

CANbox[®]

Dual CAN to LAN/WLAN Converter

Die CANbox®

1	Allgemeines	6
1.1	BESONDERE EIGENSCHAFTEN	6
1.2	EINGESETZTE MAX-MODULE	6
1.3	LAGEPLAN DER STECKVERBINDER	7
1.4	LAN / WLAN – EIGENSCHAFTEN	7
1.4.1	WLAN	7
1.4.2	Übertragungsraten	8
	Stecker der CANbox®	8
2.1	ÜBERSICHT	8
2.2	BELEGUNG DER EXTERNEN STECKVERBINDER	9
2.2.1	Steckverbinder ST 1, Spannungsversorgung	9
2.2.2	Host-Schnittstellen (Steckverbinder ST 2 und ST 3)	10
2.2.3	D-SUB-9 Stecker für CAN 1 und CAN 2	10
2.2.4	SMA-Stecker für Antenne	11
3	Inbetriebnahme	12
3.1	CANBOX® - SETUP	12
3.2	CANBOX® - CONFIGTOOL UNTER WIN32 BETRIEBSSYSTEMEN	13
3.2.1	Bedienung	13
3.2.2	CANbox® Konfiguration Menü	14
3.2.3	CANbox® Konfigurationsdialog	15
3.2.4	CANbox® Eigenschaften Dialog	16
3.3	ERSTE VERBINDUNG MIT EINER CANBOX®	17
3.3.1	Verbindung über LAN	17
3.3.2	Verbindung über serielle Schnittstelle	17
3.4	POCKET PC 2003	18
3.5	EINRICHTUNGSBEISPIELE	19
3.5.1	Einrichtungsbeispiel unter Windows XP	19
3.5.2	Einrichtungsbeispiel 1 unter POCKET PC 2003	20
3.5.3	Einrichtungsbeispiel 2 unter POCKET PC 2003	21
3.5.4	Einrichtungsbeispiel unter POCKET PC 2003 SE (Second Edition)	22
3.6	CANBOX® ALS ROUTER	23
3.6.1	Konfigurationsbeispiele	25
3.7	ERWEITERTER ROUTER (FORWARDER)	27
3.7.1	Konfigurationsbeispiele	27
3.8	KONFIGURATIONSTOOL ACCEPTANCEFILTERCALCULATOR	29
3.9	AUTOMATISCHE NEUARBITRIERUNG DES CAN-BUS IM ROUTERMODUS	30

4 CANbox® – Bibliothek	31
4.1. STRUKTUREN UND DEFINITIONEN.....	31
4.1.1 CANBOX_VERSION_INFO.....	31
4.1.2 CANBOX_TIME_STAMP.....	31
4.1.3 CANBOX_SCAN_INFO.....	31
4.1.4 CANBOX_DEVICE_INFO.....	32
4.1.5 CANBOX_RESET_EX.....	32
4.1.6 CANBOX_INTERFACE.....	32
4.1.7 CANBOX_INTERFACE_EX.....	34
4.1.8 CANBOX_INTERFACE_STATE.....	35
4.1.9 CANBOX_INTERFACE_READ.....	36
4.1.10 CANBOX_INTERFACE_WRITE.....	36
4.1.11 CANBOX_IDENTIFIER.....	36
4.1.12 CANBOX_ACCEPTANCE_FILTER.....	37
4.1.13 CANBOX_ROUTER.....	38
4.1.14 CANBOX_ROUTER_EX.....	39
4.1.15 CANBOX_LAN_CONFIG.....	40
4.1.16 CANBOX_WLAN_CONFIG.....	41
4.1.17 CANBOX_INTERFACE_INFO.....	42
4.1.18 CANBOX_IDENTIFIER_DATA.....	43
4.1.19 CANBOX_UNIVERSAL_SENDER_DATA.....	43
4.1.20 CANBOX_16.....	43
4.1.21 CANBOX_32.....	44
4.2 FEHLERCODES.....	45
4.2.1 <i>Allgemeine Fehlermeldungen der CANbox®</i>	45
4.2.2 <i>Fehlermeldungen bezüglich der Initialisierung</i>	45
4.2.3 <i>Fehlermeldungen bezüglich Freigaben</i>	45
4.2.4 <i>Fehlermeldungen bezüglich CAN-Schnittstellen</i>	45
4.2.5 <i>Fehlermeldungen bezüglich CAN – Identifiern</i>	46
4.2.6 <i>Fehlermeldungen bezüglich Akzeptanz-Filtern</i>	46
4.2.7 <i>Fehlermeldungen bezüglich Universal-Sendern</i>	46
4.2.8 <i>Allgemeine Fehlermeldungen der Bibliothek</i>	46
4.2.9 <i>Fehlermeldungen der Bibliothek bezüglich der Verbindung zur CANbox®</i> ..	46
4.3 ALLGEMEINE FUNKTIONEN.....	47
4.3.1 <i>canbox_init_lib</i>	47
4.3.2 <i>canbox_get_driver_version</i>	47
4.3.3 <i>canbox_get_error_message</i>	47
4.3.4 <i>canbox_exit_lib</i>	47
4.4 DEVICE- FUNKTIONEN.....	48
4.4.1 <i>canbox_scan_devices</i>	48
4.4.2 <i>canbox_get_device_parameter</i>	48
4.4.3 <i>canbox_open_device</i>	48
4.4.4 <i>canbox_reset_device</i>	49
4.4.5 <i>canbox_reset_device_ex</i>	49
4.4.6 <i>canbox_get_device_info</i>	49
4.4.7 <i>canbox_reset_time_stamp</i>	49
4.4.8 <i>canbox_set_time_stamp</i>	50
4.4.9 <i>canbox_set_time_stamp_by_ref</i>	50

4.4.10	<i>canbox_close_device</i>	50
4.5	SCHNITTSTELLEN- FUNKTIONEN	50
4.5.1	<i>canbox_get_interface_info</i>	50
4.5.2	<i>canbox_open_interface</i>	51
4.5.3	<i>canbox_open_interface_by_ref</i>	51
4.5.4	<i>canbox_open_interface_ex</i>	51
4.5.5	<i>canbox_open_interface_ex_by_ref</i>	52
4.5.6	<i>canbox_get_interface_state</i>	52
4.5.7	<i>canbox_get_last_interface_time</i>	52
4.5.8	<i>canbox_clear_interface</i>	52
4.5.9	<i>canbox_start_interface</i>	53
4.5.10	<i>canbox_read_interface</i>	53
4.5.11	<i>canbox_read_interface_ex</i>	53
4.5.12	<i>canbox_write_interface</i>	54
4.5.13	<i>canbox_write_interface_ex</i>	54
4.5.14	<i>canbox_stop_interface</i>	54
4.5.15	<i>canbox_close_interface</i>	55
4.6	IDENTIFIER FUNKTIONEN	56
4.6.1	<i>canbox_open_identifizier</i>	56
4.6.2	<i>canbox_open_identifizier_by_ref</i>	56
4.6.3	<i>canbox_read_identifizier</i>	56
4.6.4	<i>canbox_write_identifizier</i>	57
4.6.5	<i>canbox_close_identifizier</i>	57
4.7	AKZEPTANZ-FILTER-FUNKTIONEN	57
4.7.1	<i>canbox_open_acceptance_filter</i>	57
4.7.2	<i>canbox_open_acceptance_filter_by_ref</i>	58
4.7.3	<i>canbox_read_acceptance_filter</i>	58
4.7.4	<i>canbox_close_acceptance_filter</i>	58
4.8	UNIVERSAL-SENDER-FUNKTIONEN.....	59
4.8.1	<i>canbox_open_universal_sender</i>	59
4.8.2	<i>canbox_write_universal_sender</i>	59
4.8.3	<i>canbox_write_universal_sender_ex</i>	59
4.8.4	<i>canbox_close_universal_sender</i>	60
4.9	ROUTER FUNKTIONEN.....	60
4.9.1	<i>canbox_read_router_config</i>	60
4.9.2	<i>canbox_read_router_config_ex</i>	60
4.9.3	<i>canbox_write_router_config</i>	61
4.9.4	<i>canbox_write_router_config_ex</i>	61
4.10	ERWEITERTE ROUTER FUNKTION (FORWARDER)	61
4.10.1	<i>canbox_read_forwarder_config</i>	61
4.10.2	<i>canbox_write_forwarder_config</i>	62
4.11	HILFSFUNKTIONEN FÜR VISUAL BASIC	62
4.11.1	<i>canbox_convert_to_CANbox16</i>	62
4.11.2	<i>canbox_convert_to_CANbox16_by_ref</i>	62
4.11.3	<i>canbox_convert_from_CANbox16</i>	63
4.11.4	<i>canbox_convert_to_CANbox32</i>	63
4.11.5	<i>canbox_convert_to_CANbox32_by_ref</i>	63

4.11.6	<i>canbox_convert_from_CANbox32</i>	63
5	CANbox® – Socket Schnittstelle	64
5.1.	STRUKTUREN UND DEFINITIONEN	64
5.1.1	<i>CANBOX_SOCKET_IN</i>	64
5.1.2	<i>CANBOX_SOCKET_OUT</i>	64
5.2	FUNKTIONEN	65
5.2.1	<i>canbox_init (Index 4)</i>	65
5.2.2	<i>canbox_reset (Index 5)</i>	65
5.2.3	<i>canbox_info (Index 6)</i>	65
5.2.4	<i>canbox_set_time_stamp (Index 7)</i>	65
5.2.5	<i>canbox_exit (Index 12)</i>	66
5.2.6	<i>canbox_open_interface (Index 15)</i>	66
5.2.7	<i>canbox_open_interface_ex (Index 16)</i>	66
5.2.8	<i>canbox_get_interface_state (Index 17)</i>	66
5.2.9	<i>canbox_start_interface (Index 18)</i>	67
5.2.10	<i>canbox_clear_interface (Index 19)</i>	67
5.2.11	<i>canbox_read_interface (Index 20)</i>	67
5.2.12	<i>canbox_write_interface (Index 21)</i>	67
5.2.13	<i>canbox_stop_interface (Index 22)</i>	68
5.2.14	<i>canbox_close_interface (Index 23)</i>	68
5.2.15	<i>canbox_open_identifier (Index 25)</i>	68
5.2.16	<i>canbox_read_identifier_buffer (Index 26)</i>	68
5.2.17	<i>canbox_read_identifier_actual (Index 27)</i>	69
5.2.18	<i>canbox_write_identifier (Index 28)</i>	69
5.2.19	<i>canbox_close_identifier (Index 29)</i>	69
5.2.20	<i>canbox_open_filter (Index 30)</i>	69
5.2.21	<i>canbox_read_filter_buffer (Index 31)</i>	70
5.2.22	<i>canbox_read_filter_actual (Index 27)</i>	70
5.2.23	<i>canbox_close_filter (Index 32)</i>	70
5.2.24	<i>canbox_open_sender (Index 13)</i>	71
5.2.25	<i>canbox_write_sender (Index 24)</i>	71
5.2.26	<i>canbox_close_sender (Index 14)</i>	71
5.2.27	<i>canbox_read_router_config (Index 34)</i>	71
5.2.28	<i>canbox_write_router_config (Index 35)</i>	72
5.2.29	<i>canbox_read_lan_config (Index 36)</i>	72
5.2.30	<i>canbox_write_lan_config (Index 37)</i>	72
5.2.31	<i>canbox_read_wlan_config (Index 38)</i>	72
5.2.32	<i>canbox_write_wlan_config (Index 39)</i>	73
6	Technische Daten	74

1 Allgemeines

Bei der CANbox[®] handelt es sich um einen Adapter mit zwei High-Speed CAN Schnittstellen bis 1 MBit/s entsprechend den CAN-Spezifikationen 2.0A und 2.0B. Optional stehen auch fehlertolerante Low-Speed Schnittstellen zur Verfügung. Die Verbindung zur CANbox[®] ist dabei über verschiedene Medien möglich. Die CANbox[®] wird mit einem Datenträger ausgeliefert, auf der sich die im Software Teil beschriebenen Komponenten befinden. Die Treiber stehen für Win32- und Pocket PC 2003 - Betriebssysteme zur Verfügung. Weiterhin ermöglicht die Socketschnittstelle eine direkte Kommunikation auf TCP/IP Socket Ebene (ab Firmware Version 3.A).



Es können bis zu 10 CANboxen auf einem PC bzw. PDA gleichzeitig verwendet werden.

Das HF-dichte Metallgehäuse mit den Abmessungen 113 x 83 x 33 mm kann auch auf Hutschiene oder mit Schrauben befestigt werden, wenn ein stationärer Einsatz erforderlich ist. Die Spannungsversorgung der CANbox[®] ist sehr flexibel ausgelegt und kann direkt mit 6..60V (DC) erfolgen, z.B. auch über das Bordnetz (inkl. 42V) im Fahrzeug. Die Spannungsversorgung ist nicht galvanisch getrennt. Eine galvanische Trennung der CAN-Schnittstellen wird durch das eingesetzte X-CAN-2-Modul erreicht.

1.1 Besondere Eigenschaften

- Intelligentes, dezentral und selbständig arbeitendes Peripheriegerät
- Schnittstellenkonverter von CAN auf LAN/WLAN
- WLAN on Board
- Abmessungen 113 x 83 x 33 mm
- Versorgungsspannung: 6,0V...60V (DC)
- Montierbar auf DIN-Schiene oder ebener Fläche (Halter optional erhältlich)

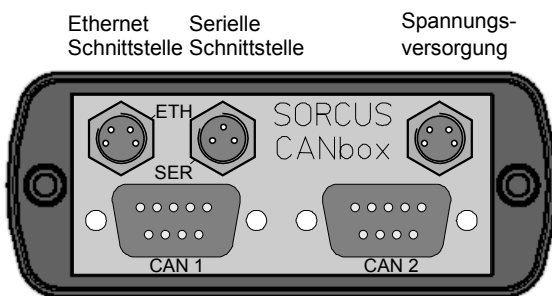
1.2 Eingesetzte MAX-Module

Das Gerät weist 2 Modul Steckplätze auf. Sie sind wie in folgender Tabelle beschrieben bestückt.

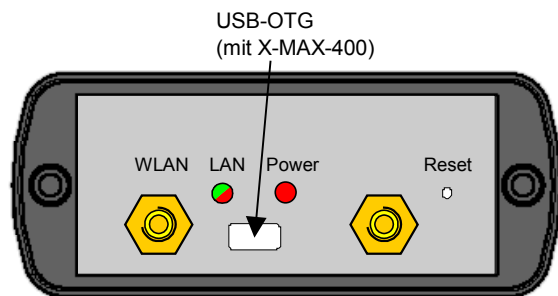
Slot Nr.	Modul	Funktion	Erläuterung
1	X-MAX-E	CPU-Modul mit Ethernet	Dieser Steckplatz ist für ein CPU-Modul vorgesehen.
2	X-CAN-2/H	2 CAN Kanäle mit High Speed Treibern	Optional auch 2 fehlertolerante CAN Kanäle oder gemischt möglich.

1.3 Lageplan der Steckverbinder

Vorderseite



Rückseite



1.4 LAN / WLAN – Eigenschaften

1.4.1 WLAN

Der WLAN – Standard 802.11b stellt je nach Region bis zu 14 Kanäle zur Verfügung. Hierbei ist zu beachten, dass nicht alle Kanäle überlappungsfrei sind. Sollten mehrere WLAN – Netze zum Einsatz kommen empfiehlt sich daher die Verwendung unterschiedlicher Kanäle, wobei immer 4 Kanäle zwischen 2 Kanälen frei bleiben sollten.

Kanal	Mittenfrequenz	Frequenz - Spektrum
1	2412 MHz	2399.5 MHz – 2424.5 MHz
2	2417 MHz	2404.5 MHz – 2429.5 MHz
3	2422 MHz	2409.5 MHz – 2434.5 MHz
4	2427 MHz	2414.5 MHz – 2439.5 MHz
5	2432 MHz	2419.5 MHz – 2444.5 MHz
6	2437 MHz	2424.5 MHz – 2449.5 MHz
7	2442 MHz	2429.5 MHz – 2454.5 MHz
8	2447 MHz	2434.5 MHz – 2459.5 MHz
9	2452 MHz	2439.5 MHz – 2464.5 MHz
10	2457 MHz	2444.5 MHz – 2469.5 MHz
11	2462 MHz	2449.5 MHz – 2474.5 MHz
12	2467 MHz	2454.5 MHz – 2479.5 MHz
13	2472 MHz	2459.5 MHz – 2484.5 MHz
14	2484 MHz	2471.5 MHz – 2496.5 MHz

WLAN – Kanäle und ihre Frequenzen

Die Reichweite eines WLAN – Gerätes beträgt typischerweise 30m innen (maximal 100m) und 100m außen (maximal 300m).

1.4.2 Übertragungsraten

Die nachfolgende Tabelle zeigt die erzielten Übertragungsraten mit der CANbox® in Abhängigkeit von den verschiedenen Übertragungsarten. Die CANbox® befand sich im **Leerlauf**, der Abstand zwischen PC und CANbox® betrug ca. 1m. Diese Angaben sind somit nur als Richtwerte zu sehen.

