

# Programmierertools für UPT und MKT-View

## Einleitung, Installation, Bedienung

### MKT Systemtechnik

---

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

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

Ablage: art85131\_Einfuehrung\_MKT\_Programmierertools.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**

# Vortragsthemen

## Teil 1 : Programmierertools für UPT und MKT-View

- Vorstellung der Hardware ( z.B. MKT-View I, II, III )
- Installation und Start des Programmierertools
- Hauptfenster mit den wichtigsten Registerkarten
- Laden oder Neu-Erstellen eines Anzeigeprogramms
- Aus CANdb importierte Signale in Variablen umsetzen
- Übertragen des Programms vom PC in das Zielsystem
- Test des Anzeigeprogramms am CAN-Bus

## Teil 2 : Programmierbare Ereignisse und Reaktionen

- Ein einfaches Beispiel : Farbumschaltung per Messwert
- Möglichkeiten und Grenzen der Event-Verarbeitung
- Alternative: Script-Sprache (für Geräte mit 32-Bit-CPU)

## Teil 3 : CAN-Logger ( -> [Dokument Nr. 85132](#) )

## Vorstellung der Hardware (1)

MKT-View / MKT-View "Plus" :



Erster Vertreter der MKT-View Familie, für Einsatz im Fahrzeug.  
"Stand der Technik" anno 2002: 16-Bit CPU, max. 40 MHz Takt, CF.

## Vorstellung der Hardware (2)

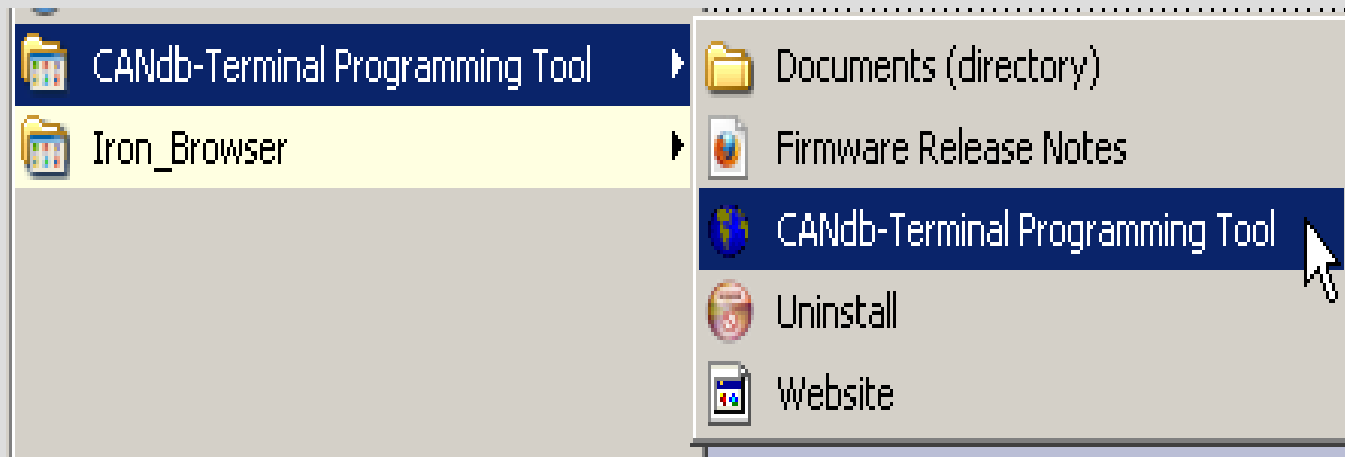
MKT-View II / III :



Zweite Generation der MKT-View Familie: 32-Bit-CPU, 72..96 MHz, 4.3" TFT mit Touch, 480 \* 272 Pixel, Ethernet, SD-Karte.

## Installation und Start des Programmiertools

- Herunterladen von [www.mkt-sys.de](http://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)

# Programmiertool : Hauptfenster

(Details im [Handbuch](#) und in den folgenden Kapiteln)

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	€00
1		24	3			\c(ti/6&7)M\c(ti/4&7		0	0	-1,-2	€24
2		120	65			Systemtechnik		0	0	-1,-2	€02
3		5	87			User		0	0	4,-2	€12
4		16	121			Programmable		0	0	6,-2	€12
5		16	144			Terminal		0	0	2,-2	€12
6		256	88			****	rtc.yr	0	0	-1,1,-	€00
7		320	88			\-**\-	rtc.mo	0	0	-1,1,-	€00
8		384	88			**	rtc.da	0	0	-1,1,-	€00
9		288	120			**	rtc.hr	0	0	-1,1,-	€00
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	0	0		
14		424	160	50	40	\btn(€2,"\\chr(0x1A)	#0	0	0		
15		424	208	50	40	\btn(€2,(v==0)?"Day-night	#0	0	0		
16		0	232	50	40	\btn(€2,"Help",0,c"F	#0	0	0		
17		56	232	50	40	\btn(€2,"Menu",0,g"M	#0	0	0		
18		112	232	50	40	\btn(€2,"Test",0,g"S	#0	0	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

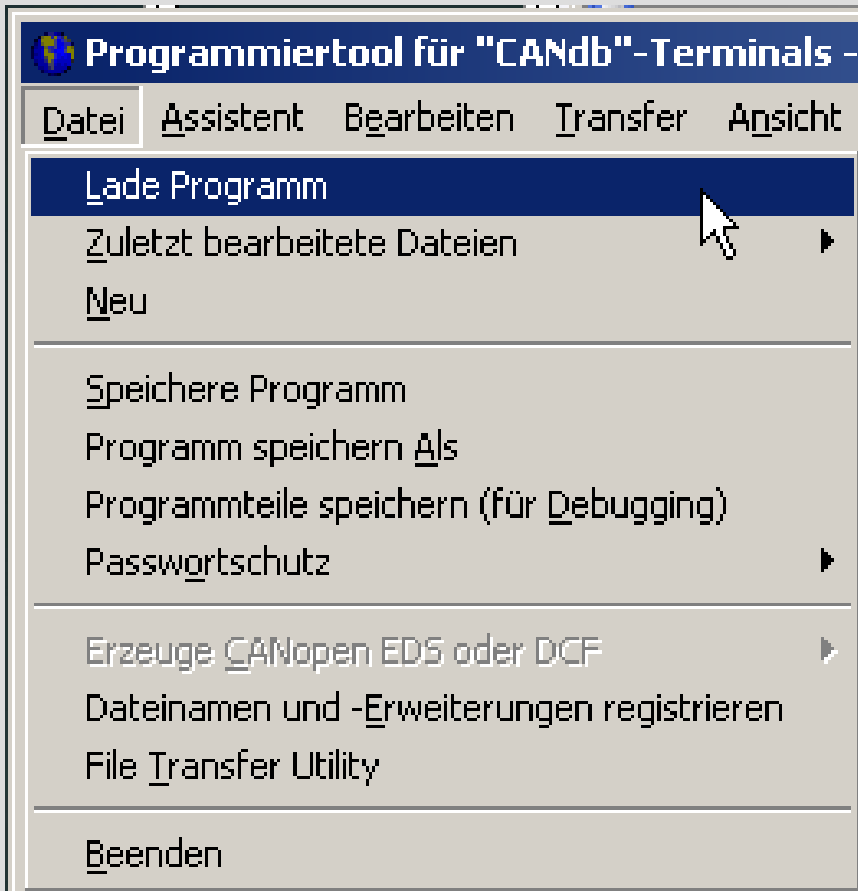
Basis 0 (2=binär, 8=oktal, 10=dezimal, 16=hex)

