



Benutzerhandbuch CANinterpreter

Versionshistorie

Version	Änderungen	Datum	Bearbeiter
V1.0	Erste Version	03.07.2013	ged
V1.0.3	Aktualisierung	20.09.2013	ged
V1.1	Beschreibung weitere PlugIn	28.02.2013	ged

Ablehnungshinweis

Die von emtas GmbH gelieferten Programme und Dokumentation werden mit großer Sorgfalt erstellt und in verschiedenen Szenarien geprüft und getestet. Trotzdem kann emtas GmbH keine Gewähr oder Haftung übernehmen, dass die Software oder die Dokumentation fehlerfrei bzw. für spezielle Anwendungszwecke geeignet ist. Insbesondere Beschreibung und die technischen Daten sind keine zugesicherten Eigenschaften im rechtlichen Sinne. Für Folgeschäden, welche aufgrund der Benutzung der Programme auftreten, wird daher jedwede juristische Verantwortung oder Haftung ausgeschlossen.

emtas GmbH hat das Recht Änderungen an den beschriebenen Produkten bzw. an der Dokumentation der Produkte ohne vorherige Mitteilung durchzuführen, wenn die Änderungen der Zuverlässigkeit, Qualitätssicherung oder dem technischen Fortschritt dienen.

Sämtliche Rechte an den Produkten einschließlich der Dokumentation liegen bei emtas GmbH. Weitergabe an Dritte oder die Vervielfältigung, auch in Auszügen, sind nur mit schriftlicher Genehmigung der emtas GmbH erlaubt. Ausgenommen davon die Kopien, welche zu Zwecken der Sicherung erstellt werden. Der Anwender trägt die Verantwortung, dass diese Kopien nicht in den Besitz Dritter gelangen.

Für Hinweise auf Fehler sind wir Ihnen dankbar und bitten um Benachrichtigung. Wir werden die Hinweise schnellstmöglich überprüfen und berücksichtigen.

Copyright

© 2014 emtas GmbH
Fritz-Haber-Str. 9
D-06217 Merseburg
Tel. +49 3461/79416-0
Fax. +49 3461/79416-10
service@emtas.de
<http://www.emtas.de>



Inhaltsverzeichnis

1 Allgemeines.....	4
2 Installation.....	4
2.1 Windows.....	4
2.2 Linux.....	4
3 Erste Schritte.....	4
4 Programmkomponenten.....	5
4.1 CAN View.....	5
5 Menü.....	6
6 Einstellungen.....	8
6.1 CAN-Einstellungen.....	8
6.2 Programmeinstellungen.....	9
7 PlugIns.....	9
7.1 CAN Object View.....	9
7.2 User-defined Interpretation.....	10
7.3 CANopen Interpretation (optional).....	11
7.4 CANopen Object View (optional).....	12
7.5 EnergyBus Interpretation (optional).....	12
7.6 EnergyBus Object View (optional).....	12
7.7 File Logger.....	12
7.8 Scripting Interpreter (optional).....	12
8 Support & Kontakt.....	15

1 Allgemeines

Vielen Dank für die Verwendung des CANinterpreters. Das Tool ist ein vielseitig einsetzbares CAN-Analysetool für Entwicklungs-, Test-, Diagnose- und Serviceaufgaben. Das Handbuch erläutert die Installation und Bedienung des Programms.

2 Installation

2.1 Windows

Zur Installation unter Windows führen Sie das Setup `setup_caninterpreter.exe` aus und folgen den Anweisungen des Setups. Das Setup richtet ein Icon oder eine Verknüpfung im Programmmenü ein, womit das Programm gestartet werden kann. Beim ersten Verbinden mit einem CAN-Interface wird nach dem Pfad der Lizenzdatei gefragt, welche dann zum Programm kopiert wird.