Übertragungsart	Übertragungsrate
Verbindung mit WLAN-PC über AdHoc	140 kB/s
Verbindung mit WLAN-PC über Accesspoint	90 kB/s
Verbindung mit LAN-PC über Accesspoint	195 kB/s
Verbindung mit LAN-PC über LAN	250 kB/s

2 Stecker der CANbox®

2.1 Übersicht

Folgende Steckverbinder sind am Gehäuse vorhanden:



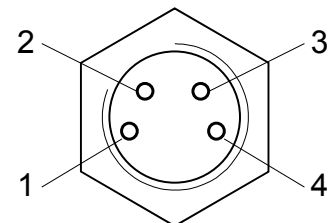
Stecker / Typ Buchse	Funktion
ST1 M8-Einbaustecker 4-Polig	Spannungsversorgung
ST2 M8-Einbaustecker 4-Polig	Ethernet Schnittstelle
ST3 M8-Einbaustecker 3-Polig	Serielle Schnittstelle für Servicezwecke
CAN 1 D-SUB-9 Stecker	CAN Bus 1
CAN 2 D-SUB-9 Stecker	CAN Bus 2
WLAN SMA Buchse	Anschluss für WLAN – Antenne

2.2 Belegung der externen Steckverbinder

Für die Spannungsversorgung und die Host-Schnittstellen werden sogenannte Sensor-Steckverbinder Baureihe M8 eingesetzt. Die Kabel dazu sind mit angespritztem 3- oder 4-poligem Stecker in 2m und 5m Länge bei vielen Distributoren erhältlich. Die Kabel in 2m Länge sind auch bei SORCUS erhältlich. Von SORCUS erhalten Sie auch die fertig konfektionierten Kabel mit D-SUB-9 Buchsen für die serielle Hostankopplung und RJ45-Steckern für den direkten Anschluss des Ethernet an ein Firmennetzwerk oder an einen PC (Cross-Over).

Die Kabel haben folgende Farbzusordnungen:

Farbe	3-Polig	4-Polig
Braun	1	1
Weiß		2
Schwarz	2	3
Blau	3	4

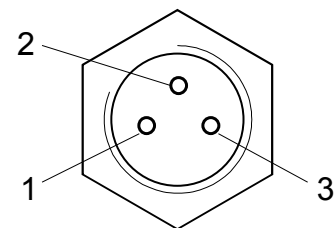


Ansicht der
M8-Stecker
von aussen!

Die Kabel der Firma Hirschmann haben die folgenden Typenbezeichnungen:

- Kabel M8 3-Polig mit Schraubverriegelung: ELKA-KV 3308
- Kabel M8 4-Polig mit Schraubverriegelung: ELKA-KV 4408

Bei Kabeln von anderen Herstellern (Binder, Phoenix etc.) gelten andere Bezeichnungen.



2.2.1 Steckverbinder ST 1, Spannungsversorgung

Am M8-Stecker ST1 muss die Spannungsversorgung der CANbox® angelegt werden. An den Kontakten 3 und 4 wird GND angeschlossen, an den Kontakten 1 und 2 eine Spannungsquelle mit einer Gleichspannung von 6..60V. Bei Verpolung der Spannungsquelle wird eine Sicherung zerstört, die nur vom Service ersetzt werden kann.

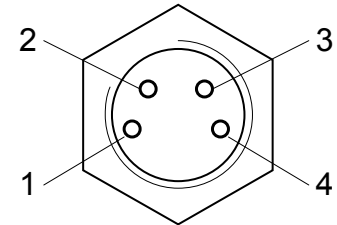
ST1	Signal	Kabelfarbe
1,2	+Vbatt	Braun, Weiß
3,4	- Vbatt (=XGND)	Schwarz, Blau

Durch den integrierten DCDC – Wandler ist die Stromaufnahme bei kleinen Eingangsspannungen wesentlich höher, als bei hohen Spannungen.

2.2.2 Host-Schnittstellen (Steckverbinder ST 2 und ST 3)

2.2.2.1 Belegung Stecker ST 2, M8-Einbaustecker 4-Pol (Ethernet):

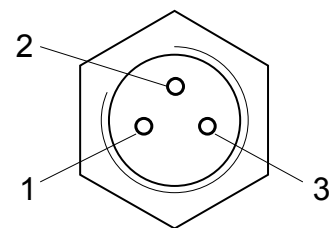
ST 2	Signal des CPU-Moduls	Signal bei X-MAX-E	RJ45 für Hub	RJ45 für PC
1	Stecker A Pin 1	TX+	1	3
2	Stecker A Pin 2	TX-	2	6
3	Stecker A Pin 3	RX+	3	1
4	Stecker A Pin 4	RX-	6	2



Die Belegung des RJ45 Steckers ist für den Anschluss des Kabels an einen Hub oder zum direkten Anschluss an einen PC angegeben.

2.2.2.2 Belegung Stecker ST 3, M8-Einbaustecker 3-Pol (serielle Host-Schnittstelle):

ST 3	Signal des CPU-Moduls	Signal bei X-MAX-E	Pin am D-SUB-9
1	Stecker A Pin 8	TXD	2
2	Stecker A Pin 6	RXD	3
3	ST5 Pin 10	XGND	5



Die Belegung der D-SUB-9 Buchse ist für den direkten Anschluss an einen PC angegeben.

2.2.3 D-SUB-9 Stecker für CAN 1 und CAN 2

Die beiden CAN-Busse sind untereinander und vom Rest der CANbox® galvanisch getrennt. Die Optionen der CAN-Schnittstellen sind im Abschnitt Software beschrieben.

D-SUB-9 Stecker für CAN-Kanal 1	
Signal	Pin D-SUB
CAN1-GND	3
CAN1-L	2
CAN1-H	7

D-SUB-9 Stecker für CAN-Kanal 2	
Signal	Pin D-SUB
CAN2-GND	3
CAN2-L	2
CAN2-H	7

2.2.4 SMA-Stecker für Antenne

An der Rückseite der CANbox[®] ist eine SMA-Buchse für den Anschluss der WLAN-Antenne vorhanden. Hier können verschiedene Antennen direkt angeschlossen werden. Erhältlich sind u.a. folgende Antennen:

- Kurze Aufsteckantenne (93mm) 90° abgewinkelt mit 2,1dBi Gewinn (Standardantenne)
- Lange Aufsteckantenne (195mm) mit Knickgelenk, 7dBi Gewinn (auf Anfrage)
- Tischantenne (190mm Höhe) mit 1,8m Anschlusskabel, 5dBi Gewinn (Standardantenne)
- Magnetfußantenne mit 1,5m Anschlusskabel, 5dBi Gewinn (auf Anfrage)

3 Inbetriebnahme

3.1 CANbox[®] - Setup

Das CANbox[®] - Setup mit Namen CANbox.exe dient zur Installation der verschiedenen Komponenten, die für den Betrieb der CANbox[®] zur Verfügung stehen. Es befindet sich auf der SORCUS-CD unter \FILES\CANbox.exe. Nach Start der Installation erscheint ein Dialog zur Auswahl der Komponenten. Es stehen folgende Komponenten zur Verfügung:

- Win32 driver: Treiber und Konfigurations-Tool zur CANbox[®] Verwaltung unter Win32-Betriebssystemen
- Win32 library: Bibliothek zur Programmierung des Zugriffs auf die CANbox[®] unter Win32-Betriebssystemen (C, C++, VB6.0, VB.NET)
- Win32 samples: Beispiele zur Programmierung des Zugriffs auf die CANbox[®] unter Win32-Betriebssystemen
- Win32 tools: Tools für die CANbox[®] unter Win32-Betriebssystemen
- POCKET PC 2003 driver: Treiber und Konfigurations-Tool zur CANbox[®] Verwaltung unter POCKET PC 2003
- POCKET PC 2003 SE driver: Treiber und Konfigurations-Tool zur CANbox[®] Verwaltung unter POCKET PC 2003 Second Edition
- POCKET PC 2003 library: Bibliothek zur Programmierung des Zugriffs auf die CANbox[®] unter POCKET PC 2003 (C, C++, VB.NET)
- POCKET PC 2003 samples: Beispiele zur Programmierung des Zugriffs auf die CANbox[®] unter POCKET PC 2003
- POCKET PC 2003 tools: Tools für die CANbox[®] unter POCKET PC 2003
- Documentation: Dokumentationen zur CANbox[®]

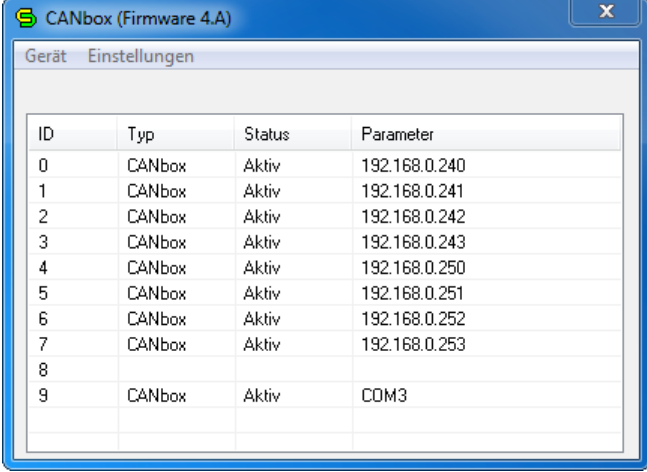
Nach Auswahl der Komponenten erscheint ein Auswahl-Dialog zur Angabe des Installationspfades. Nach Angabe eines gültigen Pfades erfolgt die Installation der ausgewählten Komponenten.

3.2 CANbox[®] - Configtool unter Win32 Betriebssystemen

3.2.1 Bedienung

Das CANbox[®] - Configtool dient zur Verwaltung und Konfiguration der CANbox[®](en). Es wird in der Systemsteuerung unter dem Namen CANbox[®] abgelegt und kann von dort gestartet werden.

Es enthält eine Liste installierter CANbox[®]en. Jeder CANbox[®] ist eine eindeutige Id zugeordnet, die für die Kommunikation mit dieser CANbox[®] erforderlich ist.



The screenshot shows a window titled "CANbox (Firmware 4.A)" with a menu bar containing "Gerät" and "Einstellungen". Below the menu bar is a table with the following data:

ID	Typ	Status	Parameter
0	CANbox	Aktiv	192.168.0.240
1	CANbox	Aktiv	192.168.0.241
2	CANbox	Aktiv	192.168.0.242
3	CANbox	Aktiv	192.168.0.243
4	CANbox	Aktiv	192.168.0.250
5	CANbox	Aktiv	192.168.0.251
6	CANbox	Aktiv	192.168.0.252
7	CANbox	Aktiv	192.168.0.253
8			
9	CANbox	Aktiv	COM3

CANbox[®] Konfiguration unter Win32-Betriebssystemen

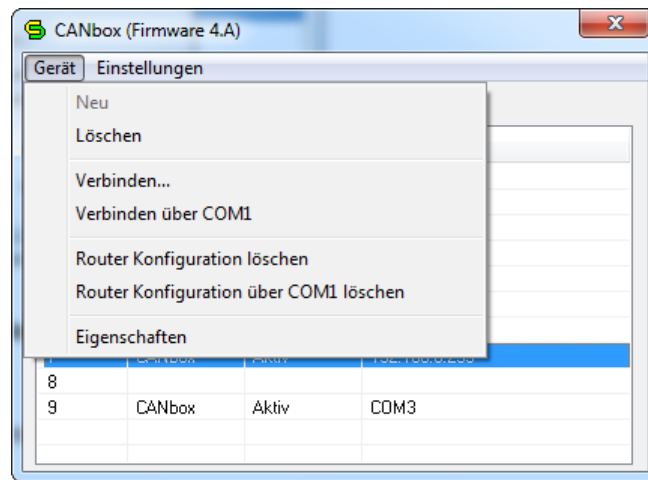
Sie können eine CANbox[®] mit einem Doppelklick auf eine leere Zeile oder nach Auswahl einer leeren Zeile mit dem Befehl Neu im Menü Gerät installieren.

Mit einem Doppelklick auf eine bereits installierte CANbox[®] oder mit dem Befehl Eigenschaften im Menü Gerät können Sie deren Kommunikationseinstellungen ändern.

Die Verbindung mit einer installierten CANbox[®] ist nur dann möglich, wenn die Kommunikationsparameter mit der tatsächlichen Konfiguration der CANbox[®] übereinstimmen. Kann die Verbindung hergestellt werden, so können die Kommunikationseinstellungen der CANbox[®] geändert werden. Änderungen werden mit einem Neustart der CANbox[®] wirksam.

Des Weiteren können die Kommunikationsparameter einer installierten CANbox[®] über eine serielle Schnittstelle geändert werden. Hierzu ist eine gekreuzte serielle Verbindung zwischen dem PC und der CANbox[®] nötig.

3.2.2 CANbox[®] Konfiguration Menü



CANbox[®] Verbindungs-Konfiguration unter Win32-Betriebssystemen

Das Menü Gerät enthält folgende Befehle:

3.2.2.1 Neu

Eine CANbox[®] installieren. Dieser Befehl öffnet den CANbox[®] Konfigurationsdialog (3.2.3)

3.2.2.2 Löschen

Eine installierte CANbox[®] aus der Liste löschen.

3.2.2.3 Verbinden

Mit einer CANbox[®] mit den konfigurierten Parametern verbinden. Dieser Befehl öffnet den Dialog CANbox[®] Einstellungen.

3.2.2.4 Verbinden über COM

Mit einer CANbox[®] über eine serielle Verbindung verbinden. Dieser Befehl öffnet den Dialog CANbox[®] Einstellungen. Die serielle Schnittstelle des PC wird im Menü Einstellungen festgelegt.

3.2.2.5 Router Konfiguration löschen

Die in der CANbox[®] gespeicherte Router Konfiguration[®] löschen. Diese Änderung wird nach einem Neustart der CANbox[®] wirksam.

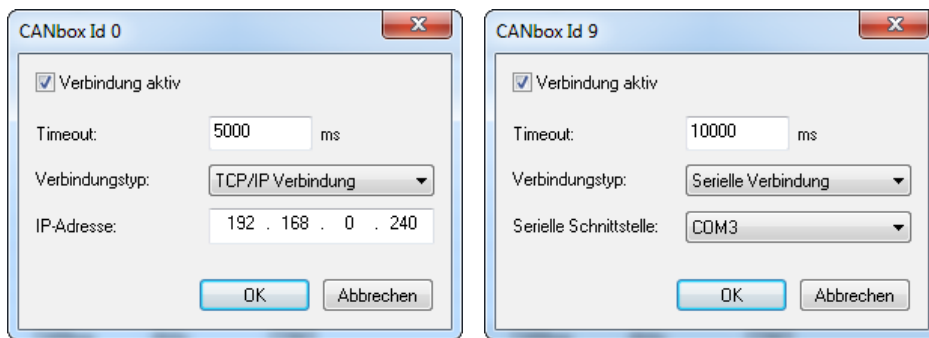
3.2.2.6 Router Konfiguration über COM löschen

Die in der CANbox[®] gespeicherte Router Konfiguration[®] über eine serielle Verbindung löschen. Diese Änderung wird nach einem Neustart der CANbox[®] wirksam.

3.2.2.7 Eigenschaften

Die Kommunikationsparameter einer installierten CANbox[®] ändern. (3.2.3)

3.2.3 CANbox® Konfigurationsdialog



CANbox® Konfigurationsdialog

In diesem Dialog werden die Verbindungsparameter zu einer CANbox® eingestellt.

Verbindung aktiv:

Hier kann die Verbindung zu einer CANbox® deaktiviert werden. Eine deaktivierte CANbox® kann vom PC nicht angesprochen werden.

Timeout:

Zeitsperre für den Versuch eine Verbindung zu der CANbox® herzustellen.

Verbindungstyp:

Wählen Sie "TCP/IP Verbindung" für eine Verbindung über das Netzwerk oder "Serielle Verbindung" bei Verwendung eines seriellen Kabels.

