



Applikationsbeispiel

Hantierungsbausteine

für XVision-Textdisplays

XV-101-K42

XV-101-K84

12/02 XSOFT-APPEXP-XV_TEXTDISPLAY V1.0

© Moeller GmbH, Bonn

Autor: O. Weiß

Alle Marken- und Produktnamen sind Warenzeichen oder eingetragene Warenzeichen der jeweiligen Titelhälter.

Alle Rechte, auch die der Übersetzung, vorbehalten.

Kein Teil dieses Anwendungshinweises darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder einem anderen Verfahren) ohne schriftliche Zustimmung der Firma Moeller GmbH, Bonn, reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Stand: April 2001

Änderungen vorbehalten.



Inhalt

1 Allgemeines

- Funktion	3
- Einsatzbereich.....	3
- Hardwarevoraussetzungen	3
- Softwarevoraussetzungen.....	3

2 Aufbau

- Verzeichnisstruktur	4
- Funktionsbausteine	4

3 Technische Informationen

- Funktionsweise der Bausteine	5
- Seitenumschaltung.....	5
- Wertanzeige / Werteingabe	5
- Alarmverarbeitung	5

4 Inbetriebnahme

- Installation der Bibliotheken	6
- Prototypen, Parameter und Parametrierung	7
- Seitenumschaltung.....	7
- Wertanzeige / Werteingabe	8
- Alarmverarbeitung	10
- Einbindung der Funktionsbausteine	12

5 Betrieb der Funktionsbausteine

- Seitenumschaltung	13
- Wertanzeige / Werteingabe	13
- Alarmverarbeitung.....	14
- Allgemeines zur Displaysteuerung.....	15

6 Anhang

- Programmbeispiele / Visualisierung.....	16
- Beispielapplikation	17



XVision Text-Displays dienen als Schnittstelle zwischen der XC-Steuerung und dem Bedienerpersonal. Die Display-Funktionen werden hier über die SPS-Applikation gesteuert. Die in diesem Applikationsbeispiel enthaltenen Funktionsbausteine unterstützen den Projektteur bei der Konfiguration häufig genutzter Displayfunktionen.

Funktion

Das Applikationsbeispiel stellt eine Seitenumschaltfunktion mit Wertanzeige- und eingabe für Integer-Variablen, sowie eine Alarmverarbeitung zur Verfügung.

Einsatzbereich

XVision-Applikationen für Text-Displays in denen die beschriebenen Funktionen genutzt werden sollen.

Hardwarevoraussetzungen

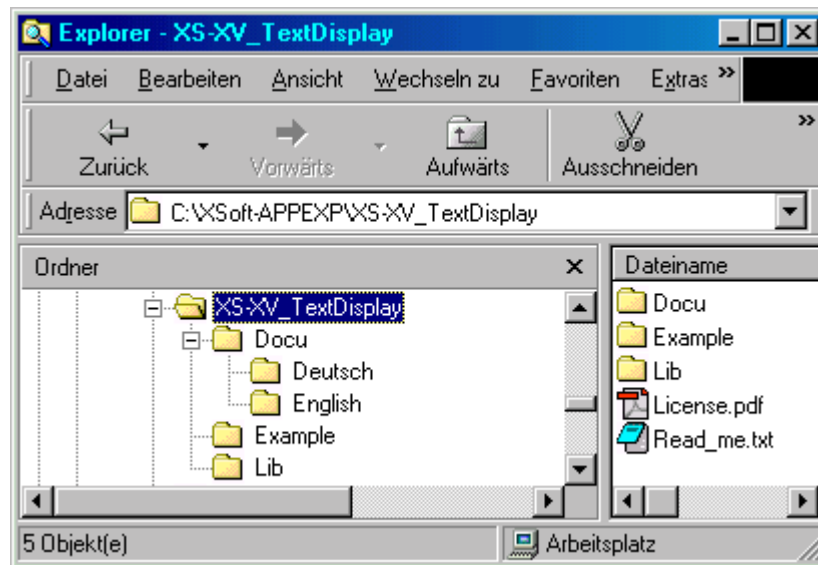
- XC-CPU101/201 mit XV-101-K42 oder XV-101-K84

Softwarevoraussetzungen

- XSoft ab V2.2..5 mit den Targets für XV-Geräte

Verzeichnisstruktur

Nach dem Entpacken der EXE-Datei erhalten Sie folgende Verzeichnisstruktur:



- | | |
|---------------------------------|--|
| ..\XS-XV_TextDisplay\DOCU\.. | -> Dokumentationen D/GB im PDF-Format |
| ..\XS-XV_TextDisplay\EXAMPLE\.. | -> SPS-Beispielprogramme mit Visualisierung und Textbeispiel |
| ..\XS-XV_TextDisplay\LIB\.. | -> Bibliotheken mit den Funktionsbausteinen, globale Variablen |

Funktionsbausteine

Das Anwendungsbeispiel stellt die Funktionsbausteine in zwei Standardbibliotheken zur Verfügung.

XSoft-APPEXP-XV_TextDisplay.Lib:

- | | |
|----------------------------|---|
| <i>XV_SelectPage</i> | -> einfache Seitenumschaltung |
| <i>XV_DisplayINTValues</i> | -> Anzeige / Eingabe von Integer-Werten |

XSoft-APPEXP-XV_TextDisplay.Lib:

- | | |
|----------------------------|----------------------|
| <i>XV_AlarmTextDisplay</i> | -> Alarmverarbeitung |
|----------------------------|----------------------|

Nach der Installation der jeweiligen Bibliothek sind die genannten Funktionsbausteine in der XSoft (Projekt) verfügbar.



Funktionsweise der Bausteine

Seitenumschaltung

In der XSoft wird mit Hilfe des XVision Tools die Textdatenbank für die Displays erstellt. Die in dieser Textdatenbank enthaltenen Texte werden (beginnend mit Text 1; ein Text in einer Zeile) linear in virtuelle ‚Seiten‘ unterteilt. Ist ein XV-101-K42 angekoppelt, so enthält eine Seite 4 Zeilen, beim XV-101-K84 werden 8 Zeilen in eine Seite gepackt und nach Bedarf angezeigt.

Über Eingänge am Funktionsbaustein ‚XV_SelectPage‘ kann gezielt auf eine Seite umgeschaltet werden. Weiterhin kann um eine Seite vor- oder zurückgeschaltet und auf die zuletzt angezeigte Seite geschaltet werden.

