Redes de Comunicação em Ambientes Industriais
Aula 12

Paulo Pedreiras pedreiras@det.ua.pt

Electronic Systems Lab-IEETA / DET
Universidade de Aveiro
Aveiro, Portugal

In the previous episode ... (1)

CAN

- ✓ Created by **Bosch**, Version 2.0 released in **1991**.
- ✓ Expanded to process control, manufacturing automation and embedded application domains
- ✓ Defines **physical** and **data link** layers only
- ✓ **Multi-master**, **broadcast**, serial **bus**
- ✓ Transmission rate from **5 Kbit/s** to **1 Mbit/s**
- ✓ Length depends on tx rate
  (aprox. 40m @ 1Mbit/s, 1000 @ 50Kbit/s)
- ✓ Maximum number of nodes depends on bus transceivers (32, 64, 128…)

In the previous episode ... (2)

- ✓ Bit encoding using **NRZ**
- ✓ Tx/Rx synchronization using **bit stuffing**
- ✓ Data **payload** between **0 and 8 bytes**
- ✓ **Source-addressing** (11/29 bits in Vers. A/B resp)
- ✓ **CSMA with Non-destructive arbitration** based on msg. IDs
- ✓ Analysis:
  - ✓ **Common analysis, acounting for the non-preempmtion**
    - ✓ **Utilization**

$$\sum_{1}^{N} \frac{C_i}{T_i} + \max_{1..N}\left(\frac{B_i + J_i + \tau}{T_i}\right) < N(2^{1/N} - 1)$$

    - ✓ **Response time**

$$Rwc_i = I_i + C_i$$

$$I_i = B_i + \sum_{j \text{ in } hp(i)} \left\lceil \frac{J_i + I_i + \tau}{T_j} \right\rceil * C_j$$

In this episode: CAN HLP

- CAN hardware implementation define only
  - Physical layer
  - MAC layer
- Leading to system design and deployment problems
  - Lack of interoperability and interchangeability
  - Lack of standard libraries for commonly required functions; complex application development
  - ...
- Several CAN higher layer protocols (HLP) proposed
  - CAL
  - CAN Kingdom
  - **CANopen**
  - DeviceNet
  - OSEK-COM/NM

