Cat. No. W904-E2-2

C200HW-CORT21-V1

CANopen Slave Unit

OPERATION MANUAL

OMRON

C200HW-CORT21-V1
CANopen Slave Unit
Operation Manual

Produced June 2001

Notice:

OMRON products are manufactured for use according to proper procedures by a qualified operator and only for the purposes described in this manual.

The following conventions are used to indicate and classify precautions in this manual. Always heed the information provided with them. Failure to heed precautions can result in injury to people or damage to the product.

⚠ **DANGER!** Indicates information that, if not heeded, is likely to result in loss of life or serious injury.

⚠ **WARNING** Indicates information that, if not heeded, could possibly result in loss of life or serious injury.

⚠ **Caution** Indicates information that, if not heeded, could result in relatively serious or minor injury, damage to the product, or faulty operation.

OMRON Product References

All OMRON products are capitalized in this manual. The word “Unit” is also capitalized when it refers to an OMRON product, regardless of whether or not it appears in the proper name of the product.

The abbreviation “Ch,” which appears in some displays and on some OMRON products, often means “word” and is abbreviated “Wd” in documentation in this sense.

The abbreviation “PLC” means Programmable Controller and is not used as an abbreviation for anything else.

Visual Aids

The following headings appear in the left column of the manual to help you locate different types of information.

Note Indicates information of particular interest for efficient and convenient operation of the product.

1, 2, 3… 1. Indicates lists of one sort or another, such as procedures, checklists, etc.

TABLE OF CONTENTS

Appendices

About this Manual:

This manual describes the installation and operation of the C200HW-CORT21-V1 CANopen Slave Unit and includes the sections described below.

Please read this manual carefully and be sure you understand the information provided before attempting to install and operate the CANopen Slave Unit. **Be sure to read the precautions provided in the following section.**

Section 1 gives a brief description of CANopen.

Section 2 describes the overall specifications and the communication performance of the Unit.

Section 3 describes the installation of the C200HW-CORT21-V1.

Section 4 describes the interface with the user via the PLC system.

Section 5 describes the message communication.

Section 6 gives an overview of the implemented error handling and status mechanisms.

Section 7 describes the troubleshooting procedures and maintenance operations.

⚠ **WARNING**	Failure to read and understand the information provided in this manual may result in personal injury or death, damage to the product, or product failure. Please read each section in its entirety and be sure you understand the information provided in the section and related sections before attempting any of the procedures or operations given.

PRECAUTIONS

This section provides general precautions for using the CANopen Slave Units, Programmable Controllers, and related devices.

The information contained in this section is important for the safe and reliable application of the CANopen Slave Units. You must read this section and understand the information contained before attempting to set up or operate a CANopen Slave Unit and PLC system.

1 Intended Audience

This manual is intended for the following personnel, who must also have knowledge of electrical systems (an electrical engineer or the equivalent).

- Personnel in charge of installing FA systems.
- Personnel in charge of designing FA systems.
- Personnel in charge of managing FA systems and facilities.

2 General Precautions

The user must operate the product according to the performance specifications described in the operation manuals.

Before using the product under conditions which are not described in the manual or applying the product to nuclear control systems, railroad systems, aviation systems, vehicles, combustion systems, medical equipment, amusement machines, safety equipment, and other systems, machines, and equipment that may have a serious influence on lives and property if used improperly, consult your OMRON representative.

Make sure that the ratings and performance characteristics of the product are sufficient for the systems, machines, and equipment, and be sure to provide the systems, machines, and equipment with double safety mechanisms.

This manual provides information for installing and operating OMRON CANopen Slave Units. Be sure to read this manual before operation and keep this manual close at hand for reference during operation.

⚠ **WARNING** It is extremely important that a PLC and all PLC Units be used for the specified purpose and under the specified conditions, especially in applications that can directly or indirectly affect human life. You must consult with your OMRON representative before applying a PLC system to the above mentioned applications.

3 Safety Precautions

⚠ **WARNING** Never attempt to disassemble any Units while power is being supplied. Doing so may result in serious electrical shock or electrocution.

⚠ **WARNING** Never touch any of the terminals while power is being supplied. Doing so may result in serious electrical shock or electrocution.

4 Operating Environment Precautions

Do not operate the control system in the following places.

- Where the PLC is exposed to direct sunlight.
- Where the ambient temperature is below 0°C or over 55°C.
- Where the PLC may be affected by condensation due to radical temperature changes.
- Where the ambient humidity is below 10% or over 90%.
- Where there is any corrosive or inflammable gas.
- Where there is excessive dust, saline air, or metal powder.
- Where the PLC is affected by vibration or shock.
- Where any water, oil, or chemical may splash on the PLC.

⚠ **Caution** The operating environment of the PLC System can have a large effect on the longevity and reliability of the system. Improper operating environments can lead to malfunction, failure, and other unforeseeable problems with the PLC System. Be sure that the operating environment is within the specified conditions at installation and remains within the specified conditions during the life of the system.

5 Application Precautions

Observe the following precautions when using the CANopen Salve Units or the PLC.

⚠ **WARNING** Failure to abide by the following precautions could lead to serious or possibly fatal injury. Always heed these precautions.

- Always ground the system to 100 Ω or less when installing the system to protect against electrical shock.
- Always turn OFF the power supply to the PLC before attempting any of the following. Performing any of the following with the power supply turned ON may lead to electrical shock:
  - Mounting or removing any Units (e.g., I/O Units, CPU Unit, etc.) or memory cassettes.
  - Assembling any devices or racks.
  - Connecting or disconnecting any cables or wiring.

⚠ **Caution** Failure to abide by the following precautions could lead to faulty operation of the PLC or the system or could damage the PLC or PLC Units. Always heed these precautions.

- Use the Units only with the power supplies and voltages specified in the operation manuals. Other power supplies and voltages may damage the Units.
- Take measures to stabilise the power supply to conform to the rated supply if it is not stable.
- Provide circuit breakers and other safety measures to provide protection against shorts in external wiring.
- Do not apply voltages exceeding the rated input voltage to Input Units. The Input Units may be destroyed.
- Do not apply voltages exceeding the maximum switching capacity to Output Units. The Output Units may be destroyed.
- Always disconnect the LG terminal when performing withstand voltage tests.
- Install all Units according to instructions in the operation manuals. Improper installation may cause faulty operation.
- Provide proper shielding when installing in the following locations:
  - Locations subject to static electricity or other sources of noise.
  - Locations subject to strong electromagnetic fields.
  - Locations subject to possible exposure to radiation.
  - Locations near power supply lines.
- Be sure to tighten Backplane screws, terminal screws, and cable connector screws securely.
- Do not attempt to take any Units apart, to repair any Units, or to modify any Units in any way.

⚠ **Caution** The following precautions are necessary to ensure the general safety of the system. Always heed these precautions.

- Provide double safety mechanisms to handle incorrect signals that can be generated by broken signal lines or momentary power interruptions.

- Provide external interlock circuits, limit circuits, and other safety circuits in addition to any provided within the PLC to ensure safety.

6 EC Directives

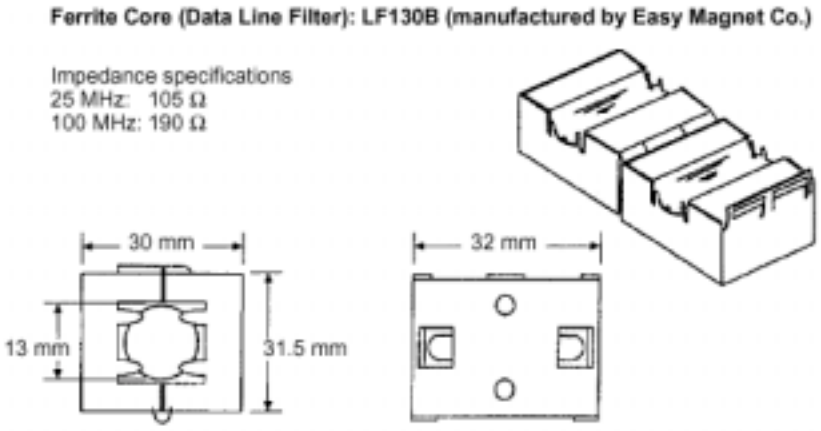
C200HW-CORT21-V1 Units that meet EC directives must be installed as follows:

1,2,3…

1. C200HW-CORT21-V1 Units are defined for installation inside control panels. All C200HW-CORT21-V1 Units must be installed within control panels.
2. Use reinforced insulation or double insulation for the DC power supplies used for the communications power supply, internal circuit power supply, and the I/O power supplies.
3. C200HW-CORT21-V1 Units that meet EC directives also meet the common emission standard (EN50081-2). When C200HW-CORT21-V1 products are built into equipment, however, the measure necessary to ensure that the standard is met will vary with the overall configuration of the control panel, the other devices connected tot the control panel, and other conditions. The user must therefore confirm that EC directives are met for the overall machine or device.

1,2,3… The following examples show means of reducing noise.

1. Noise from the communications cable can be reduced by installing a ferrite core on the communications cable within 10 cm of the C200HW-CORT21-V1 Unit.



2. Wire the control panel with as thick and short electric lines as possible and ground to 100 Ω or less.
3. Keep the communications cables as short as possible and ground to 100 Ω or less.

1 About CANopen

This section gives a brief description of CANopen.

1-1 Introduction

Based on CAN

CANopen is a networking system based on the serial bus CAN (Controller Area Network).

CANopen was developed within the scope of an 'ESPRIT' EU research programs led by Robert Bosch GmbH (initial developer of CAN) at the end of 1993. It encompasses a subset of the higher layer protocol CAL (CAN Application Layer), developed by Philips Medical Systems, and some extra functionality to meet the application area.

Application areas

CANopen was originally designed for motion-oriented industrial control systems, such as handling systems. But CANopen networks are also used in other application fields, e.g. public transportation, off-road vehicles, medical equipment, maritime electronics, and building automation.

CANopen makes full use of the powerful features of CAN. It supports the multi-master functionality, that enables (cyclic or event-driven) communication between any two or more nodes, and it offers a lot of other configuration flexibility. This flexibility makes this protocol a bit less plug-and-play than other common protocols, but it gives the system integrator more power in fine tuning the network.

CANopen Master vs. Slave

In contrast with other industrial fieldbus systems, there is no clearly defined separation between the functionality of CANopen master- and slave devices. This means that in practise there will be many different types of CANopen Master devices and slave devices, each with their own specific level of functionality.

CANopen specifications

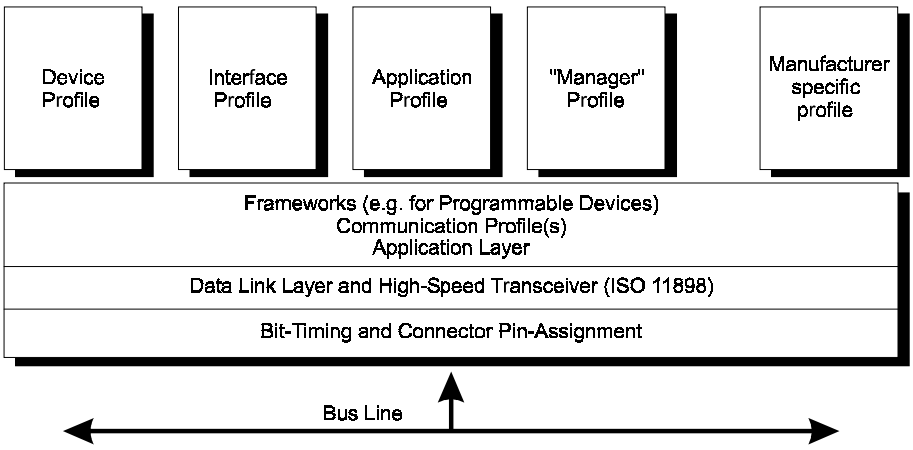
This chapter only gives a brief description of CANopen; please refer to the CANopen specifications, issued by CAN in Automation, for more detailed information.

1-2 Device description

A CANopen device is best described by its OSI reference model and its device model.

1-2-1 CANopen Reference Model

The following figure depicts the reference model of a CANopen device.



OSI model

The model is according the ISO Open Systems Interconnection (OSI) reference model. Of this seven-layer model, only three are applicable for a CANopen device (the physical layer, the data link layer and the application layer).

ISO 11898

The data link layer and a part of the physical layer are described in ISO 11898, and are usually implemented in silicon (CAN controller).

CANopen provides a standardised application layer and communication profile. The optional framework for programmable devices specifies additional communication functionality. In addition, CANopen specifies bit-timing and recommends pin-assignments for several types of connector.

Profiles

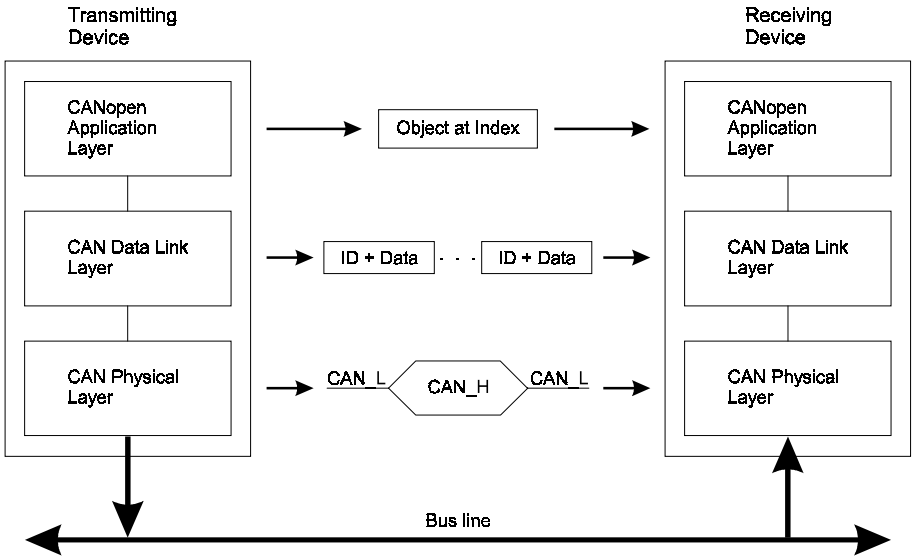
The standardised device profiles, interface profiles, and application profiles describe the default behaviour and the optional functionality of devices, interfaces, and applications.

The interaction between devices at the three ISO layers is depicted below.

At the application layer, the interaction involves the accessing of objects via an index numbering system. Refer to section 1-2-2-1 for more details about objects and indexing.

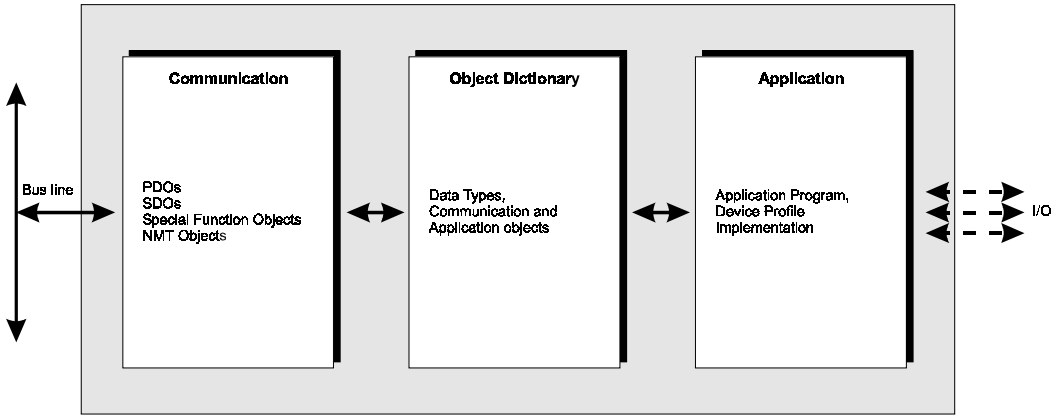
At the data link layer, these object accesses are translated to CAN message frames with certain identifiers and data blocks. The maximum number of data bytes per CAN frame is eight, therefore accesses to larger-sized objects in the device are fragmented over more than one frame.

At the CAN physical layer, the CAN frames are translated to bus signal levels and vice versa.



1-2-2 Device model

The following figure shows the general model of a CANopen device.



A CANopen device can be divided in three parts:

- Communication interface
- Object dictionary
- Application interface

These three parts are discussed in the following sections, starting with the core of a device, the object dictionary.

1-2-2-1 Object dictionary

The core of every CANopen device is its object dictionary. The structure of this dictionary is as follows:

Index (hex)	Objects
0000	Reserved for future use
0001 ~ 009F	Data type definitions
00A0 ~ 0FFF	Reserved for future use
1000 ~ 1FFF	Communication Profile Area
2000 ~ 5FFF	Manufacturer Specific Profile Area
6000 ~ 9FFF	Standardised Device Profile Area
A000 ~ FFFF	Reserved for future use

Object oriented

A CANopen device is organised in an object-oriented manner. All objects are accessed via an index number, which is a 16-bit value ranging from 0000h to FFFFh.

Index

'Simple variable' objects are accessed directly by their index.

Sub-index

Elements of 'complex variable' objects, like records or arrays, are accessed by the index number and an 8-bit sub-index. This principle is shown in the following example.

Example

The object at index 1000h is a simple variable and can be accessed directly via index 1000h.

Complex variables

The object at index 1800h is a complex variable. The first element (sub-index 00h) of a complex variable always indicates the number of elements that follow. In this case there are two elements, COB-ID and Transmission Type.

The elements of the complex variable 'Transmit PDO parameter' are accessed via index 1800h and the sub-indices 00h to 02h.

Index	Sub-index	Object
1000h	-	Device Type
1800h	-	Transmit PDO parameter
1800h	00h	Number of entries
1800h	01h	COB-ID
1800h	02h	Transmission Type

Attribute

Each variable in the object dictionary has an attribute indicating whether it is

- read-only (RO),
- write-only (WO),
- read-write (RW),
- read-write on process input (RWR), or
- read-write on process output (RWW).

A CANopen device is not required to contain all possible entries of an object dictionary.

Communication Profile

The Communication Profile Area is mandatory for all devices, but not all of the objects in this area need to be implemented (depends on the type of device).

Device Profile

For some types of device (e.g. I/O devices or drives) a so called device profile has been defined. A device profile defines the application objects of such a device. If implemented according to this profile, similar devices can easily be exchanged because they have the same interface. The objects of a device profile are located at the indices 6000h ~ 9FFFh.

Manufacturer specific Profile

A dedicated area in the object dictionary is reserved for manufacturer specific objects. These are used when no applicable device profile exists for the device or if the manufacturer requires extensions to an existing device profile. This area also enables future extensions of the CANopen protocol.

EDS file

The electronic data sheet (EDS-file) of a device lists the objects that are implemented in the device. The contents of the EDS-file as supplied with C200HW-CORT21-V1 can be found in appendix A.

CANopen specifications

Refer to the Communication Profile descriptions DS30x and/or the Device Profile descriptions DS40x for more details about the objects in the object dictionary.

CiA

These profiles can be purchased from CAN in Automation (CiA), the international trade association promoting the use of CAN.

CiA can be contacted via the web-site www.can-cia.de

1-2-2-2 Application Interface

The application interface links the I/O data to the application data objects located in the Device Profile Area (6000h ~ 9FFFh) or Manufacturer Specific Profile Area (2000h ~ 5FFFh) of the Object Dictionary.

I/O data can refer to physical I/O connections, but also to memory locations, e.g. PLC memory area(s).

If the device adheres to a CANopen device profile, the application interface can be standardised, but it can also be manufacturer specific.

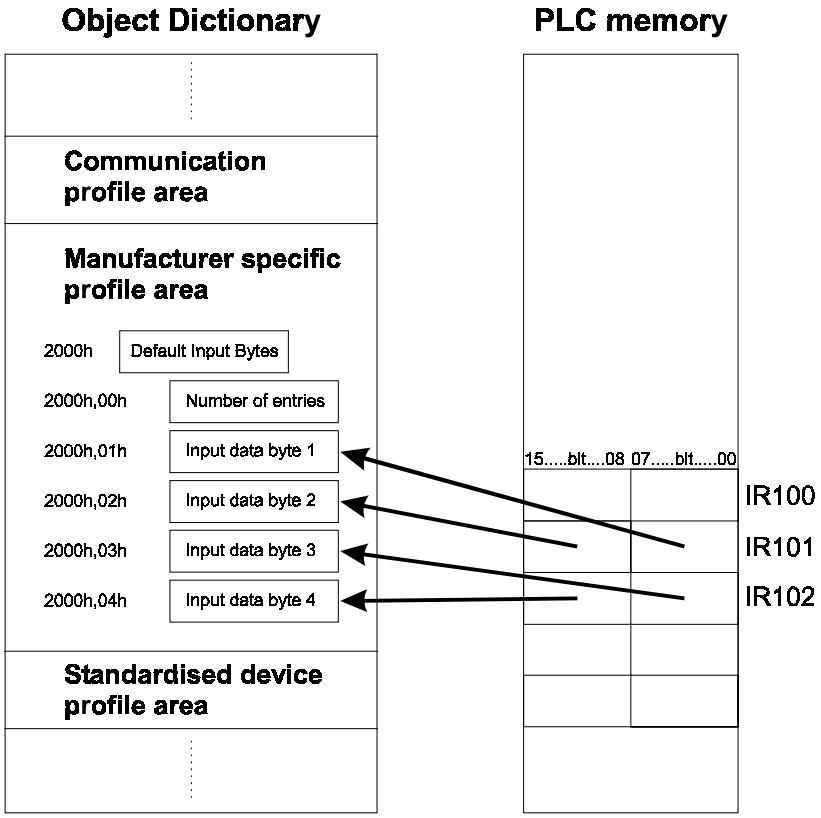
Example

The following figure gives an example which pertains to the C200HW-CORT21-V1. This Unit uses the Manufacturer Specific Profile Area in the object dictionary.

By default, the unit has four bytes input data for the CANopen network, located at index 2000h, sub-indices 01h to 04h. The IR words 101 and 102 are mapped to these objects.

In each PLC I/O refresh, the input bytes are updated with the contents of IR 101 and IR 102. In which way these bytes are transmitted over the CAN-bus is defined by the communication interface, which is described in the next sub-section.

Note that the term 'input' is used with respect to the CANopen network.



1-2-2-3 Communication Interface

The communication interface describes four different types of communication objects:

- Process Data Objects (PDOs)
- Service Data Objects (SDOs)
- Special Function Objects
- Network Management Objects (NMT)

PDOs and SDOs are used to access the entries in the object dictionary; PDOs are always directly linked to application objects, whereas SDOs can access any entry in the object dictionary.

Special Function Objects are related to the synchronisation and emergency messaging of the device.

Network Management Objects are used to configure the device, control the state of the device, and to supervise the device.

Refer to section 1-3 for more details about the communication objects.

1-3 Communication

Three communication models are applicable to CANopen:

Producer / Consumer

- Producer / Consumer model

  Each station on the network can receive the messages of any transmitting station. After receiving the message, it is the task of each individual node to decide if the message has to be accepted or not. This model describes the CAN broadcast communication capability.

Client / Server

- Client / Server model

  This model describes peer-to-peer communication. The client transmits a message; the message will be acknowledged by the server. This model is used to transfer data which exceeds the maximum data length of one message. The data to be transmitted is first segmented by the sender, and segments are transmitted one by one. The acknowledgement by the receiver can be for each segment or group of segments, or for the total message.

Master / Slave

- Master / Slave model

  The communication is always initiated by the master to which the slave has to respond. The slave is not allowed to autonomously initiate communication.

The application of these models is explained in more detail in the following sections, which describe the CANopen communication and related objects.

1-3-1 Process Data Objects (PDOs)

Purpose of PDO

A PDO defines a message frame transmitted or received via the CANopen network, with the purpose of exchanging real-time data (process data). The PDOs are directly linked to data objects in the object dictionary, and therefore no overhead is required to indicate the object. The PDO communication can be described by the producer/consumer model. Process data can be transmitted (broadcast) from one device (producer) to an arbitrary number of other devices (consumers).

Non-confirmed

PDOs are transmitted in a non-confirmed mode.

Max. PDOs

Each PDO is defined by two records:

- Communication parameters
- Mapping parameters

Both records are part of the Communication Profile Area and can both be adjusted via SDO accesses (see section 1-3-2).

The maximum number of PDOs in a CANopen network is limited to 512 Transmit PDOs (TPDOs) and 512 Receive PDOs (RPDOs). The predefined connection set (the default configuration) defines 4 TPDOs and 4 RPDOs per device. Refer to section 1-3-5 for more details about the predefined connection set.

1-3-1-1 Communication parameters

COB-ID

The communication parameters define the *COB-ID* (Communication Object IDentifier) and the *transmission type* (synchronous / asynchronous) of a data object. Refer to section 1-3-5 for more details about the communication object identifier.

Transmission modes

CANopen defines two transmission modes:

- Synchronous transmission
- Asynchronous transmission

Synchronous PDOs

Synchronous PDOs are transmitted within the synchronous window after the Sync Object (see section 1-3-3-1). The priority of synchronous PDOs should be higher than the priority of asynchronous PDOs.

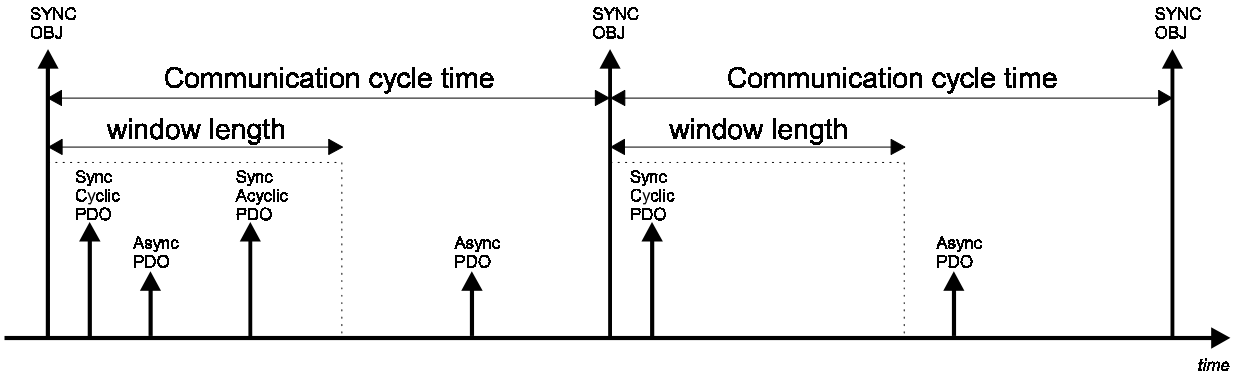
The transmission of synchronous PDOs can be subdivided into cyclic and acyclic transmission mode. For synchronous cyclic PDOs, the number of the transmission type (1 to 240) indicates the number of Sync Objects between two PDO transmissions. Acyclically transmitted synchronous PDOs are triggered by an application specific event. The message will be transmitted synchronously with the Sync Object, but not periodically.

Asynchronous PDOs

Asynchronous PDOs, and SDOs, can be transmitted at any time with respect to their priority. So they can also be transmitted within the synchronous window.

Communication cycle time
Window length

The communication cycle time (time between two Sync Objects) and the window length are both objects in the Communication Profile Area and are freely configurable.



1-3-1-2 Mapping parameters

PDO mapping

The mapping parameters define which application objects in the Manufacturer Specific or Standardised Device Profile Area are allocated to the PDO.

Granularity

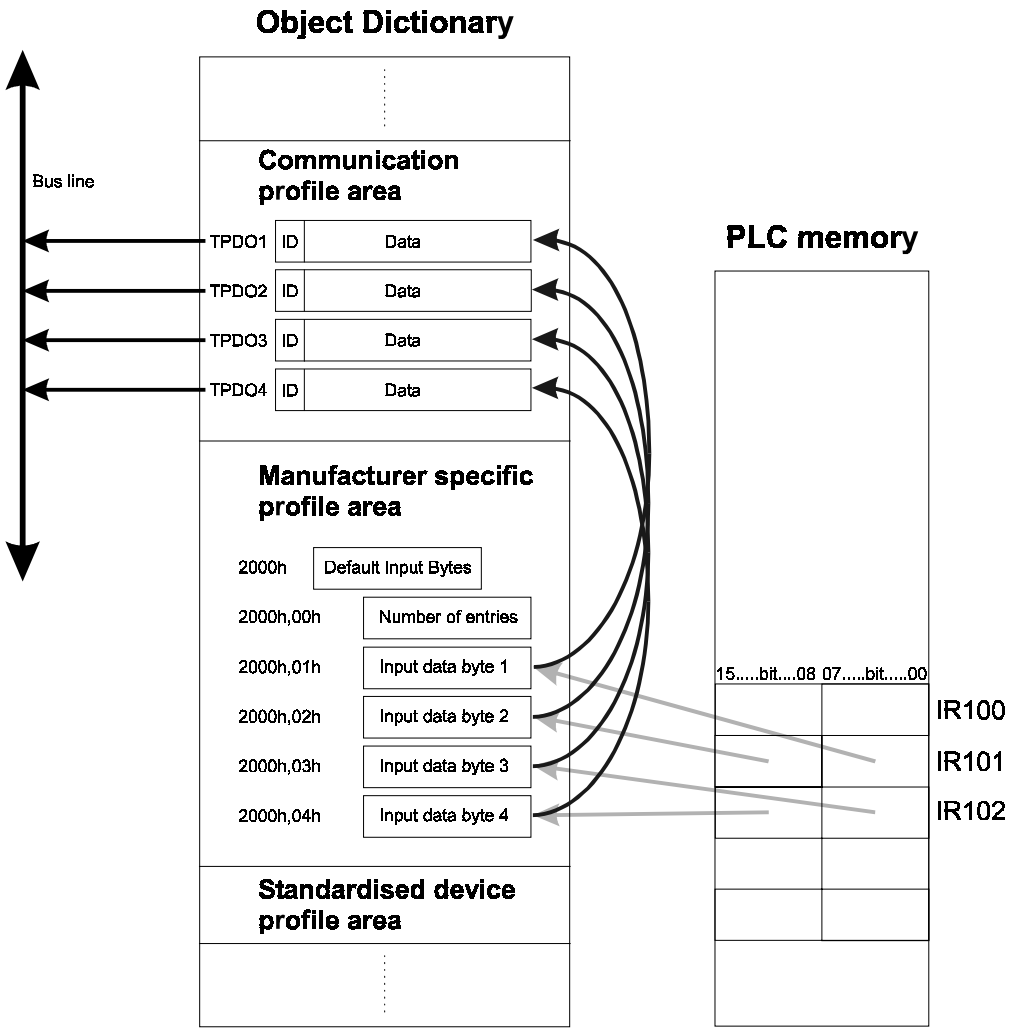
Each PDO can have up to 8 bytes of data allocated to it, and the minimum granularity of an application object is 1 bit. Thus a maximum of 64 application objects can be allocated to one PDO.

Example

The following figure shows an example of how process data is transmitted over the CANopen network. To keep the complexity of the figure to a minimum, the PDO definition in this figure is symbolised and not indicated by object entries that define the communication parameters and mapping parameters.

The data located at index 2000h, copied from the PLC memory, is mapped to TPDO1 ~ TPDO4; one byte per PDO. In the PDO, the data is preceded by a message identifier (ID) that characterises the message on the bus. The PDOs are transmitted over the CANopen network, and every node in the CANopen network will receive these messages. Depending on each PDO identifier, a node may decide to process the data, or ignore it.

The frequency and timing of the transmissions depend on the TPDO's transmission parameters.



1-3-2 Service Data Objects (SDOs)

Purpose of SDO

An SDO defines a message frame, transmitted or received via the CANopen network. Its main purpose is to configure a CANopen device, or to check its status. An SDO can address any object within a CANopen device. The SDO communication is described by the client / server model.

Each SDO is defined by one record in the Communication Profile Area:

- SDO Parameters

The SDO parameters contain the transmit- and receive communication identifier (COB-ID) and the node ID of the node to which it is communicating.

