# CiA Draft Standard Proposal 304

CANopen

## Framework for Safety-Relevant Communication

**Version 1.0**

**Date: 01.01.2001**

# Table of contents

# 1 Tables

# 2 Figures

# 3 Scope

The CANopen Framework Safety-Relevant Communication is intended to be an add on to the CANopen Application Layer and Communication Profile (see /1/).

It is assumed, that a device with the need of safety-relevant communication can use all the other features defined by the communication profile. Safety is an additional property of such devices. Only special communication objects support safety, all other objects remain normal. The manufacturer and the system integrator shall take care, that the safety requirements are allocated to safe communication objects, that the hardware and software of the device support the safety function and that the device is operated within its safe limits.

This framework describes only the data transport mechanism on a CANopen network, that allows the exchange of safety-relevant data.

Due to CANopen compatibility communication is limited to 64 safe communication objects, so up to 64 suppliers of safety-relevant objects can operate in a CANopen network. The number of consumers of the safety-relevant objects is not defined (at least one receiver is necessary).

# 4 References

/1/ CiA DS-301 Version 4.01, CANopen Application Layer and Communication Profile, June 2000

/2/ Charzinsiki, Bewertung der Fehlersicherungsverfahren im CAN-Protokoll, Universität Stuttgart, 1991

/3/ FAET; FAEM III, BIA, Prüfung und Zertifizierung von “Bussysteme für die Übertragung sicherheitsrelevanter Nachrichten”, Stand 28.05.2000

/4/ IEC 61508, International standard, Functional safety of electrical / electronic / programmable electronic safety-related systems

/5/ DIN V VDE 0801, Grundsätze für Rechner in Systemen mit Sicherheitsaufgaben

/6/ EN954-1 Safety related parts of control systems, Part 1: General principles of design

# 5 Definitions and abbreviations

**BIA**

Berufsgenossenschaftliches Institut für Arbeitssicherheit - Institute for occupational safety of accident insurance institutions

**COB-ID**

Communication object-identifier

**GFC**

Global failsafe command, short and high priority message to ensure fast system reaction (event driven, not safe)

**NMT**

Network management

**Refresh-time**

Configurable time of the periodic SRDO transmission in a source of safety-relevant information, allows to test the ->SCT in the safety node

**RTR**

Remote transmission request; property of CAN, every node can initiate a specific transmission of any other node by a remote request

**Safety controller**

safety relevant output node, that controls a safety-relevant process (e.g. a possibly dangerous motion). A safety controller has to survey all corresponding safety input nodes

**SCT**

Safeguard cycle time, configurable time to fulfill the native time expectation of a specific safety application

**SRDO**

Safety-relevant data object; only mechanism to transport safe data in an CANopen network

**SRVT**

Safety-relevant object validation time; configurable time for the correct reception of a SRDO in a given safety application

**TÜV**

Technischer Überwachungsverein - German Association for Technical Inspection

**SIL**

Safety integrity level

**AK**

Anforderungsklassen - Requirement classes

# 6 Theory of safe operation

It is absolutely essential for a possibly safe system, that there is a safe state. Then as a reaction to an emergency command, an alarm or an error, the safe state can be entered.

It is also important, that the functionality of the safeguard measures is regularly checked. A single defect in a safety-relevant communication must not override the safety circuitry! If such an error occurs, it must be detected within a short period of time in which a 2nd error is unlikely to happen.

All the systems, especially the safety-relevant circuitry must have a high reliability in order to extend the time-span between the safety-tests and minimize the down time of the whole system (e.g. if one of redundant components fails, the system has to be shut off). So the need for safety decreases the availability of a system.

The idea of safety in CAN communication is not to ensure, that there are absolutely no errors and faults. This would be rather hard to proof - anyway. The goal is to detect all possible errors and react in a predictable (safe) way.

In a safe CAN system there are sources of safe information (e.g. safety switches, light barriers, emergency stop buttons) and consumers of such information (e.g. relay, valve or drive controlling a possibly dangerous movement, safety PLC). As the "consumers" control the possible dangerous situation it is responsible for entering the safe state after any safety relevant interference. It also has to check the data integrity of the safety-relevant communication.

So the "consumers" are the safety controllers in a possibly safe CANopen system.

As the sources (safety inputs) are the origin of safe communication objects (SRDOs), their number is limited to 64. The number of safety controllers is not limited in theory, as CAN allows many consumers to listen to the same SRDO(s), i.e. many actuator devices can use the same information.

As the safety controllers are responsible for the data integrity and actuality, every safety-relevant output device has to survey all corresponding sources of safety data.

# 7 Basic communication

Communication in a safe CANopen network is based on /1/.

It is intended, that the additional safe communication is not affecting the normal operation and services on a CANopen network. Safe communication is not related to a special class of devices, so no special device profile has to be used.

