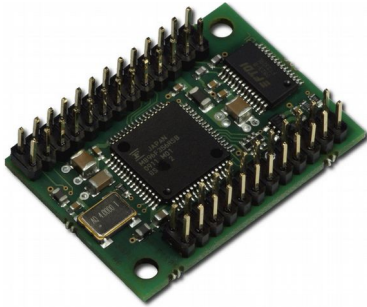


# Datenblatt Tiny-CAN M1

## USB CAN Adapter als Modul Version



### Technische Daten:

PC Interface	USB
Spannungsversorgung	5V-DC, ca. 36 mA
4 Status LEDs: Power/USB, Error, CAN-Rx, CAN-Tx	✓
CAN Interface	CAN „Line-Treiber“ nicht auf der Leiterplatte
Übertragungsraten	10 kBit/s – 1 MBit/s
Benutzerdefinierte CAN Übertragungsraten möglich	✓
CAN-Spezifikation	2.0A (11-Bit ID) und 2.0B (29-Bit ID)
Größe Empfangs-FIFO	384
Größe Sende-FIFO	72
Hardware Filter	4
Intervall Puffer	4
Firmware Update über PC möglich	✓
Silent Mode	✓
Transmit Message Request	✓
Automatic Retransmission disable	✗
Hardware Timestamp	✗
Maximale Buslast	ca. 50%
USB-Suspend-Mode	✗
Watchdog	✓ Software Watchdog
Prozessor	Fujitsu MB96F356RS (16-Bit/24MHz, 288 kB Flash, 12 kB RAM)
USB-Controller	FTDI FT232RL
Übertragungsgeschwindigkeit Prozessor <-> USB-Controller	3 Mbaud
Temperaturbereich	-20 - 85 °C
Abmessungen	29 x 41 x 15 mm
Gewicht	9g
Zertifizierung	CE
Software	Tiny-CAN API & SLCAN API
Betriebssysteme	Windows (ab XP), Apple (OS X), Linux (ab Kernel 2.6)

### Begriffserklärungen:

#### Intervall Puffer

Je nach Hardware ist eine bestimmte Anzahl von TxD-Puffern mit Intervalltimern vorhanden, diese ermöglichen das Versenden zyklischer Nachrichten in Echtzeit. Der normale Versand von CAN Nachrichten erfolgt über das „TxD-FIFO“ des Moduls.

#### Hardware Filter

Je nach Hardware ist eine bestimmte Anzahl von Filtern vorhanden, mit den Filtern ist es möglich, ein hohes Datenaufkommen auf dem USB-Bus zu reduzieren. Eine gefilterte Nachricht wird in dem dazugehörigen Puffer gespeichert, der Puffer wird immer wieder überschrieben. Alles was nach dem Durchlauf der Filter noch übrig bleibt landet im „Rx-D-FIFO“ des Moduls.

#### Silent Mode

Zusätzlich zu den Funktionen „CAN Start“ und „CAN Stop“ steht der Silent Mode zur Verfügung.

Der CAN-Controller ist nur passiv am Bus, empfangene CAN Nachrichten werden nicht mit einem ACK quittiert, bei Fehlern werden aktiv keine „Error Frames“ gesendet. Das Senden von CAN-Nachrichten ist nicht möglich. Der „Silent Mode“ wird verwendet, um einen CAN-Bus abzuhören, ohne diesen zu beeinflussen.

#### Transmit Message Request

Erfolgreich gesendete CAN Nachrichten werden in das Empfangs-FIFO zurückgeschrieben, um den Versand der Nachrichten zu bestätigen. Die Nachrichten werden durch das TxD Flag gekennzeichnet. Die Funktion kann per Software aktiviert/deaktiviert werden.

#### Automatic Retransmission disable

Das automatische wiederholte Versenden von CAN Nachrichten bei Fehlern wird unterbunden. Die Funktion kann per Software aktiviert/deaktiviert werden.

#### Hardware Timestamp

Ein Hardware Timestamp wird in der Hardware erstellt und beim Empfang bzw. dem erfolgreichen Versenden einer CAN Nachricht an der entsprechenden Stelle eingefügt.

