



Application Example

Data Handling Function Blocks

for XVision Text Displays

XV-101-K42

XV-101-K84

12/02 XSOFT-APPEXP-XV_TEXTDISPLAY V1.0

© Moeller GmbH, Bonn

Author: O. Weiß

All brand and product names are trademarks or registered trademarks of the owner concerned.

All rights reserved, also for the translation.

No part of these application notes may be reproduced in any form (printed, photocopy, microfilm or any other process) or processed, duplicated or distributed by means of electronic systems without the written permission of Moeller GmbH, Bonn.

Date: April 2001

Subject to modifications.



Contents

1 General

- Function	3
- Application range	3
- Hardware requirements	3
- Software requirements	3

2 Structure

- Folder structure	4
- Function blocks	4

3 Technical information

- Operation of function blocks	5
- Page selection	5
- Value display / value entry	5
- Alarm processing	5

4 Commissioning

- Installing the libraries	6
- Prototypes, parameters and parameter assignment.....	7
- Page selection	7
- Value display / value entry	8
- Alarm processing	10
- Incorporating the function blocks	12

5 Running the function blocks

- Page selection.....	13
- Value display / value entry	13
- Alarm processing	14
- General information on the display PLC	15

6 Appendix

- Program examples / visualization	16
- Example application	17



XVision text displays are used as an interface between the XC PLC and the operator. The display functions for this are controlled via the PLC application. The function blocks contained in this application example support the design engineer in configuring frequently used display functions.

Function

The application example features a page selection function with value display and value entry for integer variables, and also an alarm processing function.

Application range

XVision applications for text displays in which the described functions are to be used.

Hardware requirements

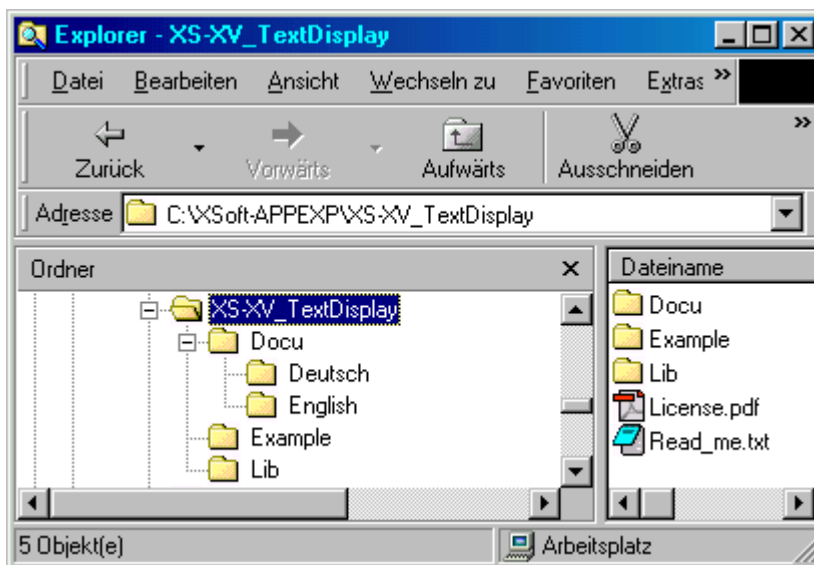
- XC-CPU101/201 with XV-101-K42 or XV-101-K84

Software requirements

- XSoft from V2.2.5 with the targets for XV devices

Folder structure

The following folder structure is produced after the EXE file is unpacked:



..\XS-XV_TextDisplay\DOCU\..	-> D/GB documentation in PDF format
..\XS-XV_TextDisplay\EXAMPLE\..	-> PLC example programs with visualization and text example
..\XS-XV_TextDisplay\LIB\..	-> Libraries with the function blocks, global variables

Function blocks

The application example provides two standard libraries of function blocks.

XSoft-APPEXP-XV_TextDisplay.Lib:

<i>XV_SelectPage</i>	-> Simple page selection
<i>XV_DisplayINTValues</i>	-> Display / entry of integer values

XSoft-APPEXP-XV_TextDisplay.Lib:

<i>XV_AlarmTextDisplay</i>	-> Alarm processing
----------------------------	---------------------

The function blocks mentioned are available in XSoft (Project) after the appropriate library has been installed.



