Specification

LIN ASC Logging Format

Author:	Peter Decker		
Version:	1.0.18 dated 2015-06-23		
Status:	released (in progress / completed / verified / released)		
Customer:			
Location:			
Printing date:	23.06.15	Number of pages:	18

Document Management

Release and Release History			
Name	**Role**	**Version**	**Date**
Rogers, Gavin (Gcr)	PLD	1.0.8	2006-10-11
Shapiro, Arkady (Sha) / Document Owner	PLD	1.0.9	2009-02-19
Shapiro, Arkady (Sha) / Document Owner	PLD	1.0.10	2009-04-17
Tinnirello, Gianfranco (Gia)	DEV	1.0.11	2009-05-19
Shapiro, Arkady (Sha) / Document Owner	PLD	1.0.12	2009-08-27
Shapiro, Arkady (Sha) / Document Owner	PLD	1.0.14	2010-02-17
Shapiro, Arkady (Sha) / Document Owner	PLD	1.0.17	2010-10-27
Ronald van Splunter (Vrd) / Document Owner	PLD	1.0.18	2015-06-23

Revision list				
Version	**Date**	**Editor**	**Section**	**Changes, comments**
1.0.0	2005-03-11	Dr	All	Created
1.0.1	2005-03-11	Gcr	All	Reviewed and released.
1.0.2	2005-08-25	Dr	3.1	Added Event Triggered Frame Info
1.0.3	2006-03-02	Dr	3.1	Added Statistic Events and LIN Message enhancements
1.0.4	2006-11-17	Sc	3.1	Removed Event Triggeres Frame Info for CANoe 6.0
1.0.5	2006-11-22	Sha	All	Reviewed and updated
1.0.6	2006-11-24	Sha	3.1	Format has been modified for CANoe/CANalyzer 6.0 SP2
1.0.7	2006-12-13	Gcr	All	Reviewed and released
1.0.8	2006-10-11	Sha	3.1	Format has been modified for CANoe/CANalyzer 7.0 (EOF timestamp)
1.0.9	2009-02-19	Sha	3.1	Format has been extended for CANoe/CANalyzer 7.1 SP3 (Response Baudrate)
1.0.10	2009-04-17	Sha	3.1	Description of StateReason files in ReceiveError has been updated
1.0.11	2009-05-18	Gia	1	Added Disclaimer
1.0.12	2009-08-27	Sha	3.1; 3.2	Format has been modified for CANoe/CANalyzer 7.2
1.0.13	2010-02-10	Sc	3.1; 3.1.1	New data field CSM for LIN Message, LIN Transmission Error, LIN Receive Error and LIN Checksum Error
1.0.14	2010-02-17	Sha	3.1; 3.1.1	Cosmetic correction to changes done in 1.0.13
1.0.15	2010-08-25	Sc	3.1.2; 3.1.3; 3.2	Added Unexpected Wakeup and Short Or Slow Response
1.0.16	2010-08-27	Sc	3.1.2; 3.2	Added Disturbance Event
1.0.17	2010-10-27	Sha	3.1.2; 3.1.3; 3.	Cosmetic correction to changes done in 1.0.16
1.0.18	2015-06-23	Vrd	3.2	Changed channel range

Table of contents

1 Disclaimer

Severability clause – Restrictions for the usage of Vector logging data formats outside of Vector products

The format specification / access functions for the Vector BLF and ASC logging data formats are made available under the restrictions and conditions cited hereafter.

Please note that Vector Informatik neither gives any guarantee nor assumes any liability beyond compulsory legal regulations for the BLF or ASC logging format respectively as well as for the access functions to the single objects.

Vector Informatik disclaims all liability for errors which might be contained in the access functions or the format specification itself.

Vector Informatik does neither provide support for the integration into your software nor for problems occurring inside your software on the customer side.

Beyond that Vector Informatik reserves the right to change the BLF or ASC data format espectively anytime without prior notification. Therefore, the compatibility of the format is not ensured.

2 Overview

The document specifies the format of LIN events in the CANoe/CANalyzer ASC logging.

3 Format

3.1 LIN Events

The section lists all LIN events in CANoe/CANalyzer ASC logging. See section 3.2 for an explanation of the symbols.

