



# **SimBox**

## **Bedienungsanleitung**

**17.12.2012**

**Version: 1.0**

---

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b> .....	<b>4</b>
<b>1 Allgemeines</b> .....	<b>7</b>
<b>2 Anschluss der SimBox</b> .....	<b>9</b>
2.1 Energieversorgung.....	10
2.2 CAN.....	10
2.3 Bedienelemente.....	11
2.4 Aktivierung und Standby.....	11
2.5 Standard-Betriebsart.....	12
2.6 Sondertasten.....	12
2.7 Firmware-Update.....	12
<b>3 SimBox-Client</b> .....	<b>13</b>
3.1 Installation des Clients.....	13
3.2 Startfenster.....	13
3.3 Konfiguration laden.....	15
3.4 Konfiguration übertragen.....	17
3.5 Firmware-Update.....	18
3.6 Konfiguration speichern.....	18
3.7 Alle Elemente löschen.....	18
3.8 Konfiguration prüfen.....	18
3.9 Setzen der Trennzeichen bei der CSV-Datei.....	18
3.10 Programm beenden.....	19
<b>4 Konfiguration</b> .....	<b>20</b>
4.1 Allgemeine Konfigurationseinstellungen.....	20
<b>5 Aktionen</b> .....	<b>22</b>
5.1 Text setzen.....	23
5.2 Wert auf Display anzeigen.....	23
5.2.1 Copy CAN signal.....	23
5.2.2 Show Value.....	24
5.3 Botschaft erstellen.....	25
5.4 Botschaft senden.....	25
5.5 Signal einer CAN-Botschaft dynamisch verändern.....	26
5.5.1 Wert setzen.....	27
5.5.2 Wert erhöhen/vermindern (Add/Subtract).....	27
5.5.3 Wert erhöhen/vermindern mit Begrenzung (AddLimit/SubtractLimit bzw. Increment/Decrement)	27
5.5.4 Wert erhöhen/vermindern mit Überlauf (AddWrap/SubtractWrap).....	27
5.6 Timer.....	27
5.6.1 Start.....	28
5.6.2 Stop.....	28
5.6.3 Reset.....	28
5.6.4 Set.....	28
5.6.5 Set Cyclic.....	28
5.6.6 Set Single.....	28
5.7 Aktion ändern.....	28
5.8 Aktion löschen.....	28
5.9 Aktionen in der Ereignisliste hervorheben.....	28
<b>6 Ereignisse</b> .....	<b>30</b>

---

6.1	Anfangsereignis .....	30
6.2	Tastaturereignis .....	30
6.3	Timer.....	31
6.4	Ereignis ändern.....	31
6.5	Ereignis löschen.....	31
6.6	Zuweisen einer Aktion zu einem bestehenden Ereignis .....	32
6.7	Eine Aktion von einem Ereignis entfernen.....	32
6.8	Ein Ereignis von allen Aktionen entbinden .....	32
6.9	Ein Aktion in der Aktionsliste bewegen.....	32
6.10	Aktionen aus einer Ereignisliste hervorheben .....	33
<b>7</b>	<b>Tutorial: Erstellung einer Konfiguration .....</b>	<b>34</b>
7.1	Eine „Hallo Welt“-Konfiguration .....	34
7.2	Erweiterte „Hallo Welt“-Konfiguration .....	34
7.3	Senden einer CAN-Nachricht .....	35
7.4	Timer-gesteuertes Senden und Anzeige des Werts .....	36
7.5	Timer-gesteuertes Senden bei Tastendruck.....	37
7.5.1	Verwenden eines zyklischen Timers.....	37
7.5.2	Verwenden eines Single-Shot-Timers.....	38
7.6	Timer-gesteuertes Senden bei Tastendruck mit Anschlagverzögerung .....	38
7.7	Simulation einer Geschwindigkeitsanzeige.....	39
<b>8</b>	<b>CSV-Konfigurationen zu den Beispielen.....</b>	<b>41</b>
<b>9</b>	<b>SimBox-Bootloader .....</b>	<b>44</b>
9.1	Bootloader.....	44
9.2	Aufruf durch die Firmware .....	44
9.3	Start nach Update-Abbruch.....	45
9.4	Start mittels Tastenkombination .....	46
<b>Anhang A:</b>	<b>Technische Daten und Support.....</b>	<b>47</b>
<b>Anhang B:</b>	<b>Anschlussbelegung.....</b>	<b>48</b>
<b>Anhang C:</b>	<b>Änderungsindex .....</b>	<b>49</b>

# Abbildungsverzeichnis

Abbildung 1: SimBox .....	7
Abbildung 2: SimBox Schnittstellen .....	9
Abbildung 3: SimBox Anschluss .....	10
Abbildung 4: SimBox Tastatur & Display .....	11
Abbildung 5: Startbildschirm .....	11
Abbildung 6: Disclaimer .....	12
Abbildung 7: Firmware-Update .....	12
Abbildung 8: SimBox Client Startfenster .....	14
Abbildung 9: Auswahl beim Laden einer CSV-Datei. ....	15
Abbildung 10: Client nach dem Anfügen einer weiteren Konfiguration .....	16
Abbildung 11: Fehlerhinweise beim Laden einer Konfiguration .....	17
Abbildung 12: Tasten zum Einstellen des Update-Modus.....	17
Abbildung 13: Konfigurationseinstellungen .....	21
Abbildung 14: Verwalten der Aktionsliste .....	22
Abbildung 15: Aktionsname setzen .....	23
Abbildung 16: Copy CAN signal .....	24
Abbildung 17: Show value .....	24
Abbildung 18: Set CAN message .....	25
Abbildung 19: Send CAN message .....	26
Abbildung 20: Modify CAN message.....	26
Abbildung 21: Timer modifizieren .....	27
Abbildung 22: Kontextmenü einer Aktion .....	29
Abbildung 23: Ereignis-Auswahl.....	30
Abbildung 24: Tastaturereignis definieren .....	31
Abbildung 25: Aktion in Ereignisliste hervorheben .....	33
Abbildung 26: Aufruf des Bootloaders .....	45
Abbildung 27: SimBox-Display bei Firmware-Update .....	45
Abbildung 28: SimBox-Display nach Abbruch eines Updates.....	45
Abbildung 29: SimBox-Display bei Anwendung der Tastenkombination .....	46

## LIZENZVERTRAG

LESEN SIE BITTE DIE LIZENZVEREINBARUNG DIESES LIZENZVERTRAGES SORGFÄLTIG, BEVOR SIE DIE SOFTWARE INSTALLIEREN. DURCH DAS INSTALLIEREN DER SOFTWARE STIMMEN SIE DEN BEDINGUNGEN DIESES LIZENZVERTRAGES ZU.

DIESE SOFTWARE-LIZENZVEREINBARUNG, NACHFOLGEND ALS "LIZENZ" BEZEICHNET, ENTHÄLT ALLE RECHTE UND BESCHRÄNKUNGEN FÜR ENDANWENDER, DIE DEN GEBRAUCH DER BEGLEITENDEN SOFTWARE, BEDIENTUNGSANLEITUNG UND SONSTIGEN UNTERLAGEN, NACHFOLGEND ALS "SOFTWARE" BEZEICHNET, REGELN.

1. DIESER LIZENZVERTRAG IST EINE VEREINBARUNG ZWISCHEN DEM LIZENZGEBER UND LIZENZNEHMER, DER DIE LIZENZ ERHÄLT, UM DIE GENANNTA SOFTWARE ZU VERWENDEN.
2. DEM LIZENZNEHMER IST BEKANNT, DASS DIES NUR EINE BESCHRÄNKTE NICHTEXKLUSIVE LIZENZ IST. DIES BEDEUTET DASS DER LIZENZNEHMER KEINERLEI RECHT AUF UNTER-LIZENZVERGABE HAT. DER LIZENZGEBER IST UND BLEIBT DER EIGENTÜMER ALLER TITEL, RECHTE UND INTERESSEN AN DER SOFTWARE.
3. DIE SOFTWARE IST URHEBERRECHTLICH GESCHÜTZTES EIGENTUM DER TELEMOTIVE AG. DAS PROGRAMM ODER TEILE DAVON DÜRFEN NICHT AN DRITTE VERMIETET, VERKAUFT, WEITERLIZENZIERT ODER SONST IN IRGENDWEINER FORM OHNE AUSDRÜCKLICHE, SCHRIFTLICHE GENEHMIGUNG DER TELEMOTIVE AG WEITERVERMARKTET WERDEN. DER ANWENDER DARF DIE SOFTWARE UND DEREN BESTANDTEILE WEDER VERÄNDERN, MODIFIZIEREN NOCH SONST IN JEGLICHER FORM RÜCKENTWICKELN ODER DEKOMPILIEREN.
4. DIESE SOFTWARE UNTERLIEGT KEINER GARANTIE. DIE SOFTWARE WURDE VERKAUFT WIE SIE IST, OHNE JEGLICHE GARANTIE. FALLS IRGENDWANN EIN BENUTZER SEIN SYSTEM ÄNDERT, TRÄGT DER LIZENZGEBER KEINE VERANTWORTUNG DAFÜR, DIE SOFTWARE ZU ÄNDERN, DAMIT SIE WIEDER FUNKTIONIERT.
5. DIESE LIZENZ ERLAUBT DEM LIZENZNEHMER, DIE SOFTWARE AUF MEHR ALS EINEM COMPUTERSYSTEM ZU INSTALLIEREN, SOLANGE DIE SOFTWARE NICHT GLEICHZEITIG AUF MEHR ALS EINEM COMPUTERSYSTEM VERWENDET WIRD. DER LIZENZNEHMER DARF KEINE KOPIEN DER SOFTWARE MACHEN ODER KOPIEN DER SOFTWARE ERLAUBEN, WENN KEINE AUTORISIERUNG DAFÜR BESTEHT. DER LIZENZNEHMER DARF LEDIGLICH ZU AUSHILFSZWECKEN KOPIEN DER SOFTWARE MACHEN. DER LIZENZNEHMER IST NICHT BERECHTIGT, DIE SOFTWARE ODER IHRE RECHTE AUS DIESER LIZENZVEREINBARUNG WEITERZUGEBEN ODER ZU ÜBERTRAGEN.
6. DER LIZENZGEBER IST GEGENÜBER DEM LIZENZNEHMER WEDER FÜR SCHÄDEN, EINSCHLIESSLICH KOMPENSATORISCHER, SPEZIELLER, BEILÄUFIGER, EXEMPLARISCHER, STRAFENDER ODER FOLGENREICHER SCHÄDEN, VERANTWORTLICH, DIE SICH AUS DEM GEBRAUCH DIESER SOFTWARE DURCH DEN LIZENZNEHMER ERGEBEN.
7. DER LIZENZNEHMER IST BEREIT, DEN LIZENZGEBER ZU SCHÜTZEN UND ZU ENTSCHÄDIGEN UND FERN ZU HALTEN VON ALLEN ANSPRÜCHEN, VERLUSTEN, SCHÄDEN, BESCHWERDEN, ODER AUSGABEN, DIE MIT DEN GESCHÄFTSOPERATIONEN DES LIZENZNEHMERS VERBUNDEN SIND ODER SICH AUS DIESEN ERGEBEN.
8. DER LIZENZGEBER HAT DAS RECHT, DIESEN LIZENZVERTRAG SOFORT ZU KÜNDIGEN UND DAS SOFTWAREBENUTZUNGSRECHT DES LIZENZNEHMERS ZU BEGRENZEN, FALLS ES ZU EINEM VERTRAGSBRUCH SEITENS DES LIZENZNEHMERS KOMMT. DIE LAUFDAUER DES LIZENZVERTRAGS IST AUF UNBESTIMMTE ZEIT FESTGELEGT.
9. DER LIZENZNEHMER IST BEREIT, DEM LIZENZGEBER ALLE KOPIEN DER SOFTWARE BEI KÜNDIGUNG DES LIZENZVERTRAGS ZURÜCKZUGEBEN ODER ZU ZERSTÖREN.
10. DIESER LIZENZVERTRAG BEENDET UND ERSETZT ALLE VORHERIGEN VERHANDLUNGEN, VEREINBARUNGEN UND ABMACHUNGEN ZWISCHEN DEM LIZENZGEBER UND LIZENZNEHMER BEZÜGLICH DIESER SOFTWARE.
11. DIESER LIZENZVERTRAG UNTERLIEGT DEUTSCHEM RECHT.
12. WENN EINE BESTIMMUNG DIESES LIZENZVERTRAGS NICHTIG IST, WIRD DADURCH DIE GÜLTIGKEIT DER VERBLEIBENDEN BESTIMMUNGEN DIESES LIZENZVERTRAGS NICHT BERÜHRT. DIESE NICHTIGE BESTIMMUNG WIRD DURCH EINE GÜLTIGE, IN ÜBEREINSTIMMUNG MIT DEN GESETZLICHEN VORSCHRIFTEN STEHENDE BESTIMMUNG MIT ÄHNLICHER ABSICHT UND ÄHNLICHEN WIRTSCHAFTLICHEN AUSWIRKUNGEN ERSETZT.
13. DER LIZENZVERTRAG KOMMT DURCH ÜBERGABE DER SOFTWARE VON DEM LIZENZGEBER AN DEN LIZENZNEHMER UND/ODER DURCH DEN GEBRAUCH DER SOFTWARE DURCH DEN LIZENZNEHMER WIRKSAM ZUSTANDE. DIESER LIZENZVERTRAG IST AUCH OHNE DIE UNTERSCHRIFT DES LIZENZGEBERS GÜLTIG.
14. DIE LIZENZ ERLISCHT AUTOMATISCH, WENN DER LIZENZNEHMER DEN HIER BESCHRIEBENEN LIZENZBESTIMMUNGEN NICHT ZUSTIMMEN ODER GEGEN DIE LIZENZBESTIMMUNGEN DIESES LIZENZVERTRAGS VERSTOßEN. BEI BEENDIGUNG IST DER LIZENZNEHMER VERPFLICHTET, SOWOHL DIE SOFTWARE, ALS AUCH SÄMTLICHE KOPIEN DER SOFTWARE IN BEREITS INSTALLIERTER FORM ODER GESPEICHERT AUF EINEM DATENTRÄGER ZU LÖSCHEN, ZU VERNICHTEN ODER DER TELEMOTIVE AG ZURÜCK ZU GEBEN.
15. DER LIZENZNEHMER HAFTET FÜR ALLE SCHÄDEN, WELCHE DEM LIZENZGEBER DURCH DIE VERLETZUNG DIESES LIZENZVERTRAGS ENTSTEHEN.

## PRODUKTHAFTUNG

FÜR ALLE ANGEBOTE, VERKÄUFE UND LIEFERUNGEN GELTEN AUSSCHLIEßLICH DIE NACHSTEHENDEN BEDINGUNGEN UND ZWAR AUCH DANN, WENN DER KÄUFER, BESTELLER UND DERGLEICHEN ANDERE BEDINGUNGEN VORSCHREIBT. ABÄNDERUNGEN SIND NUR GÜLTIG, WENN SIE SCHRIFTLICH VEREINBART WERDEN.

1. DIE TECHNISCHE DOKUMENTATION IST BESTANDTEIL DES PRODUKTES. WERDEN DIE INHALTE UND INSBESONDERE DIE SICHERHEITSHINWEISE UND HANDLUNGSANLEITUNGEN DER DOKUMENTATION NICHT BEACHTET, KANN DIES DEN AUSSCHLUSS DER PRODUKTHAFTUNG UND DER PRODUKTGEWÄHRLEISTUNG ZUR FOLGE HABEN.
2. DIE PRODUKTE GEHÖREN ZUR GRUPPE DER TESTTOOLS. BEI EINSATZ DES GERÄTES KANN EINE STÖRUNG DES ZU TESTENDEN SYSTEMS NICHT 100% AUSGESCHLOSSEN WERDEN. DAMIT KANN DIE GARANTIE EINES EINWANDFREI FUNKTIONIERENDEN SYSTEMS NICHT VOM HERSTELLER ÜBERNOMMEN WERDEN.  
DER EINSATZ DES PRODUKTES ERFOLGT AUF EIGENE GEFAHR.
3. DIE HAFTUNG FÜR DEN ERSATZ VON SCHÄDEN GEMÄß §1 DES PRODUKTHAFTUNGSGESETZES, WIRD, IM RAHMEN DES §9 PHG AUSDRÜCKLICH AUSGESCHLOSSEN, SOWEIT ZWINGENDE GESETZLICHE BESTIMMUNGEN NICHTS ANDERES VORSEHEN.  
DER HERSTELLER LEHNT IN JEDEM FALL DIE VERANTWORTUNG FÜR INDIREKTE, BEILÄUFIGE, SPEZIELLE ODER FOLGENREICHE SCHÄDEN, EINSCHLIEßLICH DEM VERLUST VON GEWINN, VON EINNAHMEN, VON DATEN, DES GEBRAUCHS, JEDEM ANDEREM WIRTSCHAFTLICHEN VORTEILS ODER SCHÄDEN AUS ANSPRÜCHEN DRITTER GEGEN DEN KUNDEN, AB, DIE AUS DIESER ABMACHUNG , OB IN EINER HANDLUNG IM VERTRAG, STRENGER VERBINDLICHKEIT, KLAGBARES DELIKT (EINSCHLIEßLICH DER NACHLÄSSIGKEIT) ODER ANDEREN GESETZLICHEN ODER GERECHTEN THEORIEN ENTSTEHT.  
DIE BEWEISPFLICHT LIEGT BEIM KÄUFER.
4. DIE TELEMOTIVE AG GEWÄHRLEISTET DIE GESETZLICHE GARANTIE GEMÄß DEUTSCHEN RECHT.  
AUßER DEN GARANTIE, DIE AUSDRÜCKLICH IN DIESER VEREINBARUNG FESTGELEGT WORDEN SIND, WERDEN ALLE PRODUKTE "GELIEFERT, WIE VERTRAGLICH VEREINBART, SOWEIT DER KUNDE VOM HERSTELLER NICHT AUSDRÜCKLICH ZUSÄTZLICHE ODER IMPLIZIERTEN GARANTIE EMPFÄNGT. DER HERSTELLER DEMENTIERT HIERMIT AUSDRÜCKLICH IRGENDWELCHE UND ALLE WEITEREN GARANTIE IRGENDWEINER ART ODER NATUR BEZÜGLICH DER PRODUKTE, OB AUSDRÜCKLICH ODER STILLSCHWEIGEND, EINSCHLIEßLICH UNBESCHRÄNKT, JEDE GARANTIE DES TITELS, DER MARKTFÄHIGKEIT, DER QUALITÄT, DER GENAUIGKEIT ODER EIGNUNG ZU EINEM BESTIMMTEN ZWECK ODER ZUM ZWECK DES KUNDEN. DER HERSTELLER STREITET AUSDRÜCKLICH IRGENDWELCHE GARANTIE AB, DIE VOM HANDELSBRAUCH, DER HANDELSSITTE ODER DER LEISTUNG EINBEZOGEN WERDEN KÖNNEN. ABGESEHEN VON DEN FESTGESETZTEN AUSDRÜCKLICHEN GARANTIE IN DIESER ABMACHUNG, SIND DIE PRODUKTE MIT ALLEN FEHLERN UND DER VOLLSTÄNDIGEN GEFAHR EINER NICHT BEFRIEDIGENDEN QUALITÄT, LEISTUNG, GENAUIGKEIT BEREITGESTELLT. DER MÖGLICHE AUFWAND WIRD VOM KUNDEN GETRAGEN. DER HERSTELLER ÜBERNIMMT KEINE GARANTIE, DASS DIE PRODUKTE FEHLERFREI ARBEITEN.
5. DIE TELEMOTIVE AG IST BERECHTIGT, MANGELHAFTE WAREN GEGEN GLEICHARTIGE EINWANDFREIE WAREN INNERHALB EINER ANGEMESSENEN FRIST EINZUTAUSCHEN ODER DEN MANGEL INNERHALB EINER ANGEMESSENEN FRIST ZU BEHEBEN. BEI DIESEM FALL ERLISCHT EIN ANSPRUCH AUF WANDLUNG ODER PREISMINDERUNG. GEWÄHRLEISTUNGSRECHTE SETZEN EINE RECHTZEITIGE MÄNGELRÜGE VORAUS.
6. DER WEITERVERKAUF, DIE WEITERGABE, SCHENKUNG, TAUSCHGESCHÄFTE ODER DER VERLEIH DER ANGEBOTENEN PRODUKTE AN DRITTE, IST OHNE FREIGABE VON TELEMOTIVE NICHT GESTATTET.
7. ALS RECHTSGRUNDLAGE IST DEUTSCHES RECHT ANZUWENDEN.

