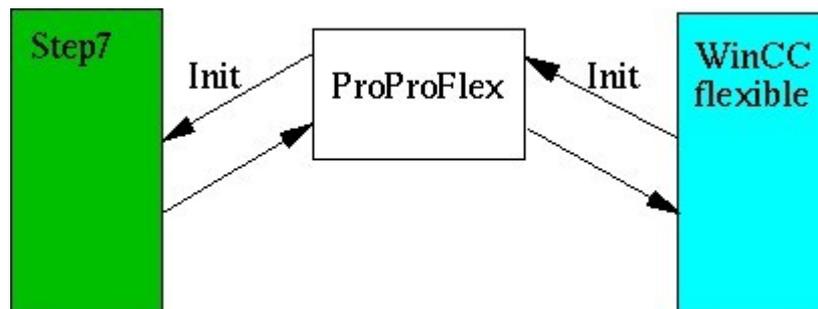


ProPro für WinCC Flexible

(ProProFlex)

Version 1.1
Datum : 21.09.2010



Erstellt mit :

Lazarus 0.9.28.2 r22279 FPC 2.2.4 i386-win32-win32/win64

durch :

© Heisch Automation
Werner Heisch

<http://www.heisch-automation.de>
<http://sites.inka.de/heisch>



Ingenieurbüro für Industrieautomatisierung

Büro: Ostring 15 D 76829 Landau / Pfalz
Postadresse: Im Vorderen Großthal 4 D 76857 Albersweiler / Pfalz
Tel: +49 6341 890117 Fax: +49 6341 890118
e-mail: hwauto@heisch-automation.de www.heisch-automation.de

Inhaltsverzeichnis

Was macht ProProFlex ?.....	3
Allgemeines.....	5
Vorteile:.....	5
- Dokumentation.....	5
- kein Meldetext wird vergessen.....	5
- Geschwindigkeit.....	5
- ProProFlex dient der Gesundheit.....	5
- WinFlex-Display-Kompatibilitätsmodus.....	6
Laufvoraussetzungen.....	7
Zur Version.....	7
Historie.....	8
Erscheinungsbild der vorliegenden Version.....	8
ToDo.....	9
Warnung vor und bei der Benutzung.....	9
Benutzte Warenzeichen.....	9
.....	9
Bedienungsanleitung.....	10
1. Projektstart.....	10
2. Start mit ProProFlex im laufenden Projekt.....	15
3. Rückübersetzung eines Störmelde-DBs aus WinCCflexible.....	16
4. Importieren der Störmeldungen nach WinCCflexible.....	19

Was macht ProProFlex ?

ProProFlex ist ein Hilfsmittel für den (teil-) automatisierten Import von Bit-Meldungen (Stör- und Meldetexte) in das WinCC-Flexible®-Programmiersystem.

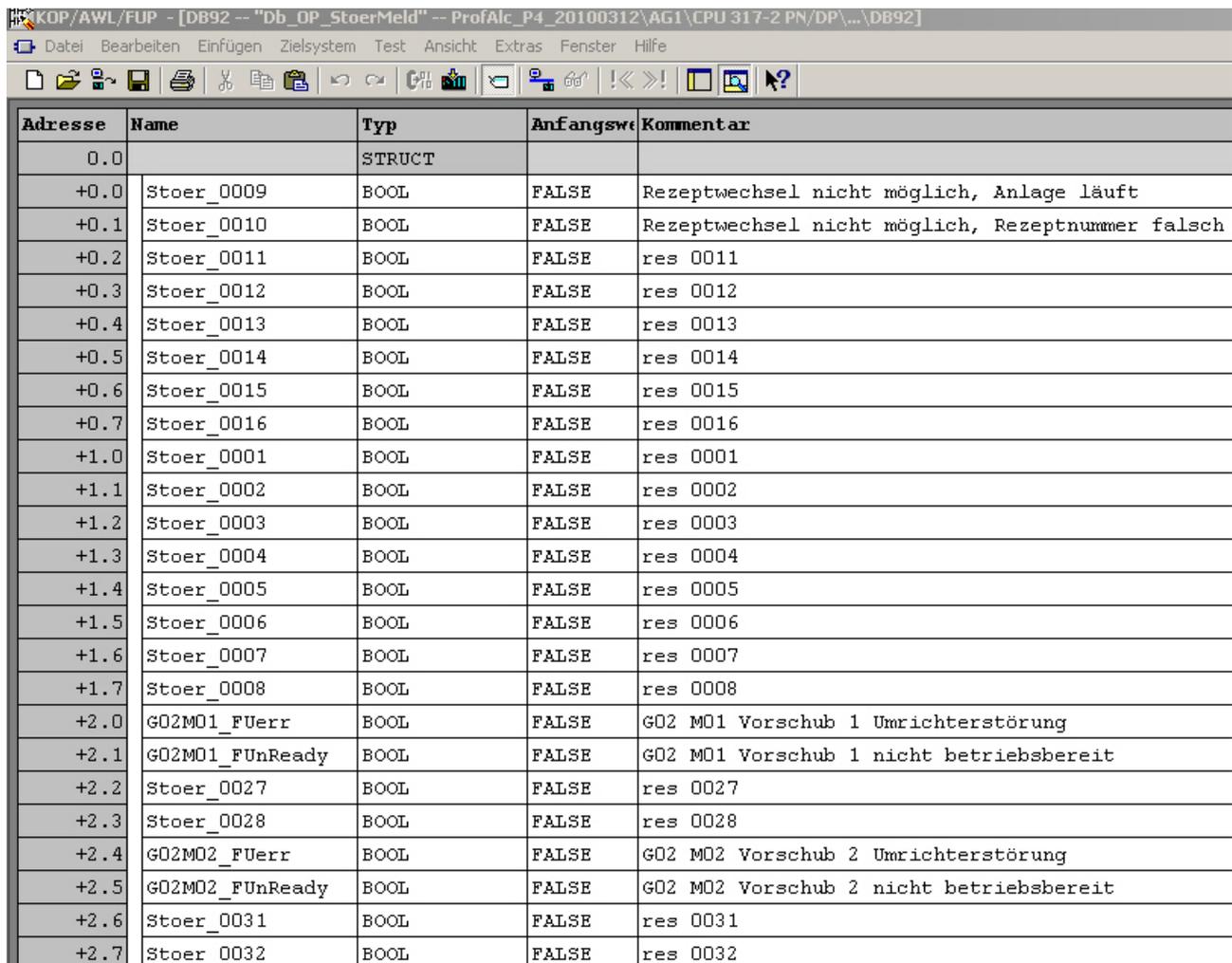
ProProFlex übersetzt einen kommentierten Störmelde-DB in eine WinCCflexible-kompatible Importdatei für Bitmeldungen.

Voraussetzungen:

- Die Bit-Meldungen befinden sich als BOOL-Variable in einem Datenbaustein.
- Der gewünschte Störmelde-Text liegt als Kommentar der Bool-Variablen vor.
- Die Störmeldebits beginnen mit DBX 0.0 des Störmelde-DBs

Die Namen der BOOL-Variablen sind beliebig.

