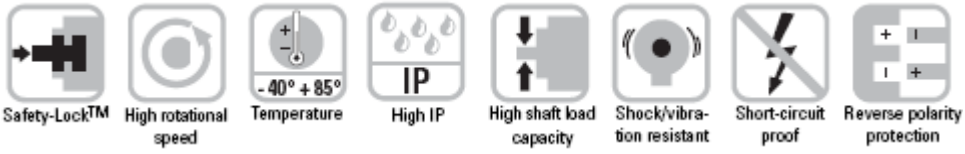
Notice technique

Kübler

Série M3658, M3678



CANopen certified

CiA200704-301V402/20-0071

l'impulsion pour l'automatisme

© Fritz Kübler GmbH

Droits d'auteur

Les droits d'auteur de la présente documentation sont protégés par la société Fritz Kübler GmbH. La présente documentation ne peut être ni modifiée, ni étendue, ni dupliquée, ni transmise à des tiers sans l'autorisation écrite de la société Fritz Kübler GmbH.

Réserve de modifications

Dans le cadre de nos efforts d'amélioration permanente de nos produits, nous nous réservons le droit d'apporter à tout moment des modifications techniques aux informations techniques contenues dans le présent document.

Aucune garantie

Fritz Kübler GmbH ne donne aucune garantie, implicite ou explicite, en rapport avec l'ensemble de la présente notice, et décline toute responsabilité en cas de dommages directs ou indirects.

Informations sur le document
Indice de modification 8-2010

Copies d'écran utilisées
CANalyzer® Sté. Vector –Informatik

Fritz Kübler GmbH Schubertstr.47
78054 VS-Schwenningen / Germany
Tél. +49 (0) 7720-3903-0
Fax +49 (0) 7720-21564
E-Mail: info@kuebler.com
Internet: www.kuebler.com

Sommaire

1 GENERALITES - CARACTERISTIQUES 1-5

CODEURS CANOPEN MONOTOUR SERIE M36X8 M=PRINCIPE DE LECTURE MAGNETIQUE.....1-5
LE PROFIL DE COMMUNICATION CANOPEN DS 301 V4.02 1-5
LE PROFIL D'APPAREIL CODEURS DS 406 V3.1 1-6
SERVICES LSS DS 305 V2.0.....1-6
TRANSMISSION DE DONNEES 1-6
OBJETS ET CODES DE FONCTIONS DANS LE PREDEFINED CONNECTION SET 1-7
OBJETS BROADCAST (DIFFUSION SUR L'ENSEMBLE DU RESEAU) 1-7
OBJETS PEER-TO PEER (APPAREIL A APPAREIL) 1-7
OBJETS D'UTILISATION RESTREINTE ET RESERVES 1-7
TRANSMISSION DES DONNEES PROCESS 1-8
PDO D'EMISSION 1 (1800H) POSITION ASYNCHRONE 1-8
PDO D'EMISSION 2 (1801H) VITESSE ASYNCHRONE 1-8
PDO D'EMISSION 3 (1802H) POSITION SYNCHRONE.....1-8
TRANSMISSION DES DONNEES SERVICE.....1-9
EXEMPLE D'UNE TRANSMISSION DE DONNEES SERVICE VERS ET DEPUIS LE CODEUR 1-10
SERVICES LSS.....1-10

2 TENSION D'ALIMENTATION ET RACCORDEMENT AU BUS CAN 2-11

AFFECTATION DU CABLE CAN 2-11
RACCORDEMENT CANOPEN CONNECTEUR M12 2-11
CARACTERISTIQUES MECANIQUES 2-11

3 PREMIERE MISE EN SERVICE - REGLAGES GENERAUX DE L'APPAREIL 3-12

VITESSE DE TRANSMISSION 3-12
NUMERO DE NŒUD.....3-12
TERMINAISON 3-12
SAVE ALL BUS PARAMETERS 3-12

4 LAYER SETTING SERVICES (LSS).....4-13

5 REGLAGES PAR DEFAUT A LA LIVRAISON.....5-15

PARAMETRES DE COMMUNICATION 5-15
PROFIL CODEUR.....5-16

6 PARAMETRES DE COMMUNICATION 6-17

DEFINITION DU TYPE DE TRANSMISSION (TRANSMISSION TYPE) DU PDO.....6-19
MAPPAGE PDO VARIABLE 6-19

7 EXEMPLE D'UN ENREGISTREMENT DE MAPPAGE VARIABLE7-20

8 STRUCTURE D'UN ENREGISTREMENT DE MAPPAGE 8-21

9 EXEMPLE DE PROGRAMMATION D'UNE APPLICATION:.....9-22

REGLAGE DES OBJETS.....9-22
Limiter Measuring Units per Revolution à 3600.....9-22
Preset value à 0 9-23
Définition des valeurs des paramètres d'émission de TPDO1 et TPDO2 9-23
Régler Producer Heartbeat à 500 ms 9-24
Définir les limites basse et haute de Work area.....9-24
Sauvegarde de tous les paramètres modifiés dans la mémoire Flash Store Parameters 1010h 9-25
Objet 1010h Sauvegarde des paramètres.....9-25
Objet 1011h : Chargement des valeurs standards 9-25
PROFIL DE COMMUNICATION – AUTRES OBJETS.....9-26
OBJET 1018H : IDENTITY OBJECT 9-26

10 OBJETS EMERGENCY.....10-27

CODES D'ERREUR SUPPORTES 10-27

11 MESSAGE EMERGENCY 11-28

EXEMPLE D'UN MESSAGE EN CAS DE DEFAUT HARDWARE :.....11-28

PROTOCOLE EMERGENCY 11-28

12 PROTOCOLE HEARTBEAT 12-28

13 OBJET CANOPEN REPERTOIRE 13-29

STRUCTURE DU REPERTOIRE D'OBJETS COMPLET: 13-29

14 PROFIL DE COMMUNICATION CANOPEN DS 301 14-30

OBJETS DE COMMUNICATION 14-30
OBJETS SPECIFIQUES AU CONSTRUCTEUR 14-30

15 PROFIL D'APPAREIL CANOPEN POUR CODEURS DS 406 V3.115-31

OBJETS SPECIFIQUES A L'APPAREIL 15-31

16 OBJETS DANS DETAILS - PROFIL CODEUR DS 306 V3.116-32

OBJET 6000H OPERATING PARAMETERS 16-32
OBJET 6001H : NOMBRE DE PAS DE MESURE PAR TOUR (RESOLUTION) 16-32
APRES UNE MODIFICATION DU FACTEUR D'ECHELLE, IL FAUT IMPERATIVEMENT REGLER AUSSI LA VALEUR DE PREPOSITIONNEMENT A ZERO OU A LA VALEUR DESIREE. 16-32
OBJET 6002H : NOMBRE TOTAL DES PAS DE MESURE 16-33
APRES UNE MODIFICATION DU NOMBRE TOTAL, IL FAUT IMPERATIVEMENT REGLER AUSSI 16-33
OBJET 6003H : VALEUR DE PREPOSITIONNEMENT 16-33
OBJET 6004H : VALEUR DE POSITION 16-33
CET OBJET PEUT ETRE MAPPE 16-33
OBJET 6030H : SPEED VALUE 16-34
CET OBJET PEUT ETRE MAPPE 16-34
OBJET 6200H : CYCLE-TIMER 16-34
OBJET 6500H : AFFICHAGE DE L'OPERATING STATUS 16-34
OBJET 6503H : ALARMS 16-35
OBJET 6504H : SUPPORTED ALARMS 16-35
OBJET 6505H : WARNINGS 16-36
OBJET 6506H : SUPPORTED WARNINGS 16-36
OBJET 6400H : WORKING AREA STATE REGISTER 16-37
OBJET 6401H : WORKING AREA LOW LIMIT 16-37
OBJET 6402H : WORKING AREA HIGH LIMIT 16-37
OBJET 2100H : 16-37
OBJET 2101H : ADRESSE DE NŒUD 16-38
OBJET 2102H : TERMINAISON DE BUS CAN DESACTIVEE/ACTIVEE 16-38
OBJET 2103H : VERSION DE FIRMWARE FLASH 16-38
OBJET 2105H : SAVE ALL BUS PARAMETERS 16-39
OBJET 2140H : CUSTOMER MEMORY (16 BYTES) 16-39
OBJET 1029H : ERROR BEHAVIOR 16-39

17 GESTION DU RESEAU 17-40

18 INSTRUCTIONS NMT 18-41

19 SIGNALISATION PAR LA LED PENDANT LE FONCTIONNEMENT 19-42

LED VERTE = ETAT DU BUS 19-42
LED ROUGE = INDICATION ERR 19-42

20 DEFINITIONS 20-43

21 ABREVIATIONS UTILISEES 21-43

22 TABLEAU DE CONVERSION DECIMAL-HEXADECIMAL 22-44

23 GLOSSAIRE 23-45

1 Généralités - Caractéristiques

Codeurs CANopen monotour Série M36X8 M=principe de lecture Magnétique

Les codeurs CANopen de la série M6358/78 supportent le profil de communication CANopen le plus récent selon **DS 301 V4.02**. Ils disposent en outre de profils spécifiques aux appareils comme les profils codeur **DS 406 V3.1**.

Les modes opératoires disponibles sont Polled Mode, Cyclic Mode et Sync Mode. Le bus CAN permet en outre la programmation de facteurs d'échelle, de valeurs de présélection, de valeurs de fin de course et de nombreux autres paramètres. A la mise sous tension, tous les paramètres, mémorisés au préalable pour les protéger contre toute coupure de courant, sont chargés depuis une mémoire Flash. Les valeurs de sortie suivantes : **position**, **vitesse**, ainsi que l'état des deux **fins de course**, peuvent se combiner de manière très variable sous la forme de **PDO** (Mappage PDO).

Ces codeurs sont disponibles avec un **connecteur M12** ou une **sortie par câble**, l'adresse de l'appareil et la vitesse de transmission se modifient à l'aide d'un logiciel.

Une LED bicolore sur la face arrière indique l'état de fonctionnement et les défauts du bus CAN, ainsi que l'état d'un diagnostic interne.
Les codeurs CAN sont disponibles dans les versions à arbre creux borgne et à arbre sortant ; grâce à leur indice de protection IP65, ils conviennent aussi pour des environnements industriels rudes.

Le profil de communication CANopen DS 301 V4.02

CANopen constitue une interface utilisateur homogène et simplifie ainsi la réalisation de systèmes réunissant les appareils les plus variés. CANopen est optimisé pour l'échange rapide de données dans des systèmes en temps réel et dispose de différents profils d'appareil qui ont été standardisés. L'association de fabricants et d'utilisateurs CAN in Automation (CiA) est responsable de l'élaboration et de la normalisation des profils correspondants.

CANopen offre :

- un accès confortable à tous les paramètres des appareils.
- une auto-configuration du réseau et des appareils
- la synchronisation des appareils dans le réseau
- un trafic des données process cyclique et lié à des événements
- une lecture ou une émission simultanée de données

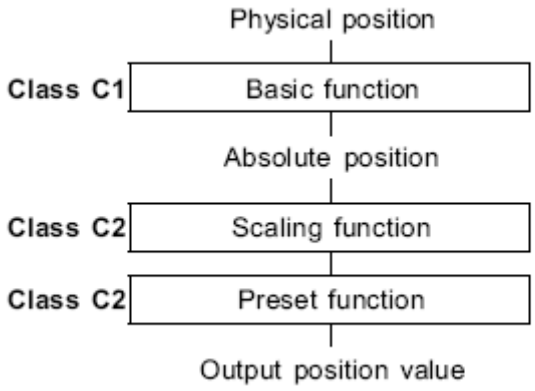
CANopen fait appel à quatre objets de communication (COB) ayant des caractéristiques différentes

- Objets de données process (PDO) pour des données en temps réel,
- Objets de données service (SDO) pour la transmission de paramètres et de programmes,
- Gestion du réseau (NMT, Life-Guarding, Heartbeat)
- Objets prédéfinis (pour la synchronisation, l'horodatage, les cas d'urgence)

Tous les paramètres de l'appareil sont sauvegardés dans un **répertoire d'objets** . Ce répertoire d'objets contient la description, le type de donnée et la structure des paramètres, ainsi que l'adresse (index).
Le répertoire se décompose en une partie profil de communication, une partie profil d'appareil et une partie spécifique au constructeur.

Le profil d'appareil Codeurs DS 406 V3.1

Ce profil fournit une définition de l'interface pour les codeurs **indépendante des constructeurs** et obligatoire.
Ce profil définit les fonctions CANopen à utiliser, ainsi que la manière de les utiliser. Cette norme permet la réalisation d'un système de bus ouvert et indépendant des constructeurs.
Le profil d'appareil se décompose en deux classes d'objets :



La **classe C1** décrit l'ensemble des fonctions de base dont le codeur doit être muni

La **classe C2** comporte de nombreuses fonctions étendues qui doivent être supportées par les codeurs de cette classe (Mandatory-obligatoires) ou qui sont en option. Les appareils de la classe C2 disposent ainsi de toutes les fonctions C1 et C2 obligatoires et d'autres fonctions optionnelles dépendant du constructeur. Le profil définit en outre une plage d'adresses pouvant être occupées par des fonctions spéciales spécifiques au constructeur.

Services LSS DS 305 V2.0

Le Layer Setting Service et le protocole (LSS) CiA DSP 305 CANopen ont été développés pour permettre la lecture et la modification des paramètres suivants par l'intermédiaire du réseau:

- Adresse de nœud
- Vitesse de transmission
- Adresse LSS

Ces possibilités augmentent la compatibilité « Plug-and-Play » de l'appareil, simplifiant sensiblement ses possibilités de configuration. Le maître LSS est en charge de la configuration de ces paramètres pour un ou plusieurs esclaves du réseau.

Transmission de données

CANopen fait appel, pour la transmission des données, à deux types de communication différents (COB=Communication Object) avec des caractéristiques différentes:

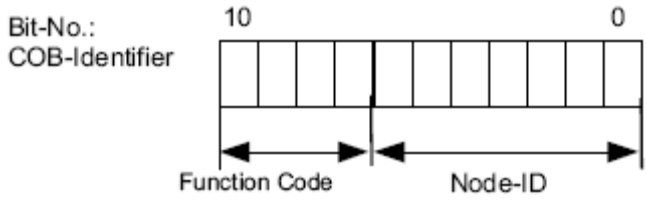
- **Objets de données process (PDO – avec possibilité de fonctionnement en temps réel)**
- **Objets de données service (SDO)**

Les objets de données process **(PDO)** sont utilisés pour l'échange hautement dynamique de données en temps réel (p. ex. position du codeur, vitesse, état des positions de référence) d'une longueur maximale de 8 octets. Ces données sont transmises avec une priorité haute (identifiant COB bas). Les PDO sont des données de diffusion en temps réel disponibles en même temps à tous les destinataires désirés. Les PDO peuvent être mappées : un mot de données de 8 octets peut comporter 4 octets pour la position et 2 octets pour la vitesse.

