

Protokoll-
Beschreibung
Protocol Description
CANopen Safety

HIT 1000 / HIT 1500

(Originalanleitung)
(Translation of original instructions)



Inhaltsverzeichnis

1	Einleitung	6
2	Funktionen des HIT 1000 / 1500 CANopen Safety	6
3	Übertragungsraten	7
4	CAN-Frames	7
5	Node-ID	7
6	Übertragungsdienste	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.....	12
6.7	Network Management Services (NMT)	12
6.8	Boot Up Protocol.....	13
7	Datenfluss im HIT 1000 / 1500 CANopen	14
7.1	Sensor Unit.....	14
7.2	Calibration & Scaling	14
7.3	Filter & Fusion	14
7.4	Transmission Unit.....	14
8	Das Object Dictionary	15
8.1	Aufbau des Object Dictionary.....	15
8.2	Struktur des gerätespezifischen Teils nach DS410.....	15
9	Einträge im Object Dictionary	16
9.1	Communication Profile Specific Entries (DS301)	16
9.1.1	Index 1000h: DeviceType (read only)	16
9.1.2	Index 1001h: ErrorRegister (read only)	16
9.1.3	Index 1003h: Pre-defined error field (read only).....	16
9.1.4	Index 1005h: SyncMessageIdentifier (read write)	16
9.1.5	Index 1008h: ManufacturerDeviceName (const)	16
9.1.6	Index 1009h: ManufacturerHardwareVersion (const).....	16
9.1.7	Index 100Ah: ManufacturerSoftwareVersion (const).....	16
9.1.8	Index 1010h: StoreParameters	16
9.1.9	Index 1011h: RestoreDefaultParameters	17
9.1.10	Index 1014h: CobIdEmergencyMessage (read write)	17
9.1.11	Index 1017h: ProducerHeartbeatTime (read write)	17
9.1.12	Index 1018h: IdentityObject	17
9.1.13	Index 1029h: Error behaviour	18

9.1.14 Index 1301h: SRDO communication parameter	18
9.1.15 Index 1381h: SRDO mapping parameter	18
9.1.16 Index 13FEh: Configuration Valid (read write).....	19
9.1.17 Index 13FFh: Safety Configuration Checksum.....	19
9.1.18 Index 1800h / 1801h / 1802h / 1803h: TPDO communication parameter	19
9.1.19 Index 1A00h / 1A01h / 1A02h / 1A03h: TPDO mapping parameter.....	20
9.1.20 Index 1F80h: NMT-Startup (read / write).....	22
9.2 Device Profile Specific Entries (DS410)	22
9.2.1 Index 6000h: Resolution (read only)	22
9.2.2 Index 6010h: Slope long16 (read only) Sichere Neigung	22
9.2.3 Index 6011h: Slope long16 operating parameter (read write).....	22
9.2.4 Index 6020h: Slope lateral16 (read only) Sichere Neigung	22
9.2.5 Index 6021h: Slope lateral16 operating parameter (read write).....	22
9.3 Manufacturer Specific Entries	23
9.3.1 Index 2001h: NodeID	23
9.3.2 Index 2002h: Baudrate	23
9.3.3 Index 4000h: Status der Neigungsmessung invertiert	23
9.3.4 Index 4010h: Slope long16 (read only) Sichere Neigung invertiert.....	23
9.3.5 Index 4020h: Slope lateral16 (read only) Sichere Neigung invertiert.....	23
9.3.6 Index 5000h: Status der Neigungsmessung.....	23
9.3.7 Index 5130h: Skalierte Beschleunigungswerte.....	24
9.3.8 Index 5131h: Skalierte Drehratenwerte.....	24
9.3.9 Index 5132h: Neigung im Raum (bewegungskompensiert).....	24
9.3.10 Weitere Indizes im Bereich 2000h bis 5FFFh (reserved).....	25
10 Layer setting services (LSS) und Protokolle.....	26
10.1 Finite state automaton, FSA.....	27
10.2 Übertragung von LSS-Diensten	28
10.2.1 LSS-Nachrichtenformat.....	28
10.3 Switch mode Protokolle	29
10.3.1 Switch mode global Protokoll.....	29
10.3.2 Switch mode selective Protokoll.....	29
10.4 Configuration Protokolle.....	30
10.4.1 Configure Node-Id Protokoll.....	30
10.4.2 Configure bit timing parameters Protokoll	30
10.4.3 Activate bit timing parameters Protokoll	31
10.4.4 Store configuration Protokoll.....	32
10.5 Inquire LSS-Address Protokolle	32
10.5.1 Inquire Identity Vendor-ID Protokoll	32
10.5.2 Inquire Identity Product-Code Protokoll.....	33
10.5.3 Inquire Identity Revision-Number Protokoll	33
10.5.4 Inquire Identity Serial-Number Protokoll.....	34
10.5.5 Inquire Node-ID Protokoll.....	34
10.6 Identification Protokolle.....	35
10.6.1 LSS identify remote slave Protokoll.....	35
10.6.2 LSS identify slave Protokoll	36
10.6.3 LSS identify non-configured remote slave Protokoll	36
10.6.4 LSS identify non-configured slave Protokoll.....	36

11 Anschluss	37
11.1 Einschalten der Versorgungsspannung	37
11.2 Einstellen der Node-ID und Baudrate mittels LSS-Dienste.....	37
11.2.1 Konfiguration der Node-ID, Ablauf	37
11.2.2 Konfiguration der Baudrate, Ablauf	38
12 Inbetriebnahme	39
12.1 CAN – Schnittstelle.....	39
12.2 EDS-Datei.....	39

Vorwort

Für Sie, den Benutzer unseres Produktes, haben wir in dieser Dokumentation die wichtigsten Hinweise zum Bedienen und Warten zusammengestellt.

Sie dient Ihnen dazu, das Produkt kennen zu lernen und seine bestimmungsgemäßen Einsatzmöglichkeiten optimal zu nutzen.

Diese Dokumentation muss ständig am Einsatzort verfügbar sein. Bitte beachten Sie, dass die in dieser Dokumentation gemachten Angaben der Softwaretechnik zu dem Zeitpunkt der Literaturerstellung entsprechen.

Entdecken Sie beim Lesen dieser Dokumentation Fehler oder haben weitere Anregungen und Hinweise, so wenden Sie sich bitte an:

HYDAC ELECTRONIC GMBH
Technische Dokumentation
Hauptstraße 27
66128 Saarbrücken
-Deutschland-
Tel: +49(0)6897 / 509-01
Fax: +49(0)6897 / 509-1726
Email: electronic@hydac.com

Die Redaktion freut sich über Ihre Mitarbeit.

„Aus der Praxis für die Praxis“

1 Einleitung

Die Neigungsgeber der Familie HIT 1000 / 1500 CANopen Safety entsprechen dem CANopen Standard gemäß folgenden Profilen und Standards:

- [1] CiA DS301, Version: 4.2.0 (21 February 2011)
CANopen application layer and communication profile
- [2] CiA DS302, Version: 4.1 (02 February 2009)
Additional application layer functions - Part 2: Network management
- [3] CiA DS303, Version: 1.4 (14 August 2006)
Additional specification - Part 2: Representation of SI units and prefix
- [4] CiA DS 304, Version V1.0.1 (01 Januar 2005)
Framework for safety-relevant communication
- [5] CiA DS305, Version: 2.2 (26 August 2008)
Layer setting services (LSS) and protocols
- [6] CiA DS410, Version 1.3.0 (23 February 2010)
Device profile for inclinometer

Dieses Handbuch beschreibt die vom HIT 1000 / 1500 CANopen Safety unterstützten Funktionen. Dabei werden Grundkenntnisse von CAN, CANopen und CANopen Safety vorausgesetzt. Die genaue Funktionsweise ist in den *oben genannten Dokumenten* beschrieben. Da diese in Englisch abgefasst sind, werden die in diesem Handbuch beschriebenen Dinge, zur besseren Zuordnung, mit dem in der Spezifikation genannten englischen Begriff bezeichnet und *kursiv* dargestellt.

2 Funktionen des HIT 1000 / 1500 CANopen Safety

- **Berechnung** der Neigung aus den Beschleunigungskomponenten in 3 Raumachsen mit **Tiefpassfunktion** bei HIT 1000 und HIT 1500 (statische Neigungsmessung = **sichere Neigung**)
- **Berechnung** der Neigung durch Fusion von Beschleunigungs- und Drehratenkomponenten bei **HIT 1500** (bewegungskompensierte Neigungsmessung, hochdynamisch = **nicht sichere Neigung**)
- Umrechnung der Neigungswerte in **Euler-Winkel** (Einheit °, ein- oder zweiachsig)
- Übertragung der aktuellen Neigungswerte als **PDO oder SRDO** bei folgenden Ereignissen:
 - **Synchron** über empfangene SYNC-Objekte (nur PDO)
 - **Asynchron** zyklisch im Bereich von 5 Millisekunden bis >1 Minute

3 Übertragungsraten

Der HIT 1000 / 1500 CANopen Safety unterstützt folgende Übertragungsraten (Baudraten):

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

Das Timing entspricht der DS301, *Bit rates and timing*.

Die verwendete Übertragungsrate ist in einem nicht-flüchtigen Speicher hinterlegt. Sie ist im Auslieferungszustand auf **250 kbit/s** eingestellt und kann über den CAN-Bus geändert werden (Siehe *Object Dictionary Index 2002h*).

4 CAN-Frames

Der HIT 1000 / 1500 CANopen Safety unterstützt die in der Spezifikation geforderten 11-bit Standard-Frames mit 11-bit Identifier sowie 29-bit Frames mit 29-bit Identifier.

5 Node-ID

Zum Betrieb des HIT 1000 / 1500 CANopen Safety in einem CANopen-Netzwerk ist es notwendig, dass eine innerhalb des Netzes einmalige *Node-ID* eingestellt wird.

Die eingestellte *Node-ID* ist in einem nicht-flüchtigen Speicher hinterlegt und kann über den CAN-Bus (siehe *Object Dictionary Index 2001h*) eingestellt werden. Im Auslieferungszustand ist die Adresse **1** eingestellt.

6 Übertragungsdienste

6.1 Service Data Object (SDO)

Bei CANopen werden alle Daten eines Gerätes (Einstellparameter und Messdaten) in einem *Object Dictionary* unter einem definierten *Index* abgelegt. Verschiedene Einträge des *Object Dictionary* werden mit einem *Subindex* noch weiter untergliedert. Mit den SDOs können nun andere Netzteilnehmer das *Object Dictionary* des HIT 1000 / 1500 CANopen Safety auslesen oder beschreiben.

Der HIT 1000 / 1500 CANopen Safety übernimmt dabei die Rolle eines *Servers*, das Gerät, das die Daten auslesen oder beschreiben will, die eines *Clients*.

Zum Transfer von Daten muss der HIT 1000 / 1500 CANopen Safety ein *Receive-SDO* besitzen, mit dem er Daten empfängt und ein *Transmit-SDO*, mit dem er die Daten sendet. Ablauf des Datentransfers:

Auslesen des Object Dictionary :

1. Ein Gerät (*Client*) sendet das *Receive-SDO* des HIT 1000 / 1500 CANopen Safety (*Server*).
In diesem *SDO* befindet eine Kennung, dass das *Object Dictionary* gelesen werden soll, sowie der gewünschte *Index* und *Subindex*.
2. Der HIT 1000 / 1500 CANopen Safety (*Server*) sendet sein *Transmit-SDO*. In diesem befinden sich ebenfalls der *Index* und der *Subindex*, sowie die gelesenen Daten.

Beschreiben des Object Dictionary :

1. Ein Gerät (*Client*) sendet das *Receive-SDO* des HIT 1000 / 1500 CANopen Safety (*Server*).
In diesem *SDO* befinden eine Kennung, dass das *Object Dictionary* beschrieben werden soll, sowie der gewünschte *Index*, *Subindex* und die einzuschreibenden Daten.
2. Der HIT 1000 / 1500 CANopen Safety (*Server*) sendet sein *Transmit-SDO*. In diesem befinden sich ebenfalls der *Index* und der *Subindex*, sowie eine Kennung, dass das *Object Dictionary* beschrieben wurde.

Sollte dabei ein Fehler auftreten, z.B. dass der angegebene *Index* nicht existiert, oder dass versucht wurde einen *read only* Eintrag zu beschreiben, oder dass die Daten nicht innerhalb des Gültigkeitsbereiches lagen, so enthält das *Transmit-SDO* eine entsprechende *Abort SDO Transfer* - Kennung und einen entsprechenden *Abort Code* (siehe [1])

Die jeweilige *COB-ID* des *SDO* entspricht dem in der DS301 festgelegten *Pre defined Connection Set* und ist nicht änderbar.

COB-IDs für 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)

Die Datenübertragung mittels *SDOs* ist zwar sehr flexibel, hat aber für die Übertragung von Messwerten oder Stellgrößen einige Nachteile: Es kann nur ein Datum gelesen werden, die Daten müssen erst mit einem *SDO* angefordert werden und dadurch dass der jeweilige *Index* und *Subindex* mit übertragen wird, steigt der so genannte Overhead weiter.

Aus diesem Grund definiert CANopen Safety so genannte *Process Data Objects*. Diese enthalten nur die notwendigen Nutzdaten. Es gibt zwei Arten von *PDOs*:

1. *Transmit-PDOs*
Hiermit kann ein Messgerät seine Messwerte senden.
2. *Receive-PDOs*
Hiermit können einem Stellglied oder einer Regelungseinheit die Stellgrößen übertragen werden.

Welche Daten sich nun in einem *PDO* befinden, wird durch das so genannte *PDO-Mapping* festgelegt. Dieses *PDO-Mapping* ist im *Object Dictionary* hinterlegt (siehe *Object Dictionary, Index 1A00h, 1A01h, 1A02h, 1A03h*).

Mit welcher ID und bei welchem Ereignis ein *PDO* übertragen wird, ist in dem *PDO-Transmission Type* festgelegt. Diese Einstellungen befinden sich ebenfalls im *Object Dictionary* hinterlegt (siehe *Object Dictionary, Index 1800h, 1801h, 1802h, 1803h*).

Ereignisse, die zum Senden eines *PDOs* führen:

1. Empfang eines *SYNC* Objektes (Synchroner Transfer).
2. Ablauf einer einstellbaren Zykluszeit im Bereich von 5 Millisekunden bis >1 Minute (Zyklischer Transfer).

Der HIT 1000 / 1500 CANopen Safety implementiert mehrere *Transmit-PDO*, welche die aktuellen Sensor- bzw. Prozesswerte beinhalten.

Die DS410 sieht als Standardeinstellung die Übertragung der aktuellen Messwerte als 16-bit oder 32-bit Wert vor.

Der HIT 1000 / 1500 CANopen Safety implementiert nur die Übertragung als 16-bit Wert.

6.3 Synchronisation Object (SYNC)

SYNC Objekte dienen zur Realisierung eines synchronen Datentransfers. Ein *SYNC* Objekt ist im Prinzip eine CAN Nachricht mit einem definierten Identifier, ohne Daten. CANopen unterscheidet zwischen *SYNC Producer* und *SYNC Consumers*. *SYNC Producer* sind Geräte am Bus, die in einstellbaren Zeitabständen ein *SYNC* senden. *SYNC Consumers* sind Geräte, die auf den Empfang eines *SYNC* reagieren. In einem CANopen Netzwerk können mehrere *SYNC* Objekte existieren. Unterschieden werden die einzelnen *SYNC* Objekte anhand der *SYNC-ID*, welche dem verwendeten CAN Identifier entspricht. Die verwendete *SYNC-ID* ist im *Object Dictionary* hinterlegt.

