



Handbuch und Bedienungsanleitung

Version 1.6

Programmiertool für UPT und MKT-View (II / III / IV)

zur Verwendung in CANopen - Netzwerken
und für Terminals mit Parametrierung per 'CANdb'



Dokument-Nr: 85110 (deutsch)

Original: C:\CBProj\UptWin\DOKU\art85110_Handbuch_UPT_Tool.odt

Autor: Wolfgang Büscher (Software-Entwicklung)

Inhaltsverzeichnis

1. Versionen des Handbuchs.....	4
2. Vorwort.....	5
3. Zweck und Eigenschaften des Terminals.....	6
4. Systemvoraussetzungen.....	7
5. Installation.....	8
6. Weiterführende Informationen.....	8
7. Beschreibung des Programmiertools.....	10
7.1 Das Hauptfenster.....	10
7.2 Das Hauptmenü.....	12
7.3 Statuszeile.....	12
7.4 Das Hilfe-System des Programmiertools.....	13
7.5 Import von CAN-Datenbanken aus *.dbc-Dateien.....	15
7.6 Das Fenster des LCD-Simulators.....	17
7.7 Kommando-Interpreter.....	18
7.8 Kommunikationskanäle.....	19
7.8.1 SDO-Kanäle.....	20
7.8.2 PDO-Kanäle.....	21
7.8.3 Kommunikationskanal für CANdb-Signale.....	22
7.9 Variablen des Anwenderprogramms.....	23
7.10 Definition von Anzeigeseiten.....	24
7.10.1 Der "einfache" Editor für Anzeigeseiten.....	25
7.10.2 Seiten-Definitions-Kopf.....	29
7.10.3 Tabellarische Definition von Anzeige-Zeilen.....	30
7.10.4 Farben.....	31
7.10.5 Der Format-String in Anzeigezeilen.....	32
7.10.6 Verwendung von Displayzeilen als Menüs.....	34
7.10.7 Überlappende Grafiken.....	35
7.11 Spezielle Anzeige-Kommandos.....	35
7.12 Definition von Ereignissen (Übersicht).....	37
7.13 Die Eingabe von Event-Definitionen mit dem Programmiertool.....	38
7.13.1 Befehle zur Definition von Ereignissen.....	39
7.14 Globale Ereignisse (Global Events).....	40
7.15 Anzeigeseiten-Übersicht.....	41
7.16 Import und Verwendung von Bitmap-Grafiken ('Icons').....	42
7.16.1 Reservierung von Speicher für Icons und Sonderfunktionen.....	43
7.17 Grundeinstellungen (General Settings).....	44
7.17.1 Allgemeine UPT-Optionen.....	45
7.18 Das Text-Array.....	46

7.19 Die Fehler-Anzeige des Programmiertools.....	46
7.20 Bildschirmfotos vom Terminal per CAN-Bus.....	47
8. Der Display-Interpreter im UPT (Sprachbeschreibung).....	48
8.1 Allgemeine Syntax.....	48
8.1.1 Numerische Ausdrücke.....	48
8.1.1.1 Zahlen.....	48
8.1.1.2 Numerische Verknüpfungs-Operatoren.....	49
8.1.1.3 Numerische Variablen.....	51
8.1.1.4 Numerische Funktionen.....	52
8.1.1.5 PDO-Funktionen des UPT-Interpreters.....	53
8.1.1.6 SYNC-Funktionen des UPT-Interpreters.....	55
8.2 String-Funktionen.....	56
8.3 Interpreter-Kommandos.....	57
8.4 Steuerung des Programablaufs per Display-Interpreter.....	57
8.4.1 Goto, Call und Return.....	57
8.5 Das Zuweisungs-Kommando (@).....	58
8.6 Weitere 'spezielle' Interpreter-Kommandos.....	59
8.6.1 Timer-Kommandos und -Funktionen.....	59
9. Übertragung zwischen Programmiertool und Gerät.....	61

1. Versionen des Handbuchs

Versionsnummer	Datum YYYY-MM-DD	Autor	Hinweise, Änderungen
V0.1	1999-11-03	W.Büscher	Erste Handbuch-Version (englisch)
V0.2	1999-12-13	W.Büscher	Erster „externer“ Einsatz. Programmiertool: Version 1.0 „beta“.
V0.3	2000-03-22	W.Büscher	Neue Interpreter-Funktionen.
V0.4	2000-05-31	W.Büscher	Weitere neue Interpreter-Funktionen. Nicht kompatibel mit „alter“ Software. Programmiertool: Version 1.1 „beta“ (oder neuer) erforderlich. Erste deutsche Übersetzung des Handbuchs.
V0.5	2000-07-07	W.Büscher	„General UPT Options“ implementiert und hier beschrieben.
V0.6	2000-07-19	W.Büscher	Implementierung von „Displayzeilen als Menüs“
V0.7	2001-02-20	W.Büscher	Tastenkombination zum Aufruf des UPT-System-Menues variabel; hier beschrieben.
V0.8	2001-03-29	W.Büscher	(in Vorbereitung)
V0.9	2001-09-05	W.Büscher	per Interpreter änderbare PDO-Komponenten
V1.0	2002-02-27	W.Büscher	Hinweise für Anwender des „CANdb-Anzeige-Gerätes“ aufgenommen
V1.1	2003-05-28	W.Büscher	Hinweise zu "DriverLinx" entfernt. Funktionierte nicht unter WinXP -> wird nicht mehr verwendet.
V1.2	2005-09-05	W.Büscher	Hinweise zum "MKT-View" <i>ohne</i> CANopen
	2007-04-23	W. Büscher	Umstieg von WORD auf OpenOffice, Dateiformat nun *.odt (OpenDocumentText), Export als "Tagged PDF" (mit Links)
V1.3	2008-09-04	W. Büscher	Ergänzungen für "MKT-View II", neuer graphischer Editor für Anzeigeseiten.
V 1.4	2011-06-08	W. Büscher	Hinweise zur Dateiübertragung ergänzt
V 1.5	2013-04-15	W. Büscher	Übernahme einiger Kapitel aus der Online-Hilfe des Programmiertools
V 1.6	2014-06-18	W. Büscher	Referenz 'CDP' (CAN Display Terminal) eingefügt
V 1.6 b	2015-04-07	W. Büscher	Es wird dringend empfohlen, statt dieses Dokumentes ausschliesslich die Dokumentation im HTML-Format zu verwenden !
V1.6 c	2016-08-02	W. Büscher	Links auf die MKT-Webseite angepasst
V1.6 d	2021-01-19	W. Büscher	Weitere Links zur Online-Hilfe eingebaut
V1.6 e	2021-05-04	W. Büscher	Die scheußlichen 'Kapitälchen' entfernt, und weiteren Ballast abgeworfen .

2. Vorwort

Dieses Handbuch beschreibt ein Windows-Programm mit dem verschiedene Bedienterminals aus MKT's „UPT“-Serie programmiert werden können (UPT= user programmable terminal).

Für eine Sondervariante des UPT-Programmiertools namens "Programmiertool für CANdb-Anzeige-Terminals" (MKT-View „Plus“, II / III / IV) existiert derzeit nur ein [Zusatz zu dieser Bedienungsanleitung](#) (Dokument Nr 85111). Ignorieren Sie bitte die Kapitel "CANopen, SDO, PDO" und alle Absätze mit dem Hinweis "nur für UPT" bzw "nur für CANopen" in diesem Dokument, falls Sie das spezielle Programmiertool mit CANdb-Importfunktion verwenden.

Diese Software wird mit allen begleitenden Materialien (incl. Bedienungsanleitung) im aktuellen Zustand ohne Anspruch auf irgendwelche Gewährleistung weitergegeben.

Darüberhinaus garantiert die Firma MKT Systemtechnik weder die Einsetzbarkeit dieser Software für einen bestimmten Einsatzzweck, die Fehlerfreiheit, Genauigkeit, Zuverlässigkeit, Aktualität; noch haftet sie für die Folgen des Einsatzes dieser Software oder der Bedienungsanleitung.

Verantwortlich für den Einsatz ist der Anwender, aber nicht MKT Systemtechnik oder deren Lieferanten, Partner oder Beschäftigte.

There are no other warranties, either expressed or implied, including but not limited to implied warranties of merchantability and fitness for a particular purpose, with respect to the software, the accompanying written materials, and any accompanying hardware. (...Es gilt die englischsprachige Original-Version...)

CANdb is registered trademark of Vector Informatik GmbH
NTCAN API is copyright (c) by ESD electronic system design GmbH
PCAN Dongle and the PCAN API is copyright (c) by PEAK-Service GmbH
Microsoft, Windows, Win98, WinNT, WinXP are registered trademarks of Microsoft Corporation

Produktnamen, die eingetragene Warenzeichen sind, werden in diesem Dokument nicht besonders gekennzeichnet. Daher bedeutet ein fehlendes (c), (r), oder (tm) - Zeichen nicht, daß ein Name kein eingetragenes Warenzeichen bzw nicht urheberrechtlich geschützt ist.

Hinweis: Im Gegensatz zur mittlerweile auch [ins Deutsche übersetzten Online-Hilfe \(HTML-Format\)](#) *ist dieses Dokument mittlerweile deutlich veraltet* ! Neue Funktionen werden zunächst nur in den HTML-Dateien beschrieben, und später -beizeiten- in *dieses* Dokument übernommen.

3. Zweck und Eigenschaften des Terminals

Der Haupteinsatzzweck des UPTs (UPT = User Programmable Terminal) ist die Anzeige von Parametern in einem CANopen-Netzwerk.

Die MKT-Views sind dagegen zum Einsatz im KFz-Bereich vorgesehen. Statt CANopen wird die Kommunikation hier durch Import einer oder mehrerer CANdb-Dateien festgelegt.

Die vom vorliegenden Programmierwerkzeug unterstützten Geräte (Terminals) sind:

- UPT515: Grafik-LCD mit 128*64 Pixel
- UPT167: 320 * 240 Pixel passiv STN LCD, monochrom, 16-Bit-CPU mit 40 MHz
- (UPT : "User Programmable Terminal", i.A. mit CANopen-Protokoll-Stack)
- MKT-View I : 320 * 240 Pixel passiv STN LCD, monochrom, 16-Bit-CPU mit 40 MHz (wird seit 2008 nicht mehr weiterentwickelt, Ersatz durch MKT-View II / III)
- MKT-View II : 480 * 272 Pixel TFT, 256 Farben, Touchscreen, ARM7-CPU mit 72 MHz
- MKT-View III : 480 * 272 Pixel TFT, 65535 Farben, Touchscreen, Cortex-CPU mit 96 MHz
- MKT-View IV : 800 * 480 Pixel TFT (7 Zoll), 65535 Farben, Touchscreen, Cortex-M4 mit 200 MHz
- MKT-View V (geplant) : Touchscreen mit mindestens 480 * 272 Pixel, Cortex-M7 mit mindestens 600 MHz, CAN-FD-fähig auf mindestens 2 Ports .

Unabhängig von CANopen (bei den UPTs) oder "CANdb" (bei den MKT-Views) werden bei allen Geräten die per CAN empfangenen Signale zunächst in internen Variablen (Anzeige-Variablen) im Gerät abgelegt. Für die Visualisierung greift das Gerät dann auf diese Variablen zu.

Das Anwenderprogramm (im UPT) kann diese Variablen (-Werte) auf verschiedene Arten auf dem Bildschirm darstellen:

- Einfache numerische Anzeige, optional mit Signal/Variablenname, Wert, physikalischer Einheit, und zusätzlichem Text für den Bediener
- Anzeige als Balkendiagramm (horizontal oder vertikal)
- Anzeige als Y(t)- oder X/X-Diagramm
- andere Sonderanzeigen, die mit Hilfe elementarer Grafikbefehle bzw (bei Geräten mit ARM- oder Cortex-CPU) mit Hilfe einer komfortablen Script-Sprache realisiert werden können.

Jede Bildschirmseite kann mehrere Werte anzeigen, und mehrere „Ereignis-Definitionen“ und „Reaktionsmethoden“ enthalten.

Eine Ereignis-Definition kann z.B. ein simples Tastatur-Ereignis sein

Eine Ereignis-Definition kann aber auch ein komplexer Ausdruck wie z.B. ein Vergleich zweier numerischer Werte/Variablen sein

Eine „Reaktionsmethode“ (event reaction) kann ein simpler Umschaltbefehl zu einer anderen Bildschirmseite sein, das Setzen einer Variablen, oder der Aufruf eines oder mehrerer Interpreterkommandos.

4. Systemvoraussetzungen

Das Anzeigeprogramm wird mit den zum Gerät passenden Programmiertool erstellt:

- 'UPT-Programmiertool I' : Wird nur noch für das 'UPT-515' mit einem mittlerweile veralteten CANopen-Standard (DS301 V3) verwendet
- 'UPT-Programmiertool II' : Für alle Geräte mit CANopen-Protokoll. Dazu zählen (fast) alle UPTs, aber auch MKT-Views mit Sonder-Firmware (für CANopen statt 'CANdb')
- Programmiertool für "CANdb"-Terminals: Für MKT-View II / III / IV; mit Einschränkungen auch für das veraltete 'MKT-View I' (wobei beachtet werden muss, dass die aktuellen Geräte -und damit auch das Programmiertool- wesentlich mehr Funktionen bieten als bei den alten Geräten mit 8- oder 16-Bit-CPU implementiert werden konnte)

Die Programmiertools für CANopen- bzw CANdb-Anzeigeterminals sind unter Windows 98, ME, 2000, Windows XP oder Windows 7 / 8 / 10 lauffähig. Windows Vista wird offiziell nicht unterstützt, und möge wie die älteren Windows-Versionen in Frieden ruhen.

Beachten Sie bitte die [Hinweise zur Installation im Online-Handbuch](#).

Zur Programmierung der MKT-Terminals benötigen Sie:

- das zum Gerät passende Programmiertool (siehe Oben)
- ein vom Programm unterstütztes CAN-Interface am PC mit der passenden, vom Hersteller bereitgestellten CAN-Treiber-Software. Geeignet u.A.:
 - alle Interfaces von Kvaser AB, die von deren CAN-API (CANLIB32) unterstützt werden
 - PEAK-CAN-Dongle, PCAN-USB der Firma Peak, und alle von Peak's CAN-API2 unterstützten Interfaces (Achtung, die CAN-API2 muss meistens extra bestellt werden)
- Ein zum Terminal passendes Adapterkabel (mit DC-Versorgung und CAN-Bus).
Siehe 'Zubehör-Katalog' im Download-Bereich auf der Webseite von MKT Systemtechnik.

Als Alternative (statt der Übertragung per CAN) ist bei einigen Terminals auch die Übertragung per RS-232, bei anderen Terminals auch per Speicherkarte (CF bzw SD) möglich.

Neuere Geräte wie MKT-View II / MKT-View III werden am besten per Ethernet (LAN) angesprochen, denn im Gegensatz zu USB hat sich diese Schnittstelle als zuverlässig, schnell, und komfortabel herausgestellt (auch zum Hochladen von Dateien, zum Debuggen per Web-Browser bei der Entwicklung von Scripten, und für die Fernwartung).

5. Installation

Das Programmiertool kann von der Webseite www.mkt-sys.de heruntergeladen werden. Details zur Installation finden Sie ausschließlich [im Online-Handbuch](#).

6. Weiterführende Informationen

... finden Sie z.B. in den folgenden Dokumenten, die z.T. im Installationsarchiv des Programmiertools enthalten sind, oder von der [MKT-Webseite](#) heruntergeladen werden können.

- Dokument Nr. [85110](#): Handbuch zum UPT-Programmiertool
- Dokument Nr. [85111](#): Zusatzinformationen für Terminals mit Parametrierung per "CANdb" (z.B. für MKT-View, MKT-View +, MKT-View II, und Andere(!) .
- Dokument Nr. [85113](#): Beschreibung des Konfigurationstools für den CAN-Logger (!)
- Dokument Nr. [85115](#): Beschreibung und Bedienung des Systemmenüs (im Terminal)
- Dokument Nr. [85116](#): Spezifikation des Download-Protokolls (nur für Programmierer)
- Dokument Nr. [85118](#): Beschreibung des optionalen CAN-Snoopers (CAN-Monitor)
- Dokument Nr. 85130: Inbetriebnahme und erste Schritte mit dem MKT-View II (i.V.)

Da die Dokumentationen bewusst allgemeingültig gehalten werden, finden Sie dort z.T. Informationen, die für ein bestimmtes Gerät nicht nutzbar sind. Z.B. werden Ihnen die in diesem Dokument enthaltenen Tipps zur Verwendung von Farben beim MKT-View III wenig helfen, wenn Sie (noch) ein altes MKT-View (1) / MKT-View "Plus" mit passivem, monochromen LCD verwenden. Der Hinweis 'Nur für aktuelle Geräte mit 32-Bit-CPU !' bedeutet u.A., dass die beschriebene Funktion für ältere Geräte wie "UPT-515", MKT-View und MKT-View "Plus" nicht nutzbar sind. Eine aktuelle, vom Entwickler verfasste Übersicht der Geräte und deren Fähigkeiten finden Sie online in der 'Feature-Matrix' der verschiedenen Geräte von MKT:

<http://www.mkt-sys.de/MKT-CD/upt/help/featmatr.htm>

Empfehlung des Entwicklers für Anwender von MKT-View I / II / III :

Eine kurze Einleitung zum Thema 'CANdb', mit dem prinzipiellen Ablauf finden Sie im Online-Hilfesystem, welches nach der Installation des Programmiertools z.B. im Verzeichnis

c:\MKT\CANdbTerminalProgTool\Help\candb_49.htm

vorliegen sollte (bzw, je nach Zielverzeichnis bei der Installation, im entsprechenden Unterverzeichnis).

Das komplette Hilfesystem ist auch (ohne Installation) online verfügbar, z.B. :

http://www.mkt-sys.de/MKT-CD/upt/help/candb_49.htm

Die vom Entwickler neu implementierten Funktionen werden grundsätzlich zunächst in der Online-Hilfedokumentiert, und danach -gelegentlich- aus den HTML-Dateien in die 'druckbaren' Handbücher (z.B. art85110_Handbuch_UPT_Tool.odt / .pdf) übernommen.

