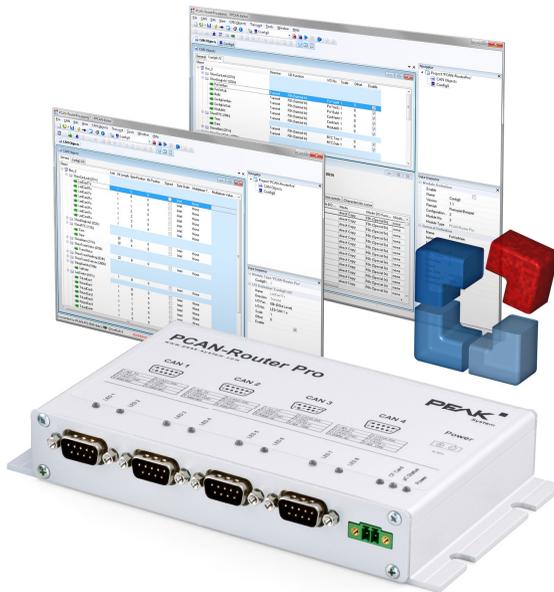


PCAN-Router Pro

4-Kanal-CAN-Router mit Datenlogger

Konfigurations-Tutorial



Dokumentversion 1.2.0 (2020-10-20)

PEAK
System

Berücksichtigte Produkte

Produktbezeichnung	Ausführung	Artikelnummer
PCAN-Router Pro		IPEH-002212
PCAN-Explorer 5		IPES-005028
PPCAN-Editor 2, PCAN-View, PEAK-Converter		

PCAN ist eine eingetragene Marke der PEAK-System Technik GmbH.
Andere Produktnamen in diesem Dokument können Marken oder eingetragene
Marken der jeweiligen Eigentümer sein. Diese sind nicht ausdrücklich durch „™“
oder „®“ gekennzeichnet.

© 2020 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
info@peak-system.com

Dokumentversion 1.2.0 (2020-10-20)

Inhalt

1	Einleitung	5
1.1	Voraussetzungen für den Betrieb	5
2	Begriff Konfiguration	7
2.1	Möglichkeiten der Konfiguration	7
2.2	Skalierung	8
2.3	CAN-Gateway-Dienste	8
2.4	Default-Werte	9
2.5	Funktionsblöcke	9
2.6	Ereignisgesteuertes Aussenden von CAN-Nachrichten	9
2.7	Kennlinien	9
3	Aufgabenliste	11
4	Lösungswege mit Erläuterungen	13
4.1	Aufgabe 1a: Durchreichen aller Nachrichten von CAN-1 nach CAN-4	13
4.2	Aufgabe 1b: Durchreichen von definierten Nachrichten von CAN-1 nach CAN-4	19
4.3	Aufgabe 1c: Durchreichen aller Nachrichten von CAN-1 nach CAN-4 mit Ausnahmen	20
4.4	Aufgabe 2a: Aufzeichnen aller Empfangsnachrichten auf CF-Karte	21
4.5	Aufgabe 2b: Extrahieren von Trace-Dateien von der CF-Karte auf den PC	23
4.6	Aufgabe 3a: Definieren von CAN-Nachrichten (z. B. Ausgabe einer Systemvariablen)	24
4.7	Aufgabe 3b: Umsetzen einer CAN-ID	31
4.8	Aufgabe 3c: (Variation 3b) Nur dann Senden, wenn Quellnachricht empfangen wurde	36

4.9	Aufgabe 3d: (Variation 3a) Nur Senden bei externer Anforderung	39
4.10	Aufgabe 4a: CAN-Signal mittels Scale und Offset bearbeiten	41
4.11	Aufgabe 4b: CAN-Signal mit Funktionsblock "Kennlinie" bearbeiten	43
4.12	Aufgabe 5a: LED-Aktivität bei CAN-Senden/Empfangen	50
4.13	Aufgabe 5b: LED manuell / bedingt ansteuern	53
4.14	Aufgabe 5c: LED extern ansteuern	58
4.15	Aufgabe 5d: Beeper ansteuern (Dauerton)	60
4.16	Aufgabe 5e: Beeper ansteuern (Tonfolge)	62
4.17	Aufgabe 6a: Ausgabe von Datum/Uhrzeit (Hardware-Diagnose)	64
4.18	Aufgabe 6b: Setzen von Datum/Uhrzeit (Hardware-Diagnose)	66
4.19	Aufgabe 6c: Ausgabe der Modul-ID (Hardware-Diagnose)	67
4.20	Aufgabe 6d: Ausgabe Firmwarestand und Konfigurations-Version (Hardware-Diagnose)	70
4.21	Aufgabe 7a: Sleep/Wake-Up über CAN	71
4.22	Aufgabe 7b: Sleep/Wake-Up über Hardware-Pin	72
4.23	Aufgabe 7c: Sleep/Wake-Up über Alarmzeit	73
4.24	Aufgabe 8a: Ändern der Bitrate	74
4.25	Aufgabe 9a: Multiplexer-Nachricht automatisch senden	76
4.26	Aufgabe 9b: Multiplexer-Nachricht auf Anfrage senden	79

1 Einleitung

Die Verwendung des PPCAN-Editors setzt eine Mindest-Qualifikation des Benutzers hinsichtlich Hardware-Verständnis und Programmierkenntnissen voraus.

Dieses Tutorial wendet sich daher an Besitzer eines PCAN-Router Pro, die komplexere Konfigurationen des Geräts benötigen und über belastbare Grundkenntnisse der Elektronik und Informatik verfügen.

Zunächst sollte der für Kunden kostenlose PPCAN-Editor anhand dieses Tutorials getestet werden. Die Häufigkeit von Verständnisproblemen (technischer Natur!) beim Durcharbeiten des Dokuments kann möglicherweise als Entscheidungshilfe für einen zukünftigen Einsatz des Editors herangezogen werden. Im abschlägigen Fall bietet die PEAK-System Technik GmbH Ihren Kunden einen Konfigurationsservice gemäß detaillierter Spezifikation.

1.1 Voraussetzungen für den Betrieb

Zum sinnvollen Bearbeiten dieses Tutorials bzw. zum Lösen der Aufgaben steht ein Gerät PCAN-Router Pro inkl. Spannungsversorgung zur Verfügung. Dessen CAN-Busse sind über PCAN-PC-Interfaces an einen Computer angeschlossen und korrekt terminiert (z. B. mit den internen DIP-Schaltern).

- Mindestens zwei CAN-Busse (z. B. 1 und 4) über PEAK-Interfaces an den PC anschließen, mit je 500 kbit/s
- Eine speziell formatierte CF-Karte (1 GByte, z. B. aus dem Lieferumfang) ist eingesteckt
- Die Software PPCAN-Editor 2 ist installiert

- └ Als CAN-Gegenstelle ist ein PCAN-View oder besser: ein PCAN-Explorer auf dem PC installiert
- └ Die beiliegende CF-Management Software (PEAK-Converter) und ein CF-Kartenleser sind installiert

Das Gerät PCAN-Router Pro stellt folgende Ressourcen zum Verknüpfen bereit:

- └ Die Geräte-ID (4-Bit, 0..15 dez.) kann im Innern des Geräts per Schalter verändert werden (siehe PCAN-Router Pro-Bedienungsanleitung)
- └ 4 CAN-Busse (#1 .. #4), Wake-Up-fähig
- └ Unterschiedliche CAN-Transceiver (HS, HS-OPTO¹, LS-DW, LS-SW)
- └ 1 CF-Karte (als virtueller CAN-Bus #5), konfigurierbar
- └ CAN-Bitraten² (10k; 20k; 33,3k; 47,6k; 50k; 83,3k; 95,2k; 100k; 125k; 250k; 500k; 1M)
- └ CAN-Nachrichten (11-Bit oder 29-Bit)
- └ Je 2 Leuchtdioden pro CAN-Bus (insgesamt 8, Status kann geschrieben und gelesen werden)
- └ Uhrzeit (RTC)
- └ Beeper
- └ Sleep-Mode durch Software-Schalter

¹ Verfügbarkeit HS-opto anfragen.

² Bitraten können frei eingestellt werden, die tatsächlichen Funktionen sind abhängig vom ausgestatteten Transceiver-Typ.

2 Begriff Konfiguration

Die Mikrocontroller-basierten Geräte von PEAK-System bieten die Möglichkeit, alle verbauten Schnittstellen miteinander zu verknüpfen. Hierzu stellt die Firmware sogenannte Funktionsblöcke bereit, mit denen die Hardware-Ressourcen virtuell zu einer Konfiguration verschaltet werden. Zum Erstellen und Editieren von Konfigurationen stellt PEAK-System den PPCAN-Editor 2 für Windows mit jedem Gerät kostenlos bereit.

Die so erstellte Datei mit der darin enthaltenen Konfiguration wird zunächst auf dem PC gespeichert, dann dem Gerät per Upload (via CAN) bekannt gegeben und dort nichtflüchtig gespeichert. Manche Geräte können mehrere Konfigurationen speichern: die jeweils gültige wird mittels eines Wahlschalters selektiert. Der Wahlschalter bestimmt gleichzeitig die Geräte-ID und den Speicherort der Konfiguration innerhalb des nichtflüchtigen Speichers.

Die mit dem PPCAN-Editor 2 erstellten Dateien können mehrere Konfigurationen enthalten. Die Geräte-ID bestimmt dabei, welche davon bei Modulstart ausgeführt wird. Daraus ergibt sich z. B. die Möglichkeit, mehrere gleiche Geräte mit unterschiedlicher ID an einem CAN-Bus zu betreiben und die identische Konfigurations-Datei auf alle Geräte zu uploaden. Die unterschiedliche ID sorgt dann dafür, dass jedes Gerät seine spezielle Konfiguration aus dem nichtflüchtigen Speicher lädt und dementsprechend seine Aufgabe ausführt.

2.1 Möglichkeiten der Konfiguration

Zur Verknüpfung von Schnittstellen gibt es die einfache Skalierung von Größen sowie die Methoden CAN-Gateway-Dienste, Default-Werte, Funktionsblöcke, ereignisgesteuertes CAN-Senden, zeitgesteuerte Aktivitäten und Kennlinien. Bei Geräten mit nur einem

CAN-Bus entfällt der Gateway-Dienst, und zeitgesteuerte Aktivitäten sind ebenfalls nicht mit jeder Hardware möglich. Die tatsächlich vorhandenen Ressourcen eines Geräts werden dem Editor durch eine spezielle Profildatei mitgeteilt, damit dieser die Verknüpfungsmöglichkeiten entsprechend freischaltet bzw. einschränkt. Die verfügbaren Ressourcen können dem Handbuch der jeweiligen Hardware entnommen werden. (Aktuelle Handbücher stehen auf unserer Webseite als kostenloser Download zur Verfügung).

2.2 Skalierung

Das grundlegendste Mittel der Manipulation von Größen ist die Verwendung der vier Grundrechenarten. Dies wird mit den Parametern SCALE und OFFSET gesteuert, die von der bekannten Geradengleichung aus der Mathematik übernommen sind. Dabei ist der Parameter SCALE für Multiplikation (wenn > 1) bzw. Division (wenn < 1) zuständig. Der Parameter OFFSET bewirkt eine Addition (wenn > 0 , positiv) bzw. Subtraktion (wenn < 0 , negativ). Als neutrale Voreinstellung ist daher SCALE=1 und OFFSET=0 gewählt.

2.3 CAN-Gateway-Dienste

Eingehende Nachrichten auf dem einen CAN-Bus werden selektiv auf dem anderen CAN-Bus ausgegeben. Oder sie können auf dem gleichen CAN-Bus mit anderer ID ausgegeben werden (z. B. Umsetzung 11-Bit \leftrightarrow 29-Bit). Oder eine eingehende Nachricht kann das Versenden einer beliebigen anderen Nachricht auslösen, wobei beide Nachrichten inhaltlich nichts miteinander zu tun haben.

2.4 Default-werte

Durch hier festgelegte Parameter kann der Zustand des Moduls nach dem Einschalten bestimmt werden, wie z. B. die Bitrate des CAN-Busses, Aktivierung permanenter Routings, Loggin-Einstellungen, den logischen Zustand von Leitungen, Leuchtdioden, etc.

2.5 Funktionsblöcke

Falls die einfache Manipulation von Messgrößen etc. mittels SCALE und OFFSET nicht ausreicht, stehen sogenannte Funktionsblöcke mit erheblich komplexeren Möglichkeiten zur Verfügung. Dies reicht vom Umsetzen per X/Y-Tabelle über HystereseFunktionen, Delays, Zähler, Tiefpass-Filter, diverse mathematische und logische Operatoren bis zum komplexen PIDT1 Regelkreis. Funktionsblöcke können sequentiell oder bedingt abgearbeitet werden.

2.6 Ereignisgesteuertes Aussenden von CAN-Nachrichten

Falls CAN-Nachrichten nur bei bestimmten Gelegenheiten gesendet werden sollen, steht außerdem ein Repertoire an Triggerbedingungen zur Verfügung. CAN-Nachrichten können außerdem von extern angefordert werden (RTR-Nachrichten).

2.7 Kennlinien

Hier lassen sich 2 bis 31 X-Werte als Eingangsparameter eintragen, denen je ein Y-Wert als Ausgangsparameter zugeordnet ist. Für den Wertebereich zwischen zwei X-Werten wird das Y-Ergebnis linear interpoliert. Anders ausgedrückt: So lassen sich (wie bei der Skalierung)

rung) SCALE und OFFSET für bis zu 31 Abschnitte eines Wertebereichs einzeln einstellen. Damit ist es möglich, Kurvensegmente in ihrer Steigung zu beeinflussen, um z. B. Plateaus zu definieren oder nichtstetige Funktionen nachzubilden.

3 Aufgabenliste

Durch die Lösung der hier aufgelisteten Aufgaben lässt sich ein Überblick über die vielfältigen Möglichkeiten der PCAN-Hardware (hier speziell des PCAN-Router Pro) erhalten.

- └ 1a) Durchreichen aller Nachrichten von CAN-1 nach CAN-4
- └ 1b) Durchreichen von definierten Nachrichten von CAN-1 nach CAN-4
- └ 1c) Durchreichen aller Nachrichten von CAN-1 nach CAN-4 mit Ausnahmen
- └ 2a) Aufzeichnen aller Empfangsnachrichten auf CF-Karte
- └ 2b) Extrahieren von Trace-Dateien von der CF-Karte auf den PC
- └ 3a) Definieren von CAN-Nachrichten (z. B. Ausgabe einer Systemvariablen)
- └ 3b) Umsetzen einer CAN-ID
- └ 3c) (Variation 3b) Nur dann Senden, wenn Quellnachricht empfangen wurde
- └ 3d) (Variation 3a) Nur Senden bei externer Anforderung
- └ 4a) CAN-Signal mittels Scale und Offset bearbeiten
- └ 4b) CAN-Signal mit Funktionsblock Kennlinie bearbeiten
- └ 5a) LED-Aktivität bei CAN-Senden/Empfangen
- └ 5b) LED manuell / bedingt ansteuern
- └ 5c) LED extern ansteuern
- └ 5d) Beeper ansteuern (Dauerton)
- └ 5e) Beeper ansteuern (Tonfolge)
- └ 6a) Ausgabe von Datum/Uhrzeit (Hardware-Diagnose)

- └ 6b) Setzen von Datum/Uhrzeit (Hardware-Diagnose)
- └ 6c) Ausgabe der Modul-ID (Hardware-Diagnose)
- └ 6d) Ausgabe Firmwarestand und Konfigurations-Version (Hardware-Diagnose)
- └ 7a) Sleep/Wake-Up über CAN
- └ 7b) Sleep/Wake-Up über Hardware-Pin
- └ 7c) Sleep/Wake-Up über Alarmzeit
- └ 8a) Ändern der Bitrate
- └ 9a) Multiplexer-Nachricht automatisch senden
- └ 9b) Multiplexer-Nachricht auf Anfrage senden

4 Lösungswege mit Erläuterungen

Weitere Information zur Benutzung des PPCAN-Editor 2 finden Sie in der Hilfe, die Sie im Programm über das Menü **Hilfe** oder die Taste **F1** erreichen.

4.1 Aufgabe 1a: Durchreichen aller Nachrichten von CAN-1 nach CAN-4

Aktion: Öffnen Sie auf ihrem Desktop durch einen Doppelklick den PPCAN-Editor.

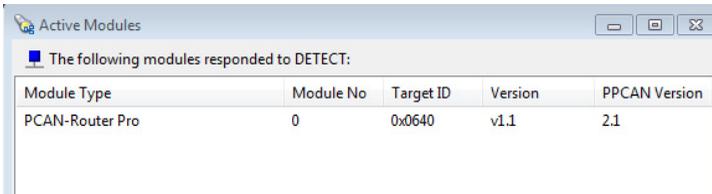
Aktion: Verbinden Sie den PPCAN-Editor mit einem PEAK-Interface z.B. PCAN-USB. Wählen Sie den Menüpunkt **CAN -> Connect** und wählen dann die entsprechende Hardware aus.

Reaktion: Das gewählte CAN-Netz erscheint in der Statuszeile des PPCAN-Editors (unten links).



Aktion: Prüfen, ob der PCAN-Router Pro am CAN-Netzwerk gefunden werden kann: Wählen Sie den Menüpunkt **Transmit -> Detect Modules** aus.

Reaktion: Anzeige des PCAN-Router Pro mit einigen Statusinformationen.



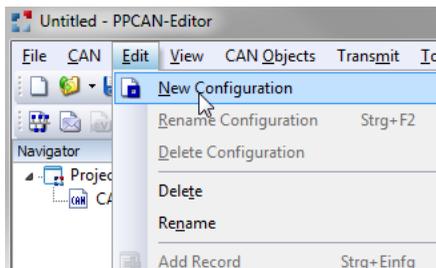
Module Type	Module No	Target ID	Version	PPCAN Version
PCAN-Router Pro	0	0x0640	v1.1	2.1

z. B. wird im Feld Module No die eingestellte Geräte-ID angezeigt (hier: 0). Im Feld Version wird die Firmware-Version angegeben.

Aktion: Anlegen einer leeren Konfigurationsdatei: den Menüpunkt **File -> New** auswählen.

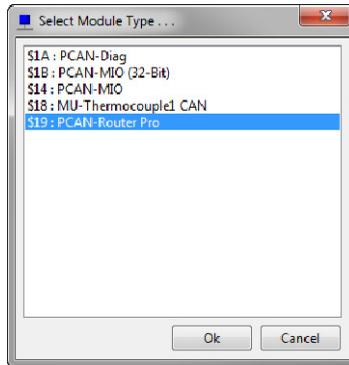
Reaktion: Es erscheint ein leeres Fenster, worin globale CAN-Objekte definiert werden können. Die Handhabung dieses Fensters (und der Einträge darin) wird Ihnen bei der Aufgabe 3a erklärt.

Aktion: Anlegen einer neuen Konfiguration innerhalb der Konfigurationsdatei: den Menüpunkt **Edit -> New Configuration** auswählen:



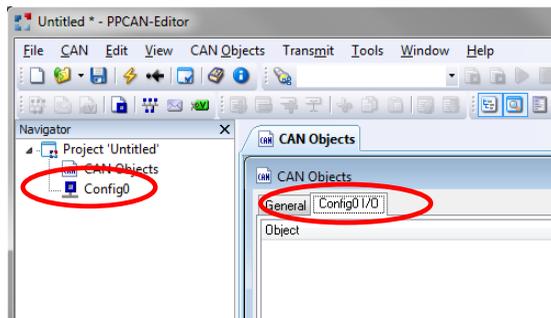
Reaktion: Es wird nach der zu konfigurierenden Hardware gefragt.

Information: Der PPCAN-Editor ist für eine Vielzahl von konfigurierbaren PCAN-Geräten mit unterschiedlichsten Ressourcen ausgelegt. Für jede konfigurierbare PCAN-Hardware gibt es daher eine Liste der verfügbaren Ressourcen, die sogenannte **Profildatei** (*.ppprf).

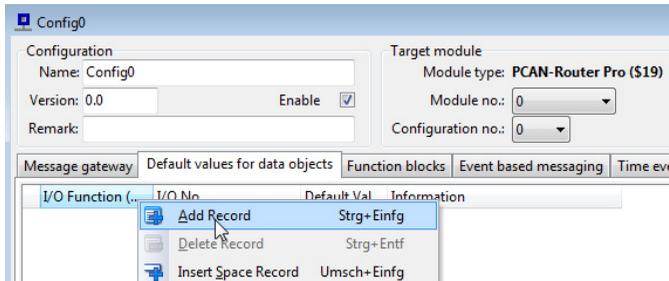


Aktion: Wählen Sie das Profil für den PCAN-Router Pro aus.

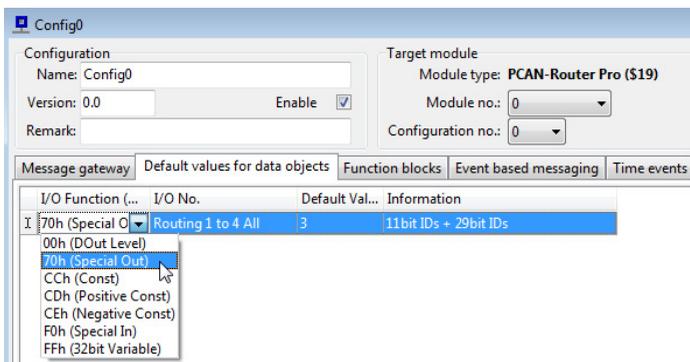
Reaktion: Neben dem Kartei-Tab General erscheint ein neuer Tab mit dem Namen der Konfiguration Config0 I/O. Im Navigationsfenster am linken Bildrand ist ein Icon Config0 entstanden.



Aktion: Durch Doppelklicken auf dieses Icon gelangt man in das Konfigurationsfenster, dort können Default-Werte für Datenobjekte gesetzt werden, z. B. Routing-Pfade. Klicken Sie auf den Tab Default values for data objects, hier muss ein neuer Eintrag erzeugt werden. So legen Sie den Eintrag an: den Menüpunkt **Edit -> Add record** auswählen oder mit der rechten Maustaste auf das Kontextmenü klicken und **Add record** auswählen:



Aktion: Um Werte in die entsprechenden Tabellenzellen einzutragen, aktiviert man den Eingabefokus der markierten Zelle mit der Taste **F2**, durch langsames Klicken oder durch Schreiben des Wertes. Die Einträge in der neuen Zeile sind wie folgt zu setzen:



1. I/O-Funktion: SpecialOut (eine Gruppe von Gerätefunktionen).
2. I/O-No: Routing 1 to 4 All.
3. Default value: **3** = Summe aus **1** (routet nur 11-Bit-IDs) und **2** (routet nur 29-Bit-IDs).
4. Information: Eine Beschreibung, was diese Zeile macht.

In diesem Fall wird das Routing von eingehenden CAN-Nachrichten am CAN-1 (Quelle) zum CAN-4 (Ziel) aktiviert. Der Wert 3 bedeutet dabei, dass sowohl 11-Bit-IDs (1) als auch 29-Bit-IDs (2) geroutet

werden. Die Routing-Funktion befindet sich in der Gruppe 0x70 SpecialOut, die folgende Weiterleitungen für CAN-Nachrichten bietet:

Quelle CAN 1	Quelle CAN 2	Quelle CAN 3	Quelle CAN 4
Routing 1 to 2 All	Routing 2 to 1 All	Routing 3 to 1 All	Routing 4 to 1 All
Routing 1 to 3 All	Routing 2 to 3 All	Routing 3 to 2 All	Routing 4 to 2 All
Routing 1 to 4 All	Routing 2 to 4 All	Routing 3 to 4 All	Routing 4 to 3 All
Routing 1 to CF All	Routing 2 to CF All	Routing 3 to CF All	Routing 4 to CF All
Routing 1 to 2 explicit	Routing 2 to 1 explicit	Routing 3 to 1 explicit	Routing 4 to 1 explicit
Routing 1 to 3 explicit	Routing 2 to 3 explicit	Routing 3 to 2 explicit	Routing 4 to 2 explicit
Routing 1 to 4 explicit	Routing 2 to 4 explicit	Routing 3 to 4 explicit	Routing 4 to 3 explicit
Routing 1 to CF explicit	Routing 2 to CF explicit	Routing 3 to CF explicit	Routing 4 to CF explicit
Routing 1 to 2 excluding	Routing 2 to 1 excluding	Routing 3 to 1 excluding	Routing 4 to 1 excluding
Routing 1 to 3 excluding	Routing 2 to 3 excluding	Routing 3 to 2 excluding	Routing 4 to 2 excluding
Routing 1 to 4 excluding	Routing 2 to 4 excluding	Routing 3 to 4 excluding	Routing 4 to 3 excluding
Routing 1 to CF excluding	Routing 2 to CF excluding	Routing 3 to CF excluding	Routing 4 to CF excluding

Dabei bedeutet explicit, dass nur definierte Nachrichten geroutet werden und excluding bedeutet, dass definierte Nachrichten vom Routen ausgespart werden. Die Handhabung der Varianten explicit und excluding wird in den beiden nächsten Beispielen (1b und 1c) gezeigt.

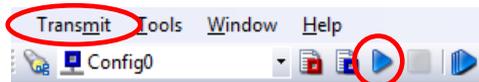
Bemerkung 1: Die Routingfunktionen explicit und excluding unterstützen nur 11-Bit-IDs.

Bemerkung 2: Die CF-Karte kann nicht als Datenquelle (z. B. für Playbacks) genutzt werden.

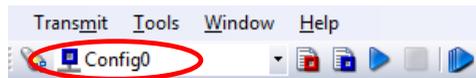
Information: Damit sind die Konfigurationsarbeiten zur Lösung dieser Aufgabe abgeschlossen.

Aktion: Tragen Sie in das Feld **Remark** den Titel der Konfiguration ein. Die Konfigurationsdatei wird als Projekt Aufgabe 1a auf dem PC gespeichert. Klicken Sie hierzu auf den Menüpunkt **File -> Save As**.

Aktion: Die Konfigurationsdatei wird über den CAN-Bus an den PCAN-Router Pro übermittelt (Upload). Dazu muss der Menüpunkt **Transmit -> Send Configuration** angewählt werden (oder das entsprechende Icon):



Wichtiger Hinweis: am oberen Fensterrand in der Toolbar muss der Eintrag Config0 in der Listbox eingestellt sein.



Reaktion: Während des Upload laufen im Output-Window des PPCAN-Editors verschiedene Statusmeldungen durch, die das Übertragungsprotokoll betreffen, deren Bedeutung in anderen Dokumenten erläutert wird.

Reaktion: Die Status-LED des PCAN-Router Pro blinkt grün während der Übertragung und Verarbeitung der Konfigurationsdatei unrythmisch. Danach blinkt die Status-LED grün mit 1 Hz und der PCAN-Router Pro ist mit seiner neuen Konfiguration betriebsbereit.

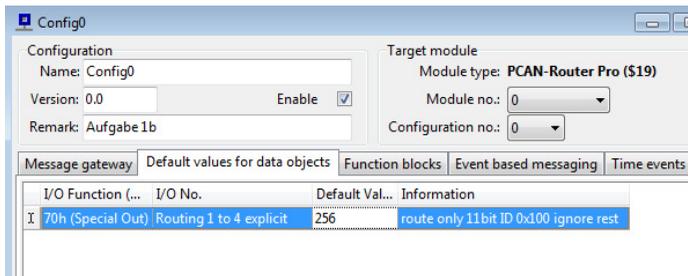
Ergebnis: Der PCAN-Router Pro überträgt nun alle an CAN-1 eingehenden Nachrichten unmodifiziert zum CAN-4 (jedoch nicht in der entgegengesetzten Richtung, dies müsste in einer weiteren Zeile festgelegt werden).

4.2 Aufgabe 1b: Durchreichen von definierten Nachrichten von CAN-1 nach CAN-4

Information: Es soll nur die ID 0x100 vom CAN-1 zum CAN-4 geroutet werden.

Aktion: Öffnen Sie die Konfiguration aus Aufgabe 1a und speichern diese unter Aufgabe 1b. Am linken Bildrand im Navigations-Fenster das Icon Config0 doppelt anklicken. Im Konfigurationsfenster zum Tab Default values for data objects wechseln.

Aktion: Dort den Eintrag aus Aufgabe 1a modifizieren:



1. I/O-Funktion: SpecialOut (bleibt unverändert).
2. I/O-No: Routing 1 to 4 explicit.
3. Default Value: hier die zu routende 11-Bit-ID angeben (0..2047dez).
4. Information: Eine Beschreibung, was diese Zeile macht.

Bemerkung: Die Routingfunktion explicit unterstützt nur 11-Bit-IDs!

Information: Damit sind die Konfigurationsarbeiten zur Lösung dieser Aufgabe abgeschlossen.

Aktion: Die Konfigurationsdatei wird als Projekt Aufgabe 1b auf dem PC gespeichert.

Aktion: Die Konfigurationsdatei wird über den CAN-Bus an den PCAN-Router Pro übermittelt (Upload).

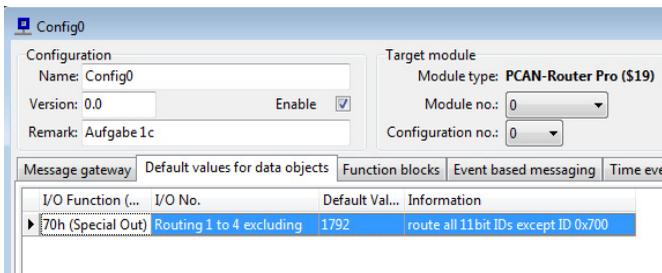
Ergebnis: Auf CAN-1 eintreffende Nachrichten werden ignoriert, lediglich ID 0x100 wird zum CAN-4 geroutet.

4.3 Aufgabe 1c: Durchreichen aller Nachrichten von CAN-1 nach CAN-4 mit Ausnahmen

Information: Es soll alles außer der ID 0x700 vom CAN-1 zum CAN-4 geroutet werden.

Aktion: Öffnen der Konfiguration aus Aufgabe 1a und speichern unter Aufgabe 1c. Am linken Bildrand im Navigations-Fenster das Icon Config0 doppelt anklicken. Im Konfigurationsfenster zum Tab Default values for data objects wechseln.

Aktion: Dort den Eintrag aus Aufgabe 1a modifizieren:



1. I/O-Function: SpecialOut (bleibt unverändert).
2. I/O-No: Routing 1 to 4 excluding.
3. Default Value: hier die herauszufilternde 11-Bit-ID angeben (0..2047dez).
4. Information: Eine Beschreibung, was diese Zeile tut (für später...).

Bemerkung: Die Routingfunktion excluding unterstützt nur 11-Bit-IDs!

Information: Damit sind die Konfigurationsarbeiten zur Lösung dieser Aufgabe abgeschlossen.

Aktion: Die Konfigurationsdatei wird als Projekt Aufgabe 1c auf dem PC gespeichert.

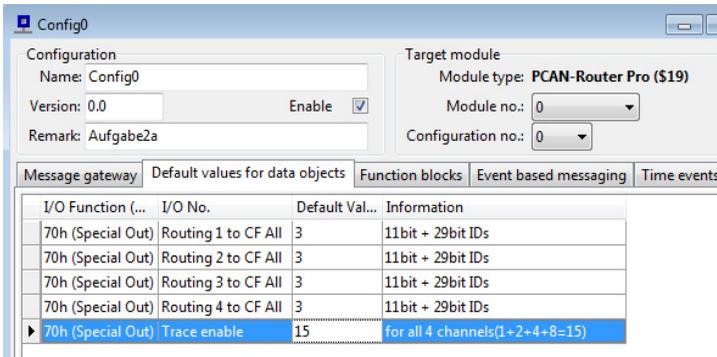
Aktion: Die Konfigurationsdatei wird über den CAN-Bus an den PCAN-Router Pro übermittelt (Upload).

Ergebnis: Alle an CAN-1 eintreffenden Nachrichten werden unmodifiziert zum CAN-4 geroutet, lediglich ID 0x700 wird verworfen.

4.4 Aufgabe 2a: Aufzeichnen aller Empfangsnachrichten auf CF-Karte

Aktion: Öffnen Sie die Konfiguration aus Aufgabe 1a und speichern diese unter Aufgabe 2a. Am linken Bildrand im Navigations-Fenster das Icon Config0 doppelt anklicken. Im Konfigurationsfenster zum Tab Default values for data objects wechseln.

Aktion: Dort den Eintrag aus Aufgabe 1a modifizieren und 4 weitere Zeilen hinzufügen:



1. I/O-Funktion: SpecialOut
2. Zeilen 1..4, I/O-No: Routing 1/2/3/4 to CF all
3. Default Value: **3** = Summe aus **1** (routet nur 11-Bit-IDs) und **2** (routet nur 29-Bit-IDs)
4. Zeile 5, I/O-No: Trace enable (1 Bit für jeden CAN-Kanal)
5. Default Value: **15** = Summe aus **1** (CAN-1), **2** (CAN-2), **4** (CAN-3) und **8** (CAN-4)
6. Information: Eine Beschreibung der Zeile (für später...)

Information: Damit sind die Konfigurationsarbeiten zur Lösung dieser Aufgabe abgeschlossen.

Aktion: Die Konfigurationsdatei wird als Projekt Aufgabe 2a auf dem PC gespeichert.

Aktion: Die Konfigurationsdatei wird über den CAN-Bus an den PCAN-Router Pro übermittelt (Upload).

Ergebnis: Alle an den vier CAN-Kanälen eintreffenden Nachrichten werden unmodifiziert auf die CF-Karte geloggt. Eine Auswertung kann später am PC erfolgen (siehe Aufgabe 2b).

Bemerkung 1: Es können auch die bereits bekannten Routingfunktionen explicit und excluding benutzt werden (unterstützen nur 11-Bit-IDs).

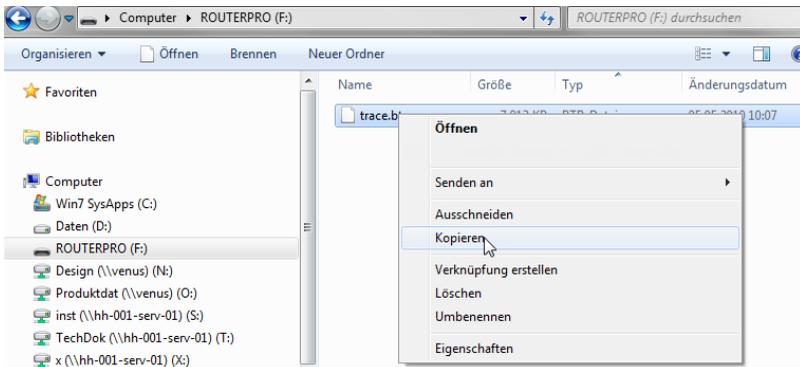
Bemerkung 2: Die CF-Karte ist auch als CAN-5 (virtueller CAN-Kanal) ansprechbar.

Bemerkung 3: Die Rückrichtung (routen von CF-Karte auf die CAN-Kanäle) ist nicht möglich.

4.5 Aufgabe 2b: Extrahieren von Trace-Dateien von der CF-Karte auf den PC

Information: Mit dem PEAK-Converter (im Lieferumfang des PCAN-Router Pro enthalten) können Traces von der CF-Karte extrahiert und in leserlichen Klartext umgewandelt werden. In diesem Format kann die Aufzeichnung von allen PEAK-Applikationen (PCAN-Explorer, PCAN-Trace, etc.) weiterverarbeitet werden.

Aktion: Zum Bearbeiten muss zunächst die CF-Karte aus dem PCAN-Router Pro entnommen werden (nur im ausgeschalteten Zustand oder im Sleep-Mode des PCAN-Router Pro). Die Karte wird dann in einen Kartenlese-Schacht des PC gesteckt, und die auf der Karte befindliche Datei TRACE.BTR wird auf die lokale Festplatte kopiert.



Aktion: Anschließend wird diese Datei (und die darin aufgezeichneten Traces) mit dem PEAK-Converter bearbeitet.

Aktion: Bevor die Karte in den PCAN-Router Pro zurückgesteckt wird, sollte sie geleert werden: Einfach mit dem Windows Explorer formatieren (FAT16 unter Windows 7 (standard)) und eine leere TRACE.BTR-Datei darauf legen. Entsprechende Dateien sind in unterschiedlichen Größen auf der beiliegenden Produkt-CD des PCAN-Router Pro abgelegt.

4.6 Aufgabe 3a: Definieren von CAN-Nachrichten (z. B. Ausgabe einer Systemvariablen)

Aktion: Anlegen einer leeren Konfigurationsdatei: den Menüpunkt **File -> New** auswählen.

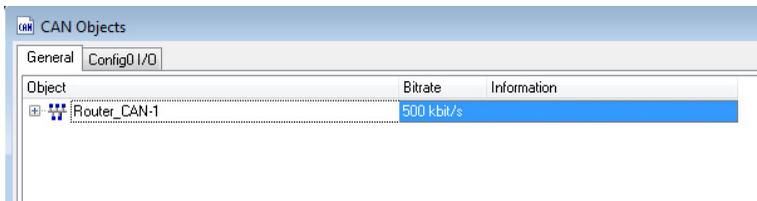
Reaktion: Es erscheint ein leeres Fenster mit den globalen CAN-Objekten für alle später in der Datei enthaltenen Konfigurationen. Falls eine Datei mehrere Konfigurationen mit unterschiedlichen CAN-Objekten enthält, müssen -alle- zunächst hier definiert werden. Sie werden später selektiv in die verschiedenen Konfigurationen importiert.

Reaktion: Im Fenster ist bereits ein CAN-Bus definiert: Bus_0, unter dem man globale CAN-Objekte hierarchisch anlegen kann. Eine Konfiguration ist damit aber noch nicht angelegt.

Beachten Sie: Dies ist nur ein Beispiel für eine mögliche Konfiguration.

Definition: Der Bus_0 erhält den Namen Router_CAN-1, das Feld Bitrate ist nur informativ. Die standardmäßig im PCAN-Router Pro verbauten Transceiver TJA1041 werden ohne weiteres Zutun auf 500 kbit/s eingestellt.

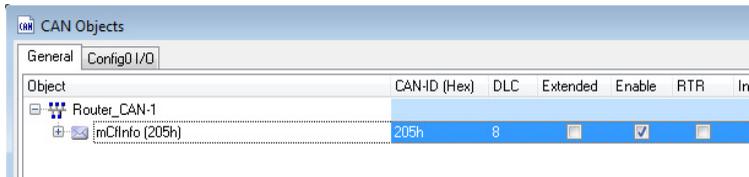
Aktion: Doppelklick auf den Namen Bus_0 und Eintragen des neuen Namens. Auswählen der Default-Bitrate 500 kbit/s aus der Dropdownliste (nur informativ!).



1. Busname: Router_CAN-1.
2. Baudrate: 500 kbit/s (nur informativ!).
3. Information: Eine Beschreibung, was diese Zeile macht.

Definition: Es soll auf diesem Bus eine CAN-Nachricht mCfInfo mit 8 Bytes Länge ausgesendet werden, CAN-ID = 0x205.

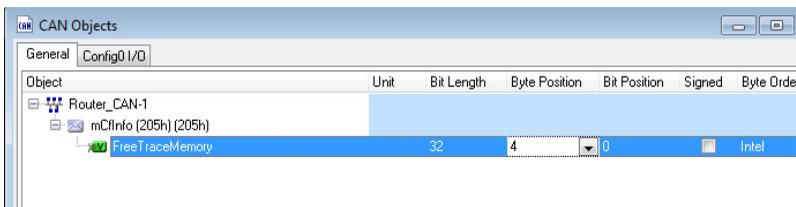
Aktion: Dazu wird im Kontextmenü (Rechtsklick auf Router_CAN-1) der Punkt **Add a new Symbol** angewählt. Damit wird eine neue CAN-Nachricht auf dem Bus Router_CAN-1 definiert, deren Parameter jetzt noch ausgefüllt werden müssen:



1. Symbolname: mCfInfo.
2. ID: 0x205.
3. DLC: 8 (Nachricht hat 8 Datenbytes).
4. Extended: Nein, es genügt eine 11-Bit-ID.
5. Enabled: Ja.
6. RTR. Nein, die Nachricht soll -nicht- nur auf Anforderung gesendet werden.
7. Information. Eine Beschreibung, was diese Zeile macht.

Definition: Diese Nachricht soll ein 32-Bit-Signal FreeTraceMemory enthalten (in den Datenbytes 5..8), das die Anzahl der noch speicherbaren CAN-Nachrichten auf der CF-Karte anzeigt.

Aktion: Innerhalb der CAN-Nachricht muss ein 32-Bit breites Datenobjekt (=CAN-Signal) angelegt werden, das die Anzahl der noch freien Plätze zeigt. Dazu wird im Kontextmenü (Rechtsklick auf die CAN-Nachricht) der Punkt **Add a new Variable** angewählt, und dann die Parameter des Signals eingetragen:



1. Variablenname: FreeTraceMemory.
2. Unit: Keine physikalische Einheit (nur informativ).

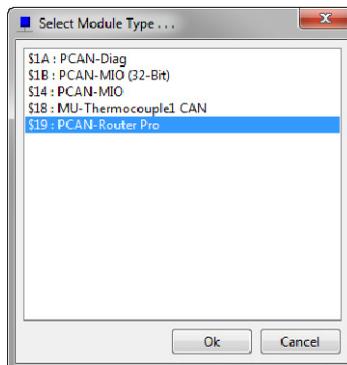
3. Bit length: 32.
4. StartByte: 4 (Signal liegt dadurch in den Datenbytes 5..8).
5. StartBit: 0.
6. Signed: Nein, nicht vorzeichenbehaftet.
7. Byte Order: Intel-Format (LSB in Byte 0 Bit 0, MSB in Byte 3 Bit 7).
8. Information: Eine Beschreibung, was diese Zeile macht.

Information: Die leere Hülle der CAN-Nachricht ist damit festgelegt, aber noch nicht einer physikalischen Datenquelle zugeordnet. Dazu muss eine Konfiguration erzeugt werden.

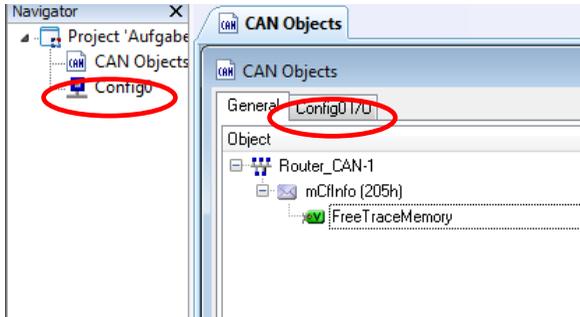
Aktion: Anlegen einer leeren Konfiguration innerhalb der Datei: den Menüpunkt **Edit -> New Configuration** auswählen.

Reaktion: Es wird nach der zu konfigurierenden Hardware gefragt.

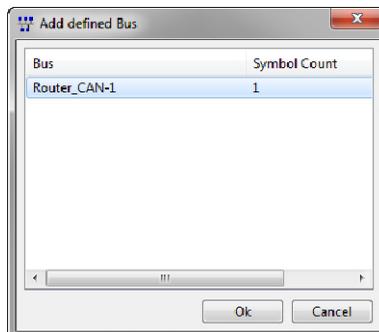
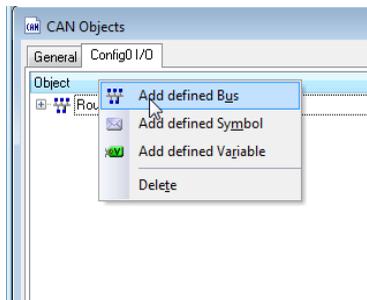
Aktion: Wählen Sie das Profil für den PCAN-Router Pro aus.



Reaktion: Es erscheint neben dem Tab General ein neuer Tab mit dem Namen der Konfiguration Config0 I/O. Außerdem ist im Navigationsfenster am linken Bildrand ein Icon Config0 entstanden.

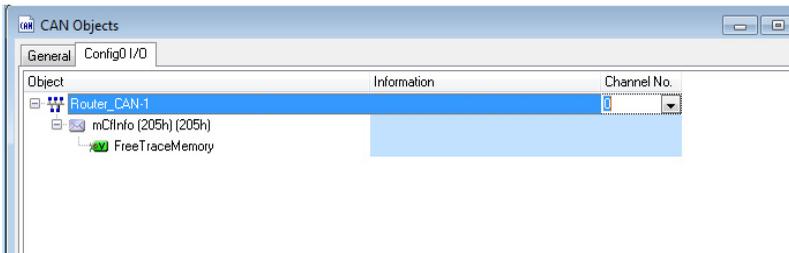


Aktion: Die global definierte CAN-Nachricht soll in dieser Konfiguration verwendet werden. Sie muss also importiert werden. Klicken Sie auf den neuen Tab Config0 I/O, dieser wird in den Vordergrund geholt und im Kontextmenü (Rechtsklick) wird **Add defined Bus** ausgewählt:



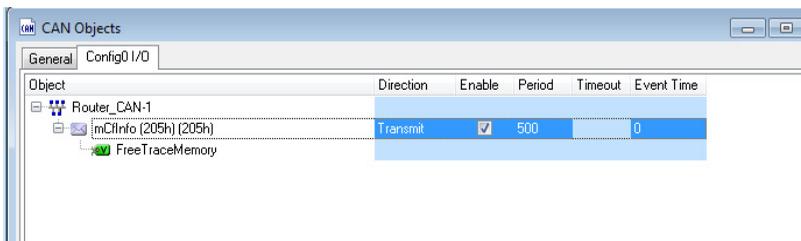
Reaktion: Der global bereits definierte CAN-Bus Router_CAN-1 wird mitsamt seiner enthaltenen Nachricht mCfInfo und der 32-Bit-Variablen FreeTraceMemory in die Konfiguration Config0 übernommen.

Aktion: Dem definierten Bus Router_CAN-1 muss eine CAN-Kanalnummer der Hardware zugewiesen werden (hier: CAN-Kanal #0):



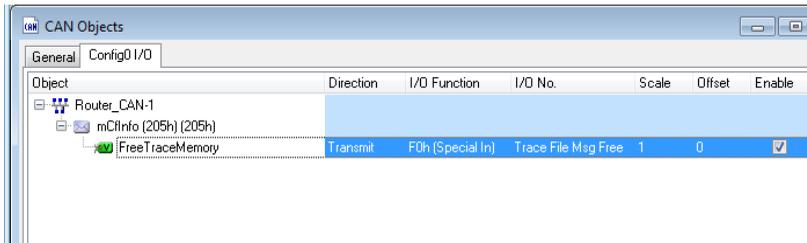
1. Channel-Number: 0 (Die Hardware-CAN-Kanäle sind intern von 0 bis 3 nummeriert).

Aktion: Jetzt müssen der Nachricht physikalische Parameter zugewiesen werden:



1. Direction: Transmit (Der PCAN-Router Pro soll Sender sein).
2. Enable: Ja, diese Nachricht soll übermittelt werden.
3. Period: 500 (Die Sendezykluszeit in ms).

Aktion: Dem enthaltenen CAN-Signal muss eine Datenquelle zugewiesen werden:



1. I/O-Funktion: F0-Special In (das ist die Datenquelle: Eine interne Firmware-Variablen).
2. I/O-Number: Trace File Msg Free (das ist der Name der Variablen).
3. Scale: 1 (keine Verstärkung/Abschwächung der Daten, Multiplikation *1).
4. Offset: 0 (keine Anhebung/Absenkung des Wertebereichs, Addition 0).
5. Enable: Ja, dieses Signal (innerhalb der Nachricht) soll verwendet werden.

Information: Damit sind die Konfigurationsarbeiten zur Lösung dieser Aufgabe abgeschlossen.

Aktion: Die Konfigurationsdatei wird als Projekt Aufgabe 3a auf dem PC gespeichert.

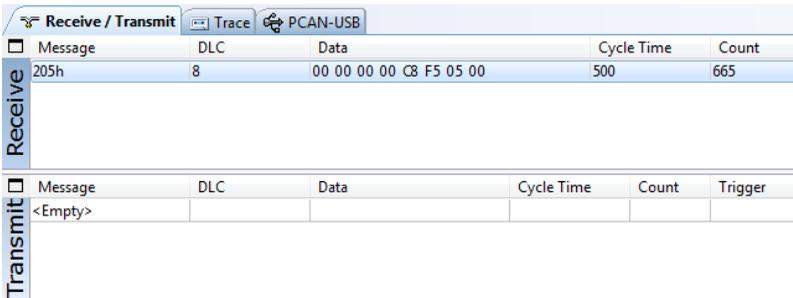
Aktion: Die Konfigurationsdatei wird über den CAN-Bus an den PCAN-Router Pro übermittelt (Upload), siehe Aufgabe 1a.

Reaktion: Im Output-Window des PPCAN-Editors laufen nun verschiedene Statusmeldungen durch, die das Übertragungsprotokoll betreffen, deren Bedeutung in anderen Dokumenten erläutert wird.

Reaktion: Die Status-LED des PCAN-Router Pro blinkt grün während der Übertragung und Verarbeitung der Konfigurationsdatei unrythmisch. Danach blinkt die Status-LED grün mit 1 Hz und der PCAN-Router Pro ist mit seiner neuen Konfiguration betriebsbereit.

Ergebnis: Der PCAN-Router Pro sendet nun am CAN-1 die Nachricht mit der ID0x205, Länge 8 mit dem 32-Bit-Wert TraceFileMsgFree (eine interne Systemvariable des Routers).

Ein am gleichen CAN-Netz angeschlossener PCAN-View stellt die Sendenachricht dar.



Receive / Transmit						
Message	DLC	Data	Cycle Time	Count		
205h	8	00 00 00 00 C8 F5 05 00	500	665		

Message	DLC	Data	Cycle Time	Count	Trigger
<Empty>					

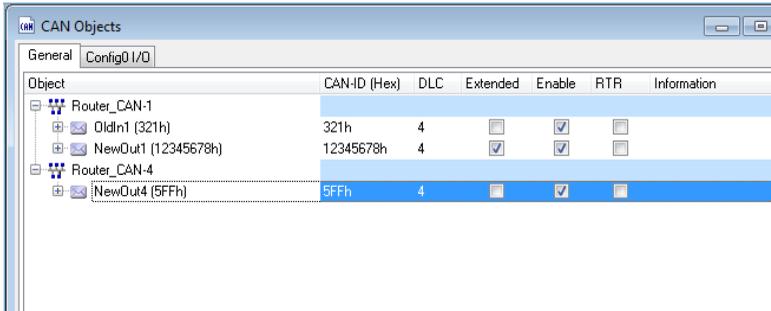
Im hier gezeigten Fall sind im Trace-Speicher auf der CompactFlash-Karte noch 390600 Nachrichten-Plätze frei (=0x05F5C8).

4.7 Aufgabe 3b: Umsetzen einer CAN-ID

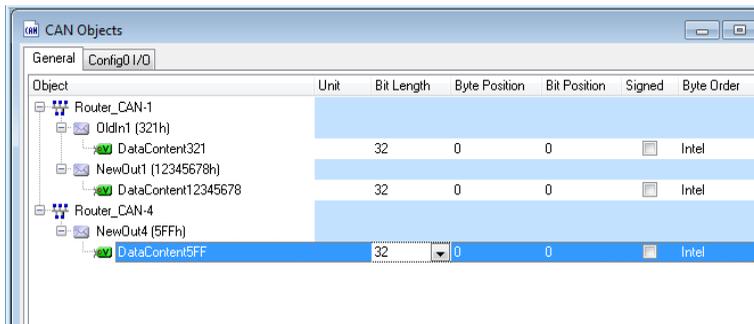
Information: Zum Umsetzen einer CAN-Nachricht auf eine andere ID müssen zunächst beide Nachrichten global definiert und einer neuen Konfiguration zugewiesen werden. Das prinzipielle Vorgehen hierbei ist bereits aus der vorherigen Aufgabe bekannt.

Definition: Der Inhalt der Empfangs-Nachricht 0x321 soll unverändert in die Sende-Nachricht 0x12345678 übernommen werden. Diese Sende-Nachricht soll außerdem mit der ID 0x5FF auf CAN-4 gesendet werden.

Aktion: Globales Definieren von zwei CAN-Bussen Router_CAN-1 und Router_CAN-4, sowie insgesamt 3 Nachrichten (**Add a new Symbol**):



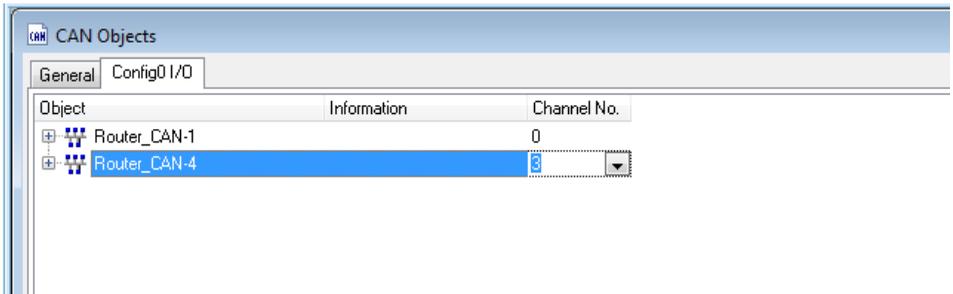
Aktion: Globales Definieren von jeweils einem 32-Bit-Signal (**Add a new Variable**) in den 3 Nachrichten:



Aktion: Anlegen einer neuen Konfiguration mit **Edit -> New Configuration** (dabei das Hardware-Profil PCAN-Router Pro auswählen).

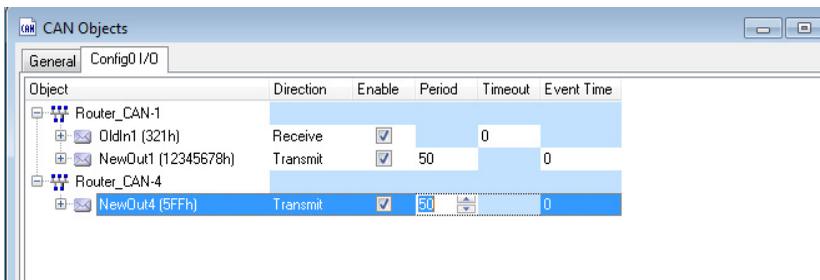
Aktion: Importieren der global definierten CAN-Objekte im Tab Config0 I/O. Dazu mit einem Rechtsklick das Kontextmenü öffnen und **Add defined Bus** auswählen. Im Auswahlfenster nacheinander die Busse Router_CAN-1 und Router_CAN-4 auswählen.

Aktion: Jetzt die Parameter der beiden Busse eingeben:



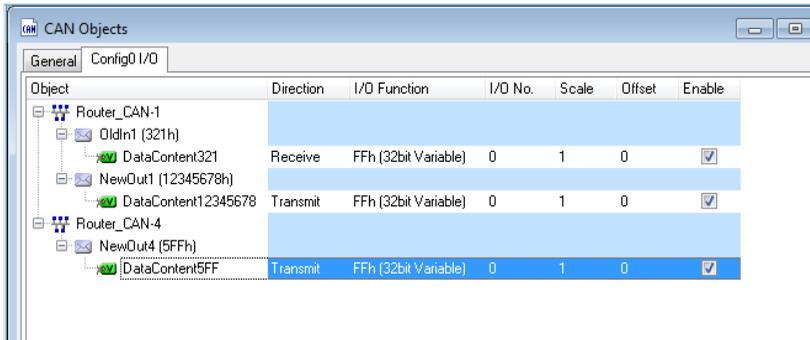
1. Channel-Number: **0** (für Router_CAN-1) und **3** (für Router_CAN-4).

Aktion: Jetzt die Parameter der drei Nachrichten eingeben:



1. Direction: Eine Empfangs-Nachricht 0x321 (Receive), zwei Sende-Nachrichten 0x12345678 und 0x5FF (Transmit).
2. Enable: Ja, diese Nachrichten sollen empfangen/übermittelt werden.
3. Period: 50 (Die Sendezykluszeit in ms).

Aktion: Jetzt die Parameter der drei Signale eingeben:



1. I/O-Funktion: FF-32-Bit-Variablen (hier wird der Inhalt der Nachricht zwischengeparkt).
2. I/O-Nummer: 0 (Es gibt 256 dieser RAM-Variablen zur freien Verfügung).
3. Scale: 1 (keine Verstärkung/Abschwächung der Daten, Multiplikation * 1).
4. Offset: 0 (keine Anhebung/Absenkung des Wertebereichs, Addition 0).
5. Enable: Ja, dieses Signal (innerhalb der Nachricht) soll verwendet werden.

Information: Damit sind die Konfigurationsarbeiten zur Lösung dieser Aufgabe bereits abgeschlossen.

Aktion: Die Konfigurationsdatei wird als Projekt Aufgabe 3b auf dem PC gespeichert.

Aktion: Die Konfigurationsdatei wird über den CAN-Bus an den PCAN-Router Pro übermittelt (Upload).

Aktion: Es wird vom oder dem PCAN-View eine 4 Byte lange CAN-Nachricht mit der ID 0x321 gesendet (auf CAN-1), in der die 4 Datenbytes mit Werten versorgt sind.

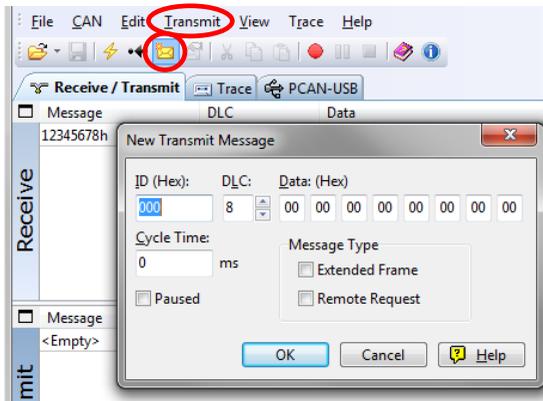
Ergebnis: Auf CAN-1 und CAN-4 erscheinen die Nachrichten 0x12345678 und 0x5FF zyklisch mit dem vorgegebenen Dateninhalt.

Receive / Transmit		Trace	PCAN-USB			
Receive	<input type="checkbox"/> Message	DLC	Data	Cycle Time	Count	
	12345678h	4	F5 A5 55 05	50	1201	
Transmit	<input type="checkbox"/> Message	DLC	Data	Cycle Time	Count	Trigger
	321h	4	F5 A5 55 05	<input checked="" type="checkbox"/> 250	94	Time

Receive / Transmit		Trace	PCAN-USB Hub			
Receive	<input type="checkbox"/> Message	DLC	Data	Cycle Time	Count	
	5FFh	4	F5 A5 55 05	50	2494	
Transmit	<input type="checkbox"/> Message	DLC	Data	Cycle Time	Count	Trigger
	<Empty>					

Information: Die CAN-Nachrichten zum Senden tragen Sie im PCAN-View oder PCAN-Explorer über den Menüpunkt **Transmit ->**

New Message oder über  in der Werkzeugleiste ein:

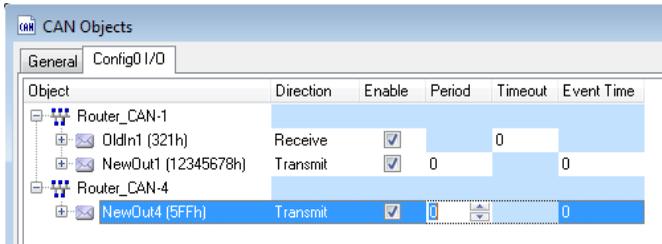


Der in der Empfangs-Nachricht 0x321 enthaltene 32-Bit-Wert (0x0555A5F5) wird im Router der RAM-Variablen #0 zugewiesen. Beim Senden der Nachrichten 0x12345678 und 0x5FF wird der Inhalt dieser Variablen #0 zurück gelesen und in die Sendenachrichten eingefügt. Der Wert kann auch an einer anderen Position innerhalb der Sendenachricht eingefügt werden, oder vorher per Berechnung modifiziert werden.

4.8 Aufgabe 3c: (Variation 3b) Nur dann Senden, wenn Quellnachricht empfangen wurde

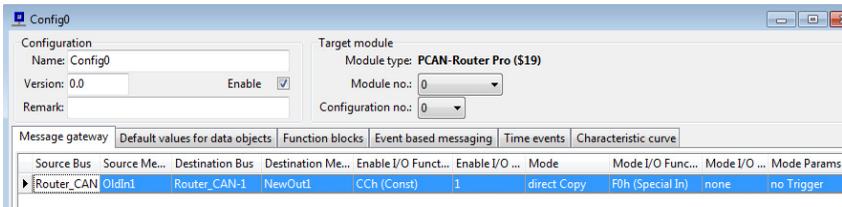
Information: Es ist auch möglich, die Sendenachrichten nur OnEvent zu senden, z. B. nur wenn die Nachricht 0x321 tatsächlich empfangen wurde, oder wenn sich der Dateninhalt geändert hat. Als Variation der vorherigen Aufgabe werden die beiden Nachrichten 0x5FF und 0x12345678 nur noch dann gesendet, wenn eine Quellnachricht 0x321 empfangen wurde.

Aktion: Im Fenster CAN-Objects, im Tab Config0 I/O ist zunächst das zyklische Senden der Nachrichten 0x5FF und 0x12345678 abzuschalten:



1. Period: 0 (nicht mehr zyklisch senden).

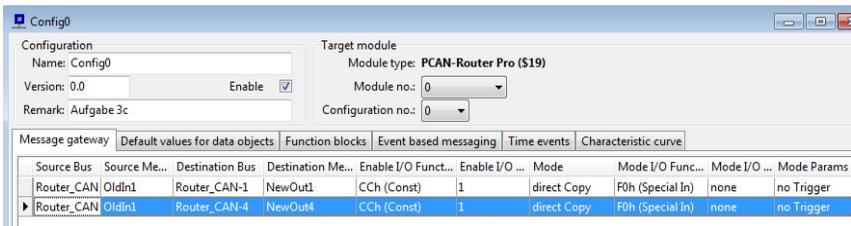
Aktion: Stattdessen müssen im Fenster Config0 (zum öffnen klicken Sie das entsprechende Icon im Fenster Navigation am linken Fensterrand), im Tab Message Gateway zwei neue Einträge erzeugt werden: Dazu das Kontextmenu öffnen (Rechtsklick) und **Add Record** wählen. Die Parameter des neuen Eintrags wie folgt konfigurieren:



1. Source-Bus: Bus#0 (= Router_CAN-1).
2. Source-Message-ID: ID 0x321 (im 11-Bit-Format).
3. Destination-Bus: Bus#0 (= Router_CAN-1).
4. Destination-Message-ID: ID 0x12345678 (im 29-Bit-Format).
5. Enable I/O-Funktion: Dieses Routing soll -immer- aktiv sein, also konstant.
6. Enable I/O-Number: auf 1 setzen.
7. Mode: Direct copy (immer dann, wenn was hereinkommt).

8. Mode I/O Function: Nicht implementiertes Feature, Wert unverändert lassen.
9. Mode I/O Number: Nicht implementiertes Feature, Wert unverändert lassen.
10. Mode Params: Nicht implementiertes Feature, Wert unverändert lassen.
11. Information: Eine Beschreibung, was diese Zeile macht.

Aktion: Das gleiche noch mal für die Nachricht 0x5FF auf dem 4. Bus:



1. Source-Bus: Bus#0 (= Router_CAN-1).
2. Source-Message-ID: ID 0x321 (im 11-Bit-Format).
3. Destination-Bus: Bus#3 (= Router_CAN-4).
4. Destination-Message-ID: ID 0x5FF (im 11-Bit-Format).

Information: Damit sind die Konfigurationsarbeiten zur Lösung dieser Aufgabe abgeschlossen.

Aktion: Die Konfigurationsdatei wird als Projekt Aufgabe 3c auf dem PC gespeichert.

Aktion: Die Konfigurationsdatei wird über den CAN-Bus an den PCAN-Router Pro übermittelt (Upload).

Ergebnis: Wird auf CAN-1 eine Nachricht mit der ID 0x321 zum PCAN-Router Pro gesendet, dann wird der Inhalt der Nachricht auf

CAN-1 in einer Nachricht mit der ID 0x12345678 und auf CAN-4 in einer Nachricht mit der ID 0x5FF ausgegeben:

Receive / Transmit		Trace	PCAN-USB			
Message	DLC	Data	Cycle Time	Count		
12345678h	4	F5 A5 55 05	19791	14		

Message	DLC	Data	Cycle Time	Count	Trigger
321h	4	F5 A5 55 05	<input type="checkbox"/> 250	123	Manual

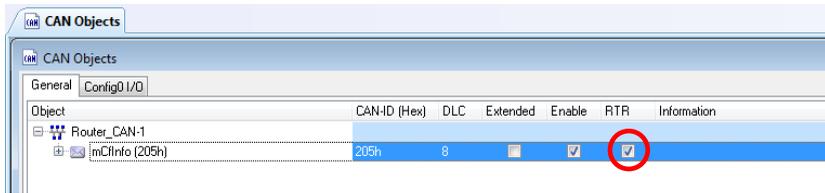
Receive / Transmit		Trace	PCAN-USB Hub			
Message	DLC	Data	Cycle Time	Count		
5FFh	4	F5 A5 55 05	23415	1324		

Message	DLC	Data	Cycle Time	Count	Trigger
<Empty>					

4.9 Aufgabe 3d: (Variation 3a) Nur Senden bei externer Anforderung

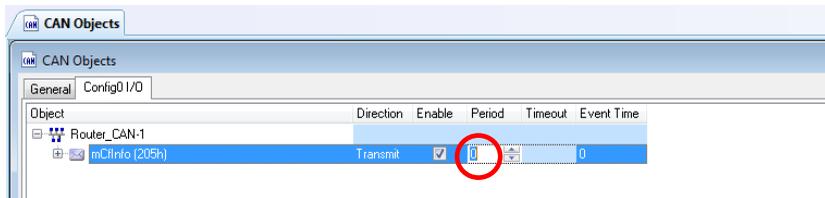
Information: Ausgehend von Aufgabe 3a soll die Information FreeTraceMemory auf Anforderung von extern ausgegeben werden (RTR = Remote Transmission Request).

Aktion: Öffnen Sie die Konfiguration aus Aufgabe 3a und speichern diese unter Aufgabe 3d. Im Tab General umstellen der CAN-Nachricht auf Transmission Request durch Setzen des RTR-Häkchens in der Nachricht (damit nur auf Anforderung gesendet wird):



1. RTR: Remote Transmission Request aktivieren.

Aktion: Im Tab Config0 I/O setzen des Sendezyklus auf 0. Damit wird die Nachricht nicht mehr selbsttätig gesendet:



1. Period: 0 (nicht mehr zyklisch Senden).

Information: Damit sind die Konfigurationsarbeiten zur Lösung dieser Aufgabe bereits abgeschlossen.

Aktion: Die Konfigurationsdatei wird als Projekt Aufgabe 3d auf dem PC gespeichert.

Aktion: Die Konfigurationsdatei wird über den CAN-Bus an den PCAN-Router Pro übermittelt (Upload).

Ergebnis: Die Nachricht mCfInfo mit der ID 0x205 wird nur gesendet, wenn vorher eine Anforderung mit ID 0x205 und Nachrichtenlänge = 0 empfangen wurde.

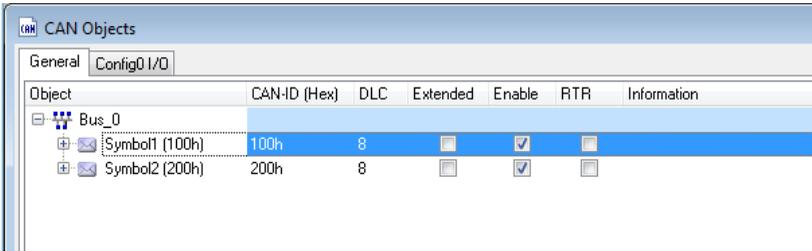
Receive / Transmit		Trace	PCAN-USB			
Receive	<input type="checkbox"/> Message	DLC	Data	Cycle Time	Count	
	205h	8	00 00 00 00 A8 F2 05 00		1	
Transmit	<input type="checkbox"/> Message	DLC	Data	Cycle Time	Count	Trigger
	205h	8	Remote Request	Wait	5	Manual

4.10 Aufgabe 4a: CAN-Signal mittels scale und offset bearbeiten

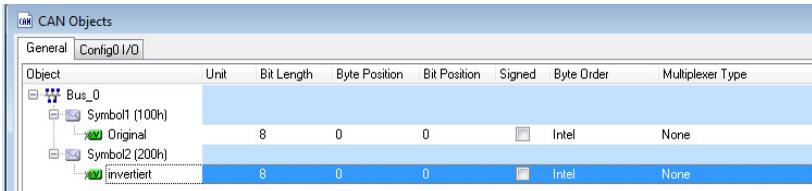
Information: Mit den Parametern SCALE und OFFSET können Werte vom CAN-Bus mit den Grundrechenarten bearbeitet werden, ohne einen Funktionsblock zu definieren. Beispielweise kann ein steigender 8-Bit-Wert (Rampe) invertiert werden (0-> 0xFF, 0xFF -> 0). Dazu wird der Originalwert aus der Empfangs-Nachricht in eine 32-Bit-Variable (es gibt keine kleineren) geschrieben und diese beim Schreiben auf die Sendenachricht mit SCALE -1 und OFFSET 255 beaufschlagt. Es ist genauso möglich, den Empfangswert gleich umzurechnen und erst dann in eine Variable zu schreiben. Man muss aber darauf achten, dass der umgerechnete Wert -immer- in die Variable passt.

Bemerkung: In den PPCAN-Editor - Referenzen werden die Funktionsblöcke, mathematische Funktionen und die I/O-Funktionen näher erläutert.

Aktion: Anlegen einer leeren Konfigurationsdatei mit **File -> New**. Dem bereits existierenden CAN-Bus werden 2 Symbole (= CAN-Nachrichten) hinzugefügt:



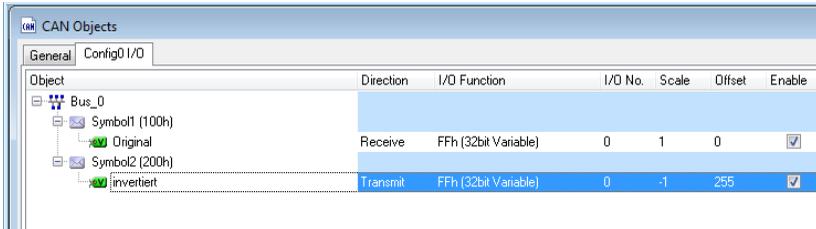
Jede Nachricht erhält noch eine 8-Bit-Variable (= CAN-Signal):



Aktion: Es wird eine neue Konfiguration (Hardware-Profil: PCAN-Router Pro) angelegt und die CAN-Objekte werden importiert. Die Objektparameter werden folgendermaßen versorgt: Die Nachricht mit der ID 0x100 wird empfangen und das darin enthaltene Signal wird in die 32-Bit-Variable #0 geschrieben. Die Nachricht mit der ID 0x200 wird zyklisch gesendet (100 ms). Das darin enthaltene Signal wird der 32-Bit-Variablen #0 entnommen, invertiert (SCALE= -1) und um 255 angehoben (OFFSET=255):



1. Direction: 0x100 wird empfangen (Receive), 0x200 wird gesendet (Transmit).
2. Enable: Ja.
3. Period: 100 (Die Sendezykluszeit in ms).



Information: Damit sind die Konfigurationsarbeiten zur Lösung dieser Aufgabe abgeschlossen.

Aktion: Die Konfigurationsdatei wird als Projekt Aufgabe 4a auf dem PC gespeichert.

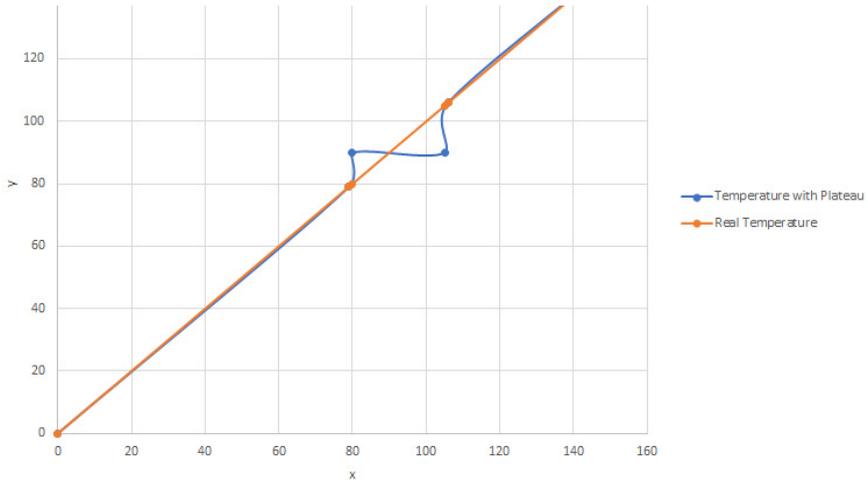
Aktion: Die Konfigurationsdatei wird über den CAN-Bus an den PCAN-Router Pro übermittelt (Upload).

Ergebnis: Die in der Empfangs-Nachricht 0x100 enthaltenen x-Werte (Byte #0) werden als $y=(-1)*x+255$ bzw. $y=255-x$ in der Sendenachricht ausgegeben. Aus einer steigenden Rampe (zunehmende x-Werte) wird eine fallende Rampe (abnehmende y-Werte).

4.11 Aufgabe 4b: CAN-Signal mit Funktionsblock "Kennlinie" bearbeiten

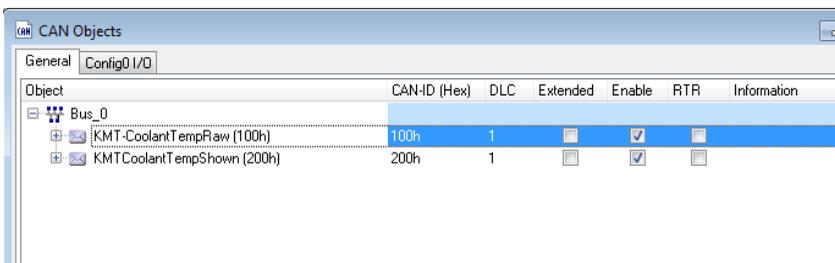
Definition: Die Kühlmitteltemperatur bzw. Motortemperatur soll zwischen 80°C und 105°C als konstant 90°C angezeigt werden (um Schwankungen zu glätten bzw. den Fahrer nicht zu beunruhigen). Sollte die Motortemperatur aufgrund eines Defekts außerhalb des

genannten Bereichs geraten, wird wieder die echte Temperatur angezeigt.



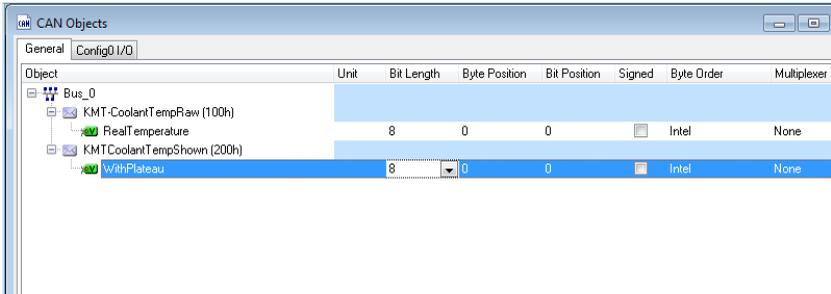
Information: Hierzu eignet sich der Funktionsblock Kennlinie, der den hereinkommenden echten Messwerten vom Motor (= Rohwert) umrechnet und so ein Plateau für die Anzeige bildet. Der umgerechnete Wert wird in einer separaten CAN-Nachricht ausgegeben.

Aktion: Anlegen einer leeren Konfigurationsdatei mit **File -> New**. Definieren zweier CAN-Nachrichten mit je einem 8-Bit-CAN-Signal:



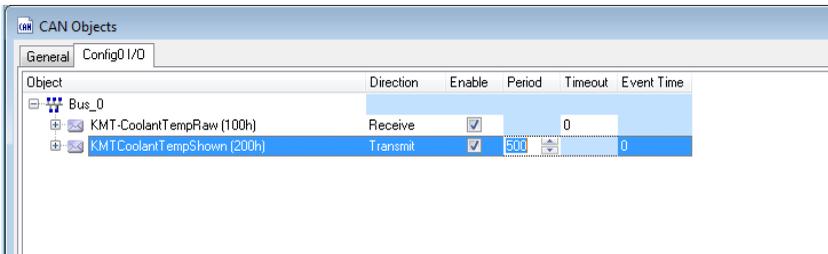
1. CAN-ID: 0x100 für den ankommenden Rohwert, 0x200 für den abgehenden Anzeigewert (mit Plateau).

2. DLC: die CAN-Nachrichten sind je ein Byte lang.
3. Enable: Ja.

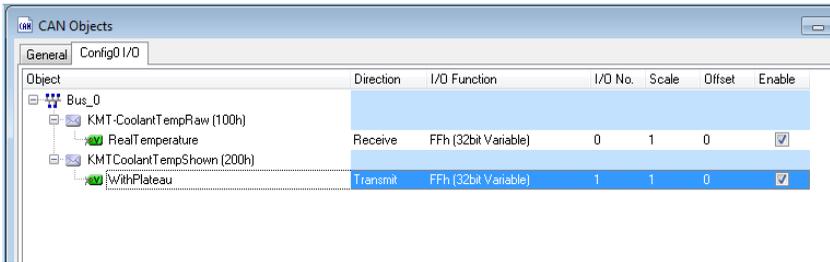


1. Bit Length: 8 (Wertebereich 256 genügt).
2. Signed: nein, kein Vorzeichen.

Aktion: Anlegen einer leeren Konfiguration innerhalb der Datei mit **Edit -> New Configuration** und das Hardware-Profil für den PCAN-Router Pro auswählen. Anschließend den global definierten Bus mit allen Nachrichten und Signalen importieren:



1. Direction: 0x100 wird empfangen (Receive), 0x200 wird gesendet (Transmit).
2. Enable: Ja
3. Period: Die Nachricht mit der manipulierten Kühlmitteltemperatur wird alle 500 ms gesendet.



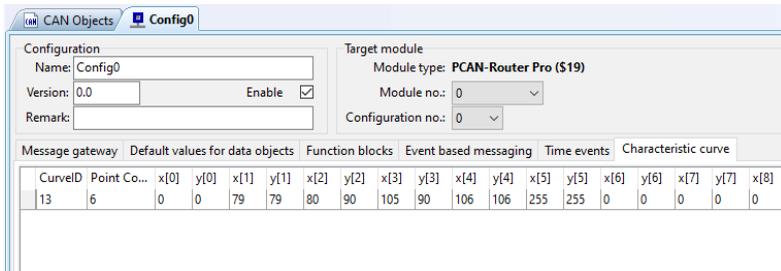
I/O-Funktion, I/O-Number: Das Signal Rohwert aus der Nachricht 0x100 wird in die Variable #0 transferiert, das Ergebnis (in der Variablen #1) wird in die Nachricht 0x200 geschrieben.

Information: Zur Umrechnung des Rohwerts (Variable #0) in den Anzeigewert (Variable #1) wird eine Kennlinie definiert und es bedarf eines Funktionsblocks, der diese Kennlinie aufruft. In der folgenden Tabelle sind die Knickpunkte der Kennlinie abgelegt. Bei Zwischenwerten wird das Ergebnis linear interpoliert:

	X	Y
Punkt 0	X=0	Y=0
Punkt 1	X=79	Y=79
Punkt 2	X=80	Y=90
Punkt 3	X=105	Y=90
Punkt 4	X=106	Y=106
Punkt 5	X=255	Y=255

Aktion: Diese Werte werden nun dem PCAN-Router Pro bekanntgegeben. Führen Sie einen Doppelklick auf **Config0** in der linken **Navigator**-Liste an. In diesem Fenster den Tab **Characteristic curve** auswählen, das Kontextmenü öffnen (Rechtsklick) und **Add Record** auswählen.

Reaktion: Es erscheint eine Tabellenzeile, die eine Kennlinie repräsentiert und mit Werten befüllt werden muss:



1. Curve-ID: 13 (eine willkürlich gewählte Nummer).
2. Point Count: 6 (Anzahl der X/Y-Wertepaare).
3. Wertepaare 0..5: die Kennlinie (Werte aus obiger Tabelle). Weitere Wertepaare werden nicht benötigt und enthalten 0.



Wichtiger Hinweis: Die X-Werte müssen aufsteigend sortiert eingetragen werden!

Information: Zuletzt muss dafür gesorgt werden, dass der Rohwert zu den X-Koordinaten der Kennlinie gelangt, und der sich ergebende Y-Wert in die 32-Bit-Variablen #1 geschrieben wird, um auf den CAN-Bus übermittelt zu werden. Dazu wird ein Funktionsblock benötigt, der die Handhabung der Kennlinie übernimmt.

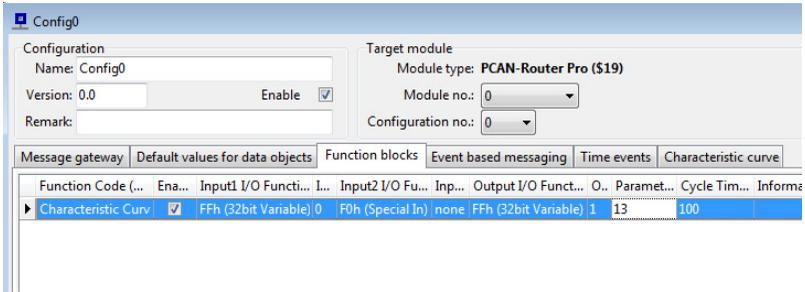
Aktion: Um einen neuen Funktionsblock anzulegen, den Tab Function blocks auswählen, das Kontextmenü öffnen (Rechtsklick) und **Add Record** auswählen.

Reaktion: Es erscheint eine Tabellenzeile, die einen Funktionsblock repräsentiert und mit Werten befüllt werden muss.

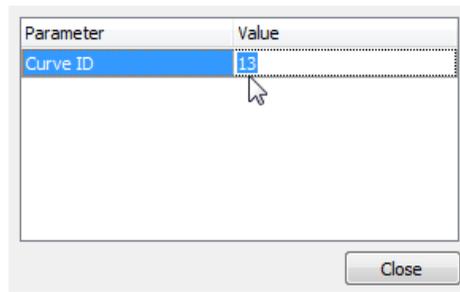
Information: Grundsätzlich hat jeder Funktionsblock zwei Eingänge (Operanden) und einen Ausgang (Ergebnis), jeweils bestehend aus I/O-Typ und I/O-Nummer, Weiterhin gibt es einen Hauptschalter (Enable) und eine Zykluszeit (wie oft wird das Ergebnis berechnet, in ms).

Hinweis: Bei der Characteristic Curve wird der zweite Eingang nicht benutzt.

Aktion: Der Funktionsblock wird gemäß folgendem Bild ausgefüllt:



1. Function Code: Characteristic Curve (die Kennlinien-Bedien-Funktion).
2. Enable: Ja, dieser Block soll aktiv sein.
3. Eingang1: FF-32-Bit-Variable, 0 (der X-Wert ist in der 32-Bit-Variablen #0).
4. Eingang2: F0-Special In, none (= nicht benutzt).
5. Ausgang: FF-32-Bit-Variable, 1 (das Y-Ergebnis kommt in die 32-Bit-Variable #1).
6. Parameter: 13 (die Nummer der bereits definierten Kennlinie). Es öffnet sich folgendes Fenster: Tragen Sie unter Value die Nummer der bereits definierten Kennlinie ein und bestätigen Sie mit **Close**.



7. Cycle time: 100 (die Umrechnung des Rohwerts findet alle 100 ms statt).

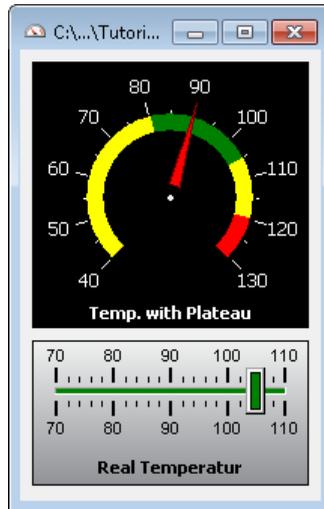
Information: Damit sind die Konfigurationsarbeiten zur Lösung dieser Aufgabe abgeschlossen.

Aktion: Die Konfigurationsdatei wird als Projekt Aufgabe 4b auf dem PC gespeichert.

Aktion: Die Konfigurationsdatei wird über den CAN-Bus an den PCAN-Router Pro übermittelt (Upload).

Ergebnis: Ein kontinuierlich steigender Eingangswert (Nachricht 0x100) wird mit einem Plateau versehen und ausgegeben (Nachricht 0x200).

Information: Sofern man über das PCAN-Explorer-Add-in Instruments Panel (Art.-Nr. IPES-005088) verfügt, lassen sich die Werte grafisch darstellen.



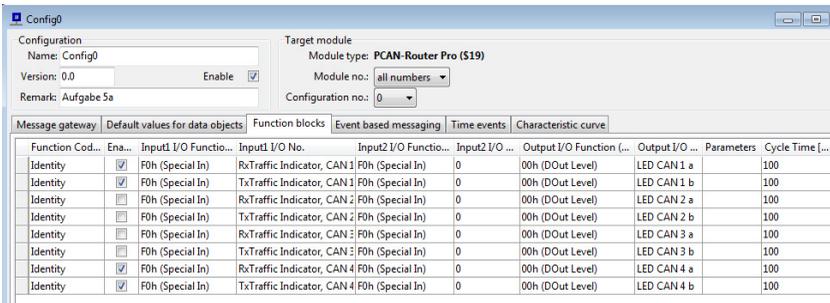
4.12 Aufgabe 5a: LED-Aktivität bei CAN-Senden/Empfangen

Information: Senden und Empfangen von CAN-Nachrichten kann visualisiert werden. Hierzu verfügt der PCAN-Router Pro über zwei LEDs für jeden CAN-Kanal. Im vorliegenden Fall sollen die LEDs CAN-Aktivität auf den Bussen signalisieren, getrennt für Rx und Tx Richtung.

Aktion: Anlegen einer leeren Konfigurationsdatei mit **File -> New**.

Aktion: Anlegen einer neuen Konfiguration innerhalb der Konfigurationsdatei: den Menüpunkt **Edit -> New Configuration** auswählen.

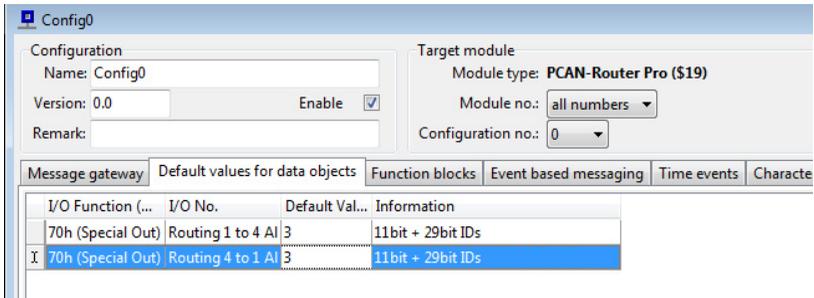
Aktion: Öffnen des Konfigurationsfenster (durch Klicken auf das Icon Config0 am linken Rand des PPCAN-Editors). Auswählen des Tabs Function Blocks. Dort muss pro LED ein neuer Eintrag erzeugt werden. Das Kontextmenü öffnen (Rechtsklick) und **Add Record** auswählen:



1. Function Code: Identity (eine Variable wird in eine andere kopiert).
2. Enable: Ja, dieser Block soll aktiv sein.
3. Eingang 1: Systemvariable Special In, TrafficIndicator (aktiv je für 100 ms).
4. Eingang 2: F0-Special In und none (= nicht benutzt).
5. Ausgang: Dout-Level und die entsprechende LED.
6. Parameter: nicht notwendig.
7. Cycle time: 100 (die Aktualisierung der LEDs findet alle 100 ms statt).

Bemerkung: Die Leuchtdauer 100 ms ist in der Firmware fest verankert und hat nichts mit dem Parameter CycleTime des Identity-Funktionsblocks zu tun.

Aktion: Jetzt ist es noch notwendig, alle eintreffenden Nachrichten (nur 11-Bit, nicht 29-Bit!) von CAN-1 auf CAN-4 zu routen und umgekehrt. Dazu im Tab Default values for data objects die entsprechenden Einstellungen aktivieren:



1. I/O-Funktion: SpecialOut (eine Gruppe von Gerätefunktionen).
2. I/O-No: Routing 1 to 4 All und Routing 4 to 1 All.
3. Default value: **3** = Summe aus **1** (routet nur 11-Bit-IDs) und **2** (routet nur 29-Bit-IDs).
4. Information: Eine Beschreibung, was diese Zeile macht.

Information: Damit sind die Konfigurationsarbeiten zur Lösung dieser Aufgabe abgeschlossen.

Aktion: Die Konfigurationsdatei wird als Projekt Aufgabe 5a auf dem PC gespeichert.

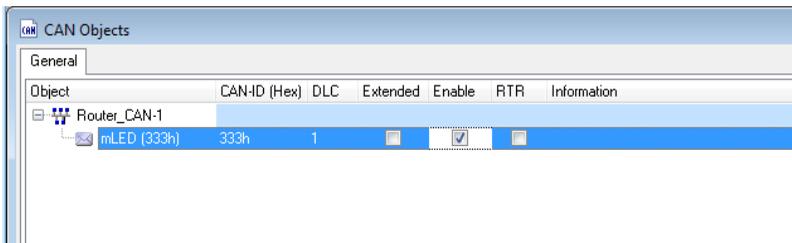
Aktion: Die Konfigurationsdatei wird über den CAN-Bus an den PCAN-Router Pro übermittelt (Upload).

Ergebnis: Bei Empfang oder Aussenden einer CAN-Nachricht wird die zugeordnete LED für 100 ms eingeschaltet.

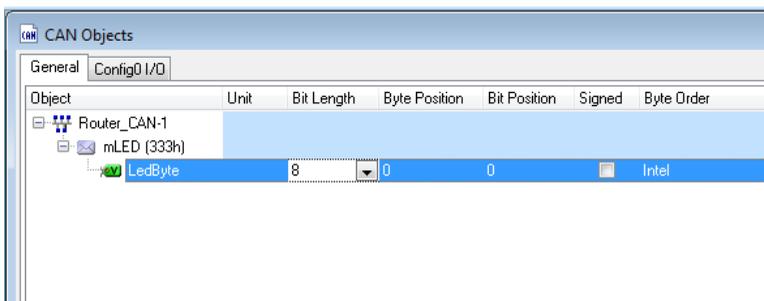
4.13 Aufgabe 5b: LED manuell / bedingt ansteuern

Information: Am Bus Router_CAN-1 soll eine 1-Byte Nachricht mLED mit einem Analogwert LedByte (0..255) empfangen werden. Wenn der Analogwert 126 überschreitet, soll die LED an sein. Liegt der Wert unter 127, bleibt die LED aus.

Aktion: Anlegen einer leeren Konfigurationsdatei mit **File -> New**. Definieren einer Empfangs-Nachricht mLED (0x333) mit einem 8-Bit-CAN-Signal LedByte:

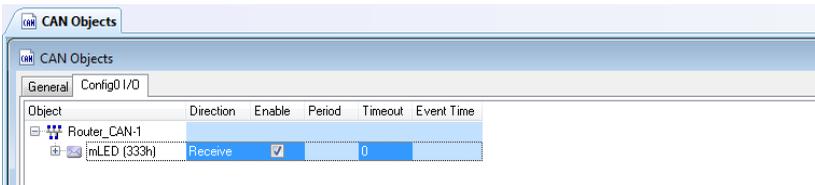


1. CAN-ID: 0x333.
2. DLC: 1 (die CAN-Nachrichten sind je ein Byte lang).
3. Enable: Ja.

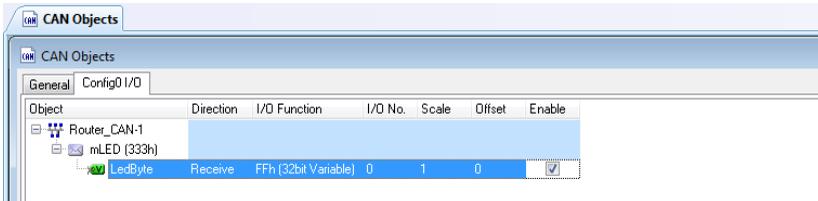


1. Bit Length: 8 (Wertebereich 256 genügt).
2. Signed: nein, kein Vorzeichen.

Aktion: Anlegen einer neuen Konfiguration innerhalb der Konfigurationsdatei: dazu den Menüpunkt **Edit -> New Configuration** auswählen. Importieren der Empfangs-Nachricht in die neue Konfiguration und Eintragen der Parameter:

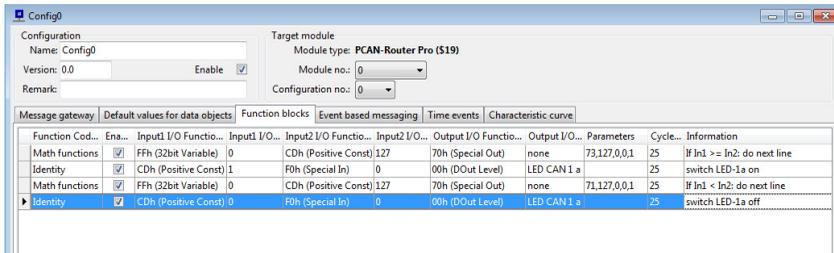


1. Direction: 0x333 wird empfangen (also Receive).
2. Enable: Ja.



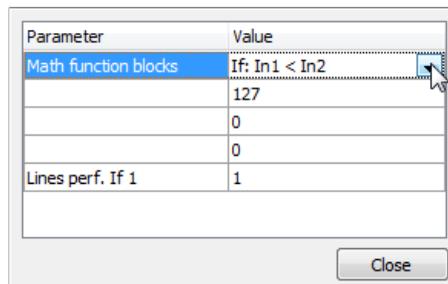
I/O-Funktion, I/O-Number: Das Signal LedByte aus der Nachricht 0x333 wird in die Variable #0 transferiert.

Aktion: Öffnen des Konfigurationsfenster (durch Klicken auf das Icon Config0 am linken Rand des PPCAN-Editors). Auswählen des Tabs Function Blocks, Öffnen des Kontextmenüs (Rechtsklick), **Add record** auswählen und vier neue Funktionsblöcke anlegen:



Zeile 1:

1. Function block: Math Function (IF).
2. Enable: Ja.
3. Input#1 I/O-Funktion: 32-Bit-Variablen.
4. Input#1 I/O-Nummer: 0.
5. Input#2 I/O-Funktion: Konstante.
6. Input#2 I/O-Nummer: Wert **127**.
7. Output I/O-Funktion: nicht verwendet (unverändert lassen).
8. Output I/O-Nummer: nicht verwendet (unverändert lassen).
9. Parameter: Vergleich, ob **In1 größer gleich 127** ist. Wenn ja, dann nächste Zeile ausführen (LED einschalten). Wenn nein, dann 1 Zeile überspringen. Es öffnet sich folgendes Fenster:



Bei dem Parameter Math function blocks, F2 oder leichtem Klick in das Feld Value. Aus der Auswahl den richtigen Wert übernehmen. Die weiteren Parameter geben Sie direkt ein. Bestätigen Sie mit **Close**.

10. Cycle: alle 25 ms wird dieser Block bearbeitet.

Zeile 2:

1. Function block: Identity (1:1 kopieren).
2. Enable: Ja.
3. Input#1 I/O-Funktion: Konstante.
4. Input#1 I/O-Number: Wert **1**.
5. Input#2 I/O-Funktion: nicht verwendet (unverändert lassen).
6. Input#2 I/O-Number: nicht verwendet (unverändert lassen).
7. Output I/O-Funktion: wird auf die folgende Hardware-Ressource geschrieben.
8. Output I/O-Number: LED CAN-1a.
9. Parameter: keine.
10. Cycle: alle 25 ms wird dieser Block bearbeitet.

Zeile 3:

1. Function block: Math Function (ELSE bzw. ein IF mit umgekehrter Bedingung).
2. Enable: Ja.
3. Input#1 I/O-Funktion: 32-Bit-Variable.
4. Input#1 I/O-Number: #0.
5. Input#2 I/O-Funktion: Konstante.
6. Input#2 I/O-Number: Wert **127**.
7. Output I/O-Funktion: nicht verwendet (unverändert lassen).

8. Output I/O-Number: nicht verwendet (unverändert lassen).
9. Parameter: Vergleich, ob **In1 kleiner 127** ist. Wenn ja, dann nächste Zeile ausführen (LED ausschalten). Wenn nein, dann 1 Zeile überspringen.
10. Cycle: alle 25 ms wird dieser Block bearbeitet.

Zeile 4:

1. Function block: Identity (1:1 kopieren).
2. Enable: Ja.
3. Input#1 I/O-Funktion: Konstante.
4. Input#1 I/O-Number: Wert **0**.
5. Input#2 I/O-Funktion: nicht verwendet (unverändert lassen).
6. Input#2 I/O-Number: nicht verwendet (unverändert lassen).
7. Output I/O-Funktion: wird auf die folgende Hardware-Ressource geschrieben.
8. Output I/O-Number: LED CAN-1a.
9. Parameter: keine.
10. Cycle: alle 25 ms wird dieser Block bearbeitet.

Information: Damit sind die Konfigurationsarbeiten zur Lösung dieser Aufgabe abgeschlossen.

Aktion: Die Konfigurationsdatei wird als Projekt Aufgabe 5b auf dem PC gespeichert.

Aktion: Die Konfigurationsdatei wird über den CAN-Bus an den PCAN-Router Pro übermittelt (Upload).

Ergebnis: Bei Empfang eines Wertes bis **126 (0x7E)** bleibt die LED-1a aus. Ab einem Wert größer gleich **127 (0x7F)** wird die LED eingeschaltet.

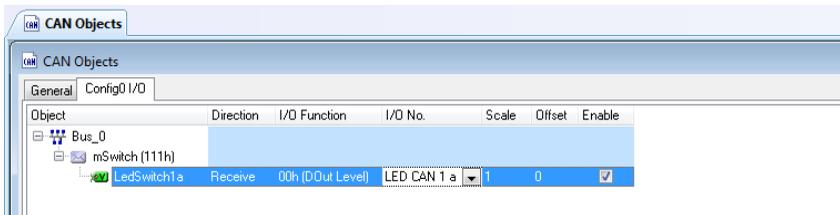
4.14 Aufgabe5c: LED extern ansteuern

Information: Zum externen Schalten einer LED genügt der Empfang eines 1-Bit-Signals über CAN. Dieses 1-Bit-Signal wird der Resource Dout-Level LED CAN-1a zugewiesen.

Aktion: Anlegen einer leeren Konfigurationsdatei mit **File -> New**. Definieren einer Empfangs-Nachricht mSwitch (0x111) mit einem 1-Bit-CAN-Signal:



Aktion: Anlegen einer neuen Konfiguration innerhalb der Konfigurationsdatei, Importieren der global definierten CAN-Nachricht und Eintragen der Parameter:



1. Direction: Receive (der Schalterwert kommt über CAN).
2. I/O-Funktion: DOut Level (interne Hardware-Ressource).
3. I/O-Number: LED CAN-1a.

Information: Damit sind die Konfigurationsarbeiten zur Lösung dieser Aufgabe abgeschlossen.

Aktion: Die Konfigurationsdatei wird als Projekt Aufgabe 5c auf dem PC gespeichert.

Aktion: Die Konfigurationsdatei wird über den CAN-Bus an den PCAN-Router Pro übermittelt (Upload).

Ergebnis: Mittels Übertragung des definierten Bits kann eine LED an- oder ausgeschaltet werden.



Tip: Wenn man das Beispiel für 8 LEDs erweitert und über einen PCAN-Explorer (Art.-Nr. IPES-005028) verfügt, kann man ein Symbolfile (gemäß Anleitung) erstellen und die Schalterwerte darin als Symbole definieren.

```
{SEND}
```

```
[mSwitch]
```

```
ID=111h
```

```
Picture=---b---a ---d---c ---f---e ---h---g
```

```
a=TxLedCan1 bit
```

```
b=RxLedCan1 bit
```

```
c=TxLedCan2 bit
```

```
d=RxLedCan2 bit
```

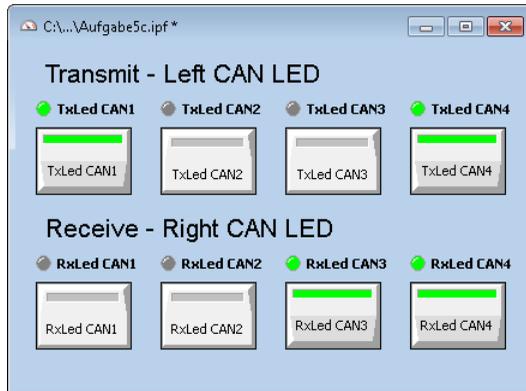
```
e=TxLedCan3 bit
```

```
f=RxLedCan3 bit
```

```
g=TxLedCan4 bit
```

```
h=RxLedCan4 bit
```

Bemerkung: Wenn man über einen PCAN-Explorer (Art.-Nr. IPES-005028) und dem Instruments Panel Add-in (Art.-Nr. IPES-005088) verfügt, lässt sich die Bedienung der Schalterbits leicht visualisieren:



4.15 Aufgabe 5d: Beeper ansteuern (Dauerton)

Information: Ähnlich wie das Aktivieren einer LED kann auch ein Dauerton des internen Beeper (oder ein definiertes Muster) von außen gesteuert werden.

Aktion: Konfigurieren des PCAN-Router Pro, so dass ein 32-Bit-Empfangssignal direkt der Firmware-Ressource BeeperPattern zugewiesen wird:



1. I/O Function: Special Out.
2. I/O Nummer: Beeper Pattern.
3. Enable: Ja.

Information: Damit sind die Konfigurationsarbeiten zur Lösung dieser Aufgabe abgeschlossen.

Aktion: Die Konfigurationsdatei wird als Projekt Aufgabe 5d auf dem PC gespeichert.

Aktion: Die Konfigurationsdatei wird über den CAN-Bus an den PCAN-Router Pro übermittelt (Upload).

Information: Zum Definieren eines Musters muss vom Anwender ein 32-Bit-Wert errechnet werden.

Die Struktur dieses Wertes ist wie folgt:

Die oberen 24 Bit definieren das Ton-Muster.
 Die unteren 5 Bit geben an, wie viele Bits des Musters abgespielt werden.
 Dazwischen sind 2 unbenutzte Bits und 1 Bit entscheidet, ob die Folge endlos (1=continuous) oder einmalig (0=one-shot) abgespielt werden soll.

Die Abspielgeschwindigkeit beträgt 100 ms pro Bit

Beispiel: Programmieren eines Dauertons:

```
10000000.00000000.00000000.00100001
mmmmmmmm.mmmmmmm.mmmmmmm          = 1* 100ms Ton
                                     .uu    = unbenutzt
                                     W      = 0=OneShot; 1=Endlos wiederholen
                                     11111 = Musterlänge: 1 Bit
```

Abspieldauer: 24 * 100 ms = **2,4 s**

Muster: *0x80, 0x00, 0x00, 0x21*

Aktion: Dieses Muster per PCAN-View (oder PCAN-Explorer) zum PCAN-Router Pro senden:

0x100-8-"21 00 00 80 00 00 00 00", 0

Zum Abschalten alle Bits wieder auf 0 setzen:

0x100-8-"00 00 00 00 00 00 00 00", 0

4.16 Aufgabe 5e: Beeper ansteuern (Tonfolge)

Information: Mit der Konfiguration aus Aufgabe 5d können verschiedene Tonfolgen ausgelöst werden.

Beispiel: Programmieren des SMS-Morsecodes:

```

10000000.00000000.00000000.00100001
mmmmmm                = S
  mm.mmmmm            = M
    mm.mmm           = S
      mmm.          = Pause=400 ms, da Länge=24 gesetzt
        uu         = unbenutzt (bzw. wdh.)
          w        = 0=OneShot, 1=Endlos wiederholen
            11111 = Muster-Länge: 1 Bit

Abspieldauer: 24 *100 ms = 2,4 s

Muster: 0xAB. 0x32. 0xA0. 0x1F
    
```

Aktion: Dieses Muster per PCAN-View oder PCAN-Explorer zum PCAN-Router Pro senden:

0x100-8-"1F A0 32 AB 00 00 00 00", 0

Zum Abschalten alle Bits wieder auf 0 setzen:

0x100-8-"00 00 00 00 00 00 00 00", 0

		Message	DLC	Data	Cycle Time	Count	
Receive	<input type="checkbox"/>	<Empty>					
Transmit	<input type="checkbox"/>	100h	4	00 00 00 00	Wait	2	Manual
		100h	4	01 00 00 80	Wait	1	Manual
		100h	4	1F A0 32 AB	Wait	1	Manual



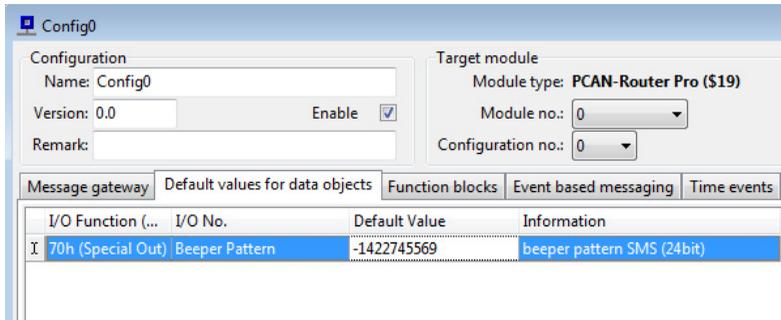
Tip: Man kann das Beep-Muster aus Aufgabe 5e auch als Konstante im Tab Default values for data objects verankern: Dazu ist bei den Default-Werten eine 32-Bit-Variable mit dem entsprechenden Wert zu belegen und darf immer nur gelesen werden. Werden hier mehrere Bitmuster hinterlegt, kann z. B. ein eingehendes CAN-Signal oder ein Rechenergebnis als Selektor zwischen verschiedenen Tonfolgen dienen.

Bemerkung 1: Zuweisen des Musters zu einer Variablen erfolgt im **Zweierkomplement** (weil der PPCAN-Editor bei Defaultwerten nur SIGNED versteht).

Dazu den Windows-Taschenrechner öffnen

- └ Windows-Taschenrechner auf **HEX** schalten
- └ **Dword** auswählen (= 32 Bit)
- └ Zahl eintippen: **AB32A01F**
- └ Vorzeichenwechsel: **+/-**
- └ Windows-Taschenrechner auf **DEZ** schalten
- └ erneut Vorzeichenwechsel: **+/-**
- └ Eintrag im PPCAN-Editor = **-1422745569**

Am linken Bildrand im Navigations-Fenster die Konfiguration Config0 anklicken. Im Konfigurationsfenster zum Tab Default values for data objects wechseln:



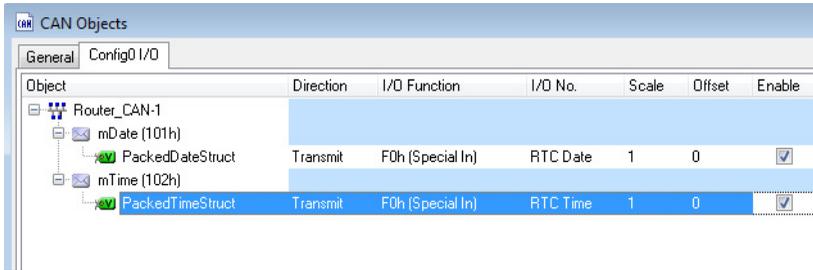
1. I/O-Funktion: Special Out.
2. I/O-Number: Beeper Pattern.
3. Default Value: -1422745569.

Bemerkung 2: Der Dauerton aus Aufgabe 5d müsste als Wert **-2147483615** im PPCAN-Editor hinterlegt werden.

4.17 Aufgabe 6a: Ausgabe von Datum/Uhrzeit (Hardware-Diagnose)

Information: Datum und Uhrzeit werden im PCAN-Router Pro von einer Echtzeituhr bereitgestellt. Die Werte können aus den internen Ressourcen ausgelesen und anschließend per CAN übertragen und z. B. angezeigt werden.

Aktion: Definieren von je einer Transmit-Nachricht (DLC = 4), die der PCAN-Router Pro zyklisch mit den Datums-/Zeitwerten befüllt und dann über das Netz Router_CAN-1 aussendet:



Information: Damit sind die Konfigurationsarbeiten zur Lösung dieser Aufgabe abgeschlossen.

Aktion: Die Konfigurationsdatei wird als Projekt Aufgabe 6a auf dem PC gespeichert.

Aktion: Die Konfigurationsdatei wird über den CAN-Bus an den PCAN-Router Pro übermittelt (Upload).

Information: Wenn man über einen PCAN-Explorer (Art.-Nr. IPES-005028) verfügt, kann man mittels einer Symbol-Datei die gepackten Strukturen für Datum und Uhrzeit im Klartext darstellen lassen.

```
{SEND}
[mDate]
ID=101h
//      Byte 0   Byte 1   Byte 2   Byte 3   Byte 4   Byte 5   Byte 6   Byte 7
//      76543210 76543210 76543210 76543210 76543210 76543210 76543210 76543210
Picture=yyyyyy mmmmm dddddd nnnn---
y=year      unsigned
m=month     unsigned
d=dayofmonth unsigned
n=dayofweek unsigned

[mTime]
ID=102h
//      Byte 0   Byte 1   Byte 2   Byte 3   Byte 4   Byte 5   Byte 6   Byte 7
//      76543210 76543210 76543210 76543210 76543210 76543210 76543210 76543210
Picture=fffffff ssssss mmmmmm hhhhhh
f=fractseconds unsigned
s=seconds     unsigned
m=minutes     unsigned
h=hours       unsigned
```

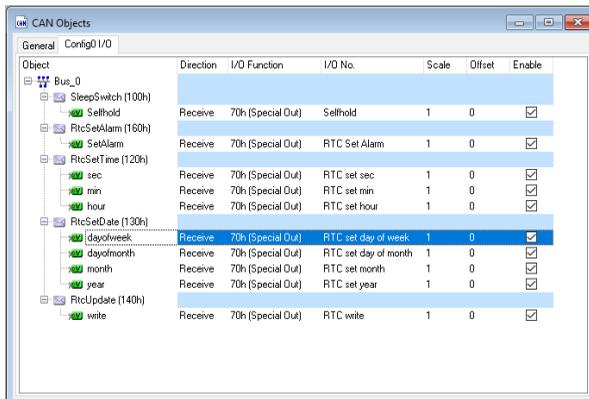
Damit werden die vom PCAN-Router Pro gesendeten Nachrichten wie folgt decodiert:

Symbol /...	Multiplexer / DLC	Data	Timeouts	Period	Count
101h RtcDate	4 <Empty>/4	0A 02 12 40 year =10 month =2 dayofmonth=18 dayofweek =4	0	100,0	7037
102h RtcTime	4 <Empty>/4	1F 12 1A 0A fractseconds=31 seconds =18 minutes =26 hours =10	0	100,0	7037

4.18 Aufgabe 6b: Setzen von Datum/Uhrzeit (Hardware-Diagnose)

Information: Zum Setzen der Uhrzeit werden die Datums- und Zeit-Elemente einzeln versorgt, es ist also keine gepackte Struktur wie in Aufgabe 6a.

Definition: Es sind 3 separate Nachrichten für Datum, Uhrzeit und den Aktivierungsbefehl zu implementieren. Die einzelnen Firmware-Variablen sind alle jeweils 8 Bit lang:



Information: Damit sind die Konfigurationsarbeiten zur Lösung dieser Aufgabe abgeschlossen.

Aktion: Die Konfigurationsdatei wird als Projekt Aufgabe 6b auf dem PC gespeichert.

Aktion: Die Konfigurationsdatei wird über den CAN-Bus an den PCAN-Router Pro übermittelt (Upload).

Ergebnis: Wenn jetzt entsprechend versorgte CAN-Nachrichten zum PCAN-Router Pro gesendet werden, kann so die interne Uhrzeit eingestellt werden:

smit	120h SetRtcTime	3 <Empty>/3	1E 1D 0C ☐ SetRtc_seconds=30 SetRtc_minutes=29 SetRtc_hours =12	Wait	0
	130h SetRtcDate	4 <Empty>/4	04 12 02 0A ☐ SetRtc_dayofweek =4 SetRtc_dayofmonth=18 SetRtc_month =2 SetRtc_year =10	Wait	0
	140h RtcWrite	1 <Empty>/1	01 write=1	Wait	0

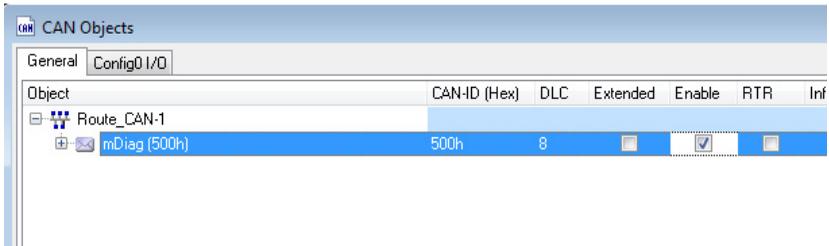
Bemerkung: Mit den gezeigten Nachrichten wird **Donnerstag, 18.Februar 2010, 12:29:30 Uhr** gesetzt.

4.19 Aufgabe 6c: Ausgabe der Modul-ID (Hardware-Diagnose)

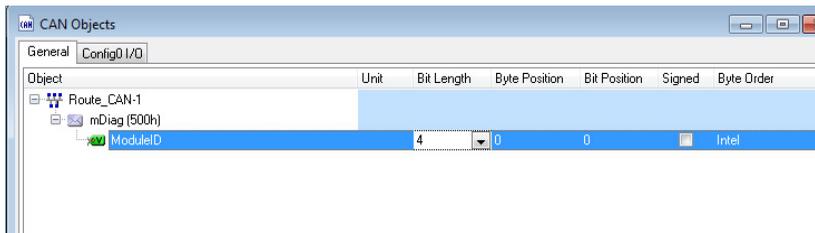
Information: Die Modul-ID ist ein 4-Bit-Wert, der ab Werk auf 0 eingestellt ist und im Innern des PCAN-Router Pro per Drehschalter verändert werden kann. Sie hat verschiedene Funktionen, z. B. wählt sie aus einer Datei mit mehreren Konfigurationen die der Schalterstellung entsprechende aus. Bei unerklärlichem Verhalten einer neuen Konfiguration ist einer der ersten Debug-Schritte das Feststellen der Modul-ID. Es könnte nämlich sein, dass man die eine Konfiguration erfolglos editiert, weil jedes Mal die andere ausgeführt wird.

Definition: Senden einer CAN-Nachricht mDiag mit der ID 0x500, Länge 8 Bytes auf dem Bus Router_CAN-1. Diese Nachricht enthält das 4-Bit-Signal ModuleID, dass die aktuelle Stellung des Modul-ID-Schalters anzeigt.

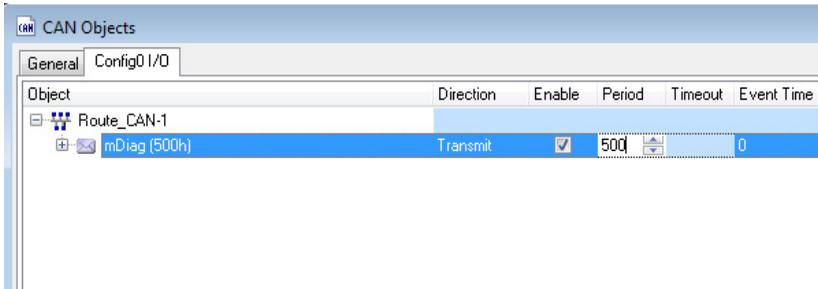
Aktion: Anlegen einer Nachricht mDiag im General-Tab, Länge = 8 Bytes:



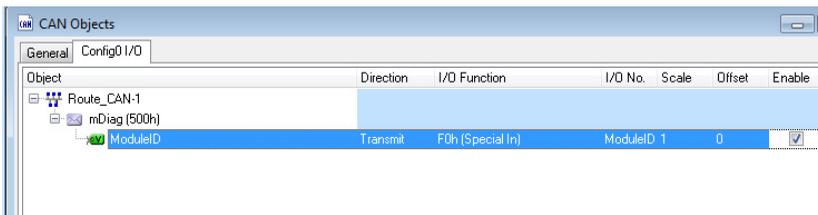
Anlegen eines neuen CAN-Signals (4-Bit lang, unsigned) im Tab General:



Importieren der Nachricht inklusiv des Signals in die Konfiguration (Zyklus 500 ms):



Befüllen des Signals mit dem Schalterwert Module-ID des PCAN-Routers Pro:



1. I/O-Funktion: F0-Special In.
2. I/O-Number: Module-ID.
3. Enable: Ja.

Information: Damit sind die Konfigurationsarbeiten zur Lösung dieser Aufgabe bereits abgeschlossen.

Aktion: Die Konfigurationsdatei wird als Projekt Aufgabe 6c auf dem PC gespeichert.

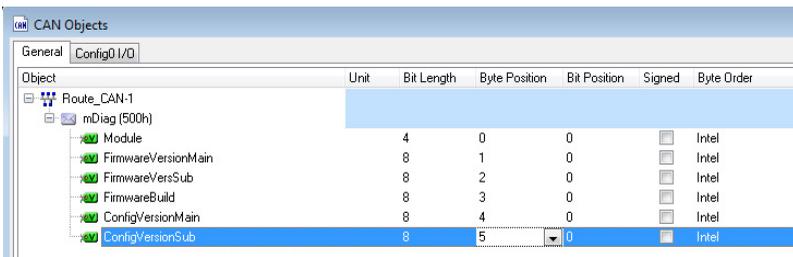
Aktion: Die Konfigurationsdatei wird über den CAN-Bus an den PCAN-Router Pro übermittelt (Upload).

Reaktion: Es wird in der Nachricht mDiag (mit der ID 0x500) die Modul-ID übertragen.

Information: Jede Änderung der Modul-ID wird sofort angezeigt, sie wird aber erst nach Reset des Geräts (z. B. Power Off/On) intern verwendet.

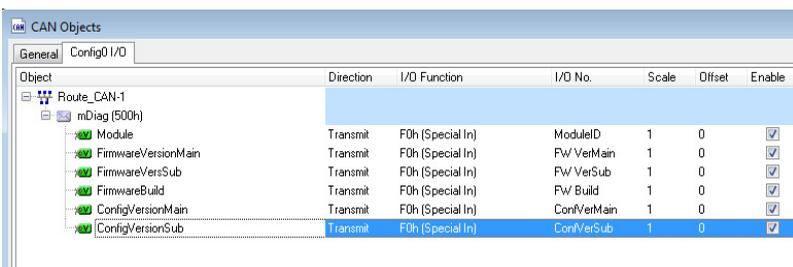
4.20 Aufgabe 6d: Ausgabe Firmwarestand und Konfigurations-Version (Hardware-Diagnose)

Aktion: Der mDiag-Nachricht werden weitere 8-Bit-Variablen hinzugefügt: Firmwarestand (3-stellig), und Konfigurations-Version (2-stellig):



Object	Unit	Bit Length	Byte Position	Bit Position	Signed	Byte Order
Route_CAN-1						
mDiag (500h)						
Module	4	0	0		<input type="checkbox"/>	Intel
FirmwareVersionMain	8	1	0		<input type="checkbox"/>	Intel
FirmwareVersSub	8	2	0		<input type="checkbox"/>	Intel
FirmwareBuild	8	3	0		<input type="checkbox"/>	Intel
ConfigVersionMain	8	4	0		<input type="checkbox"/>	Intel
ConfigVersionSub	8	5	0		<input type="checkbox"/>	Intel

Zuweisen der Datenquellen:



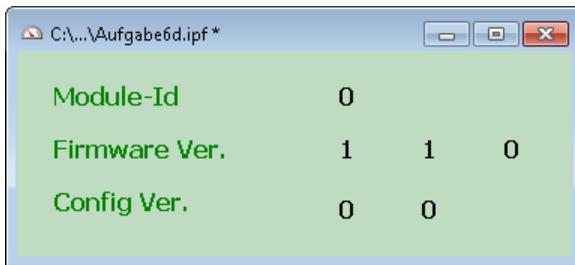
Object	Direction	I/O Function	I/O No.	Scale	Offset	Enable
Route_CAN-1						
mDiag (500h)						
Module	Transmit	FDh (Special In)	ModuleID	1	0	<input checked="" type="checkbox"/>
FirmwareVersionMain	Transmit	FDh (Special In)	Pw VerMain	1	0	<input checked="" type="checkbox"/>
FirmwareVersSub	Transmit	FDh (Special In)	Pw VerSub	1	0	<input checked="" type="checkbox"/>
FirmwareBuild	Transmit	FDh (Special In)	Pw Build	1	0	<input checked="" type="checkbox"/>
ConfigVersionMain	Transmit	FDh (Special In)	ConfVerMain	1	0	<input checked="" type="checkbox"/>
ConfigVersionSub	Transmit	FDh (Special In)	ConfVerSub	1	0	<input checked="" type="checkbox"/>

Information: Damit sind die Konfigurationsarbeiten zur Lösung dieser Aufgabe bereits abgeschlossen.

Aktion: Die Konfigurationsdatei wird als Projekt Aufgabe 6d auf dem PC gespeichert.

Aktion: Die Konfigurationsdatei wird über den CAN-Bus an den PCAN-Router Pro übermittelt (Upload).

Bemerkung: Wenn man über einen PCAN-Explorer (Art.-Nr. IPES-005028) und das Instruments Panel Add-in (Art.-Nr. IPES-005088) verfügt, kann man (mittels einer Symbol-Datei) alle gesendeten Signale lesbar darstellen:

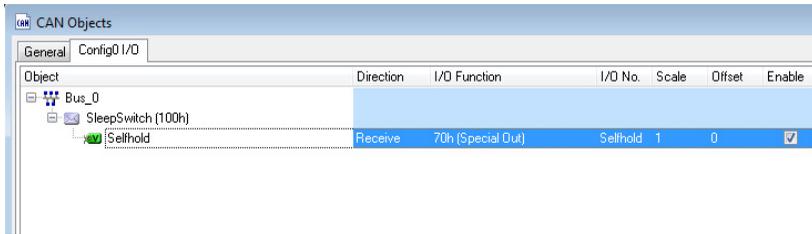


Module-Id	0		
Firmware Ver.	1	1	0
Config Ver.	0	0	

4.21 Aufgabe 7a: Sleep/Wake-Up über CAN

Information: Der PCAN-Router Pro ist an allen vier Ports mit Wake-Up-fähigen CAN-Transceivern ausgestattet. Falls sich das Gerät im energiesparenden Sleep-Mode befindet, wird eine beliebige eingehende Nachricht das Gerät aufwecken.

Aktion: Es wird eine Empfangs-Nachricht definiert, die die interne Variable Selfhold auf 0 setzt, dadurch geht der PCAN-Router Pro in den sogenannten Sleep-Mode (z. B. um die Fahrzeugbatterie zu schonen):



Information: Damit sind die Konfigurationsarbeiten zur Lösung dieser Aufgabe bereits abgeschlossen.

Aktion: Die Konfigurationsdatei wird als Projekt Aufgabe 7a auf dem PC gespeichert.

Aktion: Die Konfigurationsdatei wird über den CAN-Bus an den PCAN-Router Pro übermittelt (Upload).

Ergebnis: Wird die Nachricht ID 0x100 mit dem Signal Selfhold = 0 empfangen, dann geht der PCAN-Router Pro sofort schlafen. Die nächste Nachricht (auch 0x100) weckt das Gerät wieder auf (Status-LED blinkt).

Information: Beim Wecken wird die interne Variable Selfhold mit 1 initialisiert. Dadurch bleibt der PCAN-Router Pro (bis auf weiteres) eingeschaltet.

4.22 Aufgabe 7b: Sleep/wake-Up über Hardware-Pin

Information: Eine weitere Methode, um den PCAN-Router Pro aufzuwecken, ist die Beaufschlagung des Pins 4 an den D-Sub-Buchsen 3 und 4 mit einer Spannung zwischen 8 und 26 V. Hier kann z. B. die Klemme 15 (= Zündung) eines Fahrzeugs angeschlossen werden.

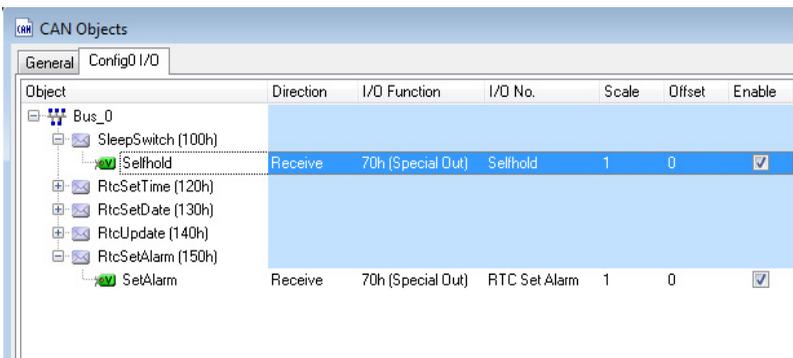
4.23 Aufgabe 7c: Sleep/Wake-Up über Alarmzeit

Information: Die dritte Methode zum Wecken des Geräts ist das Stellen des Weckers (= Alarmzeit der Echtzeit-Uhr RTC).

Bemerkung: Angenommen, die aktuelle Uhrzeit sei Donnerstag, 18. Februar 2010, 12:29:30 Uhr. Eine sinnvolle Weckzeit wäre z. B. Donnerstag, 18. Februar 2010, 12:30:00 Uhr. Das Stellen der Uhr wurde bereits in *4.18 Aufgabe 6b: Setzen von Datum/Uhrzeit (Hardware-Diagnose) auf Seite 66* demonstriert.

Information: Für diese Aufgabe werden die Empfangs-Nachrichten **RtcSetAlarm** (ID = 0x160) zum Setzen der Weckzeit (Länge = 4 Bytes) und **SleepSwitch** (ID = 0x100) zum Versetzen des PCAN-Router Pro in den Sleep-Mode benötigt (siehe dazu *Aufgabe 7a: Sleep/Wake-Up über CAN auf Seite 71*).

Aktion: Laden Sie die Konfigurationsdatei **7c**. Wenn Sie zum Probieren eine andere Weckzeit einstellen möchten, finden Sie mehr Informationen in Kapitel *4.18 Aufgabe 6b: Setzen von Datum/Uhrzeit (Hardware-Diagnose) auf Seite 66*.



Object	Direction	I/O Function	I/O No.	Scale	Offset	Enable
Bus_0						
SleepSwitch (100h)						
Selthold	Receive	70h (Special Out)	Selthold	1	0	<input checked="" type="checkbox"/>
RtcSetTime (120h)						
RtcSetDate (130h)						
RtcUpdate (140h)						
RtcSetAlarm (150h)						
SetAlarm	Receive	70h (Special Out)	RTC Set Alarm	1	0	<input checked="" type="checkbox"/>

Information: Damit sind die Konfigurationsarbeiten zur Lösung dieser Aufgabe bereits abgeschlossen.

Aktion: Die Konfigurationsdatei wird als Projekt Aufgabe 7c auf dem PC gespeichert.

Aktion: Die Konfigurationsdatei wird über den CAN-Bus an den PCAN-Router Pro übermittelt (Upload).

Information: Nach dem Setzen der Weckzeit 12:30:00 Uhr muss der PCAN-Router Pro noch mittels Selfhold = 0 in den Sleep-Mode versetzt werden (dazu Nachricht **0x100** senden).

Ergebnis: Sobald die Nachricht-ID 0x100 mit dem Signal Selfhold = 0 empfangen wird, geht der PCAN-Router Pro schlafen. Mit Erreichen der Weckzeit geht das Gerät selbsttätig wieder in Betrieb.

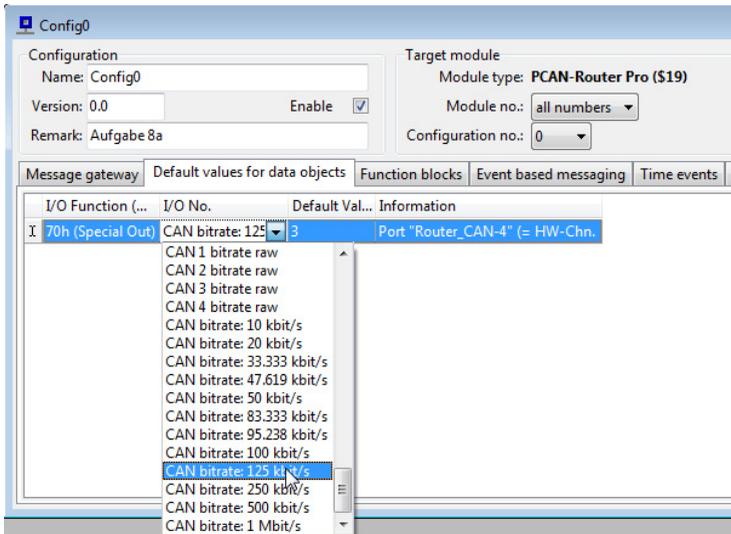
4.24 Aufgabe 8a: Ändern der Bitrate

Information: Abhängig von den bestückten CAN-Transceivern stellt der PCAN-Router Pro selbsttätig folgende Bitraten ein:

Interface-Typ	CAN-Transceiver	Default-Bitrate	Wake-Up-fähig
HS	TJA-1041 (default)	500 kbit/s	ja
HS opto ³	TJA-1040	500 kbit/s	nein
HS	TJA-1040	500 kbit/s	nein
HS	82C251	500 kbit/s	nein
LS-DW	TJA-1054	125 kbit/s	ja
LS-SW	TH-8056	33,3 kbit/s	ja

Diese Werte können beim Starten des PCAN-Router Pro durch geeignete Einträge in der Konfiguration, Tab **Default values for data objects** überschrieben werden:

³ Verfügbarkeit HS-opto anfragen.



- **I/O No.:** Hier gewünschte Übertragungsrate auswählen.
- **Default Value:** CAN-Kanal-Nummer (0...3).

Bemerkung: Da die PC-Software PPCAN-Editor nicht weiß, welche Transceiver im Gerät PCAN-Router Pro aktuell verbaut sind, werden alle denkbaren Übertragungsraten angeboten. Es ist also darauf zu achten, dass sinnvolle Werte verwendet werden, z. B. bei TH-8056 ist die maximale Bitrate 83,3 kbit/s, bei TJA-1040 ist die minimale Bitrate 40 kbit/s.

4.25 Aufgabe 9a: Multiplexer-Nachricht automatisch senden

Information: Das Beispiel sendet eine CAN-Nachricht mit variierenden Variablen, die durch einen Multiplexer (Zwischenadressierung) angesteuert werden.

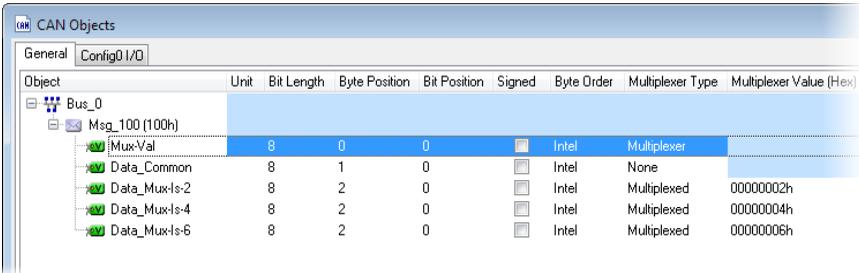
Definition: Die Beispielnachricht hat folgende Eckdaten:

- └─ CAN-ID: 100h
- └─ Länge: 3 Bytes
- └─ Bit-Belegung (Variablen):

Byte Nr./ Startbit	Bits	Benennung	Verwendung
0/0	8	Mux-Val	Multiplexer
1/0	8	Data_Common	Variable unabhängig vom Multiplexer (immer vorhanden)
2/0	8	Data_Mux-Is-2 Data_Mux-Is-4 Data_Mux-Is-6	Veränderliche Variable abhängig vom Multiplexerwert im Datenbyte 0 (hier: 2, 4 oder 6)

Aktion: Im Fenster **CAN Objects** unter **General** die neue CAN-Nachricht 100h mit den oben angegebenen Eckdaten anlegen. Dabei für die Variablen in den Spalten **Multiplexer Type** und **Multiplexer Value** die folgenden Angaben machen:

Benennung	Multiplexer Type / Value	Erläuterung
Mux-Val	Multiplexer	Wert bestimmt, welche Multiplexed-Variable verwendet wird.
Data_Common	None	Variable wird unabhängig vom Multiplexer immer in dieser CAN-Nachricht verwendet.
Data_Mux-Is-2	Multiplexed / 2	Bei den Multiplexer-Werten 2, 4 und 6 wird die jeweilig zugeordnete Variable verwendet.
Data_Mux-Is-4	Multiplexed / 4	
Data_Mux-Is-6	Multiplexed / 6	



Object	Unit	Bit Length	Byte Position	Bit Position	Signed	Byte Order	Multiplexer Type	Multiplexer Value (Hex)
Bus_0								
Msg_100 (100h)								
Mux_Val		8	0	0	<input type="checkbox"/>	Intel	Multiplexer	
Data_Common		8	1	0	<input type="checkbox"/>	Intel	None	
Data_Mux-Is-2		8	2	0	<input type="checkbox"/>	Intel	Multiplexed	0000002h
Data_Mux-Is-4		8	2	0	<input type="checkbox"/>	Intel	Multiplexed	0000004h
Data_Mux-Is-6		8	2	0	<input type="checkbox"/>	Intel	Multiplexed	0000006h

Bemerkung: Für die Angabe in der Spalte **Multiplexer Type** gibt es die folgenden Möglichkeiten:

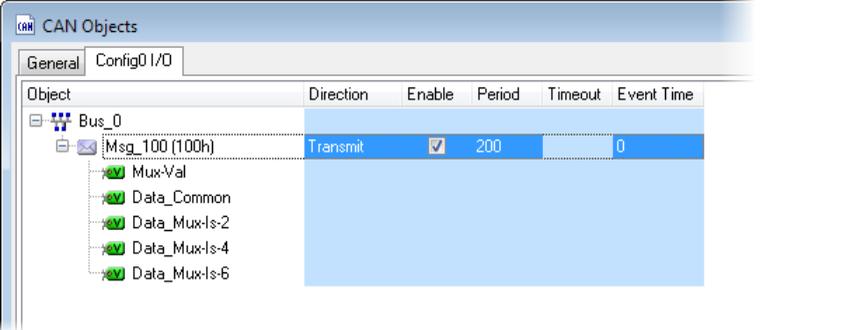
- **Multiplexer:** Die Variable enthält den Multiplexerwert (Datentyp Unsigned). Dieser Multiplexer-Typ darf nur einmal in einer Nachricht verwendet werden und muss in der Liste vor Variablen-Definitionen vom Typ Multiplexed stehen.
- **None:** Diese Variable wird in jeder Sendenachricht verwendet, unabhängig vom Multiplexer-Wert.
- **Multiplexed:** Die Variable wird nur gesendet, wenn der Wert **Multiplexer Value** dem aktuellen Multiplexer-Wert aus Mux-Val entspricht.

Information: Den Data-Variablen werden im Folgenden noch feste Testwerte für die Übertragung zugewiesen. Außerdem soll die Nachricht alle 200 ms gesendet werden. Dies geschieht in der gerätespezifischen Konfiguration (hier: Config0 I/O).

Aktion: Falls noch nicht vorhanden, dem PPCAN-Editor-Projekt eine neue Konfiguration mit **Edit > New Configuration** hinzufügen und dabei den Modultyp **PCAN-Router Pro** auswählen (siehe auch 4.1 Seite 13).

Im Kontextmenü (Rechtsklick) der CAN-Nachricht **Msg_100** den Eintrag **Add Symbol to Configuration** und dann **Config0 I/O** wählen.

Definition: In der Konfiguration Config0 müssen nun in den Fenstern **CAN Objects** als auch **Config0** Einstellungen für die CAN-Nachricht und die enthaltenen Variablen vorgenommen werden.

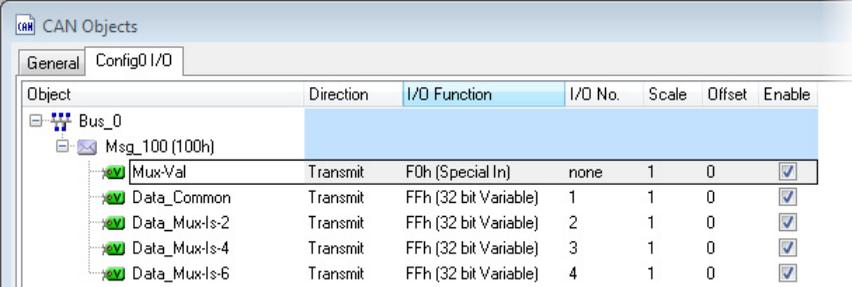


The screenshot shows the 'CAN Objects' configuration window. The 'General' tab is active, and the 'Config0 I/O' sub-tab is selected. The object hierarchy is as follows:

- Bus_0
 - Msg_100 (100h)
 - Mux-Val
 - Data_Common
 - Data_Mux-Is-2
 - Data_Mux-Is-4
 - Data_Mux-Is-6

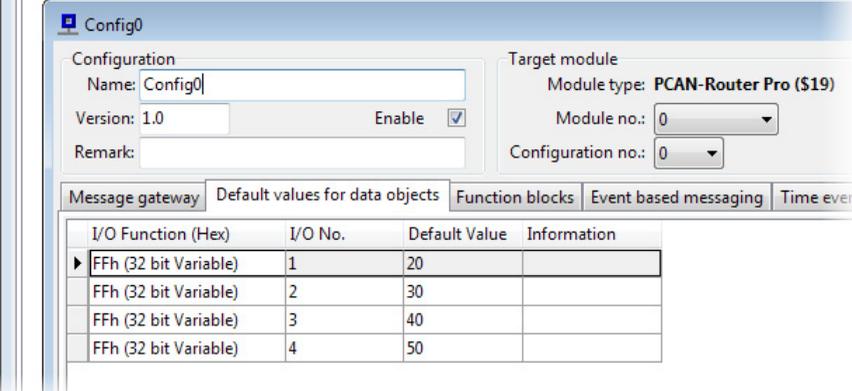
The table below shows the properties for the selected object 'Msg_100 (100h)':

Object	Direction	Enable	Period	Timeout	Event Time
Msg_100 (100h)	Transmit	<input checked="" type="checkbox"/>	200		0



The screenshot shows the 'CAN Objects' configuration window with the 'I/O Function' sub-tab selected. The table below shows the properties for the selected object 'Msg_100 (100h)':

Object	Direction	I/O Function	I/O No.	Scale	Offset	Enable
Mux-Val	Transmit	F0h (Special In)	none	1	0	<input checked="" type="checkbox"/>
Data_Common	Transmit	FFh (32 bit Variable)	1	1	0	<input checked="" type="checkbox"/>
Data_Mux-Is-2	Transmit	FFh (32 bit Variable)	2	1	0	<input checked="" type="checkbox"/>
Data_Mux-Is-4	Transmit	FFh (32 bit Variable)	3	1	0	<input checked="" type="checkbox"/>
Data_Mux-Is-6	Transmit	FFh (32 bit Variable)	4	1	0	<input checked="" type="checkbox"/>



The screenshot shows the 'Config0' configuration window. The 'Configuration' tab is active, and the 'Config0 I/O' sub-tab is selected. The configuration details are as follows:

Configuration Name: Config0
 Version: 1.0
 Enable:
 Remark:
 Target module: PCAN-Router Pro (\$19)
 Module type: PCAN-Router Pro (\$19)
 Module no.: 0
 Configuration no.: 0

The 'Message gateway' tab is selected, and the table below shows the default values for data objects:

I/O Function (Hex)	I/O No.	Default Value	Information
FFh (32 bit Variable)	1	20	
FFh (32 bit Variable)	2	30	
FFh (32 bit Variable)	3	40	
FFh (32 bit Variable)	4	50	

Bemerkung: Der Multiplexer-Variable Mux-Val ist die Ressource Special In (F0h)/none zugeordnet. Das heißt, dass sie automatisch mit jedem Sendevorgang der CAN-Nachricht (hier alle 200 ms) auf den Wert der nächsten Multiplexed-Variable gesetzt. Die Multiplexed-Variablen werden so der Reihe nach gesendet.

Information: Damit sind die Konfigurationsarbeiten dieser Aufgabe abgeschlossen und das Projekt kann auf den PCAN-Router Pro übertragen werden.

4.26 Aufgabe 9b: Multiplexer-Nachricht auf Anfrage senden

Information: Alternativ zum automatischen Durchiterieren der Multiplexed-Variablen kann eine andere Ressource verwendet werden, zum Beispiel eine 32-Bit-Variable, die per CAN beschrieben wird. So wird der Multiplexerwert von außen bestimmt.

Anwendungsmöglichkeit: Diverse Parameter sollen per CAN abgefragt werden, jedoch steht nur eine CAN-ID zum Senden zur Verfügung. Jedem Parameter wird einem Multiplexer-Wert zugeordnet. Der Multiplexer-Wert wird durch eine gesonderte Empfangsnachricht eingestellt und entsprechend als Antwort die Multiplexer-CAN-Nachricht gesendet.

In dieser Aufgabe soll der Empfang der CAN-Nachricht 1FFh (1 Byte) das Senden bereits vorhandenen Nachricht 100h auslösen. Dabei wird das Datenbyte von 1FFh als Multiplexerwert in 100h verwendet.

Bemerkung: Diese Aufgabe basiert auf der vorherigen 9a.

Aktion: Im Fenster **CAN Objects** unter **General** zusätzlich die CAN-Nachricht 1FFh mit diesen Eckdaten anlegen:

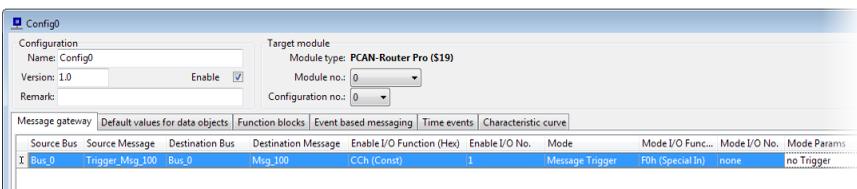
- └ CAN-ID: 1FFh (Trigger_Msg_100)
- └ Länge: 1 Byte
- └ Bit-Belegung (Variablen):

Byte Nr./ Startbit	Bits	Benennung	Verwendung
0/0	8	Request_MuxData	Wert für den Multiplexer in ID 100h

Unter **Config0 I/O** die Nachricht als Empfangsnachricht (**Receive**) einrichten und die Variable **Request_MuxData** der internen 32-Bit-Variablen 255 (I/O-Funktion FFh) zuweisen.

Ebenfalls unter **Config0 I/O** die bereits vorhandene CAN-Nachricht **Msg_100 (100h)** so abändern, dass sie nicht mehr regelmäßig gesendet wird. Dazu die Zykluszeit (**Period**) auf 0 setzen. Die Variable **Mux-Val** so abändern, dass sie ihren Wert nun aus der internen 32-Bit-Variablen 255 (I/O-Funktion FFh) erhält.

Wie in Aufgabe 3c (Seite 36) wird im **Message Gateway** der **Config0** ein neuer Eintrag angelegt, der mit dem Empfang der Nachricht **Trigger_Msg_100** das Senden der Nachricht **Msg_100** auslöst.



Information: Damit sind die Konfigurationsarbeiten dieser Aufgabe abgeschlossen und das Projekt kann auf den PCAN-Router Pro übertragen werden.