

HYDAC ELECTRONIC

Functional Safety
PL d
SIL 2

CANopen
safety easy to use

Description du
protocole
CANopen Safety

HAT 1000 / HAT 3000

Capteur angulaire
Simple tour

(Traduction de l'original)



Table des matières

1	Introduction	6
2	Fonctions de l'HAT CANopen Safety	6
3	Les vitesses de communication	7
4	CAN-Frames	7
5	Node-ID	7
6	Service de transfert	7
6.1	Service Data Object (SDO)	7
6.2	Process Data Object (PDO)	8
6.3	Synchronisation Object (SYNC)	9
6.4	Safety Relevant Data Object (SRDO)	9
6.5	Emergency Object (EMCY)	11
6.6	Heartbeat	11
6.7	Network Management Services (NMT)	12
6.8	Boot Up Protocol	12
7	Flux de données dans le HAT CANopen Safety	13
7.1	Sensor Unit	13
7.2	Calibration & Scaling	13
7.3	Transmission Unit	13
8	L'Object Dictionary	14
8.1	Structure de l'Object Dictionary	14
8.2	Structure de la pièce spécifique d'après DS406	14
9	Entrées dans l'Object Dictionary	15
9.1	Communication Profile Specific Entries (DS301)	15
9.1.1	Index 1000h : DeviceType (read only)	15
9.1.2	Index 1001h : ErrorRegister (read only)	15
9.1.3	Index 1002h: Manufacturer status register (read only)	15
9.1.4	Index 1003h: Pre-defined error field (read only)	15
9.1.5	Index 1005h: SyncMessageIdentifier (read write)	15
9.1.6	Index 1008h : ManufacturerDeviceName (const)	15
9.1.7	Index 1009h : ManufacturerHardwareVersion (const)	15
9.1.8	Index 100Ah : ManufacturerSoftwareVersion (const)	15
9.1.9	Index 1010h : StoreParameters	15
9.1.10	Index 1011h: RestoreDefaultParameters	16
9.1.11	Index 1014h: CobIdEmergencyMessage (read write)	16
9.1.12	Index 1017h: ProducerHeartbeatTime (read write)	16
9.1.13	Index 1018h: IdentityObject	17
9.1.14	Index 1029h: Error behaviour	17
9.1.15	Index 1301h: SRDO communication parameter	17

9.1.16 Index 1381h: SRDO mapping parameter	17
9.1.17 Index 13FEh: Configuration Valid (read write).....	18
9.1.18 Index 13FFh : Safety Configuration Checksum.....	18
9.1.19 Index 1800h : TPDO communication parameter	18
9.1.20 Index 1A00h: TPDO mapping parameter	19
9.1.21 Index 1F80h: NMT-Startup (read / write).....	20
9.2 Device Profile Specific Entries (DS406).....	20
9.2.1 Index 6000h: Operating Parameter (read only).....	20
9.2.2 Index 6001h: Measuring units per revolution.....	20
9.2.3 Index 6002h : Total measuring range in measuring units.....	20
9.2.4 Index 6003h : Preset value (read write)	20
9.2.5 Index 6004h : Position value (read only).....	20
9.2.6 Index 6500h : Operating Status (read only).....	20
9.2.7 Index 6501h : Single-turn resolution and Measuring step.....	20
9.2.8 Index 6503h : Alarms (read only).....	21
9.2.9 Index 6504h : Supported alarms (read only).....	21
9.2.10 Index 6505h : Warnings (read only).....	21
9.2.11 Index 6506h : Supported warnings (read only).....	21
9.2.12 Index 6507h : Profile and software version (read only).....	21
9.2.13 Index 6508h : Operating time (read only).....	21
9.2.14 Index 6509h : Offset value (read only).....	21
9.2.15 Index 650Ah : Module identification (read only).....	21
9.3 Manufacturer Specific Entries.....	21
9.3.1 Index 2001h : Node-ID.....	21
9.3.2 Index 2002h : Baudrate.....	21
9.3.3 Index 4006h: Position Value (read only) Position invertiert	22
9.3.4 Indices supplémentaires dans la plage 2000h à 5FFFh (réservé).....	22
10 Layer setting services (LSS) et protocoles	23
10.1 Finite state automaton, FSA.....	24
10.2 Transmission de services LSS.....	25
10.2.1 Format de message LSS	25
10.3 Protocoles Switch mode	26
10.3.1 Protocole Switch mode selective	26
10.3.2 Switch mode selective Protokoll.....	26
10.1 Protocoles Configuration.....	27
10.1.1 Configurer protocole Node-ID	27
10.1.2 Configurer protocole bit timing parameters	27
10.1.3 Protocole Activate bit timing parameters.....	28
10.1.4 Protocole Store configuration.....	29
10.2 Protocoles Inquire LSS-Address.....	29
10.2.1 Protocole Inquire Identity Vendor-ID	29
10.2.2 Protocole Inquire Product-Code.....	30
10.2.3 Protocole Inquire Identity Revision-Number	30
10.2.4 Protocole Inquire Identity Serial-Number	31
10.2.5 Protocole Inquire Node-ID	31
10.3 Protocoles Identification.....	32
10.3.1 Protocole identify LSS remote slave	32
10.3.2 Protocole LSS identify slave	33

10.3.3 Protocole LSS identify non-configured remote slave	33
10.3.4 Protocole LSS identify non-configured slave	33
11 Raccordement	34
11.1 Commuter la tension d'alimentation	34
11.2 Réglages des Node-ID et de la vitesse de communication avec les services LSS	34
11.2.1 Configuration du Node-ID, procédure	34
11.2.2 Configuration de la vitesse de communication, procédure	35
12 Mise en service	36
12.1 Interface CAN	36
12.2 Fichier EDS	36

Avant-propos

A l'intention des utilisateurs de notre produit, nous avons regroupé dans cette notice, les principales informations pour l'utilisation et la maintenance de l'appareil.

Cette notice a pour objectif de vous familiariser avec le produit et d'optimiser son utilisation.

Ce document doit toujours être disponible sur le lieu d'utilisation. Veuillez à ce que les indications de la technique de programmation de cette notice correspondent à la date de création.

Si, lors de la lecture de cette documentation, vous deviez détecter des erreurs ou encore si vous aviez des suggestions ou des remarques, veuillez vous adresser à :

HYDAC ELECTRONIC GMBH
Documentation technique
Hauptstrasse 27
66128 Sarrebruck
Allemagne
Tél. : +49 (0)6897 / 509-01
Fax: +49 (0)6897 / 509-1726
E-mail : electronic@hydac.com

Nous vous remercions pour votre contribution.

« De la pratique vers la pratique »

1 Introduction

Les capteurs angulaires de la famille HAT CANopen Safety correspondent au standard CANopen selon les profils et standards suivants :

- [1] CiA DS301, Version: 4.2.0 (21 Février 2011)
CANopen application layer and communication profile
- [2] CiA DS302, Version: 4.1 (02 Février 2009)
Additional application layer functions - Part 2: Network management
- [3] CiA DS303, Version: 1.4 (14 Août 2006)
Additional specification - Part 2: Representation of SI units and prefix
- [4] CiA DS 304, Version V1.0.1 (01 Janvier 2005)
Framework for safety-relevant communication
- [5] CiA DS305, Version: 2.2 (26 Août 2008)
Layer setting services (LSS) and protocols
- [6] CiA DS406, Version 3.2 (18 Décembre 2006)
Device profile for encoders

Cette notice décrit les fonctions prises en charge par le HAT CANopen Safety. Pour cela, des notions en CAN et CANopen Safety sont nécessaires. Le fonctionnement exact est décrit dans *les documents mentionnés précédemment*. Etant donné que les deux spécifications sont rédigées en anglais, les termes de cette notice sont conservés en anglais et *en italique* pour une meilleure affectation.

2 Fonctions de l'HAT CANopen Safety

- **Détermination de la mesure angulaire (simple tour) de l'arbre par rapport au boîtier (absolu)**
- Transfert de la valeur actuelle de la mesure angulaire sous forme de **PDO** ou **SRDO** dans les événements suivants :
 - **Synchronisation** par réception d'un objet SYNC (PDO seulement)
 - Cycle **asynchrone** dans une plage d'1 milliseconde à >1 minute

3 Les vitesses de communication

L'HAT CANopen Safety prend en charge les vitesses de communication suivantes (baudrate) :

- 1000 kbit/s
- 800 kbit/s
- 500 kbit/s
- 250 kbit/s
- 125 kbit/s
- 50 kbit/s
- 20 kbit/s
- 10 kbit/s

Le timing correspond à DS301, *Bit rates and timing*.

