

EUROMAP 66-1A

Protocol for Communication with Peripheral
Equipment

General Description

APPENDIX A

Version 1.1, July 2002
(16 pages)

This recommendation was prepared by the Technical Commission of EUROMAP.

History

Date	Changes
June 2002	Document revised. Correction of some typing errors. Consistent representation of hex values. Important changes: New chapter 4.5 PDO- and SDO- usage

Contents

1	INTRODUCTION	4
1.1	PURPOSE	4
1.2	SCOPE	4
1.3	DEFINITIONS, ACRONYMS AND ABBREVIATIONS	4
1.4	REFERENCES.....	4
1.5	DOCUMENT OVERVIEW.....	4
2	OVERVIEW	5
2.1	EUROMAP 66 DEVICES	5
2.2	CAN/CANOPEN COMMUNICATION PROTOCOL.....	5
2.3	EUROMAP 66 DEVICES AND CANOPEN	6
2.4	EUROMAP 66 DEVICE INTERFACE.....	6
2.5	PROFILES	6
3	COMMUNICATION WITH CANOPEN	7
3.1	CAN COMMUNICATION	7
3.2	CANOPEN COMMUNICATION	7
3.2.1	CANopen communication model.....	7
3.2.2	Predefined communication objects	8
3.2.3	Error handling	8
3.2.4	Network management	9
3.2.5	Identifier distribution	9
3.3	CAN REACTION TIMES.....	10
3.4	DATA FORMAT, SCALING AND SI UNIT	10
3.5	TRANSMISSION MECHANISMS	10
3.6	CANOPEN DEVICE PROFILES	10
4	USE OF EUROMAP 66 DEVICES	12
4.1	WIRING.....	12
4.2	SETTINGS	12
4.3	CONNECTION SETUP AND INITIALIZATION.....	12
4.4	OPERATION	13
4.5	PDO- AND SDO- USAGE.....	13
4.6	TREATMENT OF COMMUNICATION ERRORS	14
5	SUMMARY	15

1 Introduction

1.1 Purpose

EUROMAP 66 is based on CAN/CANopen. This document provides some basic information about CAN/CANopen communication.

1.2 Scope

CANopen is a networking system based on the serial bus Controller Area Network (CAN). It is specified in various CiA documents.

The purpose of this document is to provide a basic overview over CANopen services, as well as to give some guidelines for implementations for both sides, machine and EUROMAP 66 devices.

1.3 Definitions, acronyms and abbreviations

EUROMAP	European Committee of Manufacturers of Plastics and Rubber Machinery (http://www.euromap.org/).
CiA	CAN in Automation. Organization responsible for the definition of different CAN protocols, e.g. CAN Application Layer (CAL) and CANopen (http://www.can-cia.de/).
CAL	CAN Application Layer. Communication mechanisms standardized by CiA for CAN-based systems (DS 201..207).
CANopen	Communication profiles (DS 301) and device profiles (CiA DS 40x) based on CAL, standardized by CiA.

1.4 References

<u>Short name</u>	<u>Title</u>	<u>Version</u>	<u>Autor</u>
CiA DS-102	CAN Physical Layer for Industrial Applications	2.0	CiA
CiA DS-301	CANopen Application Layer and Communication Profile	4.0	CiA

1.5 Document Overview

Chapter 2 gives a overview over CANopen.

In Chapter 3, some details of CANopen communication are discussed.

Chapter 4 gives some examples for communication between control and EUROMAP 66 devices.

The last chapter summarizes the advantages of EUROMAP 66.

2 Overview

2.1 EUROMAP 66 devices

External peripheral devices are often used with plastics processing machinery for heating/cooling purposes. These can be controlled by the machine.

2.2 CAN/CANopen communication protocol

The CAN bus is widely used in industrial systems. It is economical and offers a very high transmission safety. To enable a standardized communication on the CAN bus, the CiA defined the CANopen family of protocols which contains predefined profiles for different devices.