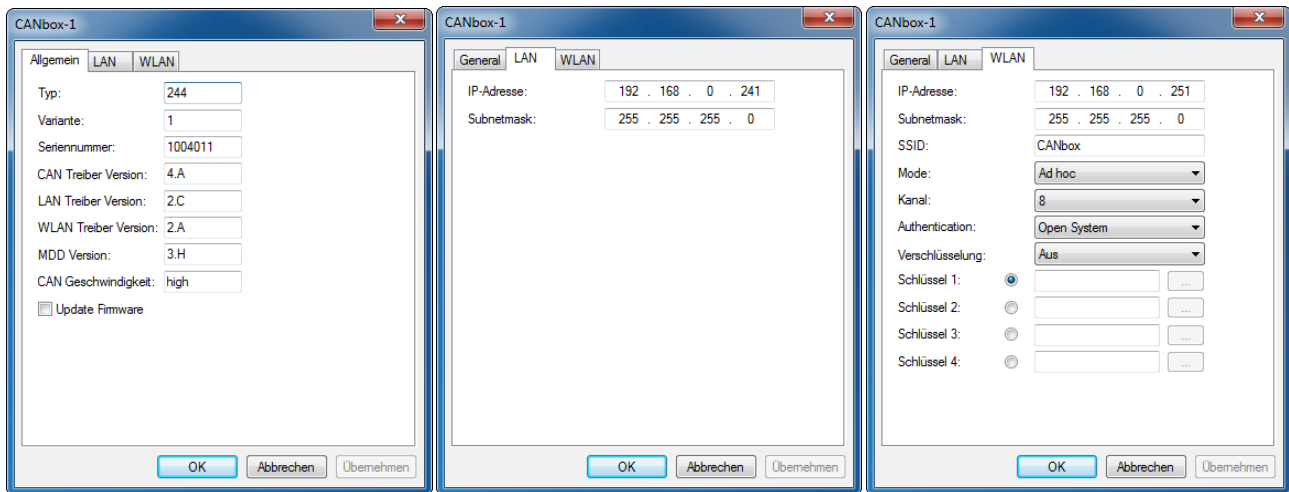
IP – Adresse:

IP – Adresse der CANbox® (nur bei "TCP/IP Verbindung")

Serielle Schnittstelle:

Schnittstelle des PC (nur bei "Serielle Verbindung")

3.2.4 CANbox® Eigenschaften Dialog



CANbox® Eigenschaften Dialog

3.2.4.1 Allgemein

Diese Seite zeigt allgemeine Informationen über die CANbox® und die geladenen Treiber. Ist die Version der geladenen Software und / oder Treiber kleiner als die des Konfigurationstools kann ein Firmware Update durch setzen des Hakens bei „Update Firmware“ und Bestätigung mit OK durchgeführt werden. Die neue Software wird nach einem Neustart der CANbox® aktiviert.

3.2.4.2 LAN

Hier werden die Parameter für eine kabelgebundene LAN Verbindung eingegeben. Diese bestehen aus der IP – Adresse und der zugehörigen Subnetmask. Die Einstellungen im Lieferzustand sind:

- IP – Adresse: 192.168.0.240
- Subnetmask: 255.255.255.0

3.2.4.3 WLAN

Diese Seite enthält alle benötigten Parameter für eine WLAN Verbindung:

Feld	Beschreibung	Lieferzustand
IP – Adresse	WLAN IP – Adresse der CANbox®	192.168.0.250
Subnetmask	WLAN Subnetmask der CANbox®	255.255.255.0
SSID	Netzwerkname max. 34 Zeichen	CANbox®
Kanal	WLAN Kanal (1 bis 14)	10
Authentication	Netzwerkauthentifizierung	Open System
Verschlüsselung	Datenverschlüsselung	Aus
Schlüssel 1	Netzwerkschlüssel (64-Bit oder 128-Bit)	
Schlüssel 2	Netzwerkschlüssel (64-Bit oder 128-Bit)	

Schlüssel 3	Netzwerkschlüssel (64-Bit oder 128-Bit)	
Schlüssel 4	Netzwerkschlüssel (64-Bit oder 128-Bit)	

Die Parameter sollten direkt nach der Installation der CANbox[®] geändert werden um Missbrauch durch Dritte vorzubeugen.

Änderung der Parameter werden nach einem Neustart der CANbox[®] wirksam.

3.3 Erste Verbindung mit einer CANbox[®]

3.3.1 Verbindung über LAN

Die Einstellungen der CANbox[®] im Lieferzustand lauten

- IP – Adresse: 192.168.0.240
- Subnetmask: 255.255.255.0

Für einen Verbindungsaufbau führen Sie folgende Schritte durch:

1. Verbinden Sie die CANbox[®] mit einem Netzkabel mit einem Switch im selben Netzwerk wie Ihr PC. Für eine direkte Verbindung benötigen Sie ein CrossOver – Kabel.
2. Ändern Sie die TCP/IP Einstellungen ihres PC. Stellen Sie die IP – Adresse auf 192.168.0.10 und die Subnetmask auf 255.255.255.0 ein.
3. Öffnen Sie das CANbox[®] Konfigurationstool in der Systemsteuerung und installieren Sie eine neue CANbox[®] mit den Liefereinstellungen. (3.2.1)
4. Verbinden Sie sich mit dieser CANbox[®] (Doppelklick).
5. Ändern Sie die Einstellungen der CANbox[®] auf die gewünschten Werte. (3.2.4)
6. Schließen Sie das CANbox[®] Konfigurationstool.
7. Stellen Sie die TCP/IP Einstellungen des PC auf die alten Werte.
8. Durch einen Neustart der CANbox[®] werden die geänderten Einstellungen übernommen.

3.3.2 Verbindung über serielle Schnittstelle

Sie benötigen ein CANbox[®] serial cable und eine serielle Schnittstelle oder einen USB – Seriell Adapter im PC.

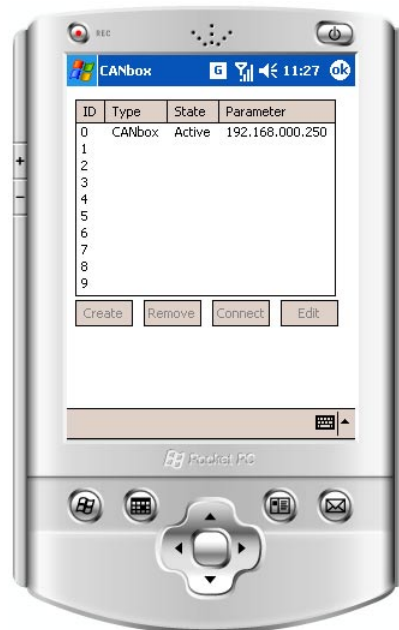
Für einen Verbindungsaufbau führen Sie folgende Schritte durch:

1. Verbinden Sie die CANbox[®] mit dem gewählten COM – Port des PC.
2. Öffnen Sie das CANbox[®] Konfigurationstool.
3. Markieren Sie den vorhandenen Eintrag oder installieren Sie eine neue CANbox[®].
4. Wenn Sie eine andere serielle Schnittstelle als COM1 verwenden wollen, wählen Sie diese im Menü Einstellungen aus.
5. Wählen Sie im Menü Gerät „Verbinden über COMx“

6. Ändern Sie die Einstellungen der CANbox[®] auf die gewünschten Werte. (3.2.4)
7. Durch einen Neustart der CANbox[®] werden die geänderten Einstellungen übernommen.

3.4 POCKET PC 2003

Auch über den Pocket PC 2003 ist eine Verbindung zur CANbox[®] möglich.



CANbox[®] Konfiguration unter POCKET PC 2003

Nach Auswahl einer verfügbaren **Board ID** kann über die Schaltfläche **Create** bzw. das Kontextmenü eine CANbox[®] vergleichbar mit der Verbindung via Netzwerk installiert werden. Ein Doppel-Klick auf eine CANbox[®] oder die Betätigung der Schaltfläche **Edit** ermöglicht die Konfiguration der CANbox[®] Verbindung.

Auch hier müssen die Kommunikationsparameter zwischen Pocket PC 2003 und CANbox[®] übereinstimmen. Nach erfolgter Verbindung können diese Parameter und andere Einstellungen geändert werden, wobei unter **POCKET PC 2003** nicht alle verfügbaren Parameter gesetzt werden können.

Die Fenster und Konfigurationen weichen nur minimal von der unter Win32-Betriebssystemen ab und können daher ähnlich verwendet werden

3.5 Einrichtungsbeispiele

In den nachfolgenden Kapiteln werden Beispiele zur Einrichtung von CANbox[®]en gegeben.

3.5.1 Einrichtungsbeispiel unter Windows XP

Im nachfolgenden Beispiel wird die Einrichtung einer AdHoc WLAN Verbindung zwischen einem PC und einer CANbox[®] mit Default-Einstellungen beschrieben. Diese Einrichtung erfolgt ohne Verwendung von Drittanbietersoftware und setzt einen fertig eingerichteten, aktivierten WLAN Adapter und eine betriebsbereite CANbox[®] voraus.

- Öffnen Sie die Netzwerkverbindungen durch Anwahl von **Alle Verbindungen anzeigen** im Menü **Verbinden mit** des **Start**-Menüs
- Wählen Sie die **Drahtlose Netzwerkverbindung** aus, die für die Verbindung mit der CANbox[®] verwendet werden soll und öffnen Sie deren Eigenschaftsdialog (Kontextmenü **Eigenschaften**)
- Wählen Sie auf der Seite **Allgemein** im Auswahlfeld **Diese Verbindung verwendet folgende Elemente** das Element **Internetprotokoll (TCP/IP)** aus und betätigen die Schaltfläche **Eigenschaften**
- Selektieren Sie das Auswahlfeld **Folgende IP-Adresse** verwenden, geben eine geeignete **IP- Adresse** (z.B. 192.168.0.200) mit **Subnetzmaske** (255.255.255.0) ein und verlassen den Dialog über die Schaltfläche **OK**

Falls das Netzwerk CANbox[®] bei den verfügbaren Netzwerken (**Drahtlose Netzwerke - Verfügbare Netzwerke - Drahtlosnetzwerke anzeigen**) nicht angezeigt wird, müssen Sie dieses mit nachfolgend beschriebener Vorgehensweise erzeugen, andernfalls wählen Sie die Netzwerk CANbox[®] aus der Liste der verfügbaren Netzwerke aus.

- Wechseln Sie auf die Seite **Drahtlose Netzwerke** und betätigen die Schaltfläche **Hinzufügen**
- Geben Sie im erscheinenden Dialog auf der Seite **Zuordnung** im Feld **Netzwerkname (SSID)** den Namen CANbox[®] ein, wählen Sie **Deaktivieren** bei der **Datenverschlüsselung** aus und aktivieren das Kontrollkästchen **Dies ist ein Computer- zu Computer-Netzwerk (AdHoc)**
- Verlassen den Dialog über die Schaltfläche **OK**

Bitte achten Sie darauf, dass keine andere Netzwerkverbindung (z.B. Firmen- Netzwerk) eine IP- Adresse der Art 192.168.0.xxx verwendet, da sonst keine Verbindung möglich ist.

3.5.2 Einrichtungsbeispiel 1 unter POCKET PC 2003

Im nachfolgenden Beispiel wird die Einrichtung einer AdHoc WLAN Verbindung zwischen einem Toshiba PDA mit integriertem WLAN und einer betriebsbereiten CANbox® mit Default-Einstellungen beschrieben. Diese Einrichtung sollte für andere PDAs ähnlich sein, kann sich aber in einigen Punkten unterscheiden.

- Stellen Sie beim PDA den Schalter für drahtlose Kommunikation auf **ON**
- Die **Wireless LED** des PDA sollte orange leuchten
- Wählen Sie im PDA Startmenü das Menü **Einstellungen** aus und wechseln Sie auf die Seite **Verbindungen**
- Dort wählen Sie das Symbol **Verbindungen** mit dem Stift aus und wechseln im nachfolgenden Dialog auf die Seite **Erweitert**
- Betätigen Sie die Schaltfläche **Netzwerkkarte**
- Auf der Seite **Drahtlos** klicken Sie im Feld **Drahtlose Netzwerke** auf **Neue Einstellungen**, falls das Netzwerk CANbox® noch nicht vorhanden ist, andernfalls können Sie sich mit dem CANbox® Netzwerk verbinden
- Es erscheint ein Dialog mit den Seiten **Allgemein** und **Authentifizierung**
- Auf der **Allgemein**-Seite geben Sie **CANbox®** als Netzwerknamen ein, wählen **Verbindet mit Firma** und aktivieren das **Kontrollkästchen Dies ist eine „Gerät zu Computer“ (AdHoc) Verbindung**
- Auf der Seite **Authentifizierung** deaktivieren Sie das Kontrollkästchen **Datenverschlüsselung (WEP aktiviert)**
- Verlassen Sie den Dialog mit der **OK**-Taste
- Wechseln Sie auf die Seite **Netzwerkkarten** und selektieren Sie bei den **Adapttern** mit dem Stift **IEEE 802.11b WLAN Adapter**
- Aktivieren Sie das Auswahlfeld **Spezifische IP-Adresse** und geben Sie eine geeignete **IP-Adresse** (z.B. 192.168.0.200) mit **Subnetmask** (255.255.255.0) ein
- Bestätigen Sie Ihre Eingabe mit der **OK**-Taste und kehren zurück zur **Start**-Seite
- Bauen Sie über das CANbox® Symbol in der Systemsteuerung eine Verbindung zur CANbox® auf
- Sollte die Verbindung nicht zu Stande kommen, sollten Sie einen Soft Reset des PDA durchführen

3.5.3 Einrichtungsbeispiel 2 unter POCKET PC 2003

Im nachfolgenden Beispiel wird die Einrichtung einer AdHoc WLAN Verbindung zwischen einem HP iPAQ POCKET PC mit integriertem WLAN und einer betriebsbereiten CANbox[®] mit Default-Einstellungen beschrieben. Diese Einrichtung sollte für andere POCKET PCs ähnlich sein, kann sich aber in einigen Punkten unterscheiden.

- Aktivieren Sie den WLAN Adapter durch Anwahl von **WLAN** über das Menü **iPAQ Wireless** im Windows **Start**-Menü. Die WLAN-LED blinkt nun orange.
- Wählen Sie im POCKET PC-**Start**-Menü das Menü **Einstellungen** aus und wechseln Sie auf die Seite **Verbindungen**
- Dort wählen Sie das Symbol **Verbindungen** mit dem Stift aus und wechseln im nachfolgenden Dialog auf die Seite **Erweitert**
- Betätigen Sie die Schaltfläche **Netzwerkkarte**
- Auf der Seite **Drahtlos** klicken Sie im Feld **Drahtlose Netzwerke** auf **Neue Einstellungen**, falls das Netzwerk CANbox[®] noch nicht vorhanden ist, andernfalls können Sie sich mit dem CANbox[®] Netzwerk verbinden
- Es erscheint ein Dialog mit den Seiten **Allgemein** und **Authentifizierung**
- Auf der **Allgemein**-Seite geben Sie **CANbox[®]** als Netzwerknamen ein, wählen **Verbindet mit Firma** und aktivieren das **Kontrollkästchen Dies ist eine „Gerät zu Computer“ (AdHoc) Verbindung**
- Auf der Seite **Authentifizierung** deaktivieren Sie das Kontrollkästchen **Datenverschlüsselung (WEP aktiviert)**
- Verlassen Sie den Dialog mit der **OK**-Taste
- Wechseln Sie auf die Seite **Netzwerkkarten** und selektieren Sie bei den **Adaptern** mit dem Stift **iPAQ USB Wireless Adapter**
- Aktivieren Sie das Auswahlfeld **Spezifische IP-Adresse** und geben Sie eine geeignete **IP-Adresse** (z.B. 192.168.0.200) mit **Subnetmask** (255.255.255.0) ein
- Bestätigen Sie Ihre Eingabe mit der **OK**-Taste und kehren zurück zur **Start**-Seite
- Bauen Sie über das CANbox[®] Symbol in der Systemsteuerung eine Verbindung zur CANbox[®] auf
- Sollte die Verbindung nicht zu Stande kommen, sollten Sie einen Soft Reset des POCKET PC durchführen

3.5.4 Einrichtungsbeispiel unter POCKET PC 2003 SE (Second Edition)

Im nachfolgenden Beispiel wird die Einrichtung einer AdHoc WLAN Verbindung zwischen einem Toshiba e800 mit integriertem WLAN und einer betriebsbereiten CANbox® mit Default-Einstellungen beschrieben. Diese Einrichtung sollte für andere Pocket PCs ähnlich sein, kann sich aber in einigen Punkten unterscheiden.