CAN HLP

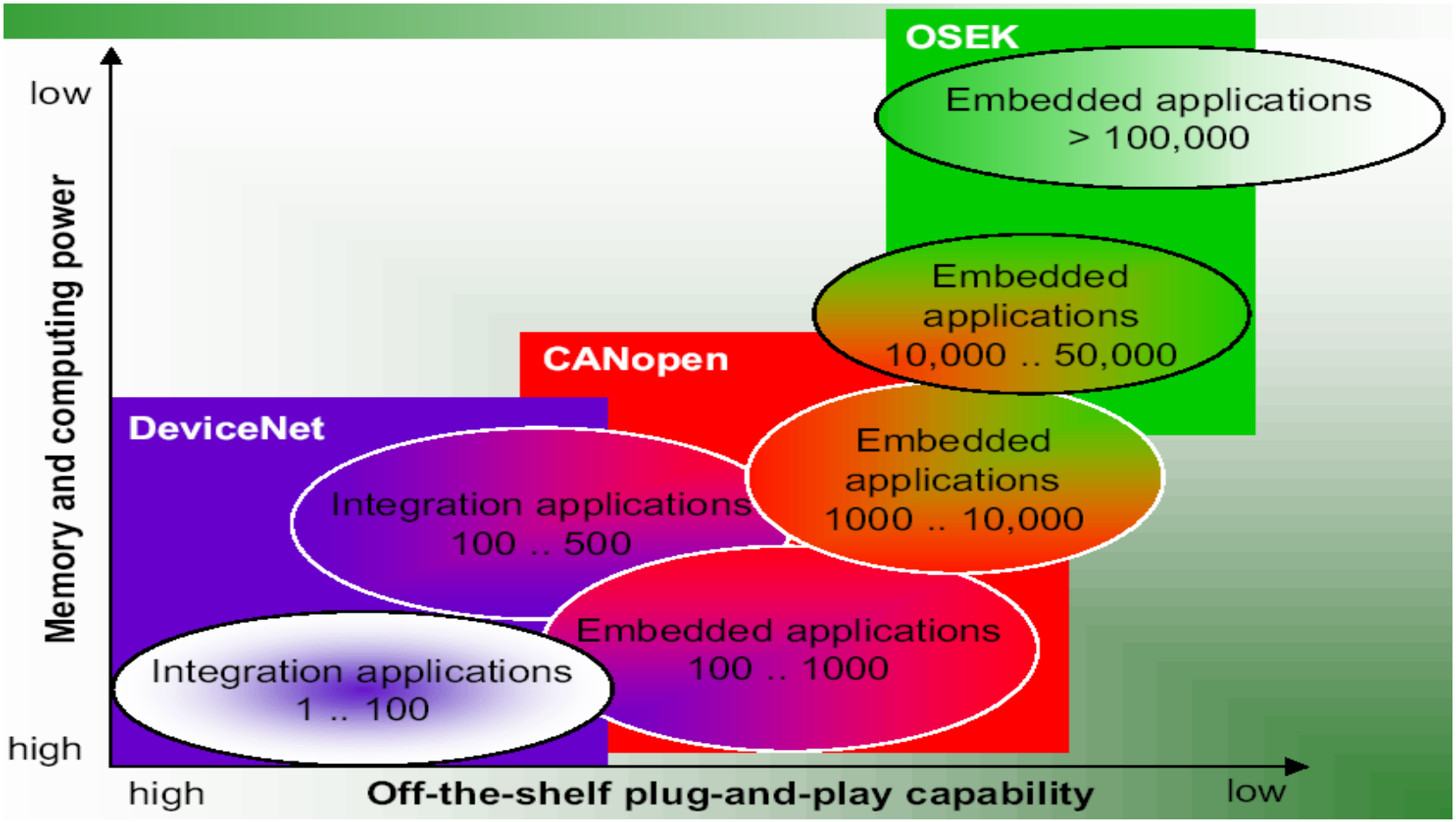


Figure: CANportuguese, CiA, June/2002

CANopen

- CANopen
  - communication and application standard for distributed systems
  - Maintained by the CAN-in-Automation (CiA) group
    http://www.can-cia.com
  - Key features:
    - Transmission process data according to the producer/consumer model
    - Standardized device description (data, parameters, functions, programs)
    - Standardized access to device parameters
    - Standardized services for device monitoring (e.g. membership functions based e.g. in heartbeat)
    - System services: synchronization message, central time-stamp message (e.g. synchronous data acquisition)
    - Emergency messages

CANopen

Physical layer issues

- CANopen requires
  - All nodes configured with the same bit rate
  - Unique node-IDs.
- Bit rates for CANopen networks given in DS-301:
  - 10, 20, 50, 125, 250, 500, 800 and 1000 kbps
  - DS-301 also includes bit timing configuration recommendations
- CAN IDs assigned by the system integrator
  - Directly on the device via DIP-switches or hexadecimal rotary switches
  - Software solution using two reserved CAN identifiers (LSS-service / layer setting service). **(NOTE:1-1 connection required!!)**.

CANopen

Object Dictionary (OD) and Electronic Data Sheet (EDS)

- CANopen provides a standardized device description (object dictionary)
- The device object dictionary has a form of a table with the same structure for all types of devices.
- Access to device's data, parameters and functions of a device based on a logical addressing scheme (16 bit index + 8 bit subindex)
- Access to the OD based on the SDO protocol
- CANopen defines device profiles for typical devices:
  - Specification of the most important parameters, data and functions per device type (e.g. input/output modules, drives, encoders)
  - CANopen-compatible devices are interchangeable at the basic functionality level

CANopen

Object Dictionary (OD) and Electronic Data Sheet (EDS)

- The object dictionary is sub-divided into standardized areas of 4096 entries each:
  - 1000-1FFF: communication area/communication profile
  - 2000-5FFF: manufacturer-specific device objects
  - 6000-9FFF: device profiles
  - A000-AFFF: network variables (NWV)
- Every OD object is associated with a value that can be read or written via SDO transfers
- The data type and meaning of each OD object must be known by the configuration tools ⇒ an electronic data sheet (EDS) describes each object dictionary entry
  - address (index/subindex), param. name, data type, access type and default value

CANopen

Object Dictionary (OD) and Electronic Data Sheet (EDS)

- CANopen data types:
  - bytes, words and double words, signed/unsigned
  - ASCII and Unicode strings
  - one-bit boolean data type
  - 32/64 floating point types in accordance with IEEE 754-1985
  - Time_of_day: millisec since midnigh (28 bits) and the days since 1/1/1984 (16 bits)
  - byte stream of undefined length (domain)
- Only a few entries of the OD are mandatory
  - [1000sub00], [1001sub00], [1018sub00], [1018sub01]
- EDS e.g.