Note, that all optional fields are enclosed by parentheses and have a different background color. The conditions under which the common optional fields can occur are described in a separate section below.

LIN Message	
LIN frame received or transmitted on a LIN channel.	
Format up to v6.0	<Time> <Channel> <ID> <Dir> <DLC> <D0>...<D7> (slave = <slave id>, state = <state>) checksum = <checksum> header time = <header time>, full time = <full time>, start of frame = <start of frame>, sync break = <sync break time> us (<number sync break bits> bits), sync delimiter = <sync delimiter time> us (<sync delimiter bits> bits)
Format from v6.1	<Time> <Channel> <ID> <Dir> <DLC> <D0>...<D7> (slave = <slave id>, state = <state>) checksum = <checksum> header time = <header time>, full time = <full time> SOF = <start of frame> BR = <baudrate> break = <SyncBreak> <SyncDel> (subId = <NAD> <MessageId> <SupplierId>) EOH = <end of header> EOB = <T0> … <T7> sim = <simulated>
Format from v7.0	<Time> <Channel> <ID> <Dir> <DLC> <D0>...<D7> (slave = <slave id>, state = <state>) checksum = <checksum> header time = <header time>, full time = <full time> SOF = <start of frame> BR = <baudrate> break = <SyncBreak> <SyncDel> (subId = <NAD> <MessageId> <SupplierId>) EOH = <end of header> EOB = <T0> … <T7> sim = <simulated> EOF = <end of frame>
Format from	<Time> <Channel> <ID> <Dir> <DLC> <D0>...<D7> (slave = <slave id>, state = <state>) checksum = <checksum> header time = <header time>, full time = <full time> SOF = <start of frame> BR = <baudrate> break = <SyncBreak> <SyncDel> (subId = <NAD> <MessageId> <SupplierId>) EOH = <end of

v7.1 SP3	header> EOB = <T0> … <T7> sim = <simulated> EOF = <end of frame> RBR = <response baudrate>
Format from v7.2	<Time> <Channel> <ID> <Dir> <DLC> <D0>...<D7> (slave = <slave id>, state = <state>) checksum = <checksum> header time = <header time>, full time = <full time> SOF = <start of frame> BR = <baudrate> break = <SyncBreak> <SyncDel> (subId = <NAD> <MessageId> <SupplierId>) EOH = <end of header> EOB = <T0> … <T7> sim = <simulated> EOF = <end of frame> RBR = <response baudrate> HBR = <header baudrate> HSO = <stop bit offset in header> RSO = <stop bit offset in response>
Format from v7.2 SP3	<Time> <Channel> <ID> <Dir> <DLC> <D0>...<D7> (slave = <slave id>, state = <state>) checksum = <checksum> header time = <header time>, full time = <full time> SOF = <start of frame> BR = <baudrate> break = <SyncBreak> <SyncDel> (subId = <NAD> <MessageId> <SupplierId>) EOH = <end of header> EOB = <T0> … <T7> sim = <simulated> EOF = <end of frame> RBR = <response baudrate> HBR = <header baudrate> HSO = <stop bit offset in header> RSO = <stop bit offset in response> CSM = <checksum model>
Example	0.073973 Li 2d Tx 8 00 f0 f0 ff ff ff ff ff checksum = 70 header time = 40, full time = 130 SOF = 0.067195 BR = 19230 break = 937125 114062 EOH = 0.069266 EOB = 0.069789 0.070312 0.070835 0.071358 0.071881 0.072404 0.072927 0.073450 sim = 1 EOF = 0.073973 RBR = 19231 HBR = 19230.769231 HSO = 26000 RSO = 26000 CSM = enhanced

