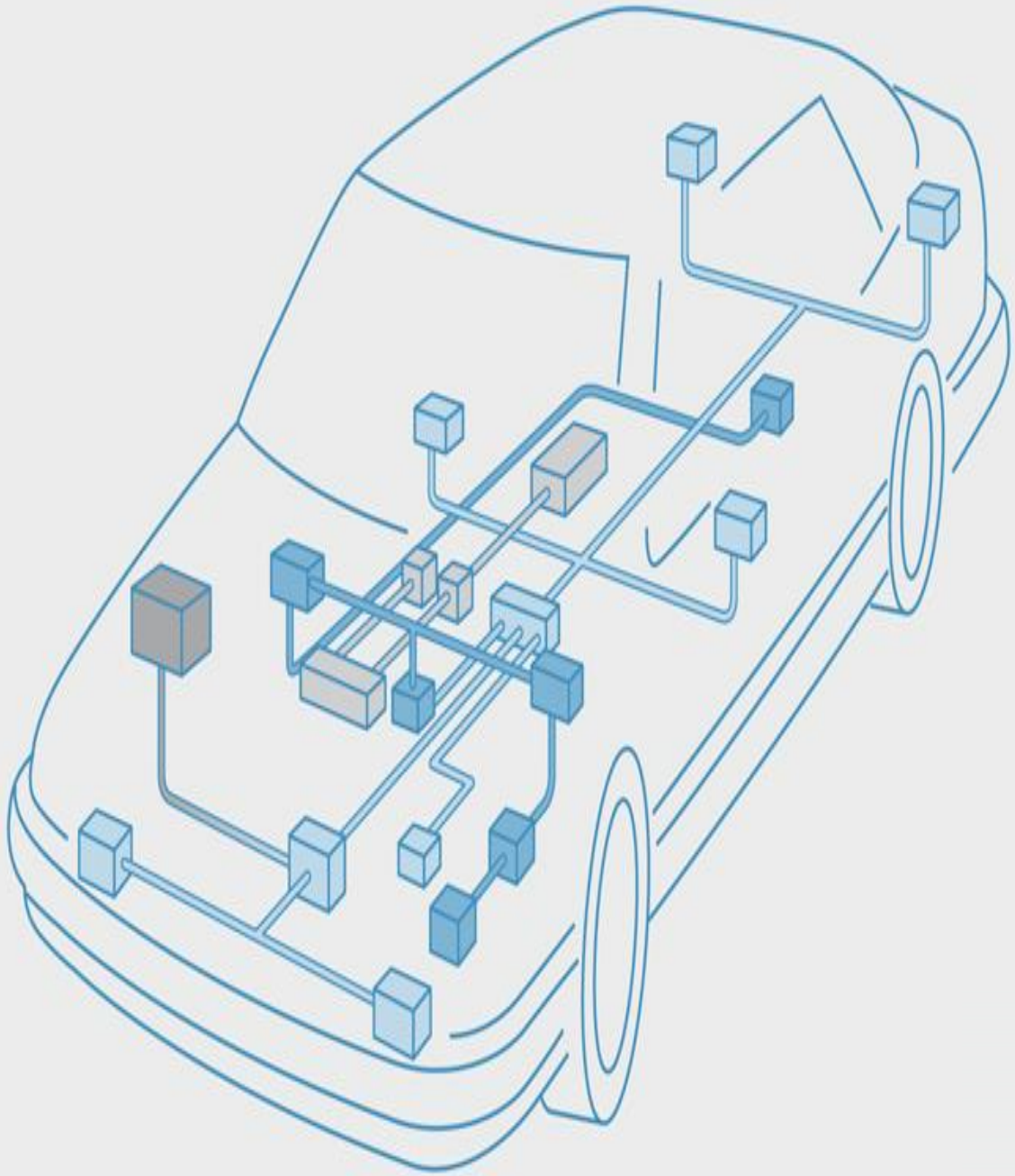
In Vehicle - Communication



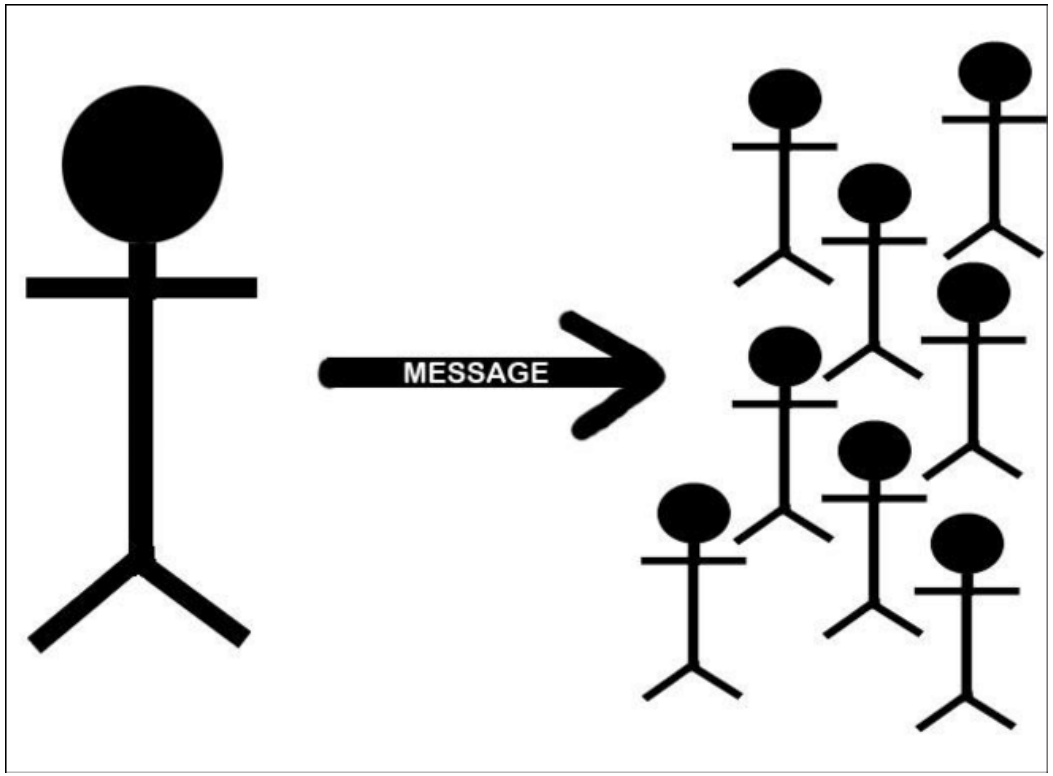
Agenda

- *Communication Basics*
- *In-Vehicle communication*
- *CAN Protocol 2.0B*
- *CAN Physical*
- *CAN inside a Microcontroller*
- *Exercises and Projects*

Introduction

Communication:

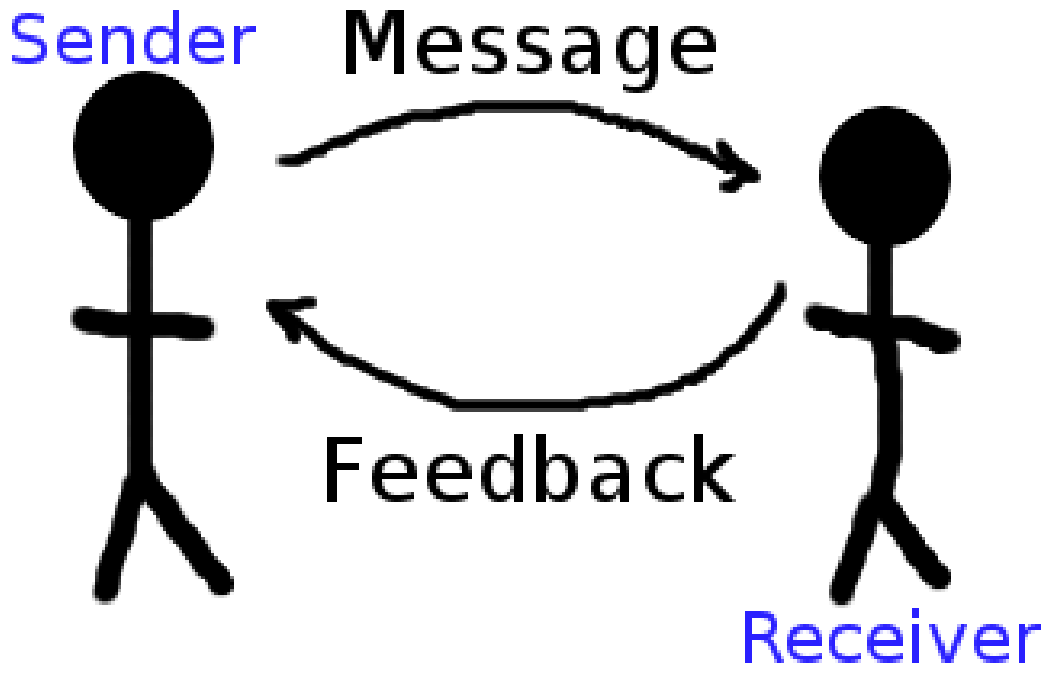
Sharing **information** between two or more bodies



Basic elements

- Transmitter, Speaker, Publisher, Receiver, Audience, Subscribers, etc. (User wants to Say info and User want to listen Info)
- Message/Information(Type of Information)
- Medium/Channel
- **And ???**

