CANopen training

CANopen

▷ Section 1 : Basic knowledge

▷ Section 2 : Application layer

▷ Section 3 : Link layer

▷ Section 4 : Physical layer

▷ Section 5 : Schneider offer overview

▷ Section 6 : Diagnostics

CANopen

Section 1: Basic knowledge

▶ Part 1: History

▶ Part 2: Main characteristics

1980: The automotive industry indicates a need for low-cost, real-time, highly reliable communication between various on-board electronic subassemblies.

1983: In partnership with several German universities, the German equipment manufacturer Robert BOSCH develops the specifications for the communication protocol **CAN = Controller Area Network**.

1985: The first CAN integrated circuits are marketed by **INTEL.**

1986: Initial prototypes presented to the USA at Detroit.

1987: The **major silicon manufacturers**: Philips, Motorola, National Semiconductors, Texas Instruments, MHS, SIEMENS offer a complete range of drivers and micro-controllers which integrate CAN.

1989: First **industrial applications**.

1987 to 1991: Establishment of **promotional organizations** for both automotive (SAE, OSEK), and industrial (CIA = CAN in Automation) applications.

1991: CAN Low Speed (125 kbps/Identifiers coded on 11 bits) becomes standard **ISO 11519-2**.

1992: MERCEDES uses a 500 kbps CAN bus to link 5 electronic subassemblies on an S class.

1993: CAN High Speed (1 Mbps/Identifiers on 11 bits) becomes standard ISO 11898 = **CAN 2.0 A**.

1995: Amendment to standard ISO 11898 introducing identifiers coded on 29 bits = **CAN 2.0 B.**

Publication by the **CiA** of communication profile **DS-301**.

1996: CAN is used on the majority of motor controls in **top-of-the-range European cars.**

1997: **300 companies** are members of the **CiA**.

1998: New set of ISO standards for **diagnostics** and **compliance testing**.

Reference specifications

CANopen was built chronologically on the basis of several specifications:

- **CAN 2.0 A and B** (originating from Robert BOSCH)

  Defines precisely the link layer and part of the physical layer

- **CAL** = CAN Application Layer (CiA):

  Provides tools for developing an application using CAN without instructions for use + further details on the physical layer

- **CANopen** (CiA):

  Defines which CAL tools to use and how.

  Ensures interoperability of products by profile descriptions.

Physical Layer

Medium:	**Shielded twisted pair** **2 or 4-wire (if power supply)**
Topology:	**Bus type** **With short tap links and end-of-line resistor**
Maximum distance:	**1000 m**
Data rate:	**1 Mbps at 25 metres, 10 Kbps at 1000 metres** **Depends on the length of the bus and the type of cable**
Max. no. of devices:	**128** **1 master and 127 slaves**

Link layer

Network access method: **CSMA/CA**

Each device can send data as soon as the bus is free.

In the event of a collision, a principle of dominant or recessive bits can arbitrate bit by bit in a non-destructive manner.

The priority of a message is given by the value of the identifier: **the identifier with the lowest value has the highest priority.**

Communication model: **Producer/Consumer**

An identifier located at the start of the message informs receivers as to the type of data contained in each message, each receiver decides whether or not to consume the data.

Max. useful data size: **8 bytes per frame**

Link layer

Security of transmission:

Among the best on industrial local area networks.

Numerous signalling and error detection devices ensure excellent security of transmission.

CANopen application layer

A CAL subassembly which defines standardized communication objects and 4 types of service:

1 . **Network administration**: parameter setting, starting, monitoring (master-slaves)

2 . Transmission of **small amounts of process data** (<= 8 bytes) in real time: **PDO** = Process Data Object (producer-consumer)

3 . Transmission of large amounts of **parameter-setting data** (> 8 bytes) using segmentation without time constraints: **SDO** = Service Data Object (client-server)

4 . Predefined messages for managing **synchronizations, time-based references, fatal errors: SFO** = Special Function Object

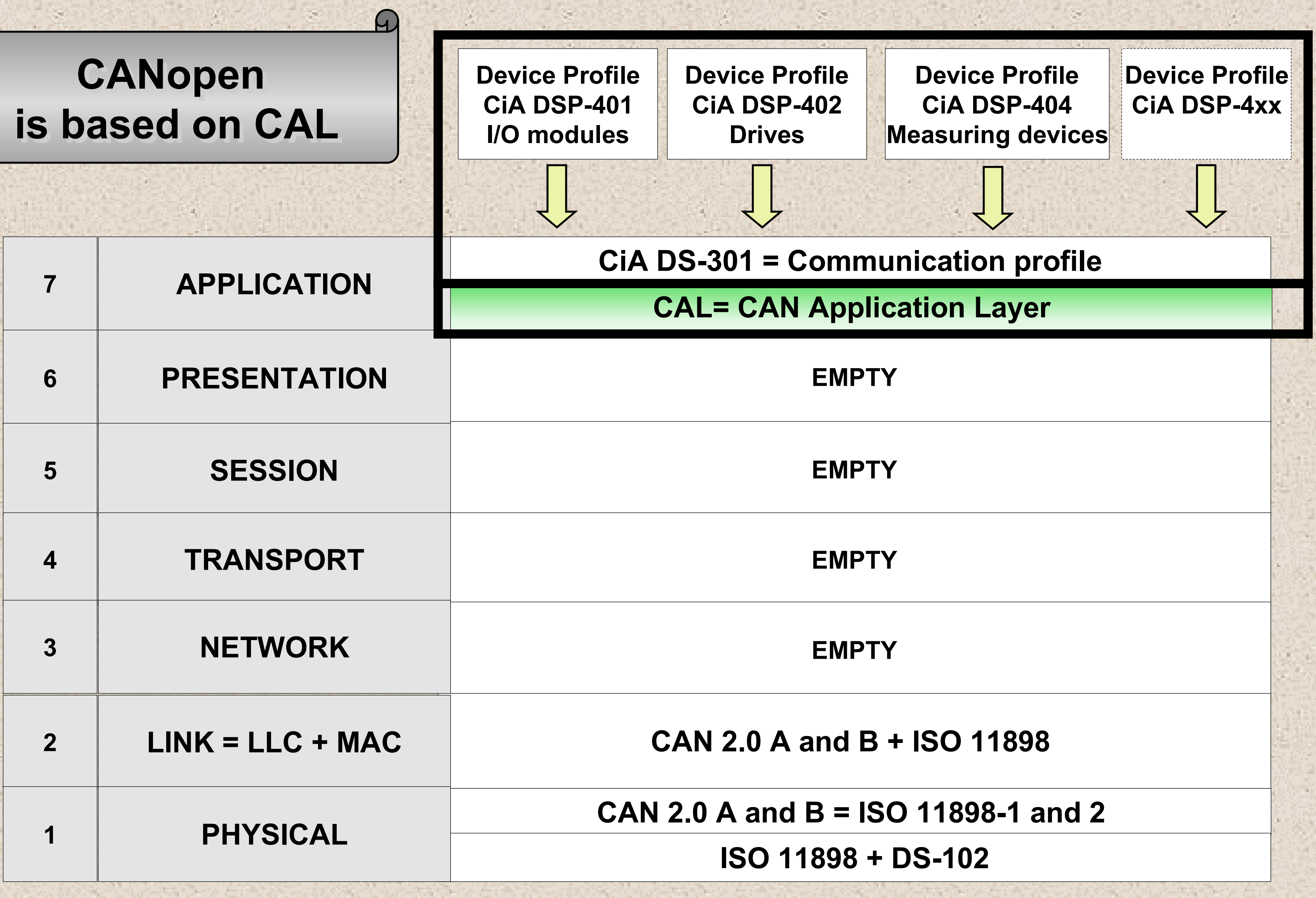
CANopen

Section 2: Application layer

▷ Part 1: CANopen basic concepts

▷ Part 2: CANopen objects and services

CANopen is based on CAL



Device Profile CiA DSP-401 I/O modules

Device Profile CiA DSP-402 Drives

Device Profile CiA DSP-404 Measuring devices

Device Profile CiA DSP-4xx

7	APPLICATION	CiA DS-301 = Communication profile CAL= CAN Application Layer
6	PRESENTATION	EMPTY
5	SESSION	EMPTY
4	TRANSPORT	EMPTY
3	NETWORK	EMPTY
2	LINK = LLC + MAC	CAN 2.0 A and B + ISO 11898
1	PHYSICAL	CAN 2.0 A and B = ISO 11898-1 and 2 ISO 11898 + DS-102

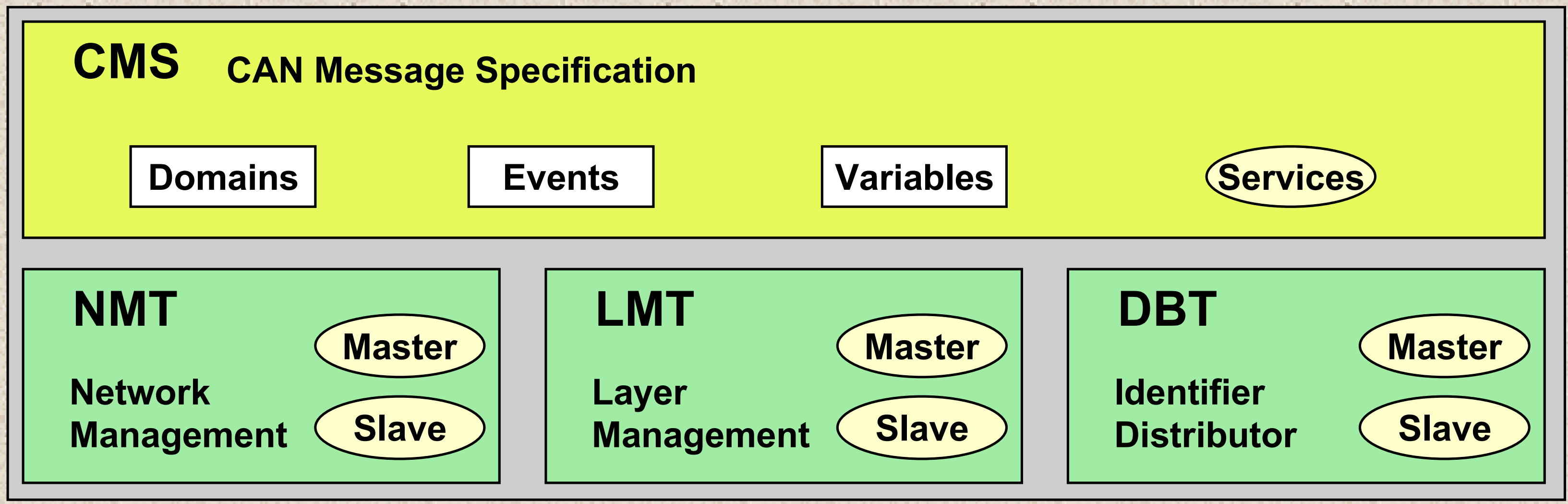
CAL = CAN Application Layer

was established in 1992 by the CAN in Automation association (CiA) in order to be able to develop open CAN systems.

