ENGINEERING

FH JOANNEUM
Electronic Engineering

Testing of Automotive Systems (Part I)

Module 4 – CAN / LIN

David Ludwig , Magna Steyr

Schedule

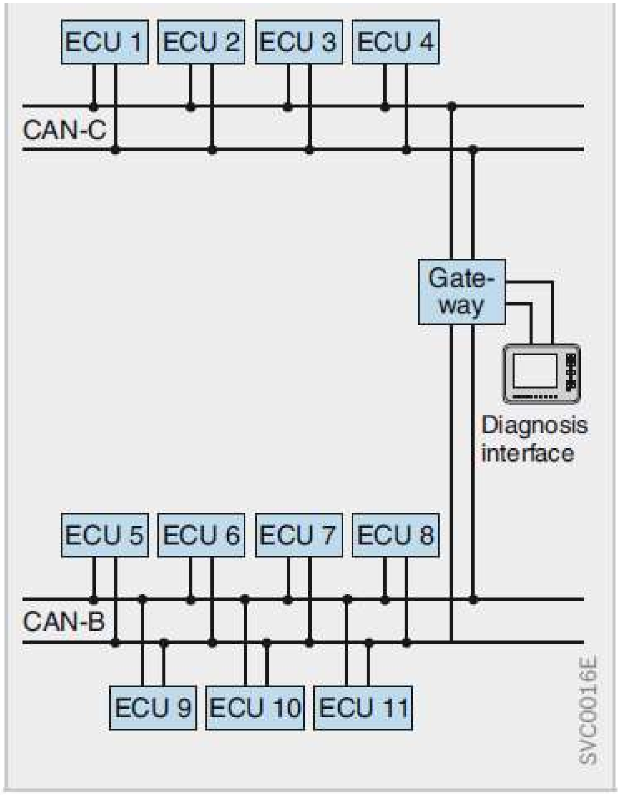
CAN-BUS

CAN-Bus

- Firstly introduced 1991, still standard in automotive
- Used in various domains in vehicles
- Different data rates are used

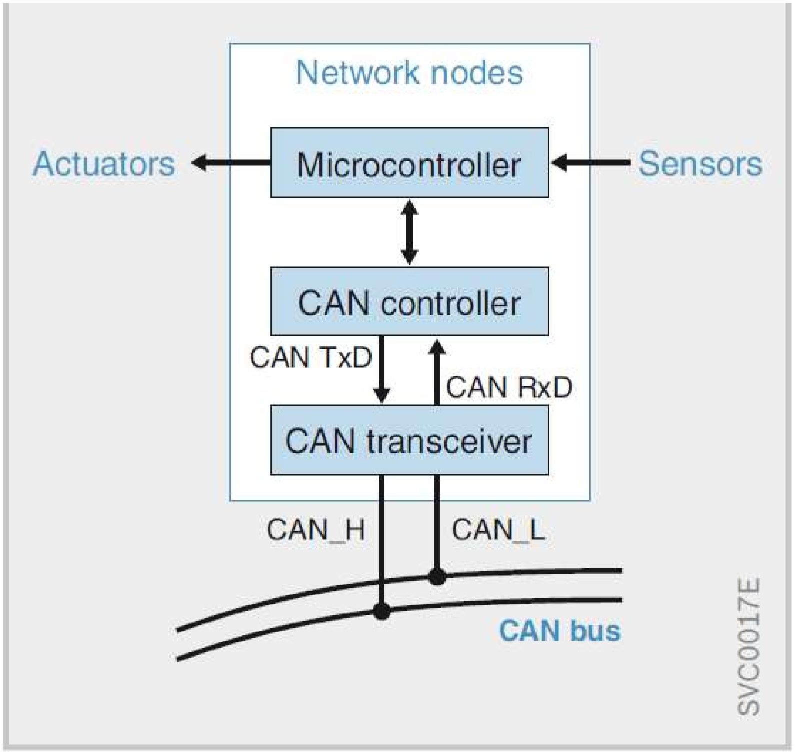
  - CAN-C 125kbit-1Mbit/sec
    - „high-speed CAN“ 500 kbit/sec
    - Still in almost every vehicle

  - CAN-B 5-125kbit/sec
    - „low-speed CAN“ 125 kbit/sec
    - More fault-tolerant
    - less used



Network nodes

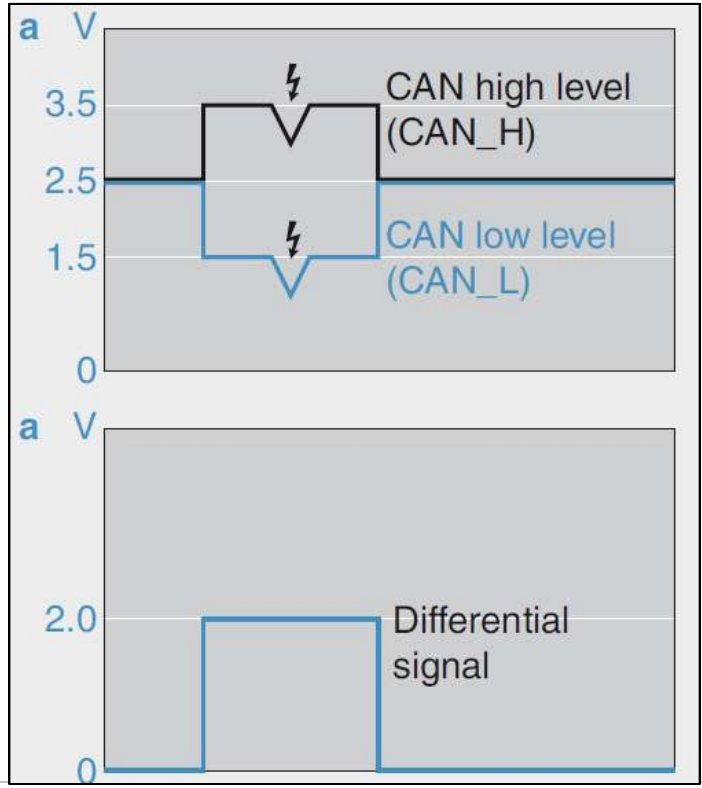
- All network nodes are connected to a bus and each node is able to receive all information sent on the bus
- The two bus lines are designated CAN_H and CAN_L
- A network node comprises of
  - Microcontroller
    - runs the application program
    - controls the CAN controller
    - prepares the sent data and evaluates received data
  - CAN Controller
    - responsible for the transmit and receive modes
    - generates the data communication bit stream
    - forwards it to the transceiver on the TxD line
  - CAN Tranceiver
    - Signal amplification, generates voltage levels
    - transmits the processed bit stream serially



Logical bus states

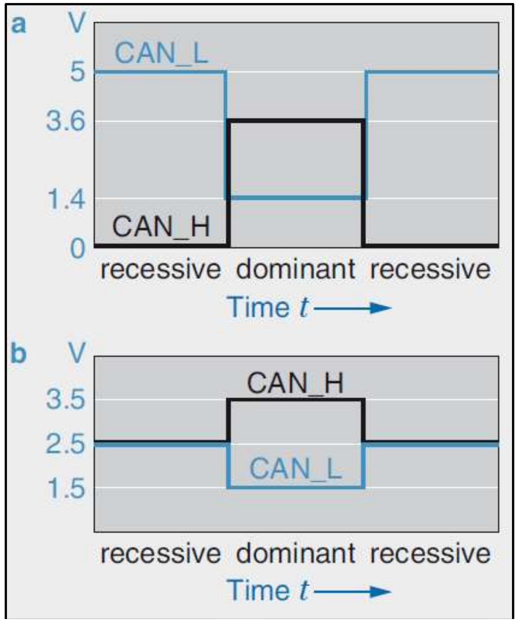
- Two states created by CAN controller
  - dominant binary “0”
  - recessive “1”
  - NRZ = non-return-to-zero
- Unshielded twisted pair cables (UTP, diameter between 0,34 and 0,6 mm²)
- Disturbance pulses have effect on both lines
- Differential amplifier subtracts CAN_L from CAN_H level
- Additional shielding reduce self emissions

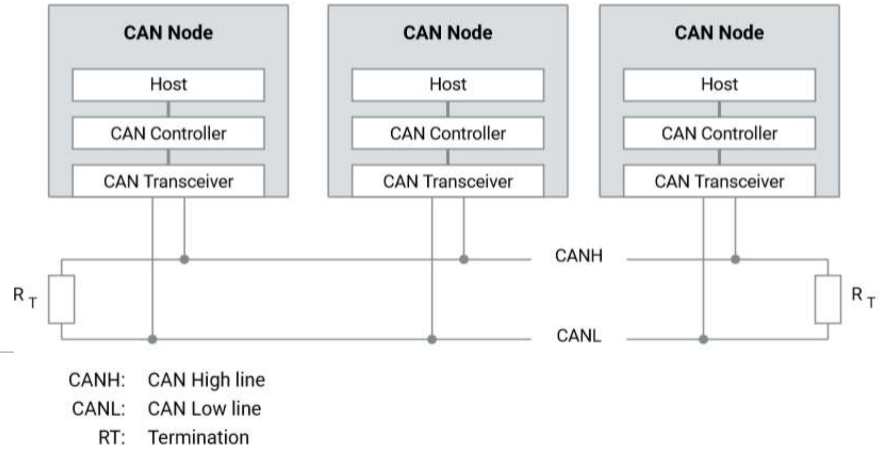
CAN great benefit -> it is extremly robust because of twisted pairs.