Les objets de données service **(SDO)** constituent le canal de communication pour la transmission de paramètres des appareils (p. ex. programmation de la résolution du codeur). Comme ces paramètres sont transmis de manière acyclique (p. ex. une seule fois lors du démarrage du réseau), les objets SDO ont une priorité basse (identifiant COB élevé).

Objets et codes de fonctions dans le Predefined Connection Set

Afin de faciliter la gestion des identifiants, CANopen utilise le « Predefined Master/Slave Connection Set » (télégramme maître/esclave prédéfini). Tous les identifiants sont définis ici dans le répertoire d'objets par des valeurs standard. Ces identifiants peuvent cependant être modifiés de manière spécifique au client par accès SDO.



L'identifiant à 11 bits se compose d'un **code de fonction à 4 bits** et d'un **numéro de nœud à 7 bits**.

Plus la valeur de l'identifiant COB est élevée, plus sa priorité est basse !

Objets Broadcast (diffusion sur l'ensemble du réseau)

object	function code (binary)	resulting COB-ID	Communication Parameters at Index
NMT	0000	0	-
SYNC	0001	128 (80h)	1005h, 1006h, 1007h
TIME STAMP	0010	256 (100h)	1012h, 1013h

Objets Peer-To Peer (appareil à appareil)

object	function code (binary)	Resulting COB-IDs	Communication Parameters at Index
EMERGENCY	0001	129 (81h) – 255 (FFh)	1014h, 1015h
PDO1 (tx)	0011	385 (181h) – 511 (1FFh)	1800h
PDO1 (rx)	0100	513 (201h) – 639 (27Fh)	1400h
PDO2 (tx)	0101	641 (281h) – 767 (2FFh)	1801h
PDO2 (rx)	0110	769 (301h) – 895 (37Fh)	1401h
PDO3 (tx)	0111	897 (381h) – 1023 (3FFh)	1802h
PDO3 (rx)	1000	1025 (401h) – 1151 (47Fh)	1402h
PDO4 (tx)	1001	1153 (481h) – 1279 (4FFh)	1803h
PDO4 (rx)	1010	1281 (501h) – 1407 (57Fh)	1403h
SDO (tx)	1011	1409 (581h) – 1535 (5FFh)	1200h
SDO (rx)	1100	1537 (601h) – 1663 (67Fh)	1200h
NMT Error Control	1110	1793 (701h) – 1919 (77Fh)	1016h, 1017h

Objets d'utilisation restreinte et réservés

COB-ID	used by object
0 (000h)	NMT
1 (001h)	reserved
257 (101h) – 384 (180h)	reserved
1409 (581h) – 1535 (5FFh)	default SDO (tx)
1537 (601h) – 1663 (67Fh)	default SDO (rx)
1760 (6E0h)	reserved
1793 (701h) – 1919 (77Fh)	NMT Error Control
2020 (780h) – 2047 (7FFh)	reserved

Transmission des données process

Le codeur CANopen dispose des **trois services PDO** PDO1 (tx) ,PDO2 (tx) et PDO3(tx). Une transmission PDO peut être déclenchée par différents événements (voir Répertoire d'objets Index 1800h) :

- **asynchrone** (géré par des événements), commandée par un temporisateur cyclique interne de l'appareil ou par un changement de valeurs process des données du capteur.
- **synchrone** en réponse à un télégramme SYNC ; (l'instruction SYNC donne l'ordre à tous les nœuds CANopen de sauvegarder leurs valeurs de manière synchrone, afin de les envoyer ensuite sur le bus conformément aux priorités définies)
- en **réponse** à un télégramme RTR (le Remote Frame=bit RTR récessif permet de demander exactement le message correspondant à l'identifiant transmis)

Selon le réglage standard, les **messages PDO** présentent la structure suivante :

Réglage par défaut du **mappage des PDO d'émission 1-3**:

Mappage	TPDO1 1800h	TPDO2 1801h	TPDO3 1802h
Objet mappé	1A00h	1A01h	1A02h
Enregistrement	0x60040020	0x60300110	0x60040020
Objet	6004h	6030h	6004h
Sous-index	00	01	00
Longueur des données	20h(32 bits)	10h(16 bits)	20h(32 bits)
	Asynchrone	Asynchrone	Synchrone

PDO d'émission 1 (1800h) Position asynchrone

ID COB par défaut : 180 + numéro de nœud : Exemple 180h + 3Fh = 1BFh

Message	Octet 0	Octet 1	Octet 2	Octet 3
1BF	LSB position	MSB position	00	00

Les valeurs de position peuvent avoir une valeur maximale de 0 – 3FFFh ou de 0 – 16383.

PDO d'émission 2 (1801h) Vitesse asynchrone

ID COB par défaut : 280 + numéro de nœud : Exemple 280h + 3Fh = 2BFh

Message	Octet 0	Octet 1
2BF	LSB vitesse	MSB vitesse

La valeur de vitesse est signée et se situe dans la plage de 0 – 1A00h ou de 0 – E600h.

PDO d'émission 3 (1802h) Position synchrone

ID COB par défaut : 380 + numéro de nœud : Exemple 380h + 3Fh = 3BFh

Message	Octet 0	Octet 1	Octet 2	Octet 3
3BF	LSB position	MSB position	00	00

Les valeurs de position peuvent avoir une valeur maximale de 0 – 3FFFh ou de 0 – 16383.

Transmission des données service

ID COB des SDO
Les identifiants suivants sont disponibles en standard pour les services SDO :

SDO (tx) (Codeur→Maître) : 580h (1408) + Numéro de nœud
SDO (rx) (Maître→Codeur) : 600h (1536) + Numéro de nœud
Il n'est pas possible de modifier les identifiants SDO !

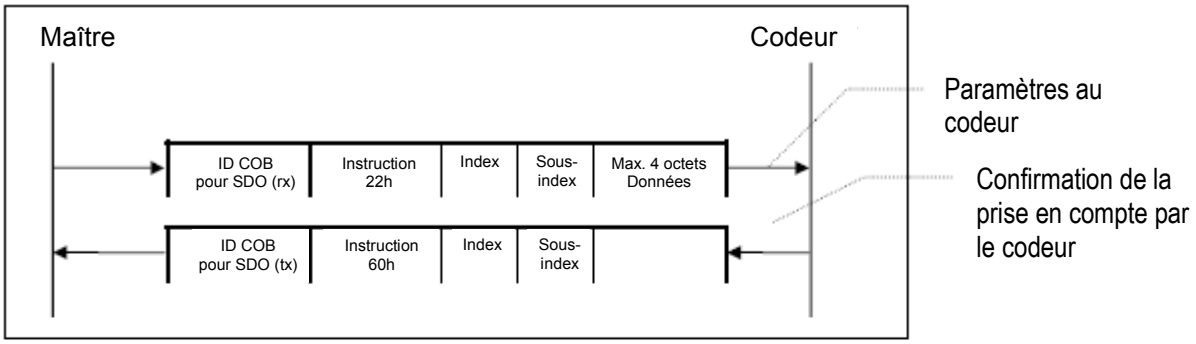
L'octet d'instruction décrit le type du message SDO :

Instruction (Expedited Protocol)	Type	Fonction
22h	SDO(rx), Initiate Download Request	Envoi des paramètres au codeur (longueur max. des données 4 octets)
23h	SDO(rx), Initiate Download Request	Envoi des paramètres au codeur (longueur des données 4 octets)
2Bh	SDO(rx), Initiate Download Request	Envoi des paramètres au codeur (longueur des données 2 octets)
2Fh	SDO(rx), Initiate Download Request	Envoi des paramètres au codeur (longueur des données 1 octet)
60h	SDO(tx), Initiate Download Response	Confirmation de la prise en compte au maître
40h	SDO(rx), Initiate Upload Request	Demande des paramètres du codeur
43h	SDO(tx), Initiate Upload Response	Paramètres au maître, longueur des données = 4 octets (unsigned 32)
4Bh	SDO(tx), Initiate Upload Response	Paramètres au maître, longueur des données = 2 octets (unsigned 16)
4Fh	SDO(tx), Initiate Upload Response	Paramètres au maître, longueur des données = 1 octet (unsigned 8)
80h	SDO(tx), Abort Domain Transfer	Le codeur envoie un code d'erreur au maître

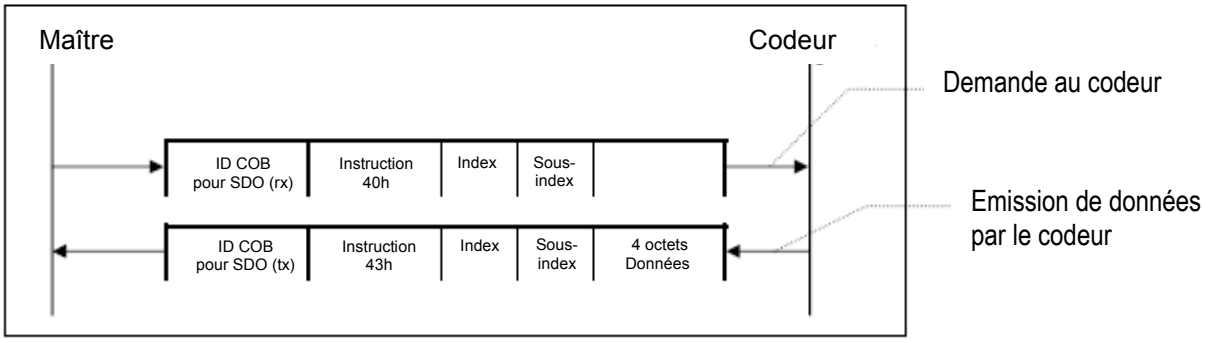
En cas d'erreur, un message d'erreur (instruction 080h) remplace la confirmation normale (response).
Le message d'erreur comprend les erreurs du protocole de communication (p. ex. octet d'instruction erroné) comme les erreurs d'accès au répertoire d'objets (p. ex. index erroné, erreur d'écriture sur un objet en lecture seule, longueur des données erronée, etc.)

Les codes d'erreur sont décrits dans le profil CANopen (DS 301) et dans le profil d'appareil (DSP 406).

Exemple d'une transmission de données service vers et depuis le codeur



Transmission de paramètres par le maître au codeur



Envoi d'une demande de paramètres par le maître au codeur

Services LSS

Exigences hardware LSS (adresse LSS)

Tous les esclaves LSS doivent posséder un enregistrement d'objet valide pour l'objet Identity [1018h] dans le répertoire d'objet afin de permettre une configuration sélective du nœud. Cet objet se compose des sous-indes suivants :

- **Vendor-ID (valeur numérique)**
- **Product-Code (valeur numérique)**
- **Revision-Number (révision importante et secondaire sous la forme d'une valeur numérique)**
- **Serial-Number (valeur numérique)**
- **LSS Master ID CAN 2021**
- **LSS Slave ID CAN 2020**

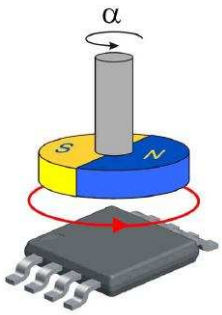
Un code produit, un numéro de révision et un numéro de série sont définis par le constructeur. L'adresse LSS doit être univoque dans le réseau.

Restrictions opérationnelles LSS
Afin de garantir des fonctionnalités LSS sans défauts, tous les appareils du réseau doivent supporter les services LSS. Il ne peut y avoir qu'un seul maître LSS. Tous les nœuds doivent démarrer avec la même vitesse de transmission.
La communication LSS ne peut avoir lieu que dans les modes "Stop" ou « Pre-Operational ».

2 Tension d'alimentation et raccordement au bus CAN

Affectation du câble CAN

Capteur :	**Capteur magnétique Hall**
	Résolution 14 bits / Précision 9 bits
Tension d'alimentation :	**10 … 30 VDC**
Consommation de courant :	typ. **22mA** sous 24 VDC
	max. 49 mA sous 10 VDC
Protection contre l'inversion de la polarité de l'alimentation :	oui
Emetteur CAN	82C251 / protégé contre les courts-circuits
Isolation galvanique	non
Communication CANopen	**DS 301 V4.02**
Profil d'appareil codeur CAN	**DS406 V 3.1**



Abréviation	Désignation	Couleur
CG	**CAN Ground**	**gris**
CL	**CAN_Low (-)**	**jaune**
CH	**CAN_High (+)**	**vert**
0V	**0 volt alimentation**	**blanc**
+V	**+UB alimentation**	**brun**

Raccordement du départ de câble

Raccordement CANopen connecteur M12

Abréviation	Désignation	Bro-che	Couleur
CG	**CAN Ground**	**1**	**Gy**
CL	**CAN_Low (-)**	**5**	**Ye**
CH	**CAN_High (+)**	**4**	**Gn**
0V	**0 volt alimentation**	**3**	**Wh**
+V	**+UB alimentation**	**2**	**Bn**

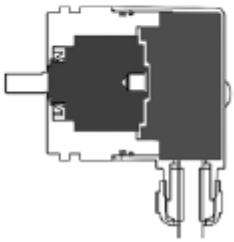
Raccordement du connecteur M12

Caractéristiques mécaniques

Mechanische Kennwerte:

Max. Drehzahl:	6000 min^{-1}
Anlaufdrehmoment:	< 0,06 Nm
Wellenbelastbarkeit radial:	40 N
Wellenbelastbarkeit axial:	20 N
Gewicht:	ca. 0,2 kg
Schutzart EN 60 529:	IP 67 (IP 69k auf Anfrage)
Arbeitstemperaturbereich:	–40 °C ... +85 °C
Werkstoffe:	Welle: nicht rostender Stahl, Flansch: Aluminium, Gehäuse: Zink-Druckgussgehäuse, Kabel: PUR
Schockfestigkeit nach DIN-IEC 68-2-27:	5000 m/s^2, 6 ms
Vibrationsfestigkeit nach DIN-IEC 68-2-6:	300 m/s^2, 10 ... 2000 Hz
Dauerschocken nach DIN-IEC 68-2-29	1000 m/s^2, 2 ms
Vibration (Breitbandrauschen) nach DIN-IEC 68-2-64	5 ... 2500 Hz, 100 m/s^2 - rms

Rundum geschützt dank Safety-Lock*plus*™ und Sensor-Protect™ Technologie



Safety-Lock*plus*™:
Flanschseitig IP69k, robuste Lagerbaugruppen mit verblockten Lagern, mechanisch geschützte Wellendichtung

Sensor-Protect™:
Vollvergossene Elektronik, getrennte mechanische Baugruppe

3 Première mise en service - Réglages généraux de l'appareil

Vitesse de transmission

La vitesse de transmission peut se modifier au moyen d'un **logiciel CANopen sur l'objet 2100h** ou au moyen du **service LSS** correspondant.
Réglage par défaut : 250 kbit/s (valeur programmée 5)

Valeur	Vit. de trans. en kbit/s
0	10
1	20
2	50
3	100
4	125
5	250
6	500
8	1000

A prendre en compte pour la vitesse de transmission correspondante

Le temps de cycle sélectionné (voir objet 1800h,sous-index 5 Event Timer) doit être supérieur à la durée de la transmission sur le bus, afin de permettre une transmission sans erreur des PDO !