CAL provides developers with numerous tools without giving any instructions for use.

Historically, this was the first step towards interoperability...

Structure of CAL



CMS defines standard communication objects and services

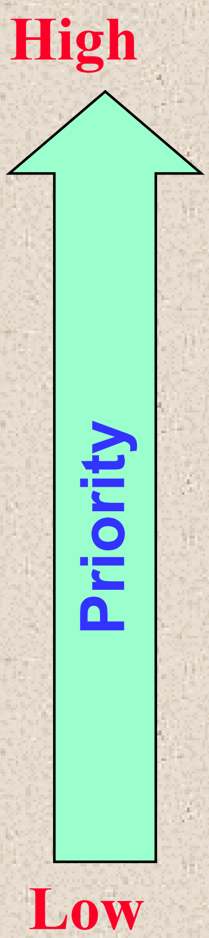
NMT initializes and supervises the network

LMT sets the parameters for the various layers

DBT is used to distribute identifiers in real time (little used).

Breakdown of CAL identifiers = COB - ID

COB – ID (decimal)	Services
0000	NMT start / stop services
0001 – 0220	CMS objects priority 0
0221 – 0440	CMS objects priority 1
0441 – 0660	CMS objects priority 2
0661 – 0880	CMS objects priority 3
0881 – 1100	CMS objects priority 4
1101 – 1320	CMS objects priority 5
1321 – 1540	CMS objects priority 6
1541 – 1760	CMS objects priority 7
1761 – 2015	NMT Node guarding
2016 – 2031	NMT, LMT, DBT
2032 – 2047	Reserved



High

Priority

Low

CANopen application layer

CANopen uses a restricted number of tools provided by CAL.

CANopen defines for a certain number of industrial devices:

what data is sent

and how by means of profiles

CANopen is based on the concept of the object dictionary:

Device Object Dictionary (OD) such as INTERBUS and PROFIBUS.

CANopen Object Dictionary = OD

The CANopen Object Dictionary is an **ordered group of objects** which can be accessed by a 16-bit index and if necessary a sub-index coded on 8 bits.

Each network node has an OD in the form of an
EDS: Electronic Data Sheet ASCII type file.

This dictionary contains all the elements which describe the node and its behaviour on the network.

Structure of the CANopen Object Dictionary

Index (hex)	Object
0000	Reserved
0001 – 009F	Data Types Area
00A0 – 0FFF	Reserved
1000 – 1FFF	**Communication profile Area**
2000 – 5FFF	**Manufacturer Specific Profile Area**
6000 – 9FFF	**Standardized Device Profile Area**
A000 – FFFF	Reserved

CANopen profiles

CANopen defines 2 types of profile:

The DS-301 communication profile:

Describes the general structure of the OD and objects found in the "Communication profile area" zone. It applies to all CANopen products.

DSP-4xx device profiles:

Describes the various associated standard objects for the various types of product (discrete I/O modules, drives, measuring apparatus).

Some objects are compulsory, others are optional, some are read-only, others are read-write.

DS-301 CANopen communication profile

The CANopen communication profile defines 4 types of message:

- **network administration** messages
- Service Data Objects = **SDO**
- Process Data Objects = **PDO**
- Special Function Objects = **SFO**

CANopen network administration messages

These messages are used to manage the various protocol layers, supervise the complete network and distribute identifiers.

They are based on the **CAL LMT, NMT and DBT** services and protocols.

Service Data Objects = SDO

These services are used to transmit **large amounts of parameter-setting data without any time restriction.**

They enable a client device to access the OD object dictionary of a server device in read or write mode.

The data can exceed 8 bytes, but in this case a data segmentation system is activated.

The result of a read or write operation is **confirmed** by a response.

An SDO requires 2 COB-IDs: one for the request, the other for the response.

Process Data Objects = PDO

These services are used to transmit **small amounts of process data (<= 8 bytes) in real time.**

They enable a **producer** device to make a variable with a maximum size of **64 bits without overhead** available to **one or more consumers.**

This service is implemented as an **" Event-type CMS object"** and therefore is **not confirmed**.

Description of PDOs

Each PDO is described by **2 objects in the OD**:

PDO Communication Parameter indicates which COB-ID is used, which type of transmission is used, and the value of the inhibit time.

PDO Mapping Parameter contains the list of objects in the OD as well as their size.

PDO transmission modes

Synchronous: by receiving a SYNC object

Acyclic: - transmission is pre-triggered by a "Remote Transmission request"

- transmission is pre-triggered by the occurrence of a "Specific event" object in the device profile

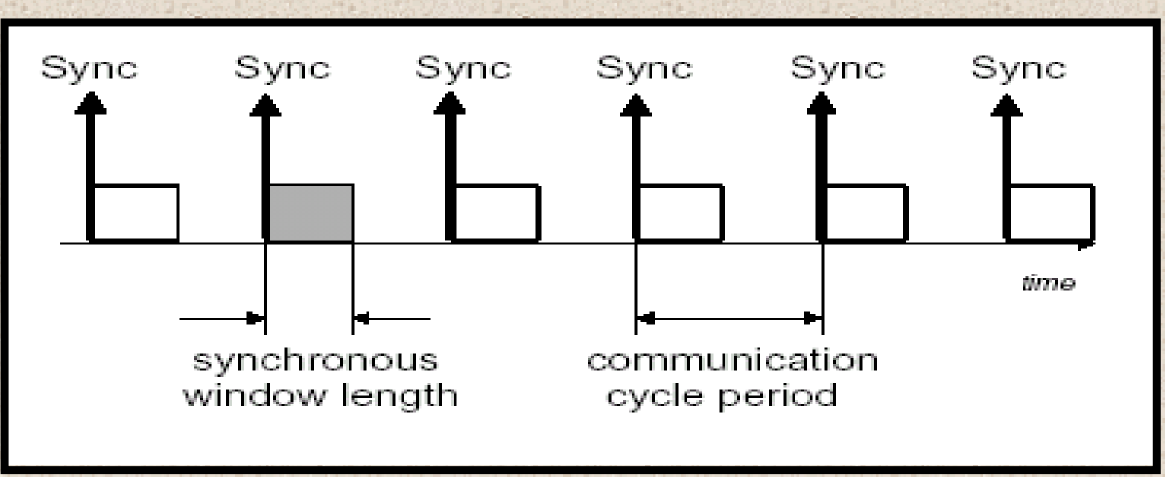
Cyclic: - transmission is triggered periodically after each 1, 2 or up to 240 SYNC objects

Asynchronous:

- transmission is triggered by a "Remote Transmission request"
- transmission is pre-triggered by the occurrence of a "Specific event" object in the device profile

Special Function Objects = SFO

SYNC = Synchronization Object:

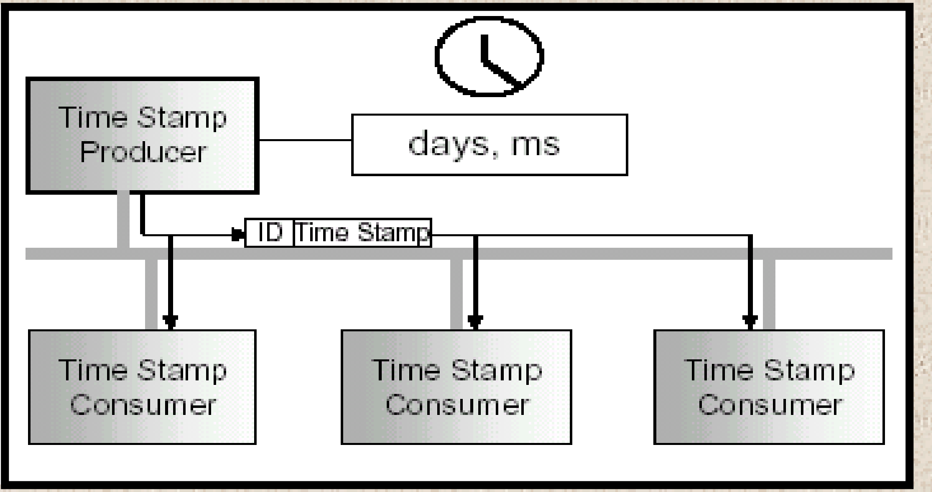


These objects are used to synchronize the acquisition of inputs or updating of outputs (axis control for example).

The "SYNC master" sends the SYNC message at a period (communication cycle period) fixed at the time of configuration.

