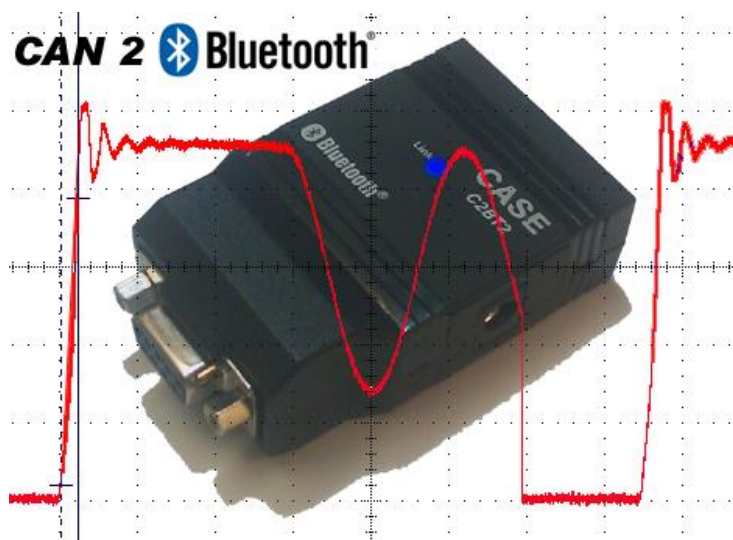

Bluetooth® wireless technology Adapter für den CAN-Bus

C2BT2

***Bluetooth®* wireless technology Adapter für den CAN-Bus**



Technische Dokumentation

Version 3.2 Stand 07.11.2012



C2BT2 – CAN-Bus auf Bluetooth® Adapter

HINWEIS

Diese Dokumentation ist mit größter Sorgfalt erstellt worden. Es wird jedoch keine Gewähr für die Freiheit von Fehlern und Irrtümern gegeben.

Die angegebenen Daten dienen allein der Produktionsbeschreibung und sind nicht als zugesicherte Eigenschaften im Rechtssinne aufzufassen.

Technische Änderungen bleiben vorbehalten.

Etwaige Schadensersatzansprüche gegen uns - gleich aus welchem Rechtsgrund - sind ausgeschlossen, soweit uns nicht Vorsatz oder grobe Fahrlässigkeit trifft.

Ein Nachdruck - auch auszugsweise - ist nur zulässig mit Zustimmung der CASE GmbH und mit genauer Quellenangabe.

Für dieses Dokument gelten die Bestimmungen zum Schutze des Urheberrechtes.

Alle Rechte vorbehalten.

Trademarks & Logos

Technologie

[Bluetooth®](#) ist eingetragenes Markenzeichen der Bluetooth® Special Interest Group ([SIG](#)). Als registriertes Mitglied der Bluetooth® SIG sind wir berechtigt den Namen Bluetooth® und die entsprechenden Logos zu verwenden. C2BT ist ein eingetragenes Markenzeichen der CASE GmbH. Der C2BT Adapter ist im End Product Listing (EPL) der Bluetooth® SIG registriert (Qualification Device ID - QDID: B013784).

[CAN](#) / [CiA®](#) and [CANopen®](#) are registered [Community Trademarks](#) of CAN in Automation e.V.

Betriebssysteme

Android™ is a [trademark of Google Inc.](#) The Android Robot can be used, reproduced, and modified freely in marketing communications.

Apple® iOS und [MFi Program](#): The Made for iPod, Made for iPhone, and Made for iPad logos mean that an electronic accessory has been designed to connect specifically to iPod, iPhone, or iPad and has been certified by the developer to meet Apple performance standards.

CASE GmbH

Computer Aided System Engineering

Zum Grenzgraben 23c
76698 Ubstadt-Weiher
Tel.: (+49) 07251/9653-0
Fax: (+49) 07251/9653-19
www.case-gmbh.de
support@case-gmbh.de

C2BT2 – CAN-Bus auf Bluetooth® Adapter

Inhaltsverzeichnis

1.....	VERSIONSVERWALTUNG	4
2.....	ÜBERSICHT	5
3.....	ANSCHLUSSSCHEMA UND FUNKTIONSBESCHREIBUNG	6
4.....	KOMMUNIKATIONSPROTOKOLL	7
4.1	Datenübertragung vom Bluetooth® fähigen Computer zum C2BT2	7
4.2	Datenübertragung vom C2BT2 zum Bluetooth® fähigen Computer	17
4.3	Übertragung in ASCII lesbarer Form	18
4.4	Berechnung der Checksumme	19
4.5	Auslieferungszustand	19
5.....	TECHNISCHE DATEN DES C2BT2	20
5.1	Elektrische Spezifikationen	20
5.2	Abmessungen	21
5.3	Anschlussbelegung der Sub-D-Buchse	21
6.....	HINWEISE ZUM ARBEITEN MIT MESSCARA-CE	22
6.1	Unterschied 1: Zeitstempelfunktion	22
6.2	Unterschied 2: Bluetooth® Pincode	22
6.3	DBC-Datei	23
7.....	ANDROID™- APPLE™- IOS UND MS-WINDOWS™ APPLIKATIONEN	24
8.....	TERMINALPROGRAMME	25

C2BT2 – CAN-Bus auf Bluetooth® Adapter

1 VERSIONSVERWALTUNG

Version 1.0 vom 27.03.2005	Dokumentname: DS4113A Originaldokument
Version 2.0 vom 19.04.2005	Dokumentname: DS4113B Gehäuseänderung
Version 2.01 vom 10.05.2005	Dokumentname: DS4113C Protokolländerung
Version 2.02 vom 07.08.2006	Dokumentname: DS4113D Informationen über Auslieferungszustand ergänzt, Ergänzung zum Übertragen des Bluetooth® Pincodes, Beschreibung DBC-File für Messcara
Version 2.03 vom 16.08.2006	Dokumentname: DS4113E Spannungsangabe auf Seite 4 richtiggestellt
Version 3.0 vom 10.03.2011	Dokumentname: DS4113F Hinweise zu ANDROID™ - und APPLE™ - iOS® Applikationen
Version 3.1 vom 29.10.2011	Anpassung der Anschlussbelegung wg. Redesign auf den C2BT3 Dokumentname: DS4113G QR-Code mit URL eingefügt
Version 3.2 vom 07.11.2012	Dokumentname: DS4113H Hinweise zur ASCII Datenübertragung ergänzt

VERFASST VON THOMAS WINKLER DIPL.-ING (FH).

