

# PCAN-MiniDisplay

Visualisierung, Aufzeichnung und  
Wiedergabe von CAN-Daten

## Benutzerhandbuch



Dokumentversion 2.4.0 (2019-11-19)

**PEAK**  
System

## Berücksichtigte Produkte

Produkt-bezeichnung	Ausführung	Firmware	Artikelnummer
PCAN-MiniDisplay	Display mit Einbaurahmen	V1.4.0 oder höher	IPEH-002262 (ab Seriennummer 150)
PCAN-MiniDisplay Case	Display im Gehäuse mit Tastern		IPEH-002262-KSM01 (ab Seriennummer 12215)

Auf der Titelseite ist das PCAN-MiniDisplay mit Einbaurahmen abgebildet.

PCAN® ist eine eingetragene Marke der PEAK-System Technik GmbH. microSD™ ist eine Marke oder eingetragene Marke von SD-3C, LLC in den Vereinigten Staaten von Amerika, in anderen Ländern oder beides.

Alle anderen Produktnamen in diesem Dokument können Marken oder eingetragene Marken der jeweiligen Eigentümer sein. Diese sind nicht ausdrücklich durch ™ oder ® gekennzeichnet.

Copyright © 2019 PEAK-System Technik GmbH

Die Vervielfältigung (Kopie, Druck oder in anderer Form) sowie die elektronische Verbreitung dieses Dokuments ist nur mit ausdrücklicher, schriftlicher Genehmigung der PEAK-System Technik GmbH erlaubt. Die PEAK-System Technik GmbH behält sich das Recht zur Änderung technischer Daten ohne vorherige Ankündigung vor. Es gelten die allgemeinen Geschäftsbedingungen sowie die Bestimmungen der Lizenzverträge. Alle Rechte vorbehalten.

PEAK-System Technik GmbH  
Otto-Röhm-Straße 69  
64293 Darmstadt  
Deutschland

Telefon: +49 (0)6151 8173-20  
Telefax: +49 (0)6151 8173-29

[www.peak-system.com](http://www.peak-system.com)  
[info@peak-system.com](mailto:info@peak-system.com)

Dokumentversion 2.4.0 (2019-11-19)

# Inhalt

<b>1</b>	<b>Einleitung</b>	<b>5</b>
1.1	Eigenschaften im Überblick	6
1.2	Lieferumfang	7
<b>2</b>	<b>Anschlüsse</b>	<b>8</b>
2.1	J1 Externe Taster	9
2.2	J2 System	9
2.3	J3 und J7 USB	10
2.4	J4 JTAG	12
2.5	J5 microSD	12
2.6	J6 Seriell	13
<b>3</b>	<b>Szenendefinition</b>	<b>14</b>
3.1	Allgemeine Beschreibung einer *.ins-Datei	14
3.2	Beispiel einer *.ins-Datei	16
3.3	Verwendung mehrerer Szenen	18
<b>4</b>	<b>Schritte zur Inbetriebnahme</b>	<b>19</b>
4.1	Voraussetzungen	19
4.2	Speicherkarte vorbereiten	19
4.3	Einschalten des Geräts	22
4.4	Menüs	22
4.5	Einstellen der Startoptionen	24
<b>5</b>	<b>CAN-Tracing</b>	<b>25</b>
5.1	CAN-Verkehr aufzeichnen	25
5.2	Aufgezeichneten CAN-Verkehr wiedergeben	29
5.3	Aufgezeichneten CAN-Verkehr auf dem PC verwenden	30

5.4	Filtern des CAN-Verkehrs bei Aufzeichnungen	32
<b>6</b>	<b>Einstellungen und Wartung</b>	<b>33</b>
6.1	Device Settings	33
6.2	Interne Statistiken	36
6.3	Memory Card	37
<b>7</b>	<b>Eigene Bilder verwenden</b>	<b>39</b>
7.1	Alternatives Startbild	40
<b>8</b>	<b>Technische Daten</b>	<b>41</b>
<b>Anhang A</b>	<b>CE-Zertifikat</b>	<b>43</b>
<b>Anhang B</b>	<b>Maßzeichnung</b>	<b>44</b>
<b>Anhang C</b>	<b>Struktur auf der Speicherkarte</b>	<b>46</b>
<b>Anhang D</b>	<b>Referenz Definitionsdateien</b>	<b>48</b>
D.1	Szenendefinition (*.ins)	48
D.2	Szenen Projektdatei (*.inp)	80
D.3	CAN-ID-Filter (*.flt)	81
<b>Anhang E</b>	<b>Ausführung mit Gehäuse</b>	<b>84</b>
E.1	Lieferumfang	84
E.2	Drucktasten	85
E.3	Anschlüsse	85
E.4	miniSD-Kartensteckplatz	86
E.5	Technische Daten Gehäuseausführung	88

# 1 Einleitung

Das PCAN-MiniDisplay wird als Mensch-Maschinen-Interface zur Visualisierung von CAN-Daten eingesetzt. Für die Anbindung an CAN-Busse verfügt es über zwei High-Speed-CAN-Anschlüsse. Die grafische Darstellung eingehender CAN-Daten wird vor Einsatz des Geräts konfiguriert und erfolgt dann über ein TFT-Display.

Das PCAN-MiniDisplay kann alternativ als Datenlogger verwendet werden. Der Nachrichtenverkehr wird in diesem Fall auf einer Speicherkarte aufgezeichnet und kann wiedergegeben oder auf einem PC ausgewertet werden.

Neben der Einbauversion des PCAN-MiniDisplays ist auch eine Ausführung im Gehäuse mit Tastern verfügbar. Damit ist es möglich, anwendungsspezifische Konfigurationen zu erstellen und zu evaluieren. Zudem kann der Anwender während des Betriebs direkt zwischen verschiedenen Konfigurationen umschalten.

Die besonderen Merkmale der Ausführung mit Gehäuse und Tastern sind in Anhang E auf Seite 84 beschrieben.



**Hinweis:** Diese Bedienungsanleitung bezieht sich auf Geräte, die ab Juli 2017 ausgeliefert wurden (siehe *Berücksichtigte Produkte* auf Seite 2). Besitzer älterer PCAN-MiniDisplays beachten bitte die Informationen zum Firmware-Update im Kapitel *Interne Statistiken* auf Seite 36.

## 1.1 Eigenschaften im Überblick

- └ CAN-Anbindungen über zwei High-Speed-CAN-Kanäle (ISO 11898-2)
  - Übertragungsraten von 40 kbit/s bis zu 1 Mbit/s
  - Erfüllt die CAN-Spezifikationen 2.0 A/B
  - NXP CAN-Transceiver TJA1041 mit Wake-Up
- └ 2 analoge Eingänge
  - Auflösung 10 Bit
  - Messbereich 0 - 16,5 V
- └ 1 digitaler Eingang und 1 digitaler Ausgang
- └ TFT-Display mit einer Auflösung von 320 x 240 Pixeln
- └ Frei konfigurierbare Visualisierung von CAN-Daten über Text-Dateien:
  - Anzeige von Texten, CAN-Daten sowie den Werten der analogen und digitalen Eingänge über Textlabel
  - Formatierbare Ausgabe von CAN-Daten als Zahl, Gleitkommazahl oder Strings unter Berücksichtigung von Enums und Multiplexern
  - Anzeige von Bitmap-Bildern mit Transparenz
  - Signalabhängige Anzeige von Bildlisten
  - Darstellung mehrerer Signalverläufe in Plottern
  - Konfigurierbarer Zugriff auf die Tasten- und Buzzer-Funktionen sowie die Display-Helligkeit über CAN
  - Senden von CAN-Frame-Sequenzen beim Start, Betrieb und Ende einer Konfiguration
- └ Steckplatz für microSD-Speicherkarte (max. 32 GByte)

- └ High-Speed USB 2.0-Anschluss für Zugriff auf die Speicherkarte mit einem PC
- └ Ausführen der Konfigurationen von der Speicherkarte
- └ Optionaler Autostart einer Konfiguration oder einer Aufzeichnung beim Einschalten
- └ Aufzeichnung eingehender CAN-Nachrichten auf einer Speicherkarte, optional inklusive auftretender Fehler
- └ CAN-ID-Filterung für Aufzeichnungen, gesondert für jeden CAN-Kanal konfigurierbar
- └ Konvertierung der Aufzeichnungen in verschiedene Ausgabeformate über eine Windows-Software
- └ Wiedergabe einer Aufzeichnung mit optionaler Loop-Funktion
- └ Verwendung optional bestückter Drucktaster zur Navigation im Menü und zum Umschalten zwischen mehreren Anzeige-Konfigurationen
- └ Maße: 80 x 55 mm
- └ Spannungsversorgung von 7 bis 30 V
- └ Betriebstemperaturbereich von -20 bis 70 °C

## 1.2 Lieferumfang

- └ PCAN-MiniDisplay mit Einbaurahmen
- └ Kabel für Versorgung, CAN und digitale Eingänge
- └ Installierte 512 MB Speicherkarte
- └ Konvertierungssoftware PEAK-Converter für Windows 10, 8.1, 7 (32/64-Bit)
- └ Konfigurationsbeispiele und Handbuch im PDF-Format

## 2 Anschlüsse

Auf der Platine sind folgende Anschlüsse vorhanden:

- └ J1 Externe Taster (2.1 Seite 9)
- └ J2 System (2.2 Seite 9)
- └ J3 und J7 USB (2.3 Seite 10)
- └ J4 JTAG (2.4 Seite 12)
- └ J5 microSD (2.5 Seite 12)
- └ J6 Seriell (2.6 Seite 13)

Für den grundsätzlichen Betrieb sind die Spannungsversorgung (J2), der CAN-Anschluss (J2) und eine microSD-Speicherkarte (J5) nötig.

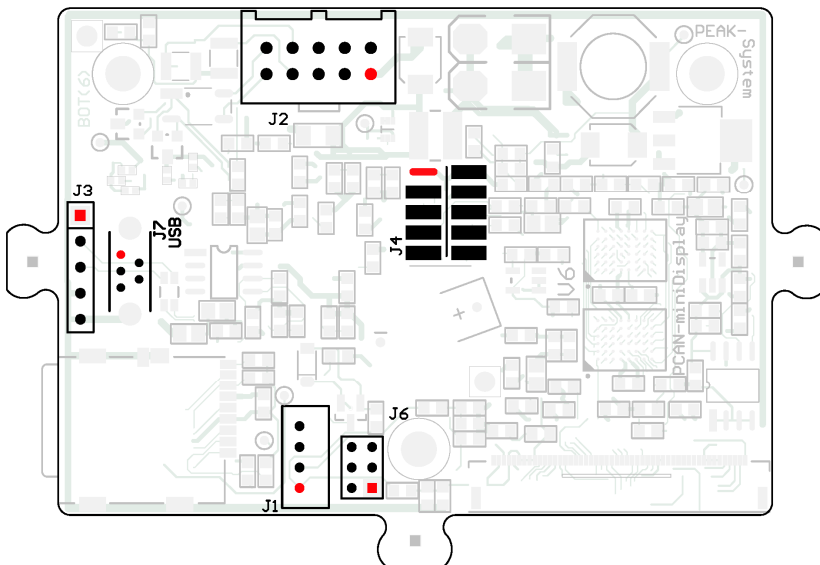


Abbildung 1: Platinenrückseite des PCAN-MiniDisplay;  
Anschlüsse J1, J2, J3, J4, J6 und J7;  
Pin 1 rot markiert



## 2.1 J1 Externe Taster

3 Taster können angeschlossen und für die Navigation durch die Menüs der Benutzeroberfläche des PCAN-MiniDisplay verwendet werden. Dies dient primär Wartungs- und Debugging-Zwecken. Zum Beispiel kann dadurch die Kommunikationskonfiguration oder das CAN-Tracing ermöglicht werden.

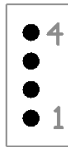


Abb. 2: Pin-Belegung J1 externe Taster

Pin	Funktion
1	Din1: Runter ↓
2	Din2: Hoch ↑
3	Din3: Eingabe ←
4	GND

Die 3 Taster verbinden die Funktionspins (1 bis 3) jeweils mit GND.

## 2.2 J2 System

- Spannungsversorgung
- 2 High-Speed-CAN-Kanäle (CAN1 und CAN 2)
- 1 digitaler Ausgang, 1 digitaler Eingang, 2 analoge Eingänge

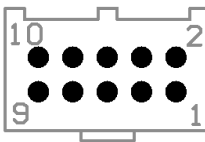


Abbildung 3: Pin-Belegung J2 System

Pin	Funktion
1	Vb (7 - 30 V DC)
2	GND
3	Dout1
4	Din4
5	Ain1
6	Ain2
7	CAN2_L
8	CAN2_H
9	CAN1_L
10	CAN1_H

Die digitalen und analogen Eingänge können für die Instrumentenanzeige genutzt werden. Weitere Informationen zu den Eingängen und dem digitalen Ausgang des J2-System-Anschlusses finden Sie im Abschnitt *Technische Daten* Seite 41.

## 2.3 J3 und J7 USB

Die Anschlüsse J3 und J7 (Mini-USB-Buchse) können für den Zugriff auf die Speicherkarte im PCAN-MiniDisplay von einem PC über eine USB-Verbindung. Der Zugriff auf die Speicherkarte kann auch bei ausgeschaltetem PCAN-MiniDisplay erfolgen. Beim Einschalten des Geräts wird die USB-Verbindung kurzzeitig unterbrochen.

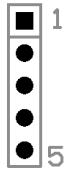


Abbildung 4: Pin-Belegung J3 USB

Pin	Funktion
1	5 V USB
2	D-
3	D+
4	GND
5	GND

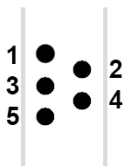


Abbildung 5: Pin-Belegung J7 USB (Mini-USB)

Pin	Funktion
1	5 V USB
2	D-
3	D+
4	nicht belegt
5	GND

### 2.3.1 USB-Verbindung trennen

Vor dem Abziehen des USB-Kabels am PC oder am PCAN-MiniDisplay sollte das Gerät im Betriebssystem abgemeldet werden. Dieser Vorgang stellt sicher, dass das Betriebssystem Schreibvorgänge auf die Speicherkarte des PCAN-MiniDisplays korrekt abgeschlossen hat.



