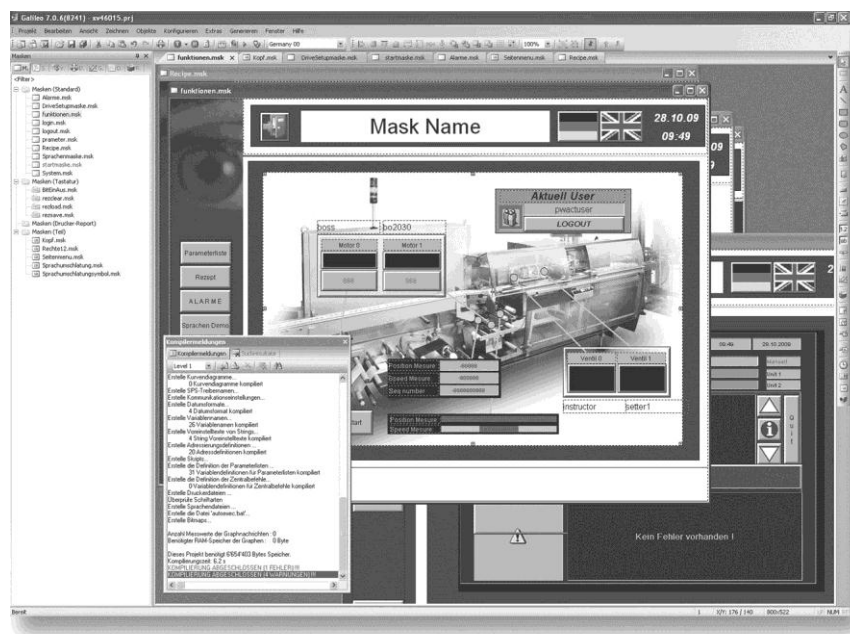


GALILEO Anwenderhilfe



Hersteller

Eaton Automation AG
Spinnereistrasse 8-14
CH-9008 St. Gallen
Schweiz
www.eaton-automation.com
www.eaton.com

Support**Region North America**

Eaton Corporation
Electrical Sector
1111 Superior Ave.
Cleveland, OH 44114
United States
877-ETN-CARE (877-386-2273)
www.eaton.com

Andere Regionen

Bitte kontaktieren Sie Ihren lokalen Lieferanten
oder senden Sie eine E-Mail an:
automation@eaton.com

Originalsprache

Deutsch

Redaktion

Jürgen Wagner

Marken- und Produktnamen

Alle in diesem Dokument erwähnten Marken- und Produktnamen sind Warenzeichen oder eingetragene Warenzeichen der jeweiligen Titelinhaber.

Copyright

© Eaton Automation AG, CH-9008 St. Gallen

Alle Rechte, auch die der Übersetzung, vorbehalten.

Kein Teil dieses Dokuments darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder einem anderen Verfahren) ohne schriftliche Genehmigung der Firma Eaton Automation AG, St. Gallen reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Änderungen vorbehalten

Inhaltsverzeichnis

1	Allgemein	6
2	Maskenaufbau	7
2.1	Einleitung	7
2.2	Grundsätzliche Performanceeinbussen	7
2.3	Zweck	7
2.4	Definitionen	7
2.4.1	Objekte:	7
2.4.2	Objektreihenfolge:	8
2.4.3	Überlappung:	9
2.4.4	Hintergrundbild:	9
2.5	Ressourcen optimierter Maskenaufbau	9
2.6	Beispiel	10
3	Projektübergreifendes Kopieren von Masken und deren Inhalt	12
3.1	Einleitung	12
3.2	Grundsätzliche Anforderung	12
3.3	Was kann kopiert werden	12
3.4	Vorbereitung und Grundeinstellungen	13
3.4.1	Empfehlung	13
3.4.2	Vorbereitung	13
3.4.3	Sprachen	14
3.5	Kopieren	15
4	Farben in GALILEO	17
4.1	Einleitung	17
4.2	Zweck	17
4.3	Farben in Objekten und Bitmaps	17
4.3.1	Farbeigenschaften der Objekte mit Farbauswahl	17
4.3.2	Farbeigenschaften der Bitmaps	17
4.3.3	Beispiel:	18
4.4	Optimierte Bitmap Farbpalette	20
4.5	Limiten	20
4.6	Zusammenfassung	20
5	Benutzerlogger	21
5.1	Einleitung	21
5.2	Einsatzgebiete	21
5.3	Funktionsweise	21
6	Grundsätzlicher Projektierungs-Ablauf - Logger	22
6.1	Benutzerlogger Projekt Einstellungen	23
6.2	Ignore-Liste	25
6.3	Log-File Einträge	27
6.4	Spezialfunktionen	28
6.4.1	Add String Tag Log (Stringvariablen schreiben)	28
6.4.2	Add Text Log (definierte Zeichenfolge schreiben)	29

Inhaltsverzeichnis

6.4.3	Add Value Tag Log (Wertvariablen schreiben)	29
6.4.4	Flush Logger (Datenbuffer speichern)	29
6.4.5	Logger ON-OFF (Logger EIN / AUS)	29
6.4.6	View Current Log File (Protokolldatei anzeigen)	29

Allgemein

Dieses Dokument versteht sich als Anwendungshilfe für das Erstellen von Projekten mit der Visualisierungssoftware GALILEO.

Das Erstellen von speziellen Objekten wie «Hilfe-Informationen», «Erweiterte Teilmasken mit Benutzerdefinierten Datentypen», «Benutzerlogger», «Graph», «Parameterlisten» und weiteren, ist nicht in der GALILEO Online-Hilfe beschrieben.

Dieses Dokument wird laufend erweitert und bezieht sich grundsätzlich nicht auf eine spezielle GALILEO Version. Darum können sich Beispiele, Screenshots und beschriebene Funktionen von der von ihnen benutzten GALILEO Version unterscheiden.

Dieses Dokument soll helfen, anhand von Beispielen spezielle Objekte in GALILEO mit weniger Aufwand so zu erstellen, dass sie sofort benutzt werden können. Es werden nicht die vollumfänglichen Funktionen, sondern nur eine sinnvolle Reihenfolge für das Erstellen der Objekte beschrieben.

Dieses Dokument erhebt keinen Anspruch auf Vollständigkeit.

2 Maskenaufbau

2.1 Einleitung

Dieses Kapitel ist eine Hilfe für den Aufbau von Projektmasken. Werden ein paar einfache Regeln beim Projektieren von Masken beachtet, können Probleme beim Maskenaufbau und beim Aktualisieren der Daten in der Runtime vermieden werden. Im Besonderen betrifft dies Graphisch anspruchsvolle Projekte.

2.2 Grundsätzliche Performanceeinbussen

Folgende Punkte können die Performance der Runtime besonders beeinträchtigen:

- Objekte, bei denen «Transparent» aktiviert ist.
- Viele sich überlappende Objekte die dynamisiert und transparent sind.
- Keine optimale Objektreihenfolge.
- Überlappende, bedienbare Objekte die gleichzeitig sichtbar sind.

2.3 Zweck

Dieses Dokument soll dem Anwender von GALILEO ermöglichen:

- schon während der Planung und Projekterstellung auf den optimalen Aufbau von Masken zu achten,
- die Probleme im Zusammenhang von nicht optimal aufgebauten Projekt-Masken, die plötzlich einen sehr langsamen Bildaufbau oder eine hohe Aktualisierungszeit aufweisen, zu erkennen und zu eliminieren.

