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 Ostring 15

Postadresse: Im Vorderen Großthal 4 +49 6341 890117 e-mail hwauto@heisch-automation.de www.heisch-automation.de

D 76829 Landau / Pfalz D 76857 Albersweiler /Pfalz Fax: +49 6341 890118

## 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	0
1. Projektstart	0
2. Start mit ProProFlex im laufenden Projekt	5
3. Rückübersetzung eines Störmelde-DBs aus WinCCflexible	6
4. Importieren der Störmeldungen nach WinCCflexible1	9

## 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:
-----------

🔣 KOP/AWL/	FUP - [DB92 "Db_OP_Sto	perMeld" ProfAlc_	P4_20100312	(\AG1\CPU 317-2 PN/DP\\DB92]
🖬 Datei Bea	arbeiten Einfügen Zielsystem	n Test Ansicht Extr	as Fenster I	Hilfe
🗋 🖻 🖥	🖬 🎒 🕺 🖻 🛍 🗠	C   C 🕯 🏜 🖂	9 60   !«	»!   🗖 🖪   😢
Adresse	Name	Тур	Anfangswe	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	GO2MO1_FUerr	BOOL	FALSE	GO2 MO1 Vorschub 1 Umrichterstörung
+2.1	G02M01_FUnReady	BOOL	FALSE	GO2 MO1 Vorschub 1 nicht betriebsbereit
+2.2	Stoer_0027	BOOL	FALSE	res 0027
+2.3	Stoer_0028	BOOL	FALSE	res 0028
+2.4	GO2MO2_FUerr	BOOL	FALSE	GO2 MO2 Vorschub 2 Umrichterstörung
+2.5	G02M02_FUnReady	BOOL	FALSE	GO2 MO2 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:

	[	Abarra
lext	Nummer	Meldeklasse
res 0003	2003	Fehler
res 0004	2004	Fehler
res 0005	2005	Fehler
ires 0006	2006	Fehler
ires 0007	2007	Fehler
ires 0008	2008	Fehler
Rezeptwechsel nicht möglich, Anlage läuft	2009	Fehler
Rezeptwechsel nicht möglich, Rezeptnummer falsch	2010	Fehler
ires 0011	2011	Fehler
ires 0012	2012	Fehler
ires 0013	2013	Fehler
ires 0014	2014	Fehler
ires 0015	2015	Fehler
ires 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
E 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-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 Betriebsysteme 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: Benutzten 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<sup>®</sup> ist ein Warenzeichen der Siemens AG WinCCFlexible<sup>®</sup> ist ein Warenzeichen der Siemens AG

ProProFlex V1.1

## Bedienungsanleitung

#### Generell:

ProProFlex unterstützt die Verwaltung der 3 Störmeldungebereiche "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 Datenblocke 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 Störmeldebits ("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 Störmeldebits ("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:

SIMATIC Manager - ProProDemo	Ansicht Extra	s Fenster Hilfe				_	
🗋 🗅 🥔 🔡 🛲 🛛 X 🖻 🛍 🕍		1- 11 m   🔁	< Kein Filt	er>	- 🏹   🔡 8		▶?
ProProDemo C:\Program Fil	es\Siemens\S	tep7\s7proj\ProPro	~1	_		_	
ProProDemo	Objektname	Symbolischer Name	Größe	Тур	Kommen	tar o ro	Nam
Er-1 Storstation Er-1 CPU315-2 DP(1) Er-1 S7-Programm(1)	DB101	dbAlarms	38	Organisationsbaus Datenbaustein	tein "Main P	rogram Sweep (Cy	
D Quellen	06102	doelams_01	30	Datenbaustein			
SIMATIC HMI-Station(1)							
⊕ <del>Z</del> Bilder ⊡ 🧏 Kommunikation							
Variablen							
E December							
Environmentalisten							
Drücken Sie F1, um Hilfe zu erhalten.			TC	P/IP -> Intel(R) PR	O/100 VE Ne		

