

PROGRAMMIERHANDBUCH

MKT-VIEW+

INHALTSVERZEICHNIS

1	HAFTUNG	4
2	ALLGEMEINES	4
3	DAS PROGRAMMIERTOOL	5
3.1	DIE FENSTER DES PROGRAMMIERTOOLS	6
3.1.1	Haupteingabefenster	7
3.1.2	Das LCD-Simulationsfenster.....	8
3.1.3	Das Watch-Fenster.....	9
3.1.4	Das Test-Kommando-Fenster.....	10
3.2	AUFRUF DES PROGRAMMIERTOOLS	10
3.3	LADEN EINES PROGRAMMS.....	10
3.4	BESCHREIBUNG DES HAUPTINGABE-FENSTERS	11
3.4.1	Beschreibung der Hauptmenüleiste.....	11
3.4.1.1	Menüpunkt Datei	11
3.4.1.2	Menüpunkt Assistent	11
3.4.1.3	Menüpunkt Transfer	12
3.4.1.4	Menüpunkt Ansicht	12
3.4.1.5	Menüpunkt Werkzeuge.....	13
3.4.1.6	Menüpunkt Icon.....	13
3.4.2	Beschreibung der verschiedenen Karteikartenseiten	14
3.4.2.1	Karteikartenseite CANdb-Daten	14
3.4.2.2	Karteikartenseite Variablen	14
3.4.2.3	Karteikartenseite globale Events.....	15
3.4.2.4	Karteikartenseite Seite #X	16
3.4.2.4.1	Beschreibung der Seite Definitionskopf.....	16
3.4.2.4.2	Beschreibung der Seite Anzeigzeilen-Definitionen	17
3.4.2.4.2.1	Eingabe von Texten.....	18
3.4.2.4.2.2	Eingabe von Buttons	22
3.4.2.4.2.3	Interpreterkommandos zum Steuern der Buttons auf einer Anzeigeseite	24
3.4.2.4.2.4	Eingabe eines Icons.....	24
3.4.2.4.2.5	Eingabe eines Bargrafen.....	25
3.4.2.4.2.6	Eingabe eines y(t) bzw. y(x) Diagramms	28
3.4.2.4.3	Beschreibung der Event-Definitionen einer Seite	31
3.4.2.4.3.1	Tastaturevents	31
3.4.2.4.3.2	Timerevents.....	32
3.4.2.4.3.3	Event durch Wertevergleich	32
3.4.2.4.3.4	Andere Ereignisse	33
3.4.2.4.4	Beschreibung der „weiteren Kommandos“	33
3.4.2.4.4.1.1	Das LINE-Kommando (li).....	34
3.4.2.4.4.1.2	Das PIXEL-Kommando (pi).....	34
3.4.2.4.4.1.3	Das FRAME-Kommando (fr).....	34
3.4.2.4.4.1.4	Das FillRectangle-Kommando (fi)	34
3.4.2.5	Beschreibung der Karteikartenseite <i>Seitenübersicht</i>	35
3.4.2.6	Karteikartenseite „Symbole“	39
3.4.2.7	Karteikartenseite „Einstellungen“.....	40
3.4.2.7.1	Terminal - Konstanten.....	40
3.4.2.7.2	Parametertabelle	40
3.4.2.7.2.1	Baudrateneinstellung des Terminals.....	40
3.4.2.7.2.2	Max. Seitenanzahl eines Programms	41
3.4.2.7.2.3	Max. Textarrayeinträge	41
3.4.2.7.2.4	Tastenkombination zum Aufruf des Systemmenüs (im Terminal)	41
3.4.2.7.2.5	EEPROM-Speicherzellen	42
3.4.2.7.3	Programmierschnittstellen-Einstellungen.....	42
3.4.2.7.4	Einstellungen des Programmierertools	43
3.4.2.8	Karteikartenseite „Textzeilen“.....	43

Programmierhandbuch MKT-View-Terminal Version 1.01

3.4.2.9	Karteikartenseite „Fehler“	44
4	INTERPRETER-KOMMANDOS UND -FUNKTIONEN.....	45
4.1	NUMERISCHE FUNKTIONEN	45
4.2	DAS ZUWEISUNGS-KOMMANDO (ASSIGN, @).....	46
4.3	„SYSTEM“-FUNKTIONEN UND -KOMMANDOS DES INTERPRETERS	46
4.4	FUNKTIONEN ZUR ABFRAGE DER ECHTZEITUHR (REAL TIME CLOCK).....	47
4.5	DIGITALE EIN- / AUSGÄNGE.....	48
4.5.1	<i>Funktionen zum Lesen der digitalen Eingänge</i>	<i>48</i>
4.5.2	<i>Das „Output“-Kommando</i>	<i>49</i>
4.6	FUNKTIONEN ZUR ABFRAGE DES DREHKNOPFES (ROTARY ENCODER, ROT.XXX)	49
4.7	TIMER-KOMMANDOS UND -FUNKTIONEN	51
4.8	SEITENUMSCHALTUNG PER INTERPRETER.....	52
4.8.1	<i>Goto, Call und Return (g,c,r)</i>	<i>52</i>
4.8.2	<i>Page-Scan-Modus (pscan)</i>	<i>53</i>
4.8.3	<i>Kommandos zum Ansteuern des Piepers.....</i>	<i>54</i>
4.8.3.1	<i>beep-Kommando</i>	<i>54</i>
4.8.3.2	<i>play-Kommando</i>	<i>55</i>
4.9	CAN - SENDE - KOMMANDO (CTX).....	57
4.10	SPEZIELLE GRAFIK-KOMMANDOS.....	58
4.10.1	<i>Invertierung der Anzeige (Display Inversion, di).....</i>	<i>58</i>
4.10.2	<i>Das CLearScreen-Kommando (cl)</i>	<i>58</i>

1 Haftung

Diese Soft- und dazugehörige Hardware kann ein CAN-Netzwerk steuern oder beeinflussen. Der Einsatz kann daher zu schweren Schäden führen. Aus diesem Grund dürfen nur Personen, die die möglichen Konsequenzen des Einsatzes dieses Produktes verstanden haben, es verwenden.

Die Haftung des Herstellers ist auf die Terminal-Hardware beschränkt, sie bezieht sich nicht auf die Folgen seines Einsatzes in einem CAN-Netzwerk. Weder MKT Systemtechnik noch deren Beschäftigte oder Partner können für den unsachgemäßen Gebrauch dieses Produktes haftbar gemacht werden. Weitere Infos bei <http://www.mkt-sys.de/>.

Wenn Sie mit den obigen Einschränkungen nicht einverstanden sind, deinstallieren Sie die Software und senden die Hardware UMGEHEND an den Lieferanten zurück. Der Kaufpreis, den Sie für das Gerät bezahlt haben, wird Ihnen dann erstattet. Kunden von MKT Systemtechnik wenden sich bitte an Herrn Habermann.

2 Allgemeines

Das Programmiertool ist aus einer Programmiersoftware für ein CANopen Terminal entstanden und wird auch weiterhin für diese Anwendung genutzt. Diesbezüglich sind bestimmte Punkte in der Bedienung vorgesehen, die nicht für die Programmierung des MKT-View+ relevant sind. Die Programmierung erfolgt im Hause MKT nach dem „Single-Source-Prinzip“. Das heißt, es gibt nur einen Programmcode, der über verschiedene Compiler für verschiedenste Zielsysteme übersetzt wird. Vorteil dieses Prinzips ist, dass der Programmcode, der auf dem PC läuft, der gleiche ist wie der auf dem MKT-View+. Durch diese Vorgehensweise ist eine schnellere und fehlerfreiere Programmierung möglich.

3 Das Programmiertool

Zur Erstellung des Anzeigeprogramms dient ein spezielles PC-Programm (das "Programmiertool"), welches unter Windows 98, Windows 2000, ME, und Windows XP lauffähig ist. Das "Anzeigeprogramm" wird in der folgenden Beschreibung "*Applikation*" genannt - nicht "*Projekt*" ! Zu einem kompletten *Projekt* würden noch zahlreiche weitere Dateien und Dokumente gehören, die nicht in der vom Programmiertool erstellten *.cvt-Datei enthalten sind.

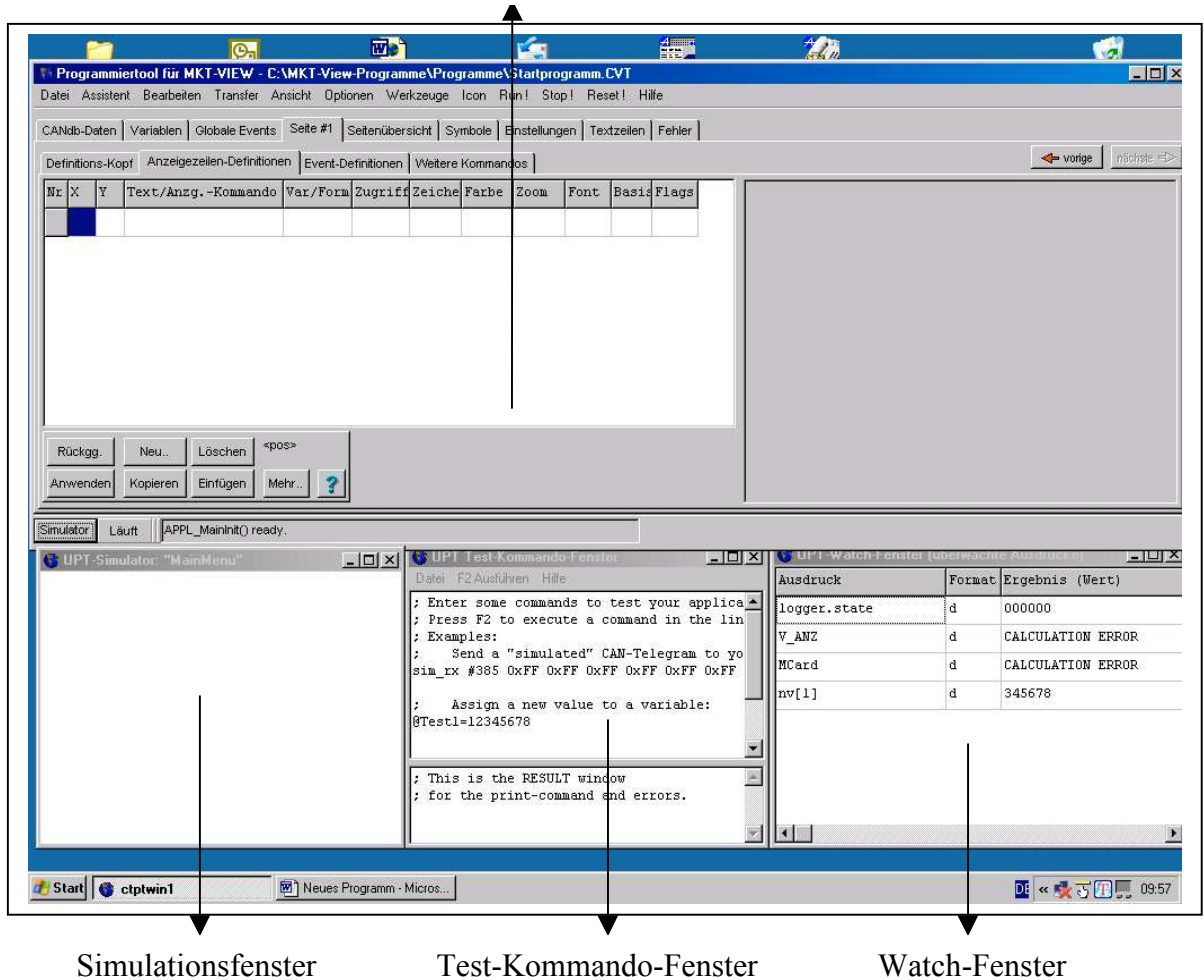
Nach Erstellung und Test der *Applikation* kann diese vom Programmiertool über die serielle Schnittstelle (COM1 oder COM2) in das Terminal geladen werden. Bei bestimmten Geräten ist auch die Übertragung per CompactFlash-Speicherkarte möglich.

Hinweise:

Da wir das MKT-View+ und das Programmiertool ständig weiterentwickeln, sollten Sie öfters überprüfen, ob eine neue Version auf unserer Homepage vorliegt. Bei dem Download wird die aktuelle Firmware mitgeladen, so dass Ihr MKT-View+ auf dem neuesten Stand bleibt. Weitere aktuelle Hinweise zur Bedienung finden Sie in der Online-Dokumentation des Programmiertools.

3.1 Die Fenster des Programmiertools

Haupteingabefenster



Das Programmiertool besteht aus den folgenden *nicht-modalen* Fenstern..

- Haupteingabe-Fenster
- LCD-Simulator-Fenster
- Watch-Fenster
- Test-Kommando-Fenster

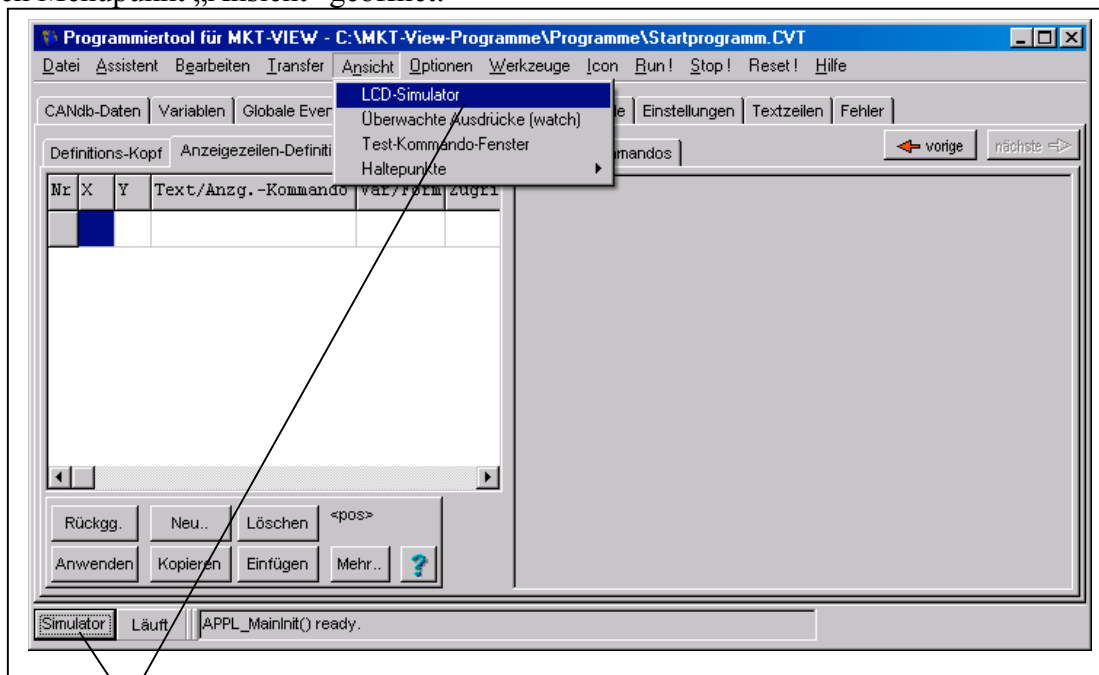
Darüber hinaus existieren etliche *modale*¹ Dialogfenster, die hier nicht aufgeführt sind.

¹ Ein *modales* Fenster blockiert bis zum Anklicken von "OK" alle anderen Fenster des Programms.

3.1.1 Haupteingabefenster

Das Haupteingabefenster wird mit dem Starten des Programms automatisch geöffnet. In diesem Fenster wird die Programmerstellung durchgeführt

Das LCD-Simulationsfenster wird entweder über den unteren Button „Simulator“ oder über den Menüpunkt „Ansicht“ geöffnet:

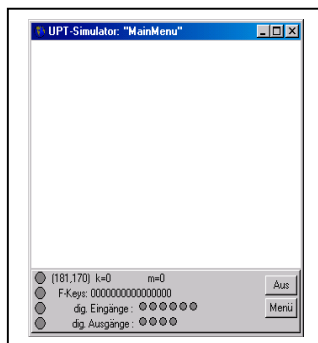


Aufruf des LCD-Simulationsfensters

Auf der CD finden Sie weiterhin verschiedene Dokumente im Word-Format, zu denen aus technischen Gründen leider keine Links in dieser Übersicht führen.

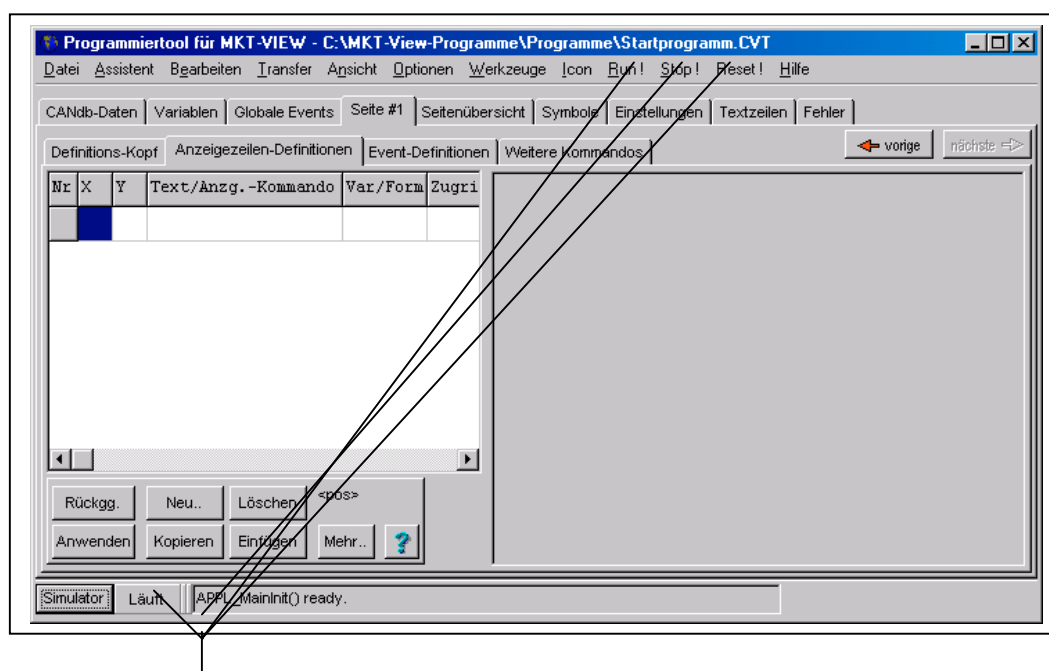
3.1.2 Das LCD-Simulationsfenster

Im LCD-Simulationsfenster wird die LCD-Anzeige simuliert. Der Nutzer hat damit die Möglichkeit, seine Programmierung zu überwachen und zu testen.



Der untere Teil des LCD-Simulationsfensters (dient zur Simulation der Ein- und Ausgänge) kann über den Button *Menu* ausgeblendet werden. Optional kann ein Hintergrundbild mit dem Tastatur-Layout ("Tastaturfolie") geladen werden, was die Ausrichtung grafischer Elemente neben den Funktionstasten erleichtert.

