



CANape Training

Übungen mit dem ECU Simulator

V19.0 | 2022-04-12

ECU Simulator Kurzvorstellung



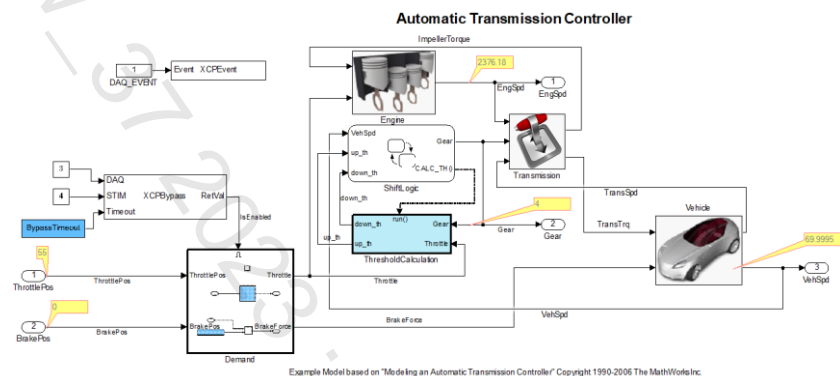
- ▶ Der ECU Simulator bildet mehrere kleine Applikationen nach:
 - ▶ *AutomaticTransmission*
 - ▶ *Wiper (Scheibenwischer)*
 - ▶ Nachbildung der Vector ECU Simulatoren: XCPsim / CANsim / ..

Überblick zur AutomaticTransmission Applikation

Die Demo-Applikation *AutomaticTransmission* steuert je nach Fahrsituation den optimalen Gang des Automatikgetriebes und berechnet dabei diverse Messgrößen.

Die Applikation wurde über ein Matlab Model realisiert.

- Eingangsgrößen: Gaspedalstellung und Bremspedalkraft
- Ausgangsgrößen: Fahrzeuggeschwindigkeit, Motordrehzahl und Gang



3/40

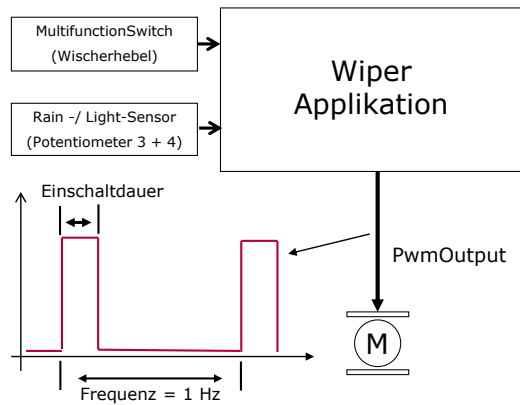
In dieser Applikation können die beiden Eingangsgrößen ThrottlePos (Gaspedalstellung) und BrakePos (Bremskraft) über die Schiebepotentiometer 1 (Accel) und 2 (Brake) im linken unteren Bereich der Simulator GUI stimuliert werden.



Überblick zur Wiper Applikation

Scheibenwischer Hebel Position:

- 1. OFF
- 2. LO: Langsamer Lauf
- 3. HI: Schneller Lauf
- 4. INT: Interval Modus
- 5. AUTO: Automatik-Modus mit Regensensor und Lichtsensor



4/40

Wie in einem realen Fahrzeug kann die Scheibenwischerapplikation in verschiedenen Modi genutzt werden. Insbesondere im Automatik-Modus (AUTO) ist die Scheibenwischergeschwindigkeit über die beiden Sensoren Rain und Light beeinflussbar. Innerhalb des Programm Codes existiert ein Kennfeld mit dem Namen *AutomaticCycleMap_m* über das die Charakteristik im Automatikmodus zur Laufzeit verstellt werden kann.

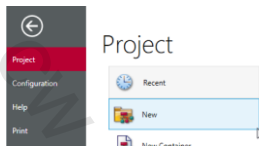


Regensensor

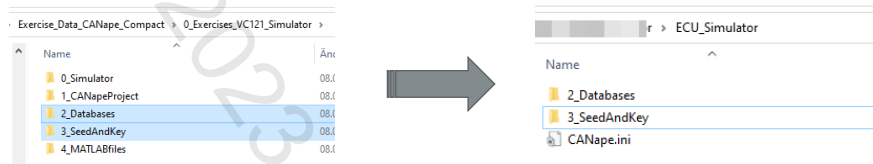
Licht

Übung 1a: Neues Projekt starten

1. Starten Sie über die CANape Menüleiste ein neues Projekt unter dem Namen *ECU_Simulator_Project* und richten Sie in dem Zuge einen Desktop Link ein.




2. Kopieren Sie vor dem Setup des Projektes zunächst alle relevanten Beschreibungsdateien zu dem simulierten Steuergerät in Ihr neues Projektverzeichnis. Dies erleichtert die potentielle Weitergabe des Projektes im Nachhinein erheblich.



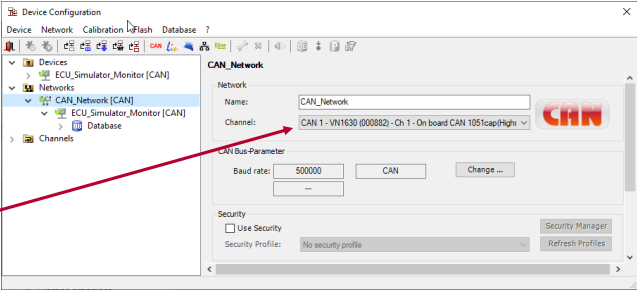
Übung 1b: Setup eines CAN-Monitors in der Gerätekonfiguration

3. Innerhalb des Projektes werden Sie mit Hilfe einer DBC-Datei CAN-Signale der Applikation messen und zusätzlich mit Hilfe einer A2L-Datei steuergeräteinterne Messgrößen auslesen. In diesem Übungsabschnitt starten Sie bitte zunächst lediglich mit dem Einrichten eines CAN-Monitor.

Start des Simulators (Kl. 15)



Der Simulator kann sowohl über den virtuellen CAN-Bus als auch über die Vector Hardware Interfaces mit einem Loop-Kabel angesprochen werden.

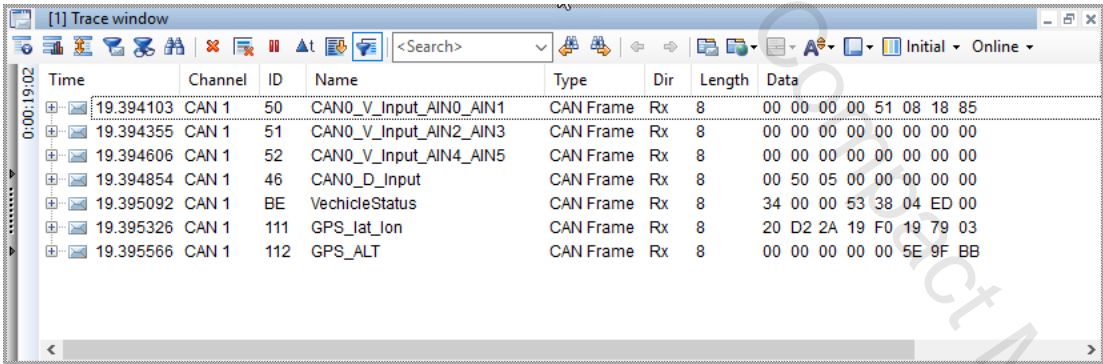


4. Nutzen Sie ein Trace Fenster um einen ersten Eindruck zu bekommen, ob ihre Gerätekonfiguration in Verbindung mit der Vector Hardware Konfiguration funktioniert.

Wenn Sie in ihrem Trace-Fenster diverse Botschaften angezeigt bekommen, dann haben Sie schon mal folgende Erkenntnisse:

- Richtige Verkabelung Ihres CAN-Netzwerkes inklusive der Abschlusswiderstände (120 Ohm)
- Richtige Einstellung der Baudrate ihres CAN-Netzwerkes in der Gerätekonfiguration von CANape
- Richtige Zuweisung der verwendeten Logischen Applikationskanälen von CANape in der Vector Hardwarekonfiguration

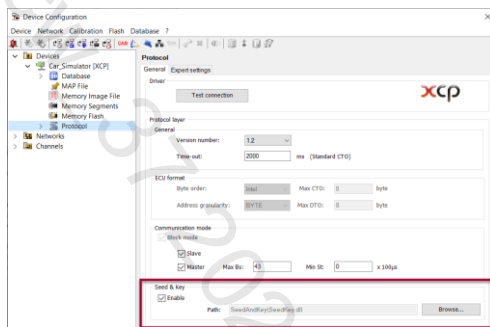
Ungewiss ist der Nachweis, ob Sie tatsächlich die richtige DBC-Datei für die Interpretation der empfangenen Nachrichten verwenden. Dies liegt in Ihrer Verantwortung.



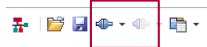
Time	Channel	ID	Name	Type	Dir	Length	Data
19.394103	CAN 1	50	CAN0_V_Input_AIN0_AIN1	CAN Frame	Rx	8	00 00 00 00 51 08 18 85
19.394355	CAN 1	51	CAN0_V_Input_AIN2_AIN3	CAN Frame	Rx	8	00 00 00 00 00 00 00 00
19.394606	CAN 1	52	CAN0_V_Input_AIN4_AIN5	CAN Frame	Rx	8	00 00 00 00 00 00 00 00
19.394854	CAN 1	46	CAN0_D_Input	CAN Frame	Rx	8	00 50 05 00 00 00 00 00
19.395092	CAN 1	BE	VehicleStatus	CAN Frame	Rx	8	34 00 00 53 38 04 ED 00
19.395326	CAN 1	111	GPS_lat_lon	CAN Frame	Rx	8	20 D2 2A 19 F0 19 79 03
19.395566	CAN 1	112	GPS_ALT	CAN Frame	Rx	8	00 00 00 00 00 5E 9F BB

Übung 1c: Setup eines XCP-Treibers in der Gerätekonfiguration

1. Starten Sie das Setup des XCP-Treibers mit der Funktion [Geräte | Neu aus Datenbasis] und nutzen folgende Dateien: Car.a2l , Car.hex , SeedKey.dll .
 - ▶ Wählen Sie beim Einrichten des Gerätes den Namen: *Car_Simulator*
2. Der Pfad auf die SeedKey.dll muss gegeben falls nach dem Setup nachkonfiguriert werden.



3. Wenn Sie den XCP-Treiber ohne Fehlermeldung zwischen dem Online-Zustand und Offline-Modus hin und her schalten können, waren Sie erfolgreich.



7/40

Beim Einrichten des „Neuen Gerätes“ erkennt CANape anhand der A2L-Datei, ob ein Seed&Key - Mechanismus zum Freischalten des XCP-Treibers notwendig ist. Der Datei-Name und der Pfad zu der Datei ist innerhalb der A2L-Datei angegeben. Da Sie in dieser Übung die A2L-Datei und die dazugehörige Seed&Key-Datei aus einem anderen Projektordner übernehmen, wird sich der relative Pfad der Seed&Key-Datei bezogen auf ihr Arbeitsverzeichnis unter Umständen ändern.

Falls CANape nach dem ersten Connect-Vorgang eine Fehlermeldung bezüglich der Seed&Key-Datei anzeigt, können Sie in der Gerätekonfiguration in den Protokoll – Einstellungen den Pfad neu angeben.