Max. SDOs

The maximum number of SDO channels in a CANopen network is 256. It is mandatory for every CANopen node to have at least one server SDO to enable initial configuration of the node.

The data field (max. 8 bytes) in a message frame of the first SDO always has the following format:

Byte 0	*Byte 1-3: Multiplexor*		*Byte 4-7: Data*
Command Specifier	16-bit Index	8-bit Subindex	1-4 byte parameter data

The command specifier in an SDO indicates whether it concerns

- a download or an upload,
- a request or a response,
- a segmented-, block- or expedited transfer,

and it contains

- the number of data bytes in the current frame,
- the end indicator and
- an alternating toggle bit for each subsequent segment.

The 3 byte 'multiplexor' is a pointer to the object in the object dictionary.

The remaining bytes carry the parameter data. If the data fits in these remaining four bytes, it is called an expedited transfer. A single SDO access to the respective object is sufficient.

If more data needs to be transferred, the data is fragmented into several segments. The first SDO transmission initiates the down- or upload, and the following transmissions contain the remaining data. In case of errors, either the client or the server can abort the transfer. The transfer is completed with a special SDO, to indicate the end of the transmission.

The SDOs following the first initiating SDO have the format shown in the figure below. Each message contains 7 bytes of data.

Byte 0	*Byte 1-7: Data*
Command Specifier	Segment data

Max. Data

The maximum amount of data that can be transferred per SDO depends on whether it is an expedited-, block- or segmented transfer.

- Expedited transfers are limited to 4 bytes
- Block transfers are limited to 127 segments
- Segmented transfers are unlimited.

Block transfer is faster than segmented transfer because the transfer is only acknowledged after the total down- or upload. In the case of segmented transfer, every client SDO segment must be acknowledged.

1-3-3 Special Function Objects

The following objects are included in the category Special Function Objects:

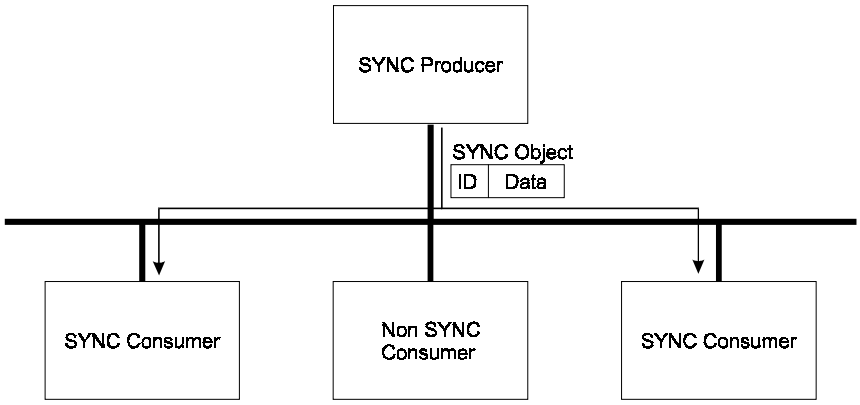
- SYNC Object
- Time Stamp Object
- EMCY Object

These objects follow the producer / consumer communication model.

1-3-3-1 SYNC Object

Process synchronisation

A SYNC Object enables the synchronisation of processes on the CANopen network. A SYNC object is transmitted by a SYNC Producer and can be processed by one or more SYNC Consumers (see figure below).



CANopen devices may be SYNC Producer, SYNC Consumer, neither, or both. Specific behaviour as producer or consumer is configurable.

SYNC Objects are also configurable, and are characterised by:

- message identifier
- communication cycle time
- window length (see also section 1-3-1-1).

A high priority message identifier is assigned to a SYNC Object to guarantee an accurate communication cycle time. By default, a SYNC Object does not contain data.

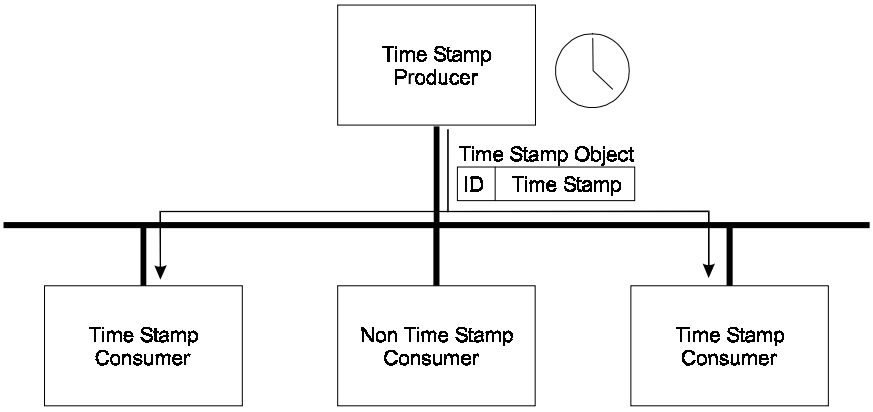
1-3-3-2 Time Stamp Object

CANopen provides a mechanism to assure synchronisation of local real-time clocks in CANopen nodes.

The Time Stamp Object contains:

- the number of days since January 1, 1984.
- the time in milliseconds after midnight

This information is encoded in 6 bytes.



CANopen devices may be Time Stamp Producer, Time Stamp Consumer, neither, or both. Specific behaviour as producer or consumer is configurable.

A high priority message identifier is assigned to a Time Stamp Object to assure immediate transmission.

Note: C200HW-CORT21-V1 does not support this function.

1-3-3-3 EMCY Object

Emergency messages are triggered by the occurrence of device internal errors and are broadcast to all other devices with high priority. An emergency message is only sent once per 'error event'. As long as no new device errors occur, no further emergency message will be sent.

EMCY identifier

Each producer has its own emergency message identifier to distinguish itself from other nodes.

Emergency code

The emergency code in the data field of the message has the following format:

Byte 0-1	*Byte 2*	*Byte 3-7*
EEC	ER	MEF

EEC = Emergency Error Code
ER = Error Register
MEF = Manufacturer-specific Error Field

The EEC's are defined by the CANopen Communication Profile.

The value of the error register (index 1001h) is specified in the device profile. If no device profile is applicable, this can be manufacturer-specific.

1-3-4 Network Management Objects (NMT)

NMT Master

The CANopen network management is node-oriented and follows a master / slave model. It requires one device in the network which fulfils the function of NMT Master; the other nodes are its NMT Slaves.

Functionality groups

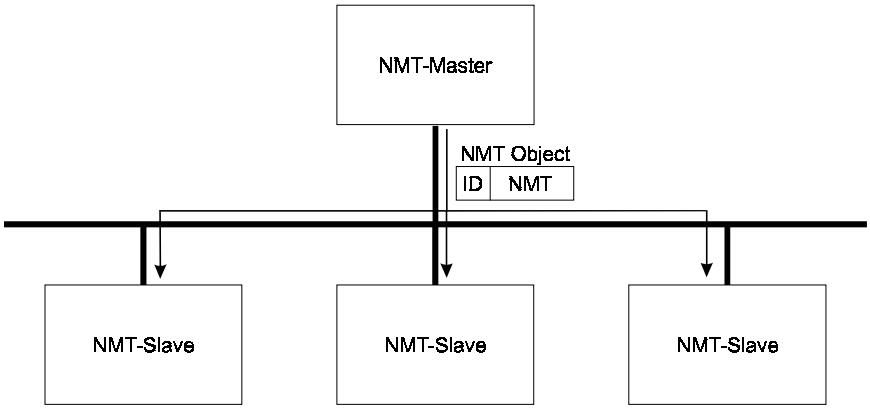
The network management provides the following functionality groups:

- Module control services

  Initialisation of NMT Slaves that should take part in the distributed application.
- Error control services

  Supervision of the nodes and the network communication status.
- Configuration control services

  Up- and downloading of configuration data from / to nodes in the network.

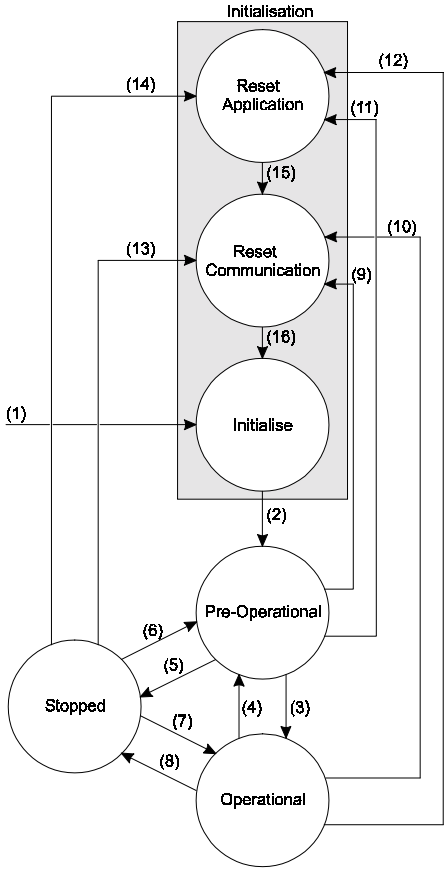


1-3-4-1 Module Control Services

CANopen State machine

Every CANopen NMT Slave functions according to the state diagram shown in the following figure. Note that this only depicts the CANopen functionality of a device. The state transitions are controlled by either

- a hardware reset,
- NMT control services, or
- device internal control services.



State transition	Description
(1)	Power-on or Hardware Reset
(2)	Finished initialisation
(3),(7)	NMT: Start_Remote_Node
(4),(6)	NMT: Enter_Pre-operational_state
(5),(8)	NMT: Stop_Remote_Node
(11),(12),(14)	NMT: Reset_Node
(9),(10),(13)	NMT: Reset_Communication
(15)	Application Reset performed
(16)	Communication Reset performed

Initialisation state

The initialisation state (the greyed box) is sub-divided into three sub-states:

- Reset Application

  The parameters of the Manufacturer Specific Area and/or Device Profile Area are set to their power-on values. After setting the values, the device autonomously enters the Reset Communication state.

- Reset Communication

  The parameters of the Communication Profile Area are set to their power-on values. After this, the device enters the Initialise state.

- Initialise

  The basic device initialisation of CANopen is executed. If the previous state was not 'Reset Communication', the objects are initialised with the stored values. If the device does not support storage of parameters, the objects are initialised with the default values according the Profile specifications.

Boot-up object

Before entering the Pre-Operational state, the standardised boot-up object is transmitted. This object indicates to the network that the respective node has completed its initialisation and will enter Pre-operational state.

Boot-up object

Before entering the Pre-Operational state, the standardised boot-up object is transmitted. This object indicates to the network that the respective node has completed its initialisation and will enter Pre-operational state.

Pre-Operational state

After power-on and internal initialisation, every device autonomously enters the Pre-Operational state. In this state the node may be configured and parameterised via SDO accesses (e.g. by using a configuration tool); no PDO communication is allowed.

Operational state

An NMT Master may switch a specific node, or all nodes simultaneously, to the Operational state. In this state, all communication objects are active. PDO transfers are possible, as well as object dictionary accesses via SDO messages.

Stopped state

By switching a device into the Stopped state, it is forced to stop all PDO and SDO communication.

1-3-4-2 Error control services

The CANopen protocol has two error control services:

- Node / life guarding
- Heartbeat

In any node, these two services are mutually exclusive. The service that is started first from the boot-up phase, will be active. In case the heartbeat service is disabled, it is possible to start Node / Life guarding.

Heartbeat

A Heartbeat Producer transmits the Heartbeat message cyclically with the frequency defined in the Heartbeat Producer time object. One or more Heartbeat Consumer may receive this message. The relationship between producer and consumer is configurable via object dictionary entries.

The Heartbeat Consumer monitors the reception of the Heartbeat within the Heartbeat consumer time. If the Heartbeat is not received within this time, a Heartbeat event is generated to notify the receiving node's application.

Node Guarding

With the node / life guarding mechanism, the NMT master polls the NMT Slaves at regular intervals. The slaves respond with their current node state. If a node state does not correspond with the expected state stored in the NMT master, or the slave does not respond at all within a configurable time frame, the respective node will be considered faulty. This process is known as Node Guarding.

Life Guarding

To check if the NMT Master is still alive, the slaves monitor internally if the master sends the Node Guarding message within the configured life-time. This is known as Life Guarding.

The preferred error control service is Heartbeat. This mechanism reduces the bus load and does not make use of remote transmission request message frames.

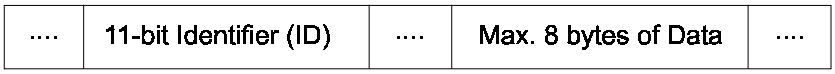
1-3-4-3 Configuration control services

Configuration control services are used for up- and downloading of configuration data from / to a module in the network. Refer to section 1-3-2 for details about how modules are accessed.

1-3-5 Predefined Connection Set

CAN message frame

The figure below shows the format of a CAN message frame. Only the most important parts of the frame are displayed: the 11-bit identifier and the data field. The other fields are irrelevant for the understanding of the communication protocol.



11-bit Identifier (ID)

The 11-bit Identifier indicates the type of message and determines the priority of the message. The lower the value of the identifier, the higher the priority.

Message priority

Message priority is only of importance when two nodes try to transmit at the same time. In that case, the device that transmits with the lowest identifier value gains the control over the bus.

Message filtering

When a message is transmitted over the network, each node will receive this message, but a node can be configured to only process specific messages. This message filtering is based on the value of the identifier.

In CANopen, a node will only process a message from another node when the identifier is known in advance, i.e. the node has been configured to filter the identifier. To achieve this, the ID of an RPDO of the receiving node must have the same value as the ID of a TPDO of the transmitter (see also section 1-3-1); the PDOs of the 2 devices are 'linked'.

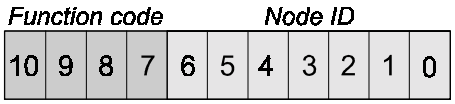
PDO linking

In CANopen, the Identifier of a communication object, like TPDO or RPDO, is called Communication Object ID or COB-ID.

To reduce the effort of so called 'PDO linking' for simple networks, CANopen has defined a mandatory default identifier allocation scheme: the Predefined Connection Set. These predefined identifiers are available in the Pre-Operational state (if no modifications have been stored).

Predefined connection set

The predefined identifier scheme of CANopen uses a 4-bit function part (Function code) and a 7-bit Node-ID part to form the 11-bit identifier.



The following table lists the identifier allocation scheme:

Function code	Node ID	Object
0	0	NMT
1	0	SYNC
2	0	Time Stamp
1	1 ~ 127	EMCY
3	1 ~ 127	PDO1 transmit (TPDO1)
4	1 ~ 127	PDO1 receive (RPDO1)
5	1 ~ 127	PDO2 transmit (TPDO2)
6	1 ~ 127	PDO2 receive (RPDO1)
7	1 ~ 127	PDO3 transmit (TPDO3)
8	1 ~ 127	PDO3 receive (RPDO1)
9	1 ~ 127	PDO4 transmit (TPDO4)
10	1 ~ 127	PDO4 receive (RPDO1)
11	1 ~ 127	SDO transmit (TSDO1)
12	1 ~ 127	SDO receive (RSDO1)
14	1 ~ 127	NMT Error control

The first three entries in the list are so called broadcast objects (Node ID = 0). All the others are peer-to-peer objects (Node ID ≠ 0).

Master-slave network

This predefined ID allocation scheme allows peer-to-peer communication between a single master device and up to 127 slave devices. If peer-to-peer communication between 2 slave devices is required, the identifiers need to be changed in such a way that the receiving identifier of one station equals the transmitting identifier of the other station. The identifiers of NMT, SDO transmit and receive, and NMT Error Control are fixed and can not be changed.

1-3-6 Device configuration

Pre-Operational state

A device can be configured during the Pre-Operational state using SDO accesses. It is possible, but not recommended to configure parameters during the Operational state.

The most common parameters to be are configured are:

- PDO communication parameters
- PDO mapping parameters
- (Producer / Consumer) Heartbeat time
- Guard time and life time factor
- SYNC Producer / Consumer
- Communication cycle period
- Synchronous window length
- (Server / Client) SDO parameters

These parameters are described in more detail in section 3-3-2.

Especially the configuration of PDO communication parameters is important if peer-to-peer communication is required between two or more NMT Slaves.

Besides the configuration of parameters in the object dictionary, the baud

rate and Node-ID must be set. Though this is possible via the network, most devices only allow mechanical setting on the unit (DIP- or rotary switches).

The C200HW-CORT21-V1 allows storage of its configuration parameters in non-volatile memory. At power-on, the Unit initialises with the stored configuration parameters (see section 3).

After the device is properly configured, an NMT master must set the device to Operational state to enable PDO communication.

1-3-7 Network length

The permissible length of a CANopen network depends on the network baud rate. The following table lists the relationship between the baud rate and the maximum network length.

Baud rate (kbit/s)	Network length (m)
10	5000 *
20	2500 *
50	1000 *
125	500
250	250
500	100
800	50
1000	25

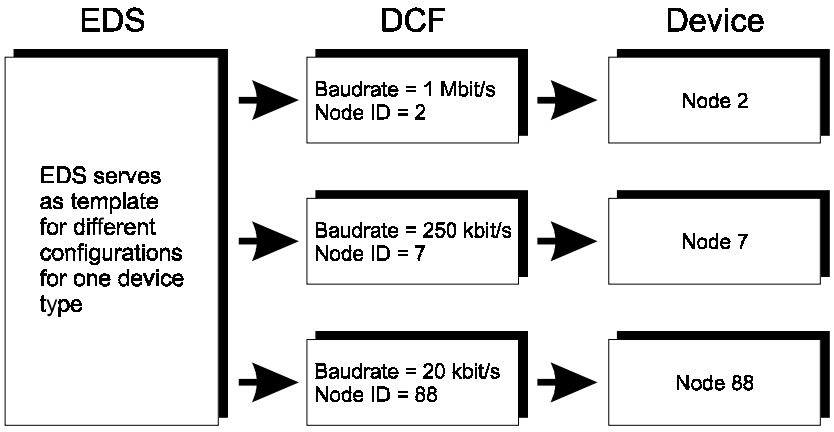
* For bus lengths greater than 1 km, a bridge or repeater device is recommended.

The bus should always be terminated with 120 Ω on both ends of the cable to ensure proper functioning of the network communication under all conditions.

1-4 Configuration files

CANopen makes use of two types of configuration files:

- EDS-file (Electronic Data Sheet)
- DCF-file (Device Configuration File)



ASCII file

The EDS-file describes the communication functionality and objects as defined in the CANopen specifications. It is a plain ASCII text file with the following structure:

- Information regarding the EDS file
  (filename and version, creation time and date, name of the creator, etc.)
- General device information
  (product name and revision, vendor name, supported baud rates, device type, etc.)
- Object dictionary with default values

The EDS-file is used by a network configurator tool and should be supplied by the vendor of the particular CANopen device. See appendix A for the contents of the EDS-file of C200HW-CORT21-V1.

The DCF is structured in nearly the same way as the EDS-file is. The DCF is the result of the EDS-file after configuration of the unit for a specific application. It describes not only the objects but also the configured values of the objects.

Note that the nodes in the figure above cannot communicate in the same network because they are not configured with the same baud rate!

1-5 CANopen conformance

Interoperability

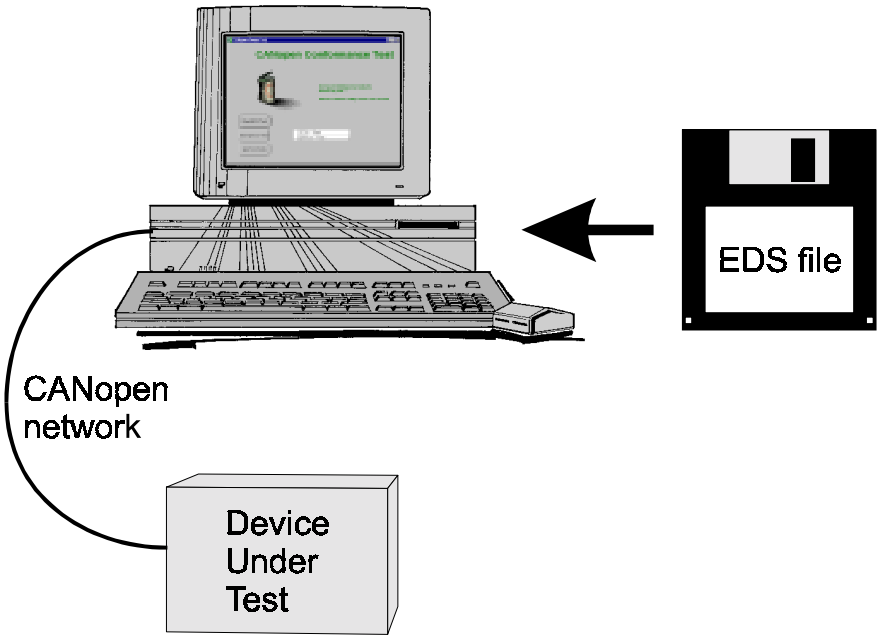
It is primarily the task of a device manufacturer to ensure interoperability (i.e. correct communication with CANopen devices of different manufacturers) by verifying the device's conformance to the CANopen specifications.

CiA certification test

CiA (the trade organisation CAN in Automation) has drawn up a conformance test specification and offers the service of an official test laboratory where CANopen devices can be certified.

The first step of the certification is the testing of the EDS-file. The content of the EDS-file must have the correct value ranges, must include all mandatory entries, references should point to existing entries and it must be consistent.

In the second step the physical CANopen device is tested. This part includes the test of the communication protocol, verification of the actual object dictionary against the EDS, and the checking of all network states and transitions.



2 Specifications and Performance

This section describes the overall specifications and the communication performance of the Unit

2-1 Overall Specifications

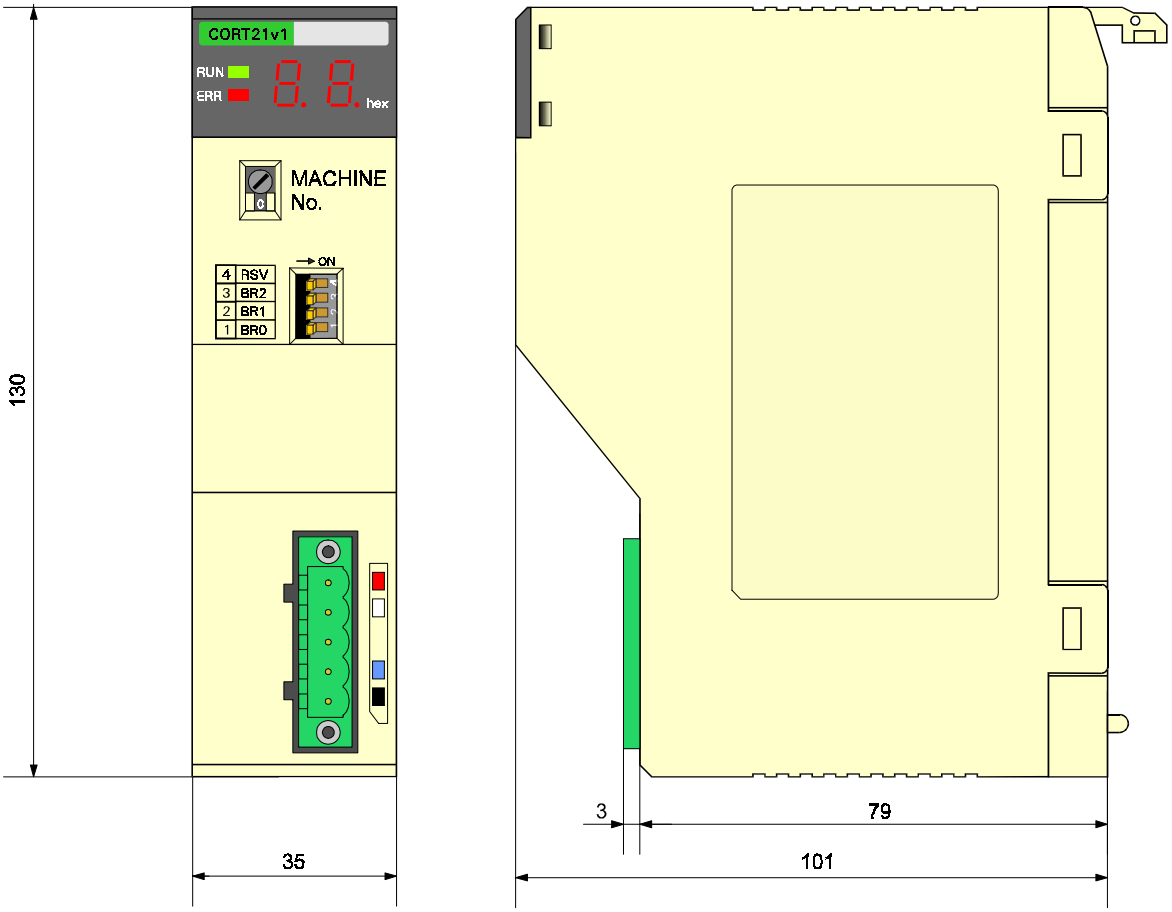
Model code		C200HW-CORT21-V1	
Installation	**Host PLC system**	C200HE, C200HG, C200HX, CS1-series	
	Maximum number of Units per PLC system.	C200HE C200HG-CPU3❑-E/CPU4❑-E C200HX-CPU3❑-E/CPU4❑-E	10
		All others	16
	Unit mounting position	CPU Rack or Expansion I/O Rack. Unit cannot be mounted to SYSMAC BUS Slave Racks.	
	Current consumption	Internal I/O power supply: 250 mA at 5V DC max. (from PLC) Network power supply: 45 mA max. (from communications connector)	
	Network power supply	24 V DC ± 10%	
	Weight	250 g	
Environment	**Storage temperature**	-20 to +75°C	
	Operating temperature	0 to +55°C	
	Operating humidity	10% to 90% (non-condensing)	
	Conformance to EMC- and environmental standards	EN50081-2 EN61131-2	
User interface	**Switch Settings**	Special I/O Unit number (0 ~ F) by rotary switch (front). Network baud rate (10 kbit/s ~ 1000 kbit/s) by 3 DIP-switches (front) CANopen node address (001 ~ 127) by 7 DIP-switches (rear)	
	Indicators	Unit status: RUN (green LED), ERR (red LED) Network status: 2 x 7-segment display	
PLC interface	**No. of IR words**	PLC → Unit: 3 words (1 word control data + 2 words CANopen transmit data) Unit → PLC: 5 words (3 words Unit status + 2 words CANopen receive data)	
	No. of DM settings	20 words	
	Amount of I/O data per Unit	Default [DM settings all 0000] : 2 words in + 2 words out With user defined DM settings: additionally up to 100 words in + 100 words out	
	Message communications	Via IOWR / IORD instructions	
Network interface	**CAN interface**	ISO 11898, High Speed CAN	
	Bus connector	5-pin male open style connector	
	Node address	1 to 127, remote setting not supported	
	Baud rate	10, 20, 50, 125, 250, 500, 800, 1000 kbit/s (note: Refer to appendix D for the configuration of any arbitrary baud rate via the DM settings.)	
	Profile	Communication Profile DS301 version 4.0 Manufacturer Specific Application Profile	

Network interface	**Supported functions**	Boot-up type	Minimum
		NMT	Slave
		Number of PDOs	Max. 64 transmit PDOs and 64 receive PDOs
		PDO transmission modes	Default: Asynchronous
			Configurable: Synchronous cyclic Synchronous acyclic
		PDO linking	Supported
		PDO mapping	Variable
		Application objects	Default configuration: Transmit objects: 4 x 8-bit Receive objects: 4 x 8-bit
			Additionally configurable: Transmit objects: 8-bit (max. 100) 16-bit (max.100) 32-bit (max. 50) 64-bit (max. 25) Receive objects: 8-bit (max. 100) 16-bit (max. 100) 32-bit (max. 50) 64-bit (max. 25) The total size of additionally configured transmit- and receive objects is limited to 100 words each.
		Number of SDOs	1 server
		Emergency Message	EMCY Producer
		Synchronisation	SYNC Consumer
		Error control services	Heartbeat (1 producer and 1 consumer) and Node guarding
		Storing of Parameters	Yes, can be stored in non-volatile memory
		User defined CAN messages	Transmission using IOWR PLC instruction Reception through configurable identifier filter
	EDS file	CORT21V1.eds	

2-2 Dimensions

Dimensions

The following figure shows the dimensions of the Unit. Refer to the PLC's Installation Guide for the dimensions of the Unit when it is mounted to the Backplane of the host PLC. (All dimensions are in mm.)



2-3 Performance

The overall performance of the C200HW-CORT21-V1 depends on the performance of the host PLC interface and on the performance of the CAN(open) interface. In the next two sub-sections, these two factors will be described first. After that, the overall performance is described.

2-3-1 PLC cycle time

PLC cycle time

The host PLC's cycle time mainly depends on the size of the PLC program and the I/O refresh times of all I/O Units connected to its I/O backplane(s).

The size of the PLC program is application specific. Apart from optimising the PLC program, the program execution time can only be decreased by using a faster CPU Unit.

I/O Refresh Time

The total I/O refresh time depends on the number and types of Units that are mounted on the backplane(s). Not all Units refresh the same amount of data.

The I/O refresh time of the C200HW-CORT21-V1 depends on the number and type of data objects that have been mapped, and whether user defined CAN messages need to be received. Refer to section 4-2-3 for details about mapping application objects on PLC memory locations and refer to section 5-8 for receiving user defined CAN messages.

Calculation

Using the default mapping mode (eight 8-bit objects allocated to the Special I/O unit's IR area), the I/O refresh time of the C200HW-CORT21-V1 is:

$t_{IORF,\ def} = 0.85$ ms

When additional application objects are mapped using the Unit settings, the I/O refresh time can be calculated with the following formula:

$$t_{IORF} = 0.85 + 0.45 \times n_a + 0.01 \times n_8 + 0.02 \times n_{16} + 0.04 \times n_{32} + 0.08 \times n_{64}\ [ms]$$

where n_a = number of data areas (0 ~ 6)

The C200HW-CORT21-V1 provides up to:

- 3 input data areas (8-bit, 16-bit, and 32/64-bit) and
- 3 output data areas (8-bit, 16-bit, and 32/64-bit).

An area is counted when the number of objects in that area is greater than 0.

n_8 = number of 8-bit input and output objects (0 ~ 200)

n_{16} = number of 16-bit input and output objects (0 ~ 200)

n_{32} = number of 32-bit input and output objects (0 ~ 100)

n_{64} = number of 64-bit input and output objects (0 ~ 50)

The total size of the input and output objects is limited to 100 words in and 100 words out.

When user defined CAN messages are received, the following time must be added to the I/O refresh time:

$t_{IORF,msg} = 0.45 + 0.12 \times n_m$

where n_m = number of received CAN messages (1 ~ 15)

The maximum IO refresh time is therefore:

$t_{IORF,max} = 0.85 + (0.45 * 6) + (0.02 * 200) + 0.45 + (0.12 * 15) = 9.8$ ms.

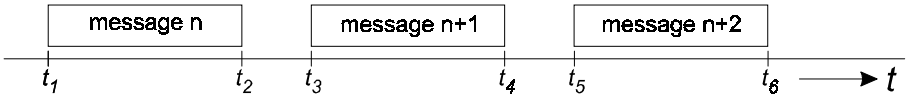
Note When the Unit is configured to receive user defined CAN messages, the I/O refresh time is not always constant over all PLC cycles. The refresh time depends on the number of CAN messages matching the identifier filter, which were received since the previous I/O refresh.

Refer to the PLC Operation Manual for more detailed calculations of the PLC cycle time.

2-3-2 CAN(open) Interface

This sub-section describes the performance of the physical layer of CAN and the performance of the CANopen interface of C200HW-CORT21-V1.

The following figure depicts the transmission of CAN messages on the bus.



