

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>2</b>
1.1	Einleitung	3
1.2	Begriffsdefinitionen	3
1.3	Vorgehensweise	4
<b>2</b>	<b>Erstellung des Testaufbaus</b>	<b>5</b>
2.1	Einführung	6
2.2	Erstellung der EDS-Dateien	6
2.3	Erstellung des Netzwerkdesigns	6
2.4	Generierung der Simulation	7
2.5	Generierung der Signalumsetzung	7
	2.5.1 Kommandozeilenparameter des 447-Generators	7
	2.5.2 Verwendung des 447-Generators	8
2.6	Hinzufügen der Datenbasis	10
2.7	Aktivieren der Signalumsetzung	11
2.8	Aktivieren der LSS-Funktionalität	11
2.9	Erstellen von Testsequenzen	12
	2.9.1 Senden eines Kommandos	13
	2.9.2 Anfordern eines Status	14

# 1 Einführung

In diesem Kapitel finden Sie die folgenden Informationen:

---

1.1	Einleitung	Seite 3
1.2	Begriffsdefinitionen	Seite 3
1.3	Vorgehensweise	Seite 4

---

## 1.1 Einleitung

Dieses Dokument beschreibt die Vorgehensweise zur Erstellung eines Testaufbaus für das Applikationsprofil CiA-447. Der Testaufbau erlaubt die signalbasierte Manipulation von Testparametern. Dabei werden alle in CiA-447 beschriebenen Signale automatisch zur Verfügung gestellt.

## 1.2 Begriffsdefinitionen

### Testaufbau:

Ein Testaufbau besteht aus einer **CANoe**-Konfiguration, CAPL-Programmen und Panels. Mit Hilfe der CAPL-Programme werden CANopen Geräte simuliert. Über deskriptive Umgebungsvariablen können Kommandos gesendet oder Stati gelesen werden.

### 447-Generator:

Der 447-Generator ist ein Generierungswerkzeug zur Erstellung eines CAPL-Programms zur Signalumsetzung. Dieses CAPL-Programm übernimmt die Abbildung der Signalwerte auf Objektwerte gemäß CiA447.

Die Bedienung und Arbeitsweise des Generators sind in Abschnitt **Generierung der Signalumsetzung** beschrieben.

### Signalumsetzung:

Die Spezifikation CiA447 definiert eine Reihe von Signalen, die in Objektwerten abgebildet werden. Ein Objektwert kann dabei mehrere Signale umfassen. Im eigentlichen Testablauf möchte man sich nicht mit diesen Signalbelegungen auseinandersetzen. Die einfache Testanweisung „Sende Kommando Innenlicht an“ soll automatisch ein SDO mit dem entsprechenden Objektwert versenden.

Für diese Umsetzung ist eine Signalumsetzung notwendig. Realisiert wird diese in einem CAPL-Programm, das in den Testaufbau von **CANoe** eingebunden wird.

Die Generierung dieses CAPL-Programms ist in Abschnitt **Generierung der Signalumsetzung** beschrieben.

### Signalbeschreibung:

Informationen über die Signalbelegung der Objektwerte sind lediglich im Standard CiA447 beschrieben. Für die Generierung der Signalumsetzung benötigt der 447-Generator allerdings Informationen, wie beispielsweise die Länge oder die Startposition des Signals.

Diese Angaben müssen in einer separaten Datei, der Signalbeschreibung, angegeben werden bevor die Signalumsetzung generiert werden kann.

## 1.3 Vorgehensweise

Abbildung 1 beschreibt die Vorgehensweise zur Generierung des Testaufbaus:

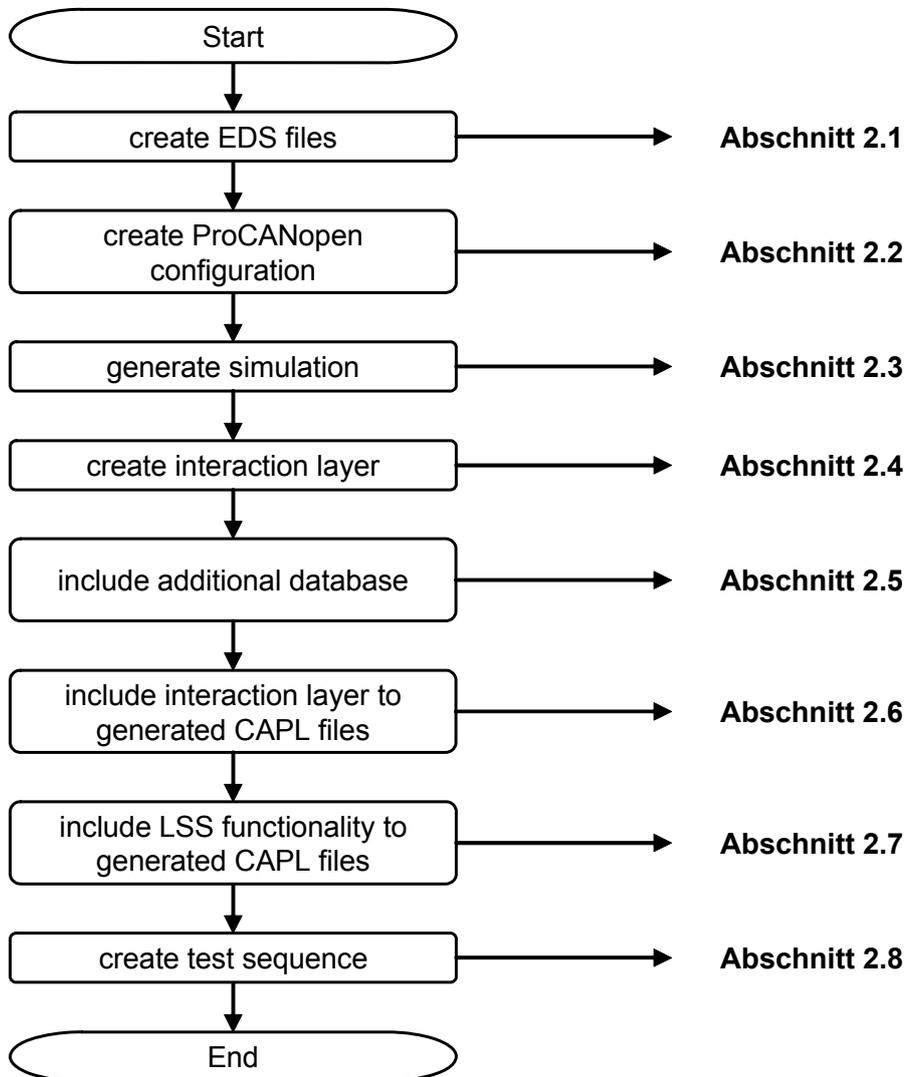


Abbildung 1: Vorgehensweise der Generierung

Im folgenden Kapitel werden die einzelnen Schritte im Detail beschrieben.



### Hinweis:

Dieses Dokument stellt keine Zusammenfassung der Handbücher dar. Es werden lediglich prinzipielle Vorgehensweisen beschrieben.

Detaillierte Informationen zur Bedienung finden Sie in den jeweiligen Handbüchern der Werkzeuge. Es wird außerdem zugrunde gelegt, dass Sie mit den im Applikationsprofil CiA447 beschriebenen Mechanismen vertraut sind.

## 2 Erstellung des Testaufbaus

In diesem Kapitel finden Sie die folgenden Informationen:

---

2.2	Erstellung der EDS-Dateien	Seite 6
2.3	Erstellung des Netzwerkdesigns	Seite 6
2.4	Generierung der Simulation	Seite 7
2.5	Generierung der Signalumsetzung Kommandozeilenparameter des 447-Generators Verwendung des 447-Generators	Seite 7
2.6	Hinzufügen der Datenbasis	Seite 10
2.7	Aktivieren der Signalumsetzung	Seite 11
2.8	Aktivieren der LSS-Funktionalität	Seite 11
2.9	Erstellen von Testsequenzen Senden eines Kommandos Anfordern eines Status	Seite 12

---

## 2.1 Einführung

Getestet werden soll ein Gerät, das nach dem Applikationsprofil CiA447 arbeitet. Alle weiteren im Netzwerk vorhandenen Geräte sollen unter Verwendung von **CANoe** simuliert werden (Restbussimulation). Die simulierten Geräte sollen am LSS-Mechanismus teilnehmen. Im Testablauf soll signalorientiert auf die in CiA-447 beschriebenen Kommandos bzw. Stati zugegriffen werden.

