



STC

Serial To CAN

Inhaltsverzeichnis

| | |
|--|----|
| 0. Änderungsdokumentation..... | 3 |
| 1. Hardware..... | 4 |
| 1.1. Eigenschaften des STC..... | 4 |
| 1.2. Pinbelegung..... | 4 |
| 1.3. Ausführungen:..... | 5 |
| 1.4. Funktionen:..... | 6 |
| 1.5. Aufbau der CAN- Botschaften des STC..... | 7 |
| 1.5.1. CAN-Botschaften für die RS232-Funktion des STC..... | 7 |
| 1.5.1.1. Datenbotschaft | 7 |
| 1.5.1.2. Frameerror..... | 7 |
| 1.5.2. Ausgabe für Digitaleingang..... | 7 |
| 1.5.3. Ausgabe für Analog/Digital-Wandler..... | 8 |
| 1.5.4. Ausgabe bei CAN-RESET..... | 8 |
| 1.5.5. CAN-Dongle-Funktion..... | 9 |
| 1.6. Serielle Ausgabe..... | 12 |
| 1.7. Leuchtanzeige..... | 13 |
| 1.8. Sleepmode..... | 14 |
| 1.9. Hinweise:..... | 15 |
| 2. Konfiguration:..... | 16 |
| 2.1. Software: CCOview..... | 16 |
| 2.1.1. Festlegung der CAN-Identifizier in einer INI-Datei..... | 24 |
| 2.1.2. Konfiguration abspeichern..... | 25 |
| 2.2. Firmwareupdate:..... | 26 |
| 3. Technische Daten..... | 27 |
| 4. Lieferumfang STC..... | 28 |
| 5. Impressum..... | 29 |
| 5.1. Firmwareupdates und Programmneuheiten..... | 29 |
| 5.2. Telefonische Unterstützung..... | 29 |
| 5.3. Fehlermeldungen und Verbesserungsvorschläge..... | 29 |
| 5.4. Anschrift..... | 29 |

0. Änderungsdokumentation

| Datum | Abschnitt | Art der Änderung | Bearbeiter | Ausgabe |
|----------|-----------|---|------------|----------|
| 30.09.05 | | STC 1.0, STC-Konfig 1.0.10 | Kulzer | 1.0.0 |
| 08.11.05 | | STC 2.0, STC-Konfig 1.0.10 | Kulzer | 1.0.1 |
| 14.11.05 | | Hinweise hinzugefügt | Kulzer | 1.0.2 |
| 17.03.06 | | Änderungen für neues Konfigurationsprogramm | Kulzer | 1.0.3 |
| 07.08.06 | | Änderung Firmenlogo | Kulzer | 1.0.4 |
| 22.05.07 | | CAN-Buslastmessung und Dongle-Funtion | Kulzer | 1.0.5 |
| 17.07.07 | | Überarbeitete Dongle-Funktion, Automatisches Senden über CAN | Kulzer | 1.0.6 |
| 20.11.07 | | Donglefunktion und Automatisches Senden gleichzeitig möglich, UART für Donglefunktion umschaltbar | Kulzer | 1.0.7 |
| 03.02.10 | | Bidirektionaler STC-Modus | Kulzer | 1.0.8 |
| 19.07.10 | | Baudrate im Donglemodus über RS232 einstellbar | Kulzer | 1.0.9 |
| | | | | |
| 25.01.12 | | Anpassung an überarbeitete Firmware | Kulzer | 01.01.00 |
| | | | | |
| | | | | |
| | | | | |

1. Hardware

1.1. Eigenschaften des STC

Mit diesem Gerät werden serielle Daten (z.B. RS232) auf den CAN umgesetzt. Dadurch ist es möglich RS232-Botschaften mit einem CAN-Tool (z.B. condalo-Datenlogger) zu visualisieren, aufzuzeichnen und zu analysieren.

Der STC verfügt über einen seriellen Anschluss, dessen Daten auf den CAN weitergegeben werden. Auch Daten auf dem CAN können zur RS232 gesendet werden. Desweiteren kann ein CAN-RS232-Dongle für geringe CAN-Buslast durch den STC realisiert werden.

Der STC verfügt zusätzlich 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) | Pin6: TX1- UART |
| Pin2: TX0- UART | Pin7: Digitaler Eingang bzw. RTS (PC-UART) |
| Pin3: RX0- UART | Pin8: Analoger Eingang (Kanal 2) |
| Pin4: RX1- UART(Serial_IN) | Pin9: Versorgung 12 V |
| Pin5: GND | |

(RX0, TX0 für Konfiguration bzw. serielle Ausgabe und
RX1, TX1 Eingang des umzuwandelnden Signals)

Pinbelegung Sub-D Stecker:



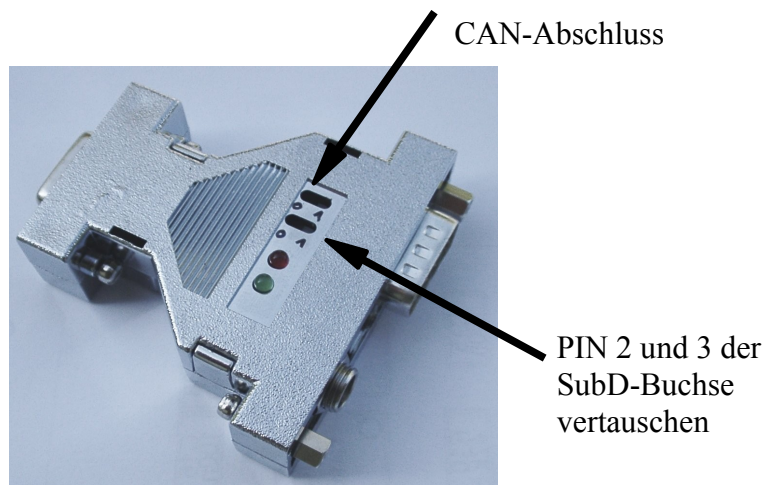
| | |
|-------------|-----------------------|
| Pin1: ---- | Pin6: GND |
| Pin2: CAN-L | Pin7: CAN-H |
| Pin3: GND | Pin8: ---- |
| Pin4: ---- | Pin9: Versorgung 12 V |
| Pin5: GND | |

1.3. Ausführungen:

STC: Pinbelegung wie beschrieben



STC2: 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)

1.4. Funktionen:

Der STC setzt alle ankommenden seriellen Daten in CAN-Botschaften um. Dabei kann die Polarität der ankommenden Signale umgeschaltet werden.

STC-Modus (Seriell nach CAN)

Es werden immer 8 Datenbytes der RS232-Schnittstelle zusammengefasst und in einer CAN-Botschaft ausgegeben. Tritt eine Pause (konfigurierbar) auf der RS232-Leitung auf, werden die letzten noch nicht gesendeten Bytes in einer CAN-Botschaft mit weniger als 8 Datenbytes gesendet.

CTS-Modus (CAN nach Seriell):

Hier werden die Datenbytes einer CAN-Nachricht ohne Steuer- und Auffüllbytes an die RS232 gesendet. Die CAN-ID wird dabei nicht beachtet. Es kann aber ein Filter für CAN-Nachrichten konfiguriert werden.

Dongle-Modus:

In diesem Modus können bidirektional CAN-Nachrichten zur RS232 gesendet und über die RS232 CAN-Nachrichten generiert werden.

Dazu müssen die Daten auf der RS232 eine vorgegebene Struktur aufweisen. Dabei ist zu beachten, dass aufgrund der unterschiedlichen Geschwindigkeiten der einzelnen Schnittstellen nicht sichergestellt werden kann, alle Daten verlustfrei zu übertragen.

Der Analog/Digitalwandler misst in konfigurierbaren Zeitabständen die anliegende Spannung und gibt den Wert in einer CAN-Botschaft mit einstellbarem Identifier aus. 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.