2.4 Definitionen

2.4.1 Objekte:

Grundsätzlich unterscheidet man zwischen zwei Arten von Objekten wenn es um Performance Beeinflussung geht: dynamische und nichtdynamische Objekte. Ein Objekt ist dann dynamisch, wenn eine oder mehrere der folgenden Bedingungen erfüllt sind:

- die Grösse ist dynamisch über Variablen einstellbar,
- die Position ist dynamisch über Variablen einstellbar,
- es ist eine Zugänglichkeit oder Sichtbarkeit definiert,
- der Objektinhalt ist veränderbar (z.B. Werteingabe /-anzeige, Wahlschalter, Zustandsanzeige mit mehreren Anzeigemöglichkeiten von Texten oder Bitmaps, ...)

Alle anderen Objekte gelten als nichtdynamisch.

2.4.2

Objektreihenfolge:

Beim Erzeugen von Objekten erhalten diese eine Reihenfolge in aufzählender Ordnung. Diese Reihenfolge kann im Menü «Ansicht» – «Tooltip» – «Objektreihenfolge» oder über die Symbolleiste «Projektierungshilfe» zur Ansicht ausgewählt werden.

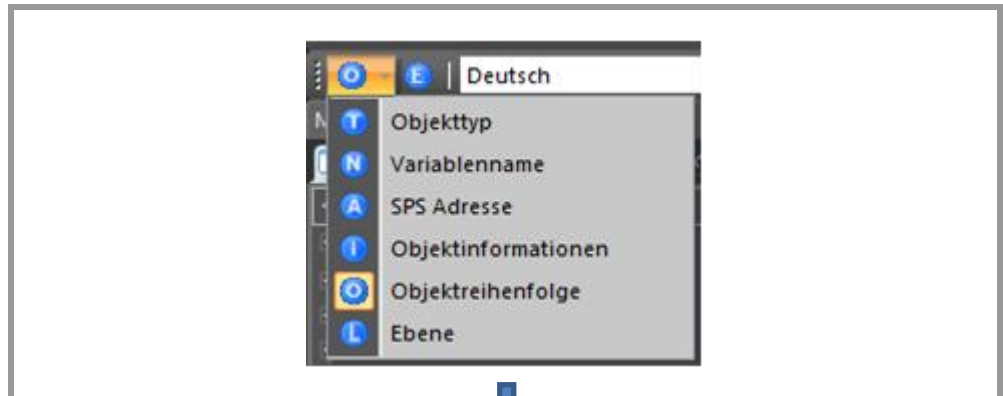


Abb. 1 Projektierungshilfe

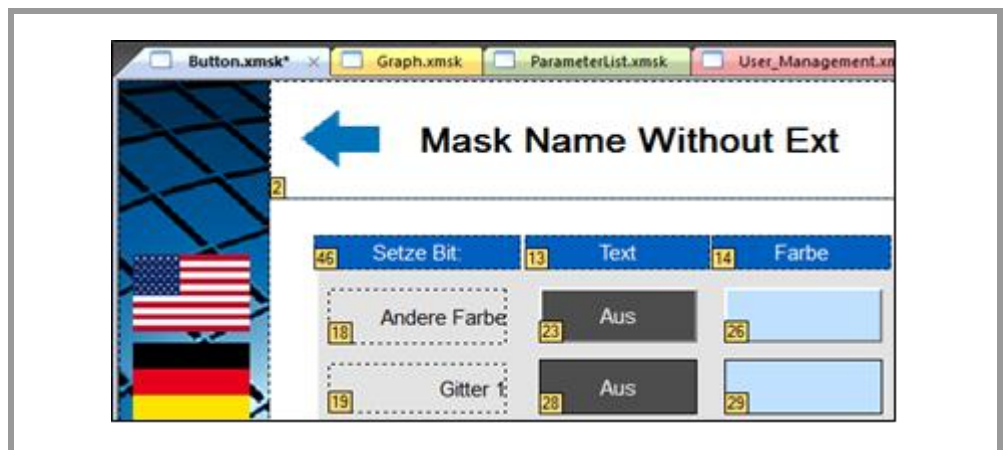


Abb. 2 Ansicht Objektreihenfolge

Soll ein Objekt in der Objektreihenfolge verschoben werden, gehen Sie wie folgt vor:

Markieren Sie das zu verschiebende Objekt. Betätigen Sie die Tasten + (Plus) oder – (Minus). Das markierte Objekt lässt sich so bei jedem Tastendruck jeweils um eine Position vor oder zurück verschieben (auch über das rechte Maustaste-Menu einstellbar – «Nach Vorne» oder «Nach Hinten»).

2.4.3

Überlappung:

Mit Überlappung von Objekten sind Objekte gemeint, die ganz oder teilweise übereinander liegen. Ändert sich ein dynamisches Objekt (d.h. ändert sich der Zustand, Variablenwert, Position, ...) so werden alle Objekte die eine höhere Objektreihenfolge haben und das dynamische Objekt überlappen, aktualisiert.

Ist in diesem dynamischen Objekt auch noch Transparent aktiviert, so werden zusätzlich auch die darunterliegenden, also in der Objektreihenfolge tiefer gelegenen Objekte, sofern sie vom dynamischen Objekt überlappt sind, aktualisiert.



Schräge Linien werden als transparentes Objekt betrachtet, also eine Linie innerhalb eines transparenten Rechtecks um die Endpunkte der Linie.

2.4.4

Hintergrundbild:

Das GALILEO-Runtime scannt die Objekte auf einer Maske der Objektreihenfolge nach durch (von 0 aufwärts) und merkt sich das erste dynamische Objekt. Bis zu diesem dynamischen Objekt werden alle nichtdynamischen Objekte als Hintergrundbild dargestellt. Ab dem ersten dynamischen Objekt werden nun die Objekte wie unter «Überlappung» beschrieben aktualisiert.

Speziell am Hintergrundbild ist, dass dieses nicht aktualisiert wird von darüber liegenden, transparenten Objekten.

2.5

Ressourcen optimierter Maskenaufbau

Gemäss den oben genannten Definitionen lassen sich folgende Grundregeln für einen optimalen Maskenaufbau bestimmen:

- Auf die Objektreihenfolge achten. Zeichnen Objekte wie Rechtecke, Linien, Polygone usw. aber auch Bilder, die statisch als Hintergrund benutzt werden, müssen die tiefste Objektreihenfolge besitzen (Objektreihenfolge siehe Kapitel 2.4.2).
- Transparente, dynamische Objekte so wenig wie möglich mit anderen Objekten überlappen, da alle darüber- und darunterliegenden Objekte bei einer Veränderung des Objektes aktualisiert werden.
- Bei transparenten Objekten muss speziell darauf geachtet werden, dass sie nicht grosse Objekte überlappen, auf denen viele andere transparente Objekte liegen welche dann alle aktualisiert werden, wenn sich ein Objekt aktualisiert. Dies kann zu einer Kettenreaktion führen welche die Performance erheblich verschlechtert (siehe Kapitel 2.6 «Beispiel»).
- Bedienbare Objekte, die übereinander liegen, sollen über die Zugänglichkeit so geregelt werden, dass immer nur ein Objekt sichtbar ist.

2.6

Beispiel

Ein Projekt mit 5 Produktionslinien.