## 1.2 Anlegen der Variablenblöcke in WinCCflexible

WinCC flexible Advance Projekt <u>B</u> earbeiten <u>A</u> nsi	d - Pr cht	oProDemo - SIM <u>Ei</u> nfügen Forma	IATIC HMI <u>t</u> Bil <u>d</u> bau	-Station(1) steine E <u>xt</u> ras	<u>F</u> enster <u>H</u> ilf	e					
🛛 📲 Neu 🔸 📂 🧮 💋 🤊	0	• × X 🖬 I	ñ .   🎸	5 🖬 🐐 -	🕴 .   Mi	. # 5 4	10	•	Ø ?∎	2.	
Deutsch (Deutschland)	• •										
Projekt 💡 🗙		Bild_1 S	rbindungen	Bitmeldung	ien 🎽 Mel	deeinstellungen	🙀 Meldel	klassen 🔀 Mel	degruppen	<b>⊲≣Variablen</b>	
Projekt								i Gana		VAR	TABL
📄 👆 Bilder		Name	Anzei	Verbindung	Datentyp	Symbol		Adresse	A 🔺	rray-Elemente	Erfassungs
	E	dbAlarms		CPU315-2 DP(1)	Word	<undefiniert></undefiniert>		DB 101 DBW 0	2	0	1 s
Bild 1		dbAlarms_QT		CPU315-2 DP(1)	Word	<undefiniert></undefiniert>		DB 102 DBW 0	1	0	1 s
E Kommunikation											

Zur Beachtung:

Obwohl Sie 160 Meldungen anlegen wollen, also 10 \* 16 Bit, = 10 \*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"

AMA Qualla	(Doppe	Nick auf Dateinane öffnet Dro	wser)			33679		e AWA Causte
		Arrested Malite press	14	Maldaliana	[ Alarma		- Countra	de DE
And the letters	a contraction of	( mehrfaches von 36)	1	( diversition	Instant		April 10	love
Triggervariable		PT_Störmeldungen 1	Ditrummer (Rart)	0	DØ-Nr	92		
Quittiervarisble (in diese Variabl	lesen somREBIT das O	PT_Qutterung-OP 1 P IQ	Bithummer (Start)	0	D8-Nr	93		_
Quittiervariable per Definition gl	schreiben schreis Array wie T	rigger variable	Ditnummer (Start)	16				
nElav.								
port-Datei:	<alams>.csv</alams>						Impor	t-Datei schreiber
					_		_	Melde Reset
								ENDE

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

## ProProFlex V1.1

2	1 Wir	nCC flexible Ad	vanced - F	ProProDe	emo - SIMi	TIC HMI-Statio	n(1)							
-	Projek	: <u>B</u> earbeiten	<u>A</u> nsicht	Einfüge	n Forma <u>t</u>	Bil <u>d</u> bausteine	E <u>x</u> tras <u>F</u> enste	er <u>Hi</u> lfe						
	P	leu 🔻 📂 開	10-0	- X	XI		₩	· • • • •	11 <del>5</del> 11		<b>•</b>	• 🔗 ?=	3	
	Deut	sch (Deutschlan	d) 💌 🗸											
	Proj	ekt	(9	×	Bild_1	<b>⊲≣Variablen</b>								۲
	Pr	ojekt SIMATIC HMI-S ➡ Bilder ➡ Bild hin	Station(1)(MI zufügen	P 277	Name	Anzei	Verbindung	Datent;	/p Symbol	ut 2	A	dresse		
		- 🔲 Vorlage			dbAlar	ms OT	CPU315-2 DP(	I) Word	<undefinier< td=""><td>rt&gt;</td><td></td><td>3102 DBW 0</td><td>10</td><td>15</td></undefinier<>	rt>		3102 DBW 0	10	15
		Bild_1						,						
e	F	ProPro Erzeugen	für V einer	VinC neue	CCFI€ en, ∨or	X besetzten fnet Browser)	AWL-Que	elle	*16	/	(C)		isch omation	
۲ ٥٠. 	с с	Meldenummer ( Triggerv	Anfang) ariable	1	dbAlar	Anzahl Meldunge (mehrfaches vor	n 160 Bitnummer (S	Meldeklas bart) 0	se Alarms DB-Nr	101		Sprathe o	Je-DE	
m	er 🕑	Quittierv (in diese	ariable lese Variable SC	n HREIBT d	dbAlar as OP !!)	mls_QT	Bitnummer (S	tart) 0	DB-Nr	102				Allgeme
99 itt di	er (	Quittierv per Defir (aus dies	ariable schr nition gleiche ser Variabler	eiben es Array v h LIEST da	vie Triggerva as OP !!)	riable	Bitnummer (S	tart)			Erzeuge neu	ie AWLQuelle		Angeine
itt 1 D	ie ie													
JS	di	Quittierbit Sch	reiben überl	appt Trigg	ger-Variabler	n: korrigiert						Mel	de-Reset	

Wählen Sie eine neue Zieldatei aus und speichern Sie sie ab. Die Dateiendung 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 ..