Übung 1d (optional): XCP-Treiber Settings sichten

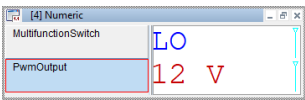
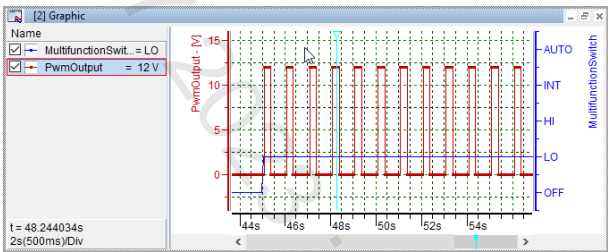
1. Untersuchen Sie die Treibereinstellungen des XCP-Treibers und vervollständigen die Angaben in den folgenden Zeilen:

- ▶ CAN-ID des XCP-Masters: _ _ _ _
- ▶ CAN-ID des XCP-Slave: _ _ _ _
- ▶ Anzahl der Messraster des XCP-Slave: _ _ _ _
- ▶ Unterstützt der Slave interne Zeitstempel? Ja/Nein?

Übung 2a: Erste Messung in der Scheibenwischer - Applikation

Die Eingangsgröße der Wiper Applikation mit dem Namen *MultifunctionSwitch* und das Ausgangssignal *PwmOutput* sind in der A2L-Datei in der Gruppe *Wiper/Wiper_Measurements* beschrieben.

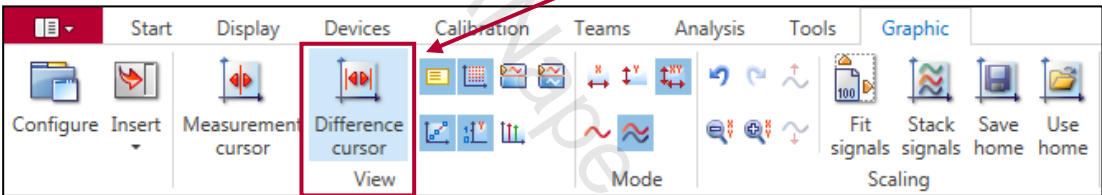
1. Konfigurieren Sie Ihre Messkonfiguration folgendermaßen:
 - > MultifunctionSwitch : Messraster: DAQ_100ms
 - > PwmOutput : Messraster: DAQ_20ms
2. Visualisieren Sie die Messgrößen in einem Grafik-Fenster und einem Ziffern-Fenster.
3. Speichern Sie ihre Messkonfiguration unter dem Namen: *MyMeasurement.cna*.
4. Analysieren Sie im Grafik-Fenster unter Zuhilfenahme der Differenzcursor die Einschaltdauer des PWM-Signals für die verschiedenen Arbeitsmodi (LO / HI / INT).



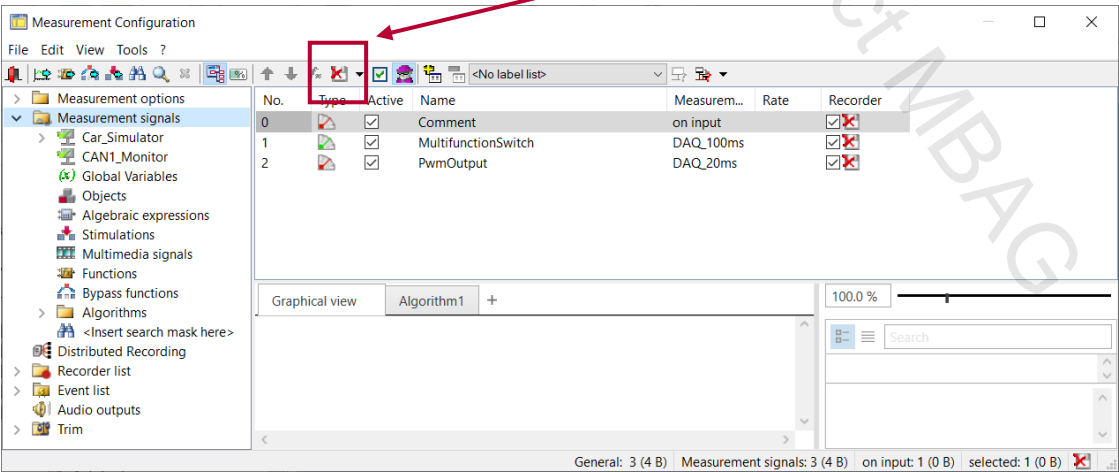
9/40

Wichtige Hinweise zur Lösung der Übung:

Für die Auswertung der Einschaltdauer werden Sie am Ende der Messung höchstwahrscheinlich in Zeitrichtung hinein zoomen wollen. Mit gedrückter linker Maustaste und einer horizontalen Mausbewegung können Sie dies erreichen. Zur Auswertung nutzen Sie die Differenzcursor. Einschalten können Sie diese über die gezeigte Menüleiste.



Während dieser Übung ist es nicht notwendig mit den Rekorden Messdateien zu erzeugen. Deaktivieren Sie von daher global sämtliche Rekorder über das gezeigte Icon in der Messkonfiguration.



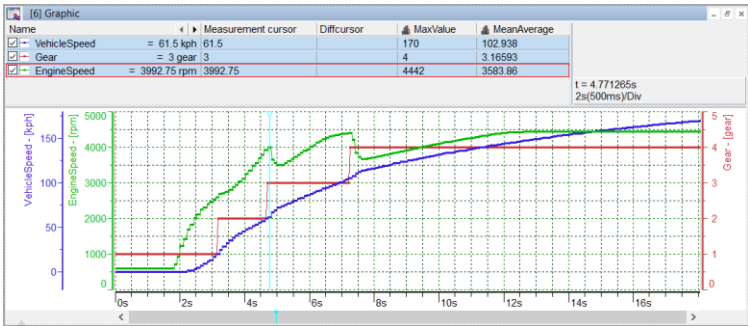
Übung 2b (optional): Grafik-Fenster mit Zusatzspalten

In dieser Übung dürfen Sie Beschleunigungstests mit der Transmission Applikation durchführen. Bremsen Sie später während ihrer Messung das Fahrzeug bis zum Stillstand ab um anschließend mit Vollgas die 100km/h – Marke zu erreichen. Ihre Aufgabe wird sein die Zeitdauer hierfür zu ermitteln und zwar erneut mit den Differenzcursor im Grafik-Fenster.

1. Erweitern Sie zuerst ihre Messkonfiguration mit den folgenden Signalen:

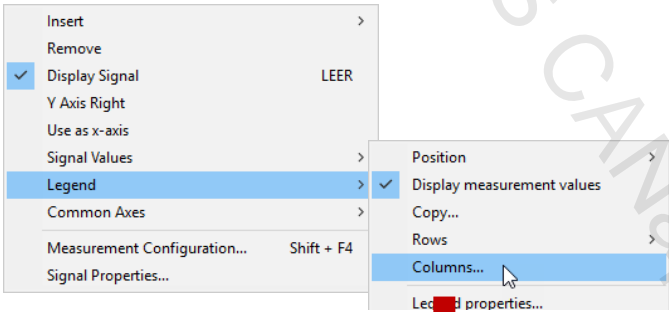
- > *EngineSpeed*, *Gear*, *VehicleSpeed*, *BrakeForce* und *ThrottlePos*
- > Hinweis: Alle Signale werden intern in einem 40ms Raster berechnet

2. Ergänzen Sie die Legende ihres Grafik-Fensters um die Spalten aus dem Screenshot (siehe Notizen)



10/40

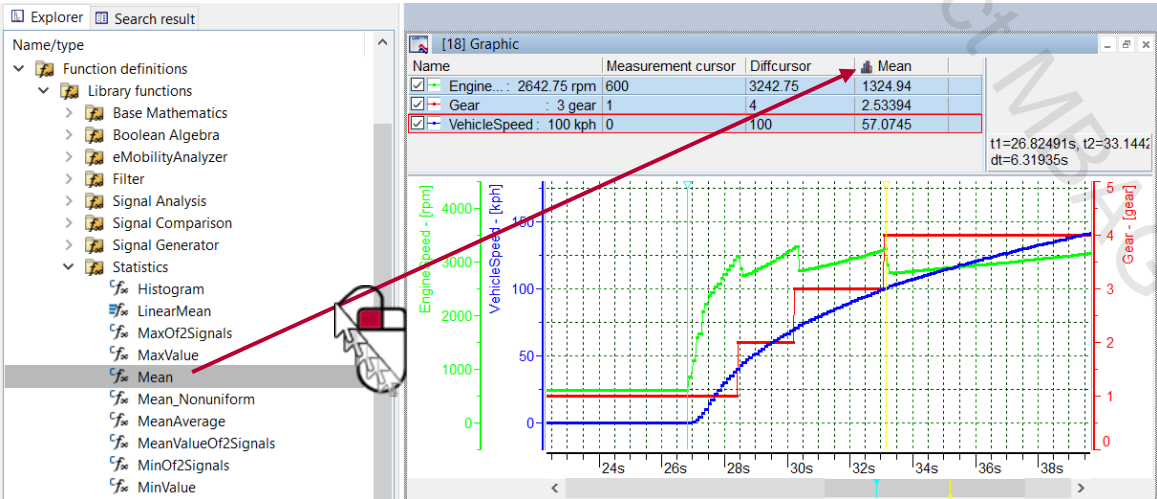
Kontextmenü des Grafik-Fensters



Column Selection

Active columns:			
Name	Exam...	Column width	Comment
<input checked="" type="checkbox"/> Name	Vehid...	216	Name (identifier) of the object
<input checked="" type="checkbox"/> Measurement cursor	131.5	127	Physical value of the object at ...
<input checked="" type="checkbox"/> Diffcursor		80	Physical value of the object at ...
<input checked="" type="checkbox"/> Max. value	144.25	120	The largest measured physical ...
<input checked="" type="checkbox"/> MeanAverage	80.3...	100	Statistics Column: Virtual.Mean...

Erstellen einer Statistik-Spalte

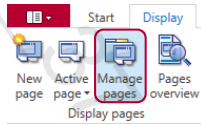


Übung 2c: Einrichten von Anzeigeseiten

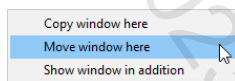
In den ersten beiden Übungen haben Sie bereits mehrere Anzeigefenster auf der CANape Oberfläche platziert. Damit die Übersichtlichkeit Ihrer Messungen in den nächsten Übungen nicht verloren geht, richten Sie sich in dieser Übung drei Anzeigeseiten ein.

Nennen Sie die erste Anzeigeseite *Transmission*, die zweite Anzeigeseite *Wiper* und die dritte Anzeigeseite *ECUsim*.

1. Nutzen Sie die in dem Screenshot angezeigte Funktion:



2. Die Anzeige-Fenster aus den letzten Übungen können Sie mit Drag&Drop auf die gewünschte Anzeigeseiten ziehen.



Übung 2d (optional): Label Listen

In den XCP-Grundlagen haben Sie gelernt, dass die Anzahl der Messsignale einerseits von der Bandbreite des Bussystems und andererseits durch den Speicherplatz für die DAQ-Listen (ECU Ressource) begrenzt ist. Aus diesem Grund kann es vorkommen, dass Sie in größeren Messkonfigurationen je nach Situation einzelne Signale vorübergehend deaktivieren/aktivieren sollten.