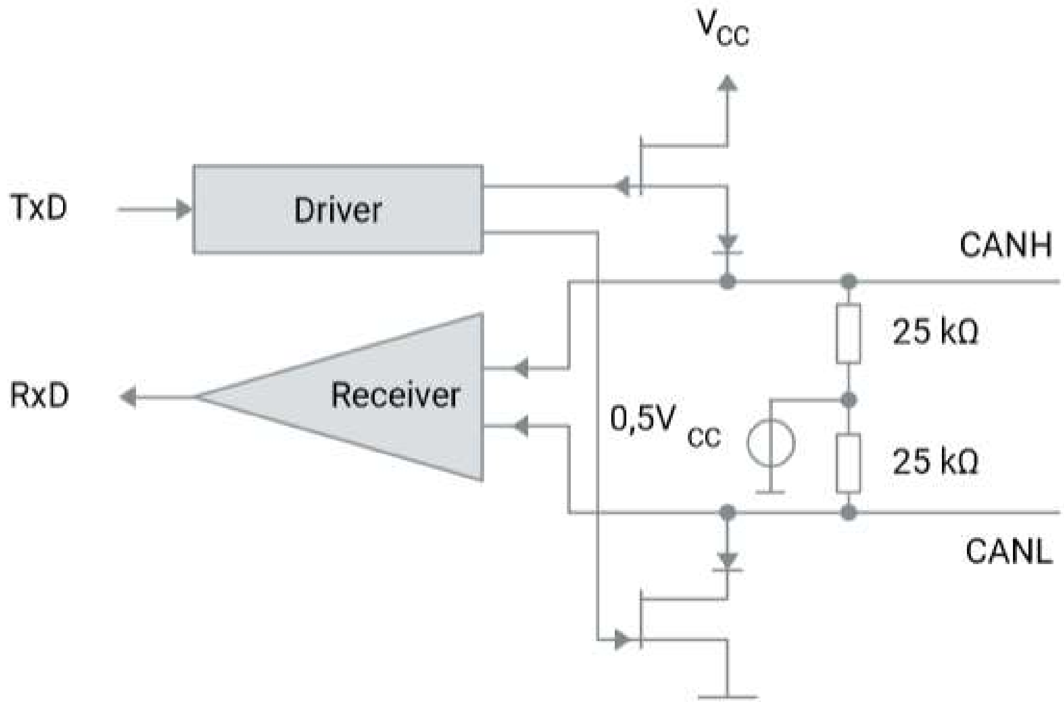
Electrical bus states

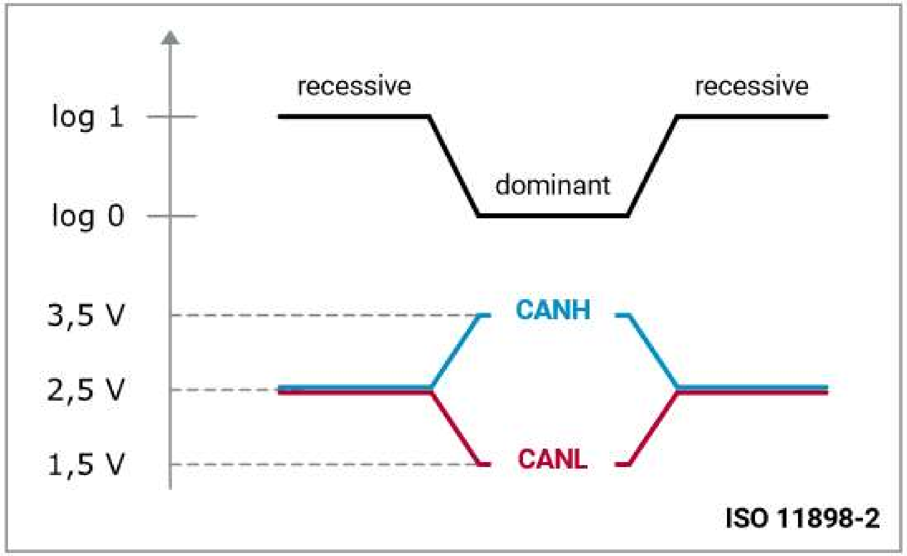
- CAN transceiver converts digital to voltage signals
  - Low-Speed CAN (a)
    - dominant CAN_H 3.6V / CAN_L 1.4V
    - recessive CAN_H 0V / CAN_L 5V
  - High-Speed CAN (b)
    - dominant CAN_H 3.5V / CAN_L 1.5V
    - recessive CAN_H 2.5V / CAN_L 2.5V
- Terminal resistor RT= 120Ω to dampen reflections (b)
- Maximum number of nodes 32 according ISO11898
- Maximum bus length (recommendation)
  - 500kbaud 100m
  - 125kbaud 500m