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



- Empfehlung für den Einstieg :  
Eins der im Programmierwerkzeug  
enthaltenden 'Demos' laden,  
z.B. 'MV2\_Demo.CVT'  
für MKT-View II
- ( mit Beispielen für verschiedene  
Typen von Anzeigeseiten,  
z.B. [numerisch](#), [Balken](#),  
[Buttons](#), [Tabellen](#), [etc](#))

## Alternativ: Erzeugen eines 'neuen' Anzeigeprogramms mit dem Assistenten

Programmiertool für "CANdb"-Terminals - C:\cbproj\UptWin1\programs\MKTview2\MV2\_Demo.CVT

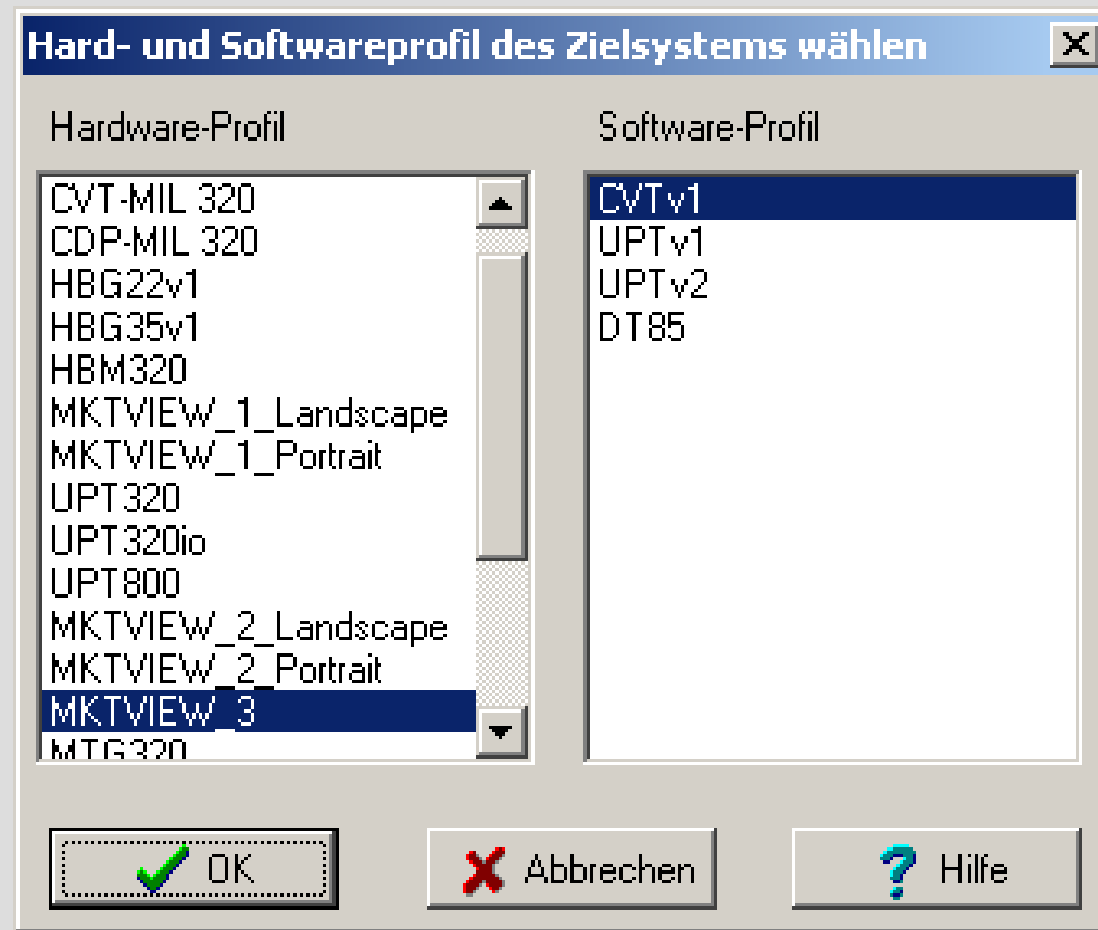
Assistent

- Neues Programm, "einfach"
- Neues Programm, "universell"
- Vordefinierte globale Events (Katalog)

Nr	do	Var/Form	Zugriff	Zeiche	Farbe	Zoom
0			0	0	0,0	\$00
1			0	0	1,0	\$00

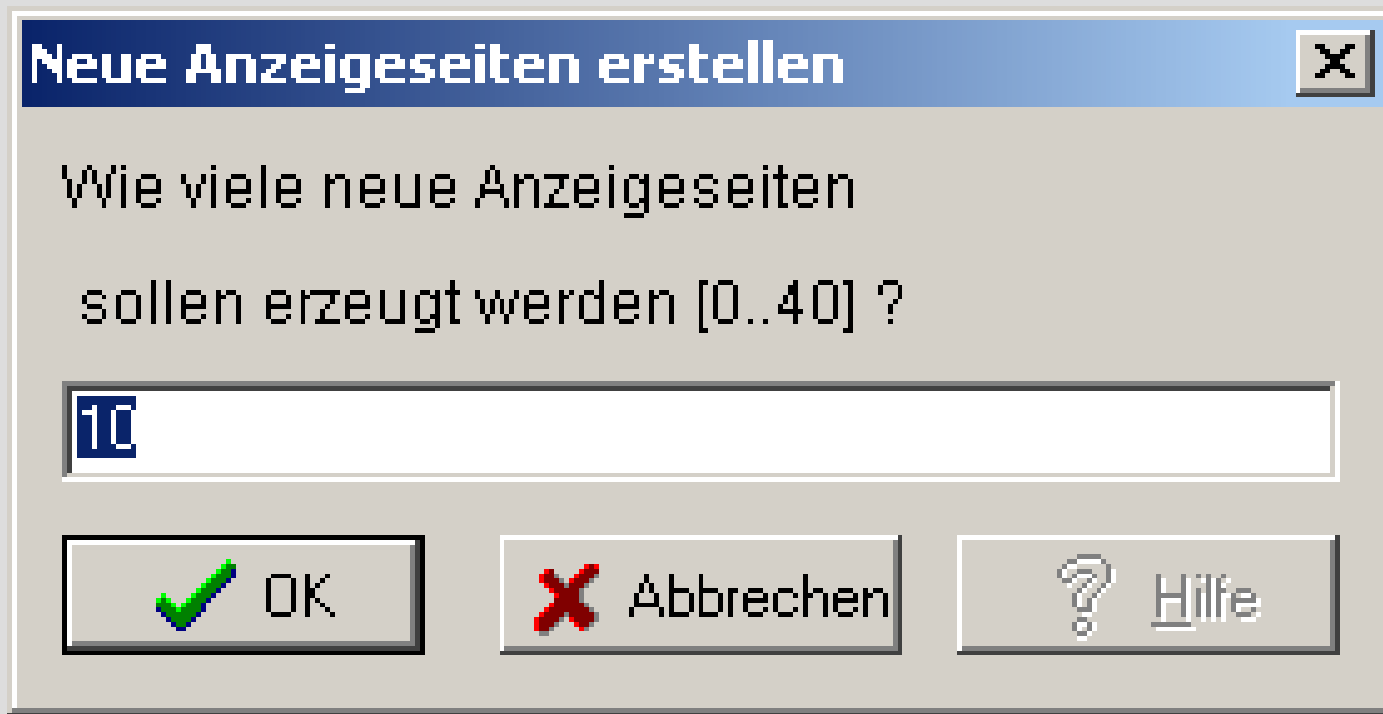


## Neues Anzeigeprogramm erstellen, Schritt 2 : Hard- und Softwareprofil des Gerätes auswählen



(Bei der Auswahl des [Hardwareprofils](#) wird u.A. das Farbmodell festgelegt:  
MKT-View II : 255; MKT-View III : 65535 Farben )