Es werden in allen Modi sowohl standard als auch extended CAN-Frames unterstützt.

1.5. Aufbau der CAN- Botschaften des STC

1.5.1. CAN-Botschaften für die STC-Funktion

CAN-ID: frei Wählbar (eingestellter Identifier)
Botschaft: Daten der RS232-Schnittstelle
Länge: 0..8 Byte

| Parameter | Byte | Bit | Erläuterung |
|-------------|------|-------|-----------------------------------|
| Dateninhalt | 0..8 | 0..63 | Datenbyte der RS232-Schnittstelle |

1.5.2. 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) |

1.5.3. 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.

1.5.4. 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 (FehlerID))
Botschaft: CAN-Reset
Länge: 2 Byte

| Parameter | Byte | Bit | Erläuterung |
|-----------|------|-------|-------------|
| Byte 1 | 0 | 0..7 | 0x00 |
| Byte 2 | 1 | 8..15 | 0x01 |

1.5.5. Ausgabe bei Datenverlust

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(FehlerID))
Botschaft: CAN-Reset
Länge: 2 Byte

| Parameter | Byte | Bit | Erläuterung |
|--|------|--------|-------------|
| Verlustzähler beim Senden auf den CAN H-Byte | 0 | 0..7 | |
| Verlustzähler beim Senden auf den CAN L-Byte | 1 | 8..15 | |
| Verlustzähler beim Empfang auf dem CAN H-Byte | 2 | 16..23 | |
| Verlustzähler beim Empfang auf dem CAN L-Byte | 3 | 24..31 | |
| Verlustzähler beim Empfang von RS232-Bytes H-Byte | 4 | 32..39 | |
| Verlustzähler beim Empfang von RS232-Bytes L-Byte | 5 | 40..47 | |
| Verlustzähler beim Empfang von RS232-Botschaften (Dongle) H-Byte | 6 | 48..55 | |
| Verlustzähler beim Empfang von RS232-Botschaften (Dongle) H-Byte | 7 | 56..63 | |

1.5.6. CAN-Dongle-Funktion

CAN-Baudrate über RS232 ändern (HEX):

Byte1: „b“ (0x62) für Baudrate ändern.
Byte2-5: neue Baudrate HB...LB

Beispiel: 500 kBit/s = 500 000 = 0x07 A1 20
=> 0x62 0x00 0x07 0xA1 0x20

1 MBit/s = 1 000 000 = 0x0F 42 40
=> 0x62 0x00 0x0F 0x42 0x40

nach dem letzten Datenbyte wird automatisch die CAN-Baudrate eingestellt.

Die hier eingestellte Baudrate wird nicht abgespeichert. Nach einem Reset des Gerätes muss dieses Kommando erneut gesendet werden

CAN-Botschaft über RS232 senden (HEX-Modus):

Byte1: „s“ (0x73) bzw. „e“ (0x65) für standard oder extended CAN-Frame
Byte2-5: CAN-ID (HighByte LowByte) immer 4 Byte lang!!!!
Byte 6: Länge (Anzahl der Datenbytes)
Byte 7-x: Datenbytes

Beispiel: ID:0x150 Len:3 Daten: 0x01 0x12 0x23
=> 0x73 0x00 0x00 0x01 0x50 0x03 0x01 0x12 0x23

nach dem letzten Datenbyte wird automatisch die CAN-Nachricht generiert.

CAN-Botschaft über RS232 senden (ASCII-Modus):

für jedes Byte der CAN-Nachricht, werden 2 Zeichen benötigt!!!

Zeichen 1: „s“ bzw. „e“ für standard oder extended CAN-Frame
Zeichen 2-9: CAN-ID (HighByte LowByte) immer 8 Zeichen lang!!!!
Zeichen 10-11: Länge
Byte 7-x: Datenbytes

Beispiel: ID:0x150 Len:3 Daten: 0x01 0x12 0x23
=> „s0000015003011223“

nach dem letzten Zeichen wird automatisch die CAN-Nachricht generiert.

Empfangene CAN-Botschaften über RS232 ausgeben (ASCII-Terminal):

Bsp: STD: 150 3 01 12 23
 EXT: 00012345 3 01 12 23

Empfangene CAN-Botschaften über RS232 ausgeben (HEX-Terminal):

Bsp: 0x0A 0x65 0x00 0x00 0x12 0x23 0x04 0x01 0x02 0x03 0x04

Byte 1: Anzahl der nach diesem Byte folgenden Bytes (für dieses CAN-Telegramm)
Byte 2: „s“ bzw. „e“ (0x73 bzw. 0x65) für standard bzw. extended Frame
Byte 3-6: Identifier der CAN-Nachricht (immer 4 Byte lang)
Byte 7: Länge der CAN-Nachricht
Byte 8-x: Datenbytes

Aufgrund der niedrigeren Datenrate der RS232-Schnittstelle kann es zu Telegrammverlusten kommen. In diesem Fall wird eine Botschaft über die Serielle Schnittstelle ausgegeben.

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.

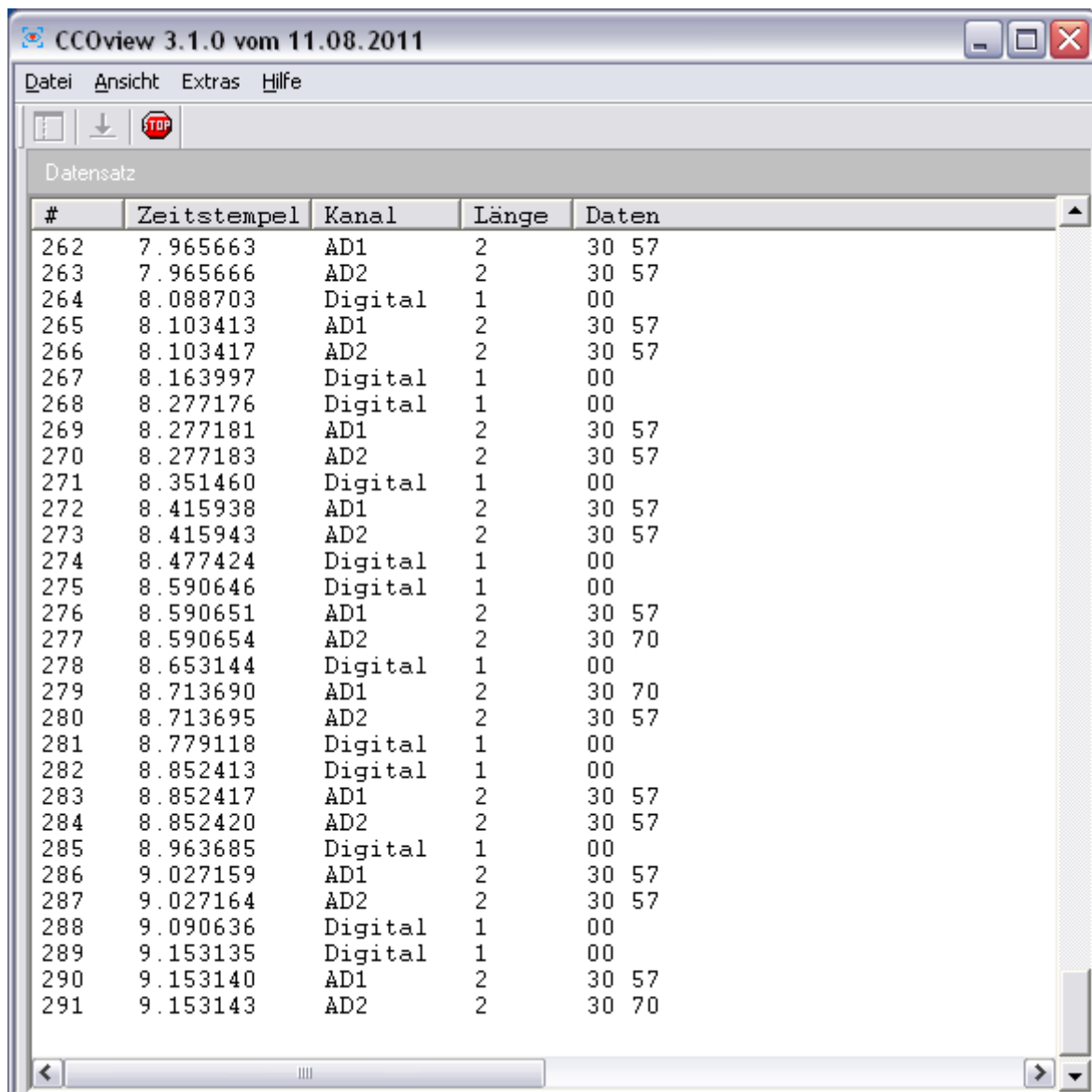
Ist die Datenlänge kleiner als 8 Byte so sind nur die Bytes bis zur Datenlänge gültig.