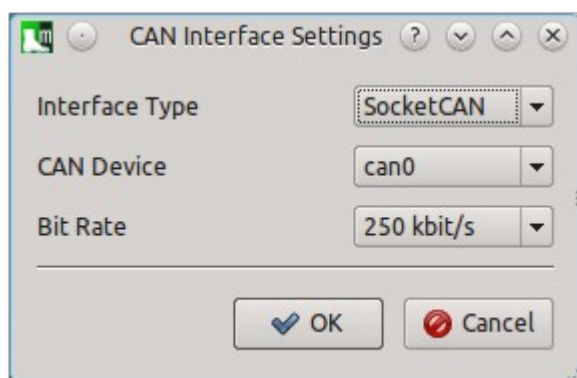
Ohne Lizenzdatei kann das Tool für eine Stunde mit 125 kBit/s zu Evaluierungszwecken verwendet werden.

2.2 Linux

Zur Installation unter Linux ist die ZIP-Datei `setup_caninterpreter.zip` in das gewünschte Verzeichnis zu kopieren und dort zu entpacken. Zum Start des Programms starten Sie das Skript `CANopenDeviceExplorer.sh` im Installationsverzeichnis. Beim ersten Verbinden mit einem CAN-Interface wird nach dem Pfad der Lizenzdatei gefragt, welche dann zum Programm kopiert wird.

3 Erste Schritte

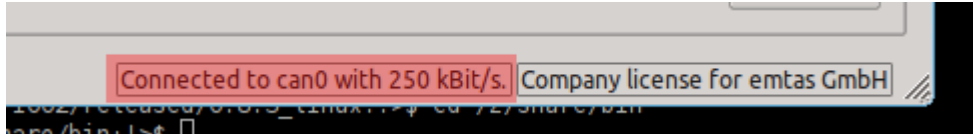
Der erste Schritt beim ersten Programmstart ist die Konfiguration des CAN-Interface. Öffnen Sie dazu die CANopen-Interface-Einstellungen unter dem Menüpunkt „Connection → CAN Interface Settings“. Es erscheint folgende Eingabemaske:



Wählen Sie nun den Typ der CAN-Schnittstelle, den Namen des CAN-Gerätes

und die Bitrate im CANopen-Netzwerk und bestätigen Sie den Dialog mit OK.

Verbinden Sie das Programm nun mit dem CAN-Interface über den Menüpunkt „Connection → Connect“. In der Statuszeile des Programms sehen Sie nun „Connected to ...“ sowie den Namen des CAN-Geräts und die aktuelle Bitrate.



4 Programmkomponenten

4.1 CAN View

Der CANinterpreter zeigt empfangene und gesendete CAN-Nachrichten an. Zum Senden von CAN-Nachrichten kann die untere CAN Tx-Tabelle verwendet werden. Die Werte für CAN-ID, DLC und die Daten können als Dezimalzahl oder als Hexadezimalzahl mit führenden 0x angegeben werden.

CAN Messages

CAN Rx Auto Scroll hex relative time Clear

	Time	CAN-ID	Type	Len	0	1	2	3	4	5	6	7
4	39.202000	1109/0x455	dB	4	0xff	0xff	0xff	0xff				
5	40.202000	1109/0x455	dB	4	0xff	0xff	0xff	0xff				
6	41.202000	1109/0x455	dB	4	0xff	0xff	0xff	0xff				
7	42.203000	1109/0x455	dB	4	0xff	0xff	0xff	0xff				
8	69.623000	0/0x000	dB	2	0x81	0x00						
9	69.628000	1824/0x720	dB	1	0x00							

CAN Tx

	Interval (ms)	ID	Type	Len	0	1	2	3	4	5	6	7	Name
1	1000	0x455	dB	4	0xff	0xff	0xff	0xff					Set Switches
2		0	dB	2	0x81	0							Reset Node
3													
4													
5													

Transmit

'Type' gibt das Format der CAN-Nachricht an:

- dB oder leer CAN base data message (Standard CAN Nachricht)
- rB CAN base RTR message
- dE CAN extended data message
- rE CAN extended RTR message

Es können bis zu 16 Sende-Nachrichten definiert werden. Die jeweils aktive

Nachricht wird bei einem Klick auf den Transmit-Button gesendet. Ist in der Spalte 'Interval(ms)' ein Wert ungleich 0 eingetragen, so wird die Transmit-Nachricht zyklisch gesendet.

