

Introduction to LIN

Webinar

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

Agenda

> **Information** **4**

Overview 6

LIN Workflow 9

LIN Physical Layer 12

LIN Communication 14

Synchronization of the LIN nodes 16

LIN Message & Scheduling 19

Message Types 27

Status & Network Management 30

Introduction to LIN slave diagnostics 34

Introduction to LIN slave configuration 39

Further Information 48

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

VectorAcademy

- Basics:
  - CAN, LIN, FlexRay, Ethernet & IP, …
- Products:
  - CANoe, CANalyzer, CANape, …
- Software components:
  - AUTOSAR, OSEK/VDX, …



Additional information:

- www.vector-academy.com
- www.vector.com /News & Events

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

Agenda

	Information	4
>	**Overview**	**6**
	LIN Workflow	9
	LIN Physical Layer	12
	LIN Communication	14
	Synchronization of the LIN nodes	16
	LIN Message & Scheduling	19
	Message Types	27
	Status & Network Management	30
	Introduction to LIN slave diagnostics	34
	Introduction to LIN slave configuration	39
	Further Information	48

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

- LIN – Local Interconnect Network
  - Network for sensors & actuators
  - Physical layer is realized as single wire system
  - Master-Slave architecture
  - Deterministic & dedicated communication principal
  - Typical use cases:
    - Mirror adjustment
    - Seat adjustment
    - Wiper control
    - Roof control
    - Car locking system
    - Air conditioning system
  - LIN clusters are always subsystems

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

LIN History



- Since the end of 1999, the LIN Consortium has been pursuing the goal of creating a comprehensive, simple, cost-effective and standardized communication concept for the sensor/actuator level

- LIN specification 2.2A
  - Physical Layer Specification
  - Protocol Specification
  - API Specification
  - Transport Layer Specification
  - Node Configuration and Identification
  - Diagnostics Specification
  - Configuration Language Specification
  - Node Capability Language Specification

Version	Date
LIN 1.0	1999-07-01
LIN 1.1	2000-03-06
LIN 1.2	2000-11-17
LIN 1.3	2002-12-13
LIN 2.0	2003-09-06
LIN 2.1	2006-11-24
LIN 2.2A	2010-12-31
ISO17987	In process

- SAE J2602 as the “US variant” of LIN is based on LIN 2.0

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

Agenda

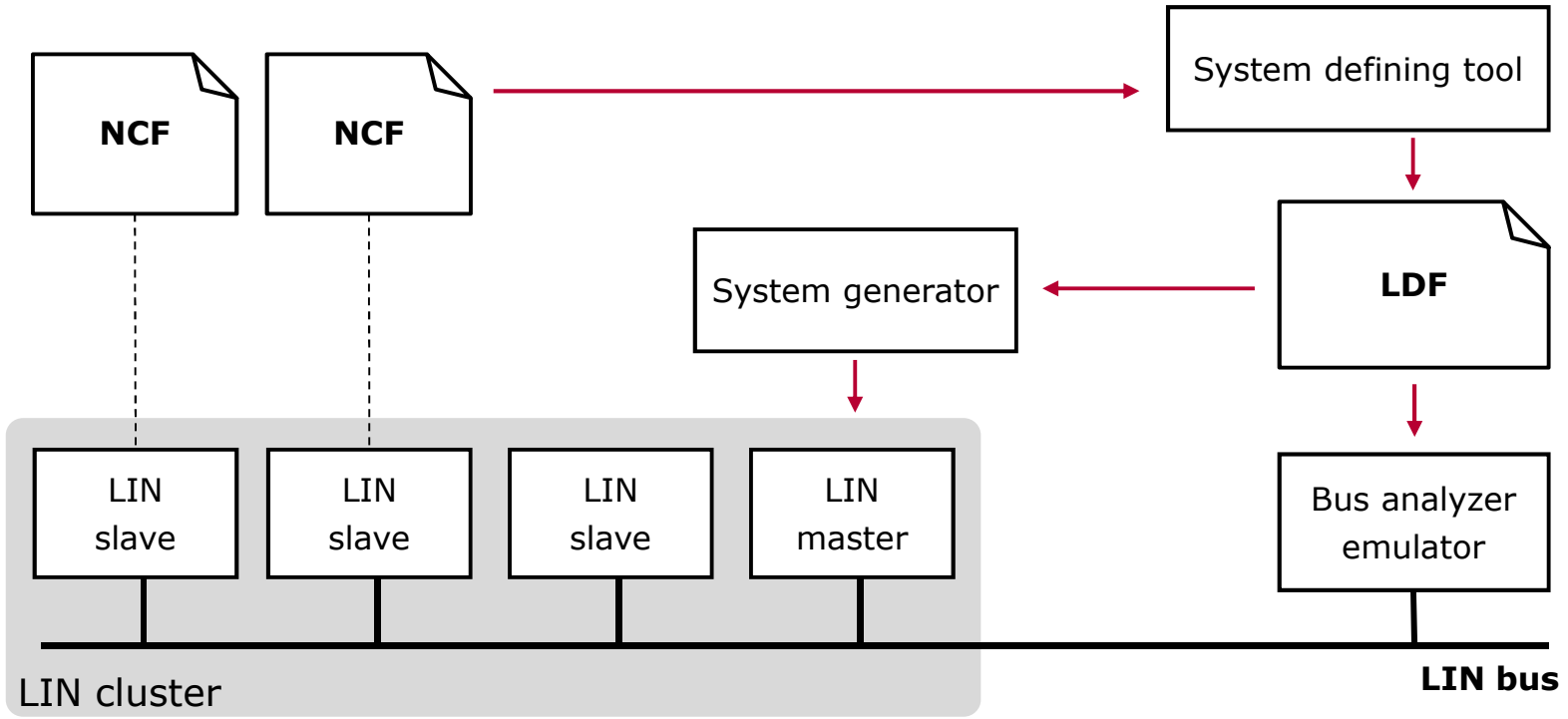
	Information	4
	Overview	6
>	**LIN Workflow**	**9**
	LIN Physical Layer	12
	LIN Communication	14
	Synchronization of the LIN nodes	16
	LIN Message & Scheduling	19
	Message Types	27
	Status & Network Management	30
	Introduction to LIN slave diagnostics	34
	Introduction to LIN slave configuration	39
	Further Information	48

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

LIN Workflow

Initial concept

- So called “off-the-shelf nodes” (NCF) are realized by suppliers
- OEM combines slaves to LIN cluster (NCFs LDF)
- Master is realized based on LDF

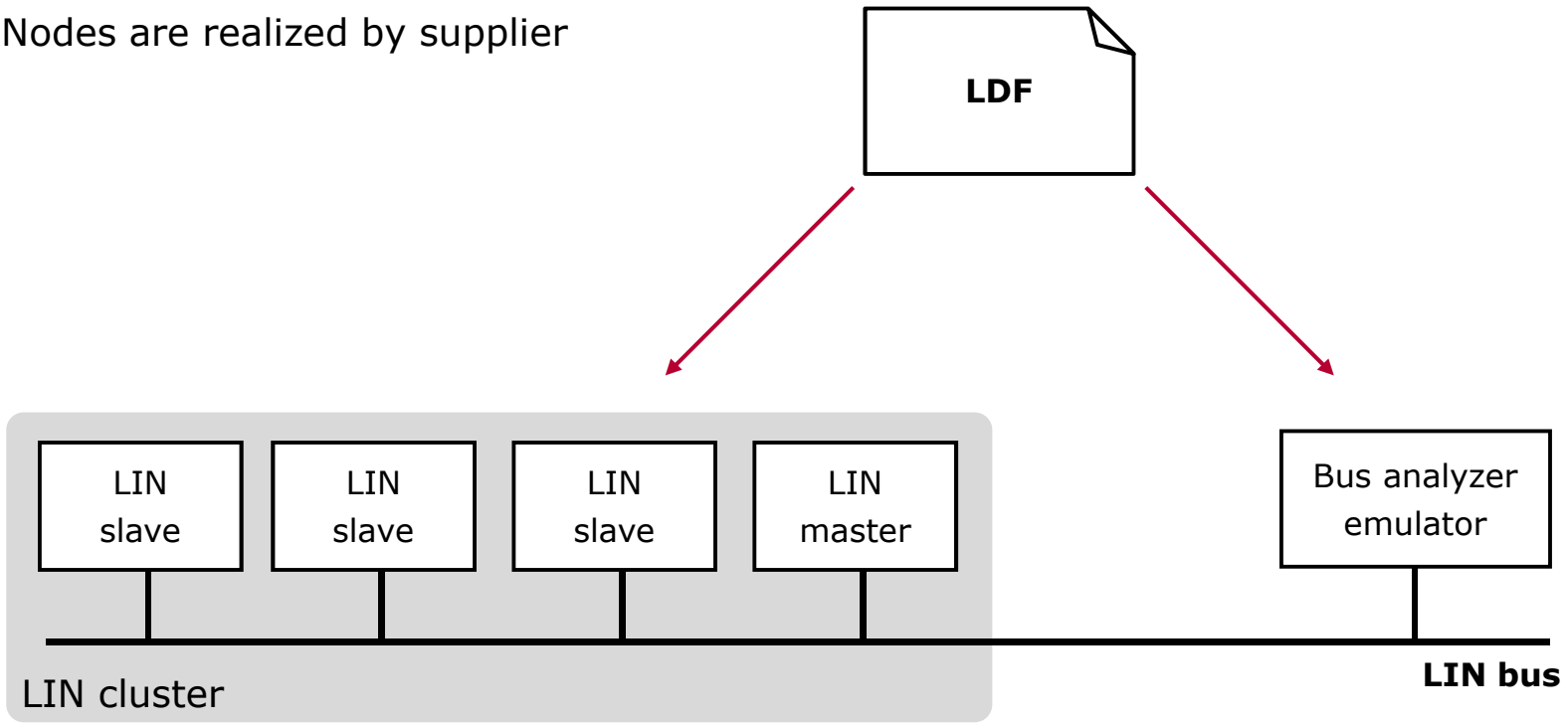


- NCF: Node Capability File
- LDF: LIN Description File

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

Typical workflow

- LIN cluster is defined by OEM
- LDF is cluster specification
- Nodes are realized by supplier



