BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

Automotive Ethernet vs. Ethernet

- The fundamental difference between automotive Ethernet vs. Ethernet in its traditional form involves the cables:
- standard Ethernet uses two twisted pairs, one for transmitting data and one for receiving.
- Automotive Ethernet, on the other hand, uses only one twisted pair that transmits and receives at the same time.
- The automotive ethernet cables are also much shorter in length to account for the unforgiving environmental conditions in the vehicle.
- Automotive Ethernet also uses a different type of signal encoding to transmit the maximum amount of data using the least amount of bandwidth.

Implementation of Automotive Ethernet

- Despite the introduction of a new standard of Ethernet specifically for the automotive industry and its clear advantages, the industry is already heavily invested in legacy networks such as CAN and is unlikely to make a complete change quickly.
- This means that there is going to be an intermediate period in which automobiles will be equipped with a mix of Ethernet and non-Ethernet ECUs.
- During this transition, both technologies will be used in tandem and it remains necessary for both protocols to be supported.

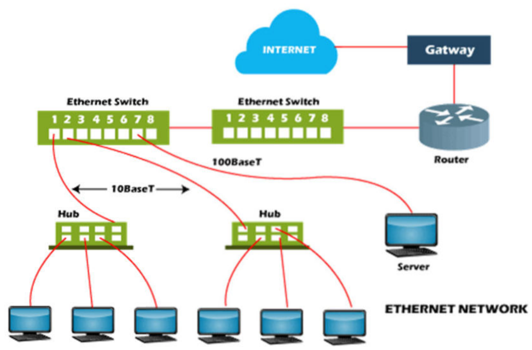
Benefits of Automotive Ethernet

- While the automotive industry will likely always use a combination of CAN and Ethernet, the eventual transition to an automotive ethernet backbone is inevitable and for good reason.
- The new generation of vehicle software supporting the driver-centric experience requires support for increased communication needs.
- Switching to Ethernet also comes with some downsides including a more costly controller and physical-layer interface, complicated EMC issues, and overheads relating to real time communication such as TSN.
- The advantages, however, significantly outweigh the disadvantages, making this transition a crucial next step in the continued growth of the automotive industry.

Benefits of Automotive Ethernet

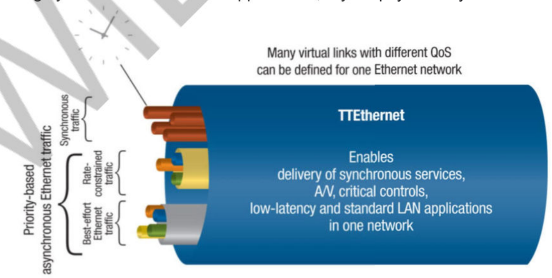
- The transition to automotive Ethernet will provide the following benefits:
- Significantly higher throughput rates (up to 10 Gbps) will be supported with multiple CAN buses aggregated to a single Ethernet link.
- This means less wiring and lower installation and maintenance costs.
- Quality of service (QoS) and time-sensitive networking (TSN) will be supported, allowing for real-time communication and the transmission of lower priority data at the same time.
- Advanced security features will protect highly-connected vehicles from hackers and viruses.
- As new components are upgraded or added, they can easily be connected (and old ones disconnected) using Ethernet's plug and play capabilities.

innovate achieve lead



TIME TRIGGERED ETHERNET- TTEthernet

Time-Triggered Ethernet is a scalable networking technology that uses time scheduling to deliver deterministic real-time communication over Ethernet. It has been specifically designed for safe and highly available real-time applications, cyber-physical systems and unified networking.



From Ethernet to TTEthernet

- Competition in this area has grown with the design of additional solutions such as Safety Ethernet that are intended to meet the requirements of engine and plant construction.
- The use of Real-Time and Safe Ethernet systems outside engine and plant construction is no option.
- There also has been made an effort to adapt Ethernet for special requirements in other areas, e.g. LXI in measurement technology or AFDX in the aerospace industry.

TTEthernet combines the proven determinism, fault-tolerance and real-time properties of the time-triggered technology with the flexibility, dynamics and legacy of “best effort” of Ethernet and is therefore suited for all types of applications.

TTEthernet brings together the high flexibility of free-form systems and the reliability and speed of statically configured systems

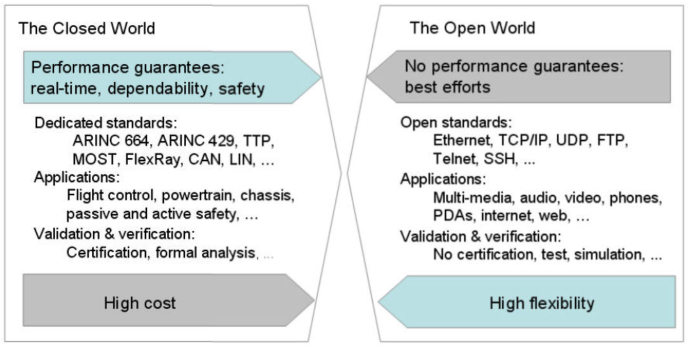
TTEthernet

The Time-Triggered Ethernet (SAE AS6802) (also known as TTEthernet or TTE)

- Ethernet is most popular LAN technology in the world
- Technology is a well-established open-world standard and very scalable
- Early version was bus-based and 10 Mbit/s, today 100 Mbit/s and 1 Gbit/s (Gigabit Ethernet) are common
- Ethernet is specified in OSI layer one and two
- Found by Xerox Palo Alto Research Center (PARC) in 1975
- Original designed as a 2.94 Mbps system to connect 100 computers on a 1 km cable
- Later, Xerox, Intel and DEC drew up a standard support 10 Mbps
- Basis for the IEEE’s 802.3 specification
- Ethernet uses the CSMA/CD media access control

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

TTEthernet combines closed-world and open-world systems on the basis of IEEE 802.3 Ethernet standards



TTEthernet Design Objectives

- TTEthernet enables the seamless communication of all applications by way of Ethernet.
- The IEEE Ethernet 802.3 standards is suited for data transmission among different applications with various requirements.
- Fault tolerance mechanisms avoid the fault propagation in the system and prevent potential hackers from unauthorized access to resources.
- Able to transmit real-time data in distributed controls and shall be suited for safety-critical applications in the future. Existing applications need not be changed.
- Time-critical messages always take precedence over less important messages in TTEthernet.
- The temporal behavior of the time-critical messages is predictable (deterministic) and can be characterized depending on the required quality.
- This system remains fully functional even if a failure occurs (supporting a single or double fault hypothesis).
- This fact accounts for the essential difference between TTEthernet and other Safe Ethernet systems.
- The behavior of TTEthernet is precisely predictable and thus formally verifiable.

TTEthernet System Properties

TTEthernet has time-triggered services that enable time triggered communication over Ethernet. These timetriggered services establish and maintain a global time, which is realized by the close synchronization of local clocks of the devices. The global time forms the basis for system properties such as,

- Temporal partitioning
- Precise diagnosis,
- Efficient resource utilization, or composability.

Temporal partitioning

- Global time operates as a "temporal firewall".
- it is not possible for a faulty application to untimely access the network.
- the switch will block faulty transmission attempts
- Failures of the switch can be masked by powerful end-to-end arguments such as CRCs or by high-integrity designs.

Efficient Resource Utilization

- Free of conflicts.
- Multiplex media access logic
- Buffer memory in the nodes, which can be minimized.
- Energy can be seen, and saved, analogously to memory.

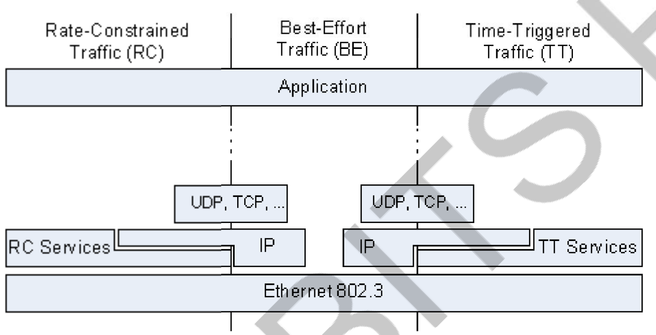
Precise Diagnosis

- simplifies the process of reconstruction of a chain of distributed events.

Composability

- The access pattern to the communication network can be defined.

Dataflow Options in TTEthernet - TTEthernet to existing communication standards



Traffic types

Three different traffic types are,

- Time-triggered (TT) traffic,
- Rate-constrained (RC) traffic,
- Best-effort (BE) traffic.
  - Type of a message can be identified based on a message's Ethernet Destination address.
  - TTEthernet is only concerned with "when" a data message is sent, not with specific contents within in a message.

TT, BE, RC

TT

- Used for time-triggered applications.
- messages are sent over the network at predefined times and take precedence over all other traffic types.
- Optimally suited for communication in distributed real-time systems.
- Typically used for brake-by-wire and steerby- wire systems that close rapid control loops over the network.
- Allows designing and testing strictly deterministic distributed systems.
- The behavior of all system components can be specified, analyzed and tested with sub-micro second precision.

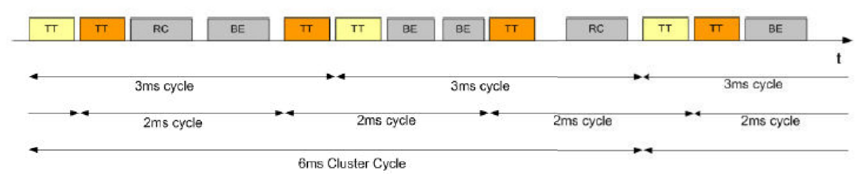
RC

- Applications with less stringent determinism and real-time requirements than strictly time-triggered applications.
- Guarantee that bandwidth is predefined for each application, and delays and temporal deviations have defined limits.
- Safety critical automotive and aerospace applications that depend on highly reliable communication and have moderate temporal quality requirements.
- multimedia systems.
- Different communication controllers may send RC messages at the same point in time to the same receiver.
- Queue up in the network switches, leading to increased transmission jitter.
- The transmission jitter can be calculated off-line and message loss is prevented.

BE

- There is no guarantee whether and when these messages can be transmitted,
- What delays occur and if BE messages arrive at the recipient.
- The remaining bandwidth of the network and have less priority than TT and RC messages.
- Typical user of BE messages are web services.
- All legacy Ethernet traffic without any QoS requirement can be mapped to this service class.
- TTEthernet implements strong partitioning between non-critical BE traffic and all other service classes.

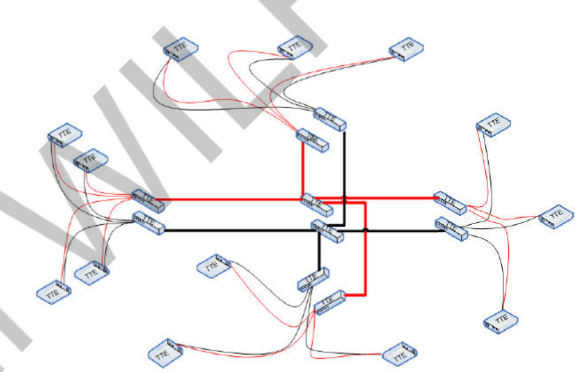
TTEthernet includes TT, RC and BE messages



TTEthernet as Transparent Synchronization Protocol

- Able to co-exist with other traffic, potentially legacy traffic, on the same physical communication network.
- The devices generating the synchronization messages may be distributed with a high number of intermediate devices in between each other.
- The transparent integration of the time-triggered services on top of message-based communication infrastructures such as standard Ethernet.
- Transparent clock mechanism that enables the concept of the permanence point in time, which allows re-establishing the send order of messages in a receiver:
- Distributed computer network that impose a dynamic delay on the transmission, reception into a dedicated field in the synchronization messages used for the synchronization protocol.
- Novel precise calculation of the permanence point in time

Safety and Fault Tolerance



Safety and Fault Tolerance

- A high level of safety is provided by the time-triggered method.
- Detects failures and irregularities in the network and certain systems.
- Can be set up with multiple redundant end systems, switches and segments.
- Thus the system will remain in operation even if faults occur. Redundant network paths are always used in faulttolerant
- The failure of a single system or messages can be tolerated without affecting the application.
- The entire system remains in operation without interrupts under the same temporal conditions.
- Multiple redundant guardians can be implemented to meet the highest safety requirements.