Im Menu des CANinterpreters kann ein Filter nach CAN-IDs definiert werden. Dabei kann eine Liste von CAN-IDs definiert werden, welche entweder angezeigt oder ignoriert werden sollen. Die Liste kann als Liste einzelner Ids (z.B: 100,200,300,400) oder als Bereich (z.B. 0x100-0x200) oder kombiniert angegeben werden. (Beispiel: 0x100-0x222,0x400,0x720, 0x740).

Der Filtertyp legt das Verhalten des Filters fest:

- PASS – es werden nur die definierten CAN-IDs angezeigt
- REJECT – es werden die definierten CAN-IDs ausgefiltert und alle anderen werden angezeigt.

Die Zeitstempel der Empfangsnachrichten können jeweils absolut und relativ (zur letzten Nachricht) dargestellt werden.

Die aufgezeichneten CAN-Nachrichten können über File → Export CAN-Logging in Textdateien exportiert werden.

Das Format der aufgezeichneten CAN-Telegramme hat folgende Form in der exportierten Textdatei:

```
3.652109 0x5a0/1440 (8): 43 18 10 02 15 00 08 00
3.654306 0x620/1568 (8): 40 18 10 04 00 00 00 00
3.653302 0x5a0/1440 (8): 43 18 10 03 00 00 00 00
Zeitstempel
      CAN-ID
          DLC
              Daten im hexadezimalen Format
```

Die Genauigkeit der Zeitstempel hängt vom verwendeten Betriebssystem und CAN-Interface ab. Der Sendezeitstempel ist nicht für alle CAN-Schnittstellen verfügbar.

5 Menü

Das Menü bietet Zugriff auf verschiedene Funktionen und Einstellungen des CANinterpreters.

File

- **Export CAN Logging**
Export der CAN-Telegramme als Text-Datei. Späterer Import und Interpretation in den PlugIns ist möglich.
- **Quit**
Beendet das Programm.

Connection

- **CAN Interface Settings**
Dialog zur Konfiguration der CAN-Schnittstelle und der aktuellen Bitrate
- **Connect**
Aufbau einer Verbindung mit dem CAN über die konfigurierte CAN-Schnittstelle
- **Disconnect**
Trennen der Verbindung mit der CAN-Schnittstelle

Settings

- **Filter Settings**
Konfiguration des Filters der CAN-Nachrichten. Der Filter wirkt sich nur auf die Hauptkomponente aus. Alle Plugins haben separate Filtermöglichkeiten.
- **Options**
Öffnet den Einstellungsdialog zur Konfiguration verschiedener Programmeinstellungen .
- **Save**
Die aktuellen Einstellungen werden gespeichert. Unter Setting → Options → „Save settings automatically at exit“ kann eine automatische Speicherung der Einstellungen beim Beenden des Programms aktiviert werden.
- **Export Settings**
Export der aktuellen Einstellungen in eine Konfigurationsdatei. Damit können beispielsweise für verschiedene Anwendungsszenarien unterschiedliche Konfigurationen gespeichert werden.
- **Import Settings**
Import der Einstellungen aus einer Konfigurationsdatei.
- **Update Licence File**
Dialog zur Auswahl einer (neuen) Lizenzdatei. Der Inhalt der bisherigen und der neuen Lizenzdatei wird angezeigt und die neue Datei kann kopiert werden.
- **Check for Updates**
Abfrage an den Webserver, ob eine neue Version des CANinterpreters verfügbar ist. Dabei werden außer der IP-Adresse keine Daten von Ihrem Computer übermittelt.

PlugIns

Aktierungsmöglichkeit für diverse Erweiterungen des CANinterpreters. Die Verfügbarkeit der PlugIns hängt von den jeweils lizenzierten PlugIns ab.