Vitesse de transmission 10 kbauds : Temps de cycle minimal 14 ms
Vitesse de transmission 20 kbauds : Temps de cycle minimal 10 ms
Vitesse de transmission 50 kbauds : Temps de cycle minimal 4 ms

Pour un temps de cycle=0 en mode Event (c.-à-d. PDO en cas de changement de valeur), la vitesse de transmission doit être au minimum de **125 Kbauds**.

Numéro de nœud

Le numéro de nœud peut également se modifier **par logiciel sur l'objet 2101h** ou au moyen du **service LSS** correspondant.
Réglage par défaut : 0x3F (63 décimal).

Le **numéro de nœud 0** est réservé et ne doit pas être utilisé pour définir un nœud.
Les numéros de nœud résultants se trouvent dans la plage **1...7Fh** hexadécimal (1...127 décimal).

Un nouveau numéro de nœud n'est pris en compte qu'au redémarrage suivant (Reset/Power-on) du codeur ou au moyen d'une instruction **NMT-Reset Node**
Tous les réglages dans le tableau d'objets sont remis à leurs valeurs standard/par défaut.

Terminaison

La **terminaison de bus CAN** peut également se modifier **par logiciel à l'aide de l'objet 2102h**.
Réglage par défaut : 0x01 (terminaison de bus activée)
Le bus CAN bouclé à travers tous les participants doit être terminé à ses deux extrémités par une résistance terminale de bus de 120 ohms placée entre CAN+ et CAN-.

Il faut noter que seule la terminaison du **dernier appareil** de la ligne du bus CAN doit être activée.

Save All Bus Parameters

Ce paramètre (**objet 2105h**) sauvegarde les paramètres désirés du bus (objet 2100h, 2101h, 2102h) de manière permanente dans la mémoire flash. Cet objet apporte une sécurité supplémentaire contre des modifications involontaires de la vitesse de transmission et de l'adresse de nœud.

Seule une sauvegarde ciblée à l'aide du paramètre **« save » (hexadécimal 0x65766173)** permet une sauvegarde permanente des paramètres du bus : **vitesse de transmission, numéro de nœud et terminaison**.

4 Layer Setting Services (LSS)

Deux conditions doivent impérativement être remplies par les appareils reliés à un réseau CANopen – tous les appareils doivent avoir la même **vitesse de transmission** et chacun doit avoir une **adresse de nœud** unique dans le réseau. Pour une utilisation sous LSS, l'appareil doit disposer s'une liaison CAN 1:1. Un mode de dialogue spécial permet ensuite la modification de la **vitesse de transmission** et de l'**adresse de nœud**. L'**ID COB 0x7E5** est envoyé du maître à l'esclave, celui-ci répondant avec l'**ID COB 0x7E4**.
Les messages LSS ont toujours une longueur de 8 octets. Les octets inutilisés sont réservés et doivent être remplis par des 0.

Une instruction « **Switch Mode Global** » est émise pour mettre un appareil en mode configuration LSS.

0x04	**0x01**	**reserved**

Par malchance, cette instruction n'est pas confirmée et seul un contrôle visuel des LED permet de vérifier si l'appareil est effectivement passé dans ce mode.

L'instruction suivante est l'appel du service « **Inquire Node-ID** » :

0x5E	**reserved**

Si cette instruction s'effectue normalement, l'esclave répond par :

0x5E	**Node ID**	**reserved**

Si aucune réponse ne revient en retour de l'appareil, il est probable que celui-ci ne supporte pas le **service LSS** ou que la vitesse de transmission ne soit pas correcte.

L'instruction « **Configure Node-ID** » est utilisée pour reconfigurer l'**adresse de nœud** :

0x11	**Node ID**	**reserved**

Le code d'erreur est contenu dans la réponse de l'appareil esclave :

0x11	**Error code**	**Error extension**	**reserved**

Le code d'erreur 0 indique l'acceptation de l'instruction – Le code d'erreur 1 indique un identifiant de nœud invalide.
Les autres codes d'erreur sont réservés. L'extension de l'erreur contient des informations spécifiques au constructeur et ne peut s'utiliser que pour le code d'erreur 0xFF.

La vitesse de transmission est activée par l'intermédiaire de l'instruction « **Configure Bit Timing Parameters** ».

0x13	**Bit timing**	**Table entry**	**reserved**

Vitesses de transmission normalisées par CiA :

Baudrate table 0x00	
Table index	**Baudrate**
0	1000 kbit/s
1	800 kbit/s
2	500 kbit/s
3	250 kbit/s
4	125 kbit/s
5	reserved
6	50 kbit/s
7	20 kbit/s
8	10 kbit/s

De nouveau, l'appareil répond par un code d'erreur :

0x13	**Error code**	**Error extension**	**reserved**

Le code d'erreur 0 indique l'acceptation de l'instruction – Le code d'erreur 1 indique un identifiant de nœud invalide.
Les autres codes d'erreur sont réservés. L'extension de l'erreur contient des informations spécifiques au constructeur et ne peut s'utiliser que pour le code d'erreur 0xFF.
Les deux paramètres spécifiques au réseau sont modifiés ; il faut maintenant sauvegarder ces nouveaux paramètres à l'aide de « **Store Configuration** » :

0x17	**reserved**

De nouveau, l'appareil répond par un code d'erreur :

0x17	**Error code**	**Error extension**	**reserved**

Le code d'erreur 0 indique l'acceptation de l'instruction – Le code d'erreur 1 indique un identifiant de nœud invalide.
Les autres codes d'erreur sont réservés. L'extension de l'erreur contient des informations spécifiques au constructeur et ne peut s'utiliser que pour le code d'erreur 0xFF.

Pour terminer le service LSS, il faut ramener l'appareil du mode de configuration LSS dans le **mode Preoperational** à l'aide de l'instruction « **Switch Mode Global** » :

0x04	**0x00**	**reserved**

L'appareil effectue **automatiquement un redémarrage (Reset node)**, après quoi tous les réglages sont activés.

5 Réglages par défaut à la livraison

Les paramètres logiciel sont réglés comme suit en usine pour la livraison.

Désignation	Réglage	Logiciel
Vitesse de transmission	**250 kbit/s**	Objet 2100h = 05h
Adresse de nœud	**63**	Objet 2101h = 3Fh
Terminaison	**Activée**	Objet 2102h = 01h

Index (hex)	Nom	Valeur standard
	Paramètres de communication	
1005h	COB-ID Sync	80h
100Ch	Guard Time	0
100Dh	Life Time Factor	0
1012h	COB-ID Time stamp	100h
1013h	High Resolution time stamp	0
1017h	Producer heartbeat time	0
1029h	Error Behaviour	0 = Comm Error
		1 = Device specific
		1 = Manufacturer Err.
1800h	**TPDO1** Communication Parameter	
01h	COB-ID	180h + Numéro de nœud
02h	Transmission Type	255 (asynch)
03h	Inhibit Time	0 [step 100 µs]
05h	Event timer	0 [step 1 ms]
1801h	**TPDO2** Communication Parameter	
01h	COB-ID	280h + Numéro de nœud
02h	Transmission Type	255 (asynch)
03h	Inhibit Time	0 [step 100 µs]
05h	Event timer	0 [step 1 ms]
1802h	**TPDO3** Communication Parameter	
01h	COB-ID	380h + Numéro de nœud
02h	Transmission Type	1 (synchrone)
03h	Inhibit Time	0 [step 100 µs]
05h	Event timer	0 [step 1 ms]
1A00h	TPDO1 Mapping	
01h	1.Mapped Object	0x60040020
1A01h	TPDO2 Mapping	
01h	1.Mapped Object	0x60300110
1A02h	TPDO3 Mapping	
01h	1.Mapped Object	0x60040020

Index (hex)	Nom	Valeur standard
	Profil Codeur	
6000h	Operating Parameter	0h Scaling off
6001h	Measuring Units per Revolution	16384 (14 Bit)
6002h	Total Measuring Range	16384 (14 Bit)
6003h	Preset value	0
6200h	Cyclic Timer (see TPDO1 Comm.Par)	0
6401h	Work area low limit	0
6402h	Work area high limit	16384
2100h	Baudrate	05h
2101h	Node number	63
2102h	Terminaison bus CAN	1 (activée)
2105h	Save All Bus Parameters	0x65766173

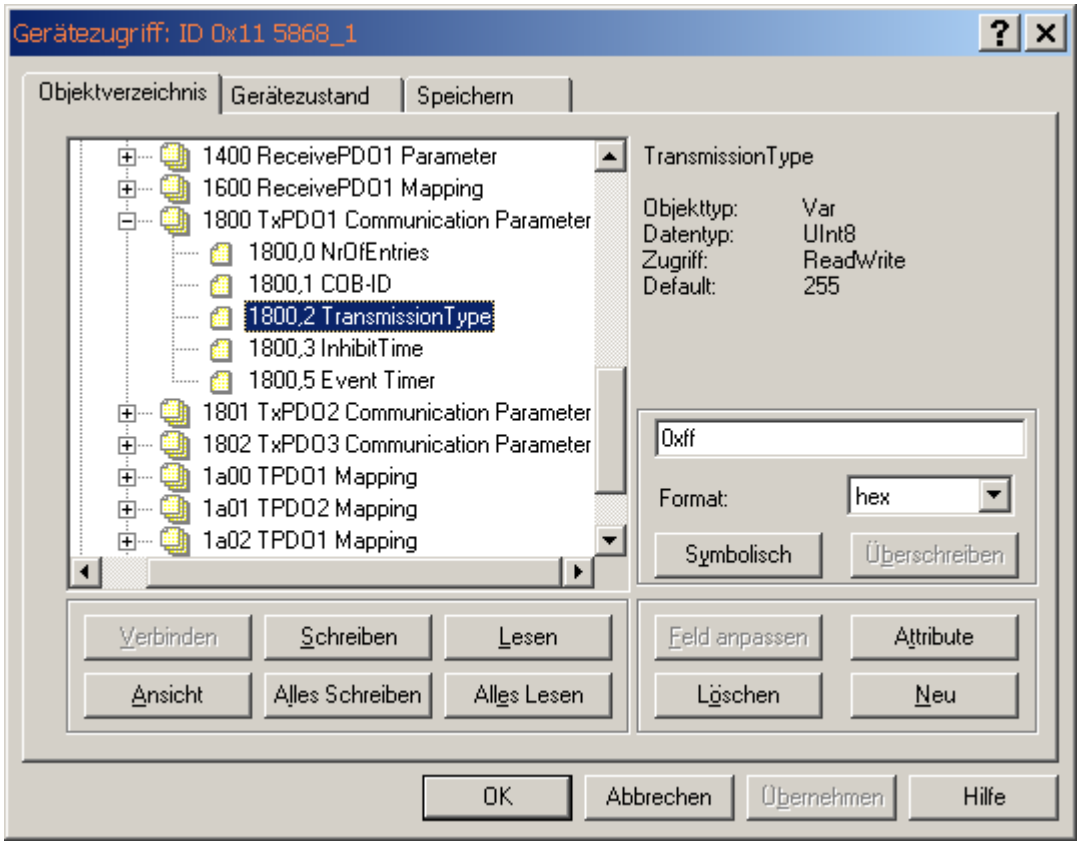
Les valeurs standard d'origine (valeurs par défaut à la livraison) peuvent être rechargées à l'aide de l'objet **1011h** **(restauration des paramètres)**.
avec l'indication « load » **(0x6C6F6164)**

Pour sauvegarder les paramètres modifiés de sorte à les protéger contre les pannes de courant, il faut impérativement les transférer dans la mémoire Flash à l'aide de l'objet **1010h** (sauvegarde des paramètres).
Toutes les données sauvegardées auparavant dans la mémoire Flash sont écrasées !

Si des erreurs sont survenues lors de la programmation des objets et si ces paramètres ont été sauvegardés dans la mémoire Flash, le codeur ne répondra plus à la mise en service suivante (il n'émettra plus que des messages **Emergency**).
Ce défaut ne peut être corrigé que par un **Reset** général du codeur.

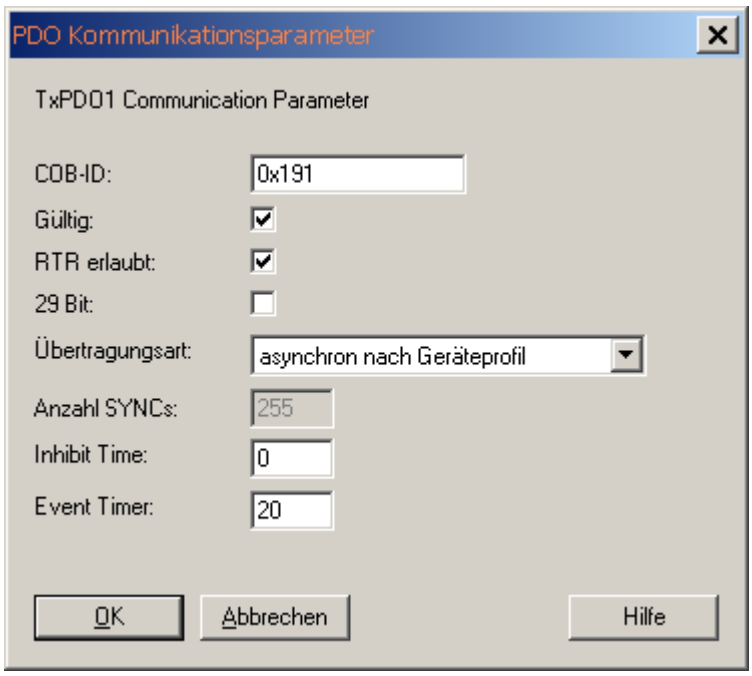
6 Paramètres de communication

L'ID COB et le type de transmission pour **PDO1** sont définis dans le répertoire d'objet, index **1800h**.