# 1 Allgemeines

Die SimBox (Abbildung 1) ist eine CAN-Simulationsbox, die CAN-Nachrichten und einfache elektronische Steuergeräte wie Zündung, elektrische Fensterheber, Dämmerungssensor, Geschwindigkeitssignal, etc. simuliert, sowie eine einfache Lösung für die Restbus-Simulation darstellt.

Mit der SimBox spart der Anwender beträchtliche Kosten, welche in der Entwicklungsphase durch teure Prototypen von Steuergeräten, überdimensionierten Geräten oder teurer Software verursacht werden. Mit der SimBox können solche einfachen CAN-Nachrichten wesentlich günstiger simuliert werden: Insgesamt stellt die SimBox damit eine flexible und kosteneffiziente Alternative für die meisten Anwendungen dar.



Abbildung 1: SimBox

## Merkmale:

### Display

(4 Zeilen à 20 Zeichen) um Status- oder CAN-Nachrichten anzuzeigen

### Folientastatur

24 frei belegbare Tasten pro Konfiguration (die Konfiguration / Tastenbelegung erfolgt durch den Kunden und ist nicht Aufgabe der Telemotive AG).

### Start-Taste

Taste zum Starten der SimBox oder Öffnen von Menüs

### Powermanagement

Wecksignal über die Start-Taste oder über den CAN-Bus; geringe Ruhestromaufnahme im Schlafmodus

### 2 x CAN

Wahlmöglichkeit zwischen High Speed und Low Speed CAN

### RS 232

Konfigurationsschnittstelle

### Einfacher funktionaler Client zur Konfiguration

Erstellen und Editieren von Konfigurationen.

Laden von Konfigurationen und Firmware-Updates auf die SimBox

**Funktionen**

Tastengesteuertes oder mit Timern (Timer  $\geq 10\text{ms}$  möglich) verknüpftes Senden von CAN Nachrichten.

Einbeziehung von Offset Werten, um CAN Nachrichten umzurechnen.

Einschränkung: Überspringen von Werten (z. B. Gang 1 direkt in Gang 3 schalten)



## 2 Anschluss der SimBox

Die Schnittstellen der SimBox sind in Abbildung 2 dargestellt.

An der rechten Seite befindet sich ein 26-poliger SUB-D-Stecker. Hier werden alle CAN Signale und die Versorgung nach außen geführt. Der in Abbildung 3 gezeigte Kabelsatz SimBox gehört zum Lieferumfang.

Über eine SUB-D-Buchse an der linken Seite (RS232 Port) kann über ein entsprechendes Kabel ein PC oder Laptop angeschlossen werden (Abbildung 3).

Updates und Konfigurationen werden über diese Schnittstelle eingespielt. Die nötige Software dazu liegt dem Gerät bei.

Das Gerät verfügt über eine Feinsicherung (1 Ampere).



Abbildung 2: SimBox Schnittstellen

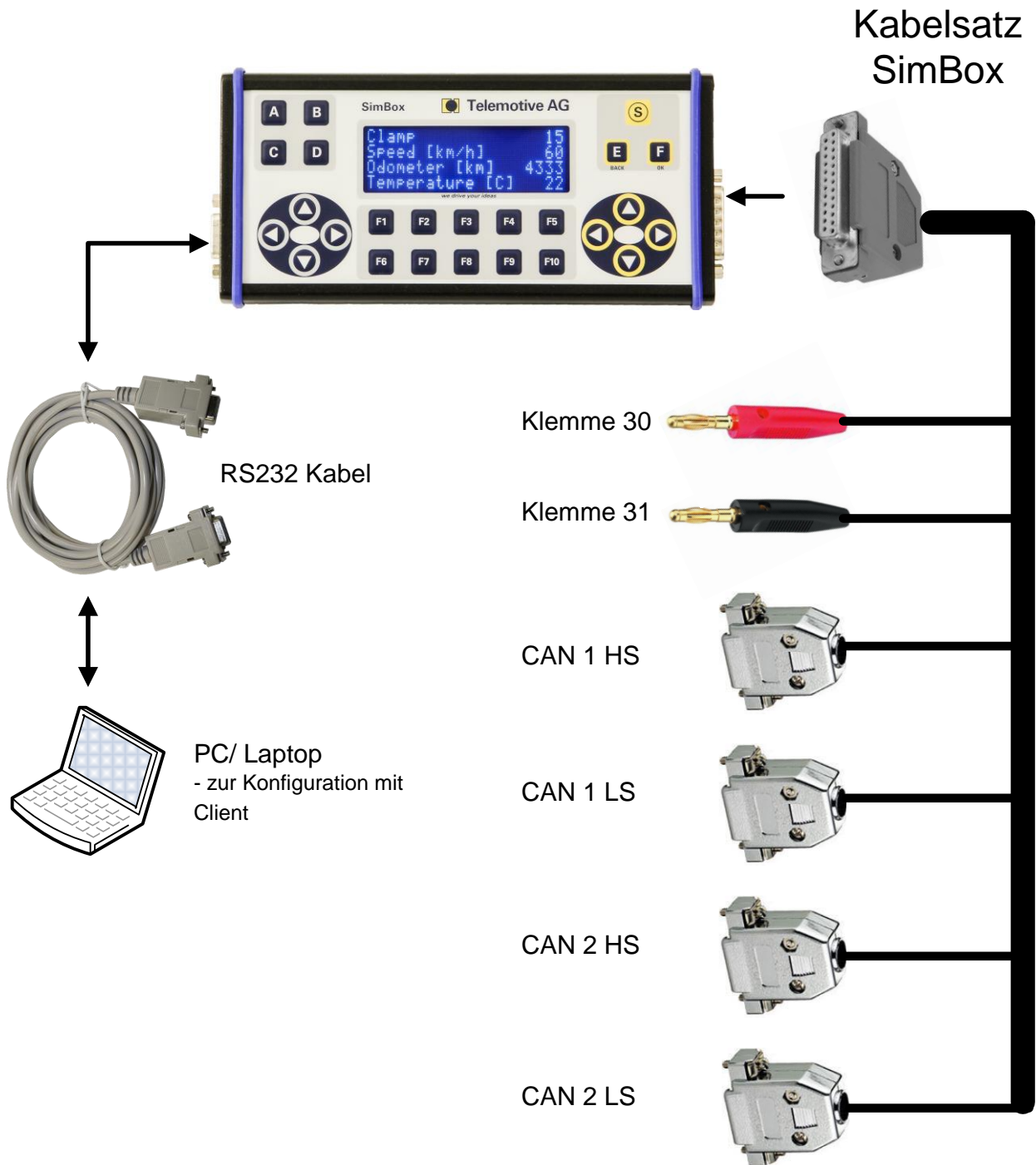


Abbildung 3: SimBox Anschluss

## 2.1 Energieversorgung

Die SimBox wird mit einer Nennspannung von 12V DC betrieben. Das Gerät arbeitet auch bei Schwankungen im Bereich von 8,5V DC bis 16V DC.

Eine Feinsicherung (1 Ampere) befindet sich an der rechten Gehäusesseite.

## 2.2 CAN

Die SimBox stellt zwei CAN 2.0B Schnittstellen zur Verfügung. Beide sind unabhängig voneinander wahlweise als High-Speed- oder Low-Speed-CAN konfigurierbar.

High- und Low-Speed-Transceiver haben getrennte Anschlüsse und Abschlusswiderstände, um zusätzliche Belastungen in bestehenden CAN Netzwerken zu vermeiden.

Die CAN Schnittstellen ermöglichen ein Wake Up der SimBox jeweils über den gewählten Transceiver (High-Speed / Low-Speed). Bei Erkennen von Busaktivität schaltet sich die SimBox automatisch ein.

## 2.3 Bedienelemente

Abbildung 4 zeigt die Tastatur und das Display der SimBox.

Es besteht aus 25 Tasten und einem Display mit 4 Zeilen und jeweils 20 Zeichen.

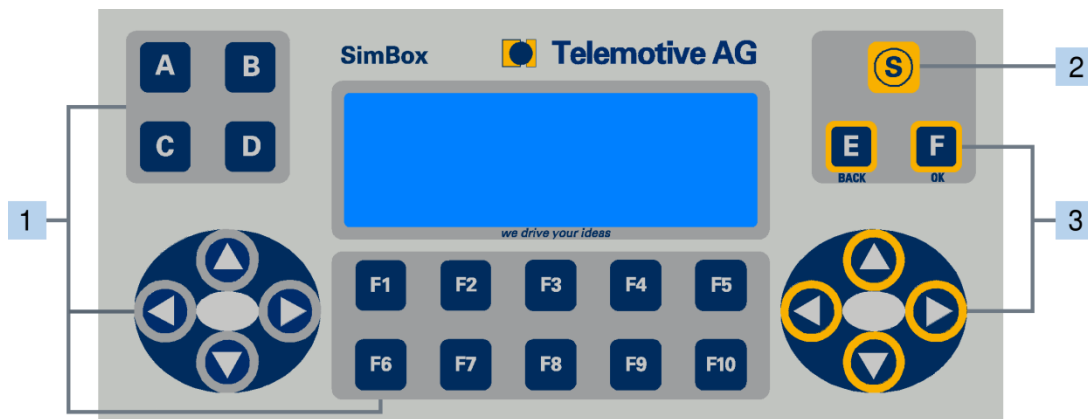


Abbildung 4: SimBox Tastatur & Display

- (1) Frei konfigurierbare Tasten
- (2) Start / Setup: Einschalten der SimBox, Aufruf von Systemfunktionen
- (3) Frei konfigurierbare Tasten, welche im Systemmodus temporär eine andere Bedeutung erhalten

## 2.4 Aktivierung und Standby

Die SimBox kann manuell durch Betätigen der Start/Setup-Taste gestartet werden. Wird auf einem der CAN-Kanäle Aktivität verzeichnet, startet das Gerät ebenfalls.

Solange die SimBox mit Spannung versorgt wird, merkt sie sich den zuletzt aktivierten Transceiver. Es kann auf den beiden CAN-Kanälen immer nur auf jeweils einem der Transceiver nach Aktivität geschaut werden. Wenn der High-Speed-CAN aktiviert war, kann das Gerät auch nur bei Aktivität auf dem High-Speed-CAN starten. Der Low-Speed-CAN wird dann nicht überwacht.

Bei verschiedenen Gelegenheiten kann sich die SimBox in einen Sleep Mode mit geringer Ruhestromaufnahme versetzen. Dies geschieht z. B. nach einem Update der Konfiguration oder der Firmware. Falls die SimBox an einem CAN-Bus mit laufender Busaktivität angeschlossen ist und auch der passende Transceiver in der SimBox aktiv war, wird sie nach dem Einschlafen durch eine CAN-Nachricht sofort wieder geweckt und muss nicht manuell gestartet werden.

Während des Starts zeigt die SimBox für kurze Zeit einen Startbildschirm (Abbildung 5), gefolgt von einem Disclaimer (Abbildung 6), der etwa drei Sekunden angezeigt wird.

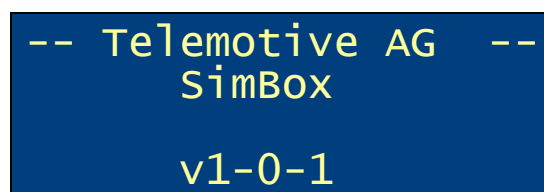


Abbildung 5: Startbildschirm



ONLY USE THIS DEVICE  
IF SAFE TO DO SO

Abbildung 6: Disclaimer

## 2.5 Standard-Betriebsart

Während des Starts lädt das Gerät eine Konfiguration aus dem Flash-Speicher. Diese Konfiguration kann vom Anwender vorgegeben werden, die Funktionen der SimBox und die Displayanzeige hängen von dieser Konfiguration ab.

## 2.6 Sondertasten

Alle Tasten der Tastatur können frei konfiguriert werden. Die Taste "S" (Start/Setup) ist hier ausgenommen. Sie dient zum Starten und zur Aktivierung des Systemmenüs.

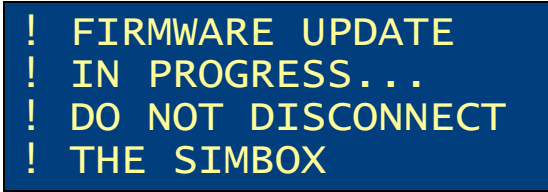
Bei aktivem Systemmenü sind alle gelb umrandeten Tasten für die Navigation durch das Menü reserviert. Die hinter diesen Tasten konfigurierte Funktion ist bei aktivem Systemmenü nicht aktiv. Wird das Systemmenü verlassen, bekommen diese Tasten die konfigurierte Funktion zurück.

Hinweis: Die Tastenfunktionen im Systemmodus sind abhängig von der verwendeten Firmwareversion.

## 2.7 Firmware-Update

Wenn eine neue Firmware verfügbar ist, kann diese vom Anwender selbst aufgespielt werden. Siehe hierzu Kapitel 3.5 Firmware-Update.

**Achtung! Die SimBox darf während des Updatevorgangs weder von der Spannungsversorgung, noch vom PC getrennt werden!**



! FIRMWARE UPDATE  
! IN PROGRESS...  
! DO NOT DISCONNECT  
! THE SIMBOX

Abbildung 7: Firmware-Update

## 3 SimBox-Client

Die SimBox kann vom Anwender mit Hilfe einer Client-Applikation konfiguriert und für verschiedene Einsatzzwecke angepasst werden. Die Client-Applikation kann auf Computern mit Microsoft Windows installiert werden.

### 3.1 Installation des Clients

Der SimBox-Client wird in Form eines Installationspakets mit integriertem Setup-Programm zur Verfügung gestellt. Zur Installation des Clients starten Sie einfach das Installationsprogramm und folgen den Anweisungen auf dem Bildschirm. Zur Installation benötigen Sie ggf. Administrator-Rechte.

Das Installationsprogramm kopiert den Client zusammen mit einem Deinstallationsprogramm und den benötigten DLLs auf den Computer und erstellt einen Eintrag im Startmenü.

Zum Entfernen des Client starten Sie das Deinstallationsprogramm. Alternativ kann das Verzeichnis mit dem Client und der Startmenüeintrag manuell gelöscht werden.

### 3.2 Startfenster

Nach dem Start des SimBox-Clients erscheint folgendes Startfenster:

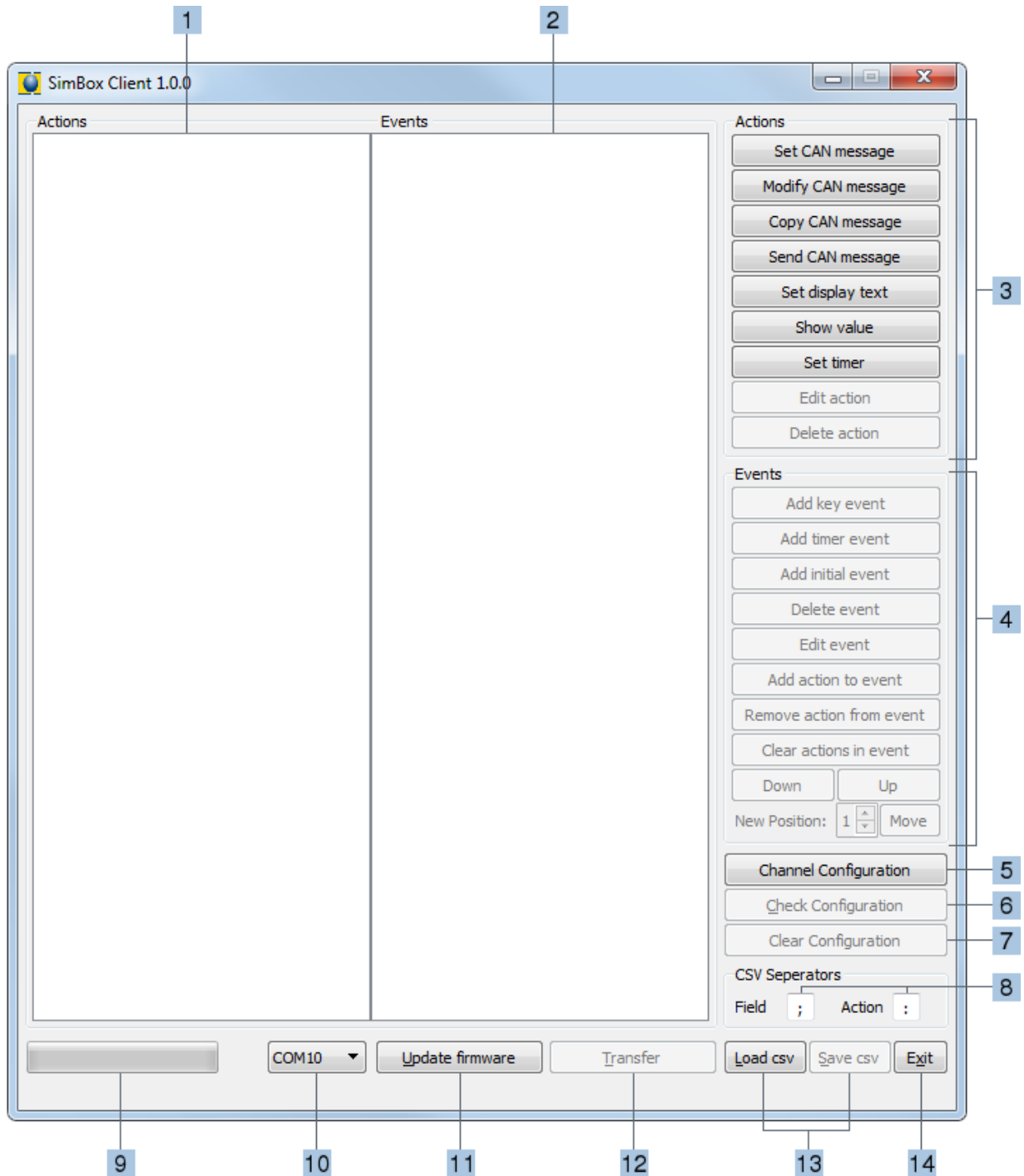


Abbildung 8: SimBox Client Startfenster

- (1) Liste der konfigurierten Aktionen
- (2) Liste der konfigurierten Ereignisse
- (3) Erstellen und Bearbeiten von Aktionen
- (4) Erstellen und Bearbeiten von Events
- (5) Konfiguration der CAN-Schnittstellen sowie des Konfigurationsnamens
- (6) Prüfen der geladenen Konfiguration
- (7) Erstellen einer leeren Konfiguration
- (8) Einstellen der Parameter der CSV-Datei
- (9) Fortschrittsanzeige bei Konfigurations- und Firmwaredownload
- (10) Auswahl der COM-Schnittstelle
- (11) Starten des Firmware-Updates
- (12) Senden der Konfiguration zur SimBox
- (13) Laden und Speichern einer Konfigurationsdatei
- (14) Beenden des Programms

Der Client ist grob aufgeteilt in zwei Listen, von denen die linke Liste die konfigurierten Aktionen und die rechte List die Ereignisse anzeigt. Rechts befinden sich Kontrolltasten: Die Gruppenbox ‚Actions‘ erlaubt das Erzeugen, Manipulieren und Löschen von Aktionen. Darunter befindet sich die Gruppenbox ‚Events‘, die das Erzeugen, Manipulieren und Löschen von Ereignissen erlaubt. In der unteren Leiste des Clients sitzen die Tasten, welche es dem Anwender ermöglichen, eine Konfiguration zu laden, zu speichern, an die SimBox zu senden sowie ein Firmware-Update durchzuführen.

### 3.3 Konfiguration laden

Durch Klick auf ‚Load CSV‘ (Abbildung 8, Markierung 13) kann eine bereits bestehende Konfiguration geladen werden. Es erscheint ein Dialogfenster zur Auswahl des Verzeichnisses und des Dateinamens. Die Konfigurationsdateien müssen im CSV-Format vorliegen. Sollte im Client zu diesem Zeitpunkt bereits eine andere Konfiguration vorhanden sein, so wird explizit nachgefragt, ob die neue Datei der alten Konfiguration hinzugefügt, die alte Konfiguration durch die neue ersetzt oder der Ladevorgang abgebrochen werden soll.