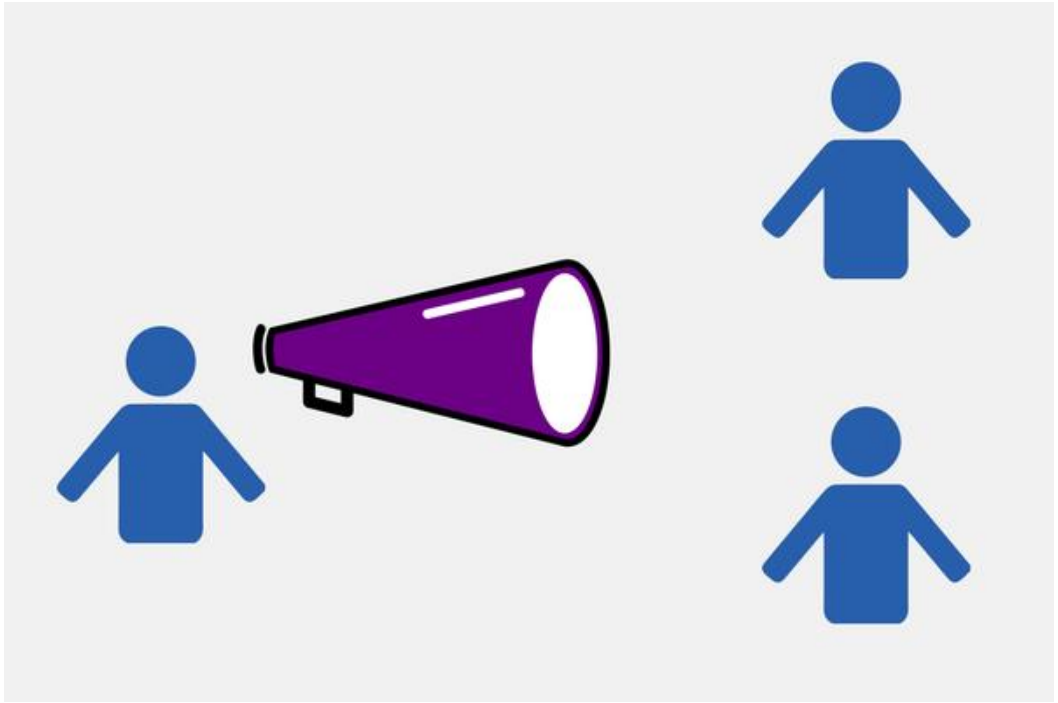
RULES!!!



Cont..

Communication network:

- **Way/method** of information flow within the **group of bodies** participating in exchange of information



Why Rules???

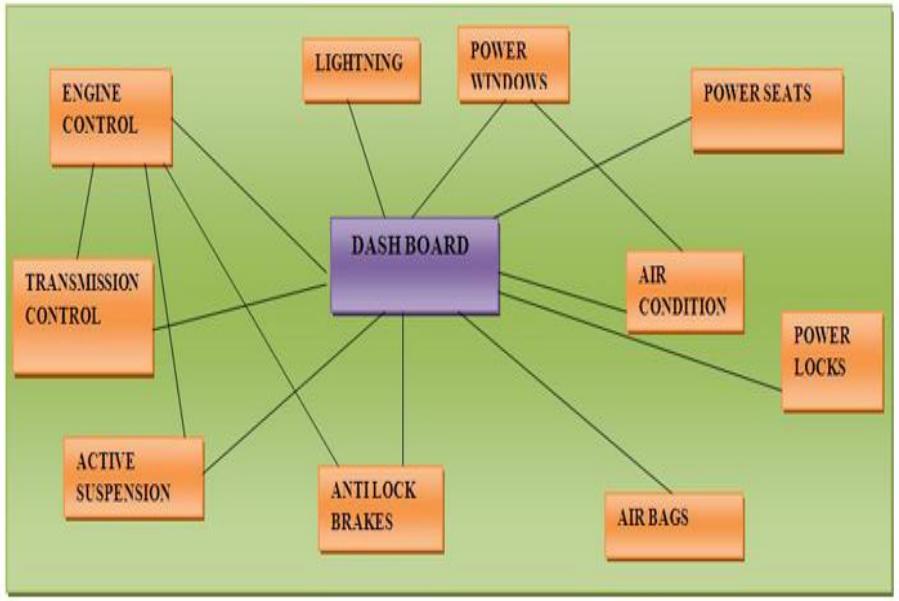
- **Understandable**
- **Clarity**
- **Correctness**
- **Completeness**

…Serves the purpose

Examples…

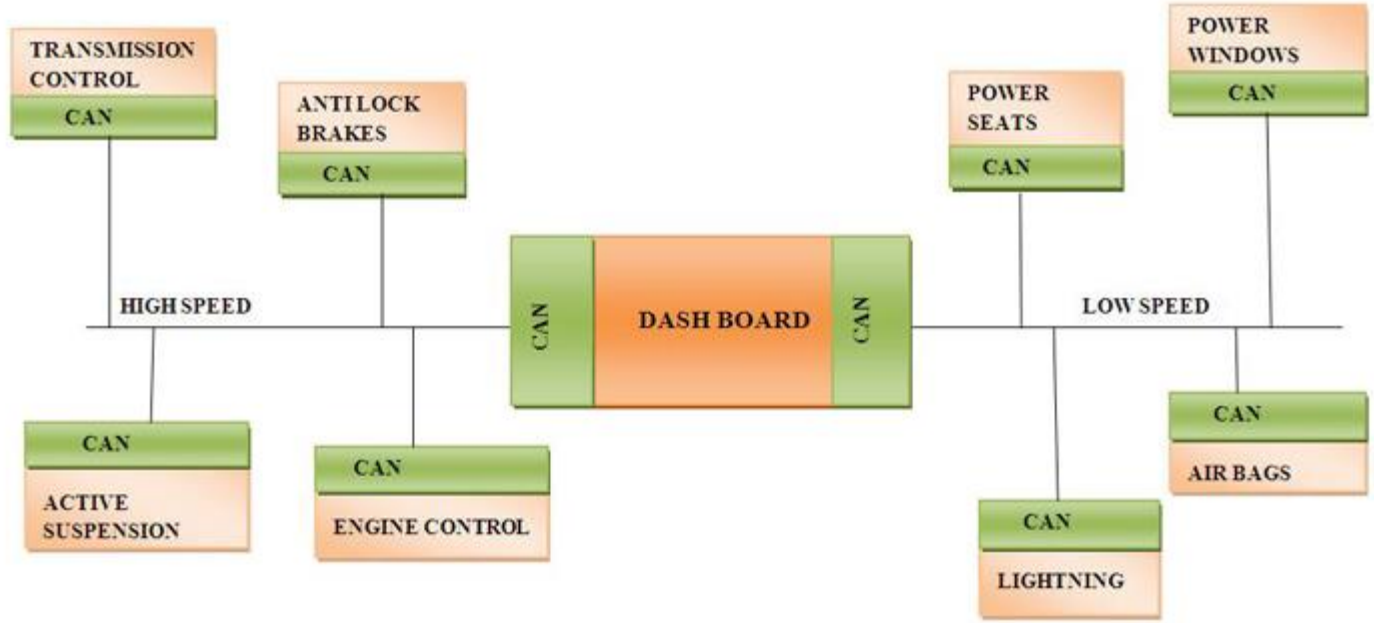
- One to One : Telephone, Mail, Messengers, Fax etc..
- Many to one : Classrooms, E-Mails, etc.
- One to Many : Classrooms, Public Announcements, Siren, etc.
- Many to Many : TV Broadcast

Why Communication inside Vehicle?



- Information exchange is necessary between two ECUs
- Good Communication Qualities
  - ❑ Availability
  - ❑ Reliability
  - ❑ Robustness
  - ❑ Clear
  - ❑ Error free