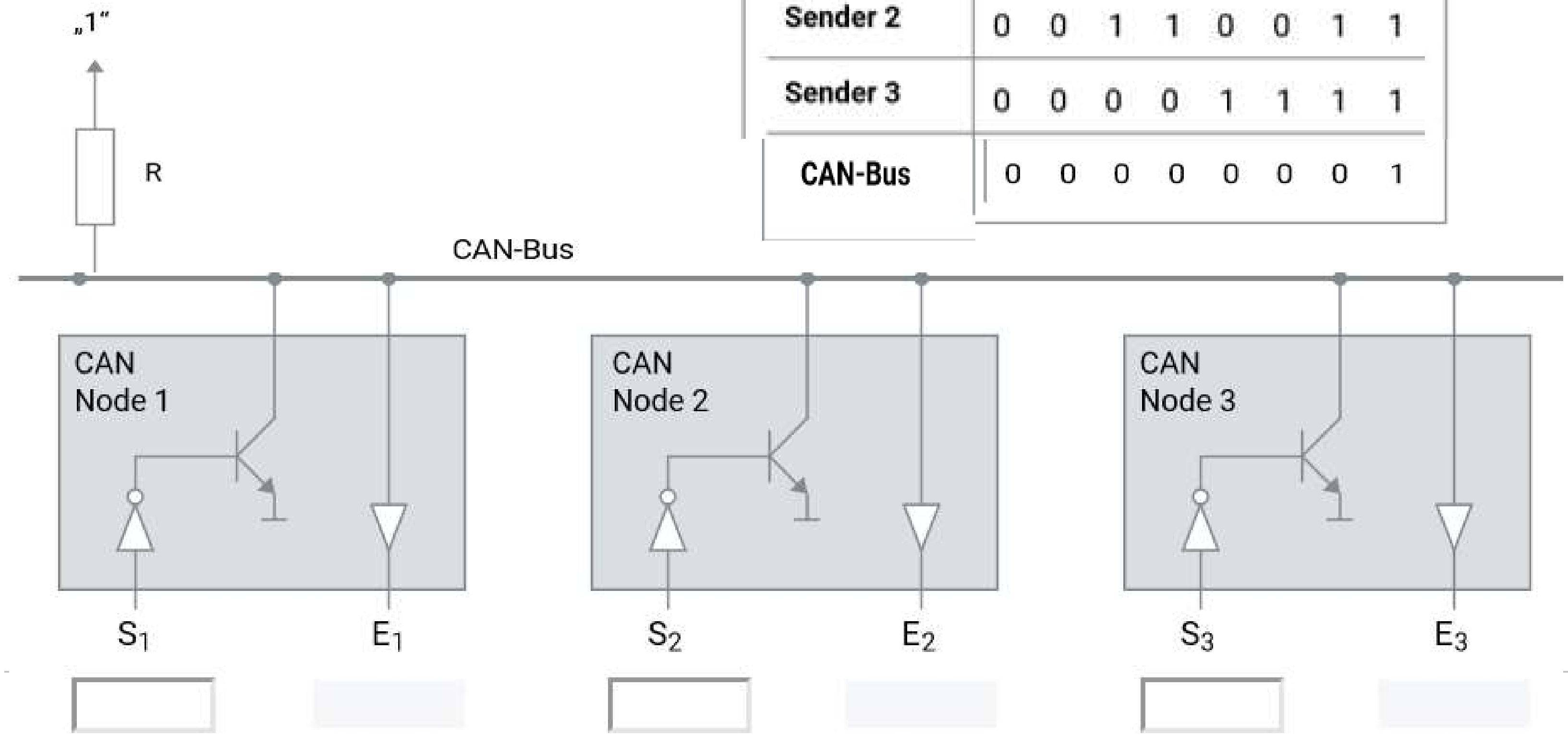
CAN Tranceiver





Exercise: Open Collector Buslogic

12 V
~ 0,7 V
~ 50 mA
~ 4 Ω
~ 0 V
~ 12 V
leitet
geschlossen



Wired-AND-Logik

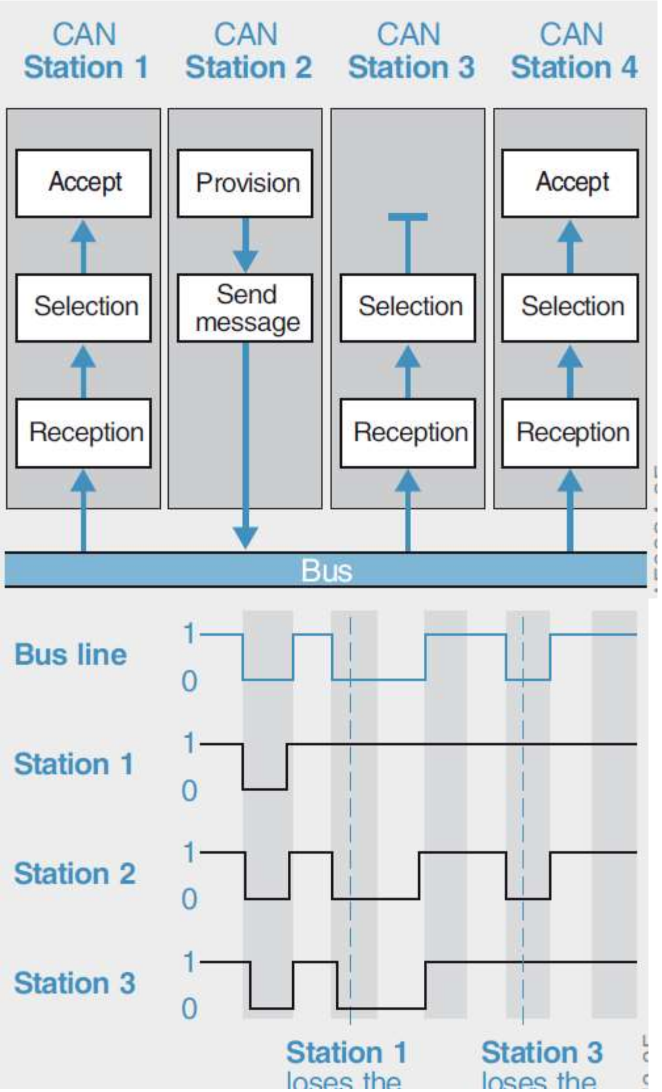
Sender 1	0	1	0	1	0	1	0	1
Sender 2	0	0	1	1	0	0	1	1
Sender 3	0	0	0	0	1	1	1	1
CAN-Bus	0	0	0	0	0	0	0	1

As long as we have a zero on one it will be pulled to zero.

There is no master, kind of democracy with no refree.

CAN Communication

- Multi-Master Principle
  - Each node may send messages any time
  - If bus is free and arbitration has been passed
- Content-based addressing
  - No addressing of nodes but of messages
  - Identifier classifies message content
  - Messages are broadcasted to all stations
  - Read only those messages which are stored in acceptance list
- Arbitration
  - Message begins with a dominant bit (start-of-frame bit), followed by the identifier
  - Message with highest priority is assigned first access
  - With highest priority bus access ~300µs
  - The higher the busload the bigger the latency



Arbitration and priorisation

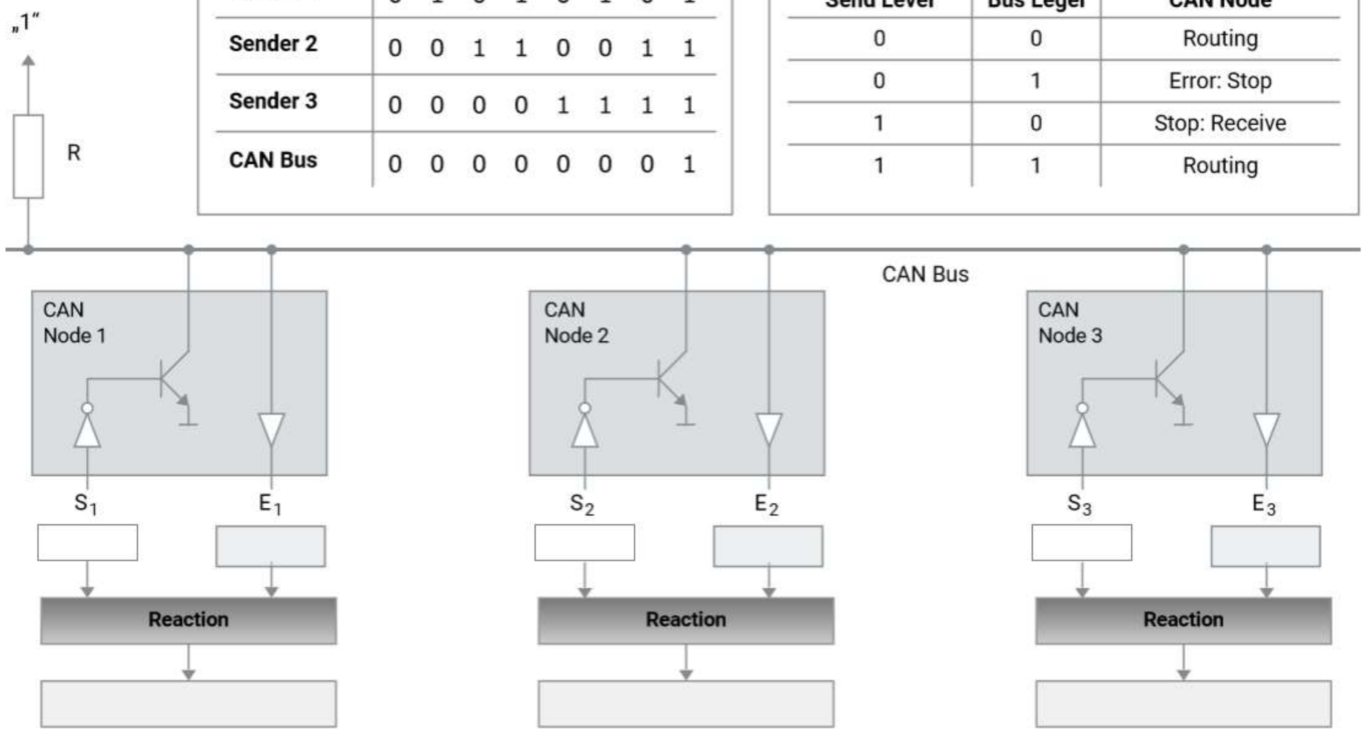
- Bitwise arbitration, wired-AND logic
- To get access to the bus a node needs message with the highest priority
- The smaller the Identifier value the higher the priority

Wired-AND-Logik

Sender 1	0	1	0	1	0	1	0	1
Sender 2	0	0	1	1	0	0	1	1
Sender 3	0	0	0	0	1	1	1	1
CAN Bus	0	0	0	0	0	0	0	1