Special Function Objects = SFO

Time Stamp Object:

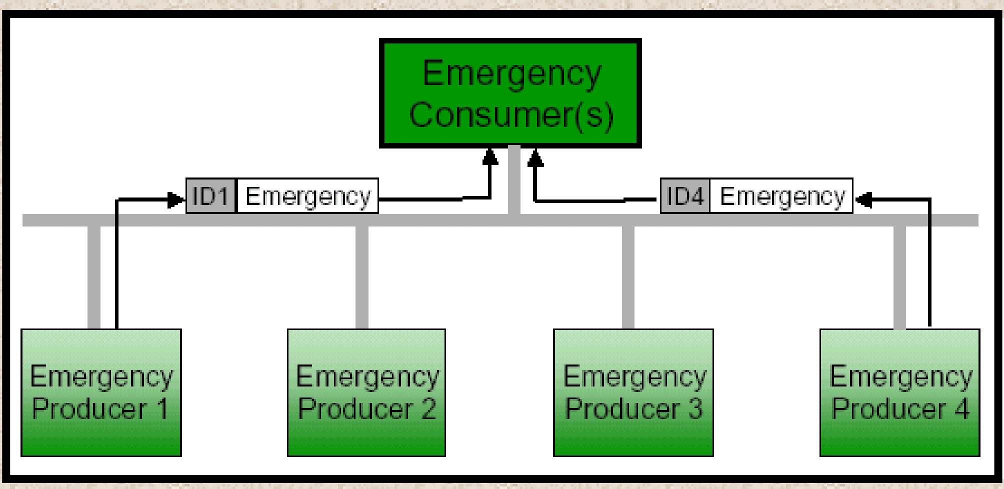


The Time-Stamp object provides a time reference common to all devices.

This time is coded on 6 bytes and represents an absolute time in ms from 1st January 1984.

Special Function Objects = SFO

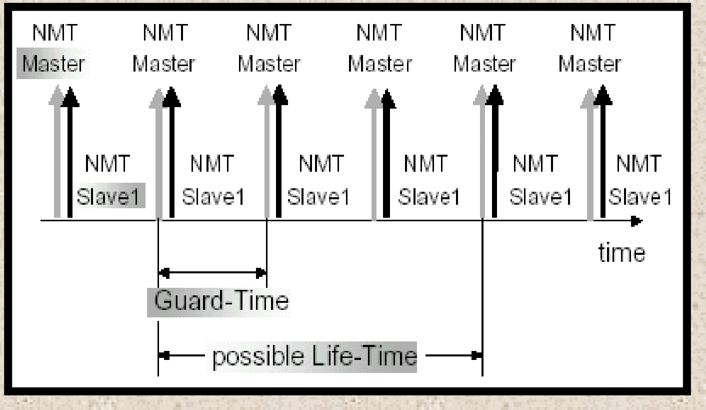
EMCY Object:



EMCY objects are triggered when an internal fault occurs in the device (current, voltage, temperature, communication, etc).

Special Function Objects = SFO

Node guard Object:



The NMT master monitors the status of slaves connected to the network by periodically sending a **Node guard object** remote frame to each slave.

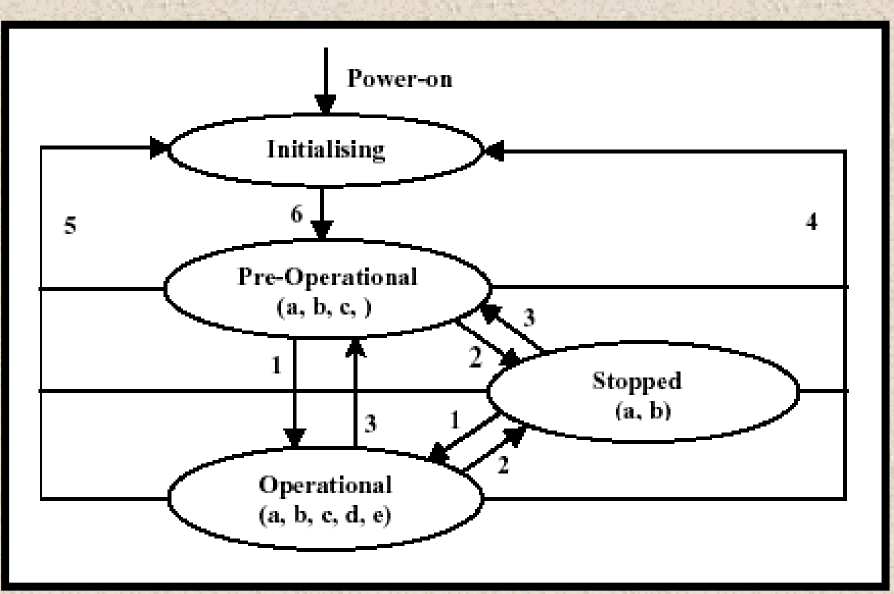
Each slave answers immediately on reception.

Slaves have the option of monitoring the NMT master: **Life guarding**.

Life Time = Guard Time x Life Time Factor

If a slave receives no polling for a period equal to the Life Time, it generates a "Life guarding" event, goes into communication fault mode and sends an EMCY object.

Network status diagram



Transitions performed by the NMT master:

1: Start_Remote_Node

2: Stop_Remote_Node

3: Enter_Pre-Operational_State

4: Reset_Node

5: Reset_Communication

6: Node initialization complete

Types of communication object authorized:

a. NMT

b. Node Guard

c. SDO

d. EMCY

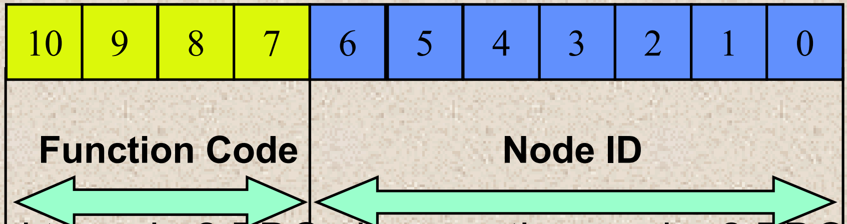
e. PDO.

Allocation of default identifiers

With the aim of reducing the network configuration phase a **compulsory system for allocating default identifiers** has been defined.

This allocation occurs in the "Pre operational" state just after the initialization phase.

It is based on dividing the COB-ID identifier into 2 parts:



10	9	8	7	6	5	4	3	2	1	0
Function Code				Node ID						

Function code is used to code 2 PDOs in reception mode, 2 PDOs in transmission mode, 1 SDO, 1 EMCY object, 1 Node Guarding Identifier, 1 SYNC object, 1 Time Stamp object and 1 node guarding.

Node ID corresponds to the product address coded by DIP switches, for example.

Allocation of default identifiers

General broadcast objects			
Object	Function Code	COB-ID	CMS Priority group
NMT	0000	0x000	0
SYNC	0001	0x080	0
TIME STAMP	0010	0x100	1

Point-to-point broadcast objects			
Object	Function Code	COB-ID	CMS Priority group
Emergency	0001	0x081-0x0FF	0, 1
PDO 1 sending	0011	0x181-0x1FF	0, 1
PDO 1 receiving	0100	0x201-0x27F	2
PDO 2 sending	0101	0x281-0x2FF	2, 3
PDO 2 receiving	0110	0x301-0x37F	3, 4
SDO as server	1011	0x581-0x5FF	6
SDO as client	1100	0x601-0x67F	6, 7
NODE GUARD	1110	0x701-0x77F	1

CANopen

Section 3: Link layer

▶ Part 1: Frame format

▶ Part 2: Exchange protection

CAN 2.0.A and CAN 2.0.B

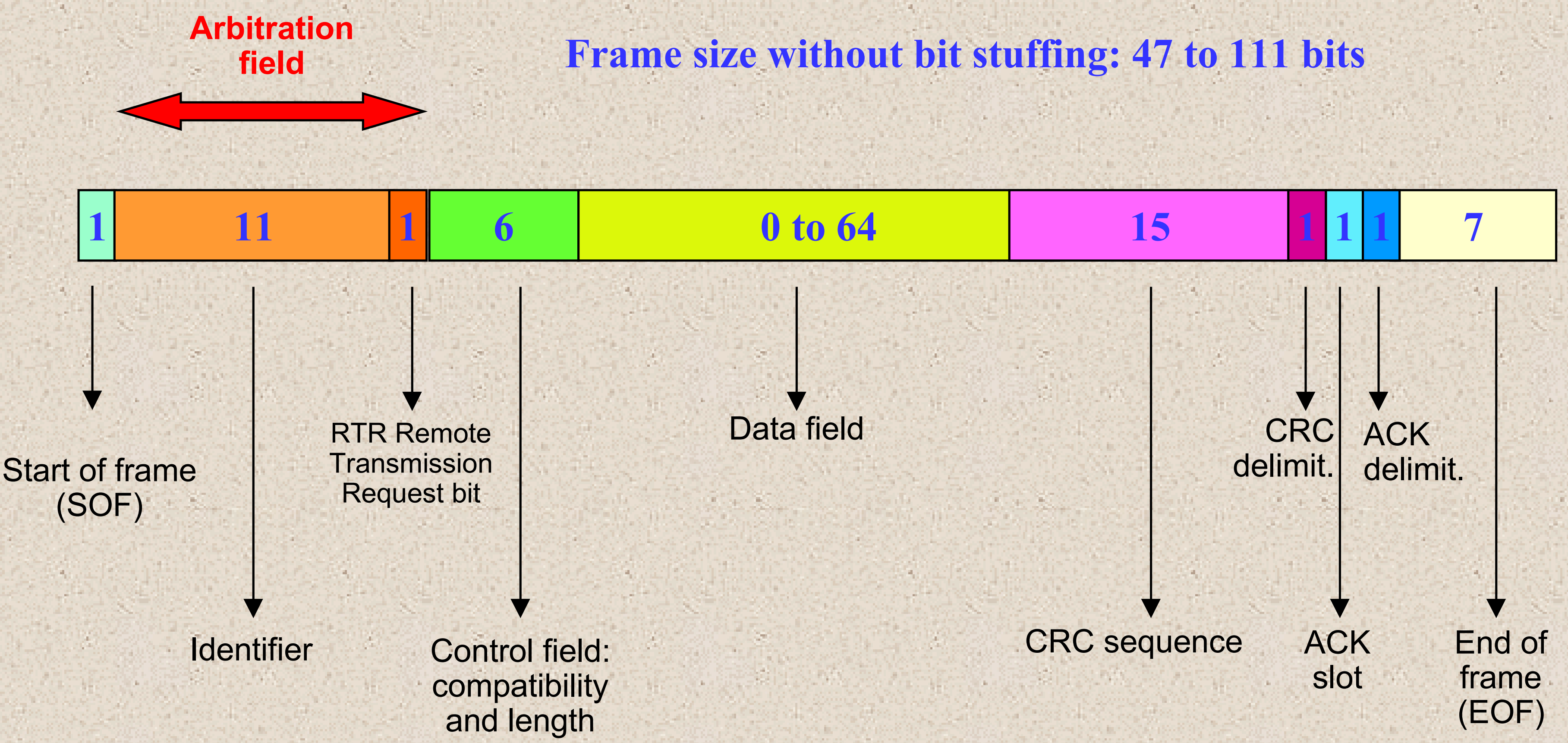
The CAN V2.0 specification consists of 2 parts: CAN 2.0.A and CAN 2.0.B.