C2BT2 – CAN-Bus auf Bluetooth® Adapter

2 ÜBERSICHT

C2BT2 Eigenschaften

- Drahtlose Übertragung der CAN-Bus Daten über Bluetooth®
- Verwendung des Serial Port Profiles (SPP)
- Unterstützung von Standard- und Extended Frames (11-Bit und 29-Bit Identifier)
- Übertragung eines Zeitstempels (16Bit Timer)
- Konfiguration über Bluetooth®
- Filtermasken einstellbar
- CAN-Bus Baudraten einstellbar
- Echo der gesendeten Daten
- Bluetooth® Class 2 zertifiziert
- Kleinste Abmessungen
- 9-pol. D-Sub Anschluss



Der C2BT2 überträgt CAN-Bus Telegramme über Bluetooth® und umgekehrt. Die Bluetooth® Antenne befindet sich im Inneren des Kunststoffgehäuses, die Funkreichweite beträgt ca. 15 m im Freifeld. Da alle CAN-Telegramme im C2BT2 mit einem Zeitstempel versehen werden, ist die zeitliche Zuordnung der eingetroffenen Nachricht immer gewährleistet. Baudraten und Filtermasken bzw. Akzeptanzfilter lassen sich über die serielle Bluetooth® Schnittstelle konfigurieren. Die Spannungsversorgung des C2BT2 reicht von 5V bis zu 35V DC. Der Anschluss an den CAN-Bus erfolgt über eine 9-polige Buchse.

Vielseitige Anwendungsgebiete in der Industrie, Automatisierung, KFZ-Industrie und eBikes

Zu Diagnosezwecken können bereits während der Produktion wichtige Daten drahtlos übertragen und angezeigt werden. Gleiches gilt für die Zulieferindustrie um Komponenten zu testen und Konfigurationsparameter zu übertragen. Aktuelle Messdaten können beispielsweise ‚online‘ während der Fahrt aus dem Motorraum auf einen PC oder PDA übertragen und dort ausgewertet werden. Beispiele für die Einsatzgebiete (Auszug)

- Zugangskontrollen, Türoffnersysteme
- Sitzverstellungen
- Erweiterung von Navigationssystemen
- Akkuzustandskontrolle bei eBikes
- Überwachung von Hochspannungsnetzen
- Sichere Datenübertragung in Baumaschinen
- Logistik - Drahtloskopplung von Trailern mit der Zugmaschine
- Reifendruckvisualisierung

C2BT2 – CAN-Bus auf Bluetooth® Adapter

3 ANSCHLUSSSCHEMA UND FUNKTIONSBESCHREIBUNG

Der C2BT2 wird über vier Anschlussleitungen angeschlossen. Im Fahrzeug können die beiden Versorgungsleitungen direkt an die Bordnetzversorgung angeschlossen werden. Die CAN-Bus Anschlüsse werden direkt mit den Leitungen CAN_H und CAN_L an den CAN-Bus angeschlossen.

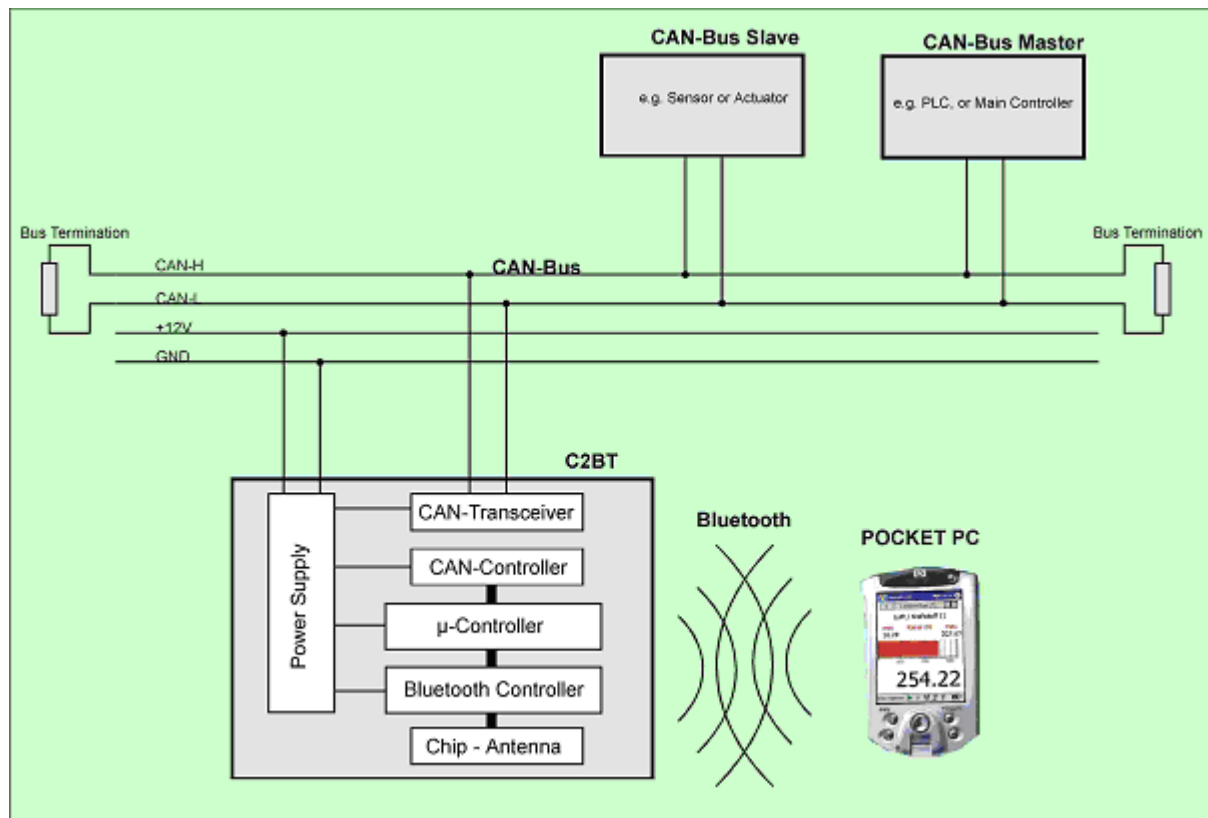


Abbildung 1 Anschlusschema

Sobald sich ein Bluetooth® fähiges Gerät (hier z.B. ein Pocket PC / PDA) in der Nähe des C2BT2 befindet, kann eine Bluetooth® Verbindung aufgebaut werden. Wurde zuvor noch nie eine Verbindung zwischen dem PDA und dem C2BT2 hergestellt, muss auf dem PDA zunächst nach Bluetooth® fähigen Geräten gesucht werden (Device Discovery). Dabei wird unter anderem der Remote Name des C2BT2 und dessen einzigartige Bluetooth® Adresse abgerufen und gespeichert. Der C2BT2 fordert dann einen Pincode an. Per Default ist dieser Pincode auf ‚0000‘ eingestellt. Der Pincode kann jedoch jederzeit nachträglich geändert werden. Nachdem der Pincode übertragen wurde, kann eine serielle Kommunikation hergestellt werden. Im PDA wird dafür eine neue COM Schnittstelle bereitgestellt und geöffnet. Die CAN-Bus Daten werden nun über diese serielle Verbindung übertragen. Mit Hilfe eines Terminal Programms können die Daten visualisiert oder aufgezeichnet werden. Konfigurationsparameter oder CAN-Bus Telegramme können im Gegenzug zum C2BT2 übertragen werden.

