K-Line

ASC Logging Format

Specification

Version 1.1 of 2015-09-21

Status	Completed
Publisher	Vector Informatik GmbH © 2016 All rights reserved. Any distribution or copying is subject to prior written approval by Vector. Note: Hardcopy documents are not subject to change management.

Document Management

Revision list

Version	Date	Editor	Section	Changes, comments
1.0	2011-02-02	Trs	All	Creation for CANoe/CANalyzer 7.5 SP2
1.1	2014-12-09	Mw	3.4	Correction: length in case of hexadecimal mode and Port syntax
1.1.1	2016-12-01	Mom	all	CI and Layout

Contents

1 Disclaimer

2 Overview

This document specifies the format of events in the CANoe/CANalyzer ASC logging used for K-Line byte events.

3 K-Line communication observer

3.1 Header

Events can either be recorded in hexadecimal ("base hex") or decimal mode ("base dec"). Timestamps are written absolute ("timestamps absolute") or relative to the preceding event ("timestamps relative").

3.2 Information as comment

In addition to the bytes sent and received on K-Line, CANoe/CANalyzer also displays the "messages" assembled from these bytes. These messages are documented in the log file only for the convenience of the reader, i.e. CANoe/CANalyzer does NOT process this information in any way when the log file is replayed. Example (lines have been wrapped):

```
date Wed Feb 2 03:15:29 pm 2011
base hex  timestamps absolute
no internal events logged
// version 7.5.0
6.695000e   COM2 Tx        5      1   97
8.901000e   COM2 Rx    10400      1   55
8.907000e   COM2 Rx    10400      1   6B
8.913000e   COM2 Rx    10400      1   8F
8.939000e   COM2 Tx    10400      1   70
8.970000e   COM2 Rx    10400      1   68
9.026000e   COM2 Tx    10400      6   82 61 11 10 86 8A
// K-Line:    9.026000   COM2 Tx    10400 11 61     6   82 61 11 10 86 8A
9.069000    COM2 Rx    10400      6   82 11 61 50 86 CA
// K-Line:    9.069000   COM2 Rx    10400 61 11     6   82 11 61 50 86 CA
12.257000   COM2 Tx    10400      6   82 61 11 1A 9B A9
// K-Line:  12.257000    COM2 Tx    10400 11 61     6   82 61 11 1A 9B A9
12.315000   COM2 Rx    10400      7   83 11 61 7F 1A 78 06
// K-Line:  12.315000    COM2 Rx    10400 61 11     7   83 11 61 7F 1A 78 06
12.365000   COM2 Rx    10400      6   83 11 61 7F 1A 78
12.366000   COM2 Rx    10400      1   06
// K-Line:  12.366000    COM2 Rx    10400 61 11     7   83 11 61 7F 1A 78 06
12.404000   COM2 Rx    10400     14   B0 11 61 5A 9B 37 4C 36 39 32 30 38 37 30
12.419000   COM2 Rx    10400     14   41 20 20 31 36 31 30 03 00 1B D1 00 00 00
12.433000   COM2 Rx    10400     14   00 00 00 4B 4F 4D 42 49 49 4E 53 54 52 55
12.448000   COM2 Rx    10400     10   4D 45 4E 54 20 52 42 34 20 D5
// K-Line:  12.448000    COM2 Rx    10400 61 11    52   B0 11 61 5A 9B 37 4C 36 39 32 30 38
37 30 41 20 20 31 36 31 30 03 00 1B D1 00 00 00 00 00 00 4B 4F 4D 42 49 49 4E 53 54 52 55 4D
45 4E 54 20 52 42 34 20 D5
```

3.3 Events

The section lists all K-Line observer events in CANoe/CANalyzer ASCII logging. See section 3.4 for an explanation of the symbols.

3.3.1 Byte event

The bytes received or sent on the serial port are listed in this format:

Format	`<time> <port> <direction> <baudrate> <length> <data>`
Example	`12.315000     COM2 Rx     10400      7    83 11 61 7F 1A 78 06`

3.3.2 Message event

Every K-Line message comment is prefixed with basic information:

Format	`"// K-Line: " <time> <port> <direction> <baudrate> <source> <destination>` `<length> <data>`
Example	`// K-Line: 12.315000 COM2 Rx     10400 61 11    7    83 11 61 7F 1A 78 06`

3.4 Symbols

Explanation of symbols used in descriptions above.

Symbol	Width in chars (hex)	Width in chars (dec)	Meaning	Range	Example (hex)
`<time>`	>=1	>=1	Time of reception of the bytes, i.e. the *end* of a byte sequence	-	12.315000
`<port>`	3-9	3-9	COM*n* for serial port *n*, or K-Line*n* for LINcab *n*	1 < *n* < 100	"COM2" "K-Line1"
`<direction>`	2	2	Were the bytes sent or received?	"Tx", "Rx"	"Tx"
`<baudrate>`	1-6	1-6	Rate the data was transferred on	>= 5	10400
`<source>`	>=1	>=1	Source address	Numeric or symbolic	"Any_ecu_example"
`<destination>`	>=1	>=1	Destination address	Numeric or symbolic	"<tester>"
`<length>`	-	1-4	Number of bytes transferred	1...1000	7
`<data>`	>=2	>=3	Bytes transported within this CAN message	…	83 11