Eine Applikation kann aus bis zu 149 Anzeigeseiten bestehen (einstellbar über das Karteikartenblatt *Einstellungen*). Die programmierte Applikation kann auf dem PC simuliert werden. Dafür dienen die Menüpunkte *Reset!*, *Run!* und *Stop!* und der Button neben dem Simulator, in dem der momentane Status angezeigt wird.



Bedienelemente für die Simulation des Programms:

- **Reset !:** Durch die Betätigung des *Reset !*-Menüpunktes wird ein Power-On-Reset des Terminals simuliert. Das heißt, es wird automatisch die Page 0 aufgerufen und ausgeführt. Es werden die entsprechenden Seitenevents der Page 0 ausgeführt und die globalen Events durchgeführt. Der aktuelle Programmzustand wird unten im Fenster neben dem Button *Simulator* als Button *Läuft* angezeigt.
- **Stop !:** Das Programm kann jederzeit durch die Betätigung des Menüpunktes *Stop!* oder durch die Betätigung des Buttons *Läuft* angehalten werden.
- **Run !:** Das Programm kann durch die Betätigung des Menüpunktes *Run !* jederzeit wieder gestartet werden. Der Unterschied zu *Reset* besteht darin, dass das Programm dann von der aktuell angewählten Seite aus startet.
- **Button unten neben den Button *Simulator*:** In diesem Button wird der aktuelle Zustand des Programms angezeigt: *Läuft* oder *Gestoppt* oder bei der Eingabe von neuen Displayelementen *Editieren*. Durch die jeweilige Betätigung des Buttons wird zwischen diesen Zuständen umgeschaltet.

3.1.3 Das Watch-Fenster

In dem Watch-Fenster können Variablen eingetragen werden, deren Wert dann bei der Simulation angezeigt und geändert werden können (siehe Programmsimulation).

3.1.4 Das Test-Kommando-Fenster

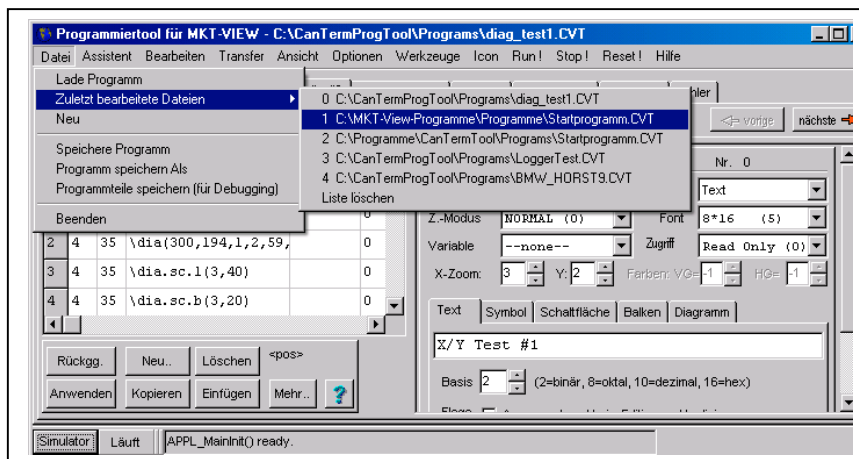
Im Test-Kommando Fenster können CAN-Messages simuliert werden (siehe Programmsimulation)

3.2 Aufruf des Programmiertools

Nach der Installation des Programmiertools wird ein Icon auf dem Desktop eingeblendet, mit dem das Programmiertool gestartet werden kann. Als erstes wird ein Warnhinweis eingeblendet, der bitte genauestens durchgelesen werden muss. Die Umschaltung der Bedienungssprache geschieht durch das Antippen der entsprechenden Fahne.

3.3 Laden eines Pogramms

Nach dem Starten des Programmiertools wird automatisch das zuletzt bearbeitete Programm geladen. Soll ein anderes Programm bearbeitet werden, kann über den Menüpunkt *Datei* ein entsprechendes Programm geladen werden, wobei mit den Menüpunkt *Zuletzt bearbeitete Dateien* ein entsprechendes Programm ausgewählt werden kann.



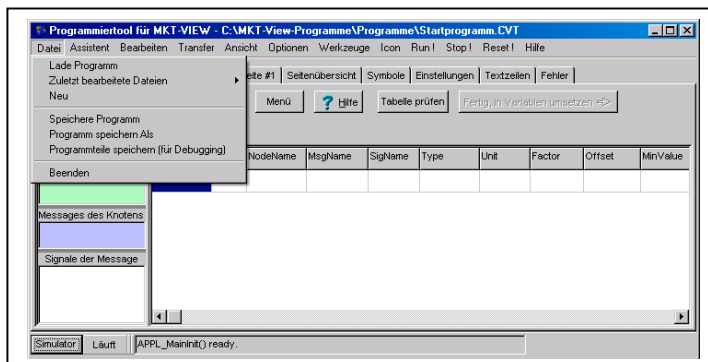
3.4 Beschreibung des Haupteingabe-Fensters

Das Haupteingabefenster dient zur Programmierung der verschiedenen Anzeigeseiten. Es wird über eine Menüleiste und über verschiedene Karteikartenseiten bedient.

Erläuterung der einzelnen Menüpunkte:

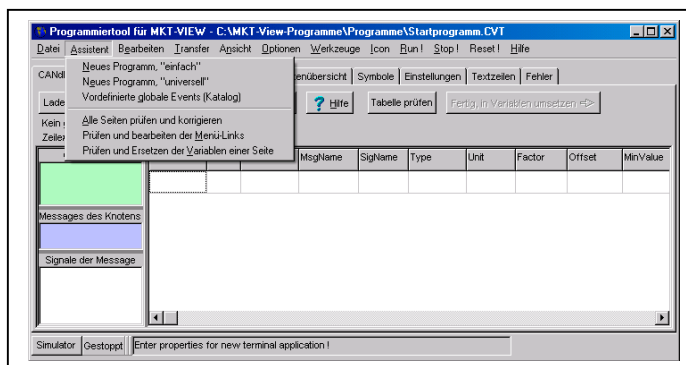
3.4.1 Beschreibung der Hauptmenüleiste

3.4.1.1 Menüpunkt Datei



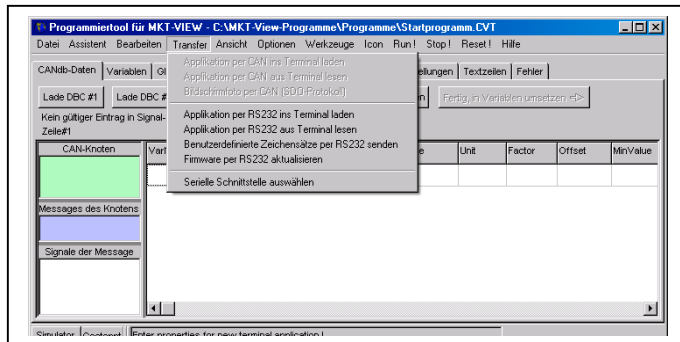
Über diesen Menüpunkt werden Programme geladen und gespeichert.

3.4.1.2 Menüpunkt Assistent



Über die verschiedenen Assistenten kann der Nutzer in mehreren Schritten neue Anzeigeprogramme anlegen, die aus n vielen Seiten bestehen, er kann sich pro CAN-Schnittstelle eine CANdb-Datei einlesen und er kann das Bedienungsprinzip festlegen. Siehe spätere Erläuterung *Erstellung eines Programms über den Assistenten*.

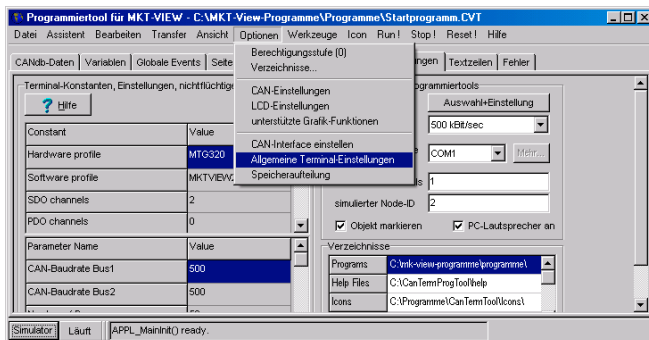
3.4.1.3 Menüpunkt Transfer



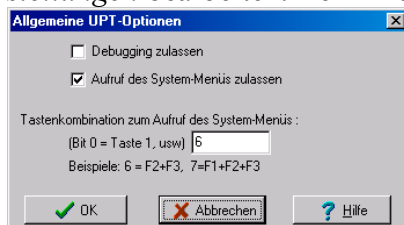
Über den Menüpunkt Transfer wird die Kommunikation mit dem Terminal gesteuert. Nach der Installation des Programmiertools müssen Sie einmalig die serielle Schnittstelle auswählen, über die Sie mit dem Terminal kommunizieren wollen. Wenn Sie das Terminal programmieren möchten, müssen Sie das MKT-View+ mit Spannung versorgen und das Terminal entweder über das Verbindungskabel Art.Nr.: 25009 oder über die Verteilerbox 70089 und die Verbindungskabel 20110+25000 mit dem PC verbinden (Programmtransfer in das MKT-View+).

3.4.1.4 Menüpunkt Ansicht

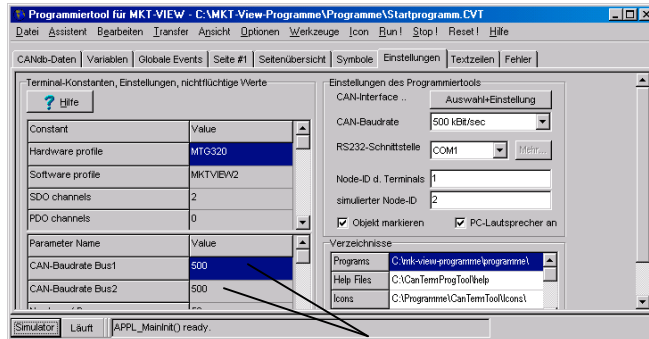
Siehe Beschreibung Programmiertoolfenster Menüpunkt Optionen



Unter den Menüpunkt Optionen sollten Sie zuerst nur den Punkt *Allgemeine Terminal-Einstellungen* bearbeiten. Beim Aufruf wird folgende Seite eingeblendet:

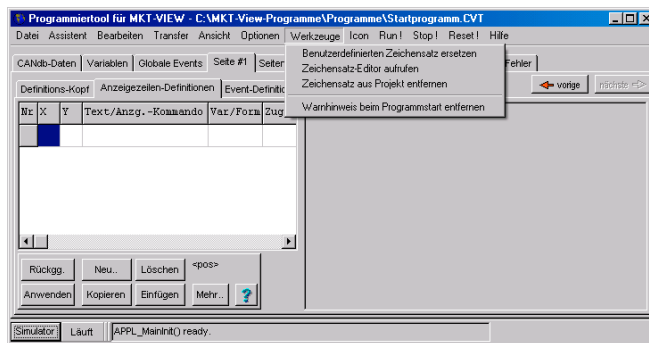


In diesem Fenster legen Sie fest mit welchen Funktionstasten Sie in das Systemmenü des Terminals kommen (siehe Hardwarebeschreibung Terminal –Systemmenü). In der Regel sollten Sie diese Einstellung nicht ändern.



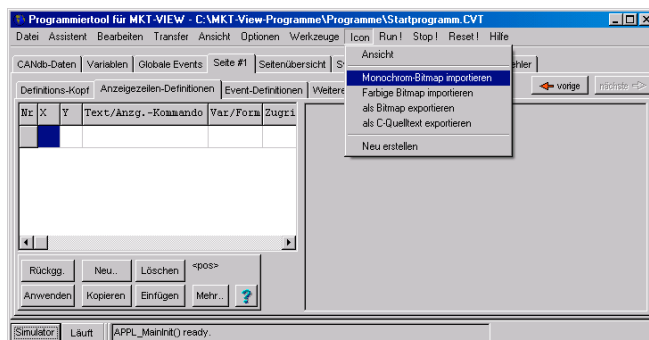
Einstellen der Baudrate für die CAN-Schnittstellen. Die Baudrateneinstellung auf der rechten Seite ist für die Programmierung des Terminals über die CAN-Schnittstelle gedacht. In dem nächsten Fenster müssen Sie in der Regel nur die gewünschten Baudraten der beiden CAN-Schnittstellen eingeben. Dies geschieht durch das Antippen des entsprechenden Feldes

3.4.1.5 Menüpunkt Werkzeuge



Es können mit dem Programmiertool bis zu 10 benutzerspezifische Zeichensätze geladen werden. Beim Laden des Programms in das MKT-View+ wird überprüft, ob der Zeichensatz schon geladen ist. Wenn nicht, wird der Benutzer aufgefordert, sie in das Terminal zu laden.

3.4.1.6 Menüpunkt Icon



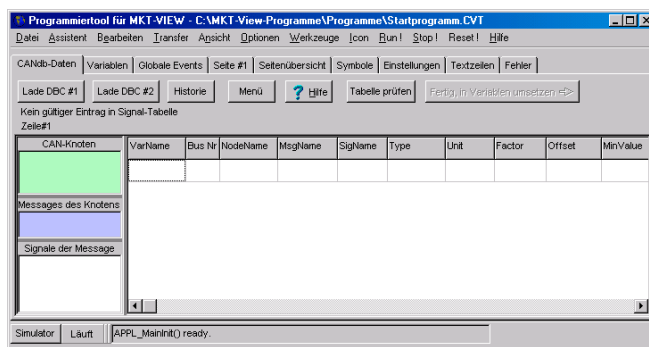
Programmierhandbuch MKT-View-Terminal Version 1.01

Über den Menüpunkt Icon können Grafiken (BMP-Format) importiert werden. Bitte nur den 2. Punkt (Monochrome Bitmaps importieren) nutzen. Die Bitmaps können Sie über mspaint.exe oder Corel Draw erstellen (max. Größe 320x240 Pixel und SW). Sie können nur Icons in Ihr Programm integrieren, wenn Sie diese vorher importiert haben.

3.4.2 Beschreibung der verschiedenen Karteikartenseiten

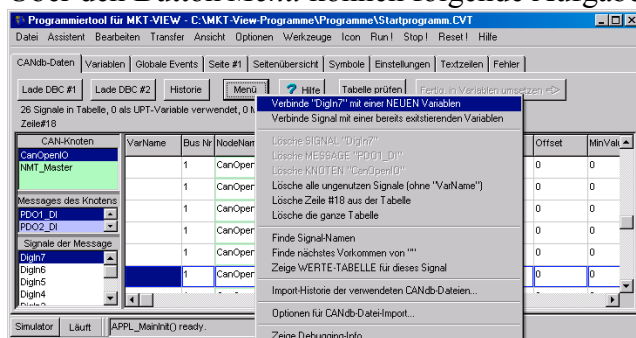
Das Hauptfenster des Programmiertools ist in verschiedene Registerkarten (hier: *Karteikarten*) aufgeteilt, die -von der logischen Abfolge her- bei der Erstellung einer neuen Terminal-Applikation von links nach rechts durchlaufen werden.

3.4.2.1 Karteikartenseite CANdb-Daten



Auf dieser Seite können Sie über die Button *Load DBC #1* und *Load DBC #2* eine oder mehrere CANdb-Dateien für die einzelnen CAN-Schnittstellen laden. Des Weiteren selektieren Sie auf dieser Seite die Signale, die Sie abbilden möchten (siehe nächster Punkt).

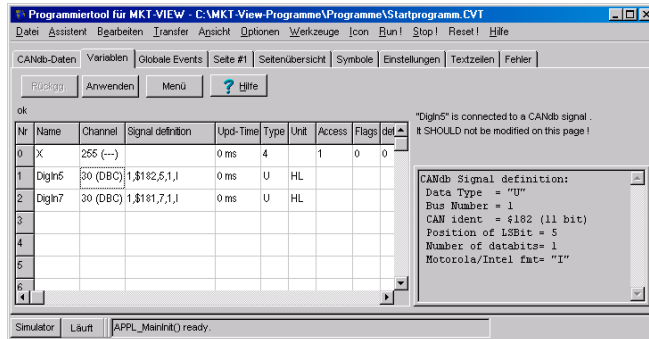
Über den Button *Menü* können folgende Aufgaben durchgeführt werden:



Im ersten Anlauf benötigen Sie nur den ersten Punkt. Die weiteren Punkte werden später erläutert, z.B. Ersetzen einer CANdb.

3.4.2.2 Karteikartenseite Variablen

Auf der Seite werden alle Signale, die in eine Variable umgewandelt wurden und alle Hilfsvariablen angezeigt.



In dem Programmierwerkzeug können zusätzlich zu den Variablen, die aus den Signalen erzeugt wurden, Hilfsvariablen erstellt und benutzt werden. In diesen Hilfsvariablen können Zwischenergebnisse abgespeichert werden oder sie können editiert und dann als variable Vergleichswerte (z.B. max. Drehzahl) benutzt werden (siehe Beschreibung Kommando-Interpreter). In der Spalte Channel wird gekennzeichnet, was für ein Typ die Variable ist.

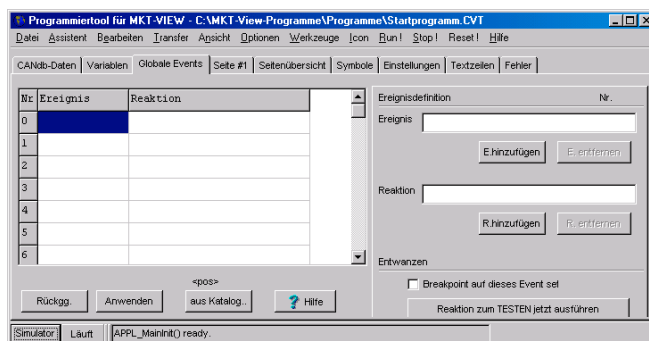
- Kanalnummer 30 kennzeichnet eine CANdb Variable
- Kanalnummer 255 kennzeichnet eine Hilfsvariable

Bei den CANdb-Variablen werden alle Spalten automatisch mit den Werten aus der CANdb-Datei ausgefüllt.

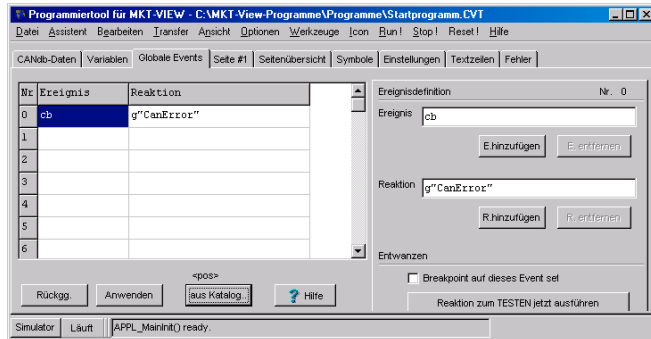
Bei der Deklaration einer Hilfsvariablen müssen Sie wie folgt vorgehen:

Sie tragen in die Spalte Name eine gewünschte Bezeichnung (erster Buchstabe groß, maximale Länge 16 Zeichen) Anschließend schreiben Sie in die Spalte Channel 255 und bestätigen danach diese Eingabe durch die Betätigung des Buttons **Anwenden**. Danach können Sie auf der rechten Seite alle gewünschten Angaben durchführen. Am Ende müssen Sie diese Angaben **unbedingt** noch mal durch die Betätigung des **Buttons Anwenden bestätigen**.