C2BT2 – CAN-Bus auf Bluetooth® Adapter

4 KOMMUNIKATIONSPROTOKOLL

Die Konfiguration und Kommunikation zwischen dem C2BT2 und einem Bluetooth® fähigen Computer ist durch das Serial Port Profile gewährleistet. Auf dem Bluetooth® fähigen Computer wird hierzu eine serielle Schnittstelle bereitgestellt. Über diese serielle Kommunikationsschnittstelle kann zu dem C2BT2 die im Folgenden beschriebene Kommunikation aufgebaut werden. Bei der Kommunikation zwischen dem C2BT2 und dem Bluetooth® fähigen Computer sind zwei verschiedene Modi der Übertragung möglich.

1. Die Übertragung in binärer Form.
2. Die Übertragung in ASCII lesbaren Form in hexadezimaler Ansicht.

Der Vorteil der ASCII lesbaren Form liegt darin, dass die CAN-Bus Daten mit einem einfachen Terminalprogramm angezeigt und aufgezeichnet werden können. Für zeitkritische Applikationen ist diese Übertragungsart jedoch nicht zu empfehlen.

Jeder Datenframe beginnt mit einem Startbyte gefolgt von einem Konfigurationsbyte und wird mit einer Checksumme abgeschlossen.

4.1 Datenübertragung vom Bluetooth® fähigen Computer zum C2BT2

Konfigurations- und CAN-Bus Daten werden in einem Datenframe mit variabler Länge zum C2BT2 übertragen. Alle Konfigurationen und Konfigurationsänderungen werden im internen Flash Speicher dauerhaft gespeichert und sind nach erneutem Einschalten wieder verfügbar.

Beispiel eines Datenframes zur Übertragung von CAN-Bus Daten:

1 Byte	1 Byte	4 Byte	1 Byte	1-8 Byte	1 Byte
Start	Configuration	Identifizier	Anzahl der Datenbytes	Daten	Checksum

Beispiel eines Datenframes zur Konfiguration der Baudrate:

1 Byte	1 Byte	1 Byte	1 Byte
Start	Configuration	Baudrate	Checksum

Der Wert des Startbytes ist immer **0x55** hexadezimal.

Start Byte:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	1	0	1	0	1	0	1

C2BT2 – CAN-Bus auf Bluetooth® Adapter

Das Configuration Byte enthält die Informationen über die Handhabung der darauffolgenden Datenbytes.

Configuration Byte

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CONFIG	STD/EXT	ECHO	TRES	SCFIG3	SCFIG2	SCFIG1	SCFIG0

- Bit 7 **CONFIG:** Configuration
 1 = Neue Konfigurationsdaten folgen
 0 = CAN - Bus - Nachrichten folgen
- Bit 6 **STD/EXT:** Standard oder Extended Frame
 1 = Daten mit Extended Frames senden
 0 = Daten mit Standard Frames senden
- Bit 5 **ECHO:**
 1 = Daten sollen zum Bluetooth® fähigen Computer zurückgesendet werden (ECHO ON)
 0 = Daten sollen **nicht** zum Bluetooth® fähigen Computer zurückgesendet werden (ECHO OFF)
- Bit 4 **TRES:** Timestamp -Timer Reset
 1 = Reset Timer
 0 = No Reset
- Bit 3-0 **SCFIG:** Set Configuration
 0001 = Baudrate einstellen
 0010 = Setze Filter und Masken
 0011 = Setze Zeitstempel
 0100 = Zeitstempel Übertragung ausschalten
 0101 = Bluetooth® Pincode einstellen
 0110 = Datenübergabe in Binärer Form
 0111 = Datenübergabe in ASCII lesbarer Form (Hexadezimale Darstellung)

Das **CONFIG** Bit 7 legt fest, ob die folgenden Daten zur Konfiguration des C2BT2 oder zur Weiterreichung an den CAN-Bus bestimmt sind. Ist das Bit = 0 so werden die Daten direkt als CAN-Message versendet.

Beispiel: Datenframe mit CONFIG = 0

1 Byte	1 Byte	4 Byte	1 Byte	1-8 Byte	1 Byte
Start	Configuration	Identifizier	Anzahl der Datenbytes	Daten	Checksum

Ist das **CONFIG** Bit = 1 werden die folgenden Bytes als Konfigurationsbytes interpretiert. Die Bits **STD/EXT**, **ECHO** und **TRES** werden nur dann angenommen, wenn das **CONFIG** Bit gelöscht ist (**CONFIG** Bit =0).

C2BT2 – CAN-Bus auf Bluetooth® Adapter

STD/EXT kennzeichnet, ob der folgende Identifier für einen Extended Frame oder Standard Frame bestimmt ist. Ist der übertragene Identifier größer als ein 11 Bit Identifier und ist das Bit **STD/EXT** gelöscht, wird die Nachricht trotzdem als Extended Frame auf dem CAN-Bus abgesetzt.

Durch Setzen des **ECHO** Bits werden die vom Bluetooth® fähigen Computer gesendeten CAN Bus Daten wieder an den Bluetooth® fähigen Computer zurückgesendet. Durch Setzen des **TRES** Bits wird der interne 16-Bit Timer auf 0 zurückgesetzt und der Timer gestartet. Die **SCFIG** Bits definieren welche Konfigurationsparameter folgen und werden nur angenommen, wenn das **CONFIG** Bit gesetzt ist.