La vitesse de communication utilisée est mémorisée dans une mémoire non volatile. Celle-ci est pré-réglée à **500 kbits/s** en sortie d'usine et peut être configurée par le biais du bus CAN (voir *Object Dictionary Index 2002h*).

4 CAN-Frames

L'HAT CANopen Safety prend en charge (d'après les spécifications requises) les frames standard 11 bits avec identifiant 11 bit et les frames 29 bit avec identifiant 29 bit.

5 Node-ID

Pour la mise en service, l'HAT CANopen Safety doit avoir un unique "*Node-ID*" réglé sur le réseau CANopen.

Le réglage *Node-ID* est, comme la vitesse de communication, mémorisé dans la mémoire non volatile et peut être modifié via bus CAN (voir *Object Dictionary Index 2001h*) ou à l'aide du LSS [5].

L'adresse par défaut est réglée à 1.

6 Service de transfert

6.1 Service Data Object (SDO)

Avec CANopen toutes les données d'un appareil (paramètres de réglage et données de mesure) sont classées dans un *Object Dictionary* sous un index défini. Certaines entrées du *Object Dictionary* sont en plus structurées dans un *Subindex*. Avec les SDO d'autres utilisateurs peuvent lire ou définir l'*Object Dictionary* du HAT CANopen Safety.

Le HAT CANopen Safety adopte une fonction de *Server*, l'appareil qui souhaite lire ou définir les données adopte une fonction de *Client*.

Pour le transfert de données, le HAT CANopen Safety doit être équipé d'un *Receive SDO* pour la réception de données et d'un *Transmit SDO* pour la transmission de données.
Déroulement du transfert de données :

Lecture de l'Object dictionary :

1. Un appareil (*Client*) envoie un *Receive-SDO* de l'HAT CANopen Safety (*Server*). Dans ce *SDO* se trouve l'indication que l'*Object Dictionary* doit être lu, ainsi que l'*Index* et le *Subindex* souhaité.
2. Le HAT CANopen Safety (*Server*) transmet son *Transmit-SDO*. Celui-ci contient également l'*Index* et le *Subindex*, ainsi que les données lues.

Description de l'Object Dictionary :

1. Un appareil (*Client*) envoie un *Receive-SDO* de l'HAT CANopen Safety (*Server*). Dans ce *SDO* se trouve l'indication que l'*Object Dictionary* doit être décrit, ainsi que l'*Index* souhaité, le *Subindex* et les données devant être écrites.
2. Le HAT CANopen Safety (*Server*) transmet son *Transmit-SDO*. Dans celui-ci se trouve également l'*Index* et le *Subindex*, ainsi que l'indication que l'*Object Dictionary* a été décrit.

Si une erreur se produit, par exemple si l'*Index* indiqué n'existe pas ou si l'on essaie de décrire une entrée *read only* ou si les données ne sont pas dans la plage de validité, alors le *Transmit-SDO* contient un message *Abort SDO Transfer* correspondant ainsi qu'un *Abort Code* concordant (voir [1])

Les *COB-ID* respectifs du *SDO* correspondent au *Pre defined Connection Set* défini dans le DS301 et ne peuvent être modifiés

COB-IDs pour Service Data Objects

SDO	COB-ID
<i>Receive – SDO</i>	600h+Node-ID
<i>Transmit – SDO</i>	580h+Node-ID

6.2 Process Data Object (PDO)

La transmission de données via *SDO* est flexible, mais présente quelques inconvénients pour la transmission de valeurs de mesures ou de paramétrage: Il n'est possible de lire qu'une date, les données doivent d'abord être demandées via un *SDO* et les *Index* et *Subindex* sont transmis à chaque fois, ce qui a pour conséquence l'augmentation de l'*Overhead*.

Pour cela le CANopen définit des *Process Data Objects*. Ceux-ci contiennent uniquement les données nécessaires. Il existe deux types de *PDO*

1. *Transmit-PDO*
Avec cela un appareil de mesure peut envoyer ses valeurs de mesure.
2. *Receive-PDO*
Permet de transmettre une valeur de paramétrage à un organe ou une unité de régulation.

Le *PDO-Mapping* définit quelles données vont être situées dans un *PDO*. Ce *PDO-Mapping* est enregistré dans le *Object Dictionary* (voir *Object Dictionary, Index 1A00*)

Dans le *PDO-Transmission Type* se trouve l'*ID* et par quel évènement un *PDO* est transféré. Ces paramètres se trouvent également dans l'*Object Dictionary* (voir *Object Dictionary, Index 1800h*).

Événements déclenchant l'envoi d'un PDO

1. Réception d'un objet SYNC (transfert synchrone)
2. Ecoulement d'un temps de cycle réglable de 1 milliseconde à >1 minute (transfert cyclique)

Le HAT CANopen Safety implique un *Transmit-SDO* qui contient les valeurs actuels du capteur ou du processus.

Le DS406 est un standard qui prévoit le transfert de la valeur de mesure actuelle en valeur 16-bit, ainsi que l'état en valeur 32-bit.

Le HAT CANopen Safety n'implique que le transfert sous forme de valeur de 32-bit.

6.3 Synchronisation Object (SYNC)

Les objets *SYNC* servent à la réalisation d'un transfert de données synchrone. Un objet *SYNC* est, en principe, un message CAN avec un Identifieur défini, sans les données. CANopen fait la distinction entre *SYNC Producer* et *SYNC Consumers*. *SYNC Producer* sont des périphériques sur le bus qui envoient le *SYNC* à intervalle de temps paramétrable. *SYNC Consumers* sont des périphériques qui réagissent à la réception du *SYNC*. Dans un réseau CANopen, plusieurs objets *SYNC* peuvent exister. Les différents objets *SYNC* sont différenciés à l'aide de leur *SYNC-ID* qui correspondent aux *CAN Identifier* utilisés. Le *SYNC-ID* utilisé est déposé dans *Object Dictionary*.

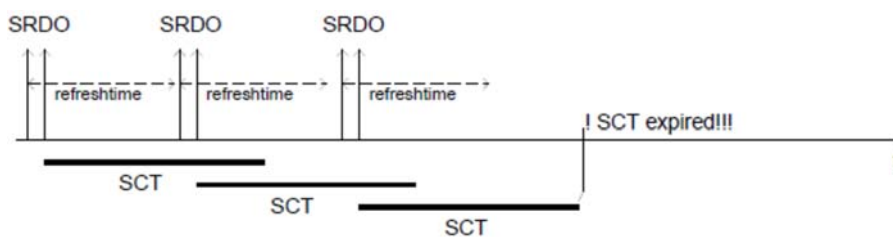
Le HAT CANopen Safety offre la fonctionnalité d'un *SYNC Consumer*. Avec le paramétrage correspondant du *PDO Transmission Type*, un *PDO* est envoyé après réception du *SYNC*. Le *SYNC-ID* est réglé sur 80h et peut être modifié dans *Object Dictionary* (voir *Object Dictionary, Index 1005h*). Sous *PDO Transmission Type* peut être introduit le comptage des objets *SYNC* reçus qui ont mené à l'envoi d'un *PDO*.

6.4 Safety Relevant Data Object (SRDO)

La transmission de données via ce que l'on appelle les *Safety Relevant Data Objects* (les données pertinentes à la sécurité) assure un transfert sécurisé des valeurs mesurées. Les *SRDO* peuvent être utilisés pour envoyer les données ou bien être désactivés.

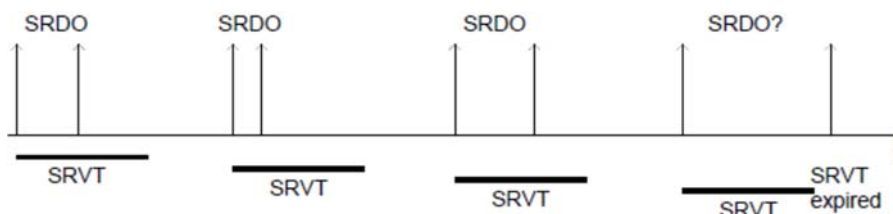
Un *SRDO* est composé de deux trames CAN indépendantes. La première trame contient les données sous forme de représentation normale, le deuxième contient les données inversées. Cela permet au destinataire de vérifier les données.