## 2.2 Erstellung der EDS-Dateien

Für die Restbussimulation des 447-Netzwerkes werden die EDS-Dateien der simulierten Geräte benötigt. Liegen für diese Geräte keine EDS-Dateien vor, müssen die Dateien mit **CANeds** erstellt werden. Um mit den simulierten Geräten Kommandos schreiben und Stati lesen zu können, müssen folgende Objekte in der EDS-Datei angelegt werden:

- 0x1000 - Device Type
- 0x1001 - Error Register
- 0x1018 - Identity Object
- 0x6000 - Virtual device support

Das Applikationsprofil CiA447 definiert, dass im Objekt 0x6000 die unterstützten virtuellen Geräte kodiert sind. Geben Sie im Objekt zwingend den resultierenden Objektwert an. Unterstützt ein Gerät beispielsweise das virtuelle Gerät Taxameter, muss im Objekt 0x6000sub1 der Wert 0x800 eingetragen werden.

Wenn das simulierte Gerät die LSS-Services unterstützen soll, müssen im Objekt 0x1018 (Identity Object) gültige Werte angegeben werden. Sind im Netzwerk mehrere baugleiche Geräte vorhanden, müssen sich diese Geräte mindestens in der Seriennummer (Objekt 0x1018sub4) unterscheiden. Der Wert der Seriennummer kann daher in einem späteren Schritt in **ProCANopen** geändert werden. Somit kann für baugleiche Geräte immer dieselbe EDS-Datei verwendet werden.

Prüfen Sie das Objektverzeichnis, bevor Sie die EDS-Datei speichern. Die Prüfung kann über das Menükommando **Prüfen | Prüfe EDS** gestartet werden. Achten Sie darauf, dass in **CANeds** lediglich die Datenbanken `v301.codb` und `v447.codb` für die Prüfung der EDS-Datei verwendet werden. Im Dialog **Ausgewählte Datenbasen** (Menükommando **Datenbasis | Bearbeite Liste**) können diese Einstellungen vorgenommen werden.

## 2.3 Erstellung des Netzwerkdesigns

Binden Sie alle Geräte in eine neue **ProCANopen**-Konfiguration ein. Weisen Sie den Knoten die erstellten EDS-Dateien zu. Achten Sie darauf, dass in **ProCANopen** das Applikationsprofil 447 eingestellt ist. Aktivieren Sie hierfür das Optionsfeld **DS 447 – Sonderfahrzeuge** im Dialog **Einstellungen | Applikationsprofil** (Menükommando **Optionen | Einstellungen ...**).

Typischerweise wird die Seriennummer nicht in den EDS-Dateien der Geräte angegeben. Prüfen Sie, ob für alle Geräte eine eindeutige Seriennummer angegeben ist. Haben Sie zwei baugleichen Geräten dieselbe EDS-Datei zugewiesen, müssen Sie die Seriennummer nach dem Einfügen der EDS-Dateien ändern.

Öffnen Sie hierfür den **Gerätezugriff** (Rechtsklick auf das Knotensymbol im Topologie-Fenster) und selektieren Sie das Objekt 0x1018sub4. Im Eingabefeld **Wert** kann eine eindeutige Seriennummer angegeben werden. Da das Objekt nicht schreibbar ist, müssen Sie zuvor die Schaltfläche **[Überschreiben]** aktivieren.

## 2.4 Generierung der Simulation

Generieren Sie die **CANoe**-Simulation für das in **ProCANopen** entworfene Netzwerk. Die Generierung wird über das Menükommando **Projekt | CANoe Konfiguration generieren...** gestartet.