Beispiel eines Datenframes mit CONFIG = 1 und SCFIG3-0 = 1

Start	Configuration	Checksum
-------	---------------	----------

4.1.1 Einstellen der Baudrate

Configuration Byte (Wert 0x81)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CONFIG	STD/EXT	ECHO	TRES	SCFIG3	SCFIG2	SCFIG1	SCFIG0
1	0	0	0	0	0	0	1

Datenframe

1Byte	1Byte	1Byte	1Byte
Start	Configuration	Baudrate	Checksum

Die Baudrate kann auf 7 feste Werte eingestellt werden:

Wert	Baudrate
1	20 kBit
2	50 kBit
3	100 kBit
4	125 kBit
5	250 kBit
6	500 kBit
7	1 Mbit

C2BT2 – CAN-Bus auf Bluetooth® Adapter

4.1.2 Einstellen der Filter und Masken

Der C2BT2 besitzt zwei Empfangspuffer mit priorisierter Nachrichtenspeicherung und stellt zwei 29-Bit Akzeptanzfilter und zwei 29-Bit Akzeptanzfiltermasken zur Verfügung. Einkommende Daten werden nur dann angenommen und über Bluetooth® übertragen, wenn die Akzeptanzfilterkriterien eingehalten werden. Nach dem Eintreffen einer gültigen CAN-Botschaft wird der Identifier dieser Nachricht zunächst mit den Filtereinstellungen verglichen. Stimmt der Identifier mit einem der Filter überein, so wird die Botschaft weiterverarbeitet. Die Akzeptanzfiltermaske bestimmt, welches Bit des Identifiers mit dem entsprechenden Filterbit verglichen werden soll.

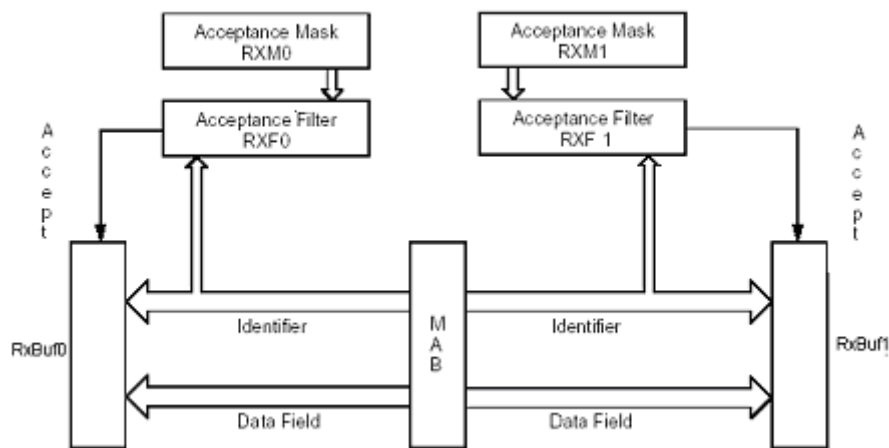
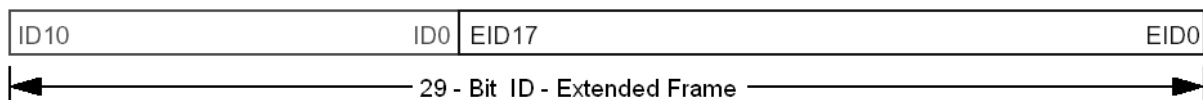


Abbildung 2: Konfiguration der Filter

Extended Frame



Standard Data Frame



Abbildung 3: Masken und Filter gelten für Standard und Extended Frames

C2BT2 – CAN-Bus auf Bluetooth® Adapter

Wahrheitstabelle der Filter/Maske

Maskenbit	Filter Bit	Identifizier Bit	Annahme/Ablehnung
0	X	X	Annahme
1	0	0	Annahme
1	0	1	Abgelehnt
1	1	0	Abgelehnt
1	1	1	Annahme

Übertragung der Filter- und Maskenparameter

Configuration Byte (Wert 0x82)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CONFIG	STD/EXT	ECHO	TRES	SCFIG3	SCFIG2	SCFIG1	SCFIG0
1	0	0	0	0	0	1	0

Datenframe

1 Byte	1 Byte	8 Byte	8 Byte	1 Byte
Start	Configuration	Filter	Masken	Checksum

Filtertabelle

Bytereihenfolge Bytename Beschreibung

Filter	Bytename	Beschreibung
1	RXF0SIDL	Receive Filter 0, Standard Identifier, Low Byte
2	RXF0SIDH	Receive Filter 0, Standard Identifier, High Byte
3	RXF0EID0	Receive Filter 0, Extended Identifier, Low Byte
4	RXF0EID8	Receive Filter 0, Extended Identifier, High Byte
5	RXF1SIDL	Receive Filter 1, Standard Identifier, Low Byte
6	RXF1SIDH	Receive Filter 1, Standard Identifier, High Byte
7	RXF1EID0	Receive Filter 1, Extended Identifier, Low Byte
8	RXF1EID8	Receive Filter 1, Extended Identifier, High Byte

Maskentabelle

Bytereihenfolge Bytename Beschreibung

Masken	Bytename	Beschreibung
1	RXM0SIDL	Receive Maske 0, Standard Identifier, Low Byte
2	RXM0SIDH	Receive Maske 0, Standard Identifier, High Byte
3	RXM0EID0	Receive Maske 0, Extended Identifier, Low Byte
4	RXM0EID8	Receive Maske 0, Extended Identifier, High Byte
5	RXM1SIDL	Receive Maske 1, Standard Identifier, Low Byte
6	RXM1SIDH	Receive Maske 1, Standard Identifier, High Byte
7	RXM1EID0	Receive Maske 1, Extended Identifier, Low Byte
8	RXM1EID8	Receive Maske 1, Extended Identifier, High Byte

Eine ausführliche Beschreibung zum Setzen von Filtern und Masken erhalten Sie im [DS220311A](#).

Bei Fragen wenden Sie sich bitte an unser Support Team unter support@case-gmbh.de.

C2BT2 – CAN-Bus auf Bluetooth® Adapter

4.1.3 Einstellen des Zeitstempels

Der C2BT2 besitzt einen 16-Bit Zähler, der im Millisekundentakt inkrementiert wird. Dieser Zählerwert kann einer eingetroffenen CAN-Botschaft zugeordnet werden und über Bluetooth® als Zeitstempel übertragen werden. Dieser Zähler kann auf einen bestimmten Wert gesetzt oder gelöscht werden. Der Wertebereich des Zeitstempels reicht von 0ms bis 59.999ms (1Min) und wird danach auf 0ms zurückgesetzt. Die Übertragung des Zeitstempels über Bluetooth® wird aktiviert, sobald der Zähler das erste Mal zurückgesetzt wird oder der Startwert übertragen wurde.