Arbitration Logic

CAN Node/ Send Level	CAN Bus Legel	Reaction of CAN Node
0	0	Routing
0	1	Error: Stop
1	0	Stop: Receive
1	1	Routing

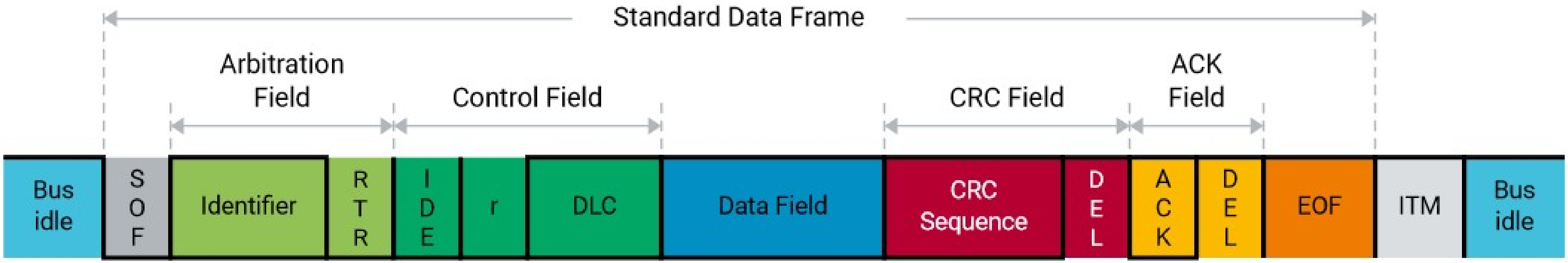


CAN Message Format

- Data Frame
  - Transmitted message contains data (e.g. current engine speed) that is provided by transmitting station (data source)
- Remote Frame
  - Stations can call-in data they need from the data source
  - Example: windshield wiper requests how wet windshield is from rain sensor
- Error Frame
  - If station detects a fault or error
- Overload Frame
  - create a delay between preceding and subsequent data or remote frame

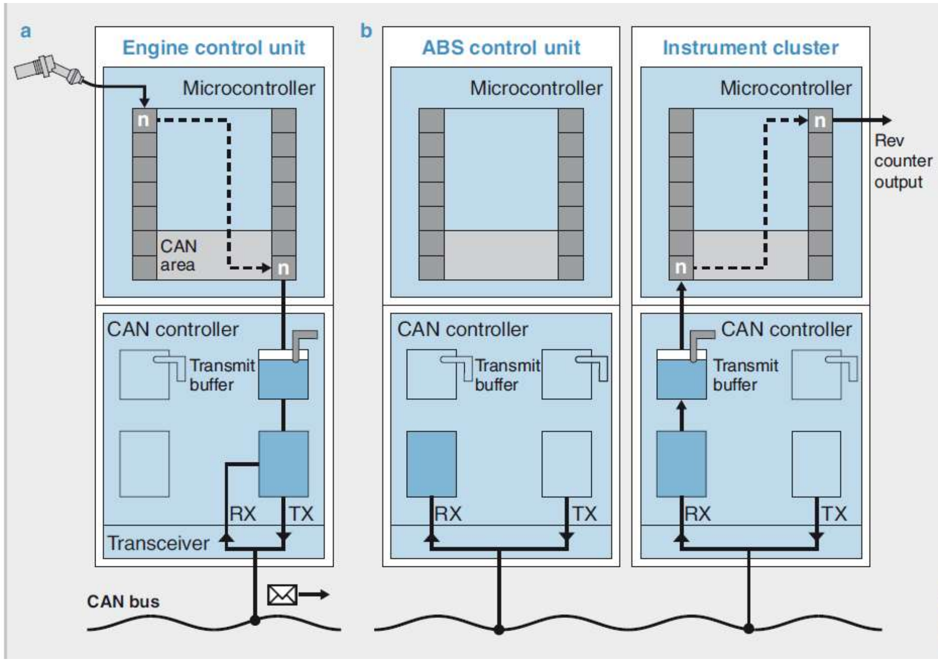
CAN Message Format

- SOF → start of frame represented by a dominant bit
- Arbitration field → 11-bit or 29-bit (extended) identifier and remote transmission request
- Control field → Identifier extension and number of data length code (DLC) in data field
- Data field → Actual message information 0...8 byte
- CRC field → 15-bit cyclic redundancy checksum (Generator polynom G(x))
- ACK field → Divison frame/polynom →acknowledge receipt by the **receiver**
- EOF → End of frame with 7 recessive bits
- ITM → Interframe space, separate successive messages



Data transfer sequence

- Engine ECU calculates engine speed from sensor on microcontroller level
- CAN Controller compiles CAN frame and generates bitstream if bus is free
- Bus availability check via CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance)
- Tranceiver generates electrical signal and puts it on the bus
- Signal is received by all stations
- CAN controller of receiving node checks message for errors and acceptance, reject or receives message

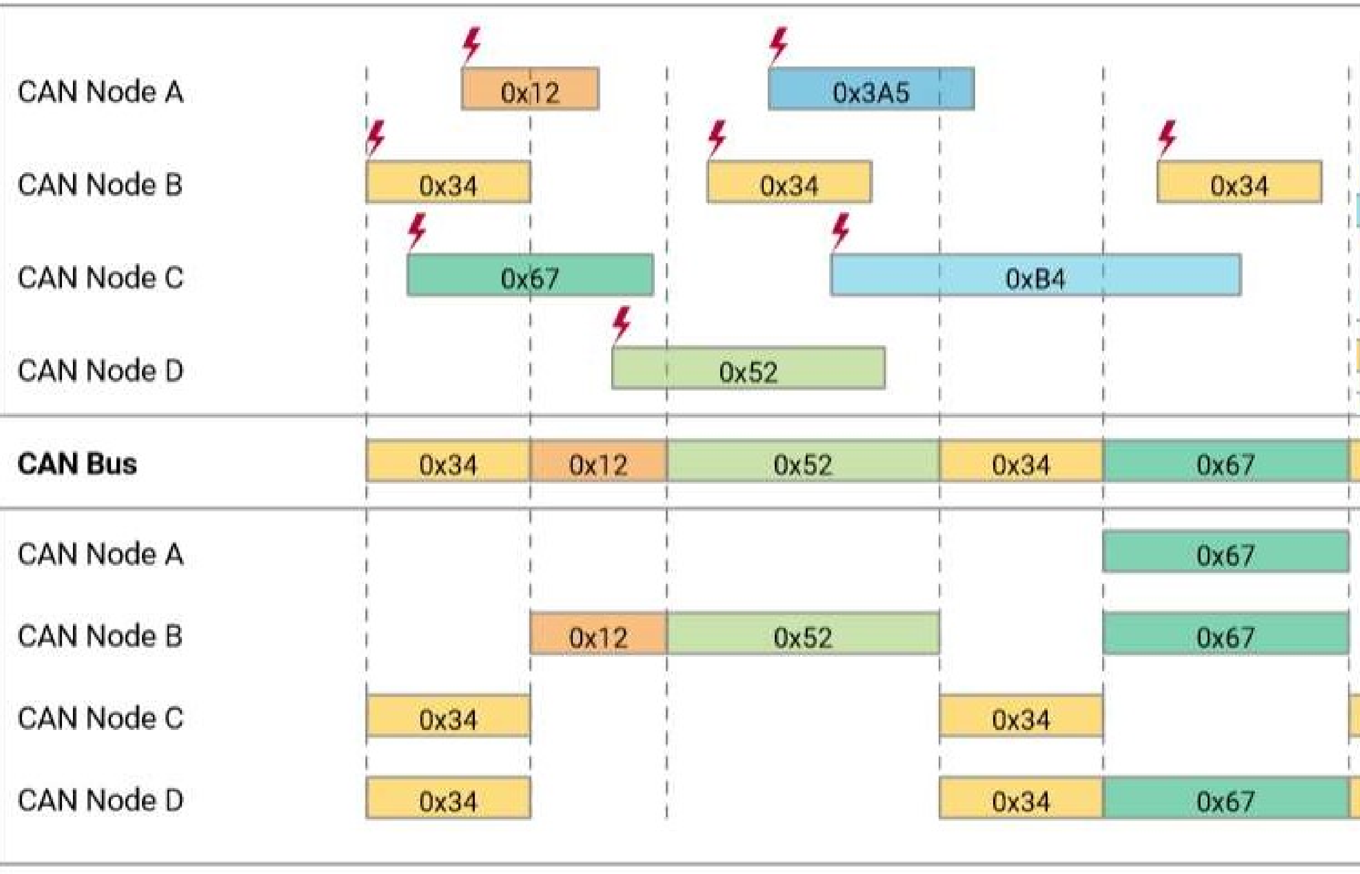


All transmitter/receiver relationships including the meaning of the messages are described in the **communication matrix**

https://elearning.vector.com/mod/page/view.php?id=343

Now it is getting stressful, because the messages are higher and it happens that a lot of messages wait too long.