1.6. 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.



The screenshot shows the CCOview 3.1.0 software window. The title bar reads "CCOview 3.1.0 vom 11.08.2011". The menu bar includes "Datei", "Ansicht", "Extras", and "Hilfe". Below the menu is a toolbar with icons for file operations and a red "STOP" button. The main area is titled "Datensatz" and contains a table with the following columns: "#", "Zeitstempel", "Kanal", "Länge", and "Daten". The table lists 30 data points (rows 262 to 291) with timestamps, channel identifiers (AD1, AD2, Digital), lengths (1 or 2), and data values (hexadecimal pairs like 30 57, 00, 30 70).

| # | Zeitstempel | Kanal | Länge | Daten |
|-----|-------------|---------|-------|-------|
| 262 | 7.965663 | AD1 | 2 | 30 57 |
| 263 | 7.965666 | AD2 | 2 | 30 57 |
| 264 | 8.088703 | Digital | 1 | 00 |
| 265 | 8.103413 | AD1 | 2 | 30 57 |
| 266 | 8.103417 | AD2 | 2 | 30 57 |
| 267 | 8.163997 | Digital | 1 | 00 |
| 268 | 8.277176 | Digital | 1 | 00 |
| 269 | 8.277181 | AD1 | 2 | 30 57 |
| 270 | 8.277183 | AD2 | 2 | 30 57 |
| 271 | 8.351460 | Digital | 1 | 00 |
| 272 | 8.415938 | AD1 | 2 | 30 57 |
| 273 | 8.415943 | AD2 | 2 | 30 57 |
| 274 | 8.477424 | Digital | 1 | 00 |
| 275 | 8.590646 | Digital | 1 | 00 |
| 276 | 8.590651 | AD1 | 2 | 30 57 |
| 277 | 8.590654 | AD2 | 2 | 30 70 |
| 278 | 8.653144 | Digital | 1 | 00 |
| 279 | 8.713690 | AD1 | 2 | 30 70 |
| 280 | 8.713695 | AD2 | 2 | 30 57 |
| 281 | 8.779118 | Digital | 1 | 00 |
| 282 | 8.852413 | Digital | 1 | 00 |
| 283 | 8.852417 | AD1 | 2 | 30 57 |
| 284 | 8.852420 | AD2 | 2 | 30 57 |
| 285 | 8.963685 | Digital | 1 | 00 |
| 286 | 9.027159 | AD1 | 2 | 30 57 |
| 287 | 9.027164 | AD2 | 2 | 30 57 |
| 288 | 9.090636 | Digital | 1 | 00 |
| 289 | 9.153135 | Digital | 1 | 00 |
| 290 | 9.153140 | AD1 | 2 | 30 57 |
| 291 | 9.153143 | AD2 | 2 | 30 70 |

1.7. Leuchtanzeige

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

Die rote Leuchtdiode ist im Normalbetrieb ununterbrochen an.

Ist für eine einstellbare Zeit kein RS232-Byte empfangen worden, 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 RS232-Botschaft, bei der CAN-Ausgabe, bei einer digitalen oder analogen Messung auf.

1.8. Sleepmode

Der STC verfügt über einen Sleepmode.

Stellt der STC für eine einstellbare Zeit kein RS232-Signal fest, so wechselt er in den Standby-Mode. Durch einen Flankenwechsel auf der RS232-Leitung wacht der STC auf und arbeitet mit der zuvor verwendeten Konfiguration. Die Stromaufnahme in diesem Modus beträgt ca. 80 – 100µA.

Es kann konfiguriert werden, ob der STC nur über RS232 oder auch über CAN-aufgeweckt werden darf.

Während aktiver RS232-Kommunikation und solange die Konfigurationssoftware (CCOview) in Betrieb ist, wechselt der STC **nicht** in den Sleepmode.

1.9. Hinweise:

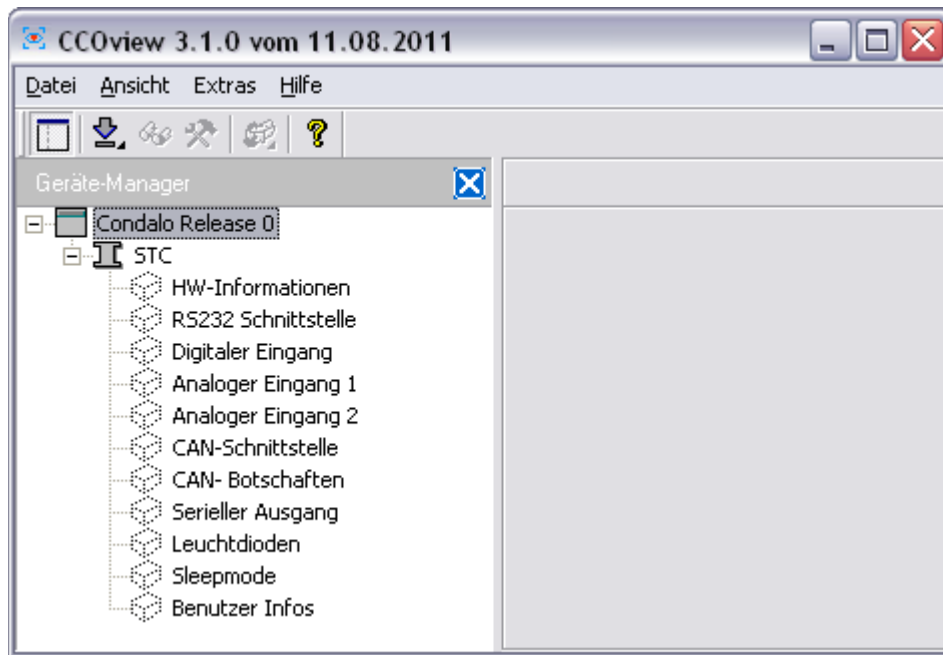
- Der STC verfügt über **keinen** eigenen Busabschluss. Es ist deshalb darauf zu achten, dass der Mess-CAN richtig abgeschlossen ist.
- Der CAN-Empfänger muss das Acknowledge für die CAN-Botschaft senden (sofern dies kein anderer Teilnehmer dieses Busses übernimmt), damit der STC seine Botschaften absetzen kann.
- Gehen Daten verloren, wird dies durch eine gesonderte CAN-Botschaft (konfigurierte Fehler ID) mitgeteilt.
- Wird für die Verbindung des PC mit dem STC kein standardisiertes serielles Kabel verwendet, so ist darauf zu achten, dass auf jeden Fall
 - 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.
- 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 STC verändert, die Firmware upgedatet, und die laufende Kommunikation auf dem Bildschirm mitverfolgt werden.