4.1.3.1 Rücksetzen des Zeitstempels

Nach Rücksetzen des Zeitstempels beginnt der Zähler wieder von Null an zu zählen.

Beispiel: CAN Daten mit Standard Identifier senden, Echo = aus, Timer Reset und Timer start

Configuration Byte (Wert 0x10)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CONFIG	STD/EXT	ECHO	TRES	SCFIG3	SCFIG2	SCFIG1	SCFIG0
0	0	0	1	0	0	0	0

Beispiel: Datenframe

1 Byte	1 Byte	4 Byte	1 Byte	1-8 Byte	1 Byte
Start	Configuration	Identifier	Anzahl der Datenbytes	Daten	Checksum

4.1.3.2 Setzen des Zeitstempels

Durch Setzen des Zeitstempels kann ein Zähler des Bluetooth® fähigen Computers mit dem Zähler des C2BT2 synchronisiert werden.

Configuration Byte (Wert 0x83)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CONFIG	STD/EXT	ECHO	TRES	SCFIG3	SCFIG2	SCFIG1	SCFIG0
1	0	0	0	0	0	1	1

Datenframe

1 Byte	1 Byte	2 Byte	1 Byte
Start	Configuration	Zeitstempel	Checksum

Bytereihenfolge	Bytename	Beschreibung
Zeitstempel		
1	TSTH	Timestamp High Byte
2	TSTL	Timestamp Low Byte

Sobald die Zeitstempeldaten übertragen sind, werden die CAN Botschaften mit einem Zeitstempel übertragen.

C2BT2 – CAN-Bus auf Bluetooth® Adapter

4.1.3.3 Zeitstempelübertragung deaktivieren

Die Übertragung des Zeitstempels wird durch die folgende Konfiguration deaktiviert.

Configuration Byte (Wert 0x84)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CONFIG	STD/EXT	ECHO	TRES	SCFIG3	SCFIG2	SCFIG1	SCFIG0
1	0	0	0	0	1	0	0

Datenframe

1 Byte	1 Byte	1 Byte
Start	Configuration	Checksum

C2BT2 – CAN-Bus auf Bluetooth® Adapter

4.1.4 Einstellen des Bluetooth® Pincodes

Der Bluetooth® Pincode ist im Auslieferungszustand auf ‚0000‘ gesetzt. Die Pincode Eingabe wird **einmalig** vom C2BT2 über Bluetooth® angefordert. Ein Datenaustausch über Bluetooth® ist erst dann möglich, wenn der im C2BT2 eingespeicherte Pincode mit dem übertragenen Pincode übereinstimmt. Ist der Pincode beim erstmaligen Kontakt zwischen PDA / PC ausgetauscht worden, so wird die Verbindung auch ohne Austausch des Pincodes aufgebaut. Bei der Änderung des Pincodes ist ein Reset des C2BT2 erforderlich. Dieser Reset wird automatisch nach der Übertragung des neuen Pincodes in den C2BT2 durchgeführt. Damit wird die Bluetooth® Kommunikation abgebrochen und muss neu initiiert werden.

Empfehlung: Bei der Übertragung eines Zeichens des jeweiligen Pincodes sind alle Zeichen erlaubt (0x00 bis 0xFF). Allerdings ist es von Vorteil, die Zeichen im ASCII lesbaren Format zu übertragen (von 0x30 bis 0x39).

Änderung des Pincodes

Configuration Byte (Wert 0x85)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CONFIG	STD/EXT	ECHO	TRES	SCFIG3	SCFIG2	SCFIG1	SCFIG0
1	0	0	0	0	1	0	1

Datenframe

1 Byte	1 Byte	4 Byte	1 Byte
Start	Configuration	Pincode	Checksum

Pincode-tabelle

Bytereihenfolge **Bytename** **Beschreibung**

Pincode		
1	PIN3	Pincodezahl ASCII lesbar
2	PIN2	Pincodezahl ASCII lesbar
3	PIN1	Pincodezahl ASCII lesbar
4	PIN0	Pincodezahl ASCII lesbar

Beispiel: Soll der Pincode 4711 übertragen werden dann sind die Bytes PIN3-0 wie folgt zu übertragen:

PIN3 = 4 (0x34)

PIN2 = 7 (0x37)

PIN1 = 1 (0x31)

PIN0 = 1 (0x31)

C2BT2 – CAN-Bus auf Bluetooth® Adapter

4.1.5 Datenübertragung in binärer bzw. ASCII Darstellung

Die Datenübertragung zwischen dem Bluetooth® fähigen Computer und dem C2BT2 kann in zwei verschiedenen Darstellungsarten erfolgen. Die Daten können als reine Binärdaten übertragen werden oder in ASCII lesbarer Form (Auslieferungszustand). Für die Übertragung in ASCII lesbarer Form ist es erforderlich, die einzelnen binären Datenbytes umzurechnen und in zwei Bytes zu übertragen. So wird zum Beispiel ein Byte mit dem hexadezimalen Wert 0x12 umgerechnet in 0x31 (Zahl ,1') und 0x32 (Zahl ,2') und übertragen. Dadurch können die Daten des C2BT2 mit einem einfachen Terminalprogramm angezeigt werden und es ist keine spezielle Auslesesoftware erforderlich.

Hinweis: Für die Übertragung in ASCII lesbarer Form ist es erforderlich, die binären Daten umzurechnen. Außerdem ist es erforderlich eine größere Anzahl an Datenbytes über die serielle Schnittstelle zu übertragen. Bei zeitkritischen Applikationen empfehlen wir daher die Übertragung der Daten im binären Format.