Index	Object	Name	Data type	Access type
1000	Variable	device type	DWORD	ro
1001	Variable	error register	BYTE	ro
…	…	…	…	…
1006	Variable	communication cycle period	DWORD	rw
…	…	…	…	…

CANopen

Device configuration (service data objects / SDO)

- SDOs are used to read/write device OD entries
- Client/server, logical 1:1 channel
- Acknowledged service: each client SDO requires a server answer
- SDO protocol runs in two phases:
  - initialization phase: indication of the addressed OD entry and the length of the data to be transferred
  - second phase: actual data is transmitted in segments (7 bytes each)
- SDO services:
  - Initiate SDO Upload, Upload SDO Segment, Initiate SDO Download and Download SDO Segment
- Data transfers of up to 4 bytes may be carried out at the initialization phase (**expedited SDO transfer**)

CANopen

Device configuration (service data objects / SDO)

- Command byte:
  - download/upload, request/response, segmented/block/expedited transfer, number of data bytes, end indicator, alternating toggle bit for each consecutive segment transfer
- E.g. 1: SDO Download service, expedite transfer, to set a device´s heartbeat:
  - OD entry [1017], set to 4 seconds (in ms, UNSIGNED16 value, i.e. 0x0F A0)

	Command byte	OD main-index	OD sub-index	Data (max. 4 bytes)
Client request	**2B**	**17 10**	**00**	**A0 0F 00 00**
Server reply	**60**	**17 10**	**00**	**00 00 00 00**

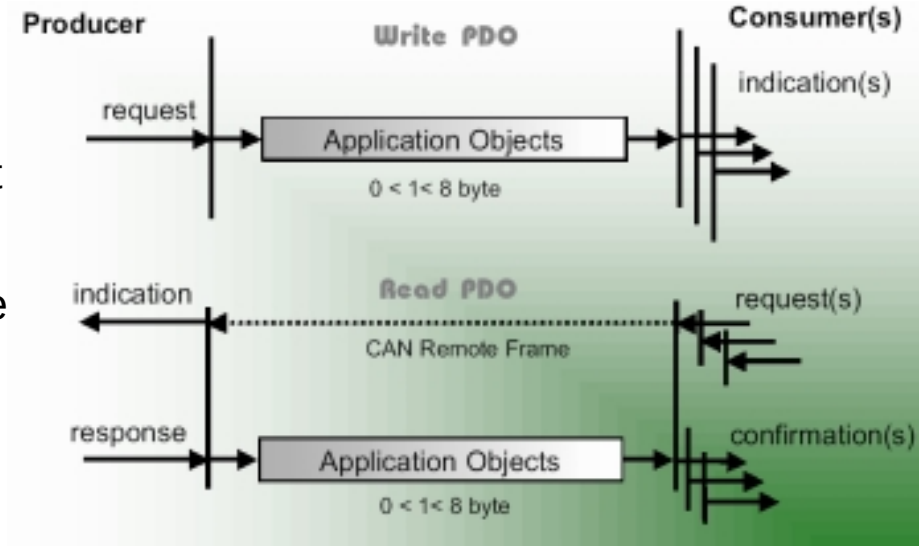
- E.g. 2: Upload service, reading the device's name (OD1008h, string):

```
40 08 10 00 00 00 00 00     // Initiate req: Read Device Name [1008]
41 08 10 00 1A 00 00 00     // Initiate resp: Fine. It's 26 bytes long
60 00 00 00 00 00 00 00     // Upload segment req, Toggle = 0
00 54 69 6E 79 20 4E 6F     // Upload segment resp, Toggle = 0
70 00 00 00 00 00 00 00     // Upload segment req, Toggle = 1
10 64 65 20 2D 20 4D 65     // Upload segment resp, Toggle = 1
....
```

CANopen

Process Data Objects (PDO)