Operation of function blocks

Page selection

The text database for the displays is created in XSoft using the XVision Tool. The texts contained in this text database are divided up consecutively (start with Text 1; one text in each line) into virtual 'pages'. If an XV-101-K42 is connected, a page consists of 4 lines. In the case of an XV-101-K84, a page contains 8 lines which can be displayed as required.

The inputs on the XV_SelectPage function block enable a specific page to be selected. It is also possible to scroll to the next or previous page or to the last page displayed.

Value display / value entry

The XV_DisplayINTValues function block works in conjunction with the page selection function. Up to 8 integer variables from an array containing up to 255 variables can be shown and edited on each page. The displayed values can also be refreshed manually or automatically in defined time intervals.

Alarm processing

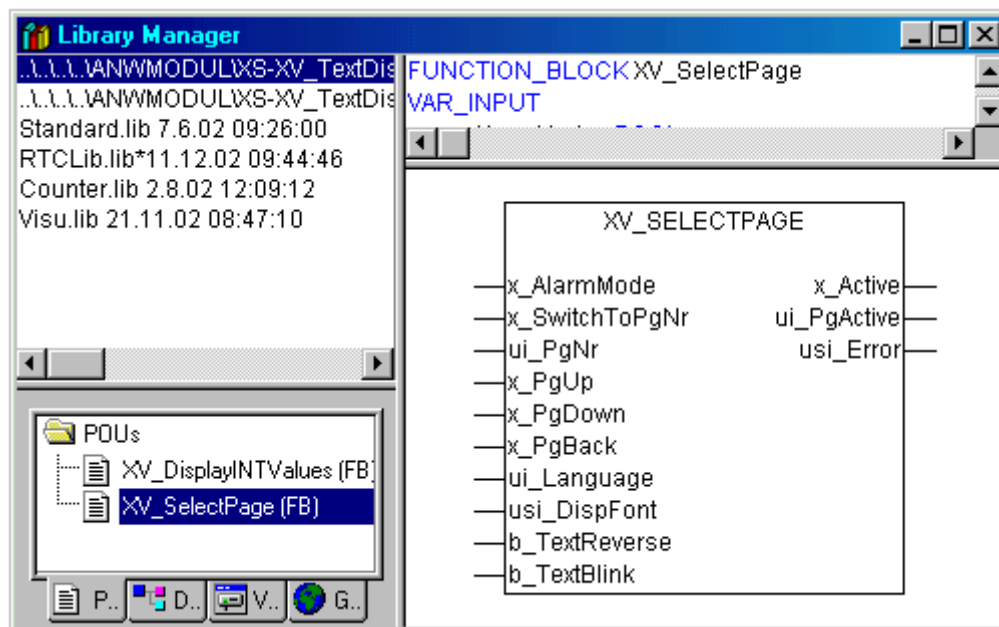
The XV_AlarmTextDisplay function block provides user-friendly alarm management with a history function as well as alarm display on the display PLC. The display function operates on a last-up message basis. The Boolean alarms are managed by means of a dynamic array.

The alarms can be displayed independently of the alarm management function can be activated or deactivated as required.

The alarm texts are taken from the text database.

Installing the libraries

The libraries contained in the application examples are incorporated directly into XSoft. Launch XSoft and add the library XSoft-APP-XV_TextDisplay.lib and XSoft-APP-XV_AlarmTextDisplay in the Library Manager (Windows menu ->Library Manager):



The picture above shows the Library Manager after the libraries have been imported.

