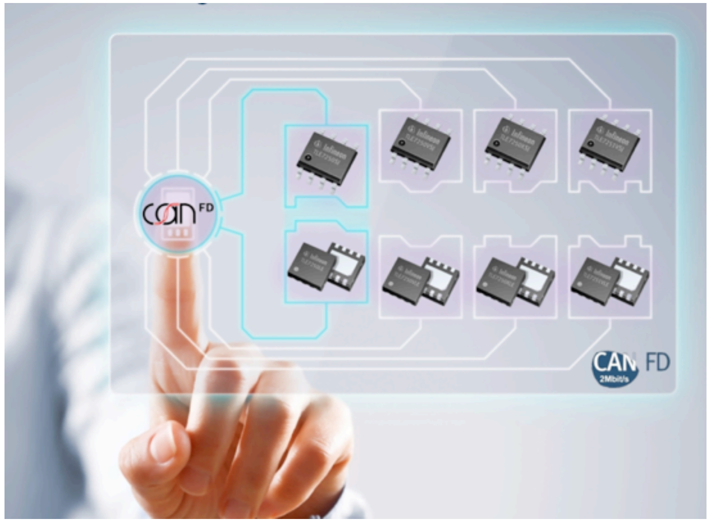
Holger Zeltwanger

The future of Classical CAN and CAN FD in aerospace applications

European Space Agency

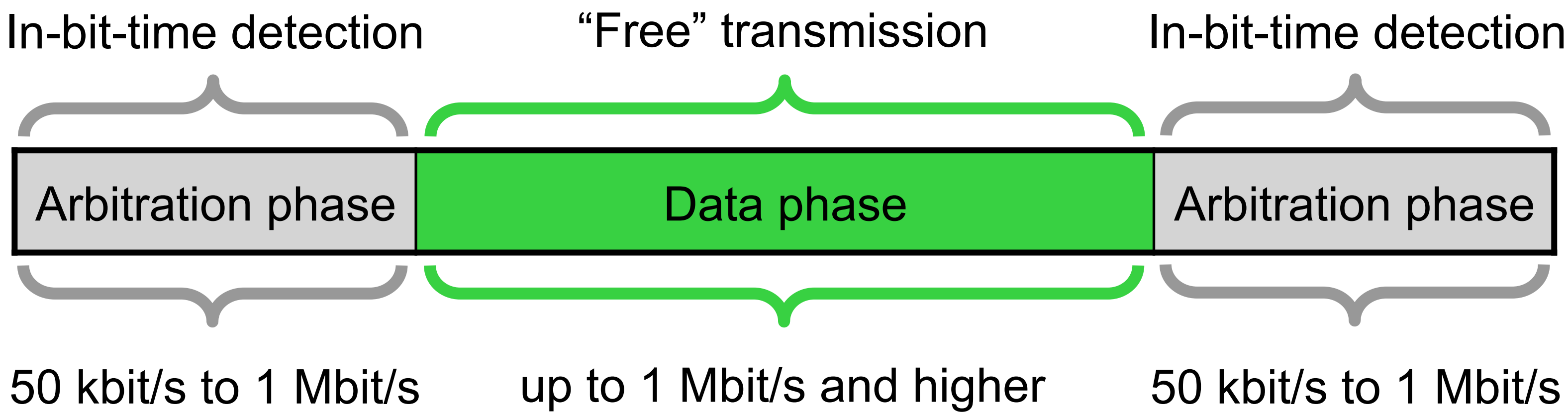
CAN in Space Workshop

Contents of presentation



- Classical CAN and CAN FD – The basics
- CAN FD data link layer protocol – Some details
- CAN FD physical layer – More details
- CAN FD system design – The challenge
- CANopen FD – The next generation

CAN with flexible data-rate



Arbitration phase

- Nodes are synchronized
- Limits: Internal transceiver delays and bus-line delay

Data phase

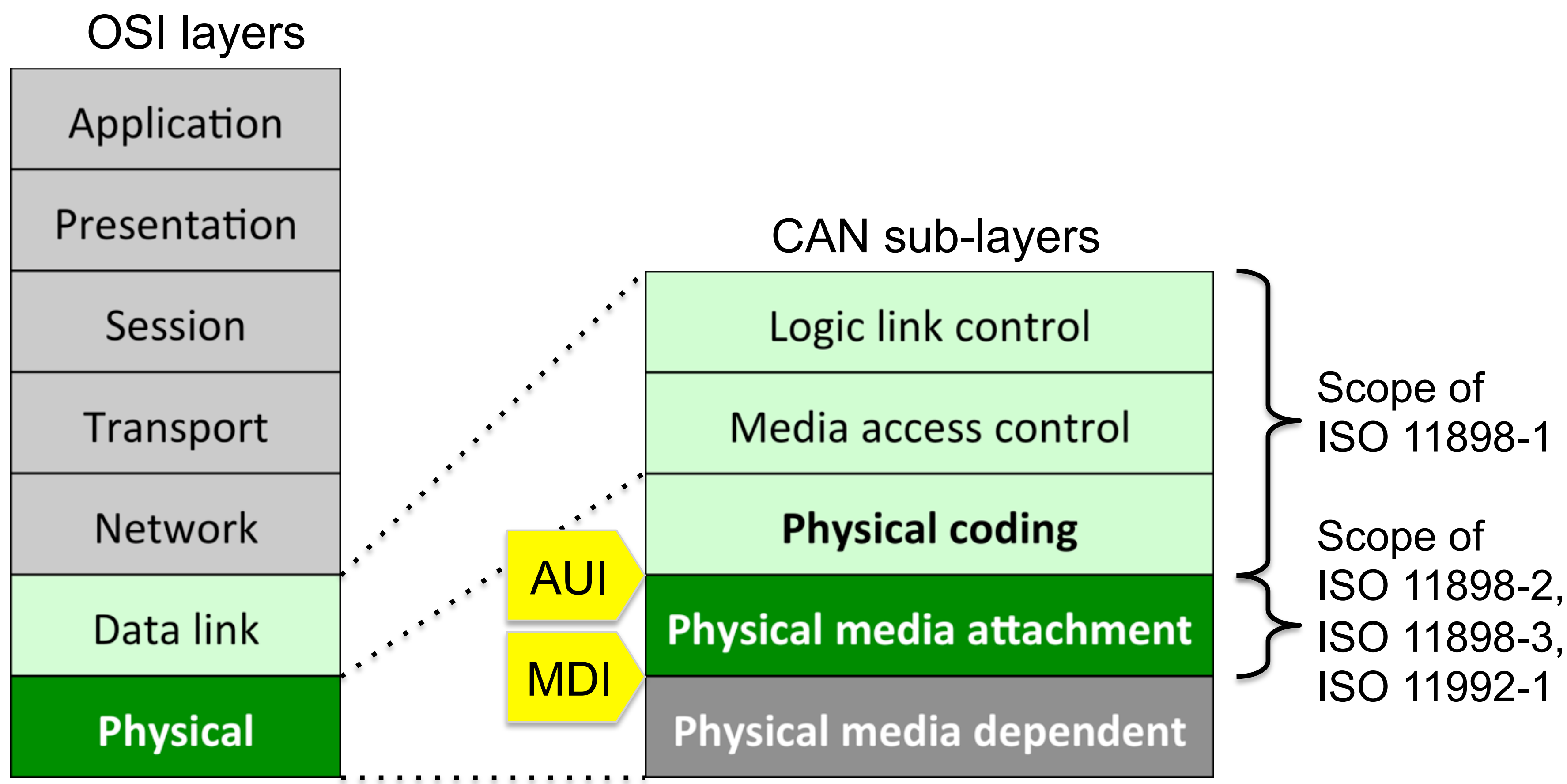
- Nodes are not synchronized
- Limits: Transceiver asymmetry and transmitter delay compensation capability (2 nominal arbitration-phase bit-times)

Faster and longer

	Header	Protected payload	Trailer
Classical CAN	SOF Arbitration field Control field	Data field (up to 8 byte) CRC field	ACK field EOF (IMF)
CAN FD8	SOF Arbitration field Control field	Control field 8-byte data field CRC field	ACK field EOF (IMF)
CAN FD64	SOF Arbitration field Control field	Control field 64-byte data field CRC field	ACK field EOF (IMF)

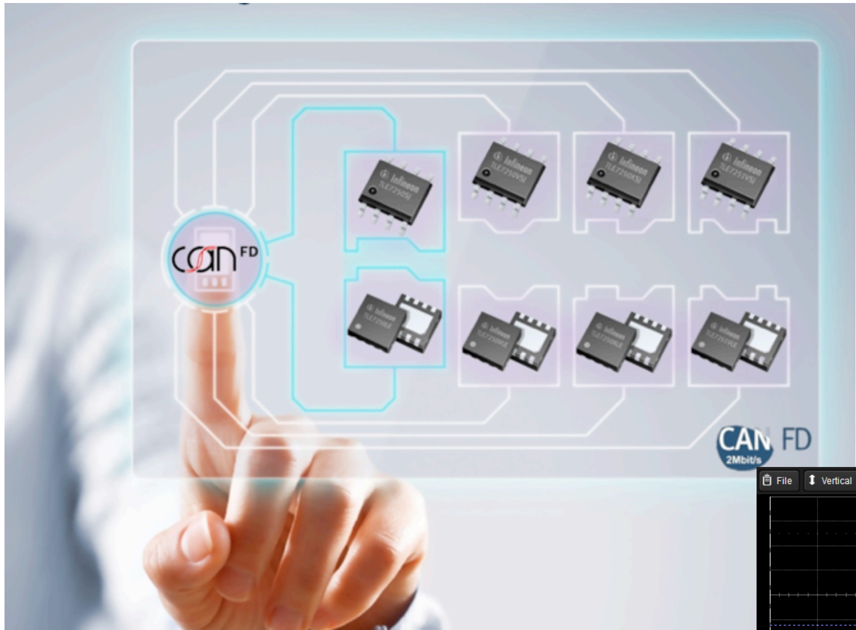
NOTE An arbitration/dataphase ratio of 1:8 would lead to approximately six-times data throughput

CAN (FD) and the OSI model



KEY
OSI (open system interconnection), AUI (attachment unit interface), MDI (media-dependent interface)

CAN FD lower layers



ISO 11898-1:2015 specifies the Classical CAN and the CAN FD data link layer plus the physical signaling sub-layer of the OSI physical layer.

