

SDO – Zugriffe ohne Ballast SDO – Zugriffe ohne Ballast (wirklich „ohne Ballast“ ?)

Autor: Wolfgang Büscher
Stand: 18.05.2001
Ablage: <WoBu>c:\proj\ea_mod\doku\art85029_Ea515_Easy_SDO.odt
Verteiler: _____ (unbekannt)

**ACHTUNG: DIESE BESCHREIBUNG IST NICHT ALS GRUNDLAGE FÜR EINEN
„VERNÜNFTIGEN“ CANOPEN-EINSTIEG ZU SEHEN !
WESENTLICHE BESTANDTEILE VON CANOPEN WERDEN HIER NICHT ERWÄHNT !**

Dieses Dokument ist lediglich als Information von Programmierern für Programmierer vorgesehen, die einen schnellen Einstieg für den Einsatz der E/A-Module vom Typ "EA-515" in CANopen-Netzwerken benötigen.

Sorry, there is no english translation of this file. If you need one, contact Mr. Buescher.

Format der CAN-Telegramme

Die in den Beispielen in diesem Dokument verwendeten CAN-Telegramme sind im folgenden Format notiert:

Identifizier (11 Bit)	Byte[0]	Byte[1]	Byte[2]	Byte[3]	Byte[4]	Byte[5]	Byte[6]	Byte[7]
-----------------------	---------	---------	---------	---------	---------	---------	---------	---------

Der CAN-Bus-Identifizier wird hier i.A. als 4-ziffrige Dezimalzahl notiert.

Die Datenbytes werden i.A. hexadezimal notiert, was das „Zusammensetzen“ zu 16- oder 32-Bit-Werten vereinfacht. Beachten Sie, dass Array-Indizes bei vernünftigen Programmiersprachen immer bei Null beginnen.

Der Prefix „0x“ oder „\$“ kennzeichnet hexadezimale Werte. Dies entspricht dem Format des CAN-Testers für DOS.

Einfache SDO-Zugriffe für CANopen-E/A-Module

Initialisierung des Netzwerks

(Gehört eigentlich nicht in eine SDO-Beschreibung, ist aber für einen schnellen Einstieg unerlässlich).

Nach dem Einschalten melden sich manche CANopen-Module mit einem Telegramm <128+NodeId> ohne Nutzdatenbytes. Dies gilt als „Bootup-Message“ und kann von Ihnen zu Testzwecken abgefragt werden.

Danach sollte Ihre Anwendung (bzw. der CANopen-Master des Netzes) einmal den Befehl „Start Remote Node“ als NMT-Broadcast senden. Das Telegramm verwendet nur 2 der 8 möglichen Nutzdatenbytes:

Identifizier (NMT)	Befehls code	Node						
0000	0x01	0x00						

Hinweis 1:

Bei entsprechender Konfiguration beginnen manche Module nun mit dem Senden von PDOs (Process Data Objects). Da die PDOs aus guten Gründen nicht Teil dieser Doku sind, wird darauf nicht weiter eingegangen. Um PDOs effektiv nutzen zu können, benötigen Sie SDOs zum Zugriff auf die entsprechenden Konfigurationsobjekte.

Hinweis 2:

Um mit einem E/A-Modul **nur** per SDO zu kommunizieren, brauchen Sie kein NMT-Startup. SDO-Zugriffe funktionieren sowohl im NMT-Zustand „pre-operational“ als auch „operational“. ABER: Einige alte EA515-Module von MKT unterbinden die Ansteuerung von digitalen Ausgängen, solange sich das Modul im Zustand „pre-operational“ befindet. Dies bezog sich leider auch auf die Ansteuerung per SDO (statt nur per PDO).

Einfache SDO-Zugriffe für CANopen-E/A-Module

SDO-Zugriffe

SDO-Zugriffe sind nach CANopen-Terminologie „confirmed services“, *nur im einfachsten Fall besteht ein Zugriff aus einem Anforderungs- und einem Antworttelegramm*¹.

Aus Sicht des SDO-CLIENTS: Zwischen Senden eines Anforderungstelegramms und Empfang des Antworttelegramms kann eine in DS301 nicht spezifizierte Zeit vergehen. Rechnen Sie mit bis zu 500 Millisekunden, auch wenn die meisten Zugriffe erheblich schneller erfolgen werden.

Die folgenden Beispiele sollen den prinzipiellen Ablauf von *einfachen* SDO-Zugriffen verdeutlichen. Für viele Anwendungen reicht dies bereits aus:

Beispiel 1: Ansteuerung von digitalen Ausgängen per SDO (8-Bit-Schreibzugriff)

(Hinweis: die Portrichtungen der 8 E/A-Leitungen müssen per DIP-Schalter o.ä. bereits passend konfiguriert sein, und das Modul im Zustand OPERATIONAL sein).