Fault-Tolerant *Capabilities*

- As a simple master-slave synchronization protocol for industrial control.
- Cross-domain usage of TTEthernet increases the probability of latent failure detection.
- TTEthernet tolerates multiple inconsistent faults.
- A more cost-efficient realization of system architectures that require tolerance of multiple concurrent failures in the system.
- TTEthernet tolerates arbitrary end system failures.
- TTEthernet tolerates arbitrary transient disturbances even in presence of permanent failures.

Network Structure

- TTEthernet supports all physical layers specified in IEEE 802.3 for switch-based networks
- Switches in TTEthernet have the central role of organizing the data communication.
- TT messages are routed in the switch according to a predefined schedule with as little delay as possible.
- Precise planning at the time of system design precludes resource conflicts at runtime.
- TT messages have the highest priority level.
- delays are precluded.
- RC messages are routed with little delay.

Network Structure

- The TT messages take priority over the RC messages.
- TT messages can delay RC messages.
- The switch is responsible for arranging several RC messages at an outgoing port.
- BE messages always have little priority.
- RC and TT messages can delay or discard BE messages at the same outgoing port.
- BE messages are transmitted after all pending RC messages.
- TTEthernet switches allow the simultaneous distribution of TT messages to groups of end systems

Supported Topologies

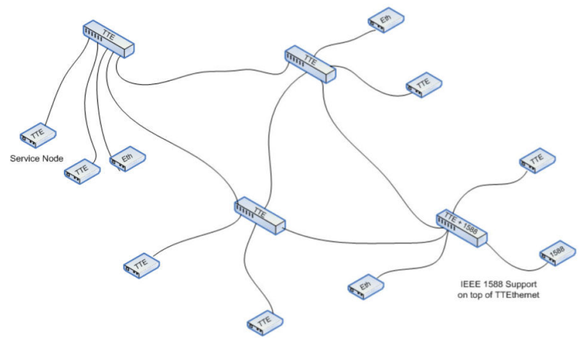
- TTEthernet allows synchronizing local clocks in a distributed computer network.
- Communication links and switches are said to form a communication channel between end systems.
- End systems can be connected directly to each other via bi-directional communication links.
- Term device to refer to a physical device that can be either end system or switch.
- A device is regarded as an end system or a switch is determined by its usage rather than its physical appearance.

Synchrony

- Events in time-triggered systems occur at predefined times with a precision at the single microsecond level.
- The system design specifies when the TT messages are transmitted by which participants and who shall receive them.
- No data congestion in the switches
- Transmits clock synchronization messages to keep the clocks of the end systems and switches in synchrony.
- A redundant hierarchical master-slave method that has a distributed fault-tolerant majority of master nodes and master switches to provide the time in the system.
- Can be combined with other mechanisms such IEEE 1588.

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

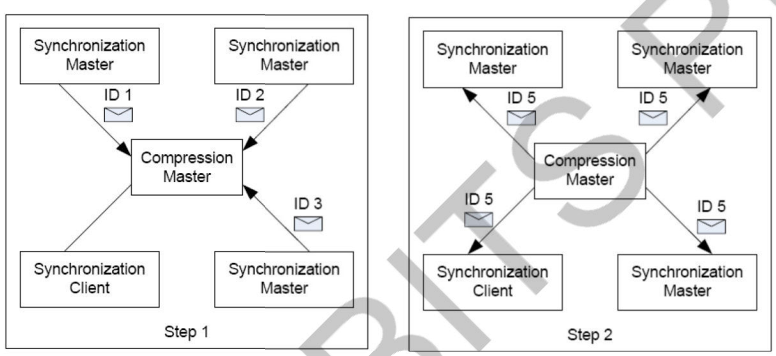
Synchrony



Synchrony

- IEEE 1588 specifies a synchronization protocol for Ethernet.
- The global time base of TTEthernet can be leveraged to synchronize native IEEE 1588 synchronization clients,
- TTEthernet provides means to compensate for delays through the TTEthernet network.
- The clock synchronization messages can be handled as native IEEE 1588 clock synchronization messages.

Synchronization Approach



Synchronization Approach

- TTEthernet takes a two-step approach to synchronization.

1) synchronization masters send protocol control frames to the compression masters.

2) compression masters then calculate an averaging value from the relative arrival times of these protocol control frames and send out a new protocol control frame.

- End systems can be configured as synchronization masters.
- Switches as compression masters.
- Switches and end systems not configured either as synchronization or compression masters will be configured as synchronization clients.

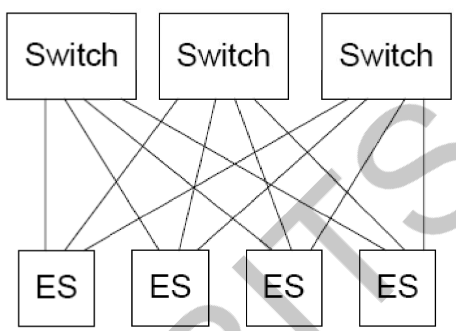
Synchronization Topology

Network Level	y Synchronization Domains, (y,z) Synchronization Priorities
Multi-Cluster Level	One Synchronization Domain, x Synchronization Priorities
Cluster Level	One Synchronization Domain, One Synchronization Priority
Device Level	Synchronization Masters, Synchronization Clients, Compression Masters

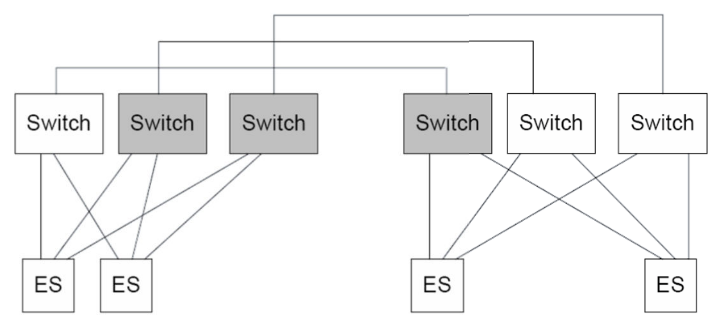
Synchronization Topology

- TTEthernet distinguishes four different levels in synchronization topology.
- TTEthernet specifies the concept of a cluster.
- A TTEthernet cluster is a group of end systems and switches that have the same synchronization priority and synchronization domain.
- Different clusters shall be able to run in isolation, but shall be able to operate in a master-slave mode, once a high priority cluster joins the network or is powered on.

TTEthernet simple cluster with three redundant channels



TTEthernet cascaded cluster with three redundant channels



BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

Synchronization Topology

- Synchronization in a multi-cluster system is usually done according a master-slave paradigm, where the devices will synchronize towards the highest synchronization priority.
- A synchronization domain is a group of TTEthernet clusters that will not synchronize to each other.
- Dataflow between two TTEthernet clusters in different synchronization domains can be done using RC or BE traffic.

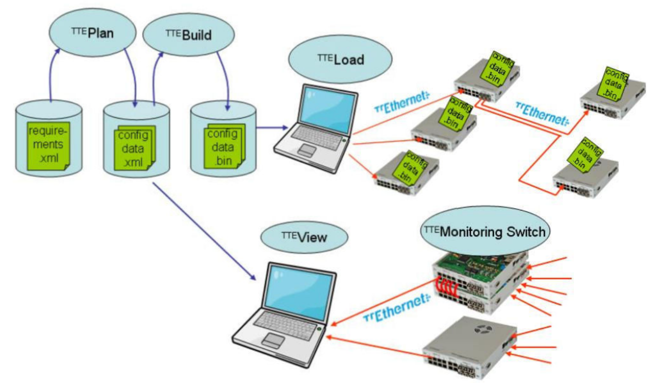
Variable Implementation

- Integration for end systems can be implemented in hardware or software.
- Can be connected to conventional Ethernet systems without affecting the predefined behavior.
- A lack of bandwidth for this additional system.

PC-based embedded systems

A PC with a conventional Network Interface Card (NIC) can send and receive BE messages.
- Equipped with dedicated software the PC can also receive and analyze TT and RC messages.
- A PC with conventional NIC and a TTEthernet stack is a software-based end system (SES) that allows the reception and transmission of TTC, RC and BE messages.
- A PC with specific TTEthernet NIC can send and receive TT, RC and BE messages with the highest temporal precision.

TTEthernet tool chain covers the entire life cycle of a network



innovate achieve lead

Standardization Activities

At present, SAE AS-2D plans to standardize TTEthernet, and to work closely with other standardization bodies in their respective industry (e.g. ISO, SAE J, IEEE and others) with target date 2012.

TTEthernet

- TTEthernet enables time-triggered communication over Ethernet networks in all application areas.
- The network provides all necessary mechanisms for applications as diverse as classical web services and time critical and safety-critical control system in airplanes.
- Existing networks can be extended step by step using TTEthernet-capable switches and end systems without the need to change existing applications and end systems.
- Reducing network solutions to established and recognized Ethernet standards opens up saving potentials that secure major advantages in competitive markets.
- Honeywell is the first company to use TTEthernet for production programs in the aerospace and automation industries.
- Used in aerospace applications but also in completely new application areas.

TTEthernet Scheduling:

In a TT-Ethernet system operating at 1 Gbps, a time-triggered message with a payload of 1200 bytes is sent every 0.2 ms. Each frame adds a header overhead of 26 bytes. Determine the bandwidth utilization percentage of this single TT message stream.

Solution:

Frame size:

1200 bytes + 26 bytes = 1226 bytes = 9808 bits

- Data rate:

  (9808 bits / 0.2×10^{-3} s) = 49.04 Mbps

- Bandwidth utilization:

  (49.04 Mbps / 1000 Mbps) x 100% = 4.904%\

Time-Triggered Ethernet (TT Ethernet) Frame Rate

A TT Ethernet network running at 100 Mbps transmits frames containing 1000 bytes of payload and a 42-byte overhead every 250 µs. Calculate the bandwidth utilization percentage for this periodic traffic.

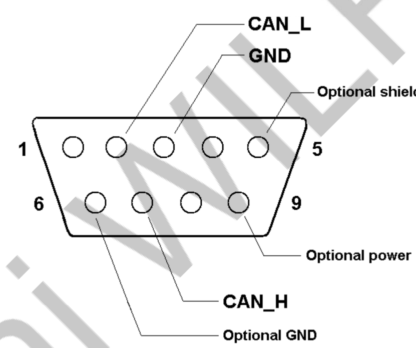
Solution:

- Frame size:
  1000 + 42 = 1042 bytes = 8336 bits
- Frame rate:
  (8336 bits / 250×10−6 s) = 33.344 Mbps
- Utilization:
  (33.344 Mbps / 100 Mbps) × 100% = 33.344%

TTCAN

CAN BUS

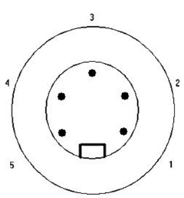
CAN BUS MALE CONNECTOR



This is a male connector viewed from the connector side, or a female connector viewed from the soldering side.