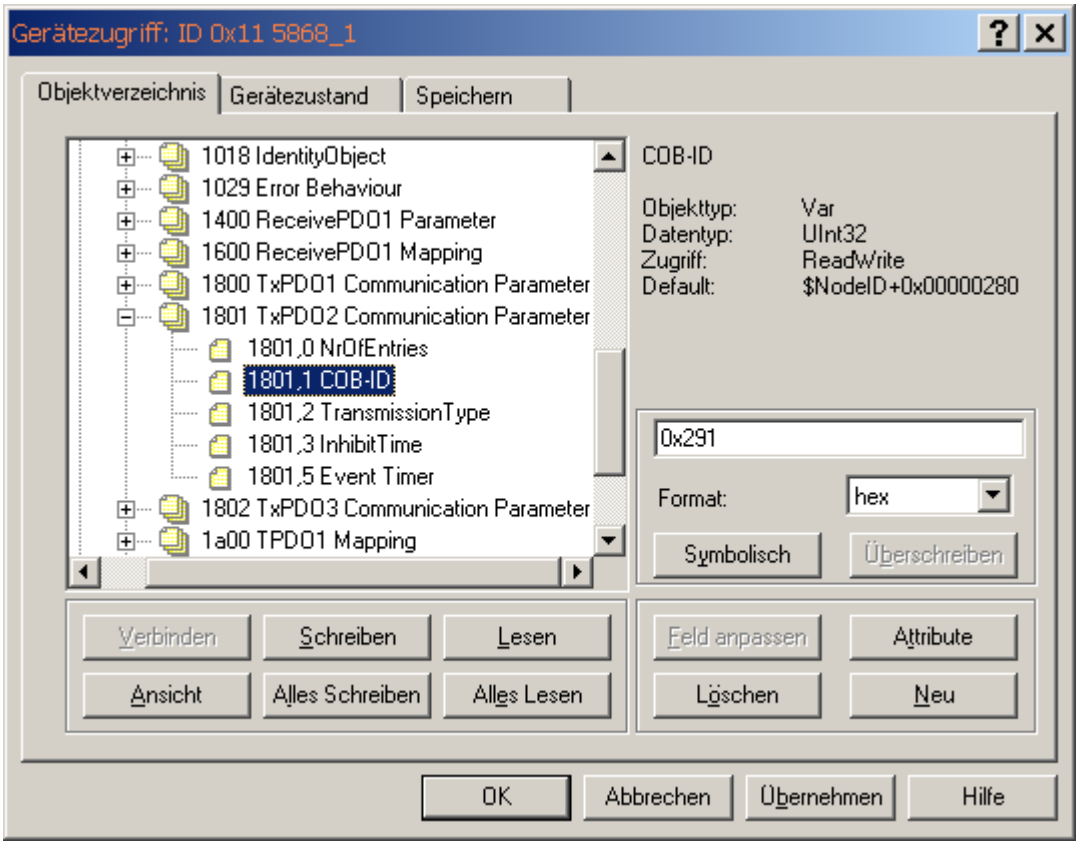


Réglages par défaut :

Validation :	**PDO valide (enabled)**	**RTR autorisé**
ID COB :	**180h + numéro de nœud réglé (ici 11h)**	
Type de transmission :	**255 = asynchrone selon le profil d'appareil**	
Event Timer :	**20 ms**	

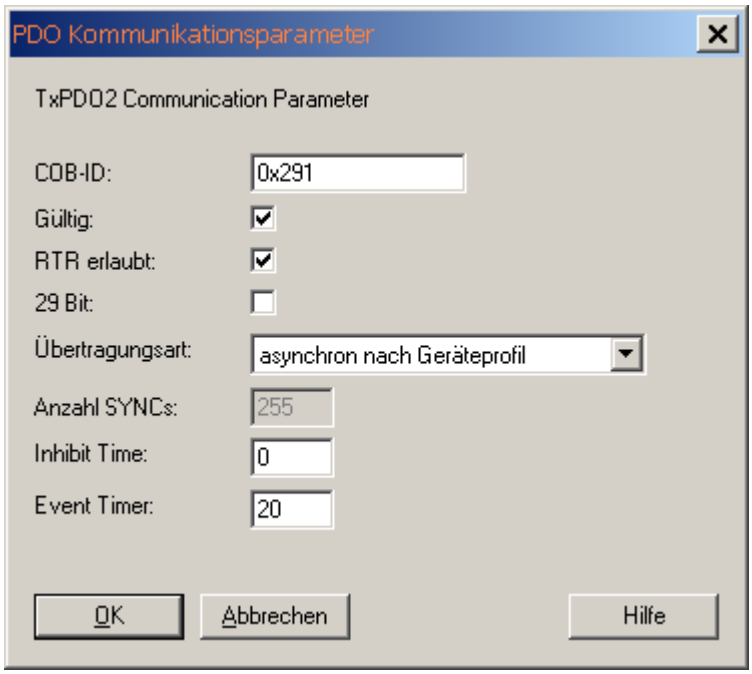


L'ID COB et le type de transmission pour **PDO2** sont définis dans le répertoire d'objet, index **1801h**.



Réglages par défaut :

Validation :	**PDO valide (enabled)**	**RTR autorisé**
ID COB :	**280h + numéro de nœud réglé (ici 11h)**	
Type de transmission :	**255 = asynchrone selon le profil d'appareil**	
Event Timer :	**20 ms**	



Définition du type de transmission (Transmission type) du PDO

transmission type	PDO transmission				
	cyclic	acyclic	synchronous	asynchronous	RTR only
0		X	X		
1-240	X		X		
241-251	- reserved -				
252			X		X
253				X	X
254				X	
255				X	

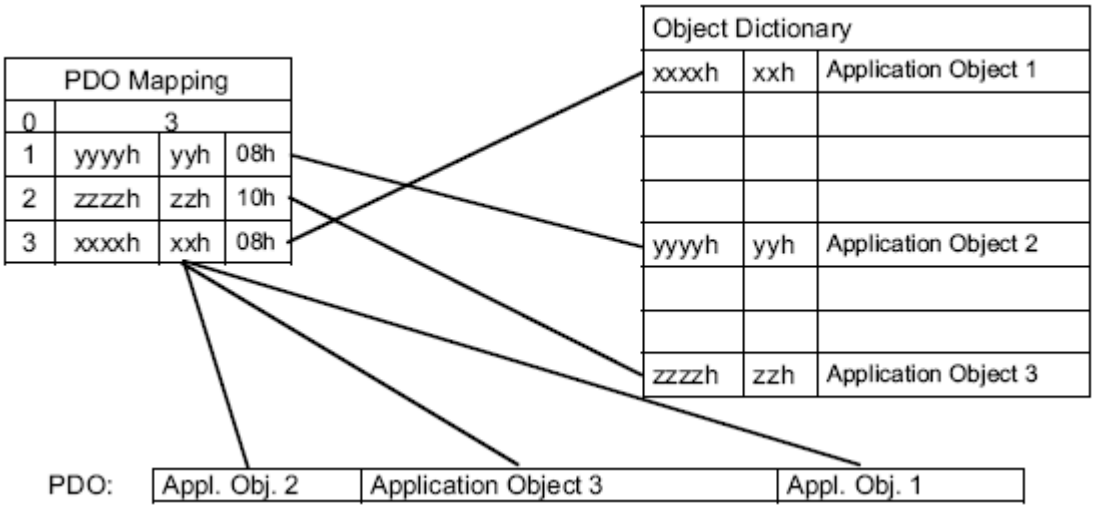
Une valeur entre 1 … 240 indique une émission **synchrone et cyclique** du PDO. Le numéro du type de transmission indique le **nombre d'impulsions SYNC** nécessaire à l'émission des PDO.
Les types de transmission 252 et 253 indiquent que le PDO n'est émis via RTR que sur demande.

Le type 254 indique que l'événement est déclenché selon l'application, alors que le numéro 255 indique un déclenchement selon le profil d'appareil. Il est en outre possible d'utiliser, pour les numéros 254/255, un **EventTimer** commandé par une horloge. La plage de valeurs de cette horloge s'étend de **1ms ... 65535 ms**.

Mappage PDO variable

Le mappage variable des différents objets signifie que l'utilisateur est en mesure de configurer le contenu des PDO d'émission en fonction de son application.

Exemple d'un enregistrement dans le tableau de mappage:
Le PDO mappé se compose de 3 enregistrements d'objets applicatifs de longueurs différentes :

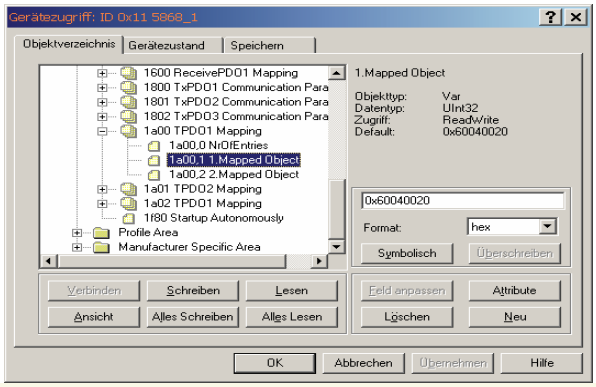


L'objet applicatif 2 occupe un octet (08h) dans le PDO d'émission. Il est suivi de l'objet applicatif 3 d'une longueur de 16 bits (10h = 2 octets), lui-même suivi finalement de l'objet applicatif 1 d'une longueur de 1 octet. En tout, 32 bits sont occupés dans ce PDO.

7 Exemple d'un enregistrement de mappage variable

Objet Mappage 1A00h

L'objet Mappage **1A00h décrit le 1er PDO d'émission**. Il est possible de mapper des objets jusqu'à une longueur de données maximale de **8 octets**. De la même manière, **l'objet 1A01h** décrit le **PDO d'émission 2** et **l'objet 1A02h** décrit le **PDO d'émission 3**.



Mappage	Mappage TPD01	Mappage TPD01	Mappage TPD01
Sous-index	00	01	02
Enregistrement	Nbre d'enregistr.	1er objet mappé	2ème objet mappé
Objet	2	6004h	6030h
Sous-index		00	01
Long. Données	Octets	20h(32 Bit)	10h(16 Bit)
		Asynchrone	Asynchrone

Les objets suivants peuvent être mappés: **(signalés en bleu)**

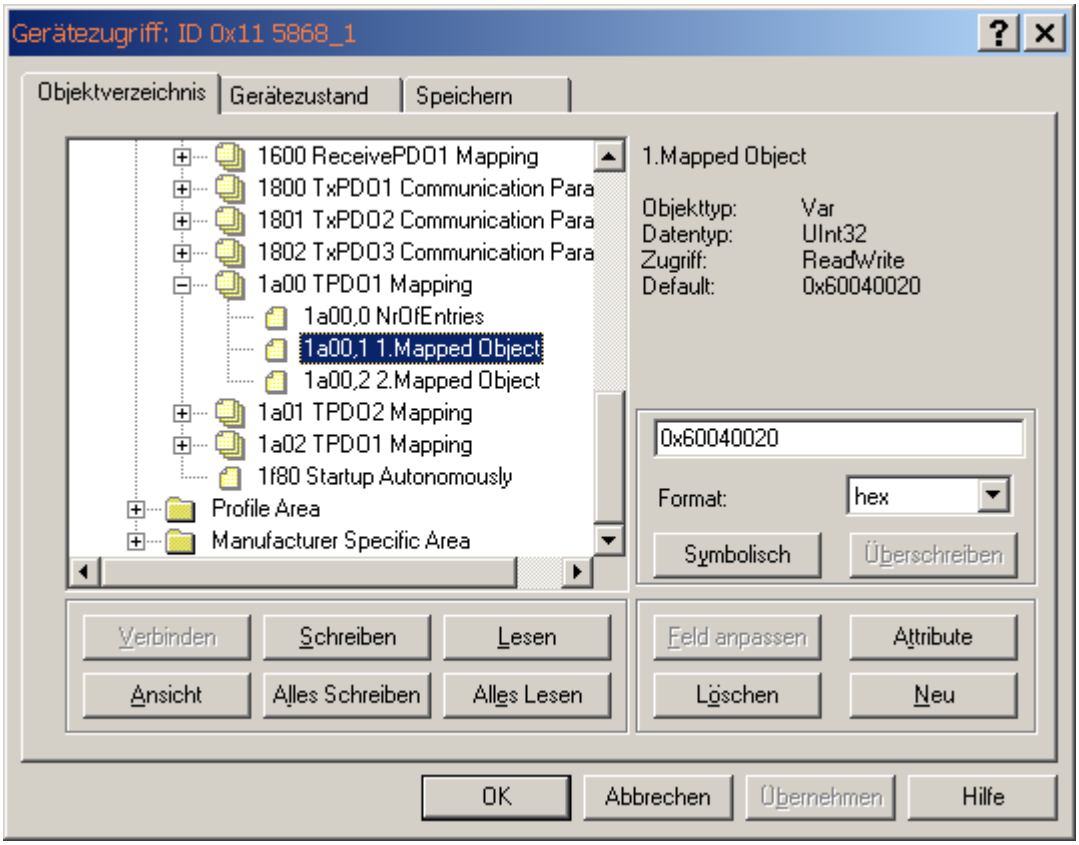
Objets spécifiques aux appareils

INDEX (hex)	Object Symb.	ATTRIB	Name	M/O C2	TYPE
6000	VAR	RW	Operating parameters	M	unsigned16
6001	VAR	RW	Measuring Units p.Revolution (MUR)	M	unsigned32
6002	VAR	RW	Total Measuring Range (TMR)	M	unsigned32
6003	VAR	RW	Preset value	M	unsigned32
6004	**VAR**	**RO**	**Position value**	**M MAP**	**unsigned32**
6030	**ARRAY**	**RO**	**Speed Value**	**O MAP**	**signed16**
6040	ARRAY	RO	Acceleration Value	O	Signed16
6200	VAR	RW	Cyclic Timer	M	unsigned16
6400	**ARRAY**	**RO**	**Working Area state**	**O MAP**	**unsigned 8**
6401	ARRAY	RW	Working Area Low Limit	O	Unsigned32
6402	ARRAY	RW	Working Area High Limit	O	Unsigned32
6500	VAR	RO	Operating Status	M	unsigned16
6501	VAR	RO	Measuring Step (Singleturn)	M	unsigned32
6502	VAR	RO	Number of revolutions	M	unsigned16
6503	**VAR**	**RO**	**Alarms**	**M MAP**	**unsigned16**
6504	VAR	RO	Supported alarms	M	unsigned16
6505	**VAR**	**RO**	**Warnings**	**M MAP**	**unsigned16**

Dans cet exemple, 2 objets de données process, l'objet 6004h et l'objet 6030h, sont mappés.

8 Structure d'un enregistrement de mappage

L'objet Mappage pour **PDO 1** est défini dans le répertoire d'objets index 1A00h. Il se compose d'un enregistrement et il peut être modifié par l'utilisateur (mappage variable).