$t_2 - t_1$ = *Message delay time*
$t_3 - t_2$ = *Message intermission*

Message delay time

Every message has a certain message delay time, which is mainly determined by the size of the data field in the message. CAN's bit-stuff mechanism (after every 5 consecutive equal value bits, a bit of the opposite polarity is added) can increase the message delay time by a maximum of 15%. The following formulas give the minimum and maximum message delay time.

$$\text{Message delay time} = \frac{\text{Number of bits in message}}{\text{Baud rate [bit/s]}} \quad [s]$$

$$\text{Message delay time}_{min.} = \frac{44 + 8 \times \text{number of data bytes}}{\text{Baud rate [bit/s]}} \quad [s]$$

$$\text{Message delay time}_{max.} = \frac{\text{trunc}(47.8 + 9.6 \times \text{number of data bytes})}{\text{Baud rate [bit/s]}} \quad [s]$$

(Maximum number of data bytes = 8)

2-3-3 Overall performance

This section describes the performance of the C200HW-CORT21-V1 in combination with the host PLC.

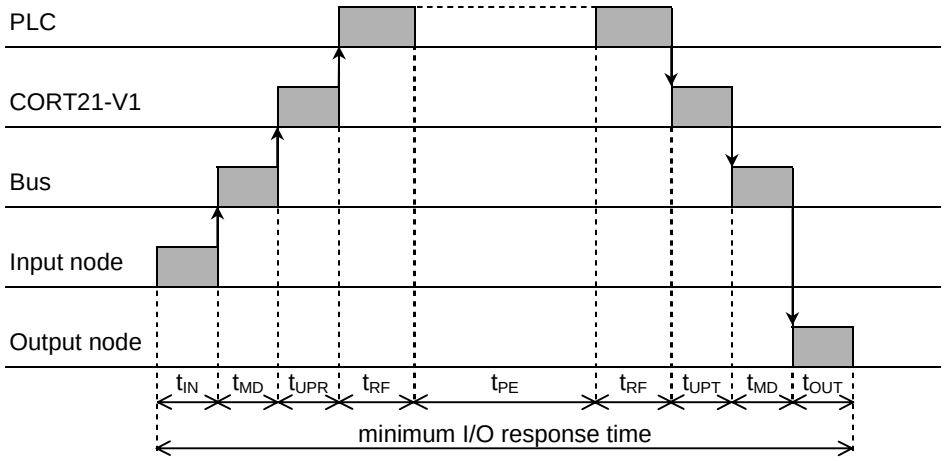
First, the I/O response time is described. This is the time between an input being set (or reset) on an Input Node and an output being set (or reset) on an Output Node, all under the condition that the Input Node and Output Node are linked to the C200HW-CORT21-V1.

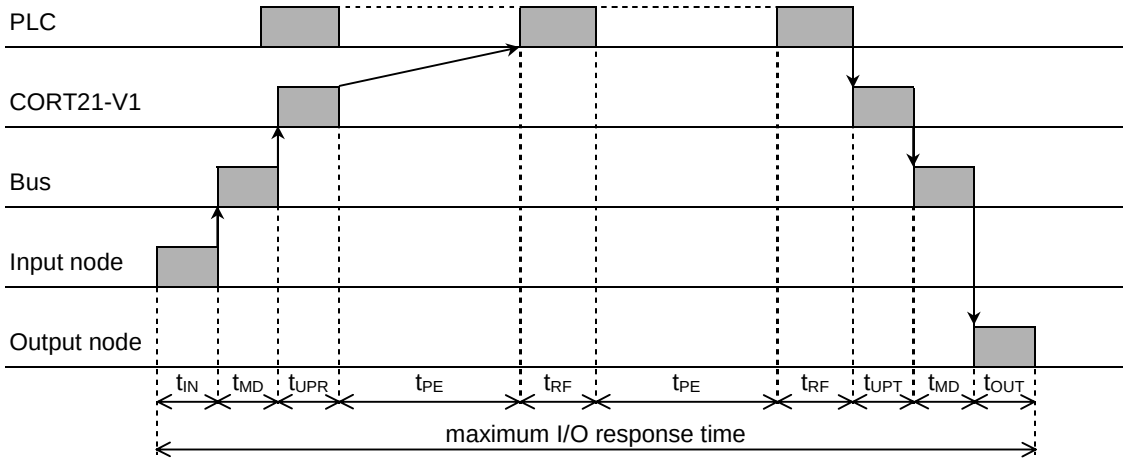
After that, the maximum transmission / reception rate of user defined CAN messages is described.

2-3-3-1 I/O response time

Minimum vs. maximum

The figures below show the minimum and maximum I/O response time. In the bottom figure, the processing of received messages is finished just after a PLC I/O refresh, and the received application data cannot be transferred to the PLC until the next I/O refresh. The other factors that can influence the I/O response time are for both cases identical and are explained below.





t_{IN} : Input Node ON (OFF) delay
t_{OUT} : Output Node ON (OFF) delay
t_{MD} : Message Delay time
t_{UPR} : Unit Processing time for Received messages
t_{UPT} : Unit Processing time for Transmitted messages
t_{PE} : Program Execution time
t_{RF} : I/O Refresh time

The Input / Output Node ON (OFF) delay

Refer to the manual of the Input / Output Node to determine its ON / OFF delay time.

The Message Delay time

The message delay time is described in section 2-3-2.

The Unit Processing time for Transmitted / Received messages

The Unit processing time for received messages depends on the number of PDOs linked to C200HW-CORT21-V1 and the message rate of these PDOs. If the message rate is higher than the unit is capable of processing (see section 2-3-2), the messages will queue up causing a delay in processing. When a receive queue overflow occurs, the Unit will notify the other nodes in the network by transmitting an emergency message.

The Unit processing time of messages to be transmitted depends on the number and type of input application objects mapped to PLC memory and to PDOs, and the rate at which these application objects change state. If many objects change state, the transmission of PDOs will queue up causing a delay in transmission. If the rate at which the objects change state requires an higher transmission rate than the Unit is capable of (see section 2-3-2), a transmit queue overflow occurs. The Unit will notify the other nodes on the network by transmitting an emergency message.

Transmit queue overflow errors can be solved by increasing the PLC cycle time or by reducing the rate at which the data objects change.

In case of high bus loads (over 50%), additional delays can be caused by high-priority (low identifier) messages which delay the transmission of lower priority messages.

The Unit processing time of I/O messages also depends on the transmission types of the receive and transmit PDOs. The fastest transfer of an application object to the PLC or onto the network can be achieved when the transmission type is set to asynchronous. The synchronous transmission type requires the reception of a SYNC object before the actual message is transmitted.

The I/O Refresh time

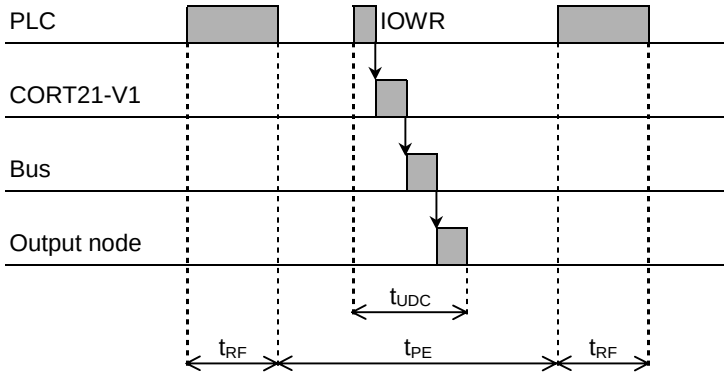
Refer to section 2-3-1 for the I/O refresh time of the C200HW-CORT21-V1.

The Program Execution time

The Program Execution time depends on the number and type of PLC instructions in the ladder program. Refer to the PLC Operation Manual for the execution times of the PLC instructions.

2-3-3-2 Transmission of user defined CAN messages

The following figure shows the sequence of transmitting a user defined CAN message.



t_{UDC} : The transmission time of a user defined CAN message

t_{RF} : The I/O Refresh time

t_{PE} : The Program Execution time

Refer to the PLC Operation manual for the execution time of an IOWR PLC instruction. After the content of the IOWR is transferred to the C200HW-CORT21-V1, the Unit immediately relays the contained message to the CAN transmission interface. The actual transmission delay of the message depends on the bus load and the message priority (identifier). Refer to section 2-3-2 for the message delay time on the bus. The time required for an output to be set or a value in the Output Node to be changed depends on the characteristics of the Output node.

The minimum time between the execution of the IOWR instruction in the PLC program and the message being transmitted on the bus is 0.5 ms.

2-3-3-3 Reception of user defined CAN messages

In order to receive user-defined CAN messages, an IOWR instruction must be executed to configure the Unit regarding :

- which messages to receive (filter identifiers by range and/or mask)
- which PLC memory locations to copy the receive messages to.

This only needs to be done once, and only needs to be repeated to change the filter and/or PLC memory allocation. Immediately after the execution of the IOWR instruction, the Unit is ready to receive messages.

The C200HW-CORT21-V1 can to store up to received 15 messages in its internal buffer. All stored messages are transferred to the PLC during the next I/O refresh. Therefore, the maximum delay time in the message reception is one PLC cycle.

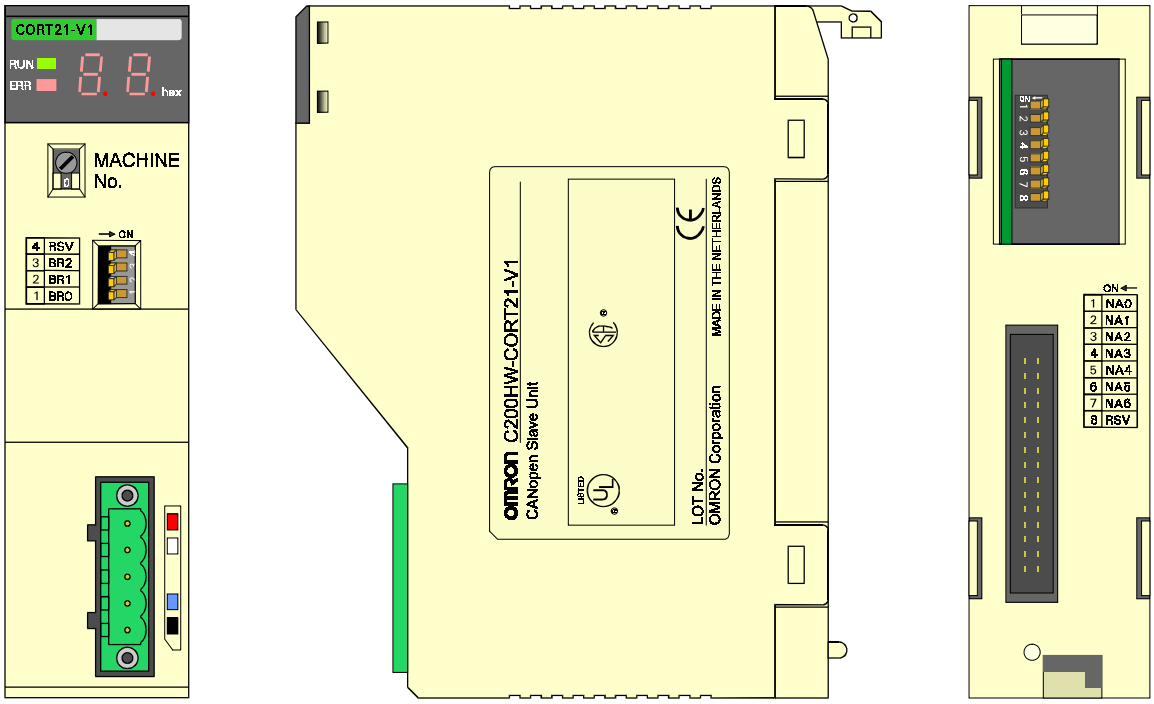
The maximum message rate that can be successfully received therefore depends strongly on the PLC cycle time. e.g. if the PLC cycle time is 20 ms, the maximum message rate is 15 messages/20 ms = 750 messages/second.

3 Installation

This section describes the installation of the C200HW-CORT21-V1

3-1 Physical layout of the Unit

The figure below shows the front, side, and rear view of the C200HW-CORT21-V1.



The front view shows the indicator LEDs, the 7-segment displays, the rotary switch, the baud rate DIP switches, and the 5-pin bus connector.

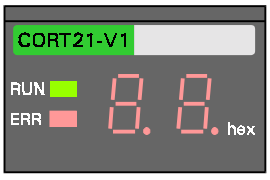
The rear view shows the Node Address DIP switches.

3-1-1 LEDs

Unit status

The C200HW-CORT21-V1 has 2 indicator LEDs. These LEDs indicate the status of the Unit in general.

Refer to section 6-5 for a detailed (functional) description of the LEDs.



3-1-2 7-segment display

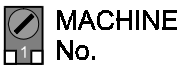
Network status

The Unit has two 7-segment displays. The displays are used to indicate the network status. In case there is a network error, an error code is shown, otherwise the node address is displayed.

Refer to section 6-5 for a detailed (functional) description of the 7-segment display.

3-1-3 Rotary Switch

The rotary switch is used to set the Special I/O Unit number (or so called “Machine No.”).



Memory allocation

The Machine number setting determines which words in the Internal Relay and Data Memory Areas are allocated to C200HW-CORT21-V1.

The valid Unit number setting range depends on the host PLC's CPU Unit, as shown in the following table.

CPU Unit models	Machine no. setting range	Setting method
C200HE, C200HG-CPU3❑-E/CPU4❑-E, C200HX-CPU3❑-E/CPU4❑-E	0 to 9	Single-digit hexadecimal
C200HG-CPU5❑-E/CPU6❑-E, C200HX-CPU5❑-E/CPU6❑-E, CS1-series	0 to F	

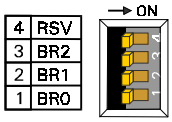
Any Machine No. in the setting range is allowed as long as it has not been set on another Special I/O Unit connected to the host PLC. If the same Machine number is used for the C200HW-CORT21-V1 and another Special I/O Unit, an "I/O Unit Over" error will occur in the PLC and it will not be possible to start up the network communication.

Note Always turn OFF the power to the PLC before changing the Machine number setting. The Unit only reads the Machine number setting during the initialisation after power-up, i.e. it is not processed at a software reset.

Use a small flat-blade screwdriver to turn the rotary switch; be careful not to damage the switch.

3-1-4 Baud rate switches

The DIP switches on the front of the Unit set the CAN communication baud rate.



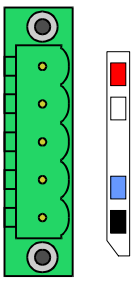
Only the first 3 switches are used (BR0, BR1, BR2); the fourth switch (RSV) is reserved for future use and should remain in the OFF position. The following table lists all possible DIP switch settings and their corresponding baud rate.

BR2	BR1	BR0	Baud rate (kbit/s)
OFF	OFF	OFF	10
OFF	OFF	ON	20
OFF	ON	OFF	50
OFF	ON	ON	125
ON	OFF	OFF	250
ON	OFF	ON	500
ON	ON	OFF	800
ON	ON	ON	1000

Note The C200HW-CORT21-V1 allows configuration of any arbitrary baud rate supported by CAN via the Unit settings (DM settings). Refer to appendix D for the definition of these Unit settings.

3-1-5 BUS Connector

The CAN bus connector is a 5-pin male open style connector. Its pins are numbered 1 to 5 starting from the top. The label next to the connector indicates the colour of the individual wire in the cable that has to be connected to the respective pin.



Pin No.	Signal	Description
1	CAN_V+	External power supply for the CAN interface (24V DC).
2	CAN_H	Data signal (dominant high)
3	CAN_SHLD	Optional CAN shield
4	CAN_L	Data signal (dominant low)
5	CAN_GND	Ground (0V of external power supply)

The CAN interface of C200HW-CORT21-V1 requires external power supply for the transceiver and opto-couplers. Therefore 24V DC ± 10% must be connected between pin 1 and 5. The current consumption Is less than 45 mA.

3-1-6 Node Address Switches

The switches at the rear of the Unit are used to set the CANopen Node Address. Only the first 7 switches are used, the last switch is reserved for future use and should remain in the OFF position. The following table lists the DIP switch settings and the corresponding Node Addresses.

ON ←	
1	NA0
2	NA1
3	NA2
4	NA3
5	NA4
6	NA5
7	NA6
8	RSV

NA6 (2^6)	NA5 (2^5)	NA4 (2^4)	NA3 (2^3)	NA2 (2^2)	NA1 (2^1)	NA0 (2^0)	Node Address
OFF	OFF	OFF	OFF	OFF	OFF	OFF	0 *
OFF	OFF	OFF	OFF	OFF	OFF	ON	1
OFF	OFF	OFF	OFF	OFF	ON	OFF	2
			⋮				⋮
ON	ON	ON	ON	ON	OFF	ON	125
ON	ON	ON	ON	ON	ON	OFF	126
ON	ON	ON	ON	ON	ON	ON	127

* Invalid Node Address. IR n+3.08 and the 7-segment display will indicate this error.

3-2 Mounting the C200HW-CORT21-V1

The C200HW-CORT21-V1 CANopen Slave Unit can be mounted to the CPU Rack or Expansion I/O Rack of any CS1, C200HX, -HG, or -HE PLC. Refer to the PLC's Operation Manuals for details on mounting Units.

Limitations

There are some limitations on mounting the Unit.

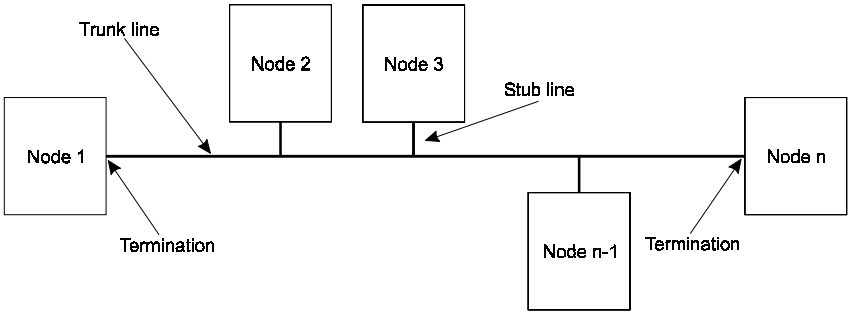
- The current consumption of all Units mounted on one backplane should not exceed the maximum output of the power supply. The C200HW-CORT21-V1 consumes up to 250 mA from the 5V supply. Verify the characteristics of all other Units on their respective Instruction Sheet or Installation Manual.
- The C200HW-CORT21-V1 is a Special I/O Unit. It can be mounted in any slot in the backplane of a CPU Rack or Expansion I/O Rack as long as its Machine number is not the same as the Machine number of another Special I/O Unit within the system.
- The maximum number of Special I/O Units that can be mounted depends on the PLC CPU type.
- For details on CS1-series PLC compatibility, see Appendix C.

CPU Unit models	Max. number of Special I/O Units
C200HE, C200HG-CPU3❑-E/CPU4❑-E, C200HX-CPU3❑-E/CPU4❑-E	10
C200HG-CPU5❑-E/CPU6❑-E, C200HX-CPU5❑-E/CPU6❑-E/CPU8❑-ZE CS1-series	16

3-3 Setting up a network

3-3-1 Network cabling

The wiring topology of a CAN network should be as close as possible to a single line structure in order to avoid signal reflections from cable branches. Stub-lines should be kept as short as possible and must be less than 30 cm at the highest baud rate (1 Mbit/s).



Termination

The bus must be terminated at both ends of the line. The cables, connectors, and termination resistors used in CANopen networks must meet the requirements defined in ISO 11898.

ISO 11898

The following tables list the electrical characteristics of a CAN cable and of the terminator according ISO 11898.

Cable Parameters	Value		
	Minimum	Nominal	Maximum
Impedance [Ω]	108	120	132
Length-related resistance [mΩ/m]		70	
Specific line delay [ns/m]		5	

Terminator Parameter	Value		
	Minimum	Nominal	Maximum
Resistance [Ω]	118	120	130

Cable length

The maximum permissible length of the cable depends on the transmission speed. Refer to section 1-3-7 for more details.

The market offers cables which, apart from the two data signal lines, also provide power supply lines and shielding (CAN_GND, CAN_SHLD, CAN_V+). This type of cable allows a central location of the network power supply and improves protection against electromagnetic disturbances.

For industrial CAN applications, OMRON recommends the use of DeviceNet cable, e.g. OMRON DCA1-5C10.

3-3-2 Network configuration

After making the physical connections of the network, the network needs to be configured.

Baud rate, node address First, all nodes must be configured to communicate on the same baud rate and all nodes must have a unique node address (or so called Node ID).

Object dictionary After that, the object dictionary of the nodes can be configured (if required) via SDO accesses. The most common parameters to configure are listed in section 1-3-6. These parameters are discussed in more detail in the following sub-sections. The last sub-section shows how the configuration can be stored to / restored from the non-volatile memory in the Unit.

Enable CANopen Note that the CANopen communication of C200HW-CORT21-V1 must first be enabled (IR n.00) before the Unit can be configured.

Start_Remote_Node After completing the desired configuration, the communication does not start until the NMT master gives the command 'Start_Remote_Node' to enter the Operational State. Refer to section 1-3-4-1 for more details about the (Pre-) Operational state and state transitions.

3-3-2-1 PDO Communication Parameters

The communication parameters of a PDO have at least two entries:

- COB-ID Communication Object Identifier
- Transmission Type Transmission/reception mode indicator

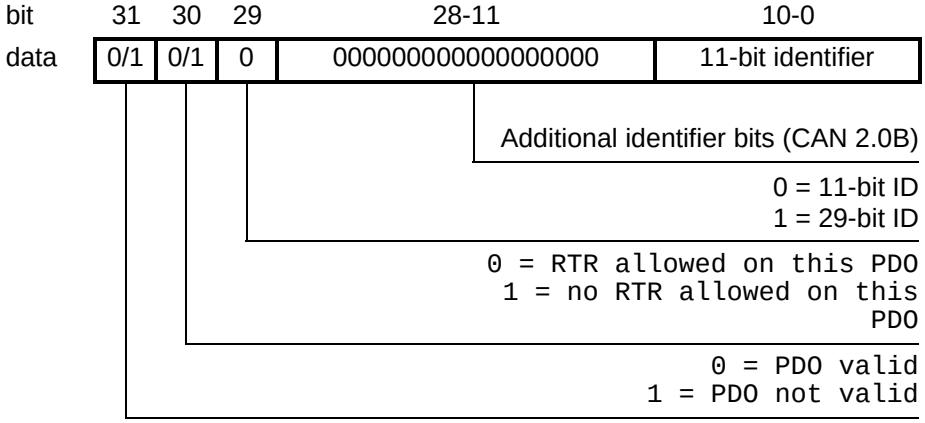
The following figure shows the communication parameters of the first transmit PDO in C200HW-CORT21-V1.

Index	Sub-index	Object	Default value
1800h	-	Transmit PDO1 parameter	-
1800h	00h	Number of entries	2
1800h	01h	COB-ID	180h + Node ID
1800h	02h	Transmission Type	254

COB-ID

To establish communication between CANopen nodes, their communication objects (PDOs) must be linked to each other. This is done by configuring the identifier part of the COB-IDs of the respective PDOs to the same value.

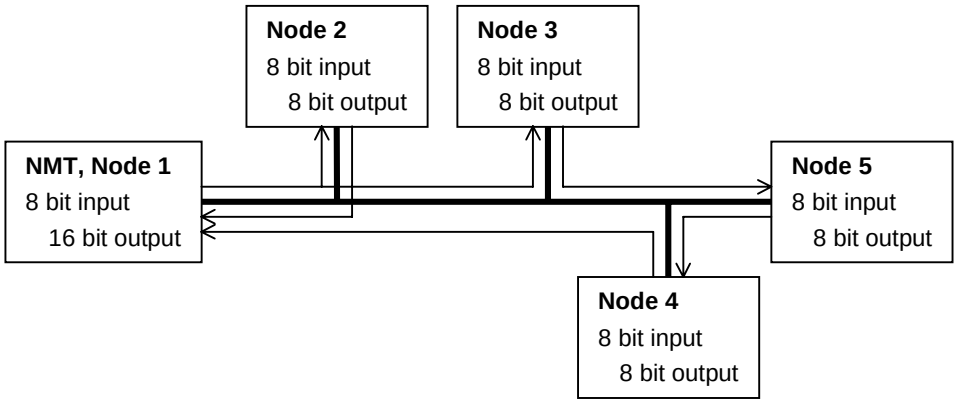
The COB-ID is a 32 bit object, which contains an 11-bit (or 29-bit) identifier:



Note: C200HW-CORT21-V1 supports only 11-bit identifiers, and no Remote Transmission Requests (RTR). Therefore all its valid PDOs have COB-ID's with the structure 40000xxxh. For PDO linking, only the identifier (xxx) is relevant.

Example

The linking of PDOs is best described by an example. The following figure shows a possible network configuration.



Nodes 2 to 5 are 8 digital input + 8 digital output modules. Node 1 is the Network Management Master, which is also able to transmit and receive data over the network. Note that the terms ‘input’ and ‘output’ are with respect to the network.

Node 1 transmits 8 bits of data to Node 2 and Node 3 (multicast).

Node 2 and Node 4 transmit their input data to Node 1.

Node 3 transmits its input data to Node 5.

Node 5 transmits its input data to Node 4.

To realise this data exchange, the configuration manager must configure the communication object identifiers (COB-IDs) of the PDOs. The configuration manager can be part of the node that also functions as the NMT master, but often a PC based configuration tool will be used. Refer to section 1-3-5 for more details about COB-IDs.

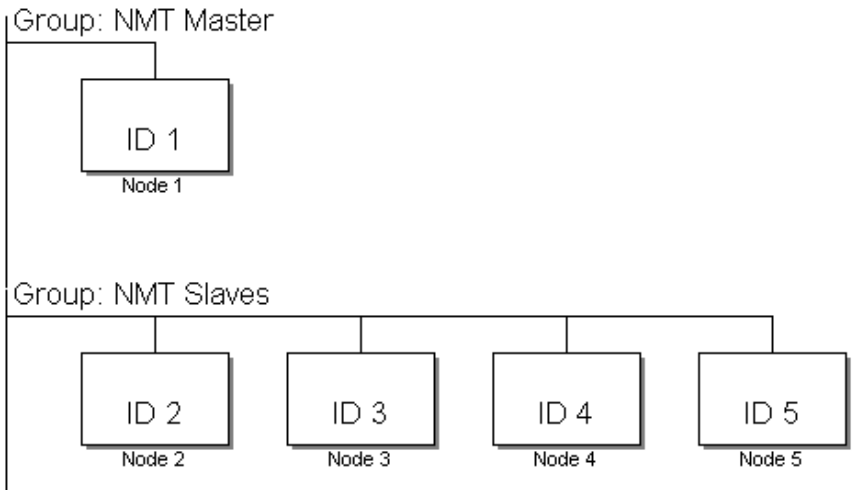
The configuration is built during the Pre-Operational state. For each data link, the COB-ID of a transmit PDO must be configured to the same value as the COB-ID of receive PDO. Note that the terms 'transmit' and 'receive' are from the node point of view.

In this case the configuration can be as follows:

COB-ID(TPDO1, Node 1) = COB-ID(RPDO1, Node 2) = COB-ID(RPDO1, Node 3)
COB-ID(RPDO1, Node 1) = COB-ID(TPDO1, Node 2)
COB-ID(RPDO2, Node 1) = COB-ID(TPDO1, Node 4)
COB-ID(TPDO1, Node 3) = COB-ID(RPDO1, Node 5)
COB-ID(TPDO1, Node 5) = COB-ID(RPDO1, Node 4)

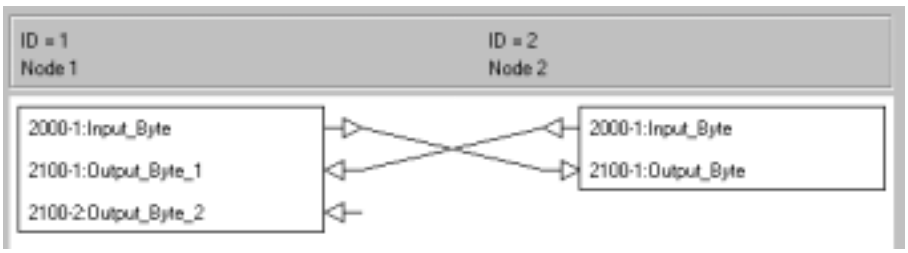
In this example it is assumed that the I/O data of Node 2 to 5 are allocated to their first transmit- and receive PDOs (TPDO1, RPDO1). The inputs of Node 1 are allocated to its TPDO1, but its 16 Outputs must be divided over two PDOs, RPDO1 and RPDO2, to allow linking each byte to a different Node.

The assignment of the COB-IDs can be done manually or automatically by a configuration tool, e.g. ProCANopen from Vector Informatik GmbH. This PC-based tool simplifies the 'PDO-linking' by using a graphical representation of the selected nodes and links. The configurator retrieves the necessary information for creating the links from the EDS-files of the respective nodes. Refer to section 1-4 for more details about EDS-files.



After making the on-screen representation of the network, the connections between the nodes can be established.

The graphical connection is the easiest way to link the application objects. The following figure shows how Node 1 is connected to Node 2.



The input byte of Node 1 is linked to the Output byte of Node 2 with two mouse clicks. The Input byte of Node 2 is connected to the first Output byte of Node 1 the same way. The configuration tool will automatically assign an available COB-ID to each connection made. Refer to the manual of a configuration tool for more details.

Transmission Type

The transmission type is an 8-bit variable which can take the values 0 to 255. The following table lists the values and their corresponding transmission mode.

Transmission Type	PDO transmission			
	Cyclic	Acyclic	Synchronous	Asynchronous
0		O	O	
1-240	O		O	
241-253	N.A.			
254				O
255	N.A.			

TPDO and RPDO

The transmission type is not only applicable to Transmit PDOs, but also to Receive PDOs.

- For Transmit PDOs, the transmission type determines when and how CAN messages are sent onto the network.
- For Receive PDOs, the transmission type determines when and how the data content of the RPDO is transferred to the mapped application objects.

Refer to section 1-3-1-1 for general information about the transmission modes.

In C200HW-CORT21-V1, asynchronous transmission is triggered by a change in data content.

- For TPDOs, a change in the value of an application object results in the transmission of the TPDO(s) to which it is mapped.
- For RPDOs, a change in data content results in an update of the mapped application object(s).

Synchronous acyclic transmission is triggered by a change in data content *and* the subsequent reception of a SYNC object.

Note that the data exchange between the application objects and the PLC memory always takes place. Its timing depends entirely on the cyclic or programmed I/O refreshes of the PLC.

3-3-2-2 PDO Mapping Parameters

The records containing mapping parameters of transmit PDOs and receive PDOs have the same structure. The figure below shows the mapping parameters of the first transmit PDO of C200HW-CORT21-V1.

Index	Sub-index	Object	Default value
1A00h	-	Transmit PDO1 mapping	-
1A00h	00h	Number of entries	1
1A00h	01h	mapped object	20000108h

By default only one object is mapped to the first transmit PDO. The maximum number of objects that can be mapped to a PDO is 64 if the granularity is 1 bit. The minimum granularity in C200HW-CORT21-V1 is 8-bits, so the maximum number of mapped objects is 8.

The variable at sub-index 01h (and higher) is a 32-bit variable that has the following structure:

Index (16 bit)	Sub-index (8 bit)	Object length (8 bit)

The first 24 bits form a pointer to the application object in the Device or Manufacturer Specific Profile Area, and the last 8 bits specify the length of the object in number of bits.

First set sub-index 0 to 0

These 32-bit variables can only be written to when sub-index 0 (number of entries) is set to 0. After configuring the mapping parameters, set sub-index 0 to the desired number of mappings. The Unit will then check if the mappings are correct and are not exceeding 8 bytes of data. If not correct, the Unit sends an abort message and keeps the previous configuration. This configuration procedure is normally taken care of by the configuration tool.

Refer to the EDS file of the C200HW-CORT21-V1 to see the default values for the mapping parameters.

3-3-2-3 Heartbeat Time

The structure of the Consumer Heartbeat Time is different from the Producer Heartbeat Time. A node can be consumer of more than one Heartbeat, but can not produce more than one Heartbeat. Therefore the Consumer Heartbeat Time is a complex variable (an array) and the Producer Heartbeat Time is a simple variable.

