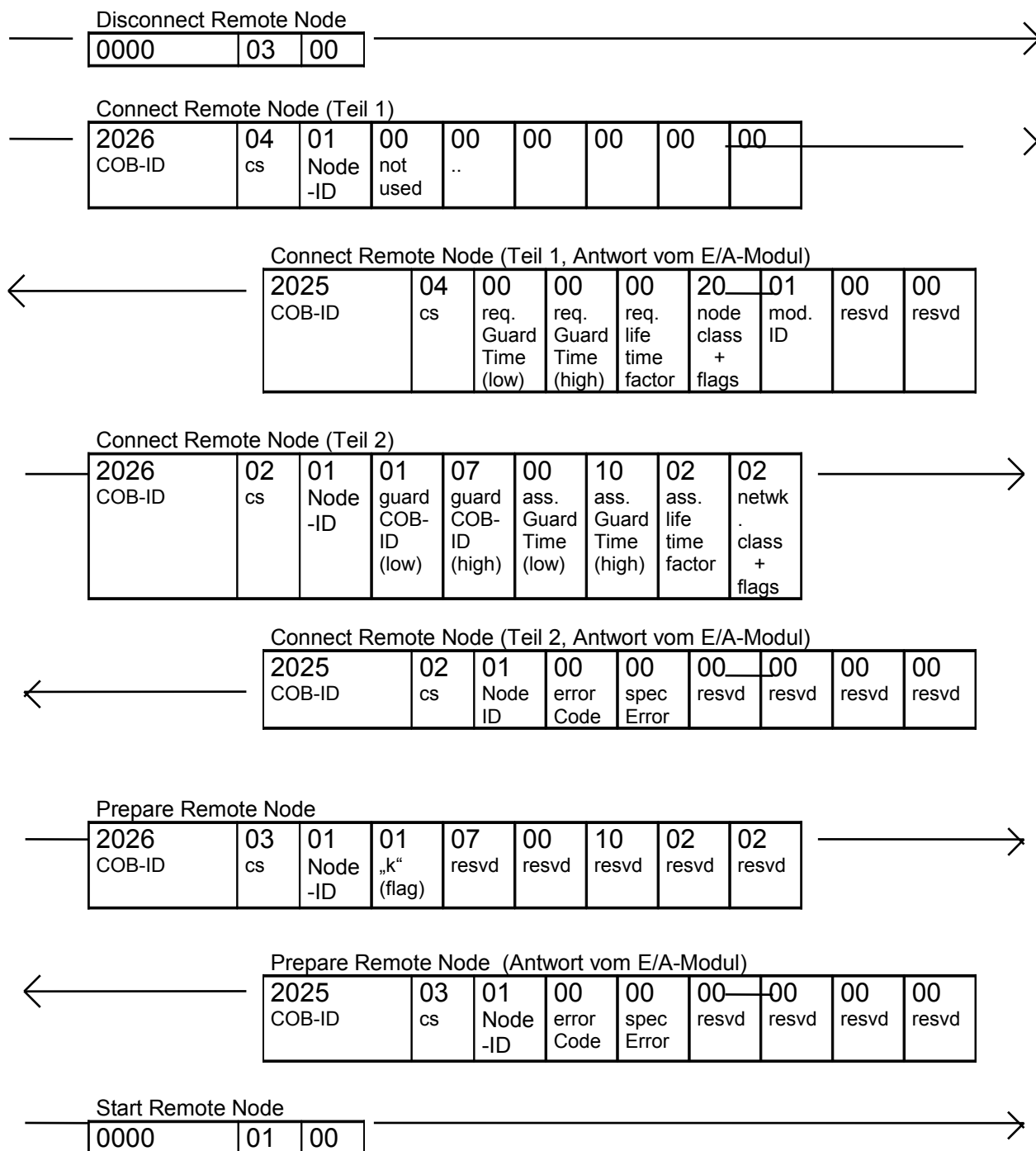
mod. ID: Beim EA515 identisch mit dem Node-ID.

guard COB-ID: Der vom Master an den Slave zugewiesene CAN-Identifizier für das Nodeguarding-Protokoll. Muß zwischen 1761 und 2015 liegen.