ISO 11898-2:2016 specifies the physical media attachment sub-layer for bit-rates up to 5 Mbit/s with optional low-power and selective wake-up features.

Conformance test plans

International standards (in preparation)

- ISO 16845-1: Classical CAN and CAN FD
- ISO 16845-2: High-speed transceiver (with low-power mode and selected wake-up functionality)

NOTE   Conformance testing is like spellchecking in human communication. It increases the probability of interoperability, but doesn't guarantee it! CAN controllers and CAN transceiver tested by different test plan implementations can have different results. Complementary interoperability tests (e.g. plug-fests) are necessary to satisfy system designers.

CAN FD plug-fests

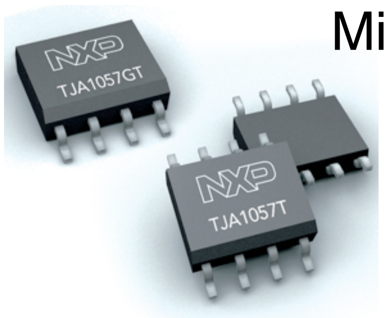
- *March 2015 in Nuremberg:* Testing of ISO CAN FD separately from non-ISO CAN FD controllers in conjunction with Denso's ringing suppression technology. Additionally, long cables (up to 250 m) were tested with a 250-kbit/s arbitration phase bit-rate and a 2-Mbit/s data-phase bit-rate.

- *March 2015 in Detroit:* Testing of ISO CAN FD separately from non-ISO CAN FD controllers using GM wiring harness.

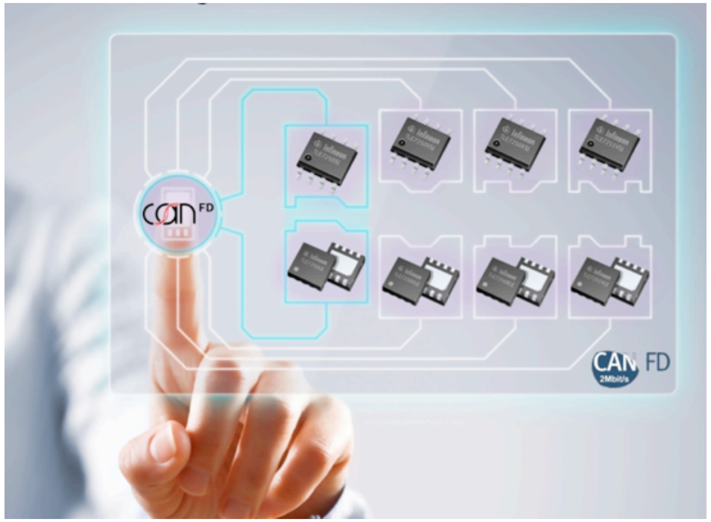
Next plug-fests in April/June 2016 in USA and Germany.

Availability of CAN FD

- ISO CAN FD cores: Bosch, Fraunhofer/Cast, IFI, Inicore, Kvaser, etc.
- Stand-alone ISO CAN FD controller: Microchip, etc.
- On-MCU ISO CAN FD modules: Cypress (formerly: Spansion/Fujitsu), Infineon, Microchip/ Atmel, NXP/Freescale, Renesas, STM, Texas Instruments, etc.
- 2-Mbit/s qualified transceiver (ISO 11898-2): Infineon, Microchip/Atmel, NXP, TI, etc.
- 5-Mbit/s qualified transceiver (ISO 11898-2): Microchip/Atmel, etc.



Contents of presentation



- Classical CAN and CAN FD – The basics
- **CAN FD data link layer protocol – Some details**
- CAN FD physical layer – More details
- CAN FD system design – The challenge
- CANopen FD – The next generation

CAN FD data frame formats

SOF	Arbitration field	Control field	Data field (payload)	CRC field	ACK field	EOF	IMF
1 bit	12 *or* 32* bit	8 *or* 9* bit	0 *to* 64* byte	28 *or* 33 bit**	2 bit	7 bit	3 bit

MSB LSB

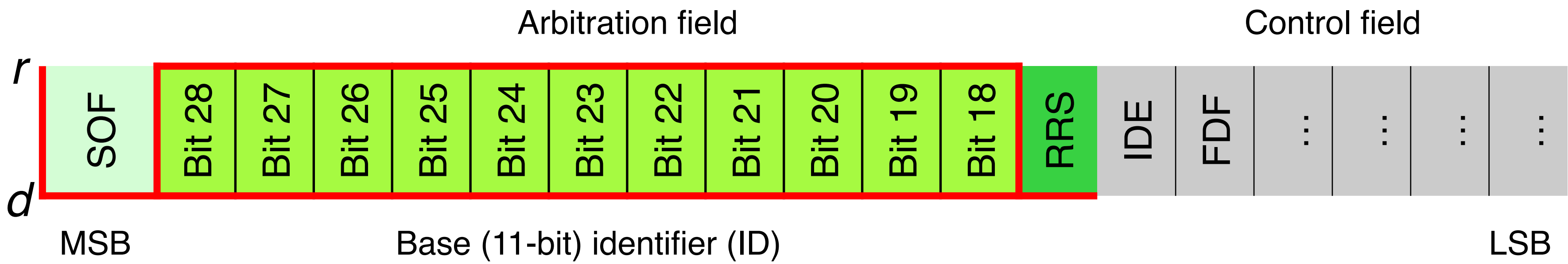
* Stuff-bits are not considered.
** With fixed stuff-bits.

LEGEND
- SOF = start-of-frame
- CRC = cyclic redundancy check
- ACK = acknowledgement
- EOF = end-of-frame
- IMF = intermission field

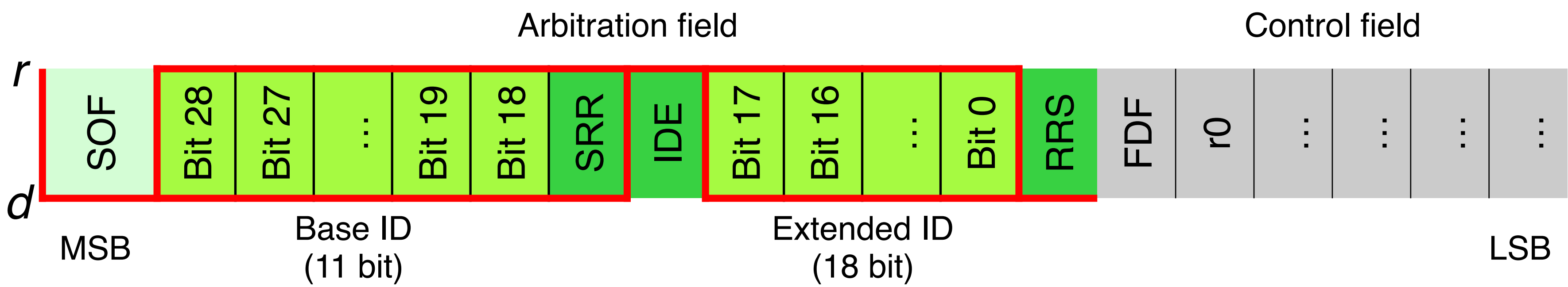
NOTE *The CAN FD protocol controller shall support also the Classical CAN protocol. Both protocols are internationally standardized in ISO 11898-1:2015. There are also non-ISO CAN FD controllers on the market, they are not compliant to the mentioned ISO standard, they don't implement the additional safe-guard features.*

SOF and arbitration field

FD base frame format (FBFF)



FD extended frame format (FEFF)

