CiA 309

Access from other networks

Part 1: Network access services

Version: 1.1.0

12 December 2006

HISTORY

Date	Changes
2004-09-15	*Publication of version 1.0 as draft standard proposal*
2006-12-12	*Publication of version 1.1 as draft standard*
	Minor editorial corrections and clarifications.
	Multiplexor parameter for PDO access services added.

General information on licensing and patents

CAN in AUTOMATION (CiA) calls attention to the possibility that some of the elements of this CiA specification may be subject of patent rights. CiA shall not be responsible for identifying any or all such patent rights.

Because this specification is licensed free of charge, there is no warranty for this specification, to the extent permitted by applicable law. Except when otherwise stated in writing the copyright holder and/or other parties provide this specification “as is” without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the correctness and completeness of the specification is with you. Should this specification prove failures, you assume the cost of all necessary servicing, repair or correction.

Trademarks

CANopen® and CiA® are registered community trademarks of CAN in Automation. The use is restricted for CiA members or owners of CANopen vendor ID. More detailed terms for the use are available from CiA.

CAN in Automation e. V.
Kontumazgarten 3
DE - 90429 Nuremberg, Germany
Tel.: +49-911-928819-0
Fax: +49-911-928819-79
Url: www.can-cia.org
Email: headquarters@can-cia.org

Contents

1 Scope

This specification specifies the services and protocols to interface CANopen networks to a TCP/IP-based network.

This set of specifications is organized as follows:

- Part 1: General principles and services
- Part 2: Modbus/TCP mapping
- Part 3: ASCII mapping

Part 1 of the *Interfacing CANopen with TCP/IP* specification specifies the network access services provided by a gateway device that give one network device (e.g. PLC or PC application) connected through *transmission control protocol/internet protocol* (TCP/IP) or other protocols (based on Ethernet, Remote Access Services, or serial links) access to devices attached to CANopen networks. The description of the transport protocol between the gateway device and the network devices using the services specified in this document is outside of the scope of this part of the specification.

2 References

/CiA301/ CiA 301, CANopen application layer and communication profile

/CiA302-2/ CiA 302-2, CANopen additional application layer functions – Part 2: Network management

/CiA302-3/ CiA 302-3, CANopen additional application layer functions – Part 3: Configuration and program download

/CiA400/ CiA 400, CANopen interface profile multi-level networking

3 Abbreviations and definitions

3.1 Abbreviations

NMT	Network Management
PAS	PDO Access Services
PDO	Process Data Object
RPDO	Receive PDO
RTR	Remote Transmission Request
SAS	SDO Access Services
SDO	Service Data Object
SRD	SDO Requesting Device
TPDO	Transmit PDO

3.2 Definitions

3.2.1 Introduction

Figure 1 shows the relations between the functional elements described in this specification.

