



## Differences between LIN2.0 and LIN2.1

Vector LIN Symposium 2008  
Ernst Christmann, Vector Informatik

## > From Where Did We Start

Outward Appearance

Clarifications in LIN Version 2.1

Changes in Content

Event Triggered Frames

Slave Identification and Configuration

Diagnostics Specification

Quo Vadis LIN ?

- ❑ Its predecessor LIN 2.0 had been functionally mature.
- ❑ But yet there were some gaps in definitions, which needed clarification.
- ❑ Despite the intention to maintain stability on the technical side, also some functional improvements had to be made.
- ❑ Many practical experiences with LIN 2.0 within the LIN community were the basis for the new LIN 2.1 standard.
- ❑ LIN 2.1 was released in November 2006.

- ❑ Section numbering is continuous now
- ❑ Page numbering is continuous also
- ❑ Transport Layer specification has its own section (formerly in “Diagnostic and Configuration Specification”)
- ❑ Diagnostics specification section added
- ❑ Enhanced small print

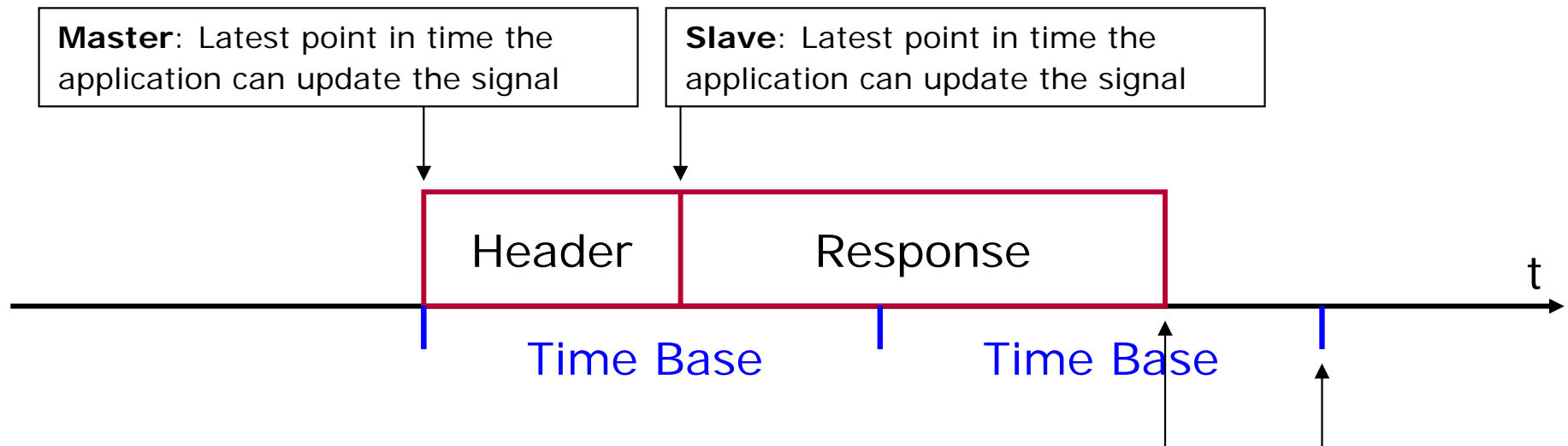
- ❑ Each signal has a unique source.
- ❑ It can be transferred in several frames.
- ❑ Byte arrays must still be mapped exactly to the frame bytes. But crossing of byte boundaries is not restricted to one per scalar signal any more.
- ❑ Byte arrays must be initialized with values in big-endian order and signal consistency must now also be granted.

# Clarifications in LIN Version 2.1

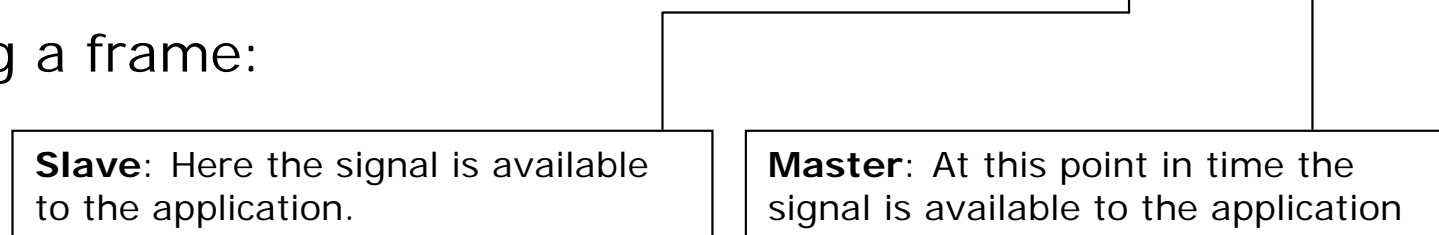
## Signal Reception and Transmission

### Clarification of timing:

Transmitting a frame:



Receiving a frame:



- ❑ Clarification of Frame Addresses:
  - ❑ 64 valid Protected Identifiers (PIDs)
    - ❑ ID 0 through ID 59 for unconditional frames
    - ❑ ID 60 Master Request Frame (MRF)
    - ❑ ID 61 Slave Response Frame (SRF)
    - ❑ ID **62** and 63 reserved (In LIN 2.0 ID 62 was usable for user specific applications.)

- ❑ “Go to Sleep”
  - ❑ on Master Request Frame with first response byte zero
  - ❑ bus is idle for at least 4 seconds (at the latest after 10 seconds)
  
- ❑ “Wake-up”
  - ❑ on a dominant level of 250  $\mu$ sec to 5 ms
  - ❑ up to three times with 150 – 250 ms spaces,
  - ❑ then a break of at least 1.5 sec
  
- ❑ “Power off” has been removed, because it actually meant “stop communication”

# Changes in Content

## Areas of Major Improvements in LIN Version 2.1

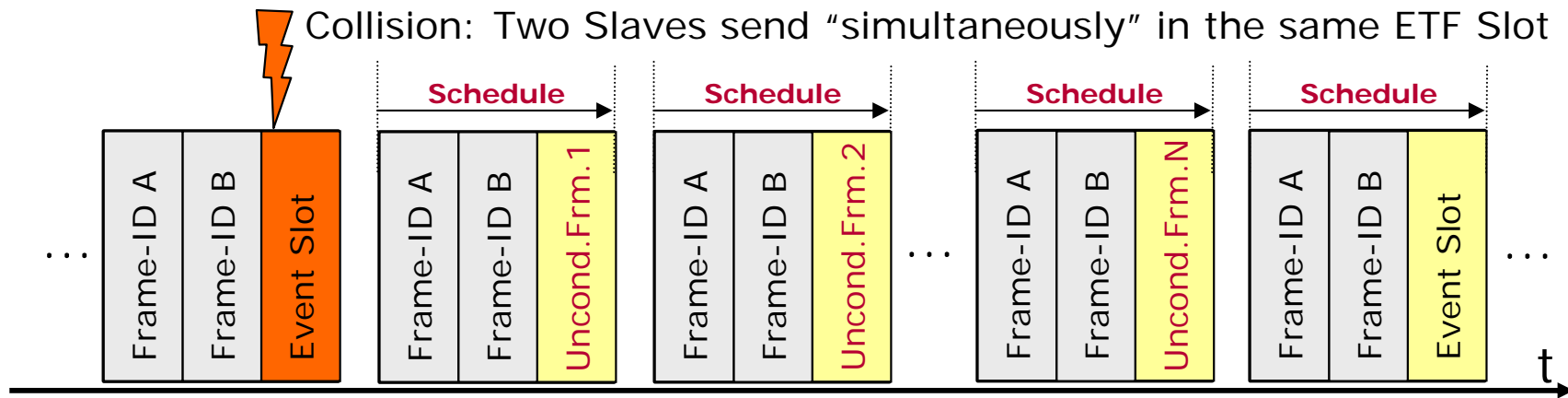
- ❑ Event Triggered Frames
- ❑ Slave Identification and Configuration
- ❑ Diagnostics

- ❑ Frames belonging to an event triggered frame cluster must:
  - ❑ ... not be sent in the same schedule as unconditional frames
  - ❑ ... have **equal length**
  - ❑ ... use the **same checksum model** (no mix of LIN1.x and 2.x slaves)
  - ❑ ... contain their **protected identifier (PID)** in the **first data byte**
  - ❑ ... be published by different slave nodes