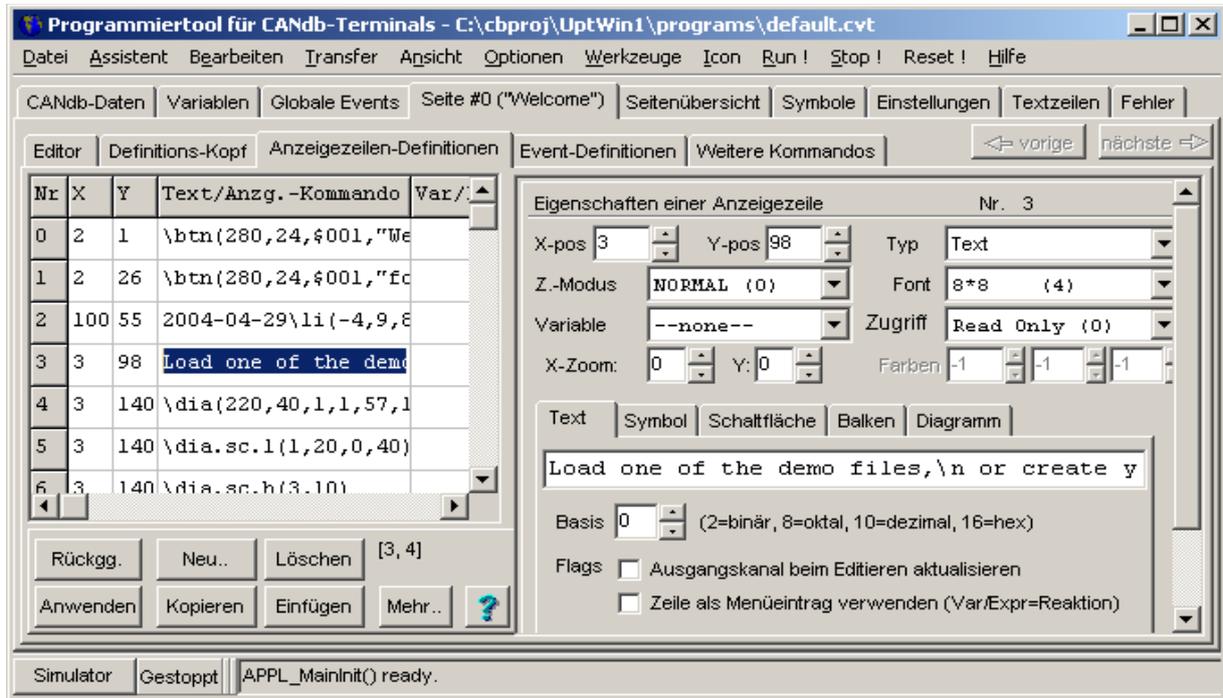
Die Online-Hilfdateien sind daher im Gegensatz zu den 'druckbaren' Handbüchern immer auf dem neuesten Stand.

Die in den folgenden Kapiteln *dieses* Handbuchs (#85110) enthaltene Beschreibung entspricht daher nicht unbedingt dem neuesten Stand der Entwicklung !

7. Beschreibung des Programmierwerkzeugs

7.1 Das Hauptfenster

Das Hauptfenster wird direkt nach dem Start des Programmierwerkzeugs angezeigt.



(Hinweis: Möglicherweise verfügt die neueste Version des Programmierwerkzeugs über mehr Funktionen als die hier gezeigten; der im Fenstertitel angezeigte Programmname hängt von der verwendeten Programm-Variante ab)

Das Hauptfenster enthält verschiedene Seiten auf „Registerkarten“ (engl. tab sheets):

- CANdb / CANopen: Dient zur Definition von Kommunikations-Kanälen (CANopen: PDO, SDO; das Tool für "CANdb-Terminals" bietet stattdessen eine Registerkarte zum Import von CANdb-Dateien)
- Variablen: für die Definition von Variablen des Anwenderprogramms.
- Globale Events: Definition von "globalen" Ereignissen (Seiten-unabhängig).
- Seite #x: Hier wird alles definiert, was auf einer bestimmten Display-Seite erscheinen soll. Auf dieser Registerkarte finden Sie u.A. auch den weiter unten beschriebenen Editor für Anzeigeseiten.
- Seitenübersicht: Zeigt eine Übersicht mit allen programmierten Anzeigeseiten an.
- Symbole: Dient zum Einfügen von Icons (kleinen Grafikbildern) in das Anwenderprogramm.
- Einstellungen: Allgemeine Einstellungen des Programmierwerkzeugs und Informationen über das Zielsystem (Terminal)
- Fehler: Zeigt verschiedene Fehlermeldungen des UPT-Systems und andere Infos an.
- Script: Registerkarte [Script-Editor](#) (<- externer Link) und -Debugger. Nur für aktuelle Geräte mit 32-Bit-CPU, wie z.B. MKT-View II / III !

Am unteren Rand des Hauptfensters befindet sich die Statuszeile, in der verschiedene Systemzustände und Fehlermeldungen angezeigt werden können. Die Statuszeile wird in einem der folgenden Kapitel genauer vorgestellt.

7.2 Das Hauptmenü



Das Hauptmenü dient zum

- Laden von UPT-Programmen aus Dateien (*.upt bei CANopen, *.cvt bei CANdb)
- Speichern von UPT-Programmen als ASCII-Datei
- Übertragen von Programmen zwischen Programmierool und Terminal (via CAN)
- Öffnen einiger Hilfsfenster für Debugging etc („View“)
- Importieren oder Erzeugen neuer Icons für das UPT-Anwenderprogramm
- Starten des Hilfe-Systems
- Start und Stoppen des Interpreters, der auch im „echten“ Terminal arbeitet

7.3 Statuszeile

In der Statuszeile werden einige Informationen über den aktuellen Systemzustand angezeigt, z.B. Meldungen oder Übertragungsfortschritt.



Durch Anklicken des „Simulator“-Button kann zum Fenster des LCD-Simulators umgeschaltet werden.

Das zweite Panel der Statuszeile (welches hier „Running“ anzeigt) liefert eine schnelle Information über den aktuellen Zustand des Programmierools oder des UPT-Anwenderprogramms. Einige mögliche Anzeigen sind:

- Running: Ihr UPT-Anwenderprogramm läuft (der Simulator arbeitet es ab)
- Stopped: Das UPT-Anwenderprogramm wurde ausirgendwelchen Gründen angehalten.
- Transfer: Derzeit läuft eine Programmübertragung zwischen Programmierool und Terminal.

Das rechte Feld in der Statuszeile kann die letzte Fehlermeldung oder eine andere System-Meldung anzeigen. Im oben gezeigten Beispiel besagt die Meldung, dass das CAN-Interface erfolgreich initialisiert werden konnte.

Sie können durch Doppelklick in das Meldungsfeld der Statuszeile zur „Error-Seite“ umschalten, auf der Sie auch ältere Fehler noch sehen können.

7.4 Das Hilfe-System des Programmiertools

Im Hilfe-System werden Sie immer die aktuellste Beschreibung des UPT-Programmiertools finden. Das komplette Hilfesystem basiert auf HTML ("ohne Gimmiks"), und ist sowohl [online](#), als auch (nach der Installation) auf der lokalen Festplatte des Entwicklers verfügbar.

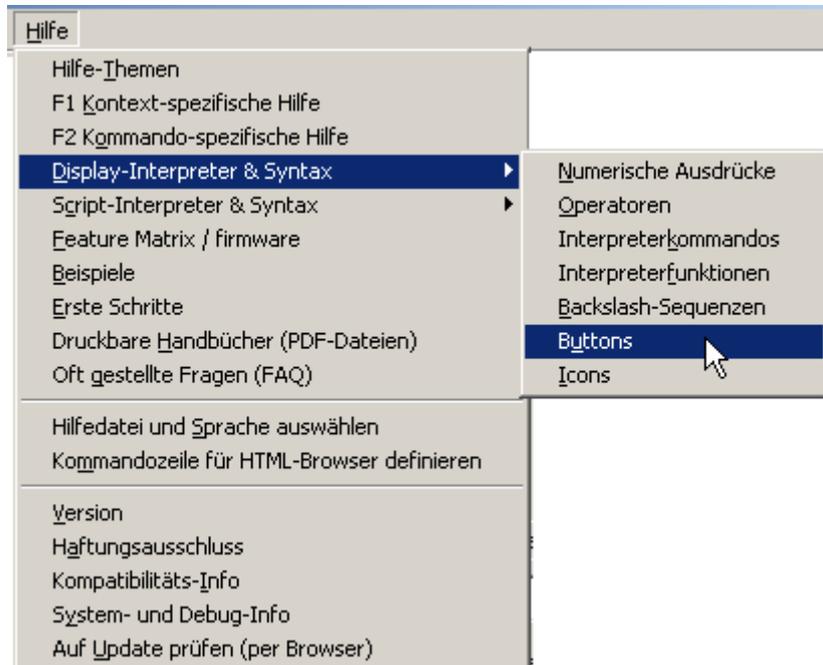


Das Hilfesystem kann folgendermaßen gestartet werden:

- durch Anklicken des Menüpunktes „Help“ im Hauptmenü, um eine Übersicht über alle verfügbaren Hilfethemen zu erhalten,
- durch Anklicken eines der zahlreichen „Help“-Buttons um eine Hilfestellung zu einem bestimmten Thema (abhängig von der aktuellen Registerkarte) zu erhalten,
- durch Drücken von F1 um eine kontext-abhängige Hilfe zum aktuellen Dialogelement zu erhalten, z.B. zu einem bestimmten Eingabefeld;
- durch Drücken von F2 um eine Hilfe zu einem UPT-Interpreter-Kommando zu erhalten, falls der Eingabecursor in einem Eingabefeld steht.

Sollten Sie Probleme haben, weil der Browser Hilfedatei nicht finden kann, können Sie den Pfad (nicht den Dateinamen!) unter „General Settings“ einstellen. Mit dieser Einstelloption können Sie gegebenenfalls auch eine anderssprachige Hilfedatei verwenden, falls eine übersetzte Hilfedatei von MKT Systemtechnik verfügbar ist.

Die *meisten* (leider nicht alle) HTML-Browser können per Kommandozeile direkt zum passenden Thema umschalten. In dem Fall empfiehlt sich der Aufruf des Hilfesystems über das Hauptmenü des Programmiertools:



Wir empfehlen dringend, bei der Einarbeitung in das Programmierool in erster Linie das Online-Hilfe-System zu verwenden, da dieses grundsätzlich auf dem neuesten Stand ist !

7.5 Import von CAN-Datenbanken aus *.dbc-Dateien

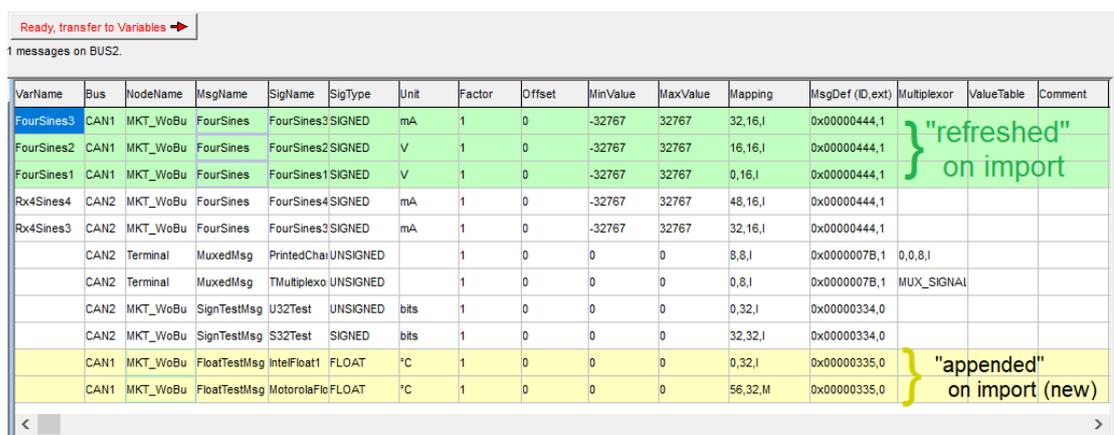
Bei Geräten für den Einsatz im Fahrzeug wird die Kommunikation per CAN / CAN FD üblicherweise durch den Import einer oder mehrerer zum KFz passenden Datenbanken festgelegt (*.dbc = Database For CAN; der Import von AUTOSAR XML ist zwar beabsichtigt, stellte sich aber noch als wesentlich komplexer heraus als ursprünglich 'befürchtet').

Details finden Sie auch zu diesem Thema in der 'Online-Hilfe' (die nach der Installation des Programmierwerkzeugs auch lokal auf Ihrer Festplatte vorhanden sein wird). Viele Optionen sind nur in der Online-Hilfe beschrieben.

Um Signaldefinitionen aus einer DBC-Datei zu importieren, schalten Sie zunächst auf die Registerkarte "CANdb" im Hauptfenster des Programmierwerkzeugs um.

Falls dort noch alte Signaldefinitionen (und Verknüpfungen zwischen CAN-Signalen und Variablen) stehen, die Sie nicht länger brauchen, klicken Sie auf *Menü ... Lösche die ganze Tabelle* bevor Sie ein oder mehrere DBC-Dateien importieren. In den meisten Fällen (wenn z.B. mehrere Datenbanken für mehrere Busse vorhanden sind, oder bereits vorhandene Verknüpfungen zwischen CAN-Signalen und Variablen erhalten bleiben sollen, importieren Sie die DBC-Datei einfach 'über' die bereits in der Tabelle vorhandenen Definitionen hinweg. Das Programmierwerkzeug erkennt bereits vorhandene Definitionen anhand des Knoten-, Message- und Signalnamens und *ersetzt* diese in der Tabelle, statt sie am Ende der Tabelle anzufügen.

Alle Zeilen in der 'CANdb'-Tabelle, die beim Neu-Laden der CAN-Datenbank erfolgreich *aktualisiert* (aber nicht *neu angelegt*) wurden, werden mit einem **grünen Hintergrund** markiert. Bei diesen Einträgen bleiben eventuell schon vorhandene Variablenamen wie im unten gezeigten Screenshot erhalten.



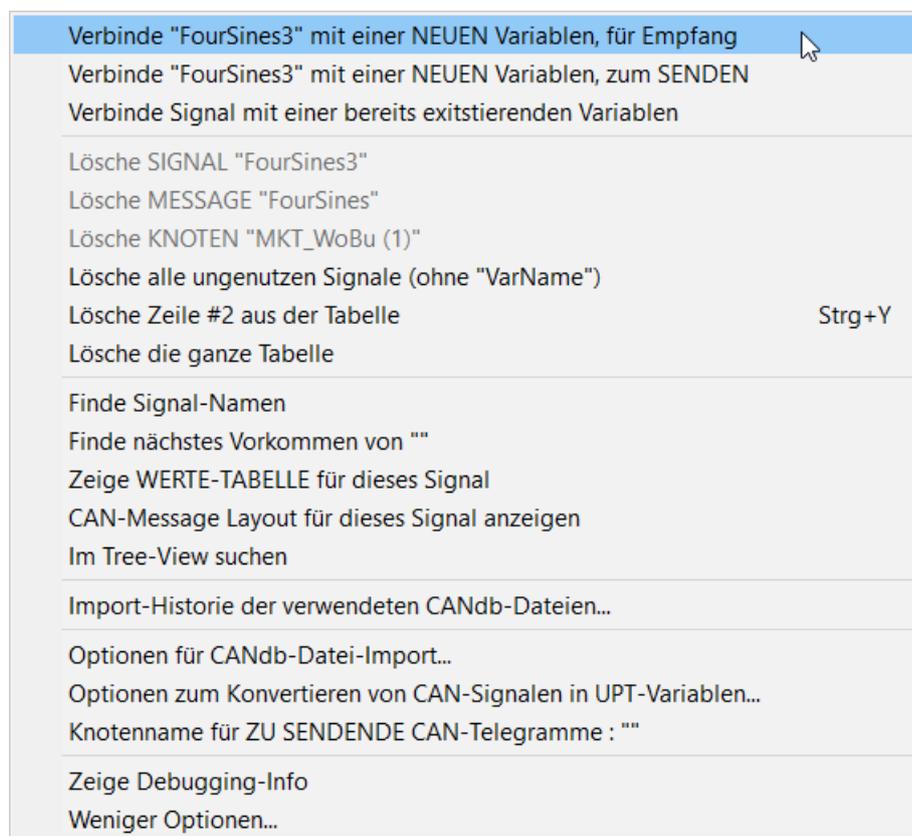
VarName	Bus	NodeName	MsgName	SigName	SigType	Unit	Factor	Offset	MinValue	MaxValue	Mapping	MsgDef (ID, ext)	Multiplexor	ValueTable	Comment
FourSines3	CAN1	MKT_WoBu	FourSines	FourSines3 SIGNED	mA		1	0	-32767	32767	32,16,I	0x00000444,1			} "refreshed" on import
FourSines2	CAN1	MKT_WoBu	FourSines	FourSines2 SIGNED	V		1	0	-32767	32767	16,16,I	0x00000444,1			
FourSines1	CAN1	MKT_WoBu	FourSines	FourSines1 SIGNED	V		1	0	-32767	32767	0,16,I	0x00000444,1			
Rx4Sines4	CAN2	MKT_WoBu	FourSines	FourSines4 SIGNED	mA		1	0	-32767	32767	48,16,I	0x00000444,1			
Rx4Sines3	CAN2	MKT_WoBu	FourSines	FourSines3 SIGNED	mA		1	0	-32767	32767	32,16,I	0x00000444,1			
	CAN2	Terminal	MuxedMsg	PrintedChar	UNSIGNED		1	0	0	0	8,8,I	0x0000007B,1	0,0,8,I		
	CAN2	Terminal	MuxedMsg	TMultiplexo	UNSIGNED		1	0	0	0	0,8,I	0x0000007B,1	MUX_SIGNAL		
	CAN2	MKT_WoBu	SignTestMsg	U32Test	UNSIGNED	bits	1	0	0	0	0,32,I	0x00000334,0			
	CAN2	MKT_WoBu	SignTestMsg	S32Test	SIGNED	bits	1	0	0	0	32,32,I	0x00000334,0			
	CAN1	MKT_WoBu	FloatTestMsg	IntelFloat1	FLOAT	°C	1	0	0	0	0,32,I	0x00000335,0			} "appended" on import (new)
	CAN1	MKT_WoBu	FloatTestMsg	MotorolaFlo	FLOAT	°C	1	0	0	0	56,32,M	0x00000335,0			

Neue Einträge aus der importierten Datei, die am Ende der Tabelle *angehängt* wurden (ohne existierende Einträge zu überschreiben) erhalten einen **gelben Hintergrund**.