NMT-Master

NMT-Slave



Nach dieser Prozedur befindet sich das E/A-Modul im Zustand „Operational“ und ist bereit zum Senden und Empfangen von Prozeßdaten-Objekten. Außerdem ist das Node-Guarding nun aktiv, so das der „Node-Guarding-Master“ von nun an periodisch per RTR-Telegramm den Modulzustand anfordern **muß** (näheres dazu im Kapitel „Node Guarding“).

### 12.4 Prozeßdatenobjekte (PDOs)

Für die Übertragung von Prozeßdaten dienen die PDOs.

Im „Receive“-PDO werden i.A. die Daten für die digitalen Ausgänge vom Steuerungssystem (z.B. Lenze-Umrichter) an die E/A-Module übertragen.

Im „Transmit“-PDO sendet das E/A-Master-Modul (welches im CANopen-Netz selbst immer SLAVE ist !) den Zustand seiner digitalen Eingänge an das Steuerungssystem (oder -in besonderen Fällen- an andere E/A-Geräte).

Der Aufbau eines PDOs wurde für ein einfaches Beispiel (Lenze-Antriebsregler) bereits in Kapitel 10.2.1 („Prozeßdatenkanäle“) vorgestellt. Für den allgemeinen Anwendungsfall kann der Aufbau eines PDO-Telegramms aber (fast) beliebig zusammengestellt werden. Dazu dient das im folgenden Kapitel beschriebene „PDO-Mapping“.

## 12.4.1 PDO - Mapping

Das variable PDO-Mapping dient zum „Einblenden“ bestimmter Objekte aus dem CANopen-Objekt-Verzeichnis in ein maximal 8 Byte langes Prozeßdaten-Telegramm. Dafür ist für den Receive-PDO (digitale Eingänge) das Objekt 1800h und für den Transmit-PDO (digitale Ausgänge) das Objekt 1A00h zuständig. Beide genannten Objekte haben den gleichen prinzipiellen Aufbau. Diese „PDO Mapping Parameter“ - Objekte entsprechen einer Liste, in der alle im PDO abgebildeten Objekte stehen. Jeder Listeneintrag enthält den Index eines abgebildeten Objektes, den Subindex des abgebildeten Objekts und die Anzahl der von diesem Objekt im PDO eingeblendeten Datenbits.

### Beispiel:

Im Transmit-PDO soll nur der Zustand der ersten 8 digitalen Eingänge eines E/A-Moduls abgebildet werden.

Das Objekt, welches die digitalen Eingänge in Gruppen von je 8 Eingängen (also „byteweise“) enthält, ist das Objekt 6000h aus „DS 401“ (z.Z. nur „DSP 401 V1.4“, Kapitel 8.1.1).

Subindex 1 von Objekt 6000h enthält immer den Zustand der ersten 8 digitalen Eingänge, es könnte notfalls auch periodisch per SDO ausgelesen werden. Um dieses Objekt in den PDO „hineinzumappen“, müssen Index, Subindex und Bit-Anzahl wie folgt zu einem 32-Bit-Wert („Unsigned 32“) zusammengefaßt werden:

MSB, Bits 31.. 24	Bits 23..16	Bits 15...8	LSB, Bits 7... 0
Index (High)	Index (Low)	Subindex	Anzahl Datenbits
60	00	01	08

Bevor dieser 32-Bit-Wert per SDO in das Mapping-Objekt 1A00h geschrieben werden kann, muß allerdings die „Listenlänge“ des Mapping-Objektes mindestens auf „1“ gesetzt werden. Dies erfolgt durch Schreiben einer „1“ in das Mapping-Objekt 1A00h, Subindex 0. Damit ist in der Liste Platz für genau einen Eintrag, der anschließend an Subindex 1 geschrieben wird. Bei dem Versuch auf ein Listenobjekt zuzugreifen welches nicht in die Liste paßt (wegen des Inhalts von Subindex 0), erfolgt ein „Error Response“ beim SDO-Zugriff auf das Mapping-Objekt.

Nach der erfolgreichen Programmierung des Mapping-Objektes hat dieses den folgenden Inhalt:

Objekt 1A00h, Subindex 0:

01h (d.h. ein Eintrag in der Liste)

Objekt 1A00h, Subindex 1:

60000108h (d.h. bilde 8 Bits von Objekt 6000, Subindex 1 im PDO ab)

Durch diese Programmierung wird das Prozeßdaten-Telegramm nur ein Byte enthalten, in dem der aktuelle Zustand der ersten acht digitalen Eingänge steht.