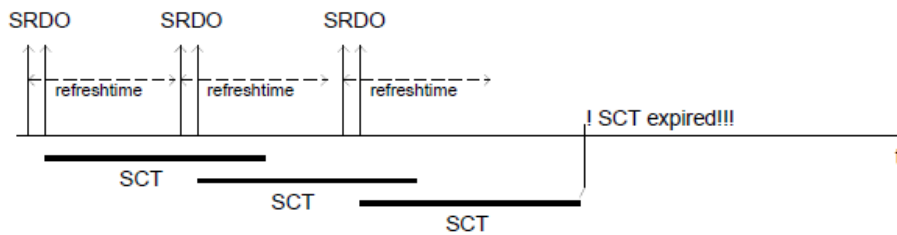
Der HIT 1000 / 1500 CANopen Safety bietet die Funktionalität eines *SYNC Consumers*. Bei entsprechender Einstellung des *PDO Transmission Type* wird beim Empfang eines *SYNC* ein *PDO* gesendet. Die *SYNC-ID* ist auf 80h voreingestellt und kann im *Object Dictionary* geändert werden (siehe *Object Dictionary, Index 1005h*). Unter *PDO Transmission Type* kann die Anzahl der empfangenen *SYNC* Objekte, die zum Senden eines *PDO*, führt eingestellt werden.

6.4 Safety-relevant Data Object (SRDO)

Die Datenübertragung mittels so genannter *Safety-relevant Data Objects* sorgt für einen sicheren Transfer von Messwerten. SRDOs können zum Senden oder Empfangen von Daten genutzt oder deaktiviert werden.

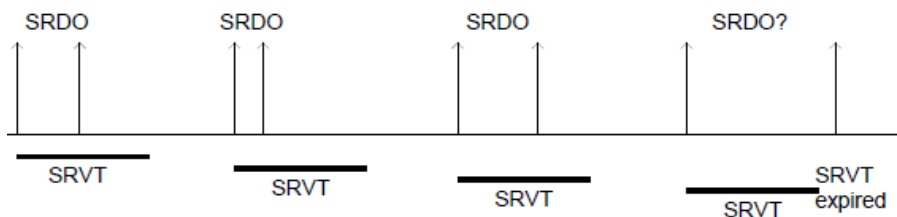
Ein SRDO besteht aus zwei eigenständigen CAN-Frames. Das erste Frame enthält die Daten in normaler Darstellung, das zweite CAN-Frame enthält die Daten in invertierter Darstellung. Dies erlaubt dem Empfänger die Daten zu verifizieren.

Die Zeit zwischen zwei SRDOs wird als *Refresh Time* bzw. *Safeguard cycle time (SCT)* bezeichnet und dient zur Timeoutüberwachung zwischen zwei SRDOs.



Quelle: CiA Draft Standard 304, CANopen Framework for safety-relevant communication, Version 1.0.1, Januar 2005, CAN in Automation (CiA), S. 9

Die Zeit zwischen den beiden Frames wird als *Safety-relevant object validation time (SRVT)* bezeichnet und ermöglicht dem Empfänger eine Timeouterkennung innerhalb eines SRDOs.



Quelle: CiA Draft Standard 304, CANopen Framework for safety-relevant communication, Version 1.0.1, Januar 2005, CAN in Automation (CiA), S. 10

Welche Daten sich nun in einem SRDO befinden wird durch das so genannt SRDO-Mapping festgelegt. Dieses Mapping ist im *Object Dictionary* hinterlegt.

Die Firmware unterstützt ein *Transmit-SRDO* und keine *Receive-SRDOs*.

Folgende Punkte sind bei der Konfiguration zu beachten:

- Das Ändern der Kommunikationsparameter ist nur im Zustand *Pre-Operational* möglich.
- Um die COB-IDs zu ändern, ist das zugehörige SRDO vorher abzuschalten (*Information Direction* ist 0).
- Die SCT auf dem Empfängerknoten ist entsprechend größer zu wählen als auf dem Senderknoten, um Synchronisationsprobleme durch Softwarelaufzeiten auszuschließen ($\pm 2,8\text{ms}$).
- Die SRVT ist so zu wählen, dass eine Unterbrechung des SRDOs auf dem CAN-Bus durch höherwertige Nachrichten nicht zu einem Fehler beim Empfänger führt.
- Die SCT und SRVT sind ebenso stark abhängig von der Bitrate.
- Die Prüfsumme ist nach Änderung für jedes betroffene SRDO neu zu berechnen und im *Object Dictionary* (siehe *Index 13FFh*) abzulegen.
- Nach Änderung der Konfiguration ist diese zurückzulesen und zu verifizieren.
- Nach erfolgter Verifizierung der Daten ist die Konfiguration als gültig zu erklären (siehe *Object Dictionary, Index 13FE, Configuration Valid*).

Folgende Punkte sind bei der Plausibilitätsprüfung der Konfiguration zu beachten:

- Die SRVT muss kleiner als die SCT sein.
- Die COB-ID der Nachricht mit den normalen Daten muss stets ungerade sein.
- Die COB-ID der Nachricht mit den invertierten Daten muss stets gerade sein.

Weitere Plausibilitätsprüfungen werden nicht durchgeführt. Für die Sinnfälligkeit der eingestellten Parameter ist der Konstrukteur der Gesamtanlage verantwortlich (siehe COB-ID Bereich DS304, siehe [4]).

Die Plausibilitätsprüfung erfolgt beim Schreiben des Wertes A5h auf *Configuration Valid*. Das Schreiben ist nur erfolgreich, wenn die Prüfung erfolgreich abgeschlossen wurde. Die Plausibilitätsprüfung erfolgt für alle SRDOs, die aktiviert sind (*Information Direction* ungleich 0).



Achtung:
SRDOs werden nur gesendet, wenn die Konfiguration gültig ist.

6.5 Emergency Object (EMCY)

EMCY Objekte werden beim Auftreten eines Fehlers gesendet. EMCY Objekte enthalten einen *Emergency Error Code*, den Inhalt des *Error register* sowie ein *Manufacturer specific Error Field*. Ist ein gemeldeter Fehler beseitigt oder verschwunden, so wird dies ebenfalls durch ein spezielles EMCY Objekt gemeldet.

Eine Emergency-Nachricht wird gesendet, wenn ein Fehler auftritt oder dieser Fehler wieder verschwindet. Die Nachricht ist folgendermaßen aufgebaut:

Fehler	Emergency ErrorCode	Manufacturer SpecificErrorField	Fehler Kategorie
Kein Fehler	0000h	0000h	Fehler beseitigt
Fehler beim Laden des User-Setup	FF00h	0001h	Herstellerspezifischer Fehler (Bit 7)
Controllerfehler	FF00h	0002h	Herstellerspezifischer Fehler (Bit 7)
Reserviert	N/A	0003h	N/A
Reserviert	N/A	0004h	N/A
CAN Error Passive	8120h	0005h	Kommunikationsfehler (Bit 4)
Recovered from Bus-off	8140h	0006h	Kommunikationsfehler (Bit 4)

Das EMCY Objekt hat die voreingestellte ID 80h+*Node-ID* und kann im *Object Dictionary* geändert werden (siehe *Object Dictionary, Index 1014h*).

6.6 Heartbeat

Mit dem *Heartbeat Protocol* kann eine Überwachung der einzelnen Teilnehmer durchgeführt werden. CANopen unterscheidet zwischen folgenden Funktionen:

1. *Heartbeat Producer*
sendet in zyklischen Abständen ein *Heartbeat* Objekt.
2. *Heartbeat Consumer*
überwacht das Senden von bestimmten *Heartbeat* Objekten.

Die Zykluszeit ist im *Object Dictionary* in Millisekunden einstellbar. Eine Zeitangabe von 0 bedeutet „*Heartbeat* nicht aktiv“.

Mit dem *Heartbeat* Objekt wird immer der Status des *Heartbeat Producers* als Byte mit übertragen.

Bedeutung des Heartbeat- Objekthinhalte

Wert	Status	Anmerkung
0	<i>BOOTUP</i>	Das Gerät hat gebootet.
4	<i>STOPPED</i>	Das Gerät ist gestoppt.
5	<i>OPERATIONAL</i>	Das Gerät arbeitet normal.
127	<i>PRE-OPERATIONAL</i>	Das Gerät sendet keine <i>PDOs</i> , kann aber <i>SDOs</i> bearbeiten.

Der HIT 1000 / 1500 CANopen Safety kann als *Heartbeat Producer* arbeiten. Die ID des *Heartbeat* ist $700h + \text{Node-ID}$. Die Zeit ist mit 0 (nicht aktiv) voreingestellt und kann geändert werden (siehe *Object Dictionary, Index 1017h*).

6.7 Network Management Services (NMT)

NMT Objekte dienen dazu Geräte zu starten, zu stoppen oder zurückzusetzen. CANopen unterscheidet zwischen folgenden Funktionalitäten:

1. *NMT Master*
steuert andere Knoten.
2. *NMT Slave*
wird von einem *Master* gesteuert.

In einem CANopen Netzwerk existiert nur ein *NMT* Objekt mit dem Identifier 0. Es werden immer 2 Bytes übertragen. Das erste Byte enthält den *Command Specifier*, der den Befehl repräsentiert, das zweite Byte enthält die *Node-ID* des Knotens, der diesen Befehl ausführen soll. Ein Wert von 0 bedeutet, dass dieser Befehl für alle Knoten gilt. Folgende Befehle sind möglich:

NMT Befehle

1. **Start Remote Node**
Der Knoten wechselt in den Zustand *Operational*.
2. **Stop Remote Node**
Der Knoten wechselt in den Zustand *Stopped*.
3. **Enter Pre-Operational**
Der Knoten wechselt in den Zustand *Pre-Operational*.

4. **Reset Node**

Der "Device Profile Specific"-OD-Bereich wird zurückgesetzt, die Baudrate wird gegebenenfalls neu initialisiert und wechselt dann in den Zustand Reset Communication.

5. **Reset Communicatiuon**

Die Kommunikationseinheit des Knotens wird zurückgesetzt und danach wechselt der Knoten in den Zustand *Pre-Operational*.

Der HIT 1000 / 1500 CANopen Safety arbeitet als *NMT Slave* und unterstützt alle *NMT* Dienste.

6.8 Boot Up Protocol

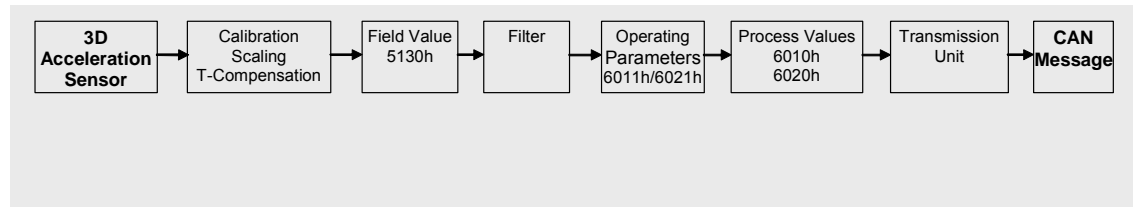
Wenn ein *NMT Slave* nach der Initialisierung in den Zustand *Pre-operational* wechselt, sendet er jeweils ein *Boot Up* Objekt. Dies ist im Prinzip nichts anders als ein *Heartbeat* Objekt mit dem Status 0.

Beim HIT 1000 / 1500 CANopen Safety ist diese Funktion realisiert.

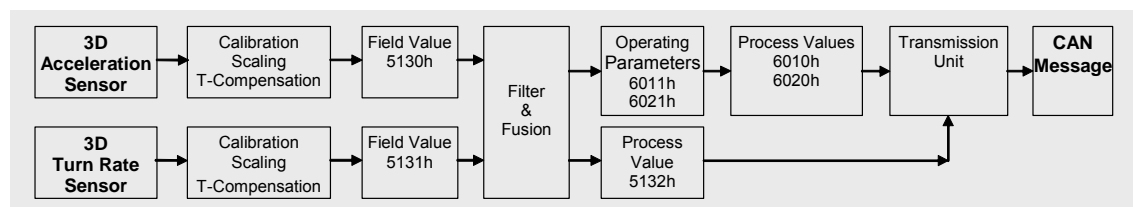
7 Datenfluss im HIT 1000 / 1500 CANopen

Nachfolgende Bilder zeigen den Datenfluss innerhalb des HIT 1000 / 1500 CANopen, sowie die jeweiligen Indizes des *Object Dictionary*.

Datenfluss im HIT 1000 CANopen



Datenfluss im HIT 1500 CANopen



7.1 Sensor Unit

Die Sensoreinheit erfasst die Neigung gegen die Horizontale und die Drehrate. Die Messwerte werden in digitaler Form bereitgestellt. Der Typ der Sensoreinheit (*Inclination Transducer*) ist in *Sensor Type* hinterlegt.

7.2 Calibration & Scaling

Die Sensorwerte werden in Calibration & Scaling aufbereitet und stehen als *FieldValue* zur Verfügung.

7.3 Filter & Fusion

Die kalibrierten Sensorwerte werden in Filter & Fusion gefiltert (HIT 1000 und HIT1500) und fusioniert (nur HIT 1500) und stehen nach der Operating Parameters Einheit als *ProcessValue* zur Verfügung.

7.4 Transmission Unit

Tritt eines der folgenden Ereignisse ein, so wird in Abhängigkeit des eingestellten *TransmissionType* der Wert des PDO gesendet.

1. Der *EventTimer* ist abgelaufen (zyklische Übertragung).
2. Ein oder mehrere *Sync* Objekte wurden empfangen (synchrone Übertragung).

8 Das Object Dictionary

8.1 Aufbau des Object Dictionary

Im *Object Dictionary* sind, wie bereits mehrfach erwähnt, alle Daten hinterlegt. In den folgenden Kapiteln sind die vom HIT 1000 / 1500 CANopen Safety unterstützten Einträge aufgeführt. Die Angabe des Index erfolgt spezifikationsgemäß immer in hexadezimaler Notation, ohne dass die hexadezimale Darstellung extra angezeigt wird. Bei jedem Eintrag ist auch die Zugriffsart angegeben. CANopen unterscheidet dabei folgende Zugriffsarten:

Zugriffsarten (Access Type) für das Object Dictionary

1. const
Kann nur gelesen werden und liefert immer den gleichen Wert.
2. read only
Kann nur gelesen werden, der Wert kann sich aber während des Betriebes ändern.
3. write only
Der Eintrag kann nur geschrieben werden.
4. read write
Der Eintrag kann geschrieben und gelesen werden.

CANopen unterscheidet innerhalb des Data Dictionary folgende Bereiche:

Bereiche des Object Dictionary:

1. Index 0000h .. 1FFFh: Communication profile specific entries
Einstellungen, die für alle CANopen-Geräte gelten. Diese Einträge sind in der DS301 und DS302 festgelegt.
2. Index 6000h .. 9FFFh: Device profile specific entries
Gerätespezifische Daten, die in einer Draft Standard festgelegt sind.
Der HIT 1000 / 1500 CANopen hat das Geräteprofil DS410 realisiert.
3. Index 2000h .. 5FFFh: Manufacturer specific entries
Herstellerspezifische zusätzliche Daten, die in keiner Spezifikation festgelegt sind.

8.2 Struktur des gerätespezifischen Teils nach DS410

Der HIT 1000 / 1500 CANopen Safety implementiert die DS410. Diese beschreibt das Verhalten und die Funktionalität von Neigungsgebern (*inclinometer*).

9 Einträge im Object Dictionary

Im Folgenden sind die vom HIT 1000 / 1500 CANopen Safety realisierten Funktionalitäten aufgezeigt. Eine detaillierte Beschreibung der Einträge kann in [1] und [6] nachgelesen werden.

9.1 Communication Profile Specific Entries (DS301)

9.1.1 Index 1000h: DeviceType (read only)

Enthält die Nummer des verwendeten Geräteprofils, hier die Nummer **410**, sowie die profilspezifische Erweiterung, hier eine 2, für die Unterstützung von 2 Achsen mit 16Bit maximaler Auflösung.

9.1.2 Index 1001h: ErrorRegister (read only)

Enthält den aktuellen Fehlerzustand (siehe *EMCY*, sowie [1]).