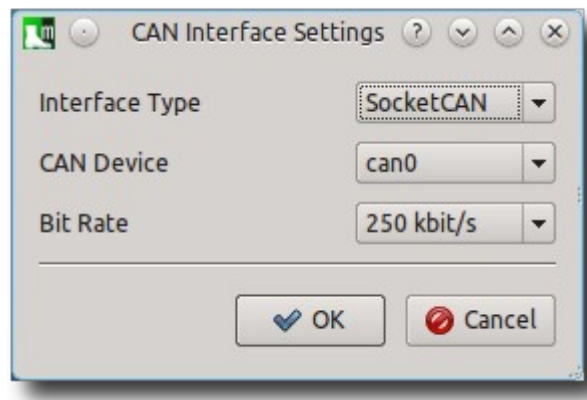
Help

- **Manual**
Anzeige des Handbuchs im PDF-Format.
- **About**
Anzeige des About-Dialogs mit Lizenzinformationen.
- **About Qt**
Information über das Qt-Framework mit Lizenzinformation der verwendeten Qt-Komponenten.

6 Einstellungen

6.1 CAN-Einstellungen

Der CAN-Einstellungsdialog dient zur Konfiguration der CAN-Schnittstelle.



- **Interface Type**
Auswahl der Art der CAN-Schnittstelle. Unter Linux werden aktuell SocketCAN und can4linux und unter Windows diverse CAN-Interfaces unterstützt.
- **CAN-Device**
Je nach Art des Interface Typs kann hier der Name des CAN-Geräts eingestellt werden. Übliche Namen für SocketCAN sind can0, can1, ... vcan0 und bei beispielsweise PCANBasic usb1 ... usb8.
- **Bit Rate**
Konfiguration der Bitrate im CAN-Netzwerk. Stellen Sie hier die Bitrate ein, welche auch für alle anderen Geräte im CAN-Netzwerk eingestellt ist. Bitte beachten Sie, dass bei der Verwendung von SocketCAN die Bitrate

vor dem Start des Programms eingestellt werden muss.

- **Advanced Settings**

Bei verschiedenen CAN-Interface-Typen besteht die Möglichkeit weitere Optionen einzustellen. Diese Optionen hängen vom CAN-Interface ab.

Mögliche Einstellungen sind:

- Listen-Only-Mode
- Nutzer-spezifische Bitrate
- Aktivierung des Abschlußwiderstands

6.2 Programmeinstellungen

Der Optionendialog bietet Zugriff auf verschiedene Programmeinstellungen.

General – Allgemeine Einstellungen

- Warning when exit with active CAN connection
Konfiguration, ob eine Warnung angezeigt werden soll, wenn das Programm bei bestehender CAN-Verbindung beendet wird.
- Save Settings automatically at exit
Konfiguration, ob die Einstellungen automatisch beim Beenden des Programms gespeichert werden sollen.
- Auto Connect to CAN after startup
Automatischer Aufbau einer Verbindung zum CAN-Interface beim Start des Programms