Sollen in den noch „freien“ 7 Bytes des Prozeßdaten-Telegramms weitere Informationen stehen, so ist zunächst Objekt 1A00h Subindex 0 zu erhöhen (z.B. auf 4). Anschließend können weitere Listeneinträge in Objekt 1A00 Subindizes 2,3,4 vorgenommen werden. Bei den E/A-Modulen „515x“ können maximal acht verschiedene Objekte in einem PDO abgebildet werden. Die Anzahl der Datenbits muß aus Geschwindigkeitsgründen 8 oder ein ganzzahliges Vielfaches davon sein.

## 12.4.2 PDO - Übertragungsarten

### 12.4.2.1 Synchroner Übertragungsarten

Die synchronen Übertragungsarten sind verwendbar, wenn ein Teilnehmer im CANopen-Netzwerk die entsprechenden SYNC-Telegramme erzeugen kann (Das E/A-Modul „515“ kann diese Telegramme zur Zeit [9/97] nur empfangen, aber nicht selber erzeugen).

Die synchrone Übertragungsart wird durch den „PDO transmission type“ im Kommunikationsparameter des entsprechenden Prozeßdatenobjekts definiert.

Ein „transmission type“ von 5 bedeutet z.B., daß nach jeweils fünf empfangenen SYNC-Messages ein Prozeßdatenobjekt gesendet wird. Details sind dem CiA Draft Standard 301 zu entnehmen.

### 12.4.2.2 Ereignisgesteuerte Übertragung

Jeder Änderung eines digitalen Einganges an Master oder Slaves kann eine Meldung der digitalen Eingänge auslösen.

Nach Werkseinstellung erfolgt dies durch alle bestückten Eingänge, durch Ändern einer Ereignis-Maske kann dies geändert werden (siehe Anhang „Codetabelle“ oder DS401). Wird diese Ereignis-Maske auf 0xFFFFFFFF gesetzt, lösen alle Eingänge eine Meldung aus. Enthält diese Maske 0x00000000, so findet keine ereignisgesteuerte Übertragung statt.

Um eine zu hohe Buslast durch ereignisgesteuerte Übertragung zu vermeiden, sind zwei Mechanismen implementiert:

- Entprellung der digitalen Eingänge (mit einstellbarer Entprellzeit)
- Definition einer „Mindest-Pausenzeit“ die zwischen dem Senden zweier ereignisgesteuerter(!) Prozeßdaten-Telegramme vergehen muß.

Die Defaultwerte sind:

- Entprellzeit 1 ms
- keine Mindest-Pause zwischen zwei ereignisgesteuerten Sendungen

## 12.4.2.3

### 12.4.3 Zyklische asynchrone Übertragung

Die Eingangsinformation kann zyklisch (z.B. alle 100 ms) mit dem Prozeßdatenkanal übertragen. Die Sendezykluszeit kann über den Parameterkanal geändert oder auch abgeschaltet werden.

Sie stellt eine **zusätzliche** Übertragungs-Initiative zu den im „CANopen“-Standard definierten Möglichkeiten dar, die speziell für den Einsatz am Lenze-Antriebsregler vorgesehen ist. In der CANopen-Terminologie könnte man diese Übertragungsart als „zyklisch aber nicht vom SYNC-Objekt abhängig“ bezeichnen, sie ist im CANopen-Standard aber nicht(?) vorgesehen .

Defaultwert:

- „Lenze-Modus“: Zyklische Sendung alle 100 ms
- „DS 401-Modus“: Zyklische Sendung abgeschaltet (0), Übertragung nur vom „PDO-Transmission Type“ abhängig.

Die Aktivierung dieser Übertragungsart ist durch Ändern von Codestelle 14 möglich (im herstellerspezifischen Objektbereich, siehe Anhang).

## 12.5 Node Guarding / Life Guarding

Zur gegenseitigen Überwachung von Steuerung und den angeschlossenen E/A-Modulen (insbesondere zum Erkennen von Verbindungs-Ausfällen) dient das Node/Life-Guarding. In der CANopen-Terminologie ist das Node-Guarding die Überwachung der E/A-Module durch den NMT-Master, das Life-Guarding läuft dagegen im E/A-Modul ab und überwacht den NMT-Master.