Beispiel:



Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	Stoer_0009	BOOL	FALSE	Rezeptwechsel nicht möglich, Anlage läuft
+0.1	Stoer_0010	BOOL	FALSE	Rezeptwechsel nicht möglich, Rezeptnummer falsch
+0.2	Stoer_0011	BOOL	FALSE	res 0011
+0.3	Stoer_0012	BOOL	FALSE	res 0012
+0.4	Stoer_0013	BOOL	FALSE	res 0013
+0.5	Stoer_0014	BOOL	FALSE	res 0014
+0.6	Stoer_0015	BOOL	FALSE	res 0015
+0.7	Stoer_0016	BOOL	FALSE	res 0016
+1.0	Stoer_0001	BOOL	FALSE	res 0001
+1.1	Stoer_0002	BOOL	FALSE	res 0002
+1.2	Stoer_0003	BOOL	FALSE	res 0003
+1.3	Stoer_0004	BOOL	FALSE	res 0004
+1.4	Stoer_0005	BOOL	FALSE	res 0005
+1.5	Stoer_0006	BOOL	FALSE	res 0006
+1.6	Stoer_0007	BOOL	FALSE	res 0007
+1.7	Stoer_0008	BOOL	FALSE	res 0008
+2.0	G02M01_FUerr	BOOL	FALSE	G02 M01 Vorschub 1 Umrichterstörung
+2.1	G02M01_FUnReady	BOOL	FALSE	G02 M01 Vorschub 1 nicht betriebsbereit
+2.2	Stoer_0027	BOOL	FALSE	res 0027
+2.3	Stoer_0028	BOOL	FALSE	res 0028
+2.4	G02M02_FUerr	BOOL	FALSE	G02 M02 Vorschub 2 Umrichterstörung
+2.5	G02M02_FUnReady	BOOL	FALSE	G02 M02 Vorschub 2 nicht betriebsbereit
+2.6	Stoer_0031	BOOL	FALSE	res 0031
+2.7	Stoer_0032	BOOL	FALSE	res 0032

ProProFlex generiert eine Import-Datei, die dann direkt in WinCC-Flexible® eingelesen werden kann.

Das Ergebnis sieht dann so aus:

Text	Nummer ▲	Meldeklasse
res 0003	2003	Fehler
res 0004	2004	Fehler
res 0005	2005	Fehler
res 0006	2006	Fehler
res 0007	2007	Fehler
res 0008	2008	Fehler
Rezeptwechsel nicht möglich, Anlage läuft	2009	Fehler
Rezeptwechsel nicht möglich, Rezeptnummer falsch	2010	Fehler
res 0011	2011	Fehler
res 0012	2012	Fehler
res 0013	2013	Fehler
res 0014	2014	Fehler
res 0015	2015	Fehler
res 0016	2016	Fehler
G02 M03 Tool 1 UP converter error	2017	Fehler
G02 M03 Tool 1 UP not ready	2018	Fehler
G02 M03 Tool 1 UP active indication error	2019	Fehler
G02 M03 Tool 1 UP punch fault	2020	Fehler
G02 M04 Tool 1 BP converter error	2021	Fehler
G02 M04 Tool 1 BP not ready	2022	Fehler
G02 M04 Tool 1 BP active indication error	2023	Fehler
G02 M04 Tool 1 BP punch fault	2024	Fehler
G02 M01 Vorschub 1 Umrichterstörung	2025	Fehler
G02 M01 Vorschub 1 nicht betriebsbereit	2026	Fehler
res 0027	2027	Fehler
res 0028	2028	Fehler
G02 M02 Vorschub 2 Umrichterstörung	2029	Fehler
G02 M02 Vorschub 2 nicht betriebsbereit	2030	Fehler
res 0031	2031	Fehler

Allgemeines

Vorteile:

- Dokumentation

Zwar müssen die Texte immer noch eingetippt werden, mit Hilfe von ProProFlex aber nur einmal, und zwar direkt im Step7-Programm.

Als Ergebnis kommt "nebenbei" ein gut dokumentierter Störmelde-DB raus.

- kein Meldetext wird vergessen:

Das Erzeugen einer Störung und deren Versorgung in Step7® ist die eine Sache, das zeitnahe Eintippen des Texts in WinCC-Flexible® nochmals eine andere. Gerade, wenn's bei der Inbetriebnahme hektisch wird.

Durch die zeitschonende Übertragung der Texte in WinCC-Flexible® ist es unproblematisch, alleine auf den Verdacht hin, was vergessen zu haben, einen Übersetzungslauf zu machen und das Ergebnis in WinCCFlexible zu übertragen.

- Geschwindigkeit:

Es muß nicht bei jeder neuen Störmeldung WinCC-Flexible® gestartet werden, damit man die Störmeldung, bzw. nicht vergisst. Auch wer WinCC-Flexible® ständig geöffnet hat, wird mit ProProFlex schneller sein, da die nervige, kopf-basierte Cha-Cha-Cha-Adressrechnung entfällt.

- ProProFlex dient der Gesundheit:

Es gibt viele Gründe, sich über WinCC flex zu ärgern. Der Verfasser tut dies aus verschiedenen Gründen. Das ist nicht gut für die Gesundheit.

ProProFlex kann dies zwar nicht komplett verhindern, wird aber zumindest im Bereich der Bitmeldungen zu merklicher Linderung führen, da die Arbeitszeit mit WinCCFlex reduziert wird.

- WinFlex-Display-Kompatibilitätsmodus

Seit Jahren werden immer wieder "sichtbare" d.h. längere Variablenfelder in WinCC flex gefordert, aber Siemens-seitig tut sich nichts, obwohl dies zu einer massiven Arbeitserleichterung für die Anwender führen würde. Der Aufwand wäre minimal .. aber Siemens regt sich nicht.

Ich halte dies für syntomatisch, wie die Siemens-Entwicklung mit Kundenforderungen umgeht. Gerade die Visualisierer sitzen dabei auf dem höchsten Ross.

(Man erinnere sich: Protool/Pro 6.0 , die erste 6er Version, hatte ein Variablenauswahl-Fenster mit 6 (sechs) Zeilen.)

Sollte jemand den " WinFlex-Kompatibilitätsmodus" als Protest verstehen: So ist er gedacht.

Laufvoraussetzungen

ProProFlex wurde unter MS Windows XP® 32Bit, der üblichen Step7-Umgebung, getestet.
Es setzt WinCC Flexible® 2008 voraus, da erst ab dieser Version ein Meldeimport möglich ist.

Zur Version

Version 1.1

Diese Version funktioniert, hat aber sicherlich noch Fehler.

Die deutschen Umlaute funktionieren, Englisch sowieso, Französisch, Spanisch .. keine Ahnung.

Die **DEMO-Version** ist kostenlos und verarbeitet maximal **160 Meldungen**, was für kleinere Projekte auf jeden Fall ausreicht.

Die **Kaufversion** verarbeitet **10000 Meldungen**. (Falls jemand mehr braucht, wird "aufgebohrt".)

Historie

Das ursprüngliche ProPro wurde von Gerald Emig, Emig-Software, geschrieben.

Es hatte die gleiche Funktion wie ProProFlex: Die Aufbereitung von Störmeldetexten zum Import in die Projektierungssoftware für OPs.

ProPro bereitete die Import-Datei für **Protool/Pro®** vor.

Nachdem Protool durch WinCCFlexible® abgelöst wurde, war erst mal wieder Handarbeit angesagt.