Für die *Anzeige* kann in vielen Fällen einfach der *Signalname* als *Variablenname* übernommen werden. Klicken Sie dazu mit der rechten Maustaste in die Spalte *VarName*, und wählen im daraufhin angezeigten Kontext-Menü die Funktion

Verbinde "XYZ" mit einer NEUEN Variablen, für Empfang

(darin ist "XYZ" wie im folgenden Screenshot der Name des CAN-Signals in der momentan selektierten Tabellenzeile) .



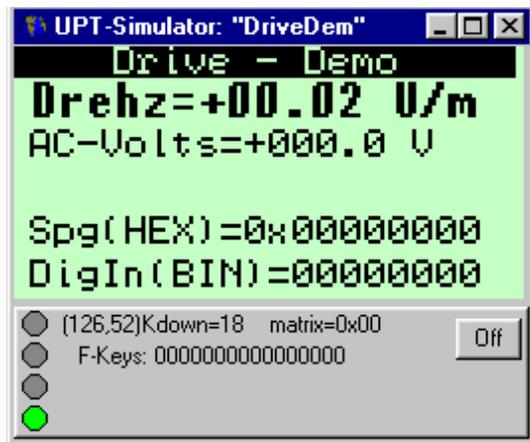
Sind alle CAN-Datenbanken importiert, und die für die Applikation (Anzeige) benötigten Variablen mit den entsprechenden CAN-Signalen (oder entsprechenden Feldern in PDUs = Process Data Units) verbunden, klicken Sie auf den nun blinkenden Button 'Fertig, in Variablen umsetzen' ('Ready, transfer to Variables'), um *Variablen* für die Anzeige zu erzeugen.

Wie bereits erwähnt, sind im Programmiertool viele weitere Optionen für den Import bzw. die Aktualisierung bereits importierter Datenbanken vorhanden, deren Beschreibung den Rahmen dieses PDF-Dokumentes sprengen würden. Verwenden Sie daher ggf. das Hilfesystem aus dem aktuellen Programmiertool. Das entsprechende Kapitel finden Sie auch Online auf der Webseite des Herstellers (Stand 01 / 2021) :

https://www.mkt-sys.de/MKT-CD/upt/help/candb_49.htm

7.6 Das Fenster des LCD-Simulators

Dieses Fenster zeigt den Inhalt des „simulierten“ LCD-Bildschirms an, den Sie sehen **würden**, nachdem Sie das UPT-Programm in das „echte“ Terminal geladen haben.



In den meisten Fällen zeigt dieses Fenster den Inhalt der „aktuellen“ UPT-Seite an (außer z.B. während der Programmübertragung).

Solange das LCD-Simulator-Fenster den Tastatureingabe-Fokus hat (erkennbar am blauen Fenstertitel), werden alle Tastatureingaben an den „Terminal-Simulator“ weitergeleitet.

Dadurch ist es möglich, fast alle Funktionen (insbesondere die „Ereignisverarbeitung“) des von Ihnen geschriebenen UPT-Anwenderprogramms zu testen.

Das LCD-Simulator-Fenster kann unabhängig vom Hauptfenster verschoben und vergrößert werden. Es

kann auch im Vollbild-Modus dargestellt werden, vorausgesetzt der verwendete PC ist schnell genug für eine flüssige Darstellung.

Durch Mausklick in das LCD-Simulator-Fenster können Sie einzelne Textzeilen selektieren und verschieben. Das Hauptfenster des UPT-Programmierwerkzeugs schaltet dann automatisch zur Definition von Display-Zeilen um, wo Sie die Eigenschaften der angeklickten oder verschobenen Zeile kontrollieren können (siehe Kapitel 7.10.3).

Das Verschieben von Displayzeilen mit gedrückter Maustaste funktioniert nicht bei grafischen Objekten, die als Grafikbefehl manuell eingegeben werden (siehe Kap. Fehler: Verweis nicht gefunden).

Auf dem Panel am unteren Rand des Simulator-Fensters sehen Sie den Zustand einiger LEDs und der Funktionstasten des simulierten UPTs. Das genaue Aussehen dieses Fensters wird sich in zukünftigen Versionen noch ändern, es hängt auch von den Fähigkeiten des verwendeten „echten“ Terminals ab. Möglicherweise wird hier auch der Zustand einiger digitaler Ein- und Ausgänge des UPTs angezeigt (z.B. UPT515 mit 6 digitalen Inputs und 4 digitalen Outputs).

7.7 Kommando-Interpreter

Der Kommando-Interpreter ist ein Teil der Firmware im UPT, er ist allerdings auch (in fast identischer Form) in der Software des UPT-Programmiertools implementiert. Er dient im Wesentlichen zum Abarbeiten Ihres „Anwenderprogramms“. Dazu gehört z.B.:

- das Berechnen numerischer Ausdrücke („Formeln“)
- das Abarbeiten von Ereignis-Definitionen und Ereignis-Reaktionsmethoden
- das Ausführen komplexer Grafik-Kommandos (siehe Hilfe-System)

Eine Übersicht einiger Interpreter-Kommandos findet sich im Anhang dieses Dokuments.

Sie können Interpreter-Kommandos zu Testzwecken auch in der Statuszeile des Programmiertools eingeben oder in einem speziellen „Test/Kommando“-Fenster.

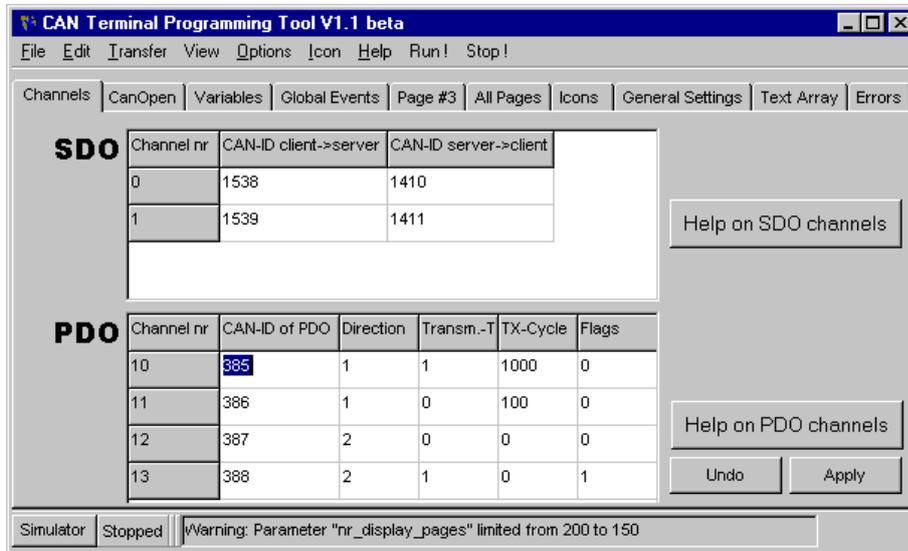
Eine Übersicht des Sprachumfangs des Interpreters finden Sie in Kapitel 8.

Alle Details finden Sie in der Online-Hilfe. Hier der direkte Link zum Kommando-Interpreter:

http://www.mkt-sys.de/MKT-CD/upt/help/progt_49.htm#interpreter

7.8 Kommunikationskanäle

Die folgende Seite des Programmierwerkzeugs dient zur Definition aller Kommunikationskanäle, mit denen das UPT-Terminal später Daten mit anderen Geräten im CANopen-Netzwerk austauschen soll:



Es gibt verschiedene Arten von Kommunikationskanälen:

- SDO - Kanal (Service Data Object, nur beim UPT bzw. Geräten mit **CANopen**)
Eine relativ langsame Punkt-zu-Punkt-Verbindung, mit der Daten aus dem Objektverzeichnis eines CANopen-kompatiblen Gerätes gelesen und/oder geschrieben werden können.
- PDO - Kanal (Process Data Object, nur beim UPT bzw. Geräten mit **CANopen**)
Ein „schneller“ Übertragungskanal, mit dem üblicherweise unquittierte Prozeßdaten mit hoher Priorität übertragen werden, zum Beispiel der Zustand digitaler Eingänge, analoger Sensoren usw. .
- CANdb-Messages (für Geräte mit Parametrierung per DBC, z.B. "MKT-View")
Entspricht ungefähr einem CANopen-RPDO, es sind allerdings beliebige 11- oder 29-Bit-CAN-Identifizierer möglich. In einer "CANdb-Message" können bis zu 64 "Signale" enthalten sein. Ein "Signal" kann mit einer UPT-Variablen verbunden sein. Die Kanalnummer ist in diesem Fall fest auf "30" eingestellt. Weitere Informationen zur Verwendung von CANdb-Message finden sie im [Zusatz 'CANdb' zu dieser Bedienungsanleitung](#).

Beachten Sie, dass das UPT-Terminal normalerweise nur an PDO-Kanälen „lauschen“ soll. Es kann derzeit keine PDO-Übertragungen von anderen Geräten (per RTR) anfordern, weil dies Kollisionen hervorrufen würde. Bei Geräten mit 16- oder 32-Bit-CPU besteht diese Einschränkung nicht.

Sie müssen mindestens **einen** Kommunikations-Kanal für jedes Gerät definieren, mit dem das UPT kommunizieren soll. Die Kanäle werden später verwendet, um Werte von externen Geräten in die Variablen Ihres Anwenderprogramms zu transportieren.

Details (online) : [Hilfesystem zum Programmierwerkzeug / "Kommunikations-Kanäle"](#)

7.8.1 SDO-Kanäle

Nur für Geräte mit CANopen-kompatiblen Übertragungsprotokollen (z.B. UPT-515).
 Nicht für Geräte ohne CANopen-Funktionalität (z.B. MKT-View) !

SDO-Kanäle bieten eine relativ langsame „Punkt-zu-Punkt“-Verbindung zum Austausch von Daten über das sog. *Object Dictionary* eines CANopen-kompatiblen Geräts.

Dazu müssen Sie die CAN-Identifizier aller SDO-Kanäle definieren, die Ihre Anwendung später zur Kommunikation benötigt. Sie finden diese Identifizier in der Beschreibung des externen Gerätes oder im CANopen Draft Standard 301 (CiA „DS301“). Die meisten I/O-Module verwenden eine Standardeinstellung (nach CiA „pre-defined connection set“ mit den folgenden Definitionen für SDO: (DS301 nennt den CAN-Identifizier „COB-ID“)

SDO(server->client) :	CAN-Identifizier	= (1408+node-ID)	= 1409...1535
SDO(client->server) :	CAN-Identifizier	= (1536+node-ID)	= 1537..1663

Alle hier angegebenen CAN-Identifizier sind dezimal notiert. Der „client->server“-ID wird vom Client (hier: UPT) zum Server (z.B. I/O-Modul) gesendet, um eine einzelne Datenübertragung anzustossen („request“). Der „server->client“-ID wird für die Antwort vom Server zurück zum Client verwendet.

Der „node ID“ wird bei vielen einfachen Geräten per DIP-Schalter o.ä. eingestellt, er liegt im Bereich 1..127.

Bitte beachten Sie:

Eine SDO-Verbindung ist lediglich eine „Punkt-zu-Punkt“-Verbindung zwischen genau **zwei** Partnern. **Sie** müssen sicherstellen, daß Sie für das UPT nur SDO-Verbindungen verwenden, die für diesen Zweck noch frei sind (und nicht bereits von anderen Geräten im Netzwerk belegt sind).

Wenn ein einfaches I/O-Modul nur einen einzigen SDO-Kanal hat, und dieser bereits von einem anderen Gerät (z.B. SPS, PLC, Antriebsregler) belegt ist, dann dürfen Sie diesen

Kanal **nicht** für die Kommunikation mit dem UPT einsetzen, weil dies schwere Kollisionen im CAN-Netzwerk mit dem „Absturz“ einiger Teilnehmer hervorrufen könnte ! (Grund: der gleiche CAN-ID würde von mehreren Teilnehmern gleichzeitig gesendet, es würden SDO-Protokollverstöße auftreten etc.)

Im UPT wird ein SDO-Kanal dazu verwendet, eine *Variable* mit einem anderen Gerät zu koppeln.

Details (online) : [Hilfesystem zum Programmierool / "SDO-Kanäle"](#)

7.8.2 PDO-Kanäle

Nur für Geräte mit CANopen-kompatiblen Übertragungsprotokollen (z.B. UPT-515).
 Nicht für Geräte ohne CANopen-Funktionalität (z.B. MKT-View) !

PDO-Kanäle dienen zum schnellen Datenaustausch von Prozeßdaten mit hoher Priorität, z.B. für digitale Eingänge, Sensoren etc.

Sie müssen die CAN-Identifizier aller PDO-Kanäle definieren, die das UPT senden oder empfangen soll. Sie finden diese Identifizier in der Beschreibung des externen Gerätes oder im CANopen Draft Standard 301 (CiA „DS301“). Ähnlich wie bei den SDOs verwenden die meisten I/O-Module eine Standardeinstellung (predefined connection set) mit dem folgenden Definitionen für PDO-CAN-Identifizier („COB-IDs“):

PDO1(tx) :	CAN-Identifizier	= (384+node-ID)	= 385...511
PDO1(rx) :	CAN-Identifizier	= (512+node-ID)	= 513...639
PDO2(tx) :	CAN-Identifizier	= (640+node-ID)	= 641...767
PDO2(rx) :	CAN-Identifizier	= (768+node-ID)	= 769...895

Alle hier gezeigten CAN-Identifizier sind dezimal notiert. Beachten Sie, daß „rx“(Empfang) und „tx“(Senden) in dieser Tabelle z.B. aus der Sicht eines I/O-Moduls zu sehen ist; d.h. die „tx“-PDOs werden vom I/O-Modul (z.B. digitale Inputs) an das UPT **gesendet**; die „rx“-PDOs werden vom UPT gesendet und vom I/O-Modul **empfangen** und werden dort z.B. für die Ansteuerung digitaler Ausgänge verwendet.

Der „node-ID“ liegt –wie bei den SDOs- im Bereich 1..127, er wird bei einfachen I/O-Modulen per DIP-Schalter eingestellt.

Beispiel: Sie wollen mit dem UPT den Zustand der digitalen Eingänge eines I/O-Moduls anzeigen.

Das I/O-Modul verwendet einen ersten PDO-Kanal(tx) zum Senden von Prozessdaten, in denen der aktuelle Zustand der digitalen Eingänge enthalten ist. Der node-ID des I/O-Moduls steht auf „1“, daher ist der resultierende CAN-Identifizier dezimal $(384+1) = 385$.

Um diesen PDO mit dem UPT zu empfangen, wollen Sie den ersten PDO-Kanal des UPTs verwenden. Tragen Sie den Wert „385“ in die Spalte „CAN-ID vom PDO“ ein und setzen Sie die Übertragungsrichtung („Direction“, aus Sicht des UPTs) auf „Empfangen“.

Das UPT ist nun in der Lage, PDO-Telegramme mit dem CAN-Identifizier 385 zu empfangen und die darin enthaltenen Werte auf verschiedene Variablen Ihres Anwenderprogramms zu „verteilen“.

Im UPT kann eine Variable an einen TX-PDO oder an einen RX-PDO „angeklemmt“ werden, aber nicht an mehrere PDO-Kanäle gleichzeitig. Sie können an einen PDO-Kanal allerdings mehrere Variablen „anklemmen“.

Hinweis !

In einem CAN-Netzwerk darf ein CAN-Identifizier nur durch ein einziges Gerät gesendet werden. Darum müssen **Sie** sicherstellen, daß alle PDOs, die das UPT sendet, von keinem anderen Gerät im gesamten Netzwerk gesendet wird. Dies kann in einem „unbekanntem“ Netzwerk eine schwierige Aufgabe sein...

Einige Komponenten einer PDO-Kanal-Definition werden in den folgenden Kapiteln beschrieben. Eine detaillierte (und aktuelle) Beschreibung finden Sie allerdings **nur** im Hilfe-System des UPT-Programmierwerkzeugs !

Details (online) : [Hilfesystem zum Programmierwerkzeug / "PDO-Kanäle"](#)

7.8.3 Kommunikationskanal für CANdb-Signale

(Nicht für CANopen, sondern ausschließlich für den im KFZ-Bereich verwendeten "Mobil-Bus", implementiert z.B. in der **MKT-View** - Serie)

Kanalnummer 30 ist beim Programmierwerkzeug für das "CANdb-Anzeige-Terminal" als Dummy für alle Variablen reserviert, die an CANdb-kompatible **Signale** gekoppelt sind.

CANdb steht als Kürzel für "CAN Data Base". In einer CAN-Datenbank-Datei (*.DBC) sind sogenannte **Messages** und **Signale** definiert. Das Programmierwerkzeug bietet die Möglichkeit, solche DBC-Dateien einzulesen und die darin enthaltenen Message- und Signaldefinitionen in die Definition von UPT-Variablen umzusetzen.

Wie dies genau funktioniert, ist ausschließlich dem Online-Hilfe-System des Programmierwerkzeugs zu entnehmen. Schalten Sie im Programmierwerkzeug auf die Seite "CANdb" um und klicken Sie auf den "Help" - Button.

Details (online) : http://www.mkt-sys.de/MKT-CD/upt/help/candb_49.htm

7.9 Variablen des Anwenderprogramms

Die folgende Tabelle des UPT-Programmierwerkzeugs dient zur Definition aller Variablen des Anwenderprogramms. Diese Variablen werden u.A. benötigt, wenn das Terminal Parameterwerte anzeigen und/oder editieren soll.

