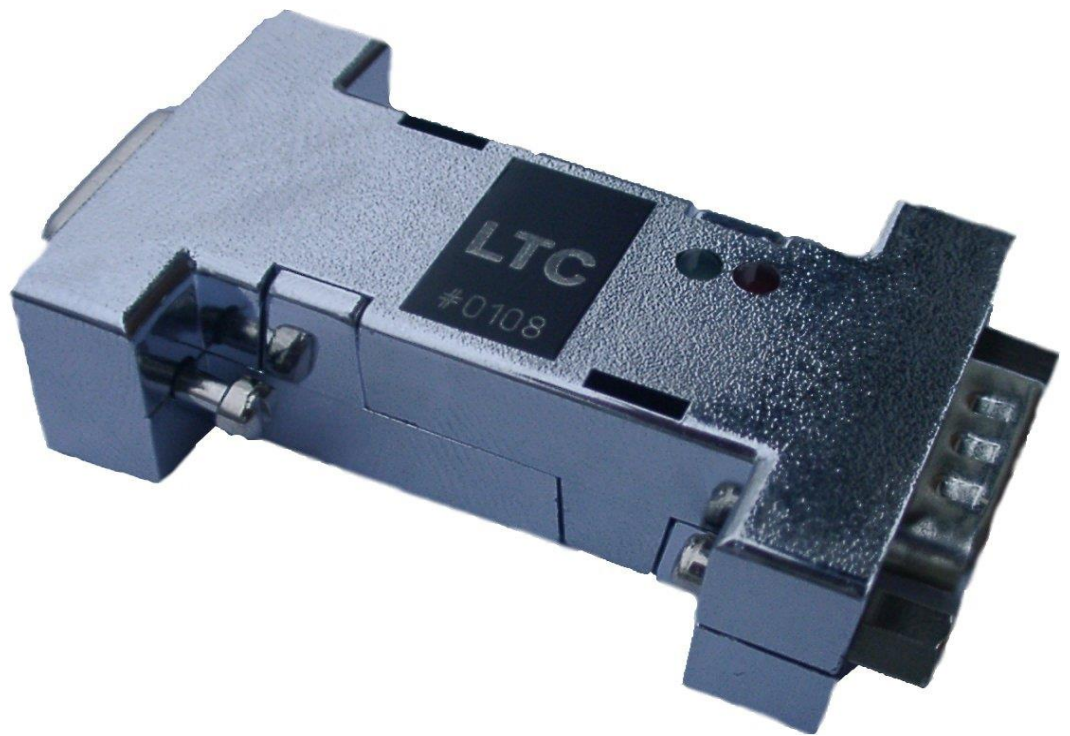


condalo

ELEKTRONIK & MECHATRONIK



LTC

LIN To CAN

condalo GmbH
Kohlstatt 3
86706 Lichtenau
08450 / 9264 – 0
info@condalo.de
www.condalo.de

0. Änderungsdokumentation	4
1. Hardware.....	5
1.1. Eigenschaften des LTC	5
1.2. Pinbelegung	5
1.3. Ausführungen:	6
1.4. Funktionen:	7
1.5. Aufbau der CAN- Botschaften des LTC.....	10
1.5.1. Modus1: 2 CAN-Botschaften für jede LIN-Nachricht.....	10
1.5.1.1. Datenbotschaft (erste CAN-Botschaft)	10
1.5.1.2. Zusatzinfos (zweite CAN-Botschaft).....	10
1.5.2. Modus2: Für jedes LIN-Byte eine CAN-Botschaft.....	11
1.5.2.1. Synchbotschaft.....	11
1.5.2.2. Datenbotschaft	11
1.5.2.3. Spike	12
1.5.2.4. Frameerror.....	12
1.5.3. Modus3: Für jede LIN-Nachricht eine CAN-Botschaft.....	13
1.5.3.1. Nachricht ohne erkannten Fehler	13
1.5.3.2. Paritybits oder Checksumme stimmt nicht mit der Berechnung überein ...	13
1.5.4. Modus4: 2 CAN-Botschaften für jede LIN-Nachricht.....	14
1.5.4.1. Datenbotschaft (erste CAN-Botschaft)	14
1.5.4.2. Zusatzinfos (zweite CAN-Botschaft).....	14
1.5.5. RS232-Modus1: Je RS232-Datenbyte eine CAN-Botschaft.....	15
1.5.5.1. CAN-Nachricht	15
1.5.6. RS232-Modus2: 8 RS232-Datenbytes werden zu einer CAN-Nachricht zusammengefasst	15
1.5.6.1. CAN-Nachricht	15
1.5.7. LIN-CAN-Gateway	16
1.5.8. Ausgabe für Digitaleingang	16
1.5.9. Ausgabe für Digitaleingang (Temperatursensor)	16
1.5.10. Ausgabe für Analog/Digital-Wandler.....	17
1.5.11. Ausgabe bei CAN-RESET	17
1.6. Filtereinstellung:.....	18
1.7. Slave-Funktion des LTC:.....	18
1.7.1. Konfiguration der Slave-Funktion	19
1.7.2. Konfiguration der Slave-Checksumme	19
1.7.3. Konfiguration der Slave-Antworten.....	20

1.8.	Serielle Ausgabe	20
1.9.	Leuchtanzeige	22
1.10.	Sleepmode	23
1.11.	Hinweise:	24
2.	Konfiguration	25
2.1.	Software: CCOview	25
2.2.	Konfiguration:	26
2.2.1.	Festlegung der CAN-Identifizier in einer INI-Datei	38
2.2.2.	Konfiguration abspeichern	39
2.3.	Firmwareupdate:	40
3.	Technische Daten	41
4.	Lieferumfang LTC	42
5.	Impressum	43
5.1.	Firmwareupdates und Programmneuheiten	43
5.2.	Fehlermeldungen und Verbesserungsvorschläge	43
5.3.	Anschrift	43

0. Änderungsdokumentation

Datum	Abschnitt	Art der Änderung	Bearbeiter	Ausgabe
30.09.05		LTC 1.0, LTC-Konfig 1.0.10	Kulzer	1.0.0
08.11.05		LTC 2.0, LTC-Konfig 1.0.10	Kulzer	1.0.1
14.11.05		Hinweise hinzugefügt	Kulzer	1.0.2
18.11.05		Hinzufügen der Auswertesoftware	Kulzer	1.0.3
07.02.06		Neue Firmware und neues CCOview	Kulzer	1.0.4
23.02.06		Sleepmode, Modus3, Onlineviewer hinzugefügt	Kulzer	1.0.5
21.03.06		Einstellbarer Identifier für CAN-Error-Ausgabe	Kulzer	1.0.6
07.08.06		Änderung Logo, Umschaltung Ausgabeformat bei AD-Wandler, zusätzliche Einstellung für Slavekonfiguration	Kulzer	1.0.7
04.10.06		Unterschiede von LTC und LTC2	Kulzer	1.0.8
24.11.06		Temperatursensor	Kulzer	1.0.9
11.01.07		Mastermodus	Kulzer	1.0.10
04.05.07		Bugfix CAN-Konfiguration, LIN-CAN-LIN-Gateway	Kulzer	1.0.11
21.01.08		Trigger bei CAN-Botschaft, CAN-LIN-Dongle	Kulzer	1.0.12
28.02.08		Erkennung WakeUp	Kulzer	1.0.13
25.02.09		RS-232-Modus und LIN-Checksumme	Kulzer	1.0.14
03.02.10		Vertauschen von Datenbytes im Donglemodus	Kulzer	1.0.15
21.04.10		Beschreibung Abtastverhalten AD-Wandler	Kulzer	1.0.16
10.03.10		Änderung der CAN-Error-Botschaft	Kulzer	1.0.17
24.05.11		Bugfix bei CAN-Offset für Donglefunktion	Kulzer	1.0.18
01.08.11		Veränderte Errorinformationen	Kulzer	1.0.19
13.02.12		Einstellbare Synchbreaklänge / Dynamische CHK	Kulzer	1.0.20
29.01.13		Änderung der CHK-Einstellung via CAN-Bus CHK-Version für jede Slaveantwort konfigurierbar	Kulzer	1.0.23
14.02.14		Ergänzung Pullup im Master und Donglemodus	Kulzer	1.0.24
04.08.14		Hinweise erweitert	Kulzer	1.0.25
23.02.14		Hinweise zum RS232-Modus	Kulzer	1.0.26

1. Hardware

1.1. Eigenschaften des LTC

Mit diesem Gerät werden LIN-Botschaften auf den CAN umgesetzt. Dadurch ist es möglich diese Daten mit einem CAN-Tool (wie den condalo-Datenlogger) zu visualisieren, aufzuzeichnen und zu analysieren.

Der LTC verfügt über einen LIN-Anschluss, dessen Daten auf den CAN umgesetzt werden. Es ist auch möglich diese Daten parallel über die serielle Schnittstelle und der mitgelieferten Software „CCOview“ am PC zu visualisieren.

Des weiteren verfügt der LTC über zwei analoge und einen digitalen Eingang deren Spannungspegel bzw. Zustände auf dem CAN bzw. seriell ausgegeben werden können.

1.2. Pinbelegung

Pinbelegung Sub-D Buchse:



Pin1: Analoger Eingang(Kanal 1)
Pin2: TX0- UART
Pin3: RX0- UART
Pin4: LIN
Pin5: GND

Pin6: TX1- UART
Pin7: Digitaler Eingang bzw. RTS (PC-UART)
Pin8: Analoger Eingang (Kanal 2)
Pin9: Versorgung 12 V

Pinbelegung Sub-D Stecker:



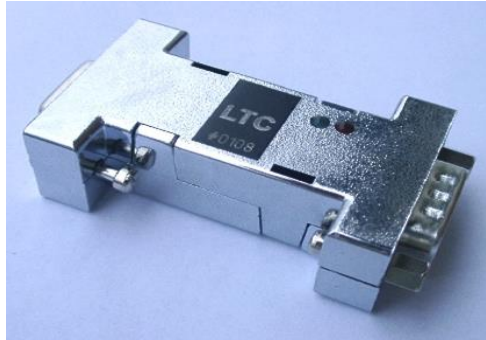
Pin1: ----
Pin2: CAN-L
Pin3: GND
Pin4: ----
Pin5: GND

Pin6: GND
Pin7: CAN-H
Pin8: ----
Pin9: Versorgung 12 V

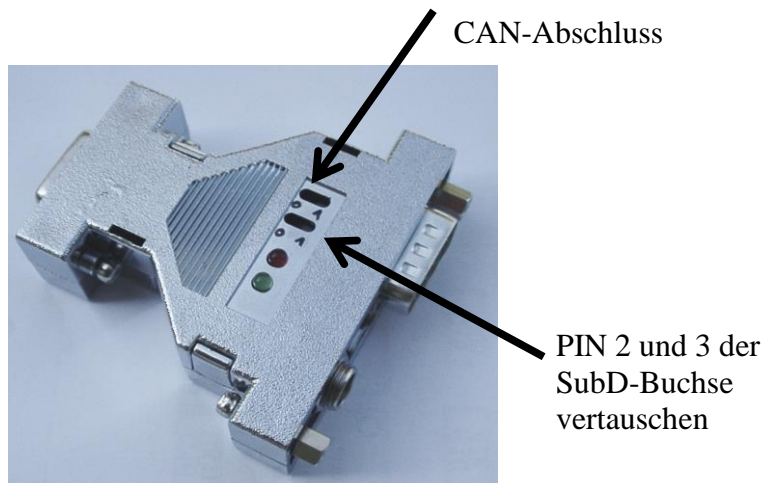
Die Versorgungsspannung kann wahlweise an der Buchse oder am Stecker angeschlossen werden. Beide 12 V Eingänge sind durch Dioden voneinander entkoppelt.