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

Hier werden die Fehler bereitgestellt, die im CANopen Gerät aufgetreten sind, und mittels *emergency object* signalisiert wurden (siehe CiA 301)

9.1.4 Index 1005h: SyncMessageIdentifier (read write)

Hier kann die *COB-ID* für das *SYNC*-Objekt eingestellt werden.

9.1.5 Index 1008h: ManufacturerDeviceName (const)

Liefert den Gerätenamen als Zeichenkette. („HIT1000“).

9.1.6 Index 1009h: ManufacturerHardwareVersion (const)

Liefert die Hardwareversion als Zeichenkette (z.B. „01.03“).

9.1.7 Index 100Ah: ManufacturerSoftwareVersion (const)

Liefert die Softwareversion als Zeichenkette (z.B. „03.02“). Die beiden ersten Zeichen bezeichnen die Version, die letzten beiden Zeichen den Revisionsstand.

9.1.8 Index 1010h: StoreParameters

Durch Einschreiben der Zeichenkette „save“ (65766173h) werden die aktuellen Einstellungen in den nicht flüchtigen Speicher übertragen.
Der HIT 1000 / 1500 CANopen Safety speichert Einstellungen nicht automatisch wenn sie geändert werden, sondern nur auf Anforderung.



ACHTUNG: Geänderte Einstellungen müssen mit *StoreParameters* explizit gesichert werden, sonst gehen sie beim Abschalten des Gerätes oder bei den *NMT-Befehlen Reset Node* und *Reset Communication* verloren.

CANopen bietet die Möglichkeit mit Hilfe verschiedener *Subindexes* verschiedene Parameterbereiche zu sichern.
Unterstützt wird hier Subindex 1, 2, 3 und 4.

Nähere Informationen sind [1] zu entnehmen.



Achtung: Die Einstellungen der Baudrate und der NodeID bleiben bei *StoreAllParameters* erhalten.

Verwendete Subindizes:

- 0: **LargestSubindexSupported (read only)**
- 1: **StoreAllParameters (read write)**
- 2: **StoreCommunicationParameters (read write)** (index von 1000h to 1FFFh)
- 3: **StoreApplicationParameters (read write)** (index von 6000h to 9FFFh)
- 4: **StoreLssParameters (read write)** (index von 2000h to 20FFFh)

9.1.9 Index 1011h: RestoreDefaultParameters

Durch Einschreiben der Zeichenkette „load“ (64616F6Ch) werden die werksseitigen Voreinstellungen in den nicht flüchtigen Speicher übertragen. Der HIT 1000 / 1500 CANopen Safety arbeitet allerdings bis zum Abschalten oder bis zur Ausführung der Befehle *Reset Node* und *Reset Communication* noch mit den aktuellen Einstellungen weiter.

CANopen bietet die Möglichkeit mit Hilfe verschiedener *Subindizes* verschiedene Parameterbereiche zu restaurieren. Unterstützt wird hier Subindex 1, 2, 3 und 4.

Nähere Informationen sind [1] zu entnehmen.



Achtung: Die Einstellungen der Baudrate und der NodeID bleiben bei *RestoreAllParameters* erhalten.

Verwendete Subindizes:

- 0: **LargestSubindexSupported (read only)**
- 1: **RestoreAllParameters (read write)**
- 2: **RestoreCommunicationParameters (read write)** (index von 1000h to 1FFFh)
- 3: **RestoreApplicationParameters (read write)** (index von 6000h to 9FFFh)
- 4: **RestoreLssParameters (read write)** (index von 2000h to 20FFFh)

9.1.10 Index 1014h: CobIdEmergencyMessage (read write)

Hier kann die *COB-ID* für das *EMCY*-Objekt eingestellt werden (siehe *EMCY*).

9.1.11 Index 1017h: ProducerHeartbeatTime (read write)

Hier kann die *Heartbeat* - Zeit in Millisekunden eingestellt werden. Der Wert 0 bedeutet, dass diese Funktion nicht aktiv ist (siehe *Heartbeat*).

9.1.12 Index 1018h: IdentityObject

Das Identity Objekt identifiziert den HIT 1000 / 1500 CANopen. Die Identifikation besteht aus vier 32bit-Zahlen. Die Kombination dieser 4 Zahlen ergibt eine weltweit eindeutige Identifikation eines Gerätes.

Verwendete Subindizes:

- 0: LargestSubIndexSupported (read only)**
- 1: VendorID (read only)**
Eindeutiger Herstellercode (DAh für HYDAC ELECTRONIC GmbH)
- 2: ProductCode (read only)**
Produktcode der Hydac Electronic (z.B: 924854)
- 3: RevisionNumber (read only)**
Revisionsnummer des Gerätes.
- 4: SerialNumber (read only)**
Seriennummer des Gerätes

9.1.13 Index 1029h: Error behaviour**Verwendete Subindizes:**

- 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

9.1.14 Index 1301h: SRDO communication parameter

Enthält die Kommunikationseinstellungen des ersten sicherheitsrelevanten Datenobjektes:

Verwendete Subindizes:

- 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)**

9.1.15 Index 1381h: SRDO mapping parameter

Enthält die Mappingeinstellungen des ersten sicherheitsrelevanten Datenobjektes:

Verwendete Subindizes:

- 0: LargestSubindexSupported (read only)**
- 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)

Im Auslieferungszustand hat der HIT 1000 / 1500 CANopen Safety folgende Einträge:

Index 1381h Statische (sichere) Neigung (HIT 1000 und HIT1500)		
Subindex	Inhalt	Bedeutung
0	6	Sechs Werte werden im SRDO übertragen.
1	60100010h	Der erste Wert im SRDO ist der Wert von <i>Index 6010h, Subindex 0</i> mit einer Breite von 10h (=16 bit)
2	40100010h	Der zweite Wert im SRDO ist der Wert von <i>Index 4010h, Subindex 0</i> mit einer Breite von 10h (=16 bit)
3	60200010h	Der dritte Wert im SRDO ist der Wert von <i>Index 6020h, Subindex 0</i> mit einer Breite von 10h (=16 bit)
4	40200010h	Der vierte Wert im SRDO ist der Wert von <i>Index 4020h, Subindex 0</i> mit einer Breite von 10h (=16 bit)
5	50000008h	Der fünfte Wert im PDO ist der Wert von <i>Index 5000 h, Subindex 0</i> ist mit einer Breite von 08h (=8 bit)
6	40000008h	Der sechste Wert im PDO ist der Wert von <i>Index 4000 h, Subindex 0</i> ist mit einer Breite von 08h (=8 bit)

9.1.16 Index 13FEh: Configuration Valid (read write)

Dieser Eintrag dient zur Validierung der SRDO Konfiguration. Durch Schreiben des Wertes A5h wird die Plausibilitätsprüfung angestoßen. Verläuft die Prüfung erfolgreich, wird der Wert akzeptiert. Andernfalls wird der Wert auf 0 gesetzt.

9.1.17 Index 13FFh: Safety Configuration Checksum

Enthält die Prüfsummen der einzelnen SRDO Konfigurationen.

Verwendete Subindizes:

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

9.1.18 Index 1800h / 1801h / 1802h / 1803h: TPDO communication parameter

Diese Einträge legen die PDO-Übertragung fest. Im Einzelnen sind dies:

Parameter zur PDO-Übertragung

1. COB-ID
Legt den Identifier für das PDO fest. Das höchstwertigste Bit (Bit31) des Eintrages gehört nicht mehr zur ID und hat die Bedeutung „disable PDO“. Ist dieses Bit gesetzt, so ist die Übertragung des PDO gesperrt.

2. *Transmission Type*
 Legt den Übertragungstyp fest.
 Werte zwischen 0 und 240 bedeuten eine synchrone Übertragung. Die Zahl steht für die Anzahl der SYNC Objekte, die empfangen werden müssen, bis das PDO gesendet wird.
 Der Wert 254 bedeutet eine herstellerspezifische Übertragung und der Wert 255 eine geräteprofilspezifische Übertragung. Bei 254 und 255 wird das PDO zyklisch gesendet, sofern eine Zeit (*Event Time*) ungleich 0 eingestellt ist.
3. *Inhibit Time*
 Dieser Wert legt das minimale Intervall für die PDO Übertragung fest, wenn als transmission type FEh and FFh gewählt wurde. Der Wert ist als Vielfaches von 100µs definiert. Der Wert 0 schaltet die *inhibit time* ab.
4. *Reserved – ohne Bedeutung*
5. *Event Time*
 Legt die Zykluszeit für asynchrone Übertragungen bei Transmission Type 254 und 255 in Millisekunden fest. Der Wert 0 bedeutet keine zeitgesteuerte Übertragung.

Verwendete Subindizes:

- 0: **LargestSubindexSupported (read only)**
- 1: **COBIDUsedByPDO (read write)**
- 2: **TransmissionType (read write)**
- 3: **InhibitTime (read write)**
- 4: **Reserved**
- 5: **EventTimer(read write)**

9.1.19 Index 1A00h / 1A01h / 1A02h / 1A03h: TPDO mapping parameter

Mit diesen Einträgen wird festgelegt, welche Daten mit dem PDO übertragen werden. Subindex 0 gibt die Anzahl der Daten im PDO an. Unter Subindex 1 ist der Index und Subindex sowie die Anzahl der Bits des ersten Datums hinterlegt, entsprechendes gilt für die Subindizes 2 bis 8.

Im **Auslieferungszustand** hat der HIT 1000 / 1500 CANopen Safety folgende Einträge:

Index 1A00h Statische Neigung (HIT 1000 und HIT1500)		
-- Deaktiviert durch TPDO communication parameter --		
Subindex	Inhalt	Bedeutung
0	3	Zwei Werte werden im <i>PDO</i> übertragen.
1	60100010h	Der erste Wert im <i>PDO</i> ist der Wert von <i>Index 6010h, Subindex 0</i> mit einer Breite von 10h (=16 bit)
2	60200010h	Der zweite Wert im <i>PDO</i> ist der Wert von <i>Index 6020h, Subindex 0</i> mit einer Breite von 10h (=16 bit)
3	50000008h	Der dritte Wert im <i>PDO</i> ist der Status der Neigungsmessung

Index 1A01h Skalierte Beschleunigungswerte (HIT 1000 und HIT1500) -- Deaktiviert durch TPDO communication parameter --		
Subindex	Inhalt	Bedeutung
0	3	Drei Werte werden im <i>PDO</i> übertragen.
1	51300110h	Der erste Wert im <i>PDO</i> ist der Wert von <i>Index 5130h, Subindex 1</i> mit einer Breite von 10h (=16 bit)
2	51300210h	Der zweite Wert im <i>PDO</i> ist der Wert von <i>Index 5130h, Subindex 2</i> mit einer Breite von 10h (=16 bit)
3	51300310h	Der dritte Wert im <i>PDO</i> ist der Wert von <i>Index 5130h, Subindex 3</i> mit einer Breite von 10h (=16 bit)

Index 1A02h Skalierte Drehratenwerte (nur HIT 1500) -- Deaktiviert durch TPDO communication parameter --		
Subindex	Inhalt	Bedeutung
0	3	Drei Werte werden im <i>PDO</i> übertragen.
1	51310110h	Der erste Wert im <i>PDO</i> ist der Wert von <i>Index 5131h, Subindex 1</i> mit einer Breite von 10h (=16 bit)
2	51310210h	Der zweite Wert im <i>PDO</i> ist der Wert von <i>Index 5131h, Subindex 2</i> mit einer Breite von 10h (=16 bit)
3	51310310h	Der dritte Wert im <i>PDO</i> ist der Wert von <i>Index 5131h, Subindex 3</i> mit einer Breite von 10h (=16 bit)

Index 1A03h Neigung bewegungskompensiert (nur HIT 1500) -- Aktiviert durch TPDO communication parameter --		
Subindex	Inhalt	Bedeutung
0	2	Zwei Werte werden im <i>PDO</i> übertragen.
1	51320110h	Der erste Wert im <i>PDO</i> ist der Wert von <i>Index 5132h, Subindex 1</i> mit einer Breite von 10h (=16 bit)
2	51320210h	Der zweite Wert im <i>PDO</i> ist der Wert von <i>Index 5132h, Subindex 2</i> mit einer Breite von 10h (=16 bit)

Verwendete Subindizes:

0: NumberOfMappedApplicationObjectsInPDO (read write)

Es sind die Werte 0 bis 8 zulässig.
Entsprechend wird kein *PDO* übertragen, oder ein *PDO* mit bis zu 8 Werten.

ObjectToBeMapped (1 bis 8) (read write)

Beim HIT 1000 / 1500 CANopen Safety folgende Werte zulässig:

10010008h = Error Register 8bit-Breite

40000008h = Staus Neigung 8bit-Breite **invertiert**

40100010h = Prozesswert Slope long16 mit 16bit-Breite **invertiert** (DS410 vgl. [6])

40200010h = Prozesswert Slope lateral16 mit 16bit-Breite **invertiert** (DS410 vgl. [6])

50000008h = Staus Neigung 8bit-Breite

51300110h = Prozesswert Beschleunigung in Richtung der X-Achse (siehe 9.3.6)
 51300210h = Prozesswert Beschleunigung in Richtung der Y-Achse
 51300310h = Prozesswert Beschleunigung in Richtung der Z-Achse

51310110h = Prozesswert Drehrate um die X-Achse (siehe 9.3.8)
 51310210h = Prozesswert Drehrate um die Y-Achse
 51310310h = Prozesswert Drehrate um die Z-Achse

51320110h = Prozesswert kompensierte Neigung (wie Slope long16 s.u.)
 51320210h = Prozesswert kompensierte Neigung (wie Slope lateral16 s.u.)

60100010h = Prozesswert Slope long16 mit 16bit-Breite (DS410 vgl. [6])
 60200010h = Prozesswert Slope lateral16 mit 16bit-Breite (DS410 vgl. [6])

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

Wird Bit 2 gesetzt, so wird automatisch bei Erreichen des „Pre-Operational“ Status in den „Operational“ Status gewechselt (DS302).
 Erlaubte Werte sind: **08h** und **0Ch**

9.2 Device Profile Specific Entries (DS410)

9.2.1 Index 6000h: Resolution (read only)

Dieser Eintrag enthält die Auflösung der Neigungswerte in Winkelgrad.
 Der HIT 1000 / 1500 CANopen Safety liefert den Wert 0Ah zurück.
 Gemäß DS410 entspricht das einer Auflösung von 0,01° (siehe [6]).

9.2.2 Index 6010h: Slope long16 (read only) Sichere Neigung

Dieser Eintrag enthält die Neigung in longitudinaler Richtung als vorzeichenbehafteter 16Bit Wert (siehe [6]).

9.2.3 Index 6011h: Slope long16 operating parameter (read write)

Dieser Eintrag enthält die Informationen zur Interpretation des Messwerts SlopeLong16.
 Der HIT 1000 / 1500 CANopen Safety liefert standardmäßig den Wert 00h zurück (siehe [6]).

Bit 7 – 1	Bit 0	Funktion
0	0	Invertierung aus
0	1	Invertierung ein

9.2.4 Index 6020h: Slope lateral16 (read only) Sichere Neigung

Dieser Eintrag enthält die Neigung in lateraler Richtung als vorzeichenbehafteter 16Bit Wert (siehe [6] – nur bei **zweiachsigen** Geräten)

9.2.5 Index 6021h: Slope lateral16 operating parameter (read write)

Dieser Eintrag enthält die Informationen zur Interpretation des Messwerts SlopeLateral16 (siehe 9.2.3).
 Der HIT 1000 / 1500 CANopen Safety liefert standardmäßig den Wert 00h zurück (siehe [6]).

9.3 Manufacturer Specific Entries

9.3.1 Index 2001h: NodeID

Unter diesem Index ist die *Node-ID* hinterlegt.

Verwendete Subindizes:

- 0: NumberOfEntries (read only)**
- 1: CurrentNodeid (read only)**
- 2: DemandedNodeid (read write)**

Damit die neue Nodeid wirksam wird, muss erst der Befehl *StoreLssParameters* übertragen werden und danach der Knoten neu gestartet werden.