Wertanzeige / Werteingabe

Der Funktionsbaustein ‚XV_DisplayINTValues‘ arbeitet zusammen mit der Seitenumschaltung. Auf jeder Seite können bis zu 8 Integer-Variablen aus einem Array mit bis zu 255 möglichen Variablen dargestellt und auch editiert werden. Der Refresh der dargestellten Werte kann hierbei automatisch in definierten Zeitabständen oder auch manuell geschehen.

Alarmverarbeitung

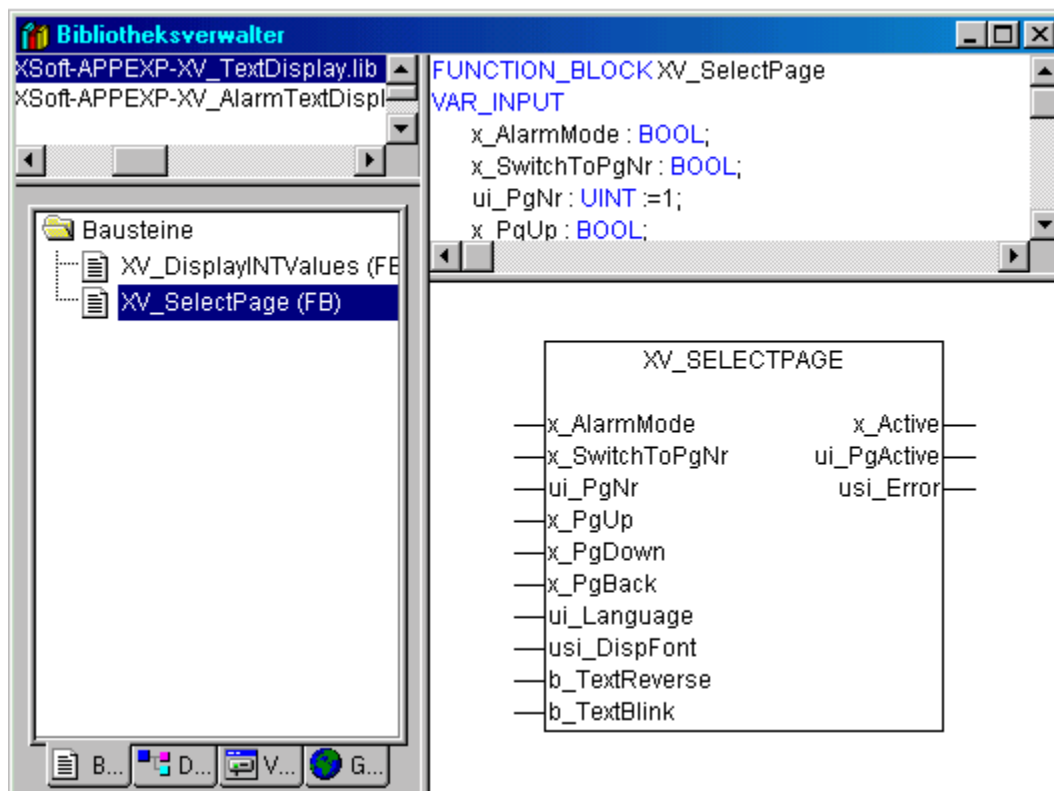
Der Funktionsbaustein ‚XV_AlarmTextDisplay‘ realisiert eine komfortable Alarmverwaltung mit Historienfunktion sowie die Alarm-Anzeige auf dem Display. Die Anzeigefunktion wurde nach dem Prinzip der Letztwertmeldung realisiert. Die boolschen Alarme werden über ein dynamisches Array verwaltet.

Die Anzeige der Alarme kann unabhängig von der Alarmverwaltung bei Bedarf zu- oder abgeschaltet werden.

Die Alarmtexte werden der Textdatenbank entnommen.

Installation der Bibliotheken

Die im Appliationsbeispiel enthaltenen Bibliotheken werden direkt in die XSoft eingebunden. Starten Sie die XSoft und fügen Sie die Bibliothek ‚XSoft-APP-XV_TextDisplay.lib‘ bzw. ‚XSoft-APP-XV_AlarmTextDisplay‘ in der Bibliotheksverwaltung (Menü ‚Fenster‘->‚Bibliotheksverwaltung‘) ein:



Das obige Bild zeigt die Bibliotheksverwaltung nach dem Importieren der Bibliotheken.

Um den benötigten Speicherplatz zu minimieren und die Parametrierung der benötigten Arrays zu dynamisieren ist ein Teil der Variablen global zu definieren. Die Definition der globalen Variablen und deren teilweise bereits beispielhafte Vorbelegung werden in zwei Export-Dateien im LIB-Unterverzeichnis zur Verfügung gestellt:

XSOFT-APPEXP-XV_TEXTDISPLAYGLOBALDATA.EXP -> Globale Variablen für den FB XV_DisplayINTValues aus der Bibliothek XSoft-APPEXP-XV_TextDisplay.Lib

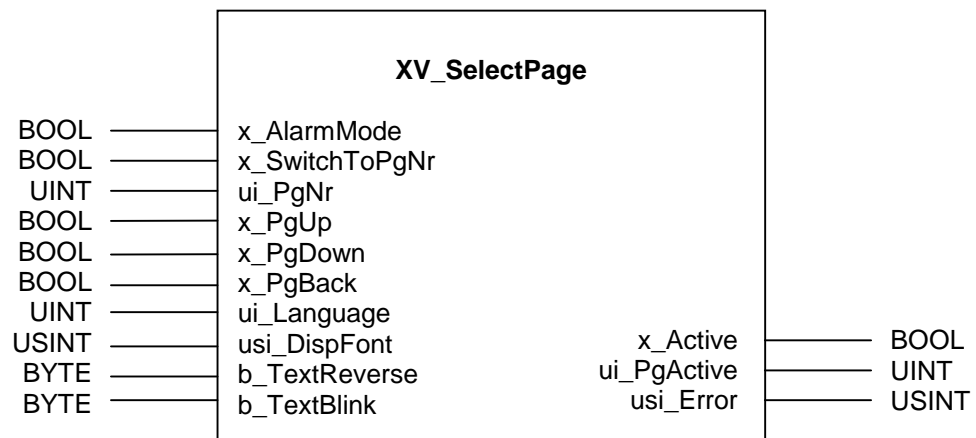
XSOFT-APPEXP-XV_ALARMTEXTDISPLAYGLOBALDATA.EXP -> Globale Variablen für den FB XV_AlarmTextDisplay aus der Bibliothek XSoft-APPEXP-XV_AlarmTextDisplay.Lib

Die benötigte Export-Datei ist über das Menü ‚Projekt‘ -> ‚Importieren‘ in das Projekt einzubinden. Die globalen Variablen werden dann automatisch in Objekte -benannt nach den Bibliotheken- unter den ‚Globalen Variablen‘ einsortiert und können direkt dort parametriert werden.

