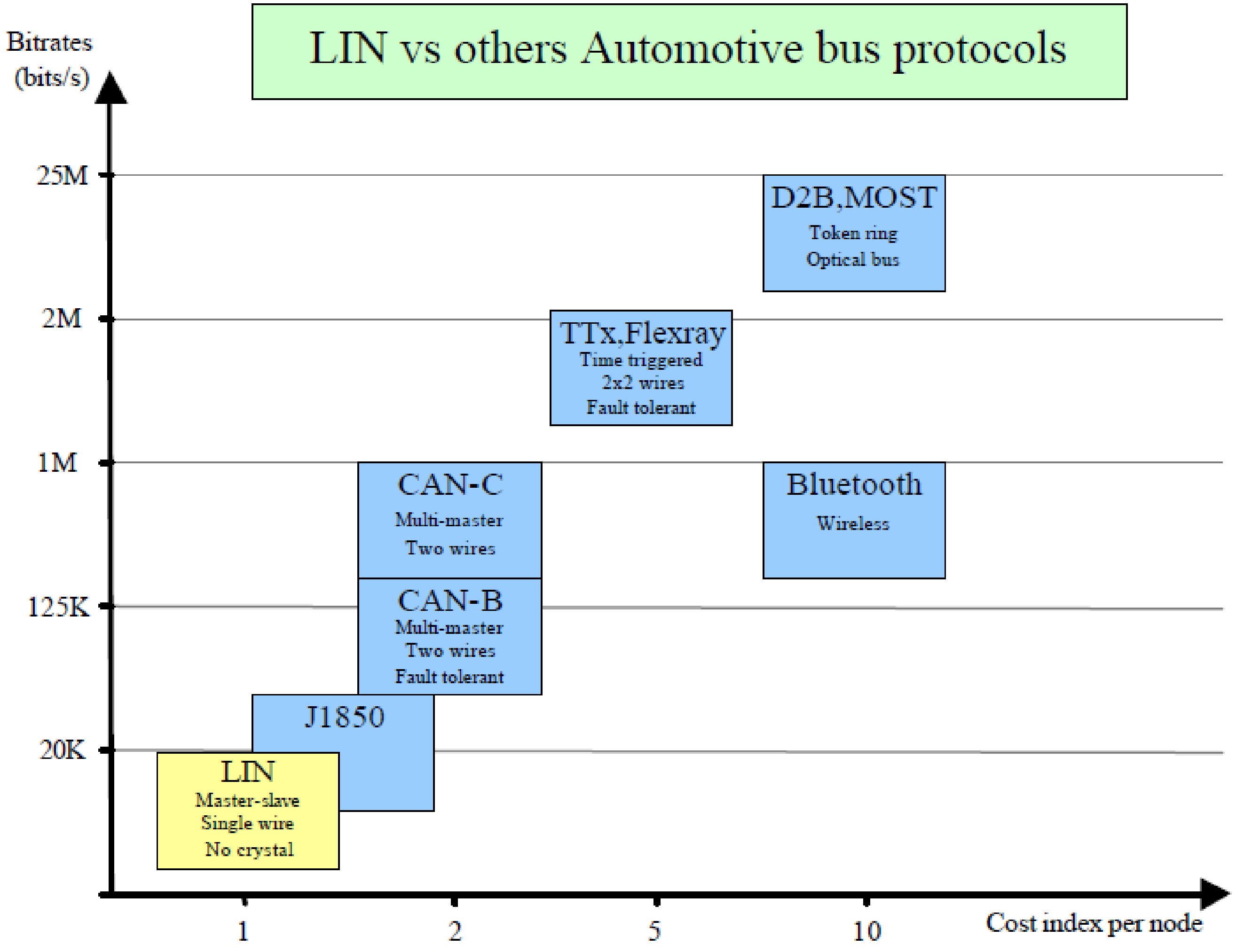
In-Vehicle Networking

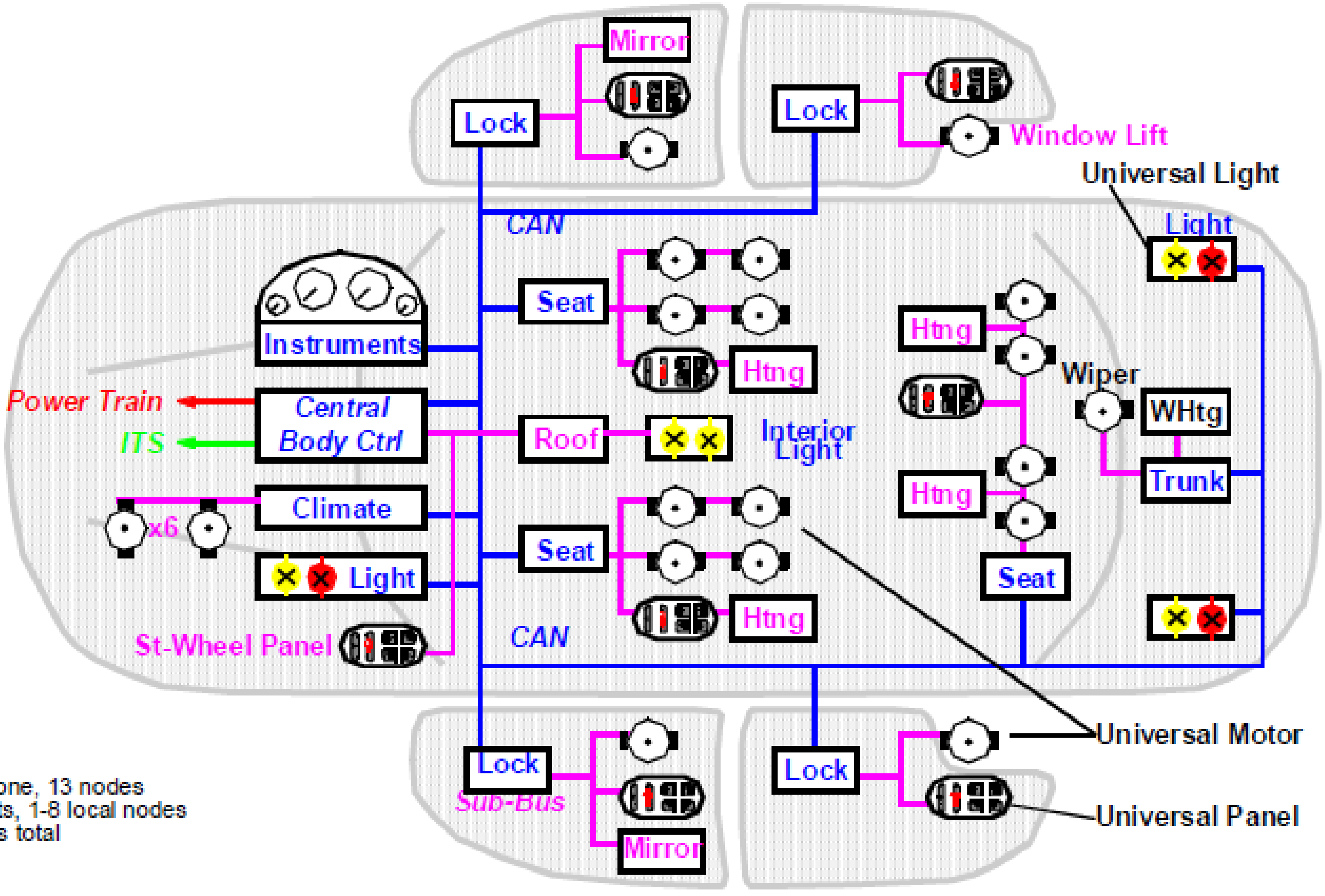
SAE Network classification

- 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.

Costs and Speeds for Automotive Networks



Automotive Body Network



1 backbone, 13 nodes
8 subnets, 1-8 local nodes
52 nodes total

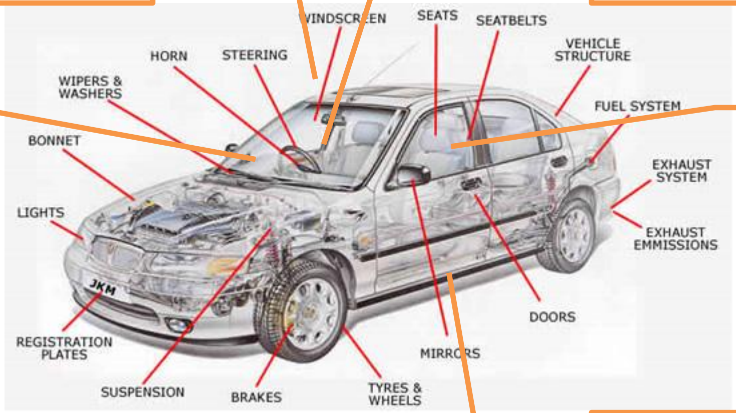
Typical LIN Applications

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

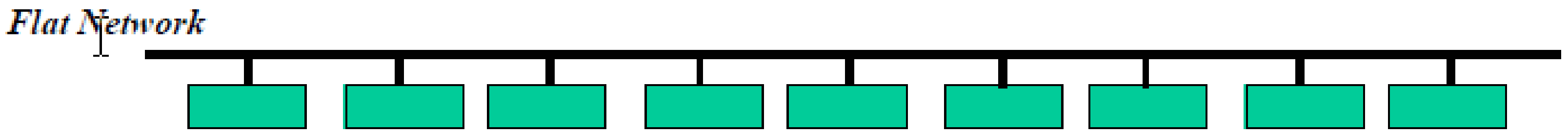
Seat:
Many Seat Position Motors, Occupancy Sensor, Control Panel



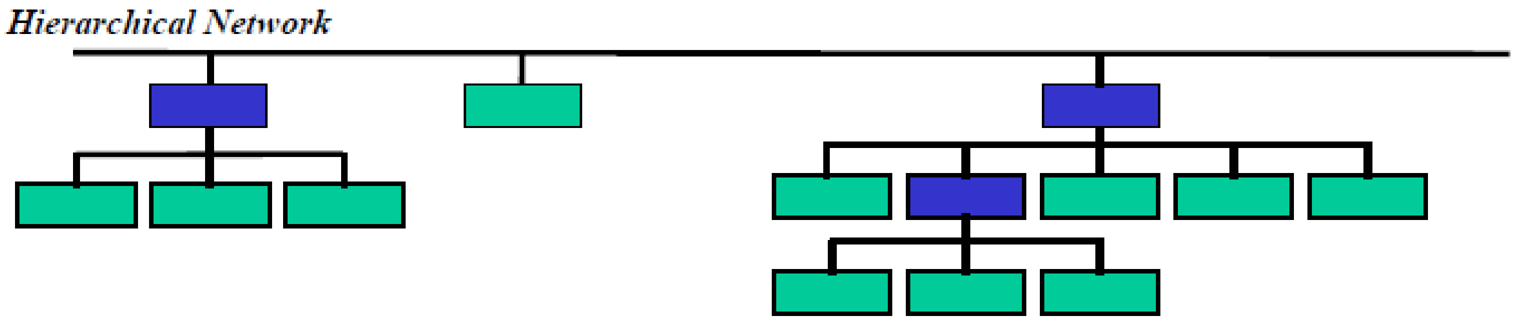
Door/window/seat:
Mirror, Central ECU, Mirror, Switch, Window Lift, Seat Control Switch, Door Lock, etc.