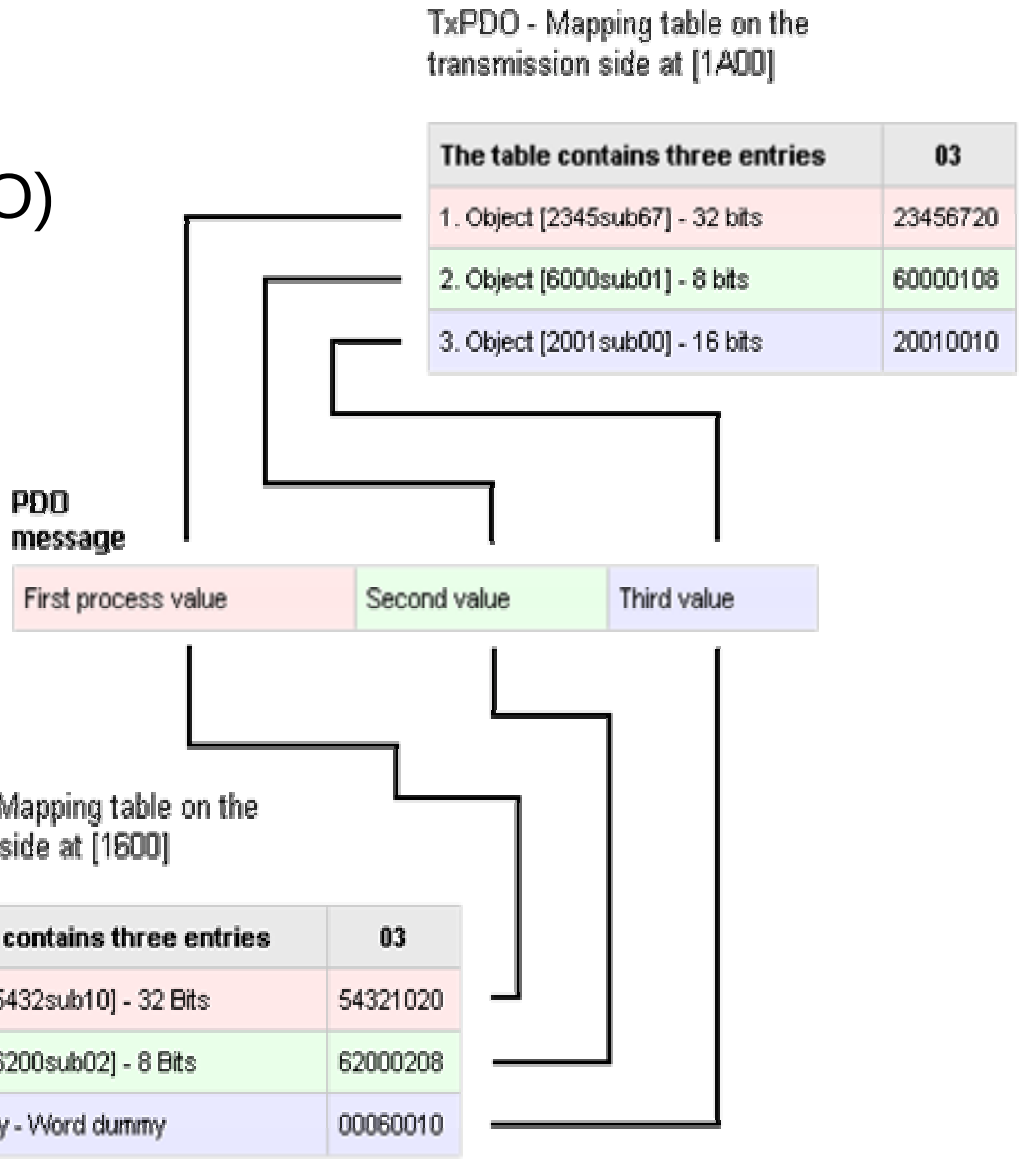
- ✓ Carry actual application data
- ✓ Broadcast, producer/consumer cooperation model
- ✓ **Unacknowledged**
- ✓ Unconstrained format (except for CAN ID range)
  - ✓ Agreement between producer and consumers (application specific)
- ✓ Two PDO services
  - ✓ Write PDO:
    - ✓ mapped to a single CAN packet
  - ✓ Read PDO:
    - ✓ mapped to a CAN remote frame
    - ✓ optional



CANopen

Process Data Objects (PDO)

- Each PDO is associated with
  - COB-ID: CAN identifier
  - PDO-mapping:
    - identifies which process variables are in the PDO data field
    - their order
    - their size
  - Communication parameters



CANopen

Process Data Objects (PDO)

- PDO communication parameters

Sub-Index	Contents	Data type
0	Largest sub-index supported	BYTE
1	COB-ID	DWORD
2	Type	BYTE
3	Inhibit time in ms	WORD
5	Event timer	WORD