# Event Triggered Frames

## Event Triggered Frame Collision Resolution in Master Node

### Collision Resolution with LIN 2.0:

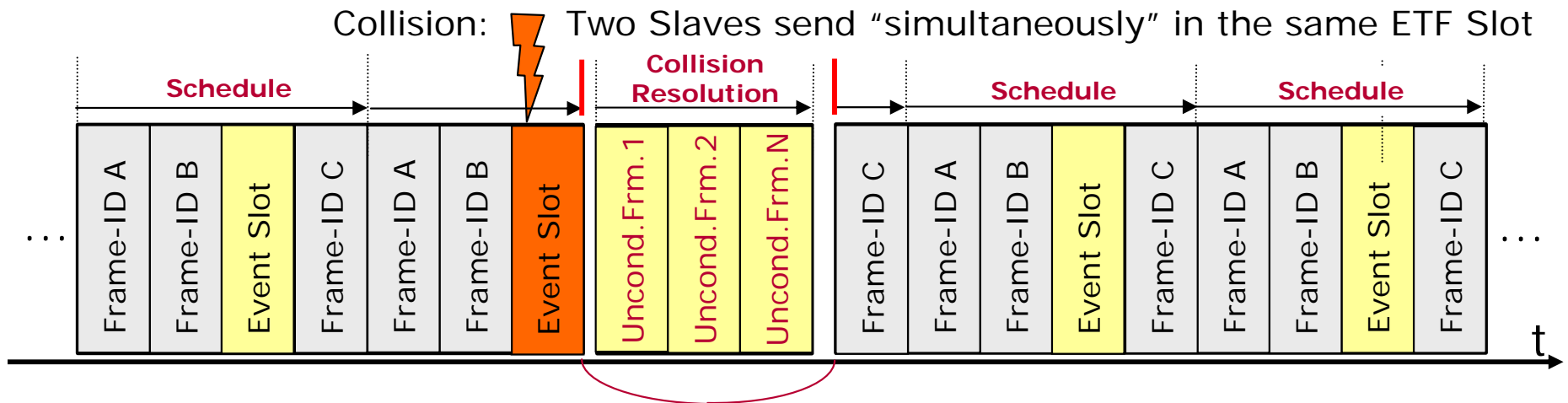


### Problem: Collision resolution may take too much time!

# Event Triggered Frames

## Event Triggered Frame Collision Resolution in Master Node

### Collision Resolution with LIN 2.1



Problem: Signal latencies may change

Note: The new resolution strategy includes the old one as a special case of tailoring the resolution schedule

# Event Triggered Frames

## Event Triggered Frame Collision Resolution in LDF

```
Event_triggered_frames
```

```
{  
    Event_Frame: Collision_Resolving_Schedule, 10,  
    uFrame_Slave1, uFrame_Slave2;  
}
```