3.1.1 LIN Error Events

LIN Transmisson Error	
A transmission error event occurs when no Slave responds to a frame header from the Master (simulated or real).	
Format up to v6.0	<Time> <Channel> <ID> TransmErr (slave = <slave id>, state = <state>) header time = <header time> full time = <full time>
Format from v6.1	<Time> <Channel> <ID> TransmErr (slave = <slave id>, state = <state>) header time = <header time>, full time = <full time> SOF = <start of frame> BR = <baudrate> break = <SyncBreak> <SyncDel> (subId = <NAD> <MessageId> <SupplierId>) EOH = <end of header>
Format from v7.2	<Time> <Channel> <ID> TransmErr (slave = <slave id>, state = <state>) header time = <header time>, full time = <full time> SOF = <start of frame> BR = <baudrate> break = <SyncBreak> <SyncDel> (subId = <NAD> <MessageId> <SupplierId>) EOH = <end of header> > HBR = <header baudrate> HSO = <stop bit offset in header>
Format from v7.2 SP2	<Time> <Channel> <ID> TransmErr (slave = <slave id>, state = <state>) header time = <header time>, full time = <full time> SOF = <start of frame> BR = <baudrate> break = <SyncBreak> <SyncDel> (subId = <NAD> <MessageId> <SupplierId>) EOH = <end of header> > HBR = <header baudrate> HSO = <stop bit offset in header> CSM = <checksum model>
Example	0.424674 Li 33 TransmErr header time = 40, full time = 166 SOF = 0.416054 BR = 19230 break = 937187 113250 EOH = 0.418122 HBR = 19230.769231 HSO = 26000 CSM = enhanced

LIN Receive Error
A receive error event may have a wide variety of causes. An external Master can cause a receive error event: • by transmitting sync break that is too short, • by not returning the correct value `0x55` in the sync field, • by assigning an incorrect parity to the frame identifier. A receive error event can also be caused: • by a Slave sending an illegal character during a Bus Idle phase (e.g. because it did not fin-

	ish transmission quickly enough and the checksum byte of the response was sent during the Bus Idle phase), • by a faulty (dominant) stop bit (i.e. framing error), • if the LIN hardware receives a character that is different from the character sent during transmission, • if the LIN hardware only receives part of a frame, at the start of a measurement (in a correctly functioning system).
Format up to v6.0	<Time> <Channel> (<ID> <DLC>) RcvError: <description> (char = <offending byte>) (slave = <slave id>, state = <state>)
Format from v6.1	<Time> <Channel> (<ID> <DLC>) RcvError: <description> (char = <offending byte>) (slave = <slave id>, state = <state>) StateReason = <StateReason> ShortError = <IsShortError> DlcTimeout = <IsDLCTimeout> HasDatabytes = <HasDatabytes> (<D0>...<D7>) SOF = <start of frame> BR = <baudrate> break = <SyncBreak> <SyncDel> (subId = <NAD> <MessageId> <SupplierId>) (EOH = <end of header>) (EOB = <T0> … <T7>)
Format from v7.1 SP3	<Time> <Channel> (<ID> <DLC>) RcvError: <description> (char = <offending byte>) (slave = <slave id>, state = <state>) StateReason = <StateReason> ShortError = <IsShortError> DlcTimeout = <IsDLCTimeout> HasDatabytes = <HasDatabytes> (<D0>...<D7>) SOF = <start of frame> BR = <baudrate> break = <SyncBreak> <SyncDel> (subId = <NAD> <MessageId> <SupplierId>) (EOH = <end of header>) (EOB = <T0> … <T7>) (RBR = <response baudrate>)
Format from v7.2	<Time> <Channel> (<ID> <DLC>) RcvError: <description> (char = <offending byte>) (slave = <slave id>, state = <state>) StateReason = <StateReason> ShortError = <IsShortError> DlcTimeout = <IsDLCTimeout> HasDatabytes = <HasDatabytes> (<D0>...<D7>) SOF = <start of frame> BR = <baudrate> break = <SyncBreak> <SyncDel> (subId = <NAD> <MessageId> <SupplierId>) (EOH = <end of header>) (EOB = <T0> … <T7>) RBR = <response baudrate> HBR = <header baudrate> HSO = <stop bit offset in header> RSO = <stop bit offset in response>
Format from v7.2 SP3	<Time> <Channel> (<ID> <DLC>) RcvError: <description> (char = <offending byte>) (slave = <slave id>, state = <state>) StateReason = <StateReason> ShortError = <IsShortError> DlcTimeout = <IsDLCTimeout> HasDatabytes = <HasDatabytes> (<D0>...<D7>) SOF = <start of frame> BR = <baudrate> break = <SyncBreak> <SyncDel> (subId = <NAD> <MessageId> <SupplierId>) (EOH = <end of header>) (EOB = <T0> … <T7>) RBR = <response baudrate> HBR = <header baudrate> HSO = <stop bit offset in header> RSO = <stop bit offset in response> CSM = <checksum model>
Example	0.554673 Li 33 8 RcvError: timeout while waiting for checksum field StateReason = 0c ShortError = 0 DlcTimeout = 0 HasDatabytes = 1 05 00 00 00 00 ff ff ff SOF = 0.546052 BR = 19230 break = 937187 112437 EOH = 0.548121 EOB = 0.548644 0.549167 0.549690 0.550213 0.550736 0.551259 0.551782 0.552305 RBR = 19231 RSO = 26000 HBR = 19230.769231 HSO = 26000 CSM = enhanced
Notes	• Optional <ID> and <DLC> are available when at least one of the following conditions is fulfilled: o The error is a result of applying LIN Stress Feature Set, i.e. CANoe knows how a correct frame looks like o FrameId has been correctly received § <DLC> is known from the database or the frame has already been correctly received during the current measurement § <DLC> is -1 otherwise • Optional <offending byte> is available if the reason of the error is a byte value which violates the protocol • Optional data bytes <D0>..<D7> are available if <HasDatabytes> flag is set and if the error is not in a header part of the frame • Optional data byte timestamps <T0>..<T7> are available if data bytes

