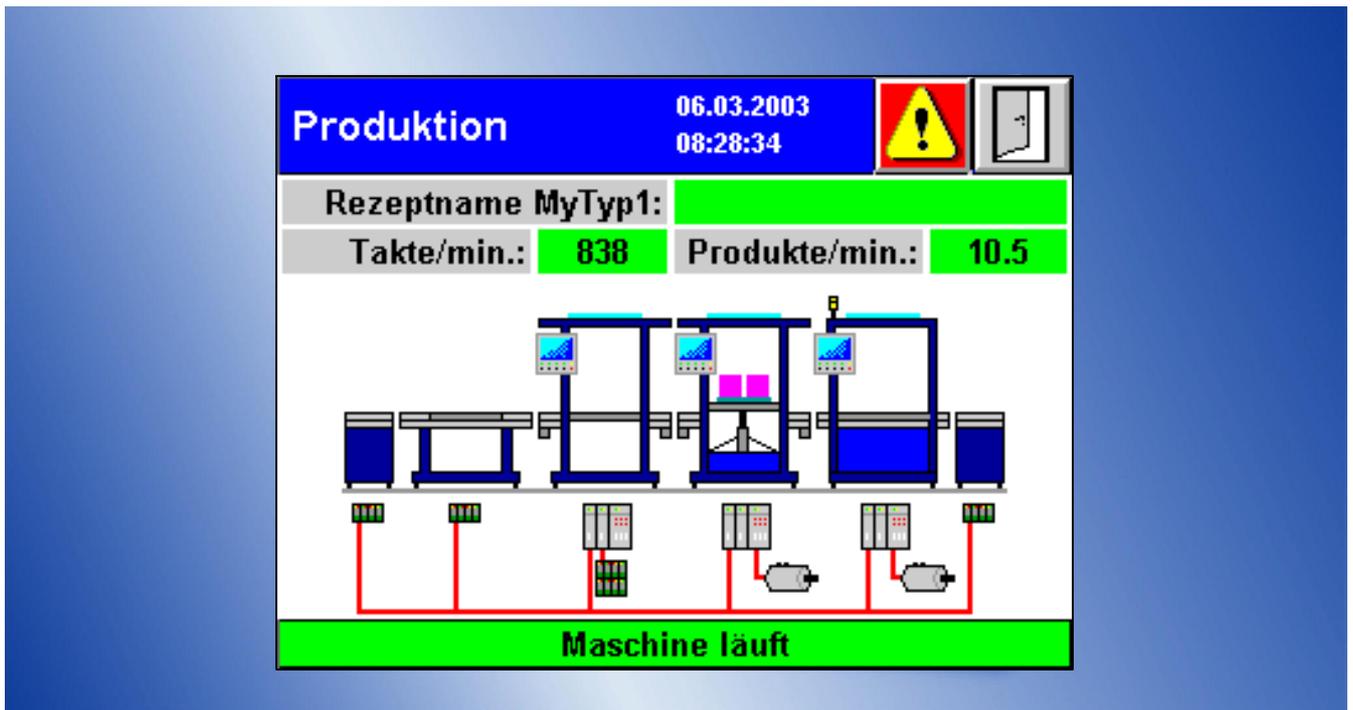




Easy PageMachine V3.50



Unterlage für zukünftige Verwendung aufbewahren!

Diese Unterlagen sind geistiges Eigentum der , der auch das ausschliessliche Urheberrecht daran zusteht. Eine inhaltliche Änderung, die Vervielfältigung oder der Nachdruck dieser Unterlagen sowie deren Weitergabe an Dritte ist nur mit der ausdrücklichen Erlaubnis der gestattet.

lehnt jede Haftung für Schäden ab, die durch die Anwendung von allenfalls falschen bzw. unzureichenden oder aufgrund fehlender Informationen in diesen Unterlagen entstehen.

behält sich das Recht vor, dieses Dokument vollständig oder teilweise zu ändern.

Symbole für Warnhinweise

Warnung vor einer allgemeinen Gefahr



Warnung vor gefährlicher elektrischer Spannung



Elektrostatisch gefährdete Bauelemente. Gehäuse oder Verbindungen dürfen nur von geschultem Personal geöffnet werden!



Hinweise

Die Version des gesamten Dokumentes wird bei jeder Änderung erhöht.
Erfolgte Änderungen sind im Kapitel Versionsgeschichte aufgeführt.

Autor G.Fischbacher

Copyright © 2011 Grossenbacher Systeme AG
Spinnereistrasse 10
CH-9008 St.Gallen
Schweiz

Inhaltsverzeichnis

Seite

1	Einleitung	3
1.1	Neue Funktionen der EPAM V3.50	3
1.2	Neue Funktionen der EPAM V3.40	4
1.3	Neue Funktionen der EPAM V3.30	4
1.4	Neue Funktionen der EPAM V3.20	5
1.5	Neue Funktionen der EPAM V3.10	6
1.6	Hinweise zur Anpassung bestehender Applikationen	7
2	Installation	10
2.1	Easy PageMachine EPAM	10
2.2	Einstellungen in Excel	10
2.3	Zeichnungsprogramm	10
2.4	Einstellungen in der CoDeSys-Entwicklungsumgebung	11
2.5	EPAM Runtime-System	15
3	Projektrealisierung	19
3.1	Zusammenstellung der Anforderungen	19
3.2	Strukturierung der Bildseiten	19
3.3	Definition des Bildseitenlayouts	19
3.4	Bilderstellung	19
3.5	Realisierung mit Excel	19
3.6	Anbindung an die Steuerung	20
3.7	Dokumentation	20
3.8	Tipps für TouchScreen Applikationen	20
4	Bilderstellung	21
4.1	PCX-Bilder erstellen	21
4.2	Icons erstellen	22
4.3	Bilder importieren	23
4.4	Bilder von Digitalkamera oder gescannt	24
4.5	Transparente PCX-Bilder	25
5	Zeichensatzerstellung	27
5.1	Definieren von Zeichensätzen	27
5.2	Erstellen von Zeichensätzen	30
5.3	Unicode-Support	31
6	Projektierung mit Excel	35
6.1	Funktionsprinzip	35
6.2	Struktur der Excel-Tabelle	36
6.3	Tabellenblätter in Excel	42
6.4	Passwortschutz eines EPAM-Projekts	43
6.5	EPAM-Makros	44
6.6	Ein kleines Projekt von A-Z	63
7	Objektdefinition	71
7.1	Objekt Page	71
7.2	Objekt Button	73
7.3	Objekt Switch	79
7.4	Objekt DropDownList	81
7.5	Objekt Radiobutton	83
7.6	Objekt Variable	85
7.7	Objekt Bar	89
7.8	Objekt Signal	91
7.9	Objekt Message	94
7.10	Objekt Meter	96

7.11	Objekt Textlist	98
7.12	Objekt HTMLBrowser	100
7.13	Objekt Alarm	101
7.14	Objekt Alarmlist	105
7.15	Objekt Alarmmail	108
7.16	Objekt DiagSig	109
7.17	Objekt Recipe	110
7.18	Objekt RecipeList	114
7.19	Objekt ScreenSaver	116
7.20	Objekt Password	118
7.21	Objekt Scrolllist	120
7.22	Objekt DataLog	121
7.23	Objekt Trend	125
7.24	Objekt Sys2Plc	127
7.25	Objekt RemoteControl	129
8	Application Notes	131
8.1	Alarmhandling	131
8.2	Rezepthandling	138
9	Systemvariablen	141
10	Fehlermeldungen	144
11	Versionsgeschichte	147
12	Stichwortverzeichnis	149

1 Einleitung

Das Visualisierungstool **Easy PageMachine (EPAM)** ist speziell für die grafische Benutzerführung mit Touch ausgelegt und erlaubt eine einfache **Parametrierung der Visualisierung ohne aufwendige Programmierung**. Zur Erstellung der einzelnen Bildschirmmasken stehen Objekte wie Button, Switch, alphanumerische Variable, Bar, Message etc. zur Verfügung. Diese Objekte werden in einer strukturierten ASCII-Datei konfiguriert, mit den PLC-Variablen verknüpft und zu kompletten Bildseiten zusammengefügt. Die verschiedenen Bildseiten werden durch „Links“ verknüpft und können z.B. durch „anklicken“ eines Button-Objektes aufgerufen werden.

Der tabellarische und somit übersichtliche Aufbau dieser ASCII-Datei (Skriptdatei) erlaubt die Erstellung mit einer Standard-Tabellenkalkulation wie Excel o.ä. Aufgrund des transparenten Datenformats wird die Projektdokumentation praktisch automatisch mit erstellt.

Ein Interpreter ermöglicht den sofortigen Test der Applikation auf dem Entwicklungs-PC. Die Visualisierung kann hierbei mit der Maus bedient werden. Änderungen können dadurch in sekundenschnelle durchgeführt und unmittelbar getestet werden. Anschliessend wird die Applikation ins Zielsystem geladen.

Voraussetzungen Entwicklungssystem:

- IBM-kompatibler PC
- Windows 2000/XP/Vista/7
- Office/Excel 2000 oder neuer
- Zeichnungsprogramm zur Erstellung von Bildern im PCX-Format.
z.B. Paint Shop Pro (Windows Demoversion ist auf CD) s.a. <http://www.jasc.com>
- FontBuilder/FontWindow zur Erstellung eigener Zeichensätze s.a. <http://www.metagraphics.com>

Voraussetzungen Zielsystem:

- Geräte der CPC-300/600-Serie (WindowsCE)
- Geräte der EP-300/370-Serie (WindowsCE)
- Geräte der SP-200/240-Serie (WindowsCE)
- Geräte der XV100,200,XVH-3xx/XV-4xx-Serie (WindowsCE)
- Geräte der XVC-600, XCC-600 Serie (VxWorks)
- Geräte der HPG-200, 300 Serie (VxWorks)
- PC mit Windows2000/XP/Vista/7
- PocketPC mit WindowCE (ARM Prozessor, Pocket Windows 2002/2003, z.B. HP iPAQ 5450)

1.1 Neue Funktionen der EPAM V3.50

V3.50

- Support für Windows-Vista/7
- Neue WindowsCE Targets: SP-240, EP370 und Runtime für WindowsCE6
- Neuer Kommunikationstreiber für Codesys V3 (PLC-Handler) inkl. Variablen-Import
- Kommunikationstreiber DRVRS7 auch für S7 CP-343 via Ethernet TCP/IP
- EPAM-Wizard: Positionierung über Cursortasten, Dimensionierung der Objekte mit Ctrl + Cursortasten, Selektion der Objekte mit Maus (Rechteck ziehen), Anzeige von EPAM4 Scroll-List und Group-Objekten, Anzeige von EPAM4 PNGs und GIF-Bildformaten, Anzeige von transparenten Objekten und Invisible Objekten (werden transparent dargestellt)
- Support für TTC-Fontdateien
- Neue Systemvariable: s_dhcp_mode zur Einstellung von DHCP-Mode via EPAM
- Neue Systemvariable s_user enthält aktuellen User aus s_user_x zu Passwordlevel s_password_x
- Neue Systemvariable s_alarm[<MyAlarm1>].active_count zeigt aktuelle Anzahl aktiver Alarmer
- Neue Systemvariable s_pageid_last, enthält letzte PageID mit Option ID > 0
- Neue Systemvariable s_pwl_required zeigt den erforderlichen PWL
- Objekt Signal: Support von JPG-Bildformat, neue Option animation-delay=x zeigt eine Bildfolge als Animation, wenn Wert > 0; neuer Datentyp REAL, Support für Umrechnung
- Objekt Button: neue Aktion 'remoteclient=drop' zur Deaktivierung des Remote-Client, neue Aktion ShiftEnd springt zum Anfang des Trend bzw. Datalog

- Objekt Scroll-List: Anzeige ohne Slider mit Option DX=0; neue Option sbimg=<scrollbar.pcx>,thumbimg=<thumb.pcx> für customized Scrollbars
- Objekt Alarm-List, Text-List, RecipeList: neue Option sbimg=<scrollbar.pcx>,thumbimg=<thumb.pcx> für customized Scrollbars
- Neue Limit-Aktion: PWL=x
- S7-Variablen-Import unterstützt symbolische Variablenamen und importiert auch Kommentare; Support für S7 TIME Variablen (IEC_TIME) und S7-Strukturen
- Support von BOOL-Variablen für Objekt-Status (VarState)
- Objekt Trend: neue Parameter: #cursorcolor=<color>,#xscalefgcolor=<color>,#xscalebgcolor=<color>
- EPAM3 Makros: Inkompatible EPAM3 Makros können nicht mit EPAM4 verwendet werden -> Fehlermeldung
- EPAM beendet die StartLogo.exe, falls diese zur Anzeige eines Logos in der Autoexec.cmd gestartet wurde.

1.2 Neue Funktionen der EPAM V3.40

V3.40 SP1

Neues Target: XV-100 StorageCard
Update DRVKeTop Treiber

V3.40

- Neue WindowsCE Targets: SP-200, EP300-07, XV-100, XVM-400, KeTop50
- Support Tastaturbedienung (Button-Option: Key=)
- PrintScreen Support für WindowsCE
- WebEPAM TrueColor Support (epamview.jar V1.0.04)
- Support DNS Settings via EPAM
- RAM-Drive vergrößert für EP-300 (16MB), CPC-300/650 (32MB)
- Neue Kommunikationstreiber für ELAU- (MAX4, C-Series, P-Series), WAGO- (750-841), Parker-SPS (C3-Series), UDP (ASCII)

V3.30 SP2

- RemoteClient V1.0.2
- Support AT-S7 SPS Kontroll-Funktionen: STOP, RUN, Reset,... (drvrs7.dll V1.4.2, rs7dll.dll V3.2.0)
- Support Windows Vista und Office 2007 (epamcom.dll V1.1.1 skip.exe V1.2.7)

V3.30 SP1

- Neues WindowsCE Target: XVH-2xx.
- Support UserColors.
- Support online Mass-System Umschaltung (z.B. mm/inch).
- Neue Kommunikationstreiber für Siemens-MPI (benötigt ProfibusDP-Slave/MPI Option), AT-S7 (Step7 kompatible Soft-PLC), Multiprog-PDD.
- Neue Address Spalte in UserVar für Siemens-MPI, AT-S7.
- Support CoDeSys PLC Kontrollfunktion (STOP, RUN, Reset,...).
- Button Objekt Ausgabe geschwindigkeitsoptimiert.
- Neue Option "Type=" für Alarm, Trend und Datalog Objekt zur Verwendung derselben Definitionsdatei für mehrere Objekte.
- Neuer Makro "Project compare" zum Vergleich von zwei EPAM Projekten
- Support für Bilder mit 16 Million Farben (24 Bit) für WindowsCE/Windows

1.3 Neue Funktionen der EPAM V3.30

- Neue WindowsCE Targets: XVH-3xx, XV-4xx, EP-300, CPC-300, CPC-600, PocketPC
- Neue Windows2000/XP-Targets: PC (WinEPAM)
- Support 19" Displays 1280x1024
- Neues Objekt: DropDownList zur Auswahl von Listenelementen

- Neues Objekt: HTML-Browser zur Anzeige von HTML-Seiten (nur Windows)
- Neue Objekte zur Datenbank-Ankopplung: DBPasswd, DBTracer (nur VxWorks mit EPAM-DB-Extension)
- RemoteControl-Objekt: unterstützt Login mit Passwort
- Rezept-Objekt: Unterstützung von mehrstufigen Rezepturen (Rezept1 lädt Rezept2)
- Rezept-Objekt: neues Kommando Rezept aus PLC sichern
- Page-Objekt: Unterstützung von relativen Fenster-Positionen
- Passwort-Objekt: neue Option „SysPW=off“ deaktiviert das datumabhängige Masterpasswort
- Passwort-Objekt: neue Option „Bitwise=AND“ erlaubt flexiblere Vergabe von Berechtigungen
- Variablen-Objekt: neue Option „CoselfOk“ schliesst automatisch die Tastatur
- Rezeptlist-Objekt: speichert Cursorposition analog zu Textliste
- Signal-Objekt: transparente Hintergrundfarbe
- Variable-Objekt:transparente Hintergrundfarbe
- Trend-Objekt: erweiterbare Funktionen zur Trendanzeige (Skalierung ein/aus, Skalenfarbe, Y-Linien, Datalog Spalte)
- Neue Button-Aktion „PrintScreen“ (nur Windows2000/XP) für Standard-Drucker
- Neue Button-Aktion: Reboot zum Neustart des Systems
- Neue Button-Aktion: FileCopy(dst=path\file.ext src=path\file.ext)
- Neue Button-Aktion: EjectVolume(Drive #Page=eject_failed #Page=eject_ok) zum Abmelden von Wechselspeicher-Geräten wie z.B. USB-Memorystick (nur Windows)
- Neue Limit-Aktionen: s_myvar=x, language=x, language=s_myvar
- Verbessertes Fehlerhandling bei Kommunikationsfehlern (SymArti)
- Sprachen können bei Sprachumschaltung einzeln (EPAM.INI: LOAD_LANGUAGE=1) in den RAMDrive geladen werden (Default: alle Sprachen im RAMDrive)
- EPAM-Wizard: Copy/Paste/Delete Funktionen mit Ctrl-C/V bzw. Delete-Taste
- Neue Kommandozeilen-Optionen für WinEPAM (Windowname, X,Y-Position)
- Neue Systemvariablen: s_alarm_tin_dt, s_alarm_tout_dt, s_alarm_tquit_dt zur freien Formatierung der Alarminfo
- Neue Systemvariablen s_myrecipe_cur_file und s_myrecipe_cur_name zeigt das aktuell angewählte Rezept in der Rezeptliste
- Neue Systemvariable: s_plcstate_<hostname> zur Anzeige des Status der (Remote)-Steuerung
- In den Spalten „Color“ können auch Farbnummern angewählt werden
- Neue Umgebungsvariable USERCOLOR=Yes bzw. =Bild.pcx (Epam.ini) zur Unterstützung von kundenspezifischen Farbpaletten (256 Farben).
- Neue Umgebungsvariable DRIVER=No (Epam.ini) zur Deaktivierung der Kommunikation auf dem Gerät
- Neue Umgebungsvariable Kbd=off (Epam.ini) zur Unterdrückung von Tastatureingaben
- Support mehrerer SymArti-Verbindungen unter WindowsCE
- Zugriff auf RemoteServer im nur Anzeigemodus (keine Eingabe)
- Release Mode EPAM_NOEXIT=yes (Epam.ini) verhindert das Beenden von EPAM auch mit Tastatur
- Makro Download prüft nun auch ob sprachspezifische Textdateien für Textlist und Alarmhilfe sowie sprachabhängige Bilder und Icons vorhanden sind. Fehlende Dateien werden in einer Listbox dargestellt.

1.4 Neue Funktionen der EPAM V3.20

- Für die Installation der Version 3.20 wird ein Produktcode benötigt. Ohne Produktcode wird die Demoversion installiert. Applikationen die mit der Demoversion erstellt wurden, haben eine limitierte Laufzeit von 1 Stunde!
- Neue Targets: MC-HPG200/300, MC-HPG200 Portrait (Hochkant), XVC600, XCC600
- Support für Geräte mit Resistiv-Touch (inkl. Kalibrierung)
- Unterstützung von „Überlagerten Objekten“ z.B. Button mit LED, Bargraph mit Wert, etc.. D.h. Objekte können übereinander gelegt werden und werden dann automatisch aktualisiert. (s.a. Demo „Überlagerte Objekte“)
- Aus/Einblenden von Objekten auf Grafiken (Hintergrund wird wieder hergestellt)
- Neues Makro „Zoom-Project“ zur automatischen Konvertierung von Projekten für verschiedene Auflösungen
- Neues Makro „Build Recipes“ zur Erstellung von Rezepten in EXCEL
- Möglichkeit zum Download von Rezepten die unter EXCEL erstellt wurden

- Download-Einstellungen werden projektspezifisch gespeichert
- Download des Projekts auch im Release-Mode mit Ramdrive ohne Neustart möglich
- Mehrfache Definition von Bildseiten wird überprüft
- Verbesserte Fehleranzeige in EXCEL mit Fehlerliste
- Anwenderspezifische Gliederungseinstellungen in EXCEL werden beibehalten
- Geschwindigkeitsoptimierung der Makros unter EXCEL
- EPAM-Wizard für die Anzeige von 700 Objekten/Seite erweitert. Es erfolgt eine Fehlermeldung, wenn diese Anzahl überschritten wird.
- Neue Systemvariable `s_recipelist_empty` kann zum Sperren des "Load" Buttons in der Rezeptliste verwendet werden (bei leerer Rezeptliste)
- Neue Systemvariable `s_toucherror` zur Anzeige von Fehlern des IR-Touch
- Neue Systemvariable `s_irtouch` zur Erkennung der IR-Geräte
- Neue Systemvariable `s_remoteclient_connected` zur Anzeige eines Remote-Zugriffs
- Neue Parameter `Retry` und `DelayOnError` im Tabellenblatt `Hosts` bei vernetzten Systemen
- Unterstützung für länderspezifische Tastaturen mit Windows Charsets (z.B. Kyrrillisch, etc.)
- Neue Button-Aktion „Close=Pagename“ schliesst Fenster „Pagename“
- Neuer Formate für Datalog zur autom. Definition der Feldbreite im Logfile
- Neue Option `pos=left/right/center` für Button, Switch, Radiobutton
- Diagnose-Signal mit Limit-Aktionen: bei Wechsel der Bedingung von aktiv -> inaktiv (gehen) wird die `Limit1`-Aktion ausgeführt, bei Wechsel der Bedingung von inaktiv -> aktiv (kommen) wird die `Limit2`-Aktion ausgeführt
- Ausblenden des Scrollbars bei der Alarmliste mit Option: `DX=0`. Neue Option `Coff` zum Ausblenden des Cursors.

1.5 Neue Funktionen der EPAM V3.10

- Vereinfachte, automatische Installation der EPAM-Makros
- Windows-Version `WinEPAM` zur Simulation unter EXCEL inkl. Kommunikation zur Steuerung
- Neues Fonthandling: Fonts können in EPAM definiert (Makro „New Font“) und mit Makro „Build Fonts“ automatisch erstellt werden (s.a. Kap. 5 S.27)
- Darstellung der verwendeten Fonts auch im EPAM-Wizard (Option: `Map Fonts`)
- Neues Objekt: `Sys2PLC` zum Datenaustausch von Systemvariablen mit der PLC (s.a. Kap. 0 S.127)
- Neues Objekt: `RemoteControl` zur Fernbedienung von vernetzten EPAM-Applikationen (s.a. Kap. 7.25 S.129)
- Transparente Images (PCX-Bilder): eine beliebige Farbe kann als Transparentfarbe definiert werden
- Textliste mit formatiertem Fliesstext: kursiv, fett, unterstrichen
- Verbesserter Sprachsupport: Texte können nun im Tabellenblatt „Text“ mehrsprachig verwaltet werden und werden mit Hilfe des Makros „Build Language Texts“ automatisch zugeordnet
- Verbesserter Variablenimport für vernetzte Anlagen. Es können Variablen von verschiedenen PLCs importiert werden (mit Definition verschiedenen Hostnamen und IP-Adressen)
- Bilder und Fonts können in eigenen Verzeichnissen, vom Projekt getrennt verwaltet werden (`PATH_IMG=`, `PATH_FNT=`)
- Globale Objekte werden automatisch in der `Initpage` eingefügt. `Initpage` wird automatisch definiert.
- Anzeige des PLC-Status (Stop/RUN) mit Systemvariable `s_plcstate` möglich
- Export der Alarmhistory als CSV (Aktion: `AlarmExport=CSV`)
- Support des Datentyps `IEC_DT` (Datum/Zeit-Eingaben, Timerfunktionen)
- Neue Option „Type=Password“ zur Eingabe von Passwörtern mit beliebigen Fonts
- Neue Systemvariablen `s_newpage`, `s_pageidx`, `s_pagename` zum direkten Seitenwechsel aus der PLC bzw. zur Anzeige der aktuellen Seite in der PLC (in Verbindung mit dem Objekt `Sys2PLC`)
- Verbesserter Projektdownload: Test auf Verbindung, Target-ID und Diskfull
- Makro „Rebuild all“ prüft ob alle verwendeten Pages auch projektiert sind

1.6 Hinweise zur Anpassung bestehender Applikationen

1.6.1 Änderungen in der EPAM V3.50

In der EPAM V3.50 wurde der Fehler 495 behoben. Im Zusammenhang mit dem ScreenSaver kann dieser Fehler zu einem Einfrieren von EPAM führen.

Details s.a.: http://bugs.easypagemachine.com/show_bug.cgi?id=495

Alle Änderungen

Eine komplette Liste aller Änderungen und behobenen Fehler seit V3.40 finden Sie unter: <http://bugs.easypagemachine.com/query.cgi> (Status: closed, Product: EPAM, Version: 3.40)

1.6.2 Änderungen in der EPAM V3.40

Aktion #PagePrev arbeitet neu wie folgt:

Seitenwechsel auf Seiten gleicher Dimension werden in einem Stack gespeichert (Ringbuffer mit den letzten 100 Seiten; die erste Seite wird nicht überschrieben, somit kann immer zum Ausgangspunkt zurück gesprungen werden). Mit #PagePrev kann in diesem Stack eine Ebene zurück gesprungen werden. #PrevPage kann nicht mehr für Fenster verwendet werden. Hierfür kann der Befehl Close oder Close=Window verwendet werden.

1.6.3 Änderungen in der EPAM V3.30

Hinweise zu Geräten mit WindowsCE-Betriebssystem:

Verzeichnis-Struktur auf WindowsCE Geräten:

- Die CompactFlash-Karte ist mit „StorageCard“ bezeichnet. (nicht „C:“)
 - Alle EPAM spezifischen Dateien befinden sich im Verzeichnis \StorageCard\EPAM\
 - Im EPAM Verzeichnis befinden sich folgende Unterverzeichnisse
- | | |
|---------|---|
| BACKUP | ...Backup Verzeichnis (*.INI und *.DAT Dateien) |
| DATA | ...Daten Verzeichnis (*.DAT) |
| FNT | ...Zeichensätze (*.TTF) |
| INI | ...*.INI Dateien |
| IMG | ...Bilder (*.PCX, *.ICO) optional |
| PROJECT | ...EPAM Projekt-Dateien |

Zeichensätze

Unter WindowsCE werden standard Windows True Type Fonts (*.TTF) verwendet. Die Zeichensätze werden nicht mehr sprachspezifisch mit dem im Projekt verwendeten Unicode-Zeichen erstellt! Daher wird für diese Fontdateien (TTF) i.d.R. wesentlich mehr Speicherplatz benötigt (z.B Arial Unicode MS → ca. 24MB!) → ggf. muss eine grössere CF-Karte eingesetzt werden oder die Windows TTF-Fonts müssen mit Standard Windows-Fonteditoren projektspezifisch angepasst werden.

Zeichensätze (*.TTF) sind im Verzeichnis FNT abgelegt und können beim Download (Option: Download Image/Fonts) optional übertragen werden.

Zeichensätze mit dem Attribut Fett (Bold) werden unter WindowsCE breiter als auf dem Desktop dargestellt.

Hierbei handelt es sich um eine „Eigenschaft“ von WindowsCE. Um eine gleiche Darstellung auf dem Entwicklungs-PC und dem Zielsystem zu erreichen wird das Fontattribut „SemiBold“ verwendet.

Aufgrund dieser Unterschiede kann es bei der Portierung von Applikationen von VxWorks (HPG-200/300 XVC-600) auf WindowsCE zu geringfügigen Abweichungen kommen bzw. Anpassung notwendig sein. Sollten Anpassungen bei den Fonts nötig sein, so können diese zentral im Tabellenblatt „Fontmap“ gemacht werden.

RAM-Drive

Das EPAM-Runtime System wird nach Power Up ins Verzeichnis \EPAM kopiert und von dort gestartet. Somit ist gewährleistet, dass die Dateien auf dem Compact Flash aktualisiert werden können. (Hinweis: Zugriff auf eine geöffnete Datei ist unter WindowsCE nicht möglich)

Ist die Option „Ramdrive“ aktiv, so wird auch das Projekt in dieses Verzeichnis kopiert.

Message-Objekt

Wird beim Message-Objekt ein Variablenwert in der Meldung angezeigt, so muss die Meldungsnummer bei allen WindowsCE Geräten als DWORD (32 Bit-Variable) definiert werden!

RemoteControl

Das EPAM-RemoteControl-Objekt unterstützt derzeit nur 256 Farben (8Bit/Pixel). Bei einem Zugriff auf ein anderes WindowsCE Gerät muss darauf geachtet werden, dass der RemoteControl-Server auf diesem Gerät im 256 Farben-Mode läuft bzw. diesen unterstützt. Dies gilt ebenfalls für die Passwort-Funktionalität, die der Server unterstützen muss.

Unsigned Datentypen

Der Überlauf eines vorzeichenlosen Datentyps (z.B. BYTE) von 0 auf 255 (Wert = 0 und Aktion SetVar-1) wurde bisher nicht festgestellt! (s.a. #bug175) Die Limit-Aktion eines definierter Limit1 von 0 wurde daher nicht ausgeführt. Dieser Fehler wurde nun behoben. D.h. bei Wert 0 und Aktion SetVar=-1 wird die Limit Aktion1 ausgeführt oder falls keine Limit-Aktion definiert ist bleibt der Wert 0.

Wird der Überlauf von 0 auf 255 bei SetVar-1 gewünscht, so muss neu die Limit1-Aktion SetVar=Limit2 definiert werden!

Release Mode

Release Mode EPAM_NOEXIT=yes (Epam.ini) verhindert das Beenden von EPAM auch mit Tastatur und mit dem Taster auf XV-3xx/4xx Geräten!

Alle Änderungen

Eine komplette Liste aller Änderungen und behobenen Fehler finden Sie unter:

<http://bugs.easypagemachine.com/query.cgi> (Status: closed, Product: EPAM)

1.6.4 Änderungen in der EPAM V3.20

Die neue Funktion „Überagerte Ojekte“ bedingt, dass die Darstellung der Objekte auf die Dimension des Objekts begrenzt wird (Clipping). Bisher wurde z.B. ein PCX-Bild eines Signals komplett dargestellt, auch wenn die Dimension (DX, DY) des Objekts zu klein definiert wurde. Neu wird nur der Teil des Bildes ausgegeben, der innerhalb der Objektdimension DX, DY liegt. Dies bedingt, dass bestehende Applikationen diesbezüglich überarbeitet werden müssen.

Die Funktion des Screensavers wurde in Bezug auf VarState und VarValue korrigiert und entspricht nun der Dokumentation.

Der Bargraph kann nun auch mit Grenzwerten z.B. 0, 100 als Differenzbalken (Fill=X) projiziert werden.

Die Dimension des Meter-Objektes wurde geringfügig angepasst, damit die Darstellung unter Windows und auf dem Zielsystem übereinstimmt.

Bei der Animation von Button, Switch, Radiobutton wird der Buttoninhalt nicht mehr verschoben.

In der Alarmhistory wird nicht mehr der älteste Alarm überschrieben, sondern die Einträge erfolgen wie folgt:

1. Ist der älteste Alarm inaktiv, wird er durch den neuen Alarm überschrieben.
2. Ist der älteste Alarm aktiv, wird der älteste inaktive quittierte Alarm überschrieben.
3. Wenn kein inaktiver quittierter Alarm vorhanden ist, wird der älteste inaktive Alarm überschrieben.
4. Wenn kein inaktiver Alarm vorhanden ist, wird der älteste Alarm überschrieben.

Bei Geräten mit Infrarot-Touch erfolgt beim Start des Gerätes ein Touchtest. Im Fehlerfall wird eine Seite mit den ausgefallenen Lichtschranken dargestellt (Fadenkreuz). Nach einem Timeout wird die Applikation gestartet und der Fehler in der Systemvariablen s_toucherror angezeigt. (s.a. Kap. Systemvariablen)

1.6.5 Änderungen in der EPAM V3.10

Bestehende Projekte können mit dem Makro „Update Objects“ aktualisiert werden. Hierbei wird automatisch ein neues Tabellenblatt „Fontmap“ eingefügt, das eine Reihe von Fontdefinitionen beinhaltet. Wurden in der Applikation anwenderspezifische Fonts verwendet, die in dieser Liste nicht enthalten sind, so müssen diese Fonts noch definiert werden. (Makro: „New Font“). Anschliessend sollte das Projekt mit „Rebuild All“ neu übersetzt werden.

Unsichtbare Passworteingaben mit Password-Font müssen neu mit der Option „Type=Password“ realisiert werden.

Verbesserter Sprachsupport:

Sie können mit Hilfe des Makros „Build language texts“ und der Option „Insert undefined text“ alle bestehenden Texte ins Tabellenblatt „Text“ übernehmen und anschliessend alle Sprachen und Texte in diesem einen Tabellenblatt zentral verwalten. Wenn Sie Änderungen im Tabellenblatt „Text“ vornehmen, können Sie anschliessend das Projekt mit „Build language texts“ aktualisieren (Option „Insert undefined text“ ist inaktiv). Hierbei werden allen Defaulttexten automatisch die entsprechenden fremdsprachigen Texte im gesamten Projekt zugeordnet.

2 Installation

2.1 Easy PageMachine EPAM

CD einlegen und im Menü EPAM anwählen. EPAM wird anschliessend auf dem angegebenen Laufwerk und Pfad installiert (aktuelle Versionen von EPAM finden Sie auch im Internet unter www.gesys.ch) bzw. auf der EPAM-Homepage www.easypagemachine.com.

Nach erfolgreicher Installation finden Sie folgende Dateien und Verzeichnisse vor:

..\..\EPAM\	
FontBuilder-Unicode	...Utility zur Erstellung von eigenen Zeichensätzen (nur VxWorks)
Images	...enthält PCX-Images
Samples	...enthält EPAM-Demoprojekte
Target	...aktuelles Runtime-System (WinEPAM und PocketPC)

Unter der Programmgruppe EPAM finden Sie eine lauffähige Demo-Applikation, sowie eine Readme.TXT Datei mit aktuellen Hinweisen.

2.2 Einstellungen in Excel

Nachfolgende Beschreibungen beziehen sich auf Excel 2007. Mindestanforderung ist Excel 2000 oder neuer.

2.2.1 EPAM-Makros installieren

Die Installation der EPAM-Makros erfolgt beim Ausführen des Setup-Programms automatisch ins Verzeichnis C:\...\Microsoft Office\Office\XLStart. Die EPAM-Makros werden somit automatisch beim Start von EXCEL installiert.



Hinweis!

Falls die EPAM Makros nicht automatisch installiert werden, ggf. die Einstellungen für Makro-Sicherheit anpassen oder das AddIn manuell über „Excel-Optionen - weitere Befehle – AddIns – Gehe zu - durchsuchen“ installieren.

2.2.2 Symbolleisten

In Excel2000, 2003 können die EPAM-Symbolleisten frei angeordnet werden. In Excel 2007 sind diese Möglichkeiten leider eingeschränkt.

Über Symbolleiste für den Schnellzugriff anpassen können die Benutzerdefinierten Symbolleisten (EPAM) in die Schnellstartleiste eingefügt werden. Hierzu unter Befehle „Add Ins“ auswählen und mit Hinzufügen in die Symbolleiste für Schnellzugriff einfügen.

Die Reihenfolge der EPAM Symbolleisten kann durch löschen der Leiste und beenden/neustarten von EXCEL geändert werden. Die zuletzt gelöschte Symbolleiste wird dabei am Ende dargestellt.



Hinweis!

Es wird empfohlen die EPAM-Projekte im XLS-Format zu speichern. Das Speichern im neuen XLSX-Format (Excel 2007) führt zu Performanceverlusten bei der Projektbearbeitung (längere Makroabarbeitungsdauer).

2.3 Zeichnungsprogramm

EPAM beinhaltet keinen eigenen Grafikeditor. Zur Bilderstellung kann jedes Standard-Zeichenprogramm verwendet werden, welches das Bildformat PCX Version 5 unterstützt.

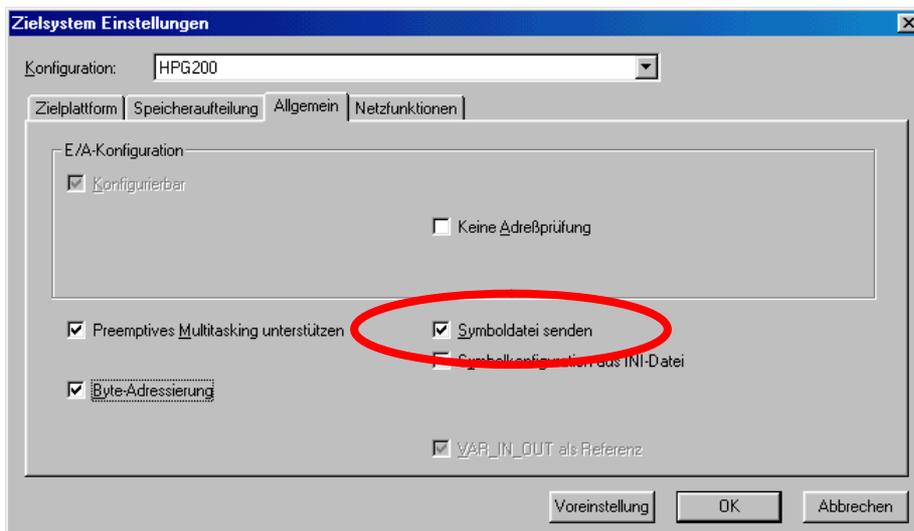
z.B. **Paint Shop Pro**

2.4 Einstellungen in der CoDeSys-Entwicklungsumgebung

Für die Kommunikation mit dem PLC-Runtimesystem sind in der CoDeSys-Entwicklungsumgebung folgende Einstellungen zu machen.

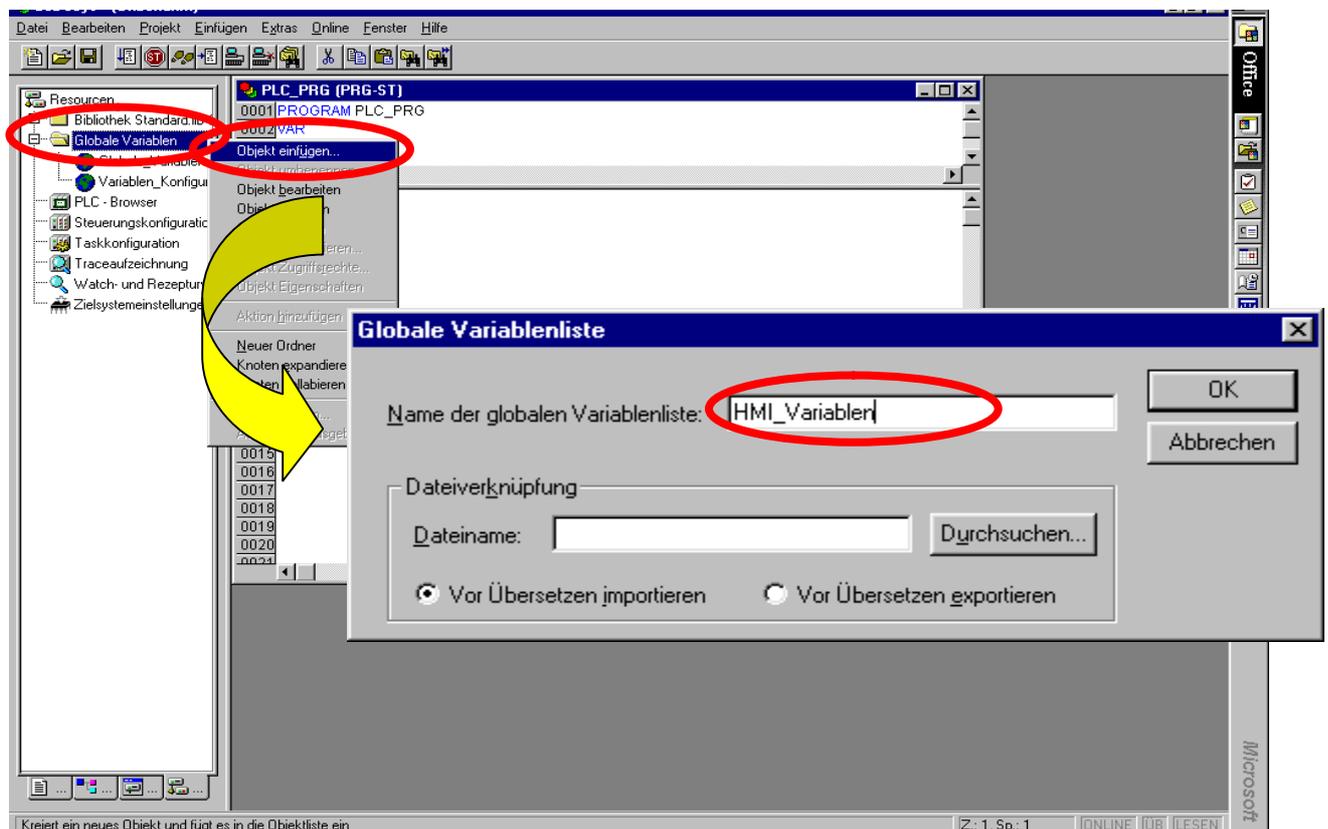
2.4.1 Zielsystem-Einstellungen

Wählen Sie das entsprechende Zielsystem z.B. HPG-200 und aktivieren Sie die Einstellung „Symboldatei senden“.

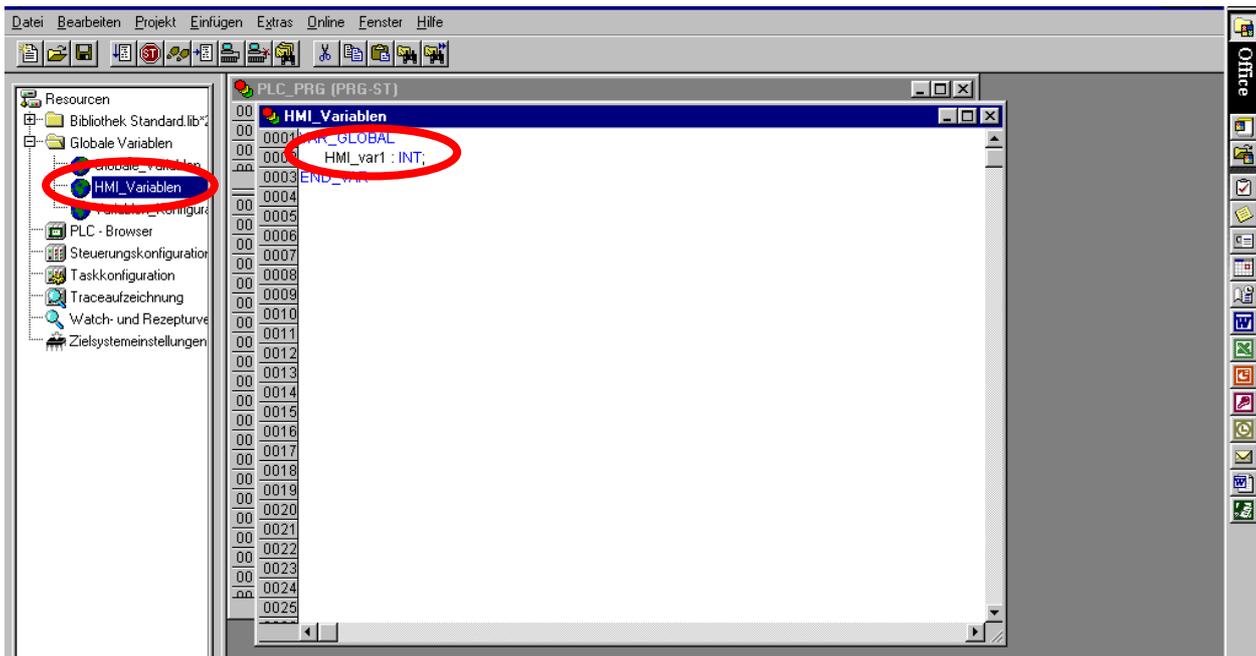


2.4.2 Globale HMI-Variablen definieren

Aus Performance-Gründen wird empfohlen, nur die für den Variablen austausch mit der Visualisierung benötigten globalen Variablen in die Symboldatei zu exportieren. Daher sollte ein eigener Bereich für die globalen HMI-Variablen im Menü „Ressourcen-Globale Variablen-Objekt einfügen“ (rechte Maustaste) definiert werden.

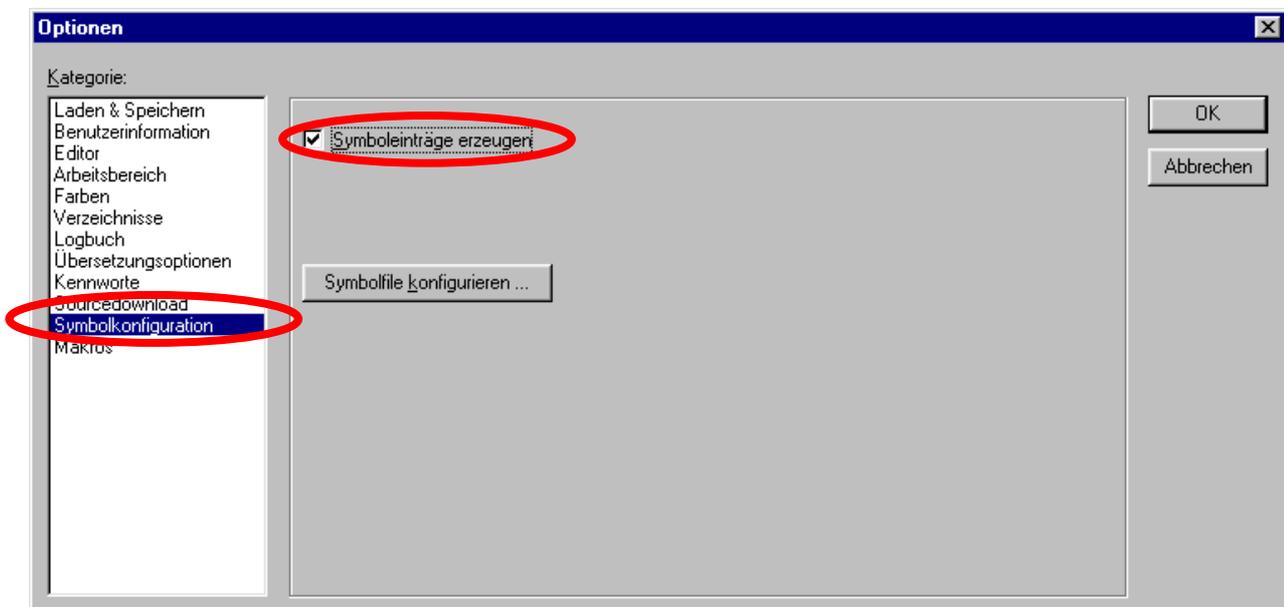


Anschließend können im Variablen-Arbeitsblatt „HMI-Variablen“ die globalen HMI-Variablen definiert werden.



2.4.3 Projekt Optionen

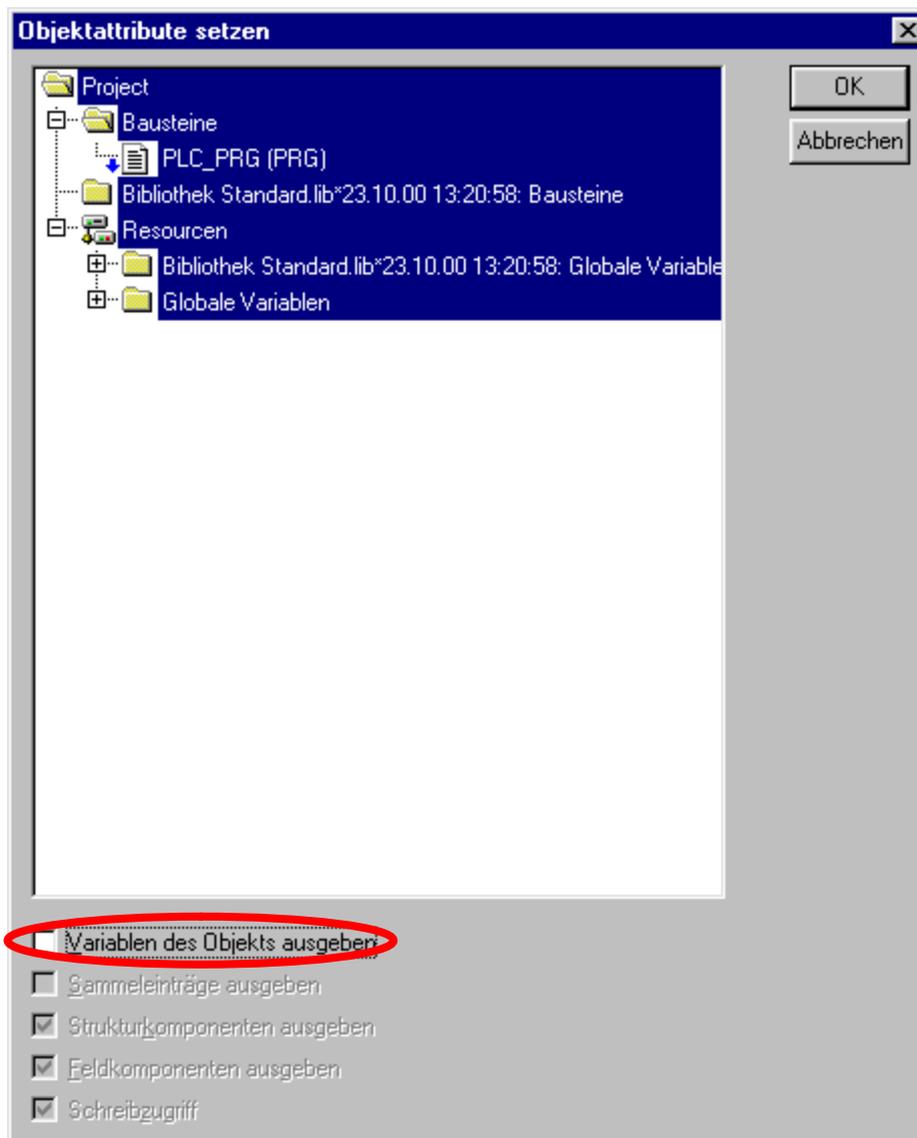
Damit die Symboldatei erstellt wird, muss unter „Projekt-Option-Symbolkonfiguration“ der Eintrag „Symboleinträge erzeugen“ aktiviert werden.



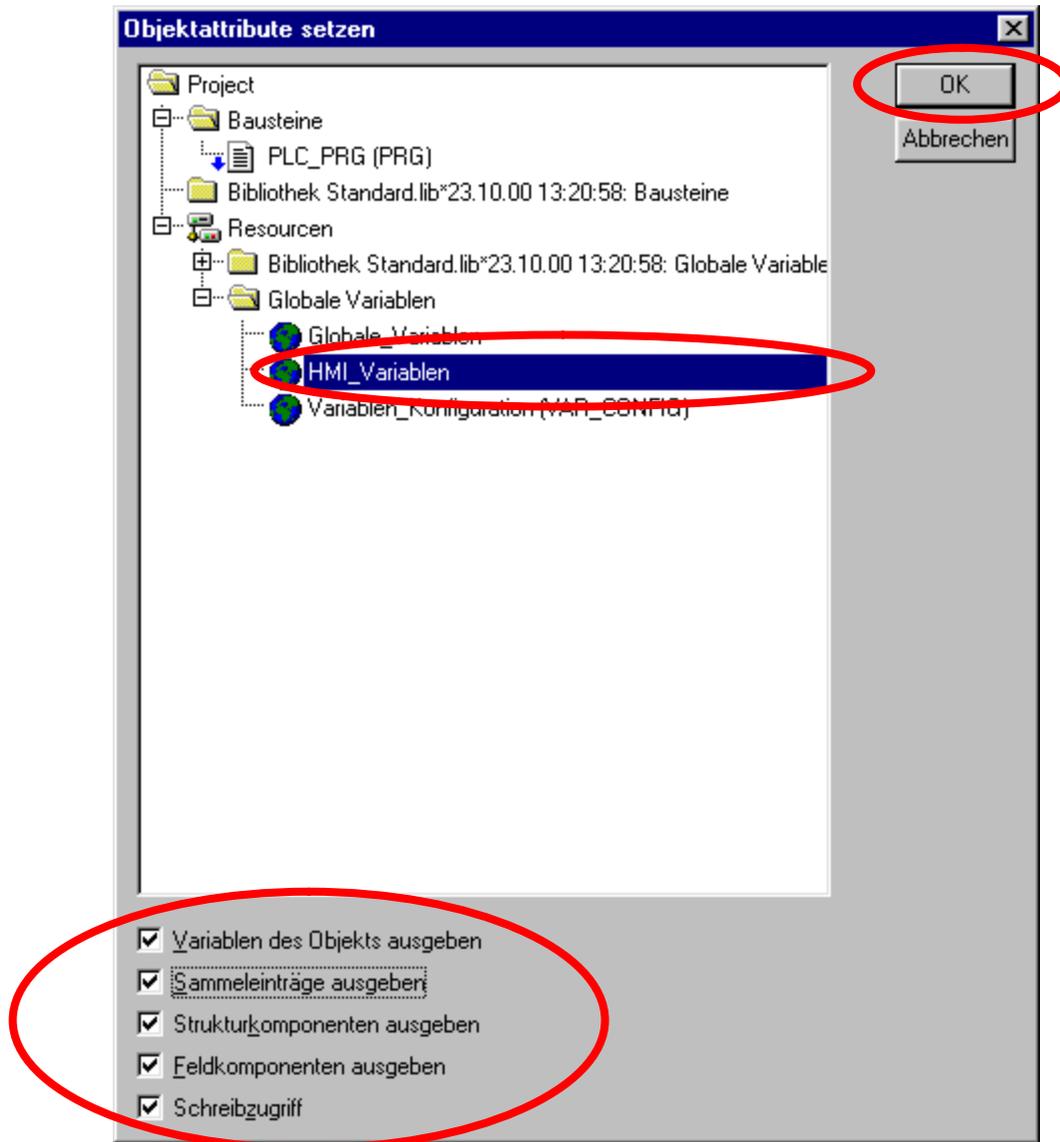
Hinweis!

Ist im Menü „Online“ die Simulation aktiviert, erscheint in „Projekt-Optionen“ der Eintrag „Symbolkonfiguration“ nicht ⇒ Im Menü „Online“ die Simulation deaktivieren.

Unter „Symbolfile konfigurieren“ kann eingestellt werden, welche Variablen in die Symboldatei exportiert werden sollen. Default sind alle Objekte markiert. Deaktivieren Sie den Eintrag „Variablen des Objektes ausgeben“.



Anschließend selektieren Sie den Eintrag „Globale Variablen-HMI-Variablen“, aktivieren die gewünschten Optionen und schliessen das Menü mit „OK“.

**Hinweis!**

Die Symboldatei wird nicht bei jeder Änderung im PLC-Projekt neu erstellt. Das Projekt sollte daher bei Änderungen innerhalb der HMI-Variablen mit „Projekt alles übersetzen“ neu erstellt werden.

2.5 EPAM Runtime-System

Das EPAM Runtime-System ist normalerweise auf dem CompactFlash des Geräts vorinstalliert. Nachfolgend sind kurz die wichtigsten Eigenschaften der einzelnen RTS aufgeführt.