Die Funktionsbausteine greifen auf Bausteine anderer Standardbibliotheken wie die Standard.Lib und Visu.Lib zurück. Der Alarmbaustein benötigt zudem noch die RTCLib.Lib. Diese Bibliotheken werden beim Import der hier beschriebenen Bibliotheken ‚XSoft-APP-XV_TextDisplay.lib‘ bzw. ‚XSoft-APP-XV_AlarmTextDisplay‘ automatisch mit eingebunden.

Prototypen, Parameter und Parametrierung

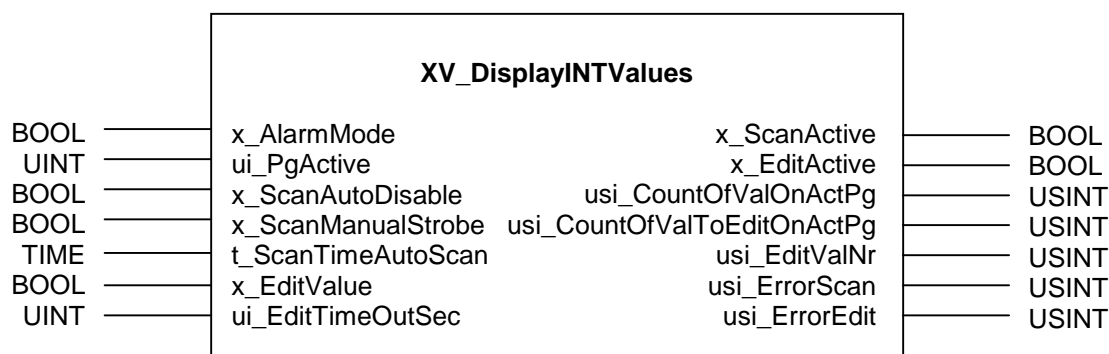
Seitenumschaltung



VAR_INPUT	
<i>x_AlarmMode</i>	TRUE: Zeile 0, 1 und 2 der Seite werden bei Umschaltung nicht mehr angezeigt -> bei Nutzung von ‚XV_AlarmTextDisplay‘, wenn Alarmanzeige aktiviert ist
<i>x_SwitchToPgNr</i>	Strobe: Umschalten auf Seite ‚ui_PgNr‘
<i>ui_PgNr</i>	Seitennummer auf die mit Strobe auf ‚x_SwitchToPgNr‘ umgeschaltet wird
<i>x_PgUp</i>	Strobe: Umschalten auf Seite + 1
<i>x_PgDown</i>	Strobe: Umschalten auf Seite - 1
<i>x_PgBack</i>	Strobe: Umschalten auf zuletzt angezeigte Seite
<i>ui_Language</i>	Sprache die beim nächsten Umschalten verwendet wird
<i>usi_DispFont</i>	Fontsatz der beim nächsten Umschaltvorgang benutzt wird
<i>b_TextReverse</i>	Zeilen, welche beim nächsten Umschaltvorgang invertiert dargestellt werden, binär codiert -> Bit 0..7 -> Text 0..7: 0 -> normal, 1 -> invertiert
<i>b_TextBlink</i>	Zeilen, welche beim nächsten Umschaltvorgang blinkend dargestellt werden, binär codiert -> Bit 0..7 -> Text 0..7: 0 -> normal, 1 -> blinkend (nur beim XV-101-K84)
VAR_OUTPUT	
<i>x_Active</i>	Umschaltvorgang aktiv
<i>ui_PgActive</i>	Nummer der aktuell angezeigten Seite
<i>usi_Error</i>	Fehlermeldung nach Umschaltung -> 0: Umschaltung ok, 1: Parameterfehler, 2: Bausteinanzahl zur Displayansteuerung überschritten, 3: Textdatenbank nicht vorhanden -> ‚ui_PgActive‘ -> 0 (leere Seite), 4: Seite (Texte) nicht vorhanden -> ‚ui_PgActive‘ -> 0 (leere Seite), 9: Anzeige nicht im Display-Mode (Input-Mode)

Im Texteditor des XVision Tools werden die Texte von 1 bis x linear durchnummeriert. Der Editor unterstützt die Seitenzuordnung durch die grafische Absetzung der Texte in 4er-Blöcke. Durch Setzen der Umgebungsvariablen SHOWXVPAGES (set SHOWXVPAGES=1) werden im Texteditor die Seitenzahlen in der Statuszeile eingeblendet, auf welcher sich der Cursor gerade befindet.

Wertanzeige / Werteingabe



VAR_INPUT	
x_AlarmMode	TRUE: Werte in den Zeilen 0, 1 und 2 werden nicht mehr angezeigt und sind auch nicht editierbar -> bei Nutzung von ‚XV_AlarmTextDisplay‘, wenn Alarmanzeige aktiviert ist
ui_PgActive	Nummer der aktuell angezeigten Seite (Output von XV_SelectPage)
x_ScanAutoDisable	FALSE: Werte werden in der Zeit ‚t_ScanTimeAutoScan‘ gescannt und angezeigt TRUE: Werte werden nicht automatisch aktualisiert -> Refresh über ‚x_ScanManualStrobe‘ -> Muß TRUE geschaltet werden, während Seitenumschaltung aktiv
x_ScanManualStrobe	Manueller Refresh der Anzeigewerte (‚x_ScanAutoDisable‘ = TRUE)
t_ScanTimeAutoScan	Scanzeit für Refresh der Anzeigewerte
x_EditValue	Umschalten in den ‚Edit-Modus‘ (Sollwerteingabe) / Sollwert selektieren
ui_EditTimeOutSec	TimeOut für die Sollwerteingabe in Sekunden
VAR_OUTPUT	
x_ScanActive	Werte-Refresh aktiv
x_EditActive	Sollwerteingabe aktiv
usi_CountOfValOnActPg	Anzahl der angezeigten Wert auf der aktuellen Seite
usi_CountOfValToEditOnActPg	Anzahl der editierbaren Werte auf der aktuellen Seite
usi_EditValNr	Index des gerade (des zuletzt) editierten Wertes
usi_ErrorScan	Fehler: Anzeigen der Werte -> 0: Kein Fehler, 1: Parameterfehler, 2: Bausteinanzahl zur Displayansteuerung überschritten
usi_ErrorEdit	Fehler: Sollwerteingabe -> 0: Kein Fehler, 1: Parameterfehler, 2: Bausteinanzahl zur Displayansteuerung überschritten, 5: Sollwerteingabe abgebrochen, 6: TimeOut bei Sollwerteingabe, 7: Kein Sollwert auf dieser Seite