CAN 2.0.A corresponds to the **standard frame** format with an **identifier coded on 11 bits**. It is **used by CANopen** and most of the application layers.

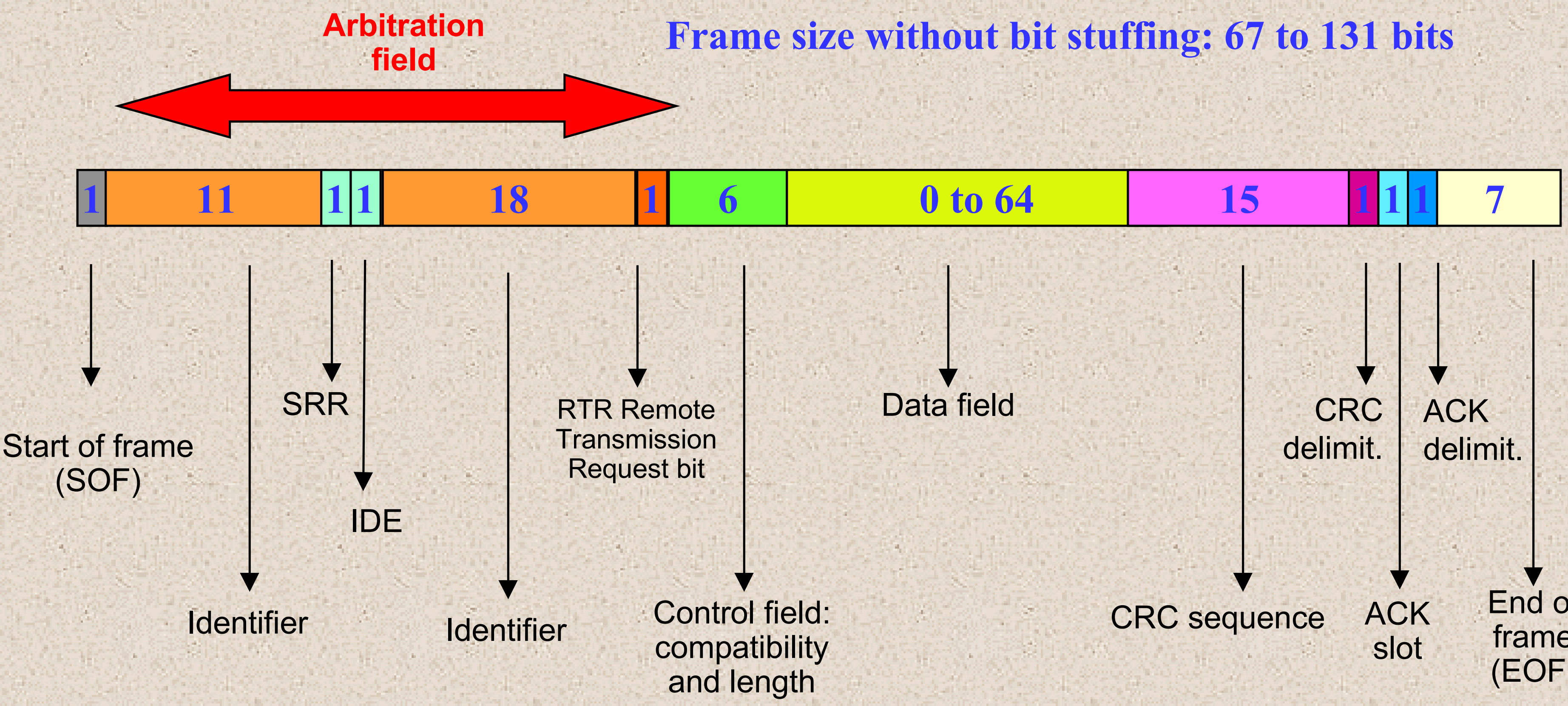
CAN 2.0.B corresponds to the **extended frame** format with an **identifier coded on 29 bits**. It is very rarely used.

Structure of the CAN 2.0.A frame

Frame size without bit stuffing: 47 to 111 bits



Structure of the CAN 2.0.B frame



Compatibility between CAN 2.0.A and CAN 2.0.B

Upward compatibility applies.

3 types of product have been defined:

CAN 2.0.A: these products prohibit exchanges of CAN 2.0.B frames since they systematically destroy them.

Passive CAN 2.0.B: these products ignore CAN 2.0.B frames without destroying them. They allow both exchange formats but only process frames in CAN 2.0.A format.

CAN 2.0.B: these products are capable of exchanges in both formats.

The 4 types of CAN frame

Data Frame: these frames transport data from a producer to consumers without any guarantee that it will be processed.

Remote Frame: these request frames are sent by a client to a server to request transmission of a data frame (the identifier will have the same value as that of the request).

Error Frame: these frames are transmitted when a station detects the presence of errors on the bus.

Overload Frame: these frames are sent to ask for an additional time lapse between successive frames (data or request).

CAN V2.0 A Data Frame

Data frame

SOF	IDENT	RTR	CTR	DATA	CRC	ACK	EOF	Interframe
1D	11X	1D	6X	0 to 64X	5X+1R	1X+1R	7R	3R

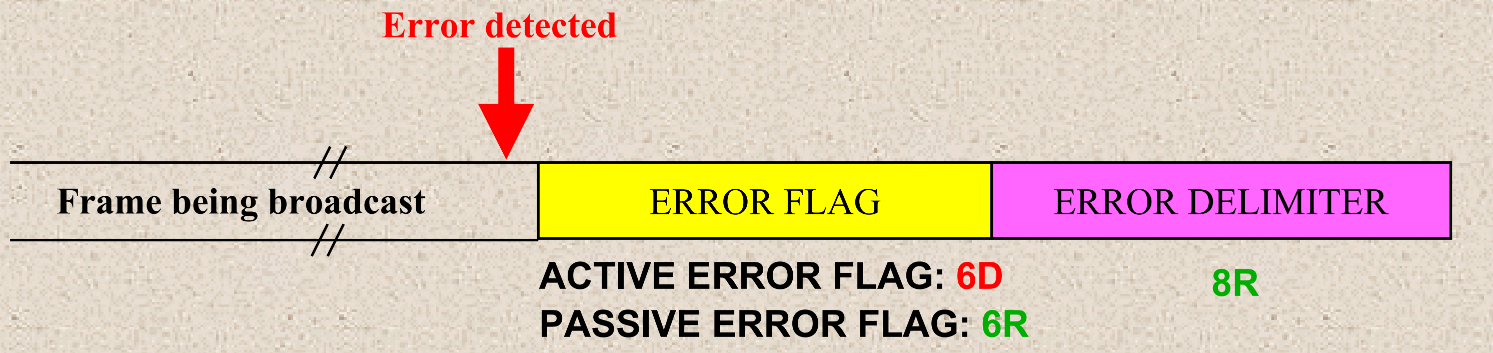
CAN V2.0 A Remote Frame

Request frame

SOF	IDENT	RTR	CTR	DATA	CRC	ACK	EOF	Interframe
1D	11X	1R	6X	0 to 64X	5X+1R	1X+1R	7R	3R

If a request frame is repeated, the response to this request takes priority.