1	-	Reserved
2	CAN_L	CAN_L bus line (dominant low)
3	CAN_GND	CAN Ground
4	-	Reserved
5	(CAN_SHLD)	**Optional** CAN shield
6	(GND)	**Optional** CAN ground
7	CAN_H	CAN_H bus line (dominant high)
8	-	Reserved (error line)
9	CAN_V+	**Optional** power

CAN BUS 5 Pin MINI-C

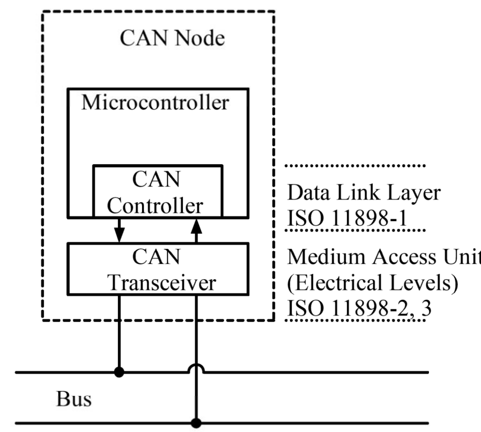


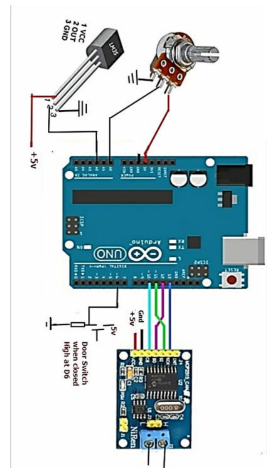
Male (pins)

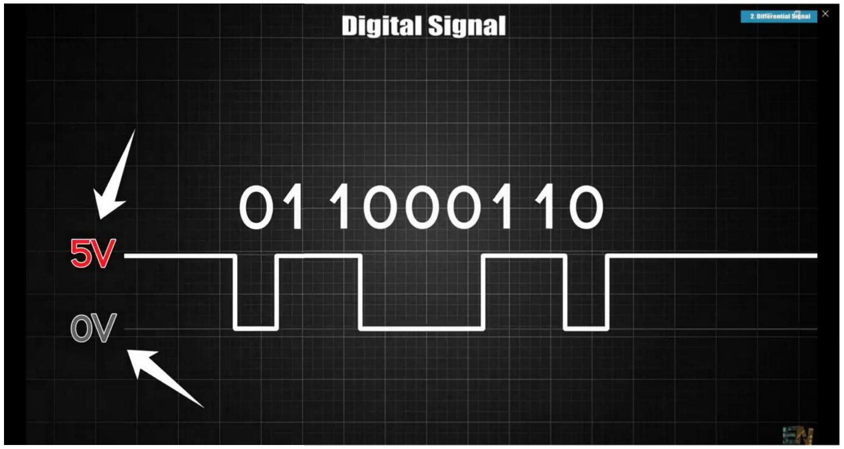
Pin	Function	DeviceNet Color
1	Drain	Bare
2	V+	Red
3	V-	Black
4	CAN_H	White
5	CAN_L	Blue

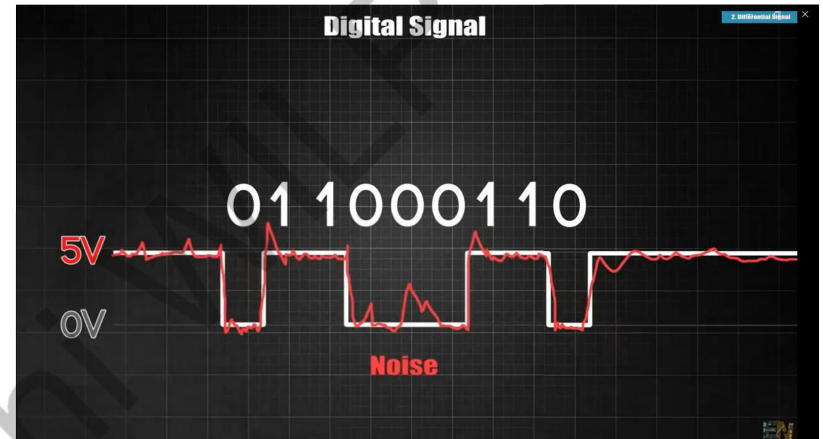
Pin	Function	Recommended cable colour
1	Power negative	Black
2	CAN_H	White
3	Optional Signal GND	Yellow
4	Optional Initiate	Gray
5	Power positive	Red
6	CAN_L	Blue

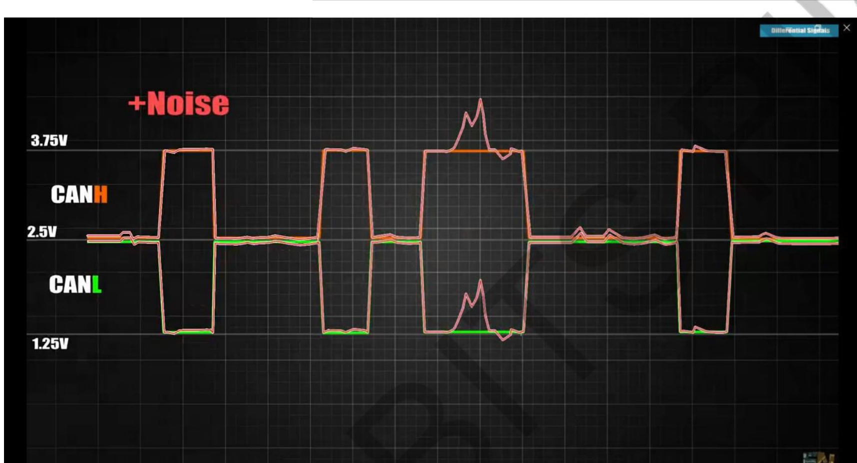
CAN NODE

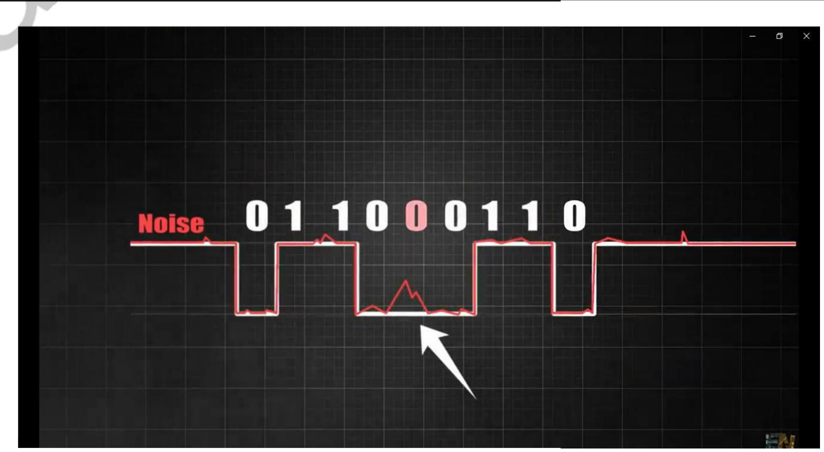




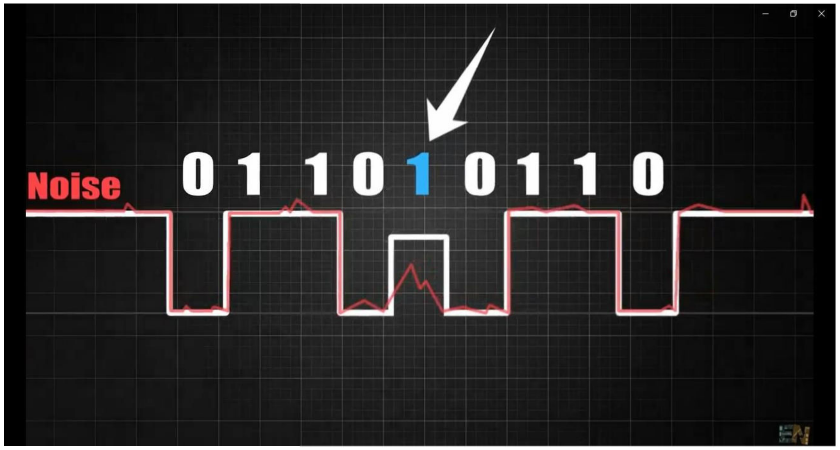

Digital Signal
01 1000110
5V
0V

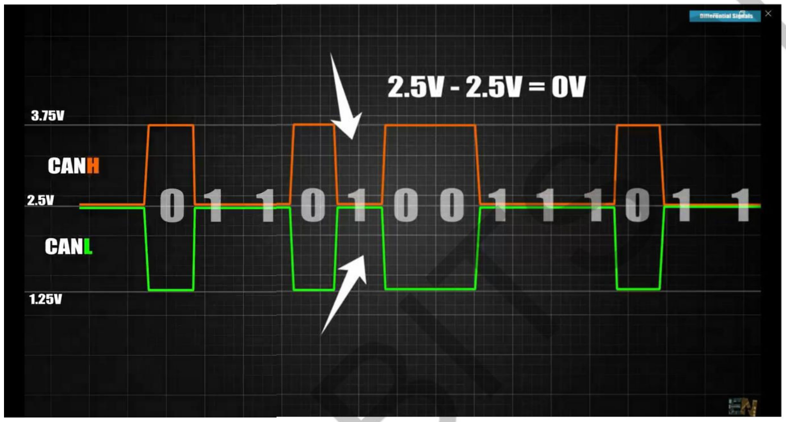

Digital Signal
01 1000110
5V
0V
Noise

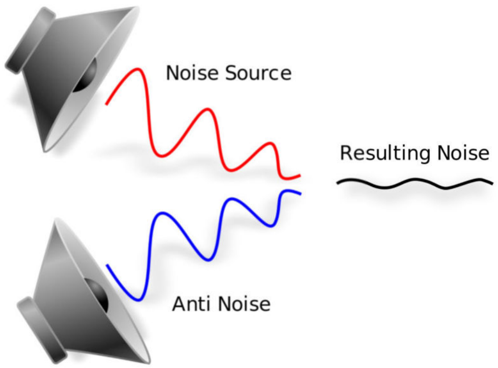

+Noise
3.75V
CANH
2.5V
CANL
1.25V


Noise
0 1 1 0 0 0 1 1 0

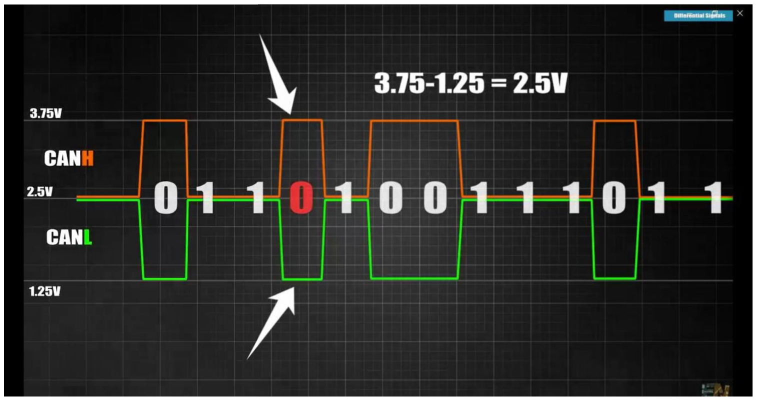
BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED
LEARNING PROGRAMMES





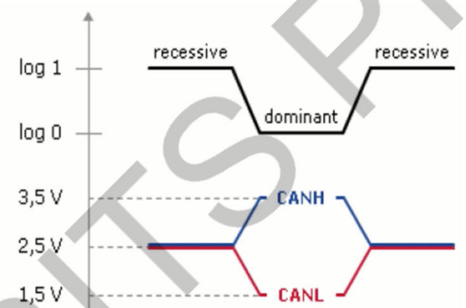


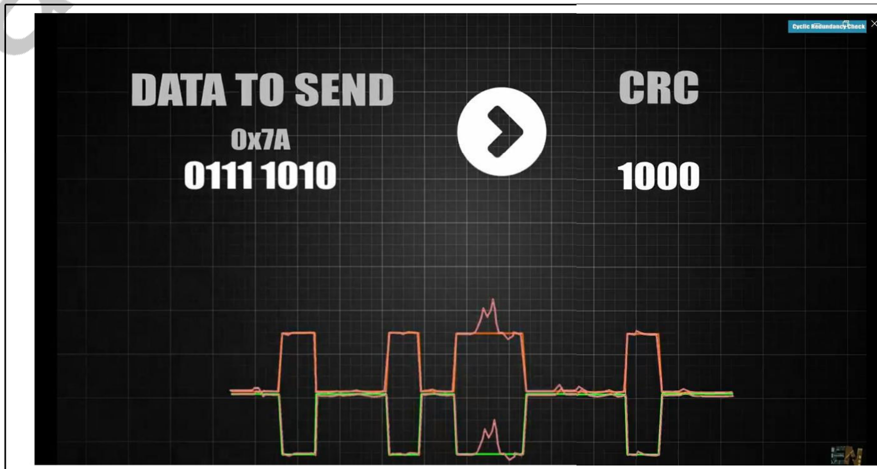
innovate achieve lead



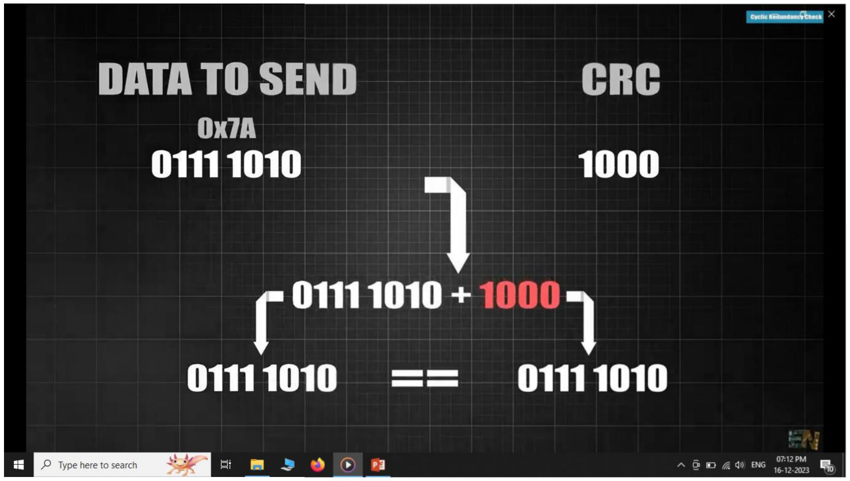
CAN BUS LOGIC LEVELS

- Logic 1 or Recessive
  CANH & CANL both are at 2.5V
- Logic 0 or Dominant
  CANH at 3.5V & CANL at 1.5V
- Idle bus state is Recessive
- Dominant state dominates
  Recessive state