9.3.2 Index 2002h: Baudrate

Unter diesem Index ist die Baudrate hinterlegt.

Die Zuordnung des Eintrages zu der Baudrate entspricht der DS305, *Layer Setting Services and Protocols*.

Eintrag	0	1	2	3	4	5	6	7	8
Baudrate	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

Damit die neue Baudrate wirksam wird, muss erst der Befehl *StoreLssParameters* übertragen werden und danach der Knoten neu gestartet werden.

Verwendete Subindizes:

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

9.3.3 Index 4000h: Status der Neigungsmessung invertiert

Identisch zu Index 5000h, jedoch **bitweise invertiert**.

9.3.4 Index 4010h: Slope long16 (read only) Sichere Neigung invertiert

Identisch zu Index 6010h, jedoch **bitweise invertiert**.

9.3.5 Index 4020h: Slope lateral16 (read only) Sichere Neigung invertiert

Identisch zu Index 6020h, jedoch **bitweise invertiert**.

9.3.6 Index 5000h: Status der Neigungsmessung

Unter diesem Index ist der Status der Neigungsmessung hinterlegt.

Verwendete Subindizes:

- 0: Status Uint8 (read only)**

Die Codierung ist bitweise wie folgt:

Bit 7 – 3	Bit 2	Bit 1	Bit 0
Reserviert Immer 0	Schwerer Fehler	Gerät in Bewegung	Sensor-Temperatur außerhalb Regelbereich

9.3.7 Index 5130h: Skalierte Beschleunigungswerte

Unter diesem Index ist der aktuelle Prozesswert für die Komponente der Erdbeschleunigung **in Richtung der X, Y und Z-Achse** des Neigungsgebers hinterlegt.

Die Werte sind in **1/100 m/s²** skaliert.

Der Prozesswert **981** entspricht **9,81 m/s² = 1g**.

Verwendete Subindizes:

- 0: NumberOfEntries (read only)**
- 1: X-Komponente Int16 (read only)**
- 2: Y-Komponente Int16 (read only)**
- 3: Z-Komponente Int16 (read only)**

9.3.8 Index 5131h: Skalierte Drehratenwerte

Unter diesem Index ist der aktuelle Prozesswert für die Drehrate **um die X, Y und Z-Achse** des Neigungsgebers hinterlegt (im Bogenmaß).

Die Werte sind in **0,2 Millirad/s** skaliert. $2 \pi/s$ entspricht einer Umdrehung/s.

Ein Prozesswert von **15000** entspricht demnach einer Drehrate von **3 rad/s**.

Verwendete Subindizes:

- 0: NumberOfEntries (read only)**
- 1: X-Komponente Int16 (read only)**
- 2: Y-Komponente Int16 (read only)**
- 3: Z-Komponente Int16 (read only)**

Die nachfolgende Zeichnung zeigt den positiven Drehsinn um eine gedachte Achse



9.3.9 Index 5132h: Neigung im Raum (bewegungskompensiert)

Unter diesem Index ist der aktuelle Prozesswert für die kompensierte Neigung gegen die Horizontale in X, Y und Z hinterlegt.

Der Wert hat eine Auflösung von $0,01^\circ$ (vgl. Index 6010h / 6020h)

Verwendete Subindizes:

- 0: NumberOfEntries (read only)**
- 1: Entsprechend Slope long16 Komponente Int16 (read only)**
- 2: Entsprechend Slope lateral16 Komponente Int16 (read only)**

9.3.10 Weitere Indizes im Bereich 2000h bis 5FFFh (reserved)



Die Indizes in diesem Bereich enthalten **wichtige Werkseinstellungen und dürfen kundenseitig nicht verändert werden!**

Achtung: Jeder **Eingriff** kann zur **Fehlfunktion** des Neigungsgebers führen!

10 Layer setting services (LSS) und Protokolle

Die LSS-Dienste und Protokolle, dokumentiert in CiA DS305 V2.2, siehe [4], unterstützen das Abfragen und Konfigurieren verschiedener Parameter des Data Link Layers und des Application Layers eines LSS-Slaves durch ein LSS-Master über das CAN Netzwerk.

Unterstützt werden folgende Parameter:

- Node-ID
- Baudrate
- LSS-Adresse, gemäß dem Identity Objekt 1018h

Der Zugriff auf den LSS-Slave erfolgt dabei über seine LSS-Adresse, bestehend aus:

- Vendor-ID
- Produkt-Code
- Revisions-Nummer und
- Serien-Nummer

Das Mess-System unterstützt folgende Dienste:

Switch mode services

- Switch mode selective
 - einen bestimmten LSS-Slave ansprechen
- Switch mode global
 - alle LSS-Slaves ansprechen

Configuration services

- Configure Node-ID
 - Node-ID konfigurieren
- Configure bit timing parameters
 - Baudrate konfigurieren
- Activate bit timing parameters
 - Baudrate aktivieren
- Store configured parameters
 - konfigurierte Parameter speichern

Inquiry services

- Inquire LSS-address
 - LSS-Adresse anfragen
- Inquire Node-ID
 - Node-ID anfragen

Identification services

- LSS identify remote slave
 - Identifizierung von LSS-Slaves innerhalb eines bestimmten Bereichs
- LSS identify slave
 - Rückmeldung der LSS-Slaves auf das vorherige Kommando
- LSS identify non-configured remote slave
 - Identifizierung von nicht-konfigurierten LSS-Slaves, Node-ID = FFh
- LSS identify non-configured slave
 - Rückmeldung der LSS-Slaves auf das vorherige Kommando

10.1 Finite state automaton, FSA

Der FSA entspricht einer Zustandsmaschine und definiert das Verhalten eines LSS-Slaves. Gesteuert wird die Zustandsmaschine durch LSS COBs erzeugt durch einen LSS-Master, oder NMT COBs erzeugt durch einen NMT-Master, oder lokale NMT-Zustandsübergänge.

Die LSS-Modes unterstützen folgende Zustände:

- (0) Initial: Pseudo-Zustand, zeigt die Aktivierung des FSAs an
- (1) LSS waiting: Unterstützung aller Dienste wie unten angegeben
- (2) LSS configuration: Unterstützung aller Dienste wie unten angegeben
- (3) Final: Pseudo-Zustand, zeigt die Deaktivierung des FSAs an

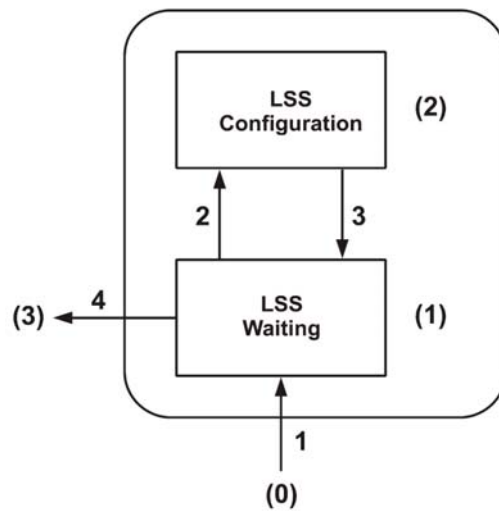


Abbildung 1: LSS-Modes

Zustandsverhalten der unterstützten Dienste

Dienste	Waiting	Configuration
Switch mode global	Ja	Ja
Switch mode selective	Ja	Nein
Activate bit timing parameters	Nein	Ja
Configure bit timing parameters	Nein	Ja
Configure Node-Id	Nein	Ja
Store configuration	Nein	Ja
Inquire LSS-address	Nein	Ja
LSS identify remote slave	Ja	Ja
LSS identify slave	Ja	Ja
LSS identify non-configured remote slave	Ja	Ja
LSS identify non-configured slave	Ja	Ja

LSS FSA Zustandsübergänge

Übergang	Ereignisse	Aktionen
1	Automatischer Übergang nach der Initialisierung beim Eintritt entweder in den NMT PRE OPERATIONAL Zustand oder NMT STOPPED Zustand, oder NMT RESET COMMUNICATION Zustand mit Node-ID = FFh.	keine
2	LSS 'switch state global' Kommando mit Parameter 'configuration_switch' oder 'switch state selective' Kommando	keine
3	LSS 'switch state global' Kommando mit Parameter 'waiting_switch'	keine
4	Automatischer Übergang, wenn eine ungültige Node-ID geändert wurde und die neue Node-ID erfolgreich im nichtflüchtigen Speicher abgelegt werden konnte UND der Zustand LSS waiting angefordert wurde.	keine

Sobald das LSS FSA weitere Zustandsübergänge im NMT FSA von NMT PRE OPERATIONAL auf NMT STOPPED und umgekehrt erfährt, führt dies nicht zum Wiedereintritt in den LSS FSA.

10.2 Übertragung von LSS-Diensten

Über die LSS-Dienste fordert der LSS-Master die einzelnen Dienste an, welche dann durch den LSS-Slave ausgeführt werden. Die Kommunikation zwischen LSS-Master und LSS-Slave wird über die implementierten LSS-Protokolle vorgenommen. Ähnlich wie bei der SDO-Übertragung, werden auch hier zwei COB-IDs für das Senden und Empfangen benutzt:

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

Tabelle 1: COB-IDs für LSS Services

10.2.1 LSS-Nachrichtenformat

Der maximal 8 Byte lange Datenbereich einer CAN-Nachricht wird von einem LSS-Dienst wie folgt belegt:

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

Tabelle 2: LSS-Nachricht

Byte 0 enthält die **Command-Specifier** (CS), danach folgen 7 Byte für die Daten.

10.3 Switch mode Protokolle

10.3.1 Switch mode global Protokoll

Das angegebene Protokoll hat den *Switch mode global service* implementiert und steuert das Zustandsverhalten des LSS-Slaves. Über den LSS-Master können alle LSS-Slaves im Netzwerk in den *Waiting Mode* oder *Configuration Mode* gebracht werden.

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

Das angegebene Protokoll hat den *Switch mode selective service* implementiert und steuert das Zustandsverhalten des LSS-Slaves. Über den LSS-Master kann nur der LSS-Slave im Netzwerk in den *Configuration Mode* gebracht werden, dessen LSS- Adressattribute der LSS-Adresse entsprechen.

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							

10.4 Configuration Protokolle

10.4.1 Configure Node-Id Protokoll

Das angegebene Protokoll hat den *Configure NMT-Address service* implementiert. Über den LSS-Master kann die Node-ID eines einzelnen LSS-Slaves im Netzwerk konfiguriert werden. Hierbei darf sich nur ein LSS-Slave im *Configuration Mode* befinden. Zur Speicherung der neuen Node-ID muss das *Store configuration protocol* an den LSS-Slave übertragen werden. Um die neue Node-ID zu aktivieren, muss der NMT-Dienst *Reset Communication* (82h) aufgerufen werden.

LSS-Master --> LSS-Slave

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

LSS-Slave --> LSS-Master

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

Error Code

- 0: Ausführung erfolgreich
- 1...254: Reserved
- 255: applikationsspezifischer Fehler aufgetreten

Specific Error

Wenn Error Code = 255 --> applikationsspezifischer Fehler aufgetreten, sonst reserviert durch die CiA

10.4.2 Configure bit timing parameters Protokoll

Das angegebene Protokoll hat den *Configure bit timing parameters service* implementiert. Über den LSS-Master kann die Baudrate eines einzelnen LSS-Slaves im Netzwerk konfiguriert werden. Zur Speicherung der neuen Baudrate muss das *Store configuration protocol* an den LSS-Slave übertragen werden.

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. Error	Reserved by CiA				
7E4h	19							

Table Selector

0: Standard CiA Baudraten-Tabelle

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: Ausführung erfolgreich
 1: selektierte Baudrate nicht unterstützt
 2...254: Reserved
 255: applikationsspezifischer Fehler aufgetreten

Specific Error

Wenn Error Code = 255 --> applikationsspezifischer Fehler aufgetreten, sonst reserviert durch die CiA

10.4.3 Activate bit timing parameters Protokoll

Das angegebene Protokoll hat den *Activate bit timing parameters service* implementiert und aktiviert die über *Configure bit timing parameters protocol* festgelegte Baudrate bei allen LSS-Slaves im Netzwerk, die sich im *Configuration Mode* befinden.

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

Der Parameter *Switch Delay* definiert die Länge zweier Verzögerungsperioden (D1, D2) mit gleicher Länge. Damit wird das Betreiben des Busses mit unterschiedlichen Baudratenparametern verhindert. Nach Ablauf der Zeit D1 und einer individuellen Verarbeitungsdauer wird die Umschaltung intern im LSS-Slave vorgenommen. Nach Ablauf der Zeit D2 meldet sich der LSS-Slave wieder mit CAN-Nachrichten und der neu eingestellten Baudrate.

Es gilt:

Switch Delay > längste vorkommende Verarbeitungsdauer eines LSS-Slaves

10.4.4 Store configuration Protokoll

Das angegebene Protokoll hat den *Store configured parameters service* implementiert. Über den LSS-Master können die konfigurierten Parameter eines einzelnen LSS-Slaves im Netzwerk in den nichtflüchtigen Speicher abgelegt werden. Hierbei darf sich nur ein LSS-Slave im *Configuration Mode* befinden.

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. Error	Reserved by CiA				
7E4h	23							

Error Code

- 0: Ausführung erfolgreich
- 1: *Store configuration* nicht unterstützt
- 2...254: Reserved
- 255: applikationsspezifischer Fehler aufgetreten

Specific Error

Wenn Error Code = 255 --> applikationsspezifischer Fehler aufgetreten, sonst reserviert durch die CiA

10.5 Inquire LSS-Address Protokolle

10.5.1 Inquire Identity Vendor-ID Protokoll

Das angegebene Protokoll hat den *Inquire LSS-Address service* implementiert. Über den LSS-Master kann die Vendor-ID eines einzelnen LSS-Slaves im Netzwerk ausgelesen werden. Hierbei darf sich nur ein LSS-Slave im *Configuration Mode* befinden.

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				

10.5.2 Inquire Identity Product-Code Protokoll

Das angegebene Protokoll hat den *Inquire LSS-Address service* implementiert. Über den LSS-Master kann der Hersteller-Gerätename eines einzelnen LSS-Slaves im Netzwerk ausgelesen werden. Hierbei darf sich nur ein LSS-Slave im *Configuration Mode* befinden.

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				

10.5.3 Inquire Identity Revision-Number Protokoll

Das angegebene Protokoll hat den *Inquire LSS-Address service* implementiert. Über den LSS-Master kann die Revisionsnummer eines einzelnen LSS-Slaves im Netzwerk ausgelesen werden. Hierbei darf sich nur ein LSS-Slave im *Configuration Mode* befinden.

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				

10.5.4 Inquire Identity Serial-Number Protokoll

Das angegebene Protokoll hat den *Inquire LSS-Address service* implementiert. Über den LSS-Master kann die Seriennummer eines einzelnen LSS-Slaves im Netzwerk ausgelesen werden. Hierbei darf sich nur ein LSS-Slave im *Configuration Mode* befinden.

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				

10.5.5 Inquire Node-ID Protokoll

Das angegebene Protokoll hat den *Inquire Node-ID service* implementiert. Über den LSS-Master kann die Node-ID eines einzelnen LSS-Slaves im Netzwerk ausgelesen werden. Hierbei darf sich nur ein LSS-Slave im *Configuration Mode* befinden.

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 und 255						

Node-ID

Entspricht der Node-ID des selektierten Gerätes. Wenn die Node-ID eben gerade erst über den *Configure Node-ID service* geändert wurde, wird die ursprüngliche Node-ID zurückgemeldet. Erst nach Ausführung des NMT-Dienstes *Reset Communication* (82h) wird die aktuelle Node-ID zurückgemeldet.