Abbildung 6: Windows-Icon **Hardware sicher entfernen** im Infobereich der Taskleiste

### 2.3.2 Einschränkung der PCAN-MiniDisplay-Funktionen

Während einer USB-Verbindung zu einem PC stehen manche Funktionen des PCAN-MiniDisplays nur eingeschränkt zur Verfügung, da das Gerät nicht gleichzeitig mit dem verbundenen PC auf die Speicherkarte zugreifen kann:

- ┌ Szenendefinitionen (Instrumente) können nicht geladen werden.
- ┌ Die Aufzeichnung und die Wiedergabe von CAN-Verkehr (Trace) funktionieren nicht.
- ┌ Die Befehle im Menü **Memory Card** funktionieren nicht.

## 2.4 J4 JTAG

Zugriff auf den Mikrocontroller zu Debugging-Zwecken.



Abbildung 7: Pin-Belegung J4 JTAG

Pin	Funktion
1	GND
2	GND
3	Reset#
4	3,3 V
5	TCK
6	TMS
7	TDO
8	TDI
9	RTCK
10	TRST

## 2.5 J5 microSD

Steckplatz für eine microSD-Speicherkarte (max. 32 GByte).

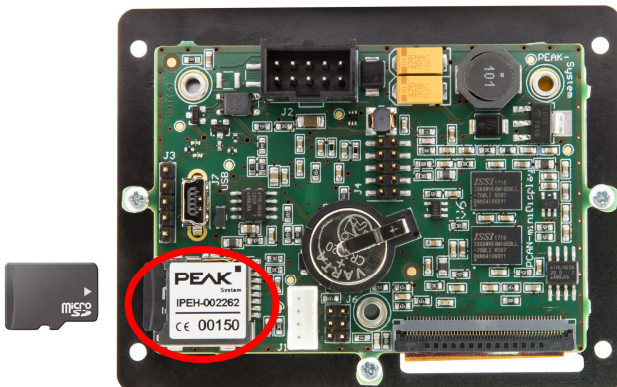


Abbildung 8: Position des microSD-Steckplatzes auf der Rückseite der Platine des PCAN-MiniDisplay

## 2.6 J6 Seriell

Stellt serielle Signale für kundenspezifische Erweiterungen zur Verfügung.

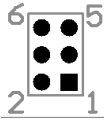


Abbildung 9: Anschlussbelegung J6  
Seriell

Pin	Funktion
1	GND
2	5 V
3	TxD RS-232
4	RxD RS-232
5	DBG
6	Buzzer



**Hinweis:** Die seriellen Pins RxD und TxD werden von der Standard-Firmware nicht unterstützt.

## 3 Szenendefinition

Das PCAN-MiniDisplay verwendet eine so genannte Szene für die Anzeige von CAN-Daten in grafischer Form.



**Hinweis:** Das Ausführen einer Anzeigen-Konfiguration sowie die Aufzeichnung und Wiedergabe von CAN-Nachrichten können nicht gleichzeitig erfolgen. Das PCAN-MiniDisplay kann nur mit einer der drei Funktionen eingesetzt werden.

### 3.1 Allgemeine Beschreibung einer \*.ins-Datei

Eine Szene ist eine Sammlung von Instrumenten, Beschriftungen, Schriftarten, etc. definiert in einer Textdatei mit der Erweiterung \*.ins. Zu Szenen gehören des Weiteren Bitmap-Grafik-Dateien (\*.bmp) und Font-Dateien (\*.fon).

Der Inhalt einer Szene-Definitionsdatei ist in verschiedene Abschnitte unterteilt jeweils mit Definitionseinträgen:

- └ [global]: allgemeine Angaben für die Szene
- └ [fontX]: Schriftarten, die von Labeln verwendet werden
- └ [messageX]: Nachrichten für die Verwendung durch Variablen, Multiplexer-Variablen und für die Zuweisung des CAN-Kanals.
- └ [enumX]: Enumeration-Wertelisten für eine benutzerdefinierte Ausgabe bestimmte Werte von Variablen.
- └ [variableX]: Variablen zur Repräsentation der Daten von CAN-Nachrichten oder der analogen und digitalen Eingänge. Ausgabe mit Labeln, dem Plotter oder Instrumenten.
- └ [txlistX]: CAN-Frame-Sequenzen, die beim Start, Betrieb und Ende einer Konfiguration gesendet werden können
- └ [labelX]: Textausgaben von festen Zeichenketten oder Variablen

- └ [instrumentX]: Bitmaps und Bitmap-Listen basierte Instrumente für die Präsentation von Variablen
- └ [plotterX]: Linienschreiber zur Darstellung von Signalverläufen

Eine Referenz auf die Schlüsselwörter, die in diesem Abschnitt verwendet werden, ist im Kapitel D.1 auf Seite 48 zu finden. Eine Beispieldatei wird folgend gezeigt.

Eine Definitionsdatei kann Kommentare enthalten, die durch doppelte Schrägstriche (//) eingeleitet werden. Alle Zeichen in der Zeile nach einem doppelten Schrägstrich werden ignoriert, wenn die \*.ins-Datei von dem PCAN-MiniDisplay interpretiert wird.

Während des Ladens erzeugt das PCAN-MiniDisplay eine binäre Version einer Definition (\*.inb). Diese wird bei zukünftigen Verwendungen viel schneller geladen, weil alle Konvertierungen für Panel-interne Nutzung bereits integriert sind. Dies ist das Standardverhalten, kann aber in den globalen Einstellungen konfiguriert werden.

Änderungen der Definition einer Szene können nur in der textbasierten Datei durchgeführt werden.



**Hinweis:** Wenn die Definition einer Szene geändert wurde, muss die vorher erzeugte Binärdatei (\*.inb) manuell auf der Speicherkarte des PCAN-MiniDisplay gelöscht werden. Dadurch sieht das Ladeprogramm, dass eine neue binäre Datei aus der Textdatei erstellt werden muss. Wenn Sie die binäre Datei nicht löschen, werden Ihre Änderungen in der Definitionsdatei keine Auswirkungen haben.

## 3.2 Beispiel einer \*.ins-Datei

Auf der Speicherkarte oder im Download-Package des PCAN-MiniDisplay finden Sie im Verzeichnis Scenes Beispiele für die vielfältigen Konfigurations-Möglichkeiten.

Das folgende, einfache Beispiel definiert:

- └ die CAN-Nachricht „ExampleMsg“ mit der ID 100h
- └ die Variable „Speed“ für die ersten 8 Bit der übertragenen Daten
- └ zwei Label zur Ausgabe des Variablenwertes
- └ die Anzeige eines Hintergrundbilds mit einem Instrument

```
[global]
format=1
compress=1           // 1 => Die Szene wird nach dem ersten Laden komprimiert
instruments=1        // Anzahl der in dieser Szene definierten Instrumente
messages=1           // Anzahl der in dieser Szene definierten CAN-Nachrichten
variables=1          // Anzahl der in dieser Szene definierten Variablen
labels=2             // Anzahl der in dieser Szene definierten Label

[message1]
enable=1             // 1 => Aktiviert die Auswertung dieser Nachricht
Name = "ExampleMsg"
canid=0x100          // ID von dieser CAN-Nachricht
frametype=0          // 0 => 11 Bit, 1 => 29 Bit
channel=0            // 0 => CAN-Kanal 1, 1 => CAN-Kanal 2
timeout=0           // 1 => Aktiviert die Funktion "timeout"

[variable1]
name="Speed"
messagename="ExampleMsg"
position=0,8         // Bitstart- und -länge
byteorder=0          // 0 => Intel, 1 => Motorola
muxtype=0            // 0 => None, 1 => Multiplexer, 2 => Multiplexed
muxval=0             // Wenn "muxtype=2", dann wird diese Variable
                    // aktualisiert, wenn der Multiplexer (muxtype=1)
                    // diesen Wert hat.

scale=1
offset=0
vartype=0           // Darstellung der CAN-Daten: 0 => unsigned,
                    // 1 => signiert, 2 => Gleitkommazahl,
                    // 4=> String, 5=> Enum

datatype=0          // Darstellung des Ergebnisses: 0 => unsigned,
                    // 1 => signiert, 2 => Gleitkommazahl,
                    // 4=> String, 5=> Enum
```



```
[label1]
name="label1"           // Name des Labels als Referenz.
position=10,20         // X-Position,Y-Position in Pixel; 0,0 ist die
                        // oberere linke Ecke
font_idx=0             // Festlegung der Schrift
length=15              // Anzahl der Zeichen
initval="Current Speed: "
fontcolor=0,175,0     // RGB-Wert
bgcolor=0,0,0         // RGB-Wert

[label2]
name="label2"           // Name des Labels als Referenz
position=140,20        // X-Position,Y-Position in Pixel; 0,0 ist die
                        // oberere linke Ecke
font_idx=0             // Festlegung der Schrift
fmt_string="%3d km/h" // Formatdefinition
var_name="Speed"       // Verweis auf die Variable
length=8               // Anzahl der Zeichen
initval "--- km/h"
fontcolor=0,255,0     // RGB-Wert
bgcolor=0,0,0         // RGB-Wert
update_rate=0

[instrument1]
pictures=1
name="Background"
no_restore=1
0="Background_320x240.bmp", "", 0,0,0,0
```



**Hinweis:** Die Textdatei (\*.ins) muss eine Leerzeile am Ende haben.

### 3.3 Verwendung mehrerer Szenen

Es ist möglich, mehr als eine Szene auf dem PCAN-MiniDisplay zu verwenden. Szenen können mit Hilfe der Digitaleingänge umgeschaltet werden (z.B. mit Drucktasten).

Um dies zu erreichen, muss eine Liste der Szene-Definitionsdatei in einer Szenen-Projektdatei (\*.inp) erstellt werden. Diese Liste definiert auch, welche Szene beim Start verwendet wird.

Eine Referenz auf die Schlüsselwörter, die in diesem Abschnitt verwendet werden, ist im Kapitel D.2 Seite 80 zu finden. Eine Beispieldatei wird folgend gezeigt.

```
[general]
version=1

[scenes]
scenes=4
startscene=2
1="speedo0.ins"
2="speedo1.ins"
3="picture1.ins"
4="picture2.ins"
```



**Hinweis:** Die Textdatei (\*.inp) muss eine Leerzeile am Ende haben.

## 4 Schritte zur Inbetriebnahme

Beim Einschalten sucht das PCAN-MiniDisplay nach den intern gespeicherten Startoptionen. Eine Szene für die Anzeige oder der Tracer für CAN-Aufzeichnungen kann automatisch gestartet werden.



**Tipp:** Bei Auslieferung ist das PCAN-MiniDisplay so konfiguriert, dass die folgende Szene gestartet wird:

```
/MiniDisplay/Scenes/Default/Default.inp
```

Dieses Kapitel beschreibt die Inbetriebnahme des PCAN-MiniDisplays Schritt für Schritt. Bitte gehen Sie jedes der folgenden Kapitel durch.

### 4.1 Voraussetzungen

- └─ microSD-Speicherkarte, max. 32 GByte (512-MByte-Karte liegt bei)
- └─ PC mit SD-Kartenleser (inklusive Adapter für microSD-Karte) oder USB-Verbindung PC zum PCAN-MiniDisplay (an J3 oder J7 angeschlossen)
- └─ Stromversorgung 7 - 30 V DC (an J2 angeschlossen)

### 4.2 Speicherkarte vorbereiten

Damit das Gerät korrekt startet, muss die Speicherkarte eine bestimmte Verzeichnisstruktur enthalten. Diese ist bei Lieferung vorhanden. Sie können Ihre eigenen Szenen der Verzeichnisstruktur hinzuzufügen.

▶ So bereiten Sie die Speicherkarte mit Ihren Szenen vor:

1. Stellen Sie eine USB-Verbindung zwischen dem PCAN-MiniDisplay und Ihrem PC her, um Zugriff auf die microSD-Speicherkarte zubekommen.

Alternativ nehmen Sie die Karte aus dem PCAN-MiniDisplay und stecken sie in einen Kartenleser, der an Ihrem PC angeschlossen ist.

Die Speicherkarte erscheint als Massenspeichergerät.

2. Kopieren Sie eine oder mehrere Szenen, die Sie zuvor erstellt haben (siehe Kapitel 3 *Szenendefinition* auf Seite 14) in das folgende Verzeichnis der Speicherkarte:

`/MiniDisplay/Scenes/`

Jede Szene muss ein eigenes Unterverzeichnis haben.

3. Melden Sie die USB-Verbindung vom Betriebssystem des PCs ab und ziehen Sie das USB-Kabel heraus.

Wenn die Speicherkarte in einem Kartenleser vorbereitet wurde, entfernen Sie die microSD-Speicherkarte aus dem Kartenleser und setzen Sie die Karte in den Steckplatz des PCAN-MiniDisplays.

► So stellen Sie bei Bedarf die Verzeichnis-Struktur auf der Karte wieder her:

1. Stellen Sie eine USB-Verbindung zwischen dem PCAN-MiniDisplay und Ihrem PC her, um Zugriff auf die microSD-Speicherkarte zubekommen.

Alternativ nehmen Sie die Karte aus dem PCAN-MiniDisplay und stecken sie in einen Kartenleser, der an Ihrem PC angeschlossen ist.

Die Speicherkarte erscheint als Massenspeichergerät.

2. Formatieren Sie die Speicherkarte als FAT32-Dateisystem.
3. Laden Sie das PCAN-MiniDisplay-Download-Package von unserer Website herunter.
4. Kopieren Sie das enthaltene Verzeichnis `MiniDisplay` inklusive aller Unterverzeichnisse und Dateien in das Root-Verzeichnis der Speicherkarte.

Ergänzende Informationen finden Sie unter Anhang C *Struktur auf der Speicherkarte* auf Seite 46.

5. Melden Sie die USB-Verbindung vom Betriebssystem des PCs ab und ziehen Sie das USB-Kabel heraus.

Wenn die Speicherkarte in einem Kartenleser vorbereitet wurde, entfernen Sie die microSD-Speicherkarte aus dem Kartenleser und setzen Sie die Karte in den Steckplatz des PCAN-MiniDisplays.

## 4.3 Einschalten des Geräts

Das PCAN-MiniDisplay wird durch Anlegen einer Versorgungsspannung im Bereich von 7 bis 30 V DC an Anschluss J2 eingeschaltet. Das Gerät startet automatisch. Wenn nach der Lieferung nichts verändert wurde, zeigt das PCAN-MiniDisplay die Szene mit dem Namen `Default.inp`.