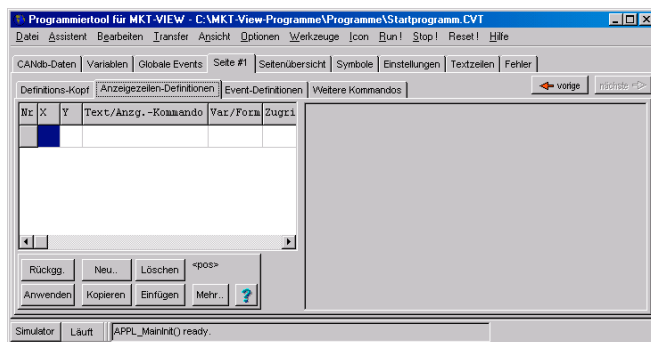
3.4.2.3 Karteikartenseite globale Events



Auf dieser Karteikartenseite können Sie Ereignisse programmieren und die Reaktionen, wenn dieses Ereignis „Wahr“ ist (siehe auch das Kapitel ‚Programmierung von Ereignissen‘). Die auf dieser Seite programmierten Ereignisse werden immer überwacht, egal welche Seite abgebildet wird. Es ist also sinnvoll, auf dieser Seite zum Beispiel die Reaktion eines CAN-Bus Off Ereignisses zu programmieren



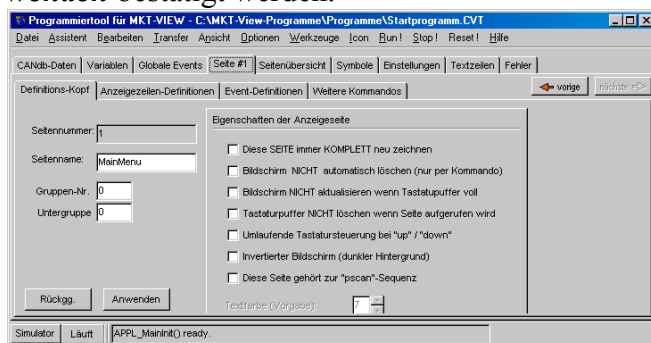
3.4.2.4 Karteikartenseite Seite #X



Auf der Karteikarte "Seite #X" wird die aktuell angewählte Seite X programmiert (siehe Kapitel Seitenprogrammierung). Die Seitenprogrammierung erfolgt über 4 Definitionskarteikarten und einer Zusatzkommandokarteikarte.

3.4.2.4.1 Beschreibung der Seite Definitionskopf

ACHTUNG alle Änderungen auf dieser Seite sollten durch die **Betätigung des Buttons Anwenden** bestätigt werden.



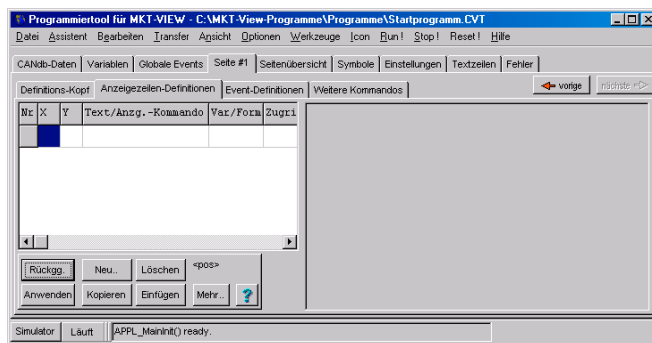
Auf dieser Seite können Sie jeder Seite einen Namen geben (max. Länge 8 Zeichen). Des Weiteren können Sie für die aktuelle Seite noch folgende Punkte festlegen:

- *Diese SEITE immer KOMPLETT neu zeichnen*
nur sinnvoll, wenn mehrer Bilder sich überlappen. Möglichst vermeiden, weil dadurch der Bildaufbau langsamer wird

Programmierhandbuch MKT-View-Terminal Version 1.01

- *Bildschirm NICHT automatisch löschen (nur per Kommando)*
nur sinnvoll, wenn der Inhalt von der alten Seite nicht gelöscht werden soll
- *Bildschirm NICHT aktualisieren wenn der Tastaturpuffer voll*
nutzen, wenn sehr schnelle Änderungen über den Drehgeber oder über die Tastatur vorgenommen werden sollen. Beim MKT-View+ kaum nötig
- *Tastaturpuffer nicht löschen wenn Seite aufgerufen wird*
siehe Online-Hilfe
- *Umlaufende Tastatursteuerung bei "up" / "down"* (Encoder links bzw. rechts drehen). Wird benutzt bei einer Menüauswahlseite, um den Auswahlbalken rundherum laufen zu lassen.
- *Invertierter Bildschirm (dunkler Hintergrund)* (weiße Schrift auf dunklem Hintergrund) kann auch per Interpreterkommando ausgelöst werden.
- *Diese Seite gehört zur "pscan"-Sequenz*
In diesem Modus schaltet das Terminal zyklisch zwischen verschiedenen Anzeigeseiten um. Dazu werden keine Event-Definitionen benötigt. Der Page-Scan-Modus kann per Interpreter gesteuert werden (per pscan.next oder pscan.prev)

3.4.2.4.2 Beschreibung der Seite Anzeigezeilen-Definitionen

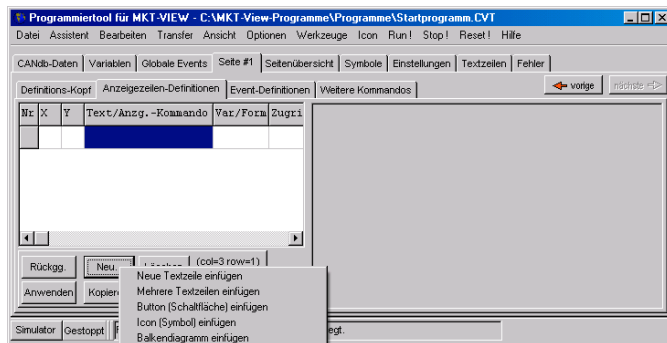


Auf der Seite *Anzeigezeilen-Definitionen* werden die einzelnen Elemente programmiert, die auf der Seite abgebildet werden sollen. Die Schaltflächen unterhalb der Tabelle haben die folgenden Funktionen:

- *Neu:* Einfügen eines neuen Anzeige-Elements (aus Liste)
- *Rückgg:* Rückgängigmachen der letzten Eingabe
- *Löschen:* Löscht das selektierte Element (hier: Zeile der Tabelle)
- *Kopieren:* Kopiert das selektierte Element (hier: Zeile) in einen Puffer
- *Einfügen:* Fügt das Element aus dem Puffer an der aktuellen Tabellenposition ein

Es können auf den Seiten *Anzeigezeilen-Definitionen*, *Event-Definitionen* und *Weitere Kommandos* insgesamt max. 30 Elemente eingegeben werden.

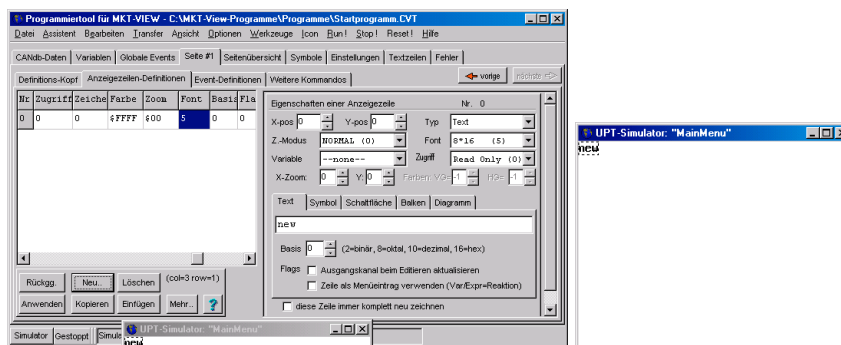
Über den Button *Neu* können folgende Elemente eingegeben werden:



- Texte
- Texte mit Signalwerten (auch über Texte)
- Buttons (Schaltflächen) für die Beschriftung der Funktionstasten
- Icon
- Balkendiagramme
- $y(t)$ bzw. $y(x)$ - Diagramme

3.4.2.4.2.1 Eingabe von Texten

Eingabe von reinen Texten



Der neue Text wird im Simulationsfenster oben links eingeblendet und kann per Maus verschoben werden.

Folgende Parameter können über das rechte Eingabefeld verändert werden:

Die Textkoordinaten (der Koordinatenursprung ist oben links)

Programmierhandbuch MKT-View-Terminal Version 1.01

Folgende Zeichenmodi lassen sich anwählen (es gibt zwei Anzeigeseiten, die abwechselnd alle 0,5 s abgebildet werden):

Zeichenmodi	Anzeigeseite 1	Anzeigeseite 2
Normal	Text normal	Text normal
Page 1	Text normal	leer
Page 2	leer	Text normal
Blink 1	Text invers	Text gelöscht
Blink 2	Text gelöscht	Text normal
Invers	Text invers	Text invers
Invers 1	Text invers	leer
Invers 2	leer	Text invers
I_Blink1	Invers	Normal
I_Blink2	Normal	Invers

- Font Auswahl des benutzten Zeichensatzes
- x und y Zoom Zoomen des selektierten Elementes

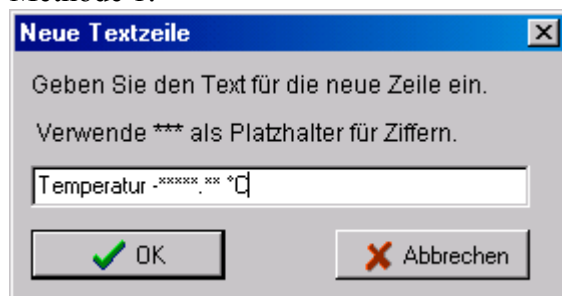
Eingabe von Texten mit Variablen:

Bei der Eingabe von Texten mit Variablen können folgende Parameter eingestellt werden:

- Die Zahlenbasis, wobei die 0 defaultmäßig der Zahlenbasis 10 entspricht
- Auswahl der eingeblendeten Variablen. Durch das Anklicken des Auswahlpfeils werden alle Variablen angezeigt und Sie können sich die gewünschte aussuchen.
- Soll die Variable editierbar sein, müssen Sie dies durch Verändern der Zugriffsberechtigung kennzeichnen (ändern in „Read/Write“). Dann können sie diese Variable ändern (siehe editieren von Werten). Nur sinnvoll bei einer Hilfsvariablen, die gesendet werden soll (siehe Message senden). Während des Editierens wird der **Wert noch nicht in die Variable übernommen**.

Dies kann durch die Anwahl des Flags „Ausgangskanal beim Editieren ändern“ angewählt werden.

Methode 1:



Bei dieser Methode geben Sie den gewünschten Text ein und tragen an der Stelle an der der Variablenwert eingetragen werden soll **** Zeichen ein. An dieser Stelle wird dann später die Zahl eingetragen. Folgende Sonderzeichen werden berücksichtigt:

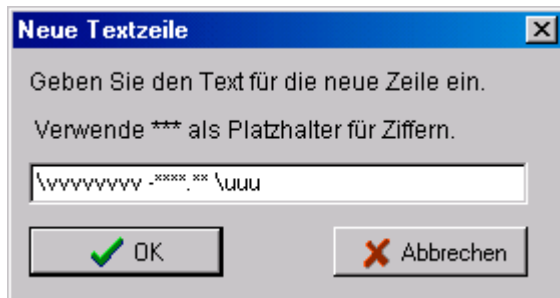
"-" bzw. "+" Zeichen: das Anzeigesystem beachtet auch das Vorzeichen

"." (Punkt) die Zahl wird als Floating-Pointe-Zahl dargestellt

Programmierhandbuch MKT-View-Terminal Version 1.01

Alle anderen Zeichen können als optisches Trennzeichen benutzt werden, z.B. um bei Binärzahlen die Bits "einzeln" (gesperrt) darzustellen: * * * * * ergibt 1 0 1 0 0 1 1

Methode 2:



Es werden an den Stellen an denen der Variablen(Signal)name abgebildet ein Platzhalter \v eingegeben und an der Stelle an der die Maßeinheit eingeblendet werden soll der Platzhalter \u. Wird jetzt diesem Text eine Variable zugeordnet wird automatisch an Stelle der Platzhalter der entsprechende Name der Variable und die entsprechende Maßeinheit eingeblendet.

Diese Methode hat folgende Vorteile:

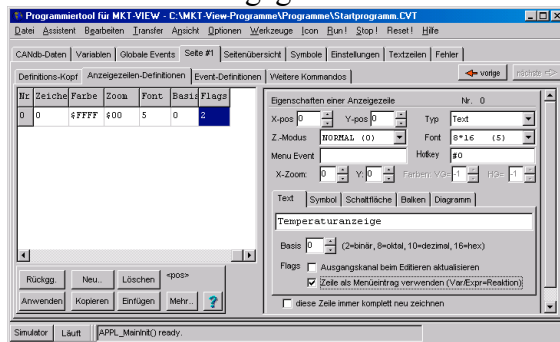
Die Texte sind formatiert, das heißt die Kommastellen bleiben alle untereinander

Diese Text lassen sich einfach kopieren und dann durch eine einfache Auswahl der gewünschten Variable anpassen

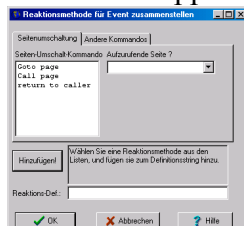
Eingabe von Texten als Menüzeileneintrag

Auf dem MKT-View+ können zur Auswahl von Anzeigeseiten Anzeigemenü erstellt werden.

Dafür wird der eingegebene Text als *Zeile als Menüeintrag verwenden* gekennzeichnet.



Anschließend geben Sie in dem Feld *Menu Event* die gewünschte Reaktion ein. Das geschieht durch ein Doppelklicken in das Feld „Menu event“. Danach wird folgendes Fenster geöffnet:



und Sie können sich die gewünschte Reaktion selektieren.

Programmierhandbuch MKT-View-Terminal Version 1.01

Zusätzlich können Sie noch einen Hotkey eintragen. Folgende Hotkeys gibt es:

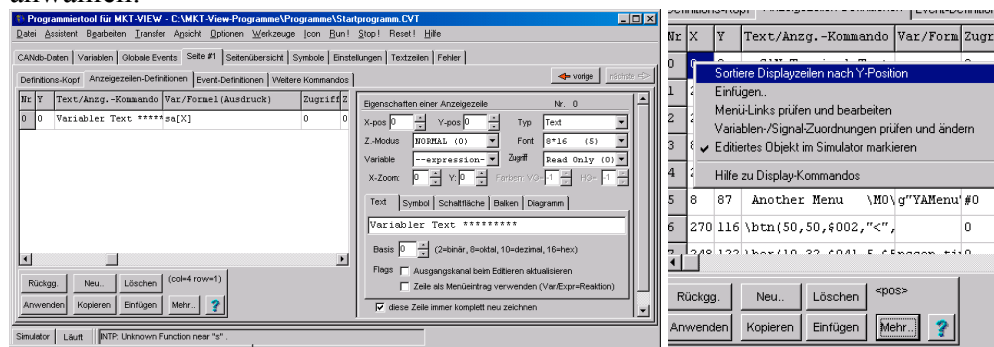
- „F1“ oberste Funktionstaste
- „F2“ zweite Funktionstaste
- „F3“ dritte Funktionstaste

Nachdem Sie alle Menütexte eingegeben haben und die entsprechenden Reaktionen programmiert haben, müssen Sie die Reihenfolge der Anwahl durch den Drehgeber festlegen.

Methode der Anwahl eines Menüpunktes:

- Durch rechts Drehen des Encoderdrehknopfes wird der nächste Menüpunkt selektiert
- Durch links Drehen des Encoderdrehknopfes wird der vorherige Menüpunkt selektiert
- Durch das Drücken des Encoderdrehknopfes wird die programmierte Reaktion ausgelöst

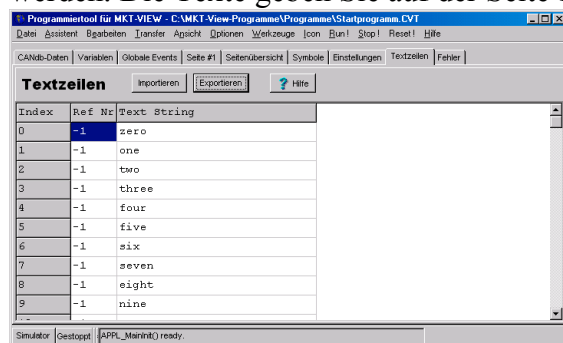
Die Reihenfolge wird durch die Reihenfolge der Eingabe festgelegt. Sie sollten, nachdem Sie alle Menüeinträge eingegeben haben, den *Mehr* Button betätigen und folgenden Menüpunkt anwählen:



Damit die Menüpunkte sortiert werden. (??)

Eingabe von Texten mit Platzhaltern für variable Texte:

Mit dem MKT-View+ können in Abhängigkeit eines Variableninhaltes Texte eingeblendet werden. Die Texte geben Sie auf der Seite *Textzeilen* ein:



Die Zuordnung der Texte zum Variableninhalt kann auf zwei verschiedenen Arten durchgeführt werden:

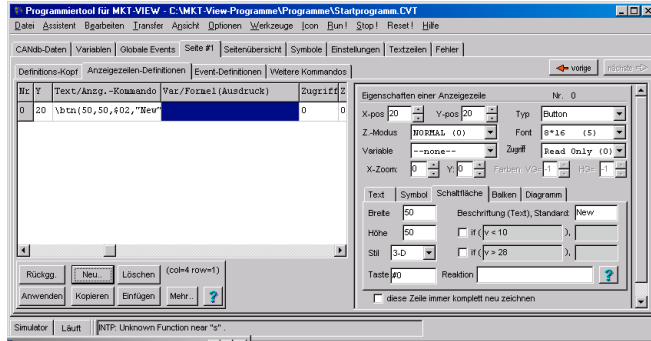
- Durch den Befehl „sa[Variablenname]“. Bei dieser Methode wird der Text genommen, der hinter dem entsprechenden Index steht.
- Durch den Befehl „sr[Variablenname]“. Bei dieser Methode wird der Text genommen, der hinter der entsprechenden *Ref.Nr.* steht

Zur Eingabe dieser Befehle muss in dem Feld *Variable* die Bezeichnung „expression“ stehen. Den Befehl müssen Sie dann in die Tabellenspalte *Var/Formel (Ausdruck)* eintragen.

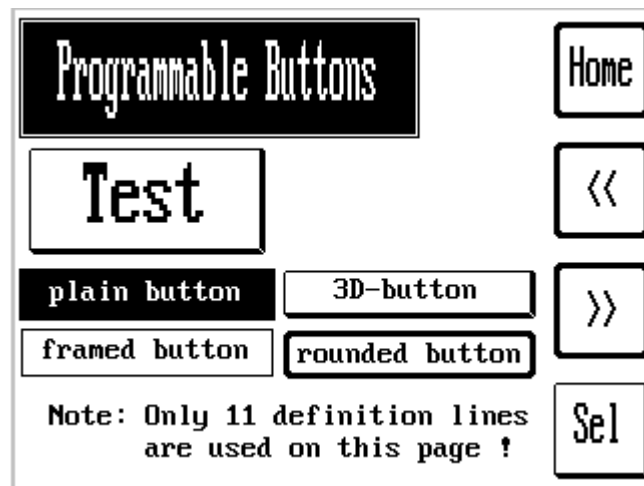
Als letztes müssen Sie noch das Flag *diese Zeile immer komplett neu zeichnen* anwählen

3.4.2.4.2.2 Eingabe von Buttons

Für die Bedienung des MKT-View+ können die Funktionstasten benutzt werden. Um diese zu beschriften gibt es die Möglichkeit, Buttons zu programmieren.