Nachdem mit WinCC-Flexible® 2008 endlich wieder ein Import möglich ist, habe ich lange auf eine neue ProPro-Version gewartet, aber scheinbar ist Gerald Emig bis heute noch nicht dazu gekommen.

Also habe ich selbst in die Tasten gegriffen .. was wiederum Auswirkungen auf's Erscheinungsbild hat.

Erscheinungsbild der vorliegenden Version

Die nicht Windows-Standard-konforme Benutzeroberfläche ist aus der Not geboren:

Zwar habe ich früher viel in Pascal geschrieben, aber dies ist das erste Windows-Programm (MS-Windows®), das ich schreibe, und zwar in "Lazarus".

"Lazarus" basiert auf Free-Pascal und ist für verschiedene Betriebssysteme verfügbar.

Die "Innereien", Codewandlung, etc, waren also kein Problem, sehrwohl aber die Erstellung der Oberfläche :

Da es bei ProProFlex um ein Programm handelt, das nur eine Detail-Funktion der eigenen täglichen

Arbeit übernehmen soll, habe ich darauf verzichtet, unbedingt die übliche "Datei" "Eigenschaften" - Kopfleiste realisieren zu wollen.

Lazarus linkt normalerweise statisch, und zwar alles, was man vielleicht irgendwann mal brauchen könnte. Dies erklärt auch die der Aufgabenstellung unangemessene Größe des Programms.

Ich hoffe, mit Hilfe dieser Beschreibung kommt man trotzdem gut klar.

ToDo

Bei Bedarf, bzw. entsprechender Nachfrage:

Einbau von Makros, um BYTE, WORD oder DWORD Variablen zu Bitmeldungen expandieren zu können.

Z.B. könnte ein per FC / FB realisierter (FU-)Antrieb als Parameter ein Byte für die Störmeldungen enthalten, die Expansion der Meldungen zur Bit-Ebene kann ProProFlex übernehmen.

Was bei der Übersetzung zwischen den verschiedenen Zeichensätzen noch zu erledigen ist, wird sich zeigen.

Warnung vor und bei der Benutzung

Hier die üblichen Warnungen zur rechtlichen Absicherung:

Hinweise für Benutzer in den USA:

Das Programm kostet Festplattenspeicherplatz, bei Betrieb zusätzlich noch Rechenzeit, außerdem wird Hauptspeicherplatz belegt. Darüber hinaus werden Daten erzeugt.

Wir werden keinesfalls Schadenersatzforderungen im 3-stelligen Milliardenbereich bezahlen.

Falls Sie damit nicht einverstanden sind: Benutzen Sie das Programm nicht !

Hinweis an alle Benutzer:

Da es keine fehlerfreien Programme gibt, ist auch dieses Programm nicht fehlerfrei.

Die Benutzung ist daher auf eigenes Risiko.

Benutzte Warenzeichen:

CPM/86® war ein Warenzeichen von Digital Research, später gekauft von Novell, der derzeitige Inhaber des Warenzeichens ist mir nicht bekannt, vermutlich noch Novell.

MS-Dos® ist ein Warenzeichen der Microsoft Corporation.

MS-Windows® ist ein Warenzeichen der Microsoft Corporation.

Step7® ist ein Warenzeichen der Siemens AG

WinCCFlexible® ist ein Warenzeichen der Siemens AG

Bedienungsanleitung

Generell:

ProProFlex unterstützt die Verwaltung der 3 Störmeldungsbereiche "Triggervariable", "Quittiervariable Lesen" und "Quittiervariable Schreiben"

Die Triggervariable muß sein, die beiden anderen Bereiche können in den jeweiligen Masken von ProProFlex an- oder abgewählt werden.

Zur Beachtung:

- !! Obwohl dies die WinCCflex-seitige Nomenklatur sein sollte, da der Blickpunkt von WinCCflex aus geht:
"Quittiervariable LESEN" ist der Bereich, in den das OP seine Quittierbits SCHREIBT.
"Quittiervariable SCHREIBEN" ist der Bereich, aus dem das OP die Quittierbits aus dem AG LIEST.
- !! Es ist eine Systemeigenschaft von WinCCflexible, daß die "Quittiervariable SCHREIBEN" in gleichen Bitblock wie die "Triggervariablen" liegen müssen.
- !! Die in WinCCFlex definierten Datenblöcke müssen vom Typ WORD, DWORD bzw. deren Arrays sein. **ProProFlex setzt WORD-Arrays voraus.**
- !! In verschiedenen Beschreibungen steht, daß bei Nutzung von "Quittiervariable SCHREIBEN" die maximale Blocklänge jeweils 16 Bit beträgt. Das ist Nonsense.

1. Projektstart

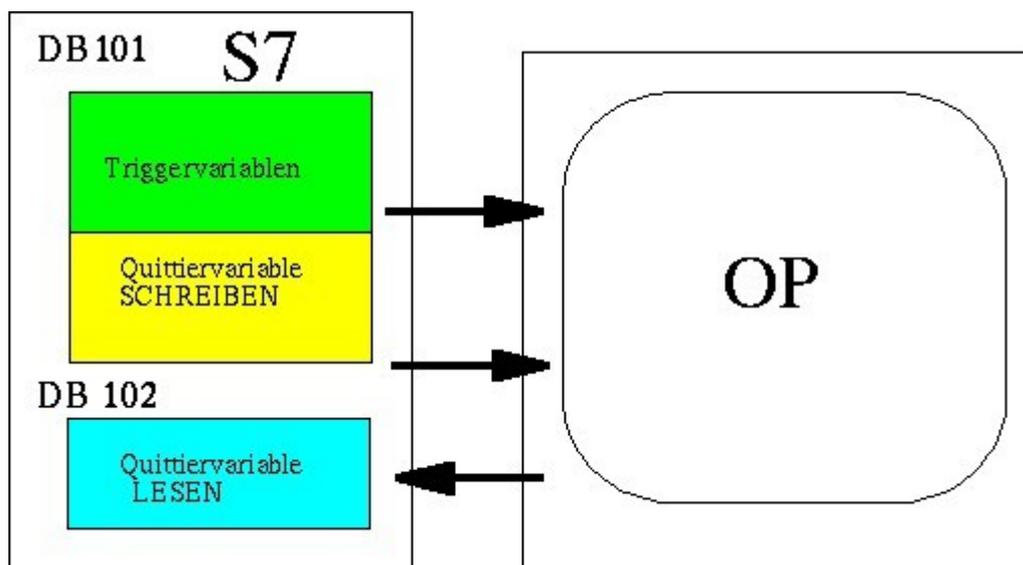
Beispiel:

Sie wollen ein neues Projekt anlegen, es sollen 160 Störmeldungen vorgesehen werden.
Die Meldungen sollen sowohl von dem OP als auch von dem AG aus quittiert werden können.

-> Sie benötigen dazu 3 Datenbereiche

1. Die Störmeldebts ("Triggervariablen")
2. die Quittierbits zur Quittierung OP-> AG ("Quittiervariable lesen" , Ack HMI)
3. die Quittierungen von AG zu OP ("Quittiervariable schreiben" Ack PLC)

Sie haben für die Störmeldebts ("Triggervariablen") den DB 101 vorgesehen.
Ihre "Quittiervariable LESEN" können wo anders liegen, sie wählen den DB 102.