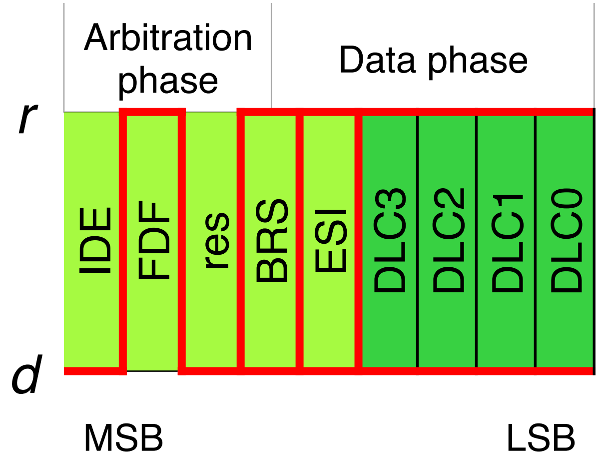


SRR doesn’t care; IDE is recessive in FEFF

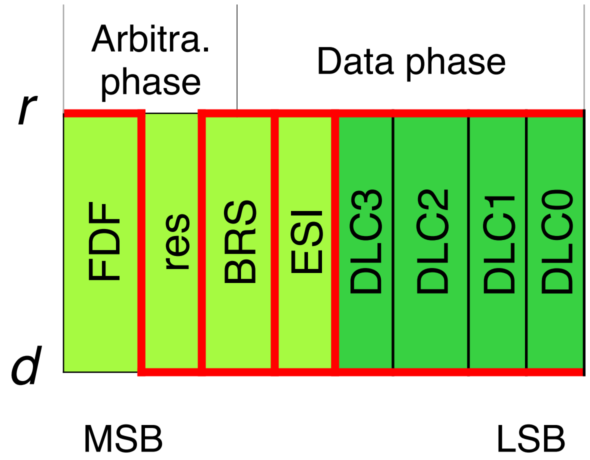
KEY: IDE (identifier extension), FDF (flexible data rate format), RRS (remote request substitution) SRR (substitute remote request)

CAN FD control field

FBFF



FEFF



No. of data bytes	Data length code (DLC)			
	DLC3	DLC2	DLC1	DLC0
0 *to* 8	As in Classical CAN			
12	r	d	d	r
16	r	d	r	d
20	r	d	r	r
24	r	r	d	d
32	r	r	d	r
48	r	r	r	d
64	r	r	r	r

KEY: IDE (identifier extension), FDF (flexible data rate format), BRS (bit rate switch; recessive, if alternate bit-rate), ESI (error state indicator; recessive, if error passive)

CAN FD data field

Byte 1	Byte2	Byte 3	...	...	Byte 62	Byte 63	Byte 64

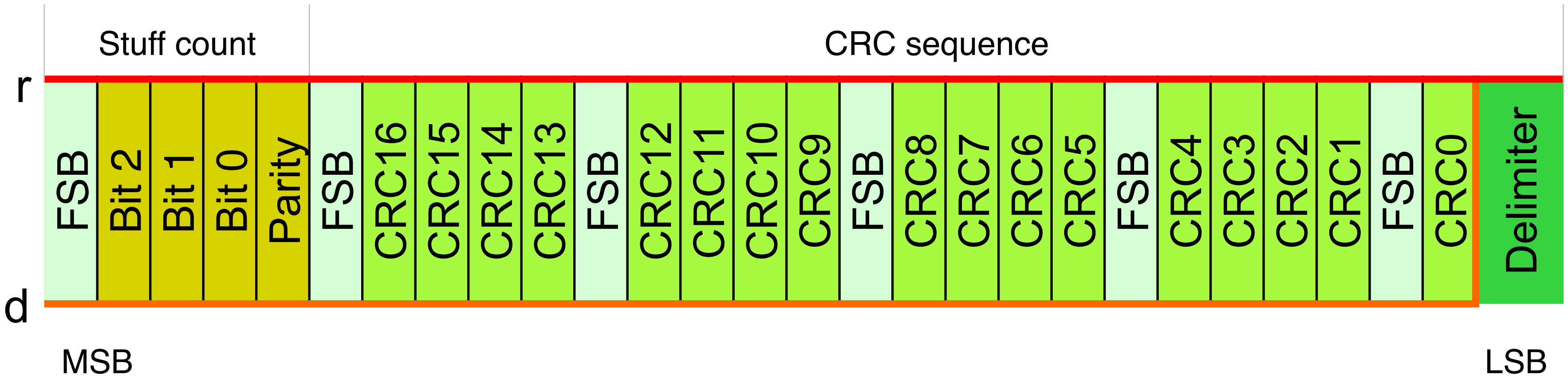
min. length of data field = 0 byte

Byte 1	Byte2	Byte 3	...	...	Byte 62	Byte 63	Byte 64
MSB							LSB

max. length of data field = 64 byte

CAN FD CRC field (1)

CRC field for DLCs ≤ 10 (≤ 16 data bytes)

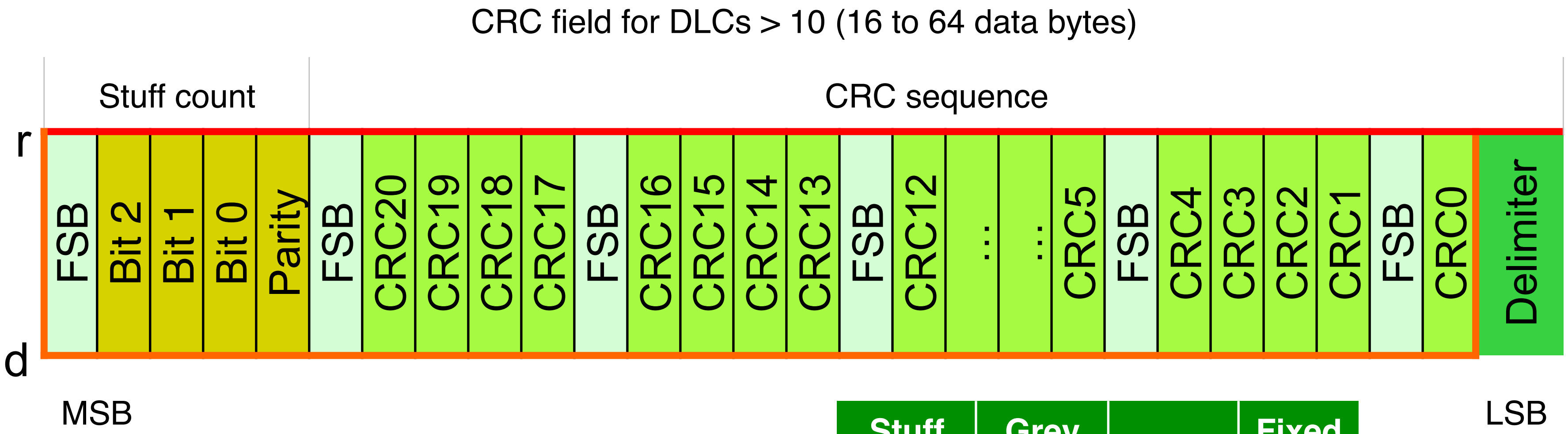


17-bit polynomial: $x^{17} + x^{16} + x^{14} + x^{13} + x^{11} + x^{6} + x^{4} + x^{3} + x^{1} + 1$
The CRC init vector is “10…0”.

The fixed stuff bits (FSB) have the opposite value of the previous bit.

NOTE During the transmission of the CRC delimiter bit, the nodes switch back to the arbitration bit-rate, therefore the CRC delimiter bit vary in length (up to arbitration bit-times). All nodes are synchronized after the next recessive-to-dominant edge.

CAN FD CRC field (2)



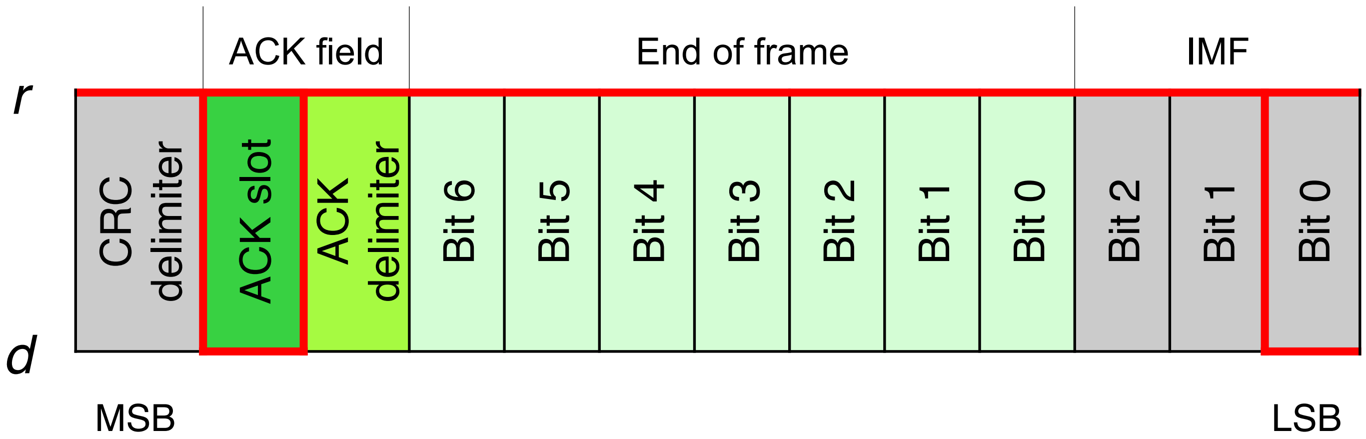
Stuff count	Grey code	Parity	Fixed SB
0	0 0 0	0	1
1	0 0 1	1	0
2	0 1 1	0	1
3	0 1 0	1	0
4	1 1 0	0	1
5	1 1 1	1	0
6	1 0 1	0	1
7	1 0 0	1	0

21-bit polynomial:

$x^{21} + x^{20} + x^{13} + x^{11} + x^{7} + x^{4} + x^{3} + 1$

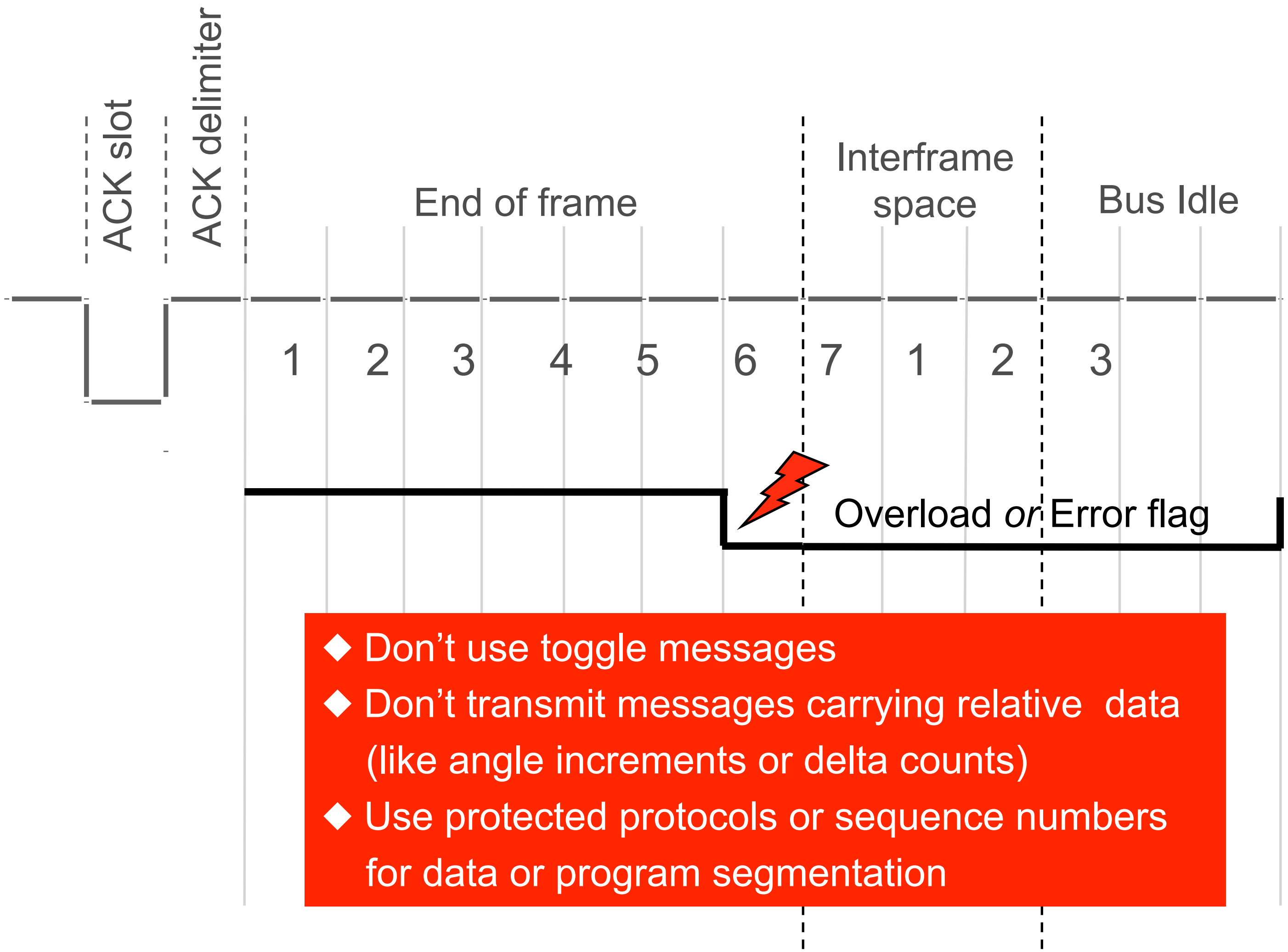
The 3-bit stuff bit counter contains the number of stuff bits in MOD8 followed by a parity bit.

ACK field, EOF, and IMF



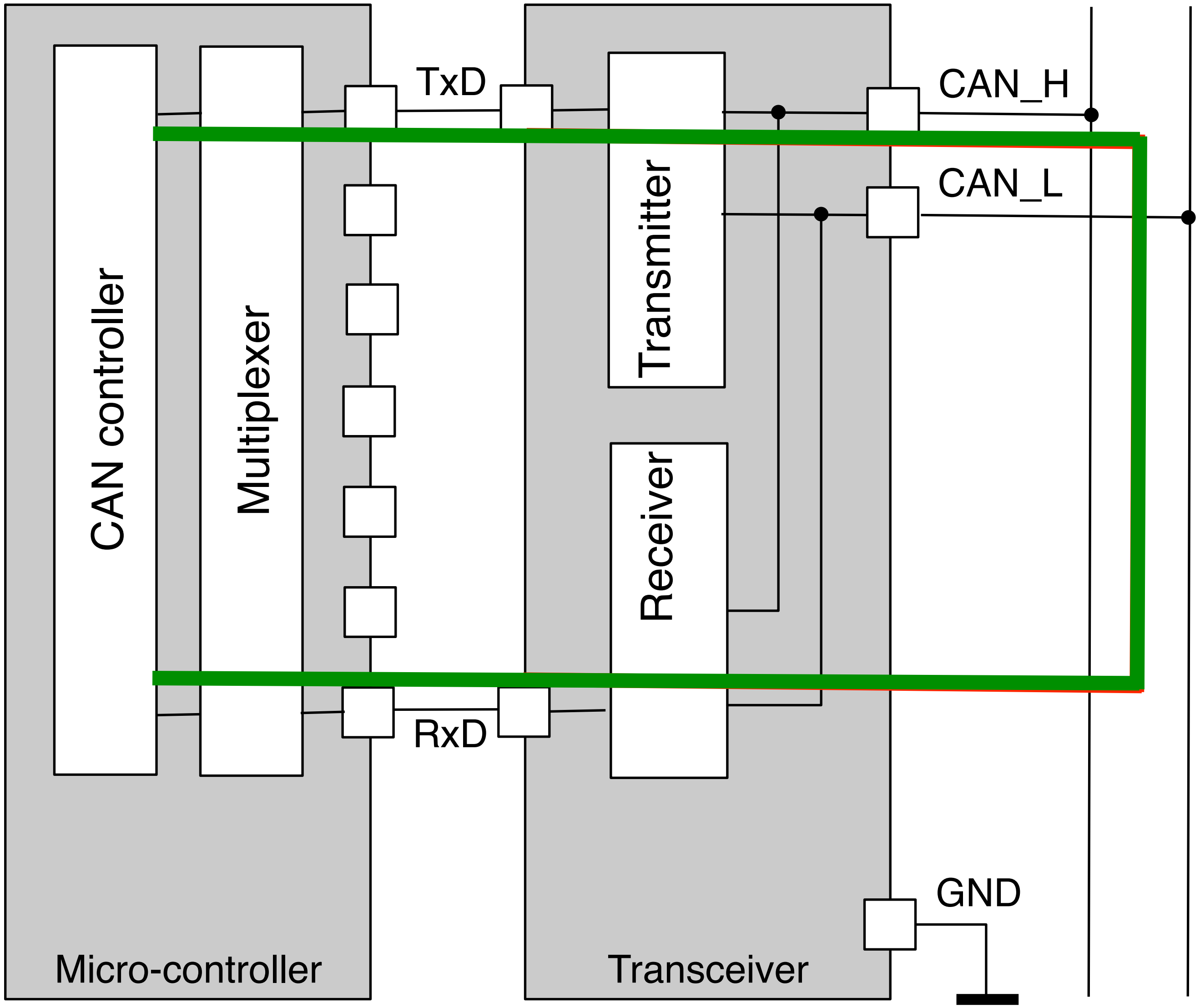
NOTE The acknowledge (ACK) field, the end of frame, and the intermission field are the same as in Classical CAN.