Definition of CiA:

CAN

CAN is a serial bus system with multi-master capabilities, that is, all CAN nodes are able to transmit data and several CAN nodes can request the bus simultaneously. The serial bus system with real-time capabilities is the subject of the ISO 11898 international standard and covers the lowest two layers of the ISO/OSI reference model. In CAN networks there is no addressing of subscribers or stations in the conventional sense, but instead, prioritized messages are transmitted. A transmitter sends a message to all CAN nodes (broadcasting). Each node decides on the basis of the identifier received whether it should process the message or not. The identifier also determines the priority that the message enjoys in competition for bus access.

...

Each CAN message can transmit from 0 to 8 bytes of user information. Of course, you can transmit longer data information by using segmentation. The maximum transmission rate is specified as 1 Mbit/s. This value applies to networks up to 40 m. For longer distances the data rate must be reduced: for distances up to 500 m a speed of 125 Kbit/s is possible, and for transmissions up to 1 km a data rate of 50 Kbit/s is permitted.

CANopen

CANopen is a networking system based on the serial bus Controller Area Network (CAN). The CANopen Communication Profile (CiA DS-301) supports both direct access to device parameters and time-critical process data communication. CANopen device profiles (CiA DS-40x) define standards for basic device functionality while providing ample scope for additional vendor-specific device features. CANopen unleashes the full power of CAN by allowing direct peer-to-peer data exchange between nodes in an organized and, if necessary, deterministic manner. The network management functions specified in CANopen simplify project design, implementation and diagnosis by providing standard mechanisms for network start-up and error management.

CANopen supports both-cyclic and event-driven communication. This makes it possible to reduce the bus load to a minimum but still maintaining extremely short reaction times. High communication performance can be achieved at relatively low baud rates, thus reducing EMC problems and minimizing cable costs.

CANopen is the ideal networking system for all types of automated machinery. One of the distinguishing features of CANopen is its support for data exchange at the supervisory

control level as well as accommodating the integration of very small sensors and actuators on the same physical network. This avoids the unnecessary expense of gateways linking sensor/actuator bus systems with higher communication networks and makes CANopen particularly attractive to original equipment manufacturers.

2.3 EUROMAP 66 devices and CANopen

Peripheral devices often contain heater controllers which are controlled from the injection moulding machine. The machine can switch the controller on and off, and preset the nominal temperature. The controller in turn can feedback the actual values as well as further data.

The "CANopen Device Profile for Measuring Devices and Closed-Loop Controllers" (CiA DP-404), has been defined for such devices. This profile supports multi-channel devices.

There are already a substantial number of device manufacturers who offer devices with CANopen interfaces. It is for obvious reasons, therefore, that a new interface definition is structured on this standard.

CANopen offers the advantage that it can be implemented at low cost. Various companies offer software modules which can be configured and/or adapted. Some of these firms also provide such configuration and/or adaptation services.

2.4 EUROMAP 66 device interface

The EUROMAP 66 interface builds on the CANopen standard. Within this standard, profiles for different appliances are defined. These profiles define different blocks of functions, e.g. for controllers. These blocks of functions can be utilized in a variety of ways, while allowing some room for interpretation. Apart from this, additional functions are required which are not contained in the profile.

The EUROMAP 66 device defines how particular blocks of functions are to be implemented. Moreover, the CANopen profile offers room for producer-specific extensions. They are used in this instance to depict the additionally required information.

2.5 Profiles

Heating/cooling appliances feature on the one hand general parameters such as e.g. nominal (setpoint) and actual temperature value. These do not depend on the producer and are contained in the controller block. In some cases, however, additional values are required such as e.g. throughflow or pressure. These are not contained in the profile and consequently have to be depicted in the producer-specific area.

In order to enable a producer-independent possible utilization all the same, device profiles are defined in the new EUROMAP 66 interface. A part of the producer specific area is firmly preset in these device profiles.

These profiles contain a list of the additional device objects. While this does not mean that all objects have to be implemented in each device, it ensures that all device specific objects are accessible in the same manner for any producers.