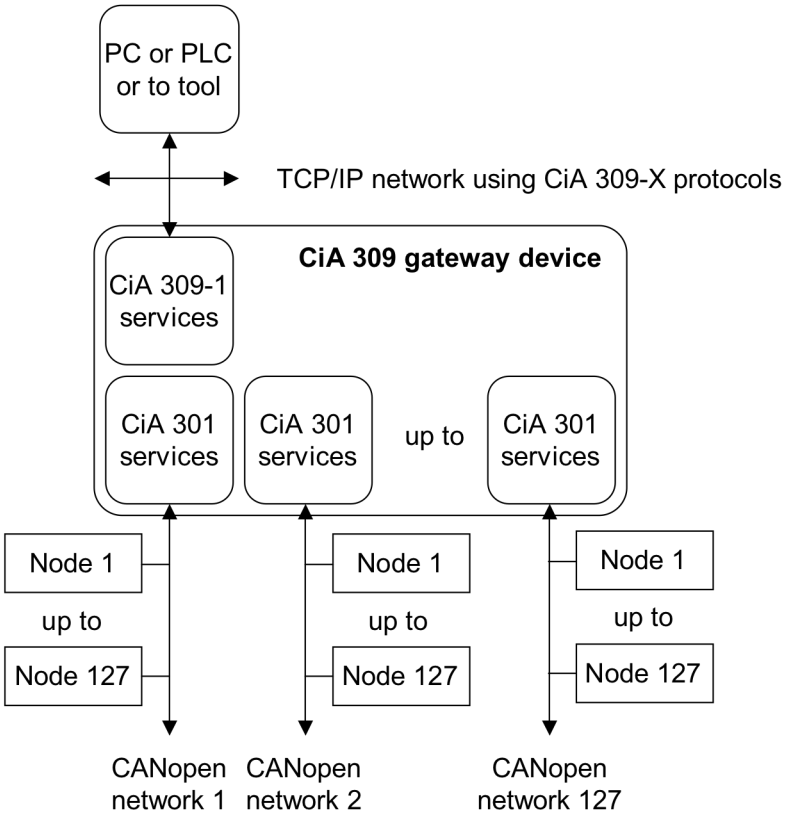

Fig. 1: Functional overview of elements referenced in this specification

3.2.2 Gateway class definitions

A CANopen gateway may support one or more of the following classes:

Class 1: The gateway is a device, acting as network slave (NMT slave functionality) within the CANopen network. The device shall provide SDO client functionality.

Class 2: The gateway is a device implementing the functionality of a class1 device, which additionally implements SDO requesting device (SRD) functionality.

Class 3: The gateway is a device within the CANopen network acting as the CANopen manager.

3.2.3 Service primitives definitions

Service primitives are the means by which the gateway application and the network application layer interact. There are four primitives:

- A *request* is issued by the gateway application to the network application layer to request a service.
- An *indication* is issued by the network application layer to the gateway application to report an internal event detected by the network application layer or indicate that a service is requested
- A *response* is issued by the gateway application to the network application layer to respond to a previously received indication
- A *confirmation* is issued by the network application layer to the gateway application to report the result of a previously issued request.

A service type defines the primitives that are exchanged between the network application layer and the gateway application for a particular service of a service object.

- A *local service* involves only the local service object. The gateway application issues a request to its local service object that executes the requested service without communicating with (a) peer service object(s).

Fig. 2: Local service

- An *unconfirmed service* involves one or more peer service objects. The gateway application or the network device application issues a request to its local service object. This request is transferred to the peer service object(s) that each pass it to its (their) application as an indication.

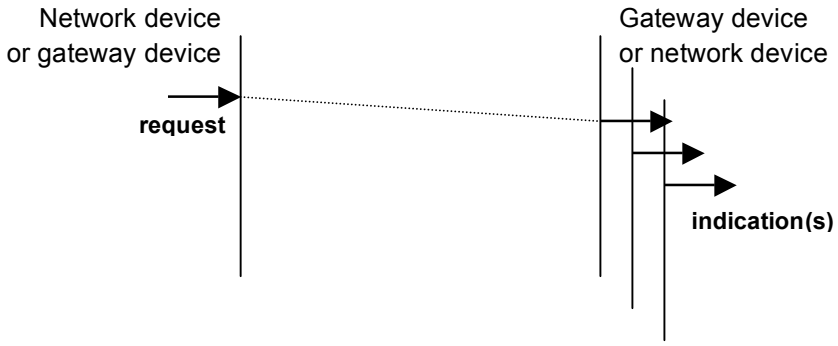

Fig. 3: Unconfirmed service

- A *confirmed service* involves only to the peer service object. The network device application or the gateway application issues a request to its local service object. This request is transferred to the peer service object that passes it to the network device application respectively to the gateway application as an indication. The network device application or the gateway applications issues a response that is transferred to the originating service object that passes it as a confirmation to the requesting service. This event is then indicated to the gateway application respectively to the network device application.

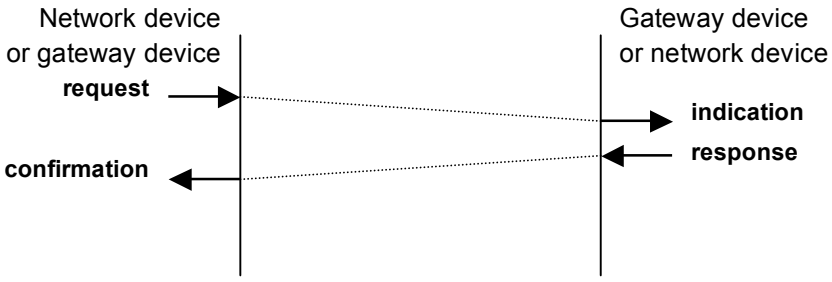

Fig. 4: Confirmed service

- A *provider-initiated service* involves only the local service object. The service object (being the service provider) detects an event not solicited by a requested service. This event is then indicated to the gateway application.