Message doubling

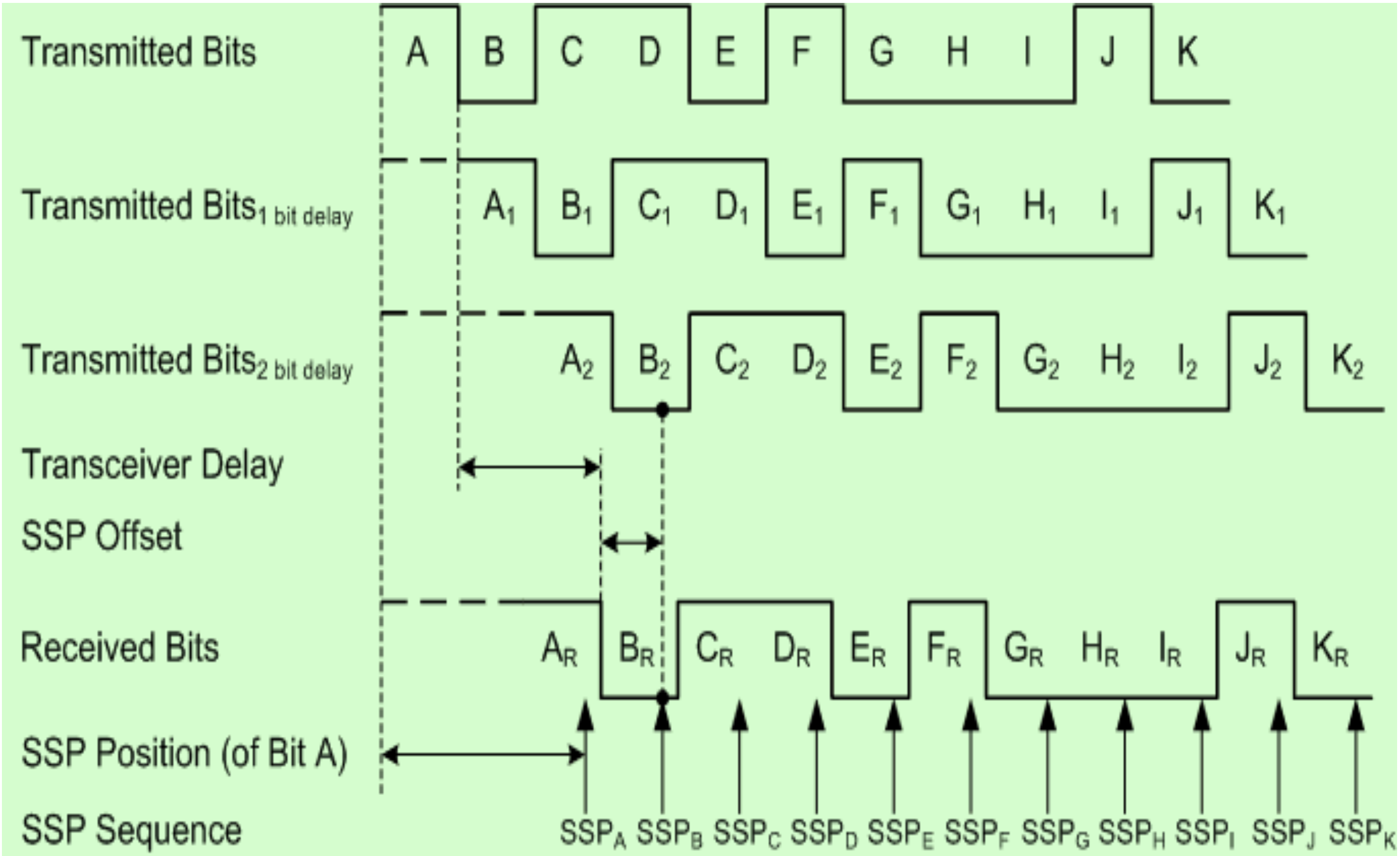


- Don’t use toggle messages
- Don’t transmit messages carrying relative data (like angle increments or delta counts)
- Use protected protocols or sequence numbers for data or program segmentation

Transmitter loop-delay

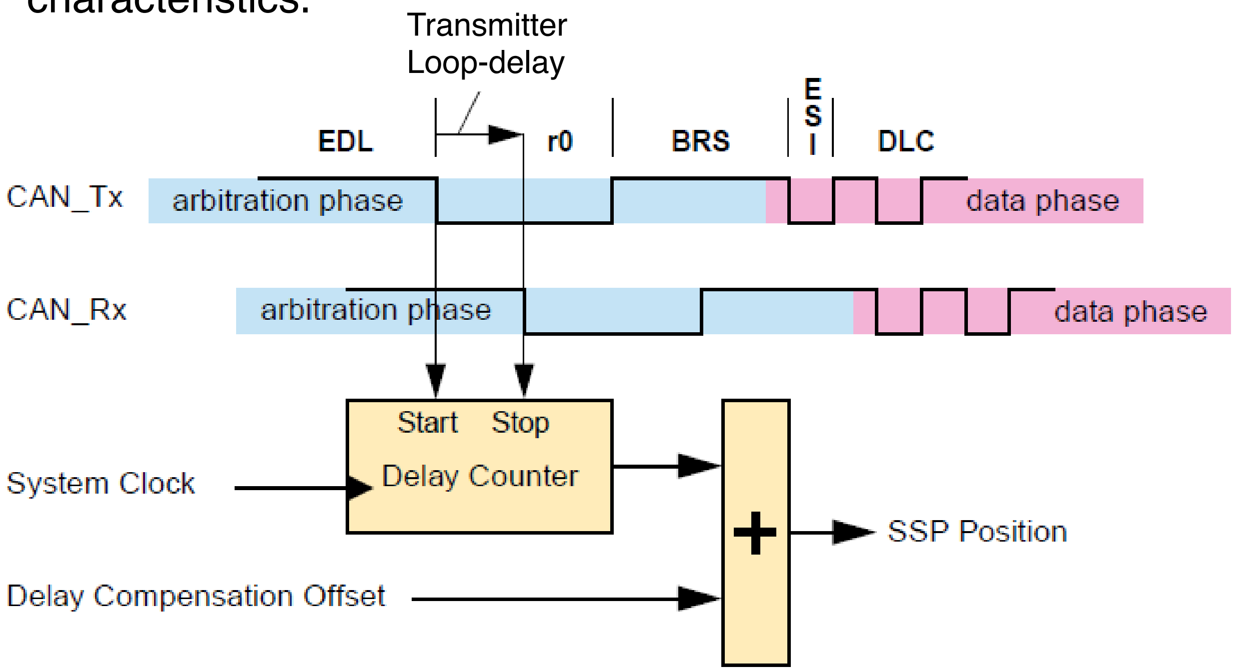


Compensation (example)



Loop-delay compensation

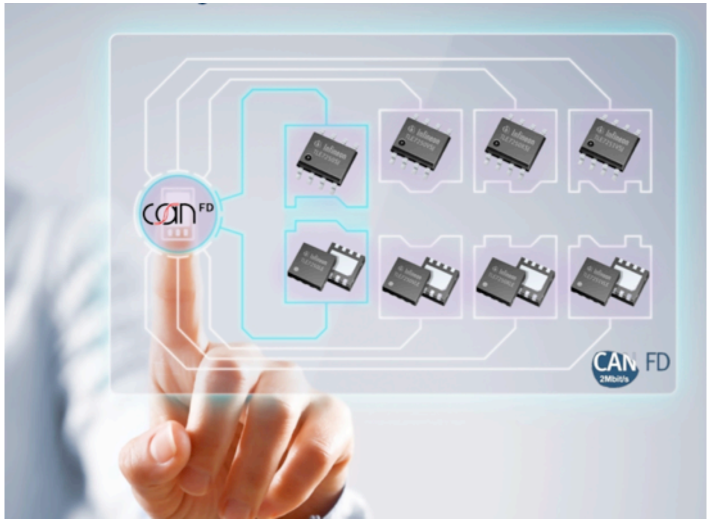
The transmitter loop-delay is measured for each CAN FD frame at the falling edge of the EDL bit. The delay compensation is independent of transceiver characteristics.



- Transmitter delay is measured in system clock periods.
- Configurable offset added to adjust *secondary sample point* (SSP) inside bit time (SSP position rounded down to next integer number of time quanta).
- Delayed transmit data compared against received data at SSP (check for bit errors).

CAN

Contents of presentation



- Classical CAN and CAN FD – The basics
- CAN FD data link layer protocol – Some details
- **CAN FD physical layer – More details**
- CAN FD system design – The challenge
- CANopen FD – The next generation

Additional parameters (1)

Requirements for a 2-Mbit/s bit-rate

- Supply voltage range: $4{,}75\ V \leq V_{CC} \leq 5{,}25\ V$
- Temperature range: $-40\ °C \leq T_j \leq 150\ °C$
- Bus-load resistor: $R_L = 60\ \Omega$
- Bus-load capacitor: $C_L = 100\ pF$
- I/O supply voltage: Full specified range
- Load on RxD: $C_L = 15\ pF$

Receive bit-time after five dominant bits:

- Bit-time: $t_{bit} = 500\ ns$
- Recessive bit-time: $400\ ns \leq t_{REC} \leq 550\ ns$

Additional parameter (2)

Requirements for a 5-Mbit/s bit-rate

- Supply voltage range: $4{,}85\ V \leq V_{CC} \leq 5{,}15\ V$
- Temperature range: $-40\ °C \leq T_j \leq 85\ °C$
- Bus-load resistor: $R_L = 60\ \Omega$
- Bus-load capacitor: $C_L = 100\ pF$
- I/O supply voltage: Full specified range
- Load on RxD: $C_L = 15\ pF$