1.3. Ausführungen:

LTC: Pinbelegung wie beschrieben



LTC2: Pinbelegung wie beschrieben



- zusätzlichem Spannungsanschluss über Kleinspannungsstecker:

- zuschaltbarer CAN-Busabschluss bzw. Pintauch an UART0:

Schalter in Stellung „1“:

- kein CAN-Busabschluss

- Standardbelegung für PIN 2 und 3 an der SUBD-Buchse (RX, TX von UART0)

Schalter in Stellung „0“:

- 120 Ohm CAN-Busabschluss mit Symmetrierung

- Pin 2 und 3 bei der SUBD-Buchse sind vertauscht (RX,TX von UART0)



Optional kann ein Temperatursensor an den LTC angesteckt werden. Er wird über den PIN2 der LTC-Buchse mit Spannung versorgt und über den „Digitalen Eingang“ abgetastet. Diese Funktion kann bei den Einstellungen für den „Digitalen Eingang“ konfiguriert werden.

Temperaturmessungen schneller als 140ms sind nicht möglich.

1.4. Funktionen:

Der LTC setzt alle ankommenden LIN-Signale in CAN-Botschaften um. Die CAN- Botschaften können mit vier verschiedenen Modi aufgebaut werden.

In allen Modi können konfigurierbare Slave-Antworten auf den LIN-Bus gesendet werden.

Der LTC kann auch als einfacher „LIN-Master“ verwendet werden. Für diese Funktion können 10 Schedule-Einträge (LIN-ID und zugehörige Länge) erstellt werden.

Es ist ebenfalls möglich, den LTC als Gateway von LIN nach CAN bzw. von CAN nach LIN zu benutzen.

MODUS 1:

Alle Datenbytes der LIN-Nachricht, werden in einer CAN-Botschaft zusammengefasst. Diese CAN-Botschaft hat einen einstellbaren Identifier. Zusatzinformationen wie Identifier, Checksumme, Bitlänge(Baudrate), der eingestellte Prescaler des internen Timers und der Zeitversatz zwischen Detektion eines LIN-Bytes (RX-Interrupt) und dem Senden der Botschaft auf den CAN (Start der Übertragung) sind in einer weiteren Botschaft (eingestellter Identifier + 1) zusammengefasst. Die gesammelten Bytes werden nach einem einstellbaren Timeout oder spätestens beim nächsten Synchbreak auf den CAN bzw. seriell ausgegeben.

MODUS 2:

Hier wird jedes ankommende LIN-Byte in einer eigenen CAN-Botschaft gesendet. Wurde ein Synch-Break erkannt, wird eine CAN-Botschaft mit dem eingestellten Identifier gesendet. Diese enthält Informationen wie Identifier, Bitlänge(Baudrate), der eingestellte Prescaler des internen Timers und der Zeitversatz zwischen Detektion eines LIN-Bytes (RX-Interrupt) und dem Senden der Botschaft auf den CAN (Start der Übertragung). Bei einem erkannten Datenbyte wird dieses in einer CAN-Botschaft (eingestellter Identifier + 1) zusammen mit dem Zeitversatz ausgegeben. Ein Spike (eingestellter Identifier + 2) oder ein Frameerror (eingestellter Identifier + 4) wird ebenfalls auf dem CAN angezeigt.

MODUS 3:

Hier wird für jede LIN-Nachricht eine CAN-Nachricht gesendet. Der CAN-Identifier ist der Identifier der LIN-Nachricht + einstellbarer Offset. In diesem Modus werden die Checksumme und die Paritybits nachgerechnet. Tritt dabei eine Unstimmigkeit mit den mitgelesenen Daten auf, werden diese Botschaften mit einem anderen, auch einstellbaren Offset für den Identifier gesendet.

MODUS 4:

Wie Modus 1, jedoch wird hier in der Nachricht mit den Zusatzinformationen die Header-Länge, Full-Länge, die Baudrate (Bitzählwert), sowie Errorinformationen angegeben.

Im MODUS 1, MODUS 3 und MODUS 4 ist es möglich mit einem Filter verschiedene LIN-Botschaften von der Übertragung auf den CAN auszuschließen.

Zusätzlich gibt es noch einen RS232-Modus. Mit diesem können Daten auf der LIN-Leitung direkt in CAN-Nachrichten umgewandelt werden. Diese Daten müssen jedoch 0V - 12V Spannungsspiegel haben und dürfen nicht schneller als 20 kBit/s sein.

Der Analog/Digitalwandler misst in konfigurierbaren Zeitabständen (1ms bis 65535ms) die anliegende Spannung und gibt den Wert in einer CAN-Botschaft mit einstellbarem Identifier aus. Diese Zeiten können eingehalten werden, egal welche Funktionen des LTC aktiviert sind. Im Highspeed-Modus können Abtastzeiten bis ca. 200µs erreicht werden. Diese ändern sich aber wenn andere

Funktionen des LTC ein bzw. ausgeschaltet werden. Es ist möglich Botschaften nur auszugeben, wenn sich die Spannung um einen einstellbaren Wert zwischen zwei Messungen ändert. Die Spannung wird in mV angegeben und die Ausgabenorm (Big Endian bzw. Little Endian) ist konfigurierbar.

Der Signalzustand des digitalen Eingangs kann in konfigurierbaren Zeitabständen auf den CAN ausgegeben werden, oder nur wenn eine Flankenänderung aufgetreten ist.

1.5. Aufbau der CAN- Botschaften des LTC

1.5.1. Modus1: 2 CAN-Botschaften für jede LIN-Nachricht

1.5.1.1. Datenbotschaft (erste CAN-Botschaft)

CAN-ID: frei Wählbar (eingestellter Identifier)
Botschaft: Dateninhalt der LIN-Botschaft
Länge: 0..8 Byte

Parameter	Byte	Bit	Erläuterung
Dateninhalt	0..x	0..63	Dateninhalt der LIN-Botschaft

1.5.1.2. Zusatzinfos (zweite CAN-Botschaft)

CAN-ID: eingestellter Identifier + 1
Botschaft: Zusatzinfos zur LIN-Botschaft
Länge: 7 Byte

Parameter	Byte	Bit	Erläuterung
LIN-Identifizier	0	0..7	Identifizier der LIN-Botschaft (incl. Paritybits)
Checksumme	1	8..15	gelesene Checksumme der LIN-Botschaft
Bitlänge (Baudrate)	2	16..23	H-Byte der Bitlänge
“	3	23..31	L-Byte der Bitlänge (Wert * 675ns = Bitlänge in μ s)
Prescaler des internen Timers	4	32..39	Eingestellter Timerprescaler im μ Controller (für Berechnung Zeitversatz)
Zeitversatz zwischen Empfang und senden	5	40..47	Wert * 23 μ s = Zeitversatz (bei Prescaler 0x02)
Errorinformationen	6	48..55	Bit 0: gesetzt bei CHK-Fehler Bit 1: gesetzt bei Parityfehler Bit 3: gesetzt bei fehlender Slave-Antwort

1.5.2. Modus2: Für jedes LIN-Byte eine CAN-Botschaft

1.5.2.1. Synchbotschaft

CAN-ID: frei Wählbar (eingestellter Identifier)
Botschaft: Dateninhalt der LIN-Botschaft
Länge: 6 Byte

Parameter	Byte	Bit	Erläuterung
Synchbreaklänge	0	0..7	H-Byte
“	1	8..15	L-Byte (Wert * 675ns = Synchbreaklänge in s)
Bitlänge (Baudrate)	2	16..23	H-Byte der Bitlänge
	3	23..31	L-Byte der Bitlänge (Wert * 675ns = Bitlänge in s)
Prescaler des internen Timers	4	32..39	Eingestellter Timerprescaler im μ Controller (für Berechnung Zeitversatz)
Zeitversatz zwischen Empfang und senden	5		Wert * 23 μ s = Zeitversatz (bei Prescaler 0x02)

1.5.2.2. Datenbotschaft

CAN-ID: eingestellter Identifier + 1
Botschaft: Dateninhalt
Länge: 2 Byte

Parameter	Byte	Bit	Erläuterung
Dateninhalt	0	0..7	Dateninhalt des aktuellen Bytes der LIN-Botschaft
Zeitversatz zwischen Empfang und senden	1	8..15	Wert * 23 μ s = Zeitversatz (bei Prescaler 0x02)

1.5.2.3. Spike

CAN-ID: eingestellter Identifier + 2
 Botschaft: Spike in aktuellem Byte erkannt
 Länge: 2 Byte

Parameter	Byte	Bit	Erläuterung
Dateninhalt	0	0..7	Dateninhalt des aktuellen Bytes der LIN-Botschaft
Zeitversatz zwischen Empfang und senden	1	8..15	Wert * 23µs = Zeitversatz (bei Prescaler 0x02)

1.5.2.4. Frameerror

CAN-ID: eingestellter Identifier + 4
 Botschaft: Frameerror (kein Synchbreak) durch die UART erkannt
 Länge: 2 Byte

Parameter	Byte	Bit	Erläuterung
Dateninhalt	0	0..7	Dateninhalt des aktuellen Bytes der LIN-Botschaft
Zeitversatz zwischen Empfang und senden	1	8..15	Wert * 23µs = Zeitversatz (bei Prescaler 0x02)

1.5.3. Modus3: Für jede LIN-Nachricht eine CAN-Botschaft

1.5.3.1. Nachricht ohne erkannten Fehler

CAN-ID: LIN-ID + einstellbarer Offset
Botschaft: Dateninhalt der LIN-Botschaft
Länge: 0..8 Byte

Parameter	Byte	Bit	Erläuterung
Dateninhalt	0..8	0..64	Dateninhalt der LIN-Nachricht

1.5.3.2. Paritybits oder Checksumme stimmt nicht mit der Berechnung überein

CAN-ID: LIN-ID + Fehleroffset
Botschaft: Dateninhalt der LIN-Botschaft
Länge: 0..8 Byte

Parameter	Byte	Bit	Erläuterung
Dateninhalt	0..8	0..63	Dateninhalt der LIN-Nachricht

1.5.4. Modus4: 2 CAN-Botschaften für jede LIN-Nachricht

Wie Modus 1, jedoch andere Zusatzinformationen

1.5.4.1. Datenbotschaft (erste CAN-Botschaft)

CAN-ID: frei Wählbar (eingestellter Identifier)
Botschaft: Dateninhalt der LIN-Botschaft
Länge: 0..8 Byte

Parameter	Byte	Bit	Erläuterung
Dateninhalt	0..x	0..63	Dateninhalt der LIN-Botschaft

1.5.4.2. Zusatzinfos (zweite CAN-Botschaft)

CAN-ID: eingestellter Identifier + 1
Botschaft: Zusatzinfos zur LIN-Botschaft
Länge: 7 Byte

Parameter	Byte	Bit	Erläuterung
LIN-Identifier	0	0..7	Identifier der LIN-Botschaft (incl. Paritybits)
Checksumme	1	8..15	gelesene Checksumme der LIN-Botschaft
Header-Länge	2	16..23	in Bitzeiten (von der fallenden Flanke des Synchbreaks bis Empfang des Identifiers)
Full-Länge	3	23..31	in Bitzeiten (von der fallenden Flanke des Synchbreaks bis Empfang letztes Byte)
Länge bis zum Senden auf den CAN	4	32..39	in Bitzeiten (von der fallenden Flanke des Synchbreaks bis absetzen der CAN-Nachricht)
Bitzeit	5	40..47	Zählerwert für 1 Bit (Wert * 675ns = Bitlänge in s)
Errorinformationen	6	48..55	Bit 0: gesetzt bei CHK-Fehler Bit 1: gesetzt bei Parityfehler Bit 3: gesetzt bei fehlender Slave-Antwort

1.5.5. RS232-Modus1: Je RS232-Datenbyte eine CAN-Botschaft

1.5.5.1. CAN-Nachricht

CAN-ID: eingestellter Identifier
Länge: 2 Byte

Parameter	Byte	Bit	Erläuterung
RS232-Datenbyte	0	0..7	Empfangenes Datenbyte
Parity-Info	1	8..15	0= kein Parity-Error 1= Parity-Error

1.5.6. RS232-Modus2: 8 RS232-Datenbytes werden zu einer CAN-Nachricht zusammengefasst

1.5.6.1. CAN-Nachricht

CAN-ID: eingestellter Identifier
Länge: 1-8 Byte

Parameter	Byte	Erläuterung
RS232-Datenbyte	0 - 7	Empfange RS232-Daten

Durch Pausen in der RS232-Übertragung können auch CAN-Nachrichten entstehen, die weniger als 8 Datenbyte enthalten (Timeoutüberwachung)

1.5.7. LIN-CAN-Gateway

Botschaften vom LIN zum CAN werden je nach oben eingestelltem Modus auf den CAN-Bus gesendet. CAN-Botschaften die auf den LIN gesendet werden sollen, müssen wie folgt aufgebaut sein.