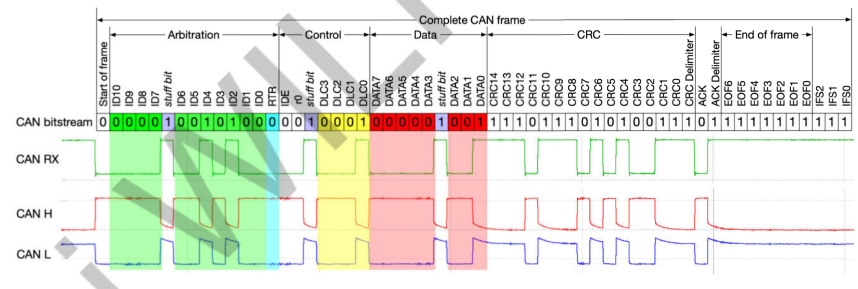


BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

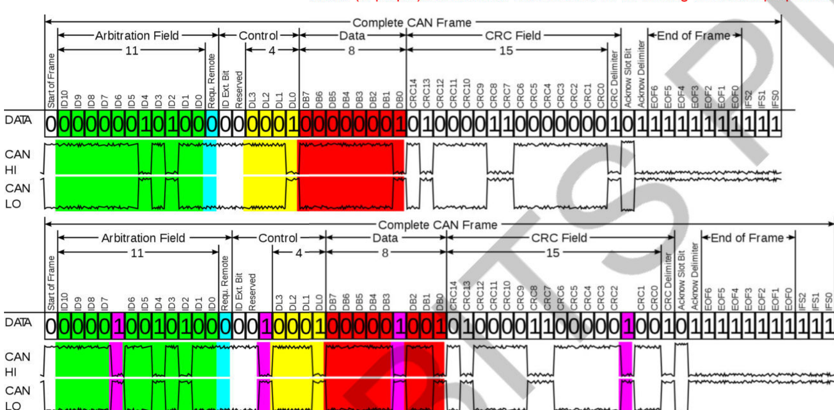


CAN BUS

A complete CAN bus frame, including stuff bits, a correct CRC, and inter-frame spacing



CAN frame before and after the addition of stuff bits (in purple). An incorrect CRC is used for bit stuffing illustration purposes.



CAN FD – FLEXIBLE DATA RATE

Simplified Diagram of Classical CAN and CAN FD Frames:

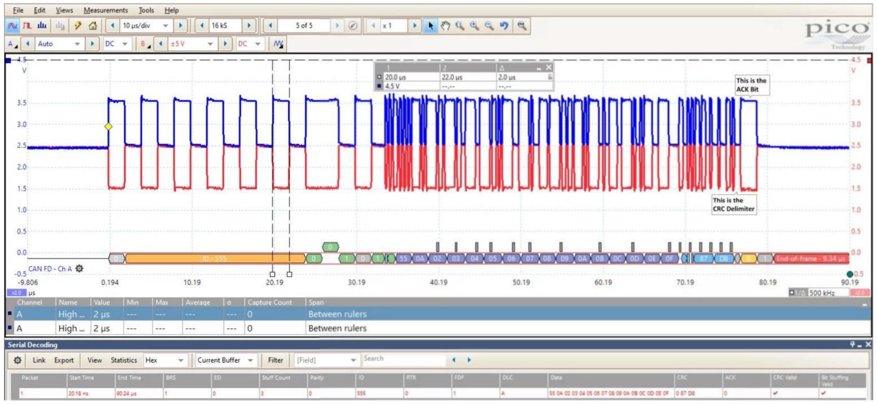
Classical CAN

CAN FD

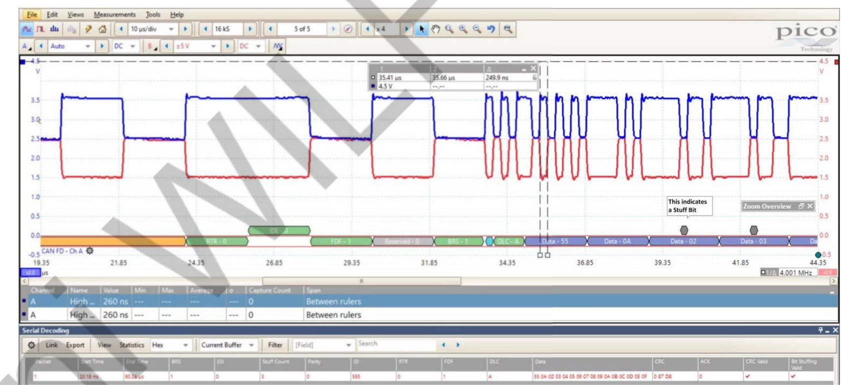
innovate achieve lead

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

CAN FD – FLEXIBLE DATA RATE



CAN FD – FLEXIBLE DATA RATE



CAN FD – FLEXIBLE DATA RATE

Coding of the numbers of data bytes and CRC bits for DLC values

DLC Indicator	DLC (Binary)	Classical CAN (Bytes)	CAN FD (Bytes)	Classical CAN CRC-	CAN FD CRC-
0	0000	0	0	15	17
1	0001	1	1	15	17
2	0010	2	2	15	17
3	0011	3	3	15	17
4	0100	4	4	15	17
5	0101	5	5	15	17
6	0110	6	6	15	17
7	0111	7	7	15	17
8	1000	8	8	15	17
9	1001	8	12	15	17
10	1010	8	16	15	17
11	1011	8	20	15	21
12	1100	8	24	15	21
13	1101	8	32	15	21
14	1110	8	48	15	21
15	1111	8	64	15	21

Table 1

Specifications	CAN	TTCAN
Full form	Controller Area Network	Time-Triggered CAN
Messaging type	Event triggered messages	Event and Time triggered messages
Network synchronization	Priority based arbitration	Global reference time
Node control	Autonomous	Autonomous, master/slave
Message retransmission	Supported	Supported
Error management	CRC Frame field, Bit monitoring, Bit stuffing, Error Frames, Overload frames	Same as mentioned in CAN
bandwidth	125 Kbps maximum	10 Mbps maximum
physical	2 wire with interference protection	2 wire with interference protection
Applications	Displays, alarm systems, lighting, A/C seat & mirror adjustments, power windows, windshield wipers, headlamps	Engine, transmission, Steering, Braking, Suspension, Safety, Assistance, Diagnostics
Cost	Medium	Higher

innovate achieve lead

Motivation of TTCAN

- Under the bitwise arbitration of CAN, the access may be delayed, if some other message is already in the process of transmission or if another message with higher priority also competes for the bus.
- Even the message with the highest priority may experience a small latency.
- The lower the priority of a message is, the higher the latency jitter for the media access

The Goals of TTCAN

- Reduce latency jitters
- Guarantee a deterministic communication pattern on the bus
- Use the physical bandwidth of a CAN network much more efficiently

TTCAN: Two Level Extension

- Extension level 1
  - Guarantees the time triggered operation of CAN based on the reference message of a time master
  - Fault tolerance of that functionality is established by redundant time masters (potential time masters)
- Extension level 2
  - A globally synchronized time base is established
  - A continuous drift correction among the CAN controllers is realized

Overview of TTCAN Mechanisms

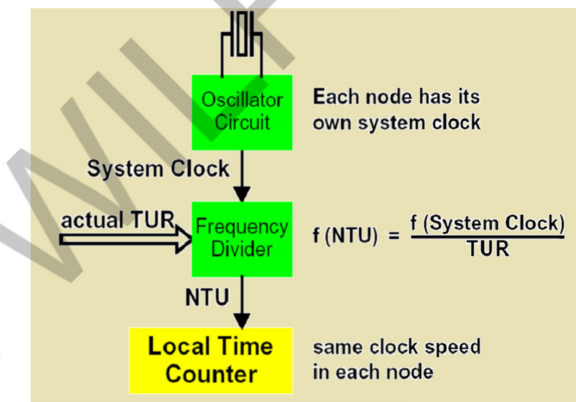
- Time bases, provided either by an internal or by an external clock
  - Local_Time, Cycle_Time, and Global_Time
- Three different types of time windows
  - Exclusive time window, arbitrating time window, and free time window
- Time-triggered and periodic communications clocked by a time master's reference message
- The system matrix specifies the sequence of messages transmitted in each basic cycle
- At least one event-trigger should be freely programmable by the control unit

Local Time

- Local_Time, a 16 bit integer value that is incremented each Network Time Unit (NTU)
- The length of the NTU is the same for all nodes
- It is generated locally, based on the local system clock period t_{sys} and the local Time Unit Ratio (TUR), NTU=TUR*t_{sys}
- Different system clocks in the nodes require different (non-integer) TUR values

– TUR is a non-integer value and may be adapted to compensate for clock drift or to synchronize to an external time base.

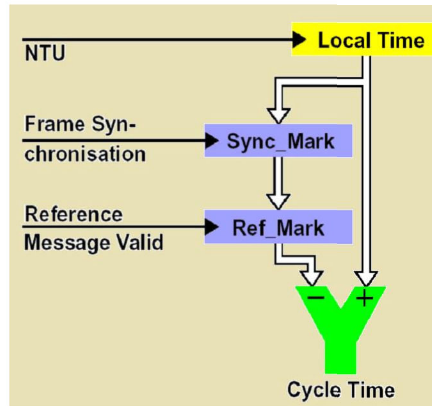
Local Time



Cycle Time

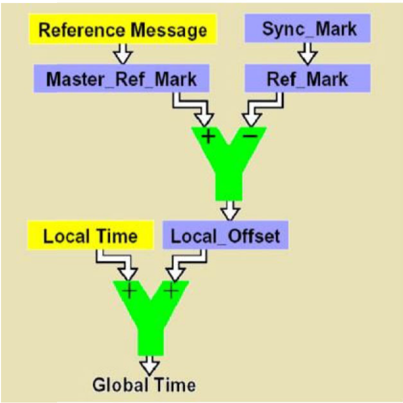
- Each valid Reference Message starts a new basic cycle and causes a reset of each node's Cycle_Time.
- The value of Local_Time is captured as Sync_Mark at the start of frame (SOF) bit of each message. When a valid Reference Message is received, this message's Sync_Mark becomes the new Ref_Mark
- Cycle_Time is the actual difference between Local_Time and Ref_Mark, restarting at the beginning of each basic cycle when Ref_Mark is reloaded
- Even in a software implementation of TTCAN, the capturing of Local_Time into Sync_Mark at each SOF must be done in hardware

Cycle Time

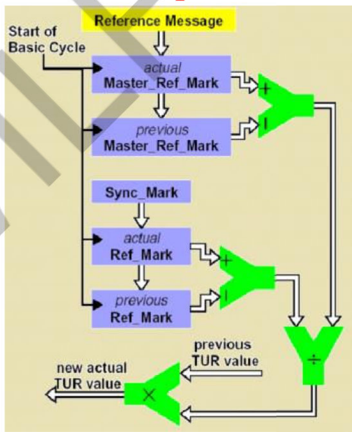


BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

Global Time



Drift Compensation



Time Windows

• Exclusive window

– Exclusively reserved for one message, without competition for the CAN network access

– The automatic retransmission of messages that could not be transmitted successfully is disabled, guaranteeing that messages in exclusive time windows are not delayed

• Arbitrating window

– During which messages can compete for the bus by the bitwise arbitrating mechanism of CAN

• Free window – Reserved for further extensions of the network

The Reference Message

• The reference message can be easily recognized by its identifier

• In extension level 1

– The reference message only holds some control information of one byte, the rest of a CAN message can be used for data transfer

• In extension level 2

– The reference message holds additional control information, e.g. the global time information of the current TTCAN time master