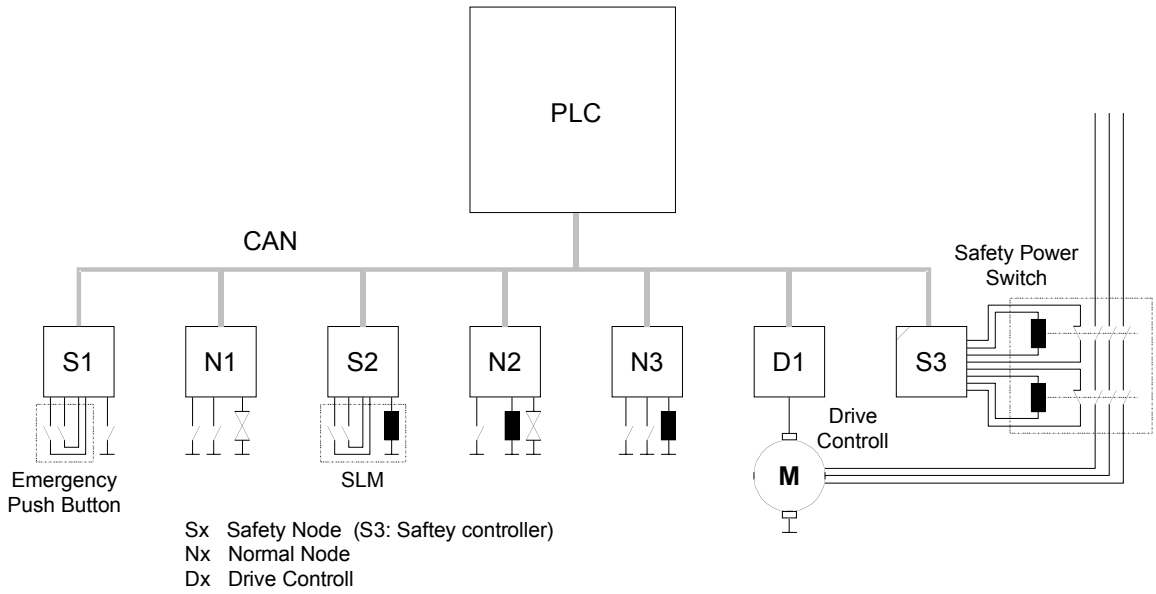


Figure 1: Example of a CANopen network with safe nodes

To ensure compatibility, the usage of identifiers and pre-defined objects has to be coordinated with the CANopen standard and existing device profiles. As there is no use of data bits in the safe communication method, it is compatible with existing device profiles.

In a CANopen network the data interface to the application program within a certain node is only via the CANopen object dictionary. The application itself has to transfer data correct, in time and in sequence to the CANopen kernel. In case of SRDO receiving the application has to collect and check SRDO data so frequently, that the time expectation can be fulfilled.

# 8 Safety-relevant communication

## 8.1 Safety-relevant data object (SRDO)

The safety-relevant data transfer is performed by means of "safety-relevant data objects (SRDO)".

An SRDO shall consist of two CAN data frames with identifiers, which shall be different in at least two bit positions. The user data in both transmissions is redundant, i.e. the meaning of the data is the same, but the data on the 2nd transmission is inverted bitwise.

SRDOs shall be transmitted periodically. If required, SRDOs may also be transmitted event-driven, e.g. to ensure fast reaction after a safety critical change on the input. RTR is not possible, SRDOs are only allowed in the network state "Operational".

An SRDO is only valid, if both CAN frames are received properly (without failure and in time). The redundant transmission is sent after the first transmission to the CAN controller with minimum delay.

There are two kinds of use for SRDOs. The first is data transmission and the second data reception. It is distinguished by the information direction. Devices where the information direction is set to transmit (tx) are SRDO producer and devices where the information direction is set to receive (rx) are called SRDO consumer. SRDOs are described by the SRDO communication parameter (26h) and the SRDO mapping parameter. The structure of this data type is explained in 8.3. The SRDO communication parameter describes the communication capabilities of the SRDO. The SRDO mapping parameter contains information about the content of the SRDOs (device variables). The indices of the corresponding Object Dictionary entries are computed by the following formulas:

SRDO communication parameter index = 1300h + SRDO-number

SRDO mapping parameter index = 1380h + SRDO-number

For each SRDO the pair of communication and mapping parameter is mandatory. The entries mentioned above are described in 8.4.

### 8.1.1 Timing requirements

As SRDOs shall be transmitted periodically in order to test the correct function of the safety-relevant communication on the CAN bus, the so called safeguard cycle time (SCT) is to be defined. It shall be survived by the safety controller.

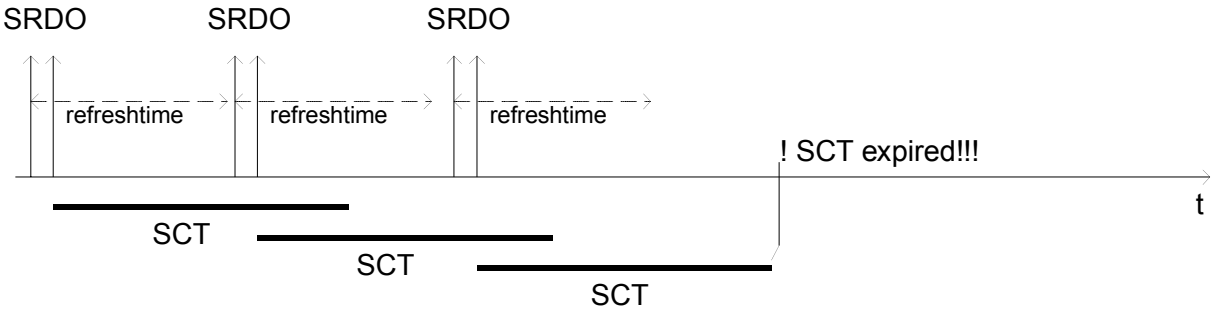


Figure 2: Example for SCT timing

A second test determines, if there is sufficient network capacity for a safety system. Both frames of an SRDO shall be received correctly within the given safety-relevant validation time (SRVT). Normally both frames are transmitted with minimum delay. If the 2nd frame is not being received within SRVT, the bus system has reduced transmission capabilities. The reaction time on a safety relevant event could be enlarged.