  - COB-ID: associated CAN message ID
  - Type:
    - synchronous, asynchronous, ...
  - Inhibit time: for asynchronous PDOs imposes a minimum time between transmissions
  - Event timer: (if >0) causes the periodic transmission of asynchronous PDOs

CANopen

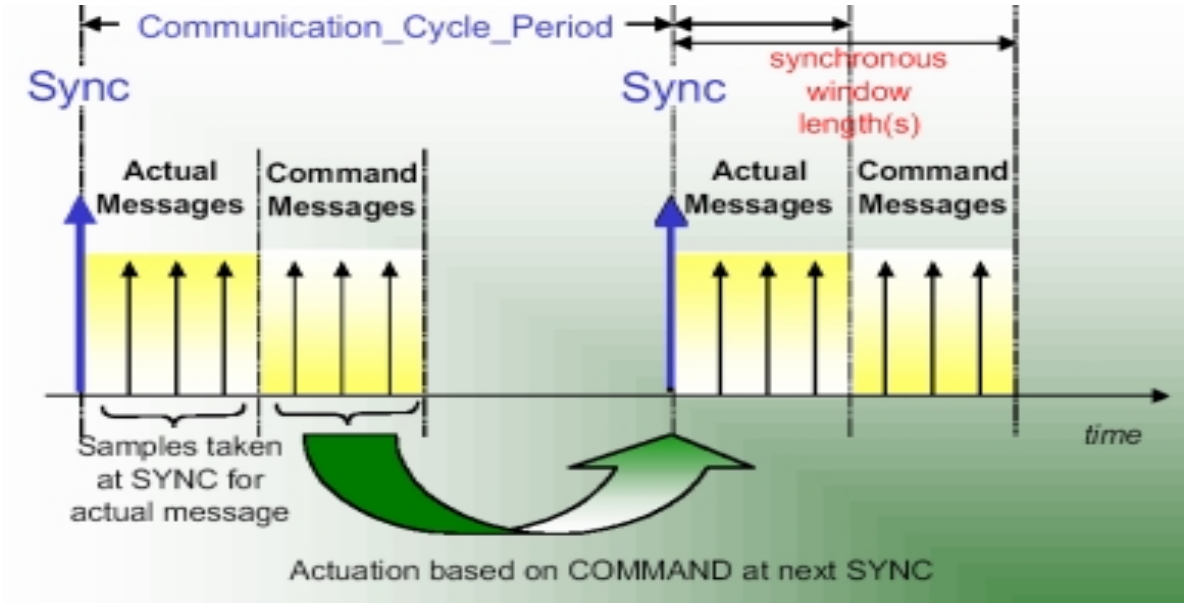
Process Data Objects (PDO)

- ***Asynchronous PDOs*** ***(***255 or 254 in the PDO type***)***

  event-controlled, automatically transmitted whenever at least one of the process variables mapped in a PDO is altered (e.g. an input value)
- ***Synchronous PDOs*** are transmitted after reception of a synchronization message (Sync Object).
  - Synchronous PDO transmission is carried out synchronously in the entire network
  - All device inputs are sampled on the arrival of the sync object
  - Sampled data transmitted after the next sync message (1 cycle constant delay)
  - Synchronous PDOs have values 1..240 in the PDO type. This number is used as a divider (e.g. if 2, transmissions occur in alternate cycles)
  - Acyclical synchronous PDO are txmitted after explicit application indication

CANopen

Process Data Objects (PDO)