	<D0>..<D7> are available • Optional response baud rate value <response baudrate) is available if data byte timestamps <T0>..<T7> are available (this value is optional up to v7.1 only)

LIN Sync Error	
This error event can only occur when an external Master is used. Synchronization errors occur if the LIN hardware cannot synchronize with an external Master. This might happen if the baud rate actually used by the Master deviates by more than 15 % from the baud rate specified by the LIN hardware. In this case the baud rate value should be modified. This error event may also occur if the Master transmits an invalid or corrupted Sync field. The synchronization error messages displays 4 time intervals (in microseconds) detected between the falling signal edges of the Sync field. The expected time interval between consecutive falling signal edges is 2 bit times. After the first failure interval has been seen the rest of array elements are initialized to 0.	
Format up to v6.0	<Time> <Channel> SyncError <TimeInterval0>..<TimeInterval3>
Format from v6.1	<Time> <Channel> SyncError <TimeInterval0>..<TimeInterval3> SOF = <start of frame> BR = <baudrate> break = <SyncBreak> <SyncDel>
Example	2.022336 L2 SyncError 208 0 0 0 SOF = 2.021077 BR = 19230 break = 937125 113312

LIN Checksum Error	
A checksum error event occurs when a Slave sends an incorrect checksum value for a frame response that is otherwise correct.	
Format up to v6.0	<Time> <Channel> <ID> CSErr <Dir> <DLC> <D0>...<D7> (slave = <slave id>, state = <state>) checksum = <checksum> header time = <header time>, full time = <full time>
Format from v6.1	<Time> <Channel> <ID> CSErr <Dir> <DLC> <D0>...<D7> (slave = <slave id>, state = <state>) checksum = <checksum> header time = <header time>, full time = <full time> SOF = <start of frame> BR = <baudrate> break = <SyncBreak> <SyncDel> (subId = <NAD> <MessageId> <SupplierId>) EOH = <end of header> EOB = <T0> … <T7> sim = <simulated>
Format from v7.0	<Time> <Channel> <ID> CSErr <Dir> <DLC> <D0>...<D7> (slave = <slave id>, state = <state>) checksum = <checksum> header time = <header time>, full time = <full time> SOF = <start of frame> BR = <baudrate> break = <SyncBreak> <SyncDel> (subId = <NAD> <MessageId> <SupplierId>) EOH = <end of header> EOB = <T0> … <T7> sim = <simulated> EOF = <end of frame>
Format from v7.1 SP3	<Time> <Channel> <ID> CSErr <Dir> <DLC> <D0>...<D7> (slave = <slave id>, state = <state>) checksum = <checksum> header time = <header time>, full time = <full time> SOF = <start of frame> BR = <baudrate> break = <SyncBreak> <SyncDel> (subId = <NAD> <MessageId> <SupplierId>) EOH = <end of header> EOB = <T0> … <T7> sim = <simulated> EOF = <end of frame> RBR = <response baudrate>
Format from v7.2	<Time> <Channel> <ID> CSErr <Dir> <DLC> <D0>...<D7> (slave = <slave id>, state = <state>) checksum = <checksum> header time = <header time>, full time = <full time> SOF = <start of frame> BR = <baudrate> break = <SyncBreak> <SyncDel> (subId = <NAD> <MessageId> <SupplierId>) EOH = <end of header> EOB = <T0> … <T7> sim = <simulated> EOF = <end of frame> RBR = <response baudrate> HBR = <header baudrate> HSO = <stop bit offset in header> RSO = <stop bit offset in response>