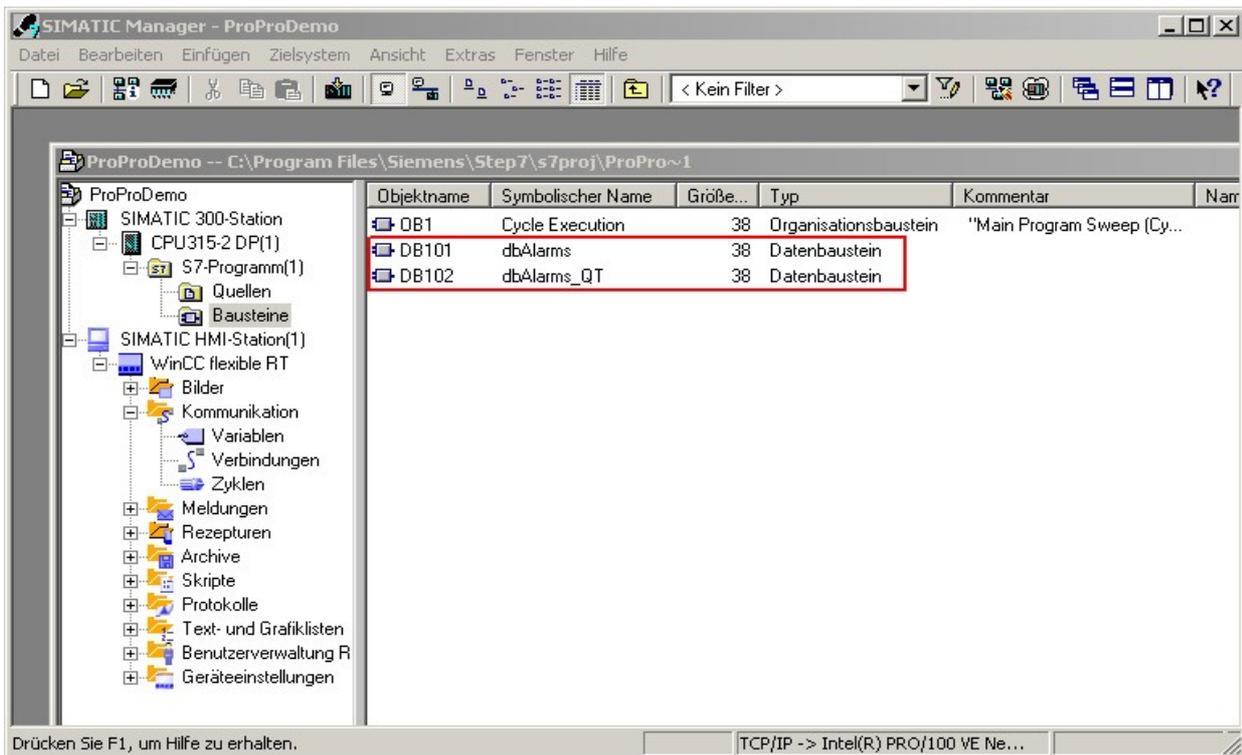


1.1 Erstellung der Störmelde-DBs in Step7

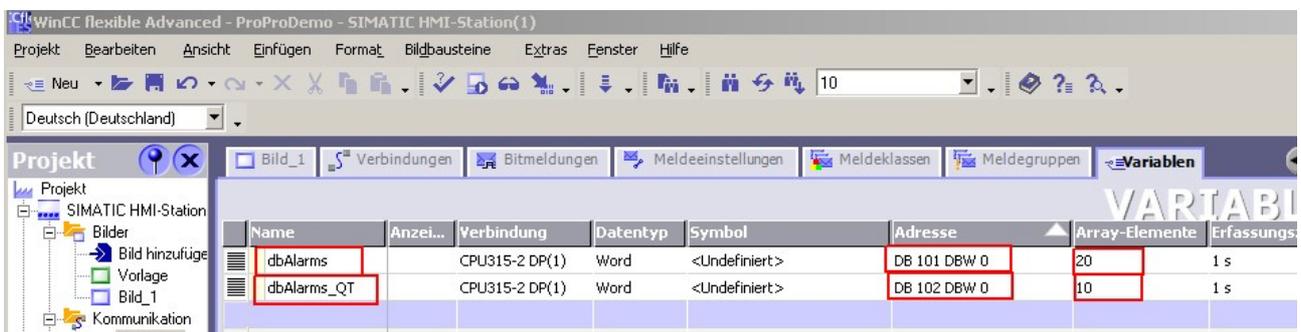
Erstellen Sie in Ihrem Step7-Projekt die beiden DBs.

- Beim eigentlichen Stoermelde-DB (DB 101, Trigger-Variable und QuittiervarSchreiben) ist Länge und Format egal, sie werden ohnehin gleich geändert
- Beim Quittier-DB (QuittiervarSchreiben) sollten Sie gleich die richtige Länge anlegen, er wird von ProProFlex nicht bearbeitet, lediglich registriert. .

Legen Sie gleich die Zuordnungsliste dafür an:



1.2 Anlegen der Variablenblöcke in WinCCflexible



Zur Beachtung:

Obwohl Sie 160 Meldungen anlegen wollen, also $10 * 16 \text{ Bit} = 10 * \text{WORD}$, muß die Variable 'dbAlarms' 20 WORDS umfassen, da darin sowohl die "Triggervariablen" als auch die "Quittiervariablen SCHREIBEN" enthalten sind.

