

PC: CAN-Low-Level Modus

Scandevil besitzt ab Bios-Version 1.2.0 einen CAN-Interface-Modus. Über eine nur wenige Befehle umfassende Schnittstelle können verschiedene Aktionen auf dem CAN-Bus-System ausgeführt werden:

- Senden von Datenpaketen zum CAN-Bus.
- Empfangen von Datenpaketen auf dem CAN-Bus.
- Protokollieren des Datenverkehrs auf dem CAN-Bus.

Hierzu können verschiedene Parameter eingestellt werden:

- 8 feste Baudraten zwischen 20kBit und 1 MBit.
- Variable Baudrate durch direkte Eingabe des gewünschten Wertes (20kBit – 1MBit)
- 11 oder 29 Bit CAN-ID.
- 2 verschiedene Empfangsfilter konfigurierbar, auch bei aktivem CAN-Bus veränderbar
- Senden von RTR-Datenanforderungen.
- Zeitstempel beim Empfang der Datenpakete.

Der Befehlssatz des CAN-Interface-Modus ist weitgehend zu anderen existierenden Adaptern kompatibel. Durch den abweichenden Controller-Typ erwarten einige Befehle jedoch ein etwas unterschiedliches Datenformat.

Serielle Datenverbindung

Die Baudrate für die virtuelle serielle Schnittstelle wird vom Scandevil ignoriert, es wird immer die maximale USB-Geschwindigkeit benutzt. Es wird kein Echo der ankommenden Daten zurückgeschickt. Alle Eingaben und Antworten erfolgen im Klartext als ASCII-Zeichen mit abschließendem CR (ASCII 13). Fehler bei der Eingabe werden mit BELL (ASCII 7) quittiert.

Die Software sollte als erste Zeichen mehrfach ein CR senden, damit eventuell vorhandene Befehlsreste gelöscht und nachfolgende Befehle sicher erkannt werden.

Aufgrund des schnellen USB-Interface zum PC können pro Sekunde bis zu 1000 CAN-Pakete zum PC übertragen werden.

BEFEHLSSATZ

Befehl: **?**

Bezeichnung: Liste aller Befehle ausgeben

Beispiel: ?[CR]

Rückgabe: ?TtRrMmHhOLABCNniFGSsVz [CR]

Eine Liste aller möglichen Befehle ausgeben.

Befehl: **S [1 . . 8]**

Bezeichnung: CAN-Bitrate (fest) einstellen, abfragen

Standardwert: S6 (500 kBit)

Beispiel: S5 [CR]

Stellt die Baudrate auf 250kBit ein

Rückgabe: [CR] für OK, [BELL] für Fehler

Beispiel: S [CR]

Zeigt die aktuell eingestellte Bitrate an

Rückgabe: S6: BTR=007FC003 BRP=4 TSEG1=16 TSEG2=8 SJW=4 (250.000 bps)

Befehl	Bitrate
S0	10 kBit
S1	20 kBit
S2	50 kBit
S3	100 kBit
S4	125 kBit
S5	250 kBit
S6	500 kBit (standard)
S7	800 kBit
S8	1 MBit

Der im Scandevil verwendete CAN-Controller kann keine Bitraten unterhalb 20 kBit erzeugen, der Befehl S0 wird aus diesem Grund mit einer Fehlermeldung quittiert.

Die Bitrate 800 kBit kann aufgrund der verwendeten Taktfrequenz des CAN-Controllers nicht genau erzeugt werden. Dies ist jedoch eine relativ selten benutzte Bitrate und ist nur aus Kompatibilitätsgründen vorhanden.

Dieser Befehl kann nicht benutzt werden, wenn der CAN-Bus mit einem der Befehle O, L, A, B aktiviert ist. Unbedingt vorher mit dem Befehl C schließen.

Befehl: s [n]

Bezeichnung: CAN-Bitrate (variabel) einstellen, abfragen

Beispiel: s300k [CR]
Stellt die Baudrate auf 300kBit ein

Beispiel: s200000 [CR]
Stellt die Baudrate auf 200kBit ein

Rückgabe: [CR] für OK, [BELL] für Fehler

Beispiel: s [CR]
Zeigt die aktuell eingestellte Bitrate an

Rückgabe: S*: BTR=005CC003 BRP=4 TSEG1=13 TSEG2=6 SJW=4 (312.500 bps)

Aufgrund der verwendeten Taktfrequenz des CAN-Controllers können nicht alle Bitraten genau erzeugt werden. Ob eine gewünschte Bitrate genau erzeugt werden kann, sollte mit dem Befehl s geprüft werden. Es sind Bitraten zwischen 20kBit und 1MBit möglich.