- Stellen Sie beim Pocket PC den Schalter für drahtlose Kommunikation auf **ON**
- Die **Wireless LED** des Pocket PCs sollte orange leuchten
- Wählen Sie im Pocket PC-**Start**-Menü das Menü **Einstellungen** aus und wechseln Sie auf die Seite **Verbindungen**
- Dort wählen Sie das Symbol **Netzwerkkarten** mit dem Stift aus
- Auf der Seite **Drahtlos** klicken Sie im obersten Feld auf **Neu hinzufügen...**, falls das Netzwerk CANbox® noch nicht vorhanden ist, andernfalls können Sie sich mit dem CANbox® Netzwerk verbinden
- Es erscheint ein Dialog mit den Seiten **Allgemein, Netzwerkschlüssel** und **802.1x**
- Auf der **Allgemein**-Seite geben Sie **CANbox®** als Netzwerknamen ein, wählen **Verbundet mit Firma** und aktivieren das Kontrollkästchen **Dies ist eine „Gerät zu Computer“ (AdHoc) Verbindung**
- Auf der Seite **Netzwerkschlüssel** wählen Sie **Öffnen** im Listenfeld **Authentifizierung** sowie **Deaktiviert** im Feld **Datenverschlüsselung**
- Auf der Seite **802.1x** sollte das Kontrollkästchen **IEEE 802.1X-Netzwerkzugriffssteuerung verwenden** inaktiv sein
- Verlassen Sie den Dialog mit der **OK**-Taste
- Wechseln Sie auf die Seite **Netzwerkkarten** und wählen Sie im Listenfeld **Meine Netzwerkkarte verbindet mit** den Eintrag **Firma** aus
- Selektieren Sie bei den **Adaptern** mit dem Stift **IEEE 802.11b WLAN Adapter**
- Aktivieren Sie das Auswahlfeld **Spezifische IP-Adresse** und geben Sie eine geeignete **IP-Adresse** (z.B. 192.168.0.200) mit **Subnetmask** (255.255.255.0) ein
- Wechseln Sie zurück auf die Seite **Drahtlos** und halten Sie den Stift im obersten Feld auf **CANbox®**
- Im Kontextmenü der **CANbox®** können Sie nun den Menü-Eintrag **Verbinden** auswählen
- Bestätigen Sie Ihre Eingaben mit der **OK**-Taste und kehren zurück zur **Start**-Seite
- Bauen Sie über das CANbox® Symbol in der Systemsteuerung eine Verbindung zur CANbox® auf

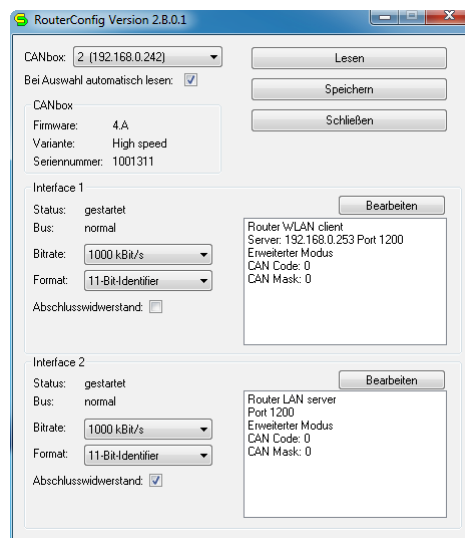
Sollte die Verbindung nicht zu Stande kommen, sollten Sie einen Soft Reset des POCKET PCs durchführen.

3.6 CANbox® als Router

Die CANbox® bietet ab der Firmware-Version 3.A zusätzlich die Möglichkeit als Router zu dienen. Hierzu können zwei CANbox®-en per LAN/WLAN miteinander verbunden werden.

Der Router Modus kann mittels Bibliotheksfunktionen und/oder dem mitgelieferten Tool „RouterConfig“ konfiguriert werden. Hierzu müssen die CANbox®-en in der Systemsteuerung installiert und verbunden sein.

Bei der Konfiguration muss jeweils eine CANbox® als TCP/IP - Server und eine CANbox® als TCP/IP - Client konfiguriert werden. Der angegebene Server-Port muss bei beiden CANbox®-en identisch sein, wobei der Wert zwischen 1200 und 65530 liegen muss.

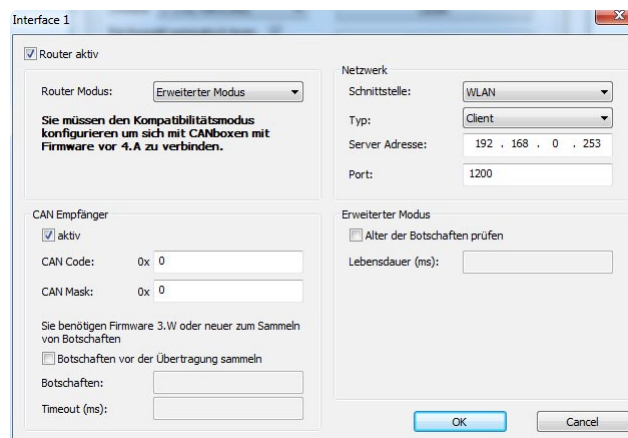


Konfigurationstool RouterConfig

Durch Klicken auf „Lesen“ werden die aktuellen Konfigurationsdaten sowie Version und Firmware aus der gewählten CANbox® gelesen. Bei CANbox®-en mit Firmware vor 3.X müssen vor dem Lesen die Interfaceparameter eingestellt werden.

Mit „Speichern“ wird die Konfiguration in die CANbox® geschrieben und aktiviert. Im Anschluss müssen CANbox®-en mit Firmware vor 3.X neu gestartet werden.

Das jeweilige CAN-Bus Interface wird durch Auswahl der Bitrate, das Botschaftsformat und den Abschlusswiderstand konfiguriert. Durch eine Klick auf „Bearbeiten“ wird das Konfigurationsfenster für die Routerverbindung geöffnet.



Router aktiv:

Die Routerfunktion für dieses Interface Ein- Ausschalten

Router Modus:

Ab Firmware Version 4.A wird der „erweiterte Modus“ unterstützt. Um diesen nutzen zu können müssen alle beteiligten CANbox®en über die Firmware 4.A oder neuer verfügen. Im erweiterten Modus ist die Übertragungssicherheit erhöht und die Angabe einer maximalen Lebensdauer von CAN-Botschaften möglich. Wir empfehlen, alle CANbox®en auf die aktuelle Firmware zu aktualisieren (siehe 3.2 CANbox® - Configtool) und nur den erweiterte Modus zu verwenden. Im „kompatiblen Modus“ kann eine Routerverbindung zu CANbox®en mit älterer Firmware aufgebaut werden.

CAN Empfänger:

Hier werden die vom CAN-Bus zu empfangenden Botschaften konfiguriert. Mit „Aktiv“ wird der Empfänger eingeschaltet. Die Werte in „CAN Code“ und „CAN Mask“ werden mit dem Tool AcceptanceFilterCalculator ermittelt.

Um die Übertragung durch das Netzwerk zu optimieren (TCP Overhead minimieren) können Botschaften vor der Übertragung gesammelt werden. Hierzu wird die Anzahl der zu sammelnden Botschaften sowie ein Timeout, nach dessen Ablauf auch einzelne Botschaften versandt werden, angegeben. Dies ist ab Firmware Version 3.W möglich.

Netzwerk:

Hier wird die Schnittstelle (LAN oder WLAN) ausgewählt. Des weiteren wird festgelegt, ob diese CANbox® als Server oder Client agiert. Um eine Verbindung herzustellen muss immer eine CANbox® als Server und eine als Client konfiguriert werden. Bei der Serverkonfiguration wird lediglich der Port angegeben. Beim Client muss die IP-Adresse der Server CANbox® angegeben werden.

Erweiterter Modus:

Wird die Prüfung des Alters der Botschaften eingeschaltet und eine maximale Lebensdauer angegeben, so stellt die Firmware sicher, dass an der Ziel CANbox® keine Botschaften auf dem CAN-Bus versandt werden, die „älter“ sind als die angegebene Lebensdauer. Das Alter einer Botschaft ist die Zeitspanne vom Empfang vom CAN-Bus der einen CANbox® bis zum Senden auf dem CAN-Bus der anderen CANbox®.

3.6.1 Konfigurationsbeispiele

3.6.1.1 Beispiel 1

CANbox[®] A Interface 1 wird mit CANbox[®] B Interface 1 über WLAN verbunden.

Von CANbox[®] A nach CANbox[®] B werden alle CAN Botschaften übertragen.

Von CANbox[®] B nach CANbox[®] A werden nur Botschaften mit Identifiern von 0x1000 bis 0x100F übertragen.

Die Interfaces der beiden CANbox[®]en werden auf die Bitrate der angeschlossenen CAN Busse eingestellt.

Beide CANbox[®]en werden auf das selbe Botschaftsformat eingestellt. (in diesem Fall 29bit)

	CANbox [®] A Interface 1	CANbox [®] B Interface 1
Router aktiv	Ja	
Router Modus	Erweitert	
CAN Empfänger aktiv	Ja	
CAN Code	0	0000100F
CAN Mask	0	1FFFFFF0
Botschaften sammeln	Nein	
Netzwerk Schnittstelle	WLAN	
Typ	Server	Client
Server Adresse	N/A	WLAN IP – Adresse von CANbox [®] A
Port	1200	
Alter prüfen	Nein	

3.6.1.2 Beispiel 2

CANbox[®] A Interface 2 wird mit CANbox[®] B Interface 2 über LAN verbunden.

Von CANbox[®] A nach CANbox[®] B werden nur Botschaften mit Identifiern von 0x2000 bis 0x3FFF übertragen.

Von CANbox[®] B nach CANbox[®] A werden keine CAN Botschaften übertragen.

Eine oder beide CANbox[®]en haben eine Firmware älter als 4.A.

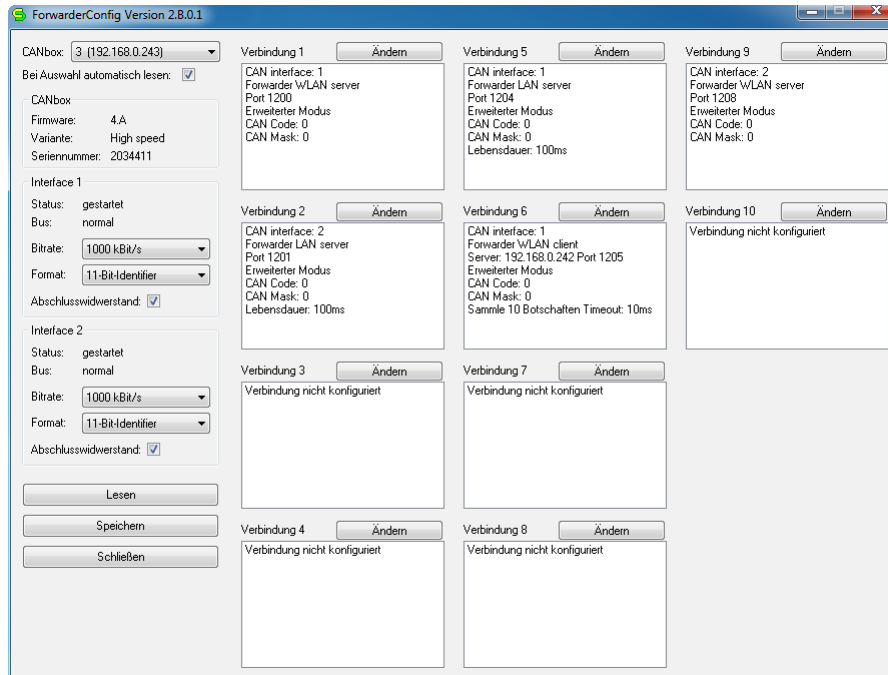
Die Interfaces der beiden CANbox[®]en werden auf die Bitrate der angeschlossenen CAN Busse eingestellt.

Beide CANbox[®]en werden auf das selbe Botschaftsformat eingestellt. (in diesem Fall 29bit)

	CANbox [®] A Interface 1	CANbox [®] B Interface 1
Router aktiv	Ja	
Router Modus	Kompatibel	
CAN Empfänger aktiv	Ja	Nein
CAN Code	00003FFF	N/A
CAN Mask	1FFFE000	N/A
Botschaften sammeln	Nein	N/A
Netzwerk Schnittstelle	LAN	
Typ	Client	Server
Server Adresse	LAN IP – Adresse von CANbox [®] B	N/A
Port	1201	
Alter prüfen	N/A	

3.7 Erweiterter Router (Forwarder)

Der Forwarder erlaubt bis zu 10 Routerverbindungen in einer CANbox®. Er kann verwendet werden ab Firmware 3.M. Alle Kombinationen aus Interface 1, Interface 2, Server, Client, LAN, WLAN können konfiguriert werden.



Konfigurationstool ForwarderConfig

Wenn „Bei Auswahl lesen“ aktiviert ist, wird die Konfiguration einer CANbox® direkt bei deren Auswahl gelesen.

Bei CANbox®en mit Firmware älter als 3.X müssen vor dem Lesen der Konfiguration die Interface Parameter eingestellt werden.

Die Konfiguration der einzelnen Verbindungen erfolgt genauso wie bei der Router Konfiguration. Zusätzlich muss das Interface für die Verbindung angegeben werden.

3.7.1 Konfigurationsbeispiele

3.7.1.1 Beispiel 1

Von CANbox® A Interface 1 sollen alle CAN Botschaften via WLAN zu 3 anderen CANbox®en (B – D) und via LAN zu 2 weiteren CANbox®en übertragen werden.

An CANbox® B und C ist Interface 1 angeschlossen, und CANbox® D bis F ist interface 2 angeschlossen.

Die CANbox®en B – F sollen nur Botschaften mit Identifiern 0x11000 – 0x11003 zu CANbox® A übertragen.

Die Interfaces der CANbox®-en werden auf die Bitrate der angeschlossenen CAN Busse eingestellt.

Die beteiligten Interfaces aller CANbox®-en werden auf das selbe Botschaftsformat eingestellt. (in diesem Fall 29bit)

CANbox® A:

Verbindung	1	2	3	4	5
Interface	1				
Forwarder aktiv	Ja				
Forwarder Modus	Erweitert				
Empfänger aktiv	Ja				
CAN Code	0				
CAN Mask	0				
Botschaften sammeln	Nein				
Netzwerk Schnittstelle	WLAN			LAN	
Typ	Server				
Server Adresse	N/A				
Port	1200	1201	1202	1203	1204
Alter prüfen	Nein				

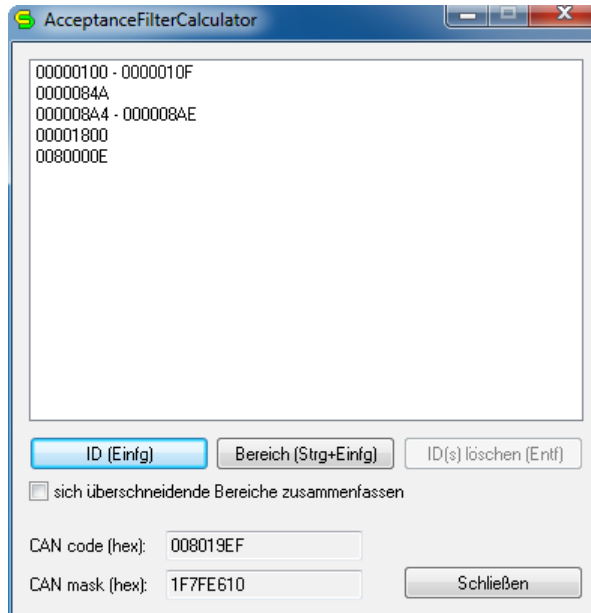
CANbox® B – F

Wählen Sie eine beliebige Verbindung, in der sie die Konfiguration speichern.

CANbox®	B	C	D	E	F
Interface	1	1	2	2	2
Forwarder aktiv	Ja				
Forwarder Modus	Erweitert				
Empfänger aktiv	Ja				
CAN Code	00011003				
CAN Mask	1FFFFFFC				
Botschaften sammeln	Nein				
Netzwerk Schnittstelle	WLAN			LAN	
Typ	Client				
Server Adresse	WLAN IP Adresse von CANbox® A			LAN IP Adresse von CANbox® A	
Port	1200	1201	1202	1203	1204
Alter prüfen	Nein				

3.8 Konfigurationstool AcceptanceFilterCalculator

Mit diesem Tool werden die CAN – Code und CAN – Mask für Akzeptanzfilter berechnet.



Konfigurationstool AcceptanceFilterCalculator

Der CAN Code und CAN – Mask werden direkt nach Einfügen oder Löschen von Identifiern neu berechnet.

Die CAN – Mask bestimmt, welche Bits eines Identifiers mit CAN – Code übereinstimmen müssen, dass die Botschaft empfangen wird.

Dies bedeutet: Eine Botschaft wird dann empfangen bzw. geroutet, wenn

$$\text{Identifier \& CAN – Mask} == \text{CAN – Code}$$

Der Nachteil dieser Filterung ist folgender:

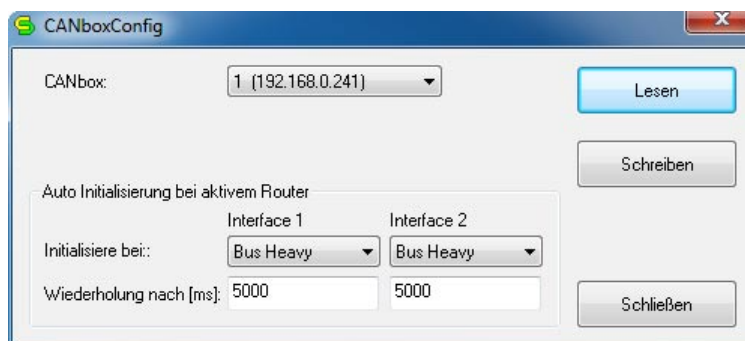
Wenn z.B. die Identifier 0x101 und 0x10E empfangen werden sollen müssen in der Maske die Bits 0 bis 3 ungültig (0) gesetzt werden, da diese in den Identifiern verschieden sind. Dies hat zur Folge, dass alle Identifier von 0x100 bis 0x10F empfangen werden.