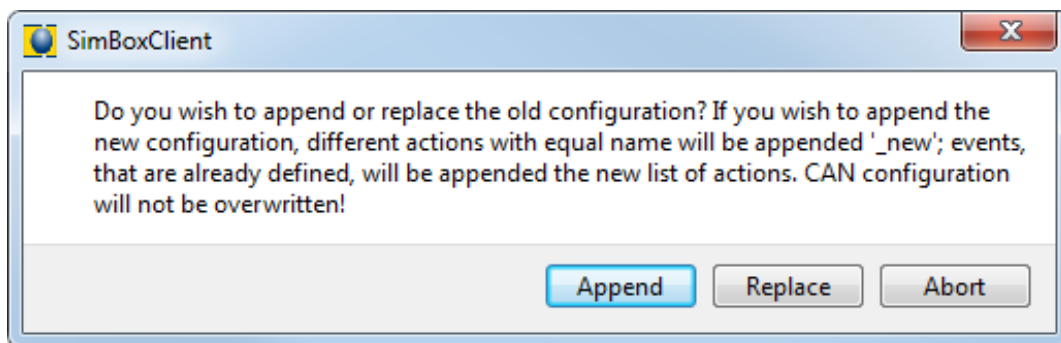


Abbildung 9: Auswahl beim Laden einer CSV-Datei.

Beim Anhängen einer neuen Konfiguration werden die neuen Daten farblich markiert. Ist beim Einfügen einer Aktion bereits eine Aktion gleichen Namens vorhanden, wird sie nur hinzugefügt, wenn die beiden Aktion nicht identisch sind. Um Konflikte bei der Benennung der Aktionen zu vermeiden, wird der neuen Aktion in diesem Fall die Endung ‚\_new‘ angehängt. In der Darstellung des Clients werden die neu hinzugekommenen Aktionen farblich hervorgehoben (sh. Abbildung 10).

Ist ein hinzuzufügendes Ereignis bereits in der alten Konfiguration definiert, werden die Aktionslisten der beiden Ereignisse verglichen. Sind sie identisch, wird der alte Eintrag beibehalten. Unterscheiden sich die Aktionslisten, wird die Liste des Ereignisses aus der neuen Konfiguration an die Aktionsliste des bestehenden Ereignisses angehängt. In der Darstellung des Clients werden in den Aktionslisten der Ereignissen alle in der hinzugefügten Konfiguration verknüpften Aktionen farblich hervorgehoben.

Beim Anhängen einer neuen Konfiguration an eine bereits geladene wird die Konfiguration der CAN-Kanäle nicht verändert. Nach dem erfolgreichen Anhängern der weiteren Konfiguration müssen Sie ggf. mögliche Konflikte zu lösen, bevor Sie die Konfiguration verwenden können.

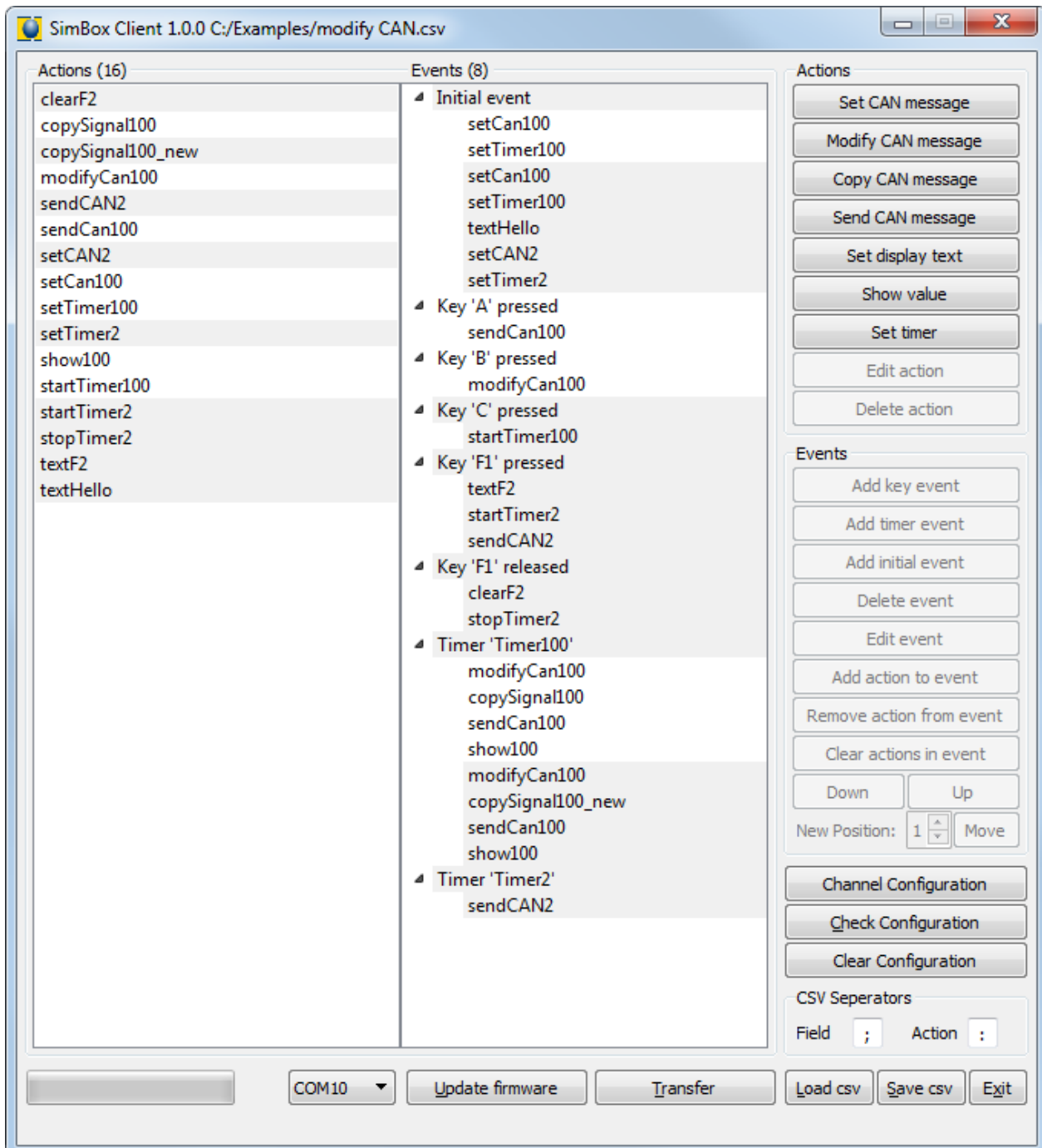


Abbildung 10: Client nach dem Anfügen einer weiteren Konfiguration

Nach dem Laden einer Konfiguration stellt der Client die erkannten Datensätze in den beiden Listen für Aktionen und Ereignisse dar. Wurden in der Datei Fehler erkannt, werden diese in einem eigenen Fenster dargestellt (Abbildung 11). Angaben von Zeilennummern (,Line') beziehen sich auf die Zeilen der CSV-Datei.



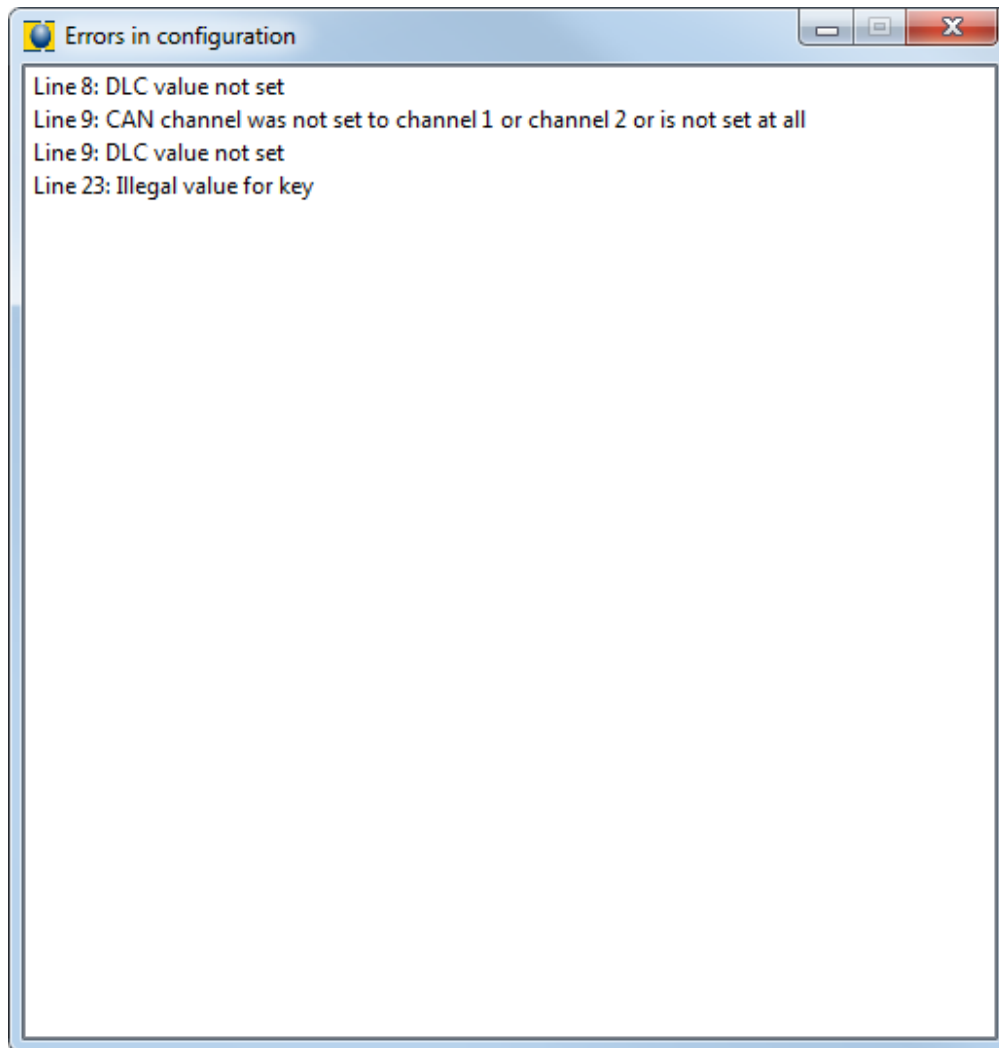


Abbildung 11: Fehlerhinweise beim Laden einer Konfiguration

Werte, die vom Client nicht erkannt werden konnten, werden mit Standardwerten belegt. Bitte berücksichtigen Sie, dass eine fehlerhafte Konfiguration in der SimBox ein unerwartetes Verhalten zeigen kann.

### 3.4 Konfiguration übertragen

Nach Auswahl der COM-Schnittstelle (Abbildung 8, Markierung 10) wird die Konfiguration durch Klick auf die Taste ‚Transfer‘ (Markierung 5) an die SimBox geschickt. Der Client bereitet die Konfiguration zur Speicherung in der SimBox auf und schreibt sie in deren Flash-Speicher.

Die SimBox muss vor dem Start des Flash-Vorgangs in den Update-Modus versetzt werden, z. B. durch gleichzeitige Betätigung der Taste ‚S‘ und der Pfeiltaste ‚Unten‘ im rechten Steuerfeld (Abbildung 12, sh. auch Kapitel 9).



Abbildung 12: Tasten zum Einstellen des Update-Modus

Es ist nicht möglich eine leere Konfiguration an die SimBox zu senden, ggfs. deaktiviert der Client die Taste ‚Transfer‘. Bei der Aufbereitung der Daten erzeugt der Client verschiedene Zwischendateien. Diese werden automatisch unter Verwendung des Dateinamens der geladenen Konfiguration mit geänderter Erweiterung (.dat, .hex) im selben Ordner gespeichert; sollte keine Datei geladen worden sein, werden die Dateien im Programmverzeichnis gespeichert.

Hinweis: Mit dem Start der Übertragung wird der aktuelle Stand der Konfiguration in ein Zwischenformat konvertiert, und dieser Schnappschuss wird an die SimBox übertragen. An der Konfiguration im Client kann zwar gearbeitet werden, dann ist jedoch ein Wiederholen der Übertragung mit der gleichen Konfiguration wie im ersten Versuch nicht mehr möglich. Bitte speichern Sie vor dem Start der Übertragung Ihre Konfiguration und vermeiden Sie es während einer laufenden Übertragung, am Client Bedienschritte vorzunehmen.

### 3.5 Firmware-Update

Um eine neue Firmware auf die SimBox zu laden, wählt man ‚Update firmware‘ im Client aus (Abbildung 8, Markierung 11). Mit Klick auf diese Taste öffnet sich ein Dateibrowser, welcher Dateien im Hex-Format anzeigt. Auch diesmal wird der Anwender aufgefordert, die Taste ‚S‘ und die Pfeiltaste ‚Unten‘ (Abbildung 12) zu drücken, um die SimBox in den Update-Modus zu setzen.

Weitere Informationen zur Aktivierung des Update-Modus finden Sie in Kapitel 9.

Hinweis: Es ist wichtig, dass die Firmware fehlerfrei an die SimBox übertragen werden kann. Bitte vermeiden Sie es, während der Datenübertragung Bedienschritte am Client vorzunehmen, und starten Sie keine weiteren Programme, welche Ihren Rechner zu sehr auslasten könnten.

### 3.6 Konfiguration speichern

Um eine Konfiguration zu speichern, klickt man auf ‚Save CSV‘ (Abbildung 8, Markierung 13), um einen Dialog zur Auswahl einer Datei zu öffnen. Wenn eine Datei mit dem gewählten Namen bereits existiert, muss das Überschreiben dieser Datei bestätigt werden. Bei einer leeren Konfiguration ist das Speichern nicht möglich, weswegen der Client die entsprechende Taste ggfs. deaktiviert.

### 3.7 Alle Elemente löschen

Um die gesamte Konfiguration aus dem Client zu löschen, klickt man ‚Clear Configuration‘ (Abbildung 8, Markierung 7).

### 3.8 Konfiguration prüfen

Es besteht die Möglichkeit, eine geladene Konfiguration auf Fehler zu überprüfen mittels Klick auf ‚Check configuration‘ (Abbildung 8). Es werden allgemeine Einstellungen geprüft wie CAN-Kanalnamen, eindeutige Namen bei Aktionen und Ereignissen, etc.. Diese Prüfung soll als Hilfe bei der Erstellung einer Konfiguration dienen und erhebt keinen Anspruch auf Vollständigkeit. Insbesondere kann sie nicht die korrekte Umsetzung einer gewünschten Funktionalität überprüfen, sondern nur formale Eigenschaften testen. Wenn dieser Konfigurationscheck ohne Fehlermeldungen beendet wird, garantiert dies also noch nicht den korrekten Ablauf der Konfiguration auf der SimBox.

### 3.9 Setzen der Trennzeichen bei der CSV-Datei

Der SimBox-Client verwendet in den CSV-Dateien, in welchen die Konfigurationen gespeichert werden können, zwei verschiedene Trennzeichen, einen Feldtrenner und ein Trennzeichen für die Aktionslisten. Die beiden Trennzeichen können bei Bedarf vom Anwender verändert werden, indem die gewünschten Zeichen in den Feldern ‚Field‘ und ‚Action‘ eingetragen werden (Abbildung 8, Markierung 8). Das Setzen dieser Zeichen sollte Experten vorbehalten bleiben, da bei inkorrekt eingetragenen Zeichen eine CSV-Datei weder korrekt gelesen noch geschrieben werden kann! Die Trennzeichen dürfen nicht anderweitig in der Konfiguration verwendet werden.

Der Feldtrenner entspricht dem Feldtrenner des CSV-Formats. Er wird von verschiedenen Applikationen, welche CSV-Dateien bearbeiten können, in Abhängigkeit von den Regionaleinstellungen unterschiedlich gehandhabt. Oft wird ein Komma verwendet, in Europa jedoch auch gerne ein Semikolon.

Der Aktionstrenner dient zur Abgrenzung der einzelnen Aktionen innerhalb der Aktionsliste, welche einem Ereignis zugeordnet ist. Hier wird standardmäßig ein Doppelpunkt verwendet.

### 3.10 Programm beenden

Um das Programm zu beenden, wählt man ‚Exit‘ in der grafischen Oberfläche aus (Abbildung 8, Markierung 14).

Befindet sich eine veränderte Konfiguration im Client, erhält der Anwender vor dem Schließen die Möglichkeit, diese Konfiguration zu speichern.

## 4 Konfiguration

Die SimBox arbeitet ereignisgesteuert. Ein Ereignis kann beispielsweise ein Tastendruck durch einen Anwender sein. Auf dieses Ereignis hin arbeitet die SimBox eine Folge von Aktionen ab, welche für dieses Ereignis konfiguriert sind. Diese Aktionen können das Versenden einer CAN-Nachricht und das Anzeigen einer zugehörigen Meldung auf dem Bildschirm sein. In den nachfolgenden Kapiteln werden die Ereignisse und Aktionen beschrieben.

Neben den Tastaturereignissen, welche ein Anwender auslösen muss, bietet die SimBox konfigurierbare Timer an. Damit können z. B. CAN-Nachrichten ohne weitere Bedienung an der SimBox zyklisch versendet werden.

Die Aktionen, welche in der SimBox konfiguriert werden können, bilden meist Einzelschritte eines Vorgangs ab, welche durch unterschiedliches Kombinieren ihren Einsatzbereich vergrößern. Für CAN-Nachrichten beispielsweise sind das Definieren des Nachrichteninhalts und das Versenden der Nachricht getrennte Aktionen. Dies ermöglicht es, den Nachrichteninhalt einmal zu definieren, und dann vor jedem Versenden durch eine weitere Aktion einen Sequenzzähler in dieser Nachricht zu erhöhen.

Beim Erstellen einer neuen Konfiguration empfiehlt es sich, zuerst die benötigten Objekte für diese Konfiguration zu ermitteln. Erst in einem nächsten Schritt sollte der SimBox-Client gestartet und mit der eigentlichen Eingabe begonnen werden. Bei der Eingabe der einzelnen Konfigurationsschritte beginnt man mit dem Anlegen von Aktionen. Sind die Aktionen erstellt, können eine einzelne oder auch mehrere Aktionen markiert und als Aktionsliste mit einem Ereignis, etwa einem Tastendruck, verbunden werden. In Kapitel 7 finden Sie einige Konfigurationsbeispiele mit Erläuterungen zu deren Erstellung.

Die Aktionen und Ereignisse können auch nachträglich noch bearbeitet werden. Der SimBox-Client kann die Konfiguration als CSV-Datei speichern und lesen. Diese Dateien können bei Bedarf auch manuell bearbeitet oder von automatisierten Tools erzeugt werden.

Die SimBox kann über 3000 Einzelschritte, bzw. Zeilen in der CSV-Datei, speichern und verarbeiten. Damit können über 1500 verschiedene CAN-Nachrichten versandt werden, wenn für jede CAN-Nachricht jeweils das Festlegen des Nachrichteninhalts und das Versenden konfiguriert werden (zwei Aktionen). Andere Aktionen, z. B. das Anzeigen eines Texts im Display, belegen ebenfalls je einen Platz im Konfigurationsspeicher der SimBox.

### 4.1 Allgemeine Konfigurationseinstellungen

Die Einstellungen für die CAN-Kanäle und der Name der Konfiguration werden eingestellt mittels ‚CAN channel configuration‘ (Abbildung 8, Markierung 5). Im Feld ‚Config name‘ (Abbildung 13) kann ein Name für die Konfiguration vergeben werden, der sich an der SimBox abrufen lässt.