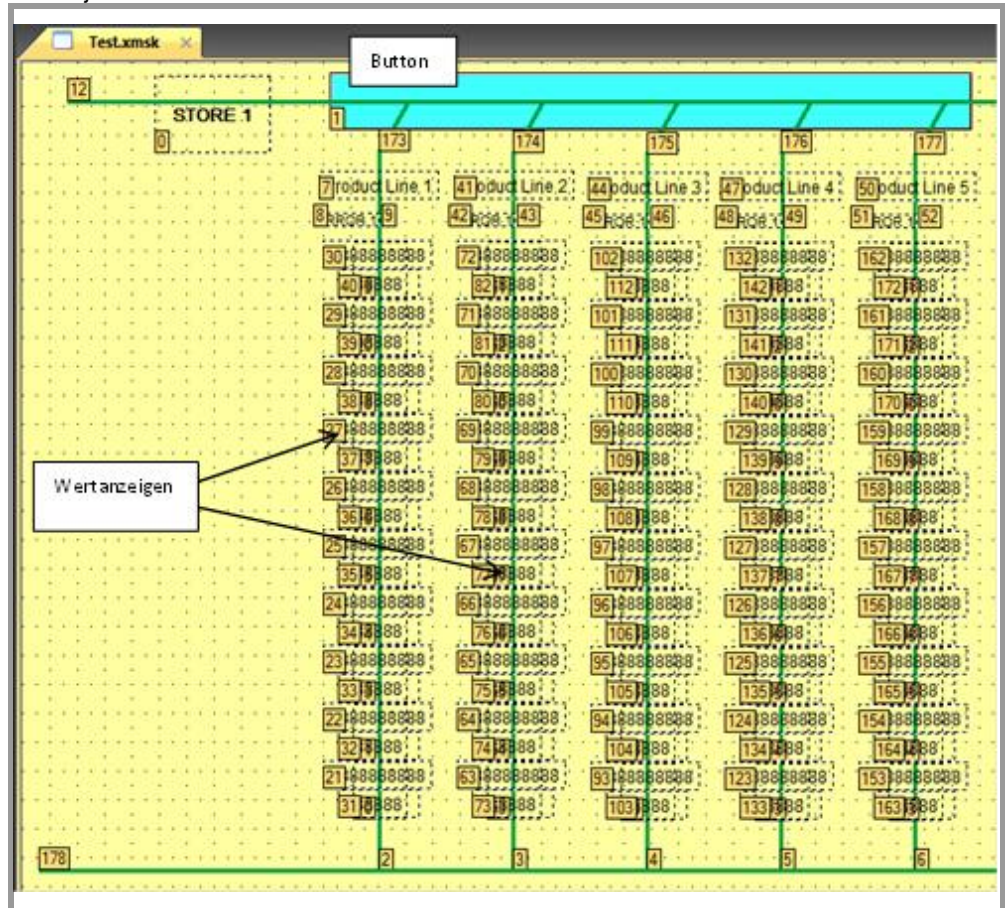


Abb. 3 Beispiel mit erheblichen Performance Einbußen

In jeder Produktionslinie sind über 20 Wertanzeigen definiert. Alle Wertanzeigen sind transparent, damit die vertikalen grünen Linien darunter sichtbar sind. Die Wertanzeigen sind aneinander anliegend, nicht überlappend angelegt.

Die vertikalen grünen Linien führen alle in den blauen Button (dieser ist ein dynamisches Objekt) oben im Bild. Da dieser Button in der Objektreihenfolge tiefer ist als die vertikalen grünen Linien, werden diese nicht als Hintergrundbild betrachtet, obwohl sie als solche gedacht sind. Sobald einer der 100 Werte ändert, wird die darunterliegende grüne Linie ebenfalls aktualisiert, weil der Wert transparent ist. Die grüne Linie wiederum ist überlappend mit der schrägen grünen Linie und wird als transparent betrachtet (siehe Kapitel 2.4.3 - Überlappung *Speziell*). Die schräge grüne Linie ist überlappend mit der horizontalen grünen Linie. Über die horizontale grüne Linie sind alle anderen Produktionslinien ebenfalls miteinander verbunden (auch überlappend). Ändert also eine Wertanzeige, dann werden alle 100 Anzeigewerte aktualisiert. Dies führt in diesem Beispiel zu langen Aktualisierungszeiten.

3 Projektübergreifendes Kopieren von Masken und deren Inhalt

3.1 Einleitung

3 Projektübergreifendes Kopieren von Masken und deren Inhalt

3.1 Einleitung

Dieses Kapitel beschreibt das optimale Vorgehen, wie Masken, Objekte und deren Parameter von Projekt A nach Projekt B kopiert werden.

3.2 Grundsätzliche Anforderung

- Ab GALILEO Version 6.01.
- Es kann nur zwischen Projekten mit der gleichen GALILEO Version kopiert werden.

3.3 Was kann kopiert werden

Es werden beim Kopieren von Objekten auf Masken die Bitmap-Dateien, Texte und Variablenzuordnungen mitkopiert.

Bitmaps

Existiert die Zielformat bereits und die Quell- und Zielformat sind unterschiedlich, kann der Bediener einen neuen Dateinamen angeben. Auch beim Importieren von Bitmaps wird der Bediener nur noch aufgefordert einen neuen Dateinamen anzugeben, wenn die Zielformat bereits existiert und die Quell- und Zielformat unterschiedlich sind.

Texte

Als Referenz für die Sprache gilt der Sprachename.

- Alle übereinstimmenden Sprachen werden ermittelt.
- Bei der ersten übereinstimmenden Sprache wird der Text gesucht.
- Wird der Text gefunden, wird die entsprechende Text-ID übernommen.
- Wird der Text nicht gefunden, wird der Text in alle übereinstimmenden Sprachen eingefügt. Wenn möglich wird dieselbe Text-ID wie im Quellprojekt verwendet. Ansonsten wird der Text zuhinterst in der Liste angefügt.
- Werden keine übereinstimmenden Sprachen gefunden, werden keine Texte eingefügt.

Maskenwechsel

Beim Einfügen von Maskenwechsel- und Teilmasken- Objekten wird der entsprechende Maskenname im Zielprojekt gesucht und zugeordnet. Existiert der Maskenname nicht, wird die Maskenzuordnung gelöscht.

Rezepte

Die Variablenzuordnungen bleiben erhalten, sofern die Variablen im Zielprojekt bereits existieren. Dies gilt für alle Variablen innerhalb des Rezeptes inklusive der Schreibschutzeinstellungen. Zudem werden die Sicherheitsabfragemasken ebenfalls übernommen, sofern die Maskendateien bereits existieren.

3 Projektübergreifendes Kopieren von Masken und deren Inhalt

3.4 Vorbereitung und Grundeinstellungen

Variablen

Beim Kopieren von Variablen werden alle Parameter mitkopiert. Auch die dynamischen Grenzwerte-Variablen, sofern welche definiert sind, werden korrekt übernommen. Dazu müssen aber die Grenzwertvariablen zuerst kopiert werden. Ebenfalls muss die Kommunikation parametrisiert sein, damit die Adressierung mitkopiert wird. Dabei wird auf die Kommunikationsnummer und die Reihenfolge in der Kommunikationstabelle (nicht auf die Kommunikationsart) geachtet.

Parameterlisten Einträge

Die Parameterlisten Einträge (im Parameterlistenmanager) wie Gruppe, Zugänglichkeit (siehe hierzu auch Anmerkung zu den «Variablen»), Nummer, etc. werden korrekt übernommen. Bei Bedarf wird die Gruppe neu erstellt.

Als Referenz gilt der Gruppenname.

Graphen

Die Variablenzuordnungen bleiben erhalten, sofern die Variablen im Zielprojekt bereits existieren. Dies gilt für alle Variablen innerhalb des Graphen, inklusive der Trigger-Variablen. Zudem werden die Sicherheitsabfragemasken ebenfalls übernommen, sofern die Maskendateien bereits existieren.

Druckerformulare

Die Variablenzuordnungen bleiben erhalten, sofern die Variablen im Zielprojekt bereits existieren. Zudem werden die Texte mitkopiert.

3.4

Vorbereitung und Grundeinstellungen

In GALILEO braucht es diverse Vorbereitungen und Einstellungen, die vor dem Kopieren gemacht werden müssen, um verschiedene Einstellungen nicht zu verlieren. Folgen Sie Schrittweise dieser Anleitung.

3.4.1

Empfehlung

Sichern Sie das Zielprojekt (unter anderem Namen oder als Backup), um bei unerwünschtem Ergebnis auf den Ursprung zurückgreifen zu können.

3.4.2

Vorbereitung

Die Kommunikation des Quell- und Zielprojekts müssen identisch sein. Ansonsten können die Adressen nicht richtig übernommen werden (bei mehreren Kommunikationen müssen die Kommunikationstypen mit der Reihenfolge in der Kommunikationstabelle korrespondieren). Überprüfen Sie beim Quellprojekt die allgemeinen Projekteinstellungen. Menü Konfigurieren → Einstellungen. Übertragen sie diese in das Zielprojekt. Z.B. in Tab «Weitere...».

Hinweis: Sind die eingesetzten Panels unterschiedlich in der Grösse, so können Objekte beim Einfügen ausserhalb der Maske angeordnet werden (z.B. wenn von einem grösseren in ein kleineres Panel kopiert wird). In diesem Fall müssen die Objekte vor dem Kopieren richtig positioniert werden.

3 Projektübergreifendes Kopieren von Masken und deren Inhalt

3.4 Vorbereitung und Grundeinstellungen

3.4.3

Sprachen

Sprachen müssen mit dem gleichen Namen angelegt werden. Die kopierten Texte werden nur in die Sprachen übernommen die auch den gleichen Namen tragen. Wir gehen in den Beispielen immer von einem Zielprojekt aus, welches noch keine Texte beinhaltet. Es ist also ein neues, leeres Projekt erstellt worden.

Es gibt nun verschiedene Möglichkeiten:

- In GALILEO Sprachen erzeugen
- Importieren
- Leeres XML Sprachfile erzeugen
- Kopieren

In GALILEO Sprachen erzeugen

Menu «Konfigurieren» → «Sprachen» auswählen. Bei einem neueröffneten Projekt wird automatisch eine Sprache 00 erstellt. Dieser Sprachename muss dem Sprachename im Quellprojekt entsprechen. Nennen wir sie Germany. Um eine neue Sprache zu erstellen klicken Sie die «rechte Maustaste» → «Sprache hinzufügen» oder «Funktionstaste F6», Namen eingeben und mit Enter-Taste bestätigen.

Importieren

Dies macht Sinn, wenn zu einem grossen Teil oder alle Texte vom Quell- ins Ziel-Projekt übernommen werden sollen.

Menu «Konfigurieren» → «Texte Exportieren und Importieren».

Damit können alle Texte die man benötigt über eine Excel Liste exportiert und ins Ziel-Projekt importiert werden.

Leeres XML Sprachfile erzeugen

Erstellen eines leeren XML Files, welches den Namen der Sprache trägt die wir später einfügen wollen. Wie schon erwähnt wird eine «Sprache00» automatisch erstellt. Darum befindet sich im Zielprojektordner ein Ordner «lang». Hier werden alle Sprachen eines Projektes in einem XML Format abgelegt. Beispiel: Sprache00_00.xml.

Dieses File kann nun beliebig oft kopiert und im Ordner «lang» eingefügt werden. Das File nach dem kopieren jeweils umbenennen und die Namen der im Quellprojekt vorhandenen Sprachen angeben.

Kopieren von XML

Beispiel: Im Quellprojektordner im Ordner «lang» befinden sich alle Sprachen des Projektes wie z.B. Germany_00.xml.

Dieses File kann direkt in das Zielprojekt in den Ordner «lang» kopiert werden. Dadurch sind sämtliche Texte enthalten, was nicht immer erwünscht ist.

3.5

Kopieren

Beim Kopieren bitte die folgende Reihenfolge einhalten:

1. Benutzerdefinierte Datentypen

Die Benutzerdefinierten Datentypen werden automatisch erzeugt, wenn die dazugehörigen instanziierten Variablen kopiert werden.

2. Variablen

Alle benötigten Variablen vom Quellprojekt in das Zielprojekt kopieren.

Es kann immer nur ein Datentyp kopiert werden. Nur Strukturen können gleichzeitig kopiert werden. Zu beachten ist, dass bei voneinander abhängigen Variablen (z.B. bei dynamischen Grenzwerten), erst die Variablen der Grenzwerte kopiert werden.

Ist die Struktur eine Instanz eines Benutzerdefinierten Datentyps, so wird dieser automatisch erstellt.

3. Datenabhängige Objekte

Es gibt viele Objekte die eine Datenbasis haben wie beispielsweise Rezepte, Parameterlisten, Graphen, Skripte.

Diese Datenbasis muss vor den Objekten kopiert werden! Rezepte und Graphnachrichten können direkt kopiert werden. Bei den Skripten geht dies auch, wobei bei vielen Skripten das Exportieren und Importieren schneller ist.

Für das Kopieren von Parameterlisten müssen zuerst alle Einträge die sich im Parameterlistenmanager befinden, kopiert werden.

4. Masken anlegen

Kopieren von einer oder wenigen Masken:

Als erstes die Teil- und Tastatur- Masken neu anlegen, danach die Standard- Masken. Beachten sie hierbei dass alle Attribute gleich sind wie im Quellprojekt.

Kopieren von vielen Masken:

Bei diesem Vorgehen können alle Maskenattribute (Grösse, Farben, usw.) und Namen übernommen werden!

Schliessen Sie das Quell- und Zielprojekt. Kopieren sie im Windows-Explorer alle *.msk - oder *.xmsk - Files (ab Version 8.0.0 sind Masken im *.xmsk Format) die benötigt werden aus dem Quellprojekt in das Zielprojekt.

Hinweis: Alle Zuweisungen von Objekten gehen dabei verloren.

Zuerst alle Teilmasken und Tastaturmasken übernehmen. Öffnen Sie das Zielprojekt in GALILEO und wechseln Sie in den Übersichtsordner Masken. Mit der rechten Maustaste auf den Masken (Standard) in das Menu «Projektmasken» wechseln. Übernehmen sie die Masken die sie brauchen in das rechte Fenster (Projektmasken). Speichern sie nun das Zielprojekt und öffnen sie alle Teil- und Tastaturmasken im Projekt. Markieren sie alle Objekte auf der Maske mit Ctrl+A und löschen sie dann alle Objekte.

Status → Es sind alle Masken mit Namen vorhanden, aber leer.

3 Projektübergreifendes Kopieren von Masken und deren Inhalt

3.5 Kopieren

5. Objekte in Masken kopieren

Nun kopieren sie die Objekte auf den Teil- und Tastatur-Masken aus dem Quellprojekt in das Zielprojekt. Immer die gleichen Masken geöffnet halten. Es wird von Maske zu Maske kopiert. Speichern Sie das Zielprojekt. Schliessen Sie das Zielprojekt und öffnen Sie es wieder. Wiederholen Sie die Schritte auch mit den Standard Projektmasken. Bei allen Objekten werden Bilder, Texte, Variablenzuweisungen usw. mitkopiert, wenn diese im Projekt vorhanden sind.

4 Farben in GALILEO

4.1 Einleitung

In GALILEO kann zu jedem Panel-Typ die Farbtiefe ausgewählt werden (typisch 256 oder 65536). Bitmaps (Bilder) und Farben in Objekten mit Farbauswahl werden immer mit einer Farbtiefe von maximal 256 Farben gespeichert. Beim Anzeigen von Bitmaps und Objekten mit Farben stellt sich nun die Frage, mit welchen Farben diese Bilder auf den Panels dargestellt werden.