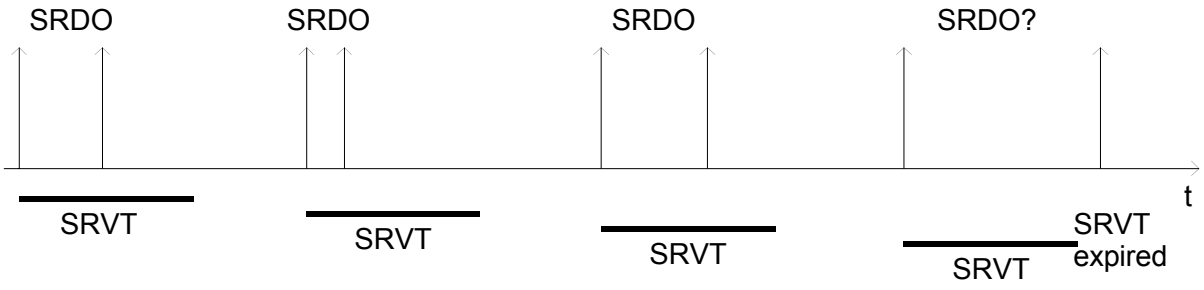


Figure 3: Example for SRVT timing

### 8.1.2 SRDO services

SRDO transmission follows the producer/consumer relationship in /1/ and consits of two CAN data frames.

**Attributes:**

- SRDO number: SRDO number [1..64] for every user type on the local device
- user type: one of the values {consumer, producer}
- data type: according to the SRDO mapping
- refresh-time: n*1 ms, n > 0
- validation-time: n*1 ms, n > 0

#### 8.1.2.1 Write SRDO

For the write SRDO service the push model is valid. There are one or more consumers of a SRDO. A SRDO shall have exactly one producer.

Through this service the producer of a SRDO sends the data of the mapped application object to the consumer(s).

Table 1: Write SRDO

| *Parameter* | *Request / Indication* |
|---|---|
| **Argument**<br>SRDO number<br>Data | **Mandatory**<br>mandatory<br>mandatory |

#### 8.1.2.2 Read SRDO

The read SRDO service is not allowed.

### 8.1.3 SRDO protocol

#### 8.1.3.1 Write SRDO protocol

The service for a SRDO write request is unconfirmed. The SRDO producer sends the process data within a SRDO to the network. There may be 1..n SRDO consumers. At the SRDO consumer(s) the reception of a valid SRDO is indicated. Figure 4 shows the write SRDO Protocol.

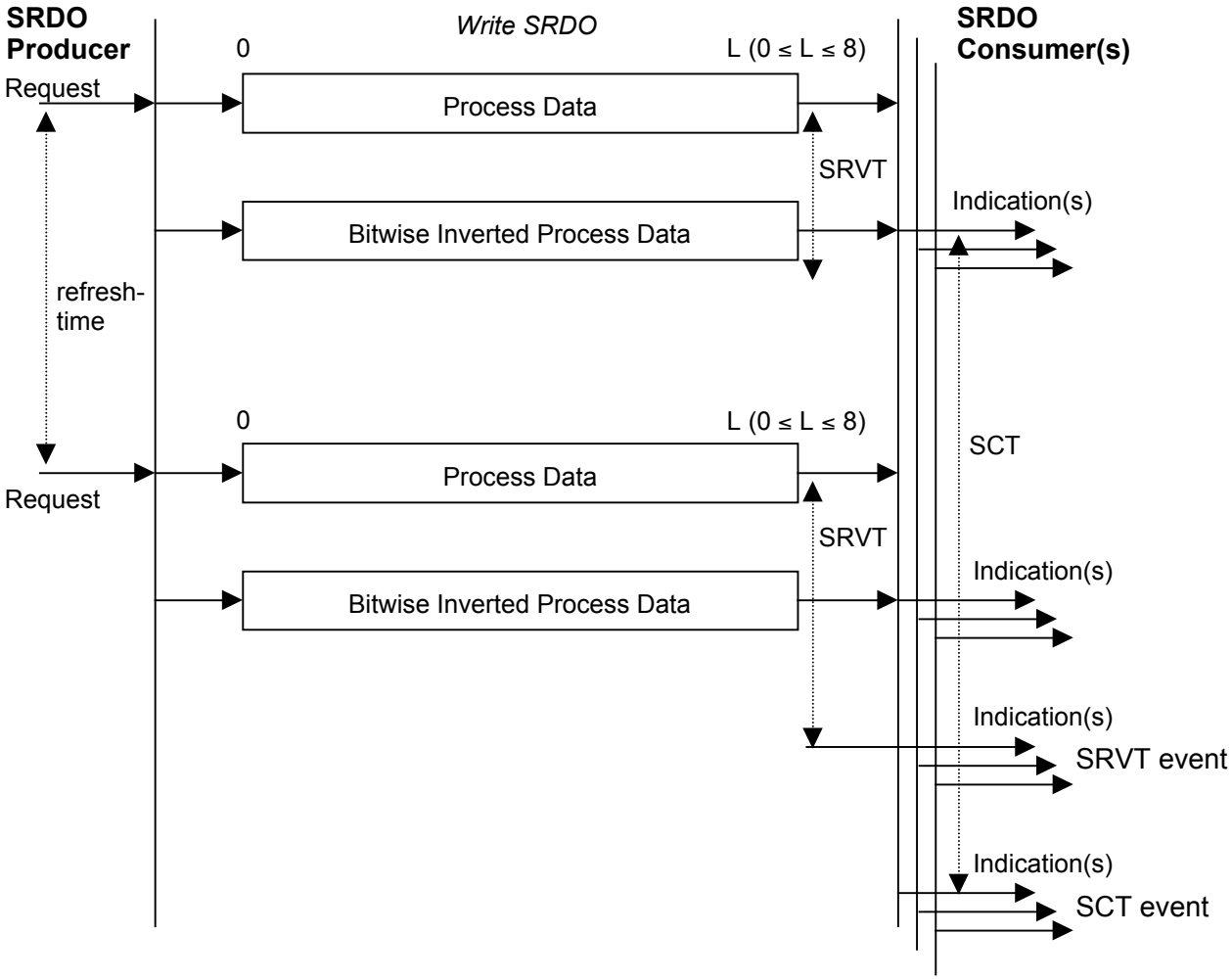


Figure 4: Write SRDO protocol

**Process-data:** up to L bytes of application data according to the SRDO mapping.

If L exceeds the number 'n' defined by the actual SRDO mapping length only the first 'n' bytes are used by the consumer. If L is less than 'n' the data of the received SRDO is not processed and an Emergency message with error code 8210h shall be produced if Emergency is supported.