La durée entre deux *SRDO* est désignée par *Refresh Time* ou *Safeguard cycle time (SCT)*. Elle sert à surveiller le Timeout entre deux *SRDO*.



Source : CiA Draft Standard 304, CANopen Framework for safety-relevant communication, Version 1.0.1, Janvier 2005, CAN in Automation (CiA), p. 9

La durée entre les deux trames est désignée par *Safety-relevant object validation time* (SRVT) et permet au destinataire la détection du Timeout au sein d'un SRDO.



Source: CiA Draft Standard 304, CANopen Framework for safety-relevant communication, Version 1.0.1, Janvier 2005, CAN in Automation (CiA), p.10

Le *SRDO-Mapping* définit quelles données se trouvent dans un SRDO. Ce mapping est enregistré dans l'*Object Dictionary*.

Le microprogramme prend en charge les *Transmit SRDO* mais ne prend pas en charge les *Receive SRDO*.

Lors de la configuration, respecter les points suivants :

- La modification des paramètres de communication n'est possible qu'en état *Pre-Operational*.
- Pour la modification des COB-ID, le SRDO correspondant doit d'abord être désactivé (*Information Direction* est 0).
- Le SCT sur le nœud du destinataire doit être choisi plus grand par rapport au nœud de l'émetteur, pour éviter des problèmes de synchronisation à cause des durées de marche du logiciel ($\pm 2,8$ ms).
- Le SRVT doit être choisi de manière à ce qu'une interruption du SRDO sur le bus CAN ne soit pas suivie par une erreur chez le destinataire causée par des messages d'une qualité supérieure.
- Les SCT et SRVT sont aussi fortement dépendants du débit binaire.
- Le total de contrôle après modification doit être recalculé et réenregistré dans l'*Object Dictionary* (voir *Index 13FFh*) pour chaque SRDO concerné.
- Après modification de la configuration, il doit être réinitialisé et vérifié.
- Après la vérification des données, la configuration doit être déclarée valide (voir *Object Dictionary, Index 13FE, Configuration Valid*).

Lors du contrôle de plausibilité de la configuration, respecter les points suivants :

- Le SRVT doit être inférieur au SCT.
- Le COB-ID du message avec les données normales doit toujours être impair.
- Le COB-ID du message avec les données inverses doit toujours être pair.

Le constructeur de l'installation est responsable de l'évidence des paramètres réglés (voir COB-ID partie DS304, voir [4]).

Un contrôle de plausibilité est effectué lorsque la valeur A5h est écrite sur Configuration Valid. L'écriture n'est effectuée avec succès que si le contrôle a été réalisé avec succès. Le contrôle de plausibilité est effectué pour tous les SRDOs, qui sont activés (*Information Direction* différent de 0).



ATTENTION :
Les SRDO ne sont envoyés que si leur configuration est valide.

6.5 Emergency Object (EMCY)

L'objet *EMCY* est envoyé lorsqu'une erreur apparaît. Les objets *EMCY* contiennent un *Emergency Error Code*, le contenu de l'*Error Register* ainsi qu'un *Manufacturer specific Error Field*. Si une erreur déclarée est éliminée ou a disparu, ceci est également signalé via un objet *EMCY* spécifique.

Un message Emergency est envoyé si une erreur survient ou disparaît. Le message est composé de la manière suivante :

Erreur	Emergency ErrorCode	Manufacturer SpecificErrorField	Catégorie d'erreur
Aucune erreur	0000h	0000h	Erreur éliminée
Une erreur est survenue lors du chargement du setup de l'utilisateur	FF00h	0001h	Erreur spécifique du constructeur (Bit 7)
Erreur de contrôleur	FF00h	0002h	Erreur spécifique du constructeur (Bit 7)
CAN Error Passive	8120h	0003h	Erreur communication (Bit 4)
Recovered from Bus-off	8140h	0004h	Erreur communication (Bit 4)

L'objet *EMCY* a un ID pré-réglé $80h + \text{Node-ID}$ et peut être modifié dans l'*Object Dictionary* (voir *Object Dictionary, Index 1014h*).

6.6 Heartbeat

Avec le protocole *Heartbeat*, une surveillance des différents périphériques peut être effectuée. CANopen fait la distinction entre les fonctions suivantes:

1. *Heartbeat Producer*
Envoie dans un intervalle (cycle) un objet *Heartbeat*.
2. *Heartbeat Consumer*
Contrôle l'envoi d'objets *Heartbeat*.

Le cycle est réglable en millisecondes dans le *Object Dictionary*. L'indication temporelle 0 signifie "*Heartbeat* non actif".

Le statut de *Heartbeat Producers* est toujours transmis par le biais de l'objet *Heartbeat* sous forme d'un byte.

Signification du contenu de l'objet Heartbeat

Valeur	Etat	Remarque
0	<i>BOOTUP</i>	L'appareil a démarré.
4	<i>STOPPED</i>	L'appareil est arrêté.
5	<i>OPERATIONAL</i>	L'appareil fonctionne normalement.
127	<i>PRE-OPERATIONAL</i>	L'appareil n'envoie pas de <i>PDO</i> mais peut traiter des <i>SDO</i> .

Le HAT CANopen Safety peut fonctionner en tant que *Heartbeat Producer*. L'ID du *Heartbeat* est $700h + Node-ID$. Le temps est pré-réglé à 0 (non actif) et peut être modifié (voir *Object Dictionary, Index 1017h*).

6.7 Network Management Services (NMT)

L'objet *NMT* sert à démarrer, arrêter et redémarrer des appareils. CANopen différencie les fonctionnalités suivantes:

1. *NMT Master*
commande les autres nœuds.
2. *NMT Slave*
est commandé par un *Master*.

Dans un réseau CANopen, il n'y a qu'un objet *NMT* avec un Identifiant 0. Il y a toujours 2 bytes transférés. Le premier byte contient le *Command Specifier* qui représente la commande, le deuxième byte étant le *Node-ID* du nœud qui doit exécuter la commande. Une valeur à 0 signifie que la commande est valable pour tous les nœuds. Les commandes suivantes sont possibles:

Commandes NMT

1. **Start Remote Node**
L'état du nœud passe en *Operational*.
2. **Stop Remote Node**
L'état du nœud passe en *Stopped*.
3. **Enter Pre-Operational**
L'état du nœud passe en *Pre-Operational*
4. **Reset Node**
La plage OD "Device Profile Specific" est remise à zéro, la *baudrate* est réinitialisée si nécessaire et passe en état *Reset Communication*.
5. **Reset Communication**
La communication du nœud est réinitialisée puis l'état du nœud reste en *Pre-Operational*.

L'HAT CANopen Safety travaille en tant que *NMT Slave* et prend en charge tous les services *NMT*.

6.8 Boot Up Protocol

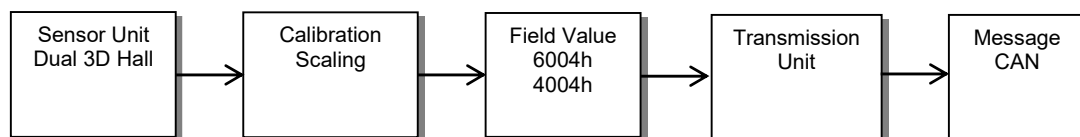
Après une réinitialisation, le *NMT Slave* envoie un objet *Boot Up*. Ceci n'est en principe pas différent d'un objet *Heartbeat* avec un statut à 0.

Cette fonction est réalisée par l'HAT CANopen Safety.

7 Flux de données dans le HAT CANopen Safety

Le schéma suivant montre le flux de données d'un HAT CANopen Safety ainsi que chaque indice de l'*Object Dictionary*.

Flux de données dans un HAT CANopen Safety



7.1 Sensor Unit

Le capteur mesure l'angle de l'arbre par rapport à la bride du corps.
Le type de capteur (*Encoder*) est enregistré dans *Sensor Type*.