– The reference message of level 2 covers 4 bytes while downwards compatibility is guaranteed. The remaining 4 bytes are open for data communication as well

innovate achieve lead

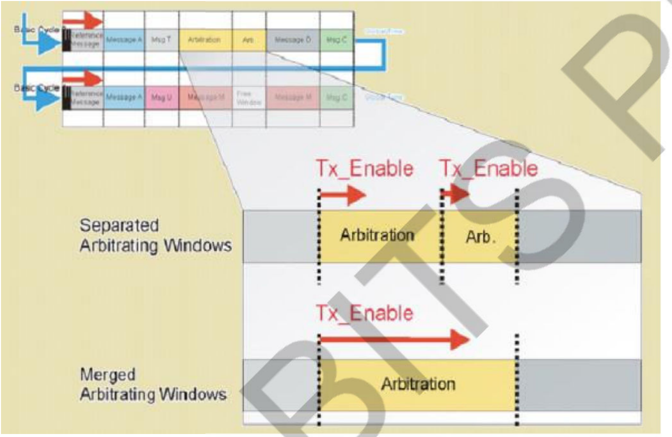
BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

The System Matrix

- In TTCAN not all basic cycles necessarily have to be the same
- Different basic cycles are distinguished by the cycle count
  – A cycle count is incremented each cycle up to the maximum value after which it is restarted again
- System matrix is obtained by combining all those different cycles
  – It represents the communication overview of a TTCAN network
- System matrix allows another useful exception
  – Ignore the columns in the case of two or more arbitrating time windows in series

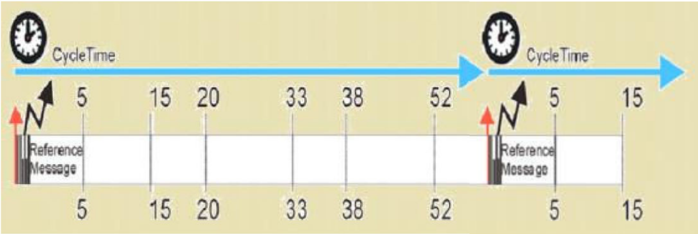
Example of a TTCAN System Matrix

Merged Arbitrating Windows



Cycle Time and Time Marks

- A time mark furthermore consists of the base mark and the repeat count information.
  – The base mark determines the number of the first basic cycle after the beginning of the matrix cycle in which the message must be sent/received.
  – The repeat count determines the number of basic cycles between two successive transmissions/receptions of the message.



innovate achieve lead

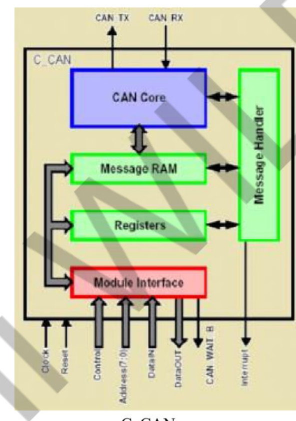
TTCAN Implementation

• The TTCAN is expanded by two functional blocks: the Trigger Memory and the Frame Synchronization Entity (FSE)

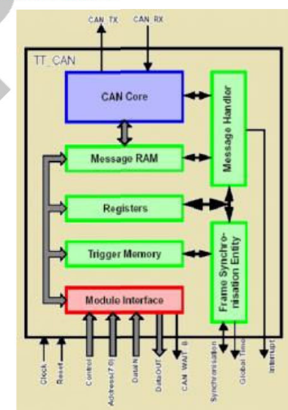
• The Trigger Memory stores the time marks of the system matrix that are linked to the messages in the Message RAM; the data is provided to the Frame Synchronization Entity

• The Frame Synchronization Entity is the state machine that controls the time triggered communication. It synchronizes itself to the reference messages on the CAN bus, controls the cycle time, and generates Time Triggers. It is divided into six blocks:
– TBB: Time Base Builder
– CTC: Cycle Time Controller
– TSO: Time Schedule Organizer
– MSA: Master State Administrator
– AOM: Application Operation Monitor
– GTU: Global Time Unit

TTCAN Implementation



C_CAN



TT_CAN

Vehicular Bus (CAN Bus Arbitration)

Two CAN nodes simultaneously attempt to transmit messages on a CAN bus at 1 Mbps. Their IDs are Node A (0x3F2) and Node B (0x1A4). Each identifier has 11 bits. Determine which node gains access to the bus, assuming standard CAN arbitration, and find the maximum possible arbitration delay (in µs).

Solution: CAN Bus Arbitration

Convert IDs to binary (11-bit):

Node A ID: 0x3F2 (01111110010)

Node B ID: 0x1A4 (00110100100)

Comparing from MSB:

First difference at bit 2:

Node A = 1 (dominant)

Node B = 0 (recessive)

Arbitration delay = 11 bits / 1 Mbps = 11 µs.

TTCAN Network Sizing

A TTCAN system operates at 1 Mbps. Each node sends a periodic message every 1 ms. Each message has 8 bytes of payload and 55 bits overhead. Calculate the maximum number of nodes that can operate simultaneously without exceeding 80% bus utilization.

Solution:

- Bits per message:
  8 bytes × 8 + 55 bits overhead = 64 + 55 = 119 bits

  Data per node per second:

  (119 bits / 1×10^{-3} s) = 119,000 bps

  Allowed bus utilization at 80%:

  $0.8 \times 1 \times 10^6$ bps = 800,000 bps

  Maximum number of nodes:

  (800,000 / 119,000) ≈ 6.72 → 6 nodes (integer value)

CAN Bus Message Arbitration Delay

Two nodes on a CAN bus at 500 kbps simultaneously send messages with identifiers Node A: (0x5A7) and Node B: (0x4D9). Determine which node wins the arbitration and calculate the maximum arbitration time (in µs).

Solution:

- Binary IDs (11 bits):
  - Node A (0x5A7): 10110100111
  - Node B (0x4D9): 10011011001
- Arbitration comparison from MSB (left to right), difference at bit 3:
  - Node A: 1 (dominant), Node B: 0 (recessive)
  - Node A wins arbitration.
- Arbitration delay (11 bits):

(11 bits / 500×10^3 bps) = 22 µs

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

TTCAN Node Limits

A TTCAN network operates at 250 kbps with each node transmitting a message of 4 bytes of payload and 47 bits overhead every 2 ms. If the bus utilization should not exceed 75%, determine the maximum number of nodes allowed.

Solution:

- Bits per message:
  4 bytes × 8 + 47 bits = 79 bits
- Data per node per second:
  79 bits 2×10^{-3} s = 39,500 bps
- Maximum bus capacity at 75%:
  0.75 × 250,000 bps =187,500 bps
- Nodes:
  (187,500 / 39,500) ≈ 4.746≈4 nodes (integer)

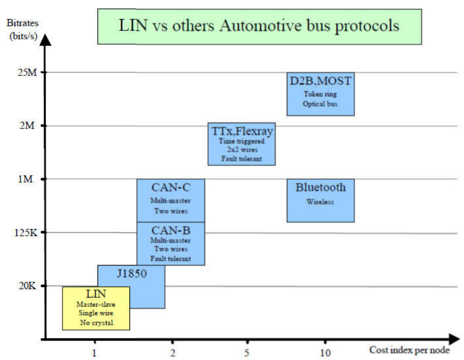
LIN

SAE Network classification

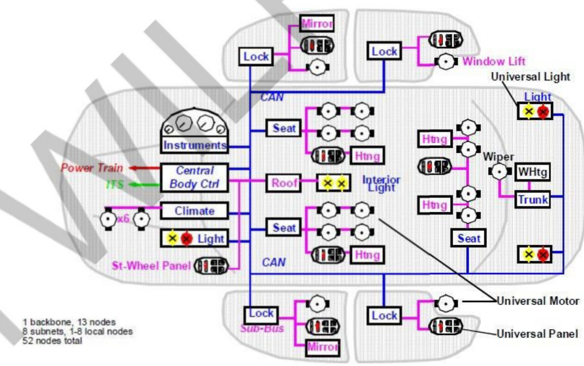
- Class A networks
  - Low Speed (<10K bits/second)
  - Convenience features: entertainment, audio, trip computer, etc.
- Class B networks
  - Medium Speed (10K b/s to 125K b/s)
  - General information transfer: instrument cluster, vehicle speed, legislated emissions data/diagnostics, etc.
- Class C networks
  - High Speed (125K b/s to 1M b/s or greater)
  - Real-time control: powertrain control, braking, vehicle dynamics, etc.

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

Costs and Speeds for Automotive Networks



Automotive Body Network

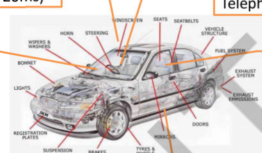


Typical LIN Applications

Roof:
(high amount of wiring) Rain Sensor, Light Sensor, Light Control, Sun Roof …
(Rain Sensor needs to be interrogated every 10-20ms)

Steering Wheel:
(very many controls are going to be positioned on the steering wheel) Cruise Control, Wiper, Turning Light, …
Optional: Climate Control, Radio, Telephone, etc.

Climate:
Many Small Motors Control Panel

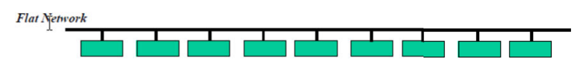


Seat:
Many Seat Position Motors, Occupancy Sensor, Control Panel

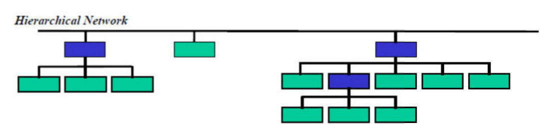
Door/window/seat:
Mirror, Central ECU, Mirror, Switch, Window Lift, Seat Control Switch, Door Lock, etc.

Hierarchical Network Struc

- CAN based network

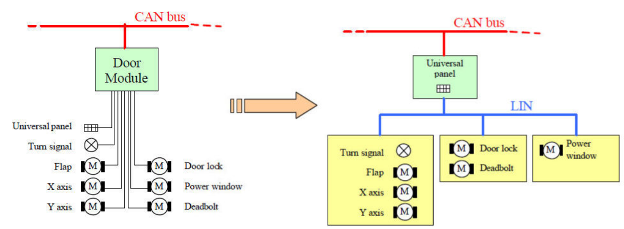


- CAN + LIN



innovate achieve lead

LIN implementation

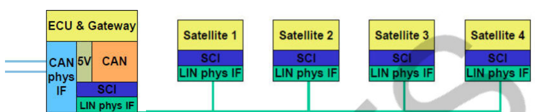


SubNets

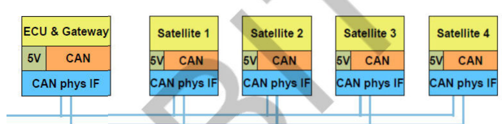
- Necessary to reduce Busload on main Bus
- Solutions
  - CAN
    - Automotive Standard Bus
    - Compatible with Main Bus
    - Expensive (Die Size/ Dual Wire)
  - Serial Sub Bus
    - no standard Bus System
    - not compatible with Main Bus
    - inexpensive
    - SCI-Based: Interface exists even on cheap devices
    - Interface can easily be reconstructed by ASIC or CPLD

Sub-Network: LIN vs. CAN

- LIN



- **Dual Wire CAN**



Comparison of general features of LIN and CAN

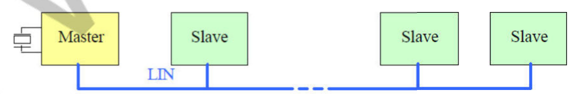
Features	LIN	CAN
Medium access control	single master	multiple masters
Typical bus speed	2.4, 9.6 and 19.2kbps	62,5...1000kbps
Multicast message routing	6 bit identifier	11/29 bit identifier
Typical size of network	2...16 nodes	4...20 nodes
Data byte per frame	0... 8	2...8
Transmission time for 4 data bytes	6 ms at 20kbps	0.8 ms at 125kbps
Error detection (data field)	8-bit checksum	15-bit CRC
Physical layer	single-wire, 12 V	twisted-pair, 5 V
Quarts/ceramic resonator	master only	Yes
Relative cost per network connection	x 0.5	x 1.0

LIN History

- LIN (Local Interconnect Network) is a cost-effective and deterministic communication system for connecting ECUs with smart sensors, actuators and controls.
- LIN1.0: 1999.7
- LIN1.1: 2000.3
- LIN1.2: 2000.11
- LIN1.3: 2002.12
- LIN2.0: 2003.9
- LIN2.1: 2006.11
- J2602: 2004.8 the Society of Automotive Engineers

Aim of LIN

- Open Standard
- Easy To Use
- Components available today
- Cheaper than CAN or J1850
- A LIN bus length is limited to 40 meters and up to 16 ECUs could be connected.