Graphische Buttons können folgendermaßen aussehen:



Ein 'hotkey' zur Aktivierung kann für jeden Button definiert werden. Normalerweise wird dazu eine der Funktionstasten in der Nähe des graphischen Buttons verwendet. Wenn der User die Taste drückt, wird der Button sofort auf dem Bildschirm invertiert, und die entsprechende Reaktionsmethode abgearbeitet.

Jeder Button hat bestimmte Eigenschaften, die auf dem folgenden Panel des Programmier-tools bearbeitet werden können:

Eigenschaften wie *x-pos*, *y-pos*, *Draw Mode*, usw. werden im allgemeinen Handbuch erklärt. Sie beziehen sich auch auf normale Textanzeigen. Hier werden nur die spezifischen Eigenschaften eines Buttons vorgestellt (in der unteren Hälfte des oben abgebildeten Panels).

Spezifische Eigenschaften eines Buttons :

- *Breite* und *Höhe* eines einzelnen Buttons, gemessen in Bildschirmpixeln
- *Stil* : „plain“=einfach, „framed“=mit Rahmen, 3-D, oder „rounded“= mit abgerundeten Ecken.
Die verschiedenen Button-Stile sind im oben abgebildeten Beispiel erkennbar.
- *Beschriftung (Text), Standard*:
Der Standard-Text, der auf der Schaltfläche erscheinen soll. Der Text wird innerhalb der Button-Fläche mittig ausgerichtet, Sie müssen allerdings selbst sicherstellen, dass der Text wirklich in die Fläche passt. Ist der Platz zu klein, können Sie einen kleineren Zeichensatz („Font“) verwenden.
- *if (XXX), ..*
Definiert einen alternativen Button-Text, um "besondere Betriebszustände" anzuzeigen. Ein Beispiel dazu finden Sie unten.
- *Taste*
Hiermit definieren Sie, welcher "hotkey" (Taste) den Button aktivieren soll. Sie können den Namen der Taste (z.B. F1) in dem Edit-Feld eingeben, oder per Doppelklick eine Taste aus einer Liste auswählen.
- *Reaktion*
Definieren Sie hier was passieren soll, wenn der Button aktiviert wird (per Tastendruck bzw. Druck auf die Schaltfläche). Die Reaktion wird oftmals ein goto-Befehl sein, der zu einer anderen Anzeigeseite umschaltet. Es kann allerdings auch ein beliebiges anderes Interpreterkommando in dieses Feld eingegeben werden. Die Befehlssyntax ist die gleiche wie bei den „Events“. Per Doppelklick in das Editierfeld können Sie eine Reaktions-Definition interaktiv erstellen.

3.4.2.4.2.3 Interpreterkommandos zum Steuern der Buttons auf einer Anzeigeseite

Normalerweise brauchen Sie keinen eigenen Interpretercode zum Steuern der graphischen Buttons zu schreiben. Jeder Button hat normalerweise einen Hotkey-Code, und wenn der Bediener die entsprechende Taste drückt, wird der graphische Button zunächst invertiert und dann die entsprechende Reaktionsmethode ausgeführt (meistens ein `goto`-Befehl).

Die graphischen Buttons lassen sich allerdings auch für Sonderzwecke einsetzen. Dazu können sie per Interpreter gesteuert werden. Es wird hier nicht auf die Details des im Terminal eingebauten Interpreters eingegangen. Die Beschreibung finden Sie in einem anderen Kapitel. Hier lediglich die Interpreterfunktionen und -Kommandos zum Steuern der graphischen Buttons:

bs (Button state)

Liest oder setzt den aktuellen Button-Zustands-Code (Button state). Dies ist ein numerischer Wert zwischen 0 und 255. Null bedeutet, dass momentan kein Button selektiert ist ("selektiert" = i.A. "auf dem Display invers gezeichnet"). Andernfalls enthält `,bs'` den HOTKEY-WERT des Buttons, der momentan selektiert ist.

Wenn Sie per Interpreteranweisung einen neuen Wert an `,bs'` zuweisen (z.B. `„bs=13“`), wird der Button mit dem hotkey-code 13 (bzw. `„ENTER“`) invers dargestellt, alle anderen Buttons normal. Der Effekt ist hier fast der Gleiche wie wenn der Bediener die ENTER-Taste drückt.

ABER: eine Wertzuweisung an `,bs'` ändert nur die graphische Anzeige, es wird aber nicht die Reaktionsmethode des Buttons abgearbeitet (vgl. `,bk'`, siehe unten).

bk (Button key)

Sendet einen Tastencode an die graphischen Buttons auf der aktuellen Seite. Kann verwendet werden, um die Reaktionsmethode eines Buttons abzuarbeiten. Verwenden Sie eine formale Zuweisung (wie `„bk=13“`). Der Effekt ist derselbe wie beim Drücken der dem Button zugeordneten Taste ("hotkey", z. B. die ENTER-Taste für Button mit dem hotkey-code 13). Kurz bevor die Reaktion des Buttons abgearbeitet wird, wird der Button auf dem Display zwecks „user-feedback“ invers dargestellt.

Wozu kann dies benutzt werden?

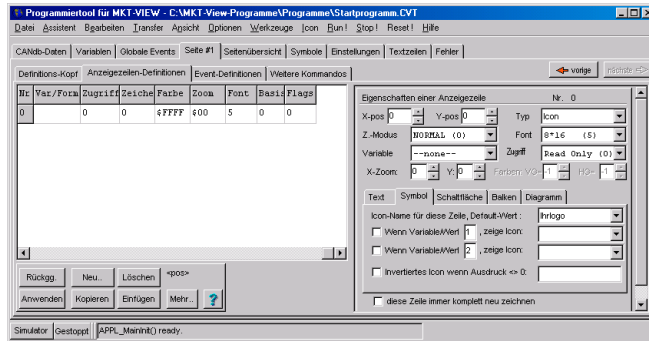
Z.B. für programmgesteuerte „Radio“-Buttons (Windows-Terminologie). Bei Radio-Buttons kann nur einer aus N Buttons `,AN'` sein, alle anderen Buttons sind `,AUS'` wie bei den Stationstasten des guten alten Dampf radios. Verwenden Sie die Funktion `,bs'`, um zu testen, welcher Button momentan „AN“ ist, oder das Kommando `'bk'`, um genau EINEN Button einzuschalten und alle anderen abzuschalten.

(Hinweis: Die „hotkeys“ müssen keine existierenden Tastencodes sein, falls die Buttons nicht vom Bediener umgeschaltet werden müssen; sondern z.B. programmgesteuert in Abhängigkeit des Wertes einer Variablen. Ein Beispiel dazu findet sich in der Anzeigeseitengalerie, Datei `„GraphicButtonDemo“`).

3.4.2.4.2.4 Eingabe eines Icons

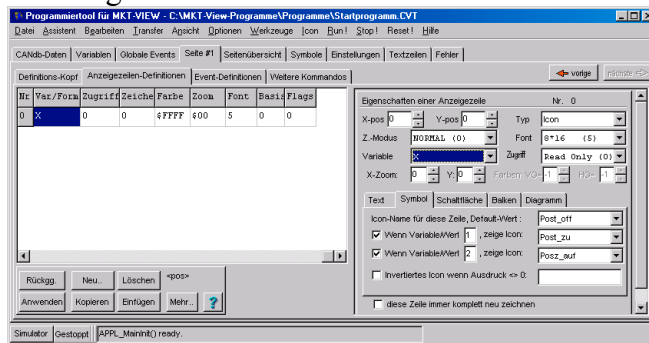
Abilden eines einfachen Icons

Programmierhandbuch MKT-View-Terminal Version 1.01



Auf jeder Seite lassen sich Icons abbilden. Bevor Sie ein Icon einblenden können, müssen Sie es in das Programm importieren (siehe Menüpunkt *Icon*).

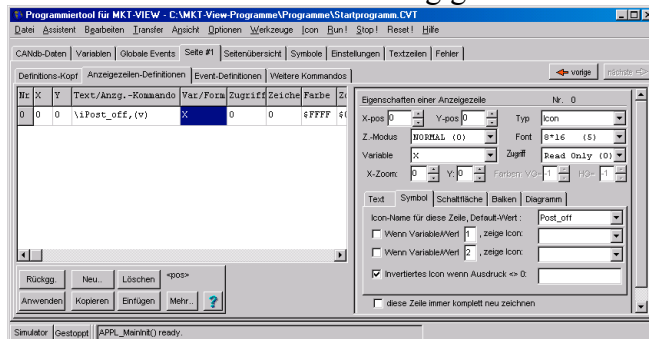
Abbilden verschiedener Icons in Abhängigkeit einer Variablen. Sie haben des Weiteren die Möglichkeit, verschiedene Icons in Abhängigkeit einer Variablen anzuzeigen:



Die Bedingung, die hiermit programmiert wird, heißt:

Wenn $X==1$ ist, wird das ,Icon Post_zu' angezeigt, wenn $X==2$ wird das ,Icon Post_auf' angezeigt, sonst wird das ,Icon Post_off' angezeigt.

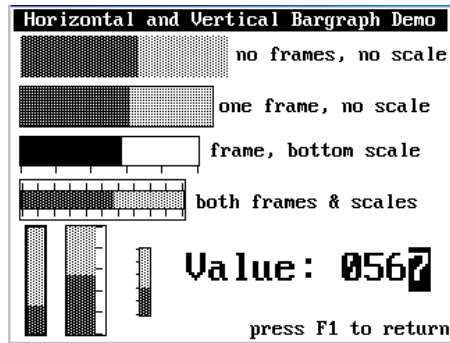
Invertieren eines Icons in Abhängigkeit einer Variablen



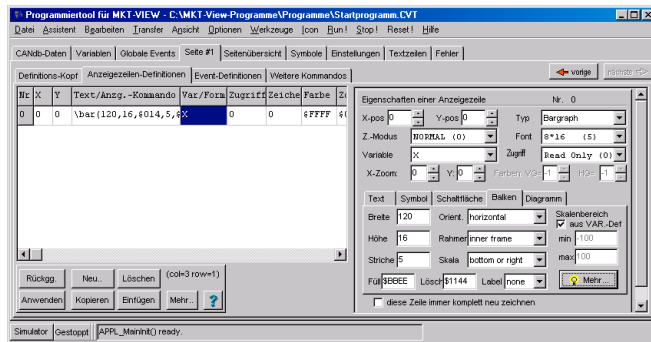
Das Icon wird automatisch invertiert, wenn $x \neq 0$ ist

3.4.2.4.2.5 Eingabe eines Bargrafen

Bargrafen können folgendermaßen aussehen:

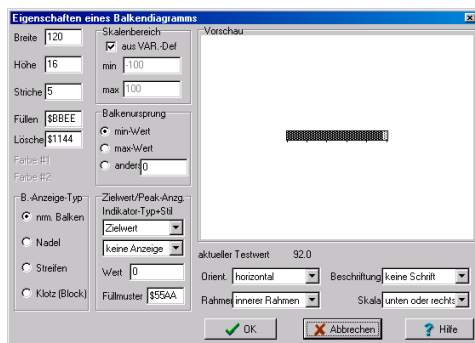


Ein Bargraf wird durch bestimmte Eigenschaften (properties) charakterisiert, die auf der Registerkarte *Anzeigezeilen-Definitionen* des Programmiertools eingestellt werden können. Weitere Optionen können nach Klicken auf den *Mehr*-Button in einem weiteren Fenster angezeigt werden. Variablenwerte können auch als Bargraf abgebildet werden. Die Eingabe erfolgt über den Button *Neu*.



In diesem Fenster geben Sie die x- und y-Positionen ein, weisen dem Bargraf eine Variable zu und legen fest, ob für den Skalenbereich die min- und max-Werte der Variablen benutzt werden sollen.

Da es viele Darstellungsmöglichkeiten für einen Bargrafen gibt, kann mit dem Button *Mehr* ein zusätzliches Fenster geöffnet werden:



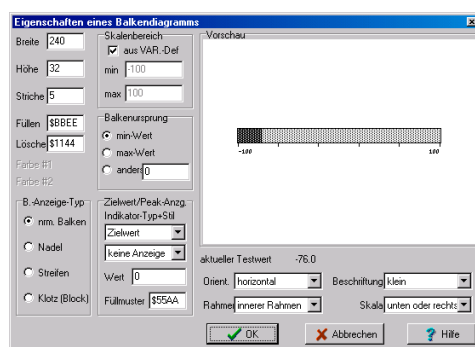
In diesem Fenster können Sie die restlichen Definitionen durchführen:

Programmierhandbuch MKT-View-Terminal Version 1.01

- *Breite und Höhe* des Bargrafen
- Anzahl der Teilstriche (*Striche*)des Bargrafen
- *Skalenbereich*: Sie können anwählen, ob die min- und max-Werte der Variablen als Skalenbereich genutzt werden sollen, oder wenn Sie ein bestimmter Bereich der Variable interessiert, können Sie den entsprechenden Bereich manuell eingeben (Spreizen)
- Sie können das Füll und Löschemuster verändern
- Sie können den Balkenursprung verändern:
 - *min-Wert*: Der Balken wird aus der Richtung des min-Wertes gefüllt
 - *max-Wert*: Der Balken wird aus der Richtung des max-Wertes gefüllt
 - *andere*: Werte hier können Sie eine Zahl zwischen den min- und max-Wert eingeben (meistens 0) jetzt läuft der Balken von rechts nach links wenn der Wert kleiner als 0 ist oder von links nach rechts wenn der Wert größer als 0 ist
- Sie können den Anzeigetyp festlegen (*B.-Anzeige-Typ*)
- Sie können einen Zielwert programmieren und in verschiedener Art und Weise anzeigen lassen (*Zielwert/Peak-Anzg.*)
- Sie können die Orientierung definieren horizontal oder vertikal (*Orient.*)
- Sie können verschiedene Rahmentypen festlegen (*Rahmen*)
- Sie können eine Beschriftung des Bargrafen anwählen (*Beschriftung*)

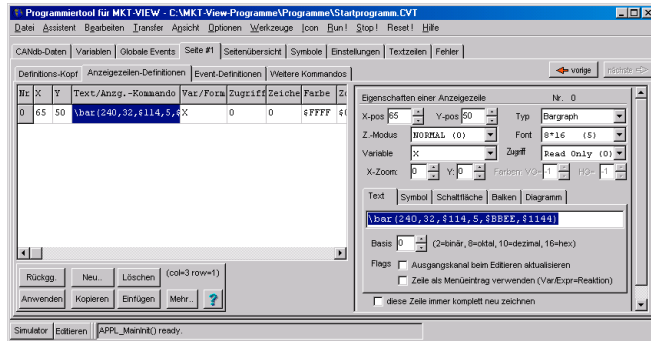
Kombinieren eines Bargrafen mit einem vorlaufenden Text + numerischer Anzeige

Sie können einen Bargraf auch mit einem Text kombinieren. Dafür geben Sie als erstes den Bargrafen ein:

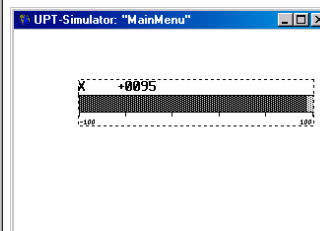
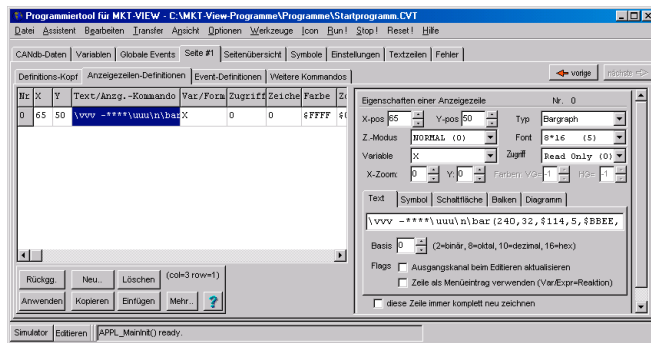


Danach wählen Sie die das Karteiblatt „Text“ aus:

Programmierhandbuch MKT-View-Terminal Version 1.01



und geben dann den gewünschten Text vor dem \bar ein



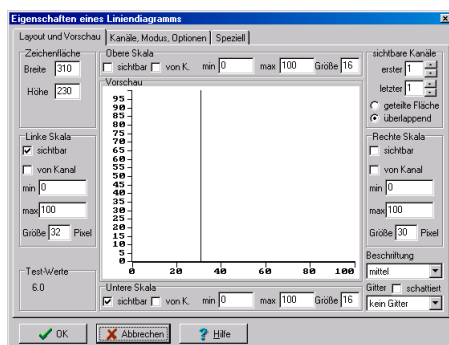
als Zeichen für eine neue Zeile zwischen dem Text und den Bargrafen gilt \n

Dieser Text/Bargraf lässt sich wieder gut kopieren und wird durch eine einfache Neuordnung der Variable angepasst.

3.4.2.4.2.6 Eingabe eines y(t) bzw. y(x) Diagramms

Auf dem MKT-View+ können max. 8 Diagrammkanäle genutzt werden. Auf jeder Seite kann max. ein Diagramm abgebildet werden. Die Kanäle können auf verschiedene Seiten aufgeteilt werden. Es gilt dann auf allen Seiten die gleiche Zeitbasis.

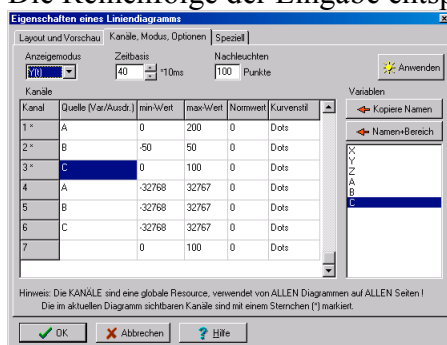
Erstellung eines y(t) Diagramms



Programmierhandbuch MKT-View-Terminal Version 1.01

Zur Programmierung des Diagramms wird ein separates Fenster geöffnet, in dem die Programmierung durchgeführt wird.