Datenübertragung in binärer Form

Configuration Byte (Wert 0x86)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CONFIG	STD/EXT	ECHO	TRES	SCFIG3	SCFIG2	SCFIG1	SCFIG0
1	0	0	0	0	1	1	0

Datenframe

1 Byte	1 Byte	1 Byte
Start	Configuration	Checksum

Beispiel: 0x55 0x86 0xDB

Datenübertragung in ASCII lesbarer Form

Configuration Byte (Wert 0x87)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CONFIG	STD/EXT	ECHO	TRES	SCFIG3	SCFIG2	SCFIG1	SCFIG0
1	0	0	0	0	1	1	1

Datenframe

1 Byte	1 Byte	1 Byte
Start	Configuration	Checksum

Beispiel: 0x55 0x87 0xDC

C2BT2 – CAN-Bus auf Bluetooth® Adapter

4.1.6 Zusammenfassung der Funktionen des Configuration Bytes

Configuration	Beschreibung
0x00	Sende Daten mit Standard Frame, Echo aus, kein TRES
0x10	Sende Daten mit Standard Frame, Echo aus, Timer Reset und Start
0x20	Sende Daten mit Standard Frame, Echo ein, kein TRES
0x30	Sende Daten mit Standard Frame, Echo ein, Timer Reset und Start
0x40	Sende Daten mit Extended Frame, Echo aus, kein TRES
0x50	Sende Daten mit Extended Frame, Echo aus, Timer Reset und Start
0x60	Sende Daten mit Extended Frame, Echo ein, kein TRES
0x70	Sende Daten mit Extended Frame, Echo ein, Timer Reset und Start
0x81	Einstellung der Baudrate
0x82	Setze Filter und Masken
0x83	Setze Zeitstempel
0x84	Zeitstempel Übertragung ausschalten
0x85	Bluetooth® Pincode einstellen
0x86	Datenübergabe in Binärer Form
0x87	Datenübergabe in ASCII lesbarer Form (Hexadezimale Darstellung)

Configuration	Byte 1	Byte 2	Data	Data	Data	Byte
0x00	Start	Configuration	4 Byte ID	1 Byte Num	Data	Checksum
0x10	Start	Configuration	4 Byte ID	1 Byte Num	Data	Checksum
0x20	Start	Configuration	4 Byte ID	1 Byte Num	Data	Checksum
0x30	Start	Configuration	4 Byte ID	1 Byte Num	Data	Checksum
0x40	Start	Configuration	4 Byte ID	1 Byte Num	Data	Checksum
0x50	Start	Configuration	4 Byte ID	1 Byte Num	Data	Checksum
0x60	Start	Configuration	4 Byte ID	1 Byte Num	Data	Checksum
0x70	Start	Configuration	4 Byte ID	1 Byte Num	Data	Checksum
0x81	Start	Configuration	1 Byte Baudrate	X	X	Checksum
0x82	Start	Configuration	8 Byte Filter	8 Byte Maske	X	Checksum
0x83	Start	Configuration	2 Byte Zeitstempel	X	X	Checksum
0x84	Start	Configuration	X	X	X	Checksum
0x85	Start	Configuration	4 Byte Pincode	X	X	Checksum
0x86	Start	Configuration	X	X	X	Checksum
0x87	Start	Configuration	X	X	X	Checksum

C2BT2 – CAN-Bus auf Bluetooth® Adapter

4.2 Datenübertragung vom C2BT2 zum Bluetooth® fähigen Computer

Jede Nachricht die zum Bluetooth® fähigen Computer gesendet wird beginnt mit dem Startbyte 0xAA. CAN-Bus Daten werden in einem Datenframe mit variabler Länge zum Bluetooth® fähigen Computer übertragen. Bei eingeschalteter ECHO Funktion sendet der C2BT2 die vom Bluetooth® fähigen Computer gesendeten Bytes zuzüglich der berechneten Checksumme wieder zurück an den Bluetooth® fähigen Computer.

Beispiel eines Echo Frames:

1 Byte	n Bytes	1 Byte
Start (0xAA)	Zurücksenden aller empfangenen Bytes	Checksum

Beispiel eines Datenframes zur Übertragung von CAN-Bus Daten:

1 Byte	1 Byte	2 Byte	4 Byte	1 Byte	1-8 Byte	1 Byte
Start	Configuration	Timestamp	Identifier	Anzahl der Datenbytes	Daten	Checksum

Hinweis: die beiden Bytes des Zeitstempels werden nur übertragen, wenn die Zeitstempelübertragung aktiviert ist (entweder durch Setzen des TRES Bits oder durch Setzen des Timestamp Registers).

Das Configuration Byte enthält die Informationen über den Busstatus und den C2BT2 Status. Das Configuration Byte wird nur mit einer eingegangenen CAN Botschaft übertragen.

Configuration Byte

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CONFIG	STD/EXT	BUSOFF	BUS128	BUS96	TBFULL	RXBOV	TSRACK

- Bit 7 **CONFIG:** Configuration Acknowledge
 1 = Neue Konfigurationsdaten wurden übernommen
 0 = Keine Konfigurationsdaten erhalten
- Bit 6 **STD/EXT:** Standard oder Extended Frame
 1 = Empfang eines Extended Frames
 0 = Empfang eines Standard Frames
- Bit 5 **BUSOFF:**
 1 = BUS OFF - Condition festgestellt
 0 = BUS ON - Condition festgestellt
- Bit 4 **BUS128:** Bus Heavy
 1 = mehr als 128 Fehler auf dem Bus festgestellt
 0 = weniger als 128 Fehler auf dem Bus festgestellt

C2BT2 – CAN-Bus auf Bluetooth® Adapter

- Bit 3 **BUS96:** Bus Light
1 = mehr als 96 Fehler auf dem Bus festgestellt
0 = weniger als 96 Fehler auf dem Bus festgestellt
- Bit 2 **TBFULL:** Transmit Buffer Full
1 = Sendepuffer ist voll, es können keine weiteren Daten gesendet werden
0 = Sendebereit
- Bit 1 **RXBOV:** Receiver Buffer Overflow
1 = eine gültige CAN-Botschaft ist eingetroffen, bevor der Empfangspuffer geleert wurde
0 = Kein Überlauf
- Bit 0 **TSRACK:** Time Stamp Reset Acknowledge
1 = Zeitstempel wurde zurückgesetzt
0 = Zeitstempel wurde nicht zurückgesetzt

4.2.1 Übertragungsformat des Zeitstempels

Der Zeitstempel besteht aus zwei Bytes und wird im folgenden Format ausgegeben:

Bytereihenfolge Zeitstempel	Bytename	Beschreibung
1	TSTH	Timestamp High Byte
2	TSTL	Timestamp Low Byte

Der Wertebereich des Zählers umfasst 2^{16} (65.536) Werte. Der Zähler wird nach jeweils einer Millisekunde inkrementiert.

4.3 Übertragung in ASCII lesbarer Form

Um die Darstellung der Daten auf einem Terminalprogramm in ASCII lesbarer Form übersichtlich zu halten, sendet der C2BT2 am Ende des Datenframes einen Zeilenumbruch (Carriage Return (0x0D) und Line Feed (0x0A)). Außerdem werden voneinander unabhängige Bytes durch Leerzeichen getrennt. Die Leerzeichen und die beiden Bytes des Zeilenumbruchs fließen nicht in die Berechnung der Checksumme mit ein.