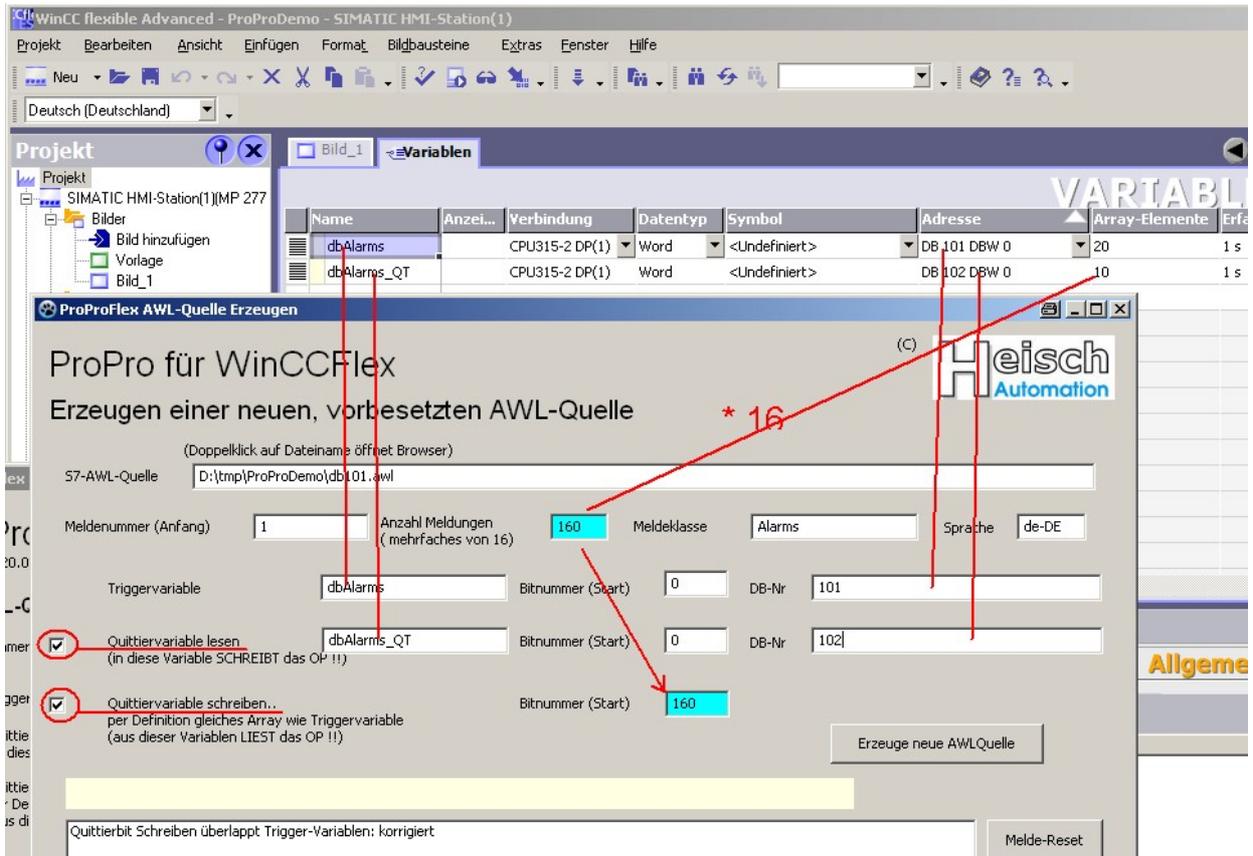
(Sollte kein Quittierung vom AG zu HMI gewünscht sein, dann natürlich 10 Worte.)

1.3 Initialisierung des Störmelde-DBs mit ProProFlex

Da zum späteren Erzeugen der WinCC-Flexible-Importdatei verschiedene Informationen bekannt sein müssen, werden dies der Einfachheit halber im Kommentarbereich des Störmelde-DBs abgelegt.

Starten Sie ProProFlex und gehen Sie zu "Neue AWL-Quelle erstellen"

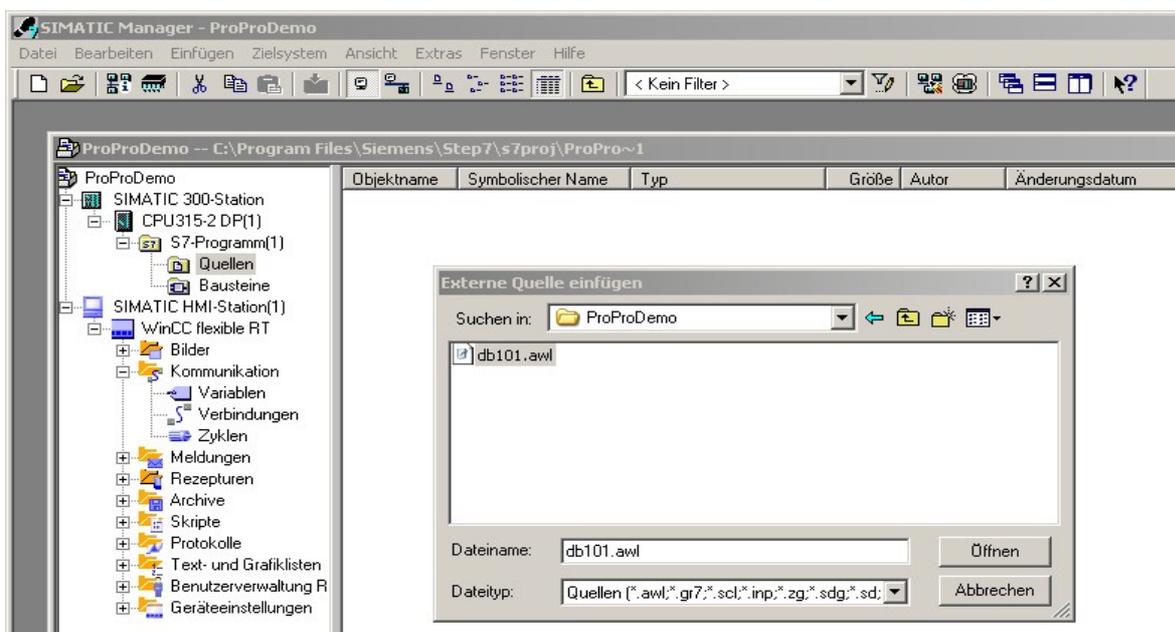
und tragen Sie dort die notwendigen Informationen ein, wie sie auch in WinCC flexible stehen :



Wählen Sie eine neue Zielfeld aus und speichern Sie sie ab. Die Dateierdung muss .AWL sein, denn es ist eine AWL-Quelle, die in Step7 importiert wird.

1.4 Import der initialisierten AWL-Quelle in Step7

Importieren Sie die neu erstellte AWL-Quelle in Step7 :
Im Ordner "Quellen": Einfügen -> Externe Quelle ..



Wenn man den AWL-Editor öffnet, sieht man die oben bereits erwähnten Zusatzinformationen im Kommentarbereich der AWL-Quelle.

Übersetzen Sie die AWL-Quelle:

Datei -> Übersetzen

Der Störmelde-DB befindet sich nun im "Baustein"-Ordner.

Passen Sie die Länge des DB 102 noch auf die notwendigen 10 Worte an.

Damit sind die Vorbereitungen abgeschlossen.

Sie können nun während der Projektierung die gewünschten Störmeldetexte im Kommentarfeld des entsprechenden Bits eintragen.

2. Start mit ProProFlex im laufenden Projekt

Sollten Sie bereits einen Störmelde-DB (Voraussetzung: Bit-Array !) erstellt haben, der den Anforderungen von ProProFlex genügt und deshalb zwischendrin auf Proproflex umsteigen wollen:

Erstellen Sie einfach, wie oben beschrieben, eine AWL-Quelle, und importieren Sie sie in ihr Step7- Projekt.

Erstellen Sie auch eine AWL-Quelle Ihres bereits vorhandenen Störmelde-DBs.
Kopieren Sie den Kommentarblock zwischen "ProProFlex_BEGIN" und "ProProFlex_END" an die entsprechende Stelle Ihres 'echten' Störmelde-DBs.

Übersetzen Sie ihren 'echten' Störmelde-DB.
Der Störmelde-DB befindet sich nun wieder im "Baustein"-Ordner und der ProProFlex -Header ist eingebaut.

Sie können nun die gewünschten Störmeldetexte im Kommentarfeld des entsprechenden Bits eintragen.

3. Rückübersetzung eines Störmelde-DBs aus WinCCflexible.

!! VORBEMERKUNG

Die Rückübersetzung einer WinCCflex-Exportdatei in einen Störmelde-DB wird nur gebraucht, wenn es Sinn macht, in einem schon fortgeschrittenen Projekt "mittendrin" auf ProProFlex-basierte Störmeldungen umzusteigen.