It is not allowed to request an SRDO by a remote transmission request (RTR).

## 8.2 Global failsafe command (GFC)

### 8.2.1 Global failsafe command usage

In order to speed up the system reaction time the "global failsafe command (GFC)" may be used.

If one transmission at least shall have been received, the global failsafe command is already valid. It allows only one reaction in a CAN network. For that reason the usage of the global failsafe command is optional. It is event driven only and not safe (no periodic time expectation).

Example: After a safety-relevant change at a device input, the global failsafe command may be transmitted first (optional), then the corresponding SRDO shall be follow to maintain safety (mandatory).

### 8.2.2 Global failsafe command service

The global failsafe command transmission follows the producer/consumer push model as described in /1/. The service is uncofirmed; the correspondig SRDO shall follow.

**Attributes:**

- user type: one of the values {consumer, producer}
- data type: nil

### 8.2.3 Global failsafe command protocol

One unconfirmed service (write GFC) is defined.

**Write GFC**

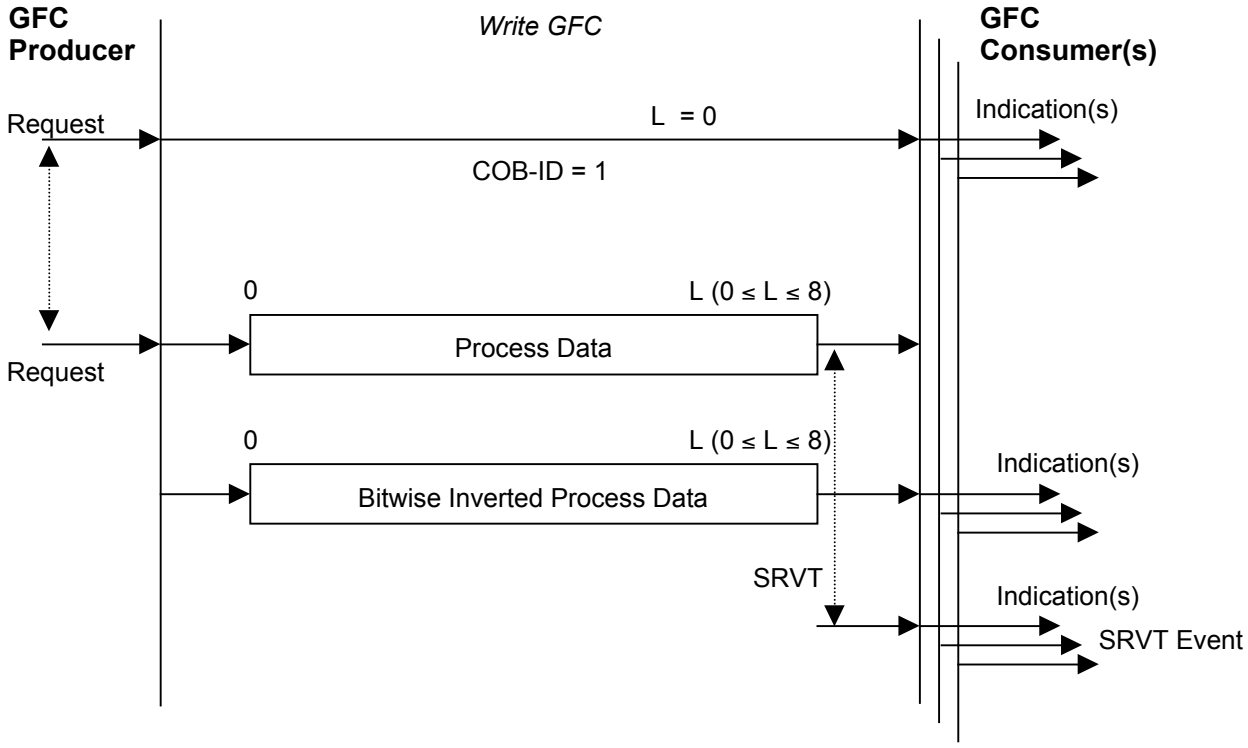


Figure 5: Write GFC protocol

## 8.3 Network initialisation and system boot-up

### 8.3.1 Initialisation procedure for safety networks

In Figure 6 the general flow chart of the network initialisation process, controlled by a NMT master application or configuration application is shown.

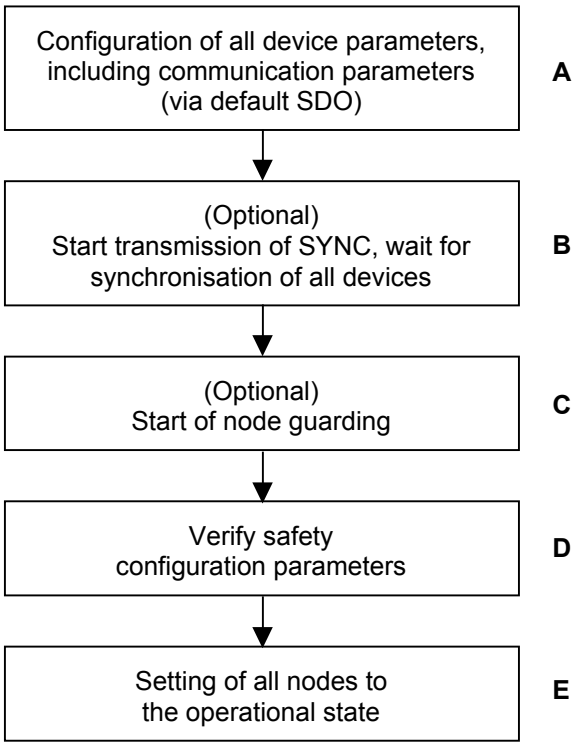


Figure 6: Flow chart of the network initialisation process for safety networks

In step A the devices are in the node state PRE-OPERATIONAL which is entered automatically after power–on. In this state the devices are accessible via their default SDO using identifiers that shall been assigned according to the pre-defined connection set. In this step the configuration of device parameters take place on all nodes which support parameter configuration. Some configuration data might be safety-relevant. So additional measures shall be taken, to ensure the safety function in the network.