## 2.5 Generierung der Signalumsetzung

In den simulierten Geräten sind die in CiA-447 beschriebenen SDO-Server und SDO-Clients implementiert. Das Senden eines Kommandos bzw. die Anforderung eines Status kann demnach bereits in einem Testprogramm implementiert werden. Beispielsweise kann die Funktion `coDownload(...)` verwendet werden, um ein bestimmtes Kommando zu senden. In den Funktionsparametern müssen Sie dann neben dem Zielknoten das Zielobjekt und den Objektwert angeben.

Um ein spezielles Signal zu senden, benötigen Sie unter Anderem folgende Angaben, um den Objektwert zu bestimmen und das SDO zu senden:

- Index + Subindex des Objekts, in dem der Signalwert abgebildet ist
- Signallänge
- Startposition des Signals

Diese Informationen sind lediglich im CiA-447-Standard beschrieben. Der Aufbau eines einzelnen Testschrittes würde zwangsweise eine genaue Betrachtung des Standards nach sich ziehen.

Um im eigentlichen Testablauf signalorientiert arbeiten zu können, ist die Generierung eines weiteren CAPL-Programms notwendig, das die Abbildung der in CiA447 beschriebenen Signalwerte auf die jeweiligen Objektwerte realisiert.

### 2.5.1 Kommandozeilenparameter des 447-Generators

Die Erstellung des CAPL-Programms wird über einen separaten Generator, den 447-Generator, realisiert. Dieses Kommandozeilenwerkzeug generiert ein CAPL-Programm, das in die bereits existierenden simulierten Geräte eingebunden werden kann. Der 447-Generator sollte für jeden simulierten Knoten aufgerufen werden.

Hierbei sind folgende Kommandozeilenparameter zu verwenden:

Parameter	Wert	Beschreibung
-m	3	Angabe des Generierungsmodus Für die Generierung der Signalumsetzung muss hier der Wert 3 angegeben werden.
-nme	<Name des DUT>	Name des zu testenden Gerätes Über diesen Namen wird im späteren Test der Empfänger eines SDOs angegeben. Für diese Adressierung wird bewusst der Name, und nicht die Knotennummer verwendet. Somit ist sichergestellt, dass der Test auch für Knoten reproduzierbar ist, die am LSS teilnehmen.
-tme	<Name des simulierten Gerätes>	Name des simulierten Gerätes Über diesen Namen wird im Testablauf der Sender eines SDOs angegeben.
-dbc	<Pfad der Datenbasis>	Pfad der Datenbasis, die vom 447-Generator angelegt wird Für jedes in CiA447 beschriebene Signal werden in dieser Datenbasis Umgebungsvariablen angelegt. Im Testablauf werden diese Umgebungsvariablen verwendet, um den zu sendenden Signalwert zu setzen oder einen Status anzufordern.

Parameter	Wert	Beschreibung
-xls	<Pfad der Datei zur Signalbeschreibung>	Pfad der Datei zur Signalbeschreibung für den 447-Generator Die Datei muss im Excel-Format vorliegen. In dieser Tabelle müssen alle zu testenden Signale aufgelistet sein. Das Format der Tabelle ist im Abschnitt Verwendung des 447-Generators beschrieben.
-out	<Pfad des CAPL-Programms zur Signalumsetzung>	Pfad der Ausgabedatei.  <b>Hinweis:</b> Es muss die Dateiendung <code>.cin</code> verwendet werden, da das generierte CAPL-Programm in die bestehenden CAPL-Programme der simulierten Knoten eingebunden wird.

Tabelle 1: Kommandozeilenparameter des 447-Generators

**Hinweis:**

Die Generierung der Signalumsetzung muss für jeden simulierten Knoten einmal durchgeführt werden. Geben Sie in der Kommandozeile für jeden Knoten dieselbe Datenbasis (Parameter **-dbc**), dieselbe Ausgabedatei (Parameter **-out**) und dasselbe CAPL-Programm zur Signalbeschreibung (Parameter **-xls**) an.

Dies hat den Vorteil, dass Sie in alle simulierten Knoten dasselbe CAPL-Programm einbinden können. Außerdem müssen Sie der **CANoe**-Konfiguration lediglich eine weitere Datenbasis hinzufügen.