### Node-Guarding:

Der NMT-Master ruft mit einer speziellen RTR-Message (CAN-Request) zyklisch ein besonderes Telegramm von seinen NMT-Slaves (hier: =E/A-Modulen) ab. Stimmt die Antwort der Slaves nicht mit der erwarteten Antwort überein, erkennt der NMT-Master einen Fehler und reagiert entsprechend. Im Antworttelegramm ist der Modulstatus und ein Togglebit enthalten. Identifier und Zykluszeit werden dem NMT-Slave beim Extended-NMT-Bootup mitgeteilt.

### Life-Guarding:

Die NMT-Slaves überwachen, ob sie im Rahmen des Node-Guardings vom NMT-Master abgefragt werden. Bleiben diese Abfragen für eine gewisse Zeit aus, sendet der betroffene NMT-Slave eine Emergency Message. Beim EA515 wird das Ausbleiben der Node-Guard-Requests auch mit der roten Modul-Fehler-LED angezeigt (natürlich nur, wenn das Node-Guarding aktiviert wurde). Die Zeit bis zum Auslösen des Fehlers ist das Produkt aus der Node-Guarding-Zyklus-Zeit und dem sogenannten **Life-Time-Faktor**, der dem NMT-Slave ebenfalls beim Extended-NMT-Bootup mitgeteilt wird.

## **Wichtiger Hinweis:**

**Das Node/Lifeguarding wird bei der Umstellung von DS301 V3.0 auf DS301 V4.0 vermutlich *entfallen* !**

**Stattdessen wird in CANopen DS301 V4.0 von der CiA das neue „Heartbeat“-Protokoll vorgeschlagen, welches das Nodeguarding –mehr oder weniger- ablösen soll(?) .**

## 12.6 Das Emergency-Telegramm

Beim Auftreten interner Fehler sendet das E/A-Modul ein „Emergency Telegramm“. Dieses ist folgendermaßen aufgebaut:

Emergency Telegram

128 + ADR COB-ID	00 error code (low)	23 error code (high)	00 error regis- ter	00 manu. spec. field	00 manu. spec. field	00 manu. spec. field	00 manu. spec. field	00 manu. spec. field
---------------------	------------------------------	-------------------------------	------------------------------	-------------------------------	-------------------------------	-------------------------------	-------------------------------	-------------------------------

Das E/A-Modul „515“ unterstützt (mindestens) die folgenden (Emergency-) Error Codes:

Error Code (hex)	Bedeutung
00xx	Fehler verschwunden oder kein Fehler
10xx	allgemeiner Fehler
23xx	zu hoher Ausgangsstrom
50xx	Modul-Hardware - Fehler
60xx	Modul-Software - Fehler
70xx	Fehler in Zusatzmodulen
80xx	Überwachung („monitoring“ ?)
81xx	Kommunikationsfehler

Im Feld „error register“ des Emergency-Telegramms wird der aktuelle Inhalt von CANopen-Objekt 1001h eingeblendet.

Das „manufacturer specific error field“ wird derzeit (10/97) noch nicht genutzt.

Über die gesendeten Emergency-Telegramme wird im E/A-Modul eine Fehler-Historie gespeichert. Dazu dient Objekt 1003h im CANopen-Objekt-Verzeichnis.

## 13. Schnellstart

### 13.1 Konfiguration für den Einsatz am Lenze-Antriebsregler

Für einen schnellen Start mit der einfachsten Konfiguration sind die folgenden Schritte nötig:

#### 1. Definition der Ausgänge des E/A-Moduls per DIP-Schalter

- Stromversorgung des E/A-Moduls einschalten.
- Schalter 12 ON. Rote Modul-Fehler-LED blinkt langsam. ( Programmiermodus )
- Schalter 11 OFF ( zukünftige Erweiterungen passiv)
- Schalter 10 OFF ( E/A-Kaskardierung aktiv)
- Schalter 9 OFF ( „Lenze-Modus“ )
- Schalter 1..8 entsprechend E/A-Richtung: OFF=Eingang, ON=Ausgang
- Schalter 12 OFF. Die Blink-Impulse der roten Modul-Fehler-LED werden kürzer.
- Modulversorgung abschalten. Damit ist die Definition der Ein/Ausgänge abgeschlossen.