At this stage only the profile for heating/cooling device profile is defined.

3 Communication with CANopen

3.1 CAN communication

CAN communication is based on message telegrams with maximum 8 byte user data. These messages, also referred to as communication objects (COB), are marked by an 11 bit identifier which defines the priority of the message at the same time.

Some important characteristics of the CAN bus are:

- Multimaster bus access technology.
- CSMA/CD access technology with bit-by-bit arbitration depending on the priority of the message. This guarantees that in case of conflict, if two or more participants want to transmit to the BUS simultaneously, the most important message is always transmitted first and no transmission time is lost.
- Short message transmission delays in view of the relatively short messages.
- Each message is secured by a 15-bit CRC field.

3.2 CANopen communication

CANopen is based on the CAN communication. It defines how the message identifiers are allocated to the devices and how the messages are structured.

CANopen devices are identified over a node number (node ID). This must normally be set at the device, for example by means of a selector switch.

Central element of each CANopen device is the objects directory. It contains all parameters of the device. This comprises communication parameters and application parameters as well as application data. Some of these objects are defined by the communication profile and by device profiles. Moreover, much room remains for producer-specific extensions.

The data objects are usually of the 8, 16 or 32-bit format with or without operational sign. Bigger data, e.g. character strings, are also possible. In these cases, however, access requires several CAN messages.

Various services enable buildup of the connection, configuration, synchronization, as well as general accesses to the objects directory.

3.2.1 CANopen communication model

In the communication model it is differentiated between the following message types with own identifier in each case:

Administrative messages (e.g. NMT)

Service data messages (SDO)

Process data messages (PDO)

Predefined messages (e.g. synchronization and emergency messages)

SDO

Service data messages enable reading and writing of data in the objects directory.

Each device has one receive- and one send-SDO. Contained in the SDO are command (e.g. write, read) as well as the identification of the object (index, subindex). Maximum 4 bytes of user data remain. With these it is possible to transmit e.g. a 32-bit value. Bigger data objects have to be transmitted in the segmented mode by means of several messages.

The SDO communication is triggered by the control. All SDOs (except "Abort Domain Transfer") are confirmed by the device.

PDO

Process data messages enable the data exchange in real-time. By means of PDO, for example, a status word can be transmitted immediately on amendment. PDO can be transmitted by both control and device. They have higher priority in relation to SDO.

Up to 8 byte of pure user data are transmitted by means of PDO. The content can be defined on building up of the connection and is then clearly defined by the identifier. In other words, no protocol overhead is created. This means that the transmission is very fast. Several send- and receive-PDO are possible per device.

The transmission of PDOs is not confirmed, i.e. there is no acknowledgment from the receiver. However, the transmission via CAN is very safe. (with a residual error probability of less than 4.7×10^{-11})

3.2.2 Predefined communication objects

The following communication objects (among others) are predefined:

SYNC object

Emergency object

SYNC object:

Devices can be synchronized with the SYNC object by the control. This synchronization can comprise both the communication and activities in the device. SYNC objects (if used) are transmitted cyclically by the control to all devices.

The SYNC object consists of a telegram of very high priority. The telegram has no data bytes. In reaction to "SYNC" the devices can for example transmit suitably configured PDO (master-slave principle) and start actions, e.g. measurements.

Emergency object: see error handling.

3.2.3 Error handling

Available for the error handling are the Node Guard Object and the Emergency Object.

Node guarding

Node guarding enables mutual monitoring of control and device. The control requests the Node Guard Telegram cyclically from each device by means of RTR (Remote Transmit). The control can identify a node error over the timeout supervision. If the request is missing, the device can identify an error and e.g. react in a defined manner.

Emergency

A device can report an error status by means of emergency object. This message has a higher priority than the PDO messages.

3.2.4 Network management

CANopen defines a simple network management (NMT). There are different statuses between which it is possible to change by means of NMT commands. They are only transmitted by the control at the start. NMT messages enjoy highest priority.

