SIEMENS Healthineers

CAN and CANopen

Theory and practice

Roman Fedoryak / Viktor Timkov
February 2023

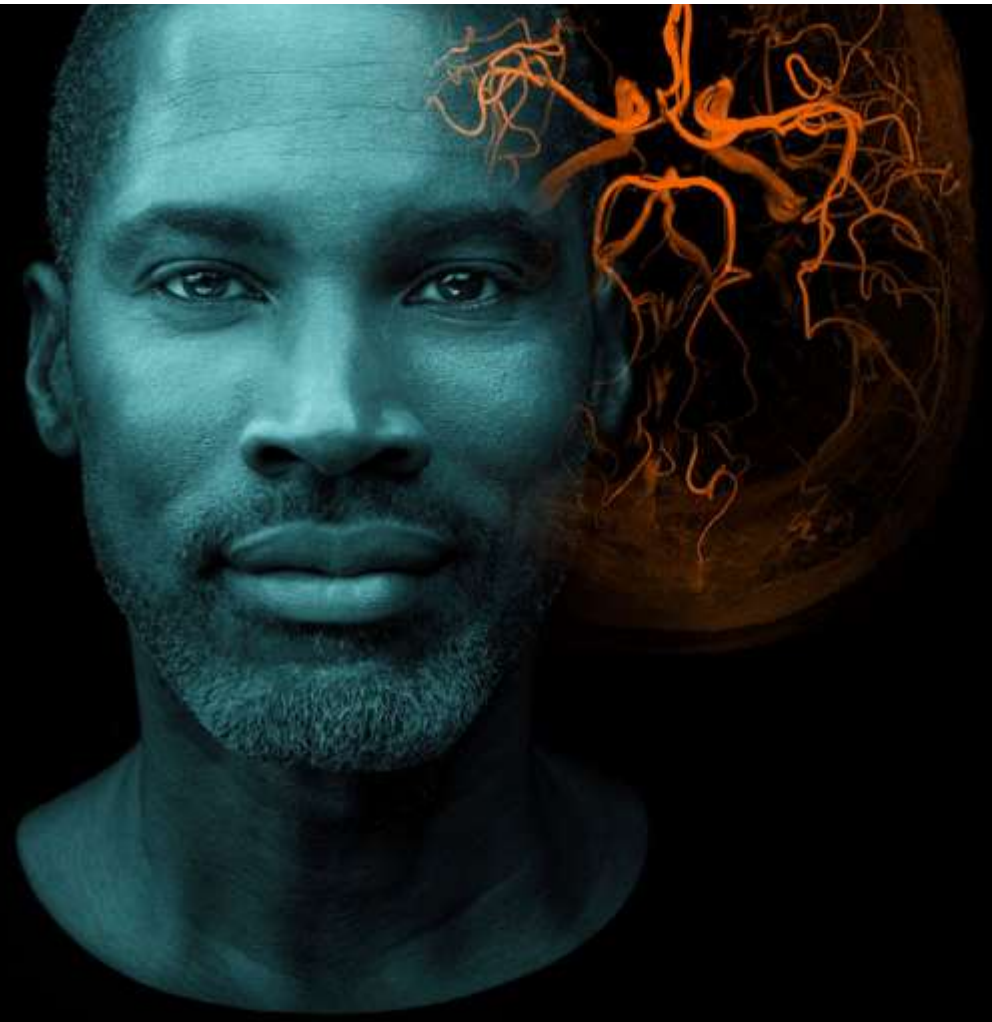


Table of Contents

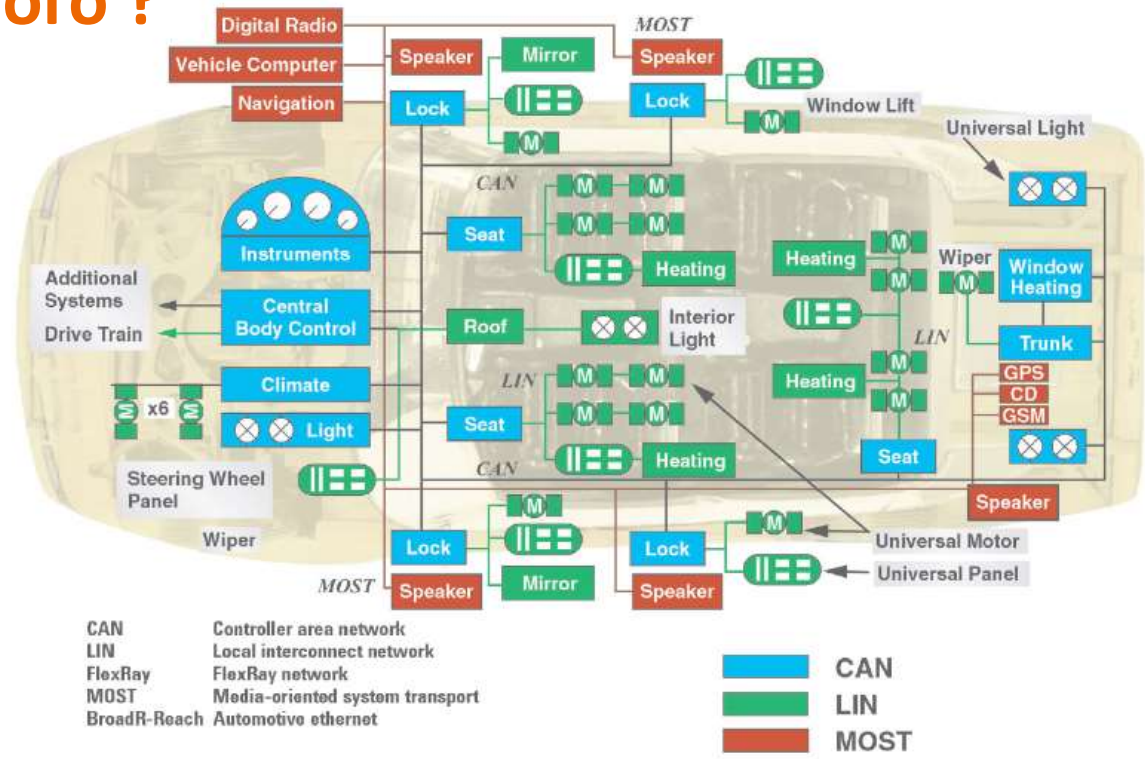
- **Controller Area Network (CAN)**
  - Для чого потрібен CAN
  - Опис фізичного рівня
  - Рецесивні та домінантні біти
  - Bit stuffing
  - Фрейм (телеграма)
  - Арбітраж
  - Контроль помилок

- **CANopen**
  - Сервіси CANopen
  - Концепт Словнику Об’єктів Objects and Dictionary
  - SDO – Service Data Object
  - PDO – Process Data Object
  - NMT – Network Management
  - Heartbeat, Bootup

Controller Area Network (CAN) – для чого ?

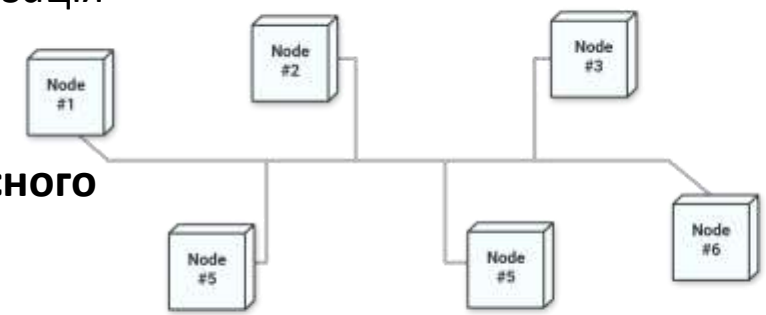
Вимоги до комунікації між контролерами (вузлами):

- Комунікація в реальному часі = прогнозованість часу доставки
- Найпростіша реалізація – 2 проводи (звита пара), шинна топологія
- Захист даних, визначення помилок, автоматична ретрансмісія при помилках
- Multi-Master = кожен вузол може ініціювати передачу
- Арбітраж при намаганні декількома вузлами вести передачу одночасно, пріоритизація даних при обміні

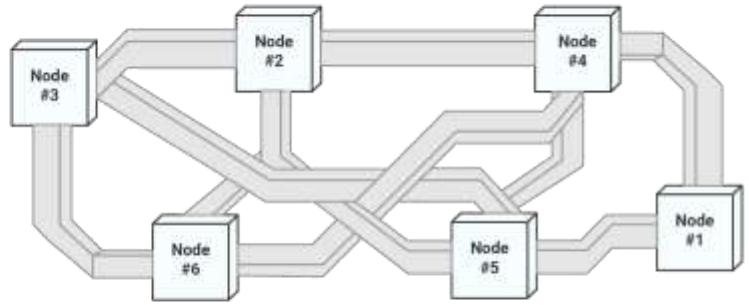


Стандартний список інтерфейсів сучасного мікроконтролера:
UART, I2C, SPI, CAN, USB, Ethernet

CAN Difference



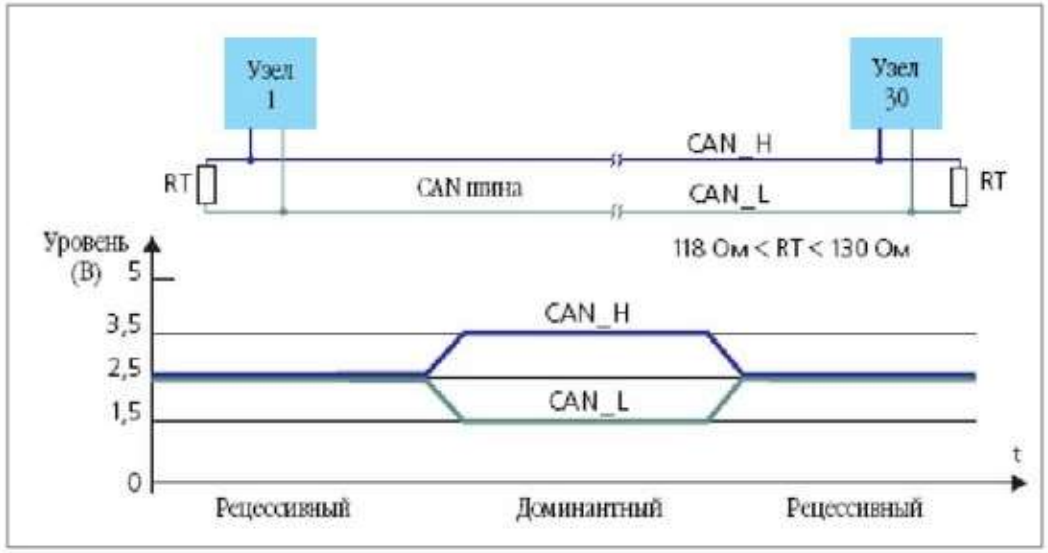
With CAN System



Without CAN System

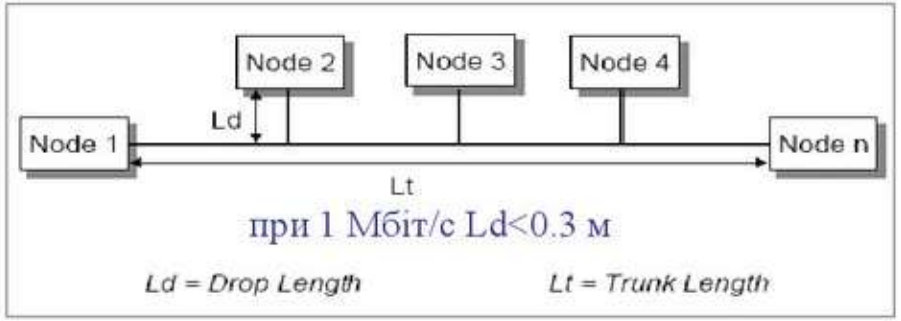
Фізичний рівень CAN

Фізичний рівень (ISO 11898-2)