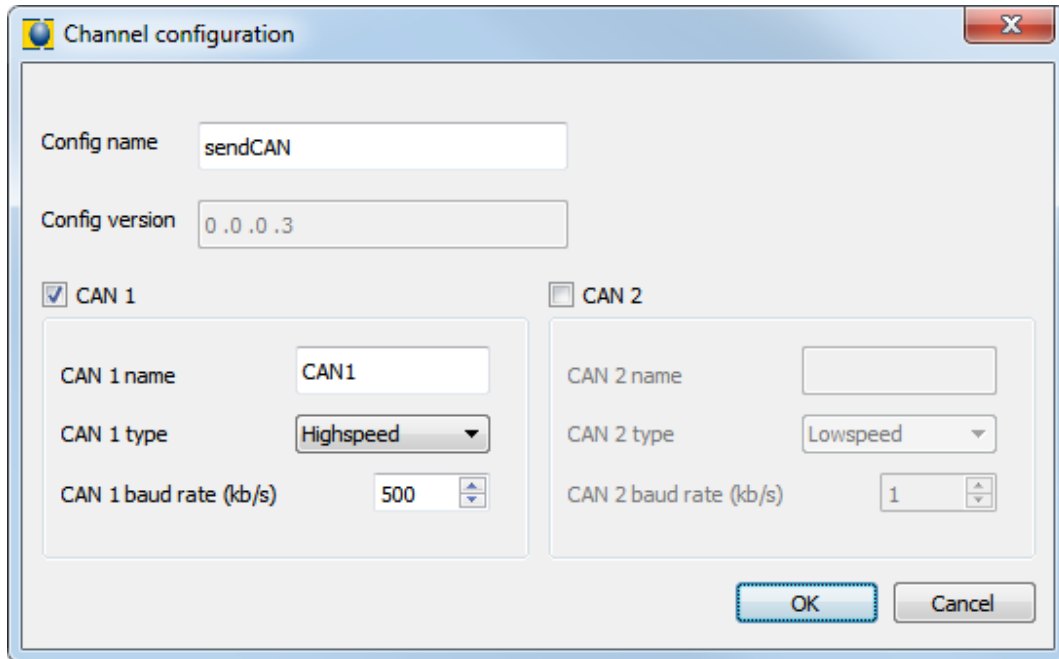


Abbildung 13: Konfigurationseinstellungen

Über die Auswahlboxen für die beiden CAN-Kanäle kann ein Kanal aktiviert oder deaktiviert werden. Für jeden Kanal kann ein Name vergeben, der Transceiver-Typ sowie die Baudrate eingestellt werden. Der Kanalname dient, wie der Name der Konfiguration, nur der Information und kann an der SimBox abgerufen werden.

Im Feld ‚Config version‘ wird eine Versionsnummer angezeigt, welche auch in der CSV-Konfigurationsdatei abgelegt ist. Anhand dieser Nummer kann der SimBox-Client beim Einlesen einer Datei erkennen, in welchem Format die Datensätze abgelegt sind und welche Eigenschaften unterstützt werden. Beim Schreiben der Konfiguration verwendet der Client eine feste Versionsangabe.

## 5 Aktionen

Ausgelöst durch ein Ereignis, z. B. ein Tastendruck auf der Tastatur, kann eine Aktion ausgeführt werden.

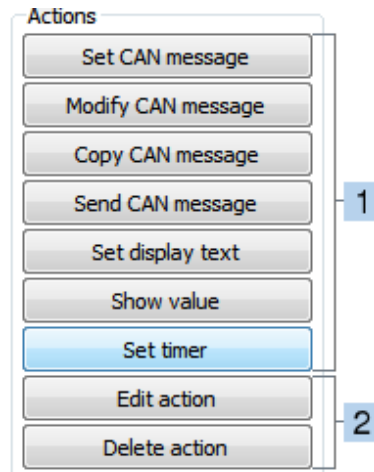


Abbildung 14: Verwalten der Aktionsliste

- (1) Anlegen von neuen Aktionen
- (2) Bearbeiten und Löschen von bereits existierenden Aktionen

Folgende Aktionen können erstellt werden:

- CAN Botschaft setzen ‚Set CAN message‘
- CAN Botschaft verändern ‚Modify CAN message‘
- CAN Signalwert in Variable kopieren ‚Copy CAN message‘
- CAN Botschaft senden ‚Send CAN message‘
- Text setzen ‚Set display text‘
- Wert einer Variablen auf Display anzeigen ‚Show value‘
- Timer steuern ‚Set timer‘

Durch Klick auf einen der oben genannten Knöpfe öffnet sich ein Dialog, mit welchem eine Aktion dieses Typs erstellt werden kann. Am unteren Rand des Dialogs befindet sich ein Eingabefeld, mit welchem der Aktion ein eindeutiger Name zugeordnet werden muss (Abbildung 15, Markierung 1). Die Aktion wird nach dem Erstellen in der Aktionsliste des Hauptfensters mit diesem Namen angezeigt, ebenso wird dieser Name beim Verknüpfen der Aktion mit einem Ereignis (Tastendruck, Timer) verwendet.

Ohne einen Aktionsnamen ist das Klicken des Ok-Buttons nicht möglich. Sollte eine Aktion mit dem angegebenen Namen schon vorhanden sein, meldet der Client diesen Konflikt. Sie können dann einen anderen Namen wählen oder den Vorgang abbrechen.

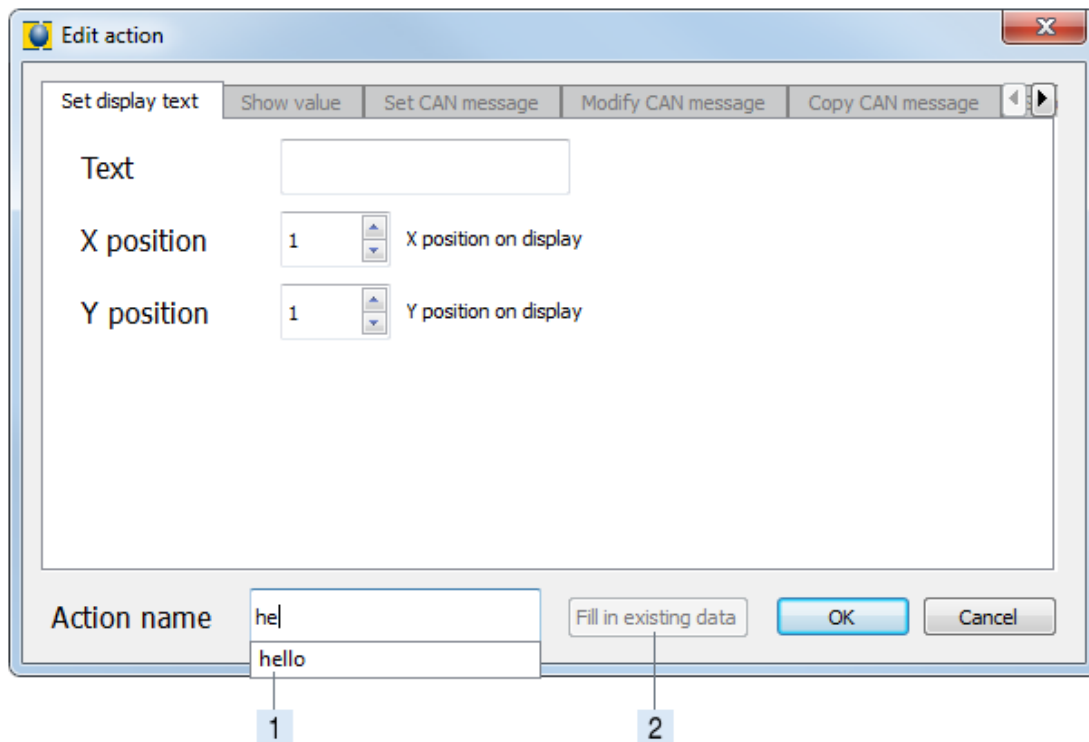


Abbildung 15: Aktionsname setzen

- (1) Name der Aktion
- (2) Übernehmen der Dateien einer bereits existierenden Aktion

Desweiteren ist es dem Anwender möglich, zur Erleichterung der Eingabe die Daten einer schon vorhandenen Aktion zu laden, indem er den schon vorhandenen Namen einträgt und dann auf den Button ‚Fill in existing data‘ (Abbildung 15, Markierung 2) klickt. Diese Taste ist ausgegraut, solange kein vorhandener Name eingetragen ist oder die existierende und die neu anzulegende Aktion nicht vom selben Typ sind.

## 5.1 Text setzen

Wird ‚Set display text‘ (sh. Abbildung 14) ausgewählt, kann ein beliebiger Text auf dem Display angezeigt werden. Es erscheint das Fenster, welches in Abbildung 15 dargestellt ist.

Nach Angabe des zu setzenden Textes werden die X- und Y-Position des Textes auf dem Display ausgewählt. Nach Klick auf den Button ‚OK‘ wird die Aktion in der Aktionsliste hinzugefügt.

Das Display besitzt 4 Zeilen zu je 20 Zeichen, der Y-Wert darf also zwischen 1 und 4 und der X-Wert zwischen 1 und 20 definiert werden. Der Wert 1 in den Feldern für X- und Y-Position beschreibt die linke obere Ecke des Displays.

## 5.2 Wert auf Display anzeigen

Um ein bestimmtes Signal aus einer CAN-Botschaft auf dem Display anzuzeigen, sind zwei Schritte notwendig. Als erstes wählt man ‚Copy CAN message‘ aus (sh. Abbildung 14), um den Wert des Signals in eine Variable in der SimBox zu speichern. Danach kann der Wert dieser Variablen in einem zweiten Schritt mit Hilfe von ‚Show Value‘ auf dem Display angezeigt werden. Im folgenden wird die Konfiguration der beiden nötigen Aktionen beschrieben.

### 5.2.1 Copy CAN signal

Ein CAN-Signal wird definiert durch die CAN-Nachricht, in welcher es transportiert wird, und die Bit-Positionen innerhalb des Datenbereichs dieser CAN-Nachricht, welche das CAN-Signal belegt. Diese Informationen müssen in der Eingabemaske in Abbildung 16 eingegeben werden.

Zum Anlegen der Aktion wählt man als erstes ‚Copy CAN message‘ aus (sh. Abbildung 14).

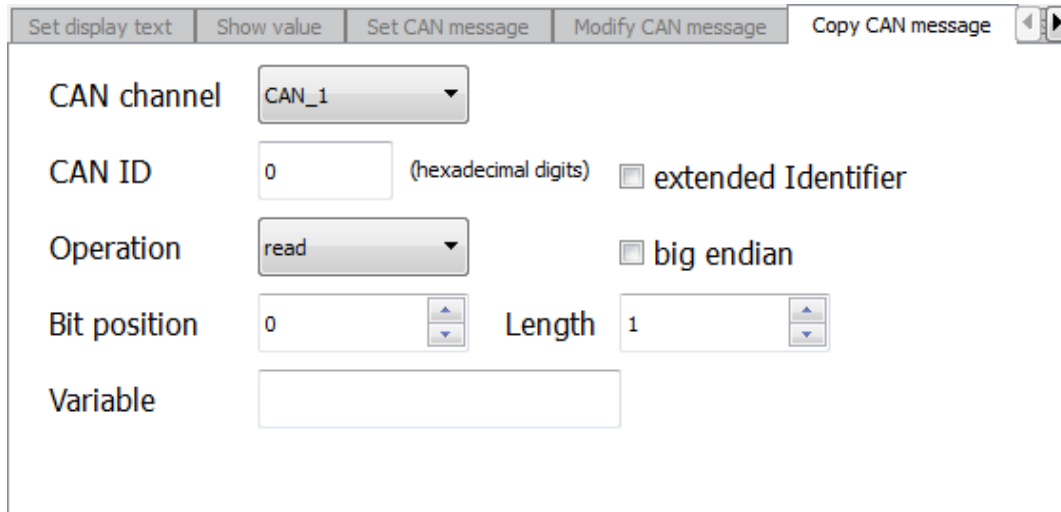


Abbildung 16: Copy CAN signal

Um ein CAN-Signal in einer Variable zu speichern, wählt man als ‚Operation‘ den Wert „read“ aus, definiert einen bestimmten Bereich der Botschaft mittels ‚Bit position‘ und ‚Length‘ aus und bestimmt einen Namen für die Variable und die Aktion.

Bei Auswahl von „read“ als Typ der Operation wird der Wert der ausgewählten Bits in der Variablen gespeichert, bei Auswahl von „write“ wird der Variableninhalt im CAN-Signal abgelegt. Das Kopieren des Werts erfolgt ohne Vorzeichenbetrachtung (unsigned).

Für Signale, welche in der CAN-Nachricht über mehrere Bytes gespeichert werden, bestimmt die Einstellung ‚big endian‘ die Reihenfolge der Bytes. Bei gesetztem Haken wird die Anordnung „big endian“ bzw. „Motorola“ verwendet, ohne Haken die Anordnung „little endian“ bzw. „Intel“.

Als Bitposition ist das LSB des CAN-Signals anzugeben. Dessen Position ist abhängig von der Einstellung des Kästchens ‚big endian‘. Liegt die Definition der Signale als Vector-DBC-Datei vor, können die Angaben aus dieser Datei eingetragen werden.

### 5.2.2 Show Value

Um den Wert anzuzeigen, wird nun ‚Show value‘ ausgewählt (sh. Abbildung 14).

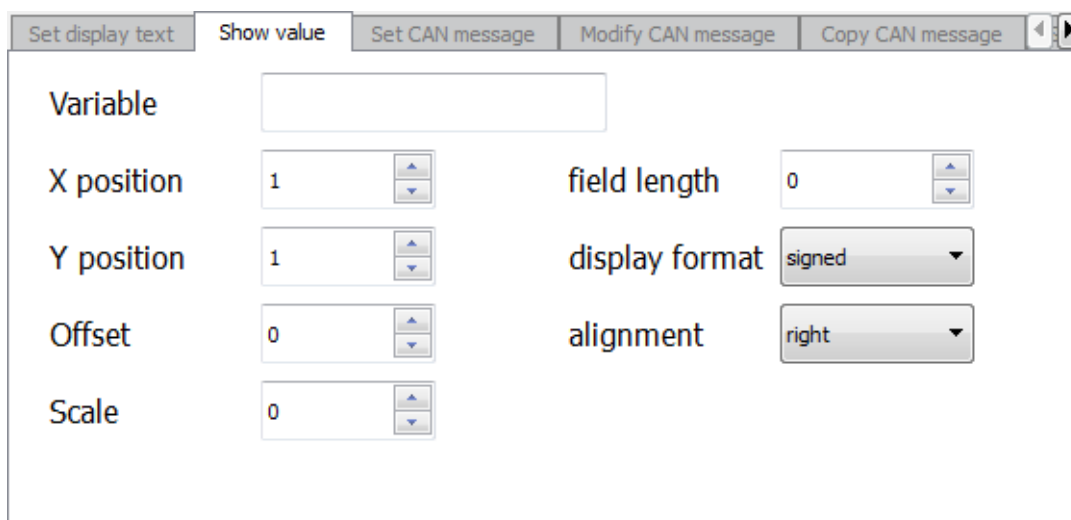


Abbildung 17: Show value



Unter ‚Variable‘ wird der Variablenname aus der zuvor erstellten Aktion eingegeben. ‚X position‘ und ‚Y position‘ legen fest, an welcher Position auf dem Display der Wert angezeigt wird (vgl. Kap. 5.1). ‚field length‘ gibt an, wie viele Stellen der Wert einnehmen soll.

Unter ‚display format‘ kann die Formatierung des Wertes verändert werden. Zur Auswahl stehen ‚signed‘ (mit Vorzeichen), ‚unsigned‘ (ohne Vorzeichen) oder ‚hex‘ (hexadezimale Darstellung).

Unter ‚alignment‘ kann die Ausrichtung des Wertes (links oder rechtsbündig) eingestellt werden. Mit den Feldern ‚Offset‘ und ‚Scale‘ kann der Botschaftswert umgerechnet werden.

Für positive Werte von ‚Scale‘ gilt:

$$\text{angezeigter Wert} = (\text{Botschaftswert} - \text{Offset}) \cdot \text{Scale}$$

Für negative Werte von ‚Scale‘ gilt:

$$\text{angezeigter Wert} = (\text{Botschaftswert} - \text{Offset}) / (-\text{Scale})$$

Hinweis: Wenn der anzuzeigende Wert mehr Zeichen benötigt als in ‚field length‘ angegeben, werden auch die Zeichen in das Display geschrieben, welche über das Feld hinausragen.

### 5.3 Botschaft erstellen

Bevor die SimBox eine CAN-Nachricht versenden kann, muss deren Inhalt definiert werden. Um eine entsprechende Aktion zu erstellen, wird ‚Set CAN message‘ ausgewählt (sh. Abbildung 14). Es erscheint folgendes Fenster:

Abbildung 18: Set CAN message

Unter ‚Channel‘ wird ausgewählt, auf welchem Kanal die Botschaft gesendet werden soll. Unter ‚DLC‘ wird die Länge der Nutzdaten in Bytes eingestellt.

Unter ‚CAN ID‘ wird der Identifier (in hexadezimaler Schreibweise) festgelegt. Soll die Nachricht mit einem Extended-Identifier mit 29 Bit Länge versendet werden, aktiviert man ‚extended Identifier‘, andernfalls werden Standard-Identifier mit 11 Bit Länge verwendet.

Der Dateninhalt wird unter ‚CAN Data‘ in hexadezimaler Schreibweise eingegeben. Bei der Zuweisung wird der Dateninhalt mit ‚CAN Mask‘ verknüpft. Nur diejenigen Bits in der CAN-Botschaft werden verändert, welche in der Maske mit 1 markiert sind, alle anderen bleiben unverändert.

### 5.4 Botschaft senden

Soll eine Botschaft gesendet werden, wird ‚Send CAN message‘ ausgewählt (sh. Abbildung 14).

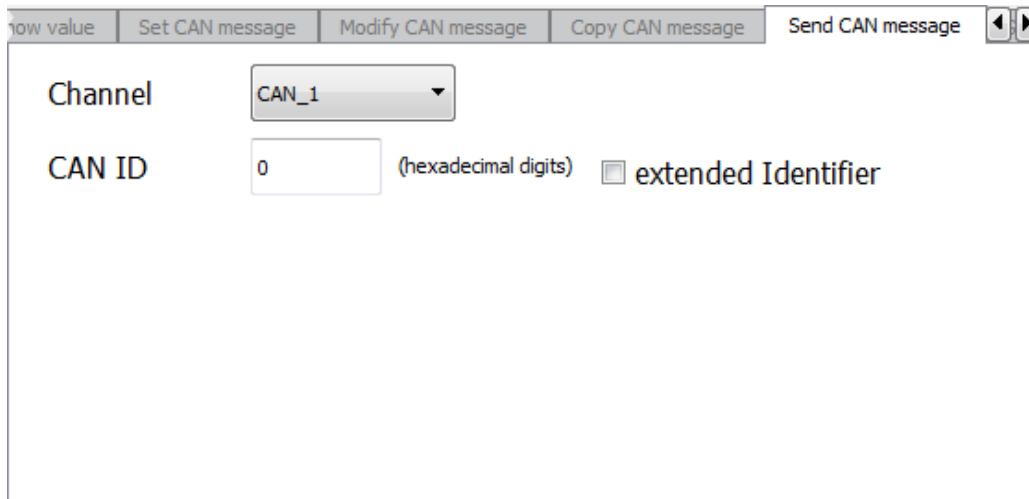


Abbildung 19: Send CAN message

Unter ‚Channel‘ wählt man den Kanal aus, unter ‚CAN ID‘ welche Botschaft gesendet werden soll. Danach wählt man einen Namen für die Aktion, und schließt den Dialog durch Klick auf ‚OK‘. Die Aktion wird dem Fenster ‚Action‘ hinzugefügt.

Vor dem Versenden einer CAN-Nachricht muss der Inhalt der Nachricht definiert werden, andernfalls können unerwartete Ergebnisse auftreten (sh. Kap 5.3). Die SimBox verwendet die CAN-ID und die Kanalnummer zur Identifizierung einer Nachricht.