Consumer Heartbeat Time

The figure below shows the structure of the Consumer Heartbeat Time for C200HW-CORT21-V1. Note that the C200HW-CORT21-V1 can only consume one Heartbeat.

Index	Sub-index	Object	Default value
1016h	-	Consumer Heartbeat Time	-
1016h	00h	Number of entries	1
1016h	01h	Heartbeat Time	0

The variable at sub-index 01h is a 32 bit variable that has the following structure:

Reserved (8 bit)	Node ID (8 bit)	Heartbeat Time (16 bit)

The Node ID section of the variable indicates from which node produces the Heartbeat to be consumed. The Heartbeat Time specifies the expected Heartbeat cycle time and thus has to be higher than the corresponding producer heartbeat time. The Heartbeat time is a multiple of 1 ms. A value of 0 means that the Heartbeat monitoring for the indicated Node ID is disabled.

Heartbeat error

If the C200HW-CORT21-V1 does not receive the Heartbeat within the Heartbeat Time, the Unit indicates this error by IR n+3.11 and on the 7-segment display. It is up to the user to take any further action. When the Unit starts receiving heartbeats again, the error indication will disappear.

Producer Heartbeat Time

The figure below shows the structure of the Producer Heartbeat Time for C200HW-CORT21-V1.

Index	Sub-index	Object	Default value
1017h	-	Producer Heartbeat Time	0

The simple variable is a 16 bit variable that contains the Producer Heartbeat Time in milliseconds. The value 0 means that the Producer Heartbeat time is disabled.

When this variable is set unequal to zero at boot-up, the Heartbeat service will be active. Node / Life guarding can not be active if the Producer Heartbeat Time and/or Consumer Heartbeat Time are set unequal to 0.

3-3-2-4 Guard Time / Life Time Factor

Condition

Node guarding can only be started when Heartbeat is not activated. Heartbeat is inactive when both the producer heartbeat time and consumer heartbeat time are set to 0.

Remote transmission request

Node guarding will start after receiving the first remote transmission request (RTR) on COB-ID 700h + Node ID. After receiving the RTR, the C200HW-CORT21-V1 responds with the current CANopen state. The RTRs are transmitted at a fixed time interval called the guard time.

Guard Time

The figure below shows the structure of the guard time.

Index	Sub-index	Object	Default value
100Ch	-	Guard Time	0

The simple variable is a 16 bit variable that contains the Guard Time in milliseconds. The value 0 means that the Guard Time is not used.

Node Life Time

The Node Life Time is the Guard Time multiplied with the Life Time Factor. The figure below shows the structure of the Life Time Factor.

Index	Sub-index	Object	Default value
100Dh	-	Life Time Factor	0

The simple variable is an 8 bit variable; if not used, it is set to 0.

Life Guarding error

A Life Guarding Error occurs when the C200HW-CORT21-V1 did not receive an RTR within the configured Node Life Time. This error is indicated by IR n+3.11 and on the 7-segment display. It is up to the user to take appropriate action. The error indication disappears when RTRs are received again within the set time. Note that Life Guarding is inactive when either the Guard Time or Life Time Factor is set to 0.

Node Guarding error

A Node Guarding Error occurs when the C200HW-CORT21-V1 does not reply within the NMT Master's configured Node Life Time or responds with an unexpected CANopen state. It is up to the NMT master to take appropriate action.

3-3-2-5 SYNC Producer / Consumer

The C200HW-CORT21-V1 can only consume SYNC Objects and cannot produce them.

Index	Sub-index	Object	Default value
1005h	-	COB-ID SYNC message	00000080h

The COB-ID SYNC message is a 32-bit variable. The upper 21 bits in the variable should be set to 0 in C200HW-CORT21-V1 (these are only relevant for SYNC producers and devices with 29-bit identifiers). The lower 11-bits specify the communication identifier of the SYNC Object.

3-3-2-6 Communication Cycle Period

The Communication Cycle Period is only relevant on devices that generate the SYNC Object. It specifies the period in which the SYNC Objects need to be transmitted.

Index	Sub-index	Object	Default value
1006h	-	Communication Cycle Period	0

The Communication Cycle Period object is a 32-bit variable. It indicates the period in microseconds. When set to zero, the object is not used.

Since C200HW-CORT21-V1 is only a SYNC consumer, it does not use this object.

3-3-2-7 Synchronous Window Length

The Synchronous Window Length object specifies the time window with respect to the transmission of the SYNC object in which the synchronous messages are to be transmitted.

Index	Sub-index	Object	Default value
1007h	-	Synchronous Window Length	0

The Synchronous Window Length object is a 32-bit variable and the time unit is in microseconds. If the value is set 0, the object is not used.

In C200HW-CORT21-V1 the Synchronous Window Length cannot be specified. Synchronous TPDOs will be transmitted as quickly as possible after receiving the SYNC message, but since the number of synchronous TPDOs is user-configurable, the window size cannot be specified in advance.

3-3-2-8 SDO Parameters

The structure of client- or server SDO Parameters is shown below.

Index	Sub-index	Object	Default value
1200h	-	Server SDO parameter	-
1200h	00h	Number of entries	2
1200h	01h	COB-ID Client → Server (rx)	600h + Node ID
1200h	02h	COB-ID Server → Client (tx)	580h+Node ID

The variables of this complex data type are read-only and can therefore not be changed. The configurator can always access the device through these identifiers.

The C200HW-CORT21-V1 only supports the mandatory server SDO. This entry is used to configure the Unit.

The variables on sub-index 01h and 02h are both 32-bit variables. The upper 21 bits are used to indicate whether the SDO is invalid and whether a 29-bits identifier is used. C200HW-CORT21-V1 only uses 11-bit identifiers, and therefore these bits are fixed to 0.

3-3-2-9 Storing / Restoring of Parameters

Storing Parameters

The C200HW-CORT21-V1 allows storage of parameters of the Communication Profile Area and of the Manufacturer Specific Profile Area in non-volatile memory. Appendix B lists the stored parameters.

At power-on, the Unit is initialised with the stored parameter values.

The parameters are stored via an SDO access to the object at index 1010h. The following figure shows the structure of this object.

Index	Sub-index	Object	Default value
1010h	-	Store Parameters	-
1010h	00h	Number of entries	1
1010h	01h	Save all parameters	1

Reading from sub-index 01h returns the value ‘1’, which means that parameters are only saved on command. The parameters are saved when the ASCII codes of the signature text ‘save’ are written to sub-index 01h.

Note that the character 's' is sent as the least significant byte, therefore the 32-bit 'value' of the signature text becomes 65766173h:

Store Parameter Signature

e	v	a	s
65h	76h	61h	73h

Any other value than 65766173h written to sub-index 01h will result in an abort response.

Storing of all parameters takes approximately four seconds. Writing to any of the parameters that need to be stored, or issuing another store or restore command during this period will not be possible and will result in an abort message. If the Unit is reset during these four seconds, the Unit will:

- restart with factory default values
- indicate an error on the 7-segment display (see section 6-5)
- send an emergency message to indicate an non-volatile memory error (see section 6-2, EEC = FF00h).

This error is also indicated when it is not possible to write to the non-volatile memory. The error is resolved when a store command has been carried out successfully.

Restoring Parameters

The restoring of parameter values to the default values, as defined in the communication and manufacturer specific profile specification, is done via a SDO access to the object at index 1011h. The following figure shows the structure of this object.

Index	Sub-index	Object	Default value
1011h	-	Restore Default Parameters	-
1011h	00h	Number of entries	1
1011h	01h	Restore all parameters	1

Reading from sub-index 01h returns the value ‘1’, which means that the Unit supports the restoring of parameters. The parameters are restored when the ASCII codes for the signature text ‘load’ are written to sub-index 01h.

Note that the character 'l' is sent as the least significant byte, therefore the 32-bit 'value' of the signature text becomes 64616F6Ch:

Restore Parameter Signature

d	a	o	l
64h	61h	6Fh	6Ch

Any other value than 64616F6Ch written to sub-index 01h will result into an abort response. If the correct signature is transmitted and the Unit is not able to restore the parameters due to a non-volatile memory error, an emergency message is transmitted (see section 6-2, EEC = FF00h).

Note The restored default parameter values become active after the Unit has been restarted by a (software) reset or after receiving the NMT command ‘Reset_Communication’ or ‘Reset_Node’.

The storing / restoring of parameter values can also be done with a network configuration tool that sends the correct signature.

4 PLC interface

This section describes the interface with the user via the PLC system. This includes Unit settings to configure the Unit plus the control and status areas.

4-1 Input and Output Data

Link between 2 buses

The C200HW-CORT21-V1 forms a link between two bus systems: the host PLC's I/O bus on one side, and the CANopen network on the other. The Unit can be considered as a slave to both systems: the I/O bus communication is controlled by the CPU of the host PLC, the CANopen communication is controlled by the CANopen NMT master.

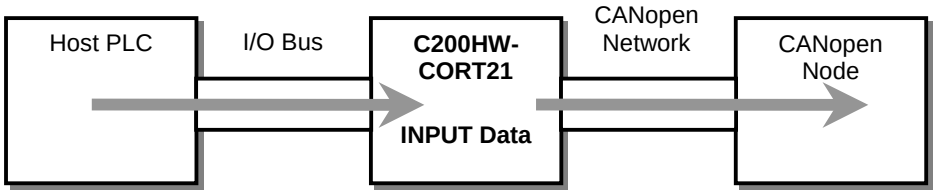
Definition

Being a slave to two systems may cause confusion as to which data should be considered 'input data' and which is 'output data'.

In this manual all I/O data communication is defined from the point of view of the CANopen network. When configuring the CANopen network, the C200HW-CORT21-V1 will be considered as an I/O 'slave' unit. Instead of having physical input and output signal connections, it exchanges its I/O data with the host PLC.

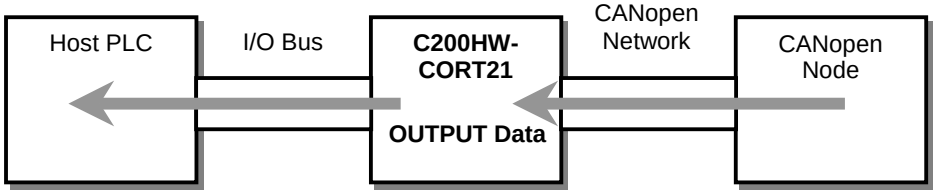
INPUT Data

INPUT Data is process data which the C200HW-CORT21-V1 transmits onto the CANopen network. The Unit reads the input data from the mapped PLC areas during the PLC's I/O refresh. The moment of transmission to other CANopen nodes depends on the configured transmission type.



OUTPUT Data

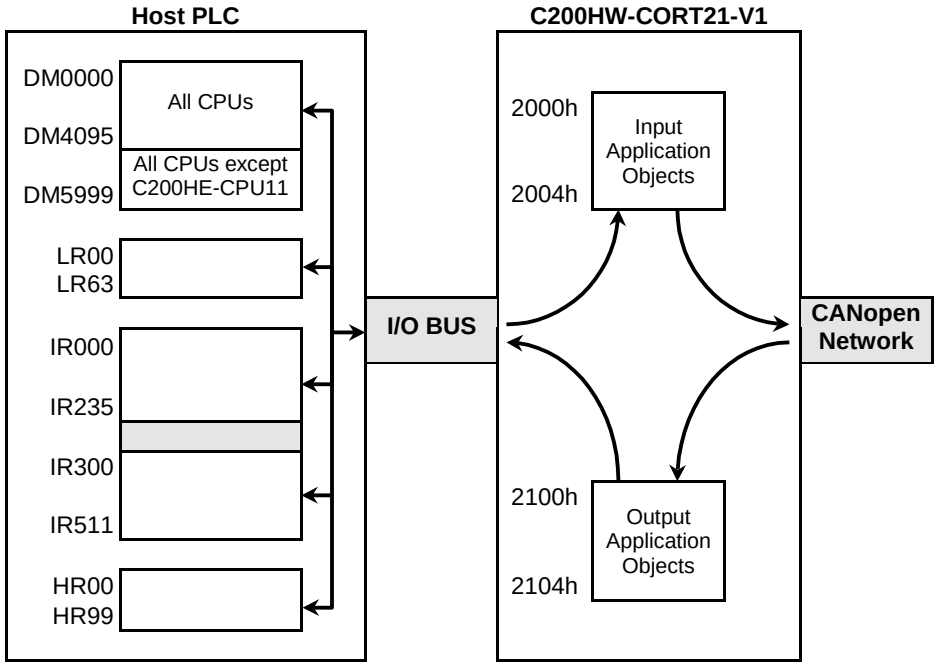
OUTPUT Data is process data which the C200HW-CORT21-V1 receives from the CANopen network. The Unit writes the output data to the mapped PLC areas during the PLC's I/O refresh. If the transmission type of a receive PDO is set to synchronous, the data of the respective PDO is only transferred to the PLC after the reception of the configured SYNC Object. Asynchronous PDO data is always immediately transferred to the PLC at the time of the next I/O refresh.



4-2 I/O Data mapping

Data flow

The figure below shows the flow of remote I/O data in the PLC system.



The core of the C200HW-CORT21-V1 is the object dictionary. The Input and Output Application Objects are part of this object dictionary.

Every PLC I/O refresh, the input data is transferred via the I/O bus on the PLC backplane to the corresponding Input Application Objects. The actual transmission of the application objects over the CANopen Network depends on the TPDO mapping and the TPDO communication parameters.

Output data, received from the CANopen Network, is first stored in the corresponding Output Application Objects in the Unit. The transmission type of the RPDO determines at which PLC I/O refresh the data is written to the Host PLC. The data of an asynchronous RPDO is always transferred during the I/O refresh. The data of a synchronous RPDO is only transferred after the reception of the configured SYNC Object.

I/O refresh

By default, I/O refreshes are executed at the end of each PLC program cycle, but they can also be triggered by the IORF instruction. See the applicable PLC CPU Operation Manual for more details.

4-2-1 Application Objects

C200HW-CORT21-V1 does not follow a pre-defined CANopen Device Profile. It uses the Manufacturer Specific Profile Area (see section 1-2-2-1).

The following figure shows the structure of the Manufacturer Specific Profile Area of the C200HW-CORT21-V1. The input application objects are located at indices 2000h ~ 2004h and the output application objects at indices 2100h ~ 2104h.

Index	Sub-index	Object	Default value
2000h	-	Default_Input_Bytes	-
2000h	00h	Number of entries	4
2000h	01h	8_bit_input1	0
2000h	02h	8_bit_input2	0
2000h	03h	8_bit_input3	0
2000h	04h	8_bit_input4	0
2001h	-	8-bit input network variables	-
2001h	00h	Number of entries	100
2001h	01h	8_bit_input1	0
	⋮	⋮	
2001h	64h	8_bit_input100	0
2002h	-	16-bit input network variables	-
2002h	00h	Number of entries	100
2002h	01h	16_bit_input1	0
	⋮	⋮	
2002h	64h	16_bit_input100	0
2003h	-	32-bit input network variables	-
2003h	00h	Number of entries	50
2003h	01h	32_bit_input1	0
	⋮	⋮	
2003h	32h	32_bit_input50	0
2004h	-	64-bit input network variables	-
2004h	00h	Number of entries	25
2004h	01h	64_bit_input1	0
	⋮	⋮	
2004h	19h	64_bit_input25	0

Index	Sub-index	Object	Default value
2100h	-	Default_Output_Bytes	-
2100h	00h	Number of entries	4
2100h	01h	8_bit_output1	0
2100h	02h	8_bit_output2	0
2100h	03h	8_bit_output3	0
2100h	04h	8_bit_output4	0
2101h	-	8-bit output network variables	-
2101h	00h	Number of entries	100
2101h	01h	8_bit_output1	0
	⋮	⋮	
2101h	64h	8_bit_output100	0
2102h	-	16-bit output network variables	-
2102h	00h	Number of entries	100
2102h	01h	16_bit_output1	0
	⋮	⋮	
2102h	64h	16_bit_output100	0
2103h	-	32-bit output network variables	-
2103h	00h	Number of entries	50
2103h	01h	32_bit_output1	0
	⋮	⋮	
2103h	32h	32_bit_output50	0
2104h	-	64-bit output network variables	-
2104h	00h	Number of entries	25
2104h	01h	64_bit_output1	0
	⋮	⋮	
2104h	19h	64_bit_output25	0

Note The object names are the internal device names and do not have to correspond with the names used in the network configuration tool. The names chosen in the figure above are just to indicate the type of the object.

The C200HW-CORT21-V1 supports 4 different data types for input and output data:

8-bit data objects
16-bit data objects
32-bit data objects
64-bit data objects

The Default_Input_Bytes and Default_Output_Bytes are the application objects that are by default mapped to the four transmit and receive PDOs of the predefined connection set (see also section 1-3-5).

Network variables

The objects located at the index 2001h ~ 2004h and 2101h ~ 2104h are so called network variables. These objects are defined in the EDS-file (see appendix A) as dynamic channels because they are not present in the object dictionary structure until the specific object is linked to another device.

The mapping of the application objects is twofold:

1. Mapping of the application objects onto PDOs
2. Mapping of the application objects onto PLC memory locations

4-2-2 Mapping of the Application Objects onto PDOs

When the Unit starts up with the default values as defined in the communication profile, only the Default_Input_Bytes and Default_Output_Bytes are mapped. The mapping of these objects is listed in the following table.

PDO	Mapped Data Object	
	Index	Sub-index
TPDO1	2000h	01
TPDO2	2000h	02
TPDO3	2000h	03
TPDO4	2000h	04
RPDO1	2100h	01
RPDO2	2100h	02
RPDO3	2100h	03
RPDO4	2100h	04

The PDOs listed in the table are the PDOs of the predefined connection set (see section 1-3-5).

The objects located at indices 2001h ~ 2004h and 2101h ~ 2104h can be mapped by configuring the PDO mapping parameters to point to these objects (see section 3-3-2-2). In this way it is also possible to change the mapping of the predefined connection set. For example it is possible to assign object 2001h with sub-index 01 to TPDO1 and to assign the Default_Input_Bytes to other transmit PDOs than TPDO1 ~ TPDO4.

The most common way of mapping the objects is by using a network configuration tool. This tool assigns the objects to the correct PDO when linking the objects of two devices together (see the end of section 3-3-2-1).

To prevent having to map objects to PDOs and setting the associated communication parameters every time the Unit is powered-up, the Unit enables storage of the parameters in non-volatile memory (see section 3-3-2-9).

4-2-3 Mapping of the Application Objects onto PLC memory locations

Default mapping

The application objects at index 2000h and 2100h (Default_Input_Bytes, Default_Output_Bytes) are mapped to the status and control area of the C200HW-CORT21-V1 (see section 4-3). Their data content is transferred between the PLC and the Unit in each I/O refresh.

Additional mapping

If additional objects are required, the Unit settings of the C200HW-CORT21-V1 need to be configured. It is possible to map the additional application objects to the DM, LR, IR and HR areas of the PLC memory. The input and output application objects can be mapped independently up to a maximum of 100 input words and 100 output words.

Unit settings

The mapping to PLC memory locations depends on the Unit settings made in a dedicated DM area. The assigned DM area depends on the Machine number setting:

MACHINE No.	DM area*	
0	DM1000 to DM1019	All C200H PLC models
1	DM1100 to DM1119	
2	DM1200 to DM1219	
3	DM1300 to DM1319	
4	DM1400 to DM1419	
5	DM1500 to DM1519	
6	DM1600 to DM1619	
7	DM1700 to DM1719	
8	DM1800 to DM1819	
9	DM1900 to DM1919	
A	DM2000 to DM2019	All C200H PLC models except: C200HE, C200HG-CPU3❑-E/CPU4❑-E, C200HX-CPU3❑-E/CPU4❑-E
B	DM2100 to DM2119	
C	DM2200 to DM2219	
D	DM2300 to DM2319	
E	DM2400 to DM2419	
F	DM2500 to DM2519	

***** Alternatively starting at DM7000 to DM8500, selected by PLC setup of C200H: DM6602 ≠ 0000 (see Operation Manual of CPU Unit).

***** For CS1 allocations, see Appendix C.

The first word in the DM area allocated to the Unit will be indicated by **DM m**, the last word by **DM m+19**.

DM m ~ DM m+15 are used to configure the mapping of the application objects onto PLC memory locations.

DM m+16 ~ DM m+19 are for setting the baud rate (see appendix D).

The values entered in the Unit settings area are only transferred to the Unit during initialisation, i.e. at power on and at Special I/O Unit restart.

Default mode

The Unit is said to operate in Default Mode when the values of all Unit settings are zero. In Default Mode, only the application objects of the predefined connection set are exchanged with the PLC via the control and status area (see section 4-3).

The following table lists the DM words for configuration of the I/O data mapping, with the possible values and their meaning.

Setting values in BCD

Except for the definition of the start address in the PLC CPU, all values are in BCD. To distinguish between start addresses in different PLC memory areas, the first digit of the 'start address' indicates the PLC memory area, and the following three digits indicate the address in the area in BCD format.

DM word	Value	Meaning
m	**Slave input mapping mode**	
	≠ 0001	Default input mapping, from IR n+1 and IR n+2 only. Any settings in DM m+1 ~ DM m+7 are ignored.
	= 0001	Additional input mapping by DM m+1 ~ DM m+7
m+1	**Start address range of 8-bit input area**	
	0000 ~ 5999	DM0000 ~ DM5999 (C200HE: 0000 ~ 4095)
	A000 ~ A235	IR000 ~ IR235
	A300 ~ A511	IR300 ~ IR511
	B000 ~ B099	HR00 ~ HR99
	C000 ~ C063	LR00 ~ LR63
	other	Setting error *[1]
m+2	**Number of 8-bit input objects**	
	0000 ~ 0100	0 ~ 100 8-bit objects to be transferred from PLC to C200HW-CORT21-V1 *[2]
	other	Setting error *[1]
m+3	**Start address range of 16-bit input area**	
	See m+1	
m+4	**Number of 16-bit input objects**	
	0000 ~ 0100	0 ~ 100 16-bit objects to be transferred from PLC to C200HW-CORT21-V1 *[2]
	other	Setting error *[1]
m+5	**Start address range of 32- / 64-bit input area**	
	See m+1	
m+6	**Number of 32-bit input objects**	
	0000 ~ 0050	0 ~ 50 32-bit objects to be transferred from PLC to C200HW-CORT21-V1 *[2]
	other	Setting error *[1]
m+7	**Number of 64-bit input objects**	
	0000 ~ 0025	0 ~ 25 64-bit objects to be transferred from PLC to C200HW-CORT21-V1 *[2]
	other	Setting error *[1]
m+8	**Slave output mapping mode**	
	≠ 0001	Default output mapping, to IR n+6 and IR n+7 only. Any settings in DM m+9 ~ DM m+15 are ignored.
	= 0001	Additional output mapping by DM m+9 ~ DM m+15
m+9	**Start address range of 8-bit output area**	
	See m+1	
m+10	**Number of 8-bit output objects** (valid if DM m = 0001)	
	0000 ~ 0100	0 ~ 100 8-bit objects to be transferred from C200HW-CORT21-V1 to PLC *[2]
	other	Setting error *[1]
m+11	**Start address range of 16-bit output area**	
	See m+1	
m+12	**Number of 16-bit output objects**	
	0000 ~ 0100	0 ~ 100 16-bit objects to be transferred from C200HW-CORT21-V1 to PLC *[2]
	other	Setting error *[1]
m+13	**Start address range of 32- / 64-bit output area**	
	See m+1	
m+14	**Number of 32-bit output objects**	
	0000 ~ 0050	0 ~ 50 32-bit objects to be transferred from C200HW-CORT21-V1 to PLC *[2]
	other	Setting error *[1]
m+15	**Number of 64-bit output objects**	
	0000 ~ 0025	0 ~ 25 64-bit objects to be transferred from C200HW-CORT21-V1 to PLC *[2]
	other	Setting error *[1]

*[1] The setting error(s) is indicated in IR n+3 and IR n+4 (see section 4-3-2) and the ERR LED will be blinking. Only the control and status area is transferred, so no transfer of additional I/O data and also no CANopen communication.

*[2] If the specified number of objects would make the area exceed the boundary of the PLC's DM, LR, IR or HR area, it is classified as a setting error and the error handling will be the same as in *[1].

Note A setting error also occurs when the total amount of data of all input or output areas exceeds 100 words. The error handling is the same as in *[1].

Note The DM settings m+1, m+3, m+5, m+9, m+11, and m+13 indicate the *start address* of the respective objects in the PLC memory. The amount of memory that is allocated after this start address depends on the number and type of mapped objects. See m+1 for the available memory per PLC area. Note that a 32-bit object requires 2 words and a 64-bit object 4 words and therefore these objects can not start at the highest PLC address in the memory range. This will result in a DM setting error and IR n+4 will indicate that the boundary of the respective PLC memory area is exceeded.

It is possible to have both 32-bit *and* 64-bit objects. In the PLC memory area, the 64-bit objects will start directly after the 32-bit objects.

Allocation of objects to PLC memory locations:

The 8-bit objects with an odd sub-index in the object dictionary will be in the least significant byte of a PLC memory location. The even sub-indexes at the most significant part.

The least significant 16-bits of a 32-bit or a 64-bit object will be at the lowest PLC address. The higher bits are allocated to the next PLC address(es).

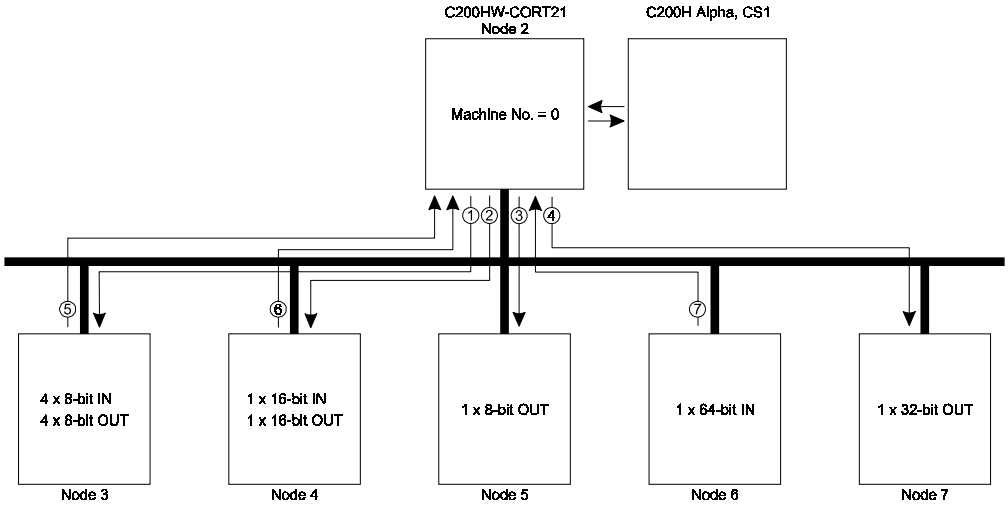
⚠ **Caution** The Unit cannot check the validity of the contents of any PLC data area, from which input data is to be transferred. Any data present in the area will be transferred to the input application objects in the object dictionary of the Unit.

⚠ **Caution** The Unit does not check if the selected output areas in the host PLC is in use by other Units, e.g. other fieldbus Master or Slave Units mounted in the same system. If so, this Unit’s data may overwrite another Unit’s data, or vice versa.

4-2-4 Mapping Example

This section gives an example of how application objects can be mapped to PDOs and PLC memory locations.

The following figure shows the network configuration. It contains the C200HW-CORT21-V1 and 5 other CANopen nodes. The machine number of the C200HW-CORT21-V1 is set to 0.



There are in total 7 logical connections to the C200HW-CORT21-V1. From the C200HW-CORT21-V1 point of view, connections 1 to 4 are transmit PDOs and connections 5 to 7 are receive PDOs.

On the CANopen side, the communication and mapping parameters of the PDOs must be configured to establish the communication relationship as is depicted in the figure above. On the PLC side, the DM settings must be configured because the predefined connection set is not sufficient.

The following table shows a possible configuration for this example.

Connection Number	PDO	Application Object		PLC address
		Index	Sub Index	
1	TPDO1	2000h	01h	IR101, bit 00 ~ 07
		2000h	02h	IR101, bit 08 ~ 15
		2000h	03h	IR102, bit 00 ~ 07
		2000h	04h	IR102, bit 08 ~ 15
2	TPDO2	2002h	01h	LR10
3	TPDO3	2001h	01h	IR050, bit 00 ~ 07
4	TPDO4	2003h	01h	HR00, HR01
5	RPDO1	2100h	01h	IR106, bit 00 ~ 07
		2100h	02h	IR106, bit 08 ~ 15
		2100h	03h	IR107, bit 00 ~ 07
		2100h	04h	IR107, bit 08 ~ 15
6	RPDO2	2102h	01h	IR350
7	RPDO3	2104h	01h	DM0100 ~ DM0103

The next two sub-sections show how this configuration can be realised.

4-2-4-1 CANopen configuration

The table below lists the communication and mapping parameters of the 7 PDOs. Note that this is just an example and that it is also possible to choose other values than the ones listed in the table.

Note that the COB-IDs of the other nodes' PDOs (nodes 3 ~ 7) should be set to the same value as the COB-ID of C200HW-CORT21-V1 for the respective connection. So every connection has one unique COB-ID and this must be set on the receiving node as well as on the transmitting node.

In this example, the transmission type of all PDOs is set to Asynchronous (254).

Index	Sub-index	Object	Default value
1400h	-	Receive PDO1 parameter	-
1400h	00h	Number of entries	2
1400h	01h	COB-ID	202h
1400h	02h	Transmission Type	254
1401h	-	Receive PDO2 parameter	-
1401h	00h	Number of entries	2
1401h	01h	COB-ID	302h
1401h	02h	Transmission Type	254
1402h	-	Receive PDO3 parameter	-
1402h	00h	Number of entries	2
1402h	01h	COB-ID	402h
1402h	02h	Transmission Type	254
1600h	-	Receive PDO1 mapping	-
1600h	00h	Number of entries	4
1600h	01h	mapped object	21000108h
1600h	02h	mapped object	21000208h
1600h	03h	mapped object	21000308h
1600h	04h	mapped object	21000408h
1601h	-	Receive PDO2 mapping	-
1601h	00h	Number of entries	1
1601h	01h	mapped object	21020110h
1602h	-	Receive PDO3 mapping	-
1602h	00h	Number of entries	1
1602h	01h	mapped object	21040140h
1800h	-	Transmit PDO1 parameter	-
1800h	00h	Number of entries	2
1800h	01h	COB-ID	182h
1800h	02h	Transmission Type	254

Index	Sub-index	Object	Default value
1801h	-	Transmit PDO2 parameter	-
1801h	00h	Number of entries	2
1801h	01h	COB-ID	282h
1801h	02h	Transmission Type	254
1802h	-	Transmit PDO3 parameter	-
1802h	00h	Number of entries	2
1802h	01h	COB-ID	382h
1802h	02h	Transmission Type	254
1803h	-	Transmit PDO4 parameter	-
1803h	00h	Number of entries	2
1803h	01h	COB-ID	482h
1803h	02h	Transmission Type	254
1A00h	-	Transmit PDO1 mapping	-
1A00h	00h	Number of entries	4
1A00h	01h	mapped object	20000108h
1A00h	02h	mapped object	20000208h
1A00h	03h	mapped object	20000308h
1A00h	04h	mapped object	20000408h
1A01h	-	Transmit PDO2 mapping	-
1A01h	00h	Number of entries	1
1A01h	01h	mapped object	20020110h
1A02h	-	Transmit PDO3 mapping	-
1A02h	00h	Number of entries	1
1A02h	01h	mapped object	20010108h
1A03h	-	Transmit PDO4 mapping	-
1A03h	00h	Number of entries	1
1A03h	01h	mapped object	20030120h

4-2-4-2 PLC configuration

By default, the C200HW-CORT21-V1 only transfers the four input and output bytes mapped to the PDOs of the predefined connection set via the control and status area. This is not enough for the network configuration of this example. The following additional data objects need to be transferred:

1 x 16-bit OUT	Data transmitted by node 4
1 x 16-bit IN	Data received by node 4
1 x 8-bit IN	Data received by node 5
1 x 64-bit OUT	Data transmitted by node 6
1 x 32-bit IN	Data received by node 7

Note that the data to / from node 3 is transferred via the Default_Input_Bytes (index 2000h) and Default_Output_Bytes (index 2100h) which have a fixed allocation to the control and status area of the C200HW-CORT21-V1.