Sub Bus Concept

Basic Requirements:

- Satisfy Need for a Standard for Sub Busses
- Cost driven: The solution must be cheaper than CAN
- Reliability: Same Level as CAN expected
- Long Term Solution
- Logical Extension to CAN
- Scalable: Capability to extend Systems with additional nodes
- Lowering Cost of Satellite nodes:
  - No Crystal or Resonator
  - Easy implementation
  - Simple State Machines
- Low Reaction Time (100 ms max)
- Predictable Worst Case Timing

LIN Concept

Technical Solution

- Low cost single-wire implementation (enhanced ISO 9141)
- Speed up to 20Kbit/s (limited for EMI-reasons)
- Single Master / Multiple Slave Concept
  - No arbitration necessary
- Low cost silicon implementation based on common UART/SCI interface hardware
  - Almost any Microcontroller has necessary hardware on chip!
- Self synchronization without crystal or ceramics resonator in the slave nodes
  - Significant cost reduction of hardware platform
- Guaranteed latency times for signal transmission (Predictability)

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

Master / Slave Protocol

Master Task
- Determines order and priority of messages.
- Monitors Data and check byte and controls the error handler.
- Serves as a reference with its clock base (stable clock necessary)
- Receives Wake- Up Break from slave nodes

• Slave Task
- Is one of 2-16 members on the bus
- Receives or transmits data when an appropriate ID is sent by the master.
- The node serving as a master can be slave, too!

Master / Slave Protocol

Master
- has control over the whole Bus and Protocol
- The master controls which message at what time is to be transferred over the bus. It also does the error handling.
- To accomplish this the master
  - sends Sync Break
  - sends Sync Byte
  - sends ID-Field
  - monitors Data Bytes and Check Byte, and evaluates them on consistance
  - receives WakeUp Break from slave nodes when the bus is inactive and they request some action.
  - serves as a reference with it's clock base (stable clock necessary)

Master/Slave Protocol

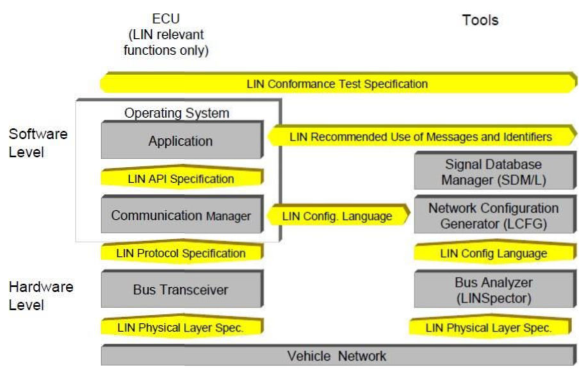
Slave
- Is one of 2-16 Members on the Bus and receives or transmits Data when an appropriate ID is sent by the master.
  - Slave snoops for ID.
  - According to ID, slave determines what to do.
- either receive data
- or transmit data
- or do nothing.
  - When transmitting the slave
- sends 1, 2, 4, or 8 Data Bytes
- sends Check-Byte
  - The node serving as a master can be slave, too!

Schedule table

- The master task (in the master node) transmits frame headers based on a schedule table.
- The schedule table specifies the identifiers for each header and the interval between the start of a frame and the start of the following frame.
- The master application may use different schedule tables and select among them.

innovate achieve lead

LIN Standard - Overview



LIN NETWORK

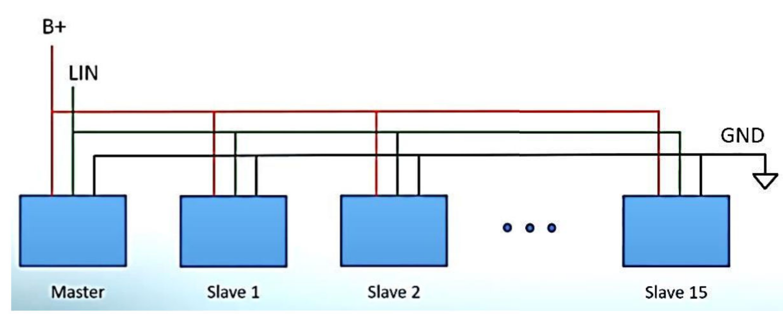
- Physical Layer
- Node Model
- LIN Protocol
- Firmware Case Study
- Case Study for Body Control Module Systems

LIN NETWORK

Asynchronous	**Message-Oriented**
Single-Master	**6 Bit ID**
1-Wire Communication	**Checksum**
Max 20 Kbps	**Data Max -> 8 Bytes**

LIN Physical Layer

- Typical network …



LIN PHYSICAL LAYER

- In the **OSI model**, the **Physical Layer** is concerned with "putting the bits on the wire."
- The LIN physical layer is based on ISO 9141 (the K-line bus).
- It consists of the bidirectional bus line LIN which is connected to the transceiver of every bus node and is connected via a termination resistor and a diode to the positive battery node, V_{BAT}.

ISO 9141-2

This protocol has an asynchronous serial data rate of 10.4 kbit/s.
It is somewhat similar to RS-232; however, the signal levels are different, and communications happen on a single, bidirectional line without additional handshake signals. ISO 9141-2 is primarily used in Chrysler, European, and Asian vehicles.

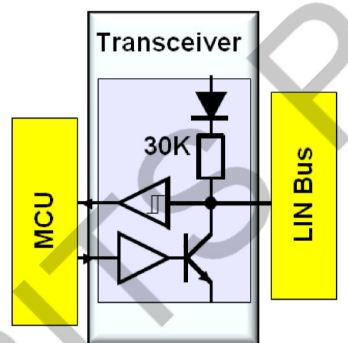
- pin 7: K-line
- pin 15: L-line (optional)
- UART signaling
- K-line idles high, with a 510 ohm resistor to V_{batt}
- The active/dominant state is driven low with an open-collector driver.
- Message length is Max 260Bytes. Data field MAX 255.

LIN Physical Layer

- LIN protocol is UART based
- LIN bus has pull-up resistors in every slave (typ. 30K) and master node (1K in Master)
- neoVIs have ability to switch on Master Pull-Up Resistor for Master Simulation

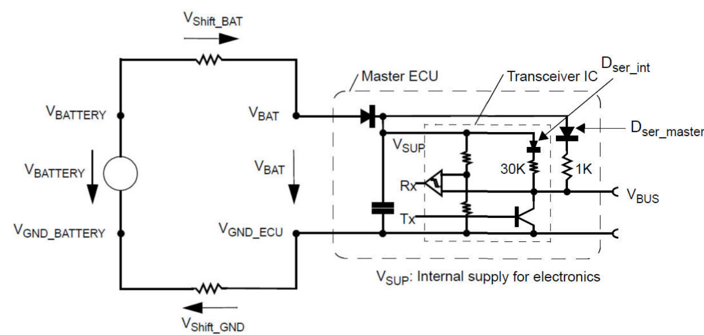
Transceiver

- The LIN bus operates between 9 V and 18 V.
- Typically, the microcontroller LIN I/O pins voltage levels are adjusted to the LIN bus levels by a transceiver.
- This allows the microcontroller to operate at 3/5 V levels, while the bus operates at higher levels.



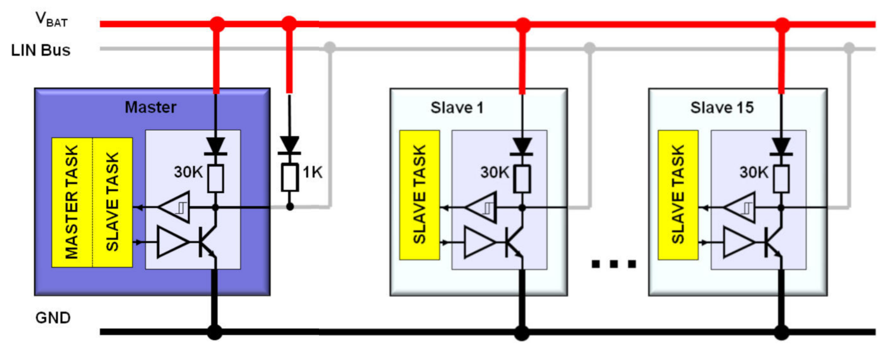
Simplified LIN ECU Node:

- Electronic Control Unit (ECU) is a generic term for any embedded system that controls one or more of the electrical system or subsystems in a transport vehicle.



A "Master" LIN ECU Node (From the LIN Standard)

- Diodes D_{ser_int} and D_{ser_master} are mandatory to prevent uncontrolled powering of the ECU node from the bus line, in the case of a battery loss.



LIN Cluster - A LIN cluster is composed of a single-wire bus, a Master node and up to 15 Slave nodes.

Key Characteristics

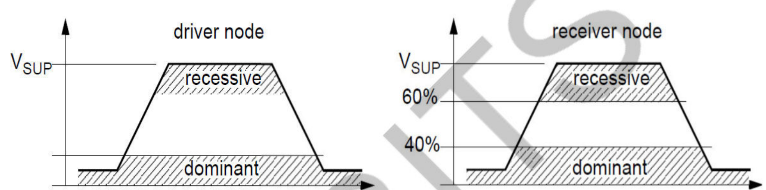
- Single Master, Multiple Slaves (up to 15 Slaves)
- Single wire plus ground signaling: (VBAT, GND, LIN (9 V-18 V))
- From 1 kbit/s up to 20 kbit/s
- Dominant/Recessive bits (like the CAN bus)
- Total length of bus line: 40 meters max
- Terminations: Master 1 kΩ, Slave 30 kΩ

Signal Levels

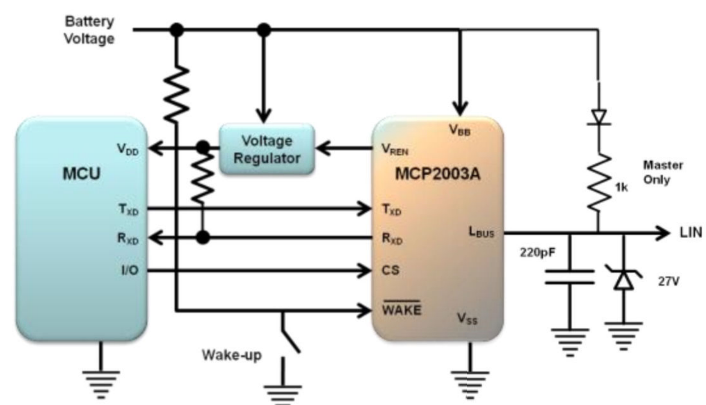
•Dominant - Bus LOW - Logic 0
•Recessive - Bus HIGH - Logic 1

A LIN Network implements Wired *and* Signalling:

•All nodes must be HIGH (Recessive) in order to transmit a Logic 1
•Only one node LOW (Dominant) will transmit a Logic 0

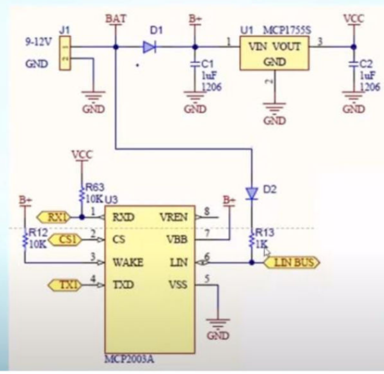


Typical LIN Node Configurations - Stand Alone Transceiver MCP2003B

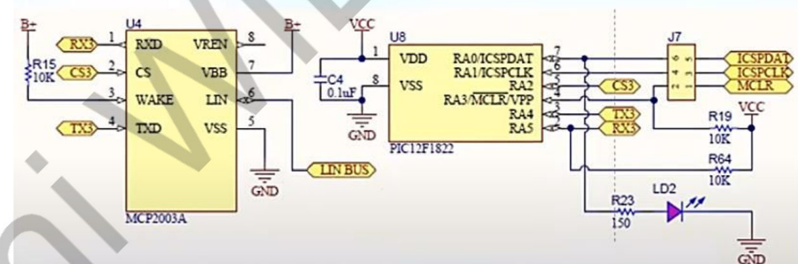


LIN Physical Layer

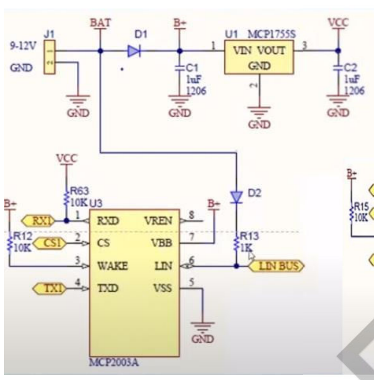
LIN Master example



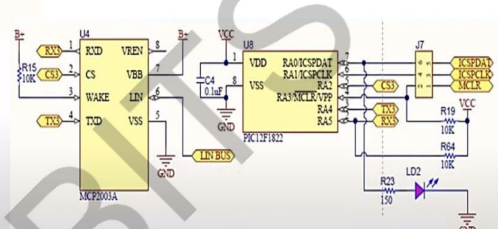
- LIN Slave example



LIN MASTER

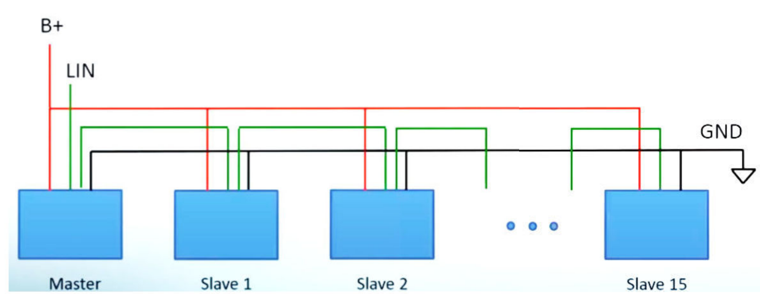


LIN Slave



LIN Physical Layer

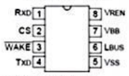
- Typical network (auto-addressing technique) ...