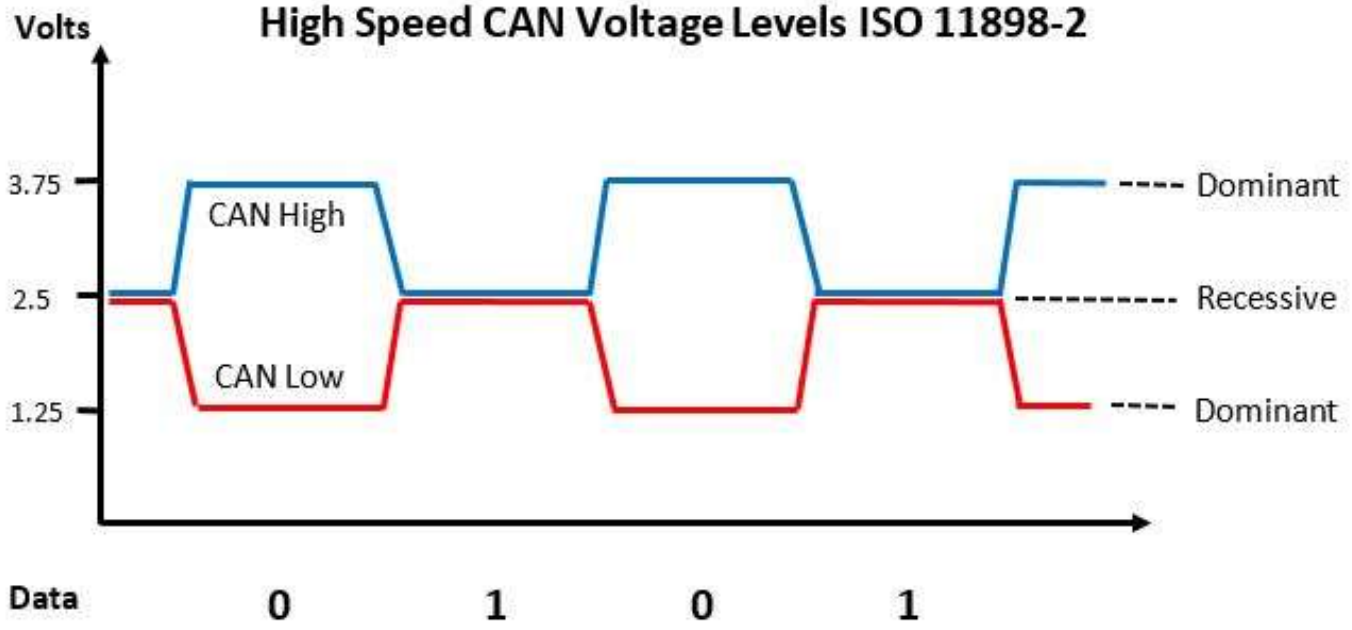


Bit Rate	Bus Length	Nominal Bit-Time
1 Mbit/s	30 m	1 µs
800 kbit/s	50 m	1,25 µs
500 kbit/s	100 m	2 µs
250 kbit/s	250 m	4 µs
125 kbit/s	500 m	8 µs
62,5 kbit/s	1000 m	20 µs
20 kbit/s	2500 m	50 µs
10 kbit/s	5000 m	100 µs

- симетрична (диференційна) передача по напрузі CAN_H між CAN_L, вита пара 120 Ом;
- можливість подачі живлення окремою парою проводів CAN_V+ GND;
- топологія – шина, с короткими відгалуженнями;
- довжина лінії до 1000 м;
- бітова швидкість від 10 кбіт/с до 1 Мбіт/с, підтримка пристроєм 20 Кбіт/с – обов'язкова;
- два термінальних резистори 120 Ом (108-132) в кінцях лінії;
- рівні для домінантного та рецесивного біта;
- максимум 64 пристрої на сегмент



Рецесивні та домінантні біти



Замість використання двійкових значень як «0» та «1» в специфікації CAN введені терміни:

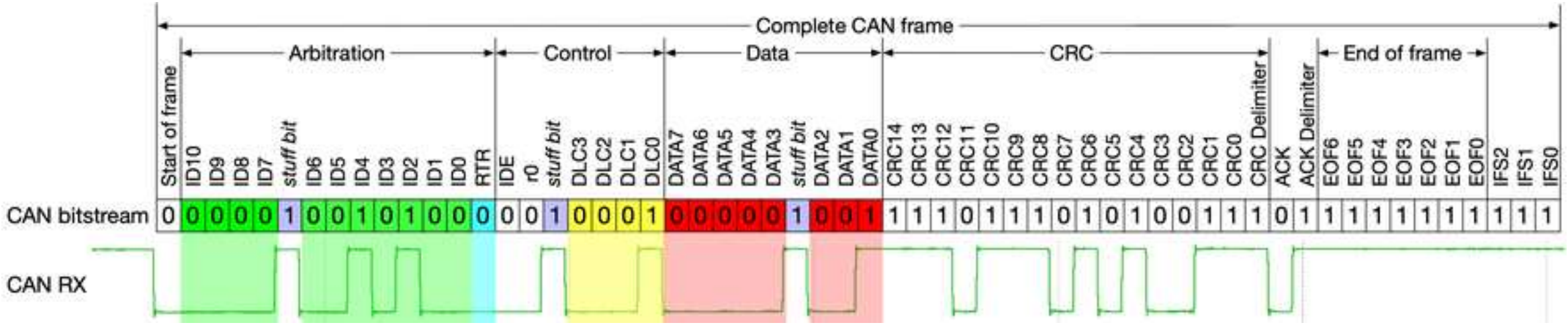
- Домінантний біт (0)
- Рецесивний біт (1)

Домінантний біт завжди перемагає (домінує) над рецесивним

Якщо шина в рецесивному стані, перевести її в домінантне може будь-який вузол.
Якщо шина в домінантному стані, перевести її в рецесивне не зможе ніякий вузол.

Фрейм (телеграма)

Data Frame, Remote Frame



Field name	Length (bits)	Purpose
Start-of-frame	1	Denotes the start of frame transmission
Identifier (green)	11	A (unique) identifier which also represents the message priority
Stuff bit	1	A bit of the opposite polarity to maintain synchronisation; see Bit stuffing, below
Remote transmission request (RTR) (blue)	1	Must be dominant (0) for data frames and recessive (1) for remote request frames
Identifier extension bit (IDE)	1	Must be dominant (0) for base frame format with 11-bit identifiers
Reserved bit (r0)	1	Reserved bit. Must be dominant (0), but accepted as either dominant or recessive.
Data length code (DLC) (yellow)	4	Number of bytes of data (0–8 bytes)[a]
Data field (red)	0–64 (0-8 bytes)	Data to be transmitted (length in bytes dictated by DLC field)
CRC	15	Cyclic redundancy check
CRC delimiter	1	Must be recessive (1)
ACK slot	1	Transmitter sends recessive (1) and any receiver can assert a dominant (0)
ACK delimiter	1	Must be recessive (1)
End-of-frame (EOF)	7	Must be recessive (1)
Inter-frame spacing (IFS)	3	Must be recessive (1)

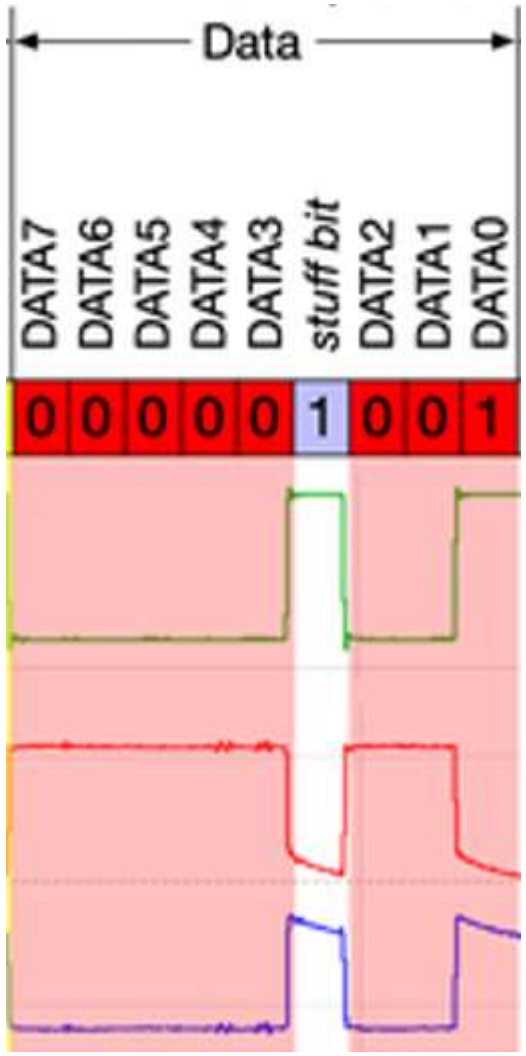
Bit Stuffing

Bit Stuffing:
Біт протилежної полярності додається після 5 послідовних бітів однієї полярності

Послідовність з доданими передавачем стафф-бітами розшифровує приймач, видаляючи стафф-біти

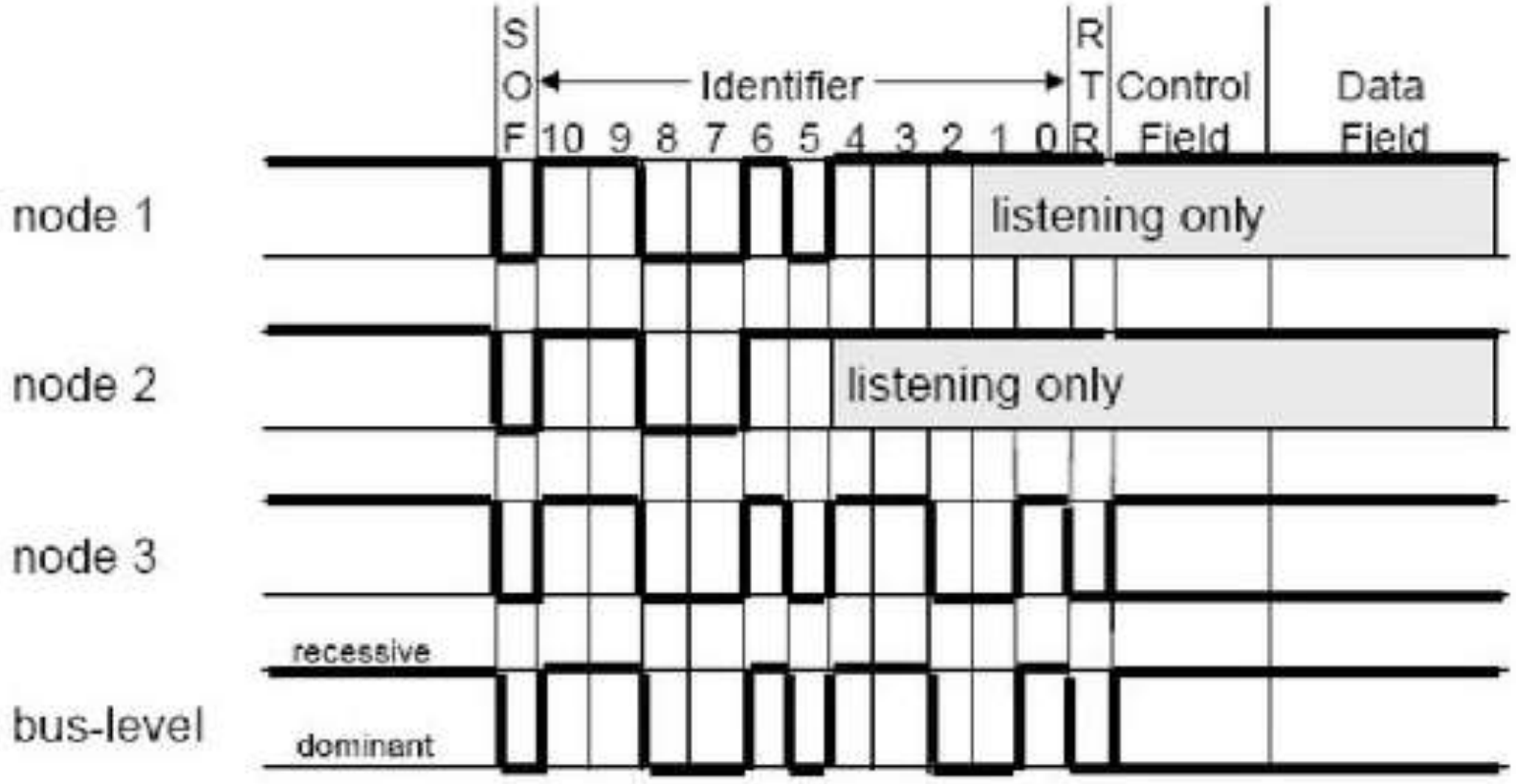
(Не всі частини телеграми стаффляться – CRC delimitier, ACK, EOF are not stuffed)