SIMATIC Manager - ProProDemo		- 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1			
Datei Bearbeiten Einrugen Zielsystem		s Fenster Hilfe	< Kein Filter >	V 198	
ProProDemo C:\Program File	s\Siemens\S	tep7\s7proj\ProPro	-1		
ProProDemo 	Objektname	Symbolischer Name	Тур	Größe Autor	Änderungsdatum
Buellen     Bausteine     BiMATIC HMI-Station(1)     Bilder     WinCC flexible RT     WinCC flexible RT     Werbindungen     Sylen     Werbindungen     Sylen     Meldungen     Archive     Skinte	[	xterne Quelle einfüge Suchen in:	n Demo		? ×
Protokolle Text- und Grafiklisten Benutzerverwaltung R Geräteeinstellungen		Dateiname: db101.av	wl *.awl/*.gr7/*.scl/*.inp/*.zg,*.	.sdg;*.sd; 💌	Öffnen Abbrechen

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 Störnmelde-DB wird nur gebraucht, wenn es Sinn macht, in einem schon fortgeschrittenen Projekt "mittendrin" auf ProProFlexbasierte Störmeldungen umzusteigen.

#### **Problem:**

Es gibt in dieser Richtung nahezu beliebig viele Möglichkeiten:

Viele Triggervariable, die irgendwo liegen, ständiger Wechsel Triggervariable - "Quittiervariable Schreiben", zwischen den Blöcken die "Quittiervariable 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 Triggervariablenberiche 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 "Quittiervariable Schreiben") belegt einen DB. Für diese Quittiervariable 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:

G00_Gr	<sub>₹</sub> Variablen						
Rechteck					В	ITMELD	Uſ
Text	Nu 🔺 Meldek	Trigger-¥ariable	Bitn Hilfe	t Meldegruppe	Quittiervariable Lese	n Bit Qu Quittiervarial	ol E
res 0001	2001 🛨 Fehler	PT_Störmeldungen 1	0 🕂	▼ <keine melde="" th="" ▼<=""><th>PT_Quittierung-OP 1</th><th>🕶 0 🛛 🛨 PT_Störmeldu</th><th>. 🔻 1</th></keine>	PT_Quittierung-OP 1	🕶 0 🛛 🛨 PT_Störmeldu	. 🔻 1
res 0002	2002 Fehler	PT_Störmeldungen 1	1	<keine meldegru<="" th=""><th>PT_Quittierung-OP 1</th><th>1 PT_Störmeldung</th><th>g 1</th></keine>	PT_Quittierung-OP 1	1 PT_Störmeldung	g 1
E rec 0002	2002 Eablar	NT Stärmoldungen 1	2	Kaina Maldaanu	DT Quillionupa OD 1	2 DT Stärmoldum	- 1

G00_Gr 🔀 Bitmeldu	ingen 🔫	■Variablen									
											VARIA
Name	Anzei	Verbindung		Datentyp	Symbol		Adresse	A	rray-Elemente	Erfassungszyk	il Kommentar
PT_Störmeldungen 1		AG1	•	Word 🔹	<ul> <li><undefiniert></undefiniert></li> </ul>	-	DB 92 DBW 0	▼ 12	28	1 s	•
PT_Quittierung-OP 1		AG1		Word	<undefiniert></undefiniert>		DB 93 DBW 0	64	+ 64 Wort	e = 1024	Bit
Db_OP_IO_mpx.OP		AG1		Byte	<undefiniert></undefiniert>		DB 96 DBB 21	1		15	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.

TIC	CHMI-Station(1)	Ausgabeordner D:\tmp\P4	
,	Тур	Exportdatei	Opti
	Verbindungen	D:\tmp\P4\Connections.csv	
1	Variablen	D:\tmp\P4\Tags.csv	
1	Textlisten	D:\tmp\P4\Textlists.csv	
	Analog- und Bitmeldu	D:\tmp\P4\Alarms.csv	
		Datei	

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...