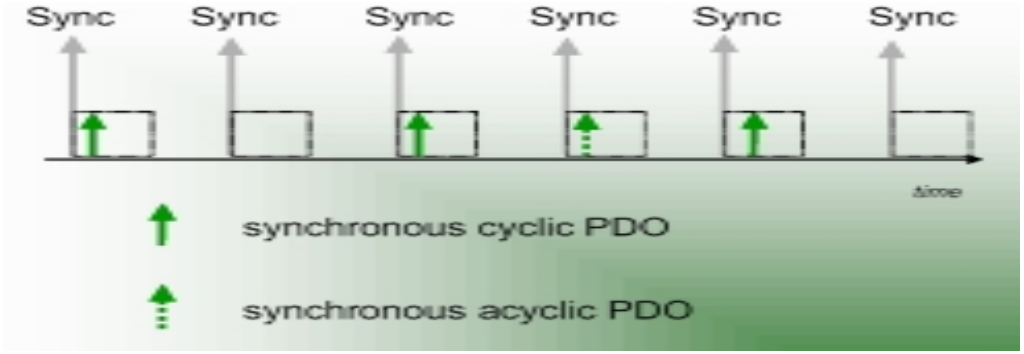


Quasi-simultaneous sampling

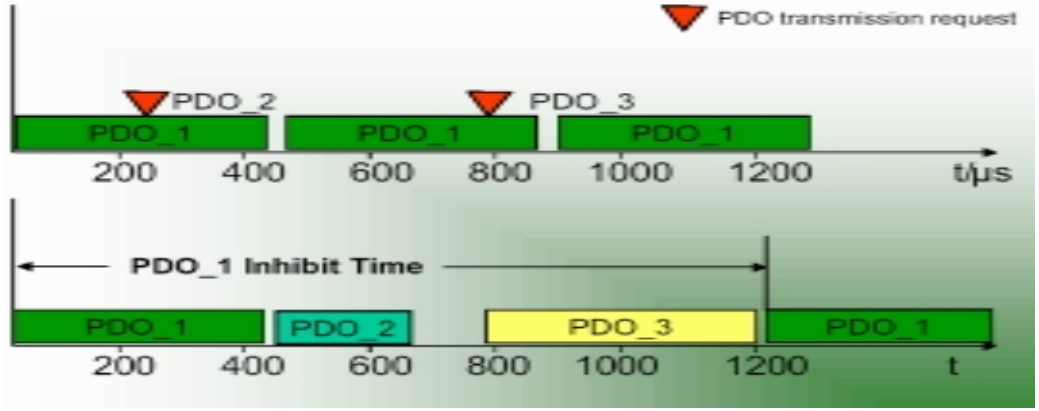
CANopen

Process Data Objects (PDO)

Acyclical synchronous transmissions



Avoiding startvation using Inhibit time (forcing a mit)



CANopen

Node Monitoring via Node-Guarding and Heartbeat Mesgs

- Communication status information about nodes provided by two mechanisms:
  - Cyclic querying of the node state by a NMT-Master
    - NMT master sends a remote frame to each system node requesting its current state
    - If a node fails to reply generates node-guarding event on the NMT
    - Mechanism based on low-priority messages (ID=1972+node-ID)
  - Heartbeat
    - Nodes transmit its communication status at regular intervals
    - Heartbeat interval is configurable (OD 1017h)
    - heartbeat consumer time is configured at OD entry 1016h
- In heartbeat or guarding messages nodes transmit its communication status:
  - 0x00 - Bootup; 0x04 - Stopped; 0x05 - Operational; 0x7F - Pre-Operational

CANopen

Emergency

- ✓ Node monitoring only conveys the communication state, not the actual node status
- ✓ Nodes require a high priority CAN identifier to indicate error situations
- ✓ CAN ID of the error message registered at OD 1014 (optional)

Error code	Error register	Vendor specific error field

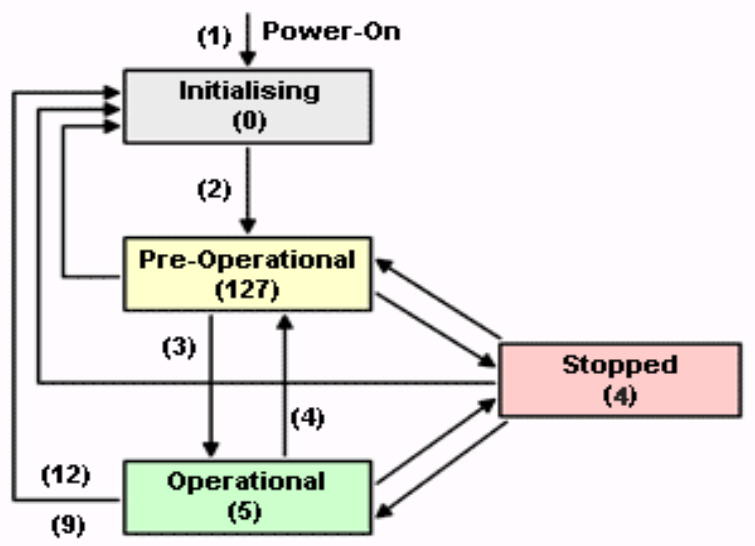
Error frame, 8 bytes

Error code (hex)	Error description
00xx	Error Reset / No Error
10xx	Generic Error
2xxx	Current
3xxx	Voltage
4xxx	Temperature
50xx	Device Hardware
6xxx	Device Software
70xx	Additional Modules
8xxx	Monitoring
90xx	External Error
F0xx	Additional Functions
FFxx	Device Specific

CANopen

CANopen network management (NMT)

- CANopen provides network management services for
  - control of the communication state of network nodes
  - node monitoring
- Node states:
  - Initialization: hardware initialization, reset of device parameters, communication set-up
  - Pre-operational: used for configuration (no PDO transactions)
  - Operational: PDO transactions enabled
  - Stopped: only heratbeat/guard messages can be txmitted

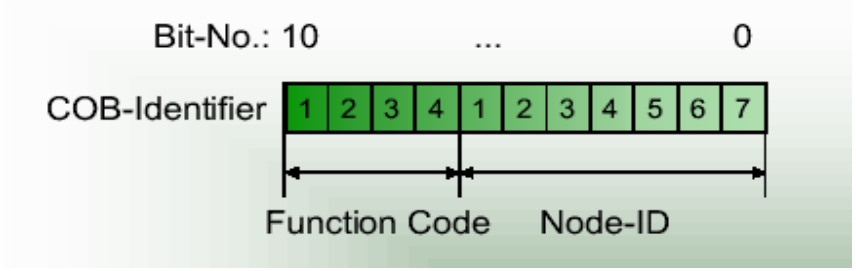


- Network master may change the node's state, either individual or globally
  - ID 0, two bytes: state + node number (0 means all)

CANopen

Predefined use of message identifiers

- CANopen supports a predefined allocation of message identifiers (Predefined Connection Set)
  - Allows to operate simple systems, up to 127 nodes, without reconfiguration
  - Comprises:
    - One emergency message
    - Synchronization and time stamp messages
    - One SDO-connection per device
    - The NMT-messages for node control and node monitoring
    - Up to 4 transmit and 4 receive PDOs per device
  - CANopen networks allow a maximum of 127 nodes
  - Rely on 11bit CAN ID to specify the COB-ID



  - Each network function requires a different ID

CANopen

Predefined connection set

Communication object	COB-ID(s) hex	Slave nodes
NMT node control	000	Receive only
Sync	080	Receive only
Emergency	080 + NodeID	Transmit
TimeStamp	100	Receive only
PDO	180 + NodeID 200 + NodeID 280 + NodeID 300 + NodeID 380 + NodeID 400 + NodeID 480 + NodeID 500 + NodeID	1. Transmit PDO 1. Receive PDO 2. Transmit PDO 2. Receive PDO 3. Transmit PDO 3. Receive PDO 4. Transmit PDO 4. Receive PDO
SDO	580 + NodeID 600 + NodeID	Transmit Receive
NMT node monitoring (node guarding/heartbeat)	700 + NodeID	Transmit
LSS	7E4 7E5	Transmit Receive

CANopen

Layer Setting Services (LSS) (optional)

- Configuration, by software, of the node ID and baudrate
- 1:1 connection required
  - LSS messages are always 8 byte
  - Messages to device use COB-ID 0x7E5
  - Device replies using COB-ID 0x7E4
- Sequence of operations
  - Switch mode, Inquire node ID, Configure node ID, Configure bit timing parameters
- Baudrate according to a standardized table
- Baudrate scanning
  - When the baudrate is not known in advance each possible configuration must be tried until success

Summary:

- **CANopen**
  - **CAN High Layer Protocol**
  - **Object-oriented Modeling of Device and Network**
  - **Interoperability between Devices**
  - **Interchangeability of Devices**
  - **Off-the-shelf Plug-and-play Capability**
  - **Off-the-shelf Configuration and Analysis Tools (not addressed)**
  - **Standardized Communication Services**
    - Peer-to-peer communication
    - Segmented Data Transfer
  - **Network and Node Configuration**
  - **Network and Node Error Handling**

Summary:

- ✓ **CANopen**
  - ✓ Protocols:
    - ✓ Process Data Object (PDO) Protocol
    - ✓ Service Data Object (SDO) Protocols
    - ✓ Synchronization (SYNC) Protocol
    - ✓ Time Stamp (TIME) Protocol (not addressed)
    - ✓ Emergency (EMCY) Protocol
    - ✓ Network Management Protocols:
      - NMT Message Protocol
      - Boot-Up Protocol
      - Error Control Protocol