Byte:	MSB		LSB
	index (16 bit)	sub-index (8 bit)	object length (8 bit)

Réglage par défaut du **mappage des PDO d'émission**:

Mappage	TPDO1	TPDO2	TPDO3
1er mappage	**0x60040020**	**0x60300110**	**0x60040020**
Objet	**6004h**	**6030h**	**6004h**
Sous-index	**00**	**01**	**00**
Longueur des données	**20h(32 bits)**	**10h(16 bits)**	**20h(32 bits)**
	Asynchrone	**Asynchrone**	**Synchrone**

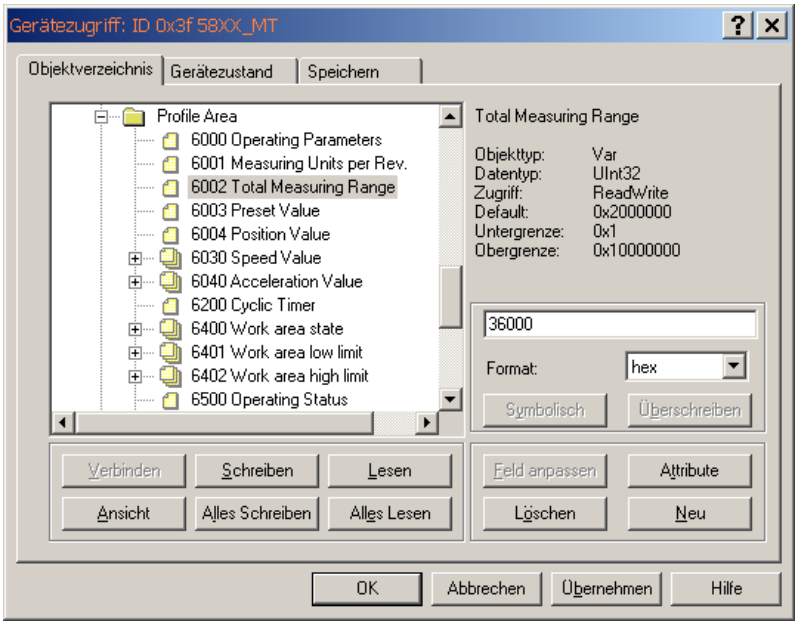
Le codeur CANopen supporte un **mappage variable** sur les 3 PDO d'émission.

9 Exemple de programmation d'une application:

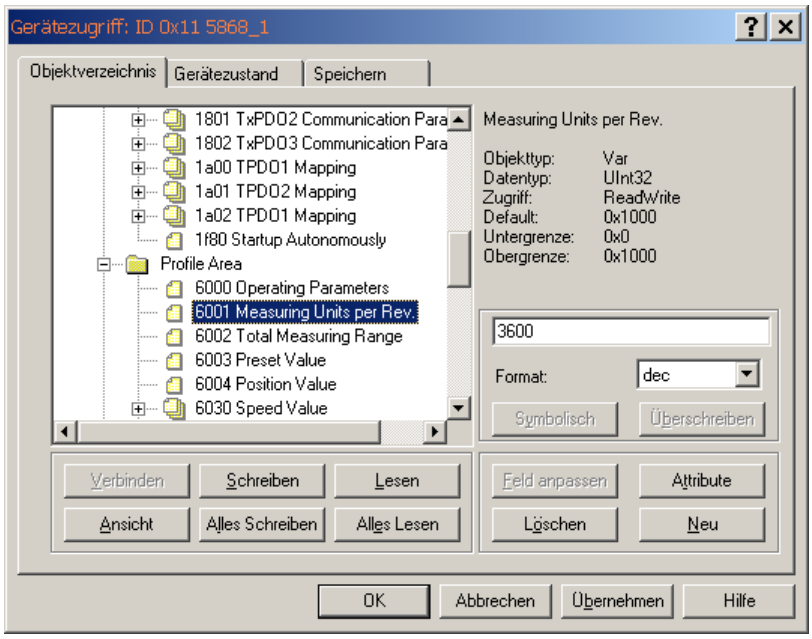
Réglage des objets

- Limiter **Total Measuring Range** à 3600
- **Measuring Units per Revolution** doit être réglé à 3600 pas par tour
- **Positionswert** (valeur de position) doit être réglé à 0.
- L'émission de TPDO1 (Position) doit être commandée par un événement toutes les 10 ms.
- L'émission de TPDO2 (Speed) doit être commandée par un événement toutes les 20 ms.
- **Producer Heartbeat** doit être réduit à 500 ms.
- Les valeurs de Work area Limit seront 1000 et 3500
- Les nouveaux paramètres doivent être sauvegardés dans la mémoire Flash.

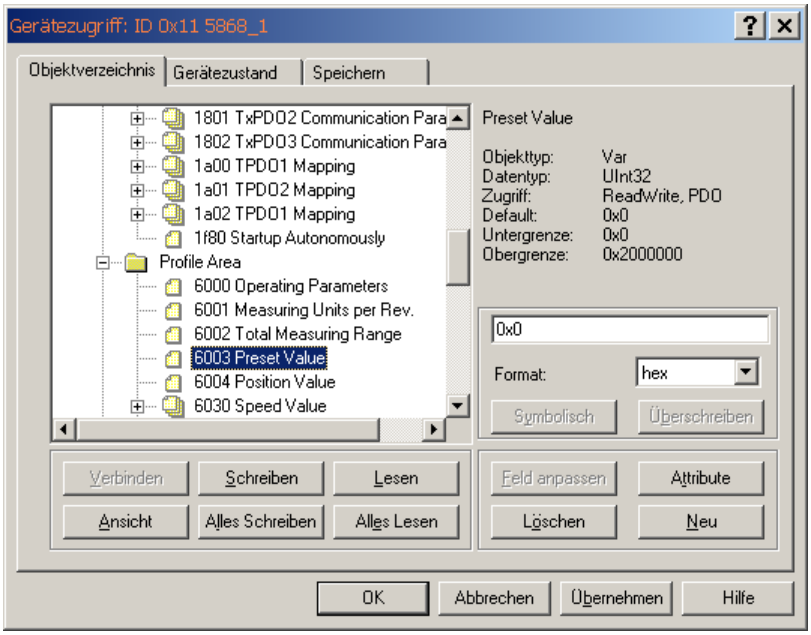
Limiter Total Measuring Range à 3600



Limiter Measuring Units per Revolution à 3600

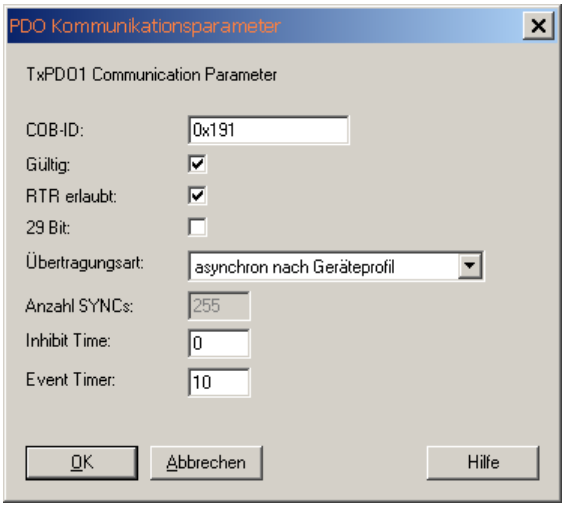


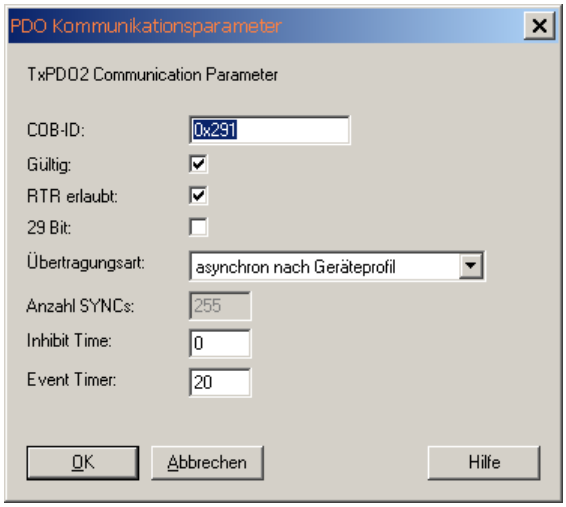
Preset value à 0



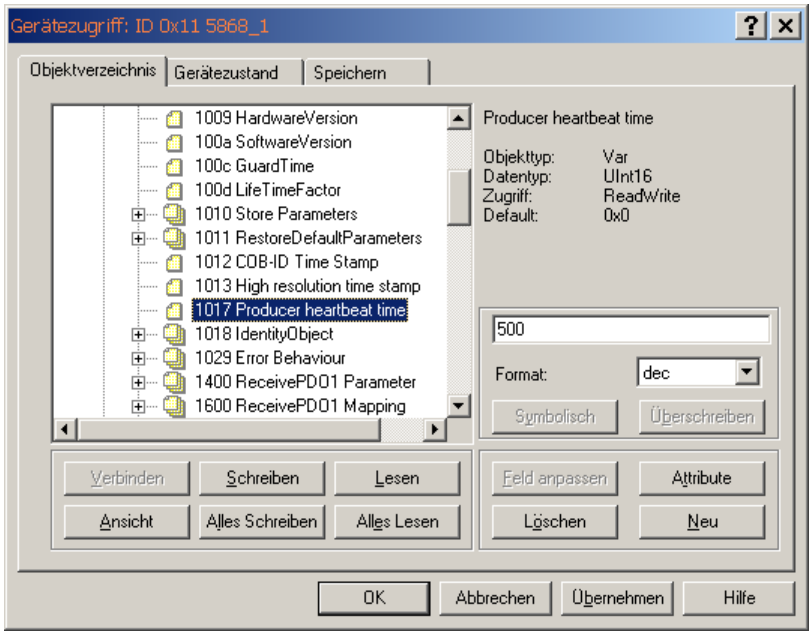
Définition des valeurs des paramètres d'émission de TPDO1 et TPDO2

Le type 254 indique que l'événement est déclenché selon l'application, alors que le numéro 255 indique un déclenchement **selon le profil d'appareil**. Il est en outre possible d'utiliser, pour les numéros 254/255, un **EventTimer** commandé par une horloge. La plage de valeurs de cette horloge s'étend de **1ms ... 65535 ms**.

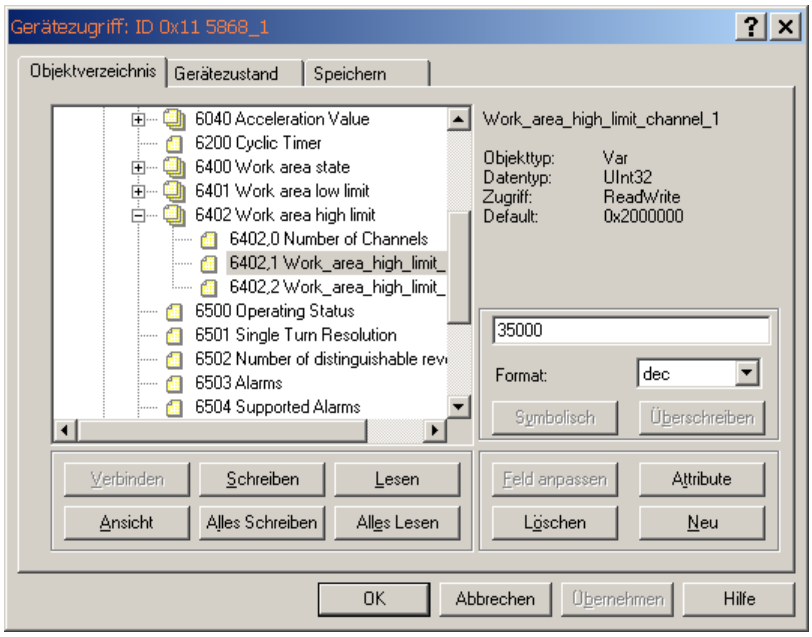




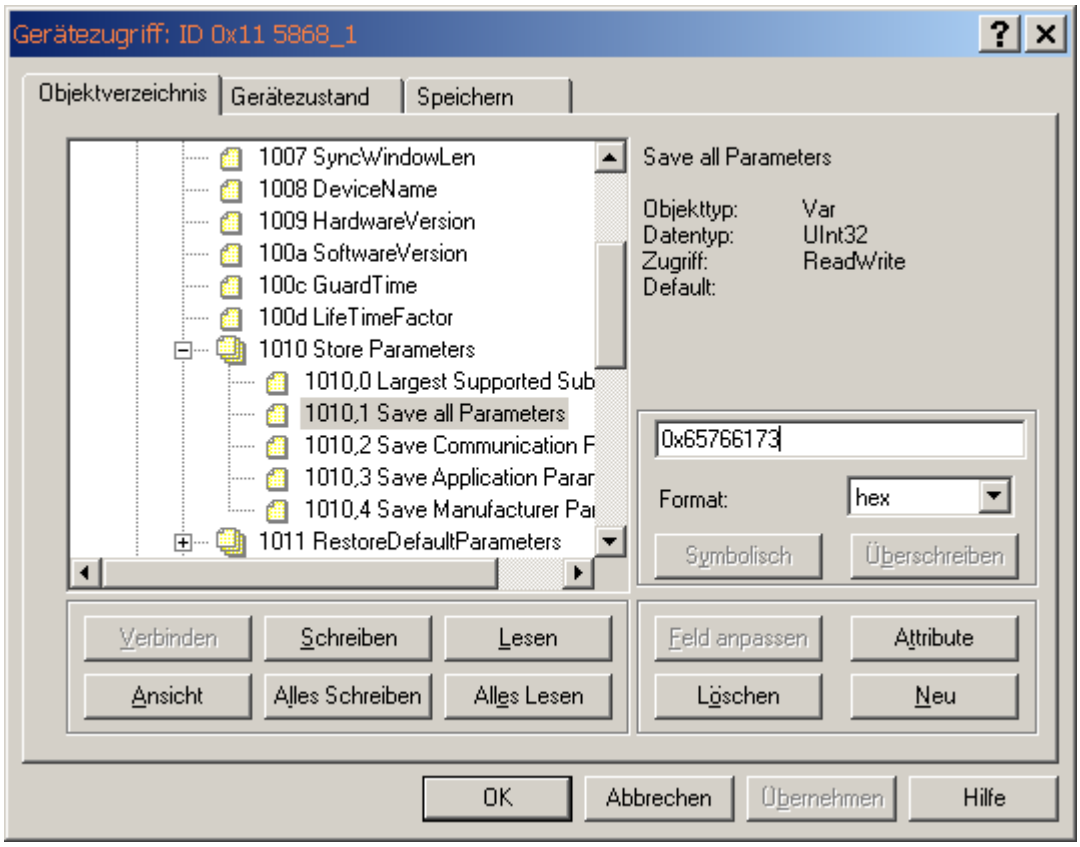
Régler Producer Heartbeat à 500 ms



Définir les limites basse et haute de Work area



Sauvegarde de tous les paramètres modifiés dans la mémoire Flash
Store Parameters 1010h



Objet 1010h Sauvegarde des paramètres

L'instruction « save » dans le sous-index 1h (save all Parameters) commande la sauvegarde des paramètres dans la mémoire non volatile (EEPROM).
Ce même sous-index permet la sauvegarde de tous les objets de communication, objets applicatifs et objets spécifiques au constructeur. Cette opération nécessite environ 14 ms.
Afin d'éviter une sauvegarde par mégarde, cette instruction n'est exécutée que si la chaîne de caractères « save » est enregistrée comme mot de code dans ce sous-index.
Un accès en lecture au sous-index 1h fournit des informations sur la fonctionnalité de sauvegarde.

Désignation	Contenu	Remarque
Octet 3	73h	(code ASCII pour « s »)
Octet 2	61h	(code ASCII pour « a »)
Octet 1	76h	(code ASCII pour « v »)
Octet 0	65h	(code ASCII pour « e »)

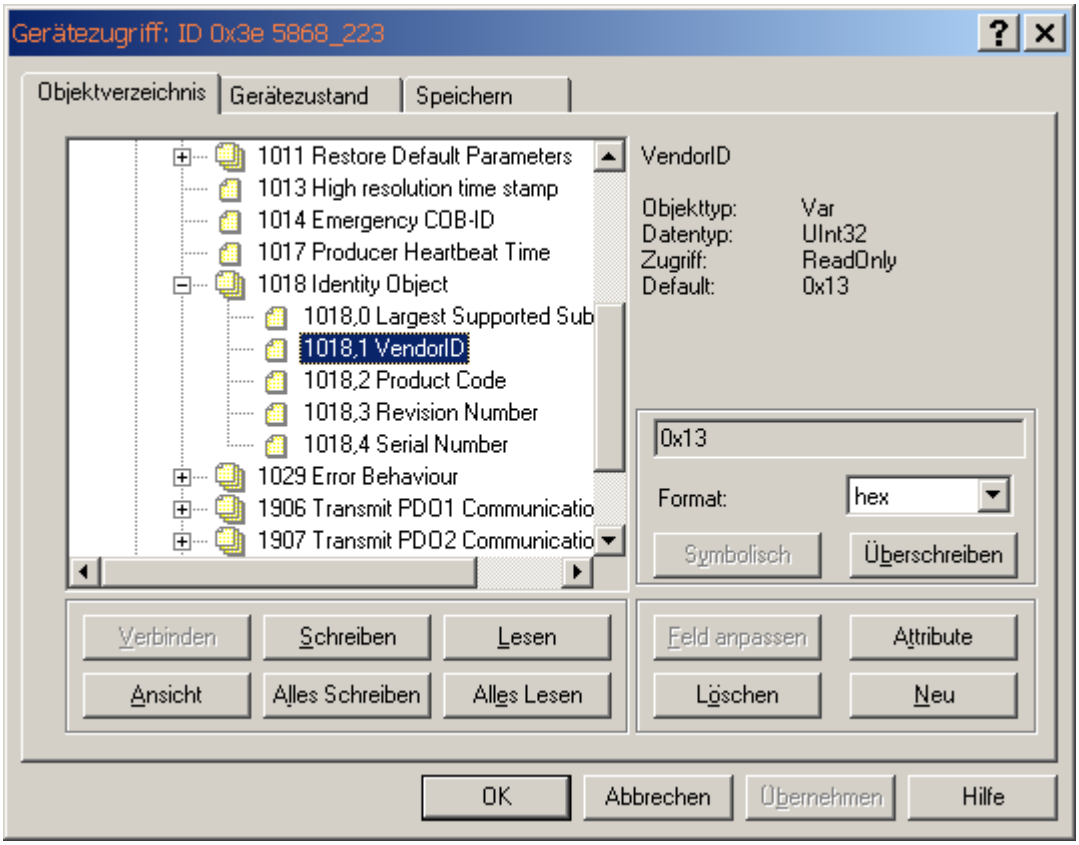
Objet 1011h : Chargement des valeurs standards

L'instruction « load » dans le sous-index 1h permet la réinitialisation de tous les paramètres à leurs valeurs standards. Afin d'éviter un chargement des valeurs standards par mégarde, cette instruction n'est exécutée que si la chaîne de caractères « load » est enregistrée comme mot de code dans ce sous-index.

Désignation	Contenu	Remarque
Octet 3	6Ch	(code ASCII pour « l »)
Octet 2	6Fh	(code ASCII pour « o »)
Octet 1	61h	(code ASCII pour « a »)
Octet 0	64h	(code ASCII pour « d »)

Profil de communication – autres objets
Objet 1018h : Identity Object

Informations sur le constructeur et sur l'appareil :



1018 RECORD Device – Identification, lecture seule

Sous-index 0h : Nombre de sous-index
renvoie la valeur 4
Sous-index 1h : Lecture seule
renvoie **l'identifiant du constructeur (000000013h) Fritz Kübler GmbH**
Sous-index 2h : renvoie le **code produit**
(p. ex. 0x365821FF Codeur CANopen)
Sous-index 3h : Lecture seule
renvoie **le numéro de révision du logiciel**
(p. ex. 102)
Sous-index 4h : Lecture seule
renvoie le **numéro de série** à 8 chiffres du codeur

10 Objets Emergency

Les objets Emergency apparaissent dans des situations d'erreurs dans un réseau CAN ; ils sont déclenchés en fonction des événements et émis sur le bus avec une **priorité haute**.

Important : Un objet Emergency n'est déclenché qu'une fois par « event ». Aucun nouvel objet n'est généré tant que l'erreur est présente. Lorsque l'erreur est éliminée, un nouvel objet Emergency est généré avec le contenu 0 (Error Reset ou No Error) et émis sur le bus.

Codes d'erreur supportés

Les codes d'erreur sont indiqués en rouge

Error Code (hex)	Meaning
00xx	Error Reset or No Error
10xx	Generic Error
20xx	Current
21xx	Current, device input side
22xx	Current inside the device
23xx	Current, device output side
30xx	Voltage
31xx	Mains Voltage
32xx	Voltage inside the device
33xx	Output Voltage
40xx	Temperature
41xx	Ambient Temperature
42xx	Device Temperature
50xx	Device Hardware
60xx	Device Software
61xx	Internal Software
62xx	User Software
63xx	Data Set
70xx	Additional Modules
80xx	Monitoring
81xx	Communication
8110	CAN Overrun (Objects lost)
8120	CAN in Error Passive Mode
8130	Life Guard Error or Heartbeat Error
8140	recovered from bus off
8150	Transmit COB-ID collision
82xx	Protocol Error
8210	PDO not processed due to length error
8220	PDO length exceeded
90xx	External Error
F0xx	Additional Functions
FFxx	Device specific

11 Message Emergency

Byte	0	1	2	3	4	5	6	7
Content	Emergency Error Code (see Table 21)		Error register (Object 1001H)	Manufacturer specific Error Field				

Figure 34: Emergency Object Data

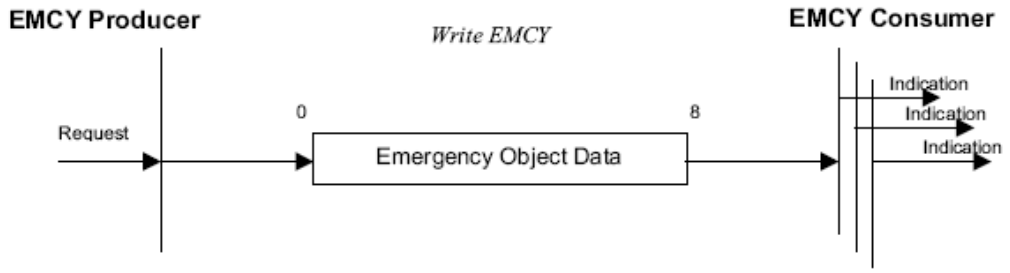
Exemple d'un message en cas de défaut hardware :

Transfer Data	00	52	09	80	56	20	50	2E

[Errcode]	**5200**	**Défaut hardware du capteur**
[Error Register]	**09**	**Registre d'erreur**
[ManufacturerSpecific1]	**80**	Octets de diagnostic
[ManufacturerSpecific2]	**56**	Octets de diagnostic
[ManufacturerSpecific3]	**20**	Octets de diagnostic
[ManufacturerSpecific4]	**50**	Octets de diagnostic
[ManufacturerSpecific5]	**2E**	Octets de diagnostic

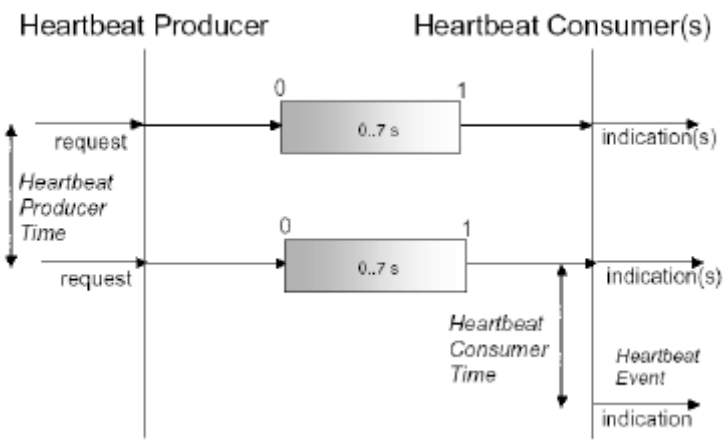
Protocole Emergency

Définition d'un message service « non confirmé »



Le comportement en cas d'erreur est décrit dans l'**objet 1029h Error Behavior**.

12 Protocole Heartbeat



En alternative au **Node-Guarding**, il faut dorénavant utiliser le protocole moderne **Heartbeat**. Ce protocole est activé en définissant dans l'**objet 1017h** Producer Heartbeat Time une valeur > 0.
Un « Heartbeat Producer » (émetteur Heartbeat) émet de message Heartbeat de manière cyclique. Un ou plusieurs « Heartbeat Consumers » (récepteurs Heartbeat) peuvent recevoir ce message Heartbeat.
Si l'émission cyclique de ce message Heartbeat disparaît, un « Heartbeat Event » se déclenche. Le comportement en cas de défaut est décrit dans l'objet 1029h sous-index 1 « Communication Error ».

13 Objet CANopen Répertoire

Structure des enregistrements du répertoire d'objets:

Index (hex)	Sous-index (hex)	Objet	Nom	Type	Attr.	M/O

Index: Adresse 16 bits de l'enregistrement
Sous-Index: Pointeur 8 bits pour les sous-enregistrements;
n'est utilisé que pour des structures de données complexes (p. ex. tableaux);
s'il n'y a pas de sous-enregistrement: sous-index = 0

Objet:	NULL	Enregistrement sans données
	DOMAIN	Quantité importante, variable de données, p. ex. code de programme
	DEFTYPE	Définition des types de données, p. ex. boolean, float, unsigned 16, etc.
	DEFSTRUCT	Définition d'un enregistrement, p. ex. structure de mappage PDO
	VAR	Valeur individuelle, p. ex. boolean, float, unsigned 16, string, etc.
	ARRAY	Champ de données de même type, p. ex. données unsigned 16
	RECORD	Champ de données de types différents

Nom : Brève description de la fonction
Type : Type de données, p. ex. boolean, float, unsigned 16, integer, etc.
Attr. : Attribut définit les droits d'accès à l'objet :

	rw	Lecture/écriture
	ro	Lecture uniquement
	const	Lecture uniquement, la valeur est une constante
M/O	M	Mandatory : l'objet doit être implémenté dans l'appareil
	O	Optional : l'objet ne doit pas être implémenté dans l'appareil

Structure du répertoire d'objets complet:

Index (hex)	Objet
0000	Non utilisé
0001 - 001F	types de données statiques
0020 - 003F	types de données complexes
0040 - 005F	types de données spécifiques au constructeur
0060 - 0FFF	réservé
1000 - 1FFF	profil de communication
2000 - 5FFF	profil spécifique au constructeur
6000 - 9FFF	profil d'appareil standardisé
A000 - FFFF	réservé

14 Profil de communication CANopen DS 301

Objets de communication

INDEX (hex)	OBJECT SYMBOL	ATTRIB	Name	M/O	TYPE
1000	VAR	CONST	Device Type	M	Unsigned32
1001	VAR	RO	Error Register	M	Unsigned8
1002	VAR	RO	Manufacturer Status	O	Unsigned32
1003	RECORD	RO	Predefined Error Field	O	Unsigned32
1004	ARRAY	RO	Number of PDO supported	O	Unsigned32
1005	VAR	RW	COB-ID Sync message	O	Unsigned32
1006	VAR	RW	Communication cycle period	O	Unsigned32
1007	VAR	RW	synchr.window length	O	Unsigned32
1008	VAR	CONST	Manufacturer Device Name	O	visible string
1009	VAR	CONST	Manufacturer Hardware Version	O	visible string
100A	VAR	CONST	Manufacturer Software Version	O	visible string
100B	VAR	RO	Node-ID	O	Unsigned32
100C	VAR	RW	Guard Time	O	Unsigned32
100D	VAR	RW	LifeTime Factor	O	Unsigned32
1010	VAR	RW	Store parameters (Device Profile)	O	Unsigned32
1011	VAR	RW	Restore parameters (Device Profile)	O	Unsigned32
1014	VAR	RO	COB_ID Emcy	O	Unsigned32
1015	VAR	RW	IInhibit Time Emcy	O	Unsigned32
1017	VAR	RW	Producer Heartbeat time	O	Unsigned16
1018	RECORD	RO	Identity Object	M	PDOComPar
1029	ARRAY	RW	Error Behaviour	O	Unsigned8
1800	**RECORD**		**1st transmit PDO Comm. Par.**	**O**	**PDOComPar**
1801	**RECORD**		**2nd transmit PDO Comm. Par.**	**O**	**PDOComPar**
1802	**RECORD**		**3rd transmit PDO Comm. Par.**	**O**	**PDOComPar**
1A00	**ARRAY**		**1st transmit PDO Mapping Par.**	**O**	**PDOMapping**
1A01	**ARRAY**		**2nd transmit PDO Mapping Par.**	**O**	**PDOMapping**
1A02	**ARRAY**		**3rd transmit PDO Mapping Par.**	**O**	**PDOMapping**

Objets spécifiques au constructeur

2100	VAR	RW	Baud Rate	O	Unsigned 8
2101	VAR	RW	Node number	O	Unsigned 8
2102	VAR	RW	CAN Bus Termination	O	Unsigned 8
2103	VAR	RO	Firmware Flash Version	O	Unsigned16
2105	VAR	RW	Save All Bus Parameters	O	Unsigned32
2140	Array	RW	Customer Memory	O	Unsigned32

15 Profil d'appareil CANopen pour codeurs DS 406 V3.1

Objets spécifiques à l'appareil

INDEX (hex)	Object Symb.	ATTRIB	Name	M/O C2	TYPE
6000	VAR	RW	Operating parameters	M	unsigned16
6001	VAR	RW	Measuring Units p.Revolution (MUR)	M	unsigned32
6002	VAR	RW	Total Measuring Range (TMR)	M	unsigned32
6003	VAR	RW	Preset value	M	unsigned32
6004	**VAR**	**RO**	**Position value**	**M MAP**	**unsigned32**
6030	**ARRAY**	**RO**	**Speed Value**	**O MAP**	**signed16**
6040	ARRAY	RO	Acceleration Value	O	Signed16
6200	VAR	RW	Cyclic Timer	M	unsigned16
6400	**ARRAY**	**RO**	**Working Area state**	**O MAP**	**unsigned 8**
6401	ARRAY	RW	Working Area Low Limit	O	Unsigned32
6402	ARRAY	RW	Working Area High Limit	O	Unsigned32
6500	VAR	RO	Operating Status	M	unsigned16
6501	VAR	RO	Measuring Step (Singleturn)	M	unsigned32
6502	VAR	RO	Number of revolutions	M	unsigned16
6503	**VAR**	**RO**	**Alarms**	**M MAP**	**unsigned16**
6504	VAR	RO	Supported alarms	M	unsigned16
6505	**VAR**	**RO**	**Warnings**	**M MAP**	**unsigned16**
6506	VAR	RO	Supported warnings	M	unsigned16
6507	VAR	RO	Profile and SW version	M	unsigned32
6508	VAR	RO	Operating time	M	unsigned32
6509	VAR	RO	Offset value (calculated)	M	signed32
650A	VAR	RO	Module Identification	M	signed32
650B	VAR	RO	Serial Number	M	unsigned32

VAR = Variable
ARRAY = Tableau de variables
RW = Lecture/écriture
RO = Lecture seule
const = Constante
Name = Nom de l'objet
M/O = Obligatoire ou optionnel
MAP = L'objet peut être mappé

16 Objets dans Details - Profil Codeur DS 306 V3.1

Objet 6000h Operating Parameters

Bit 0 : Séquence du code : 0 = croissant pour une rotation en sens horaire (cw)
1 = croissant pour une rotation en sens antihoraire (ccw)
Défaut : Bit = 0

Bit 2 : Fonction d'échelle : 0 = disable, 1 = enable; Standard : Bit = 0 (voir Objet 6001,6002)
Défaut : Bit = 1

Bit14 : Startup Mode: 0 = Bootup après Pre-Operational, 1 = Bootup après Operational
Défaut : Bit = 0

Bit15 : Event Mode: 0 = émission de la position selon TPDO 1800h, 1 = émission de chaque changement de position
Défaut : Bit = 0

Bit	Fonction	Bit = 0	Bit =1	C1	C2
0	**Séquence du code**	**CW**	**CCW**	**m***	**m***
1	Commissioning Diagnostic Control	Disabled	Enabled	o	o
2	**Activation du facteur d'échelle**	**Disabled**	**Enabled**	**o**	**m**
3	Measuring direction	Forward	Reward	o**	o**
4..11	Reserved for further use				
12	Manufacturer specific parameter	N.A.	N.A.	o	o
13	Manufacturer specific parameter	N.A.	N.A	o	o
14	**Lancement automatique en mode OP**	**Disabled**	**Enabled**	**o**	**o**
15	**Event Mode Position**	**Disabled**	**Enabled**	**o**	**o**

*m = la fonction doit être supportée
o = optionnel

Objet 6001h : Nombre de pas de mesure par tour (Résolution)

Ce paramètre permet le réglage de la résolution requise par tour. Le codeur calcule en interne le facteur d'échelle correspondant. Le facteur d'échelle calculé MURF (par lequel la valeur de la position physique sera multipliée) se calcule selon la formule suivante :

MURF = Nombre de pas de mesure par tour (6001h) / résolution physique monotour (6501h)

Contenu des données :

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^{23} \dots 2^{16}$	$2^{31} \dots 2^{24}$

Plage de valeurs : 1….résolution physique maximale (16384) 14 bits
Réglage par défaut : 16384 (14 bits)

Après une modification du facteur d'échelle, il faut impérativement régler aussi la valeur de prépositionnement à zéro ou à la valeur désirée.

Objet 6002h : Nombre total des pas de mesure

Ce paramètre permet le réglage du nombre total de pas de mesure. La résolution physique maximale est affectée d'un facteur. Ce facteur est toujours < 1. Après avoir atteint la position totale des pas de mesure, le codeur se remet à zéro.

Contenu des données :

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^{23} \dots 2^{16}$	$2^{31} \dots 2^{24}$

Plage de valeurs : 1….résolution physique maximale (16384) 14 bits
Réglage par défaut :16384 (14 bits)

Après une modification du nombre total, il faut impérativement régler aussi **la valeur de prépositionnement à zéro ou à la valeur désirée.**

Objet 6003h : Valeur de prépositionnement

La valeur de la position du codeur est réglée à cette valeur de prépositionnement.
Il est ainsi possible, par exemple, d'aligner la position zéro du codeur et la position zéro de la machine.

Contenu des données :

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^{23} \dots 2^{16}$	$2^{31} \dots 2^{24}$

Plage de valeurs : 0….résolution physique maximale (16384) 14 bits
Réglage par défaut :0

Après ne transmission de la valeur de prépositionnement, il faut effectuer une instruction de sauvegarde par l'objet 1010h pour mémoriser cette valeur de manière permanente. Dans le cas contraire, ce prépositionnement ne sera pris en compte que tant que le codeur restera sous tension.

Objet 6004h : Valeur de position

Le codeur émet la valeur de la position courante (le cas échéant affectée du facteur d'échelle).

Contenu des données :

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^{23} \dots 2^{16}$	$2^{31} \dots 2^{24}$

Plage de valeurs : 0….résolution physique maximale (16384) 14 bits
0 .. 3FFFh

Cet objet peut être mappé

Objet 6030h : Speed Value

Le codeur émet la vitesse calculée courante (le cas échéant affectée du facteur d'échelle) sous la forme d'une **valeur signée en 16 bits**. La vitesse est émise en **trs/min. (rpm)** ; elle est calculée à intervalles de synchronisation fixes de **60 ms**.
Le signe dépend du sens de rotation de l'axe. Les valeurs négatives indiquent une rotation à gauche (**CCW**), les valeurs positives une rotation à droite (**CW**).

Contenu des données :

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Plage de valeurs : 0....vitesse maximale 6500 tours/min

Pour les valeurs supérieures à 6500 tours/min., un message d'avertissement est émis et le bit d'avertissement « Dépassement de la vitesse de rotation bit 0 » de l'objet Warning 6505h est activé.

Cet objet peut être mappé

Objet 6200h : Cycle-Timer

Définit le temps du cycle selon lequel la position courante est émise via PDO 1 (voir objet 1800h). Cette sortie commandée par temporisateur est activée dès qu'un temps de cycle > 0 est défini.

Cet objet n'existe plus que pour des raisons de compatibilité avec les versions de profils plus anciennes. Utiliser l'Event Timer Subindex (05h) dans le PDO d'émission actuel à la place de cet objet.

Contenu des données :

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Plage de valeurs : 0 ... FFFFh (65535) donne le temps de cycle en millisecondes
Valeur standard = 0h

Objet 6500h : Affichage de l'Operating Status

Cet objet indique l'état des réglages programmés pour l'objet 6000h.

Contenu des données :

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Contenu des données : voir l'objet 6000h

Objet 6503h : Alarms

En plus des erreurs signalées par les messages d'urgence (emergency messages), l'objet 6503h dispose d'autres messages d'erreur. Le bit d'erreur correspondant est mis à 1 tant que l'erreur est présente.

Contenu des données :

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

N° de bit	Désignation	Valeur = 0	Valeur = 1
Bit 0	**Erreur de position**	Valeur de position valide	Erreur de position
Bit 1	**Vérification du hardware**	Pas de défaut	Défaut
Bit 2..15	**Non utilisés**		

Dans les deux cas, un message d'urgence **(ID=80h+numéro de nœud)** est envoyé en même temps que le code d'erreur **1000h (Generic error)** lors de l'apparition d'une alarme.

Cet objet peut être mappé

Objet 6504h : Supported Alarms

Cet objet indique les messages d'alarme supportés par le codeur (voir objet 6503h).

Contenu des données :

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Plage de valeurs voir objet 6503h
Un bit à 1 signifie que le message d'alarme est supporté
Exemple :
Bit 0 = 1 L'affichage du défaut de position est supporté

Objet 6505h : Warnings

Les messages d'avertissement signalent le dépassement des tolérances de paramètres internes du codeur. A l'opposé des messages d'alarme ou d'urgence, la valeur de la mesure peut être valide malgré tout lors de l'apparition d'un message d'avertissement. Le bit d'avertissement correspondant est mis à 1 tant que le dépassement de la tolérance ou l'avertissement est présent.

Contenu des données :

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

N° de bit	Désignation	Valeur = 0	Valeur = 1
Bit 0	**Dépassement de la vitesse de rotation**	Pas de dépassement	Dépassement
Bit 1	**Non utilisé**		
Bit 2	**Etat du Watchdog**	Système en ordre	Reset exécuté
Bit 3..15	**Non utilisés**		

Si le bit 0 est activé, un message Emergency (ID=80h+numéro de nœud) est émis en même temps que le **code d'erreur 4200h** (Device-specific).
Si le bit 2 ou le bit 3 est activé, un message Emergency (ID=80h+numéro de nœud) est émis en même temps que le **code d'erreur 5200h** (Device hardware).

Cet objet peut être mappé

Objet 6506h : Supported Warnings

Cet objet indique les messages d'avertissement supportés par le codeur (voir objet 6505h).

Contenu des données :

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Plage de valeurs voir objet 6505h
Un bit à 1 signifie que le message d'avertissement est supporté

Objet 6400h : Working Area State Register

Cet objet contient l'état actuel de la position du codeur par rapport aux limites programmées. Les drapeaux sont activés ou désactivés en fonction de la position des deux valeurs extrêmes. Cette comparaison avec les deux valeurs extrêmes s'effectue en « temps réel » et peut s'utiliser pour le positionnement en temps réel ou pour le déclenchement en fin de course.

Work_area_state							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
					Inférieur à LowLimit1	supérieur à HighLimit1	hors de la zone 1

Plage de valeurs 8 bits **Contenu des données voir bit 0...7**

Il faut vérifier les deux valeurs limites objet 6401h et objet 6402h afin que les signaux de sortie s'activent correctement !

Cet objet peut être mappé

Objet 6401h : Working Area Low Limit
Objet 6402h : Working Area High Limit

Ces deux paramètres permettent de définir la zone de travail. L'état, à l'intérieur ou à l'extérieur de cette zone, peut être signalé au moyen de bits drapeaux (**Objet 6400h Working Area State**). Ces marqueurs de zone peuvent également faire office de fins de course logiciels.

Contenu des données :

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^{23} \dots 2^{16}$	$2^{31} \dots 2^{24}$

Plage de valeurs : 1….résolution physique maximale (16384) 14 bits
Réglage par défaut :16384 (14 bits) **Working Area High Limit**
0 **Working Area Low Limit**

Objet 2100h :

Cet objet permet de modifier la vitesse de transmission par le logiciel. En standard, cette valeur est réglée à FFh, ce qui signifie que le réglage par hardware de la vitesse de transmission a la priorité. Si la valeur est réglée entre 1..9 et si ce paramètre est sauvegardé à l'aide de l'**objet 2105h Save All Bus Parameters**, l'appareil utilisera la vitesse de transmission modifiée au redémarrage ou au Reset Node suivant.

Byte 0
$2^7 \dots 2^0$

Contenu des données :

Plage de valeurs 1… 8
Réglage par défaut : 05h

Une nouvelle vitesse de transmission n'est prise en compte qu'au redémarrage suivant (Reset/Power-on) du codeur ou au moyen d'une instruction **NMT-Reset Node**

Objet 2101h : Adresse de nœud

Cet objet permet de modifier l'adresse de nœud par le logiciel. En standard, cette valeur est réglée à 0x3Fh. Si la valeur est réglée entre 1..127 et si ce paramètre est sauvegardé à l'aide de l'**objet 2105h Save All Bus Parameters**, l'appareil utilisera l'adresse de nœud modifiée au redémarrage ou au Reset Node suivant.

Contenu des données :

Byte 0
$2^7 \dots 2^0$

Plage de valeurs 1 ...127 ou 1..7Fh
Réglage par défaut : 0x3Fh

Le **numéro de nœud 0** est réservé et ne doit pas être utilisé pour définir un nœud.
Les numéros de nœud résultants se trouvent dans la plage **1...7Fh** hexadécimal ou (1...127).

Un nouveau numéro de nœud n'est pris en compte qu'au redémarrage suivant (Reset/Power-on) du codeur ou au moyen d'une instruction **NMT-Reset Node**

Objet 2102h : Terminaison de bus CAN désactivée/activée

Cet objet permet de d'activer la terminaison de bus par le logiciel. En standard, cette valeur est réglée à 1.

Contenu des données :

Byte 0
$2^7 \dots 2^0$

Plage de valeurs 0..1
Réglage par défaut : 1

Objet 2103h : Version de firmware Flash

Cet objet indique la version actuelle du firmware sous la forme d'une valeur hexadécimale en 16 bits. Cette valeur permet de vérifier le niveau courant de l'appareil.

Contenu des données :

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Plage de valeurs jusqu'à FFFFh
Exemple : **4FA6h Firmware actuel**

Objet 2105h : Save All Bus Parameters

Ce paramètre sauvegarde les paramètres désirés du bus (objet 2100h, 2101h, 2102h) de manière permanente dans la mémoire Flash. Cet objet apporte une sécurité supplémentaire contre des modifications involontaires de la vitesse de transmission et de l'adresse de nœud.
Seule une sauvegarde ciblée à l'aide du paramètre **« save » (hexadécimal 0x65766173)** permet une sauvegarde permanente des paramètres du bus : **vitesse de transmission, numéro de nœud et terminaison**.

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^{23} \dots 2^{16}$	$2^{31} \dots 2^{24}$

Plage de valeurs : **« save » en hexadécimal 0x65**

Objet 2140h : Customer Memory (16 Bytes)

Ces 4 paramètres constituent une zone d'enregistrement pour l'utilisateur. Il est possible d'enregistrer **4 mots de données de 4 octets maximum**. Le format de cette zone n'est pas vérifié, il est possible d'enregistrer n'importe quel format.

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^{23} \dots 2^{16}$	$2^{31} \dots 2^{24}$

Plage de valeurs : Chiffres, alphanumérique
Réglage par défaut :0

Objet 1029h : Error Behavior

Lorsqu'une erreur grave est détectée, l'appareil doit passer automatiquement dans le mode **Pre-Operational**. Cet objet permet de déterminer le comportement de l'appareil lors de l'apparition d'une erreur. Les classes d'erreurs suivantes sont couvertes :

1029h,sous-index 1 Erreurs de communication **Défaut = 0**

- Etat Bus off de l'interface CAN
- Apparition d'un événement Life guarding
- Echec de la surveillance du Heartbeat

1029h,sous-index 2 Device Profile Specific **Défaut = 1**

- Erreurs du capteur et du contrôleur
- Erreurs de température

1029h,sous-index 3 Manufacturer Specific **Défaut = 1**

- Erreur interne

La valeur des classes d'erreurs se décompose comme suit :

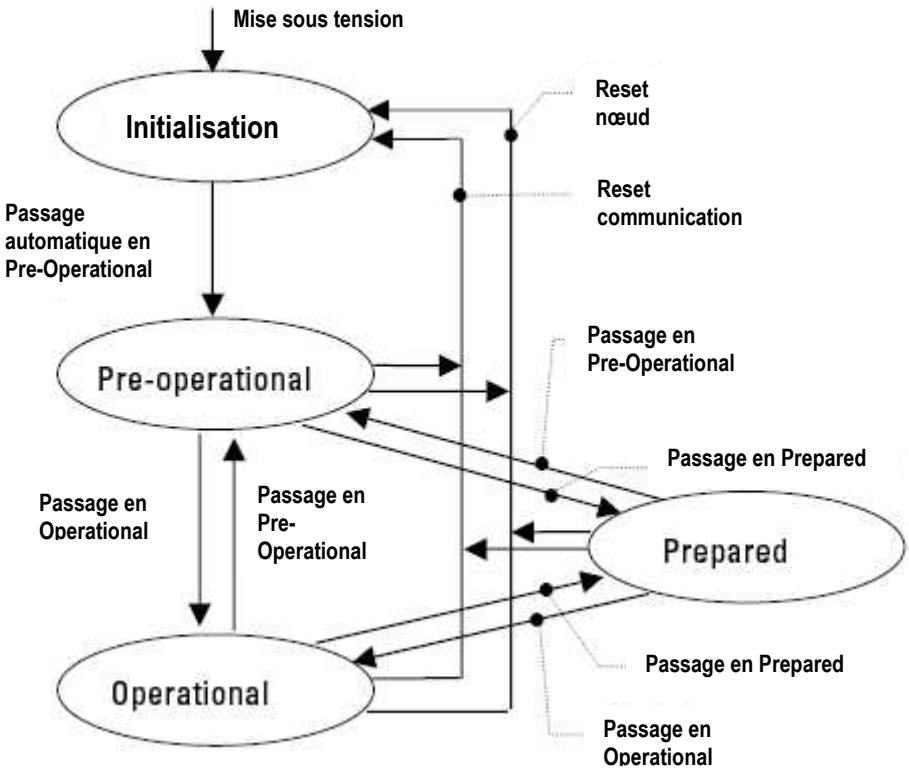
Byte 0
$2^7 \dots 2^0$

Plage de valeurs 8 bits

- **0 Mode Pre-Operational (uniquement si le mode Operational était actif auparavant)**
- **1 Pas de changement de mode**
- **2 Mode Stopped**
- **3 .. 127 réservés**

17 Gestion du réseau

Le codeur supporte la gestion de réseau simplifiée (minimum boot-up) définie dans le profil pour les « minimum capability devices ».
Le diagramme d'état selon DS 301 suivant représente les différents états des nœuds et les instructions réseau correspondantes (commandées par le maître du réseau à l'aide de services NMT).



Initialisation : Etat initial de l'appareil après sa mise sous tension, après un Reset ou après la mise en service de l'appareil. Après l'exécution des routines de Reset/d'initialisation, le nœud passe automatiquement dans l'état Pre-operational. Les LED indiquent l'état instantané.

Pre-operational: Il est maintenant possible de s'adresser au nœud CAN par des messages SDO ou des instructions NMT à l'aide de l'identifiant standard. La programmation des paramètres du codeur ou de la communication a lieu maintenant.

Operational: Le nœud est actif. Les valeurs du process sont émises par l'intermédiaire des PDO. Toutes les instructions NMT peuvent être exploitées.

Prepared ou **Stopped:** Dans cet état, le nœud n'est plus actif : ni la communication SDO ni la communication PDO ne sont plus possibles. Le nœud peut être mis dans l'état Operational ou Pre-Operational à l'aide d'instructions NMT.

18 Instructions NMT

Toutes les instructions NMT sont transmises sous la forme d'objets NMT non confirmés. Grâce au modèle de communication Broadcast (diffusé sur tout le réseau), les instructions NMT sont reconnues par tous les participants.
Un objet NMT est structuré de la manière suivante :

ID COB = 0

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Octet 0 = octet de l'instruction
Octet 1 = numéro de nœud

L'ID COB de l'objet NMT est toujours 0
Le nœud destinataire est défini par le numéro de nœud. Si le numéro de nœud est 0, l'instruction s'adresse à tous les nœuds.

Octet d'instruction (hex)	Description
01h	Start_Remote_Node : passage en Operational
02h	Sop_Remote_Node : passage en Prepared
80h	Enter_Pre-Operational_State : passage en Pre-Operational
81h	Reset_Node : reset du noeud[1]
82h	Reset_Communication : reset de la communication[2]

[1] Tous les paramètres du répertoire d'objet entier sont mis aux valeurs qu'ils ont à la mise en service.
[2] Seuls les paramètres de la section Profil de communication du répertoire d'objets sont mis aux valeurs qu'ils ont à la mise en service.

19 Signalisation par la LED pendant le fonctionnement

LED verte = état du BUS

LED rouge = indication ERR

Indication	LED	Signification	Cause de l'erreur	Indications complémentaires
Bus éteinte		**Pas de communication avec le maître [2]**	**Coupure de la ligne de données vitesse de transmission erronée Ligne de données inversée**	**Observer la combinaison avec la LED ERR Si la LED ERR est aussi éteinte, vérifier la tension d'alimentation [3]**
Bus clignotante env. 250 ms		**Communication avec le maître Etat Pre-Operational**		**Communication SDO**
Bus clignotante env. 1 sec.		**Communication avec le maître Etat Stopped**		**Communication SDO impossible Uniquement instructions NMT**
Bus allumée		**Communication avec le maître Etat Operational**		**Transfert PDO actif**
ERR éteinte		**L'appareil fonctionne sans erreur.**		**Observer la combinaison avec la LED BUS**
ERR clignotante		**Communication avec le maître interrompue**	**Combinaison avec l'état du BUS**	**LED BUD verte clignotante ou allumée - en fonction de l'objet 1029h Error Behaviour**
ERR allumée		**Etat BUS OFF**	**Court-circuit sur le bus ou vitesse de transmission erronée**	
Bus+ERR clignotantes		**Mode Global LSS activé**	**L'appareil attend une instruction LSS**	**Il est possible de quitter le mode LSS à l'aide d'une instruction globale**

.
Les différents messages des LED peuvent également apparaître en combinaison les unes avec les autres.

[2] Le maître peut être un API ou un second partenaire de communication
[3] Tension de fonctionnement

20 Définitions

Explication des symboles :

Ce symbole accompagne des parties de texte à prendre en compte de manière particulière afin de garantir le bon fonctionnement de l'appareil et d'exclure tout risque.
Ce symbole accompagne des indications importantes pour la bonne utilisation du codeur. Le non-respect de ces indications peut entraîner des défauts du codeur ou de son environnement.

Ce symbole indique une particularité.

Réglage d'usine par défaut des paramètres

21 Abréviations utilisées

CAL	CAN Application Layer. Couche d'application (couche 7) dans le modèle de communication CAN
CAN	Controller Area Network
CiA	CAN in Automation. Association internationale des utilisateurs et constructeurs de produits CAN
CMS	CAN Message Specification. Elément de service de CAL
COB	Communication Object. Unité de transport dans le réseau CAN (message CAN). Les données sont envoyées sur le réseau à l'intérieur d'un COB.
ID COB	Identifiant COB. Désignation univoque d'un message CAN. Cet identifiant détermina la priorité du COB dans le réseau.
DBT	Distributor. Elément de service de CAL, responsable de l'attribution dynamique d'identifiants.
DS	Draft Standard ; Projet de norme
DSP	Draft Standard Proposal ; Proposition de projet de norme
ID	Identifiant, voir ID COB
LMT	Layer Management. Elément de service de CAL, responsable de la configuration des paramètres dans les différentes couches du modèle de communication
LSB	Least Significant Bit/Byte ; bit/octet de poids le plus faible
MSB	Most Significant Bit/Byte ; bit/octet de poids le plus fort
NMT	Network Management. Elément de service de CAL, responsable de l'initialisation, de la configuration et de la gestion des erreurs dans le réseau.
OSI	Open Systems Interconnection. Modèle de couches pour la description des zones fonctionnelles dans un système de communication de données.
PDO	Process Data Object. Objet pour l'échange de données process.
RTR	Remote Transmission Request ; télégramme de requête de données
SDO	Service Data Object ; objet de communication permettant au maître d'accéder au répertoire d'objets d'un nœud.
SYNC	Télégramme de synchronisation. Les participants au bus répondent à l'instruction SYNC en retournant la valeur de leur process.

22 Tableau de conversion décimal-hexadécimal

Lors de l'utilisation de chiffres, les valeurs décimales sont fournies sous la forme de chiffres sans extension (p. ex. 1408), les valeurs binaires sont suivies de l'extension b (p. ex. 1101b) et les valeurs hexadécimales de l'extension h (p. ex. 680h).

Dez	Hex	Dez	Hex	Dez	Hex	Dez	Hex
0	**00**	32	**20**	64	**40**	96	**60**
1	**01**	33	**21**	65	**41**	97	**61**
2	**02**	34	**22**	66	**42**	98	**62**
3	**03**	35	**23**	67	**43**	99	**63**
4	**04**	36	**24**	68	**44**	100	**64**
5	**05**	37	**25**	69	**45**	101	**65**
6	**06**	38	**26**	70	**46**	102	**66**
7	**07**	39	**27**	71	**47**	103	**67**
8	**08**	40	**28**	72	**48**	104	**68**
9	**09**	41	**29**	73	**49**	105	**69**
10	**0A**	42	**2A**	74	**4A**	106	**6A**
11	**0B**	43	**2B**	75	**4B**	107	**6B**
12	**0C**	44	**2C**	76	**4C**	108	**6C**
13	**0D**	45	**2D**	77	**4D**	109	**6D**
14	**0E**	46	**2E**	78	**4E**	110	**6E**
15	**0F**	47	**2F**	79	**4F**	111	**6F**
16	**10**	48	**30**	80	**50**	112	**70**
17	**11**	49	**31**	81	**51**	113	**71**
18	**12**	50	**32**	82	**52**	114	**72**
19	**13**	51	**33**	83	**53**	115	**73**
20	**14**	52	**34**	84	**54**	116	**74**
21	**15**	53	**35**	85	**55**	117	**75**
22	**16**	54	**36**	86	**56**	118	**76**
23	**17**	55	**37**	87	**57**	119	**77**
24	**18**	56	**38**	88	**58**	120	**78**
25	**19**	57	**39**	89	**59**	121	**79**
26	**1A**	58	**3A**	90	**5A**	122	**7A**
27	**1B**	59	**3B**	91	**5B**	123	**7B**
28	**1C**	60	**3C**	92	**5C**	124	**7C**
29	**1D**	61	**3D**	93	**5D**	125	**7D**
30	**1E**	62	**3E**	94	**5E**	126	**7E**
31	**1F**	63	**3F**	95	**5F**	127	**7F**

23 Glossaire

Vitesse de transmission
Vitesse de transmission. Elle est en relation avec la synchronisation nominale des bits. La vitesse de transmission maximale dépend de nombreux facteurs qui affectent la durée du parcours des signaux sur le bus. Il existe une relation essentielle entre la vitesse de transmission maximale, la longueur du bus et le type de câble. Différentes vitesses de transmission, entre 10 kbit/s et 1 Mbit/s, sont définis dans CANopen.

CANopen
CANopen est un protocole basé sur CAN développé à l'origine pour des systèmes de commande industriels. Ses spécifications englobent aussi bien divers profils d'appareils que le cadre pour des applications spécifiques. Les réseaux CANopen s'utilisent également sur des véhicules tout-terrain, dans l'électronique navale, dans les appareils médicaux et les trains. Sa couche d'applications très souple d'utilisation et les nombreuses caractéristiques optionnelles sont idéales pour la mise en œuvre de solutions sur mesure. Il existe en outre de nombreux outils de configuration. L'utilisateur peut ainsi définir des profils d'appareil spécifiques pour ses applications. Vous trouverez davantage d'informations sur CANopen dans l'Internet, à l'adresse **www.can-cia.org**.

Fichier EDS
Le ficher EDS (Electronic Data Sheet – Fiche technique électronique) est fourni par le constructeur d'un appareil CANopen. Il a un format standardisé pour la description des appareils. Le ficher EDS contient des informations sur :

- Description du fichier (nom, version, date de création, etc.)
- Informations générales sur l'appareil (nom et code du constructeur)
- Nom et type de l'appareil, version, adresse LMT
- Vitesses de transmission supportées et capacité au démarrage
- Description des objets supportés et de leurs attributs

Numéro de nœud
Dans un réseau CANopen, chaque appareil est identifié par son numéro de nœud (Node-ID). Les numéros de nœud autorisés se trouvent dans la plage de 1-127 et ne peuvent apparaître qu'une fois dans un réseau.

Gestion du réseau
Différentes tâches en rapport avec la configuration, l'initialisation et la surveillance des participants au réseau sont à réaliser dans un système distribué. L'élément de service « Gestion de réseau (NMT) »défini dans CANopen fournit cette fonctionnalité.

PDO
Les objets Données process (PDO) constituent les moyens de transport pour la transmission de données process (objets applicatifs). Un PDO est émis par un Producer (émetteur) et peur être reçu par un ou plusieurs Consumers (récepteurs).

Mappage PDO
Un PDO peut avoir une taille jusqu'à 8 octets. Il peut servir à transporter plusieurs objets applicatifs. Le mappage PDO décrit la définition et la disposition des objets applicatifs à l'intérieur du champ de données du PDO.

SDO
Les objets Données service (Service Data Objects, SDO) assurent le transfert confirmé de données de longueur quelconque entre deux participants au réseau. Le transfert des données s'effectue en mode Client-Serveur.