#### CAN Safe

Vermeidet Fehler auf dem CAN Bus beim Abstecken des USB-Kabels oder Ausschalten des Computers

#### Maximale Buslast

Bei 1 MBit/s, Standard Frames mit 8 Byte Datenlänge, alle Hardware Filter aus

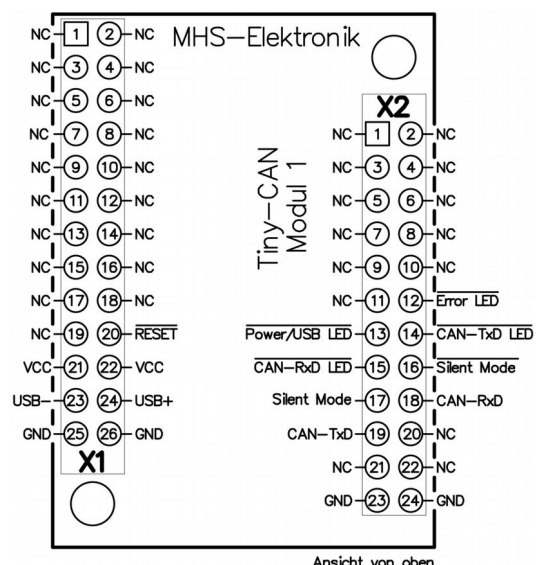
#### USB-Suspend-Mode

Reduzierung des Stromverbrauchs im Standby-Modus. Die Elektronik des Moduls wird zum größten Teil Spannungslos geschaltet, ein Empfang oder Senden von CAN Nachrichten ist nicht möglich.

#### Watchdog

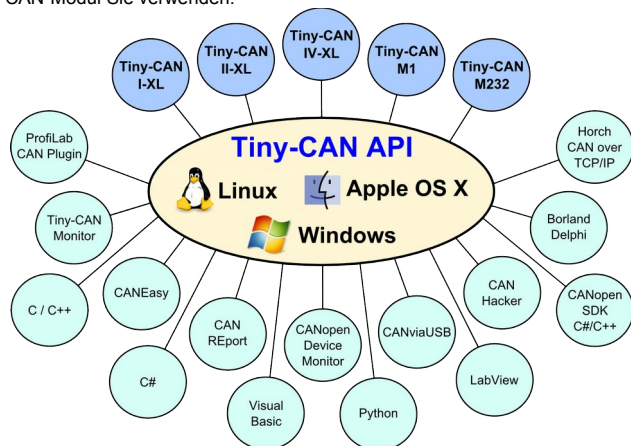
Überwachung des Prozessors

### Verdrahtung



## Tiny-CAN API

Die Tiny-CAN API ist unsere plattformübergreifende Schnittstelle für alle Tiny-CAN-Hardwareadapter. Sie nutzt die gleichen API-Aufrufe – ganz gleich, ob Sie mit Windows, OS X oder Linux arbeiten und welches Tiny-CAN-Modul Sie verwenden.

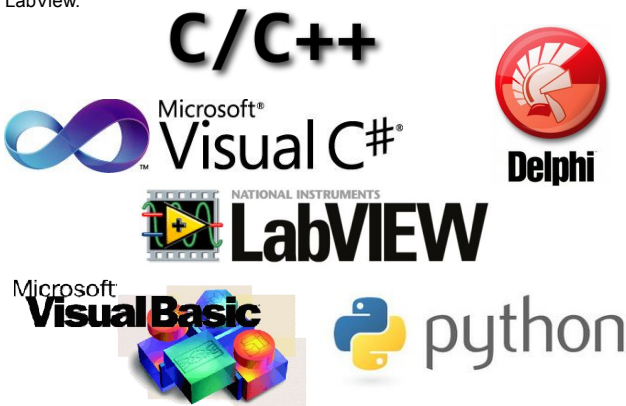


### Die Vorteile der Tiny-CAN-API auf einen Blick:

- Plug-and-Play-Unterstützung der Hardware: Der Treiber erkennt automatisch, wenn die Hardware an- oder abgesteckt wird.
- Ereignissteuerung über „Callback“ oder „Wait“ Funktionen
- Callback-Funktionen blockieren den „Main“-Thread des Treibers nicht
- Asynchrone Prozessaufrufe: Die API-Funktionen können von mehreren Threads und Callback-Funktionen aufgerufen werden.
- Größe für Empfangs- und Send-FIFO einstellbar
- Informationsdienst über Treiber und verwendete Hardware
- Hardware-spezifische Funktionen
- Vollständige Offenlegung der API

## Applikationsentwicklung

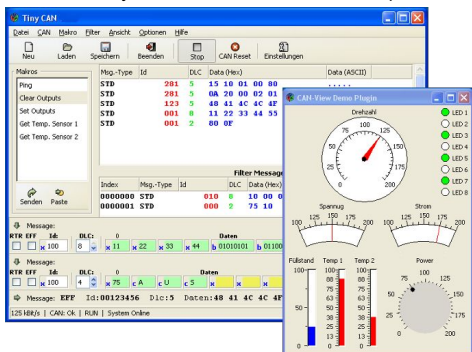
Das kostenlose Development Paket unterstützt folgende Programmiersprachen/Umgebungen: C/C++, C#, Visual VB, Python, Delphi, LabView.