  ensured via **Standards**



Engine System

ECU

- Control Timing
- Idle Speed
- A/F Mixture

Speed Control Units

- Cruise
- Adaptive Cruise
- Manual

Gear

- Parking
- Reverse
- Neutral
- Drive

Braking System

Breaking Line Value

- Pos1 Open
- Pos2 Close
- Pos3 Partial

Steering System

Lane Assist

- Send warning
- No warning

Lane Keep

- Correct right
- Correct left
- No correct

Lane Centering

- Continuous correction
- Disabled

Infotainment System

Audio

- Radio (AM/FM)
- CD/DVD
- Aux 3.5mm
- USB
- Bluetooth

Voice Commands

- Text
- Phone

Electrical System

Head Lights

- On
- High
- Off

Brake Lights

- On
- Off

Turn Left Front Light

- On
- Off

Turn Left Rear Light

- On
- Off

HVAC System

Temperature

- Low
- Med
- High

Head Direction

- Foot
- Foot & Body
- Body & Face
- Windscreen

Defrost Rear

- On
- Off

Instrument Cluster

Speedometer

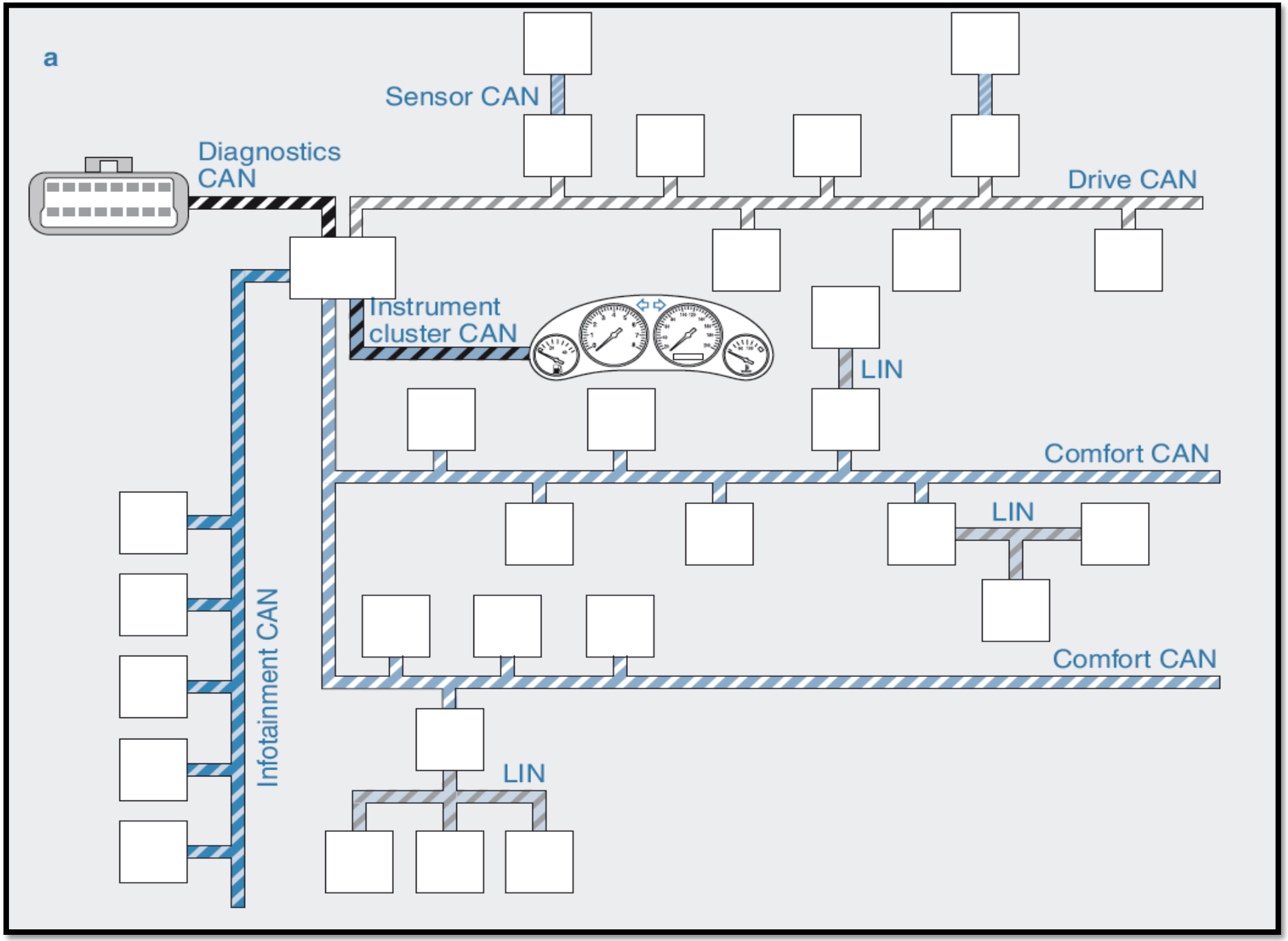
- Zero
- Low
- Med
- High

Low Fuel

- Alert off
- Alert on

Washer Fluid

- Alert off
- Alert on


a
Sensor CAN
Diagnostics CAN
Drive CAN
Instrument cluster CAN
LIN
Comfort CAN
LIN
Infotainment CAN
Comfort CAN
LIN

Good qualities of In-Veh Communication system:

Robustness against environment

- Avoid Information damage

Availability to the user

- Avoid late response
- Avoid overwriting

Correctness and Completeness

- No Old data
- Avoid misinterpretation
- Understandable

Reliability to perform intended operation

- Transfer information from/to right user
- Right Interval

Usage/Programmability

Terminologies…

Addressing

- **Destination dependent**
  - MAC addressing(Ethernet)
  - Slave address (I2C, SPI, USB, UART)
- **Message dependent (CAN)**
  - Destination is unknown. It is Broadcast.
  - CAN – Identifier for the message

Terminologies…

Bus access method

- Master – Slave
- Multi Master
- Time Triggered

Data transfer modes

- Synchronous
- Asynchronous

Terminologies…

Topologies

- Star
- Bus
- Ring, etc..

- **Grouping/Packet/Packing/Parsing**

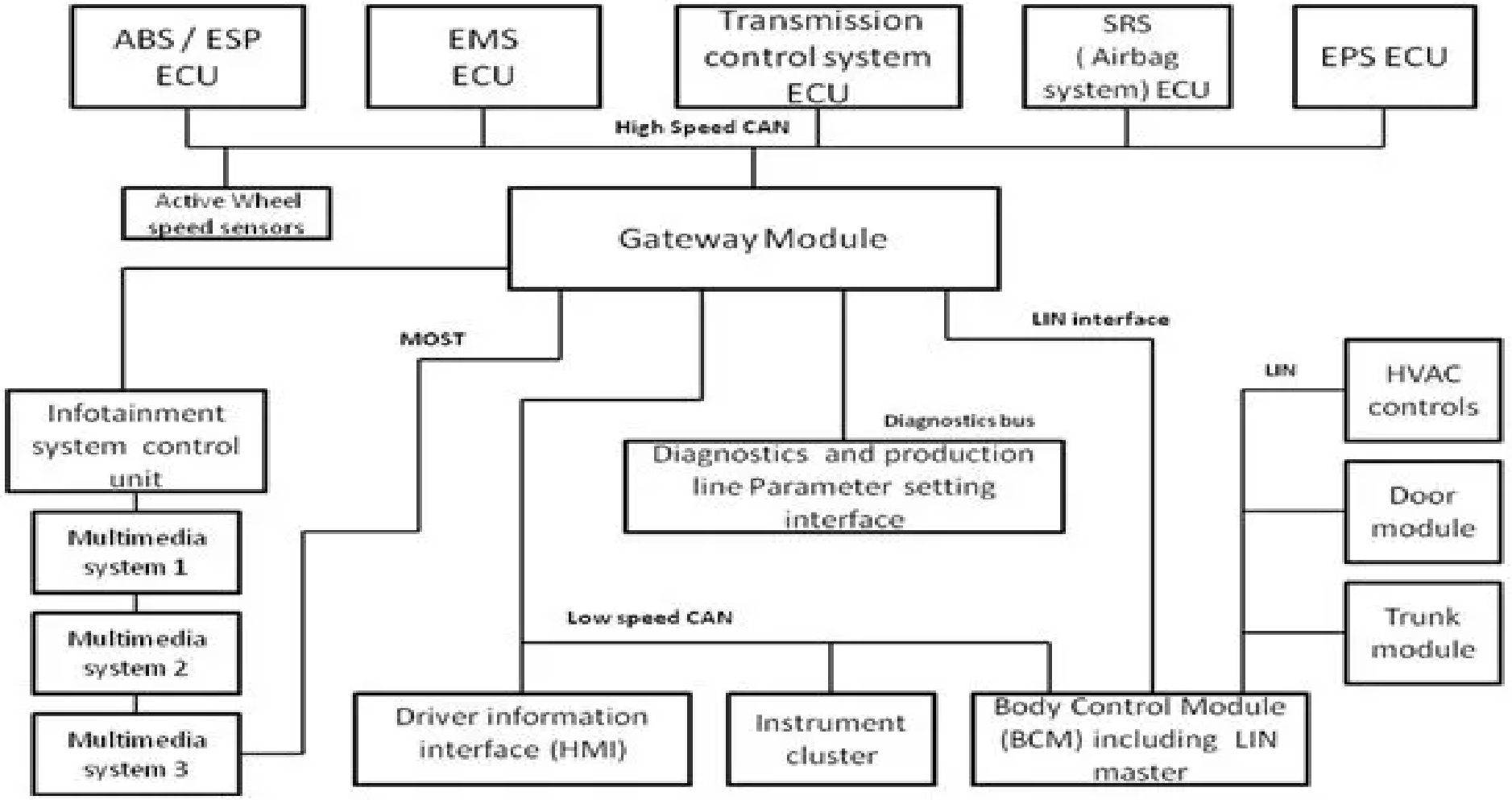
Open Systems Interconnection – Communication model

7	Application Layer	Human-computer interaction layer, where applications can access the network services
6	Presentation Layer	Ensures that data is in a usable format and is where data encryption occurs
5	Session Layer	Maintains connections and is responsible for controlling ports and sessions
4	Transport Layer	Transmits data using transmission protocols including TCP and UDP
3	Network Layer	Decides which physical path the data will take
2	Data Link Layer	Defines the format of data on the network
1	Physical Layer	Transmits raw bit stream over the physical medium

Communication Protocols
(Common for In-Vehicle communication)