This is done from a configuration application or tool which resides on the node that is the client for the default SDOs. For devices that support these feature the selection and/or configuration of PDOs, the mapping of application objects (PDO mapping), the selection and/or configuration of SRDOs, the mapping of application objects (SRDO mapping), the configuration of additional SDOs and optionally the setting of COB-IDs may be performed via the default SDO objects.

In many cases a configuration is not necessary as default values are defined for all application and communication parameters.

If the application requires the synchronisation of all or some nodes in the network, the appropriate mechanisms may be initiated in the optional step B. It may be used to ensure that all nodes except safety nodes are synchronised by the SYNC object before entering the node state OPERTAIONAL in step E. The first transmission of SYNC object starts within 1 sync cycle after entering the PRE–OPERATIONAL state.

In step C node guarding may be activated (if supported) using the guarding parameters configured in step A.

In step D there shall be a method performed for the check of the autenticy of the configuration data. It covers the following safety relevant configuration entries:

- SRDO numbers(s) used
- Time expection (refresh time for TX, SCT for RX and the SRVT between two telegrams )
- Information direction
- Mapping parameter

The checksum of the respective configuration entries shall be defined, that the safe application may check if a safety configuration tool has been used and that the content of the configuration entries has not been changed in state OPERATIONAL. In case of mismatch the safety node shall not transmit SRDOs; the safety controller shall be enter (stay) in safe state.

### 8.3.2 Network states for safety nodes

The "safe state" of a device is not a CANopen communication state and falls not in this scope.

#### 8.3.2.1 Pre-operational

In the PRE-OPERATIONAL state, communication via SDOs is possible, however SRDO and PDO communication is not allowed. Configuration of SRDOs, PDOs, device parameters and also the allocation of application objects (PDO mapping) may be performed by a configuration application. The node may be switched into the OPERATIONAL state directly by sending a Start_Remote_Node request. In the PRE-OPERATIONAL state the application is in the safe state.

#### 8.3.2.2 Operational

In the OPERATIONAL state SRDO, PDO and NMT communication objects are active, however object dictionary access via SDO is not possible. For the safe application this is the only working state (e.g. motor on). Safety communication is only supported in this state.

#### 8.3.2.3 Stopped

By switching a safety device in the STOPPED state communication is limited to receiving the NMT object. The application shall be in the safe state.

#### 8.3.2.4 States and communication object relation

Table 2 shows the relation between communication states and communication objects. Services on the listed communication objects may only be executed if the device involved in the communication are in the appropriate communication states.

Table 2: States and communication objects

| | INITIALISING | PRE-OPERATIONAL | OPERATIONAL | STOPPED |
|---|---|---|---|---|
| PDO | | | X | |
| SDO | | X | (1) | |
| SRDO | | | X | |
| Synchronization object | | | | |
| Time stamp object | | | | |
| Emergency object | | X | X | |
| Boot-up object | X | | | |
| NMT object | | X | X | X |

(1) Writing to a safety entry in the OPERATIONAL state leads to an abort message (abort code: 0800 0022h). Reading of a safety entry in the OPERATIONAL state is allowed.

### 8.3.3 Pre-defined connection set

In order to reduce configuration effort for simple networks a mandatory default identifier allocation scheme is in /1/ defined.

This pre-defined connection set is extended to support one SRDO for every safety node with a Node–ID between 1 and 64. Devices with a Node-ID, which is higher than 64, shall not have pre-defined COB-ID assigned.

Table 3: Broadcast objects of the pre-defined connection set

| object | function code (binary) | resulting COB–IDs | Communication parameters at index |
|---|---|---|---|
| GFC | 0000 | 001h | - |

Table 4: Peer-to-peer objects of the pre-defined connection set

| object | function code (binary) | resulting COB–IDs | | Communication parameters at index | channel direction |
|---|---|---|---|---|---|
| | | normal data | complement data | | |
| SRDO (Node-ID 1 - 32) | 0010 | 257 - 319[(1)] (101h - 13Fh) | 258 - 320[(1)] (102h - 140h) | 1301h | tx |
| SRDO (Node-ID 33 - 64) | 0010 | 321 - 383[(1)] (141h - 17Fh) | 322 - 384[(1)] (142h - 180h) | 1301h | rx |

(1) Every second COB-ID is used.

## 8.4 Object dictionary

### 8.4.1 Specification of additional complex data types

#### 8.4.1.1 SRDO communication parameter record specification

Table 5: SRDO communication parameter record

| Index | Sub-index | Field in SRDO communication parameter record | Data type |
|---|---|---|---|
| 0026h | 0 h | Number of entries | UNSIGNED8 |
| | 1 h | Information direction (TX or RX) | UNSIGNED8 |
| | 2 h | Refresh-time/SCT (in ms) | UNSIGNED16 |
| | 3 h | SRVT (in ms) | UNSIGNED8 |
| | 4 h | Transmission type | UNSIGNED8 |
| | 5 h | COB ID1 | UNSIGNED32 |
| | 6 h | COB ID2 | UNSIGNED32 |

### 8.4.2 Communication profile specification

#### 8.4.2.1 Overview additional object dictionary entries for communication

Table 6 gives an overview over the additional object dictionary entries defined by the communication profile for safety devices.

