

# LIN Master Driver for Black Box Conformance Test

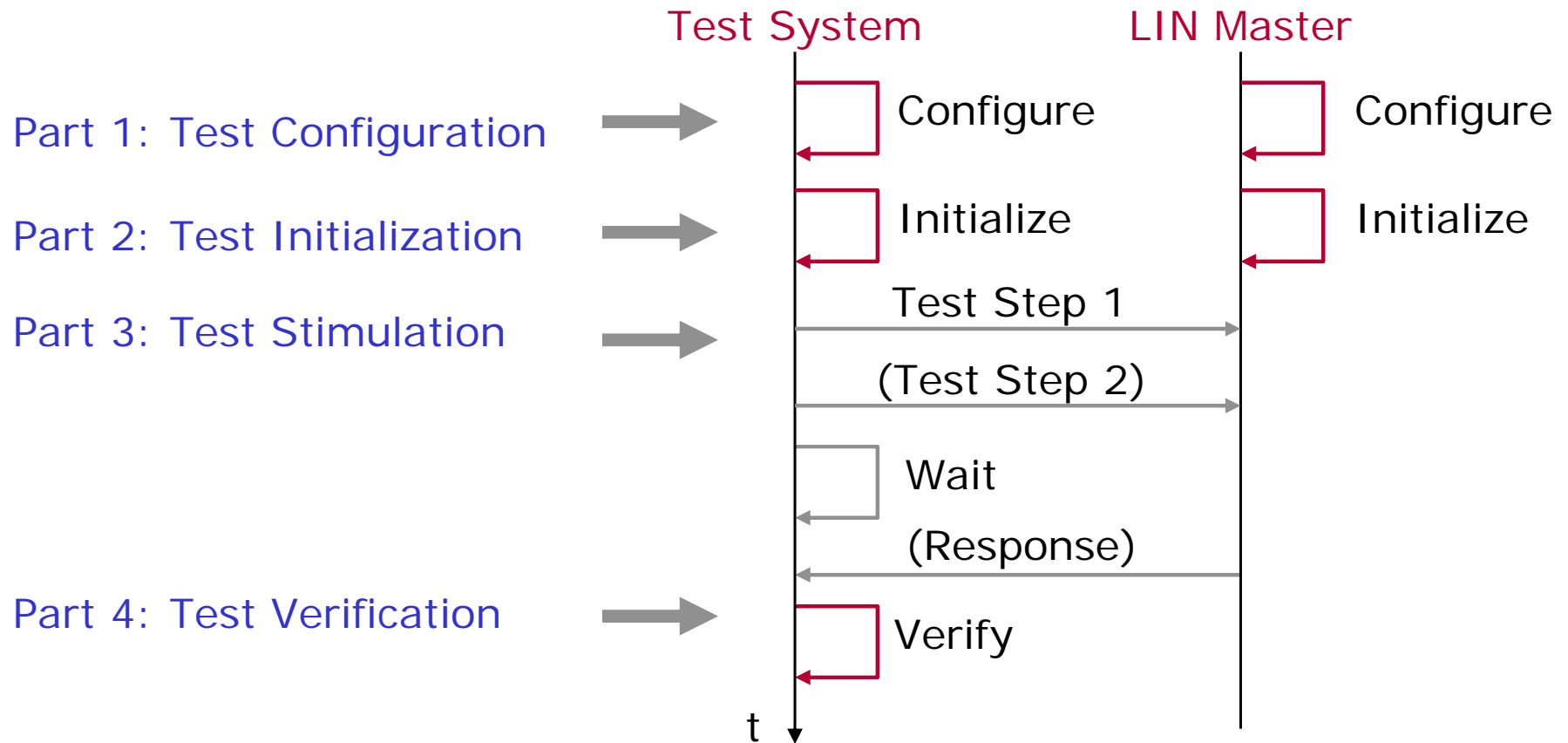
Vector LIN Symposium 2008  
Andreas Pick, Vector Informatik

# Agenda

- ❑ Introduction
- ❑ Gray Box Master Test
- ❑ Black Box Master Test
  - ❑ Goals
  - ❑ Concept
  - ❑ Implementation
- ❑ Conclusion

# Introduction

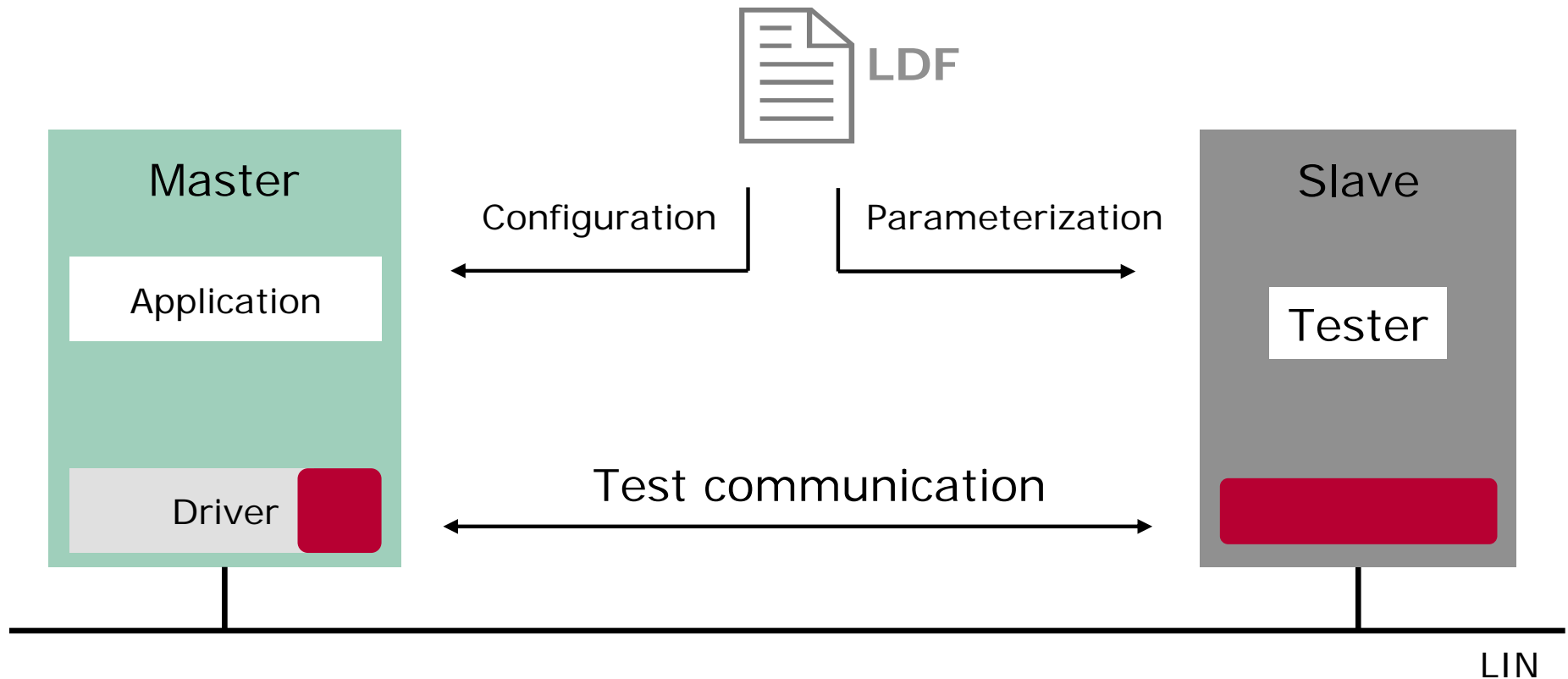
- ❑ LIN Master conformance test specification
  - ❑ Defines configuration, initialization and verification
  - ❑ Does not however define a method of stimulation