VAR_GLOBAL	In ,XV_TextDisplay' unter ,Globale Variablen'
usi_DisplayINTValues_CountOfVal (CONSTANT)	Dimensionierung (Größe des Array) aller dargestellten Werte Defaultwert: 50
i_ar_DisplayINTValues	Werte (Soll- und Istwerte), Parametrierung der einzelnen Werte in den nachfolgenden Arrays
ui_ar_DisplayINTValues_Page	Seiten auf welchen die über den Index zugeordneten Werte aus ,i_ar_DisplayINTValues' dargestellt werden sollen, 0 -> ab hier keine Anzeige mehr Default: Werte 1..20 zugeordnet
usi_ar_DisplayINTValues_Line	Zeilen in welchen die über den Index zugeordneten Werte aus ,i_ar_DisplayINTValues' dargestellt werden sollen Default: Werte 1..20 zugeordnet
usi_ar_DisplayINTValues_Column	Spalten in welchen die über den Index zugeordneten Werte aus ,i_ar_DisplayINTValues' dargestellt werden sollen Default: Werte 1..20 zugeordnet
x_ar_DisplayINTValues_EditEnable	TRUE: der über den Index zugeordnete Wert ist editierbar FALSE: der über den Index zugeordnete Wert ist nicht editierbar Default: Werte 1..20 zugeordnet, alle editierbar

Die Dimensionierung des Arrays (,usi_DisplayINTValues_CountOfVal') für die Soll- und Istwerte muß über die Vorbelegung in der Deklaration geschehen.

Die übrigen Parameter für die Soll- und Istwerte:

,ui_ar_DisplayINTValues_Page'

,ui_ar_DisplayINTValues_Line'

,ui_ar_DisplayINTValues_Column' und

,ui_ar_DisplayINTValues_EditEnable' sollten auch in der Vorbelegung der Deklaration übergeben werden.

Die Soll- und Istwerte selber (,i_ar_DisplayINTValues') sind natürlich jederzeit entnehmbar und beschreibbar.

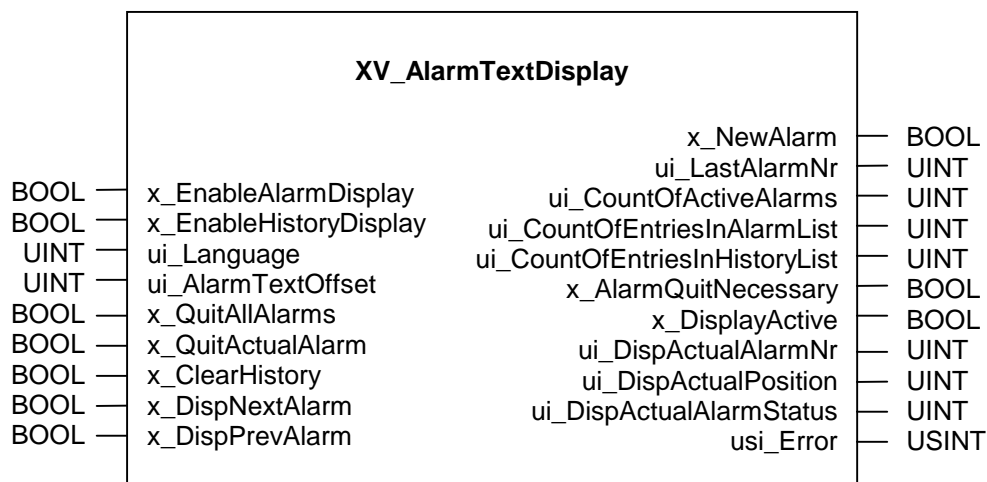
Die Zuordnung der einzelnen Parameter zu den jeweiligen Soll- und Istwerten erfolgt über den Index. So wird z.B. der Wert ,i_ar_DisplayINTValue'[1] auf der Seite ,ui_ar_DisplayINTValues_Page'[1] in der Zeile ,usi_ar_DisplayINTValues_Line'[1] und der Spalte ,usi_ar_DisplayINTValues'[1] dargestellt.

Editierbar ist dieser Wert, wenn ,x_ar_DisplayINTValues'[1] auf TRUE gesetzt ist, bei FALSE ist er nur ein Anzeigewert und wird im Editier-Modus übersprungen.

Bei der Parametrierung der Seiten in ,ui_ar_DisplayINTValues_Page' ist zu beachten, dass ab der ersten eingetragenen ,0' keine Werte mehr gescannt und angezeigt werden. So kann die Anzeige der Werte im laufenden Betrieb begrenzt werden.

Die Anzeige der Integer-Werte erfolgt generell 6-stellig.

Alarmverarbeitung



VAR_INPUT	
x_EnableAlarmDisplay	Freigabe Alarm-/History-Anzeige auf Display
x_EnableHistoryDisplay	Umschalten auf History-Anzeige (nur wenn 'x_EnableAlarmDisplay' = TRUE)
ui_Language	Sprache
ui_AlarmTextOffset	Offset für ersten Alarmtext
x_QuitAllAlarms	Quittiere alle anstehenden Alarme
x_QuitActualAlarm	Quittiere den aktuell auf dem Display angezeigten Alarm
x_ClearHistory	Lösche alle Historien-Einträge
x_DispatchNextAlarm	Zeige nächsten Alarm
x_DispatchPrevAlarm	Zeige vorherigen Alarm
VAR_OUTPUT	
x_NewAlarm	Neuer Alarm
ui_LastAlarmNr	Zuletzt gekommener Alarm
ui_CountOfEntriesInAlarmList	Anzahl der anstehenden Alarme
ui_CountOfEntriesInHistoryList	Anzahl der Historieneinträge
x_AlarmQuitNecessary	Quittierung bei aktuellem Alarm erforderlich
x_DisplayActive	Alarm-/Historien-Anzeige freigegeben
ui_DispatchActualAlarmNr	Nr. des aktuell angezeigten Alarms
ui_DispatchActualPosition	Position des aktuell angezeigten Alarms innerhalb der Liste
ui_DispatchActualAlarmStatus	Status des aktuell angezeigten Alarms: 0: kein Alarm, 1: gekommen, 2: gegangen, 3: quittiert, 4: nicht quittiert, 5: gegangen und nicht quittiert
usi_Error	Fehler bei der Anzeige: 0: OK, 1: Parameterfehler, 2: Bausteinanzahl zur Displayansteuerung überschritten, 3: Textdatenbank nicht vorhanden, 4: Text nicht vorhanden