Um eine zyklische Botschaft zu versenden, verwenden Sie bitte ein Timer-Ereignis (siehe Kapitel 5.6 und 6.3)

## 5.5 Signal einer CAN-Botschaft dynamisch verändern

Um eine Botschaft während des Betriebs der SimBox zu verändern, wählt man ‚Modify CAN message‘ aus (sh. Abbildung 14). In folgendem Fenster können die Parameter dieser Aktion verändert werden:

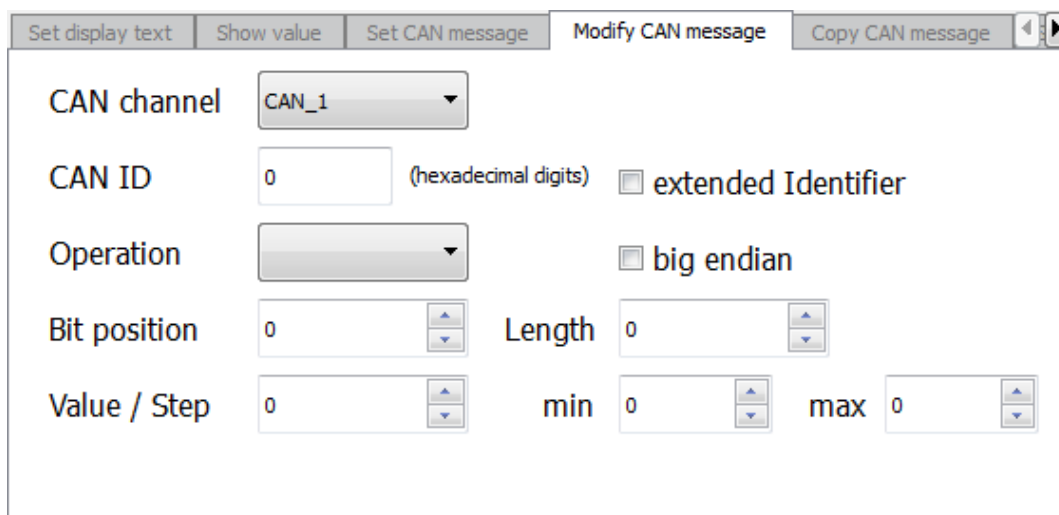


Abbildung 20: Modify CAN message

Unter ‚CAN channel‘ wählt man den CAN Kanal aus. Unter ‚CAN ID‘ wird der Identifier der zu beeinflussenden Botschaft in hexadezimaler Notation eingegeben. Unter ‚Operation‘ wird die Art der Änderung ausgewählt. Unter ‚Bit position‘ wird die Nummer des LSB (niederwertigstes Bit) der zu verändernden Bits festgelegt. Der Wert ‚Length‘ legt fest, wie viele Bitstellen verändert werden sollen. Standardmäßig wird die Bytefolge ‚Little Endian‘ („Intel“) verwendet, um dies zu ändern, wird ‚Big Endian‘ („Motorola“) aktiviert.

Unter ‚Value / Step‘ wird festgelegt, welcher Wert zugewiesen bzw. in welchen Schritten der Wert erhöht werden soll. Für die Operationen, welche den Wertebereich begrenzen, können die Grenzen in den Feldern ‚min‘ und ‚max‘ angegeben werden. Zum Deaktivieren der Grenzen werden ‚min‘ und ‚max‘ gleichzeitig auf 0 gesetzt.

Im folgenden werden die möglichen Einstellungen für ‚Operation‘ beschrieben.

### 5.5.1 Wert setzen

Um den Dateninhalt einer Botschaft zu setzen, wählt man ‚Set‘ unter ‚Operation‘ aus. Unter ‚Value‘ wird der zu setzende Wert eingetragen.

### 5.5.2 Wert erhöhen/vermindern (Add/Subtract)

Mit ‚Add‘ bzw. ‚Subtract‘ kann der Wert eines CAN-Signals erhöht oder vermindert werden. Die Grenzen ‚min‘ und ‚max‘ werden nicht berücksichtigt, bei Überlauf werden die obersten Bits (Bitpositionen außerhalb der angegebenen Länge) ignoriert.

### 5.5.3 Wert erhöhen/vermindern mit Begrenzung (AddLimit/SubtractLimit bzw. Increment/Decrement)

Wird unter ‚Operation‘ die Option ‚ADDLIMIT‘ oder ‚SUBTRACTLIMIT‘ ausgewählt, kann der Wert eines Signals erhöht oder vermindert werden. Bei Erreichen der Min- oder Max-Grenze wird der Wert nicht weiter verändert.

‚INCREMENT‘ oder ‚DECREMENT‘ sind Synonyme für ‚ADDLIMIT‘ und ‚SUBTRACTLIMIT‘.

### 5.5.4 Wert erhöhen/vermindern mit Überlauf (AddWrap/SubtractWrap)

Wird unter ‚Operation‘ ‚ADDWRAP‘ oder ‚SUBTRACTWRAP‘ ausgewählt, kann der Wert eines Signals erhöht oder vermindert werden. Bei Erreichen der Min- oder Max-Grenze tritt ein Überlauf auf das andere Ende des Wertebereichs auf (Wraparound).

## 5.6 Timer

Um eine Timer-Aktion zu erstellen, wird ‚Set timer‘ ausgewählt (sh. Abbildung 14).

Unter ‚Timer name‘ wird der Name des Timers gewählt. Der Name des Timers muss in der Konfiguration bereits bekannt sein, d. h. es muss bereits ein Timer-Ereignis mit dem Namen des gewünschten Timers angelegt sein. Es ist nicht möglich, eine Timer-Aktion anzulegen, ohne dass der anzusprechende Timer vorher in der Liste ‚Timer name‘ ausgewählt wurde.

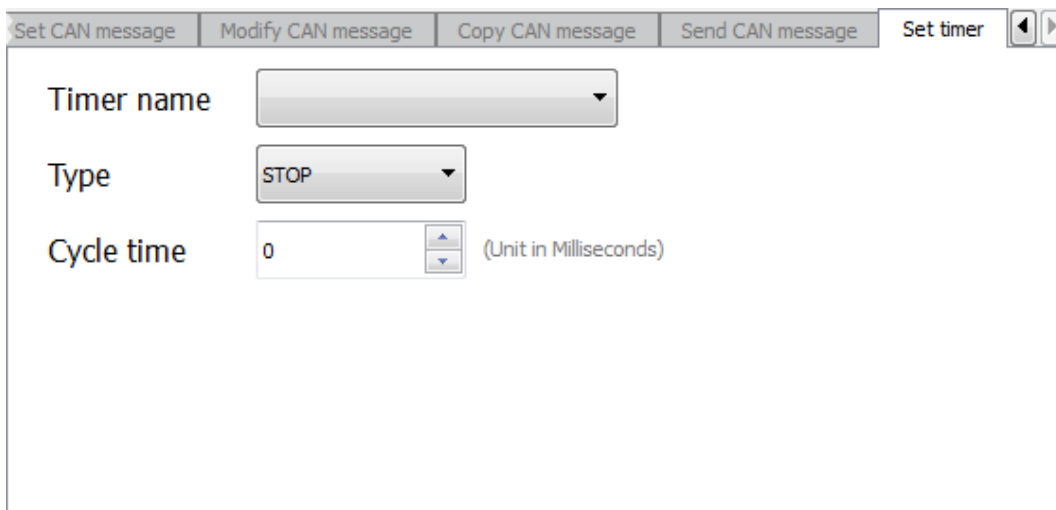


Abbildung 21: Timer modifizieren

Unter ‚Cycle time‘ wird die Zykluszeit in Millisekunden eingegeben. Bitte beachten Sie, dass die Timer in der SimBox in einem bestimmten Raster arbeiten (10 ms).

Unter ‚Type‘ wird die Art der Timer-Aktion ausgewählt. Vor der erstmaligen Verwendung eines Timers muss eine Aktion mit einem Typ „Set“, „Set Cyclic“ oder „Set Single“ aufgerufen werden, um die Zykluszeit und eine korrekte Betriebsart einzustellen.

### 5.6.1 Start

Startet den Timer und setzt dabei die abgelaufene Zeit des aktuellen Zyklus zurück.

### 5.6.2 Stop

Hält den Timer an. Timer-Ereignisse für diesen Timer werden nicht mehr ausgelöst.

### 5.6.3 Reset

Setzt die abgelaufene Zeit des Timers zurück. Der Timerzyklus beginnt von neuem.

Wenn ein Timer gestartet wurde, aber vor Ablauf seiner Zykluszeit mit dieser Aktion zurückgesetzt wird, dann beginnt seine Zykluszeit von neuem. Ein Timer-Ereignis wird durch das Rücksetzen nicht ausgelöst. Diese Aktion kann verwendet werden, um Watchdog-Timer oder Nachlaufzeiten zu realisieren.

### 5.6.4 Set

sh. „Set Cyclic“.

### 5.6.5 Set Cyclic

Initialisiert die Zykluszeit des Timers und konfiguriert ihn als zyklischen Timer.

Der Timer muss mit einem separaten Start-Befehl gestartet werden. Bei Ablauf des Timers wird ein Timer-Ereignis erzeugt, und der Zyklus wird automatisch neu gestartet. Zum Beenden der Timer-Ereignisses muss ein Stop-Befehl verwendet werden.

### 5.6.6 Set Single

Initialisiert die Zykluszeit des Timers und konfiguriert ihn als einfachen Timer (Single Shot).

Der Timer muss mit einem separaten Start-Befehl gestartet werden. Bei Ablauf des Timers wird ein einmaliges Ereignis erzeugt, und der Timer wird angehalten.

## 5.7 Aktion ändern

Um eine Aktion zu verändern, muss diese über die Aktionsliste mit der Maus ausgewählt werden. Nach Klick auf ‚Edit action‘ (Abbildung 14, Markierung 2) erscheint das entsprechende Fenster, in welchem die gespeicherten Parameter dieser Aktion dargestellt werden. Diese können nun verändert werden.

Vorsicht ist geboten beim Ändern des Aktionsnamens. Tippt der Anwender einen schon vorhandenen Namen ein, wird die veränderte Aktion tatsächlich unter diesem Namen gespeichert und ersetzt zugleich die bereits vorhandene Aktion. Deshalb erhält der Anwender beim Speichern eine entsprechende Warnung. Durch Bestätigen mit ‚Yes‘ wird die schon vorhandene Aktion überschrieben, bei ‚No‘ öffnet sich wieder der Dialog zum Ändern der Aktion, und mit ‚Abort‘ wird der Änderungsvorgang abgebrochen, ohne die Eingaben zu speichern.

## 5.8 Aktion löschen

Um eine Aktion zu löschen, wählt man eine Aktion aus und klickt auf ‚Delete action‘ (sh. Abbildung 14).

## 5.9 Aktionen in der Ereignisliste hervorheben

Es ist möglich, alle Ereignisse hervorzuheben, die mit einer Aktion assoziiert sind. Mit Rechtsklick auf eine Aktion kann ein Kontextmenü geöffnet und mit dem Menüpunkt ‚Highlight attached events‘ alle verknüpften Ereignisse markiert werden (Abbildung 22).

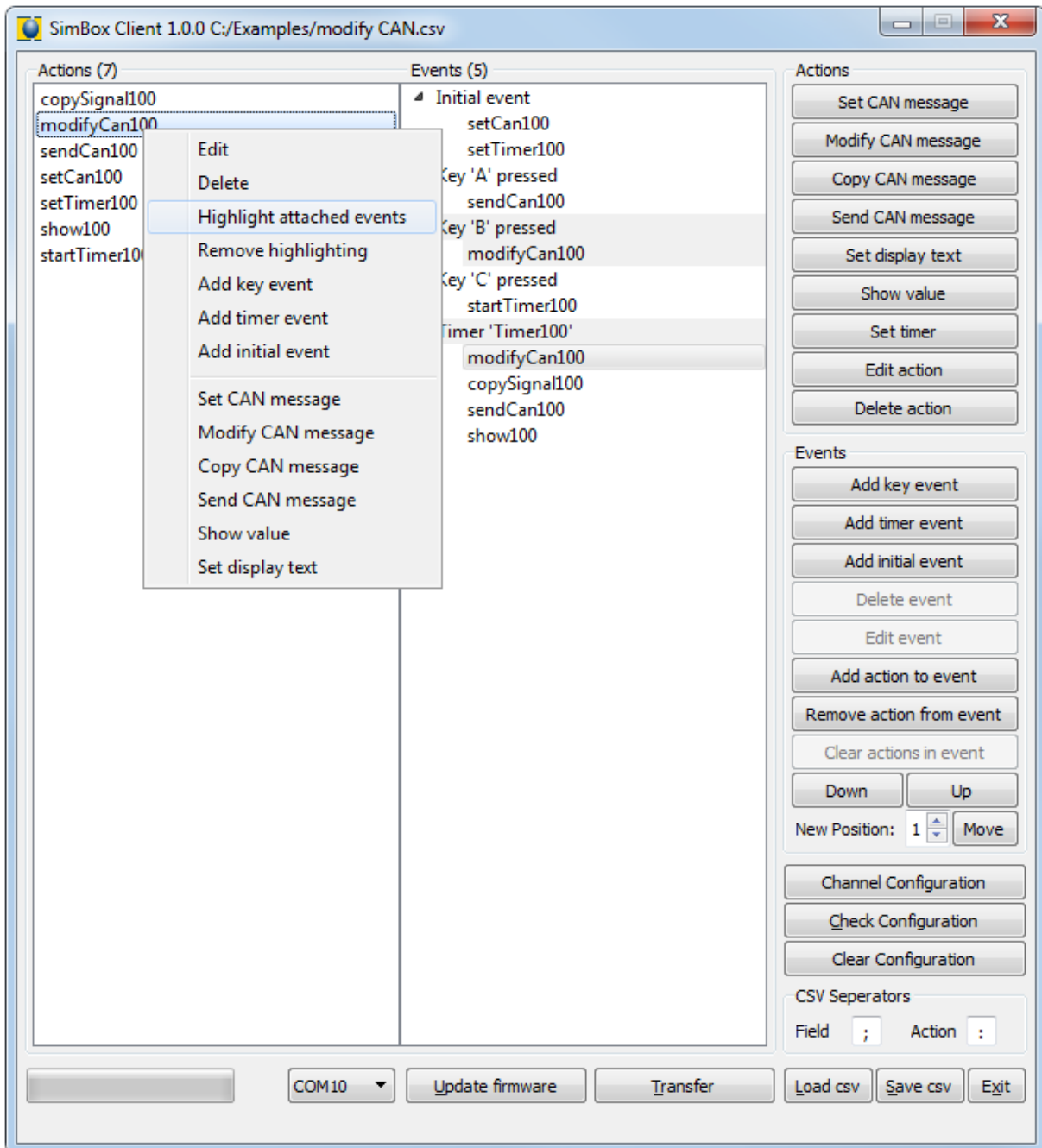


Abbildung 22: Kontextmenü einer Aktion

Danach werden alle Events farblich hervorgehoben, die mit dieser Aktion verbunden sind. Mit Klick auf ‚Remove highlighting‘ werden sämtliche Hervorhebungen zurückgenommen.

Hinweis: In Abbildung 22 ist bereits der Zustand nach Aufruf der Markierungsfunktion dargestellt.

## 6 Ereignisse

In der SimBox können verschiedene Arten von Ereignissen für die Konfiguration genutzt werden: ein Anfangsereignis, ein Tastenereignis und verschiedene Timer, die in der SimBox durchlaufen. Jedem Ereignis ist eine Ereignisaktionsliste zugeteilt, welche die Aktionen auflistet, die beim Auftreten des Ereignisses abzuarbeiten sind. Dabei bestimmt die Reihenfolge der Aktionen die Reihenfolge der Abarbeitung, wobei die oberste Aktion zuerst und die letzte Aktion zuletzt abgearbeitet wird. Um ein Ereignis zu erstellen, muss zuerst eine Aktion ausgewählt werden (Klick auf den jeweiligen Aktionsnamen in der Aktionsliste). Mit gedrückter Strg-Taste können mehrere Aktionen ausgewählt werden. Beim Erstellen des Ereignisses werden die markierten Aktionen gleich mit diesem Ereignis verknüpft.

Wenn Sie bereits ein bestimmtes Ereignis in Ihrer Konfiguration angelegt haben, können Sie es nicht erneut hinzufügen. Sie können das bestehende Ereignis ändern, indem Sie Aktionen zu seiner Aktionsliste hinzufügen oder löschen, oder das Ereignis zuerst löschen und anschließend neu erstellen.

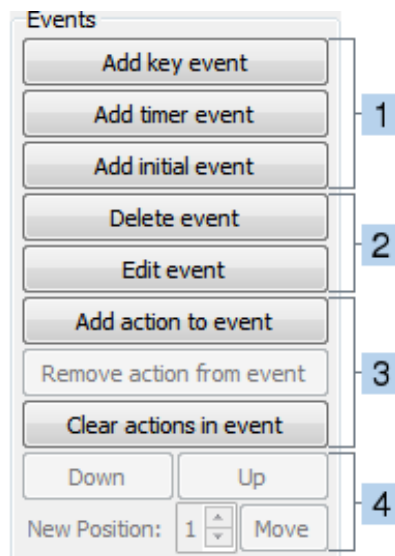


Abbildung 23: Ereignis-Auswahl

- (1) Erstellen von Ereignissen
- (2) Bearbeiten eines Ereignisses
- (3) Hinzufügen und Löschen von Aktionen
- (4) Bearbeiten der Reihenfolge in einer Aktionsliste

### 6.1 Anfangsereignis

Klickt man auf ‚Add initial event‘ (Abbildung 23, Markierung 1) wird ein Anfangsereignis festgelegt; dessen Aktionen werden beim Start der SimBox einmalig ausgeführt. Das Anfangsereignis bietet sich an, um die Daten von CAN-Nachrichten und Timern zu initialisieren.

Ein Anfangsereignis kann nur einmal definiert werden, wenn der Anwender ein weiteres Mal auf den entsprechenden Button klickt, kann er damit nur das Anfangsereignis ersetzen, jedoch kein weiteres hinzufügen. Zu einem bereits definierten Anfangsereignis können jedoch, wie zu den anderen Ereignissen auch, weitere Aktionen angefügt werden (sh. Kapitel 6.6).

### 6.2 Tastaturereignis

Jede der frei programmierbaren Tasten der SimBox kann ein Ereignis auslösen. Zuerst wählt man eine oder mehrere Aktionen aus und klickt danach mit der Maus auf die Taste ‚Add key event‘ (Abbildung 23, Markierung 1). Folgender Dialog erscheint:

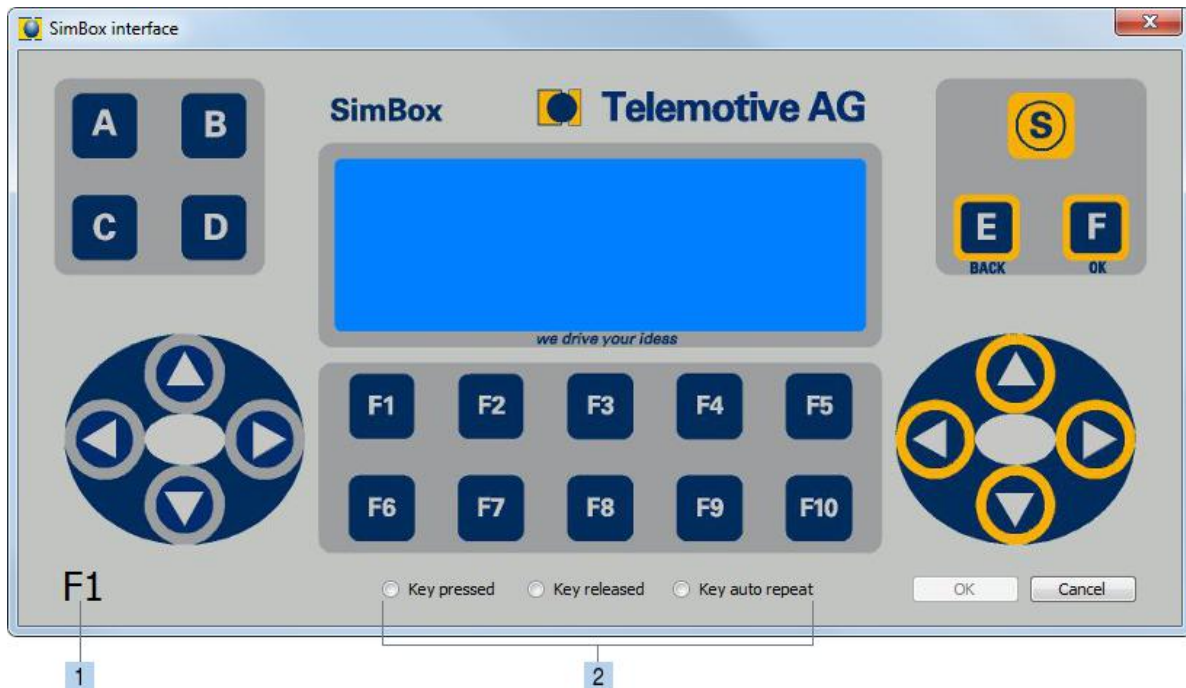


Abbildung 24: Tastaturereignis definieren

- (1) Ausgewählte Taste
- (2) Art der Tastenbetätigung

Ein Tastaturereignis wird definiert durch den Namen der Taste und der Art der Betätigung (Drücken, Loslassen, Autorepeat). Im Auswahldialog wird beim Klick auf eine Taste deren Name in der unteren linken Ecke angezeigt (Abbildung 24, Markierung 1). Die Art der Betätigung wird mit den Auswahlknöpfen im unteren Bereich des Dialogs eingestellt (Abbildung 24, Markierung 2):

- ‚Key pressed‘: Ereignis wird ausgelöst, wenn die Taste gedrückt wird.
- ‚Key released‘: Ereignis wird ausgelöst, wenn die Taste losgelassen wird.
- ‚Auto repeat‘: Ereignis wird wiederholt ausgelöst, wenn die Taste länger gedrückt gehalten wird.