CAN-ID: LIN-ID + eingestellter Offset
Botschaft: Dateninhalt der LIN-Botschaft
Länge: 0..8 Byte

Parameter	Byte	Bit	Erläuterung
Dateninhalt	0..8	0..64	Dateninhalt der LIN-Nachricht

Über den Baumzweig „CAN-Einstellungen“ können die Datenbytes der jeweiligen Ausgabebotschaft vertauscht werden.

1.5.8. Ausgabe für Digitaleingang

CAN-ID: frei Wählbar (eingestellter Identifier)
Botschaft: Zustand des Digitaleingang
Länge: 1 Byte

Parameter	Byte	Bit	Erläuterung
Pegel am Digitaleingang	0	0..7	0x01 (High-Pegel oder steigende Flanke) 0x00 (Low-Pegel oder fallende Flanke)

1.5.9. Ausgabe für Digitaleingang (Temperatursensor)

CAN-ID: frei Wählbar (eingestellter Identifier)
Botschaft: Temperatur
Länge: 2 Byte

Parameter	Byte	Bit	Erläuterung
Temperatur	0	0..7	H-Byte bzw. L-Byte der Messung (je nach eingestellter Ausgabenorm)
“	1	8..15	L-Byte bzw. H-Byte der Messung (je nach eingestellter Ausgabenorm)

Das Messergebnis wird in 0,1°C bzw. 0,1°F ausgegeben.

1.5.10. Ausgabe für Analog/Digital-Wandler

CAN-ID: frei Wählbar (eingestellter Identifier)
Botschaft: Spannung am Analogeingang
Länge: 2 Byte

Parameter	Byte	Bit	Erläuterung
Messwert	0	0..7	H-Byte bzw. L-Byte der Messung (je nach eingestellter Ausgabenorm)
“	1	8..15	L-Byte bzw. H-Byte der Messung (je nach eingestellter Ausgabenorm)

Das Messergebnis wird in mVolt ausgegeben.

1.5.11. Ausgabe bei CAN-RESET

Bei einem Fehler auf dem CAN (Acknowledge, Bitstuffing) wird ein Reset des CAN-Controllers durchgeführt. Damit festgestellt werden kann, wann und wie oft solch ein Fehler vorkommt, wird bei jedem CAN-Reset eine CAN-Botschaft abgesetzt.

CAN-ID: frei Wählbar (eingestellter Identifier)
Botschaft: CAN-Reset
Länge: 1 Byte

Parameter	Byte	Bit	Erläuterung
Byte 1	0	0..7	0x01 = Fehler auf dem CAN 0x02 = Überlauf des FIFO-Puffers

Bei einer Aufzeichnung mit dem condalo-Datenlogger II setzt sich eine Botschaft wie folgt zusammen:

Eine Botschaft im CCO-File hat immer 16 Byte. Jede Botschaft hat
5 Byte Zeitstempel,
1 Byte Datenlogger- Kanal,
2 Byte CAN-ID mit Datenlänge,
maximal 8 Byte Dateninhalt.

Ist die Datenlänge kleiner als 8 Byte so sind nur die Bytes bis zur Datenlänge gültig.

1.6. Filtereinstellung:

Der LTC verfügt über einen Sperrfilter mit Code und Maske. Code, Maske und der empfangene Identifier werden Bitweise verglichen. An der Stelle, wo die Maske eine logische „1“ hat, muss der empfangene Identifier den gleichen Bitwert wie der Code haben, damit die Botschaft mit diesem Identifier auf den CAN gesendet wird.

z.B. empfangene ID: 0x32, Maske: 0xFF, Code:0x32

	0011 0010	(Code)
	<u>1111 1111</u>	(Maske)
=	0011 0010	(dieser Wert muss der empfangenen ID entsprechen)
		hier wird nur ID 0x32 durchgelassen

oder: empfangene ID: 0x32, Maske: 0xFC, Code:0x32

	0011 0010	(Code)
	<u>1111 1100</u>	(Maske)
=	0011 00xx	(dieser Wert muss von Bit2 bis Bit7 der ID entsprechen)
		hier werden ID 0x30 bis 0x33 durchgelassen

1.7. Slave-Funktion des LTC:

Mit dem LTC können konfigurierbare Slave-Antworten auf den LIN-Bus gesendet werden. Dazu sind für die einzelnen Identifier die Antwortlänge, und die gewünschten Daten einzugeben. Die benötigte Checksumme wird vom LTC berechnet. Dazu muss bei der Konfiguration des LTC bei den „LIN-Einstellungen“ die verwendete LIN-Version angegeben werden.

Ist die automatische Erkennung der LIN-Chk-Version aktiv, wird im Slavemodus die Version 2.x für die Berechnung hergenommen.

Diese Funktion des LTC kann auch über den CAN zur Laufzeit konfiguriert werden. Die Konfigurationsbotschaften sind wie folgt aufgebaut:

1.7.1. Konfiguration der Slave-Funktion

CAN-ID: Offset für CAN-Konfiguration + 0x40
Botschaft: SlaveFunktion EIN / AUS
Länge: 2 Byte

Parameter	Byte	Bit	Erläuterung
Befehl 1	0	0..7	0x00
Befehl 2	1	8..15	0x01 (SlaveFunktion EIN) 0x00 (SlaveFunktion AUS)

CAN-ID: Offset für CAN-Konfiguration + 0x40
Botschaft: Konfiguration Speichern/Auslesen
Länge: 2 Byte

Parameter	Byte	Bit	Erläuterung
Befehl 1	0	0..7	0x01
Befehl 2	1	8..15	0x00 (Slave-Daten löschen) 0x01 (Einstellungen ins EEprom übernehmen) 0x02 (Slave-Daten auslesen) 0x03 (CHK-Einstellung auslesen)

1.7.2. Konfiguration der Slave-Checksumme

CAN-ID: Offset für CAN-Konfiguration + 0x40
Botschaft: Invertierung der Checksumme EIN/AUS
Länge: 2 Byte

Parameter	Byte	Bit	Erläuterung
Befehl 1	0	0..7	0x02
Befehl 2	1	8..15	0x01 (normale Checksumme) 0x02 (invertierte Checksumme)

Diese Änderung wird auch im Eeprom abgespeichert und ist dadurch nach einem Power-ON-Reset noch verfügbar.

1.7.3. Konfiguration der Slave-Antworten

CAN-ID: Offset für CAN-Konfiguration + LIN-ID
Botschaft: Slave-Antwort
Länge: 0..8 Byte

Parameter	Byte	Bit	Erläuterung
Slave-Antwort	0..8	0..63	Dateninhalt der Slave-Antwort

1.8. Serielle Ausgabe

Es ist möglich die gleichen Daten wie auf dem CAN-Bus auch über die serielle Schnittstelle auf dem PC darzustellen.

Im Onlineviewer der mitgelieferten PC-Software „CCOview“ werden die Daten in korrekter zeitlicher Abfolge mit einem Zeitstempel versehen angezeigt.

Neben dem Zeitstempel enthält die dargestellte Botschaft den Kanal (woher sie kommt), die Länge und die darzustellenden Informationen.

CCDview 1.0.3 vom 31.01.2006

Datei Einstellungen Ansicht Hilfe

Datensatz

#	Zeitstempel	Kanal	Länge	Daten
4642	10.417012	LIN	8	11 C1 C2 C3 C4 C5 C6 66
4643	10.481453	LIN	10	D3 55 55 55 55 55 55 55 55
4644	10.481477	LIN	10	32 11 22 33 44 55 66 77 83 9E
4645	10.481488	LIN	4	CF B1 B2 9B
4646	10.481494	LIN	8	8E A1 A2 A3 A4 A5 A6 27
4647	10.542956	LIN	8	11 C1 C2 C3 C4 C5 C6 66
4648	10.542977	LIN	10	D3 55 55 55 55 55 55 55 55
4649	10.542983	LIN	10	32 11 22 33 44 55 66 77 83 9E
4650	10.542994	LIN	4	CF B1 B2 9B
4651	10.604979	LIN	8	8E A1 A2 A3 A4 A5 A6 27
4652	10.605002	LIN	8	11 C1 C2 C3 C4 C5 C6 66
4653	10.666967	LIN	10	D3 55 55 55 55 55 55 55 55
4654	10.666984	LIN	10	32 11 22 33 44 55 66 77 83 9E
4655	10.666994	LIN	4	CF B1 B2 9B
4656	10.667000	LIN	8	8E A1 A2 A3 A4 A5 A6 27
4657	10.730456	LIN	8	11 C1 C2 C3 C4 C5 C6 66
4658	10.730476	LIN	10	D3 55 55 55 55 55 55 55 55
4659	10.730482	LIN	10	32 11 22 33 44 55 66 77 83 9E
4660	10.730491	LIN	4	CF B1 B2 9B
4661	10.792957	LIN	8	8E A1 A2 A3 A4 A5 A6 27
4662	10.792976	LIN	8	11 C1 C2 C3 C4 C5 C6 66
4663	10.857957	LIN	10	D3 55 55 55 55 55 55 55 55
4664	10.857980	LIN	10	32 11 22 33 44 55 66 77 83 9E
4665	10.857991	LIN	4	CF B1 B2 9B
4666	10.857996	LIN	8	8E A1 A2 A3 A4 A5 A6 27
4667	10.919909	LIN	8	11 C1 C2 C3 C4 C5 C6 66
4668	10.919929	LIN	10	D3 55 55 55 55 55 55 55 55
4669	10.919935	LIN	10	32 11 22 33 44 55 66 77 83 9E
4670	10.919944	LIN	4	CF B1 B2 9B
4671	10.982409	LIN	8	8E A1 A2 A3 A4 A5 A6 27
4672	10.982428	LIN	8	11 C1 C2 C3 C4 C5 C6 66
4673	11.044917	LIN	10	D3 55 55 55 55 55 55 55 55
4674	11.044938	LIN	10	32 11 22 33 44 55 66 77 83 9E
4675	11.044945	LIN	4	CF B1 B2 9B
4676	11.044951	LIN	8	8E A1 A2 A3 A4 A5 A6 27
4677	11.107966	LIN	8	11 C1 C2 C3 C4 C5 C6 66
4678	11.107986	LIN	10	D3 55 55 55 55 55 55 55 55
4679	11.107992	LIN	10	32 11 22 33 44 55 66 77 83 9E
4680	11.108002	LIN	4	CF B1 B2 9B

9 [0][2] COM:2 Baudrate:115200 ID:0x0C

1.9. Leuchtanzeige

Der LTC verfügt über zwei Leuchtdioden die den momentanen Betriebszustand anzeigen.

Die rote Leuchtdiode ist im Normalbetrieb ununterbrochen an.

Werden Fehler auf dem CAN festgestellt, gibt diese einen Blinkcode (kurze Unterbrechungen des Leuchtsignals) aus.

Ist kein LIN-Signal vorhanden, so blinkt die Leuchtdiode zweimal kurz nach einer langen Pause auf.

Sind die „EIN“ und „AUS“-Phase der roten LED gleich lang, so ist die Standardkonfiguration im EEprom gespeichert.

Die grüne Leuchtdiode blinkt bei je nach konfigurierter Ausgabe entweder bei einer ankommenden LIN-Botschaft, bei der CAN-Ausgabe, bei einer digitalen oder analogen Messung auf.

1.10. Sleepmode

Der LTC verfügt über zwei verschiedene Sleepmodes.

Sleepmode 1:

In diesem Modus misst der LTC die Spannung am Analogeingang 1. Fällt die Spannung unter einen konfigurierbaren Wert, gibt der LTC CAN-Botschaften mit der anliegenden Spannung aus. Die Stromaufnahme in diesem Modus beträgt ca. 1500 – 1800 μ A.