4.2 Zweck

Dieses Kapitel erklärt, wie die GALILEO Software Objekte und Bitmaps in Bezug auf die Farben behandelt.

4.3 Farben in Objekten und Bitmaps

4.3.1 Farbeigenschaften der Objekte mit Farbauswahl

Bei gezeichneten Objekten und bei der Farbauswahl in Objekten kann aus der Farbpalette, welche im Menu «Konfigurieren» - «Farbpalette» definiert ist, die Farbe ausgewählt werden. Diese Farbpalette kann individuell abgeändert werden indem man nicht benutzte durch neue Farben ersetzt. Oder man erstellt eine komplett neue Farbpalette. Dies kann sinnvoll sein, wenn Farben gebraucht werden, die nicht in der bestehenden Palette vorhanden sind. Es kann aber nur eine Farbpalette pro Projekt verwendet werden. Dabei ist man auf 256 Farben beschränkt. Die Objekte werden auf dem Panel genau so dargestellt, wie im GALILEO definiert, da die Farbpalette auf das Panel übertragen wird.

4.3.2 Farbeigenschaften der Bitmaps

Bitmaps werden beim kompilieren in das PCX – Format gewandelt. Dabei wird für jedes einzelne Bitmap eine Farbpalette erstellt mit maximal 256 Farben. Ist die Farbtiefe des Bitmaps grösser als 256 Farben, wird beim kompilieren ins PCX – Format eine auf das Bitmap optimierte Farbpalette mit 256 Farben anhand des Farbraumes des Bitmaps erstellt. Mit dem Erstellen einer individuellen Farbpalette pro Bitmap erreicht man, dass die farbliche Charakteristik des Bildes erhalten bleibt. Weitere Informationen zum PCX-Format siehe: http://de.wikipedia.org/wiki/Picture_exchange

4 Farben in GALILEO

4.3 Farben in Objekten und Bitmaps

4.3.3

Beispiel:

Bildeigenschaften Originalbild mit einer Farbtiefe von 60641 Farben

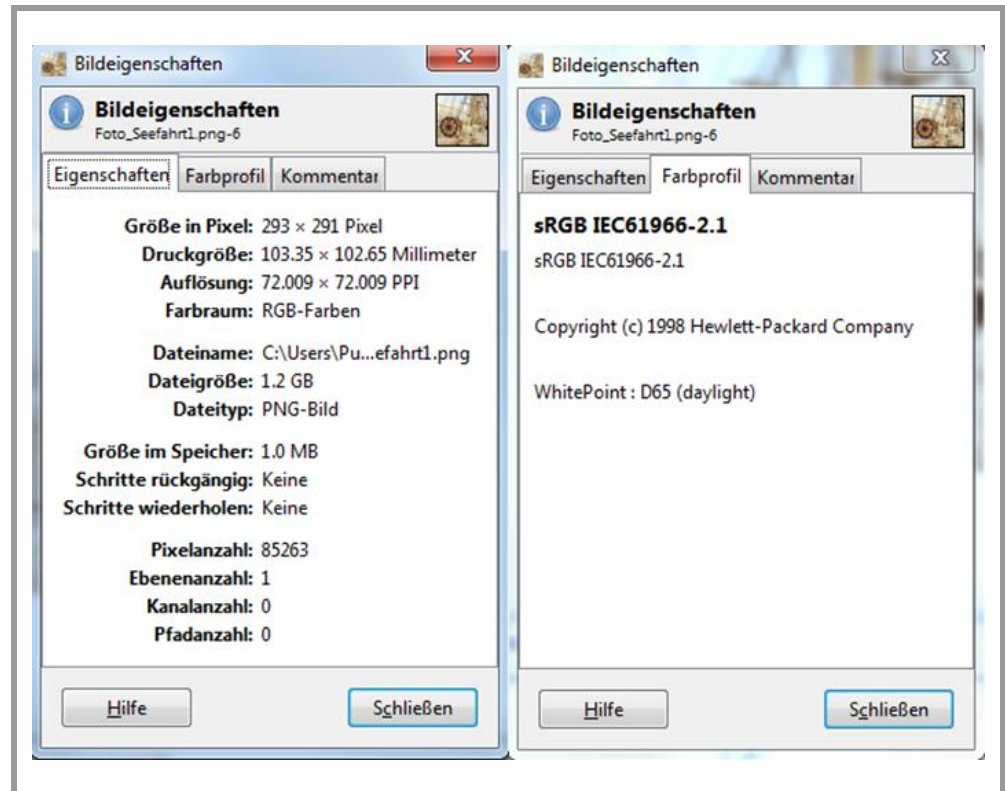


Abb. 5 Bildeigenschaften Originalbild



Abb. 6 Farbraumanalyse Originalbild

4 Farben in GALILEO 4.3 Farben in Objekten und Bitmaps

Bildeigenschaften nach dem Kompilieren – optimierte Farbpalette mit 256 Farben im PCX – Format

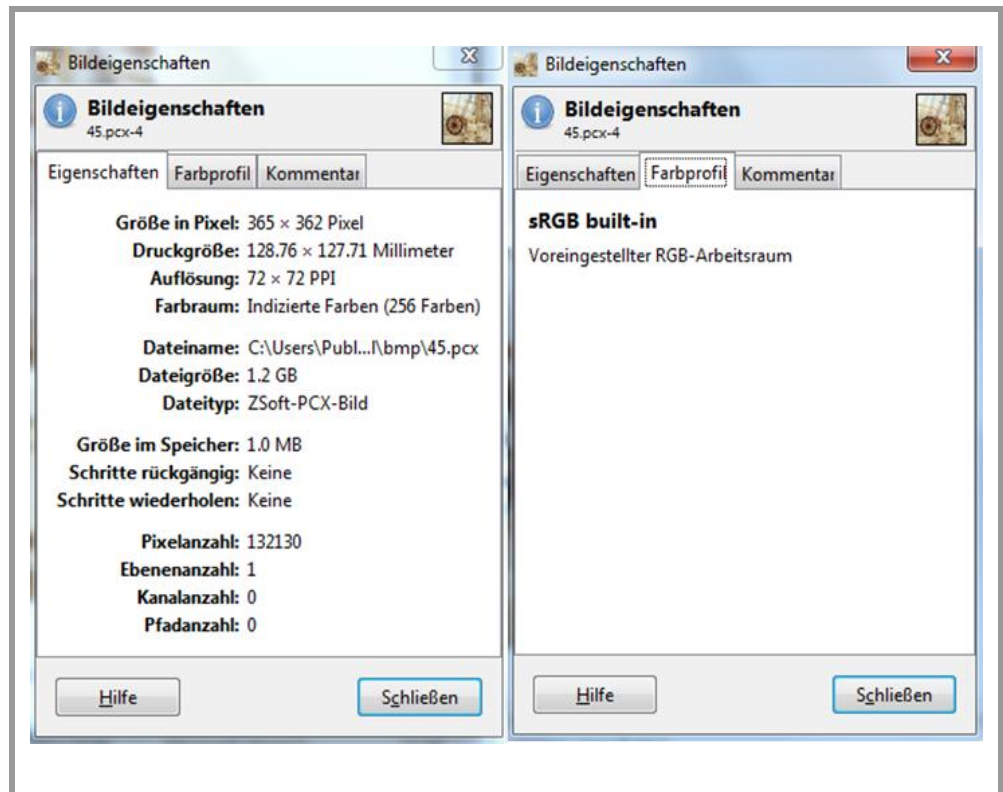


Abb. 7 Bildeigenschaften kompiliertes Bild

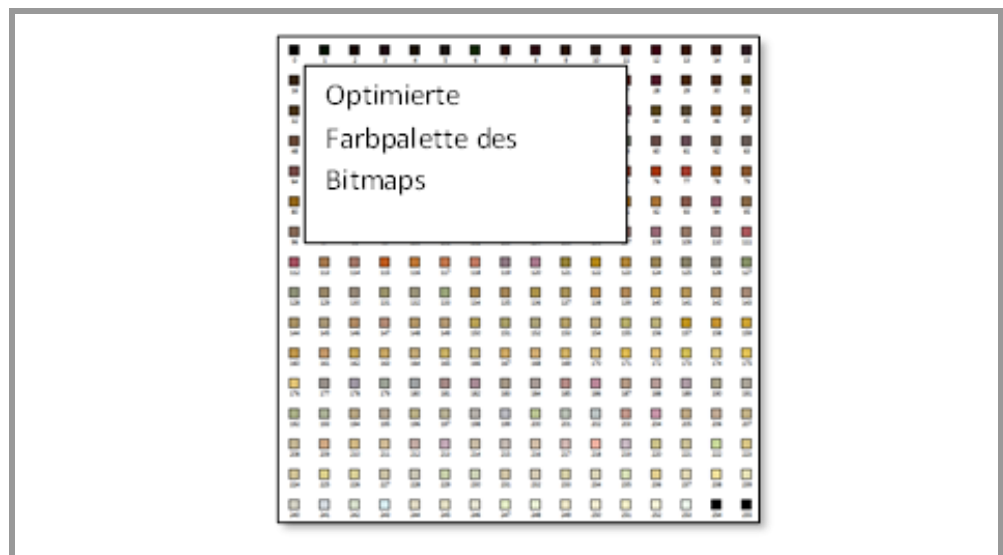


Abb. 8 Farbraumanalyse Originalbild

Diese erstellte Farbpalette wird nun mit dem Bitmap auf das Panel geladen.

4 Farben in GALILEO

4.4 Optimierte Bitmap Farbpalette

4.4

Optimierte Bitmap Farbpalette

Anhand der folgenden 2 Bilder mit ihren Farbpaletten erkennt man, dass jede Farbpalette individuell erstellt wird mit den für den Farbraum des Bildes charakteristischen Farben.

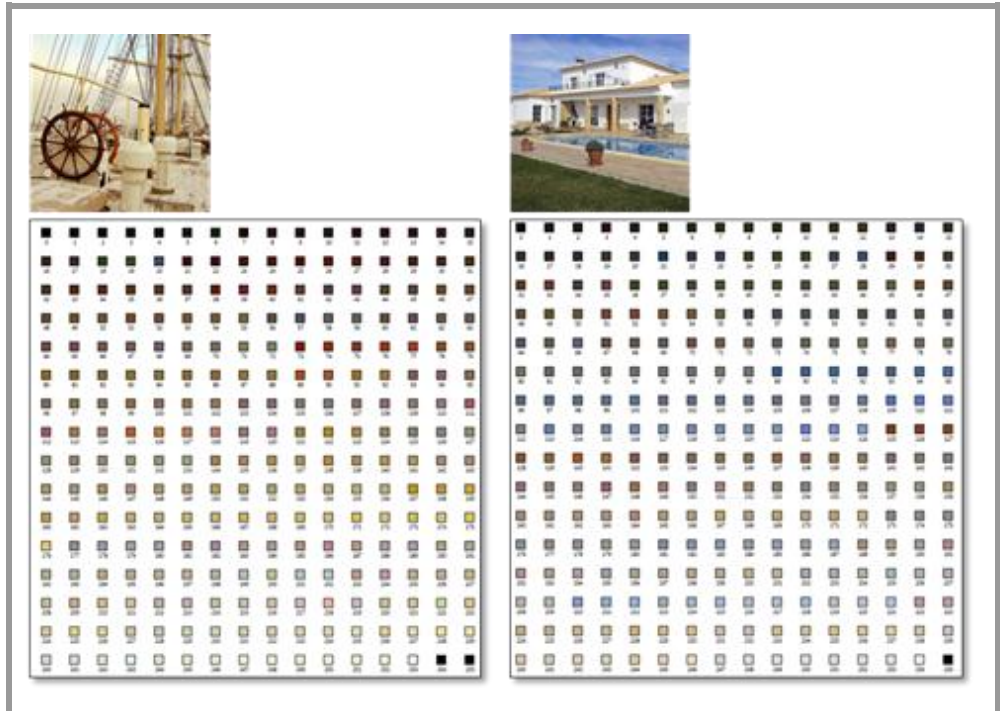


Abb. 9 Optimierte Farbpaletten

4.5

Limiten

Das Panel kann maximal 65536 Farben anzeigen. Dies bedeutet, dass die Farbpalette insgesamt für die Objekte mit Farbauswahl und alle optimierten Bitmap-Farbpaletten zusammen nicht mehr als 65536 unterschiedliche Farben aufweisen dürfen. Wenn alle Bilder und Bitmaps je 256 unterschiedliche Farben in ihrer Farbpalette hätten, könnten maximal 255 Bilder in ihren optimierten Farben dargestellt werden ($65536 / 256 - 1 \times$ Farbpalette für Objekte mit Farbauswahl).

4.6

Zusammenfassung

- Zum Zeichnen von Objekten und für die Farbauswahl der Objekte stehen 256 Farben zur Verfügung. Diese Farbpalette kann bei Bedarf selber angepasst oder mit neuen Farben ergänzt werden.
- Für jedes Bitmap wird eine eigene, auf das Bild optimierte Farbpalette mit 256 Farben angelegt und mit dem Bitmap auf das Panel übertragen. Somit bleibt für jedes Bitmap die Farbcharakteristik erhalten (im Header des *.pcx Format enthalten).
- Auch bei mehreren Bildern auf derselben Maske hat jedes Bild eine eigene optimierte Farbpalette.

5 Benutzerlogger

5.1 Einleitung

Alle Benutzereingaben wie Daten verändern und Funktionsaufrufe werden automatisch vom Benutzerlogger protokolliert wenn er aktiviert ist. Sollen einzelne Werte oder Funktionen nicht protokolliert werden, so können Listen von Variablen und Funktionen erstellt werden, welche bei der Protokollierung ignoriert werden.

Hinweis:
Werte, die durch die Steuerung verändert werden, werden nicht protokolliert.

5.2 Einsatzgebiete

Einsatzgebiete für den Benutzerlogger sind:

- Qualitätssicherung
- Statistische Auswertung von Daten
- Fehlersuche bei Problemen
- Protokollierte Produktion

5.3 Funktionsweise

In einem Datenbuffer werden alle Benutzereingaben aufgezeichnet. Sobald im Datenbuffer 50 Einträge sind, oder nach maximal 1 Minute, werden die Log-Daten vom Datenbuffer in ein Log-File geschrieben.

Wird das Panel mit aktiviertem Benutzerlogger von der Versorgungsspannung getrennt, gehen alle Log-Daten die noch im Datenbuffer sind, verloren. Beim Neustart wird der Benutzerlogger automatisch wieder gestartet.

6 Grundsätzlicher Projektierungs-Ablauf - Logger

5.3 Funktionsweise

6

Grundsätzlicher Projektierungs-Ablauf - Logger

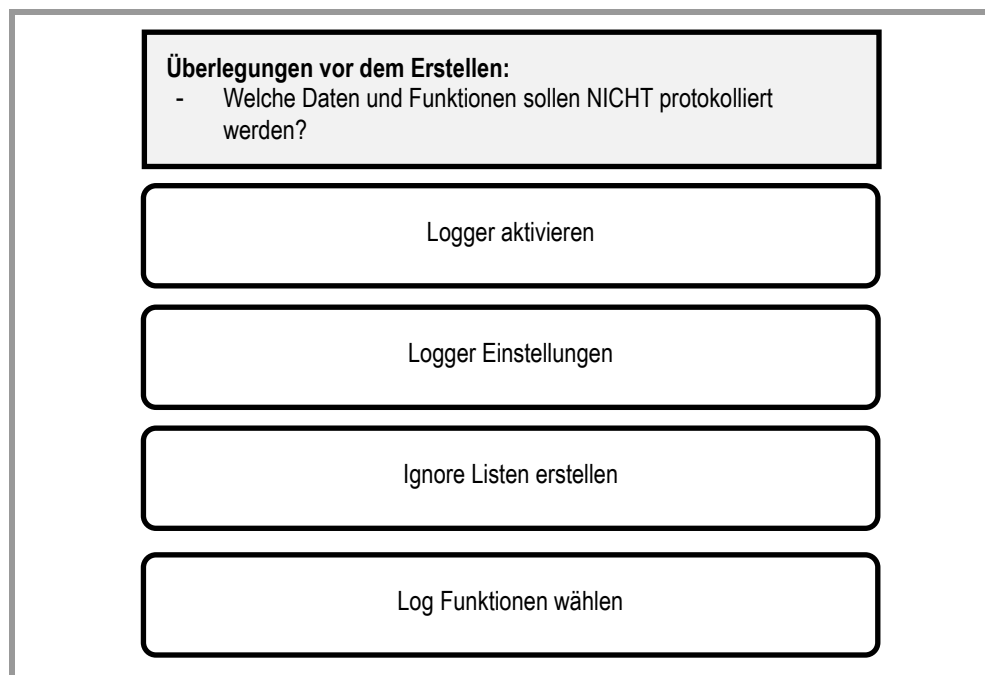


Abb. 10 Projektierungs-Ablauf

6.1

Benutzerlogger Projekt Einstellungen

Im Menü «Konfigurieren» - «Einstellungen...» öffnet sich das Fenster «Projekt Einstellungen».

Über den Reiter «Benutzerlogger» können die nachfolgenden Einstellungen vorgenommen werden:

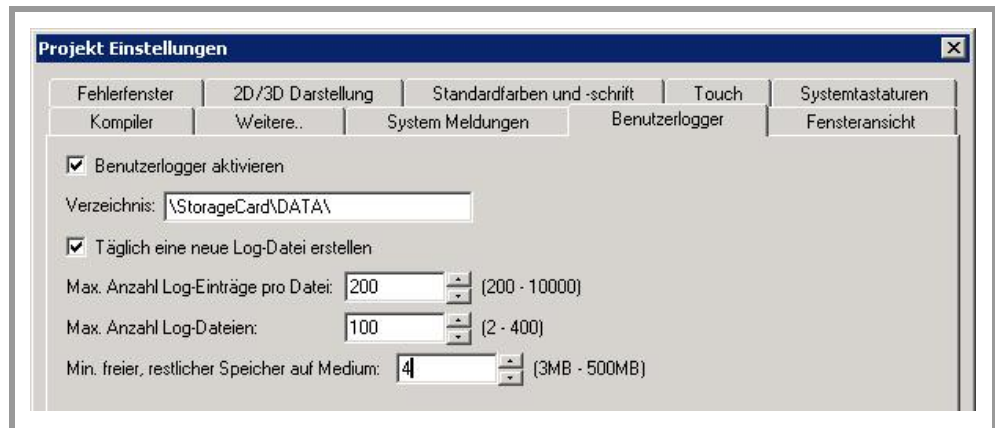


Abb. 11 Projekt Einstellungen

Einstellung	Beschreibung
Benutzerlogger aktivieren	Aktiviert den Benutzerlogger
Verzeichnis	Angaben mit dem Pfad wo die Protokolldateien gespeichert werden sollen. Der Pfad darf nicht das gleiche Medium sein wie jenes, welches auf dem GRSW betrieben wird (Betriebssicherheit). Ansonsten wird dies mit einer Systemfehlermeldung angezeigt und der Benutzerlogger kann nicht verwendet werden.
Täglich eine neue Log-Datei erstellen	Mit dieser Option wird gewährleistet, dass innerhalb einer Protokolldatei keine Einträge unterschiedlicher Tage enthalten sind. Das heisst, dass bei einem Datumswechsel und neuem Log-Eintrag automatisch eine neue Protokolldatei angelegt wird, auch wenn die maximale Anzahl Einträge noch nicht erreicht ist.
Max. Anzahl Log-Einträge pro Datei	Maximale Anzahl Einträge pro Protokolldatei. Wird die maximale Anzahl Log-Einträge erreicht, wird automatisch eine neue Protokolldatei angelegt und der Index um den Wert 1 erhöht. Hat seit der letzten Erstellung einer Protokolldatei ein Datumswechsel stattgefunden, beginnt der Index wieder bei 0.
Max. Anzahl Log-Dateien	Maximale Anzahl Protokolldateien die angelegt werden. Ist die maximale Anzahl erreicht, wird ab dann die älteste Protokolldatei gelöscht, bevor eine neue Protokolldatei angelegt wird.

6 Grundsätzlicher Projektierungs-Ablauf - Logger

6.1 Benutzerlogger Projekt Einstellungen

Einstellung	Beschreibung
Min freier, restlicher Speicher auf Medium	Minimaler freier Speicher auf dem Medium. Dieser muss zur erfolgreichen Erzeugung einer Protokolldatei vorhanden sein. Ist der freie Platz geringer als gefordert, wird bei jedem Versuch die Protokolldatei zu beschreiben eine Systemfehlermeldung angezeigt. Damit die Fehlermeldung nicht mehr erscheint, muss genügend freier Medienspeicher bereitgestellt oder der Logger ausgeschaltet werden.

Tab. 1 Projekt Einstellungen

6.2

Ignore-Liste

Mit der sogenannten Ignore-Liste können Variablen und Funktionen für das Loggen deaktiviert werden.

Es gibt zwei Arten von Ignore-Listen:

- Variablen die nicht protokolliert werden sollen

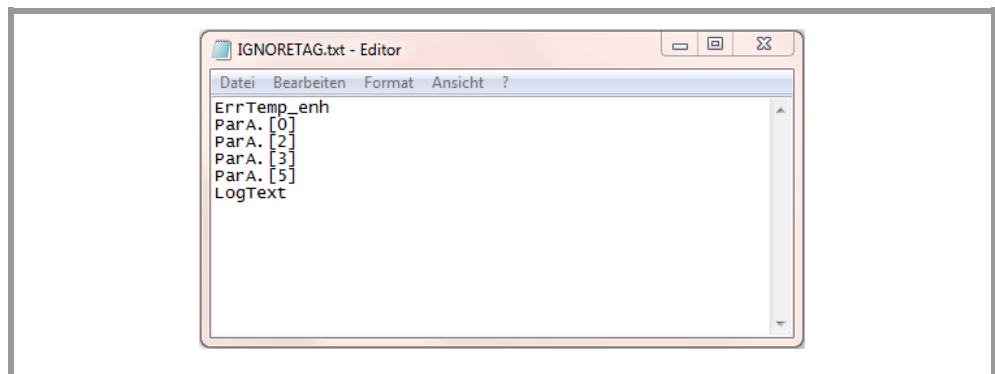


Abb. 12 Ignore-Liste: IGNORETAG.txt

- Funktionen die nicht protokolliert werden sollen

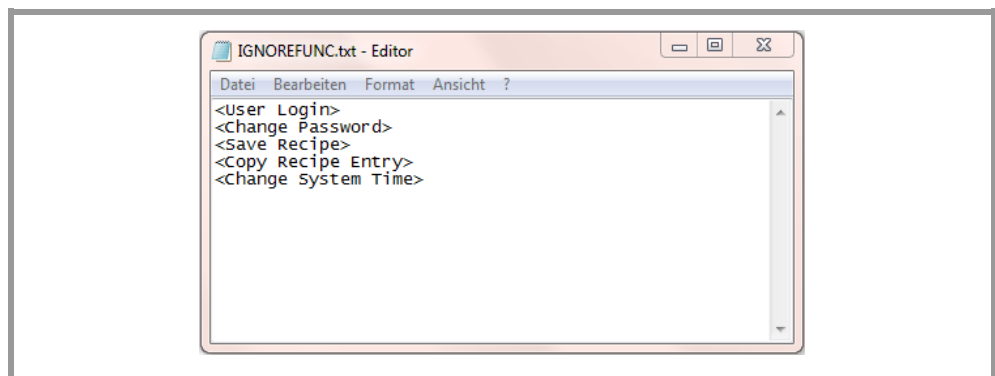


Abb. 13 Ignore-Liste: IGNOREFUNC.txt

Die Dateien haben feste Namen und heissen «IGNORETAG.TXT» bzw. «IGNOREFUNC.TXT» und müssen im ASCII-Format gespeichert sein. Diese Files müssen im Ordner «DATA» abgelegt sein.

Beispiel:

\\StorageCard\\DATA

6 Grundsätzlicher Projektierungs-Ablauf - Logger

6.2 Ignore-Liste

Beim Hochstarten von GRSW werden diese beiden Dateien gelesen (sofern vorhanden) und interpretiert.

Dabei werden die aufgeführten Variablen bzw. Funktionen auf eine interne Ignore-Liste gesetzt, so dass diese dann nicht mehr in den Protokolldateien aufgeführt werden.

Die Namen auf der Ignore-Liste müssen mit den Variablen-Namen, wie sie in der Protokolldatei erfasst sind, übereinstimmen. Die Gross-/Kleinschreibung wird ignoriert.

Pro Zeile muss 1 Eintrag stehen. Jede Zeile muss durch einen Zeilenumbruch getrennt sein.

6.3

Log-File Einträge

Jede Protokolldatei hat am Anfang einen Header der aus folgenden Einträgen besteht:

Log File:	C:\Log\GalileoLog_CONTROLSTATUSDEMO_2012-06-12_014.CSV
Project:	\StorageCard\appl\CONTROLSTATUSDEMO
Program Version:	8.1.0 (12345)

Tab. 2 Log-File Header

Danach folgen die eigentlichen Log-Einträge.

Timestamp	User Unit	Event	Tag/Function Name	Old Value	New Value
2012-05-23 08:00:30		<Info>	<Startlog>		
2012-05-23 08:05:15		<Function>	<Userlogin>		Müller
2012-05-23 14:20:12	Müller m/min	<Tag Change>	<MachineSpeed>	100	120
2012-05-23 14:20:15	Müller	<Function>	<Userlogin>		Meier
2012-05-23 14:50:23	Meier	<Tag Change>	<ValveV1>	0	1
.....					
.....					

Tab. 3 Log-Einträge

6 Grundsätzlicher Projektierungs-Ablauf - Logger

6.4 Spezialfunktionen

6.4

Spezialfunktionen

Funktionstaste mit Parameter.

Gruppe:

- User Logger

Funktion:

- Add String Tag Log
- Add Text Log
- Add Value Tag Log
- Flush Logger
- Logger On/Off
- View Current Log File

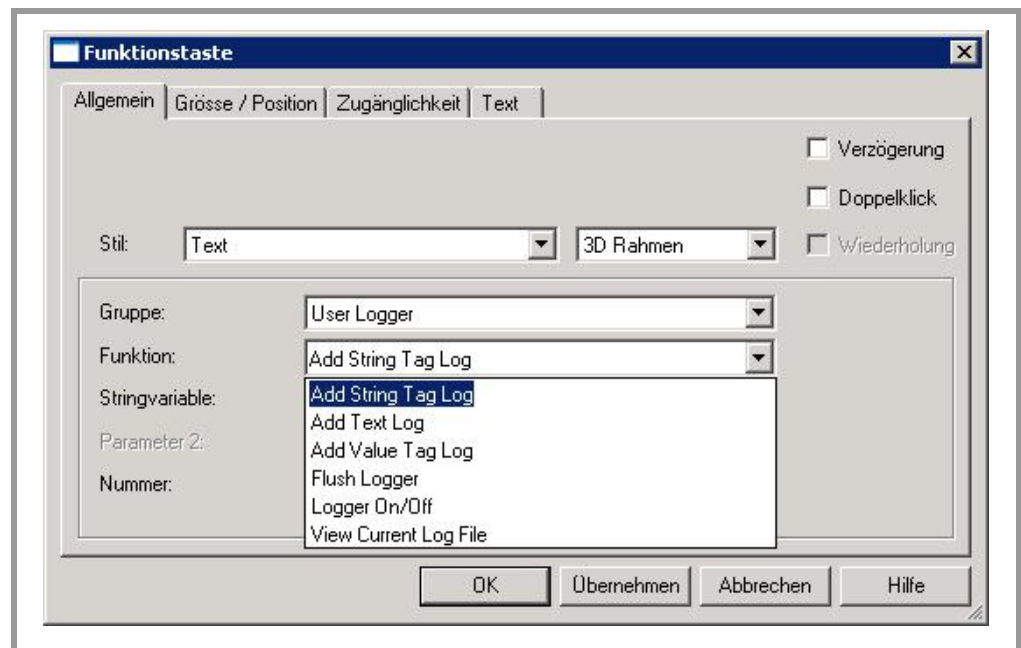


Abb. 14 Spezialfunktionen

6.4.1

Add String Tag Log (Stringvariablen schreiben)

Der Wert einer Stringvariablen kann als Log-Eintrag in die Protokolldatei geschrieben werden.

6.4.2 Add Text Log (definierte Zeichenfolge schreiben)

Eine fix definierte Zeichenfolge kann als Log-Eintrag in die Protokolldatei geschrieben werden.

6.4.3 Add Value Tag Log (Wertvariablen schreiben)

Der Wert einer Wertvariablen kann als Log-Eintrag in die Protokolldatei geschrieben werden.

6.4.4 Flush Logger (Datenbuffer speichern)

Das Speichern der Log-Einträge vom Datenbuffer in die Protokolldatei kann manuell mit dieser Funktion ausgelöst werden (z.B. vor dem Trennen der Speisespannung des Panels).

6.4.5 Logger ON-OFF (Logger EIN / AUS)

Wird das Benutzerloggen gestartet, wird dies mit dem Eintrag „Start Logging“ im Datenbuffer dokumentiert.

Wird das Benutzerloggen beendet, wird dies mit dem Eintrag „Stop Logging“ im Datenbuffer dokumentiert und alle vorhandenen Log-Einträge im Datenbuffer in die Protokolldatei gespeichert.

6.4.6 View Current Log File (Protokolldatei anzeigen)

Die aktuelle Protokolldatei kann mit Hilfe eines externen Programmes (z.B. Notepad CE) angezeigt werden.

Um den in Windows CE 5.0 integrierten Texteditor (ab Version 2.26.1) zu nutzen, kann folgender Parameter eingegeben werden:

- NotepadCE.exe