Dieser Befehl kann nicht benutzt werden, wenn der CAN-Bus mit einem der Befehle O, L, A, B aktiviert ist. Unbedingt vorher mit dem Befehl C schließen.

Befehl: O

Bezeichnung: CAN-Bus öffnen (Open, Bus aktiv)

Beispiel: O [CR]

Rückgabe: [CR] für OK, [BELL] für Fehler

Die CAN-Kommunikation wird gestartet, Senden und Empfangen von Datenpaketen ist nun möglich. Dieser Befehl ist nur möglich, wenn der CAN-Bus inaktiv ist.

Befehl: C

Bezeichnung: CAN-Bus schließen (Close, Bus inaktiv)

Beispiel: C [CR]

Rückgabe: [CR] für OK, [BELL] für Fehler

Die CAN-Kommunikation wird beendet.

Dieser Befehl kann nur benutzt werden, wenn der CAN-Bus mit einem der Befehle O, L, A, B aktiviert ist.

Befehl: L

Bezeichnung: CAN-Bus zum Loggen öffnen (Listen only, Bus passiv)

Beispiel: L [CR]

Rückgabe: [CR] für OK, [BELL] für Fehler

Die CAN-Kommunikation wird nur zum loggen der Daten auf dem Bus geöffnet. Der Scandevil Controller verhält sich absolut passiv. Es können keine CAN-Pakete gesendet sondern nur empfangen werden.

Dieser Befehl kann nur benutzt werden, wenn der CAN-Bus inaktiv ist.

Befehl: A

Bezeichnung: CAN-Bus öffnen (Open, no Ack, Bus aktiv)

Beispiel: A [CR]

Rückgabe: [CR] für OK, [BELL] für Fehler

Die CAN-Kommunikation wird gestartet, Senden und Empfangen von Datenpaketen ist nun möglich.

Dieses ist ein Testmodus, der vor allem zur Überprüfung der gesendeten Daten auf dem Bus benötigt wird. Es ist kein 2. CAN-Gerät auf dem Bus erforderlich, das ein gesendetes Can-Paket bestätigt (acknowledge).

Dieser Befehl ist nur möglich, wenn der CAN-Bus inaktiv ist.

Befehl: B

Bezeichnung: CAN-Bus öffnen (Open, Self Receive, Bus aktiv)

Beispiel: B [CR]

Rückgabe: [CR] für OK, [BELL] für Fehler

Die CAN-Kommunikation wird gestartet, Senden und Empfangen von Datenpaketen ist nun möglich. Dieses ist ein Testmodus, bei dem die gesendeten Datenpakete selber wieder empfangen werden. Dieser Modus ist besonders dafür geeignet, um die Empfangsfilter zu testen. Dieser Befehl ist nur möglich, wenn der CAN-Bus inaktiv ist.

Befehl: V

Bezeichnung: Versionsnummer des Bootloaders und Bios ausgeben

Beispiel: V [CR]

Rückgabe: Vxxyy [CR]

Die Versionsnummern des Bootloaders und Bios im Scandevil werden ausgegeben.
xx = Bootloader-Version (10 = Version 1.0)
yy = Bios-Version (12 = Version 1.2)

Befehl: N

Bezeichnung: Die Seriennummer des Scandevil wird ausgegeben (kompatibel, kurz)

Beispiel: N [CR]

Rückgabe: Nxxxx [CR]

Zur Kompatibilität mit anderen CAN-Controllern wird die Seriennummer des Gerätes 4-stellig ausgegeben.

Befehl: n

Bezeichnung: Die Seriennummer des Scandevil wird ausgegeben (lang)

Beispiel: n [CR]

Rückgabe: n12345-67890-123 [CR]

Die Seriennummer des Scandevil wird komplett ausgegeben.

Befehl: **i**

Bezeichnung: Identifikation ausgeben

Beispiel: i[CR]

Rückgabe: iDIAMEX * SCANDEVIL[CR]

Identifikationstext des verwendeten CAN-Interface ausgeben.

Befehl: **F**

Bezeichnung: CAN-Status abfragen

Beispiel: F[CR]

Rückgabe: Fttrrssss (Mode: Open)[CR]

Nach einem Fehler bei der CAN-Übertragung kann hier der Status des CAN-Controllers abgefragt werden.

tt = Anzahl der Fehler bei TX

rr = Anzahl der Fehler bei RX

ssss = Fehlercode (siehe Datenblatt des LPC17xx Register CANxICR, Bits 16..31)

Befehl: **Z [0 | 1]**

Bezeichnung: Zeitstempel ein-/ausschalten, Status abfragen

Beispiel: Z1[CR]

Rückgabe: [CR] für OK, [BELL] für Fehler

Beispiel: Z[CR]

Rückgabe: Z0 (Timestamp: Off)[CR]

Mit Z1 wird an jede CAN-Nachricht vom Interface zu PC ein 16-Bit Zeitstempel angehängt. Der 16-Bit Wert wird im Millisekunden Bereich erhöht und endet bei 59999d = EA5Fh, ein Durchlauf dauert genau 1 Minute.

Befehl: **G[0|1|2]**

Bezeichnung: Filterart auswählen, Status abfragen (Standard G1)

Beispiel: G1[CR]

Rückgabe: [CR] für OK, [BELL] für Fehler

Beispiel: G[CR]

Rückgabe: G1 (Filter: Masked)[CR]

Zur Filterung der empfangenen Datenpakete stehen beim Scandevil zwei unterschiedliche Filter zur Verfügung, die verschiedene Methoden zur Paketfilterung benutzen.

Befehl	Filterart	Filterbefehle
G0	Filter aus	
G1	Software-Filter Bitmaskiert	M = Id, m = Maske
G2	Hardware-Filter Low-High	H = 29-Bit, h = 11-Bit

Bei ausgeschaltetem Filter werden alle auf dem Bus übertragenen Datenpakete zum PC übertragen. Beachten Sie, dass es nicht möglich ist, mehr als 1000 Pakete pro Sekunde zu übertragen (USB = 1ms/Paket).

Zur Filterung der Pakete kann einer der beiden Filter angewählt werden.

Mit G1 wird ein Filter aktiviert, bei dem die empfangenen IDs aufgrund einer Bitmaske ausgefiltert werden. Siehe Befehle m, M.

Diese Art der Filterung ist weitgehend kompatibel mit anderen CAN-Interfaces und existierender Software. Bei CAN-Bussen mit hohen Datenraten und Traffic sollte jedoch ein Hardware-Filter konfiguriert werden um die Reaktionszeit des Scandevil zu erhöhen.

Mit G2 wird ein Filter aktiviert, bei dem die empfangenen IDs aufgrund einer Anfangs- und Endadresse ausgefiltert werden. Siehe Befehle h, H.

Der Hardware-Filter ist gegenüber dem Software-Filter vorzuziehen, da der CAN-Controller die Pakete selbständig filtert und auch bei hohen Datenraten und Traffic auf dem CAN-Bus nicht die Reaktionszeit herabsetzt.

Befehl: **M [xxxxxxxx]**

Bezeichnung: Software Filter-Id einstellen (11 oder 29 bit) (Standardwert 00000000)

Beispiel: M7E0 [CR]

Rückgabe: [CR] für OK, [BELL] für Fehler

Beispiel: M[CR]

Rückgabe: Z000007E0 [CR]

29-Bit Filter-Id: Maximalwert 1FFFFFFF.

11-Bit-Filter-Id: Maximalwert 7FF

Es existiert nur ein Filter, der gemeinsam für 11-Bit und 29-Bit-Filterung benutzt wird.

Der Software-Filter wird nur angewendet, wenn dieser zuvor mit dem Befehl G1 (Standardwert nach dem Einschalten) aktiviert wurde. Der Software-Filter kann jederzeit geändert werden.

Befehl: **m [xxxxxxxx]**

Bezeichnung: Software Filter-Maske einstellen (11 oder 29 bit) (Standardwert 00000000)

Beispiel: m7F0 [CR]

Rückgabe: [CR] für OK, [BELL] für Fehler

Beispiel: m[CR]

Rückgabe: Z000007F0 [CR]

29-Bit Filter-Maske: Maximalwert 1FFFFFFF.

11-Bit-Filter-Maske: Maximalwert 7FF

Die Filter-Maske gibt an, welche Bits der Filter-Id zur Paketfilterung benutzt werden sollen. Nur 1-Bits werden zur Filterung benutzt, 0-Bits werden ignoriert.

Beispiel:

```
M7E0      11111100000  ID
m7F0      1111110000  MASKE
-----
          1111110xxxx  FILTER
```

Alle Adressen zwischen 7E0 und 7EF werden vom Filter erkannt und zum PC übertragen.

Es existiert nur eine Maske, die gemeinsam für 11-Bit und 29-Bit-Filterung benutzt wird.

Der Software-Filter wird nur angewendet, wenn dieser zuvor mit dem Befehl G1 (Standardwert nach dem Einschalten) aktiviert wurde. Die Software-Filtermaske kann jederzeit geändert werden.

Befehl: **H [aaaaaaaaaeEEEEEE]**

Bezeichnung: Hardware Filter-Adressen Anfang-Ende für 29-Bit einstellen, abfragen.

Beispiel: H12340001234FFFF [CR]

Rückgabe: [CR] für OK, [BELL] für Fehler

Beispiel: H [CR]

Rückgabe: H12340000-1234FFFF [CR]

Die Maximalwerte für Anfangs- und Endadresse ist 1FFFFFFF. Die Endadresse muss immer gleich oder größer als die Anfangsadresse sein.

Hardware-Filter für 11-Bit und 29-Bit können unabhängig voneinander definiert werden.

Der Hardware-Filter wird nur angewendet, wenn dieser zuvor mit dem Befehl G2 aktiviert wurde.

Der Hardware-Filter kann jederzeit geändert werden.

Befehl: **h [aaaeEE]**

Bezeichnung: Hardware Filter-Adressen Anfang-Ende für 11-Bit einstellen, abfragen.

Beispiel: h7007FF [CR]

Rückgabe: [CR] für OK, [BELL] für Fehler

Beispiel: h [CR]

Rückgabe: h700-7FF [CR]

Die Maximalwerte für Anfangs- und Endadresse ist 7FF. Die Endadresse muss immer gleich oder größer als die Anfangsadresse sein.

Hardware-Filter für 11-Bit und 29-Bit können unabhängig voneinander definiert werden.

Der Hardware-Filter wird nur angewendet, wenn dieser zuvor mit dem Befehl G2 aktiviert wurde.

Der Hardware-Filter kann jederzeit geändert werden.

Befehl: **t[*iiindd*...]**

Bezeichnung: Sende 11-Bit Datenpaket

Beispiel: t78981122334455667788 [CR]

Rückgabe: [CR] für OK, [BELL] für Fehler

Beispiel: t6001AB [CR]

Rückgabe: [CR] für OK, [BELL] für Fehler

Es wird ein 11-Bit Paket mit der ID *iii* gesendet. *n* gibt die Anzahl der zu sendenden Bytes an. Die Anzahl der Datenbytes *dd* muss exakt mit dem Wert von *n* übereinstimmen.

Die Sendefunktion ist nur aktiv, wenn der CAN-Bus mit einem der Befehle O, A, B zuvor geöffnet wurde.

Befehl: **T[*iiiiiiiindd*...]**

Bezeichnung: Sende 29-Bit Datenpaket

Beispiel: t1890ABCD81122334455667788 [CR]

Rückgabe: [CR] für OK, [BELL] für Fehler

Beispiel: t1FFF10001AB [CR]

Rückgabe: [CR] für OK, [BELL] für Fehler

Es wird ein 29-Bit Paket mit der ID *iiiiiii* gesendet. *n* gibt die Anzahl der zu sendenden Bytes an. Die Anzahl der Datenbytes *dd* muss exakt mit dem Wert von *n* übereinstimmen.

Die Sendefunktion ist nur aktiv, wenn der CAN-Bus mit einem der Befehle O, A, B zuvor geöffnet wurde.

Befehl: **r[*iiin*]**

Bezeichnung: Sende 11-Bit RTR-Datenanforderung (Remote Transmission Request)

Beispiel: `r7890` [CR]

Rückgabe: [CR] für OK, [BELL] für Fehler

Beispiel: `t6008` [CR]

Rückgabe: [CR] für OK, [BELL] für Fehler

Es wird eine 11-Bit RTR-Datenanforderung mit der ID *iii* gesendet. *n* gibt die Anzahl der anzufordernden Datenbytes an. Es werden keine Datenbytes gesendet.

Die Sendefunktion ist nur aktiv, wenn der CAN-Bus mit einem der Befehle O, A, B zuvor geöffnet wurde.

Befehl: **R[*iiiiiiin*]**

Bezeichnung: Sende 29-Bit RTR-Datenanforderung (Remote Transmission Request)

Beispiel: `R1890ABCD0` [CR]

Rückgabe: [CR] für OK, [BELL] für Fehler

Beispiel: `R1FFF10008` [CR]

Rückgabe: [CR] für OK, [BELL] für Fehler

Es wird eine 29-Bit RTR-Datenanforderung mit der ID *iiiiiiin* gesendet. *n* gibt die Anzahl der anzufordernden Datenbytes an. Es werden keine Datenbytes gesendet.

Die Sendefunktion ist nur aktiv, wenn der CAN-Bus mit einem der Befehle O, A, B zuvor geöffnet wurde.

DATENPAKETE

Zum Senden und Empfangen der CAN-Datenpakete wurde folgendes Format festgelegt. Alle Daten bestehen aus lesbaren ASCII-Zeichen, alle Werte werden im Hexadezimal-Format übertragen. Sende- und Empfangspakete sind nahezu identisch, bei den empfangenen Daten kann optional ein Zeitstempel angehängt werden (siehe Befehl Z).

11 Bit-CAN-Datenpaket senden

`tiiiindd...`

`t` = Kennung für 11-Bit Paket mit Daten
`iii` = 11-Bit Sende-Id
`n` = Anzahl der nachfolgenden Bytes
`dd` = Zu sendende Datenbytes (2 Hexzeichen pro Byte)

29 Bit-CAN-Datenpaket senden

`Tiiiiiiiiindd...`

`T` = Kennung für 29-Bit Paket mit Daten
`iiiiiiii` = 29-Bit Sende-Id
`n` = Anzahl der nachfolgenden Bytes
`dd` = Zu sendende Datenbytes (2 Hexzeichen pro Byte)

11 Bit-CAN-RTR-Paket anfordern

`riiin`

`r` = Kennung für 11-Bit RTR-Anforderung (Remote Transmission Request)
`iii` = 11-Bit Sende-Id
`n` = Anzahl der angeforderten Bytes

Bei einer RTR-Anforderung werden keine Daten gesendet. `n` gibt in diesem Fall die Anzahl der Bytes an, die die Gegenstelle auf die Anforderung senden soll.

29 Bit-CAN-RTR-Paket anfordern

`Riiiiiiiiin`

`R` = Kennung für 29-Bit RTR-Anforderung (Remote Transmission Request)
`iiiiiiii` = 29-Bit Sende-Id

Bei einer RTR-Anforderung werden keine Daten gesendet. `n` gibt in diesem Fall die Anzahl der Bytes an, die die Gegenstelle auf die Anforderung senden soll.

11 Bit-CAN-Datenpaket empfangen

`tiiiindd...zzzz`

`t` = Kennung für 11-Bit Paket mit Daten

`iii` = 11-Bit Empfangs-Id

`n` = Anzahl der nachfolgenden Bytes

`dd` = Die empfangenen Datenbytes (2 Hexzeichen pro Byte)

`zzzz` = Optionaler Zeitstempel in Millisekunden, wenn mit Befehl Z1 aktiviert.

29 Bit-CAN-Datenpaket empfangen

`Tiiiiiiiindd...zzzz`

`T` = Kennung für 29-Bit Paket mit Daten

`iiiiiii` = 29-Bit Empfangs-Id

`n` = Anzahl der nachfolgenden Bytes

`dd` = Die empfangenen Datenbytes (2 Hexzeichen pro Byte)

`zzzz` = Optionaler Zeitstempel in Millisekunden, wenn mit Befehl Z1 aktiviert.

11 Bit-RTR-Anforderung empfangen

`riiinzzzz`

`r` = Kennung für 11-Bit-Anforderung (Remote Transmission Request)

`iii` = 11-Bit Empfangs-Id

`n` = Anzahl der angeforderten Bytes

`zzzz` = Optionaler Zeitstempel in Millisekunden, wenn mit Befehl Z1 aktiviert.

29 Bit-RTR-Anforderung empfangen

`Riiiiiiiinzzzz`

`R` = Kennung für 29-Bit-Anforderung (Remote Transmission Request)

`iiiiiii` = 29-Bit Empfangs-Id

`n` = Anzahl der angeforderten Bytes

`zzzz` = Optionaler Zeitstempel in Millisekunden, wenn mit Befehl Z1 aktiviert.