6 послідовних домінантних бітів – це флаг помилки (див. нижче Error Frame)



Арбітраж

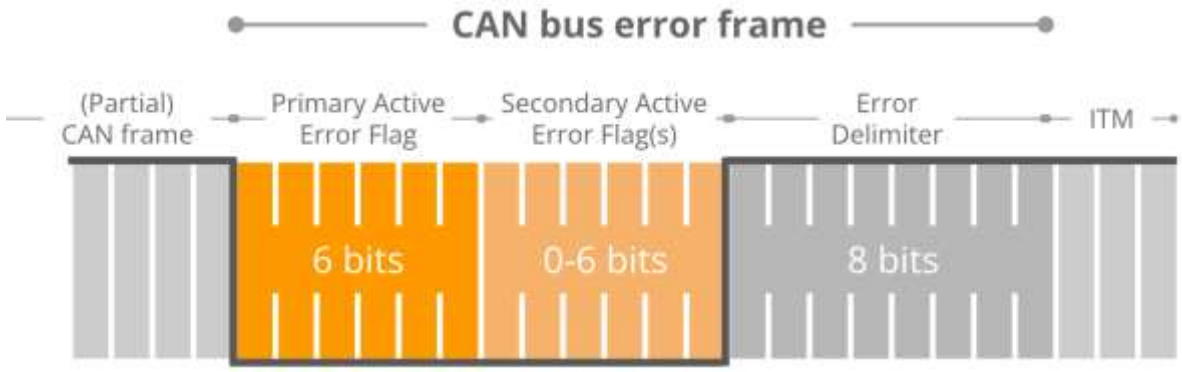
Канальний рівень. Метод доступу (BOSH CAN 2.0A/B, ISO 11898-1)



Вузол, що передавав рецесивний біт в той час коли на шині знаходиться домінантний біт переданий іншим вузлом, програє арбітраж та переходить в пасивний режим.

Пріоритет фрейму тим вищий,
чим ближчий до нуля його ID !!!

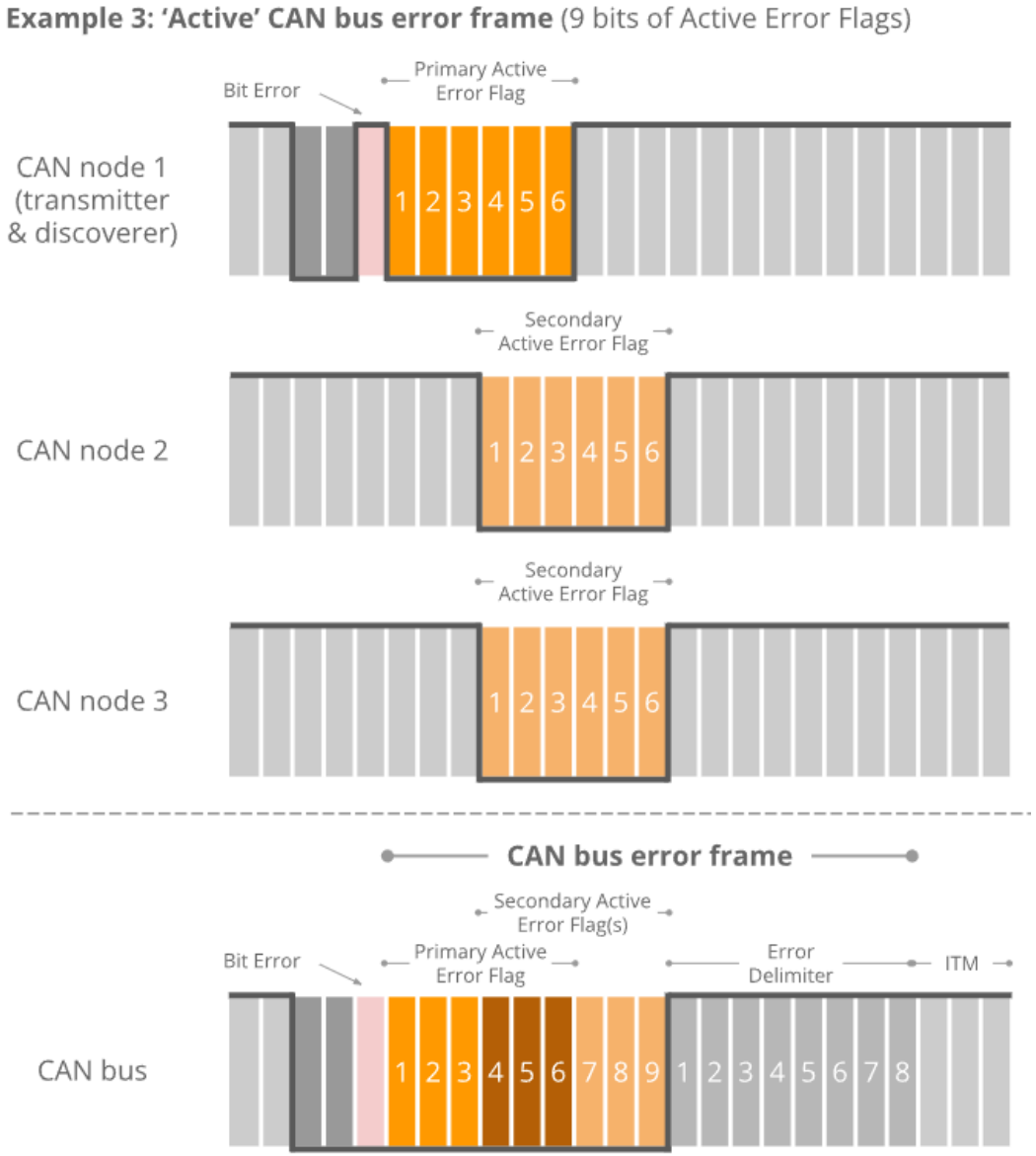
Error Frame



Example 3: 9 bits of error flags

Here, CAN node 1 has already transmitted a sequence of 3 dominant bits when it discovers a Bit Error and begins sending 6 dominant bits.

Once halfway through the primary Active Error Flag, nodes 2 and 3 recognize the Bit Stuffing Error (due to the 3 initial dominant bits being followed by another 3 dominant bits) and they begin raising their error flags. The result is that the sequence of dominant bits from error flags becomes 9 bit long.



Фрейм (телеграма)

SIEMENS Healthineers

Overload Frame

Майже не використовуються !

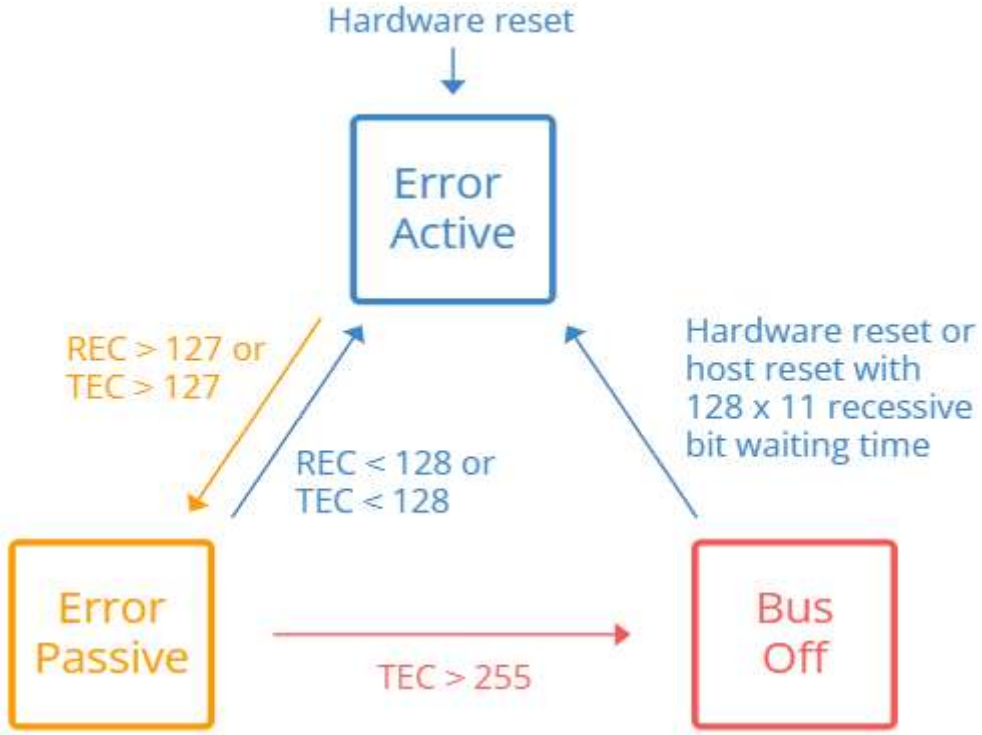
Контроль помилок

Error frames, **error types**, **counters** and **states**.