10.6 Identification Protokolle

10.6.1 LSS identify remote slave Protokoll

Das angegebene Protokoll hat den *LSS identify remote slaves service* implementiert. Über den LSS-Master können LSS-Slaves im Netzwerk in einem bestimmten Bereich identifiziert werden. Alle LSS-Slaves, die dem angegebenen Vendor-ID, Product-Code, Revision-No. – Bereich und Serial-No. – Bereich entsprechen, antworten mit dem *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				

10.6.2 LSS identify slave Protokoll

Das angegebene Protokoll hat den *LSS identify slave service* implementiert. Alle LSS-Slaves, die den im *LSS identify remote slaves protocol* angegebenen LSS-Adress-Attributen entsprechen, antworten mit diesem Protokoll.

LSS-Slave --> LSS-Master

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

10.6.3 LSS identify non-configured remote slave Protokoll

Das angegebene Protokoll hat den *LSS identify non-configured remote slave service* implementiert. Über den LSS-Master werden alle nichtkonfigurierten LSS-Slaves (Node-ID = FFh) im Netzwerk identifiziert. Die betreffenden LSS-Slaves antworten mit dem *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							

10.6.4 LSS identify non-configured slave Protokoll

Das angegebene Protokoll hat den *LSS identify non-configured slave service* implementiert. Alle LSS-Slaves, die eine ungültige Node-ID (FFh) besitzen, antworten nach Ausführung des *LSS identify non-configured slave protocol* mit diesem Protokoll.

LSS-Slave --> LSS-Master

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

11 Anschluss

Der Anschluss kann mit Hilfe der beigelegten gerätespezifischen Steckerbelegung durchgeführt werden, siehe Bedienanleitung HIT 1000 / 1500 CANopen Safety (Mat.nr. 669959, die Bedienanleitung ist Teil des Lieferumfangs HIT 1000 / 1500 CANopen).

11.1 Einschalten der Versorgungsspannung

Nachdem der Anschluss und alle Einstellungen vorgenommen worden sind, kann die Versorgungsspannung eingeschaltet werden.

Nach dem Einschalten der Versorgungsspannung und Beendigung der Initialisierung geht das Mess-System in den Vor-Betriebszustand (PRE-OPERATIONAL). Dieser Zustand wird durch die Boot-Up-Meldung „**COB-ID 700h+Node ID**“ bestätigt. Falls das Mess-System einen internen Fehler erkennt, wird eine Emergency-Meldung mit dem Fehlercode übertragen (siehe 6.5).

Im PRE-OPERATIONAL-Zustand ist zunächst nur eine Parametrierung über Service-Daten-Objekte möglich. Es ist aber möglich, PDOs unter Nutzung von SDOs zu konfigurieren. Ist das Mess-System in den Zustand OPERATIONAL überführt worden, ist auch eine Übertragung von PDOs möglich (siehe 6.7).

11.2 Einstellen der Node-ID und Baudrate mittels LSS-Dienste

11.2.1 Konfiguration der Node-ID, Ablauf

Annahme:

- LSS-Adresse unbekannt
- der LSS-Slave ist der einzige Teilnehmer in Netzwerk
- es soll die Node-ID 12 dez. eingestellt werden

Vorgehensweise:

- LSS-Slave mit dem Command Specifier 04 *Switch mode global protocol*, Mode = 1 in den *Configuration Mode* bringen. (siehe 10.3)
- Command Specifier 17 *Configure NMT-Address protocol*, Node-ID = 12 ausführen. (siehe 10.4.1)
--> Rückmeldung abwarten und erfolgreiche Ausführung überprüfen,
--> Error Code = 0.

- Command Specifier 23 *Store configuration protocol* ausführen.
--> Rückmeldung abwarten und erfolgreiche Ausführung überprüfen,
--> Error Code = 0.
(siehe 10.4.4)
- Versorgungsspannung des LSS-Slaves aus-, danach wieder einschalten. Die neue Konfiguration ist jetzt aktiv.

11.2.2 Konfiguration der Baudrate, Ablauf

Annahme:

- LSS-Adresse unbekannt
- der LSS-Slave ist der einzige Teilnehmer in Netzwerk
- es soll die Baudrate 125 kbit/s eingestellt werden

Vorgehensweise:

- NMT- Command Specifier *Stop Remote Node* (02h) aufrufen, um den LSS-Slave in den *Stopped state* zu bringen. Der LSS-Slave sollte keine CAN-Nachrichten mehr senden --> Heartbeat abgeschaltet (siehe 6.7).
- LSS-Slave mit dem Command Specifier *Switch mode global protocol*, Mode = 1 in den *Configuration Mode* bringen. (siehe 10.3.1).
- Command Specifier 19 *Configure bit timing parameters protocol* ausführen, Table Selector = 0, Table Index = 4
--> Rückmeldung abwarten und erfolgreiche Ausführung überprüfen,
--> Error Code = 0
(siehe 10.4.2).
- Command Specifier 21 *Activate bit timing parameters protocol* aufrufen, damit die neue Baudrate aktiv wird. (siehe 10.4.3).
- Command Specifier 23 *Store configuration protocol* ausführen.
--> Rückmeldung abwarten und erfolgreiche Ausführung überprüfen,
--> Error Code = 0.
(siehe 10.4.4).
- Versorgungsspannung des LSS-Slaves aus-, danach wieder einschalten. Die neue Konfiguration ist jetzt aktiv.

12 Inbetriebnahme

12.1 CAN – Schnittstelle

Die CAN-Bus-Schnittstelle ist durch die internationale Norm ISO/DIS 11898 definiert und spezifiziert die zwei untersten Schichten des CAN Referenz-Models.

Die Konvertierung der Mess-System-Information in das CAN-Protokoll (CAN 2.0A) geschieht über einen CAN-Kontroller. Die Funktion des CAN-Kontrollers wird durch einen Watchdog überwacht.

Das CANopen Kommunikationsprofil (CiA Standard DS301, siehe [1]) basiert auf dem CAN Application Layer (CAL) und beschreibt, wie die Dienste von Geräten benutzt werden. Das CANopen Profil erlaubt die Definition von Geräteprofilen für eine dezentralisierte E/A.

Das Mess-System mit CANopen Protokoll unterstützt das Geräteprofil für Inclinometer (CiA Draft Standard 410, Version 1.3.0, siehe [6]).

Die Kommunikations-Funktionalität und Objekte, welche im Messgeräteprofil benutzt werden, werden in einer EDS-Datei (Electronic Data Sheet) beschrieben. Wird ein CANopen Konfigurations-Hilfsprogramm benutzt (z.B. CANSETTER), kann der Benutzer die Objekte (SDO's) des Mess-Systems auslesen und die Funktionalität programmieren.

12.2 EDS-Datei

Die EDS-Datei (elektronisches Datenblatt) enthält alle Informationen über die Mess-System-spezifischen Parameter sowie Betriebsarten des Mess-Systems. Die EDS-Datei wird durch das CANopen-Netzwerkkonfigurationswerkzeug eingebunden, um das Mess-System ordnungsgemäß zu konfigurieren bzw. in Betrieb nehmen zu können.

Die EDS-Datei die zum Gerät passt, kann am Gerätenamen und der dem Gerät entsprechenden Major-Revisionsnummer des Gerätes im Dateinamen erkannt werden

Die Datei inklusive der vorliegenden CANopen-Schnittstellenbeschreibung befindet sich zum Download auf unserer Homepage unter

<http://www.hydac.com/de-de/produkte/sensorik/show/Material/index.html>

HYDAC ELECTRONIC GMBH

Hauptstr. 27
D-66128 Saarbrücken
Germany

Web: www.hydac.com
E-Mail: electronic@hydac.com
Tel.: +49 (0)6897 509-01
Fax.: +49 (0)6897 509-1726

HYDAC Service

Für Fragen zu Reparaturen steht Ihnen der HYDAC Service zur Verfügung.

HYDAC SERVICE GMBH

Hauptstr. 27
D-66128 Saarbrücken
Germany

Tel.: +49 (0)6897 509-1936
Fax.: +49 (0)6897 509-1933

Anmerkung

Die Angaben in dieser Bedienungsanleitung beziehen sich auf die beschriebenen Betriebsbedingungen und Einsatzfälle. Bei abweichenden Einsatzfällen und/oder Betriebsbedingungen wenden Sie sich bitte an die entsprechende Fachabteilung.

Bei technischen Fragen, Hinweisen oder Störungen nehmen Sie bitte Kontakt mit Ihrer HYDAC-Vertretung auf.

Protocol Description CANopen Safety

HIT 1000 / HIT 1500

(Translation of original instructions)



Table of Contents

1	Introduction	6
2	Functions of HIT 1000 / 1500 CANopen Safety.....	6
3	Transmission rates	7
4	CAN Frames.....	7
5	Node ID	7
6	Transmission services.....	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.....	12
6.7	Network Management Services (NMT)	12
6.8	Boot Up Protocol.....	13
7	Data flow in the HIT 1000 / 1500 CANopen	14
7.1	Sensor Unit.....	14
7.2	Calibration & Scaling	14
7.3	Filter & Fusion	14
7.4	Transmission Unit.....	14
8	The Object Dictionary	15
8.1	Set-up of the Object Dictionary	15
8.2	Structure of the device-specific part according to DS410.....	15
9	Entries in the Object Dictionary	16
9.1	Communication Profile Specific Entries (DS301)	16
9.1.1	Index 1000h: DeviceType (read only)	16
9.1.2	Index 1001h: ErrorRegister (read only)	16
9.1.3	Index 1003h: Pre-defined error field (read only).....	16
9.1.4	Index 1005h: SyncMessageIdentifier (read write)	16
9.1.5	Index 1008h: ManufacturerDeviceName (const)	16
9.1.6	Index 1009h: ManufacturerHardwareVersion (const).....	16
9.1.7	Index 100Ah: ManufacturerSoftwareVersion (const).....	16
9.1.8	Index 1010h: StoreParameters	16
9.1.9	Index 1011h: RestoreDefaultParameters	17
9.1.10	Index 1014h: CobIdEmergencyMessage (read write)	17
9.1.11	Index 1017h: ProducerHeartbeatTime (read write)	17
9.1.12	Index 1018h IdentityObject	17
9.1.13	Index 1029h: Error behaviour	18

9.1.14 Index 1301h: SRDO communication parameter	18
9.1.15 Index 1381h: SRDO mapping parameter	18
9.1.16 Index 13FEh: Configuration Valid (read write).....	19
9.1.17 Index 13FFh: Safety Configuration Checksum.....	19
9.1.18 Index 1800h / 1801h / 1802h / 1803h: TPDO communication parameter	19
9.1.19 Index 1A00h / 1A01h / 1A02h / 1A03h: TPDO mapping parameter.....	20
9.1.20 Index 1F80h: NMT Startup (read / write).....	22
9.2 Device Profile Specific Entries (DS410)	22
9.2.1 Index 6000h: Resolution (read only)	22
9.2.2 Index 6010h: Slope long16 (read only) Safe inclination	22
9.2.3 Index 6011h: Slope long16 operating parameter (read write).....	22
9.2.4 Index 6020h: Slope lateral16 (read only) Safe inclination.....	22
9.2.5 Index 6021h: Slope lateral16 operating parameter (read write).....	22
9.3 Manufacturer Specific Entries	23
9.3.1 Index 2001h: NodeID	23
9.3.2 Index 2002h: Baudrate	23
9.3.3 Index 4000h: Status of the inclination measurement inverted	23
9.3.4 Index 4010h: Slope long16 (read only) Safe inclination inverted	23
9.3.5 Index 4020h: Slope lateral16 (read only) Safe inclination inverted	23
9.3.6 Index 5000h: Status of the inclination measurement.....	23
9.3.7 Index 5130h: Scaled acceleration values	24
9.3.8 Index 5131h: Scaled rotational values	24
9.3.9 Index 5132h Inclination in space (motion compensated).....	24
9.3.10 Further indices within the range of 2000h to 5FFFh (reserved).....	25
10 Layer setting services (LSS) and protocols	26
10.1 Finite state automaton, FSA.....	27
10.2 Transmission of LSS services.....	28
10.2.1 LSS message format	28
10.3 Switch mode protocols.....	29
10.3.1 Switch mode global protocol	29
10.3.2 Switch mode selective protocol.....	29
10.4 Configuration protocols.....	30
10.4.1 Configure Node ID protocol	30
10.4.2 Configure bit timing parameters protocol	30
10.4.3 Activate bit timing parameters protocol	31
10.4.4 Store configuration protocol	32
10.5 Inquire LSS address protocols.....	32
10.5.1 Inquire Identity Vendor ID protocol.....	32
10.5.2 Inquire Identity Product Code Protocol.....	33
10.5.3 Inquire Identity Revision-Number protocol	33
10.5.4 Inquire Identity Serial Number protocol	34
10.5.5 Inquire Node ID protocol	34
10.6 Identification Protocols.....	35
10.6.1 LSS identify remote slave protocol.....	35
10.6.2 LSS identify slave protocol.....	36
10.6.3 LSS identify non-configured remote slave protocol	36
10.6.4 LSS identify non-configured slave Protocol.....	36
11 Connection	37

11.1 Switching on the supply voltage.....	37
11.2 Setting the Node ID and Baud rate by means of LSS services.....	37
11.2.1 Configuration of the Node ID, sequence	37
11.2.2 Configuration of the Baud rate, sequence.....	38
12 Commissioning	39
12.1 CAN interface	39
12.2 EDS file	39

Preface

This manual provides you, as user of our product, with key information on the operation and maintenance of the equipment.

It will acquaint you with the product and assist you in obtaining maximum benefit in the applications for which it is designed.

Keep the manual in the vicinity of the instrument for immediate reference. Please note that the data on the software technology provided in this manual refers to that available at the time of publication.

If you discover errors while reading the documentation or have additional suggestions or tips, please contact us at:

HYDAC ELECTRONIC GMBH
Technical Documentation
Hauptstrasse 27
66128 Saarbruecken
-Germany-
Phone: +49(0)6897 / 509-01
Fax: +49(0)6897 / 509-1726
Email: electronic@hydac.com

We look forward to receiving your input.

“Putting experience into practice”

1 Introduction

The inclinometer HIT 1000/1500 CANopen Safety series meet the CANopen standards according to the following profiles and standards:

- [1] CiA DS301, Version: 4.2.0 (21 February 2011)
CANopen application layer and communication profile
- [2] CiA DS302, Version: 4.1 (02 February 2009)
Additional application layer functions - Part 2: Network management
- [3] CiA DS303, Version: 1.4 (14th August 2006)
Additional specification - Part 2: Representation of SI units and prefix
- [4] CiA DS304
Version: 1.0.1, (1st January 2005)
Framework for safety-relevant communication
- [5] CiA DS305, Version: 2.2 (26th August 2008)
Layer setting services (LSS) and protocols
- [6] CiA DS410, Version 1.3.0 (23rd February 2010)
Device profile for inclinometer

This manual describes the functions supported by the HIT 1000 / 1500 CANopen Safety. A basic knowledge of CAN and CANopen and CANopen Safety is assumed. The exact function is described in the a.m. *Draft Standards*. Since both specifications are issued in English, the features described in this manual are identified using the English description from the specification and are shown in *italics*, for clarity.

2 Functions of HIT 1000 / 1500 CANopen Safety

- **Inclination calculation** based on the acceleration components on 3 spatial axes with **low pass function** for HIT 1000 and HIT 1500 (statical inclination measurement = **safe inclination**)
- **Inclination calculation** based by **the fusion** of acceleration components and turn rate components for **HIT 1500** (motion compensated inclination measurement, highly dynamic = **unsafe inclination**)
- Conversion of the inclination values into **Euler angle** (unit °, single or double axis)
- Transmission of the actual inclination value as a **PDO or SRDO** for the following events:
 - **Synchronously** in relation to received SYNC objects (PDO only)
 - **Asynchronously**, cyclically, within the range from 5 millisecond to >1 minute

3 Transmission rates

HIT 1000 / 1500 CANopen Safety supports the following transmission rates (Baud rates):

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

The timing complies with DS301, *Bit rates and timing*.

The transmission rate used is stored in a non-volatile memory.