## 2.5.2 Verwendung des 447-Generators

In der Datenbasis, die in der Kommandozeile angegeben wurde, werden für jedes in CiA447 beschriebene Signal Umgebungsvariablen angelegt, über die der Testablauf gesteuert werden kann. Die Information, für welche Signale die Generierung durchgeführt werden muss, und die Beschreibung der Abbildung der Signale auf die Objektwerte sind in der Excel-Tabelle enthalten. Diese Tabelle muss vor der Generierung basierend auf dem Applikationsprofil CiA447 erstellt werden.

Folgende Punkte sind beim Anlegen einer Tabelle zur Signalbeschreibung zu beachten:

- Weisen Sie dem Tabellenblatt, auf dem sich die Tabelle befindet, den Namen **Signals** zu
- Die in Tabelle 2 beschriebenen Spalten müssen vorhanden sein. Über eine Initialisierungsdatei müssen Sie dem 447-Generator mitteilen, in welcher Spalte welche Information zu finden ist.

**Hinweis:**

Weisen Sie der Initialisierungsdatei denselben Namen wie der Excel-Datei zu und legen Sie die Datei im selben Verzeichnis ab. Verwenden Sie dabei als Dateierweiterung `.ini`. Fügen Sie dieser Initialisierungsdatei die Sektion **[XLSImport]** hinzu. Die Angabe der Spalten erfolgt über die in Tabelle 2 beschriebenen Schlüssel.

Weisen Sie dem Schlüssel die Spaltennummer (nullbasiert) zu, in der sich der entsprechende Wert befindet. Die Beschriftungen der Spalten sind demnach frei wählbar.

Folgende Spalten müssen in der Tabelle enthalten sein:

Spalte	Bedeutung	Schlüssel in Initialisierungsdatei
Index	hexadezimaler Index des Objekts, in dem der Signalwert abgebildet wird.	ObjectIndex
Subindex	hexadezimaler Subindex des strukturierten Objekts, in dem der Signalwert enthalten ist. Bei nicht strukturierten Objekten ist der Subindex immer 0.	ObjectSubIndex
Objektgröße	Größe des Objekts in Bit	ObjectSize
Signalname	<p>Name des Signals. Dieser Name wird für die Bildung der Umgebungsvariablen-namen verwendet. Es bietet sich an, möglichst deskriptive und kurze Namen zu verwenden da Sie Ihren Testablauf unter Verwendung dieser Umgebungsvariablen erstellen werden.</p> <hr/> <p><b>Hinweis:</b> Signalnamen sollten möglichst eindeutig sein. Werden sie mehrfach vergeben, bildet der 447-Generator automatisch einen eindeutigen Signalnamen. Für die Bildung des Namens werden der Index, der Subindex und die Position des Signals im Objekt verwendet.</p>	SignalName
Offset des Signals	Offset des Signals im Objektwert. Die Angabe des Offsets erfolgt in Bit (nullbasiert)	SignalOffset
Größe des Signals	Signalgröße in Bit	SignalSize
Reserviertes Signal	Einige Signale sind im CiA447 Standard reserviert. Für diese Signale werden keine Umgebungsvariablen generiert. Reservierte Signale müssen in Tabelle in dieser Spalte mit einem <b>x</b> gekennzeichnet werden.	SignalReserved
Signal wird verwendet	In dieser Spalte kann angegeben werden, ob das beschriebene Signal verwendet wird. Verwendete Signale sind in dieser Spalte mit einem <b>x</b> zu kennzeichnen.	SignalUsed
Kommando oder Status	In dieser Spalte wird angegeben, ob es sich beim Signal um ein Kommando oder einen Status handelt. Ein Kommando wird über ein <b>x</b> gekennzeichnet. Handelt es sich um einen Status, tragen Sie in die Zelle keinen Wert ein.	CommandOrStatus