The following table list the DM settings that have to be made to transfer the additional data objects.

DM word	Value	Description
1000	0001	Additional input mapping
1001	A050	Start address of 8-bit input area = IR050
1002	0001	1 x 8-bit input object
1003	C010	Start address of 16-bit input area = LR10
1004	0001	1 x 16-bit input object
1005	B000	Start address of 32- / 64-bit input area = HR00
1006	0001	1 x 32-bit input object
1007	0000	0 x 64-bit input object
1008	0001	Additional output mapping
1009	0000	Start address of 8-bit output area = DM0000
1010	0000	0 x 8-bit output object
1011	A350	Start address of 16-bit output area = IR350
1012	0001	1 x 16-bit output object
1013	0100	Start address of 32- / 64-bit output area = DM0100
1014	0000	0 x 32-bit output object
1015	0001	1 x 64-bit output object

4-3 Control and Status area

After initialisation of the Unit (RUN LED is ON), the control and status words are exchanged between the PLC and the Unit during each I/O refresh. The mapping of the control words and unit status words depends on the Machine number set by the rotary switch at the front of the Unit.

Machine number	IR area	
0	IR100 ~ IR107	All C200H PLC models
1	IR110 ~ IR117	
2	IR120 ~ IR127	
3	IR130 ~ IR137	
4	IR140 ~ IR147	
5	IR150 ~ IR157	
6	IR160 ~ IR167	
7	IR170 ~ IR177	
8	IR180 ~ IR187	
9	IR190 ~ IR197	
A	IR400 ~ IR407	All C200H PLC models except: C200HE, C200HG-CPU3❑-E/CPU4❑-E, C200HX-CPU3❑-E/CPU4❑-E
B	IR410 ~ IR417	
C	IR420 ~ IR427	
D	IR430 ~ IR437	
E	IR440 ~ IR447	
F	IR450 ~ IR457	

For allocation in CS1-series PLC's, see Appendix C.

The first word in the IR area allocated to the Unit will be indicated by **IR n**, the last word by **IR n+7**. The following table gives an overview of the meaning of these words.

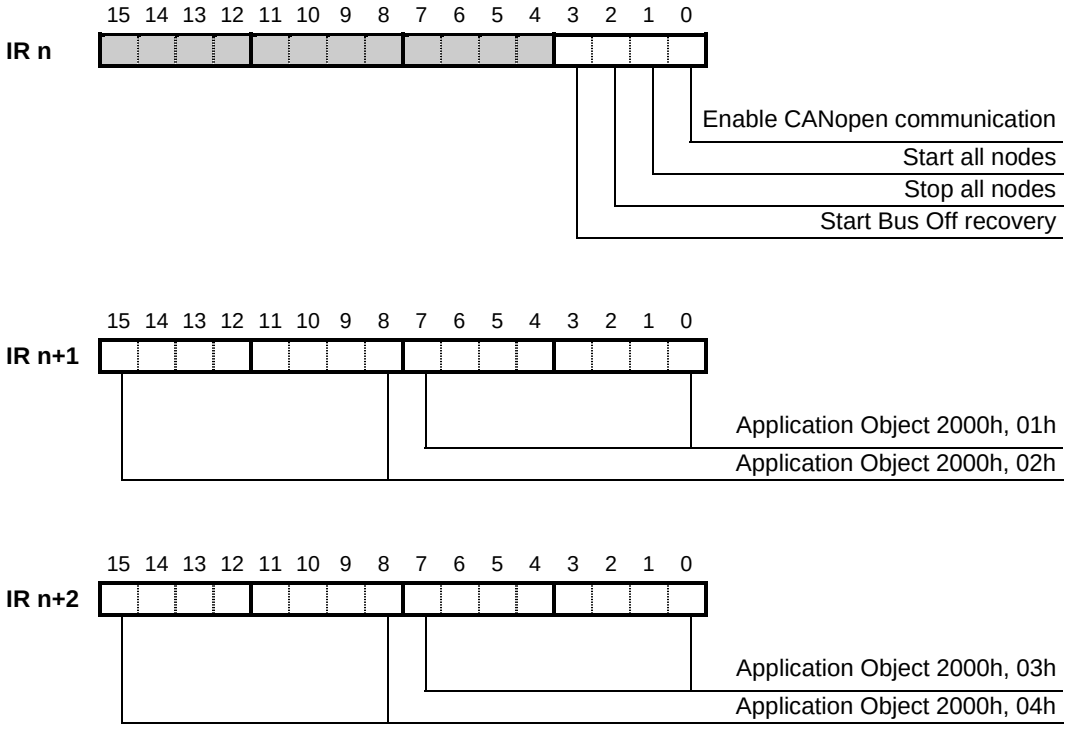
IR word	Meaning	Data Direction
n	Control word	PLC → C200HW-CORT21-V1
n+1	Input word 1	
n+2	Input word 2	
n+3	Status word 1	C200HW-CORT21-V1 → PLC
n+4	Status word 2	
n+5	Status word 3	
n+6	Output word 1	
n+7	Output word 2	

Refer to the next two sub-sections for a detailed explanation of the Control and Status words.

4-3-1 Control words

The three control words, IR n ~ IR n+2, are shown below.

Any bits of the control words which are not assigned to a specific function, can freely be used as work bits. These bits will be ignored by the C200HW-CORT21-V1.



IR n.00 Enable CANopen communication

The main purpose of this bit is enabling / disabling the CANopen communication, but it also allows the CANopen network to detect if the host PLC is in abnormal operation.

This bit in the PLC memory needs to be set (in the PLC program or forced by the user) to enable the CANopen communication. In case of a fatal PLC error, IR n.00 will revert to 0 and the Unit will transmit an emergency message to notify the other nodes on the CANopen network. After transmission of this message, the C200HW-CORT21-V1 will refrain from CANopen communication until the bit is set to ON again.

Switching of IR n.00 does not have any effect on the contents of the object dictionary.

0: CANopen communication is disabled.

The CANopen data exchange between PLC and CAN-bus is stopped. On a falling edge, the C200HW-CORT21-V1 will first send out an emergency telegram before disabling the communication. The status bits IR n+3.01 and IR n+3.00 will become 0 and the 7-segment display will indicate FFh. In this mode it is not possible to access the object dictionary, but it is still possible to transmit / receive user defined CAN messages (see section 5).

1: CANopen communication is enabled.

When this bit is first set after power-on or reset of the Unit, the C200HW-CORT21-V1 will send out a standardised boot-up message. After this, the Unit enters Pre-Operational state.

After a 1 → 0 transition was made to disable the CANopen communication, the following 0 → 1 transition will result in the transmission of an emergency telegram to notify the network that communication will resume. After this, the Unit re-enters Pre-Operational state.

Refer to section 6-2 for more details about the emergency message.

IR n.01 **Start all nodes**

The function of this is bit is to request all nodes on the CANopen network (including this Unit) to enter Operational state by broadcasting the NMT command 'Start_Remote_Node' (see section 1-3-4-1).

CANopen communication must be enabled (IR n.00 = 1), and IR n.02 (see below) should not be set simultaneously.

0: No action.

1: On a rising edge, the NMT command 'Start_Remote_Node' is broadcast once.

IR n.02 **Stop all nodes**

The function of this is bit is to request all nodes on the CANopen network (including this Unit) to enter Pre-Operational state by broadcasting the NMT command 'Enter_Pre-Operational_State' (see section 1-3-4-1).

CANopen communication must be enabled (IR n.00 = 1).

0: No action.

1: On a rising edge, the NMT command 'Enter_Pre-Operational_State' is broadcast once.

IR n.03 **Start Bus Off recovery**

The function of this bit is to start a Bus Off recovery after the Unit has gone "Bus Off" (See IR n+3.10) due to an abnormal rate of errors on the bus. If the cause of the error is not removed, the recovery may not be successful.

0: No action.

1: On a rising edge, the Bus Off recovery sequence is started.

After a successful Bus Off recovery, the C200HW-CORT21-V1 will enter Pre-Operational state.

IR n.04 ~ IR n.15 **Not used** by C200HW-CORT21-V1.

IR n+1, IR n+2 **Application Objects at index 2000h**

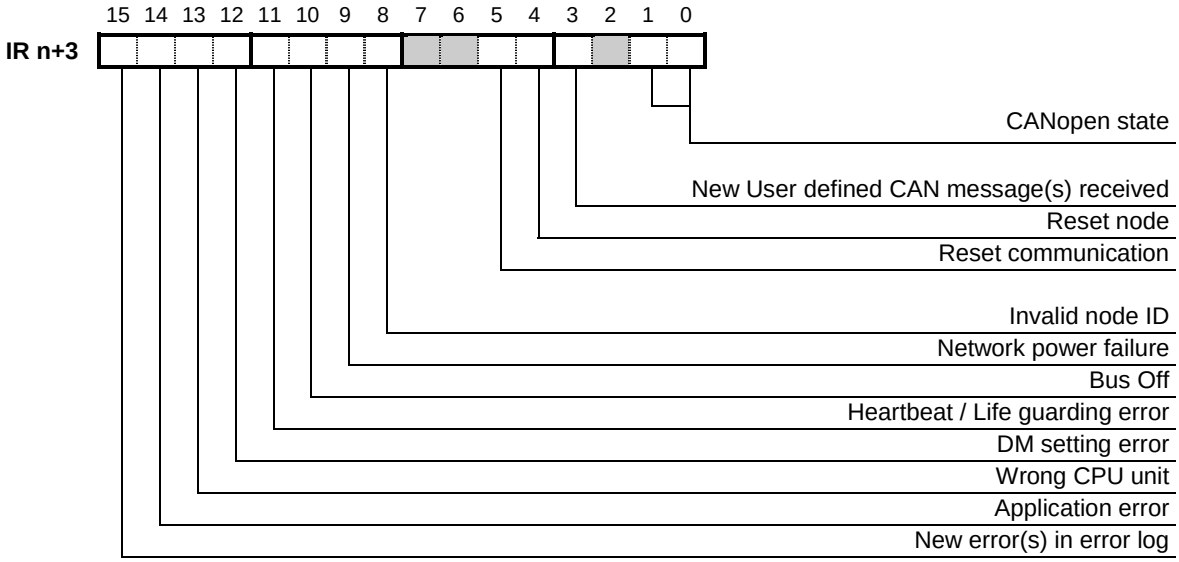
The manufacturer specific profile of C200HW-CORT21-V1 has four single-byte input objects allocated at index 2000h with sub-indices 01h to 04h.

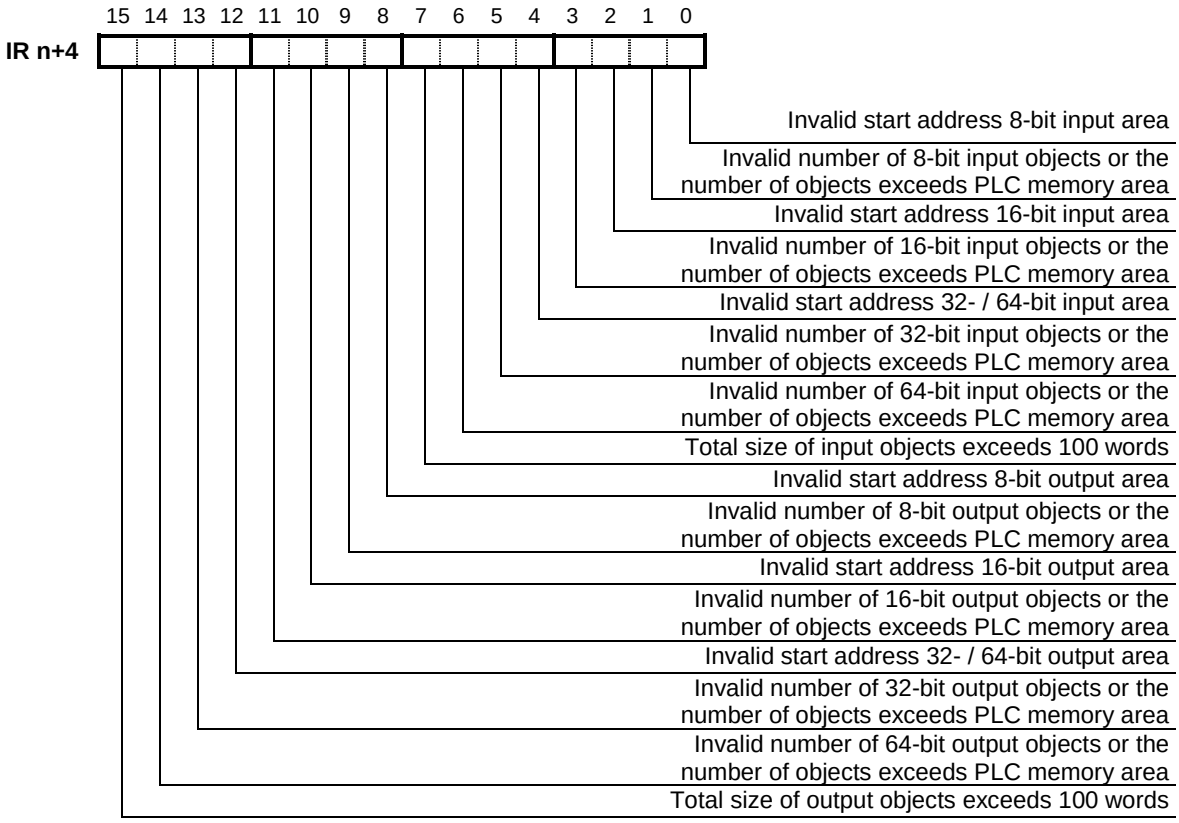
The content of IR n+1 and IR n+2 is mapped to these input application objects, which are - by default - assigned to the 4

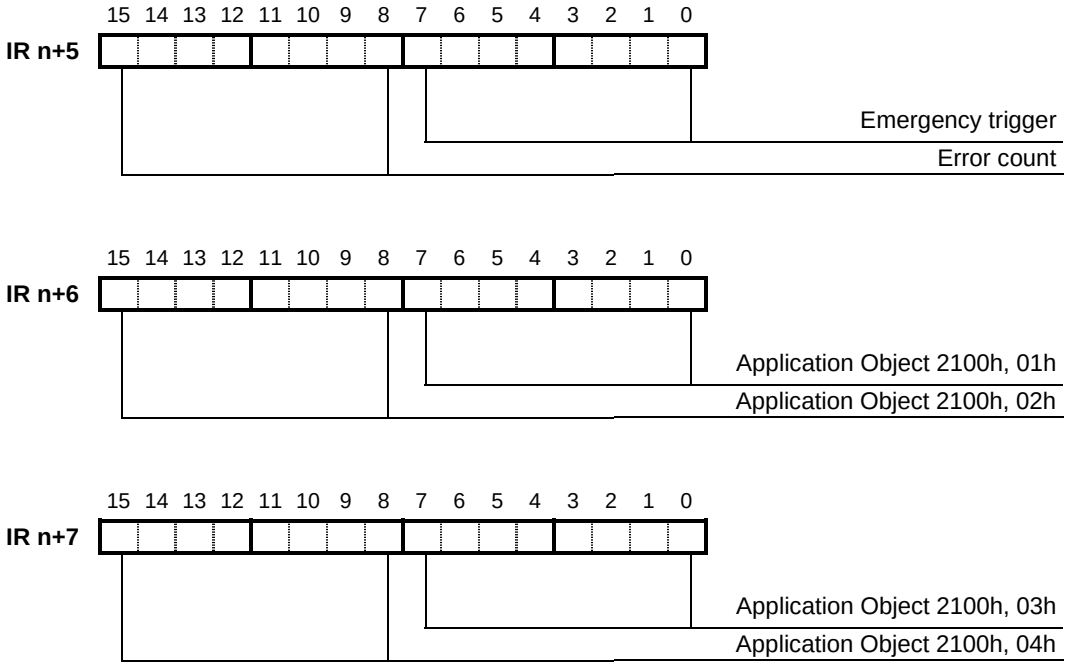
transmit PDOs of the predefined connection set.

4-3-2 Status words

The five status words of the C200HW-CORT21-V1 are shown below.







IR n+3.00, IR n+3.01 **CANopen state**

These two bits indicate the CANopen state of the C200HW-CORT21-V1.

IR n+3.01	IR n+3.00	CANopen state
0	0	**Not active.** CANopen is not active when: • Initialisation is not finished yet. • Unit has gone 'Bus Off' due to non-recoverable errors detected on the CANopen network. • IR n.00 is not set.
0	1	**Pre-Operational** This state is entered autonomously after initialisation or after Bus Off recovery, and after receiving the NMT command 'Enter_Pre-Operational_State'. The Unit is accessible via SDOs to allow configuration.
1	0	**Operational** This state is entered after receiving the NMT command 'Start_Remote_Node'. The Unit is fully operational and can be accessed via SDOs and PDOs.
1	1	**Stopped** This state is entered after receiving the NMT command 'Stop_Remote_Node' In this state all data communication (via SDOs as well as PDOs) is disabled.

Refer to section 1-3-4-1 for more details about the CANopen state machine.

IR n+3.02 **Not used** by C200HW-CORT21-V1.

IR n+3.03 **New user defined CAN message(s) received**

The C200HW-CORT21-V1 supports the reception of user-defined (i.e. non-CANopen) CAN messages. Refer to section 5-8 for details. This bit is set when at least one specified CAN message matching the identifier filter has been received since the previous I/O refresh. The content of the newly received message(s) has been transferred to the designated PLC area and can be processed when this bit is ON.

0: No new user defined CAN message was received since the previous I/O refresh.

1: New user defined CAN message(s) have been received and were transferred to the PLC memory.

IR n+3.04 **Reset node**

This bit is set for one PLC scan when the C200HW-CORT21-V1 has received the NMT command 'Reset_Node'. This command will reset all application objects to zero and set the objects of the communication profile area to the power-on values.

0: The NMT command 'Reset_Node' was not received during the last PLC scan.

1: The NMT command 'Reset_Node' was received during the last PLC scan.

Refer to section 1-3-4-1 for more details about the CANopen state machine.

IR n+3.05 **Reset communication**

This bit is set for one PLC scan when the C200HW-CORT21-V1 has received the NMT command 'Reset_Communication'. This command will set the objects of the communication profile area to the power-on values.

0: The NMT command 'Reset_Communication' was not received during the last PLC scan.

1: The NMT command 'Reset_Communication' was received during the last PLC scan.

Refer to section 1-3-4-1 for more details about the CANopen state machine.

IR n+3.06, IR n+3.07 **Not used** by C200HW-CORT21-V1.

IR n+3.08 **Invalid node ID**

The node ID is checked when CANopen is enabled (IR n.00 is set). If the node ID is invalid, CANopen cannot be enabled.

0: The selected node ID is within the allowed range (1..127)

1: The selected node ID is not within the allowed range (1..127). All node address switches are OFF. This error is also indicated on the 7-segment display (see section 6-5).

IR n+3.09 Network power failure

The network transmission interface of the C200HW-CORT21-V1 is galvanically isolated and requires 24 V DC power supply via pin 1 and 5 of the bus connector.

0: No network power failure.

1: Network power failure.

This error is also indicated on the 7-segment display (see section 6-5).

IR n+3.10 Bus Off

If the C200HW-CORT21-V1 detects an abnormal rate of errors on the bus, the Unit will go 'Bus Off'. This means that all CAN(open) communication is stopped. The Unit will attempt to go online again after toggling IR n.03 ‘Start Bus Off recovery’.

0: Unit is online.

1: Unit has gone Bus Off.

This error is also indicated on the 7-segment display (see section 6-5).

If a network power failure is detected during Bus Off, IR n+3.10 is reset and IR n+3.9 is set. When the network power failure is solved, the Bus Off recovery sequence is started automatically.

After a successful Bus Off recovery, the C200HW-CORT21-V1 will enter pre-operational state.

IR n+3.11 Heartbeat / Life guarding error

The C200HW-CORT21-V1 can monitor the heartbeat of another node *or* perform life guarding on the NMT master that guards the C200HW-CORT21-V1.

In case the Heartbeat mechanism is enabled and the heartbeat producer fails to send the heartbeat within the set heartbeat time (see section 3-3-2-3), IR n+3.11 is set.

In case Life Guarding is enabled and the NMT master fails to request the status within the Node Life Time (see section 3-3-2-4), IR n+3.11 is set.

IR n+3.11 is reset when heartbeat or remote transmission requests from the NMT master have resumed.

0: No Heartbeat / Life guarding error

or the Heartbeat consumer / Life guarding is not enabled.

1: Heartbeat / Life guarding error.

This error is also indicated on the 7-segment display (see section 6-5).

IR n+3.12 DM setting error

If an error is made in the Unit's DM settings (see section 4-2-3), IR n+3.12 is set.

0: No DM setting error. IR n+4 = 0000

1: DM setting error. See IR n+4 for details.

The ERR-LED will be blinking to indicate a non-fatal error, and the CANopen communication is disabled.

IR n+3.13 Wrong CPU Unit

The C200HW-CORT21-V1 can only be mounted on a C200H Alpha- or CS1-series PLC. If any other PLC-series CPU is detected, IR n+3.13 is set.

0: No error

1: The Unit is mounted on a PLC type which is not supported (e.g. C200H, C200HS). The ERR-LED will be ON to indicate a fatal error, and the CANopen communication is disabled.

IR n+3.14 Application error

0: No error

1: Application error.

The ERR-LED is ON and the CANopen communication is disabled. This is a fatal error due to hardware failure and/or memory corruption in the unit; if this error recurs after power off → on, contact your OMRON representative.

IR n+3.15 New error(s) in error log

The C200HW-CORT21-V1 has an error log for errors which have occurred in IOWR / IORD message communication. If new errors have occurred since the last access to the error log, IR n+3.15 is set.

0: No new error(s) in the error log since the last read-access.

1: New error(s) have occurred since the last read-access.

See section 5-3 for more details about the error log.

IR n+4 When status bit IR n+3.12 (DM setting error) is set, the bits of IR n+4 indicate which DM setting(s) caused the error(s).

IR n+5 The most recent Emergency trigger and respective error count are indicated in IR n+5.

The Emergency trigger is a bit-field indicating the reason why an emergency message was transmitted, see section 6-2. The error count indicates the number of times this error has occurred. When an error is resolved, the error count byte of IR n+5 is zero. When all errors have been resolved, the Emergency trigger byte is also zero.

IR n+6, IR n+7 Application Objects at index 2100h

The manufacturer specific profile of C200HW-CORT21-V1 has four single-byte output objects allocated at index 2100h with sub-indices 01h to 04h.

These output application objects have a fixed mapping to IR n+6 and IR n+7.

The 4 receive PDOs of the predefined connection set are - by default - assigned to these output application objects.

5 Message communication, IOWR / IORD

This section describes the message communication. The PLC program instructions IOWR and IORD are used to transfer the messages to and from the Unit.

5-1 Message communication

Functions

The message communication is used for the following functions:

1. Reading the error log
2. Writing to the local CANopen object dictionary
3. Reading from the local CANopen object dictionary
4. Transmitting a user-defined CAN message
5. Receiving a user-defined CAN message

The error log stores the 25 most recent errors that have occurred due to message communication. These errors can be read with the IORD instruction.

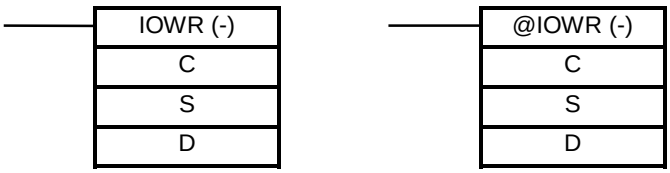
Functions 2 and 3 can be used to read and modify the object dictionary of the Unit.

In order to provide additional flexibility for advanced users, the Unit can be used to transmit and receive user-defined CAN messages. This allows interfacing the C200HW-CORT21-V1 with devices that operate according to vendor- or application-specific CAN-based protocols, different from CANopen. This functionality can be used simultaneously with the CANopen communication.

The next sections will first describe the general use of the IOWR / IORD instructions and the error log. After that, the five message communication functions of the C200HW-CORT21-V1 are explained in detail.

5-2 IOWR / IORD

The ladder symbols for IOWR and IORD are shown below.



C Control code

value: Specifies the function to execute (see below)

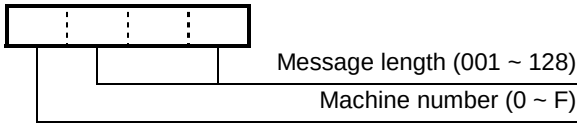
S First source word

value: The start address of the PLC data area that contains the message to be transferred to the Unit.

D Destination information

On C200H Alpha PLCs:

value: Combination of the Machine number of the target Unit and the message length (number of words in BCD).

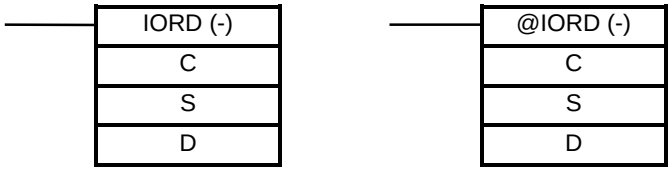


On CS1 PLCs:

The Machine Number and message length occupy 2 words:

D: Machine number of the Unit

D+1: Message length (number of words in BCD)



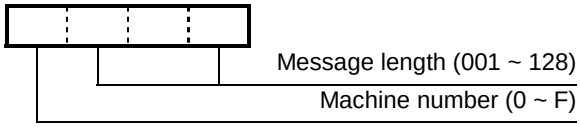
C Control code

value: Specifies the function to execute (see below)

S Source information

On C200H Alpha PLCs:

value: Combination of the Machine number of the Unit and the message length (number of words in BCD).



On CS1 PLCs:

The Machine Number and message length occupy 2 words:

S: Machine number of the Unit

S+1: Message length (number of words in BCD)

D First destination word

value: The start address of the PLC data area to where the message from the Unit is to be transferred.

The following table lists the control codes and message lengths for the C200HW-CORT21-V1.

Function	Control code	Message Length	
		IORD	IOWR
Reading the error log	#0000	1 - 26	N/A.
Writing to the local object dictionary	#0001	N/A.	4 - 7
Reading from the local object dictionary	#0002	2 - 5	2
Transmitting a user defined CAN message	#0003	N/A.	3 - 6
Receiving a user defined CAN message	#0004	N/A.	4

PLC EQ-flag

After the execution of the IOWR or IORD instruction, the PLC EQ-flag is set when the syntax of the instruction, the control code, and the message length are valid.

Note The EQ-flag does not indicate that the content of the message is correct. Therefore it is important to check the error log during the programming and testing of the PLC application.

5-3 Error log

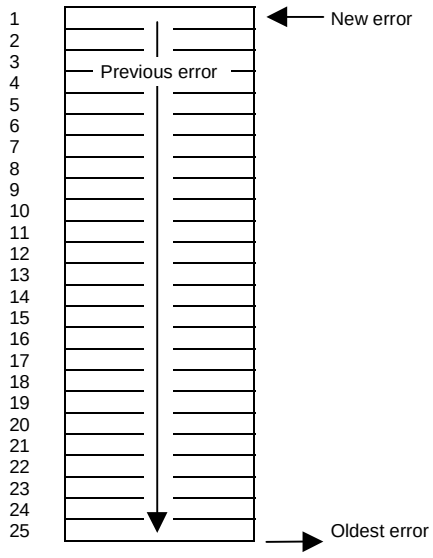
Log size

The C200HW-CORT21-V1 records the 25 most recent errors that have occurred in the IOWR / IORD message communication. The most recent error will always be put at the first position in the list, shifting previous error down. When 25 errors are present in the error log, the oldest error code is removed when a new error occurs.

Indication

Status bit IR n+3.15 is set when one or more errors have occurred since the last time the error log was read. The error log can be read with the IORD instruction. Reading the error log will not clear the log, it will only reset IR n+3.15. The error log is only cleared after a reset or on power-up.

Error log



Possible error codes

The following table lists all possible error codes, the meaning of the error codes and to which function they are applicable.

Error code	Description	Applicable to
0350	Invalid control code, or Function is inactive due to current state i.e. CANopen is disabled	IOWR / IORD
0360	Message length is out of range	Reading the error log
0361	Message length is out of range or does not match the specified data length in the IOWR message structure	Writing to the local object dictionary
0362	Message length is out of range or does not match the data length of the object entry	Reading from the local object dictionary
0363	Message length is out of range or does not match the specified data length in the IOWR message structure	Transmitting a user-defined CAN message
0364	Message length is out of range	Receiving a user-defined CAN message

Error code	Description	Applicable to
0410	Object entry does not exist	Writing to the local object dictionary
0411	Sub-index does not exist	Writing to the local object dictionary
0412	Object entry is read-only	Writing to the local object dictionary
0413	Specified data length in the IOWR message structure is out of range or does not match the object entry	Writing to the local object dictionary
0414	Invalid parameter value	Writing to the local object dictionary
0415	Unable to access specified object. The Unit is currently storing or restoring parameters to / from non-volatile memory.	Writing to the local object dictionary
0420	Object entry does not exist	Reading from the local object dictionary
0421	Sub-index does not exist	Reading from the local object dictionary
0422	Object entry is write-only	Reading from the local object dictionary
0423	The object entry to read was never indicated (by a successful IOWR with control code 2) since power-on or reset.	Reading from the local object dictionary
0424	Unable to access specified object. The Unit is currently storing or restoring parameters to / from non-volatile memory.	Reading from the local object dictionary
0430	Specified data length in the IOWR message structure is out of range	Transmitting a user defined CAN message
0431	Unable to transmit the CAN message	Transmitting a user defined CAN message
0440	Invalid PLC start address	Receiving a user defined CAN message

Note Message length refers to the length of the message specified in the IOWR or IORD instruction.

5-4 Reading the error log

Control code #0000

The error log can be read with the IORD instruction. The following figure shows the IORD message structure.

IORD message structure

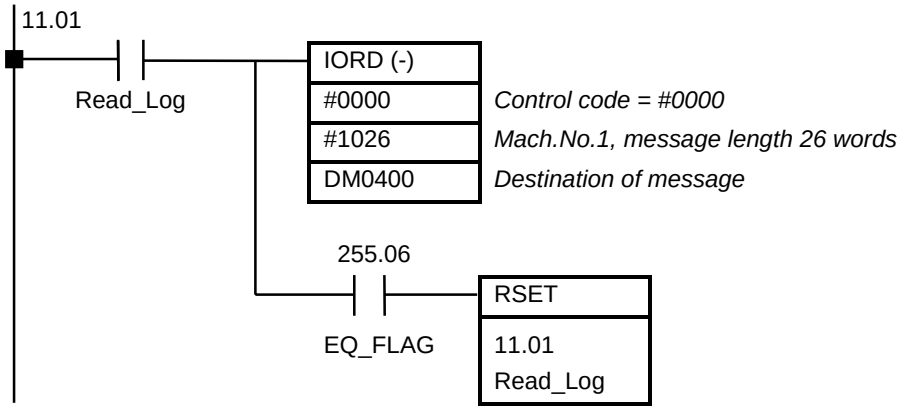
word n	Number of new error codes since the last access to the log [BCD]	range: 0000 ~ 9999
word n+1	Error code 1	
word n+2	Error code 2	
⋮	⋮	
word n+24	Error code 24	
word n+25	Error code 25	

Word n in the message structure indicates how many new errors have occurred since the last time the error log was read. Word n will not count any further than 9999 new errors. Only the 25 most recent errors are stored in the error log.