When supplied, it is set to **250 kbit/s** and can be changed via the CANbus (see *Object Dictionary Index 2002h*).

4 CAN Frames

The HIT 1000 / 1500 CANopen Safety supports the 11 bit base frames with 11 bit identifiers, as well as 29 bit frames with 29 bit identifiers, required in the specification.

5 Node ID

To operate the HIT 1000 / 1500 CANopen Safety in a CANopen network a unique *Node ID* must be set within the network.

The set *Node ID* is stored in a non-volatile memory and can be adjusted via the CAN bus (see *Object Dictionary Index 2001h*).
When supplied, the address **1** is set.

6 Transmission services

6.1 Service Data Object (SDO)

With CANopen, all the device's data (setting parameters and measured data) are filed in an *Object Dictionary* under a specified *Index*. Various entries of the *Object Dictionary* are sub-divided still further using a *sub index*. Using the *SDOs*, other network nodes can read from or write to the *Object Dictionary* of the HIT 1000 / 1500 CANopen Safety. HIT 1000 / 1500 CANopen Safety takes on the role of a *Server*, and the device intending to read or write the data, takes on the role of a *Client*.

To transmit data the HIT 1000 / 1500 CANopen Safety must have a *Receive-SDO* serving for the reception of data and a *Transmit-SDO* which serves for sending data.
Sequence of the data transmission:

Reading from Object Dictionary:

1. A device (*Client*) sends the Receive-SDO of the HIT 1000 / 1500 CANopen Safety (*Server*).
This *SDO* contains an identification to say that the *Object Dictionary* is to be read, as well as the *Index* and *sub index*.
2. The HIT 1000 / 1500 CANopen Safety (*Server*) sends its Transmit-SDO. The *Index* and the *sub index*, as well as the read data are also included herein.

Writing to the Object Dictionary:

1. A device (*Client*) sends the Receive-SDO of the HIT 1000 / 1500 CANopen Safety (*Server*).
This *SDO* contains an identification, to say that the *Object Dictionary* is to be written to, as well as the required index, sub index and the data to be entered.
2. The HIT 1000 / 1500 CANopen Safety (*Server*) sends its Transmit-SDO. This *SDO* also contains the index and the sub index, as well as an identification to say the *Object Dictionary* has been written to.

If an error should occur, e.g. the specified *Index* does not exist, or an attempt is made to write to a *read only* entry, or the data is not within the valid range, then the Transmit-SDO receives an appropriate *Abort SDO Transfer* identification and a corresponding *Abort Code* (see [1]).

The particular *COB-ID* of the *SDO* corresponds to the *Pre-defined Connection Set* defined in the DS301 and cannot be altered.

COB IDs for 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)

Data transmission using *SDOs* is indeed very flexible, but has some disadvantages when transmitting measured values or actuating variables: only one piece of data can be read, the data must first be requested with an *SDO* and because the relevant *Index* and *Sub index* is also transmitted, what is known as the overhead increases further.

For this reason CANopen Safety defines what is known as *Process Data Objects*. These contain only the necessary useful data. There are two types of *PDO*:

1. *Transmit PDOs*
These enable the measuring instrument to send its measured values.
2. *Receive PDOs*
These enable the actuating variables to be transmitted to an actuator or a controller.

What is known as the *PDO Mapping* stipulates which data is now in a *PDO*. This *PDO Mapping* is deposited in the *Object Dictionary* (siehe *Object Dictionary*, Index 1A00h, 1A01h, 1A02h, 1A03h).

The *PDO Transmission Type* stipulates with which ID and for which event a *PDO* is transmitted. These settings are also deposited in the *Object Dictionary* (siehe *Object Dictionary*, Index 1800h, 1801h, 1802h, 1803h).

Events which result in a PDO being sent:

1. Receipt of a *SYNC* object (synchronous transmission).
2. Expiry of an adjustable cycle time within the range from 5 milliseconds to >1 minute (cyclical transmission).

HIT 1000 / 1500 CANopen Safety implements more *Transmit-PDOs*, which include the actual sensor or process values.

The DS410 stipulates the transmission of the actual measured value as a 16 bit value or an 32 bit value, as the default setting.

HIT 1000 / 1500 CANopen Safety only implements the transmission as a 16 bit value.

6.3 Synchronisation Object (SYNC)

SYNC objects are used to implement a synchronous data transmission. A *SYNC* object is in fact a CAN message with a defined identifier, without data. CANopen differentiates between *SYNC Producers* and *SYNC Consumers*. *SYNC Producers* are devices on the bus which send a *SYNC* at adjustable time intervals. *SYNC Consumers* are devices which react to receiving a *SYNC*. In a CANopen network several *SYNC* objects can exist. The individual *SYNC* objects are differentiated by means of the *SYNC ID* which corresponds to the CAN identifier used. The *SYNC ID* used is stored in the *Object Dictionary*.

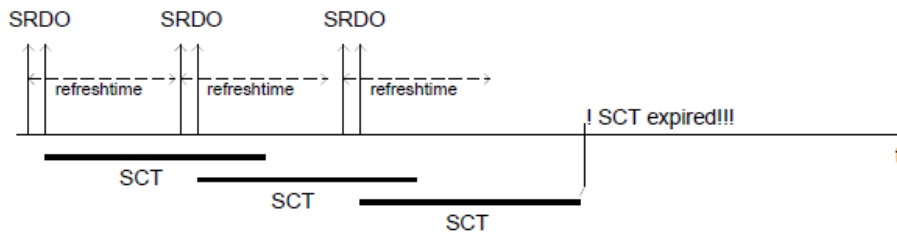
The HIT 1000 / 1500 CANopen Safety provides the functionality of a *SYNC Consumer*. If the *PDO Transmission Type* is set appropriately, a *PDO* is sent on receipt of a *SYNC*. The *SYNC-ID* is pre-set to 80h and can be changed in the *Object Dictionary* (see *Object Dictionary*, Index 1005h). In *PDO Transmission Type* the number of received *SYNC* objects which result in a *PDO* being sent, can be set.

6.4 Safety-relevant Data Object (SRDO)

The data transmission via what is known as *Safety-relevant Data Objects* ensures a safe transfer of the measured values. *SRDOs* can be used or deactivated in order to send or receive data.

SRDOs consist of two autonomous CAN frames. The first frame has data in normal display, the second CAN frame has the data in inverted display. It enables the receiver to verify the data.

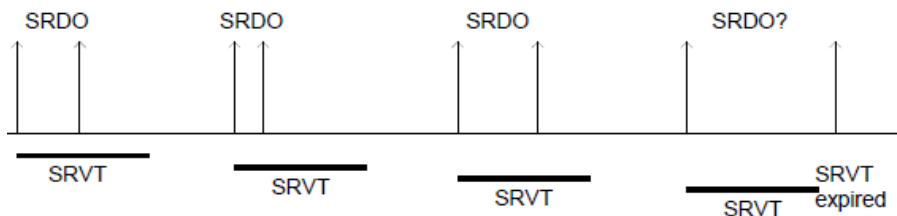
The time between two SRDOs is called *Refresh Time* or *Safeguard cycle time (SCT)* and serves for timeout monitoring between two SRDOs.



Source: CiA Draft Standard 304, CANopen Framework for safety-relevant communication

Version: 1.0.1, January 2005, CAN in Automation (CiA), S. 9

The time between the two frames is called *Safety-relevant object validation time (SRVT)* and enables the receiver to have a timeout recognition within one SRDO.



Source: CiA Draft Standard 304, CANopen Framework for safety-relevant communication,

Version: 1.0.1, January 2005, CAN in Automation (CiA), S. 10

What is known as the *SRDO Mapping* stipulates which data is now in a SRDO. This mapping is stored in the *Object Dictionary*.

The firmware supports one Transmit-SRDO and no Receive-SRDOs.

The following items should be observed during configuration:

- The modification of the communication parameters is possible only in *Pre-operational status*.
- To modify the COB-IDs, the corresponding *SRDO* must first be deactivated (*Information Direction* ist 0).
- The *SCT* on the receiver node must be chosen accordingly higher than on the sender node in order to avoid synchronisation problems due to software runtimes (± 2.8 ms).
- The *SRVT* is to be chosen in such a way that an interruption of the *SRDOs* on the CANbus can not lead to a failure at the receiver due to higher valued messages.
- The *SCT* and *SRVT* are as well highly depending on the bit rate.
- After being modified, the checksum must be calculated anew for each *SRDO* accordingly and then stored in the *Object Dictionary* (see *Index 13FFh*).
- After having modified the configuration a read-back must be carried out for verification.
- After verification of the data, the configuration can be defined as valid (see *Object Dictionary*, *Index 13FE*, *Configuration Valid*).

The following items are to be observed at the plausibility test of the configuration:

- The SRVT must be lower than the SCT.
- The COB-ID in the message using the normal data must always be an odd number.
- The COB-ID in the message using the inverted data must always be an even number.

Further plausibility tests are not carried out. For the significance of the set parameters, the designer of the total system is responsible (see COB-ID Bereich DS304, see [4]).

The plausibility test is carried out when writing the value A5h to *Configuration Valid*. The writing is successful only if the test has been successfully completed. The plausibility test is done for all the SRDOs activated (Information Direction unequal with 0).



Important note:

The SRDOs are sent out only if the configuration is valid.

6.5 Emergency Object (EMCY)

EMCY objects are sent if an error occurs. *EMCY* objects contain an *Emergency Error Code*, the contents of an *Error register* as well as a *Manufacturer specific Error Field*. If a notified error is eliminated or disappears, this is also notified by a special *EMCY* object.

An emergency message will be sent, if an error occurs or if this error disappears. The message is structured as follows:

Error	Emergency ErrorCode	Manufacturer SpecificErrorField	Error Category
No Error	0000h	0000h	Error eliminated
Error while loading User Set-up	FF00h	0001h	Manufacturer-specific error (bit 7)
Controller error	FF00h	0002h	Manufacturer-specific error (bit 7)
Reserved	N/A	0003h	N/A
Reserved	N/A	0004h	N/A
CAN Error passive	8120h	0005h	Communication error (bit 4)
Recovered from Bus-off	8140h	0006h	Communication error (bit 4)

The *EMCY* object has the pre-set ID $80h + \text{Node ID}$ and can be changed in the *Object Dictionary* (see *Object Dictionary, Index 1014h*).

6.6 Heartbeat

Using the *Heartbeat Protocol* the individual nodes can be monitored. CANopen differentiates between the following functions:

1. *Heartbeat Producer*
sends a *Heartbeat* object in cyclical intervals.
2. *Heartbeat Consumer*
monitors the sending of certain *Heartbeat* objects.

The cycle time can be adjusted in the *Object Dictionary* to milliseconds. If a time interval of 0 is specified, this means "*Heartbeat* not active".

With the *Heartbeat* object the status of the *Heartbeat Producers* is also transmitted in the form of bytes.

Meaning of the Heartbeat object contents

Value	Status	NOTE
0	<i>BOOTUP</i>	The device has booted up.
4	<i>STOPPED</i>	The device has been stopped.
5	<i>OPERATIONAL</i>	The device is working normally.
127	<i>PRE-OPERATIONAL</i>	The device does not send any <i>PDOs</i> , but can modify <i>SDOs</i> .

HIT 1000 / 1500 CANopen Safety can operate as a *Heartbeat Producer*. The ID of the *Heartbeat* is 700h + *Node-ID*. The time has been pre-set to 0 (not active) and can be modified (see *Object Dictionary, Index 1017h*).

6.7 Network Management Services (NMT)

NMT objects are used to start, stop or reset devices. CANopen differentiates between the following functionalities:

1. *NMT Master*
controls other nodes.
2. *NMT Slave*
is controlled by a *Master*.

In a CANopen network only one *NMT* object exists with the Identifier 0. Two bytes are always transmitted. The first byte contains the *Command Specifier*, which represents the command, the second byte contains the *Node ID* of the node which carries out this command. A value of 0 indicates that this command is valid for all nodes. The following commands are possible:

NMT commands

1. **Start Remote Node**
The node changes to the *Operational* condition.
2. **Stop Remote Node**
The node changes to the *Stopped* condition.
3. **Enter Pre-Operational**
The node changes to the *Pre-Operational* condition.

4. **Reset Node**

The "Device Profile Specific" OD range is reset, the Baud rate is, if necessary, re-initialised and then changes to the Reset Communication condition.

5. **Reset Communication**

The communication unit of the node is reset and then the node changes to the *Pre-Operational* condition.

HIT 1000 / 1500 CANopen Safety operates as a *NMT Slave* and supports all the *NMT* services.

6.8 Boot Up Protocol

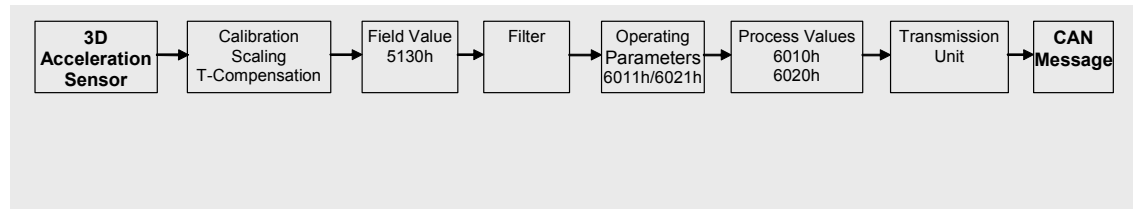
If a *NMT Slave* changes to the *Pre-operational* condition after initialisation, it sends a *Boot Up* object in each case. In principle this is no different to a *Heartbeat* object with the status 0.

In the HIT 1000 / 1500 CANopen Safety this function is implemented.

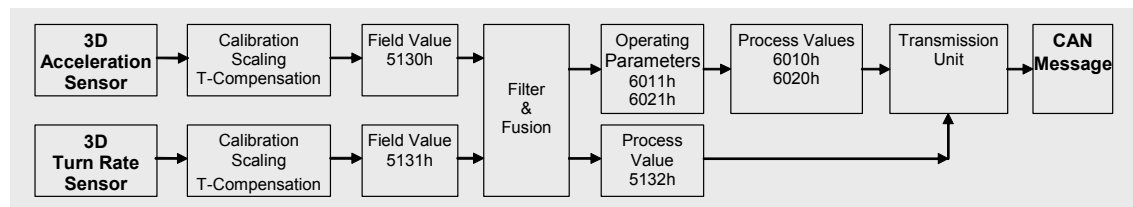
7 Data flow in the HIT 1000 / 1500 CANopen

The following figures show the flow of data within the HIT 1000 / 1500 CANopen, and the relevant indices of the *Object Dictionary*.

Data flow in the HIT 1000 CANopen



Data flow in the HIT 1500 CANopen



7.1 Sensor Unit

The sensor unit measures the inclination towards the horizontal plane and the gyro. The measured values are made digitally available. The type of the sensor unit (*Inclination Transducer*) is stored under *the Sensor Type*.

7.2 Calibration & Scaling

The sensor values are processed in the *Calibration & Scaling* menu point and made available as a *FieldValue*.

7.3 Filter & Fusion

The calibrated sensor values are filtered (HIT 1000 und HIT 1500) and merged (only HIT 1500) in the *Filter & Fusion* menu point and made available after the operating parameters unit as a *ProcessValue*.

7.4 Transmission Unit

If one of the following events occurs, then depending on the preset *TransmissionType*, the value of the PDO is sent.

1. The *EventTimer* has expired (cyclical transmission).
2. One or more *Sync* objects have been received (synchronous transmission).

8 The Object Dictionary

8.1 Set-up of the Object Dictionary

As already mentioned, all the data are stored in the *Object Dictionary*. The entries supported by the HIT 1000 / 1500 CANopen Safety are listed in the following chapters. The index is always shown in hexadecimal notation, according to the specification, without the hexadecimal representation being shown extra. For each entry, the access type must also be specified. CANopen differentiates between the following Access Types:

Access Types for the Object Dictionary

1. const
Read-only, and always delivers the same value.
2. read only
Read-only, however, the value can be changed during operation.
3. write only
The entry is write-only.
4. read write
The entry can be read and written.