#### 2. Einstellen von CAN-Adresse, Master/Slave-Nummer und CAN-Baudrate am E/A-Modul (Lenze-Standard: 500 kBit/sec,

Antriebsregler auf CAN-Adresse 1, E/A-Master-Modul auf CAN-Adresse 2)

- Schalter 1 OFF, Schalter 2 ON, Schalter 3 bis 6 OFF (CAN-Adresse, vgl. Seite 30)
- Schalter 7 OFF, Schalter 8 OFF ( Modul wird E/A-Master )
- Schalter 9 OFF, Schalter 10 OFF, Schalter 11 ON ( 500 kBit/s, vgl. Seite 31 )
- CAN-Verbindung zum Antriebsregler herstellen
- Modulversorgung einschalten.  
Mode-Anzeige (grün) dauernd an, Betriebs-Anzeige (grün) blinkt.

#### 3. Einstellungen am Antriebsregler

(siehe „Konfiguration“, „Systembus“ im Lenze-Systemhandbuch)

- kontrollieren ob die Baudrate des Antriebsreglers mit der Baudrate aller E/A-Module übereinstimmt (C0351)
- sicherstellen, daß der Antriebsregler mit CAN-IN3 und CAN-OUT3 entsprechend der im Anhang ( Kapitel 13.3, Seite 58 ) definierten CAN-Identifizier sendet bzw. empfängt
- Sendezyklus für CAN-OUT3-Telegramm auf 100 ms setzen, damit die Überwachung im E/A-Modul keine Fehler meldet.
- Antriebsregler für CAN-BUS-Masterbetrieb konfigurieren (C0352)



## 13.2 Konfiguration für den Einsatz im CANopen-Netzwerk

Für einen schnellen Start mit der einfachsten Konfiguration sind die folgenden Schritte nötig:

### 1. Definition der Ausgänge des E/A-Moduls per DIP-Schalter

- Stromversorgung des E/A-Moduls einschalten.
- Schalter 12 ON. Rote Modul-Fehler-LED blinkt langsam. ( Programmiermodus )
- Schalter 11 OFF ( zukünftige Erweiterungen passiv)
- Schalter 10 ON ( E/A-Kaskardierung passiv)
- Schalter 9 ON ( „DS401-Modus“ )
- Schalter 1..8 entsprechend E/A-Richtung: OFF=Eingang, ON=Ausgang
- Schalter 12 OFF. Die Blink-Impulse der roten Modul-Fehler-LED werden kürzer.
- Modulversorgung abschalten. Damit ist die Definition der Ein/Ausgänge abgeschlossen.

### 2. Einstellen von CAN-Adresse (=„Node ID“) und CAN-Baudrate am E/A-Modul

(z.B.: 125 kBit/sec, CAN-Adresse 2)

- Schalter 1 OFF, Schalter 2 ON, Schalter 3 bis 6 OFF (CAN-Adresse, vgl. Seite 30)
- Schalter 7 OFF, Schalter 8 OFF
- Schalter 9 OFF, Schalter 10 ON, Schalter 11 ON ( 125 kBit/s, vgl. Seite 31 )
- CAN-Verbindung zur Steuerung herstellen
- Modulversorgung einschalten.  
Mode-Anzeige (grün) dauernd an, Betriebs-Anzeige (grün) blinkt bis „Start Node“ empfangen wurde.

---

# Anhang

## 13.2.1

### 13.3 Verwendete CAN-Identifizier

( „Adr“ = mit dem DIP-Schalter eingestellte Geräteadresse, Bereich 1..63 bzw. 1..127 )