Fig. 5: Provider initiated service

Unconfirmed and confirmed services are called *remote access*.

3.2.4 Parameter definitions

COB-ID

Includes CAN-ID and some control bits as defined in /CiA301/.

Data

Data read or written from the network device.

Data type

Type of data to be read or written.

Data type 1st to 64th

Type of data of the objects mapped into a PDO.

Emergency

Data received by a CANopen emergency message as defined in /CiA301/, It includes the emergency error code, the error register, and the manufacturer-specific error field as defined in /CiA301/.

Error msg number

Error code indicating the kind of failure.

Error msg text

Textual description of the failure.

Failure

Information that indicates that the service has been not completed successfully.

Gateway class

Information that declares the gateway device's functionality (*see clause 3.1.1*).

GuardTime

Time value as provided in the guard_time object as defined in /CiA301/.

HeartbeatConsumerTime

Time value as provided in the heartbeat_consumer_time objects as defined in /CiA301/.

HeartbeatProducerTime

Time value as provided in the heartbeat_producer_time object as defined in /CiA301/.

Implementation class

Information that is transport protocol specific and is defined in the other parts of this specification.

Length

Length of data to be read or written.

LifeTimefactor

Time value as provided in life_time_factor object as defined in /CiA301/.

Multiplexor

Index and sub-index to access the CANopen object dictionary entry.

Network

Number identifying uniquely the CANopen sub-network (see /CiA400/. If this parameter is not supported, the service request applies to the unique or default network.

Nbr_objects

Number of objects mapped into a PDO.

Node-ID

Number identifying uniquely the device in a CANopen network. If this parameter is not supported, the service request applies to the default node-ID. If this parameter is supported, it shall become the default node-ID for the next service requests.

Offset

Offset value for direct access to a CANopen object. This parameter is limited to objects of type OCTET_STRING, VISIBLE_STRING, UNICODE_STRING and DOMAIN. The unit shall be UNSIGNED32. If this parameter is not supported, the object shall be read or written from offset 0.

Product code

Unique number identifying the device assigned by the manufacturer as defined in /CiA301/.

Protocol version

number that indicates the CiA 309-1 version the gateway device is based on.

Reason

Code providing information why the service was not successful.

Revision number

Unique number identifying the device's version assigned by the manufacturer as defined in /CiA301/.

SDO timeout

Time value for the time-out of all SDO clients.

Serial number

Unique number identifying the device within its class assigned by the manufacturer as defined in /CiA301/.

Success

Information that indicates that the service has been completed successfully.

TxType

Transmission type code as defined in /CiA301/.

Vendor-ID

Unique identifier assigned to a company by CiA as defined in /CiA301/.

3.2.5 CANopen network, device and object addressing

The gateway device may support more than one CANopen network. If multiple CANopen network interfaces are implemented, the CANopen networks shall be numbered uniquely (e.g. CANopen Net 1, CANopen Net 2, etc.).

The CANopen networks may be of one-level or multi-level type. One-level networks are connected physically to the gateway device. Multi-level networks are connected logically by means of cascaded network architectures. Multi-level networks are reached via the SDO network indication service as defined in /CiA400/. A routing table shall be used as defined in /CiA400/ in order to map gateway devices in logical networks to the physical port.

In each CANopen network the connected CANopen devices are uniquely addressed by the node-ID (see also /CiA301/). The gateway device itself also provides its own unique node-ID.

In each CANopen device the objects are addressed uniquely by the 16-bit index and the 8-bit sub-index as defined in /CiA301/.

4 Network access service specification

4.1 SDO access services

4.1.1 General

The services specified in this clause are used to initiate and configure SDO services accessing any object in the object dictionary of any node on any of the CANopen networks linked to the gateway device.