CANopen differentiates between the following areas of the data dictionary:

Areas of the Object Dictionary:

1. Index 0000h .. 1FFFh: Communication profile specific entries
Settings which apply to all CANopen devices. These entries are defined in DS301 and DS302.
2. Index 6000h .. 9FFFh: Device profile specific entries
Device-specific data which are defined in a Draft Standard. HIT 1000 / 1500 CANopen has implemented the device profile DS410.
3. Index 2000h .. 5FFFh: Manufacturer-specific entries
Manufacturer-specific, additional data which are not defined in any specification.

8.2 Structure of the device-specific part according to DS410

The HIT 1000 / 1500 CANopen Safety implements DS410. This describes the behaviour and the functionality of *inclinometers*.

9 Entries in the Object Dictionary

Listed below are the functionalities implemented by the HIT 1000 / 1500 CANopen Safety. A detailed description of the entries can be found in profiles [1] and [6].

9.1 Communication Profile Specific Entries (DS301)

9.1.1 Index 1000h: DeviceType (read only)

Contains the number of the device profile being used, in this case, number **410**, as well as the profile-specific extension, in this case 2, for the support of 2 planes with 16 bit maximum resolution.

9.1.2 Index 1001h: ErrorRegister (read only)

Contains the actual error condition (see *EMCY*, and [1]).

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

Errors are made available, which have occurred in the CANopen device and which were signalled via *emergency object* (see CiA 301)

9.1.4 Index 1005h: SyncMessageIdentifier (read write)

Use this to adjust the *COB ID* for the *SYNC* object.

9.1.5 Index 1008h: ManufacturerDeviceName (const)

Provides the device name as a character string. ("HIT1000").

9.1.6 Index 1009h: ManufacturerHardwareVersion (const)

Provides the hardware version as a character string (e.g. "01.03").

9.1.7 Index 100Ah: ManufacturerSoftwareVersion (const)

Provides the software version as a character string (e.g. "03.02"). The first two characters indicate the version, the last two characters indicate the revision status.

9.1.8 Index 1010h: StoreParameters

By entering the character string "save" (65766173h) the current settings are transferred to the non-volatile memory.

HIT 1000 / 1500 CANopen Safety does not automatically store settings if modified, modifications are stored only when requested.



WARNING: Any changed settings must be saved explicitly using *StoreParameters*, otherwise they will be lost when the instrument is switched off or when the *NMT* commands *Reset Node* and *Reset Communication* are carried out.

CANopen has the option of restoring different parameter areas with the help of various *Sub indeces*.

The sub indeces 1, 2, 3 and 4 are supported.

For further information, please see profile [1].



Warning: The Baud rate and NodeID settings remain unchanged with *StoreAllParameters*.

Sub indices used:

- 0: Largestsub indexSupported (read only)
- 1: StoreAllParameters (read write)
- 2: StoreCommunicationParameters (read write) (index of 1000h to 1FFFh)
- 3: StoreApplicationParameters (read write) (index of 6000h to 9FFFh)
- 4: StoreLssParameters (read write) (index von 2000h to 20FFh)

9.1.9 Index 1011h: RestoreDefaultParameters

By entering the character string "load" (64616F6Ch) the factory settings are transferred into the non-volatile memory.

However, the HIT 1000 / 1500 CANopen Safety goes on working with the actual settings until switched off or until the commands *Reset Node* and *Reset Communication* are performed.

CANopen has the option of restoring different parameter areas with the help of various *Sub indeces*.

The sub indeces 1, 2, 3 and 4 are supported.

For further information, please see profile [1].



Warning: The Baud rate and NodeID settings remain unchanged with *RestoreAllParameters*.

Sub indices used:

- 0: Largestsub indexSupported (read only)
- 1: RestoreAllParameters (read write)
- 2: RestoreCommunicationParameters (read write) (index of 1000h to 1FFFh)
- 3: RestoreApplicationParameters (read write) (index of 6000h to 9FFFh)
- 4: RestoreLssParameters (read write) (index of 2000h to 20FFh)

9.1.10 Index 1014h: CobIdEmergencyMessage (read write)

Use this to adjust the *COB ID* for the *EMCY* object (see *EMCY*).

9.1.11 Index 1017h: ProducerHeartbeatTime (read write)

Use this to adjust the *Heartbeat* time to milliseconds. The value 0 indicates that this function is not active (see *Heartbeat*).

9.1.12 Index 1018h IdentityObject

The Identity Object identifies the HIT 1000 / 1500 CANopen. The identification consists of four 32bit numbers. The combination of these 4 numbers produces a device ID which is unique worldwide.

Sub indices used:

- 0: Largestsub indexSupported (read only)**
- 1: VendorID (read only)**
Unique manufacturer code (DAh for HYDAC ELECTRONIC GmbH)
- 2: ProductCode (read only)**
Hydac Electronic product code (z.B: 924854)
- 3: RevisionNumber (read only)**
Revision number of the device
- 4: SerialNumber (read only)**
Serial number of the device

9.1.13 Index 1029h: Error behaviour**Sub indices used:**

- 0: No. 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

9.1.14 Index 1301h: SRDO communication parameter

Contains the communication settings of the first safety relevant data object.

Sub indices used:

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

9.1.15 Index 1381h: SRDO mapping parameter

Contains the mapping settings of the first safety relevant data object.

Sub indices used:

- 0: Largestsub indexSupported (read only)**
- 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)

When supplied, the HIT 1000 / 1500 CANopen Safety has the following entries:

Index 1381h static (safe) inclination (HIT 1000 and HIT 1500)		
sub index	Content	Meaning
0	6	Six values are transferred in the <i>SRDO</i> .
1	60100010h	The first value in the <i>SRDO</i> is the value of <i>Index 6010h</i> , <i>sub index 0</i> with a width of 10h (=16 bits)
2	40100010h	The second value in the <i>SRDO</i> is the value of <i>Index 4010h</i> , <i>sub index 0</i> with a width of 10h (=16 bits)
3	60200010h	The third value in the <i>SRDO</i> is the value of <i>Index 6020h</i> , <i>sub index 0</i> with a width of 10h (=16 bits)
4	40200010h	The fourth value in the <i>SRDO</i> is the value of <i>Index 4020h</i> , <i>sub index 0</i> with a width of 10h (=16 bits)
5	50000008h	The fifth value in the <i>SRDO</i> is the value of <i>Index 5000h</i> , <i>sub index 0</i> with a width of 08h (=8 bits)
6	40000008h	The sixth value in the <i>PDO</i> is the value of <i>Index 4000h</i> , <i>sub index 0</i> with a width of 08h (=8 bits)

9.1.16 Index 13FEh: Configuration Valid (read write)

This entry does not serve for validation of the *SRDO* configuration. A plausibility test is triggered when writing the value A5h. If the test is successful the value is accepted. Otherwise, the value is set to 0.

9.1.17 Index 13FFh: Safety Configuration Checksum

Contains the checksums of the *SRDO* configurations.

Sub indices used:

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

9.1.18 Index 1800h / 1801h / 1802h / 1803h: TPDO communication parameter

These entries determine the *PDO* transmission. In detail these are:

Parameters for *PDO* transmission

1. *COB ID*
Determines the identifier for the *PDO*. The highest value bit (Bit31) of the entry no longer belongs to the ID and has the meaning "*disable PDO*". If this bit is set, then transmission of the *PDO* is disabled.

2. *Transmission Type*
Determines the transmission type.
Values between 0 and 240 mean synchronous transmission. The figure represents the number of SYNC objects which have to be received before the PDO is sent. The value 254 indicates a manufacturer-specific transmission and the value 255 a device-profile-specific transmission. With 254 and 255 the PDO is sent cyclically, providing a time (*Event Time*) other than 0 is set.
3. *Inhibit Time*
defines the minimum interval for the PDO if FEh and FFh were chosen as transmission type. The value is defined as a multiple of 100µs. The value 0 switches off the *inhibit time*.
4. *Reserved – of no significance*
5. *Event Time*
Determines the cycle time for asynchronous transmissions for Transmission Types 254 and 255 in milliseconds. The value 0 denotes no time-controlled transmission.

Sub indices used:

- 0: **Largestsub indexSupported (read only)**
- 1: **COBIDUsedByPDO (read write)**
- 2: **TransmissionType (read write)**
- 3: **InhibitTime (read write)**
- 4: **Reserved**
- 5: **EventTimer(read write)**

9.1.19 Index 1A00h / 1A01h / 1A02h / 1A03h: TPDO mapping parameter

Use these entries to determine which data is transferred with the PDO. Sub index 0 indicates the amount of data in the PDO. The index and sub index as well as the number of bits of the first date are stored in sub index 1, this also applies for the sub indices 2 to 8.

When supplied, the HIT 1000 / 1500 CANopen Safety has the following entries:

Index 1A00h Strategical Inclination (HIT 1000 und HIT 1500)		
-- Deactivated by TPDO communication parameter --		
sub index	Content	Meaning
0	3	Two values are transferred in the <i>PDO</i> .
1	60100010h	The first value in the <i>PDO</i> is the value of <i>Index 6010h</i> , <i>sub index 0</i> with a width of 10h (=16 bits)
2	60200010h	The second value in the <i>PDO</i> is the value of <i>Index 6020h</i> , <i>sub index 0</i> with a width of 10h (=16 bits)
3	50000008h	The third value in the <i>PDO</i> is the status of the inclination measurement

Index 1A01h: Scaled acceleration values (HIT 1000 and HIT1500) -- Deactivated by TPDO communication parameter --		
sub index	Content	Meaning
0	3	Three values are transferred in the <i>PDO</i> .
1	51300110h	The first value in the <i>PDO</i> is the value of <i>Index 5130h, sub index 1</i> with a width of 10h (=16 bits)
2	51300210h	The second value in the <i>PDO</i> is the value of <i>Index 5130h, sub index 2</i> with a width of 10h (=16 bits)
3	51300310h	The third value in the <i>PDO</i> is the value of <i>Index 5130h, sub index 3</i> with a width of 10h (=16 bits)

Index 1A02h: Scaled acceleration values (HIT 1500 only) -- Deactivated by TPDO communication parameter --		
sub index	Content	Meaning
0	3	Three values are transferred in the <i>PDO</i> .
1	51310110h	The first value in the <i>PDO</i> is the value of <i>Index 5131h, sub index 1</i> with a width of 10h (=16 bits)
2	51310210h	The second value in the <i>PDO</i> is the value of <i>Index 5131h, sub index 2</i> with a width of 10h (=16 bits)
3	51310310h	The third value in the <i>PDO</i> is the value of <i>Index 5131h, sub index 3</i> with a width of 10h (=16 bits)

Index 1A03h Inclination motion compensated (HIT 1500 only) -- activated by TPDO communication parameter --		
sub index	Content	Meaning
0	2	Two values are transferred in the <i>PDO</i> .
1	51320110h	The first value in the <i>PDO</i> is the value of <i>Index 5132h, sub index 1</i> with a width of 10h (=16 bits)
2	51320210h	The second value in the <i>PDO</i> is the value of <i>Index 5132h, sub index 2</i> with a width of 10h (=16 bits)

Sub indices used:

0: NumberOfMappedApplicationObjectsInPDO (read write)

Values 0 to 8 are permitted.

This means: no *PDO* is transmitted, or one *PDO* with up to 8 values is transmitted.

ObjectToBeMapped (1 to 8) (read write)

In HIT 1000 / 1500 CANopen Safety the following values are permitted among others:

10010008h = Error Register 8bit width

40000008h = Staus inclination 8bit width **inverted**

40100010h = Process value Slope long16 with 16bit width **inverted**
(DS410 comp. [6])

40200010h = Process value Slope lateral16 with 16bit width **inverted**
(DS410 comp. [6])

50000008h = Staus Neigung 8bit width

51300110h = Process value acceleration in direction of the X-axis (see 9.3.6)
 51300210h = Process value acceleration in direction of the Y-axis
 51300310h = Process value acceleration in direction of the Z-axis

51310110h = Process value gyro around the X-axis (see 9.3.8)
 51310210h = Process value gyro around the Y-axis
 51310310h = Process value gyro around the Z-axis

51320110h = Process value compensated inclination
 (same as Slope long16 see below.)
 51320210h = Process value compensated inclination
 (same as Slope lateral16 see below.)

60100010h = Process value Slope long16 mit 16bit width (DS410 comp. [6])
 60200010h = Process value Slope lateral16 mit 16bit width (DS410 comp. [6])

9.1.20 Index 1F80h: NMT Startup (read / write)

If bit 2 is set, status is changed automatically to "Operational" state immediately after reaching "Pre-Operational" state (DS302).
 Permitted values are: **08h** and **0Ch**

9.2 Device Profile Specific Entries (DS410)

9.2.1 Index 6000h: Resolution (read only)

This entry contains the resolution of the inclination values in angle degrees.
 HIT 1000 / 1500 CANopen Safety returns the value 0Ah.
 According to DS410, this corresponds with a resolution of 0.01° (see [6]).

9.2.2 Index 6010h: Slope long16 (read only) Safe inclination

This entry contains the inclination in longitudinal direction as a signed 16 bit value (see [6]).

9.2.3 Index 6011h: Slope long16 operating parameter (read write)

This entry contains the information on the interpretation of the measured value SlopeLong16.
 HIT 1000 / 1500 CANopen Safety by default returns the value 00h (see [6]).

Bit 7 – 1	Bit 0	Function
0	0	Inversion off
0	1	Inversion on

9.2.4 Index 6020h: Slope lateral16 (read only) Safe inclination

This entry contains the inclination in lateral direction as a signed 16 bit value (see [6] - only in devices providing **2 planes**).

9.2.5 Index 6021h: Slope lateral16 operating parameter (read write)

This entry contains the information on the interpretation of the measured value SlopeLateral16 (see 9.2.3).
 HIT 1000 / 1500 CANopen Safety by default returns the value 00h (see [6]).

9.3 Manufacturer Specific Entries

9.3.1 Index 2001h: NodeID

The *Node ID* is stored in this index.

Sub indices used:

- 0: NumberOfEntries (read only)**
- 1: CurrentNodeid (read only)**
- 2: DemandedNodeid (read write)**

In order for the new *Node ID* to be effective, the command *StoreLssParameters* must first be transmitted and the node must be restarted afterwards.

9.3.2 Index 2002h: Baudrate

The Baud rate is stored in this index.

The allocation of the entry to the Baud rate complies with DS305, *Layer Setting Services and Protocols*.

Entry	0	1	2	3	4	5	6	7	8
Baud rate	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

In order for the new *Baud rate* to be effective, the command *StoreLssParameters* must first be transmitted and the node must be restarted afterwards.

Sub indices used:

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

9.3.3 Index 4000h: Status of the inclination measurement inverted

Identical with Index 5000h, but **inverted bitwise**.

9.3.4 Index 4010h: Slope long16 (read only) Safe inclination inverted

Identical with Index 6010h, but **inverted bitwise**.

9.3.5 Index 4020h: Slope lateral16 (read only) Safe inclination inverted

Identical with Index 6020h, but **inverted bitwise**.

9.3.6 Index 5000h: Status of the inclination measurement

The Status of the inclination measurement is deposited under this index.

Sub indices used:

- 0: Status Uint8 (read only)**

The coding carried out is bit by bit as follows:

Bit 7 – 3	Bit 2	Bit 1	Bit 0
Reserved Always 0	Serious error	Device in movement	Sensor temperature outside of control range

9.3.7 Index 5130h: Scaled acceleration values

The current process value for the component of the gravitational acceleration in the direction of the X, Y and Z axis of the inclinometer are deposited in this index.

The values are scaled into **1/100 m/s²**.

The process value **981** corresponds with **9.81 m/s² = 1g**.

Sub indices used:

- 0: NumberOfEntries (read only)**
- 1: X component Int16 (read only)**
- 2: Y component Int16 (read only)**
- 3: Z component Int16 (read only)**

9.3.8 Index 5131h: Scaled rotational values

The current process value for the component of the rotational value around the X, Y and Z-axes of the inclinometer are deposited in this index (measured in radians).