Sleepmode 2:

Stellt der LTC keine Buskommunikation für eine Einstellbare Zeit fest, so wechselt er in den Standby-Mode. Durch einen LOW-Pegel auf der Busleitung wacht der LTC auf und arbeitet mit der zuvor verwendeten Konfiguration. Die Stromaufnahme in diesem Modus beträgt ca. 180 – 220 μ A.

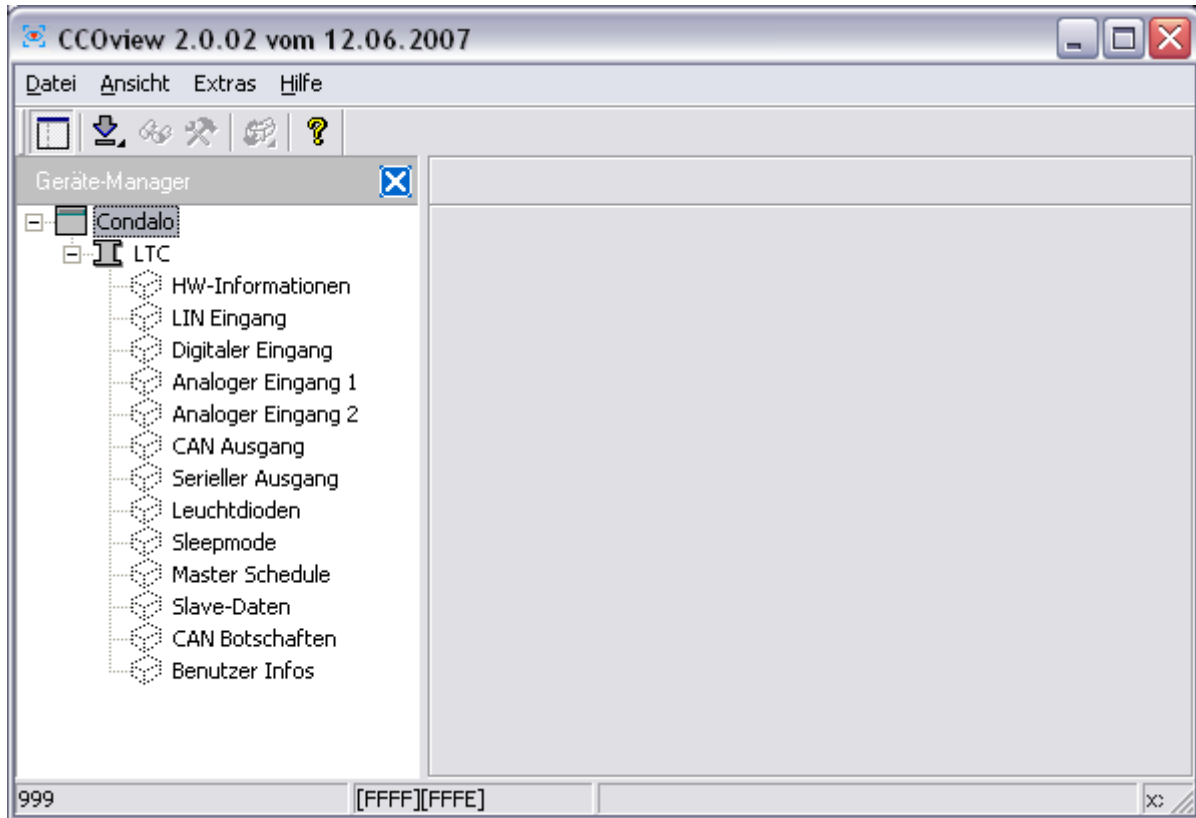
Während aktiver Buskommunikation und solange die Konfigurationssoftware in Betrieb ist, wechselt der LTC **nicht** in den Sleepmode.

1.11. Hinweise:

- Der LTC verfügt über keinen eigenen Busabschluss. Es ist deshalb darauf zu achten, dass der CAN-Bus mit 120 Ohm (zwischen CAN-High und CAN-Low) an beiden Enden abgeschlossen wird.
- Mindestens ein CAN-Teilnehmer muss das Acknowledge für die CAN-Botschaft senden, damit der LTC seine Botschaften absetzen kann.
- Die LIN-Daten werden auf die UART des Controllers weitergegeben. Datenbytes werden dabei richtig erkannt und können somit weiterverwendet werden. Bei einem Synchbreak stellt die UART einen Frameerror fest. Ist nun der Abstand zwischen den letzten zwei Flanken groß genug für einen Synchbreak (13 Bitzeiten LOW) ist ein Synchbreak erkannt worden.
- Im Master- oder Dongle-Modus muss ein 1 kOhm Widerstand zwischen die LIN-Leitung und Versorgungsspannung extern geschaltet werden.
- Der Mastermodus und der Donglemodus dürfen auf keinen Fall gleichzeitig aktiviert werden.
- Ist der Zeitliche Abstand zwischen zwei Flanken auf dem LIN-Bus kleiner als eine Bitzeit, so wird von einem Spike auf der LIN-Leitung ausgegangen.
- Wird für die Verbindung des PC mit dem LTC kein standardisiertes serielles Kabel (1:1) verwendet, so ist darauf zu achten, dass
 - die RX-Leitung des PC (PIN2),
 - die TX-Leitung des PC (PIN3),
 - der RTS (PIN7) und
 - GND (PIN5)angeschlossen sind, damit eine Konfiguration bzw. ein Firmwareupdate durchgeführt werden kann.
- Der Prescaler in der CAN-Ausgabebotschaft in den verschiedenen Modi wird für die Auswertung mit der LIN-Auswertesoftware benötigt. Er ist von außen nicht einstellbar.
- Ein Frameerror wird immer durch die eingebaute UART des Mikrocontrollers erkannt.
- Für den Temperatursensor muss mindestens PIN2, PIN5 und PIN7 des Sensors mit der LTC-Buchse verbunden werden. Die Ausgabe der Temperatur im CCOview (Onlineausgabe) ist nicht möglich.
- Die fehlerfreie Nutzung eines USB-RS232-Dongle ist erst ab der Firmwareversion 2.0 gewährleistet.

2. Konfiguration

2.1. Software: CCOview



Mit Hilfe der Software „CCOview“ können die Einstellungen des LTC verändert, die Firmware upgedatet, und die laufende Kommunikation auf dem Bildschirm mitverfolgt werden.

Ist der LTC mit dem seriellen Anschluss eines PCs verbunden und an eine Spannungsversorgung angeschlossen, so erscheint beim Start dieses Programms folgendes Fenster.

Alle momentan eingestellten Konfigurationen werden ausgelesen und angezeigt.

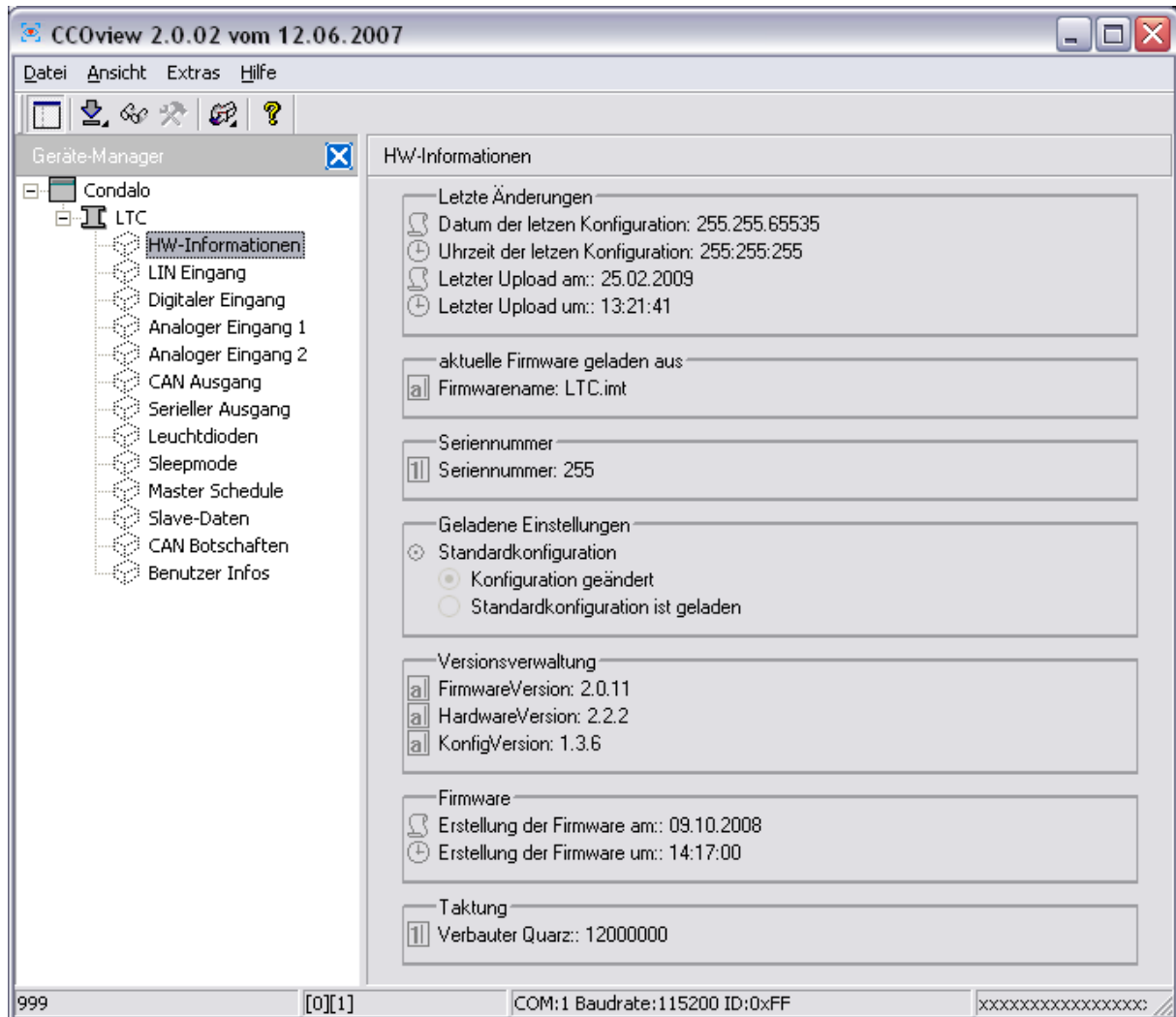
Die einzelnen Konfigurationsmöglichkeiten sind in den verschiedenen Zweigen des Konfigurationsbaumes untergebracht.

An diese Software können gleichzeitig mehrere Geräte angeschlossen werden. Dazu werden die verfügbaren seriellen Schnittstellen immer wieder abgefragt.

Die dabei nicht benötigten Schnittstellen, können im Menü unter „Extras → Einstellungen“ ab- bzw. ausgewählt werden.