Exercise: Typical CAN-Communication



Communication Matrix

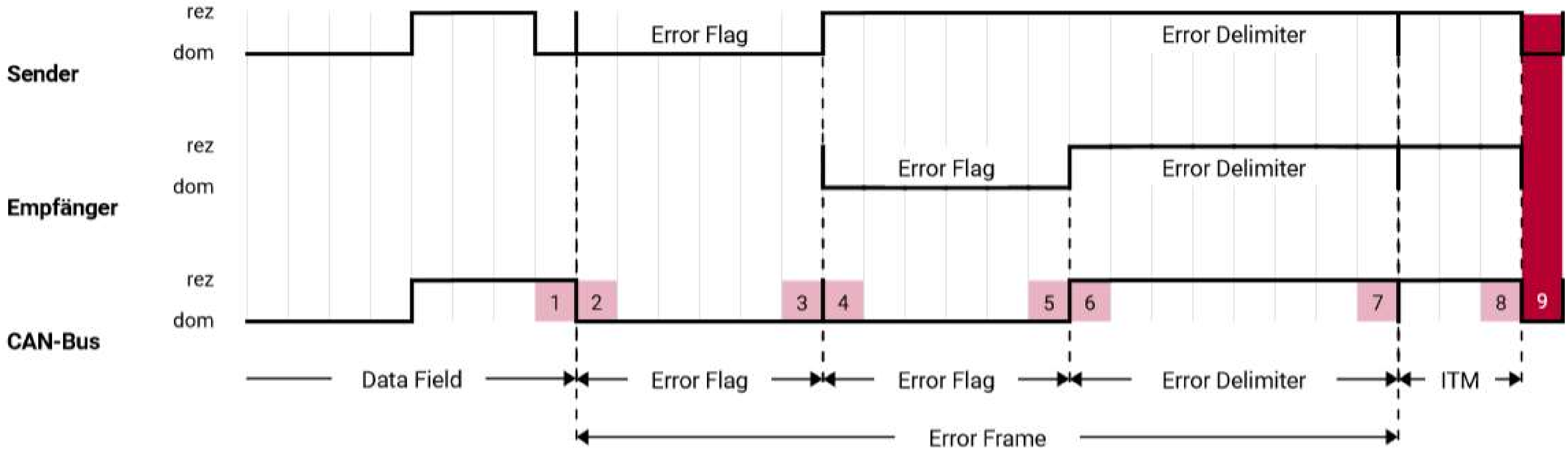
Data Frame	CAN Knoten A	CAN Knoten B	CAN Knoten C	CAN Knoten D
0x12	Sender	Empfänger		
0x34		Sender	Empfänger	Empfänger
0x52		Empfänger		Sender
0x67	Empfänger	Empfänger	Sender	Empfänger
0xB4	Empfänger		Sender	
0x3A5	Sender	Empfänger	Empfänger	Empfänger

CAN Error processing

- If an error is detected by a node , message transfer is aborted
- All nodes have to be informed by the node who has detected the error
- Error flag is put on the bus by six dominant bits
- This is a violation of the bitstuffing rule
  - Bitstuffing rule: after five bits of same kind transmission of one complementary bit
- Violation of bitstuffing rule causes error flag at all other nodes
- After error flag, delimiter and intermission data frame is send again (if allowed by arbitration)
- Error counters detect recurring errors, CAN-Controller will be cut from bus at counter overload

Error processing

- 1-2 Bus monitoring error detected by sender, error flag set and bit stuffing rule violated
- 3-4 Receiver detects bitstuffing rule violation and stets error flag, sender in delimiter state
- 5-6 Receiver has finished error flag, both sender and receiver go in delimiter mode
- 7-9 Delimiter and intermission finished, data frame starts again with SOF

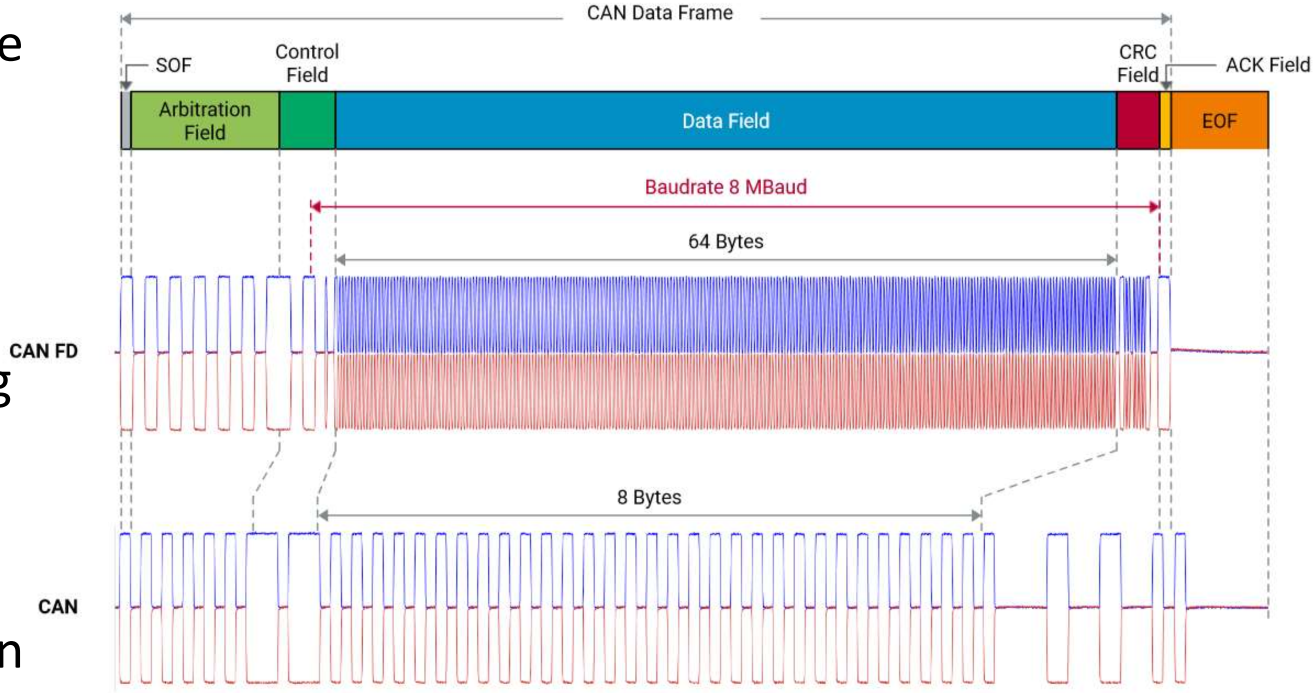


CAN-FD

FH | JOANNEUM
Electronic Engineering

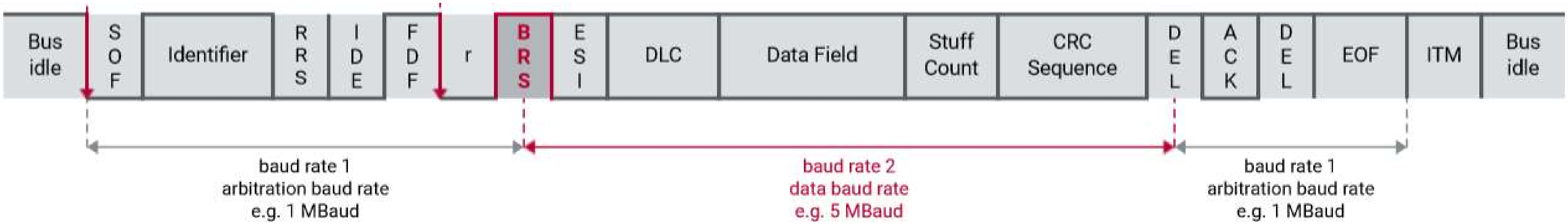
CAN FD Motivation and principle

- Bandwidth requirements increase due to additional vehicle functions
- General exchange CAN by other bus technologies with higher bandwidth not suitable regarding costs and development efforts
- Bandwidth limitating factor for CAN is parallel node communication during arbitration (and acknowledge ACK) period
- Idea: Increase the bandwidth only during data field