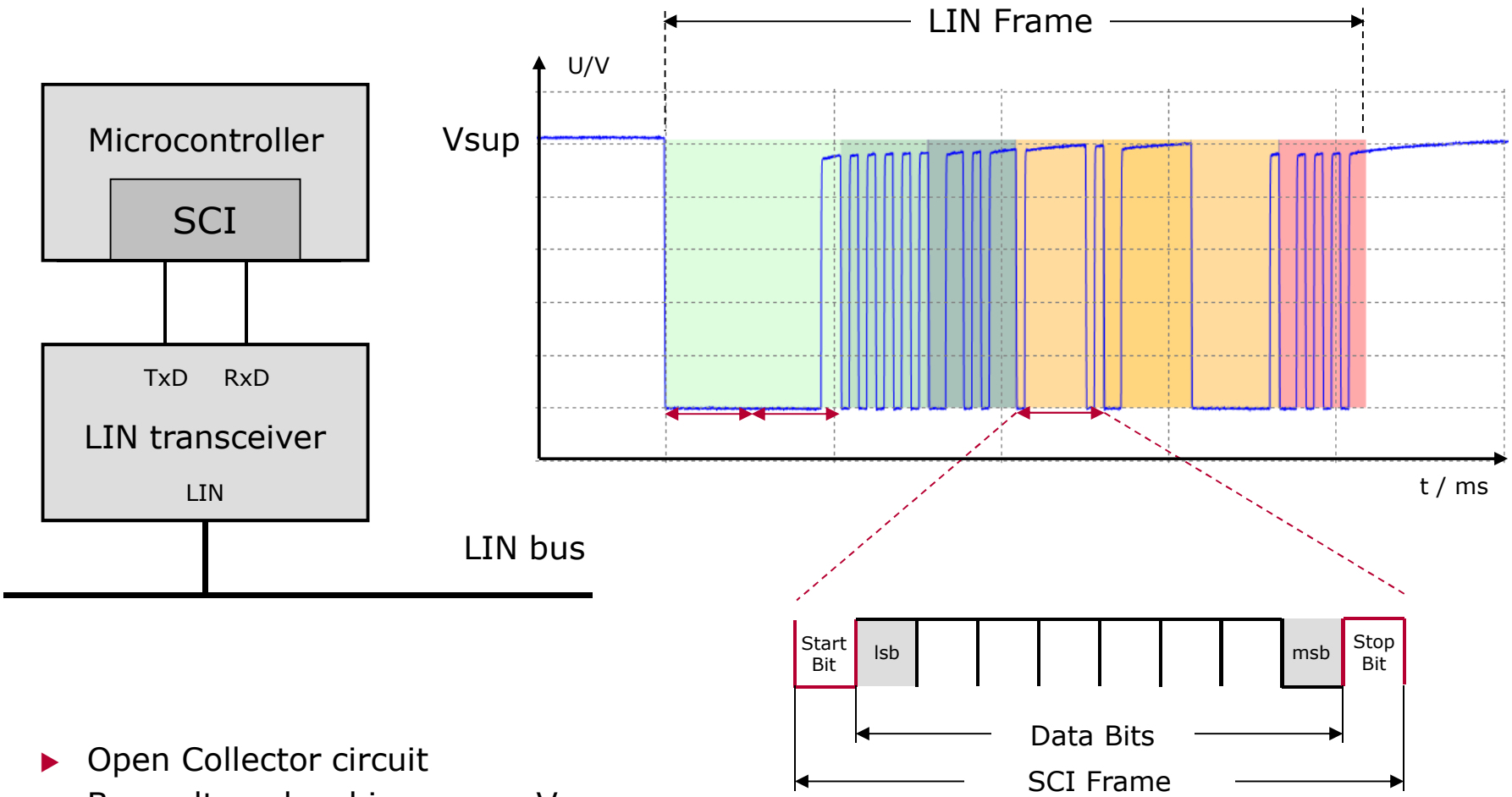
© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

Agenda

	Information	4
	Overview	6
	LIN Workflow	9
>	**LIN Physical Layer**	**12**
	LIN Communication	14
	Synchronization of the LIN nodes	16
	LIN Message & Scheduling	19
	Message Types	27
	Status & Network Management	30
	Introduction to LIN slave diagnostics	34
	Introduction to LIN slave configuration	39
	Further Information	48

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

LIN Physical Layer



- Open Collector circuit
- Bus voltage level is approx. V_{Supply}
- Bus is terminated within the LIN Master (1kOhm)

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

Agenda

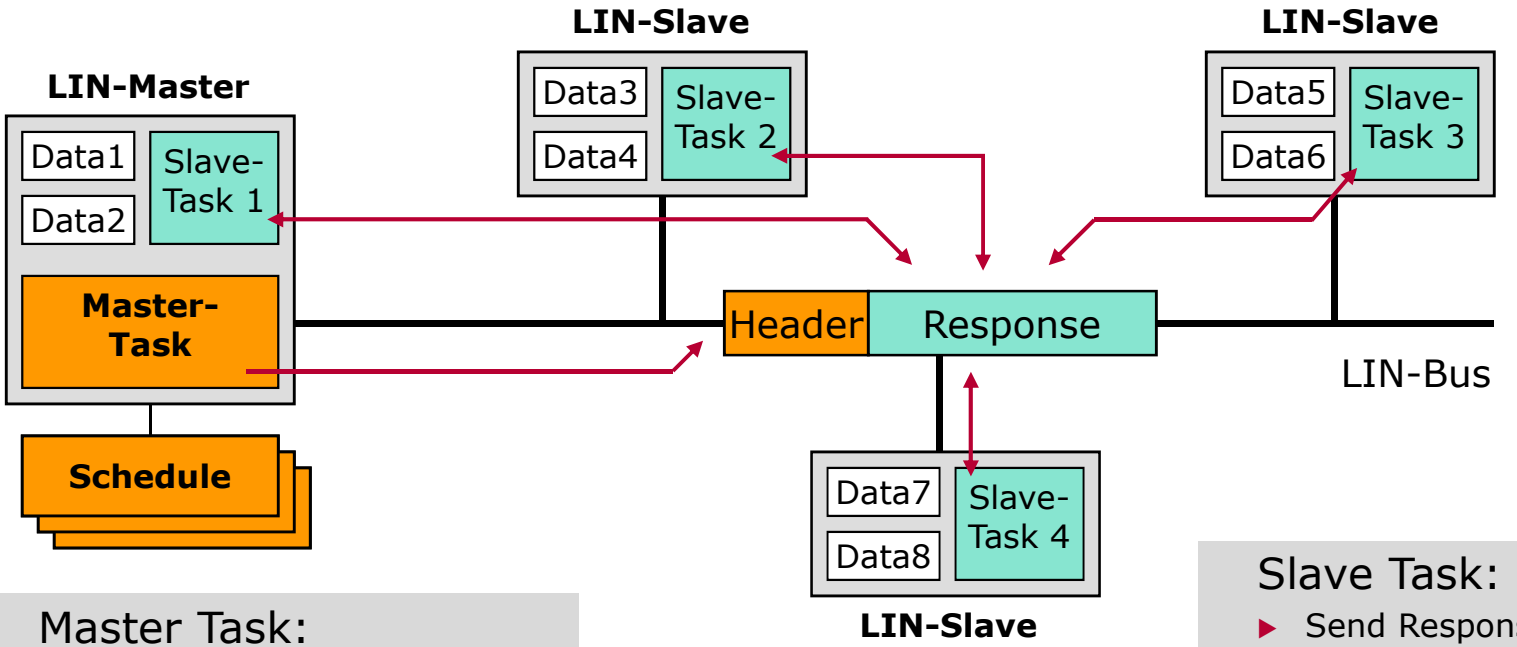
	Information	4
	Overview	6
	LIN Workflow	9
	LIN Physical Layer	12
>	**LIN Communication**	**14**
	Synchronization of the LIN nodes	16
	LIN Message & Scheduling	19
	Message Types	27
	Status & Network Management	30
	Introduction to LIN slave diagnostics	34
	Introduction to LIN slave configuration	39
	Further Information	48

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

LIN Communication

Centrally controlled message distribution system

- LIN nodes do not have equal rights due to master-slave architecture
- LIN master delegates communication (Delegated Token Principle)
- Message distribution based on message addressing via Broadcast



Master Task:

- Send Headers deterministic due to LIN schedule

Slave Task:

- Send Response
- Read Response
- Ignore Response

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

Agenda

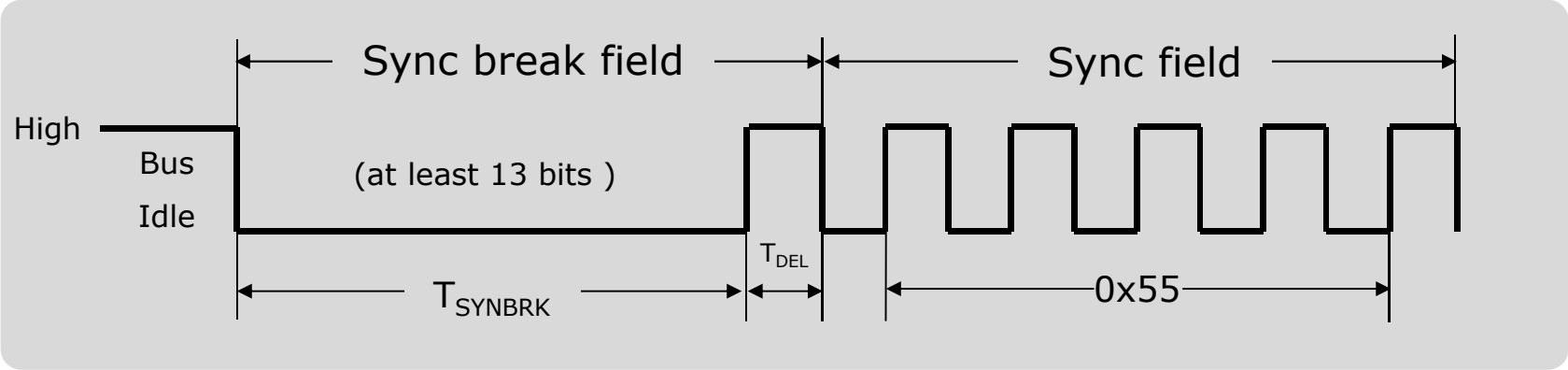
	Information	4
	Overview	6
	LIN Workflow	9
	LIN Physical Layer	12
	LIN Communication	14
>	**Synchronization of the LIN nodes**	**16**
	LIN Message & Scheduling	19
	Message Types	27
	Status & Network Management	30
	Introduction to LIN slave diagnostics	34
	Introduction to LIN slave configuration	39
	Further Information	48

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

Synchronization of the LIN nodes

Initial synchronization

- LIN nodes are not synchronized before communication (Bus Idle)
- For cost and effort reasons
  - No clock line
  - RC resonators instead of high precision clocks lead to tolerances up to +/-14 %



Sync break field:

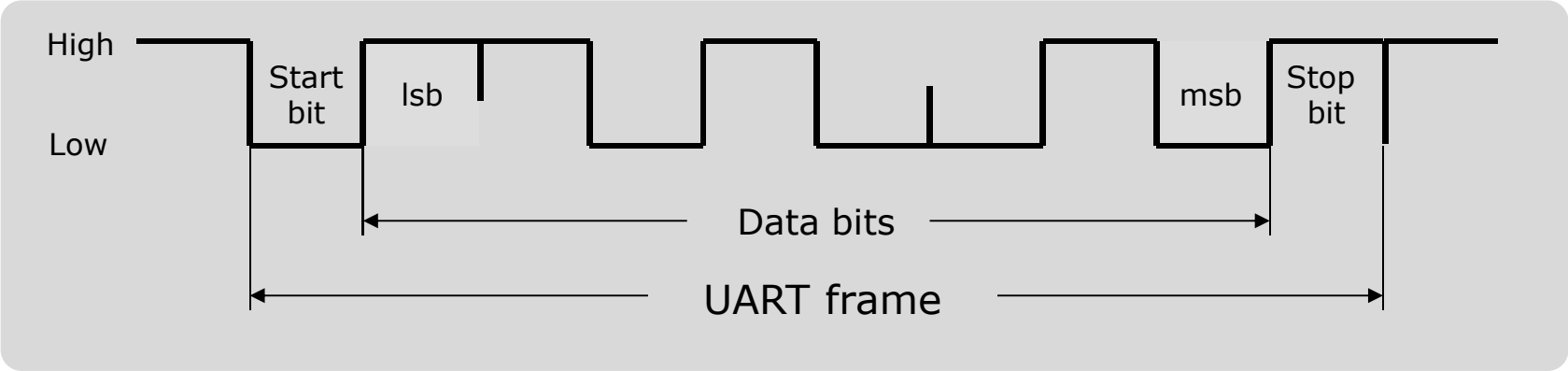
- Minimum length: 13 bits dominant, 1 bit recessive (calculated)
- Typical length: 18 bits dominant, 2 bits recessive (50% clock rate)

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

Synchronization of the LIN Nodes

Resynchronization

- Asynchronous transmission method



- Frame type: UART (Universal Asynchronous Receiver and Transmitter)
- Coding: 8N1

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

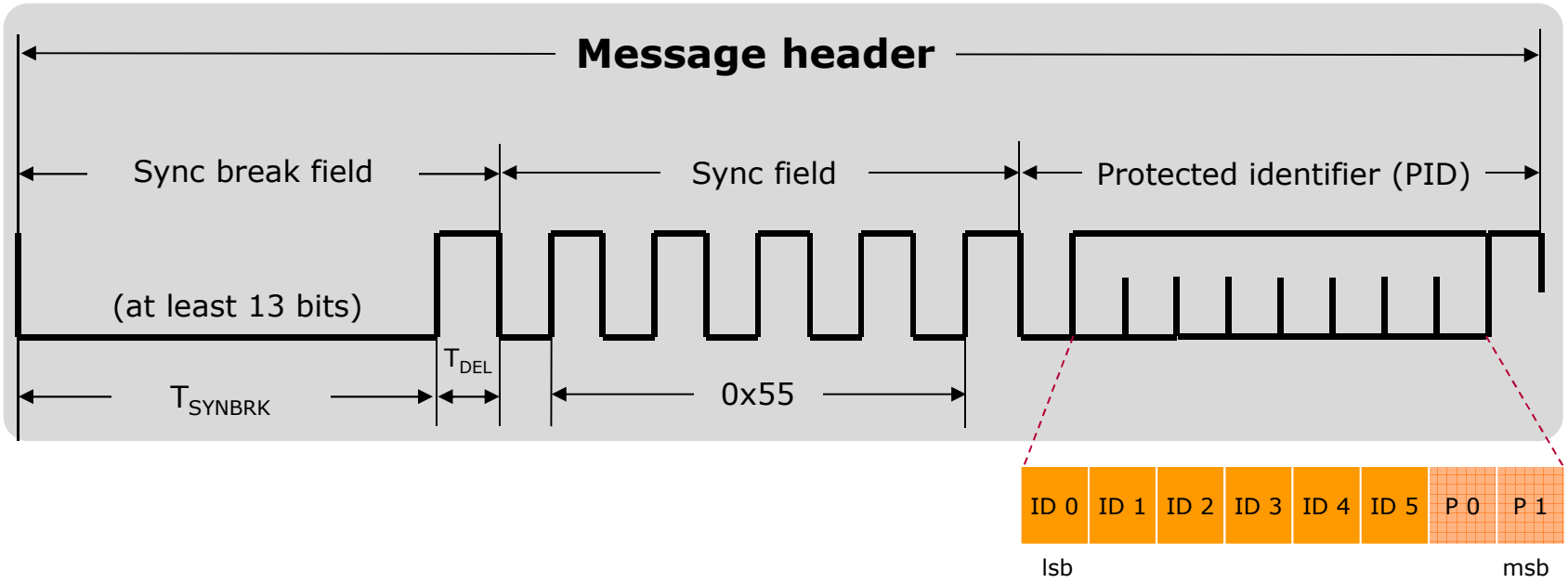
Agenda

	Information	4
	Overview	6
	LIN Workflow	9
	LIN Physical Layer	12
	LIN Communication	14
	Synchronization of the LIN nodes	16
>	**LIN Message & Scheduling**	**19**
	Message Types	27
	Status & Network Management	30
	Introduction to LIN slave diagnostics	34
	Introduction to LIN slave configuration	39
	Further Information	48

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

Header

- The message header is sent by the master task
- The message header is used for synchronization
- The message header comprises the Identifier



Address range: 0-63

- 60, 61: Diagnostics
- 62, 63: Reserved

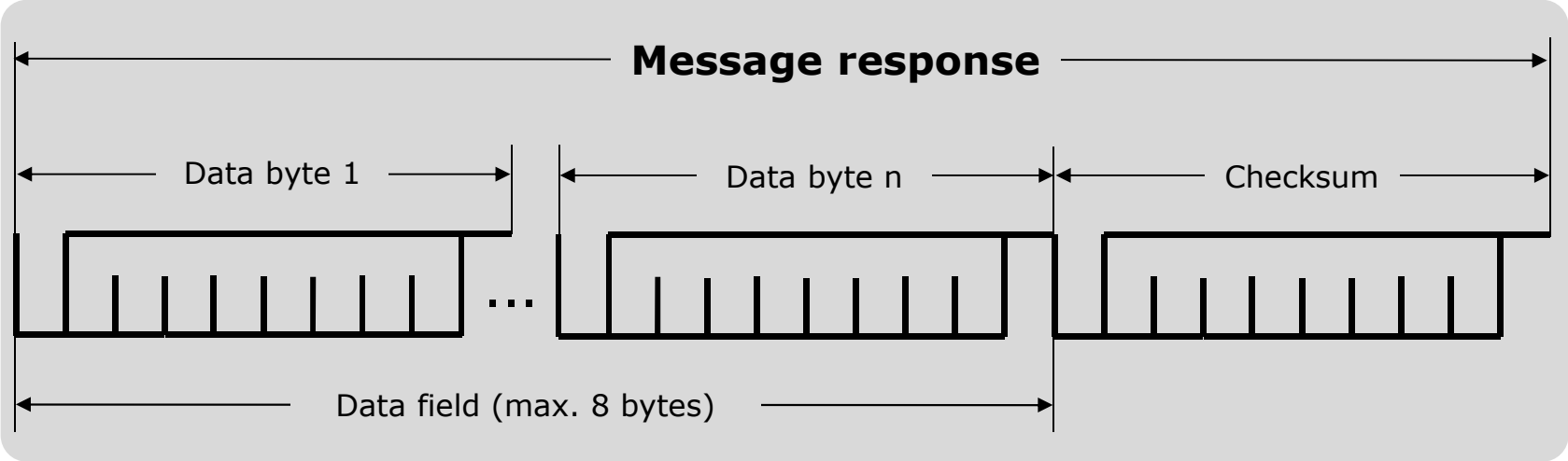
Parity:

- P0 = XOR with ID0, ID1, ID2, ID4 (even parity)
- P1 = XOR with ID1, ID3, ID4, ID5 (odd parity)

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

Response

- The message response is sent by a slave task
- The message response contains the data and the checksum
- Payload: 1 to 8 bytes
- Checksum, dependent on the LIN version