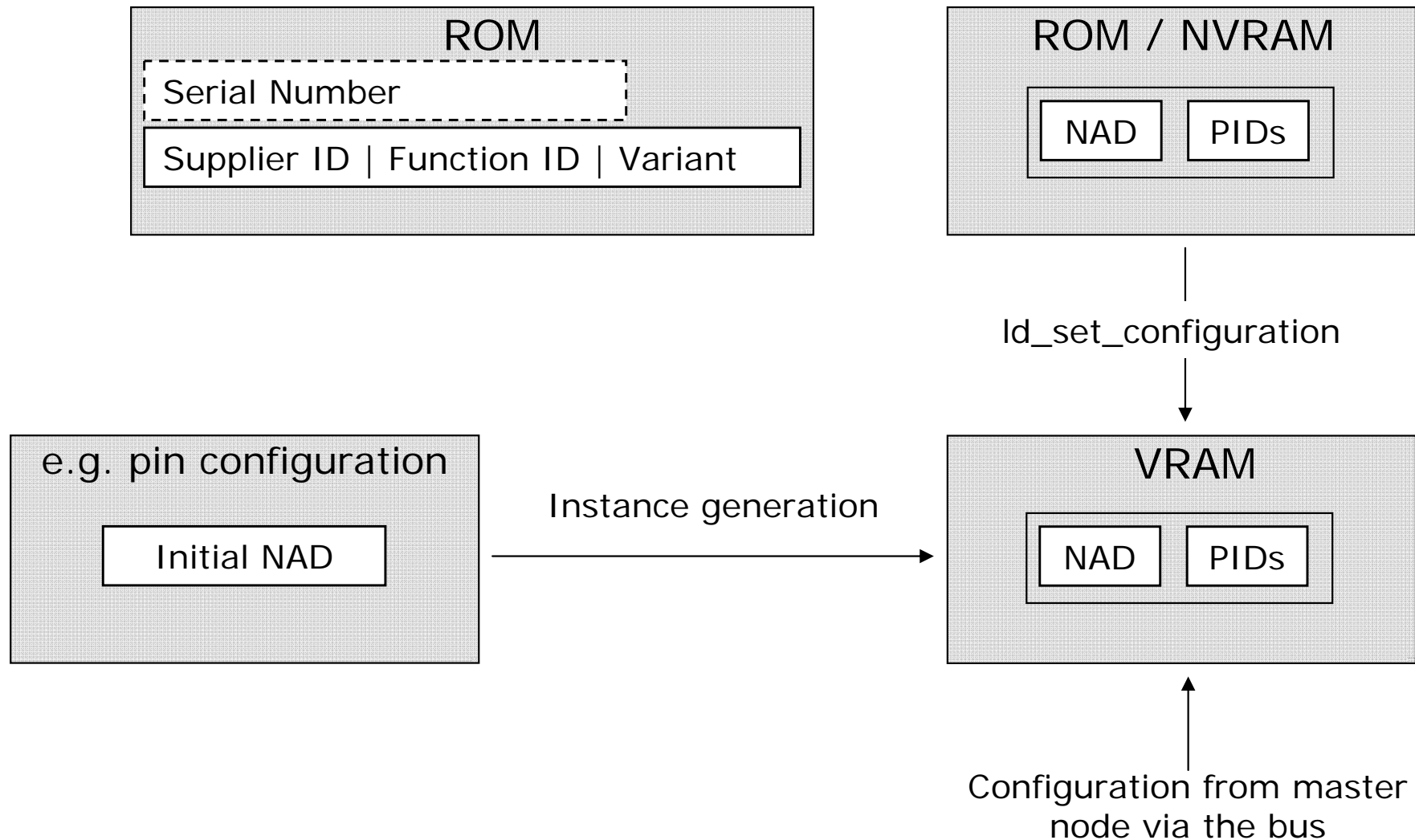
---

```
Schedule_tables
```

```
{  
    Normal_Table {  
        ...  
        Event_Frame          delay 20 ms ;  
        ...  
    }  
    Collision_Resolving_Schedule {  
        uFrame_Slave1      delay 20 ms ;  
        uFrame_Slave2      delay 20 ms ;  
    }  
}
```

# Slave Identification and Configuration

## Definition of a Slave Node Memory Model



# Slave Identification and Configuration

## Three Configuration Types for Slaves

### ❑ Not Configured Slave

- ❑ After every reset the slave node does not have a valid configuration. It must first be configured by the master. The configuration data is stored in VRAM.

### ❑ Preconfigured Slave

- ❑ The configuration data is normally stored in ROM. The slave node has a valid configuration after reset. But changes made to the configuration data will be lost after re-initialization.

### ❑ Fully Configurable Slave

- ❑ The slave node stores the configuration data in NVRAM, so it will be active and fully configured after reset.

# Slave Identification and Configuration

## New Service: Assign NAD via Slave Node Position Detection

- ❑ Service Identifier B5 introduced
- ❑ SID B5 means **Assign NAD via Slave Node Position Detection**
- ❑ Reason for introduction:  
Automate configuration of identical slaves after reset
- ❑ Problem:  
For legal reasons there is a delay in the development of a standard for SNPD
- ❑ Methods “Daisy Chain” and “Shunt” are in discussion

# Slave Identification and Configuration

## New Service: Save Configuration

- ❑ Service Identifier B6 **Save Configuration** introduced
- ❑ Reason for introduction:  
Slaves obeying to this Service need not be configured after reset.
- ❑ Master Request Frame:

<b>NAD</b>	<b>PCI</b>	<b>SID</b>	<b>Unused</b>				
NAD	0x01	0xB6	0xFF	0xFF	0xFF	0xFF	0xFF

- ❑ Slave Response Frame:

<b>NAD</b>	<b>PCI</b>	<b>SID</b>	<b>Unused</b>				
NAD	0x01	0xF6	0xFF	0xFF	0xFF	0xFF	0xFF

# Slave Identification and Configuration

## New Service: Assign Frame Identifier Range

- ❑ Service Identifier B7 **Assign Frame Identifier Range** introduced
- ❑ Reason for introduction:  
A LIN Network can be configured **up to four times faster**.