A device must at least support the following states ("Minimum Capability Device"):

Initialization	Status after power up. The device initializes itself autonomously and changes to the pre-operational status.
Pre-operational	The device permits accesses from the object directory by means of SDO. This means that it can be parameterized. PDO communication is impossible.
Operational	The device is operable. It is synchronized if appropriate and can communicate via both SDO and PDO channels.
Prepared	In this status neither SDO nor PDO communication is possible. Node guarding, however, is active.

3.2.5 Identifier distribution

The message identifier is made up of a function code and the node ID as standard. Below an overview:

Object	Function code (binary)	Resulting COB-ID	Remark
NMT	0000	0	Control -> on / all devices
SYNC	0001	128	Control -> all devices
TIME STAMP	0010	256	Control -> all devices

Broadcast objects in the predefined master/slave connection set

Object	Function code (binary)	Resulting COB-ID	Remark
EMERGENCY	0001	129 - 255	Device -> control
PDO1 (tx)	0011	385 - 511	Device -> control
PDO1 (rx)	0100	513 - 639	Control -> device
PDO2 (tx)	0101	641 - 767	Device -> control
PDO2 (rx)	0110	769 - 895	Control -> device
SDO (tx)	1011	1409 - 1535	Device -> control
SDO (rx)	1100	1537 - 1663	Control -> device
Node Guard	1110	1793 - 1919	Control <-> device

Peer-to-peer objects in the predefined master/slave connection set

The message with the lowest identifier (COB-ID) has the highest priority.

Within the objects the device with the lowest node number has highest priority. However, this comes hardly ever to bear because the bus is never so intensively charged in practice.

3.3 CAN reaction times

The CAN bus allows baud rates between 10 kbit/s and 1 Mbit/s.

At a baud rate of 250 kbit/s as proposed in the EUROMAP 66 interface, the transmission of a message with 8 byte user data takes approx. 500 μ s. Per millisecond, therefore, about 2 messages can be transmitted on the bus.

This means that, at this baud rate, a highly prioritized message, e.g. PDO1, is typically transmitted after maximum 500 μ s and has arrived at the receiver after 1 ms.

With confirmed messages (SDO) it is possible to transmit and receive higher-prioritized messages between request and response.

3.4 Data format, scaling and SI unit

The objects commonly used in controllers are defined in the communication profile DSP-404. Object 7402h, for example, is the controller setpoint. Information on the interpretation of the values is placed in further objects. In object 6406h the SI unit and in object 6407h the number of digits after the decimal point for the setpoint, for example. The control has to read or if appropriate write these objects only once since they never change in operation.

The EUROMAP 66 device interface of heating/cooling devices proposes for the data a 16-bit integer format with one digit after the decimal point. This enables the most efficient transmission. In this way it is possible to transmit e.g. up to four values cyclically with one PDO.

3.5 Transmission mechanisms

The control can read values from the device object directory at any time. However, there exist more efficient possibilities also.

Depending on requirement and possibility of the device it is possible to select (e.g.) cyclic or event-driven transmission. The following possibilities exist for this purpose in principle:

- Maximum 4 16-bit values (also other combinations, max. 8 byte) can be transmitted cyclically by the device. In this case it is necessary to map corresponding objects into the PDO and the PDO must be parameterized accordingly (by means of SDO).
- Values can be monitored. The device sends a PDO with an "Alarm-state" word as soon as a monitored value has altered in a defined manner (e.g. alteration by a certain value, exceeding or falling short of a threshold). Based on the bit it is possible to detect which object triggered the alarm and has to be read from the objects directory by means of SDO. For this purpose the corresponding threshold or alteration values must be put and the trigger released. The alarm word, moreover, must be mapped into the PDO (also by means of SDO).

3.6 CANopen device profiles

CANopen devices make all data available via the object directory. The CANopen communication profile defines (a.o.) this object directory and the access mechanisms to the individual objects.