7.2 Calibration & Scaling

Les valeurs du capteur sont préparées par le biais de la fonction *Calibration & Scaling* et sont disponibles comme *FieldValue*.

7.3 Transmission Unit

Si l'un des événements suivants se produit, alors la valeur du PDO sera envoyée en fonction du *Transmission Type* paramétré.

1. L'*EventTimer* est écoulé (transfert cyclique).
2. Un ou plusieurs objets *Sync* ont été reçus (transfert synchrone).

8 L'Object Dictionary

8.1 Structure de l'Object Dictionary

Toutes les données sont enregistrées dans *l'Object Dictionary*. Le chapitre suivant décrit toutes les entrées prises en charge par l'HAT CANopen Safety. L'entrée de l'Index, conformément à la spécification, est toujours en notation hexadécimale (sans que la représentation décimale soit explicitement affichée). Pour chaque entrée, un type d'accès est donné. CANopen distingue les types d'accès suivants:

Types d'accès (Acces Type) pour l'Object Dictionary

1. const
Peut uniquement être lu et fournit toujours la même valeur.
2. read only
Peut uniquement être lu, mais la valeur peut changer pendant la mise en service.
3. write only
Peut uniquement être écrit.
4. read write
Peut être lu et écrit.

CANopen différencie les domaines suivants dans le Data Dictionary :

Domaines de l'Object Dictionary :

1. Index 0000h .. 1 FFFh: Communication profile specific entries
Réglages qui sont valables pour tous les appareils CANopen. Ces saisies sont définies dans le DS301 et DS302
2. Index 6000h .. 9FFFh: Device profile specific entries
Données spécifique à l'appareil définies dans un Draft Standard.
Le HAT CANopen peut réaliser le profil DS406.
3. Index 2000h .. 5FFFh: Manufacturer specific entries
Données supplémentaires spécifiques du fabricant qui ne sont définies dans aucune spécification.

8.2 Structure de la pièce spécifique d'après DS406

Le HAT CANopen Safety implique le DS406. Elle décrit le comportement et les fonctionnalités des appareils de mesure angulaire.

9 Entrées dans l'Object Dictionary

Voici les fonctionnalités réalisées par le HAT CANopen Safety. La description détaillée des entrées peut être lue dans dans le et .

9.1 Communication Profile Specific Entries (DS301)

9.1.1 Index 1000h : DeviceType (read only)

Contient le numéro du profil d'équipement utilisé :

406 (0196h) - *Device profile number*

Ainsi que l'extension spécifique au profil :

1 (0001h) - *Single Turn absolute rotary encoder*

9.1.2 Index 1001h : ErrorRegister (read only)

Contient l'état actuel des erreurs (voir *EMCY*, ainsi que [1]).

9.1.3 Index 1002h: Manufacturer status register (read only)

Restitue toujours 0

9.1.4 Index 1003h: Pre-defined error field (read only)

Les erreurs qui sont apparues dans l'appareil CANopen et qui ont été signalées par le biais de l'*Emergency object* (voir CiA 301) sont affichées ici

9.1.5 Index 1005h: SyncMessageIdentifier (read write)

Permet de régler les *COB-ID* pour les objets *SYNC*.

9.1.6 Index 1008h : ManufacturerDeviceName (const)

Fournit le nom de l'appareil en chaîne de caractères („HAT 1000“ ou „HAT 3000“).

9.1.7 Index 1009h : ManufacturerHardwareVersion (const)

Fournit la version du matériel en chaîne de caractères (par exemple « 01.01 »).

9.1.8 Index 100Ah : ManufacturerSoftwareVersion (const)

Fournit la version du logiciel en chaîne de caractères (par exemple « 01.07 »). Les deux premiers caractères indiquent la version, les deux derniers le niveau de mise à jour.

9.1.9 Index 1010h : StoreParameters

Les paramètres actuels peuvent être sauvegardés dans la mémoire non volatile par l'écriture de la chaîne de caractères «save» (65766173h).

Lorsque les paramètres sont modifiés, le HAT CANopen Safety ne les mémorise pas automatiquement, cela se passe uniquement sur demande.



Attention: Les modifications des réglages doivent être explicitement sauvegardées par le *StoreParameters*, sinon elles seront perdues à la mise à l'arrêt de l'appareil ou par les commandes *NMT (Reset Node)* et *Reset Communication*.

CANopen offre la possibilité à l'aide des différents *Subindex* de restaurer les différentes plages de réglage.
Les Subindex 1, 2, 3 et 4 sont pris en charge.

Pour de plus amples informations voir [1].



Attention: Les paramètres de la vitesse de transmission du Baudrate et du NodeID ne sont pas modifiés pour la commande *StoreAllParameters*.

Subindex utilisés :

- 0: **LargestSubindexSupported (read only)**
- 1: **StoreAllParameters (read write)**
- 2: **StoreCommunicationParamters (read write)** (index de 1000h à 1FFFh)
- 3: **StoreApplicationParamteres (read write)** (index de 6000h à 9FFFh)
- 4: **StoreLssParameters (read write)** (index de 2000h à 20FFh)

9.1.10 Index 1011h: RestoreDefaultParameters

Les préreglages usine peuvent être transférés dans la mémoire non volatile par l'écriture de la chaîne de caractères « load » (64616F6Ch).

Le HAT CANopen Safety travaille cependant encore avec les paramètres actuels jusqu'à l'arrêt ou jusqu'à l'exécution des commandes *Reset Node* et *Reset Communication*.

CANopen offre la possibilité à l'aide des différents *Subindex* de restaurer les différentes plages de réglage.
Les subindex 1, 2, 3 et 4 sont pris en charge.

Pour de plus amples informations voir [1]



Attention: Les paramètres de la vitesse de transmission du Baudrate et du NodeID ne sont pas modifiés pour la commande *RestoreAllParameters*.

Subindex utilisés:

- 0: **LargestSubindexSupported (read only)**
- 1: **RestoreAllParameters (read write)**
- 2: **RestoreCommunicationParamters (read write)** (index de 1000h à 1FFFh)
- 3: **RestoreApplicationParamteres (read write)** (index de 6000h à 9FFFh)
- 4: **RestoreLssParameters (read write)** (index de 2000h à 20FFh)

9.1.11 Index 1014h: CobIdEmergencyMessage (read write)

Permet de régler les *COB-ID* pour les objets *EMCY* (voir *EMCY*).

9.1.12 Index 1017h: ProducerHeartbeatTime (read write)

Permet de régler le temps de *Heartbeat* - (en millisecondes). La valeur 0 signifie que la fonction n'est pas active (voir *Heartbeat*).

9.1.13 Index 1018h: IdentityObject

Identity Object identifie le HAT CANopen Safety. L'identification se compose de 4 chiffres de 32 bits. La combinaison de ces 4 chiffres forme une identification unique dans le monde pour un appareil.

Subindex utilisés :

- 0: LargestSubIndexSupported (read only)**
- 1: VendorID (read only)**
Code fabricant explicite (DAh pour HYDAC ELECTRONIC GmbH)
- 2: ProductCode (read only)**
Code produit d'Hydac Electronic (p.ex. : 925010)
- 3: RevisionNumber (read only)**
Numéro de révision de l'appareil
- 4: SerialNumber (read only)**
Numéro de série de l'appareil

9.1.14 Index 1029h: Error behaviour

Subindex utilisés:

- 0: Nr of Error Classes (read only)**
- 1: Communication Error (read write)**
 - 0 : Change to NMT state Pre-operational
 - 1 : No change of the NMT state
 - 2 : Change to NMT state Stopped
- 2: Specific Error Class (read write)**
 - 0 : Change to NMT state Pre-operational
 - 1 : No change of the NMT state
 - 2 : Change to NMT state Stopped

9.1.15 Index 1301h: SRDO communication parameter

Comprend les paramètres de communication du premier objet de données relié à la sûreté :

Subindex utilisés:

- 0: LargestSubindexSupported (read only)**
- 1: Information direction (read write)**
- 2: Refresh-time or SCT (read write)**
- 3: Validation time (read write)**
- 4: Transmission type (read only)**
- 5: COB-ID 1 (read write)**
- 6: COB-ID 2 (read write)**