Tabelle 2: Spalten in der Excel Tabelle

Index	Objektgröße	Sub-Index	reserviert	Startbit	Signallänge	Signalname	wird verwendet	Kommando/Status
6005	8	0		0	8	Switch illumination level status	X	
6006	8	0		0	8	Switch illumination level command	X	x
6007	8	0		6	2	ignition start name	X	
6007	8	0		4	2	ignition on	X	
6007	8	0		2	2	accessory	X	
6007	8	0		0	2	ignition lock	X	
6009	8	0		0	4	central locking system status	X	
6009	8	0	X	4	4	central locking system status reserved	X	
600A	8	0		0	4	central locking system command	X	X
600A	8	0	X	4	4	central locking system command reserved	X	X
600B	24	0	X	18	6	window status reserved	X	
600B	24	0		15	3	window status 3rd row left	X	
600B	24	0		12	3	window status 3rd row right	X	
600B	24	0		9	3	window status rear left	X	
600B	24	0		6	3	window status rear right	X	
600B	24	0		3	3	window status front passenger	X	
600B	24	0		0	3	window status driver	X	
600C	16	0	X	12	4	window command reserved	X	X
600C	16	0		10	2	window command 3rd row left	X	X
600C	16	0		8	2	window command 3rd row right	X	X
600C	16	0		6	2	window command rear left	X	X
600C	16	0		4	2	window command rear right	X	X
600C	16	0		2	2	window command front passenger	X	X
600C	16	0		0	2	window command driver	X	X
600D	16	0	X	12	4	door status reserved	X	
600D	16	0		9	3	door status rear left	X	
600D	16	0		6	3	door status rear right	X	
600D	16	0		3	3	door status front passenger	X	
600D	16	0		0	3	door status driver	X	

Abbildung 2: Beispiel einer Excel-Tabelle zur Signalbeschreibung

Es folgt ein Beispiel einer Initialisierungsdatei, die für dieses Tabellenformat angelegt werden muss. Der Schlüssel **SkipRows** gibt an, wie viele Reihen beim Einlesen der Excel-Datei nicht beachtet werden. Der Wert 1 bedeutet demnach, dass die erste Reihe nicht eingelesen wird, da sie in diesem Beispiel die Überschriften für die Spalten enthält. Die Nummer hinter dem jeweiligen Schlüssel gibt an, in welcher Spalte sich der entsprechende Eintrag befindet. Die Angabe der Nummer erfolgt nullbasiert!

**Beispiel:**

```
[XLSImport]
SkipRows=1
ObjectIndex=0
ObjectSize=1
ObjectSubIndex=2
SignalReserved=3
SignalOffset=4
SignalSize=5
SignalName=6
SignalIsUsed=7
CommandOrStatus=8
```

## 2.6 Hinzufügen der Datenbasis

Binden Sie die vom 447-Generator erstellte Datenbasis (im Parameter **-dbc** haben Sie einen Pfad für die Datenbasis angegeben) in die generierte **CANoe**-Konfiguration ein.

## 2.7 Aktivieren der Signalumsetzung

Nach der Generierung müssen Sie das CAPL-Programm zur Signalumsetzung in die CAPL-Programme Ihrer simulierten Knoten importieren. Öffnen Sie die **CANoe**-Konfiguration und anschließend den CAPL Browser für einen Ihrer generierten CANopen-Knoten. Binden Sie die Signalumsetzung in den simulierten Knoten (Rechtsklick auf den Eintrag **Include files** und Auswahl des Menükommandos **Add**) ein.

Entfernen Sie die Funktion `coOnUploadResponse (...)` aus der Gruppe **Callback function** Ihres simulierten Knotens. Die Funktion wird im importierten CAPL-Programm überschrieben und muss deshalb aus dem CAPL-Programm des simulierten Knotens entfernt werden.