Zum Development Paket gehören auch zahlreiche Applikationsbeispiele.

## Tiny-CAN-Monitor

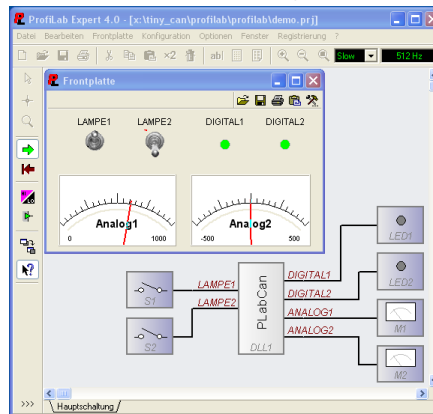
**Kostenloser CAN-Monitor** für Windows, Linux und OS X. Das Programm ist ein GNU - Open Source Projekt und wurde mit Gtk+ in der Sprache "C" entwickelt.



## ProfiLab CAN Plugin

Mit wenigen Mausklicks werden CAN-Botschaften in Signale aufgeteilt, die dann als analoger oder digitaler Ein/Ausgang im Schaltungssymbol erscheinen. Auch das verarbeiten von "Muxer"-Nachrichten ist ohne Probleme möglich. Die einzelnen Signale können dann mit allen im ProfiLab enthaltenen Elementen verbunden werden, z.B. Schalter, Taster, Displays, Analoge Zeigerinstrumente usw. Die Programmierung erfolgt rein auf grafischer Ebene.

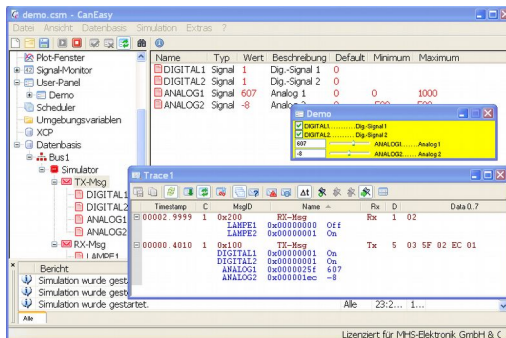
ProfiLab lässt sich nicht nur zur Visualisierung sondern auch zur Steuerung einsetzen. Zahlreiche Bauteile zur Verknüpfung und Verarbeitung von Signalen stehen zur Verfügung. Mit dem integrierten Compiler ist es möglich, fertige Projekte sozusagen in Stand-Alone-Anwendungen zu verwandeln. Diese, von der ProfiLab-Software erstellten Anwendungen, können Sie auf jedem Windows-PC starten, ohne dass dort die Originalsoftware installiert sein muss.



## Third Party Tools

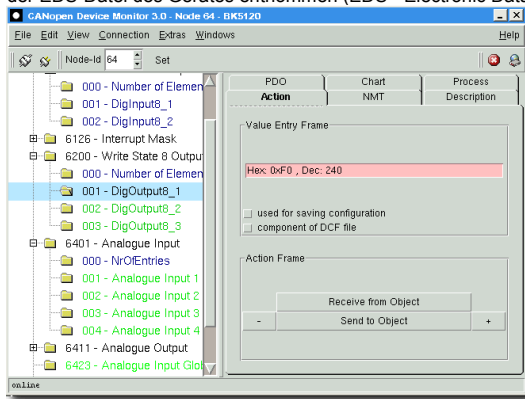
### CanEasy – Firma Schleißheimer

Can Easy 3.0 ist eine windowsbasierte Simulations-, Analyse- und Testumgebung für Entwicklung und Test von CAN-Bus-Systemen. Einfachheit ist das oberste Prinzip beim Einrichten, Bedienen und Erweitern des Systems.



### CANopen Device-Monitor – Firma Port

Der CANopen Device-Monitor von port dient der grafischen Inspektion und Konfiguration von CANopen Geräten im CANopen-Netzwerk. Die integrierte Skriptfähigkeit erlaubt neben dem Zugriff auf die implementierten CANopen-Dienste, die Ausführung von Steueraufgaben im Netzwerk und die Realisierung von Test- und Steuerapplikationen mit geringem Aufwand. Die Informationen zur Darstellung des Geräte-Objektverzeichnisses werden der EDS-Datei des Gerätes entnommen (EDS - Electronic Data Sheet).



**Weitere Third Party Tools:** CAN-REport, CANHacker, CANviaUSB, BUSMASTER, CarPort