- CAN
- Flexray
- LIN
- Ethernet
- MOST

and so on…



CAN Protocol

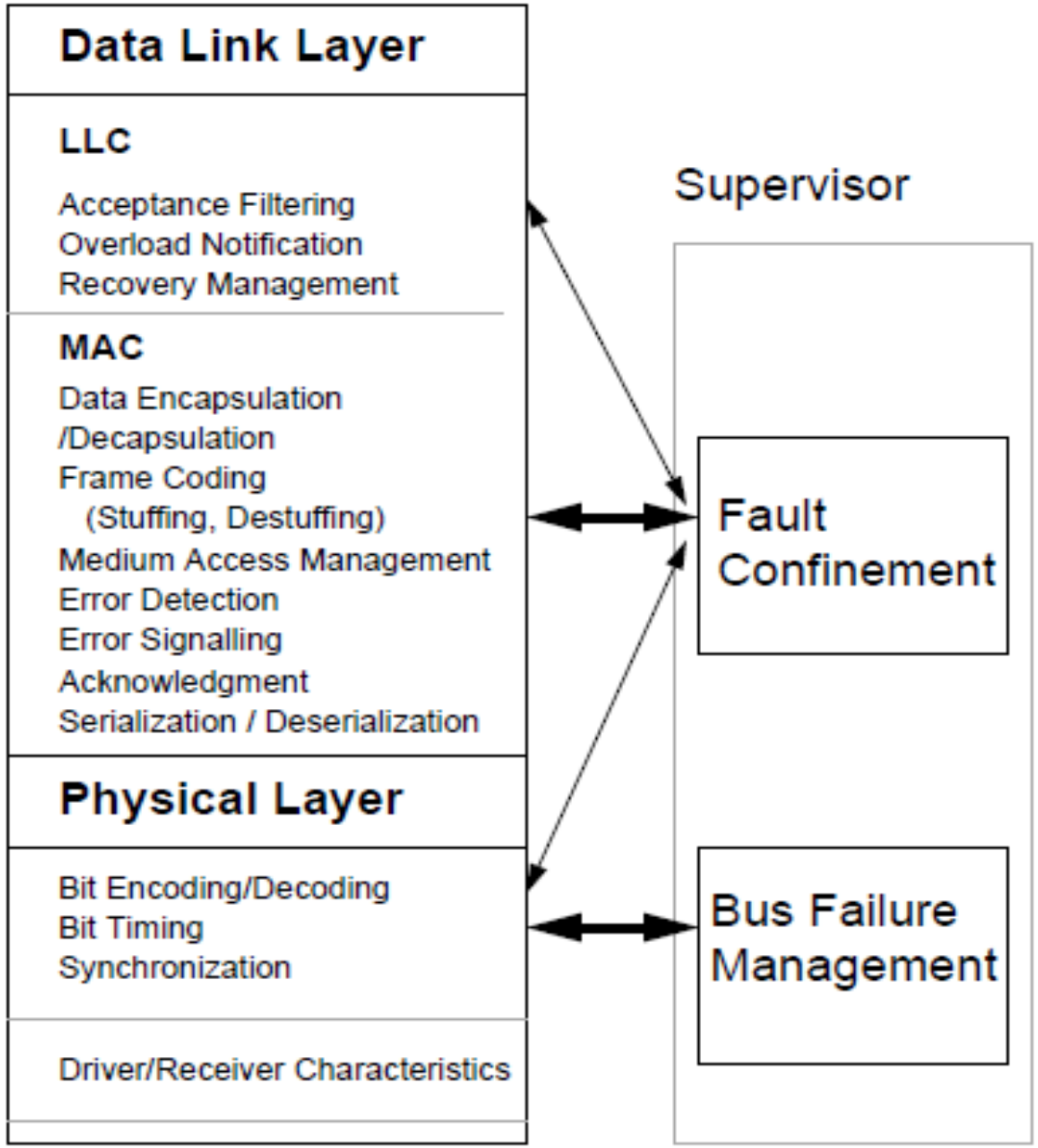
- The CAN bus was developed by BOSCH as a multi-master, message broadcast system that specifies a maximum signaling rate of 1 megabit per second (bps).

- In a CAN network, many short messages like temperature or RPM are broadcast to the entire network, which provides for data consistency in every node of the system.

- CAN is a message based communication protocol.
  - In an address-based protocol, the data packets contain the address of the device for which a message is intended.
  - In a message-based protocol, every message is identified by a pre-defined ID rather than an address.

CAN Protocol

- ISO11898-1 and ISO11898-2 is mainly used for describing Data Link layer and Physical Layer specifications of CAN Communication
- Each layer defines the standards necessary for effective communication for the applied system.
- System incorporating the ISO CAN Communication shall comply with the layer specific requirements



CAN Protocol benefits

Low cost: Since a CAN serial bus uses two wires (with high-volume and low-cost production), it offers a good price-to-performance ratio.

Reliable: CAN offers excellent error-detection and error-handling mechanisms, which provides highly reliable transmission. It's also largely immune to electromagnetic interference.
Flexible: CAN nodes are not limited by the protocol and can be easily connected or disconnected.
Fast: CAN supports **a** data rate of 1 MBit/s @ 40m bus length.
Multi-master communication: Any node can access the bus
Fault confinement: Faulty nodes do not disturb the communication.
Broadcast capabilities: Messages can be sent to one /many/all nodes.
Standardized: ISO has standardized the CAN protocol via ISO-DIS 11898 (for high-speed applications) and ISO-DIS 11519-2 (for low-speed applications). The CAN protocol is also standardized by industry organizations, such as the SAE-Society of Automotive Engineers.

Terminologies

PDU: Protocol Data Unit

SDU: Service Data Unit

PCI : Protocol Control Information

Message:
Set of **Information** to be transferred over the CAN bus.
Along with information, important Protocol supporting elements are also necessary

Terminologies

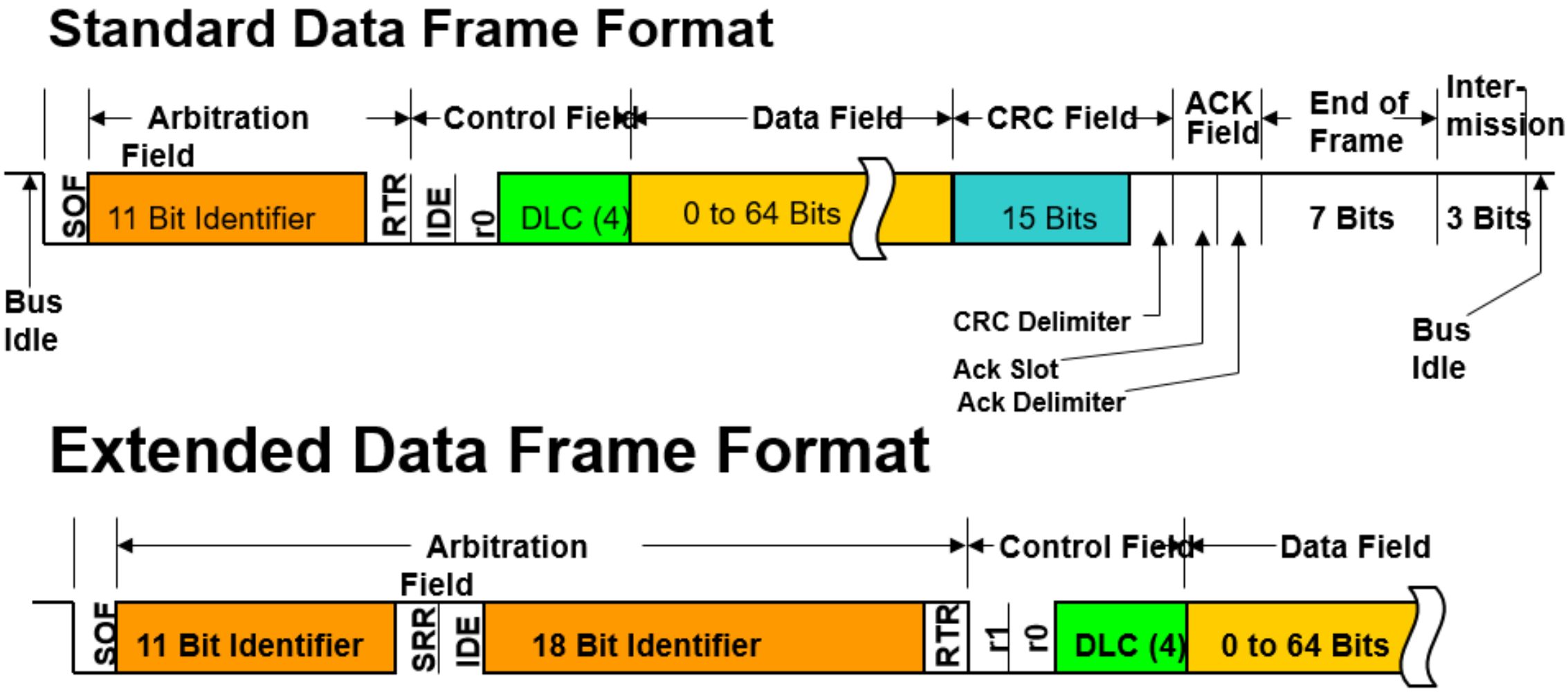
CAN Frame: Protocol specific unit that encapsulates the message to be transferred. Final form of the PDU, before serialization

Arbitration: ???

Acceptance filtering : ???

- Communication mode : ???
- Addressing mode : ???
- Bus Access method : ???
- Bus Value and Bit Timing : ???
- Frame Format and Type : ???
- Error Detection : ???

Frame Format :



CAN Protocol: Standard Frame fields

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 (see Remote Frame, below)
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)

CAN Protocol: Standard Frame fields continue …

- SOF – Start of Frame bit. It indicates start of message and used to synchronize the nodes on a bus. A dominant bit in the field marks the start of frame.

- IDENTIFIER – It serves dual purpose one, to determine which node has access to the bus and second to identify the type of message.

- RTR – Remote Transmission Request. It identifies whether it's a data frame or a remote frame .RTR is dominant when it is a data frame and recessive when it is a remote frame.

- IDE – Identifier Extension. It is used to specify the frame format. Dominant bit is for standard frame and recessive for extended frame.

- R0 – Reversed bit. Not used currently and kept for future use.

- DLC – Data Length Code. It is 4 bit data length code that contains the number of bytes being transmitted.

CAN Protocol: Standard Frame fields continue …

- DATA– Used to store up to 64 data bits of application data to be transmitted.

- CRC– Cyclic Redundancy Check. The 16-bit (15 bits plus delimiter) cyclic redundancy check (CRC) contains the checksum of the preceding application data for error detection.

- ACK – Acknowledge (ACK) field. It compromises of the ACK slot and the ACK delimiter. When the data is received correctly the recessive bit in ACK slot is overwritten as dominant bit by the receiver.

- EOF– End of Frame (EOF). The 7-bit field marks the end of a CAN frame (message) and disables Bit – stuffing, indicating a stuffing error when dominant.

- IFS – Inter Frame Space that specifies minimum number of bits separating consecutive messages. It provides the intermission between two frames and consists of three recessive bits known as intermission bits. This time allows nodes for internal processing before the start of next frame.