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



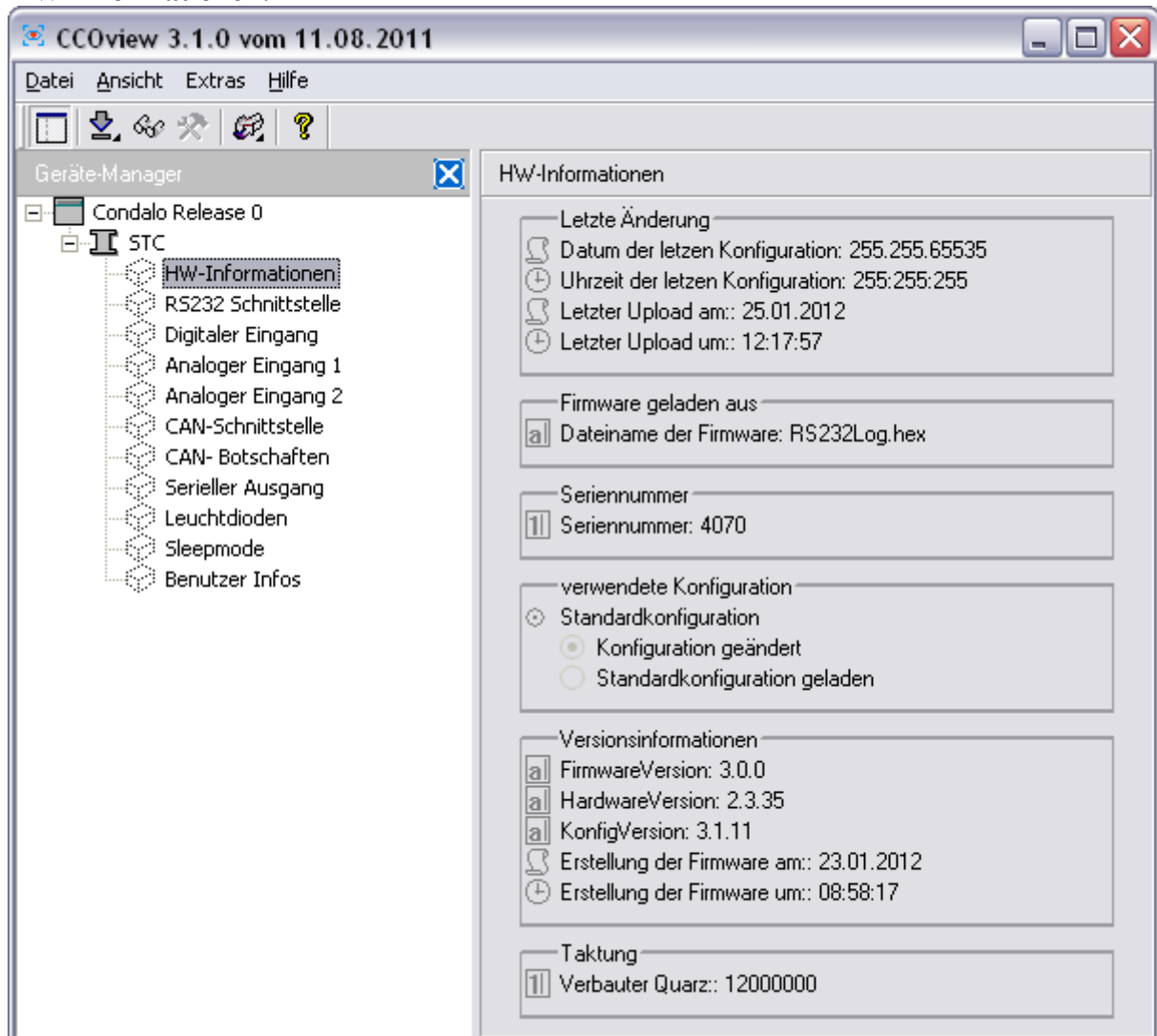
Alle momentan eingestellten Konfigurationen werden ausgelesen und angezeigt.

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

An diese Software können gleichzeitig mehrere Geräte angeschlossen werden. Dazu werden die verfügbaren COM-Ports immer wieder abgefragt.

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

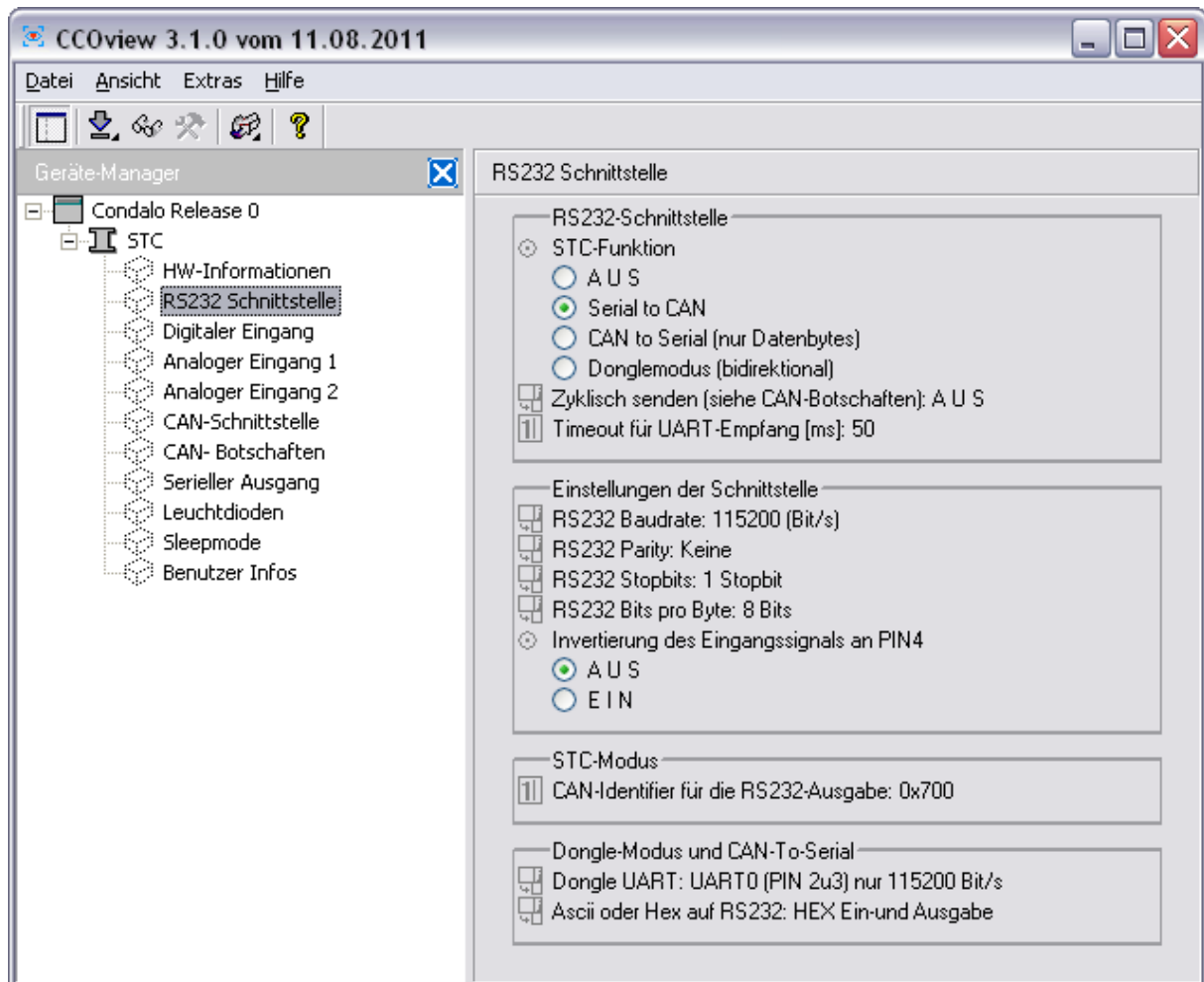
HW-Informationen:



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

Mögliche Einstellungen sind:

RS232 Eingang:



Einstellungen:

Funktion:

AUS,

STC-Mode: RS232-Daten auf den CAN senden.