Das angesprochene E/A-Modul hat hier die „Knotennummer“ (node ID) 2.

Der Objektindex für die digitalen Ausgänge muss dem Device-Profil oder dem Systemhandbuch entnommen werden, beim EA515-8I/O ist dies Objekt 0x6200.

Subindex 1 adressiert die ersten 8 digitalen Outputs.

Identifizier	Befehls - code	Index (low)	Index (high)	Sub-index	Data 1	Data 2	Data 3	Data 4
1536 + 2 = 1538	(write 8) 0x2F	L(0x6200) = 0x00	H(0x6200) = 0x62	0x01	(8 outputs) 0xFF	(unused) 0x00	(unused) 0x00	(unused) 0x00

Das EA-Modul sendet daraufhin ein Antworttelegramm, z.B:

Identifizier	Befehls - code	Index (low)	Index (high)	Sub-index	Data 1	Data 2	Data 3	Data 4
1408 + 2 = 1410	(w.resp) 0x60	L(0x6200) = 0x00	H(0x6200) = 0x62	0x01	0x00	0x00	0x00	0x00

Die 3 höherwertigen Bits im Befehlscode (command specifier) zeigen ob der Schreibzugriff erfolgreich war. Verknüpfen Sie den empfangenen Befehlscode mit 0xE0. Das Ergebnis 0x60 bedeutet „Schreibzugriff erfolgreich“ (die letzten 4 Datenbytes im Antworttelegramm sind zu ignorieren!).

Bei Fehlern würde die Verknüpfung 0x80 ergeben (die letzten 4 Datenbytes im Antworttelegramm enthalten dann einen Fehlercode nach CANopen).

Wenn Sie nach XXX Millisekunden noch keine Antwort erhalten haben, sollten Sie den CAN-Bus und die verwendeten CAN-Identifizier überprüfen. Beachten Sie, dass **ein** SDO-Server (hier: E/A-Modul) nicht mehrere SDO-Aufträge gleichzeitig bearbeiten kann. Darum müssen Sie vor dem Senden des nächsten Auftrags erst die Antwort vom SDO-Server abwarten.

¹ Der hier beschriebene „einfache“ SDO-Zugriff heißt bei CANopen „expedited transfer“

Einfache SDO-Zugriffe für CANopen-E/A-Module

Beispiel 2: Abfrage von digitalen Eingängen per SDO (8-Bit-Lesezugriff)

Der Objektindex für die digitalen Eingänge muss dem Device-Profil oder dem Systemhandbuch entnommen werden, beim EA515-8I/O ist dies Objekt 0x6000.

Subindex 1 adressiert die ersten 8 digitalen Inputs.

Das angesprochene E/A-Modul hat hier die „Knotennummer“ (node ID) 2.

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

1536 + 2 = 1538	(read) 0x40	L(0x6000) = 0x00	H(0x6000) = 0x60	0x01	(unused) 0x00	(unused) 0x00	(unused) 0x00	(unused) 0x00
-----------------	-------------	------------------	------------------	------	---------------	---------------	---------------	---------------

Das EA-Modul sendet daraufhin ein Antworttelegramm, in Data1 stehen die Zustände der ersten 8 digitalen Inputs:

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

1408 + 2 = 1410	(w.resp) 0x4F	L(0x6000) = 0x00	H(0x6000) = 0x60	0x01	0x01	0x00	0x00	0x00
-----------------	---------------	------------------	------------------	------	------	------	------	------

Die 3 höherwertigen Bits im Befehlscode (command specifier) zeigen ob der Lesezugriff erfolgreich war.

Verknüpfen Sie den empfangenen Befehlscode mit 0xE0. Das Verknüpfungsergebnis 0x40 bedeutet „Lesezugriff erfolgreich“ (die letzten 4 Datenbytes im Antworttelegramm enthalten dann –beim Expedited Transfer- den gelesenen Wert).

Bei Fehlern würde die Verknüpfung 0x80 ergeben (die letzten 4 Datenbytes im Antworttelegramm enthalten dann einen Fehlercode nach CANopen).

Im oben angeführten Beispiel steht in Data1 der Wert 0x01, d.h. Bit 0 ist gesetzt. Bei Objekt 0x6000.1 bedeutet dies, dass der erste digitale Eingang „high“ ist (und die Eingänge 2..8 „low“).