- LIN Transceiver examples

MCP2003A

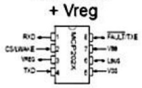
Stand Alone Transceiver



- 3rd Generation LIN transceiver
- No integrated Vreg
- LIN 1.3, 2.X and SAE J2602 compliant
- E-Temp
- DFN/SOIC/PDIP 8L packages
- Industry Std pinout
- Note 1

MCP2021A

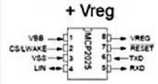
Transceiver + Vreg



- 3rd Generation LIN transceiver
- 3.3V and 5V 70mA Vreg Options available
- LIN 1.3, 2.X and SAE J2602 compliant
- E-Temp
- DFN/SOIC/PDIP 8L packages
- Note 1

MCP2025

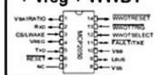
Transceiver + Vreg



- 3rd Generation LIN transceiver
- 3.3V and 5V 70mA Vreg Options available
- LIN 1.3, 2.X and SAE J2602 compliant
- E-Temp
- DFN/SOIC/PDIP 8L packages
- 8L pinout with Reset "This is the difference between 2021A and 2025
- Note 1

MCP2050

Transceiver + Vreg + WWDT

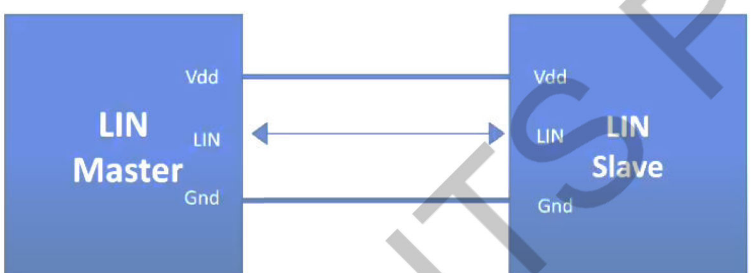


- 3rd Generation LIN transceiver
- 3.3V and 5V 70mA Vreg Options available
- LIN 1.3, 2.X and SAE J2602 compliant
- Resistor Programmable WWDT
- Vbat output ratio pin
- E-Temp
- SOIC/PDIP 14L, QFN 20L packages
- Note 1

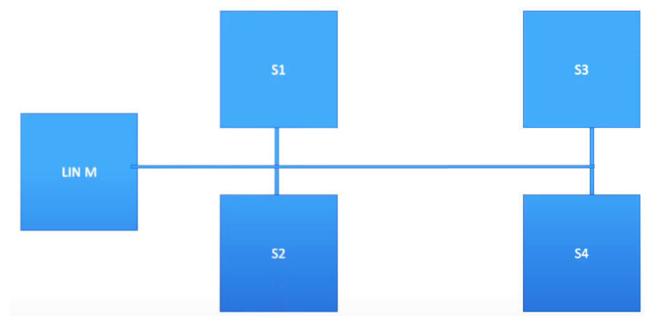
Note 1: Meets OEM Hardware Requirements for LIN, CAN and FlexRay Interfaces in Automotive Applications", Version 1.3, 2012

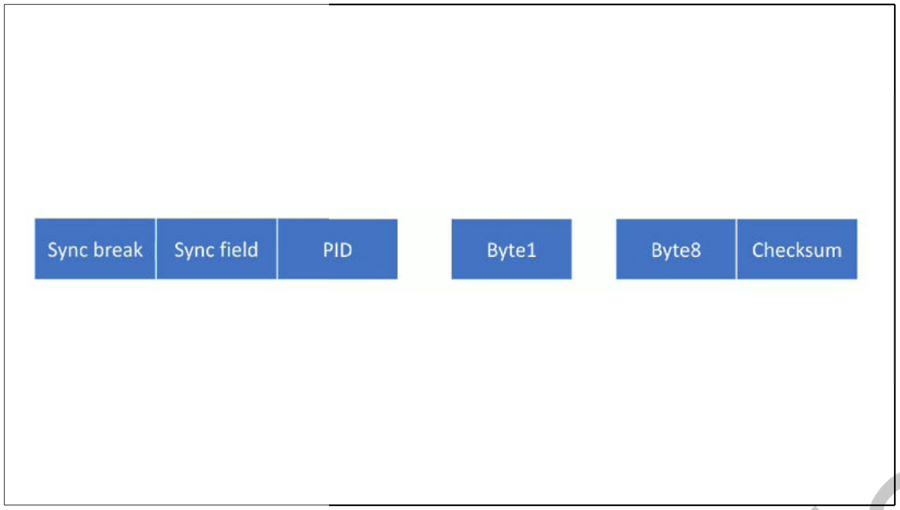
What Do You Need for LIN?

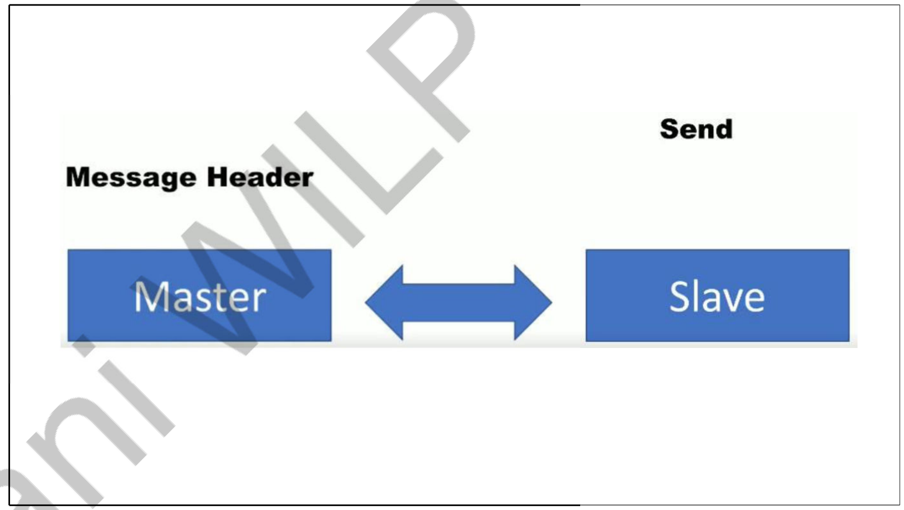
- **neoVI Fire2 / ION / Plasma / ValueCAN 4-2EL**
- **Vehicle Spy** provides full LIN support.
- **Databases --** LDF Database

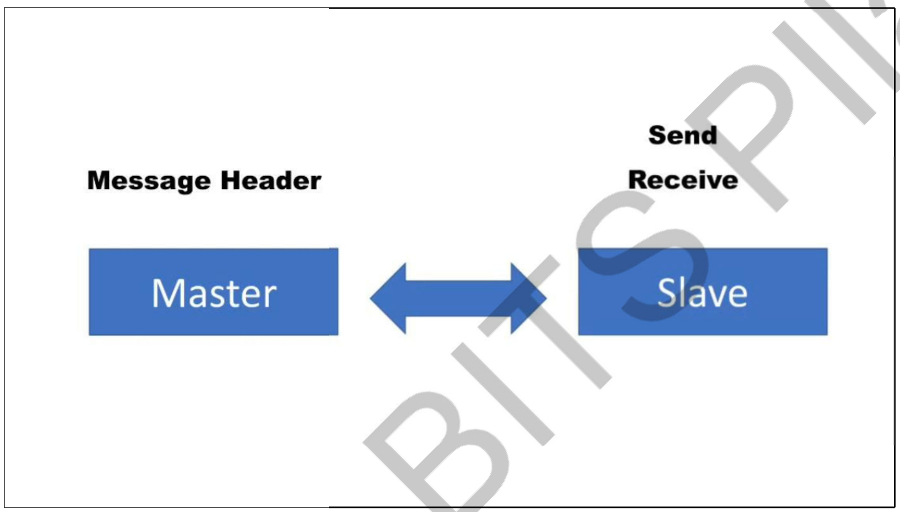


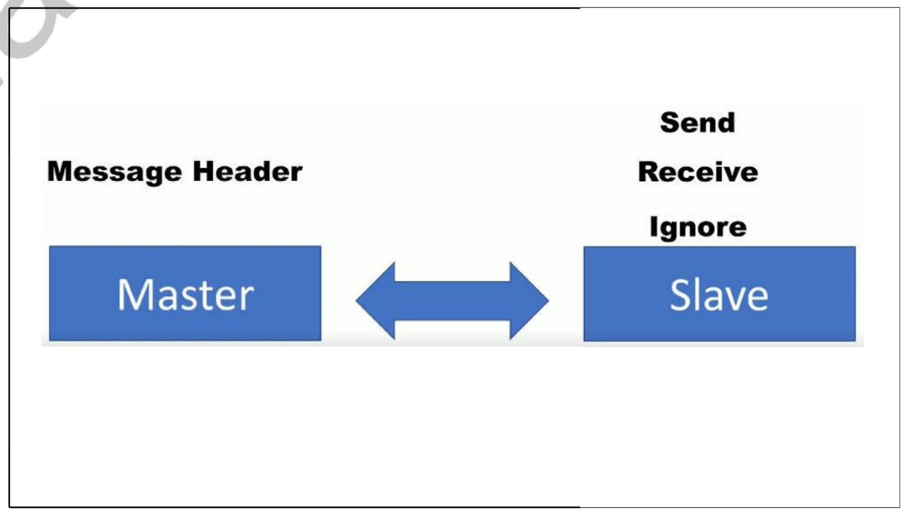
MULTI SLAVE - CLUSTER




Sync break
Sync field
PID
Byte1
Byte8
Checksum

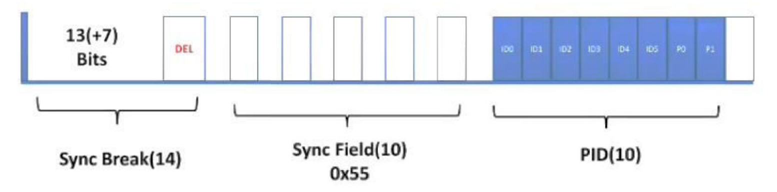

Message Header
Send
Master
Slave


Message Header
Send
Receive
Master
Slave


Message Header
Send
Receive
Ignore
Master
Slave

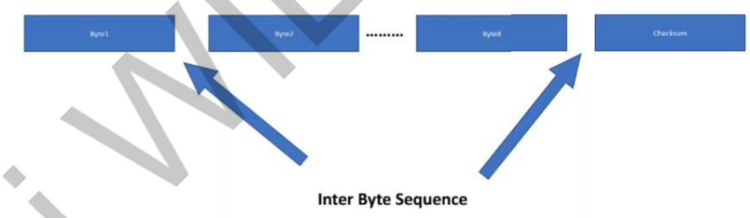
BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