CAN FD Motivation and principle

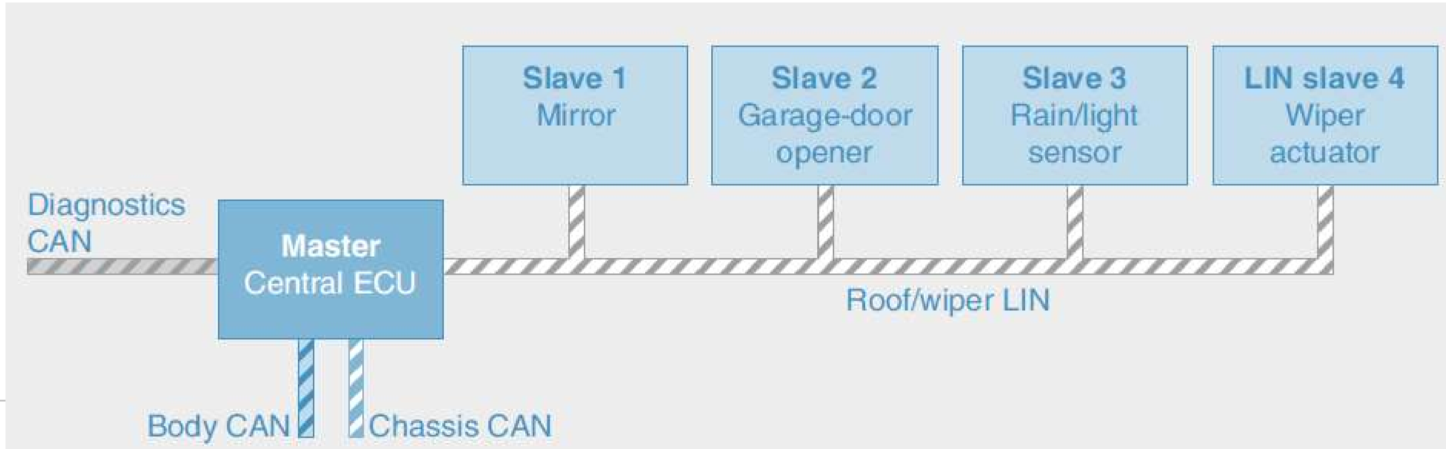
- CAN Reserve bit is used as CAN FD indicator
- FDF (Flexible data rate format) indicates transmission of CAN or CAN FD frame
- Bit rate switch (BRS) announces increase of baud rate
- Switch back to baudrate 1 at CRC delimiter
- Backwards compatibility to CAN, but no upwards compatibility



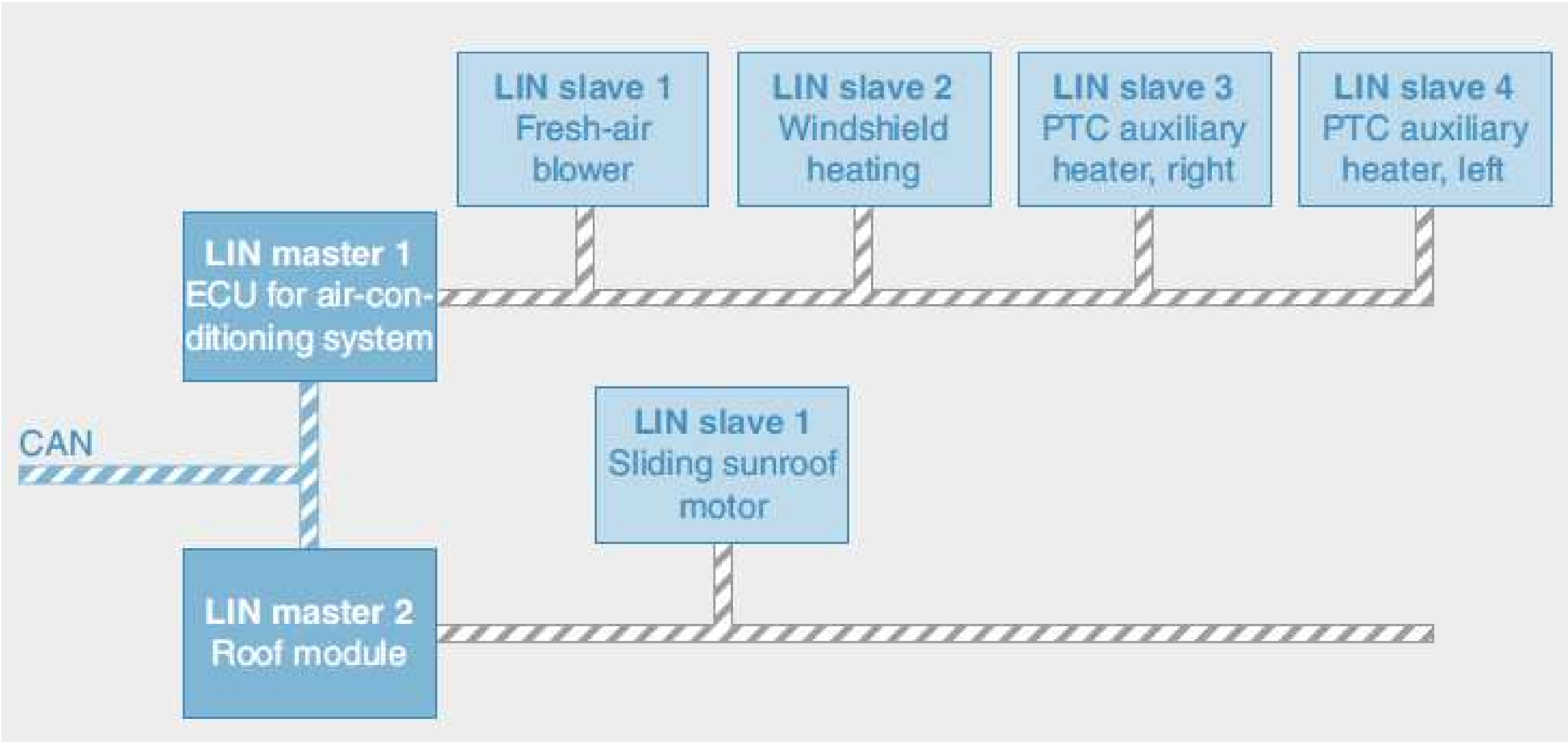
LIN-BUS

LIN (local interconnect network) – Bus

- Low-cost unshielded single-wire bus system
- Low data rates max 20kbit/sec and limited to 16 bus subscribers
- Local subsystem within demarcated installation space (e.g. door)
- Master/slave topology with connection to superordinate CAN
- Time-synchronous communication where master defines time grid
- Mainly comfort appications: Door modules, sunroof, seat adjustment etc.