Some of the variables must be defined as global variables in order to minimize the memory required and ensure faster parameter assignment of the required arrays. The definition of the global variables and the values that are partly pre-assigned as examples are provided in two export files in the LIB subfolder:

XSOFT-APPEXP-XV_TEXTDISPLAYGLOBALDATA.EXP -> Global variables for the XV_DisplayINTValues function block from the XSoft-APPEXP-XV_TextDisplay.Lib library

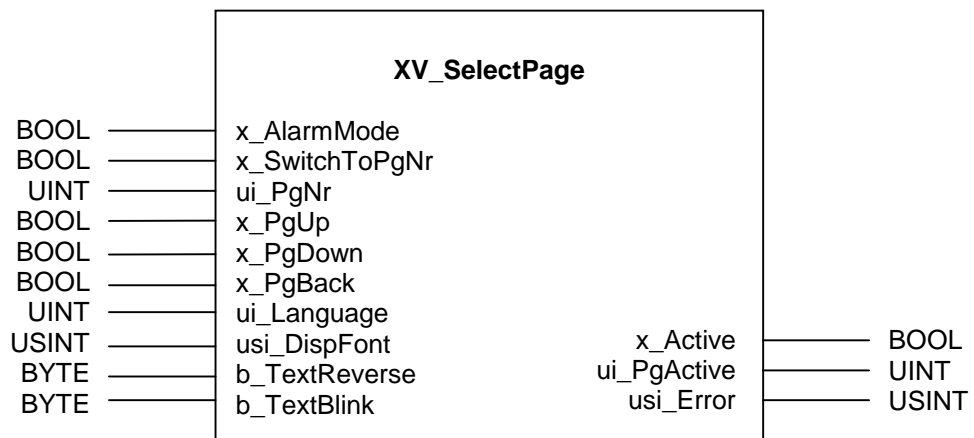
XSOFT-APPEXP-XV_ALARMTEXTDISPLAYGLOBALDATA.EXP -> Global variables for the XV_AlarmTextDisplay function block from the XSoft-APPEXP-XV_AlarmTextDisplay.Lib library

The export file required can be incorporated into the project via Project -> Import. The global variables are then automatically sorted under Global Variables into objects that are named according to the corresponding libraries, and can be assigned parameters directly.

The function blocks access the function blocks of other standard libraries such as Standard.Lib and Visu.Lib. The alarm function block also requires the RTCLib.Lib. These libraries are automatically incorporated when the libraries XSoft-APP-XV_TextDisplay.lib and XSoft-APP-XV_AlarmTextDisplay described here are imported.

