

LIN_GW Module Documentation

CSM GmbH, Filderstadt, Germany
www.csm.de/unicom/

June 26, 2024

Date	Version	Name	Changes
2014-08-12	1.00	CSM/RN	first release
2015-02-06	1.03	CSM/RN	UDS over LIN
2015-09-11	1.05	CSM/TO	code download, LIN broadcast
2016-03-02	1.06	CSM/RN	MULTI_MODULE
2016-10-14	1.09	CSM/RN	configurable baud rate and receive frame delay
2016-11-18	1.10	CSM/RN	typos corrected
2019-04-25	1.11	CSM/RN	more typos corrected
2020-07-01	1.12	CSM/TO	corrected broadcast NAD description
2020-07-16	1.30	CSM/RN	VIO mapping for Rev.D changed
2020-07-31	1.40	CSM/TO	new option byte for broadcast behaviour
2021-12-10	1.60	CSM/RN	2nd uds gateway form
2024-06-25	1.70	CSM/TO	Download WMBA command

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 © 2016 by CSM GmbH

Contents

1	Introduction	3
2	Overview	4
3	Loading and Configuration	5
3.1	MODULE Command	5
3.2	CONFIG_INTERFACE Command	7
4	FASTFLASH	8
5	LIN_GW Commands	9
5.1	Generic Commands	9
5.1.1	LIN_GW::READ_VERSION (2)	9
5.2	Raw LIN Related Commands	11
5.2.1	LIN_GW::CONFIG_MODULE (1)	11
5.2.2	LIN_GW::RECEIVE_WAKEUP (94)	13
5.2.3	LIN_GW::SEND_WAKEUP (95)	14
5.2.4	LIN_GW::SEND_FRAME (96)	15
5.2.5	LIN_GW::RECEIVE_FRAME (97)	16
5.3	UDS over LIN Related Commands	17
5.3.1	LIN_GW::UDS_CONFIGURE (98)	17
5.3.2	LIN_GW::UDS_GATEWAY (99,100)	19
5.3.3	LIN_GW::DOWNLOAD_TOOLBOX (21)	20
5.3.4	LIN_GW::DOWNLOAD_CODE (14)	21
5.3.5	LIN_GW::DOWNLOAD_WMBA (22)	23
5.4	LIN_GW::ErrorCodes	24

Chapter 1

Introduction

LIN_GW is a module for extending the *UCBASE* software running on *UNICOM3*. It implements LIN protocol, both *LIN Master* and *LIN Slave* mode. Based on LIN protocol, UDS messages can be sent and received. Further, a mechanism to download RAM code and program flash data with UDS over LIN is realized.

Chapter 2

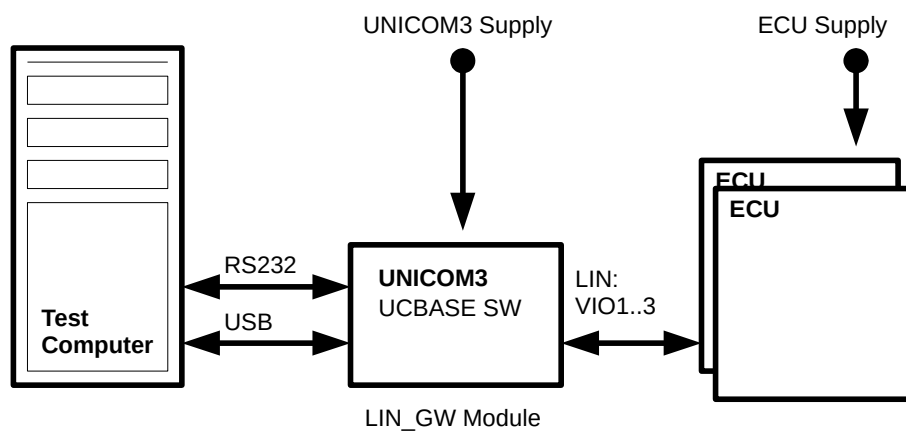
Overview

To use UNICOM3 device with LIN_GW module, UCBASE software version 3.08 or newer must be installed on UNICOM3.

The figure below shows the components of the system.

The LIN bus is realized on UNICOM's side by using VIO2 or VIO3 (Rev.C) resp. VIO1, VIO2 and VIO3 (Rev.D)

Up to 2 (3) ECUs can be connected to one UNICOM at once.



Chapter 3

Loading and Configuration

3.1 MODULE Command

This command downloads and runs the LIN_GW module.