The values are scaled into **0.2 Millirad/s**. **2π/s** corresponds with one rotation/s.

To say, a process value of **15000** corresponds with a gyro of **3 rad/s**.

Sub indices used:

- 0: NumberOfEntries (read only)**
- 1: X component Int16 (read only)**
- 2: Y component Int16 (read only)**
- 3: Z component Int16 (read only)**

The following drawing shows the positive turning direction about an imaginary axis.



9.3.9 Index 5132h Inclination in space (motion compensated)

The current process value for the compensated inclination to horizontal in the X, Y and Z axes are deposited in this index.

The value has a resolution of 0.01° (compare Index 6010h / 6020h)

Sub indices used:

- 0: NumberOfEntries (read only)**
- 1: According to Slope long16 Component Int16 (read only)**
- 2: According to Slope lateral16 Component Int16 (read only)**

9.3.10 Further indices within the range of 2000h to 5FFFh (reserved)



The indices within this range contain **important factory settings which must not be modified by the customer!**

Warning: Each **Intervention** may lead to **malfunction** of the inclinometer!

10 Layer setting services (LSS) and protocols

The LSS services and protocols, documented in CiA DS305 V2.2 (see [4]), are used for the inquiry or the modification of certain parameters of the Data Link Layer and the Application Layer of a LSS slave by a LSS master via the CAN network.

The following parameters are supported:

- Node-ID
- Baud rate
- LSS address, according to identity object 1018h

Access to the LSS slave thereby is made by its LSS address, consisting of:

- Vendor ID
- Product Code
- Revision number
- Serial number

The measuring system supports the following services:

Switch mode services

- Switch mode selective
 - To address a specific LSS slave
- Switch mode global
 - To address the total of LSS slaves

Configuration services

- Configure Node ID
 - Configure Node ID
- Configure bit timing parameters
 - Configure Baud rate
- Activate bit timing parameters
 - Activate Baud rate
- Store configured parameters
 - Store configured parameters

Inquiry services

- Inquire LSS address
 - Inquire LSS address
- Inquire Node ID
 - Inquire Node ID

Identification services

- LSS identify remote slave
 - Identification of LSS slaves within a certain array
- LSS identify slave
 - Response of all LSS slaves to the previous command
- LSS identify non-configured remote slave
 - Identification of non-configured LSS slaves, Node ID = FFh
- LSS identify non-configured slave
 - Response of all LSS slaves to the previous command

10.1 Finite state automaton, FSA

The FSA is equivalent to a state machine and defines the behaviour of a LSS slave. This state machine is controlled by LSS COBs produced by the LSS master or NMT COBs generated by a NMT master or local NMT state transitions.

The LSS Modes support the following states:

- (0) Initial: Pseudo-State, shows the activation of the FSAs
- (1) LSS waiting: Support for all services as indicated below
- (2) LSS configuration: Support for all services as indicated below
- (3) Final: Pseudo-State, shows the deactivation of the FSAs

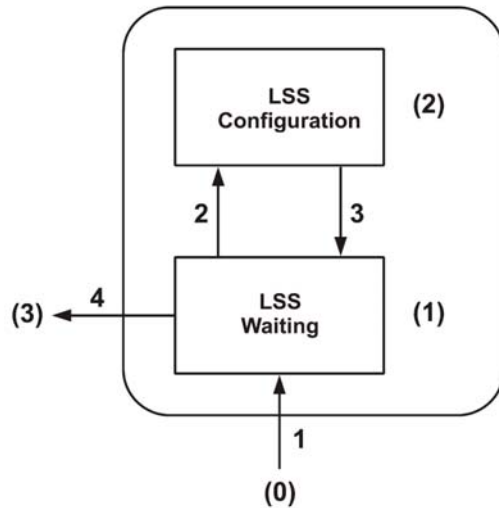


Figure 1: LSS Modes

State behavior of the supported services

Services	Waiting	Configuration
Switch mode global	Yes	Yes
Switch mode selective	Yes	No
Activate bit timing parameters	No	Yes
Configure bit timing parameters	No	Yes
Configure Node ID	No	Yes
Store configuration	No	Yes
Inquire LSS address	No	Yes
LSS identify remote slave	Yes	Yes
LSS identify slave	Yes	Yes
LSS identify non-configured remote slave	Yes	Yes
LSS identify non-configured slave	Yes	Yes

LSS FSA state transitions

Transition	Events	Actions
1	Automatic transition after initialisation when entering either the state NMT PRE OPERATIONAL state or NMT STOPPED state, or NMT RESET COMMUNICATION state with Node ID = FFh.	None
2	LSS 'switch state global' command with parameters 'configuration_switch' or 'switch state selective' command	None
3	LSS 'switch state global' command with parameter 'waiting_switch'	None
4	Automatic transition when an invalid Node ID has been changed and the new Node ID could be successfully stored in non-volatile memory AND the state of LSS waiting has been requested.	None

Once the LSS FSA further state transitions of the NMT FSA of NMT PRE goes through OPERATIONAL STOPPED and vice versa, this will not lead to re-entry into the LSS FSA.

10.2 Transmission of LSS services

By means of LSS services, the LSS master requests the particular services to be performed by the LSS slave. Communication between LSS master and LSS slave is performed by means of implemented LSS protocols.

Similar as in the case of SDO transmission, wo COB IDs for sending and receiving are used in this case as well:

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

Table 1: COB IDs for LSS services

10.2.1 LSS message format

The data field of a CAN message being max. 8 bytes long is used by a LSS service as follows:

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

Table 2: LSS message

Byte 0 contains the **Command-Specifier** (CS), afterwards 7 bytes of data are following.

10.3 Switch mode protocols

10.3.1 Switch mode global protocol

The given protocol has implemented the *Switch mode global service* and controls the state behavior of the LSS slave. By means of the LSS master all LSS slaves in the network can be switched to *Waiting Mode* or *Configuration Mode*.

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 protocol

The given protocol has implemented the *Switch mode selective service* and controls the state behavior of the LSS slave. By means of the LSS master only this LSS slave in the network can be switched to *Configuration Mode*, whose LSS address attributes correspond with the LSS address.

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							

10.4 Configuration protocols

10.4.1 Configure Node ID protocol

The given protocol has implemented the *Configure NMT-address service*. By means of the LSS master the Node ID of a single LSS slave in the network can be configured. Only one LSS slave may be switched to *Configuration Mode* for this purpose. For storage of the new Node ID the *Store configuration protocol* must be transmitted to the LSS slave. To activate the new Node ID the NMT service *Reset Communication (82h)* must be addressed.

LSS-Master --> LSS-Slave

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

LSS-Slave --> LSS-Master

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

Error Code

0: Protocol successfully completed
 1...254: Reserved
 255: application specific error occurred

Specific Error

if Error Code = 255 --> application specific error occurred,
 otherwise reserved by CiA

10.4.2 Configure bit timing parameters protocol

The given protocol has implemented the *Configure bit timing parameters service*. By means of the LSS master the Baud rate of a single LSS slave or of all LSS slaves in the network can be configured. For the storage of the new Node ID the *Store configuration protocol* must be transmitted to the 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. Error	Reserved by CiA				
7E4h	19							

Table Selector

0: Standard CiA Baud rate table

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: Protocol successfully completed
 1: selected Baud rate not supported
 2...254: Reserved
 255: application specific error occurred

Specific Error

if Error Code = 255 --> application specific error occurred,
 otherwise reserved by CiA

10.4.3 Activate bit timing parameters protocol

The given protocol has implemented the *Activate bit timing parameters service* and activates the Baud rate defined via *Configure bit timing parameters protocol* in all LSS Slaves in the network being in *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

The parameter *Switch Delay* defines the length of two delay periods (D1, D2) with equal length. These are necessary to avoid operating the bus with differing Baud rate parameters. After the expiration of the time D1 and after an individual processing duration, the switching is performed internally inside the LSS slave. After the expiration of the time D2, the LSS slave responds with CAN messages and the newly configured Baud rate.

The following applies:

Switch Delay > longest occurring processing duration of a LSS slave

10.4.4 Store configuration protocol

The given protocol has implemented the *Store configured parameters service*. By means of the LSS master the configured parameters of a single LSS slave in the network can be stored into the non-volatile memory. Only one LSS slave may be switched to *Configuration Mode* for this purpose.

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. Error	Reserved by CiA				
7E4h	23							

Error Code

- 0: Protocol successfully completed
- 1: *Store configuration* not supported
- 2...254: Reserved
- 255: application specific error occurred

Specific Error

if Error Code = 255 --> application specific error occurred,
otherwise reserved by CiA

10.5 Inquire LSS address protocols

10.5.1 Inquire Identity Vendor ID protocol

The given protocol has implemented the *Inquire LSS address service*. By means of the LSS master the Vendor ID of a single LSS slave in the network can be read out. Only one device can be switched to *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			

10.5.2 Inquire Identity Product Code Protocol

The given protocol has implemented the *Inquire LSS address service*. By means of the LSS master the manufacturer's product name of a single LSS slave in the network can be read out. Only one LSS slave may be switched to *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			

10.5.3 Inquire Identity Revision-Number protocol

The given protocol has implemented the *Inquire LSS address service*. By means of the LSS master the Identity Revision Number of a single LSS slave in the network can be read out. Only one LSS slave may be switched to *Configuration Mode* for this purpose.

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			

10.5.4 Inquire Identity Serial Number protocol

The given protocol has implemented the *Inquire LSS address service*. Via the LSS master the Serial Number of a single LSS slave in the network can be read out. Only one LSS slave may be switched to *Configuration Mode* for this purpose.

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		

10.5.5 Inquire Node ID protocol

The given protocol has implemented the *Inquire Node ID service*. Via the LSS master the Serial Number of a single LSS slave in the network can be read out. Only one LSS slave can be switched to *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 and 255						

Node-ID

Corresponds to the Node ID of the selected device. If the Node ID has been recently changed by *the Configure Node ID service*, the original Node ID is returned. Only after implementation of the NMT Service *Reset Communication* (82h), the current Node ID is returned.

10.6 Identification Protocols

10.6.1 LSS identify remote slave protocol

The given protocol has implemented the *LSS identify remote slaves service*. By means of the LSS master LSS slaves in the network can be identified within a certain range. All LSS slaves with matching Vendor ID, Product Code, Revision No. - area and Serial No. - area, respond by the *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				

10.6.2 LSS identify slave protocol

The given protocol has implemented the *LSS identify slave service*. All LSS slaves with matching LSS address attributes given in the *LSS identify remote slaves protocol*, respond by this protocol.

LSS-Slave --> LSS-Master

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

10.6.3 LSS identify non-configured remote slave protocol

The specified protocol has implemented the *LSS identify non-configured remote slave service*. By means of the LSS-Master all non-configured LSS Slaves (Node ID = FFh) in the network are identified. The related LSS Slaves respond by the *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							

10.6.4 LSS identify non-configured slave Protocol

The specified protocol has implemented the *LSS identify non-configured slave service*. All LSS slaves having an invalid Node ID (FFh) respond by this protocol after having completed the *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 Connection

The connection can be carried out by means of the enclosed device specific pin assignment, see user manual HIT 1000 / 1500 CANopen Safety (part no.: 669959, the manual is supplied with the HIT 1000 / 1500 CANopen).

11.1 Switching on the supply voltage

After connection and all settings have been carried out, the supply voltage can be switched on.

After POWER ON and finishing the initialisation, the measuring system enters the PRE-OPERATIONAL state. This status is acknowledged by the Boot-Up message "COB ID 700h+Node ID". If the measuring system detects an internal error, an emergency message containing the error code is transmitted (see 6.5).

In PRE-OPERATIONAL state, at first, only a parameter setting via Service Data Objects is possible. But it is possible to configure PDOs with the help of SDOs. If the measuring system has entered the OPERATIONAL state, a transmission of PDOs is possible as well (see 6.7).

11.2 Setting the Node ID and Baud rate by means of LSS services

11.2.1 Configuration of the Node ID, sequence

Assumption:

- LSS address unknown
- only one LSS slave should be in the network
- the Node ID 12 dec. shall be set

Procedure:

- Switch LSS-Slave to *Configuration Mode* using command Specifier 04 *Switch mode global protocol*, Mode = 1. (see 10.3)
- Perform command Specifier 17 *Configure NMT-Address protocol*, Node-ID = 12. (siehe 10.4.1)

--> Wait for feedback and check successful execution,
--> Error Code = 0.

- Carry out command Specifier 23 *Store configuration protocol*.
--> Wait for feedback and check successful execution,
--> Error Code = 0 (see 10.4.4).
- Switch off the supply voltage of the LSS slave, then switch on again.
Now the new configuration is activated.

11.2.2 Configuration of the Baud rate, sequence

Assumption:

- LSS address unknown
- only one LSS slave should be in the network
- the Baud rate 125 kbit/s shall be set

Procedure:

- Perform NMT command Specifier *Stop Remote Node* (02h) in order to switch LSS-Slave into *Stopped state*. The LSS slave shouldn't send any more CAN-messages
--> Heartbeat switched off (see 6.7).
- Switch LSS-Slave to *Configuration Mode* using command Specifier *Switch mode global protocol*, Mode = 1.
(see 10.3.1).
- Perform command Specifier 19 *Configure bit timing parameters protocol*, Table Selector = 0, Table Index = 4
--> wait for feedback and check successful execution,
--> Error Code = 0
(see 10.4.2).
- Carry out command Specifier 21 *Activate bit timing parameters protocol* in order to activate the new Baud rate.
(see 10.4.3).
- Carry out Command Specifier 23 *Store configuration protocol*.
--> Wait for feedback and check successful execution,
--> Error Code = 0 (see 10.4.4).
- Switch off the supply voltage of the LSS slave, then switch on again.
Now the new configuration is activated.

12 Commissioning

12.1 CAN interface

The CAN bus interface is defined by the international standard ISO/DIS 11898 and specifies the two lowest layers of the ISO/DIS CAN Reference Model.

The conversion of the measuring system information to the CAN message format (CAN 2.0A) is done by a CAN-controller. The function of the CAN-controller is controlled by a watchdog.

The CANopen Communication Profile (CiA standard DS301, see [1]) is a subset of CAN Application Layer (CAL) and describes how the services are used by the devices. The CANopen Profile allows the definition of device profiles for decentralized I/O.

The measuring system with CANopen protocol supports the Device Profile for inclinometers (CiA Draft Standard 410, Version 1.3.0, see [6]).

The communication functionality and objects, used in the measuring device profile, are described in an EDS-File (Electronic Data Sheet). When using a CANopen Configuration Tool (e.g.: CANSETTER), the user can read out the objects of the measuring system (SDOs) and program the functionality.

12.2 EDS file

The EDS (electronic datasheet) contains all information on the measuring system-specific parameters and the measuring system's operating modes. The EDS file is integrated using the CANopen network configuration tool to correctly configure or operate the measuring system.

The EDS file matching the device can be identified by the device name and by the device related major revision number in its file name.

The file including the its CANopen interface description can be found and downloaded from our homepage.

<http://www.hydac.com/uk-en/products/sensors/show/Material/index.html>

HYDAC ELECTRONIC GMBH

Hauptstr. 27
D-66128 Saarbruecken
Germany

Web: www.hydac.com
E-Mail: electronic@hydac.com
Tel.: +49 (0)6897 509-01
Fax.: +49 (0)6897 509-1726

HYDAC Service

For enquiries regarding repairs, please contact HYDAC Service.

HYDAC SERVICE GMBH

Hauptstr. 27
D-66128 Saarbruecken
Germany

Tel.: +49 (0)6897 509-1936
Fax: +49 (0)6897 509-1933

NOTE

The information in this manual relates to the operating conditions and applications described. For applications and operating conditions not described. please contact the relevant technical department.

If you have any questions, suggestions, or encounter any problems of a technical nature, please contact your Hydac representative.