

# LIN\_REPEATER Module Documentation

CSM GmbH, Filderstadt, Germany  
[www.csm.de/unicom/](http://www.csm.de/unicom/)

November 15, 2018

Date	Version	Name	Changes
2015-09-23	1.0	CSM/RN	first release
2017-11-10	1.1	CSM/RN	adjustable baud rate
2018-11-15	1.2	CSM/RN	typos corrected

All concepts and procedures introduced in this document are intellectual properties of CSM GmbH. Copying or usage by third parties without written permission of CSM GmbH is strictly prohibited. All trademarks mentioned in this document are properties of their respective owners. **This document is subject to changes without notice!**



CSM GmbH Computer-Systeme-Messtechnik  
Raiffeisenstrasse 36 70794 Filderstadt-Bonlanden  
Phone ++49 711 77964 0 Fax ++49 711 77964 40  
mailto:unicom@csm.de http://www.csm.de

Copyright © 2017 by CSM GmbH

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Overview</b>	<b>4</b>
<b>3</b>	<b>Loading and Configuration</b>	<b>5</b>
3.1	MODULE Command . . . . .	5
3.2	CONFIG_MODULE Command . . . . .	7
<b>4</b>	<b>LIN_REPEATER Module Commands</b>	<b>8</b>
4.1	LIN_REPEATER::READ_VERSION (2) . . . . .	8
4.2	LIN_REPEATER::CONFIG_RECEIVE_DATA (11) . . . . .	10
4.3	LIN_REPEATER::CONFIG_TRANSMIT_DATA (12) . . . . .	12
4.4	LIN_REPEATER::FETCH_DATA (13) . . . . .	14
<b>5</b>	<b>LIN_REPEATER Error Codes</b>	<b>16</b>
5.1	LIN_REPEATER::ErrorCodes . . . . .	16

# Chapter 1

## Introduction

*LIN\_REPEATER* is an *UCBASE* module that expands UCBASE by a LIN repeater functionality. If activated, UNICOM3 can receive a pre-defined *LIN* frame. If successful received, it responds with another pre-defined frame. That is done in the background without any further necessary commands. It can act as *LIN master* or *LIN Slave* and can handle standard and extended *LIN checksums* according to the LIN specification V2.1, 2006-11-24 with some restrictions.

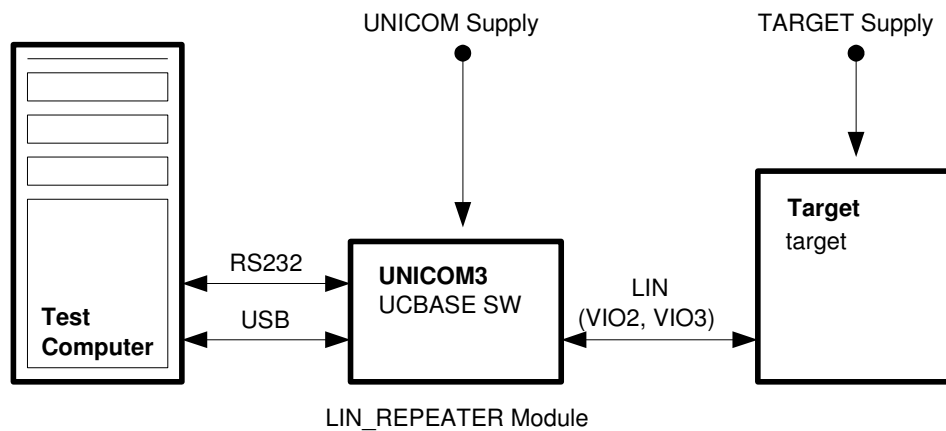
The module is designed to run together with other modules in parallel on the same UNICOM (Multi Module).

## Chapter 2

# Overview

To use UNICOM device with LIN\_REPEATER module, UNICOM must have at least hardware revision "C", and UCBASE software version V3.50 (!) or newer must be installed on UNICOM.

The figure below shows the components that are basically needed, and there connections.



Both *K-Line/LIN* lines of UNICOM (VIO2: DSUB62/14 and VIO3: DSUB62/15) can be used for LIN communication, V\_VIO: DSUB62/34 must be connected to +12V and GND: DSUB62/35) must be connected to GND.

The baudrate is fix, according to the value specified by the module command. No automatic baudrate recognition takes place in slave mode.

## Chapter 3

# Loading and Configuration

### 3.1 MODULE Command

This command downloads and runs the LIN\_REPEATER module on a specified slot.