CTS-Mode: CAN-Daten auf die RS232 senden.

DongleMode: Bidirektional Daten austauschen.

Zyklisches Senden vordefinierter Nachrichten (siehe CAN-Botschaften)

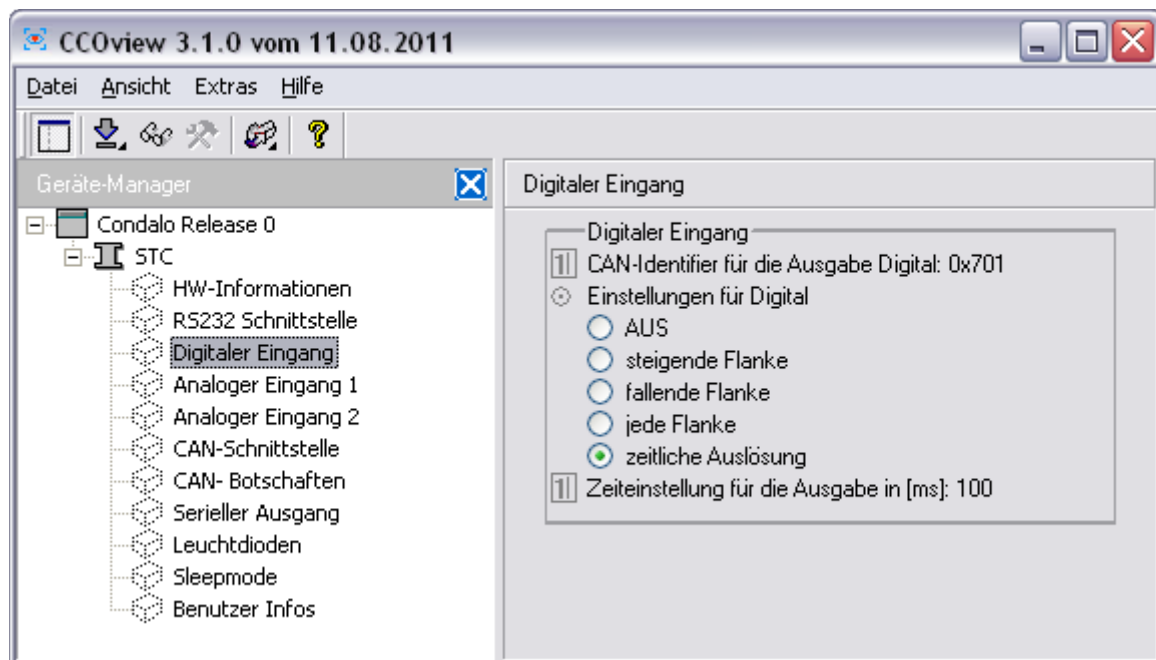
Timeout (Zeit, nach der das Ende einer RS232-Botschaft erkannt wird -> Senden der CAN-Botschaft auch wenn z.B. noch keine 8 Datenbytes gesammelt wurden)

Einstellungen für die Schnittstelle: RS232-Baudrate, RS232-Parität, RS232-Stopbits, RS232 Bits pro Byte, Signalinvertierung.

Identifizier für die CAN-Ausgabe

Dongle und CTS-Mode: Uart für die Datenausgabe und Datenformat (Ascii oder Hex).

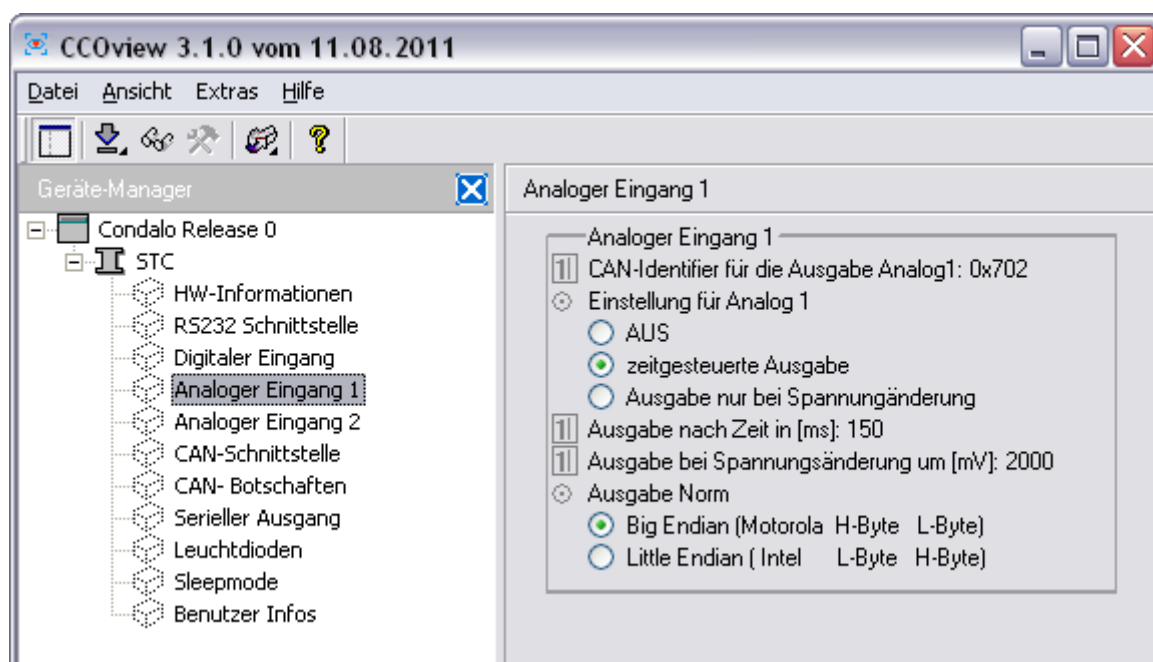
Digitaler Eingang:



Einstellungen:

Identifizier für die CAN-Ausgabe,
 Auslöseereignis für die Ausgabe,
 Zeit in ms für die zyklische Messung

Analoger Eingang:



Einstellungen:

Identifizier für die CAN-Ausgabe des Analogwertes

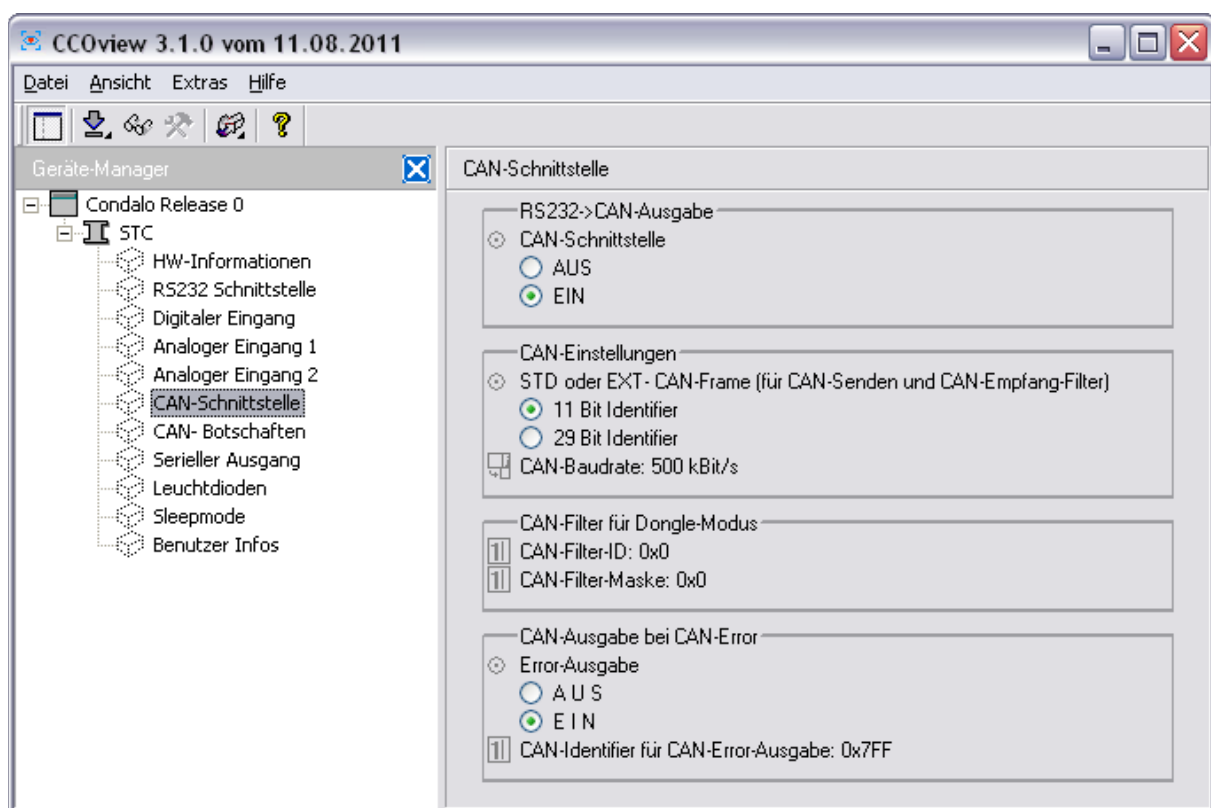
Auslöseereignis für die Ausgabe

Zeit in ms für die zyklische Messung

Spannung in mV, bei deren Änderung zur vorherigen Ausgabe eine CAN-Ausgabe erfolgen soll.

Ausgabe-Norm für den Messwert.

CAN Ausgang:



Einstellungen:

CAN-Schnittstelle EIN oder AUS.

Standard oder extended Identifier.

CAN-Baudrate

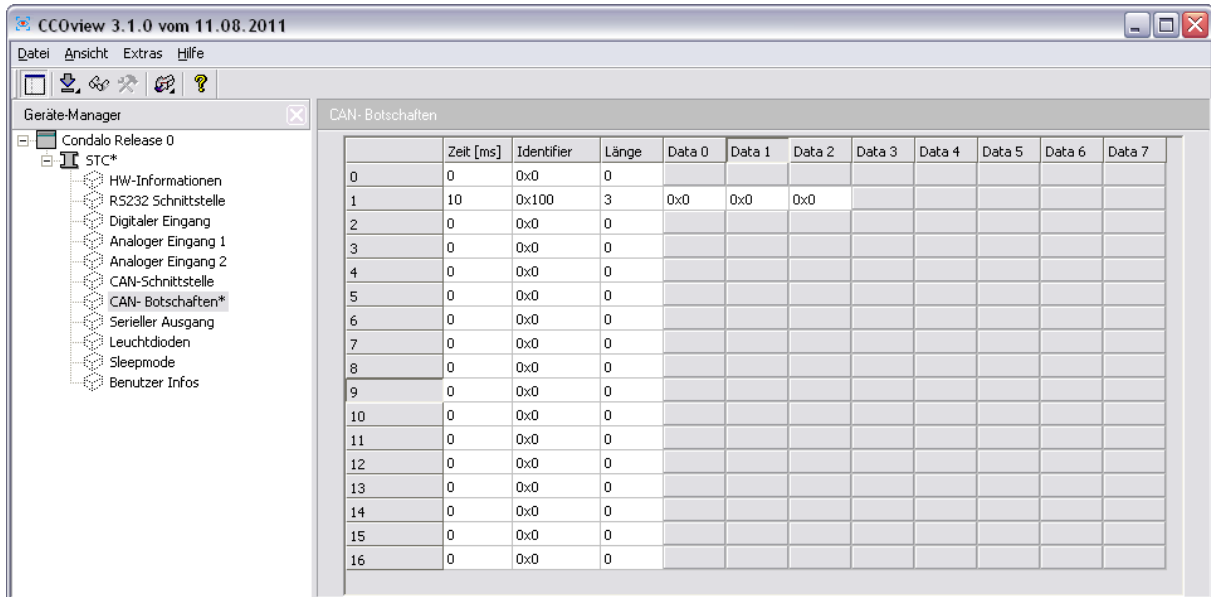
CAN-Empfangsfilter für Dongle und CTS-Funktion:

FilterID für die Dongle-Funktion (Code)

FilterID für die Dongle-Funktion (Maske)

Identifizier für die CAN-Error-Ausgabe

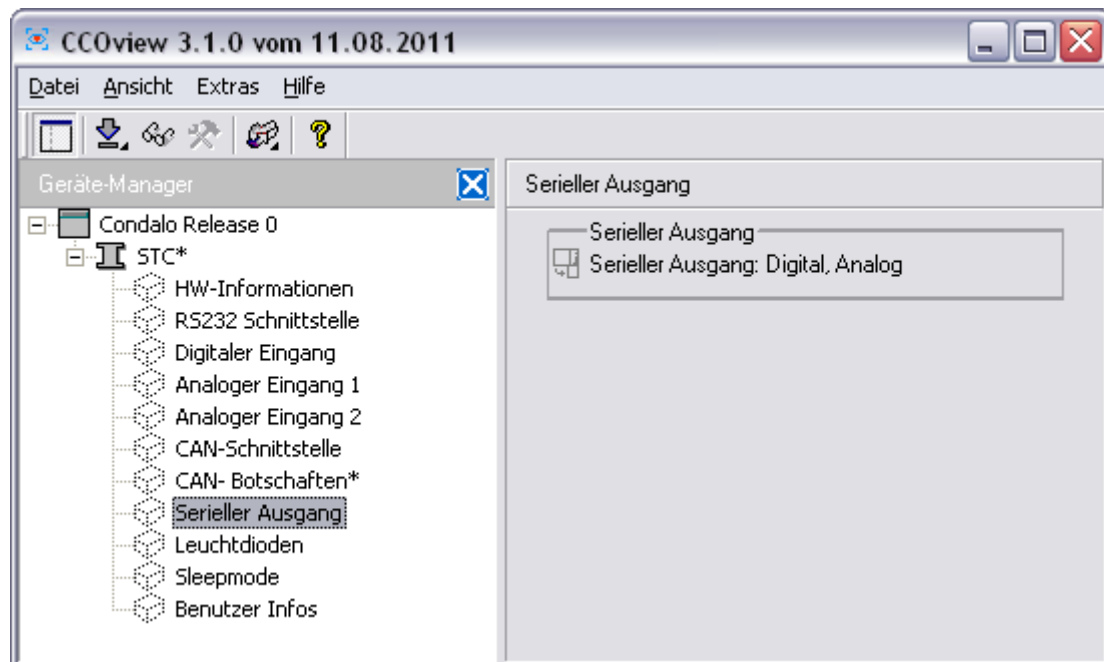
CAN-Botschaften



| | Zeit [ms] | Identifizier | Länge | Data 0 | Data 1 | Data 2 | Data 3 | Data 4 | Data 5 | Data 6 | Data 7 |
|----|-----------|--------------|-------|--------|--------|--------|--------|--------|--------|--------|--------|
| 0 | 0 | 0x0 | 0 | | | | | | | | |
| 1 | 10 | 0x100 | 3 | 0x0 | 0x0 | 0x0 | | | | | |
| 2 | 0 | 0x0 | 0 | | | | | | | | |
| 3 | 0 | 0x0 | 0 | | | | | | | | |
| 4 | 0 | 0x0 | 0 | | | | | | | | |
| 5 | 0 | 0x0 | 0 | | | | | | | | |
| 6 | 0 | 0x0 | 0 | | | | | | | | |
| 7 | 0 | 0x0 | 0 | | | | | | | | |
| 8 | 0 | 0x0 | 0 | | | | | | | | |
| 9 | 0 | 0x0 | 0 | | | | | | | | |
| 10 | 0 | 0x0 | 0 | | | | | | | | |
| 11 | 0 | 0x0 | 0 | | | | | | | | |
| 12 | 0 | 0x0 | 0 | | | | | | | | |
| 13 | 0 | 0x0 | 0 | | | | | | | | |
| 14 | 0 | 0x0 | 0 | | | | | | | | |
| 15 | 0 | 0x0 | 0 | | | | | | | | |
| 16 | 0 | 0x0 | 0 | | | | | | | | |

Definition der zyklisch zu sendenden CAN-Nachrichten.