„Lenze“-Modus, Werkseinstellung, nur bei Firmware „LENZE 9374“ verfügbar !			
Funktion	Identifizier	Identifizier max	Anzahl
Netzwerkmanagement	<b>0</b>	<b>0</b>	<b>1</b>
E/A - Eingänge (Eingang 1...32) vom E/A-Master-Modul zum Antriebsregler	<b>767 + Adr</b>	<b>831</b>	<b>63</b>
E/A - Ausgänge (Ausgang 1...32) vom Antriebsregler zu den E/A-Modulen	<b>768 + Adr</b>	<b>830</b>	<b>63</b>
interne Messages von Slave1 zum E/A-Master	<b>896 +Adr</b>	<b>959</b>	<b>63</b>
interne Messages von Slave 2 zum E/A-Master	<b>960 + Adr</b>	<b>1023</b>	<b>63</b>
interne Messages von Slave 3 zum E/A-Master	<b>1024 +Adr</b>	<b>1087</b>	<b>63</b>
interner Parameterkanal vom E/A-Master-Modul zu seinen Slaves 1...3	<b>1664 + Adr</b>	<b>1727</b>	<b>63</b>
SDO1 (tx) vom E/A-Master-Modul zum Leitsystem	<b>1408 + Adr</b>	<b>1471</b>	<b>63</b>
SDO1 (rx) vom Leitsystem zum E/A-Master-Modul	<b>1536 + Adr</b>	<b>1599</b>	<b>63</b>
SDO2 (tx) vom E/A-Modul zum Konfigurationstool (GDC)	<b>1472 + Adr</b>	<b>1535</b>	<b>63</b>
SDO2 (tx) vom Konfigurationstool (GDC) zum E/A-Modul	<b>1600 + Adr</b>	<b>1663</b>	<b>63</b>

## Anhang

„DS401“-Modus mit interner E/A-Master / Slave – Kaskardierung (predefined connection set nach DS301 V3.0 Kapitel 8.4.1) <b>IN ZUKUNFT UNMÖGLICH !!!</b>			
Funktion	Identifizier	Identifizier max	Anzahl
Netzwerkmanagement (einf., NMT-Master → NMT-Slave)	<b>0</b>	<b>0</b>	<b>1</b>
Netzwerkmanagement (erw., NMT-Slave → NMT-Master)	<b>2025</b>	<b>2025</b>	<b>1</b>
Emergency-Message	<b>128 + Adr</b>	<b>191</b>	<b>63</b>
PDO (transmit) (z.B. digitale Eingänge 1...32) vom E/A-Master-Modul zur Steuerung	<b>384 + Adr</b>	<b>447</b>	<b>63</b>
PDO (receive) (z.B. digitale Ausgänge 1...32) von der Steuerung zu den E/A-Modulen	<b>512 + Adr</b>	<b>575</b>	<b>63</b>
interne Messages von Slave1 zum E/A-Master	<b>896 +Adr</b>	<b>959</b>	<b>63</b>
interne Messages von Slave 2 zum E/A-Master	<b>960 + Adr</b>	<b>1023</b>	<b>63</b>
interne Messages von Slave 3 zum E/A-Master	<b>1024 +Adr</b>	<b>1087</b>	<b>63</b>
interner Parameterkanal vom E/A-Master-Modul zu seinen Slaves 1...3	<b>1664 + Adr</b>	<b>1727</b>	<b>63</b>
SDO (tx) vom E/A-Master-Modul zum Leitsystem	<b>1408 + Adr</b>	<b>1471</b>	<b>63</b>
SDO (rx) vom Leitsystem zum E/A-Master-Modul	<b>1536 + Adr</b>	<b>1599</b>	<b>63</b>
Nodeguarding (Identifizier-Vergabe nur durch NMT-Bootup)	<b>1760 + xxx</b>	<b>2015</b>	<b>255</b>

„DS401“-Modus ohne interne E/A-Master / Slave - Kaskardierung (predefined connection set nach DS301 V3.0 Kapitel 8.4.1)			
Funktion	Identifizier	Identifizier max	Anzahl
Netzwerkmanagement (einf., NMT-Master → NMT-Slave)	<b>0</b>	<b>0</b>	<b>1</b>
Netzwerkmanagement (erw., NMT-Slave → NMT-Master)	<b>2025</b>	<b>2025</b>	<b>1</b>
Emergency-Message	<b>128 + Adr</b>	<b>255</b>	<b>127(*)</b>
PDO (tx) (z.B. digitale Eingänge 1...8)	<b>384 + Adr</b>	<b>511</b>	<b>127</b>
PDO (rx) (z.B. digitale Ausgänge 1...8)	<b>512 + Adr</b>	<b>639</b>	<b>127</b>
SDO (tx) vom E/A-Master-Modul zum Leitsystem	<b>1408 + Adr</b>	<b>1535</b>	<b>127</b>
SDO (rx) vom Leitsystem zum E/A-Master-Modul	<b>1536 + Adr</b>	<b>1663</b>	<b>127</b>
Nodeguarding (Identifizier-Vergabe nur durch NMT-Bootup)	<b>1760 + xxx</b>	<b>2015</b>	<b>255</b>

(\*) 127 Node-IDs sind nur verfügbar, wenn auf die „interne Master/Slave-Kaskardierung“ (d.h. das Zusammenschalten von bis zu vier E/A - Modulen zu einem „logischen“ Modul) verzichtet wird.

# Anhang

## 14. Codetabelle

Hinweis: Eine Beschreibung des Zusammenhangs zwischen "Codestelle" (bzw Codenummer) und "CANopen-Objekt-Index" finden Sie in Kapitel 12.1.3 .

Code-nr.	Inhalt	Datentyp, [Einheit]	Zugriff r = read w= write	Einstell- / Anzeige-möglichkeiten	Werks-einstellung	Bemerkungen, Verweise
Codestellen für die elementare Modul-Konfiguration						
5	Bootup-Zeit	16 Bit, [ms]	r,w	0.. 65000 ms, 0xFFFF = keine Boot-Message	0	Zeitintervall zwischen Power-ON und Senden der Boot-Message
10	Portrichtungs-Definitionen	32 Bit	r,w (*)	Bitvektor: 0 = Eingang, 1 = Ausgang	00000000b	LSB=Master MSB=Slave3
11	Entprellzeit für digitale Eingänge	8 Bit, [ms]	r,w	0=aus ... 255 ms	1 ms	
12	Ereignis-Maske für digitale Eingänge	32 Bit	r,w	Bitvektor: 0 = kein IRQ 1 = Interrupt	1111..1111bin.	ähnlich CANopen-Objekt 6006h
13	Filterkonstanten der analogen Eingänge	16 Bit	r,w	siehe Handbuch-Zusatz für analoge Eingänge	0 = keine Filterung	
14	Zusätzlicher Sende-Zyklus für TPDO („Extra PDO TX Cycle“)	16 Bit, [ms]	r,w	0 = nur durch Ereignis ... 65000 ms	100 ms	unabhängig von der Sendepause
15	Mindest-Sende-Pause bei ereignisgesteuertem Senden	8 Bit, [ms]	r,w	0 = keine Pause ... 255 ms	0	für Buslast-Begrenzung, speziell für LENZE geplant
16	Internen Master-Slave-Verbund ermöglichen / verhindern	8 Bit	r,w	0 = keine Kaskardierung 1 = Kaskardierung möglich	1	siehe Kapitel 10.2.3
17	„autonomes“ Konfigurations-Speichern aktivieren	8 Bit	r,w	0 = Speichern nur auf Befehl 1 = Speichern nach jedem Schreiben	1	vgl. CANopen-Objekt 1010h
18	Hardware-Modell	8 Bit	r ..			n.f. Hersteller
19	Modell-Optionen	8 Bit	r ..			n.f. Hersteller
Codestellen zur Definition von Überwachungs- und Anzeige-Zeiten des Moduls, die auf den Normalbetrieb keinen Einfluß haben.						
30	PDO-RX-Monitor (Überwachungszeit, wenn kein Life-Guarding)	16 Bit, [ms]	r,w	0=keine Überwachung ... 25500 ms	0	Auflösung 100 ms, speziell für LENZE geplant

---

## Anhang

(\*) das Beschreiben dieser Codestelle wirkt sich erst nach Neustart aus.

## Anhang

### Codetabelle (Fortsetzung)

Code-nr.	Inhalt	Datentyp, [Einheit]	Zugriff r = read w = write	Einstell- / Anzeigemöglichkeiten	Werkeinstellung	Bemerkungen, Verweise
Codestellen zur Definition von CAN-Identifizier-Basisadressen, zu denen noch die am DIP-Schalter eingestellte Geräteadresse (Bereich 1..63) addiert werden muß.						
50	PDO-RX	16 Bit	r,w (*)	siehe CANopen	je nach Modus (LENZE, DS401)	(dig. Outputs)
51	PDO-TX	16 Bit	r,w (*)	siehe CANopen	je nach Modus	(dig. Inputs)
52	Slave1	16 Bit	r	-	896	
53	Slave2	16 Bit	r	-	960	
54	Slave3	16 Bit	r	-	1024	
55	Master->Slaves	16 Bit	r	-	1664	

## Anhang

### Codetabelle (Fortsetzung)

Code-nr.	Inhalt	Datentyp, [Einheit]	Zugriff r = read w= write	Einstell- / Anzeigemöglichkeiten	Werks-einstellung	Bemerkungen, Verweise
Codestellen für Anzeige- und Diagnosezwecke						
93	Drive Ident	16 Bit-Zahl , bei 9374: „FIX32“ !	r	z.B. 515x bei 9374: 93740000		
98	Compilation Date	Char-Array	r	z.B. 07/26/00 für „26. Juli 2000“	aktuelles Datum, an dem die Firmware kompiliert wurde.	für Service-Zwecke. Ein Subindex pro Buchstabe !
99	S/W Version	String[4], bei 9374: „FIX32“ !	r	z.B. „0.01“ bei 9374: 10000 = „V1.0“		
101	interne Fehler- und Zustandscodes	32 Bit	r	(1) EAM_error_flags (2) EAM_slave_errors (3) IOP_status (4) CAL_comm_pha.	-	interne Fehler und Zustände, nur für interne Verwendung
102	Angeschlossene Slaves	16 Bit	r	Bitvektor: 0 = kein Slave erkannt +1 = Slave 1 ok, +2 = Slave 2 ok, +4 = Slave 3 ok	-	Der Test wird erst beim Lesen dieser Code-stelle durchgeführt
103	Hauptschleifengeschwindigkeit	16 Bit	r	liefert Schleifen/sec z.B. 3000	-	abhängig von CPU-Last
104	Klemmenzustand	32 Bit	r	Bitkombination, Bit 0 = erster Eingang, nur für LENZE (9374)	-	ähnelt Objekt 0x6120.1 aus CANopen DS401
120	A/D-Wandler-Kanal setzen	8 Bit	w	0 ... 7 (CPU-interne Multiplexer-Nummer)		
121	A/D-Wandler lesen	10 Bit	r	0 = 0 V .. 307 = 1.5 V .. 511 = 2.5 V .. 1023 = 5.0 V	-	NUR zum Test der internen „digitalen“ Eingangspegel
123	CAN-Register-Zugriff	8 Bit	r	liest den Inhalt eines der 256 CAN-Controller-Register		NUR für Diagnose-Zwecke
202	Lenze-EKZ	4*4 Zeichen	r	EKZ auf 4 Subcodes aufgeteilt, z.B. „33S9374C 23000“	-	Nur bei Firmware „9374“ !

# Anhang

Code-nr.	Inhalt	Datentyp, [Einheit]	Zugriff r = read w= write	Einstell- / Anzeige-möglichkeiten	Werks-einstellung	Bemerkungen, Verweise
Codestellen-Bereiche im EA515, wenn bestimmte Erweiterungen (Hard- oder Software) installiert sind. Genauere Informationen sind den angegebenen Dokumentationen (i.A. "Systemhandbuch-Zusätze") zu entnehmen. Nicht alle in dieser Übersicht aufgeführte Codestellen werden in "Ihrem" Modul vorhanden sein !						
256 .... 281	Codestellen für (Software-) Zählermodul	-		siehe Systemhandbuch-Zusatz "E/A 515x Zählermodul"		
512 .... 767	Codestellen für Erweiterung "4 analoge Eingänge"	-		siehe Systemhandbuch-Zusatz "E/A-Modul 515 - 4 Ain"		
640 .... 655	Codestellen für Erweiterung „4 analoge Ausgänge“	-		siehe Systemhandbuch-Zusatz „E/A-Modul 515 - 4 Aout“		
768 .... 1023	Codestellen für Erweiterung "PID-Regler-Modul "	-		siehe Systemhandbuch-Zusatz "EA515 mit PID-Regler"		
1024 .... 1039	Codestellen für Erweiterung "SPS-BCD-Gateway "	-		siehe Kurzbeschreibung "EA515 mit Sonder-Software BCD - Gateway"		
Sonder-Codestellen für diverse „Sonderzwecke“						
10196 = Obj. 382B (hex)	Zustand „Operational“ ohne NMT	-	r,w	0 = „Operational“ nur nach Empfang von „Start Node“ 0x0089 = „Operational“ nach Ablauf der Boot-Zeit	0 (Normalfall)	existiert nur bei Sondersoftware „EA5_NNMT“