Table 6: Safety communication objects

| Index | Object | Name | Type | Acc.[1] | M/O |
|---|---|---|---|---|---|
| 1300h | VAR | GFC parameter | UNSIGNED8 | rw | O |
| **SRDO communication parameter** | | | | | |
| 1301h | RECORD | 1st SRDO parameter | SRDO parameter (26h) | rw | M |
| 1302h | RECORD | 2nd SRDO parameter | SRDO parameter (26h) | rw | M/O* |
| ::::: | ::::: | ::::: | ::::: | ::::: | ::::: |
| 1340h | RECORD | 64th SRDO parameter | SRDO parameter (26h) | rw | M/O* |
| 1341h | reserved | | | | |
| ::::: | ::::: | | | | |
| 1380h | reserved | | | | |
| **SRDO mapping parameter** | | | | | |
| 1381h | ARRAY | 1st SRDO mapping | UNSIGNED32 | rw | M |
| 1382h | ARRAY | 2nd SRDO mapping | UNSIGNED32 | rw | M/O* |
| ::::: | ::::: | ::::: | ::::: | ::::: | ::::: |
| 13C0h | ARRAY | 64th SRDO mapping | UNSIGNED32 | rw | M/O* |
| 13C1h | reserved | | | | |
| ::::: | ::::: | | | | |
| 13FDh | reserved | | | | |
| 13FEh | VAR | Configuration valid | UNSIGNED 8 | rw | M |
| 13FFh | ARRAY | Safety configuration checksum | UNSIGNED16 | ro | M |

[1] Access type listed here may vary for certain sub-indices. See detailed object specification.

* If a device supports SRDOs, the according SRDO communication parameter and SRDO mapping entries in the Object Dictionary are mandatory. These may be read only.

#### 8.4.2.2 Detailed specification of additional communication profile specific objects

**Object 1300h: Global failsafe command parameter**

OBJECT DESCRIPTION

| | |
|---|---|
| INDEX | 1300h |
| Name | Global failsafe command parameter |
| Object code | VAR |
| Data type | UNSIGNED 8 |
| Category | Optional |

ENTRY DESCRIPTION

| | |
|---|---|
| Access | rw |
| PDO mapping | No |
| Value range | 0: GFC is not valid<br>1: GFC is valid |
| Default value | 0 |

**Object 1301h - 1340h: SRDO communication parameter**

Contains the communication parameters for the SRDOs the device is able to receive or to transmit. The type of the SRDO communication parameter (26h) is described in 8.4.1.1.

The sub-index 0h contains the number of highest entry within the communication record.

At sub-index 1h resides the information direction of the SRDO. The SRDO information direction allows to select if it is used as a receive SRDO or as a transmit SRDO in the operational state. There may be SRDOs fully configured (e.g. by default) but not used, and therefore set to "not valid" (deleted). The feature is necessary for devices supporting more than 1 SRDO, because each device with a Node-ID in the range from 1 to 64 has only default identifiers for the first SRDO. It is not allowed to change the COB-ID 1 or COB-ID 2 while the SRDO exists (value unequal to 0).

Sub-index 2h contains the refresh-time or the SCT depending on the information direction (see 8.1.1).

Sub-index 3h contains the SRDO validation time, if it is a receive SRDO (see 8.1.1).

The transmission type (sub-index 4h) defines the transmission / reception character of the SRDO. It is defined as type 254. /1/ describes the usage of this entry. On an attempt to change the value of the transmission type an abort message (abort code: 0609 0030h) is generated.

At sub-index 5h and sub-index 6h resides the two COB-IDs of the SRDO. These entries were defined as UNSIGNED32 for compatibility reasons to the COB-ID definitions in /1/. The entry shall be interpreted as defined in Figure 7. The COB-IDs may only be changed in the range from 101h to 180h. Every SRDO uses two following COB-IDs from this range, where the first COB-ID is odd and the second COB-ID is even.

| | UNSIGNED32 | |
|---|---|---|
| | MSB | LSB |
| bits | 31 - 11 | 10 - 0 |
| 11-bit-ID | reserved (=0) | 11 bit identifier |

Figure 7: Structure of SRDO COB-ID entry

OBJECT DESCRIPTION

| | |
|---|---|
| INDEX | 1301h - 1340h |
| Name | SRDO communication parameter |
| Object code | RECORD |
| Data type | SRDO parameter |
| Category | Conditional;<br>Mandatory for each supported SRDO |

ENTRY DESCRIPTION

| | |
|---|---|
| Sub-index | 0h |
| Description | Number of entries |
| Entry category | Mandatory |
| Access | ro |
| PDO mapping | No |
| Value range | 6 |
| Default value | 6 |

| | |
|---|---|
| Sub-index | 1h |
| Description | Information direction |
| Entry category | Mandatory |
| Access | rw |
| PDO mapping | No |
| Value range | 0: does not exist / is not valid<br>1: exists / is valid for transmit (tx)<br>2: exists / is valid for receive (rx)<br>3 - 255: reserved |
| Default value | see: pre-defined connection set |

| | |
|---|---|
| Sub-index | 2h |
| Description | tx: refresh-time in ms<br>rx: SCT in ms |
| Entry category | Mandatory |
| Access | rw |
| PDO mapping | No |
| Value range | 1 - 65535 |
| Default value | tx: 25 ms<br>rx: 50 ms |

| Sub-index | 3h |
|---|---|
| Description | tx: not used<br>rx: SRDO validation time in ms |
| Entry category | Conditional |
| Access | rw |
| PDO mapping | No |
| Value range | 1 - 255 |
| Default value | 20 ms |

| Sub-index | 4h |
|---|---|
| Description | Transmission type |
| Entry category | Mandatory |
| Access | ro |
| PDO mapping | No |
| Value range | 254 |
| Default value | 254 |

| Sub-index | 5h |
|---|---|
| Description | COB-ID 1 |
| Entry category | Mandatory |
| Access | rw |
| PDO mapping | No |
| Value range | 257, 259 - 383 |
| Default value | Index 1301h: 0FFh + (2 x Node-ID),<br>Index 1302h - 1340h: disabled |

| Sub-index | 6h |
|---|---|
| Description | COB-ID 2 |
| Entry category | Mandatory |
| Access | rw |
| PDO mapping | No |
| Value range | 258, 260 - 384, |
| Default value | Index 1301h: 100h + (2 x Node-ID),<br>Index 1302h - 1340h: disabled |