Nr	Name	Channel	PDO/SDO-Definition	Upd-Time	Type	Access
0	AnOut1	1 (SDO)	ob=\$6411.01	1000 ms	3	1
1	AnOut2	1 (SDO)	ob=\$6411.02	1000 ms	3	1
2	DigIn	10 (PDO)	cb=8, fb=0	2000 ms	5	1
3	DigIn2	10 (PDO)	cb=8, fb=8	3000 ms	5	1
4	Drehz1	0 (SDO)	ob=\$6401.01	500 ms	3	1
5	PDOtest	10 (PDO)	cb=16, fb=24	1500 ms	3	1
6	PDOtest2	10 (PDO)	cb=1, fb=48	1500 ms	3	1
7	Spanng	0 (SDO)	ob=\$6401.02	1000 ms	3	1

Zu jeder Variablen gehören:

- Name
- Kanal (für Kommunikation)
- PDO/SDO – Definition
- Update – Time
- Data Type
- Access Rights
- Flags
- Default Value
- Minimum Value
- Maximum Value
- Factor, Divisor, Offset

Die meisten Variablen im Anwenderprogramm werden über einen Kommunikationskanal mit einem externen Gerät verbunden sein. Variablen können allerdings auch (ohne Anbindung über Kommunikationskanal) zur Aufnahme von Zwischenergebnissen bei numerischen Berechnungen verwendet werden usw.

Anhand der Kanalnummer erkennt das UPT auch, um welchen Kanaltyp es sich handelt:

- Kanal 0..9 wird für SDO-Client-Kanäle verwendet bzw. reserviert,
- Kanal 10..19 dient für PDO-Kanäle (mit „Reserve“),
- Kanal 20.29 dient für derzeit noch nicht implementierte „serielle“ Kanäle,
- Kanal 30 dient für "**Signale**" die per "CANdb"-Datei definiert wurden,
- Kanal 255 bedeutet „kein Anschluß an Kommunikationskanal“.

Details zu den anderen Spalten (Komponenten) einer Variablen-Definition finden Sie im Hilfesystem des Programmierwerkzeugs.

Details (online) : http://www.mkt-sys.de/MKT-CD/upt/help/progt_49.htm#variables

7.10 Definition von Anzeigeseiten

Die Registerkarte „Page #n“ (Seite #n) dient zur Definition einer einzelnen Bildschirmseite, die Sie später auf dem LCD des UPTs sehen können.

Die Definition einer Bildschirmseite umfaßt

- einen Seiten-Definitions-Kopf (Header), in dem einige Informationen zusammengefaßt sind die sich auf die gesamte Seitendefinition beziehen
- mehrere Text-Anzeige-Definitionen (Anzeigefelder, Editierfelder, statische Texte)
- optionale Kommandos für Grafikausgaben (und andere Spezialitäten)
- Ereignis-Definitionen (siehe Kapitel 7.12)

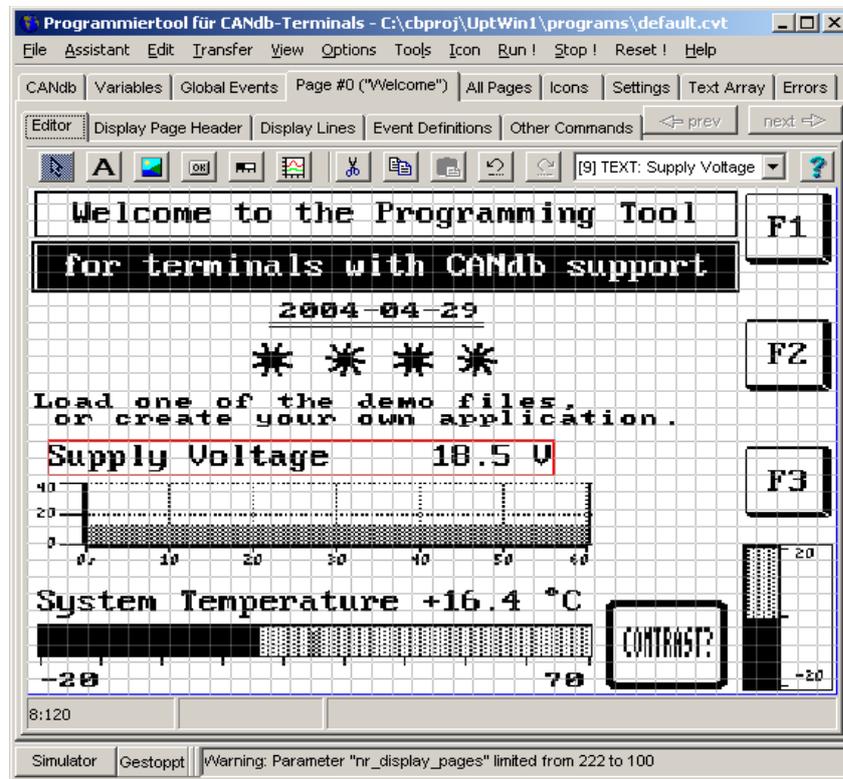
Um im Programmiertool zu einer anderen Bildschirmseite Ihres Anwenderprogramms umzuschalten, können Sie die Seiten-Übersicht verwenden auf der alle Bildschirmseiten als Miniaturen dargestellt sind (siehe Kapitel 7.15).

Am einfachsten lassen sich Anzeigeseiten mit dem unten kurz vorgestellten graphischen Seiten-Editor erstellen. Fortgeschrittene Anwender bevorzugen aber meistens die Definition von Anzeigeseiten ohne viel Mausklicks durch direkte Texteingabe in tabellarischer Form. Die verschiedenen Möglichkeiten zur Definition von Anzeigeseiten werden in *diesem* Dokument (#85110) nur kurz umrissen. Alle Details finden Sie -wie immer- im aktuelleren Hilfesystem (HTML), z.B. online unter:

http://www.mkt-sys.de/MKT-CD/upt/help/progt_49.htm#page_defs

7.10.1 Der "einfache" Editor für Anzeigeseiten

Für die Erstellung oder Bearbeitung einfacher Anzeigeseiten reicht oft der im Folgenden beschriebene 'einfache' graphische Editor:



Die Toolbar des "einfachen" Anzeigeseiten-Editors enthält die folgenden Funktionen (Buttons) :

-  (Pfeil-Werkzeug) : Verschieben, Selektieren oder Bearbeiten von Elementen der Anzeige.
 Solange dieses Tool aktiv ist, kann -bei gedrückter linker Maustaste- das markierte Anzeigeelement verschoben werden.
 Ein kurzer, einzelner Klick mit der linken Maustaste selektiert *ein* Objekt zum Ausschneiden / Kopieren / Einfügen.
 Um mehrere Elemente zu markieren, halten Sie die STRG-Taste ("Control-Taste") gedrückt, oder ziehen mit der linken Maustaste einen Selektionsrahmen auf. Details zur Mehrfachauswahl folgen [unten](#).
 Per Doppelklick wird ein spezieller Einstelldialog gestartet, in dem das markierte Element bearbeitet werden kann.
-  (Text-Werkzeug) : Einfügen einer alphanumerischen Anzeigezeile.
 Aktivieren Sie zunächst dieses Tool (durch Klick auf das Symbol in der Toolbar).
 Danach kann mit einem einzelnen Klick eine neue Textzeile auf der Anzeige platziert werden.
 Nach Einfügen der Textzeile kann diese in einem speziellen Einstelldialog bearbeitet werden.

(Icon-Werkzeug) : Einfügen eines Icons (Symbol; Bitmap-Grafik).

Mit diesem Tool können Sie kleine Grafiken ("Icons") auf der Seite einfügen.

Hinweis: Die Grafiken müssen vorher auf der Registerkarte "[Symbole](#)" (engl. "Icons") importiert werden.

Nach Einfügen der Grafik kann deren Erscheinungsbild ggf. angepasst werden (Farbe, Vergrößerung, unterschiedliche Symbole in Abhängigkeit einer Variablen, etc.) Verwenden Sie dazu das Pfeil-Werkzeug, und einen Doppelklick in die neu eingefügte Grafik.



(Button-Werkzeug) : Fügt einen neuen Button (graphische Schaltfläche) ein.

Selektieren Sie dieses Tool, und klicken Sie in die Zeichenfläche, um dort einen neuen Button einzufügen.

Danach kann der Button in einem speziellen Einstelldialog konfiguriert werden.



(Bargraph-Werkzeug) : Fügt ein neues Balkendiagramm ein.

Wenn dieses Tool aktiv ist, wird per Mausklick in die Zeichenfläche dort ein neues Balkendiagramm eingefügt.

Dieses kann danach in einem speziellen Einstelldialog konfiguriert werden.



(Diagramm-Werkzeug) : Fügt ein Y(t) oder X/Y-Diagramm ein.

Ist dieses Tool aktiv, kann per Mausklick ein neues Diagramm eingefügt werden.



(Ausschneiden) : Schneidet das selektierte Anzeige-Element aus ("cut").

Selektieren Sie zunächst das Element, welches entfernt werden soll (mit dem Pfeil-Tool), und klicken danach das Tool zum Ausschneiden an (Schere).

Beim Ausschneiden wird das selektierte Objekt in eine interne Ablage kopiert (s.U.).



(Kopieren) : Kopiert das selektierte Element in eine interne Ablage ("copy").

Selektieren Sie zunächst das Element, welches in die Ablage kopiert werden soll (mit dem Pfeil-Tool),

und klicken danach auf das Tool zum Kopieren .

Das selektierte Objekt wird in eine interne Ablage kopiert .

Hinweis: diese Ablage ist nicht mit der Windows-Zwischenablage zu verwechseln !



(Einfügen) : Fügt das Element aus der internen Ablage wieder auf der Anzeige ein ("paste").

Wählen Sie ggf zunächst mit dem Pfeil-Tool das zu überschreibende Element.

Wenn Sie beim Einfügen nichts überschreiben wollen, selektieren Sie vorher nichts (z.B. Klick ins Leere).

Das in der internen Ablage gespeicherte Element wird auf der Zeichenfläche eingefügt, und kann anschließend verschoben werden.

War vor dem Einfügen ein Element markiert, so wird dieses durch das eingefügte Element *ersetzt*.

War vor dem Einfügen kein Element markiert, so wird das neue Element an der Position eingefügt, die durch den Einfüge-Cursor (roter Punkt) markiert ist.

Der Einfüge-Cursor kann (vor dem Einfügen) durch einen kurzen Mausklick "ins Leere" umgesetzt werden. Beim Einfügen eines Elementes wird der Einfügekursor entsprechend der Größe des eingefügten Objektes weiterbewegt, so daß sich beim erneuten Einfügen keine

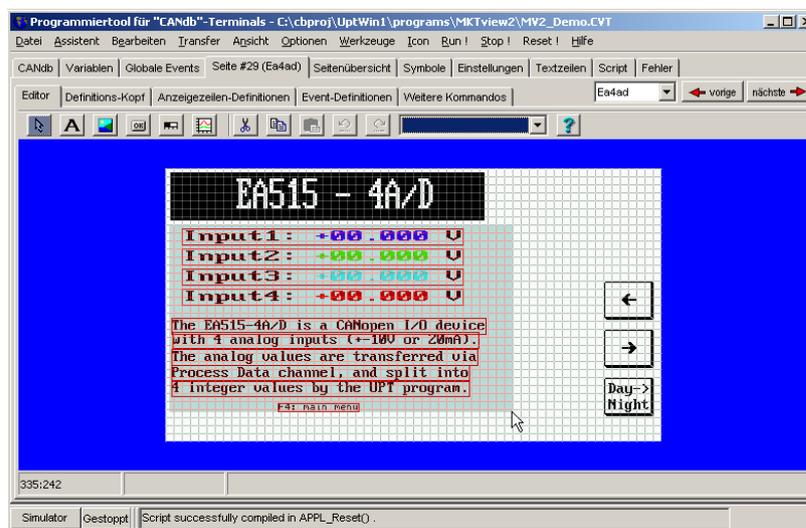
Überlappung ergibt.

-  (Rückgängig, engl. "Undo") : Letzte Aktion(en) rückgängig machen.
 Ein Klick auf den 'UNDO'-Button macht die letzte Änderung auf der aktuellen Anzeigeseite rückgängig.
 In diesem Editor können nacheinander bis zu 20 Aktionen rückgängig gemacht werden, da auf einem entsprechend großen Stapelspeicher abgelegt werden.
 Vorsicht: Wenn Sie die aktuelle Anzeigeseite im Editor verlassen, können die einzelnen Änderungen auf der Seite nicht mehr rückgängig gemacht werden !

-  (Wiederrufen, engl. "Redo" oder "Un-Undo") : Letzte Unfo-Aktion widerrufen.
 Wenn Sie versehentlich zu oft "UNDO" aufgerufen haben (s.O.), können Sie die letzten UNDO-Aktionen durch Anklicken von REDO wiederrufen (d.h. das "Rückgängig-machen wieder rückgängig machen", englisch "un-undo").
 Im Anzeigeseiten-Editor können bis zu 20 vorhergehende Undo-Aktionen rückgängig gemacht werden, da auch zum Wiederrufen ein entsprechend großer Speicher zur Verfügung steht.
 Vorsicht: Wenn Sie die aktuelle Anzeigeseite im Editor verlassen, kann -ählich wie bei Undo- auch kein REDO für die aktuelle Seite mehr erfolgen !

Das selektierte Objekt wird in der Zeichenfläche mit einem (bzw mehreren) roten Rahmen markiert.

Um **mehrere Objekte** zu markieren, aktivieren Sie zunächst wie oben beschrieben das 'Pfeil'-Werkzeug. Drücken Sie danach die STRG-Taste (bzw CTRL) und halten sie gedrückt, um per Klick mit der linken Maustaste weitere Objekte zur aktuellen Selektion hinzugefügt werden. Alternativ können mehrere 'nebeneinander' liegende Objekte wie folgt markiert werden: Bewegen Sie den Mauszeiger zunächst über einen freien Bereich (in dem kein Objekt steht), links oberhalb der linken oberen Ecke des ersten zu markierenden Objekts. Drücken und halten Sie anschließend die linke Maustaste, während sie die Maus nach rechts unten über den zu markierenden Bereich bewegen. Dadurch wird ein transparentes Rechteck aufgespannt, z.B.:



(Screenshot aus dem 'graphischen Editor' mit mehreren per Maus selektierten Objekten; linke Maustaste noch gedrückt)

Beim Loslassen der Maustaste verschwindet das aufgespannte Markierungs-Rechteck, alle davon berührten Objekte bleiben aber selektiert (erkennbar an den roten Rahmen).

Um die so markierten Objekt auf dem Bildschirm zu verschieben, greifen Sie *eines dieser Objekte* mit der linken Maustaste (denn andernfalls, beim Klick außerhalb der Selektion werden alle Objekte de-selektiert), und verschieben die Objekte mit gedrückter Maustaste an die neue Position. Sind mehrere Objekte selektiert, beziehen sich auch 'Ausschneiden', 'Kopieren' und 'Einfügen' auf diese Objekte. Damit können z.B. ganze Gruppen von Objekten von einer Anzeigeseite auf andere Seiten übernommen werden.

Mit der Combo-Box in der Toolbar des Editors können die Anzeigzeilen selektiert werden (als Alternative zum Selektieren per Maus in der Zeichenfläche). Diese Funktion ist z.B. dann nötig, wenn ein Anzeigeelement komplett verdeckt ist, und daher nicht direkt mit der Maus selektiert werden kann.

Beim Selektieren eines Objektes in der Zeichenfläche (Pfeil-Werkzeug) wird auch der entsprechende Eintrag in der Combo-Liste umgeschaltet. Die Anzeige in der Combo-Liste ist folgendermaßen zu interpretieren:

[Zeilennummer] ELEMENT-TYP : Element-Text

Beispiel (aus dem Screenshot am Anfang dieses Kapitels) :

[9] TEXT : Supply Voltage

Dies bedeutet:

Das aktuell selektierte Anzeige-Element ist in Zeile Nummer 9 definiert (Array-Index; die Zählung beginnt bei Null);
es handelt sich um eine einfache TEXT-Anzeige (alphanumerisch);
der Anzeigetext beginnt mit der Zeichenkette "Supply Voltage".

Graphische Elemente können mit dem Anzeige-Editor an einem 8*8-Pixel-Gitter ausgerichtet werden. Um dieses Gitter anzuzeigen, oder Koordinaten automatisch am Gitter auszurichten, wählen Sie im Hauptmenü "Optionen"... "Seiten-Editor". Ist die Option "Koordinaten an 8*8-Pixel-Gitter" aktiv, rasten die Koordinaten beim Verschieben eines Objektes automatisch auf den nächstgelegenen Gitterpunkt ein.

Zur Zeit werden die folgenden Element-Typen vom "einfachen" Anzeigeseiten-Editor unterstützt:

- TEXT : einfacher Text, mit oder ohne Anzeige einer Variablen (Sternchen als Platzhalter)
- ICON : kleine Bitmap-Grafik ("Symbol")
- BUTTON : graphische Schaltfläche
- BARGRAPH : Balkendiagramm
- DIAGRAM : Y(t) oder X/Y-Diagramm

Hinweis:

Sonderfunktionen, wie z.B. die in einem anderen Kapitel erwähnten Interpreterkommandos zum Erzeugen von Grafiken, können mit dem "einfachen" Editor nicht erstellt oder bearbeitet werden !

Eine aktuellere Beschreibung (als in dieser PDF-Datei) finden Sie, wie üblich, [online](#) .

7.10.2 Seiten-Definitions-Kopf

Der Seiten-Definitions-Kopf (page definition header) enthält Informationen einer Bildschirmseite. Dazu gehören z.B. die folgenden **Optionen**:

- Seite ständig komplett neu zeichnen

Wenn diese Option aktiviert ist, wird die komplette Bildschirmseite ständig komplett neu gezeichnet. Dies erfordert eine recht hohe CPU-Leistung und führt zu einer recht niedrigen Bild-Aufbau-Rate, kann aber nötig sein falls Sie mit überlappter oder bewegter Grafik arbeiten. Nur ein *kompletter* Neuaufbau des Bildschirms beinhaltet das Löschen des Bildschirms, dadurch wird unter allen Umständen verhindert dass alter „Schrott“ auf dem Display übrigbleibt, wenn sich z.B. ein Icon durch ständiges Neu-Zeichnen mit variablen Koordinaten über den Bildschirm bewegt.

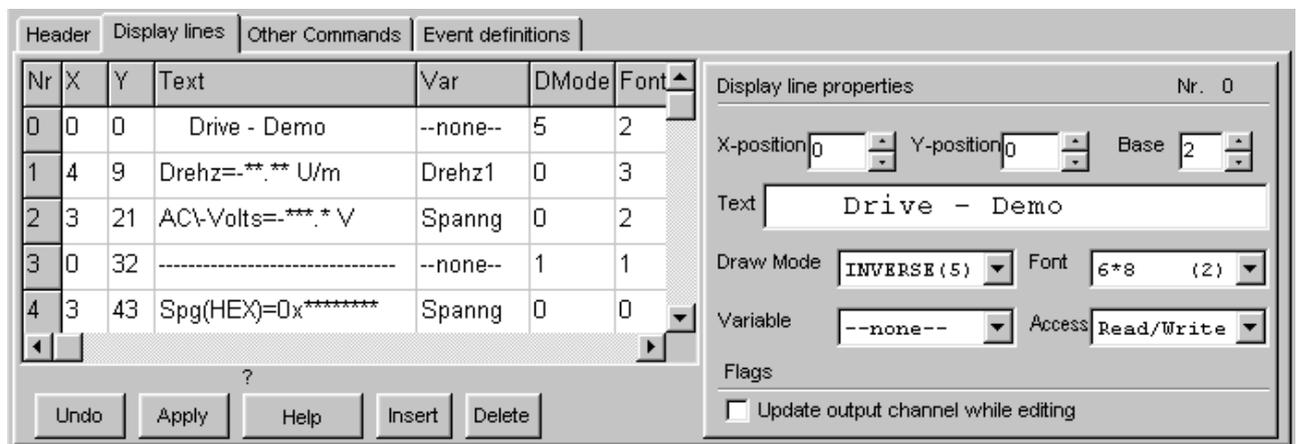
Ein typisches Neuzeichnen des Displays mit der Option „always redraw“ benötigt circa 250ms (beim UPT515 mit einer 8051-kompatiblen CPU und 128*64-Pixel-Display). Daraus ergibt sich eine Bild-Aufbau-Rate von etwa 4 Bildschirmen pro Sekunde (dies hängt von der Komplexität des angezeigten Bilds ab).

Ohne die Option „..ständig neu zeichnen“ werden nur die Elemente neu gezeichnet, deren anzuzeigender Wert (Variablen) sich seit der letzten Aktualisierung geändert hat. Dadurch wird etwas CPU-Rechenzeit eingespart, und das interpretierte UPT-Programm reagiert etwas schneller auf Benutzereingaben.

Details (online): http://www.mkt-sys.de/MKT-CD/upt/help/progt_49.htm#page_def_header

7.10.3 Tabellarische Definition von Anzeige-Zeilen

Um Anzeigezeilen erstellen, die später auf dem Display angezeigt werden sollen, dient (neben dem "einfachen" Seiten-Editor) die Registerkarte „Display Lines“ :



In der linken Tabelle werden alle derzeit definierten Anzeigezeilen der aktuellen Bildschirmseite angezeigt.

Das „property“-Panel rechts zeigt detaillierte Informationen über eine einzelne Anzeigezeile (die durch Mausklick in die Tabelle ausgewählt werden kann).

Sie können die Eigenschaften (properties) einer Anzeigezeile sowohl durch direkte Eingabe in der Tabelle (links) oder auch auf dem „property“-Panel (rechts) eingeben. Die Information wird während der Eingabe sofort in beiden Teilen aktualisiert.

Falls das (simulierte) LC-Display sichtbar ist, können Sie den Effekt Ihrer Eingabe sofort dort sehen. Wenn Sie z.B. die Y-Position eines Anzeigetextes mit dem Up/Down-Scroller auf dem „Property“-Panel ändern, sehen Sie sofort wie sich der Text auf dem simulierten LC-Display mitbewegt.

Falls sich einige graphische Elemente oder/und Textzeilen auf dem Display überlappen, hängt das Resultat von der Reihenfolge des Zeichnens (drawing order) ab. Mehr dazu in Kapitel 7.10.7. Die Reihenfolge der Definitionszeilen kann per Maus geändert werden. Dazu bei gedrückter linker Taste einen Eintrag in der Spalte "Nr" nach oben oder unten bewegen. Ein schwarzer horizontaler Markierungsbalken zeigt die Einfügeposition. Diese Tabellen-Zeilen-Verschiebung funktioniert auch in anderen Tabellen des Programmierwerkzeugs.

Der Button „Insert“ dient zum Einfügen neuer Display-Zeilen mit Texten oder Icons (an der Position, die in der Tabelle markiert ist).

Der Button „Delete“ entfernt die in der Tabelle markierte Display-Zeile.

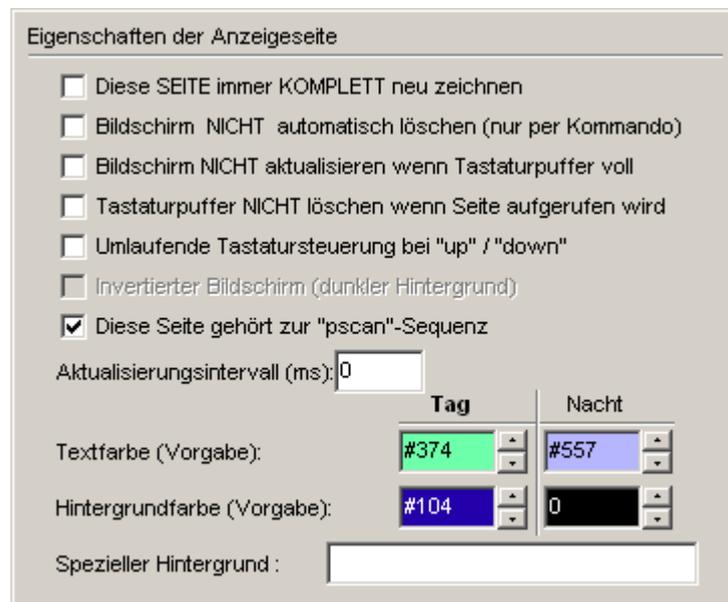
Einige Sonderfunktionen zum Editieren der Anzeige-Zeilen können durch ein Popup-Menü aktiviert werden, indem mit der rechten Maustaste in die Tabelle geklickt wird.

Details (online): http://www.mkt-sys.de/MKT-CD/upt/help/progt_49.htm#page_def_disp_lines

7.10.4 Farben

Für Geräte mit Farbdisplay (z.B. MKT-View II) können bei der Definition von Anzeige-Elementen (z.B. Texten, Buttons, Bargraphen) individuelle Farben definiert werden.

Es empfiehlt sich aber, die Vorder- und Hintergrundfarben nur auf der Registerkarte "Definitions-Kopf" (bzw "Display Page Header") zu definieren, und die Farben in den einzelnen Anzeigezeilen (d.h. Farben für individuelle Anzeigeelemente auf "-1" zu setzen. Dies bedeutet, dass für die Vorder- und Hintergrundfarbe die für die komplette Anzeigeseite geltenden Vorgabe-Werte (defaults) verwendet werden. Dadurch wird die im [ONLINE-HILFE-SYSTEM](#) beschriebene Tag/Nacht-Umschaltung sehr vereinfacht.



Eigenschaften der Anzeigeseite

- Diese SEITE immer KOMPLETT neu zeichnen
- Bildschirm NICHT automatisch löschen (nur per Kommando)
- Bildschirm NICHT aktualisieren wenn Tastaturpuffer voll
- Tastaturpuffer NICHT löschen wenn Seite aufgerufen wird
- Umlaufende Tastatursteuerung bei "up" / "down"
- Invertierter Bildschirm (dunkler Hintergrund)
- Diese Seite gehört zur "pscan"-Sequenz

Aktualisierungsintervall (ms):

	Tag	Nacht
Textfarbe (Vorgabe):	#374	#557
Hintergrundfarbe (Vorgabe):	#104	0

Spezieller Hintergrund :

Details zur Verwendung von Farben (online):

http://www.mkt-sys.de/MKT-CD/upt/help/colours_49.htm

7.10.5 Der Format-String in Anzeigezeilen

Der „format string“ ist der wichtigste Teil einer Anzeige-Zeilen-Definition.

Ein „Format-String“ kann...

- ein einfacher Anzeigetext sein (wie z.B. „Hauptmenue“)
- besondere „Platzhalter“ enthalten wie z.B. „****“ für numerische Ziffern
- benutzt werden um kleine Bilder anzuzeigen (in Backslash-Sequenzen wie „\iMyIcon“)

Spezielle Zeichen in einem Format-String sind:

- *
wird durch numerische Ziffer oder Zeichen eines Stringausdrucks ersetzt
- wird durch das Vorzeichen eines numerischen Wertes ersetzt (+,-)
- \\
dient zur Anzeige eines *einzelnen* Backslash-Zeichens
- \-
zeigt einen Bindestrich (bzw. immer „Minus“-Zeichen) an
- \cN
setzt die Textfarbe auf (N), nur für Farbdisplays
- \CN
setzt die Hintergrundfarbe auf (N), nur für Farbdisplays
- \mN
setzt den Zeichenmodus (normal, invers, blinkend, etc)
- \i<Icon-Name>[, <Invert-Flag>]
fügt ein Icon an der aktuellen Zeichenposition ein (siehe Unten)

Einige Format-String-Beispiele:

```
Voltage= *****.** V
Icons: \iF_ok,kd0 \iF_no \iF_stop
DrawModes: \m0 Normal \m5 Invers \m8 Blink
```

Einfügen von Icons in einer **Anzeigezeile**:

Siehe zweites Beispiel. Hier wurden drei Icons (namens „F_ok“, „F_no“ und „F_stop“) in einer einzelnen Anzeigezeile eingefügt. Jedes Icon wird bei der Anzeige wie ein „großer Buchstabe“ behandelt. Dies hat zur Folge, dass die Änderung der X,Y-Position der Anzeigezeile auch die drei Icons mitbewegt.

Sie können auch einen optionalen „Zustands“-Ausdruck in die Icon-Backslash-Sequenz aufnehmen, wenn Sie das Icon in Abhängigkeit von irgendwelchen Betriebszuständen invertieren wollen (als numerischer Ausdruck nach dem Icon-Namen, getrennt durch Komma). Diese Funktion soll die Implementierung von „grafischen Funktionstasten“ in Ihrer Anwendung vereinfachen. Positionieren Sie die Funktionstasten direkt oberhalb der Tasten F1..F4. Im oben gezeigten Beispiel wird das Icon „F_ok“ invertiert sobald die erste Funktionstaste gedrückt wird (weil der Ausdruck „kd0“ den Wert TRUE hat, solange F1 gedrückt ist).

Hinweise:

- Falls das Icon in einer „\i<Icon-Name>“-Sequenz unbekannt ist (weil nicht auf der „Icon“-Seite definiert), wird die Backslash-Sequenz wie normaler Text angezeigt.
- Wenn Sie in einer Anzeigezeile Icons in Abhängigkeit von numerischen Ausdrücken invertieren, sollten Sie die Option „always redraw this page completely“ im Seiten-Definitions-Kopf (header) aktivieren. Grund: Der Interpreter könnte andernfalls nicht erkennen, wann sich das Ergebnis des „Invertierungs-Ausdrucks“ ändert, denn der Ausdruck wird nur beim Neu-Zeichnen der Zeile berechnet.
- Dies gilt auch, wenn Sie in einer Backslash-Sequenz zum Umschalten des Zeichen-Modus („\m<Modus>“) für „Modus“ keine feste Zahl, sondern einen numerischen Ausdruck verwenden.
- Wenn Sie mehrere Icons in einer einzigen Anzeigezeile voneinander „trennen“ wollen, verwenden sie das Space-Zeichen (Leerzeichen). Die Breite eines Leerzeichens hängt vom Zeichensatz der Zeile ab. Für eine kleine Schrittweite der Leerzeichen verwenden Sie z.B. den 4*6-Pixel-Zeichensatz.
- Falls Sie ein einzelnes Icon mit variabler Koordinate zeichnen wollen, verwenden Sie den Grafikbefehl „Icon“ (Kap.Fehler: Verweis nicht gefunden) statt einer Icon-Backslash-Sequenz im Format-String. Damit lassen sich (in bescheidenem Rahmen) auch bewegte Grafiken realisieren.
- Bei neueren Geräten (mit Farbdisplays und 32-Bit-CPU) stehen noch wesentlich mehr Funktionen zur Verfügung, deren Beschreibung den Rahmen dieses 'druckbaren' Dokumentes sprengen würde. Details finden Sie, wie bereits bekannt, im Hilfesystem. Online-Link :

http://www.mkt-sys.de/MKT-CD/upt/help/progt_49.htm#format_string

7.10.6 Verwendung von Displayzeilen als Menüs

Sie können normale Anzeigezeilen auch als Menüzeilen verwenden, um ein nicht scrollendes (einseitiges) Menü zu realisieren.

Zu diesem Zweck setzen Sie das Flag „**This line is a menu item**“ auf dem Panel „display line properties“. Dadurch wird aus einer normalen *Anzeigezeile* eine *Menüzeile*, die vom Bediener z.B. per CURSOR-Tastensteuerung oder Drehknopf angewählt werden kann.

Danach müssen Sie noch definieren, was das UPT-Programm tun soll wenn der Bediener in Ihrem Menü (auf dem neuen Menüpunkt) die ENTER-Taste drückt. In den meisten Fällen wird dies ein goto-Befehl zu einer anderen Display-Seite sein. Beispiel:

g"Menu2" (vorausgesetzt, Ihr Programm hat eine Seite namens „Menu2“)

Dieses UPT-Interpreter-Kommando muß in der Spalte „Var/Expression“ der Display-Zeilen-Definitionstabelle eingetragen werden.

Es existieren einige UPT-Interpreter-*Funktionen*, die Ihnen eine erweiterte „programmgesteuerte“ Kontrolle dieser Menüs erlauben:

- **mi** : liefert die aktuelle Zeilennummer (index, 0..n) des aktuellen Menüpunktes.
- **mm**: liefert den aktuellen Menü-Modus. Dies kann einer der folgenden Werte sein:
 0 = Menü ist „Aus“, d.h. der Menü-Selektions-Balken ist nicht sichtbar.
 1 = Menü-Modus „Anwählen“, d.h. der Menü-Selektions-Balken ist sichtbar, der Bediener kann den „aktuellen“ Menüpunkt per CURSOR-Tasten verschieben.
 Andere „modi“ sind nur bei numerischen Eingabefeldern gültig, z.B. 2 = „Wert wird editiert“.

Analog zu den Abfrage-*Funktionen* existieren einige Einstell-*Prozeduren* für die oben beschriebene Art von Menüs. Diese haben die gleichen Namen wie die entsprechenden Funktionen:

- **mi**(<new_item>) dient zum programmgesteuerten Setzen des „aktuellen“ Menüpunktes
- **mm**(<new_mode>) dient zum programmgesteuerten Setzen des Menü-Modus

Ein Beispiel zu dieser Art von Menüs finden Sie in der Datei ..\programs\NEWMENUS.UPT .

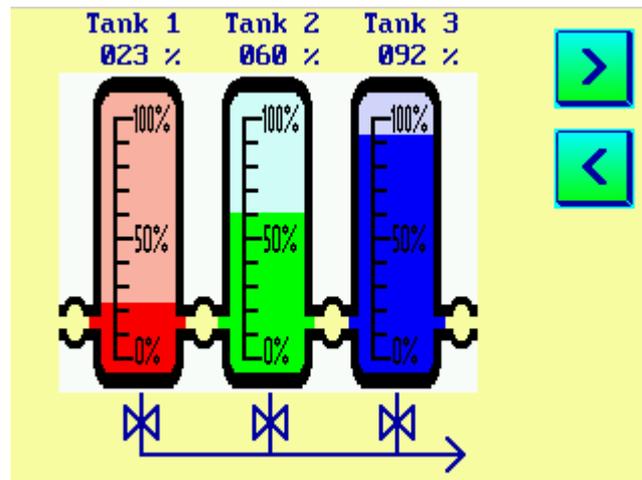
Seit März 2001 ist es auch möglich, den so realisierten Menüzeilen eigene „Hotkeys“ zuzuweisen. Näheres dazu finden Sie im Hilfesystem des Programmiertools.

Hinweis: Wenn Sie eine Display-Zeile als „Menüpunkt“ verwenden, können Sie in dieser Zeile keinen numerischen Wert abbilden !

7.10.7 Überlappende Grafiken

Für bestimmte Spezialeffekte auf dem Display kann es nötig sein, einige Anzeigeelemente (Text oder Grafik) **vor** oder **hinter** anderen Elementen zu platzieren bzw. sich nur teilweise überlappen zu lassen.

Im folgenden Beispiel bestehen die drei "Farbtanks" aus jeweils einer transparenten Bitmap. Der Füllstand wurde durch vertikale Balkendiagramme (Bargraphen) ohne Skala realisiert:



Damit dieser Effekt (Bitmap 'vor' dem Balkendiagrammen) funktioniert, müssen beim Aktualisieren der Seite zunächst die farbigen Balkendiagramme, und danach die Bitmaps gezeichnet werden. Andernfalls würden die Balkendiagramme die Bitmaps überdecken (denn Balkendiagramme unterstützen im Gegensatz zu Bitmaps keine Transparenz).

Die Reihenfolge beim Zeichnen wird durch die Zeilennummer in der Anzeigeseiten-Definitionstabelle festgelegt:

Das Element mit der Zeilennummer „0“ wird als erstes gezeichnet, danach das Element mit der Zeilennummer „1“ und so weiter.

Falls Sie mit überlappender Darstellung arbeiten, sollten Sie die Option „always redraw page completely“ im Seiten-Definition-Kopf aktivieren.

Weitere Details sind Sie (wie immer) im Hilfesystem, online verfügbar unter:

http://www.mkt-sys.de/MKT-CD/upt/help/progt_49.htm#overlapping_graphics

7.11 Spezielle Anzeige-Kommandos

Die folgenden speziellen Anzeige-Kommandos des UPT-Interpreters dienen zum Zeichnen verschiedener Grafik-Elemente auf dem LC-Display:

icon (ic)	zeichnet ein kleines Bild („icon“)
line (li)	zeichnet eine Linie
pixel (pi)	setzt ein einzelnes Pixel
frame(fr)	zeichnet einen rechteckigen Rahmen
fill_rect(fi)	zeichnet ein verschiedenartig ausgefülltes Rechteck

Um das Erscheinungsbild einiger graphischer Elemente zu beeinflussen, dient das folgende Kommando:

draw mode(dm) setzt den Zeichenmodus

Auch bei den speziellen Anzeige-Kommandos gelten die in Kapitel 7.10.7 gemachten Anmerkungen zur *Reihenfolge* des Zeichnens, falls sich die Grafiken überlappen.

Eine detaillierte Beschreibung aller Interpreter-Kommandos finden sie **nur** im Windows-Hilfe-System des UPT-Programmiertools; hier wieder der Link zur Online-Dokumentation:

http://www.mkt-sys.de/MKT-CD/upt/help/progt_49.htm#display_commands

7.12 Definition von Ereignissen (Übersicht)

Mit *Ereignis* (event) ist hier das Auftreten einer bestimmten Situation gemeint, die Sie in Ihrer UPT-Anwendung irgendwie verarbeiten wollen.

Ein *Ereignis* kann z.B. sein:

- Der Bediener drückt eine bestimmte Taste des Terminals (oder lässt eine Taste los)
- Der Wert einer Variablen (die z.B. via CAN empfangen wurde) überschreitet einen bestimmten Wert
- Ein Bit eines digitalen Eingangs nimmt einen bestimmten Zustand an
- und viele Kombinationen (die als boolescher Ausdruck formuliert werden)

Zu jedem *Ereignis* definieren Sie auch eine passende *Reaktion*.

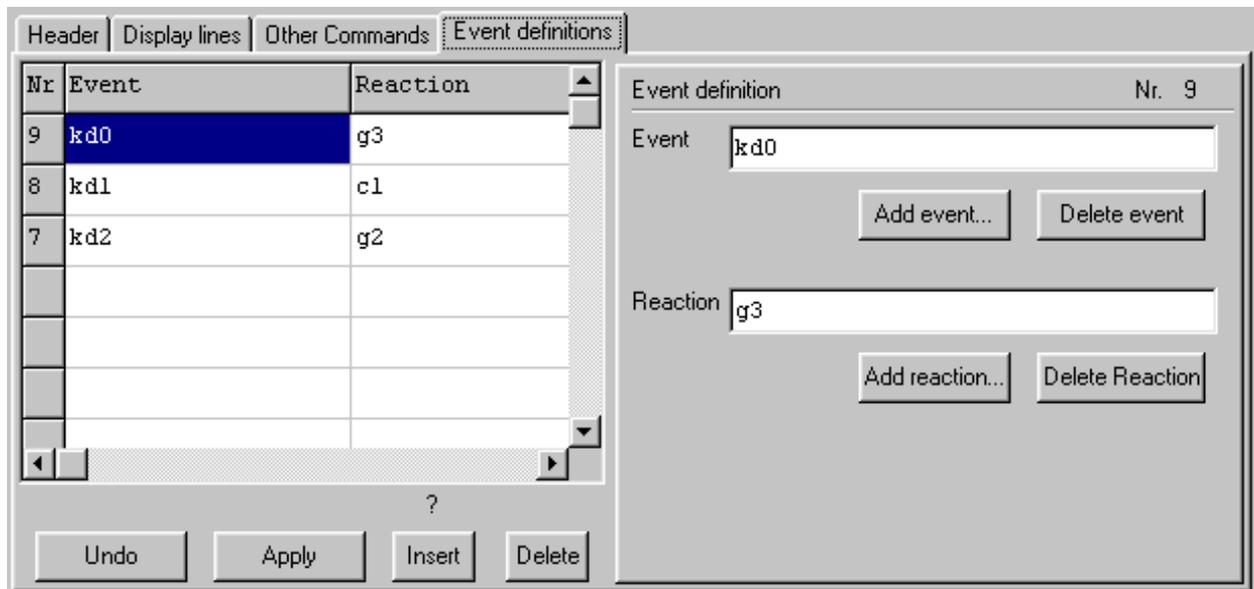
Mögliche *Reaktionen* sind zum Beispiel:

- Umschalten zu einer bestimmten Display-Seite
- Setzen einer Variable auf einen bestimmten Wert (der dann möglicherweise per CAN abgesendet wird)

Zur Definition von *Ereignissen* und *Reaktionen* wird im UPT eine einfache Programmiersprache verwendet, Sie können einfache Ereignisse und Reaktionen aber auch aus einer Auswahlliste des UPT-Programmiertools wählen (die dann automatisch in den entsprechenden Interpreter-Code umgesetzt wird).

7.13 Die Eingabe von Event-Definitionen mit dem Programmierool

Um *Ereignisse* und *Reaktionen* zu definieren, dient die folgende Registerkarte des UPT-Programmierools:



In diesem Beispiel sehen Sie drei Abfragen für Tastatur-*Ereignisse*, die jeweils zum Umschalten der aktuellen Anzeigeseite führen.

7.13.1 Befehle zur Definition von Ereignissen

Um verschiedene Ereignisse abzufragen, dienen (u.A.) die folgenden Funktionen des UPT-Interpreters:

Ereignis-Typ	Parameter (wenn nötig)	Description
kd	key-name	Taste wurde gedrückt (key down)
ku	key-name	Taste wurde losgelassen (key up)
kb	key-name	Taste im Puffer (key buffered)
kh		Irgendeine Taste gedrückt (key hit)
kc		liest und entfernt Taste aus System-Tasten-Puffer (key code)
cb		CAN-Fehler: Bus Off
ce		CAN-Fehler (schwerwiegend, allgemein)
cw		CAN-Warnung
ch		CAN-Hardware-Fehler
co		CAN-Fehler: Überlauf
ct		CAN-Fehler beim Senden (transmit)
t0 ... t3		User-Timer (Nr. 0...3) abgelaufen
pe		aktuelle Seite grade neu aufgerufen (Page Enter)
pq		aktuelle Seite wird gleich verlassen (Page Quit)

(Hinweis: Darüberhinaus existieren weitere Event-Funktionen, die nur im Hilfe-System des UPT-Programmiertools beschrieben sind !)

Der UPT-Interpreter wertet alle Event-Definitionen als Boolesche Ausdrücke aus. Diese Ausdrücke können auch komplizierte numerische Berechnungen und Vergleiche enthalten.

Wenn das Ergebnis einer Event-Definition (bzw. der Berechnung des *Ausdrucks*) nicht NULL ergibt, dann führt der UPT-Interpreter auch die entsprechende Ereignis-Reaktions-Methode aus. Eine Reaktionsmethode besteht i.A. aus einer Reihe von Interpreter-Kommandos, im einfachsten Fall aus einem *goto*-Befehl der zu einer anderen Anzeige-Seite umschaltet.

Eine genauere (und aktuellere!) Beschreibung der Programmiersprache für Events findet sich im Hilfe-System des Programmiertools.

Eine kurze Beschreibung von numerischen Ausdrücken findet sich in Kapitel 8.1.1 dieses Dokuments.

7.14 Globale Ereignisse (Global Events)

Sie können auch Ereignisse definieren, die unabhängig von der aktuellen Display-Seite des UPTs verarbeitet werden sollen.

Beispiel: Sie wollen –unabhängig von der aktuellen Display-Seite- eine bestimmte „Fehlermeldungs-Seite“ aufrufen, falls ein schwerer Fehler auf dem CAN-Bus auftritt. Dies könnten Sie folgendermaßen programmieren:

```
Event-Def.: ce (bedeutet: "bei einem CAN-Fehler".....)
Reaction:   c"CanError"    (... rufe die Seite "CanError" auf)
```

Definieren Sie dieses Event und die Reaktionsmethode auf der Registerkarte „Global Events“, damit Sie das Ereignis „CAN-Fehler“ nicht nur auf bestimmten Display-Seiten abfangen sondern immer (Bedenken Sie: Die Anzahl von Kommandozeilen auf einer Display-Seite ist begrenzt!).

Es gibt einige Regeln, die Sie bei der Programmierung von „globalen Ereignissen“ beachten sollten:

- Verwenden Sie „global events“ nur für Ereignisse, die wirklich unabhängig von der aktuellen Bildschirmseite ausgewertet werden müssen.
- Verwenden Sie in den Reaktionsmethode von globalen Ereignissen keine Aufrufe von Display-Ausgabe-Funktionen, weil dies die normale Bildschirmausgabe aus Ihrer Anzeige-Seiten-Definitionen verursachen könnte.
- Falls Sie ein bestimmtes Ereignis *global* verarbeiten (wie „ce“ im obigen Beispiel), sollten Sie dieses Ereignis nicht auch noch als *lokales* Ereignis auf irgendeiner Bildschirmseite abfangen. Dies schadet dem „Event-Handler“ in der UPT-Firmware zwar nicht, führt aber zu undurchschaubaren Programmen. Wenn mehrere Events gleichzeitig eintreffen, werden die entsprechenden Reaktionsmethoden nacheinander abgearbeitet. Dies führt u.U. zu Verzögerungen beim Bildschirm-Update.

Bei neueren Geräten mit 32-Bit-CPU wie z.B. MKT-View II / III / IV wird dringend empfohlen, statt der oben beschriebenen 'Globalen Events' die Script-Sprache zu verwenden, da sich so wesentlich übersichtlicher definieren lässt, wie sich das Programm (oder auch die Anzeige) in bestimmten Situationen verhalten soll. Aktuelle Informationen dazu finden Sie auch dazu in der Online-Hilfe:

„Events“: http://www.mkt-sys.de/MKT-CD/upt/help/progt_49.htm#event_defs

Script-Sprache: http://www.mkt-sys.de/MKT-CD/upt/help/scripting_49.htm

7.15 Anzeigeseiten-Übersicht

Die Registerkarte „All Pages“ enthält eine Übersicht über alle Display-Seiten, die im UPT-Anwenderprogramm definiert sind.



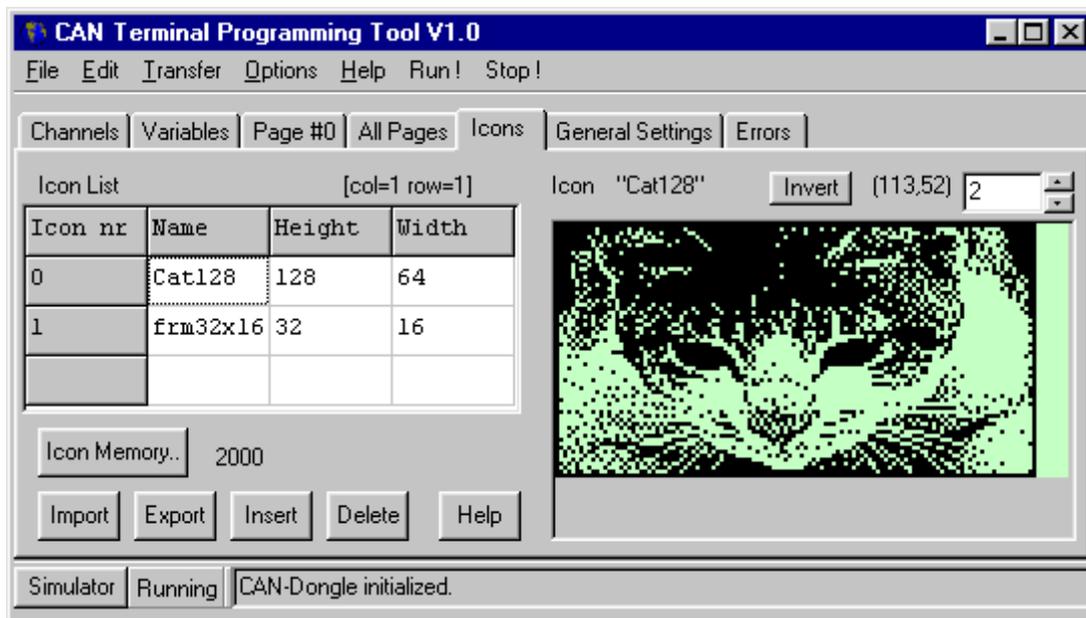
Durch einfaches Anklicken eines Bildes in dieser Übersicht kann zu dieser Seite umgeschaltet werden, die dann in der Anzeige des LCD-Simulators erscheint.

Durch Doppelklick auf ein Bild in der Übersicht wird dies zur aktuellen Seite (wie beim Einfach-Klick), ausserdem wird die Registerkarte des UPT-Programmiertools zur Display-Seiten-Definition umgeschaltet (siehe Kapitel 7.10).

Die jeweils aktuelle Display-Seite wird durch einen farbigen Rahmen markiert. Diese Markierung wird auch umgeschaltet, wenn der UPT-Simulator Ihr Anwenderprogramm abarbeitet und dort eventuell vorhandene *goto*- oder *call*-Anweisungen ausführt. Sollte dies stören, stoppen Sie die Abarbeitung des UPT-Anwenderprogramms (Befehl „Stop!“ im Hauptmenü).

7.16 Import und Verwendung von Bitmap-Grafiken ('Icons')

Die Registerkarte „Icons“ dient zum Importieren und Editieren von kleinen Bildern, die Sie in das UPT-Anwenderprogramm einbauen können.



Icons können aus monochromen Windows-Bitmap-Dateien importiert werden.

Falls das zu programmierende Gerät Farb-Bitmaps unterstützt (wie z.B. das „UPT167 Color“), können Sie mit dem Programmiertool auch kleine Farbbilder importieren. In diesem Fall müssen Sie allerdings die Funktion „Import color bitmap“ aus dem Hauptmenü verwenden, nicht den „Import“-Button auf der Icon-Definitionsseite. Weitere Informationen über Farbgrafiken im UPT finden Sie im Hilfesystem des Programmiertools.

Um Icons in das UPT-Programm einbinden zu können, muss zunächst genug Speicher dafür freigemacht werden. Dies erfolgt z.B. durch Reduzieren der Anzahl von Display-Seiten. Der Speicher, der durch das Verringern der Anzahl von Anzeigeseiten frei wird, dient dann zur Aufnahme von Icons (oder anderen Daten im UPT). Das Reservieren von Speicher für Icons erfolgt mit einem speziellen Dialogfenster, welches im folgenden Kapitel vorgestellt wird.

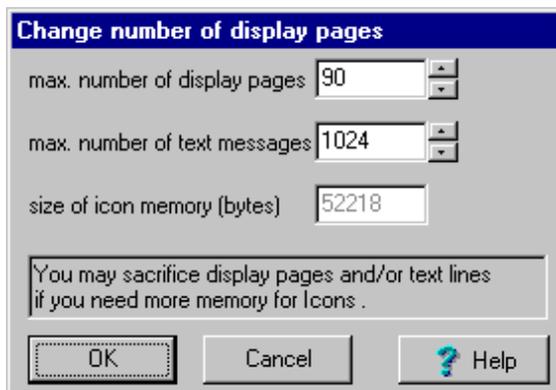
7.16.1 Reservierung von Speicher für Icons und Sonderfunktionen

Icons und Display-Seiten-Definitionen müssen sich den gleichen Speicher im „echten“ UPT teilen. Falls Sie in Ihrer Anwendung Icons (oder andere Sonderfunktionen) nutzen wollen, müssen Sie einige der normalerweise verfügbaren Display-Seiten opfern.

Sobald Sie die „maximale Anzahl Display-Seiten“ in Ihrer UPT-Anwendung verringern, erhalten Sie mehr Speicher zur Aufnahme von Icons.

Beispiel: Ein bestimmtes UPT-Terminal hat Speicher für 50 Display-Seiten, Sie wollen aber nur 40 Seiten davon verwenden. Es bleiben dann circa $(50-40) * 1600$ Bytes im FLASH-Speicher des UPTs frei, den Sie für Icons nutzen können.

Seit April 2000 muss vom Display-Seiten-Speicher auch ein Teil für *Text-Strings* abgezogen werden (siehe Beschreibung des „Text-Arrays“ in Kapitel 7.18).



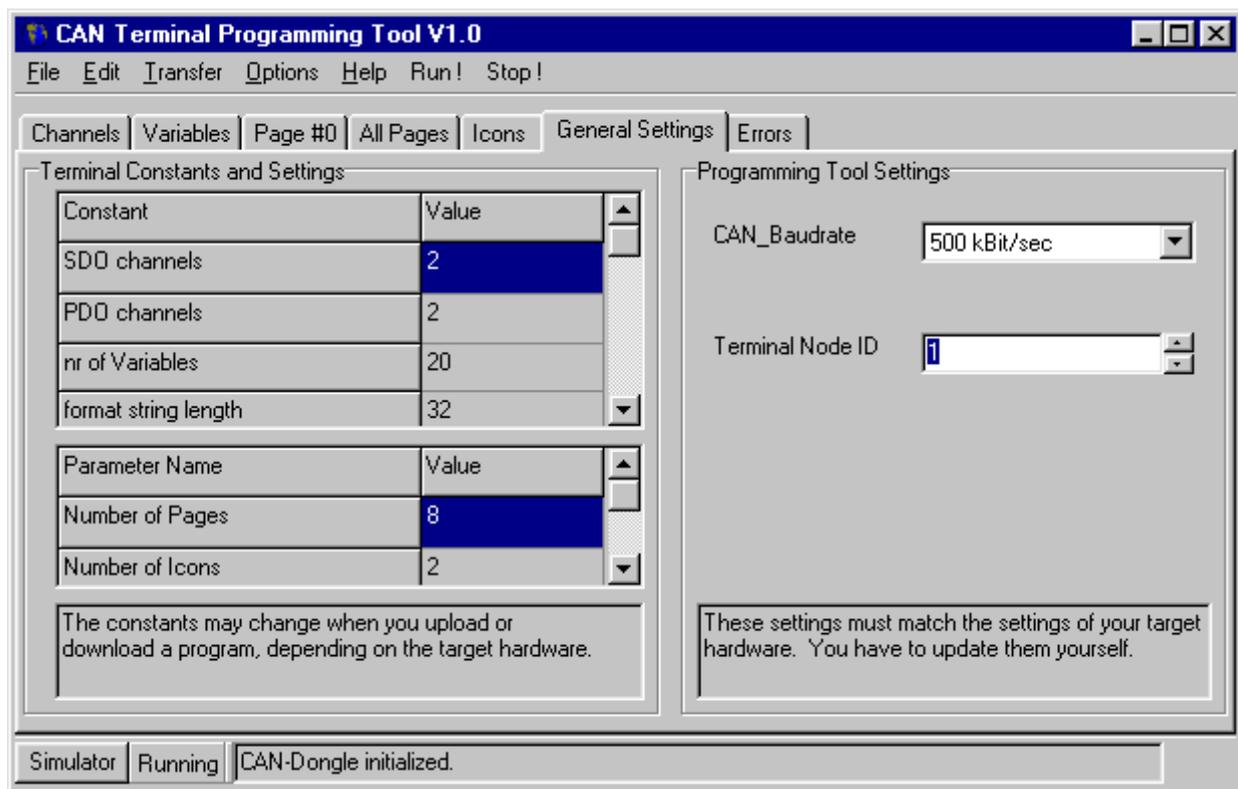
Klicken Sie auf den „IconMemory“-Button (auf der Icon-Seite) um diesen Dialog zum Reservieren von Icon-Speicher zu starten.

Hinweis: Die Größe des Icon-Speichers („size of icon memory“) die in der Dialogbox angezeigt wird, bezieht sich auf den verfügbaren FLASH-Speicher der für Icons verwendet werden *könnte*. Sollten bei der Darstellung einer Display-Seite mit vielen (großen) Icons einige Icons fehlen, könnte das an der RAM-Knappheit im „echten“ UPT liegen. Das UPT-Programmierool hat keine Informationen über die RAM-Verwaltung im „echten“ UPT wenn eine bestimmte Display-Seite angezeigt wird. In der aktuellen Firmware-Version hat das UPT keine Möglichkeit, Objekte dynamisch aus dem RAM zu entfernen falls das RAM knapp wird. Trotz ausreichendem ROM könnte daher die Anzeige von „zu vielen Icons gleichzeitig“ fehlschlagen !

7.17 Grundeinstellungen (General Settings)

Die Registerkarte „General Settings“ dient zum Einstellen, z.T. auch nur zur Anzeige einiger Eigenschaften des Terminals, die nichts mit Ihrer UPT-Anwendungssoftware zu tun haben. Dazu gehören:

- CAN – Baudrate, um den Kontakt zwischen Programmierwerkzeug und UPT herzustellen
- „Terminal node ID“ des UPTs, der im Normalfall auf „1“ steht

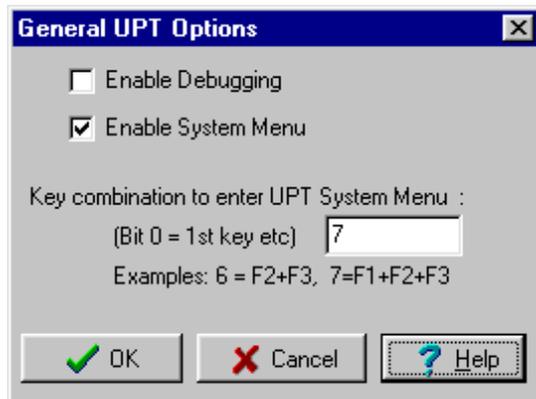


Die Einträge hinter *Terminal Node ID* und *CAN-Baudrate* muß mit den Einstellungen im Terminal übereinstimmen.

Beim Terminal „UPT515“ müssen Sie diese Einstellungen im fest eingebauten *Setup-Menü* durchführen, welches Sie (bei der Firmware vom 29.05.2000) nach gleichzeitigem Drücken von F2 und F3 erreichen können. Per Cursor-Up/Down müssen Sie die Menüpunkte „CAN-Baudrate“ und „Modul/NodeID“ im UPT515 auslesen und in die oben gezeigte Registerkarte übertragen bzw. an die Einstellungen des Programmierwerkzeugs anpassen. Nach dem Ändern der Einstellungen im UPT515 müssen Sie die geänderte Konfiguration mit dem Menüpunkt „Save & Exit“ dauerhaft abspeichern !

7.17.1 Allgemeine UPT-Optionen

Der Dialog „General UPT Options“ dient zum Einstellen einiger Optionen, die später auch in das UPT übertragen werden (beim Programm-Upload). Dieser Dialog kann aus dem Hauptmenü oder durch Anklicken der entsprechenden Zeile in der „General Settings“ – Tabelle gestartet werden.



Die Option **Enable System Menu** kann abgeschaltet werden, falls Sie das im UPT fest eingebaute „System-Menue“ nicht benötigen (weil sie z.B. die Tastenkombination F2+F3 selbst benötigen oder um dem Bediener den Zugang zum System-Menü zu verwehren).

Wenn „Enable System Menu“ abgeschaltet ist, kann das System-Menü im UPT nur noch durch Drücken von F2+F3 bei power-on aktiviert werden.

Die Option **Enable Debugging** ist nur für den Programmierer der UPT-Firmware vorgesehen (weil der keinen Target-Debugger hatte). Sie sollten diese Option abgeschaltet lassen.

Die Option **key combination to enter the UPT's system menu** dient seit Februar 2001 zum Ändern der Tastenkombination zum Aufruf des UPT-Systemmenues, weil die früher verwendete Kombination F2+F2 mit der Applikation eines Anwenders kollidierte.

Bitte Vorsicht beim Ändern dieser Kombination, denn nicht alle Tastaturen erlauben die Abfrage beliebiger gleichzeitig gedrückter Tastenkombinationen ! Falls Sie eine ‚unmögliche‘ Kombination eingeben, können Sie nur noch per F2+F3 *direkt nach Power-ON* in das Systemmenü gelangen !

7.18 Das Text-Array

Das Text-Array ist ein Speicher für Text-Zeilen, die Sie in Ihrer Anwendung einsetzen können um *Meldungstexte* anzuzeigen, die durch numerische Werte gesteuert werden (z.B. zur Umsetzung von „Fehlercodes“ eines Sensors in „Klartexte“ für das Display).

Um das Text-Array nutzen zu können, müssen Sie leider einige Display-Seiten opfern. Dies funktioniert ähnlich wie beim Icon-Speicher, siehe Dialogbox in Kapitel 7.16.1.

Um auf einzelne Zeilen des Text-Arrays in Ihrem UPT-Programm zuzugreifen, verwenden Sie einige String-Funktionen in Text-Anzeige-Kommandos (siehe Hilfesystem zu „sa[]“ und „sr[]“).

Ein einfaches Beispiel zum Zugriff auf Text-Arrays finden Sie im Beispielprogramm „demo1.upt“ (oder später).

7.19 Die Fehler-Anzeige des Programmiertools

Die Registerkarte „Errors“ des UPT-Programmiertools dient zur Anzeige von Fehlern, die bei der Programmübertragung oder beim Abarbeiten des UPT-Programms mit dem Simulator auftreten können.

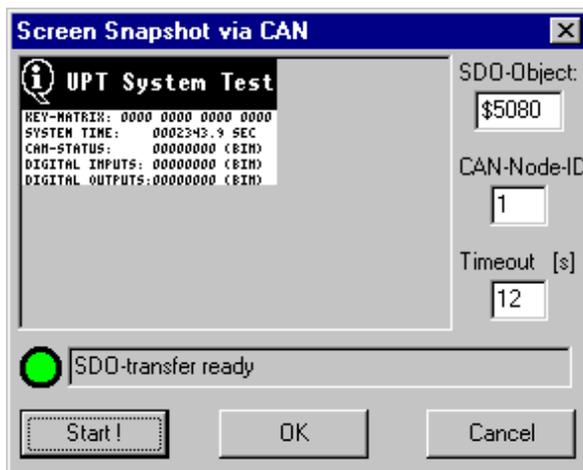
Falls Fehler bei der Programmübertragung auftreten, könnten in den Fehlermeldungen einige hexadezimal codierte SDO-Fehlercodes auftauchen. Eine Tabelle mit den häufigsten SDO-Fehlercodes (nach CANopen) finden Sie in Kapitel Fehler: Verweis nicht gefunden dieses Dokuments.

7.20 Bildschirmfotos vom Terminal per CAN-Bus

(nur bei Geräten mit CANopen verfügbar !)

Um für die Anwender Ihres UPT-Programms eine Dokumentation zu erstellen, können Sie „Bildschirmfotos“ vom LCD eines *echten* Terminals machen. Sie benötigen dazu keine Kamera, sondern nur das Terminal mit CAN-Bus-Verbindung zum PC. Damit können Sie zu beliebigen Zeiten „Schnappschüsse“ des LC-Displays anfertigen.

Starten Sie dazu den „Snapshot“-Dialog im „Transfer“-Menu des Programmierwerkzeugs.



Die Parameter am rechten Rand sind für die Kommunikation zwischen Tool und Terminal. Sie brauchen diese normalerweise nicht zu ändern (nur falls Sie Bildschirmfotos von Terminals machen wollen, die **kein** UPT sind).

Klicken Sie auf den „Start!“-Button um den aktuellen Inhalt des LC-Displays des angeschlossenen Terminals in einen Puffer zu übertragen, der dann vom PC-Programm ausgelesen wird. Die Übertragung des Pufferinhalts erfolgt (Pixel für Pixel) im Hintergrund, ohne das Terminal-Programm weiter zu stören. Bei hoher Auslastung der CPU im Terminal kann dies eine ganze Weile dauern.

Nach Abschluß der Übertragung (wenn die „LED“ hellgrün ist), klicken Sie auf „OK“ um das empfangene Bild als Bitmap auf der Festplatte zu speichern, oder auf „CANCEL“ um das Bild zu verwerfen (=Dialog schließen ohne das Bild abzuspeichern).

Der „Screen-Snapshot via CAN“ funktioniert leider (noch) nicht bei Farbterminals, da dort nicht genug Speicher zum Puffern einer kompletten Bildschirmseite zur Verfügung steht.

Bei Geräten mit 32-Bit-CPU und Ethernet-Schnittstelle besteht evtl. die Möglichkeit, den Inhalt des Grafikbildschirms mit dem eingebauten Web-Server auszulesen. Geben Sie dazu im Web-Browser die IP-Adresse des Terminals ein (diese wird beim Aufruf des Systemmenüs angezeigt, falls das Gerät über TCP/IP mit integriertem Web-Server verfügt; z.B. 192.168.0.100:0080 . Port 0080 ist der HTTP-Server).

8. Der Display-Interpreter im UPT (Sprachbeschreibung)

Hinweis: Der Display-Interpreter steht in allen programmierbaren Anzeige-Terminals von MKT zur Verfügung. Er ist nicht mit der wesentlich flexibleren *Script-Sprache* zu verwechseln - näheres zur Script-Sprache in der Online-Hilfe :

http://www.mkt-sys.de/MKT-CD/upt/help/scripting_49.htm

8.1 Allgemeine Syntax

8.1.1 Numerische Ausdrücke

Die Betriebs-Firmware des UPTs enthält –wie auch der im Programmiertool eingebaute UPT-Simulator- einen Interpreter für numerische Ausdrücke. Diese Ausdrücke dienen unter anderem für:

- komplexe Ereignis-Definitionen, z.B. Vergleiche
- Berechnung von Werten die an Variablen zugewiesen werden.
- Flexible Parameter, die beim Aufruf verschiedener System-Kommandos berechnet werden

Einige Beispiele für numerische Ausdrücke:

```
(Voltage * 100 ) / 230  
(Voltage > 230)      ( ein Vergleich, z.B. für Events )  
(ce & kd0)          ( CAN-Error UND (F1 gedrückt) )
```

Numerische Ausdrücke setzen sich aus den folgenden Syntax-Elementen zusammen:

- Zahlen
- Verknüpfungsoperatoren
- Variablen
- Funktionsaufrufen

8.1.1.1 Zahlen

Der Interpreter für numerische Ausdrücke verarbeitet Zahlen in den folgenden Formaten:

- dezimal mit optionalem „#“-Präfix
- hexadezimal mit dem Präfix „0x“ oder „\$“-prefix („C“ oder Pascal – Stil)
- binär mit dem Präfix „%“

Jede Zahl kann auch über ein Vorzeichen (+ oder -) verfügen, welches den Präfix vorangestellt sein muß.

Die Ziffern von hexadezimalen Zahlen können groß oder klein geschrieben werden (a..f, A..F).

Einige Beispiele für gültige Zahlen:

```
12345      -12345      #12345
0x0ABCD   -0xABCD     $ABCD
%010101
```

Hinweis: Der numerische Interpreter kennt keine Fließkommazahlen, er arbeitet intern immer mit vorzeichenbehafteten 32-Bit-Integer-Werten. Darum ist „-123.45“ hier keine gültige Zahl !

8.1.1.2 Numerische Verknüpfungs-Operatoren

Der numerische Interpreter kennt die folgenden *einfachen* Operatoren (mit zwei „Eingängen“ und einem „Ausgang“):

```
+   Addieren
-   Subtrahieren
*   Multiplizieren
/   Division (ohne Rest)
%   Modulo (Berechnet den Disivionsrest)

==  Vergleiche: „gleich“
<> ungleich
<   kleiner als
<=  kleiner oder gleich
>   größer als
>=  größer oder gleich

&   bitweise UND-Verknüpfung
&&  boolesche UND-Verknüpfung
|   bitweise ODER-Verknüpfung
||  boolesche ODER-Verknüpfung
^   bitweises EXOR

~   bitweises NOT (prefix)
!   boolesches NOT (prefix)
```

Verknüpfungen sind nur für Integer-Werte definiert; das Verhalten mit Fliesskommawerten ist für die bitweisen Operatoren nicht definiert.

Ein Leckerbissen für alle C-Programmierer ist das „arithmetische if-then-else“:

Syntax: <arg1>?<arg2>:<arg3>

Funktion:

```
if (<arg1> notequal Zero)
  then result:=<arg2> ;
  else result:=<arg3>
```

Boolesche Verknüpfungen ergeben entweder den Wert TRUE oder FALSE, die hier durch die Werte 1 bzw. 0 repräsentiert werden.

Eingänge für boolesche Verknüpfungen sind: TRUE, wenn das Argument nicht Null ist, oder FALSE, wenn das Argument Null ist.

Die Priorität der Verknüpfungsoperatoren bestimmt die Reihenfolge beim Abarbeiten eines Formelausdrucks. Der Interpreter kennt nur zwei verschiedene Prioritätsstufen:

- niedrige Priorität für *Strichrechnung* wie Addition, Subtraktion, aber auch Vergleiche, OR;
- hohe Priorität für *Punktrechnung* wie Multiplikation, Division, Modulo, AND.

Im Zweifelsfall sollten Sie Klammern einsetzen. Ein geklammerter Ausdruck wird immer mit der höchsten Priortität ausgewertet.

8.1.1.3 Numerische Variablen

Der numerische Interpreter erkennt nur Variablen-Namen die mit einem Großbuchstaben beginnen (dies vereinfacht die Analyse von Ausdrücken, wichtig bei langsamen CPUs wie dem C515 im UPT515).

Im UPT515 darf ein Variablenname niemals länger als 8 Buchstaben sein.

Normalerweise stammen die *Inhalte* von Variablen aus Kommunikationskanälen des UPTs. Sie können allerdings auch Werte an Variablen mit dem Kommando „@“ zuweisen.

Falls eine Variable an einen Kommunikationskanal angekoppelt ist, können Sie ausser auf den *normalen Inhalt* der Variablen auch auf die folgenden Variablen-Komponenten zugreifen:

- .fl liest den Inhalt des „flag“-Felds der Variablen,
- .ut liest den Inhalt des „update timers“ der Variablen,
- .in liest den zuletzt vom Kommunikationskanal empfangenen (nicht skalierten) „Eingangswert“
- .ed liest den Inhalt des „Editier-Wertes“ der auch auf dem Display angezeigt wird,
- .ou liest den Inhalt des letzten zum Kommunikationskanal gesendeten „Ausgangswert“
- .de liest den Inhalt des „Default-Wertes“ aus der Variablen-Definitions-Tabelle
- .mi liest den Inhalt des „Min-Wertes“ aus der Variablen-Definitions-Tabelle
- .ma liest den Inhalt des „Max-Wertes“ aus der Variablen-Definitions-Tabelle
- .fa liest den Inhalt des Skalierungs-Faktors aus der Variablen-Definitions-Tabelle
- .di liest den Inhalt des Skalierungs-Divisors aus der Variablen-Definitions-Tabelle
- .of liest den Inhalt des Skalierungs-Offsets aus der Variablen-Definitions-Tabelle
- .uo liest die „output update flags“ der Variablen.

Wenn Sie keinen speziellen Komponenten-Namen beim Zugriff angeben (oder den Punkt zwischen Variablennamen und Komponenten-Namen vergessen), verwendet der Interpreter den „Editier-Wert“ der Variablen. Der „Editier-Wert“ ist derzeit auch der einzige Wert, den Sie per Zuweisungs-Kommando („@“) direkt ändern können.

8.1.1.4 Numerische Funktionen

Der Formelinterpreter erkennt *Funktionen* nur am Großbuchstaben (dies erleichtert die Unterscheidung von *Variablen*, die immer mit einem Kleinbuchstaben beginnen).

Auch alle *Ereignisse* (events) können als Teil eines numerischen Ausdrucks verwendet werden, weil sie Interpreter-intern wie normale *Funktionen* ausgewertet werden.

Der einzige Unterschied zwischen Event-Abfrage-Funktionen und anderen numerischen Funktionen ist der Ergebnistyp: Event-Abfrage-Funktionen liefern entweder den Wert „0“ oder „1“ (FALSE, TRUE), dagegen können numerische Funktionen auch andere Werte liefern.