Hierarchical Network Struc

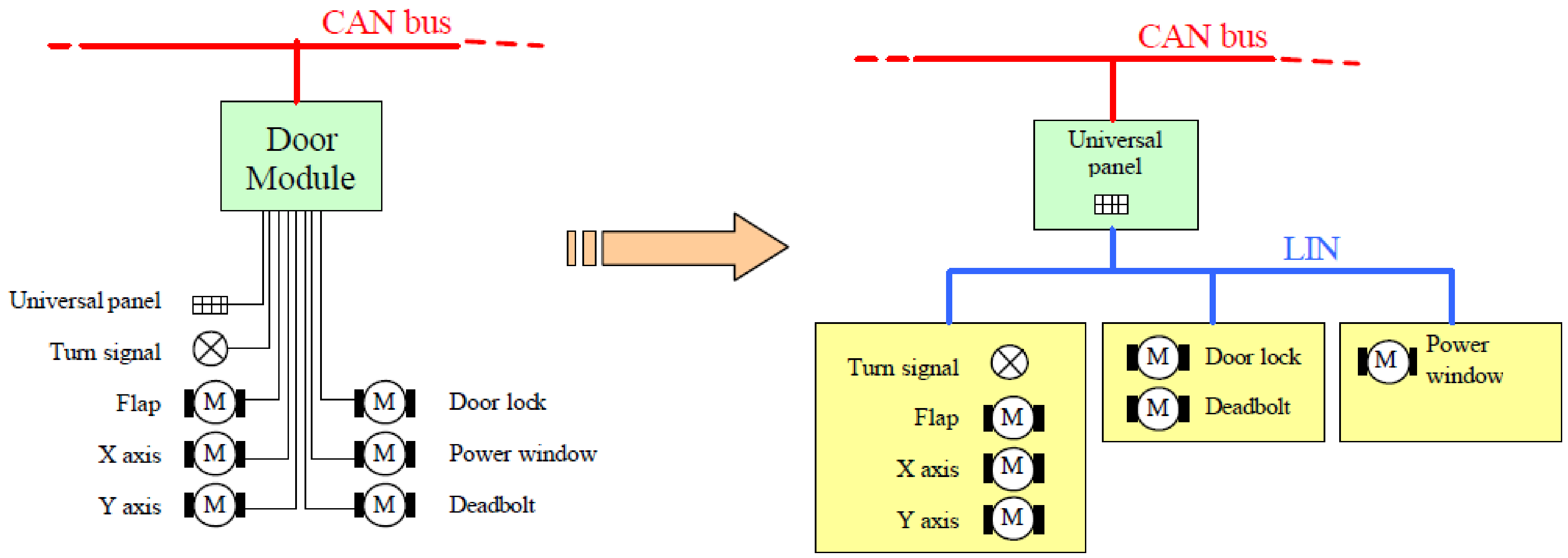
- CAN based network



- CAN + LIN



LIN implementation

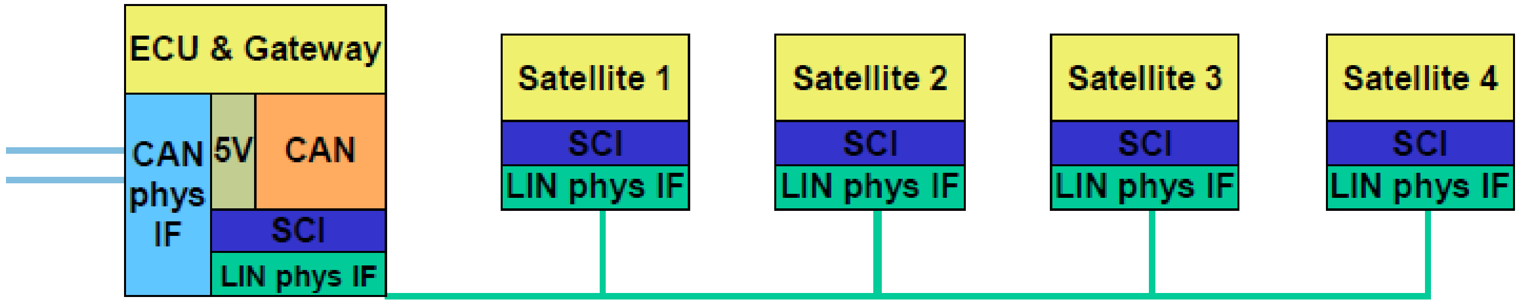


SubNets

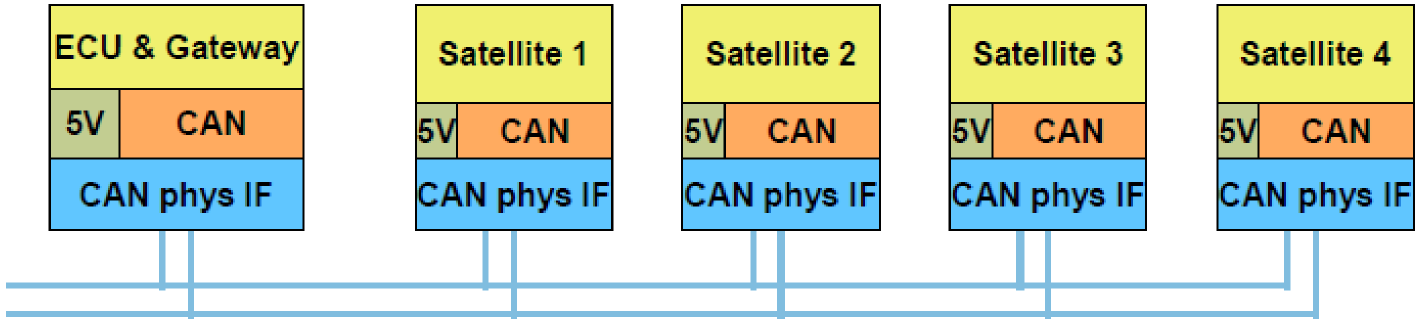
- Necessary to reduce Busload on main Bus
- Solutions
  - CAN
    - Automotive Standard Bus
    - Compatible with Main Bus
    - Expensive (Die Size/ Dual Wire)
  - Serial Sub Bus
    - no standard Bus System
    - not compatible with Main Bus
    - inexpensive
    - SCI-Based: Interface exists even 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

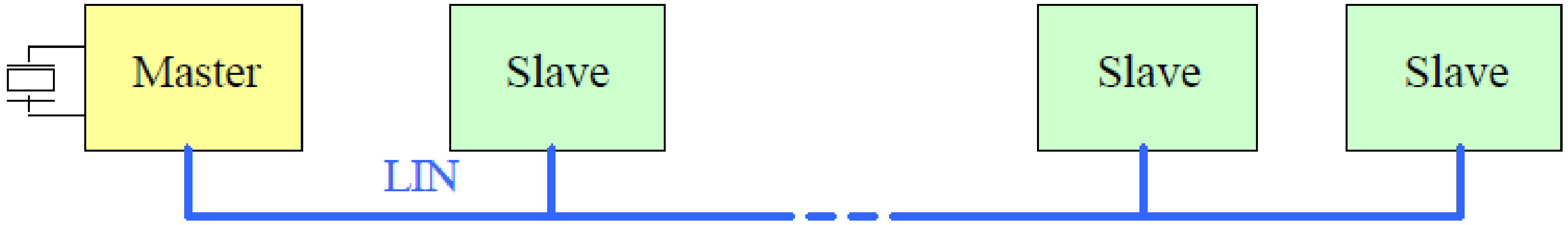
Features	LIN	CAN
Medium access control	single master	multiple masters
Typical bus speed	2.4, 9.6 and 19.2kbps	62,5…1000kbps
Multicast message routing	6 bit identifier	11/29 bit identifier
Typical size of network	2…16 nodes	4…20 nodes
Data byte per frame	0… 8	2…8
Transmission time for 4 data bytes	6 ms 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)

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

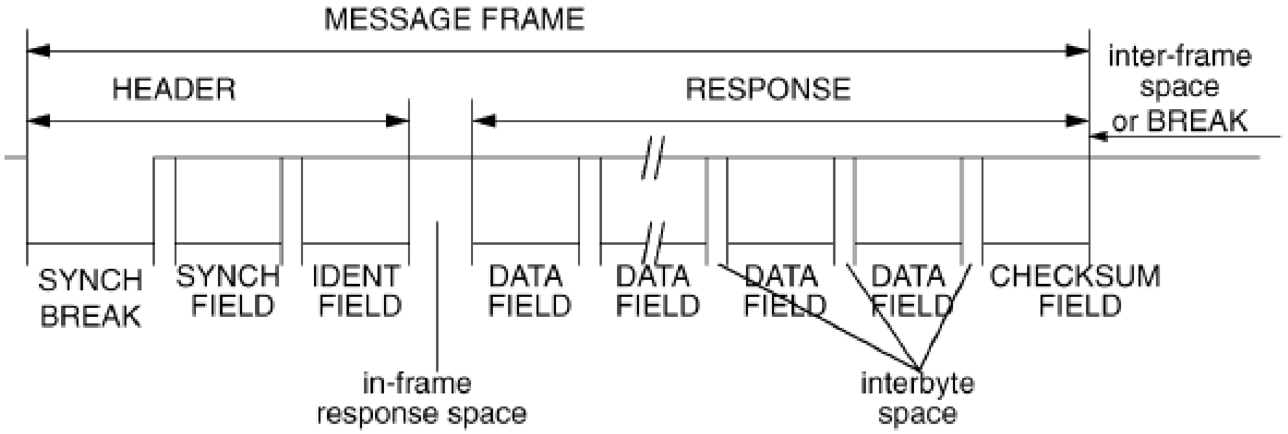
- Is one of 2-16 Members on the Bus and receives or transmits Data when an appropriate ID is sent by the master.
  - Slave snoops for ID.
  - According to ID, slave determines what to do.
- either receive data
- or transmit data
- or do nothing.
  - When transmitting the slave
- sends 1, 2, 4, or 8 Data 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.

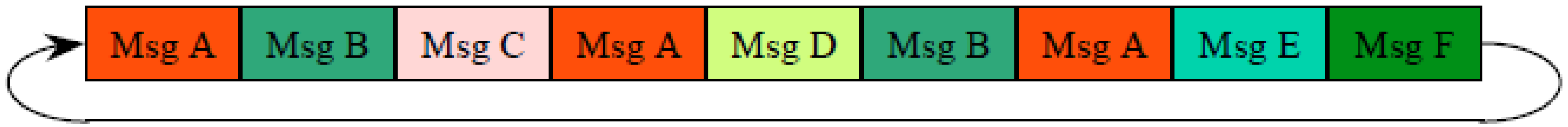
LIN protocol offers message timing predictability

- Time Triggered Approach
- Message Length is known
  - Number of transmitted data bytes is known
    - Minimum length can be calculated
  - Each Message has length budget of 140% of it's minimum length
    - maximum allowed length is known
    - distance between beginning of two messages