The message length specified in the source information of the IORD instruction must lie within the range 1 ~ 26. If a length greater than 26 is specified, the Unit will respond with a negative acknowledge and the error code 0360 is stored in the error log. It is recommended to always read the entire log (26 words), and check the number of new errors at word n.

Example

The following example shows how the entire error log is read and copied to DM0400-DM0425. Note that the C200H syntax of the IOWR instruction is used.



5-5 Writing to the local object dictionary

Control code #0001

The following figure shows the IOWR message structure.

IOWR message structure

word n	Index object entry [hex]		range: 0000h ~ FFFFh
word n+1	Sub-Index object entry [hex]		range: 0000h ~ 00FFh
word n+2	Data length		range: 0001 ~ 0008
word n+3	Data byte 2	Data byte 1	**Note:** Data byte 1 is the least significant byte.
word n+4	Data byte 4	Data byte 3	
word n+5	Data byte 6	Data byte 5	
word n+6	Data byte 8	Data byte 7	

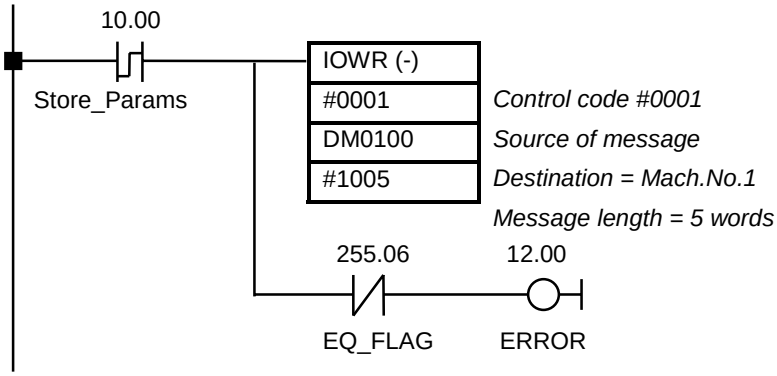
The maximum amount of data that can be written to an object entry with the IOWR instruction is 8 bytes. Data can only be written to object entries with the attribute WO (write-only), RW (read-write), RWR (read-write on input application objects), or RWW (read-write on output application objects).

If an object entry is a simple variable, the value for the Sub-Index (word n+1) should be set to zero.

Refer to the CANopen Communication Profile Specification or the EDS-file (see appendix A) for the data length of a specific object entry.

Example The following example shows how to store all configuration data in non-volatile memory (see section 3-3-2-9). In this example, the Machine number is set to ‘1’.

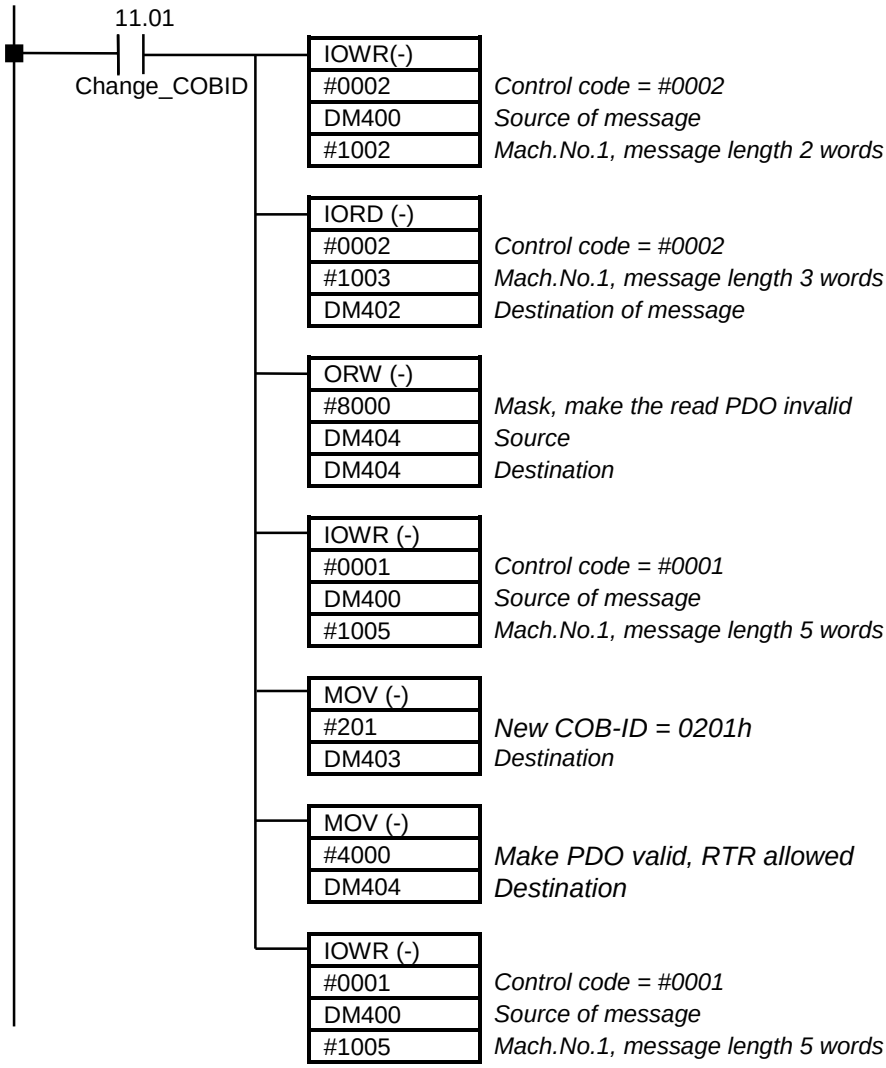
Note that the C200H syntax of the IOWR instruction is used.



PLC memory address	Content	Meaning
DM0100	1010	Index 1010h = Store Parameters
DM0101	0001	Subindex 01h = Save all
DM0102	0004	Data length = 4 bytes
DM0103	6173 = 'as'	Data = 6576 6173h = 'evas'
DM0104	6576 = 'ev'	

Example

To modify the COB-ID of a PDO:



Address	Start value	New value	Description
DM400	#1400	#1400	PDO index number
DM401	#0001	#0001	PDO sub-index
DM402	Don't care	#0004	Data length
DM403	Don't care	#0201	Data = #40000201
DM404	Don't care	#4000	

Explanation:

A PDO should be disabled (made invalid) before its COB-ID can be changed. This is done by setting the most significant bit of the COB-ID (see 3-3-2-1). The procedure is to first read the PDO's current COB-ID, then set the disable bit and write it back to the PDO. After that the PLC can set the COB-ID to the desired value, and write this to the PDO, while resetting the disable bit.

5-6 Reading from the local object dictionary

Control code #0002

Reading from the local object dictionary requires a combination of the IOWR and IORD instructions. The IOWR instruction indicates to the Unit which object entry to read. After that, the indicated object entry is read with the IORD instruction.

The reading of a local object entry must be preceded by the corresponding IOWR instruction at least once after power-on or reset. After an object entry has been indicated with the IOWR instruction, this object entry can be read as often as is needed.

If another object entry should be read, a new IOWR instruction must be executed first, to indicate the index and sub-index of that object.

Note The IORD instruction can only be executed after a successful execution of the IOWR instruction.

The following figure shows the IOWR and IORD message structure.

IOWR message structure

word n	Index object entry [hex]	range: 0000h ~ FFFFh
word n+1	Sub-Index object entry [hex]	range: 0000h ~ 00FFh

IORD message structure

word n	Data length		range: 0001 ~ 0008
word n+1	Data byte 2	Data byte 1	**Note:** Data byte 1 is the least significant byte.
word n+2	Data byte 4	Data byte 3	
word n+3	Data byte 6	Data byte 5	
word n+4	Data byte 8	Data byte 7	

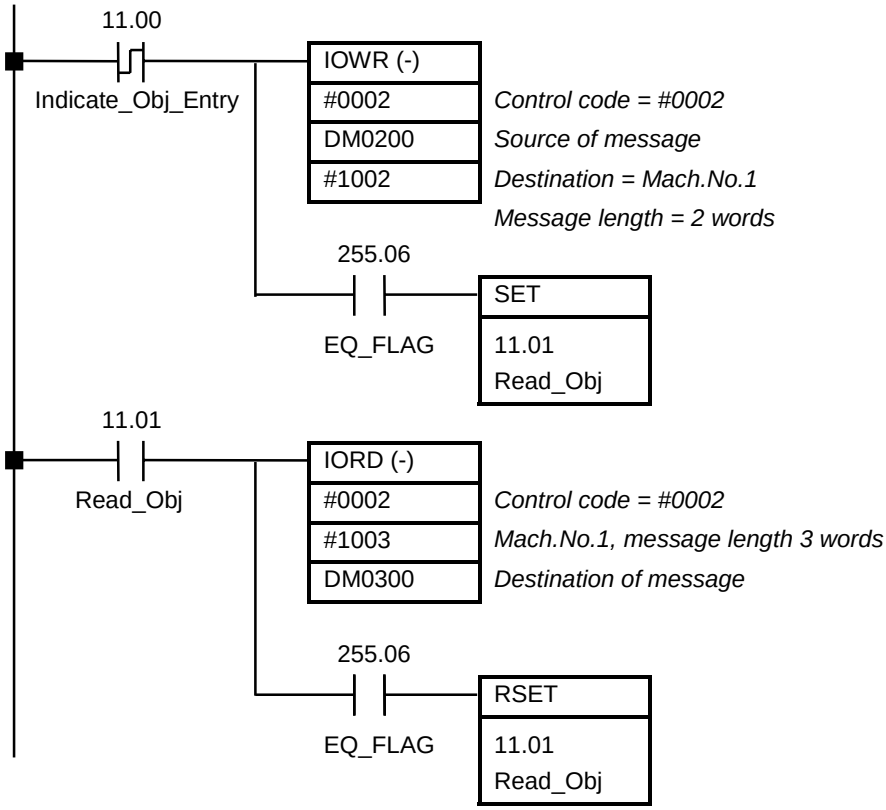
The maximum length of an object entry that can be read is 8 bytes. Data can only be read from object entries with the attribute RO (read-only), RW (read-write), RWR (read-write on input application objects), or RWW (read-write on output application objects).

If an object entry is a simple variable, the value for the Sub-Index (word n+1) should be set to 0000h.

Refer to the CANopen Communication Profile Specification or the EDS-file (see appendix A) for the data length of a specific object entry.

Example The following example shows how the COB-ID of RPDO1 (object index 1400h, sub-index 1) is read. In this example, the Mach.No. is set to '1'.

PLC memory address	Contents	Meaning
DM0200	1400	Index of object to be read
DM0201	0001	Sub-index



Note that the C200H syntax of the IOWR and IORD instruction is used.

On the rising edge of IR11.00, the Index and Sub-Index are transmitted to the Unit. After successful execution of the IOWR instruction, IR11.01 is set. When IR 11.01 is set, the indicated object entry is read and written to PLC address DM0300. If the reading of the object entry was successful, IR11.01 is reset. The result is e.g.:

PLC memory address	Contents	Meaning
DM0300	0004	Data length = 4 bytes
DM0301	0210	COB-ID of RPDO1 = 40000210h
DM0302	4000	(See Section 3-3-2-1)

In this example, RPDO1 uses identifier 210h.

If another object entry is to be read:

- Change DM0200 and DM0201
- Change the message length of the IORD instruction if the length of the object entry is different

Generate a rising edge of IR11.00

5-7 Transmitting a user defined CAN message

Control code #0003

The C200HW-CORT21-V1 allows transmission of arbitrary CAN messages with the IOWR instruction. If it is not required that the CANopen protocol is active at the same time, IR n.00 can be set to '0'.

⚠ Caution This function is only for advanced users that have knowledge of the CAN protocol. The C200HW-CORT21-V1 does not check the validity of the CAN message.

The frame of a CAN message is shown in the picture below.

*	Identifier (11 bits)	*	Data length (DLC)	Data byte 1	--------	Data byte 8	*

* Frame control bits, not important for this function

To transmit a CAN message, it is necessary to specify the identifier of the frame, the data length (number of data bytes) and the actual data (up to a maximum of 8 bytes). The figure below shows the IOWR message structure.

IOWR message structure

word n	Identifier [hex]		range: 0000h ~ 07FFh
word n+1	Data length		range: 0000 ~ 0008
word n+2	Data byte 2	Data byte 1	
word n+3	Data byte 4	Data byte 3	
word n+4	Data byte 6	Data byte 5	
word n+5	Data byte 8	Data byte 7	

Note The C200HW-CORT21-V1 ignores bits 11 to 15 of word n.

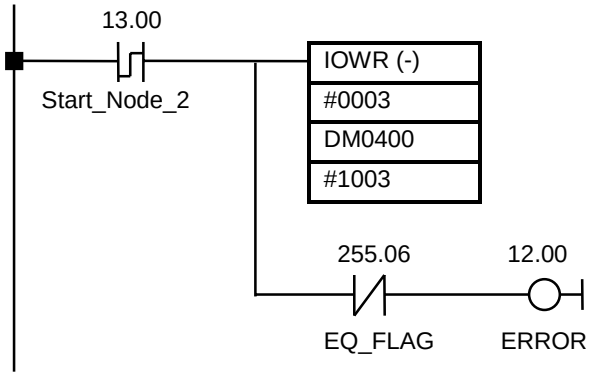
Transmission time-out

If no other node is connected to the network, or if the CAN interface of the Unit is in error, the C200HW-CORT21-V1 will not be able to transmit the message. After a time-out of 50 ms, the error code 0431 is added to the error log. If more IOWR transmission requests are issued during this time-out period, the transmission of these messages will be aborted. An error code 0431 is added to the error log for each message that was aborted.

Example The following example shows how the NMT command ‘Start_Remote_Node’ is transmitted to node 2. In this example, the Machine number is set to ‘1’. Note that the C200H syntax of the IOWR instruction is used.

All NMT commands have the identifier value 0000h. The first data byte indicates the function (‘Start_Remote_Node’ = 01h) and the second data byte the Node ID of the targeted node (Node ID = 02h).

If the Node ID is set to 00h, the command will be transmitted to all nodes.



PLC memory address	Contents	Meaning
DM0400	0000	ID = 0000h → NMT command
DM0401	0002	Data length = 2 bytes
DM0402	0201	Data byte 1 = 01h, Data byte 2 = 02h

5-8 Receiving a user defined CAN message

Control code #0004

The C200HW-CORT21-V1 can receive arbitrary CAN messages with a combination of the IOWR instruction and the I/O refresh. If it is not required that the CANopen protocol is active at the same time, IR n.00 can be set to ‘0’.

The frame of a CAN message is shown in the picture below.

*	Identifier (11 bits)	*	Data length (DLC)	Data byte 1	--------	Data byte 8	*

* Frame control bits, not important for this function

The IOWR instruction is used to instruct the Unit:

- to receive messages with specific identifier(s)
- to transfer the received message to a specific PLC area.

The figure below shows the IOWR message structure.

IOWR message structure

word n	Minimum identifier [hex]	range: 0000h ~ 07FFh
word n+1	Maximum identifier [hex]	range: 0000h ~ 07FFh
word n+2	Identifier mask [hex]	range: 0000h ~ 07FFh
word n+3	PLC start address	

Identifier filter

Word n to word n+2 tell the C200HW-CORT21-V1 which messages to receive, based on the identifier of the message. Word n and word n+1 indicate an identifier range.

To further narrow down the number of identifiers within this range, word n+2 indicates which bits in word n (the minimum identifier) must be matched and which are don't cares.

bit value '0' = don't care
bit value '1' = must match

The use of word n to word n+2 is explained with a few examples.

Example

1. Configuring C200HW-CORT21-V1 to receive one specific identifier, e.g. 0200h.

   Set the minimum and maximum identifier to the same value and set the identifier mask to any value, e.g. 'all bits must match':

   word n = 0200, word n+1 = 0200, word n+2 = 07FF

2. Configuring C200HW-CORT21-V1 to receive a range of identifiers, e.g. 010Ah to 0114h.

   Set the minimum and maximum identifier value to cover the range and set the identifier mask to 'all bits don't care':

   word n = 010A, word n+1 = 0114, word n+2 = 0000

3. Configuring C200HW-CORT21-V1 to receive specific identifiers within a range, e.g. receive all odd identifiers within the range 010Ah to 0114h.

   Set the minimum identifier to the first odd identifier in the range, the maximum identifier to the maximum value and the identifier mask to match the first bit of the first (odd) identifier:

   word n = 010B, word n+1 = 0114, word n+2 = 0001

Note The C200HW-CORT21-V1 ignores bits 11 to 15 of word n ~ word n+2 of the IOWR message structure. Also, the Unit does not check the validity of word n ~ word n+2 e.g. when the identifier value of word n is greater than word n+1, no messages will be received.

PLC start address

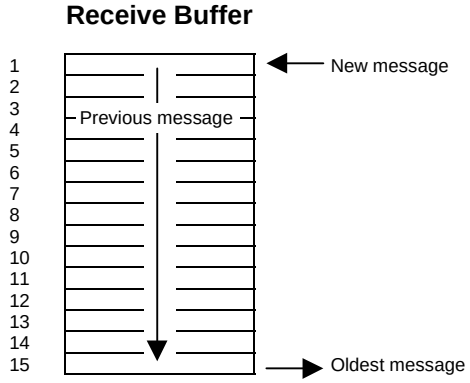
Word n+3 in the IOWR message structure indicates the start address of the PLC memory area where the received CAN messages must be transferred during the I/O refresh. The table below shows the definition of the PLC start address.

	PLC start address		
Word n+3	0000 ~ 5900	DM0000 ~ DM5900	(C200HE: 0000 ~ 4000)
	A000 ~ A140	IR000 ~ IR140	
	A300 ~ A420	IR300 ~ IR420	

Note When an incorrect PLC start address is set, error code 0440 is added to the error log and received messages are no longer transferred to the PLC.

Message buffer

The received message(s) are first stored in a buffer in the Unit. Up to 15 messages can be stored. The last received message is always stored at the top of the buffer and if there are already 15 messages stored, the oldest message is shifted out of the buffer.



New messages

During the I/O refresh, the messages stored since the last I/O refresh are transferred to the PLC and IR n+3.03 is set to indicate that new user defined CAN messages have been received. If no messages matching the identifier filter have been received since the last I/O refresh, no data is transferred to the PLC and IR n+3.03 is set to '0'.

The figure below shows the structure of the receive buffer. The representation in the PLC memory after transferring the buffer to the PLC is identical.

Receive buffer structure

word n	Number of new messages received since the last I/O refresh [BCD]		range: 0000 ~ 9999
word n+1	Identifier message 1 [hex]		range: 0000h ~ 07FFh
word n+2	Data length		range: 0000 ~ 0008
word n+3	Data byte 2	Data byte 1	
word n+4	Data byte 4	Data byte 3	
word n+5	Data byte 6	Data byte 5	
word n+6	Data byte 8	Data byte 7	
word n+7	Identifier message 2 [hex]		range: 0000h ~ 07FFh
word n+8	Data length		range: 0000 ~ 0008
word n+9	Data byte 2	Data byte 1	
word n+10	Data byte 4	Data byte 3	
word n+11	Data byte 6	Data byte 5	
word n+12	Data byte 8	Data byte 7	
⋮	⋮	⋮	
word n+85	Identifier message 15 [hex]		range: 0000 ~ 07FFh
word n+86	Data length		range: 0000 ~ 0008
word n+87	Data byte 2	Data byte 1	
word n+88	Data byte 4	Data byte 3	
word n+89	Data byte 6	Data byte 5	
word n+90	Data byte 8	Data byte 7	

Word n in the buffer structure indicates how many new messages have been received since the last time the buffer was transferred to the PLC. If the value received in the PLC memory exceeds 15, messages were shifted out of the buffer without being transferred to the PLC.

Clearing the buffer

After a reset, or after power-on, the buffer is cleared when an IOWR message with control code #0004 is successfully executed, even if the content of the IOWR is the same as a previous IOWR.

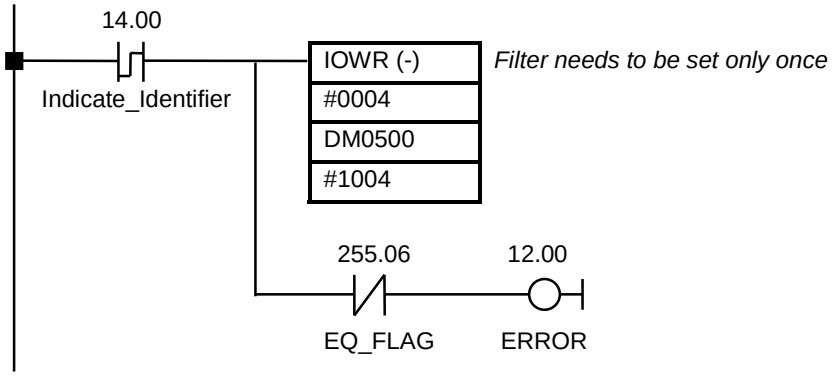
Example

The following example shows how an arbitrary CAN message is read. In this example, the Machine number is set to '1'. Note that the C200H syntax of the IOWR instruction is used.

Assume that:

- a 32-bit counter unit on the network transmits its actual counter value every time the count changes. On average, the count changes every 2 ms. The identifier of the message is 280h.
- a second node transmits its 8-bit input value every 4 ms with identifier 282h.
- a third node transmits its 16-bit clock value every 200 μs with identifier 281h.
- the scan time of the PLC is (fixed to) 8 ms.

The C200HW-CORT21-V1 should be programmed to only receive the 32-bit counter value and the 8-bit input value. The Unit should ignore the clock messages, which would overflow the buffer (40 messages per PLC scan). The received messages must be transferred to DM0600.



PLC memory address	Contents	Meaning
DM0500	0280	Minimum identifier
DM0501	0282	Maximum identifier
DM0502	0001	Identifier mask (only even ID's)
DM0503	0600	Messages to DM0600
⋮	⋮	
DM0600	0006	6 messages since last IO refresh
DM0601	0280	Message with ID = 0280
DM0602	0004	Length = 4 bytes
DM0603	1004	Data = 000F1004h
DM0604	000F	
DM0605	0000	Ignore
DM0606	0000	

PLC memory address	Contents	Meaning
DM0607	0280	Message with ID = 0280
DM0608	0004	Length = 4 bytes
DM0609	1003	Data = 000F1003h
DM0610	000F	
DM0611	0000	Ignore
DM0612	0000	
DM0613	0282	Message with ID = 0282
DM0614	0001	Length = 1 byte
DM0615	0055	Data = 55h
DM0616	0000	Ignore
DM0617	0000	
DM0618	0000	
DM0619	0280	Message with ID = 0280
DM0620	0004	Length = 4 bytes
DM0621	1002	Data = 000F1002h
DM0622	000F	
DM0623	0000	Ignore
DM0624	0000	
DM0625	0280	Message with ID = 0280
DM0626	0004	Length = 4 bytes
DM0627	1001	Data = 000F1001h
DM0628	000F	
DM0629	0000	Ignore
DM0630	0000	
DM0631	0282	Message with ID = 0282
DM0632	0001	Length = 1 byte
DM0633	00AA	Data = AAh
DM0634	0000	Ignore
DM0635	0000	
DM0636	0000	

The snapshot of the PLC memory above shows that the counter has reached the value 000F1004. Note that the most recent message is at the lowest PLC address.

6 Error Handling and Status

This section gives an overview of the implemented error handling and status indication mechanisms.

6-1 Introduction

The C200HW-CORT21-V1 uses several error detection and error handling mechanisms.

Error checking CAN

CANopen is based on the serial bus protocol of CAN (see section 1-2-1). The data link layer of the CAN protocol combines 5 error detection mechanisms (CRC check, frame check, Ack check, bit check, bit stuffing check). This combination results in a Hamming distance of 6. This means that at least 6 bits in the message frame must have been disturbed to possibly remain undetected. The overall residual error probability is extremely low and this makes CAN-based protocols very reliable and suitable for harsh environments.

Error checking CANopen

Besides the error detection implemented in the data link layer, CANopen adds some higher layer error handling mechanisms:

- Emergency message - Used to notify the status of the Unit to other nodes on the network.
- Heartbeat or Node / Life guarding protocol - Used for mutual life checking.
  See sections 1-3-4-2, 3-3-2-3, 3-3-2-4 for details.
- Error register, Predefined error field - Object entries that indicate the actual error(s) and record the occurred errors since power-on or the last reset. Another node (or the host PLC) can read these entries to retrieve the error information.

The C200HW-CORT21-V1 also has some manufacturer specific status indication mechanisms:

- Fatal PLC error handling - Mechanism to notify the network of the current status of the PLC.
- Status words - IR n+3 ~ IR n+5 indicate the current status of the Unit to the PLC.
  See section 4-3-2 for details.
- LEDs, 7-segment display - Visual status indication of the Unit.

This chapter will give a detailed description of these higher layer error handling and status mechanisms. The heartbeat and Node / Life guarding protocol, and the status words are not described in this chapter because they are already described in detail in the referred sections.

Flow diagrams

To get a better understanding of how the error handling and status mechanisms are related, this chapter concludes with some flow diagrams that describe the basic functionality of the Unit.

6-2 Emergency message

A general description of the emergency message can be found in section 1-3-3-3.

The figure below shows the meaning of each byte in the emergency message.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
EEC_{LSB}	EEC_{MSB}	ER	Comm Err	Device Err	EMCY trigger	00	00

EEC = Emergency Error Code

ER = Error register

Byte 0 and 1 form a 16-bit error code.

Byte 2 to 5 are bit-field registers.

Byte 6 and 7 are fixed to 00h.

EEC (byte 0 and 1)

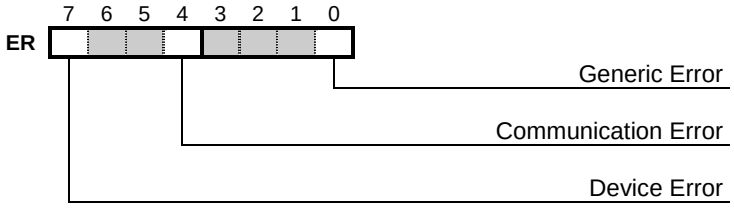
The 16-bit Emergency Error Code depends on the type of error. The table below describes the three possible error codes.

EEC (hex)	Description
0000	Error reset. At least one error has been resolved.
8100	Communication error
FF00	Device specific error

ER (byte 2)

The error register (byte 2), has the same value as the object at index 1001 (see section 6-3).

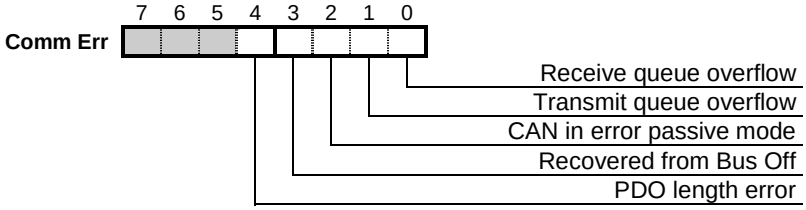
The figure below shows the error register.



The error register indicates the type of actual error(s). It can either be a device specific error or a communication error. Any type of error will also set the generic error bit (bit 0 is always set when bit 4 or bit 7 is set).

Comm Err (byte 3)

In case bit 4 in the error register is set, Comm Err (byte 3) gives more detailed information on which communication errors are present.



Comm Err.00 Receive queue overflow

This bit is set when the C200HW-CORT21-V1 is not able to process all the incoming messages that it was configured to receive. This bit is turned off as soon as the receive queue did not overflow for at least 100 ms.

0: No overflow during the last 100 ms.

1: Overflow of the receive queue.

Comm Err.01 Transmit queue overflow

This bit is set when the C200HW-CORT21-V1 is not able to transmit all TPDOs requested by the host PLC. This bit is turned off as soon as the transmit queue did not overflow for at least 100 ms.

0: No overflow during the last 100 ms.

1: Overflow of the transmit queue.

Comm Err.02 CAN in Error Passive Mode

This bit is set when the communication interface of the Unit detects a high occurrence of errors on the bus. The Unit will still communicate but will not send high-priority error messages on the bus (hence "Error Passive"), because the errors could be related to the Unit itself.

0: No high occurrence of errors on the bus or the errors have been resolved before going Bus-Off.

1: The communication interface has gone in Error Passive Mode due to a high occurrence of errors on the bus.

Comm Err.03 Recovered from Bus Off

If the high error rate on the bus remains after going to Error Passive Mode, the interface will go to "Bus Off" state, and refrain from further communication.

At that moment, the Unit is unable to indicate to the network that it has stopped communicating. After communication resumes, it can indicate that it has recovered from Bus Off.

The Bus Off recovery sequence can be started by setting IR n.03. This will only be successful when the problem that caused Bus Off has been resolved.

0: No recent Bus Off recovery.

1: The Unit has recovered from Bus Off. This bit is negated with the next Emergency Message or at least within 100 ms after the bus off recovery.

Comm Err.04 **PDO length error**

If the C200HW-CORT21-V1 receives a PDO with fewer data bytes than configured, the Unit sends out an emergency message with this bit set.

This bit is negated as soon as a PDO is received with the correct number of data bytes or more. If a PDO is received with too many data bytes, the additional bytes are ignored.

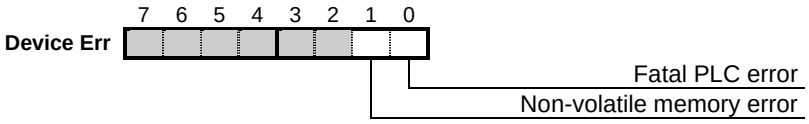
0: No PDO length error.

1: PDO length error. Too few bytes were received.

Comm Err.05 ~ Comm Err.07 **Not used** by C200HW-CORT21

Device Err (byte 4)

In case bit 7 in the error register is set, Device Err (byte 4) gives more detailed information on which device specific errors are present.



Device Err.00 **Fatal PLC error**

In normal operation, a falling edge of IR n.00, which should be set and kept ON by the host PLC program, indicates a Fatal PLC error. When this occurs, an emergency message is sent out with this bit set. This bit is negated when IR n.00 is set again.

Note that switching to/from Program mode also clears IR n.00 and also causes these emergency messages to be sent.

0: No Fatal PLC error. IR n.00 is set. CANopen is enabled.

1: Fatal PLC error. A falling edge of IR n.00 is detected. CANopen is disabled.

Device Err.01 **Non-volatile memory error**

This bit is set when the Unit is unable to store or restore the parameters to / from non-volatile memory. Refer to section 3-3-2-9 for details. This bit is negated again when the store / restore is successful after another attempt.

0: No non-volatile memory error.

1: Non-volatile memory error. Try to store the parameters again and/or restart the Unit.

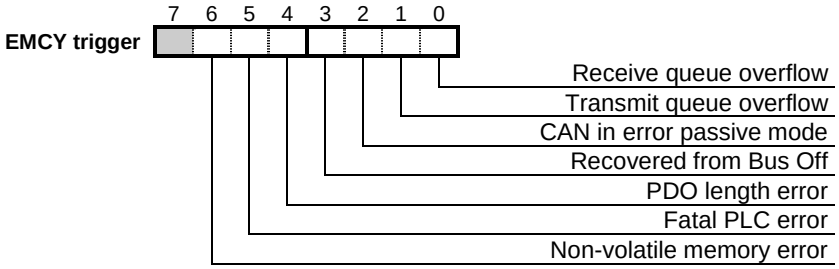
Device Err.02 ~ Device Err.07 **Not used** by C200HW-CORT21-V1

EMCY trigger (byte 5)

The EMCY trigger indicates which error has caused the emergency message to be transmitted. The reason can be that an new error has occurred (EEC ≠ 0) or that an error has been resolved (EEC = 0).

Note When the last error has been resolved, the emergency message will contain 8 zero's!

The following figure shows the bit definitions of the EMCY trigger.



6-3 Error Register, Predefined Error Field

The C200HW-CORT21-V1 has two object entries in the object dictionary that indicate the actual and previous errors since the last time the Unit was reset.

Error Register – Object entry 1001h

The error register is a simple 8-bit variable that stores the actual errors. This byte is also transmitted as part of the emergency message and a description of this byte can be found in the section that describes the emergency message (section 6-2)

Predefined Error Field – Object entry 1003h

The Predefined Error Field is an array of 32-bit variables. The figure below shows the structure of this object entry.