1. Erweitern Sie die Messkonfiguration um weitere Messsignale aus den beiden Applikationen *AutomaticTransmission* und *Wiper*.
2. Nutzen Sie die Möglichkeit mit Label-Listen ihre Messkonfiguration zu steuern. Erstellen Sie für die beiden Applikationen *Transmission* und *Wiper* je eine Label-Liste und testen diese. Sichten Sie dabei die hilfreichen Symbole, die Ihnen CANape bei den Signalnamen in der Legende der Anzeigefenster einfügt.

Hinweis: Auf der Notizenseite finden Sie eine kleine Anleitung für die Vorgehensweise.

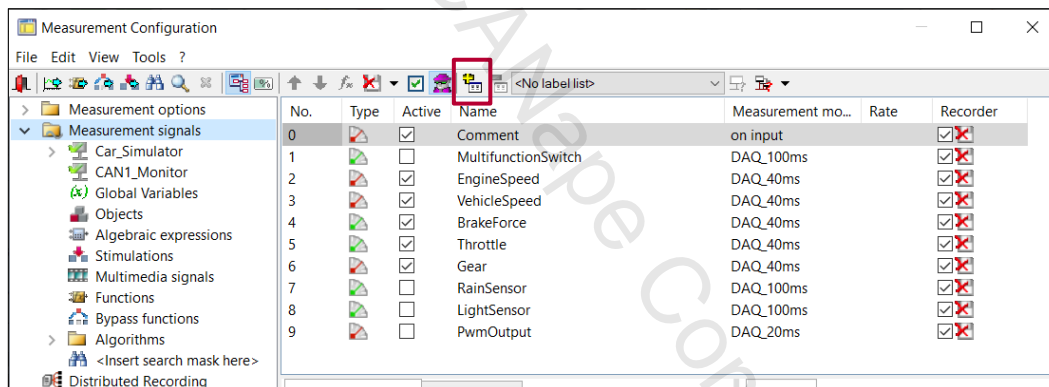
3. Testen Sie in einer neuen, leeren CANape Konfiguration die Import-Funktion von Label-Listen. Nutzen Sie die hierfür die Funktion [Datei | Messsignalliste importieren...] in der Messkonfiguration.

12/48

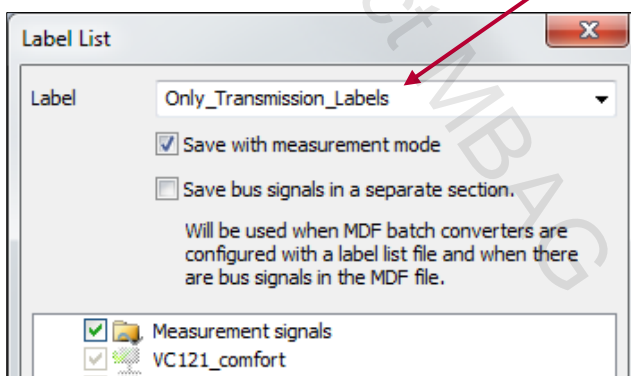
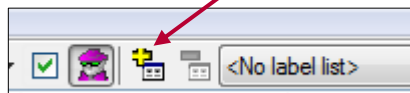
Zur Erstellung einer Labelliste gehen Sie folgendermaßen vor:

Öffnen Sie die Messkonfiguration (Shortcut <F4>)

Aktivieren bzw. deaktivieren Sie ihre gewünschten Messsignale über die Spalte „Aktiv“ (Active).



Nutzen Sie nun folgendes Icon aus dem obigen Screenshot und vergeben anschließend ein Namen für diese Liste

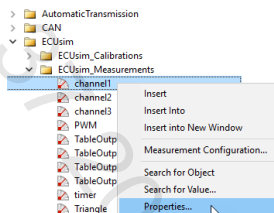


Übung 3a - Part 1: Y-Achsen Skalierung bei Messungsstart

In dieser Übung werden Sie die Y-Achsen-Skalierungen ihrer Messsignale neu konfigurieren. Die ursprünglichen Konfigurationseinstellungen, die CANape aus den Beschreibungsdateien (A2L / DBC) entnommen hatte, werden dabei überschrieben.

1. Erweitern Sie Ihre CANape Konfiguration um folgende Signale:

- ▶ *channel1*, *channel2* und *channel3* aus der Gruppe *SIGGen_Measurements*
- ▶ Über den Symbol-Explorer können Sie sich einen Überblick über die Eigenschaften der Signale verschaffen, z.B. alternative Messraster, Default-Skalierung, ...

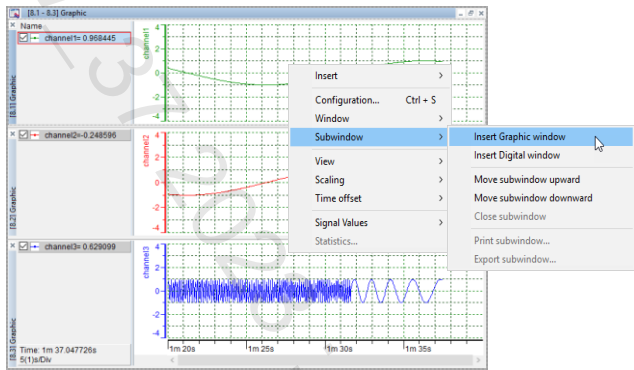


2. Testen Sie zunächst einmal die Shortcuts F, E, D, B in Verbindung mit den Icons



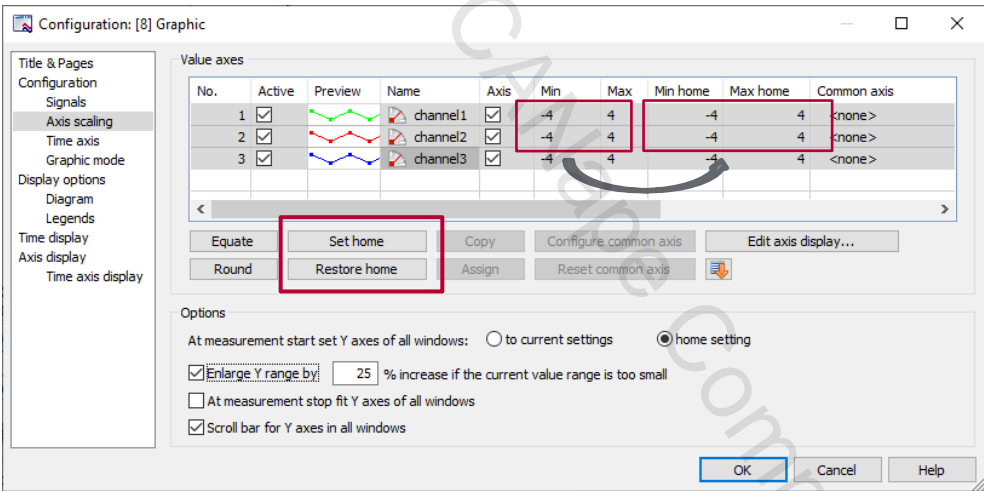
Übung 3a - Part 2: Y-Achsen Skalierung bei Messungsstart

- 3. Konfigurieren Sie nun manuell über das Kontextmenü des Grafik-Fenster Skalierungen von +/- 4 Volt und speichern diese als Grundeinstellung für den Messungsstart ab.
 - ▶ Kontrollieren Sie in den Messoptionen, ob CANape bei Messungsstart diese Grundeinstellungen übernimmt.
- 4. Nutzen Sie abschließend die Teilfenster im Grafik-Fenster um die Signale einzeln darzustellen.



14/48

Die Y-Achsenkalierung im Grafik-Fenster kann u.a. über die Funktion [Konfiguration] im Kontextmenü eingestellt und anschließend als Grundeinstellung gespeichert werden.



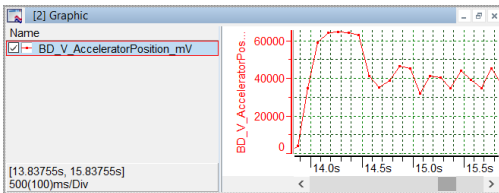
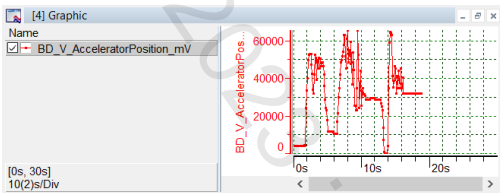
Übung 3b: Zeit-Achsen Skalierung bei Messungsstart

Die Grafik-Fenster in CANape können während der Messung benutzerdefinierte Zeitdauern anzeigen. Diese Periodendauern können im Grafik-Fenster über den Shortcut <t> konfiguriert werden.

1. Suchen Sie über den Symbol-Explorer das CAN-Signal ..AccelerationPosition_mV und ziehen dies in zwei verschiedene Grafik-Fenster. Richten Sie sich hierfür zuerst eine neue Anzeigeseite mit dem Namen CAN ein. Das CAN-Signal wird mit der Botschaft BD_V_Sensors_0_1 übertragen und kann über das Potentiometer 1 (Accel) im Wert verändert werden.

► Frage: Müssen Sie in der Messkonfiguration bezüglich der Messraster aktiv werden?

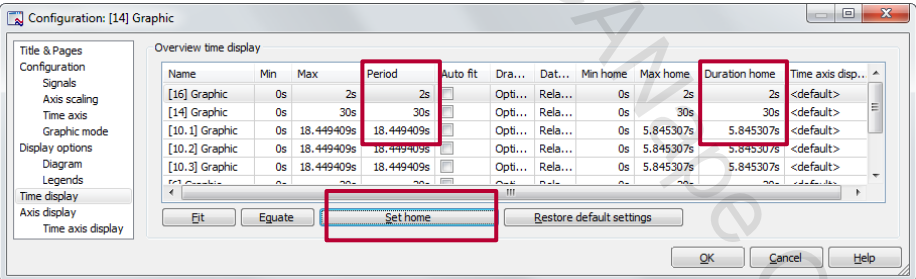
2. Konfigurieren Sie für ein Grafik-Fenster einen Zeitbereich/Periode von 30sec und für das zweite Grafik-Fenster eine Periode von 2sec.
3. Am Ende der Messung soll der Zeitbereich nicht automatisch eingepasst werden.



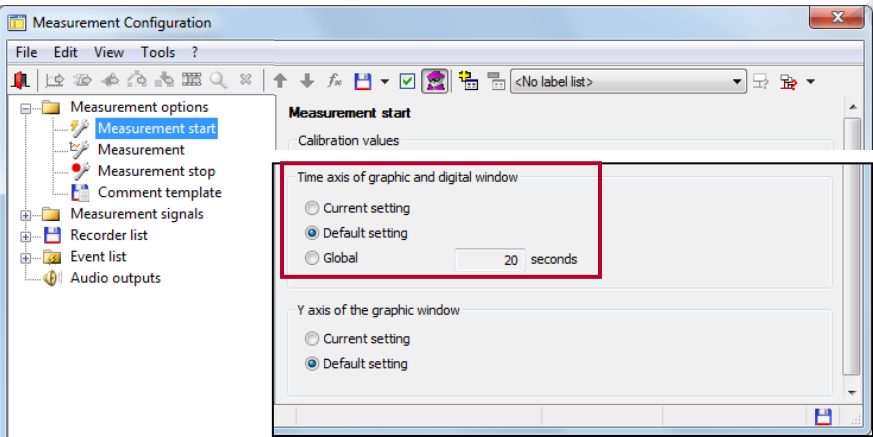
15/48

Wichtige Hinweise zur Lösung der Übung:

Die gewünschten Zeitbereiche von 30sec bzw. 2sec können über die Funktion [Zeitdarstellung..] im Kontextmenü der Grafikfenster eingestellt werden.

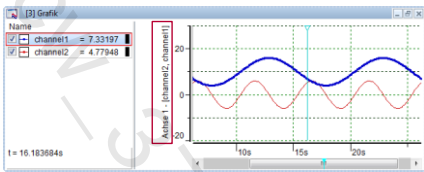


Beachten Sie die Konfiguration in den Messoptionen bezüglich der Zeitachsenskalierung bei Messungsstart.



Übung 3c (optional): Gemeinschaftsachsen im Grafik-Fenster

- 1. Fügen Sie die Signale *channel1* und *channel2* nacheinander über den Symbol-Explorer in ein neues Grafik-Fenster ein. Beim Einfügen des zweiten Signal sollten Sie das Signal auf der Y-Achse des zuerst eingefügten Signals fallen lassen. CANape wird bei dieser Vorgehensweise automatisch für beide Signale eine Gemeinschaftsachse einrichten.



- 2. Modifizieren Sie die Gemeinschaftsachse über den Dialog der Funktion [Konfiguration]. Verändern Sie die Farbe und den Namen der Achse.

Value axes									
No.	Act...	Preview	Name	Axis	Min	Max	Min home	Max home	Common axis
1	<input checked="" type="checkbox"/>		channel1	<input checked="" type="checkbox"/>	-20	30	-1.175e+38	3.403e+38	Achse 1
2	<input checked="" type="checkbox"/>		channel2	<input checked="" type="checkbox"/>	-20	30	-1.175e+38	3.403e+38	Achse 1

Equale

Set home

Copy

Configure common axis

Edit axis display...

Round

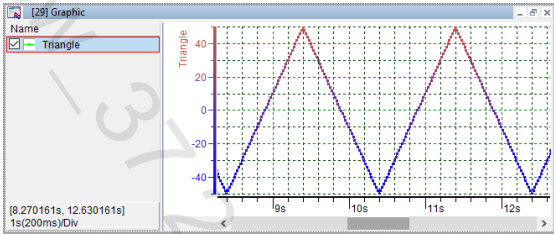
Restore home

Assign

Reset common axis

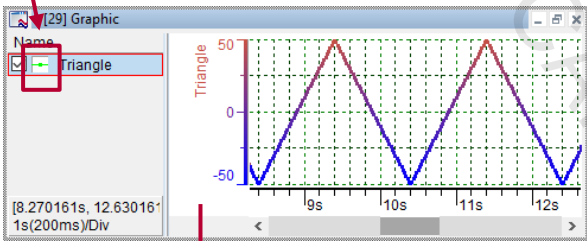
Übung 4a (optional): Farbfunktionen für die Darstellung von Messwerten

1. Visualisieren Sie in dieser Übung das Signal *Triangle* aus der SWC *ECUsim* in einem Grafik-Fenster und benutzen eine Farbfunktion vom Typ *Gradient*. Im Ergebnis sollte Ihr Kurvenverlauf so aussehen, wie in dem Screenshot angezeigt. Zur Umsetzung dieser Übung entnehmen Sie bitte die Hinweise aus der Notizenseite.



17/48

Farbfunktionen können am schnellsten per Doppelklick auf das farbige Icon in der Legende angelegt werden.



Display Options

Line style: Line

Thickness of line: 1

Marking: Point

Type of: Connect

Color function...

OK Cancel Help

Color functions

Display Color

Global color function (active)

Local color function

Global Color Function

Color Function for Object 'Triangle'

☐ No color function

☒ Select color function

Triangle_Color_Function New... Edit...

Type: Gradient

Comment:

☐ Determine color in dependency on the time

☐ Object

Select...

OK Abbrechen Hilfe

Edit color function 'Triangle_Color'

General

Gradient

Start: -50

Final: 50

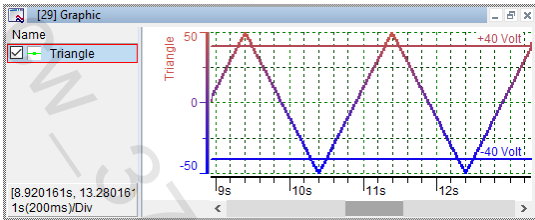
Preview:

OK Abbrechen Hilfe

Im Beispiel wurde der Typ of color „Gradient“ genutzt.

Übung 4b (optional): Achsen mit Hilfslinien für das Grafik-Fenster

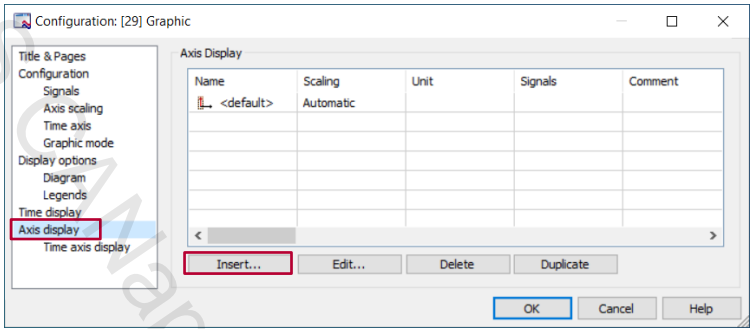
- 1. Konfigurieren Sie eine neue Achsendarstellung für das Signal *Triangle* im Grafik-Fenster. Siehe dazu die Hinweise in der Notizenseite.



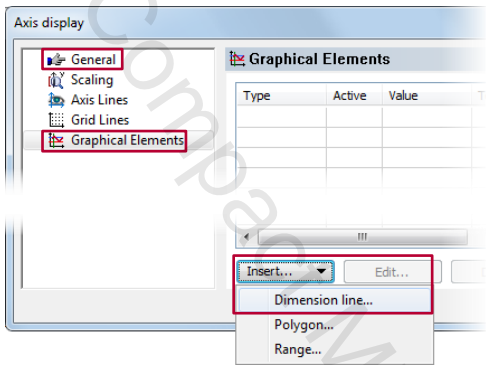
Hilfslinien gehören zur Achse

18/40

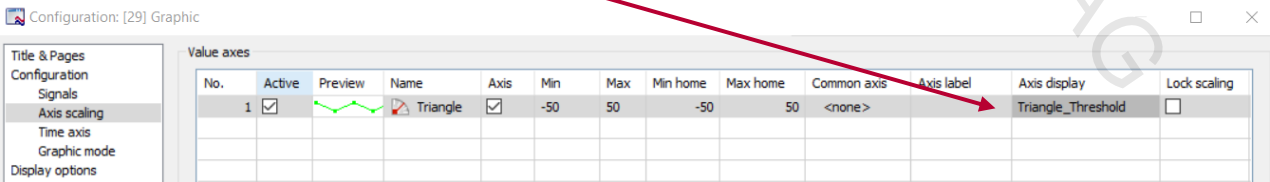
- Öffnen Sie das Konfigurationsmenü des Grafik Fensters.
- Wählen Sie Achsendarstellungen und dann einfügen



- Vergeben Sie unter Allgemeines einen Namen und fügen Sie unter Grafische Elemente eine neue Maßlinie ein.



- Weisen Sie unter Achsenskalierung dem Signal die neue Achsendarstellung zu.

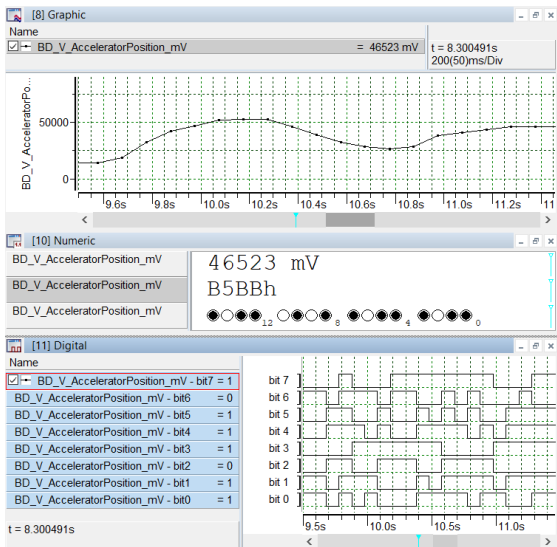
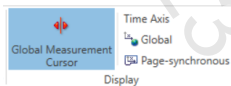


Übung 4d: Ziffern-Fenster und Digital-Fenster

In dieser Übung sammeln Sie Erfahrungen mit dem Ziffern- und Digital-Fenster.

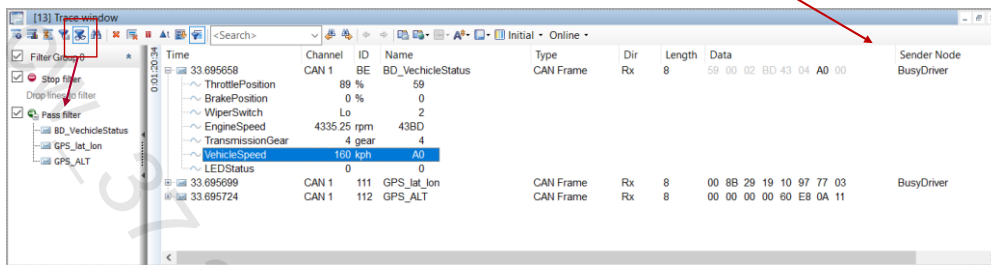
- 1. Stellen Sie das CAN-Signal mit dem Namen *...AcceleratorPosition_mV* in unterschiedlichen Wertedarstellungen im Ziffern-Fenster dar: Neben dem physikalischen Wert soll der Signalwert als Rohwert (dez oder hex) und in binärer Form angezeigt werden.
- 2. Fügen Sie das CAN-Signal auch in ein Digital-Fenster ein und lassen dort lediglich die untersten 8 Bit anzeigen.

Hinweis: der globale Messmarker ist nun hilfreich.



Übung 5a – Part 1: Trace – Fenster

1. Überprüfen Sie zunächst welche Botschaften (CAN-IDs) in dem Netzwerk aktuell versendet werden.
2. Konfigurieren Sie eine weitere Spalte zur Anzeige der Sendeknoten

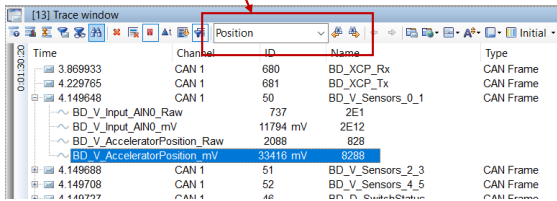


3. Nutzen Sie Analysefilter vom Typ *Pass filter*, wie im Screenshot angezeigt
4. Werden alle Botschaften aus der DBC versendet? Werfen Sie hierfür einen Blick in die Datenbasis
5. Welche Zykluszeit können Sie bei der ID 0xBE messen?

Übung 5a – Part 2: Trace – Fenster

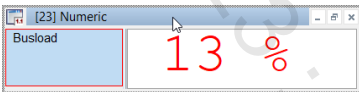
7. Welchen Wertebereich durchläuft das Signal ..AccelerationPosition_mV aus der Botschaft mit der ID 0x50, wenn Sie den Schieberegler 1 (Accel) betätigen? Nutzen Sie die Suchfunktion im Trace-Fenster um das Signal zu finden.

- ▶ Raw value: _ _ _ _ _ - _ _ _ _ _
- ▶ Physical value: _ _ _ (min) - _ _ _ (max)



8. Ziehen Sie das Signal ..AccelerationPosition_mV in ein Grafik-Fenster und überprüfen dort erneut den Wertebereich des Signals. Nutzen Sie den Globalen Cursor bei dieser Übung.