Error Frame



Error detected

Frame being broadcast

ERROR FLAG

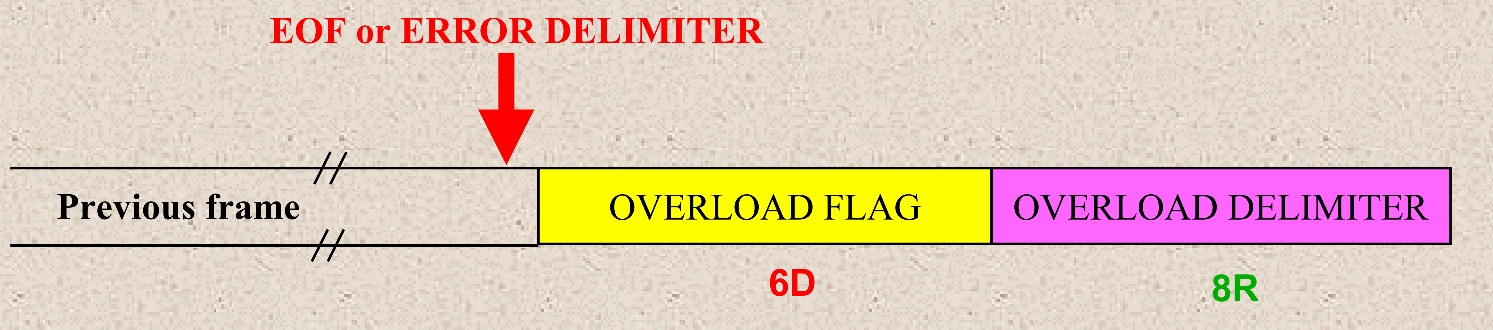
ERROR DELIMITER

ACTIVE ERROR FLAG: 6D

PASSIVE ERROR FLAG: 6R

8R

Overload Frame



EOF or ERROR DELIMITER

Previous frame

OVERLOAD FLAG

OVERLOAD DELIMITER

6D

8R

Protection mechanisms

At bit level: when 5 identical bits are transmitted, an additional "stuffing" bit with the opposite value is introduced intentionally. This bit is tested and eliminated by the receiver.

At frame structure level, the delimiters CRC Delimiter, ACK Delimiter, End of Frame, Error Delimiter, Overload Delimiter are integrated to enable the structure to be checked.

At content validity level: a CRC sequence enables receivers to check the consistency of the data received.

ACK slot: this window enables the sender to know that his message has been received correctly by at least one station (dominant bit).

Error counters

Each node must always contain two counters:

TEC Transmit Error Counter and REC Receive Error Counter.

These counters are incremented and decremented using a sophisticated weighting mechanism etched in the silicon.

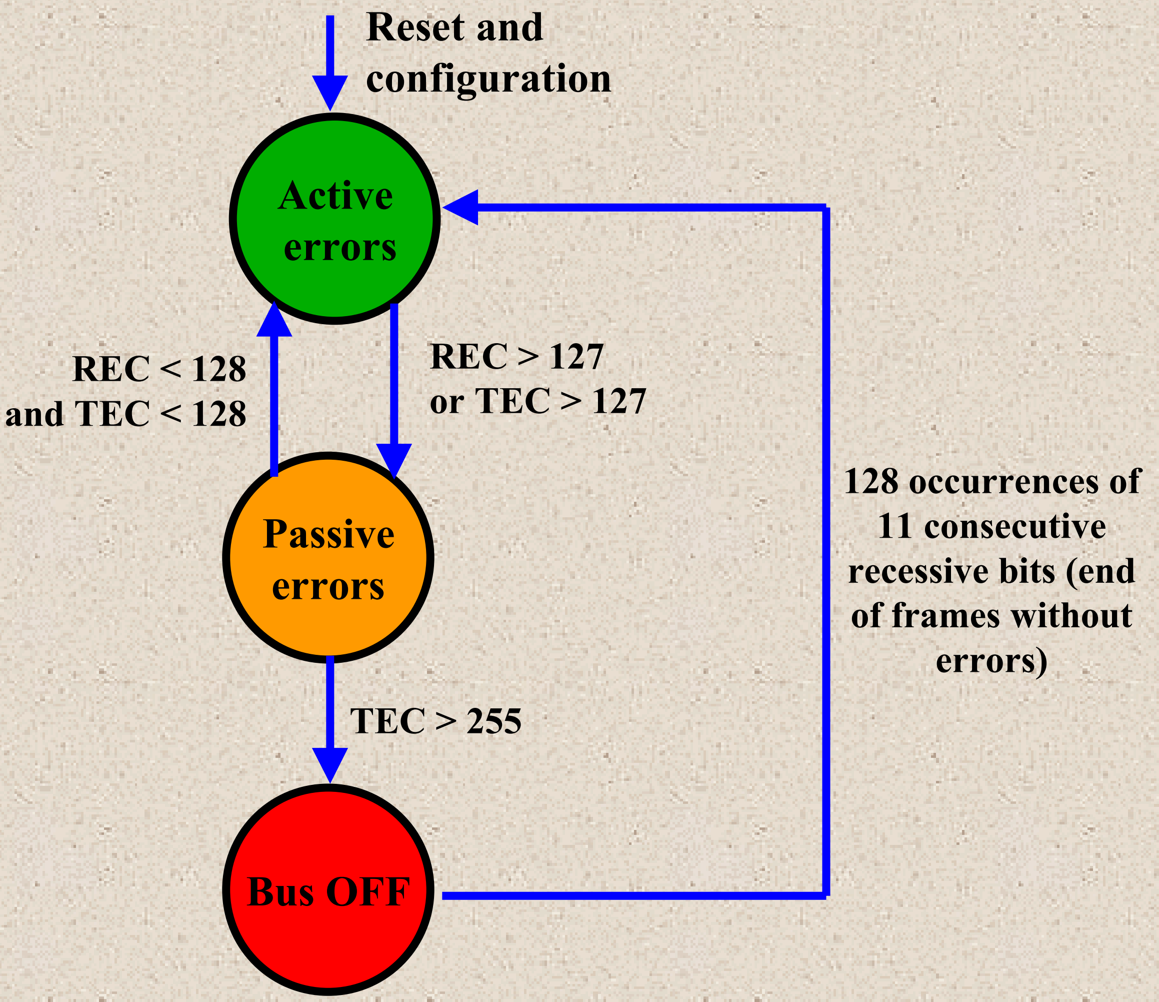
Depending on the value of these counters, the node will be in one of the following 3 states:

Active errors

Passive errors

Bus OFF (sending driver disconnected from the bus).

Counter value/node status



Behaviour in the event of detection of a communication error

Active error status: On detection of the fault, the node sends an Error Frame with an ACTIVE ERROR FLAG field.

The 6 dominant bits sent which contravene the stuffing bit law cause a chain reaction in the other nodes which destroys the active frame.

Passive error status: On detection of the fault, the node sends an Error Frame with a PASSIVE ERROR FLAG field.

The recessive bits sent have no effect on the frame currently being sent.

Bus OFF status: The node is disconnected and monitors the bus.

CANopen

Section 4: Physical layer

▷ Part 1: Network characteristics

▷ Part 2: Recommended connections

CANopen physical layer

Device Profile CiA DSP-401 I/O modules

Device Profile CiA DSP-402 Drives

Device Profile CiA DSP-404 Measuring devices

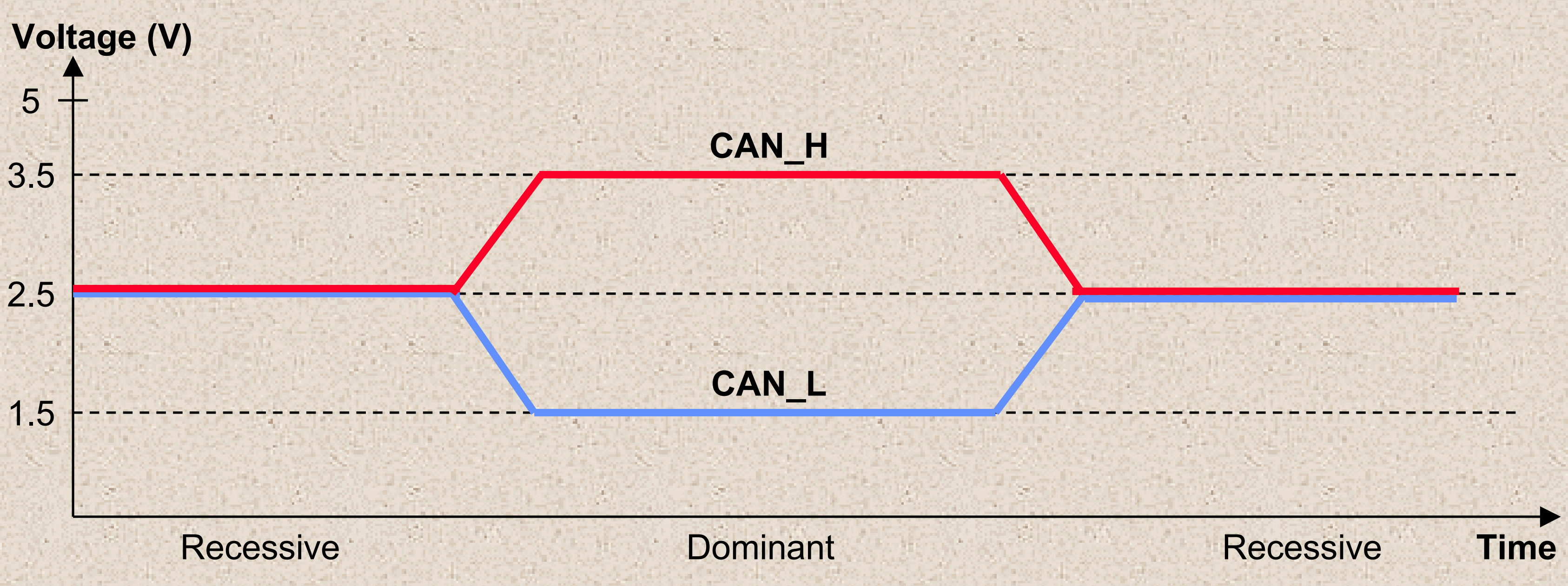
Device Profile CiA DSP-4xx

7	APPLICATION	CiA DS-301 = Communication profile CAL= CAN Application Layer
6	PRESENTATION	EMPTY
5	SESSION	EMPTY
4	TRANSPORT	EMPTY
3	NETWORK	EMPTY
2	LINK = LLC + MAC	CAN 2.0 A and B + ISO 11898
1	PHYSICAL	CAN 2.0 A and B + ISO 11898 ISO 11898 + DS-102 + DRP-301-1