Hinweis: Das Tastaturereignis Autorepeat setzt erst mit einer gewissen Verzögerung nach Betätigen der Taste ein. Soll eine Aktion unmittelbar bei Betätigung der Taste und zugleich mittels Autorepeat ausgeführt werden, dann muss diese Aktion sowohl in den Aktionslisten zum Ereignistyp ‚Key pressed‘ als auch ‚Auto repeat‘ erscheinen.

### 6.3 Timer

Ein Timer-Ereignis wird angelegt, indem man die gewünschten Aktionen in der Aktionsliste markiert und danach auf ‚Add timer event‘ klickt (Abbildung 23, Markierung 1). Daraufhin öffnet sich ein Dialog zur Eingabe eines Namens für diesen Timer. Hier kann ein neuer Name für diesen Timer angegeben werden, die Namen der bislang definierten Timer werden als Vorlage angeboten.

Wenn nach dem Start des Timers dessen Zykluszeit abgelaufen ist, wird das Ereignis ausgelöst und die verknüpften Aktionen werden ausgeführt.

### 6.4 Ereignis ändern

Durch Klick auf ‚Edit event‘ (Abbildung 23, Markierung 2) lassen sich nachträglich die Ereignisse in einem gewissen Umfang verändern. Bei Tastaturereignissen lässt sich der Name der Taste und die Art der Betätigung ändern, bei Timer-Ereignissen der Name des Timers.

### 6.5 Ereignis löschen

Um ein Ereignis zu löschen, wählt man das Ereignis aus und klickt auf ‚Delete event‘ (Abbildung 23, Markierung 2).

## 6.6 Zuweisen einer Aktion zu einem bestehenden Ereignis

Es gibt mehrere Arten der Zuweisung einer Aktion an ein bereits konfiguriertes Ereignis.

1. Eine oder mehrere Aktionen aus der Aktionsliste zuweisen:  
Der Anwender kann eine oder mehrere Aktionen aus der Aktionsliste markieren und dann entweder via Drag & Drop oder der Taste ‚Add action to event‘ (Abbildung 23, Markierung 3) in ein Ereignis einfügen. Lässt er bei einem Drag & Drop-Vorgang die Maus über einem Ereignis los bzw. markiert ein Ereignis, dann werden die markierten Aktionen an das Ende der jeweiligen Aktionsliste angefügt. Lässt er stattdessen die Aktionen über einer Aktion in der Aktionsliste eines Ereignisses los bzw. markiert eine solche Aktion, so werden die markierten Aktionen an dieser Stelle eingefügt.  
Bei Zuweisen von Aktionen an einen Event mit Hilfe der Taste ‚Add action to event‘ muss im rechten Fenster ‚Events‘ das Ziel der Zuweisung markiert werden. Das Anhängen der Aktionen an die Aktionsliste des Ereignisses geschieht in gleicher Art und Weise wie bei Verwendung von Drag & Drop.
2. Eine Aktion der Aktionsliste eines Ereignisses in die Aktionsliste eines anderen Ereignisses einfügen:  
Dem Anwender ist es möglich, die gesamte Aktionsliste eines Ereignisses in die Aktionsliste eines anderen Ereignisses zu verschieben. Diese Möglichkeit besteht nur via Drag & Drop, der Anwender muss ein Ereignis markieren und dieses Ereignis über einem anderen Ereignis bzw. in der Aktionsliste eines anderen Ereignisses loslassen. Um ein versehentliches Verschieben einer Aktionsliste zu verhindern, muss der Anwender diesen Vorgang in einer Rückfrage bestätigen.

## 6.7 Eine Aktion von einem Ereignis entfernen

Möchte der Nutzer ein Ereignis entfernen, so kann er diese Aktion markieren und per Klick auf ‚Remove action from event‘ (Abbildung 23, Markierung 3) entfernen.

## 6.8 Ein Ereignis von allen Aktionen entbinden

Man kann die Aktionsliste eines Ereignisses komplett löschen, indem man das Ereignis selektiert und mittels Klick auf ‚Clear actions in event‘ (Abbildung 23, Markierung 3) die Aktionsliste löscht.

## 6.9 Ein Aktion in der Aktionsliste bewegen

Die Abarbeitung einer Aktion hängt von der Position der Aktion in der Aktionsliste ab. Möchte der Anwender eine bestimmte Aktion in der Liste verschieben, so dass sie vor oder nach einer anderen Aktion abgearbeitet wird, so markiert er die jeweilige Aktion. Mit Klick auf ‚Up‘ oder ‚Down‘ (Abbildung 23, Markierung 4) lässt sich die Position um eine Stelle nach oben oder unten verschieben. Beim Verschieben über das Ende der Aktionsliste hinaus rollt der Eintrag an das andere Ende der Liste. Möchte der Anwender nun diese Aktion auf eine bestimmte Stelle setzen, so kann er dies, indem er die Aktion markiert, dann die gewünschte Position einstellt und per ‚Move‘ die Aktion an die entsprechende Stelle bewegt.

Die Reihenfolge in Aktionslisten lässt sich auch mittels Drag & Drop verschieben. Fassen Sie eine der Aktionen aus einer Aktionsliste mit der Maus an und verschieben Sie sie einfach an die gewünschte Stelle.



## 6.10 Aktionen aus einer Ereignisliste hervorheben

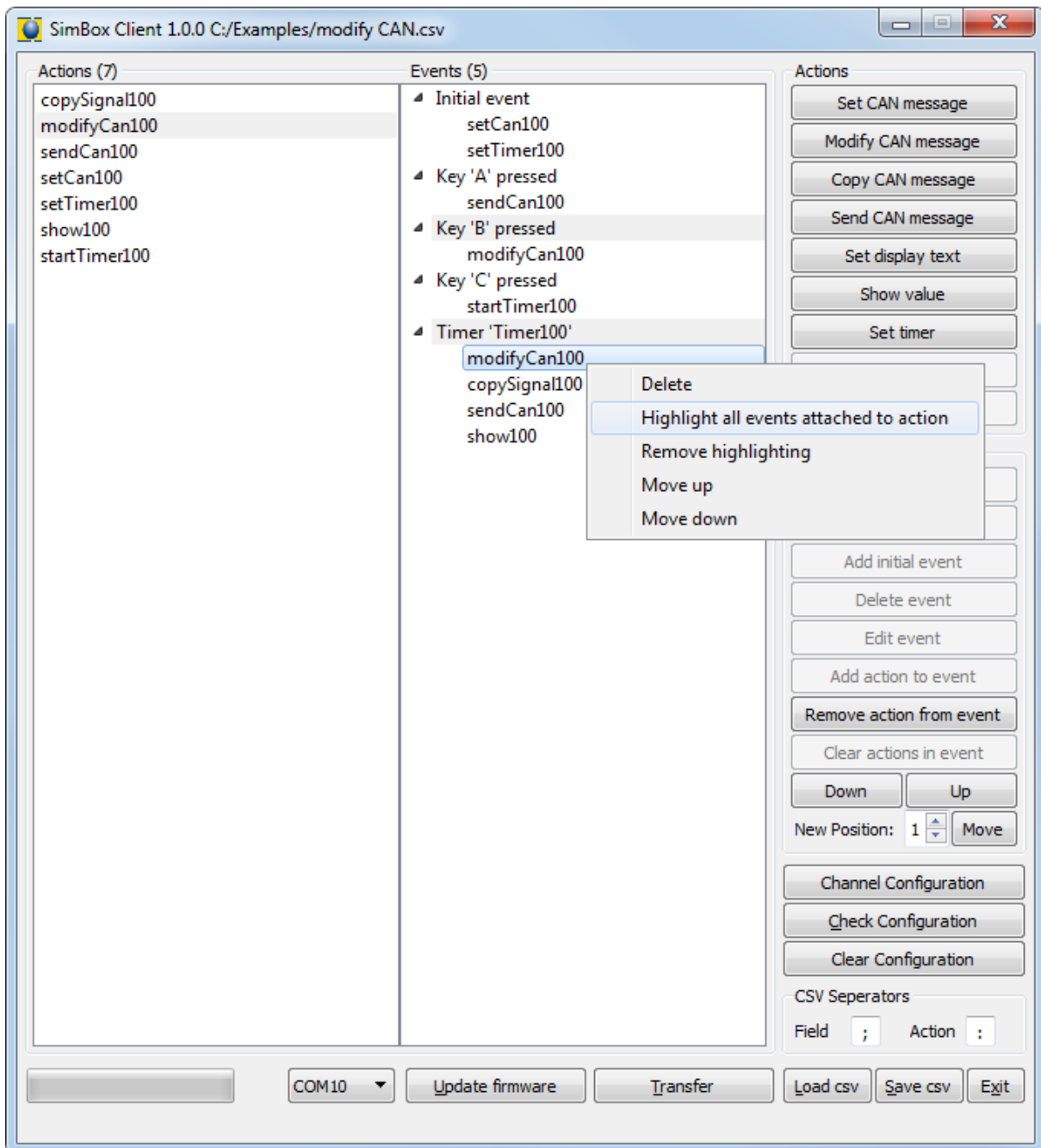


Abbildung 25: Aktion in Ereignisliste hervorheben

Es ist möglich, aus der Aktionsliste eines Ereignisses heraus eine der verbundene Aktionen hervorzuheben, und dadurch sowohl alle anderen Ereignisse, welche diese Aktion verwenden, als auch die Aktion selbst im linken Aktionsfenster schnell über die Hervorhebung auffinden zu können. Dazu klickt der Anwender mit der rechten Maustaste auf eine Aktion in der Aktionsliste eines Ereignisses und öffnet das Kontextmenü des Eintrags. Beim Klicken auf die ‚Highlight all events attached to this action‘ (Abbildung 25) markiert der Client die Aktion und alle assoziierten Ereignisse.

Hinweis: In Abbildung 25 ist die Hervorhebung bereits dargestellt.

## 7 Tutorial: Erstellung einer Konfiguration

Die SimBox reagiert entsprechend ihrer Konfiguration auf verschiedene Ereignisse und führt daraufhin die entsprechenden Aktionen aus. Ein Ereignis kann beispielsweise das Betätigen einer Taste sein, die ausgeführte Aktion das Darstellen eines Texts auf dem Display oder das Versenden einer CAN-Nachricht.

Bei der Erstellung einer Konfiguration ist es sinnvoll, zuerst die Aktionen zu definieren und erst später diese mit den auslösenden Ereignissen zu verbinden.

Bei der Neuerstellung einer Konfiguration sollten zuerst mit dem Knopf 'CAN channel configuration' (sh. Abbildung 8) der Name und die Parameter für die beiden CAN-Anschlüsse der SimBox eingestellt werden.

### 7.1 Eine „Hallo Welt“-Konfiguration

Bei diesem Beispiel soll beim Einschalten der SimBox der Text „Hallo Welt“ auf das Display der SimBox geschrieben werden.

Die Aktion, welche hier ausgeführt werden soll, ist „Set display text“ (sh. Abbildung 14). Das Ereignis, welches diese auslöst, soll der Start der SimBox-Firmware sein. Dieses Ereignis kann im Client über den Knopf ‚Add initial event‘ (sh. Abbildung 23) angesprochen werden.

Nach Öffnen des Clients wird zuerst ein Name für diese Konfiguration vergeben. Im Dialog, welcher sich bei Klick auf ‚CAN channel configuration‘ (sh. Abbildung 8) öffnet, werden dazu ein Name eingetragen und die Einstellungen mit ‚OK‘ übernommen.

Nun wird die Aktion angelegt. Nach Klick auf den Knopf ‚Set display text‘ (sh. Abbildung 14) öffnet sich der Dialog zur Eingabe des Textes (sh. Abbildung 15). Im Feld Text wird „Hallo Welt“ eingetragen. Bei ‚X pos‘ und ‚Y pos‘ wird jeweils 1 eingetragen, dies ist die linke obere Ecke des Displays. Unter ‚Action name‘ wird „hello“ eingetragen, und anschließend der Dialog mit Klick auf ‚OK‘ geschlossen.

Diese Aktion wird nun als Startereignis verknüpft. Markieren Sie dazu im linken Fenster ‚actions‘ den Eintrag „hello“. Klicken Sie anschließend auf den Knopf ‚Add initial event‘ (sh. Abbildung 23).

Die Konfiguration kann nun als CSV-Datei gespeichert und auf die SimBox gebracht werden. Die nötigen Schritte sind in diesen Kapiteln beschrieben:

- Speichern: Kap. 3.6
- Laden der Konfiguration in den Client: Kap. 3.3
- Übertragen der Konfiguration an die SimBox: Kap. 3.4

Beim Start der SimBox werden kurz die Startbildschirme der SimBox-Firmware gezeigt, anschließend erscheint im Display der Text „Hallo Welt“. Ggf. schaltet sich die SimBox nach der erfolgreichen Übertragung der Konfiguration aus; schalten Sie sie in diesem Fall bitte durch Betätigen der „S“-Taste wieder ein.

### 7.2 Erweiterte „Hallo Welt“-Konfiguration

Die Konfiguration soll nun so verändert werden, dass beim Start nur „Hallo“ erscheint. Bei Betätigen der Taste ‚F1‘ soll dieser Text auf „Hallo Welt“ erweitert werden. Ein Druck auf Taste ‚F2‘ soll den gesamten Text löschen.

Dies erfordert drei gleichartige Aktionen mit verschiedenen Parametern: Schreiben des Texts „Hallo“, Schreiben von „Welt“ hinter den vorigen Text, und Schreiben von Leerzeichen über diesen gesamten Bereich, um den zuvor geschriebenen Text wieder zu löschen. Die auslösenden Ereignisse sind einmal das Startereignis, und zweimal ein Tastendruck.

Im ersten Schritt werden in der oben erstellten „Hallo Welt“-Konfiguration die nötigen Aktionen verändert bzw. neu erstellt. Nach dem Laden der Konfiguration wird in der Aktionsliste die Aktion „hello“ markiert, dann mit ‚Edit action‘ (sh. Abbildung 14) der Dialog zum Ändern der Parameter aufgerufen. Im Feld Text wird das Wort „Welt“ gelöscht.

Anschließend werden jeweils mit ‚Set display text‘ (sh. Abbildung 14) zwei neue Aktionen angelegt. Die erste erhält für die Parameter ‚Text‘, ‚X pos‘, ‚Y pos‘ und ‚Action name‘ (sh. Abbildung 15) die Werte „Welt“, 7, 1 und „world“, die zweite „ „ (zehn Leerzeichen), 1, 1 und „clear“. Die neuen Aktionen „world“ und „clear“ werden nun mit den Tasten ‚F1‘ bzw. ‚F2‘ verknüpft. Dazu wird in der

Aktionsliste der jeweilige Aktionsname markiert, mit Klick auf ‚Add key event‘ der Dialog für die Tastenauswahl aufgerufen, und anschließend wird auf der Abbildung der SimBox-Tastatur die jeweilige Taste und der Ereignistyp ‚Key pressed‘ angeklickt und mit Klick auf ‚OK‘ bestätigt. Im Fenster Ereignis wird jeweils eine neue Zeile hinzugefügt.

Nach dem Übertragen der Konfiguration auf die SimBox erscheint in deren Display zuerst nur das Wort „Hallo“. Nachdem die Taste ‚F1‘ betätigt wurde, ist der vollständige Text „Hallo Welt“ zu sehen. Durch Druck auf ‚F2‘ kann dieser nun gelöscht werden.

Bei erneutem Druck auf ‚F1‘ erscheint nun lediglich das Wort „Welt“ auf dem Display. Soll der gesamte Text erscheinen, stehen zwei Wege offen: Zum einen kann die Aktion „world“ so abgeändert werden, dass sie den gesamten Text „Hallo Welt“ an die Position (1;1) des Bildschirms schreibt. Zum anderen können die beiden Aktionen „hello“ und „world“ gleichzeitig markiert und dann mit der Taste ‚F1‘ verknüpft werden. Der Eintrag in der Ereignis-Liste führt nun zwei Aktionen für diesen Tastendruck auf ‚F1‘ an: „world“ und „hello“.

### 7.3 Senden einer CAN-Nachricht

In diesem Beispiel soll auf Tastendruck hin eine CAN-Nachricht versandt werden. Der Inhalt des ersten Bytes der CAN-Nachricht soll mit einem weiteren Tastendruck um eins erhöht werden.

Für das Versenden einer CAN-Nachricht sind verschiedene Schritte notwendig:

- Definieren der CAN-Nachricht:  
In der SimBox muss ein Speicherplatz für eine CAN-Nachricht vorgesehen und mit Daten gefüllt werden. Dies geschieht mit der Aktion ‚Set CAN message‘. Für jede CAN-Nachricht soll diese Aktion sinnvollerweise einmal mit dem Startereignis verknüpft werden.
- Versenden der Nachricht:  
Nachdem die Nachricht definiert wurde, kann sie auf dem CAN-Bus versandt werden. Dazu dient die Aktion ‚Send CAN message‘.
- Ändern der Daten einer CAN-Nachricht:  
Zur Simulation von verschiedenen Betriebszuständen sollen Datenfelder innerhalb einer CAN-Nachricht verändert werden. Für das Austauschen von wenigen fest vorgegebenen Werten kann die Aktion ‚Set CAN message‘, ggf. unter Verwendung einer Maske zum Setzen von bestimmten Bits, verwendet werden (sh. Kap. 0), zum Verändern von Werten wie Motortemperatur oder Geschwindigkeit empfiehlt sich die Aktion ‚Modify CAN signal‘ (sh. Kap. 5.5).

In diesem Beispiel soll das Versenden und das Ändern eines Signals von Tastendrücken gesteuert werden.

Nach dem Öffnen des Clients mit einer leeren Konfiguration sollten mit dem Knopf ‚CAN channel configuration‘ (sh. Abbildung 8) die Parameter der CAN-Busse festgelegt werden. Die SimBox muss mit den gleichen Einstellungen konfiguriert werden wie die anderen Teilnehmer auf dem CAN-Bus.

Anschließend werden die verschiedenen Aktionen definiert. Zuerst muss die CAN-Nachricht mit einer ID 0x100 definiert und mit den Ausgangsdaten gefüllt werden. Die DLC wird in diesem Beispiel auf „2“ eingestellt, der gesamte Datenbereich mit „0“ initialisiert, d. h. ‚CAN Data‘ wird auf „00 00“ eingestellt, ‚CAN Mask‘ auf „FF FF“. Diese Werte werden in den Dialog eingetragen, welcher sich bei Klick auf ‚Set CAN message‘ (sh. Abbildung 14) öffnet.

Diese Nachricht kann nach Ausführung der Set-Aktion bereits versandt werden. Mit Klick auf ‚Send CAN message‘ (sh. Abbildung 14) öffnet sich ein Dialog, in den nun die identische CAN-ID eingegeben wird.