- ❑ Current implementation of Master test
  - ❑ Stimulation via test application
  - ❑ Test application controlled via LIN by tester (Slave)
  
- + Test application suitable for all Master driver implementations
- + Full coverage of tests within short time
  
- Specific test LDF and test application necessary
- Master driver configured according to test LDF
- Application and LIN driver configuration will be changed after test

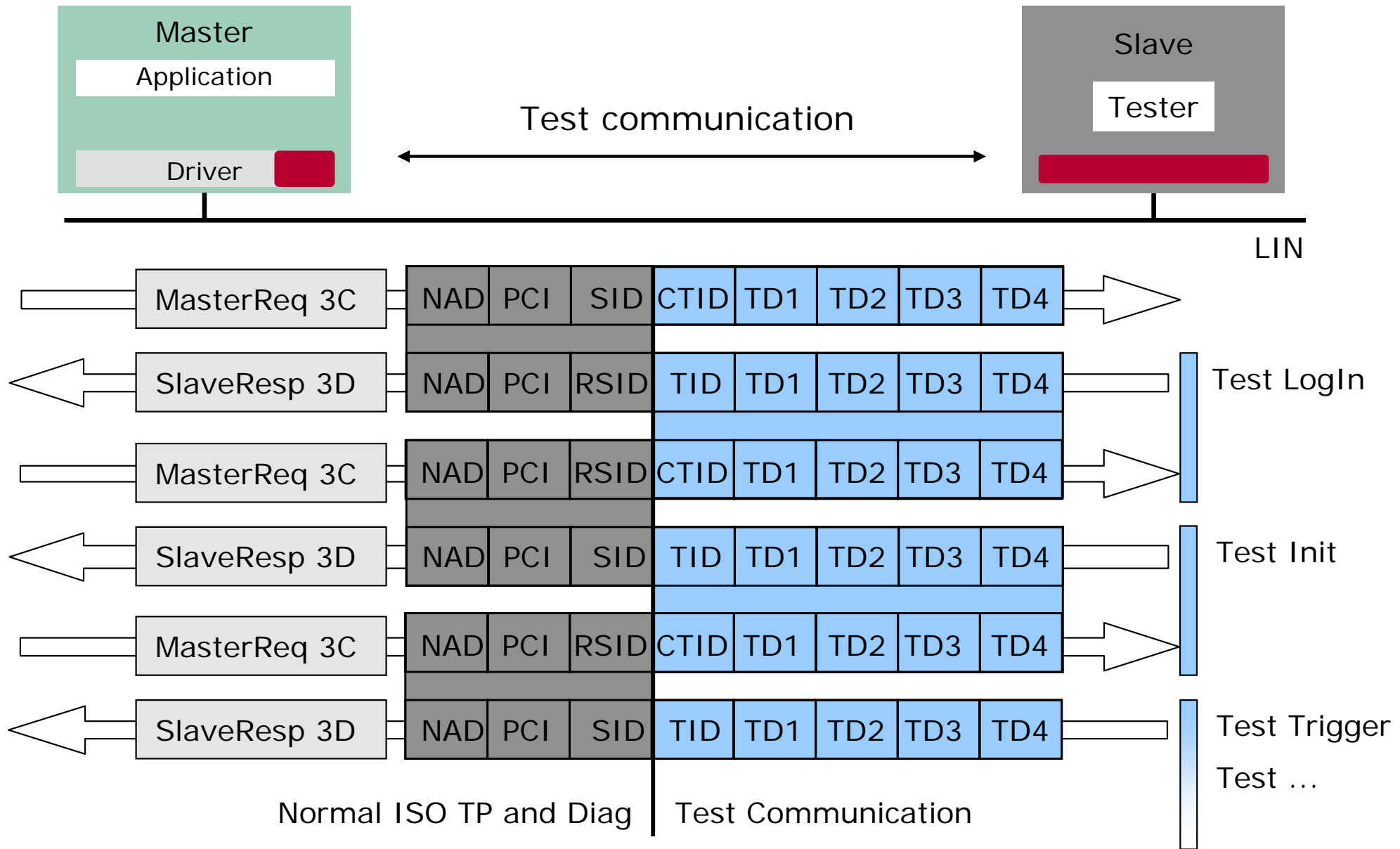
- ❑ Main goals for an improved Master test
  - ❑ Single source ECU stack for test and series production
  - ❑ Code for test support must be removable before production
  - ❑ No conflict with existing Master functionality
  - ❑ Normal communication must not be disturbed
  