VAR_GLOBAL	In ,XV_AlarmTextDisplay' unter ,Globale Variablen'
ui_AlarmTextDisplay_AlarmIndexMax (CONSTANT)	Dimensionierung (Größe des Arrays) für Alarme Defaultwert: 50
ui_AlarmTextDisplay_HistoryIndexMax (CONSTANT)	Dimensionierung (Größe des Arrays) des Historienspeichers Defaultwert: 50
x_ar_AlarmTextDisplay_Alarms	,INPUT' für FB: Alarme, Parametrierung der Alarme mit nachfolgendem Array
x_ar_AlarmTextDisplay_NoQuitNecessary	,INPUT' für FB: Einstellung der Quittierung für den über den Index zugeordneten Alarm -> FALSE: Alarm ist zu quittieren TRUE: Alarm bedarf keiner Quittierung
x_ar_AlarmTextDisplay _DispAlarmsQuitNecessary	OUTPUT für FB: Zeigt an welche Alarme noch zu quittieren sind

Bei der Dimensionierung des Alarmfeldes (,ui_AlarmTextDisplay_AlarmIndexMax') ist zu berücksichtigen, daß die Zykluszeit mit zunehmender Größe des Arrays ansteigt.

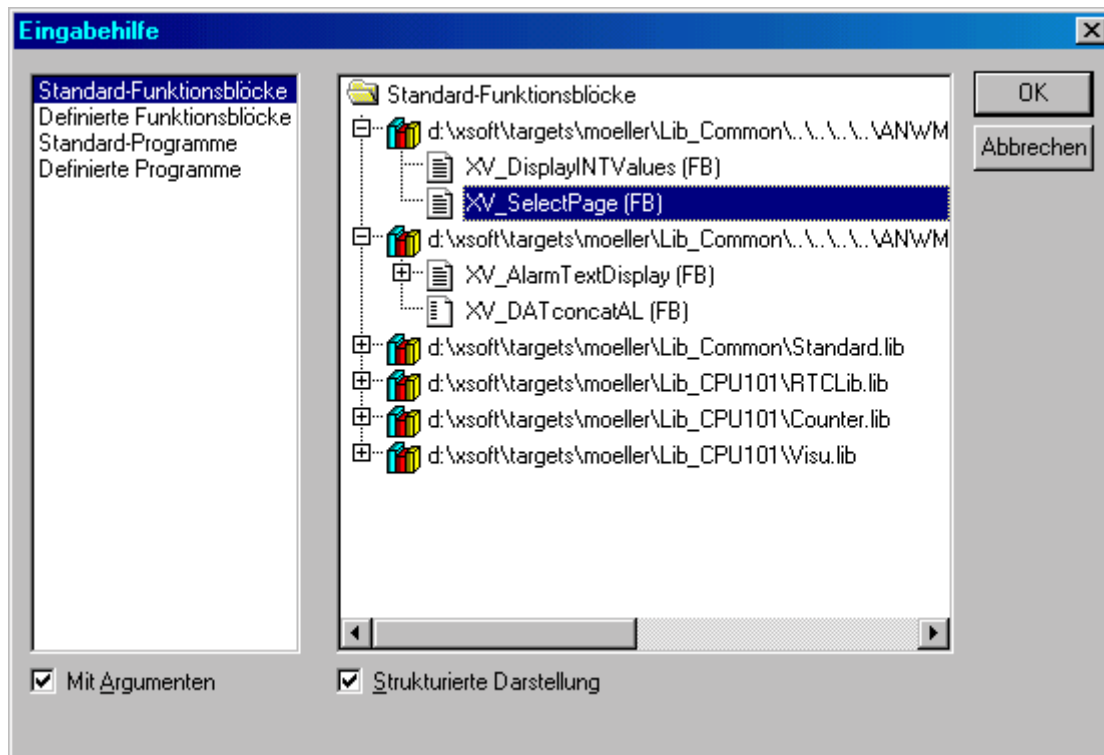
Bei gleichzeitiger Verwendung von Seitenumschaltung und Alarmverarbeitung, sollten die Alarmtexte im Texteditor der XSoft in einer separaten Spalte (Sprache) definiert werden um Überschneidungen mit den Texten aus der Seitenumschaltung zu vermeiden -> Parameter ,ui_Language' am Baustein entsprechend einstellen.

Beachte: Für die korrekte Anzeige des Alarmstatus und für den korrekten Eintrag in den Historienspeicher, ist es erforderlich, daß die Uhr gestellt wurde !

Einbindung der Funktionsbausteine

Die Einbindung ist jeweils durch den Aufruf des entsprechenden Funktionsbausteines zu realisieren, wie z.B. „CAL XV_...“ bei der Programmierung in AWL.

Sie finden die Funktionsbausteine in den Bibliotheken unter den Standard-Funktionsbausteinen:



Der Aufruf des Funktionsbausteines sollte zyklisch durchlaufen werden um eine kontinuierliche und schnellstmögliche Auftragsabarbeitung zu gewährleisten.

Aus der Bibliothek 'XSoft-APPEXP-XV_AlarmTextDisplay' ist nur der Funktionsbaustein 'XV_AlarmTextDisplay' zu nutzen. Alle andern Objekte in dieser Bibliothek werden intern benötigt.



Seitenumschaltung

Die Seitenumschaltung wird über den Funktionsbaustein ‚XV_SelectPage‘ realisiert.

Wie schon in Kapitel 3 beschrieben werden mit Hilfe des FBs die in der Textdatenbank enthaltenen Texte (beginnend mit Text 1; ein Text in einer Zeile) linear in virtuelle ‚Seiten‘ unterteilt. Ist ein XV-101-K42 angekoppelt, so enthält eine Seite 4 Zeilen, beim XV-101-K84 werden 8 Zeilen in eine Seite gepackt und nach Bedarf angezeigt. So werden z.B. beim 8-zeiligen Display auf Seite 1 die Texte 1 bis 8 angezeigt; die Seite 2 stellt die Texte 9 bis 16 dar usw.. Die Texte einer Seite beginnen dabei immer in Spalte 0 und Zeile 0 des Displays.

Nach Start der Applikation, also beim ersten Durchlauf des FBs, wird automatisch auf die Seite geschaltet, die am Eingang ‚ui_PgNr‘ eingestellt ist; vorausgesetzt diese ist gültig. Ist der Wert nicht parametrierbar, bzw. mit ‚0‘ belegt, so wird auf Seite 1 geschaltet.