CAN Protocol: Extended Frame fields

Field name	Length (bits)	Purpose
Start-of-frame	1	Denotes the start of frame transmission
Identifier A (green)	11	First part of the (unique) identifier which also represents the message priority
Substitute remote request (SRR)	1	Must be recessive (1)
Identifier extension bit (IDE)	1	Must be recessive (1) for extended frame format with 29-bit identifiers
Identifier B (green)	18	Second part of the (unique) identifier which also represents the message priority
Remote transmission request (RTR) (blue)	1	Must be dominant (0) for data frames and recessive (1) for remote request frames (see Remote Frame, below)
Reserved bits (r1, r0)	2	Reserved bits which must be set 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 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)

CAN Protocol: Extended Frame fields

- It is same as 11-bit identifier with some added fields

- SRR- Substitute Reverse Request. The SRR bit is always transmitted as a recessive bit to ensure that, in the case of arbitration between a Standard Data Frame and an Extended Data Frame, the Standard Data Frame will always have priority if both messages have the same base (11 bit) identifier.

- R1- It is another bit not used currently and kept for future use.

Data Length Code

Number of Data Bytes	Data Length Code			
	DLC3	DLC2	DLC1	DLC0
0	d	d	d	d
1	d	d	d	r
2	d	d	r	d
3	d	d	r	r
4	d	r	d	d
5	d	r	d	r
6	d	r	r	d
7	d	r	r	r
8	r	d	d	d

Frame Types

- Based on purpose of frame:
  - Data frame: a frame containing node data for transmission
  - Remote frame: a frame requesting the transmission of a specific identifier
  - Error frame: a frame transmitted by any node detecting an error
  - Overload frame: a frame to inject a delay between data or remote frame

- Based on Arbitration field : Standard or Extended
  - Standard
  - Extended

- Based on Date rate:
  - Classical (0 to 8 bytes of Data)
  - CAN Flexible Data rate (0 to 64 bytes of Data)

CAN Protocol: DATA Frame

The data frame is the only frame for actual data transmission. There are two message formats:

- Standard frame format: with 11 identifier bits
- Extended frame format: with 29 identifier bits

The CAN standard requires that the implementation must accept the base frame format and may accept the extended frame format, but must tolerate the extended frame format.

CAN Protocol: REMOTE Frame

Generally data transmission is performed on an autonomous basis with the data source node (e.g., a sensor) sending out a data frame. It is also possible, however, for a destination node to request the data from the source by sending a remote frame.

There are two differences between a data frame and a remote frame. Firstly the RTR-bit is transmitted as a dominant bit in the data frame and secondly in the remote frame there is no data field. The DLC field indicates the data length of the requested message (not the transmitted one).

RTR = 0 ; DOMINANT in data frame
RTR = 1 ; RECESSIVE in remote frame

In the event of a data frame and a remote frame with the same identifier being transmitted at the same time, the data frame wins arbitration due to the dominant RTR bit following the identifier.

CAN Protocol: ERROR Frame

The error frame consists of two different fields:
The first field is given by the superposition of ERROR FLAGS (6–12 dominant/recessive bits) contributed from different stations.
The following second field is the ERROR DELIMITER (8 recessive bits).
There are two types of error flags:
Active Error Flag : six dominant bits – Transmitted by a node detecting an error on the network that is in error state "error active".
Passive Error Flag : six recessive bits – Transmitted by a node detecting an active error frame on the network that is in error state "error passive".
There are two error counters in CAN:
Transmit error counter (TEC) and Receive error counter (REC)
When TEC or REC is greater than 127 and less than 255, a Passive Error frame will be transmitted on the bus.
When TEC and REC is less than 128, an Active Error frame will be transmitted on the bus.
When TEC is greater than 255, then the node enters into Bus Off state, where no frames will be transmitted.

CAN Protocol: OVERLOAD Frame

The overload frame contains the two bit fields Overload Flag and Overload Delimiter. There are two kinds of overload conditions that can lead to the transmission of an overload flag:

The internal conditions of a receiver, which requires a delay of the next data frame or remote frame.
Detection of a dominant bit during intermission.

The start of an overload frame due to case 1 is only allowed to be started at the first bit time of an expected intermission, whereas overload frames due to case 2 start one bit after detecting the dominant bit. Overload Flag consists of six dominant bits. The overall form corresponds to that of the active error flag.
The overload flag's form destroys the fixed form of the intermission field. As a consequence, all other stations also detect an overload condition and on their part start transmission of an overload flag. Overload Delimiter consists of eight recessive bits. The overload delimiter is of the same form as the error delimiter.

Important Points to be known in CAN Arch

- Acceptance filtering
- Arbitration
- Encapsulation / Decapsulation
- Bit stuffing
- Serialization/ De-Serialization
- Ack
- Transfer rate (Baud, Bit timing, FD)
- Error types

Acceptance filtering:

Basic CAN: Basic CAN has only one Message buffer for Receive and Transmit messages. The received message is accepted or ignored after acceptance filtering. The decision to process a message or to ignore it is also achieved by acceptance filtering. This acceptance filtering of the node is done by software in Basic CAN. To reduce the software load at the nodes, there is a possibility to ignore some messages by ignoring specific identifiers. This is realized by bit mask for the message identifiers.

Full CAN: In Full CAN, there are 8 to 16 memory buffers for every transmitted or received message. Here the acceptance filtering is done by hardware and not by the software. Every buffer can be configured to accept messages with specific ID's.

Since the acceptance filtering is done by hardware, the software load is greatly reduced. With different buffers for different messages ensures more time for the processing of the received messages and the transmitted message can be handled according to the priority levels. Configuring each buffer for every message ensures also the data consistency in Full CAN

Arbitration:

- It is a mechanism which resolves the conflict when two or more nodes try to send the message at the same time.
- In this technique whenever the bus is free any unit can transmit a message. If two or more units starts transmitting at the same time access to the bus is conflicted, but this problem can be solved by arbitration using identifier.
- During arbitration every transmitter compares the value of transmitted bit with bit value on the bus. If the bit value is same, the node continues to send the bits. But at any time if transmitted bit value is different from bus value the dominant bit overwrites the recessive bits.
- The arbitration field of the CAN message consists of an 11- or 29-bit identifier and a remote transmission (RTR) bit. The identifier having lowest numerical value has the highest priority.
- RTR simply distinguishes between remote frame for which RTR is recessive and data frame for which RTR is dominant. If both data frame and remote frame with the same identifier is initiated at the same time data frame will prevail over remote frame.
- With the concept of arbitration neither information nor time is lost.

Error Types:

- CRC Error
- Form error
- Ack Error
- Bit Error
- Stuff Error

Error types continue …

CRC error: In this stage a 15-bit cyclic redundancy check value is calculated by transmitting node and is transmitted in the CRC field. This value is received by all nodes. Then all the nodes calculate CRC value and matches the results with the transmitted value. If values differ than an Error Frame is generated. Since one of the nodes did not receive the message properly it is resent.

ACK error: When transmitting node sends a message, a recessive bit is sent in acknowledgement slot. After message is received acknowledge slot is replaced by dominant bit which would acknowledge that at least one node correctly received the message. If this bit is recessive, then none of the node has received the message properly.

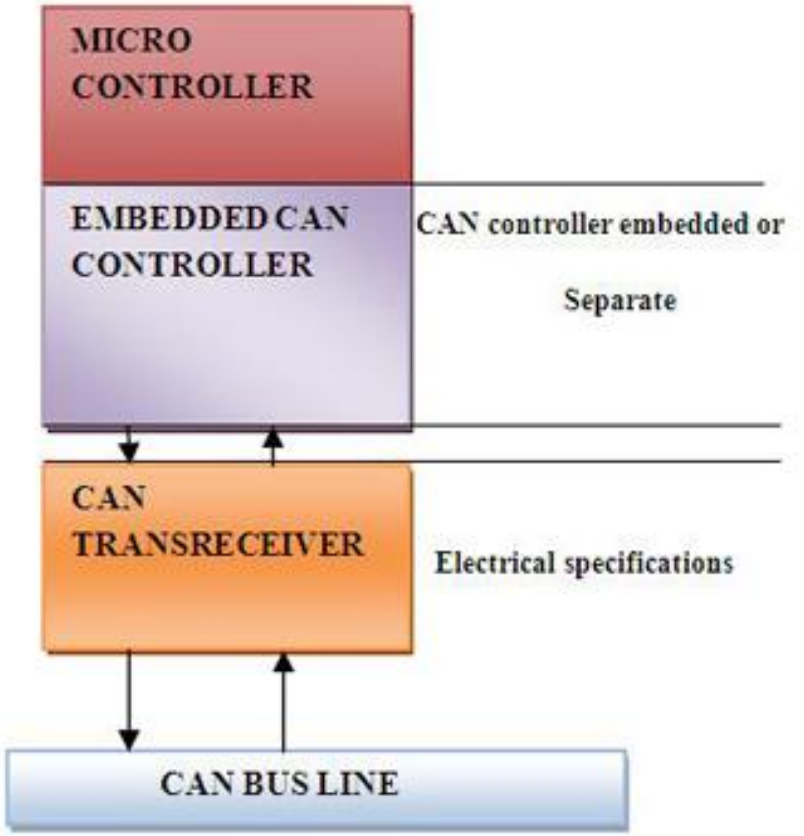
Form error: End of frame, Inter-frame space, Acknowledge Delimiter are fields that are always recessive, if any node detects dominant bit in one of these fields than CAN protocol calls it a violation and a Form Frame is generated and original message is resent after certain period.

Error types continue …

Bit error: A node that is sending the bit always monitors the bus. If the bit sent by transmitter differs from the bit value on the bus then error frame is generated. But there is an exception in case of arbitration field or Acknowledge slot where a recessive bit is sent and a dominant bit is received. Then no Bit Error is occurs when dominant bit is monitored.