Additionally, some mandatory and optional objects are defined for the communication ("Communication Profile Area"). See [DS-301].

The identification of the objects in the directory is effected by means of index and subindex. The index ranges available for the different objects and object types are defined in the communication profile.

The CANopen device profiles define device-specific mandatory and optional objects ("Standardized Device Profile Area"). See [DSP-404]. It defines in the main how the functionality of a device must be implemented, if it is implemented at all.

Device profiles exist for different types of devices. A type of device "EUROMAP 66", however, does not exist. A definition of a corresponding profile, furthermore, would take up very much time. A profile meeting the requirements rather closely is the CiA DSP 404 ("CANopen Device Profile for Measuring Devices and Closed-Loop Controllers").

The EUROMAP 66 heating/cooling device profile is based on DSP 404. EUROMAP 66, however, defines additional extensions specific to temperature control units.

4 Use of EUROMAP 66 devices

4.1 Wiring

CAN devices are wired as a bus. They always feature two CAN connectors. The wiring is from the control to the first connector of the first device, then from the second connector to the next device, and so on. The CAN bus must be terminated to avoid reflections.

4.2 Settings

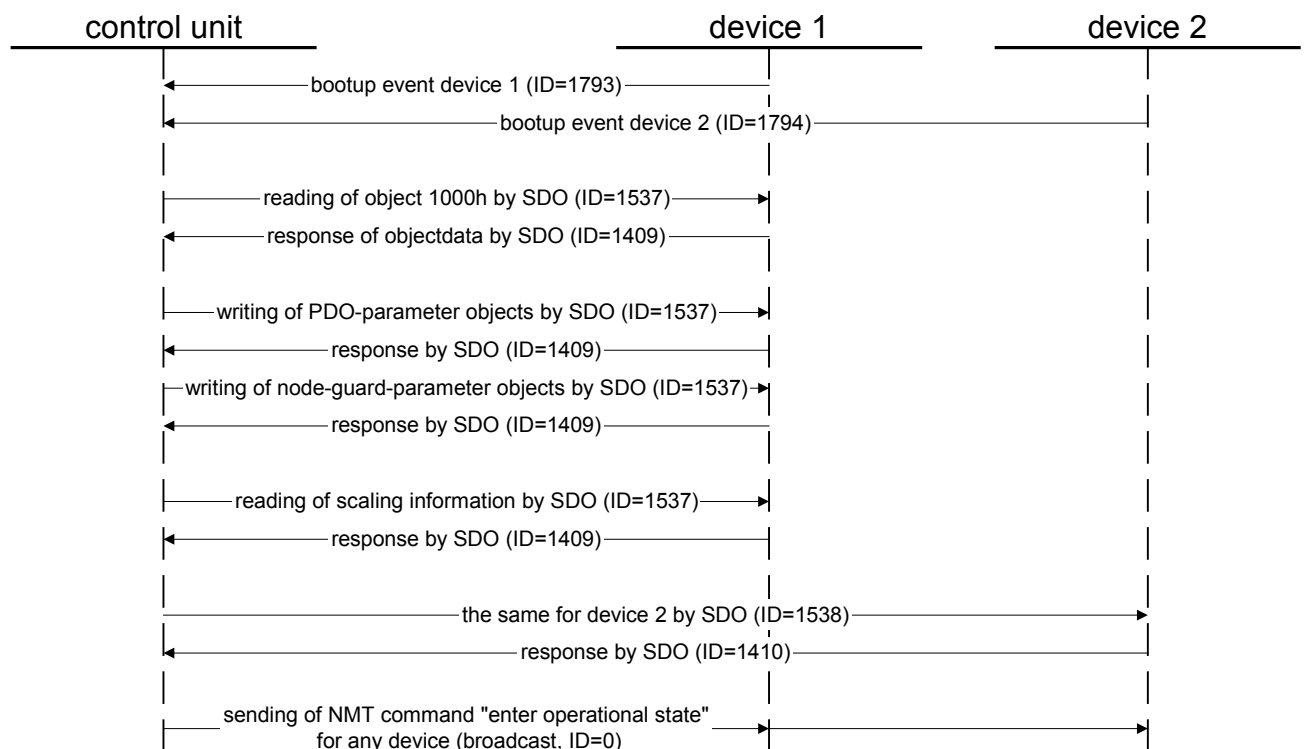
For the control to be able to communicate with the peripheral device, baud rate as well as Node number (Node-ID) must correspond with the value specified by the control.