**Object 1381h - 13C0h: SRDO mapping parameter**

Contains the mapping for the SRDOs the device is able to receive or transmit. The type of the SRDO mapping parameter is an array of type UNSIGNED32. The sub-index 0h contains the number of valid entries within the mapping array. This half of the number of entries is also the number of the application variables which shall be received/transmitted with the corresponding SRDO. The sub-indices from 1h to number of entries contain the information about the mapped application variables. The structure and the procedure for the mapping is described in /1/. For changing the SRDO mapping first the SRDO shall be deleted.

OBJECT DESCRIPTION

| | |
|---|---|
| INDEX | 1381h - 13C0h |
| Name | SRDO mapping parameter |
| Object code | ARRAY |
| Data type | UNSIGNED32 |
| Category | Conditional;<br>Mandatory for each supported SRDO |

ENTRY DESCRIPTION

| | |
|---|---|
| Sub-index | 0h |
| Description | Number of mapped application objects in SRDO |
| Entry category | Mandatory |
| Access | ro;<br>rw if dynamic mapping is supported |
| PDO mapping | No |
| Value range | 0: deactivated<br>2, 4 - 128: activated |
| Default value | (device profile dependent) |

| | |
|---|---|
| Sub-index | 1h, 3h, 5h - 7Fh |
| Description | SRDO mapping for the n-th application object to be mapped (data not inverted) |
| Entry category | Conditional;<br>depends on number and size of object be mapped |
| Access | rw |
| PDO mapping | No |
| Value range | UNSIGNED32 |
| Default value | (device profile dependent) |

| | |
|---|---|
| Sub-index | 2h, 4h, 6h - 80h |
| Description | SRDO mapping for the n-th application object to be mapped (data inverted) |
| Entry category | Conditional;<br>depends on number and size of object be mapped |
| Access | rw |
| PDO mapping | No |
| Value range | UNSIGNED32 |
| Default value | (device profile dependent) |

**Object 13FEh Configuration valid**

Contains an acknowledgement flag for a valid configuration. The value for the valid flag is A5h (165d) all other values are not valid. After a write access to the safety-relevant parameter the entry of object 13FEh is automatically 0 or rather “not valid”. If the configuration is finished it shall be acknowledged with an entry A5h or rather “valid” in object 13FEh.

OBJECT DESCRIPTION

| | |
|---|---|
| INDEX | 13FEh |
| Name | Configuration valid |
| Object code | VAR |
| Data type | UNSIGNED 8 |
| Category | Mandatory |

ENTRY DESCRIPTION

| | |
|---|---|
| Access | rw |
| PDO mapping | No |
| Value range | 0 – FFh<br>0-A4h :Configuration is not valid<br>A5h: Configuration valid<br>A6h – FFh: Configuration is not valid |
| Default value | 0 |

**Object 13FFh: Safety configuration checksum**

OBJECT DESCRIPTION

| | |
|---|---|
| INDEX | 13FFh |
| Name | Safety configuration checksum |
| Object code | ARRAY |
| Data type | UNSIGNED16 |
| Category | Mandatory |

ENTRY DESCRIPTION

| | |
|---|---|
| Sub-index | 0h |
| Description | Number of entries |
| Entry category | Mandatory |
| Access | rw |
| PDO mapping | No |
| Value range | 1 - 64 |
| Default value | 1 |

| | |
|---|---|
| Sub-index | 1h |
| Description | Checksum |
| Entry category | Mandatory |
| Access | rw |
| PDO mapping | No |
| Value range | UNSIGNED16 |
| Default value | 0 |

| | |
|---|---|
| Sub-index | 2h – 40h |
| Description | Checksum |
| Entry category | Conditional;<br>Mandatory for each additional supported SRDO |
| Access | rw |
| PDO mapping | No |
| Value range | UNSIGNED16 |
| Default value | 0 |

Generator polynomial $G(x) = X^{16}+x^{12}+x^{5}+1$

The order for data which are checked by the CRC:

**- Communication parameter**

a) Information direction 1 byte = $a_7...a_0$

b) Refreshtime or SCT 2 byte = $b_{15}...b_0$

c) SRVT 1 byte = $c_7...c_0$

d) COB-ID 1 4 byte = $d_{31}...d_0$

e) COB-ID 2 4 byte = $e_{31}...e_0$

**- Mapping parameter**

f) Sub-index 0 1 byte (Number of mapped application objects in SRDO) = $f_7...f_0$

$g^1$) Sub-index 1 byte (SRDO mapping for the nth application object to be mapped) = $g^1{}_7...g^1{}_0$

$h^1$) Mapping data 4 byte (2 byte index, 1 byte sub-index, 1 byte data length) = $h^1{}_{31}...h^1{}_0$

.

.

.

$g^{128}$) Sub-index 1 byte (SRDO mapping for the nth application object to be mapped) = $g^{128}{}_7...g^{128}{}_0$

$h^{128}$) Mapping data 4 byte (2 byte index, 1 byte Sub-index, 1 byte data length) = $h^{128}{}_{31}...h^{128}{}_0$

$$D(x) = x^n + .....+x^0$$

$$D(x) = a_7+...+a_0+b_{15}+...+b_0+c_7+...+c_0+d_{31}+...+d_0+e_{31}+...+e_0+f_7+...+f_0+g^1{}_7+... +g^1{}_0+h^1{}_{31}+...+h^1{}_0+.....+g^{128}{}_7+...+g^{128}{}_0+h^{128}{}_{31}...h^{128}{}_0$$

# 9 Annex

## 9.1 Hardware

In a safe system the hardware and the software are interdependent on each other. Depending on the level of safety required various structures may be useful.