3.9 Automatische Neuarithrierung des CAN-Bus im Routermodus

Ist der CAN-Bus stark gestört geht der CAN Controller zunächst in den Zustand „heavy“, dann in den Zustand „Off“. Im Zustand „Off“ wird der Controller vom CAN – Bus getrennt. Eine automatische Neuarithrierung erfolgt nicht. Im Zustand „heavy“ erfolgt eine Neuarithrierung nur dann, wenn ein anderer Teilnehmer am Bus Botschaften sendet. In beiden Zuständen werden keine Botschaften über den Bus versandt.

Ab der Firmware 3.W kann im Router- oder Forwardermodus für jedes Interface eine „Auto Initialisierung“ des CAN Controllers konfiguriert werden. Hierbei werden das Level („heavy“ oder „Off“), in dem die aktive Neuarithrierung erfolgt, sowie ein Wiederholungs-timeout festgelegt.

Hierfür wird das Tool „CANboxConfig“ mitgeliefert.



CANboxConfig Konfiguration Autoreset

Nach Auswahl der CANbox[®] wird durch einen klick auf „Lesen“ die aktuelle Konfiguration ausgelesen.

Initialisiere bei:

Hier wird das Fehlerlevel für die Neuarithrierung des jeweiligen Interface festgelegt.

Wiederholung nach:

Legen Sie den mindest Zeitabstand für zwei Neuarithrierungen fest.

Durch einen Klick auf „Schreiben“ wird die Konfiguration in die CANbox[®] geschrieben.

4 CANbox[®] – Bibliothek

Die Bibliothek stellt alle zur CAN Kommunikation notwendigen Funktionen zur Verfügung. Alle Funktionen haben einen Fehlercode als Rückgabewert. Die Funktion liefert CB_ERROR_OK zurück, falls sie erfolgreich ausgeführt wurde. Die CANbox[®] Bibliothek ermöglicht es dem Benutzer, die CAN Hardware zu konfigurieren und zu initialisieren. Des Weiteren bietet sie Funktionen zum Empfangen und Senden von CAN Identifier Daten. Die Bibliothek wird sowohl auf Win32 als auch auf POCKET PC 2003 Betriebssystemen zur Verfügung gestellt.

4.1. Strukturen und Definitionen

4.1.1 CANBOX_VERSION_INFO

Struktur zur Aufnahme von Versions-Informationen.

4.1.1.1 Elemente

USHORT	usBuild:	Build Nummer
UCHAR	ucMinor:	Nebenversion
UCHAR	ucMajor:	Hauptversion
UCHAR	ucDay:	Erstellungstag
UCHAR	ucMonth:	Erstellungsmonat
USHORT	usYear:	Erstellungsjahr

4.1.2 CANBOX_TIME_STAMP

Struktur mit 64-Bit-Zeitstempel.

4.1.2.1 Parameter

ULONG	ulTimeLo:	Low – Dword des Zeitstempels
ULONG	ulTimeHi:	High – Dword des Zeitstempels

4.1.3 CANBOX_SCAN_INFO

Struktur zur Aufnahme von Device Informationen installierter CANboxen.

4.1.3.1 Elemente

USHORT	usDeviceId:	Id der CANbox [®]
USHORT	usConnection:	Art der Verbindung
ULONG	ulTimeout:	Timeout – Zeit

4.1.4 CANBOX_DEVICE_INFO

Struktur zur Aufnahme von Device Informationen verbundener CANboxen.

4.1.4.1 Parameter

USHORT	usType:	Typ der CANbox [®]
USHORT	usVariant:	Variante der CANbox [®]
ULONG	ulSerial:	Serien-Nummer der CANbox [®]
CANBOX_VERSION_INFO	sDriverCan:	Version des CAN Treibers
CANBOX_VERSION_INFO	sDriverLan:	Version des LAN Treibers
CANBOX_VERSION_INFO	sDriverWlan:	Version des WLAN Treibers

4.1.5 CANBOX_RESET_EX

Struktur zur Übergabe bei einem erweiterten Reset.

4.1.5.1 Parameter

USHORT	usRouterRestart:	Neustart der Router nach Reset
--------	------------------	--------------------------------

4.1.6 CANBOX_INTERFACE

Struktur zur Parametrierung einer CAN-Schnittstelle unter Angabe der Bit-Rate.

4.1.6.1 Parameter

ULONG	ulBufferSizeMax:	Maximale Anzahl an Telegrammen, die auf der CANbox [®] zwischengespeichert werden (0-150.000; 0: maximale Anzahl)
USHORT	usIndex:	Index der Schnittstelle (1, 2 bei Verwendung der Bibliothek; 0, 1 bei Verwendung der Socket Schnittstelle)
USHORT	usBitRate:	Bit-Rate
USHORT	usShutOff:	Ein-/ Ausschalten des Abschlusswiderstands
USHORT	usFormat:	CAN-Spezifikation (11-Bit-/ 29-Bit-Identifizier)

4.1.6.2 Bit-Raten

CB_BITRATE_0010_KBIT:	10 kBit/s
CB_BITRATE_0020_KBIT:	20 kBit/s
CB_BITRATE_0050_KBIT:	50 kBit/s
CB_BITRATE_0080_KBIT:	80 kBit/s
CB_BITRATE_0083_3_KBIT:	83.3 kBit/s
CB_BITRATE_0100_KBIT:	100 kBit/s
CB_BITRATE_0125_KBIT:	125 kBit/s
CB_BITRATE_0250_KBIT:	250 kBit/s
CB_BITRATE_0500_KBIT:	500 kBit/s
CB_BITRATE_0666_KBIT:	666 kBit/s
CB_BITRATE_0800_KBIT:	800 kBit/s
CB_BITRATE_1000_KBIT:	1 MBit/s

4.1.6.3 Abschlusswiderstand

CB_NO_SHUT_OFF:	Ausgeschaltet
CB_SHUT_OFF:	Eingeschaltet

4.1.6.4 Format

CB_FORMAT_MIXED:	11-Bit- und 29-Bit-Identifizier
CB_FORMAT_11_BIT:	11-Bit-Identifizier
CB_FORMAT_29_BIT:	29-Bit-Identifizier
CB_FORMAT_STD:	Standard-Modus
CB_FORMAT_BUFFER:	Puffer-Modus: Jeder Identifizier bzw. Akzeptanzfilter besitzt eine über usBufferSize konfigurierbare Puffergröße.

4.1.7 CANBOX_INTERFACE_EX

Erweiterte Struktur zur Parametrierung einer CAN-Schnittstelle unter Angabe der Bit-Timing-Parameter.

4.1.7.1 Parameter

ULONG	ulBufferSizeMax:	Maximale Anzahl an Telegrammen, die auf der CANbox [®] zwischengespeichert werden (0-150.000; 0: maximale Anzahl)
USHORT	usIndex:	Index der Schnittstelle (1, 2 bei Verwendung der Bibliothek; 0, 1 bei Verwendung der Socket Schnittstelle)
USHORT	usTq:	Time Quantum in Nanosekunden
USHORT	usTseg1:	Zeit vor dem nominellen Abtastzeitpunkt
USHORT	usTseg2:	Zeit nach dem nominellen Abtastzeitpunkt
USHORT	usTsjw:	Synchronisationssprungweite
USHORT	usSpB:	Zahl der Abtastungen pro Bit
USHORT	usShutOff:	Ein-/ Ausschalten des Abschlusswiderstands
USHORT	usFormat:	CAN-Spezifikation (11-Bit-/ 29-Bit-Identifizier)

4.1.7.2 Abschlusswiderstand

CB_NO_SHUT_OFF:	Ausgeschaltet
CB_SHUT_OFF:	Eingeschaltet

4.1.7.3 Format

CB_FORMAT_MIXED:	11-Bit- und 29-Bit-Identifizier
CB_FORMAT_11_BIT:	11-Bit-Identifizier
CB_FORMAT_29_BIT:	29-Bit-Identifizier
CB_FORMAT_STD:	Standard-Modus
CB_FORMAT_BUFFER:	Puffer-Modus: Jeder Identifizier bzw. Akzeptanzfilter besitzt eine über usBufferSize konfigurierbare Puffergröße.

4.1.8 CANBOX_INTERFACE_STATE

Struktur zur Aufnahme des Status einer Schnittstelle.

4.1.8.1 Parameter

CANBOX_TIME_STAMP	sTime:	Zeit-Stempel der letzten Bus-Status-Änderung
USHORT	usBusState:	Bus-Status
USHORT	usBusInfo:	Info zum Bus-Status
USHORT	usUserState:	User-Status

4.1.8.2 Bus-Status

CB_BUS_STATE_ON:	CAN-Schnittstelle nimmt am Bus-Verkehr teil.
CB_BUS_STATE_HEAVY:	Stark gestörter Bus
CB_BUS_STATE_OFF:	Aufgrund von Busstörungen hat sich der CAN-Controller automatisch vom CAN-Bus zurückgezogen.
CB_BUS_STATE_LINE:	Vom Transceiver signalisierter Leitungsfehler (nur für fehler-tolerante Schnittstellen)

4.1.8.3 Bus-Info

CB_BUS_INFO_STUFF:	Stuff-Error (mehr als 5 gleiche Bits in einem Telegramm hintereinander)
CB_BUS_INFO_FORM:	Form-Error (Format-Fehler im Telegramm)
CB_BUS_INFO_ACK:	Ack-Error (Gesendetes Telegramm wurde von keinem Teilnehmer quittiert)
CB_BUS_INFO_BIT_LO:	Bit-Error-0 (beim Senden eines Bits wurde der eine 0 statt eine 1 zurückgelesen)
CB_BUS_INFO_BIT_HI:	Bit-Error-1 (beim Senden eines Bits wurde der eine 1 statt eine 0 zurückgelesen)
CB_BUS_INFO_CRC:	CRC-Error (fehlerhafte Prüfsumme)

4.1.8.4 User-Status

CB_USER_STATE_OFF:	Schnittstelle wurde noch nicht konfiguriert.
CB_USER_STATE_STARTED:	Schnittstelle wurde konfiguriert und aktiviert.
CB_USER_STATE_STOPPED:	Schnittstelle wurde konfiguriert und deaktiviert.

4.1.9 CANBOX_INTERFACE_READ

Struktur mit Empfangs-Daten einer Schnittstelle.

4.1.9.1 Parameter

CANBOX_TIME_STAMP	sTime:	Zeitstempel
ULONG	uId:	CAN Id des Identifiers
ULONG	ulDataSize:	Anzahl gültiger Datenbytes des Identifiers
UCHAR	ucData[8]: (ucData0 bis ucData7 bei VB)	Datenbytes des Identifiers

4.1.10 CANBOX_INTERFACE_WRITE

Struktur mit Sende-Daten einer Schnittstelle.

4.1.10.1 Parameter

CANBOX_TIME_STAMP	sTime:	Zeitstempel
CANBOX_HANDLE	hIdentifier:	Handle des zu sendenden Identifiers
ULONG	ulDataSize:	Anzahl zu sendender Datenbytes des Identifiers
UCHAR	ucData[8]: (ucData0 bis ucData7 bei VB)	Datenbytes des Identifiers

4.1.11 CANBOX_IDENTIFIER

Struktur zur Konfiguration eines Identifiers.

4.1.11.1 Parameter

ULONG	uId:	CAN Identifier. Im gemischten Modus kann das höchstwertige Bit zur Kennzeichnung eines 29-Bit-Identifiers gesetzt werden. Dieses Bit ist dann auch in den zurückgelieferten Daten gesetzt.
USHORT	usType:	Typ des Identifiers
USHORT	usFlags:	Optionen für den Identifiers
USHORT	usTimeout:	Zeit in μ s, die bei aktiven Empfängern auf den Empfang der Antwort gewartet wird.
USHORT	usReserved:	Reservierter Parameter, der bei Verwendung der Socket Schnittstelle auf den nullbasierten Index der zugehörigen Schnittstelle gesetzt werden muss.
USHORT	usBufferSize:	Puffergröße; bei Verwendung der Socket Schnittstelle hat dieser Parameter keine Bedeutung.

4.1.11.2 Typen

CB_DEVICE_RECEIVER:	Standard-Empfänger (passiv)
CB_DEVICE_RECEIVER_ACTIVE:	Aktiver Empfänger
CB_DEVICE_TRANSMITTER:	Standard-Sender (aktiv)
CB_DEVICE_TRANSMITTER_PASSIVE:	Passiver Sender (im gemischten Modus nicht verfügbar)

4.1.11.3 Optionen

CB_FLAG_ACKNOWLEDGE:	Generierung einer Bestätigung bei Sendern
CB_FLAG_EXCLUSIVE:	Exklusive Pufferreservierung auf dem CAN-Controller (im gemischten Modus nicht verfügbar)
CB_FLAG_FORMAT_29_BIT:	Kennzeichnung eines 29-Bit-Identifiers im gemischten Modus

4.1.12 CANBOX_ACCEPTANCE_FILTER

Struktur zur Konfiguration eines Akzeptanzfilters. Ein Akzeptanzfilter ermöglicht den Empfang einer Gruppe von Identifiern. Es werden alle Identifier empfangen, die das Kriterium

$(ID \& \text{Filter.ulMask}) == (\text{Filter.ulId} \& \text{Filter.ulMask})$ erfüllen. Sollen beispielsweise die Identifier 0xA (10dez) und 0xB (11dez) empfangen werden, so ist für den Akzeptanzfilter $\text{ulId}=0x0000000A$ und $\text{ulMask}=0x0000000E$ zu setzen.

4.1.12.1 Parameter

ULONG	ulId:	Identifier des Filters. Im gemischten Modus kann das höchstwertige Bit zur Kennzeichnung eines 29-Bit-Identifiers gesetzt werden. Dieses Bit ist dann auch in den zurückgelieferten Daten gesetzt.
ULONG	ulMask:	Bit-Maske des Filters
USHORT	usType:	Typ des Filters
USHORT	usFlags:	Optionen für den Filter
USHORT	usReserved:	Reservierter Parameter, der bei Verwendung der Socket Schnittstelle auf den nullbasierten Index der zugehörigen Schnittstelle gesetzt werden muss.
USHORT	usBufferSize:	Puffergröße; bei Verwendung der Socket-Schnittstelle hat dieser Parameter keine Bedeutung.

4.1.12.2 Typen

CB_FILTER_DATA: Akzeptanzfilter für Daten Frames
CB_FILTER_REMOTE: Akzeptanzfilter für Remote Frames (im gemischten Modus nicht verfügbar)

4.1.12.3 Optionen

CB_FLAG_FORMAT_29_BIT: Kennzeichnung eines Filters für 29-Bit-Identifizier im gemischten Modus

4.1.13 CANBOX_ROUTER

Struktur zur Konfiguration eines Routers. Ein Router ermöglicht eine Verbindung zweier CANboxen über LAN bzw. WLAN. Eine Gruppe von Identifiern (siehe Akzeptanzfilter) kann empfangen und über LAN/WLAN an die andere CANbox[®] versendet werden. Von der anderen CANbox[®] über LAN/WLAN empfangene Daten können auf dem CAN Bus versendet werden.

4.1.13.1 Parameter

USHORT	usIndex:	Reservierter Parameter, der zu Eins gesetzt werden muss.
USHORT	usType:	Typ des Routers
USHORT	usMode:	Modus des Routers
USHORT	usFlags:	Optionen für den Router
ULONG	ulServerIp:	IP-Adresse der Server CANbox [®]
USHORT	usServerPort:	Port-Adresse der Server CANbox [®]
USHORT	usReserved:	Reservierter Parameter, der bei Verwendung der Socket Schnittstelle auf den nullbasierten Index der zugehörigen Schnittstelle gesetzt werden muss.
ULONG	ulCanId:	Identifizier des CAN-Akzeptanz-Filters. (siehe 3.8) (0xFFFFFFFF: es werden keine Daten am lokalen CAN-Bus empfangen)
ULONG	ulCanMask:	Bit-Maske des CAN-Akzeptanz-Filters. (siehe 3.8)

4.1.13.2 Typen

CB_ROUTER_LAN_SERVER: Verbindung über LAN als TCP/IP - Server
CB_ROUTER_LAN_CLIENT: Verbindung über LAN als TCP/IP - Client
CB_ROUTER_WLAN_SERVER: Verbindung über WLAN als TCP/IP - Server
CB_ROUTER_WLAN_CLIENT: Verbindung über WLAN als TCP/IP - Client

4.1.13.3 Modus

CB_ROUTER_INACTIVE: Router ist nicht aktiv

CB_ROUTER_ACTIVE: Router ist aktiv

4.1.13.4 Optionen

CB_FLAG_FORMAT_29_BIT: Kennzeichnung eines Akzeptanz-Filters für 29-Bit-Identifizier im gemischten Modus

4.1.14 CANBOX_ROUTER_EX

Struktur zur Konfiguration eines Routers mit Latenzzeit-Berücksichtigung. Ein Router ermöglicht eine Verbindung zweier CANboxen über LAN bzw. WLAN. Eine Gruppe von Identifiern (siehe Akzeptanzfilter) kann empfangen und über LAN/WLAN an die andere CANbox[®] versendet werden. Von der anderen CANbox[®] über LAN/WLAN empfangene Daten können auf dem CAN Bus versendet werden.