Format from v7.2 SP3	<Time> <Channel> <ID> CSErr <Dir> <DLC> <D0>...<D7> (slave = <slave id>, state = <state>) checksum = <checksum> header time = <header time>, full time = <full time> SOF = <start of frame> BR = <baudrate> break = <SyncBreak> <SyncDel> (subId = <NAD> <MessageId> <SupplierId>) EOH = <end of header> EOB = <T0> … <T7> sim = <simulated> EOF = <end of frame> RBR = <response baudrate> HBR = <header baudrate> HSO = <stop bit offset in header> RSO = <stop bit offset in response> CSM = <unknown|classic|enhanced|error>
Example	0.462829 Li 33 CSErr Tx 8 05 00 00 00 00 ff ff ff checksum = 86 header time = 40, full time = 130 SOF = 0.456050 BR = 19230 break = 937187 114875 EOH = 0.458122 EOB = 0.458645 0.459168 0.459691 0.460214 0.460737 0.461260 0.461783 0.462306 sim = 1 EOF = 0.462829 RBR = 19231 HBR = 19230.769231 HSO = 26000 RSO = 26000 CSM = <checksum model>

LIN Spike Event	
A spike event occurs when a short (normally less than 1 bit time) dominant signal has been detected on a LIN channel.	
Format up to v6.0	<Time> <Channel> Spike <Dir> <SpikeLength> microseconds
Format from v6.1	<Time> <Channel> Spike <Dir> <SpikeLength> microseconds SOF = <start of frame> BR = <baudrate>
Example	5.990958 L2 Spike Rx 56 microseconds SOF = 5.990902 BR = 9615

LIN Dominant Signal	
A dominant signal event occurs when a LIN channel remains in the dominant state for a time, which is longer than a valid wakeup frame and not a valid sync break.	
Format up to v6.0	<Time> <Channel> Dominant signal <DomSigType> <DomSigLength> microseconds
Format from v6.1	<Time> <Channel> Dominant signal <DomSigState> <DomSigLength> microseconds SOF = <start of frame> BR = <baudrate>
Example	8.976802 Li Dominant signal detected 5003 microseconds SOF = 8.971798 BR = 9615 8.977000 Li Dominant signal finished 5201 microseconds SOF = 8.971798 BR = 9615

3.1.2 LIN Info Events

LIN Baudrate	
This info event is only displayed when an external Master is configured. This info event is send by the LIN hardware at the start of a measurement and whenever the baud rate changes by more than 0.5 % during a measurement. If this info event is displayed, then the LIN hardware is synchronized with the baud rate of the external Master.	
Format	<Time> <Channel> Baudrate <baudrate>
Example	0.0188 Li Baudrate 9615

LIN DLC Info	
This info event is only displayed when an external Master is configured and the LIN hardware successfully detected the DLC of an unknown frame. This DLC value is set as the expected DLC value for this frame. An error is displayed if the same frame is received with a different DLC. If the master mode of LIN hardware is activated and a frame's DLC is not specified, then the DLC is determined using the frame's identifier.	
Format	<Time> <Channel> <ID> DlcInfo <DLC >
Example	12.6375 Li 20 DlcInfo 4

LIN Checksum Info	
This info event is displayed when the LIN hardware successfully detected the checksum model of an unknown frame. This checksum model is set as the expected checksum model for this frame. An error is displayed if the same frame is received with a different checksum model.	
Format	<Time> <Channel> <ID> CSInfo <checksum model info>
Example	0.0201 Li 22 CSInfo Using classic checksum