Problem:

Es gibt in dieser Richtung nahezu beliebig viele Möglichkeiten:

Viele Triggervariable, die irgendwo liegen, ständiger Wechsel Triggervariable - "Quittvariable Schreiben", zwischen den Blöcken die "Quittvariable lesen", die ja beliebig stehen kann, gemischt mit anderen Daten u.s.w.

Dies alles wäre auch noch aufbereitbar, aber der notwendige hohe Aufwand wäre nicht mehr angemessen.

Sollten sich in einem DB mehrere Triggervariablenbereiche befinden, funktioniert die ProProFlex-Vorwärtsrichtung (DB -> WinCCflex) nicht, es wäre also obsolet, dies abzudecken.

Ansatz:

ProProFlex beschränkt sich überwiegend auf die rudimentäre Aufgabe, den Anwender von dem Tippen der Störmeldetexte in den / die Störmelde-DBs zu entlasten.

Es wird davon ausgegangen:

Ein Triggervariablenblock (mit oder ohne "Quittvariable Schreiben") belegt einen DB.

Für diese Quittvariable wird eine DB erzeugt.

Es sind natürlich mehrere Triggervariablenblöcke, jeder in einem eigenen DB, möglich.

3.1 Export der Bitmeldungen aus WinCCflexible

Gehen Sie zu WinCCflexible und schauen Sie sich die Störmeldebereiche an:

The image shows two screenshots from the WinCCflexible software interface. The top screenshot displays the 'BITMELDUNGEN' table, and the bottom screenshot displays the 'VARIABLEN' table.

BITMELDUNGEN Table:

Text	Nu...	Meldek...	Trigger-Variab...	Bitn...	Hilfet...	Meldegruppe	Quittvariable Lesen	Bit Qu...	Quittvariabl...
res 0001	2001	Fehler	PT_Störmeldungen 1	0		<Keine Melde...	PT_Quittierung-OP 1	0	PT_Störmeldu...
res 0002	2002	Fehler	PT_Störmeldungen 1	1		<Keine Meldegru...	PT_Quittierung-OP 1	1	PT_Störmeldung...
res 0003	2003	Fehler	PT_Störmeldungen 1	2		<Keine Meldegru...	PT_Quittierung-OP 1	2	PT_Störmeldung...

VARIABLEN Table:

Name	Anzei...	Verbindung	Datentyp	Symbol	Adresse	Array-Elemente	Erfassungszykl...	Kommentar
PT_Störmeldungen 1		AG1	Word	<Undefiniert>	DB 92 DBW 0	128	1 s	
PT_Quittierung-OP 1		AG1	Word	<Undefiniert>	DB 93 DBW 0	64 64 Worte = 1024 Bit	1 s	
Db_OP_IO_mpx.OP...		AG1	Byte	<Undefiniert>	DB 96 DBB 21	1	1 s	Endschalter Rich

Dann müssen die in WinCCflexible hinterlegten Störmeldetexte exportiert werden.

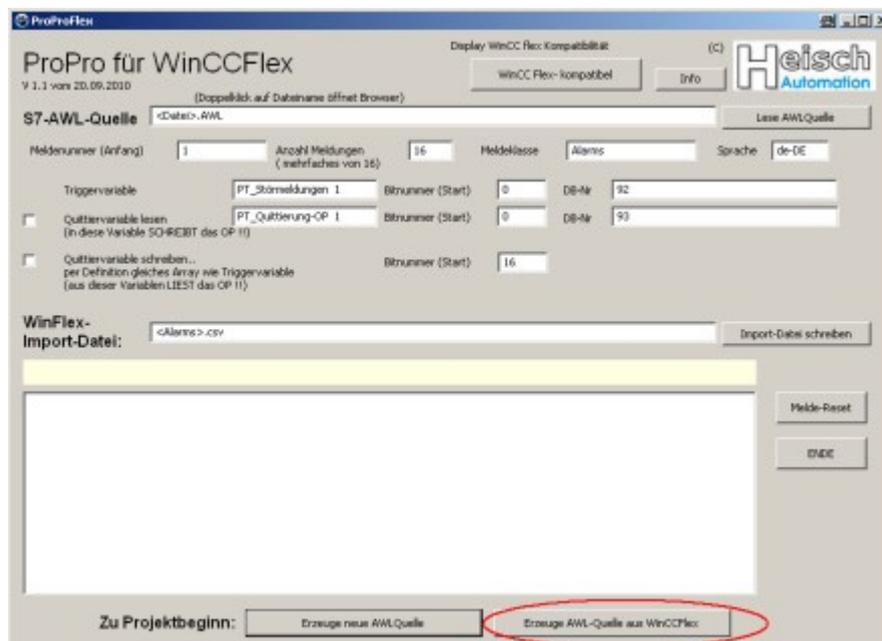
!!! Es macht Sinn, für jedes OP/TP ein eigenes Unterverzeichnis anzulegen, da die Dateinamen fest vorgegeben sind.