Die ASCII Übertragung dient ausschließlich zur Anzeige der CAN-Bus Botschaften im ASCII Format. Alle Daten, die zum C2BT per Bluetooth übertragen werden, müssen im Binärformat übertragen werden. Somit können beispielsweise Konfigurationsdaten auch nur im Binären Format übertragen werden!.

Beispiel:

C2BT2 -> PC: Senden einer fehlerfreien CAN-Bus Nachricht mit Standard Identifier, eingeschalteter Zeitstempelübertragung, und 3 Datenbytes:

AA 00 0A13 0685 03 03 A6 FC F7

C2BT2 – CAN-Bus auf Bluetooth® Adapter

4.4 Berechnung der Checksumme

Die Checksumme errechnet sich aus der Summe aller übertragenen Bytes einschließlich des Startbytes.

Beispiel: Konfiguration zur Datenübertragung in binärer Form

Startbyte = 0x55

Configuration Byte = 0x86

Checksumme = 0x55 + 0x86 = 0xDB

4.5 Auslieferungszustand

Der C2BT2 wird in der folgenden Konfiguration ausgeliefert:

CAN Baudrate: 500kBit/s

Bluetooth® Pincode: '0000'

Datenübertragung: ASCII-Format

Filter: 0xFFFFFFFFFFFFFFFF (alle CAN-Botschaften werden durchgereicht)

Masken: 0x0000000000000000 (alle CAN-Botschaften werden durchgereicht)

Device Name: Inquiry – Remote Name Request: 'C2BT2 www.case.eu'

Profil: Serial Port Profile (SPP – Link Establishment)

C2BT2 – CAN-Bus auf Bluetooth® Adapter

5 TECHNISCHE DATEN DES C2BT2

5.1 Elektrische Spezifikationen

GENERELLE SPEZIFIKATIONEN

Die in Tabelle 1 aufgeführten maximalen Grenzwerte kennzeichnen die maximal zulässigen Betriebsparameter des C2BT2. Die Überschreitung bzw. Nichteinhaltung dieser Betriebsparameter kann zu dauerhaften Schäden am C2BT2 führen. Die empfohlenen Betriebsparameter in Tabelle 2 kennzeichnen die Umgebungsbedingungen für die der C2BT2 ausgelegt wurde.

Tabelle 1. Maximale Grenzwerte

Symbol	Parameter	Min	Max	Einheit
Vbat	Versorgungsspannung	-30	+35	V
PinRF	Eingestrahlte HF-Leistung		+15	dBm
PoutRF	Sendeausgangsleistung bei fo=2.4GHz	-2	+4	dBm
Ts	Lagertemperatur	-40	+100	°C
	ESD Human Body Model		2000	V

Tabelle 2. empfohlene Betriebsparameter

Symbol	Parameter	Min	Max	Einheit
Vbat	Versorgungsspannung	+5	+30	V
To	Arbeitstemperatur	-20	+85	°C
	Luftfeuchte		98	%RH

Tabelle 3. Elektrische Parameter bei verschiedenen Versorgungsspannungen ohne Busabschluss

Symbol	Parameter	@Vbat=9V	@Vbat=12V	@Vbat=24V	Einheit
Ir	Stromaufnahme ohne Bluetooth® Verbindung	10,5	7,5	4,3	mA
Is	Stromaufnahme bei bestehender Bluetooth® Verbindung und geringer Buslast @ 500kbps	16	11,7	6,8	mA
It	Stromaufnahme bei bestehender Bluetooth® Verbindung und hoher Buslast @ 500kbps	20	15,5	9	mA

C2BT2 – CAN-Bus auf Bluetooth® Adapter

5.2 Abmessungen

Alle Abmessungen in Inch bzw. [mm]

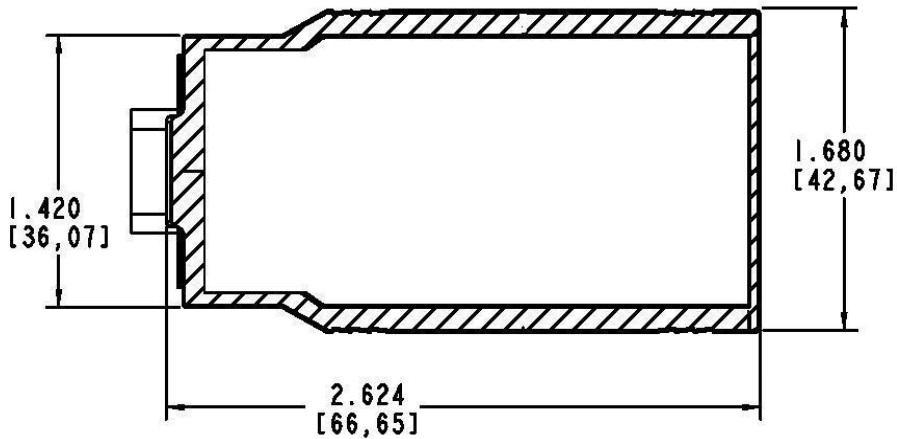


Abbildung 4: Abmessungen X,Y

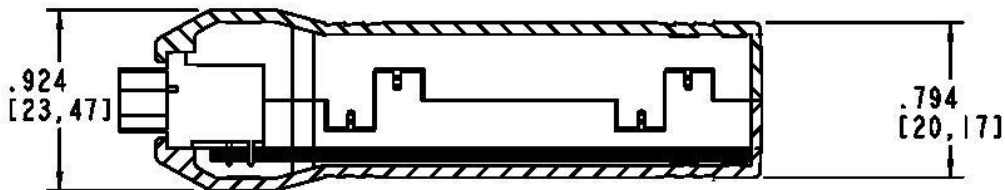
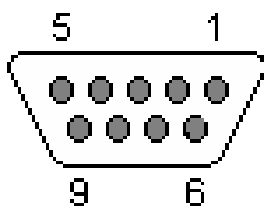


Abbildung 5: Abmessungen Z

5.3 Anschlussbelegung der Sub-D-Buchse



Pin1:	N.C.
Pin2:	CAN Low
Pin3:	GND
Pin4:	N.C.
Pin5:	N.C.
Pin6:	GND
Pin7:	CAN High
Pin8:	N.C.
Pin9:	+Vbat

Abbildung 6: Anschlussbelegung

C2BT2 – CAN-Bus auf Bluetooth® Adapter

6 HINWEISE ZUM ARBEITEN MIT MESSCARA-CE

MesscaRA-CE ist eine Visualisierungssoftware für CAN Bus Daten auf Bluetooth®-fähigen PDAs. MesscaRA-CE unterstützt die C2BT2 Hardware und ermöglicht so die drahtlose Visualisierung der CAN- Daten über die C2BT2 Bluetooth® Schnittstelle.