Index	Sub-index	Object	Default value
1003h	-	Predefined Error Field	-
1003h	00h	Number of errors	0
1003h	01h	Error code	0
1003h	02h	Error code	0
	⋮	⋮	
1003h	08h	Error code	0

Error history

The array is able to store 8 different errors to provide an error history. The format of the 32-bit error code is shown below.

byte 3	byte 2	byte 1	byte 0
Error count	EMCY trigger	Emergency Error Code	

By default, the error history is empty and sub-index 0 indicates 0 errors. As soon as an error occurs and an emergency message is sent out, the Emergency Error Code (byte 0 and 1 in the emergency message) and corresponding EMCY trigger (byte 5 in the emergency message) are copied in the error code structure at sub-index 01h. The Error count (byte 3 in the error code structure) at the same sub-index is incremented by one

There are two scenarios when the next error occurs.

- The next error is a different error, i.e. a type of error that is not yet logged in the error history:

  The previous error(s) are shifted down in the list (moved to a higher sub-index) and the current error is added at sub-index 01h.
- The next error has occurred before and is already in the error history:

  Only the Error count (byte 3) of the entry with the same Emergency Error code and EMCY trigger is incremented by one.

Note that sub-index 0 is always updated to the number of errors in the error history at the moment that a new entry is added.

Clearing the error history The error history can be cleared by writing ‘0’ to sub-index 00h.

6-4 Fatal PLC error handling

The C200HW-CORT21-V1 informs the CANopen network when a fatal PLC error has occurred. A fatal PLC error stops the execution of the ladder program and resets the complete IR area.

Bit IR n.00 has to be set to enable the CANopen communication. When a fatal PLC error occurs, IR n.00 is reset, and the Unit will transmit an emergency message:

EEC	=	FF00h (device specific)
ER	=	81h or 91h (bits 7 and 1 are set)
Comm Err	=	Unchanged
Device Err	=	01h or 03h (bit 0 is set)
EMCY trigger	=	20h

When IR n.00 is set again, another emergency message is transmitted to indicate that the error has been resolved:

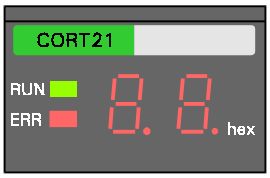
EEC	=	0000h
ER	=	Unspecified. Depends on other possible errors.
Comm Err	=	Unspecified. Depends on other possible errors.
Device Err	=	00h or 02h (bit 0 is cleared)
EMCY trigger	=	20h

Note If bit IR n.00 is set while the host PLC is in MONITOR- or RUN mode, the C200HW-CORT21-V1 will also send the emergency messages if the host PLC is switched between MONITOR / RUN mode and PROGRAM mode. Switching between these modes also clears the IR area (provided that the I/O status hold bit, SR252.12, is not set).

Refer to the Operation manual of your PLC system for more details.

6-5 LEDs, 7-Segment Display

The Unit has two LEDs and two 7-segment displays to visualise its status. The layout is shown in the picture below.



The two LEDs (RUN, ERR) indicate the status of the Unit in general.

LEDs

LED	Colour	State	Description
RUN	Green	OFF	Fatal error or no power supply via backplane
		Blinking	Initialising the hardware / software
		ON	The Unit is initialised, and no fatal errors were detected
ERR	Red	OFF	No errors
		Blinking	Non-fatal error due to incorrect Unit settings Details in IR n+4
		ON	Fatal error Details in IR n+3.13, IR n+3.14

The 7-segment display is used to indicate the status of CANopen.

7-segment display

Display		Dots	Description
State	**Value**		
OFF	-	OFF	RUN LED is OFF or Blinking or ERR LED is Blinking See description above
ON	FFh	OFF	CANopen communication disabled (IR n.00 = 0) Note: It is possible to send / receive user defined CAN messages (see section 5)
	01h ~ 7Fh		Display value = Node ID, CANopen state = Operational
	01h ~ 7Fh	ON	Display value = Node ID, CANopen state = Pre-Operational
	01h ~ 7Fh	Blinking	Display value = Node ID, CANopen state = Stopped
Blinking *	01h	OFF	Invalid node ID (node ID is set to 0) See section 3-1-6
	02h		Network power failure See section 3-1-5
	03h		Heartbeat / Life guarding error See section 3-3-2-3, 3-3-2-4
	04h		Bus Off See section 4-3-2, IR n+3.10
	05h		Non-volatile memory error See section 3-3-2-9
	06h ~ FFh		Reserved

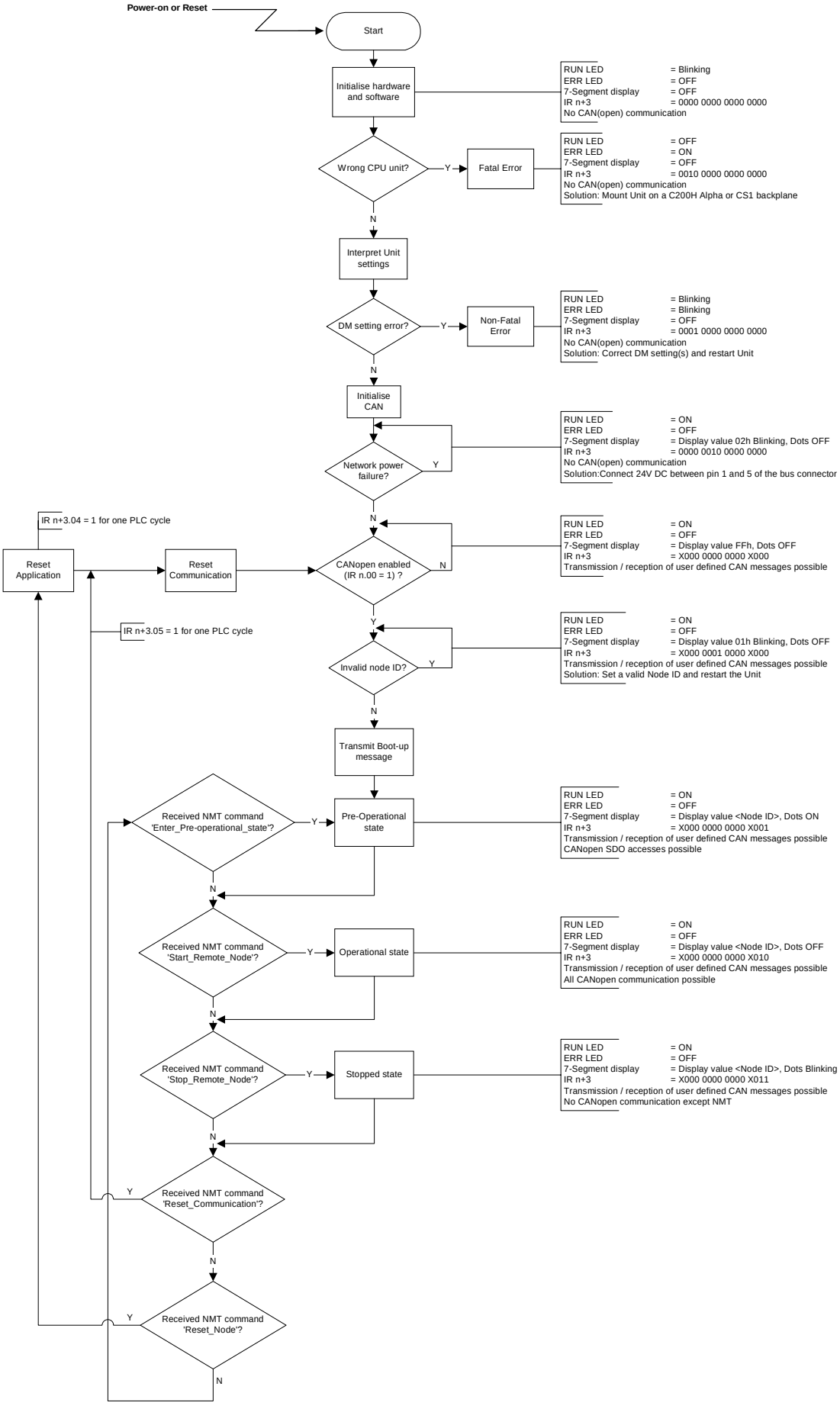
* If more than one error has occurred, the errors are displayed cyclically in ascending order with a one second interval time.

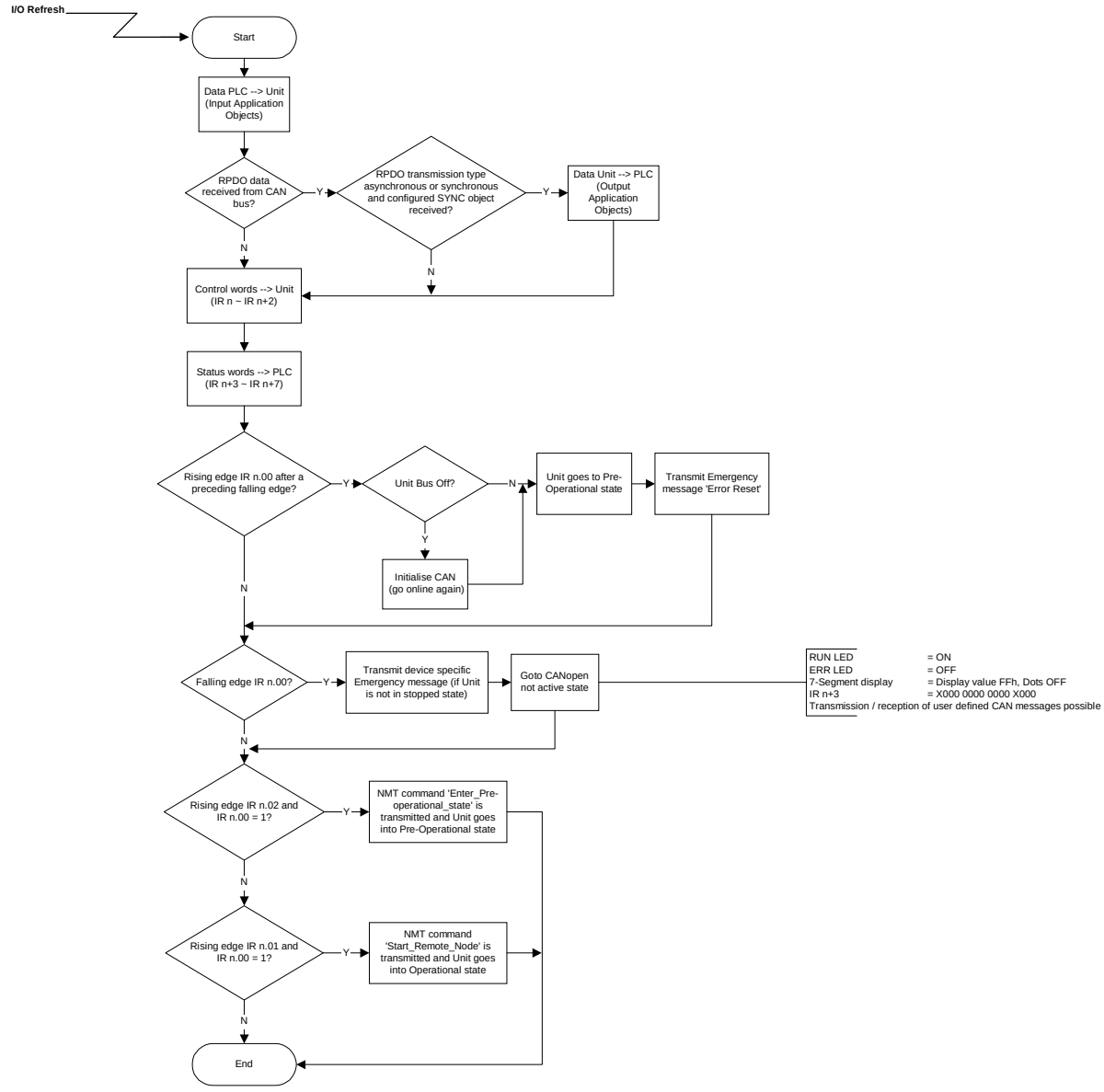
6-6 Flow diagrams

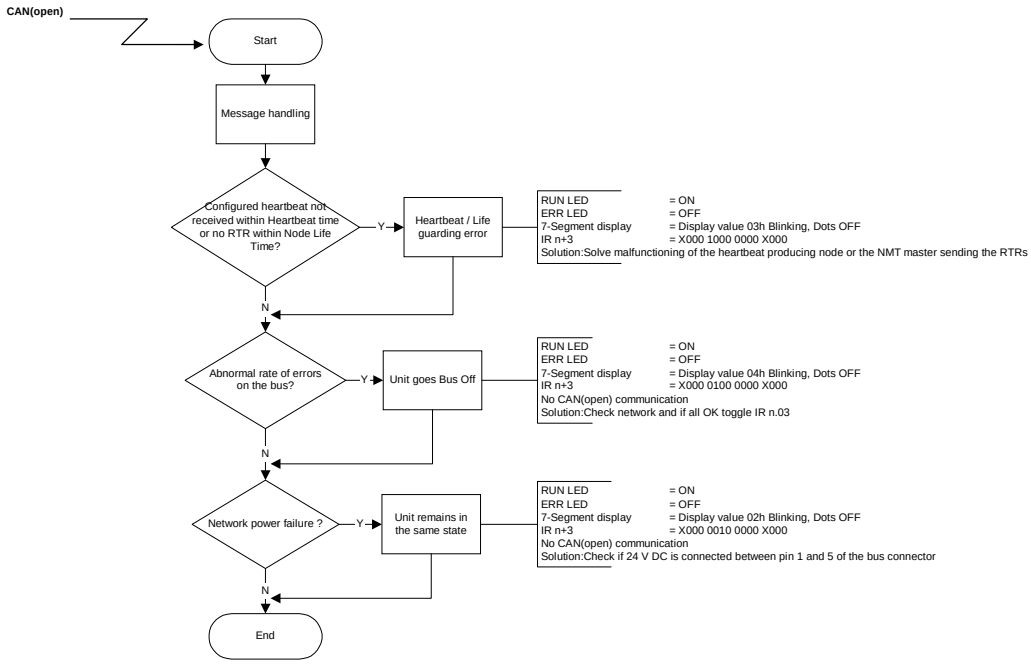
The next two pages show basic flow diagrams of the C200HW-CORT21-V1. These flow diagrams give a better understanding of the different states and state transitions, and they show the relationship of the different status and error handling mechanisms.

The first diagram describes the flow after power-on or a reset. The other two diagrams describe the interrupt driven processes: I/O refresh and CAN(open).

Note that these diagrams only describe the fundamentals of the unit's functionality.


Power-on or Reset
Start
Initialise hardware and software
RUN LED = Blinking
ERR LED = OFF
7-Segment display = OFF
IR n+3 = 0000 0000 0000 0000
No CAN(open) communication
Wrong CPU unit?
Y
Fatal Error
RUN LED = OFF
ERR LED = ON
7-Segment display = OFF
IR n+3 = 0010 0000 0000 0000
No CAN(open) communication
Solution: Mount Unit on a C200H Alpha or CS1 backplane
N
Interpret Unit settings
DM setting error?
Y
Non-Fatal Error
RUN LED = Blinking
ERR LED = Blinking
7-Segment display = OFF
IR n+3 = 0001 0000 0000 0000
No CAN(open) communication
Solution: Correct DM setting(s) and restart Unit
N
Initialise CAN
Network power failure?
Y
RUN LED = ON
ERR LED = OFF
7-Segment display = Display value 02h Blinking, Dots OFF
IR n+3 = 0000 0010 0000 0000
No CAN(open) communication
Solution:Connect 24V DC between pin 1 and 5 of the bus connector
N
IR n+3.04 = 1 for one PLC cycle
Reset Application
Reset Communication
CANopen enabled (IR n.00 = 1) ?
N
RUN LED = ON
ERR LED = OFF
7-Segment display = Display value FFh, Dots OFF
IR n+3 = X000 0000 0000 X000
Transmission / reception of user defined CAN messages possible
Y
IR n+3.05 = 1 for one PLC cycle
Invalid node ID?
Y
RUN LED = ON
ERR LED = OFF
7-Segment display = Display value 01h Blinking, Dots OFF
IR n+3 = X000 0001 0000 X000
Transmission / reception of user defined CAN messages possible
Solution: Set a valid Node ID and restart the Unit
N
Transmit Boot-up message
Received NMT command 'Enter_Pre-operational_state'?
Y
Pre-Operational state
RUN LED = ON
ERR LED = OFF
7-Segment display = Display value <Node ID>, Dots ON
IR n+3 = X000 0000 0000 X001
Transmission / reception of user defined CAN messages possible
CANopen SDO accesses possible
N
Received NMT command 'Start_Remote_Node'?
Y
Operational state
RUN LED = ON
ERR LED = OFF
7-Segment display = Display value <Node ID>, Dots OFF
IR n+3 = X000 0000 0000 X010
Transmission / reception of user defined CAN messages possible
All CANopen communication possible
N
Received NMT command 'Stop_Remote_Node'?
Y
Stopped state
RUN LED = ON
ERR LED = OFF
7-Segment display = Display value <Node ID>, Dots Blinking
IR n+3 = X000 0000 0000 X011
Transmission / reception of user defined CAN messages possible
No CANopen communication except NMT
N
Y
Received NMT command 'Reset_Communication'?
N
Y
Received NMT command 'Reset_Node'?
N


I/O Refresh
Start
Data PLC --> Unit (Input Application Objects)
RPDO data received from CAN bus?
Y
RPDO transmission type asynchronous or synchronous and configured SYNC object received?
Y
Data Unit --> PLC (Output Application Objects)
N
N
Control words --> Unit (IR n ~ IR n+2)
Status words --> PLC (IR n+3 ~ IR n+7)
Rising edge IR n.00 after a preceding falling edge?
Y
Unit Bus Off?
N
Unit goes to Pre-Operational state
Transmit Emergency message 'Error Reset'
Y
Initialise CAN (go online again)
N
Falling edge IR n.00?
Y
Transmit device specific Emergency message (if Unit is not in stopped state)
Goto CANopen not active state
RUN LED = ON
ERR LED = OFF
7-Segment display = Display value FFh, Dots OFF
IR n+3 = X000 0000 0000 X000
Transmission / reception of user defined CAN messages possible
N
Rising edge IR n.02 and IR n.00 = 1?
Y
NMT command 'Enter_Pre-operational_state' is transmitted and Unit goes into Pre-Operational state
N
Rising edge IR n.01 and IR n.00 = 1?
Y
NMT command 'Start_Remote_Node' is transmitted and Unit goes into Operational state
N
End


CAN(open)
Start
Message handling
Configured heartbeat not received within Heartbeat time or no RTR within Node Life Time?
Y
Heartbeat / Life guarding error
RUN LED = ON
ERR LED = OFF
7-Segment display = Display value 03h Blinking, Dots OFF
IR n+3 = X000 1000 0000 X000
Solution:Solve malfunctioning of the heartbeat producing node or the NMT master sending the RTRs
N
Abnormal rate of errors on the bus?
Y
Unit goes Bus Off
RUN LED = ON
ERR LED = OFF
7-Segment display = Display value 04h Blinking, Dots OFF
IR n+3 = X000 0100 0000 X000
No CAN(open) communication
Solution:Check network and if all OK toggle IR n.03
N
Network power failure ?
Y
Unit remains in the same state
RUN LED = ON
ERR LED = OFF
7-Segment display = Display value 02h Blinking, Dots OFF
IR n+3 = X000 0010 0000 X000
No CAN(open) communication
Solution:Check if 24 V DC is connected between pin 1 and 5 of the bus connector
N
End

7 Troubleshooting and Maintenance

This section describes the troubleshooting procedures and maintenance operations needed to keep the CANopen network operating properly.

7-1 Error Indicators

Refer to section 6 for a detailed description of the error indicators of the C200HW-CORT21-V1.

7-2 Troubleshooting

Possible problems have been divided into the following categories:

- PLC Errors
- Start-up problems
- Configuration problems
- I/O data communication problems
- Message communication problems

General Note: when replacing the C200HW-CORT21-V1, make sure that any CANopen configuration data that was stored in the Unit's non-volatile memory is written to the new Unit before putting the system in operation.

PLC Errors

Description	Possible cause	Possible remedy
An I/O verification error occurred.	The current PLC configuration is not the same as it was when the I/O table was registered.	Check the I/O table with the I/O table verification operation and correct it if necessary. After correction, perform the I/O Table Create operation.
An I/O Unit Over error occurred.	The MACHINE No. rotary switch setting is incorrect.	Make sure that the setting does not exceed the maximum allowed Unit number. See section 3-1-3. If the Machine number exceeds the PLC's limit, adjust the Machine number and restart the Unit. Make sure that the PLC program and DM settings are adapted to the new Machine number.
	Two Units claim the same Machine number.	Adjust the MACHINE No. switch setting and restart the Unit. Make sure that the PLC program and DM settings are adapted to the new Machine number.
A Special I/O Unit error occurred.	The Unit is not connected properly or was removed while the power was on.	Turn the power off, make sure that the Unit is connected properly and turn the power on again.
An I/O Bus error occurred.	The Unit is not connected properly.	Turn the power off, make sure that the Unit is connected properly and turn the power on again.
	The Unit is defective.	Replace the C200HW-CORT21-V1.

Start-up problems

Description	Possible cause	Possible remedy
No LEDs are ON or Blinking.	The PLC's power is off.	Turn the PLC's power supply on.
	The Unit is defective.	Replace the C200HW-CORT21-V1.
The RUN LED is Blinking.	The same Unit number has been set on another Special I/O Unit, causing an I/O UNIT OVER error in the PLC.	Make sure that the same Unit number is not used by more than one Special I/O Unit and restart the PLC.
	I/O table verification error. A Unit has been removed or the Unit number setting has been changed causing an I/O SET ERR error in the PLC.	Verify the I/O table and/or register the I/O table again.
	Another Special I/O Unit or Interrupt Input Unit has not been initialised, causing a CPU WAIT'G error in the PLC.	Check and/or remove the defective Unit.
	The Unit is defective.	Replace the C200HW-CORT21-V1.
ERR LED is Blinking.	A non-fatal error due to incorrect Unit settings. The cause is reported in IR n+4.	Try to solve the cause and restart the Unit. If this does not help, replace the Unit.
ERR LED is ON.	A fatal error was detected at start-up. The cause is reported in IR n+3.13, IR n+3.14	Try to solve the cause and restart the Unit. If this does not help, replace the Unit.

Configuration problems

Description	Possible cause	Possible remedy
Not able to access (read / write) an object entry.	The object entry does not exist or has the wrong attribute.	Check the EDS-file (Appendix A) if the object entry exists. If the object exists, check the attribute of the object. To read an object, the attribute has to be RO, RW, RWW, or RWR. To write to an object, the attribute has to be WO, RW, RWW, or RWR.
	The CANopen state of the Unit is 'Stopped' or 'not active'. IR n+3.01, IR n+3.00 = 00 (not active) or IR n+3.01, IR n+3.00 = 11 (Stopped)	If 'not active', wait until the Unit is initialised, set IR n.00, and/or start the Bus Off recovery sequence by toggling IR n.03. If 'Stopped', send the NMT command 'Enter_Pre-operational_State' or 'Start_Remote_Node'.
	The SDO access has the wrong syntax.	Check the CANopen communication profile specification for the correct syntax of an SDO access.
	The network cabling is not correct.	Check if the correct pins of the bus connector are connected (see section 3-1-5), if there are no short circuits, if the stub-lines are not too long, and if the network is terminated at the appropriate places (see section 3-3-1).
	The Unit operates at a different baud rate than the configuration tool.	Configure both to communicate on the same baud rate.
	The Unit is defective.	Replace the C200HW-CORT21-V1.

Configuration problems (continued)

Description	Possible cause	Possible remedy
Not able to store / restore parameters.	Incorrect signature.	To store parameters, the signature 'save' has to be written to sub-index 01h of index 1010h. To restore parameters, the signature 'load' has to be written to sub-index 01h of index 1011h. See section 3-3-2-9 for details.
	Non-volatile memory error. When the Unit detects a non-volatile memory error, an emergency message is transmitted (see section 6-2) and error code 05h will be blinking on the 7-segment display.	Try to store the configuration again and / or restart the Unit. Note that storing the configuration takes approx. four seconds; the Unit should not be reset in this period. If the error keeps occurring, replace the Unit.

I/O data communication problems

Description	Possible cause	Possible remedy
7-segment display value is Blinking.	The 7-segment display indicates the cause. Refer to section 6-5 for a list with error codes.	Try to solve the cause and/or restart the Unit. If this does not help, replace the Unit.
7-segment display dots are ON.	The Unit is in 'Pre-Operational' state. Also indicated by IR n+3.00 and IR n+3.01.	Send the NMT command 'Start_Remote_Node' to the Unit or set IR n.01. The Unit should go to Operational state and the dots should go OFF.
7-segment display dots are Blinking.	The Unit is in 'Stopped' state. Also indicated by IR n+3.00 and IR n+3.01.	Send the NMT command 'Start_Remote_Node' to the Unit or set IR n.01. The Unit should go to Operational state and the dots should go OFF.
No I/O data is exchanged with the PLC although the RUN LED is ON. 7-segment display value is not Blinking and the dots are OFF.	The network cabling is not correct.	Check if the correct pins of the bus connector are connected (see section 3-1-5), if there are no short circuits, if the stub-lines are not too long, and if the network is terminated at the appropriate places (see section 3-3-1).
	The Unit operates on a different baud rate than the other units on the network.	Configure all the units on the network to the same baud rate.
	The CANopen communication is disabled. IR n.00 is not set. Display value is FFh.	Set IR n.00 ON.
	Application object(s) not mapped to a PDO.	Configure the desired application object(s) to a PDO and configure the communication parameters of that PDO (see section 3-3-2-1 and section 3-3-2-2).
	PDO is not enabled.	The highest bit of the COB-ID (sub-index 01h of the PDO Communication Parameters) determines whether the PDO is valid or not. Normally, the configuration tool enables the PDO (by clearing this bit) when the application object mapped to the respective PDO is linked to another Unit. Check if this bit is cleared and if not, clear it manually.
	Wrong transmission type	Set the correct transmission type. See section 3-3-2-1 for the possible transmission modes.
	Application object(s) not mapped to a PLC address. If an application object is not mapped to a PLC address, it might appear that the Unit is not communicating.	Configure the Unit settings to map the desired application object(s) to a PLC address.

I/O data communication problems (continued)

Description	Possible cause	Possible remedy
	Another Special I/O Unit makes use of the same data area(s), e.g. C200HW-PRM21, -DRM21, -PRT21, -DRT21. When areas overlap, one Unit overwrites the data area of the other, depending on their respective Machine Numbers.	Check the mapping of the C200HW-CORT21-V1 and the other Special I/O Unit. If an overlap exists, the Unit settings of one of the units should be changed. Adapt the PLC program accordingly.
	The Unit is defective.	Replace the C200HW-CORT21-V1.
Data in the mapped PLC area is being reset.	Data that is mapped in the IR- or LR area is reset when • The Host PLC is switched between RUN / MONITOR mode and PROGRAM mode. • A fatal PLC error occurs (see section 6-4)	If the Host PLC was not switched to / from PROGRAM mode, check if a fatal PLC error has occurred and try to solve the cause of the error. Refer to the Operation Manual of the PLC for details.
	The Unit is defective.	Replace the C200HW-CORT21-V1.
Not all CANopen PDOs sent by other nodes arrive in the host PLC.	Other nodes update their SDO data more frequently than the I/O refresh rate of the host PLC. If the same SDO is received twice or more per PLC cycle, only the last data will be transferred to the PLC.	Reduce the host PLC's cycle time (faster CPU type, move control tasks and I/O to another PLC), or Reduce the update rate of the transmitting node to match the PLC cycle time.
	Other nodes transmit more SDOs linked to the unit, than can be processed by the unit's CAN interface. The CAN interface transmits an Emergency message (receive queue overflow).	Reduce the number of output data objects and RPDOs. Reduce the update rate of the transmitting nodes.
Not all CANopen PDOs sent by the Unit arrive at other nodes.	The Unit transmits all data messages, but the processing at the receiving node takes longer than the Unit's I/O refresh cycle	Increase the PLC scan time, or only perform I/O refreshes with the Unit when necessary (by IOWR command).
	The host PLC produces more (changes of state in) data objects than can be transmitted by the CAN interface. The CAN interface transmits an Emergency message (transmit queue overflow).	Reduce the number of input data objects on TPDOs. Increase the PLC scan time
ERR LED is switched ON.	An application error was detected. IR n+3.14 is set.	Restart the Unit. If this does not help, replace the Unit.

Message communication problems

Description	Possible cause	Possible remedy
IOWR instruction not executed successfully.	The IOWR instruction contains syntax errors. (EQ-flag is not set)	Check if the correct Machine number and message length is specified in the IOWR instruction.
	The control code is invalid or the specified message length does not correspond with the function indicated by the control code. (EQ-flag is not set). The contents of the message is invalid.	Read the error log to find the cause. Check if the control code lies within the range #0001 to #0004. Check the message length. Check the contents of the message.
	The Unit is defective.	Replace the C200HW-CORT21-V1.

Message communication problems (continued)

Description	Possible cause	Possible remedy
IORD instruction not executed successfully.	The IORD instruction contains syntax errors. (EQ-flag is not set)	Check if the correct Machine number and message length is specified in the IORD instruction.
	The control code is invalid or the specified message length does not correspond with the function indicated by the control code or When the local object dictionary is being read, the object entry to read was never indicated since power-on or the last reset. (EQ-flag is not set).	Read the error log to find the cause. Check if the control code equals #0000 or #0002. Check the message length. In case of reading the local object dictionary, check if the object entry to read has been indicated by an IOWR.
	The Unit is defective.	Replace the C200HW-CORT21-V1.
Not able to access (read / write) an object entry.	The object entry does not exist or has the wrong attribute.	Read the error log to find the cause. Check the EDS-file (Appendix A) to see if the object entry exists. If the object exists, check the attribute of the object. To read an object, the attribute must be RO, RW, RWW, or RWR. To write to an object, the attribute must be WO, RW, RWW, or RWR.
	The specified data length in the IOWR message structure does not correspond with the data length of the object entry.	Check the data length of the object entry
User defined CAN message is not received by another node.	An incorrect identifier or data length was set in the IOWR message structure.	Compare the identifier of the receiving node with the one specified in the IOWR message structure. Check if the data length corresponds with the number of data bytes that the receiving node expects.
	The network cabling is not correct.	Check the error log. If the error log indicates that the Unit was unable to transmit the message, then check if the correct pins of the bus connector are connected (see section 3-1-5), if there are no short circuits, if the stub-lines are not too long, and if the network is terminated at the appropriate places (see section 3-3-1).
	The receiving node has a 'low-speed' (fault-tolerant') CAN interface.	'Low-speed' CAN devices use a different physical layer from the C200HW-CORT21-V1's 'high-speed' CAN. Converters between 'high-speed' and 'low-speed' CAN are commercially available.
	The Unit operates on a different baud rate than the other node on the network.	Configure all the units on the network to the same baud rate.

Message communication problems (continued)

No user defined CAN messages are received.	The identifier filter was not set or was incorrect.	Check the values of the identifier filter (word n ~ word n+2 of the IOWR message structure) and execute an IOWR instruction that sets the identifier filter.
	An incorrect PLC start address in the IOWR message structure was indicated.	Check the PLC start address (word n+3 in the IOWR message structure).
	The network cabling is not correct.	Check if the correct pins of the bus connector are connected (see section 3-1-5), if there are no short circuits, if the stub-lines are not too long, and if the network is terminated at the appropriate places (see section 3-3-1).
	The transmitting node has a 'low-speed' (fault-tolerant') CAN interface.	'Low-speed' CAN devices use a different physical layer from the C200HW-CORT21-V1's 'high-speed' CAN. Converters between 'high-speed' and 'low-speed' CAN are commercially available.
	The Unit operates on a different baud rate than the other node on the network.	Configure all the units on the network to the same baud rate.
Not all user defined CAN messages are received	The C200HW-CORT21-V1 can buffer up to 15 messages between PLC I/O refreshes. If the receive buffer is not transferred to the PLC memory before the buffer overflows, messages will be lost.	Reduce the data production rate of the transmitting node. Reduce the PLC scan time or change to a lower baud rate (receive buffer is filled at a lower rate).
ERR LED is switched ON.	An application error was detected. IR n+3.14 is set.	Restart the Unit. If this does not help, replace the Unit.