- ❑ Master Request Frame:

NAD	PCI	SID	D1	D2	D3	D4	D5
NAD	0x06	0xB7	Start index	PID (index)	PID (index+1)	PID (index+2)	PID (index+3)

- ❑ Slave Response Frame:

NAD	PCI	SID	Unused				
NAD	0x01	0xF7	0xFF	0xFF	0xFF	0xFF	0xFF

# Slave Identification and Configuration

## Consequences of Service "Assign Frame Identifier Range"

- ❑ Service Identifier B1 **Assign Frame Identifier** "Removed"
  - ❑ Reason: To configure a LIN network by individually assigning frame IDs may take quite long causing inertia effects.
  - ❑ In mixed networks with 2.0 slaves SID B1 is still needed!
- ❑ The two byte "**Message Identifiers**" removed from node attribute "Configurable Frames" in LDF
- ❑ Hence diagnostic service B2 "**Read by Identifier**" changed:

Identifier	Interpretation	Length of Response
0	LIN Product Identification	5 + RSID
1	Serial Number	4 + RSID
2 - 15	Reserved	–
16 - 31	<u>Msg IDs 1-16</u> <b>Reserved</b>	3 + RSID
32 - 63	User defined	User defined
64 - 255	Reserved	–

- ❑ The section was completely revised. The objective was mainly to increase the precision of the definitions, but not to introduce new functionality.
  
- ❑ There are two main focuses
  - ❑ Definition of diagnostic classes
  - ❑ Improved definition and reworked implementation requirements of the Multi Frame Transport Layer protocol
  
- ❑ Yet functional addressing was added.

- ❑ Smart and simple devices (intelligent sensors, actuators)
  - ❑ Fault indication is always signal based.
  - ❑ Fault memory handling is done by the master node.
  - ❑ Specific diagnostic support is not required.
  - ❑ Diagnostic frames usage is mainly limited to node configuration.
- ➔ Single frame transport protocol support suffices.

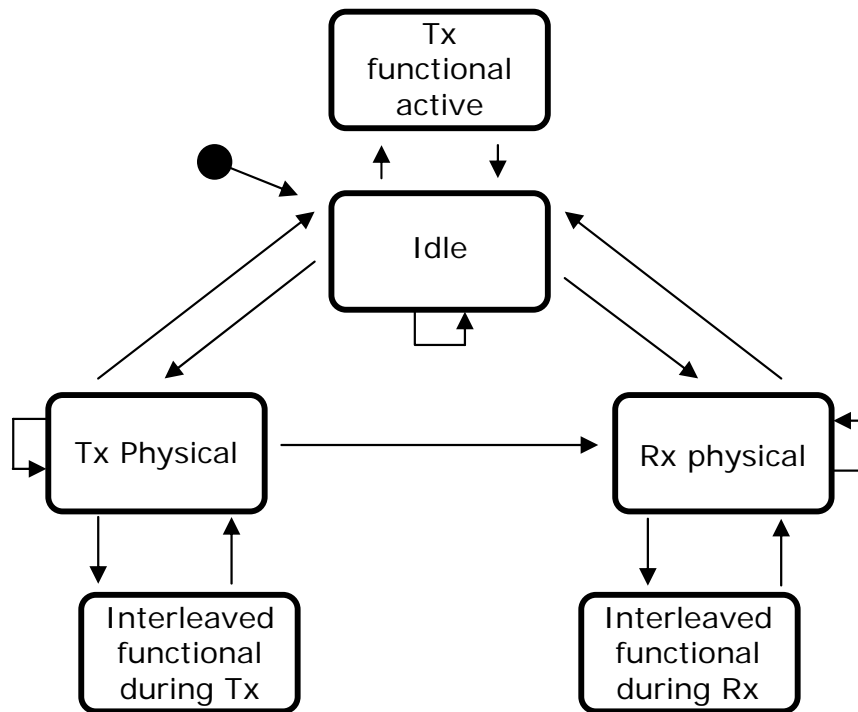
- ❑ Class II slave nodes provide node identification support.
    - ❑ E.g. part number is required by vehicle manufacturers. More specific diagnostic support for these tasks is not required.
  - ❑ Fault indication is always signal based.
  - ❑ Fault memory handling is done by the master node.
  - ❑ Provided diagnostic services (UDS):
    1. Node identification (SID 0x22) defined by the user e.g. reading HW and SW version, HW part number, diagnostic version
    2. Reading data parameter (SID 0x22) if applicable. Data parameter means: every data that can be read from the ECU, e.g. oil temperature, vehicle speed
    3. Writing parameters (SID 0x2E) if applicable
- ➔ Full transport layer required to support multi-frame transmissions.