Description of the network

Twisted differential pair:	1 pair if CAN-H/CAN-L 2 pairs if CAN-H/CAN-L + p. supply
Characteristic line impedance:	120 ohms nominal
Line terminators:	120 ohms at each end
Wire resistance:	70 milli-ohms/metre nominal
Propagation time:	5 ns/metre nominal
Topology:	Bus type with the shortest possible tap links

Electrical levels



Recessive status:	VCAN_H - VCAN_L = 0V (-0.5 V to + 50 mV)
Dominant status:	VCAN_H - VCAN_L = 2V (1.5 V to 3 V)
Tx output current:	Higher than 25 mA
Protection:	Against short-circuits

Data rate - bus length - cable cross-section for 32 slaves maximum

Data rate	Bus length	Cable x-section
1 Mbps	25 m	0.25 mm^2 AWG 23
800 Kbps	50 m	0.34 mm^2 AWG 22
500 Kbps	100 m	0.34 mm^2 AWG 22
250 Kbps	250 m	0.34 mm^2 AWG 22
125 Kbps	500 m	0.5 mm^2 AWG 20
50 Kbps	1000 m	0.75 mm^2 AWG 18
20 Kbps	1000 m	0.75 mm^2 AWG 18
10 Kbps	1000 m	0.75 mm^2 AWG 18

Data rate - bus length - cable cross-section for 100 slaves maximum

Data rate	Bus length	Cable x-section
1 Mbps	25 m	0.34 mm^2 AWG 22
800 Kbps	50 m	0.6 mm^2 AWG 20
500 Kbps	100 m	0.6 mm^2 AWG 20
250 Kbps	250 m	0.6 mm^2 AWG 20
125 Kbps	500 m	0.75 mm^2 AWG 18
50 Kbps	1000 m	1 mm^2 AWG 17
20 Kbps	1000 m	1 mm^2 AWG 17
10 Kbps	1000 m	1 mm^2 AWG 17

Recommended connections

In its recommendation DR-303-1, the CiA provides a list of suitable connectors classified into 3 categories with their pin signal description.

General use

9-pin SUB D connector DIN 41652, multi-pole connector (ribbon cable to 9-pin SUB-D), RJ10 and RJ45

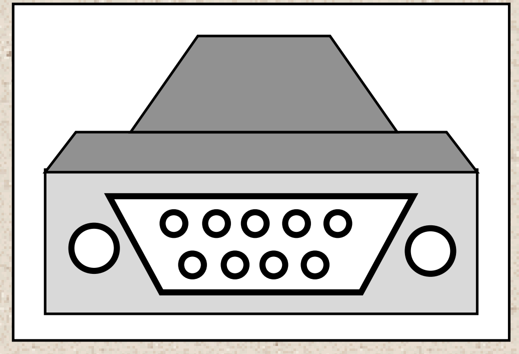
Industrial use

5-pin Mini Style, 5-pin Micro Style, Open Style

Special use

7-pin round connector, 8-pin round connector, 9-pin round connector, 12-pin round connector, Hand Brid Harting.

9-pin SUB D connector DIN 41652

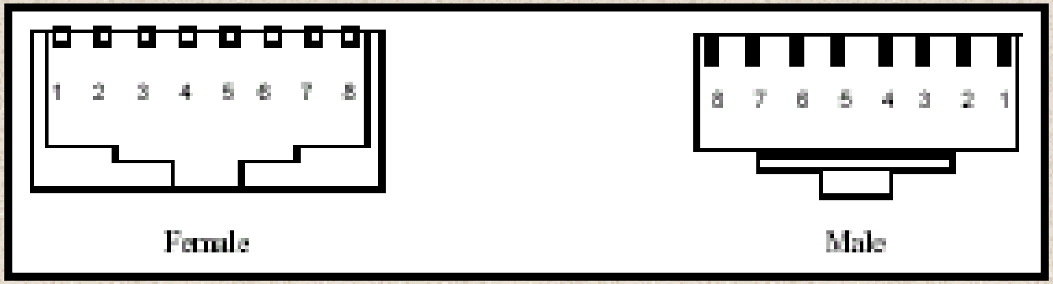


Male product end

Pin Signal Description:

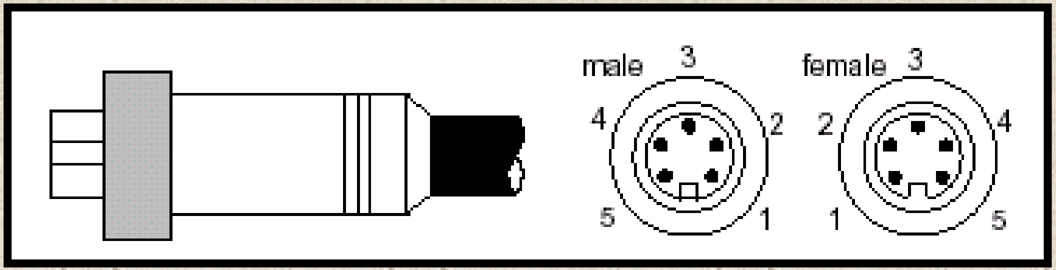
1:	Reserved
2:	CAN_L = CAN_L bus line dominant low
3:	CAN_GND = CAN Ground
4:	Reserved
5:	(CAN_SHLD) Optional CAN Shield
6:	(GND) Optional Ground
7:	CAN_H = CAN_H bus line dominant high
8:	Reserved
9:	(CAN_V+) Optional CAN external positive supply

RJ45 connector



Pin	Signal Description
1:	CAN_H = CAN_H bus line (dominant high)
2:	CAN_L = CAN_L bus line (dominant low)
3:	CAN_GND = Ground/0 V/V-
4:	Reserved
5:	Reserved
6:	(CAN_SHLD) = Optional CAN Shield
7:	CAN_GND = Ground/0 V/V-
8	(CAN_V+) = Optional CAN external positive supply

5-pin Mini Style connector: ANSI/B93.55M-1981



Male product end

Pin Signal Description:

1: (CAN_SHLD) = Optional CAN Shield

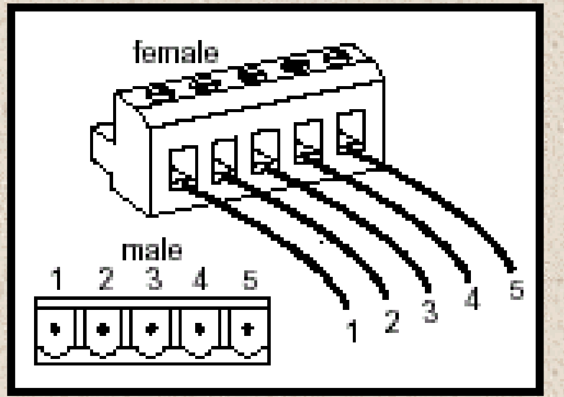
2: (CAN_V+) = Optional CAN external positive supply

3: CAN_GND = Ground/0V/V-

4: CAN_H = CAN_H bus line (dominant high)

5: CAN_L = CAN_L bus line (dominant low)

Open Style connector



Male product end

Pin Signal Description:

1: CAN_GND = Ground/0 V/V-

2: CAN_L = CAN_L bus line (dominant low)

3: (CAN_SHLD) = Optional CAN Shield

4: CAN_H = CAN_H bus line (dominant high)

5: (CAN_V+) = Optional CAN external positive supply

Recommended suppliers

Cables

- U.I.LAPP GmbH

Schultze-Delitsch-Str. 25
D-70565 Stuttgart Germany

http://www.lappcable.com

Connectors

- ERNI Elektroapparate GmbH

Seestrasse 9 D-73099 Adelberg Germany

- ERNI Connectique S.a.r.l, France

27 bis, avenue des Sources/CP 638 F-69258 LYON Cedex 09,

http://connect.erni.com/

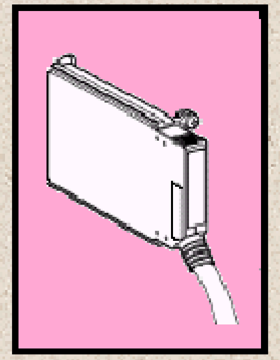
CANopen

Section 5: Schneider offer overview

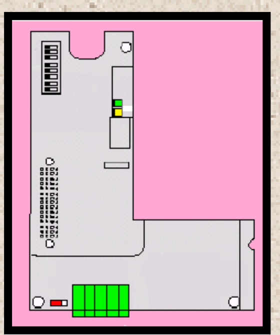
▷ Part 1: Global offer overview overview

▷ Part 2: Product characteristics

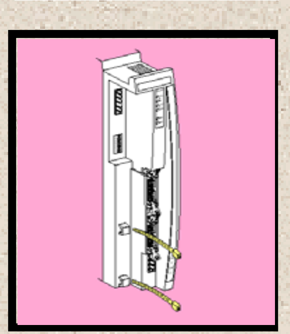
Products



CANopen master PCMCIA card for Premium PLC:
TSXCPP100



CANopen communication card for ATV58 variable speed drive:
VW3A58308



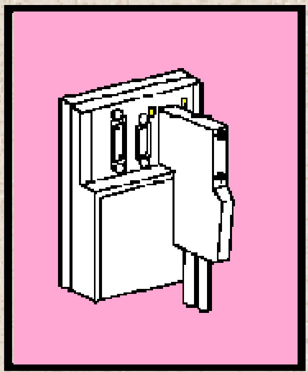
CANopen communication module for TEGO POWER / QUICKFIT motor starters:

APP1CCO0 = 16I + 8O on bus

APP1CCO2 = 16I + 8O on bus + 16I + 16O locally

Accessories

CANopen Sycon master configuration software from HILSCHER:
LXLFBCM



Junction box for connecting the CANopen master card to 1 or 2 CANopen buses:
TSXCPPACC1

Schneider does not provide cables or connectors for CANopen, but recommends suppliers.