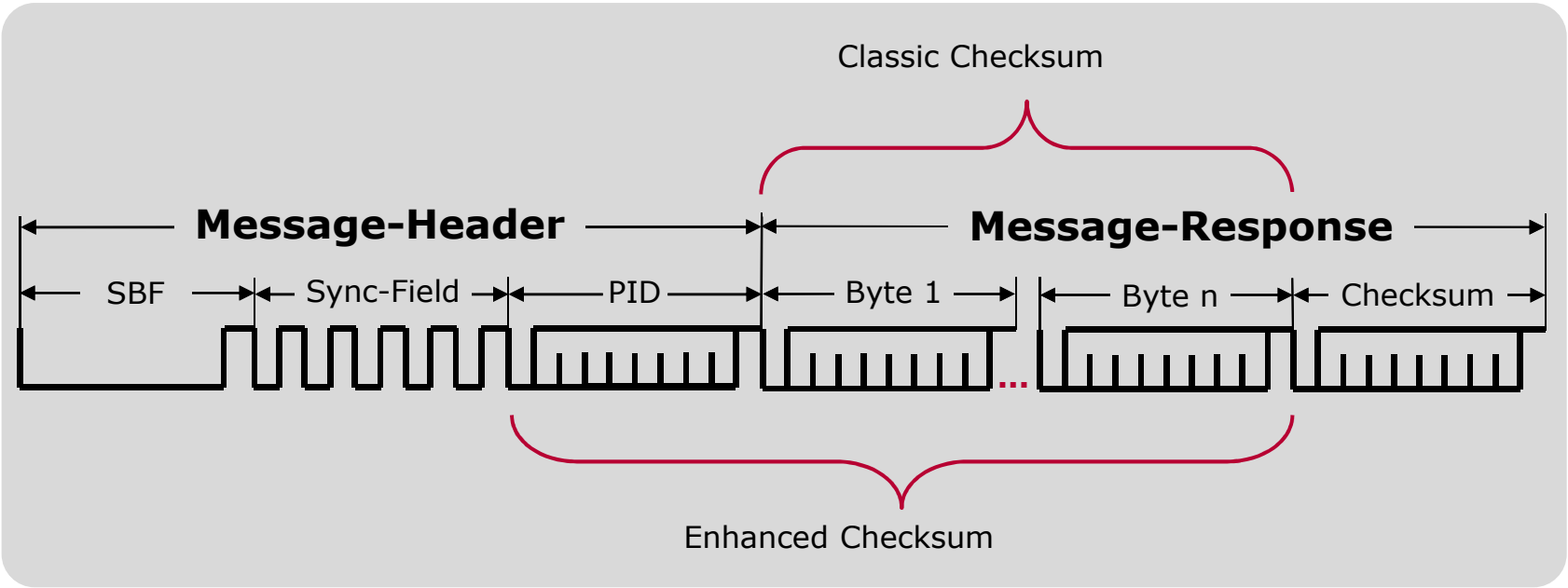


- n – Number of data byte

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

Checksum

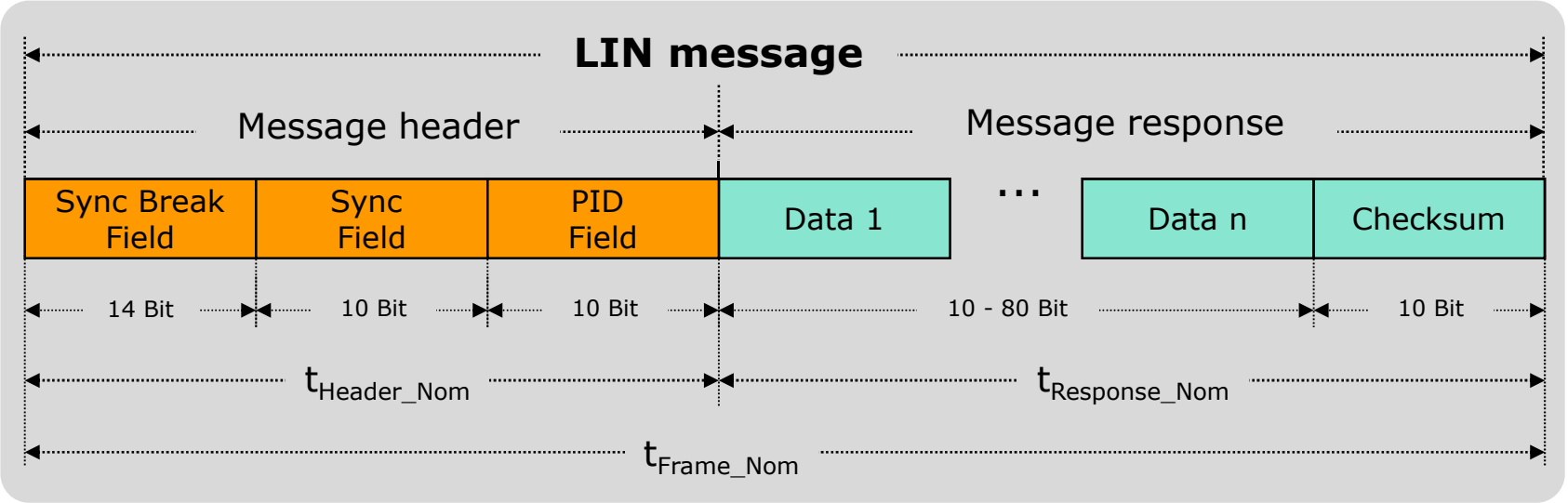
- In LIN there are two check sum models:
  - Classic Checksum (LIN 1.1, LIN 1.2 and LIN 1.3)
  - Enhanced Checksum (LIN 2.0, LIN 2.1 and LIN 2.2)
- In all LIN versions Diagnostic Frames have a Classic Checksum



© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

LIN message duration

- Setting up a schedule table is based on the duration of the LIN messages



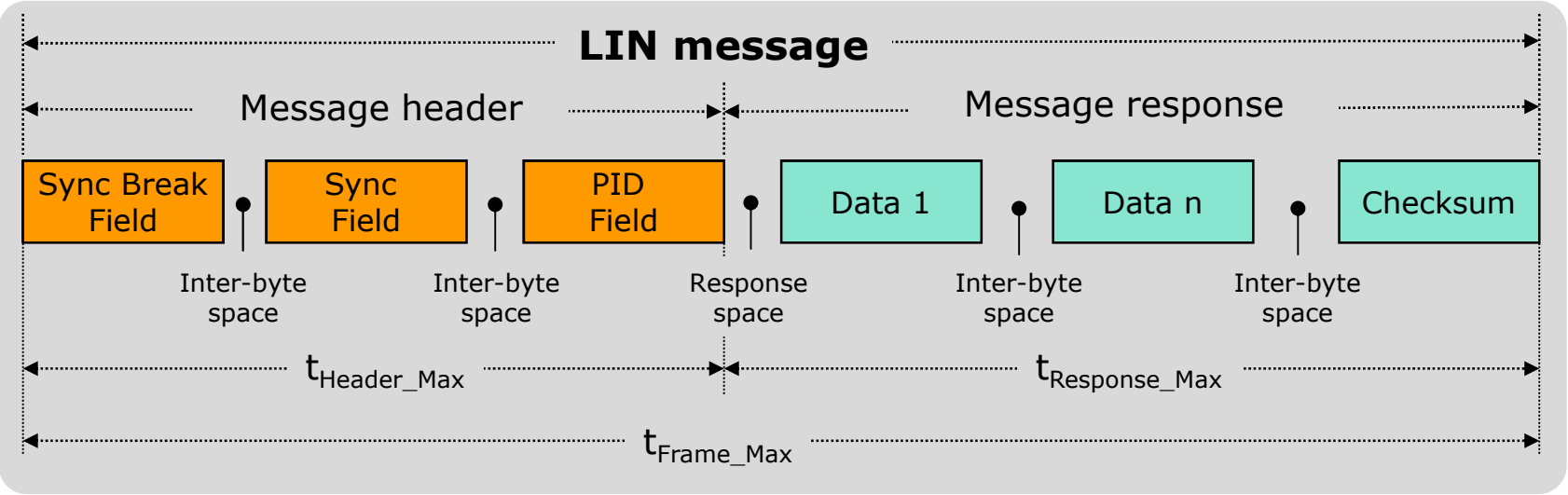
- $t_{Frame_Nom} = t_{Header_Nom} + t_{Response_Nom} = (n \cdot 10 + 44) \cdot t_{Bit}$

- n – Number of data bytes (Payload)
- $t_{Bit} = Baudrate^{-1}$ e.g. $(19.2kBit/s)^{-1} = 52.1\mu s$

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

Time Reserve

- LIN slaves often are low performant and delays may occur



- So a time reserve of up to 40 % is available for transmitting a LIN message

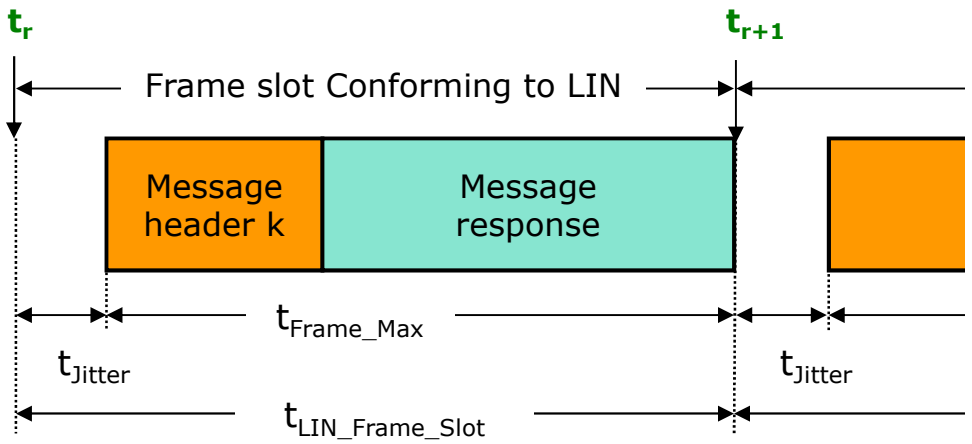
- $t_{Frame_Max} = 1.4 \bullet t_{Frame_Nom} = [1.4 \bullet (n \bullet 10 + 44)] \bullet t_{Bit}$

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

LIN conforming schedule

LIN schedule

t_r	Message Header (ID k)
t_{r+1}	Message Header (ID n)
t_{r+2}	Message Header (ID 0)
⋮	

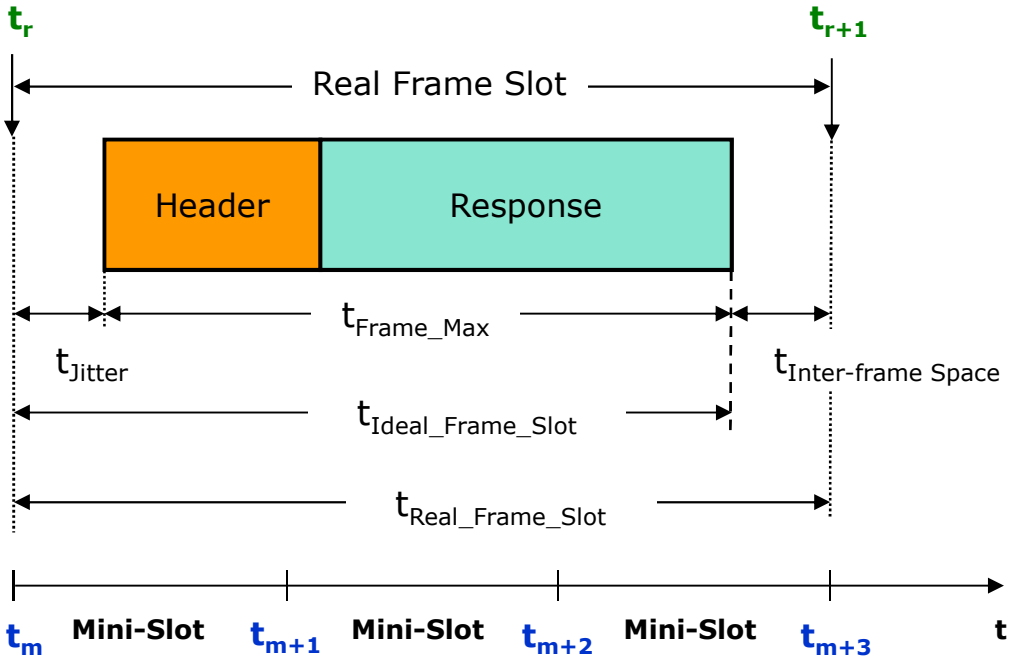


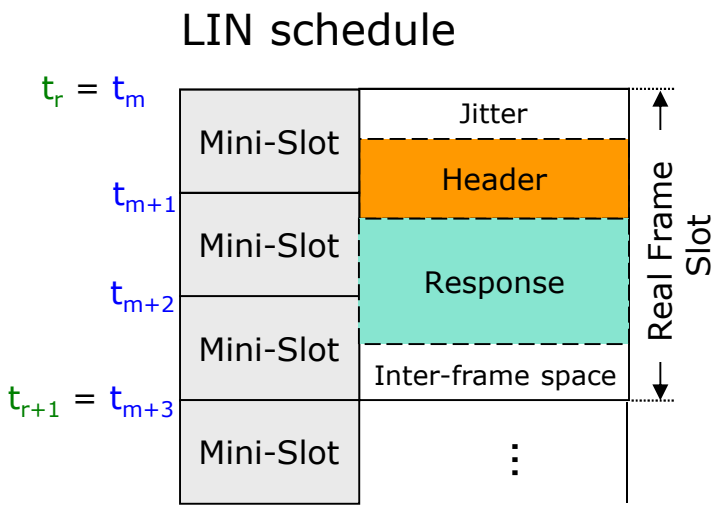
- LIN master controls the communication in the LIN cluster
- LIN frame slot width: $t_{LIN_Frame_Slot} = 1.4 \bullet t_{Frame_Nom} + t_{Jitter} = t_{Frame_Max} + t_{Jitter}$
- Deterministic data transmission, no overload situations
- Note: Each slot has an individual width depending on response length