3 ProProFlex									
ProPro fü		Flex	Dapi	winCC Rex &	x-kompati	e .	(c Info	ĴК	eisch Automation
S7-AWL-Quelle	Date:>.AWL						100	Les	e AWLQuelle
Heldenunmer (Anfan	g) [1	Anzahl Meldungen ( mehrfaches von 16)	16	Heldeklasse	Alara	5		Sprache	de-DE
Triggervariabl	• 1	7T_Störmeldungen 1	Bitnummer (Start)	0	D8-Nr	92	_	_	_
C Quittervariable	le lesen ble SO-REIBT des OP I	PT_Quittlerung-OP_1	Bitmanener (Start)	0	D8-Nr	90			
Quittervariable per Definition (aus dieser Va	le schreiben gleiches Array wie Trig risblen LIEST das OP I	gervariable }	Bitzummer (Start)	16					
WinFlex- Import-Datei:	cAlams>.csv							Import	Datai schreiben
0					_				Melde-Reset
									DADE
Zu P	rojektbeginn:	Erzeuge neue A	WLQuelle	Erme	age AWL-Q	sadle ace Wi	NCCPlex	D	

WCC lesen	_OX
ProPro für WinCCFlex WinCC-Export-Datei lesen und Erzeugen einer AWL-Quelle	« Heisch
WinFlex-Export-Datei: (Doppelklick auf Dateiname öffnet Browser) D:\tmp\P4\Alarms.csv	Export-Datei lesen
Meldenummer (Anfang) 2001 Anzahl Meldungen 1024 Meldeklasse Alarms (mehrfaches von 16)	Sprache de-DE
Triggervariable PT_Störmeldungen 1 Bitnummer (Start) 0 DB-Nr 92	-
Quittiervariable lesen     PT_Quittiervarg-CP 1     Bitnummer (Start)     0     D8-Nr     93     (in diese Variable SCHREIBT das OP !!)	
Quttiervariable schreiben (Array = Triggervariable) Bitnummer (Start) I024 (Ack_PLC)	
S7-AWL-Quelle D:\tmp\P4\db92.AWL D:\tmp\P4\db92.AWL	Erzeuge AWLQuelle
Analyse 2 unterschiedliche Sprachen de-DE: 1024 Meldetexte Er-E0: 11 Meldetexte	Melde-Reset
Investigation         Ergebnis:         Triggervariablen         "PT_Störmeldungen 1" bei Meldenummern [20013024] und Bits [01023]         1 Quittiervar.Lesen         "PT_Quittierung-OP 1" bei Meldenummern [20013024] und Bits [01023]         1 Quittiervar.Schreiben:         "PT_Störmeldungen 1" bei Meldenummern [20013024] und Bits [10242047]         Fazit:         Block 1: Triggervariable beinhaltet "Quittierung Schreiben", PT_Quittierung-OP 1         Trigger: "PT_Störmeldungen 1" Lange Bereich 1024 Gesamt 2048 Quittier.Lesen "PT_Quittierung-OP 1" Bereich 1024         Plausibilitätscheck fertig	ZURÜCK

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 ..



Quelle db101_neu generieren	×
Hinweis: Automatisches Generieren von Einzeld Menü 'Extras' > 'Einstellungen' unter d	quellen je Baustein: er Lasche 'Quellen'
Pfad: ProProDemo\SIMATIC 300-Station\CPU3	315-2 DP(1)\
Nichtgewählte Bausteine:	Gewählte Bausteine:
DB102 dbAlarms_QT OB1 Cycle Execution Alle >	DB101 dbAlarms
Name/Familie:	
Referenzierte Bausteine einbeziehen     Nach Programmstruktur sortieren	Absolut     Sumbolisch
Quelle enthält Prüfsumme der Bausteine	
ОК	Abbrechen Hilfe