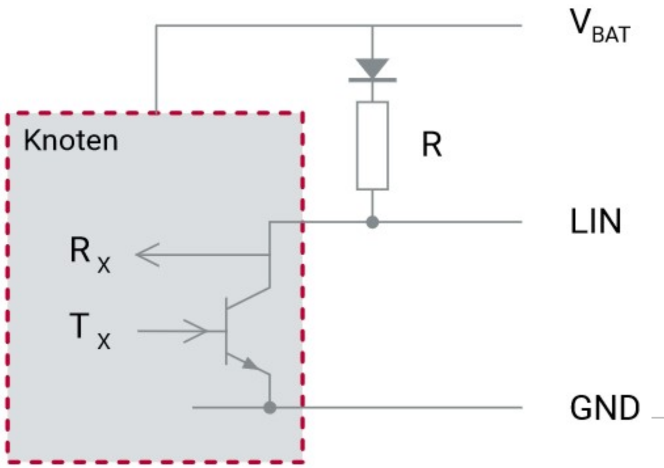


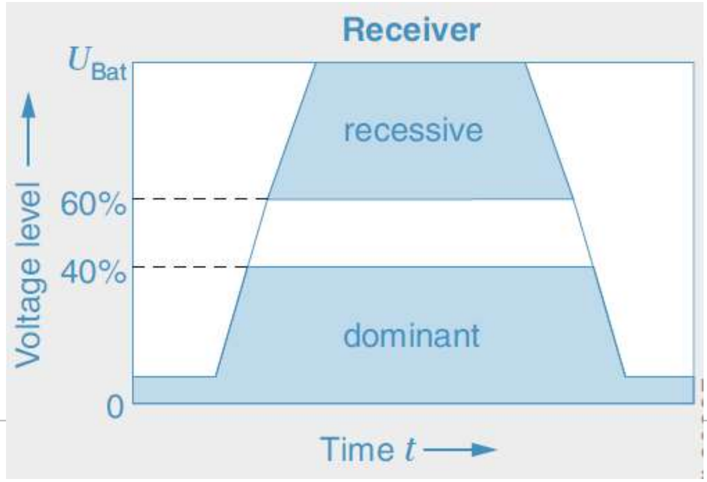
LIN networking example – AC control

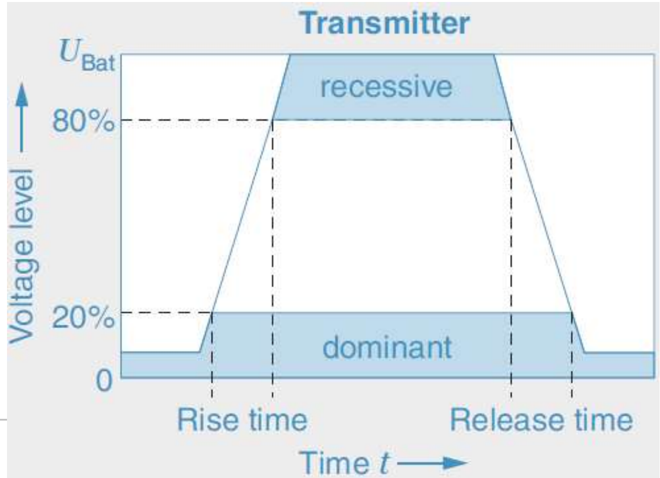


LIN data transmission system

- Open collector circuit, wired-AND logic same as CAN
- Dominant level corresponds to ~ 0V (vehicle ground) and represents logical 0
- Recessive level corresponds to VBAT and represents logical 1
  - Pull-up Resistance 1kΩ (master) and 30kΩ (slaves)
- Stable data transfer by means of tolerance zones for transmitter and receiver
- Data rate limited to 20kbit/sec

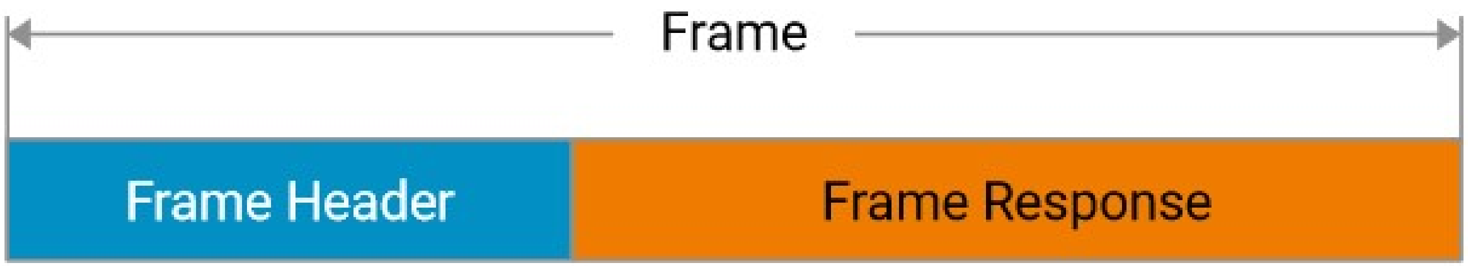






LIN bus access – message format

- Master initiates slave(s) response with frame header
- Slaves answer with frame response
- Deterministic„Delegated token“ bus access → No collisions or arbitration
- Predictable data transfer, defined **schedule**
- LIN configuration in *.ldf-file
- LIN standard describes several types of frames
- Several types of frame, unconditional frame described here, unique header and response

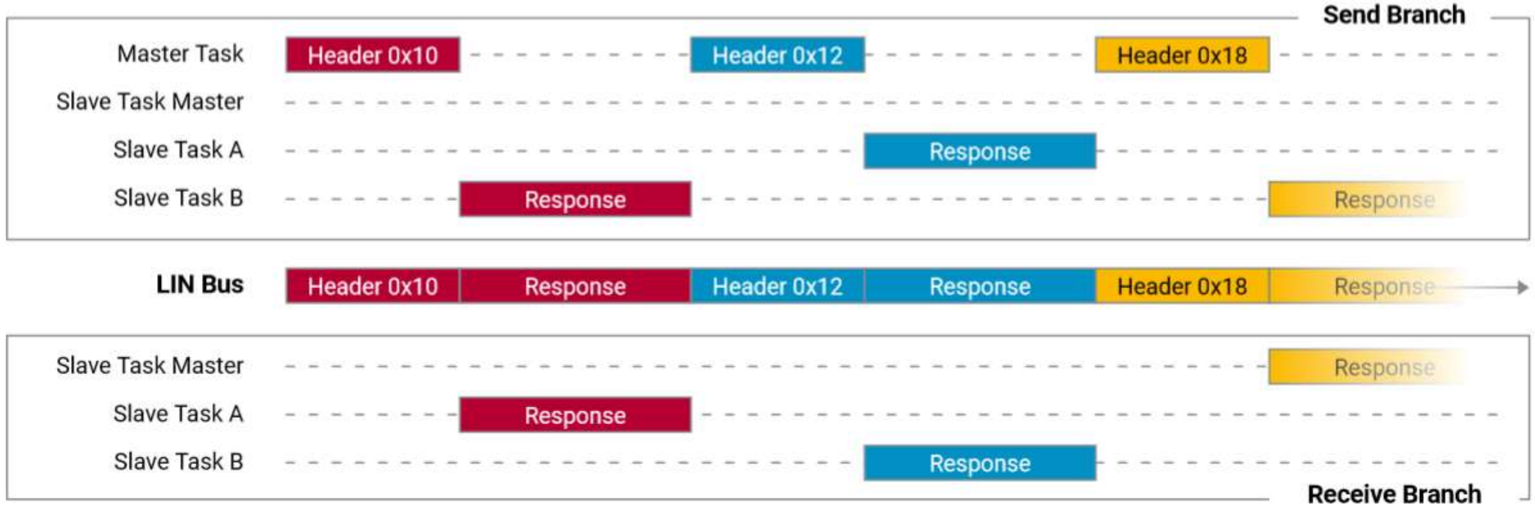


LIN Schedule

Communication Matrix

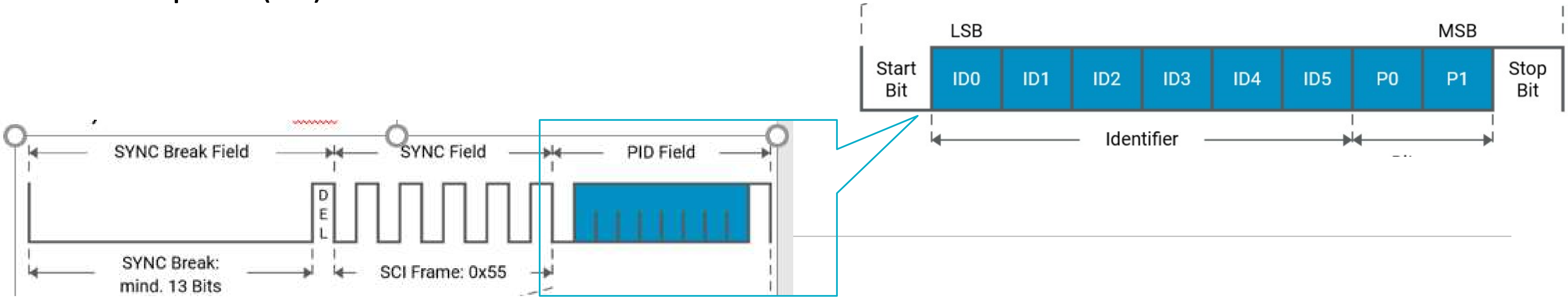
Communication Cycle

Schedule		Frame \ Slave Task	Slave Task Master	Slave Task A	Slave Task B
T_1	Frame Slot 1	Unconditional Frame ID=0x10		Receiver	Sender
T_2	Frame Slot 2	Unconditional Frame ID=0x12		Sender	Receiver
T_3	Frame Slot 3	Unconditional Frame ID=0x18	Receiver		Sender
T_4	Frame Slot 4	Unconditional Frame ID=0x1C	Receiver	Sender	Receiver
T_5	Frame Slot 5	Unconditional Frame ID=0x20	Receiver		Sender
T_6	Frame Slot 6	Unconditional Frame ID=0x24	Sender	Receiver	