CAN Error Types:

1. Bit Error [Transmitter]
2. Bit Stuffing Error [Receiver]
3. Form Error [Receiver]
4. ACK Error [Transmitter]
5. CRC Error [Receiver]

1	Bit Error	Node transmits a dominant/recessive bit, but reads back the opposite logical level
2	Bit Stuffing Error	Node detects a sequence of 6 bits of the same logical level between the SOF and CRC
3	Form Error	Node detects a bit of an invalid logical level in the SOF/EOF fields or ACK/CRC delimiters
4	ACK Error	Node transmits a CAN message, but the ACK slot is not made dominant by receiver(s)
5	CRC Error	Node calculates a CAN message CRC that differs from the transmitted CRC field value



TEC: Transmit Error Counter

REC: Receive Error Counter

TEC +8	Transmitter raises primary error flag
REC +8	Receiver raises primary error flag
REC +1	Receiver raises secondary error flag
REC -1	Receiver successfully receives message
TEC -1	Transmitter successfully sends message

Q&A - CAN

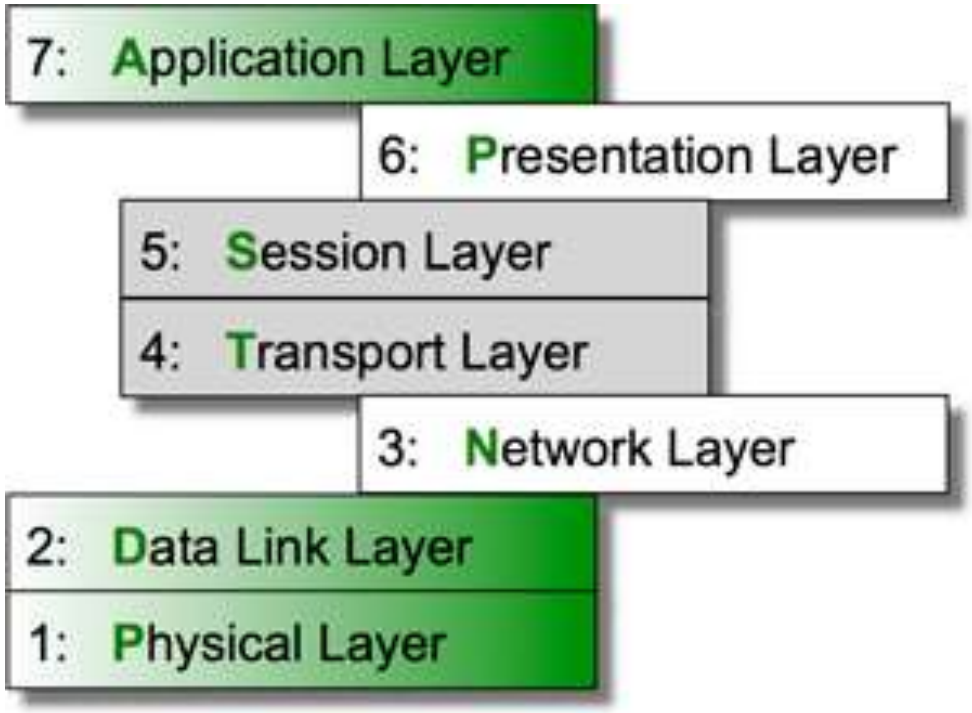
- **Questions so far? (CAN only related questions)**

CANopen

Developer: CiA (CAN in Automation) www.can-cia.org

Standard documents:

CiA 301 - **Application Layer and Communication Profile**



Device Profile Specification

- CiA DSP-401: I/O Modules
- CiA DSP-402: Drives and Motion Control
- CiA DSP-403: Human Machine Interface
- CiA WD-404: Measuring Devices and Closed-Loop Controllers
- CiA DSP-406: Encoders
- CiA WD-408: Proportional Hydraulic Valves
- CiA WD-409: Door Control (Railways)
- CiA WDP-4XX: Brake Control (Railways)
- CiA WDP-4XX: Train Bus Gateways

Under development are device profiles for diesel engines, maritime-specific modules and medical-specific systems.

Interface Profile Specification

- CiA DSP-405: IEC 1131 Programmable Devices

Application Framework Specification

- CiA WD-407: Public Transportation

© CiA

Сервіси CANopen

#1 Network Management (NMT)

#2 Synchronization (SYNC)

#3 Emergency (EMCY)

#4 Timestamp (TIME) [PDO]

#5 Process Data Object [PDO]

#6 Service Data Object [SDO]

#7 Node monitoring (Heartbeat) [SDO]

Робота з об’єктами OD
Predefined Connection Set

Для доступу до об’єктів та сервісів CANopen розроблена єдина схема розподілу ідентифікаторів (Basic CAN, 11 bit)

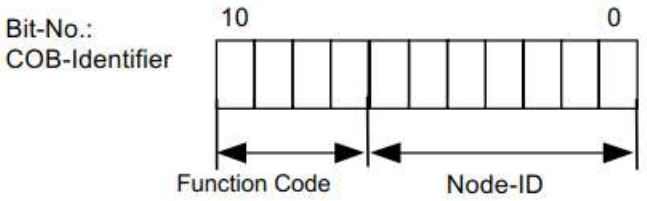


Figure 50: Identifier allocation scheme for the pre-defined connection set

Node-ID: 7 bit 1 ~ 127 (0x7F), 0 = broadcast

Table 33: Broadcast Objects of the Pre-defined Connection Set

object	function code (binary)	resulting COB-ID	Communication Parameters at Index
NMT	0000	0	-
SYNC	0001	128 (80h)	1005h, 1006h, 1007h
TIME STAMP	0010	256 (100h)	1012h, 1013h

Table 34: Peer-to-Peer Objects of the Pre-defined Connection Set

object	function code (binary)	Resulting COB-IDs	Communication Parameters at Index
EMERGENCY	0001	129 (81h) – 255 (FFh)	1014h, 1015h
PDO1 (tx)	0011	385 (181h) – 511 (1FFh)	1800h
PDO1 (rx)	0100	513 (201h) – 639 (27Fh)	1400h
PDO2 (tx)	0101	641 (281h) – 767 (2FFh)	1801h
PDO2 (rx)	0110	769 (301h) – 895 (37Fh)	1401h
PDO3 (tx)	0111	897 (381h) – 1023 (3FFh)	1802h
PDO3 (rx)	1000	1025 (401h) – 1151 (47Fh)	1402h
PDO4 (tx)	1001	1153 (481h) – 1279 (4FFh)	1803h
PDO4 (rx)	1010	1281 (501h) – 1407 (57Fh)	1403h
SDO (tx)	1011	1409 (581h) – 1535 (5FFh)	1200h
SDO (rx)	1100	1537 (601h) – 1663 (67Fh)	1200h
NMT Error Control	1110	1793 (701h) – 1919 (77Fh)	1016h, 1017h

Концепт Словнику Об’єктів Object Dictionary (OD)

- **Об’єкт – дані, з іменем, типом та атрибутами, описується такою табличною структурою:**

Index (hex)	Object (Symbolic Name)	Name	Type	Attrib.	M/O

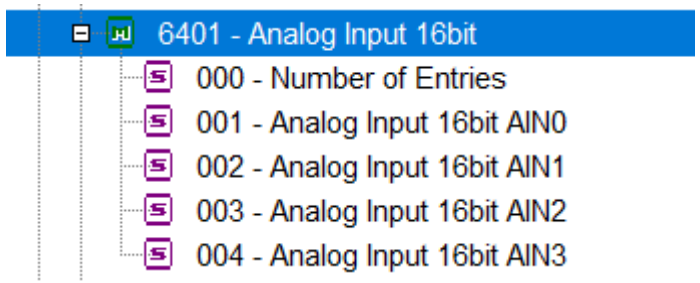
- Index (Subindex) – позиція всередині словника OD
- Symbolic Name – використовується як ім’я змінної в С
- Name – короткий опис функціоналу
- Attribute – права доступу до об’єкту: RW, RO, WO, Const
- M/O – Mandatory (required by standard) or Optional

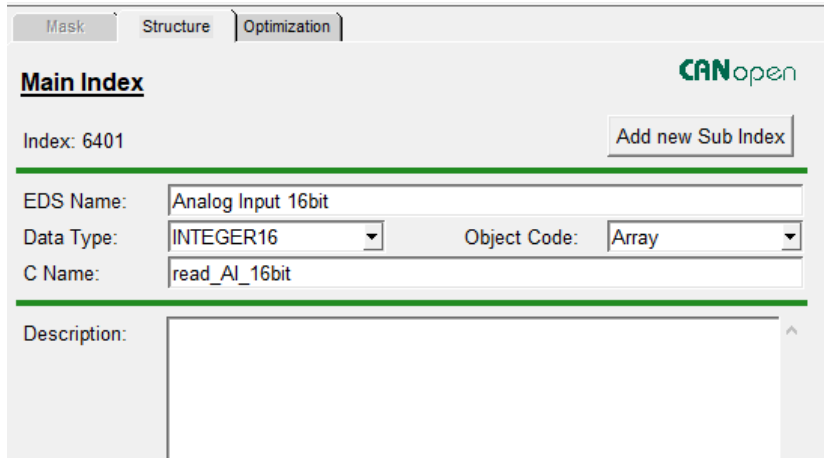
- **Структура OD**

Index (hex)	Object
0000	not used
0001-001F	Static Data Types
0020-003F	Complex Data Types
0040-005F	Manufacturer Specific Complex Data Types
0060-007F	Device Profile Specific Static Data Types
0080-009F	Device Profile Specific Complex Data Types
00A0-0FFF	Reserved for further use
1000-1FFF	Communication Profile Area
2000-5FFF	Manufacturer Specific Profile Area
6000-9FFF	Standardised Device Profile Area
A000-BFFF	Standardised Interface Profile Area
C000-FFFF	Reserved for further use

Концепт Словнику Об’єктів
Object Dictionary (OD)

Приклади опису об’єктів (демонстрація in live)

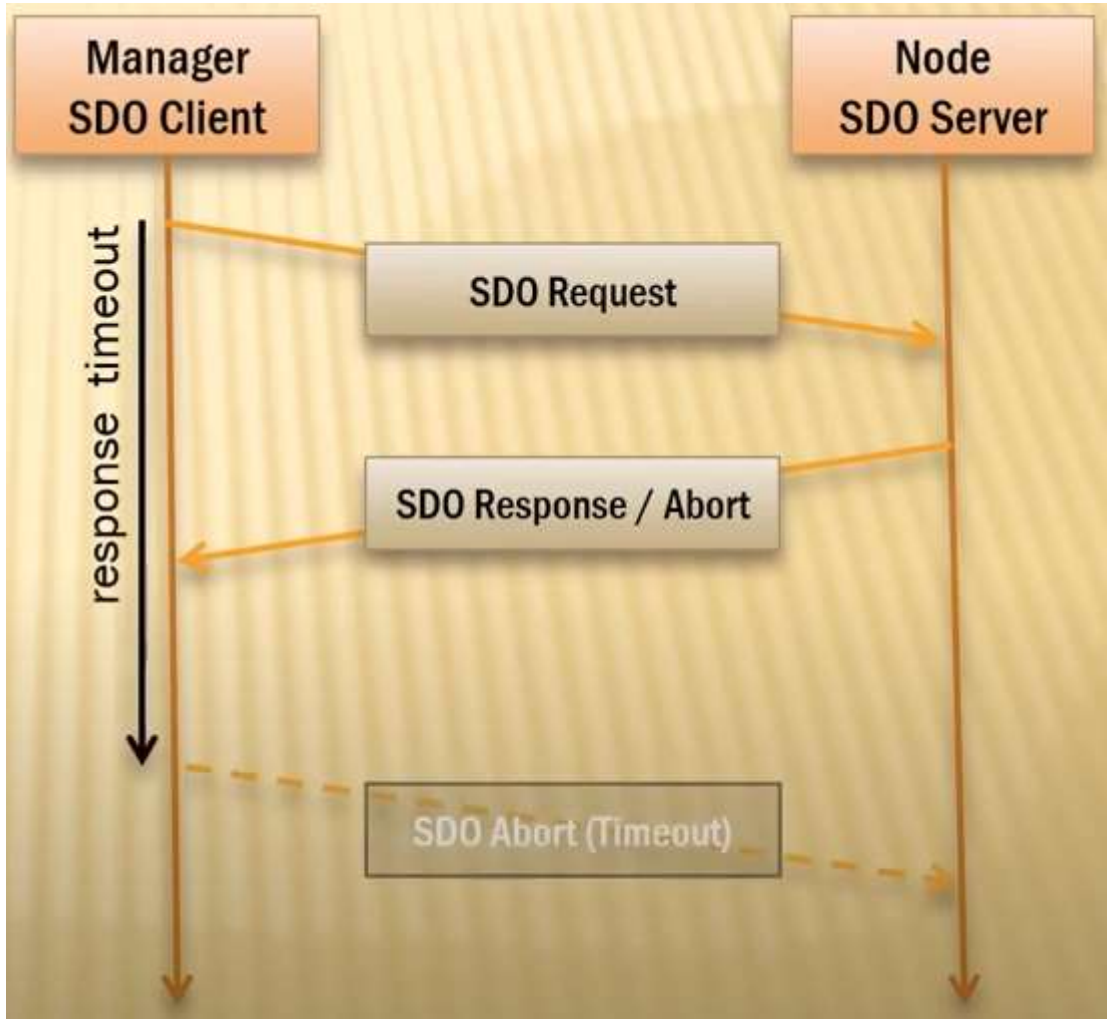




Файли OD – типи EDS, XDD (демонстрація in live)

Опис словника об’єктів в С коді (демонстрація in live)

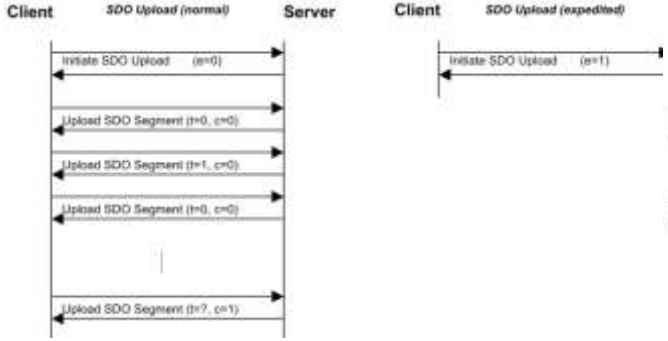
Сервіси CANopen:
SDO – Service Data Object

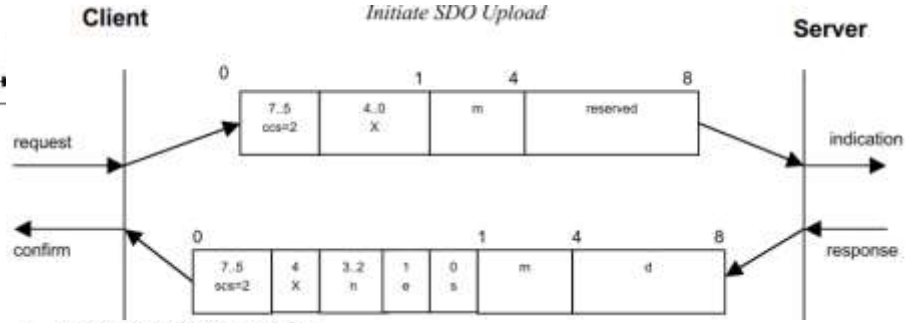


Сервіси CANopen:
SDO – Service Data Object

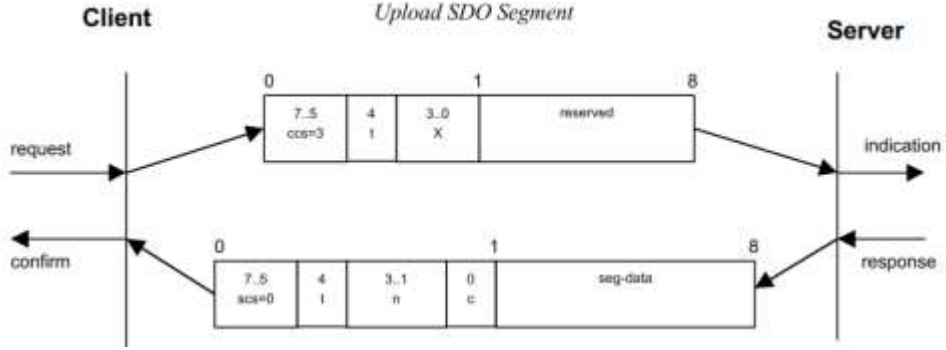
Six confirmed services (SDO Download, SDO Upload, Initiate SDO Upload, Initiate SDO Download, Download SDO Segment, and Upload SDO Segment) + one unconfirmed service (Abort SDO Transfer) are defined for Service Data Objects doing the **standard segmented/expedited transfer.**

Eight confirmed services (SDO Block Download, SDO Block Upload, Initiate SDO Upload, Initiate SDO Block Download, Download SDO Segment, Upload SDO Segment, End SDO Upload and End SDO Block Download) + one unconfirmed service (Abort SDO Block Transfer) are defined for Service Data Objects doing the **optional block Transfer**.





- **ccs:** client command specifier
  2: initiate upload request
- **scs:** server command specifier
  2: initiate upload response
- **n:** Only valid if e = 1 and s = 1, otherwise 0. If valid it indicates the number of bytes in d that do not contain data. Bytes [8-n, 7] do not contain segment data.
- **e**: transfer type
  0: normal transfer
  1: expedited transfer
- **s:** size indicator
  0: data set size is not indicated
  1: data set size is indicated
- **m**: multiplexor. It represents the index/sub-index of the data to be transfer by the SDO.
- **d:** data
  e = 0, s = 0: d is reserved for further use.
  e = 0, s = 1: d contains the number of bytes to be uploaded. Byte 4 contains the lsb and byte 7 contains the msb.
  e = 1, s = 1: d contains the data of length 4-n to be uploaded, the encoding depends on the type of the data referenced by index and sub-index
  e = 1, s = 0: d contains unspecified number of bytes to be uploaded.
- **X:** not used, always 0
- **reserved:** reserved for further use , always 0



- **ccs:** client command specifier
  3: upload segment request
- **scs:** server command specifier
  0: upload segment response
- **t:** toggle bit. This bit must alternate for each subsequent segment that is uploaded. The first segment will have the toggle-bit set to 0. The toggle bit will be equal for the request and the response message.
- **c:** indicates whether there are still more segments to be uploaded.
  0: more segments to be uploaded
  1: no more segments to be uploaded
- **seg-data:** at most 7 bytes of segment data to be uploaded. The encoding depends on the type of the data referenced by index and sub-index
- **n:** indicates the number of bytes in seg-data that do not contain segment data. Bytes [8-n, 7] do not contain segment data. n = 0 if no segment size is indicated.
- **X:** not used, always 0
- **reserved:** reserved for further use, always 0

Сервіси CANopen:
SDO - сервіс доступу до об’єктів OD вузла мережі

Практичний приклад читання об’єктів: (**Node-ID = 0x1C :** 0x61C = 0x600 + Node_ID, 0x59C = 0x580 + Node_ID)

1017 : Producer Heartbeat Time

```
Transmit: 0x61c 8-Data: 40 17 10 00 00 00 00 00  |@       |SDO (rx) INI UPL REQ | Producer heartbeat time
Receive:  0x59c 8-Data: 4b 17 10 00 d0 07 00 00  |K       |SDO (tx) ini upl rsp | Producer heartbeat time=0x07d0 (2000 dec)
```

100A : Manufacturer Software Version - Segmented Transfer case

```
Transmit: 0x61c  8-Data: 40 0a 10 00 00 00 00 00  |@       |SDO (rx) INI UPL REQ | manufacturer software version
Receive:  0x59c  8-Data: 41 0a 10 00 14 00 00 00  |A       |SDO (tx) ini upl rsp | manufacturer software version DataLen=20
Transmit: 0x61c  8-Data: 60 00 00 00 00 00 00 00  |`       |SDO (rx) UPL SEG REQ |
Receive:  0x59c  8-Data: 00 56 41 30 30 41 20 41  | VA00A A|SDO (tx) upl seg rsp | StringData: VA00A A
Transmit: 0x61c  8-Data: 70 00 00 00 00 00 00 00  |p       |SDO (rx) UPL SEG REQ |
Receive:  0x59c  8-Data: 10 70 70 6c 69 63 61 74  | pplicat|SDO (tx) upl seg rsp | StringData: pplicat
Transmit: 0x61c  8-Data: 60 00 00 00 00 00 00 00  |`       |SDO (rx) UPL SEG REQ |
Receive:  0x59c  8-Data: 03 69 6f 6e 00 00 00 00  | ion    |SDO (tx) upl seg rsp | StringData: ion
```

Сервіси CANopen:
PDO – Process Data Object

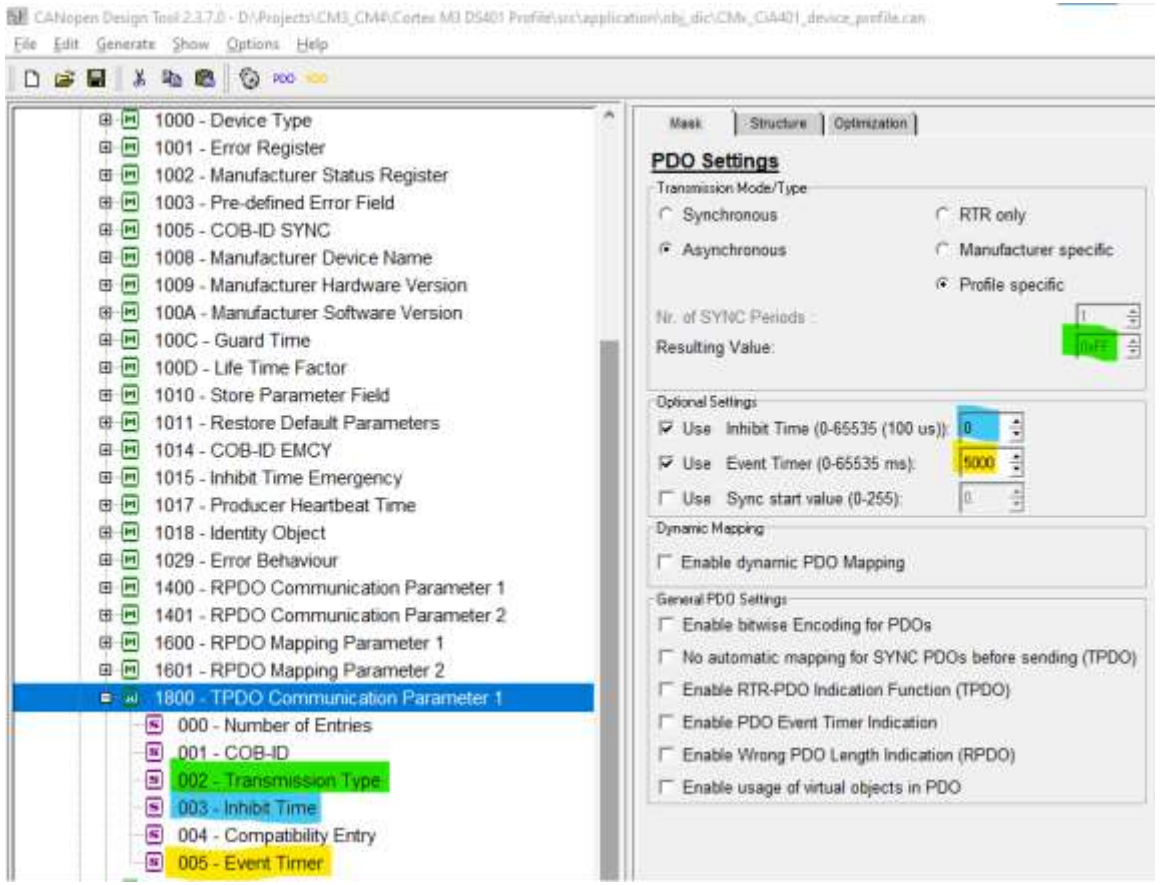
- Дозволяє передавати декілька об’єктів зі словника (OD) в одній телеграмі
- Multicast – передача можлива одночасно декільком вузлам
- Можливі тригери передачі PDO:
  - Application SW
  - Time driven (cyclic PDO)
  - Synchronous (SYNC related)
  - Event (COS = change-of-state)
  - Combination of abovementioned

Сервіси CANopen:
PDO – Process Data Object

PDO Communication Parameters:

Transmit PDO Parameters:
- TPDO1…TPDO512
  - Index 1800h…19FFh

Receive PDO Parameters
RPDO1…RPDO512
Index 1400h…15FFh



Сервіси CANopen: PDO – Process Data Object

- **Time trigger – встановлюється значенням параметру**
  Index 1800h…19FFh, Subindex 5 (Event Timer) = наприклад 5000 ms (0x1388)
  NodeID=0x1C, TPDO1 COB_ID = 0x180. 0x19C=COB_ID + NodeID

```
R0 01:21:56,7936: 0x19c  1-Data: ff                    |        |PDO1 (tx)
R0 01:22:01,7935: 0x19c  1-Data: ff                    |        |PDO1 (tx)
R0 01:22:06,7935: 0x19c  1-Data: ff                    |        |PDO1 (tx)
R0 01:22:11,7935: 0x19c  1-Data: ff                    |        |PDO1 (tx)
```

Недоліки використання Event Timer:

- Дані можуть бути незмінними, але полосу пропускання – займають
- Таймери від різних вузлів несинхронізовані

Сервіси CANopen: PDO – Process Data Object

- **Event Driven trigger – встановлюється значенням параметру**
  Index 1800h…19FFh, Subindex 2 (Transmission type)
  наприклад 0xFE (Manufacturer Specific), 0xFF (Device Profile specific)

```
R0 01:41:39,4804: 0x19c  1-Data: ff                    |        |PDO1 (tx)
R0 01:41:40,0634: 0x19c  1-Data: f7                    |        |PDO1 (tx)
R0 01:41:40,2214: 0x19c  1-Data: ef                    |        |PDO1 (tx)
R0 01:41:40,6044: 0x19c  1-Data: fe                    |        |PDO1 (tx)
R0 01:41:40,7254: 0x19c  1-Data: ed                    |        |PDO1 (tx)
```

Перевага Event Driven Trigger:

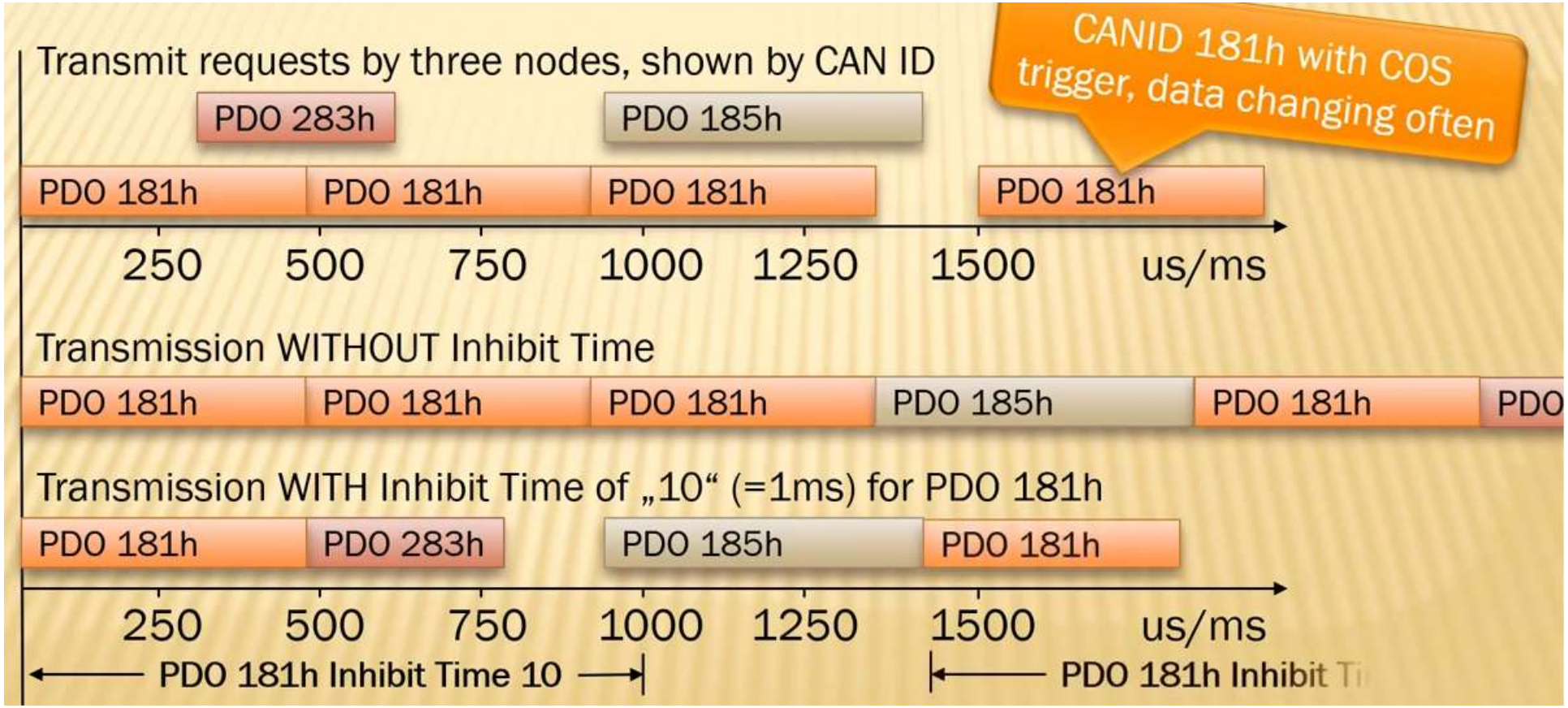
- Дані передаються тільки коли змінюються (зберігається полоса пропускання)

Недоліки Event Driven Trigger:

- Якщо приймальний вузол загрузиться пізніше, він не буде знати які дані актуальні на зараз
- Треба враховувати потреби інших вузлів (use Inhibit Time Parameter – next slide)

Сервіси CANopen:
PDO – Process Data Object

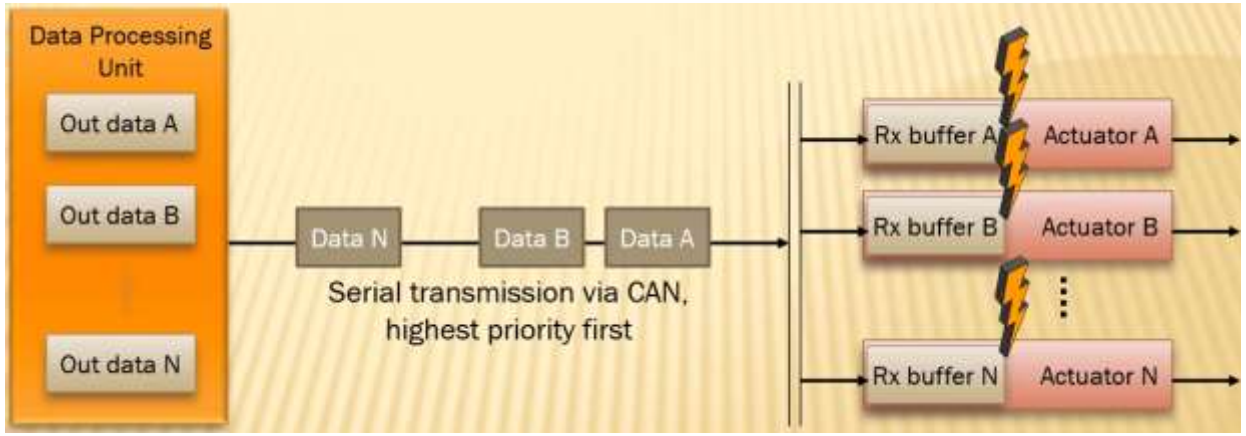
- **Inhibit Time – встановлюється значенням параметру**
  Index 1800h…19FFh, Subindex 3 (Inhibit Time)



Сервіси CANopen: PDO – Process Data Object

- **SYNC- Related PDO – встановлюється значенням параметру**
  Index 1800h…19FFh, Subindex 2 (Transmission type = 1~254)
  Transmission type = number of SYNC messages

  SYNC-Related PDO передається тільки після появи визначеної кількості SYNC на шині.

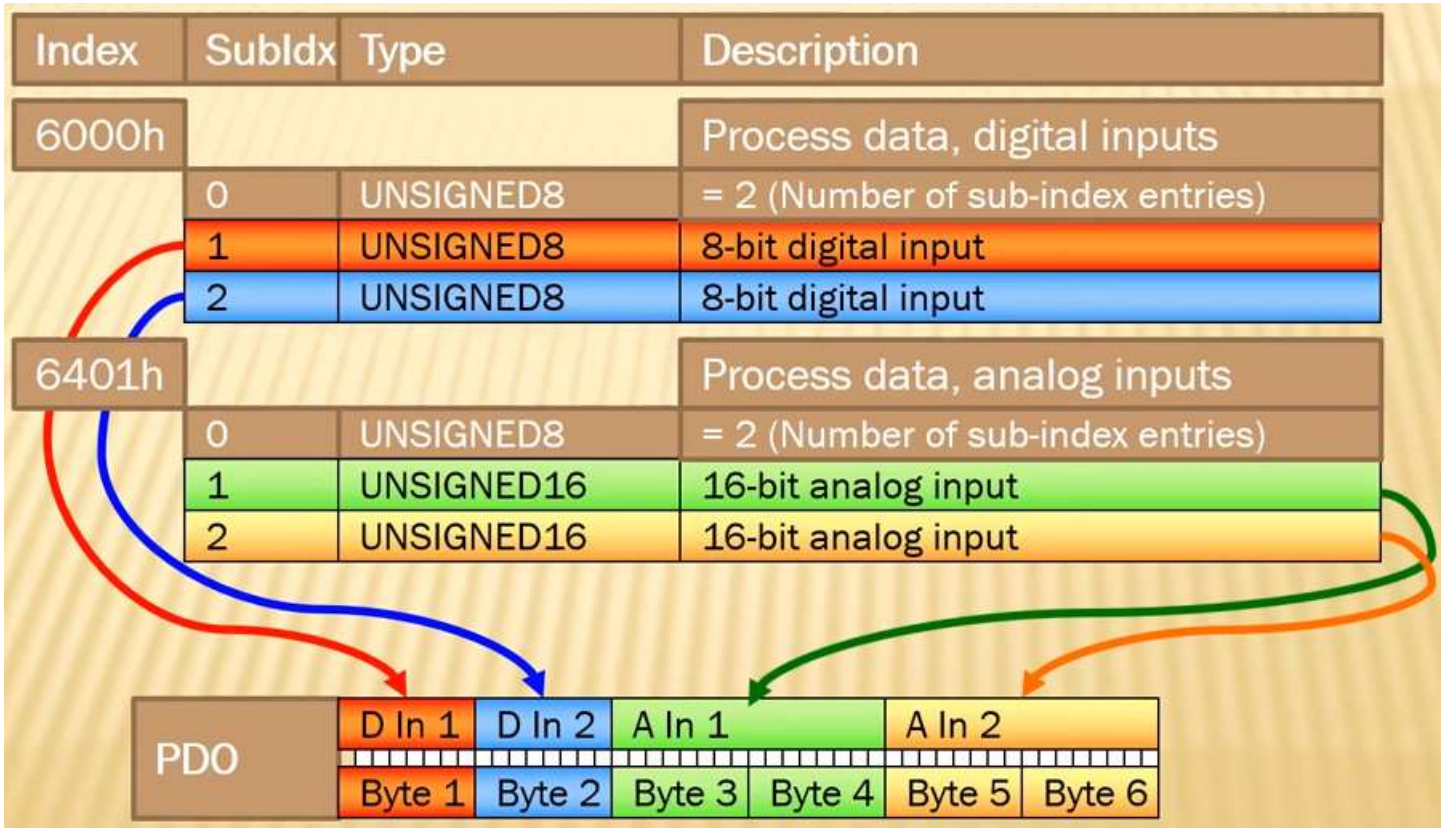



```
T0 02:12:09,1280: 0x080   0-RemoteFrame                          |SYNC
T0 02:12:09,5580: 0x080   0-RemoteFrame                          |SYNC
T0 02:12:09,9187: 0x080   0-RemoteFrame                          |SYNC
R0 02:12:09,9192: 0x29c   8-Data: fe 0f fe 0f 3b 0a fe 0f   |    ;    |PDO2 (tx)
```

NodeID=0x1C, TPDO2 COB_ID = 0x280. 0x29C=COB_ID + NodeID

Сервіси CANopen: Mapping objects in PDO

Декілька об’єктів зі словника можуть передаватися в одному PDO.
Маппінг – це процес визначення які саме об’єкти передаються в PDO.

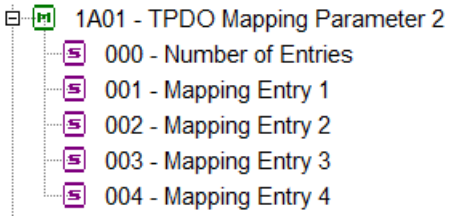


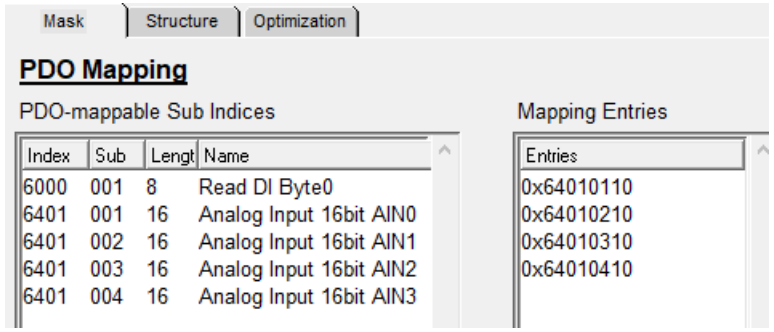
Сервіси CANopen:
Mapping objects in PDO

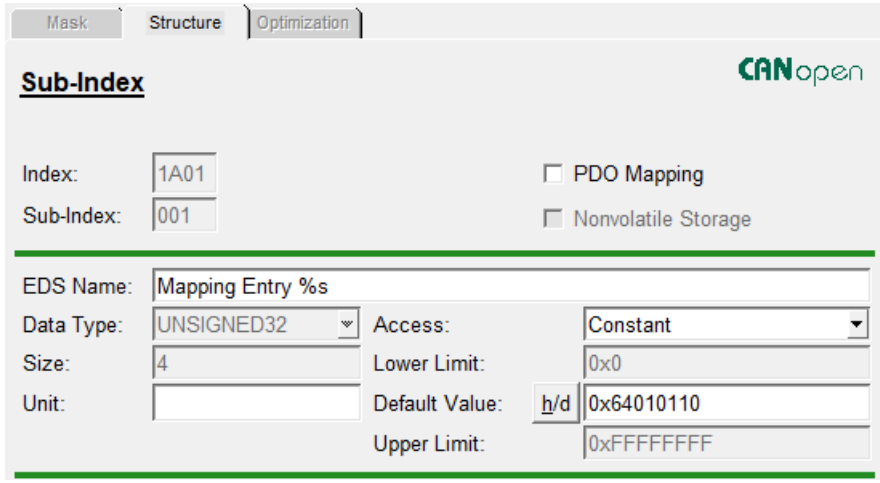
PDO Mapping Parameters:

Transmit PDO Mapping Parameters:
TPDO1…TPDO512
Index 1A00h…1BFFh

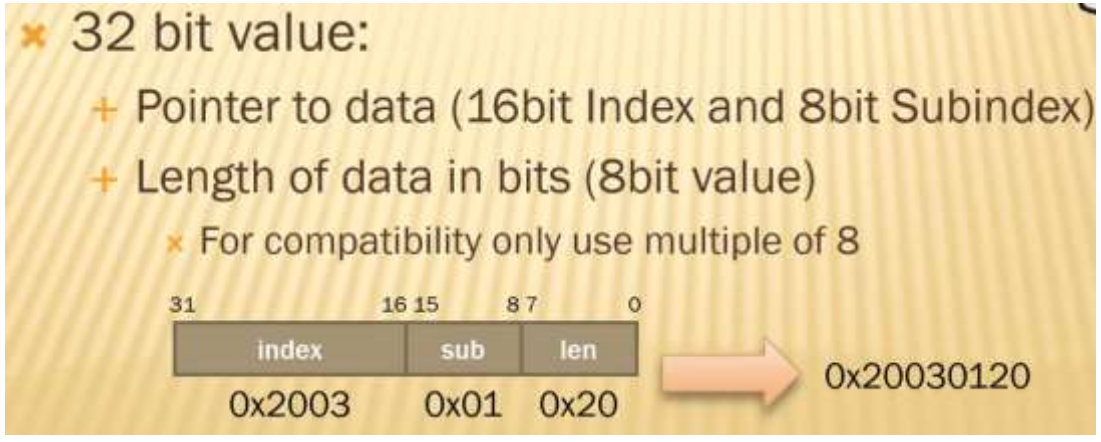
Receive PDO Mapping Parameters
RPDO1…RPDO512
Index 1600h…17FFh



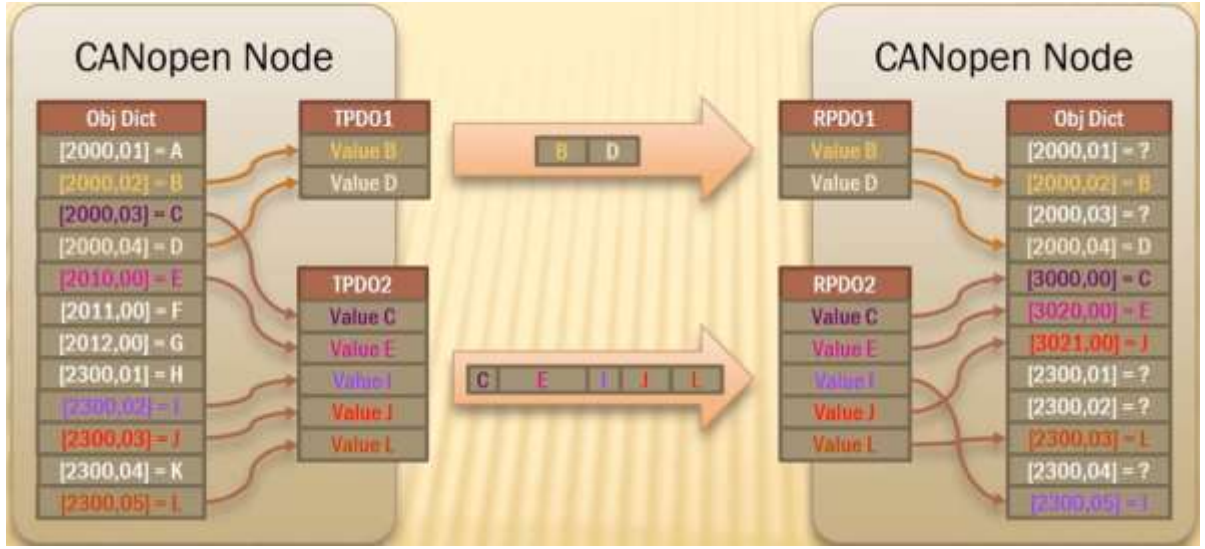


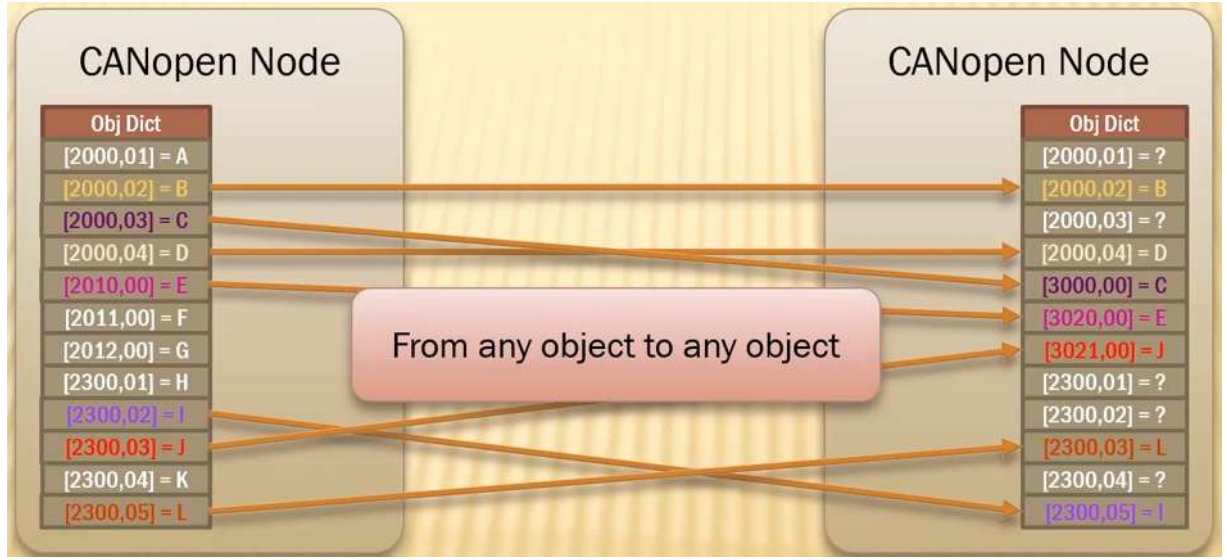


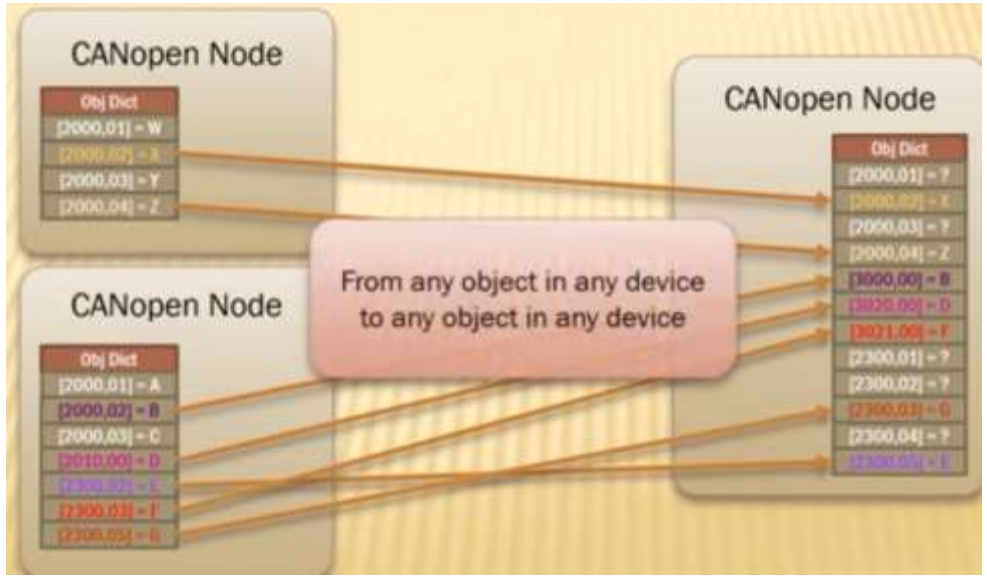
Mapping Entries:



Сервіси CANopen:
PDO Transmit to Receive Mapping







Сервіси CANopen: NMT – Network Management

В процесі роботи вузла – можливі декілька станів (картинка справа), в кожному зі станів можливо користуватися тим чи іншим сервісом CANopen (табличка знизу).

Перехід між станами контролюється NMT Master, через об’єкт з COB-ID = 000

	INITIALISING	PRE-OPERATIONAL	OPERATIONAL	STOPPED
PDO			X	
SDO		X	X	
Synchronisation Object		X	X	
Time Stamp Object		X	X	
Emergency Object		X	X	
Boot-Up Object	X			
Network Management Objects		X	X	X

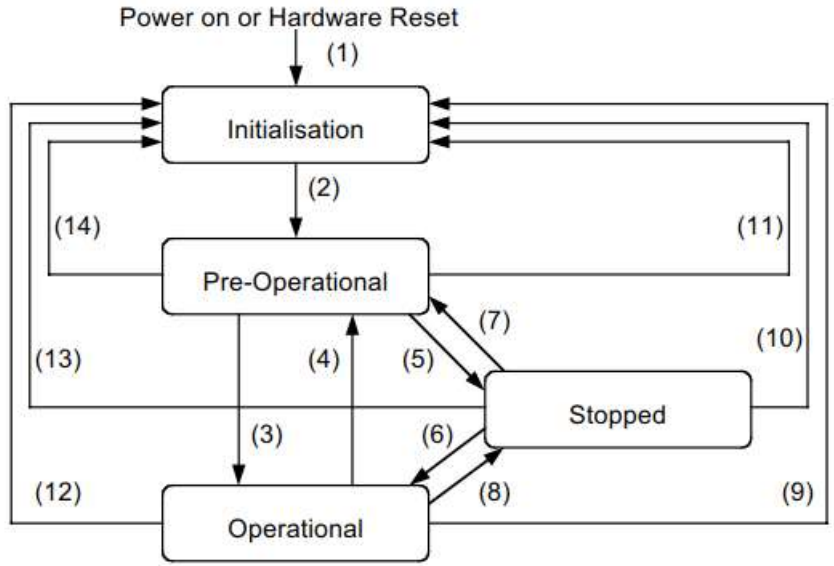
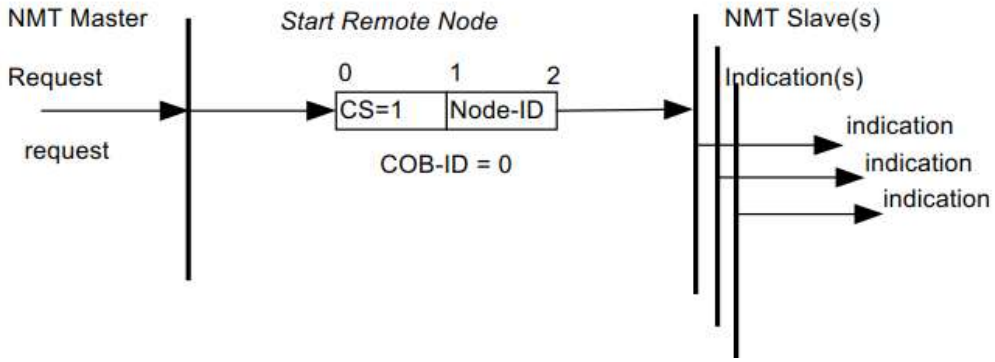


Table 31: Trigger for State Transition

(1)	At Power on the initialisation state is entered autonomously
(2)	Initialisation finished - enter PRE-OPERATIONAL automatically
(3),(6)	Start_Remote_Node indication
(4),(7)	Enter_PRE-OPERATIONAL_State indication
(5),(8)	Stop_Remote_Node indication
(9),(10),(11)	Reset_Node indication
(12),(13),(14)	Reset_Communication indication

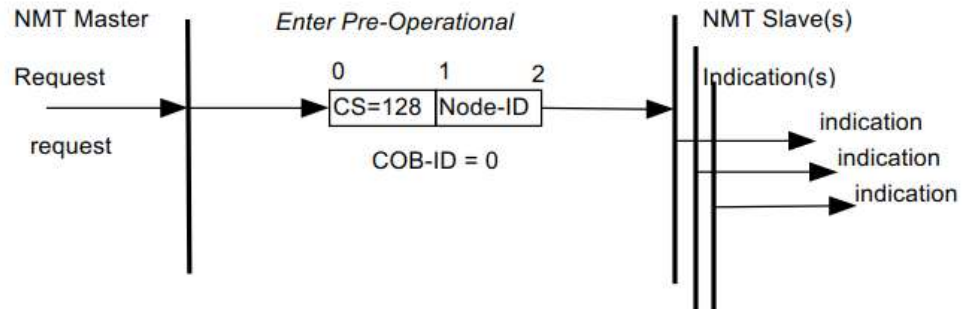
Сервіси CANopen: NMT – Network Management

Start Remote Node Protocol



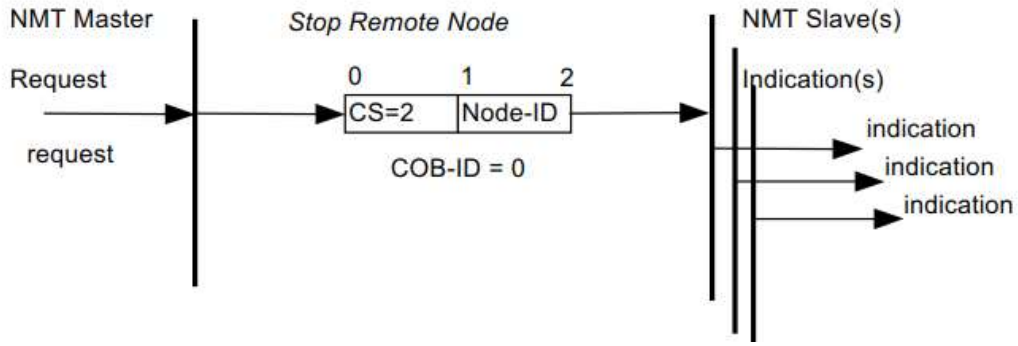
Enter Pre-Operational Protocol

The protocol is used to implement the 'Enter_Pre-Operational' service.

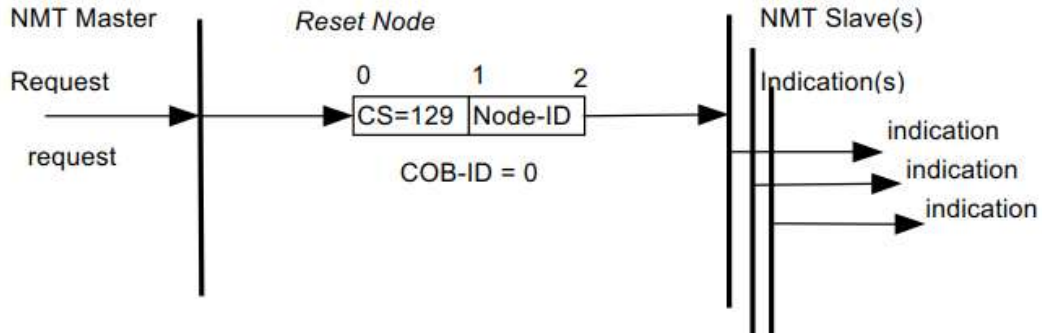


Stop Remote Node Protocol

This protocol is used to implement the 'Stop Remote Node' service.



Reset Node Protocol



Сервіси CANopen: Heartbeat, Bootup

Heartbeat – циклічне повідомлення що несе в собі дані про поточний стан (NMT state) вузла.

Налаштовується об’єктом 1017hex, в мілісекундах.

COB-ID = 1792 (0x700) + NodeID

```
R0 00:12:48,0947: 0x71c  1-Data: 05      | NMT Nodeguarding/Heartbeat 28 Operational
R0 00:12:49,0947: 0x71c  1-Data: 05      | NMT Nodeguarding/Heartbeat 28 Operational
:send:nmt 2 nodeid
T0 00:12:49,3497: 0x000  2-Data: 02 1c   | NMT Stop Remote Node 28
R0 00:12:50,0947: 0x71c  1-Data: 04      | NMT Nodeguarding/Heartbeat 28 Stopped
R0 00:12:51,0947: 0x71c  1-Data: 04      | NMT Nodeguarding/Heartbeat 28 Stopped
.....
:send:nmt 128 nodeid
T0 00:12:57,1171: 0x000  2-Data: 80 1c   | NMT Enter Pre-Operational Node 28
R0 00:12:58,0947: 0x71c  1-Data: 7f      | NMT Nodeguarding/Heartbeat 28 Pre-Operational
R0 00:12:59,0947: 0x71c  1-Data: 7f      | NMT Nodeguarding/Heartbeat 28 Pre-Operational
R0 00:13:00,0947: 0x71c  1-Data: 7f      | NMT Nodeguarding/Heartbeat 28 Pre-Operational
.....
:send:nmt 129 nodeid
T0 00:18:25,3057: 0x000  2-Data: 81 1c   | NMT Reset Node 28
H
R0 00:18:25,3335: 0x71c  1-Data: 00      | NMT Bootup Node 28
```

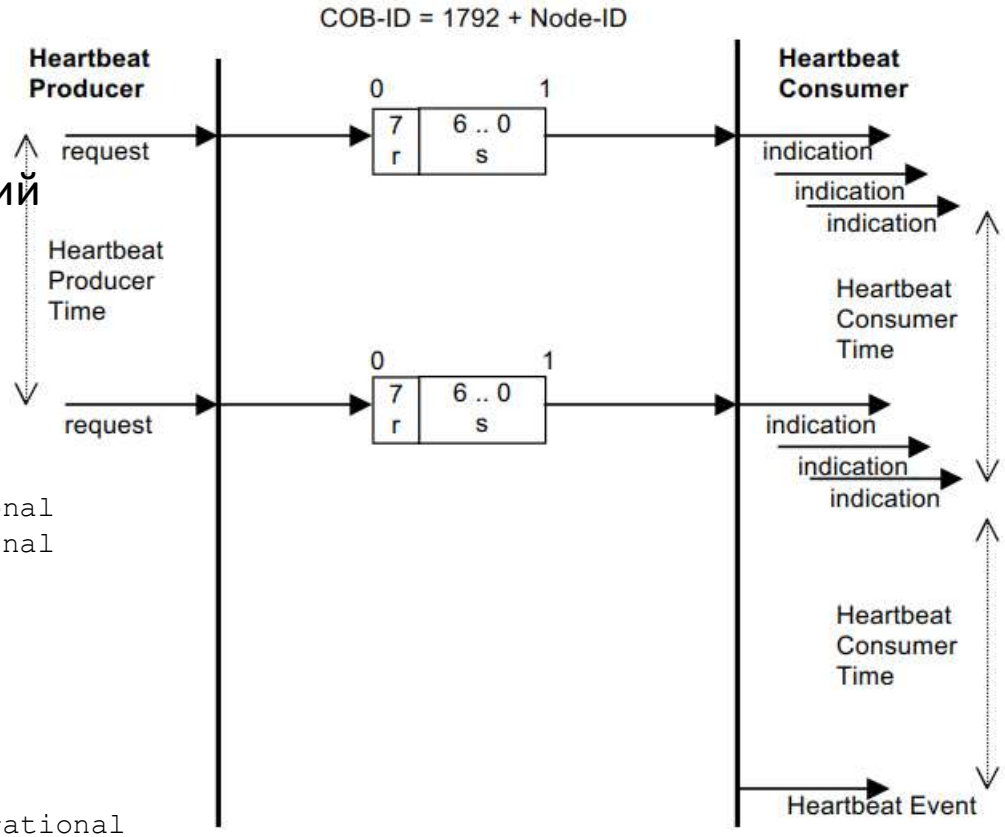


Figure 42: Heartbeat Protocol

r: reserved (always 0)
s: the state of the Heartbeat producer
- 0: BOOTUP
- 4: STOPPED
- 5: OPERATIONAL
- 127: PRE-OPERATIONAL

Q&A

- **Questions so far?**

Thank you
for your enthusiasm!

.....

Roman Fedoryak
SIEMENS HEALTHCARE LIMITED LIABILITY COMPANY
SHS TE ME PLM SD ESW UKR1
14 Uhorska Street
79034 Lviv, Ukraine

mailto: roman.fedoryak@siemens-healthineers.com

.....