9.1.16 Index 1381h: SRDO mapping parameter

Comprend les paramètres de communication du premier objet mapping relié à la sûreté :

Subindex utilisés:

- 0: LargestSubindexSupported (read write)**
- 1: Mapping Object 1 Data Not Inverted (read write)**
- 2: Mapping Object 1 Data Inverted (read write)**
- 3: Mapping Object 2 Data Not Inverted (read write)**
- 4: Mapping Object 2 Data Inverted (read write)**

- 5: Mapping Object 3 Data Not Inverted (read write)
- 6: Mapping Object 3 Data Inverted (read write)
- 7: Mapping Object 4 Data Not Inverted (read write)
- 8: Mapping Object 4 Data Inverted (read write)
- 9: Mapping Object 5 Data Not Inverted (read write)
- 10: Mapping Object 5 Data Inverted (read write)
- 11: Mapping Object 6 Data Not Inverted (read write)
- 12: Mapping Object 6 Data Inverted (read write)
- 13: Mapping Object 7 Data Not Inverted (read write)
- 14: Mapping Object 7 Data Inverted (read write)
- 15: Mapping Object 8 Data Not Inverted (read write)
- 16: Mapping Object 8 Data Inverted (read write)

En sortie d'usine, l'HAT CANopen Safety a les données suivantes :

Index 1381h		
Sous-index	Sommaire	Signification
0	2	Deux valeurs sont transmises dans le SRDO.
1	60040020h	La première valeur dans le SRDO est la valeur de l'Index 6004h, Subindex 0 avec une largeur de 20h (=32 bit).
2	40040020h	La deuxième valeur dans le SRDO est la valeur de l'Index 4004h, Subindex 0 avec une largeur de 20h (=8 bit).

9.1.17 Index 13FEh: Configuration Valid (read write)

Cet enregistrement sert à valider la configuration SRDO. Le contrôle de plausibilité est lancé par l'écriture de la valeur A5h. Si le contrôle se déroule avec succès, la valeur est acceptée. Autrement la valeur est placée à 0.

9.1.18 Index 13FFh : Safety Configuration Checksum

Contient les sommes de contrôle de chaque configuration SRDO.

Subindex utilisés :

- 0: LargestSubindexSupported (read only)
- 1: Checksum-1 (read write)

9.1.19 Index 1800h : TPDO communication parameter

Cet enregistrement définit les paramètres pour la transmission PDO. Plus précisément :

Paramètres pour la transmission PDO

1. *COB-ID*
Définit l'identifiant du PDO. Le bit de poids le plus fort (Bit 31) n'appartient plus à l'ID, il a la signification "disable PDO". Si le bit est occupé, le transfert des PDO est bloqué.
2. *Transmission Type*
Définit le type de transmission.
Des valeurs entre 0 et 240 signifient une synchronisation de la transmission. Le chiffre correspond au nombre d'objets SYNC qui doivent être reçus pour que le PDO soit envoyé.
La valeur 254 signifie une transmission spécifique au fabricant et la valeur 255 une

transmission spécifique au type profil de l'appareil. Le PDO est envoyé cycliquement à 254 et 255, à condition qu'un temps (*Event Time*) différent de 0 soit réglé.

3. *Inhibit Time*

Cette valeur détermine l'intervalle minimal pour la transmission PDO, si FEh et FFh ont été choisis comme type de transmission. La valeur est définie comme multiple de 100µs. La valeur 0 décommute l'*inhibit time*.

4. *Reserved – sans signification*

5. *Event Time*

Définit le temps de cycle asynchrone pour les transmissions de type 254 et 255 (en millisecondes). La valeur 0 signifie: aucune transmission cyclique.

Subindex utilisés :

0: **LargestSubindexSupported (read only)**

1: **COBIDUsedByPDO (read write)**

2: **TransmissionType (read write)**

3: **InhibitTime (read write)**

4: **Reserved**

5: **EventTimer(read write)**

9.1.20 Index 1A00h: TPDO mapping parameter

Avec cette entrée, on définit quelles sont les données transmises avec un PDO. Subindex 0 donne le nombre de données du PDO. Sous le Subindex 1, nous avons l'Index et le Subindex ainsi que le nombre de bits des premières données définies (il en est de même pour les Subindex 2 à 8).

En sortie d'usine, l'HAT CANopen Safety a les données suivantes :

Index 1A00h		
-- Désactivé par le TPDO communication parameter --		
Sous-index	Sommaire	Signification
0	1	Une valeur est transmise dans le <i>PDO</i> .
1	60040020h	La première valeur dans le <i>PDO</i> est la valeur de l' <i>Index 6004h</i> , <i>Subindex 0</i> avec une largeur de 20h (=32 bit).

Subindex utilisés :

0: **NumberOfMappedApplicationObjectsInPDO (read write)**

Les valeurs 0 à 8 sont autorisées.

En conséquence, soit aucun PDO, soit un PDO contenant jusqu'à 8 valeurs sont transmis.

ObjectToBeMapped (1 à 8) (read write)

Les valeurs suivantes sont admises pour le HAT CANopen Safety :

10010008h = Error Register largeur 8bit

10020020h = Manufacturer Status Register largeur 32bit

40040020h = valeur process Position Value **inversée** avec largeur 32bit (DS406 [6])

60040020h = valeur process Position Value avec largeur 32bit (DS406 [6])

65030010h = Alarms avec largeur 16bit (DS406 [6])

9.1.21 Index 1F80h: NMT-Startup (read / write)

Si le bit 2 est défini, alors on passe automatiquement au statut "Operational" lorsque le statut "Pre-Operational" est atteint (DS302).

Les valeurs autorisées sont : **08h** et **0Ch**

9.2 Device Profile Specific Entries (DS406)

9.2.1 Index 6000h: Operating Parameter (read only)

Cet enregistrement contient les paramètres pour la restitution de la *Position Value* ([6]).

Code sequence

Commissioning diagnostic control

Commissioning bit

Scaling function control

Measuring direction

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
msh	msh	msh	msh	r	r	r	r	r	r	r	r	md	sfc	cdc	cs
MSB															LSB

Signal	Value	Definition
cs	0b	Code sequence = CW
	1b	Code sequence = CCW
cdc	0b	Commissioning diagnostic control not supported
	1b	Commissioning diagnostic control supported
sfc	0b	Scaling function control disabled
	1b	Scaling function control enabled
md	0b	Measuring direction forward
	1b	Measuring direction reward
r		Reserved for further use
msh	0b	Manufacturer-specific function disabled
	1b	Manufacturer-specific function enabled

9.2.2 Index 6001h: Measuring units per revolution (read only)

Cet enregistrement indique quelles unités de mesure sont comptabilisées par rotation ([6]).

9.2.3 Index 6002h : Total measuring range in measuring units (read only)

Cet enregistrement indique la taille de la plage de mesure en unités de mesure ([6]).

9.2.4 Index 6003h : Preset value (read write)

Cet enregistrement contient le point zéro du capteur en unités de mesure (*Preset value*, [6]).

9.2.5 Index 6004h : Position value (read only)

Cet enregistrement contient la valeur de position mesurée (*Position Value*, [6]).

9.2.6 Index 6500h : Operating Status (read only)

Correspond à Index 6000h [6].

9.2.7 Index 6501h : Single-turn resolution and Measuring step (read only)

Indique à combien de pas de mesure une rotation complète correspond [6].

9.2.8 Index 6503h : Alarms (read only)

Restitue 1 en cas de détection d'erreur de mesure, sinon 0 [6].

9.2.9 Index 6504h : Supported alarms (read only)

Restitue toujours 1 [6].

9.2.10 Index 6505h : Warnings (read only)

Restitue toujours 0 [6].

9.2.11 Index 6506h : Supported warnings (read only)

Restitue toujours 0 [6].

9.2.12 Index 6507h : Profile and software version (read only)

Restitue le logiciel de l'appareil et la version du profil [6].

9.2.13 Index 6508h : Operating time (read only)

Compteur d'heures de service, non réalisé pour HAT, restitue toujours FFFFFFFFh.

9.2.14 Index 6509h : Offset value (read only)

Sous cet index, la valeur Offset est consignée pour cette position.

9.2.15 Index 650Ah : Module identification (read only)

Sous cet index, les données fabricant concernant l'Offset et les positions minimale et maximale sont consignées.

Subindices utilisés:

- 0: NumberOfEntries (read only)
- 1: Manufacturer offset value (read only)
- 2: Manufacturer min position value (read only)
- 3: Manufacturer max position value (read only)

9.3 Manufacturer Specific Entries

9.3.1 Index 2001h : Node-ID

La *Node ID* est consignée sous cet index.

Subindices utilisés:

- 4: NumberOfEntries (read only)
- 5: CurrentNodeid (read only)
- 6: DemandedNodeid (read write)

Pour que la nouvelle Node-ID prenne effet, la commande *StoreLssParameters* devra être envoyée, et le périphérique redémarré.

9.3.2 Index 2002h : Baudrate

La vitesse de communication est consignée sous cet index.

L'attribution des entrées pour le taux de transfert correspond à la description faite dans le DS305, *Layer Setting Services and Protocols*.

Entrée	0	1	2	3	4	5	6	7	8
Vitesse de transmission	1000 kbit/s	800 kbit/s	500 kbit/s	250 kbit/s	125 kbit/s	reserved	50 kbit/s	20 kbit/s	10 kbit/s

Pour que le nouveau taux de transfert prenne effet, la commande *StoreLssParameters* devra être envoyée, puis il faut redémarrer le périphérique.

Subindex utilisés :

- 0: **NumberOfEntries (read only)**
- 1: **CurrentBaudrate (read only)**
- 2: **DemandedBaudrate (read write)**

9.3.3 Index 4006h: Position Value (read only) Position invertiert

Identique à l'index 6004h, mais **inversé au niveau des bit**.

9.3.4 Indices supplémentaires dans la plage 2000h à 5FFFh (réservé)



Les indices dans cette plage contiennent des **paramétrages usine importants et ne doivent pas être modifiés par les clients !**
Attention : toute **intervention** peut provoquer un **dysfonctionnement** du capteur angulaire !

10 Layer setting services (LSS) et protocoles

Les services LSS et les protocoles documentés en CiA DS305 V2.2, voir [4], prennent en charge l'interrogation et la configuration des différents paramètres du *Data Link Layer* et de l'*Application Layer* d'un LSS-Slave via le réseau CAN par le biais d'un LSS-Master.

Les paramètres suivants sont soutenus :

- Node-ID
- Vitesse de transmission
- Adresse LSS, selon l'Identity Objekt 1018h

L'accès au LSS-Slave s'effectue ici via son adresse LSS composée de :

- ID Vendeur
- Code produit
- Numéro de révision et
- Numéro de série

Le capteur soutient les services suivants :

Switch mode services

- Switch mode selective
 - répondent à un certain LSS-Slave
- Switch mode global
 - répondent à tous les LSS-Slaves

Configuration services

- Configure Node-ID
 - Configurer Node-ID
- Configure bit timing parameters
 - Configurer la vitesse de communication
- Activate bit timing parameters
 - Activer la vitesse de communication
- Store configured parameters
 - Enregistrer les paramètres configurés

Inquiry services

- Inquire LSS-address
 - Interroger adresse LSS
- Inquire Node-ID
 - Interroger Node-ID

Identification services

- LSS identify remote slave
 - Identification de LSS-Slave dans une plage définie
- LSS identify slave
 - Retour d'info des LSS-Slaves au commando précédent
- LSS identify non-configured remote slave
 - Identification de LSS-Slaves non configurés, Node-ID = FFh
- LSS identify non-configured slave
 - Retour d'info des LSS-Slaves au commando précédent

10.1 Finite state automaton, FSA

Le FSA correspond à une machine d'état et définit le comportement d'un LSS-Slave. La machine d'état est pilotée par LSS COBs générés par un LSS-Master, ou NMT COBs générés par un NMT-Master ou des transitions d'état NMT locales.

Les LSS-Modes soutiennent les états suivants :

(0) Initial : pseudo état, indique l'activation du FSA

(1) LSS waiting : soutient tous les services comme indiqué ci-dessous

(2) LSS configuration : soutient tous les services comme indiqué ci-dessous

(3) Final : pseudo-état, indique la désactivation du FSA

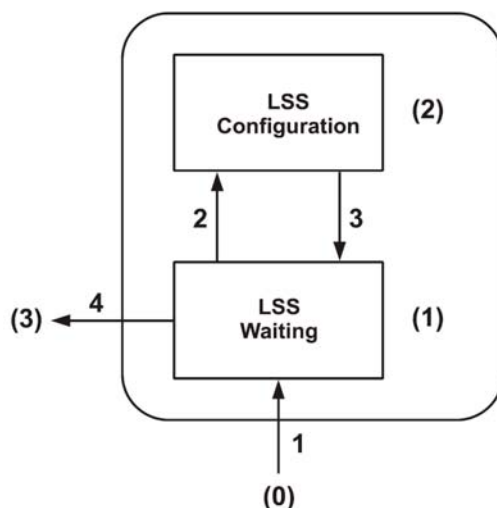


Figure 1 : LSS-Modes

Comportement de l'état des services soutenus

Services	Waiting	Configuration
Switch mode global	Oui	Oui
Switch mode selective	Oui	Non
Activate bit timing parameters	Non	Oui
Configure bit timing parameters	Non	Oui
Configure Node-ID	Non	Oui
Store configuration	Non	Oui
Inquire LSS-address	Non	Oui
LSS identify remote slave	Oui	Oui
LSS identify slave	Oui	Oui
LSS identify non-configured remote slave	Oui	Oui
LSS identify non-configured slave	Oui	Oui

Transition d'état LSS FSA

Transition	Evénements	Actions
1	Transition automatique après l'initialisation pour l'entrée soit dans l'état NMT PRE OPERATIONAL ou l'état NMT STOPPED ou encore l'état NMT RESET COMMUNICATION avec Node-ID = FFh.	aucun
2	Commando LSS 'switch state global' avec 'configuration_switch' de paramètre ou commando 'switch state selective'	aucun
3	Commando LSS 'switch state global' avec paramètre 'waiting_switch'	aucun
4	Transition automatique lorsqu'une Node-ID non valide a été modifiée et que la nouvelle Node ID a été correctement stockée dans la mémoire non volatile et l'état LSS waiting a été demandé.	aucun

Dès que le LSS FSA subit d'autres transitions d'état dans le NMT FSA du NMT PRE OPERATIONAL sur NMT STOPPED et inversement, ceci ne conduit pas au retour dans le LSS FSA.

10.2 Transmission de services LSS

Le LSS-Master interroge chaque service au travers des LSS-Services qui sont alors exécuté par le LSS-Slave. La communication entre LSS-Master et LSS-Slave est effectuée via les protocoles LSS implémentés.

Comme pour la transition SDO, on utilise aussi ici deux COB-IDs pour l'envoi et la réception :

COB-ID	Signification
7E4h	LSS-Slave → LSS-Master
7E5h	LSS-Master → LSS-Slave

Tableau 1 : COB-IDs pour services LSS

10.2.1 Format de message LSS

La plage de données maximale de 8 byte d'un message CAN est occupée comme suit par un service LSS :

CS	Données						
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7

Tableau 2 : Message LSS

Byte 0 contient le **Command-Specifier** (CS), 7 bytes suivent ensuite pour les données.

10.3 Protocoles Switch mode

10.3.1 Protocole Switch mode selective

Le protocole indiqué a implémenté le *Switch mode global service* et commande le comportement d'état du LSS-Slave. Tous les LSS-Slaves dans le réseau peuvent être amenés en *Waiting Mode* ou *Configuration Mode* via le LSS-Master.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
COB-ID	CS	MODE		Reserved by CiA				
7E5h	04	0 = Waiting Mode 1 = Configuration Mode						

10.3.2 Switch mode selective Protokoll

Le protocole indiqué a implémenté le *Switch mode selective service* et commande le comportement d'état du LSS-Slave. Via le LSS-Master, seul le LSS-Slave dans le réseau peut être amené en *Configuration Mode* dont les attributs LSS correspondent à l'adresse LSS.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
COB-ID	CS	Vendor-ID				Reserved by CiA		
7E5h	64	LSB		MSB				

	0	1	2	3	4	5	6	7
COB-ID	CS	Product-Code				Reserved by CiA		
7E5h	65	LSB		MSB				

	0	1	2	3	4	5	6	7
COB-ID	CS	Revision-Number				Reserved by CiA		
7E5h	66	LSB		MSB				

	0	1	2	3	4	5	6	7
COB-ID	CS	Serial-Number				Reserved by CiA		
7E5h	67	LSB		MSB				

LSS-Slave --> LSS-Master

	0	1	2	3	4	5	6	7
COB-ID	CS	Reserved by CiA						
7E4h	68							

10.1 Protocoles Configuration

10.1.1 Configure protocole Node-ID

Le protocole indiqué a implémenté le *Configure NMT-Adresse service*. Le Node-ID d'un LSS-Slave peut être configuré dans le réseau à l'aide du LSS-Master. Ici, seul un LSS-Slave peut se trouver en *Configuration Mode*. Pour enregistrer le nouveau Node-ID, le *Store configuration protocol* doit être transmis au LSS-Slave. Afin d'activer le nouveau Node-ID, le service NMT *Reset Communication* (82h) doit être appelé.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
COB-ID	CS	Node-ID	Reserved by CiA					
7E5h	17	1...127 et 255						

LSS-Slave --> LSS-Master

	0	1	2	3	4	5	6	7
COB-ID	CS	Error Code	Spec. Erreur	Reserved by CiA				
7E4h	17							

Error Code

0 : Exécution réussie

1...254 : Réserve

255 : Apparition d'une erreur spécifique à l'application

Specific Error

Si Error Code = 255 --> apparition d'une erreur spécifique à l'application, sinon réservé par la CiA.

10.1.2 Configurer protocole bit timing parameters

Le protocole indiqué a implémenté le *Configure bit timing parameters*. La vitesse de communication d'un LSS-Slave peut être configurée dans le réseau à l'aide du LSS-Master. Pour enregistrer la nouvelle vitesse de communication, le *Store configuration protocol* doit être transmis au LSS-Slave.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
COB-ID	CS	Table Selector	Table Index	Reserved by CiA				
7E5h	19	0	0..8					

LSS-Slave --> LSS-Master

	0	1	2	3	4	5	6	7
COB-ID	CS	Error Code	Spec. Erreur	Reserved by CiA				
7E4h	19							

Table Selector

0 : Tableau de vitesses de communication CiA standard

Table Index

0 : 1 Mbit/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

Error Code

0 : Exécution réussie
 1 : Vitesse de communication sélectionnée non soutenue
 2...254 : Réservé
 255 : Apparition d'une erreur spécifique à l'application

Specific Error

Si Error Code = 255 --> apparition d'une erreur spécifique à l'application, sinon réservé par la CiA

10.1.3 Protocole Active bit timing parameters

Le protocole indiqué a implémenté l'*Activate bit timing parameters service* et activé la vitesse de communication déterminée via *Configure bit timing parameters protocol* pour tous les LSS-Slaves dans le réseau qui se trouvent en *Configuration Mode*.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
COB-ID	CS	Switch Delay [ms]		Reserved by CiA				
7E5h	21	LSB	MSB					

Switch Delay

Le paramètre *Switch Delay* définit la longueur de deux périodes de temporisation (D1, D2) de même longueur. Grâce à cela, on évite le fonctionnement du bus avec différents paramètres de vitesse de communication. Après le temps D1 et un temps de traitement, la commutation est effectuée en interne dans le LSS Slave. Après le temps D2, le LSS Slave transmet en retour avec des messages CAN et la nouvelle vitesse de communication paramétrée.

La formule suivante s'applique :

Switch Delay > durée de traitement existante la plus longue d'un LSS-Slave.

10.1.4 Protocole Store configuration

Le protocole indiqué a implémenté le *Store configured parameters service*. Les paramètres configurés d'un LSS Slave particulier dans le réseau peuvent être déposés dans la mémoire non volatile. Ici, seul un LSS-Slave peut se trouver en *Configuration Mode*.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
COB-ID	CS	Reserved by CiA						
7E5h	23							

LSS-Slave --> LSS-Master

	0	1	2	3	4	5	6	7
COB-ID	CS	Error Code	Spec. Erreur	Reserved by CiA				
7E4h	23							

Error Code

0 : Exécution réussie

1 : *Store configuration* non soutenu

2...254 : Réserve

255 : Apparition d'une erreur spécifique à l'application

Specific Error

Si Error Code = 255 --> apparition d'une erreur spécifique à l'application, sinon réservé par la CiA

10.2 Protocoles Inquire LSS-Address

10.2.1 Protocole Inquire Identity Vendor-ID

Le protocole indiqué a implémenté le *Inquire LSS-Adresse service*. Le Vendor-ID d'un LSS-Slave peut être lu dans le réseau à l'aide du LSS-Master. Ici, seul un LSS-Slave peut se trouver en *Configuration Mode*.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
COB-ID	CS	Reserved by CiA						
7E5h	90							

LSS-Slave --> LSS-Master

	0	1	2	3	4	5	6	7
COB-ID	CS	Vendor-ID (= Index 1018h:01)				Reserved by CiA		
7E4h	90	LSB			MSB			

10.2.2 Protocole Inquire Product-Code

Le protocole indiqué a implémenté le *Inquire LSS-Adresse service*. Le nom d'appareil du fabricant d'un LSS-Slave peut être lu dans le réseau à l'aide du LSS-Master. Ici, seul un LSS-Slave peut se trouver en *Configuration Mode*.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
COB-ID	CS	Reserved by CiA						
7E5h	91							

LSS-Slave --> LSS-Master

	0	1	2	3	4	5	6	7
COB-ID	CS	Product-Code (= Index 1018h:02)				Reserved by CiA		
7E4h	91	LSB			MSB			

10.2.3 Protocole Inquire Identity Revision-Number

Le protocole indiqué a implémenté le *Inquire LSS-Adresse service*. Le numéro de révision d'un LSS-Slave peut être lu dans le réseau à l'aide du LSS-Master. Ici, seul un LSS-Slave peut se trouver en *Configuration Mode*.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
COB-ID	CS	Reserved by CiA						
7E5h	92							

LSS-Slave --> LSS-Master

	0	1	2	3	4	5	6	7
COB-ID	CS	Revision-Number (= Index 1018h:03)				Reserved by CiA		
7E4h	92	LSB			MSB			

10.2.4 Protocole Inquire Identity Serial-Number

Le protocole indiqué a implémenté le *Inquire LSS-Adresse service*. Le numéro de série d'un LSS-Slave peut être lu dans le réseau à l'aide du LSS-Master. Ici, seul un LSS-Slave peut se trouver en *Configuration Mode*.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
COB-ID	CS	Reserved by CiA						
7E5h	93							

LSS-Slave --> LSS-Master

	0	1	2	3	4	5	6	7
COB-ID	CS	Serial-Number (= Index 1018h:04)				Reserved by CiA		
7E5h	93	LSB		MSB				

10.2.5 Protocole Inquire Node-ID

Le protocole indiqué a implémenté le *Inquire Node-ID service*. Le Node-ID d'un LSS-Slave peut être lu dans le réseau à l'aide du LSS-Master. Ici, seul un LSS-Slave peut se trouver en *Configuration Mode*.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
COB-ID	CS	Reserved by CiA						
7E5h	94							

LSS-Slave --> LSS-Master

	0	1	2	3	4	5	6	7
COB-ID	CS	Node-ID		Reserved by CiA				
7E4h	94	1...127 et 255						

Node-ID

Correspond au Node-ID de l'appareil sélectionné. Si la Node-ID vient justement d'être modifiée via le *Configure Node-ID service*, la Node-ID d'origine est retournée. La Node-ID actuelle est retournée seulement après exécution du service NT *Reset Communication* (82h).

10.3 Protocoles Identification

10.3.1 Protocole identify LSS remote slave

Le protocole indiqué a implémenté le *LSS identify remote slaves service*. Tous les LSS-Slaves peuvent être identifiés dans le réseau dans une certaine plage via le LSS-Master. Tous les LSS-Slaves qui correspondent au Vendor-ID, Product-Code, à la plage de Revision-No et à la plage de Serial-No. indiqués, répondent avec le *LSS identify slave protocol*.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
COB-ID	CS	Vendor-ID				Reserved by CiA		
7E5h	70	LSB		MSB				

	0	1	2	3	4	5	6	7
COB-ID	CS	Product-Code				Reserved by CiA		
7E5h	71	LSB		MSB				

	0	1	2	3	4	5	6	7
COB-ID	CS	Revision-Number-Low				Reserved by CiA		
7E5h	72	LSB		MSB				

	0	1	2	3	4	5	6	7
COB-ID	CS	Revision-Number-High				Reserved by CiA		
7E5h	73	LSB		MSB				

	0	1	2	3	4	5	6	7
COB-ID	CS	Serial-Number-Low				Reserved by CiA		
7E5h	74	LSB		MSB				

	0	1	2	3	4	5	6	7
COB-ID	CS	Serial-Number-High				Reserved by CiA		
7E5h	75	LSB		MSB				

10.3.2 Protocole LSS identify slave

Le protocole indiqué a implémenté le *LSS identify slave service*. Tous LSS-Slaves correspondant aux attributs de la LSS-Adress indiqués dans le *LSS identify remote slaves protocol* répondent à ce protocole.

LSS-Slave --> LSS-Master

	0	1	2	3	4	5	6	7
COB-ID	CS	Reserved by CiA						
7E4h	79							

10.3.3 Protocole LSS identify non-configured remote slave

Le protocole indiqué a implémenté le *LSS identify non-configured remote slave service*. Tous les LSS-Slaves (Node-ID = FFh) non configurés ont été identifiés dans le réseau via le LSS-Master. Les LSS-Slaves concernés répondent avec le *LSS identify non-configured remote slave protocol*.

LSS-Slave --> LSS-Master

	0	1	2	3	4	5	6	7
COB-ID	CS	Reserved by CiA						
7E4h	76							

10.3.4 Protocole LSS identify non-configured slave

Le protocole indiqué a implémenté le *LSS identify non-configured slave service*. Tous les LSS-Slaves, qui possèdent une Node-ID (FFh) non valide, répondent avec ce protocole après exécution du *LSS identify non-configured slave protocol*.

LSS-Slave --> LSS-Master

	0	1	2	3	4	5	6	7
COB-ID	CS	Reserved by CiA						
7E4h	80							

11 Raccordement

Le raccordement peut être réalisé à l'aide du branchement spécifique fourni, voir notice d'utilisation HAT CANopen Safety (la notice fait partie de la fourniture du HAT CANopen Safety).

11.1 Commuter la tension d'alimentation

Après que le branchement et que tous les réglages ont été effectués, la tension d'alimentation peut être commutée.

Après la commutation de la tension d'alimentation et l'achèvement de l'initialisation, le capteur passe en état de pré-service (PRE-OPERATIONAL). Cet état est confirmé par le message Boot-Up „**COB-ID 700h+Node-ID**“. Si le capteur détecte un défaut interne, un message d'urgence est transmis avec le code de défaut (voir 6.5).

Dans l'état PRE-OPERATIONAL seul un paramétrage via le service des données objets est possible. Il est toutefois possible de configurer les PDOs en utilisant les SDOs. Si le capteur a été transformé en état OPERATIONAL, une transmission des PDOs est aussi possible (voir 6.7).

11.2 Réglages des Node-ID et de la vitesse de communication avec les services LSS

11.2.1 Configuration du Node-ID, procédure

Hypothèse :

- Adresse LSS inconnue
- Le LSS-Slave est le seul participant dans le réseau
- Il faut paramétrer la Node-ID 12 déc.

Mode opératoire :

- Placer le LSS-Slave avec le Command Specifier 04 *Switch mode global protocol*, Mode = 1 dans le *Configuration Mode*. (voir 0)
- Exécuter Command Specifier 17 *Configure NMT-Address protocol*, Node-ID = 12. (voir 10.1.1)
--> Attendre l'info retour et contrôler la réussite de l'exécution,
--> Error Code = 0

- Exécuter Command Specifier 23 *Store configuration protocol*.
--> attendre l'info retour et contrôler la réussite de l'exécution,
--> Error Code = 0
(voir 10.1.4)
- Décommuter puis recommuter la tension d'alimentation du LSS-Slave. La nouvelle configuration est désormais active.

11.2.2 Configuration de la vitesse de communication, procédure

Hypothèse :

- Adresse LSS inconnue
- Le LSS-Slave est le seul participant dans le réseau.
- Il faut paramétrer la vitesse de communication à 125 kbit/s.

Mode opératoire :

- Appeler NMT- Command Specifier *Stop Remote Node* (02h), afin de placer le LSS-Slave dans le *Stopped state*. Le LSS-Slave ne devrait plus envoyer de messages CAN --> Heartbeat décommuté (voir 6.7).
- Placer le LSS-Slave avec le Command Specifier *Switch mode global protocol*, Mode = 1 en *Configuration Mode*. (voir 10.3.1).
- Exécuter Command Specifier 19 *Configure bit timing parameters protocol*, Table Selector = 0, Table Index = 4
--> Attendre l'information de retour et contrôler la réussite de l'exécution,
--> Error Code = 0
(voir 10.1.2).
- Appeler Command Specifier 21 *Activate bit timing parameters protocol*, afin que la nouvelle vitesse de communication soit active. (voir 10.1.3).
- Exécuter Command Specifier 23 *Store configuration protocol*.
--> Attendre le retour d'info et contrôler la réussite de l'exécution,
--> Error Code = 0
(voir 10.1.4).
- Décommuter puis recommuter la tension d'alimentation du LSS-Slave. La nouvelle configuration est désormais active.

12 Mise en service

12.1 Interface CAN

L'interface bus CAN est définie par la norme ISO/DIS 11898 et elle spécifie les deux couches inférieures du modèle de référence CAN.

La conversion de l'information du capteur en protocole CAN (CAN 2.0A) se déroule via un contrôleur CAN. La fonction du contrôleur CAN est surveillée par un Watchdog.

Le profil de communication CANopen (CiA Standard DS301, voir [1]) repose sur le CAN Application Layer (CAL) et décrit comment les services des appareils sont utilisés. Le profil CANopen permet de définir les profils des appareils pour une E/S décentralisée.

Le système de mesure avec protocole CANopen soutient le profil de l'appareil pour le capteur angulaire (CiA Draft Standard 406, version 1.3.0, voir [6]).

La fonctionnalité de communication et les objets utilisés dans le profil du capteur sont décrits dans un fichier EDS (Electronic Data Sheet). Si un programme d'assistance de configuration CANopen est utilisé (p.ex. CANSETTER), l'utilisateur peut lire les objets (SDO's) du capteur et programmer ce dernier.

12.2 Fichier EDS

Le fichier EDS (fiche de données électronique) contient toutes les informations sur les paramètres spécifiques aux systèmes de mesure ainsi que le mode de fonctionnement des systèmes de mesure. Le fichier EDS est intégré à l'outil de configuration réseau CANopen, pour assurer un paramétrage et une utilisation optimisés.

Le fichier EDS adapté à l'appareil peut être identifié au nom de l'appareil et au numéro de révision correspondant dans le nom du fichier.

Le fichier comprenant la description de cette interface CANopen Safety se trouve pour téléchargement sur notre site internet sous

<http://www.hydac.com/de-fr/produits/capteurs/show/Material/index.html>

HYDAC ELECTRONIC GMBH

Hauptstr. 27
D-66128 Sarrebruck
Allemagne

Web : www.hydac.com
E-mail : electronic@hydac.com
Tél. : +49 (0) 6897 / 509 – 01
Fax : +49 (0)6897 / 509-1726

HYDAC Service

Pour toute question concernant les réparations, HYDAC Service se tient à votre disposition.

HYDAC SERVICE GMBH

Hauptstr. 27
D-66128 Sarrebruck
Allemagne

Tél. : +49 (0) 6897 / 509 – 1936
Fax : +49 (0) 6897 / 509 – 1933

Remarque

Les indications de cette notice se réfèrent aux conditions de fonctionnement et cas d'utilisation décrits. Pour des conditions d'utilisation et/ou de fonctionnement différentes, veuillez vous adresser au service technique compétent.

Pour toute question technique, demande de conseils ou en cas de panne, veuillez vous mettre en relation avec votre représentant HYDAC.