2.2. Konfiguration:

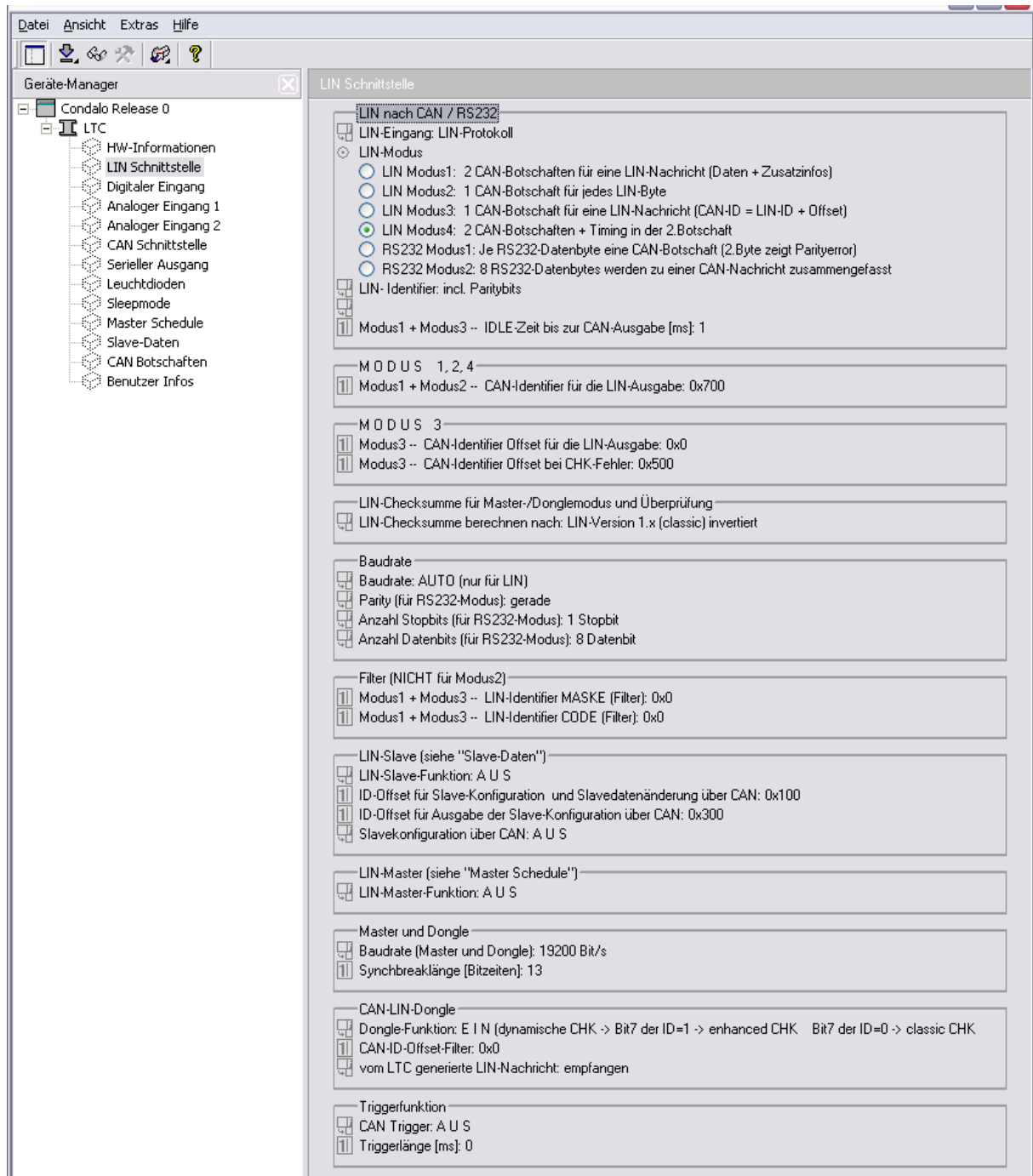
HW-Informationen:



Im Zweig HW-Informationen werden verschiedenen Infos zur Firmware, Seriennummer und zur Hardware hinterlegt. Hier kann nichts verändert werden.

Mögliche Einstellungen sind:

LIN Eingang:



Einstellungen:

- LIN-Eingang: Auswertung des LIN-Protokolls oder RS232-Daten
- LIN-Modus: In welcher Form die Daten auf den CAN gesendet werden
- LIN-Identifizierer können mit Paritybits bzw. ohne in der CAN-Nachricht darstellen.
- IDLE-Zeit: Zeit, nach der das Ende der LIN-Botschaft erkannt wird -> Senden der CAN-Botschaft

- Identifizierer für die CAN-Ausgabe (MODUS 1 + MODUS 2 + MODUS 4)

- Offset für CAN-Identifizierer im MODUS 3
- Offset für CAN-Identifizierer bei Checksummenfehler oder Parityfehler im MODUS 3

- Version (Spezifikation) nach der die Checksumme berechnet wird. Gilt für die Überprüfung von empfangenen LIN-Nachrichten im MODUS 3. Wird hier eine invertierte CHK ausgewählt, so wird diese in allen Funktionen (Slave-Mode, Dongle-Mode) invertiert berechnet. Die Version der CHK kann in diesen Modi aber gesondert ausgewählt werden.

- Baudrate des LIN-Busses (auch automatische Erkennung möglich). Für Dongle bzw. Mastermodus gilt diese Einstellung nicht.
- Einstellungen für die RS232-Schnittstelle (nur gültig, wenn kein LIN-Auswertung aktiv ist) Gilt nicht für die Kommunikation mit dem CCOview.

- Sperrfilter für die LIN-Identifizierer (nicht im Aufzeichnungsmodus MODUS 2 verwendbar)
- Slave-Funktion bzw. Slave-Konfiguration über CAN
Ist die Slave-Konfiguration aktiv, kann die Slave-Funktion über den CAN-Bus ein- bzw. ausgeschaltet werden. Ebenfalls können die LIN-Slave-Daten zur Laufzeit über den CAN verändert werden. Im Dongle-Modus und im Master-Slave-Modus kann über den CAN auch zwischen normaler und invertierter Checksumme umgeschaltet werden.

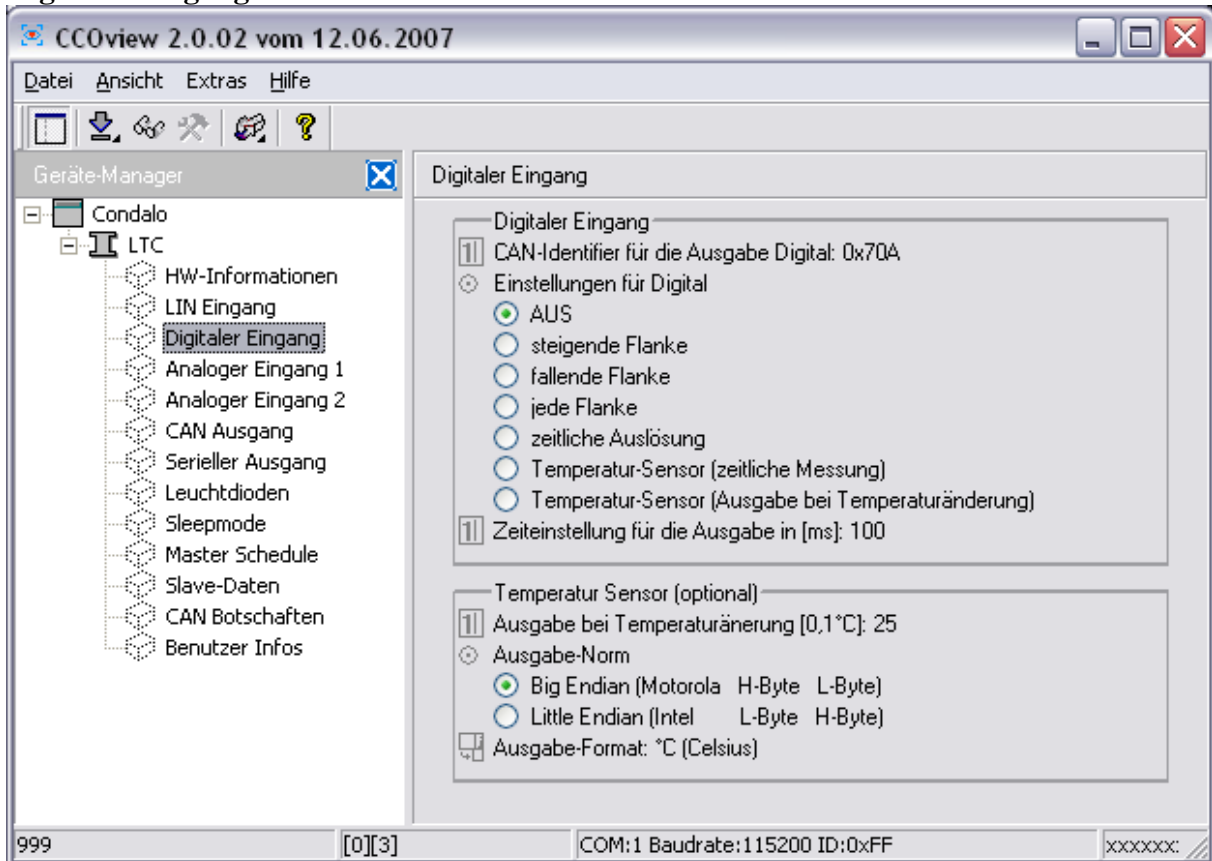
- Master-Funktion: Hier wird die „Master Schedule“ kontinuierlich abgearbeitet

- Baudrate für Master und Donglemodus gültig
- Im Master und Donglemodus kann die Syncbreaklänge hier eingestellt werden (in Bitzeiten)

- CAN-LIN-Dongle: CAN-Botschaften die zu dem eingestellten CAN-Filter passen, werden direkt an den LIN-Bus gesendet. Botschaften auf dem LIN, werden je nach eingestelltem Modus zum CAN gesendet. (hier wird die Baudrate der Masterfunktion verwendet.)
Ist dynamische CHK ausgewählt, dann entscheidet Bit7 der ID über die LIN-Checksumme (Bit7=1 → enhanced CHK Bit7=0 → classic CHK). Dieses Bit wird in der LIN-Nachricht durch die Paritybits wieder überschrieben
- Es ist möglich, LIN-Botschaften die durch den CAN-Bus generiert wurden von der LIN-Auswertung auszuschließen.

- CAN-Trigger: Unter der Auswahl „CAN-Botschaften“ können CAN-Signale definiert werden, bei deren Empfang die UART-TX-Leitung für eine einstellbare Zeit von -12V auf +12V geschaltet wird.

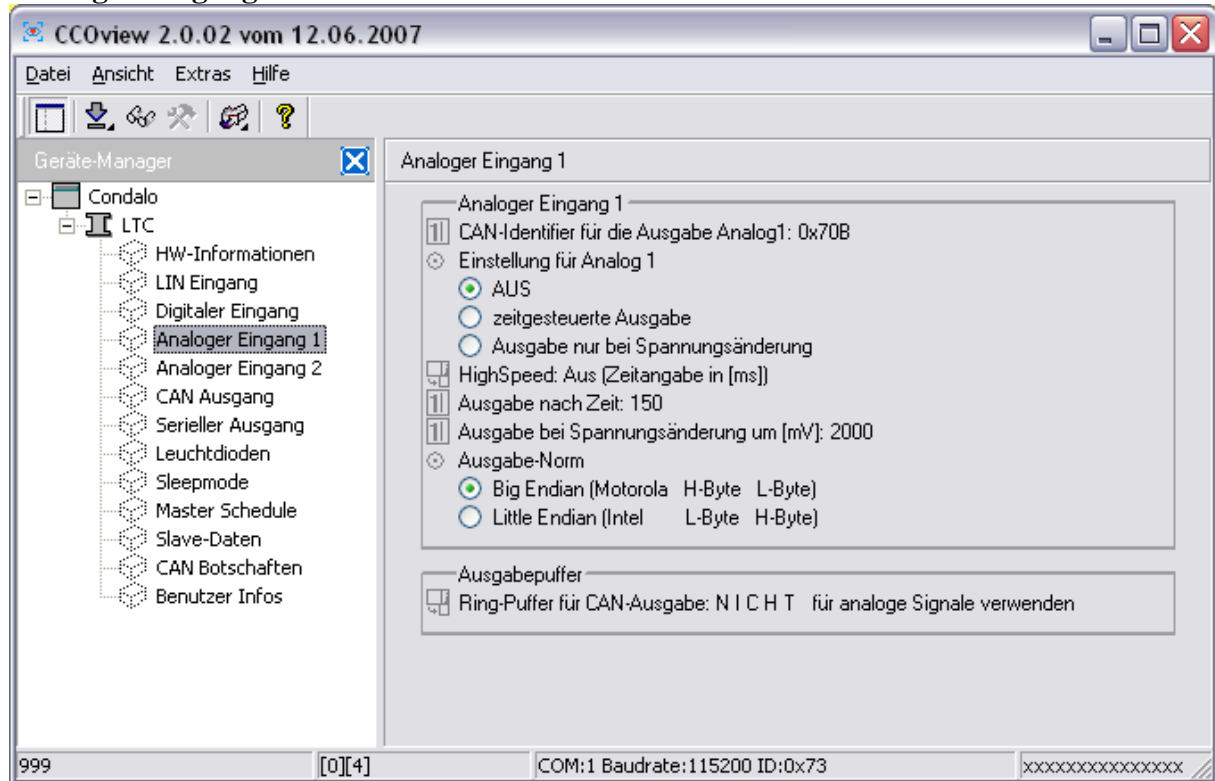
Digitaler Eingang:



Einstellungen:

- Identifizier für die CAN-Ausgabe des Digitalwertes bzw. der Temperatur
- Auslöseereignis für die Digital-Ausgabe bzw. Verwendung als Eingang für den Temperatursensor
- Zeit in ms für die zyklische Messung (für den Temperatursensor keine Werte unter 150ms verwenden)
- Wert für die Temperaturänderung (Ausgabe bei Temperaturänderung) in 0,1 °C
- Ausgabe-Norm für den Temperatursensorwert
- Ausgabeformat für die Temperaturausgabe

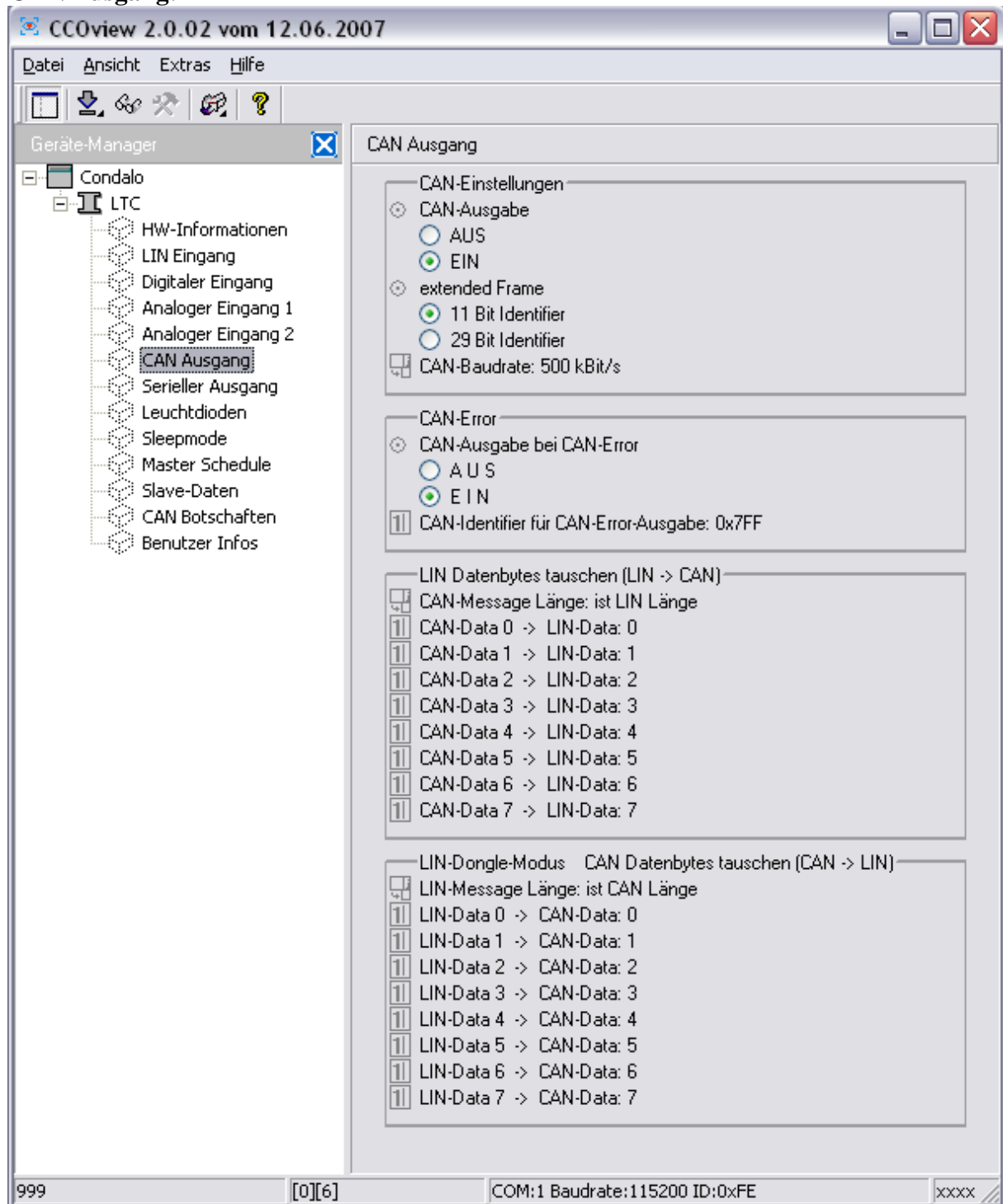
Analoger Eingang:



Einstellungen:

- Identifizier für die CAN-Ausgabe des Analogwertes
- Auslöseereignis für die Ausgabe
- Zeit in ms für die zyklische Messung (1ms bis 65535 im Normal-Modus)
Zykluszeiten ab ca. 200µs sind im Highspeed-Modus möglich (wenn keine anderen Funktionen des LTC aktiv sind).
- Spannung in mV, bei deren Änderung zur vorherigen Ausgabe eine CAN-Ausgabe erfolgen soll.
- Ausgabe-Norm für den Messwert.
- Der LTC verfügt über einen Ringpuffer für zu sendende CAN-Botschaften. Falls dieser auch für die Analogwerte benutzt werden soll, kann dies unter diesem Punkt eingestellt werden. Wird der Ringpuffer **nicht** verwendet, werden Analogwerte erst auf den CAN gesendet, wenn die Daten aus dem Ringpuffer verarbeitet sind. So kann z.B. bei hoher CAN-Buslast ein Vorrang für LIN-Botschaften konfiguriert werden.

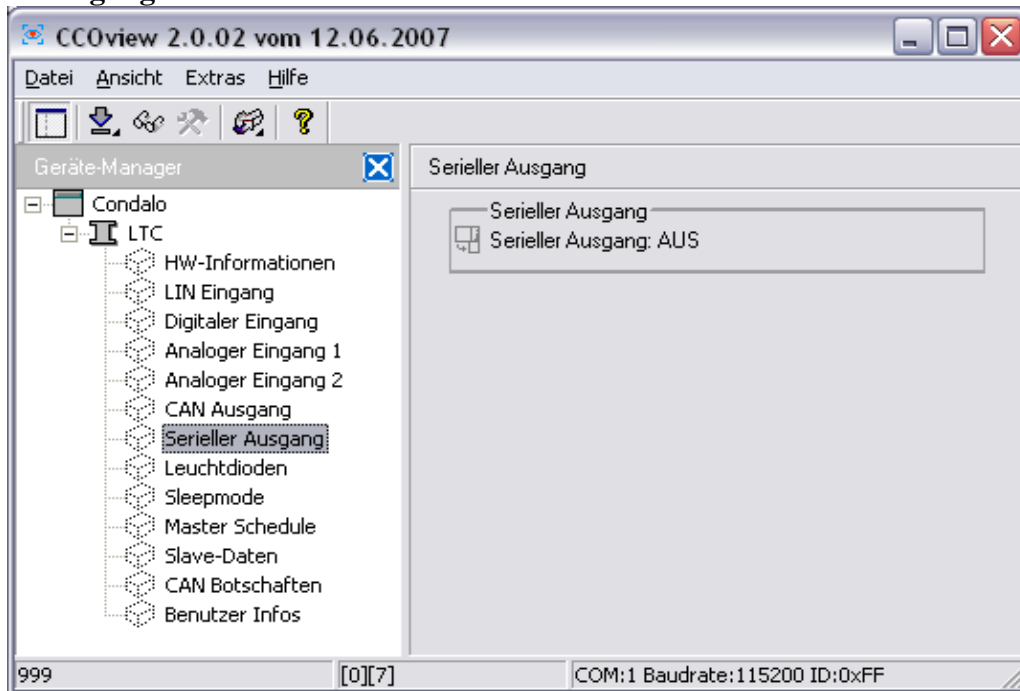
CAN Ausgang:



Einstellungen:

- CAN-Ausgabe EIN bzw. AUS
 - 11Bit oder 29Bit Identifier
 - CAN-Baudrate
 - CAN-Ausgabe einer gesonderten Botschaft, nachdem ein Fehler auf dem CAN festgestellt wurde.
 - Identifier für die gesonderte CAN-Botschaft
- Im LIN-Donglemodus können sowohl empfangene CAN-Datenbytes in der LIN-Ausgabe Nachricht, als auch empfangene LIN-Datenbytes in der CAN-Ausgabe-Nachricht vertauscht werden.
- Es ist auch möglich, die Länge der Ausgabebotschaft unabhängig von der Eingangsbiotschaft zu verändern.

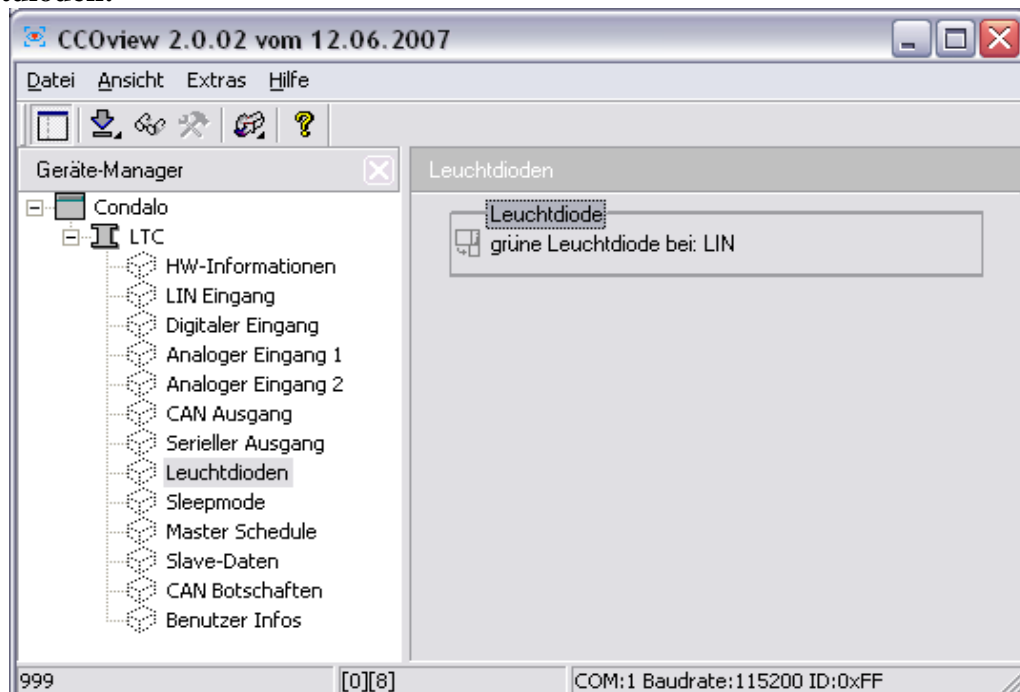
Serieller Ausgang:



Einstellungen:

Im seriellen Trace können LIN, Digitale, Analoge Werte dargestellt werden.