Command, form 1 (unload module)

byte 0	byte 1	byte 2	byte 3
len	ecu	cmd	cks
3	0xC0	20,40,41,42,43	

Command, form 2 (load module, default baudrate)

byte 0	byte 1	byte 2	byte 3	...	byte N-2	byte N-1	byte N
len	ecu	cmd	mod 1	...	mod m	EOS	cks
N=m+4	0xC0	20,40,41,42,43				0	

Command, form 3 (load module, with baudrate)

byte 0	byte 1	byte 2	byte 3	...	byte N-2	byte N-5	
len	ecu	cmd	mod 1	...	mod m	EOS	
N=m+8	0xC0	20,40,41,42,43				0	

	byte N-4	...	byte N-1	byte N
	Baudrate			cks
	MSB	...	LSB	

len length of telegram
ecu target address
cmd command code

mod filename of module (here: lin_gw.mod)
EOS end-of-string (0)
Baudrate (optional) LIN baudrate in baud. Default: 19200
cks checksum of telegram

Response

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

len length of telegram
ecu source address
status result status
cks checksum of telegram

Remarks

- After loading the module with command 20, at least one interface slot must be configured for *MODULE* interface using the *CONFIG_UNICOM(1)* command of UCBASE software. After that, the commands that LIN_GW implements can be used.
- If the module has been loaded with one of the *MULTI_MODULE* commands (40..43), exactly the slot must be configured for *MODULE* interface where the module is loaded to. Please refer *ucbase.pdf* for more information about *MULTI_MODULE*.
- The signal line which is used for communication with target device can be selected by the ecu number (slot) of the command telegram:
 ecu = 1000xxxx (e.g. 0x80): VIO1 is selected (Rev.C: send as master only)
 ecu = 1001xxxx (e.g. 0x90): VIO2 is selected
 ecu = 1010xxxx (e.g. 0xA0): VIO3 is selected
 ecu = 1011xxxx (e.g. 0xB0): VIO3 is selected (Rev.C), VIO1 is selected (Rev.D)
- After loading the module, it acts as *LIN Master* per default. To enable the *LIN Slave* mode use the *CONFIG_MODULE(1)* command (ref. chapter 5.2.1 on page 11).
- To let the VIO lines work, the V_VIO pin must be connected to +12V, and one of the GND pins must be connected to the corresponding GND of the +12V supply.
- The used LIN baud rate is fixed to 19200 baud.

3.2 CONFIG_INTERFACE Command

This command is not implemented by the LIN_GW module. For configuration refer to CONFIG_MODULE(1) command (chapter 5.2.1 on page 11).

Chapter 4

FASTFLASH

No FASTFLASH is implemented by the LIN_GW module.

Chapter 5

LIN_GW Commands

5.1 Generic Commands

5.1.1 LIN_GW::READ_VERSION (2)

This command reports about the module version information.

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	cks
19	xx					

len length of telegram
ecu source address
status result status
ver 1..16 version string
cks checksum of telegram

Remarks

- As version string `LIN_GW``_____``V``x``.``yy` should be reported.

5.2 Raw LIN Related Commands

5.2.1 LIN_GW::CONFIG_MODULE (1)

This command switches between *LIN Master* mode (default) and *LIN Slave* mode.

Command

byte 0	byte 1	byte 2	byte 3	byte 4
len	ecu	cmd	mode	
4	xx	1	0,1	

len	length of telegram
ecu	target address
cmd	command code
mode	0: master mode 1: slave mode
cks	checksum of telegram

Response

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

len	length of telegram
ecu	source address
status	result status
cks	checksum of telegram

Remarks

- In *LIN Master Mode*, UNICOM always sends a LIN header by itself. If SEND_FRAME(96) command (chapter 5.2.4 on page 15) is being executed, it sends the frame data, followed by the checksum thereafter. If RECEIVE_FRAME(97) command (chapter 5.2.5 on page 16) is being executed, it waits for receiving the frame data and the checksum thereafter.
- In *LIN Slave Mode*, UNICOM waits for receiving a LIN header with the specified ID. If SEND_FRAME(96) command is being executed, it sends the frame data, followed by the checksum thereafter. If RECEIVE_FRAME(97) command is being executed, it waits for receiving the frame data and the checksum thereafter.
- In *LIN Slave Mode*, UNICOM only waits for receiving a LIN header while executing one of these commands. Receiving in background is not (yet)

implemented.

- In LIN Slave Mode, UNICOM doesn't support (yet) automatic baud rate recognicing.
- The mode is switched for all signal lines at once independently of the selected slot.

5.2.2 LIN_GW::RECEIVE_WAKEUP (94)

This command waits for receiving a *Wakeup Pulse* from target device (minimum 250 us).

Command

byte 0	byte 1	byte 2	byte 3
len	ecu	cmd	cks
3	xx	94	

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

Response

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

len length of telegram
ecu source address
status result status
cks checksum of telegram

Remarks

- Since VIO1 is only be able to send, not to receive, the command doesn't work with slot = 0.

5.2.3 LIN_GW::SEND_WAKEUP (95)

This command sends a *Wakeup Pulse* to target device. It is realized by sending a *break* with a length of 13 bits.

Command

byte 0	byte 1	byte 2	byte 3
len	ecu	cmd	cks
3	xx	95	

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

Response

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

len length of telegram
ecu source address
status result status
cks checksum of telegram

5.2.4 LIN_GW::SEND_FRAME (96)

This command sends a LIN frame. Depending on the current mode (LIN Master or LIN Slave) UNICOM acts accordingly.

Command

byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	...	byte N-1	byte N
len	ecu	cmd	ckstype	ID	data 1	...	data n	cks
N=5+n	xx	96	0,1			...		

len	length of telegram
ecu	target address
cmd	command code
ckstype	0: classic checksum 1: diagnostic checksum
ID	Frame ID, with or without parity bits
data	Frame data, max. 8 bytes
cks	checksum of telegram

Response

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

len	length of telegram
ecu	source address
status	result status
cks	checksum of telegram

Remarks

- Since VIO1 is only able to send, not to receive, this command can only be executed in LIN Master Mode when slot is 0.
- Refer CONFIG_MODULE(1) (chapter 5.2.1 on page 11) for information about difference between the modes.

5.2.5 LIN_GW::RECEIVE_FRAME (97)

This command receives a LIN frame. Depending on the current mode (LIN Master or LIN Slave) UNICOM acts accordingly.

Command

byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6
len	ecu	cmd	ckstype	ID	length	cks
6	xx	97	0,1		0..8	

len	length of telegram
ecu	target address
cmd	command code
ckstype	0: classic checksum 1: diagnostic checksum
ID	Frame ID, with or without parity bits
cks	checksum of telegram

Response

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

len	length of telegram
ecu	source address
status	result status
data	received data bytes (max. 8)
cks	checksum of telegram

Remarks

- Since VIO1 is only able to send, not to receive, this command can't be executed when slot is 0.
- Refer CONFIG_MODULE(1) (chapter 5.2.1 on page 11) for information about difference between the modes.

5.3 UDS over LIN Related Commands

5.3.1 LIN_GW::UDS_CONFIGURE (98)

With this command, the *Send ID*, *Receive ID* and *Net Address (NAD)* can be configured.

Command

byte 0	byte 1	byte 2	byte 3	byte 4	
len	ecu	cmd	Send ID	Rec ID	
6/7	xx	98			

	byte 5	byte 6	byte 7	byte 6/7/8
	NAD	RFD	Opt7F	cks

len	length of telegram
ecu	target address
cmd	command code
Send ID	LIN ID for sending UDS messages (default: 0x3C)
Rec ID	LIN Receive ID of received UDS messages (default: 0x3D)
NAD	Net Address (default: 0x05)
RFD	(optional) Receive Frame Delay in ms. Default: 0
Opt7F	(optional) Behaviour when receiving broadcast (0x7F) message. Default: 1. See Remarks
cks	checksum of telegram

Response

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

len	length of telegram
ecu	source address
status	result status
cks	checksum of telegram

Remarks

- This command defines the communication parameters for UDS over LIN. It only must be executed if the default settings doesn't match.

- The Receive Frame Delay delays fetching receive frames in master mode when trying to receive an UDS telegram.
- The *Opt7F* byte defines the behaviour when a message with broadcast NAD (0x7F) is received. The following options are available:

Option	Description
0	only UDS data is returned
1	NAD and length are returned for short answers (default)
2	not used at the moment
3	NAD and length are returned for all answers

5.3.2 LIN_GW::UDS_GATEWAY (99,100)

With this command, single UDS commands can be sent over LIN, and receive the response.

Command, form 1

byte 0	byte 1	byte 2	byte 3	byte 4	...	byte N-1	byte N
len	ecu	cmd	SID	data 1	...	data n	cks
N=4+n	xx	99			...		

Command, form 1

byte 0	byte 1	byte 2	byte 3	
len	ecu	cmd	to	
N=4+n	xx	100		

byte 4	byte 5	...	byte N-1	byte N
SID	data 1	...	data n	cks
		...		

len	length of telegram
ecu	target address
cmd	command code
to	timeout time in units to 100 msec. If 0, no UDS response telegram is being expected from target device.
SID	code of UDS command to be executed
data	additional data if any
cks	checksum of telegram

Response

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

len	length of telegram
ecu	source address
status	result status
stat	response code
data	additional data if any
cks	checksum of telegram

5.3.3 LIN_GW::DOWNLOAD_TOOLBOX (21)

This command downloads executable code into RAM of target device using UDS over LIN.

Command

byte 0	byte 1	byte 2	byte 3	...	byte N-2	byte N-1	byte N
len	ecu	cmd	tbx 1	...	tbx n	EOS	cks
N=4+n	xx	21		...		0	

len	length of telegram
ecu	target address
cmd	command code
tbx	name of toolbox file on UNICOM's storage medium (SRECORD format)
EOS	End-Of-String (0)
cks	checksum of telegram

Response

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

len	length of telegram
ecu	source address
status	result status
cks	checksum of telegram

Remarks

- The used protocol strongly depends on the target device. Here, the LDM L1,L2,L3 devices are supported.
- Target must be prepared by sending some more UDS commands e.g. "Enter Programming Session" using the UDS_GATEWAY command (refer chapter 5.3.2 on page 19), before this command can be used.

5.3.4 LIN_GW::DOWNLOAD_CODE (14)

This command downloads code for programming it into the flash memory of target device. A toolbox that is able to program the data must be running on target.

Command, Form 1

byte 0	byte 1	byte 2	byte 3	...	byte N-6	byte N-5	
len	ecu	cmd	code 1	...	code n	EOS	
N=8+n	xx	14		...		0	

	byte N-4	byte N-3	byte N-2	byte N-1	byte N
	startaddr				cks
	MSB			LSB	

Command, Form 2

byte 0	byte 1	byte 2	byte 3	...	byte N-14	byte N-13	
len	ecu	cmd	code 1	...	code n	EOS	
N=16+n	xx	14		...		0	

	byte N-12	byte N-11	byte N-10	byte N-9	byte N-8	byte N-7	byte N-6	byte N-5	
	startaddr				endaddr				
	MSB			LSB	MSB			LSB	

	byte N-4	byte N-3	byte N-2	byte N-1	byte N
	offset				cks
	MSB			LSB	

len	length of telegram
ecu	target address
cmd	command code
code	name of file on UNICOM's storage medium that contains the code to be downloaded (binary format)
EOS	End-Of-String (0)
startaddr	destination address of code
endaddr	end address of code
offset	data offset in file
cks	checksum of telegram

Response

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

len	length of telegram
ecu	source address
status	result status
cks	checksum of telegram

Remarks

- In command form 1, the entire file is downloaded to the device, starting from the *startaddr*.
- The used protocol strongly depends on the target device. Here, the LDM L1,L2,L3 devices are supported.
- A toolbox must be running on target device that can receive and program the code data, using the DOWNLOAD_TOOLBOX command (ref. chapter 5.3.3 on page 20)

5.3.5 LIN_GW::DOWNLOAD_WMBA (22)

This command downloads a binary file to an address range in memory. This command uses the UDS service WriteMemoryByAddress (0x3D) for downloading.

Command

byte 0	byte 1	byte 2	byte 3	...	byte N-7	byte N-6	
len	ecu	cmd	file 1	...	file n	EOS	
N=9+n	xx	22		...		0	

	byte N-5	byte N-4	byte N-3	byte N-2	byte N-1	byte N
	startaddr				portion	cks
	MSB			LSB		

len	length of telegram
ecu	target address
cmd	command code
file	name of binary file on UNICOM's storage medium
EOS	End-Of-String (0)
addr	start address in memory
portion	the portion size for each WriteMemoryByAddress service command
cks	checksum of telegram

Response

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

len	length of telegram
ecu	source address
status	result status
cks	checksum of telegram

Remarks

- The size of the binary file has to be aligned with the selected portion size.

5.4 LIN_GW::ErrorCodes

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

Error	Code	Description
NO_ERROR	0xA0	No error occurred
NOT_CONFIGURED_ERROR	0x90	Requested service not available
PARAMETER_ERROR	0xB0	Wrong parameter in command telegram
LENGTH_ERROR	0xB3	Wrong command telegram length
ECU_RECEIVE_ERROR	0xC4	unexpected data byte received from target device
ECU_TIMEOUT_ERROR	0xC5	No response from target device within timeout time
NO_VIO_ERROR	0xE0	No VIO supply voltage connected or signal line is clamped to GND
NO_WAKEUP_ERROR	0xE1	No valid wakeup pulse received (e.g. too short)
UDS_FORMAT_ERROR	0xE8	Wrong telegram format while receiving UDS message
UDS_BUFFER_OVERFLOW_ERROR	0xE9	Response message too large (internal error)
UDS_SEQUENCE_ERROR	0xEA	Wrong sequence counter received
UDS_NEGATIVE_RESPONSE_ERROR	0xEB	Target sent a negative response
UDS_WRONG_RESPONSE_ERROR	0xEC	Target sent a response that was not expected at this point
UNKNOWN_COMMAND_ERROR	0xFF	Command code not supported by the module

Furthermore, the error codes described in `ucbase.pdf` can occur.