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

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

ProProFlex V1.1

S-s	SIMATIC Manager - ProProDemo									
Datei Bearbeiten Einfügen Zielsystem Ansicht Extras Fenster Hilfe										
	😂   🎛 🛲   X 🖻 🛍 🕍		□ □-□- IIII €	< Kein Filter >	🖸 🏹   🚼 🎯					
	🎒 ProProDemo C:\Program File	es\Siemens\S	tep7\s7proj\ProPro	~1						
	ProProDemo	Objektname	Symbolischer Name	Тур	Größe Autor	Änderungsdatum				
	SIMATIC 300-Station	🖹 db101		AWL-Quelle	13694	10.09.2010 15:48:44				
	□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □	🖹 db101_neu	I	AWL-Quelle	21535	10.09.2010 15:57:42				
	S7-Programm(1)									
	Bausteine	0	ielle evnortieren			2 X				
	SIMATIC HMI-Station(1)									
	🖻 🛄 WinCC flexible RT		Suchen in: 🛛 🖾 ProPr	oDemo	▼ 🖶 🔁 🐨 🎟 •					
	🗄 🗠 🔁 Bilder	I	db101.awl							
	S Kommunikation									
	🕀 🏣 Meldungen									
	🕀 🔁 Rezepturen									
	🗄 🔚 Archive									
	Text- und Grafiklisten	<u>ا</u> (	ateiname: db101_n	eu.AWL	Speich	hern				
	🗄 塔 Benutzerverwaltung R	,		× 1× 7× 1×, × ×	L X L = Abbros	shan				
	🗄 🔚 Geräteeinstellungen		Juellen	awi;gr/; .sci;".inp;".zg;".:	sag; .sa; 🔪 Abbied					
		_								

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

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

ProProFlex	×
ProPro für WinCCFlex       Display WinCC flex. Kompatibilität       (C)         V 1.1 vom 20.09.2010       (Doppekkick auf Dateiname öffnet Browser)       WinCC Flex. kompatibel       Info	
S7-AWL-Quelle     D. (init) (Frendbelind(datro) away       Meldenummer (Anfang)     1     Anzahl Meldungen (mehrfaches von 16)     160     Meldeklasse     Alarms     Sprache     de-DE	J
Triggervariable     dbAlarms     Bitnummer (Start)     0     DB-Nr     92       Outtiervariable lesen     dbAlarms_QT     Bitnummer (Start)     0     DB-Nr     93	
Quittiervariable schreihen     Bitnummer (Start)     160       per Definition gleiches Array wie Triggervariable (aus dieser Variablen LIEST das OP !!)     Bitnummer (Start)     160	
WinFlex- Import-Datei:         D:\tmp\ProProDemo\alarms.csv         Import-Datei schreiben	]
AWL-Quelle lesen AWL-Quelle lesen AWL-Quelle gelesen : 160 Meldungen Melde-Reset	
ENDE	
Zu Projektbeginn:         Erzeuge neue AWLQuelle         Erzeuge AWL-Quelle aus WinCCFlex	

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 Quittiervariablen 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 ->

					1.00	1	
WinCC flexible Advanced	l - Pro	ProDemo	- SIMATIC HMI-Stat	ion(1)			
Projekt <u>B</u> earbeiten <u>A</u> nsid	ht E	infügen	Forma <u>t</u> Bil <u>d</u> bausteine	E <u>x</u> tras <u>F</u> enster	Hilfe		
	~	VV	BA 12	A	B. 14 4 8	dhAlarma OT	- 022
	(3.	~ *			·M • [ M •2 •		
Deutsch (Deutschland)	· +						
Projekt (🌱 🗙		Bild_1	S <sup>™</sup> Verbindungen	Bitmeldungen	Meldeeinstellungen	🙀 Meldeklassen	Meldegruppen - Varia
Image: Projekt							17.0
SIMATIC HMI-Station							
E Bilder	N	ame	Anzei Verl	pindung Datent	yp Symbol	Adresse	Array-Elem
Bild hinzufüge		dbAlarms	CPU	315-2 DP(1) Word	<undefiniert></undefiniert>	DB 101 DE	3W 0 20
Vorlage		dbAlarms	QT CPU:	315-2 DP(1) 🔻 Word	<ul> <li><undefiniert></undefiniert></li> </ul>	▼ DB 102 DE	3W 0 🔽 10
Bild_1					Concession of the second se		
Variablen			100000 MA				
		CSA-IW	port				X
Zvklen		Lauria	100 m c - 40				
🖃 🧺 Meldungen		ISIMATIC	HMI-Station(1)				
🔤 📈 Analogmeldur		Aktiv	Тур	Überschreiben	Importd	atei	Optio
			Verbindungen				
🖻 😽 Einstellungen			Useisblas	E			
		는 끝구	variablen	-			
- Meldekla:	-11	<u> </u>	Textlisten				
Meldegru			Analog- und Bitmeldu	V	D:\tmp\Pr	oProDemo\alarms.csv	
					Die Det	at any day die Apalan in	a d Dites al duran a la constituit de
					Die Dat	ei, aus der die Analog- u	ina bitmelaungen importiert we
🕀 💆 Text- und Grafiklis						Im	portieren Abbrechen
	di						