#### Command, form 1 (unload module)

byte 0	byte 1	byte 2	byte 3
len	ecu	cmd	cks
3	0xC0	20,40..43	

#### Command, form 2 (load module)

byte 0	byte 1	byte 2	byte 3	...	byte N-2	byte N-1	byte N
len	ecu	cmd	mod 1	...	mod n	eos	cks
N=4+n	0xC0	20,40..43		...		0	

#### Command, form 3 (load module, with LIN baud rate)

byte 0	byte 1	byte 2	byte 3	...	byte N-6	byte N-5	
len	ecu	cmd	mod 1	...	mod n	eos	
N=8+n	0xC0	20,40..43		...		0	

byte N-4	...	byte N-1	byte N
baud rate			cks
MSB	...	LSB	

<b>len</b>	length of telegram
<b>ecu</b>	target address
<b>cmd</b>	command code
<b>mod</b>	filename of LIN_REPEATER module
<b>eos</b>	end-of-string (0)
<b>baud</b>	(optional) LIN baud rate (32 bit value), default: 19200
<b>cks</b>	checksum of telegram

### Response

byte 0	byte 1	byte 2	byte 3
len	ecu	status	cks
3	0xC0		

<b>len</b>	length of telegram
<b>ecu</b>	source address
<b>status</b>	result status
<b>cks</b>	checksum of telegram

### Remarks

- According to the used command code 40, 41, 42 or 43, the module is loaded into slot 0, 1, 2 or 3. With 20, module is loaded on all slots (monolytic).
- After successful execution of the MODULE command, the interface slot where the module has been loaded must be configured for *MODULE* interface using the *UCBASE::CONFIG\_UNICOM(1)* command.
- If slot 0 or 1 is used, LIN communication will be executed over VIO2, otherwise over VIO3.
- After download, the repeater functionality is stopped at first, and the module is pre-configured for acting as LIN Slave.
- The LIN\_REPEATER module can't be loaded more than once into different slots since it uses the internal asynchronous serial interface of UNICOM which only once exists.

## 3.2 CONFIG\_MODULE Command

By sending an UCBASE::CONFIG\_INTERFACE(4) command to a slot where the LIN\_REPEATER module is loaded and which is configured for MODULE interface, the CONFIG\_MODULE(4) command of the module is executed.

With this command, the LIN\_REPEATER module can be configured for acting as *LIN Slave* or, respectively, as *LIN Master*.

### Command

byte 0	byte 1	byte 2	byte 3	byte 4	byte 5
len	ecu	cmd	slot	mode	cks
N	0xC0	4	0..3	0,1	

<b>len</b>	length of telegram
<b>ecu</b>	target address
<b>cmd</b>	command code
<b>slot</b>	slot number where the LIN_REPEATER is resident
<b>mode</b>	0: master mode, else: slave mode
<b>cks</b>	checksum of telegram

### Response

byte 0	byte 1	byte 2	byte 3
len	ecu	status	cks
3	0xC0		

<b>len</b>	length of telegram
<b>ecu</b>	source address
<b>status</b>	result status
<b>cks</b>	checksum of telegram

### Remarks

- The default mode is *LIN Slave* after loading the module.



## Chapter 4

# LIN\_REPEATER Module Commands

### 4.1 LIN\_REPEATER::READ\_VERSION (2)

This command reports about the version information of LIN\_REPEATER.

#### Command

byte 0	byte 1	byte 2	byte 3
len	ecu	cmd	cks
3	xx	2	

**len** length of telegram  
**ecu** target address  
**cmd** command code  
**cks** checksum of telegram

#### Response

byte 0	byte 1	byte 2	byte 3	...	byte 18	byte 19
len	ecu	status	ver 1	...	ver 16	check
19	xx					

**len** length of telegram  
**ecu** source address  
**status** result status

<b>ver 1..16</b>	version string
<b>cks</b>	checksum of the response telegram

#### Remarks

- As version string LIN\_Reptr\_Vx.y should be reported.

## 4.2 LIN\_REPEATER::CONFIG\_RECEIVE\_DATA (11)

With this command, either the receive functionality can be stopped, or data to be received can be configured and the receive functionality can be started.

### Command form 1: stop

byte 0	byte 1	byte 2	byte 3
len	ecu	cmd	cks
3	xx	11	

### Command form 2: start

byte 0	byte 1	byte 2	byte 3	byte 4	
len	ecu	cmd	ckstype	ID	
N=5+n	xx	11	0, 1		

byte 5	...	byte N-1	byte N
data 1	...	data n	cks
	...		

<b>len</b>	length of telegram
<b>ecu</b>	target address
<b>cmd</b>	command code
<b>ckstype</b>	0: standard checksum, else: extended one
<b>ID</b>	LIN ID, without parity bits (0..63)
<b>data</b>	up to 8 frame data bytes to be received
<b>cks</b>	checksum of telegram

### Response

byte 0	byte 1	byte 2	byte 3
len	ecu	status	cks
3	xx		

<b>len</b>	length of telegram
<b>ecu</b>	source address
<b>status</b>	result status
<b>cks</b>	checksum of telegram

### Remarks

- After start of LIN\_REPEATER module the receiver is stopped.
- The receiving functionality is being enabled if at least one receive data byte per frame is defined with the command.
- A LIN frame is being received and stored if
  - the ID matches with the specified one
  - the checksum over the frame is correct
- A response frame is only sent by the LIN\_REPEATER module if data is matching with the specified one (content and number of bytes) and the transmitter is enabled by the CONFIG\_TRANSMIT\_DATA(12) command (ref. chapter 4.3 on page 12).
- However, a received frame can be fetched with the FETCH\_DATA(12) command even if transmitter is off or received data doesn't match (ref. chapter 4.4 on page 14).
- The complete repeater functionality only works if both receiver and transmitter are properly configured and started.

### 4.3 LIN\_REPEATER::CONFIG\_TRANSMIT\_DATA (12)

With this command, either the transmit functionality can be stopped, or data to be transmitted can be configured and the transmit functionality can be started.

#### Command form 1: stop

byte 0	byte 1	byte 2	byte 3
len	ecu	cmd	cks
3	xx	12	

#### Command form 2: start

byte 0	byte 1	byte 2	byte 3	byte 4	
len	ecu	cmd	ckstype	ID	
N=5+n	xx	12	0, 1		

byte 5	...	byte N-1	byte N
data 1	...	data n	cks
	...		

<b>len</b>	length of telegram
<b>ecu</b>	target address
<b>cmd</b>	command code
<b>ckstype</b>	0: standard checksum, else: extended one
<b>ID</b>	LIN ID, without parity bits (0..63)
<b>data</b>	up to 8 frame data bytes to be transmitted
<b>cks</b>	checksum of telegram

#### Response

byte 0	byte 1	byte 2	byte 3
len	ecu	status	cks
3	xx		

<b>len</b>	length of telegram
<b>ecu</b>	source address
<b>status</b>	result status
<b>cks</b>	checksum of telegram

### Remarks

- After start of LIN\_REPEATER module the transmitter is stopped.
- The transmit functionality is being enabled if at least one transmit data byte per frame is defined with the command.
- A response frame is only transmitted by the LIN\_REPEATER module if data is matching with the specified one (content and number of bytes) by the CONFIG\_RECEIVE\_DATA(11), ref. chapter 4.2 on page 10.
- The complete repeater functionality only works if both receiver and transmitter are properly configured and started.

## 4.4 LIN\_REPEATER::FETCH\_DATA (13)

This command reports the received frame data if any.

### Command

byte 0	byte 1	byte 2	byte 3
len	ecu	cmd	cks
3	xx	13	

**len** length of telegram  
**ecu** target address  
**cmd** command code  
**cks** checksum of telegram

### Response form 1, no (new) data received

byte 0	byte 1	byte 2	byte 3
len	ecu	status	cks
3	xx		

### Response form 2, data received

byte 0	byte 1	byte 2	byte 3	byte 4	...	byte N-1	byte N
len	ecu	status	ID	data 1	...	data n	cks
N=4+n	xx				...		

**len** length of telegram  
**ecu** source address  
**status** result status  
**ID** received ID without parity bits (0..63)  
**data** received frame data, up to 8  
**cks** checksum of telegram

### Remarks

- Receiver must be started before LIN frames can be received using the CONFIG\_RECEIVE\_DATA(11) command (ref. chapter 4.2 on page 10).
- Every received frame can be fetched only once.

- If a new frame is received while another one is already stored, the previous one will be lost without error.



## Chapter 5

# LIN\_REPEATER Error Codes

### 5.1 LIN\_REPEATER::ErrorCodes

The following table describes possible error codes reported by the *status* parameter of the response telegrams, and their meanings.

Error	Code	Description
NOT_CONFIGURED_ERROR	0x90	service is currently not available
RESOURCE_ERROR	0x92	resource conflict: another module which is already loaded uses the same hardware
NO_ERROR	0xA0	No error occurred
PARAMETER_ERROR	0xB0	Wrong parameter in command telegram
ECU_RECEIVE_ERROR	0xC4	Error while receiving
UNKNOWN_COMMAND_ERROR	0xFF	command telegram with unknown command code received

Refer `ucbase.pdf` for more possible error codes.