The baudrate for EUROMAP 66 devices is fixed to 250kbaud. The Node ID must be set at the device as per the pertaining setting instructions.

4.3 Connection setup and initialization

On taking up the connection with the device, the control should first check whether or not the device exists and whether e.g. the device type meets the expectations. This can be done by reading of object 1000h (device type) and 2000h (Eoromap66 Device Profile). However, it is also possible to check further objects (e.g. information on the manufacturer).

An initialization sequence might look as follows:

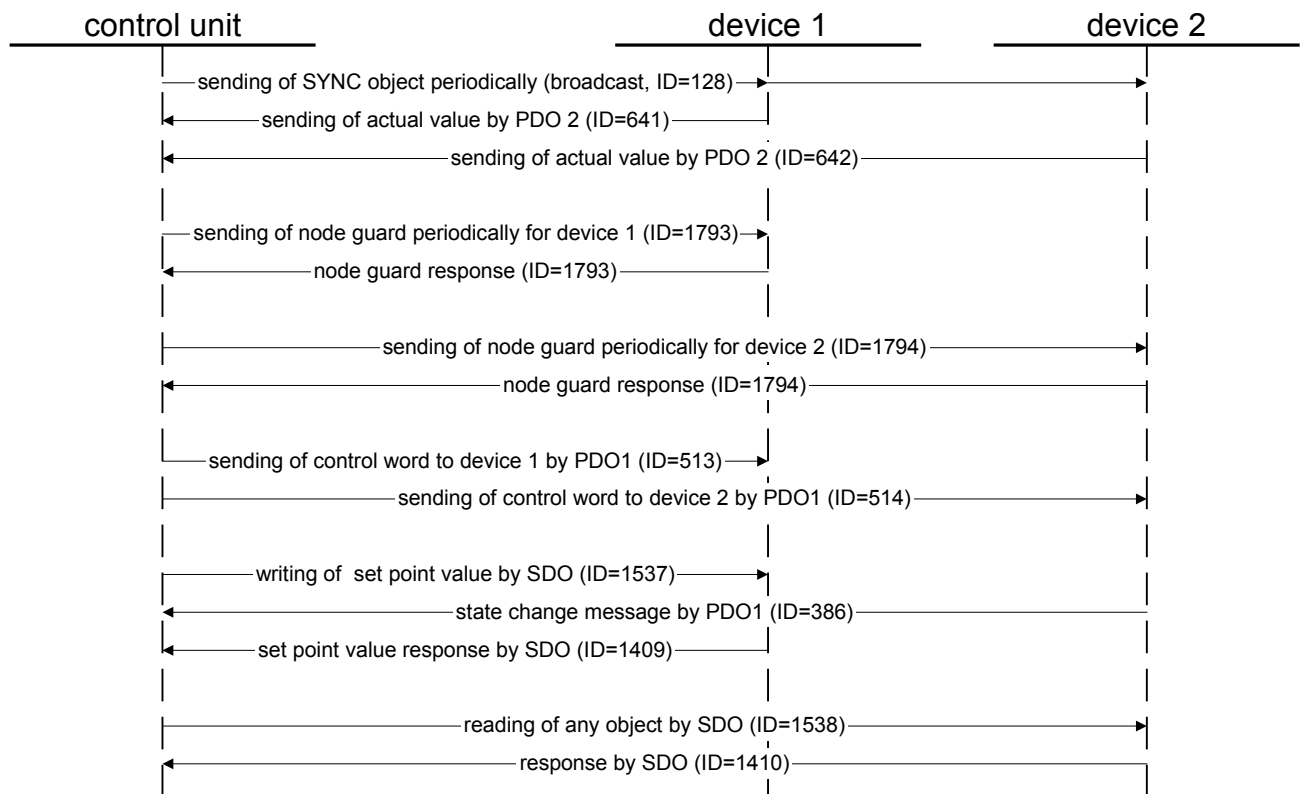


With the PDO parameters it is determined which objects are contained in the PDO and when the latter will be transmitted. The transmission can be e.g. event-driven (i.e. in case of an alteration of the object contained in the PDO), or cyclically (e.g. after every fifth SYNC message).

When switched on, the device sends a bootup event message. If interpreted by the control and reproduced onto a display, this permits to detect very quickly whether or not a device is correctly wired and whether or not the Node ID setting is correct.

4.4 Operation

There are many possibilities for communication in operation. A typical sequence might look as follows:



To be observed: Messages with a low identifier are transmitted with higher priority. So a PDO message may be transmitted between SDO request and SDO response.

On the basis of the ID the control detects what type of message is concerned and from which device it comes.

4.5 PDO- and SDO- usage

As seen in 3.2.1 CANopen communication model, data exchange between device and control is done by (confirmed) SDOs and (unconfirmed) PDOs.

With each SDO, one object can be read or written. Because SDO communication is confirmed, the device is able to reject invalid values, such as values out of limit or inadequate mode words, and the control will immediately recognize it. Thus, this is the preferred communication for the setting of mode words and (rarely changing) setpoints.

PDOs, in contrary, may contain several objects. They may be transmitted cyclic, acyclic, synchronous or asynchronous (e.g. if a status bit changes). Thus, PDOs are mainly used for fast and spontaneous exchange of control and status information. They may also be used for cyclic exchange of effective values and often changing setpoints.

Because PDO communication is not confirmed, error mechanisms have to be implemented by the application, e.g. with additional objects. In EUROMAP 66, this is the case e.g. with control word and control errors, but not with setpoints, etc.

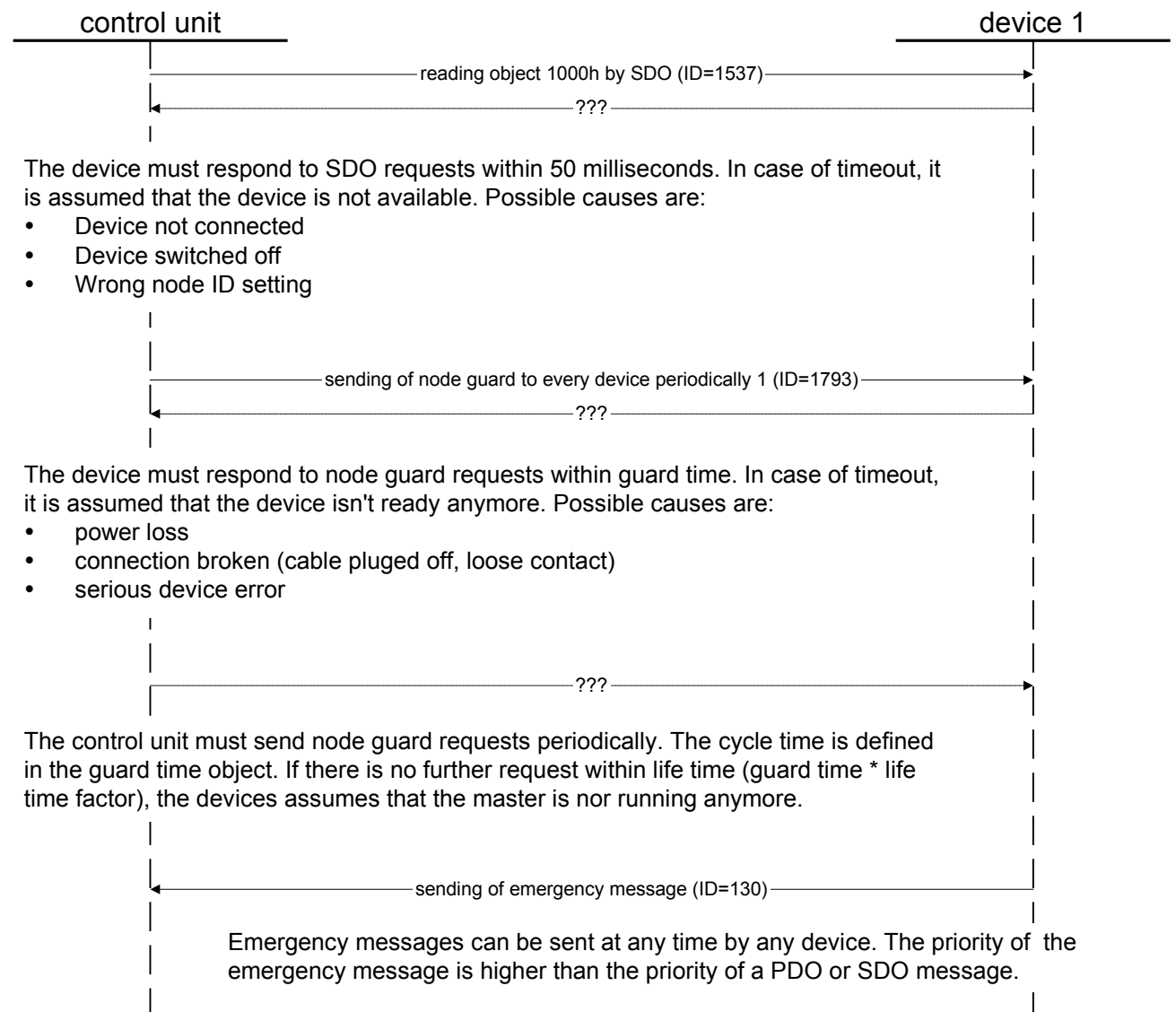
Thus, on setting up the communication (PDO-mapping), the following is recommended:

- Use SDO communication for the setting of mode words and (rarely changing) setpoints.
- Use PDOs for fast and spontaneous exchange of control and status information.

4.6 Treatment of communication errors

CANopen defines highly efficient error detection possibilities. This means that devices can transmit error messages at any time with a high message priority (emergency).

If node guarding is activated, both control and device are able to identify an interruption of the connection or a failure of the counter side.



5 Summary

This document describes the EUROMAP 66 peripheral device interface. The interface is based on CANopen.

EUROMAP 66 interface characteristics:

- A EUROMAP 66 device is in principle a CANopen "Closed Loop Controller".
- It defines specific extensions to the CANopen device profile.
- It defines different device profiles (for the time being only for temperature control units, however).

The EUROMAP 66 device interface offers various advantages:

Advantages of the CAN bus:

- Very high transmission safety
- Fast (real-time capability)
- Multi-master principle
- Economical
- Industrial standard

Advantages of the CANopen protocol:

- Standardized communication mechanisms
- Standardized, extendable device profiles
- Supported and used by many producers
- Low-cost integration by various SW suppliers
- The EUROMAP 66 devices can be operated jointly with other CANopen devices at the same bus

Advantages of the EUROMAP 66 device interface:

- It is based on the CANopen standard
- It is suitable for easy integration in the control
- Manufacturer specific options are not supported
- In a guaranteed manner simply to use
- The Scope of all objects is clearly determined

EUROMAP

Europäisches Komitee der Hersteller von Kunststoff- und Gummimaschinen

European Committee of Machinery Manufacturers for the Plastics and Rubber Industries

Comité Européen des Constructeurs de Machines pour Plastiques et Caoutchouc

Comitato Europeo Costruttori Macchine per Materie Plastiche e Gomma

See you again

<http://www.euromap.org>

Copyright by EUROMAP