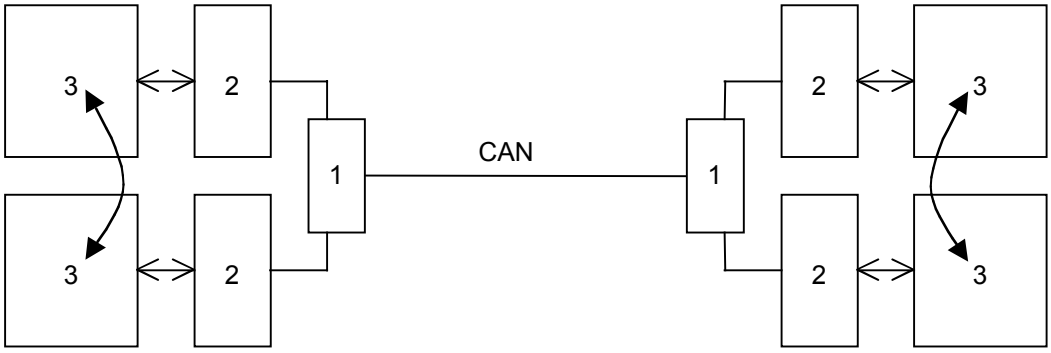


Figure 8: One transceiver and two CAN controllers, redundant μC for SIL 2 and SIL 3 (IEC 61508) or AK 4 and AK 6 (DIN V VDE 801) (C-Model /3/)

1) Transceiver
2) CAN controller
3) Microcontroller

## 9.2 Configuration mechanism

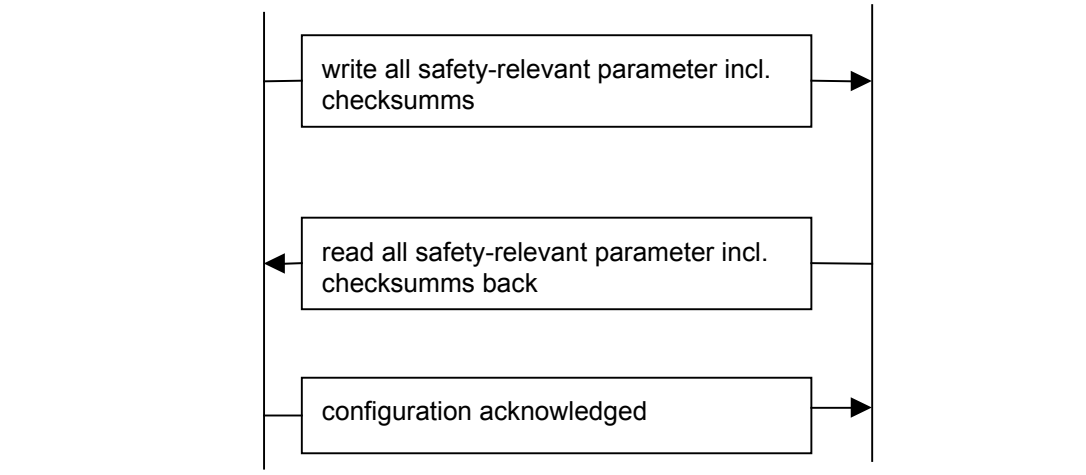


Figure 9: Principle of a safe configuration

All safety parameter are downloaded to the safety device. After that, the parameters shall be uploaded to the configuration tool. The data shall be compared and if it is without failure it shall be acknowledged.

The first configuration (the configuration of baudrate and Node-ID) shall be enforced accordingly the category of the EN954-1 (/6/).

## 9.3 Mathematical analysis of CANopen Safety

The worst case residual error probability of the CAN Protocol is

$$P_{Rest} = 7 \cdot 10^{-9} \approx 1 \cdot 10^{-8} \text{ /2/}$$

For model C (/3/) sending the same data twice the result is

$$P = P_{Rest}^2$$

The residual error rate per hour is

$$\Lambda \equiv 3600 \cdot P \cdot \nu \cdot (m-1) \cdot 100$$

$\nu$: safety relevant messages per second

m: number of safety relevant devices = max. 64

P: residual error probability

The error rate per hour for SIL 3 shall be < $10^{-7}$, for SIL 2 it shall be < $10^{-6}$ and SIL 1 < $10^{-5}$(/3/). For SIL 3 is the SRDO transfer limited to 44 SRDO per second. This results in a **refreshtime of 23 ms** with 64 safety nodes.

## 9.4 Limits and recommendations

- In general the use of remote frames in a safety relevant CANopen network is not recommended. (RTR to SRDOs is not possible anyway).

The use of "node guarding" in a safety-relevant CANopen network is restricted. If required, use instead the optional "heart beat" (SRDOs have a implicit guarding mechanism).

In a network with safety-relevant devices it shall be not allowed for non safety-relevant nodes to use the identifier range of the CANopen Safety.

The implementation of CANopen Safety shall be allowed only in safety devices.

## 9.5 Rules of implementation

- The first cyclic transmit of an SRDO shall be delayed for 0.5 ms * Node-ID.
- A transmit SRDO shall be built by two different ways from the application.
- A received SRDO shall be compared bit by bit (modulo 2) in the application (data and identifier).
- It shall be guaranteed that the message buffers of the CAN controller are dynamically tested
- The application shall check that the two telegrams of a SRDO are received in chronological order (high priority identifier first). It is important to mark an old received SRDO as invalid.
- The application shall check the parameter in the transition from pre-operational to operational. The CRC-Entry in the object dictionary shall be equal to the CRC calculation of the safety device and the “configuration-valid” – flag shall be valid.

**Important:**

- The rules for implementation of hard-, soft- and firmware in a safety device are defined DIN V VDE 0801 or IEC 61508!