Wird das Gerät über den Menüeintrag **Power off** ausgeschaltet, kann es über Pin 1 des J1-Anschlusses (siehe Kapitel *J1 Externe Taster* auf Seite 9) wieder eingeschaltet werden. Beim PCAN-MiniDisplay mit Gehäuse drücken Sie einfach die Runter-Taste ↓.

## 4.4 Menüs

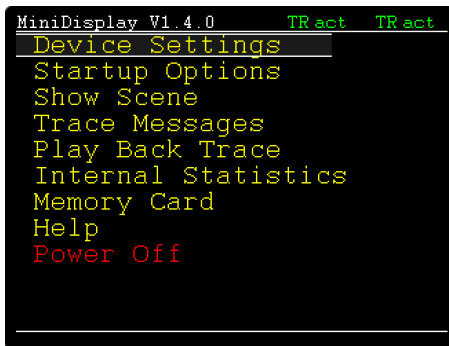


Abbildung 10: Hauptmenü

Das PCAN-MiniDisplay bietet eine Benutzeroberfläche für den Zugriff auf die einzelnen Funktionen und die Konfiguration der Einstellungen.

Die Navigation erfolgt über die digitalen Eingänge des Steckers J1, welche zum Beispiel mit Drucktasten ausgerüstet werden können. Details über die Verbindung siehe Kapitel 2.1 auf Seite 9.

Das PCAN-MiniDisplay Case ist mit den entsprechenden Tasten bestückt. In diesem Fall erfolgt das Navigieren durch die Menüs und Funktionen mit den Tasten Hoch ↑ und Runter ↓. Ein ausgewähltes Element wird mit der Eingabetaste ↵ aktiviert.



Während der Anzeige einer Szene bringt Sie das Drücken der Eingabetaste ↵ zurück in das Hauptmenü.



**Tipp:** Die Verwendung des Menüs wird für die reine Anzeige einer Szene nicht benötigt.

#### 4.4.1 Statusanzeige


In der Benutzeroberfläche zeigen Icons rechts oben auf dem Display den Status der CAN-Kommunikation für die CAN-Kanäle 1 und 2 (High-Speed-CAN) an.

Icon	Bedeutung
	CAN-Verkehr: T = Transmit (Senden), R = Receive (Empfangen) Blinken: Aus-/eingehende CAN-Nachrichten Grün: Fehlerfreies Senden/Empfangen Gelb, rot: Fehler beim Senden/Empfangen
	Informiert über den Status des CAN-Controllers ( <b>active</b> , <b>passive</b> , <b>bus-off</b> ). Informiert über den Status des CAN-Controllers ( <b>active</b> , <b>passive</b> , <b>bus-off</b> ). Wenn aufgrund zu vieler Sende-Fehler der Bus-Off-Zustand erreicht wurde, werden keine CAN-Nachrichten mehr empfangen oder gesendet. Das PCAN-MiniDisplay setzt den CAN-Controller nach einer bestimmten Zeit automatisch zurück.

## 4.5 Einstellen der Startoptionen

Über den Hauptmenü-Eintrag **Startup Options** können Sie das Verhalten beim Einschalten einrichten.

▶ So zeigen Sie eine Szene beim Start an:

 **Hinweis:** Bei Auslieferung ist das PCAN-MiniDisplay so konfiguriert, dass die Szene mit Namen `Default.inp` gestartet wird.

1. Setzen Sie den Marker  nach **Start scene**.
2. Klicken Sie auf **..** nach **Select scene**.
3. Wählen Sie eine Szene (\*.ins) oder ein Projekt (\*.inp), welche beim Start angezeigt werden soll.

Die ausgewählte Datei wird neben **Select scene** **..** angezeigt.

4. Klicken Sie auf **OK**.

▶ So können Sie den Tracer beim Einschalten starten:

1. Setzen Sie den Marker  nach **Start tracer**.
2. Bei **Tracer timeout** geben Sie optional den Zeitraum zum Stoppen des Tracers ein, wenn keine CAN-Nachrichten empfangen werden. Der Eintrag **Never** lässt den Tracer weiterlaufen.
3. Klicken Sie auf **OK**.



## 5 CAN-Tracing

### 5.1 CAN-Verkehr aufzeichnen



**Hinweis:** Die Aufzeichnung oder Wiedergabe von CAN-Nachrichten ist nicht gleichzeitig mit der Anzeige einer Szene möglich.

➤ Menüpunkt **Trace Messages**

Mit dieser Funktion wird der gesamte eingehende CAN-Verkehr inklusive RTR-Frames und Error-Frames auf der Speicherkarte des PCAN-MiniDisplay aufgezeichnet. Dabei wird auch die zeitliche Abfolge berücksichtigt.

**Trace CAN 1 / Trace CAN 2**

Mit diesen Einstellungen werden die Informationen, die aufgezeichnet werden sollen, ausgewählt. Folgende Optionen stehen zur Auswahl:

- └ **Off** (keine Aufzeichnung)
- └ **Data** (CAN-Daten)
- └ **Errors** (Error-Frames)
- └ **Data + Err.** (CAN-Daten und Error-Frames)

**Trace file size**

Diese Option wird verwendet, um die maximale Größe einer Aufzeichnung auf einen Wert von 10 Mbyte bis zu 1 GByte zu begrenzen.

### Trace timeout

Wenn innerhalb dieser Zeitspanne keine CAN-Nachricht empfangen wird, versetzt sich das Gerät in den Sleep-Modus. Zum erneuten Starten des Gerätes ist ein Tastendruck oder ein Wake-Up über den CAN-Bus erforderlich. Der Eintrag **Never** lässt den Tracer weiterlaufen.

 **Hinweis:** Der Sleep-Modus wird nicht bei Szenen unterstützt.

### Multi file trace

Ist diese Option aktiviert, wird die Aufzeichnung in mehreren Dateien gespeichert. Ist die maximale Größe der Trace-Datei **Trace file size** erreicht, wird die Aufzeichnung in einer neuen Datei fortgesetzt. Der unter **File** angegebene Dateiname wird dafür um eine fortlaufende Nummer ergänzt. Ist die Option deaktiviert, endet die Aufzeichnung bei Erreichen der maximalen Datei-Größe.

### Trace type

Diese Option legt den Typ der Aufnahme fest. Folgende Optionen stehen zur Auswahl:

- **linear** Die Aufzeichnung ist fortlaufend. Jeder neuen Trace-Datei wird ein zweistelliger Index von 0 bis 99 angefügt.
- **circular** Es wird eine bestimmte Anzahl von Trace-Dateien bei der Aufzeichnung erzeugt. Ist die maximale Zahl der Dateien erreicht, wird die älteste gelöscht und der Index aller übrigen Dateien um 1 reduziert. Abschließend wird eine neue Datei erzeugt, in der die Aufzeichnung fortgeführt wird.

### **File count circular**

Dieser Wert gibt die maximale Anzahl der Trace-Dateien an, die bei einer Aufzeichnung vom **Trace type circular** erstellt werden.

### **Session type**

Mit dieser Einstellung wird festgelegt, ob die bei einer Aufzeichnung vom **Trace type circular** erstellten Dateien überschrieben werden oder nicht. Folgende Optionen stehen zur Auswahl:

- └ **Single session** Mit jeder Aufzeichnung wird die Datei-Sequenz mit dem Namen `trace_xx.btr` überschrieben.
- └ **Multi session** Mit jeder Aufzeichnung wird eine neue Sequenz von Dateien erzeugt.

Eine Trace-Datei kann später zur Wiedergabe der aufgezeichneten CAN-Nachrichten auf dem CAN-Bus verwendet werden (siehe Abschnitt 5.2 Seite 29). Alternativ ist auf einem PC die Konvertierung der Aufzeichnung in ein anderes Format zur Weiterverwendung und Auswertung möglich (siehe Abschnitt 5.3 Seite 30).



**Hinweis:** Es ist möglich, einen Filter für den eingehenden CAN-Verkehr zu verwenden. Damit werden nur CAN-Nachrichten mit spezifischen IDs durchgelassen. Information über Filterung siehe Kapitel 5.4 auf Seite 32.

► So führen Sie eine Aufzeichnung durch:

1. Stellen Sie sicher, dass keine USB-Verbindung zwischen dem PCAN-MiniDisplay und Ihrem PC besteht.
2. Drücken Sie **Start**.

Die Daten werden in die bei **File** angezeigte Datei geschrieben.

3. Beenden Sie die Aufzeichnung mit **Stop tracing**.

Anzeige	Bedeutung
<b>File</b>	Name der Trace-Datei für die aktuelle Aufzeichnung. Der Dateiname ( <code>trc00000.btr</code> ) wird automatisch mit einer fortlaufenden Nummer erzeugt. Bei einer Aufzeichnung mit mehreren Dateien wird ein weiterer zweistelliger Index angefügt.
<b>Start time</b>	Gibt den Start-Zeitpunkt der Aufzeichnung an.
<b>CAN queue level</b>	Aktueller und maximaler Füllstand der Empfangswarteschlange (letzterer in Klammern). Falls die Warteschlange einen Füllstand von 100 Prozent erreicht hat, wurden mit großer Wahrscheinlichkeit einige eingehende CAN-Nachrichten nicht aufgezeichnet.
<b>CAN messages total</b>	Anzahl der CAN-Nachrichten, die bisher aufgenommen wurden.
<b>File size</b>	Aktuelle Größe der Trace-Datei in kByte und bereits benutzter Speicher relativ zum zur Verfügung stehenden Speicherplatz.  Während der Aufzeichnung wächst die Trace-Datei in 512-Byte-Blöcken, die jeweils 25 CAN-Nachrichten enthalten. In 1 MByte passen demnach 51200 CAN-Nachrichten.
<b>File count</b>	Anzahl der Dateien, die aufgenommen wurden. Bei einer Aufzeichnung vom <b>Trace type</b> <b>circular</b> wird die Anzahl unabhängig vom aktuellen Index linear hochgezählt.
<b>Trace type</b>	Zeigt den verwendeten Trace type an.
<b>Session type</b>	Zeigt den verwendeten Session type an.

## 5.2 Aufgezeichneten CAN-Verkehr wiedergeben

➤ Menüpunkt **Play Back Trace**

Das PCAN-MiniDisplay kann die CAN-Nachrichten aus einer Binary-Trace-Datei (\*.btr) auf dem angeschlossenen CAN-Bus wiedergeben. Dabei wird die zeitliche Abfolge der CAN-Nachrichten, wie sie ursprünglich bei der Aufzeichnung der Trace-Datei auftrat, beibehalten.

▶ So geben Sie eine Trace-Datei wieder:

1. Stellen Sie sicher, dass keine USB-Verbindung zwischen dem PCAN-MiniDisplay und Ihrem PC besteht.
2. Wählen Sie im Hauptmenü **Play Back Trace** aus.
3. Verwenden Sie **Select file** und wählen Sie die Trace-Datei (\*.btr) für die Wiedergabe aus der Liste.
4. Falls Sie die Trace-Datei nicht wiederholt abspielen möchten, ändern Sie **Loop mode** von **infinite** auf die gewünschte Anzahl von Zyklen (**1** bis **9**).
5. Falls Sie eine Trace-Datei verwenden, die CAN-Nachrichten von mehreren CAN-Kanälen enthält, wählen Sie die CAN-Kanäle **Channel 1 / Channel 2**, die für die Trace-Wiedergabe verwendet werden sollen.
6. Klicken Sie auf **Play**, um die Wiedergabe zu starten.
7. Sie können die Wiedergabe auch unterbrechen (**Pause playback**). Sie haben dann die folgenden Möglichkeiten:

Funktion	Durchgeführte Aktion
<b>Exit</b>	Beendet die Wiedergabe.
<b>Restart</b>	Beginnt die Wiedergabe erneut am Anfang der Trace-Datei.

Funktion	Durchgeführte Aktion
<b>Reset</b>	Setzt den CAN-Controller zurück. Nützlich nach der Fehlerbeseitigung am CAN-Bus.
<b>Continue</b>	Setzt die Wiedergabe an der Stelle fort, wo zuvor unterbrochen wurde.

### 5.3 Aufgezeichneten CAN-Verkehr auf dem PC verwenden

Der aufgezeichnete CAN-Verkehr kann mit einem PC über eine USB-Verbindung von der Speicherkarte des PCAN-MiniDisplay geladen werden. Er befindet sich in binärkodierten Trace-Dateien \*.btr im folgenden Verzeichnis:

/MiniDisplay/Traces/

Für die Weiterverwendung müssen Sie die Daten in ein geeignetes Format umwandeln. Dazu steht auf der Speicherkarte des PCAN-MiniDisplay die Windows-Software PEAK-Converter zur Verfügung.

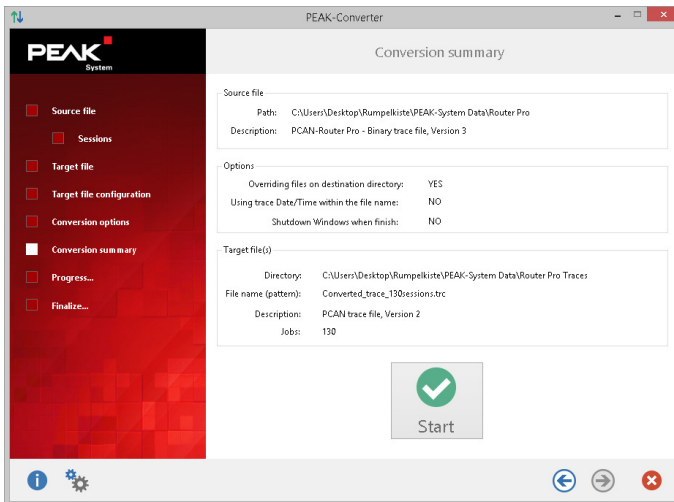


Abbildung 11: Benutzeroberfläche des PEAK-Converter

### Mögliche Konvertierungsziele:

Zielformat	Datei- endung	Erläuterung/Verwendungsmöglichkeiten
PCAN-Trace	.trc	Textbasiertes Trace-Format von PEAK-System; Betrachtung und Wiedergabe der CAN-Nachrichten in der Software PCAN-Explorer oder PCAN-Trace.
Vector ASC Trace	.asc	Textbasiertes Trace-Format der Firma Vector, das auch von anderen Drittanbieterprogrammen genutzt werden kann.
Character Separated Values (CSV)	.csv	Allgemeines, textbasiertes Format zum Import in eine Tabellenkalkulation (Semikolon als Separator).

### ➤ So können Sie Trace-Daten weiterverwenden:

1. Verbinden Sie das PCAN-MiniDisplay per USB-Kabel mit dem PC. Das PCAN-MiniDisplay braucht dazu nicht eingeschaltet zu sein.
2. Starten Sie unter Windows von der Speicherkarte des PCAN-MiniDisplay das Konvertierungsprogramm `PEAK-Converter.exe` aus dem Verzeichnis `/MiniDisplay/Tools`.
3. Wählen Sie als Quelle (Source) eine Trace-Datei (`*.btr`). Die Trace-Dateien finden Sie im folgenden Verzeichnis: `/MiniDisplay/Traces/`
4. Geben Sie eine Zieldatei an (Destination) und wählen Sie das gewünschte Zielformat.

## 5.4 Filtern des CAN-Verkehrs bei Aufzeichnungen

Bei der Ausführung einer Szene oder der Aufzeichnung einer Trace-Datei wird der gesamte CAN-Verkehr verarbeitet. Es ist jedoch möglich, die eingehenden CAN-Nachrichten zu filtern.

Dafür wird eine Textdatei mit einer Liste der zu filternden CAN-IDs angelegt. Die aufgeführten IDs passieren den Filter. Alle anderen nicht. Die Filter-Datei wird auf der Speicherkarte abgelegt und über die Geräteeinstellungen des PCAN-MiniDisplay geladen.

► So wenden Sie einen CAN-ID-Filter an:

1. Erstellen Sie eine Filter-Datei mit der Erweiterung `*.flt` mit einem Texteditor. Das Format der Datei ist im Anhang D.3 auf Seite 81 beschrieben.
2. Listen Sie die CAN-IDs auf, die den Filter passieren können.
3. Verbinden Sie das PCAN-MiniDisplay per USB-Kabel mit dem PC. Das PCAN-MiniDisplay braucht dazu nicht eingeschaltet zu sein.
4. Kopieren Sie die Filter-Datei auf die Speicherkarte in das Verzeichnis `/MiniDisplay/Filter/`.
5. Navigieren Sie auf dem PCAN-MiniDisplay zu den **Device Settings**. Wählen Sie **Ch.1 Filter** für CAN-Kanal 1 und **Ch.2 Filter** für CAN-Kanal 2.
6. Wählen Sie die zuvor erstellte Filter-Datei (`*.flt`).
7. Verwenden Sie **OK** oder **Save & OK**, um die geänderten Einstellungen zu übernehmen.



## 6 Einstellungen und Wartung

### 6.1 Device Settings

Hier legen Sie die Einstellungen für die Anbindung an einen CAN-Bus und jene für die Benutzung des Geräts fest.

Wenn Sie Einstellungen geändert haben, speichern Sie diese dauerhaft mit **Save&OK** ab. Falls Sie nur vorübergehend (bis zum nächsten Ausschalten des Geräts) mit den geänderten Einstellungen arbeiten möchten, wählen Sie **OK**. Beim nächsten Einschalten des Geräts sind wieder die vorherigen Einstellungen aktiv.

#### 6.1.1 Ch.1/2 Detect CAN bitrate

Falls die Übertragungsrate des angeschlossenen CAN-Busses nicht bekannt ist, besteht die Möglichkeit, diese durch das PCAN-MiniDisplay automatisch erkennen zu lassen. Dazu muss Datenverkehr auf dem CAN-Bus vorhanden sein.

Es werden Übertragungsraten aus der folgenden Reihe erkannt (kbit/s): 1000; 800; 500; 250; 200; 125; 100; 95,2; 83,3; 50,0; 47,6; 33,3; 20,0; 10,0.

#### 6.1.2 Ch.1/2 CAN bitrate

Manuelle Auswahl der CAN-Übertragungsraten. Neben den festen Übertragungsratenwerten enthält die Liste acht benutzerdefinierte Übertragungsraten. Diese werden in der Einstellung **User CAN bitrates** verwaltet.

### 6.1.3 Ch.1/2 Listen-only mode

Soll das Gerät den Datenverkehr auf dem CAN-Bus nicht beeinflussen, also als reiner Beobachter eingesetzt werden, muss der Beobachtungsmodus (Listen-Only-Modus) aktiviert werden (**On**). So antwortet das Gerät nicht auf eingehende CAN-Nachrichten mit einer Bestätigung (Acknowledge) und sendet keine CAN-Nachrichten oder Error-Frames.

### 6.1.4 Ch.1/2 Filter

CAN-ID-Filterung wird nur für Aufzeichnungen verwendet und kann unabhängig für jeden CAN-Kanal eingerichtet werden, indem eine Filterdatei (\*.flt) von der Speicherkarte ausgewählt wird. Die gewählte Datei wird neben dem Menüpunkt angezeigt und für den entsprechenden CAN-Kanal angewendet.

Um die CAN-ID-Filterung zu **deaktivieren**, verwenden Sie **Exit** im Bildschirm zur Dateiauswahl. **None** wird dann neben dem Menüpunkt angezeigt.

Mehr Information über Filterung befindet sich im Kapitel 5.4 auf Seite 32.

### 6.1.5 User CAN bitrates

Für eine Anpassung an spezifische Verhältnisse auf dem CAN-Bus besteht für Experten die Möglichkeit, die Bus-Timing-Register (BTR) des integrierten CAN-Controllers direkt zu beeinflussen. Die RegisterEinstellungen verhalten sich analog zu denen des CAN-Controllers SJA1000 bei 16 MHz Taktfrequenz.

Es können acht benutzerdefinierte Einträge editiert werden. Jeder Eintrag enthält einen 2-Byte-Wert (4 Hexadezimalstellen) für die Bus-Timing-Register und einen beliebigen Bezeichner. Benutzerdefinierte Übertragungsraten erscheinen später mit dem Bezeichner

in der Liste **CAN bitrate** unterhalb der festen Übertragungsratenwerte.



**Tip:** Mit dem Bit Rate Calculation Tool von PEAK-System können Sie die Registerwerte für Ihre Bitrate ermitteln. Die Software ist für Android, iOS und Windows kostenfrei auf unserer Website erhältlich.

Die Tabelle zum Editieren zeigt für jeden Eintrag die Parameter, die sich aus den angegebenen Registerwerten ergeben: die Übertragungsrate (Bitrate), den Sample Point (SP) und die Synchronisierungssprungweite (SJW).

Um einen Eintrag zurückzusetzen (Name = UserX, Bitrate = undef), stellen Sie den BTR-Wert auf 0000 ein.

### 6.1.6 Beeper

Das PCAN-MiniDisplay kann zu diversen Ereignissen Signaltöne ausgeben. Unter anderem wird eine Änderung des CAN-Bus-Status signalisiert. Bei **Off** gibt das Gerät keine akustischen Signale aus.

### 6.1.7 Display rotation

Wenn diese Option aktiviert ist (**On**), wird die Bildausgabe um 180 Grad gedreht.

### 6.1.8 Screensaver timeout

Die Helligkeit des Bildschirms wird nach der gewählten Dauer der Nichtbenutzung reduziert. Der Stromverbrauch wird damit reduziert.

### 6.1.9 Show startscreen

Nach dem Einschalten zeigt das PCAN-MiniDisplay eine Bitmap-Datei für ein paar Sekunden als Startbildschirm an. Dieses Verhalten kann hier ein- und ausgeschaltet werden.

Die Bitmap-Datei kann durch eine eigene Grafik ersetzt werden. Mehr Informationen dazu finden Sie im Abschnitt 7.1 auf Seite 40.

### 6.1.10 Date & time

Über **Set** können das Datum und die Uhrzeit des Geräts eingestellt werden. Datum und Uhrzeit werden beim Speichern von Dateien auf der Speicherkarte verwendet.


### 6.1.11 Reset file index

Dateinamen der zu speichernden Traces erhalten eine Nummer von einem Zähler. Der aktuelle Zählerstand wird in Klammern angezeigt und kann durch Klicken auf **Reset** auf 0 zurückgesetzt werden.

## 6.2 Interne Statistiken


Die Seite gibt eine Übersicht über Geräteinterna. Die einzelnen Angaben werden in der Regel zu Supportzwecken benötigt. Auf dem Bildschirm wird auch die Seriennummer des PCAN-MiniDisplay angezeigt.

Außerdem stehen hardwarenahe Aktionen zur Verfügung, mit denen das Gerät gewartet werden kann. Diese sind im Folgenden kurz erläutert.

 **Wichtiger Hinweis:** Die falsche Anwendung dieser Funktionen kann zur Unbenutzbarkeit des Gerätes führen.

## Update Firmware

Dateien für Firmware-Updates (\*.bin) werden auf der Speicherkarte im Verzeichnis /MiniDisplay/Firmware/ abgelegt. Der Updatevorgang wird durch Auswahl einer Firmware-Datei gestartet.

 **Wichtiger Hinweis:** Ältere PCAN-MiniDisplays ohne Gehäuse (IPEH-002262, vor Seriennummer 150) und mit Gehäuse (IPEH-002262-KSM01, vor Seriennummer 12215) müssen erst auf die Firmware 1.1.7 aktualisiert werden, bevor diese auf die Firmware 1.2.1 oder höher geupdatet werden können.

## Factory Defaults


Alle Einstellungen werden auf den Standardzustand zurückgesetzt, der durch die aktuelle Firmware vorgegeben ist.

## Bootloader

Startet den Bootloader für ein Firmware-Update per CAN.

## 6.3 Memory Card

Verzeichnisse und Bitmaps auf der Speicherkarte können mit dem PCAN-MiniDisplay angezeigt werden.

 **Hinweis:** Das PCAN-MiniDisplay kann nicht auf die Speicherkarte zugreifen, solange eine USB-Verbindung zu einem PC besteht.

## Show Directory

Anzeige der Verzeichnisse auf der Speicherkarte, um zu sehen, welche Dateien vorhanden sind.

### View Bitmap

Nur Bitmap-Dateien (\*.bmp) werden in den Verzeichnissen angezeigt (zum Beispiel von Instrumenten).

Klicken Sie auf ein Bitmap um es anzuzeigen. Klicken Sie ein weiteres Mal um die Ansicht zu verlassen.

## 7 Eigene Bilder verwenden

Beim PCAN-MiniDisplay können eigene Bilder benutzt werden. Das können projektspezifische Bilder beim Einschalten der Anzeige oder eigene Instrumentenanzeigen sein.

Die folgende Tabelle enthält Informationen zu den notwendigen Bildeigenschaften, damit eigene Bilder auf dem PCAN-MiniDisplay verwendet werden können:

Eigenschaften der Bilder	
Dateiname	* .bmp
Format	Windows-Bitmap
Auflösung	max. 320 x 240 Pixel
Farbtiefe	24 Bit
Transparenz	Ohne Alphakanal, Transparenz durch Schwarz-Weiß-Maske (keine Abstufung möglich)

Die folgende Grafik veranschaulicht, wie ein eigenes Bild mit Transparenz erzeugt werden kann:

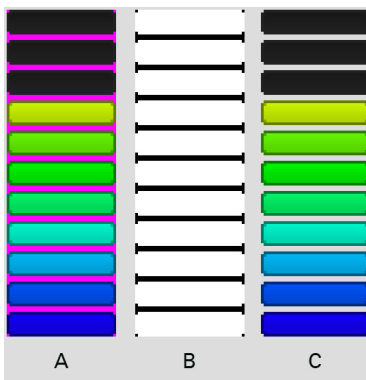


Abbildung 12: Beispielbilder aus der Instrumentenanzeige vom PCAN-MiniDisplay

- A: Ausgangsbild ohne Transparenz
- B: Transparenzmaske (schwarz und weiß)
- C: Ausgangsbild mit Transparenz



**Hinweis:** Das PCAN-MiniDisplay stellt die Bilder mit eingeschränktem Farbumfang dar.

## 7.1 Alternatives Startbild

Jedem PCAN-MiniDisplay kann ein alternatives Startbild zugeordnet werden. Dazu muss eine Bitmap-Datei auf der internen Speicherkarte liegen. Standardmäßig wird das Hersteller-Startbild (PEAK-Logo) angezeigt.

Die folgende Tabelle zeigt, welcher Dateiname und Speicherpfad eingehalten werden muss:

<b>Eigenschaften des Startbilds</b>	
Dateiname	Intro.bmp
Ablagepfad auf der internen Speicherkarte	/MiniDisplay/Intro.bmp



## 8 Technische Daten

### Ein- und Ausgänge

Digitaler Eingang (J2, Pin 4)	10 k $\Omega$ Pull-Up-Widerstand Schaltschwellen: 0: $\leq 1,8$ V (Schalten gegen Masse) 1: $\geq 6,6$ V
Digitaler Ausgang (J2, Pin 3)	Low-Side-Schalter max. Ausgangsspannung 32 V max. Ausgangsstrom 500 mA
Analoge Eingänge (J2, Pin 5 und 6)	Eingangsspannungsbereich 0 bis 16,5 V 10-Bit A/D-Wandler Wertebereich Variable: 0 = 0 V 1023 = 16,5 V

### Display

Art	a-Si TFT active matrix
Auflösung	320 x 240 Pixel
Wirksamer Bereich	57,6 x 43,2 mm (B x L)

### Speicherkarte

Art	microSD
Max. Größe	32 GByte
Dateisystem	FAT; FAT32 falls Größe > 2 GByte
Name des USB-Geräts	PCAN-MiniDisplay

### CAN

Spezifikation	ISO 11898-2, High-Speed-CAN 2.0A (Standard-Format) und 2.0B (Extended-Format)
Übertragungsraten	40 kbit/s - 1 Mbit/s
Transceiver	NXP TJA1041T, Wake-up-fähig
Interne Terminierung	nicht vorhanden

### Versorgung

Versorgungsspannung	7 - 30 V DC
Stromaufnahme	150 mA (bei 12 V) typisch
Stromaufnahme im Screensaver-Modus	70 mA (bei 12 V) typisch

### Umgebung

Betriebstemperatur	-20 - +70 °C
Temperatur für Lagerung und Transport	-30 - +80 °C
Relative Luftfeuchte	15 - 90 %, nicht kondensierend

### Maße

Größe	Mit Rahmen: 82 x 62 x 18,3 mm (B x L x T) Ohne Rahmen: 80 x 55 x 18,3 mm (B x L x T) Siehe auch Maßzeichnungen Anhang B Seite 44
Gewicht	55 g

### Konformität

EMV	Richtlinie 2014/30/EU DIN EN 61326-1:2013-07
RoHS 2	Richtlinie 2011/65/EU DIN EN 50581 VDE 0042-12:2013-02

# Anhang A CE-Zertifikat

## EU Declaration of Conformity



This declaration applies to the following product:

Product name: PCAN-MiniDisplay  
Item number(s): IPEH-002262(-KSM01)  
Manufacturer: PEAK-System Technik GmbH  
Otto-Roehm-Strasse 69  
64293 Darmstadt  
Germany

**CE** We declare under our sole responsibility that the mentioned product is in conformity with the following directives and the affiliated harmonized standards:

### EU Directive 2011/65/EU (RoHS 2)

DIN EN 50581 VDE 0042-12:2013-02

Technical documentation for the assessment of electrical and electronic products with respect to the restriction of hazardous substances;  
German version EN 50581:2012

### EU Directive 2014/30/EU (Electromagnetic Compatibility)

DIN EN 61326-1:2013-07

Electrical equipment for measurement, control and laboratory use - EMC requirements - Part 1:  
General requirements (IEC 61326-1:2012);  
German version EN 61326-1:2013

Darmstadt, 22 February 2019

A handwritten signature in black ink, appearing to read "Uwe Wilhelm".

Uwe Wilhelm, Managing Director

## Anhang B Maßzeichnung

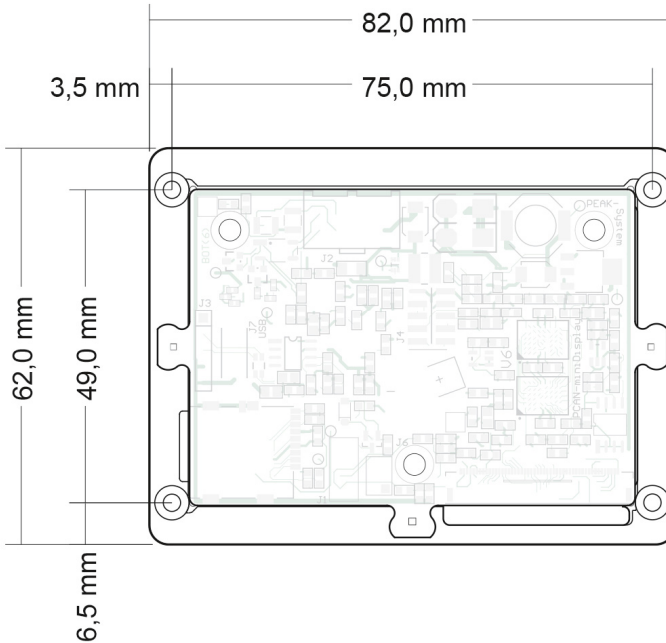


Abbildung 13: Sicht auf das PCAN-MiniDisplay mit Einbaurahmen (IPEH-002262).  
Die Einbautiefe beträgt 18,3 mm.

Die Abbildung entspricht nicht der Originalgröße.

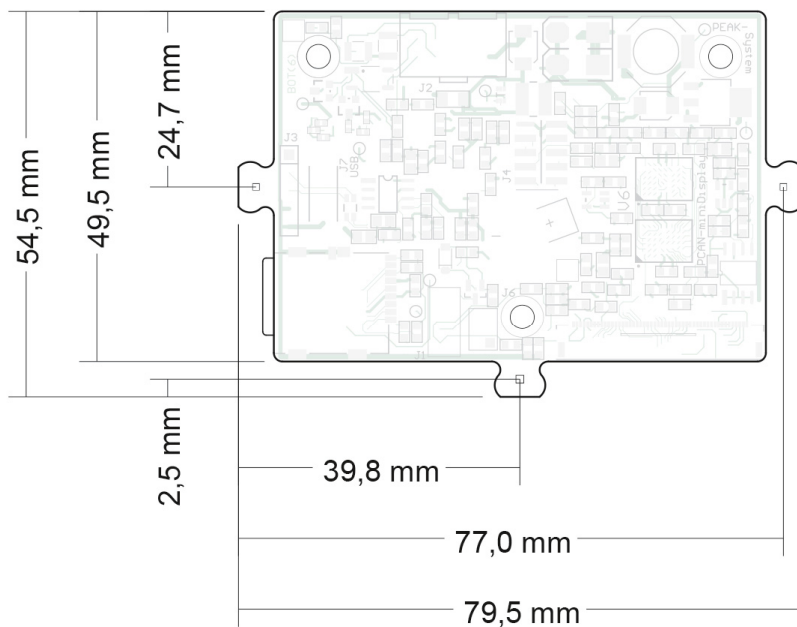


Abbildung 14: Sicht auf das PCAN-MiniDisplay ohne Einbaurahmen (IPEH-002262).  
Die Einbautiefe beträgt 18,3 mm.

Die Abbildung entspricht nicht der Originalgröße.

## Anhang C Struktur auf der Speicherkarte

Die folgende Tabelle gibt die Struktur des Verzeichnisses MiniDisplay auf der Speicherkarte wieder und erläutert die Funktionen.

Verzeichnisstruktur	Funktion
/MiniDisplay/	Festgelegtes Root-Verzeichnis für Dateien, auf die das PCAN-MiniDisplay zugreift oder die mit dem Gerät im Zusammenhang stehen.
Intro.bmp	Startbild beim Einschalten des Gerätes (320 x 240 Pixel).
Documentation/	Dokumentation zum PCAN-MiniDisplay im PDF-Format.
Filters/	Ablageort für Filterdateien (*.flt)
Firmware/*.bin	Ablageort der Binärdateien (*.bin) für ein Update der PCAN-MiniDisplay-Firmware.  <b>Bitte beachten Sie:</b> Wenn Sie eine Firmware kleiner als 1.2.x verwenden, müssen Sie zunächst die Firmware 1.1.7 aufspielen um auf eine Version 1.2.x upgraden zu können.
Help/*.dhp	Dateien der Gerätehilfe (*.dhp). Es empfiehlt sich, diese Dateien im Zusammenhang mit einem Firmware-Update zu aktualisieren.
HTML/	Daten für die Darstellung der ReadMe-Dateien.
Resources	Das Verzeichnis beinhaltet Schriften, Bilder und weitere Ressourcen für die Erstellung eigener Szenen. PCAN-Explorer-Projekte zum Testen der Szenen-Beispiele sind ebenfalls vorhanden.
Scenes/	Dieses Verzeichnis ist der Ablageort für Szenen und enthält die Default-Szene und verschiedene Beispiele. Das Gerät ist bei Auslieferung so konfiguriert, dass automatisch die Default-Szene nach dem Einschalten angezeigt wird.
*.ins	Szenendefinitionsdatei (siehe Kapitel 3 <i>Szenendefinition</i> Seite 14)

Verzeichnisstruktur	Funktion
*.inb	Binäres Gegenstück zu einer Szene (*.ins), das schneller geladen werden kann (siehe Kapitel 3 <i>Szenendefinition</i> Seite 14).
*.inp	Liste mit Szenen, zwischen denen mit den Digitaleingängen umgeschaltet werden kann.
*.fon	Bitmap-Schrift zur Verwendung in einer Szene.
*.bmp	Grafik zur Verwendung in einem Instrument.
Tools/	Ablageort für Windows-Programme für die Arbeit mit dem PCAN-MiniDisplay.
PEAK-Converter.exe	Windows-Programm PEAK-Converter zur Umwandlung einer aufgezeichneten, binären Trace-Datei (*.btr) in ein anderes Format.
Traces/	Festgelegtes Unterverzeichnis für aufgezeichnete Traces.
trc00000.btr	Binärkodierte Trace-Daten von der Aufzeichnungsfunktion, verwendbar zur Wiedergabe oder anderweitig nach einer Konvertierung auf dem PC; fortlaufende Nummerierung durch den internen Zähler.

# Anhang D Referenz Definitionsdateien

Die Schlüsselwörter der folgenden Definitionsdateien werden in diesem Anhang beschrieben:

- ↳ D.1 *Szenendefinition (\*.ins)* beginnt im nächsten Abschnitt
- ↳ D.2 *Szenen Projektdatei (\*.inp)* Seite 80
- ↳ D.3 *CAN-ID-Filter (\*.flt)* Seite 81

## D.1 Szenendefinition (\*.ins)

Folgend werden alle verfügbaren Schlüsselwörter für eine Szenendefinition beschrieben. Sie werden in eine Textdatei (\*.ins) geschrieben und in Abschnitte gegliedert. Abschnittsnamen werden in eckigen Klammern angegeben, z. B. [global]. Einträge sind jeweils in einer Zeile mit einem Schlüsselwort, mit einem Gleichheitszeichen (keine Leerzeichen!) und dem zugewiesenen Wert gesetzt.



**Hinweis:** Die Textdatei (\*.ins) muss eine Leerzeile am Ende haben.

### D.1.1 Abschnitt [global]

#### format

Bestimmt die Dateiformatversion. Zu diesem Zeitpunkt ist nur die Version 1 vorhanden und gültig.


```
format=1
```



## compress

Ist die Funktion compress aktiv wird eine komprimierte, binäre Version (\*.inb) der Textdatei (\*.ins) erzeugt, womit die Szene bei nächsten Start schneller geladen werden kann.


Wert	Erklärung
0	Es wird keine komprimierte Version der Szene erzeugt. Die Textdatei (*.ins) wird für die Ausführung der Szene geladen.
1	Es wird eine komprimierte Version der Szene erzeugt. Die Binärdatei (*.inb) wird für die Ausführung der Szene geladen.

 **Hinweis:** Wenn die Definition einer Szene geändert wurde, muss die vorher erzeugte Binärdatei (\*.inb) auf der Speicherkarte des PCAN-MiniDisplay gelöscht werden. Ansonsten wird keine neue Binärdatei erstellt und eventuelle Änderungen in der Textdatei werden ignoriert.

## protect

Die Funktion protect bezieht sich auf die Erzeugung der binären Version einer Szene durch die Funktion compress. Ist protect aktiv wird die Binärdatei bei der Erzeugung verschlüsselt und kann fortan nur auf dem Gerät ausgeführt werden, auf dem sie erzeugt wurde.

Wert	Erklärung
0	Die Binärdatei der Szene wird nicht verschlüsselt.
1	Die Binärdatei der Szene wird verschlüsselt.

 **Hinweis:** Ist die Definition der Szene vertraulich, kann der Inhalt der Textdatei (\*.ins) nach Erzeugung der Binärdatei (\*.inb) gelöscht werden. Die Binärdatei kann ohne weiteres über die leere Textdatei gestartet werden, da deren Inhalt ignoriert wird, so lange eine Binärdatei existiert.

## bitrate0 bitrate1

Bestimmt die Bitrate des High-Speed CAN-Kanals 1 (bitrate0) und des High-Speed CAN-Kanals 2 (bitrate1). Wenn diese Schlüsselwörter weggelassen werden, gelten die im Menü **Device Settings** definierten Bitraten (**BusOff** bei Auslieferung). Wenn eine Szene mit einer festgelegten Bitrate geschlossen wurde, bleibt die definierte Bitrate aktiv.



**Hinweis:** Während der Inbetriebnahme des Gerätes ist die im Menü **Device Settings** definierte Bitrate für kurze Zeit aktiv, bevor die Bitrate der Szene aktiviert wird.

Mögliche Bitraten (in kbit/s): 1000, 800, 500, 250, 200, 100, 95, 83, 50, 47, 33, 20, 10

```
bitrate0=250  
bitrate1=20
```

## instruments

Anzahl von Instrumenten in dieser Szene, Bereich: 1 ... 32

```
instruments=4
```

## plotter

Anzahl von verwendeten Plottern, Bereich: 1 ... 4

```
plotter=1
```

## messages

Anzahl von verwendeten CAN-Nachrichten, Bereich: 1 ... 96

```
messages=1
```

## enums

Anzahl von verwendeten Enums, Bereich: 1 ... 20

```
enums=2
```

## variables

Anzahl von verwendeten Variablen, Bereich: 1 ... 96

```
variables=7
```

## labels

Anzahl von verwendeten Labels bzw. Beschriftungen,  
Bereich: 1 ... 64

```
labels=5
```

## fonts

Anzahl von zusätzlichen Schriftarten, die für dieses Projekt definiert  
wurden, Bereich: 1 ... 27

Es stehen 3 interne Schriftarten zur Verfügung. Um die internen  
Schriftarten zu verwenden, schreiben Sie font\_idx statt eines  
Schriftnamens. Weitere Informationen siehe D.1.3 *Abschnitt [fontX]*  
auf Seite 56.

```
fonts=2
```

## flashing\_time

Blinkfrequenz für Instrumente mit dem Schlüsselwort "flashing",  
Bereich 50 ... 2500, Standardwert 500. Der Wert ist die Zeit in  
Millisekunden für die "on-time" und die "off-time".

```
flashing_time=250 // 250 ms an, 250 ms aus, Ergebnis ist 2 Hz
```

## D.1.2 Abschnitt [instrumentX]

Jedes Instrument wird durch einen Abschnitt [instrumentX] eingeleitet. Dabei ist X eine fortlaufende Nummer, beginnend bei 1. Die Anzahl der Instrumente wird durch das Schlüsselwort "instruments" im Abschnitt [global] angegeben.

```
[instrument1]
```

### **pictures**

Zahl der Bilder pro Instrument, Bereich: 1 ... 99

```
pictures=4
```

### **name**

Name des Instruments (max. 23 Zeichen). Nur für die Dokumentation. Die Zeichenfolge muss in Anführungszeichen geschrieben werden.

```
name="turn_left"
```

### **var\_name**

Name der Variablen, die das Instrument kontrolliert. Der String muss in Anführungszeichen geschrieben werden.

```
var_name="speed"
```

### **initval**

Bestimmt den Anfangswert des Instruments, bis die ersten CAN-Daten für dieses Instrument empfangen werden.

```
initval=267
```

## no\_restore

Bestimmt wie der Hintergrund während des Zeichnens des Instruments gehandhabt wird.

Wert	Erklärung
0 (default)	Bevor die Instrument-Sequenz gezeichnet wird, wird der Hintergrund gespeichert. Dann könnte der Hintergrund wieder hergestellt werden, bevor eine neue Instrument Sequenz erstellt wird. Dies kann nützlich sein, wenn die Sequenzen unterschiedliche Größen haben.
1	Die Bildfolge wird erstellt ohne den Hintergrund zu speichern - es ist nur dann sinnvoll, wenn alle Bilder des Instruments die gleiche Größe haben und an der gleichen Position erstellt werden. Wird verwendet zur Optimierung der CPU-Last.

```
no_restore=1
```

## restore

Wie no\_restore oben, jedoch mit umgekehrter Logik.

## transparency

Legt fest, ob die Transparenz-Bereiche berücksichtigt werden müssen während des Zeichnens.

Wert	Erklärung
0	Eine Sequenz wird erstellt, ohne den Hintergrund zu berücksichtigen. Das wird verwendet, wenn der Hintergrund ein Teil der Sequenz ist.

1	<p>Beim Laden der *.ins-Datei wird jedes einzelne Bild mit dem Hintergrund verrechnet. Daher muss Instrument0 mit einem globalen Hintergrundbild als einziges Element mit dem Schlüsselwort "no_restore=1" definiert werden. Eine Überlappung verschiedener Transparenz-Bereiche ist hier nicht möglich.</p> <p>Es sind 2 Möglichkeiten für die Neuberechnung verfügbar:</p> <ul style="list-style-type: none"><li>a) Das Bild hat eine Transparenz-Farbe (RGB-Wert 255.0.255).</li><li>b) Das Bild hat einen Alphakanal. Der Alphakanal wird durch eine separate Graustufen-Bitmap definiert. Die Graustufenwerte definieren die Transparenz des Vordergrundbildes.</li></ul> <p>RGB Wert 255.255.255: das Vordergrundbild ist zu 100 % sichtbar, kein Teil des Hintergrunds ist zu sehen. RGB Wert 127.127.127: das Vordergrundbild ist zu 50 % sichtbar und 50 % des Hintergrunds sind zu sehen. RGB Wert 0,0,0: kein Vordergrundbild ist sichtbar, 100 % des Hintergrundbildes sind zu sehen.</p>
2	<p>Für diesen Wert gelten die selben Erläuterungen wie beim Wert 1. Nur ist beim Wert 2 eine Überlappung verschiedener Transparenz-Bereiche möglich.</p>

```
transparency=1
```

## offset

Alle Bilder der Bilderliste dieses Instruments werden in x- und y-Richtung bewegt werden (Offset).

1. Parameter: x offset (Bereich: 0 ... 319)
2. Parameter: y offset (Bereich: 0 ... 239)

```
offset=100,50
```

## background

Definiert ein individuelles Hintergrundbild für dieses Instrument.

Wert	Erklärung
0	Kein Hintergrund für dieses Instrument.
1	Das Instrument verfügt über ein individuelles Hintergrundbild, das verwendet wird, wenn Transparenz=1. Beim Laden der *.ins-Datei werden die Bilder im Hintergrund berechnet. Standardmäßig (background=0) ist das instrument0 das globale Hintergrundbild. Dennoch, für jedes Instrument kann ein individuelles Hintergrundbild definiert werden. Das individuelle Hintergrundbild <u>muß</u> das letzte Bild der Bildliste sein. Das Hintergrundbild hat keinen Alpha-Kanal, so ersetzt es das globale Hintergrundbild vom instrument0. Typischerweise hat ein individuelles Hintergrundbild die gleiche Größe wie ein Einzelbild des Instruments.

```
background=1
```

## update\_rate

Aktualisierungsrate dieses Instruments in Millisekunden. Wenn nicht definiert, wird die Standard-Aktualisierungsrate von 50 ms verwendet. Ein Instrument kann mit jeder neu eingehenden CAN-Nachricht aktualisiert werden. Allerdings könnte es sein, dass der aktuelle Wert aufgrund einer zu hohen Aktualisierungsrate nicht gelesen wird. Dieses Schlüsselwort begrenzt die Aktualisierungsrate, um sicherzustellen, dass die Aktualisierung des Instruments zuverlässig bleibt.

```
update_rate=500 // Zeichnet das Instrument erneut
                // (höchstens zwei mal pro Sekunde)
```

## Picture List

Die Bilderliste des Instruments besteht aus der Anzahl an Einträgen, die mit dem Schlüsselwort „pictures“ angegeben wurde. Der Wert der Variable, die über "var\_name" zugewiesen wurde, bestimmt welches Bild aus der Bilderliste angezeigt wird.

## Formatierung eines Bildlisteneintrags:

<index>=<pic-name>,<transparency-name>,<xpos>,<ypos>,<low-val>,<high-val>

Teil des Bildlisteneintrags	Erklärung
<index>	Nummer des Bildes. Muss im Bereich {0 ... Bilder - 1}.
<pic-name>	Dateiname des Bildes ohne Pfad. Das Bild muss im gleichen Verzeichnis gespeichert werden, in dem die *.ins-Datei gespeichert wurde.
<transparency-name>	Dateiname des Graustufenbildes (Alphakanal). Wenn dieses Bild nicht vorhanden ist, muss eine leere Zeichenkette "" vorgesehen werden. Dateiname ohne Pfad angeben (wie <pic-name>).
<xpos>	x Position {0 ... 319}. Wird dem Parameter x-Offset hinzugefügt.
<ypos>	y Position {0 ... 239}. Wird dem Parameter y-Offset hinzugefügt.
<low-val>	Der niedrigste Wert der Variable für Anzeige des Bildes.
<high-val>	Der höchste Wert der Variable für Anzeige des Bildes.

```

0="arrow_left_off.bmp","",20.100,0,3
  // Bild anzeigen "arrow_left_off.bmp"
  // an Position x=20, y=100
  // wenn die Variable im Bereich von 0 bis 3
1="arrow_left_on.bmp","",20.100,4,7
  // Bild anzeigen "arrow_left_on.bmp"
  // an Position x=20, y=100
  // wenn die Variable im Bereich von 4 bis 7

```

### D.1.3 Abschnitt [fontX]

Jede verwendete Schriftart wird mit einem Abschnitt [fontX] eingeleitet. Dabei ist X eine fortlaufende Nummer, beginnend bei 1. Die Anzahl der Schriftarten wird durch das Stichwort "fonts" im Abschnitt [global] angegeben.

[font1]



## font

Dateiname der Bitmap-Schrift. Die Zeichenfolge muss in Anführungszeichen geschrieben werden. Erweiterung eines Schriftart-Dateinamen ist \*.fon.

```
font="arial_20.fon"
```

Die folgenden Schriftarten sind auf dem PCAN-MiniDisplay verfügbar:

- └─ lucida\_13\_18.fon
- └─ century\_13\_16.fon
- └─ century\_16\_18\_numbers.fon  
(nur Ziffern von 0 bis 9)
- └─ Palatino\_Linotype\_12\_14.fon
- └─ Palatino\_Linotype\_16\_20.fon
- └─ Palatino\_Linotype\_24\_28.fon
- └─ Palatino\_Linotype\_24\_32\_numbers.fon  
(nur Ziffern von 0 bis 9)
- └─ Palatino\_Linotype\_24\_32\_numbers\_bold.fon  
(nur Ziffern von 0 bis 9)
- └─ Courier\_13\_18\_bold.fon  
(fett formatiert)
- └─ gear\_40\_49.fon  
(Ziffern von 0 bis 7 werden angezeigt als N, 1, 2, 3, 4, 5, 6, R in einer Größe von 40 x 49 Pixel)

## name

Name der Schrift, als Referenz für ein Element, das diese Schriftart verwendet.

```
name="arial_20"
```

## type

Typ der Schriftart. Derzeit sind nur einfarbige Schriftarten mit fester Größe verfügbar (type=0).

```
type=0
```

### D.1.4 Abschnitt [messageX]

Die Definition jeder CAN-Nachricht wird durch einen Abschnitt [messageX] eingeleitet. Dabei ist X eine fortlaufende Nummer, beginnend bei 1. Die Anzahl der Nachrichten wird durch das Schlüsselwort "messages" im Abschnitt [global] angegeben.

Variablen können auf die so definierten CAN-Nachrichten referenzieren.

```
[message1]
```

## enable

Wert	Erklärung
0	Auswertung der jeweiligen [messageX] deaktiviert.
1	Auswertung der jeweiligen [messageX] aktiviert.

```
enable=1
```

## canid

Angabe der 11-Bit oder 29-Bit CAN ID einer Nachricht als ganzzahlige Werte (dezimal oder hex).

```
canid=256  
canid=0x100
```

## frametype

Definiert den CAN-Frame-Typ als Standard Frame oder Extended Frame.

Wert	Erklärung
0	Standard Frame (11-Bit CAN-ID)
1	Extended Frame (29-Bit CAN-ID)

```
frametype=0 // Standard Frame mit 11-Bit-ID
```

## channel

Definiert den CAN-Kanal, der diese Nachricht empfängt.

Wert	Erklärung
0	CAN-Kanal 1
1	CAN-Kanal 2

```
channel=0 // Verwendet den ersten CAN-Kanal
```

## timeout

Wenn eine [messageX] in der angegebenen Zeitspanne (bis zu 2<sup>32</sup> Millisekunden möglich) nicht empfangen wurde, werden alle Variablen dieser Nachricht auf den jeweiligen "timeoutval" gesetzt (siehe *timeoutval* auf Seite 67).

Wert	Erklärung
0	timeout deaktiviert
>0	timeout ist aktiviert. Der Wert gibt die Timeout-Verzögerung in Millisekunden an.

```
timeout=1000
```

## Beispiel für eine [messageX]

```
[message1]
enable=1           // Auswertung der Nachrichten ist aktiviert
name="testmessage1" // Name der Nachricht
canid=0x125
frametype=0       // 0: 11-Bit, 1: 29-Bit
channel=0
timeout=1000
```

## Beispiel für eine [variableX] die auf eine [messageX] referenziert

```
[variable1]
name="Var1_mux"
messagename="testmessage1"
timeoutval=133
...
```

### D.1.5 Abschnitt [enumX]

Jedes Enum wird durch einen Abschnitt [enumX] eingeleitet. Dabei ist X eine fortlaufende Nummer, beginnend bei 1. Die Anzahl der Enums wird durch das Schlüsselwort "enums" im Abschnitt [global] angegeben.

```
[enum1]
```

#### name

Der Name des Enums dient als Referenz für die Verwendung durch eine Variable (max. 23 Zeichen).

```
name="enum_test"
```

## datatype

Gibt den Datentyp der Variablen an, die das Enum verwendet.

Wert	Erklärung
0	(für das Enum können nur positive Zahlen verwendet werden)
1	(für das Enum können positive und negative Zahlen verwendet werden)

```
datatype=1
```

## count

Gibt die Anzahl der Einträge der Enum-Werteliste an. Die maximale Zahl der Einträge ist nur durch den Arbeitsspeicher limitiert.

```
count=2
```

## length

Angabe der Länge der längsten Zeichenkette, die mit dem Enum ausgegeben wird.

```
length=13
```

## Enum-Werteliste

Die Werteliste des Enums besteht aus der Anzahl an Einträgen, die mit dem Schlüsselwort „count“ angegeben wurde. Der Wert der Variable, die das Enum verwendet, bestimmt, welcher Wert aus der Liste verwendet wird.

Ein Eintrag besteht aus einem ganzzahligen Wert und der Zeichenkette, die anstelle des Wertes angezeigt werden soll. Die Zeichenkette sollte maximal 16 Zeichen lang sein.

```
0="Zeichenkette"
```

## Example for a [enumX]

```
[enum1]
name="enum_test"
datatype=0           // 0 => unsigned, 1 => signed
count=3             // Anzahl der Enum-Wertelisten-Einträge
length=7            // Zeichen- Anzahl der längsten Zeichenkette
0="Minimum"         // Enum-Werteliste
128="Middle"
255="Maximum"
```

### D.1.6 Abschnitt [variableX]

Jede Variable wird durch einen Abschnitt [variableX] eingeleitet. Dabei ist X eine fortlaufende Nummer, beginnend bei 1. Die Anzahl der Variablen wird durch das Schlüsselwort "variables" im Abschnitt [global] angegeben.

```
[variable1]
```

#### name

Der Name der Variablen dient als Referenz für die Verwendung durch ein Instrument oder ein Label (max. 23 Zeichen).

```
name="Speed"
```

Es gibt 3 spezielle Variablennamen zur Ansteuerung von Gerätefunktionen oder Parametern:

```
name="dev_keys"
```

Benennen Sie Ihre Variable `dev_keys`, um die Tasten-Funktion des PCAN-MiniDisplay anzusprechen. Die Variable kann die folgenden Werte haben: 0 für keine Taste gedrückt, 1 für Enter, 2 für Key up und 3 für Key down.

```
name="dev_brightness"
```

Benennen Sie Ihre Variable `dev_brightness`, um die Display-Helligkeit des PCAN-MiniDisplay einzustellen. Ganzzahlige Werte von 0 bis 100 sind gültig.

```
name=" dev_buzzer"
```

Benennen Sie Ihre Variable `dev_buzzer`, um den Buzzer des PCAN-MiniDisplay auszulösen. Ganzzahlige Werte von 0 bis 14 sind gültig. Der Buzzer-Sound wird nur einmal bei Änderung des Variablenwertes abgespielt.

### messageName

Der Name verweist auf die jeweilige [messageX] (max. 23 Zeichen).

```
messageName="testmessage1"
```



**Hinweis:** Seit der Firmware 1.2.x ersetzt "messageName" die Parameter „frameType“ und „canId“. Sobald „messageName“ gesetzt wurde, werden die Parameter „source“ und „channel“ aus der jeweiligen [messageX] übernommen.

### source

Typ der Quelle. Verfügbare Quellentypen: CAN, AIN und DIN.

```
source=CAN
```

### channel

Der zu verwendende Kanal (Zählung beginnt bei 0). Die Anzahl der unterstützten Kanäle hängt vom Quellentyp ab.

Quellenart	Wert	Erklärung
CAN	0	CAN-Kanal 1
	1	CAN-Kanal 2
AIN	0	Interner NTC
	1	Externer Eingang

Quellenart	Wert	Erklärung
	2	Externer Eingang
DIN	4	Externer Eingang (Schalter gegen GND)

```
channel=0
```

## position

1. Parameter: Start-Bit-Position des Signals, Bereich: 0 ... 63  
Hinweis: Die Bits werden unterschiedlich in Intel und Motorola-Format gezählt.
2. Parameter: Bit-Länge des Signals, Bereich: 1 ... 64

```
position=0,8 // Erstes Byte des CAN-Datenfeldes
position=8,16 // Bytes 2 und 3 des CAN-Datenfeldes
```

## byteorder

Definiert die Byte-Reihenfolge der CAN-Daten.

Wert	Erklärung
0	Intel ("Little Endian")
1	Motorola ("Big Endian")

```
byteorder=0 // Intel format
```

## muxtype

Die Datenstruktur einer CAN-Nachricht wird mit Variablen definiert. Mit Hilfe eines Multiplexers können für dieselbe CAN-Nachricht verschiedene Datenstrukturen definiert werden. Dafür wird zuerst eine Variable als Multiplexer angelegt. Deren Wert gibt an, welche Datenstruktur die CAN-Nachricht enthält.

Mit "muxtype" wird angegeben, ob die zu konfigurierende Variable der Multiplexer ist, oder abhängig von einem Multiplexer in den Daten der Nachricht ist.



Wert	Erklärung
0	Die Multiplexer-Funktion wird nicht verwendet.
1	Die Variable wird als Multiplexer verwendet. Der Multiplexer muss vor allen anderen Variablen dieser CAN-Nachricht definiert werden. Pro CAN-ID darf nur ein Multiplexer angelegt werden.
2	Die Variable ist abhängig vom Wert des Multiplexers in den Daten der CAN-Nachricht enthalten.

```
muxttype=0
```

### **muxval**

Definiert den Multiplexer-Wert, der per Dezimalwert (von 0 bis  $2^{32} - 1$ ) oder im Hexadezimalwert (von 0 bis 0xFFFFFFFF) angegeben werden kann. Der unsigned-Wert kann eine Länge von 1 bis 32 Bit haben.

```
muxval=103
```

### **scale**

Multiplikator für einen eingehenden Signal-Wert. Der Wert wird als Gleitkommazahl gespeichert (32-Bit).

```
scale=2.5
```

### **offset**

Offset für einen eingehenden Signal-Wert. Der Wert wird als Gleitkommazahl gespeichert (32-Bit).

```
offset=5.2
```

### **maxraw**

Maximaler Rohwert (Wert aus CAN-Nachricht vor der Scale/Offset-Berechnung) mit internem Datentyp Gleitkommazahl 32-Bit. Alle Werte, die größer sind als dieser, werden auf diesen begrenzt. Verwenden Sie entweder maxraw ODER maxval.

## maxval

Maximaler physikalischer Wert (Wert aus CAN-Nachricht nach Scale/Offset-Berechnung) mit internem Datentyp Gleitkommazahl 32-Bit. Alle Werte, die größer sind als dieser, werden auf diesen begrenzt. Verwenden Sie entweder maxraw ODER maxval.

## minraw

Minimaler Rohwert (Wert aus CAN-Nachricht vor der Scale/Offset-Berechnung) mit internem Datentyp Gleitkommazahl 32-Bit. Alle Werte, die kleiner sind als dieser, werden auf diesen begrenzt. Verwenden Sie entweder minraw ODER minval.

## minval

Minimaler physikalischer Wert (Wert aus CAN-Nachricht nach Scale/Offset-Berechnung) mit internem Datentyp Gleitkommazahl 32-Bit. Alle Werte, die kleiner sind als dieser, werden auf diesen begrenzt. Verwenden Sie entweder minraw ODER minval.

## vartype

Datentyp der Variablen nach Auslesen aus der CAN-Nachricht.

Wert	Erklärung
0	Ganzzahl ohne Vorzeichen
1	Ganzzahl mit Vorzeichen
2	Gleitkommazahl (32-Bits)
4	Zeichenkette (String)

```
vartype=1
```

Eine signed Ganzzahl-Variable mit einer angenommenen Bit-Länge von 8 hat einen Wertebereich von -128 bis +127.

## datatype

Datentyp der Variablen nach Auslesen aus der CAN-Nachricht und weiteren Berechnungen mit Scale und Offset.

Wert	Erklärung
0	Ganzzahl ohne Vorzeichen
1	Ganzzahl mit Vorzeichen
2	Gleitkommazahl (32-Bits)
4	Zeichenkette (String)
5	Enum (für Wertausgabe mit einem Enum)

```
datatype=2
```

Der Wert wird intern als Gleitkommazahl gespeichert. Dies ist sinnvoll, wenn der Wert als Bruchzahl angezeigt wird (siehe Label).

## timeoutval

Definiert den Wert der Variablen, wenn das Ereignis "timeout" eintritt (siehe *timeout* auf Seite 59). Der Wert wird mit den Parametern Scale und Offset verrechnet. Ist kein "timeoutval" definiert, wird der Variablenwert auf 0 gesetzt.

```
timeoutval=133
```

## enum

Der Name verweist auf das jeweilige [enumX] (max. 23 Zeichen).

```
enum="enum_test"
```

## Beispiel einer Variablen

```
[variable1]
name="odometer"
message="testmessage1" // Name der Nachricht
position=8,16 // Start-Bit und Bit-Länge
byteorder=1 // 0: Intel, 1: Motorola
scale=1.5 // Wert=raw * scale + offset
offset=0 // immer Gleitkommazahl
vartype=0 // Typ der eingehenden CAN-Daten:
// 0: unsigned, 1: signiert, 2: Gleitkommazahl
minval=3 // Untere Grenze ist 3.
maxval=99999 // Jeder Wert > 99999 wird zu 99999.
datatype=1 // Typ der CAN-Daten nach Scale & Offset:
// 0: unsigned, 1: signed, 2: float,
// 4: string, 5: enum
```

## Beispiel für Multiplexer-Variablen

```
[message1]
enable=1
name="testmessage1"
canid=0x1
frametype=0
channel=0
timeout=0

[variable1]
name="multiplexer"
messagenname="testmessage1"
position=0,8
byteorder=1
muxtype=1           // 1=multiplexer
scale=1
offset=0
vartype=0
datatype=0

[variable2]
name="multiplexed1"
messagenname="testmessage1"
position=8,16
byteorder=1
muxtype=2           // 2=multiplexed
muxval=103          // Diese Variable wird aktualisiert, wenn der
                    // Multiplexer (Variable 1) diesen Wert hat.
scale=1
offset=0
vartype=0
datatype=0
timeoutval=0

[variable3]
name="multiplexed2"
messagenname="testmessage1"
position=8,8
byteorder=1
muxtype=2           // 2=multiplexed
muxval=104          // Diese Variable wird aktualisiert, wenn der
                    // Multiplexer (Variable 1) diesen Wert hat.
scale=1
offset=0
vartype=0
datatype=0
timeoutval=0
```

### D.1.7 Abschnitt [labelX]

Jede Beschriftung wird durch einen Abschnitt [labelX] eingeleitet. Dabei ist X eine fortlaufende Nummer, beginnend bei 1. Die Anzahl der Beschriftungen wird durch das Schlüsselwort "labels" im Abschnitt [global] angegeben.

```
[label1]
```

#### **name**

Name des Labels (max. 15 Zeichen), zur Dokumentation.

```
name="odometer"
```

#### **position**

X- und Y-Position des Labels.

1. Parameter: X-Position (horizontal), Bereich: 0 - 31
2. Parameter: Y-Position (vertikal), Bereich: 0 - 239

```
position=10,30
```

#### **font\_name**

Name der Schrift (max. 63 Zeichen), Verweis auf eine Schrift aus dem Abschnitt Schriftarten.

```
font_name="Palatino_Linotype_12_14.fon"
```

Das Beispiel wählt eine Schriftart mit einer Breite von 12 Pixel und einer Höhe von 14 Pixeln. Weitere Schriften siehe Liste der verfügbaren Schriftarten.

#### **font\_idx**

Index einer eingebetteten Schriftart.

Wert	Erklärung
0	"Courier_8_12", Breite=8, Höhe=12
1	"Courier_10_14", Breite=10, Höhe=14
2	"Courier_12_18", Breite=12, Höhe=18

```
font_idx=2
```

## length

Anzahl der Zeichen bzw. Länge der auszugebenden Zeichenfolge. Maximum ist 39 Zeichen.

```
length=5
```

## fmt\_string

Format-Angabe entsprechend der printf-Funktion, bekannt aus verschiedenen Programmiersprachen, zur Ausgabe des Variablenwertes.

Wird keine Angabe zur Formatierung gemacht, wird eine Standard-Formatierung entsprechend des datatype der Variablen verwendet.

```
fmt_string="%02d km"
```

## var\_name

Verweis auf eine Variable aus dem Abschnitt Variablen (max. 23 Zeichen).

```
var_name="speed"
```

## initval

Anfangswert, der nach dem Start ausgegeben wird, bevor eine relevante Variable empfangen wird (max. 39 Zeichen).

```
initval="-----"
```

## fontcolor

Definiert die Schriftfarbe als 8 Bit-RGB-Wert. Die Farbanteile rot, grün und blau haben einen Wertebereich von 0 bis 255.

```
fontcolor=255,255,255 // white
```

## bgcolor

Definiert die Hintergrundfarbe als 8 Bit-RGB-Wert. Die Farbanteile rot, grün und blau haben einen Wertebereich von 0 bis 255.

```
bgcolor=255,0,0 // red
```

## update\_rate

Definiert die Aktualisierungsrate eines Labels in Millisekunden. Mit dieser Geschwindigkeit wird das Label neu gezeichnet, unabhängig vom Empfang einer relevanten Variablen.

```
update_rate=100 // Label wird nicht öfter als alle 100 ms neu  
gezeichnet.  
update_rate=0 // Das Label wird so schnell wie möglich  
// gezeichnet (jedesmal, wenn die Variable  
// empfangen wird)
```

## sector

Der Wertebereich einer Variablen kann in bis zu 5 Sektoren unterteilt werden. Für jeden Sektor kann eine andere Schriftfarbe und Hintergrundfarbe definiert werden. Jeder Sektor kann das Schlüsselwort "flashing" auf 1 oder 0 einstellen. Ein Sektor kann aktiviert (1) oder deaktiviert (0) werden.

```
sector=1 // Sektor aktiviert  
range=0,10  
fontcolor=255,0,0 // leuchtend rot  
offcolor=127,0,0 // hellrot  
bgcolor=0,0,0 // schwarz  
flashing=1  
sector=1 // Sektor aktiviert  
range=11,244
```



```
fontcolor=0,255,0 // leuchtend grün
bgcolor=0,0,0     // schwarz
flashing=0       // kein blinken

sector=1         // Sektor aktiviert
range=245,255
fontcolor=255,0,0 // leuchtend rot
offcolor=127,0,0 // hellrot
bgcolor=0,0,0    // schwarz
flashing=1
```

**Beispiel:** Ein 8-Bit vorzeichenlose Variable hat einen Wertebereich von 0 bis 255. Wenn der Wert gleich 0 bis 10 ist, wird der Wert in rot und blinkend angezeigt. Im Bereich von 11 bis 244 wird er in grün dargestellt. Im Bereich von 245 bis 255 wird er in rot und blinkend angezeigt.

## range

Ein Sektor wird für einen bestimmten Bereich definiert.

```
range=20,50
```

Dieser Sektor ist gültig, wenn die Variable im Bereich von 20 bis 50 ist (20 und 50 enthalten).

## offcolor

Jeder Sektor kann das Schlüsselwort "flashing" haben. Für den Fall das der Sektor aktiv ist und das Label blinkt, wechselt die Farbe zwischen "fontcolor" und "offcolor".

```
offcolor=127,0,0 // mittleres rot
```

## Beispiel für ein Label

```

[labell]
name="odometer"           // Name des Labels als Referenz
position=10,30           // x-pos,y-pos in Pixel
                          // 0,0 ist der linken oberen Ecke
//font_name=""           // Verwenden Sie font_name ODER font_idx
font_idx=2               // Anwendung der integrierten Schrift 2
length=8                 // Anzahl der Zeichen des Labels
fmt_string="%05d km"     // Angabe zur Formatierung nach printf()
var_name="odomtr"       // Verweis auf Variable
startval="---- km"      // Anfangswert der angezeigt wird.
fontcolor=0,255,0       // RGB-Wert (leuchtend grün)
bgcolor=127,0,0         // RGB-Wert (mittleres rot)

sector=1                  // Neuer Sektor, 1: aktiviert
range=0,10                // Anwendbar für Werte innerhalb 0 bis 10
fontcolor=0,255,0        // Schriftfarbe für diesen Bereich
bgcolor=0,0,0            // Hintergrundfarbe für diesen Bereich
flashing=0

sector=1
range=1,20
fontcolor=255,0,0
offcolor=127,0,0
bgcolor=0,0,0
flashing=1

```

Sobald das Schlüsselwort "sector" verwendet wird, werden die Parameter für "fontcolor" und "bgcolor" diesem Sektor zugeordnet.

### D.1.8 Abschnitt [plotterX]

Jeder Plotter wird durch einen Abschnitt [plotterX] eingeleitet. Dabei ist X eine fortlaufende Nummer, beginnend bei 1. Die Anzahl der Plotter wird durch das Schlüsselwort "plotter" im Abschnitt [global] definiert.

#### name

Name des Plotters (max. 15 Zeichen), nur als Referenz.

```
name="Vehicle Speed"
```

## **position**

X- und Y-Position des Plotters.

```
position=100,50
```

## **size**

Abmessung des Plotters in Pixel.

1. Parameter: Breite
2. Parameter: Höhe

```
size=100,50
```

## **plotcolor**

Definiert die Farbe der Diagrammlinie als 8 Bit-RGB-Wert. Die Farbanteile rot, grün und blau haben einen Wertebereich von 0 bis 255.

```
plotcolor=255,0,0 // Farbe ist 100% rot  
plotcolor=127,127,127 // Farbe ist mittelgrau
```

## **bgcolor**

Definiert die Hintergrundfarbe als 8 Bit-RGB-Wert. Die Farbanteile rot, grün und blau haben einen Wertebereich von 0 bis 255.

```
bgcolor=255,255,255 // weißer Hintergrund
```

## **enable**

Aktiviert oder deaktiviert diesen Plotter.

```
enable=1
```

## channels

Anzahl der Kanäle für dieses Plotter-Diagramm. Bis zu 4 Kanäle können in einem Plotter angezeigt werden. Die Kanal-Indizes werden verwendet um Name, Variable etc. anzugeben.

```
channels=2
```

## timespan

Zeit in Millisekunden, die die x-Achse darstellen soll.

```
timespan=5000 // 5 s um das komplette Diagramm zu zeichnen
```

## var\_nameX

Name der Datenquelle (max. 23 Zeichen). X kann 1, 2, 3 oder 4 sein für bis zu 4 Variable in einem Plotter-Diagramm. Siehe Schlüsselwort "channels".

```
var_name1=speed  
var_name2=acceleration
```

## yminX

Minimalwert der vertikalen Achse. Ein Wert kleiner als dieser wird auf diesen Minimalwert beschränkt.

```
ymin1=10
```

## ymaxX

Maximalwert der vertikalen Achse. Ein Wert größer als dieser wird auf diesen Maximalwert beschränkt.

```
ymax2=250
```

## scroll\_x

Activates scrolling along the X axis for the plotter.

```
scroll_x=1
```

### D.1.9 Abschnitt [txlist\_X]

Jede Sendeliste wird durch einen Abschnitt [txlist\_X] eingeleitet. X ist dabei einer von drei Sende-Typen: init, cyclic oder exit. Für jeden Sende-Typ können maximal 30 Nachrichten definiert werden.

```
[txlist_X]
```

#### txlist\_init

Die Sendeliste wird einmal beim Starten einer Szene ausgeführt.

```
[txlist_init]
```

#### txlist\_cyclic

Die Sendeliste wird zyklisch während der Szene ausgeführt.

```
[txlist_cyclic]
```

#### txlist\_exit

Die Sendeliste wird einmal beim Beenden einer Szene ausgeführt.

```
[txlist_exit]
```

Nach der Einleitung des Abschnittes [txlist\_X] folgen Parameter, deren Reihenfolge eingehalten werden muss:

```
[txlist_X]
count=1
delay,bus,message-ID,frameformat,message-type,length,data[0],
data[1],data[2],data[3],data[4],data[5],data[6],data[7]
```

#### count

Gibt die Anzahl der definierten CAN-Nachrichten in der jeweiligen Sendeliste [txlist\_X] an.

```
count=1
```

## delay

Legt eine Verzögerung in Millisekunden für die Übertragung der Sendenachricht fest.

```
150, bus, message-ID, frameformat, messagetype, length, data[0], ...
```

## bus

Legt den CAN-Kanal für die Sendenachricht fest.

Wert	Erklärung
0	CAN-Kanal 1
1	CAN-Kanal 2

```
150, 1, message-ID, frameformat, messagetype, length, data[0], ...
```

## message-ID

Gibt die ID der CAN-Nachricht in 11-Bit oder 29-Bit ganzzahligen Werten (dezimal oder hex) an.

```
delay, bus, 0x180, frameformat, messagetype, length, data[0], ...
```

## frameformat

Definiert den CAN-Frame-Typ als Standard Frame oder Extended Frame.

Wert	Erklärung
s	Standard Frame (11-Bit CAN-ID)
e	Extended Frame (29-Bit CAN-ID)

```
delay, message-ID, s, messagetype, length, data[0], ...
```

## messagetype

Hier kann festgelegt werden, ob es sich um ein Daten- oder Remoteframe handelt.

Wert	Erklärung
d	Daten-Frame
r	Remote-Frame

```
delay, bus, message-ID, frameformat, d, length, data[0], ...
```

## Length

Gibt die Länge der CAN-Nachricht (0 bis 8 Datenbyte) an.

```
delay,bus,message-ID,frameformat,message-type,4,data[0],...
```

## data[x]

Angabe der Daten der CAN-Nachricht in ganzzahligen 8-Bit-Werten (dezimal oder hex).

```
delay,bus,message-ID,frameformat,message-type,length,0x11,...
```

## Beispiele für Sendelisten

```
[txlist_init] // Wird einmal beim Starten einer Szene gesendet.
count=6
100,0,0x80,s,d,4,0x11,0x22,0x33,0x44
200,0,0x81,s,d,5,0x11,0x22,0x33,0x44,0x55
250,0,0x82,s,d,8,0x11,0x22,0x33,0x44,0x55,0x66,0x77,0x88
100,1,0x80,s,d,4,0x11,0x22,0x33,0x44
200,1,0x81,s,d,5,0x11,0x22,0x33,0x44,0x55
250,1,0x82,s,d,8,0x11,0x22,0x33,0x44,0x55,0x66,0x77,0x88

[txlist_exit] // Wird einmal beim Beenden einer Szene gesendet.
count=6
100,0,0x180,s,d,4,0x11,0x22,0x33,0x44
200,0,0x181,s,d,5,0x11,0x22,0x33,0x44,0x55
250,0,0x182,s,d,8,0x11,0x22,0x33,0x44,0x55,0x66,0x77,0x88
100,1,0x180,s,d,4,0x11,0x22,0x33,0x44
200,1,0x181,s,d,5,0x11,0x22,0x33,0x44,0x55
250,1,0x182,s,d,8,0x11,0x22,0x33,0x44,0x55,0x66,0x77,0x88

[txlist_cyclic] // Wird während einer Szene zyklisch gesendet.
count=4
100,0,0x44,s,d,4,0x01,0x02,0x03,0x04
200,0,0x48,s,d,5,0x21,0x22,0x23,0x24,0x25
100,1,0x44,s,d,4,0x01,0x02,0x03,0x04
200,1,0x48,s,d,5,0x21,0x22,0x23,0x24,0x25
```

## D.2 Szenen Projektdatei (\*.inp)



**Hinweis:** Die Textdatei (\*.inp) muss eine Leerzeile am Ende haben.

### D.2.1 Abschnitt [global]

#### version

Versions-Format. Derzeit nur 1.

```
version=1
```

### D.2.2 Abschnitt [scenes]

#### scenes

Anzahl der verwendeten Szenen.

```
scenes=4
```

#### startscene

Diese Szene wird nach dem Start angezeigt, Standard ist 1.

```
startscene=2
```

#### Szenenliste

Format eines Szenen-Listeneintrag:

```
<index>=<filename>
```

Die Szenen sind fortlaufend nummeriert, beginnend mit 1. Als Parameter muss der Dateiname der Szene angegeben werden. Wurde in einer Szene das Schlüsselwort "compress=1" verwendet und wurde diese bereits geladen, so wird die komprimierte Version verwendet. Das hilft die Ladezeit zu verringern.



```
[scenes]
scenes=4
startscene=2
1="speedo0.ins"
2="speedo1.ins"
3="picture1.ins"
4="pictucce2.ins"
```

## D.3 CAN-ID-Filter (\*.flt)

Die Verwendung eines CAN-ID-Filters (\*.flt-Datei) wird im Menü **Device Settings** des PCAN-MiniDisplay bestimmt. Bei einer Aufzeichnung können CAN-IDs und ID-Bereiche, die in einer Filter-Datei aufgelistet werden, den Filter passieren. Alle anderen nicht.

### D.3.1 Formatbeschreibung der Filter-Datei

- └ Eine .flt-Text-Datei hat mehrere Abschnitte, die jeweils mit dem Abschnittsnamen in eckigen Klammern eingeleitet werden.
- └ Abhängig vom Abschnitt können Schlüsselworte, IDs oder CAN-ID-Bereiche enthalten sein.
- └ Eine CAN-ID wird durch einen dezimalen bzw. hexadezimalen Wert (letztere mit Präfix 0x) angegeben.
- └ Aus Performance-Gründen empfiehlt es sich, CAN-IDs in aufsteigender Reihenfolge aufzulisten.
- └ Ein Kommentar kann eingesetzt werden, beginnend mit einem doppelten Schrägstrich.

Sektion/Schlüsselwort	Erklärung	Beispieleinträge
[global]	Dieser Abschnitt ist obligatorisch und muss die folgenden zwei Schlüsselwörter enthalten.	

Sektion/Schlüsselwort	Erklärung	Beispieleinträge
version	Versions-Format. Derzeit ist nur 1 gültig.	version=1
enable	Aktiviert den Filter. Derzeit ist nur 1 gültig.	enable=1
[single_11bit]	Definiert 11-Bit-CAN-IDs, die den Filter passieren können.	0x100 1023
[range_11bit]	Definiert 11-Bit-CAN-ID-Bereiche, die den Filter passieren können.	4-13 0x200-0x340 // Alle 11-Bit-IDs: 0x000-0x7ff
[single_29bit]	Definiert 29-Bit-CAN-IDs, die den Filter passieren können.	0x123 0x11111 125000
[range_29bit]	Definiert 29-Bit-CAN-ID-Bereiche, die den Filter passieren können.	500-550 0x9000-0x10000 0x1F80000-0x1FA0000 // Alle 29-Bit-IDs: 0x0000000-0x1FFFFFFF

### D.3.2 Beispiel Filter-Datei

```
[global]
version=1
enable=1

// Dies ist ein Kommentar
[single_11bit]
0x100
1023 // = 0x3FF - Ein weiterer Kommentar
0x5AB

[range_11bit]
4-13
0x200-0x340
0x7f0-0x7fe

[single_29bit]
0x123
0x11111
125000 // = 0x1E848

[range_29bit]
```

500-550  
0x9000-0x10000  
0x1F80000-0x1FA0000



**Tipp:** Auf der Speicherkarte des PCAN-MiniDisplay befindet sich das Beispiel `Example.flt` im Verzeichnis: `/MiniDisplay/Filters`.

## Anhang E Ausführung mit Gehäuse

Dieser Abschnitt beschreibt die Besonderheiten der Ausführung mit Gehäuse und Drucktasten (IPEH-002262-KSM01).

Diese Produktausführung bietet eine Umgebung, um die Entwicklung von Konfigurationen für das PCAN-MiniDisplay zu erleichtern.



Abbildung 15: Ausführung mit Gehäuse und Tasten


### E.1 Lieferumfang

- PCAN-MiniDisplay im Kunststoffgehäuse inklusive Gegenstecker (Phoenix Contact FMC 1,5/10-ST-3,5 1952348 und Phoenix Contact FMC 1,5/4-ST-3,5 1952283)
- Installierte Speicherkarte 512 MBytes
- Konvertierungssoftware PEAK-Converter für Windows 10, 8.1, 7 (32/64-Bit)
- Konfigurationsbeispiele und Handbuch im PDF-Format

## E.2 Drucktasten

Die drei Tasten Hoch ↑, Runter ↓ und Eingabe ↵ sind geräteintern an J1 des PCAN-MiniDisplays angebunden (für Details siehe 2.1 Seite 9). Die Verwendung hängt vom aktuellen Modus des Geräts ab.

Aktueller Modus	Hoch ↑	Runter ↓	Eingabe ↵
Szene-Display	Zur nächsten Szene wechseln, wenn mehrere Szenen definiert sind		Verlassen der Szene zum Hauptmenü
Menüs	Wechseln zum vorherigen oder nächsten Eintrag aus einem Menü oder einer Liste		Aktiviert den gewählten Eintrag
Off (angelegte Versorgungsspannung)	Ohne Funktion	Start des Geräts	Ohne Funktion

 **Hinweis:** Die digitalen Eingänge von J1 sind auch auf dem 4-poligen Anschluss verfügbar.

## E.3 Anschlüsse

Die Ausführung mit Gehäuse bietet drei Anschlüsse auf der Rückseite:

- Mini-USB-Buchse
- 4-polige Steckverbindung zur passenden Federklemmenleiste (J1 intern)

Pin	Funktion
1	Din1: Runter
2	Din2: Hoch
3	Din3: Eingabe
4	GND



- 10-polige Steckverbindung zur passenden Federklemmenleiste (J2 intern)

Pin	Funktion
1	Vb (7 - 30 V DC)
2	GND
3	Dout1
4	Din4
5	Ain1
6	Ain2
7	CAN2_L
8	CAN2_H
9	CAN1_L
10	CAN1_H



## E.4 miniSD-Kartensteckplatz

Der microSD-Kartensteckplatz ist durch Öffnen des Gehäuses zugänglich.

- So greifen Sie auf den microSD-Kartensteckplatz zu:
1. Entfernen Sie die vier Kunststoffkappen vorsichtig von der Unterseite des Gehäuses.
  2. Entfernen Sie die vier Schrauben, die durch die Kappen abgedeckt wurden.
  3. Entfernen Sie die Oberseite des Kunststoffgehäuses (mit dem Fenster).

Der miniSD-Kartensteckplatz ist unterhalb der Display-Leiterplatte positioniert, in der Nähe der einzelnen Taste.

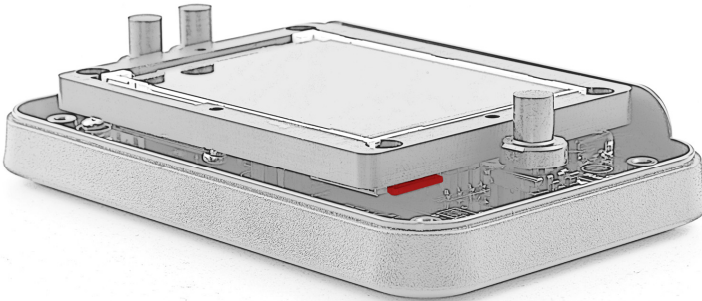


Abbildung 16: Position des microSD-Kartensteckplatzes

4. Drücken Sie leicht auf die Karte bis der Federmechanismus diese freigibt (Klick-Geräusch).
5. Entfernen Sie die microSD-Karte vorsichtig mit einem geeigneten Werkzeug (z.B. Pinzette).
6. Zum Einlegen einer Karte drücken Sie bis der Federmechanismus schließt (Klick-Geräusch).
7. Um das Gehäuse wieder zusammenzusetzen, befolgen Sie die ersten drei Schritte in umgekehrter Reihenfolge.

## E.5 Technische Daten Gehäuseausführung

Die Technischen Daten des PCAN-MiniDisplay Case unterscheiden sich geringfügig. Die Unterschiede sind in der folgenden Tabelle angegeben. Alle weiteren Angaben finden Sie in Technische Daten auf Seite 41.

Technische Daten (nur IPEH-002262-KSM01)	
Größe	116,5 x 24,5 x 72,5 mm (B x H x T)
Gewicht	138 g
Schutzart (DIN EN 60529)	IP20
Gegensteckertypen	4-pol.: Phoenix Contact FMC 1,5/4-ST-3,5 1952283 10-pol.: Phoenix Contact FMC 1,5/10-ST-3,5 1952348