LIN Scheduler Mode Change	
This info event is only displayed when a Master is simulated and a frame header of a new schedule table is transmitted for the first time. This info event may appear on starting a measurement.	
Format	<Time> <Channel> SchedModChng prior scheduler mode = <schedule table index>, next scheduler mode = <schedule table index>
Example	0.1000 Li SchedModChng prior scheduler mode = 2, next scheduler mode = 0

LIN Slave Timeout	
The enhanced LIN hardware capabilities can be used to define a timeout for a FSM state to a pre-defined error state on exceeding this timeout and a Slave timeout message is displayed.	
Format	<Time> <Channel> SlaveTimeout slave-id = <slave id>, current state = < state>, following state = <state>
Example	1.0012 Li SlaveTimeout slave-id = 0, current state = 0, following state = 1

LIN Event Triggered Frame Info	
This info event is displayed when an event triggered frame is received or transmitted on a LIN channel. IMPORTANT: This event is generated up to CANoe/CANalyzer 5.2 only.	
Format	<Time> <Channel> <ID> EvTrigFrmInfo <ETFName> <description>
Example	1.909165 Li 3a EvTrigFrmInfo ETF_MotorStates No response

LIN Statistic Info	
This info event transports bus statistics. (Bus load [in range 0..1], obsolete value, obsolete value, transmitted frames, received frames, transmission errors) IMPORTANT: This event is generated up to CANoe/CANalyzer 5.2 only.	
Format	<Time> <Channel> Statistic <ChannelNum> <BusLoad> <Bursts total> <Bursts overrun> <Frames sent> <Frames received> <Frames unanswered>
Example	1.999580 Li Statistic 1 0.903601 0 0 0 73 0

LIN Short or slow response	
This event occurs if a set of receive errors could be a valid header followed by a short or slow response. IMPORTANT: This event is generated from CANoe/CANalyzer 7.5 only	
Format	<Time> <Channel> <ID> <DLC> ShortOrSlowResponse: NumRespBytes = <NumberOfResponseBytes> <D0>…<D8> SlowResponse = <IsSlowResponse> InterruptedByBreak = <ResponseWasInterruptedByBreak> SOF = <start of frame> BR = <baudrate> break = <SyncBreak> <SyncDel> (subId = <NAD> <MessageId> <SupplierId>) EOH = <end of header> EOB = <T0> … <T8> HBR = <header baudrate> HSO = <stop bit offset in header> CSM = <checksum model>
Example	1.298765 Li 1 8 ShortOrSlowResponse: NumRespBytes = 9 11 12 13 14 15 16 17 18 99 SlowResponse = 1 InterruptedByBreak = 0 SOF = 1.279516 BR = 19230 break = 937250 102625 EOH = 1.281570 EOB = 1.283679 1.285759 1.287839 1.289927 1.292007 1.294087 1.296167 1.298244 HBR = 19230.769231 HSO = 26000 CSM = unknown

LIN Disturbance event	
This event occurs if CANoe/CANalyzer explicitly caused to disturb one bit or a sequence of bits. IMPORTANT: This event is generated from CANoe/CANalyzer 7.5 only	
Format	<Time> <Channel> DisturbanceEvent Type = <DisturbanceType> ByteIndex = <ByteIndex> BitIndex = <BitIndex> BitOffset = <BitOffsetInSixteenthBits> Length = <DisturbanceLengthInSixteenthBits> Header = <IDorFF> Disturbing header = <IDorFF>
Example	1.323661 Li DisturbanceEvent Type = dominant ByteIndex = 1 BitIndex = 6 BitOffset = 0 Length = 16 Header = 2D Disturbing header = FF

3.1.3 LIN Sleep/Wakeup Events

LIN Sleep Mode
A Sleep mode event occurs: • At the start of a measurement in order to report the initial state of the LIN hardware, • Every time the mode (Wakeup or Sleep) of LIN hardware changes.