In case that the gateway device supports multi-level networking as defined in /CiA400/, the network service indication shall be handled transparently.

4.1.2 Upload SDO

This service shall initiate an SDO upload service.

Parameter	*Indication*	*Response*
Argument	**Mandatory**	
Network	Optional	
Node-ID	Optional	
Multiplexor	Mandatory	
Data type	Optional	
BOOLEAN	Selection	
UNSIGNED8	Selection	
UNSIGNED16	Selection	
UNSIGNED24	Selection	
UNSIGNED32	Selection	
UNSIGNED40	Selection	
UNSIGNED48	Selection	
UNSIGNED56	Selection	
UNSIGNED64	Selection	
INTEGER8	Selection	
INTEGER16	Selection	
INTEGER24	Selection	
INTEGER32	Selection	
INTEGER40	Selection	
INTEGER48	Selection	
INTEGER56	Selection	
INTEGER64	Selection	
REAL32	Selection	
REAL64	Selection	
TIME_OF_DAY	Selection	
TIME_DIFFERENCE	Selection	
OCTET_STRING	Selection	
VISIBLE_STRING	Selection	
UNICODE_STRING	Selection	
DOMAIN	Selection	
Offset	Optional	
Length	Optional	
Remote result		**Mandatory**
Success		Selection
Data		Mandatory
Length		Optional
Failure		Selection
Reason		Mandatory

4.1.3 Download SDO

This service shall initiate an SDO download service.

Parameter	*Indication*	*Response*
Argument	**Mandatory**	
Network	Optional	
Node-ID	Optional	
Multiplexor	Mandatory	
Data type	Optional	
BOOLEAN	Selection	
UNSIGNED8	Selection	
UNSIGNED16	Selection	
UNSIGNED24	Selection	
UNSIGNED32	Selection	
UNSIGNED40	Selection	
UNSIGNED48	Selection	
UNSIGNED56	Selection	
UNSIGNED64	Selection	
INTEGER8	Selection	
INTEGER16	Selection	
INTEGER24	Selection	
INTEGER32	Selection	
INTEGER40	Selection	
INTEGER48	Selection	
INTEGER56	Selection	
INTEGER64	Selection	
REAL32	Selection	
REAL64	Selection	
TIME_OF_DAY	Selection	
TIME_DIFFERENCE	Selection	
OCTET_STRING	Selection	
VISIBLE_STRING	Selection	
UNICODE_STRING	Selection	
DOMAIN	Selection	
Offset	Optional	
Length	Optional	
Data	Mandatory	
Remote result		**Mandatory**
Success		Selection
Failure		Selection
Reason		Mandatory

4.1.4 Configure SDO timeout

This service shall configure the time-out for all Client-SDOs on the gateway device.

Parameter	*Indication*	*Response*
Argument	**Mandatory**	
Network	Optional	
SDO Timeout	Mandatory	
Remote result		**Mandatory**
Success		Selection
Failure		Selection
Reason		Mandatory

4.2 PDO access services

4.2.1 General

The services specified in this clause are used to configure and initiate PDO services in the gateway device. They include:

- Configure a receive PDO
- Configure a transmit PDO
- Request to read a PDO
- Request to write a PDO
- Indicate a received PDO

The two PDO configuration services are intended to create PDOs in the gateway device. If the gateway device implements an object dictionary, the PDO communication and mapping parameters entries shall be set-up accordingly.

The two PDO request services are intended to control the PDOs in accordance to the configured PDO transmission type /CiA301/.

The data types VISIBLE_STRING, OCTET_STRING, and UNICODE_STRING, as well as DOMAIN shall not be used as PAS data type parameter.

NOTE: The PAS services are not intended to configure the PDO communication and mapping entries of the object dictionary of the remote nodes. Accessing the CANopen nodes individually by means of SAS services may do the PDO configuration.

4.2.2 Configure RPDO

This service shall create an RPDO in the gateway device.