## Neues Anzeigeprogramm erstellen, Schritt 3 : Anzahl der zu erstellenden Seiten festlegen

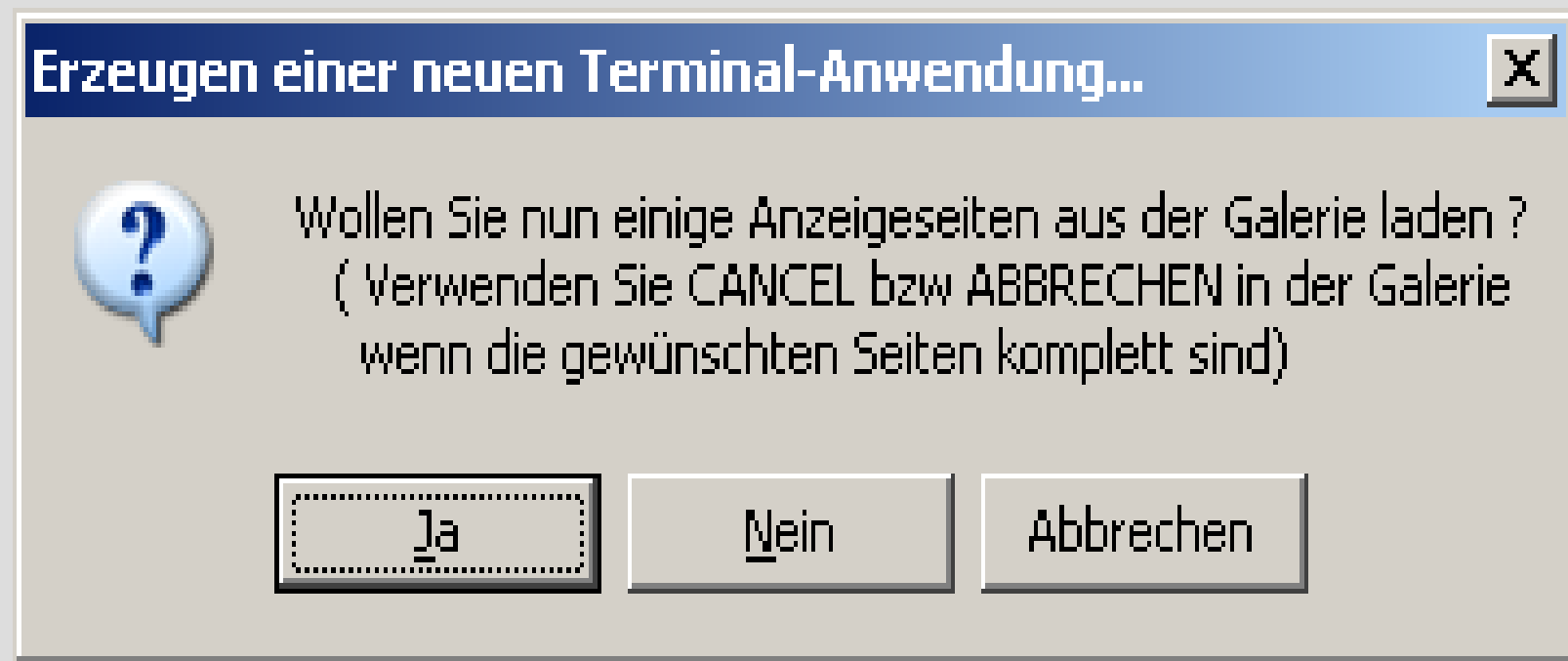


Neue Anzeigeseiten erstellen

Wie viele neue Anzeigeseiten  
sollen erzeugt werden [0..40] ?

(die Anzahl von Anzeigeseiten kann später noch geändert werden.  
Neuere Geräte wie MKT-View II / III unterstützen bis zu 255 Seiten)

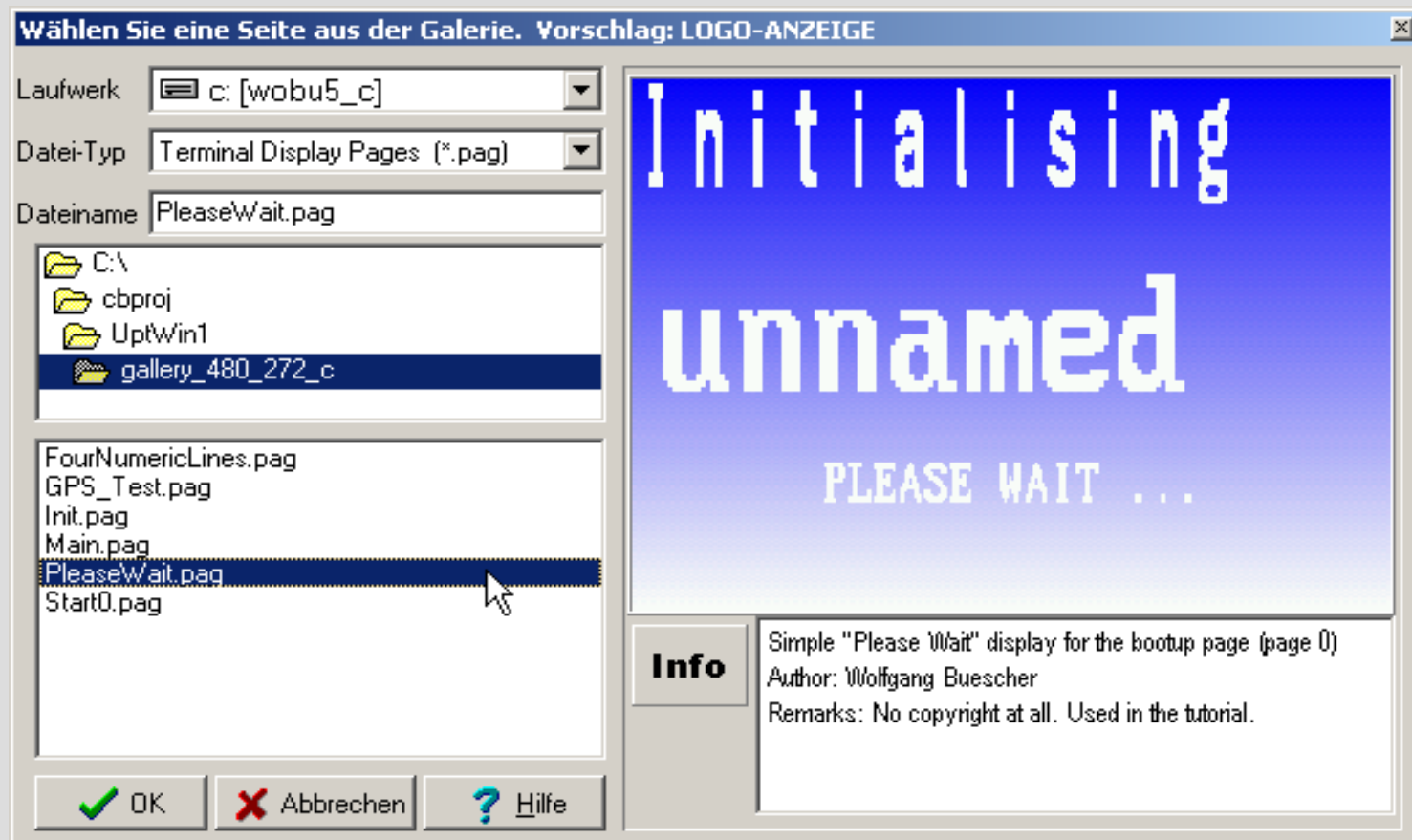
## Neues Anzeigeprogramm erstellen, Schritt 4 : Anzeigeseiten aus der Galerie (oder aus anderen Projekten) übernehmen