2.5.1 Runtime-System für Geräte mit VxWorks

Bei Geräten mit VxWorks (HPG-200/300, XVC/XCC-601) hat das CompactFlash die Laufwerksbezeichnung „C:“

Verzeichnisstruktur Laufwerk „C:“

C:\Backup	...EPAM Backup-Verzeichnis (Kopie von INI-Files und Rezepturen)
C:\Data	...EPAM Data-Verzeichnis (Rezepturen und Datalog-Dateien)
C:\EPAM	...EPAM RTS und Projekt
DNLD	...temporäres Download-Verzeichnis (wird nach Download gelöscht)
\Project	...EPAM Projekt
epam.out	...EPAM RTS
restore.out	...Utility zur Wiederherstellung von verlorenen INI und Rezeptdateien
drvarti.out	...SymARTI-Treiber
rcs.out	...RemoteControl-Server
boxpc.out	...Display-Konfiguration für XCC/XVC-601
EPAM.INI	...EPAM-Einstellungen (s.a. Tabellenblatt EPAM)
C:\INI	...EPAM INI-Files (sysvar.ini, alarm.ini)
Autoexec.INI	...Start EPAM-RTS und RemoteControl-Server
Config.ini	...wird in EXCEL als Verbindungstest beim Projekt-Download verwendet
„Target“.SYS	...z.B. HPG200.sys wird in EXCEL zum Test des eingestellten Target verwendet

Das komplette Runtime-System kann mit Hilfe des Programms „SetupTargetFirmware-Vx.x.exe“ erstellt werden.

2.5.2 Runtime-System für Geräte mit WindowsCE

Bei Geräten mit WindowsCE (XVH-300, XV-4xx, EP-300, CPC-650) hat das CompactFlash die Laufwerksbezeichnung „StorageCard“

Verzeichnisstruktur Laufwerk „StorageCard“

StorageCard\EPAM\Backup	...EPAM Backup-Verzeichnis (Kopie von INI-Files und Rezepturen)
StorageCard\EPAM\Data	...EPAM Data-Verzeichnis (Rezepturen und Datalog-Dateien)
StorageCard\EPAM\FNT	...EPAM projektspezifische Windows-TrueType Fonts
StorageCard\EPAM\IMG	...EPAM projektspezifische Bilder (optional, wenn PATH_IMG= in EPAM.INI gesetzt ist)
StorageCard\EPAM\INI	...EPAM INI-Files (sysvar.ini, alarm.ini)
StorageCard\EPAM	...EPAM RTS und Projekt
DNLD	...temporäres Download-Verzeichnis (wird nach Download gelöscht)
\Project	...EPAM Projekt
wceepam.exe	...EPAM RTS
drvarti.dll	...SymARTI-Treiber
cesysutl.dll	...HW-spezifische Funktionen (z.B. Backlight, IP-Adresse, etc.)
EPAM.INI	...EPAM-Einstellungen (s.a. Tabellenblatt EPAM)
HMI.BAT	...Start EPAM-RTS

Das komplette Runtime-System kann mit Hilfe des Programms „SetupTargetFirmwarexxx-Vx.x.exe“ erstellt werden.



Die Windows Fonts (*.TTF) werden im Verzeichnis EPAM\FNT global abgelegt und müssen daher nur beim ersten Projekt-Download übertragen werden. (s.a. Option: Download Image/Fonts)

Einschränkungen der WindowsCE Version

Das AlarmMail-Objekt ist nicht implementiert.

2.5.3 WinEPAM Runtime-System für PC/IPC

Das WinEPAM Runtime-System besteht aus den Dateien:

Winepam.exe	...EPAM RTS
Drvari.dll	...SymARTI-Treiber
Drvrs7.dll.dll	...AT-S7-Treiber
Drvmpi.dll	...MPI-Treiber (benötigt eine Hilscher DP Karte)

Diese Dateien befinden sich im EPAM-Verzeichnis auf dem Entwicklungs-PC. Der Projekt Download ist nur in ein Verzeichnis möglich. In der Defaulteinstellung wird im EPAM-Projektverzeichnis ein Ordner „Target“ erstellt. In diesem Verzeichnis befinden sich alle notwendigen Projekt-Dateien (ohne winepam.exe und drvari.dll).

Weiters wird ein Link auf dem Desktop erstellt, der den Aufruf von WinEPAM.exe beinhaltet.

Um das EPAM-Projekt auf eine PC/IPC zu installieren, muss nur das Target-Verzeichnis, das WinEPAM RTS und der Link auf dem Desktop kopiert werden. Der Link und die Pfadangaben in EPAM.INI müssen ggf. entsprechend der Verzeichnis-Struktur auf dem PC/IPC angepasst werden.

Kommandozeilen-Parameter WinEPAM

Aufruf: winepam script.txt [Option]

Option	Beschreibung
/?	Info Anzeige
/alarmini=off	Alarmhistory Alarm.ini wird nicht geschrieben (Default: on)
/usercolors	Alle Farbdefinitionen aus Imagedateien verwenden (inkl. 0-15)
/toff	Touch driver aus (Bedienung mit Maus)
/rWxH	Fenstergrösse (Default: Fullscreen)
/wposXxY	Fensterposition
/wname=name	Fenstername
/plc	Kommunikation zur SPS aktiv (Default: off)

Der Aufruf von WinEPAM sollte ohne Angabe der Projektdatei (script.txt) erfolgen, damit die Einstellungen aus der EPAM.INI Datei übernommen werden. Beim Aufruf mit einer Projektdatei läuft EPAM im Simulationsmode. (wie Kammera aus EXCEL)



WinEPAM kann mehrfach auf einem PC gestartet werden, der Window-Name (/wname=) muss dann aber unterschiedlich sein!



**Das Runtime-System für PC ist Runtime lizenzpflichtig!
Bestellbezeichnung: WinEPAM Runtime-Lizenz Standard-PC**

Einschränkungen der Windows Version

Systemfunktionen wie Touch-Beep, Backlight-Einstellung, IP-Adresse anzeigen/ändern sind in der Windows Version nicht möglich. Die Kommunikation zur Steuerung erfolgt über TCP/IP, d.h. es wird ein PC mit Ethernet bzw. WLAN-Adapter benötigt.

Das AlarmMail-Objekt ist nicht implementiert.

2.5.4 Runtime-System für PocketPC (Target PocketPC-240x320)

Das Runtime-System für PocketPC (z.B. iPAQ mit PocketPC2002) ist für PDAs mit ARM-Prozessor und WindowsCE (PocketPC2002) ausgelegt. Getestet wurde das RTS auf einem HP-iPAQ 5450 mit XScale Prozessor (PXA270), WindowsCE 4.2 (PocketPC2002) und WLAN.



Grundsätzlich sollte das RTS auf jedem PDA mit ARM und WindowsCE funktionieren. Dies kann aber nicht garantiert werden und muss im Einzelfall geprüft werden!

Runtime-System für den PocketPC:

wceepam.exe	...EPAM RTS
drvarti.dll	...SymARTI-Treiber
cesysutl.dll	...HW-spezifische Funktionen (z.B. Backlight, IP-Adresse, etc.)

Diese Dateien befinden sich im EPAM-Verzeichnis Target\PocketPC2002 auf dem Entwicklungs-PC. Der Projekt Download ist nur in ein Verzeichnis möglich. Um ein EPAM-Projekt auf den PocketPC zu übertragen verwendet man am besten Microsoft Active Sync.

Kopieren Sie das Runtime-System mit dem Windows-Explorer in das Verzeichnis das mit dem PocketPC synchronisiert wird (z.B. Eigene Dateien\My PocketPC Documents\EPAM). Anschliessend wählen Sie beim Projekt-Download ebenfalls dieses Verzeichnis und synchronisieren dann Ihren PocketPC. (hierfür in Microsoft Active Sync-Options-Sync Options Files aktivieren)

Auf dem PocketPC können Sie dann EPAM durch Aufruf von wceepam.exe starten.



**Das Runtime-System für PocketPC ist Runtime lizenzpflichtig!
Bestellbezeichnung: EPAM Runtime-Lizenz PocketPC**

Einschränkungen der Pocket-Windows Version

Systemfunktionen wie Touch-Beep, Backlight-Einstellung, IP-Adresse anzeigen/ändern sind in der Pocket-PC Version (z.B. für iPAQ) nicht möglich. Die Kommunikation zur Steuerung erfolgt über TCP/IP, d.h. es wird ein Pocket-PC mit Ethernet- bzw. WLAN-Adapter benötigt.

Das AlarmMail-Objekt ist nicht implementiert.

2.5.5 EPAM-Konfiguration auf dem Zielsystem (EPAM.INI)

Das EPAM-RTS wird mit Hilfe der Datei EPAM.INI (Tabellenblatt EPAM) konfiguriert. Im Normalfall werden diese Einträge automatisch in EXCEL gesetzt. Die Datei kann aber auch manuell editiert werden, um bestimmte Einstellungen zu testen. (Achtung! Die Datei wird beim Projekt-Download überschrieben!)

[ENVIRONMENT]	Comment
EPAM_VARLIST=_DRVVLST.TXT	Dateiname der Variablen-Liste (fix)
EPAM_DRVPARAM=DRVPARAM.TXT	Dateiname der Driver Parameter (fix)
EPAM_PROJECT=Project.TXT	EPAM-Projektname
PATH_EPAM=\StorageCard\EPAM\PROJECT	EPAM-Projektpath
PATH_DATA=\StorageCard\EPAM\DATA	EPAM-Datapath
PATH_BACKUP=\StorageCard\EPAM\BACKUP	EPAM-Backuppath
PATH_INI=\StorageCard\EPAM\INI	EPAM-INI-Path
SHOW_INFO=	Debug-Info: SHOW_INFO=t ...zeigt Bildaufbauzeit in ms links oben (Default: off)
EPAM_NOEXIT=NO	Disable Exit-Button in Dialog Box (EPAM-Error) (Default: No)
EPAM_RDONLY=NO	Disable alle Schreibzugriffe auf Disk (Default: enable)
INIT_PICTURE=startup.PCX	Startup-Bild
RUNMODE=0	Wird nicht mehr benötigt
EPAM2RAM=NO	Install RAMDrive (EPAM: bzw. \EPAM)
EPAM_NOBEEP=NO	Disable Touch Beep
VIDEO_MODE=VESA640X480X256	Videomode (nur VxWorks, Default: VGA)
PROJECTVERSION=V1.0	Project Version
PROJECTNAME=Project	Project Name
PROJECTPROGRAMMER=	Project Programmer
PROJECTTARGET=CPC600-10	Project Target
RAMDRV_SIZE_KB=8192	Grösse des RAM Drive in kB
PATH_LOG=\EPAM\LOG	Path für Datalog
LOCALHOST=xxx.xxx.xxx.xxx	IP des local host (Default: 127.0.0.1)
PATH_IMG=	EPAM-Imagepath (absolut)
PATH_FNT=\StorageCard\EPAM\FNT	EPAM-Fonts (absolut)
ORIENTATION=Landscape	Orientierung des Bildschirms (Landscape or Portrait)
INPUT_DEVICE=Touch	Input Device: Touch/Maus
LOAD_LANGUAGE=-1	Default: -1 = alle Sprachen in RAMDrive laden; 1 = nur 1 Sprache in RAMDrive laden Yes = alle 256 Farben von PCX-Bild laden (0-255); USERCOLOR=Bild.PCX → feste Farbalette mit 256 Farben laden
USERCOLORS=No	No = Kommunikationstreiber deaktivieren
DRIVER=Yes	No = Keyboardeingaben deaktivieren
KBD=Yes	No = Keyboardeingaben deaktivieren

Alle gelb markierten Einträge werden nicht über EXCEL-Makros gesetzt und können ggf. manuell geändert werden.

3 Projektrealisierung

EPAM wurde mit dem Ziel entwickelt, grafische Benutzeroberflächen möglichst einfach und rasch zu erstellen. Die Grundidee für die Vorgehensweise bei der Projektrealisierung ist daher „**Fast prototyping**“. D.h. es wird ein Funktionsmuster erstellt und unmittelbar getestet. Hiermit soll eine professionelle Projektabwicklung (Pflichtenheft, Konzept, Realisierung, Inbetriebnahmen, Test, etc.) durch die Möglichkeit einer sehr frühen Verifizierung des Pflichtenheftes durch den Kunden, anhand eines funktionsfähigen Musters, effizient unterstützt werden.

Unser Vorschlag für eine mögliche Projektabwicklung ist daher wie folgt:

3.1 Zusammenstellung der Anforderungen

Die Anforderungen an eine grafische Benutzeroberfläche sollten in einem Pflichtenheft festgehalten werden. Speziell zu beachten sind hier Computerkenntnisse des Endanwenders, Dialogsprachen etc.

3.2 Strukturierung der Bildseiten

Dies bezieht sich auf die Aufteilung der verschiedenen Ein-/Ausgaben auf verschiedene Bildseiten. Bei diesem Schritt ist es empfehlenswert die verschiedenen Benutzerprofile zu betrachten. z.B. Operatorprofil für Produktion- und Einrichtparameter oder Serviceprofil für Einstellungs- und Maschinenparameter etc.

Eine Benutzeroberfläche ist dann einfach zu bedienen, wenn die gewünschten Aktionen mit möglichst wenig Eingaben erreicht werden können.

3.3 Definition des Bildseitenlayouts

Dieser Schritt liefert Grundlagen für die Erstellung der Bilder, Texte und ggf. Zeichensätze. Erfahrungsgemäss ist die Bilderstellung ein erheblicher Teil des Zeitaufwandes in einem Visualisierungsprojekt und Änderungen im Bildseitenlayout bedingen oft auch erhebliche Änderungen in den Bildern. Daher ist es sinnvoll im ersten Schritt nur mit Texten zu arbeiten und den Entwurf unmittelbar zu testen. Eine optische Aufbesserung durch Icons und Bilder kann dann später immer noch erfolgen.

3.4 Bilderstellung

Die Bilderstellung für EPAM (Bilder und Icons) erfolgt mit einem Standard-Zeichnungsprogramm z.B. Paint Shop Pro. Bilder für EPAM können im PCX-Format mit 256 Farben erstellt werden.

3.5 Realisierung mit Excel

Nun können Sie schon in die Realisierung einsteigen und Ihre Bildseiten im Excel definieren und verknüpfen. Siehe auch Kap. 6 Projektierung mit Excel, S.35

3.6 Anbindung an die Steuerung

Die Anbindung an die Steuerung erfolgt durch Definition der symbolischen Variablennamen in den Spalten VarValue, VarState, Limit1 und Limit2. Die Variablennamen können hierbei durch Import der Symboldatei aus der CoDeSys-Programmierungsumgebung übernommen werden.

Kommunikations-Prinzip:

Die Kommunikation zwischen EPAM und PLC basiert auf dem Prinzip Read/Write von einzelnen Variablen bzw. ganzen Strukturen (Records). D.h. der Kommunikationstreiber fordert Istwerte von der Steuerung variablenweise an. **Geänderte Sollwerte werden unmittelbar als einzelne Variable an die Steuerung gesendet und wieder zurückgelesen. D.h. ein Sollwert kann von der PLC wieder zurückgesetzt werden, was dann auch in der Visualisierung ersichtlich ist.**

Beim Starten von EPAM werden alle Variablen eingelesen und initialisiert. Ansonsten werden die Variablen zyklisch abgefragt und nur die geänderten Werte auf dem Bildschirm aktualisiert. Es werden dabei nur diejenigen Variablen abgefragt, die zur Zeit benötigt werden. D.h. die Variablen aller gleichzeitig geöffneten Bildseiten (Fenster).



Das Lesen und Schreiben von Variablen erfolgt unmittelbar, d.h. NICHT synchron zum PLC-Zyklus!

3.7 Dokumentation

Aufgrund des transparenten ASCII-Datenformats wird die Projektdokumentation praktisch automatisch miterstellt. Zusätzliche Hinweise können Sie über die Excel-Funktion „Einfügen-Notiz“ an beliebigen, nicht mit dem Objektpräfix ‚#‘ versehenen Zeilen, einfügen. Diese Notizen sind nur in der Excel-Datei vorhanden und haben daher keine Auswirkung auf Ausführungsgeschwindigkeit bzw. Speicherbedarf auf dem Zielsystem.

Im Idealfall ist die Arbeit nun abgeschlossen und die Benutzeroberfläche ist auf dem Zielsystem lauffähig. In der Praxis wird dieser Ablauf wiederholt werden müssen, da während der Projektrealisierung die Anforderungen aufgrund neuer Erkenntnisse öfters geändert oder erweitert werden. Doch auch dies ist mit EPAM kein Problem, da Änderungen und Erweiterungen, auch nachträglich dank Excel, einfach und rasch durchgeführt werden können.

3.8 Tipps für TouchScreen Applikationen



Verwenden Sie möglichst helle Hintergrundfarben. Dies vermindert die Sichtbarkeit von Fingerabdrücken und verbessert die Ablesbarkeit bei heller Umgebung.



Beschränken Sie sich bei den Farben möglichst auf die Grundfarben, rot, grün, blau, gelb, magenta, cyan, schwarz und weiss. Bei Flachdisplays verfügen nur diese Farben über einen optimalen Ablesewinkel.



Definieren Sie Ihre touchaktiven Bereiche „fingergerecht“ (ein Finger ist kein Mauszeiger!).



Nutzen Sie die Möglichkeiten zum Ein- bzw. Ausblenden von Objekten und stellen Sie dem Anwender möglichst nur die Aktionsfelder zur Verfügung, die im Moment benötigt werden. Dadurch wird die Anwendung intuitiv und einfach zu bedienen. Die Vorteile des TouchScreens kommen dadurch erst richtig zum Tragen.



Aktivieren Sie den „Beep“ als akustischen Feedback.



Treffen Sie Massnahmen zur Vermeidung von Fehlbedienungen wie z.B. Screensaver, zusätzliche Sicherheitsabfragen bei kritischen Aktionen etc.

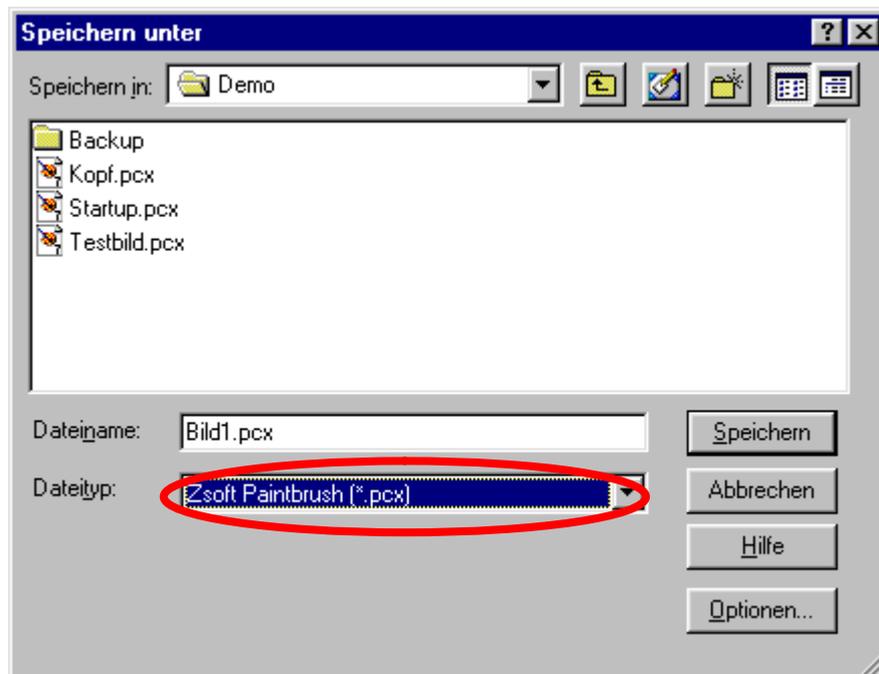
4 Bilderstellung

4.1 PCX-Bilder erstellen

Paint Shop Pro starten, „Datei-Neu“ anwählen und Bilddimension sowie Farbtiefe (gemäss Zielsystem: z.B. VxWorks 256 Farben, WindowsCE EP-300: 16 Millionen Farben) anwählen.



Bild erstellen und im PCX-Format, Version 5 im Projektverzeichnis mit „Datei-Speichern unter...“ speichern.



Hinweise für die Bearbeitung von 256 Farben-Bilder mit Farbpaletten:

Aus Performance-Gründen sollten Sie alle PCX-Bilder im selben Format mit derselben Farbpalette mit 256 Farben erstellen. Ansonsten werden die Farbpaletten der PCX-Bilder zur Laufzeit geladen und die PCX-Bilder zur Laufzeit ins richtige Format konvertiert.

Der EPAM-Makro „PCX-Colortranslation“ konvertiert 16-Farben-Bilder (z.B. Icons) in 256-Farben-Bilder und passt die ersten 16 Farben von 256-Farben-Bildern entsprechend der EPAM-Farbpalette an (Diese entspricht den 16-Windowsfarben).

Benutzerdefinierte Farbpalette

Alternativ kann auch mit einer benutzerdefinierten Farbpalette gearbeitet werden. Hierbei können alle 256 Farben frei definiert werden. Durch den Eintrag USERCOLOR=bild.pcx in EPAM.INI wird die benutzerdefinierte Farbpalette beim Start von EPAM aus dem bild.pcx geladen. In diesem Fall müssen alle Bilder mit dieser Farbpalette erstellt werden. (USERCOLOR=Yes bewirkt dasselbe, allerdings wird hier die Farbpalette bei jedem Bild im Projekt neu geladen).

Im Tabellenblatt UserColor können optional die verwendeten UserFarben 0-255 und die zugehörigen RGB-Werte für die Darstellung im Wizard definiert werden.

Farbnummer/Name	R (0-255)	G (0-255)	B (0-255)
0	0	0	0
1	128	0	0
...			

4.2 Icons erstellen

EPAM-Icons werden im selben Format wie PCX-Bilder, aber mit der Dateierweiterung MyIcon.ICO gespeichert (Dateierweiterung muss eingegeben werden, sonst wird die Datei als *.PCX gespeichert).

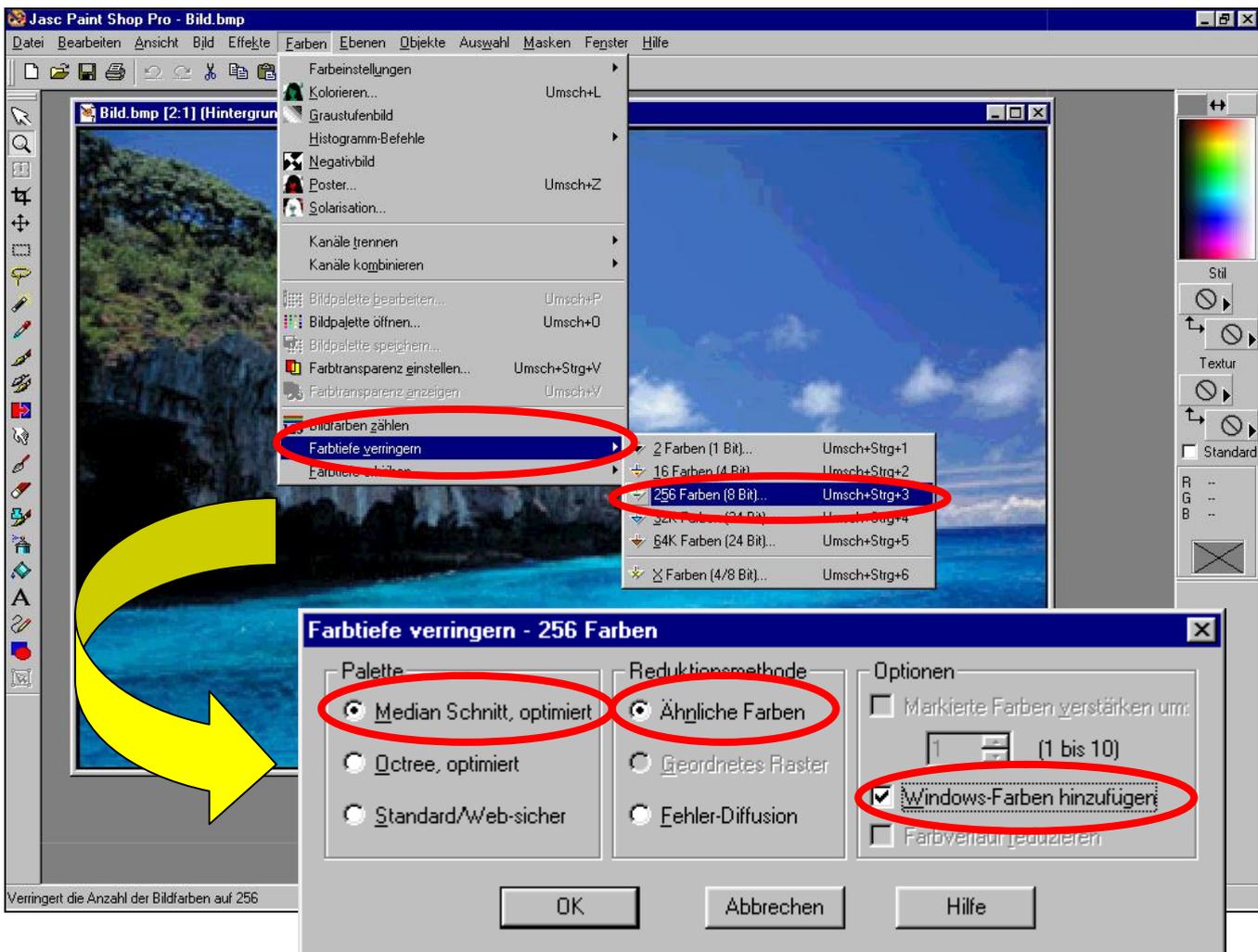
Hinweise für die Bearbeitung von 256 Farben-Icons mit Farbpaletten: (für Zielsysteme mit nur 256 Farben)

Die Icons sollten mit der Einstellung 16 Farben erstellt und anschliessend mit dem Makro PCX-Colortranslation („256“) in 256-Farbenbilder konvertiert werden. Werden Icons mit 256 Farben erstellt, so müssen alle Bilder und Icons dieser Seite dieselbe Farbpalette verwenden.

4.4 Bilder von Digitalkamera oder gescannt

Hinweise für die Bearbeitung von 256 Farben-Bilder mit Farbpaletten: (für Zielsysteme mit nur 256 Farben)

Fotorealistische Bilder können im Paint Shop Pro auf 256 Farben reduziert werden. Wählen Sie hierfür im Menü „Farben-Farbtiefe verringern-256 Farben“ mit Palette „Median Schnitt, optimiert“, Reduktionsmethode „Ähnliche Farben“ und der Option „Windows-Farben hinzufügen“. Alle so erstellten Bilder müssen anschliessend mit dem EPAM-Makro „PCX-Colortranslation“ konvertiert werden. Hierbei wird die Farbpalette aller PCX-Bilder im Projektverzeichnis so angepasst, dass die ersten 16 Farben der Farbpalette den in EPAM verwendeten Windows-Farben entsprechen. Dies ist notwendig, damit z.B. ein Button auf diesem Fotobild in der richtigen Farbe dargestellt wird. Befinden sich im Projektverzeichnis Bilder mit 16 Farben (z.B. Icons), so werden diese ins Format mit 256 Farben konvertiert, hierbei werden die restlichen 240 Farben der Farbpalette mit der Farbe schwarz initialisiert. Somit ist es möglich, fotorealistische Bilder mit Icons zu mischen.



Einschränkung

Bei der Darstellung eines fotorealistischen Bildes mit 256 Farben (volle Farbpalette), wird die Farbpalette des entsprechenden Bildes geladen. D.h. wenn mehrere Bilder auf derselben Bildseite dargestellt werden sollen, müssen diese Bilder dieselbe Farbpalette verwenden. Unter Paint Shop Pro kann im Menü „Farben-Bildpalette öffnen“ die gewünschte Farbpalette für diese Bilder geladen werden. Es können aber PCX-Bilder, die nur die ersten 16 Farben benutzen (z.B. Icons) mit fotorealistischen Bildern gemischt werden.

4.5 Transparente PCX-Bilder

Die Abmasse von PCX-Bilder sind immer über eine Bildhöhe und eine Bildbreite definiert. Das bedeutet, sie bieten keine beliebigen Konturen und beinhalten deshalb immer Bildinformationen auf einem rechteckigen Bildhintergrund. EPAM bietet nun die Möglichkeit, jedem projektierten PCX-Bild (.pcx) eine Farbe zu definieren, welche transparent interpretiert wird.

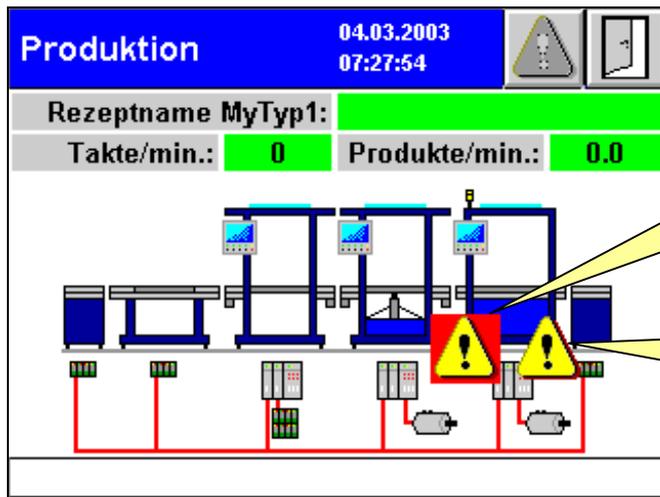
Beispiel:

PCX-Bild mit einer Bildinformation (gelb-schwarzes Warnungssymbol) auf einem roten Bildhintergrund



Definition eines PCX-Bildes mit und ohne Option Transparency=red

Object	Text/File	Font	X	Y	DX	DY	Color	Back-color	Format	Action	Limit1	Limit2	Action Limit1	Action Limit2	Var-Value	Var-Type	Var-State	Option
#Signal	AlarmOn.pcx		205	150	34	34	black	grey										
#Signal	AlarmOn.pcx		245	150	34	34	black	grey										Transparency=red



Ohne Option Transparency=red

Mit Option Transparency=red wird anstelle der Farbe rot der darunterliegende Hintergrund dargestellt.

5 Zeichensatzerstellung

Mit Hilfe des Tabellenblatts „Fontmap“, den EPAM-Makros „New Font“ und „Build Fonts“ sowie des Utilities „FontBuilder-Unicode“ (nur VxWorks) werden die Zeichensätze erstellt. Dabei werden die gewünschten Windows TrueType-Schriftarten automatisch in das auf dem Zielsystem benötigte Format konvertiert und erstellt. (nur VxWorks)

Bei Geräten mit WindowsCE werden direkt die Windows-Fonts (*.TTF) verwendet.



Hinweis!

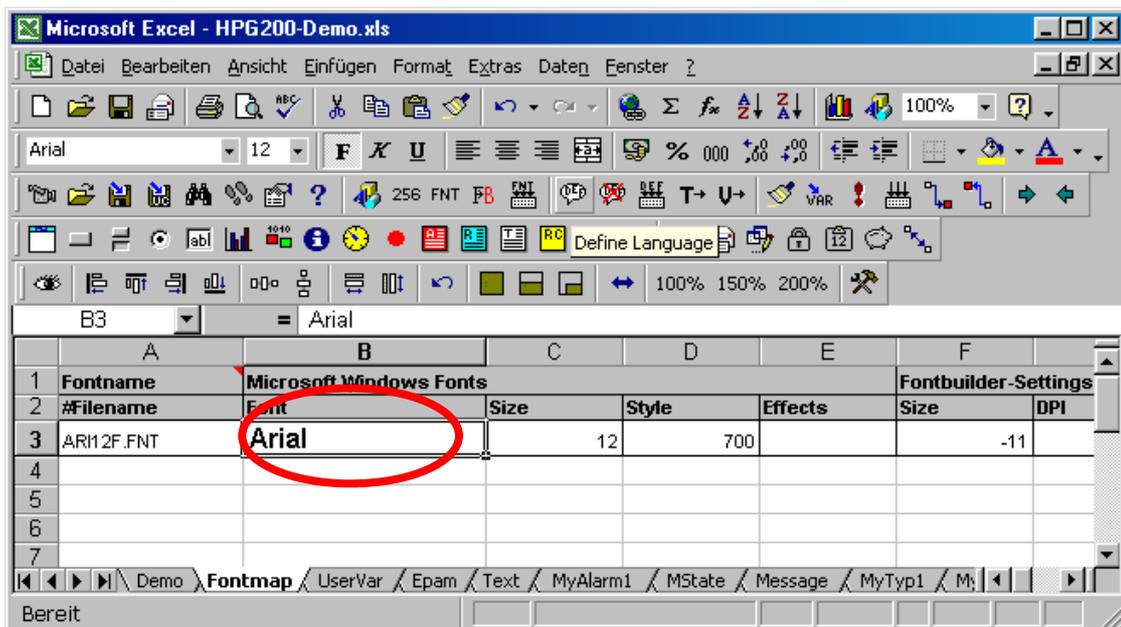
Es werden nur TrueType Schriftarten unterstützt.

5.1 Definieren von Zeichensätzen

Führen Sie den EPAM-Makro „New Font“ aus. Geben Sie den gewünschten Namen (max. 8 Zeichen) des Zeichensatzes an und schliessen das Menü mit „OK“.



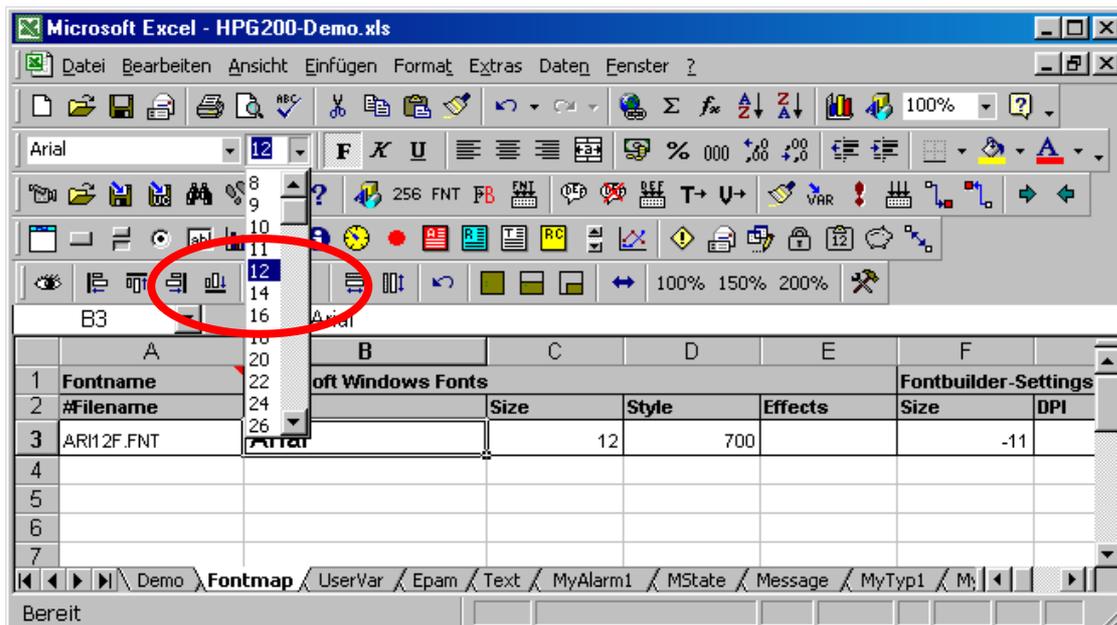
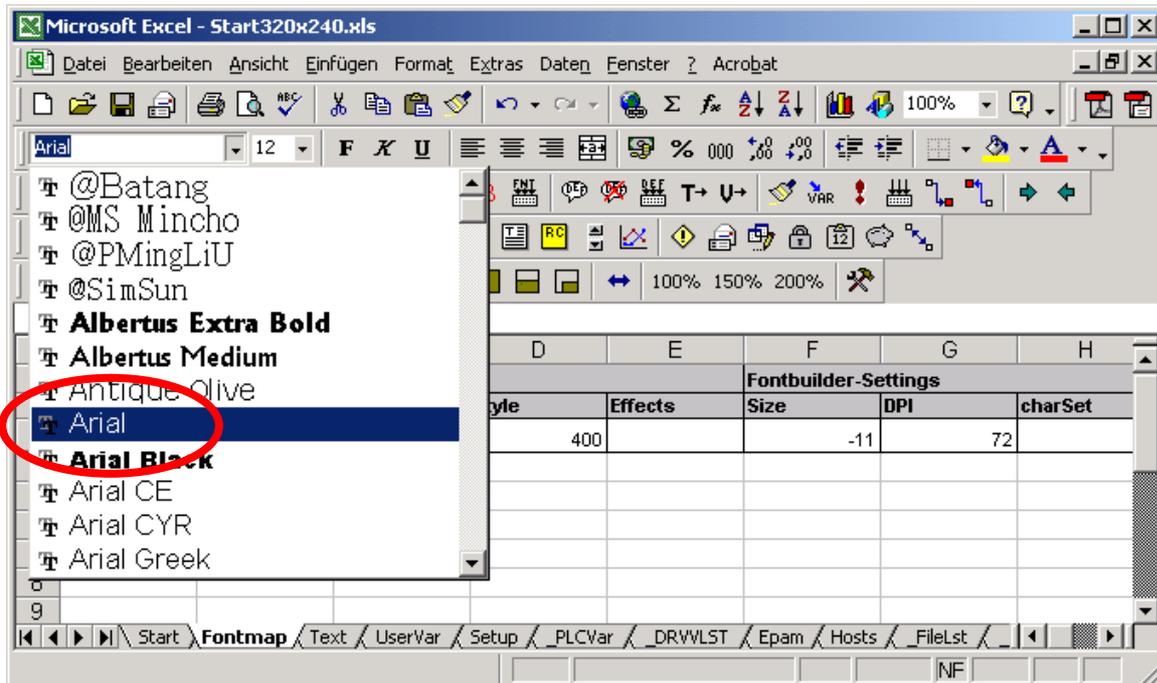
Im Tabellenblatt „Fontmap“ wird nun Ihr Zeichensatzname eingetragen. Markieren Sie nun folgendes Feld...

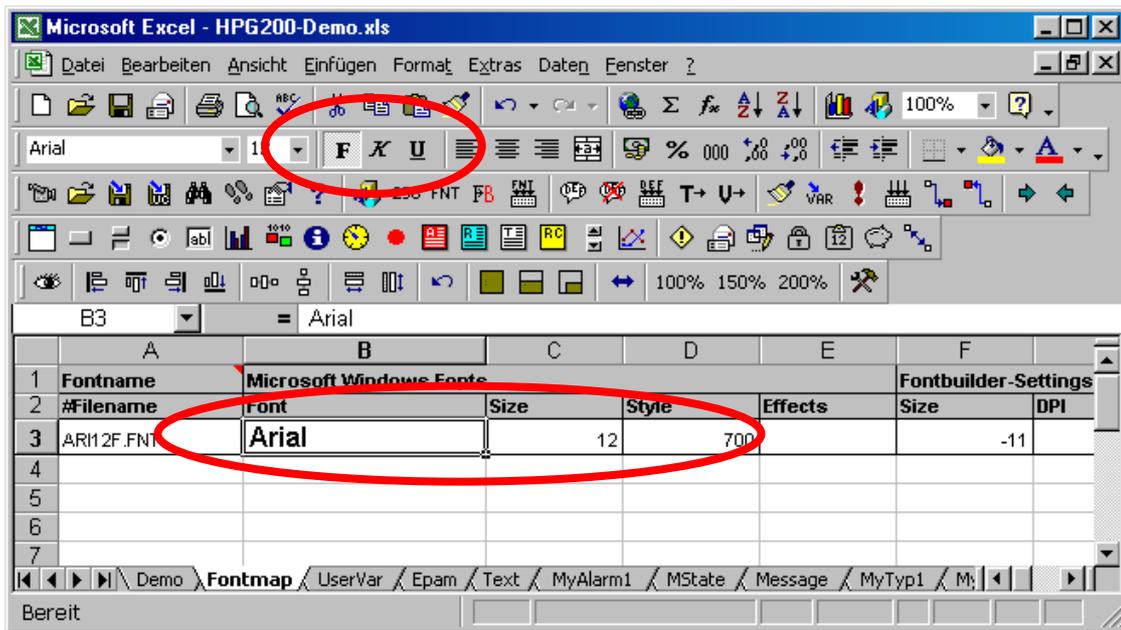


Hinweis!

Dieser Text wird beim späteren Ausführen des EPAM-Makros „Build Fonts“ automatisch durch den Namen der selektierten TrueType Schriftart ersetzt.

Definieren Sie nun die Eigenschaften Ihres gewünschten Zeichensatzes (TrueType Schriftart, Schriftgröße, Schrifteffekt)...

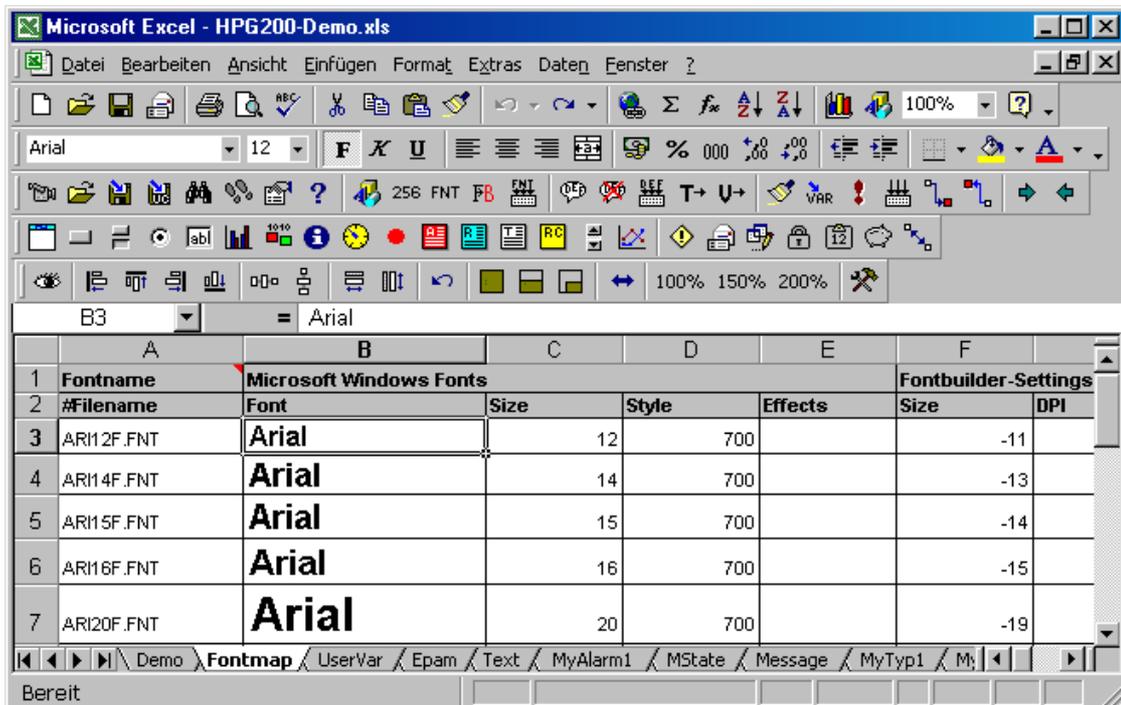




Hinweis!

Der Textinhalt in Spalte Font, Size, Style und Effects muss nicht manuell editiert werden. Diese Parameter werden beim späteren Ausführen des EPAM-Makros „Build Fonts“ automatisch gesetzt.

Diese Schritte führen Sie nun für Ihre weiteren Schriftsätzen aus...



Hinweis!

Bei VxWorks sind folgende Fonts vordefiniert und können nicht geändert werden:
 System72, Sysfnt72, Sys06x11, Arial12

5.2 Erstellen von Zeichensätzen

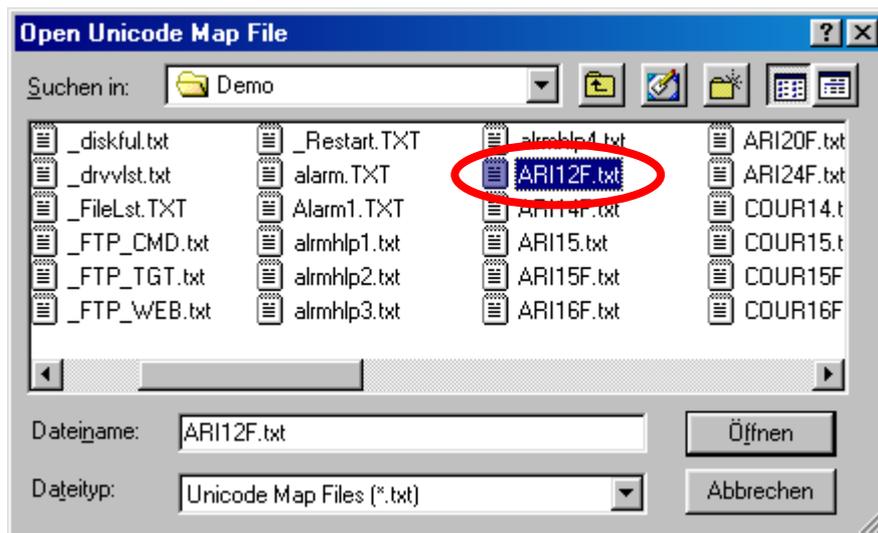
Dieses Kapitel ist nur für Geräte mit VxWorks Betriebssystem relevant!

Führen Sie nun den EPAM-Makro „Build Fonts“ aus... Es werden nun sämtliche Zeichensatz-Dateien (*.FNT) sowie die zugehörigen Map-Dateien (gleicher Name wie Font, jedoch mit Dateierdung .TXT) erstellt.

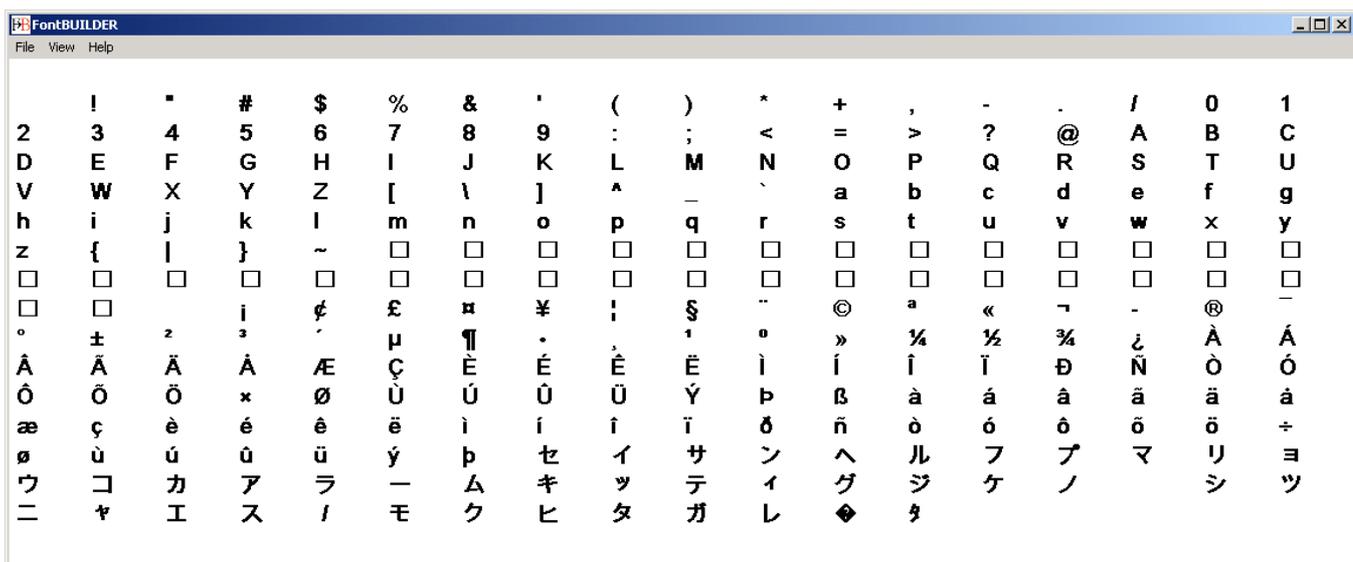
5.2.1 Einschränkungen

Mit Hilfe des Utilities „FontBuilder-Unicode“ und den erzeugten Map-Dateien, müssen die erstellten Fonts noch geprüft werden.

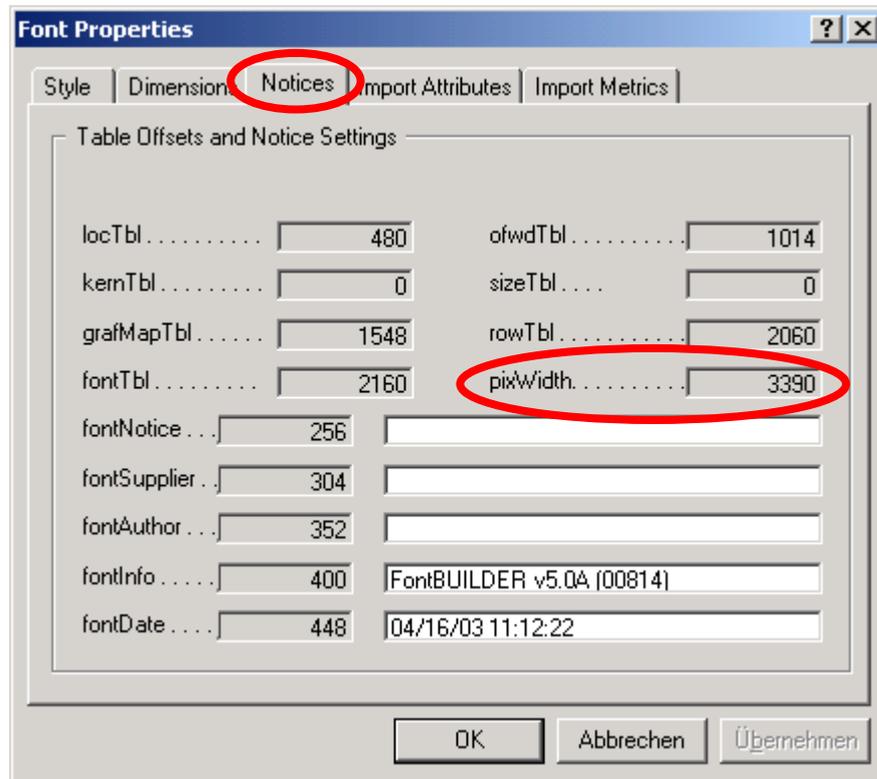
Öffnen Sie in „File-Mapfile“die gewünschte Map-Datei.



Der neu erstellte Font wird nun mit allen benötigten Zeichen dargestellt:



Prüfen sie nun folgende Eigenschaften:



Im Menü „View-Properties“, wird im Feld pixWidth die gesamte Breite des erstellten Zeichensatzes angezeigt. Dieser **Wert muss <= 32767 sein**. D.h. bei einer durchschnittlichen Breite der Zeichen von 10 Pixel, ist **die Anzahl der verfügbaren Zeichen pro Sprache auf maximal ca. 3200 limitiert!**

5.3 Unicode-Support

5.3.1 Funktionsweise

Fremdsprachen können als Unicode-Sprache definiert werden. Die Eingabe der Texte erfolgt mit Hilfe des Microsoft-Officetool Visual-Keyboard oder mit Word im Menü „Einfügen-Symbol“ und der im Office mitgelieferten Unicode Zeichensätze (z.B. Arial-Unicode-MS).

Alle Textdateien die Unicode-Texte beinhalten (EPAM-Sprachdateien, Meldungen, Alarmer, etc.), werden im Format Unicode-Textdatei gespeichert (EPAM-Makro „Save as Unicode-Text“ bzw. direkt in Word mit „Speichern unter“, Dateityp: „Codierter Text“).

Beim Aufruf von EPAM via EPAM-Makro „Start EPAM“ werden alle Unicode-Textdateien wieder in „normale“ ASCII-Datei umgewandelt. Dabei wird eine Mapdatei fb_map.txt erstellt, die alle für diese Applikation benötigten Unicodezeichen enthält. Mit dieser wird anschließend mit Hilfe des Makros „Build Fonts“, aus einem Windows-TrueType Font (z.B. Arial Unicode MS) ein EPAM-kompatibler Font erstellt. (nur VxWorks)

Bei Geräten mit WindowsCE werden direkt die Windows-Fonts (*.TTF) verwendet.

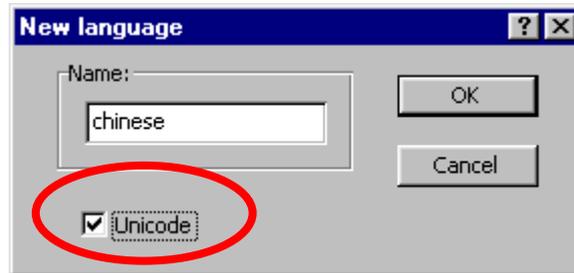


Hinweis!

Die Eingabe von Fremdsprachen-Texten in EXCEL (auch Unicode-Zeichen) kann mit Hilfe des Office Tools „Visual-Keyboard“ und Installation der entsprechenden Tastaturtreiber erfolgen. Das Visual-Keyboard (VkeyInst.EXE) kann von der Microsoft-Homepage heruntergeladen werden: <http://www.microsoft.com/downloads> Search: „Visual Keyboard“

5.3.2 Definition einer Unicode-Sprache

Unter Excel ist es möglich eine Fremdsprache (EPAM-Makro „Define Language“) als Unicode-Sprache zu definieren:



**Die Defaultsprache (Spalte B) kann nicht als Unicode-Sprache definiert werden!
Sprachabhängige Fonts müssen eindeutige Namen haben.**

Es werden nun am Ende der Tabelle zwei neue Spalten für Text/File, Font und ein Unterverzeichnis mit dem Sprachnamen angelegt. Die Spalte Font ist mit dem Kommentar „LanguageUC“ gekennzeichnet (dieser darf nicht gelöscht werden!).

5.3.3 Eingabe der Unicode-Texte

Menütexte

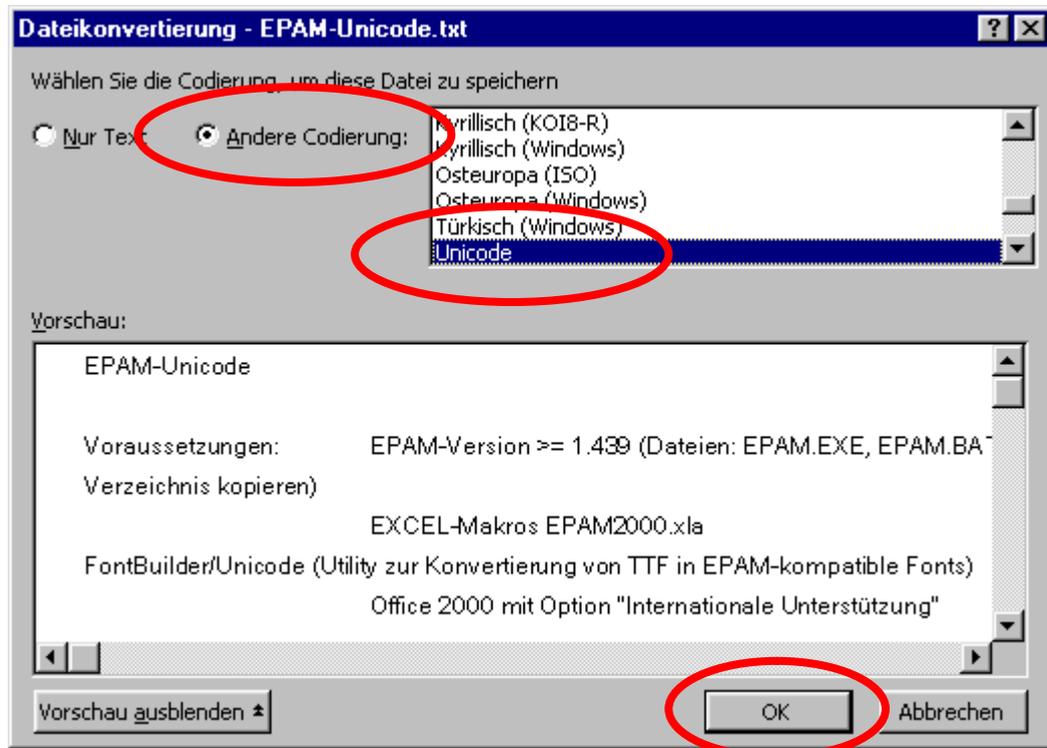
In der Sprachspalte können nun Unicode-Texte z.B. mit dem Zeichensatz „Arial Unicode MS“ angezeigt werden. Die Eingabe der Texte erfolgt entweder direkt mit dem Microsoft-Officetool „Visual-Keyboard“ oder in Word über Menü „Einfügen-Symbol“, Schriftart „Arial Unicode MS“. Anschliessend können die Texte mit Kopieren-Einfügen von Word in die Excel-Tabelle übernommen werden.

Meldungen, Alarme

Meldungs- und Alarmtexte können ebenfalls mehrsprachig definiert werden. Die Texte werden in den entsprechenden Tabellenblättern analog zu den Menütexten definiert. Beim Aufruf des EPAM-Makros „Start EPAM“ werden die Alarm und Meldungstexte automatisch als eigene Textdateien in den entsprechenden Sprach-Unterverzeichnissen angelegt. Dieser Vorgang kann auch manuell mit dem EPAM-Makro „Save as Unicode Text“ erfolgen. Hierbei wird nur das aktuell angewählte Excel-Tabellenblatt als Unicode-Textdatei gespeichert (2-Byte-Code).

Textdateien für Objekt Textlist

Normale Textdateien die mit Hilfe des Objekts Textlist angezeigt werden sollen, werden am einfachsten direkt mit Word erstellt und anschliessend mit „Datei-speichern unter“, Dateityp: „Codierter Text“, mit folgender Option gespeichert.



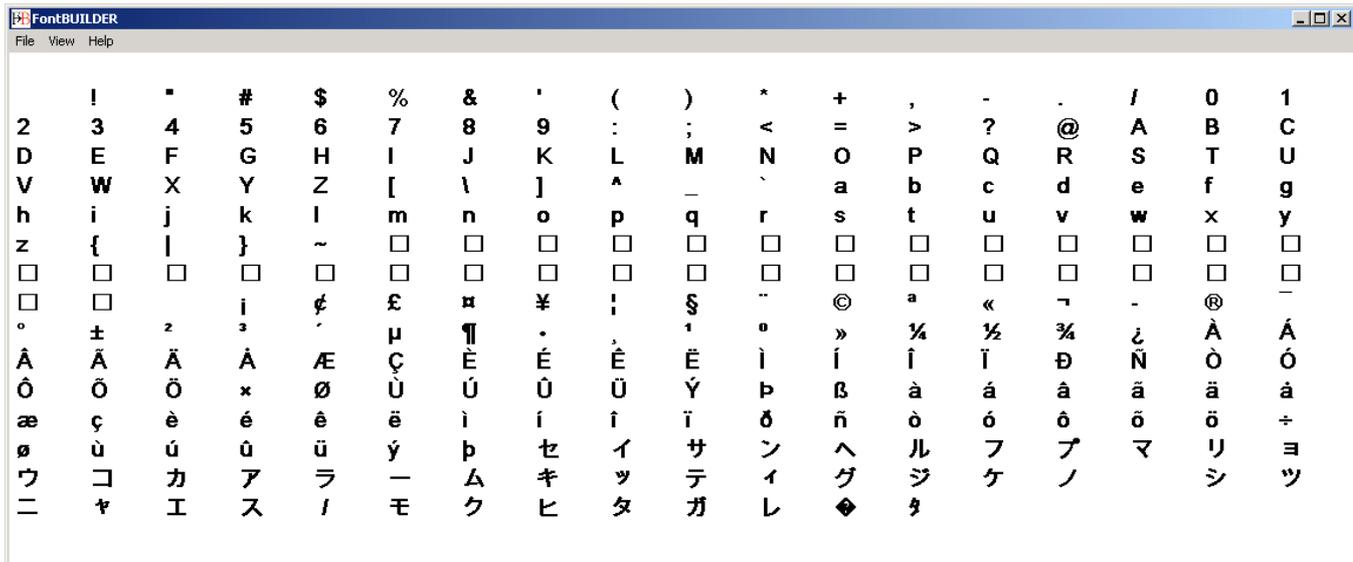
5.3.4 Start EPAM

Beim Aufruf von EPAM werden alle Unicode-Textdateien automatisch wieder in „normale“ Textdateien konvertiert, wobei alle Unicodezeichen als Sonderzeichen im Format \xHHHH dargestellt werden (HHHH = Hexcode). Im Font werden diese Zeichen ab der Position 257 eingefügt. Gleichzeitig wird eine Mapdatei fb_map.txt erzeugt, mit dem anschliessend mit Hilfe des Makros „Build Fonts“, aus einem Windows-TrueTypeFont (z.B. Arial Unicode MS) ein EPAM-kompatibler Font erstellt werden kann, der alle benötigten Zeichen enthält. (nur VxWorks)
Bei Geräten mit WindowsCE werden direkt die Windows-Fonts (*.TTF) verwendet.

5.3.5 Definieren von Unicode-Zeichensätzen

Dieses Kapitel ist nur für Geräte mit VxWorks Betriebssystem relevant!

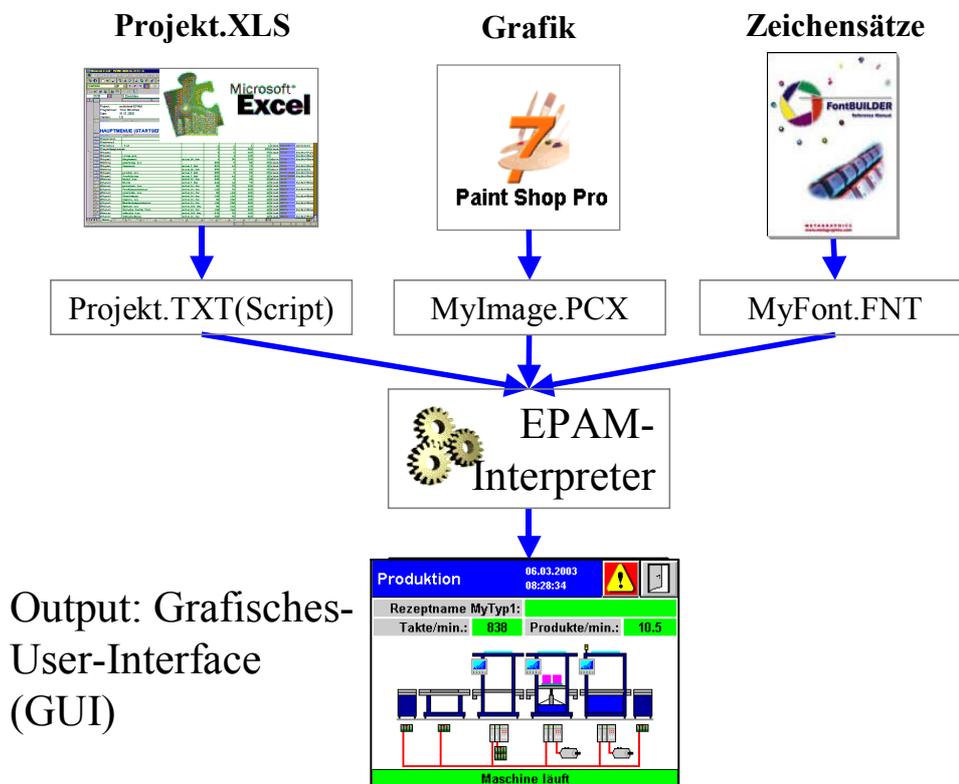
Die Erstellung von Unicode-Zeichensätzen erfolgt analog der Erstellung von Standard-Zeichensätzen (siehe Kap. 5 Zeichensatzerstellung, S.27). Der verwendete Windows Schriftsatz muss allerdings die benötigten Unicodezeichen beinhalten z.B. Arial Unicode MS (ggf. muss der internationale Sprachsupport für MS-Office installiert werden). Sie können dies mit Hilfe des Utilities FontBuilder (Makro: F->B) prüfen, indem Sie die entsprechende Mapdatei des Fonts z.B. UNI14.txt im FontBuilder öffnen (Import – Mapfile). Der Font und die benötigten Unicodezeichen sollten nun in der Anzeige sichtbar sein.



6 Projektierung mit Excel

6.1 Funktionsprinzip

EPAM ist ein Interpreter, d.h. die Objekte und Bildseiten sind in einer strukturierten, tabellarischen ASCII-Datei (sog. Scriptdatei) definiert und werden vom EPAM in eine grafische Präsentation auf dem Bildschirm umgesetzt (vergleichbar mit einem Internet-Browser). Die Scriptdatei enthält die Definitionen der einzelnen Bildseiten (Pages) und der darin enthaltenen Objekte und wird mit Excel erstellt. Grafiken werden als PCX-Bilddateien mit einem Standard Zeichenprogramm erstellt und über den Dateinamen in der Scriptdatei referenziert. Analog verhält es sich mit den Fontdateien.



Beim Aufruf von EPAM (EPAM-Makro „Start EPAM“) wird das Excel-Tabellenblatt als ASCII-Textdatei gespeichert. Hiermit sind die Projektdaten unabhängig von der verwendeten Excel-Version.

Vorteile dieses Konzeptes:

- einfachste Bildseiten und Objektdefinition mit Excel
- transparente, lesbare Datenbasis
- unterstützt verschiedene Hardware-Plattformen (1/4 VGA 320x240 mit 16 Graustufen/Farben, 640x480, 800x600 bis 1280x1024 mit 16 Millionen Farben)
- integrierte Kommunikation zu CoDeSys-PLC über symbolische Namen
- Farben und Zeichensätze frei wählbar
- online Sprachumschaltung, auch Unicode (z.B. chinesische Schriftzeichen)
- keine teure Windows-Entwicklungsumgebung

In Visualisierungs-Applikationen wird immer eine Vielzahl von Listen verwaltet. Daher ist es naheliegend für diese Aufgabe ein speziell dafür entwickeltes Standard-Softwarepaket wie Excel einzusetzen.

Vorteile von Excel:

- übersichtliche Darstellung auf dem Bildschirm und auf dem Ausdruck (Projektdokumentation)
- bestehende Objekte und ganze Bildseiten können einfach kopiert werden
- Formeln und automatisches Ausfüllen von Zellen durch „ziehen“ können genutzt werden
- Änderungen können einfach und rasch durchgeführt werden
- durch die Verwendung von Excel-Makros kann die Projektierung auf kundenspezifische Bedürfnisse angepasst und nach belieben erweitert werden
- Eingabehilfen und Test des Projektes direkt von Excel aus, durch einfaches Anklicken von vordefinierten Icons

Aufruf von EPAM:

Sie können EPAM direkt aus der Excel-Oberfläche mit dem EPAM-Makro „Start EPAM“ starten und die Simulation mit der Taste ESC jederzeit wieder verlassen.

6.2 Struktur der Excel-Tabelle

Zur Definition der Bildseiten stehen Objekte zur Verfügung. Pro Zeile in der Excel-Tabelle kann ein Objekt definiert werden. Eine Bildseite beginnt mit dem Objekt **#Page=name** und endet mit einer Leerzeile bzw. mit der Zeile die nicht mit dem Objektpräfix '#' beginnt. Das Page-Objekt definiert die Bildseite (Position und Dimension) in dem alle nachfolgenden Objekte platziert werden. Ein Objekt beginnt jeweils mit #Objekt. Alle Zeilen, die nicht das Objektpräfix in der ersten Spalte beinhalten, sind Kommentarzeilen.

Die Spalten der Excel-Tabelle beinhalten die Objekteigenschaften und haben folgende Bedeutung:

Spalte	Bedeutung
Object	Objektname z.B. #Page=Name
Text/File	Name einer PCX-Bild- oder ASCII-Textdatei bzw. ein Textstring für die Defaultsprache
Font	optionaler, objektspezifischer Font für die Defaultsprache
X,Y,DX,DY	X, Y-Position, Breite und Höhe des Objektes in Pixel (Bezug links oben!)
Color	Vordergrundfarbe (<i>Farbname oder Farbnummer</i>)
Backcolor	Hintergrundfarbe (<i>Farbname oder Farbnummer</i>)
Format	objektspezifische Formatdefinitionen
Action	Aktion auf Touchberührung
Limit1	unterer Grenzwert: konstanter Wert, PLC-Variable oder Systemvariable für Grenzwert
Limit2	oberer Grenzwert: konstanter Wert, PLC-Variable oder Systemvariable für Grenzwert
Action Limit1	Aktion bei Unterschreitung von Limit1 (Variablenwert < Limit1)
Action Limit2	Aktion bei Überschreitung von Limit2 (Variablenwert > Limit2)
VarValue	PLC-Variable bzw. Systemvariable für Objektwert
VarType	Variablentyp
VarState	PLC-Variable bzw. Systemvariable für Objektstatus
Option	objektspezifische Optionen
C-Funktion	optionale Anwender C-Funktion

6.2.1 Spalte Object

In der Spalte Object werden die EPAM-Objekte definiert. Es können folgende Objekte definiert werden:

Object	Bedeutung	Projektierung
#Alarm	Alarmüberwachung mit History (512 Alarmmeldungen), Zeitstempel für Alarm „Kommen“, „Gehen“ und „Quittiert“	Global
#Alarmlist	Ausgabe der Alarmereignisse innerhalb eines rechteckigen Bereiches	
#Alarmmail	Email Benachrichtigung aufgrund von Alarmereignissen	Global
#Bar	Darstellung eines Wertes in einem rechteckigen Balken	
#Button	nicht einrastender, touchaktiver Bereich	
#DataLog	Aufzeichnung von PLC-Daten/Variablen in eine DataLog-Datei	Global
#DBPasswd	Zentrale User/Passwort-Verwaltung über eine mySQL-Datenbank1	Global
#DBTrace	DataLog-Objekt zur Auzeichnung von Daten in einer mySQL-Datenbank1	Global
#DiagSig	Diagnose Signal, Darstellung von blinkenden Punkten (z.B. Maschinenbild) bei Alarmen zur Diagnose	
#DropDownList	touchaktiver Bereich, Auswahl einer aus mehreren Möglichkeiten aus einer Liste	
#HTMLBrowser	Ausgabe einer HTML-Datei innerhalb eines rechteckigen Bereichs	
#Message	Ausgabe von Meldungen in Form von Text- oder Bild-Information	
#Meter	Darstellung eines Wertes in einem Halb-/Kreis/ benutzerdefinierten Winkel	
#Page	Dimension der Bildseite	
#Password	Passwortverwaltung	Global
#RadioButton	touchaktiver Bereich, Auswahl einer aus mehreren Möglichkeiten	
#Recipe	Rezepturverwaltung	Global
#RecipeList	Ausgabe der Rezeptliste innerhalb eines rechteckigen Bereiches	
#ScreenSaver	Bildschirmschoner	Global
#Scrolllist	Scroll-Liste, Darstellung von Objekten als Scroll-Liste z.B. Parameterliste	
#Signal	Darstellung von Zuständen bzw. statischen Bildern und Texten	
#Switch	einrastender, touchaktiver Bereich	
#Textlist	Ausgabe einer ASCII-Textdatei innerhalb eines rechteckigen Bereichs	
#Trend	Darstellung der DataLog-Datei als Trenddiagramm	
#Variable	Darstellung einer numerischen/alphanumerischen Variable	
#Sys2Plc	Übertragung von EPAM-Systemvariablen an die Steuerung (z.B. Aktive Seite)	Global
#RemoteControl	Fernbedienung eines anderen HPGs via Ethernet (vgl. PC-anyware)	

6.2.2 Spalte Text/File

Die Spalte Text/File enthält den Text eines Objektes, den Dateinamen eines PCX-Bildes/Icons oder den Namen einer Textdatei (zusätzliches Tabellenblatt) mit objektspezifischen Einstellungen. Texte können aus der Textliste (Tabellenblatt „Text“) ausgewählt werden und mit dem EPAM-Makro „Add Text“ ins Tabellenblatt „Text“ eingefügt werden.

Zur Auswahl und Eingabe von PCX-, ICO- und TXT-Dateien kann der EPAM-Makro „Open file“ verwendet werden.

6.2.3 Spalte Font

Die Spalte Font enthält den Dateinamen einer Fontdatei (*.FNT), die für die Darstellung des Textes verwendet wird. Die Auswahl und Eingabe von FNT-Dateien kann via Pulldownmenü erfolgen.

1 Benötigt die Option „EPAM-DB-Client“ auf dem Gerät (derzeit nur für VxWorks verfügbar) und den EPAM-DB-Server im Netzwerk

6.2.4 Spalten X,Y,DX,DY

Die Spalten X, Y, DX, DY definieren die Position und die Dimension des Objektes in Pixel relativ zum Nullpunkt der aktuellen Bildseite. Der Bezugspunkt (Nullpunkt) ist links oben.



Automatische Positionierung

Die Felder *Font*, *X*, *Y*, *DX*, *DY*, *Color* und *Backcolor* können auch leer bleiben. In diesem Fall werden die Werte vom vorherigen Objekt übernommen. D.h. *Font*, *DX*, *DY*, *Color* und *Backcolor* müssen nur bei Änderungen eingetragen werden. Bleiben die Felder X,Y leer, so werden die nach-folgenden Objekte automatisch nacheinander positioniert. Hierbei wird um die aktuelle Breite (DX) in X-Richtung vorgerückt. Bei Überschreitung der aktuellen Bildseitenbreite erfolgt ein Zeilenumbruch entsprechend der aktuellen Objekthöhe (DY). Sollen verschiedene Objekte untereinander in einer Spalte positioniert werden, so muss nur die X-Koordinate für alle Objekte gleich gesetzt werden, die Y-Koordinate wird dann automatisch ermittelt.



EPAM-Wizard

Mit Hilfe des EPAM-Wizards lassen sich auf einfachste Weise die Positionen X, Y und die Dimensionen DX, DY von Objekten ändern.

6.2.5 Spalte Color, Backcolor

Die Felder *Color* bzw. *Backcolor* sowie die Aktionen *Color=x* bzw. *Backcolor=x* können Zahlenwerte bzw. die nachfolgenden Farbnamen beinhalten. Die Standardfarbpalette entspricht der Palette des Zeichnungsprogramms Paint Shop Pro (Standard Windows-VGA-Farbpalette) und lautet wie folgt:

Index	Farbname
0	black
1	dark red
2	dark green
3	brown
4	dark blue
5	dark magenta
6	dark cyan
7	grey
8	dark grey
9	red
10	green
11	yellow
12	blue
13	magenta
14	cyan
15	white

6.2.6 Spalte Format

Die Spalte Format enthält objektspezifische Darstellungsvarianten.

6.2.7 Spalte Action

In der Spalte Action wird die objektspezifische Aktion definiert, die bei der Anwahl des Objektes mit Touch oder Maus ausgeführt werden soll. Mögliche Aktionen sind z.B. Bildwechsel, Variablenwert verändern, etc.



Mehrfache Aktionen

Mit dem Zeichen '&' können mehrere Aktionen (Spalte Action) ausgelöst werden.

z.B.: `SetVar=1 & #Page=name` ...setzt Variable auf 1 und wechselt anschliessend auf die Bildseite *name*

Nach der Aktion #Page=name werden keine Aktionen mehr ausgeführt!

Standard Tastaturtabelle

Die Aktion Key=keycode kann mit normalen ASCII-Zeichen, einem der nachfolgenden Tastennamen bzw. mit dem entsprechenden Tastencode definiert werden.

F1	...Taste F1 entspricht Tastencode \x3b00
F2	...Taste F2 entspricht Tastencode \x3c00
F3	...Taste F3 entspricht Tastencode \x3d00
F4	...Taste F4 entspricht Tastencode \x3e00
F5	...Taste F5 entspricht Tastencode \x3f00
F6	...Taste F6 entspricht Tastencode \x4000
F7	...Taste F7 entspricht Tastencode \x4100
F8	...Taste F8 entspricht Tastencode \x4200
F9	...Taste F9 entspricht Tastencode \x4300
F10	...Taste F10 entspricht Tastencode \x4400
F11	...Taste F11 entspricht Tastencode \x4500
F12	...Taste F12 entspricht Tastencode \x4600
ESC	...Taste ESC entspricht Tastencode \x1b
CursorUp oder CUp	...Taste Cursor up entspricht Tastencode \x4800
CursorDown oder CDown	...Taste Cursor down entspricht Tastencode \x5000
CursorLeft oder Cleft	...Taste Cursor left entspricht Tastencode \x4b00
CursorRight oder CRight	...Taste Cursor right entspricht Tastencode \x4d00
PageUp oder PgUp	...Taste Page up entspricht Tastencode \x4900
PageDown oder PgDn	...Taste Page down entspricht Tastencode \x5100
Home	...Taste Home entspricht Tastencode \x4700
End	...Taste End entspricht Tastencode \x4f00
Insert	...Taste Insert entspricht Tastencode \x5200
Backspace	...Taste Backspace entspricht Tastencode \x08
Return oder Enter	...Taste Return/Enter entspricht Tastencode \x0d
Delete oder Del	...Taste Delete entspricht Tastencode \x5300

6.2.8 Spalte *Limit1*, *Limit2*

Die Spalten Limit1 und Limit2 definieren den objektspezifischen unteren und oberen Grenzwert des Variablenwertes. Der Grenzwert kann als Konstante, als Systemvariable oder als PLC-Variable definiert werden. Grenzwertvariablen müssen vom selben Typ sein wie die VarValue-Variable.



Grenzwerte

Die Grenzwerte Limit1 und Limit2 sind Bestandteil des Wertebereiches. D.h. eine Grenzwertüberschreitung liegt vor, wenn der **Wert kleiner bzw. grösser** ist als Limit1 bzw. Limit2.

6.2.9 Spalte *Action Limit1*, *Action Limit2*

Die Spalten Action Limit1 und Action Limit2 definieren die Aktionen die bei **Unter- bzw. Überschreitung** des Wertebereiches, der durch Limit1 bzw. Limit2 definiert ist, ausgeführt werden sollen. Mögliche Aktionen sind z.B. Farbumschlag, Bildwechsel etc.

6.2.10 Spalte *VarValue*

Die Spalte VarValue enthält den Namen einer PLC-Variable oder einer Systemvariable. Systemvariablen sind globale Variablen im EPAM, die nicht zur Kommunikation mit der PLC bestimmt sind. PLC-Variablen werden in der folgenden Syntax definiert:

[[/Kommunikationstreiber-Name/]Hostname/]Variablenname

Beispiel: /ARTI/PLC/HMIVar1 ...Variable HMIVar1 von PLC mit Kommunikationstreiber ARTI
 PLC/HMIVar1 ...Variable HMIVar1 von PLC mit Default Kommunikationstreiber (=ARTI)
 HMIVar1 ...Variable HMIVar1 von lokaler PLC mit Default Kommunikationstreiber

Treibernamen und Hostnamen sind optional und müssen dann spezifiziert werden, wenn Variablen einer anderen Steuerung gelesen werden sollen.

Mit Hilfe des Pulldown-Feldes können Variablen aus der Liste „UserVar“ ausgewählt werden. Bei der Übersetzung des Projektes wird geprüft, ob alle in EPAM verwendeten Variablen in der Liste „UserVar“ definiert sind und der Datentyp der Variable mit dem Objektdatentyp übereinstimmt.



Indizierte Variablenadressierung

Mit Hilfe einer Systemvariablen z.B. ‚s_index‘ und der Button-Action SetIndex=x bzw. SetIndex können Variablennamen zur Laufzeit geändert und indiziert gelesen werden. Der Variablenname ist in diesem Fall wie folgt zu definieren (Spalte VarValue):

MyVariable%**s_index**%xy

Platzhalter für Index (Name einer Systemvariablen vom Typ: WORD)

Beim Aufbau der Bildseite wird dann automatisch anstelle des Platzhalters %s_index% der aktuelle Wert der Indexvariablen eingefügt und diese Variable abgefragt.

z.B.: MyVariable1xy



Der Bereich der Indexvariablen kann im Tabellenblatt UserVar in den Spalten Limit1 und Limit2 definiert werden.

Anwendung:

Die Möglichkeit der indizierten Variablenadressierung, kombiniert mit dem Objekt Scrollist ermöglicht eine sehr effiziente Erstellung von Parameterlisten z.B. für eine beliebige Anzahl Achsen. D.h. **mit nur einer** Bildseite können Parameter für mehrere Achsen, Temperaturregler, etc. eingegeben werden.



Wird die Index-Variable auf der aktuellen Bildseite geändert, so muss die Bildseite neu aufgebaut werden: SetIndex=x & #Page=aktuellePage

6.2.11 Spalte VarType

Die Spalte VarType definiert den Objektdatentyp und zeigt welche Variablentypen einem Objekt zugeordnet werden können. Bei der Übersetzung des Projektes wird geprüft, ob der Variablentyp (UserVar) mit dem Objektdatentyp übereinstimmt.

Es werden folgende Grunddatentypen der CoDeSys unterstützt:

BOOL, BYTE, DINT, DWORD, INT, REAL, SINT, STRING:[xx], UDINT, UINT, USINT und WORD



Der Datentyp IEC_TIME dient in EPAM zur Darstellung und Eingabe von Zeitwerten und wird in der PLC als Datentyp TIME interpretiert.



Der Datentyp IEC_DT dient in EPAM zur Darstellung und Eingabe von Datum- und Zeitwerten und wird in der PLC als Datentyp DT interpretiert.

Achtung!

Gültiger Wertebereich: 1.1.1970 bis 31.12.2037



Der Datentyp TIME in EPAM ist für die Darstellung von Uhrzeit-/Datumsvariablen reserviert.



Komplexe Datentypen wie Strukturen oder Arrays werden in EPAM als STRING-Variablen mit der entsprechenden Länge STRING:xx behandelt. xx ist hierbei die Länge des Datentyps in Byte.



Neben einfachen Variablen können auch Array- und Strukturelemente gelesen bzw. geschrieben werden.

6.2.12 Spalte VarState

Die Spalte VarState enthält einen Variablennamen für den Objektstatus. Die Variable muss vom Typ WORD, INT, UINT oder BOOL (nur 0 und 1) sein. Mit Hilfe des Objektstatus kann jedes Objekt in EPAM einen der folgenden Zustände annehmen:

Objektstatus = 0	...Objekt ist sichtbar und aktiv, d.h. der Bereich X, Y, DX, DY wird mit dem Objekt dargestellt
Objektstatus = 1	...Objekt ist nicht sichtbar und inaktiv (off), d.h. der Bereich X, Y, DX, DY wird mit der Hintergrundfarbe der aktuellen Bildseite gelöscht. Grenzwerte werden NICHT überwacht!
Objektstatus = 2	...Objekt ist sichtbar aber inaktiv (disable), d.h. der Bereich X, Y, DX, DY wird schraffiert dargestellt
Objektstatus = 4	...Objekt blinkt mit ca. 1Hz, d.h. der Bereich X, Y, DX, DY wird abwechselnd mit der aktuellen Hintergrundfarbe der Bildseite gelöscht und anschliessend wieder das Objekt dargestellt.
Objektstatus = 8	...Objekt blinkt mit ca. 2Hz

Die Änderung des Objektstatus erfolgt über die Objektstatus-Variable (VarState) durch Setzen des entsprechenden Wertes.



Objektstatus bei Bildwechsel

Nach einem Bildwechsel werden alle mit einer Objektstatus-Variable versehenen Objekte mit dem Objektstatus nicht sichtbar und inaktiv (off) initialisiert. Erst nachdem der aktuelle Objektstatus gelesen wurde, wird das Objekt entsprechend dargestellt.

Dieses Vorgehen verhindert ungewollte Aktionen während dem Bildaufbau auf dem Zielsystem! (Beim der Simulation auf der Entwicklungsumgebung werden immer alle Objekte dargestellt!)

6.2.13 Spalte Option

In der Spalte Option werden die objektspezifische Optionen definiert. Mögliche Aktionen sind z.B. DX=, Scroll, Pos=, etc.



Mehrfache Optionen

Mit dem Zeichen ',' können mehrere Optionen projiziert werden.

z.B.: Pos=Center,PWL=1,Scroll ...Positioniert Objekt zentriert, Objekt ist mit Passwortlevel versehen und Objekt kann gescrollt werden.

6.3 Tabellenblätter in Excel

Das erste Tabellenblatt enthält die Definitionen der verschiedenen Bildseiten, die Objekte, Aktionen etc. und die zugehörigen Variablen. Daneben gibt es weitere Tabellenblätter mit zusätzlichen Informationen, z.B. zu Objekten. Diese Tabellenblätter werden bei Bedarf automatisch angelegt.

Es existieren folgende Tabellenblätter:

Tabellenblatt -Typ	Bedeutung	Anzahl
Project	Im Tabellenblatt „Project“ sind alle Bildseiten und deren Objekte definiert. Dieses Tabellenblatt MUSS das erste Tabellenblatt sein!	1
Text	Im Tabellenblatt „Text“ können die projektspezifischen Texte verwaltet werden. Alle Texte die im Tabellenblatt „Text“ definiert wurden, können über das Pulldownfeld der Spalte Text/File angewählt werden. Bei mehrsprachigen Applikationen können in diesem Tabellenblatt die sprachabhängigen Textpaare definiert werden. (s.a. Makro Build Language Text) Beispiel: 1.Default English Francais Hallo Hello Salue	1
UserVar	Im Tabellenblatt „UserVar“ sind alle Variablen definiert. Variablen können von der CoDeSys Programmierumgebung mit dem Makro „PLC Variable Import“ in die „UserVar“-Liste importiert werden. Bei diesem Vorgang werden die bestehenden Variablen gelöscht und die Liste wird neu erstellt. Des Weiteren wird überprüft, ob alle im Project-Tabellenblatt verwendeten Variablen auch im Tabellenblatt „UserVar“ definiert sind und ob der Datentyp übereinstimmt.	1
UserColor	Farbdefinitionen (Farbnummer/Name, R, G, B) zur richtigen Darstellung einer benutzerdefinierten Farbpalette (UserColor) im Wizard	1 optional
Epam	Das Tabellenblatt „Epam“ enthält die Epam-Einstellungen für das entsprechende Zielsystem und sollte nicht verändert werden. (s.a. EPAM.INI)	1
Setup	Das Tabellenblatt „Setup“ enthält verschiedene Einstellungen und sollte nicht verändert werden.	1
DRVParam	Das Tabellenblatt „DRVParam“ enthält die Einstellungen für die Kommunikation zwischen EPAM und CoDeSys und sollte nicht verändert werden.	1
Hosts	Das Tabellenblatt „Hosts“ enthält die Einstellungen für die Kommunikation mit verschiedenen PLC's via Ethernet (TCP/IP). (s.a. Makro: Variable Import)	1
Alarm	Das Tabellenblatt „Alarm“ enthält die Alarmdefinitionen des Objektes Alarm.	1 pro Alarm-Objekt
Alarmmail	Das Tabellenblatt „Alarmmail“ enthält die Email-Definitionen zum Alarmobjekt.	1
Datalog	Das Tabellenblatt „Datalog“ enthält die Variablen-Definitionen zum Datalogobjekt. Pro Datalogobjekt wird ein Tabellenblatt „Datalog“ angelegt und über den Blattnamen referenziert.	1 pro Datalog-Objekt
Message	Das Tabellenblatt „Message“ enthält die Definitionen des Objektes „Message“. Pro Message-Objekt wird ein Tabellenblatt „Message“ angelegt und über den Blattnamen referenziert. Mehrere Message-Objekte können aber auch dasselbe Tabellenblatt „Message“ verwenden.	1 pro Message-Objekt
Recipe	Das Tabellenblatt „Recipe“ enthält die Variablendefinitionen zu einem Rezepturtyp. In einem Projekt können mehrere verschiedene Rezepturtypen definiert werden (z.B. Produkt- und Maschinenparameter). Die zugehörigen Variablendefinitionen werden über den Blattnamen referenziert.	1 pro Recipe-Objekt
Trend	Das Tabellenblatt „Trend“ enthält die Definitionen zum Trendobjekt. Pro Trendobjekt wird ein Tabellenblatt „Trend“ angelegt und über den Blattnamen referenziert. Mehrere Trendobjekte können dasselbe Tabellenblatt „Trend“ verwenden.	1 pro Trend-Objekt
Sys2Plc	Das Tabellenblatt „Sys2Plc“ enthält die Variablendefinitionen zum Sys2Plc-Objekt. Pro Sys2Plc-Objekt wird ein Tabellenblatt „Sys2Plc“ angelegt und über den Blattnamen referenziert.	1 pro Sys2Plc-Objekt
Fontmap	Das Tabellenblatt „Fontmap“ enthält die Font-Definitionen (Name, Typ, Style...)	1
_FileLst	System-Tabellenblatt mit Liste aller benötigten Dateien	1



Der **Typ eines Tabellenblattes** ist als Kommentar in der ersten Zelle (A1) hinterlegt und **darf nicht geändert werden**.

6.4 Passwortschutz eines EPAM-Projekts

Das EPAM-Projekt (*.xls) lässt sich über Excel sichern, indem man unter „Speichern unter-Extras-Allgemeine Optionen“ ein Lese/Schreibkennwort vergibt.

6.5 EPAM-Makros

Nach der Installation von EPAM erscheinen beim Start von Excel die EPAM-Symbolleisten:

- Easy PageMachine
- EPAM Objects
- EPAM Wizard
- EPAM-DB

Diese EPAM-Symbolleisten enthalten verschiedene Zusatzfunktionen in Form von Excel-Makros.

6.5.1 EPAM-Symbolleiste „Easy PageMachine“



EPAM-Makro „Start EPAM“

- Speichert das aktuelle Projekt (gesamte Excel-Tabelle). Das erste Tabellenblatt „Project“ und alle sprachabhängigen Spalten werden im Format „Text mit TABs getrennt“ gespeichert. Anschliessend wird die Simulation unter Windows (WinEpam.EXE) mit dem aktuellen Projekt gestartet.



EPAM-Makro „Open File“

- Erlaubt die Eingabe von Dateinamen (*.PCX, *.ICO, *.TXT) mit Hilfe eines komfortablen Dateiauswahl-Dialogs. Die selektierte Datei wird dabei ins aktuelle Projektverzeichnis kopiert und der Dateiname in die ausgewählte Excel-Zelle übernommen.



EPAM-Makro „Save Worksheet as *.TXT“

- Speichert das aktuelle Tabellenblatt im Format „Text mit TABs getrennt“. Der Dateiname wird aus dem Tabellenblatt-Namen mit der Dateierweiterung „.TXT“ gebildet.



EPAM-Makro „Save as Unicode Textfile“

- Speichert das aktuelle Tabellenblatt im Format „Unicodetext *.txt“. Der Dateiname wird aus dem Tabellenblatt-Namen mit der Dateierweiterung „.TXT“ gebildet.



Voraussetzung: Excel 2000 mit internationalem Sprachsupport.



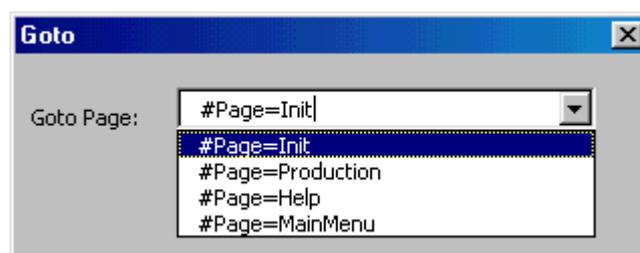
EPAM-Makro „Search“

- EXCEL-Suchfunktion.



EPAM-Makro „Goto“

- Ermöglicht den Sprung im Excel-Tabellenblatt „Project“ auf eine gewünschte projektierte Bildseite `#Page=name`.





EPAM-Makro „Project Settings“

- Ermöglicht die Eingabe von Projekt-Information wie Projektnamen, Programmierer und Version
- Auswählen des Zielsystems und der Darstellung (Portrait=Hochformat)
- Setzt anhand des gewählten Zielsystems die Anzahl verwendbarer Farben und die maximal verfügbare Grösse des RAM-Drives
- Eingabe der IP-Adresse
- Allenfalls Eingabe der RAM-Drive-Grösse in einem dem Zielsystem entsprechenden Bereich
- Ausgabe von Projekt-Informationen wie Anzahl Bildseiten, Anzahl PLC-Variablen, Projekt-Grösse und verwendete Datalog-Grösse und überprüft die zur Verfügung stehenden RAM-Drive-Grösse
 - Datalog-Grösse und Projekt-Grösse \leq RAM-Drive-Grösse
 ⇒ Datalogging und Projekt können ab RAM-Drive betrieben werden (Siehe auch Makro „Download Project“)
 - Datalog-Grösse $<$ RAM-Drive-Grösse
 aber Datalog-Grösse und Projekt-Grösse $>$ RAM-Drive-Grösse
 ⇒ nur noch Datalogging kann ab RAM-Drive betrieben werden (Siehe auch Makro „Download Project“)
 - Datalog-Grösse $>$ RAM-Drive-Grösse
 ⇒ weder Datalogging noch Projekt kann ab RAM-Drive betrieben werden (Siehe auch Makro „Download Project“)
- Auswählen des PLC-Types, Default: Codesys

The screenshot shows the 'Project Info' dialog box with the following fields and options:

- Project: EP-300-10-Demo
- Programmer: Fis
- Version: V3.40
- Target: EP300-10 (selected), Portrait (checkbox)
- Colors: 65536, ScreenX: 640, ScreenY: 480
- PLC-Type: Comm.Settings (selected)
- Simulation: Fullscreen (checkbox), Communication to PLC (checkbox)
- IP-Address: 192.168.0.99
- Project Size: 28790272 Bytes
- Datalog Size: 102400 Bytes
- RAM-Drive Size: 16384 kBytes, used: 0 %
- Total Pages: 1659, Total Variables: 2263
- Total Objects: 7308

The 'Communication Settings' dialog box is also shown, with the following table of options:

PLC	Protocol	Interface
<input checked="" type="checkbox"/> CoDeSys	SymARTI	Ethernet, TCP/IP
<input type="checkbox"/> Siemens	MPI	Profibus-DP/MPI
<input type="checkbox"/> AT-S7	RS7	Ethernet, TCP/IP
<input type="checkbox"/> MultiProg	PDD	Ethernet, TCP/IP
<input type="checkbox"/> UDP	ASCII	Ethernet, UDP

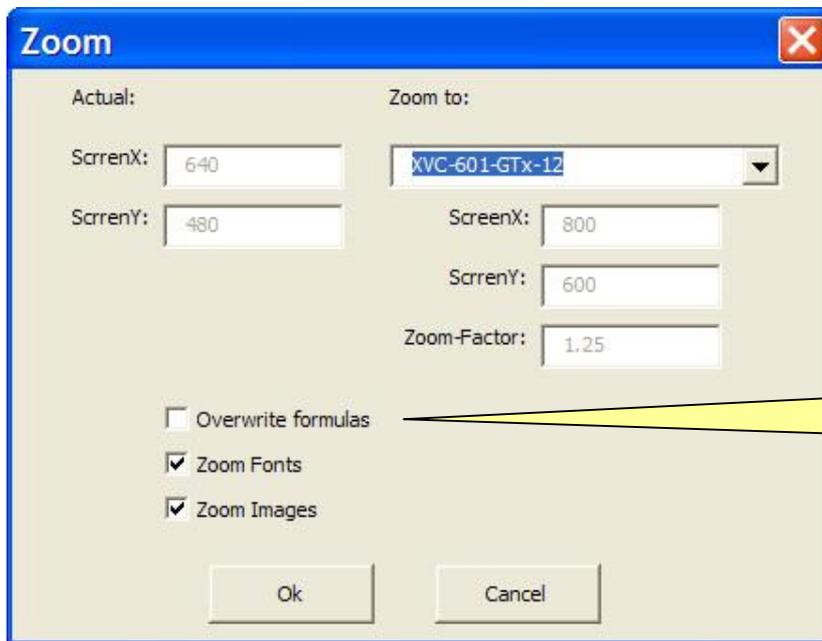
Callouts provide additional information:

- Option: Fullscreen, d.h. Simulation erfolgt unter Windows im Full-screenmodus.
- Option: Comm. to PLC, d.h. Simulation erfolgt mit PLC-Kommunikation.



EPAM-Makro „Zoom Project“

- Konvertierung des Projekts für verschiedene Bildschirmauflösungen inkl. Zeichensätze (Fonts) und Bilder (optional).



Zellbezüge (Formeln) werden beibehalten (Default) oder überschrieben



EPAM-Makro „Compare project“

- Zwei EPAM Projekte vergleichen



EPAM-Makro „EPAM Version“

- Versions-Info über den aktuell verwendeten EPAM-Makro. Eingabe des Produktcodes.



EPAM-Makro „Open Drawing Program“

- Startet das Zeichnungsprogramm mit dem aktuell selektierten PCX-Bild. Beim ersten Start muss der Pfad des Zeichnungsprogramms mit Hilfe eines „Datei öffnen“ Dialogs eingegeben werden. Der aktuelle Pfad wird dann im Tabellenblatt „Setup“ gespeichert.



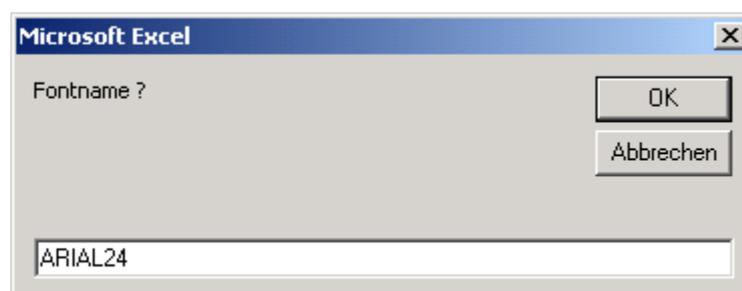
EPAM-Makro „PCX-Colortranslation“

- Konvertiert alle PCX-Bilder und Icons (*.ICO) im Projektverzeichnis. Bilder mit 16 Farben werden auf 256 Farben konvertiert, wobei die restlichen 240 Farben der Farbpalette auf Schwarz gesetzt werden. Bei Bildern mit 256 Farben werden die ersten 16 Farben auf die Farben der EPAM-Standard-Farbpalette geändert.



EPAM-Makro „New-Font“

- Erstellt die Definition eines neuen Fonts (*.FNT) im Tabellenblatt „Fontmap“.





EPAM-Makro „Start FontBuilder“

- Starten den FontBuilder. (nur VxWorks) Beim ersten Start muss der Pfad des Programmes mit Hilfe eines „Datei öffnen“ Dialogs eingegeben werden. Der aktuelle Pfad wird dann im Tabellenblatt „Setup“ gespeichert.



EPAM-Makro „Build-Fonts“

- Erstellt die Fontdateien anhand der Fontdefinitionen im Tabellenblatt „Fontmap“. (nur VxWorks)



EPAM-Makro „Define Language“

- Definiert eine neue Sprache in der EPAM-Applikation. Hierbei werden in allen sprachabhängigen Tabellenblättern zwei weitere Sprachspalten für Text/File und Font am Ende angefügt. Sprachabhängige Tabellenblätter sind Message, Alarm und Project. Im aktuellen Projektverzeichnis wird zusätzlich ein Unterverzeichnis mit dem Sprachnamen definiert. In diesem Unterverzeichnis werden alle sprachabhängigen Dateien (*.TXT, *.PCX, *.ICO, *.FNT) der entsprechenden Sprache abgelegt.



EPAM-Makro „Delete Language“

- Löscht eine mit „Define Language“ definierte Sprache inkl. sprachabhängige Unterverzeichnis (Abfrage) aus der EPAM-Applikation.



EPAM-Makro „Build Language Texts“

- Automatische „Übersetzungsfunktion“ für mehrsprachige Applikationen. Die sprachabhängigen Texte werden gemäss der Textdefinitionen im Tabellenblatt „Text“ automatisch in allen sprachabhängigen Tabellenblätter („Project“, „Alarm“, „Message“) in die Spalte Text/File der dazugehörigen Sprache eingefügt.

**Option „Insert undefined text“**

Wird die Option „Insert undefined text“ aktiviert, werden nicht definierte Texte aus den Tabellenblatt „Project“, „Alarm“ und „Message“ (Spalte Text/File) ins Tabellenblatt „Text“ eingefügt.



EPAM-Makro „Add Text“

- Fügt den Text der aktuellen Zelle ins Tabellenblatt „Text“ ein. Der Text kann anschliessend in der Spalte Text/File mit Hilfe des Pulldownfeldes ausgewählt werden.



EPAM-Makro „Add UserVar“

- Fügt den Text der aktuellen Zelle ins Tabellenblatt „UserVar“ als Variable ein. Die Variable kann anschliessend in der Spalte VarValue, Limit1, Limit2 bzw. VarState mit Hilfe des Pulldownfeldes ausgewählt werden.



EPAM-Makro „Build Recipes“

- Erstellt benutzerdefinierte Rezeptdateien entsprechend den Definitionen im Tabellenblatt „Recipe“ ab Spalte D.



EPAM-Makro „Update Objects“

- Aktualisiert die Objekteigenschaften von bestehenden EPAM-Projekten mit den aktuellen Pulldownfeldern und Optionen.



EPAM-Makro „PLC Variable Import“

- Importiert Variablendefinitionen aus dem CoDeSys-Projekt (Symboldatei *.SYM) ins aktuelle EPAM-Projekt. Hierbei werden alle dem logischen PLC-Namen zugeordneten Variablen im Tabellenblatt „UserVar“ gelöscht und neu importiert. Anschliessend werden alle Variablen im EPAM-Projekt überprüft. Nicht definierte Variablen und Typenkonflikte werden angezeigt.



Werden unter PLC zusätzliche logische PLC-Namen definiert, können die dazugehörigen IP-Adressen eingegeben werden. Anschliessend wird die entsprechende Symolddatei importiert (Diese Einträge werden automatisch im Tabellenblatt „Hosts“ eingetragen).

**Option „SymARTI“**

Default Standard für alle Grossenbacher Geräte. Zusatzparameter für Kommunikation mit Fremdgeräten via SymARTI-Protokoll z.B. ELAU, WAGO, Parker

**Import von S7 Variablen**

S7 Variablen können als AWL importiert werden. Die benötigten DBs werden im Simatic Manager – DB Editor mit Datei – Quelle generieren (Ref. Bausteine einbeziehen, sortieren, Operanden absolut) als AWL exportiert. Diese Datei kann dann in EPAM importiert werden. Als Variablen-Name kann DB Nummer, DB Name oder DB Kommentar verwendet werden.



EPAM-Makro „Build VarList“

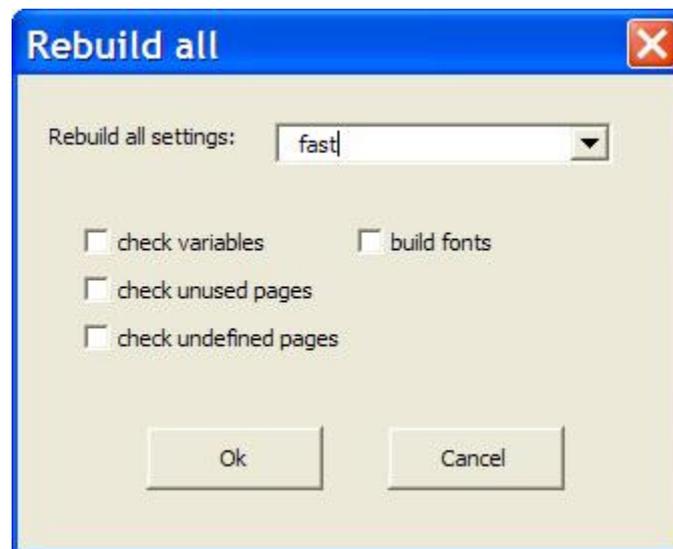
- Erstellt eine Liste aller im Projekt verwendeten Variablen (Datei: „_DRVVLST.TXT“). Diese Liste wird vom Kommunikationstreiber benötigt. Beim Start von EPAM auf dem Zielsystem wird ein Variablenabbild mit allen projektierten Variablen erzeugt und alle Variablen werden gelesen.



EPAM-Makro „Rebuild Project“

Im Dialog „Rebuild all“ können folgende Optionen gewählt werden:

- Fast: speichert nur alle Tabellenblätter
- Complete: führt einen kompletten Rebuild durch inkl. Erstellung der Fonts
- Check variables: prüft Variablendefinitionen und Datentypen (nur notwendig bei Variablenänderungen)
- Build Fonts: erstellt alle Zeichensätze (nur notwendig bei Zeichensatzänderungen)
- Check unused pages: Projekt auf nicht verwendete Pages prüfen
- Check undefined pages: Projekt auf nicht definierte Pages prüfen



Der EPAM-Makro „Rebuild Project“:

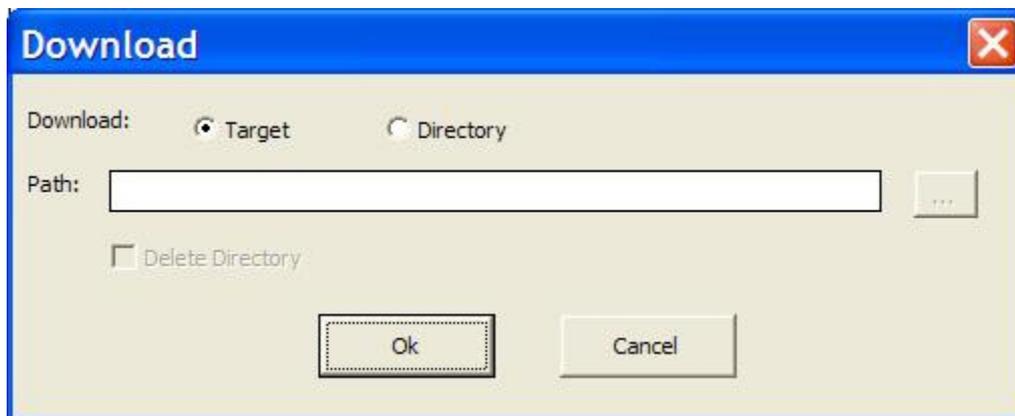
- Übersetzt das gesamte Projekt und speichert alle Tabellenblätter im Format „Text mit TABs“ getrennt
- Speichert alle Sprachen
- Überprüft ob alle verwendeten Dateien vorhanden sind (Bilder, Fonts, Textdateien)
- Erstellt die Variablenliste (Makro: Build VarList)
- Überprüft die verwendete Datalog-Grösse mit der zur Verfügung stehenden RAM-Drive-Grösse
 - Datalog-Grösse > RAM-Drive-Grösse



- Startet den Projekt-Download (EPAM-Makro: „Download Project“) nach fehlerfreiem Rebuild



EPAM-Makro „Download Project“

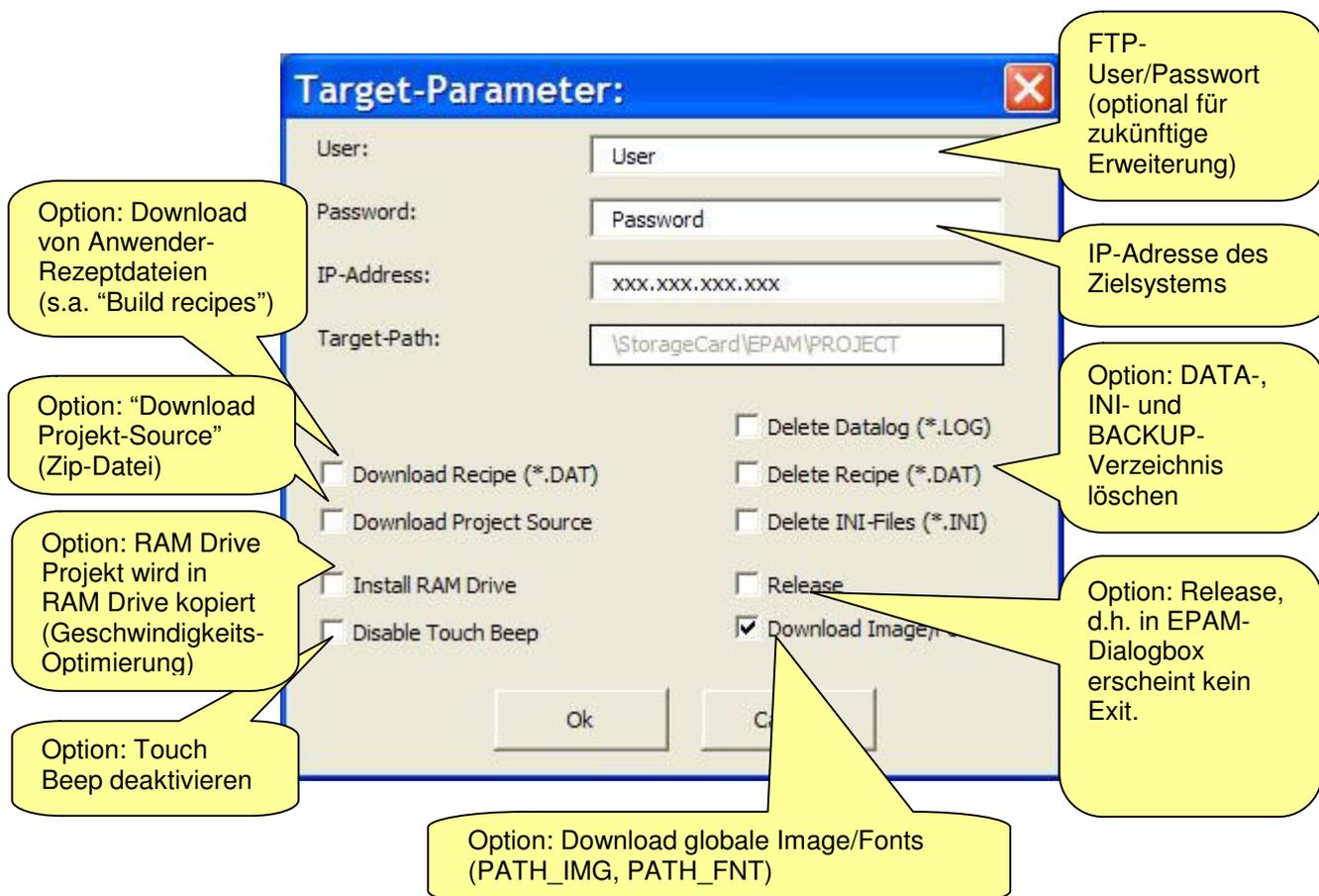


Das Projekt kann auf das Zielsystem (Target) oder in ein Verzeichnis (Directory) kopiert werden. In der Einstellung „Directory“ kann mit dem Button „...“ der Pfad gewählt werden (Default: Unterverzeichnis Target im aktuellen Projektverzeichnis).

Bei den Zielsystemen PC bzw. PocketPC ist nur ein Download in ein Verzeichnis möglich!

Download auf das Zielsystem (Target)

- Erstellt eine Liste aller im Projekt verwendeten Dateien (*.PCX, *.ICO, *.TXT, *.FNT) im System-Tabellenblatt „_FileLst“ und transferiert diese via Windows-FTP auf das Zielsystem. Das Zielsystem muss hierfür via Ethernet mit dem Entwicklungsrechner verbunden sein. In der nachfolgenden Dialogbox können die Download-Parameter eingegeben werden:





Projekt-Verzeichnis

Der Projekt-Verzeichnisname „PROJECT“ ist auf dem Zielsystem (Target-Path) vordefiniert. Das zuletzt geladene Projekt wird gestartet, bestehende Projekte werden gelöscht.



Projekt-Download

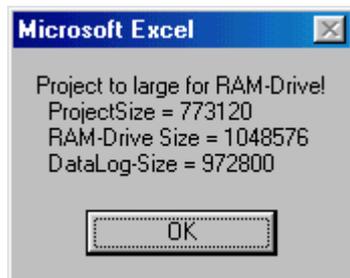
Das Projekt wird ins Verzeichnis ..\EPAM_DNLD_ geladen, EPAM automatisch verlassen, das Projektverzeichnis gelöscht und das Download-Verzeichnis in „PROJECT“ umbenannt. Anschliessend erfolgt ein Projekt-Restart und EPAM wird mit der neuen Applikation gestartet. D.h. **während dem Download wird der doppelte Speicherplatz auf dem Flash benötigt!**



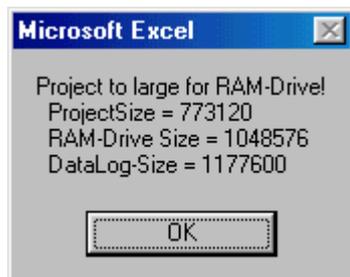
Option „Ram-Drive“

Zur Geschwindigkeitsoptimierung kann das Projekt ab RAM-Drive betrieben werden (empfohlen). Die Option kann je nach Datalog- und Projekt-Grösse mit „Install RAM Drive“ aktiviert werden:

- Datalog-Grösse und Projekt-Grösse \leq RAM-Drive-Grösse
⇒ Datalogging und Projekt können ab RAM-Drive betrieben werden
- Datalog-Grösse $<$ RAM-Drive-Grösse aber
Datalog-Grösse und Projekt-Grösse $>$ RAM-Drive-Grösse
⇒ nur noch Datalogging kann ab RAM-Drive betrieben werden



- Datalog-Grösse $>$ RAM-Drive-Grösse
⇒ weder Datalogging noch Projekt kann ab RAM-Drive betrieben werden



Download Image/Fonts

Image und Fonts können projektunabhängig in einem eigenen Verzeichnis global abgelegt werden. Diese Verzeichnisse können mit PATH_IMG= bzw. PATH_FNT= im Tabellenblatt EPAM (EPAM.INI) definiert werden. Ist mind. ein Pfad gesetzt, so kann beim Download die Option Download Image/Fonts angewählt werden. Ist die Option nicht angewählt, so werden die globalen Bild oder/und Fontdateien nicht übertragen!

Wichtig!

Alle globalen Dateien werden nicht in den RAM-Drive kopiert! Die Ausführungszeit ist daher geringer!



Option „Delete INI-Files“

Wird ein neues Projekt auf das Zielsystem geladen, sollte die Option „delete INI-Files“ aktiviert werden. Ansonsten korrespondieren die INI-Dateien unter Umständen nicht mit den INI-Dateien des neuen Projekts. Die INI-Dateien im EPAM-Backup-Verzeichnis werden in diesem Falle auch automatisch gelöscht.

INI-Dateien beinhalten Werte von Systemvariablen und die Alarmhistory.



EPAM-Makro „Upload Project“

- Upload des gesamten Projekts (ZIP-Datei mit allen benötigten Projektdateien)



Bei einem Projekt upload erfolgt eine Aufforderung das Projekt abzuspeichern. Dieses speichern einer Datei in ein beliebiges Verzeichnis dient lediglich dazu, das Verzeichnis zu bestimmen, in welches das gewünschte Projekt anschliessend gespeichert werden soll. Ein Projekt-Upload kann nur ausgeführt werden, wenn bereits ein Projekt-Download mit Option „Download Project Source“ erfolgte und die Zip-Datei auf dem Zielsystem existiert.



EPAM-Makro „Gruppierung“

- Standard Excel-Kommando „Gruppierung“ und dient dazu, die Excel-Tabelle zu organisieren. In EPAM-Projekten lassen sich damit z.B. alle Zeilen einer Bildseite zusammenfassen. Anschliessend kann die Bildseite links neben der Tabelle aus- und eingeblendet werden.



EPAM-Makro „Update Objects“ hebt Gruppierungen auf



EPAM-Makro „Gruppierung aufheben“

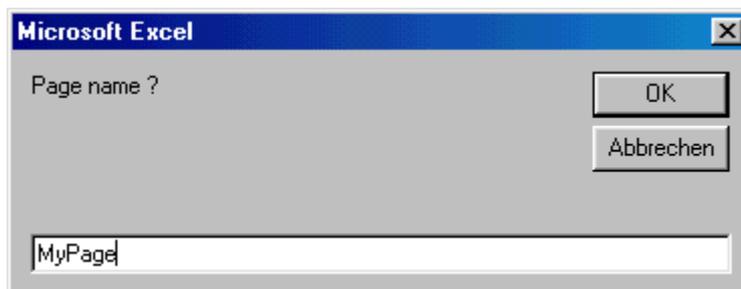
- Standard Excel-Kommando „Gruppierung aufheben“

6.5.2 EPAM-Symbolleiste „EPAM Objects“



EPAM-Makro „NewPage“

- Erstellt ein neues Page-Objekt und fügt es in der aktuellen Zeile ein. Der Name der Bildseite kann in nachfolgender Dialogbox eingegeben werden. Der Name einer Bildseite muss eindeutig sein und darf ausser „_“ keine Sonderzeichen beinhalten.



EPAM-Makro „NewButton“

- Erstellt ein neues Button-Objekt und fügt es in der aktuellen Zeile ein.



EPAM-Makro „NewSwitch“

- Erstellt ein neues Switch-Objekt und fügt es in der aktuellen Zeile ein.



EPAM-Makro „NewDropDownList“

- Erstellt ein neues DropDownList-Objekt und fügt es in der aktuellen Zeile ein.



EPAM-Makro „NewRadioButton“

- Erstellt ein neues RadioButton-Objekt und fügt es in der aktuellen Zeile ein.



EPAM-Makro „NewVariable“

- Erstellt ein neues Variablen-Objekt und fügt es in der aktuellen Zeile ein.



EPAM-Makro „NewBar“

- Erstellt ein neues Bar-Objekt und fügt es in der aktuellen Zeile ein.



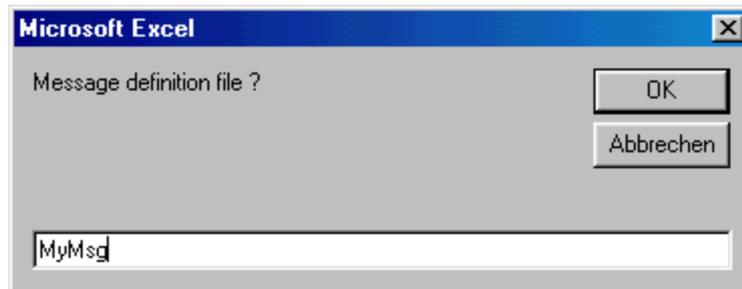
EPAM-Makro „NewSignal“

- Erstellt ein neues Signal-Objekt und fügt es in der aktuellen Zeile ein.



EPAM-Makro „NewMessage“

- Erstellt ein neues Message-Objekt und fügt es in der aktuellen Zeile ein. Der Name der Meldungs-Definitionsdatei kann in nachfolgender Dialogbox eingegeben werden. Der Name der Meldungs-Definitionsdatei muss eindeutig sein und darf ausser „_“ keine Sonderzeichen und max. 8 Charakter beinhalten (ISO 9660, 8.3 mit eingeschränktem Zeichensatz). Anschliessend wird ein Tabellenblatt „Message“ mit dem eingegebenen Namen erstellt. In diesem Tabellenblatt können dann die Meldungen definiert werden.



EPAM-Makro „NewMeter“

- Erstellt ein neues Meter-Objekt und fügt es in der aktuellen Zeile ein.



EPAM-Makro „NewDiagnoseSignal“

- Erstellt ein neues DiagnoseSignal-Objekt und fügt es in der aktuellen Zeile ein.



EPAM-Makro „NewAlarmList“

- Erstellt ein neues Alarm-Listen-Objekt und fügt es in der aktuellen Zeile ein.



EPAM-Makro „NewRecipeList“

- Erstellt ein neues Rezeptur-Listen-Objekt und fügt es in der aktuellen Zeile ein.



EPAM-Makro „NewTextList“

- Erstellt ein neues Text-Listen-Objekt und fügt es in der aktuellen Zeile ein. Mit Hilfe des nachfolgenden „Datei öffnen“ Dialogs kann die ASCII-Textdatei ausgewählt werden, die angezeigt werden soll. Diese Datei wird allenfalls ins aktuelle Projektverzeichnis kopiert.



EPAM-Makro „NewHTMLBrowser“

- Erstellt ein neues HTML-Browser-Objekt und fügt es in der aktuellen Zeile ein. (nur Windows/WindowsCE)



EPAM-Makro „NewRemoteControl“

- Erstellt ein neues RemoteControl-Objekt und fügt es in der aktuellen Zeile ein.



EPAM-Makro „NewScrollList“

- Erstellt ein neues Scroll-Listen-Objekt und fügt es in der aktuellen Zeile ein.



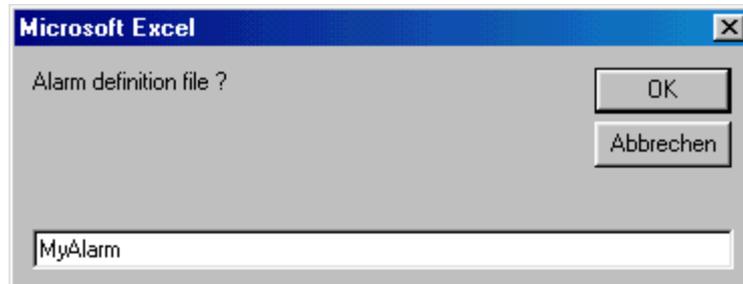
EPAM-Makro „NewTrend“

- Erstellt ein neues Trend-Objekt und fügt es in der aktuellen Zeile ein. Der Name der Trend-Parameterdatei kann in nachfolgender Dialogbox eingegeben werden. Der Name der Trend-Parameterdatei muss eindeutig sein und darf ausser „_“ keine Sonderzeichen und max. 8 Charakter beinhalten (ISO 9660, 8.3 mit eingeschränktem Zeichensatz). Anschliessend wird ein Tabellenblatt „Trend“ mit dem eingegebenen Namen erstellt. In diesem Tabellenblatt können dann Trend-Parameter definiert werden.



EPAM-Makro „NewAlarm“

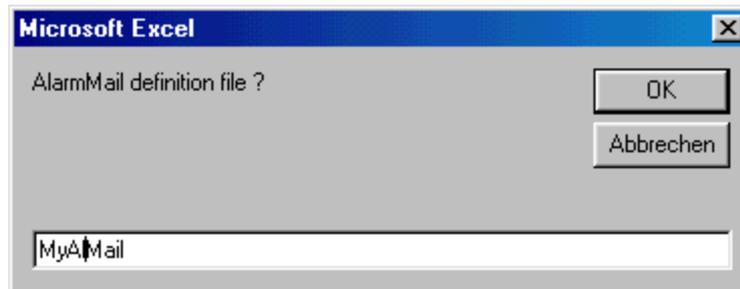
- Erstellt ein neues Alarm-Objekt und fügt es in der aktuellen Zeile ein. Der Name der Alarm-Definitionsdatei kann in nachfolgender Dialogbox eingegeben werden. Der Name der Alarm-Definitionsdatei muss eindeutig sein und darf ausser „_“ keine Sonderzeichen und max. 8 Charakter beinhalten (ISO 9660, 8.3 mit eingeschränktem Zeichensatz). Anschliessend wird ein Tabellenblatt „Alarm“ mit dem eingegebenen Namen erstellt. In diesem Tabellenblatt können dann die Alarmmeldungen definiert werden.





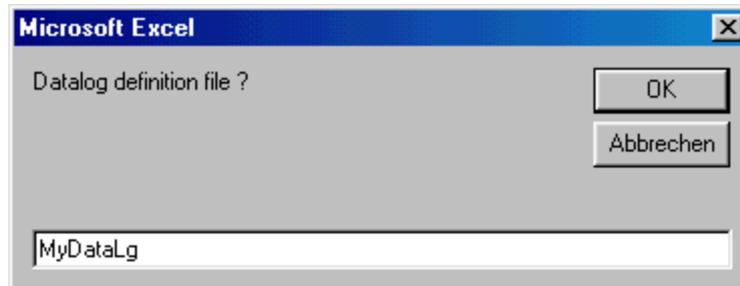
EPAM-Makro „NewAlarmMail“

- Erstellt ein neues AlarmMail-Objekt und fügt es in der aktuellen Zeile ein. Der Name der AlarmMail-Definitionsdatei kann in nachfolgender Dialogbox eingegeben werden. Der Name der AlarmMail-Definitionsdatei muss eindeutig sein und darf ausser „_“ keine Sonderzeichen und max. 8 Charakter beinhalten (ISO 9660, 8.3 mit eingeschränktem Zeichensatz). Anschliessend wird ein Tabellenblatt „AlarmMail“ mit dem eingegebenen Namen erstellt. In diesem Tabellenblatt können dann die AlarmMail-Parameter definiert werden.



EPAM-Makro „NewDataLog“

- Erstellt ein neues DataLog-Objekt und fügt es in der aktuellen Zeile ein. Der Name der DataLog-Definitionsdatei kann in nachfolgender Dialogbox eingegeben werden. Der Name der DataLog-Definitionsdatei muss eindeutig sein und darf ausser „_“ keine Sonderzeichen und max. 8 Charakter beinhalten (ISO 9660, 8.3 mit eingeschränktem Zeichensatz). Anschliessend wird ein Tabellenblatt „DataLog“ mit dem eingegebenen Namen erstellt. In diesem Tabellenblatt können dann die DataLog-Parameter definiert werden.



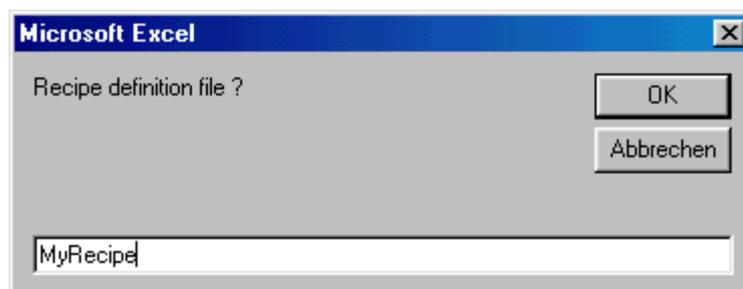
EPAM-Makro „NewPassword“

- Erstellt ein neues Password-Objekt und fügt es in der aktuellen Zeile ein.



EPAM-Makro „NewRecipe“

- Erstellt ein neues Rezept-Objekt und fügt es in der aktuellen Zeile ein. Der Name der Rezept-Definitionsdatei kann in nachfolgender Dialogbox eingegeben werden. Der Name der Rezept-Definitionsdatei muss eindeutig sein und darf ausser „_“ keine Sonderzeichen und max. 8 Charakter beinhalten (ISO 9660, 8.3 mit eingeschränktem Zeichensatz). Anschliessend wird ein Tabellenblatt „Recipe“ mit dem eingegebenen Namen erstellt. In diesem Tabellenblatt können dann die Rezeptvariablen definiert werden.



EPAM-Makro „NewScreenSaver“

- Erstellt ein neues ScreenSaver-Objekt und fügt es in der aktuellen Zeile ein.



EPAM-Makro „NewSys2PLC“

- Erstellt ein neues Sys2Plc-Objekt und fügt es in der aktuellen Zeile ein. Der Name der Sys2Plc-Definitionsdatei kann in nachfolgender Dialogbox eingegeben werden. Der Name der Sys2Plc -Definitionsdatei muss eindeutig sein und darf ausser „_“ keine Sonderzeichen und max. 8 Charakter beinhalten (ISO 9660, 8.3 mit eingeschränktem Zeichensatz). Anschliessend wird ein Tabellenblatt „Sys2Plc“ mit dem eingegebenen Namen erstellt. In diesem Tabellenblatt können dann die Sys2Plc -Variablen definiert werden.



**EPAM-Demos**

Einige Beispiel-Excel-Tabellen finden Sie im EPAM-Verzeichnis (EPAM): ***EPAM\SAMPLES***

**Definitionen der Excel-Tabelle grafisch darstellen**

Auf dem Entwicklungssystem können nach dem Start von EPAM die Definitionen der einzelnen Objekte aus der Excel-Tabelle, spaltenweise mit der Tastenkombination „Alt I“ in Textform im Bild angezeigt werden.

Mit der Taste Printscreen kann unter Windows eine Kopie der EPAM-BildschirmAusgabe gemacht werden.

Mit der Tastenkombination „Alt p“ kann im aktuellen Projektverzeichnis ein Screenshot des aktuellen Bildinhaltes in Form eines PCX-Bildes erstellt werden. Der Dateiname wird dabei aus den ersten 4 Zeichen des Pagenamen und weiteren 4 Zeichen einer fortlaufenden Nummer erstellt (z.B. STAR0000.PCX).

6.5.3 EPAM-Symboleiste „EPAM Wizard“

Mit Hilfe des EPAM-Wizards lassen sich auf einfachste Weise die Positionen X , Y und die Dimensionen DX , DY von Objekten ändern.



EPAM-Wizard-Makro „Refresh“

- Öffnen des EPAM-Wizards. EPAM-Wizard visualisiert im EPAM-Wizard-Fenster die aktive Bildseite. Objekte dieser Bildseite können nun bearbeitet werden. Änderungen werden direkt in der Excel-Tabelle nachgeführt. Möchten Sie eine andere Bildseite visualisieren und bearbeiten, springen Sie in der Excel-Tabelle in die gewünschte Bildseite und führen erneut den EPAM-Wizard-Makro „Refresh“ aus.



Führen Sie bei aktivem EPAM-Wizard in der Excel-Tabelle Änderungen aus, werden diese Änderungen nicht automatisch im EPAM-Wizard-Fenster nachgeführt. Um das EPAM-Wizard-Fenster zu aktualisieren, führen Sie bitte erneut den EPAM-Wizard-Makro „Refresh“ aus.



EPAM-Wizard-Makro „Align Left“

- Richtet selektierte Objekte anhand des zuletzt selektierten Objekts links aus. Das Selektieren von mehreren Objekten erfolgt mit gedrückter Ctrl-Taste.



EPAM-Wizard-Makro „Align Top“

- Richtet selektierte Objekte anhand des zuletzt selektierten Objekts oben aus. Das Selektieren von mehreren Objekten erfolgt mit gedrückter Ctrl-Taste.



EPAM-Wizard-Makro „Align Right“

- Richtet selektierte Objekte anhand des zuletzt selektierten Objekts rechts aus. Das Selektieren von mehreren Objekten erfolgt mit gedrückter Ctrl-Taste.



EPAM-Wizard-Makro „Align Bottom“

- Richtet selektierte Objekte anhand des zuletzt selektierten Objekts unten aus. Das Selektieren von mehreren Objekten erfolgt mit gedrückter Ctrl-Taste.



EPAM-Wizard-Makro „Spacing Horizontal“

- Richtet selektierte Objekte in einem einheitlichen horizontalen Abstand aus. Das Selektieren von mehreren Objekten erfolgt mit gedrückter Ctrl-Taste.



EPAM-Wizard-Makro „Spacing Vertical“

- Richtet selektierte Objekte in einem einheitlichen vertikalen Abstand aus. Das Selektieren von mehreren Objekten erfolgt mit gedrückter Ctrl-Taste.



EPAM-Wizard-Makro „Format Widths“

- Ändert von selektierten Objekten die Objektbreite (DX) anhand des zuletzt selektierten Objekts. Das Selektieren von mehreren Objekten erfolgt mit gedrückter Ctrl-Taste.



Makro „Format Heights“

- Ändert von selektierten Objekten die Objekthöhe (DY) anhand des zuletzt selektierten Objekts. Das Selektieren von mehreren Objekten erfolgt mit gedrückter Ctrl-Taste.



EPAM-Wizard-Makro „Undo“

- vorhergehenden EPAM-Wizard-Aktionen werden rückgängig gemacht.



EPAM-Wizard-Makro „Full Screen“

- EPAM-Wizard-Fenster wird in voller Grösse dargestellt (maximale Höhe des EPAM-Wizard-Fensters entspricht der Höhe des Excel-Tabellenbereichs).



EPAM-Wizard-Makro „Half Screen“

- EPAM-Wizard-Fenster wird in halber Grösse dargestellt.



Makro „Small Screen“

- EPAM-Wizard-Fenster wird in kleiner Grösse dargestellt.



EPAM-Wizard-Makro „AutoSize“

- EPAM-Wizard-Fenster wird in automatischer Grösse dargestellt.



EPAM-Wizard-Makro „100%“

- Objekte im EPAM-Wizard-Fenster werden mit 100% Grösse dargestellt.



EPAM-Wizard-Makro „150%“

- Objekte im EPAM-Wizard-Fenster werden mit 150% Grösse dargestellt.



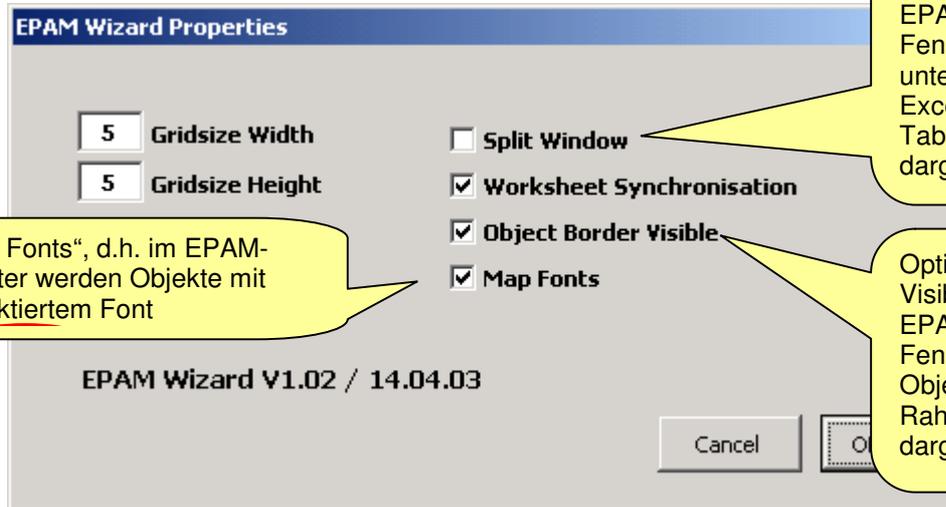
EPAM-Wizard-Makro „200%“

- Objekte im EPAM-Wizard-Fenster werden mit 200% Grösse dargestellt.



EPAM-Wizard-Makro „Properties“

- Versions-Info über den aktuell verwendeten EPAM-Wizard-Makro
- Einstellung von Optionen
- Einstellung des Grids in X- und Y-Richtung



Worksheet Synchronisation

Ist die Option „Worksheet Synchronisation“ aktiv, werden im EPAM-Wizard-Fenster durchgeführte Änderungen direkt in der Excel-Tabelle nachgeführt. (Default)



Kopieren und Löschen von Objekten im Wizard

Mit den Tasten Strg-C/Strg-V können markierte Objekte im Wizard kopiert werden. Die Taste „Löschen“ löscht markierte Objekte.

6.5.4 EPAM-Symboleiste „EPAM DB“

Die EPAM-DB Symboleiste beinhaltet folgende Funktionen (s.a. Dokumentation EPAM-DB-Server):

- DB-Setup ...Setup des EPAM-Datenbank-Servers und Definition des VarLog-Tabellenblattes
- DBPasswd ...Objekt DBPasswd für zentrale User/Passwortverwaltung auf dem EPAM DB-Server
- DBTracer ...Objekt DBTracer für Aufzeichnung von Variablen auf dem EPAM DB-Server
- Import Varlog ...Import von Variablen in die VarLog-Liste. Alle in der VarLog-Liste definierten Variablen, werden bei Änderung auf dem EPAM DB-Server protokolliert.

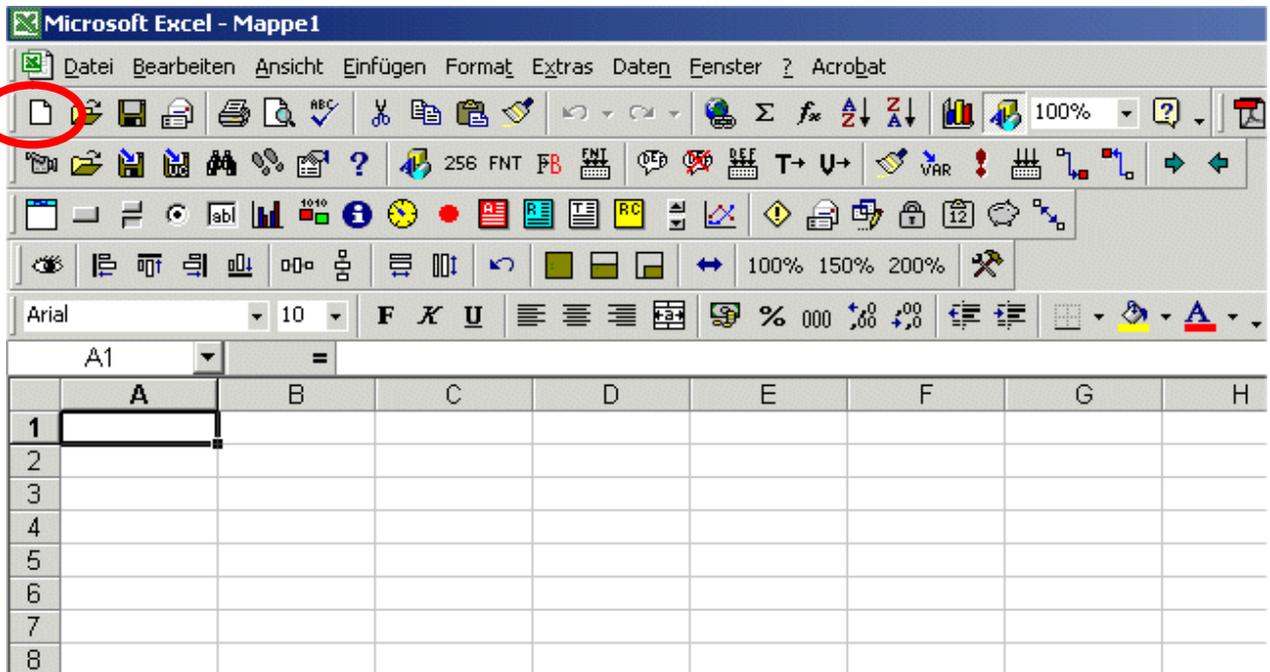


Die EPAM-DB Funktionen sind derzeit nur für VxWorks verfügbar! Zur Nutzung der DB-Funktionen wird die Runtime-Erweiterung „EPAM DB Extension“ und der EPAM DB-Server benötigt. (s.a. Dokumentation EPAM-DB Extension und EPAM-DB Server)

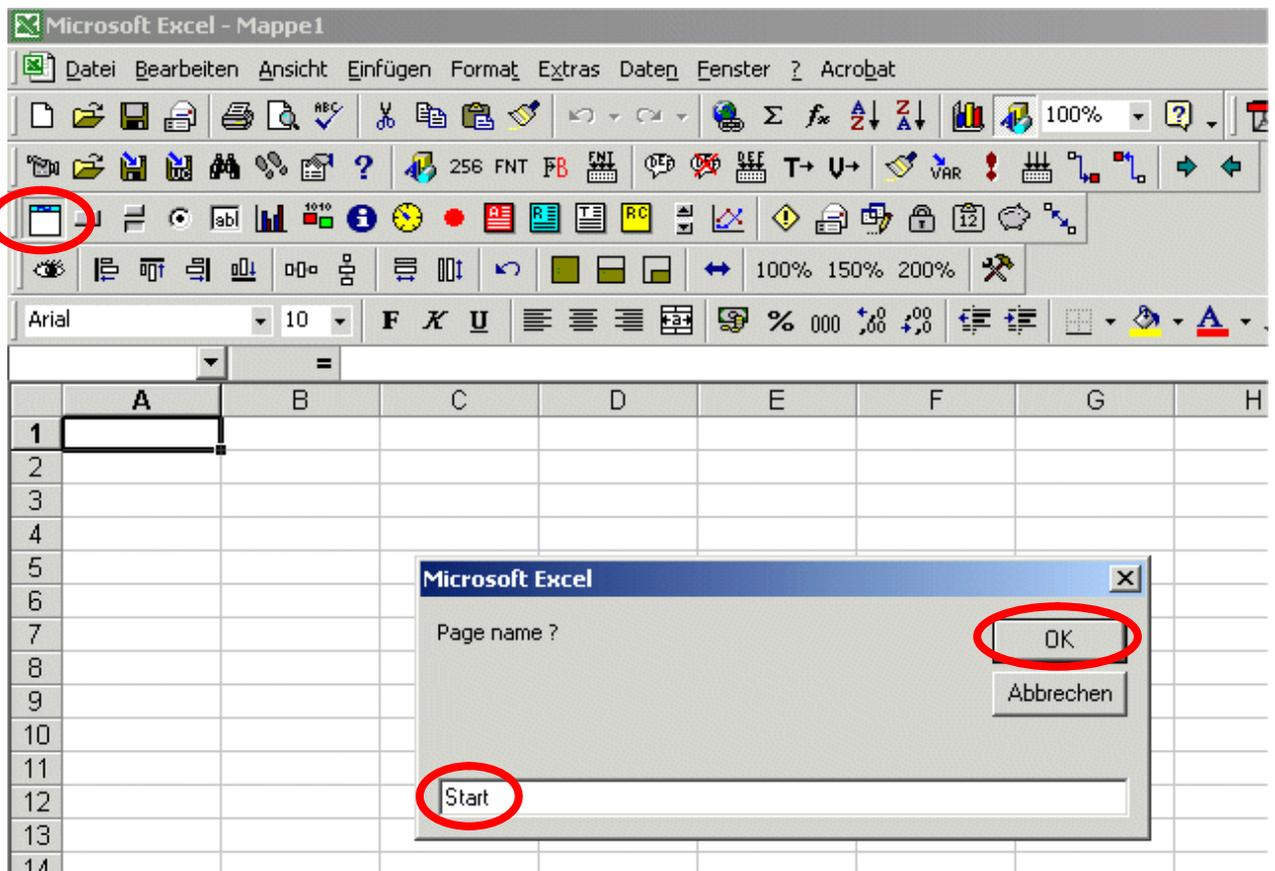
6.6 Ein kleines Projekt von A-Z

Der Ablauf zur Erstellung eines Projekts sieht wie folgt aus:

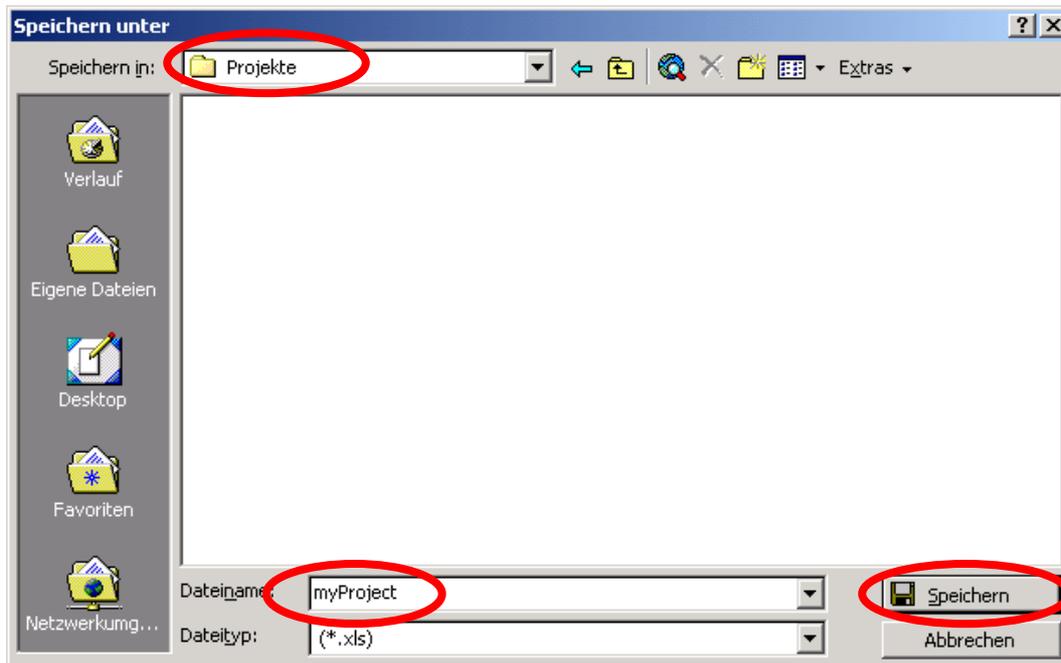
1. **Neues Projekt öffnen:** öffnen Sie eine neue EXCEL-Arbeitsmappe



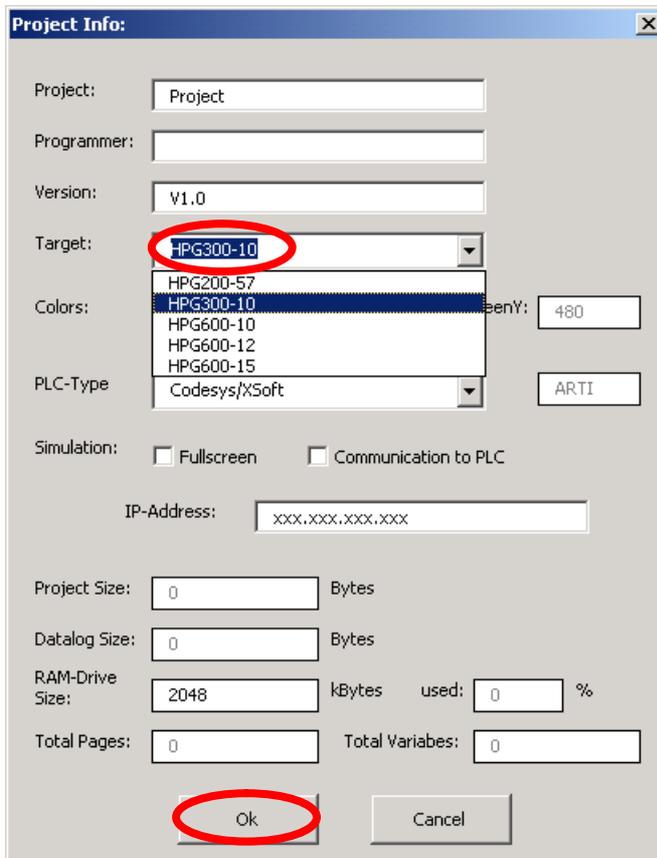
2. **Bildseiten und Objekte definieren:** definieren Sie eine neue Bildseite mit dem Makro „NewPage“ und geben Sie der Seite einen eindeutigen Namen: z.B. „Start“



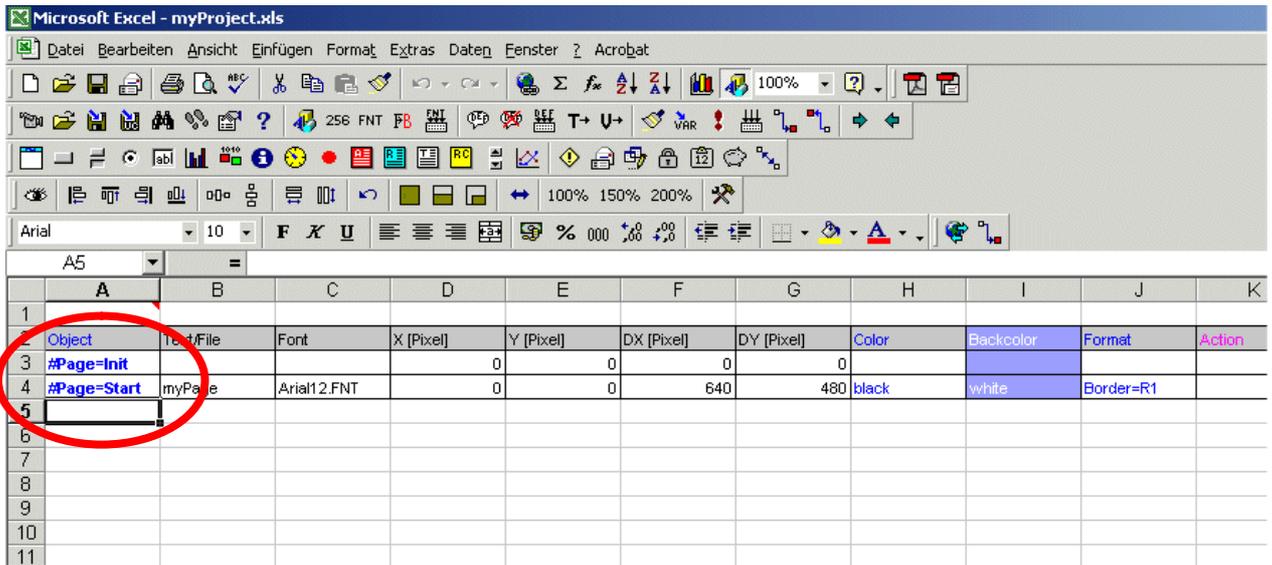
Hierbei werden Sie aufgefordert die EXCEL-Mappe zu speichern (Projektverzeichnis wählen und Name der EXCEL-Datei eingeben) ...



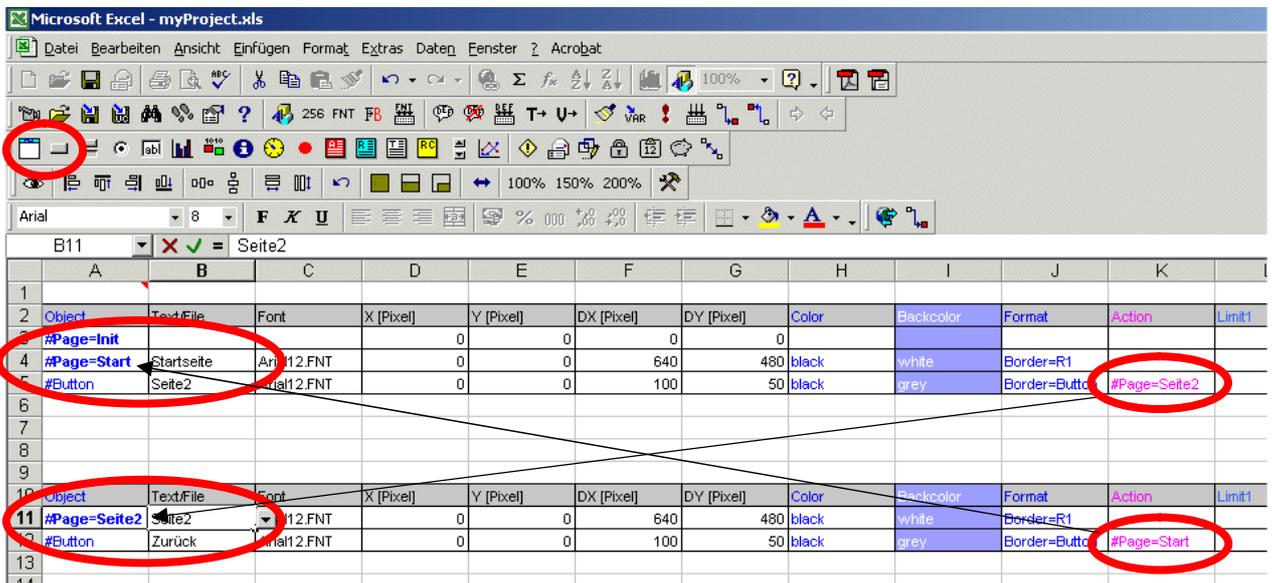
und es erscheint der Dialog „Projekt Info“, wo Sie neben Projektinformationen das Zielsystem (Target) definieren können, für welches das Projekt erstellt wird: z.B. HPG300-10



Sie haben in Ihrer EXCEL-Tabelle nun zwei Objekte **#Page=Init** und **#Page=Start**. Der Cursor steht in der nächsten leeren Zeile.



Dort können Sie nun mit den Makros „NewButton“, „NewVariable“, etc. weitere Objekte für die Bildseite „Start“ definieren. Um eine weitere Bildseite zu projektieren definieren Sie nach dem letzten Objekt, evtl. nach einer Leerzeile (Übersichtlichkeit), eine neue Seite z.B. „Seite2“ mit dem Makro „NewPage“. Mit den Button-Aktionen **#Page=name** können Sie einen Seitenwechsel von der Seite Start auf die Seite2 und wieder zurück definieren.



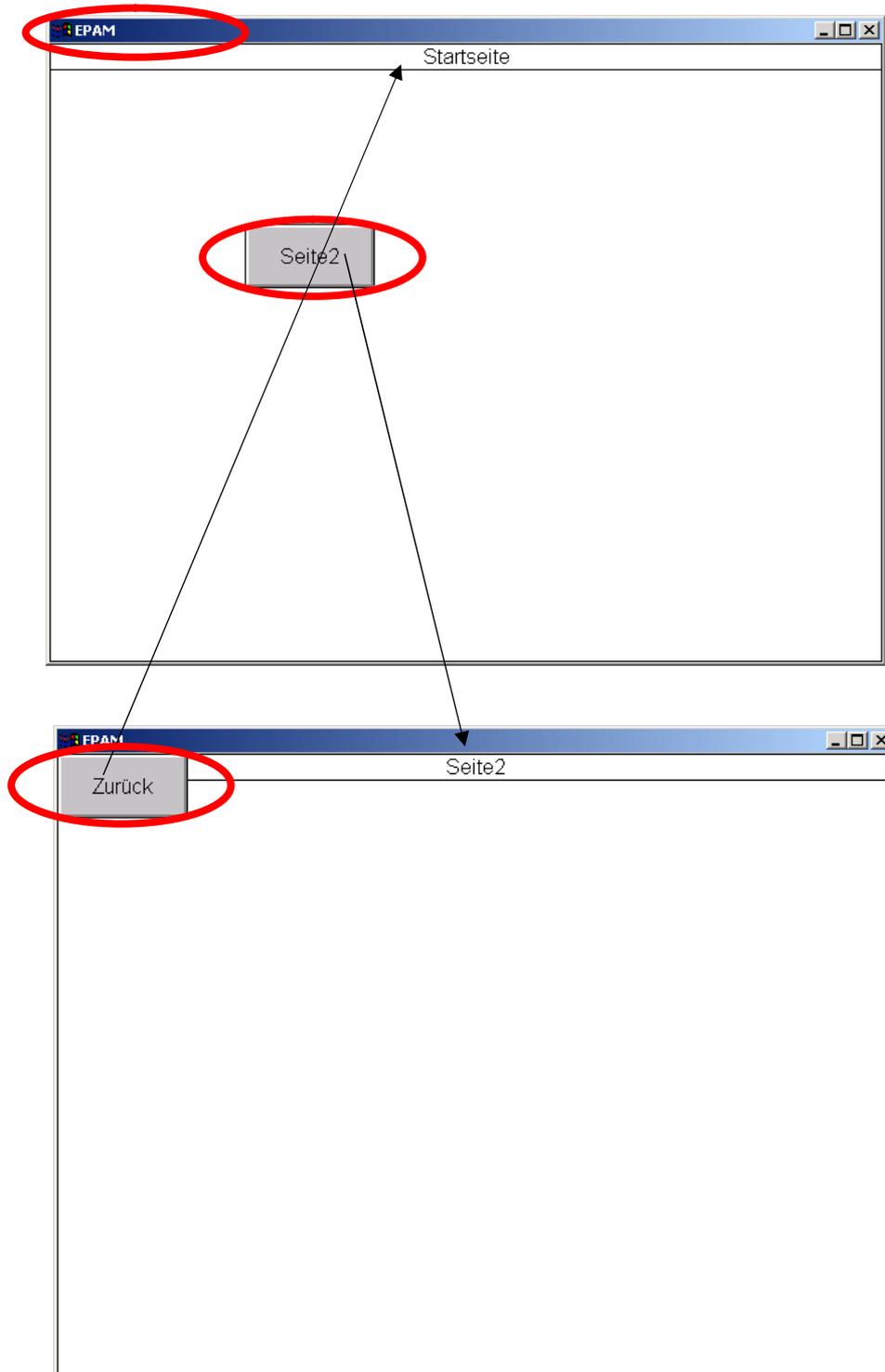
3. **Definition der Objekteigenschaften:** In den Spalten können die Objekteigenschaften mit Hilfe der **Pulldownfelder** verändert werden. Die grafische Positionierung kann mit dem Makro „Refresh“ des **EPAM-Wizards** erfolgen. Hierbei muss vorher die gewünschte Seite durch anklicken einer Zelle innerhalb der Seite angewählt werden. Die angewählte Page wird dann im Wizard dargestellt und die Objekte können mit der Maus verschoben bzw. die Dimension verändert werden. Die Änderungen werden direkt in die EXCEL-Tabelle übernommen (X,Y,DX,DY).

The screenshot shows the Microsoft Excel interface with the EPAM Wizard. The spreadsheet has the following data:

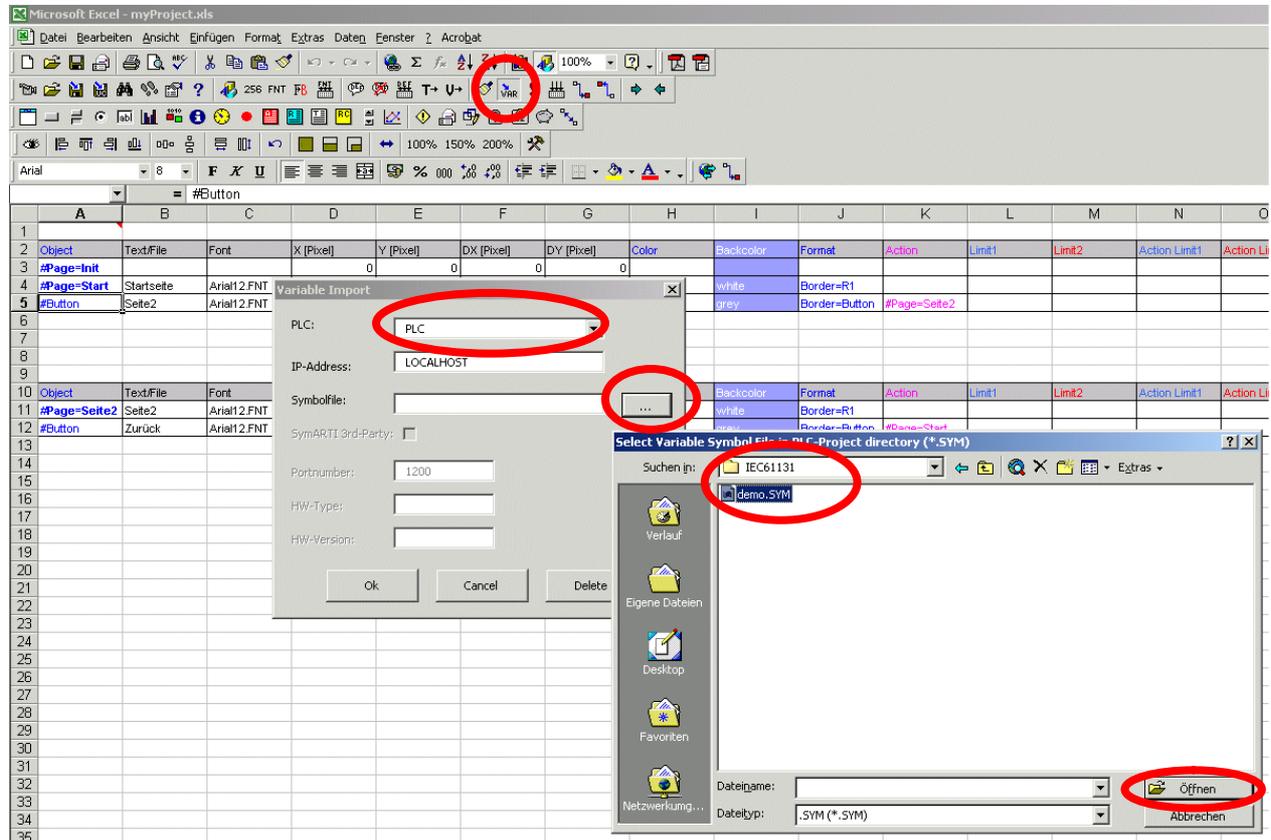
	A	B	C	D	E	F	G	H	I	J	K	L
1												
2	Object	Text/File	Font	X [Pixel]	Y [Pixel]	DX [Pixel]	DY [Pixel]	Color	Backcolor	Format	Action	Limit1
3	#Page=Init	Startseite										
4	#Page=Start	Startseite	Arial12.FNT									
5	#Button	Seite2	Arial12.FNT									
6												
7												
8												
9												
10	Object	Text/File	Font	X								
11	#Page=Seite2	Seite2	Arial12.FNT									
12	#Button	Zurück	Arial12.FNT									
13												
14												
15												
16												
17												
18												
19												
20												
21												
22												
23												

The dialog box 'myProject: Start "Startseite"' is open, showing a preview of the 'Startseite' page with a button labeled 'Seite2'.

4. **Simulation der Applikation auf dem Entwicklungsrechner:** Mit Hilfe des Makros „Start EPAM“ kann jederzeit die Simulation unter Windows aufgerufen werden und die Applikation getestet werden.



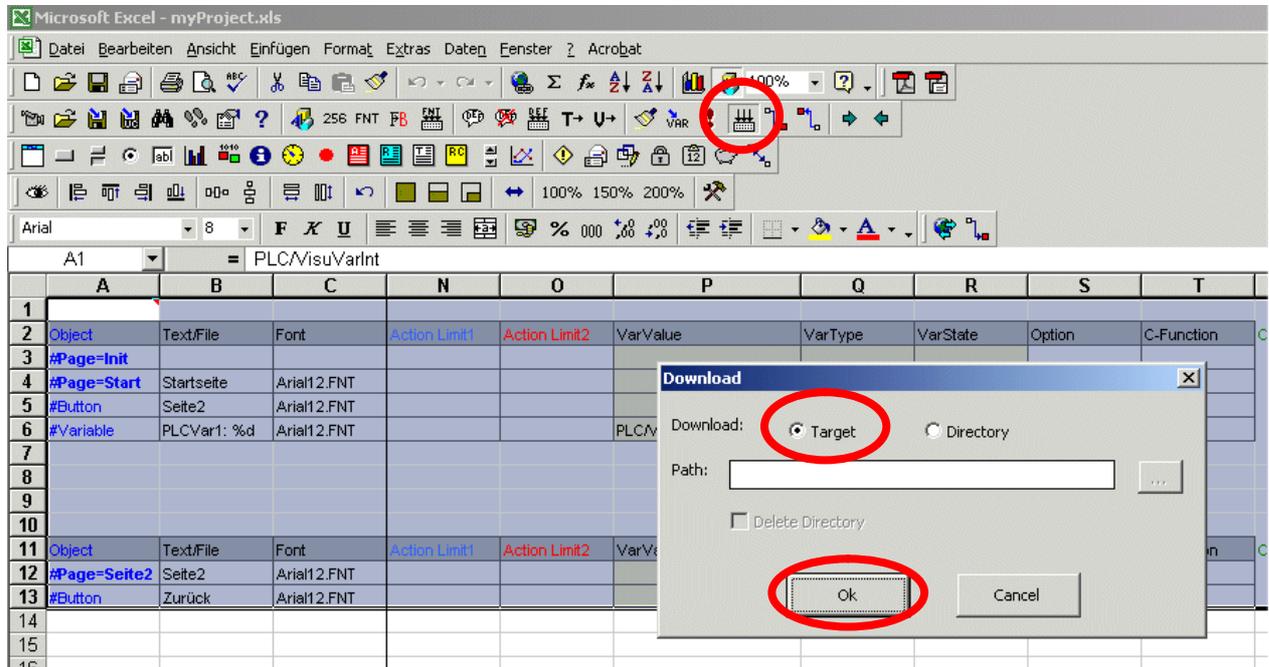
5. **Import von PLC-Variablendefinitionen:** Mit Hilfe des Makros „PLC Variable Import“ können Sie die Variablendefinitionen aus der CoDeSys direkt einlesen und müssen die Variablen nicht nochmals eingeben (siehe auch Kap. 2.4 Einstellungen in der CoDeSys-Entwicklungsumgebung, S.11). Sie werden hierbei aufgefordert die PLC auszuwählen (Default: ist die lokale PLC definiert, es können aber weitere PLCs definiert werden, die via Ethernet vernetzt sind) und können nun die Symboldatei auswählen, welche die Variablendefinition enthält (*.SYM-Datei).



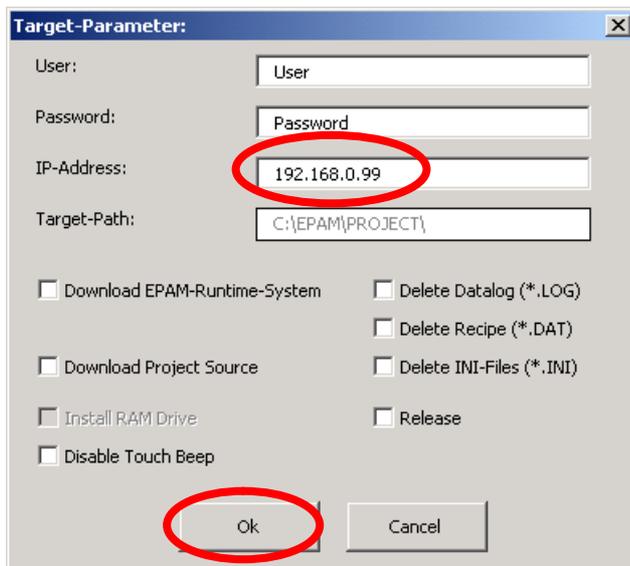
Mit „Ok“ werden die Variablen ins Tabellenblatt „UserVar“ importiert und können nun im Pulldownmenü z.B. in der Spalte „VarValue“ ausgewählt und einem Objekt zugeordnet werden. Anschliessend muss noch der Datentyp des Objekts definiert werden (Spalte VarType). Der Objekttyp und der Variablentyp müssen übereinstimmen. Ein Typkonflikt wird ggf. bei einem „Rebuild All“ angezeigt.

	A	B	C	N	O	P	Q	R	S
1									
2	Object	Text/File	Font	Action Limit1	Action Limit2	VarValue	VarType	VarState	Option
3	#Page=Init								
4	#Page=Start	Startseite	Arial12.FNT						
5	#Button	Seite2	Arial12.FNT						
6	#Variable	PLCVar1: %d	Arial12.FNT			PLC/VisuVarInt		WORD	
7						PLC/VisuVarInt			
8						PLC/VisuVarIntMin			
9						PLC/VisuVarReal			
10						PLC/VisuVarRealMax			
11	Object	Text/File	Font	Action Limit1	Action Limit2	PLC/VisuVarRealMin	Type	VarState	Option
12	#Page=Seite2	Seite2	Arial12.FNT			PLC/VisuVarSint			
13	#Button	Zurück	Arial12.FNT			PLC/VisuVarSintMax		WORD	
14									
15									

6. **Projekt übersetzen und aufs Zielsystem laden:** mit dem Makro „Rebuild all“ wird nun das Projekt übersetzt und geprüft. Anschliessend erscheint eine Auswahl „Download“. Sie können das Projekt nun in ein lokales Verzeichnis oder auf das Zielsystem (Target, Defaulteinstellung) laden.



Mit der Einstellung Target (Default) erscheint anschliessend der Dialog „Target Parameter“. Hier müssen Sie die IP-Adresse des Zielsystems eintragen. Mit „Ok“ wird der Download gestartet. Nach erfolgreichem Download startet die EPAM-Applikation automatisch mit dem neuen Projekt. Zuvor sollten Sie das PLC-Projekt auf das Zielsystem laden, damit die Kommunikation mit der PLC funktioniert und die definierten Variablen in EPAM von der PLC gelesen werden können.



Sollten Sie keine Verbindung zum Zielsystem herstellen können, prüfen Sie bitte Ihre Netzwerkeinstellungen. Der PC des Entwicklungssystems muss eine IP-Adresse haben und diese muss im selben Subnetz wie das Zielsystem sein (erste 3 Nummern des IP-Adresse gleich, letzte Nummer unterschiedlich!).

Prüfen Sie die Netzwerkverbindung mit folgendem Befehl in der Kommandozeile: Ping xxx.xxx.xxx.xxx (xxx = IP-Adresse des Zielsystems).

7 Objektdefinition

Die Objektattribute in den nachfolgenden Tabellen sind zur besseren Übersicht in der ersten Spalte beschrieben. In der Excel-Tabelle wird jedes Objekt in einer Zeile beschrieben, d.h. jedes Attribute wird in einer Spalte dargestellt.



Die Länge einer Zeile ist derzeit auf 512 Zeichen pro Sprache beschränkt! Bei Unicode-Sprachen sind max 85 Zeichen (=512/6) möglich!

Die verwendete Schreibweise hat folgende Bedeutung:

grau hinterlegter Text	...keine oder fest vordefinierte Attribute. Blau hinterlegt sind neue Features der Version 3.30
fett geschriebener Text	...reservierte Worte
<i>kursiv geschriebener Text</i>	...sind anwenderspezifische Eingaben

7.1 Objekt Page

#Page=Name		frei wählbarer, eindeutiger Name der Bildseite
Text/File	Bild.PCX	<ul style="list-style-type: none"> Name einer PCX-Bilddatei für das Hintergrundbild
	<i>Text</i>	<ul style="list-style-type: none"> Textstring als Seitentitel (Text wird zentriert ausgegeben) Text kann mit Zeilentrenner ' ' (ASCII-Zeichen 124 bzw. 7CH) auf mehrere Zeilen aufgeteilt werden. In diesem Fall wird der Text linksbündig dargestellt.
Font	Font.FNT	<ul style="list-style-type: none"> optionaler Font für den Titel
X,Y,DX,DY	<i>Integerwerte</i>	<ul style="list-style-type: none"> Position, Breite und Höhe der Bildseite (Bezug links oben!) Relative Position zum aktuellen Fenster: → durch Angabe von +/-X bzw. +/-Y wird das Fenster relativ zum aktuellen Fenster positioniert
Color	<i>Farbname oder Farbnummer</i>	<ul style="list-style-type: none"> Farbe des Titeltexes
Backcolor	<i>Farbname oder Farbnummer</i>	<ul style="list-style-type: none"> Hintergrundfarbe der Bildseite
Format		<ul style="list-style-type: none"> ohne Eingabe kein Rand
	Border=Button	<ul style="list-style-type: none"> Umrandungstyp Button
	Border=Input	<ul style="list-style-type: none"> Umrandungstyp Eingabefeld
	Border=Rx	<ul style="list-style-type: none"> Umrandungstyp Rechteck mit Breite x Pixel (1, 3, 5, etc.)
	Border=Shadow	<ul style="list-style-type: none"> Umrandungstyp Rechteck mit Schatten (3D Effekt)
	Border=Signal	<ul style="list-style-type: none"> Umrandungstyp Anzeigefeld
Action		
Limit1		
Limit2		
Action Limit1		
Action Limit2		
VarValue		
VarType		
VarState		
Option		<ul style="list-style-type: none"> ohne Eingabe normale Bildseite
	ID=x	<ul style="list-style-type: none"> Seitennummer die in der Systemvariablen s_pageidx gesetzt wird, wenn die Seite aktiv ist (s.a. Systemvariablen)
	Page=Dialog	<ul style="list-style-type: none"> Eingaben nur in aktueller Bildseite erlaubt
	Transparency=Farbname	<ul style="list-style-type: none"> Farbname, welcher transparent interpretiert wird (in Verbindung mit einer PCX-Bilddatei)

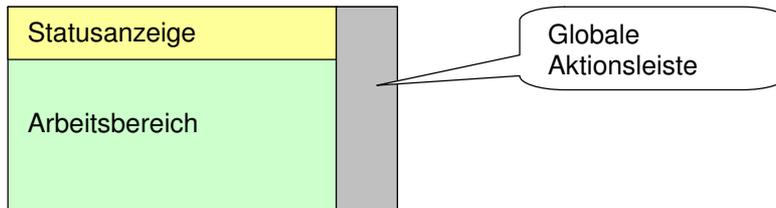
C-Function	C-Funktionsname	<ul style="list-style-type: none"> Name der C-Funktion (siehe Einbindung von C-Funktionen)
------------	-----------------	---



Frames

Eine Bildseite kann durch Definition von mehreren Pageobjekten in verschiedene Bereiche (Frames) aufgeteilt werden. In den einzelnen Bereichen können Bildinhalte unabhängig voneinander geändert werden. (Beispiel: siehe Demoprojekt „#Page=ObjectMeter“)
 Damit können globale Aktionsleisten oder Statusanzeigen an einer Stelle zentral und seitenunabhängig definiert werden.

Beispiel:



Fenstertechnik

Bildseiten mit unterschiedlicher Dimension werden übereinander gelegt und gleichzeitig am Bildschirm dargestellt und aktualisiert (Fenstertechnik). Objekte die von einer überlagerten Bildseite teilweise oder ganz verdeckt sind, werden als unsichtbar gekennzeichnet und nicht mehr aktualisiert. Alle anderen Objekte bleiben voll bedienbar (Ausnahme: Option Page=Dialog). Eine überlagerte Bildseite kann mit der Aktion close geschlossen werden, wenn die darunter liegende Bildseite grösser ist (DX oder DY). Die Aktion ‚close‘ bleibt ohne Wirkung, wenn nur eine sichtbare Bildseite aktiv ist.

7.1.1 Projektierung globaler Objekte

Object	Text/	File	Font	X	Y	DX	DY	Color	Backg.	Action	Value	VarType
#Page=Init				0	0	0	0					
#Password												
#Recipe												
#Alarm												
#Alarmmail												
#DataLog												
#ScreenSaver												
#Signal										#Page=MyActionPage	MyVar	INT
#Page=Start				0	0	640	480					
....												
....												
#Page=MyPage												
...												
...												

Globale Seite („Initialisierungs-Seite“)
 - im Projekt als erste Seite definiert
 - DX=0 und DY=0

Globale projizierte Objekte
 - Password, Recipe, etc.

Direkt der „Initialisierungs-Seite“ folgende Bildseite.
 Diese Bildseite ist nach dem Start von EPAM ersichtlich.



Globale Objekte

Globale Objekte müssen **in der ersten Bildseite** im Tabellenblatt „Project“ („Initialisierungs-Seite“) projiziert werden. Diese Seite muss mit den Dimensionen DX=0 und DY=0 definiert werden. Somit ist die Seite als globale Bildseite definiert und immer aktiv. Somit lassen sich auch Bildseitenwechsel von der PLC aus steuern (mit Variablengrenzwert und Action Limit #Page=x).

7.2 Objekt Button

#Button		nicht einrastender, touchaktiver Bereich
Text/File	<i>Up.PCX,Down.PCX</i>	<ul style="list-style-type: none"> Name von PCX-Bilddateien für Zustände „nicht betätigt“ und „betätigt“ durch ‘,’ getrennt
	<i>Up.ICO,Down.ICO</i>	<ul style="list-style-type: none"> Name von PCX-Bilddateien mit der Dateierweiterung .ICO für Zustände „nicht betätigt“ und „betätigt“ durch ‘,’ getrennt werden als Icons zentriert im Button dargestellt
	UpText,DownText	<ul style="list-style-type: none"> Textstrings für Zustände „nicht betätigt“ und „betätigt“ durch ‘,’ getrennt Text kann mit Zeilentrenner ‘ ’ (ASCII-Zeichen 124 bzw. 7CH) auf mehrere Zeilen aufgeteilt werden
	<i>Icon.ICO</i>	<ul style="list-style-type: none"> PCX-Bilddatei mit der Dateierweiterung .ICO wird als Icon zentriert im Button dargestellt
	<i>Text</i>	<ul style="list-style-type: none"> Textstring (Text wird zentriert ausgegeben) Text kann mit Zeilentrenner ‘ ’ (ASCII-Zeichen 124 bzw. 7CH) auf mehrere Zeilen aufgeteilt werden
Font	<i>Font.FNT</i>	<ul style="list-style-type: none"> optionaler Font für den Text
X,Y,DX,DY	<i>Integerwerte</i>	<ul style="list-style-type: none"> Position, Breite und Höhe des Buttons (Bezug links oben!)
Color	<i>Farbname oder Farbnummer UpColor,Downcolor</i>	<ul style="list-style-type: none"> Farbe des Textes für Zustände „nicht betätigt“ und „betätigt“
BackColor	<i>Farbname oder Farbnummer UpBackColor,DownBackColor</i>	<ul style="list-style-type: none"> Hintergrundfarbe des Buttons für Zustände „nicht betätigt“ und „betätigt“
Format	Border=Button	<ul style="list-style-type: none"> Umrandungstyp Button bei Text- bzw. Iconausgabe fest vordefiniert
	Invisible	<ul style="list-style-type: none"> Unsichtbares, touchaktives Feld (wird bei Betätigung nicht invertiert)
Action0, Action1, ... oder Action1 & Action2 & ...	Allgemeine Aktionen	
	#Page=Name	<ul style="list-style-type: none"> Bildseitenwechsel zur Bildseite <i>Name</i>
	#PagePrev	<ul style="list-style-type: none"> Bildseitenwechsel zur letzten Bildseite
	#PageHome	<ul style="list-style-type: none"> Bildseitenwechsel zur 1. Bildseite (Startseite)
	Close	<ul style="list-style-type: none"> Bildseite (Fenster) schliessen
	Close=Name	<ul style="list-style-type: none"> Bildseite (Fenster) <i>Name</i> schliessen
	EjectVolume(Drive #Page=eject_failed #Page=eject_ok)	<ul style="list-style-type: none"> Abmelden eines Wechselspeichers (z.B. USB-Memorystick; nur Windows) Drive ...Bezeichnung des Wechselspeichers (z.B. StorageCard2 oder F:) #Page=eject_failed ...optionale Page die bei Fehler angezeigt wird (Wechselspeicher können nur abgemeldet werden, wenn kein Programm mehr darauf zugreift!) #Page=eject_ok ...optionale Page die bei erfolgreicher Abmeldung angezeigt wird (bedingt auch Angabe von #Page=eject_failed)
	Exit	<ul style="list-style-type: none"> Programm verlassen (zurück zum Betriebssystem)
FileCopy(dst=path\file.ext src=path\file.ext #Page=copy_err #Page=copy_ok)	<ul style="list-style-type: none"> Kopieren einer Datei „src“ nach „dst“ #Page=copy_err ...optionale Page die bei Fehler angezeigt wird #Page=copy_ok ...optionale Page die bei erfolgreichem Kopieren angezeigt wird (bedingt auch Angabe von #Page=copy_err) #Page= ... optional kopieren ohne Fehlermeldung 	

Key=keycode	<ul style="list-style-type: none"> Zuordnung eines Tastencodes bzw. Tastennames (siehe auch Tastaturtabelle): ASCII-Zeichen oder \xnnnn (HEX-Tastaturcode)
Language=default	<ul style="list-style-type: none"> Online Sprachumschaltung zur Defaultsprache
Language=name	<ul style="list-style-type: none"> Online Sprachumschaltung zur Sprache <i>name</i>
PLCcmd=[[/Driver/][Host]:Command	<ul style="list-style-type: none"> SPS-Commando auslösen (Abh. Von Kommunikationstreiber) Commands: <ul style="list-style-type: none"> CreateBootProject ResetCold ResetOriginal ResetWarm Start Stop
PrintScreen	<ul style="list-style-type: none"> Bildschirminhalt auf Standard-Drucker ausgeben (nur Windows)
rcinput_enable=	<ul style="list-style-type: none"> Nodisable RemoteControl Input Yesenable RemoteControl Input
RemoteClient=drop	<ul style="list-style-type: none"> Beendet Remote Client
Reboot	<ul style="list-style-type: none"> Neustart des Systems (systemabhängig!)
SetIndex	<ul style="list-style-type: none"> Indirektes Setzen der Indexvariable für indizierten Variablenzugriff, d.H. die Indexvariable muss vorgängig eingegeben worden sein
SetIndex=x	<ul style="list-style-type: none"> Direktes Setzen der Indexvariable für indizierten Variablenzugriff
SetVar=x	<ul style="list-style-type: none"> Variablenwert auf x setzen (Strings müssen mit einschliessenden, einfachen Hochkomma z.B. 'String' definiert werden) x kann auch der Name einer Systemvariablen sein
SetVar+x	<ul style="list-style-type: none"> Variablenwert um x inkrementieren
SetVar-x	<ul style="list-style-type: none"> Variablenwert um x dekrementieren
SetVar=NotVar	<ul style="list-style-type: none"> Variablenwert invertieren (0/1)
System=myprg.exe	<ul style="list-style-type: none"> Aufruf eines Programms (nur Windows)
Msg=x	<ul style="list-style-type: none"> Meldung mit Nummer x ausgeben
PWL=x	<ul style="list-style-type: none"> Passwortlevel auf x (rück)setzen
TipVar=x	<ul style="list-style-type: none"> Variablenwert auf x setzen, solange Button gedrückt, anschliessend wird die Variable wieder auf 0 gesetzt
Touch_calibrate	<ul style="list-style-type: none"> Resistiv-Touch kalibrieren
Systemspezifische Aktionen	
Backlight=x	<ul style="list-style-type: none"> Hintergrundbeleuchtung setzen (0-100%)
Backlight+x	<ul style="list-style-type: none"> Hintergrundbeleuchtung um x inkrementieren
Backlight-x	<ul style="list-style-type: none"> Hintergrundbeleuchtung um x dekrementieren
CFGINI=Read	<ul style="list-style-type: none"> IP-Konfiguration lesen
CFGINI=Write	<ul style="list-style-type: none"> IP-Konfiguration schreiben
Contrast=x	<ul style="list-style-type: none"> Kontrast setzen (0-100%) (nur passiv-LCD)
Contrast+x	<ul style="list-style-type: none"> Kontrast um x inkrementieren (nur passiv-LCD)
Contrast-x	<ul style="list-style-type: none"> Kontrast um x dekrementieren (nur passiv-LCD)
GetDT	<ul style="list-style-type: none"> aktualisieren sämtlicher RTC-Systemvariablen s_tm_day, s_tm_mon, etc.
Save=SysVar	<ul style="list-style-type: none"> Systemvariablen in sysvar.ini speichern
SetDate	<ul style="list-style-type: none"> Systemzeit setzen (Werte werden aus RTC-Systemvariablen übernommen)
SetTime	<ul style="list-style-type: none"> Systemdatum setzen (Werte werden aus RTC-Systemvariablen übernommen)
Objektspezifische Aktionen in Verbindung mit Objekt Scrollist	

Scrollx=x	<ul style="list-style-type: none"> Objekte in Scroll-Liste um x Pixel horizontal verschieben
Scrolly=x	<ul style="list-style-type: none"> Objekte in Scroll-Liste um x Pixel vertikal verschieben
Objektspezifische Aktionen in Verbindung mit Objekt Alarm/Alarmlist	
AlarmDelete	<ul style="list-style-type: none"> Alarmhistory löschen
AlarmExport=CSV	<ul style="list-style-type: none"> Alarmhistory wird als CSV-Datei ins EPAM-Dataverzeichnis C:\DATA gespeichert
AlarmFilter=activ	<ul style="list-style-type: none"> Alarmfilter setzen: aktive Alarmer anzeigen
AlarmFilter=activ notquit	<ul style="list-style-type: none"> Alarmfilter setzen: aktive oder nicht quittierte Alarmer anzeigen
AlarmFilter=activ+notquit	<ul style="list-style-type: none"> Alarmfilter setzen: aktive und nicht quittierte Alarmer anzeigen
AlarmFilter=all	<ul style="list-style-type: none"> Alarmfilter setzen: alle Alarmer anzeigen
AlarmFilter=notquit	<ul style="list-style-type: none"> Alarmfilter setzen: nicht quittierte Alarmer anzeigen
AlarmInfo=1 bzw. 2	<ul style="list-style-type: none"> Alarminfo des selektierten Alarms aufrufen
AlarmQuit	<ul style="list-style-type: none"> selektierten Alarm einzeln quittieren
AlarmQuitall	<ul style="list-style-type: none"> alle Alarmer quittieren
AlarmSort=FIFO	<ul style="list-style-type: none"> Alarm in Alarmliste sortieren: ältester Alarm zuerst
AlarmSort=LIFO	<ul style="list-style-type: none"> Alarm in Alarmliste sortieren: neuester Alarm zuerst
AlarmSort=Priority	<ul style="list-style-type: none"> Alarm in Alarmliste sortieren: Alarm mit hoher Priorität (=niedrige Alarmnummer) zuerst
AlarmType=myalarmtype	<ul style="list-style-type: none"> Alarmtyp der Alarmliste setzen (nötig sobald mehrere Alarmobjekte projiziert wurden)
Objektspezifische Aktionen in Verbindung mit Objekt Recipe/RecipeList	
Csave=list	<ul style="list-style-type: none"> Element aus Rezeptliste (z.B. myRecipeType) speichern, mit Abfrage falls Datei existiert (siehe Objekt Recipe)
Csave=myrecipetype	<ul style="list-style-type: none"> Rezepturtyp (z.B. myRecipeType) speichern, mit Abfrage falls Datei existiert, der Dateiname wird aus der Systemvariablen 's_myrecipetype_file' entnommen
Delete=list	<ul style="list-style-type: none"> Element aus Rezeptliste (z.B. myRecipeType) löschen
Delete=myrecipetype	<ul style="list-style-type: none"> Rezepturtyp (z.B. myRecipeType) löschen, der Dateiname wird aus der Systemvariablen 's_myrecipetype_file' entnommen
Load_dat=LW:	<ul style="list-style-type: none"> Alle Rezeptfiles *.DAT von Laufwerk LW: laden
Load=list	<ul style="list-style-type: none"> Element aus Rezeptliste (z.B. myRecipeType) laden
Load=myrecipetype	<ul style="list-style-type: none"> Rezepturtyp (z.B. myRecipeType) laden, der Dateiname wird aus der Systemvariablen 's_myrecipetype_file' entnommen
Load=C:\DATA\MyType1\myrecipefile.dat	<ul style="list-style-type: none"> Rezepturdatei (z.B. C:\DATA\MYTYP1\REC1.DAT) laden
Save_dat=LW:	<ul style="list-style-type: none"> alle Rezeptdateien *.DAT auf Laufwerk LW: kopieren
Save=list	<ul style="list-style-type: none"> Element aus Rezeptliste (z.B. myRecipeType) speichern, bestehende Dateien werden überschrieben
Save=myrecipetype	<ul style="list-style-type: none"> Rezepturtyp (z.B. myRecipeType) speichern, bestehende Dateien werden überschrieben, Dateiname und Rezeptname werden aus den Systemvariablen 's_myrecipetype_file' bzw. 's_myrecipetype_name' entnommen
Sort=File	<ul style="list-style-type: none"> Rezeptliste nach Dateiname sortieren
Sort=Name	<ul style="list-style-type: none"> Rezeptliste nach Rezeptname sortieren
Sort=Number	<ul style="list-style-type: none"> Rezeptliste nach Rezeptname numerisch sortieren (nur bei Verwendung numerischer Dateinamen möglich)
Sort=Time	<ul style="list-style-type: none"> Rezeptliste nach Zeit sortieren
Sort=Type	<ul style="list-style-type: none"> Rezeptliste nach Rezepttyp sortieren
Type=myrecipetype	<ul style="list-style-type: none"> Rezepturtyp (z.B. myRecipeType) setzen
Type=off	<ul style="list-style-type: none"> Rezepturtyp rücksetzen (alle)
Objektspezifische Aktionen in Verbindung mit Objekt DataLog	
LogDelete=mydatalog	<ul style="list-style-type: none"> Datalogdatei in LOG-Verzeichnis löschen
LogSave=mydatalog	<ul style="list-style-type: none"> Datalogdatei ins DATA-Verzeichnis speichern

	Save_log=LW:	<ul style="list-style-type: none"> alle Datalogdateien *.DAT auf Laufwerk LW: kopieren
	Objektspezifische Aktionen in Verbindung mit Objekt Trend	
	Online	<ul style="list-style-type: none"> Trend in Online-Modus schalten
	ShiftCursor=x	<ul style="list-style-type: none"> Trend scrollen um +/-x Datenpunkte.
	ShiftGrid=x	<ul style="list-style-type: none"> Trend scrollen um +/-x Zeiteinheiten
	ShiftPage=x	<ul style="list-style-type: none"> Trend scrollen um +/-x Seiten
	Zoom-	<ul style="list-style-type: none"> Zoom Trend (Auflösung Zeitachse um eine Einheit verkleinern)
	Zoom+	<ul style="list-style-type: none"> Zoom Trend (Auflösung Zeitachse um eine Einheit vergrössern)
	ZoomX-	<ul style="list-style-type: none"> Zoom Trend (Auflösung X-Achse um eine Einheit verkleinern)
	ZoomX+	<ul style="list-style-type: none"> Zoom Trend (Auflösung X-Achse um eine Einheit vergrössern)
	ZoomY-	<ul style="list-style-type: none"> Zoom Trend (Auflösung Y-Achse um eine Einheit verkleinern)
	ZoomY+	<ul style="list-style-type: none"> Zoom Trend (Auflösung Y-Achse um eine Einheit vergrössern)
Limit1	<i>Wert/Variablenname</i>	<ul style="list-style-type: none"> Unterer Grenzwert Wert, PLC-Variable bzw. Systemvariable für Grenzwert (liegt der aktuelle Wert ausserhalb des Grenzwertes, wird der Button gesperrt)
Limit2	<i>Wert/Variablenname</i>	<ul style="list-style-type: none"> Oberer Grenzwert Wert, PLC-Variable bzw. Systemvariable für Grenzwert (liegt der aktuelle Wert ausserhalb des Grenzwertes, wird der Button gesperrt)
Action Limit1 Action Limit2	#Page=Name	<ul style="list-style-type: none"> Bildseitenwechsel zur Bildseite <i>Name</i>
	Alarm=x	<ul style="list-style-type: none"> Alarm x auslösen (x ist Alarmnummer)
	BackColor=x	<ul style="list-style-type: none"> Hintergrundfarbe auf x ändern
	Backlight=x	<ul style="list-style-type: none"> Hintergrundbeleuchtung setzen (0-100%)
	Close	<ul style="list-style-type: none"> Bildseite (Fenster) schliessen
	Close=Name	<ul style="list-style-type: none"> Bildseite (Fenster) <i>Name</i> schliessen
	Color=x	<ul style="list-style-type: none"> Farbe auf x ändern
	Contrast=x	<ul style="list-style-type: none"> Kontrast setzen (0-100%) (nur passiv-LCD)
	Exit	<ul style="list-style-type: none"> Programm verlassen (zurück zum Betriebssystem)
	FastFlash	<ul style="list-style-type: none"> Objektstatus auf Blinken mit 2 Hz setzen
	Flash	<ul style="list-style-type: none"> Objektstatus auf Blinken mit 1 Hz setzen
	Language=name	<ul style="list-style-type: none"> Online Sprachumschaltung zur Sprache <i>name</i>
	Language=s_msysvar	<ul style="list-style-type: none"> Online Sprachumschaltung zur Sprache in s_msysvar
	Load=x	<ul style="list-style-type: none"> Rezeptdatei mit Namen x laden
	Msg=x	<ul style="list-style-type: none"> Meldung mit Nummer x ausgeben
	PWL=x	<ul style="list-style-type: none"> PWL auf x setzen
	SetVar=x	<ul style="list-style-type: none"> Variablenwert auf x setzen
	SetVar=Limit1	<ul style="list-style-type: none"> Variablenwert auf Limit1 setzen
	SetVar=Limit2	<ul style="list-style-type: none"> Variablenwert auf Limit2 setzen
	s_msysvar=x	<ul style="list-style-type: none"> Systemvariablenwert auf x setzen
VarValue	<i>Variablenname</i>	<ul style="list-style-type: none"> PLC-Variable bzw. Systemvariable
VarType	BOOL	<ul style="list-style-type: none"> Datentyp Bool (8 Bit)
	BYTE	<ul style="list-style-type: none"> Datentyp Byte (8 Bit)
	DINT	<ul style="list-style-type: none"> Datentyp Doppelinteger (32 Bit)
	DWORD	<ul style="list-style-type: none"> Datentyp Doppelwort (32 Bit)
	IEC_DT	<ul style="list-style-type: none"> Datentyp IEC_DT (32 Bit)
	IEC_TIME	<ul style="list-style-type: none"> Datentyp IEC_TIME (32 Bit)
	INT	<ul style="list-style-type: none"> Datentyp Integer (16 Bit)
	REAL	<ul style="list-style-type: none"> Datentyp Gleitkomma (32 Bit)
	SINT	<ul style="list-style-type: none"> Datentyp Shortinteger (8 Bit)
	STRING	<ul style="list-style-type: none"> Datentyp String 80 Byte (ohne Angabe: default 80 Byte)

	STRING:xx	<ul style="list-style-type: none"> Datentyp String xx Byte
	UDINT	<ul style="list-style-type: none"> Datentyp Doppelinteger ohne Vorzeichen (32 Bit)
	UINT	<ul style="list-style-type: none"> Datentyp Integer ohne Vorzeichen (16 Bit)
	USINT	<ul style="list-style-type: none"> Datentyp Shortinteger ohne Vorzeichen (8 Bit)
	WORD	<ul style="list-style-type: none"> Datentyp Wort (16 Bit)
VarState	<i>Variablenname</i>	<ul style="list-style-type: none"> PLC-Variable bzw. Systemvariable für Objektstatus
Option	Key=keycode	<ul style="list-style-type: none"> Zuordnung eines Tastencodes bzw. Tastennames (siehe auch Tastaturtabelle) zur Aktivierung der Button-Aktion: ASCII-Zeichen oder \xnnnn (HEX-Tastaturcode)
	NoBeep	<ul style="list-style-type: none"> Objektspezifischer Touch Beep deaktivieren
	Pos=Center	<ul style="list-style-type: none"> Text zentriert
	Pos=Left	<ul style="list-style-type: none"> Text linksbündig
	Pos=Right	<ul style="list-style-type: none"> Text rechtsbündig
	PWL=x	<ul style="list-style-type: none"> notwendiger Passwortlevel zur Freigabe
	Scroll	<ul style="list-style-type: none"> Position des Objektes kann mit Scroll-Objekt geändert werden
	Switch=x	<ul style="list-style-type: none"> x = Wertebereich für Action0,Action1,... z.B.: <0:1:2..5:>5 Erlaubte Werte: Konstante Zahl z.B. 5 <Zahl ...kleiner als >Zahl ...grösser als Zahl..Zahl ...Bereich von bis : ...Seperator
	Timeout=x	<ul style="list-style-type: none"> Timeout in Sekunden nach der die Aktion automatisch ausgeführt wird (repetierend) Timeout=0 führt Aktion einmal aus
	Transparency=Farbname	<ul style="list-style-type: none"> Farbname, welcher transparent interpretiert wird (in Verbindung mit einer PCX-Bilddatei)
	Type=dynamic	<ul style="list-style-type: none"> Button-Aktion wird auch bei Maus-Move Events ausgeführt
C-Function	<i>C-Funktionsname</i>	<ul style="list-style-type: none"> Name der C-Funktion (siehe Einbindung von C-Funktionen)



Bildschirmtastatur

Die Aktion **Key=keycode** dient zur Erstellung von Bildschirmtastaturen. Hierbei wird bei Betätigung des maus- bzw. touchaktiven Feldes der entsprechende Tastencode generiert und somit die Tastatur simuliert (siehe auch Tastaturtabelle mit Tastenbezeichnungen).

Beispiel: **Key=a** ...simuliert die Taste „a“



Mehrsprachige Applikationen

Die Aktion **Language=name** bzw. **Language=Default** ermöglicht eine online Sprachumschaltung für beliebige Sprachen. Weitere Sprachen werden dabei in entsprechenden Unterverzeichnissen abgelegt. *name* bezeichnet das jeweilige Unterverzeichnis.

Um eine Applikation mehrsprachig zu erstellen, definieren Sie mit dem EPAM-Makro „Define Language“ eine neue Sprache. Es wird nun ein Verzeichnis mit dem eingegebenen Namen (max. 8 Zeichen) angelegt und in allen sprachabhängigen Tabellen am Ende zwei neue Spalten für Text/File bzw. Font eingefügt.



Wichtig!

Die Bezeichnung in der ersten Zeile dieser Spalten entspricht dem Verzeichnisnamen und als Notiz ist „Language“ hinterlegt (Kennzeichnung der Sprachspalten!). Diese Bezeichnungen dürfen nicht entfernt werden!

Nun können Sie alle sprachabhängigen Definitionen in den entsprechenden Sprachspalten definieren (die sprachunabhängigen Definitionen können Sie leer lassen, diese werden dann automatisch von der Defaultsprache übernommen). Bei Bedarf können Sie auch andere Fonts definieren.

Alle sprachabhängigen Dateien (*.TXT, *.FNT, *.PCX, *.ICO) müssen im entsprechenden Unterverzeichnis abgelegt werden.

Wenn Sie nun das nächste mal den EPAM-Makro „Start EPAM“ aufrufen um den Interpreter zu starten, so werden automatisch alle Sprachdateien erstellt.



Variable Aktion

Es können mehrere Aktionen durch Komma getrennt definiert werden. In diesem Fall wird abhängig vom aktuellen Variablenwert (0, 1, 2, ...) die entsprechende Aktion ausgeführt. So können z.B. in Abhängigkeit einer Variable verschiedene Bildseiten angewählt werden.

Beispiel:

Aktion #Page=Wert0,#Page=Wert1

...Wechsel auf die Bildseite „Wert0“, wenn der Variablenwert 0 ist

...Wechsel auf die Bildseite „Wert1“, wenn der Variablenwert 1 ist

Mit Hilfe der **Option ,switch=’** können auch Wertebereiche für die einzelnen Aktionen definiert werden.

Beispiel:

Aktion #Page=Range0,#Page=Range1,#Page=Range3 mit Option:switch=<0:0..5:>5

...Wechsel auf die Bildseite „Range0“, wenn der Variablenwert < 0 ist

...Wechsel auf die Bildseite „Range1“, wenn der Variablenwert im Bereich 0 bis 5 liegt

...Wechsel auf die Bildseite „Range2“, wenn der Variablenwert > 5 ist

7.3 Objekt Switch

#Switch		einrastender, touchaktiver Bereich
Text/File	<i>Bild0.PCX,Bild1.PCX,...</i>	<ul style="list-style-type: none"> Name von PCX-Bilddateien für Zustände 0,1,... durch ',' getrennt
	<i>Icon0.ICO,Icon1.ICO,...</i>	<ul style="list-style-type: none"> Name von PCX-Bilddateien mit der Dateierweiterung .ICO für Zustände 0,1,... durch ',' getrennt werden als Icons zentriert im Button dargestellt
	<i>Text0,Text1,...</i>	<ul style="list-style-type: none"> Textstrings für Zustände 0,1,... durch ',' getrennt Text kann mit Zeilentrenner ' ' (ASCII-Zeichen 124 bzw. 7CH) auf mehrere Zeilen aufgeteilt werden
Font	<i>Font.FNT</i>	<ul style="list-style-type: none"> optionaler Font für den Text
X,Y,DX,DY	<i>Integerwerte</i>	<ul style="list-style-type: none"> Position, Breite und Höhe des Switches (Bezug links oben!)
Color	<i>Farbname oder Farbnummer Color0,Color1,...</i>	<ul style="list-style-type: none"> Farbe des Textes für Zustände 0,1,...
BackColor	<i>Farbname oder Farbnummer BackColor0,BackColor1,...</i>	<ul style="list-style-type: none"> Hintergrundfarbe des Switches für Zustände 0,1,...
Format	Border=Button	<ul style="list-style-type: none"> Umrandungstyp Button bei Text- bzw. Iconausgabe fest vordefiniert
Action	SetVar+1	<ul style="list-style-type: none"> bei Betätigung des Switch-Objektes wird der Variablenwert automatisch um 1 inkrementiert und die zugehörige Bild/Textinformation angezeigt. Ist der aktuelle Wert grösser/gleich als die Anzahl definierter Zustände - 1, so wird der Wert 0 gesetzt
Limit1	<i>Wert/Variablenname</i>	<ul style="list-style-type: none"> Unterer Grenzwert Wert, PLC-Variable bzw. Systemvariable für Grenzwert
Limit2	<i>Wert/Variablenname</i>	<ul style="list-style-type: none"> Oberer Grenzwert Wert, PLC-Variable bzw. Systemvariable für Grenzwert
Action Limit1 Action Limit2	#Page=Name	<ul style="list-style-type: none"> Bildseitenwechsel zur Bildseite <i>Name</i>
	Alarm=x	<ul style="list-style-type: none"> Alarm x auslösen (x ist Alarmnummer)
	BackColor=x	<ul style="list-style-type: none"> Hintergrundfarbe auf x ändern
	Backlight=x	<ul style="list-style-type: none"> Hintergrundbeleuchtung setzen (0-100%)
	Close	<ul style="list-style-type: none"> Bildseite (Fenster) schliessen
	Close=Name	<ul style="list-style-type: none"> Bildseite (Fenster) <i>Name</i> schliessen
	Color=x	<ul style="list-style-type: none"> Farbe auf x ändern
	Contrast=x	<ul style="list-style-type: none"> Kontrast setzen (0-100%) (nur passiv-LCD)
	Exit	<ul style="list-style-type: none"> Programm verlassen (zurück zum Betriebssystem)
	FastFlash	<ul style="list-style-type: none"> Objektstatus auf Blinken mit 2 Hz setzen
	Flash	<ul style="list-style-type: none"> Objektstatus auf Blinken mit 1 Hz setzen
	Language=name	<ul style="list-style-type: none"> Online Sprachumschaltung zur Sprache <i>name</i>
	Language=s_msysvar	<ul style="list-style-type: none"> Online Sprachumschaltung zur Sprache in s_msysvar
	Load=x	<ul style="list-style-type: none"> Rezeptdatei mit Namen x laden
	Msg=x	<ul style="list-style-type: none"> Meldung mit Nummer x ausgeben
	PWL=x	<ul style="list-style-type: none"> PWL auf x setzen
	SetVar=x	<ul style="list-style-type: none"> Variablenwert auf x setzen
SetVar=Limit1	<ul style="list-style-type: none"> Variablenwert auf Limit1 setzen 	
SetVar=Limit2	<ul style="list-style-type: none"> Variablenwert auf Limit2 setzen 	
s_msysvar=x	<ul style="list-style-type: none"> Systemvariablenwert auf x setzen 	
VarValue	<i>Variablenname</i>	<ul style="list-style-type: none"> PLC-Variable bzw. Systemvariable
VarType	BOOL	<ul style="list-style-type: none"> Datentyp Bool (8 Bit)
	BYTE	<ul style="list-style-type: none"> Datentyp Byte (8 Bit)
	SINT	<ul style="list-style-type: none"> Datentyp Shortinteger (8 Bit)
	USINT	<ul style="list-style-type: none"> Datentyp Shortinteger ohne Vorzeichen (8 Bit)

	INT	• Datentyp Integer (16 Bit)
	UINT	• Datentyp Integer ohne Vorzeichen (16 Bit)
	WORD	• Datentyp Wort (16 Bit)
VarState	<i>Variablenname</i>	• PLC-Variable bzw. Systemvariable für Objektstatus
Option	Pos=Center	• Text zentriert
	Pos=Left	• Text linksbündig
	Pos=Right	• Text rechtsbündig
	PWL=x	• notwendiger Passwortlevel zur Freigabe
	Scroll	• Position des Objektes kann mit Scroll-Objekt geändert werden
	Transparency=Farbname	• Farbname, welcher transparent interpretiert wird (in Verbindung mit einer PCX-Bilddatei)
C-Function	<i>C-Funktionsname</i>	• Name der C-Funktion (siehe Einbindung von C-Funktionen)

7.4 Objekt DropDownList

#DropDownList		Auswahl eines Elements aus einer statischen DropDownList
Text/File	<i>Bild0.PCX,Bild1.PCX,...</i>	<ul style="list-style-type: none"> Name von PCX-Bilddateien für Zustände 0,1,... durch ',' getrennt
	<i>Icon0.ICO,Icon1.ICO,...</i>	<ul style="list-style-type: none"> Name von PCX-Bilddateien mit der Dateierweiterung .ICO für Zustände 0,1,... durch ',' getrennt werden als Icons zentriert im Button dargestellt
	<i>Text0,Text1,...</i>	<ul style="list-style-type: none"> Textstrings für Zustände 0,1,... durch ',' getrennt Text kann mit Zeilentrenner '\n' (ASCII-Zeichen 124 bzw. 7CH) auf mehrere Zeilen aufgeteilt werden
Font	<i>Font.FNT</i>	<ul style="list-style-type: none"> optionaler Font für den Text
X,Y,DX,DY	<i>Integerwerte</i>	<ul style="list-style-type: none"> Position, Breite und Höhe des DropDownListfelds (Bezug links oben!)
Color	<i>Farbname oder Farbnummer</i>	<ul style="list-style-type: none"> Schriftfarbe
BackColor	<i>Farbname oder Farbnummer</i>	<ul style="list-style-type: none"> Hintergrundfarbe
Format		<ul style="list-style-type: none"> ohne Eingabe kein Rand
	Border=Button	<ul style="list-style-type: none"> Umrandungstyp Button
	Border=Input	<ul style="list-style-type: none"> Umrandungstyp Eingabefeld
	Border=Rx	<ul style="list-style-type: none"> Umrandungstyp Rechteck mit Breite x Pixel (1, 3, 5, etc.)
	Border=Shadow	<ul style="list-style-type: none"> Umrandungstyp Rechteck mit Schatten (3D Effekt)
Border=Signal	<ul style="list-style-type: none"> Umrandungstyp Anzeigefeld 	
Action		<ul style="list-style-type: none"> bei Betätigung des DropDownList-Objektes wird der Variablenwert automatisch auf den Wert der zugehörigen Bild/Textinformation gesetzt. (z.B. Auswahl Bild1.pcx → Wert = 1)
Limit1	<i>Wert/Variablenname</i>	<ul style="list-style-type: none"> Unterer Grenzwert Wert, PLC-Variable bzw. Systemvariable für Grenzwert
Limit2	<i>Wert/Variablenname</i>	<ul style="list-style-type: none"> Oberer Grenzwert Wert, PLC-Variable bzw. Systemvariable für Grenzwert
Action Limit1 Action Limit2	#Page=Name	<ul style="list-style-type: none"> Bildseitenwechsel zur Bildseite <i>Name</i>
	Alarm=x	<ul style="list-style-type: none"> Alarm x auslösen (x ist Alarmnummer)
	BackColor=x	<ul style="list-style-type: none"> Hintergrundfarbe auf x ändern
	Backlight=x	<ul style="list-style-type: none"> Hintergrundbeleuchtung setzen (0-100%)
	Close	<ul style="list-style-type: none"> Bildseite (Fenster) schliessen
	Close=Name	<ul style="list-style-type: none"> Bildseite (Fenster) <i>Name</i> schliessen
	Color=x	<ul style="list-style-type: none"> Farbe auf x ändern
	Contrast=x	<ul style="list-style-type: none"> Kontrast setzen (0-100%) (nur passiv-LCD)
	Exit	<ul style="list-style-type: none"> Programm verlassen (zurück zum Betriebssystem)
	FastFlash	<ul style="list-style-type: none"> Objektstatus auf Blinken mit 2 Hz setzen
	Flash	<ul style="list-style-type: none"> Objektstatus auf Blinken mit 1 Hz setzen
	Language=name	<ul style="list-style-type: none"> Online Sprachumschaltung zur Sprache <i>name</i>
	Language=s_msysvar	<ul style="list-style-type: none"> Online Sprachumschaltung zur Sprache in s_msysvar
	Load=x	<ul style="list-style-type: none"> Rezeptdatei mit Namen x laden
	Msg=x	<ul style="list-style-type: none"> Meldung mit Nummer x ausgeben
	PWL=x	<ul style="list-style-type: none"> PWL auf x setzen
SetVar=x	<ul style="list-style-type: none"> Variablenwert auf x setzen 	
SetVar=Limit1	<ul style="list-style-type: none"> Variablenwert auf Limit1 setzen 	
SetVar=Limit2	<ul style="list-style-type: none"> Variablenwert auf Limit2 setzen 	
s_msysvar=x	<ul style="list-style-type: none"> Systemvariablenwert auf x setzen 	
VarValue	<i>Variablenname</i>	<ul style="list-style-type: none"> PLC-Variable bzw. Systemvariable
VarType	BOOL	<ul style="list-style-type: none"> Datentyp Bool (8 Bit)
	BYTE	<ul style="list-style-type: none"> Datentyp Byte (8 Bit)

	SINT	• Datentyp Shortinteger (8 Bit)
	USINT	• Datentyp Shortinteger ohne Vorzeichen (8 Bit)
	INT	• Datentyp Integer (16 Bit)
	UINT	• Datentyp Integer ohne Vorzeichen (16 Bit)
	WORD	• Datentyp Wort (16 Bit)
VarState	<i>Variablenname</i>	• PLC-Variable bzw. Systemvariable für Objektstatus
Option	Coff	• Anzeige der Liste ohne Cursor
	DX=0	• Anzeige ohne Listenpfeil rechts
	LineHeight=x	• Zeilenhöhe der Liste (Default: Zeichenhöhe)
	Maxlines=x	• Anzahl Zeilen im Listenfeld (Default: Alle Elemente)
	Open=Down	• Listenfeld öffnet nach unten (Default)
	Open=Up	• Listenfeld öffnet nach oben
	Pos=Center	• Text zentriert
	Pos=Left	• Text linksbündig
	Pos=Right	• Text rechtsbündig
	PWL=x	• notwendiger Passwortlevel zur Freigabe
	Scroll	• Position des Objektes kann mit Scroll-Objekt geändert werden
	Transparency=Farbname	• Farbname, welcher transparent interpretiert wird (in Verbindung mit einer PCX-Bilddatei)
C-Function	<i>C-Funktionsname</i>	• Name der C-Funktion (siehe Einbindung von C-Funktionen)

7.5 Objekt Radiobutton

#RadioButton		<ul style="list-style-type: none"> touchaktiver Bereich, Auswahl einer aus mehreren Möglichkeiten
Text/File	<i>Bild0.PCX,Bild1.PCX</i>	<ul style="list-style-type: none"> Name von PCX-Bilddateien für Zustände „inaktiv“ und „aktiv“ durch ‘,’ getrennt
	<i>Icon0.ICO,Icon1.ICO</i>	<ul style="list-style-type: none"> Name von PCX-Bilddateien mit der Dateierweiterung .ICO für Zustände „inaktiv“ und „aktiv“ durch ‘,’ getrennt werden als Icons zentriert im Button dargestellt
	<i>Text0,Text1</i>	<ul style="list-style-type: none"> Textstrings für Zustände „inaktiv“ und „aktiv“ durch ‘,’ getrennt Text kann mit Zeilentrenner ‘]’ (ASCII-Zeichen 124 bzw. 7CH) auf mehrere Zeilen aufgeteilt werden
Font	<i>Font.FNT</i>	<ul style="list-style-type: none"> optionaler Font für den Text
X,Y,DX,DY	<i>Integerwerte</i>	<ul style="list-style-type: none"> Position, Breite und Höhe des Radiobuttons (Bezug links oben!)
Color	<i>Farbname oder Farbnummer Color0,Color1</i>	<ul style="list-style-type: none"> Farbe des Textes für Zustände „inaktiv“ und „aktiv“
Backcolor	<i>Farbname oder Farbnummer BackColor0,BackColor1</i>	<ul style="list-style-type: none"> Hintergrundfarbe des Radiobuttons für Zustände „inaktiv“ und „aktiv“
Format	Border=Button	<ul style="list-style-type: none"> Umrandungstyp Button bei Text- bzw. Iconausgabe fest vordefiniert
Action1 & Action2	#Page=Name	<ul style="list-style-type: none"> Bildseitenwechsel zur Bildseite <i>Name</i>
	SetIndex=x	<ul style="list-style-type: none"> Index für indizierten Variablenzugriff setzen
	SetVar=x	<ul style="list-style-type: none"> Variablenwert auf x setzen
Limit1	<i>Wert/Variablenname</i>	<ul style="list-style-type: none"> Unterer Grenzwert Wert, PLC-Variable bzw. Systemvariable für Grenzwert
Limit2	<i>Wert/Variablenname</i>	<ul style="list-style-type: none"> Oberer Grenzwert Wert, PLC-Variable bzw. Systemvariable für Grenzwert
Action Limit1 Action Limit2	#Page=Name	<ul style="list-style-type: none"> Bildseitenwechsel zur Bildseite <i>Name</i>
	Alarm=x	<ul style="list-style-type: none"> Alarm x auslösen (x ist Alarmnummer)
	Backcolor=x	<ul style="list-style-type: none"> Hintergrundfarbe auf x ändern
	Backlight=x	<ul style="list-style-type: none"> Hintergrundbeleuchtung setzen (0-100%)
	Close	<ul style="list-style-type: none"> Bildseite (Fenster) schliessen
	Close=Name	<ul style="list-style-type: none"> Bildseite (Fenster) <i>Name</i> schliessen
	Color=x	<ul style="list-style-type: none"> Farbe auf x ändern
	Contrast=x	<ul style="list-style-type: none"> Kontrast setzen (0-100%) (nur passiv-LCD)
	Exit	<ul style="list-style-type: none"> Programm verlassen (zurück zum Betriebssystem)
	FastFlash	<ul style="list-style-type: none"> Objektstatus auf Blinken mit 2 Hz setzen
	Flash	<ul style="list-style-type: none"> Objektstatus auf Blinken mit 1 Hz setzen
	Language=name	<ul style="list-style-type: none"> Online Sprachumschaltung zur Sprache <i>name</i>
	Language=s_mysysvar	<ul style="list-style-type: none"> Online Sprachumschaltung zur Sprache in <i>s_mysysvar</i>
	Load=x	<ul style="list-style-type: none"> Rezeptdatei mit Namen x laden
	Msg=x	<ul style="list-style-type: none"> Meldung mit Nummer x ausgeben
	PWL=x	<ul style="list-style-type: none"> PWL auf x setzen
SetVar=x	<ul style="list-style-type: none"> Variablenwert auf x setzen 	
SetVar=Limit1	<ul style="list-style-type: none"> Variablenwert auf Limit1 setzen 	
SetVar=Limit2	<ul style="list-style-type: none"> Variablenwert auf Limit2 setzen 	
s_mysysvar=x	<ul style="list-style-type: none"> Systemvariablenwert auf x setzen 	
VarValue	<i>Variablenname</i>	<ul style="list-style-type: none"> PLC-Variable bzw. Systemvariable
VarType	BOOL	<ul style="list-style-type: none"> Datentyp Bool (8 Bit)
	BYTE	<ul style="list-style-type: none"> Datentyp Byte (8 Bit)
	INT	<ul style="list-style-type: none"> Datentyp Integer (16 Bit)
	SINT	<ul style="list-style-type: none"> Datentyp Shortinteger (8 Bit)
	UINT	<ul style="list-style-type: none"> Datentyp Integer ohne Vorzeichen (16 Bit)

	USINT	• Datentyp Shortinteger ohne Vorzeichen (8 Bit)
	WORD	• Datentyp Wort (16 Bit)
VarState	<i>Variablenname</i>	• PLC-Variable bzw. Systemvariable für Objektstatus
Option	Pos=Center	• Text zentriert
	Pos=Left	• Text linksbündig
	Pos=Right	• Text rechtsbündig
	PWL=x	• notwendiger Passwortlevel zur Freigabe
	Scroll	• Position des Objektes kann mit Scroll-Objekt geändert werden
	Transparency=Farbname	• Farbname, welcher transparent interpretiert wird (in Verbindung mit einer PCX-Bilddatei)
C-Function	<i>C-Funktionsname</i>	• Name der C-Funktion (siehe Einbindung von C-Funktionen)



Funktionsweise des RadioButtons

Im Objekt Radiobutton wird der aktuelle Objektwert mit dem Sollwert der Aktion SetVar=x verglichen. Ist der Wert gleich dem Sollwert, so wird der entsprechende RadioButton aktiv dargestellt, ansonsten inaktiv. Wird der RadioButton betätigt, so wird der entsprechende Sollwert gesetzt.

Für eine Auswahl mit mehreren Elementen können nun mehrere RadioButton-Objekte definiert werden. Die Verknüpfung der einzelnen Auswahlelemente erfolgt durch Zuordnung derselben PLC-Variable.

7.6 Objekt Variable

#Variable		Darstellung einer numerischen/alphanumerischen Variable
Text/File	<p>Text</p> <p><code>%[Flags][Width].[Prec][Type]</code></p> <p>Text</p> <p><code>%[Format-Unit0],[Format-Unit1]</code></p>	<ul style="list-style-type: none"> Textstring mit Formatangabe (Bezug links, oben) Bsp. Geschw. <code>%5.2d rpm</code> <p>gültige Formatangaben nach %:</p> <p>Flags:</p> <p><code>+</code> ...optionale Ausgabe immer mit Vorzeichen <code>0</code> ...optionale Ausgabe mit Vornullen</p> <p>Width: Zahl ...optionale Anzahl Vorkommastellen</p> <p>Prec: Zahl ...optionale Anzahl Nachkommastellen</p> <p>Type:</p> <p><code>b</code> ...Binärdarstellung (Wort) <code>lb</code> ...Binärdarstellung (Doppelwort) <code>d</code> ...Datenformat Integer (Wort) <code>ld</code> ...Doppelte Genauigkeit (Doppelwort) <code>u</code> ...Datenformat Unsigned Integer (Wort) <code>lu</code> ...Datenformat Unsigned Integer (Doppelwort) <code>e</code> ...Exponentialdarstellung <code>f</code> ...Gleitkommadarstellung <code>x</code> ...Hexadezimale Darstellung (Wort) <code>lx</code> ...Hexadezimale Darstellung (Doppelwort) <code>c</code> ...Darstellung als Character <code>s</code> ...String (ohne Längenangabe 80 Zeichen)</p> <p>Zeit/Datumsdarstellung englisch (VarType TIME)</p> <p><code>%a</code> ...abgekürzter Name des Wochentages <code>%A</code> ...voller Name des Wochentages <code>%b</code> ...abgekürzter Name des Monats <code>%B</code> ...voller Name des Monats (englisch) <code>%c</code> ...lokale Darstellung von Datum und Zeit <code>%d</code> ...Tag des Monats (01-31) <code>%H</code> ...Stunde (00-23) <code>%I</code> ...Stunde (01-12) <code>%j</code> ...Tag im Jahr (001-366) <code>%m</code> ...Monat (01-12) <code>%M</code> ...Minute (00-59) <code>%p</code> ...lokales Äquivalent von AM oder PM <code>%S</code> ...Sekunde (00-59) <code>%U</code> ...Woche im Jahr (00-53) (Sonntag ist erster Wochentag) <code>%w</code> ...Wochentag (0-6) (Sonntag ist 0) <code>%W</code> ...Woche im Jahr (00-53) (Montag ist erster Wochentag) <code>%x</code> ...lokale Darstellung des Datums <code>%X</code> ...lokale Darstellung der Zeit <code>%y</code> ...Jahr ohne Jahrhundert (00-99) <code>%Y</code> ...Jahr mit Jahrhundert <code>%Z</code> ...Name der Zeitzone falls existent (hardwareabhängig)</p>

	Text %[HHH:MM:SS:MSMSMS]T Text	<ul style="list-style-type: none"> • Zeitdarstellung (IEC_TIME) Bsp. Zyklus %[M:SS:MSMSMS]T HHH ...Stunden mit 3 Stellen 0-9 MM ...Minuten mit 2 Stellen 0-9 SS ...Sekunden mit 2 Stellen 0-9 MSMSMS ...Millisekunden mit 3 Stellen 0-9 : ...Trenner • Eingabe / Ausgaben werden direkt umgerechnet • Konstanten für Limits werden im Format von CoDeSys angegeben T#10H5M2S100MS
	Text %[dd.mm.YYYY HH:MM:SS]DT Text	<ul style="list-style-type: none"> • Zeit und Datumsdarstellung (VarType IEC_DT) Bsp. %[dd.mm.YYYY HH:MM:SS]DT dd ...Tag mit 2 Stellen mm ...Monat mit 2 Stellen YYYY ...Jahr mit 4 Stellen HH ...Stunden mit 2 Stellen MM ...Minuten mit 2 Stellen SS ...Sekunden mit 2 Stellen : ...beliebiger Trenner • Eingabe / Ausgaben werden direkt umgerechnet • Konstanten für Limits werden im folgenden Format angegeben DT#04d04m2003Y12H30M03S
Font	Font.FNT	<ul style="list-style-type: none"> • optionaler Font für den Text
X,Y,DX,DY	Integerwerte	<ul style="list-style-type: none"> • Position, Breite und Höhe der Variable (Bezug links oben!)
Color	Farbname oder Farbnummer	<ul style="list-style-type: none"> • Farbe der Variable
Backcolor	Transparent	<ul style="list-style-type: none"> • Keine Hintergrundfarbe (aktueller Hintergrund)
	Farbname oder Farbnummer	<ul style="list-style-type: none"> • Hintergrundfarbe der Variable
Format		<ul style="list-style-type: none"> • ohne Eingabe kein Rand
	Border=Button	<ul style="list-style-type: none"> • Umrandungstyp Button
	Border=Input	<ul style="list-style-type: none"> • Umrandungstyp Eingabefeld
	Border=Rx	<ul style="list-style-type: none"> • Umrandungstyp Rechteck mit Breite x Pixel (1, 3, 5, etc.)
	Border=Shadow	<ul style="list-style-type: none"> • Umrandungstyp Rechteck mit Schatten (3D Effekt)
	Border=Signal	<ul style="list-style-type: none"> • Umrandungstyp Anzeigefeld
	Invisible	<ul style="list-style-type: none"> • Objekt unsichtbar
Action	#Page=Name	<ul style="list-style-type: none"> • Bildseitenwechsel zur Eingabe-Bildseite <i>Name</i>
	#Page=Name% <i>s_language</i> %	<ul style="list-style-type: none"> • Bildseitenwechsel zur Eingabe-Bildseite <i>Name+Wert der Systemvariablen s_xy vom Typ STRING</i>. Der Name der Bildseite wird dynamisch generiert. Damit können länderspezifische Tastaturlayouts realisiert werden.
	Edit	<ul style="list-style-type: none"> • Eine durch Option Set_Focus in den Eingabemodus gesetzte Variable lässt sich direkt editieren
	SetVar=x	<ul style="list-style-type: none"> • Variablenwert auf x setzen (Strings müssen mit einschliessenden, einfachen Hochkomma z.B. 'String' definiert werden)
	SetVar+x	<ul style="list-style-type: none"> • Variablenwert um x inkrementieren
	SetVar-x	<ul style="list-style-type: none"> • Variablenwert um x dekrementieren
	SetVar=NotVar	<ul style="list-style-type: none"> • Variablenwert invertieren (0/1)
Limit1	Wert/Variablenname	<ul style="list-style-type: none"> • Unterer Grenzwert Wert, PLC-Variable bzw. Systemvariable für Grenzwert
Limit2	Wert/Variablenname	<ul style="list-style-type: none"> • Oberer Grenzwert Wert, PLC-Variable bzw. Systemvariable für Grenzwert
Action Limit1 Action Limit2	#Page=Name	<ul style="list-style-type: none"> • Bildseitenwechsel zur Bildseite <i>Name</i>
	Alarm=x	<ul style="list-style-type: none"> • Alarm x auslösen (x ist Alarmnummer)
	Backcolor=x	<ul style="list-style-type: none"> • Hintergrundfarbe auf x ändern

	Backlight=x	<ul style="list-style-type: none"> Hintergrundbeleuchtung setzen (0-100%)
	Close	<ul style="list-style-type: none"> Bildseite (Fenster) schliessen
	Close=Name	<ul style="list-style-type: none"> Bildseite (Fenster) <i>Name</i> schliessen
	Color=x	<ul style="list-style-type: none"> Farbe auf x ändern
	Contrast=x	<ul style="list-style-type: none"> Kontrast setzen (0-100%) (nur passiv-LCD)
	Exit	<ul style="list-style-type: none"> Programm verlassen (zurück zum Betriebssystem)
	FastFlash	<ul style="list-style-type: none"> Objektstatus auf Blinken mit 2 Hz setzen
	Flash	<ul style="list-style-type: none"> Objektstatus auf Blinken mit 1 Hz setzen
	Language=name	<ul style="list-style-type: none"> Online Sprachumschaltung zur Sprache <i>name</i>
	Language=s_msysvar	<ul style="list-style-type: none"> Online Sprachumschaltung zur Sprache in <i>s_msysvar</i>
	Load=x	<ul style="list-style-type: none"> Rezeptdatei mit Namen x laden
	Msg=x	<ul style="list-style-type: none"> Meldung mit Nummer x ausgeben
	PWL=x	<ul style="list-style-type: none"> PWL auf x setzen
	SetVar=x	<ul style="list-style-type: none"> Variablenwert auf x setzen
	SetVar=Limit1	<ul style="list-style-type: none"> Variablenwert auf Limit1 setzen
	SetVar=Limit2	<ul style="list-style-type: none"> Variablenwert auf Limit2 setzen
	s_msysvar=x	<ul style="list-style-type: none"> Systemvariablenwert auf x setzen
	Unit=x	<ul style="list-style-type: none"> Einheiten-Umschaltung
VarValue	<i>Variablenname</i>	<ul style="list-style-type: none"> PLC-Variable bzw. Systemvariable
VarType	BOOL	<ul style="list-style-type: none"> Datentyp Bool (8 Bit)
	BYTE	<ul style="list-style-type: none"> Datentyp Byte (8 Bit)
	DINT	<ul style="list-style-type: none"> Datentyp Doppelinteger (32 Bit)
	DWORD	<ul style="list-style-type: none"> Datentyp Doppelwort (32 Bit)
	IEC_DT	<ul style="list-style-type: none"> Datentyp IEC_DT (32 Bit)
	IEC_TIME	<ul style="list-style-type: none"> Datentyp IEC_TIME (32 Bit)
	INT	<ul style="list-style-type: none"> Datentyp Integer (16 Bit)
	REAL	<ul style="list-style-type: none"> Datentyp Gleitkomma (32 Bit)
	SINT	<ul style="list-style-type: none"> Datentyp Shortinteger (8 Bit)
	STRING	<ul style="list-style-type: none"> Datentyp String 80 Byte (ohne Angabe: default 80 Byte)
	STRING:xx	<ul style="list-style-type: none"> Datentyp String xx Byte
	TIME	<ul style="list-style-type: none"> Datentyp Time, Darstellung von Uhrzeit-/Datumsvariablen
	UDINT	<ul style="list-style-type: none"> Datentyp Doppelinteger ohne Vorzeichen (32 Bit)
	UINT	<ul style="list-style-type: none"> Datentyp Integer ohne Vorzeichen (16 Bit)
	USINT	<ul style="list-style-type: none"> Datentyp Shortinteger ohne Vorzeichen (8 Bit)
WORD	<ul style="list-style-type: none"> Datentyp Wort (16 Bit) 	
VarState	<i>Variablenname</i>	<ul style="list-style-type: none"> PLC-Variable bzw. Systemvariable für Objektstatus
Option	CloseIfOk	<ul style="list-style-type: none"> Führt einen Close-Befehl aus, wenn der Variablenwert gültig ist (z.B. zum Schliessen der Bildschirm-Tastatur bei korrekter Eingabe)
	HelpText=x	<ul style="list-style-type: none"> Beim Start der Eingabe wird die Systemvariable <i>s_helptext</i> auf den Wert x gesetzt (siehe auch Systemvariablen)
	Pos=Center	<ul style="list-style-type: none"> Variable zentriert
	Pos=Left	<ul style="list-style-type: none"> Variable linksbündig
	Pos=Right	<ul style="list-style-type: none"> Variable rechtsbündig
	PWL=x	<ul style="list-style-type: none"> notwendiger Passwortlevel zur Freigabe
	Scroll	<ul style="list-style-type: none"> Position des Objektes kann mit Scroll-Objekt geändert werden
	Set_Focus	<ul style="list-style-type: none"> Setzt die Variable in den Eingabemodus (nur für eine Variable pro Bildseite möglich)
	Type=Password	<ul style="list-style-type: none"> Ein- bzw. Ausgabe der Variable erfolgt nicht in Klartext sondern mit dem Charakter „*“ (Findet Verwendung bei der Eingabe und Ausgabe von Passwörtern)
C-Function	<i>C-Funktionsname</i>	<ul style="list-style-type: none"> Name der C-Funktion (siehe Einbindung von C-Funktionen)
	<i>Unit-Umrechnungsfunktionen</i>	<ul style="list-style-type: none"> Vordefinierte Umrechnungsfunktion oder frei definierbarer

		Faktor z.B. [,mm_inch] oder [,*2.54] • [[Funktion-Unit0],Funktion-Unit1]
--	--	---

**Ganzzahlwerte mit Dezimalpunkt**

Es ist möglich Ganzzahlwerte auf dem Bildschirm mit Dezimalpunkt darzustellen. Ein Wert 1000 Gramm kann auf dem Bildschirm in der Einheit kg 1.000 dargestellt und auch eingegeben werden. Dadurch kann in den meisten Fällen PLC-seitig auf eine aufwendige Gleitkomma-Arithmetik verzichtet werden (Geschwindigkeit!).

**Zeit/Datumsfunktionen**

Der Variablentyp Time ermöglicht eine Zeit/Datumsanzeige. Die aktuelle Uhrzeit wird dabei einmal pro Sekunde aktualisiert. Für die Eingabe von Uhrzeit und Datum stehen spezielle Systemvariablen zur Verfügung (siehe auch Systemvariablen).

**Aktuelle Uhrzeit auf der PLC**

Das Objekt Variable unterstützt den Download der Uhrzeit auf die PLC (einmal pro Sekunde). Hierzu muss dem Objekt eine PLC-Variable mit folgender Struktur zugeordnet werden:

Globale Variable:

```
VAR_GLOBAL
    EpamTime      :      EpamTimeType;
END_VAR
```

PLC-Datentyp:

```
TYPE EpamTimeType:
    STRUCT
        tm_time      :      DT;          (* Time since 1. January 1970, 00:00:00 *)
        tm_sec       :      WORD;       (* Seconds after the minute - [0,59] *)
        tm_min       :      WORD;       (* Minutes after the hour - [0,59] *)
        tm_hour      :      WORD;       (* Hours since midnight - [0,23] *)
        tm_day       :      WORD;       (* Day of the month - [1,31] *)
        tm_mon       :      WORD;       (* Months of the year [1,12] *)
        tm_year      :      WORD;       (* Year since 1900 *)
        tm_wday      :      WORD;       (* Days since Sunday - [0,6] *)
        tm_yday      :      WORD;       (* Days since 1. January - [0,365] *)
        tm_isdst     :      WORD;       (* Daylight-saving-time flag 0 = off, 1 = on, -1 = not avail *)
    END_STRUCT
END_TYPE
```

**Systemvariablen**

Nachfolgende Systemvariablen werden beim Start der Eingabe mit den aktuellen Werten definiert und können z.B. auf der Bildschirmtastatur-Seite angezeigt werden:

s_edit_val	...letzter Wert vor Eingabe Datentyp STRING
s_input_val	...aktueller Wert der Eingabe Datentyp STRING
s_limit1	...unterer Grenzwert Datentyp STRING
s_limit2	...oberer Grenzwert Datentyp STRING
s_helptext	...enthält die aktuelle Helptextnummer Datentyp WORD

**Bildschirmtastatur**

Wird eine Bildschirmtastatur über die Variablen-Aktion „#Page=name“ aufgerufen, so muss diese als Fenster geöffnet werden! D.h. die Bildseite des Variablen-Objekts und der Tastatur müssen verschieden gross sein.

7.7 Objekt Bar

#Bar		Darstellung eines Wertes in einem rechteckigen Balken
Text/File		
Font		
X,Y,DX,DY	<i>Integerwerte</i>	<ul style="list-style-type: none"> Position, Breite und Höhe des Balkens (Bezug links oben!)
Color	<i>Farbname oder Farbnummer</i>	<ul style="list-style-type: none"> Farbe des Balkens (Füllfarbe)
BackColor	<i>Farbname oder Farbnummer</i>	<ul style="list-style-type: none"> Hintergrundfarbe des Balkens (Löscharbe)
Format		<ul style="list-style-type: none"> ohne Eingabe kein Rand
	Border=Button	<ul style="list-style-type: none"> Umrandungstyp Button
	Border=Input	<ul style="list-style-type: none"> Umrandungstyp Eingabefeld
	Border=Rx	<ul style="list-style-type: none"> Umrandungstyp Rechteck mit Breite x Pixel (1, 3, 5, etc.)
	Border=Shadow	<ul style="list-style-type: none"> Umrandungstyp Rechteck mit Schatten (3D Effekt)
Action	SetVar=x	<ul style="list-style-type: none"> Variablenwert auf x
	SetVar+x	<ul style="list-style-type: none"> Variablenwert um x inkrementieren
	SetVar-x	<ul style="list-style-type: none"> Variablenwert um x dekrementieren
Limit1	<i>Wert/Variablenname</i>	<ul style="list-style-type: none"> Unterer Grenzwert Wert, PLC-Variable bzw. Systemvariable für Grenzwert
Limit2	<i>Wert/Variablenname</i>	<ul style="list-style-type: none"> Oberer Grenzwert Wert, PLC-Variable bzw. Systemvariable für Grenzwert
Action Limit1 Action Limit2	#Page=Name	<ul style="list-style-type: none"> Bildseitenwechsel zur Bildseite <i>Name</i>
	Alarm=x	<ul style="list-style-type: none"> Alarm x auslösen (x ist Alarmnummer)
	BackColor=x	<ul style="list-style-type: none"> Hintergrundfarbe auf x ändern
	Backlight=x	<ul style="list-style-type: none"> Hintergrundbeleuchtung setzen (0-100%)
	Close	<ul style="list-style-type: none"> Bildseite (Fenster) schliessen
	Close=Name	<ul style="list-style-type: none"> Bildseite (Fenster) <i>Name</i> schliessen
	Color=x	<ul style="list-style-type: none"> Farbe auf x ändern
	Contrast=x	<ul style="list-style-type: none"> Kontrast setzen (0-100%) (nur passiv-LCD)
	Exit	<ul style="list-style-type: none"> Programm verlassen (zurück zum Betriebssystem)
	FastFlash	<ul style="list-style-type: none"> Objektstatus auf Blinken mit 2 Hz setzen
	Flash	<ul style="list-style-type: none"> Objektstatus auf Blinken mit 1 Hz setzen
	Language=name	<ul style="list-style-type: none"> Online Sprachumschaltung zur Sprache <i>name</i>
	Language=s_msysvar	<ul style="list-style-type: none"> Online Sprachumschaltung zur Sprache in <i>s_msysvar</i>
	Load=x	<ul style="list-style-type: none"> Rezeptdatei mit Namen x laden
	Msg=x	<ul style="list-style-type: none"> Meldung mit Nummer x ausgeben
	PWL=x	<ul style="list-style-type: none"> PWL auf x setzen
	SetVar=x	<ul style="list-style-type: none"> Variablenwert auf x setzen
	SetVar=Limit1	<ul style="list-style-type: none"> Variablenwert auf Limit1 setzen
SetVar=Limit2	<ul style="list-style-type: none"> Variablenwert auf Limit2 setzen 	
s_msysvar=x	<ul style="list-style-type: none"> Systemvariablenwert auf x setzen 	
VarValue	<i>Variablenname</i>	<ul style="list-style-type: none"> PLC-Variable bzw. Systemvariable
VarType	BYTE	<ul style="list-style-type: none"> Datentyp Byte (8 Bit)
	DINT	<ul style="list-style-type: none"> Datentyp Doppelinteger (32 Bit)
	DWORD	<ul style="list-style-type: none"> Datentyp Doppelwort (32 Bit)
	INT	<ul style="list-style-type: none"> Datentyp Integer (16 Bit)
	REAL	<ul style="list-style-type: none"> Datentyp Gleitkomma (32 Bit)
	SINT	<ul style="list-style-type: none"> Datentyp Shortinteger (8 Bit)
	UDINT	<ul style="list-style-type: none"> Datentyp Doppelinteger ohne Vorzeichen (32 Bit)
	UINT	<ul style="list-style-type: none"> Datentyp Integer ohne Vorzeichen (16 Bit)
	USINT	<ul style="list-style-type: none"> Datentyp Shortinteger ohne Vorzeichen (8 Bit)
	WORD	<ul style="list-style-type: none"> Datentyp Wort (16 Bit)

VarState	<i>Variablenname</i>	<ul style="list-style-type: none"> • PLC-Variable bzw. Systemvariable für Objektstatus
Option	Fill=Down	<ul style="list-style-type: none"> • Füllrichtung von oben nach unten
	Fill=Left	<ul style="list-style-type: none"> • Füllrichtung von rechts nach links
	Fill=Right	<ul style="list-style-type: none"> • Füllrichtung von links nach rechts
	Fill=Up	<ul style="list-style-type: none"> • Füllrichtung von unten nach oben
	Fill=x	<ul style="list-style-type: none"> • Füllrichtung links und rechts (Balkenmitte = $(\text{Limit1} + \text{Limit2}) / 2$)
	Fill=y	<ul style="list-style-type: none"> • Füllrichtung unten und oben (Balkenmitte = $(\text{Limit1} + \text{Limit2}) / 2$)
	PWL=x	<ul style="list-style-type: none"> • notwendiger Passwortlevel zur Freigabe
	Scroll	<ul style="list-style-type: none"> • Position des Objektes kann mit Scroll-Objekt geändert werden
C-Function	<i>C-Funktionsname</i>	<ul style="list-style-type: none"> • Name der C-Funktion (siehe Einbindung von C-Funktionen)



Die Limit-Aktion (z.B. Farbumschlag) wird bei Über- bzw. Unterschreitung der Grenzwerte ausgeführt. Eine Limit Aktion z.B. bei 80% des Wertes ist nicht möglich. Hierfür können zwei Barobjekte definiert werden (s.a. Demo „überlagerte Objekte“)

7.8 Objekt Signal

#Signal		Darstellung von Zuständen bzw. statischen Bildern und Texten
Text/File	<i>Bild0.PCX,Bild1.PCX,...</i>	<ul style="list-style-type: none"> Name von PCX-Bilddateien für Zustände 0,1,... durch ',' getrennt
	<i>Icon0.ICO,Icon1.ICO,...</i>	<ul style="list-style-type: none"> Name von PCX-Bilddateien mit der Dateierweiterung .ICO für Zustände 0,1,... durch ',' getrennt werden als Icons zentriert dargestellt
	<i>Text0,Text1,...</i>	<ul style="list-style-type: none"> Textstrings für Zustände 0,1,... durch ',' getrennt Text kann mit Zeilentrenner ' ' (ASCII-Zeichen 124 bzw. 7CH) auf mehrere Zeilen aufgeteilt werden
		<ul style="list-style-type: none"> wird kein Text eingegeben, so wird das Signal ohne Text ausgegeben
Font	<i>Font.FNT</i>	<ul style="list-style-type: none"> optionaler Font für den Text
X,Y,DX,DY	<i>Integerwerte</i>	<ul style="list-style-type: none"> Position, Breite und Höhe des Signals (Bezug links oben!)
Color	<i>Farbname oder Farbnummer Color0,Color1,...</i>	<ul style="list-style-type: none"> Farbe des Textes für Zustände 0,1,...
Backcolor	Transparent	<ul style="list-style-type: none"> Keine Hintergrundfarbe (aktueller Hintergrund)
	<i>Farbname oder Farbnummer BackColor0,BackColor1,...</i>	<ul style="list-style-type: none"> Hintergrundfarbe des Signales für Zustände 0,1,...
Format		<ul style="list-style-type: none"> ohne Eingabe kein Rand
	Border=Button	<ul style="list-style-type: none"> Umrandungstyp Button
	Border=Input	<ul style="list-style-type: none"> Umrandungstyp Eingabefeld
	Border=Rx	<ul style="list-style-type: none"> Umrandungstyp Rechteck mit Breite x Pixel (1, 3, 5, etc.)
	Border=Shadow	<ul style="list-style-type: none"> Umrandungstyp Rechteck mit Schatten (3D Effekt)
	Border=Signal	<ul style="list-style-type: none"> Umrandungstyp Anzeigefeld
	Frame=x	<ul style="list-style-type: none"> Umrandungstyp Rahmen mit Rahmenstärke x Pixel (1, 3, 5, etc.) und der jeweiligen Hintergrundfarbe des Signales für Zustände 0,1,... Der Bereich innerhalb des Rechtecks wird nicht überschrieben
Action		<ul style="list-style-type: none"> Nur Anzeige, keine Eingaben
Limit1	<i>Wert/Variablenname</i>	<ul style="list-style-type: none"> Unterer Grenzwert Wert, PLC-Variable bzw. Systemvariable für Grenzwert
Limit2	<i>Wert/Variablenname</i>	<ul style="list-style-type: none"> Oberer Grenzwert Wert, PLC-Variable bzw. Systemvariable für Grenzwert
Action Limit1 Action Limit2	#Page=Name	<ul style="list-style-type: none"> Bildseitenwechsel zur Bildseite <i>Name</i>
	Alarm=x	<ul style="list-style-type: none"> Alarm x auslösen (x ist Alarmnummer)
	Backcolor=x	<ul style="list-style-type: none"> Hintergrundfarbe auf x ändern
	Backlight=x	<ul style="list-style-type: none"> Hintergrundbeleuchtung setzen (0-100%)
	Close	<ul style="list-style-type: none"> Bildseite (Fenster) schliessen
	Close=Name	<ul style="list-style-type: none"> Bildseite (Fenster) <i>Name</i> schliessen
	Color=x	<ul style="list-style-type: none"> Farbe auf x ändern
	Contrast=x	<ul style="list-style-type: none"> Kontrast setzen (0-100%) (nur passiv-LCD)
	Exit	<ul style="list-style-type: none"> Programm verlassen (zurück zum Betriebssystem)
	FastFlash	<ul style="list-style-type: none"> Objektstatus auf Blinken mit 2 Hz setzen
	Flash	<ul style="list-style-type: none"> Objektstatus auf Blinken mit 1 Hz setzen
	Language=name	<ul style="list-style-type: none"> Online Sprachumschaltung zur Sprache <i>name</i>
	Language=s_msysvar	<ul style="list-style-type: none"> Online Sprachumschaltung zur Sprache in s_msysvar
	Load=x	<ul style="list-style-type: none"> Rezeptdatei mit Namen x laden
	Msg=x	<ul style="list-style-type: none"> Meldung mit Nummer x ausgeben
PWL=x	<ul style="list-style-type: none"> PWL auf x setzen 	
SetVar=x	<ul style="list-style-type: none"> Variablenwert auf x setzen 	

	SetVar=Limit1	• Variablenwert auf Limit1 setzen
	SetVar=Limit2	• Variablenwert auf Limit2 setzen
	s_mysysvar=x	• Systemvariablenwert auf x setzen
VarValue	<i>Variablenname</i>	• PLC-Variable bzw. Systemvariable
VarType	BOOL	• Datentyp Bool (8 Bit)
	BYTE	• Datentyp Byte (8 Bit)
	INT	• Datentyp Integer (16 Bit)
	REAL	• Datentyp Gleitkomma (32 Bit)
	SINT	• Datentyp Shortinteger (8 Bit)
	UINT	• Datentyp Integer ohne Vorzeichen (16 Bit)
	USINT	• Datentyp Shortinteger ohne Vorzeichen (8 Bit)
	WORD	• Datentyp Wort (16 Bit)
VarState	<i>Variablenname</i>	• PLC-Variable bzw. Systemvariable für Objektstatus
Option		• Default: Signal zentriert
	Animation-Delay=x	• Animation von Zustand 1 bis n mit Zeitdelay von x/10s, wenn Wert > 0, sonst Zustand 0
	Pos=Center	• Signal zentriert
	Pos=Left	• Signal linksbündig
	Pos=Right	• Signal rechtsbündig
	Scroll	• Position des Objektes kann mit Scroll-Objekt geändert werden
	Switch=x	• x = Wertebereich für Zustände 0,1,... z.B.: <0:1:2..5:>5 Erlaubte Werte: Konstante Zahl z.B. 5 <Zahl ...kleiner als >Zahl ...grösser als Zahl..Zahl ...Bereich von bis : ...Seperator
Transparency=Farbname	• Farbname, welcher transparent interpretiert wird (in Verbindung mit einer PCX-Bilddatei)	
C-Function	<i>C-Funktionsname</i>	• Name der C-Funktion (siehe Einbindung von C-Funktionen)
	Unit-Umrechnungsfunktionen	• Vordefinierte Umrechnungsfunktion oder frei definierbarer Faktor z.B. [,mm_inch] oder [,*2.54] • [[Funktion-Unit0],Funktion-Unit1]



Bildseitenwechsel in der PLC auslösen

Mit Hilfe des Signal-Objektes können Sie durch Definition von Grenzwerten (Limit1/2) und der Aktion '#Page=name' auch Bildseitenwechsel aufgrund von Variablenwerten auslösen. In diesem Fall bleibt die Spalte Text/File leer. Um diese Bildseitenwechsel an jeder beliebigen Stelle auszuführen, muss das Objekt global definiert werden (siehe auch globale Objekte).



Signalzustand

Es können mehrere Signalzustände durch Komma getrennt definiert werden. In diesem Fall wird abhängig vom aktuellen Variablenwert (0, 1, 2, ...) der entsprechende Text, bzw. die PCX-Bilddatei dargestellt.

Beispiel:

PCX-Bilddateien Icon0.ico,Icon1.ico,Icon2.ico

...Wechsel auf PCX-Bilddatei „Icon0.ico“, wenn der Variablenwert 0 ist

...Wechsel auf PCX-Bilddatei „Icon1.ico“, wenn der Variablenwert 1 ist

...Wechsel auf PCX-Bilddatei „Icon2.ico“, wenn der Variablenwert 2 ist

Mit Hilfe der **Option ,switch='** können auch Wertebereiche für die einzelnen Zustände definiert werden.

Beispiel:

PCX-Bilddateien Icon0.ico,Icon1.ico,Icon2.ico mit Option:switch=<0:0..5:>5

...Wechsel auf PCX-Bilddatei „Icon0.ico“, wenn der Variablenwert < 0 ist

...Wechsel auf PCX-Bilddatei „Icon1.ico“, wenn der Variablenwert im Bereich 0 bis 5 liegt

...Wechsel auf PCX-Bilddatei „Icon2.ico“, wenn der Variablenwert > 5 ist

Wichtig!

Bei Bildern (*.ICO bzw. *.PCX) dürfen keine Leerzeichen zwischen den `,' angegeben werden, da diese sonst als Dateiname interpretiert werden!

Um einen Farbumschlag bei gleichem Text zu erreichen, muss der Text mehrfach definiert werden.

7.9 Objekt Message

#Message		Ausgabe von Meldungen in Form von Text- oder Bild-Information
Text/File	<i>Meldung.TXT</i>	<ul style="list-style-type: none"> Name der Meldungs-Definitionsdatei mit den Meldungsnummer und den Meldungstexten
Font	<i>Font.FNT</i>	<ul style="list-style-type: none"> optionaler Font für den Text
X,Y,DX,DY	<i>Integerwerte</i>	<ul style="list-style-type: none"> Position, Breite und Höhe des Signals (Bezug links oben!)
Color	<i>Farbname oder Farbnummer</i>	<ul style="list-style-type: none"> Farbe des Textes der Meldungsfensters
BackColor	<i>Farbname oder Farbnummer</i>	<ul style="list-style-type: none"> Hintergrundfarbe des Meldungsfensters
Format		<ul style="list-style-type: none"> ohne Eingabe kein Rand
	Border=Button	<ul style="list-style-type: none"> Umrandungstyp Button
	Border=Input	<ul style="list-style-type: none"> Umrandungstyp Eingabefeld
	Border=Rx	<ul style="list-style-type: none"> Umrandungstyp Rechteck mit Breite x Pixel (1, 3, 5, etc.)
	Border=Shadow	<ul style="list-style-type: none"> Umrandungstyp Rechteck mit Schatten (3D Effekt)
	Border=Signal	<ul style="list-style-type: none"> Umrandungstyp Anzeigefeld
Action	SetVar=x	<ul style="list-style-type: none"> Variablenwert auf x setzen
Limit1		<ul style="list-style-type: none">
Limit2		<ul style="list-style-type: none">
Action Limit1 Action Limit2		<ul style="list-style-type: none">
VarValue	<i>Variablenname</i>	<ul style="list-style-type: none"> PLC-Variable bzw. Systemvariable
VarType	SINT	<ul style="list-style-type: none"> Datentyp Shortinteger (8 Bit)
	INT	<ul style="list-style-type: none"> Datentyp Integer (16 Bit)
	STRING:xx	<ul style="list-style-type: none"> Datentyp String xx Byte
	UINT	<ul style="list-style-type: none"> Datentyp Integer ohne Vorzeichen (16 Bit)
	WORD	<ul style="list-style-type: none"> Datentyp Wort (16 Bit)
VarState	<i>Variablenname</i>	<ul style="list-style-type: none"> PLC-Variable bzw. Systemvariable für Objektstatus
Option		<ul style="list-style-type: none"> Default: Meldung linksbündig
	Pos=Center	<ul style="list-style-type: none"> Meldung zentriert
	Pos=Left	<ul style="list-style-type: none"> Meldung linksbündig
	Pos=Right	<ul style="list-style-type: none"> Meldung rechtsbündig
	PWL=x	<ul style="list-style-type: none"> notwendiger Passwortlevel zur Freigabe
	Scroll	<ul style="list-style-type: none"> Position des Objektes kann mit Scroll-Objekt geändert werden
	Transparency=Farbname	<ul style="list-style-type: none"> Farbname, welcher transparent interpretiert wird (in Verbindung mit einer PCX-Bilddatei)
C-Function	<i>C-Funktionsname</i>	<ul style="list-style-type: none"> Name der C-Funktion (siehe Einbindung von C-Funktionen)



Meldungstexte mit Variablenwerten

Ein Meldungstext kann zusätzliche Formatangaben (z.B. %d, siehe auch Objekt Variable) beinhalten. In diesem Fall muss der Variablentyp (VarType) als Stringtyp mit der Länge der Meldungsnummer (2 Byte) und der Länge des variablen Parameters definiert werden (bei %d nochmals 2 Byte, d.h. in diesem Fall mit String:4). Die Länge des Strings richtet sich nach der grössten Formatangabe in der Meldungsdatei. Beim Aufruf einer Meldung mit Parametern in der Steuerung, müssen die variablen Argumente entsprechend der Formatangabe vorher gesetzt werden.

Beispiel: Zusätzlich zur jeweiligen Meldungsnummer werden der Meldung noch Variablenwerte übergeben (INT,REAL)

Globale Variable:

```
VAR_GLOBAL
    VisuMessage      AT %MB200 :    ARRAY[0..5] OF BYTE
    MessageNumber    AT %MW200 :    INT;
    MessageReal      AT %MD202 :    REAL;
    MessageINT       AT %MW202 :    INT;
END_VAR
```

Wichtig!

Wird beim Message-Objekt ein Variablenwert in der Meldung angezeigt, so muss die Meldungsnummer bei allen WindowsCE Geräten als DWORD (32 Bit-Variable) definiert werden!

7.9.1 Tabellenblatt Message

Im Tabellenblatt „Message“ werden den Meldungsnummern die Meldungstexte und die meldungsspezifischen Eigenschaften zugewiesen. Das Tabellenblatt „Message“ ist wie folgt aufgebaut:

Number	Text/File	Font	Color	BackColor
1	Beliebige Meldung 1	ARI16F.FNT	white	yellow
2	Beliebige Meldung 2	ARI20F.FNT	white	red
3	Beliebige Meldung 3 mit Variable %3d	ARI16F.FNT	white	red
4	Beliebige Meldung 4 mit Variable %2.2f	ARI20F.FNT	white	red
...
10	myMsg.PCX			
...

Meldungsspezifische
Font und Farbdefinition



Die Liste der Meldungen muss aufsteigend nach Meldungsnummern sortiert, beginnend bei der niedrigsten Meldungsnummer erstellt werden!



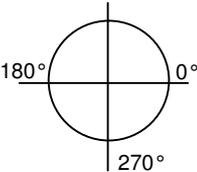
Meldungen von PLC aus löschen:

Definieren Sie hierzu eine Leermeldung ohne Text. In diesem Fall wird nur der Rechteckbereich gelöscht.



Meldungstexte müssen für alle Sprachen definiert werden. Ist kein Text definiert, so wird eine „Leermeldung“ angezeigt. Es erfolgt keine Anzeige des Defaulttext!

7.10 Objekt Meter

#Meter		Darstellung eines Wertes in einem Halb-/Kreis/ benutzerdefinierten Winkel
Text/File	<i>Bild.PCX</i>	<ul style="list-style-type: none"> Name einer PCX-Bilddatei für das Hintergrundbild Der Hintergrund und die Skala werden nicht gezeichnet (nur Zeiger)
	<i>Text</i>	<ul style="list-style-type: none"> Textstring (Der Text wird beim $\frac{3}{4}$-Kreis im leeren $\frac{1}{4}$ und bei den Halbkreisen auf-/abwärts unter/über dem Objekt dargestellt)
Font	<i>Font.FNT</i>	<ul style="list-style-type: none"> optionaler Font für den Text
X,Y,DX,DY	<i>Integerwerte</i>	<ul style="list-style-type: none"> Position, Breite und Höhe des Rechtecks inkl. Skala (Bezug links oben!). Wird die Formatoption Up/Down bzw. Left/Right definiert, so wird das Meterobjekt innerhalb des Rechteckes an der Mittellinie ausgerichtet. D.h. wenn zwei Meterobjekte z.B. Up und Down mit den selben Koordinaten definiert werden, so werden diese an der Mittellinie gespiegelt.
Color	<i>Farbname oder Farbnummer</i>	<ul style="list-style-type: none"> Farbe der Skala und des Zeigers)
BackColor	<i>Farbname oder Farbnummer</i>	<ul style="list-style-type: none"> Hintergrundfarbe (Halb-/Kreisfarbe/Winkelfarbe)
Format 	Angle=Begin,Angle,Scalenum	<ul style="list-style-type: none"> Benutzerdefinierter Winkel Anfangswinkel, Winkelgrösse, Anzahl Skalaabschnitte ...negative Winkelgrösse => Uhrzeigersinn ...positive Winkelgrösse => Gegenuhrzeigersinn) Skalaabschnitte werden in 5 Teilabschnitte unterteilt Darstellung ohne Skala möglich (Scale = 0). Die Skala kann auf eine Kommastelle genau dargestellt werden.
	Circle	<ul style="list-style-type: none"> $\frac{3}{4}$-Kreis (default) (Minimalwert bei 225°)
	Down	<ul style="list-style-type: none"> Halbkreis nach unten (Min. bei 180°)
	Left	<ul style="list-style-type: none"> Halbkreis nach links (Min. bei 270°)
	Right	<ul style="list-style-type: none"> Halbkreis nach rechts (Min. bei 270°)
	Up	<ul style="list-style-type: none"> Halbkreis nach oben (Min. bei 180°)
Action	SetVar=x	<ul style="list-style-type: none"> Variablenwert auf x
	SetVar+x	<ul style="list-style-type: none"> Variablenwert um x inkrementieren
	SetVar-x	<ul style="list-style-type: none"> Variablenwert um x dekrementieren
Limit1	<i>Wert/Variablenname</i>	<ul style="list-style-type: none"> Unterer Grenzwert Wert, PLC-Variable bzw. Systemvariable für Grenzwert
Limit2	<i>Wert/Variablenname</i>	<ul style="list-style-type: none"> Oberer Grenzwert Wert, PLC-Variable bzw. Systemvariable für Grenzwert
Action Limit1 Action Limit2	#Page=Name	<ul style="list-style-type: none"> Bildseitenwechsel zur Bildseite <i>Name</i>
	Alarm=x	<ul style="list-style-type: none"> Alarm x auslösen (x ist Alarmnummer)
	BackColor=x	<ul style="list-style-type: none"> Hintergrundfarbe auf x ändern
	Backlight=x	<ul style="list-style-type: none"> Hintergrundbeleuchtung setzen (0-100%)
	Close	<ul style="list-style-type: none"> Bildseite (Fenster) schliessen
	Close=Name	<ul style="list-style-type: none"> Bildseite (Fenster) <i>Name</i> schliessen
	Color=x	<ul style="list-style-type: none"> Farbe auf x ändern
	Contrast=x	<ul style="list-style-type: none"> Kontrast setzen (0-100%) (nur passiv-LCD)
	Exit	<ul style="list-style-type: none"> Programm verlassen (zurück zum Betriebssystem)
	FastFlash	<ul style="list-style-type: none"> Objektstatus auf Blinken mit 2 Hz setzen
	Flash	<ul style="list-style-type: none"> Objektstatus auf Blinken mit 1 Hz setzen
	Language=name	<ul style="list-style-type: none"> Online Sprachumschaltung zur Sprache <i>name</i>
	Language=s_msysvar	<ul style="list-style-type: none"> Online Sprachumschaltung zur Sprache in <i>s_msysvar</i>
	Load=x	<ul style="list-style-type: none"> Rezeptdatei mit Namen x laden
	Msg=x	<ul style="list-style-type: none"> Meldung mit Nummer x ausgeben
PWL=x	<ul style="list-style-type: none"> PWL auf x setzen 	

	SetVar=x	<ul style="list-style-type: none"> • Variablenwert auf x setzen
	SetVar=Limit1	<ul style="list-style-type: none"> • Variablenwert auf Limit1 setzen
	SetVar=Limit2	<ul style="list-style-type: none"> • Variablenwert auf Limit2 setzen
	s_msysvar=x	<ul style="list-style-type: none"> • Systemvariablenwert auf x setzen
VarValue	<i>Variablenname</i>	<ul style="list-style-type: none"> • PLC-Variable bzw. Systemvariable
VarType	BYTE	<ul style="list-style-type: none"> • Datentyp Byte (8 Bit)
	DINT	<ul style="list-style-type: none"> • Datentyp Doppelinteger (32 Bit)
	DWORD	<ul style="list-style-type: none"> • Datentyp Doppelwort (32 Bit)
	INT	<ul style="list-style-type: none"> • Datentyp Integer (16 Bit)
	REAL	<ul style="list-style-type: none"> • Datentyp Gleitkomma (32 Bit)
	SINT	<ul style="list-style-type: none"> • Datentyp Shortinteger (8 Bit)
	UDINT	<ul style="list-style-type: none"> • Datentyp Doppelinteger ohne Vorzeichen (32 Bit)
	UINT	<ul style="list-style-type: none"> • Datentyp Integer ohne Vorzeichen (16 Bit)
	USINT	<ul style="list-style-type: none"> • Datentyp Shortinteger ohne Vorzeichen (8 Bit)
	WORD	<ul style="list-style-type: none"> • Datentyp Wort (16 Bit)
VarState	<i>Variablenname</i>	<ul style="list-style-type: none"> • PLC-Variable bzw. Systemvariable für Objektstatus
Option	Scroll	<ul style="list-style-type: none"> • Position des Objektes kann mit Scroll-Objekt geändert werden
	Transparency=Farbname	<ul style="list-style-type: none"> • Farbname, welcher transparent interpretiert wird (in Verbindung mit einer PCX-Bilddatei)
C-Function	<i>C-Funktionsname</i>	<ul style="list-style-type: none"> • Name der C-Funktion (siehe Einbindung von C-Funktionen)



Die Limit-Aktion (z.B. Farbumschlag) wird bei Über- bzw. Unterschreitung der Grenzwerte ausgeführt. Eine Limit Aktion z.B. bei 80% des Wertes ist nicht möglich.

7.11 Objekt Textlist

#Textlist		Ausgabe einer ASCII-Textdatei innerhalb eines rechteckigen Bereiches
Text/File		<ul style="list-style-type: none"> Keine-Textdatei (in Verbindung mit einer projektierten Systemvariable)
	<i>Text.TXT</i>	<ul style="list-style-type: none"> Name der ASCII-Textdatei
	s_alarm_txtinfo	<ul style="list-style-type: none"> Variable mit dem Namen der ASCII-Textdatei mit der alarmspezifischen Textinformation
	<i>Text.CSV</i>	<ul style="list-style-type: none"> CSV-Datei in Tabellenform darstellen
Font	<i>Font.FNT</i>	<ul style="list-style-type: none"> optionaler Font für den Text
X,Y,DX,DY	<i>Integerwerte</i>	<ul style="list-style-type: none"> Position, Breite und Höhe der Textliste (Bezug links oben!)
Color	<i>Farbname oder Farbnummer</i>	<ul style="list-style-type: none"> Farbe des Textes der Textliste
BackColor	<i>Farbname oder Farbnummer</i>	<ul style="list-style-type: none"> Hintergrundfarbe der Textliste
Format		<ul style="list-style-type: none"> ohne Eingabe kein Rand
	Border=Button	<ul style="list-style-type: none"> Umrandungstyp Button
	Border=Input	<ul style="list-style-type: none"> Umrandungstyp Eingabefeld
	Border=Rx	<ul style="list-style-type: none"> Umrandungstyp Rechteck mit Breite x Pixel (1, 3, 5, etc.)
	Border=Shadow	<ul style="list-style-type: none"> Umrandungstyp Rechteck mit Schatten (3D Effekt)
	Border=Signal	<ul style="list-style-type: none"> Umrandungstyp Anzeigefeld
Action		<ul style="list-style-type: none">
Limit1		<ul style="list-style-type: none">
Limit2		<ul style="list-style-type: none">
Action Limit1 Action Limit2		<ul style="list-style-type: none">
VarValue	<i>Variablenname</i>	<ul style="list-style-type: none"> Systemvariable STRING-Systemvariable, welche den Dateinamen beinhaltet oder Systemvariable, welche die Einsprungszeilennummer in die Textliste angibt bzw. die aktuelle Zeilennummer speichert
VarType	INT	<ul style="list-style-type: none"> Datentyp Integer (16 Bit)
	STRING	<ul style="list-style-type: none"> Datentyp String 80 Byte (ohne Angabe: default 80 Byte)
	UINT	<ul style="list-style-type: none"> Datentyp Integer ohne Vorzeichen (16 Bit)
	WORD	<ul style="list-style-type: none"> Datentyp Wort (16 Bit)
VarState		<ul style="list-style-type: none">
Option	DX=x	<ul style="list-style-type: none"> Breite des Scrollbars in Pixel (Default: 25 Pixel)
	Format	<ul style="list-style-type: none"> Aktiviert die Darstellung von Fliesstext und formatierter Textdateien
	sbimg=<scrollbar.pcx>	<ul style="list-style-type: none"> Image für kundenspezifischen Scrollbar (Hintergrund)
	thumbimg=<thumb.pcx>	<ul style="list-style-type: none"> Thumb Image (Slider)
C-Function	<i>C-Funktionsname</i>	<ul style="list-style-type: none"> Name der C-Funktion (siehe Einbindung von C-Funktionen)



Darstellung von Fliesstext und formatiertem Text

Wird die Option Format aktiviert, kann mit dem Objekt Textlist auch Fliesstext bzw. formatierter Text dargestellt werden. Der Text wird hierbei entsprechend der gewählten Schriftgröße und Dimension des Textfensters automatisch eingepasst. Der Zeilenumbruch erfolgt zwischen Wörtern (keine Silbentrennung). Ein Zeilenumbruch mit <CRLF> wird hierbei ignoriert.

Zeilenumbruch in Textdatei projektieren:

Darstellung **Fett** in Textdatei projektieren:

myText

Darstellung Unterstrichen in Textdatei projektieren:

<u>myText</u>

Darstellung *Kursiv* in Textdatei projektieren:

<i>myText</i>



Darstellung unterschiedlicher Textdateien in derselben Textliste

Wird mit Hilfe einer Aktion SetVar='MyTextFile.txt' sowie einer Systemvariable eine Textliste projiziert, können in derselben Textliste unterschiedliche Textdateien dargestellt werden.

Beispiel:

Definition der Textliste in der Bildseite *Help*:

Object	Text/File	Font	X	Y	DX	DY	Color	Back-color	Format	Action	Action Limit1	Action Limit2	Var-Value	Var-Type	Var-State
#Page=Help			10	10	300	220	black	grey							
#Textlist		C:\UR16F.FNT	5	50	290	165	black	grey					S_Helpfile	STRING	

Bildseitenwechsel zur Bildseite *Help* und setzen der Variable s_helpfile mit help1.txt:

Object	Text/File	Font	X	Y	DX	DY	Color	Back-color	Format	Action	Action Limit1	Action Limit2	Var-Value	Var-Type	Var-State
#Page=Seite1			10	10	300	220	black	grey							
#Button	Help.ico		200	40	40		black	grey		setVar='help1.txt' & #Page=Help			S_Helpfile	STRING	

Bildseitenwechsel zur Bildseite *Help* und setzen der Variable s_helpfile mit help2.txt:

Object	Text/File	Font	X	Y	DX	DY	Color	Back-color	Format	Action	Action Limit1	Action Limit2	Var-Value	Var-Type	Var-State
#Page=Seite2			10	10	300	220	black	grey							
#Button	Help.ico		200	40	40		black	grey		setVar='help2.txt' & #Page=Help			S_Helpfile	STRING	



Darstellung von CSV-Datei in Tabellenform

In der CSV-Datei (TAB-getrennt) kann in der ersten Zeile ein Format-Header angegeben werden. Ohne diesen Header wird die CSV-Datei mit gleich breiten Spalten und linksbündigem Text dargestellt.

```
#header width=100 pos=center \t width=50 pos=right \t width=70 pos=left
```

7.12 Objekt **HTMLBrowser**

#HTMLBrowser		Ausgabe einer HTML-Datei innerhalb eines rechteckigen Bereiches (Browser-Fenster)
Text/File	URL	<ul style="list-style-type: none"> • URL einer HTML-Seite lokal oder im Intranet/Internet
Font		<ul style="list-style-type: none"> •
X,Y,DX,DY	<i>Integerwerte</i>	<ul style="list-style-type: none"> • Position, Breite und Höhe des Browser-Fenster (Bezug links oben!)
Color		<ul style="list-style-type: none"> •
BackColor		<ul style="list-style-type: none"> •
Format		<ul style="list-style-type: none"> •
Action		<ul style="list-style-type: none"> •
Limit1		<ul style="list-style-type: none"> •
Limit2		<ul style="list-style-type: none"> •
Action Limit1 Action Limit2		<ul style="list-style-type: none"> •
VarValue	<i>Variablenname</i>	<ul style="list-style-type: none"> • PLC oder Systemvariable welche die URL beinhaltet
VarType	STRING	<ul style="list-style-type: none"> • Datentyp String 80 Byte (ohne Angabe: default 80 Byte)
VarState		<ul style="list-style-type: none"> •
Option		<ul style="list-style-type: none"> •
C-Function	<i>C-Funktionsname</i>	<ul style="list-style-type: none"> • Name der C-Funktion (siehe Einbindung von C-Funktionen)



Voraussetzung HTML-Browser

Das HTMLBrowser-Objekt ist nur unter Windows verfügbar! Auf dem System muss der InternetExplorer installiert sein. Wird eine URL im Internet definiert (z.B. <http://www.easypagemachine.com/>), so muss das System über das Netzwerk mit dem Internet verbunden sein (z.B. via Gateway). Alternativ können HTML-Seiten auch lokal auf dem Gerät gespeichert werden. In diesem Fall muss der vollständige Pfad der HTML-Seite angegeben werden. (z.B. C:\MyHTML\default.htm)

7.13 Objekt Alarm

#Alarm		Alarmüberwachung
Text/File	<i>MyAlarm.TXT</i>	<ul style="list-style-type: none"> Name der Alarm-Definitionsdatei mit den Alarmvariablen und den Alarmmeldungstexten
Font		<ul style="list-style-type: none">
X,Y,DX,DY		<ul style="list-style-type: none">
Color		<ul style="list-style-type: none">
Backcolor		<ul style="list-style-type: none">
Format		<ul style="list-style-type: none">
Action	#Page=Name	<ul style="list-style-type: none"> Bildseitenwechsel zur Bildseite <i>Name</i> bei Alarmeingängen (Kommen von Alarmen)
Limit1		<ul style="list-style-type: none">
Limit2		<ul style="list-style-type: none">
Action Limit1 Action Limit2		<ul style="list-style-type: none">
VarValue	<i>Variablenname</i>	<ul style="list-style-type: none"> PLC-Variable PLC-Variable die den Alarmbuffer spezifiziert
VarType	STRING	<ul style="list-style-type: none"> Datentyp Array of Word (Länge entsprechend Anzahl Alarme)
VarState		<ul style="list-style-type: none">
Option	Type=name	<ul style="list-style-type: none"> Alarmtyp (erlaubt die Definition von mehreren Alarmobjekten mit derselben Alarmdefinition)
C-Function	<i>C-Funktionsname</i>	<ul style="list-style-type: none"> Name der C-Funktion (siehe Einbindung von C-Funktionen)



Das Objekt Alarm wird global in der ersten Seite im Tabellenblatt „Project“ („Initialisierungs-Seite“) projiziert!

7.13.1 Tabellenblatt Alarm

Im Tabellenblatt „Alarm“ werden den Alarmvariablen die Alarmmeldungstexte, die alarmspezifischen Eigenschaften, Aktionen und Informationen zugewiesen. Das Tabellenblatt „Alarm“ ist wie folgt aufgebaut:

Variable	Alarmnumber	Text/File	Font	Color	Backcolor	Action Alarm Info	Alarm Helptext
[0].0		1 = Alarm(s) aktiv (Richtung: EPAM -> PLC)					
[0].1		1 = Alarmquittierung von Visualisierung (Richtung: EPAM -> PLC)					
[0].2		1 = Alarmquittierung von PLC (Richtung: PLC -> EPAM)					
[0].3		0 = alle Alarme quittiert (Richtung: EPAM -> PLC)					
[0].4		1 = Alarm.INI file (History) gespeichert (Richtung: EPAM -> PLC)					
[0].5		1 = disable Action Page=name (Richtung: PLC -> EPAM)					
[0].6-[0].15		reserviert					
[1].0	1	Beliebiger Alarmtext 1		black	red	#Page=Diagnose, #Page=Foto1	alrmhlp1.txt
[1].1	2	Beliebiger Alarmtext 2		black	red	#Page=Diagnose, #Page=Foto1	alrmhlp1.txt
[1].2	3	Beliebiger Alarmtext 3		black	red	#Page=Diagnose, #Page=Foto1	alrmhlp1.txt
...
[1].15	16	Beliebiger Alarmtext 16		black	yellow	#Page=Diagnose, #Page=Foto2	Alrmhlp2.txt
[2].0	17	Beliebiger Alarmtext 17		black	yellow	#Page=Diagnose, #Page=Foto2	Alrmhlp3.txt
[2].1	18	Beliebiger Alarmtext 18		black	yellow	#Page=Diagnose, #Page=Foto2	Alrmhlp3.txt
...
[2].14	31	Beliebiger Alarmtext 31		black	white	#Page=Diagnose, #Page=Foto3	
[2].15	32	Beliebiger Alarmtext 32		black	white		

Alarmspezifische Farbdefinition
Die Fontdefinition erfolgt in der
Alarmliste

Aktion AlarmInfo:
Alarminfo des selektierten Alarms
aufrufen. z.B:
AlarmInfo=1 ⇒ # Page=Diagnose
AlarmInfo=2 ⇒ # Page=Foto3

Alarm Helptext:
Name der Textdatei mit der
alarmspezifischen Textinformation



Alarmnummern müssen eindeutig definiert werden!



Alarmspezifische Fonts (Spalte Font) werden derzeit nicht unterstützt!



Aufbau Alarmbuffer

Variable spezifiziert das Datenwort und Datenbit im Alarmbuffer. Der Alarmbuffer ist wortorientiert aufgebaut und die Startadresse wird beim Alarmobjekt definiert. Jedes Bit ab dem Datenwort 1 stellt einen Alarm dar. Es können max. $252 \cdot 8 = 2016$ Alarme definiert werden, die letzten 512 Alarme werden in einem Ringbuffer gespeichert (Alarmhistory). Die Alarmhistory und die aktuelle Einstellung (Sortierung, Filter) wird in der Datei MyAlarm.ini netzausfallsicher gespeichert.

Definition in der PLC:

CoDeSys:

Globale Variable:

```
VAR_GLOBAL
  VisuAlarm          AT %MW1000   :   ARRAY[0..2]OF WORD
  AlarmFlagActiv     AT %MX1000.0  :   BOOL;
  AlarmQuitFromVisu AT %MX1000.1  :   BOOL;
  AlarmQuitFromPLC  AT %MX1000.2  :   BOOL;
  AlarmNoQuitActive  AT %MX1000.3  :   BOOL;
  AlarmIniWritten    AT %MX1000.4  :   BOOL;
  AlarmActionDisable AT %MX1000.5  :   BOOL;
  Alarm1             AT %MX1002.0  :   BOOL;
  Alarm2             AT %MX1002.1  :   BOOL;
  Alarm3             AT %MX1002.2  :   BOOL;
  Alarm4             AT %MX1002.3  :   BOOL;
  ...
  ...
  Alarm26            AT %MX1005.1  :   BOOL;
  Alarm27            AT %MX1005.2  :   BOOL;
  Alarm28            AT %MX1005.3  :   BOOL;
  Alarm29            AT %MX1005.4  :   BOOL;
  Alarm30            AT %MX1005.5  :   BOOL;
  Alarm31            AT %MX1005.6  :   BOOL;
  Alarm32            AT %MX1005.7  :   BOOL;
END_VAR
```

Alternativ kann man die Alarme auch über Alarm[x].x adressieren.

Step7:

In der Step7 empfiehlt es sich ein Array einer Bitstruktur anzulegen. Damit kann man die Alarme transparent wie in EPAM ansprechen. Z.B. Alarm[x].Bit[x]

```
Alarm ARRAY[1..x]
  STRUCT
    Bit          ARRAY [0..15] BOOL
  END_STRUCT
```

7.13.2 Ablauf des Alarmhandlings

EPAM prüft zyklisch die Alarmvariablen und trägt Alarme entsprechend dem Status mit Zeitstempel „Kommen“ bzw. „Gehen“ in den Alarmbuffer ein. Bei Alarmeingängen („Kommen“ von Alarmen) kann optional die Aktion Bildseitenwechsel #Page=name ausgeführt werden. Die Alarme werden wie folgt in den Alarmbuffer eingetragen:

1. Ist der älteste Alarm inaktiv und quittiert, wird er durch den neuen Alarm überschrieben.
2. Ist der älteste Alarm inaktiv und nicht quittiert, wird der älteste inaktive nicht quittierte Alarm überschrieben.
3. Wenn kein inaktiver Alarm vorhanden ist, wird der älteste aktive quittierte Alarm überschrieben.
4. Wenn kein inaktiver quittierter Alarm vorhanden ist, wird der älteste Alarm überschrieben. (in diesem Fall ist der Alarm "verloren", d.h. er wird nicht mehr in der Liste angezeigt -> mehr als 512 aktive Alarme!)

7.13.3 Alarmquittierung

Alarmlen können sowohl von EPAM wie auch von der PLC quittiert werden.

Quittierung über EPAM:

Von EPAM aus können Alarmlen einzeln oder alle zusammen quittiert werden, hierbei wird auch das Bit 1 im Statusdatenwort (Alarmquittierung durch Visualisierung) gesetzt und zur PLC gesendet. Anschließend muss das Bit 1 in der PLC wieder zurückgesetzt werden.

Quittierung über PLC:

Von der PLC können nur alle Alarmlen gemeinsam durch setzen des Bit 2 (Alarmquittierung durch Steuerung) quittiert werden. Bei der Alarmquittierung wird die Quittierzeit des quittierten Alarms bzw. aller Alarmlen gesetzt.

7.13.4 Alarmanzeige

Alarmlmeldungen können mit Hilfe des Objekts Alarmlist am Bildschirm dargestellt werden. Im weiteren stehen diverse alarmlnummerabhängige Diagnosemöglichkeiten zur Verfügung (siehe Objekt Alarmlist).

7.13.5 Alarmhistory exportieren

Mit der Buttonaktion **AlarmExport=CSV** kann die Alarmhistory als CSV-Datei ins EPAM-Dataverzeichnis C:\DATA gespeichert werden. Dies erfolgt in folgendem Format:

```
Export of alarmbuffer: myalarm1 @ 2003-04-29 15:03:29
Number of records: 3 sort = lifo
Nr;in[s];out[s];quit[s]
8;1051628587;0;0
3;1051628530;1051628533;1051628539
14;1051628178;1051628591;0
```



Hinweis

Der Zeitstempel erfolgt in Sekunden seit 1.1.1970. Mit der Formel =Zelle/86400+25569 kann der Zeitstempel in Excel in Klartext umgerechnet werden.

7.14 Objekt Alarmlist

#Alarmlist		Ausgabe der Alarmereignisse innerhalb eines rechteckigen Bereiches
Text/File	no= %[Width][Type], tin= Format, tout= Format, tquit= Format, sep= keycode	<ul style="list-style-type: none"> Formatangabe (Bezug links, oben) Keywords: <ul style="list-style-type: none"> no= ...Keyword für die Alarmnummer tin= ...Keyword für Zeit Alarm „Kommen“ tout= ...Keyword für Zeit Alarm „Gehen“ tquit= ...Keyword für Zeit Alarm „Quittiert“ sep= ...Keyword für den Seperator Die einzelnen Keywords sind durch Kommas zu trennen. Nach dem Keyword erfolgt die Formatangabe. Keyword no=: <ul style="list-style-type: none"> ...ohne Keyword no= default Ausgabe %5d der Alarmnummer no= ...nur Keyword no= keine Ausgabe der Alarmnummer no=%... ...Keyword no= mit Formatangabe Ausgabe der Alarmnummer gemäss Format gültige Formatangaben nach %: Width: Zahl ...optionale Anzahl Vorkommastellen Type: d ...Datenformat Integer (Wort) Keyword tin=, tout=, tquit=: <ul style="list-style-type: none"> ...ohne Keywords tin=, tout=, tquit= default Ausgabe von Datum/Zeit txxx= nur Keyword tin=, tout=, tquit= keine Ausgabe von Datum/Zeit txxx=%... ...Keyword tin=, tout=, tquit= mit Formatangabe Ausgabe von Datum/Zeit gemäss Format gültige Formatangaben: <ul style="list-style-type: none"> %d ...Tag des Monats (01-31) %H ...Stunde (00-23) %I ...Stunde (01-12) %j ...Tag im Jahr (001-366) %m ...Monat (01-12) %M ...Minute (00-59) %S ...Sekunde (00-59) %U ...Woche im Jahr (00-53) %w ...Wochentag (0-6), %W ...Woche im Jahr (00-53) %y ...Jahr ohne Jahrhundert (00-99) %Y ...Jahr mit Jahrhundert Keyword sep=: <ul style="list-style-type: none"> ...ohne Keywords sep= default Ausgabe des Separators sep= nur Keyword sep= keine Ausgabe des Separators sep=x ...Keyword sep= mit Keycode Ausgabe des Separators gemäss Keycode

Font	<i>Font.FNT</i>	<ul style="list-style-type: none"> • optionaler Font für den Text
X,Y,DX,DY	<i>Integerwerte</i>	<ul style="list-style-type: none"> • Position, Breite und Höhe der Alarmliste (Bezug links oben!)
Color	<i>Farbname oder Farbnummer</i>	<ul style="list-style-type: none"> • Farbe des Textes der Alarmliste
Backcolor	<i>Farbname oder Farbnummer</i>	<ul style="list-style-type: none"> • Hintergrundfarbe der Alarmliste
Format		<ul style="list-style-type: none"> • ohne Eingabe kein Rand
	Border=Button	<ul style="list-style-type: none"> • Umrandungstyp Button
	Border=Input	<ul style="list-style-type: none"> • Umrandungstyp Eingabefeld
	Border=Rx	<ul style="list-style-type: none"> • Umrandungstyp Rechteck mit Breite x Pixel (1, 3, 5, etc.)
	Border=Shadow	<ul style="list-style-type: none"> • Umrandungstyp Rechteck mit Schatten (3D Effekt)
	Border=Signal	<ul style="list-style-type: none"> • Umrandungstyp Anzeigefeld
Action		<ul style="list-style-type: none"> •
Limit1		<ul style="list-style-type: none"> •
Limit2		<ul style="list-style-type: none"> •
Action Limit1 Action Limit2		<ul style="list-style-type: none"> •
VarValue	<i>Variablenname</i>	<ul style="list-style-type: none"> • Systemvariable Systemvariable, welche den in der Alarmliste aktuell selektierten Alarm speichert
VarType	WORD	<ul style="list-style-type: none"> • Datentyp Wort (16 Bit)
VarState		<ul style="list-style-type: none"> •
Option	AlarmFilter=activ	<ul style="list-style-type: none"> • Alarmfilter setzen: aktive Alarmer anzeigen
	AlarmFilter=activ notquit	<ul style="list-style-type: none"> • Alarmfilter setzen: aktive oder nicht quittierte Alarmer anzeigen
	AlarmFilter=activ+notquit	<ul style="list-style-type: none"> • Alarmfilter setzen: aktive und nicht quittierte Alarmer anzeigen
	AlarmFilter=all	<ul style="list-style-type: none"> • Alarmfilter setzen: alle Alarmer anzeigen
	AlarmFilter=notquit	<ul style="list-style-type: none"> • Alarmfilter setzen: nicht quittierte Alarmer anzeigen
	AlarmSort=FIFO	<ul style="list-style-type: none"> • Alarm in Alarmliste sortieren: ältester Alarm zuerst
	AlarmSort=LIFO	<ul style="list-style-type: none"> • Alarm in Alarmliste sortieren: neuester Alarm zuerst
	AlarmSort=Priority	<ul style="list-style-type: none"> • Alarm in Alarmliste sortieren: Alarm mit hoher Priorität (=niedrige Alarmnummer) zuerst
	Coff	<ul style="list-style-type: none"> • Cursor ausschalten
	DX=x	<ul style="list-style-type: none"> • Breite des Scrollbars in Pixel (Default: 25 Pixel, 0=Scrollbar ausblenden)
	sbimg=<scrollbar.pcx>	<ul style="list-style-type: none"> • Image für kundenspezifischen Scrollbar (Hintergrund)
	thumbimg=<thumb.pcx>	<ul style="list-style-type: none"> • Thumb Image (Slider)
	<i>C-Funktionsname</i>	<ul style="list-style-type: none"> • Name der C-Funktion (siehe Einbindung von C-Funktionen)
C-Function		<ul style="list-style-type: none"> •

Beispiel Formatangaben:

Alarm mit Datum und Quittierzeit anzeigen:

"tin=%d-%m %H:%M,tquit=%H:%M,sep=|" => "13-09 08:34|08:40| 1|Alarm 1"

Das Objekt Alarmlist unterstützt folgende Darstellungsarten. Sie können mit Hilfe von Button-Aktionen und/oder über Fixformatierungen der Alarmliste in Spalte Option gewählt/definiert werden:

7.14.1 Alarmtyp

- Anzeigen nach Typ der Alarme

7.14.2 Alarmfilter

- Anzeige aller Alarme
- nur aktive Alarme
- aktive und nicht quittierte Alarme
- aktive oder nicht quittierte Alarme
- nicht quittierte Alarme

7.14.3 Alarmsortierung

- Anzeige nach Priorität (kleine Alarmnummer = hohe Priorität)
- Anzeige nach Zeit: neuester Alarm zuerst (last in first out)
- Anzeige nach Zeit: ältester Alarm zuerst (first in first out)

7.14.4 Alarmdiagnose/Alarm-Systemvariablen

Anhand der in der Alarmliste selektierten Alarmmeldung kann mit der Aktion AlarmInfo=1 bzw. AlarmInfo=2 auf die im Tabellenblatt „Alarm“ unter Action Alarm Info projektierten 2 Bildseiten gesprungen werden.

Im weiteren kann eine Textliste mit der in Spalte Text/File eingetragenen Systemvariable s_alarm_txtinfo projektiert werden. Anhand der in der Alarmliste selektierten Alarmmeldung kann mit einer Aktion #page=name auf die Bildseite mit der projektierten Textliste gesprungen werden. Nun wird in dieser Textliste die im Tabellenblatt „Alarm“ unter Alarm Helptext projektierte Textdatei eingeblendet.



Im Zusammenhang mit der Alarmliste werden die Alarminformationen des zuletzt selektierten Alarms in folgende Systemvariablen in EPAM gespeichert.

s_alarm_nr	...Alarmnummer, Datentyp: WORD
s_alarm_text	...Alarmtext, Datentyp: STRING
s_alarm_tin	...Zeit Alarm kommen, Datentyp: STRING
s_alarm_tout	...Zeit Alarm gehen, Datentyp: STRING
s_alarm_tquit	...Zeit Alarm quittiert, Datentyp: STRING
s_alarm_tin_dt	...Zeit Alarm kommen, Datentyp: IEC_DT
s_alarm_tout_dt	...Zeit Alarm gehen, Datentyp: IEC_DT
s_alarm_tquit_dt	...Zeit Alarm quittiert, Datentyp: IEC_DT
s_alarm_info	...Alarminfo, Datentyp: STRING
s_alarm_txtinfo	...Alarm Helptext, Datentyp: STRING

7.15 Objekt Alarmmail

#Alarmmail		Email Benachrichtigung aufgrund von Alarmereignissen
Text/File	MyAlarmmail.TXT	• Name der Alarmmail-Definitionsdatei (Parameterdatei)
Font		•
X,Y,DX,DY		•
Color		•
Backcolor		•
Format		•
Action		•
Limit1		•
Limit2		•
Action Limit1 Action Limit2		•
VarValue		•
VarType		•
VarState		•
Option		•
C-Function		•



Funktion von Alarmmail

Mit Hilfe des Alarmmail-Objekts können Alarmmeldungen als Email weitergeleitet werden (**nur auf dem Zielsystem**). Hierbei kann konfiguriert werden, welche Alarmnummer an welche Email-Adressen geleitet werden soll. **Die Funktionen Wochentag und Von, Bis sind zur Zeit nicht implementiert!**

Voraussetzungen

Verbindung zu einem Internet Service Provider z.B. via Gateway (Modem-Router oder Standleitung). Die Gateway-Adresse kann auf dem Zielsystem definiert werden. Um eine Email als SMS weiterzuleiten, wird ein Dienst eines Handynetz-Betreibers benötigt. Dieser Dienst wird z.B. von DiAx in der Schweiz oder D2 in Deutschland zur Verfügung gestellt.

Folgende Umgebungsvariablen müssen definiert werden (EPAM.INI):

SMTP_SERVER=postausgang ...Name des Postausgang-Server (SMTP-Server)
SMTP_FROM=email ...Email-Adresse des Absenders



Das Objekt Alarmmail wird global in der ersten Seite im Tabellenblatt „Project“ („Initialisierungs-Seite“) projiziert!

7.15.1 Tabellenblatt Alarmmail

Im Tabellenblatt „Alarmmail“ werden alle objektspezifischen Parameter definiert. Das Tabellenblatt „Alarmmail“ ist wie folgt aufgebaut:

Alarmnummer	Day of Week	From	To	Email
1	Reserviert	Reserviert	Reserviert	1234567890@gsm.myprovider.com
1	Reserviert	Reserviert	Reserviert	name1@email.com
1	Reserviert	Reserviert	Reserviert	name2@email.com

7.16 Objekt DiagSig

#DiagSig		Diagnose Signal, Darstellung von blinkenden Punkten
Text/File	<i>Alarmnummer1,.....</i>	• Alarmnummer bei der das Diagnose Signal aktiv wird
	<i>Ganzzahl1,.....</i>	• Ganzzahlwert bei dem das Diagnose Signal aktiv wird
	<i>Textstring1,.....</i>	• Stringwert bei dem das Diagnose Signal aktiv wird
Font		•
X,Y,DX,DY	<i>Integerwerte</i>	• Position, Breite und Höhe des Diagnosesignals (Bezug links oben!)
Color	<i>Farbname oder Farbnummer</i>	• Farbe des Diagnosesignals
Backcolor		•
Format		•
Action	#Page=Name	• Bildseitenwechsel zur Bildseite <i>Name</i>
	Close	• Bildseite (Fenster) schliessen
	Close=Name	• Bildseite (Fenster) <i>Name</i> schliessen
Limit1		•
Limit2		•
Action Limit1		• Aktion bei Bedingung aktiv -> inaktiv (z.B. Alarm gehen)
	#Page=Name	• Bildseitenwechsel zur Bildseite <i>Name</i>
	Close	• Bildseite (Fenster) schliessen
	Close=Name	• Bildseite (Fenster) <i>Name</i> schliessen
Action Limit2		• Aktion bei Bedingung inaktiv -> aktiv (z.B. Alarm kommen)
	#Page=Name	• Bildseitenwechsel zur Bildseite <i>Name</i>
	Close	• Bildseite (Fenster) schliessen
	Close=Name	• Bildseite (Fenster) <i>Name</i> schliessen
VarValue		• Ohne Angabe einer Variable wird überprüft, ob die unter Text/File projektierte <i>Alarmnummer</i> aktiv ist
	<i>Variablenname</i>	• PLC-Variable bzw. Systemvariable
VarType		• Ohne Angabe einer Variable wird überprüft, ob die unter Text/File projektierte <i>Alarmnummer</i> aktiv ist
	INT	• Datentyp Integer (16 Bit)
	STRING	• Datentyp String 80 Byte (ohne Angabe: default 80 Byte)
	UINT	• Datentyp Integer ohne Vorzeichen (16 Bit)
	WORD	• Datentyp Wort (16 Bit)
VarState	<i>Variablenname</i>	• PLC-Variable bzw. Systemvariable für Objektstatus
Option		• Objektstatus statisch
	FastFlash	• Objektstatus Blinken mit 2 Hz
	Flash	• Objektstatus Blinken mit 1 Hz
C-Function	<i>C-Funktionsname</i>	• Name der C-Funktion (siehe Einbindung von C-Funktionen)



Funktion des Diagnose Signal

Mit Hilfe des Diagnose Signal kann z.B. aufgrund von Alarmmeldungen ein gezielter Hinweis auf die Alarmursache an der Maschine gegeben werden. D.h. es kann z.B. auf einem Maschinenfoto an der Stelle platziert werden, die den Fehler verursacht hat (z.B. defekter Endschalter). Das Objekt Diagnose Signal überprüft den spezifizierten Textstring mit dem aktuellen Variablenwert bzw. ob die entsprechende Alarmnummer aktiv ist und stellt einen blinkenden Punkt dar sofern die Bedingung zutrifft.

7.17 Objekt Recipe

#Recipe		Rezepturverwaltung
Text/File	<i>MyRecipe.TXT</i>	<ul style="list-style-type: none"> Name der Rezept-Definitionsdatei
Font		<ul style="list-style-type: none">
X,Y,DX,DY		<ul style="list-style-type: none">
Color		<ul style="list-style-type: none">
Backcolor		<ul style="list-style-type: none">
Format		<ul style="list-style-type: none">
Action	#Page=Name	<ul style="list-style-type: none"> Bildseitenwechsel zur Bildseite <i>Name</i> wird bei der Buttonaktion <i>csave=MyRecipeTyp</i> zur Bestätigung aufgerufen, wenn die Rezeptdatei bereits existiert
Limit1		<ul style="list-style-type: none">
Limit2		<ul style="list-style-type: none">
Action Limit1 Action Limit2		<ul style="list-style-type: none">
VarValue	<i>Variablenname</i>	<ul style="list-style-type: none"> PLC-Variable mit folgender Funktion: <ol style="list-style-type: none"> ...Download Request von PLC (EPAM wird von PLC aufgefordert, das aktuelle Rezept bzw. die vom Anwender geänderten Rezeptwerte neu zu laden und in die PLC zu schreiben) ...Up/Download von EPAM beendet (Status von EPAM an PLC, dass der Up/Download beendet ist) ...Upload Request von PLC (EPAM wird von PLC aufgefordert, die Rezeptwerte aus der PLC zu lesen und in ACTUAL.DAT zu speichern) ...Download läuft (Status von EPAM an PLC, dass das aktuelle Rezept bzw. die vom Anwender geänderten Rezeptwerte geladen und in die PLC geschrieben werden) ...Upload läuft (Status von EPAM an PLC, dass die Rezeptwerte aus der PLC gelesen werden und in ACTUAL.DAT gespeichert wird) ...Rezept in aktuellem Dateinamen (<i>s_myrecipe_file</i>) mit aktuellem Namen (<i>s_myrecipe_name</i>) speichern Download Request von PLC (EPAM wird von PLC aufgefordert, das letzte Rezeptfile neu zu laden und in die PLC zu schreiben → Änderungen vom Anwender verwerfen)
VarType	INT	<ul style="list-style-type: none"> Datentyp Integer (16 Bit)
	UINT	<ul style="list-style-type: none"> Datentyp Integer ohne Vorzeichen (16 Bit)
	WORD	<ul style="list-style-type: none"> Datentyp Wort (16 Bit)
VarState		<ul style="list-style-type: none">

Option		<ul style="list-style-type: none"> vom Anwender geänderte Rezeptwerte im ACTUAL.DAT werden bei EPAM-Start in die PLC geschrieben
	NoDownload	<ul style="list-style-type: none"> Weder die Rezeptwerte des aktuell geladenen Rezeptes noch die vom Anwender geänderten Rezeptwerte im ACTUAL.DAT werden bei EPAM-Start in die PLC geschrieben Hinweis: ACTUAL.DAT wird nicht aktualisiert! (wird nicht benötigt!)
	NoActual	<ul style="list-style-type: none"> Rezeptwerte des aktuell geladenen Rezeptes und nicht die vom Anwender geänderten Rezeptwerte im ACTUAL.DAT werden bei EPAM-Start in die PLC geschrieben Hinweis: ACTUAL.DAT wird nicht aktualisiert! (wird nicht benötigt!)
	Filename=Auto	<ul style="list-style-type: none"> Rezeptdateinamen automatisch vergeben (alphanumerisch)
	Filename=Auto10	<ul style="list-style-type: none"> Rezeptdateinamen automatisch vergeben (numerisch)
C-Function	<i>C-Funktionsname</i>	<ul style="list-style-type: none"> Name der C-Funktion (siehe Einbindung von C-Funktionen)



Das Objekt Recipe wird global in der ersten Seite im Tabellenblatt „Project“ („Initialisierungs-Seite“) projiziert!

7.17.1 Tabellenblatt Recipe

Im Tabellenblatt „Recipe“ werden alle Rezepturvariablen mit Typ und allenfalls einem Initialisierungswert angegeben. Das Tabellenblatt „Recipe“ ist wie folgt aufgebaut:

Variable	VarType	Value
#Recipe=Default		Path=
MyRecipeVar1	WORD	123
MyRecipeVar2	WORD	345
...		
#Checksum=		

Rezepttyp Default: kein Eintrag
Rezepttype entspricht der
Rezeptbezeichnung.
z.B. Type=MyRecipe

Rezept Verzeichnis Default: path=
Die Rezeptur wird in das rezepttypen-
spezifische Verzeichnis abgelegt.
z.B. C:\DATA\MyRecipe



Systemvariablen in Rezepturen

In der Rezeptur können Systemvariablen als Rezeptvariable und auch als Wert definiert werden. In diesem Fall wird der aktuelle Wert der Systemvariablen ins Rezept gespeichert und ggf. in die PLC geschrieben.



Konsistenz von Rezeptdateien

Zur Sicherstellung der Datenkonsistenz von Rezeptdateien werden diese mit einer Checksumme abgeschlossen. Die Checksumme wird beim Speichern einer Rezeptur automatisch berechnet und in der letzten Zeile eingefügt. Werden diese Dateien von Hand geändert oder mit Excel neu erstellt, so muss der Anwender die Datei mit der Sequenz ‚#Checksum=<CRLF>‘ abschliessen. Fehlt dieser Eintrag, oder **ist die Checksumme falsch, so ist die Rezeptdatei ungültig und wird nicht geladen!**



Konsistenz von Rezeptwerten

Das Speichern und Laden der Rezeptdaten erfolgt NICHT zyklussynchron! Die Datenkonsistenz über alle Rezeptwerte muss über den Rezeptstatus in der PLC überprüft werden. Ein Rezept ist vollständig geladen, wenn der Rezeptstatus 2 ist.



Mehrstufige Rezepturen

Mit dem Rezept-Datentyp RECIPE:mytype wird eine Rezeptvariable vom Typ STRING definiert, die den Namen eines Rezepts vom Typ mytype beinhaltet. Damit wird nach dem Laden des Rezepts automatisch das entsprechende Rezept vom Typ mytype geladen.

7.17.2 Rezepturverwaltung

Die Variablen im Tabellenblatt „Recipe“ (Datei *MyRecipe.TXT*) definieren die für eine Rezeptur benötigten Prozessvariablen in der Form Name, Typ und Wert. Die Default-Rezeptur *MyRecipe.TXT* enthält die Defaultwerte für die angegebenen Variablen (Wert=Defaultwert). Die Rezepturvariablen werden üblicherweise global angelegt (siehe auch globale Objekte) und mit den Defaultwerten initialisiert. Jede Änderung einer Rezepturvariable wird überwacht und geänderte Werte werden in der Datei 'ACTUAL.DAT' netzausfallsicher gespeichert. Beim nächsten Starten von EPAM werden daher die aktuellen Variablenwerte wieder geladen und auch zur Steuerung übertragen (Default). Dieses Handling kann mit den verschiedenen Optionen geändert werden.

Es ist möglich mehrere Rezeptobjekte mit verschiedenen Rezepttypen (Type) zu definieren. Hiermit können z.B. Rezepturen für produktspezifische Einstellungen und maschinenabhängige Konfigurationen getrennt verwaltet werden.



Anlegen der Rezeptverzeichnisse

Die Rezepturen werden in die rezepttypenspezifischen Verzeichnisse abgelegt. Die rezepttypenspezifischen Verzeichnisse *MyRecipe* werden im EPAM-Dataverzeichnis C:\DATA und EPAM-Backupverzeichnis C:\BACKUP automatisch angelegt.

z.B.

Rezepttyp *MyTyp1* in Verzeichnis C:\DATA\MyTyp1\ bzw. C:\BACKUP\MyTyp1\

Rezepttyp *MyTyp2* in Verzeichnis C:\DATA\MyTyp2\ bzw. C:\BACKUP\MyTyp2\

7.17.3 Rezept laden

Rezepte werden mit der Dateierweiterung *.DAT gespeichert. Mit Hilfe der Systemvariable '*s_myRecipeType_file*' kann der Dateiname einer bestehenden Rezeptdatei (*.DAT) eingegeben werden (max. 8 Zeichen ohne Dateierweiterung). Mit der Aktion *Recipe=load* werden dann die Variablenwerte der durch '*s_myRecipeType_file*' definierten Rezeptdatei geladen.

Eine komfortablere Möglichkeit stellt das Objekt *RecipeList* dar. Hier kann aus einer Liste existierender Rezeptdateien mit der Aktion *load=list* ein Rezept ausgewählt und geladen werden (siehe auch Objekt *RecipeList*).

7.17.4 Rezept speichern

Durch Angabe eines Dateinamens in der Systemvariable '*s_myRecipeType_file*' und optional einer Rezeptbezeichnung (Systemvariable '*s_myRecipeType_name*'), sowie der Aktion *save=myRecipeType* werden die aktuellen Werte der Rezeptvariablen in einer neuen Rezeptdatei (*.DAT) gespeichert. Existierende Dateien werden dabei überschrieben.

Im weiteren können Rezepte auch mit Hilfe des Objektes *RecipeList* gespeichert werden. Die Aktion *csave=myRecipeType* ermöglicht eine Abfrage, falls die Datei bereits existiert. In diesem Fall wird die Bildseite aufgerufen, die im entsprechenden Rezeptobjekt definiert ist.

7.17.5 Rezept in EXCEL erstellen

Im Tabellenblatt „Recipe“ können ab der Spalte D anwenderspezifische Rezepte in EXCEL vordefiniert und mit dem Makro „Build Recipes“ automatisch generiert werden. Hierfür können ab der Spalte D folgende Eingaben gemacht werden:

Zelle D1: Name der Rezeptdatei ohne Erweiterung (max. 8 Zeichen)

Zelle D2: Kommentar

Zelle D3: Rezeptname „#Recipe=myName“

Ab Zelle D4: Rezeptwerte entsprechend den definierten Rezeptvariablen (analog zu Spalte Value)

7.18 Objekt RecipeList

#RecipeList		Ausgabe der Rezeptliste innerhalb eines rechteckigen Bereiches
Text/File	<i>Text</i> % <i>[Flags]</i> <i>[Width]</i> <i>[Type]</i> <i>Text</i> ...	<ul style="list-style-type: none"> Formatstring mit Formatangabe (Bezug links, oben) Bsp. %-9f %-16n %d-%m-%Y %H:%M ohne Formatangabe: %12f %-12t %n gültige Formatangaben nach %: <i>Flags:</i> - ...optionale Ausgabe linksbündig 0 ...optionale Ausgabe mit Vornullen <i>Width:</i> Zahl ...optionale Feldbreite <i>Type:</i> f ...optionale Anzeige des Dateinamens n ...optionale Anzeige des Rezeptnamens ... (Beschreibung) t ...optionale Anzeige des Rezepttyps
Font	<i>Font.FNT</i>	<ul style="list-style-type: none"> optionaler Font für den Text
X,Y,DX,DY	<i>Integerwerte</i>	<ul style="list-style-type: none"> Position, Breite und Höhe der Alarmliste (Bezug links oben!)
Color	<i>Farbname oder Farbnummer</i>	<ul style="list-style-type: none"> Farbe des Textes der Rezeptliste
BackColor	<i>Farbname oder Farbnummer</i>	<ul style="list-style-type: none"> Hintergrundfarbe der Rezeptliste

Format		<ul style="list-style-type: none"> • ohne Eingabe kein Rand
	Border=Button	<ul style="list-style-type: none"> • Umrandungstyp Button
	Border=Input	<ul style="list-style-type: none"> • Umrandungstyp Eingabefeld
	Border=Rx	<ul style="list-style-type: none"> • Umrandungstyp Rechteck mit Breite x Pixel (1, 3, 5, etc.)
	Border=Shadow	<ul style="list-style-type: none"> • Umrandungstyp Rechteck mit Schatten (3D Effekt)
	Border=Signal	<ul style="list-style-type: none"> • Umrandungstyp Anzeigefeld
Action		<ul style="list-style-type: none"> •
Limit1		<ul style="list-style-type: none"> •
Limit2		<ul style="list-style-type: none"> •
Action Limit1 Action Limit2		<ul style="list-style-type: none"> •
VarValue	<i>Variablenname</i>	<ul style="list-style-type: none"> • Systemvariable Systemvariable, welche das in der Rezeptliste aktuell selektierte Rezept speichert
VarType	WORD	<ul style="list-style-type: none"> • Datentyp Wort (16 Bit)
VarState		<ul style="list-style-type: none"> •
Option	DX=x	<ul style="list-style-type: none"> • Breite des Scrollbars in Pixel (Default: 25 Pixel)
	Type=myrecipetype	<ul style="list-style-type: none"> • Rezepturtyp (z.B. myRecipeType) setzen
	Type=off	<ul style="list-style-type: none"> • Rezepturtyp rücksetzen (alle)
	sbimg=<scrollbar.pcx>	<ul style="list-style-type: none"> • Image für kundenspezifischen Scrollbar (Hintergrund)
	thumbimg=<thumb.pcx>	<ul style="list-style-type: none"> • Thumb Image (Slider)
C-Function	<i>C-Funktionsname</i>	<ul style="list-style-type: none"> • Name der C-Funktion (siehe Einbindung von C-Funktionen)



Rezepte laden/speichern/löschen

In der Rezeptliste werden alle *.DAT Dateien des aktuellen Rezepttyps angezeigt. Es kann eine Datei ausgewählt und mit Hilfe der Button-Aktionen `Load=myRecipeType/Load=list`, `Save=myRecipeType/Save=List` bzw. `Delete=myRecipeType/Delete=list` ein Rezept geladen, gespeichert oder gelöscht werden.



Wechseln zwischen den Rezepttypen

Zwischen den verschiedenen Rezepttypen kann mit Hilfe der Button-Aktion `Type=myRecipeType` gewechselt werden. Dabei wird auch automatisch das Rezeptverzeichnis gewechselt.



Wechseln des Rezeptlisten-Verzeichnis

Im weiteren kann durch setzen der Systemvariable `s_recipe_path` (mit Hilfe einer Button-Aktion), der aktuelle Pfad gewechselt werden.

z.B.

`SetVar='A:'` ...Rezept von/auf Diskette laden/speichern

`SetVar=""` ...zurück ins aktuelle Verzeichnis



Rezeptliste sortieren

Die Rezeptliste kann nach Dateiname, Rezeptname, Zeit und Rezepttyp durch setzen der nachfolgenden Button-Aktionen sortiert werden:

`Sort=File` ...Rezeptliste nach Dateinamen sortieren

`Sort=Name` ...Rezeptliste nach Rezeptnamen sortieren

`Sort=Time` ...Rezeptliste nach Dateidatum sortieren

`Sort=Type` ...Rezeptliste nach Rezepttyp sortieren

`Sort=Number` ...Rezeptliste nach Rezeptname numerisch sortieren (nur bei Verwendung numerischer Dateinamen möglich)

7.19 Objekt ScreenSaver

#ScreenSaver		Bildschirmschoner nach definierten Zeiten wird ein Text/Bild auf dem Bildschirm dargestellt und die Hintergrundbeleuchtung (Backlight) reduziert
Text/File	<i>Bild0.PCX,Bild1.PCX,...</i>	<ul style="list-style-type: none"> Name einer PCX-Bilddatei optionale weitere Namen von PCX-Bilddateien
	<i>Text0,Text1,...</i>	<ul style="list-style-type: none"> Textstring optional weitere Textstrings Text kann mit Zeilentrenner ' ' (ASCII-Zeichen 124 bzw. 7CH) auf mehrere Zeilen aufgeteilt werden
		<ul style="list-style-type: none"> wird weder Bild noch Text eingegeben, wird beim Aktivieren des Bildschirmschoners nur die Hintergrundbeleuchtung (Backlight) reduziert
Font	<i>Font.FNT</i>	<ul style="list-style-type: none"> optionaler Font für den Text
X,Y,DX,DY		<ul style="list-style-type: none">
Color	<i>Farbname oder Farbnummer</i>	<ul style="list-style-type: none"> Farbe des Textes des Bildschirmschoners
BackColor	<i>Farbname oder Farbnummer</i>	<ul style="list-style-type: none"> Hintergrundfarbe des Bildschirmschoners
Format		<ul style="list-style-type: none"> ohne Eingabe Default: random
	random	<ul style="list-style-type: none"> Text bzw. Bild wird zufällig am Bildschirm positioniert
	move	<ul style="list-style-type: none"> Text bzw. Bild wird auf dem Bildschirm verschoben
Action	click	<ul style="list-style-type: none"> Bildschirmschoner wird beim ersten Click an einer beliebigen Stelle auf dem Bildschirm geschlossen
	click=inside	<ul style="list-style-type: none"> Bildschirmschoner wird nur bei einem Click innerhalb des Textes/Bildes geschlossen. Sind weitere Texte/Bilder definiert, so müssen diese innerhalb von 4 Sekunden bestätigt werden um den Bildschirmschoner zu beenden (Schutz vor unbeabsichtigter Bedienung).
Limit1		<ul style="list-style-type: none">
Limit2		<ul style="list-style-type: none">
Action Limit1 Action Limit2		<ul style="list-style-type: none">
VarValue	<i>Variablenname</i>	<ul style="list-style-type: none"> PLC-Variable bzw. Systemvariable mit folgender Funktion: 0 ...aktiver Bildschirmschoner nicht beenden 1 ...aktiver Bildschirmschoner beenden
VarType	INT	<ul style="list-style-type: none"> Datentyp Integer (16 Bit)
	UINT	<ul style="list-style-type: none"> Datentyp Integer ohne Vorzeichen (16 Bit)
	WORD	<ul style="list-style-type: none"> Datentyp Wort (16 Bit)
VarState	<i>Variablenname</i>	<ul style="list-style-type: none"> PLC-Variable bzw. Systemvariable für Objektstatus mit folgender Funktion: 0 ...Bildschirmschoner wird nach <i>time1</i>, <i>time2</i> aktiviert 1 ...Bildschirmschoner wird deaktiviert
Option	Timeout= <i>time1,time2</i>	<ul style="list-style-type: none"> Zeit in Minuten oder Name einer Systemvariablen <i>time1</i> ...Zeit in Minuten bis zur Aktivierung des Bildschirmschoners und Reduzierung der Hintergrundbeleuchtung (Backlight 50%) <i>time2</i> ...optionale Zeit in Minuten bis zum Ausschalten der Hintergrundbeleuchtung (Backlight 0%)
C-Function	<i>C-Funktionsname</i>	<ul style="list-style-type: none"> Name der C-Funktion (siehe Einbindung von C-Funktionen)

**Bilschirmschoner in der PLC deaktivieren**

Bei aktivem Screensaver erfolgt während der Darstellung des Textes bzw. PCX-Bildes nur eine Aktualisierung der globalen Objekte Alarm, DataLog, und Recipe. Der Screensaver kann jedoch bei wichtigen Ereignissen durch Setzen der Variable in VarState von der Steuerung aus deaktiviert werden.

**Aktiver Bilschirmschoner bei Alarmevent**

Erfolgt bei aktivem Screensaver ein Alarmevent (eingehender Alarm, austretender Alarm), wird der aktive Screensaver beendet.

**Schutz vor Fehlbedienung**

Die Angabe von weiteren Text bzw. PCX-Bildern (jeweils durch Komma getrennt) bewirkt, dass die Deaktivierung des Bilschirmschoners mehrere Stufen durchläuft. D.h. das erste Touchereignis bewirkt die Anzeige des nächsten Text/Bildes, das dann wiederum bestätigt werden muss um den Bilschirmschoner zu beenden. Dadurch kann das unbeabsichtigte Auslösen von Funktionen praktisch ausgeschlossen werden.

7.20 Objekt Password

#Password		Passwortverwaltung
Text/File		•
Font		•
X,Y,DX,DY		•
Color		•
Backcolor		•
Format		•
Action		• passwortgeschützte Objekte werden sichtbar aber inaktiv dargestellt (disable)
	#Page=Name	• Bildseitenwechsel zur Bildseite <i>Name</i> bei Betätigung eines passwortgeschützten Objektes (typischerweise Dialog zur Passwordeingabe). Bei erfolgreicher Eingabe des Passwortes wird die Aktion des passwortgeschützten Objektes nicht automatisch ausgeführt. Passwortgeschützte Objekte werden sichtbar und aktiv dargestellt.
	off	• passwortgeschützte Objekte werden nicht sichtbar und inaktiv dargestellt (off)
Limit1		•
Limit2		•
Action Limit1	#Page=Name	• Bildseitenwechsel zur Bildseite <i>Name</i> nach erfolgter fehlerhafter Passwordeingabe (nur in Verbindung mit Action #Page=Name)
Action Limit2		•
VarValue	s_password	• Systemvariable
VarType	STRING	• Datentyp String 80 Byte (ohne Angabe: default 80 Byte)
VarState		•
Option	Bitwise=AND	• PWL wird bitweise ausgewertet und nicht nach Höhe der Berechtigungsstufe
	Master_PW=x	• Definition des Masterpasswortes Passwort mit höchster Berechtigungsstufe
	SysPW=Off	• Systempasswort (Berechnung des Passwortes aus Tag und Monat) deaktivieren (Default: ein)
	Timeout=time	• Zeit in Minuten ohne Toucherreignis bis die aktuelle Berechtigung automatisch zurückgesetzt wird (PWL=0)
	Keep_PWL	• Aktive Berechtigungsebene bleibt nach weiterer, allerdings fehlerhafter Passwordeingabe, erhalten
C-Function	C-Funktionsname	• Name der C-Funktion (siehe Einbindung von C-Funktionen)



Funktion der Passwortverwaltung

Mit Hilfe der Passwortverwaltung können bis 32767 Berechtigungsstufen realisiert werden. Jedem Objekt kann mit der Option PWL=x ein bestimmter Passwortlevel zugeordnet werden. Der Passwortlevel nach Programmstart ist 0. Durch Eingabe des Masterpasswortes (Master_PW=x) wird die höchste Berechtigungsstufe (32767) gesetzt. Diese Stufe kann auch durch Eingabe des Wertes Tag * Monat + Tag erreicht werden. Mit den Systemvariablen s_password_1, s_password_2, ... vom Typ STRING, können Passwörter mit den entsprechenden Berechtigungsstufen 1, 2, ... definiert werden (**fortlaufend, lückenlos!**). Mit der Button-Aktion PWL=x kann die aktuelle Berechtigungsstufe z.B. beim Verlassen einer Bildseite zurückgesetzt werden.

Der Zugriff ist erlaubt wenn der aktuelle PWL >= der Option PWL des Objekts ist.

Option Bitwise=AND

Mit dieser Option können die 15 Bit des PWL für 15 Berechtigungsstufen verwendet werden, die dann matrixförmig den Funktionen zugeordnet werden können.

Mit den Systemvariablen s_password_1, s_password_2, ... vom Typ STRING, können Passwörter mit den entsprechenden Berechtigungsstufen 1 (Bit0=1), 2 (Bit1=1), ... definiert werden (**fortlaufend, lückenlos!**).

Die Verknüpfung des aktuellen PWL mit dem objektspezifischen erfolgt durch logische AND Verknüpfung: Zugriff erlaubt wenn: (aktueller PWL AND Option PWL) > 0

Beispiel:

PasswortBit	...	Bit 3	Bit 2	Bit 1	Bit 0		EPAM-Projekt
Akt. PWL		8	4	2	1		Option-PWL
Funktion/Stufe		Master	Service	Einrichter	Bediener		
Bild1		1	0	0	1	→	9 (=0x09)
Bild2		1	0	1	0	→	10 (=0x0A)
Bild3		1	1	0	1	→	13 (=0x0D)
...							

Der Service hat Zugriff auf Bild3, aber nicht auf Bild2 und 1.

Der Bediener hat Zugriff auf Bild 1 und 3, aber nicht auf Bild2.

Beim Login wird der aktuelle PWL entsprechen gesetzt (z.B. Service Bit2=1 → PWL=4). Damit sind alle Objekte zugänglich, die in der Option PWL das Bit 2 = 1 haben.



Das Objekt Password wird global in der ersten Seite im Tabellenblatt „Project“ („Initialisierungs-Seite“) projiziert!

7.21 Objekt Scrollist

#Scrollist		Scroll-Liste
Text/File		•
Font		•
X,Y,DX,DY	<i>Integerwerte</i>	• Position, Breite und Höhe der Scroll-Liste (Bezug links oben!)
Color	<i>Farbname oder Farbnummer</i>	• Farbe des Textes der Scroll-Liste
BackColor	<i>Farbname oder Farbnummer</i>	• Hintergrundfarbe der Scroll-Liste
Format		• ohne Eingabe kein Rand
	Border=Button	• Umrandungstyp Button
	Border=Input	• Umrandungstyp Eingabefeld
	Border=R_x	• Umrandungstyp Rechteck mit Breite x Pixel (1, 3, 5, etc.)
	Border=Shadow	• Umrandungstyp Rechteck mit Schatten (3D Effekt)
	Border=Signal	• Umrandungstyp Anzeigefeld
Action	Scrolly=x	• Scroll-Liste um x Pixel vertikal verschieben
Limit1		•
Limit2	<i>Integerwert</i>	• Wert muss folgendermassen berechnet: Anzahl Einträge in Scroll-Liste – sichtbare Einträge in Scroll-Liste + 1
Action Limit1 Action Limit2		•
VarValue	<i>Variablenname</i>	• Systemvariable speichert die Scrollposition der Scroll-Liste
VarType	INT	• Datentyp Integer (16 Bit)
	UINT	• Datentyp Integer ohne Vorzeichen (16 Bit)
	WORD	• Datentyp Wort (16 Bit)
VarState		•
Option	DX=x	• Breite des Scrollbars in Pixel (Default: 25 Pixel)
	sbimg=<scrollbar.pcx>	• Image für kundenspezifischen Scrollbar (Hintergrund)
	thumbimg=<thumb.pcx>	• Thumb Image (Slider)
C-Function	<i>C-Funktionsname</i>	• Name der C-Funktion (siehe Einbindung von C-Funktionen)



Funktion der Scroll-Liste

Mit Hilfe der Scroll-Liste können alle nachfolgenden Objekte, welche die Option ‚scroll‘ definiert haben, in X- oder Y-Richtung verschoben werden. Objekte die hierbei ganz oder teilweise ausserhalb des Bereiches der Scroll-Liste liegen, werden nicht dargestellt. Die Objekte werden hierbei positioniert, als ob der Bildschirm die benötigte Dimension hätte (z.B. 800 Pixel hoch)

Das Objekt Scrolliste ermöglicht somit die Gestaltung von Parameterlisten mit Standard-Objekten in beliebiger Form. In Kombination mit der Möglichkeit der indizierten Variablen-adressierung können hiermit mit einer Bildseite Parameter für mehrere Achsen eingegeben werden.

Pro Bildseite kann nur EINE Scroll-Liste gleichzeitig aktiv sein!

7.22 Objekt DataLog

#DataLog		Aufzeichnung von PLC-Daten/Variablen in eine DataLog-Datei
Text/File	<i>MyDatalog.TXT</i>	<ul style="list-style-type: none"> Name der Datalog-Definitionsdatei (Parameterdatei) mit den Datalog-Parametern
Font		<ul style="list-style-type: none">
X,Y,DX,DY		<ul style="list-style-type: none">
Color		<ul style="list-style-type: none">
BackColor		<ul style="list-style-type: none">
Format		<ul style="list-style-type: none">
Action		<ul style="list-style-type: none">
Limit1		<ul style="list-style-type: none">
Limit2		<ul style="list-style-type: none">
Action Limit1 Action Limit2		<ul style="list-style-type: none">
VarValue	<i>Variablenname</i>	<ul style="list-style-type: none"> PLC-Variable PLC-Variable der Datalogstruktur
VarType	STRING	<ul style="list-style-type: none"> Datentyp Array of Byte (Länge entsprechend der Länge der Datenlogstruktur)
VarState		<ul style="list-style-type: none">
Option	Type=name	<ul style="list-style-type: none"> Definition des Datalog-Types. Damit können mehrere Datalog-Objekte mit derselben Datalogdatei definiert werden.
C-Function		<ul style="list-style-type: none">



Funktion des DataLog

Das Datalog-Objekt dient zur Aufzeichnung von PLC-Daten in eine Datei. Die Einträge erfolgen als ASCII-Text. Die Spalten sind durch einen Separator getrennt. Jeder Eintrag belegt eine Zeile und wird durch einen Zeitstempel (s.a. [Alarmhistory exportieren](#)) ergänzt.

Die Datenschnittstelle zur PLC ist eine beliebige Struktur von elementaren PLC-Variablen. Dabei muss das erste Recordfeld (Ctrl) stets vom Typ DWORD sein. Über diese Variable kann die PLC das DataLog-Objekt steuern. Jedes weitere Strukturelement wird protokolliert, wenn ein entsprechender Eintrag in der Parameterdatei existiert.



Achtung:

Die gesamt Länge dieser Struktur darf 80 Byte nicht überschreiben!

Da die einzelnen Variablen in der Struktur wordaligned abgelegt sind (gerade Adressen) , sind die dadurch entstehenden zusätzlichen Bytes einzurechnen!

Das DataLog-Objekt eignet sich nicht zur dauernden Aufzeichnung von rasch ändernden Werten auf CompactFlash. Die Schreibzyklen von CompactFlash-Speichern sind begrenzt! (Typisch 100'000 Schreibzyklen, Details siehe Spezifikation der entsprechenden CompactFlash™).

Das Datalog-Objekt zeichnet die Daten daher standardmässig im RAM-Drive auf. Mit Hilfe von Button-Aktionen können diese Daten bei Bedarf auf dem CompactFlash™ gesichert werden.

Datalog-Recordfeld Ctrl mit folgenden Funktionen:

- 16#0000001 Ctrl-Bit „Trigger“
PLC fordert EPAM durch Setzen des Ctrl-Bits „Trigger“ auf, mit dem Aufzeichnen der Datalog-Einträge zu beginnen (mit Eintrag #dt=x in der Parameterdatei) bzw. nur einen Datalog-Eintrag aufzuzeichnen (ohne Eintrag #dt=x in der Parameterdatei).
- 16#0000002 Ctrl-Bit „Acknowledge Trigger“
EPAM teilt PLC durch Setzen des Ctrl-Bits „AcknowledgeTrigger“ mit, dass der Datalog-Eintrag aufgezeichnet wurde (ohne Eintrag #dt=x im Parameterdatei). Das Rücksetzen des Ctrl-Bits „Acknowledge Trigger“ muss durch PLC erfolgen.
- 16#0000004 Ctrl-Bit „Reset“
PLC fordert EPAM durch Setzen des Ctrl-Bits „Reset“ auf, die Datalogdatei im EPAM-Logverzeichnis zu löschen (Ramdrive EPAM:).
- 16#0000008 Ctrl-Bit „Save“
PLC fordert EPAM durch Setzen des Ctrl-Bits „Save“ auf, die Datalogdatei ins EPAM-Dataverzeichnis zu speichern (C:\DATA).
- 16#0000010 Ctrl-Bit „HMI-Reset“
EPAM teilt PLC durch Setzen des Ctrl-Bits „HMI-Reset“ mit, dass die Datalogdatei im EPAM-Logverzeichnis (Ramdrive EPAM:) mit der Button-Aktion logdelete=MyDatalog gelöscht wurde. Das Rücksetzen des Ctrl-Bits „HMI-Reset“ muss durch PLC erfolgen.
- 16#0000020 Ctrl-Bit „HMI-Save“
EPAM teilt PLC durch Setzen des Ctrl-Bits „HMI-Save“ mit, dass die Datalogdatei ins EPAM-Dataverzeichnis (C:\DATA) mit der Button-Aktion logsave=MyDatalog gespeichert wurde. Das Rücksetzen des Ctrl-Bits „HMI-Save“ muss durch PLC erfolgen.

Triggern der Logfunktion

1. Mit Eintrag #dt=x im der Parameterdatei:
 - Durch Setzen des Ctrl-Bits „Trigger“ in der PLC wird der Datalog-Vorgang gestartet.
 - Durch Rücksetzen des Ctrl-Bits „Trigger“ mit der PLC wird der Datalog-Vorgang gestoppt.
2. Ohne Eintrag #dt=x in der Parameterdatei:
 - Durch Setzen des Ctrl-Bits „Trigger“ mit der PLC wird ein Datalog-Eintrag gespeichert.
 - Wurde der Datalog-Eintrag geschrieben, wird anschliessend das Ctrl-Bit „Acknowledge Trigger“ durch das EPAM gesetzt.
 - Anschliessend muss das Ctrl-Bit „Trigger“ mit der PLC wieder zurückgesetzt werden. Nun kann mit dem weiteren Setzen des Ctrl-Bits „Trigger“ mit der PLC der nächste Datalog-Eintrag gespeichert werden.

Dateihandling

Wenn kein Dateiname bestimmt wurde (#file=), erzeugt das Objekt den Dateinamen automatisch aus dem Datum (yymmdd.log). In diesem Fall wird auf jeden neuen Kalendertag eine neue Datei erstellt.

Wenn noch keine Datei vorhanden ist, wird eine neue Datei angelegt. Ist bereits eine Datei vorhanden, werden die Datalogzeilen am Ende der Datei angefügt. Wenn die angegeben Grösse der Datei erreicht wurde, werden die ältesten Datalog-Einträge überschrieben (Ringbuffer).

Wichtig!

Alle Zeilen einer Datalog-Datei müssen gleich lang sein! → Max. Länge des Datentyps bei Formatstring beachten!



Das Objekt DataLog muss global in der ersten Seite im Tabellenblatt „Project“ („Initialisierungs-Seite“) projiziert werden!

7.22.1 Tabellenblatt DataLog

Im Tabellenblatt „DataLog“ werden alle objektspezifischen Parameter definiert. Zusätzlich werden alle zu registrierenden Variablen mit Format und Typ angegeben. Als Formatstring wird die ANSI-C Notation verwendet. Das Tabellenblatt „DataLog“ ist wie folgt aufgebaut:

Format	VarType	Comment
#size=100		Grösse des Logfile in kB
#file=MyLog.log		Logfile name
#seperator=;		Separator
#format=user		format=user ...fixed width user format, larger values will be truncated (e.g. Format %3d: Value 1000 -> 999) format=auto ...fixed width format corresponding to VarType (format will be defined automatically, so that the largest value for VarType can be stored) format=V3.10 ...old compatible mode, user is responsible to define formats which are big enough to hold the largest value
#dt=5		Timescale[s]
#timeformat= %d.%m.%Y %H:%M:%S		Timeformat
%1u	BOOL	BOOL-Variable (Dezimale Darstellung ohne Vorzeichen)
%1hu	BOOL	BOOL-Variable (Hexadezimale Darstellung)
%3u	BYTE	BYTE-Variable (Dezimale Darstellung ohne Vorzeichen)
%2x	BYTE	BYTE-Variable (Hexadezimale Darstellung)
%6hu	WORD	WORD-Variable (Dezimale Darstellung ohne Vorzeichen)
%4hx	WORD	WORD-Variable (Hexadezimale Darstellung)
%8lu	DWORD	DWORD-Variable (Dezimale Darstellung ohne Vorzeichen)
%8lx	DWORD	DWORD-Variable (Hexadezimale Darstellung)
%4d	SINT	SINT-Variable (Dezimale Darstellung)
%6d	INT	INT-Variable (Dezimale Darstellung)
%12ld	DINT	DINT-Variable (Dezimale Darstellung)
%3u	USINT	USINT-Variable (Dezimale Darstellung ohne Vorzeichen)
%5hu	UINT	UINT-Variable (Dezimale Darstellung ohne Vorzeichen)
%12lu	UDINT	UDINT-Variable (Dezimale Darstellung ohne Vorzeichen)
%20.4f	REAL	REAL-Variable (Gleitkomma Darstellung)
%e	REAL	REAL-Variable (Exponential Darstellung)
%d.%m.%Y %H:%M:%S %^S	IEC_DT	DT-Variable (Datum und Zeit Darstellung)



Variablen vom Typ STRING werden im DataLog nicht unterstützt!

Beispiel einer Typendefinition (IEC61131):

Hierzu muss dem Objekt eine PLC-Variable mit folgender Struktur zugeordnet werden:

Globale Variable:

```
VAR_GLOBAL
    Datalog          :    DatalogType;
END_VAR
```

PLC-Datentyp:

```
TYPE DatalogType:
    STRUCT
        Ctrl          :    DWORD;
        TimeStamp     :    DWORD;    ...Zeitstempel von PLC (>0), Zeitstempel von EPAM (=0)
        VisuVarBoolDecimal :    BOOL;
        VisuVarBoolHex   :    BOOL;
        VisuVarByteDecimal :    BYTE;
        VisuVarByteHex   :    BYTE;
        VisuVarWordDecimal :    WORD;
        VisuVarWordHex   :    WORD;
        VisuVarDwordDecimal :    DWORD
        VisuVarDwordHex   :    DWORD;
        VisuVarSintDecimal :    SINT;
        Dummy1          :    SINT;    ...zusätzlich eingerechnetes Byte (wordaligned)
        VisuVarIntDecimal :    INT;
        VisuVarDintDecimal :    DINT;
        VisuVarUsintDecimal :    USINT;
        Dummy2          :    USINT;    ...zusätzlich eingerechnetes Byte (wordaligned)
        VisuVarUintDecimal :    UINT;
        VisuVarUdintDecimal :    UDINT;
        VisuVarRealDecimal :    REAL;
        VisuVarRealDecimal :    REAL;
        VisuVarDateTime :    DT;
    END_STRUCT
END_TYPE
```

7.23 Objekt Trend

#Trend		Darstellung der DataLog-Datei als Trenddiagramm
Text/File	<i>MyTrendDef.TXT</i>	<ul style="list-style-type: none"> Name der Trend-Definitionsdatei (Parameterdatei) mit den Trend-Parametern
Font	<i>Font.FNT</i>	<ul style="list-style-type: none"> optionaler Font für den Text
X,Y,DX,DY	<i>Integerwerte</i>	<ul style="list-style-type: none"> Position, Breite und Höhe des Trends (Bezug links oben!)
Color	<i>Farbname oder Farbnummer</i>	<ul style="list-style-type: none"> Farbe des Textes des Trends
BackColor	<i>Farbname oder Farbnummer</i>	<ul style="list-style-type: none"> Hintergrundfarbe des Trends
Format		<ul style="list-style-type: none">
Action		<ul style="list-style-type: none">
Limit1		<ul style="list-style-type: none">
Limit2		<ul style="list-style-type: none">
Action Limit1 Action Limit2		<ul style="list-style-type: none">
VarValue		<ul style="list-style-type: none">
VarType		<ul style="list-style-type: none">
VarState		<ul style="list-style-type: none">
Option	Type=name	<ul style="list-style-type: none"> Definition des Trend-Types. Damit können mehrere Trend-Objekte mit derselben Trenddatei definiert werden.
C-Function		<ul style="list-style-type: none">



Funktion von Trend

Mit Hilfe des Trend-Objekt können Daten, die mit dem Datalog-Objekt aufgezeichnet wurden, in Form eines Trends dargestellt werden (max. 4 Kurven in einem Trend). Die Werte können hierbei Online und Offline dargestellt werden. Die Manipulation des aktuellen Ausschnitts erfolgt mit Hilfe von Button-Aktionen. (siehe Objekt Button).

Der Trend wird normalerweise als Y/T-Diagramm dargestellt. Mit der Option `format_user=%ld` wird der Wert des Zeitstempels (Doppelwort im Datalog) im Userformat z.B. als Ganzzahl (1,2,3,...) dargestellt. Somit können X/Y-Diagramme z.B. von Temperaturreglern dargestellt werden.

7.23.1 Tabellenblatt Trend

Im Tabellenblatt „Trend“ werden alle objektspezifischen Parameter definiert. Das Tabellenblatt „Trend“ ist wie folgt aufgebaut:

Format		Comment
#Title=		...Diagrammtitle
#Seperator=;		...Seperator in datalogfile
#Orientation=horizontal		...Orientation horizontal or vertical
#Flow=Right2Left		...Flow of Trend Left2Right or Right2Left
#File=datalog.log		...[Path/]Name of the datafile (path optional, without path the PATH_LOG will be used) Filename my contain a system variable e.g. #File=dlog%s_idx%.log
#CursorColor=<color>		...Cursor color (optionally)
#XScaleFgColor=<color>		...XScale foreground color (optionally)
#XScaleBgColor=<color>		...XScale background color (optionally)
#X=[t]		...Name of X-Axis
#format_time=%H:%M:%S		...Time/User Format of the X-axis
#Format_date=%d-%m-%y		...Date Format of the X-axis (should be empty if Format user is selected)
#range=900		...Range of X-Axis in seconds/user
#dt=1		...Sampletime in seconds
#GridX=180		...Lettering Grid on X
#dtTolerance=0		...number of missing datapoints before break the curve
#ScaleX=On		...scale drawing off/on
#y=[C]		...1. Y-Curve (Name,color)
#Type=INT		...IEC Datatype (BYTE, SINT, USINT, WORD, INT, UINT, DWORD, DINT, UDINT, REAL)
#color=green		...curve-color
#ScaleColor=green		...scale-color
#max=150		...Ymax
#min=0		...Ymin
#GridY=15		...Lettering Grid on Trend 1
#Grid=Off		...on = draw horicontal grid lines
#Scale=On		...scale drawing off/on
#DataLogCol=1		...select data column within datalog file
#y=[C]		...2. Y-Curve (Name,color)
#Type=INT		...IEC Datatype (BYTE, SINT, USINT, WORD, INT, UINT, DWORD, DINT, UDINT, REAL)
#color=blue		...curve-color
#ScaleColor=blue		...scale-color
#max=20		...Ymax
#min=0		...Ymin
#GridY=2		...Lettering Grid on Trend 2
#Grid=Off		...on = draw horicontal grid lines
#Scale=On		...scale drawing off/on
#DataLogCol=2		...select data column within datalog file
#y=[bar]		...3. Y-Curve (Name,color)
#Type=REAL		...IEC Datatype (BYTE, SINT, USINT, WORD, INT, UINT, DWORD, DINT, UDINT, REAL)
#color=red		...curve-color
#ScaleColor=red		...scale-color
#max=10		...Ymax
#min=0		...Ymin
#GridY=1		...Lettering Grid on Trend 3
#Grid=Off		...on = draw horicontal grid lines
#Scale=On		...scale drawing off/on
#DataLogCol=3		...select data column within datalog file
#y=[bar]		...4. Y-Curve (Name,color)
#Type=REAL		...IEC Datatype (BYTE, SINT, USINT, WORD, INT, UINT, DWORD, DINT, UDINT, REAL)
#color=brown		...curve-color
#ScaleColor=brown		...scale-color
#max=3		...Ymax

#min=0	...Ymin
#GridY=1	...Lettering Grid on Trend 4
#Grid=Off	...on = draw horicontal grid lines
#Scale=On	...scale drawing off/on
#DataLogCol=4	...select data column within datalog file

7.24 Objekt Sys2Plc

#Sys2Plc		Übertragung von EPAM-Systemvariablen an die Steuerung (Bidirektionale Kommunikation)
Text/File	<i>MySysPlc.TXT</i>	• Name der Sys2Plc-Definitionsdatei
Font		•
X,Y,DX,DY		•
Color		•
BackColor		•
Format		•
Action		•
Limit1		•
Limit2		•
Action Limit1		•
Action Limit2		•
VarValue		•
VarType		•
VarState		•
Option		•
C-Function		•



Funktion des Sys2Plc

Das Sys2Plc-Objekt dient zur Kommunikation der EPAM-Systemvariablen zwischen EPAM und der PLC. Als Datenschnittstelle zwischen EPAM und PLC dient die Sys2Plc-Definitionsdatei.

Verhalten:

Startup: Systemvariablen werden auf PLC-Variablen geschrieben

Betrieb: nach Änderung von Systemvariablen werden diese auf die entsprechenden PLC Variablen geschrieben (ereignisorientiert). Geänderte PLC-Variablen werden zyklisch auf die Systemvariablen geschrieben (**Zyklus: 0.5s**)



Das Objekt Sys2Plc wird global in der ersten Seite im Tabellenblatt „Project“ („Initialisierungs-Seite“) projiziert!

7.24.1 Tabellenblatt *Sys2Plc*

Im Tabellenblatt „Sys2Plc“ werden alle zu übertragenden Variablen (Variablenpaare) angegeben. Das Tabellenblatt „Sys2Plc“ ist wie folgt aufgebaut:

System-Variable	PLC-Variable	
s_msysvar	PLC/myplcvar	
...	...	
...	...	



Die EPAM-Systemvariable und die entsprechend dazugehörige PLC-Variable (Variablenpaar) müssen denselben Datentyp haben.



Bildseitenwechsel in der PLC auslösen

Mit Hilfe des Sys2Plc-Objektes können Sie durch Definition des Variablenpaares EPAM-Systemvariable **s_newpage** und einer beliebigen PLC-Variable z.B. PLC/NewPagename Bildseitenwechsel auslösen. Wird im PLC-Projekt beispielsweise die Variable PLC/NewPagename mit dem Wert '#Page=*name*' gesetzt, wird anschliessend der Wert '#Page=*name*' an die EPAM-Systemvariable übertragen und es erfolgt der Bildseitenwechsel.

7.25 Objekt RemoteControl

#RemoteControl		Fernbedienung von vernetzten EPAM-Applikationen
Text/File	xxx.xxx.xxx.xxx	<ul style="list-style-type: none"> IP-Adresse der Remote-Bedieneinheit
Font	Font.FNT	<ul style="list-style-type: none"> optionaler Font für den Text
X,Y,DX,DY	Integerwerte	<ul style="list-style-type: none"> Position, Breite und Höhe des RemoteControls (Bezug links oben!)
Color	Farbname oder Farbnummer	<ul style="list-style-type: none"> Farbe des Textes des RemoteControls
BackColor	Farbname oder Farbnummer	<ul style="list-style-type: none"> Hintergrundfarbe des RemoteControls
Format		<ul style="list-style-type: none">
Action		<ul style="list-style-type: none">
Limit1		<ul style="list-style-type: none">
Limit2		<ul style="list-style-type: none">
Action Limit1 Action Limit2		<ul style="list-style-type: none">
VarValue	Variablenname	<ul style="list-style-type: none"> PLC- oder Systemvariable mit aktueller IP-Adresse der Remote-Bedieneinheit
VarType	STRING	<ul style="list-style-type: none"> Datentyp STRING (IP-Adresse: xxx.xxx.xxx.xxx)
VarState		<ul style="list-style-type: none">
Option	Password=myPassword PROTO_NAME=RFB PROTO_MAJOR=3 PROTO_MINOR=3 PROTO_PORT=5900	<ul style="list-style-type: none"> Passwort für RemoteControl-Server (im Klartext) Optimale Einstellungen für Standard VNC Protokoll
C-Function		<ul style="list-style-type: none">



Funktion des RemoteControl

Das RemoteControl-Objekt dient zur Fernbedienung von mehreren über Ethernet vernetzten EPAM-Applikationen. Hierbei kann z.B. von einer Hauptvisualisierung aus eine Bedieneinheit, die an einem anderen Anlagenteil steht, fernbedient werden. Das RemoteControl-Objekt stellt hierbei den Rahmen dar indem das aktuelle Bild der Remote-Bedieneinheit dargestellt wird. Eingaben z.B. mit dem Touch werden ebenfalls an die Remote-Bedienstation übertragen und wirken wie eine lokale Bedienung an der Remote-Bedienstation. Umgekehrt werden nur Änderungen des Bildinhaltes der Remote-Bedienstation übertragen (vgl. PC-Anyware).



Remote-Control von gleich grossen Bildschirmen

Soll von einem Gerät aus via RemoteControl-Objekt auf ein anderes Gerät mit gleicher Auflösung (z.B. 320x240) zugegriffen werden (Fullscreen), so muss auf dem Gerät ein Invisible-Button mit der Aktion #Page=xy und einem Timeout projiziert werden (z.B. rechts unten, DX, DY 1 Pixel). Damit wird die Remote-Bedienung nach dem definierten Timeout ohne Bedienung automatisch wieder beendet.

Andernfalls muss das Gerät mit dem RemoteControl-Objekt eine höhere Auflösung (grösserer Bildschirm) als das fernbediente Gerät haben. Z.B. in einem VGA-Gerät (640x480) wird der Bildschirm eines 1/4VGA-Gerätes (320x240) dargestellt.



Systemvariablen

Nachfolgende Systemvariablen werden im Zusammenhang mit dem RemoteControl-Objekt verwendet:

s_rcinput_enabled	...zeigt den aktuellen Status 1=Eingabe erlaubt, 0=Eingabe disabled
s_rc_password	...Password für RemoteControl-Server (alternative zur Option Password=)
s_remoteclient_connected	...zeigt auf dem Remote-Geräte (mit dem RemoteControl-Server) an, ob ein Client verbunden ist (1 = RemoteControl-Client verbunden)

8 Application Notes

8.1 Alarmhandling

8.1.1 Alarmdefinition

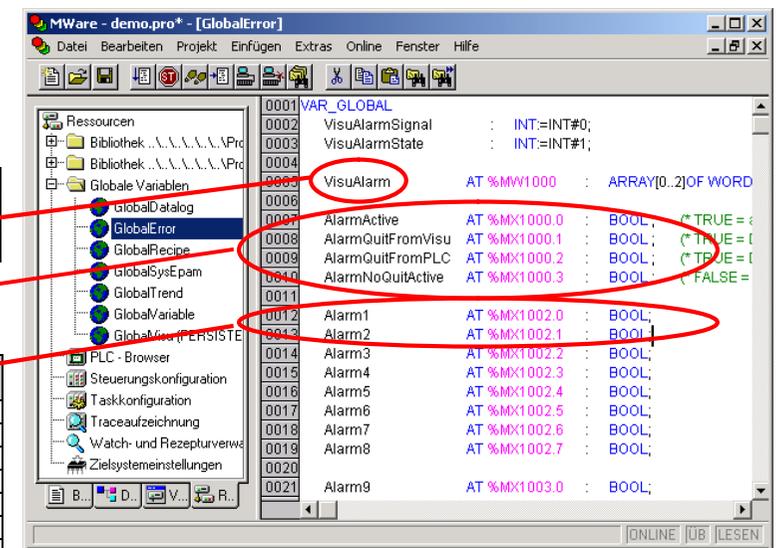
Globales EPAM-Objekt #Alarm

Object	Text/File	Font	X [Pixel]	Y [Pixel]	DX [Pixel]	DY [Pixel]	Color	Backcolor	Format	VarValue	VarType
#Page=Init			0	0	0	0	Black	white			
#Alarm	MyAlarm1.TXT									PLC/VisuAlarm	STRING
#Page=Production			0	0	320	240	black	white			

Alarmdefinition im Tabellenlatt MyAlarm1

Variable	Alarm number	Text/File	Font	Color	Backcolor	Action Alarm Info	Alarm Helptext
[0].0	1	1 = Alarm(s) activ					
[0].1	1	1 = Delete alarm request from visualization					
[0].2	1	1 = Delete alarm request from PLC					
[0].3	0	0 = all Alarm(s)					
[0].4 [0].15		reserved					
[1].0	1	Beliebiger Alarmtext 1	Arial8.FNT	black	red	#Page=Diagnose,#Page=Foto1	Alrmhlp1.txt
[1].1	2	Beliebiger Alarmtext 2	Arial8.FNT	black	red	#Page=Diagnose,#Page=Foto1	Alrmhlp1.txt
...
[1].15	16	Beliebiger Alarmtext 16	Arial8.FNT	black	yellow	#Page=Diagnose,#Page=Foto2	Alrmhlp2.txt
[2].0	17	Beliebiger Alarmtext 17	Arial8.FNT	black	yellow	#Page=Diagnose,#Page=Foto2	Alrmhlp3.txt
...
[2].14	31	Beliebiger Alarmtext 31	Arial8.FNT	black	white	#Page=Diagnose,#Page=Foto3	...
[2].15	32	Beliebiger Alarmtext 32	Arial8.FNT	black	white

IEC61131 PLC-Alarmbuffer „VisuAlarm“



Alarmdefinition beinhaltet:

- Alarmbit → Alarmnummer
- Text, Font, Farbdefinition
- Bildseiten zu Diagnosezwecken
- Hilfetext

Funktionsweise

Globales Alarmobjekt prüft PLC-Alarmbuffer zyklisch (IEC61131 Variable VisuAlarm)

Eingehende Alarme (Kommen)

- Neu eingegangene Alarme $[x].x = 1$ (IEC61131 Variablen Alarm1..32) werden mit Kommen-Zeitstempel gekennzeichnet und in den internen EPAM-Alarmbuffer eingetragen (Alarmhistory 512 Alarme)
- Statusbit [0].0 (IEC61131 Variable AlarmActive) und [0].3 (IEC61131 Variable AlarmNoQuitActive) werden von EPAM gesetzt

Austretende Alarme (Gehen)

- Bei austretenden Alarmen $[x].x = 0$ (IEC61131 Variablen Alarm1..32) wird in den internen EPAM-Alarmbuffer der Gehen-Zeitstempel eingetragen

Quittieren der Alarme

- Bei Alarmquittierung über EPAM wird in den internen EPAM-Alarmbuffer der Quittierungs-Zeitstempel eingetragen und das Statusbit [0].1 (IEC61131 Variable AlarmQuitFromVisu) von EPAM gesetzt.
- Bei Alarmquittierung über die PLC wird das Statusbit [0].2 (IEC61131 Variable AlarmQuitFromPLC) in der PLC gesetzt. Bei sämtlichen Alarmen wird in dem internen EPAM-Alarmbuffer der Quittierungs-Zeitstempel eingetragen
- Das Statusbit [0].3 (IEC61131 Variable AlarmNoQuitActive) bleibt aktiv bis alle Alarme quittiert wurden

8.1.2 Alarmdarstellung

Alarmdefinition im Tabellenlatt MyAlarm1

Variable	Alarm number	Text/File	Font	Color	Backcolor	Action Alarm Info	Alarm Helptext
[0].0		1 = Alarm(s) activ					
[0].1		1 = Delete alarm request from visualization					
[0].2		1 = Delete alarm request from PLC					
[0].3		0 = all Alarm(s)					
[0].4-[0].15		reserved					
[1].0	1	Beliebiger Alarmtext 1	Arial8.FNT	black	red	#Page=Diagnose,#Page=Foto1	Alrmhlp1.txt
[1].1	2	Beliebiger Alarmtext 2	Arial8.FNT	black	red	#Page=Diagnose,#Page=Foto1	Alrmhlp...
...
[1].15	16	Beliebiger Alarmtext 16	Arial8.FNT	black	yellow	#Page=Diagnose,#Page=Foto2	Alrmhlp2.txt
[2].0	17	Beliebiger Alarmtext 17	Arial8.FNT	black	yellow	#Page=Diagnose,#Page=Foto2	Alrmhlp3.txt
...
[2].14	31	Beliebiger Alarmtext 31	Arial8.FNT	black	white	#Page=Diagnose,#Page=Foto3	...
[2].15	32	Beliebiger Alarmtext 32	Arial8.FNT	black	white		

IEC61131 PLC-Alarmbuffer „VisuAlarm“

Alarm1 := TRUE
Alarm2 := TRUE

Alarmdefiniton beinhaltet:

- Alarmbit → Alarmnummer
- Text, Font, Farbdefinition
- Bildseiten zu Diagnosezwecken
- Hilfetext

EPAM-Objekt #AlarmList

Object	Text/File	Font	X [Pixel]	Y [Pixel]	DX [Pixel]	DY [Pixel]	Color	Backcolor	Format	VarValue	VarType	Option
#Page=AlarmList			0	0	320	240	black	white				
#Alarmlist	No=%4d,sep=SY\$08X16.FNT		0	0	320	160	red	white	...			DX=40,alarmfilter=activ/notquit,alarmsort=lifo, AlarmType=MyAlarm1

Formatierung der Alarmzeile
Alarm Nr. Alarmtext



Formatierung der Alarmliste

Funktionsweise

Das Objekt AlarmList stellt den internen EPAM-Alarmbuffer als Liste dar

Darstellung des Alarmbuffers in der Liste

- Der aktuell selektierte Alarm wird invers dargestellt
- Die Darstellung der Alarmliste (Sortierung, Filter) kann durch Optionen vordefiniert werden und allenfalls über Buttonaktionen verändert werden
- Die aktuellen Alarminformationen des selektierten Alarms sind in den Systemvariablen s_alarm_nr, s_alarm_text, s_alarm_tin, s_alarm_tout, s_alarm_tquit, s_alarm_info, s_alarm_txtinfo hinterlegt (Alarmdiagnose)

Quittieren der Alarme

- Alarme können über die Buttonaktion AlarmQuit einzeln quittiert werden bzw. über die Buttonaktion AlarmQuitAll gesamthaft quittiert werden

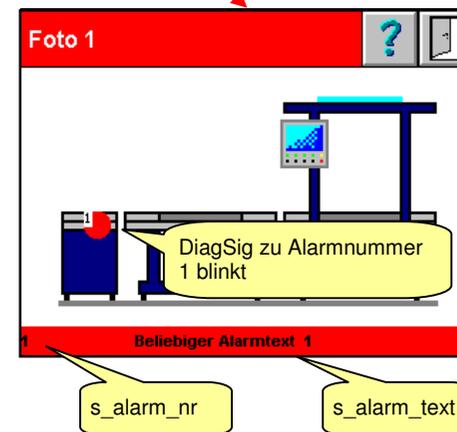
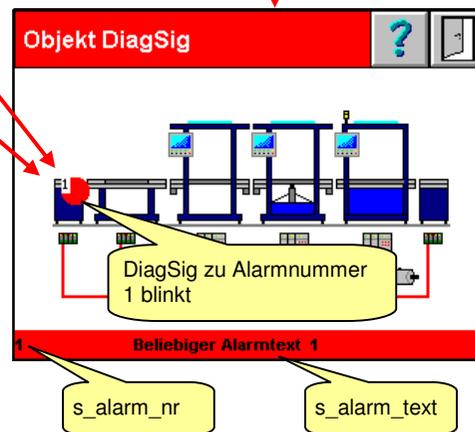
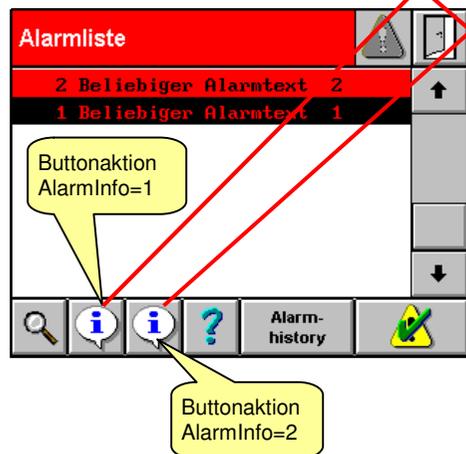
8.1.3 Alarmdiagnose

Alarmdefinition im Tabellenlatt MyAlarm1

Variable	Alarm number	Text/File	Font	Color	Backcolor	Action Alarm Info	Alarm Helptext
[0].0		1 = Alarm(s) activ					
[0].1		1 = Delete alarm request from visualization					
[0].2		1 = Delete alarm request from PLC					
[0].3		0 = all Alarm(s)					
[0].4-[0].15		reserved					
[1].0	1	Beliebiger Alarmtext 1	Arial8.FNT	black	red	{Page=Diagnose,#Page=Foto1	alarmhlp1.txt
[1].1	2	Beliebiger Alarmtext 2	Arial8.FNT	black	red	{Page=Diagnose,#Page=Foto1	alarmhlp1.txt
...
[1].15	16	Beliebiger Alarmtext 16	Arial8.FNT	black	yellow	{Page=Diagnose,#Page=Foto2	Alarmhlp2.txt
[2].0	17	Beliebiger Alarmtext 17	Arial8.FNT	black	yellow	{Page=Diagnose,#Page=Foto2	Alarmhlp3.txt
...
[2].14	31	Beliebiger Alarmtext 31	Arial8.FNT	black	white	{Page=Diagnose,#Page=Foto3	...
[2].15	32	Beliebiger Alarmtext 32	Arial8.FNT	black	white	{Page=Diagnose,#Page=Foto3	...

EPAM-Objekt #DiagSig

Object	Text/File	Font	X [Pixel]	Y [Pixel]	DX [Pixel]	DY [Pixel]	Color	Backcolor	Format	VarValue	VarType
#Page=ObjectDiagSig			0	0	320	240	Black	white			
#Signal	Machine_pos		25	60	270	140	black	white	...		
#DiagSig	1		32	115	20	20	red		...	s_alarm_nr	STRING
#Variable	%4d		...	220	80	20	black	red	...	s_alarm_nr	WORD
#Variable	%s		...	80	220	20	black	red	...	s_alarm_text	WORD

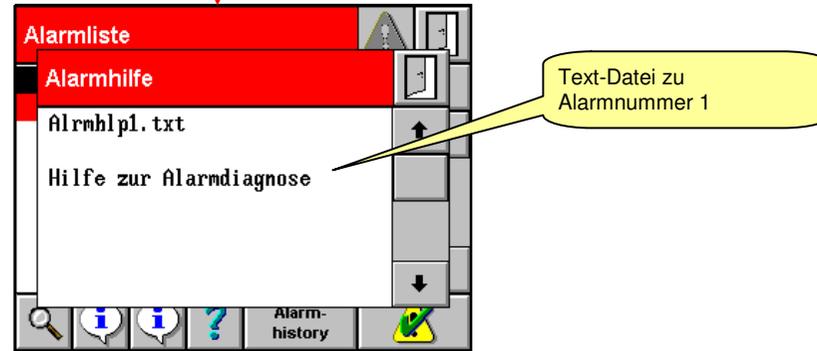
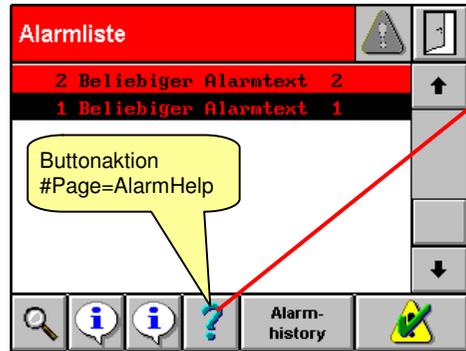


Alarmdefinition im Tabellenlatt MyAlarm1

Variable	Alarm number	Text/File	Font	Color	Backcolor	Action Alarm Info	Alarm Helptext
[0].0		1 = Alarm(s) activ					
[0].1		1 = Delete alarm request from visualization					
[0].2		1 = Delete alarm request from PLC					
[0].3		0 = all Alarm(s)					
[0].4-[0].15		reserved					
[1].0	1	Beliebiger Alarmtext 1	Arial8.FNT	black	red	#Page=Diagnose,#Page=Foto1	Alarmhlp1.txt
[1].1	2	Beliebiger Alarmtext 2	Arial8.FNT	black	red	#Page=Diagnose,#Page=Foto1	Alarmhlp1.txt
...
[1].15	16	Beliebiger Alarmtext 16	Arial8.FNT	black	yellow	#Page=Diagnose,#Page=Foto2	Alarmhlp2.txt
[2].0	17	Beliebiger Alarmtext 17	Arial8.FNT	black	yellow	#Page=Diagnose,#Page=Foto2	Alarmhlp3.txt
...
[2].14	31	Beliebiger Alarmtext 31	Arial8.FNT	black	white	#Page=Diagnose,#Page=Foto3	
[2].15	32	Beliebiger Alarmtext 32	Arial8.FNT	black	white		

EPAM-Objekt #TextList / Variable s_alarm_txtinfo

Object	Text/File	Font	X [Pixel]	Y [Pixel]	DX [Pixel]	DY [Pixel]	Color	Backcolor	Format	VarValue	VarType
#Page=AlarmHelp			15	30	290	180	Black	white	...		
...		
#TextList	s_alarm_txtinfo...		0	40	290	140	black	white	...		STRING



Funktionsweise

Buttonaktion AlarmInfo=1

- Anhand des in der Alarmliste selektierten Alarms wird mit der Aktion AlarmInfo=1 auf die in der Alarmdefinition Tabellenblatt „MyAlarm1“ unter Action Alarm Info projizierte Bildseite gesprungen

Buttonaktion AlarmInfo=2

- Anhand des in der Alarmliste selektierten Alarms wird mit der Aktion AlarmInfo=2 auf die in der Alarmdefinition Tabellenblatt „MyAlarm1“ unter Action Alarm Info projizierte Bildseite gesprungen

Textliste mit Variable s_alarm_txtinfo

Auf einer beliebigen Bildseite kann eine Textliste mit der in Spalte Text/File eingetragenen Systemvariable s_alarm_txtinfo projiziert werden. Anhand des in der Alarmliste selektierten Alarms wird mit einer Aktion #page=name auf die Bildseite mit der projizierten Textliste gesprungen. Nun wird in dieser Textliste die im Tabellenblatt „MyAlarm1“ unter Alarm Helptext projizierte Textdatei eingeblendet.

8.2 Rezepthandling

8.2.1 Rezeptdefinition

Globales EPAM-Objekt #Recipe

Object	Text/File	Font	X [Pixel]	Y [Pixel]	DX [Pixel]	DY [Pixel]	Color	BackColor	Format	Action	VarValue
#Page=Init			0	0	0	0	Black	white			
#Recipe	MyTyp1.TXT									#Page=MyTyp1Overwrite	PLC/VisuMyTyp1UpDnle
#Page=Production			0	0	320	240	black	white			

IEC61131 PLC Rezeptstatus-Variable:

- 1 ...Download-Anforderung von PLC
- 2 ...Up/Download beendet
- 3 ...Upload Anforderung von PLC
- 4 ...Download läuft
- 5 ...Upload läuft

Rezeptdefinition im Tabellenblatt MyAlarm1

Variable	VarType	Value
#Recipe=Default		Path=
PLC/VisuAx1.mmild	INT	
PLC/VisuAx1.mmiFunc	SINT	
PLC/VisuAx1.mmiNomVel	DINT	
PLC/VisuAx1.mmiNomAcc	DINT	
PLC/VisuAx1.mmiScaleN	INT	
PLC/VisuAx1.mmiScaleM	INT	
PLC/VisuAx1.mmiStopDec	INT	
PLC/VisuAx1.mmiNullPos	DINT	
PLC/VisuAx1.mmiVelKpRed	DINT	
PLC/VisuAx1.mmiCamTable	INT	
...	...	
#Checksum=		

Bestätigungs-Bildseite:

Diese Seite wird aufgerufen, wenn eine mit der Aktion **csave=MyTyp1** zu speichernde Rezeptur-Datei bereits existiert. (Auf dieser Bildseite kann eine Abfrage „Datei überschreiben“ projiziert werden)

Rezept-Optionen:

- Keine Option: aktuelle Rezeptwerte werden beim Start von EPAM in die PLC geschrieben (ACTUAL.DAT)
- NoActual: Beim Start von EPAM werden die Werte der zuletzt geladenen Rezeptur in die PLC geschrieben
- NoDownload: Rezeptwerte werden beim Start von EPAM nicht in die PLC geschrieben

Funktionsweise

- Globales Rezeptobjekt mit PLC Rezeptstatus-Variable (IEC61131 Variable VisuMyTyp1UpDnload)
- Die dazugehörigen Systemvariablen **s_MyTyp1_file** und **s_MyTyp1_name** behinhalten Dateinamen (ISO 9669, 8.3 mit eingeschränktem Zeichensatz) bzw. den Rezeptnamen (80 Zeichen) einer Rezeptur.
- Mit den Button-Aktionen **load=MyTyp1**, **save=MyTyp1**, **csave=MyTyp1** und **delete=MyTyp1** wird die in der Systemvariable **s_MyTyp1_file** definierte Rezeptur geladen, geschrieben bzw. mit Abfrage (#Page=MyTyp1RecipeOverwrite) überschrieben oder gelöscht.
- Rezepturen werden mit **xxxxx.DAT** (x = Inhalt de Variable **s_MyTyp1_file**) ins Verzeichnis **C:\DATA\MyTyp1** geschrieben.

8.2.2 Rezeptliste

Globales EPAM-Objekt #Recipe

Object	Text/File	Font	X [Pixel]	Y [Pixel]	DX [Pixel]	DY [Pixel]	Color	Backcolor	Format	Action	VarValue	VarType	VarState	Option
#Page=Init			0	0	0	0	Black	white						
#Recipe	MyTyp1.TXT									#Page=MyTyp1Overwrite	PLC/VisuMyTyp1UpDnload	INT		
#Page=Production			0	0	320	240	black	white						

EPAM-Objekt #RecipeList

Object	Text/File	Font	X [Pixel]	Y [Pixel]	DX [Pixel]	DY [Pixel]	Color	Backcolor	Format	Action	VarValue	VarType	Option
#Page=MyTyp1RecipeHandling			0	0	320	240	black	white					
#Recipelist	%-9f %-16n %d-%m-%Y %H:%M	SYS06X11.FNT	1	40	318	110	black	white					
#Signal	Name:	ARIAL10F.FNT	1	155	114	20	black	grey					
#Variable	%8s	ARIAL10F.FNT	1	175	114	25	black	yellow		#Page=KeybAlnum	S MyTyp1_file	STRING	
#Signal	Beschreibung:	ARIAL10F.FNT	119	155	200	20	black	grey					
#Variable	%15s	ARIAL10F.FNT	119	175	199	25	black	yellow		#Page=KeybAlnum	S MyTyp1_name	STRING	
#Button	save.ico		0	200	80	40	black	grey		Csave=MyTyp1			
#Button	open.ico		80	200	80	40	black	grey		#Page=MyTyp1RecipeLoadQuit			
#Button	trash.ico		160	200	80	40	black	grey		#Page=MyTyp1RecipeDeleteQuit			
#Button	sort.ico		240	200	80	40	black	grey		#Page=RecipeListSort			

Formatierung der Rezeptliste
Dateiname, Beschreibung, Datum

Buttons and actions shown in the interface:

- Buttonaktion #Page=MyTyp1RecipeLoadQuit
- Buttonaktion Csave=MyTyp1 prüft ob die Datei R2.DAT bereits vorhanden ist:
 - Nicht vorhanden => speichern
 - Vorhanden => Wechsel auf Bildseite #Page=MyTyp1Overwrite
- Buttonaktion Delete=List löscht die in der Rezeptliste selektierte Rezeptur R1.DAT
- Buttonaktion Load=list lädt die in der Rezepturliste selektierte Rezeptur R1.DAT
- Buttonaktion Save=MyTyp1 speichert die Datei R2.DAT

9 Systemvariablen

Systemvariablen sind globale Variablen die nur innerhalb der Visualisierung benötigt werden. Systemvariablen beginnen mit dem Präfix 's_name'. Folgende Systemvariablen sind vordefiniert:

Systemvariable	Bedeutung	Datentyp
Systemvariablen in Verbindung mit Objekt Alarm/Alarmlist		
s_alarm_active	Variable wird gesetzt, wenn Alarme aktiv sind	INT
s_alarm_info	Name der projektierten Bildseite der Aktion Alarminfo des zuletzt selektierten Alarmes in der Alarmliste	STRING
s_alarm_nr	Alarmnummer des zuletzt selektierten Alarmes in der Alarmliste	WORD
s_alarm_text	Alarmtext des zuletzt selektierten Alarmes in der Alarmliste	STRING
s_alarm_tin	Zeit Alarm „Kommen“ des zuletzt selektierten Alarmes in der Alarmliste	STRING
s_alarm_tout	Zeit Alarm „Gehen“ des zuletzt selektierten Alarmes in der Alarmliste	STRING
s_alarm_tquit	Zeit Alarm „Quittiert“ des zuletzt selektierten Alarmes in der Alarmliste	STRING
s_alarm_tin_dt	Zeit Alarm „Kommen“ des zuletzt selektierten Alarmes in der Alarmliste	IEC_DT
s_alarm_tout_dt	Zeit Alarm „Gehen“ des zuletzt selektierten Alarmes in der Alarmliste	IEC_DT
s_alarm_tquit_dt	Zeit Alarm „Quittiert“ des zuletzt selektierten Alarmes in der Alarmliste	IEC_DT
s_alarm_txtinfo	Variable mit dem Namen der ASCII-Textdatei mit der alarmspezifischen Textinformation (in Verwendung mit Textliste)	STRING
s_alarm_type	Variable mit dem Namen des Alarmtyps (bei Verwendung mehrerer Alarmobjekte)	STRING
s_alarm[myalarm].active_count	Anzahl aktiver Alarme von Alarmtype „myalarm“	WORD
Systemvariablen in Verbindung mit Objekt Recipe/Recipelist		
s_myrecipetype_file	Rezeptdatei für jeden definierten Rezepturtyp (ohne Extension)	STRING
s_myrecipetype_name	Rezeptname für jeden definierten Rezepturtyp	STRING
s_myrecipetype_cur_file	Aktuell in der Rezeptliste selektierte Rezeptdatei für jeden definierten Rezepturtyp (ohne Extension)	STRING
s_myrecipetype_cur_name	Aktuell in der Rezeptliste selektierter Rezeptname für jeden definierten Rezepturtyp	STRING
s_myrecipetype_dnload_max	Anzahl Rezeptvariablen des entsprechenden Rezepturtyps für Download (Die Systemvariable wird für jeden definierten Rezepturtyp angelegt)	WORD
s_myrecipetype_dnload_act	Aktuelle Anzahl geladener Rezepturvariablen für Download (Die Systemvariable wird für jeden definierten Rezepturtyp angelegt)	WORD
s_myrecipetype_upload_max	Anzahl Rezeptvariablen des entsprechenden Rezepturtyps für Upload (Die Systemvariable wird für jeden definierten Rezepturtyp angelegt)	WORD
s_myrecipetype_upload_act	Aktuelle Anzahl geladener Rezepturvariablen für (Die Systemvariable wird für jeden definierten Rezepturtyp angelegt)	WORD
s_recipe_type	Aktuell gewählter Rezepttyp	STRING
s_recipe_path	Aktueller Verzeichnis-Pfad aller der Rezepturen	STRING
s_recipelist_empty	1 bei leerer Rezeptliste, 0 mind. 1 Rezept in der Liste	INT
Systemvariablen in Verbindung mit Objekt Trend		
s_mytrend_c1	Aktueller Wert bei Cursorposition	wie Trend
s_mytrend_c2	Aktueller Wert bei Cursorposition	wie Trend
s_mytrend_c3	Aktueller Wert bei Cursorposition	wie Trend
s_mytrend_c4	Aktueller Wert bei Cursorposition	wie Trend
s_trend_t_sec	Zeit bei Cursorposition (Sekunden)	DWORD
s_trend_t_min	Zeit bei Cursorposition (Minuten)	DWORD
s_trend_t_hour	Zeit bei Cursorposition (Stunden)	DWORD
s_trend_t_mday	Zeit bei Cursorposition (Tag)	DWORD
s_trend_t_mon	Zeit bei Cursorposition (Monat)	DWORD
s_trend_t_year	Zeit bei Cursorposition (Jahr seit 1900)	DWORD

s_trend_t_wday	Zeit bei Cursorposition (Wochentag)	DWORD
s_trend_t_sec	Zeit bei Cursorposition (Minuten)	DWORD
s_trend_t	„Rohwert“ der X-Position	DWORD
Systemvariablen in Verbindung mit Objekt Passwort		
s_password	Aktuelle Passwordeingabe	STRING
s_pwl	Aktueller Passwortlevel	WORD
s_pwl_required	Aktuelle benötigter Passwortlevel	WORD
s_password_x	Definiertes Passwort für Berechtigungsstufe x	STRING
s_user_x	Definiertes Username für Berechtigungsstufe x	STRING
Systemvariablen in Verbindung mit dem Objekt Page		
s_newpage	Neuer Bildseitenname Durch Setzen dieser Variable, erfolgt ein Bildseitenwechsel auf Seite s_newpage	STRING
s_pageidx	Aktuelle Bildseiten-ID. Ohne Option ID ist der Wert 0 (in Verbindung mit Option ID)	UINT
s_pageid_last	Aktuelle Bildseiten-ID der Page mit ID > 0 (in Verbindung mit Option ID)	UINT
s_pagename	Aktuelle Bildseitenname	STRING
Allgemeine Systemvariablen		
s_backlight	Aktuelle Einstellung der Hintergrundbeleuchtung (0-100%, Default: 100%)	WORD
s_contrast	Aktuelle Einstellung des Kontrast (0-100%, Default 50%) Nur passiv LCDs!	WORD
s_dns1_ip	DNS1 Adresse des Zielsystems (Eingabe erfolgt im Format xxx.xxx.xxx.xxx)	STRING
s_dns2_ip	DNS2 Adresse des Zielsystems (Eingabe erfolgt im Format xxx.xxx.xxx.xxx)	STRING
s_dhcp_mode	0 = DHCP disabled, statische IP Adresse 1 = DHCP enabled	INT
s_irtouch	1 bei IR-Touchscreen; 0 andere	INT
s_rcinput_enabled	1 RemoteControl Input enabled, 0 disabled	INT
s_rc_password	Passwort für RemoteControl-Server	STRING
s_remoteclient_connected	1 wenn Zugriff via RemoteClient aktiv	INT
s_target_ip	Aktuelle IP-Adresse des Zielsystems (Eingabe erfolgt im Format xxx.xxx.xxx.xxx)	STRING
s_toucherror	0 o.k. 1 Touchtest Error (nur IR-Touch)	INT
s_gateway_ip	Aktuelle IP-Adresse des Gateways (Eingabe erfolgt im Format xxx.xxx.xxx.xxx)	STRING
s_subnetmask	Aktuelle Subnet-Maske des Zielsystems (Eingabe erfolgt im IP-Format xxx.xxx.xxx.xxx)	STRING
s_language	Aktuelle Sprache	STRING
s_plcstate	Aktueller Status (1=Stop, 0=Run) der lokalen Steuerung (nur VxWorks → sollte nicht mehr verwendet werden!)	WORD
s_plcstate_<hostname>	Aktueller Status der (remote) Steuerung <hostname> 0=undefiniert 1=Run 2=Stop	WORD
s_unit_idx	Aktuelles Mass-System 0, 1, ...	WORD
Systemvariablen in Verbindung mit Werteingaben		
s_input_val	Aktueller Eingabewert	STRING
s_edit_val	Zuletzt eingegebener Wert	STRING
s_limit1	Aktueller unterer Grenzwert	STRING
s_limit2	Aktueller oberer Grenzwert	STRING
s_helptext	Aktuelle Helptext-Nummer	WORD
Systemvariablen in Verbindung mit Datum/Zeit Ein-und Ausgaben		
s_tm_day	Tag (1-31)	WORD
s_tm_mon	Monat (1-12)	WORD

s_tm_year	Jahr (1980-2099)	WORD
s_tm_hour	Stunden (00-23)	WORD
s_tm_min	Minuten (0-59)	WORD
s_tm_sec	Ist-Sekunden (00-59) für Anzeige	WORD
s_tm_nsec	nominal-Sekunden (00-59) für Eingabe	WORD
s_tm_wday	Wochentag (0-6; 0 = Sonntag)	WORD
s_tm_yday	Jahrestag (0-365; 0 = 1.Jänner)	WORD
s_tm_isdst	s_tm_isdst > 0 ...Sommerzeit s_tm_isdst 0 ...Normalzeit s_tm_isdst < 0 ...Information steht nicht zur Verfügung	WORD
Systemvariablen in Verbindung mit der Projektierung		
s_epam_date	Aktuelle EPAM-Version	STRING
s_epam_version	Aktuelles EPAM-Datum (Erstellungsdatum)	STRING
s_projectname	Aktueller Projektname	STRING
s_projectprogrammer	Aktueller Projektprogrammierer	STRING
s_projecttarget	Aktuelles Projekt-Zielsystem	STRING
s_projectversion	Aktuelle Projektversion	STRING
Systemvariablen in Verbindung mit der Datenbank (EPAM-DB)		
s_dbconnection	DB Verbindungs-Status: 0 = nicht verbunden; 1 = verbunden	INT
s_dbpasswd_change_err	DB Passwort-Änderungs-Status: 0 = o.k. 1 = user nicht eingelogged 2 = neues Passwort1 ungültig 3 = neues Passwort2 ungültig 4 = altes Passwort ungültig 5 = Passwort 1 ungleich Passwort2	INT
s_dbpasswd_expires	DB Passwort, Tage bis zum Ablauf (Defaultwert = 0xFFFFFFFF)	DWORD
s_dbpasswd_login_err	DB Login Status: 0 = o.k. 1 = Username ungültig 2 = Passwort ungültig 3 = Userstatus ungültig (nicht aktiv) 4 = User abgelaufen 5 = Passwort abgelaufen 6 = zu viele ungültige Logins	INT
s_dbpasswd_name	DB Username	STRING
s_dbpasswd_pw	DB Passwort	STRING
s_dbpasswd_pw1	DB neues Passwort	STRING
s_dbpasswd_pw2	DB neues Passwort Wiederholung	STRING
s_varlog_comment	Kommentar in DB bei Variablen-Änderung	STRING

Speicherung von Systemvariablen



Systemvariablen werden in der Datei 'SYSVAR.INI' bei folgenden Aktionen netzausfallsicher gespeichert und beim nächsten Start von EPAM wieder geladen:

- Sprachumschaltung
- Rezeptur laden, speichern, löschen
- Button-Aktion: save=SysVar

10 Fehlermeldungen

Nachfolgend finden Sie eine Liste der möglichen EPAM Fehlermeldungen die zur Laufzeit auftreten können. Fehlermeldungen mit dem Text „Error in Line xxx“ beziehen sich auf die entsprechende Zeilen in der Excel-Datei:

Fehlermeldung	Bedeutung	mögliche Ursache/Abhilfe
Exit programm ???	EPAM soll verlassen werden	Taste ESC wurde gedrückt
EPAM Demoversion Will exit after 1 hour	EPAM Demoversion installiert. Die Applikation hat eine limitierte Laufzeit von einer Stunde!	Für die Vollversion wird ein Produktcode benötigt.
Error in <i>Line</i> File not found <i>Filename</i>	die in Zeile <i>Line</i> definierte Datei <i>Filename</i> konnte nicht geöffnet werden	Datei ist bereits geöffnet (z.B. durch eine andere Anwendung). Datei ist nicht im aktuellen Projektverzeichnis
Error in <i>Line</i> File read error <i>Filename</i>	Datei <i>Filename</i> konnte nicht korrekt gelesen werden	Datei unvollständig oder defekt
Error in <i>Line</i> File write error <i>Filename</i>	Datei <i>Filename</i> konnte nicht korrekt geschrieben werden	zu wenig freier Speicherplatz auf dem Disk
No more dynamic memory (Heap) <i>Line</i>	kein dynamischer Speicher verfügbar in Programmzeile <i>Line</i>	zu wenig freier Speicher
Error in <i>Line</i> Unknown object <i>#Object</i>	<i>#Object</i> existiert nicht	Syntaxfehler
Error in <i>Line</i> Page not found <i>Page</i>	Bildseitenwechsel auf nicht vorhandene Bildseite	Syntaxfehler oder Bildseite nicht vorhanden
Systemerror: Heapcheck	Systemfehler: dynamischer Speicherbereich nicht mehr konsistent	Rechner neu starten, bei erneutem Fehler technischen Support kontaktieren
No page in textfile	Scriptdatei enthält keine Bildseite	keine oder ungültige Skriptdatei
Systemerror in <i>line</i> xxx: Nullpointer	Systemfehler in Programmzeile <i>Line</i> : Nullpointer	Programmzeile notieren und technischen Support kontaktieren
Error in <i>Line</i> Action not allowed Action	Aktion Action nicht erlaubt	Syntaxfehler bzw. Aktion nicht unterstützt
Error in <i>Line</i> No action defined	Aktion fehlt	Spalte Aktion enthält Space(s)
Error in <i>Line</i> Limit not allowed	ungültiger Grenzwert	Syntaxfehler oder Datentypkonflikt
Error in <i>Line</i> Variabletype not allowed <i>Vartyp</i>	ungültiger Variablentyp <i>Vartyp</i>	Syntaxfehler
Error in <i>Line</i> Limit action not allowed <i>action</i>	ungültige Limit Aktion <i>action</i>	Syntaxfehler bzw. Aktion nicht unterstützt
Error in <i>Line</i> SetVar action not allowed <i>action</i>	ungültige SetVar Aktion <i>action</i>	Syntaxfehler bzw. Aktion nicht unterstützt
Error in <i>Line</i> Value not allowed <i>value</i>	ungültiger Variablenwert <i>value</i>	Syntaxfehler oder Datentypkonflikt z.B. String statt Wert
Error in <i>Line</i> Unknown key <i>Key</i>	ungültige Taste <i>Key</i>	siehe Standard Tastaturtabelle
Error in <i>Line</i> No key defined	Taste fehlt	Aktion 'Key=' ohne Tastencode
Error in <i>Line</i> Unknown color definition <i>color</i>	ungültige Farb- bzw. Hintergrundfarbdefinition <i>color</i>	Farbname bzw. Nummer ungültig (siehe Standard Farbpalette)
To many system variables	max. Anzahl Systemvariablen überschritten	mehr als 1024 Systemvariablen definiert
Not enough memory for system variables <i>Name</i>	zu wenig dynamischer Speicher für Systemvariable <i>Name</i>	siehe „No more dynamic memory (Heap)“
Alarmnumber not allowed	ungültige Alarmnummer	Alarmnummer muss eine Ganzzahl sein
To many alarms defined	mehr als 1008 Alarme definiert	
Error in <i>Line</i> Unknown PCX format <i>Filename</i>	PCX-Dateiformat passt nicht zu aktueller Videoauflösung	z.B. wurde versucht im 16-Farben VGA-Mode ein 256 Farben PCX-Bild anzuzeigen
Inconsistent order in language file <i>Filename line</i> xxx	Sprachdatei und Scriptdatei sind inkonsistent	Zeilennummer stimmt nicht überein (in Scriptdatei oder der Sprachdatei wurden Objekte eingefügt oder gelöscht)

Inconsistent alarm data in file <i>name</i>	die Daten der Alarmhistory sind nicht konsistent	Datei fehlerhaft, unvollständig oder nicht mehr kompatibel zur aktuellen EPAM-Version. Abhilfe: Datei löschen
Inconsistent sysvar.ini file	die Daten der SYSVAR.INI Datei sind nicht konsistent	Datei fehlerhaft, unvollständig oder nicht mehr kompatibel zur aktuellen EPAM-Version. Abhilfe: Datei löschen
Inconsistent recipe variable in file <i>name</i>	der Datentyp einer Rezeptvariable stimmt nicht mit dem Datentyp eines Objektes überein, das dieselbe Variable verwendet	Rezeptvariablen und Objektvariablen müssen denselben Datentyp haben
Invalid recipe file	Ungültige Rezeptdatei Name	Datei ist keine Rezepturdatei oder Syntaxfehler in Datei Abhilfe: Datei löschen; Rezept neu speichern
To many recipe variables defined	Zu viele Rezeptvariablen definiert	Es können max. 6553 Rezeptvariablen definiert werden
Type conflict variable	Typkonflikt Rezept-Variable Name	Typ in Rezeptvariable und im aktuellen Objekt stimmen nicht überein
Undefined variable in file <i>varname</i>	Fehler beim Laden von Rezeptdateien	Variable <i>varname</i> in aktueller Rezeptur ist nicht Bestandteil der Rezepturdefinition
Error writing recipe-variable: <i>varname</i>	Fehler beim Schreiben einer Rezeptvariable	Variable <i>varname</i> ist auf PLC nicht definiert
Error on DRV: <ARTi> Host: <PLC> There is no SDD assigned to the channel.	Keine Symboldatei vorhanden	Codesys Projektoptionen überprüfen
failed to read variable <i>Variablename</i>	Variable nicht in Symboldatei	Altes, falsches PLC-Projekt auf der Steuerung
Error on DRV: <ARTi> Host: <PLC> No project on target	Kein Projekt auf der Steuerung	PLC-Projekt auf Steuerung übertragen
Error on DRV: <ARTi> Host: <PLC> No file on the target or faild to open.	Keine Symbolfile auf dem Target	PLC-Projekt mit Symboldatei übertragen
Error on DRV: <ARTi> Host: <PLC> SDD has changed	Neues Projekt auf Target geladen	Mit OK quittieren.
Timeout writing variable	Zeitüberschreitung beim Schreiben einer Variable	Kommunikationsfehler
Error on DRV: <ARTi> Host: <xxx> Host not running	Kommunikation zum Host <xxx> konnte nicht aufgebaut werden	Keine Verbindung zur PLC xxx bzw. falsche IP-Adresse
Login was refused by the target	Login zur Steuerung konnte nicht möglich	Steuerung ist in einem Zustand, der keinen Zugriff auf Variablen erlaubt (z.B. beim Booten oder bei Online-Change) → evtl. Timeout in DRVParam zu klein (sollte >= Retry * DelayOnError in hosts sein!)
Language not found	Die aktuell eingestellte Sprache ist nicht vorhanden (sysvar.ini)	In der Datei SYSVAR.INI ist eine Sprache eingestellt, die im Projekt nicht existiert. Projekt mit Option „Delete INI-Files“ neu laden
Touch initialization failed	Touch konnte nicht initialisiert werden	Hardwareproblem mit der seriellen Schnittstelle zum Touch, falsche Konfiguration in AUTOEXEC.INI oder Touch defekt.
Invalid textfile	Ungültige Konfigurationsdatei des Objekts SYS2PLC	Datei ist keine Sys2PLC-Datei oder Syntaxfehler in Datei
Failed to register sysvar: <i>varname</i>	Fehler in SYS2PLC, Systemvariable konnte nicht angelegt werden	Zu viele Systemvariablen definiert (max. 1536), oder Typkonflikt
Incomplete trendparameters	Ungültige Konfigurationsdatei des Objekts TREND	Falsche oder ungültige Trenddatei, Syntaxfehler in Trenddatei
Invalid y-range on trend: value	Fehlerhafte Skalierung einer Y-Achse	Falsche oder ungültige Trenddatei,

		Syntaxfehler in Trenddatei
Failed to create semaphor	Interner Systemfehler	
Unknown error	unbekannter Fehler	

11 Versionsgeschichte

Index - **Erstausgabe vom 21.09.01**
39245-031-96-401

Autor G. Fischbacher

Revision	Date / Visum	Modification : Remarks, News, Attention
Version 1.30		Überarbeitung/Anpassung für EPAM-Version 1.30
Version 2.00		Überarbeitung/Anpassung für EPAM-Version 2.00 Neue Objekte: Alarmmail, DiagnoseSignal, Datalog, Meter, Trend Unicode Support Indizierte Variablenadressierung
Version 3.00	04.02 / DLen	Überarbeitung/Anpassung für EPAM-Version 3.0 Installation, Bilderstellung, Integration in IEC61131
Version 3.00.1	05.02 / Fis	Anpassung für XSoft
Version 3.10	05.03 / DLen	Überarbeitung/Anpassung für EPAM-Version 3.10
Version 3.20	06-04 / Fis	Überarbeitung/Anpassung für EPAM-Version 3.20
Version 3.30	03-06 / Fis	Überarbeitung/Anpassung für EPAM-Version 3.30
Version 3.40	05-09 / Fis	Überarbeitung/Anpassung für EPAM-Version 3.40
Version 3.50	08-11 / Fis	Überarbeitung/Anpassung für EPAM-Version 3.50

Freigabe

	Erstellt	Geprüft	Freigegeben
Name	G.Fischbacher
Datum
Visum

Grossenbacher Systeme AG
Spinnereistrasse 10
CH-9008 St.Gallen
Schweiz

Tel : +41 (0)71 243 29 29
Fax : +41 (0)71 243 29 28
Email : info gesys.ch
Homepage : www.gesys.ch

12 Stichwortverzeichnis

”

„Compare project“ 46

A

Aktionen mehrfach 38

Aktuelle Uhrzeit auf der PLC 88

Alarmanzeige 104

Alarmdiagnose 107

Alarmfilter 107

Alarmhandling 103

Alarmhistory 103

Alarmhistory speichern 104

Alarmquittierung 104

Alarmsortierung 107

Alarm-Systemvariablen 107

Alarmtyp 107

Alle Änderungen 7, 8

Änderungen in der EPAM V3.10 8

Änderungen in der EPAM V3.20 8

Änderungen in der EPAM V3.30 7

Änderungen in der EPAM V3.40 7

Änderungen in der EPAM V3.50 7

Anlegen der Rezeptverzeichnisse 113

Aufbau Alarmbuffer 103

Aufbau Alarmbuffer in Step7 103

B

Benutzerdefinierte Farbpalette 22, 42

Berechnung des Passwortes aus Tag und Monat
118

Bilderstellung 19

Bilder importieren 23

Bilder von Digitalkamera oder gescannt 24

PCX-Bilder erstellen 21

Bildschirmtastatur 77, 88

Bildseitenwechsel in der PLC auslösen 92, 128

Bildschirmschoner bei Alarmevent 117

Bildschirmschoner in der PLC deaktivieren 117

D

Definitionen der Excel-Tabelle grafisch darstellen 58

Dokumentation 20

E

Einstellungen in der CoDeSys -

Entwicklungsumgebung 11

Email 147

EPAM beenden 73

EPAM DB 62

EPAM Runtime-System 15

EPAM Wizard 59

EPAM.INI 18, 42

EPAM-Demos 58

EPAM-Homepage www.easypagemachine.com 10

EPAM-Konfiguration auf dem Zielsystem (EPAM.INI)

18

EPAM-Makros 44

Add Text 47

Add UserVar 47, 48

Build Language Texts 47

Build VarList 49

Build-Fonts 47

Define Language 47

Delete Language 47

Download Project 50

EPAM Version 46

Goto 44

Gruppierung 52

Gruppierung aufheben 52

NewAlarm 55

NewAlarmList 54

NewAlarmMail 56

NewBar 53

NewButton 53

NewDataLog 56

NewDiagnoseSignal 54

NewDropDownList 53

New-Font 46

NewHTMLBrowser 54

NewMessage 54

NewMeter 54

NewPage 53

NewPassword 56

NewRadioButton 53

NewRecipe 57

NewRecipeList 54

NewRemoteControl 54

NewScreenSaver 57

NewScrollList 54

NewSignal 53

NewSwitch 53

NewSys2PLC 57

NewTextList 54

NewTrend 55

NewVariable 53

Open Drawing Program 46

Open File 44

PCX-Colortranslation 46

PLC Variable Import 48

Project Settings 45

Rebuild Project 49

Save as Unicode Textfile 44

Save Worksheet as *.TXT 44

Search 44

Start EPAM 44

Start FontBuilder 47

Update Objects 48

Upload Project 52

EPAM-Symbolleiste

Easy PageMachine 44

EPAM-Symbolleiste

EPAM Objects 53

EPAM-Symbolleiste

- EPAM Wizard 59
- EPAM-Symbolleiste
 - EPAM DB 62
- EPAM-Wizard 38
- EPAM-Wizard-Makros
 - 100% 60
 - 150% 60
 - 200% 60
 - Align Bottom 59
 - Align Left 59
 - Align Right 59
 - Align Top 59
 - AutoSize 60
 - Format Heights 60
 - Format Widths 60
 - Full Screen 60
 - Half Screen 60
 - Properties 61
 - Refresh 59
 - Small Screen 60
 - Spacing Horizontal 59
 - Spacing Vertical 59
 - Undo 60
- Erstellung von eigenen Rezeptdateien 113
- Excel Tabelle
 - Spalte Action 38
 - Spalte Action Limit1, Action Limit2 39
 - Spalte Color, Backcolor 38
 - Spalte Font 37
 - Spalte Format 38
 - Spalte Limit1, Limit2 39
 - Spalte Object 37
 - Spalte Option 41
 - Spalte Text/File 37
 - Spalte VarState 41
 - Spalte VarType 40
 - Spalte VarValue 39
 - Spalte X,Y,DX,DY 38

F

- Fehlermeldungen 144
- Fenstertechnik 72
- Formatangaben 85, 105, 114
- Fotorealistische Bilder 24
- Frames 72
- Funktion der Passwortverwaltung 118
- Funktion der Scroll-Liste 120
- Funktion des Diagnose Signal 109
- Funktion von Alarmmail 108
- Funktion von DataLog 121
- Funktion von RemoteControl 129
- Funktion von Sys2Plc 127
- Funktion von Trend 125
- Funktionsprinzip 35
- Funktionsweise des RadioButtons 84

G

- Ganzzahlwerte mit Dezimalpunkt 88
- Globale Objekte 37, 72, 101, 108, 111, 119, 122, 127
- Grenzwerte 39
- Grossenbacher Systeme AG im Internet

- www.gesys.ch 10
- Grunddatentypen der CoDeSys 40

H

- Hinweise zu Geräten mit WindowsCE-Betriebssystem 7
- Homepage 147

I

- Import von S7 Variablen** 48
- Indizierte Variablenadressierung 40, 120
- Installation 10
 - Easy PageMachine EPAM 10
 - Einstellungen in Excel 10
 - EPAM-Makros installieren 10
 - Zeichnungsprogramm 10

K

- Kommandozeilen-Parameter WinEPAM 16
- Kommunikation 20
- Komplexe Datentypen
 - Strukturen, Array 40
- Konsistenz von Rezeptdateien 112
- Konsistenz von Rezeptwerten 112
- Kopieren und Löschen von Objekten im Wizard 61

M

- Mehrsprachige Applikationen 77
- Mehrstufige Rezepturen 112
- Meldungen von PLC aus löschen 95
- Meldungstexte mit Variablenwerten 95

N

- Neue Funktionen der EPAM V3.10 6
- Neue Funktionen der EPAM V3.20 5
- Neue Funktionen der EPAM V3.30 4
- Neue Funktionen der EPAM V3.40 4
- Neue Funktionen der EPAM V3.50 3
- New DBPasswd 62
- New DBTracer 62

O

- Objekt Alarm 101
- Objekt Alarmlist 105
- Objekt Alarmmail 108
- Objekt Bar 89
- Objekt Button 73
- Objekt DataLog 121
- Objekt DiagnoseSignal 109
- Objekt DropDownList 81
- Objekt HTMLBrowser 100
- Objekt Message 94
- Objekt Meter 96
- Objekt Page 71
- Objekt Password 118
- Objekt Radiobutton 83
- Objekt Recipe 110
- Objekt RecipeList 114
- Objekt RemoteControl 129
- Objekt ScreenSaver 116
- Objekt Scrolllist 120
- Objekt Signal 91

Objekt Switch 79
Objekt Sys2Plc 127
Objekt Textlist 98
Objekt Trend 125
Objekt Variable 85
Objektstatus 41
Objektstatus bei Bildwechsel 41
Option Bitwise=AND 119
Optionen mehrfach 41

P

Passwortschutz eines EPAM-Projekts 43
PCX-Bilder erstellen 21
Positionierung automatisch 38
Projektierung globaler Objekte 72, 101, 108, 111,
119, 122, 127
Projektierung mit Excel 35
Projektrealisierung 19

R

Release Mode 8
Remote-Control von gleich grossen Bildschirmen
129
Rezept in EXCEL erstellen 113
Rezept laden 113
Rezept speichern 113
Rezepte laden/speichern/löschen 115
Rezeptliste sortieren 115
Rezepturverwaltung 113
Runtime-System für Geräte mit VxWorks 15
Runtime-System für Geräte mit WindowsCE 15
Runtime-System für PocketPC (Target PocketPC-
240x320) 17

S

Schutz vor Fehlbedienung 117
Screenshot 58
Signalzustand 92
Speicherung von Systemvariablen 143
Sprachumschaltung 77
Standard Farbpalette 38
Struktur der Excel-Tabelle 36
Systemvariablen 88, 129, 141
Systemvariablen in Rezepturen 112
SYSVAR.INI 143

T

Tabellenblatt Alarm 102
Tabellenblatt Alarmmail 108
Tabellenblatt DataLog 123
Tabellenblatt Message 95

Tabellenblatt Recipe 112
Tabellenblatt Sys2Plc 128
Tabellenblatt Trend 126
Tabellenblätter in Excel 42
Tastaturtabelle 39
Tipps für TouchScreen Applikationen 20
Transparente PCX-Bilder 25

U

Unicode
Definition einer Unicode-Sprache 32
Eingabe der Unicode-Texte 32
Funktionsweise 31
Unicode-Support 31
Unsigned Datentypen 8
UserColor 42
USERCOLOR 22

V

Variable Aktion 78
Variablendefinition 20
Versionsgeschichte 147
Visual-Keyboard 31, 32
Voraussetzung HTML-Browser 100
Voraussetzungen
Entwicklungssystem 3
Zielsystem 3

W

Wechseln des Rezeptlisten-Verzeichnis 115
Wechseln zwischen den Rezepttypen 115
WindowsCE
Zeichensätze 7
WindowsCE
RAM-Drive 7
WindowsCE
Message-Objekt 7
WindowsCE
RemoteControl 8
WindowsCE
Message-Objekt 95
WinEPAM Runtime-System für PC/IPC 16

Z

Zeichensatzerstellung
Definieren von Unicode-Zeichensätzen 34
Definieren von Zeichensätzen 27
Einschränkungen 30
Kreieren von Zeichensätzen 30
Zeichensatzerstellung 27
Zeit/Datumsfunktionen 74, 85, 88, 114

