

Script-Sprache für UPT und MKT-View II / III / IV

Einleitung, Anwendungsfälle, Programmierung

MKT Systemtechnik

Autor: Wolfgang Büscher
Dipl.-Ing. Soft- und Hardware-Entwicklung
buescher@mkt-sys.de

Stand: 2016-08-18 (JJJJ-MM-DD)

Ablage: art85133_Einfuehrung_MKTview_Scriptsprache.odp/pdf; Online: [hier](#)

Hinweis: Diese Präsentation enthält Links zur Dokumentation auf der [MKT-Webseite](#). In den meisten PDF-Readern müssen Hyperlinks extra 'erlaubt' werden, z.B. Foxit:

Edit .. Preferences .. Trust Manager .. [-] Enable Safe Reading Mode

Inhaltsübersicht

Teil 1 : Einleitung

- Applikationen “mit und ohne Script”
- Die Entwicklungsumgebung (Programmiertool)
- Installation und Inbetriebnahme der Software
- Übertragen einer Applikation vom PC in's Zielsystem (per CAN, Speicherkarte oder Web-Browser)

Teil 2 : Typische Anwendungsfälle der Script-Sprache

- Reaktion auf 'besondere' Benutzereingaben (Events)
- **Selbstdefinierte 'Popup-Menüs' mit variablem Inhalt** (-> Beispiel in Online-Hilfe; Datei “popup1.cvt”)

Teil 3 : Programmierung und Testmöglichkeiten ... t.b.d.

Script-Sprache für UPT und MKT-View II / III / IV

Teil 1 : Einleitung

Vorteil von Applikationen mit Script-Sprache

im Vergleich mit älteren Applikationen (ohne Script) :

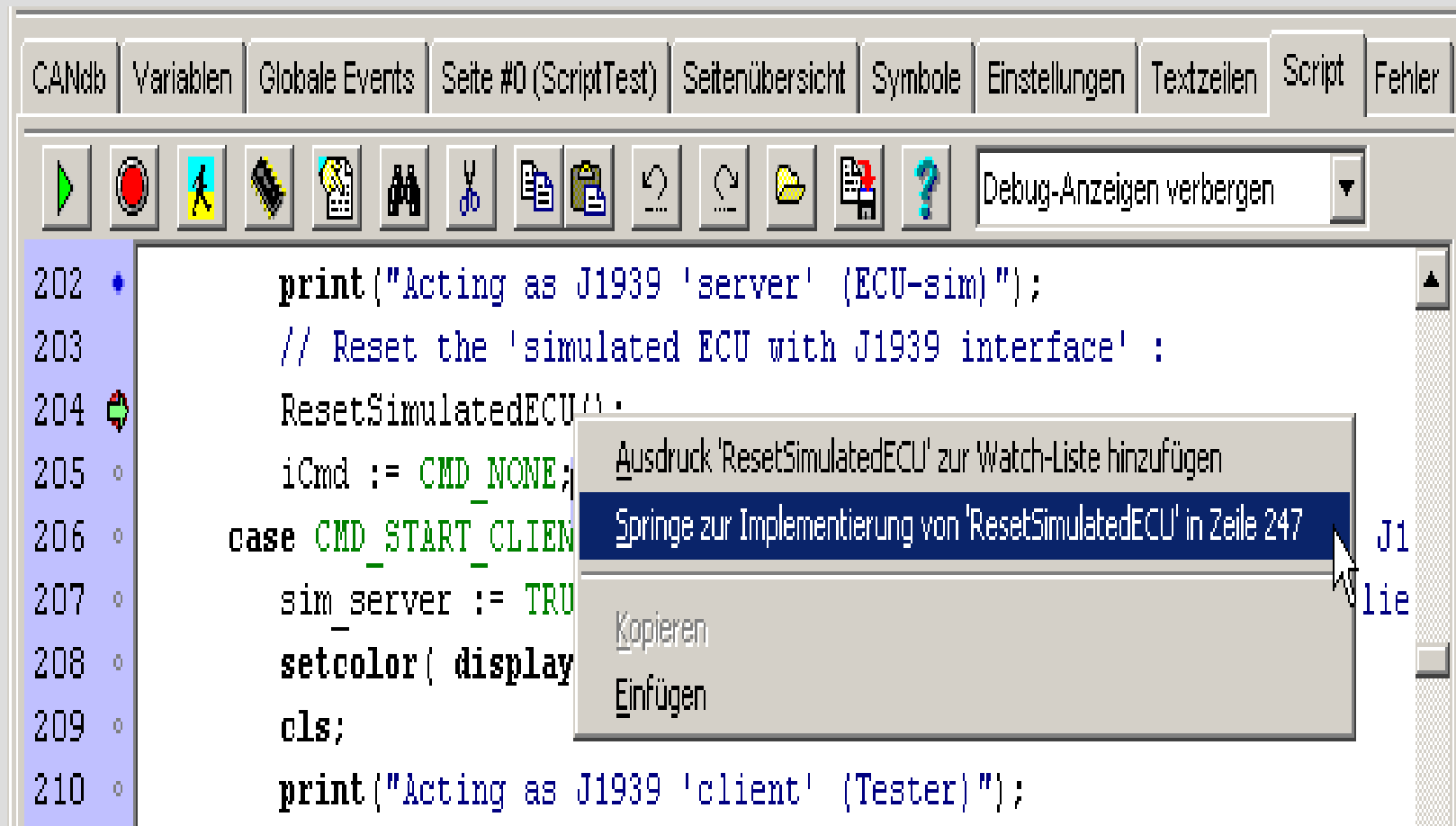
- Wesentlich schneller, denn Scripte werden *compiliert* (die Display-Event-Abfragen werden direkt *interpretiert*)
- Hochsprache, vergleichbar mit IEC 61131 Structured Text
- Spaghetti-Code kann vermieden werden, Programm bleibt auch bei großem Umfang überschaubar
- Ereignisse können (im Gegensatz zu den 'Display-Events') nicht nur einmal pro 'main loop', sondern wenige Millisekunden nach dem Eintreffen verarbeitet werden
- CAN-Telegramme können direkt per Script verarbeitet werden, was z.B. die Implementierung eigener Protokolle ermöglicht

Nachteil :

- Script-Sprache ist nur für Geräte mit 32-Bit-CPU verfügbar
- Für den Entwickler ist eine etwas längere Einarbeitungszeit erforderlich

Script-Sprache für UPT und MKT-View II / III / IV

Einleitung : Entwicklungsumgebung (Programmiertool)



The screenshot shows a software interface with a menu bar at the top containing: CANdb, Variablen, Globale Events, Seite #0 (ScriptTest), Seitenübersicht, Symbole, Einstellungen, Textzeilen, Script, Fehler. Below the menu bar is a toolbar with various icons for execution and debugging, and a dropdown menu labeled "Debug-Anzeigen verbergen". The main area is a code editor with a line number column on the left (202-210) and a vertical scrollbar on the right. The code is as follows:

```
202 print("Acting as J1939 'server' (ECU-sim)");
203 // Reset the 'simulated ECU with J1939 interface' :
204 ResetSimulatedECU();
205 iCmd := CMD_NONE;
206 case CMD_START_CLIENT
207     sim_server := TRUE;
208     setcolor( display );
209     cls;
210     print("Acting as J1939 'client' (Tester)");
```

A context menu is open over the function call `ResetSimulatedECU()` on line 204. The menu items are:

- Ausdruck 'ResetSimulatedECU' zur Watch-Liste hinzufügen
- Springe zur Implementierung von 'ResetSimulatedECU' in Zeile 247
- Kopieren
- Einfügen

The second menu item is highlighted, and a mouse cursor is pointing at it. To the right of the code editor, the text "J1" and "lie" is partially visible.

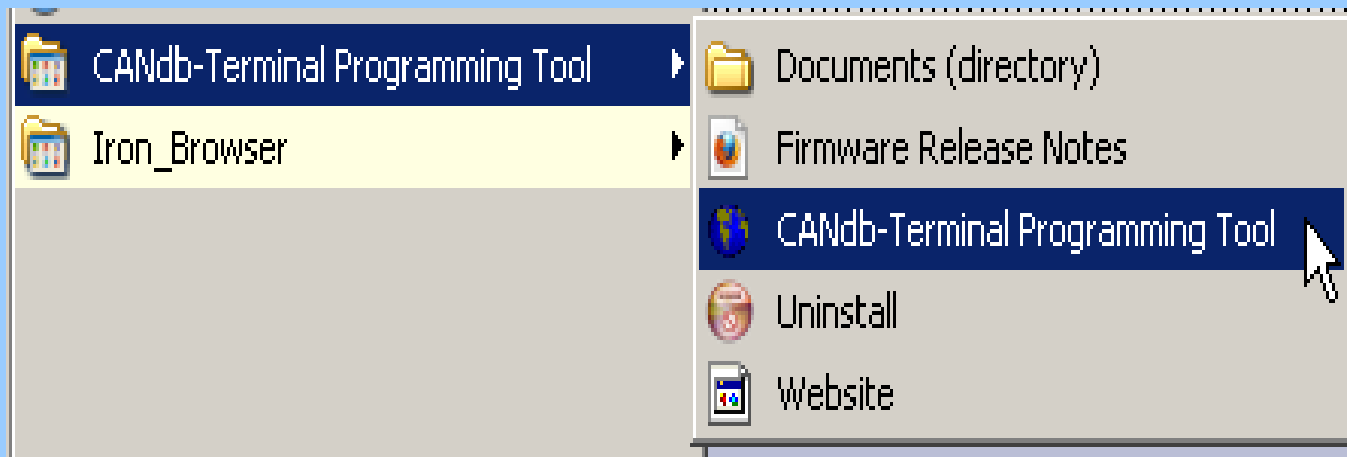
Details zum Script-Editor / Debugger im [Hilfesystem](#) !

Script-Sprache für UPT und MKT-View II / III / IV

Installation und Start des Programmierertools

(Auszug aus [Präsentation Nr. 85131](#))

- Herunterladen von www.mkt-sys.de (Downloads, Software, InstallCANdbTerminal.exe)
- Um Probleme mit Windows-UAC zu vermeiden:
Installation nach C:\MKT\CANdbTerminalProgTool\... (!)
- Aufruf per Windows-Start-Menü: 'Alle Programme' ..



'Documents': Verzeichnis mit allen Dokumenten
und druckbaren Handbüchern im PDF-Format
(darunter auch diese Präsentation)

Script-Sprache für UPT und MKT-View II / III / IV

Hauptfenster des Programmiertools (Auszug aus Präsentation Nr. 85131)

Programmiertool für "CANdb"-Terminals - C:\cbproj\UptWin1\programs\MKTview2\MV2_Demo.CVT

Datei Assistent Bearbeiten Transfer Ansicht Optionen Werkzeuge Icon Run! Stop! Reset! Hilfe

CANdb Variablen Globale Events Seite #0 (Start0) Seitenübersicht Symbole Einstellungen Textzeilen Script Fehler

Editor Definitions-Kopf Anzeigzeilen-Definitionen Event-Definitionen Weitere Kommandos

Nr	Name	X	Y	W	H	Text/Anzg.-Kommando	Var/Form	Zugriff	Zeiche	Farbe	Zoom
0		0	0			\fr(0,0,432,84)\fr(2		0	0	0,0	100
1		24	3			\c(ti/6&7)M\c(ti/4&7		0	0	-1,-2	124
2		120	65			Systemtechnik		0	0	-1,-2	102
3		5	87			User		0	0	4,-2	112
4		16	121			Programmable		0	0	6,-2	112
5		16	144			Terminal		0	0	2,-2	112
6		256	88			****	rtc.yr	0	0	-1,1,-	100
7		320	88			\-**-	rtc.mo	0	0	-1,1,-	100
8		384	88			**	rtc.da	0	0	-1,1,-	100
9		288	120			**	rtc.hr	0	0	-1,1,-	100
10		320	120			:**:	rtc.mi	0	0		
11		384	120			**	rtc.se	0	0		
12	PST	454	6	20	100	\bar(\$045,5,\$BBEE,\$1pscan.ti		0	0		
13		424	112	50	40	\btn(\$2,"\\chr(0x1B)	#0				
14		424	160	50	40	\btn(\$2,"\\chr(0x1A)	#0				
15		424	208	50	40	\btn(\$2,(v==0)"Day-night	#0				
16		0	232	50	40	\btn(\$2,"Help",0,c"F	#0				
17		56	232	50	40	\btn(\$2,"Menu",0,g"M	#0				
18		112	232	50	40	\btn(\$2,"Test",0,g"S	#0				

Eigenschaften einer Anzeigzeile Nr. 4
 X-pos 16 Y-pos 121 Typ Text
 Breite 0 Höhe 0 Font <user#1> (10)
 Z.-Modus NORMAL (0) Zugriff Read Only (0)
 Variable --none-- Farben 6 -2 -2
 X-Zoom: 2 Y: 1
 Control ID
 Menüeintrag
 immer komplett neu zeichnen
 per Touchscreen bedienbar

Text Symbol Schaltfläche Balken Diagramm

Programmable
 Basis 0 (2=binär, 8=oktal, 10=dezimal, 16=hex)

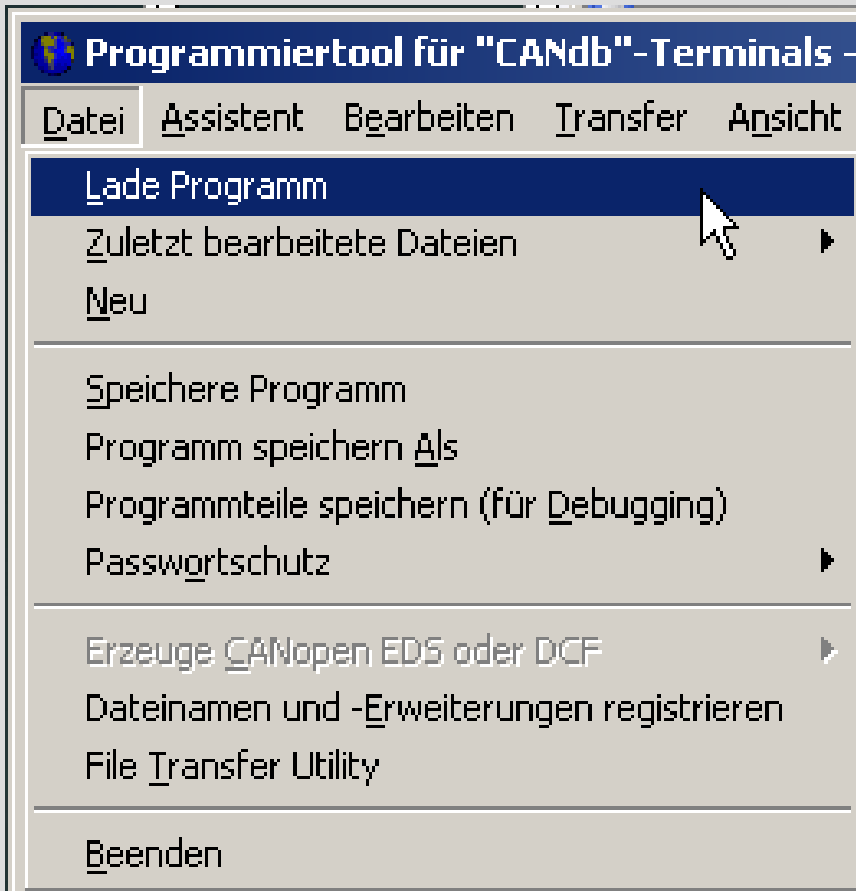
UPT-Simulator: page 0

Rückgg. Neu.. Löschen [6, 5]
 Anwenden Kopieren Einfügen Mehr.. ?

Simulator Gestoppt Warning: Parameter "nr_display_pages" limited from 256 to 255

Laden eines bestehenden Anzeigeprogramms

(Auszug aus Präsentation Nr. 85131)



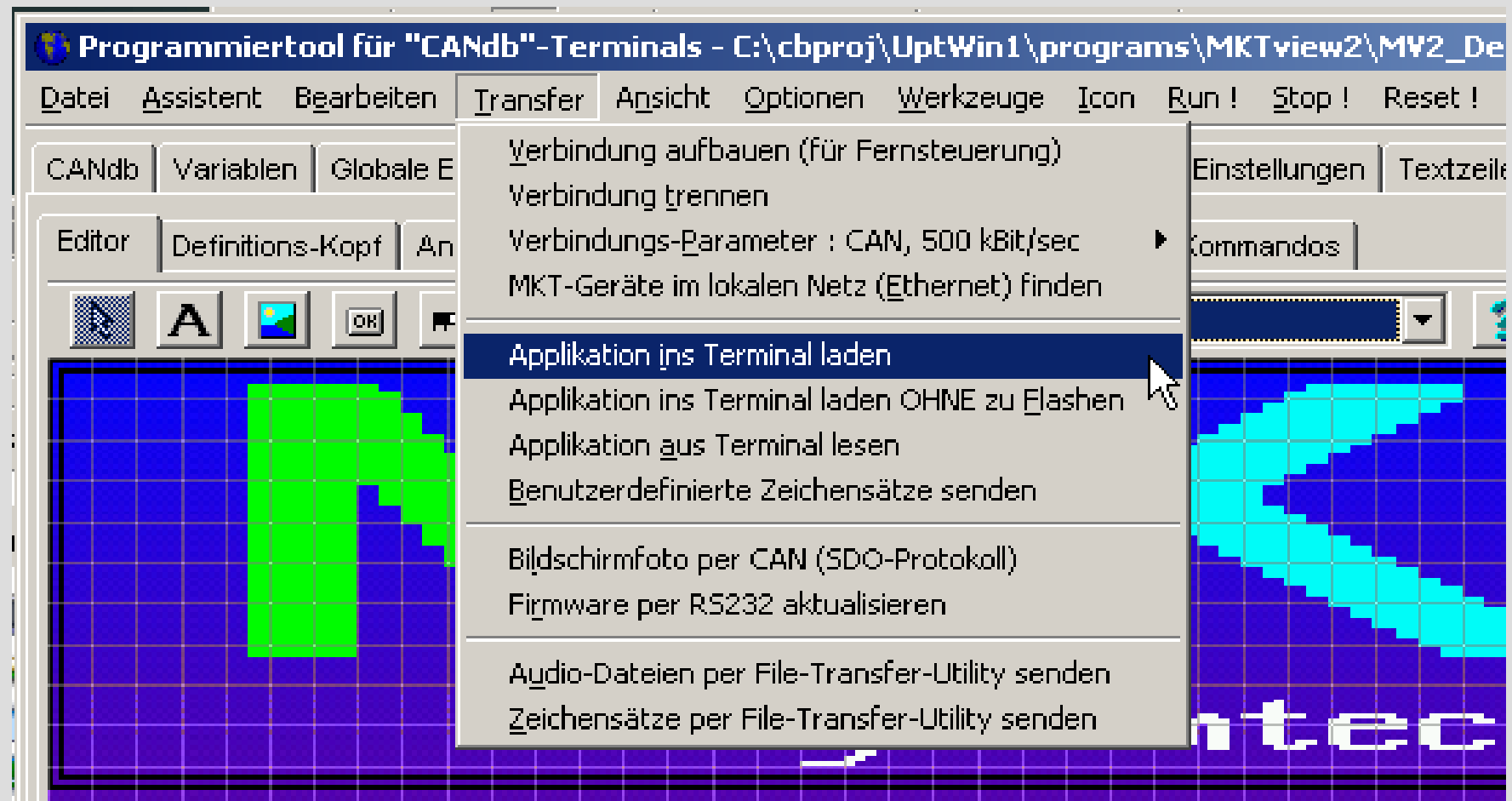
- Empfehlung für den Einstieg :
Eins der im Programmierwerkzeug
enthaltenden 'Demos' laden,
z.B. 'MV2_Demo.cvt'
für MKT-View II / III
bzw. 'MV4_Demo.cvt'
für MKT-View IV

(mit Beispielen für verschiedene
Typen von Anzeigeseiten,
z.B. numerisch, Balken,
Buttons, Tabellen, etc)

Übertragen des Programms vom PC per CAN-Bus

(Auszug aus Präsentation Nr. 85131)

- Auf Register 'Einstellungen' das CAN-Interface auswählen
- Im Menü 'Transfer' das Medium auf 'CAN' setzen, dann :



Übertragen des Programms vom PC per Speicherkarte

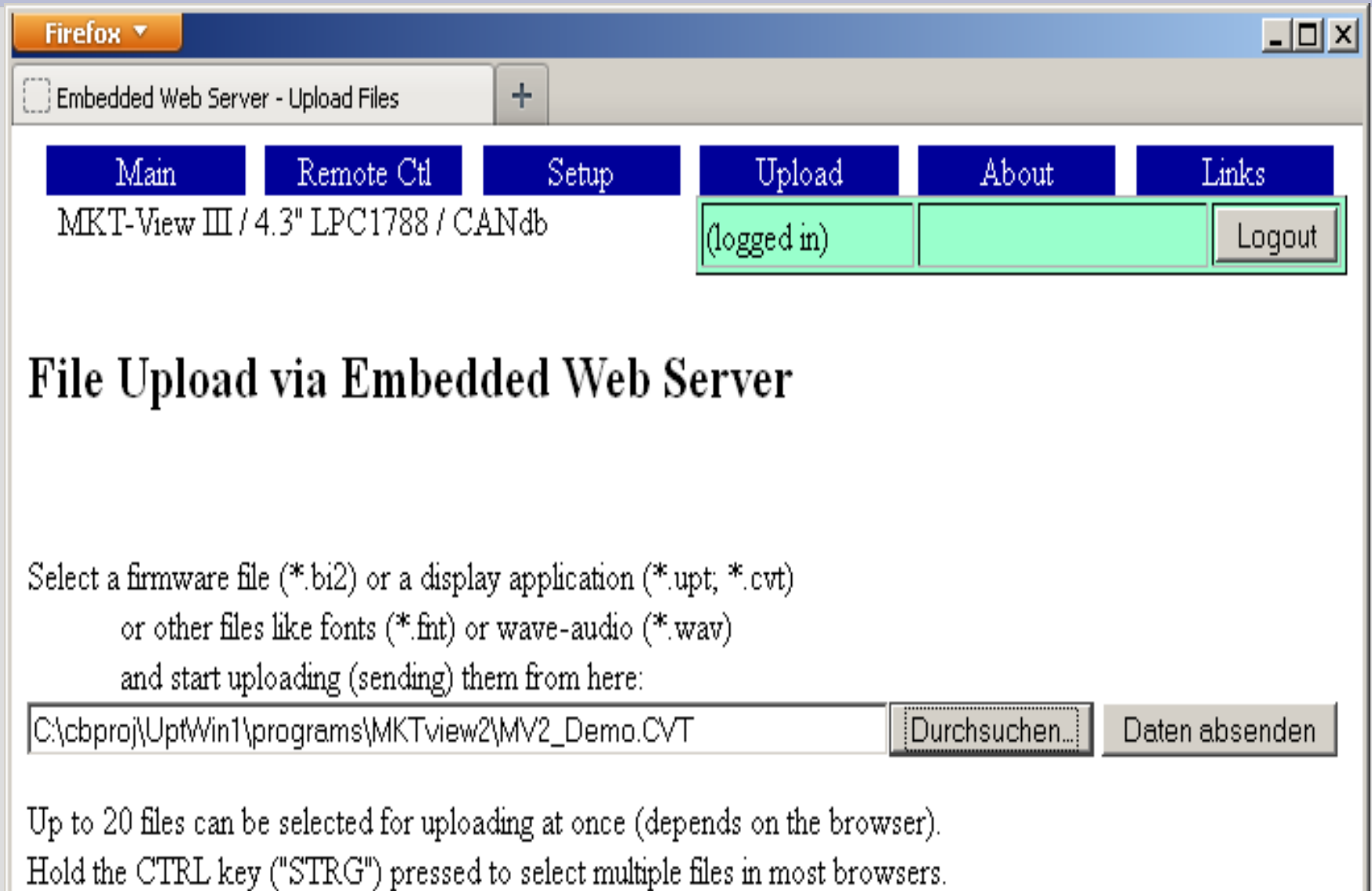
(Auszug aus Präsentation Nr. 85131)

- Die Display-Applikation (*.cvt) in das Wurzelverzeichnis auf eine Speicherkarte mit FAT-Dateisystem kopieren
- Speicherkarte 'sicher entfernen' und in das Gerät einsetzen
- Gerät einschalten und Systemmenü aufrufen (F2 + F3)
- 'Load Program from FILE', Datei auswählen, ENTER .

```
Main system menu (2)
EXIT ?
Load program from FILE
Transfer via CAN = ON
Other transfers ... ▶
Audio Recorder
User Settings
System Setup
System Test
Network Setup
Diagnostics
General Settings
POWER-flag = TRUE
Current page =001
Run Mode =001
```

Übertragen des Programms vom PC per Browser / HTTP

(Auszug aus Präsentation Nr. 85131)



The screenshot shows a Firefox browser window with the address bar displaying "Embedded Web Server - Upload Files". The page features a navigation menu with buttons for "Main", "Remote Ctl", "Setup", "Upload", "About", and "Links". Below the menu, the text "MKT-View III / 4.3" LPC1788 / CANdb" is visible. A green box highlights the "Upload" button, which contains the text "(logged in)", and a "Logout" button. The main heading is "File Upload via Embedded Web Server". Below this, instructions state: "Select a firmware file (*.bi2) or a display application (*.upt; *.cvt) or other files like fonts (*.fnt) or wave-audio (*.wav) and start uploading (sending) them from here:". A file input field contains the path "C:\cbproj\Upt\Win1\programs\MKTview2\MV2_Demo.CVT", with "Durchsuchen..." and "Daten absenden" buttons. A footer note says: "Up to 20 files can be selected for uploading at once (depends on the browser). Hold the CTRL key ("STRG") pressed to select multiple files in most browsers."

Script-Sprache für UPT und MKT-View II / III / IV

Ende der Einleitung

Weiterführende Informationen (Web-Links) :

- [Online-Hilfesystem zum Programmierwerkzeug von MKT](http://www.mkt-sys.de/MKT-CD/upt/help/progt_49.htm)
www.mkt-sys.de/MKT-CD/upt/help/progt_49.htm
- ['Druckbare' Handbücher von MKT im PDF-Format](http://www.mkt-sys.de/Hilfesystem_aus_Terminal_Programmierwerkzeug/heupt)
www.mkt-sys.de/Hilfesystem_aus_Terminal_Programmierwerkzeug/heupt
- [Inhaltsverzeichnis der 'MKT-CD' mit Download-Links](http://www.mkt-sys.de/MKT-CD/index.htm)
(Programmierwerkzeuge, CAN-Tester, CAN-Logger, Utilities)
www.mkt-sys.de/MKT-CD/index.htm
- [Homepage von MKT Systemtechnik](http://www.mkt-sys.de)
www.mkt-sys.de

Script-Sprache für UPT und MKT-View II / III / IV

Teil 2 : Typische Anwendungsfälle von Scripten

- Verknüpfen von Signalen (d.h. Erzeugen "berechneter" Signale),
- Implementieren von Protokollen (auch für CAN), die in der Firmware nicht direkt implementiert sind, z.B. J1939;
- Verarbeiten von Ereignissen, deren Komplexität die Implementierung als 'Event' (in der Anzeige-Seiten-Definition) nicht zulässt,
- Lesen und Schreiben von Dateien, z.B. für Event-Logs, automatisch erzeugte Fehlerprotokolle, usw.
- Realisieren einfacher (SPS-ähnlicher) Ablaufsteuerungen, allerdings ohne 'harte' Echtzeitanforderungen
- Implementieren von Algorithmen, die mit den Display-eigenen 'Event'-Definitionen zu aufwändig oder zu unübersichtlich wären.

weitere Einsatzmöglichkeiten im Handbuch ...

Script-Sprache für UPT und MKT-View II / III / IV

Event - Handler

Beispiel für 'Ereignisse'

- Empfang eines bestimmten CAN-Telegramms
- Ablauf eines Timers
- Tastendruck, Touchscreen, Drehknopf
- Interaktion des Bedieners mit einem der Anzeigeelemente (z.B. graphische Buttons wie im folgenden Beispiel)

Eine komplette Übersicht aller in der Script-Sprache möglichen Event-Handler finden Sie im online verfügbaren Handbuch zur [Script-Sprache](#) !

Script-Sprache für UPT und MKT-View II / III / IV

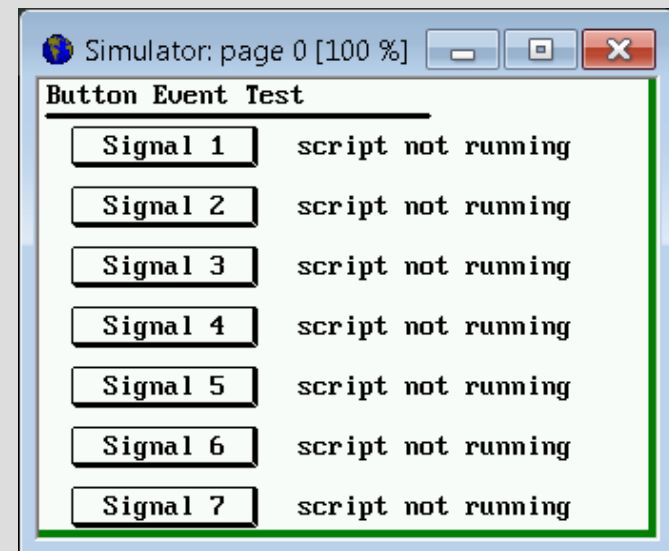
Beispielprogramm : Button-Event-Demo (1)

Quelle (bei Installation des Tools im empfohlenen Verzeichnis):

C:\MKT\CANdbTerminalProgTool\Programs\script_demos\
ButtonEventDemo.cvt

Per Doppelklick auf die Datei im Explorer (oder Total Commander) wird automatisch das Programmierwerkzeug gestartet, und die Applikation (inkl. des in der CVT-Datei enthaltenen Scripts) geladen.

Die erste Anzeigeseite erscheint im Simulator, das Script läuft aber noch nicht !

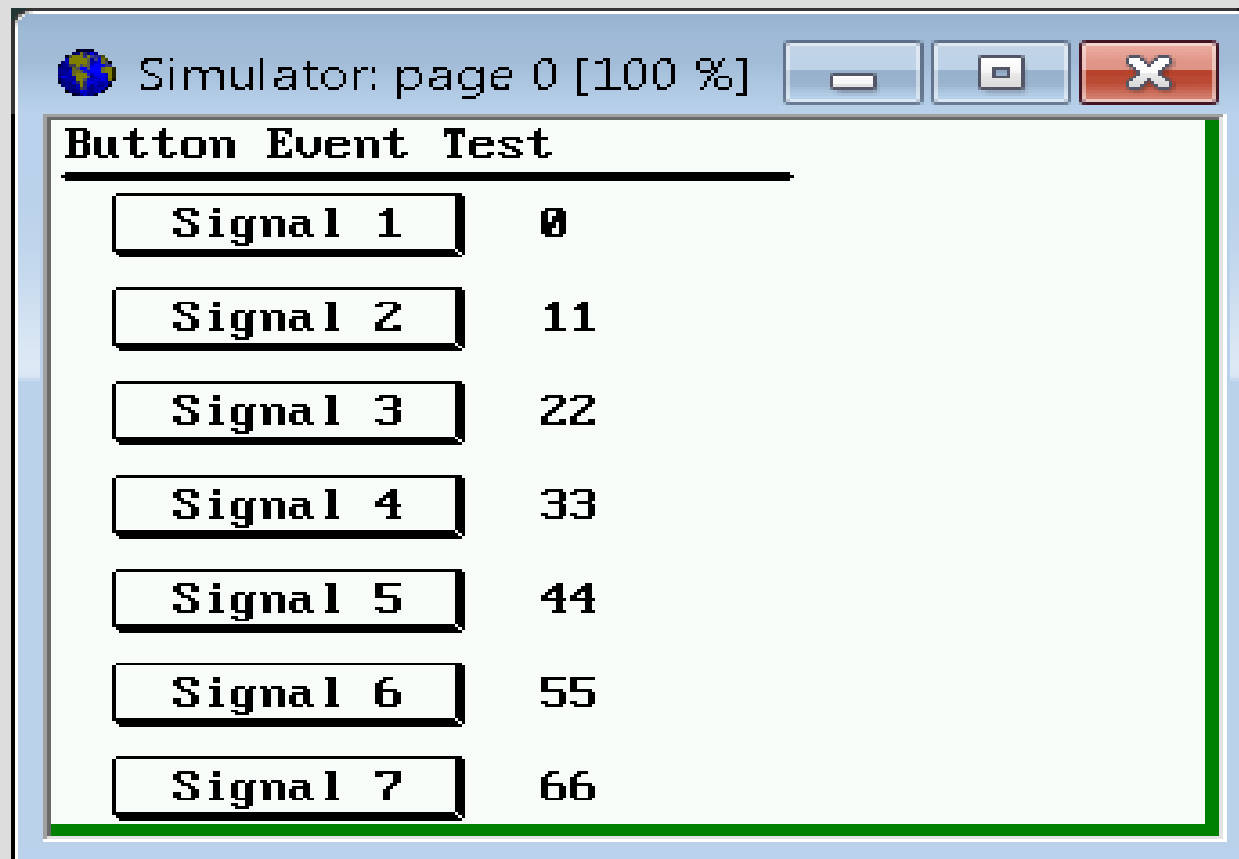


Script-Sprache für UPT und MKT-View II / III / IV

Beispielprogramm : Button-Event-Demo (2)

Durch Klicken auf "Reset" im Hauptmenü des Tools wird ein 'Kaltstart' des Gerätes simuliert.

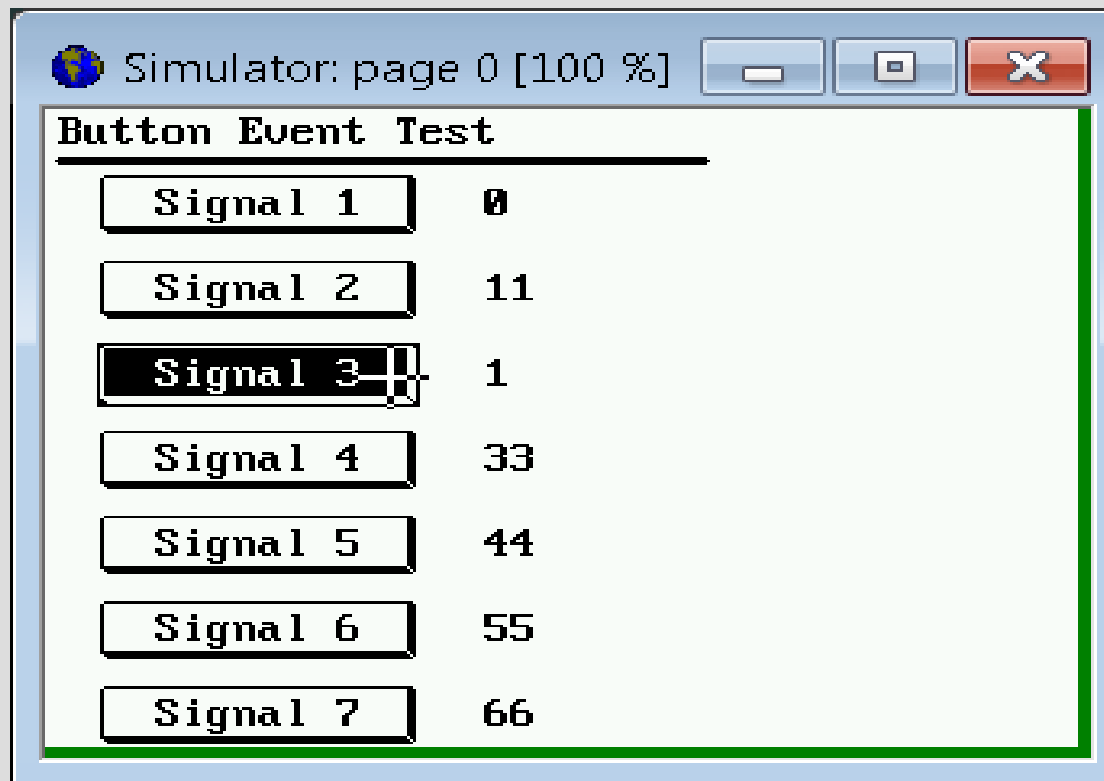
Dabei wird u.A. das Script compiliert und gestartet.



Script-Sprache für UPT und MKT-View II / III / IV

Beispielprogramm : Button-Event-Demo (3)

Im Simulatorfenster einen der Buttons per Maus drücken und gedrückt halten. Dies simuliert den Touchscreen im MKT-View.

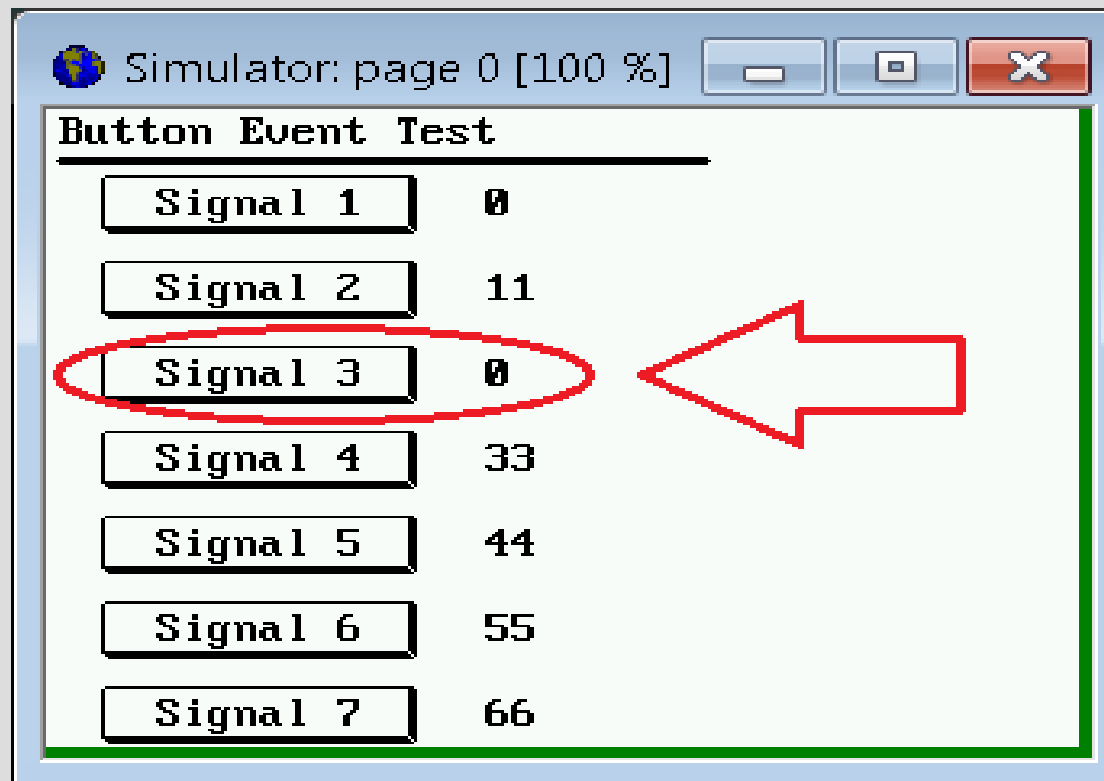


Durch Drücken des Buttons ändert sich der daneben angezeigte Wert. Beim Loslassen ebenfalls...

Script-Sprache für UPT und MKT-View II / III / IV

Beispielprogramm : Button-Event-Demo (4)

Sowohl beim Drücken als auch beim Loslassen des Buttons wird ein 'Event-Handler' im Script aufgerufen.



Der Event-Handler setzt in diesem einfachen Beispiel das Signal beim Drücken auf Eins, und beim Loslassen auf Null. Jedem der sieben Buttons ist ein eigenen Signal zugeordnet.

Script-Sprache für UPT und MKT-View II / III / IV

Beispielprogramm : Button-Event-Demo (5)

Der Event-Handler für die 7 Buttons im Programmierertool:

```
106 //-----
107 // Common handler for all 'visible elements which interact with the user'
108 //   on the current display page, aka "Control Elements"
109 //-----
110
111 func OnControlEvent(
112     int event,      // [in] type of the event, like evClick, etc
113     int controlID, // [in] control identifier (from page-def-table)
114     int param1,    // [in] 1st message parameter, depends on event
115     int param2 )  // [in] 2nd message parameter, depends on event
116 // Called when 'something happens' with a certain control element
117 // (button, menu item, edit field, etc) on the current display page .
118 // param1: client-X-coordinate or keyboard code (depends on event-type)
119 // param2: client-Y-coordinate (where applicable) .
120 local int i;
121 select( event )
122 case evPenDown: // the TOUCH-PEN was just pressed over a display element
123     select( controlID ) // on WHICH control element was the touch pen pressed
124         case idButton1: // 1st button PRESSED, or...
125         case idButton2: // 2nd button PRESSED, or...
126         ...
```

Funktionsname des Event-Handlers

Typ des Ereignisses

Debug-Anzeigen verbergen

Edit 0 : 0 Exec: line 54, PC=0188 script running.

Simulator Läuft Script successfully compiled in APPL_Reset() .

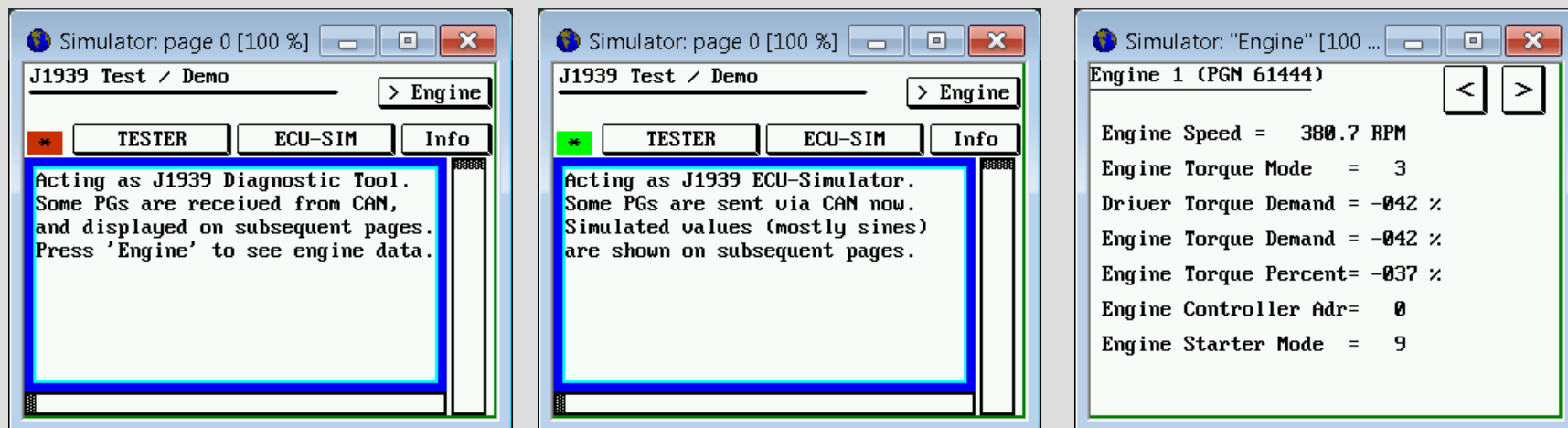
[Details zur Implementierung dieses Event-Handlers in der Online-Hilfe !](#)

Script-Sprache für UPT und MKT-View II / III / IV

Event-Handler und CAN-Bus-Protokolle

Ähnlich wie auf das Drücken und Loslassen eines Buttons kann das Script auch auf andere Ereignisse reagieren.

Mit Hilfe der Event-Handler und [CAN-Bus-Funktionen](#) können im Script z.B. eigene Protokolle implementiert werden (J1939, OBD II, ...). Per Script wurde z.B. die Simulation eines Steuergerätes mit J1939-Diagnose-Interfaces implementiert, und (im gleichen Script) ein einfacher J1939-"Diagnose"-Tester:



[Weitere Informationen zum J1939-Support per Script finden Sie in der Online-Hilfe !](#)