Parameter	***Indication***	***Response***
Argument Network PDO number COB-ID TxType Nbr_objects 1st mapped object Data type Multiplexor 2nd mapped object Data type Multiplexor … 64th mapped object Data type Multiplexor	**Mandatory** Optional Mandatory Mandatory Mandatory Mandatory Mandatory Selection Selection Optional Selection Selection … Optional Selection Selection	
Remote result Success Failure Reason		**Mandatory** Selection Selection Mandatory

4.2.3 Configure TPDO

This service shall create a TPDO in the gateway device.

Parameter	*Indication*	*Response*
Argument	**Mandatory**	
Network	Optional	
PDO number	Mandatory	
COB-ID	Mandatory	
TxType	Mandatory	
Nbr_objects	Mandatory	
1st mapped object	Mandatory	
Data type	Selection	
Multiplexor	Selection	
2nd mapped object	Optional	
Data type	Selection	
Multiplexor	Selection	
…	…	
64th mapped object	Optional	
Data type	Selection	
Multiplexor	Selection	
Remote result		**Mandatory**
Success		Selection
Failure		Selection
Reason		Mandatory

4.2.4 Read PDO data

This service shall read the data received by an RPDO. If an RPDO is configured with transmission type 252 or 253 /CiA301/, the gateway devices shall trigger it by means of an RTR.

Note: It is not recommended to use RTR.

The received data shall be transmitted in the remote result.

Parameter	*Indication*	*Response*
Argument	**Mandatory**	
Network	Optional	
PDO number	Mandatory	
Remote result		**Mandatory**
Success		Selection
Network		Optional
PDO number		Mandatory
Nbr_objects		Mandatory
Data 1st object		Conditional
…		…
Data 64th object		Conditional
Failure		Selection
Reason		Mandatory

4.2.5 Write PDO data

This service shall trigger the transmission of a PDO. The actual transmission of the PDO shall be triggered according to the configured PDO transmission type /CiA301/.

Parameter	*Indication*	*Response*
Argument Network PDO number Nbr_objects Data 1st object … Data 64th object	**Mandatory** Optional Mandatory Mandatory Conditional … Conditional	
Remote result Success Failure Reason		**Mandatory** Selection Selection Mandatory

4.2.6 RPDO received

This service shall signal that new PDO data has been received and shall provide the received data.

Parameter	*Request*
Argument Network PDO number Nbr_objects 1st object value … 64th object value	**Mandatory** Optional Mandatory Mandatory Conditional … Conditional

4.3 CANopen NMT services

4.3.1 General

The services specified in this clause are used to control a CANopen node or a CANopen network and associated error control services.

4.3.2 Start node

This service shall set CANopen nodes into NMT state OPERATIONAL. For Class 1 and Class 2 devices this service shall trigger a CANopen Request NMT service /CiA302-2/ for Class 3 devices the service shall trigger a Start Remote Node service /CiA301/.

Parameter	*Indication*	*Response*
Argument Network Node-ID All	**Mandatory** Optional Selection Selection	
Remote result Success Failure Reason		**Mandatory** Selection Selection Mandatory

4.3.3 Stop node

This service shall set CANopen nodes into NMT state STOPPED. For class 1 and class 2 devices this service shall trigger a CANopen request NMT service /CiA302-2/ for class 3 devices the service shall trigger a stop remote node service /CiA301/.

Note: The remote result for class 1 and 2 devices is only the confirmation of the SDO request; for class 3 devices the remote result is based on the error control services as defined in /CiA301/.

Parameter	*Indication*	*Response*
Argument Network Node-ID All	**Mandatory** Optional Selection Selection	
Remote result Success Failure Reason		**Mandatory** Selection Selection Mandatory

4.3.4 Set node to pre-operational

This service shall set CANopen nodes into NMT state PRE-OPERATIONAL. For class 1 and class 2 devices this service shall trigger a CANopen request NMT service /CiA302-2/ for class 3 devices the service shall trigger an Enter Pre-operational service /CiA301/.

Parameter	*Indication*	*Response*
Argument Network Node-ID All	**Mandatory** Optional Selection Selection	
Remote result Success Failure Reason		**Mandatory** Selection Selection Mandatory

4.3.5 Reset node

This service shall set CANopen nodes into NMT state RESET APPLICATION. For class 1 and class 2 devices this service shall trigger a CANopen request NMT service as specified in /CiA302-2/ for class 3 devices the service shall trigger a reset node service.

Parameter	*Indication*	*Response*
Argument Network Node-ID All	**Mandatory** Optional Selection Selection	
Remote result Success Failure Reason		**Mandatory** Selection Selection Mandatory

4.3.6 Reset communication

This service shall set CANopen nodes into NMT state RESET COMMUNICATION. For class 1 and class 2 devices this service shall trigger a CANopen request NMT service as specified /CiA302-2/ for class 3 devices the service shall trigger a reset communication service.

Parameter	Indication	Response
Argument Network Node-ID All	**Mandatory** Optional Selection Selection	
Remote result Success Failure Reason		**Mandatory** Selection Selection Mandatory

4.3.7 Enable node guarding

This service is only available for class 3 devices. It shall start node guarding for the device specified by node-ID with the parameters given by GuardTime and LifeTimeFactor. If heartbeat is already activated on the addressed node, the service request shall be rejected.

Parameter	*Indication*	*Response*
Argument Network Node-ID GuardTime LifeTimefactor	**Mandatory** Optional Mandatory Mandatory Mandatory	
Remote result Success Failure Reason		**Mandatory** Selection Selection Mandatory

4.3.8 Disable node guarding

This service is only available for class 3 devices. It shall stop node guarding for the device specified by node-ID.

Parameter	*Indication*	*Response*
Argument Network Node-ID	**Mandatory** Optional Mandatory	
Remote result Success Failure Reason		**Mandatory** Selection Selection Mandatory

4.3.9 Start heartbeat consumer

This service shall start the consumption of heartbeat messages transmitted by a CANopen device specified by node-ID.

Parameter	*Indication*	*Response*
Argument Network Node-ID HeartbeatConsumerTime	**Mandatory** Optional Mandatory Mandatory	
Remote result Success Failure Reason		**Mandatory** Selection Selection Mandatory

4.3.10 Disable heartbeat consumer

This service shall stop the consumption of heartbeat messages transmitted by a CANopen device specified by node-ID.

Parameter	*Indication*	*Response*
Argument Network Node-ID	**Mandatory** Optional Mandatory	
Remote result Success Failure Reason		**Mandatory** Selection Selection Mandatory

4.3.11 Error control event received

This service shall signal the NMT status or error control events received from a CANopen node specified by node-ID.

Parameter	*Request*
Argument	**Mandatory**
Network	Optional
Node-ID	Mandatory
Status	Selection
Error Code	Selection

4.4 Device failure management services

4.4.1 General

The services specified in this clause are used to manage failures within the gateway device or within any other CANopen device.

4.4.2 Read device error

This service shall read EMCY message information received from the CANopen device specified by the node-ID parameter.

Parameter	*Indication*	*Response*
Argument	**Mandatory**	
Network	Optional	
Node-ID	Optional	
Remote result		**Mandatory**
Success		Selection
Network		Optional
Node-ID		Mandatory
Error		Selection
Error Msg number		Mandatory
Error Msg text		Optional
Emergency		Selection
Emergency code		Mandatory
Error register		Optional
Manufacturer error		Optional
Failure		Selection
Reason		Mandatory

4.4.3 Emergency event received

This service shall signal the reception of an emergency message in the gateway device transmitted by a CANopen device specified by the node-ID.

Parameter	*Request*
Argument	**Mandatory**
Network	Optional
Node-ID	Mandatory
Emergency code	Mandatory
Error register	Mandatory
Manufacturer error	Optional

4.5 CANopen interface configuration services

4.5.1 General

The services described in this clause are used to configure and parameterize the CANopen interface of the gateway device.

4.5.2 Initialize gateway

This service shall initiate the CANopen initialization of the gateway device. It shall perform a power-on equivalent reset of the CANopen interface. It is used to initialize the bit-timing parameters.

Parameter	*Indication*	*Response*
Argument Network CAN bit timing	**Mandatory** Optional Optional	
Remote result Success Failure Reason		**Mandatory** Selection Selection Mandatory

4.5.3 Store configuration

This service shall command the gateway device to store its CANopen interface configuration.

Parameter	*Indication*	*Response*
Argument Network Storage specifier	**Mandatory** Optional Optional	
Remote result Success Failure Reason		**Mandatory** Selection Selection Mandatory

4.5.4 Restore configuration

This service shall command the gateway device to restore its CANopen interface configuration.

Parameter	*Indication*	*Response*
Argument Network Storage specifier	**Mandatory** Optional Optional	
Remote result Success Failure Reason		**Mandatory** Selection Selection Mandatory

4.5.5 Set heartbeat producer

This service shall set the CANopen heartbeat producer time in the gateway device.

Parameter	*Indication*	*Response*
Argument Network Node-ID HeartbeatProducerTime	**Mandatory** Optional Mandatory Mandatory	
Remote result Success Failure Reason		**Mandatory** Selection Selection Mandatory

4.5.6 Set node-ID

This service shall set the CANopen node-ID in the gateway device for the CANopen network given in the *network* parameter.

Parameter	*Indication*	*Response*
Argument Network Node-ID	**Mandatory** Optional Mandatory	
Remote result Success Failure Reason		**Mandatory** Selection Selection Mandatory

4.5.7 Start emergency consumer

This service shall start the consumption of emergency messages. The relation between node-ID producing an emergency message and the COB-ID has to be explicitly known.

Parameter	*Indication*	*Response*
Argument Network Node-ID COB-ID	**Mandatory** Optional Mandatory Mandatory	
Remote result Success Failure Reason		**Mandatory** Selection Selection Mandatory

4.5.8 Stop emergency consumer

This service shall stop the consumption of emergency messages. The relation between node-ID producing an emergency message and the COB-ID has to be explicitly known.

Parameter	*Indication*	*Response*
Argument Network Node-ID COB-ID	**Mandatory** Optional Mandatory Mandatory	
Remote result Success Failure Reason		**Mandatory** Selection Selection Mandatory

4.6 Gateway management services

4.6.1 General

The services specified in this clause are used to manage the gateway device.

4.6.2 Set default network

This service shall set the default network number, which shall be used for all services.

Parameter	*Indication*	*Response*
Argument DefaultNetwork	**Mandatory** Mandatory	
Remote result Success Failure		**Mandatory** Selection Selection

Reason		Mandatory

4.6.3 Set default node-ID

This service shall set the default node-ID, which shall be used for all services.

Parameter	*Indication*	*Response*
Argument Default node-ID	**Mandatory** Mandatory	
Remote result Success Failure Reason		**Mandatory** Selection Selection Mandatory

4.6.4 Get version

This service shall get information on the gateway device and its CANopen interface.

Parameter	*Indication*	*Response*
Argument Network	**Mandatory** Optional	
Remote Result Success Vendor-ID Product code Revision number Serial number Gateway class Protocol version Implementation class Failure Reason		**Mandatory** Selection Mandatory Mandatory Mandatory Mandatory Mandatory Mandatory Mandatory Selection Mandatory

4.7 Controller management services

4.7.1 General

The services specified in this clause are used to manage a programmable controller implemented within the gateway device or in a remote node on a CANopen sub-network.

Upload and download of the controller program shall be done using SDO upload and SDO write services.

For a programmable controller on a remote node, the object dictionary entries specified in /CiA302-3/ for program control shall be implemented and the gateway shall translate the service into a SDO access to these object dictionary entries.

4.7.2 Reset controller

This service shall reset the controller function.

Parameter	*Indication*	*Response*
Argument Network Node-ID	**Mandatory** Optional Optional	
Remote result Success Failure Reason		**Mandatory** Selection Selection Mandatory

4.7.3 Start controller

This service shall switch the controller function to RUN.

Parameter	*Indication*	*Response*
Argument Network Node-ID	**Mandatory** Optional Optional	
Remote result Success Failure Reason		**Mandatory** Selection Selection Mandatory

4.7.4 Stop controller

This service shall switch the controller function to STOP.

Parameter	*Indication*	*Response*
Argument Network Node-ID	**Mandatory** Optional Optional	
Remote result Success Failure Reason		**Mandatory** Selection Selection Mandatory

4.8 Manufacturer-specific services

4.8.1 General

The manufacturer of a gateway device may define additional services.