Die folgenden Funktionen liefern numerische Werte:

cs: CAN-Status, bitcodierter Status des CAN-Controllers
tv0 ... tv3 : Timer Values der programmierbaren Timer
ti: Globale Zeit seit Systemstart, wird alle 100ms erhöht

kd, ku, kb: Tastatur-Abfrage-Funktionen. Siehe auch Kapitel 7.13.1 (events).

lim(value,min,max): Limit-Funktion.

returns: value, if (value>=min) and (value<=max)
min, if (value<min)
max, if (value>max).

pn: liest die aktuelle Display-Seiten-Nummer.

inp: liest den aktuellen Zustand der digitalen Inputs des UPTs („onboard“).

out: liest(!) den aktuellen Zustand der digitalen Outputs des UPTs („onboard“).

pdo: Einige Funktionen zum Zugriff auf PDO-Kanäle. Siehe Unten.

syn: Einige Funktionen zum Zugriff auf das CANopen-SYNC-Telegramm. Siehe Unten.

Eine detaillierte und komplette Erläuterung aller Funktionen finden Sie **ausschliesslich** im Online-Hilfesystem des UPT-Programmiertools !

8.1.1.5 PDO-Funktionen des UPT-Interpreters

Der numerische Interpreter enthält einige Funktionen zum Zugriff auf PDO-Kanäle. Einige dieser Funktionen können für Event-Definitionen oder als Teil von numerischen Ausdrücken verwendet werden.

Falls Sie einfach nur Werte aus empfangenen PDOs in Variablen übertragen wollen, brauchen Sie diese Funktionen **nicht** ! (Verbinden Sie zu diesem Zweck Variablen mit PDO-Kanälen).

Alle PDO-Funktionen, die im folgenden beschrieben werden, erfordern einen *Index* mit dem angegeben wird, mit welchem PDO-Kanal die Funktion arbeiten soll.

Sie können dazu ganz einfach die Angabe `pdo0`, `pdo1`, `pdo2` oder `pdo3` verwenden. Falls der Index flexibel sein muss, verwenden Sie einen Ausdruck in eckigen Klammern wie bei einem Array, z.B. „`pdo [PdoNr]`“ oder ähnliches.

Ein Punkt (.) trennt das Schlüsselwort „`pdo`“ von der Struktur-Komponente (siehe Beispiele). Die Namen der Struktur-Komponenten bestehe i.A. aus drei Buchstaben, sie können aber meistens durch den ersten Buchstaben abgekürzt werde, z.B. ist „`pdo [0] . cnt`“ das gleiche wie „`pdo [0] . c`“.

pdo [N] . cnt

Liefert den aktuellen Inhalt des PDO-Zählers. Verwenden Sie diese Funktion um zu testen, wie oft ein bestimmter PDO-Kanal (N) seit dem Systemstart übertragen wurde.

pdo [N] . cyc

Die aktuelle Zykluszeit für einen Sende-PDO in Millisekunden. Kann als formale Zuweisung per Interpreter auch geändert werden, z.B. `pdo[0].cyc = 1234` .

Der PDO-Sendezyklus entspricht weitgehend dem "PDO Event Timer" in CANopen DS301 V4.0, siehe Objekt 0x1400 subindex 5.

pdo [N] . dat [I]

Liefert das [i]-te Datenbyte aus PDO-Kanal Nummer [N]. Normalerweise werden Sie eine Variable mit einem PDO-Kanal verbinden, um auf einzelne Bytes eines PDO-Nutzdatenfelds zuzugreifen. (Diese Funktion wird daher nur für einige Sonderfälle gebraucht, z.B. wenn der Byte-Index variabel sein muß).

Kann als formale Zuweisung per Interpreter auch geändert werden.

pdo [N] . id

Dient zum Auslesen oder Ändern des aktuell verwendeten CAN identifiers eines PDOs.

Das Setzen eines neuen CAN-Identifiers erfolgt als formale Zuweisung (die allerdings mit Nebeneffekten verbunden ist.), z.B. `pdo[0].id = 2047`.

SIE SOLLTEN DIESE FUNKTION ÄUßERST SORGFÄLTIG EINSETZEN, DENN DER INTERPRETER ÜBERPRÜFT DIE GÜLTIGKEIT DES CAN-IDENTIFIERS NICHT !

pdo [N] . rcv

Diese Funktion testet, ob seit dem letztem Aufruf **dieser** Funktion (mit identischem N) ein PDO-Telegramm empfangen wurde.

Das Flag wird durch das Auslesen intern zurückgesetzt. Diese Funktion ist für den Einsatz als *Event* vorgesehen. In Kombination mit dem programmierbaren Timern eignet sie sich gut zu Überwachungszwecken für periodisch übertragene PDOs.

pdo [N] . trn

Diese Funktion testet, ob seit dem letztem Aufruf **dieser** Funktion (mit identischem N) ein PDO-Telegramm gesendet wurde.

Das Flag wird durch das Auslesen intern zurückgesetzt. Diese Funktion ist nur für den Einsatz als *Event* vorgesehen.

pdo [N] . tx

Prozedur zum unverzüglichen Anstoßen einer PDO-Sendung per Interpreter.

pdo [N] . mod

Diese Funktion testet, ob der Inhalt eines PDO-Telegramms geändert wurde, aber das Telegramm aber noch nicht abgeschickt.

Resultat: TRUE = Das PDO-Datenfeld wurde geändert aber noch nicht gesendet

FALSE= Das PDO-Datenfeld wurde seit der letzten Übertragung nicht geändert.

pdo [N] . syn

Diese Funktion liefert den Inhalt eines Zählers, der die SYNC-Telegramme für einen PDO-Kanal seit der letzten PDO-Übertragung zählt. Dieser Zähler wird intern zur Implementierung der PDO-Übertragungsarten 2..241 verwendet (siehe „Transmission Type“ nach CANopen, DS301).

8.1.1.6 SYNC-Funktionen des UPT-Interpreters

Der UPT-Interpreter enthält einige Funktionen, mit denen das CANopen- SYNC-Telegramm gesteuert bzw. überwacht werden kann. Für die normale SYNC-gesteuerte PDO-Übertragung sind diese Funktionen nicht erforderlich (alle PDO-Kanäle mit SYNC-gesteuerter Übertragung verarbeiten das SYNC-Telegramm selbst, ohne Hilfe des UPT-Interpreters).

syn.rcv (abgekürzt als `syn.r`)

Diese Funktion testet, ob ein SYNC-Telegramm seit dem letzten Aufruf dieser Funktion empfangen wurde.

Das entsprechende interne Flag wird durch den Aufruf dieser Funktion gelöscht. Der Zweck dieser Funktion ist der Einsatz in Event-Definitionen.

Z.B. können Sie damit einen eigenen „Sync-Empfangs-Wächter“ für periodisch empfangene SYNC-Telegramme programmieren.

Definieren Sie „`syn.rcv`“ als globales Event und ein Timer-Start-Kommando wie „`ts0(50)`“ als dazugehörige Reaktion. Dadurch wird Timer 0 für 500ms bei jedem SYNC-Empfang (neu) gestartet. Definieren Sie ein weiteres globales Event („`t0`“) und eine Reaction (z.B. `g"NoSync"`). Legen Sie eine Display-Seite namens „NoSync“ an. Diese Seite wird dann angezeigt, wenn das UPT für 500ms (oder länger) keine SYNC-Telegramme mehr empfängt.

syn.cnt (abgekürzt als `syn.c`)

Diese Funktion liefert den Inhalt eines Zählers für alle SYNC-Telegramme. Sie können diesen Wert zu Testzwecken einsetzen (z.B. um festzustellen, ob das UPT SYNC-Telegramme empfängt, ohne das Flag „`syn.rcv`“ zu löschen).

8.2 String-Funktionen

Im Mai 2000 wurde das sogenannte *Text-Array* beim UPT515 implemented. Einige String-Funktionen dienen zum Lese-Zugriff auf einzelne Zeilen dieses Array.

Die meisten String-Funktionen beginnen mit dem Buchstaben „s“ um sie von numerischen Funktionen zu unterscheiden. Derzeit (Mai 2000) kann das Ergebnis der String-Funktionen nur als Parameter für den Display-Anzeige-Befehl (Kapitel 7.10.3) eingesetzt werden.

sa [<text-array-INDEX>]

Liefert eine Zeile des Text-Arrays, definiert durch den Text-Array-INDEX. Der Text-Array-INDEX läuft immer von 0(=erste Zeile) bis N.

sr [<text-array-REFERENCE>]

Liefert eine Zeile des Text-Arrays, definiert durch die Text-REFERENZ-Nummer. Die Referenznummer wird bei der Eingabe des Text-Arrays mit eingegeben (siehe Kapitel 7.18). Text-Referenz-Nummern können beliebige 32-Bit-Zahlen sein. Die Suche nach der Referenz-Nummer erfolgt „linear“ über das Text-Array (d.h. eine Sortierung des Text-Arrays nach Referenz-Nummern ist unnötig). Darum eignet sich die „sr[]“-Funktion besonders zum Umsetzen beliebiger Fehlercodes in entsprechende Textmeldungen, wenn die Fehlercodes nicht „aufsteigend sortiert“ vorliegen.

(Möglicherweise gibt es noch weitere Stringfunktionen, die hier noch nicht aufgeführt sind. Untersuchen Sie ggf. das Windows-Hilfe-System des Programmiertools)

8.3 Interpreter-Kommandos

Die UPT-Firmware enthält einen einfachen Kommando-Interpreter, der für folgende Aufgaben eingesetzt wird:

- Abarbeiten von „Reaktionsmethoden“ bei Events, z.B. goto(g) oder call(c) – Kommando
- Graphik-Kommandos für komplexere Anzeigen
- Kommandos zum Ansteuern von Signal-LEDs und digitalen onboard-Ausgängen (UPT515),
- Kommandos zum Starten und Stoppen der *programmierbaren Timer*
- Zuweisungen von Werte an Display- und Systemvariablen, z.B. :
GearboxSwitchRequest := GearboxSwitchRequest + 1 // leg' mal den nächsten Gang ein
sys.audio_vol := 100 // Lautstärke für Audio-Wiedergabe auf Maximum

Einige Interpreter-Kommandos (*Prozeduren*) erwarten eine Parameterliste, andere benötigen keine Parameter.

Die meisten Kommando-Parameter sind numerische Ausdrücke.

Zu Testzwecken können Sie auch Kommandos über die Statuszeile des Programmiertools an den Interpreter schicken (oder über das neue „Test/Command“-Fenster).

8.4 Steuerung des Programmlaufs per Display-Interpreter

Hinweis: Für Geräte mit 32-Bit-CPU, z.B. MKT-View II / III / IV, empfehlen wir dringend, zur Steuerung des Programmlaufs die [Script-Sprache](#) zu verwenden !

8.4.1 Goto, Call und Return

Diese Kommandos des UPT-Kommando-Interpreters dienen zur Umschaltung zwischen verschiedenen Anzeigeseiten Ihrer Applikation. Diese Kommandos müssen (derzeit) in abgekürzter Form eingegeben werden.

Syntax:

```
goto:    g<page-number> oder g"page-name"  
call:    c<page-number> oder c"page-name"  
return:  r
```

Goto und **Call** schalten beide zu einer anderen Anzeigeseite um, wobei die neue Seite durch ihre Seitennummer oder (vorzugsweise) durch ihren Seitennamen angegeben wird.

Call speichert darüberhinaus die Nummer der aktuellen Seite (vor dem Aufruf) auf einem speziellen Stapel-Speicher (call stack), bevor zur neuen Seite umgeschaltet wird. Die so *aufgerufene* Seite kann dann später per *return* zur *aufrufenden* Seite zurückkehren.

Call + Return sollten immer kombiniert verwendet werden. Wenn Sie eine Seite per *call* aufrufen, sollten Sie diese auch per *return* wieder verlassen und nicht per *goto* wieder zum „caller“ zurückspringen (dies würde einen Überlauf des call-stacks verursachen, ausserdem würde dies keinen Sinn ergeben).

Der Grund für den Einsatz von **Call & Return** statt **Goto** ist, wenn mehrere verschiedene Aufrufer eine gemeinsame „Unter-Seite“ (wie ein „Unterprogramm“) aufrufen müssen. Nachdem diese „Unter-Seite“ ihre Arbeit getan hat (zum Beispiel eine Warnung anzeigen und auf Tastendruck warten), kann die normale Programmabarbeitung fortgesetzt werden.

Hinweis: Weder **Goto** noch **Call** haben einen Effekt, wenn das Sprungziel (=“neue“ Seite) identisch mit der aktuellen (=“alten“) Seite ist.

8.5 Das Zuweisungs-Kommando (@)

Dieses Kommando dient zur Zuweisung eines numerischen Wertes an eine Variable.

Syntax:

```
@<Varname> = <expression>
```

Beispiel:

```
@Ypos=(Voltage*63)/230
```

Hinweis:

- Sie können nur Werte an Variablen zuweisen, die in der Variablen-Tabelle definiert sind.
- Sie sollten keine Werte an Variablen zuweisen, die ihren „Input“ von einem Kommunikationskanal erhalten.

8.6 Weitere 'spezielle' Interpreter-Kommandos

.. wurden aus Gründen der Übersichtlichkeit aus diesem Dokument entfernt:

- Grafikbefehle (aus denen früher z.B. Balkendiagramme und Zeigerinstrumente zusammengesetzt werden mussten)
- Befehle zum Ansteuern der Tastatur-LEDs
- Befehle zum Ansteuern eventuell vorhandener digitaler Ausgänge ("onboard" im MKT-View)

8.6.1 Timer-Kommandos und -Funktionen

Das UPT verfügt über vier *User-Timer*, mit denen voneinander unabhängige Zeitintervalle von bis zu 65534*100ms erzeugt werden können (mehr als eine Stunde). Diese Timer sind eigentlich zum Auslösen von *Ereignissen* vorgesehen, Sie können sie allerdings auch für andere Zwecke einsetzen (Bewegen von Grafiken etc).

Um diese Timer zu steuern, bietet der UPT-Interpreter einige **Prozeduren** und **Funktionen** :

Timer - Prozeduren:

ts0 ... ts3 (timer start)

Startet einen der vier programmierbaren Timer für ein bestimmtes Zeitintervall. Der Timer wird von da an bis auf Null abwärts zählen. Wenn der Wert Null erreicht wird, bleibt der Timer stehen und löst ein *Ereignis* (t0..t3) aus.

Falls der Timer erneut gestartet wird, bevor er Null erreicht hat, wird kein Event ausgelöst. Die maximale Startwert dieser Timer ist 65534 (* 100 ms).

Syntax: ts0(<timer starting value in 100ms>)

Beispiel: ts0(40) Startet den ersten Timer für ein Intervall von 4 Sekunden. Nach vier Sekunden wird das *Event* „t0“ (s.U.) ausgelöst, welches Sie in einer globalen oder lokalen Event-Definition abfragen können.

tr0 ... tr3 (timer reset)

Setzt einen der vier Timer zurück. Dadurch wird der Timer gestoppt und das Auslösen des entsprechenden Timer-Events verhindert.

Intern enthält ein *zurückgesetzter* bzw. *gestoppter* Timer den Wert 0xFFFF (=65536).

Dies ist auch der Startzustand nach Power-ON.

Ein *abgelaufener* Timer enthält dagegen intern den Wert 0 !

Timer - Funktionen:

t0 ... t3

Testet (für Events), ob ein Timer *abgelaufen* ist, d.h. ob der Timer-Inhalt von 1 auf 0 dekrementiert wurde.

Ergebnis:

0 (=FALSE) bedeutet „der Timer läuft nicht oder ist noch nicht **abgelaufen**“,

1 (=TRUE) bedeutet „der Timer ist **abgelaufen**“.

tv0 ... tv3 (timer value)

Liest den Inhalt eines Zählregisters. Der Inhalt der Zählregister der vier programmierbaren Timer läuft **abwärts** vom Startwert bis Null. Alle 100 ms erfolgt ein Zähltakt.

ti (global time)

Liest den aktuellen Inhalt des „System-Taktgebers“ als 32-Bit-Wert. Dieser 32-Bit-Zähler wird alle 100 Millisekunden **inkrementiert**. Er hat nichts mit den vier „programmierbaren Timern“ zu tun und kann weder gestoppt noch durch das Anwenderprogramm modifiziert werden. Nach mehr als 6 Jahren ununterbrochenen(!) Betriebs des UPT-Terminals wird dieser Wert plötzlich negativ, weil sein Wert von $((2 \text{ hoch } 31)-1)$ auf $(2 \text{ hoch } 31)$ springt (was für 32-Bit-Werte eine negative Zahl ist). Nach Power-On des UPTs startet „ti“ immer bei Null.

In einigen Demo-Programmen wird diese Funktion verwendet, um nach dem Einschalten für kurze Zeit ein Firmen-Logo anzuzeigen.

9. Übertragung zwischen Programmiertool und Gerät

Um das mit dem Programmiertool erstellte Programm (*.upt bzw *.cvt) in das programmierbare Gerät ("Display") zu laden, bieten sich je nach Hardware verschiedene Methoden an. Details dazu finden Sie nur in der [Online-Hilfe](#) . Hier nur eine kurze Übersicht:

- Übertragung per Speicherkarte (SD memory card bzw Compact Flash)
- Übertragung per RS-232 (leider bei vielen PCs nicht mehr vorhanden)
- Übertragung per CAN-Bus (erfordert ein geeignetes Interface, oder ein CAN-via-UDP-fähiges Gerät als 'Brücke' zwischen Ethernet (LAN) und CAN)
- Übertragung direkt per Ethernet (LAN)

Zusätzlich zur Applikation (*.upt, *.cvt) müssen in einigen Fällen weitere Dateien in das Gerät geladen werden, z.B. Audio-Dateien, benutzerdefinierte Zeichensätze. Dazu kann (z.B. wenn kein USB-Kartenleser vorhanden ist) das im Programmiertool integrierte File-Transfer-Utility verwendet werden.