4.1.14.1 Parameter

USHORT	usIndex:	Reservierter Parameter, der zu Eins gesetzt werden muss.
USHORT	usType:	Typ des Routers
USHORT	usMode:	Modus des Routers
USHORT	usFlags:	Optionen für den Router
ULONG	ulServerIp:	IP-Adresse der Server-CANbox [®]
USHORT	usServerPort:	Port-Adresse der Server-CANbox [®]
USHORT	usReserved:	Reservierter Parameter, der bei Verwendung der Socket Schnittstelle auf den nullbasierten Index der zugehörigen Schnittstelle gesetzt werden muss.
ULONG	ulCanId:	Identifizier des CAN-Akzeptanz-Filters. (0xFFFFFFFF: es werden keine Daten am lokalen CAN-Bus empfangen)
ULONG	ulCanMask:	Bit-Maske des CAN-Akzeptanz-Filters.
ULONG	ulLatency:	Maximale Latenzzeit, bis zu der die Daten auf den CAN-Bus gestellt werden. (5- 10.000 ms)
USHORT	usCollect:	Anzahl der Botschaften, die vor TCP – Übertragung gesammelt werden
USHORT	usCollectTimeout:	Zeitspanne, nach der Botschaften spätestens via TCP übertragen werden

4.1.14.2 Typen

CB_ROUTER_LAN_SERVER:	Verbindung über LAN als TCP/IP-Server
CB_ROUTER_LAN_CLIENT:	Verbindung über LAN als TCP/IP-Client
CB_ROUTER_WLAN_SERVER:	Verbindung über WLAN als TCP/IP-Server
CB_ROUTER_WLAN_CLIENT:	Verbindung über WLAN als TCP/IP-Client
CB_FORWARDER_LAN_SERVER:	Verbindung über LAN als Server (erw. Modus)
CB_FORWARDER_LAN_CLIENT:	Verbindung über LAN als Client (erw. Modus)
CB_FORWARDER_WLAN_SERVER:	Verbindung über WLAN als Server (erw. Modus)
CB_FORWARDER_WLAN_CLIENT:	Verbindung über WLAN als Client (erw. Modus)

4.1.14.3 Modus

CB_ROUTER_INACTIVE:	Router ist nicht aktiv
CB_ROUTER_ACTIVE:	Router ist aktiv
CB_FORWARDER_ACTIVE:	Router ist aktiv (erw. Modus)

4.1.14.4 Optionen

CB_FLAG_ROUTER_SYNC_MODE:	Aktivierung der Latenzzeit-Berücksichtigung
CB_FLAG_FORMAT_11_BIT:	Kennzeichnung eines Akzeptanz-Filters für 11-Bit-Identifizier im gemischten Modus
CB_FLAG_FORMAT_29_BIT:	Kennzeichnung eines Akzeptanz-Filters für 29-Bit-Identifizier im gemischten Modus
CB_FLAG_FORMAT_MIXED:	Es werden intern Akzeptanzfilter für 11- und 29-Bit-Identifizier angelegt

4.1.15 CANBOX_LAN_CONFIG

Struktur zur Abfrage bzw. Einstellung der LAN-Parameter der CANbox[®].

4.1.15.1 Parameter

ULONG	ullp:	IP-Adresse der CANbox [®]
ULONG	ulSubnetmask:	Subnetzmaske der CANbox [®]

4.1.16 CANBOX_WLAN_CONFIG

Struktur zur Abfrage bzw. Einstellung der WLAN Parameter der CANbox[®].

4.1.16.1 Parameter

ULONG	ullp:	IP-Adresse der CANbox [®]
ULONG	ulSubnetmask:	Subnetzmaske der CANbox [®]
UCHAR[34]	ucSSID:	Netzwerkname mit bis zu 34 Zeichen
USHORT	usMode:	AdHoc Verbindung oder Verbindung über Accesspoint
USHORT	usOwnChannel:	WLAN-Kanal von 1 bis 14
USHORT	usAuthentication:	Netzwerkauthentifizierung <i>Offen</i> oder <i>Gemeinsam</i> verwendet
USHORT	usEncryption:	Datenverschlüsselung <i>Deaktiviert</i> oder <i>WEP</i>
USHORT	usKeyId:	Null-basierter Index des verwendeten Schlüssels
UCHAR[16]	ucKey0:	Schlüssel 0 (usKeyId=0)
UCHAR[16]	ucKey1:	Schlüssel 1 (usKeyId=1)
UCHAR[16]	ucKey2:	Schlüssel 2 (usKeyId=2)
UCHAR[16]	ucKey3:	Schlüssel 3 (usKeyId=3)

4.1.16.2 Modi

CB_WLAN_ACCESS_POINT	Verbindung über Accesspoint
CB_WLAN_AD_HOC	AdHoc Verbindung

4.1.16.3 Netzwerkauthentifizierung

CB_WLAN_OPEN_SYSTEM	Offen
CB_WLAN_SHARED_KEY	Gemeinsam verwendet

4.1.16.4 Datenverschlüsselung

CB_WLAN_ENCRYPTION_OFF	Deaktiviert
CB_WLAN_ENCRYPTION_WEP	WEP

4.1.17 CANBOX_INTERFACE_INFO

Struktur zur Abfrage des Zustands eines CAN Interface

4.1.17.1 Parameter

CANBOX_INTERFACE_STATE sState: Status des CAN Bus

USHORT	usValid:	wenn 0 unterstützt die CANbox® die weiteren Datenfelder nicht
ULONG	ulBufferSizeMax:	Puffergröße des Interface
USHORT	usIndex:	Index der Schnittstelle (1 oder 2)
USHORT	usBitRate:	Bitrate
USHORT	usShutOff:	1 = Abschlusswiderstand eingeschaltet
USHORT	usFormat:	Datenformat (11, 29 oder mixed)
USHORT	usTq:	Time Quantum in Nanosekunden
USHORT	usTseg1:	Zeit vor dem nominellen Abtastzeitpunkt
USHORT	usTseg2:	Zeit nach dem nominellen Abtastzeitpunkt
USHORT	usTsjw:	Synchronisationssprungweite
USHORT	usSpB:	Zahl der Abtastungen pro Bit

4.1.18 CANBOX_IDENTIFIER_DATA

Struktur mit Identifier Daten

4.1.18.1 Parameter

CANBOX_TIME_STAMP	sTime:	Zeitstempel bei Telegramm-Empfang
ULONG	ulld:	CAN Id des Identifiers. Im gemischten Modus ist das höchstwertige Bit zur Kennzeichnung eines 29-Bit-Identifiers gesetzt, falls dies beim Öffnen gesetzt wurde.
ULONG	ulDataSize:	Anzahl gültiger Datenbytes
UCHAR	ucData[8] (ucData0Datenbytes bis ucData7 bei VB):	

4.1.19 CANBOX_UNIVERSAL_SENDER_DATA

Struktur zum Versenden einer CAN-Botschaft über ein allgemeines Sende-Objektes. Mit einem allgemeinen Sende-Objekt können beliebige Identifier mittels ID Angabe versendet werden.

4.1.19.1 Parameter

ULONG	ulld:	CAN- Identifier. Im gemischten Modus kann das höchstwertige Bit zur Kennzeichnung eines 29-Bit-Identifiers gesetzt werden.
USHORT	usFlags:	Optionen
ULONG	ulDataSize:	Anzahl der zu versendenden Nutzdaten
UCHAR	ucData[8] (ucData0Nutzdaten bis ucData7 bei VB):	

4.1.19.2 Optionen

CB_FLAG_FORMAT_29_BIT:	Kennzeichnung eines 29-Bit-Identifiers im gemischten Modus
------------------------	------------------------------------------------------------

4.1.20 CANBOX_16

Hilfs-Struktur zur Verwendung der CANbox[®] unter Visual Basic.

4.1.20.1 Parameter

UCHAR	ucLoByte:	Niederwertiges Byte
UCHAR	ucHiByte:	Höherwertiges Byte

4.1.21 CANBOX_32

Hilfs-Struktur zur Verwendung der CANbox[®] unter Visual Basic.

4.1.21.1 Parameter

UCHAR	ucLoByte:	Niederwertigstes Byte
UCHAR	ucByte1:	Höherwertiges Byte des LoWord
UCHAR	ucByte2:	Niederwertiges Byte des HiWord
UCHAR	ucHiByte:	Höherwertigstes Byte

4.2 Fehlercodes

CB_ERROR_OK: Kein Fehler.

4.2.1 Allgemeine Fehlermeldungen der CANbox[®]

CB_ERROR_SIZE_INPUT: Größe der Eingangsdaten ungültig.
CB_ERROR_SIZE_OUTPUT: Größe der Ausgangsdaten ungültig.
CB_ERROR_DATA: Übergebene Daten fehlerhaft.
CB_ERROR_STATE: Ungültiger Status.
CB_ERROR_MEMORY: Fehler beim Zugriff auf den RAM-Speicher.
CB_ERROR_FLASH: Fehler beim Zugriff auf den Flash-Speicher.
CB_ERROR_EEPROM: Fehler beim EEPROM – Zugriff.
CB_ERROR_NOT_SUPPORTED: Funktion wird nicht unterstützt (Versions-Kontrolle).

4.2.2 Fehlermeldungen bezüglich der Initialisierung

CB_ERROR_CAN_INIT: Fehler bei der Initialisierung der CAN-Firmware.
CB_ERROR_HARDWARE_INIT: Fehler bei der Initialisierung der CPU.
CB_ERROR_MEMORY_INIT: Fehler bei der Speicher-Reservierung.

4.2.3 Fehlermeldungen bezüglich Freigaben

CB_ERROR_CAN_EXIT: Fehler bei der Freigabe des CAN-MDDs.

4.2.4 Fehlermeldungen bezüglich CAN-Schnittstellen

CB_ERROR_INTERFACE_INDEX: Ungültiger Schnittstellen-Index.
CB_ERROR_INTERFACE_STATE: Ungültiger Schnittstellen-Zustand (für die gewünschte Operation).
CB_ERROR_INTERFACE_OPEN: Fehler beim Öffnen der Schnittstelle.
CB_ERROR_INTERFACE_OPEN_BITRATE: Die Schnittstelle wurde bereits mit einer anderen Bitrate geöffnet.
CB_ERROR_INTERFACE_OPEN_FORMAT: Die Schnittstelle wurde bereits mit einem anderen Format (11-Bit/29-Bit) geöffnet.
CB_ERROR_INTERFACE_BITRATE: Nicht unterstützte Bitrate.
CB_ERROR_INTERFACE_SHUT_OFF: Ungültiger Wert für den Abschlusswiderstand.
CB_ERROR_INTERFACE_FORMAT: Ungültiger Wert für das CAN-Format.
CB_ERROR_INTERFACE_INIT: Fehler bei der Initialisierung einer CAN – Schnittstelle.
CB_ERROR_INTERFACE_CONTROL: Fehler bei der Steuerung CAN – Schnittstelle.
CB_ERROR_INTERFACE_EXIT: Fehler bei der Freigabe CAN – Schnittstelle.

4.2.5 Fehlermeldungen bezüglich CAN – Identifiern

CB_ERROR_IDENTIFIER_INIT:	Fehler bei der Initialisierung eines CAN – Identifiers.
CB_ERROR_IDENTIFIER_TYPE:	Ungültiger Identifier – Typ.
CB_ERROR_IDENTIFIER_EXCLUSIVE:	Ungültiger Wert für das Exklusiv-Flag.
CB_ERROR_IDENTIFIER_READ:	Fehler beim Lesen der Daten.
CB_ERROR_IDENTIFIER_WRITE:	Fehler beim Schreiben der Daten.

4.2.6 Fehlermeldungen bezüglich Akzeptanz-Filtern

CB_ERROR_FILTER_INIT:	Fehler bei der Initialisierung eines Filters.
CB_ERROR_FILTER_TYPE:	Ungültiger Filter-Typ.
CB_ERROR_FILTER_SET:	Fehler beim Setzen des Filters.

4.2.7 Fehlermeldungen bezüglich Universal-Sendern

CB_ERROR_SENDER_INIT:	Fehler bei der Initialisierung eines Universal-Senders.
CB_ERROR_SENDER_WRITE:	Fehler beim Schreiben eines Universal-Senders.

4.2.8 Allgemeine Fehlermeldungen der Bibliothek

CB_ERROR_NO_INIT:	Fehlende Initialisierung.
CB_ERROR_PARAMETER:	Falscher Parameter.
CB_ERROR_HANDLE:	Ungültiges Handle.
CB_ERROR_HANDLE_TYPE:	Falsches Typ des Handles.

4.2.9 Fehlermeldungen der Bibliothek bezüglich der Verbindung zur CANbox®

CB_ERROR_CONNECTION_INIT:	Verbindung konnte nicht aufgebaut werden.
CB_ERROR_CONNECTION_IN_USE:	CANbox® wird bereits anderweitig angesprochen.
CB_ERROR_CONNECTION_TIMEOUT:	Timeout während des Zugriffs auf die CANbox®.
CB_ERROR_CONNECTION_RECEIVE:	Fehler beim Daten-Empfang von der CANbox®.
CB_ERROR_CONNECTION_SEND:	Fehler beim Daten-Senden zur CANbox®.
CB_ERROR_CONNECTION_EXIT:	Verbindung konnte nicht abgebaut werden.
CB_ERROR_DEVICE_NOT_INSTALLED:	Die angesprochene CANbox® ist in der Systemsteuerung nicht installiert.

4.3 Allgemeine Funktionen

4.3.1 canbox_init_lib

Initialisierung der Bibliothek. Diese Funktion muss einmalig vor allen anderen Funktion aufgerufen werden. Andernfalls liefern alle Funktionen den Fehler CB_NO_INIT.

4.3.1.1 Parameter

Keine.

4.3.2 canbox_get_driver_version

Liefert Informationen über den verwendeten Treiber.

4.3.2.1 Parameter

CANBOX_VERSION_INFO* psInfo Zeiger auf Struktur zur Aufnahme der Versions-Informationen

4.3.3 canbox_get_error_message

Liefert den Fehlertext zur einer Fehlernummer.

4.3.3.1 Parameter

CANBOX_ERROR Error: Fehler-Code
USHORT* pusSize: Größe des Zeichen-Puffers
char* szMessage: Zeichen-Puffer zur Aufnahme der Fehlerbeschreibung

4.3.4 canbox_exit_lib

Freigabe der verwendeten Ressourcen. Diese Funktion ist als letzte Funktion aufzurufen.

4.3.4.1 Parameter

Keine

4.4 Device- Funktionen

4.4.1 canbox_scan_devices

Liefert Informationen über die installierten CANbox[®]en.

Wird der Parameter pasInfo == NULL übergeben, wird in *pusCount die Anzahl der installierten CANbox[®]en eingetragen.

4.4.1.1 Parameter

USHORT*	pusCount:	Größe des zur Verfügung gestellten Puffers zur Aufnahme der Informationen. Hier wird anschließend die Anzahl der zurückgegebenen Info-Strukturen eingetragen.
CANBOX_SCAN_INFO*	pasInfo:	Struktur-Feld zur Aufnahme der Device Informationen

4.4.2 canbox_get_device_parameter

Liefert Informationen über die Verbindungskonfiguration der gewählter CANbox[®]. Im Puffer werden bei TCP/IP Verbindungen die IP-Adresse, bei Seriellen Verbindungen der Name der Schnittstelle gespeichert.

DWORD	dwId:	Id der gewählten CANbox [®]
char*	szText:	Puffer zur Aufnahme der Information
DWORD	dwSize:	Größe des Puffers in szText

4.4.3 canbox_open_device

Öffnen des Devices mit der übergebenen Id. Aufbau der Verbindung zum Device.

4.4.3.1 Parameter

USHORT	usId:	Id des zu öffnenden Devices
CANBOX_HANDLE*	phDevice:	Zeiger auf Handle zur Aufnahme des Device Handles

4.4.4 canbox_reset_device

Zurücksetzen des Devices. Alle auf dem Device geöffneten Identifier werden geschlossen und die Bus-Kommunikation wird beendet. Die Empfangspuffer werden geleert. Der Zeitstempel wird zurückgesetzt. Die Verbindung bleibt bestehen, das Handle behält seine Gültigkeit. Aktive Router starten nach dem Reset neu.

4.4.4.1 Parameter

CANBOX_HANDLE hDevice: Bei canbox_open_device erhaltenes Handle des Devices

4.4.5 canbox_reset_device_ex

Zurücksetzen des Devices. Alle auf dem Device geöffneten Identifier werden geschlossen und die Bus-Kommunikation wird beendet. Die Empfangspuffer werden geleert. Der Zeitstempel wird zurückgesetzt. Die Verbindung bleibt bestehen, das Handle behält seine Gültigkeit. Zusätzlich kann festgelegt werden, ob aktive Router nach dem Reset neu starten sollen.

4.4.5.1 Parameter

CANBOX_HANDLE hDevice: Bei canbox_open_device erhaltenes Handle des Devices

CANBOX_RESET_EX sResetEx: Struktur zur Parametrierung des Reset-Verhaltens

4.4.6 canbox_get_device_info

Liefert Informationen über das Device.

4.4.6.1 Parameter

CANBOX_HANDLE hDevice: Bei canbox_open_device erhaltenes Handle des Devices

CANBOX_DEVICE_INFO* psInfo: Zeiger auf Info-Struktur zur Aufnahme der Informationen

4.4.7 canbox_reset_time_stamp

Zurücksetzen des Zeitstempels auf den Wert 0. Der Zeitstempel hat eine Auflösung von 0,1µs.

4.4.7.1 Parameter

CANBOX_HANDLE hDevice: Bei canbox_open_device erhaltenes Handle des Devices

4.4.8 canbox_set_time_stamp

Setzen des Zeitstempels auf den übergebenen Wert. Der Zeitstempel hat eine Auflösung von 0,1µs.

4.4.8.1 Parameter

CANBOX_HANDLE hDevice: Bei canbox_open_device erhaltenes Handle des Devices

CANBOX_TIME_STAMP sTime: Zu setzender Wert.

4.4.9 canbox_set_time_stamp_by_ref

Setzen des Zeitstempels auf den übergebenen Wert. Der Zeitstempel hat eine Auflösung von 0,1µs.

4.4.9.1 Parameter

CANBOX_HANDLE hDevice: Bei canbox_open_device erhaltenes Handle des Devices

CANBOX_TIME_STAMP* psTime: Zeiger auf Struktur mit zu setzendem Wert

4.4.10 canbox_close_device

Beendet die Kommunikation mit dem Device.

4.4.10.1 Parameter

CANBOX_HANDLE hDevice: Bei canbox_open_device erhaltenes Handle des Devices

4.5 Schnittstellen- Funktionen

4.5.1 canbox_get_interface_info

4.5.1.1 Parameter

CANBOX_HANDLE hDevice: Bei canbox_open_device erhaltenes Handle des Devices

USHORT usIndex Index des Interface (1 oder 2)

CANBOX_INTERFACE_INFO* pInfo: Zeiger auf Struktur zur Aufnahme der Informationen

4.5.2 canbox_open_interface

Öffnet eine CAN-Schnittstelle und parametriert sie anhand der übergebenen Werte. Die Bit-Rate wird übergeben; die Bit-Timing-Parameter werden daraus bestimmt.

4.5.2.1 Parameter

CANBOX_HANDLE	hDevice:	Bei canbox_open_device erhaltenes Handle des Devices
CANBOX_INTERFACE	slInterface:	Struktur zur Parametrierung der Schnittstelle
CANBOX_HANDLE*	phInterface:	Zeiger auf Handle zur Aufnahme des Schnittstellen Handles

4.5.3 canbox_open_interface_by_ref

Öffnet eine CAN-Schnittstelle und parametriert sie anhand der übergebenen Werte. Die Bit-Rate wird übergeben; die Bit-Timing-Parameter werden daraus bestimmt.

4.5.3.1 Parameter

CANBOX_HANDLE	hDevice:	Bei canbox_open_device erhaltenes Handle des Devices
CANBOX_INTERFACE*	psInterface:	Zeiger auf eine Struktur zur Parametrierung der Schnittstelle
CANBOX_HANDLE*	phInterface:	Zeiger auf Handle zur Aufnahme des Schnittstellen Handles

4.5.4 canbox_open_interface_ex

Öffnet eine CAN-Schnittstelle und parametriert sie anhand der übergebenen Werte. Die Bit-Timing-Parameter werden direkt übergeben.

4.5.4.1 Parameter

CANBOX_HANDLE	hDevice:	Bei canbox_open_device erhaltenes Handle des Devices
CANBOX_INTERFACE_EX	slInterface:	Struktur zur Parametrierung der Schnittstelle
CANBOX_HANDLE*	phInterface:	Zeiger auf Handle zur Aufnahme des Schnittstellen Handles

4.5.5 canbox_open_interface_ex_by_ref

Öffnet eine CAN-Schnittstelle und parametriert sie anhand der übergebenen Werte. Die Bit-Timing-Parameter werden direkt übergeben.

4.5.5.1 Parameter

CANBOX_HANDLE	hDevice:	Bei canbox_open_device erhaltenes Handle des Devices
CANBOX_INTERFACE_EX*	psInterface :	Zeiger auf eine Struktur zur Parametrierung der Schnittstelle
CANBOX_HANDLE*	phInterface :	Zeiger auf Handle zur Aufnahme des Schnittstellen Handles

4.5.6 canbox_get_interface_state

Liefert den Status einer CAN-Schnittstelle.

4.5.6.1 Parameter

CANBOX_HANDLE	hInterface:	Bei canbox_open_interface bzw. canbox_open_interface_ex erhaltenes Handle der Schnittstelle
CANBOX_INTERFACE_STATE*	psState:	Zeiger auf Struktur zur Aufnahme des Status

4.5.7 canbox_get_last_interface_time

Liefert den letzten Zeitpunkt an dem eine Botschaft von der Schnittstelle empfangen wurde.

4.5.7.1 Parameter

CANBOX_HANDLE	hInterface:	Bei canbox_open_interface bzw. canbox_open_interface_ex erhaltenes Handle der Schnittstelle
CANBOX_TIME_STAMP*	psTime:	Zeiger auf Struktur zur Aufnahme des Zeitstempels

4.5.8 canbox_clear_interface

Leert den Datenpuffer der Schnittstelle.

4.5.8.1 Parameter

CANBOX_HANDLE hInterface: Bei canbox_open_interface bzw. canbox_open_interface_ex erhaltenes Handle der Schnittstelle

4.5.9 canbox_start_interface

Startet eine gestoppte CAN-Schnittstelle.

4.5.9.1 Parameter

CANBOX_HANDLE hInterface: Bei canbox_open_interface bzw. canbox_open_interface_ex erhaltenes Handle der Schnittstelle

4.5.10 canbox_read_interface

Liest einen Datensatz (empfangene Botschaft) aus dem Datenpuffer der CAN-Schnittstelle.

4.5.10.1 Parameter

CANBOX_HANDLE hInterface: Bei canbox_open_interface bzw. canbox_open_interface_ex erhaltenes Handle der Schnittstelle

CANBOX_INTERFACE_READ* psData: Zeiger auf Struktur zur Aufnahme des Datensatzes

4.5.11 canbox_read_interface_ex

Liest mehrere Datensätze (empfangene Botschaften) aus dem Datenpuffer der CAN-Schnittstelle.

4.5.11.1 Parameter

CANBOX_HANDLE hInterface: Bei canbox_open_interface bzw. canbox_open_interface_ex erhaltenes Handle der Schnittstelle

USHORT* pusCount: Anzahl gewünschter bzw. gelieferter Datensätze

CANBOX_INTERFACE_READ* pasData: Zeiger auf Struktur zur Aufnahme der Datensätze

4.5.12 canbox_write_interface

Stellt die übergebenen Daten des übergebenen Identifiers auf den CAN-Bus.

4.5.12.1 Parameter

CANBOX_HANDLE	hInterface:	Bei canbox_open_interface bzw. canbox_open_interface_ex erhaltenes Handle der Schnittstelle
CANBOX_INTERFACE_WRITE*	psData:	Zeiger auf Struktur mit Informationen über den zu sendenden Identifier und die zu schreibenden Daten

4.5.13 canbox_write_interface_ex

Stellt die übergebenen Daten der übergebenen Identifier auf den CAN-Bus.

4.5.13.1 Parameter

CANBOX_HANDLE	hInterface:	Bei canbox_open_interface bzw. canbox_open_interface_ex erhaltenes Handle der Schnittstelle
USHORT*	pusCount:	Anzahl zu sendender Datensätze
CANBOX_INTERFACE_WRITE*	pasData:	Zeiger auf Struktur-Array mit Informationen über die zu sendenden Identifier und die zu schreibenden Daten

4.5.14 canbox_stop_interface

Stoppt eine gestartete CAN-Schnittstelle. Die CAN-Schnittstelle nimmt nicht mehr am Bus-Verkehr teil.

4.5.14.1 Parameter

CANBOX_HANDLE	hInterface:	Bei canbox_open_interface bzw. canbox_open_interface_ex erhaltenes Handle der Schnittstelle
---------------	-------------	---------------------------------------------------------------------------------------------

4.5.15 canbox_close_interface

Schließt eine geöffnete CAN-Schnittstelle. Die CAN-Schnittstelle nimmt nicht mehr am Bus-Verkehr teil. Alle Identifier, Akzeptanzfilter und Universal-Sender der Schnittstelle werden ebenfalls geschlossen.

4.5.15.1 Parameter

CANBOX_HANDLE hInterface: Bei canbox_open_interface bzw. canbox_open_interface_ex erhaltenes Handle der Schnittstelle

4.6 Identifier Funktionen

4.6.1 canbox_open_identifier

Öffnet einen CAN Identifier und parametriert ihn anhand der übergebenen Werte.

4.6.1.1 Parameter

CANBOX_HANDLE	hInterface:	Bei canbox_open_interface bzw. canbox_open_interface_ex erhaltenes Handle der Schnittstelle
CANBOX_IDENTIFIER	sIdentifier:	Struktur zur Parametrierung des Identifiers
CANBOX_HANDLE*	phIdentifier	Zeiger auf Handle zur Aufnahme des Identifier - Handles

4.6.2 canbox_open_identifier_by_ref

Öffnet einen CAN Identifier und parametriert ihn anhand der übergebenen Werte.

4.6.2.1 Parameter

CANBOX_HANDLE	hInterface:	Bei canbox_open_interface bzw. canbox_open_interface_ex erhaltenes Handle der Schnittstelle
CANBOX_IDENTIFIER*	psIdentifier	Zeiger auf eine Struktur zur Parametrierung des Identifiers
CANBOX_HANDLE*	phIdentifier	Zeiger auf Handle zur Aufnahme des Identifier – Handles

4.6.3 canbox_read_identifier

Liest Daten eines CAN Identifiers.

4.6.3.1 Parameter

CANBOX_HANDLE	hIdentifier:	Bei canbox_open_identifier erhaltenes Handle des Identifiers
CANBOX_IDENTIFIER_DATA*	psData	Zeiger auf Struktur zur Aufnahme der Daten

4.6.4 canbox_write_identifier

Setzt Daten eines CAN Identifiers.

4.6.4.1 Parameter

CANBOX_HANDLE	hIdentifier:	Bei canbox_open_identifier erhaltenes Handle des Identifiers
CANBOX_IDENTIFIER_DATA*	psData	Zeiger auf Struktur mit Informationen über die zu sendenden Daten

4.6.5 canbox_close_identifier

Schließt einen geöffneten CAN Identifier.

4.6.5.1 Parameter

CANBOX_HANDLE	hIdentifier:	Bei canbox_open_identifier erhaltenes Handle des Identifiers
---------------	--------------	--------------------------------------------------------------

4.7 Akzeptanz-Filter-Funktionen

4.7.1 canbox_open_acceptance_filter

Öffnet einen Akzeptanz-Filter und parametrier ihn anhand der übergebenen Werte.

4.7.1.1 Parameter

CANBOX_HANDLE	hInterface	Bei canbox_open_interface bzw. canbox_open_interface_ex erhaltenes Handle der Schnittstelle
CANBOX_ACCEPTANCE_FILTER	sFilter	Struktur zur Parametrierung des Filters.
CANBOX_HANDLE*	phFilter	Zeiger auf Handle zur Aufnahme des Filterhandles

4.7.2 canbox_open_acceptance_filter_by_ref

Öffnet einen Akzeptanz-Filter und parametrier ihn anhand der übergebenen Werte.

4.7.2.1 Parameter

CANBOX_HANDLE	hInterface	Bei canbox_open_interface bzw. canbox_open_interface_ex erhaltenes Handle der Schnittstelle
CANBOX_ACCEPTANCE_FILTER*	psFilter	Zeiger auf eine Struktur zur Parametrierung des Filters
CANBOX_HANDLE*	phFilter	Zeiger auf Handle zur Aufnahme des Filterhandles

4.7.3 canbox_read_acceptance_filter

Liest Daten eines Akzeptanz-Filters.

4.7.3.1 Parameter

CANBOX_HANDLE	hFilter	Bei canbox_open_acceptance_filter erhaltenes Handle des Filters
CANBOX_IDENTIFIER_DATA*	psData	Zeiger auf Struktur zur Aufnahme der Daten

4.7.4 canbox_close_acceptance_filter

Schließt einen geöffneten Akzeptanz-Filter.

4.7.4.1 Parameter

CANBOX_HANDLE	hFilter	Bei canbox_open_acceptance_filter erhaltenes Handle des Filters
---------------	---------	-----------------------------------------------------------------

4.8 Universal-Sender-Funktionen

4.8.1 canbox_open_universal_sender

Öffnet einen Universal-Sender. Dabei handelt es sich um ein Sende-Objekt, mit dem durch Angabe der jeweiligen Id beliebige CAN-Botschaften versendet werden können.

4.8.1.1 Parameter

CANBOX_HANDLE	hInterface:	Bei canbox_open_interface bzw. canbox_open_interface_ex erhaltenes Handle der Schnittstelle
CANBOX_HANDLE*	phSender	Zeiger auf Handle zur Aufnahme des Universal Sender Handles

4.8.2 canbox_write_universal_sender

Sendet eine Botschaft mittels eines Universal-Senders.

4.8.2.1 Parameter

CANBOX_HANDLE	hSender:	Bei canbox_open_universal_sender erhaltenes Handle des Universal-Senders
CANBOX_UNIVERSAL_SENDER_DATA *	psData	Zeiger auf eine Struktur zur Übergabe der zu sendenden Daten

4.8.3 canbox_write_universal_sender_ex

Sendet eine oder mehrere Botschaft mittels eines Universal-Senders. Sollten nicht alle Botschaften direkt gesendet werden können, werden diese in einem Puffer gespeichert und versendet, sobald der CAN-Bus dies zulässt.

4.8.3.1 Parameter

CANBOX_HANDLE	hSender:	Bei canbox_open_universal_sender erhaltenes Handle des Universal-Senders
USHORT*	pusCoun	Anzahl zu sendender Datensätze t
CANBOX_UNIVERSAL_SENDER_DATA *	pasData	Zeiger auf eine Struktur zur Übergabe der zu sendenden Daten

4.8.4 canbox_close_universal_sender

Schließt einen geöffneten Universal-Sender.

4.8.4.1 Parameter

CANBOX_HANDLE	hSender	Bei canbox_open_universal_sender erhaltenes Handle des Universal-Senders
---------------	---------	--------------------------------------------------------------------------

4.9 Router Funktionen

4.9.1 canbox_read_router_config

Liefert die aktuelle Parametrierung des Router der Schnittstelle.

4.9.1.1 Parameter

CANBOX_HANDLE	hInterface	Bei canbox_open_interface bzw. canbox_open_interface_ex erhaltenes Handle der Schnittstelle
CANBOX_ROUTER*	psRouter	Zeiger auf Struktur zur Aufnahme der Parametrierung des Routers

4.9.2 canbox_read_router_config_ex

Liefert die aktuelle Parametrierung des Router der Schnittstelle.

4.9.2.1 Parameter

CANBOX_HANDLE	hInterface	Bei canbox_open_interface bzw. canbox_open_interface_ex erhaltenes Handle der Schnittstelle
CANBOX_ROUTER_EX*	psRouter	Zeiger auf Struktur zur Aufnahme der Parametrierung des Routers

4.9.3 canbox_write_router_config

Parametriert den Router anhand der übergebenen Werte.

4.9.3.1 Parameter

CANBOX_HANDLE	hInterface	Bei canbox_open_interface bzw. canbox_open_interface_ex erhaltenes Handle der Schnittstelle
CANBOX_ROUTER	sRouter	Struktur zur Parametrierung des Routers

4.9.4 canbox_write_router_config_ex

Parametriert den Router anhand der übergebenen Werte.