Zum Ändern des ersten Bytes wird eine Aktion ‚Modify CAN message‘ (sh. Abbildung 14) eingerichtet, welche die gleichen Adressinformation der CAN-Nachricht beinhaltet:

Channel: CAN\_1; CAN ID: 100; extended Identifier: nein; Operation: ADD; Bit position: 0; Length: 8; Value/Step: 1; min: 0; max: 0; ActionName: modifyCan100.

Diese Aktionen werden nun mit den gewünschten Ereignissen verknüpft: setCan100 mit dem Startevent (‚Add initial event‘), sendCan100 mit der Taste A, modifyCan100 mit der Taste B (‚Add key event‘). Bei den Tasten soll jeweils der Druck auf die Taste die Aktion ausführen.

## 7.4 Timer-gesteuertes Senden und Anzeige des Werts

Für viele Anwendungsfälle soll eine CAN-Nachricht automatisch mit einer vorgegebenen Zykluszeit versendet werden. Um den Versand der Nachricht auszulösen, kann in der SimBox ein Timer definiert werden, der beim Ablauf ein Timer-Ereignis auslöst. Durch diesen Ereignis können wiederum verschiedene Aktionen, etwa der Versand einer CAN-Nachricht, ausgelöst werden.

Um einen Timer-Ereignis auszulösen, sind verschiedene Schritte notwendig:

- Festlegen des Timer-Ereignisses:  
Durch Verknüpfen von Aktionen mit einem Timer-Ereignis mittels ‚Add timer event‘ wird der Name eines Timers innerhalb der Konfiguration bekannt gemacht und kann für Aktionen zum Einstellen des Timers verwendet werden.
- Definieren eines Timers:  
Dies geschieht durch Setzen der Zykluszeit des Timers. Für noch unbekannte Timer wird ein Speicherplatz in der SimBox reserviert und mit den Parametern einer der „Set“-Datentypen der Timer-Aktion gefüllt.
- Starten des Timers:  
Das Setzen der Timer-Parameter startet den Timer nicht automatisch. Dazu muss eine Timer-Aktion mit dem Typ „Start“ aufgerufen werden. Ein Timer kann danach wieder angehalten und neu gestartet werden, z. B. um kurz Busruhe der SimBox herzustellen.

Das Beispiel aus dem vorigen Kapitel wird nun um einen Timer mit einer Zykluszeit von 500 ms erweitert. An diesen wird die Aktion zum Senden der CAN-Nachricht gebunden.

Vor dem Verknüpfen der Aktionen mit diesem Timer legen wir weitere Aktionen zur Anzeige des Datenbytes an, welches in der CAN-Nachricht verändert wird. Zum Anzeigen von Werten aus einer CAN-Nachricht müssen diese erst in eine Variable kopiert werden. Der Variablenwert kann dann unter Angabe von Formatoptionen zur Anzeige gebracht werden.

Zuerst wird also eine Aktion ‚Copy CAN message‘ (sh. Abbildung 14) zum Kopieren eines Signals aus einer CAN-Nachricht in eine Variable angelegt. Durch die Angabe eines Variablennamens in dieser Kopier-Aktion wird automatisch ein entsprechender Speicherplatz reserviert. Nach Klick auf den Knopf ‚Copy CAN message‘ (sh. Abbildung 14) werden die folgenden Werte in die Felder eingetragen, die Informationen zum CAN-Signal entsprechen denen der Aktion ‚Modify CAN message‘:

CAN channel: CAN\_1; CAN ID: 100; Operation: read; Bit position: 0; Length: 8; Variable: variable100; Action name: copySignal100.

Zur Anzeige des Werts wird mittels Klick auf ‚Show value‘ (sh. Abbildung 14) die passende Aktion mit folgenden Parametern erzeugt:

Variable: variable100; X position: 1; Y position: 2; Offset: 0; Scale: 0; field length: 3; display format: unsigned; alignment: right; Action name: show100.

Im vorigen Beispiel wurde das Versenden der CAN-Nachricht durch einen Tastendruck ausgelöst. In diesem Beispiel soll dies automatisch durch den Timer erfolgen, gleichzeitig soll der Wert des ersten Bytes in der CAN-Nachricht erhöht und im Display angezeigt werden. Deshalb werden die Aktionen mit den Namen modifyCan100, sendCan100, copySignal100 und show100 gleichzeitig markiert und durch Klick auf den Knopf ‚Add timer event‘ (sh. Abbildung 23) dem Timer zugewiesen.

Das Markieren von mehreren Einträgen in der Aktionsliste erfolgt, indem die weiteren Einträge bei gedrückter Strg-Taste angeklickt werden. Beim Verknüpfen mit dem Timer muss der gewünschte Timer-Name ausgewählt werden, hier ‚Timer100‘.

Nun werden die Aktionen zum Steuern des Timers angelegt. Durch Klick auf ‚Set timer‘ (sh. Abbildung 14) wird der Dialog zur Eingabe der Parameter des Timers aufgerufen. Als Timer-Name wird ‚Timer100‘ ausgewählt. Im Feld ‚Type‘ wird Set\_Cyclic eingestellt, bei ‚Cycle time‘ die Zykluszeit mit 500 ms eingestellt, und als Aktionsname der Text „setTimer100“ eingegeben.

In ähnlicher Weise wird eine zweite Aktion zum Starten des Timers angelegt, hier werden die Felder so ausgefüllt:

Timer name: Timer100; Type: Start; Action name: startTimer100.

Bevor der Timer verwendet werden kann, muss seine Zykluszeit initialisiert werden. Dies geschieht am besten im Startereignis. Dazu wird im Hauptfenster des Clients die Aktion ‚setTimer100‘ aus dem linken Feld ‚Actions‘ auf das Anfangsereignis ‚Initial event‘ im rechten Feld ‚Events‘ gezogen.

Jetzt sollten dem Anfangsereignis die beiden Aktionen ‚setCan100‘ und ‚setTimer100‘ zugeordnet sein.

Nun fehlt noch das Starten des Timers. Eine Aktion mit dem Namen startTimer100 ist bereits angelegt, diese muss nur noch mit einer Taste verknüpft werden. In diesem Beispiel wird sie auf die Taste C gelegt.

Nach Betätigen der Taste C kann nun in der Anzeige der SimBox beobachtet werden, wie sich der Wert des ersten Bytes in der CAN-Nachricht verändert. Die CAN-Nachricht mit diesem Wert wird mit jedem Zyklus zum Senden freigegeben.

## 7.5 Timer-gesteuertes Senden bei Tastendruck

In diesem Beispiel soll eine CAN-Nachricht mit einer vorgegebenen Zykluszeit versendet werden, solange eine bestimmte Taste gedrückt wird.

In den folgenden Beispielen wird jeweils eine CAN-Nachricht versendet. Dazu müssen eine SET\_CAN- und eine SEND\_CAN-Aktion definiert werden. Die SET\_CAN-Aktion wird an das Anfangsereignis gebunden. An welches Ereignis das Versenden der Nachricht gebunden wird, unterscheidet sich in den folgenden Beispielen.

### 7.5.1 Verwenden eines zyklischen Timers

Eine einfache Möglichkeit zur Umsetzung dieses Beispiels ist die Verwendung eines zyklischen Timers zum Versenden der CAN-Nachricht. Nach dem Start des Timers läuft dieser von allein. Da das Versenden bei Loslassen der Taste gestoppt werden soll, ist hier noch ein Konfigurationsschritt notwendig: der Timer muss gestoppt werden.

Zur Steuerung des Timers sind drei Schritte notwendig:

- Initialisieren der Timer-Parameter (Zykluszeit), z. B. gebunden an das Anfangsereignis,
- Starten des Timers, gebunden an das Drücken einer Taste (Key pressed),
- Stoppen des Timers, gebunden an das Loslassen dieser Taste (Key released).

Das Versenden der CAN-Nachricht wird an dieses Timer-Ereignis gebunden.

Das Timer-Ereignis wird erst bei Ablauf des Timers ausgelöst. Damit die CAN-Nachricht ohne Verzögerung bereits mit dem Tastendruck versendet wird, wird die SEND\_CAN-Aktion zusätzlich noch an das ‚Key pressed‘-Ereignis gebunden.

Für das Versenden der CAN-Nachricht sind zwei Schritte notwendig: Initialisieren der Daten und der Sendeauftrag.

Im Folgenden werden kurz die Bedienschritte im Client umrissen:

Zuerst wird das allgemeine Setup über den Button „Channel configuration“ aufgerufen. Neben den Parametern für die CAN-Schnittstellen können Sie dort einen Namen für diese Konfiguration hinterlegen.

Danach werden die beiden Aktionen angelegt, welche mit der CAN-Nachricht in Verbindung stehen. Bei Bedarf können jetzt auch schon Aktionen angelegt werden, die eine Rückmeldung im Display geben („Set display text“; im CSV-Quelltext im nächsten Kapitel enthalten). Mit den Buttons „Set CAN Message“ und „Send CAN Message“ wird der jeweilige Dialog zum Anlegen der Aktionen aufgerufen. Im Beispiel erhalten sie die Namen „setCAN2“ und „sendCAN2“.

Vor dem Anlegen der Aktionen zur Steuerung des Timers wird erst der Timer-Ereignis angelegt. Beim Anlegen des Timer-Ereignisses wird dem Client der Name dieses Timers bekannt gemacht, danach können dann die Aktionen zum Steuern dieses Timers konfiguriert werden.

Das Anlegen des Timer-Ereignisses erfolgt durch Verbinden von Aktionen mit einem neuen Timer. Zuerst werden im linken Fenster „Actions“ die gewünschten Aktionen ausgewählt, in diesem Beispiel die Aktion „sendCAN2“. Dann wird durch Klick auf „Add timer event“ der Dialog zur Eingabe des Timer-Namens geöffnet. Nach dem erfolgreichen Anlegen des Ereignisses erscheint ein passender Eintrag im rechten Fenster „Events“.

Nun können die drei Aktionen zum Steuern des Timers angelegt werden. Der Dialog wird über den Button „Set timer“ aufgerufen. Der Name des Timers aus dem bereits angelegten Ereignis kann in diesem Dialog aus einer Liste ausgewählt werden.

Nach dem Anlegen dieser Aktionen können die Zuordnungen der Aktionen zur Tastaturbedienung und zum Initialisierungsereignis vorgenommen werden. Teilweise wäre dies bereits zu einem frü-

heren Zeitpunkt möglich gewesen, jetzt sind aber die Aktionen vollständig definiert und können in einem Rutsch zugeordnet werden. Die genaue Zuordnung können sie dem Listing der CSV-Datei entnehmen.

### 7.5.2 Verwenden eines Single-Shot-Timers

Die Aufgabenstellung aus diesem Beispiel kann alternativ auch durch Verwendung eines einfachen Single-Shot-Timers anstelle des zyklischen Timers gelöst werden.

In der Aktion zum Initialisieren des Timers wird „Type“ von SET\_CYCLIC auf SET\_SINGLE umgestellt. Durch den Tastendruck wird dieser Timer gestartet, erzeugt nach der eingestellten Zeit ein einzelnes Timer-Ereignis und stoppt. In diesem Timer-Ereignis kann der Timer erneut gestartet werden und löst wiederum nach Ablauf seiner Alarmzeit erneut ein Timer-Ereignis aus. Als weitere Änderung muss also noch die Aktion zum Starten des Timers in die Aktionsliste des Timer-Ereignisses eingefügt werden.

Die Verwendung von zyklischen Timern bietet Vorteile im Vergleich zu der hier vorgestellten Lösung. Wegen des erneuten Startens des Timers in seiner eigenen Ereignisliste fallen Programmlaufzeiten bei der Genauigkeit der Zykluszeiten stärker ins Gewicht.

Hinweis: Die SimBox unterstützt das Starten eines Timers aus der eigenen Ereignisliste heraus bis Firmware-Version 1.0.5 noch nicht.

## 7.6 Timer-gesteuertes Senden bei Tastendruck mit Anschlagverzögerung

Wie im vorangegangenen Beispiel soll hier eine CAN-Nachricht in regelmäßigen Abständen versendet werden, solange eine Taste gedrückt wird. Der Abstand der ersten Nachricht zur ersten Wiederholung soll jedoch anders eingestellt sein als der Abstand der weiteren Wiederholungen.

Diese Aufgabe unterteilt sich grob in zwei verschiedene Phasen. Die erste Phase umfasst den ersten Abstand, also vom Druck der Taste bis zur ersten Wiederholung der CAN-Nachricht. Der zweite Abschnitt umfasst die ständige Wiederholung dieser Nachricht bis zum Loslassen der Taste.

Diese zweite Phase ähnelt sehr stark dem Beispiel aus Kapitel 7.5. Wir können also diese Konfiguration als Ausgangspunkt verwenden und erweitern.

Ein wesentlicher Unterschied ist die andere Wartezeit vor der ersten Wiederholung. Um diese zu realisieren, wird ein eigener Timer verwendet, welcher mit der gewünschten Wartezeit konfiguriert wird. Dieser Timer soll nur die zweite Phase anstoßen und dann den Ablauf nicht mehr beeinflussen. Deshalb wird er als Single-Shot-Timer angelegt. Gestartet wird die erste Phase und damit dieser Timer durch einen Tastendruck, gestoppt wird er durch sein Ablaufen, oder auch durch das Loslassen der Taste. Das Anhalten des Timers bei Loslassen der Taste darf nicht vergessen werden. Bei sehr kurzen Tastendrücken würde sonst nach Loslassen der Taste doch noch das zugehörige Timer-Ereignis ausgelöst und die zweite Phase gestartet.

Die zweite Phase umfasst das zyklische Versenden der CAN-Nachricht mit gleichen Abständen. Dies ist in Kapitel 7.5.1 bereits beschrieben, die dort vorgestellten Schritte können übernommen werden. Der Auslöser ist allerdings nicht mehr ein Tastendruck, sondern das Ende der ersten Phase, markiert durch den Ablauf des oben erwähnten Timers. Demnach müssen die Aktionen zum Starten des zyklischen Timers und Versenden einer CAN-Nachricht in das Ereignis des Single-Shot-Timers eingehängt werden, welcher im Beispiel den Namen „TimerDelay“ trägt.

Dies ist auch der erste Schritt zur Veränderung der Konfiguration. Danach können die drei Aktionen zum Steuern des Single-Shot-Timers angelegt werden, ‚START‘, ‚STOP‘ und ‚SET\_SINGLE‘. Diese Aktionen müssen den passenden Ereignissen zugeordnet werden, ebenso wie die Aktionen zum Steuern des zyklischen Timers.

Die Aktionen zum Initialisieren der beiden Timer und der CAN-Nachricht können an das Anfangsereignis angebunden werden, so dass die Parameter gleich beim Starten der SimBox eingestellt werden. An das Ereignis „Tastendruck“ der gewünschten Taste werden die Aktionen zum Starten des Single-Shot-Timers „TimerDelay“ und zum Versenden der CAN-Nachricht (erste Nachricht) angebunden. An das Ereignis des Timers „TimerDelay“ werden wiederum, wie bereits oben beschrieben, die Aktionen zum Starten des zyklischen Timers sowie zum Versenden der CAN-Nachricht (erste Nachricht nach Anschlagverzögerung) angebunden. An das Ereignis des zyklischen Timers wird nur die Aktion zum Versenden der CAN-Nachricht angebunden (jede weitere Nachricht).

Auf das Ereignis „Loslassen der Taste“ müssen beide Aktionen zum Stoppen der Timer gebunden werden, da während der Anschlagverzögerung ein anderer Timer aktiv ist als danach. Wenn eine der Aktionen fehlt, wird die Konfiguration, je nach Zeitpunkt des Loslassens der Taste, zu einem unerwünschten Ergebnis führen.

Hinweis: Die Tastatur der SimBox besitzt eine Anschlagverzögerung und eine Autorepeat-Funktion mit fest eingestellten Parametern. Wenn dies Ihren Anforderungen entspricht, können direkt die Tastaturereignisse „Key pressed“ und „Auto repeat“ verwendet werden.

## 7.7 Simulation einer Geschwindigkeitsanzeige

Im folgenden wird kurz umrissen, wie Messwerte, wie etwa die Fahrzeuggeschwindigkeit, durch einen Bediener veränderbar konfiguriert werden können. Bitte entnehmen Sie Details zu der Konfiguration dem CSV-Quelltext im folgenden Kapitel. Für dieses Beispiel wurde die Konfiguration aus Kapitel 7.4 verändert und erweitert.

Die Geschwindigkeit soll mit einer Auflösung von 0,1 km/h in einem 16-Bit-Wert in einer CAN-Nachricht versendet werden. Der Anwender soll durch Tastendruck den simulierten Wert im Bereich von 0 bis 100 km/h nach oben oder unten in Schritten von 5 km/h verändern können. Dabei soll die Autorepeat-Funktion der Tastatur ebenfalls verwendet werden, damit man durch eine entsprechend lange Betätigung der Tasten den Wert in größerem Umfang ändern kann. Zur Kontrolle soll der simulierte Wert im Display dargestellt werden, hier in der Einheit km/h.

Die Basis der Datenhaltung ist das CAN-Signal, in welchem der Messwert verschickt wird. Für unser Geschwindigkeitssignal wird angenommen, dass es in der CAN-Nachricht mit der ID 0x100 in Byte 2 und 3 im Little-Endian-Format gespeichert ist. Die Auflösung soll 0,1 km/h betragen, in dieser Einheit muss mit dem CAN-Signal gearbeitet werden. Ausgehend vom gewünschten Einstellbereich 0 bis 100 km/h muss der Rohdatenwert des CAN-Signals also im Bereich 0 bis 1000 verändert werden, wobei die Schrittweite 5 km/h einer Schrittweite von 50 des CAN-Signals entspricht.

Zum Verändern des Werts dieses CAN-Signals nach oben und unten legen wir zwei ähnliche Aktionen ‚Modify CAN message‘ an. Die Bitposition für das LSB des CAN-Signals wird mit 16 angegeben. Achtung: Bei Big-Endian-Notation liegt das LSB nicht im ersten, sondern im letzten Byte des zugehörigen Datenbereichs, die korrekte Bitposition unterscheidet sich! Die Länge des CAN-Signals beträgt im Beispiel zwei Byte, es muss also ein Wert von 16 Bit in das Längenfeld eingetragen werden.

Für die Aktion ‚Modify CAN message‘ können verschiedene Typen von Operatoren eingetragen werden. Für die Kategorien ‚Add‘ und ‚Subtract‘ gibt es jeweils drei verschiedene Varianten zur Auswahl, welche je nach Anwendungsfall konfiguriert werden können. Bei dieser Aufgabenstellung ist es erwünscht, dass bei Dauerbedienung der Tasten für höhere oder niedrigere Geschwindigkeit der simulierte Wert an den Bereichsgrenzen stehen bleibt und nicht weiter verändert würde, also nicht bei Erreichen und Unterschreiten von 0 nach Stillstand auf einmal Höchstgeschwindigkeit gemeldet wird. Hier kommen die Varianten ‚ADDLIMIT‘ bzw. ‚SUBTRACTLIMIT‘ zum Einsatz. Wenn bei Aufruf der Aktion die eingestellten Bereichsgrenzen erreicht werden, wird der Wert nur bis dorthin und nicht weiter verändert. Die Grenzen für den Wertebereichs werden in den Feldern ‚min‘ und ‚max‘ eingetragen. Hier sind die Rohdatenwerte des CAN-Signals anzugeben, also 0 und 1000 für unser Beispiel. In das Feld ‚Value / Step‘ wird die Schrittweite ebenfalls in Rohdateneinheiten eingetragen, also der Wert 50. Die beiden Aktionen tragen im CSV-Listing die Namen ‚increaseSpeed‘ und ‚decreaseSpeed‘.

Um diesen Wert zur Anzeige zu bringen, sind zwei Aktionen notwendig, nämlich ‚Copy CAN message‘ und ‚Show value‘. Mit der ersten Aktion wird der Rohdatenwert des CAN-Signals in eine Variable umkopiert, und diese kann mit der zweiten Aktion im Display zur Anzeige gebracht werden. Vor dem Wert der aktuell gesendeten Geschwindigkeit kann noch ein erläuternder Text eingefügt werden. Diese drei Aktionen tragen im Beispiel die Namen ‚copySpeed‘, ‚showSpeed‘ und ‚textSpeed‘.

Die Aktion ‚showSpeed‘ enthält einen Umrechnungsfaktor. Die Rohdaten des Geschwindigkeitssignals sind in Schritten von 0,1 km/h angegeben, die Aktion ‚Show value‘ gibt jedoch Ganzzahlen aus. Deshalb wird der Rohdatenwert durch 10 geteilt und in der Auflösung 1 km/h dargestellt. Bei dem Parameter ‚Scale‘ wird dazu -10 eingetragen. Das negative Vorzeichen bedeutet, dass vor der Anzeige der Wert aus der Variable ‚varSpeed‘ durch 10 geteilt werden soll. Ein positives Vorzeichen weist die SimBox an, den Wert aus der Variable zu multiplizieren.

Nun müssen die Aktionen noch mit Tastenereignissen verknüpft werden. Die Aktionen ‚increaseSpeed‘, ‚copySpeed‘ und ‚showSpeed‘ werden an das Ereignis ‚pressed‘ der Taste ‚LEFT\_UP‘ verbunden. Bitte beachten Sie dabei die Reihenfolge in der Aktionsliste des Ereignisses! Die Aktion ‚increaseSpeed‘ soll an erster Stelle stehen, denn hier werden die Daten verändert. Danach folgt ‚copySpeed‘, um den veränderten Inhalt in die Variable zu kopieren, und darauf wiederum ‚showSpeed‘, um den Wert der Variablen zur Anzeige zu bringen. Würde ‚increaseSpeed‘ an letzter Stelle stehen, dann würde die SimBox bei Tastenbetätigung einen alten Wert des Geschwindigkeitssignals zur Anzeige bringen, und anschließend einen veränderten Wert als CAN-Nachricht versenden – Sie sehen dann nicht mehr, was tatsächlich versendet wird.

Wenn ein Signalwert über einen größeren Wertebereich hin verändert werden soll, ist es bequemer, den Autorepeat der Tastatur zu nutzen anstatt mehrmals eine Taste betätigen zu müssen. Deshalb wird in dieser Konfiguration die gleiche Aktionsliste, welche an das ‚pressed‘-Ereignis angebunden ist, auch an das Ereignis ‚auto repeat‘ der Taste ‚LEFT\_UP‘ angebunden. Bitte beachten Sie auch hier die Reihenfolge der Aktionen.

Für die Verringerung der simulierten Geschwindigkeit erzeugen Sie bitte, ähnlich wie für die Taste ‚LEFT\_UP‘, zwei Ereignisse für die Taste ‚LEFT\_DOWN‘, wobei hier die Aktion ‚increaseSpeed‘ gegen die Aktion ‚decreaseSpeed‘ ausgetauscht wird.

Um gleich beim Start der SimBox die Anzeige des aktuellen Werts zu erhalten, werden die Aktionen ‚textSpeed‘, ‚copySpeed‘ und ‚showSpeed‘ dem initialen Ereignis hinzugefügt. Auch hier muss die Reihenfolge beachtet werden: ‚copySpeed‘ muss vor ‚showSpeed‘ bearbeitet werden.

Die Konfiguration dieses Kapitels ist abgeleitet aus dem Beispiel aus Kapitel 7.4, dort ist der Start des zyklischen Timers an die Taste ‚C‘ gebunden. Erst nach Betätigen dieser Taste werden CAN-Nachrichten mit fester Zykluszeit versandt. Alternativ kann der Start des Timers auch in das initiale Ereignis verschoben werden, um die SimBox sofort bei Einschalten mit dem Versenden der CAN-Botschaften beginnen zu lassen. Beachten Sie bitte auch hier die Reihenfolge: Ein Timer muss zuerst initialisiert werden, bevor er gestartet werden darf.

Um die zusätzlichen Bits für das Geschwindigkeitssignal unterzubringen, ist die DLC-Angabe für die CAN-Nachricht auf vier Byte erweitert worden.





```
TIMER;Timer100;modifyCan100:copySignal100:sendCan100:show100;;1
KEY_PRESSED;A;sendCan100
KEY_PRESSED;B;modifyCan100
KEY_PRESSED;C;startTimer100
```

#### Timer-gesteuertes Senden bei Tastendruck (zyklischer Timer) aus Kap. 7.5.1:

```
SimBoxClientConfiguration;Pushbutton test
SPECVERSION;0.0.0.3
CAN_1;CAN1;1;highspeed;500000;0000
CAN_2;CAN2;1;highspeed;500000;0000
ACTION;SET_TEXT;clearF2;;0;0;4;4 ;
ACTION;SEND_CAN;sendCAN2;;0;CAN_2;0;222;;;0;0;0;;;
ACTION;SET_CAN;setCAN2;;0;CAN_2;0;222;2;22 22 ;ff ff
ACTION;SET_TIMER;setTimer2;;0;Timer2;set_cyclic;100
ACTION;SET_TIMER;startTimer2;;0;Timer2;start;0
ACTION;SET_TIMER;stopTimer2;;0;Timer2;stop;0
ACTION;SET_TEXT;textF2;;0;0;4;4;F2
ACTION;SET_TEXT;textHello;;0;0;1;1;Pushbutton test
INITIALIZATION;textHello;setCAN2;setTimer2
TIMER;Timer2;sendCAN2;;1
KEY_PRESSED;F2;textF2:startTimer2:sendCAN2
KEY_RELEASED;F2;clearF2:stopTimer2
```

#### Timer-gesteuertes Senden bei Tastendruck (Single-Shot-Timer) aus Kap. 7.5.2:

```
SimBoxClientConfiguration;Pushbutton test
SPECVERSION;0.0.0.3
CAN_1;CAN1;1;highspeed;500000;0000
CAN_2;CAN2;1;highspeed;500000;0000
ACTION;SET_TEXT;clearF2;;0;0;4;4 ;
ACTION;SEND_CAN;sendCAN2;;0;CAN_2;0;222;;;0;0;0;;;
ACTION;SET_CAN;setCAN2;;0;CAN_2;0;222;2;22 22 ;ff ff
ACTION;SET_TIMER;setTimer2;;0;Timer2;set_single;100
ACTION;SET_TIMER;startTimer2;;0;Timer2;start;0
ACTION;SET_TIMER;stopTimer2;;0;Timer2;stop;0
ACTION;SET_TEXT;textF2;;0;0;4;4;F2
ACTION;SET_TEXT;textHello;;0;0;1;1;Pushbutton test
INITIALIZATION;textHello;setCAN2;setTimer2
TIMER;Timer2;startTimer2:sendCAN2;;1
KEY_PRESSED;F2;textF2:startTimer2:sendCAN2
KEY_RELEASED;F2;clearF2:stopTimer2
```

#### Timer-gesteuertes Senden mit Anschlagverzögerung aus Kap. 7.6:

```
SimBoxClientConfiguration;Pushbutton test
SPECVERSION;0.0.0.3
CAN_1;CAN1;1;highspeed;500000;0000
CAN_2;CAN2;1;highspeed;500000;0000
ACTION;SET_TEXT;clearF2;;0;0;4;4 ;
ACTION;SEND_CAN;sendCAN2;;0;CAN_2;0;222;;;0;0;0;;;
ACTION;SET_CAN;setCAN2;;0;CAN_2;0;222;2;22 22 ;ff ff
ACTION;SET_TIMER;setTimer2;;0;Timer2;set_cyclic;100
ACTION;SET_TIMER;setTimer-Delay;;0;Timer-Delay;set_single;1000
```



## 9 SimBox-Bootloader

Die Firmware der SimBox ist grob in drei Teile mit verschiedenen Aufgaben unterteilt:

- Bootloader
- Firmware
- Konfiguration

Die Firmware und vor allem die Konfiguration sind die Bestandteile, welche der Anwender im alltäglichen Umgang sieht. Sie bestimmen die Funktionalität der SimBox. Jedes Mal, wenn eine neue Konfiguration mit dem SimBox-Client erstellt und an die SimBox geschickt wird, wird der Teil des Flash-Speichers, in welchem die SimBox die Konfiguration ablegt, neu geschrieben. Die Firmware kann ebenfalls ausgetauscht werden, wenn neue Funktionen oder Fehlerbereinigungen verfügbar sind.

### 9.1 Bootloader

Der Bootloader tritt in der SimBox nur in Erscheinung, wenn eine neue Konfiguration oder eine neue Firmware in den Flash-Speicher geschrieben werden soll. Er nimmt bei einem Update die neuen Daten von einer Client-Applikation entgegen und schreibt sie in den Speicher. Obwohl beim Einschalten der SimBox zuallererst der Bootloader gestartet wird, bleibt er im normalen Betrieb unsichtbar im Hintergrund und zeigt sich erst auf ausdrückliche Anforderung. In Ausnahmesituationen kann er jedoch entscheiden, die Firmware nicht zu starten und stattdessen ein Update anzufordern.

Wenn sich der Bootloader im Update-Modus befindet, zeigt er dies im Display der SimBox mit unterschiedlichen Meldungen an. Diese werden in den folgenden Abschnitten beschrieben. In allen Fällen wird in der rechten unteren Ecke des Displays angezeigt, wenn der Bootloader mit dem SimBox-Client auf dem PC kommuniziert – dann erscheint dort ein sich drehender Balken. Bleibt der Balken längere Zeit stehen (>10 s) oder zeigt sich dort ein Buchstabe, findet kein Datenaustausch statt.

Nach dem erfolgreichen Abschluss eines Updates der Firmware oder der Konfiguration versucht der Bootloader, die SimBox in einen Standby-Modus zu schalten. In Abhängigkeit von den äußeren Umständen kann die SimBox sich auch sofort wieder einschalten, z. B. wenn die SimBox an einen laufenden Testaufbau mit Aktivität auf den CAN-Bussen angeschlossen ist und von dieser sofort wieder aufgeweckt wird.

**Achtung: Ein Update der Konfiguration oder der Firmware sollte niemals in einem laufenden System vorgenommen werden! Je nach eingespielter Konfiguration kann die SimBox den Betrieb der anderen Geräte an einem CAN-Bus stören, deshalb soll nach einem Update und vor Anschluss der SimBox an ein bestehendes System erst die korrekte Funktion dieser Konfiguration sichergestellt werden.**

Der Bootloader ist mit einer Timeout-Funktion ausgestattet. Wenn innerhalb einer bestimmten Zeitspanne (ca. 40 s) keine Kommunikation mit einem Client stattfindet, schaltet sich die SimBox ebenfalls in einen Standby-Zustand.

**Achtung: Wenn ein Update-Vorgang unterbrochen wird, kann das Gerät in einem nicht funktionsfähigen Zustand verbleiben. Bitte stellen Sie sicher, dass die SimBox an einer stabilen Spannungsversorgung angeschlossen ist und der Client auf dem PC störungsfrei durchlaufen kann.**

### 9.2 Aufruf durch die Firmware

Nach dem Einschalten der SimBox startet normalerweise die Firmware, liest die aktive Konfiguration und führt die dort niedergelegten Anweisungen aus. Soll eine neue Firmware oder eine neue Konfiguration in die SimBox eingespielt werden, muss nicht nur die entsprechende Funktion im SimBox-Client gestartet werden, vor dem Start des Update-Vorgangs muss sich die SimBox im Update-Modus befinden. Dies muss über die Firmware erfolgen, und damit ist der nötige Bedienschritt auch abhängig von der Firmware-Version, welche sich vor dem Update auf der SimBox befindet.

Um die SimBox z. B. bei einer Firmware-Version 1.0.5 in den Update-Modus zu schalten, werden im rechten Seite der SimBox-Tastatur gleichzeitig die beiden Tasten ‚S‘ und ‚Down‘ gedrückt:



Abbildung 26: Aufruf des Bootloaders

Auf dem Display der SimBox erscheint daraufhin die folgende Meldung des Bootloaders. In der rechten unteren Ecke des Displays befindet sich ein Buchstabe, welcher sich bei Kontaktaufnahme durch einen Client in einen rotierenden Balken verwandelt.

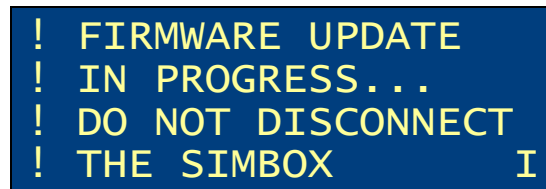


Abbildung 27: SimBox-Display bei Firmware-Update

Bei älteren Versionen der SimBox-Firmware (z. B. v1.0.1) findet eine andere Tastenkombination Verwendung: Dort sind gleichzeitig die Tasten ‚F9‘, ‚F10‘ und die ‚Down‘-Taste aus dem rechten Steuerfeld zu drücken.

### 9.3 Start nach Update-Abbruch

In einigen Fällen kann der Bootloader erkennen, wenn ein Update der Firmware oder der Konfiguration nicht erfolgreich war. Er vermutet dann, dass die Firmware unvollständig und nicht lauffähig ist. In diesem Zustand könnte sie auch nicht mehr durch den Anwender bedient werden, ein Aufruf des Bootloaders zu einem neuerlichen Update-Versuch wäre dann nicht mehr möglich.

Ein anderer Fehlerfall ist, dass sich keine Firmware in der SimBox befindet, welche der Bootloader starten könnte. Natürlich ist auch in diesem Fall eine Aktivierung des Bootloaders durch die Firmware nicht möglich.

Deshalb startet der Bootloader in diesen Fällen direkt in den Update-Modus, und ein Anwender kann sofort einen erneuten Update-Versuch unternehmen. Der SimBox-Bootloader zeigt diesen Betriebszustand durch eine eigene Anzeige im Display dar (Abbildung 28). Versuchen Sie bitte, bei Erscheinen dieser Anzeige den letzten Update-Versuch zu wiederholen (Firmware oder Konfiguration).

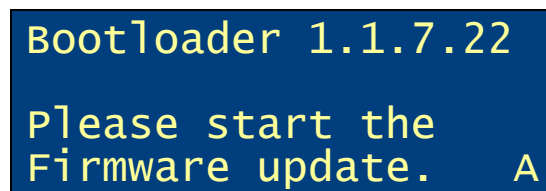


Abbildung 28: SimBox-Display nach Abbruch eines Updates

Wenn die Aufforderung des SimBox-Clients erscheint, die SimBox in den Update-Modus zu schalten, muss an der SimBox keine weitere Bedienung mehr erfolgen – der Update-Modus ist bereits aktiviert. Beachten Sie bitte, dass der Timeout des Bootloaders aktiv ist und sich die SimBox ausschalten kann, bevor der Client den Datenaustausch beginnt. Wiederholen Sie in diesem Fall bitte das Update.

Der Bootloader unterscheidet bei einer Wiederholung von abgebrochenen Updates nicht zwischen den Speicherbereichen für Firmware und Konfiguration. Dadurch kann eine Situation entstehen, wo nach einem fehlerhaften Update der Firmware mit nachfolgendem erfolgreichem Update der Konfiguration der Bootloader versucht, eine nur unvollständig geflashte Firmware zu starten. Dann

kann der Update-Modus nicht mehr durch Bedienung in der Firmware aufgerufen werden. Bitte folgen Sie in diesem Fall den Beschreibungen im nächsten Kapitel.

## 9.4 Start mittels Tastenkombination

In manchen Fällen kann die SimBox in einen Zustand geraten, wo weder die Firmware in der Lage ist, durch Bedienschritte den Update-Modus zu aktivieren, noch der Bootloader selbst einen Fehlerzustand erkennt und automatisch in den Update-Modus wechselt. Solange der Bootloader selbst noch intakt ist, kann mittels einer Tastenkombination beim Einschalten der SimBox der Update-Modus aufgerufen und eine neue Firmware geladen werden.

Um diesen Modus zu aktivieren, stellen Sie zuerst sicher, dass die SimBox ausgeschaltet ist bzw. sich im Standby-Zustand befindet. Der Standby-Zustand zeichnet sich durch eine geringe Stromaufnahme aus, das Display ist ausgeschaltet. Mit defekter Firmware wird die SimBox meist ebenfalls nichts im Display anzeigen. Wenn Sie den Betriebszustand der SimBox nicht feststellen können, dann trennen Sie sie einfach von der Spannungsversorgung.

Nun halten Sie gleichzeitig die Tasten ‚F9‘, ‚F10‘ und die ‚Down‘-Taste aus dem rechten Steuerfeld gedrückt. In diesem Zustand starten Sie die SimBox durch Anstecken an die Spannungsversorgung oder durch zusätzliches Drücken der Taste ‚S‘. Bei erfolgreichem Start des Bootloaders erscheint im Display der SimBox die in Abbildung 29 dargestellte Anzeige.

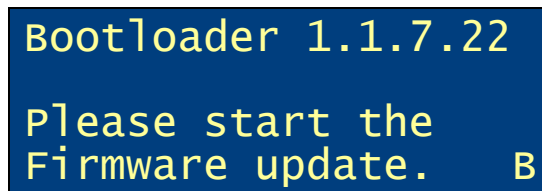


Abbildung 29: SimBox-Display bei Anwendung der Tastenkombination

Hinweis: Bereiten Sie am besten zuerst den SimBox-Client für den neuen Update-Versuch vor und schalten Sie erst danach die SimBox ein, so dass Sie umgehend den Update-Vorgang starten können. Der Timeout des Bootloaders ist auch in diesem Fall aktiviert und kann zum Ausschalten der SimBox führen, während der Client vorbereitet wird.

## Anhang A: Technische Daten und Support

### Allgemeine Daten

Spannungsversorgung	8,5V..16V, 12V (typ.)
Stromaufnahme (ca.)	200mA...250mA (depending on display backlight)
Standby-Verbrauch (ca.)	0,1mA
Temperaturbereich (in Betrieb)	0 °C to 50 °C (32 °F to 122 °F)
Temperaturbereich (Lagerung)	-20 °C to 70 °C (-4 °F to 158 °F )
Gewicht (ca.)	370g (0,81 pounds)

### Gehäuse

Abmessungen (ca.)	154mm x 85mm x 25mm (6.81" x 3.35" x 0.98")
Bedienelemente	Folientastatur mit 25 Tasten
Anzeigen	Alphanumerisches Display, 20 Zeichen x 4 Zeilen
Anschlüsse	26-pin D-Sub Stecker (Spannungsversorgung und CAN-Busse) 9-pin D-Sub Buchse (RS232, Konfiguration und Firmwareupdate)

### Support:

<http://www.telemotive.de/>

<mailto:produktsupport@telemotive.de>

## Anhang B: Anschlussbelegung

Die SimBox verfügt über zwei Anschlüsse im D-SUB-Format. Ein 26-poliger Stecker an der rechten Seite führt alle für eine Simulation notwendigen Signale inklusive der Spannungsversorgung und ein 9-poliger Stecker an der linken Seite dient zur Verbindung mit einem PC. Hier können Konfigurationen und Updates auf die SimBox übertragen werden. Der 9-polige Anschluss entspricht der üblichen RS232-Belegung und kann direkt an einen COM-Port eines PCs angeschlossen werden.

26-poliger D-SUB Anschluss (DA26), rechte Seite:

Pin	Beschreibung
1, 2	Clamp 30 (positive supply voltage)
3	Digital out 0
4	CAN channel 2: CAN_H, low speed
5	CAN channel 1: CAN_H, low speed
6	Analog input 3: ground
7	Analog input 2: ground
8	Analog input 1: ground
9	Analog input 0: ground
10, 11	Clamp 31 (negative supply voltage)
12	Digital out 1
13	CAN channel 2: CAN_L, low speed
14	CAN channel 1: CAN_L, low speed
15	Analog input 3: signal
16	Analog input 2: signal
17	Analog input 1: signal
18	Analog input 0: signal
19	CAN channel 2: CAN_L, high speed
20	CAN channel 2: CAN_H, high speed
21	CAN channel 1: CAN_L, high speed
22	CAN channel 1: CAN_H, high speed
23	Digital input 3
24	Digital input 2
25	Digital input 1
26	Digital input 0

9-poliger D-SUB Anschluss (DE9), linke Seite:

Pin	Beschreibung
1	n.c.
2	TxD (SimBox, connects to RxD of PC)
3	RxD (SimBox, connects to TxD of PC)
4	n.c.
5	GND
6	n.c.
7	RTS
8	CTS
9	n.c.



**Anhang C: Änderungsindex**

<b>Version</b>	<b>Datum</b>	<b>Änderung</b>	<b>Autor</b>
1.0	17.12.2012	Erste Version	Helmut Sochor