Scripting

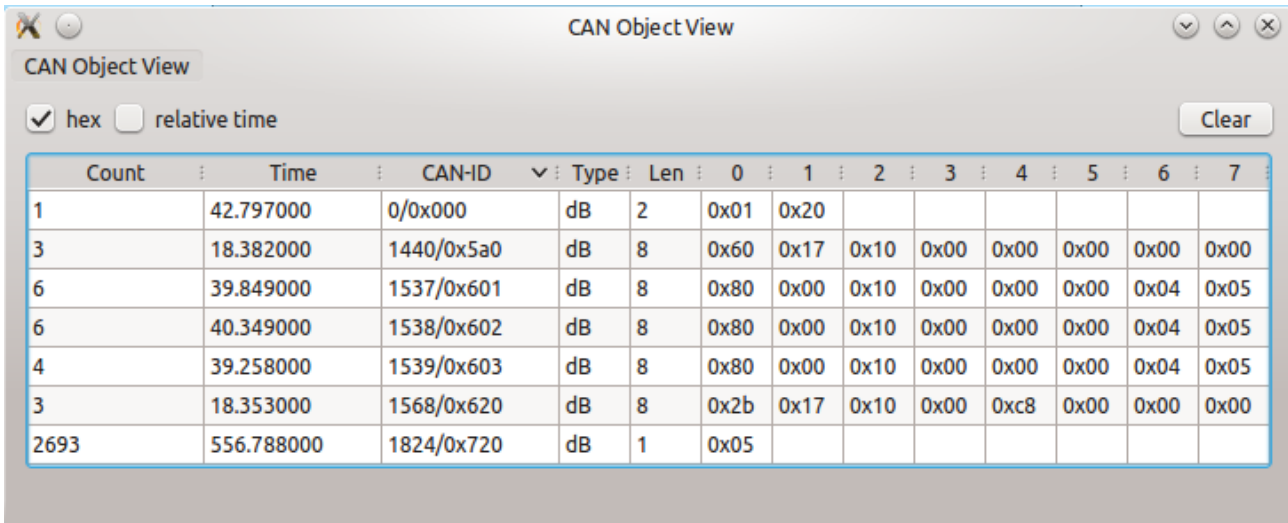
- Autostart Script at connect
Das Autostart-Skript wird automatisch gestartet, wenn der Scripting Interpreter aktiv ist und eine Verbindung mit dem CAN hergestellt wird. In Verbindung mit „Auto Connect to CAN after startup“ wird das Skript automatisch nach dem Start des Programms gestartet.

7 PlugIns

7.1 CAN Object View

Das CAN Object View zeigt alle empfangenen CAN-Nachrichten in der

sogenannten „Object View“-Ansicht an. Dabei werden tabellarisch alle empfangenen CAN-IDs mit den letzten empfangenen Werten angezeigt.



The screenshot shows a window titled "CAN Object View" with a "Clear" button and radio buttons for "hex" (checked) and "relative time". Below is a table with columns: Count, Time, CAN-ID, Type, Len, and 8 data bytes (0-7).

Count	Time	CAN-ID	Type	Len	0	1	2	3	4	5	6	7
1	42.797000	0/0x000	dB	2	0x01	0x20						
3	18.382000	1440/0x5a0	dB	8	0x60	0x17	0x10	0x00	0x00	0x00	0x00	0x00
6	39.849000	1537/0x601	dB	8	0x80	0x00	0x10	0x00	0x00	0x00	0x04	0x05
6	40.349000	1538/0x602	dB	8	0x80	0x00	0x10	0x00	0x00	0x00	0x04	0x05
4	39.258000	1539/0x603	dB	8	0x80	0x00	0x10	0x00	0x00	0x00	0x04	0x05
3	18.353000	1568/0x620	dB	8	0x2b	0x17	0x10	0x00	0xc8	0x00	0x00	0x00
2693	556.788000	1824/0x720	dB	1	0x05							

Die Tabelle kann der Anzahl, der Empfangszeit und der CAN-ID sortiert werden. Eine Filterung ist analog zum CAN View verfügbar und das CAN Object View PlugIn gehört zum Standardlieferumfang des CANinterpreters.

7.2 User-defined Interpretation

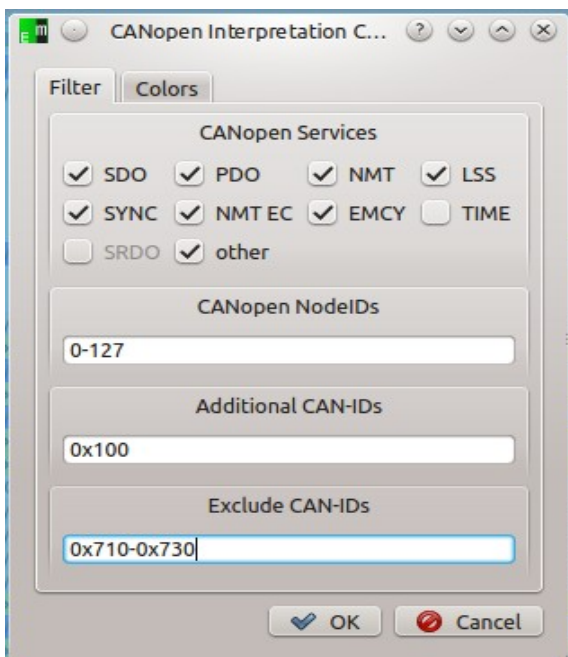
Dieses PlugIn ermöglicht eine Anwender-spezifische Interpretation der CAN-Daten.