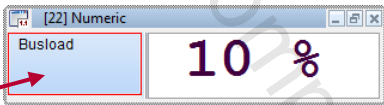
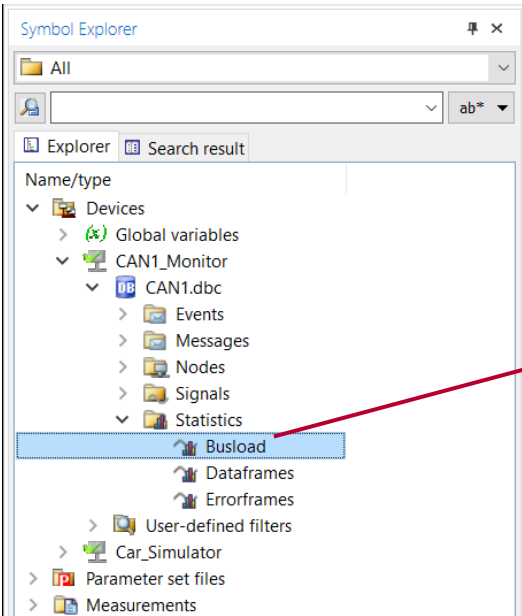
9. Zeigen Sie in einem Ziffern-Fenster die aktuelle Buslast ihres CAN-Netzwerkes an. Entnehmen Sie hierfür die Hinweise aus der Notizenseite.



21/40

Es existiert in CANape kein Standard - Busstatistikfenster wie in den Vector Tools CANalyzer und CANoe. Über einen kleinen Umweg ist die Buslast dennoch messbar.

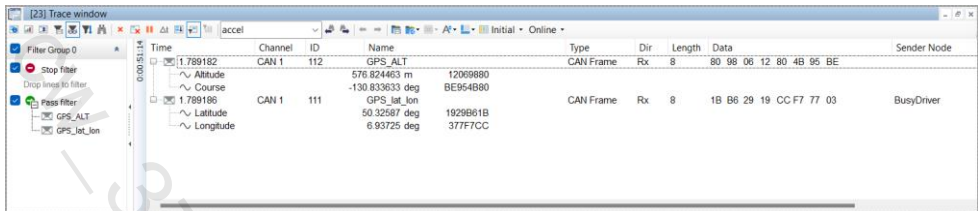
Sobald in der Gerätekonfiguration ein CAN-Treiber eingerichtet ist, kann über diesen Geräteeintrag die Buslast erfasst werden. Allerdings funktioniert dies nur bei einem realen CAN-Netzwerk und leider nicht bei dem Virtuellen CAN-Netzwerk.



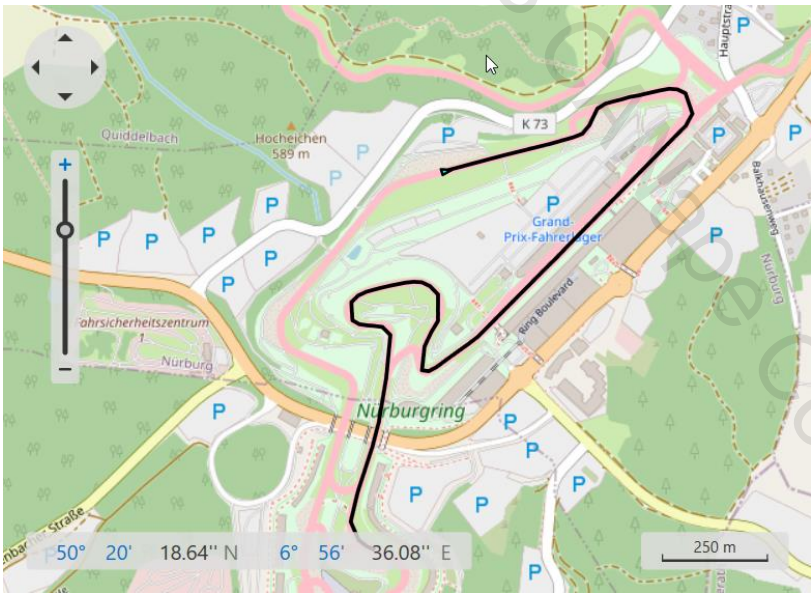
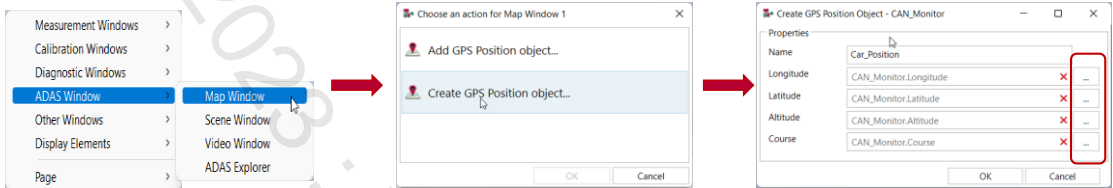
Übung 5b (optional): ADAS-Fenster vom Typ Map - Fenster

Der Simulator versendet zwei CAN-Botschaften mit GPS-Daten, die Sie nun analysieren dürfen.

1. Filtern Sie die beiden Botschaften mit einem Analysefilter heraus, wie im Screenshot angezeigt.



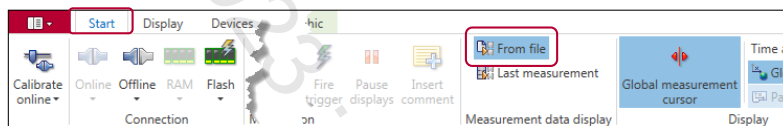
2. Platzieren Sie nun ein Map-Fenster in dem die Signale Latitude, Longitude, Altitude und Course aus den beiden Botschaften angezeigt werden um die Fahrstrecke zu analysieren.



Übung 6a: MDF-Dateien speichern mit den Rekordern

Während dieser Übung sollen Sie mehrfach die Fahrzeuggeschwindigkeit mit dem Poti1 (Accel) erhöhen und mit dem Poti2 (Brake) reduzieren. Dabei sind die Gangwechsel bzw. die Schaltzeitpunkte von Interesse.

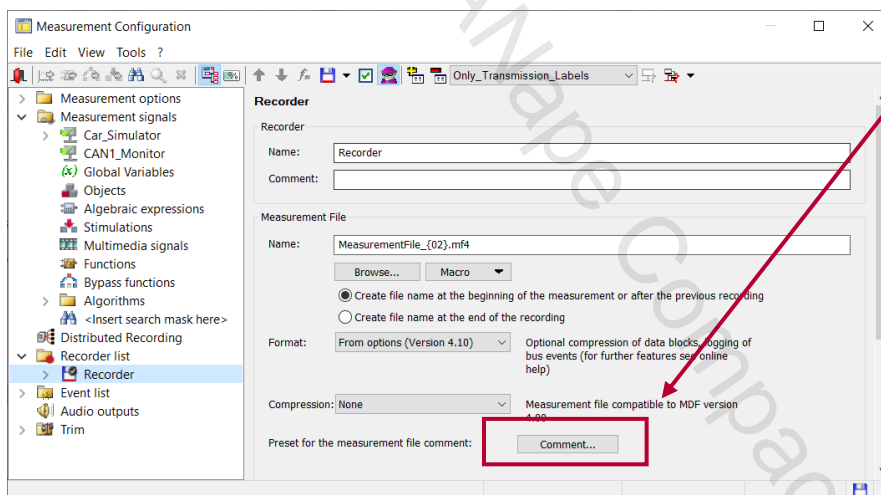
- (1) Speichern Sie mit einem Rekorder mehrere Messdateien ab, wobei CANape den Messdateinamen unter Verwendung eines Makros automatisch verändert.
 - a) Speichern Sie lediglich die Messsignale der Applikation *Transmission*.
 - b) Richten Sie sich zum Starten der Speicherung einen Trigger ein, der bei dem Gangwechsel vom zweiten in den dritten Gang auslöst.
 - c) Speichern Sie eine Vorlaufzeit von 4sec und eine Nachlaufzeit von 6sec.
 - d) Speichern Sie die Trigger-Zeitfenster jeweils in separate Dateien.
 - e) Richten Sie ihre Kommentarvorlage so ein, dass in sämtlichen Trigger-Messdateien ein Default-Text ihrer Wahl eingetragen wird.
- (2) Zur Sichtung Ihrer MDF-Dateien verwenden Sie die Funktion [From File]



23/40

Wichtige Hinweise zur Lösung der Übung:

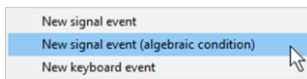
Teilaufgabe 1e: Die Default Kommentare für die Messdatei müssen vorab über den Kommentardialog definiert werden.



Übung 6b (optional): Trigger mit algebraischer Bedingung

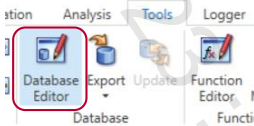
In der letzten Übung wurde lediglich ein einzelnes Signal als Trigger-Eingangsgröße beobachtet. In dieser Übung soll eine Trigger-Bedingung formuliert werden, bei der zwei Eingangsgrößen überprüft werden.

- (1) Formulieren Sie eine algebraische Bedingung bei der der Schaltvorgang vom zweiten in den dritten Gang detektiert wird, aber zudem sichergestellt ist, dass die Bremskraft unter 25 N liegt.



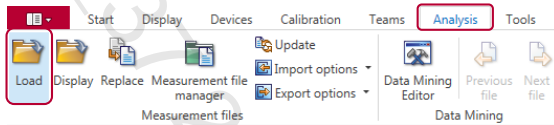
Testen Sie ihre neue Trigger-Bedingung und achten darauf, dass die Trigger-Bedingung aus der vorigen Übung deaktiviert ist.

Hinweis: In der Car.a2l Datei scheint eine falsche Einheit bei dem BrakeForce Signal hinterlegt zu sein. Normalerweise sind Änderungen in der A2L-Datei nur den Software-Ingenieuren vorbehalten, aber ändern Sie ausnahmsweise die Einheit auf Newton [N].



Übung 6c (optional): Logging Datei im BLF-Format

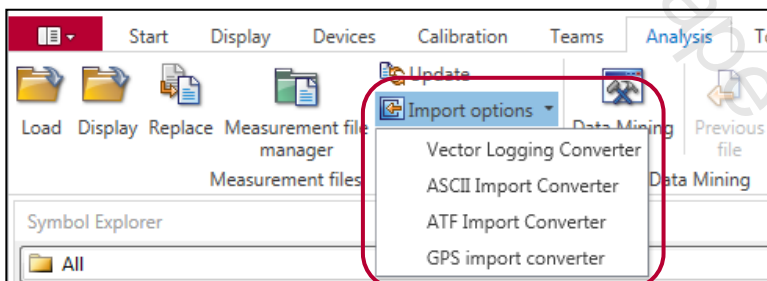
- (1) Erweitern Sie ihre Rekorderliste um einen Bus Logging-Rekorder, der im CANalyzer / CANoe typischen BLF-Format die Messdaten abspeichert.
 - a) Loggen Sie mit diesem Rekorder den kompletten CAN-Bus mit.
- (2) Nach der Aufzeichnungsphase laden Sie die BLF-Datei in den Symbol Explorer und fügen dann das Signal „...AcceleratorPosition_mv“ in ein Offline-Auswertefenster ein. Zum Laden der Datei nutzen Sie die markierte Funktion aus dem Screenshot.



25/40

Da CANape grundsätzlich auf Basis von physikalischen Signalen arbeitet, müssen die Rohdaten der CANalyzer/CANoe typischen Logging-Dateien in das signalorientierte MDF-Format konvertiert werden. Die Konvertierung wird automatisch in CANape angeregt, sobald eine BLF-Datei zur Sichtung ausgewählt wird.

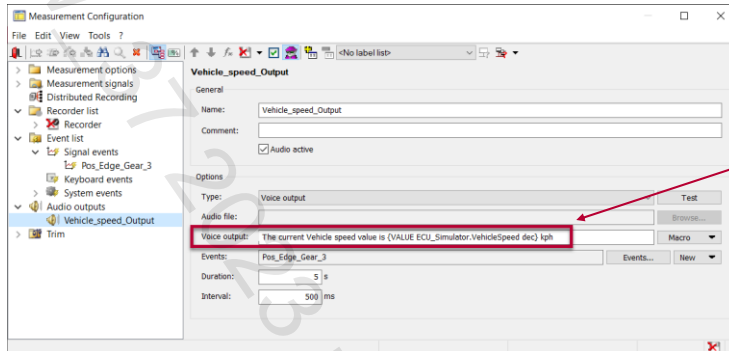
Das BLF-Format unterstützt die Möglichkeit einen Pfad auf eine DBC-Datei mitzuspeichern. Ist dies der Fall wird CANape bei dem Konvertierungsprozess automatisch darauf zurück greifen. Ist der Pfad auf die passende DBC-Datei nicht hinterlegt, dann muss manuell in den Optionen des Vector Logging Konverters eine Kommunikations-Datenbasis hinterlegt werden.



Übung 6d (optional): Sprachausgabe in CANape

In der Übung 6a+b haben Sie Signalereignisse kennengelernt mit deren Hilfe Sie die Speicherung der MDF-Dateien gestartet haben. Die gleichen Ereignisse können Sie nutzen um eine Sprachausgabe zu steuern.

1. Konfigurieren Sie eine neue Audioausgabe innerhalb der Messkonfiguration, wobei Sie eines der Signalereignisse aus Übung 6a+b wiederverwenden. Verwenden Sie den Typ Sprachausgabe und nutzen hierin ein Makro, welches Ihnen einen Signalwert ausgibt. Siehe Screenshot.



Signalwert
Ausgabe über
Makro

Übung 7a: Messdateien laden und vergleichen mit Hilfe von Zeitoffsets

1. Stellen Sie das Signal *VehicleSpeed* aus beiden Messdateien **Acceleration_StandingStart_00/01.MDF** im selben Grafik Fenster dar und vergleichen Sie die beiden Kurven
 - ▶ Startet die Beschleunigung in beiden Dateien zur gleichen Zeit? Falls nicht, verschieben Sie eine Datei, um die Kurven vergleichen zu können.
 - ▶ Welche Messdatei enthält die schnellere Beschleunigung?
 - ▶ Welche Messdatei hat die höhere Endgeschwindigkeit? Nutzen Sie hierfür gerne die Funktion [Signalwerte | Wert suchen..] aus dem Kontextmenü des Grafik-Fensters.
 - ▶ Kommentieren Sie den höchsten Wert aus den beiden Dateien mit Hilfe der Funktion [Signalwerte | Kommentieren..] aus dem Kontextmenü des Grafik-Fensters.
2. Vergleichen Sie das *VehicleSpeed* Signal der Dateien **Acceleration_TriggeredFile_01/-03.MDF** in einem weiteren Grafik Fenster.
 - ▶ Richten Sie die Messdateien am Trigger aus.

Übung 7b (optional): Druckfunktionen in CANape

In CANape können die Anzeigeseiten direkt in einer Druckansicht dargestellt werden und wenn gewünscht komplett mit Fuß- und Kopfzeile ausgedruckt oder als PDF-Datei gespeichert werden. Sie können aber jederzeit beliebige Fenster zur Dokumentation in andere Windows Programme ziehen.

1. Ziehen Sie ein beliebiges Anzeigefenster aus der CANape Oberfläche auf den Desktop und lassen es dort fallen. CANape legt Ihnen dieses Fenster als Bild-Datei ab.
2. Ziehen Sie ein Anzeigefenster in ein Word-Dokument und sichten das Ergebnis.

Übung 8: Data Mining mit algebraischer Methode

Für die Data Mining Übung kann es hilfreich sein sämtliche geladene Messdateien aus anderen Analyseaufgaben zu schließen. Vor diesem Hintergrund beginnen Sie für diese Übung ein neues Projekt und starten dabei mit einer neuen Konfiguration.

1. Laden Sie aus dem Verzeichnis `..\Measure` die Messdatei mit dem Namen `Analog_Digital_Input_File_00.MDF` und fügen die beiden angegebenen Signale aus Teilaufgabe (2) in ein Grafik Fenster ein.
2. Konfigurieren Sie eine Methode, die überprüft, ob die beiden folgenden digitalen Signale gleichzeitig den Wert 1 aufzeigten.
 - ▶ `CAN0_D_Input_DIN1_Dig`
 - ▶ `CAN0_D_Input_DIN2_Dig`

Durchsuchen Sie nicht nur die geladene Messdatei, sondern alle Messdateien aus der Serie `(..Input_File_xx)`.

3. Prüfen Sie, wie oft die beiden folgenden Analog-Eingänge gleichzeitig über 1000mV / 2000mV waren. Hierfür lassen Sie ihre algebraische Methode zweimal laufen und ändern dabei den Schwellwert.
 - ▶ `CAN0_V_Input_AIN1_mV`
 - ▶ `CAN0_V_Input_AIN2_mV`

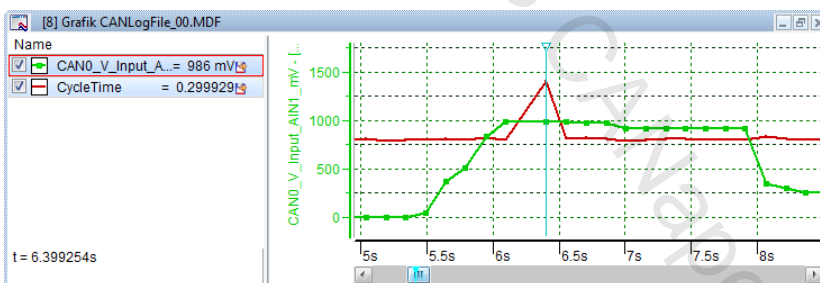
Übung 9a: Virtuelle Messdateikanäle für die Offline-Analyse

In dem Verzeichnis `..\Measure` sind mehrere Messdateien mit den Namen **CANLogFile_xx.MDF** abgespeichert. In diesen Dateien sind Auffälligkeiten bezüglich der Sendeabstände der gemessenen CAN-Signale zu finden.

1. Laden Sie von der Messdatei mit den Namen **CANLogFile_00.MDF** das CAN-Signal `AIN1_mV` in ein Grafik-Fenster.
2. Analysieren Sie mit Hilfe der Funktion `CycleTime` aus der CANape-Funktionsbibliothek den Verlauf des Sendeabstandes des CAN-Signals. Welche maximale Sendeabstände sind in dieser Messdatei festzustellen?
3. Ersetzen Sie nun die Messdatei **CANLogFile_00.MDF** durch die Datei **CANLogFile_01.MDF** und beobachten dabei das Ergebnis der Funktion `CycleTime`.

30/40

Das Ergebnis Ihrer Übung sollte in etwa so aussehen, wie es in dem folgenden Screenshot dargestellt ist.



Übung 9b (optional): Kaskadierung von Funktionen

Stellen Sie die beiden Signale *Speed* und *GPS_speed* aus der Datei `..\Measure\VideoGps.mdf` in einem Grafik-Fenster dar.

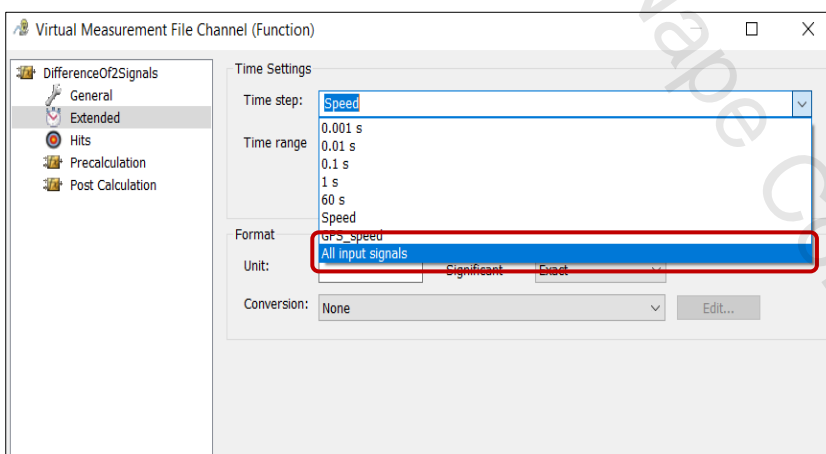
1. Berechnen Sie die Differenz zwischen den beiden Geschwindigkeitssignalen mit der Funktion *DifferenceOf2Signals*.
 - ▶ Stellen Sie sicher, dass für jeden Signalwert der beiden Eingangssignale die Differenz berechnet wird. (Hinweise zum Zeitraster der Berechnung der Funktion siehe Notizenseite)
2. Wenden Sie die Betragsfunktion *AbsoluteValue* auf das berechnete Differenzsignal an.
3. Optional: Stellen Sie die beiden berechneten virtuellen Messdateikanäle in einem Teilfenster dar. Erzeugen Sie in den Teilfenstern jeweils eine sinnvolle Gemeinsame Achse.

31/40

Wichtiger Hinweis zur Lösung der Teilaufgabe 1:

Das Zeitraster in welchem eine Funktion aufgerufen wird, kann im Konfigurationsdialog des virtuellen Messdateikanal im Reiter [Erweitert] definiert.

Soll eine Funktion für jeden neuen verfügbaren Wert aller Eingangssignale berechnet werden, so muss als Zeitraster [Alle Eingangssignale] gewählt werden.



Übung 9c: Data Mining mit einer Funktion als Suchmethode

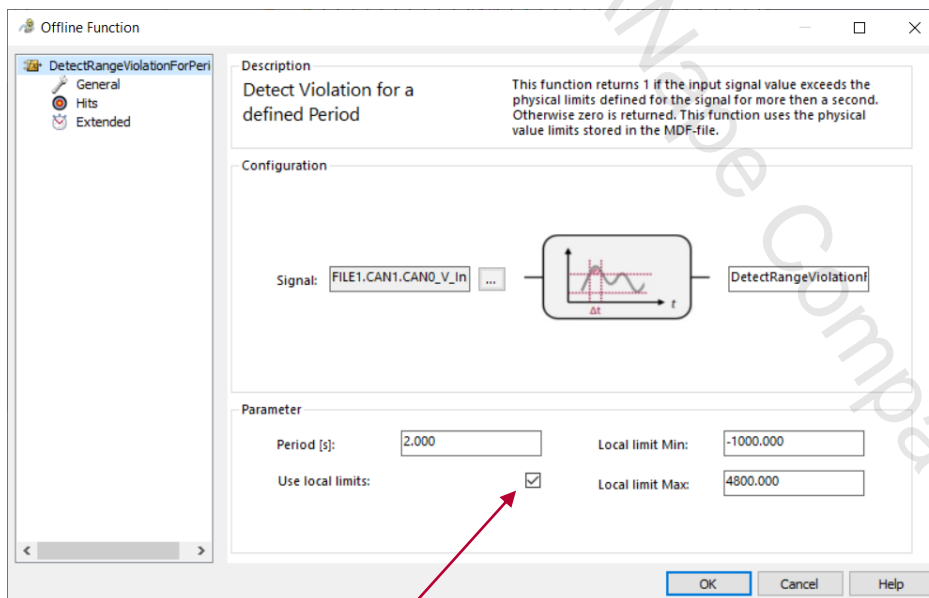
In dem Verzeichnis `..\Measure` sind mehrere Messdateien mit den Namen **CANLogFile_xx.MDF** abgespeichert. Es soll untersucht werden, ob für das CAN-Signal `...CAN0_V_Input_AIN1_mV` in diesen Dateien ein kritischer Schwellwert von 4,8 Volt für eine Zeitdauer von 2 sec überschritten wurde.

1. Konfigurieren Sie auch hierfür eine neue Konfiguration mit einem Namen Ihrer Wahl und schließen gegeben falls sämtliche Messdateien im Symbol Explorer.
2. Laden Sie von der Messdatei mit dem Namen **CANLogFile_00.MDF** das CAN-Signal `...CAN0_V_Input_AIN1_mV` in ein Grafik-Fenster.
3. Analysieren Sie das angezeigte Signal mit der Bibliotheksfunktion *DetectRangeViolationForPeriod*.
Hinweis: Die Funktion gibt eine 1 zurück, sobald das Signal für eine definierte Zeitdauer eine kritische Obergrenze über- bzw. eine Untergrenze unterschritten hat.
 - Konfigurieren Sie die lokalen Parameter der Funktion gemäß den Vorgaben. (Hinweise zu den lokalen Parametern siehe Notizenseite)
4. Nutzen Sie die Funktion als Prüfmethode für ihre nächste Data Mining Konfiguration, um alle **CANLogFile_xx.MDF** Messdateien zu durchsuchen.

32/40

Wichtiger Hinweis zur Lösung der Teilaufgabe 3:

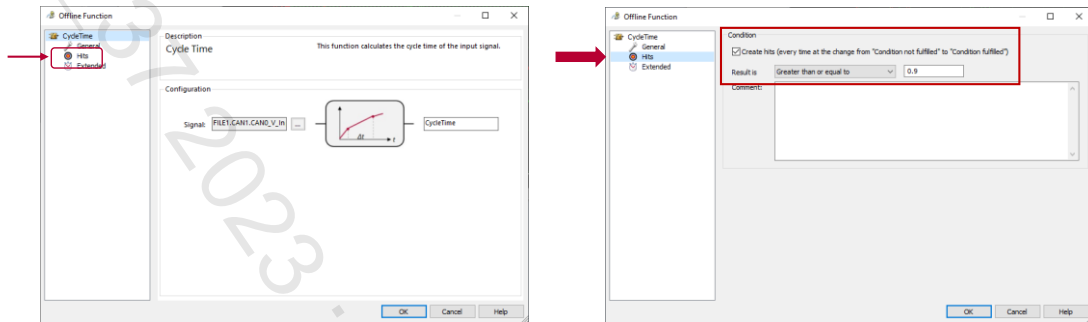
Die Ober- und Untergrenze sowie die Zeitdauer, ab welcher eine Bereichsverletzung durch die Funktion *DetectRangeViolationForPeriod* angezeigt werden soll, sind als lokale Parameter veränderbar. Damit die eingegebenen Grenzwerte und nicht die vordefinierten Werte verwendet werden, muss die Checkbox [Lokale Grenzwerte verwenden] aktiviert sein.



Übung 9d (optional): Data Mining mit einer Funktion als Suchmethode

Beim Konfigurieren Virtueller Messdateikanäle können die Hit-Bedingungen für eine spätere Data Mining Analyse direkt definiert werden.

1. Nutzen Sie die Funktion *CycleTime* als Prüfmethode für ihre nächste Data Mining Konfiguration und überprüfen Sie, ob bei dem Signal *..AIN1_mV* ein Sendeabstand von 0,9 sec überschritten war.
2. Richten Sie ihre Dateifilterliste in der Data Mining Konfiguration so ein, dass sämtliche **CANLogFile_XX.MDF** -Dateien aus dem Verzeichnis *\Measure* untersucht werden.



Übung 10: Berechnung eines virtuellen Messsignals

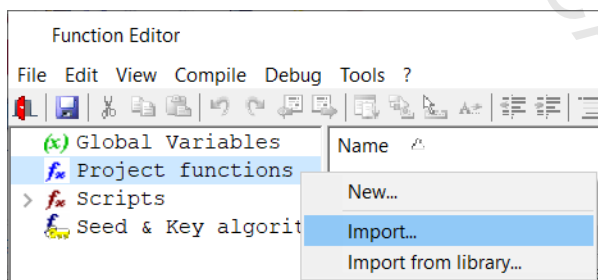
In dieser Übung nutzen Sie eine Funktionen während einer Online-Messung. Die Funktion für diese Übung ist nicht Bestandteil der CANape Funktionsbibliothek. Stattdessen können Sie den Quellcode der Funktion in den vorbereiteten Exercise-Daten im Abschnitt *Scripts_n_Functions* finden.

1. Importieren Sie zuerst die Funktion *Calc_DutyCycle* aus den Exercise-Daten in ihr CANape Projekt über den Funktionseditor und kompilieren anschließend diese (siehe Notizenseite).
2. Instanzieren Sie die Funktion in der Messsignalliste und verknüpfen die Eingangsgröße mit dem *PwmOutput* Signal der Wiper Applikation.
3. Zeigen Sie das Ergebnis der Funktion in einem Ziffern-Fenster an und platzieren Sie zusätzlich ein write- Fenster, weil dort kleine Zwischenergebnisse ausgegeben werden.
4. Optional: Wenn Sie Programmiererfahrung haben, dürfen Sie gerne ihre Funktion erweitern. Hier könnten Sie z.B. mit Hilfe einer globalen Variable den Maximalwert der Duty-Cycle erfassen. Die globale Variable könnten Sie in einem Parameterfenster darstellen.

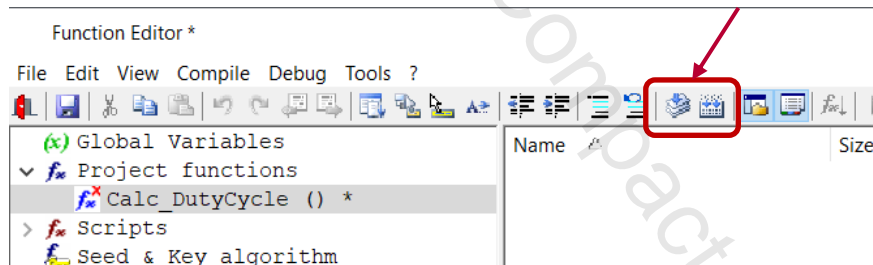
34/40

Wichtiger Hinweis zur Lösung der Teilaufgabe 1:

Mit einem Rechtsklick auf den Eintrag *Projekt Funktionen* können Sie Funktionen u.a. aus *.cne – Dateien importieren.



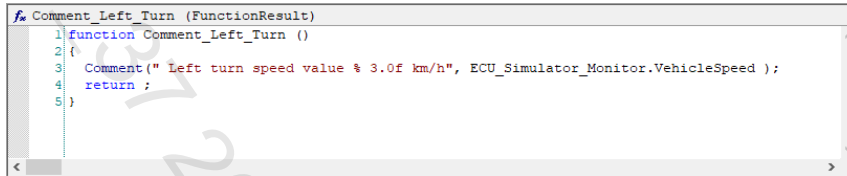
Compile /Compile all



Übung 11 (optional): Schreiben einer eigenen Funktion

In dieser Übung werden Sie eine Funktion programmieren, die während einer Messung auf Tastendruck einen vorbereiteten Kommentar in die Messdatei schreibt. In der Praxis könnte der Tastendruck durchaus eine Taste auf ihrem Lenkrad sein, deren Betätigung auf einem Signal abgebildet wird. In unserer Übung werden Sie ein PC-Tastatur – Event nutzen.

1. Schreiben Sie zuerst zwei kleine Einzeiler-Funktionen, innerhalb derer Sie mit der internen Funktion *Comment* einen Kommentar ihrer Wahl definieren, ähnlich dem abgebildeten Beispiel.



```
f* Comment_Left_Turn (FunctionResult)
1 function Comment_Left_Turn ()
2 {
3   Comment(" Left turn speed value % 3.0f km/h", ECU_Simulator_Monitor.VehicleSpeed );
4   return ;
5 }
```

2. Definieren Sie in der Event-Liste innerhalb der Messkonfiguration zwei Tastatur-Events, z.B. *Comment_Left_Turn* auf der Taste <a> und *Comment_Right_Turn* auf Taste .
3. Instanzieren Sie die Funktion abschließend in der Messsignalliste mit den entsprechenden Aufruf - Events

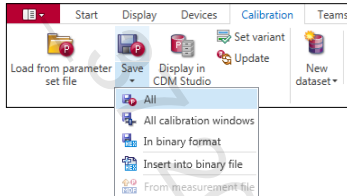
Übung 13: Kalibrieren von Gerätevariablen in der SWC Wiper

In der Übung 2a haben Sie das Ausgangssignal der SWC *Wiper* gemessen, nachdem Sie die Applikation über den Virtuellen Scheibenwischerhebel aktiviert hatten. In dieser Übung werden Sie die Scheibenwischergeschwindigkeit über verschiedene Parameter für die verschiedenen Arbeitsmodi verstellen.

1. Messen Sie die Signale *PwmOutput* und *MultifunctionSwitch* genauso wie in Übung 2a.
2. Platzieren Sie auf der gleichen Anzeigeseite ein Verstell-Fenster mit sämtlichen Parametern aus der Gruppe *Wiper*.
3. Kalibrieren Sie die Scheibenwischergeschwindigkeit in den Modi HI and LO nach Ihrem Belieben (Parameter *HiDuty* und *LoDuty*) und beobachten dabei die Auswirkung an dem *PwmOutput* Signal.
4. Nutzen Sie den Direkt- und Indirekt-Verstellmodus und nutzen auch mal die Möglichkeit im Offline Zustand im Spiegelspeicher neue Parameterwerte vorzubereiten, die Sie dann anschließend ins Steuergerät laden.

Übung 14: Speichern und Laden von Parametersätzen

1. Speichern Sie sich unterschiedliche Parametersätze ab, wobei Sie beispielsweise die Parameter *IntervalCycle*, *IntervalInnerCycle* und das Kennfeld *AutomaticCycleMap* aus der Gruppe *Wiper* verändern.
 - a) Speichern Sie nur die Parameter aus ihren Verstell-Fenstern.
 - b) Speichern Sie alle Objekte aus dem Spiegelspeicher als Parametersatzdatei ab.



2. Laden Sie abschließend unterschiedliche Datensätze in ihr Steuergerät zurück und beobachten dabei die Verstell-Fenster. Nutzen Sie hierfür die Menüfunktion [Verstellen | Laden von Parametersatzdatei]

Übung 15 (optional): Laden von Parametersätzen im vCDMstudio

1. Vergleichen Sie die unterschiedlichen Parametersätze unter Verwendung des vCDMstudio. Beim Sichten der Datensätze nutzen Sie bitte die vorgefertigten Filter aus dem Navigation Fenster.
2. Erstellen Sie sich nun eine leere Parametersatzdatei im vCDMStudio und mischen per Drag&Drop beliebig viele Parameter aus zwei bestehenden Parametersatzdateien zusammen. Benennen Sie den gerade erstellten Parametersatz in MergeData.par um.
3. Schließen Sie nun sämtliche Spalten mit Ausnahme des Parametersatzes MergeData.par und öffnen stattdessen eine neue Spalte mit den Werten des aktuellen Gerätes. Im folgenden sollen mit Hilfe des vCDMstudio die Werte der Parametersatzdatei in das aktuelle Gerät geladen werden. (Siehe dazu die Vorgehensweise in der Notizenseite.)

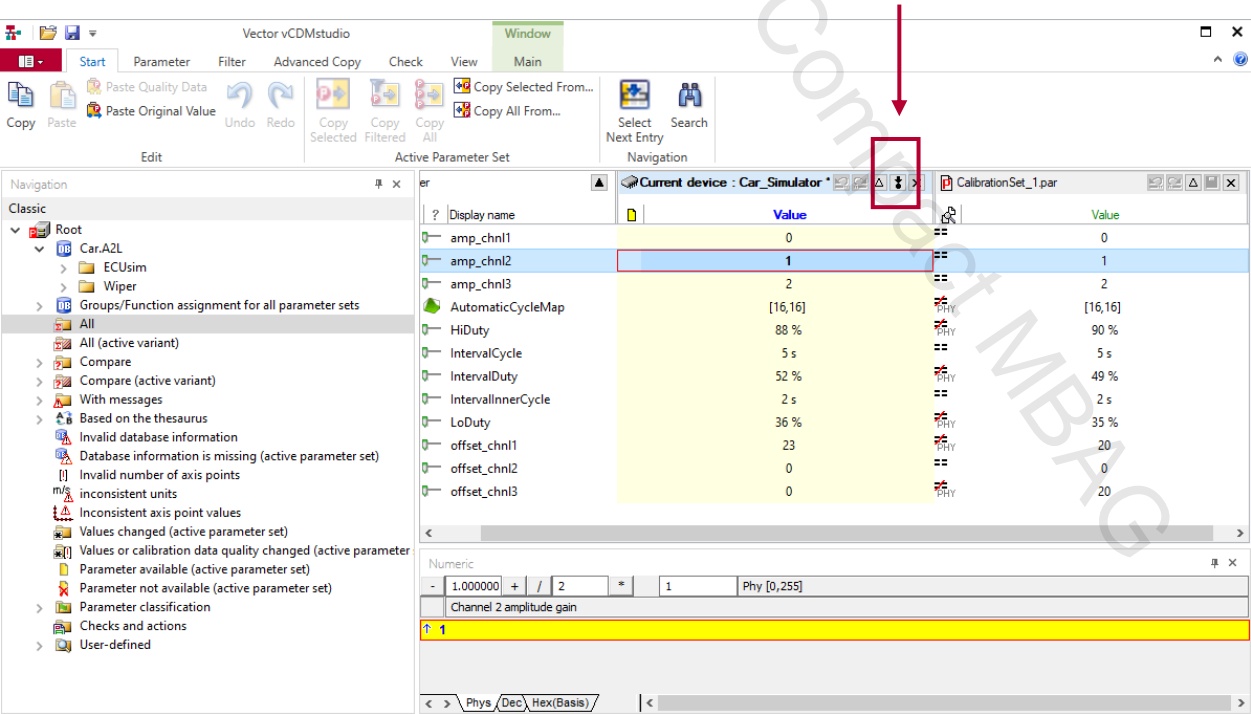
38/48

Wichtiger Hinweis zur Lösung der Teilaufgabe 3:

Das Laden einer Parametersatzdatei in den Speicher des aktuellen Gerätes geschieht im vCDMstudio in zwei getrennten Schritten:

1.Schritt: Die Werte einer Parametersatzdatei werden zunächst in die Spalte des Aktuellen Gerätes kopiert. Dies können Sie entweder mit Drag&Drop durchführen oder Sie nutzen die Funktionen [Alle kopieren in...] ([Copy all into..]) bzw. [Alle in aktiven Parametersatz kopieren] ([Copy all in active parameter set]) aus dem Kontextmenü des Spaltenkopfes der Parametersatzdatei.

2.Schritt: Der eigentliche Download-Vorgang lösen Sie über das markierte Icon aus.



Übung 16 (optional): Erstellen eines Panels in CANape

1. Erstellen Sie mit dem Panel Designer aus CANape eine Oberfläche, in der Sie die Signale *EngineSpeed*, *VehicleSpeed* und *Gear* aus der AutomaticTransmission Applikation anzeigen.
 - Die Bitmaps sind im Verzeichnis *\Bitmaps* zu finden.



39/48

Bei der Ausrichtung des Zeigerinstruments auf dem statischen Hintergrundbild müssen Sie ein wenig experimentieren, bis Sie die optimale Größe des Zeigers gefunden haben. In unserem Beispiel ist der angedeutete Radius des Zeigers auf die längeren Skalierungsstriche ausgerichtet.

Die Nullpunkt- bzw. Vollausschlagwinkel sind hier auf $\pm 120^\circ$ eingestellt. Je nach Hintergrundbild müssen Sie diese Winkleinstellungen anpassen. Vergessen Sie nicht die dazu passenden Messwerte (300km/h) zu konfigurieren..

Properties

Alarm Settings

Alarm Display: False

Lower Limit: 30

Lower Limit Color: Green

Upper Limit: 80

Upper Limit Color: Red

Aperture Angle

From: -120

To: 120

Appearance Needle

Color: 183; 0; 50

Pivot Distance %: 0

Shape: TriangleWideFilled

General

Control Name: Meter_KMH

Symbol

Symbol: vehicle_speed_in_kmh_h

Name: vehicle_speed_in_kmh_h

Database: C:\Users\Public\Documents\sldemo_autotrans_dist_calc

Node: All Networks

Message: n/a

Symbol Filter: Signal

Value Range

Maximum: 300

Minimum: 0

digital_instrument_panel.xvp

Übung 17 (optional): Skript zum Versenden von CAN-Botschaften

Die Tankwarnleuchte des ECU Simulators lässt sich mittels des CAN-Signals *BD_P_FuelLamp* ein- und ausschalten. Schreiben Sie ein Skript welches im Sekundentakt die Warnleuchte ein- bzw. ausschaltet. Ein Rohwert von 0x8000 schaltet die Warnleuchte ein. Ein Rohwert von 0x000 schaltet die Warnleuchte aus. Das CAN-Signal *BD_P_FuelLamp* wird in der CAN-Botschaft *BD_P_LampCommand_6_7_8_9* (ID = 0xA1) versendet.

- (1) Nutzen Sie Gerätevariablen in Verbindung mit den Skript-Funktionen *device.BeginModify()* und *device.EndModify()*.
- (2) Den Sekundentakt können Sie im Skript mit einer Endlosschleife in Verbindung mit der Funktion *Sleep(1000)* realisieren.
- (3) Starten und Stoppen Sie das Skript über den Task Manager von CANape.
- (4) Zusatzfragen:
 - ▶ Könnte das Versenden der CAN-Nachricht auch über eine Funktion in der Messkonfiguration erreicht werden? Welche Vor- und Nachteile ergeben sich.
 - ▶ Welche anderen Möglichkeiten gibt es in CANape einen Skript zu starten?
 - ▶ Welche Möglichkeiten gibt es die Blinkdauer der Warnleuchte dynamisch zu steuern?

Übung 18 (optional): Logger konfigurieren – Permanente Bus Aufzeichnung

Konfigurieren Sie den VC121 Simulator, so dass die CAN Botschaften auf einem realen Kanal eines Vector Hardware Interface gesendet wird. *Für mehr Informationen, siehe Notizenseite.*

1. Die CAN Kommunikation des VC121 Simulators soll mittels eines angeschlossenen GL2400 Loggers aufgezeichnet werden.
 - ▶ Erzeugen Sie in der Vector Logger Suite eine Konfiguration, welche
 - > Permanent, die gesamte CAN Kommunikation des VC121 aufzeichnet.
 - > Bei aktiver Aufzeichnung LED1 blinken lässt.
 - ▶ Schließen Sie den Logger an Ihren Rechner an und laden Sie die Konfiguration.
 - ▶ Trennen Sie den Logger von Ihrem Rechner. Schließen Sie den Logger an den CAN Stecker an, auf welchem der VC121 Simulator seine CAN Botschaften versenden. Versorgen Sie den Logger mit Spannung.
2. Das aufgezeichnete CAN Log soll mittels der Vector Logger Suite in eine signalorientierte MDF Datei exportiert werden.
 - ▶ Schließen Sie den Logger an Ihren Rechner an und exportieren Sie die Aufzeichnung.
3. Setzen von Markern während der Testfahrt.
 - ▶ Bei Betätigung der Fernbedienung soll ein Marker gesetzt werden. Dies ermöglicht das Markieren besonderer Fahrsituationen.

41/42

Konfiguration eines realen CAN-Kanals zur Ausgabe der simulierten CAN-Botschaften:

Stecken Sie ein CAN-fähiges Vector Hardware Interface an Ihren Rechner an (z.B. VN160).

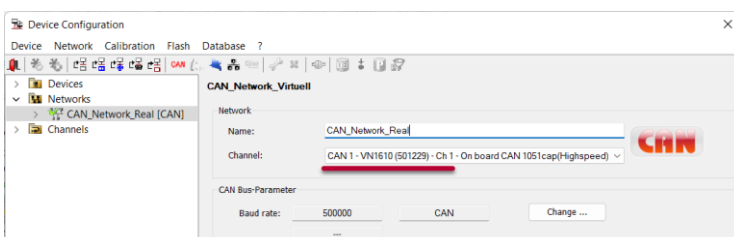
Schalten Sie den Motor des VC121 Simulators aus. Öffnen Sie anschließend die Einstellungen.



Wählen Sie im Dropdown Menü [Interface Channel] einen CAN Kanal Ihres angeschlossenen Vector Hardware Interface aus (z.B. VN1610/ch1).



Schließen Sie zur Aufzeichnung den Logger an den ausgewählten CAN-Stecker an. Eventuell benötigen Sie hierfür noch ein Y-Kabel. Vergessen Sie nicht die Verwendung von Abschlusswiderständen.



Exercise 19 (optional): Configure GL Data Logger – Trigger, Filter, XCP

1. Configure a trigger

- ▶ Use the triggered logging instead of the permanent long-term logging
- ▶ The logger shall record on a change of the B1 switch (signal **BD_D_Switch_B1**). A pre-trigger time of 60 s and a post-trigger time of 40 s shall be recorded.

2. Filtering CAN messages / Sleep

- ▶ The CAN messages with the CAN-ID 0x150 – 0x155 shall not be recorded. Configure a corresponding stop filter.
- ▶ The sleep mode of the GL2400 shall be activated if the CAN bus is iddle for more than 20 s.

3. XCP measurement with the GL2400 logger

- ▶ The signals **PwmOutput**, **MultifunctionSwitch**, **RainSensor** and **LightSensor** shall be recorded via XCP.
- ▶ Configure a DAQ measurement for these signals.

42/42

General

☒ Clear window Within a function, the content of the Write window can be deleted.

Decimal places: The time display in the Write window cannot be more exactly than given in the general settings.

Buffer size: KByte Buffer size for Write window.
Value between 1 - 10.000 KByte possible.