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

Real schedule

- Not every possible slot width is realized
- Slot width is adjustable in steps to optimize performance (Mini-Slots)





(1) Ideal frame slot width: $t_{Ideal_Frame_Slot} = 1.4 \bullet t_{Frame_Nom} + t_{Jitter}$

(2) Integer multiple of mini slot: $t_{Real_Frame_Slot} = n \bullet t_{Time_Base} \geq t_{Ideal_Frame_Slot}$

(3) Real frame slot width: $t_{Real_Frame_Slot} = t_{Ideal_Frame_Slot} + t_{Inter_Frame_Space}$ (Surplus)

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

Agenda

	Information	4
	Overview	6
	LIN Workflow	9
	LIN Physical Layer	12
	LIN Communication	14
	Synchronization of the LIN nodes	16
	LIN Message & Scheduling	19
>	**Message Types**	**27**
	Status & Network Management	30
	Introduction to LIN slave diagnostics	34
	Introduction to LIN slave configuration	39
	Further Information	48

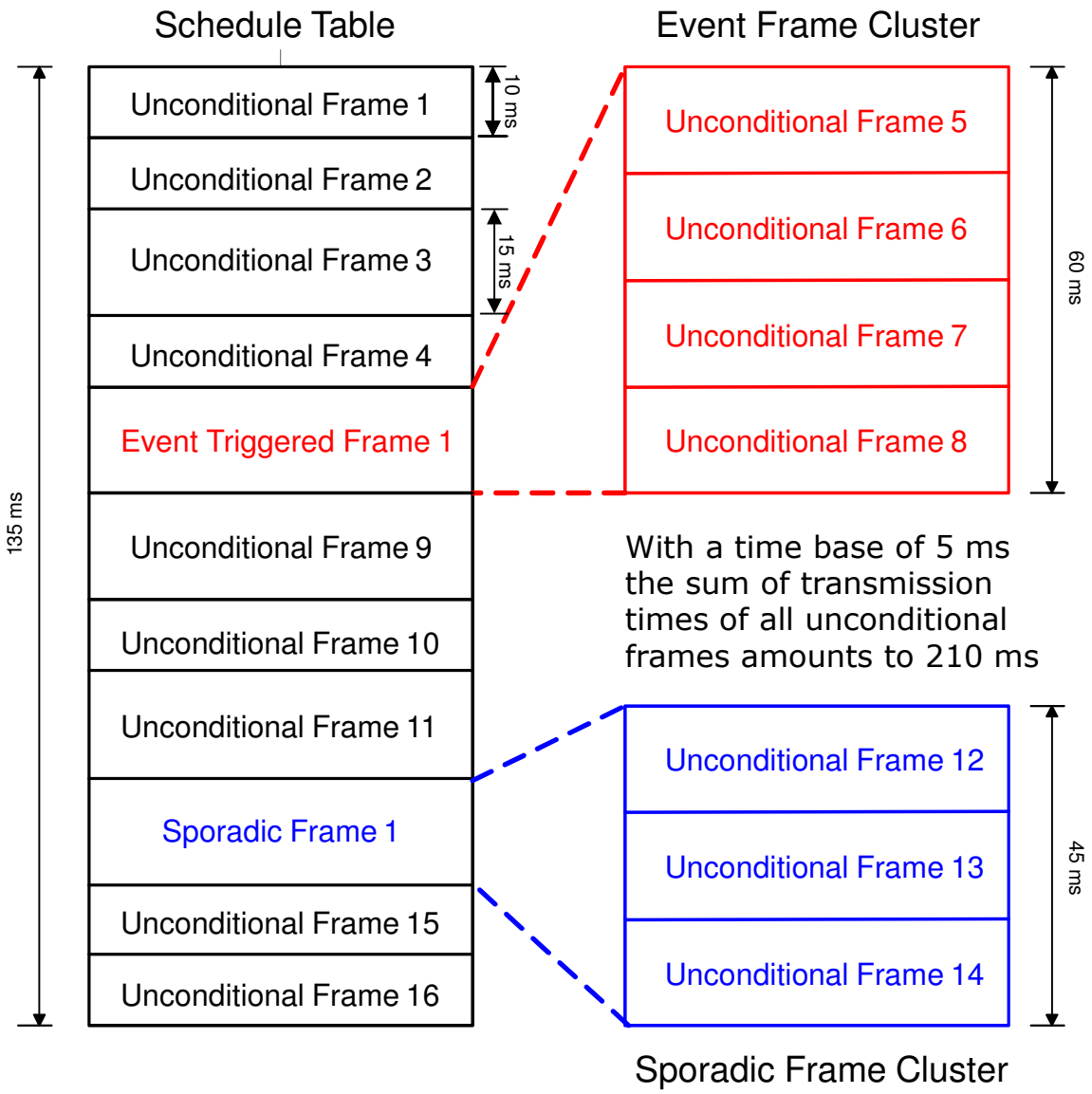
© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

Message Types

- Unconditional frame
  - Normal LIN messages with identifiers between 0 and 59
  - The message is sent in every turn of the schedule it belongs to
  - Each header has its corresponding response
- Diagnostic frame (ID 60-61)
  - ID=60: Master request frame (= Diagnostic request)
  - ID=61: Slave request frame (= Diagnostic response)
- Event triggered frame (ID 0-59)
  - Unconditional frames which need to be sent only seldom by LIN slaves can be integrated in one event triggered frame cluster
  - An ETF-Header may be answered by more than one response (!!!Collision!!!)
  - Collisions may occur but have to be resolved by the LIN master
- Sporadic frame (ID 0-59)
  - Unconditional frames which need to be sent only seldom by the LIN master can be integrated in one sporadic frame cluster
  - Each sporadic frame header has its corresponding response

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

Message Types



By grouping frames in sporadic frame and event triggered frame clusters the cycle time of a schedule can be diminished substantially.

Please find more details as part of the current LIN specification 2.2A

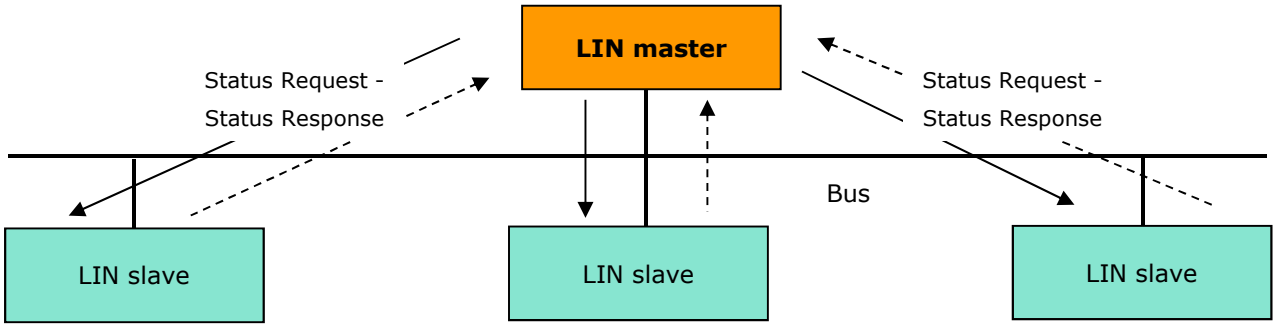
© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

Agenda

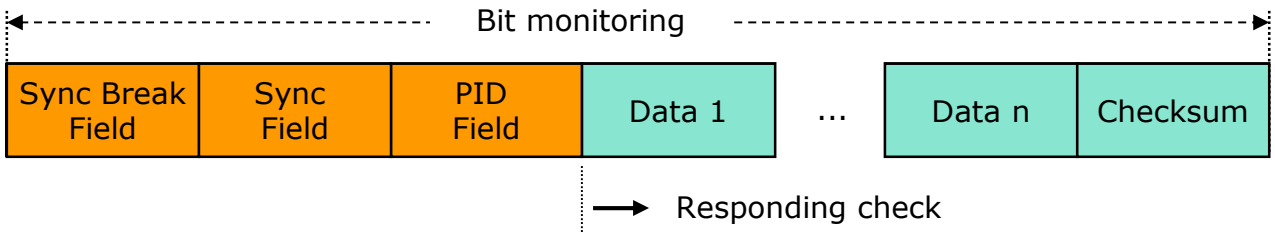
	Information	4
	Overview	6
	LIN Workflow	9
	LIN Physical Layer	12
	LIN Communication	14
	Synchronization of the LIN nodes	16
	LIN Message & Scheduling	19
	Message Types	27
>	**Status & Network Management**	**30**
	Introduction to LIN slave diagnostics	34
	Introduction to LIN slave configuration	39
	Further Information	48

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

Status & Network Management

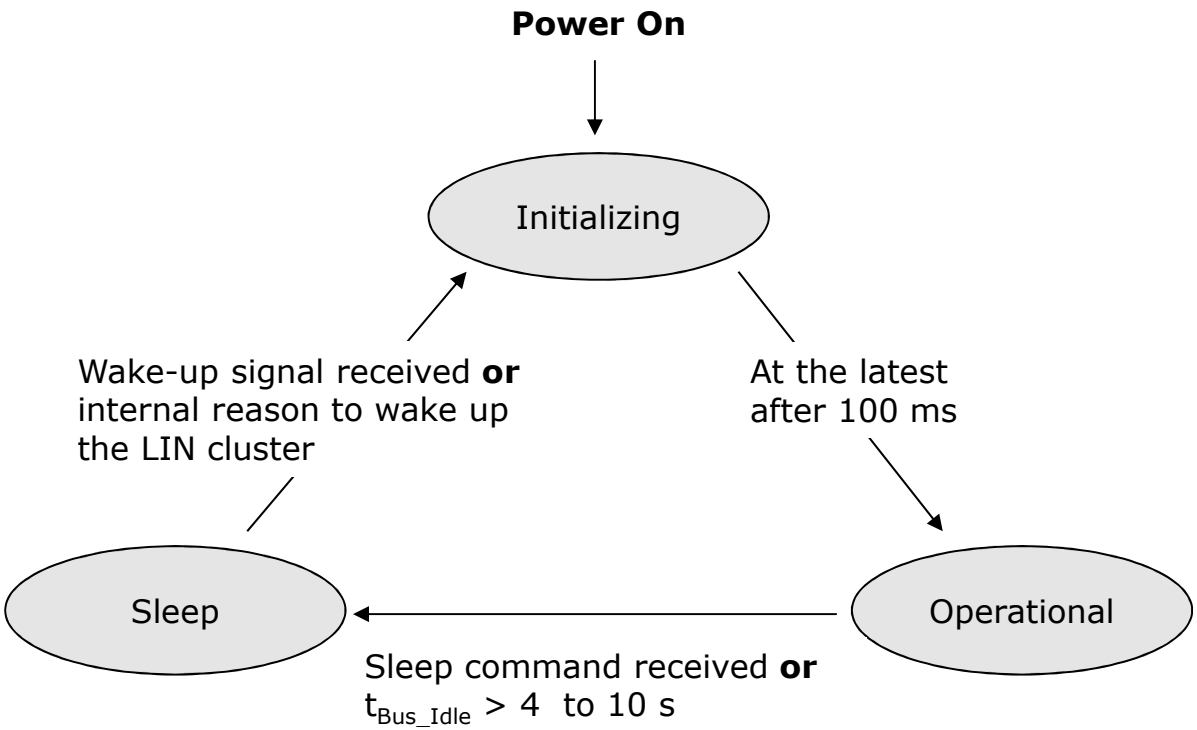


- Error signaling with the help of a "Status bit" (Response_Error)
- Use of Unconditional Frames
  - Message header: Status request
  - Message response: Status response (with Response_Error)
- LIN master collects the status bits
- Error handling is not part of the LIN specification



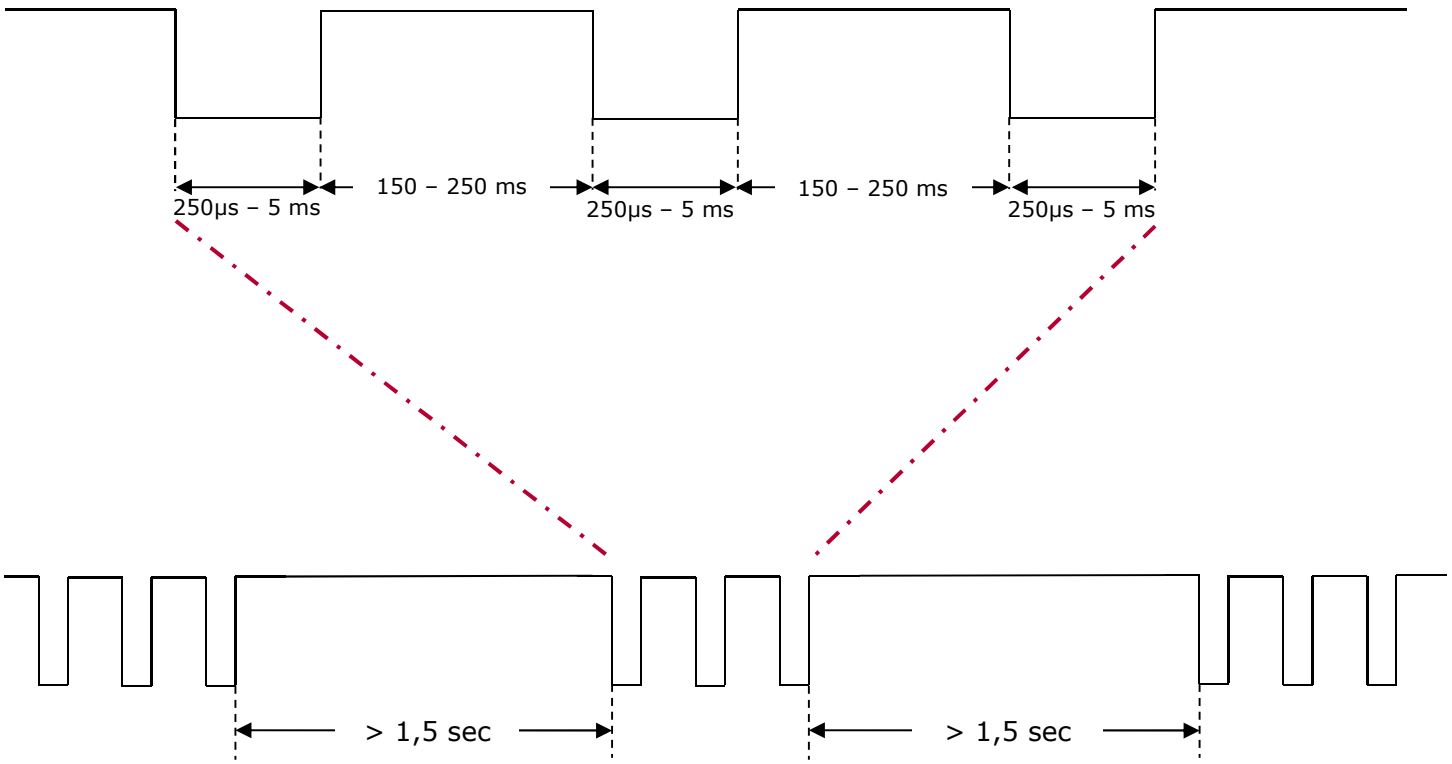
© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

Network Management



© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

Does a wake-up-signal have no effect, it must be repeated:



© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

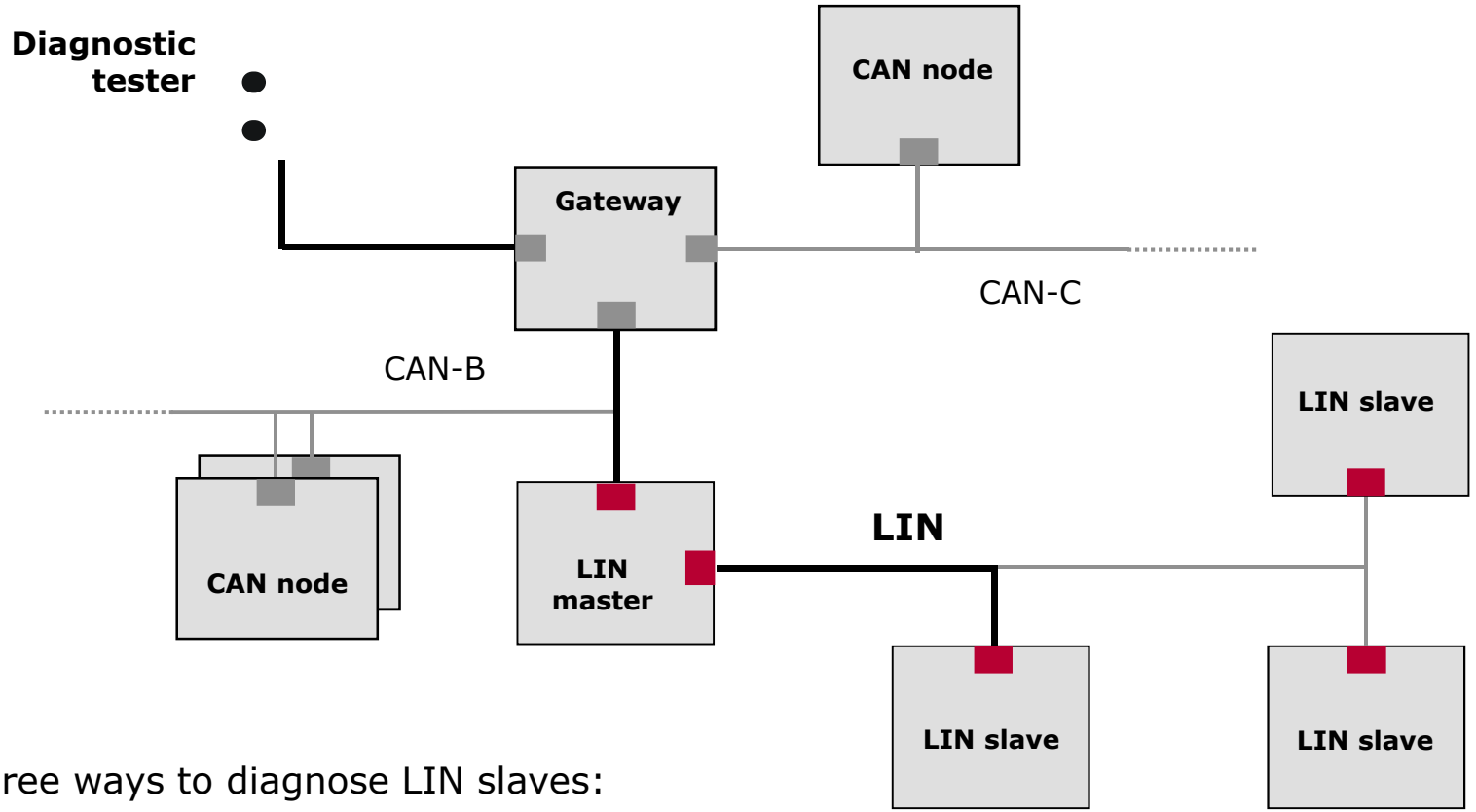
Agenda

	Information	4
	Overview	6
	LIN Workflow	9
	LIN Physical Layer	12
	LIN Communication	14
	Synchronization of the LIN nodes	16
	LIN Message & Scheduling	19
	Message Types	27
	Status & Network Management	30
>	**Introduction to LIN slave diagnostics**	**34**
	Introduction to LIN slave configuration	39
	Further Information	48

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

Introduction to LIN slave diagnostics

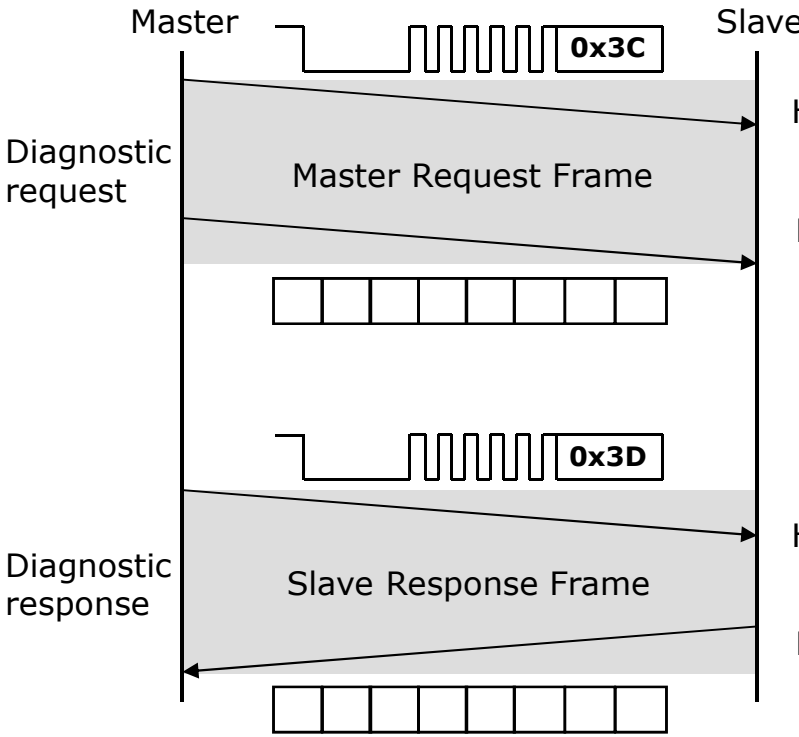
Diagnostics with Diagnostic Tester



Three ways to diagnose LIN slaves:

- Signal-based diagnostics
- TP-based diagnostics (based on CAN ISO15765-2)
- User defined diagnostics

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.



SID: Service identifier

Header

Response

NAD	PCI	SID	D1	D2	D3	D4	D5

PCI=0x0x

0: Single Frame (SF)

x: Number of valid bytes after PCI

Header

Response

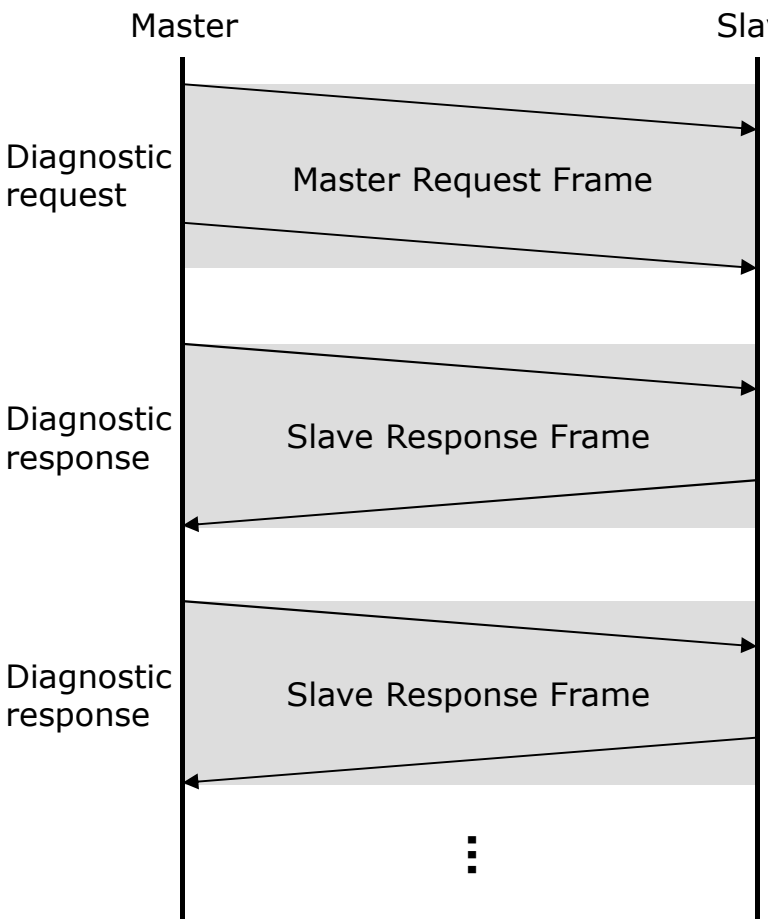
NAD	PCI	RSID	D1	D2	D3	D4	D5

RSID:	Response service identifier
Positive response:	RSID = SID + 0x40 D1-D5 = Service-dependent
Negative response:	RSID = 0x7F D1 = SID D2 = Error Code

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

Introduction to LIN slave diagnostics

Segmented Diagnostic Protocol per ISO 15765



Header

Response

Single **F**rame (PCI=0x0x, x number of valid bytes)

NAD	PCI	SID	D1	D2	D3	D4	D5

Header

Response

First **F**rame (PCI=0x1y, y = LEN extension)

NAD	PCI	LEN	RSID	D1	D2	D3	D4

Header

Response

Consecutive **F**rame (PCI=0x2z, z Counter 0-15)

NAD	PCI	D1	D2	D3	D4	D5	D6

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

Introduction to LIN slave diagnostics

Diagnostic Classes

Slave Diagnostic Class	1	2	3	UDS - SID
Diagnostic Transport Protocol Requirements				
Single frame transport only	+			
Full transport protocol (multi-segment)		+	+	
Required Configuration Services				
Assign frame identifier range	+	+	+	0xB7
Read by identifier (0 = product id)	+	+	+	0xB2 0x00
Read by identifier (all others)	optional	optional	+	0xB2 0xXX
Assign NAD	optional	optional	optional	0xB0
Conditional change NAD	optional	optional	optional	0xB3
Pos. response on supported config. services	+	+	+	SID + 0x40
Required UDS Services				
Read data by identifier:				0x22
– hardware and software version		+	+	0x22
– hardware part number (OEM specific)		+	+	0x22
– diagnostic version		+	+	0x22
Read by identifier (parameters)		+	+	0x22
Write by identifier (parameters)		if applicable	if applicable	0x2E
Session control			+	0x10
Read by identifier (sensor and actuator data)			+	0x22
I/O control by identifier			+	0x2F
Read and clear DTC (fault memory)			+	0x19, 0x14
Routine control			if applicable	0x31
Other diagnostic services			if applicable	...
Flash Reprogramming Services				
Flash programming services			optional	0xXX

Transport Layer

- **Class 2 + 3** need TP
- **Class 3** has more Diagnostic Services

from LIN 2.1 on

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

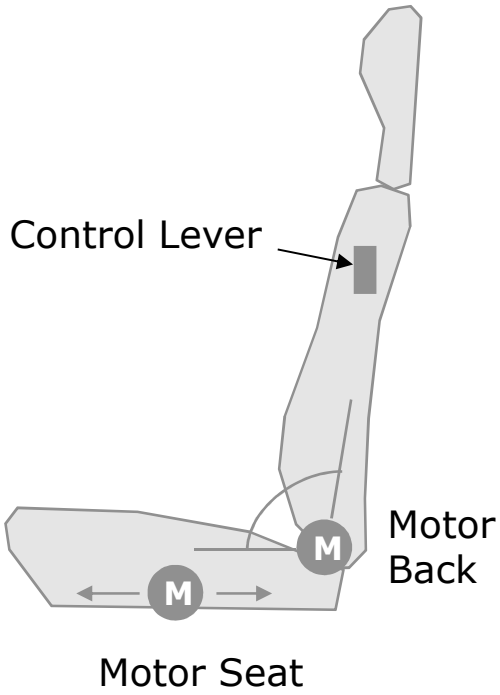
Agenda

	Information	4
	Overview	6
	LIN Workflow	9
	LIN Physical Layer	12
	LIN Communication	14
	Synchronization of the LIN nodes	16
	LIN Message & Scheduling	19
	Message Types	27
	Status & Network Management	30
	Introduction to LIN slave diagnostics	34
>	**Introduction to LIN slave configuration**	**39**
	Further Information	48

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

Introduction to LIN slave configuration

Configuration of a LIN Cluster: Before NAD allocation



LIN

LIN-Slave
MotorSeat
LIN 2.1
Initial NAD: 0x01
LPI: 6000/1/0

LIN Slave
MotorSeatBack
LIN 2.1
Initial NAD: 0x02
LPI: 6000/1/0

LIN Slave
ControlLever
LIN 2.0
Initial NAD: 0x01
LPI: 5000/0/0

LIN Master
BodyEcu
Master

CAN

NCF/LDF

Properties "Motor"

Supplier ID = 0x6000
Function ID = 0x0001
Variant = 0x00

initial NAD = 0x01, 0x02

NCF/LDF

Properties "ControlLever"

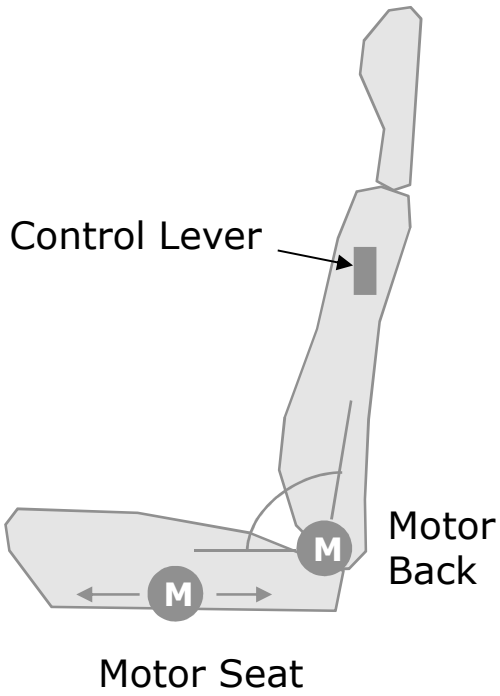
Supplier ID = 0x5000
Function ID = 0x0000
Variant = 0x00

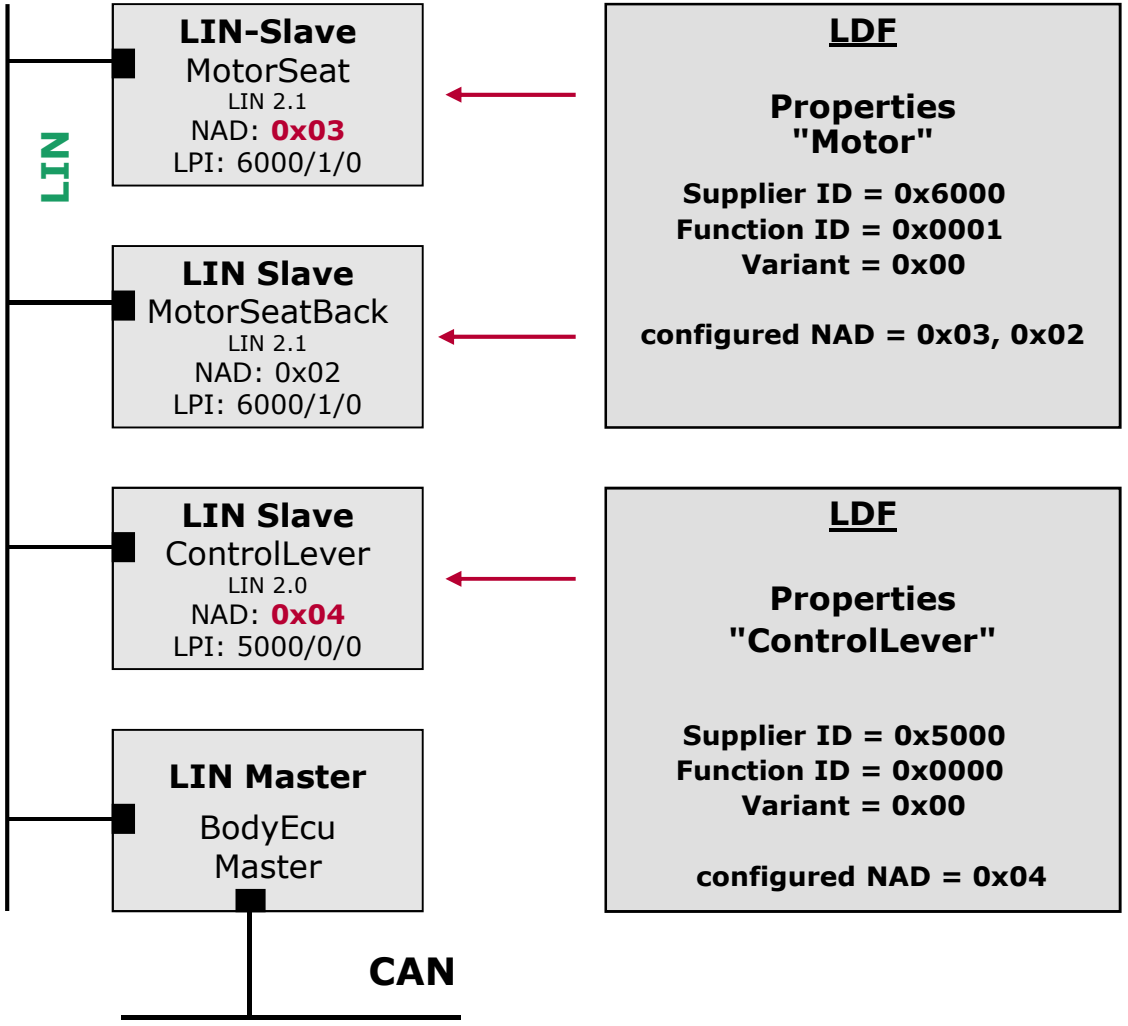
initial NAD = 0x01

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

Introduction to LIN slave configuration

Configuration of a LIN Cluster: After NAD allocation

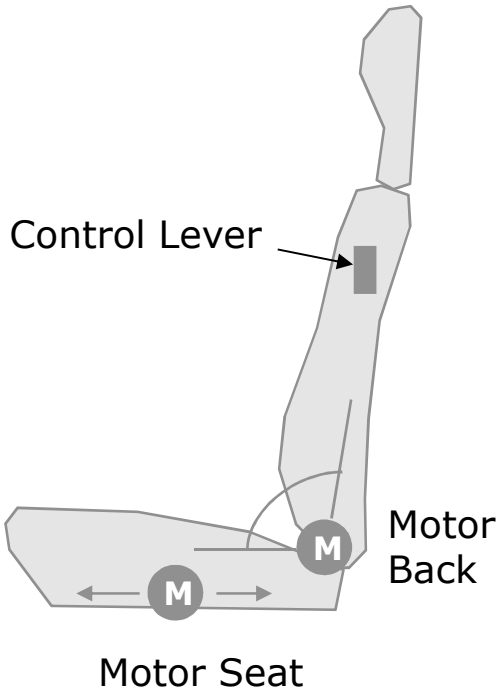


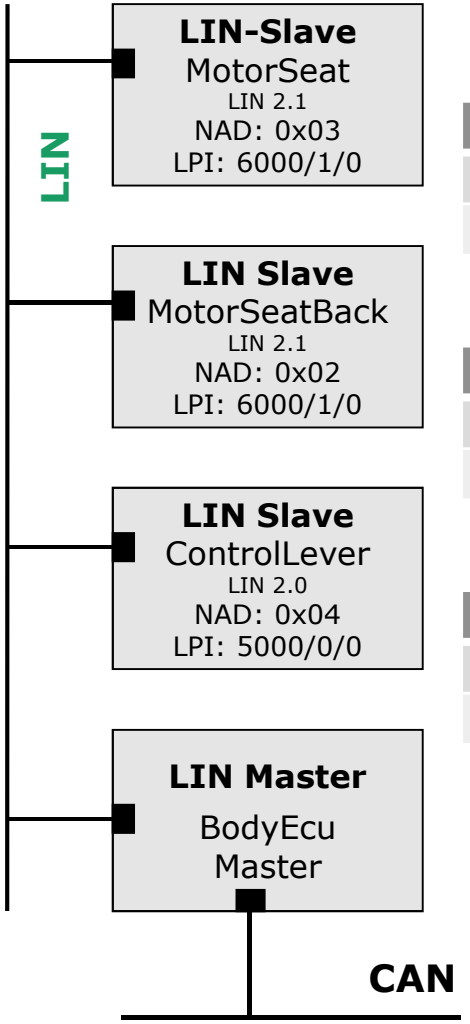


© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

Introduction to LIN slave configuration

Configuration of a LIN Cluster: Frame ID



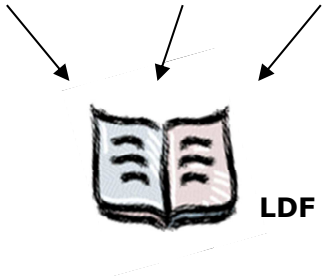


Configurable frames:

NAD 0x03	Msg Idx	ID	PID
MotorControl_Seat	0	0x13	0xD3
MotorState_Seat	1	0x14	0x14

NAD 0x02	Msg Idx	ID	PID
MotorControl_SeatBack	0	0x15	0x55
MotorState_SeatBack	1	0x16	0xD6

NAD 0x04	Msg Idx	ID	PID
ControlLeverControl	0x3000	0x17	0x97
ControlLeverState	0x4000	0x18	0xD8



© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

Diagnostic services overview

- LIN Diagnostics is based on services from the UDS specification
- Configuration frames are unsegmented (Single Frames)
- LIN Slaves are configurable since LIN2.0 (“Off-the-shelf-nodes”)
- The following services are used for LIN slave configuration:
  - 0xB0 - Assign NAD
  - 0xB1 - Assign Frame ID (LIN 2.0)
  - 0xB2 – Read by Identifier
  - 0xB3 – Conditional change NAD (not part of ISO17987 anymore)
  - 0xB4 – Data Dump
  - 0xB5 – Assign NAD via SNPD (Slave Node Position Detection)
  - 0xB6 – Save Configuration (since LIN2.1)
  - 0xB7 – Assign Frame ID Range (since LIN2.1)

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

Assign NAD (0xB0)

Request	Initial NAD	0x06	**0xB0**	Supplier ID (LSB)	Supplier ID (MSB)	Function ID (LSB)	Function ID (MSB)	New NAD
Positive Response	Initial NAD	0x01	**0xF0**	0xFF	0xFF	0xFF	0xFF	0xFF
Negative Response	Initial NAD	0x03	0x7F	SID=0xB0	Error Code	0xFF	0xFF	0xFF

Parameter	Wildcard
NAD	0x7F
Supplier ID	0x7FFF
Function ID	0xFFFF

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

Introduction to LIN slave configuration

Read by Identifier

Diagnostic request

NAD	PCI	SID	D1	D2	D3	D4	D5
NAD	0x06	**0xB2**	ID	Supl ID (LSB)	Supl ID (MSB)	Fct ID (LSB)	Fct ID (MSB)

ID	Meaning	Number of Bytes in Response
0	**Product Identification**	**RSID + 5 Bytes**
1	**Serial Number**	**RSID + 4 Bytes**
2-15	**Reserved**	**-**
16-31	**Message ID 1-16 (LIN 2.0)**	**RSID + 3 Bytes**
32-63	**User Defined**	**User Defined**
64-255	**Reserved**	**-**

	ID	NAD	PCI	RSID	D1	D2	D3	D4	D5	
Diagnostic response	0	NAD	0x06	**0xF2**	Supl ID (LSB)	Supl ID (MSB)	Fct ID (LSB)	Fct ID (MSB)	Variant	
Diagnostic response	1	NAD	0x05	**0xF2**	Serial (LSB)	Serial	Serial	Serial (MSB)	0xFF	
Diagnose response	16-31	NAD	0x04	**0xF2**	Msg ID1 (LSB)	Msg ID1 (MSB)	Current PID	0xFF	0xFF	LIN 2.0
Diagnose response	32-63	NAD	0x0X	**0xF2**	user defined	user defined	user defined	user defined	user defined	

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

Save configuration

Request	NAD	0x01	**0xB6**	0xFF	0xFF	0xFF	0xFF	0xFF
Positive response	NAD	0x01	**0xF6**	0xFF	0xFF	0xFF	0xFF	0xFF
Negative response	NAD	0x03	0x7F	SID=0xB6	Error code	0xFF	0xFF	0xFF

Node configuration API im LIN slave

- **`ld_read_configuration (&data, &length)`**
  - After receipt of the "Save configuration" commands the application can read out the current configuration of the LIN driver. The LIN driver copies the current NAD and all PIDs to the "data" parameter. The data can then be saved in nonvolatile memory.
- **`ld_set_configuration (&data, length)`**
  - After a reset the application can pass the saved configuration to the LIN driver. The LIN driver then accepts the passed NAD and PIDs.

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

Assign frame ID range

Request	NAD	0x06	**0xB7**	Start index	PID (index)	PID (index+1)	PID (index+2)	PID (index+3)
Positive response	NAD	0x01	**0xF7**	0xFF	0xFF	0xFF	0xFF	0xFF
Negative response	NAD	0x03	0x7F	SID=0xB7	Error code	0xFF	0xFF	0xFF

Example

```
Frames {
  MotorControl_Seat: 0x13 ...
  MotorState_Seat:   0x14 ...
}
Node_attributes {
  MotorSeat {
    ...
    configurable_frames
    {
      MotorControl_Seat;
      MotorState_Seat;
    }
  }
}
```

	NAD	PCI	SID	D1	D2	D3	D4	D5
Request	NAD	0x06	0xB7	0	0xD3	0x14	0xFF	0xFF

Identifier 0x13 -> PID 0xD3
Identifier 0x14 -> PID 0x14
Not to be changed -> 0xFF

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

Agenda

Information	4
Overview	6
LIN Workflow	9
LIN Physical Layer	12
LIN Communication	14
Synchronization of the LIN nodes	16
LIN Message & Scheduling	19
Message Types	27
Status & Network Management	30
Introduction to LIN slave diagnostics	34
Introduction to LIN slave configuration	39
> Further Information	**48**

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

Further Information

- LIN Information & Specification
  - http://www.vector.com/
  - http://www.vector.com/vi_training_elearning_en.html
  - http://vector.com/vi_lin_spec_download_en.html



- LIN Webinar schedule
  - Introduction to LIN
  - Setting up the Vector Embedded Software for LIN ECUs:
    Tuesday, October 14, 2014 9:00 am, Europe Summer Time (Berlin, GMT+02:00)
    Tuesday, October 14, 2014 4:00 pm, Korea Time (Seoul, GMT+09:00)
    Tuesday, October 14, 2014 3:00 am, Eastern Daylight Time (New York, GMT-04:00)
  - Analysis and Testing of LIN ECUs:
    Tuesday, October 21, 2014 9:00 am, Europe Summer Time (Berlin, GMT+02:00)
    Tuesday, October 21, 2014 4:00 pm, Korea Time (Seoul, GMT+09:00)
    Tuesday, October 21, 2014 3:00 am, Eastern Daylight Time (New York, GMT-04:00)

© 2014 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.