Format	<Time> <Channel> SleepModeEvent <simulated> <description text>
Example	0.7772 Li SleepModeEvent 0 entering sleep mode due to sleep mode frame

LIN Wakeup Frame	
A Wakeup-Frame event is displayed when a wakeup request has been detected on a LIN channel.	
Format up to v6.0	<Time> <Channel> WakeupFrame <Dir> <WakeupByte>
Format from v6.1	<Time> <Channel> WakeupFrame <Dir> <WakeupByte> SOF = <start of frame> BR = <baudrate> LengthCode = <WakeupLengthInfo>
Example	2.318672 Li WakeupFrame Tx 00 SOF = 2.317671 BR = 19230 LengthCode = 0
Notes	For LIN 2.0 the <WakeupByte> is always 0

LIN Unexpected wakeup	
This event occurs if an unexpected byte received in bus idle phase of wake mode could be a wakeup frame. IMPORTANT: This event is generated from CANoe/CANalyzer 7.5 only	
Format for LIN 2.x	<Time> <Channel> Unexpected wakeup: approx. <Width> us SOF = <start of frame> BR = <baudrate>
Format for LIN 1.x	<Time> <Channel> Unexpected wakeup: Signal = <WakeupByte> SOF = <start of frame> BR = <baudrate>
Example	0.892363 Li Unexpected wakeup: approx. 260 us SOF = 0.891843 BR = 19230

3.1.4 Common optional fields

Below is a list of optional fields which can occur in the events above.

- "slave = <slave id>, state = <state>" appears when LIN FSM (Finite State Machine) is simulated via CAPL functions
- "subId = <NAD> <MessageId> <SupplierId>" appears for dynamic frames (LIN 2.0) only

3.2 Symbols

Symbol	Width in chars (hex)	Width in chars (dec)	Meaning	Range	Example	Special
<baudrate>		5	Event's baudrate [in bits/sec]		19230	
<Channel>	5	5	The number of the LIN channel	I, 2..255, *	Li L2 L*	- "Li" describes the channel 1 - '*' describes the channel wildcard
<checksum>	2	3	Checksum byte value	0..255	70	
<checksum model>	7	7	Expected checksum model for checksum value	unknown, classic, enhanced, error	enhanced	
<checksum model info>		23	CS Info event only: Textual description of checksum model. Possible values: • "Using enhanced checksum" • "Using classic checksum"		Using enhanced checksum	
<Dir>		4	direction of transmission	Rx, Tx	Rx	
<DLC >		2	data length code	-1..8	5	
<NumberOfResponseBytes>		1	Number of response bytes; can include the checksum.	1..9	9	
<D0>...<D7>	2 (per value)	3 (per value)	Data byte values	0..255	03	Number of values corresponds to DLC
<D0>…<D8>	2 (per value)	3 (per value)	Response byte values; can include the checksum byte.	0..255	2A	Number of values corresponds to NumberOfResponse Bytes
<description>			Textual description		illegal character while waiting for LIN id	
<DomSigState>		10	LIN Dominant Signal		detected	

			only: Signal's state. Possible values: • "detected" • "continuing" • "finished"			
<DomSig-Length>		8	LIN Dominant Signal only: Length of the signal [in us]		5201	
<end of frame>		>=9	Absolute timestamp indicating end of LIN frame [in seconds]		1.020930	usually 6 decimal places
<ETFName>			ETF Info event only: Name of event-triggered frame		ETF_MotorStates	
<end of head-er>		>=9	Absolute timestamp indicating end of LIN header [in seconds]		1.022998	usually 6 decimal places
<Full time>			Duration of the entire frame [in bit times]		130	Up to v6.0 only
<HasData-bytes>		1	LIN Receive Error only: Flag indicating if there data byte values	0..1	1	
<Header Time>			Duration of the frame header [in bit times]		40	Up to v6.0 only
<header bau-drate>		12	Event's baudrate measured in header [in bits/sec as float value]		19230.769231	Accuracy: 6 decimal places
<ID>	>= 12	>= 12	Frame identifier or frame symbolic name		2d MotorCon-trol	
<IsDLCTimeou t>		1	LIN Receive Error only: Flag indicating if the error is a result of an attempt to resolve DLC	0..1	0	
<IsShortError>		1	LIN Receive Error only: Specifies the detail level of the event	0..1	1	
<MessageId>	4	4	16 bit message identifi-er	0..0xFFFF	1234	Available for dynamic frames only
<NAD>	2	3	Node address	0..0xFF	02	Available for dynamic frames only
<offending byte>	2	3	LIN Receive Error only: byte which caused the protocol violation	0..0xFF	72	