Stuff error: Bit stuffing – It is a very common technique used in telecommunication and data transmission to insert non -informative bits to have same bit rates or to fill the frames .These extra bits are removed by data link layer to retrieve the original message. This same technique is used in bit error. CAN bus is never idle because it uses NRZ method. After five consecutive bits of the same value, a bit with a complement or opposite value is stuffed into the bit stream. If six bits of the same value are detected between SOF and CRC delimiter, error frame is generated. Upon detection of errors, the transmission is aborted and frame is repeated. If errors continue, then the station or node may switch itself off to prevent the bus from being tied up

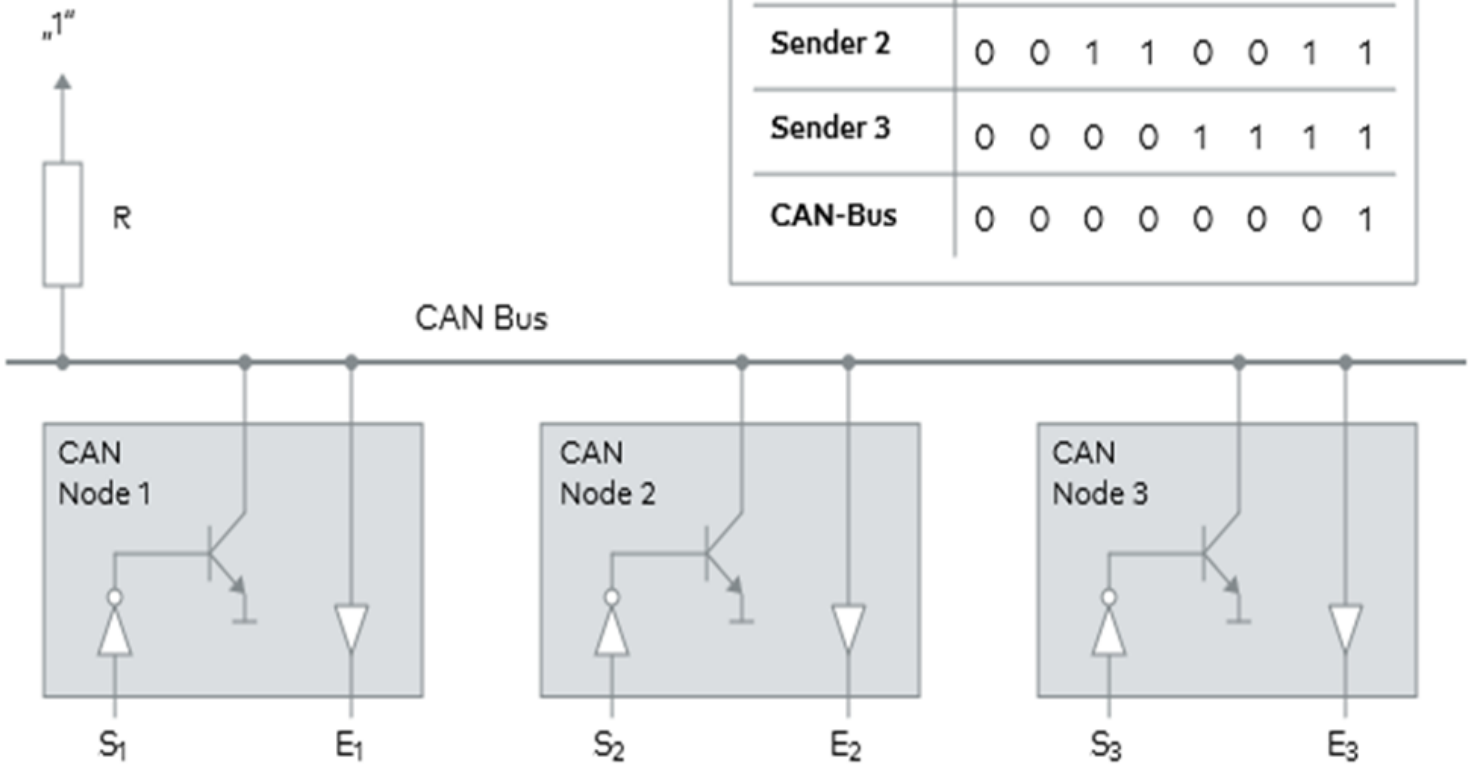
CAN HW implementation



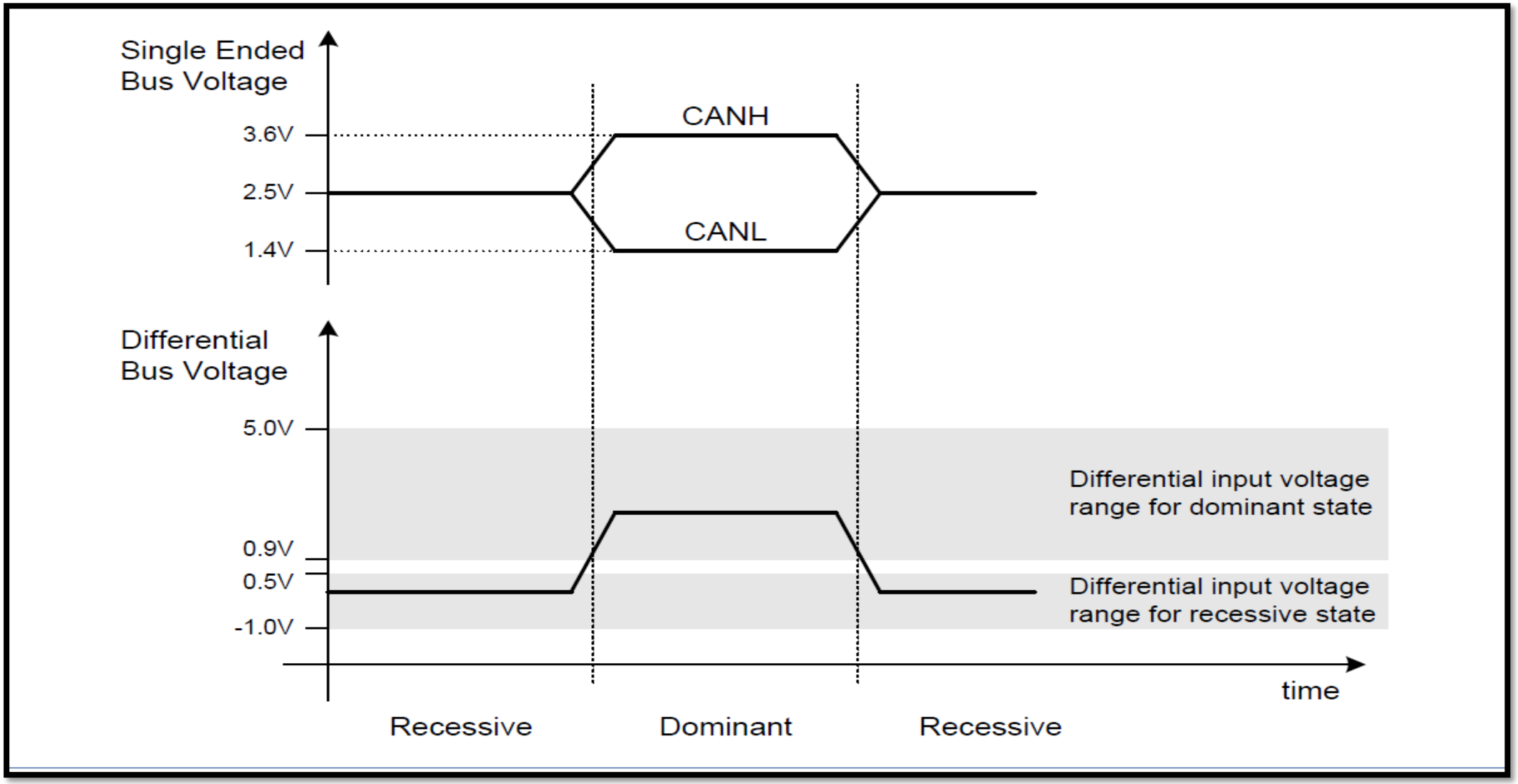
CAN Bus Logic

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

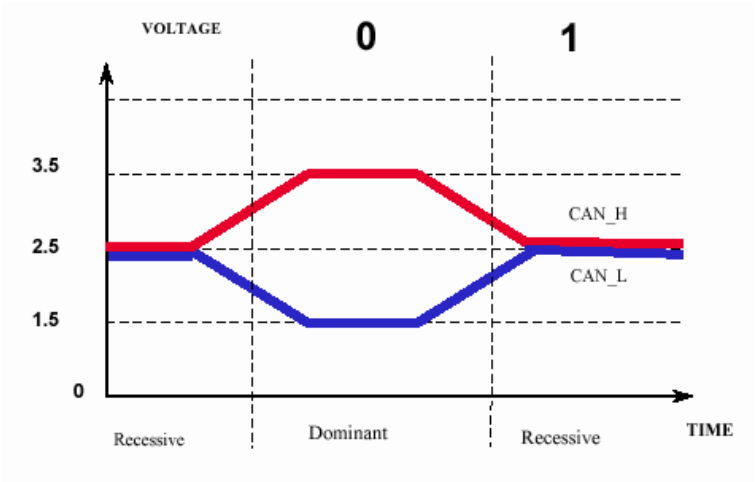


CAN Physical : Voltage levels

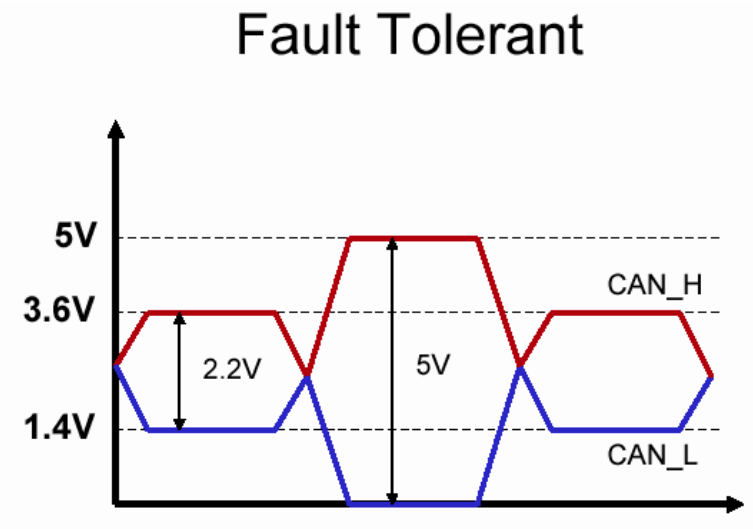


CAN Physical : Voltage levels

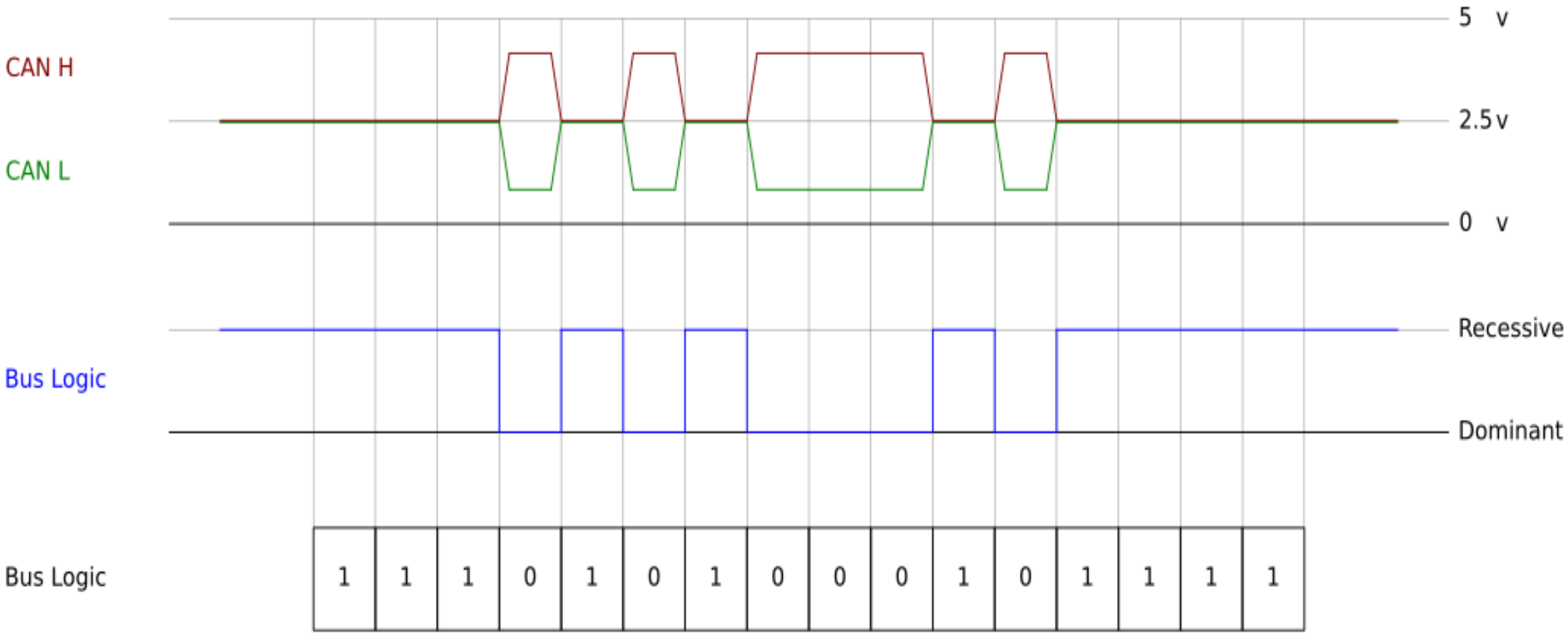
Normal

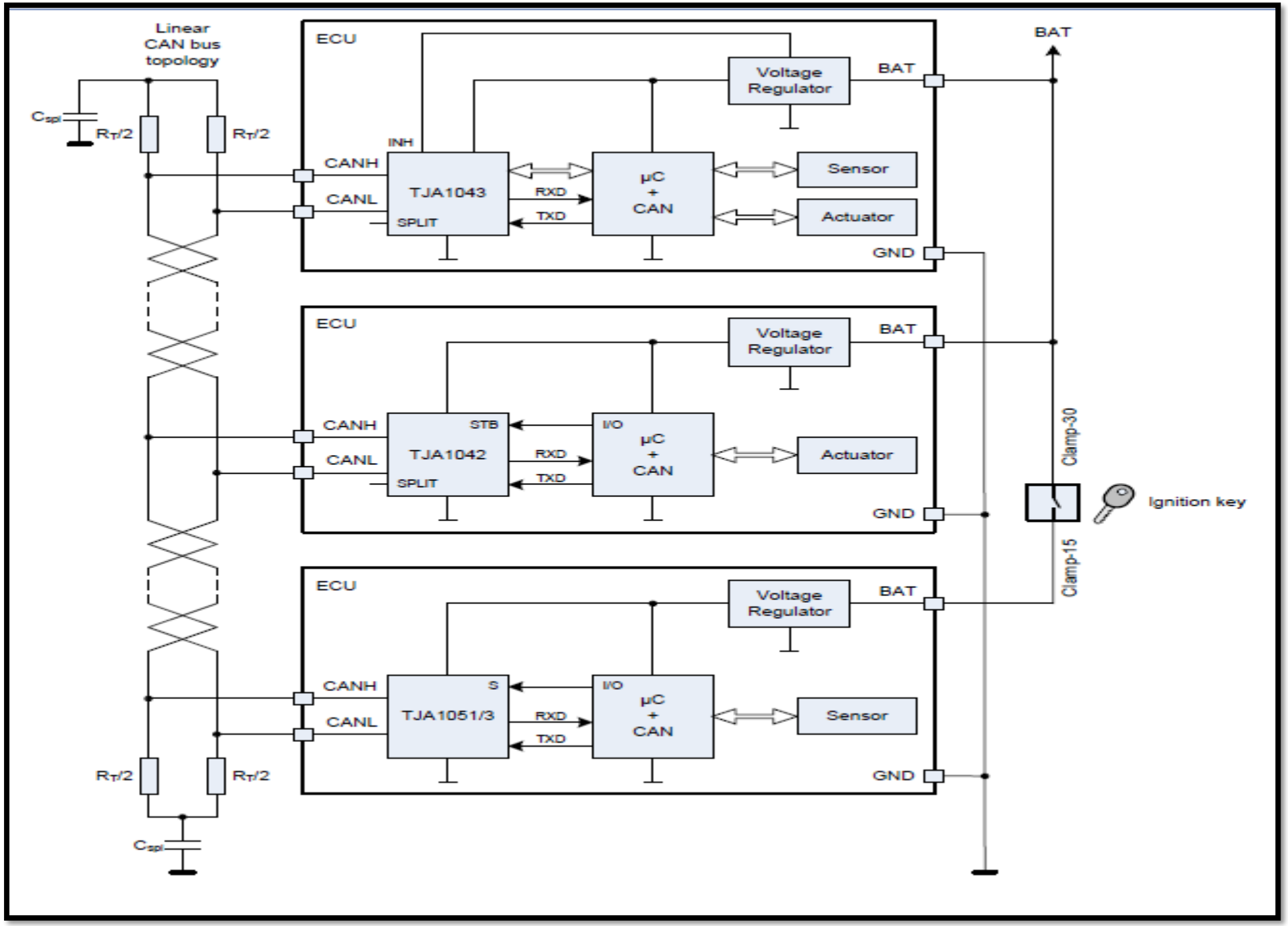


Fault Tolerant

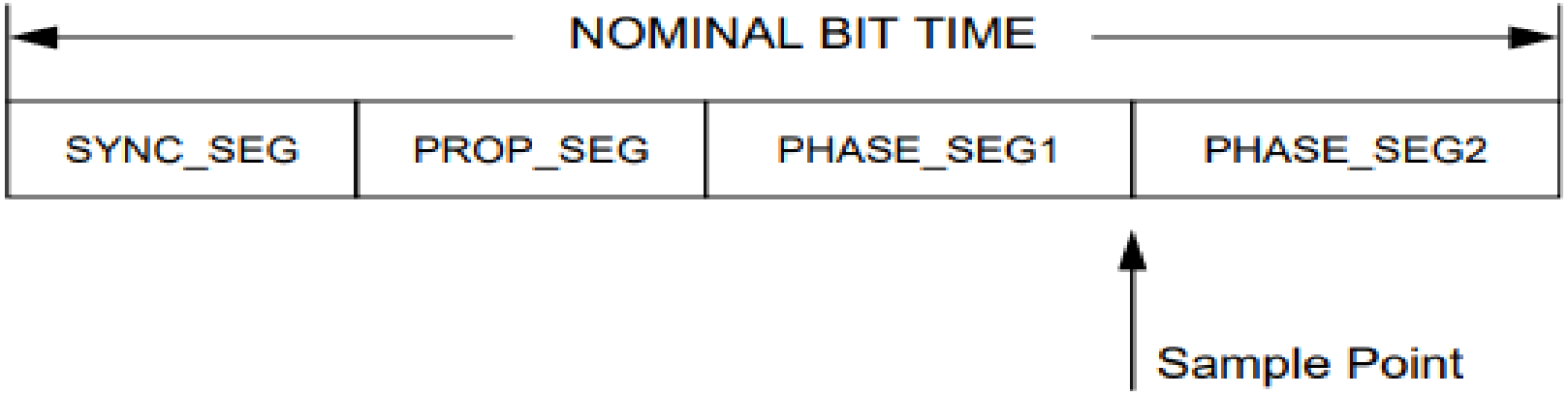


CAN Physical : Voltage levels

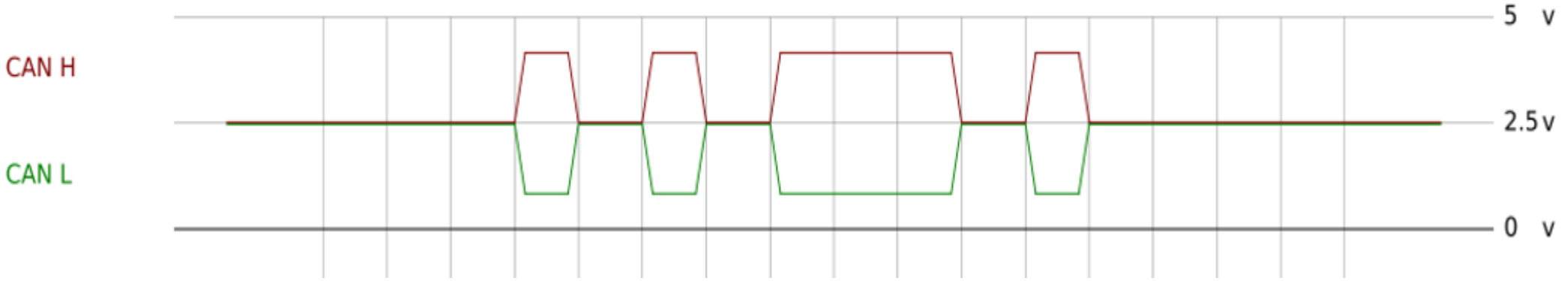



Linear
CAN bus
topology
C_{spl}
$R_T/2$
$R_T/2$
ECU