7-3 Maintenance

This section describes the routine cleaning and inspection recommended as regular maintenance.

7-3-1 Cleaning

Clean the CANopen Slave Units regularly as described below in order to keep it in optimal operating condition.

- Wipe the Unit with a dry, soft cloth for regular cleaning.
- When a spot cannot be removed with a dry cloth, dampen the cloth with a neutral cleanser, wring out the cloth, and wipe the Unit.
- A smudge may remain on the Unit from gum, vinyl, or tape that was left on for a long time. Remove the smudge when cleaning.

⚠ **Caution** Never use volatile solvents such as paint thinner or benzene or chemical wipes. These substances could damage the surface of the Unit.

7-3-2 Inspection

Be sure to inspect the system periodically to keep it in its optimal operating condition. In general, inspect the system once every 6 to 12 months, but inspect more frequently if the system is used with high temperature or humidity or under dirty / dusty conditions.

Inspection Equipment

Prepare the following equipment before inspecting the system.

Required Equipment

Have a standard and Philips-head screwdriver, multimeter, alcohol, and a clean cloth.

Equipment that could be needed

Depending on the system conditions, a synchroscope, oscilloscope, thermometer, or hygrometer (to measure humidity) might be needed.

Inspection Procedure

Check the items in the following table and correct any items that are below standard.

Item		Standard	Equipment
Environmental conditions	Ambient temperature	0 to 55°C	Thermometer
	Ambient humidity	10 to 90%	Hygrometer
	Dust/dirt accumulation	None	---
Installation	Are the Units installed securely?	No looseness	---
	Are the communications connectors fully inserted?	No looseness	---
	Are the external wiring screws tight?	No looseness	---
	Are the connecting cables undamaged?	No damage	---

7-3-3 Replacement of Units

A defective Unit may have an effect on the entire network communication performance. Therefore it is important to immediately replace a defective Unit in the network. It is recommended to have spare Units available to restore network operation as quickly as possible.

Precautions

Observe the following precautions when replacing a defective Unit:

- After replacement, make sure that there are no errors with the new Unit.
- When a Unit is being returned for repair, attach a sheet of paper detailing the problem and return the Unit to your OMRON dealer.
- If there is a faulty contact, try wiping the contact with a clean, lint-free cloth dampened with alcohol.

After replacing a Unit, set the new Unit's switches to the same settings as the old Unit. If the Unit requires configuration data other than the predefined connection set, store the settings in its non-volatile memory.

Hot swapping

The CANopen network allows connection and disconnection of nodes while in operation. The node that is connected to the network will not be operational until it has received the NMT command 'Start_Remote_Node'. Adding a new node to an existing configuration will most likely require configuring of that node and the node(s) that it needs to communicate with.

Note Do not plug or unplug the C200HW-CORT21-V1 on the PLC backplane while the PLC is powered. Doing so may result in damage to the Unit and/or the PLC system.

Appendix A
EDS-file C200HW-CORT21-V1

```
[FileInfo]
CreatedBy=OMRON Europe B.V. - ADDC FA Communications Group
ModifiedBy=OMRON Europe B.V. - ADDC FA Communications Group
Description=EDS file for C200HW-CORT21-V1
CreationTime=04:50PM
CreationDate=01-29-2001
ModificationTime=03:45PM
ModificationDate=07-10-2001
FileName=cort21v1.eds
FileVersion=1
FileRevision=1
EDSVersion=1

[DeviceInfo]
VendorName=OMRON Europe B.V.
VendorNumber=0x00000083
ProductName=CANopen Slave for C200H Alpha
ProductNumber=1
RevisionNumber=1
OrderCode=C200HW-CORT21-V1
BaudRate_10=1
BaudRate_20=1
BaudRate_50=1
BaudRate_125=1
BaudRate_250=1
BaudRate_500=1
BaudRate_800=1
BaudRate_1000=1
SimpleBootUpMaster=0
SimpleBootUpSlave=1
Granularity=8
DynamicChannelsSupported=3
CompactPDO=3
GroupMessaging=0
NrOfRXPDO=64
NrOfTXPDO=64
LSS_Supported=0

[DummyUsage]
Dummy0001=0
Dummy0002=0
Dummy0003=0
Dummy0004=0
Dummy0005=1
Dummy0006=1
Dummy0007=1

[Comments]
Lines=8
Line1=EDS file for C200HW-CORT21-V1
Line2=
Line3=Default four 8-bit TX/RX objects
Line4=Possible network variables:
Line5=100 TX/RX 08-bit objects
Line6=100 TX/RX 16-bit objects
Line7=50  TX/RX  32-bit objects
Line8=25  TX/RX  64-bit objects

[MandatoryObjects]
SupportedObjects=3
1=0x1000
2=0x1001
3=0x1018

[1000]
ParameterName=Device Type
ObjectType=0x7
DataType=0x0007
AccessType=ro
DefaultValue=0
PDOMapping=0

[1001]
ParameterName=Error Register
ObjectType=0x7
DataType=0x0005
AccessType=ro
DefaultValue=0
PDOMapping=0

[1018]
ParameterName=Identity Object
ObjectType=0x9
SubNumber=2

[1018sub0]
ParameterName=number of entries
ObjectType=0x7
DataType=0x0005
AccessType=ro
DefaultValue=1
PDOMapping=0

[1018sub1]
ParameterName=Vendor Id
ObjectType=0x7
DataType=0x0007
AccessType=ro
DefaultValue=0x00000083
PDOMapping=0

[OptionalObjects]
SupportedObjects=30
1=0x1003
2=0x1005
3=0x1008
4=0x1009
5=0x100A
6=0x100B
7=0x100C
8=0x100D
9=0x1010
10=0x1011
11=0x1014
12=0x1016
13=0x1017
14=0x1200
15=0x1400
16=0x1401
17=0x1402
18=0x1403
19=0x1600
20=0x1601
21=0x1602
22=0x1603
23=0x1800
24=0x1801
25=0x1802
26=0x1803
27=0x1A00
28=0x1A01
29=0x1A02
30=0x1A03

[1003]
ParameterName=Pre-defined Error Field
ObjectType=0x8
SubNumber=9

[1003sub0]
ParameterName=Number of Errors
ObjectType=0x7
DataType=0x0005
AccessType=rw
DefaultValue=0
PDOMapping=0
LowLimit=0
HighLimit=0
```

[1003sub1]
ParameterName=Standard Error Field
ObjectType=0x7
DataType=0x0007
AccessType=ro
DefaultValue=0
PDOMapping=0

[1003sub2]
ParameterName=Standard Error Field
ObjectType=0x7
DataType=0x0007
AccessType=ro
DefaultValue=0
PDOMapping=0

[1003sub3]
ParameterName=Standard Error Field
ObjectType=0x7
DataType=0x0007
AccessType=ro
DefaultValue=0
PDOMapping=0

[1003sub4]
ParameterName=Standard Error Field
ObjectType=0x7
DataType=0x0007
AccessType=ro
DefaultValue=0
PDOMapping=0

[1003sub5]
ParameterName=Standard Error Field
ObjectType=0x7
DataType=0x0007
AccessType=ro
DefaultValue=0
PDOMapping=0

[1003sub6]
ParameterName=Standard Error Field
ObjectType=0x7
DataType=0x0007
AccessType=ro
DefaultValue=0
PDOMapping=0

[1003sub7]
ParameterName=Standard Error Field
ObjectType=0x7
DataType=0x0007
AccessType=ro
DefaultValue=0
PDOMapping=0

[1003sub8]
ParameterName=Standard Error Field
ObjectType=0x7
DataType=0x0007
AccessType=ro
DefaultValue=0
PDOMapping=0

[1005]
ParameterName=COB-ID SYNC
ObjectType=0x7
DataType=0x0007
AccessType=rw
DefaultValue=0x00000080
PDOMapping=0
LowLimit=0x00000001
HighLimit=0x000007FF

[1008]
ParameterName=Manufacturer Device Name
ObjectType=0x7
DataType=0x0009
AccessType=const
DefaultValue=C200HW-CORT21-V1
PDOMapping=0

[1009]
ParameterName=Manufacturer Hardware Version
ObjectType=0x7
DataType=0x0009
AccessType=const
DefaultValue=01.00
PDOMapping=0

[100a]
ParameterName=Manufacturer Software Version
ObjectType=0x7
DataType=0x0009
AccessType=const
DefaultValue=01.00
PDOMapping=0

[100b]
ParameterName=Node-ID
ObjectType=0x7
DataType=0x0007
AccessType=ro
DefaultValue=$NODEID
PDOMapping=0

[100c]
ParameterName=Guard Time
ObjectType=0x7
DataType=0x0006
AccessType=rw
DefaultValue=0x0000
PDOMapping=0

[100d]
ParameterName=Life Time Factor
ObjectType=0x7
DataType=0x0005
AccessType=rw
DefaultValue=0x00
PDOMapping=0

[1010]
ParameterName=Store Parameter Field
ObjectType=0x8
SubNumber=2

[1010sub0]
ParameterName=Number of Entries
ObjectType=0x7
DataType=0x0005
AccessType=ro
DefaultValue=1
PDOMapping=0

[1010sub1]
ParameterName=Save all Parameters
ObjectType=0x7
DataType=0x0007
AccessType=rw
PDOMapping=0

[1011]
ParameterName=Restore Default Parameters
ObjectType=0x8
SubNumber=2

[1011sub0]
ParameterName=Number of Entries
ObjectType=0x7
DataType=0x0005
AccessType=ro
DefaultValue=0x01
PDOMapping=0

[1011sub1]
ParameterName=Restore all Default Parameters
ObjectType=0x7
DataType=0x0007
AccessType=rw
PDOMapping=0

[1014]
ParameterName=COB-ID EMCY
ObjectType=0x7
DataType=0x0007
AccessType=ro
DefaultValue=$NODEID+0x80
PDOMapping=0

```
[1016]
ParameterName=Consumer Heartbeat
              Time
ObjectType=0x8
SubNumber=2

[1016sub0]
ParameterName=Number of Entries
ObjectType=0x7
DataType=0x0005
AccessType=ro
DefaultValue=1
PDOMapping=0

[1016sub1]
ParameterName=Consumer Heartbeat
              Time
ObjectType=0x7
DataType=0x0007
AccessType=rw
DefaultValue=0x00000000
PDOMapping=0
LowLimit=0x00000000
HighLimit=0x007FFFFF

[1017]
ParameterName=Producer Heartbeat
              Time
ObjectType=0x7
DataType=0x0006
AccessType=rw
DefaultValue=0x0000
PDOMapping=0

[1200]
ParameterName=Server SDO Parameter
ObjectType=0x9
SubNumber=3

[1200sub0]
ParameterName=Number of Entries
ObjectType=0x7
DataType=0x0005
AccessType=ro
DefaultValue=0x02
PDOMapping=0

[1200sub1]
ParameterName=COB-ID Client -> Server
ObjectType=0x7
DataType=0x0007
AccessType=ro
DefaultValue=$NODEID+0x600
PDOMapping=0

[1200sub2]
ParameterName=COB-ID Server -> Client
ObjectType=0x7
DataType=0x0007
AccessType=ro
DefaultValue=$NODEID+0x580
PDOMapping=0

[1400]
ParameterName=RPDO #0
              Communication
              Parameter
ObjectType=0x9
SubNumber=3

[1400sub0]
ParameterName=Number of Entries
ObjectType=0x7
DataType=0x0005
AccessType=ro
DefaultValue=2
PDOMapping=0

[1400sub1]
ParameterName=COB-ID
ObjectType=0x7
DataType=0x0007
AccessType=rw
DefaultValue=$NODEID+0x40000200
PDOMapping=0
LowLimit=0x40000001
HighLimit=0xC00007FF

[1400sub2]
ParameterName=Transmission Type
ObjectType=0x7
DataType=0x0005
AccessType=rw
DefaultValue=0xFE
PDOMapping=0
LowLimit=0x00
HighLimit=0xFE

[1401]
ParameterName=RPDO #1
              Communication
              Parameter
ObjectType=0x9
SubNumber=3

[1401sub0]
ParameterName=Number of Entries
ObjectType=0x7
DataType=0x0005
AccessType=ro
DefaultValue=2
PDOMapping=0

[1401sub1]
ParameterName=COB-ID
ObjectType=0x7
DataType=0x0007
AccessType=rw
DefaultValue=$NODEID+0x40000300
PDOMapping=0
LowLimit=0x40000001
HighLimit=0xC00007FF

[1401sub2]
ParameterName=Transmission Type
ObjectType=0x7
DataType=0x0005
AccessType=rw
DefaultValue=0xFE
PDOMapping=0
LowLimit=0x00
HighLimit=0xFE

[1402]
ParameterName=RPDO #2
              Communication
              Parameter
ObjectType=0x9
SubNumber=3

[1402sub0]
ParameterName=Number of Entries
ObjectType=0x7
DataType=0x0005
AccessType=ro
DefaultValue=2
PDOMapping=0

[1402sub1]
ParameterName=COB-ID
ObjectType=0x7
DataType=0x0007
AccessType=rw
DefaultValue=$NODEID+0x40000400
PDOMapping=0
LowLimit=0x40000001
HighLimit=0xC00007FF

[1402sub2]
ParameterName=Transmission Type
ObjectType=0x7
DataType=0x0005
AccessType=rw
DefaultValue=0xFE
PDOMapping=0
LowLimit=0x00
HighLimit=0xFE

[1403]
ParameterName=RPDO #3
              Communication
              Parameter
ObjectType=0x9
SubNumber=3

[1403sub0]
ParameterName=Number of Entries
```

ObjectType=0x7
DataType=0x0005
AccessType=ro
DefaultValue=2
PDOMapping=0

[1403sub1]
ParameterName=COB-ID
ObjectType=0x7
DataType=0x0007
AccessType=rw
DefaultValue=$NODEID+0x40000500
PDOMapping=0
LowLimit=0x40000001
HighLimit=0xC00007FF

[1403sub2]
ParameterName=Transmission Type
ObjectType=0x7
DataType=0x0005
AccessType=rw
DefaultValue=0xFE
PDOMapping=0
LowLimit=0x00
HighLimit=0xFE

[1600]
ParameterName=RPDO #0 Mapping Parameter
ObjectType=0x9
SubNumber=2

[1600sub0]
ParameterName=Number of Entries
ObjectType=0x7
DataType=0x0005
AccessType=rw
DefaultValue=1
PDOMapping=0
LowLimit=0
HighLimit=8

[1600sub1]
ParameterName=Mapping Entry #1
ObjectType=0x7
DataType=0x0007
AccessType=rw
DefaultValue=0x21000108
PDOMapping=0

[1601]
ParameterName=RPDO #1 Mapping Parameter
ObjectType=0x9
SubNumber=2

[1601sub0]
ParameterName=Number of Entries
ObjectType=0x7
DataType=0x0005
AccessType=rw
DefaultValue=1
PDOMapping=0
LowLimit=0
HighLimit=8

[1601sub1]
ParameterName=Mapping Entry #1
ObjectType=0x7
DataType=0x0007
AccessType=rw
DefaultValue=0x21000208
PDOMapping=0

[1602]
ParameterName=RPDO #2 Mapping Parameter
ObjectType=0x9
SubNumber=2

[1602sub0]
ParameterName=Number of Entries
ObjectType=0x7
DataType=0x0005
AccessType=rw
DefaultValue=1
PDOMapping=0
LowLimit=0
HighLimit=8

[1602sub1]
ParameterName=Mapping Entry #1
ObjectType=0x7
DataType=0x0007
AccessType=rw
DefaultValue=0x21000308
PDOMapping=0

[1603]
ParameterName=RPDO #3 Mapping Parameter
ObjectType=0x9
SubNumber=2

[1603sub0]
ParameterName=Number of Entries
ObjectType=0x7
DataType=0x0005
AccessType=rw
DefaultValue=1
PDOMapping=0
LowLimit=0
HighLimit=8

[1603sub1]
ParameterName=Mapping Entry #1
ObjectType=0x7
DataType=0x0007
AccessType=rw
DefaultValue=0x21000408
PDOMapping=0

[1800]
ParameterName=TPDO #0 Communication Parameter
ObjectType=0x9
SubNumber=3

[1800sub0]
ParameterName=Number of Entries
ObjectType=0x7
DataType=0x0005
AccessType=ro
DefaultValue=2
PDOMapping=0

[1800sub1]
ParameterName=COB-ID
ObjectType=0x7
DataType=0x0007
AccessType=rw
DefaultValue=$NODEID+0x40000180
PDOMapping=0
LowLimit=0x40000001
HighLimit=0xC00007FF

[1800sub2]
ParameterName=Transmission Type
ObjectType=0x7
DataType=0x0005
AccessType=rw
DefaultValue=0xFE
PDOMapping=0
LowLimit=0x00
HighLimit=0xFE

[1801]
ParameterName=TPDO #1 Communication Parameter
ObjectType=0x9
SubNumber=3

[1801sub0]
ParameterName=Number of Entries
ObjectType=0x7
DataType=0x0005
AccessType=ro
DefaultValue=2
PDOMapping=0

[1801sub1]
ParameterName=COB-ID
ObjectType=0x7
DataType=0x0007
AccessType=rw

```
DefaultValue=$NODEID+0x40000280
PDOMapping=0
LowLimit=0x40000001
HighLimit=0xC00007FF

[1801sub2]
ParameterName=Transmission Type
ObjectType=0x7
DataType=0x0005
AccessType=rw
DefaultValue=0xFE
PDOMapping=0
LowLimit=0x00
HighLimit=0xFE

[1802]
ParameterName=TPDO #2
              Communication
              Parameter
ObjectType=0x9
SubNumber=3

[1802sub0]
ParameterName=Number of Entries
ObjectType=0x7
DataType=0x0005
AccessType=ro
DefaultValue=2
PDOMapping=0

[1802sub1]
ParameterName=COB-ID
ObjectType=0x7
DataType=0x0007
AccessType=rw
DefaultValue=$NODEID+0x40000380
PDOMapping=0
LowLimit=0x40000001
HighLimit=0xC00007FF

[1802sub2]
ParameterName=Transmission Type
ObjectType=0x7
DataType=0x0005
AccessType=rw
DefaultValue=0xFE
PDOMapping=0
LowLimit=0x00
HighLimit=0xFE

[1803]
ParameterName=TPDO #3
              Communication
              Parameter
ObjectType=0x9
SubNumber=3

[1803sub0]
ParameterName=Number of Entries
ObjectType=0x7
DataType=0x0005
AccessType=ro
DefaultValue=2
PDOMapping=0

[1803sub1]
ParameterName=COB-ID
ObjectType=0x7
DataType=0x0007
AccessType=rw
DefaultValue=$NODEID+0x40000480
PDOMapping=0
LowLimit=0x40000001
HighLimit=0xC00007FF

[1803sub2]
ParameterName=Transmission Type
ObjectType=0x7
DataType=0x0005
AccessType=rw
DefaultValue=0xFE
PDOMapping=0
LowLimit=0x00
HighLimit=0xFE

[1a00]
ParameterName=TPDO #0 Mapping
              Parameter
ObjectType=0x9
SubNumber=2

[1a00sub0]
ParameterName=Number of Entries
ObjectType=0x7
DataType=0x0005
AccessType=rw
DefaultValue=1
PDOMapping=0
LowLimit=0
HighLimit=8

[1a00sub1]
ParameterName=Mapping Entry #1
ObjectType=0x7
DataType=0x0007
AccessType=rw
DefaultValue=0x20000108
PDOMapping=0

[1a01]
ParameterName=TPDO #1 Mapping
              Parameter
ObjectType=0x9
SubNumber=2

[1a01sub0]
ParameterName=Number of Entries
ObjectType=0x7
DataType=0x0005
AccessType=rw
DefaultValue=1
PDOMapping=0
LowLimit=0
HighLimit=8

[1a01sub1]
ParameterName=Mapping Entry #1
ObjectType=0x7
DataType=0x0007
AccessType=rw
DefaultValue=0x20000208
PDOMapping=0

[1a02]
ParameterName=TPDO #2 Mapping
              Parameter
ObjectType=0x9
SubNumber=2

[1a02sub0]
ParameterName=Number of Entries
ObjectType=0x7
DataType=0x0005
AccessType=rw
DefaultValue=1
PDOMapping=0
LowLimit=0
HighLimit=8

[1a02sub1]
ParameterName=Mapping Entry #1
ObjectType=0x7
DataType=0x0007
AccessType=rw
DefaultValue=0x20000308
PDOMapping=0

[1a03]
ParameterName=TPDO #3 Mapping
              Parameter
ObjectType=0x9
SubNumber=2

[1a03sub0]
ParameterName=Number of Entries
ObjectType=0x7
DataType=0x0005
AccessType=rw
DefaultValue=1
PDOMapping=0
LowLimit=0
HighLimit=8

[1a03sub1]
ParameterName=Mapping Entry #1
ObjectType=0x7
DataType=0x0007
```

```
AccessType=rw
DefaultValue=0x20000408
PDOMapping=0

[ManufacturerObjects]
SupportedObjects=2
1=0x2000
2=0x2100

[2000]
ParameterName=Default_Input_Bytes
ObjectType=0x8
SubNumber=5

[2000sub0]
ParameterName=Number of elements
ObjectType=0x7
DataType=0x0005
AccessType=ro
DefaultValue=4
PDOMapping=0

[2000sub1]
ParameterName=Default_Input_Byte1
ObjectType=0x7
DataType=0x0005
AccessType=ro
DefaultValue=0x00
PDOMapping=1

[2000sub2]
ParameterName=Default_Input_Byte2
ObjectType=0x7
DataType=0x0005
AccessType=ro
DefaultValue=0x00
PDOMapping=1

[2000sub3]
ParameterName=Default_Input_Byte3
ObjectType=0x7
DataType=0x0005
AccessType=ro
DefaultValue=0x00
PDOMapping=1

[2000sub4]
ParameterName=Default_Input_Byte4
ObjectType=0x7
DataType=0x0005
AccessType=ro
DefaultValue=0x00
PDOMapping=1

[2100]
ParameterName=Default_Output_Bytes
ObjectType=0x8
SubNumber=5

[2100sub0]
ParameterName=Number of elements
ObjectType=0x7
DataType=0x0005
AccessType=ro
DefaultValue=4
PDOMapping=0

[2100sub1]
ParameterName=Default_Output_Byte1
ObjectType=0x7
DataType=0x0005
AccessType=rww
DefaultValue=0x00
PDOMapping=1

[2100sub2]
ParameterName=Default_Output_Byte2
ObjectType=0x7
DataType=0x0005
AccessType=rww
DefaultValue=0x00
PDOMapping=1

[2100sub3]
ParameterName=Default_Output_Byte3
ObjectType=0x7
DataType=0x0005
AccessType=rww
DefaultValue=0x00
PDOMapping=1

[2100sub4]
ParameterName=Default_Output_Byte4
ObjectType=0x7
DataType=0x0005
AccessType=rww
DefaultValue=0x00
PDOMapping=1

[DynamicChannels]
NrOfSeg=8
Type1=0x0005
Dir1=ro
Range1=0x2001-0x2001
PPOffset1=0
MaxCnt1=100
Type2=0x0006
Dir2=ro
Range2=0x2002-0x2002
PPOffset2=0
MaxCnt2=100
Type3=0x0007
Dir3=ro
Range3=0x2003-0x2003
PPOffset3=0
MaxCnt3=50
Type4=0x001B
Dir4=ro
Range4=0x2004-0x2004
PPOffset4=0
MaxCnt4=25
Type5=0x0005
Dir5=rww
Range5=0x2101-0x2101
PPOffset5=0
MaxCnt5=100
Type6=0x0006
Dir6=rww
Range6=0x2102-0x2102
PPOffset6=0
MaxCnt6=100
Type7=0x0007
Dir7=rww
Range7=0x2103-0x2103
PPOffset7=0
MaxCnt7=50
Type8=0x001B
Dir8=rww
Range8=0x2104-0x2104
PPOffset8=0
MaxCnt8=25
```

Appendix B
Stored Parameters

Index	Sub index	Name	Default Value According Profile
1005h	-	COB-ID SYNC Message	0000 0080h
1016h	1	Consumer Heartbeat Time	0000 0000h
1017h	-	Producer Heartbeat Time	0000h
1400h	1	COB-ID RPDO1	0000 0200h + Node ID
	2	Transmission Type RPDO1	FEh
1401h	1	COB-ID RPDO2	0000 0300h + Node ID
	2	Transmission Type RPDO2	FEh
1402h	1	COB-ID RPDO3	0000 0400h + Node ID
	2	Transmission Type RPDO3	FEh
1403h	1	COB-ID RPDO4	0000 0500h + Node ID
	2	Transmission Type RPDO4	FEh
1404h ~ 143Fh	1	COB-ID RPDO5 ~ RPDO64	8000 0781h
	2	Transmission Type RPDO5 ~ RPDO64	FEh
1600h	1	1st mapped object RPDO1	2100 0108h
	2 ~ 8	2nd ~ 8th mapped object RPDO1	0000 0000h
1601h	1	1st mapped object RPDO2	2100 0208h
	2 ~ 8	2nd ~ 8th mapped object RPDO2	0000 0000h
1602h	1	1st mapped object RPDO3	2100 0308h
	2 ~ 8	2nd ~ 8th mapped object RPDO3	0000 0000h
1603h	1	1st mapped object RPDO4	2100 0408h
	2 ~ 8	2nd ~ 8th mapped object RPDO4	0000 0000h
1604h ~ 163Fh	1 ~ 8	1st ~ 8th mapped object RPDO5 ~ RPDO64	0000 0000h
1800h	1	COB-ID TPDO1	0000 0180h + Node ID
	2	Transmission Type TPDO1	FEh
1801h	1	COB-ID TPDO2	0000 0280h + Node ID
	2	Transmission Type TPDO2	FEh
1802h	1	COB-ID TPDO3	0000 0380h + Node ID
	2	Transmission Type TPDO3	FEh
1803h	1	COB-ID TPDO4	0000 0480h + Node ID
	2	Transmission Type TPDO4	FEh
1804h ~ 183Fh	1	COB-ID TPDO5 ~ TPDO64	8000 0780h
	2	Transmission Type TPDO5 ~ TPDO64	FEh

Index	Sub index	Name	Default Value According Profile
1A00h	1	1st mapped object TPDO1	2000 0108h
	2 ~ 8	2nd ~ 8th mapped object TPDO1	0000 0000h
1A01h	1	1st mapped object TPDO2	2000 0208h
	2 ~ 8	2nd ~ 8th mapped object TPDO2	0000 0000h
1A02h	1	1st mapped object TPDO3	2000 0308h
	2 ~ 8	2nd ~ 8th mapped object TPDO3	0000 0000h
1A03h	1	1st mapped object TPDO4	2000 0408h
	2 ~ 8	2nd ~ 8th mapped object TPDO4	0000 0000h
1A04h ~ 1A3Fh	1 ~ 8	1st ~ 8th mapped object TPDO5 ~ TPDO64	0000 0000h

Appendix C
CS1 PLC series compatibility

The C200HW-CORT21-V1 can also be installed in OMRON CS1 PLC systems, which provide a C200H-compatible I/O bus. However, the internal memory organisation in the CS1 PLC differs from that of the C200H series. The following table shows the relation between the C200H-series memory addresses used throughout this manual, and the corresponding addresses in the CS1 PLC series.

Function		**instead of C200H addresses**	**CS1 will use addresses**
Control & Status area	(MACHINE No.= 0 to 9)	IR100 ~ IR199	CIO2000 ~ CIO2159
	(MACHINE No.= A to F)	IR400 ~ IR459	
Unit settings		DM1000 ~ DM2599 or DM7000 ~ DM8599	D20000 ~ D21599
I/O data mapping areas		IR000 ~ IR235	CIO0000 ~ CIO0235
		IR300 ~ IR511	CIO0300 ~ CIO0511
		HR00 ~ HR99	HR000 ~ HR099
		LR00 ~ LR63	CIO1000 ~ CIO1063
		DM0000 ~ DM5999	D00000 ~ D05999
Error flag	(MACHINE No.= 0 to 9)	AR00.00 ~ AR00.09	AR418.00 ~ AR418.15
	(MACHINE No.= 0 to F)	IR280.00 ~ IR280.15	
Restart flag	(MACHINE No.= 0 to 9)	AR01.00 ~ AR01.09	AR502.00 ~ AR502.15
	(MACHINE No.= 0 to F)	IR281.00 ~ IR281.15	

Appendix D
Baud rate configuration via Unit settings

The standard CANopen baud rates are selectable via the DIP-switches at the front of the Unit (see section 3-1-4). However, if a different baud rate is required, it is possible to configure this via the Unit settings. The Unit settings only overrule the DIP-switch settings when all the configured values are within the specified ranges. By default, these Unit settings are zero and therefore the DIP-switches determine the baud rate.

The Unit settings are made in a dedicated DM area. Refer to section 4-2-3 for the relationship between the Machine Number and the location of the dedicated DM area.

The DM settings directly configure the hardware registers of the CAN interface that determine the baud rate. Incorrect settings will cause improper functioning of the CAN interface. Users with little CAN experience are advised to contact their distributor to assist in setting the correct values.

The table below lists the DM words for configuration of the baud rate.

DM word	Value	Meaning
m+16	0000 ~ 0063	BRP value The Baud Rate Prescaler value determines the length of the time quanta within one bit time.
m+17	0000 ~ 0003	SJW value The (Re)Synchronisation Jump Width defines the maximum number of time quanta a bit time may be shortened or lengthened by one resynchronisation.
m+18	0002 ~ 0015	TSEG1 value TSEG1 is the time segment within a bit time before the sample point.
m+19	0001 ~ 0007	TSEG2 value TSEG2 is the time segment within a bit time after the sample point.

$$\text{Baud rate} = \frac{8\,000}{(\text{BRP} + 1) \times (3 + \text{TSEG1} + \text{TSEG2})} \text{ [kbit/s]}$$

Recommended: $\text{TSEG2} \geq \text{SJW}$

$\text{TSEG1} > \text{TSEG2}$

$\text{TSEG1} + \text{TSEG2} \geq 5$

The following table lists the values of BRP, SJW, TSEG1, and TSEG2 for the standard CANopen baud rates that can be selected with the DIP-switches.

Baud rate (kbit/s)	BRP	SJW	TSEG1	TSEG2
10	0049	0000	0012	0001
20	0024	0000	0012	0001
50	0009	0000	0012	0001
125	0003	0000	0012	0001
250	0001	0000	0012	0001
500	0000	0000	0012	0001
800	0000	0000	0006	0001
1000	0000	0000	0004	0001

Index

I (continued)

L

M

N

O

P

R

S

T

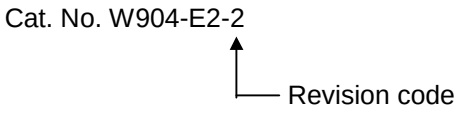
U

V-W

0-9

Revision History

A manual revision code appears as a suffix to the catalog number on the front cover of the manual.



The following table outlines the changes made to the manual during each revision. Page numbers refer to the previous version.

Revision code	Date	Revised content
1	July 2000	Original production
2	June 2001	CANopen functionality added to existing user defined CAN communication.

Regional Headquarters
OMRON EUROPE B.V.
Wegalaan 67-69,
NL-2132 JD Hoofddorp
The Netherlands
Tel: +31 (0)23 5681-300
Fax: +31 (0)23 5681-388

OMRON

Authorized Distributor:

Cat. No. W904-E2-2

Note: Specifications subject to change without notice.

Printed in the Netherlands