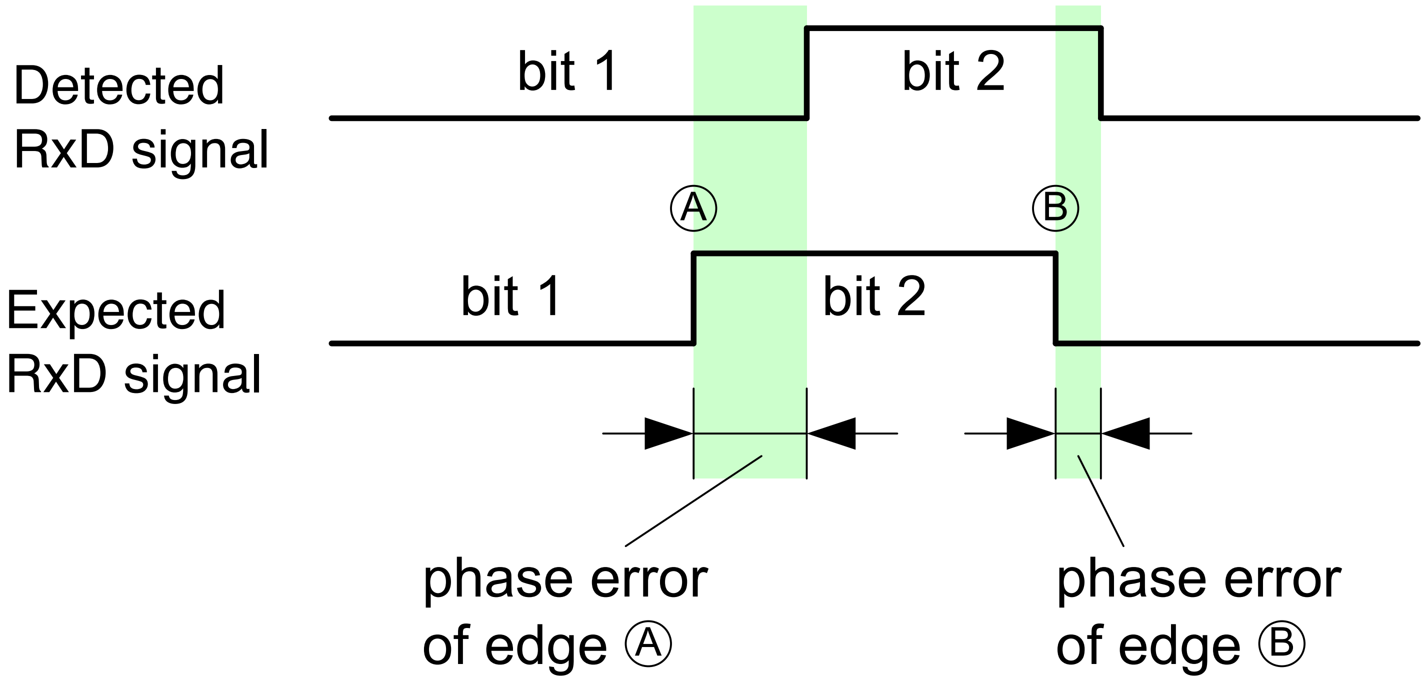
Receive bit-time after five dominant bits:

- Bit-time: $t_{bit} = 200\ ns$
- Recessive bit-time: $120\ ns \leq t_{REC} \leq 220\ ns$

Phase errors

Definition: For each node, the distance (time) between the detected and the expected position of a falling or raising edge is called the phase error of that edge.

Example:



Phase error reasons

Accumulating errors:

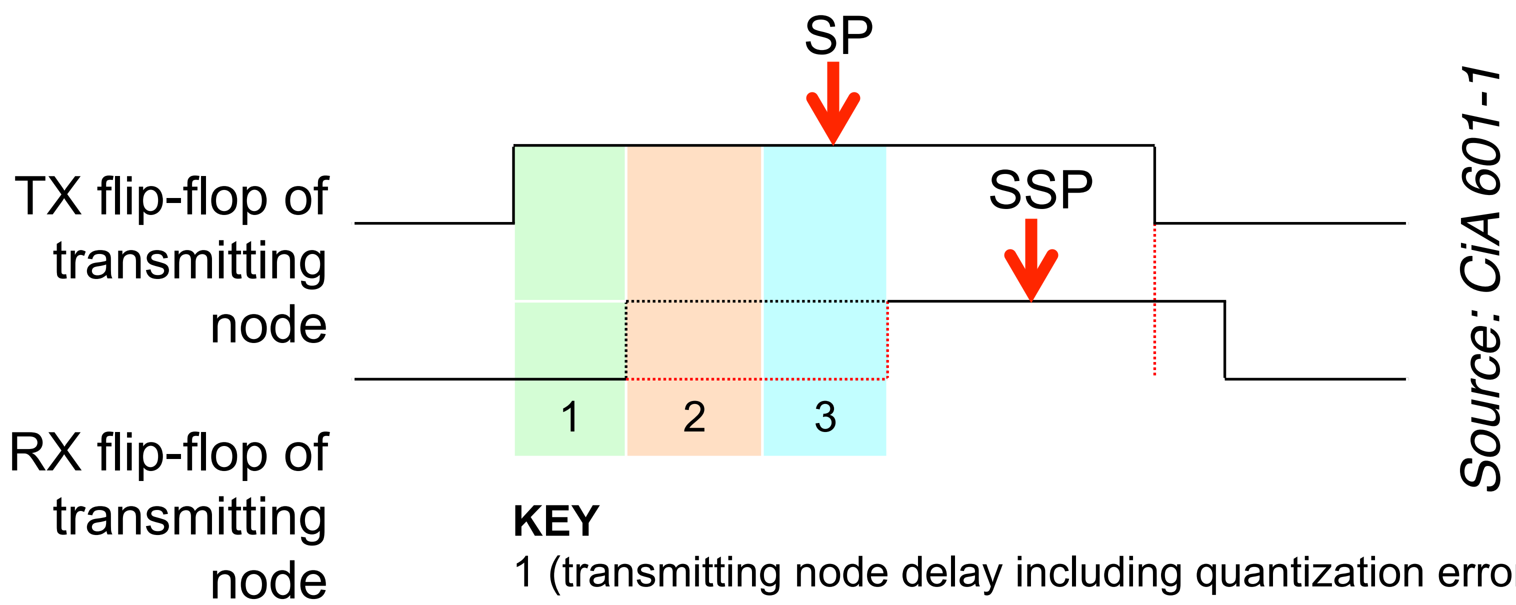
- Different actual bit-rates in sender and receiver (e.g. due to frequency tolerance of oscillators)

Non-accumulating errors:

- Temporary shifts of bit edges or internal shifts in the receiver (e.g. due to bit asymmetries)

This requires to calculate the allowed oscillator frequency and calculate the available phase margin. The oscillator tolerance up to 4 Mbit/s is equal to Classical CAN.

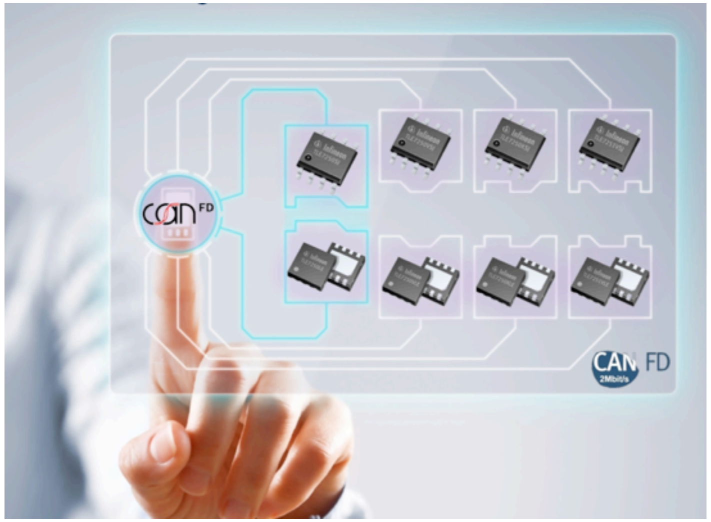
Recessive bit sampling



KEY
1 (transmitting node delay including quantization error)
2 (bit asymmetry)
3 (unstable RX signal due to ringing)
SP (sample point)
SSP (secondary sample point)

NOTE System designer should select physical layer components with minimized asymmetry values and should avoid reflections caused by not matching impedance, for example.

Contents of presentation



- Classical CAN and CAN FD – The basics
- CAN FD data link layer protocol – Some details
- CAN FD physical layer – More details
- **CAN FD system design – The challenge**
- CANopen FD – The next generation

CAN FD network design

- Use line topologies with short stubs
- Use micro-controller CAN ports with symmetric slew rates
- Use micro-controller CAN ports with symmetric CMOS thresholds
- Use short distances between micro-controller and transceiver (in order to avoid parasitic loads and emissions on the board)
- The resistance between micro-controller and transceiver should be in the range of the wire impedance (60 Ω)
- TxD and RxD should be as symmetric as possible
- Avoid additional loop-delays (e.g. optocouplers)
- Use for galvanic isolation DC/DC converter