WICHTIGER HINWEIS: Es bestehen Unterschiede zwischen der C2BT2 Standard Firmware und der Kommunikationsfirmware zur Kommunikation zwischen MesscaRA-CE und dem C2BT2

6.1 Unterschied 1: Zeitstempelfunktion

Der Zeitstempel wird immer übertragen. Der Zeitstempel wird immer nach dem Identifier der entsprechenden CAN Botschaft übertragen. Er lässt sich nicht ausschalten!

6.2 Unterschied 2: Bluetooth® Pincode

Der Bluetooth® Pincode „0000“ wird nicht gesetzt und führt somit auch nicht zum Reset des C2BT2. Die Funkverbindung bleibt bestehen!

C2BT2 – CAN-Bus auf Bluetooth® Adapter

6.3 DBC-Datei

Die Zuordnung der CAN – Daten zu den entsprechenden Darstellungsarten in der PDA - Software MesscaRA-CE geschieht mittels DBC Datei.

6.3.1 Beispiel einer DBC-Datei

```
VERSION "HIPBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBNNNNNNNNNN/4/%%%/4/%"**4NNN//"
```

```
BS_:
```

Dezimale CAN Adresse (= 0x601)

```
BU_: ANIN0 USER
```

1: Lo Byte first, 0: Hibase first

```
BO_ 1537 ANIN0_MSG0: 0 SIM_53200158
```

```
SG_ ANIN0_1 : 0|8@1+ (1,0) [0|220] "Volt" Vector__XXX
```

+: Unsigned Format, -: Signed Format

```
SG_ ANIN0_2 : 8|8@1+ (1,0) [-10|200] "Volt" Vector__XXX
```

```
SG_ ANIN0_3 : 16|16@1- (0.1,0) [-100|2000] "°C" Vector__XXX
```

```
SG_ ANIN0_4 : 32|16@1- (0.1,0) [-100|2000] "°C" Vector__XXX
```

```
SG_ ANIN0_5 : 48|8@1- (0.1,0) [-10|20] "°C" Vector__XXX
```

Wertebereich

```
SG_ ANIN0_6 : 56|8@1+ (1,0) [0|255] "°C" Vector__XXX
```

0.1: Auflösung, 0: Offset

```
BA_DEF_BO_ "GenMsgSendType" STRING;
```

```
BA_DEF_BO_ "GenMsgCycleTime" INT 1 1000;
```

```
BA_DEF_BO_ "GenMsgCycleTimeus" INT 1000 1000000;
```

```
BA_DEF_ "Baudrate" INT 0 1000000;
```

```
BA_DEF_DEF_ "GenMsgSendType" "cyclic";
```

```
BA_DEF_DEF_ "GenMsgCycleTime" 100;
```

```
BA_DEF_DEF_ "GenMsgCycleTimeus" 1000000;
```

```
BA_DEF_DEF_ "Baudrate" 500000;
```

Wert beginnt ab Bit 56 und besitzt die Länge von 8 Bit

```
BA_ "Baudrate" 500000;
```

```
VAL_ 1536 ANIN0_T01 -32768 "NO_SENSOR" ;
```

C2BT2 – CAN-Bus auf Bluetooth® Adapter

7 ANDROID™- APPLE™- IOS UND MS-WINDOWS™ APPLIKATIONEN

Hinweis: Die C2BT Serie unterstützt vollständig die Bluetooth® Klasse 2.0. Das verwendete BT-Modul ist von der Bluetooth® Special Interest Group zertifiziert und erfüllt die Anforderungen des SPP (seriellen Port Profiles).

Microsoft

Auf Microsoft Plattformen sind keine Probleme bekannt. Tests wurden mit den folgenden Plattformen durchgeführt:

1. PC basierend: Windows 98, XP, VISTA, Windows 7
2. Pocket PC / PDA: Windows CE, Windows Mobile Pocket PC 6.x



Android™ Applikationen

Die Connectierbarkeit unter Android™ bis V2.2 (FROYO 05.2010) war nicht möglich. Abhängig vom Gerätehersteller und der Versionen 2.3 bis 2.3.7 (Gingerbread) bestanden Verbindungsprobleme, die ein Auffinden eines C2BT nicht ermöglichten. Ab der Version 3.0 (Honeycomb 02.2011) sind uns keine Probleme mehr bekannt. Weitere Informationen zum Betriebssystem Android™ und den Versionsnummern erhalten Sie unter [http://de.wikipedia.org/wiki/Android_\(Betriebssystem\)](http://de.wikipedia.org/wiki/Android_(Betriebssystem))



Externe Geräte, die mit Apple Geräten kommunizieren, müssen im Zuge einer MFI Lizenzierung von Apple zertifiziert werden. Die Verbindung des C2BT – Adapters zu einem Gerät von Apple (Iphon, iPad, etc.) ist daher noch nicht möglich. Allerdings befinden wir uns im MFI Zertifizierungsprozess ([MFI Programm](#)).

C2BT2 – CAN-Bus auf Bluetooth® Adapter

8 TERMINALPROGRAMME

Zur schnellen Inbetriebnahme und ersten Tests der C2BT Adapter können wir die folgenden Terminalprogramme empfehlen:

Plattform:	Beschreibung
Windows PC: Windows 98 Windows XP Windows VISTA Windows 7	<u>Name: Hyperterminal - Hersteller: Microsoft</u> bis Win XP als Standard vorhanden http://technet.microsoft.com/en-us/library/cc736511(WS.10).aspx <u>Name: Hterm - Hersteller: Tobias Hammer</u> Kostenloses Terminalprogramm zum Senden und Empfangen in ASCII und Binär http://www.der-hammer.info/terminal/ <u>Name: Docklight - Hersteller: FuH</u> Terminalprogramm zum Senden und Empfangen in ASCII und Binär mit erweiterten Macrofunktionen http://www.docklight.de/
Windows Mobile: 2002/2003/2003SE/5/6	Name mToken, Hersteller: Coung Networks http://www.choung.net/products/discontinued/
Android: ab Version 2.3.x (Gingerbread)	Name: bTerm, Hersteller: SENA Kostenloses APP – einfaches Terminalprogramm, einfache Visualisierung APP im Android Market unter „Bterm“ Info's unter www.sena.com/download/manual_bterm/overview.html