Message Header

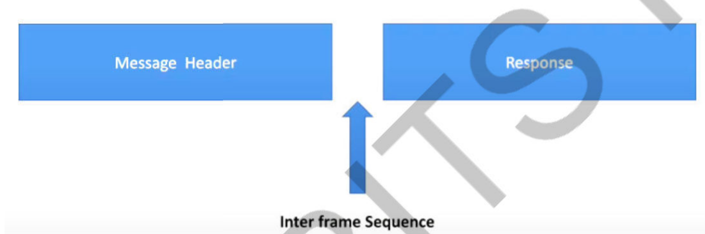


PID – Protective Identifier
P0, P1 – Parity Bit

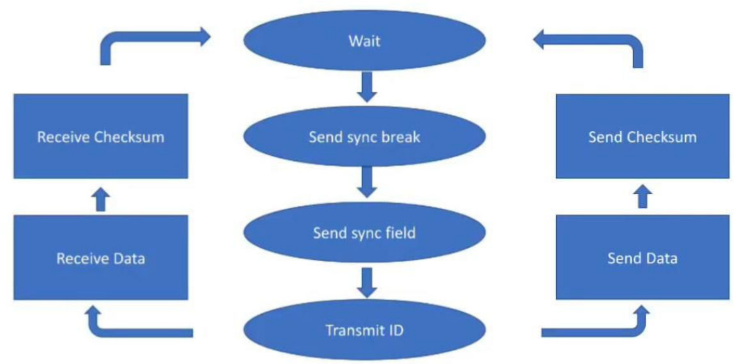
Response



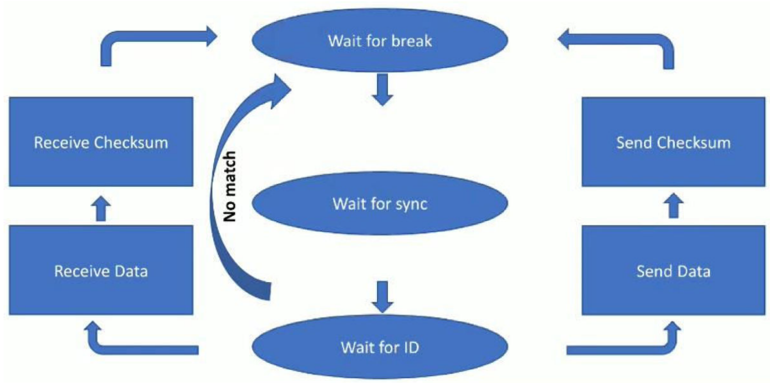
Inter Frame Sequence



LIN MASTER



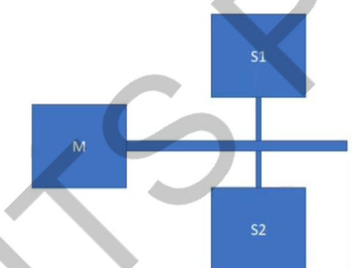
LIN SLAVE



Type of frame	ID	Decimal
Unconditional/Event triggered/Sporadic	00-3B	0-59
Diagnostic	3C/3D	60/61
User defined	3E	62
Reserved	3F	63

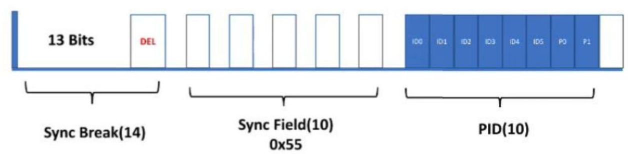
Message Types

Type of frame	ID	Decimal
Unconditional/Event triggered/Sporadic	00-3B	0-59
Diagnostic	3C/3D	60/61
User defined	3E	62
Reserved	3F	63

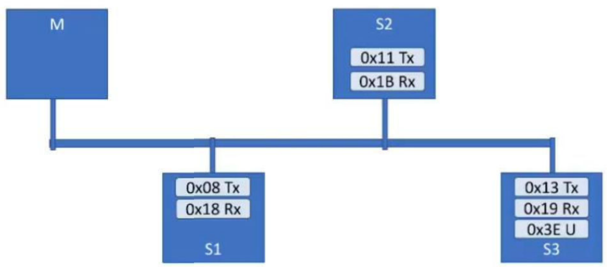


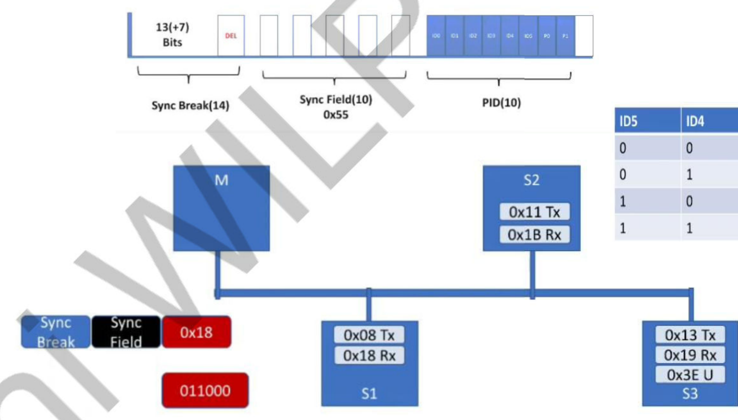
LIN Datarate

ID5	ID4	Bytes
0	0	2
0	1	2
1	0	4
1	1	8



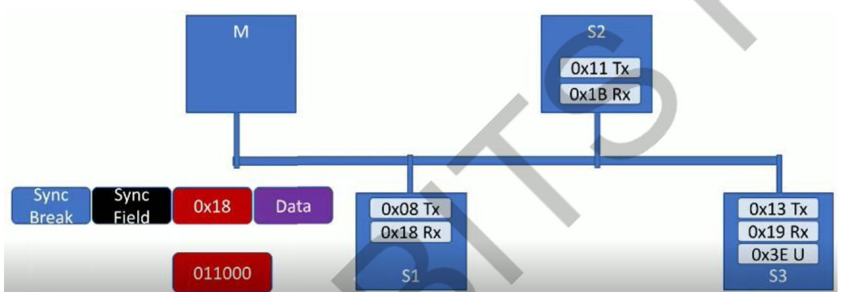
LIN Example





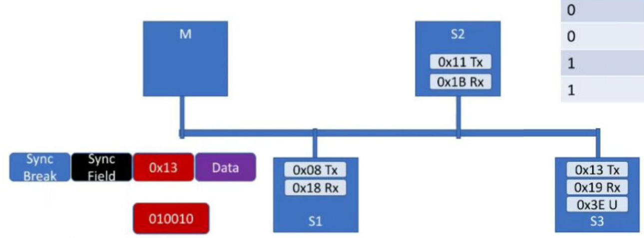
ID5	ID4	Bytes
0	0	2
0	1	2
1	0	4
1	1	8

ID5	ID4	Bytes
0	0	2
0	1	2
1	0	4
1	1	8



LIN Example2

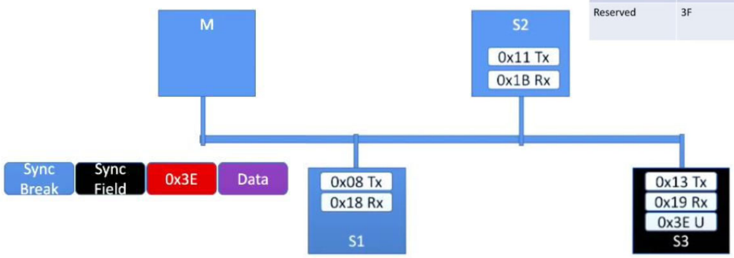
ID5	ID4	Bytes
0	0	2
0	1	2
1	0	4
1	1	8

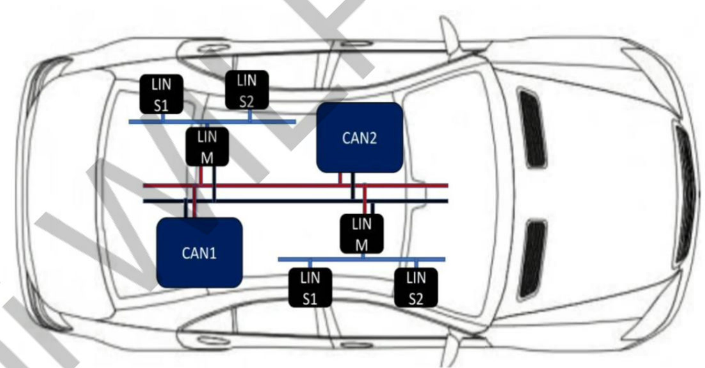


Example 3

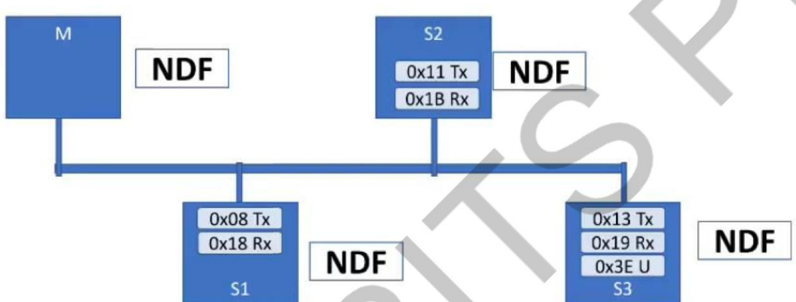
ID5	ID4	Bytes
0	0	2
0	1	2
1	0	4
1	1	8

Type of frame	ID	Decimal
Unconditional/Event triggered/ Sporadic	00-3B	0-59
Diagnostic	3C/3D	60/61
User defined	3E	62
Reserved	3F	63

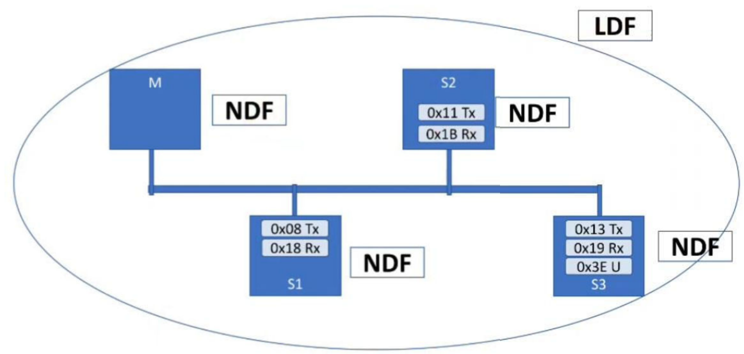




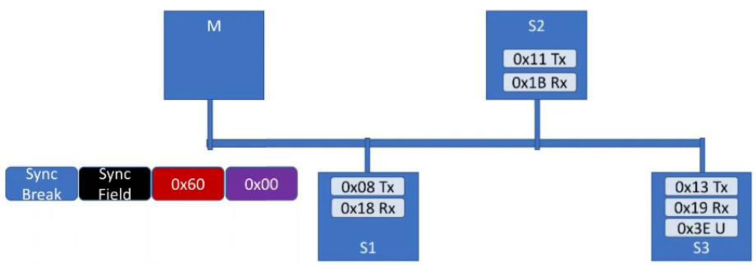
NODE DESCRIPTION FILE (NDF) - LIN DESCRIPTION FILE (LDF)



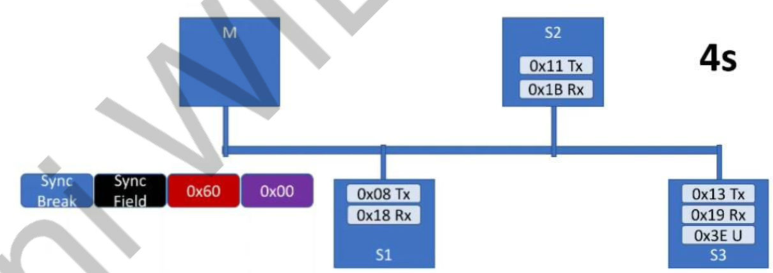
LIN Cluster



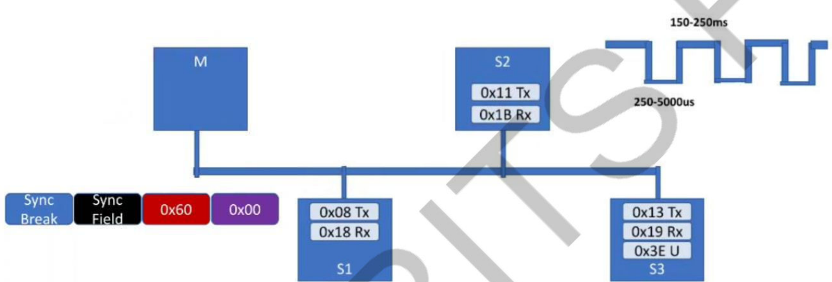
LIN Bus sleep



LIN Bus sleep



LIN Bus wakeup



LIN SCHEDULING