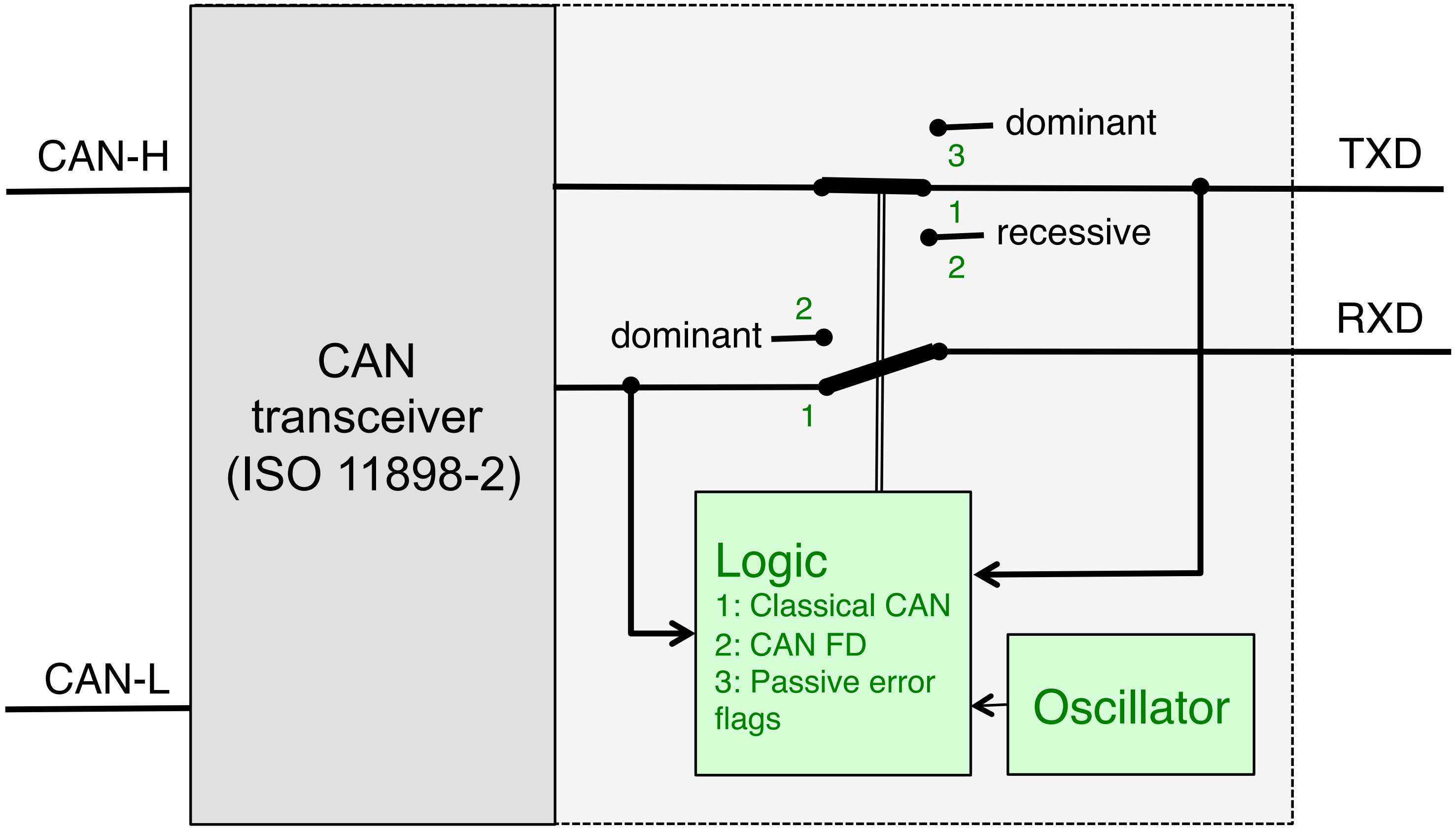
Classical CAN and CAN FD

Theoretically, you can share the network lines with Classical CAN and CAN FD nodes under the following conditions:

- When all nodes support CAN FD and Classical CAN frames;
- When the Legacy CAN nodes use NXP’s FD shield transceivers or Kvaser’s FD filtering transceivers;
- When the Legacy CAN nodes are switched off by means of selective wake-up transceivers (ISO 11898-6) during CAN FD communication.

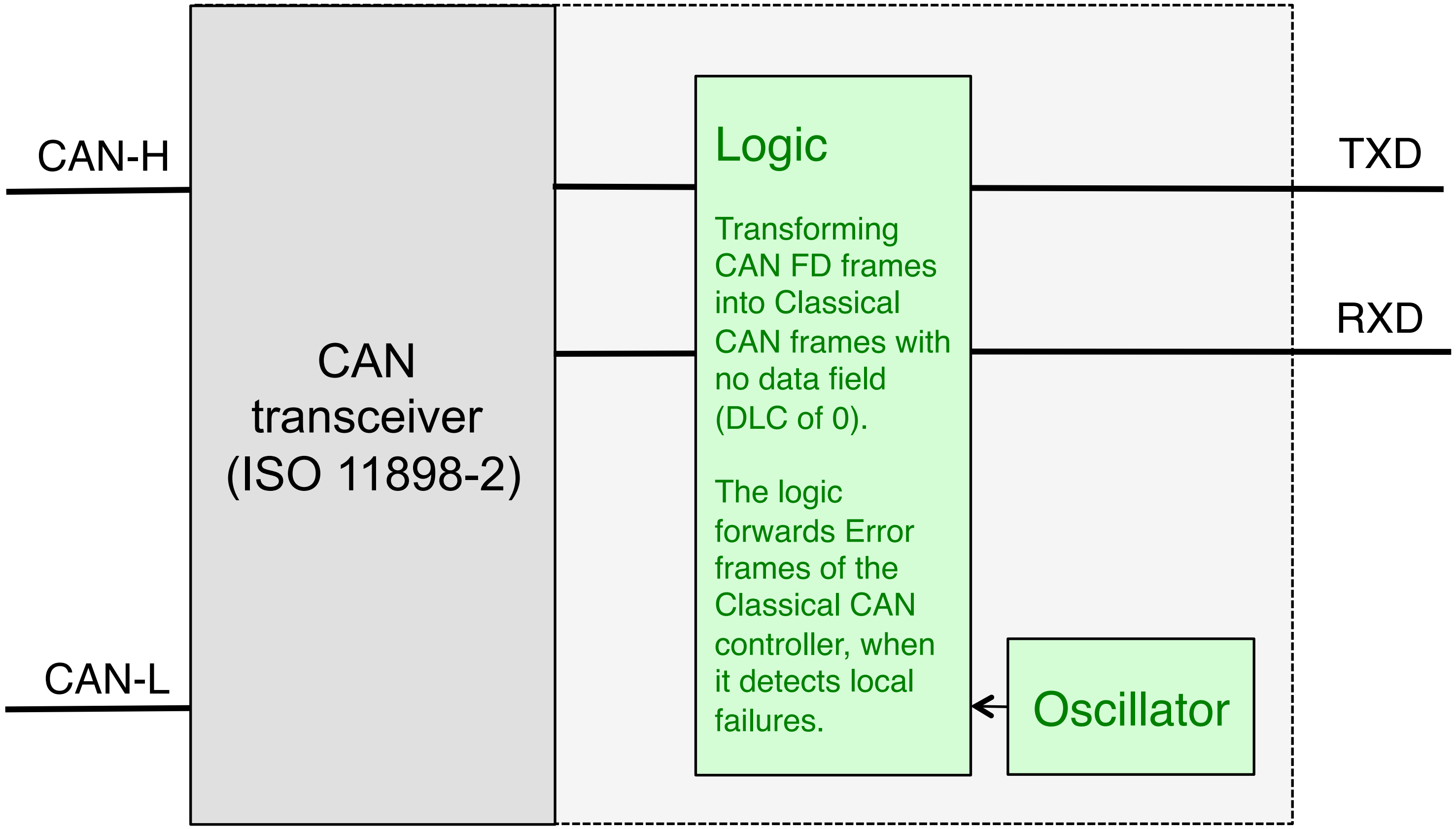
Another solution is the strict separation of CAN FD and Classical CAN communication running on two network segments, which are interconnected by means of a router device.

FD “shield” transceiver



When the Classical CAN controller runs into error passive state, the FD shield transceiver generates an active error frame on behalf of the Classical CAN controller, in order to guarantee network-wide data consistency. Therefore it implements a Receive Error Counter.

FD “filter” transceiver



The FD “filter” transceiver requires a pre-configured bit timing. The delay caused by the added logic is in the range of a few nanoseconds.

Additional specifications

Already released CiA internally as Draft Standard Proposal (DSP):

- CiA 601-1: CAN FD node and system design – Part 1: Physical interface implementation
- CiA 601-4: CAN FD node and system design – Part 4: Ringing suppression

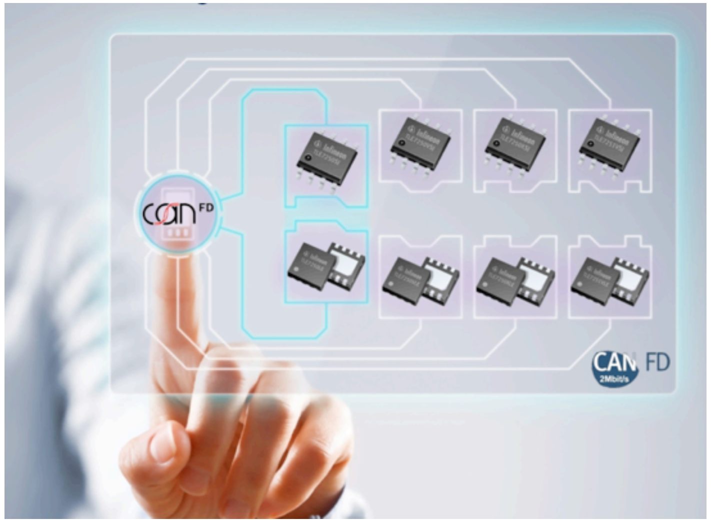
(personalized copies are available on request also for non-CiA members)

Under development:

- CiA 601-2: CAN FD node and system design – Part 2: Controller interface recommendation
- CiA 601-3: CAN FD node and system design – Part 3: System design recommendations
- CiA 601-5: CAN FD node and system design – Part 5: Reference topologies for non-automotive applications
- CiA 603: CAN time-stamping (also for Classical CAN implementations)

CAN

Contents of presentation



- Classical CAN and CAN FD – The basics
- CAN FD data link layer protocol – Some details
- CAN FD physical layer – More details
- CAN FD system design – The challenge
- **CANopen FD – The next generation**

HLP history

- 1991: CAN Kingdom
- 1992: CAN Application Layer (CAL)
- 1992: CAN calibration protocol (CCP)
- 1994: Smart Distributed System (SDS)
- 1994: IEC 62026-3 (DeviceNet)
- 1994: SAE J1939 (Trucks and buses)
- **1994: EN 50325-4 (CANopen)**
- 1999: ISO 11992-2/-3 (Truck/trailer)
- 2000: IEC 61162-3 (NMEA 2000)
- 2002: ISO 11783 (Isobus)
- 2004: ISO 14229/15765-2 (OBDII/ISO-TP)
- 2007: Arinc 825
- 2010: Universal measurement and calibration protocol (XCP)
- **2013: IEC 61375-3-3 (CANopen in rail vehicle)**

CANopen FD

CiA 301 version 5.0

- NMT: No protocol change
- Heartbeat: Not protocol change
- SDO: New USDO protocol
- PDO: Up to 64 byte
- EMCY: No protocol change
- SYNC: No protocol change
- TIME: No protocol change
- SRDO: Not yet discussed
- (LSS: Not yet discussed)

PDO with up to 64 byte

- More process data can be mapped into a single PDO. This could improve the throughput even without higher data-phase bit-rates.
- More use cases for the optional dummy mapping (broad- and multicast PDOs).
- Existing CANopen (device) profiles need to be updated.
- Bit-wise mapping is not more recommended, (there are just 64 mapping entries).
- In some applications, padding of unused bytes is necessary (proposal: 55_h or AA_h with as less as possible stuff-bits).
- (No remotely requested PDOs anymore)

USDO download (preliminary)

Client **Server**

USDO download request

Byte	0	1	2 to 3	4	5	6	7 to 63
	Destination address (DA)	**Command specifier**	**Index**	**Sub-index**	**Data type**	**Size**	**Application data**

DA	Description
00_h	Broadcast
01_h to $7F_h$	Unicast
80_h to FF_h	Multicast

USDO download response

Byte	0	1	2 to 3	4
	Destination address (DA)	**Command specifier**	**Index**	**Sub-index**

The CAN-ID identifies the sender of the telegram

Client to Server: 600_h + Client_Node_ID

Server to Client: 580_h + Server_Node_ID

USDO download (preliminary)

Client **Server**

USDO download request

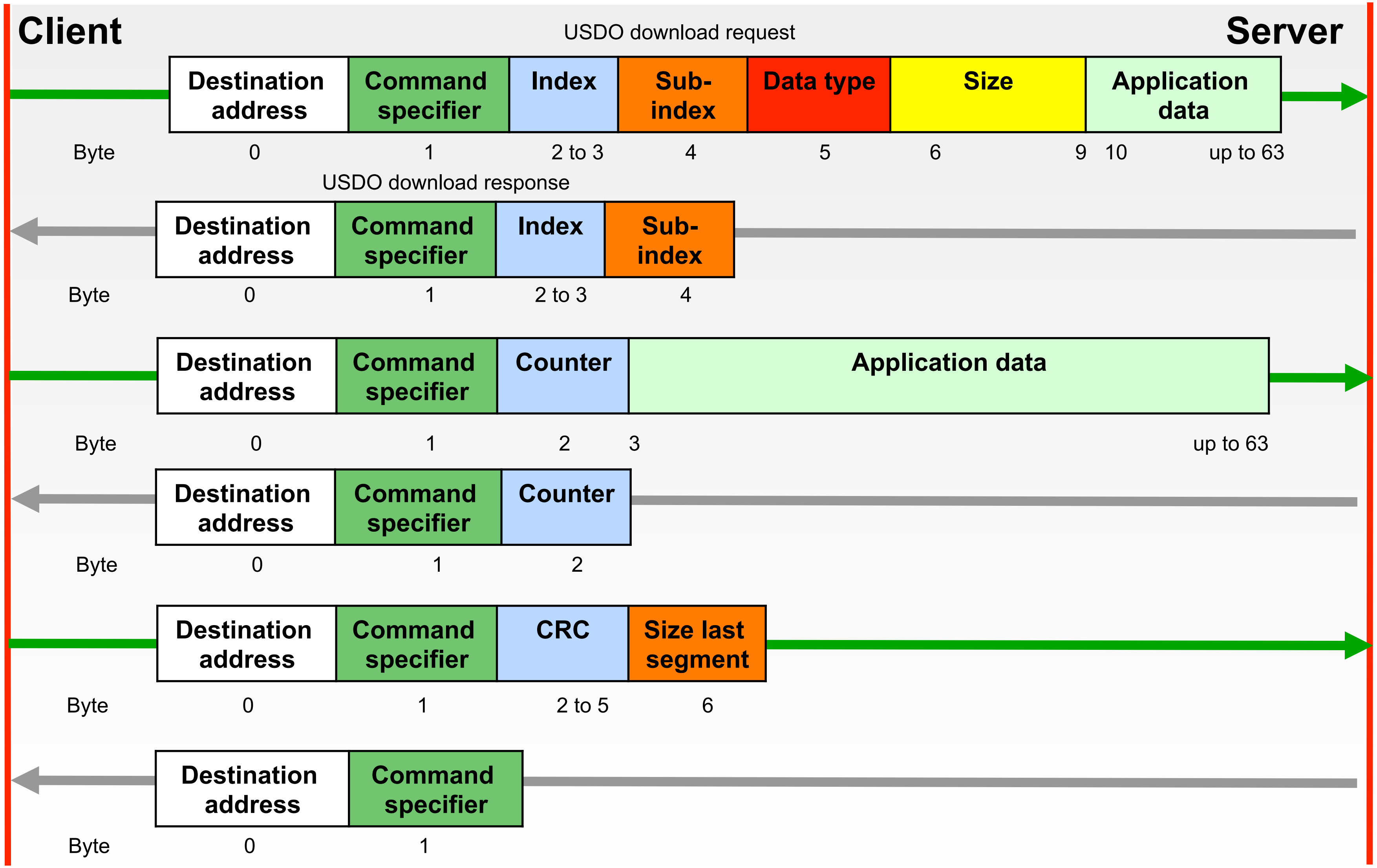
Destination address	Command specifier (CS)	Index	Sub-index	Data type	Size	Application data
Byte 0	1	2 to 3	4	5	6	7 to 63

CS	Description
01/02	Client/Server CS upload expedited (single frame)
03/04	Client/Server CS download expedited (single frame)
...	Multiple sub-index access
...	Segmented USDO transfer
...	Long distance routing
...	Functional addressing

USDO download response

Destination address	Command specifier (CS)	Index	Sub-index
Byte 0	1	2 to 3	4

USDO download segmented

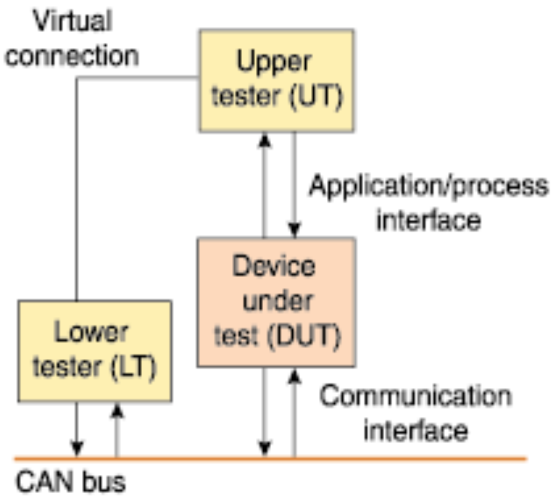


Other CANopen documents

- CiA 302-2 and CiA 302-3 (programmable devices with NMT master functionality) update for CANopen FD
- CiA 311 (XDD file specification) improvements for CANopen FD
- CANopen device and application profile updates supporting "long" PDOs (CiA 401, CiA 402, etc.)
- CiA 6XX-X (CANopen FD physical layer specification references)

Conformance testing

- CiA 310: CANopen FD conformance test plan (CTP)
- CiA provides approved XDP files for CANopen FD
  XDD checker tool
- CANopen FD XDD checker tool
- CANopen FD lower (and upper) tester



Interoperability testing

- Option 1: “Golden” system testing
- Option 2: CANopen FD plug-fests

Roadmap schedule

- CANopen FD implementation workshop in September 2016
- Complete CANopen FD roadmap (for product release and sales) in October 2016
- Planned release of CiA 301 version 5.0.0, CiA 302-2 and CiA 302-3 in November 2016 as Draft Standard Proposals (for prototyping)
- CANopen FD plug-fests in spring 2017

Necessary actions

- Additional experts for CiA document development (CiA 301, CiA 302 series, CiA 310, CiA 311, CiA 402, etc.)
- Approved XPD files for CiA 301, etc. (CiA)
- XDD checker tool
  (Emtas and Vector are prepared to provide it free-of-charge)
- CANopen FD conformance test tool
  (tender to be invited)

Questions and further info

Technical info: www.can-cia.org *and* service@can-cia.org

General (product) information: www.can-newsletter.org