Mit einem Strobe auf ‚x_PgUp‘ wird zur folgenden Seite umgeschaltet, mit ‚x_PgDown‘ eine Seite zurück, mit ‚x_PgBack‘ auf die zuvor angezeigte Seite und mit ‚x_SwitchToPgNr‘ wird auf die Seite umgeschaltet, welche an ‚ui_PgNr‘ parametrierbar ist. Ist die Seitennummer ungültig oder ‚0‘ so wird die Seite gelöscht und der Fehlercode ‚4‘ generiert. Als aktive Seitennummer wird auf dem Ausgangsparameter ‚ui_PgActive‘ der Wert ‚0‘ ausgegeben.

Solange die Seiten auf die umgeschaltet wird gültig sind, wird die Seitennummer in ‚ui_PgActive‘ entsprechend aktualisiert. Der Ausgang ‚x_Active‘ signalisiert, daß z.Zt. eine Seitenumschaltung aktiv ist. Bei jeder Umschaltung kann Sprache (‚ui_Language‘), Font (‚usi_DisFont‘) und die Textdarstellung (‚b_TextReverse‘, ‚b_TextBlink‘) neu eingestellt werden. Bei großen Fontsätzen ist zu berücksichtigen, daß sich die Zeilenanzahl halbiert.

Die Textdarstellung ist binär parametrierbar, so daß jeder Text einzeln eingestellt werden kann -> Bit 0 für Text in Zeile 0, Bit 1 für Text in Zeile 1 usw..

Bei Aktivierung des Eingangs ‚x_AlarmMode‘ wird bei einer Umschaltung das Löschen und die Anzeige der ersten drei Displayzeilen unterdrückt. Dies sollte bei gleichzeitigem Gebrauch des Alarmbausteins ‚XV_AlarmTextDisplay‘ genutzt werden. Ansonsten wird der angezeigte Alarm bei einer Seitenumschaltung überschrieben.

Befindet sich das Display im Eingabe-Mode, ist der Seitenwechsel inaktiv (usi_Error = 9).

Wertanzeige / Werteingabe

Der Funktionsbaustein ‚XV_DisplayINTValues‘ arbeitet zusammen mit der Seitenumschaltung. Auf jeder Seite können hiermit bis zu 8 Integer-Variablen aus dem globalen Array ‚i_ar_DisplayINTValues‘ dargestellt und editiert werden. Dabei wird jeder Wert aus dem Array genau einmal dargestellt. Die Parametrierung der darzustellenden Soll- und Istwerte ist ausführlich in Kapitel 4 beschrieben.

Die Seitennummer ‚ui_PgActive‘ bestimmt, welche Variablen angezeigt werden. Die Aktualisierung der Anzeige geschieht bei jedem Seitenwechsel automatisch einmal innerhalb der Scanzeit ‚t_ScanTimeAutoScan‘. Vorausgesetzt, die automatische Abtastung ‚x_ScanAutoDisable‘ = FALSE ist nicht mit ‚x_ScanAutoDisable‘ = TRUE deaktiviert worden.

Die automatische Abtastung muß abgeschaltet werden, während die Seitenumschaltung aktiv ist. Das heißt, der Eingang kann mit dem Ausgang ‚x_Active‘ des FBs ‚XV_SelectPage‘ direkt beschaltet werden. Ist die automatische Abtastung permanent deaktiviert (‚x_ScanAutoDisable‘ = TRUE), so kann mit dem Eingang ‚x_ScanManualStrobe‘ eine manuelle Abtastung erfolgen.

Der Ausgang ‚x_ScanActive‘ zeigt an, ob gerade ein Scan-/Abtastvorgang mit Refresh der Anzeige erfolgt. ‚usi_CountOfValOnActPg‘ gibt die Anzahl der auf der aktuellen Seite dargestellten Werte an. Dem Ausgang ‚usi_CountOfValToEditOnActPg‘ kann entnommen werden, wieviele der dargestellten Werte editierbar sind.

Wird ein Fehler bei der Anzeige (usi_ErrorScan) gemeldet, so werden alle Werte bis zu dem Wert, wo der Fehler aufgetreten ist noch dargestellt.

Mit dem ersten Strobe am Eingang ‚x_EditValue‘ wird in den Variableneingabe-Modus gewechselt. Der Cursor steht auf dem ersten, editierbaren Wert. Durch weiteres Tippen auf ‚x_EditValue‘ wird jeweils zum nächsten editierbaren Wert weitergeschaltet. Die Reihenfolge wird dabei durch die Deklaration bestimmt. Erfolgt eine Werteingabe mit ‚ENTER‘, so wird der Editier-Modus verlassen. Mit ‚ESC‘ oder durch



erneutes Betätigen von ‚x_EditValue‘, wenn der Cursor bereits auf dem letzten editierbaren Wert steht, wird die Eingabe abgebrochen. Die Eingabe kann jedoch jederzeit wieder mit ‚x_EditValue‘ neu gestartet werden.

Der Ausgang ‚x_EditActive‘ zeigt den Editier-Modus an, und der Ausgang ‚usi_EditValNr‘ gibt Auskunft über den gerade editierten, oder nach Beendigung des Editiermodus, über den zuletzt editierten Sollwert.

Bis auf den gerade editierten Wert werden alle dargestellten Werte auch im Editier-Modus mit den definierten Bedingungen weiterhin aktualisiert. Auch beim Wechsel von einem Sollwert zum nächsten erfolgt ein Refresh der angezeigten Werte.

Bei Aktivierung des Eingangs ‚x_AlarmMode‘ wird bei jedem Refresh die Anzeige aller Variablen, welche in den ersten drei Displayzeilen definiert sind, unterdrückt. Dies sollte genutzt werden, wenn der Alarmbaustein ‚XV_AlarmTextDisplay‘ gleichzeitig aktiviert ist. Ansonsten werden die Variablen evtl. über die Alarminformationen geschrieben. Durch entsprechendes Plazieren der Variablen können auch Werte innerhalb des Alarmtextes dargestellt werden.

Alarmverarbeitung

Die Alarmverwaltung ist aktiv sobald der Funktionsbaustein ‚XV_AlarmTextDisplay‘ aufgerufen wird; unabhängig von ‚x_EnableAlarmDisplay‘.

Der Ausgang ‚x_NewAlarm‘ signalisiert, daß mindestens ein Alarm im globalen Alarmfeld ‚x_ar_AlarmTextDisplay_Alarms‘ aktiv ist. Der Ausgang ‚ui_LastAlarmNr‘ zeigt die Nummer des zuletzt gekommenen Alarms und auf dem Ausgang ‚ui_CountOfActiveAlarms‘ wird die Anzahl der z.Zt aktiven Alarmer ausgegeben. Der Ausgang ‚ui_CountOfEntriesInAlarmList‘ gibt die Anzahl der Einträge in der Alarmliste aus. Hier werden alle aktiven oder auch noch nicht quittierten Alarmer hinzugerechnet.

Der Ausgang ‚ui_CountOfEntriesInHistoryList‘ beinhaltet die Anzahl der Einträge im Historienspeicher. Der Historienspeicher ist als Ringbuffer ausgebildet. Er beinhaltet Informationen über gekommene, gegangene und quittierte Alarmer. Der Ausgang ‚x_AlarmQuitNecessary‘ wird nur bei eingeschalteter Alarmanzeige gesetzt und zeigt an, dass noch Alarmer zu quittieren sind. Das globale Feld ‚x_ar_AlarmTextDisplay_DispAlarmsQuitNecessary‘ zeigt an, welche Alarmer noch zu quittieren sind.

Mit ‚x_EnableAlarmDisplay‘ = TRUE wird die Alarmanzeige eingeschaltet und der Ausgang ‚x_DisplayActive‘ wird gesetzt. Ist zusätzlich ‚x_EnableHistoryDisplay‘ gesetzt, wechselt das Display in die Historien-Anzeige. Hier werden die Texte der Alarmer aus dem Historienspeicher mit Datum/Uhrzeit/Status angezeigt.

Die Anzeige erfolgt 2-zeilig ab Zeile 0/Spalte 0: 1. Zeile -> Datum/Uhrzeit/Status, 2. Zeile -> Alarmtext.

```
2002-12-01-15:34:22+
Dies ist Alarm 1
```

Der Status wird immer durch ein Zeichen direkt hinter der Uhrzeit dargestellt.

In der Historienanzeige bedeuten:

```
+    gekommen
-    quittiert
x    gegangen
```

Die Statuszeile ist auf 20 Zeichen begrenzt, damit sie in gleicher Weise auf jedem Display angezeigt werden kann.

Mit den Tastern ‚x_DispNextAlarm‘ und ‚x_DispPrevAlarm‘ kann hier wie auch in der Alarmanzeige zum nächsten oder vorherigen Alarm gewechselt werden. ‚x_ClearHistory‘ löscht den Historienspeicher.



Im Alarm-Anzeige-Modus (Anzeige als Letztwertmeldung) wird neben Alarmtext , Datum, Uhrzeit und Status, in der dritten Zeile dargestellt, welcher (zeitlich vom letzten her sortiert) Alarm von wieviel aktiven Alarmen gerade angezeigt wird.

```
2002-12-01-15:34:22+
Dies ist Alarm 1
Alarm 1/5
```

Der Status wird auch hier durch ein Zeichen direkt hinter der Uhrzeit dargestellt.

In der Alarmanzeige bedeuten:

- + gekommen
- quittiert
- * gegangen und nicht quittiert

Der Eingang ,x_QuitActualAlarm' quittiert den aktuell angezeigten Alarm (nur wenn AlarmDisplay aktiv). Mit ,x_QuitAllAlarm' können alle in der Alarmliste befindlichen Alarme quittiert werden (auch wenn AlarmDisplay nicht aktiv ist).

Der Ausgang ,ui_ActualAlarmNr' zeigt an, welcher Alarmtext gerade angezeigt wird. An ,ui_DisplActualPosition' wird ausgegeben, auf welcher Position sich der Zeiger innerhalb der Liste der aktiven Alarme befindet. Die Position wird zeitlich nach Letztwertmeldung sortiert bestimmt.

Am Ausgang ,ui_DisplActualAlarmStatus' ist zu entnehmen, welchen Status der gerade angezeigte Alarm hat:

- 0 kein Alarm
- 1 Alarm gekommen
- 2 Alarm gegangen
- 3 Alarm aktiv und quittiert
- 4 Alarm aktiv und nicht quittiert
- 5 Alarm gegangen und nicht quittiert.

Allgemeines zur Displaysteuerung

Die drei beschriebenen Funktionsbausteine greifen auf die Bibliothek Visu.Lib zurück, welche alle Steuerfunktionen für das Display beinhaltet.

Die Bausteine aus der Visu.Lib können natürlich weiterhin auch zusammen mit den hier beschriebenen Funktionsbausteinen genutzt werden um weitere Displayfunktionen auszuführen. Allerdings muß hierbei berücksichtigt werden, daß keine zeitlichen Überschneidungen entstehen und so evtl. Display-Ansteuerfunktionen nicht oder nur teilweise ausgeführt werden. Außerdem ist zu berücksichtigen, daß maximal 10 Display-Aufträge gleichzeitig aktiv im Stack bearbeitet werden können.

Jeder weitere gestartete Auftrag wird abgewiesen, wenn bereits 10 aktiv sind. Es ist immer der Fehlercode (Error Code = 2, wenn Stack voll) nach Auftragsabarbeitung zu berücksichtigen. Sinnvoll ist es die Aufträge möglichst nacheinander abzuarbeiten.