- ❑ Class III slave nodes require extended diagnostic support.
  - ❑ Internal fault memory, along with associated reading and clearing services.
  - ❑ Optionally, reprogramming (flash/eeprom reprogramming) of the slave node. This requires an implementation of a LIN boot-loader.
- ❑ Services are implemented according standard ISO 14229-1 (UDS).
  
- ➔ Full transport layer required to support multi-frame transmissions.
- ➔ UDS component required for session handling.

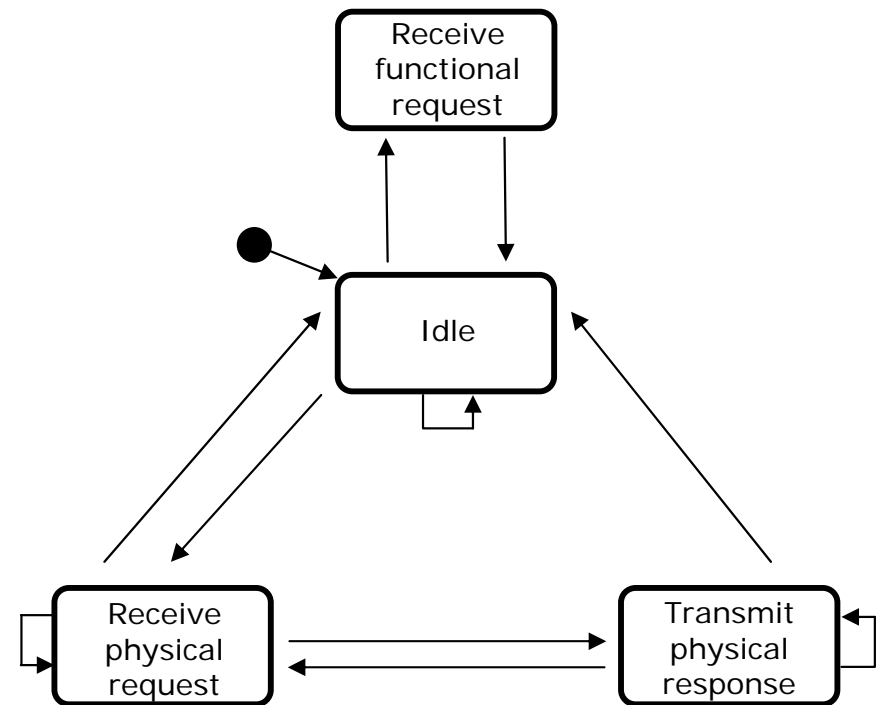
- Functional addressing supported (NAD 0x7E)

NAD Value	Description
0	Reserved for go to sleep command
1 – 125 (0x7D)	Slave Node Addresses (NAD)
126 (0x7E)	Functional node address (functional NAD), only used for diagnostics, using the transport layer
127 (0x7F)	Slave node address broadcast (broadcast NAD)
128 (0x80) – 255 (0xFF)	Free Usage. Diagnostic frames with the first byte in the range 128 (0x80) to 255 (0xFF) are allocated for free usage since the LIN 1.2 standard.

- Resulting states of the transport layer communication handler:



Master node transmission handler



Slave node transmission handler

- ❑ Test specification still outstanding, but is expected by mid 2008.
  
- ❑ With LIN 2.1 the main objectives were achieved:
  - ❑ A stable, future-proof standard
  - ❑ Easy to be used
  - ❑ Clear and concise in its statements
  - ❑ Full backward compatibility  
(except for the “removal” of the configuration service “Assign Frame ID”)
  
- ❑ The LIN community has contributed a lot and has helped with experience to make LIN a good and solid success.

Thank you for your attention.

For detailed information about Vector  
and our products please have a look at:

[www.lin-solutions.com](http://www.lin-solutions.com)

**Author:**

Ernst Christmann, Sven Hesselmann

Vector Informatik GmbH

Ingersheimer Str. 24

70499 Stuttgart