## 2.8 Aktivieren der LSS-Funktionalität

Soll ein simulierter Knoten die LSS-Services unterstützen, gehen Sie bitte wie folgt vor:



1. Importieren Sie die Datei `LSS_slave.cin` in das CAPL-Programm des simulierten Knotens, der die LSS-Services unterstützen soll (Rechtsklick auf den Eintrag **Include files** und Auswahl des Menükommandos **Add**).
2. Löschen Sie die Funktion `start` aus der Gruppe **System** des simulierten Knotens.
3. Löschen Sie die Funktion `PreStart` aus der Gruppe **System** des simulierten Knotens.
4. Fügen Sie dem simulierten Knoten die Funktion `showNodeID (...)` hinzu. Diese ist wie folgt zu implementieren:

```
showNodeID (byte nodeID)
{
    putValue (Node<Knotennummer>_ID, nodeID);
}
```



### Hinweis:

Der Platzhalter `<Knotennummer>` ist durch die Nummer zu ersetzen, die dem Knoten im **ProCANopen**-Projekt zugewiesen wurde.

Simulierte Knoten, welche die LSS-Services unterstützen, schicken unmittelbar nach dem Messungsstart keinen Boot-up. Auf den generierten Panels ist die Knotennummer 255 zu sehen (siehe Abbildung 3). Erst wenn das LSS abgeschlossen wurde und das Gateway dem Knoten eine eindeutige Knotennummer zugewiesen hat, wird die korrekte Knotennummer im Panel angezeigt. Dies wird über die Umgebungsvariable `Node<Knotennummer>_ID` realisiert.



Abbildung 3: Knoten mit LSS Unterstützung (ID noch nicht konfiguriert)

## 2.9 Erstellen von Testsequenzen

Beim Start des 447-Generators geben Sie eine Datenbasis an, die im Rahmen der Generierung erstellt wird. In dieser werden Umgebungsvariablen angelegt, die für die Erstellung des Testablaufs verwendet werden.

Folgende Umgebungsvariablen werden in der Datenbasis einmalig angelegt:

Name	Bedeutung
ID_ActualClient	In dieser Variable wird angegeben, welcher Knoten das im nächsten Testschritt angegebene Signal senden soll.
ID_Dut	In dieser Variable wird angegeben welcher Knoten das im nächsten Testschritt angegebene Signal empfangen soll.
ID_<Knotenname>	Der Platzhalter <Knotenname> wird vom 447-Generator durch den in <b>ProCANopen</b> angegebenen Knotennamen ersetzt. Dieser Variable muss die aktuelle Knotennummer des Gerätes zugewiesen werden. Für Geräte mit fixer Knotennummer ist dies die Nummer, die dem Knoten in <b>ProCANopen</b> zugewiesen wurde. Nimmt der Knoten am LSS teil, muss die Variable nach Ablauf des LSS gesetzt werden. Hierfür bietet sich die Funktion <code>showNodeID(...)</code> an, die Sie bereits in die CAPL-Programme der simulierten Knoten eingefügt haben.

Tabelle 3: Umgebungsvariablen zur Steuerung der SDO Kommunikation

Folgende Variablen werden für jedes in der Signalbeschreibung angegebene Signal angelegt. Der Platzhalter <Signalname> wird vom 447-Generator durch den in der Signalbeschreibung angegebenen Namen ersetzt.

Name	Bedeutung
coval_<Signalname>	<p>Variablen mit dem Präfix <code>coval_</code> (command value) werden im Testprogramm dazu verwendet, den Wert eines Signals zu setzen.</p> <hr/> <p><b>Beispiel:</b></p> <pre>putValue(coval_additional_function_4,3);</pre> <p>1.Funktionsparameter: Name der Umgebungsvariable des Kommandos 2.Funktionsparameter: Wert des Signals</p>
sc_<Signalname>	<p>Variablen mit dem Präfix <code>sc_</code> (send command) werden im Testprogramm dazu verwendet, das zuvor gesetzte Kommando zu senden. Es wird ein SDO Download auf das Objekt ausgelöst, in dem das angegebene Signal enthalten ist.</p> <hr/> <p><b>Beispiel:</b></p> <pre>putValue(sc_additional_function_4,1);</pre> <p>1.Funktionsparameter: Name der Umgebungsvariable des Kommandos 2. Funktionsparameter: Auslösen des SDOs. Hier muss der Wert 1 angegeben werden</p>