4.9.4.1 Parameter

CANBOX_HANDLE	hInterface	Bei canbox_open_interface bzw. canbox_open_interface_ex erhaltenes Handle der Schnittstelle
CANBOX_ROUTER_EX	sRouter	Struktur zur Parametrierung des Routers

4.10 Erweiterte Router Funktion (Forwarder)

4.10.1 canbox_read_forwarder_config

4.10.1.1 Parameter

CANBOX_HANDLE	hInterface	Bei canbox_open_interface bzw. canbox_open_interface_ex erhaltenes Handle der Schnittstelle
CANBOX_ROUTER_EX*	psRouter	Zeiger auf Struktur zur Aufnahme der Parametrierung des Routers; das Element usIndex muss vor dem Aufruf auf den Index (0..9) des gewünschten Parametersatzes gesetzt werden

4.10.2 canbox_write_forwarder_config

4.10.2.1

Parameter

CANBOX_HANDLE	hInterface	Bei canbox_open_interface bzw. canbox_open_interface_ex erhaltenes Handle der Schnittstelle
CANBOX_ROUTER_EX	sRouter	Struktur zur Parametrierung des Routers; das Element usIndex muss vor dem Aufruf auf den Index (0..9) des gewünschten Parametersatzes gesetzt werden

4.11 Hilfsfunktionen für Visual Basic

Da Visual Basic für einige Strukturen nicht die notwendige Ausrichtung zulässt, wurden die Hilfsstrukturen CANBOX_16 und CANBOX_32 eingeführt. Diese ersetzen in einigen Strukturen 16-Bit- bzw. 32-Bit-Variablen, um die korrekte Ausrichtung der Struktur sicher zu stellen. Die folgenden Funktionen dienen der Konvertierung in dieses Formats bzw. aus diesem Format.

4.11.1 canbox_convert_to_CANbox16

Konvertierung eines 16-Bit-Wertes in eine CANBOX_16-Struktur.

4.11.1.1 Parameter

USHORT	usValue	16-Bit-Wert zur Konvertierung
--------	---------	-------------------------------

4.11.2 canbox_convert_to_CANbox16_by_ref

Konvertierung eines 16-Bit-Wertes in eine CANBOX_16-Struktur.

4.11.2.1 Parameter

USHORT	usValue	16-Bit-Wert zur Konvertierung
CANBOX_16*	psValue	Zeiger auf eine CANBOX_16-Struktur für konvertierten Wert

4.11.3 canbox_convert_from_CANbox16

Konvertierung einer CANBOX_16-Struktur in einen 16-Bit-Wert.

4.11.3.1 Parameter

CANBOX_16* psValue Zeiger auf eine CANBOX_16- Struktur zur Konvertierung

4.11.4 canbox_convert_to_CANbox32

Konvertierung eines 32-Bit-Wertes in eine CANBOX_32-Struktur.

4.11.4.1 Parameter

ULONG ulValue 32-Bit-Wert zur Konvertierung

4.11.5 canbox_convert_to_CANbox32_by_ref

Konvertierung eines 16-Bit-Wertes in eine CANBOX_32-Struktur.

4.11.5.1 Parameter

ULONG ulValue 32-Bit-Wert zur Konvertierung
CANBOX_32* psValue Zeiger auf eine CANBOX_32-Struktur für konvertierten Wert

4.11.6 canbox_convert_from_CANbox32

Konvertierung einer CANBOX_32-Struktur in einen 32-Bit-Wert.

4.11.6.1 Parameter

CANBOX_32* psValue Zeiger auf eine CANBOX_32-Struktur zur Konvertierung

5 CANbox[®] – Socket Schnittstelle

Die Kommunikation mit der CANbox[®] kann statt über die Bibliothek auch durch direktes Ansprechen über die Socket Schnittstelle erfolgen. Hierzu stellt die CANbox[®] sowohl über die LAN- als auch die WLAN Schnittstelle den Port 1024 als Server-Port zur Verfügung.

Weiterhin steht der Port 1040 als Server-Port zum Zurücksetzen der Verbindung auf Port 1024 zur Verfügung. Hierzu muss lediglich eine Verbindung zu Port 1040 aufgebaut werden, wobei die IP Adressen der beiden Client-Ports identisch sein müssen.

5.1. Strukturen und Definitionen

Nachdem eine Verbindung zur CANbox[®] hergestellt wurde, erfolgt die Kommunikation mit Hilfe der Strukturen CANBOX_SOCKET_IN und CANBOX_SOCKET_OUT.

5.1.1 CANBOX_SOCKET_IN

Struktur zur Senden von Daten zur CANbox[®].

5.1.1.1 Parameter

USHORT	usMacro:	0xEE44
USHORT	usSizeIn:	Anzahl der übergebenen Bytes + 8
USHORT	usSizeOut:	Anzahl der maximal zurück erwarteten Bytes
USHORT	usTask:	0x0090
USHORT	usFunc:	Index der gewünschte Funktion (siehe unten)
UCHAR	usData[1440]	Nutzdaten

5.1.2 CANBOX_SOCKET_OUT

Struktur zum Empfang von Daten von der CANbox[®].

5.1.2.1 Parameter

USHORT	usMacro:	0x0144, falls kein Fehler aufgetreten ist
USHORT	usSizeOut/ usError:	Anzahl der tatsächlich zurückgelieferten Bytes + 6 bzw. Fehlercode im Fehlerfall
USHORT	usUnused:	Reservierter Parameter
UCHAR	usData[1440]	Nutzdaten

5.2 Funktionen

5.2.1 canbox_init (Index 4)

Initialisierung der CANbox[®]. Diese Funktion muss einmalig vor allen anderen Funktion aufgerufen werden. Andernfalls liefern alle Funktionen den Fehler CB_NO_INIT.

5.2.1.1 Parameter

Hin Keine

Zurück Keine

5.2.2 canbox_reset (Index 5)

Schließt alle geöffneten Identifier, Akzeptanz-Filter und Schnittstellen. Setzt den Zeitstempel auf 0. Leert die Software-Puffer.

5.2.2.1 Parameter

Hin CANBOX_RESET_EX

Zurück Keine

5.2.3 canbox_info (Index 6)

Liefert Informationen über das Device.

5.2.3.1 Parameter

Hin Keine

Zurück CANBOX_DEVICE_INFO Info-Struktur zur Aufnahme der Informationen

5.2.4 canbox_set_time_stamp (Index 7)

Setzt den Zeitstempel. Der Zeitstempel hat eine Auflösung von 0,1µs.

5.2.4.1 Parameter

Hin CANBOX_TIME_STAMP Gewünschter Wert des Zeitstempels

Zurück Keine

5.2.5 canbox_exit (Index 12)

Freigabe der verwendeten Ressourcen. Diese Funktion ist als letzte Funktion aufzurufen.

5.2.5.1 Parameter

Hin Keine

Zurück Keine

5.2.6 canbox_open_interface (Index 15)

Öffnet eine CAN-Schnittstelle und parametriert sie anhand der übergebenen Werte. Die Bit-Rate wird übergeben; die Bit-Timing-Parameter werden daraus bestimmt.

5.2.6.1 Parameter

Hin CANBOX_INTERFACE Parametrierung der Schnittstelle

Zurück Keine

5.2.7 canbox_open_interface_ex (Index 16)

Öffnet eine CAN-Schnittstelle und parametriert sie anhand der übergebenen Werte. Die Bit-Timing-Parameter werden direkt übergeben.

5.2.7.1 Parameter

Hin CANBOX_INTERFACE_EX Parametrierung der Schnittstelle

Zurück Keine

5.2.8 canbox_get_interface_state (Index 17)

Liefert den Status einer CAN-Schnittstelle.

5.2.8.1 Parameter

Hin USHORT Null-basierter Index der Schnittstelle

Zurück CANBOX_INTERFACE_STATE Struktur zur Aufnahme des Status

Ab Firmware 3.X auch möglich:

Zurück CANBOX_INTERFACE_INFO Struktur zur Aufnahme des erweiterten Status

5.2.9 canbox_start_interface (Index 18)

Startet eine gestoppte CAN-Schnittstelle.

5.2.9.1 Parameter

Hin USHORT Null-basierter Index der Schnittstelle
Zurück Keine

5.2.10 canbox_clear_interface (Index 19)

Leert den Datenpuffer der Schnittstelle.

5.2.10.1 Parameter

Hin USHORT Null-basierter Index der Schnittstelle
Zurück Keine

5.2.11 canbox_read_interface (Index 20)

Liest die ältesten Daten aus dem Datenpuffer der Schnittstelle.

5.2.11.1 Parameter

Hin USHORT Null-basierter Index der Schnittstelle
Zurück CANBOX_IDENTIFIER_DATA Array von Identifier Daten zur Aufnahme der Daten

5.2.12 canbox_write_interface (Index 21)

Stellt die übergebenen Daten auf den CAN-Bus. Für den Parameter ulld ist statt der Id das Handle des jeweiligen Identifiers anzugeben.

5.2.12.1 Parameter

Hin CANBOX_IDENTIFIER_DATA Array von Identifier Daten, die gesendet werden sollen.
Zurück Keine

5.2.13 canbox_stop_interface (Index 22)

Stoppt eine laufende CAN-Schnittstelle.

5.2.13.1 Parameter

Hin USHORT Null-basierter Index der Schnittstelle
Zurück Keine

5.2.14 canbox_close_interface (Index 23)

Schließt eine geöffnete CAN-Schnittstelle.

5.2.14.1 Parameter

Hin USHORT Null-basierter Index der Schnittstelle
Zurück Keine

5.2.15 canbox_open_identifizier (Index 25)

Öffnet einen CAN Identifizier und parametrieren ihn anhand der übergebenen Werte.

5.2.15.1 Parameter

Hin CANBOX_IDENTIFIER Parametrierung des Identifiziers
Zurück CANBOX_HANDLE Handle des Identifiziers für Zugriffe

5.2.16 canbox_read_identifizier_buffer (Index 26)

Liest die ältesten Daten eines CAN Identifiziers aus dem Daten-Puffer der zugehörigen Schnittstelle. Diese Funktion sollte nicht verwendet werden, wenn große Datenmengen zwischen der CANbox[®] und dem anderen Verbindungs-Teilnehmer ausgetauscht werden sollen, da diese Funktion den gesamten Daten-Puffer der zugehörigen Schnittstelle nach dem ältesten Telegramm des Identifiziers durchsucht. Stattdessen sollte dann die Funktion canbox_read_interface mit maximaler Nutzdatenmenge verwendet werden.

5.2.16.1 Parameter

Hin CANBOX_IDENTIFIER Identifizier
Zurück CANBOX_IDENTIFIER_DATA Struktur zur Aufnahme der Identifizier Daten

5.2.17 canbox_read_identifizier_actual (Index 27)

Liest die aktuellen Daten eines CAN Identifiers, als Zeit-Stempel wird der aktuelle Wert des Zeitstempels geliefert. Diese Funktion sollte nicht verwendet werden, wenn große Datenmengen zwischen der CANbox[®] und dem anderen Verbindungs-Teilnehmer ausgetauscht werden sollen, da die Effizienz aufgrund der kleinen Paketgröße sehr gering ist. Stattdessen sollte dann die Funktion canbox_read_interface mit maximaler Nutzdatenmenge verwendet werden.

5.2.17.1 Parameter

Hin CANBOX_HANDLE Handle des Identifiers
Zurück CANBOX_IDENTIFIER_DATA Struktur zur Aufnahme der Identifier-Daten

5.2.18 canbox_write_identifizier (Index 28)

Setzt Daten eines CAN Identifiers.

5.2.18.1 Parameter

Hin CANBOX_IDENTIFIER_DATA Struktur zum Setzen der Daten
Zurück Keine

5.2.19 canbox_close_identifizier (Index 29)

Schließt einen geöffneten CAN Identifier.

5.2.19.1 Parameter

Hin CANBOX_HANDLE Handle des Identifiers
Zurück Keine

5.2.20 canbox_open_filter (Index 30)

Öffnet einen CAN-Akzeptanzfilter und parametriert ihn anhand der übergebenen Werte.

5.2.20.1 Parameter

Hin CANBOX_ACCEPTANCE_FILTER Parametrierung des Akzeptanzfilters
Zurück CANBOX_HANDLE Handle des Akzeptanzfilters für Zugriffe

5.2.21 canbox_read_filter_buffer (Index 31)

Liest die ältesten Daten eines CAN-Akzeptanzfilters aus dem Daten-Puffer der zugehörigen Schnittstelle. Diese Funktion sollte nicht verwendet werden, wenn große Datenmengen zwischen der CANbox[®] und dem anderen Verbindungs-Teilnehmer ausgetauscht werden sollen, da diese Funktion den gesamten Daten-Puffer der zugehörigen Schnittstelle nach dem ältesten Telegramm des Akzeptanzfilters durchsucht. Stattdessen sollte dann die Funktion `canbox_read_interface` mit maximaler Nutzdatenmenge verwendet werden.

5.2.21.1 Parameter

Hin CANBOX_ACCEPTANCE_FILTER Akzeptanzfilter
Zurück CANBOX_IDENTIFIER_DATA Struktur zur Aufnahme der Akzeptanzfilter-Daten

5.2.22 canbox_read_filter_actual (Index 27)

Liest die zuletzt empfangenen Daten eines CAN-Akzeptanzfilters, als Zeit-Stempel wird der aktuelle Wert des Zeitstempels geliefert. Diese Funktion sollte nicht verwendet werden, wenn große Datenmengen zwischen der CANbox[®] und dem anderen Verbindungs-Teilnehmer ausgetauscht werden sollen, da die Effizienz aufgrund der kleinen Paketgröße sehr gering ist. Stattdessen sollte dann die Funktion `canbox_read_interface` mit maximaler Nutzdatenmenge verwendet werden.

5.2.22.1 Parameter

Hin CANBOX_HANDLE Handle des Akzeptanzfilters
Zurück CANBOX_IDENTIFIER_DATA Struktur zur Aufnahme der Akzeptanzfilter-Daten

5.2.23 canbox_close_filter (Index 32)

Schließt einen geöffneten CAN-Akzeptanzfilter.

5.2.23.1 Parameter

Hin CANBOX_HANDLE Handle des Akzeptanzfilters
Zurück Keine

5.2.24 canbox_open_sender (Index 13)

Öffnet einen Universal-Sender.

5.2.24.1 Parameter

Hin	USHORT	Null-basierter Index der Schnittstelle
Zurück	CANBOX_HANDLE	Handle des Universal-Senders für Zugriffe

5.2.25 canbox_write_sender (Index 24)

Stellt die übergebenen Daten auf den CAN-Bus. Für den Parameter ulld ist statt der Id das Handle des jeweiligen Identifiers anzugeben.

5.2.25.1 Parameter

Hin	CANBOX_UNIVERSAL_SENDER_DATA	Array von Identifier-Daten, die gesendet werden sollen. Im Parameter hSender ist zusätzlich jeweils das Handle des Universal-Senders anzugeben.
-----	------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------

Zurück Keine

5.2.26 canbox_close_sender (Index 14)

Schließt einen geöffneten Universal-Sender.

5.2.26.1 Parameter

Hin	CANBOX_HANDLE	Handle des Universal-Senders
-----	---------------	------------------------------

Zurück Keine

5.2.27 canbox_read_router_config (Index 34)

Liefert die Parametrierung eines CAN Routers.

5.2.27.1 Parameter

Hin	USHORT	Null-basierter Index der Schnittstelle
Zurück	CANBOX_ROUTER	Struktur zur Aufnahme der Router Konfiguration

5.2.28 canbox_write_router_config (Index 35)

Parametriert den Router anhand der übergebenen Werte.

5.2.28.1 Parameter

Hin CANBOX_ROUTER Router-Konfiguration

Zurück Keine

5.2.29 canbox_read_lan_config (Index 36)

Liefert die aktuellen EEPROM Einstellungen für die LAN-Verbindung.

5.2.29.1 Parameter

Hin Keine

Zurück CANBOX_LAN_CONFIG Struktur zur Aufnahme der LAN-Konfiguration

5.2.30 canbox_write_lan_config (Index 37)

Setzt die aktuellen EEPROM Einstellungen für die LAN-Verbindung. Die Einstellungen werden nach einem Neustart der CANbox[®] wirksam

5.2.30.1 Parameter

Hin CANBOX_LAN_CONFIG LAN Konfiguration

Zurück Keine

5.2.31 canbox_read_wlan_config (Index 38)

Liefert die aktuellen EEPROM Einstellungen für die WLAN Verbindung.

5.2.31.1 Parameter

Hin Keine

Zurück CANBOX_WLAN_CONFIG Struktur zur Aufnahme der WLAN Konfiguration

5.2.32 canbox_write_wlan_config (Index 39)

Setzt die aktuellen EEPROM Einstellungen für die WLAN Verbindung. Die Einstellungen werden nach einem Neustart der CANbox[®] wirksam

5.2.32.1 Parameter

Hin CANBOX_WLAN_CONFIG WLAN Konfiguration

Zurück Keine

6 Technische Daten

Parameter	min.	typ.	max.	Einheit	Anmerkung
Versorgungsspannung	6	12	60	V	
Leistungsaufnahme (abhängig von den aufgesteckten MAX- Modulen)	1	5	12	W	min.: ohne Module max.: begrenzt durch DC/DC- Wandler
Umgebungstemperatur	0	20	70	°C	
Abmessungen Breite		83		mm	Komplettes Gerät
Höhe		33		mm	
Tiefe		113		mm	
Schutzart		IP54			