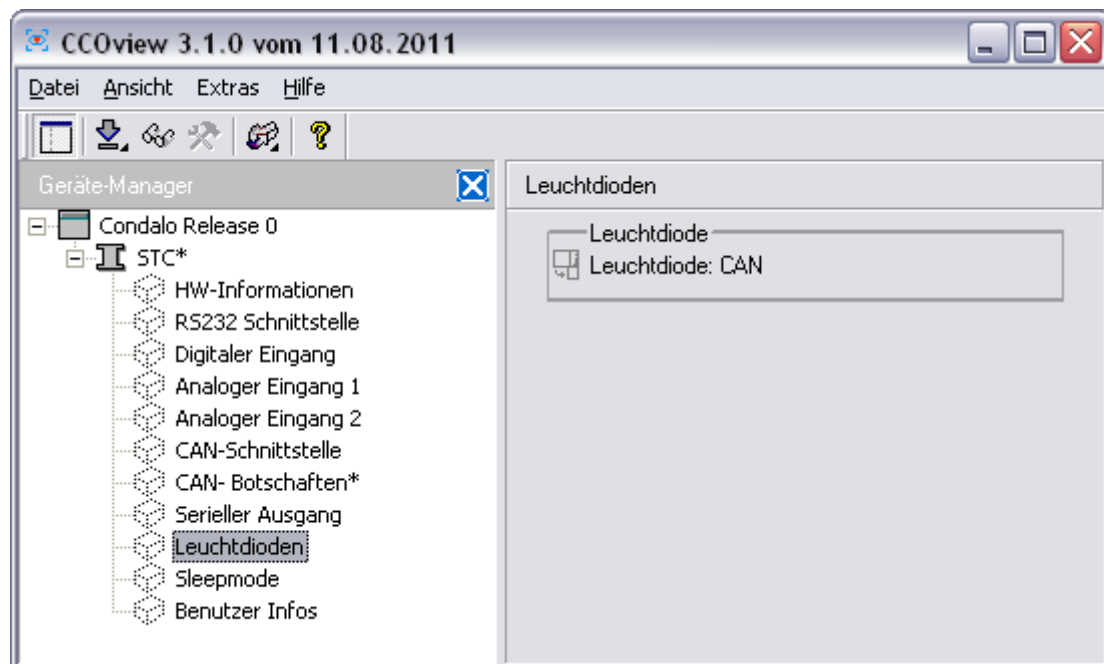
Serieller Ausgang:



Einstellungen:

Im seriellen Trace können Digitale oder Analoge Werte dargestellt werden.

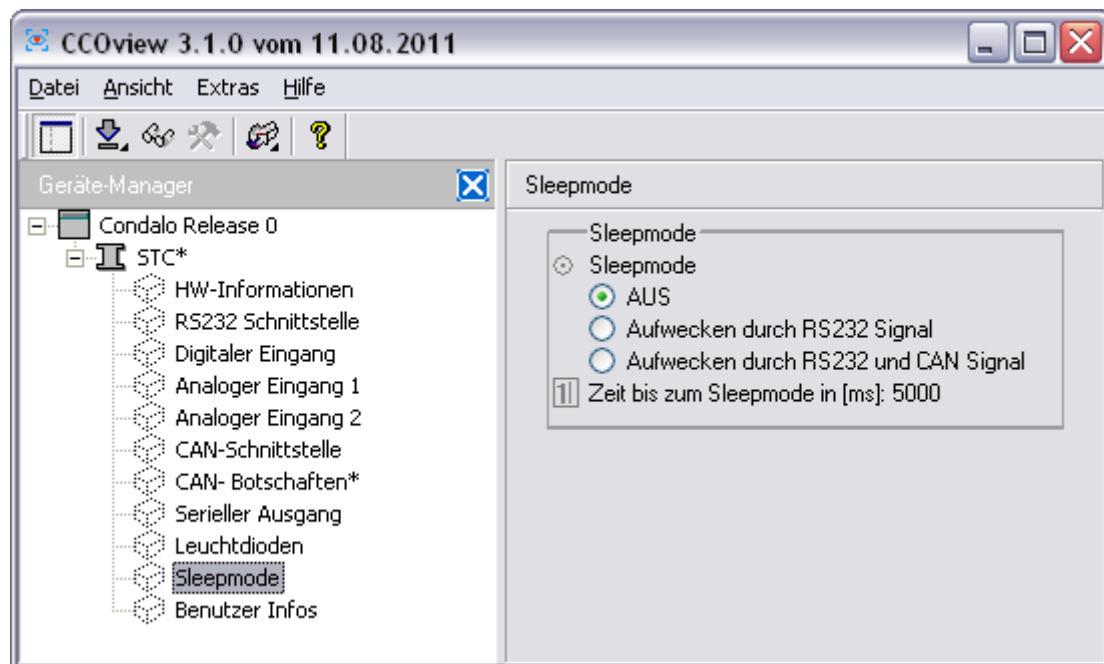
Leuchtdioden:



Einstellungen:

Ereignis, bei dem die Grüne LED aufleuchten soll.

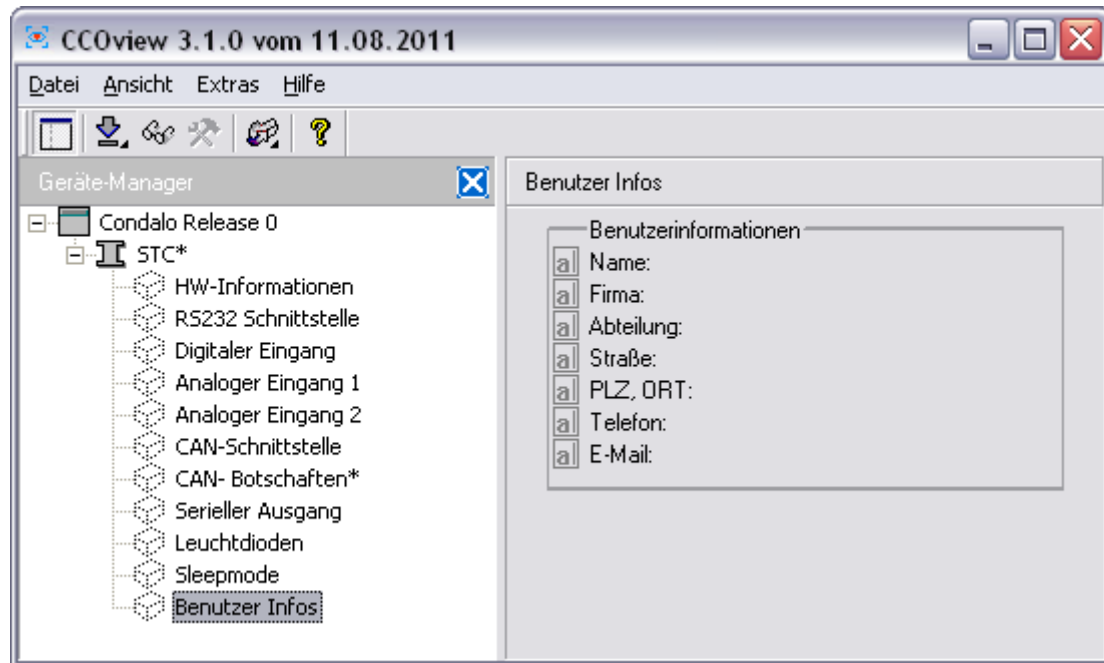
Sleepmode:



Aufwecken durch RS232 oder CAN/RS232

Dauer der Busruhe bis zum Sleepmode

Benutzer Infos:



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

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

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

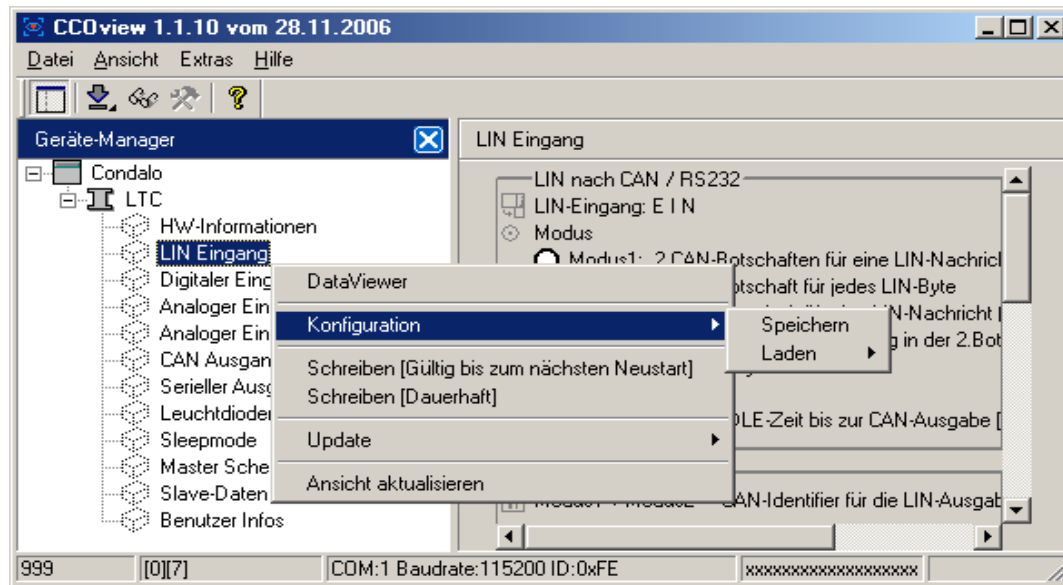
Es ist möglich, symbolische Namen für CAN-Identifizier mit Hilfe einer INI-Datei zu vergeben. Befindet sich die INI-Datei im selben Ordner wie „CCOview.exe“, so können bei den CAN-Identifiern entweder die vorgelegten symbolischen Namen, sowie auch freie Eingaben verwendet werden.

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] |

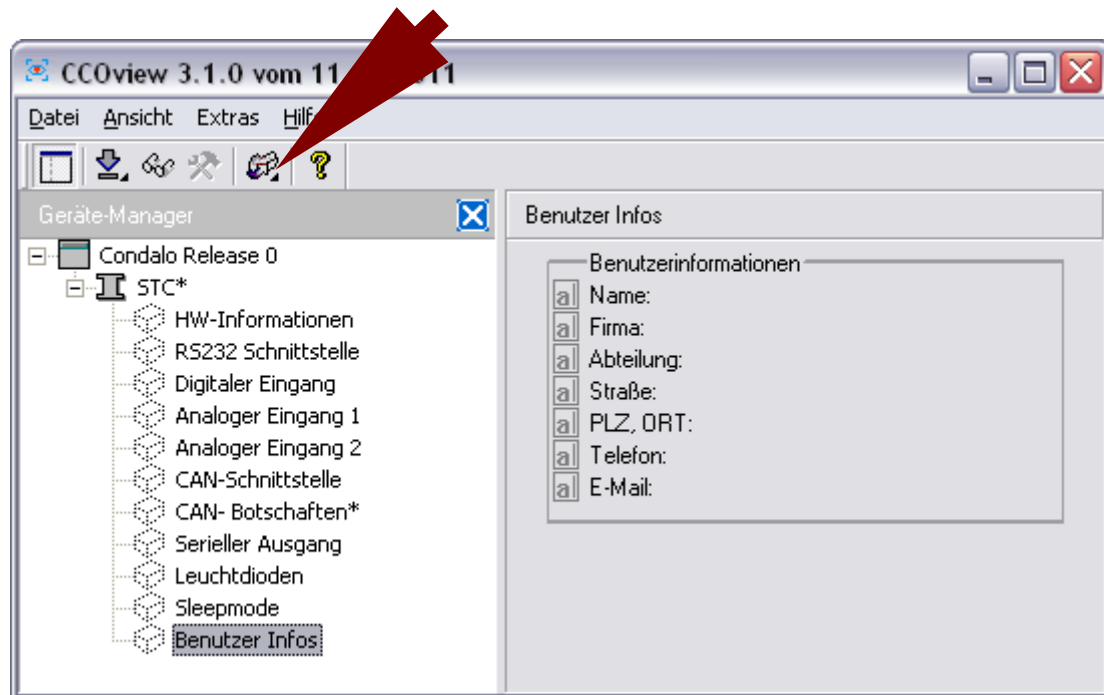
2.1.2. Konfiguration abspeichern



Die Verschiedenen Einstellungen des STC 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 STC geladen werden.

2.2. Firmwareupdate:

Die Firmware („*.hex“ bzw. „*.imt“ Datei) muss in den selben Order kopiert werden, in dem sich auch die Programmdatei des Konfigurationstools befindet.



Durch den "Update-Button" kann eine neue Firmware auf den STC geladen werden. Die gewünschte Firmware wird im Untermenü dieses Befehls ausgewählt.

3. Technische Daten

Funktionen:

RS232 nach CAN 2.0A, 2.0B Umsetzer (CAN-Highspeed-Transceiver)

Messung von zwei Spannungen 0 .. 25V

Überwachung eines digitalen Eingangs

Betriebsanzeige über zwei LEDs

Anschlüsse:

9pol Sub-D Buchse: Spannungsversorgung, RS232 (TX und RX),
2x Analog, 1x Digital, RS232 für Konfiguration

9pol Sub-D Stecker: Spannungsversorgung, CAN

Sonstiges:

Betriebsspannung: 12 V DC (7V DC - 30V DC)
Stromaufnahme max. 45mA

Temperaturbereich: -40°C bis +85°C

Abmessungen: 63mm x 17mm x 33mm

Gewicht: 22 Gramm

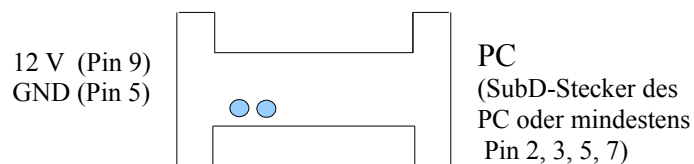
Material: Kunststoffgehäuse mit Metallüberzug

Analoge Eingänge: 0 - 25 V (9:1 vorgeteilt)
10 Bit Auflösung
 ± 2 LSB absolute Genauigkeit
Messzyklus 1ms bis 65535ms

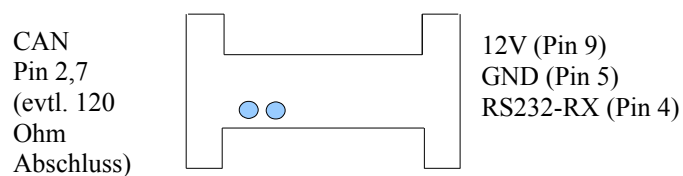
Digitaler Eingang: max. 50 V ohne zeitliche Beschränkung

Anschlusschema:

Konfiguration:



LIN To CAN-Umsetzung:



4. Lieferumfang STC

- 1 STC (RS232 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. Telefonische Unterstützung

| | |
|---------------|-------------------|
| Markus Kulzer | 08450 - 9264 - 43 |
| Zentrale | 08450 - 9264 - 0 |

5.3. Fehlermeldungen und Verbesserungsvorschläge

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

5.4. 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