Name	Bedeutung
gs_<Signalname>	<p>Variablen mit dem Präfix <b>gs_</b> (get status) werden im Testprogramm dazu verwendet, eine Statusanforderung zu senden. Es wird ein SDO Upload auf das Objekt ausgelöst, in dem das angegebene Signal enthalten ist.</p> <p><b>Beispiel:</b></p> <pre>putValue(gs_additional_function_4,1);</pre> <p>1.Funktionsparameter: Name der Umgebungsvariablen des Status 2.Funktionsparameter: Auslösen des SDOs. Hier muss der Wert 1 angegeben werden</p>
stval_<Signalname>	<p>Variablen mit dem Präfix <b>stval_</b> (status value) werden verwendet, um den gelesenen Statuswert zu prüfen. Wenn zuvor eine Statusanforderung gesendet wurde, ist der empfangene Wert nach der Ausführung des SDO Upload in dieser Umgebungsvariable enthalten.</p> <p><b>Beispiel:</b></p> <pre>value = getValue(stval_additional_function_4);</pre> <p>Funktionsparameter: Name der Umgebungsvariable des Status</p>

Tabelle 4: Umgebungsvariablen zur Parametrierung der Signalwerte

## 2.9.1 Senden eines Kommandos

Das folgende Beispiel zeigt, wie ein Testablauf aussehen könnte:



### Beispiel:

```
putValue(ID_Dut, getValue(ID_Gateway)); // (1)
putValue(ID_ActualClient, getValue(ID_Taxameter)); // (2)
putValue(coval_additional_function_4, 3); // (3)
putValue(sc_additional_function_4, 1); // (4)
```

Mit der Anweisung (1) wird die Nummer des aktuell zu testenden Gerätes gesetzt. Die Variable `ID_Gateway` muss vor der Testausführung mit der Knotennummer des Gateway initialisiert werden.

Anweisung (2) legt fest, welcher Knoten das nächste Kommando senden soll. Die Variable `ID_Taxameter` kann entweder beim Start des Tests (Geräte mit fixer Knotennummer) oder nach der Vergabe der Knotennummer durch das Gateway (Funktion `showNodeID` in den simulierten Geräten) gesetzt werden.

Mit Anweisung (3) wird der Wert des Signals `additional_function_4` auf den Wert 3 gesetzt.

Anweisung (4) hat zur Folge, dass das Signal `additional_function_4` per SDO Download übertragen wird. Alle weiteren Signale, die im selben Objektwert abgebildet sind, werden auf **don't care** gesetzt. Das gleichzeitige Senden mehrerer Kommandos wird in der aktuellen Version noch nicht unterstützt. Anweisung (3) zieht deshalb zwingend Anweisung (4) nach sich.

Im weiteren Verlauf der Testsequenz können weitere Signale gesendet werden, ohne dass der Wert für `ID_Dut` und `ID_ActualClient` erneut angegeben werden muss (jeweils Anweisung (3) und (4) verwenden).

## 2.9.2 Anfordern eines Status

Die Festlegung der Sender und Empfänger erfolgt wie beim Senden von Kommandos. Folgende Anweisung löst die Anforderung eines Status aus und wertet den empfangenen Wert aus:

```
putValue(ID_Dut, getValue(ID_Gateway));           // (1)
putValue(ID_ActualClient, getValue(ID_Taxameter)); // (2)
putValue(gs_additional_function, 1);             // (3)
res = TestWaitForEnvVar(stval_additional_function, 100) // (4)
```

Die Anweisung (3) sendet einen SDO Upload Request vom Knoten **Taxameter** an das Gateway.

Die Funktion `TestWaitForEnvVar` der Anweisung (4) ist Teil der Test Service Library von **CANoe**. Sie wartet, bis die im Parameter 1 angegebene Umgebungsvariable gesetzt wird.

Im weiteren Verlauf des Tests könnten Sie den Parameter `res` auswerten und entsprechende Einträge in der Report-Datei erzeugen. Mit dem Test Feature Set kann außerdem geprüft werden, ob der erwartete Status in der Variable `stval_additional_function` vorhanden ist.



---

**Verweis:**

Weitere Informationen zum Test Feature Set entnehmen Sie bitte dem Handbuch oder der Onlinehilfe von **CANoe**.

---