Recommended suppliers

Cables

- U.I.LAPP GmbH

Schultze-Delitsch-Str. 25
D-70565 Stuttgart Germany

http://www.lappcable.com

Connectors

- ERNI Elektroapparate GmbH

Seestrasse 9 D-73099 Adelberg Germany

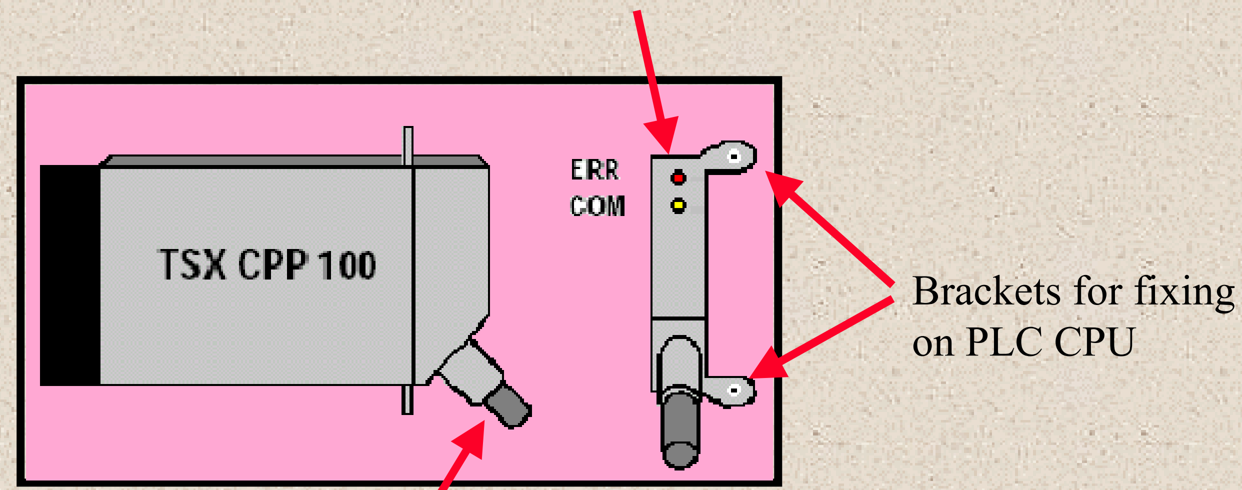
- ERNI Connectique S.a.r.l, France

27 bis, avenue des Sources / CP 638 F-69258 LYON Cedex 09,

http://connect.erni.com/

CANopen master card: TSXCPP100

Type III PCMCIA card inserted in the processor module communication port slot (only 1 CANopen master per PLC).



Connecting cable, 0.6 m long, with male 15-pin SUBD at the end

CANopen master card: TSXCPP100

Compatibility:

Premium processors >= V5.0 except for TSX57103.

PL7 >= V4.0

Sycon configurator >= 2.630

The Sycon configurator is used to generate the ***.co configuration file which can be imported from PL7 or loaded directly into the card.

CANopen master card: TSXCPP100

Functions supported:

Bus management

Implicit exchange of PDOs (8 bytes max.) using %MW words

Explicit exchange of SDOs by WRITE_VAR and READ_VAR functions

Sending any identifier by the SEND_REQ function

Fault identification, diagnostics and history by the SEND_REQ function

CANopen master card: TSXCPP100

Processor	Max. size of configuration file (1)	Max. size of I/O data (PDO)	
		Mast task	Fast task
TSXP57-203	16 Kb	512 %MW (256 + 256)	64 %MW (32 + 32)
TSXP57-253	16 Kb	512 %MW (256 + 256)	64 %MW (32 + 32)
TSXP57-303	32 Kb	1024 %MW (512 + 512)	128 %MW (64 + 64)
TSXP57-353	32 Kb	1024 %MW (512 + 512)	128 %MW (64 + 64)
TSXP57-453	64 Kb	3584 %MW (1792 + 1792)	256 %MW (128 + 128)

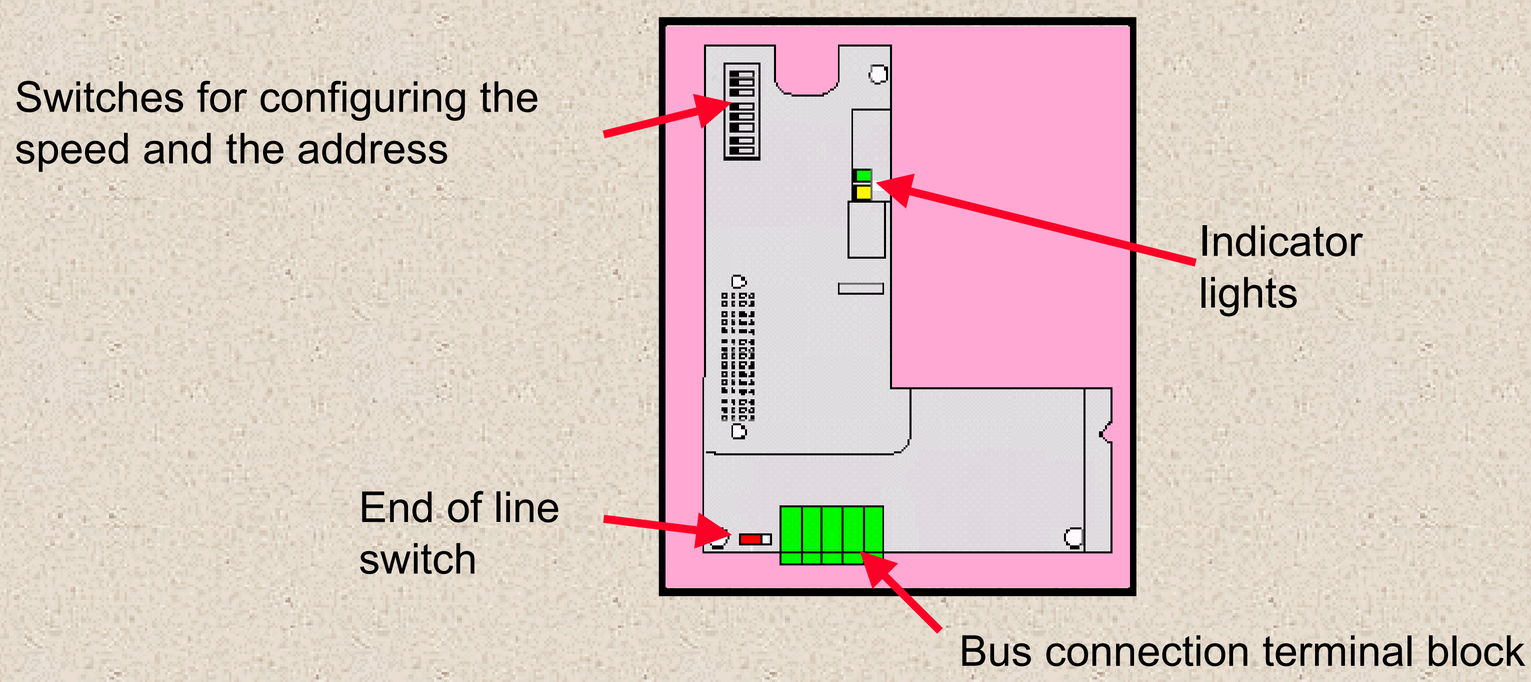
(1) This maximum size can be exceeded if the configuration file is loaded directly into the card using the Sycon software. The maximum file size authorized by Sycon is 256 Kb.

For 10 ATV58: Configuration file size = 4.6 Kb I/O data = 2 X 20 words

For 20 ATV58: Configuration file size = 7.5 Kb I/O data = 2 X 40 words

CANopen communication card for ATV58: VW3A58308

L-shaped electronic card connected to the drive control card.


Switches for configuring the speed and the address
Indicator lights
End of line switch
Bus connection terminal block

Supplied with floppy disk containing the .eds description file for the device.

Compatible with all ATV58 and ATV58F control cards.

CANopen communication card for ATV58: VW3A58308

Main characteristics:

Communication speed (125, 250, 500 Kbps or 1 Mbps) and address configured by switches

Conforms to CAN V2.0.A: identifiers on 11 bits

The CAN controller used on the card is active 2.0B: it accepts frames with identifiers on 11 and 29 bits.

Identifier value allocated by the address

Conforms to DS402 V1.1 CANopen profile: "Device profile for drives and motion control - Velocity Mode"

Conforms to Drivecom profile

CANopen communication card for ATV58: VW3A58308

PDO services supported (control):

1 PDO in reception mode (control) non-synchronised containing **4 bytes** corresponding to 2 words:

CMDD:	DRIVECOM control register
LFRD:	DRIVECOM speed reference in rpm

1 PDO in send mode (monitoring) non-synchronised sent on change of status* containing **4 bytes** corresponding to 2 words:

ETAD:	DRIVECOM status word
RFRD:	DRIVECOM motor speed in rpm

*** PDOs are sent by the drive at intervals of 50 ms minimum and one second maximum.**

CANopen communication card for ATV58: VW3A58308

SDO services supported (configuration, adjustment):

1 SDO in reception mode: processing a request to read or write drive communication internal variables: **1 to 4 bytes**

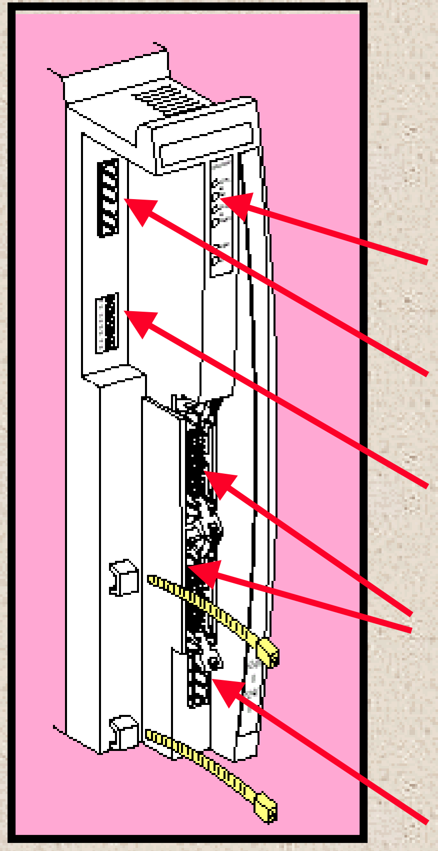
1 SDO in send mode: processing the response to a request to read or write drive communication internal variables: 1 to 4 bytes

NMT services: network management status graph

Node guarding object: periodic transmission of drive status

EMCY object : sent in the event of a drive fault

CANopen communication module for TEGO POWER/QUICKFIT



Communication module placed on the left side of the motor starters

Indicator lights

Bus connection terminal block

Switches for configuring the speed and address

Local I/O connection terminal blocks

(only on APP1CCO2 module)

24V power supply

CANopen communication module for TEGO POWER/QUICKFIT

Main characteristics:

Communication speed (125, 250, 500 Kbps or 1 Mbps) and address configured by switches

Conforms to CAN V2.A: identifier on 11 bits

Identifier values allocated by the address

Conforms to the "I/O module" DS401 CANopen profile

CANopen communication module for TEGO POWER/QUICKFIT

PDO services supported on APP1CCO0:

1 PDO in reception mode (control) non-synchronised containing 1 byte:

Control of 8 TEGO sub-base contactors

1 PDO in send mode (monitoring) non-synchronised sent on change of status containing 2 bytes:

State of 8 contactor auxiliary contacts

State of 8 circuit-breaker auxiliary contacts

CANopen communication module for TEGO POWER/QUICKFIT

PDO services supported on APP1CCO2:

1 PDO in reception mode (control) non-synchronised containing 4 bytes:

- Control of 8 TEGO sub-base contactors
- Control of 16 outputs external to the sub-base
- 4th bytes unused

1 PDO in send mode (monitoring) non-synchronised sent on change of status containing 4 bytes:

- State of 8 contactor auxiliary contacts
- State of 8 circuit-breaker auxiliary contacts
- State of 16 inputs external to the sub-base

CANopen communication module for TEGO POWER/QUICKFIT

No configuration or parameter setting

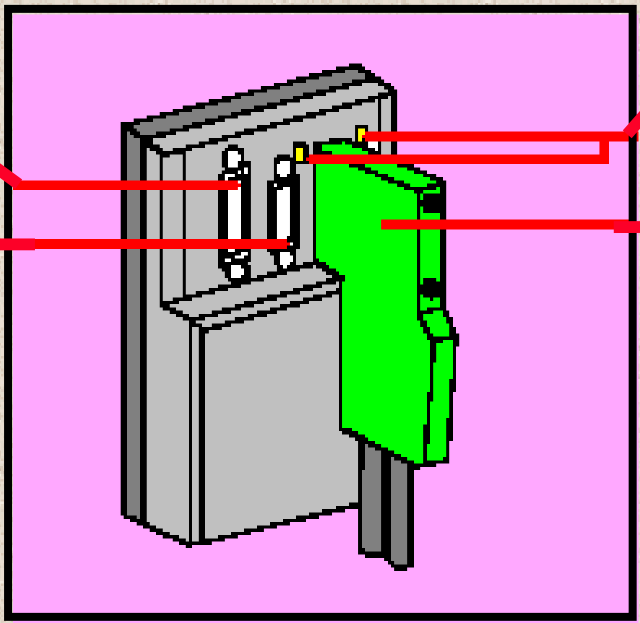
NMT services:	network management status graph
Node guarding object:	periodic transmission of motor starter status
EMCY object:	sent in the event of a drive fault

CANopen Sycon master configuration software from HILSCHER

Used to generate a configuration file for the CANopen master PCMCIA card.

This configuration can be imported from PL7 PRO/PL7 Junior or loaded directly by inserting the card in the PC.

Junction box for connecting the master card



15-pin SUBD sub-base for connecting the master PCMCIA card

9-pin SUBD sub-base for connecting the slaves on channel A

Activity indicators channels A and B

9-pin SUBD connector for connecting the slaves on channel B

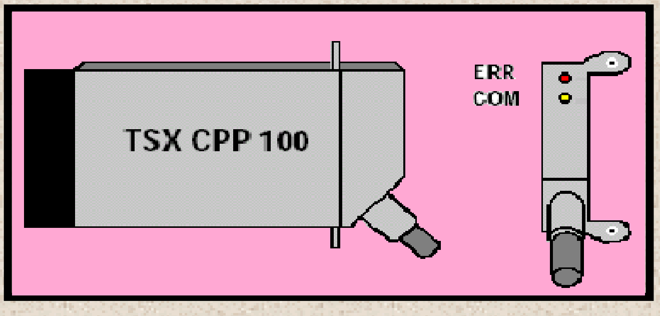
CANopen

Section 6: Diagnostics

▶ Part 1: Indicator lights

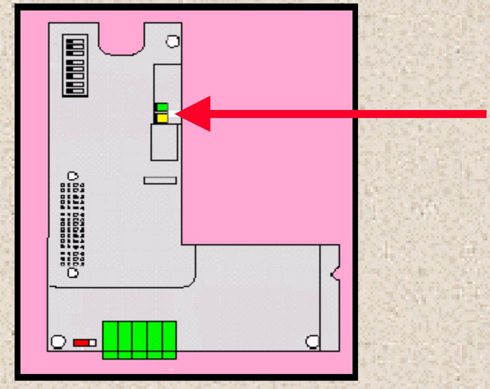
▶ Part 2: Diagnostic PLC objects

CANopen master card: TSXCPP100



LEDs		Meaning
ERR	**COM**	
Off	Off	No power supply to card or configuration transfer in progress
	Flashing (Irregular)	No configuration in the card
	Flashing (Regular)	Card configured and ready, bus not activated or no CANopen firmware
	On	Bus configured and active, no error
On steady	Off	Error detected, bus controller stopped
	Flashing	Card faulty, configuration error or synchronization error between the card and the PLC (for more information, refer to the module status diagnostic data)
	On	Bus configured and active, at least one bus subscriber cannot be reached or is signalling an error

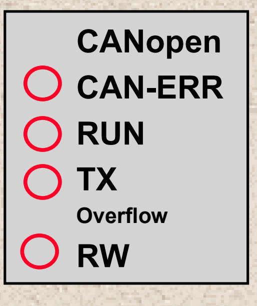
CANopen communication card for ATV58: VW3A58308



Green LED = OK
Red LED = COM

LED	Status	Meaning
Green LED "OK"	**Fixed green**	No fault.
	Flashing green	Address 0 or communication fault (CNF) or internal fault (ILF)
Yellow LED "COM"	**Flashing yellow**	Receiving PDO or SDO
	Off	Not receiving PDO or SDO

CANopen communication module for TEGO POWER/QUICKFIT



CANopen

○ CAN-ERR

○ RUN

○ TX

Overflow

○ RW

The first 4 LEDs relate to CANopen communication

CAN-ERR	Run	TX	RX	Meaning
0	1	0	0	OK: operational.
1	1	0	0	Bus Off: an error has occurred on CAN, the module has been disconnected. The CAN bus needs to be reset.
1	1/0	0	0	The data monitoring system has sent a message, the RUN LED flashes until a response has been received.
1	1/0	0	0	A synchronization error has occurred during the monitored message. The module has not received a synchronization "telegram".
1	0	1	0	A message has appeared which is too long to be sent. The module must be reset in order to return to normal.
1	0	0	1	A message has appeared which is too long to be received. The module must be reset in order to return to normal.

CANopen communication module for TEGO POWER/QUICKFIT

○ I/O RUN
○ I/O ERR

The 2 bottom LEDs relate to the module and the I/O

Green LED	Lights up if there is no communication error on CAN-Open and no errors during the I/O cycle
Red LED	Lights up if there is an error on the module or the I/O
	1 on steady: Module fault
	2 continuous flashing: Error during the I/O cycle
	3 intermittent flashing: Rapid flashing: start of error message : 1st sequence of slow flashing: Error code : 2nd sequence of slow flashing: type of error (argument)

Error table

1st sequence	2nd sequence	Description of the error
error code	Type (argument)	
1 pulse	0 pulse	- EEPROM memory error
	1 pulse	- RAM memory error
4 pulses	0 pulse	- I/O error

Implicit diagnostic objects

2 bits:

%Iy.MOD.ERR = module fault

%Iy.1.ERR = channel fault

24 words:

%IWy.1.0 to %Iwy.1.23 = used to determine:

the channel status

of the various slaves

the description of the last fault

Word %Iwy.1.0 is a general status word which, depending on the type of fault, is used to perform more detailed diagnostics.

Explicit diagnostic function

SEND_REQ(ADR#y.1.SYS, 16#0F, %MWi:L, %MWj:L, %M Wk:4):

used to access identification and status of the CANopen master

SEND_REQ(ADR#y.1.SYS, 16#31, %MWi:3, %MWj:L, %MWk:4):

depending on the value of %Mwi, used to:

perform diagnostics on a slave

or to determine the version/status of the CANopen master PCMCIA card

or to read the message handling error log (SDO)

Word %Iwy.1.0 is a general status word which, depending on the type of fault, is used to perform more detailed diagnostics.