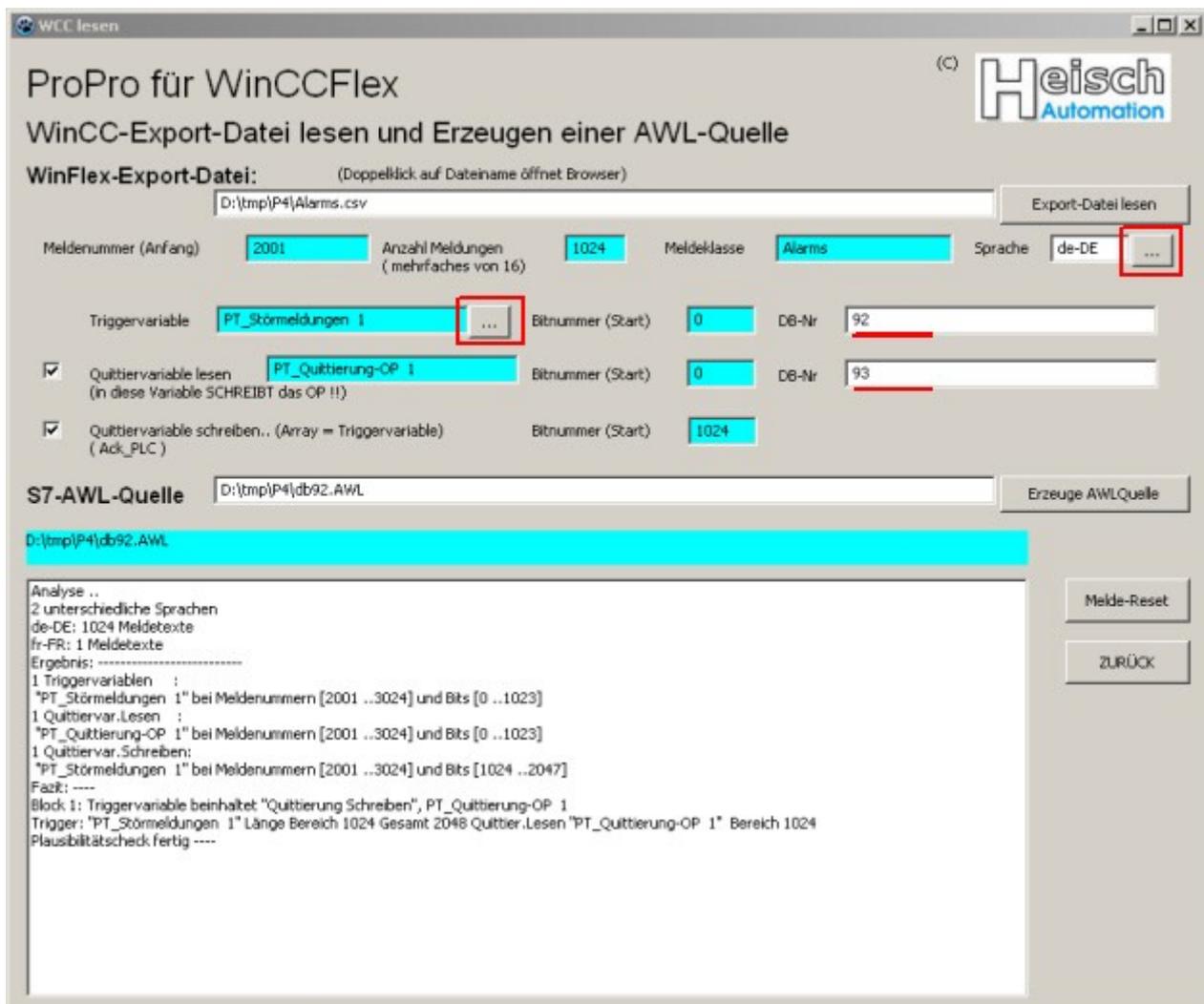
Projekt -> IM-/Export -> CSV Export



3.2 Übersetzen der exportierten Meldungen mit ProProFlex in eine AWL-Quelle

Wählen Sie in der Hauptmaske "Erzeuge AWL-Quelle aus WinCCFlex" an..





Wählen Sie die exportierte Meldedatei aus und lesen Sie sie ein.

ProProFlex wird sie analysieren und daraus die einen Vorschlag erarbeiten (blaue Felder).

Falls die Exportdatei mehrere Triggervariablen enthält, kann die entsprechende mit dem Feld (rot) ausgewählt werden. Gleiches gilt für die unterschiedlichen Sprachen.

Die DB-Nummerern müssen von Hand eingetragen werden, (s.o.)

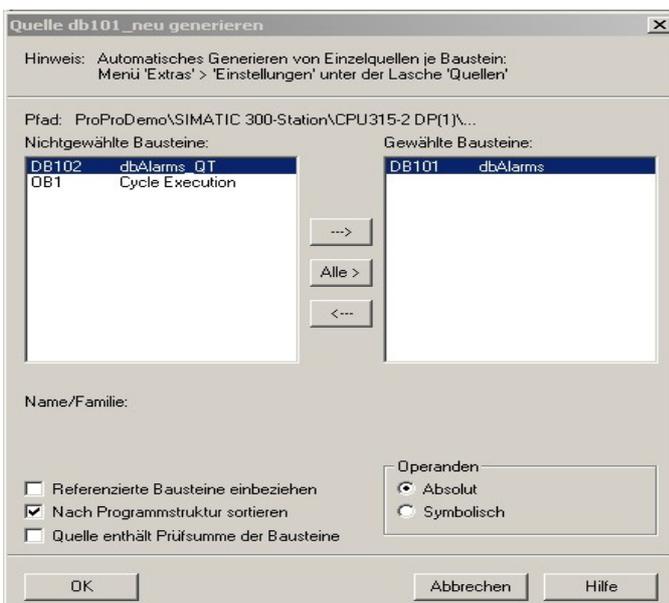
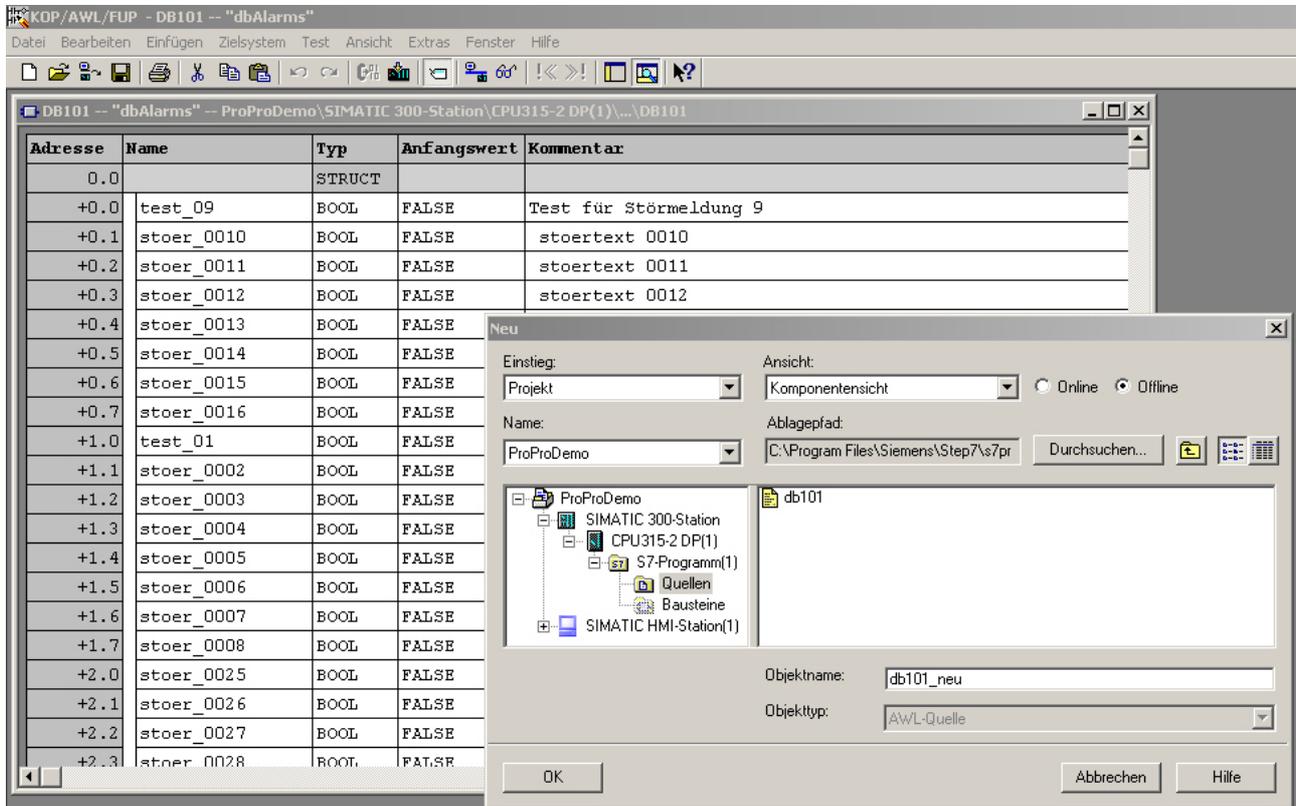
!!! ProProFlex kann dann, wenn keine "Quittiervariable Schreiben" benutzt wird, nicht immer auf die Meldeanzahl schliessen. Bitte kontrollieren Sie die Meldeanzahl zur Sicherheit an Hand der Variablen-Definition in WinCC flex. (Siehe oben)

Wenn alles passt:

Speichern Sie die AWL-Quelle des Störmelde-DBs. Er kann dann wie unter 1.4 beschrieben, in Step7 importiert werden.