- ❑ Further goals
  - ❑ No or little adaptation of application
  - ❑ Minimal test preparation time
  - ❑ Minimal resources

# Black Box LIN Master Test - Concept



- ❑ Test communication
  - ❑ Controlled by Tester (Slave) not LIN Master
  - ❑ MasterReq and SlaveResp used for TP Single Frame communication
  - ❑ New diagnostic service is required

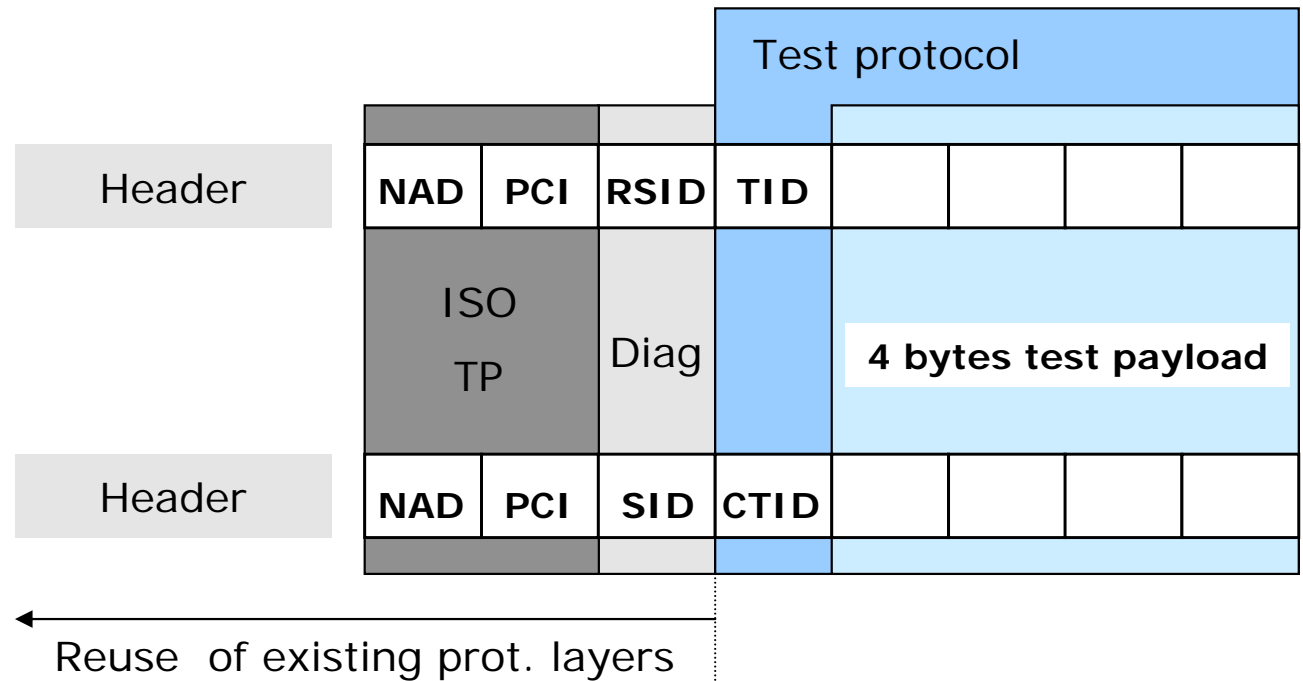
# Black Box LIN Master Test - Concept



# Black Box LIN Master Test - Concept

## Communication Format

- ❑ Tester Command
  - ❑ SlaveResp 0x3D
- ❑ Master Confirmation
  - ❑ MasterReq 0x3C



- ❑ Test function addressing based on existing Diag communication structure
  - ❑ Test identifier – **TID**
  - ❑ Positive confirmation test identifier – **CTID** = TID + 0x40
  - ❑ Negative confirmation test identifier – **CTID** = 0x7F

# Black Box LIN Master Test - Implementation

## Test command interface

	Parameters	Test ID								
Test LogIn	none	0x00								
Tester Command	<table border="1"><tr><td>NAD</td><td>PCI</td><td>F8</td><td>00</td><td>FF</td><td>FF</td><td>FF</td><td>FF</td></tr></table>	NAD	PCI	F8	00	FF	FF	FF	FF	
NAD	PCI	F8	00	FF	FF	FF	FF			
		unused								
Master Confirmation	<table border="1"><tr><td>NAD</td><td>PCI</td><td>B8</td><td>40</td><td>FF</td><td>FF</td><td>FF</td><td>FF</td></tr></table>	NAD	PCI	B8	40	FF	FF	FF	FF	
NAD	PCI	B8	40	FF	FF	FF	FF			

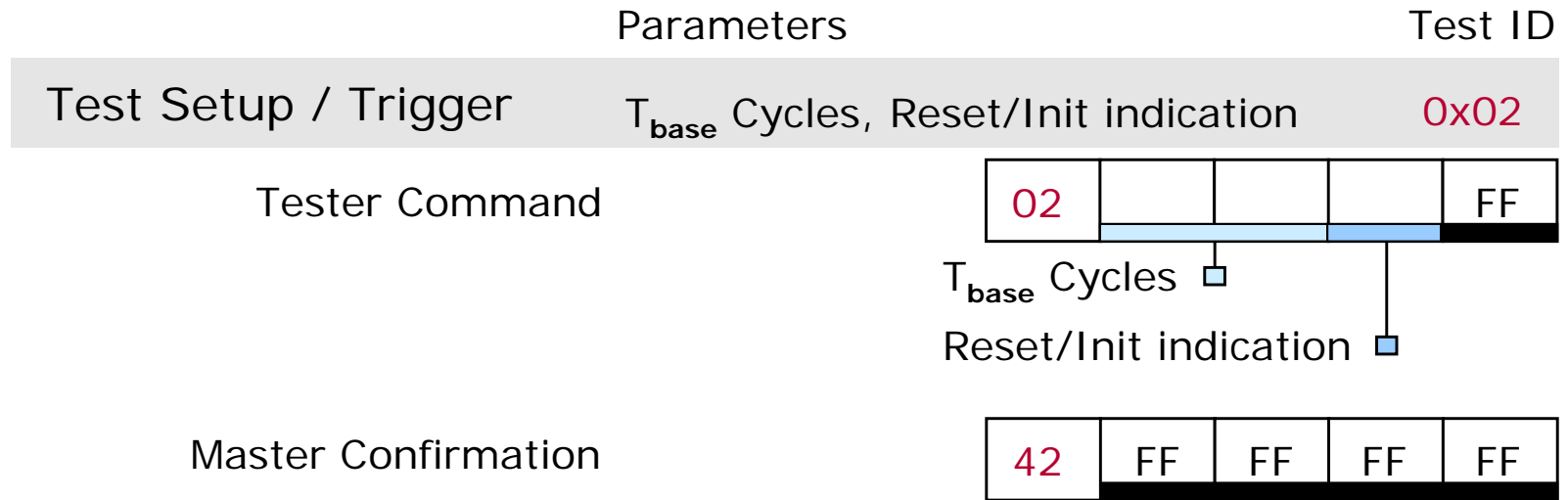
- ❑ Master waiting for LogIn command of tester

Test Presence / Idle	none	0x01					
Tester Command		<table border="1"><tr><td>01</td><td>FF</td><td>FF</td><td>FF</td><td>FF</td></tr></table>	01	FF	FF	FF	FF
01	FF	FF	FF	FF			
		unused					
Master Confirmation		<table border="1"><tr><td>41</td><td>FF</td><td>FF</td><td>FF</td><td>FF</td></tr></table>	41	FF	FF	FF	FF
41	FF	FF	FF	FF			

- ❑ Executed in idle status indicating tester presence

# Black Box LIN Master Test - Implementation

## Test command interface



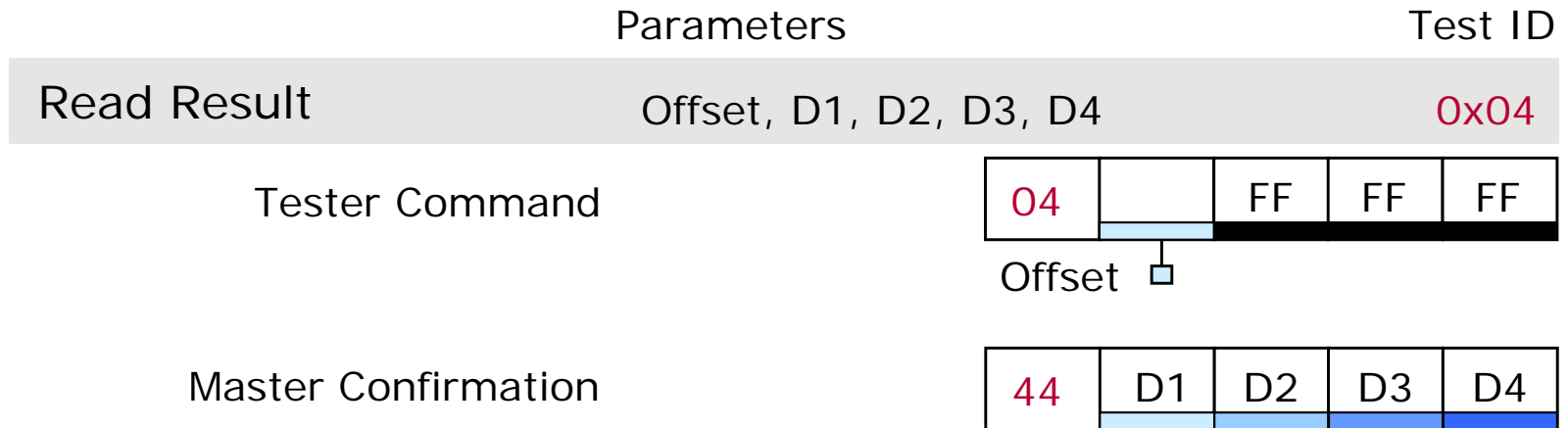
- ❑ Set test cycles to specify test time or indicate reset



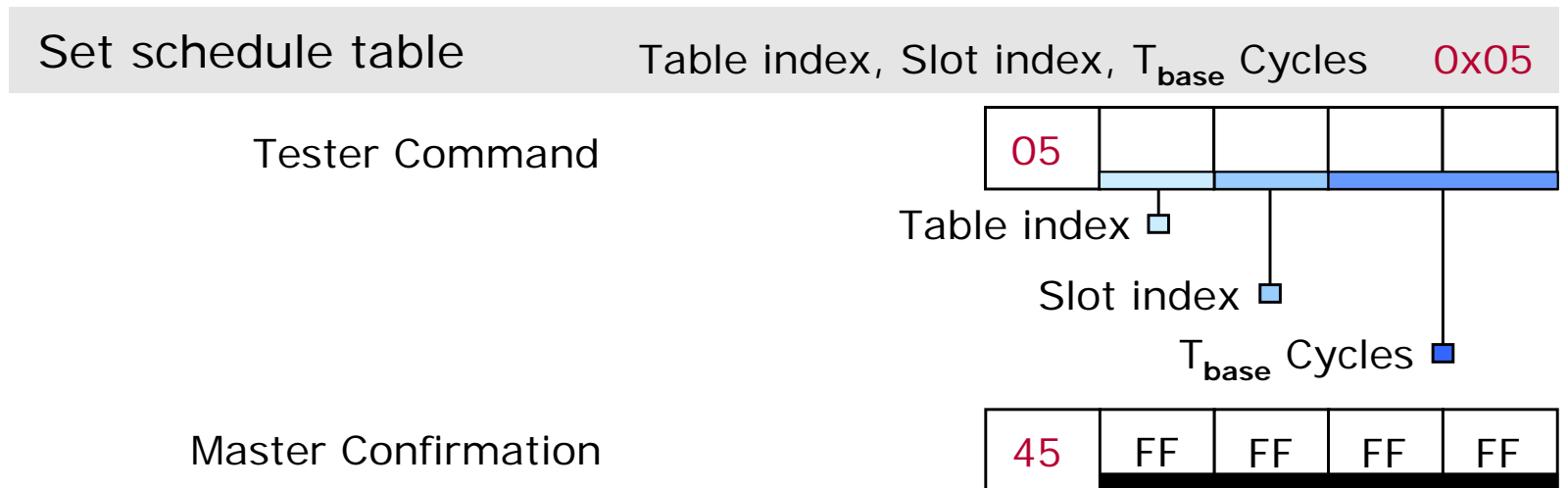
- ❑ D1-D3 contain additional parameters
- ❑ Stored in linear test buffer, Offset used for addressing buffer location

# Black Box LIN Master Test - Implementation

## Test command interface



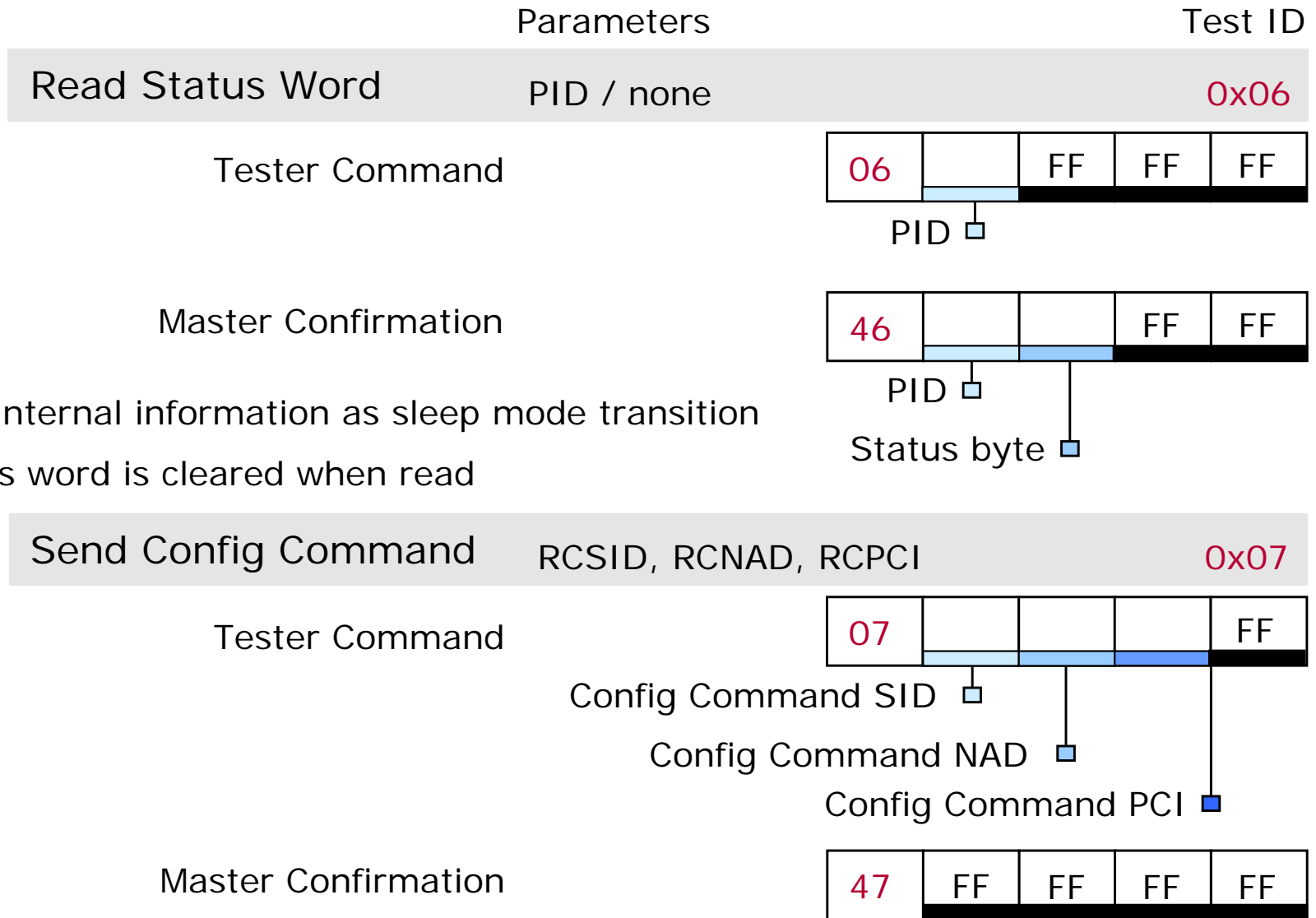
- Results read from linear test buffer, Offset used for addressing buffer location



- Results read from linear test buffer, Offset used for addressing buffer location

# Black Box LIN Master Test - Implementation

## Test command interface

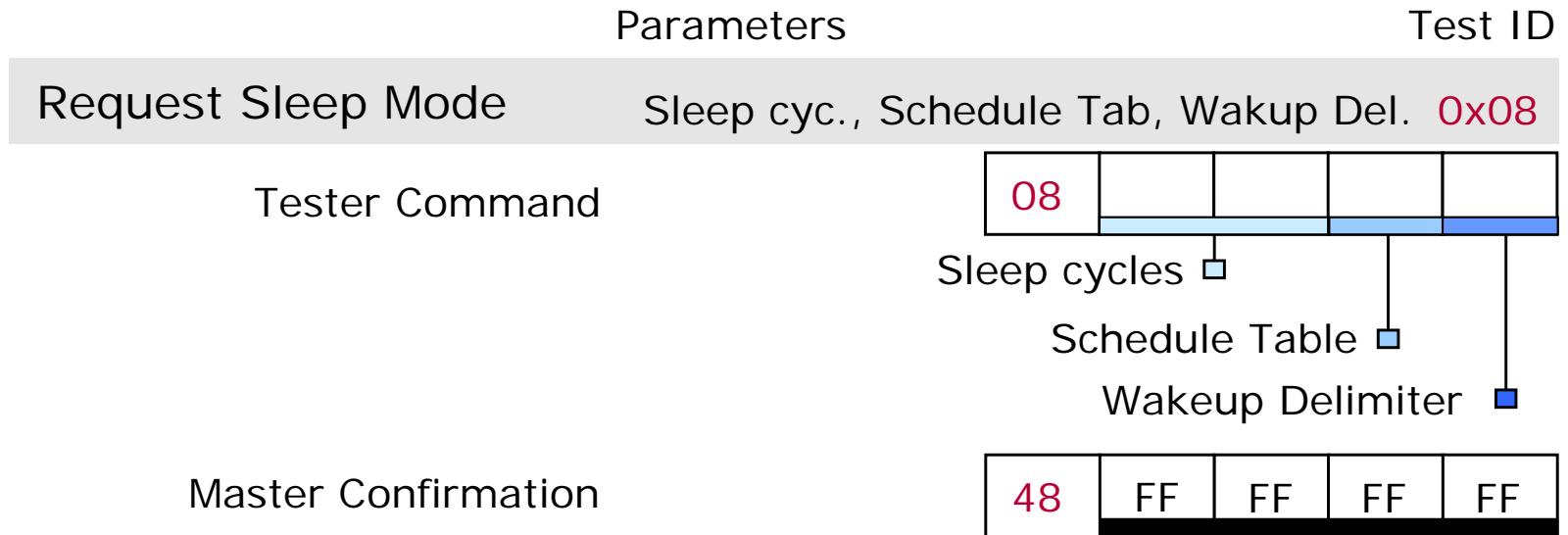


- ❑ Provides internal information as sleep mode transition
- ❑ The status word is cleared when read

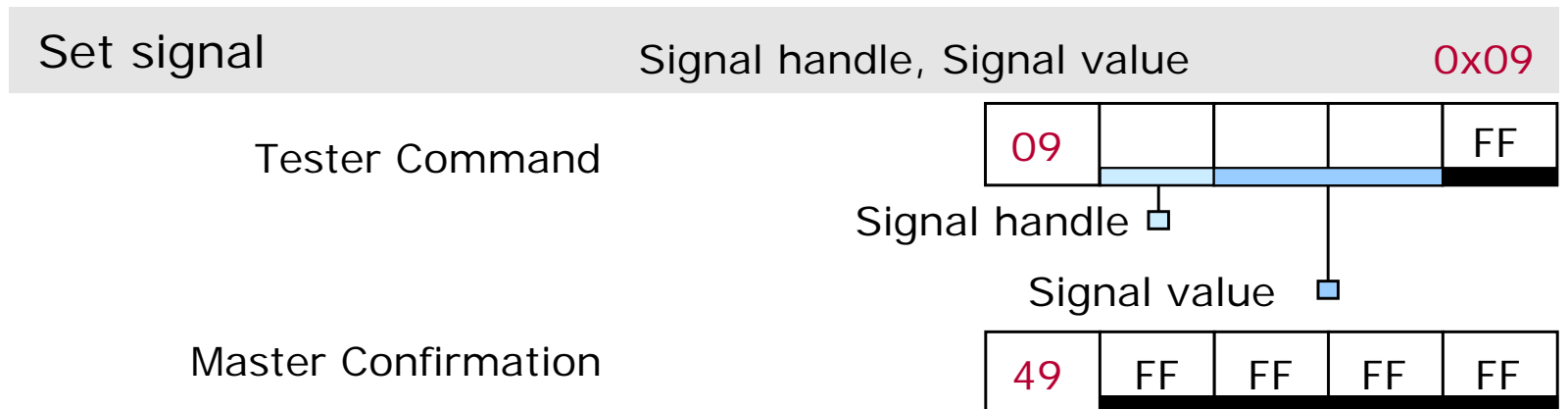
- ❑ According to command "Load Parameters" command might be used

# Black Box LIN Master Test - Implementation

## Test command interface



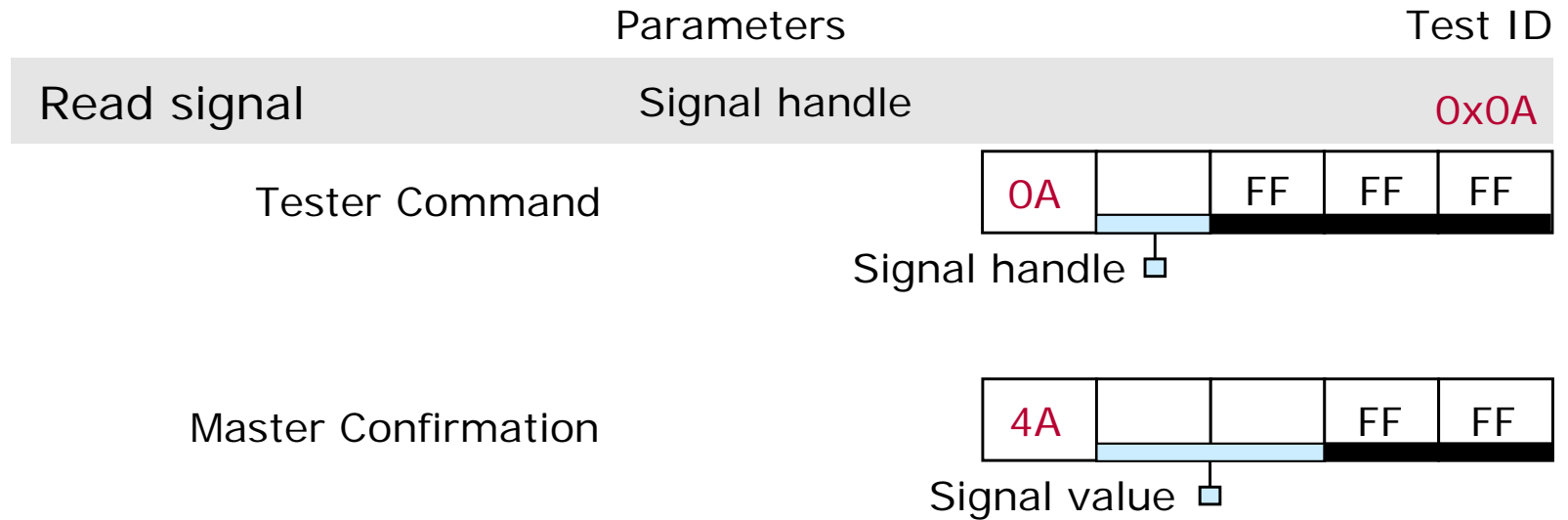
- Wakeup delimiter of 0 specifies to wait for external wake up



- Signal handle is derived from LDF signal order

# Black Box LIN Master Test - Implementation

## Test command interface



- ❑ For byte arrays "Read Result" service can be used
- ❑ Test services 0x00 – 0x0A fully cover LIN 2.0 conformance test specification
- ❑ Additional services might cover new test cases

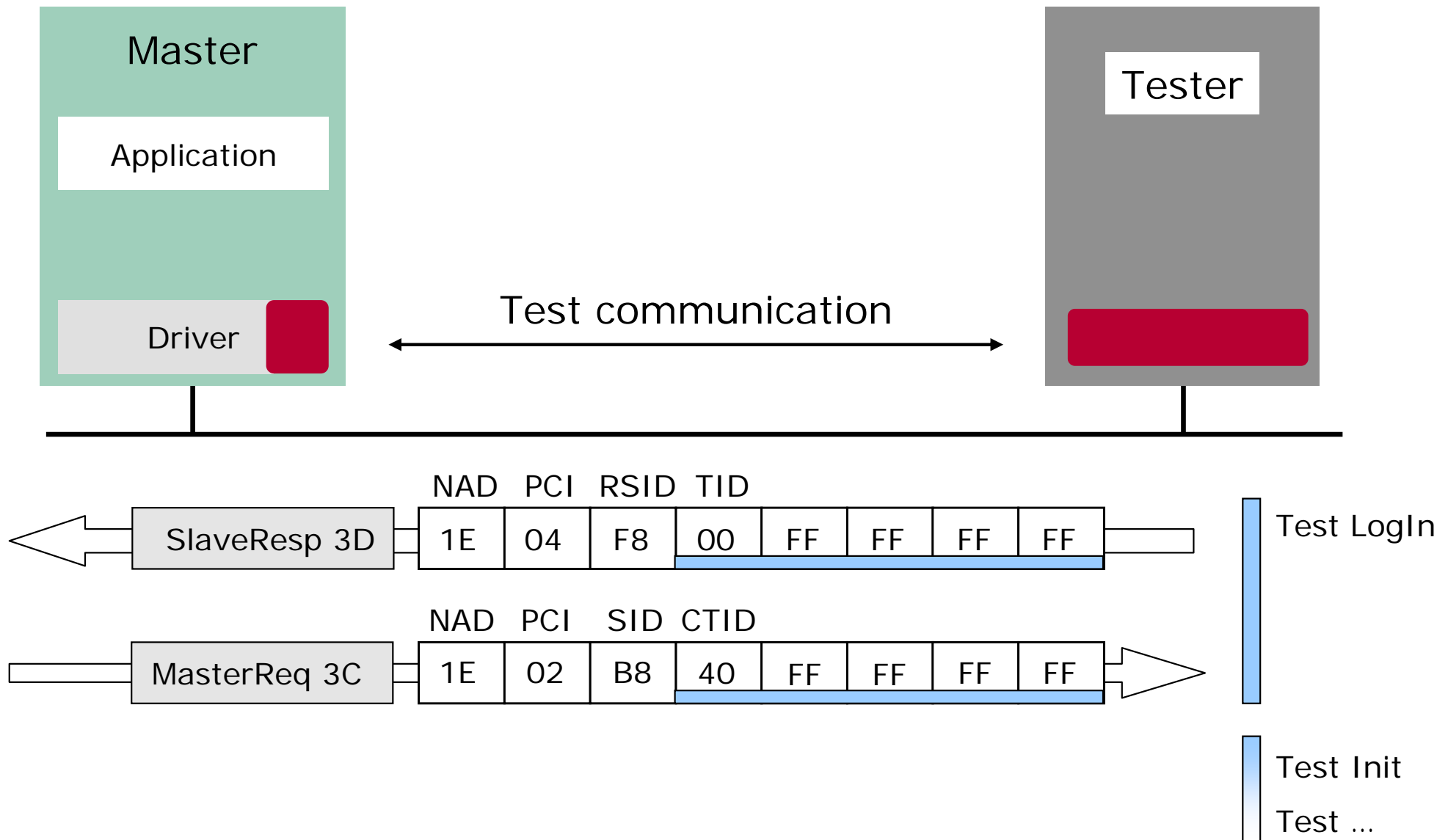
# Black Box LIN Master Test - Implementation

## Test Command Interface Summary

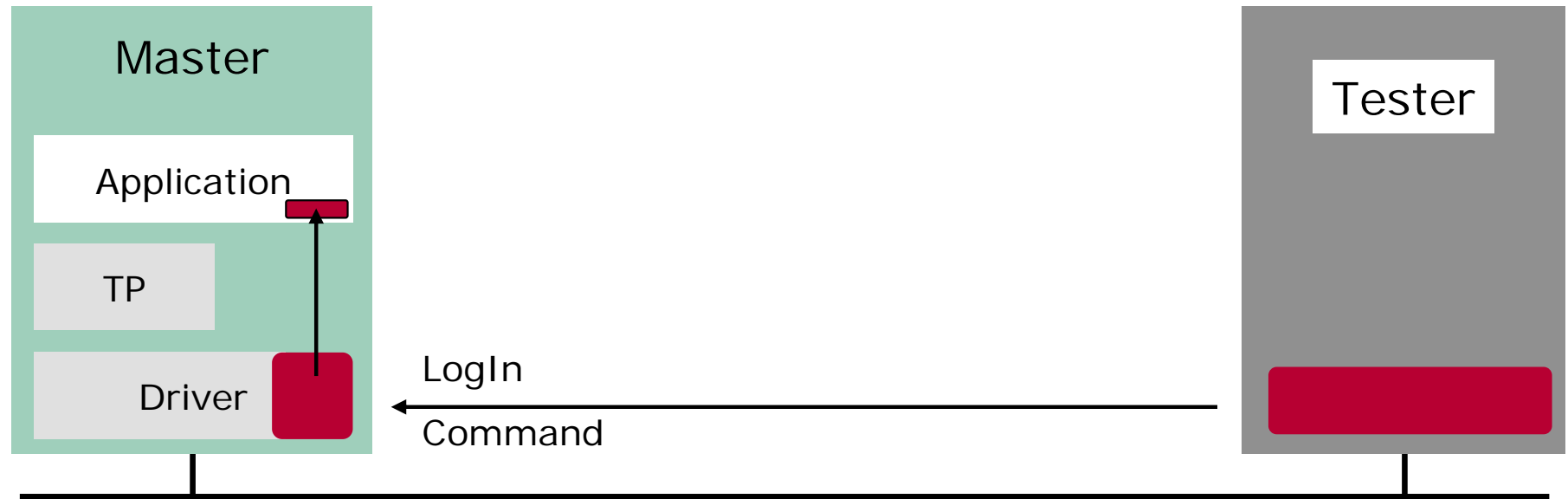
	Parameters	Test ID
Test LogIn	none	0x00
Test Presence / Idle	none	0x01
Test Setup / Trigger	$T_{\text{base}}$ Cycles, Reset/Init indication	0x02
Load Parameters	Offset, D1, D2, D3	0x03
Read Result	Offset, D1, D2, D3, D4	0x04
Set schedule table	Table index, Slot index, $T_{\text{base}}$ Cycles	0x05
Read Status Word	PID / none	0x06
Send Config Command	RCSID, RCNAD, RCPCI	0x07
Request Sleep Mode	Sleep cyc., Schedule Tab, Wakup Del.	0x08
Set signal	Signal handle, Signal value	0x09
Read signal	Signal handle	0x0A