1. Festlegung der Diagrammgröße
2. Als nächstes wird die Anzahl der Kanäle festgelegt durch die Angabe der sichtbaren Kanäle erster bzw. letzter sichtbarer Kanal.
3. Als nächstes bestimmen Sie, ob alle Kanäle in einem Diagramm dargestellt werden sollen oder ob jeder Kanal ein separates Diagramm zugewiesen bekommen soll.
4. Jetzt weisen Sie jedem Kanal eine Variable zu
Die Reihenfolge der Eingabe entspricht der Reihenfolge der Abbildung



5. Jetzt wird die Zeitbasis für die x-Achse bestimmt. **Die Punkte 4 und 5 müssen zwingend durch eine Betätigung des Buttons *Anwenden* bestätigt werden**

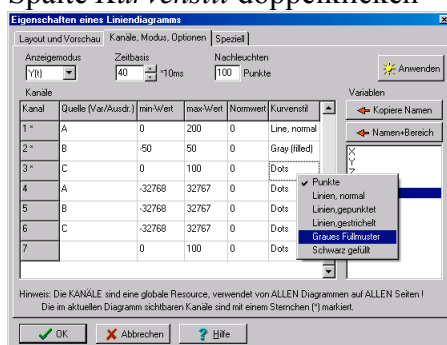
Die Position des Diagramms können Sie zurzeit nur durch die Änderung der x- und y-Koordinaten in der linken Tabelle.

Vorgehensweise:

1. in der linken Tabelle die erste Zeile, in der \dia steht, selektieren
2. dann die neue Position über die x- bzw. y-Position einstellen

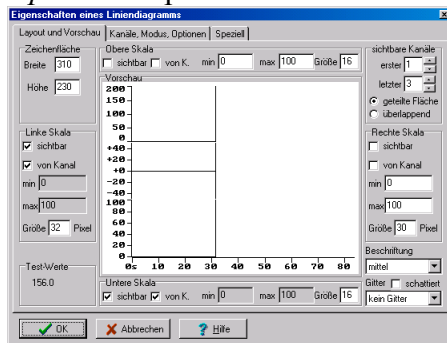
Eine andere Methode ist die, dass sie die obere linke Ecke des Diagramms mit der Maus verschieben

- Als nächstes bestimmen Sie den Kurvenstil, in dem Sie in das entsprechende Feld der Spalte *Kurvenstil* doppelklicken



Programmierhandbuch MKT-View-Terminal Version 1.01

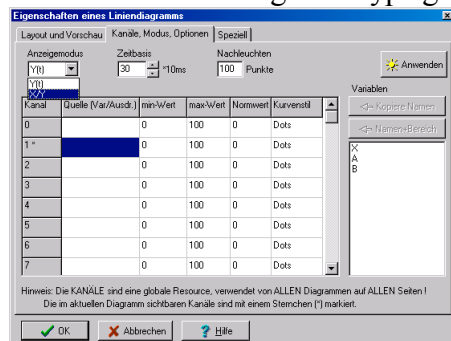
- Mit der nächsten Eingabe bestimmen Sie, ob die Skaleneinteilung von den min- und max-Werten der Variablen abgeleitet werden soll. Wenn Sie eine Spreizung des Bereiches durchführen möchten, können Sie die min und max Werte auf der Seite *Kanäle, Modus, Optionen*, *Optionen* anpassen



Zusätzlich können Sie die Größen der Skalenbereiche anpassen und bestimmen, ob diese abgebildet werden sollen.

Erstellung eines $y(x)$ Diagramms

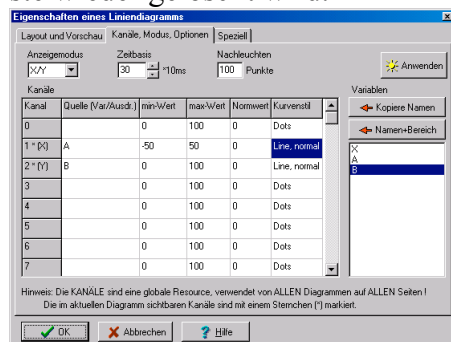
Die Anwahl des Diagrammtyps geschieht auf der *Kanäle, Modus Optionen*- Karteiseite



Die Anzahl der sichtbaren Kanäle muss 2 sein (letzter Kanal – erster Kanal = 1).

Der Kurvenstil der beiden Kanäle muss Linie normal sein

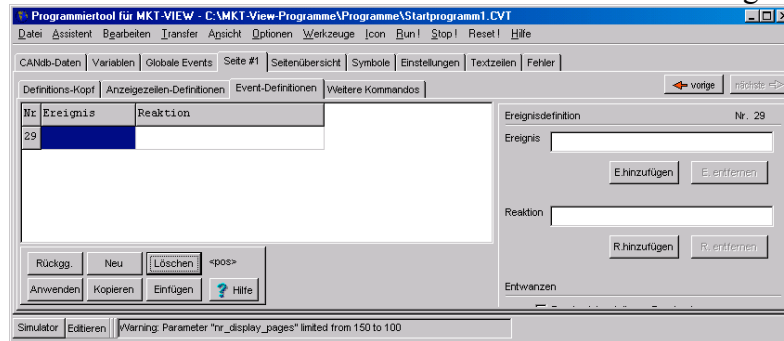
Mit der Eingabe Nachleuchten geben Sie vor, nach wie viel gezeichneten Elementen das älteste wieder gelöscht wird.



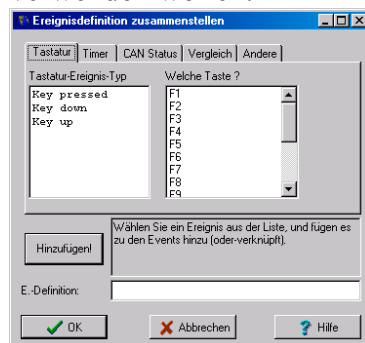
3.4.2.4.3 Beschreibung der Event-Definitionen einer Seite

Sie können jeder Seite eigene „Seitenevents“ zuordnen. Diese werden dann nur ausgeführt solange diese Seite abgebildet wird.

Über den Button *Neu* müssen Sie als erstes eine Eventeingabezeile erzeugen:



Durch die Betätigung des Buttons *Einfügen* können Sie dann festlegen, welchen Event Sie verwenden wollen:



Folgende Eventdefinitionen gibt es:

3.4.2.4.3.1 Tastaturevents

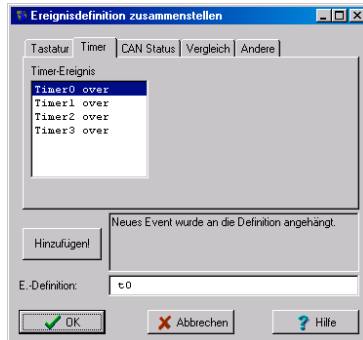
Es gibt folgende Tastaturevents

1. Key pressed – Taste wurde betätigt. Diese Eingabe wird aus dem Tastaturpuffer abgearbeitet und sollte vorrangig für die Abfrage einer Taste benutzt werden. Dieses Ereignis wird nur einmal pro Drücken der Taste ausgeführt.
2. Key0 down – Taste ist gedrückt. Das Ereignis wird, so lange die Taste gedrückt ist und so oft wie das Terminal die Reaktion ausführen kann, ausgeführt
3. Key up – Taste ist nicht gedrückt. Das Ereignis wird, so lange die Taste losgelassen ist und so oft wie das Terminal die Reaktion ausführen kann, ausgeführt

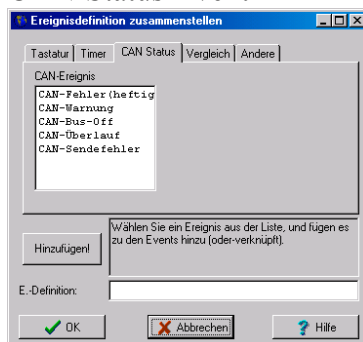
Nachdem Sie den Typ des Tastaturevents ausgesucht haben, ordnen Sie diesem eine Funktionstaste zu. Beim MKT-View+ die Tasten F1, F2, F3

3.4.2.4.3.2 Timerevents

Das MKT-View+ hat vier programmierbare Timer, die als Rückwärtszähler ausgelegt sind. Die Zeitbasis ist 100 ms. Die Timer lösen ein Ereignis aus, wenn sie auf 0 angekommen sind.



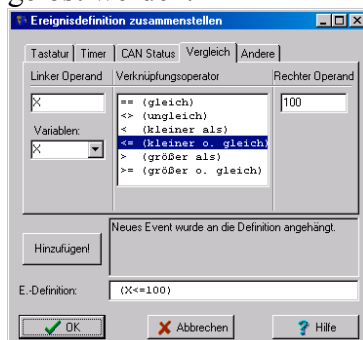
CAN-Status-Event



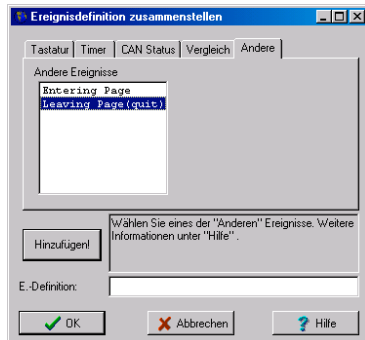
Es können auch auf verschiedene CAN-Bus-Zustände Events ausgelöst werden. Sinnvoll nur als Globale Events.

3.4.2.4.3.3 Event durch Wertevergleich

Es können Ereignisse durch Variablenvergleiche mit Konstanten oder anderen Variablen ausgelöst werden.



3.4.2.4.3.4 Andere Ereignisse



Andere Events können sein:

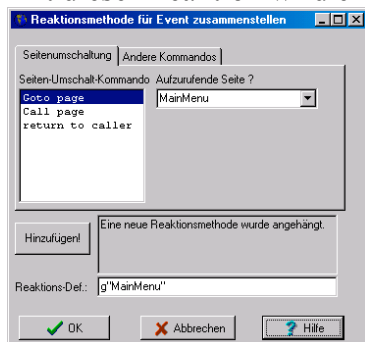
- Entering Page: Das Ereignis wird nur beim Aufruf der Seite einmal ausgelöst.
- Leaving Page: Das Ereignis wird nur einmal beim Verlassen der Seite ausgelöst

Beschreibung der Reaktionen

Als Reaktion können verschiedene Interpreterkommandos ausgelöst werden.

Reaktion „goto Page“

Mit dieser Reaktion wird eine neue Seite aufgerufen



Reaktion „call Page“

Mit dieser Reaktion wird ebenfalls eine neue Seite aufgerufen. Der Unterschied zur goto-Reaktion liegt darin, dass man bei der call-Reaktion mit der „return“ Anweisung zu der Seite zurückspringt, von der diese aufgerufen wurde. Dies kann genutzt werden, wenn von verschiedenen Seiten eine andere Seite aufgerufen werden soll, und der Rücksprung wieder auf die aufrufende Seite erfolgen soll.

3.4.2.4.4 Beschreibung der „weiteren Kommandos“

In der Tabelle der Karteikartenseite *Weitere Kommandos* können Sonderbefehle eingetragen werden, wobei mehrere Befehle in einer Zeile geschrieben werden können. Als Trennzeichen müssen sie den „:“ –Doppelpunkt verwenden. Sie können jede der Koordinaten durch eine Variable ersetzen und sich so z.B. eine eigene Bargrafen-Funktion generieren. Einen Auszug der Befehle liefert die folgende Aufzählung.

3.4.2.4.4.1.1 Das LINE-Kommando (li)

Dieses Grafikkommando zeichnet eine dünne Linie auf dem Grafikbildschirm.

Syntax:

li(<x1>, <y1>, <x2>, <y2>, <width>)

Zeichnet eine Linie mit der angegebenen Stärke („width“ gemessen in Pixel).

Hinweis: Auch Grafik-Linien können per Backslash-Sequenz im Format-String einer normalen Display-Zeile angezeigt werden. Im Gegensatz zum Interpreter-Befehl „li“ sind die Koordinaten dann ‚relativ‘ zur Koordinate der Display-Zeile.

3.4.2.4.4.1.2 Das PIXEL-Kommando (pi)

Dieses Kommando setzt ein einzelnes Pixel auf dem Grafikbildschirm.

Syntax:

pi(<x>, <y>)

3.4.2.4.4.1.3 Das FRAME-Kommando (fr)

Dieses Grafikkommando zeichnet einen rechteckigen Rahmen.

Das Innere des Rahmens wird nicht beeinflusst (es wird weder gelöscht noch gefüllt).

Syntax:

fr(<x1>, <y1>, <x2>, <y2>)

darin ist

x1 = linker Rand

y1 = oberer Rand

x2 = rechter Rand

y2 = unterer Rand

Hinweis: Auch rechteckige Rahmen können mittlerweile als Backslash-Sequenz innerhalb einer „normalen“ Display-Zeile erzeugt werden.

3.4.2.4.4.1.4 Das FillRectangle-Kommando (fi)

Füllt einen rechteckigen Bereich des Bildschirms.

Syntax:

fi(<x1>, <y1>, <x2>, <y2> [, <pattern0> , <pattern1>])

darin ist

x1=left border, y1=top, x2=right border, y2=bottom,

pattern1, pattern2 = optional "filling" pattern, default=0xFF

Der alte Inhalt der gefüllten Fläche wird überschrieben, außer in einem Spezialfall: Bei Zeichenmodi, die nicht in beide alternierenden Display-„Pages“ schreiben, bleibt der alte Inhalt während einer der beiden Blink-Phasen erhalten. Dies eignet sich z.B. als Warnmarkierung (sparsam einsetzen!).

Das optionale Füllmuster ermöglicht auch bei monochromen Displays, verschiedene Schattierungen zu realisieren. Beide Füllmuster sind 8-Bit-Masken; ein "1"-Bit entspricht einem gesetzten, ein "0"-Bit einem gelöschten Pixel. <Pattern0> wird in allen Grafikzeilen mit grader Y-Koordinate verwendet, <Pattern1> für alle ungraden Zeilen. Einige Beispiele für "sinnvolle" Füllmuster:

fi(10,10,30,20, 0xFF, 0xFF) zeichnet ein "massives schwarzes" Rechteck

fi(10,22,30,30, 0xFF, 0x00) zeichnet ein Rechteck aus dünnen horizontalen Linien

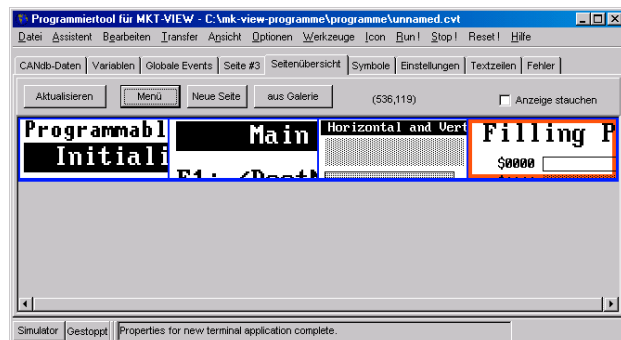
fi(10,32,30,40, 0x55, 0x55) zeichnet ein Rechteck aus dünnen vertikalen Linien

fi(10,42,30,50, 0x55, 0xAA) zeichnet ein grau schattiertes Rechteck (50 Prozent Schwärzung)

Wird kein Füllmuster angegeben, zeichnet der Interpreter ein schwarz gefülltes Rechteck (wie mit dem Füllmuster 0xFF, 0xFF).

3.4.2.5 Beschreibung der Karteikartenseite *Seitenübersicht*

Auf dieser Seite werden Ihnen alle Seiten Ihres Programms als Übersicht angezeigt.

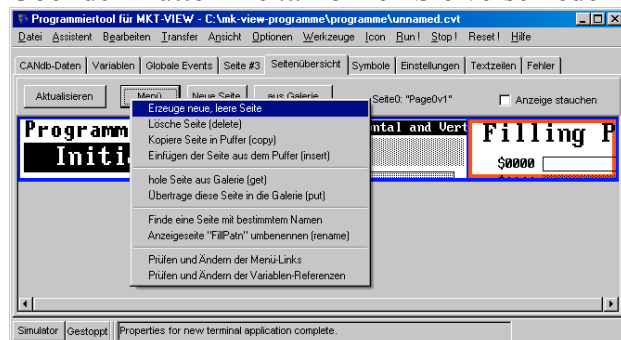


Die Seite, die Sie aktuell bearbeiten, wird mit einem roten Rand versehen. Durch ein Doppelklicken auf eine andere Seite können Sie diese Seite auswählen.

Aus Zeitgründen werden nicht alle Seiten immer neu aktualisiert, deshalb müssen Sie dies über den Button *Aktualisieren* bei Bedarf manuell durchführen.

Auf Grund der Übersichtlichkeit werden nur die oberen 128x64 Pixel dargestellt. Sie können über das Flag *Anzeige stauchen* die Auflösung auf 320x240 Pixel ändern

Über den Button *Menü* können Sie verschiedene Vorgänge auslösen:



Neue Seite einfügen

Jeder neuen Seite sollten Sie möglichst gleich einen eindeutigen und *sprechenden* Namen geben, um bei der Programmierung der Seitenumschaltung eine eindeutige Auswahl treffen zu können.

Eine Seite löschen

Die selektierte Seite wird gelöscht

Eine Seite in den Zwischenpuffer kopieren

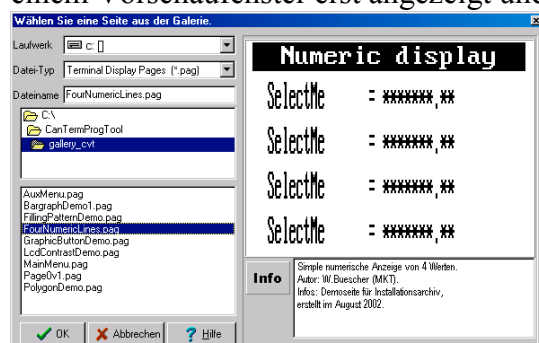
Wenn Sie eine ähnliche Seite aus Ihrem Programm noch mal benötigen, können Sie diese in den Puffer kopieren und sie dann mit dem Menüpunkt „Einfügen einer Seite aus dem Puffer“ in ihrem Applikation verdoppeln.

Eine Seite aus dem Zwischenpuffer einfügen

siehe Punkt davor

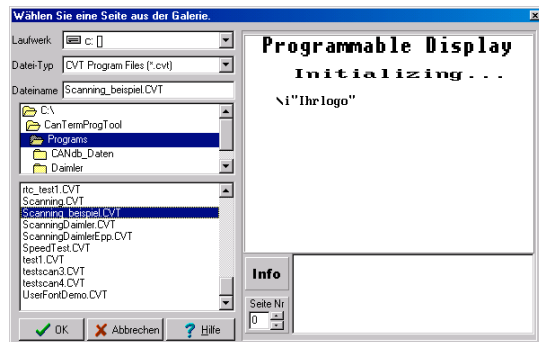
Eine Seite aus der Galerie einzufügen

Der Dateityp muss dafür auf *.pag stehen. Sie haben die Möglichkeit, Musterseiten aus einer Galerie zu laden und diese in Ihrem Programm zu integrieren. Die Seite wird Ihnen dann in einem Vorschaufenster erst angezeigt und kurz beschrieben.



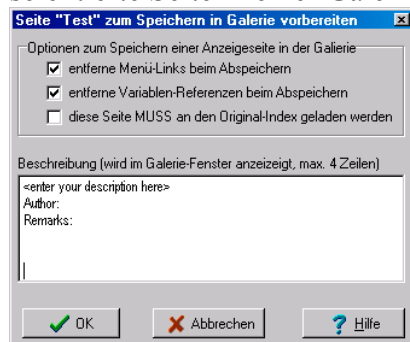
Eine Seite aus einem vorhandenen Programm einfügen

Der Dateityp muss dafür auf *.csv stehen. Sie haben die Möglichkeit Musterseiten aus einem vorhandenen Programm zu laden und diese in Ihrem Programm zu integrieren. Die Seite wird Ihnen dann in einem Vorschaufenster erst angezeigt.



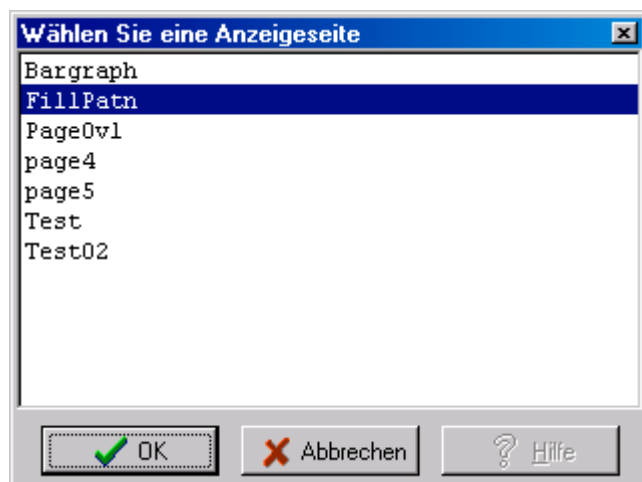
Eine erstellte Seite in die Galerie übertragen

Wenn Sie eine Seite auch in anderen Programmen nutzen möchten, können Sie die vorher selektierte Seite in einer Galerie speichern



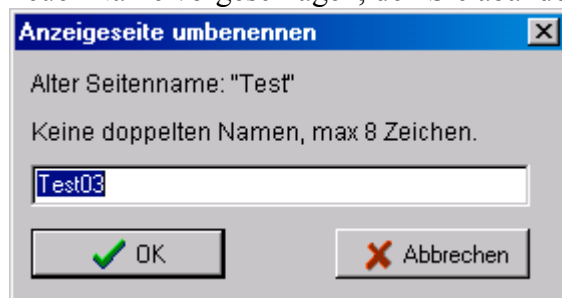
Dabei sollten alle Menü-Links und Variablen-Referenzen entfernt werden. Vor dem Abspeichern sollten Sie eine kurze Beschreibung dieser Seite eingeben!

Eine Seite finden und anwählen

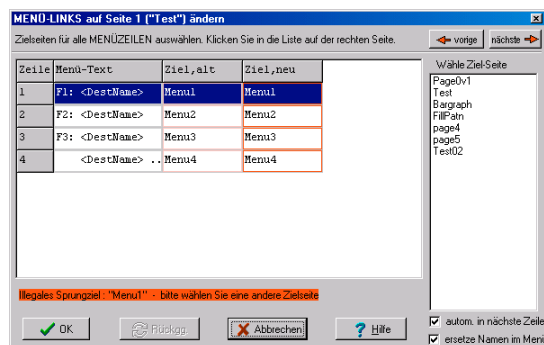


Eine Seite umbenennen

Einer Seite einen neuen Namen geben (max. 8 Zeichen). Nach der Auswahl wird Ihnen ein neuer Name vorgeschlagen, den Sie abändern können.



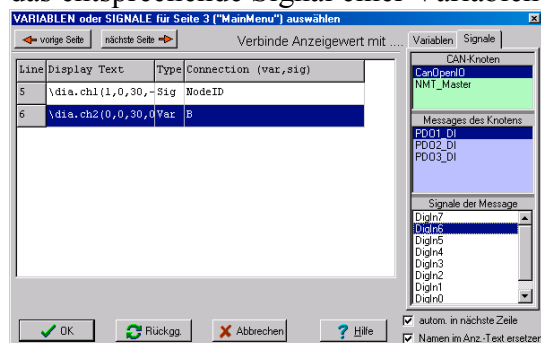
Die Menü-Links prüfen oder ändern



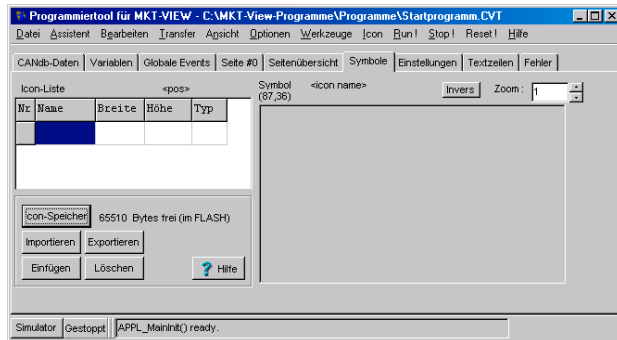
Die einzelnen Seiten sind mit Links untereinander verlinkt (z.B. Aufruf einer anderen Seite über Tastenbetätigung). Diese Links müssen bei Seiten, die Sie aus der Galerie oder aus vorhandenen Seiten eingefügt haben, neu vergeben werden. Um Ihnen dies zu vereinfachen, werden Ihnen die möglichen Ziele links in einer Liste zur Auswahl angezeigt.

Prüfen und Ändern der Variablen-Referenzen

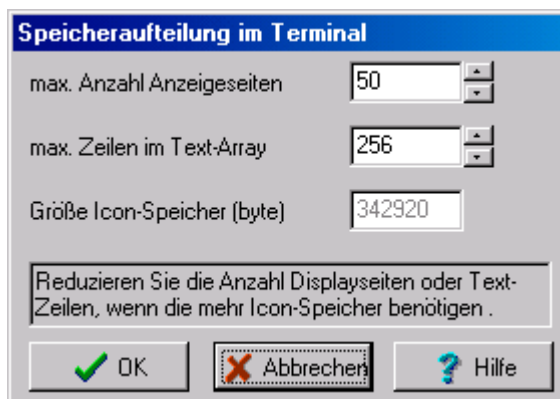
Es werden alle Seiten überprüft, ob den auf der Seite integrierten Variablen in den Texten ein gültiger Variablenname zugeordnet ist. Wenn nicht, wird Ihnen automatisch auf einer Seite eine Vorschlagsliste angeboten. Aus dieser können Sie sich das passende Signal bzw., wenn das entsprechende Signal einer Variablen zugeordnet ist, eine passende Variable aussuchen.



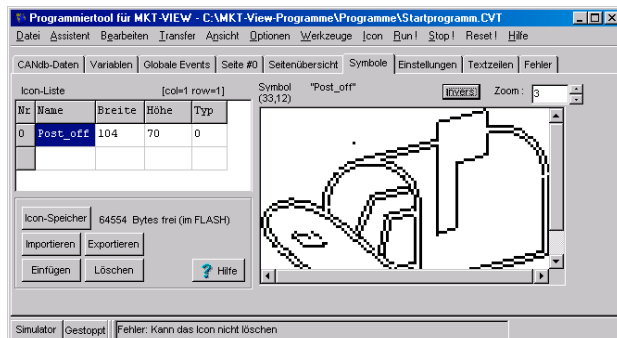
3.4.2.6 Karteikartenseite „Symbole“



Auf dieser Seite wird Ihnen eine Übersicht der importierten Icons angezeigt. Es wird Ihnen angezeigt, wie viel Speicherplatz noch für weitere Icons zur Verfügung steht. Sie können über den Button Icon-Speicher den reservierten Speicherplatz für die Icons ändern, in dem Sie die Anzahl der Blätter oder Zeilen des Text-Arrays ändern.



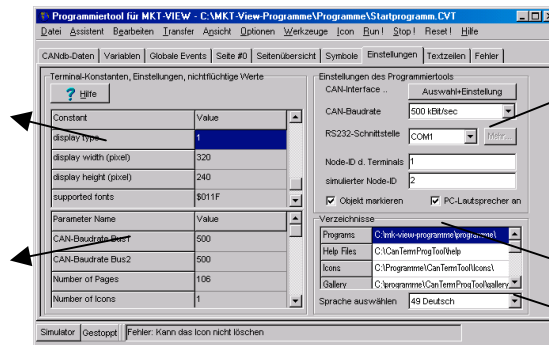
Zusätzlich haben Sie die Möglichkeit, neue Icons zu importieren bzw. zu exportieren. Über den Button *invers* haben Sie die Möglichkeit, das Icon zu invertieren. Des Weiteren können Sie in dem linken Anzeigefenster das Icon nachbearbeiten, in dem Sie mit der Maus einzelne Pixel invertieren bzw. löschen. Um dies einfacher zu gestalten, können Sie das Icon vorher über die Zoomfunktion vergrößern.



3.4.2.7 Karteikartenseite „Einstellungen“

Terminal Konstanten
Tabelle

Parameter-Tabelle



Einstellung der PC-Programmiertool-Schnittstellen

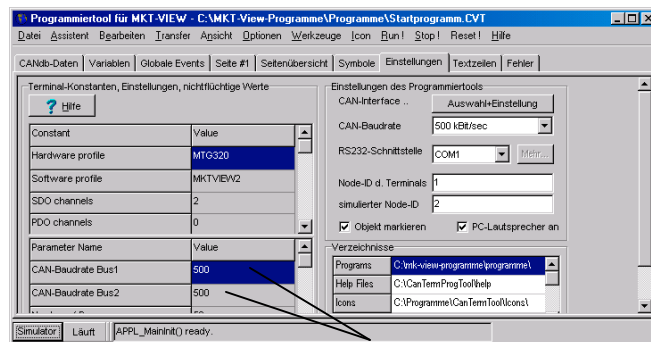
Verzeichnisdefinitionen
Sprachauswahl

3.4.2.7.1 Terminal - Konstanten

Die Konstanten in dieser Tabelle dienen in der Regel nur zur Information und sind bis auf Ausnahmen nicht änderbar. Bitte keine Änderungen der editierbaren Konstanten durchführen (nur in Absprache mit MKT).

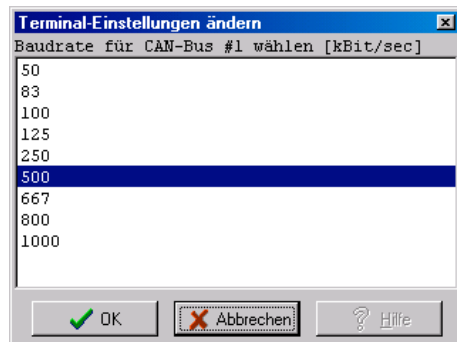
3.4.2.7.2 Parametertabelle

3.4.2.7.2.1 Baudrateneinstellung des Terminals



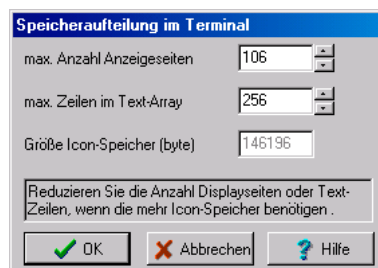
Einstellen der Baudrate für die CAN-Schnittstellen.

- Die *CAN-Baudrate* unter *Einstellungen des Programmiertools* wird für die Programmierung des Terminals über die CAN-Schnittstelle verwendet
- Die *CAN-Baudrate Bus1* und *CAN-Baudrate Bus2* unter *Terminal-Konstanten, Einstellungen, nichtflüchtige Werte* wird für den „Normalbetrieb“ des Terminals verwendet. Durch anklicken des entsprechenden Eintrages wird die Seite *Terminal-Einstellungen ändern* geöffnet, auf der die benötigte Baudrate ausgewählt werden kann



3.4.2.7.2.2 Max. Seitenanzahl eines Programms

Da der Speicherplatz im MKT-View+ begrenzt ist, können Sie festlegen, wie viele Seiten und wie viele Textzeilen Sie in Ihrem Terminal benötigen. Den Rest des Speichers können Sie dann für die Abbildung von Icons nutzen. Die max. Seitenanzahl beträgt 149, wenn Sie keine Textarrays und Icons nutzen.

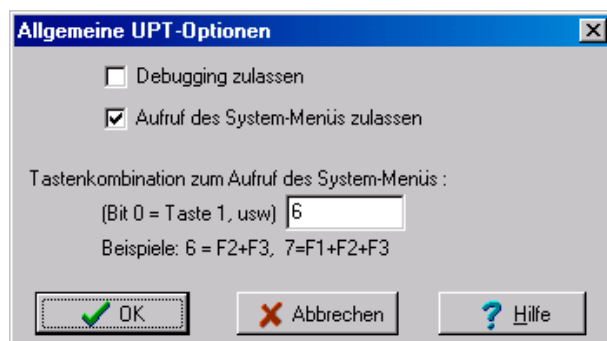


Den Eintrag *Größe Icon-Speicher (byte)* können Sie nicht ändern, er wird automatisch bestimmt.

3.4.2.7.2.3 Max. Textarrayeinträge

Bedienung siehe vorigen Punkt. Die max. Anzahl beträgt 1024 Texte mit einer max. Länge von 26 Zeichen.

3.4.2.7.2.4 Tastenkombination zum Aufruf des Systemmenüs (im Terminal)



In dem Systemmenü des Terminals lassen sich verschiedene Tests und Einstellungen direkt durchführen. Mit den Einstellungen *General Options* und *Key code for System Menu* können Sie festlegen, ob das Systemmenü vom Benutzer aufgerufen werden kann und wenn ja, mit welcher Tastenkombination.

3.4.2.7.2.5 EEPROM-Speicherzellen

Das MKT-View+ besitzt 32 (nv[0] bis nv[31]) EEPROM-Speicherzellen (32-Bit) in denen Ergebnisse dauerhaft abgespeichert werden können.

Speichern eines Werts in das EEPROM

Durch eine einfache Zuweisung können Werte in das EEPROM gespeichert werden. Die Speicherung erfolgt zuerst in einem so genannten Schattenspeicher und wird erst durch den Befehl nvs dauerhaft ins EEPROM gespeichert. Da das EEPROM max. 100000-mal beschrieben werden darf, müssen Sie sicherstellen, dass der Befehl nvs nicht zu oft ausgelöst wird.

Beispiel: @nv[0]=Temp1:nvs

Der Wert der Variable Temp1 wird in nv[0] Schattenspeicher gespeichert. Mit dem nachfolgenden Befehl nvs wird dann der Wert ins EEPROM gespeichert.

Sinnvolle Anwendungen sind:

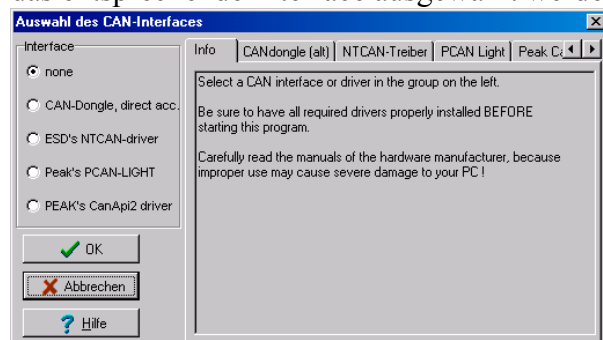
- Speichern der zu letzt benutzten Seite, um diese dann beim Starten des Terminals wieder aufzurufen.
- Speichern des Display-Modus (normale Darstellung bzw. inverse Darstellung)

3.4.2.7.3 Programmierschnittstellen-Einstellungen

Die Programmierung des MKT-View+ kann über die RS232-Schnittstelle oder über die CAN-Bus-Schnittstelle durchgeführt werden.

Auswahl und Einstellungen des CAN-Interfaces für die Programmierung

Bei der Programmierung über die CAN-Bus-Schnittstelle muss zuerst unter *CAN-Interface* das entsprechende Interface ausgewählt werden.



Die in der Liste aufgeführten Interfaces werden unterstützt. Anschließend muss unter *CAN-Baudrate* die passende Baudrate (in der Regel 500 kBit/s) und unter *Node-ID d. Terminals* die erforderliche Node-ID eingegeben werden.

Auswahl und Einstellungen des RS232-Interfaces (im PC, für Programmübertragung)

Normalerweise erfolgt die Programmierung des MKT-View+ über die RS232 Schnittstelle. Es muss unter *RS232-Schnittstelle* die richtige Schnittstelle ausgewählt werden (z.B. COM1), über die die Programmierung durchführen werden soll.

3.4.2.7.4 *Einstellungen des Programmiertools*

Markierung des editierten Objektes im Simulationsfenster

Die Angabe *Objekt markieren* legt fest, ob das zurzeit editierbare Objekt im Display markiert werden soll. In diesem Fall kann das Objekt mit der Maus verschoben werden.

PC-Lautsprecher ein bzw. ausschalten

Mit der Angabe *PC-Lautsprecher an* kann der PC-Lautsprecher ein- oder ausgeschaltet werden. Ein Ausschalten ist während des Testes sinnvoll, wenn sehr viele Töne erzeugt werden.

Verzeichnisse festlegen

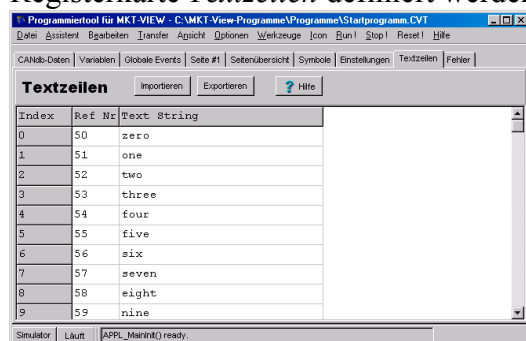
In dieser Tabelle werden die Pfade eingetragen, wo die entsprechenden Komponenten abgespeichert sind.

Sprache / Nationalität auswählen

Zurzeit kann das Programmiertool in den Sprachen Deutsch und Englisch bedient werden.

3.4.2.8 Karteikartenseite „Textzeilen“

In Abhängigkeit einer Variablen (Signals) kann ein Text abgebildet werden. Programmtechnisch wird dies mit einem ‚Text-Array‘ realisiert. Die Strings in diesem Array können auf der Registerkarte *Textzeilen* definiert werden :



Die Vorgehensweise wird unter ... beschrieben.

Die bislang implementierten Stringfunktionen sind:

appl.fname

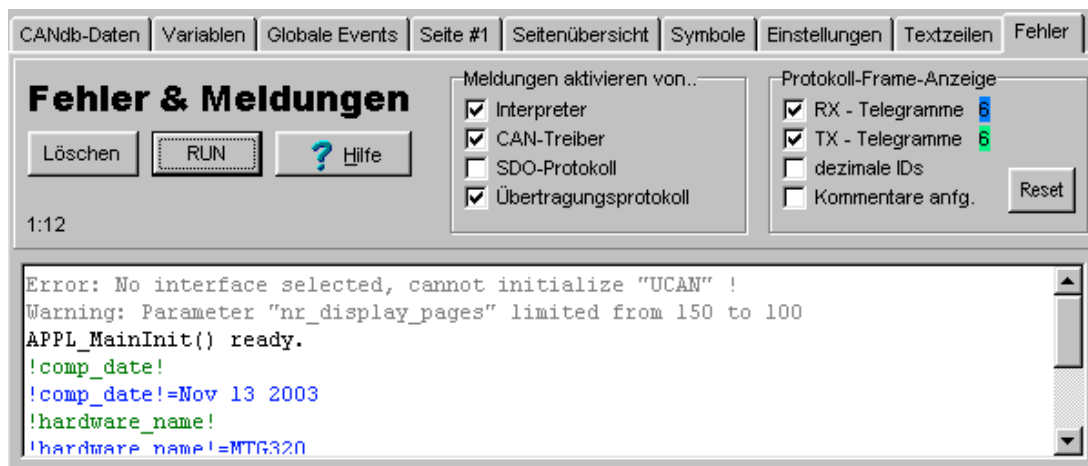
Liefert den kompletten Dateinamen der „Applikation“. Dies ist der Name der *.upt bzw *.cvt-Datei, die in das Terminal geladen wurde.

appl.name

Liefert den Namen der „Applikation“ (ähnlich wie appl.fname), allerdings ohne Pfad und Dateinamenserweiterung.

3.4.2.9 Karteikartenseite „Fehler“

Auf der Anzeigesite *Fehler* werden alle Fehler angezeigt, die während der Datenübertragung zum MKT-View+ auftreten.



Wenn Fehler während des Up- oder Downloads zum MKT-View+ auftreten, können Sie an Hand der CAN-Meldungen erkennen, welches Objekt den Fehler ausgelöst hat (Datenübertragung erfolgt nach CANopen Spezifikation –SDO-Transfer)

4 Interpreter-Kommandos und -Funktionen

Einige Interpreterkommandos und -Funktionen wurden bereits bei der Beschreibung des Programmiertools vorgestellt. Eine **komplette, aktuelle** Übersicht findet sich einzig und allein im Online-Hilfesystem des Programmiertools! Die folgende Beschreibung hat daher nur „informellen“ Charakter.

4.1 Numerische Funktionen

Der numerische Interpreter erkennt nur Funktionsnamen, die mit einem Kleinbuchstaben beginnen. Alle numerischen Funktionen (auch die speziellen "Event-Abfrage-Funktionen") können Teil eines numerischen Ausdrucks sein. Der einzige Unterschied zwischen „Event-Abfrage-Funktionen“ und den im Folgenden angeführten Funktionen ist der Typ des ‚Rückgabewertes‘ (return): Die „Event-Abfrage-Funktionen“ liefern einen booleschen Wert (TRUE oder FALSE), „normale“ Funktionen liefern i.A. einen numerischen Wert.

(Möglicherweise existieren mittlerweile noch weitere Funktionen außer den hier angeführten):

- bk, bs : Button Key, Button-State (nur bei manchen Geräten)
- cs: CAN-Status, bitweise codierter Status des CAN-Controllers
- isin(X): Integer-Sinus aus ROM-Tabelle. X=0...1023 liefert eine volle Sinus-Periode.
- tv0...tv3: (timer values) Werte der programmierbaren Timer
- ti: Systemzeit (time), wird alle 100 Millisekunden inkrementiert
- kd, ku, kb: (key down, key up, key buffered) Tastatur-Abfrage-Funktionen
- lim: (value,min,max) Arithmetische Begrenzungsfunktion.
- nv: Lese- und Schreibzugriff auf nichtflüchtige numerische Werte (non-volatile)
- pn: (page number) liefert die aktuelle Seitennummer
- inp: (inputs) liest den Zustand der digitalen Eingänge des UPT's
- out: Lese(!)- und Schreibzugriff für die digitalen Ausgänge des UPT's (digital outputs)
- sys.xxx: liest bestimmte Systemwerte wie Versorgungsspannung, Temperatur, etc (bei bestimmten Terminals)
- rtc.xxx: (real time clock) Zugriff auf die batteriegepufferte Echtzeituhr, falls vorhanden
- rot.xxx: (rotary knob) Lesen von Modus und Position des Drehknopfes (bei bestimmten Terminals)
- y: enthält den "aktuellen Wert" (value) der Variablen in der aktuellen Anzeigezeile
- mm, mi : Menü-Modus, Menü-Index

4.2 Das Zuweisungs-Kommando (assign, @)

Dieses Kommando weist einer Variablen einen neuen Wert zu.

Syntax:

@<Varname> = <expression>

Beispiel:

@Ypos=(Voltage*63)/230

Hinweise:

- Sie können nur Werte an Variablen zuweisen, die in der Variablen-Tabelle mit dem Programmierwerkzeug definiert wurden (es gibt keine „automatischen“ Variablen wie in der Programmiersprache „BASIC“).
- Sie sollten keine Werte an Variablen zuweisen, die ihren „Eingang“ von einem Kommunikationskanal erhalten, weil der zugewiesene Wert i.A. schnell wieder vom Kommunikationskanal (per CAN) wieder überschrieben werden.
- Auch einige im Interpreter fest eingebaute Kommandos können mit einer formellen Zuweisung aufgerufen werden. Wenn der Operand hinter dem at-Zeichen mit einem Kleinbuchstaben beginnt, handelt es sich nicht um eine (benutzerdefinierte) Variable sondern um ein „Variablen ähnliches Interpreterkommando“.

4.3 „System“-Funktionen und -Kommandos des Interpreters

Dienen zum Aufruf einiger Sonderfunktionen des Terminals, die normalerweise nur über das Systemmenü des Terminals erreichbar sind.

sys.menu

Dieses Kommando kann verwendet werden, um eine bestimmte Funktion im „System-Menü“ der Terminals aus dem Anwenderprogramm (der „Applikation“) aufzurufen. Diese Funktion diente in erster Linie zu Tests während der Entwicklung des Terminals beim Hersteller. Weitere Infos werden Sie zukünftig in einem gesonderten Dokument finden.

sys.edit_contrast

Dient zum Optimieren der LCD-Kontrast-Steuerung. Die genaue Funktion und Bedienung des „LCD-Kontrast-Kennlinien-Editors“ wird in einem gesonderten Dokument beschrieben. Sie können dieses Kommando einsetzen, um dem Bediener den Aufruf dieser Einstellfunktion aus Ihrer Applikation zu ermöglichen, ohne dass sich der Bediener durch das Systemmenü des Terminals „hangeln“ muss.

sys.set_time

Öffnet eine Anzeigeseite, auf der die Echtzeituhr eingestellt werden kann (Datum und Uhrzeit). Nur bei Terminals vorhanden, in denen eine batteriegepufferte Echtzeituhr eingebaut ist. Hinweis: Im Systemmenü des Terminals wird das internationale Datumsformat nach ISO 8601 verwendet. Das Datumsformat ist YYYY-MM-DD (DD=Tag), das Zeitformat hh:mm:ss (Stunde:Minute:Sekunde). Suchen Sie im Web nach "ISO 8601", wenn Sie weitere Informationen zu dieser Normierung benötigen. Es wird empfohlen, dieses Format auch in Ihrer eigenen Anwendung einzusetzen, auch wenn Sie vielleicht kein Programmierer sind. Ein internationaler Anwender wird unter der Angabe „12.05.03“ mit Sicherheit nicht das Datum „12. Mai 2003“ vermuten !

sys.vsup

Liefert die aktuelle Versorgungsspannung *des Terminals*, gemessen „kurz hinter der Verpolschutzdiode“. Der Wert ist daher nicht sehr präzise. Sie können diese Funktion zu Diagnosezwecken einsetzen, wenn das Terminal z.B. aus dem Bordnetz eines KFZ versorgt wird. Einheit: 0.1 Volt, d.h. der Wert 123 bedeutet „12.3 Volt“.

sys.temp

Liefert die aktuelle Temperatur *im* Terminal, gemessen in der Nähe des Displays. Die Terminal-Firmware verwendet diesen Wert für die automatische Kontraststeuerung. Sie können diesen Wert zu Testzwecken in Ihrer Anwendung auf dem Display anzeigen, etc. Beachten Sie, dass die meisten Displays unter -15 °C sehr träge werden, und unter -20°C kaum noch ablesbar sind. Der von der Funktion „sys.temp“ zurückgegebene Wert ist eine Integerzahl mit dem Skalierungsfaktor 0.1 °C, der Wert -185 bedeutet daher z.B. „-18.5 Grad Celsius“.

4.4 Funktionen zur Abfrage der Echtzeituhr (real time clock)

Hinweis: Die MKT-View+ Firmwarevarianten verfügen über eine Echtzeituhr !

Falls die MKT-View+-Firmware die Echtzeituhr unterstützt, wird der aktuelle Wert der Uhr in jeder „Hauptschleife“ einmal ausgelesen und in einem Zwischenspeicher abgelegt. Dadurch werden „Übertragsprobleme“ vermieden, wenn Sie beispielsweise um 23:59:59.999 die Uhrzeit auslesen (und auf dem Display anzeigen) und kurz danach das Datum. Datum und Uhrzeit werden *intern* immer gleichzeitig ausgelesen, auch wenn die folgenden Funktionen so aussehen, als würden Sie per Interpreter auf die einzelnen Werte getrennt zugreifen können (intern erfolgt das Lesen der RTC immer kurz vor einer Anzeige-Seiten-Aktualisierung).

Die folgenden *numerischen Funktionen* lesen Werte aus dem Zwischenspeicher der Echtzeituhr (nicht direkt aus den Zählregistern der Uhr) :

rtc.yr

Jahreszahl, zulässiger Bereich 2000..2000+n.

rtc.mo

Monatszahl des Jahres, Bereich 1..12.

rtc.da (oder rtc.dm, day of month)

Tageszahl des Monats, Bereich 1..31.

rtc.dw (day of week)

Tageszahl der Woche, Bereich 1=Montag..7=Sonntag.

Der Wert 0 zeigt an, dass bei der Berechnung ein Fehler auftrat (i.A. unzulässige Jahreszahl im RTC-Register).

rtc.hr

Stundenwert, Bereich 0..23.

rtc.mi

Minutenwert, Bereich 0..59.

rtc.se

Sekundenwert, Bereich 0..59.

Auf der Uhrenplatine sitzt eine kleine Batterie. Ist diese Batterie entladen (oder die RTC defekt), muss nach der Reparatur oder bei zu großer Abweichung das Datum und die Uhrzeit neu eingestellt werden. Dies kann im System/Setup Menü des Terminals erfolgen (aber nicht durch das Programmierool!). Aus dem Anwenderprogramm kann der Einstelldialog per Kommando `sys.set_time` aufgerufen werden.

Tipp: Wenn Sie statt Tages- und Monatszahl lieber entsprechende Namen auf dem Display anzeigen wollen, verwenden Sie dazu ein paar Zeilen des Textarrays.

4.5 Digitale Ein- / Ausgänge

Der in der Firmware enthaltene Interpreter bietet einige Funktionen und Kommandos zum Zugriff auf die digitalen Ein- und Ausgänge des Terminals, die in den folgenden Unterkapiteln beschrieben werden. Die Anzahl der Ein- und Ausgänge hängt von der verwendeten Terminalvariante ab (siehe Gerätehandbuch).

4.5.1 Funktionen zum Lesen der digitalen Eingänge

Syntax:

inp

Liefert den Zustand aller im Terminal eingebauten digitalen Eingänge als Bitkombination (Bit 0 = erster Eingang, usw.).

inp1 (seit April 2003)

Liefert den Zustand des ersten(!) digitalen Eingangs. Das Ergebnis ist Null (0) oder Eins (1), je nach Klemmenspannung.

inp2

Liefert den Zustand des zweiten digitalen Eingangs. Das Ergebnis ist auch hier Null (0) oder Eins (1).

Die Anzahl verfügbarer digitaler Eingänge, deren Klemmen-Nummern, und die zulässigen Eingangspegel sind dem Gerätehandbuch zu entnehmen.

4.5.2 Das „Output“-Kommando

Das Kommando *out* dient zum Setzen der digitalen Ausgänge.

Es gibt zwei Arten, um die Ausgänge zu Schalten

out(<8-bit-value>)

Mit diesem Kommando können Sie **alle digitalen Ausgänge gleichzeitig** schalten. Bit 0 steuert den ersten Ausgang, u.s.w.

outN(<neuer Zustand>)

Schaltet einen **einzelnen digitalen Ausgang** (Nummer N, 1...8)

out(<output nr>, <new state>)

Schaltet ebenfalls einen **einzelnen digitalen Ausgang** (mit output_nr 0...7). Welcher Kanal geschaltet werden soll, kann hier als Variable übergeben werden.

Beispiele:

- out(\$FF) schaltet alle Ausgänge ein
- out(\$00) schaltet alle Ausgänge aus
- out(0, 1) schaltet den ersten(!) Ausgang ein
- out1(1) schaltet ebenfalls den ERSTEN(!) digitalen Eingang ein (Syntax und Kanalnummer wie bei der inp-Funktion).
- out(3, 0) schaltet den vierten (!) Ausgang aus
- out4(0) schaltet ebenfalls den vierten (!) Ausgang aus
- out(out^\$02) invertiert den Zustand des zweiten digitalen Ausgangs.

Hinweise:

- Es gibt nicht nur das **Kommando out** zum Setzen der digitalen Ausgänge, sondern auch eine **Funktion out** die den **aktuellen Zustand** der digitalen Ausgänge(!) zurückliest - nicht zu verwechseln mit der Funktion *inp* .
- Die digitalen Ein- und Ausgänge können vom Programmierwerkzeug simuliert werden. Die aktuellen Zustände werden als runde Symbole im Simulatorfenster angezeigt. Die grünen Symbole sind die Eingänge, die gelben die Ausgänge. Dass beim MKT-View+ die Ein- und Ausgänge auf identischen Anschlussklemmen liegen, berücksichtigt der Simulator allerdings nicht.

4.6 Funktionen zur Abfrage des Drehknopfes (rotary encoder, rot.xxx)

Normalerweise wird der Drehknopf als Ersatz für die Cursortasten (UP/DOWN) verwendet, um sich durch die Menüs zu bewegen. Drücken des Knopfes hat den gleichen Effekt wie die ENTER-Taste bei anderen Terminals (identischer Tastencode!).

Programmierhandbuch MKT-View-Terminal Version 1.01

Ein kurzer „Doppelklick“ des Drehknopfes schaltet die Funktion des Knopfes zwischen „Cursor hoch/runter“ und „Cursor links/rechts“ um (wichtig in manchen Editierfeldern). Wenn der Bediener den Knopf dreht oder drückt, trägt die Terminal-Firmware normalerweise entsprechende Tastencodes im Tastaturpuffer ein, so dass der Unterschied zwischen „Knopf“ und „normaler Tastatur“ aus Sicht der Applikation minimal ist.

Der Interpreter bietet die folgenden *Funktionen* (und Prozeduren), um den Drehknopf für „spezielle Zwecke“ einzusetzen:

rot.f

Ermittelt oder beeinflusst die aktuelle „Funktion“ des Drehknopfes. Mögliche Werte sind:

rot.f = 0 : der Knopf erzeugt KEINE CURSOR-TASTEN-CODES

rot.f = 1 : der Knopf erzeugt die Tastencodes CURSOR_UP und CURSOR_DOWN (default nach power-on)

rot.f = 2 : der Knopf erzeugt die Tastencodes CURSOR_LEFT und CURSOR_RIGHT

Hinweis: *rot.f* beeinflusst nur die Erzeugung von ‚Cursor-Tasten-Codes‘, aber nicht die folgende Funktion:

rot.p

Liefert die aktuelle Position des Drehknopfs als Integerzahl.

Hinweis: Die Zuweisung eines neuen Wertes an *rot.p* ändert intern nur einen Offset, beeinflusst die aktuelle Menüposition allerdings nicht.

Wird der Drehknopf im Uhrzeigersinn gedreht, wird *rot.p* inkrementiert, entgegen den Uhrzeigersinn führt zum dekrementieren. Der erste Encoder, der in den Terminals eingesetzt wurde, erzeugte 20 Schritte pro voller Umdrehung (360°).

Hinweis: Wie oben erwähnt, müssen Sie den Encoder nicht "selbst" abfragen, wenn Sie ihn zur Navigation in Menüs oder zum Editieren von Werten einsetzen. Die Terminal-Firmware erledigt das selbst, solange Sie *rot.f* nicht auf Null setzen.

rot.l (kleines "L" wie "left")

Liefert 1 (= TRUE) wenn der Knopf seit dem letzten Funktionsaufruf LINKS (im Gegenurzeigersinn) gedreht wurde.

rot.r

Liefert 1 (= TRUE) wenn der Knopf seit dem letzten Funktionsaufruf RECHTS (im Uhrzeigersinn) gedreht wurde.

rot.lb, rot.rb

Ähnlich wie *rot.l* bzw. *rot.r*, mit dem Unterschied, dass diese „gepufferten“ Funktionen (b=buffered) den Wert 1 (=TRUE) für jeden einzelnen Schritt des Drehknopfes liefern (d.h. hier wird intern mitgezählt). Wenn beispielsweise der Knopf seit der letzten Abfrage 5 Schritte (schnell) nach links gedreht wurde, liefert *rot-lb* fünfmal nacheinander den Wert "TRUE", während *rot.l* in diesem Fall nur einmal TRUE liefert (weil *rot.l* nicht die Anzahl der Schritte berücksichtigt, sondern nur die Drehrichtung als solche).

Diese Funktion eignet sich besonders für die Seitenumschaltung per Drehknopf, wenn der Bediener durch einen „schnellen Dreh“ eine bestimmte Anzahl von Seiten weiter-schalten will.

4.7 Timer-Kommandos und -Funktionen

Die Firmware enthält einige Funktionen und Prozeduren, mit der vier Timer gesteuert werden können. Diese Timer können Zeitintervalle bis zu $65534 \cdot 100\text{ms}$ (> 1 Stunde) mit einer Auflösung von 100 Millisekunden erzeugen. Normalerweise werden die Timer mit den programmierbaren Events abgefragt, sie können jedoch auch für andere Zwecke eingesetzt werden.

Timer-Prozeduren

ts0 ... ts3 (timer start)

Startet einen der vier Timer mit der als Parameter angegebenen Intervallzeit. Der Timer zählt dann von diesem Startwert abwärts bis Null; dann wird ein Timer-Event produziert. Wird der Timer neu gestartet, bevor er abgelaufen ist, wird kein Event erzeugt. Der maximale Startwert beträgt $65534 \cdot 100\text{ms}$.

Syntax: `ts0 (<Timer-Startwert in 100ms>)`

Beispiel: `ts0 (40)` startet den ersten Timer für ein 4-Sekunden-Intervall. Nach 4 Sekunden wird das Event "t0" erzeugt, welches Sie in den globalen oder lokalen Event-Definitionen abfragen können.

tr0 ... tr3 (timer reset)

Rücksetzen eines der vier Timer. Der Timer wird gestoppt, er erzeugt dann kein Event mehr. Intern enthält ein gestoppter Timer den Wert `0xFFFF`. Dies ist auch der Default-Zustand nach power-on.

Timer-Funktionen

Die folgenden Timer-Funktionen sind derzeit implementiert. Sie liefern einen booleschen oder numerischen Rückgabewert an den Aufrufer.

t0 ... t3

Testet, ob ein bestimmter Timer abgelaufen ist, d.h. wenn er beim Abwärtszählen den Übergang von 1 auf 0 durchlaufen hat. Rückgabewert:

Null (=FALSE) wenn der Timer nicht läuft oder noch nicht den Wert Null erreicht hat
Eins (=TRUE) wenn der Timer abgelaufen ist (wonach er stoppt !). Das Flag „Timer ist abgelaufen“ wird durch diese Abfrage gelöscht, damit ein Event nur einmal ausgelöst wird.

tv0 ... tv3 (timer value)

Liest den aktuellen Zählerstand eines Timers. Dieser Wert läuft vom Startwert abwärts bis auf Null (ein Schritt in 100 Millisekunden).

ti (global time)

Liest den aktuellen Stand des „System-Taktgebers“ als 32-Bit-Wert. Dies ist ein 32-bit-Zähler, der alle 100 Millisekunden **inkrementiert** wird. Er hängt nicht von den vier User-Timern ab und kann weder gestoppt noch umprogrammiert werden („read-only“).

Programmierhandbuch MKT-View-Terminal Version 1.01

Dieser Timer läuft erst nach 6 Jahren ununterbrochenem Betriebs des Terminal über (von $((2^{31})-1)$ nach (2^{31}) , d.h. der Wert wird dann -als 32-Bit-Integer-Zahl- negativ. Nach Power-On startet dieser Zähler immer bei Null (er hat nichts mit einer eventuell vorhandenen Echtzeituhr zu tun!).

Einige Demo-Programme verwenden diesen Timer (ti), um nach dem Einschalten für kurze Zeit ein Firmenlogo anzuzeigen.

4.8 Seitenumschaltung per Interpreter

Normalerweise schaltet die Applikation per ‚goto‘ oder ‚call‘ und ‚return‘ zwischen den benutzerdefinierten Anzeigeseiten um. Für spezielle Anwendungen steht mit dem page-scan-Modus eine weitere Methode für die Seitenumschaltung zur Verfügung.

4.8.1 Goto, Call und Return (g,c,r)

Diese Kommandos des Interpreters dienen zum Umschalten zwischen den Anzeigeseiten Ihrer Applikation. Alle Kommandos müssen in der abgekürzten Form (als EIN Buchstabe) eingegeben werden. Die „langen“ Namen dienen nur als Gedankenstütze..

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, die durch ihre Seitennummer oder (vorzugsweise) durch den Seitennamen definiert wird.

‚Call‘ speichert darüber hinaus auch die Nummer der aktuellen („alten“) Seite in einem speziellen Stapelspeicher (call-stack) bevor die Umschaltung zur „neuen“ Seite erfolgt. Die so aufgerufene Seite kann später zum „Aufrufer“ mit Hilfe des return-Kommandos zurückspringen. ‚Call + return‘ müssen immer zusammen verwendet werden. Wenn Sie eine Seite per ‚call‘ aufrufen, müssen Sie später auch per ‚return‘ wieder zum Aufrufer zurückkehren, keinesfalls mit per ‚goto‘ zurück zum Aufrufer! (dies würde einen Überlauf des Interpreter-call-stacks verursachen, und ergibt darüber hinaus auch keinen Sinn).

Ein sinnvoller Einsatz für ‚call + return‘ statt ‚goto‘ ist z.B. eine von vielen Aufrufern (hier: Display-Seiten) genutzte gemeinsame Anzeigeseite (ähnlich wie ein Unterprogramm). Nachdem das „Unterprogramm“ seine Arbeit erledigt hat (zum Beispiel eine Fehlermeldung angezeigt und auf eine Taste gewartet hat), kann der Normalbetrieb auf einer der „Hauptseiten“ fortgesetzt werden.

Hinweis: Weder ‚goto‘ noch ‚call‘ haben einen Effekt, wenn das Sprungziel (=die „neue“ Seite) mit der aktuellen Seite identisch ist, egal ob per Seitennummer oder per Seitenname aufgerufen wird.

4.8.2 Page-Scan-Modus (pscan)

In diesem Modus schaltet das Terminal zyklisch zwischen verschiedenen Anzeigeseiten um. Dazu werden keine Event-Definitionen benötigt. Der Page-Scan-Modus kann per Interpreter gesteuert werden:

`pscan=N`

Aktiviert den Page-Scan-Modus. N ist das Seiten-Umschalt-Intervall in Sekunden. Mit „`pscan=5`“ wird z.B. alle 5 Sekunden zur nächsten Seite der Scan-Sequenz umgeschaltet. (Welche Seiten zur Scan-Sequenz gehören sollen wird mit dem Programmierwerkzeug definiert).

`pscan=0`

Schaltet den Page-Scan-Modus aus. Dies ist der Default-Zustand nach dem Einschalten !

`pscan.next`, `pscan.prev`

Schaltet von der aktuellen Seite sofort auf die nächste, zur Scan-Sequenz gehörenden Seite um, bzw. mit „`pscan.prev`“ zur vorhergehenden Seite.

Diese Kommandos sind z.B. zum Durchblättern der Hauptseiten per programmierbarer Buttons vorgesehen (als Reaktion auf die Betätigung eines Buttons, oder -sofern vorhanden- Drehen des Knopfes, etc).

Hinweise und Tipps:

- Wenn eine Seite zur Page-Scan-Sequenz gehören soll, setzen Sie das Checkmark *include this page in the pscan-sequence* im Register Display Page Header. Nur Anzeigeseiten, bei denen dieses Flag gesetzt ist, werden beim Page-Scan-Zyklus aufgerufen.
- Um den Page-Scan-Modus ein- und auszuschalten, bietet sich ein Softkey an (programmierbarer Button). Verwenden Sie z.B. die folgende Anweisung als Button-Reaktions-Methode:

```
pscan=5-pscan
```

Wie funktioniert dies ? „`pscan`“ kann (formell) wie eine Variable verwendet werden, nach dem Einschalten ist der Wert 0 (Null). Mit der genannten Anweisung wird aus 0 eine 5 (= 5 Sekunden), auf 5 wird 0, und so weiter. Sie können auch die Beschriftung des Buttons abhängig vom Zustand des Page-Scan-Modus definieren, wie im folgenden Beispiel (Quelltext einer Button-Definition):

```
\btn(55,50,$002,(pscan>0)?"Stop":"Scan",59,pscan=5-pscan)
```

- Um den aktuellen Stand des Page-Scan-Timers abzufragen, verwenden Sie die numerische Funktion *pscan.timer*. Dies ist ein Count-Down-Timer. Beim Erreichen des Zählerstands 0 (Null) erfolgt die nächste Page-Scan-Seitenumschaltung. Die Einheit des Zählers ist 0.1 Sekunden.
- Beim Betätigen einer beliebigen Taste wird der Page-Scan-Timer automatisch mit der doppelten Intervallzeit neu geladen, d.h. Page-Scan "pausiert" solange der Bediener Tastatureingaben macht; Page-Scanning wird aber nicht (automatisch) abgeschaltet.

4.8.3 Kommandos zum Ansteuern des Piepers

Hinweis:

Geräte ohne Pieper kennen diese Befehle nicht, der Interpreter könnte dort eine Fehlermeldung ausgeben!

- `beep(<frequency> [,<time> [,<volume> [,<frequency_mod> [,<amplitude_mod>]]]])`

Nur der erste Parameter (Frequenz in Hz) ist zwingend, alle weiteren Parameter sind optional. Siehe Beispiele bei der Beschreibung des Befehls. Das *beep*-Kommando kann als *be* abgekürzt werden, wenn der Platz in einer Kommandozeile knapp wird.

- `play(<string of tones>)`
Dient zum Abspielen von Klingeltönen, Alarmen, oder kurzen Melodien. Siehe Beispiele bei der Beschreibung des Befehls.

4.8.3.1 *beep*-Kommando

Erzeugt einen einzelnen Ton, der optional frequenzmoduliert werden kann (Aufwärts- oder Abwärtsrampe).

Syntax:

```
be( <freq>, [ <time>, [ <volume>, [ <fm>, [ <am> ] ] ] ] )
```

Parameter:

freq

Tonfrequenz in Hertz. Der Wert Null schaltet den Ton aus, falls keine anderen Parameter folgen.

Achtung: Die minimale Frequenz beträgt bei Geräten mit 24-MHz-Takt 366 Hz; bei Geräten mit 40-MHz-Takt 610 Hz. Falls die angegebene Tonfrequenz nicht realisiert werden kann, wird die nächstgelegene realisierbare Frequenz erzeugt.

time

Länge des Tons, gemessen in 100-Millisekunden-Schritten. Wenn dieser und alle weiteren Parameter fehlen, wird ein ‚endloser‘ Ton produziert, der mit dem Kommando `beep(0)` oder `be(0)` wieder abgeschaltet werden muss.

volume

Relative Lautstärke in Prozent, im Bereich 0 bis 100. Der Pieper wird intern mit einem Pulsbreitenmodulator angesteuert, wobei das Tastverhältnis für die Lautstärkeneinstellung dient. So lassen sich zwar verschiedene Lautstärken erzielen, das Spektrum des Tons ändert sich allerdings auch (je nach Tastverhältnis). Der *volume*-Wert 100 erzeugt die maximale Lautstärke mit einem Tastverhältnis von 1:1.

fm

Frequenzmodulation. Kann zur Realisierung von Sirenen-ähnlichen Klängen eingesetzt werden. Die Einheit ist ‚Hertz pro 100 Millisekunden‘. Bei positivem Wert steigt die Frequenz an solange der Ton dauert, bei negativem Wert fällt sie ab.

am

Amplitudenmodulation. Kann für allmählich anschwellende Töne verwendet werden, die Funktion ist allerdings wegen der Pulsweitenmodulation nicht sehr effektiv (eine Lautstärke von 10% kann kaum von 50% unterschieden werden). Auf dem PC (Programmierool, Simulator), ist die Amplitudenmodulation nicht möglich. Ein mit dem PC-Speaker erzeugter Ton kann mit einfachen Mitteln nicht pulsweitenmoduliert werden (unter Windows erst recht nicht..).

Beispiele für das *beep*-Kommando:

be(1800)

Startet einen „endlosen“ Ton mit maximaler Lautstärke, und der Frequenz 1800 Hz.

be(0)

Beendet einen vorher gestarteten Ton

be(1000,5)

Erzeugt einen einzelnen, nicht modulierten Ton mit 1000 Hz und einer Dauer von 0.5 Sekunden.

be(150,20,50,100)

Erzeugt einen 2 Sekunden dauernden Ton, dessen Frequenz von 150 Hz bis auf 2150 Hz (=150 + 2 sec * 100 Hz/0.1sec) ansteigt.

4.8.3.2 *play*-Kommando

Erzeugt eine Sequenz von Tönen, z.B. Klingeltöne, Alarmer, oder kurze Melodien.

Syntax:

pl(<string>)

Der String ist eine Zeichenkette aus den folgenden Buchstaben:

a...l

12 Halbtöne in der ersten Oktave. 'a'=262 Hz, 'b'=277 Hz, 'j'=440 Hz, 'k'=466 Hz, 'l'=494 Hz.

(Die Buchstaben haben nichts mit historischen Noten-Namen zu tun, z.B. ist der „Kammerton A“ ist hier "j" !)

m...x

12 Halbtöne in der zweiten Oktave. 'm'=523 Hz 'x'=988 Hz.

A...L

12 Halbtöne in der dritten Oktave. 'A'=1046 Hz 'L'=1975 Hz.

M...X

12 Halbtöne in der vierten Oktave. 'M'=2093 'X'=3951Hz.

' ' (Leerzeichen)

Fügt eine Pause in die abgespielte Melodie ein, mit der aktuellen Tondauer (0..9)

0...9

Setzt die Dauer der nachfolgenden Noten und Pausen:

'0' = die kürzeste realisierbare Dauer, abhängig vom Timer-Interrupt (meistens 10 Millisekunden)

'1' = 50 Millisekunden

Programmierhandbuch MKT-View-Terminal Version 1.01

'2' = 100 Millisekunden
'3' = 150 Millisekunden
'4' = 200 Millisekunden
'5' = 400 Millisekunden
'6' = 600 Millisekunden
'7' = 1000 Millisekunden
'8' = 1500 Millisekunden
'9' = 2000 Millisekunden

!vN

volume. N=1 erzeugt einen sehr leisen Ton, N=9 den lautesten Ton.

!aN

attack interval. Der Parameter 'N' definiert, wie schnell ein Ton von „Null“ bis zum Maximum (Wert von !v) anschwillt.

!dN

decay interval. Der Parameter 'N' definiert, wie schnell ein Ton vom Maximum bis auf den Sustain-Level abfällt.

!sN

sustain level. Haltepegel von Dauertönen. Mit !s0 können Töne erzeugt werden, die keinen Dauerton, sondern nur einen kurzen abfallenden Ton haben, z.B. Glocken, Klavier (oder eher Cemballo).

!rN

release interval. Nachklingdauer bis zur völligen Stille.

!lN

loop. N=1...9 definiert die Anzahl der Schleifen-Wiederholungen. Das Schleifenende muß mit !le (loop end) markiert werden. Schleifen im Melodie-String sind nicht schachtelbar.

Hinweise:

- Zwischen zwei Noten werden keine automatischen Pausen eingefügt !
- Um eigene Melodien zu „komponieren“, kann der Simulator im Terminal-Programmierwerkzeug verwendet werden (*View...Test Command Window*), Töne werden allerdings nur produziert, wenn das SmallPort-Utility korrekt installiert wurde, außerdem muß der Simulator im Zustand *Läuft* sein (nicht *Gestoppt*). Verwenden Sie die F2-Taste um das *play*-Kommando in einer Zeile des Kommandofensters wiederholt abzuspielen, bis die Melodie fertig ist) .

Beispiele für das *play*-Kommando (Abk. pl)

- pl("5m4o5qo2 5rq4ol5m")
(deutsche Nationalhymne)
- pl("!d7!s0 6qmo9h6 hoq9m")
(Westminster-Klang alias "Big Ben"-Melodie)

4.9 CAN - Sende - Kommando (ctx)

Kann verwendet werden, um ein (einzelnes) CAN-Telegram mit beliebigem Identifier (11- oder 29-bit) und beliebigem Datenfeld zu senden.

Wir raten dringend davon ab, dieses Kommando zu verwenden, denn es kann sehr gefährlich sein ein CAN-Telegramm in ein nur teilweise bekanntes CAN-Netzwerk zu senden, d.h. ohne Kenntnis des verwendeten „CAN ID – Managements“ (welches z.B. in einem Kommunikationsprofil definiert ist, bzw in einer CANdb-Datei). Denken Sie daran, wenn sich die Laufkatze des CAN-gesteuerten Krans plötzlich mit Vollgas in die Hallenwand bohrt !

Die prinzipielle Syntax des CAN-Sende-Kommandos ist:

```
ctx( identifier, databyte, ...)
```

Der Identifier kann den Zusatz „.x“ erhalten, wenn ein eXtended frame (29-bit-ID) gesendet werden soll. Ohne Präfix erwartet der Interpreter den CAN-Identifier in hexadezimaler Schreibweise; wenn nötig verwenden Sie das Zeichen '#' wenn der ID in dezimaler Form geschrieben werden muß. Wenn Bit 29 im Identifier gesetzt ist (hexadezimale Maske 0x20000000), wird ebenfalls ein extended frame gesendet (beachten Sie, dass Bits immer von 0 bis N gezählt werden, eine Integer-Zahl im Interpreter besteht aus Bits 0..31) .

Die Datenbytes (max. 8) sind im einfachsten Fall hexadezimale Konstanten, oder dezimale Werte (wenn das '#'-Zeichen vorangestellt wird). Darüber hinaus sind auch numerische Ausdrücke möglich, mit Variablen und Funktionsaufrufen.

Ein ganz besonderer Service wurde noch eingebaut: Es ist möglich, 16- und 32-bit Werte mit INTEL- oder MOTOROLA-Byte-Reihenfolge zu senden:

- .w oder .iw = INTEL WORD (16 bit),
- .l (kleines "L") oder .il = INTEL 'LONG' (32 bit),
- .mw = MOTOROLA WORD, .ml = MOTOROLA 'LONG' ('LONG' = 32 bit).

Einige Beispiele zum *ctx*-Kommando:

```
ctx(4D2,1,2,3,4,5,6,7,8) ; preferred notation separated by comma
ctx(#1234, $7B,$7C,$7D,$7E,$7F,$80) ; DECIMAL ID but HEX-DATA
ctx($4D2, #123,124,125,126,127,128) ; HEX-ID but DECIMAL Data
ctx(#1234.x, 11,22,33,44,55,66,77,88) ; EXTENDED frame (29-Bit-ID)
ctx(#1234.s, 11,22,33,44,55,66,77,88) ; STANDARD frame (11-Bit-ID)
ctx(4D2, $ABCD.w , #32767.w) ; zwei INTEL-WORDS
ctx(4D2, $12345678.l , #12345678.l) ; zwei INTEL-LONGs
ctx(4D2, $ABCD.mw, #32767.mw) ; zwei MOTOROLA-WORDS
ctx(4D2, (ti).il, (ti).ml ) ; TIME a la Intel and Motorola
```

Bei Terminals mit zwei CAN-Schnittstellen existieren zwei CAN-Sende-Kommandos. Das Kommando *ctx* sendet immer mit der ersten Schnittstelle (ebenso wie das Kommando *ctx1*).

Um ein Telegramm über die zweite Schnittstelle zu senden, verwenden Sie das Kommando *ctx2*(identifier, databyte, ...)

Die Parameterliste für dieses Kommando hat selbe Bedeutung wie bei *ctx* .

Warnung: Wie oben erwähnt, kann es sehr gefährlich sein, auf diese Art CAN-Telegramme zu senden !

4.10 Spezielle Grafik-Kommandos

Einige Grafikkommandos wurden bereits bei der Beschreibung des Programmiertools vorgestellt. Eine komplette, aktuelle Übersicht findet sich einzig und allein im Online-Hilfesystem des Programmiertools ! Die folgende Beschreibung hat daher bestenfalls informellen Charakter.

4.10.1 Invertierung der Anzeige (Display Inversion, *di*)

Bei bestimmten Geräten mit 16-bit CPU ist es möglich, per Interpreterbefehl *das komplette Display* zu invertieren.

Syntax:

- di=0 keine Display-Inversion, schwarze Schrift auf weißen Grund
- di=1 Display-Inversion, weiße Schrift auf schwarzem Grund

Im Gegensatz zum Kommando *dm* (welches nur die danach ausgegebenen Zeichen beeinflusst) wirkt sich *di* immer auf den *gesamten Bildschirm* aus. Die Invertierung wird z.T. auf Hardware-Ebene durchgeführt, das Kommando ist dadurch relativ schnell und kann z.B. dazu verwendet werden, den Bildschirm ein paar Mal aufblitzen zu lassen, um die Aufmerksamkeit des Bedieners auf sich zu lenken.

4.10.2 Das CLearScreen-Kommando (*cl*)

Dieses Kommando zum Löschen des Bildschirms führt *während des nächsten Bildaufbaus* die folgenden Aktionen durch:

- Löschen des kompletten Bildschirms
- Neuzeichnen aller auf der aktuellen Seite definierten Anzeigezeilen

Erst nachdem der Neu-Aufbau komplett ist, wird die Grafik auf das LC-Display übertragen. Dieses Kommando wird nur für komplexe Anzeigeseiten benötigt, z.B. wenn sie eine per Interpreterkommandos gezeichnete Grafik wieder löschen wollen (wie z.B. im Beispiel „Pixel Plot“).

Im Normalfall benötigen Sie dieses Kommando nicht, weil der Bildschirm bei einer Seitenumschaltung automatisch gelöscht wird