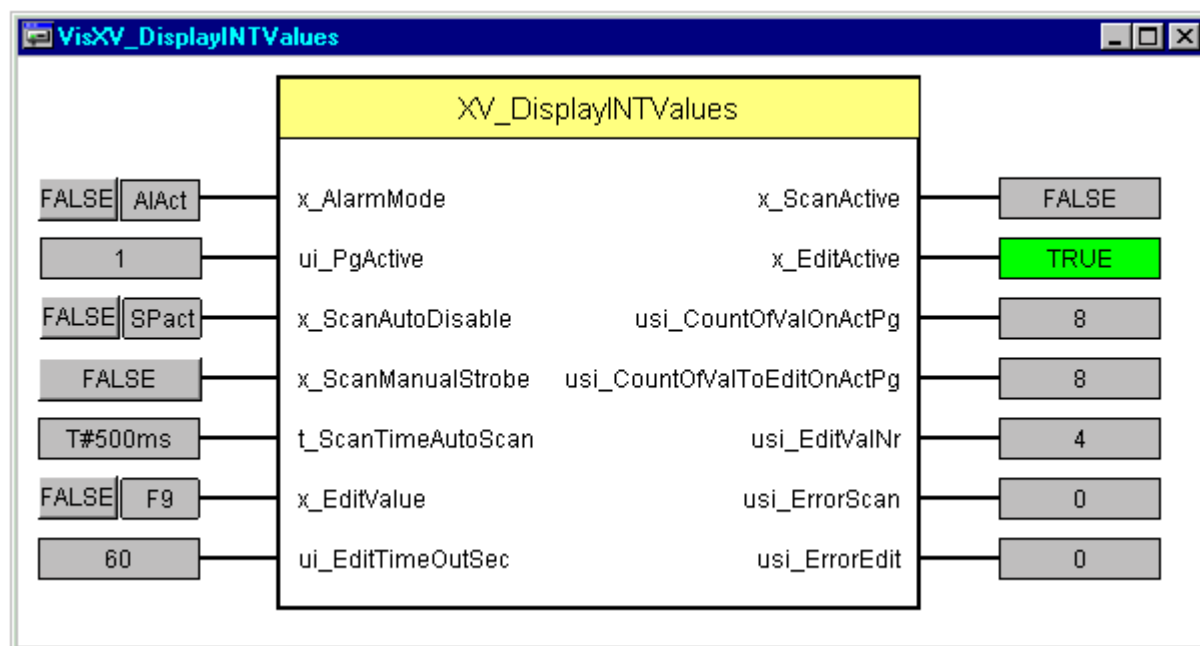
Programmbeispiele / Visualisierung

Im Unterverzeichnis `..\XS-XV_TextDisplay\EXAMPLE\.` finden Sie Programmbeispiele in Form von Export-Files.

Hier sind zwei Export-Files `VisXV_...` zu finden, welche neben dem Programmbeispiel auch eine gleichnamige Visualisierung für jeden FB beinhalten:

VisXV_TextDisplay.Exp

Beispielprogramme `VisXV_SelectPage`, `VisXV_DisplayINTValues` mit den gleichnamigen Visualisierungen



z.B.: Visualisierung `VisXV_DisplayINTValues`

Die Beispielprogramme sind so parametrierbar, daß die entsprechenden FBs von dort aus komplett bedient werden können. Zusätzlich sind einige Parameter noch innerhalb der Beispielprogramme miteinander verknüpft. Ebenso können einige Eingänge zusätzlich auch über Tastatur bedient werden (Seite vor, > / zurück, < sowie das Umschalten auf Sollwerteingabe mit ,F9' und zurück zur zuvor angezeigten Seite mit ,Shift').

Für die Parametrierung der globalen Daten des FBs `VisXV_DisplayINTValues` kann die ebenfalls enthaltene Visualisierung `VisXV_DisplayINTValuesGlobalData` im SPS-Status ,HALT' oder ,RUN' genutzt werden.

VisXV_AlarmTextDisplay.Exp

Beispielprogramm `VisXV_AlarmTextDisplay` mit gleichnamiger Visualisierung.

Hier werden zusätzlich die Tasten ,CursorUp', ,CursorDown' und ,ENTER' des Text-Displays genutzt um innerhalb der Alarmanzeige zu scrollen bzw. Alarmer zu quittieren.

Um die Alarmanzeige sinnvoll mit den Beispielprogrammen zur Seitenumschaltung und Wertanzeige `VisXV_SelectPage` und `VisXV_DisplayINTValues` zu verbinden, ist in diesen Programmen der zu Anfang auskommentierte Block zwischen den beiden Linien (*-----*) zu aktivieren. Hier werden die Eingänge `x_AlarmMode` von `VisXV_SelectPage` und `VisXV_DisplayINTValues` mit der Alarmanzeige verknüpft. Entsprechende Hinweise finden sie auch in den Programmköpfen.

Einige Defaulteinstellungen der Variablen sind auf ein 8-Zeiliges Display abgestimmt.

Beim 4-zeiligen Display sind entsprechende Anpassungen in der Global-Deklaration erforderlich.

Die Textdatei `VisXV_TextDisplay.Xtd` kann für die Visualisierungen auf das Display heruntergeladen werden.



Für die Inbetriebnahme des Beispiels gehen Sie wie folgt vor:

1. Öffnen Sie ein neues Projekt
2. Laden Sie in der Steuerungskonfiguration unter ‚Weitere Parameter‘ die Textdatenbank ‚VisXV_TextDisplay‘ auf die Steuerung
3. Fügen Sie die benötigte Bibliothek ‚XSoft-APPEXP-XV_TextDisplay‘ bzw. ‚XSoft-APPEXP XV_AlarmTextDisplay‘ wie beschrieben in der Bibliotheksverwaltung ein
4. Importieren Sie die benötigten Globaldefinitionen XS-OFT-APPEXP-XV_TEXTDISPLAYGLOBALDATA.EXP bzw. XS-OFT-APPEXP-XV_ALARMTEXTDISPLAYGLOBALDATA.EXP
5. Importieren Sie die gewünschten Beispielprogramme XS-VISXV_TEXTDISPLAY.EXP bzw. XS-VISXV_ALARMTEXTDISPLAY.EXP
6. Wenn Sie die Alarmverwaltung zusammen mit der Seitenumschaltung und der Wertanzeige betreiben wollen, aktivieren Sie wie beschrieben die markierten, auskommentierten Stellen in den Programmen ‚VisXV_SelectPage‘ und ‚VisXV_DisplayINTValues‘
7. Fügen Sie nun die Aufrufe der gewünschten Beispielprogramme in Ihr Hauptprogramm ‚PLC_PRG‘ ein
8. ‚Alles bereinigen‘ - ‚Alles übersetzen‘ – herunterladen und starten

Beispielapplikation

Im Unterverzeichnis ..\XS-XV_TextDisplay\EXAMPLE\.. finden Sie eine Beispielapplikation XS-EXPXV_TEXTDISPLAY.EXP. Die zugehörige Textdatei für die Displaytypen ‚AppXV_TextDisplayK42‘ bzw. ‚AppXV_TextDisplayK84‘, kann mit dem XVision Tool auf das Display heruntergeladen werden. Beim Import werden die zugehörigen Globaldefinitionen bereits mitgeladen.

Das Beispielprogramm ‚AppXV_TextDisplay‘ bedient die Anzeige von 6 Seiten, welche teilweise mit Variablen bestückt sind. Ebenso ist die Alarmverwaltung mit eingebunden. 8 Alarmtexte sind definiert. Die Alarmer können auch über die gleichnamige Visualisierung ‚AppXV_TextDisplay‘ ein- oder ausgeschaltet werden.

Die Bedienung der Applikation erfolgt über die Tastatur:

F1:	Gehe zur Startseite
F2:	Alarm-Display Ein/Aus
F3:	Produktion
F4:	Motordrehzahl
F5:	Motortemperatur
F6:	Wechsel Landessprache (deutsch/englisch)
F7:	Uhr stellen (auf den 11.12.2002, 10:00:00 Uhr)
F9:	Sollwerteingabe
>:	Nächste Seite
<:	Vorherige Seite
CursorUp:	Zeige vorigen Alarm
CursorDown:	Zeige nächsten Alarm
ENTER:	Quittiere aktuellen Alarm
CLEAR:	Quittiere alle Alarmer