(auch dies ist kein 'Muss'

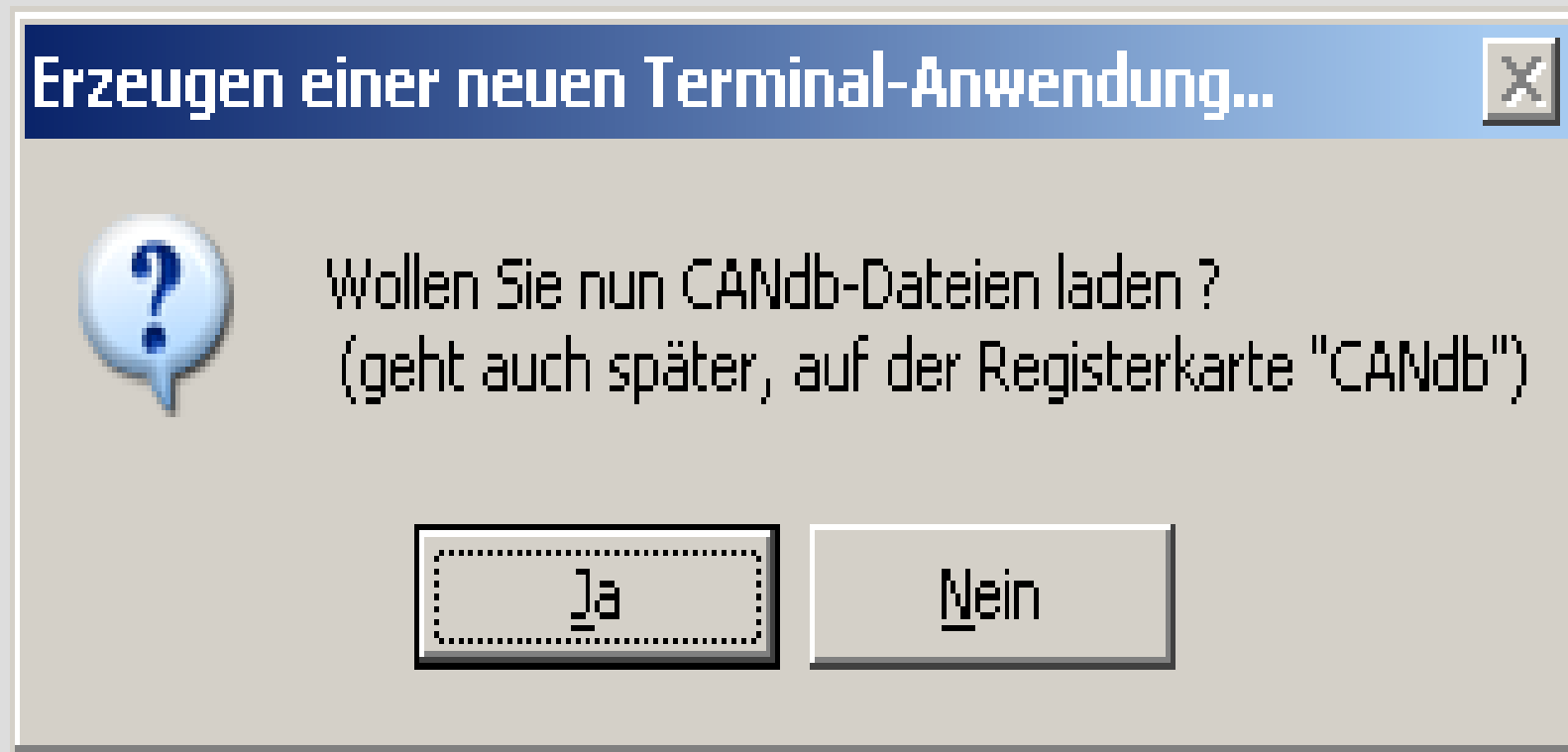
- die Seiten können auch neu mit dem Editor erstellt werden)

## Neues Anzeigeprogramm erstellen, Schritt 5 : Startseite, Menüseite, und andere Anzeigeseiten einfügen



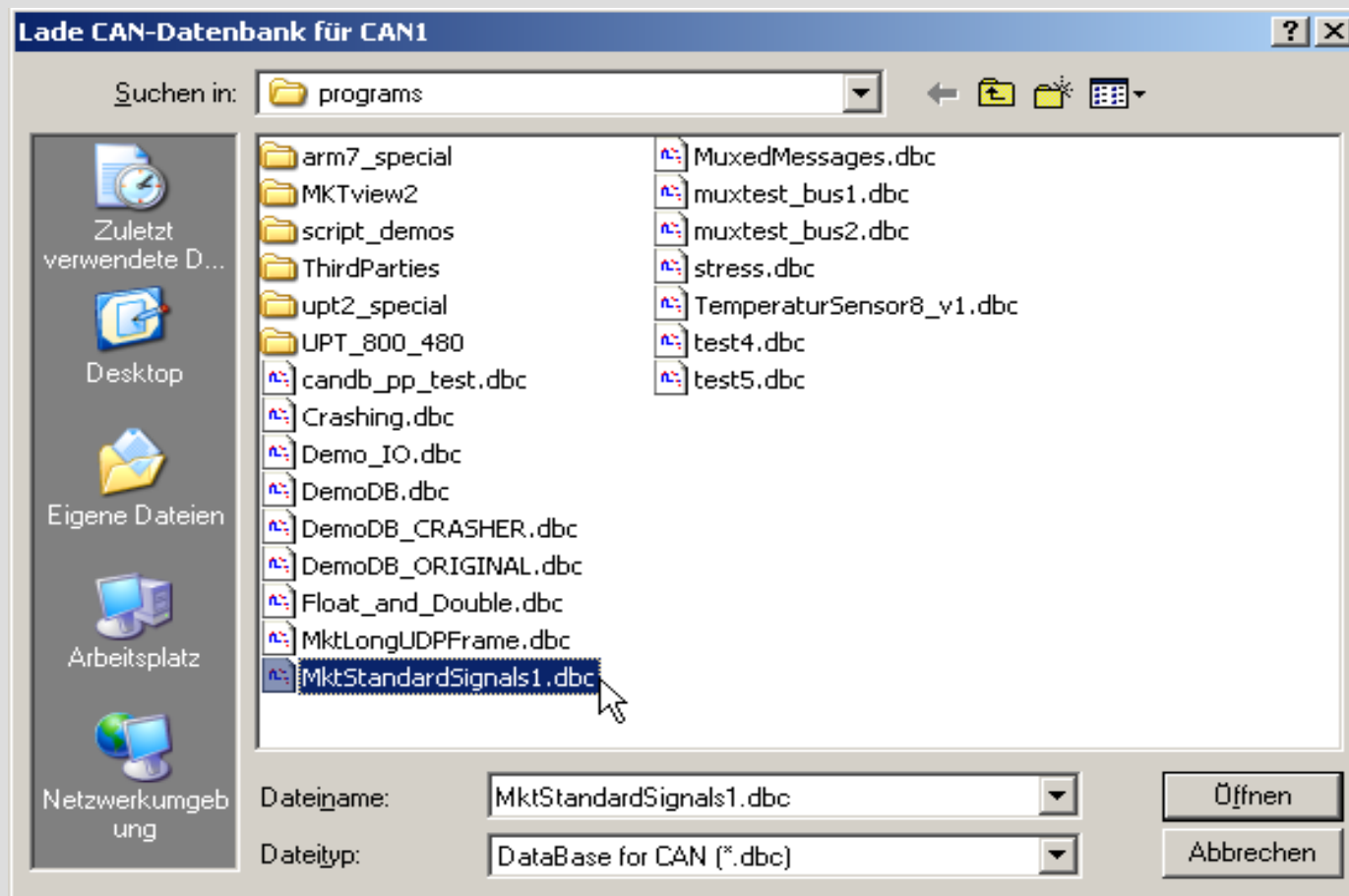
Seite Null wird beim Einschalten angezeigt, noch bevor die CAN-Protokolle laufen. Sie dient z.B. als 'Logo-Anzeige'. Tipps zur [Seitenumschaltung](#) folgen später ...