<response baudrate>		5	Event's baudrate measured in response [in bits/sec]		19230	
<simulated>		1	Flag indicating whether event has been simulated by CANoe/CANalyzer	0..1	0	
<schedule table index>			Schedule Mode Change event only: Index of schedule table		0	
<slave id>		2	FSM Identifier	0..63	2	Available with FSMs only
<SpikeLength>		6	LIN Spike Signal only: Length of the signal [in us]		56	
<start of frame>		>=9	Absolute timestamp indicating start of event [in seconds]		1.020930	usually 6 decimal places
<state>		3	Current state of the FSM	0..254	10	Available with FSMs only
<StateReason>	2	3	LIN Receive Error only: The lower 4 bits indicate the LIN hardware state at the time the error has occurred, while the upper 4 bits indicate the reason of the error Values for the state: 0: Bus idle 1: Waiting for Synch-Break 2: Waiting for Synch-Field 3: Waiting for frame ID 4-12: Waiting for data byte or checksum byte depending on the frame length. E.g. value 4 for FrameLength=0, value 12 for FrameLength=8 14: Consecutive event (i.e. event resulting from further data interpretation, after already notified error for first offending byte) 15: Not expected event (i.e. not WakeupRe-	0..0xFF	11	

			quest) during sleep mode. Occurs for LIN hardware in Master mode only Values for the reason: 0: Timeout 1: Received an unexpected byte violating protocol. In this case, <offending byte> field contains its value 2: Received a byte with framing error (with dominant stop bit). In this case, <offending byte> field contains its value 3: Unexpected Break field 4: Unidentified error			
<stop bit offset in header>		8	Early stop bit offset in frame header for UART timestamps [in ns]		26000	
<stop bit offset in response>		8	Early stop bit offset in frame response for UART timestamps [in ns]		26000	
<SupplierId>	4	4	Supplier Identifier	0..0xFFFF	2211	Available for dynamic frames only
<SyncBreak>		6	Break length [in ns]		937062	
<SyncDel>		6	Break delimiter length [in ns]		111750	
<sync break time>			Break length [in us and bit times]		937 us (17.820627 bits)	Up to v6.0 only
<sync delimiter time>			Break delimiter length [in us and bit times]		112 us (2.140654 bits)	Up to v6.0 only
<T0> … <T7>		>=9 (per value)	Absolute timestamp indicating end of data byte [in seconds]		1.023521	usually 6 decimal places Number of values corresponds to DLC
<Time>		>=9	Absolute or relative event's time [in seconds]		1234.5678	usually 6 decimal places
<TimeInter-		5 (per	LIN Sync Error only:		208	

val0>..<TimeInterval3>		value)	Time intervals [in us] detected between the falling signal edges of the Sync field			
<WakeupByte>	2	3	Wakeup event only: Byte value used by wakeup frame		80	
<WakeupLengthInfo>		1	Wakeup event only: Code of wakeup length: • 0 – OK • 1 – too short • 2 – too long		0	
<IsSlow-Response>		1	Short or slow response only: Non-zero, if the response was too slow; otherwise zero.		1	
<Response-WasInterrupt-edByBreak>		1	Short or slow response only: Non-zero, if the response was interrupted by a sync break; otherwise zero.		0	
<Width>		3-4	The width of a wakeup signal in microseconds. Valid for LIN 2.x only.		260	
<Disturb-anceType>			Disturbance event only: The type of disturbance that was executed.	dominant, recessive, header, bitstream, variableBitstream		
<ByteIndex>			A 0-indexed byte index (i.e. 0 is the first byte, 9 is the checksum in a dlc 8 frame). If referring to a header, 0 is the sync field and 1 is the PID.	0..9 in response, 0..1 in header		
<BitIndex>			A 0-indexed bit index (i.e. 0 is the first data bit, 8 is the stop bit, 9 is the first bit in inter-byte space).	0..255		
<BitOffsetInSixteenthBits>			An offset in 1/16th bits into a bit specified with <ByteIndex> and <BitIndex>	0..15		
<Disturb-anceLength-InSixteenth-Bits>			Disturbance event only: The length of a dominant or recessive dis-			

			turbance in units of 1/16th bits.			
<IDorFF>	2	3	A valid lin identifier or FF/255, if not applicable.	0..63, 255		