Nach dem gleichen Schema können auch andere Objekte gelesen werden, z.B. analoge Eingänge etc.

Bei 16-Bit-Daten steht das LSB (Bits 7..0) in „Data1“, das MSB (Bits 15..8) in „Data2“.

Bei 32-Bit-Daten steht das LSB (Bits 7..0) in „Data1“, das MSB (Bits 31..24) in „Data4“.

Die benötigten SDO-Befehlscodes finden Sie im Anhang.

Einfache SDO-Zugriffe für CANopen-E/A-Module

Beispiel 3: Abfrage von analogen Eingängen per SDO (16-Bit-Lesezugriff)

Der Objektindex für die analogen Eingänge muss dem Device-Profil DS401 oder dem Systemhandbuch entnommen werden, beim EA515-8I/O ist dies Objekt 0x6401.

Subindex 1 adressiert den ersten Analogeingang, Subindex 2 den zweiten Analogeingang usw.

Das im folgenden Beispiel angesprochene E/A-Modul hat die „Knotennummer“ (node ID) 2.

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

1536 + 2 = 1538	(read) 0x40	L(0x6401) = 0x01	H(0x6401) = 0x40	0x01	(unused) 0x00	(unused) 0x00	(unused) 0x00	(unused) 0x00
--------------------	----------------	---------------------	---------------------	------	------------------	------------------	------------------	------------------

Das EA-Modul sendet daraufhin ein Antworttelegramm, in Data1..Data2 steht der aktuelle Analogwert im CANopen-Format („linksbündiger 16-Bit-Wert mit Vorzeichen, d.h. -32767 .. +32767):

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

1408 + 2 = 1410	(w.resp) 0x4B	L(0x6401) = 0x01	H(0x6401) = 0x64	0x01	0x34	0x12	0x00	0x00
--------------------	------------------	---------------------	---------------------	------	------	------	------	------

Die 3 höherwertigen Bits im Befehlscode (command specifier) zeigen ob der Lesezugriff erfolgreich war.

Verknüpfen Sie den empfangenen Befehlscode mit 0xE0. Das Verknüpfungsergebnis 0x40 bedeutet „Lesezugriff erfolgreich“ (die letzten 4 Datenbytes im Antworttelegramm enthalten dann –beim Expedited Transfer- den gelesenen Wert).

Bei Fehlern würde die Verknüpfung 0x80 ergeben (die letzten 4 Datenbytes im Antworttelegramm enthalten dann einen Fehlercode nach CANopen).

Im oben angeführten Beispiel steht in Data1 „0x34“ und in Data2 „0x12“. Zu einem 16-Bit-Wert zusammengesetzt ergibt dies 0x1234 (low byte first !). Wie dieses Ergebnis zu interpretieren ist, hängt vom der Quelle (hier: Analogeingang) ab. Bei MKT's EA515 könnte der Wert 0x7FFF beispielsweise für „+20mA“ oder „+10V“ stehen. Näheres dazu finden Sie entweder im Systemhandbuch oder in CANopen DS401.

Eine Übersicht über einige SDO-Befehlscodes finden Sie im Anhang.

Einfache SDO-Zugriffe für CANopen-E/A-Module

Übersicht einiger SDO-Befehlscodes :

Der Befehlscode setzt sich **unter anderem** aus dem Command Specifier und der Länge zusammen. Häufig benötigte Kombinationen sind z.B. (in HEXADEZIMALER SCHREIBWEISE IM „C“-STIL):

Code	Bedeutung
0x40	Read Request, Aufforderung zum Lesen eines Parameters
0x43	Read Response mit 32 Bit Daten im Telegramm
0x4B	Read Response mit 16 Bit Daten im Telegramm
0x4F	Read Response mit 8 Bit Daten im Telegramm
0x23	Write Request mit 32 Bit Daten im Telegramm
0x2F	Write Request mit 8 Bit Daten im Telegramm
0x60	Write Response, d.h. ein Parameter wurde erfolgreich gesetzt
0x80	Error Response, mit 32-Bit-Fehlercode in Data1..Data4

Bit 0 im Befehlscode (size indicator) eines empfangenen Telegramms sollten Sie ignorieren, weil nicht alle Geräte dieses Flag unterstützen. Wenn dieses Bit gesetzt ist, definieren Bits 3..2 die Anzahl Datenbytes (codiert: 4-n).

Hinweis: Dies sind nur einige mögliche Kombinationen (für den „expedited transfer“). Weitere Informationen zum SDO-Protokoll finden Sie nur in der CANopen-Spezifikation (DS301, www.can-cia.de).