7.3 CANopen Interpretation (optional)

Time Stamp	CAN-ID	Type	Node-Id	Data	Interpretation
1517.332000	1824/0x720	Error Control	32	05	t=0 operational
1517.506000	1568/0x620	SDO	32	2b 17 10 00 50 c3 00 00	w Req exp 1017:0 Val 0xc350 initiate Download Request Access Type: write Transfer Type: expedited Index: 1017 Subindex: 0 Object: Producer heartbeat time
1517.531000	1824/0x720	Error Control	32	05	t=0 operational toggle Bit 0 Node State operational
1517.531000	1440/0x5a0	SDO	32	60 17 10 00 00 00 00 00	w Res - 1017:0
1535.595000	0/0x000	NMT	32	81 20	res node
1535.596000	1824/0x720	Error Control	32	00	t=0 Boot Up
1536.019000	0/0x000	NMT	32	82 20	res comm
1536.026000	1824/0x720	Error Control	32	00	t=0 Boot Up
1536.483000	0/0x000	NMT	32	02 20	stop
1536.987000	0/0x000	NMT	32	01 20	start
1537.020000	416/0x1a0	PDO	32	00 00	0x0
1588.153000	1568/0x620	SDO	32	40 17 10 00 00 00 00 00	r Req - 1017:0
1588.160000	1440/0x5a0	SDO	32	4b 17 10 00 00 00 00 00	r Res exp 1017:0 Val 0x0
1588.319000	1568/0x620	SDO	32	40 17 10 00 00 00 00 00	r Req - 1017:0
1588.326000	1440/0x5a0	SDO	32	4b 17 10 00 00 00 00 00	r Res exp 1017:0 Val 0x0
1595.584000	0/0x000	NMT	32	81 20	res node
1595.617000	1824/0x720	Error Control	32	00	t=0 Boot Up
1596.264000	0/0x000	NMT	32	01 20	start
1596.280000	416/0x1a0	PDO	32	00 00	0x0 Status word (6041:0) = 0x0

Der CANopen Interpreter interpretiert alle empfangene CAN-Nachrichten entsprechend dem CANopen-Protokoll und zeigt den CANopen-Typ der Nachricht, die Quell- oder Zielknotennummer sowie dienstspezifische Daten in lesbarer Form an. Anhand der EDS- bzw. DCF-Dateien der Geräte in Netzwerk ist zudem eine Interpretation der PDO-Dateninhalte entsprechend des Mappings des Senders möglich.

Eine Filterung der CAN-Nachrichten ist nach verschiedenen Kriterien möglich:



- **CANopen Dienste:** Es werden nur die aktivierten CANopen-Dienste angezeigt
- **CANopen NodeID:** Es werden nur die ausgewählten Knotennummern angezeigt. Eine Definition von Bereichen wie z.B. (1,2,30-50) ist möglich.
- **Additional CAN-IDs:** CAN-IDs, welche durch die vorherigen Filter ausgeschlossen wurden, können selektiv wieder zugelassen werden. Eine Definition von Bereichen wie z.B.(1-0x100,0x400) ist möglich.
- **Exclude CAN-IDs:** CAN-IDs, welche

die vorherigen Filter passieren, können selektiv ausgeschlossen werden. Eine Definition von Bereichen wie z.B..1-0x100,0x400) ist möglich.

Interpretierte Nachrichten können als Textdatei exportiert und wieder importiert werden. Zudem ist der Import von uninterpretierten CAN-Nachrichten im Format des CANinterpreters möglich.

7.4 CANopen Object View (optional)

Der CANopen Object View kombiniert die CANopen Interpretation mit einer Objektview-Ansicht so das alle empfangene CAN-IDs mit ihren letzten Werten interpretiert dargestellt werden

7.5 EnergyBus Interpretation (optional)

Die optionale EnergyBus Interpretation zeigt den Zustand von Geräten in einem EnergyBus (CiA 454)-Netzwerk an.

7.6 EnergyBus Object View (optional)

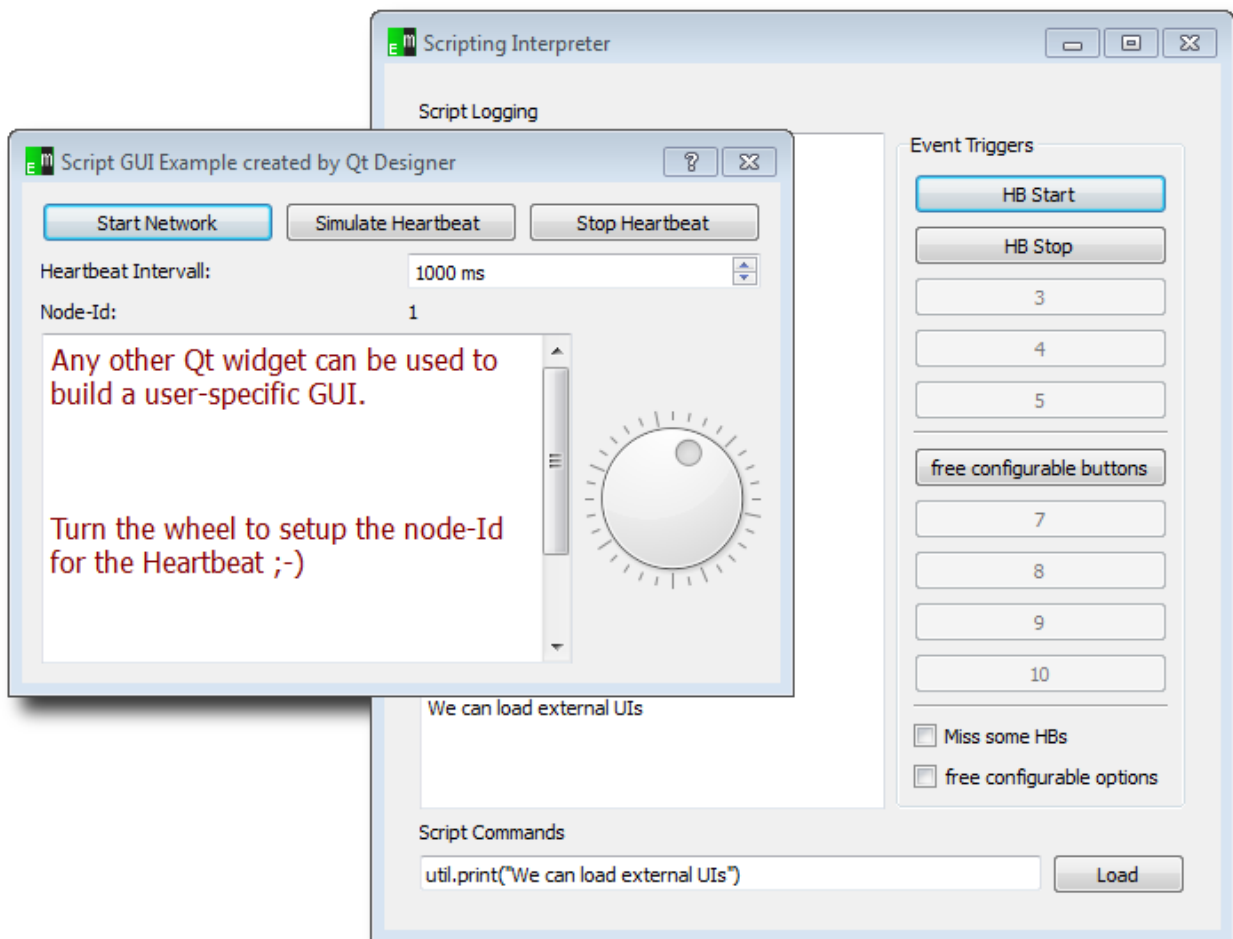
Der EnergyBus Object View kombiniert die EnergyBus Interpretation mit einer Objektview-Ansicht so das alle empfangene CAN-IDs mit ihren letzten Werten entsprechend der EnergyBus-Spezifikation interpretiert dargestellt werden

7.7 File Logger

Der File Logger ermöglicht das direkte Aufzeichnen von CAN-Nachrichten in Logdateien entsprechend bestimmter Triggerbedingungen

7.8 Scripting Interpreter (optional)

Im Scripting-Interpreter können QtScript(Javascript)-Skripte mit speziellen Kommandoerweiterungen für CANopen ausgeführt werden.



Das Erstellen und die Benutzung eigener Benutzeroberflächen ist mithilfe des Qt Designers möglich.

Alle zusätzlichen CAN/CANopen-spezifischen Kommandos sind in dem separaten Dokument „CANopen Scripting Interpreter - API Reference“ (cde_script_api.pdf) erläutert.

Nachfolgendes Beispiel zeigt die Verwendung eines QtScript-Skripts:

```
// print something to console and set device to operational
util.print("Test of simple device");
nmt.preopNetwork();
nmt.startNode(32);
i = 0;
util.print("We are in " + util.pwd());

// set node id for SD0 access
sdo.setNodeId(32);

// loop over objects 0x4000 to 0x4010
for (object = 0x4000; object < 0x4005; object ++){
    str = "Test object ";
    str = str + object;
    util.print(str);
}
```

```

// write value to object 0x4000..
result = sdo.write(object, 0x0, UNSIGNED32, i);
if (result == "SDO_OK") {
    util.print(" Write OK");
} else {
    util.print(" Write NOT OK");
}
// wait a bit to allow device update its internal values
util.msleep(10);

// read from 0x4100.. and expect same value
result = sdo.read(object+0x100, 0x0, 0x07);
if (result == i) {
    util.print("Read OK");
} else {
    util.print(" Read NOT OK");
}
i++;
}

// user defined function that can be called from Scripting tab
function urk(count) {
    for (i = 0; i < count; i++) {
        nmt.startNetwork();
        nmt.stopNetwork();
        var dlc = 4;
        var canId = 0x100 + i;
        can.sendBaseFrame(canId, dlc, 1 , 2 , 3 ,4 , 0, 0, 0 ,0);
    }
}
// call user defined function
urk(4);
// setup cyclic timer every 2 seconds
timerId = util.every(2000, "urk(10)" );

```

Das Beispiel liegt dem CANinterpreter als example1.js bei.

Skriptkommandos werden in der Eingabezeile unter 'Script Commands' eingegeben. Komplette Skriptdateien können geladen werden, wenn der Pfad zur Skriptdatei in der Eingabezeile steht und der Button 'Load' geklickt wird.

Die Eingabezeile merkt sich die eingegebenen Kommandos, und mit den Cursor-Tasten nach oben und unten können ältere Kommandos erneut ausgewählt werden. Die Kommandohistorie wird beim Beenden des Programms gespeichert.

8 Support & Kontakt

Bei Fragen und Problemen unterstützt Sie unser Supportteam gern per Email (service@emtas.de) oder telefonisch unter der 03461/794160. Falls sich das Gerät anders verhält als erwartet ist oft ein Logging der CAN-Kommunikation bei der Analyse der Ursachen hilfreich. Bitte schicken Sie uns daher bei solchen Fragen einen Mitschnitt der CAN-Kommunikation per Email; idealerweise auch vor einer telefonischen Kontaktaufnahme.