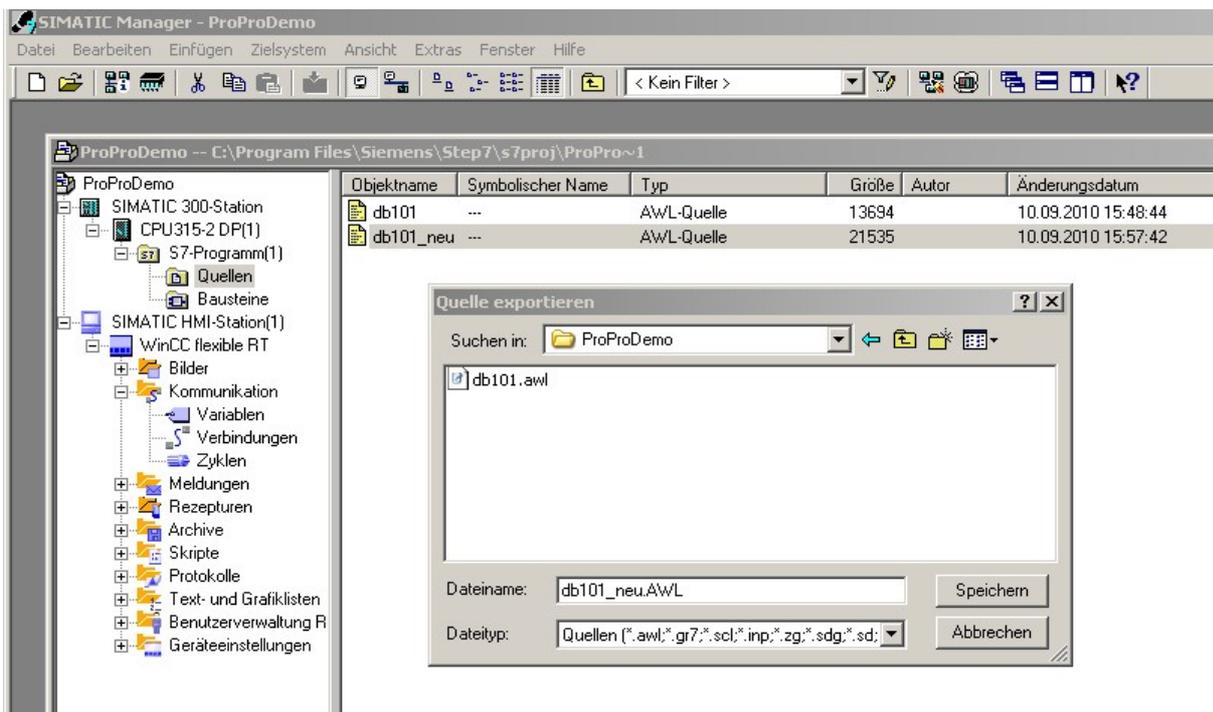
4. Importieren der Störmeldungen nach WinCCflexible.

Übersetzen Sie Ihren Störmelde-DB in eine AWL-Quelle:
Öffnen Sie den DB, dann Datei ->Quelle generieren ..



Die AWL-Quelle befindet sich nun noch in Step7, im Ordner Quellen.
Von dort wird sie raus-exportiert:

in "Quellen" : Bearbeiten -> Quelle Exportieren ...



Die Quelle liegt nun in einem Unterverzeichnis und wird mit ProProFlex weiterbearbeitet:

Einlesen der AWL-Quelle und übersetzen in eine WinCCflex-Import-Datei.

Anwählen der zu lesenden AWL-Quelle und einlesen.

An Hand der Informationen im DB-Kopf der AWL-Quelle werden die entsprechenden Felder gesetzt (gelb) und festgestellt, ob Quittvariablen Lesen und Schreiben benutzt werden.

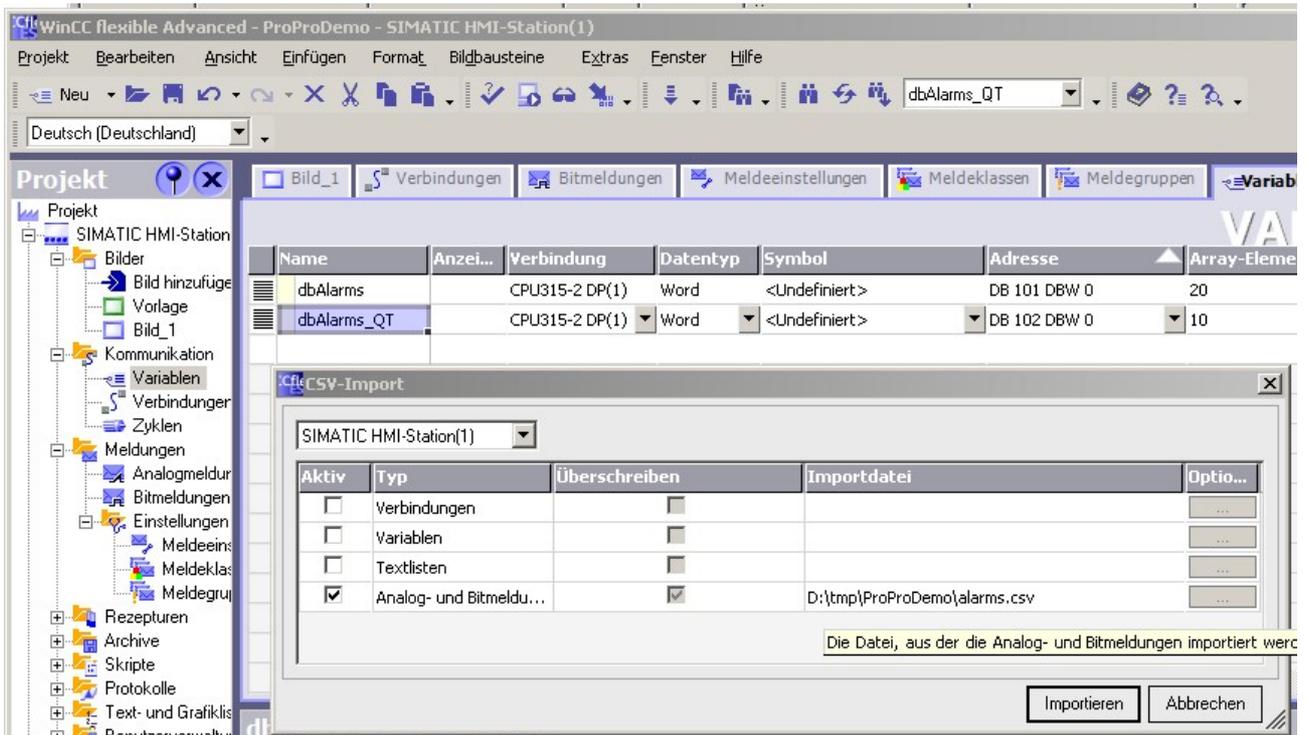
Die Werte können noch geändert werden.

Dann wird die WinCCflex-Import-Datei erstellt.

Einlesen der Import-Datei in WinCC flexible

In WinCCflexible:

Projekt -> Im /Export -> CSV-Import ->



... die Meldungen sind importiert:

