

SEND_KLINE Module Documentation

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

June 10, 2024

Date	Version	Name	Changes
2015-07-27	1.0	CSM/RN	first release
2016-02-24	1.1	CSM/RN	selectable channel
2017-06-21	1.2	CSM/TO	KWP2000 GW, Download
2017-10-27	1.3	CSM/RN	clear fifo command
2018-06-13	1.4	CSM/TO	Wakeup gateway command
2019-05-07	1.5	CSM/RN	InterByte Delay
2019-07-19	1.6	CSM/RN	even parity
2020-07-03	1.8	CSM/RN	VIO1 with Rev.D devices
2020-16-03	1.9	CSM/RN	VIO1 on slot 3
2024-06-10	2.3	CSM/RN	Logging

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 © 2024 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	SEND_KLINE Commands	8
4.1	SEND_KLINE::READ_VERSION (2)	8
4.2	SEND_KLINE::CLEAR_FIFO (85)	10
4.3	SEND_KLINE::KLINE_GATEWAY (86)	11
4.4	SEND_KLINE::KLINE_WAKEUP_GATEWAY (87)	13
4.5	SEND_KLINE::KWP2000_GATEWAY (99)	15
4.6	SEND_KLINE::KWP2000_DOWNLOAD (22)	17
4.7	SEND_KLINE::LOG (200)	19
4.8	SEND_KLINE::ErrorCodes	21

Chapter 1

Introduction

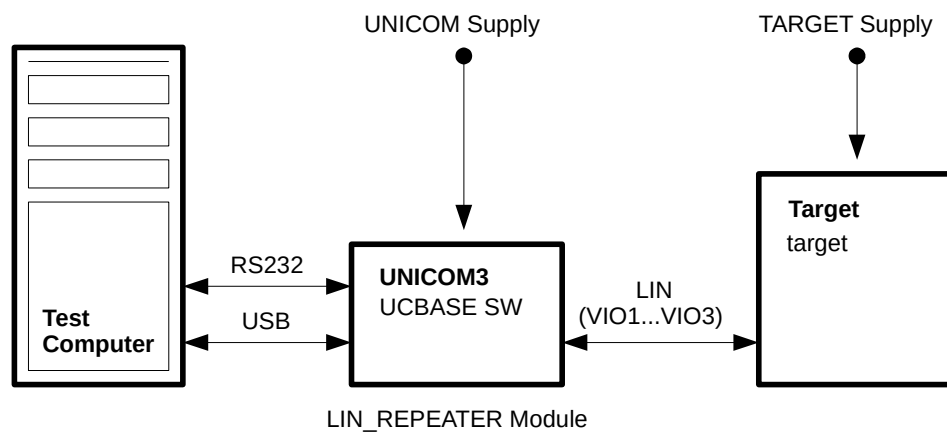
SEND_KLINE is a module for extending the *UCBASE* software running on *UNI-COM3*. It realizes sending and receiving data over the K-Line interface of *UNI-COM3*. It provides a gateway function to use the KWP2000 protocol over K-Line.

Chapter 2

Overview

To use UNICOM3 device with SEND_KLINE module, UCBASE software version 3.21 or newer must be installed on UNICOM3.

The figure below shows the components of the system.



UNICOM uses VIO2 (Pin 14 of DSUB62 connector) or VIO3 (Pin 15 of DSUB62 connector) for K-Line communication.

In order to supply the internal K-Line driver of UNICOM, an additional power supply of 12V must be connected between V_VIO (pin 34) and GND (pin 35).

Chapter 3

Loading and Configuration

3.1 MODULE Command

This command downloads and runs the SEND_KLINE 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)

byte 0	byte 1	byte 2	byte 3	...	byte N-4	byte N-3	
len	ecu	cmd	mod 1	...	mod n	EOS mod	
N=6+n	0xC0	20,40,41,42,43				0	

byte N-2	byte N-1	byte N
baud/10		cks
MSB	LSB	

Command, form 3 (load module, incl. ibd)

byte 0	byte 1	byte 2	byte 3	...	byte N-7	byte N-6	
len	ecu	cmd	mod 1	...	mod n	EOS mod	
N=9+n	0xC0	20,40,41,42,43				0	

byte N-5	byte N-4	byte N-3	byte N-2	byte N-1	byte N
baud/10		ibd		parity	cks
MSB	LSB	MSB	LSB		

len length of telegram

ecu	target address
cmd	command code
mod	filename of module (here: <code>send_kline.mod</code>)
EOS mod	end-of-string of module filename (0)
baud/10	(optional) 1/10 of Baud Rate of K-Line communication (default: 9600)
ibd	(optional) InterByte Delay in us (default: 0)
parity	(optional) 0: no parity (8N1), 1: even parity (8E1), default: no parity
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 successful execution of the **MODULE** command, at least one of the interface slots must be configured for *MODULE Interface* (15) using the *CONFIG_UNICOM* command.
- If **MULTI_MODULE(4x)** command is used, exactly that slot must be set to **MODULE** where the module is loaded to.

3.2 CONFIG_INTERFACE Command

This command is not implemented by the *SEND_KLINE* module.

Chapter 4

SEND_KLINE Commands

4.1 SEND_KLINE::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 `SEND_KLINE_Vx.yy` should be reported.

4.2 SEND_KLINE::CLEAR_FIFO (85)

This command clears the internal receive and send FIFO of K-Line interface.

Command

byte 0	byte 1	byte 2	byte 3
len	ecu	cmd	cks
3	xx	85	

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

4.3 SEND_KLINE::KLINE_GATEWAY (86)

With this command, data bytes can be sent/received over the K-Line interface of UNICOM.

Command

byte 0	byte 1	byte 2	byte 3	byte 4	
len	ecu	cmd	rec_cnt	timeout	
N=5+n	xx	86			

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

len	length of telegram
ecu	target address
cmd	command code
rec_cnt	number of requested response bytes
timeout	receive timeout in units to 10 ms
data	data bytes to be sent
cks	checksum of telegram

Response

byte 0	byte 1	byte 2	byte 3	byte 4	
len	ecu	status	rec_status	dummy	
N=5+n	xx			...	

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

len	length of telegram
ecu	source address
status	result status
rec_status	status of receive process
dummy	not used here, will always report 0
data	received data bytes
cks	checksum of telegram

Remarks

- If module received the command over slot 0 and UNICOM is Rev.D, VIO1 is used for communication. If UNICOM is not Rev.D, VIO2 is used in this case.
- If module received the command over slot 1, VIO2 is used for communication.
- If module received the command over slot 2, VIO3 is used for communication.
- If module received the command over slot 3 and UNICOM is Rev.D, VIO1 is used for communication. If UNICOM is not Rev.D, VIO3 is used in this case.
- If *rec_cnt* is 0, no data is expected to receive.
- If no send data is specified, no data will be sent but data can be received.
- As consequence, if *rec_cnt* is 0 and no send data is specified, the command will do exactly nothing.

4.4 SEND_KLINE::KLINE_WAKEUP_GATEWAY (87)

This command has the same functionality as command SEND_KLINE::KLINE_GATEWAY (87). However, with this command a configurable wakeup pulse can be send prior to the data bytes.

Command

byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	
len	ecu	cmd	idle_t	low_t	high_t	
N=8+n	xx	87	0 ... 255	1 ... 255	1 ... 255	

byte 6	byte 7	byte 8	...	byte N-1	byte N
rec_cnt	timeout	data 1	...	data n	cks
			...		

len	length of telegram
ecu	target address
cmd	command code
idle_t	idle time in [ms]
low_t	low time in [ms]
high_t	high time in [ms]
rec_cnt	number of requested response bytes
timeout	receive timeout in units to 10 ms
data	data bytes to be sent
cks	checksum of telegram

Response

byte 0	byte 1	byte 2	byte 3	byte 4	
len	ecu	status	rec_status	dummy	
N=5+n	xx			...	

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

len	length of telegram
ecu	source address
status	result status
rec_status	status of receive process
dummy	not used here, will always report 0
data	received data bytes

cks checksum of telegram

Remarks

- With *idle time*, a wait time before the actual wakeup pulse can be specified
- *low time* specifies the length of the "low" phase of the wakeup pulse
- *high time* specifies the length of the "high" phase of the wakeup pulse
- If module received the command over slot 0 and UNICOM is Rev.D, VIO1 is used for communication. If UNICOM is not Rev.D, VIO2 is used in this case.
- If module received the command over slot 1, VIO2 is used for communication.
- If module received the command over slot 2, VIO3 is used for communication.
- If module received the command over slot 3 and UNICOM is Rev.D, VIO1 is used for communication. If UNICOM is not Rev.D, VIO3 is used in this case.
- If *rec_cnt* is 0, no data is expected to receive.
- If no send data is specified, no data will be sent but data can be received.
- As consequence, if *rec_cnt* is 0 and no send data is specified, the command will do exactly nothing.

4.5 SEND_KLINE::KWP2000_GATEWAY (99)

With this command, KWP2000 messages can be sent and received over the K-Line interface of UNICOM.

Command

byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	
len	ecu	cmd	opt	timeout	send_id	
N=7+n	xx	99				

	byte 6	byte 7	...	byte N-1	byte N
	rec_id	data 1	...	data n	cks
			...		

len	length of telegram
ecu	target address
cmd	command code
opt	option byte, see remarks
timeout	receive timeout in units of 20 ms
send_id	KWP2000 send ID
rec_id	KWP2000 receive ID
data	data bytes to be sent
cks	checksum of telegram

Response

byte 0	byte 1	byte 2	byte 3	byte 4	
len	ecu	status	rec_status	dummy	
N=5+n	xx			...	