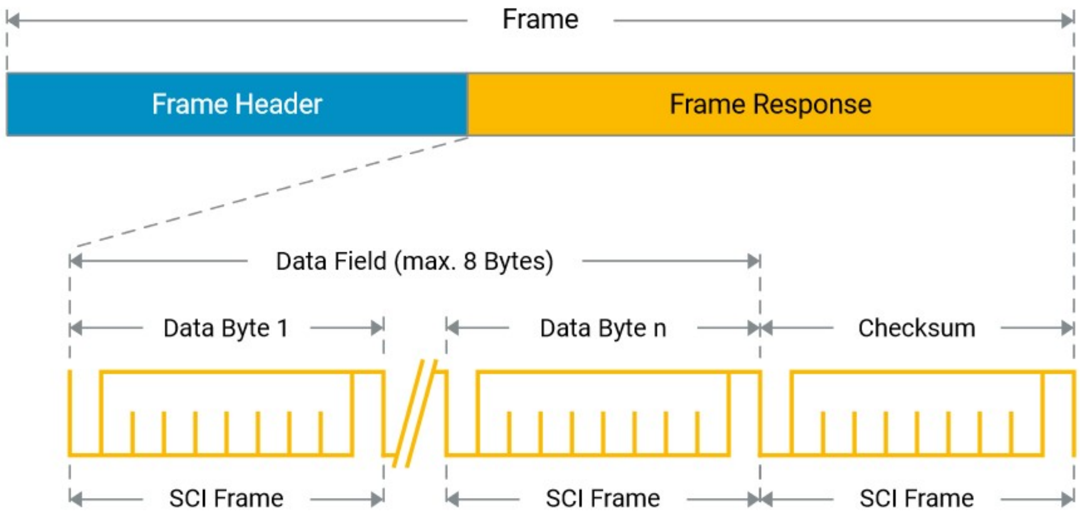
LIN Frame header

- Master task
- Sync break field initiates begin of the header
- Sync Field includes clock frequency and is used for synchronization with slaves
- Protected identifier (PID) field
  - Unique identifier for slave communication, defined in *ldf
  - 6-bit identifier + 2 parity bits
- Between header and response frame there is a pause time called response space (RS)



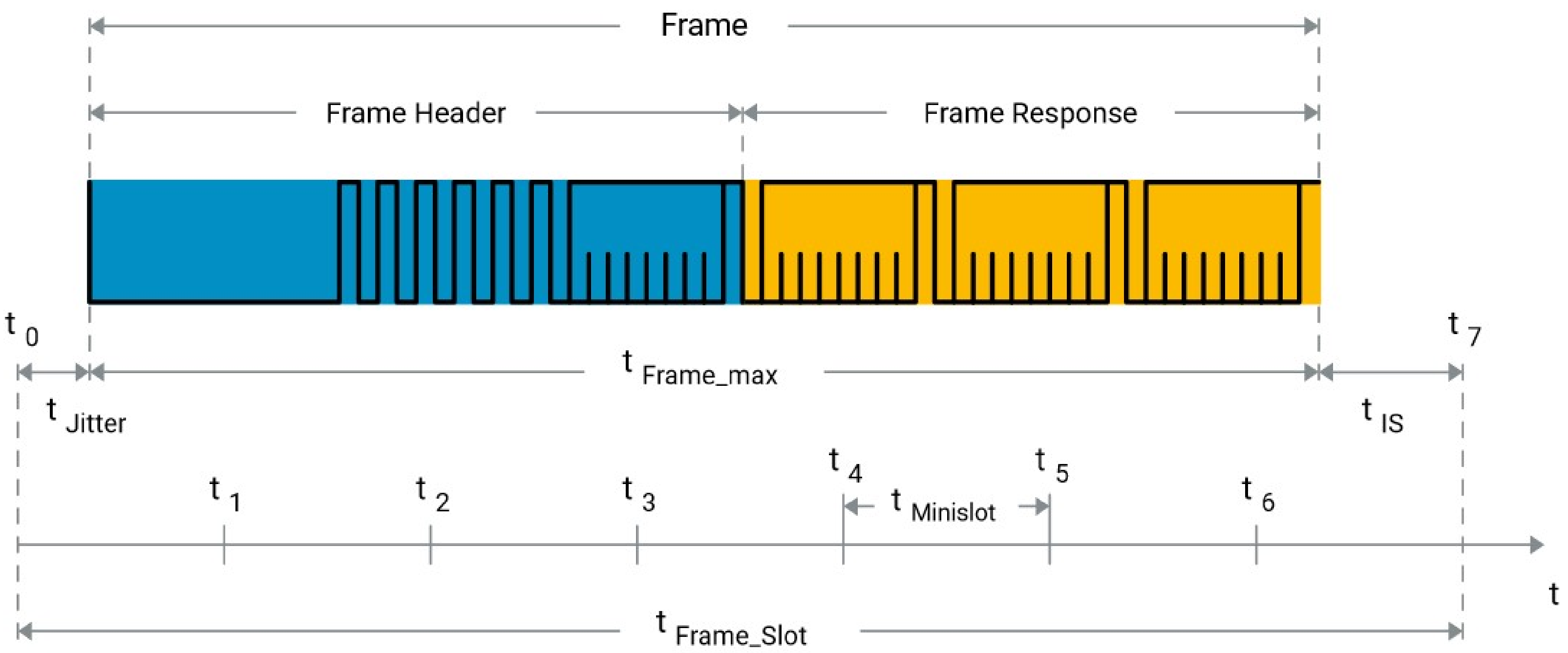
LIN Frame response

- Slave task
- Slaves answer with max. 8 byte response to the frame header (data field)
- Also slave task of the master can be accessed
- Bytes are transferred from LSB to MSB



LIN Frame – time base

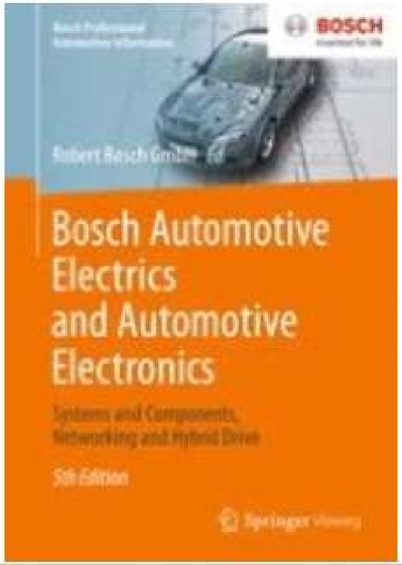
- The LIN schedule is organized in time slots to transfer one frame
- Size of the slots is defined by minislots as communication time base
- Time reserve to ensure communication (Jitter before and Interframe Space after frame) up to 40%



LIN Error Detection

- Bitmonitoring (sender task)
  - Compare each bit on the bus with bus level
- Checksum (Receiver task)
  - Check of incoming data bytes w/o PID (classic/enhanced checksum)
  - Sum of received data and received checksum has to 0xFF
- Parity check (receiver task)
  - Check of P0 and P1
- Slave responding check (receiver task)
  - Check whether frame response after header is transferred
- Sync field check (receiver task)
  - Check whether master sync rate is in between tolerance range
- Reaction to errors part of individual design, not described in the standard

References and quotations



Buchreihe: Bosch Professional Automotive Information

Herausgeber: Robert Bosch GmbH

Verlag: Springer Fachmedien Wiesbaden

Print ISBN: 978-3-658-01783-5
Electronic ISBN: 978-3-658-01784-2

Enthalten in: Springer Professional "Wirtschaft+Technik" , Springer Professional "Technik"

https://elearning.vector.com
https://www.eecs.umich.edu/courses/eecs461/doc/CAN_notes.pdf
https://www.uni-kassel.de/eecs/fileadmin/datas/fb16/Fachgebiete/FSG/Download/Lehre/PFS/
http://www.lin-subbus.org