... die Meldungen sind importiert:

WinCC flexible Advance	d - ProProDemo - SIMATIC HM	I-Station(1)							
Projekt <u>B</u> earbeiten <u>A</u> nsie	:ht <u>Ei</u> nfügen Forma <u>t</u> Bil <u>d</u> ba	usteine E <u>x</u> tras	Eenster Hilfe						
🛛 🖂 Neu 🔹 📂 📕 Ю 🧸	Q-XX MB.IS		II.IM.I	👸 🍕 🖏 dbAlarms	: QT	• . @ ?	8		
Deutech (Deutechland)			7 11 30						
Degrace (Degrace and)		<u> </u>			_				
Projekt 🤗 🗙	Bild_1 Strindunge	n ABitmeldu	ngen 🎽 Meldeein	stellungen 🛛 🙀 Melde	eklassen	Meldegrupper	1 - <b>∞≣</b> Vā	ariablen	
Projekt									
SIMATIC HMI-Station									
E Filder	Text	Nummer	A Meldeklasse	Trigger-Variable	Bitn	. Quittiervariabl	. Bit Qu.	Quittiervariabl.	Bit Qu
Vorlage	Test für Störmeldung 1	1	Fehler	dbAlarms	0	dbAlarms QT	0	dbAlarms	160
Bild_1	stoertext 0002	2	Fehler	dbAlarms	1	dbAlarms_QT	1	dbAlarms	161
🖻 🦙 Kommunikation	stoertext 0003	3	Fehler	dbAlarms	2	dbAlarms QT	2	dbAlarms	162
	stoertext 0004	4	Fehler	dbAlarms	3	dbAlarms QT	3	dbAlarms	163
Zuklen	stoertext 0005	5	Fehler	dbAlarms	4	dbAlarms OT	4	dbAlarms	164
🖃 🚾 Meldungen	stoertext 0006	6	Fehler	dbAlarms	5	dbAlarms QT	5	dbAlarms	165
- 🔀 Analogmeldur	stoertext 0007	7	Fehler	dbAlarms	6	dbAlarms QT	6	dbAlarms	166
	stoertext 0008	8	Fehler	dbAlarms	7	dbAlarms QT	7	dbAlarms	167
Meldeeins	Test für Störmeldung 9	9	Fehler	dbAlarms	8	dbAlarms_QT	8	dbAlarms	168
🙀 Meldekla:	stoertext 0010	10	Fehler	dbAlarms	9	dbAlarms QT	9	dbAlarms	169
Meldegruj	stoertext 0011	11	Fehler	dbAlarms	10	dbAlarms QT	10	dbAlarms	170
⊞ 🕰 Rezepturen	stoertext 0012	12	Fehler	dbAlarms	11	dbAlarms_QT	11	dbAlarms	171
E Skripte	stoertext 0013	13	Fehler	dbAlarms	12	dbAlarms_QT	12	dbAlarms	172
🗄 🨾 Protokolle	stoertext 0014	14	Fehler	dbAlarms	13	dbAlarms QT	13	dbAlarms	173
🗄 🚈 Text- und Grafiklis	stoertext 0015	15	Fehler	dbAlarms	14	dbAlarms QT	14	dbAlarms	174
Enutzerverwaltur	stoertext 0016	16	Fehler	dbAlarms	15	dbAlarms_QT	15	dbAlarms	175
E Sprachunterstützung	stoertext 0017	17	Fehler	dbAlarms	16	dbAlarms_QT	16	dbAlarms	176
Projektsprachen	stoertext 0018	18	Fehler	dbAlarms	17	dbAlarms QT	17	dbAlarms	177
Grafik sammlung									