Neues Anzeigeprogramm erstellen, Schritt 6a :  
CAN-Datenbasen importieren ("CANdb", \*.dbc; optional)



(auch dies ist kein 'Muss', denn  
CANdb-Dateien können auch später noch hinzugefügt werden)

## Neues Anzeigeprogramm erstellen, Schritt 6b : CAN-Datenbasen importieren ("CANdb", \*.dbc; optional)



( Für diese Präsentation wird eine Beschreibung von MKT's "Standard-Signalen" geladen, denn diese können später -als Ersatz für ein Steuergerät- mit dem vom Autor entwickelten [CAN-Tester](#) gesendet werden ).

## Neues Anzeigeprogramm erstellen, Schritt 7 (optional) : Befehle zur Seitenumschaltung aus **Katalog** auswählen

### Katalog mit "globalen Event-Definitionen"

Wählen Sie Funktionen die in den GLOBALEN EVENTS der Applikation codiert werden sollen

- Umschalten zur nächsten / vorhergehenden Anzeigeseite per Drehknopf (relativ langsam)
- Umschalten zur nächsten / vorhergehenden Anzeigeseite per Taste F2 und F3
- Umschalten auf Seite Nr. "1" mit Taste F1 (Seite "0" sollte für Einschalt-Logo reserviert sein)
- Umschalten auf Seite namens "CanError" wenn der CAN-Controller "BUS-OFF" geht
- Nummer der aktuellen Seite in nichtflüchtiger Variablen " nv[0] " ablegen

Sie können den erzeugten Code später modifizieren, siehe auch : [Event-Definitions-Sprache](#).

Wenn ein bestimmtes EVENT schon abgefragt wird, wird die alte Definition überschrieben !

Klicken Sie auf ABBRECHEN wenn Sie Ihrer Anwendung keine dieser Funktionen hinzufügen wollen.

 OK

 Abbrechen

 Hilfe

# Anzeigeprogramm bearbeiten: Zuordnung von CAN-Signalen und Anzeige-Elementen

(ursprünglicher Ablauf, bei alten Programmierertools war dies die einzige Methode)

- 1.) CAN-Datenbank (\*.dbc) importieren (Registerkarte 'CANdb')
- 2.) Die für die Anzeige wichtigen Signale auswählen, und daraus Variablen (für die Anzeige) erzeugen

The screenshot shows the 'Programmierool für "CANdb"-Terminals' interface. The 'Fertig, in Variablen umsetzen' button is circled in red. Below it, a table lists CAN signals with columns: VarName, Bus, NodeName, MsgName, SigName, SigType, and Un. A red arrow points to the 'FourSines1' row, which is also circled in red. A tooltip is visible over the 'FourSines1' row, containing the text: 'Doppelklick hier ...', 'Verbinde "FourSines1" mit einer NEUEN Variablen, für Empfang', 'Verbinde "FourSines1" mit einer NEUEN Variablen, zum SENDEN', and 'Verbinde Signal mit einer bereits existierenden Variablen'.

VarName	Bus	NodeName	MsgName	SigName	SigType	Un
FourSines4	CAN1	MKT_WoBu	FourSines	FourSines4	SIGNED	mA
FourSines3	CAN1	MKT_WoBu	FourSines	FourSines3	SIGNED	mA
FourSines2	CAN1	MKT_WoBu	FourSines	FourSines2	SIGNED	V
FourSines1	CAN1	MKT_WoBu	FourSines	FourSines1	SIGNED	V

- 3.) Die Variablen wie beschrieben zur Anzeige bringen



# Anzeigeprogramm bearbeiten: Zuordnung von CAN-Signalen und Anzeige-Elementen

( alternativer Ablauf, nur bei neueren Programmierertools möglich)

- 1.) Anzeigeseiten neu erstellen oder wiederverwerten
- 2.) Per Dialog 'Prüfen und Ersetzen der Variablen..' zuordnen:

The screenshot shows a software interface titled "Programmiertool für 'CANdb'-Terminals". The menu is open, showing options like "Neues Programm, 'einfach'", "Neues Programm, 'universell'", and "Vordefinierte globale Events (Katalog)". The option "Prüfen und Ersetzen der Variablen einer Seite" is highlighted in blue.

Nr	...	...	...	...	...	Var/Form	Zugriff	Zeit
0						X	0	0
1	40	128			\vvvvvv=****.* \uu	X	0	0
2	40	160			\vvvvvv=****.* \uu	X	0	0
3	40	192			\vvvvvv=****.* \uu	X	0	0

# Anzeigeprogramm bearbeiten: Zuordnung von CAN-Signalen und Anzeige-Elementen

( alternativer Ablauf, nur bei neueren Programmierertools möglich)

3.) Auf Register 'Signale' die anzuzeigenden CAN-Signale wählen:

VARIABLEN oder SIGNALE für Seite 0 ("page0") auswählen

Verbinde Anzeigewert mit ... Variablen Signale Script

Line	Display Text	Type	Connection (var,sig,expr)
0	\vvvvvv=*****.*	\Var	X
1	\vvvvvv=*****.*	\Var	X
2	\vvvvvv=*****.*	\Var	X
3	\vvvvvv=*****.*	\Var	X

CAN-Knoten

- Master
- MKT\_WoBu
- Terminal

Messages des Knotens

- FourSines
- SignTestMsg
- ThreeSines

Signale der Message

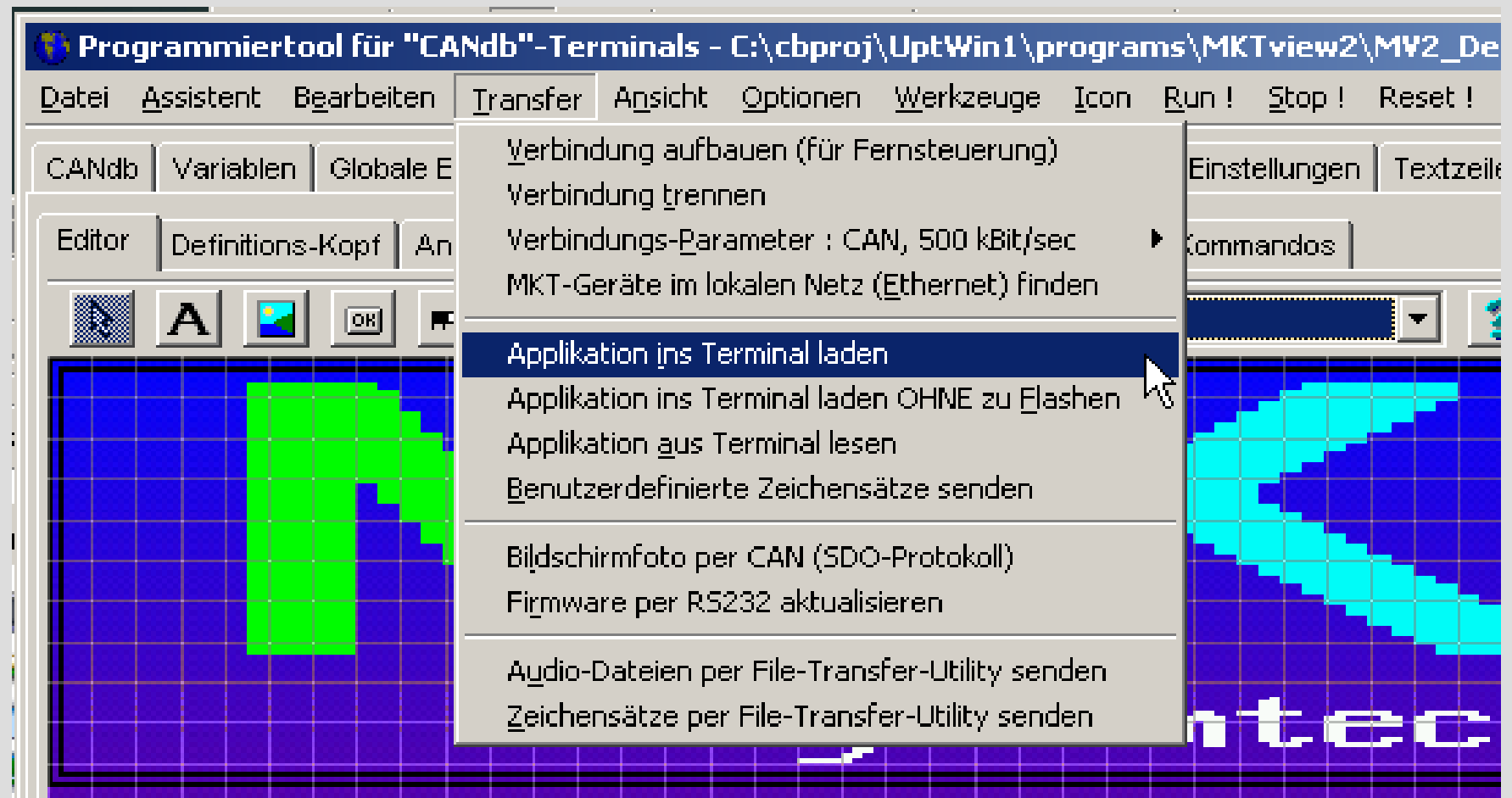
- FourSines4
- FourSines3
- FourSines2
- FourSines1

autom. in nächste Zeile  
 Namen im Anz.-Text ersetzen

OK Rückgg. Abbrechen Hilfe

# Übertragen des Programms vom PC in das Zielsystem Methode 1 (für *alle* Geräte) : per CAN-Bus

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



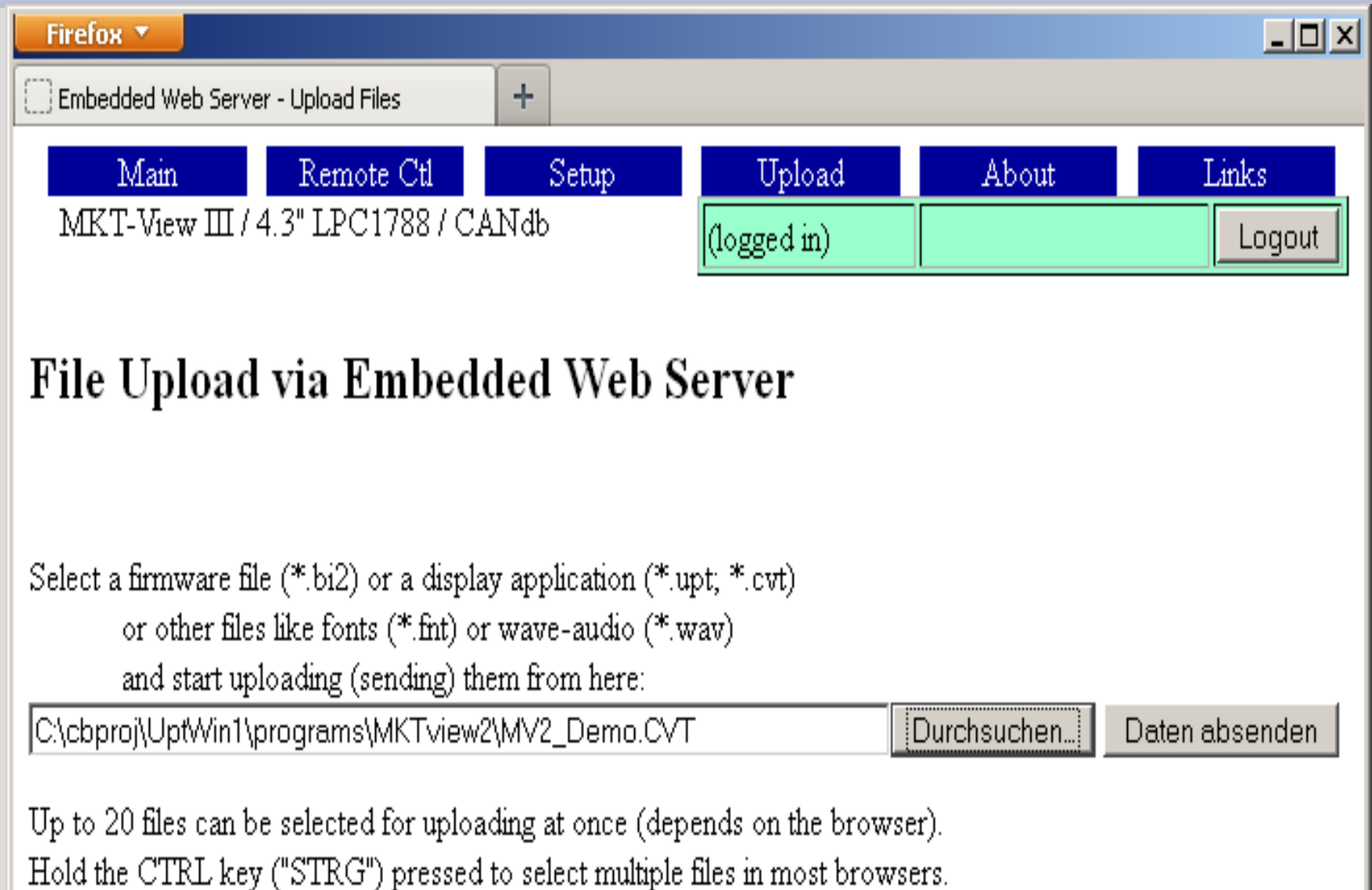
# Übertragen des Programms vom PC in das Zielsystem

## Methode 2 (für *manche* Geräte) : per Speicherkarte

- 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 in das Zielsystem Methode 3 (für *moderne* Geräte) : per [Browser / HTTP](#)



Firefox

Embedded Web Server - Upload Files

Main Remote Ctl Setup Upload About Links

MKT-View III / 4.3" LPC1788 / CANdb

(logged in) Logout

## File Upload via Embedded Web Server

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:

C:\cbproj\Upt\Win1\programs\MKTview2\MV2\_Demo.CVT

Durchsuchen... Daten absenden

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.

## Test des Anzeigeprogramms am CAN-Bus

- Laden des Test/Demoprogramms "MV2\_DEMO.CVT"
- Als Ersatz für ein Steuergerät: [CAN-Tester](#) mit Script 'MktStandardSignals1.cts' ( Beispiel : nächste Seite )
- Alternative: Programmierwerkzeug mit [CAN-Interface](#) (z.B. von Kvaser) im Simulatorbetrieb als Ersatz für ein 'echtes' Anzeigegerät, d.h. Betrieb des PCs am 'echten' CAN-Bus)  
oder
- [Abspielen einer Aufnahme vom CAN-Bus](#) im Vector-Format (im Hauptmenü: Werkzeuge.. CAN-Logfile abspielen)

# Testen der Display-Applikation : Ersatz eines Steuergerätes durch CAN-Tester (1)

Experimental CAN / CANopen Tester for Windows V1.52

Datei Einstellungen Ansicht Suche Run Meldungen Pause! Trigger Tools Hilfe

Rx+Tx N=0 sorted 59:4 Stop Lösch Tx 1:1 Stop Lösch

```
+000104: 333      67 24 D7 70 FE 64      ;[T] N=83 P=187
+000020: 334      01 00 00 00 01 00 00 00 ;[T] N=84 P=185
+000020: 00000444 FC 7F D3 02 AD 99 9C FB ;[T] N=83 P=187
+000041: 12345678 1A 01 3D 01 6B 01 A7 01 ;[T] N=83 P=185
```

**MKT-Standard-Testsignale #1**

### Sende Standard-Test-Signale

Signaldefinitionen: c:\cbproj\CanTest\programs\MktStandardSignals1.dbc  
CAN-IDs / Signale:  
0x333.s: ThreeSines1, ThreeSines2, ThreeSines3  
0x444.x: FourSines1, . . . . . , FourSines4

Die im Moment GESENDETEN Signale haben einen sinusförmigen Verlauf mit unterschiedlichen Frequenzen und z.T. unterschiedlichen Amplituden .

Mit einem zweiten PC kann der Verlauf dieser Signale angezeigt werden. Dazu auf dem zweiten PC auf "Empfang" klicken, bevor die CAN-Verbindung hergestellt wird (sonst gibt's üble CAN-ID-Kollisionen / Protokollfehler).

Hilfe Beenden Empfang

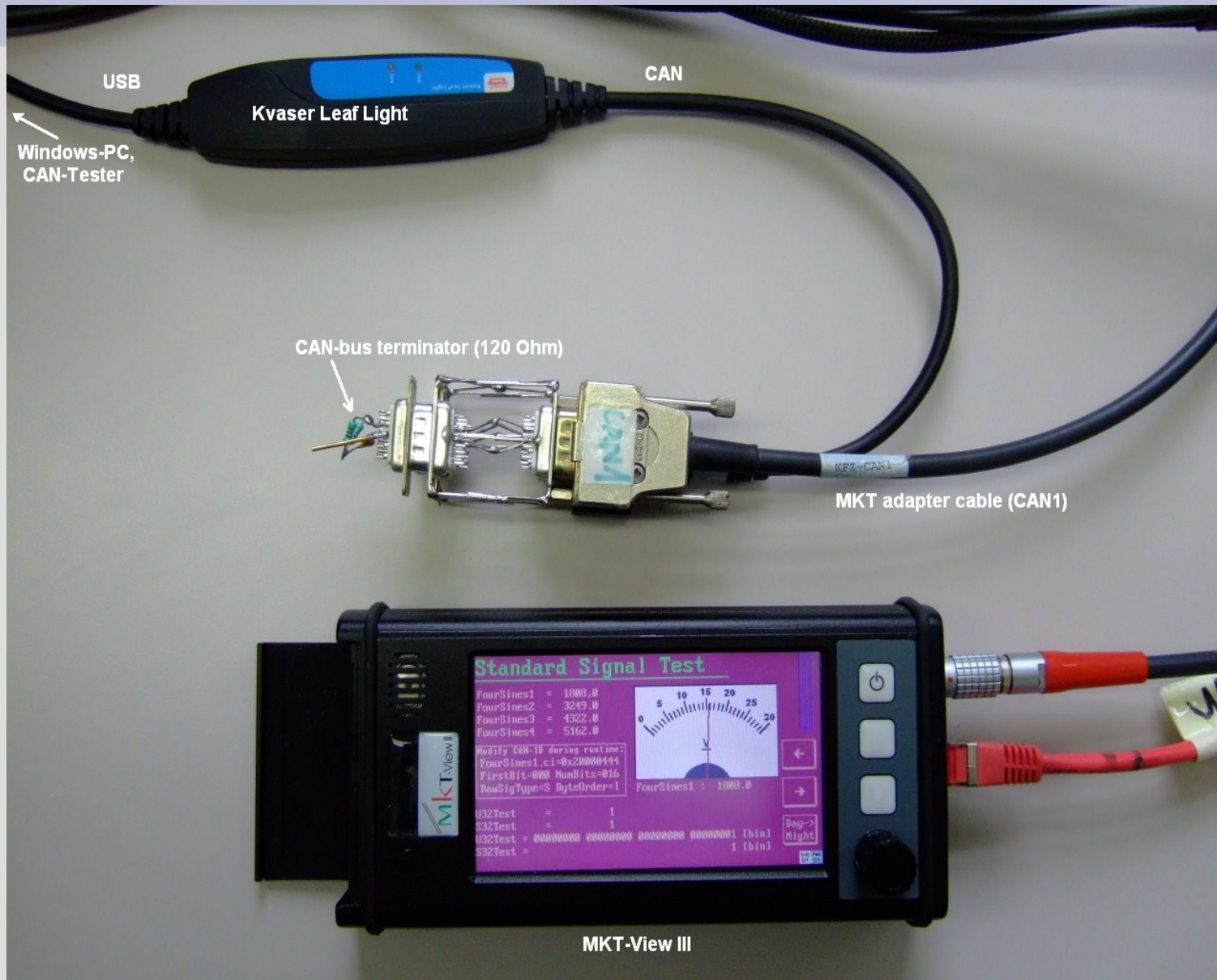
Commands [running] Run: line=49 1:1 Stop

```
; MktStandardSignals1.TXT = Testprogramm zum Senden der "MKT-Standard-Testsignale"
;   Diente urspruenglich fuer "UPT"-Entwicklung mit CAN-Logger" ,
;   später aber auch für andere Standard-Testfälle .
;   Diese MESSAGES (mit den entsprechenden Signalen) flitzen oft mit 500 kBit/sec
;   über den MKT-internen, "laborübergreifenden CAN-Test-BUS" .
;
; Autor: Wolfgang Büscher
; Stand: 2000-04-06 (ISO 8601 YYYY-MM-DD)
```

Meldungen 1:4 Stop Lösch

Loaded CANtest command file from "C:\cbproj\CanTest\programs\MktStandardSignals1.cts" .  
Loaded CANopen OD for SDO server from "C:\cbproj\CanTest\Objects\MP85A.eds" .  
CAN: UCAN\_Init: Loaded Kvaser CANLIB32 V4.2 .

# Testen der Display-Applikation : Ersatz eines Steuergerätes durch CAN-Tester (2)





# Testen der Display-Applikation : Ersatz eines Steuergerätes durch CAN-Log-Replay

The screenshot shows a software interface for testing a display application. The main window is titled "Programmierwerkzeug für 'CANdb'-Terminals" and contains a table of signal definitions. A secondary window, "UPT-Simulator: 'StdSigs'", displays a graphical user interface for a signal test. A third window, "Logfile PLAYING", shows the playback progress of a log file.

Nr	Name	X	Y	W	H	Text/Anzg.-Kommando	Var/Form	Zu
0		0	0			Standard Signal Test		0
1		3	41			\vvvvvvvvvv = *****	FourSines:0	
2		3	57			\vvvvvvvvvv = *****	FourSines:0	
3		3	73			\vvvvvvvvvv = *****	FourSines:0	
4		3	89			\vvvvvvvvvv = *****	FourSines:0	
5		3	115			Modify CAN-ID during		0
6		7	128			\vvvvvvvvvvvv=0x****	FourSines:1	
7		7	144			FirstBit=***		
8		111	144			NumBits=***		
9		7	160			DevSigType=\s(v)1"		
10						er=\s(v)1"M		
11						vvvv = ****		
12						vvvv = ****		
13						= *****		
14						= *****		
15						og m"		

**UPT-Simulator: "StdSigs" Display:**

**Standard Signal Test**

FourSines1 = -2209.0  
 FourSines2 = -3609.0  
 FourSines3 = -4639.0  
 FourSines4 = -5436.0

Modify CAN-ID during runtime:  
 FourSines1.ci=0x20000444  
 FirstBit=000 NumBits=016  
 RawSigType=S ByteOrder=1

FourSines1 : -2209.0

U32Test = 1  
 S32Test = 1  
 U32Test = 00000000 00000000 00000000 00000001 [bin]  
 S32Test = 1 [bin]

Day-> Night

Logfile PLAYING: MktStandardSignals1.asc, 89 kB, 5.3 sec, Loop-Modus checked.

Simulator läuft | Switched from page 5 to page 4 ("StdSigs"), reason: script.goto\_page

# Programmierertools für UPT und MKT-View

## Ende des ersten Teils

Weiterführende Informationen (Web-Links) :

- [Online-Hilfesystem zum Programmiertool 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](http://www.mkt-sys.de/MKT-CD/upt/help/progt_49.htm)
- ['Druckbare' Handbücher von MKT im PDF-Format](http://www.mkt-sys.de/MKT-CD/upt/help/progt_49.htm#documents_in_pdf_format)  
[www.mkt-sys.de/MKT-CD/upt/help/progt\\_49.htm#documents\\_in\\_pdf\\_format](http://www.mkt-sys.de/MKT-CD/upt/help/progt_49.htm#documents_in_pdf_format)
- [Inhaltsverzeichnis der 'MKT-CD' mit Download-Links](http://www.mkt-sys.de/MKT-CD/index.htm)  
(Programmierertools, CAN-Tester, CAN-Logger, Utilities)  
[www.mkt-sys.de/MKT-CD/index.htm](http://www.mkt-sys.de/MKT-CD/index.htm)
- [Homepage von MKT Systemtechnik](http://www.mkt-sys.de)  
[www.mkt-sys.de](http://www.mkt-sys.de)

# Programmierertools für UPT und MKT-View

## Teil 2 : Event-Steuerung

Prinzip : "Wenn <Ereignis> dann <Reaktion>" . Beispiel:  
Umschaltung von Farben abhängig von Messwerten

The screenshot displays the 'Programmiertool für CANdb' interface. The main window shows a table of event definitions with columns for 'Nr', 'Ereignis', and 'Reaktion'. The event 'FourSines1 > 16000' is highlighted with a red circle, and its reaction 'disp.Bar1.fc = red' is also circled. A red arrow points from the event to the reaction. Below the table, a simulation window titled 'UPT-Simulator: "EventTest"' shows the current state of four variables: 'FourSines1 = + 25707', 'FourSines2 = - 28733', 'FourSines3 = + 11497', and 'FourSines4 = + 9806'. The value '25707' is circled in red. Below the text, four bar graphs represent the variables, with the first bar (FourSines1) filled with red. The simulation window also includes a 'Day-Night' indicator and a 'Reaktion zum TESTEN jetzt ausführen' button. The bottom status bar shows 'Simulator Läuft' and 'Switched from page 7 to page 8 ("EventTest")'.

Nr	Ereignis	Reaktion
99	FourSines1 < -16000	disp.Bar1.fc = blue : REM blue = too cold (or wh
98	FourSines1 > 16000	disp.Bar1.fc = red : REM red = too hot (or wh
97	(FourSines1 > -16000) && (FourSines1 < 16000)	disp.Bar1.fc = -1 : REM default colour from
96	FourSines2 < -16000	
95	FourSines2 > 16000	
94	(FourSines2 > -16000) && (F	
93	FourSines3 < -16000	
92	FourSines3 > 16000	
91	(FourSines3 > -16000) && (F	
90	FourSines4 < -16000	
89	FourSines4 > 16000	
88	(FourSines4 > -16000) && (F	
87		

UPT-Simulator: "EventTest"  
Event Test (1)  
FourSines1 = + 25707  
FourSines2 = - 28733  
FourSines3 = + 11497  
FourSines4 = + 9806

Day-Night  
Reaktion zum TESTEN jetzt ausführen

# Programmierertools für UPT und MKT-View

## Möglichkeiten der Event-Steuerung

### Beispiel für 'Ereignisse'

- Vergleich von Variablen (Werten) wie im vorherigen Bild
- Tastaturabfrage (Taste gedrückt, Taste losgelassen, ..)
- Ablauf eines Timers
- ....

### Beispiele für 'Reaktionen'

- Ändern der Farbe eines Elements wie im vorherigen Bild
- Umschalten der aktuellen Anzeigeseite ("goto")
- Starten oder Stoppen eines Timers
- Zuweisen eines neuen Wertes an eine Variable
- ....

Für komplexere Abfragen und Verknüpfungen wird empfohlen, statt der 'Display-Events' die wesentlich vielseitigere Script-Sprache zu verwenden !

# Programmierertools für UPT und MKT-View

## Alternative zur den 'Display-Events': Script-Sprache

### Vorteil :

- 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