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

len	length of telegram
ecu	source address
status	result status
rec_status	status of receive process
dummy	not used here, will always report 0
data	received data bytes
cks	checksum of telegram

Remarks

- If module received the command over slot 0 and UNICOM is Rev.D, VIO1 is used for communication. If UNICOM is not Rev.D, VIO2 is used in this case.
- If module received the command over slot 1, VIO2 is used for communication.
- If module received the command over slot 2, VIO3 is used for communication.
- If module received the command over slot 3 and UNICOM is Rev.D, VIO1 is used for communication. If UNICOM is not Rev.D, VIO3 is used in this case.
- The following options are available through the *opt* byte:
 - Bit 0: if this bit is set, a 50ms Wake-Up pulse is send prior to the KWP2000 message
 - Bit 1: if this bit is set, only sending is performed, but no receiving
- When data is send, the receive FIFO is cleared prior to sending, so that the answer to the sent message will be received properly.
- If no send data is specified, no data will be sent but data can be received.
- As consequence, if the *opt* byte is set to 0x02 (no reception) and no send data is specified, the command will do exactly nothing.
- The header of the KWP2000 message is generated automatically, the message looks like this: *0x80 / 0x80length SendID RecID [length] data1 dataN cks*
- If the amount of data is smaller then 127, the length information is stored in the lower 7 bits of the first byte (0x80 | length). In this case, the optional length byte is not included in the message. If 128 or more bytes are to be sent, the length is stored in the additional length byte.
- When receiving, the *SEND_KLINE* module automatically determines which of the two formats described above the answer is in.

4.6 SEND_KLINE::KWP2000_DOWNLOAD (22)

This command can be used to download data, utilizing the KWP2000 protocol over K-Line. It handles all the involved KWP2000 commands, *request download* (0x34), *transfer data* (0x36) and *transfer exit* (0x37).

Command

byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	
len	ecu	cmd	wait	send_id	rec_id	
N=12+n	xx	22				

byte 6	byte 7	byte 8	byte 9	
start				dummy
MSB		LSB	0x00	

byte 10	...	byte N-2	byte N-1	byte N
file 1	...	file n	EOS	cks
	...		0	

len	length of telegram
ecu	target address
cmd	command code
wait	wait time in ms, see remarks
send_id	KWP2000 send ID
rec_id	KWP2000 receive ID
start	the start address
dummy	0x00, filler byte
file	the name of the data file
EOS	End-of-string for the file name
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

- If module received the command over slot 0 and UNICOM is Rev.D, VIO1 is used for communication. If UNICOM is not Rev.D, VIO2 is used in this case.
- If module received the command over slot 1, VIO2 is used for communication.
- If module received the command over slot 2, VIO3 is used for communication.
- If module received the command over slot 3 and UNICOM is Rev.D, VIO1 is used for communication. If UNICOM is not Rev.D, VIO3 is used in this case.
- The *wait* parameter is used as wait time in between *transfer data (0x36)* commands.
- The *request download (0x34)* command is build as follows: *0x34 start/3 0x00 size/2 0x00*.
- The size is automatically determined from the given file
- The portion size used for the *transfer data (0x36)* commands is taken from the ECUs answer to the *request download (0x34)* command.
- The *transfer data (0x36)* commands looks as follows: *0x36 data1 data2 dataN*
- command formats other then the ones described above are currently not supported

4.7 SEND_KLINE::LOG (200)

With this command, logging of all sent and received data bytes into a file inside of UNICOM can be enabled and disabled.

Command form 1: enable

byte 0	byte 1	byte 2	byte 3	...	byte N-2	byte N-1	byte N
len	ecu	cmd	file	...	file	eos	cks
N	xx	200		...		0	

Command form 2: disable

byte 0	byte 1	byte 2	byte 3
len	ecu	cmd	cks
3	xx	200	

len	length of telegram
ecu	target address
cmd	command code
file	name of file where data should be logged
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

- Log data is being store in ASCII format into the file and looks like this:

```
*** send kline V2.30 log file ***
Send: 01 02 03 04 05 06 07 08
Recv: 08 07 06 05 04 03 02 01
*** eof ***
```

- The resulting file can be uploaded by using the file commands (ref. `ucbase.pdf`).
- Before uploading, logging must be stopped by using the second form of command (`disable`).

4.8 SEND_KLINE::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	function/command not available
LENGTH_ERROR	0xB3	Wrong command telegram length
WRONG_ECHO_ERROR	0xC8	Wrong echo message received while downloading the toolbox
DOWNLOAD_ERROR	0xE1	KWP2000 protocol error during file download
UNKNOWN_COMMAND_ERROR	0xFF	Command code not supported by the module

More error codes can be found in `ucbase.pdf`