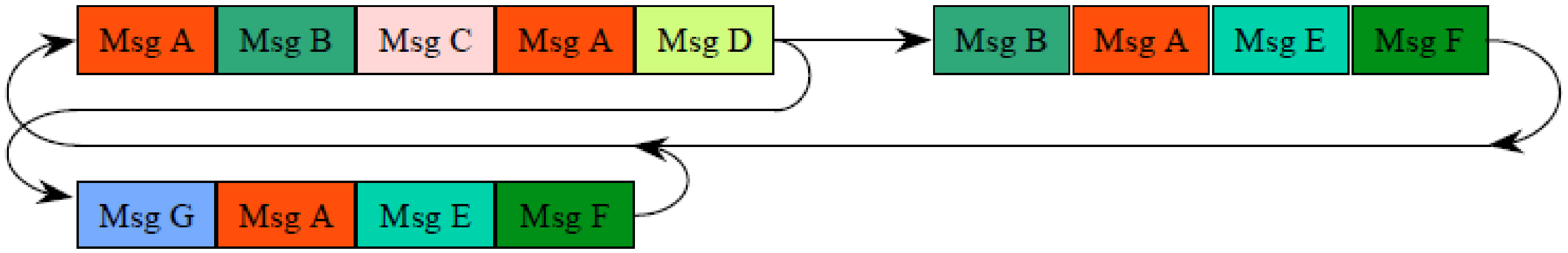


LIN protocol offers message timing predictability

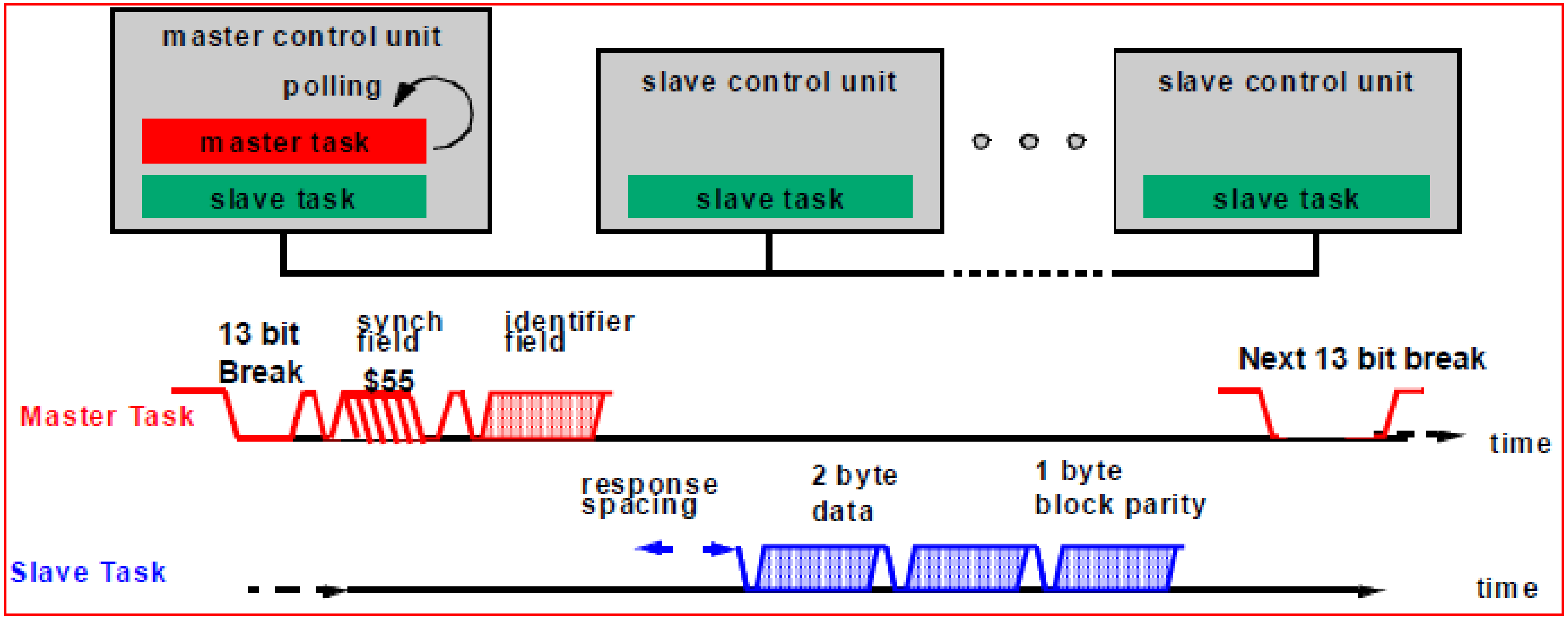
- Message sequence is known
  - Master uses scheduling table



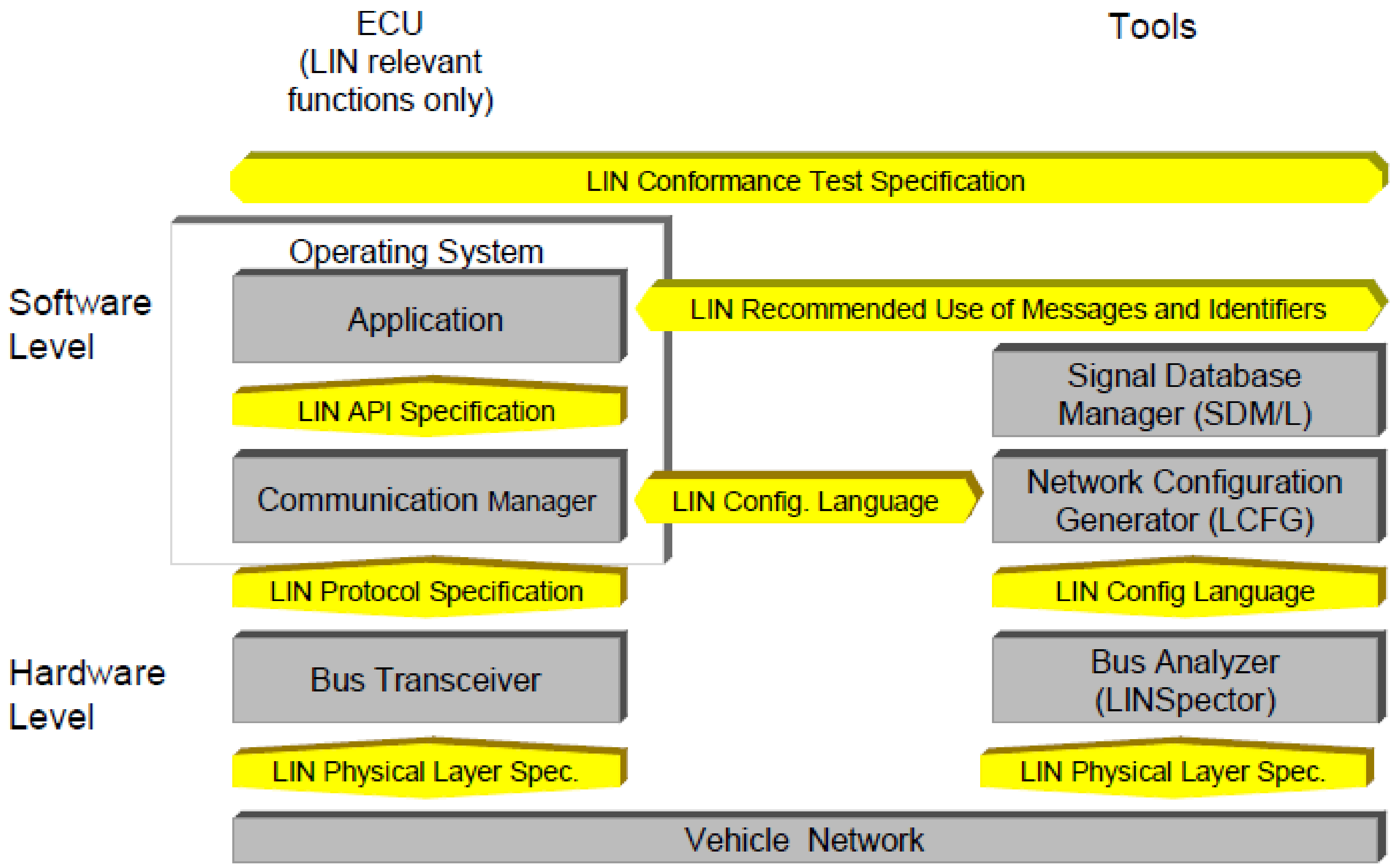
- Use of different scheduling tables is possible
- Provides Flexibility



Data Transmission



LIN Standard - Overview

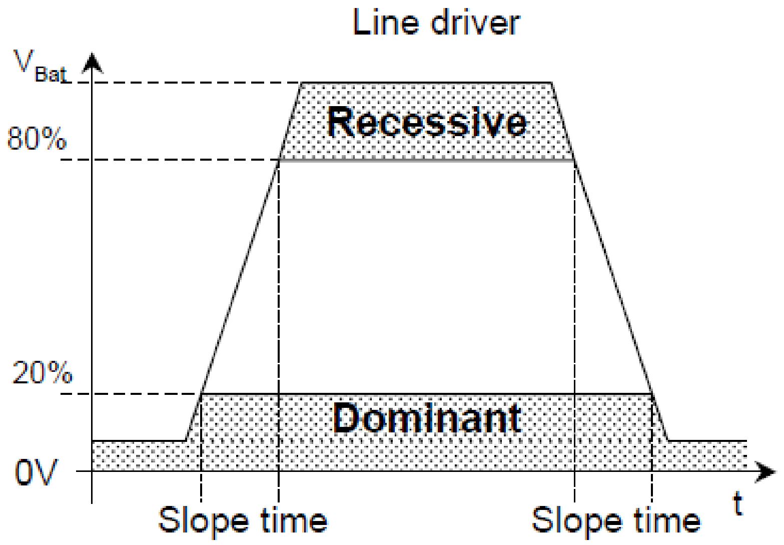

ECU
(LIN relevant functions only)
Tools
LIN Conformance Test Specification
Software Level
Operating System
Application
LIN Recommended Use of Messages and Identifiers
Signal Database Manager (SDM/L)
LIN API Specification
Communication Manager
LIN Config. Language
Network Configuration Generator (LCFG)
LIN Protocol Specification
LIN Config Language
Hardware Level
Bus Transceiver
Bus Analyzer (LINSpector)
LIN Physical Layer Spec.
LIN Physical Layer Spec.
Vehicle Network

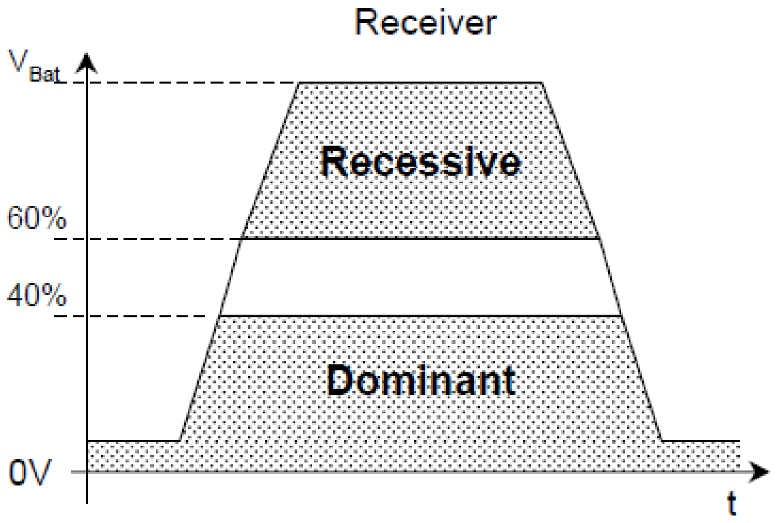
LIN (Local Interconnect Network)

- Low speed serial network designed for body control in automotive applications
- Physical layer based on ISO 9141 (the K-line).
  - Single wire plus ground
  - Voltage level signaling
  - Dominant/recessive bit levels
  - Max 40 m wire length
  - 1-20 kbit/s
- Data Link
  - Media Access Control: Master/slave
  - Serial asynchronous byte oriented communication
- UART compatible
  - 64 uniquely identified messages
- Transport layer for diagnostic support
- Includes application layer development framework

LIN Physical Layer

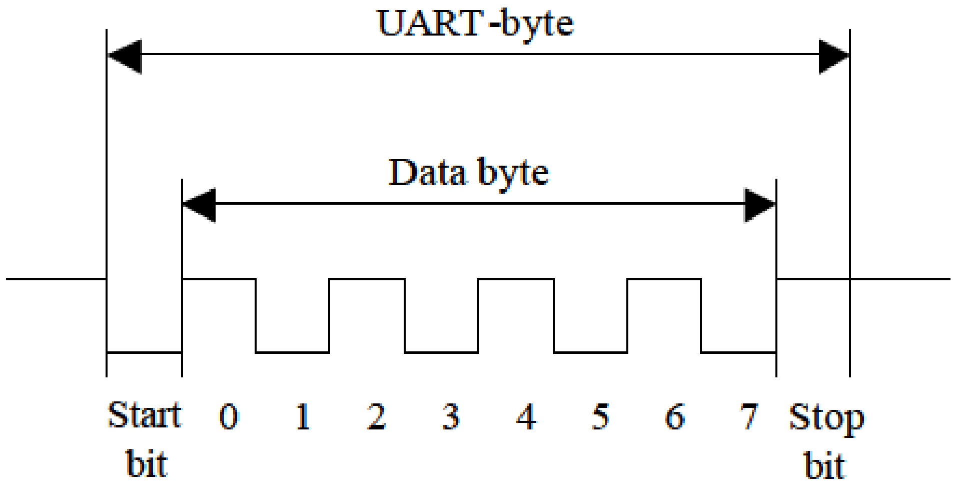
- Single Wire
- ISO 9141 Compliant





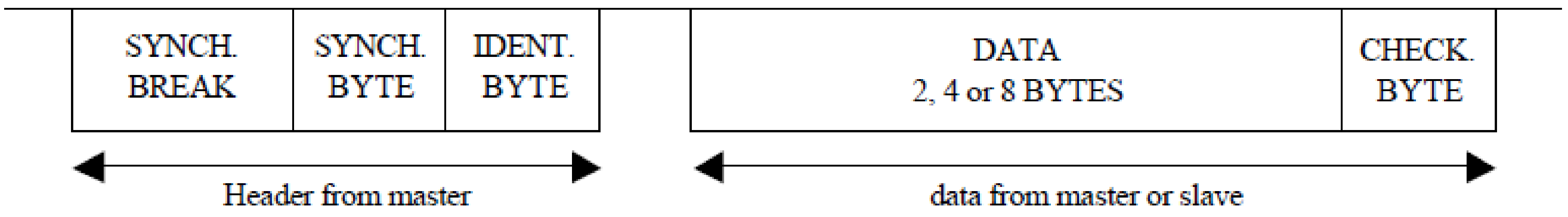
Technical overview

- a single master and up to 16 slaves
- no arbitration
- deterministic traffic behavior and guarantees the latency times
- single wire
- Maximum data rate 20kbit
- byte-encoded according to the UART-protocol



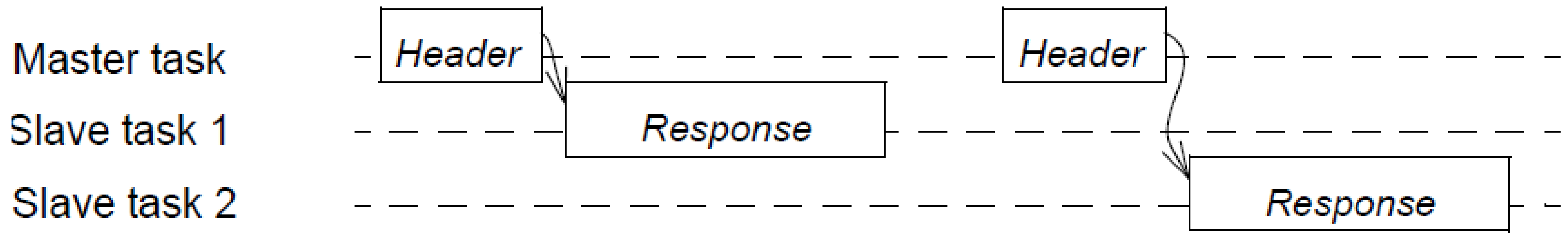
LIN frame

- Synchronization break,
- Synchronization byte,
- Identifier byte,
- Data bytes,
- Ckecksum byte.



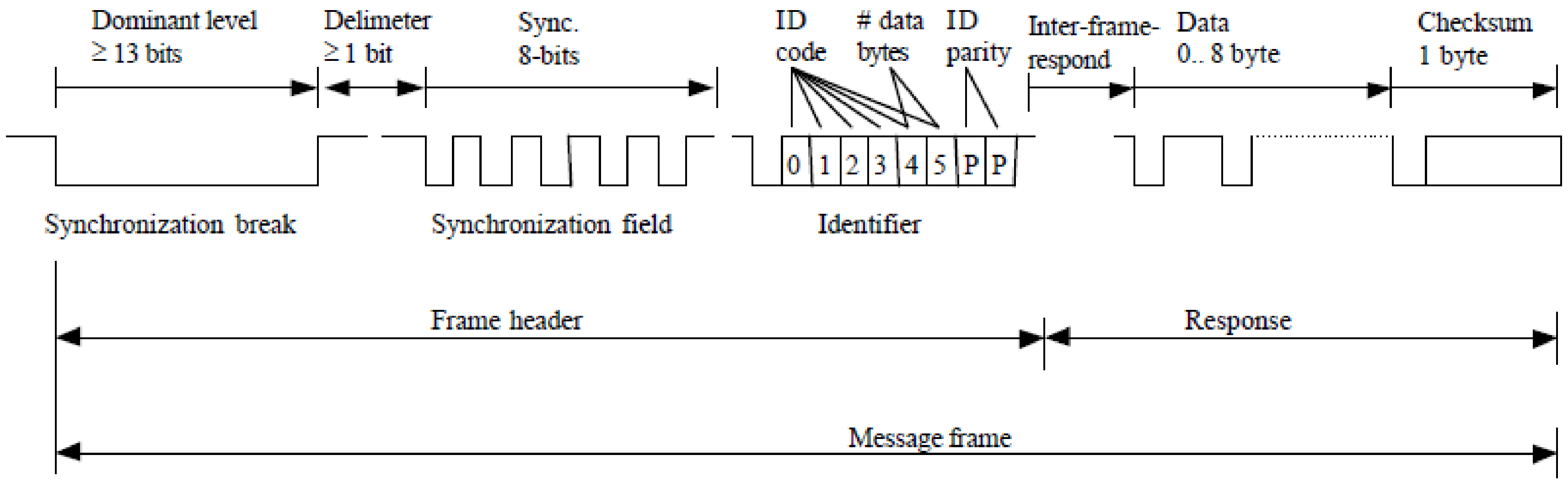
LIN frame

- A frame consists of a header (provided by the master task) and a response (provided by a slave task).



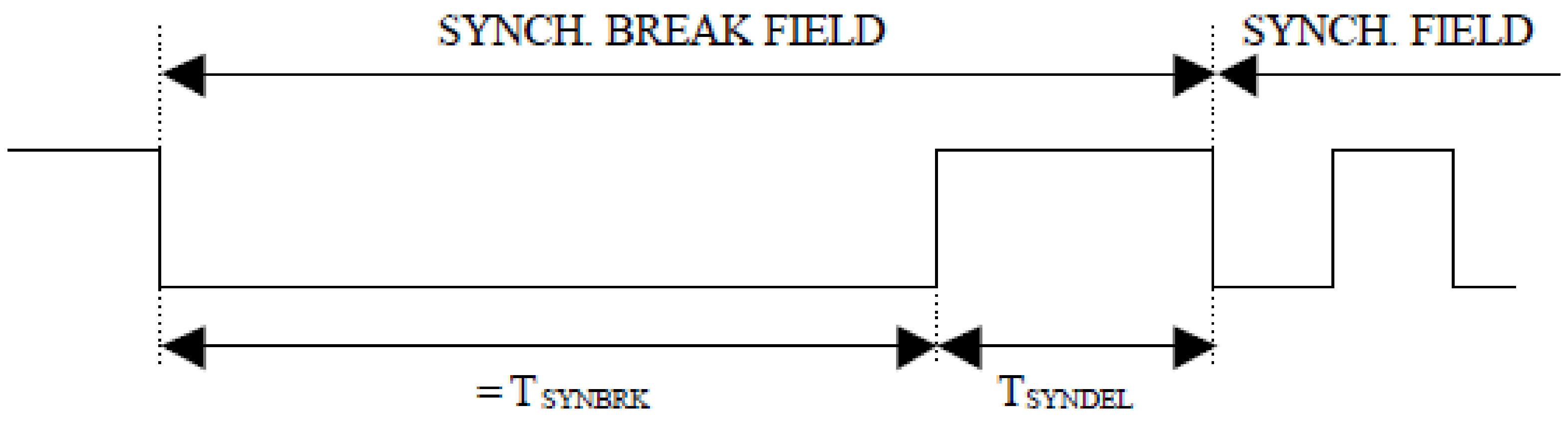
LIN message frame

- LIN message frame consists of a header and a response part.
- header: the SYNC-break, SYNC-field and the identifier- (ID) field



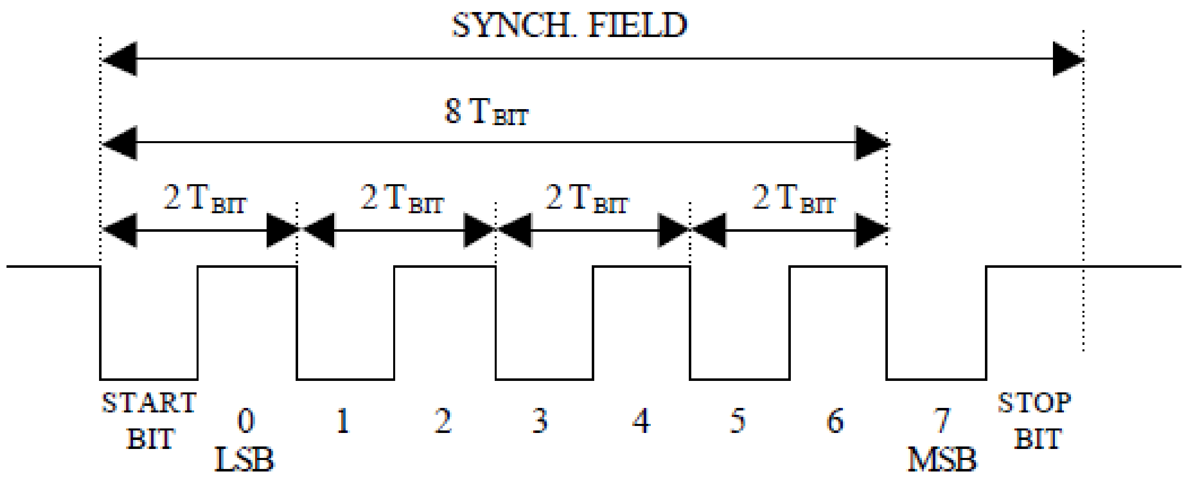
Synchronization break

- Marks the Beginning of a Message Frame
- consists of at least 13 bits of zeroes.
- the slaves are allowed to have a baud rate that differs with 15% to the masters
- Slave: to detect that a message is transmitted on the bus.
- For the master node, implemented in a microcontroller, the procedure of sending a SYNC-break involves some tampering with the UART-protocol.



Synchronization field

- Specific Pattern for Determination of Time Base
- master send a header, slaves synchronize their clocks every time a new message is received.
- Master: accurate resonator as a time reference.
- Slaves: synchronizes its clock from the falling edge of the start bit to the bit 7 of the SYNC-byte and divides it by 8.



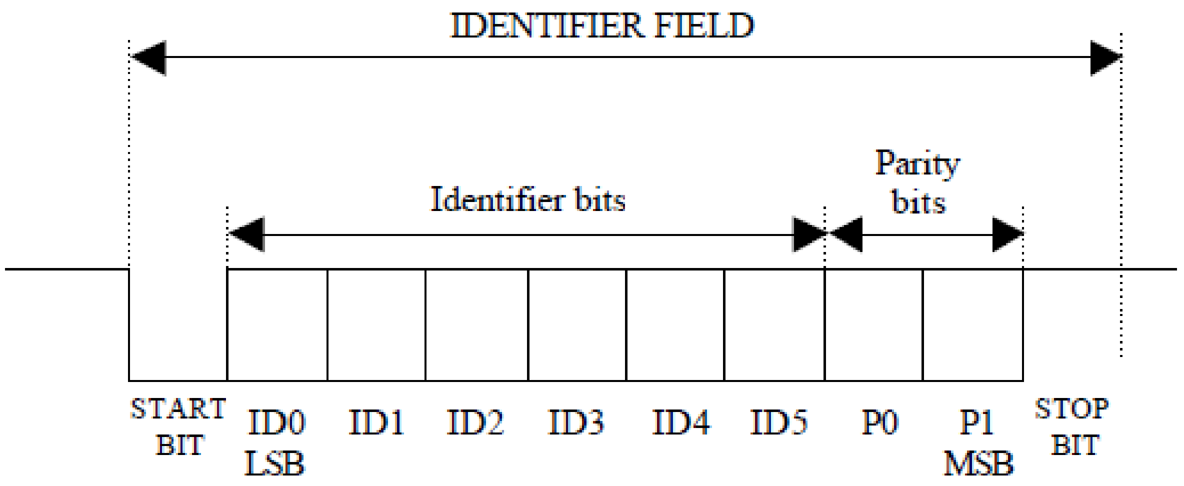
Identifier field

- The slave nodes on the network are addressed by the ID- field.
- Message Identifier: Incorporates Information about the sender, the receiver(s), the purpose, and the Data field length.
  - 2 Parity Bits protect the highly sensitive ID-Field.
  - Length 6 Bits, 64 Message Identifiers are possible.
  - 4 classes of 1/2/4/8 Data Bytes. The length coding is in the 2 LSB of the ID-Field. Each class has 16 Identifiers.

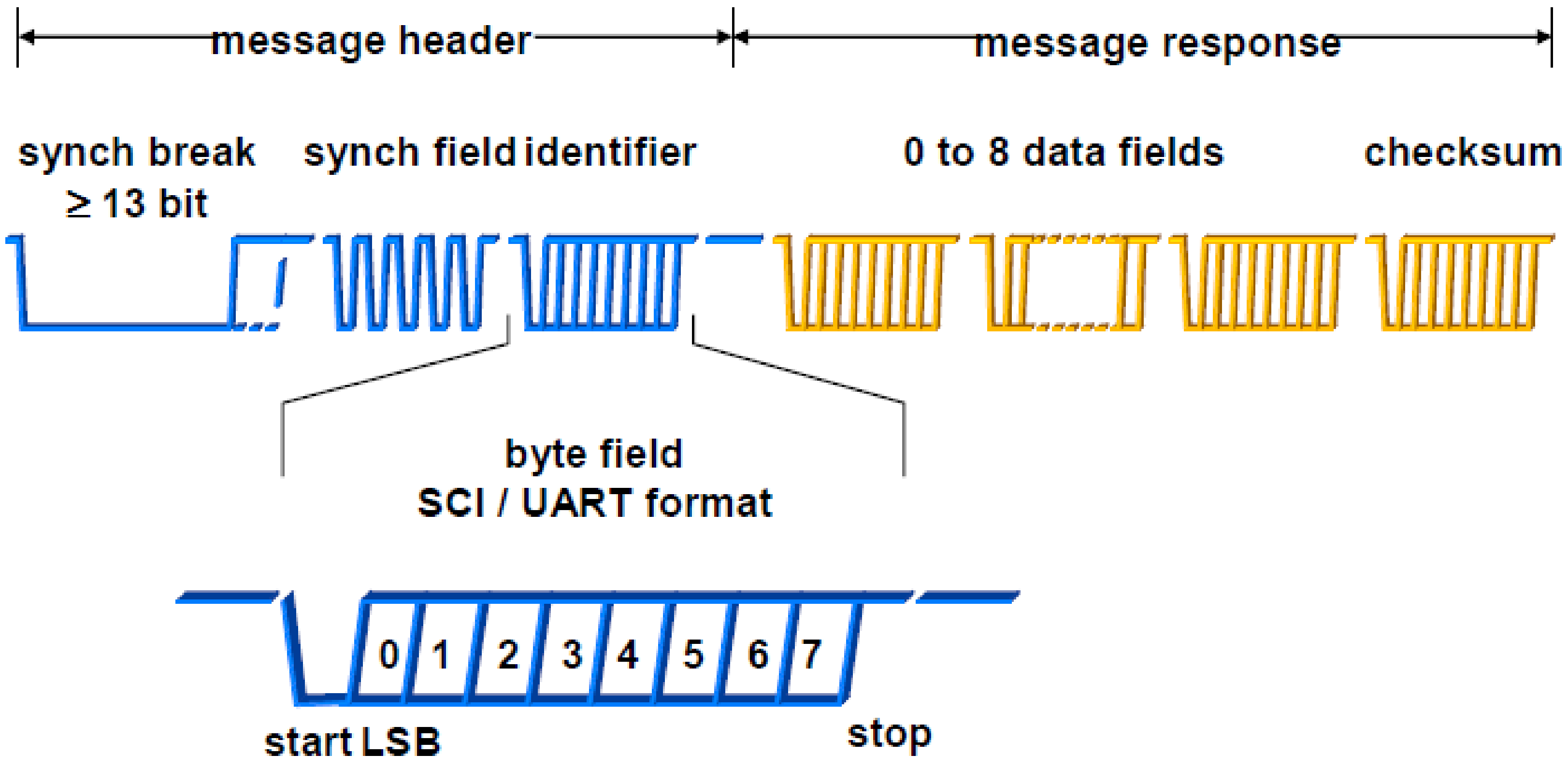
ID range	Frame Length
00-31 0x00-0x1F 000000-011111	2
32-47 0x20-0x2F 100000-201111	4
48-63 0x30-0x3F 110000-111111	8

Identifier

- The identifier field is sent by the master node to all LIN nodes
- This identifier normally contains one of 64 different values and includes 2 parity bits in the 8 bit data
- The identifier is normally associated with a collection of signals that are subsequently transmitted on the LIN bus
- In a specific case this can initiate SLEEP mode in the LIN slave nodes – in this case no further data is transmitted on the LIN bus

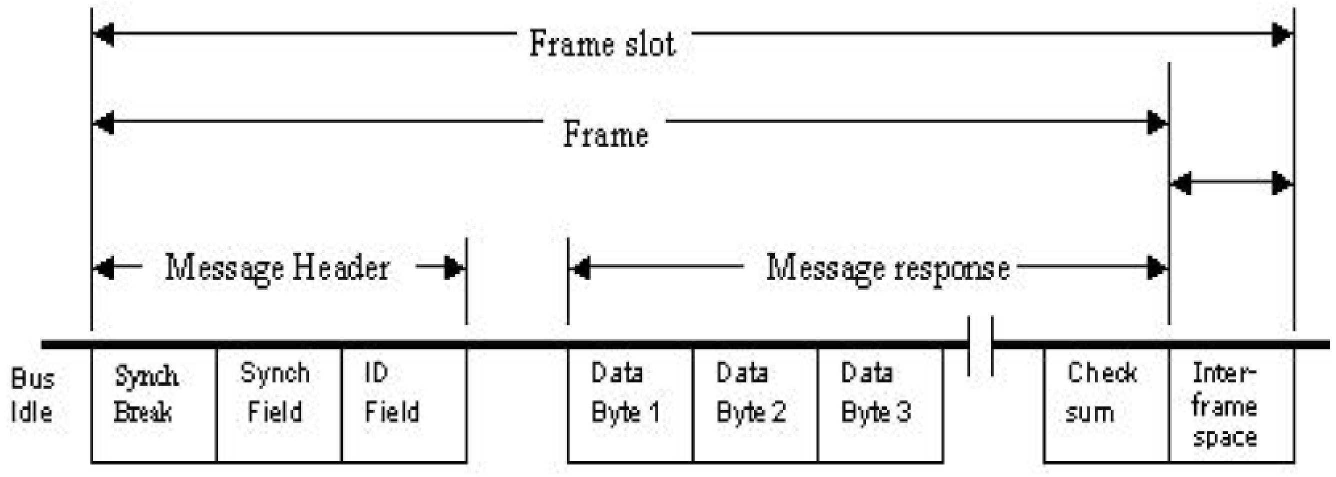


LIN Message Frame



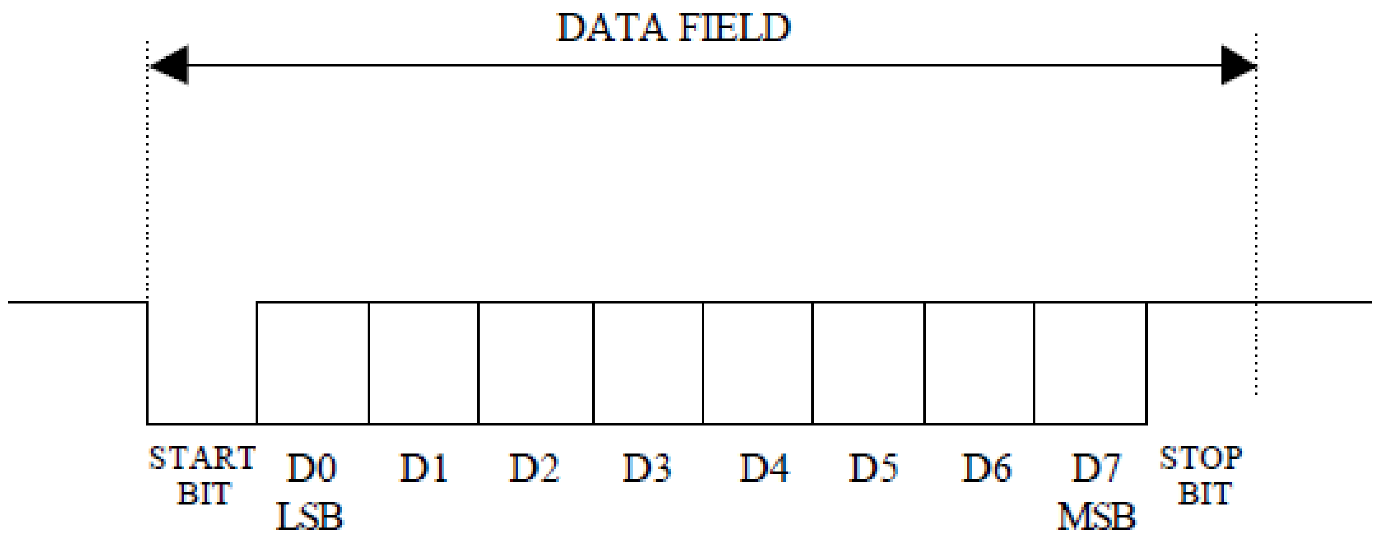
response

- The response contains one to eight data bytes and one checksum byte.
- The slave task is connected to the identifier and receives the response, verifies the checksum and uses the data transport.
- Messages are created when the master node sends a frame containing a header.
- The slave node(s) then fills the frame with data depending on the header sent from the master.



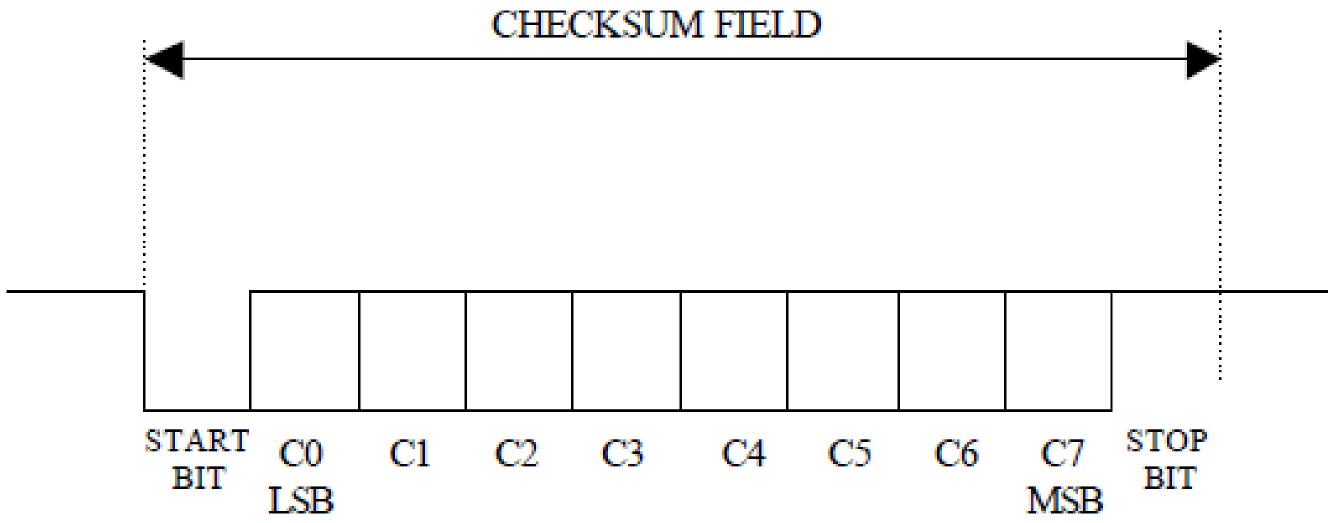
Data byte

- The data length is defined by ID5 and ID6 from the identifier field.
- It can be 2, 4 or 8bytes.



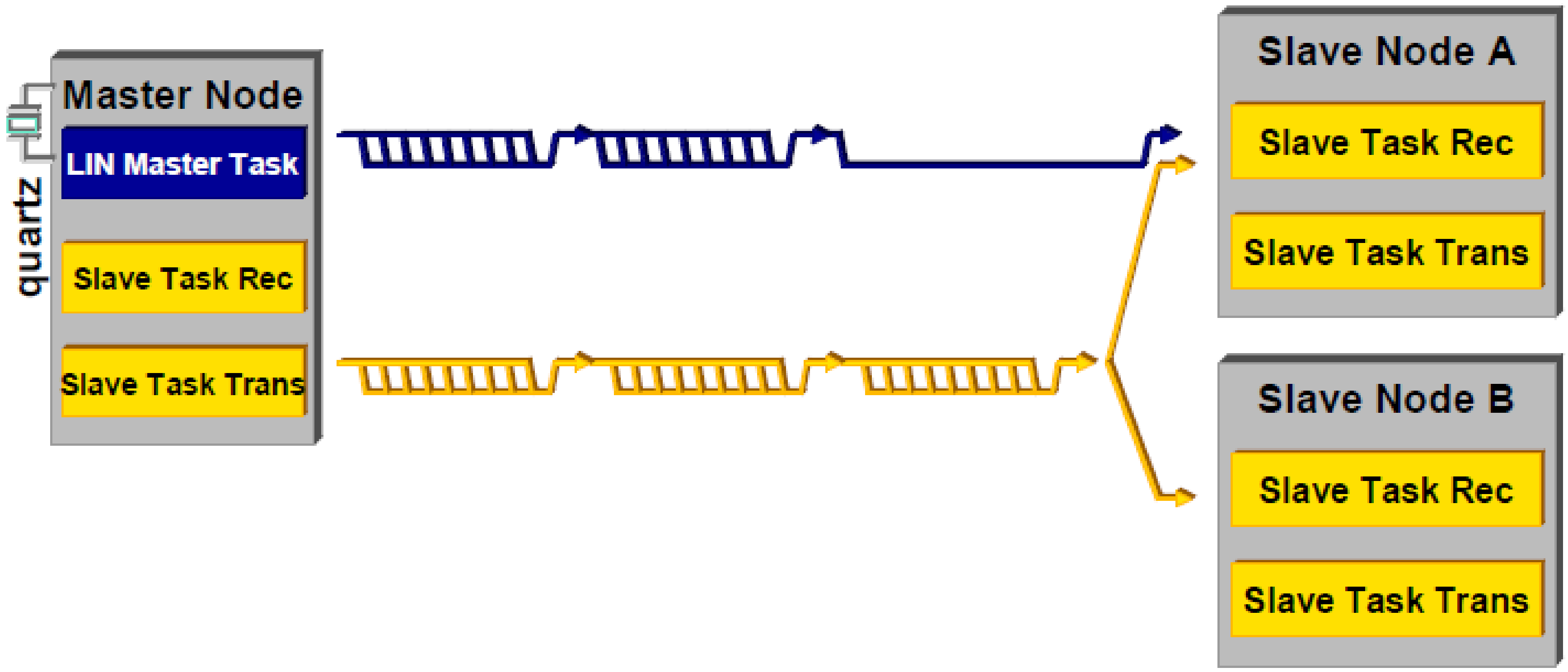
Cheksum byte

- The checksum (CRC) is computed only on the data field.
- All other fields are not included.



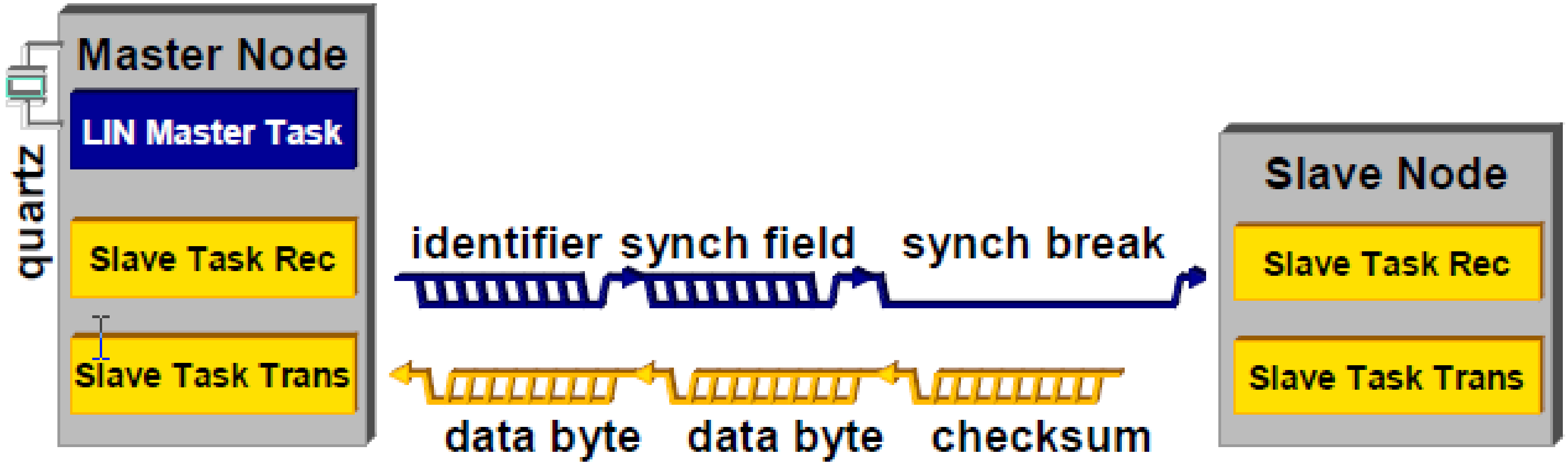
LIN Communication – Data from Master to Slave(s)

- a command frame (CMD frame).
- master sends a complete message frame for one ore more slaves



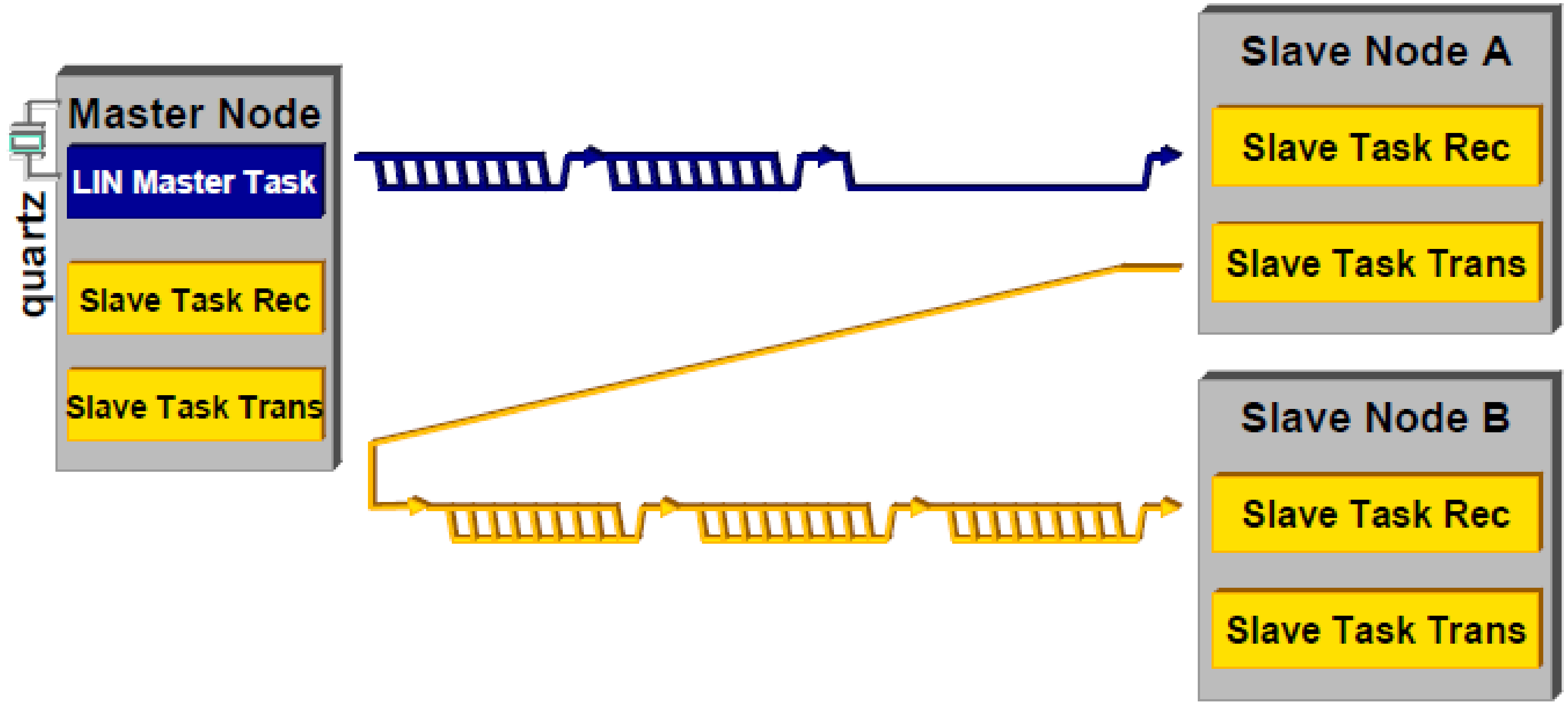
LIN Communication – Data from Slave to Master

- request frames (REQ frame).
- master requests a response from a specific slave (polling)



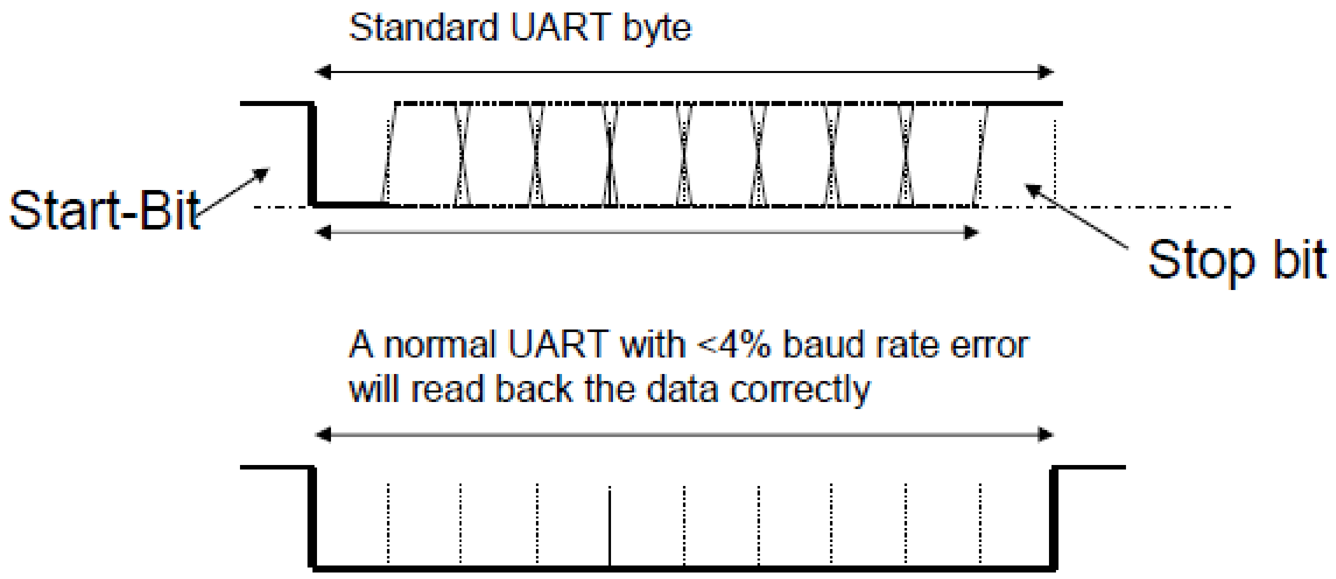
LIN Communication – Data from Slave to Slave

- information sent between sensor and actuator.
- one slave sends its response to one or more slaves



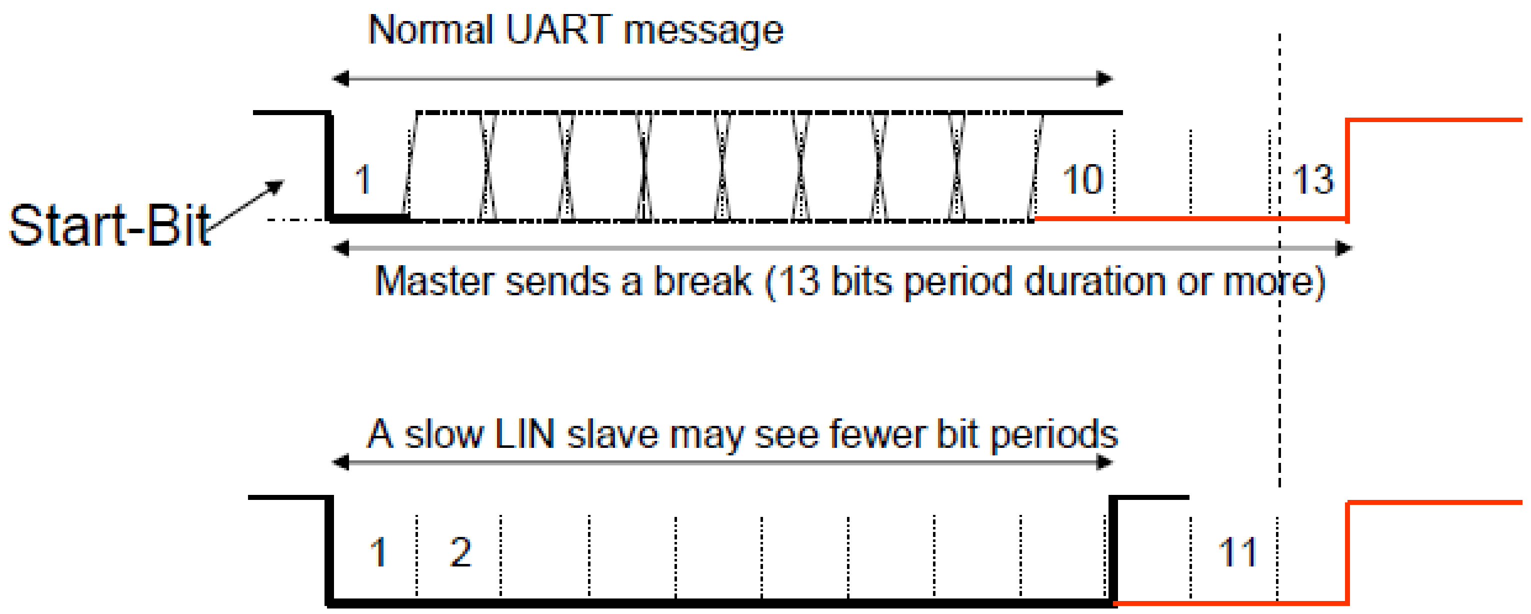
Frame Synchronisation - UART

- Initial conditions: +/- 4% baud rate accuracy relative the transmitting source
- A standard transmission of data will require matched send and receiver baud rates


Standard UART byte
Start-Bit
Stop bit
A normal UART with <4% baud rate error will read back the data correctly

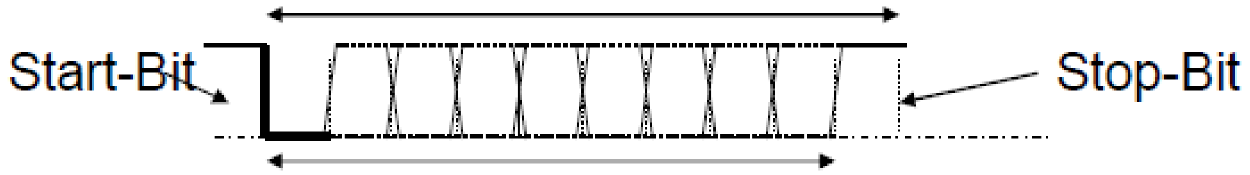
Frame Synchronisation - LIN

- Initial conditions: +/- 15% baud rate accuracy relative the the LIN master transmitting the synchronisation frame
- A synch break must be at least 13 bit periods in duration to allow for

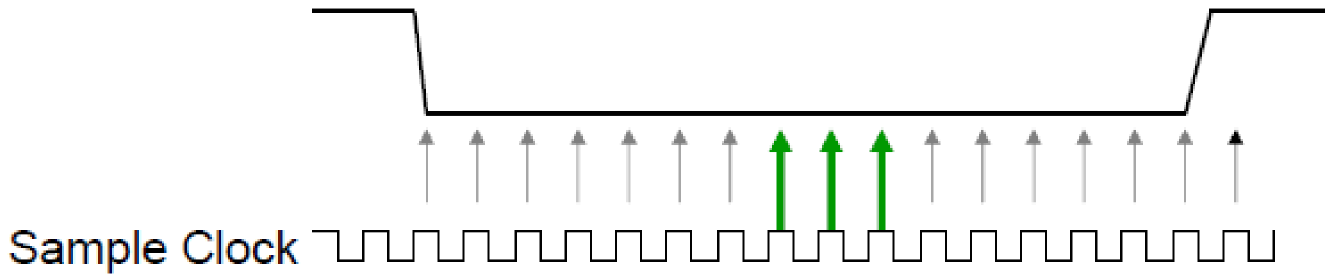

Normal UART message
Start-Bit
1
10
13
Master sends a break (13 bits period duration or more)
A slow LIN slave may see fewer bit periods
1
2
11

Bit-Synchronisation

- A start bit transition to a low logic level (dominant) indicates a start of a byte, least significiant first and completing with a logic high level (resesessive) bit to indicate the STOP bit

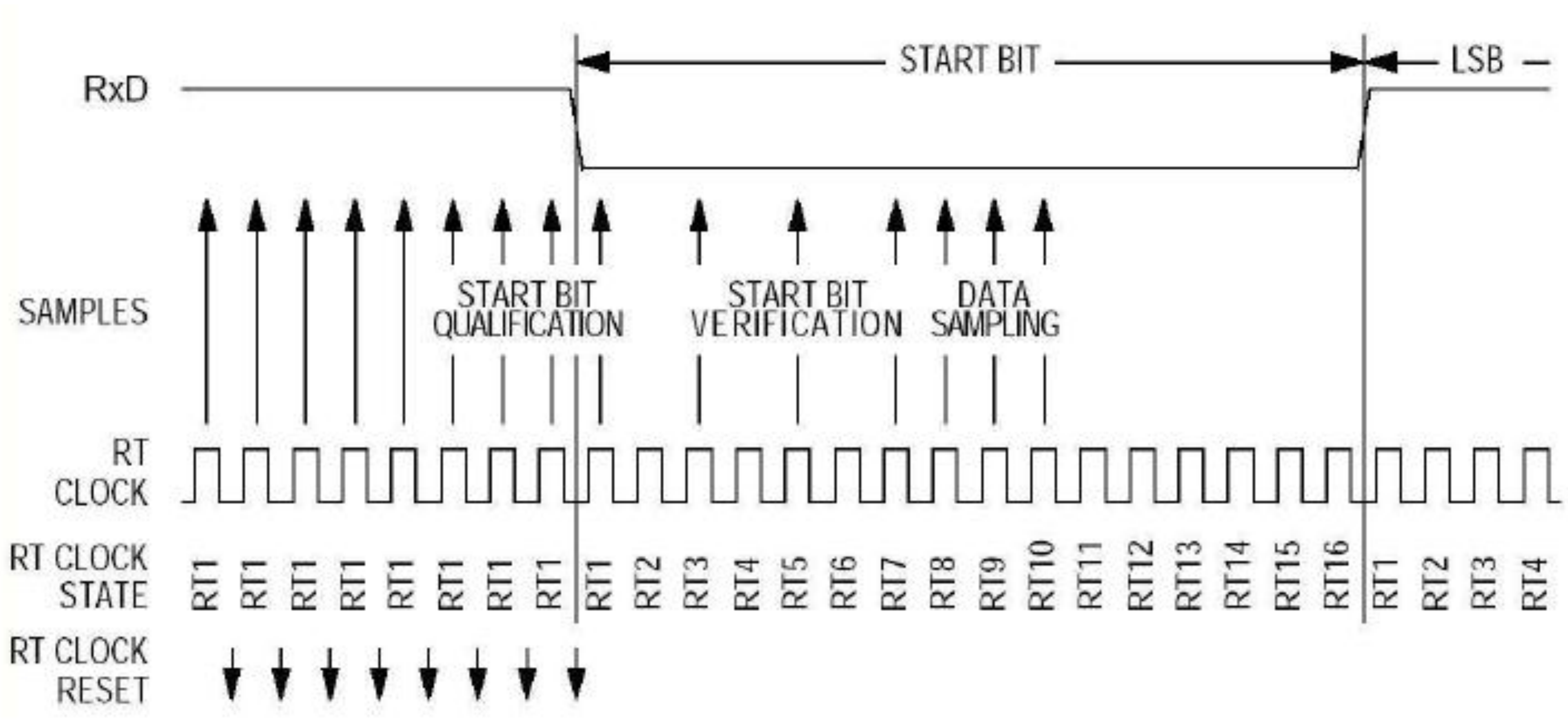


Data is sampled in the middle of the bit field:

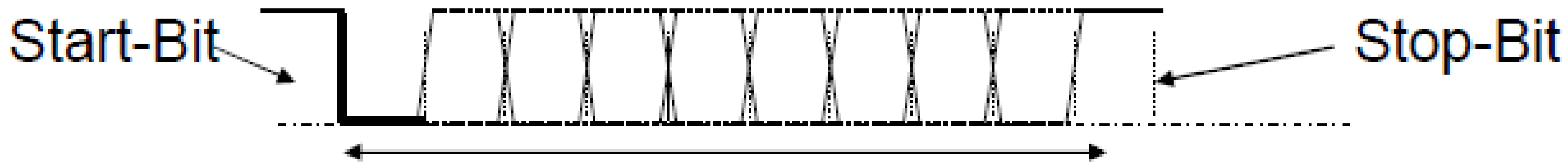


Bit Sampling

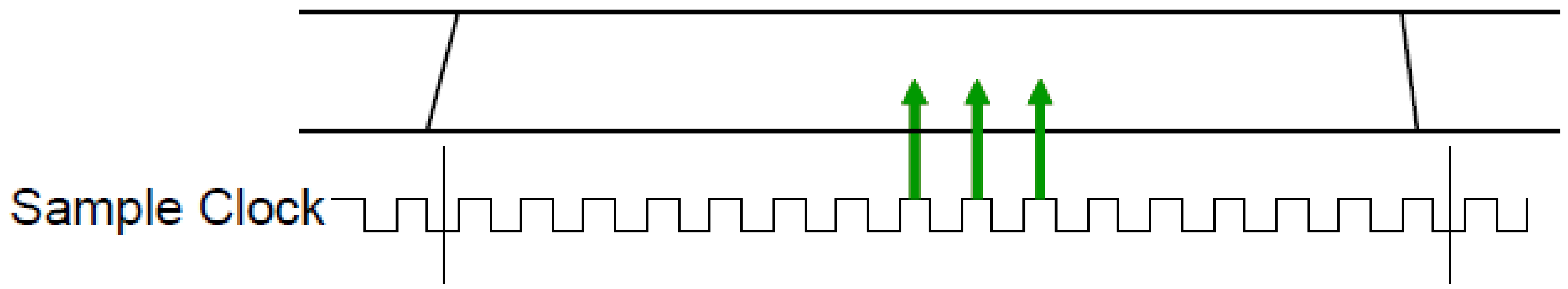
- Within a UART, clocked high speed sampling is used to aquire bit state


START BIT
LSB
RxD
SAMPLES
START BIT QUALIFICATION
START BIT VERIFICATION
DATA SAMPLING
RT CLOCK
RT CLOCK STATE
RT1 RT1 RT1 RT1 RT1 RT1 RT1 RT1 RT1 RT2 RT3 RT4 RT5 RT6 RT7 RT8 RT9 RT10 RT11 RT12 RT13 RT14 RT15 RT16 RT1 RT2 RT3 RT4
RT CLOCK RESET

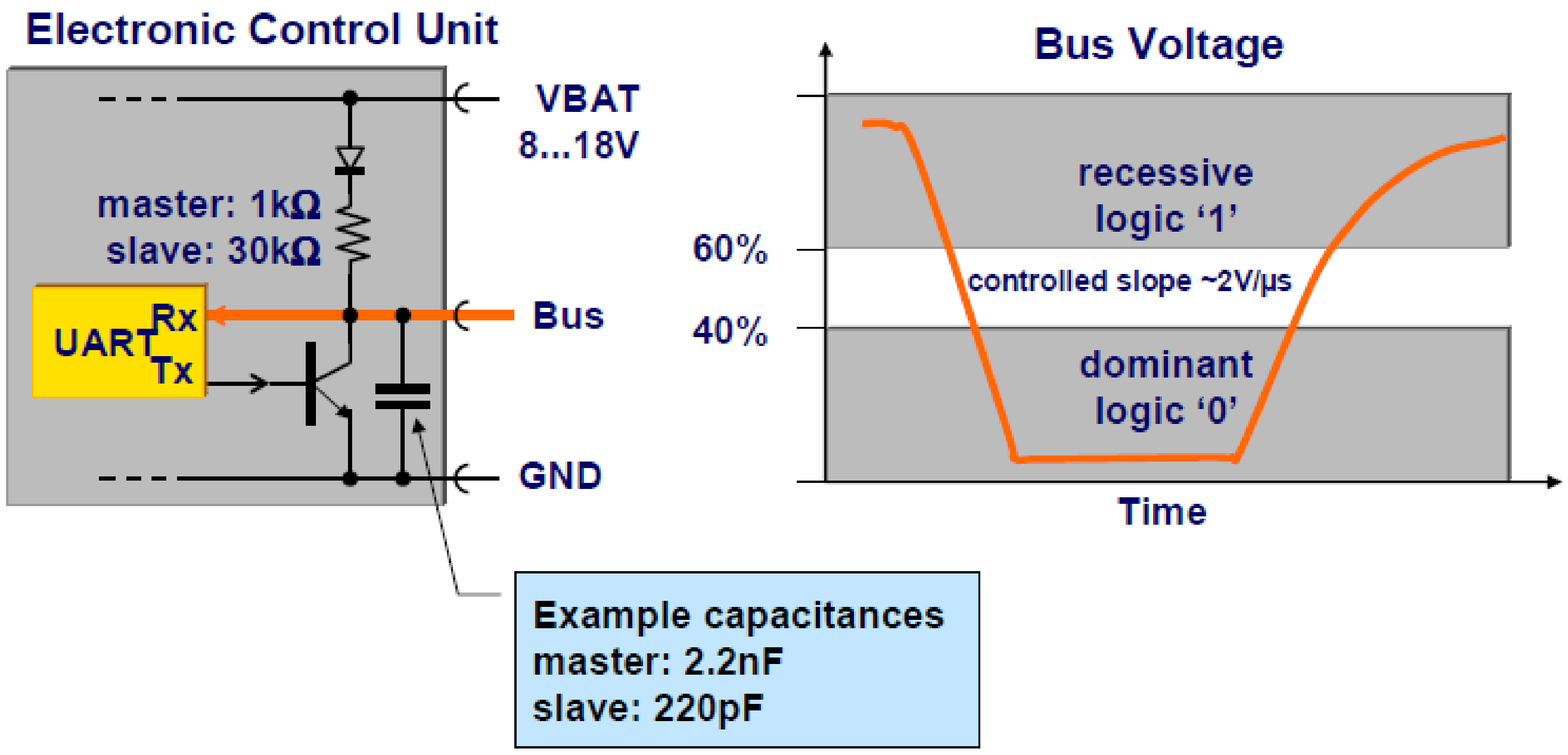
Bit-Synchronisation



- After recognition of a Low level in the start bit, the data is sampled at a rate 16 times the bit rate expected. The middle 3 samples must all agree for an error free reception of the data.
- A stop bit is expected after 1 start bit and 8 data bits in a typical message



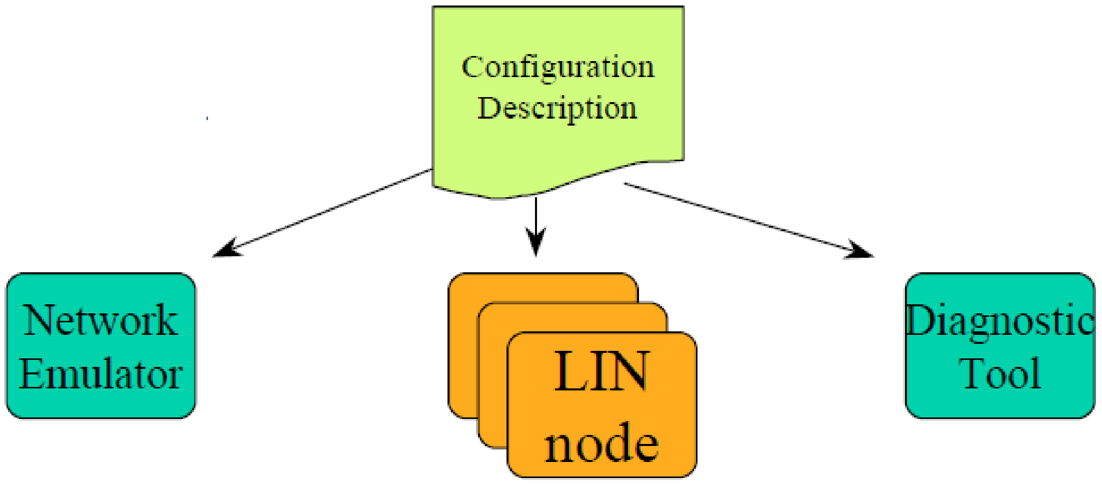
LIN Physical Interface



- **Note: The LIN specification refers to the ECU connector voltages !**

Network Configuration

- LIN Concept includes configuration interface:
  - LIN description file describes complete LIN network and also contains all information necessary to monitor the network.
  - LIN Configuration Language Description is part of the LIN Specification and gives tools the possibility to configure the network and the nodes, diagnose the traffic, and/or simulate missing nodes.



The Work-flow

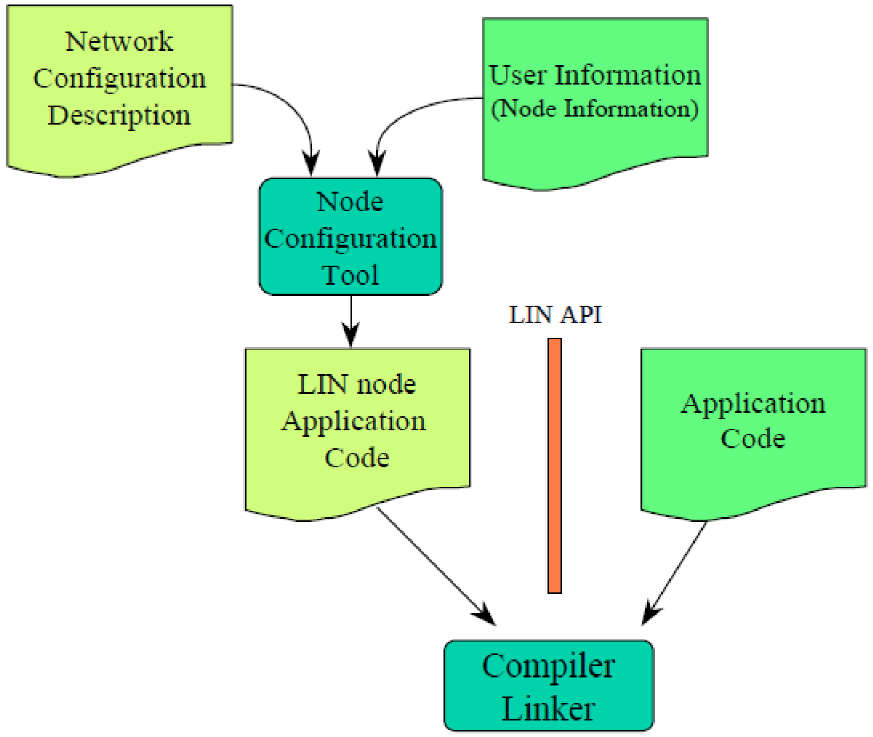
- **Data Input**
  - Definition of objects
  - Definition of relations between the objects
- **Data Processing**
  - Logical Signal Mapping
  - Signal Packing (Frame Editor/Frame Compiler)
  - Timing Analysis
- **Data Output**
  - Configuration file generation
  - Various optional customer-defined post-operations

Application Programmers Interface

Standard API

• simplifies design of Application Code

• opens up the market for competition.


Network Configuration Description
User Information (Node Information)
Node Configuration Tool
LIN API
LIN node Application Code
Application Code
Compiler Linker

LIN Tools by VCT

- **LIN Database Manager (LDM)**

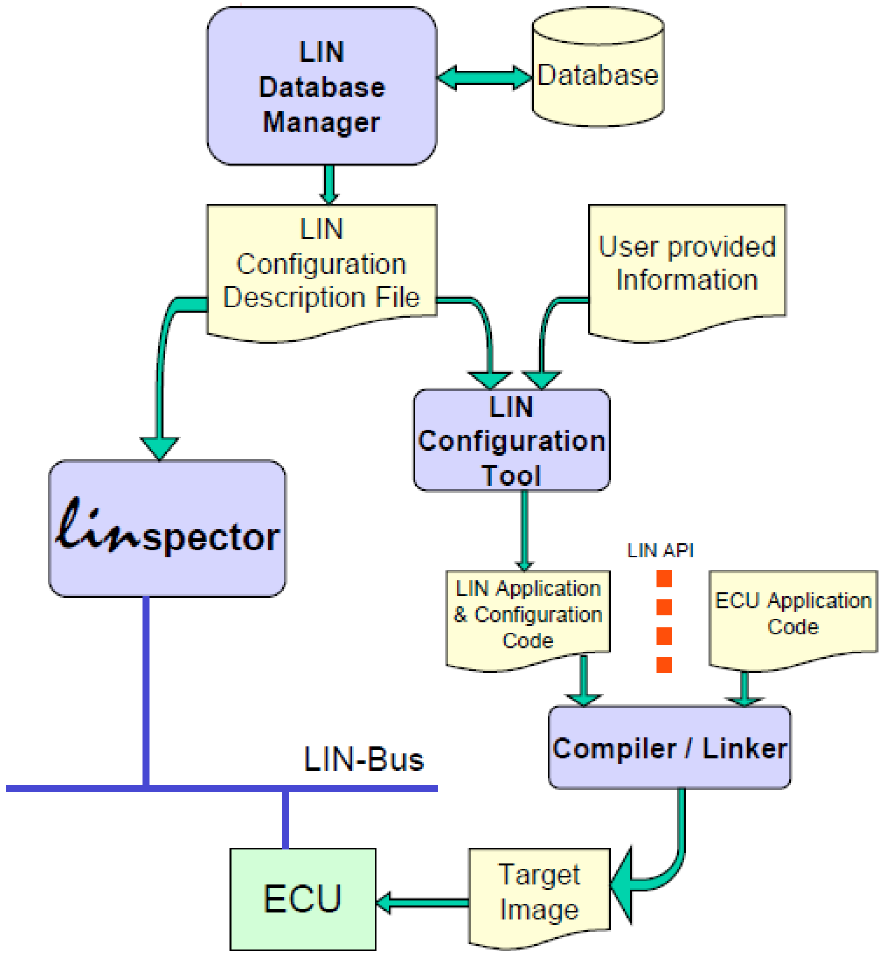
The LDM is a standalone offline tool, providing a user-friendly Windows interface for logically describing and configuring LIN systems at a high abstraction level.

- **LIN Configuration Tool (lcfg) and**

LIN Application Programmer's Interface (API) The LIN API provides the embedded SW developer an abstraction from details of information transfer. Together with the LIN Configuration Tool and an optimized embedded SW package the user gets correctness and quality together with efficiency and reconfiguration flexibility.

- **LINspector**

a highly flexible tool for testing and verifying communication for compliance with the LIN standard.



~ END ~