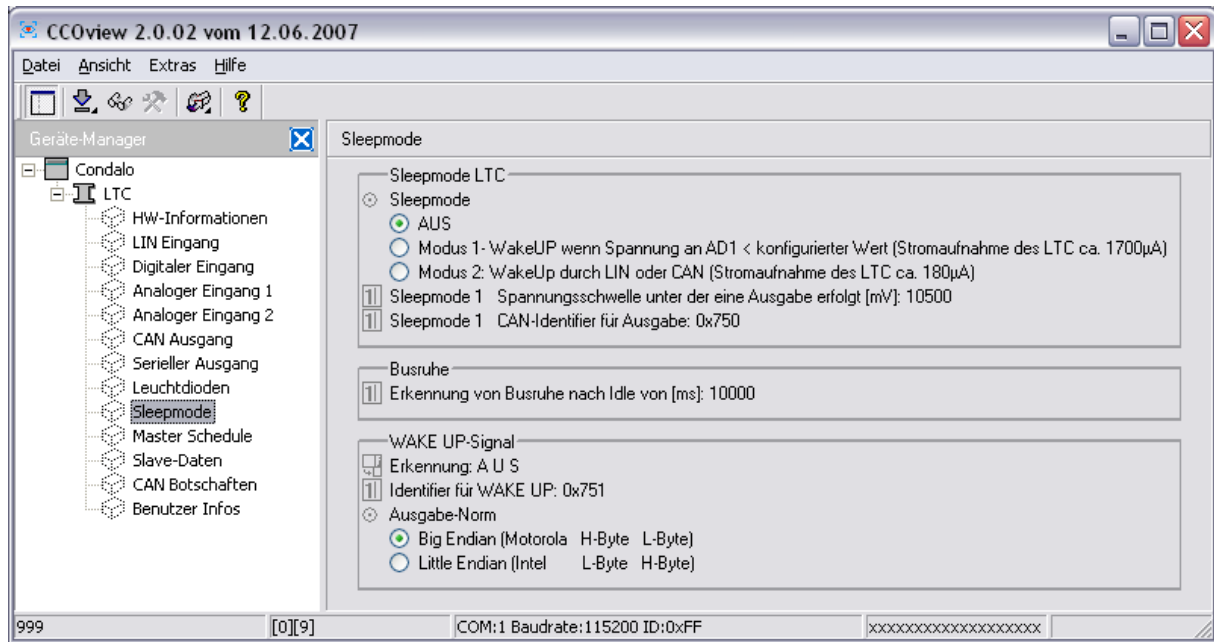
Leuchtdioden:



Einstellungen:

Ereignis, bei dem die Grüne LED aufleuchten soll.

Sleepmode:

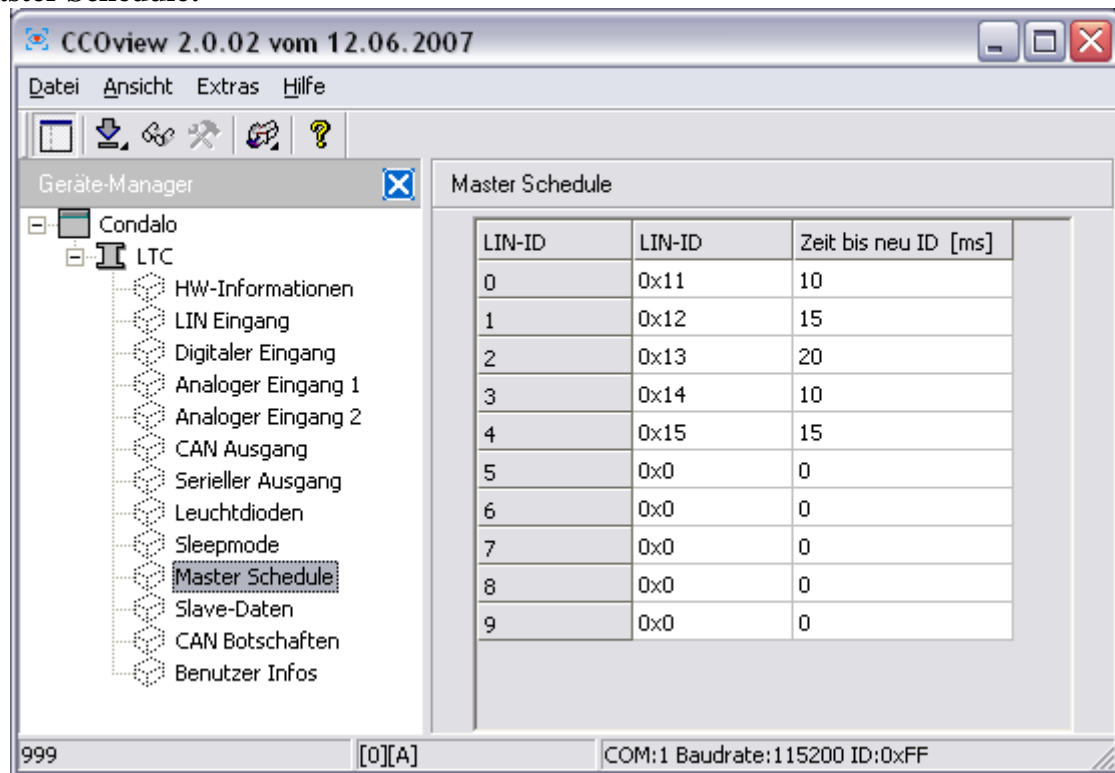


Einstellungen:

- Auswahl des Sleepmodes.
- Spannung, unterhalb der eine CAN-Ausgabe erfolgt (Sleepmode 1)
- CAN-ID für die Spannungsausgabe (Sleepmode 1)
- Erkennung von Busruhe nach einer Idlezeit von ...ms.
(Gilt für Sleepmode des LTC und für Wakeuperkennung)
- WAKE UP Erkennung (NICHT, wenn der LTC selbst im Sleepmode ist)
Gesonderte Botschaft mit der Länge des WakeUp-Signals wird gesendet
- Identifizier für Ausgabe der Länge des WAKE-UP-Signals
- Ausgabenorm für die Längenangabe

Sleepmode: nachdem kein LIN-Signal **und** kein CAN-Signal mehr am LTC anliegt **und** das CCOview beendet ist.

Master Schedule:



Einstellungen:

Diese Funktion muss unter dem Zweig „**LIN Eingang**“ eingeschaltet werden.

- LIN-ID die vom Master gesendet werden sollen.

Eingabe kann mit bzw. ohne Paritybits erfolgen. Die Paritybits werden auf jeden Fall neu berechnet. Die Anzeige ist immer ohne Paritybits.

- ZEIT: vom Beginn des Synchbreaks der LIN-Botschaft mit dieser ID bis zum Beginn des Synchbreaks der nächsten ID.

Felder, bei denen als Zeit „0“ eingestellt ist, werden beim Senden der einzelnen ID's übersprungen.

Slave-Funktion:

CCOview 3.2.0 vom 22.02.2012

Geräte-Manager

- Condalo Release 0
 - LTC
 - HW-Informationen
 - LIN Schnittstelle
 - Digitaler Eingang
 - Analoger Eingang 1
 - Analoger Eingang 2
 - CAN Schnittstelle
 - Serieller Ausgang
 - Leuchtdioden
 - Sleepmode
 - Master Schedule
 - Slave-Daten
 - CAN Botschaften
 - Benutzer Infos

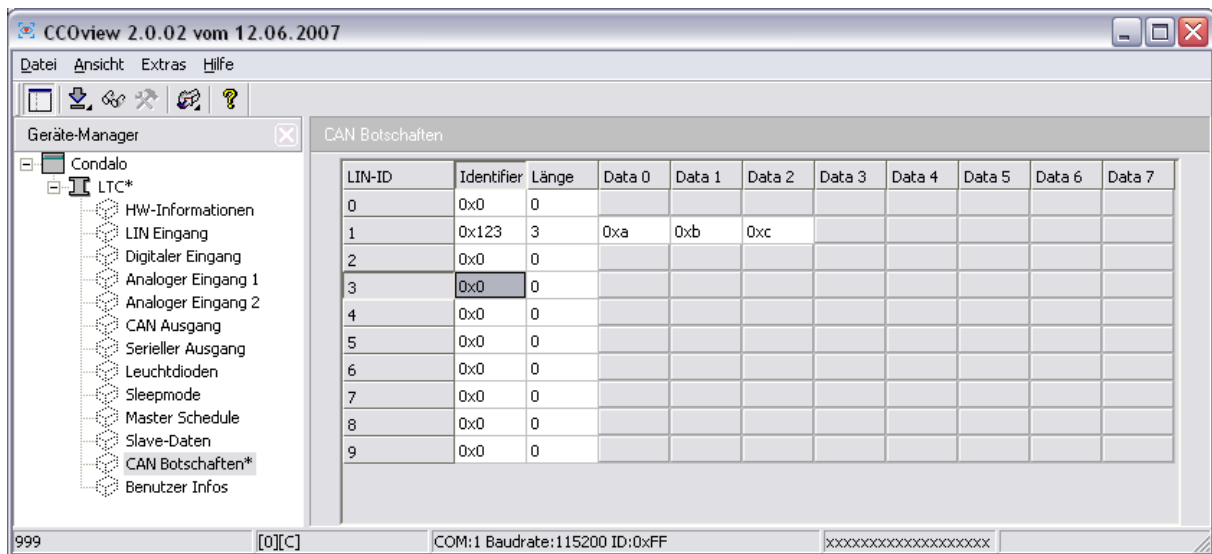
Slave-Daten

LIN-ID	CHK-Version	Datenlänge	Daten 1	Daten 2	Daten 3	Daten 4	Daten 5	Daten 6	Daten 7	Daten 8	Checksumme
9	2	0									0x0
10	2	0									0x0
11	2	0									0x0
12	2	0									0x0
13	2	0									0x0
14	2	0									0x0
15	2	0									0x0
16	2	0									0x50
17	1	2	0x1	0x2							0xFF
18	1	4	0x2	0x3	0x4	0x5					0xFF
19	2	6	0x3	0x4	0x5	0x6	0x7	0x8			0x2C
20	2	0									0x0
21	2	0									0x0
22	2	0									0x0
23	2	0									0x0
24	2	0									0x0
25	2	0									0x0

Einstellungen:

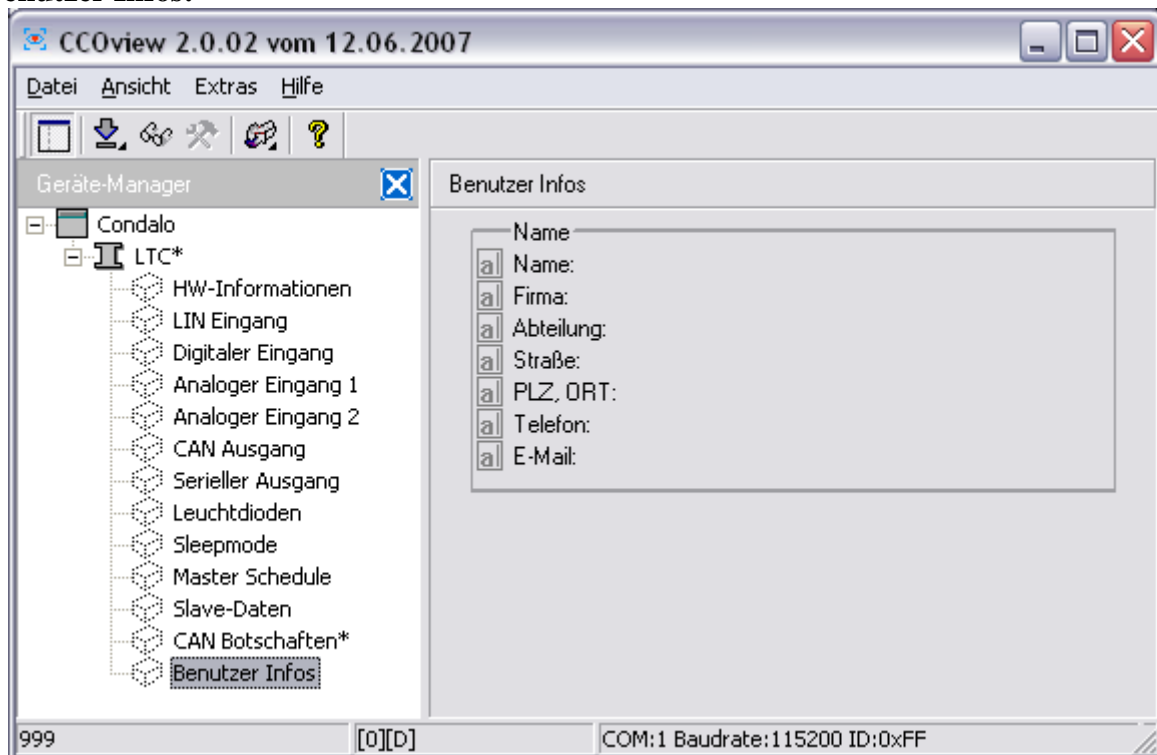
- Diese Funktion muss unter dem Zweig „**LIN Schnittstelle → Slave Funktion**“ eingeschaltet werden.
- Die Anzeige der LIN-ID ist dezimal.
- Die CHK-Version (1 oder 2) gibt die Berechnungsvorschrift für die gesendete Checksumme an. Ist die Slavekonfiguration über CAN aktiviert, wird für die neuen Slavedaten die hier eingestellte Berechnungsvorschrift verwendet.
- Wird als Datenlänge „0“ eingetragen, wird auf diese LIN-ID keine Slave-Antwort gesendet.
- Die Checksumme für die LIN-Botschaft wird beim Schreiben der Konfiguration automatisch berechnet. Ist unter „LIN Schnittstelle → LIN-Checksumme berechnen nach“ eine invertierte CHK ausgewählt, wird auch hier die invertierte CHK berechnet (aber nach der Version in Spalte 1 dieser Tabelle).

CAN Botschaften:



- Diese Funktion wird im Zweig „LIN Eingang“ unter Triggerfunktion eingeschaltet. Wird auf dem CAN-Bus eine Nachricht empfangen, die mit einer der eingestellten Botschaften übereinstimmt, wird die RS232-TX-Leitung von -12V auf +12V geschaltet.

Benutzer Infos:



Wurde eine Einstellung eines Zweiges verändert, so wird an das jeweilige Bauelement ein * - Stern angehängt. Durch den Befehl „Speichern dauerhaft“ bzw. „Speichern temporär“ im Kontextmenü des Baumes kann die Änderung zum LTC übertragen werden.

Die Änderungen können sowohl im EEPROM gespeichert als auch nur temporär(bis zum nächsten Reset) verändert werden.

2.2.1. Festlegung der CAN-Identifizier in einer INI-Datei

Es ist möglich, symbolische Namen für CAN-Identifizier mit Hilfe einer INI-Datei zu vergeben. Befindet sich die INI-Datei im selben Ordner wie „CCOview.exe“, so können bei den CAN-Identifiern entweder die vorgelegten symbolischen Namen, sowie auch freie Eingaben verwendet werden.

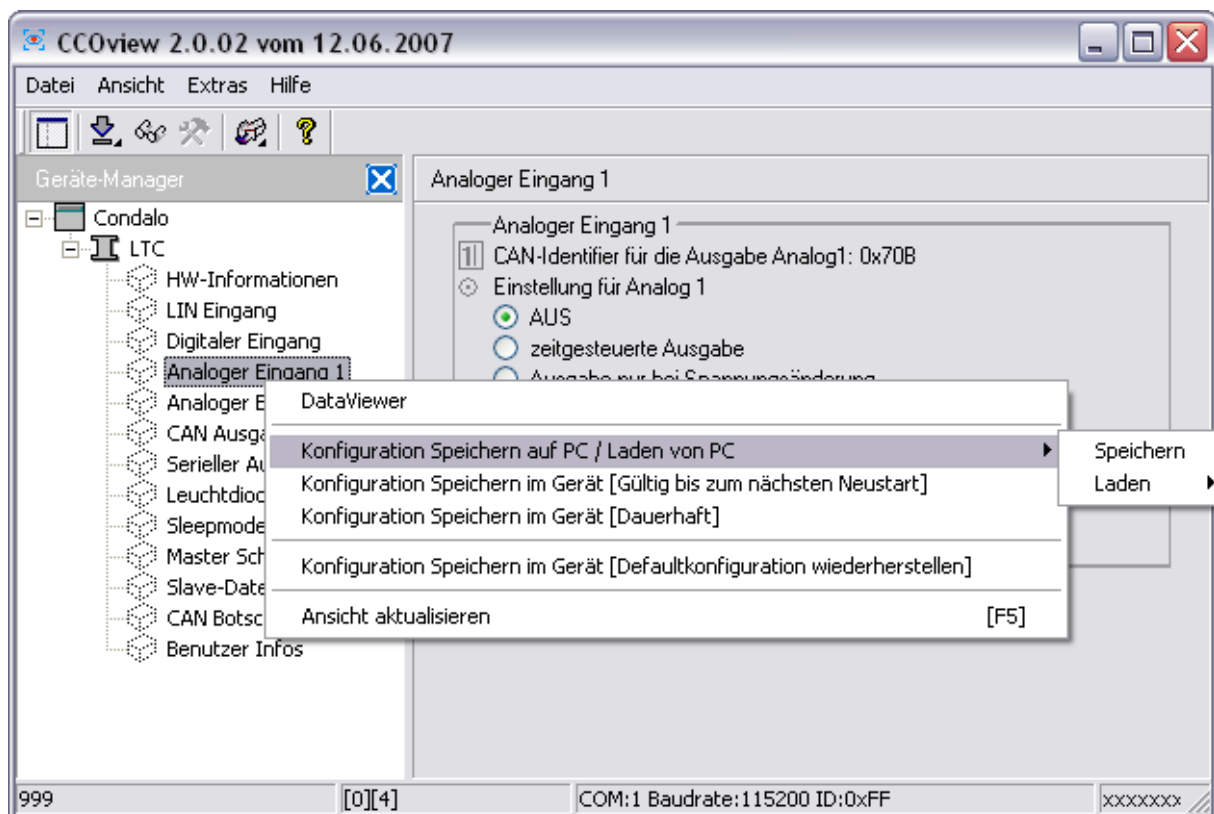
Die INI-Datei ist wie folgt aufgebaut. Als Dateiname ist xTC.ini zu wählen.

```
[Head]
Version=10
NumberOfEntries=3
[Item0]
Device=LTC
Name=Modus1 + Modus2 -- CAN-Identifizier für die LIN-Ausgabe
ID=[NameA1;0x203][NameA2;100][NameA3;200][NameA4;0x123]
[Item1]
Device=LTC
Name=Modus3 -- CAN-Identifizier Offset für die LIN-Ausgabe
ID=[NameB1;0x203][NameB2;100][NameB3;200][NameB4;0x123]

[Item2]
Device=BTC
Name=Identifizier für die BSD- Ausgabe
ID=[NameA1;0x203][NameA2;100][NameA3;200][NameA4;0x123]
```

Device =	Gerät, für welches die Einstellung gedacht ist (*.IMT)
NumberOfEntries =	Anzahl der in der INI-Datei enthaltenen ID-Datensätze
Item0; Item1; ...ItemN =	Nummerierung der einzelnen ID-Datensätze
Name =	Überschrift der CAN-ID Einstellung im CCOview
ID=	[symbolischer Name; zugehörige CAN-ID]

2.2.2. Konfiguration abspeichern

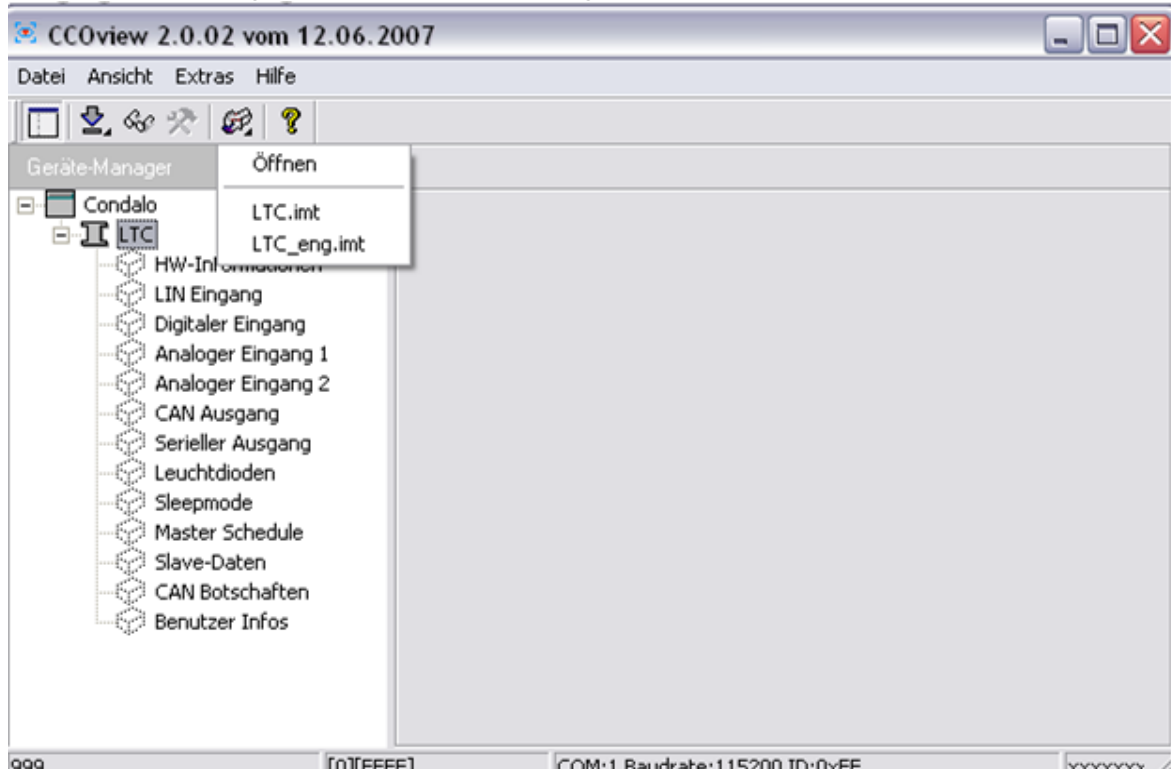


Die Verschiedenen Einstellungen des LTC lassen sich in einem Konfigurationsfile abspeichern. Damit ist es möglich, verschiedenen Geräten die gleiche Einstellung zuzuweisen.

Im Kontextmenü des Konfigurationsbaumes können die zuvor abgespeicherten Daten ausgewählt und auf den LTC geladen werden.

2.3. Firmwareupdate:

Die Firmware (*.hex-Datei) muss in den selben Ordern kopiert werden, in dem sich auch die Programmdatei des Konfigurationstools befindet.



Über den Button "Update" kann eine neue Firmware auf den LTC geladen werden. Die gewünschte Firmware wird im Untermenü dieses Befehls ausgewählt. Dateien, die sich im gleichen Ordner befinden,

3. Technische Daten

Funktionen:

LIN nach CAN 2.0A bzw. CAN 2.0B
Messung von zwei Spannungen 0 .. 25V
Überwachung eines digitalen Eingangs
Betriebsanzeige über zwei LEDs

Anschlüsse des Gerätes:

9pol Sub-D Buchse: Spannungsversorgung, LIN, 2x Analog,
1x Digital, RS232 für Konfiguration
9pol Sub-D Stecker: Spannungsversorgung, CAN

Sonstiges:

Betriebsspannung: 12 V DC (7V DC - 30V DC)
Stromaufnahme max. 40mA
Temperaturbereich: -40°C bis +85°C
Abmessungen: 63mm x 17mm x 33mm
Gewicht: 22 Gramm
Material: Kunststoffgehäuse mit Metallüberzug
Analoge Eingänge: 0 - 25 V (9:1 vorgeteilt)
10 Bit Auflösung
± 2 LSB absolute Genauigkeit
Messzyklus 1ms bis 65535ms, im
Highspeed-Modus minimal 200µs
Digitaler Eingang: max. 50 V ohne zeitliche Beschränkung

Anschlusschema:

Konfiguration:

PC
(mindestens
Pin 2, 3, 5, 7)



12 V (Pin 9)
GND (Pin 5)

LIN To CAN-Umsetzung:

12V (Pin 9)
GND (Pin 5)
LIN-Bus (Pin 4)



CAN
Pin 2,7
(evtl. 120 Ohm
Abschluss)

4. Lieferumfang LTC

- 1 LTC (LIN To CAN Konverter)
- 1 CD mit Software
- 1 Dokumentation

5. Impressum

5.1. Firmwareupdates und Programmneuheiten

finden Sie in unserem Downloadbereich unter

<http://www.condalo.de/pid130.html>

5.2. Fehlermeldungen und Verbesserungsvorschläge

markus.kulzer@condalo.de oder info@condalo.de

5.3. Anschrift

condalo GmbH

Kohlstatt 3
86706 Lichtenau
Deutschland

Tel.: 08450 - 9264 - 0
Fax: 08450 - 9264 - 50
E-Mail: info@condalo.de
Web: www.condalo.de