Voltage
Regulator
BAT
INH
CANH
CANL
TJA1043
SPLIT
RXD
TXD
µC
+
CAN
Sensor
Actuator
GND
ECU
Voltage
Regulator
BAT
CANH
CANL
TJA1042
STB
SPLIT
RXD
TXD
I/O
µC
+
CAN
Actuator
GND
ECU
Voltage
Regulator
BAT
CANH
CANL
TJA1051/3
S
RXD
TXD
I/O
µC
+
CAN
Sensor
GND
BAT
Clamp-30
Clamp-15
Ignition key
$R_T/2$
$R_T/2$
C_{spl}

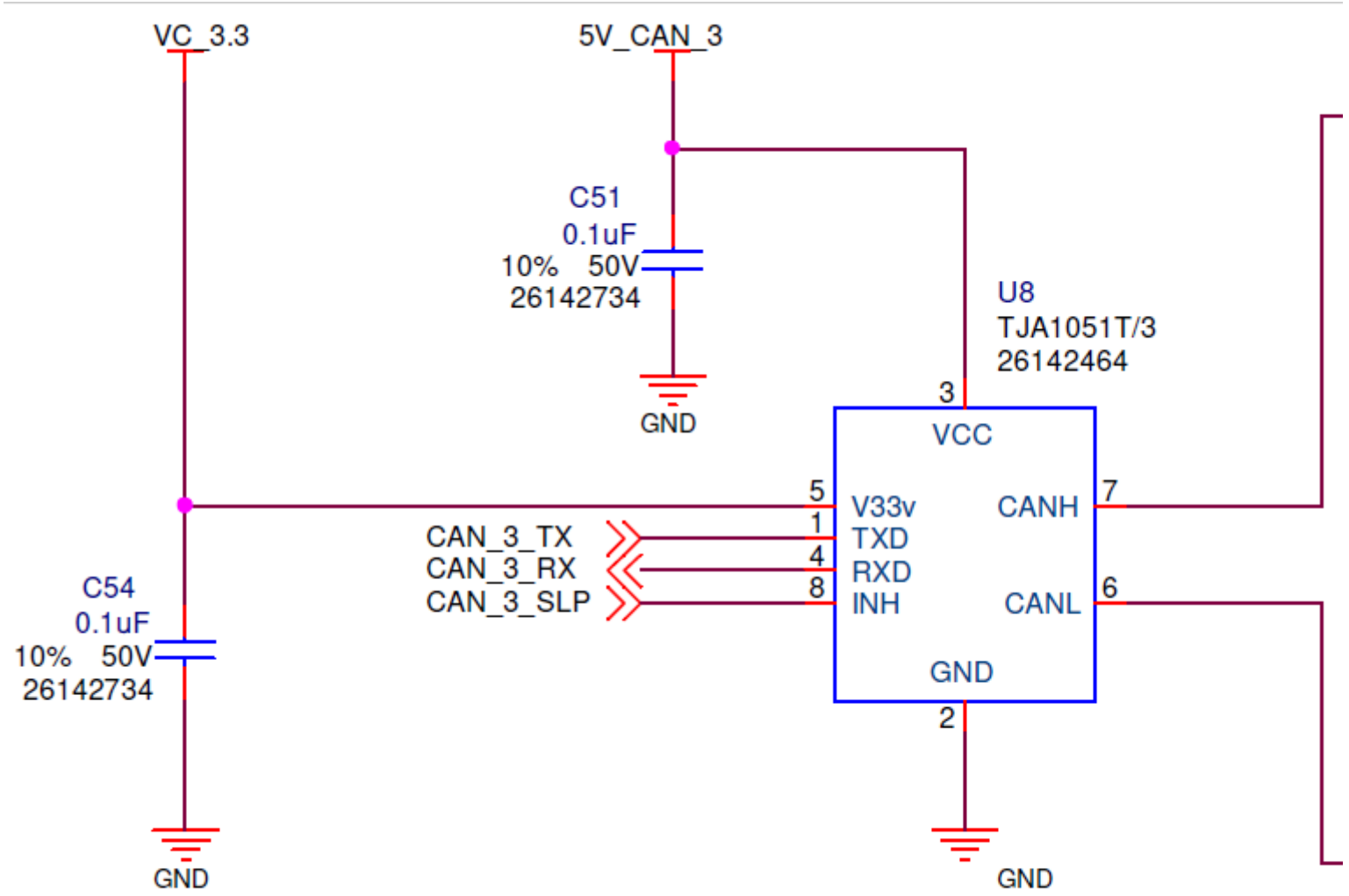
Bit Timing



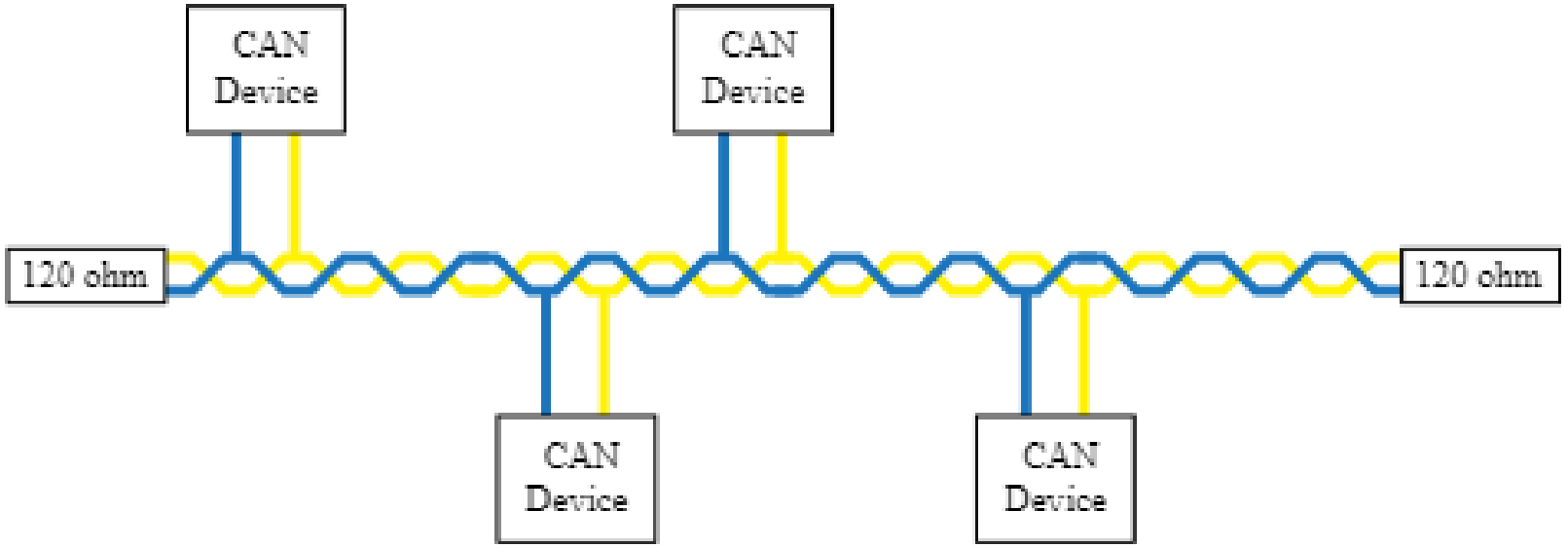
- SYNCHRONIZATION SEGMENT (SYNC_SEG)
- PROPAGATION TIME SEGMENT (PROP_SEG)
- PHASE BUFFER SEGMENT1 (PHASE_SEG1)
- PHASE BUFFER SEGMENT2 (PHASE_SEG2)



CAN Physical : Transceiver



A properly configured bus looks like this diagram:



References:

- ISO11898
- ISO7498 -1