# Black Box LIN Master Test - Implementation

## Example: LogIn



# Black Box LIN Master Test - Implementation

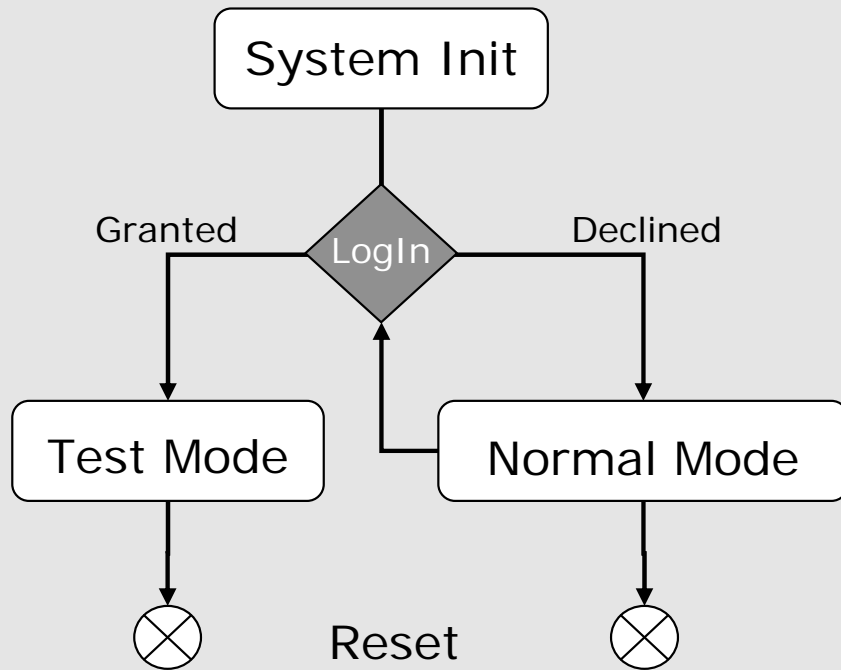


- ❑ Application must support test mode
- ❑ Test mode can only be entered
  - ❑ During time window after init – e.g. 10 sec
  - ❑ After security access
  - ❑ Depending on configuration (pre compile / runtime / HW)
  - ❑ ...

# Black Box LIN Master Test - Implementation

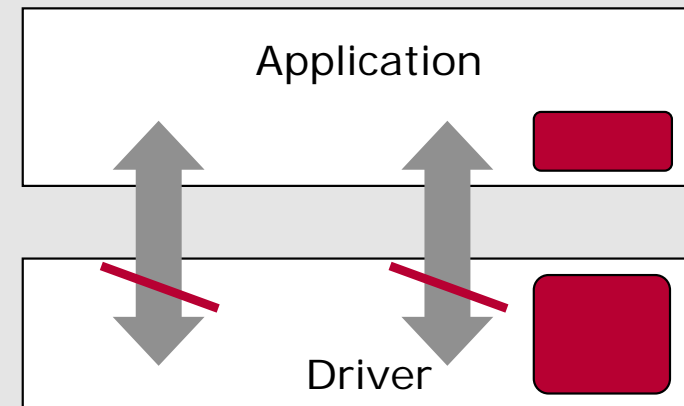
- ❑ Two possible strategies for interaction of application with LIN driver

- ❑ Isolated test and application mode
- ❑ No application task in test mode



- + No conflict between test and application
- CPU load different to normal application

- ❑ Interaction blocked when testing
- ❑ Application informed on test mode
- ❑ Normal application execution



- + Minimal changes to normal application
- Exception handling in application

- ❑ Required LIN driver extensions
  - ❑ State engine for test entry setup and performance
  - ❑ Test command interface
  - ❑ Application interface for test mode indication
  - ❑ References to LDF symbolic names e.g. schedules, signals, frames
  - ❑ Test schedule table with MasterReq and SlaveResp
  
- ➔ LIN Master driver extension with test server capabilities

Expected resource requirements:

20-30 % of LIN driver RAM and ROM

- + Single ECU stack possible for test and series production
- + Test code can be removed from application and driver
- + No conflict with existing Master functionality
- + Test and normal communication independent of each other
- + Extensible and open for standardization
  
- Application must support test mode
- Additional code and resource demands

- ❑ Prototype implementation for LIN 2.1 conformance test
- ❑ Get experience with customer projects
- ❑ Feedback from OEMs and suppliers

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)

**Authors:**

Andreas Pick, Sven Hesselmann

Vector Informatik GmbH

Ingersheimer Str. 24

70499 Stuttgart