Prototypes, parameters and parameter assignment

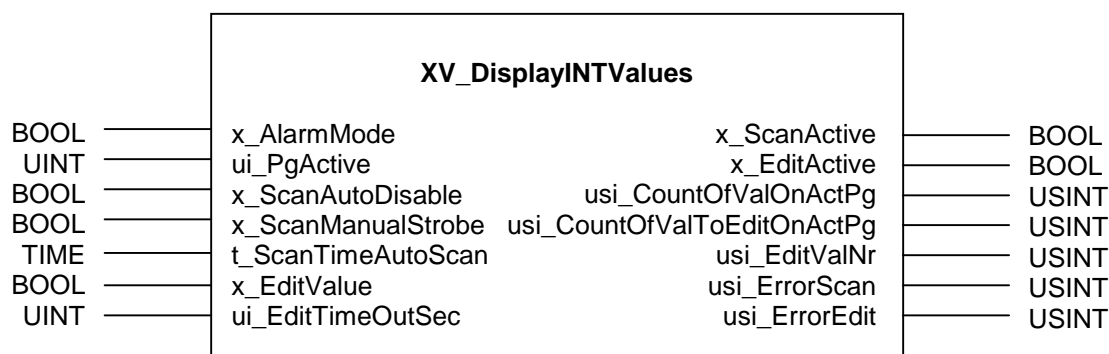
Page selection



VAR_INPUT	
x_AlarmMode	TRUE: Line 0, 1 and 2 of the page are no longer shown when the page is selected -> Used with XV_AlarmTextDisplay when alarm display is activated
x_SwitchToPgNr	Strobe: Selection of page defined by ui_PgNr
ui_PgNr	Page number to be selected with strobe on x_SwitchToPgNr
x_PgUp	Strobe: Select page + 1
x_PgDown	Strobe: Select page - 1
x_PgBack	Strobe: Select the last page displayed
ui_Language	Language to be used with the next selection
usi_DispFont	Font to be used with the next selection
b_TextReverse	Lines which are to be inverted with the next selection, binary coded -> Bit 0..7 -> Text 0..7: 0 -> normal, 1 -> inverted
b_TextBlink	Lines which are to flash with the next selection, binary coded -> Bit 0..7 -> Text 0..7: 0 -> normal, 1 -> flashing (only with XV-101-K84)
VAR_OUTPUT	
x_Active	Selection active
ui_PgActive	Number of the currently displayed page
usi_Error	Error message after selection -> 0: Selection ok, 1: Parameter error, 2: Function block number for display control exceeded, 3: Text database not present -> ui_PgActive -> 0 (empty page), 4: Page (texts) not present -> ui_PgActive -> 0 (empty page), 9: Display not in Display mode (Input mode)

The texts are numbered consecutively from 1 to x in the text editor of the XVision Tool. The editor supports the page allocation by the graphical assignment of the texts in blocks of 4.
The page number of the current cursor position can be shown in the status line of the text editor by setting the environment variable SHOWXVPAGES (set SHOWXVPAGES=1).

Value display / value entry



VAR_INPUT	
x_AlarmMode	TRUE: Values in lines 0, 1 and 2 are no longer displayed and also cannot be edited -> Used with XV_AlarmTextDisplay when alarm display is activated
ui_PgActive	Number of the currently displayed page (Output of XV_SelectPage)
x_ScanAutoDisable	FALSE: Values are scanned and displayed in the time defined by t_ScanTimeAutoScan TRUE: Values are not automatically refreshed -> Refresh via x_ScanManualStrobe -> TRUE has to be activated, when page selection is active
x_ScanManualStrobe	Manual refresh of display values (x_ScanAutoDisable = TRUE)
t_ScanTimeAutoScan	Scan time for refreshing display values
x_EditValue	Switch to Edit mode (setpoint entry) / Setpoint selection
ui_EditTimeOutSec	TimeOut for setpoint entry in seconds
VAR_OUTPUT	
x_ScanActive	Value refresh active
x_EditActive	Setpoint entry active
usi_CountOfValOnActPg	Number of the displayed value on the current page
usi_CountOfValToEditOnActPg	Number of values that can be edited on the current page
usi_EditValNr	Index of the value currently (last) edited
usi_ErrorScan	Errors: Display of values -> 0: No errors, 1: Parameter error, 2: Function block number for display control exceeded
usi_ErrorEdit	Errors: Setpoint entry -> 0: No errors, 1: Parameter error, 2: Function block number for display control exceeded, 5: Setpoint entry aborted, 6: TimeOut on Setpoint entry, 7: No setpoint on this page



VAR_GLOBAL	In XV_TextDisplay under Global Variables
usi_DisplayINTValues_CountOfVal (CONSTANT)	Dimensioning (array size) of all values displayed Default value: 50
i_ar_DisplayINTValues	Values (setpoint and actual values), parameter setting of individual values in the following arrays
ui_ar_DisplayINTValues_Page	Pages on which the values from i_ar_DisplayINTValues assigned via the index are to be displayed, 0 -> From here no more display Default: Values 1..20 assigned
usi_ar_DisplayINTValues_Line	Lines in which the values from i_ar_DisplayINTValues assigned via the index are to be displayed, Default: Values 1..20 assigned
usi_ar_DisplayINTValues_Column	Columns in which the values from i_ar_DisplayINTValues assigned via the index are to be displayed, Default: Values 1..20 assigned
x_ar_DisplayINTValues_EditEnable	TRUE: The value assigned via the index can be edited FALSE: The value assigned via the index cannot be edited Default: Values 1..20 assigned, all can be edited

The size of the array (usi_DisplayINTValues_CountOfVal) for the setpoints and actual values must be pre-defined in the declaration.

The other parameters for setpoints and actual values:

ui_ar_DisplayINTValues_Page

ui_ar_DisplayINTValues_Line

usi_ar_DisplayINTValues_Column and

ui_ar_DisplayINTValues_EditEnable should also be pre-defined in the declaration.

The setpoint values and actual values themselves (i_ar_DisplayINTValues) can naturally be removed and overwritten at any time.

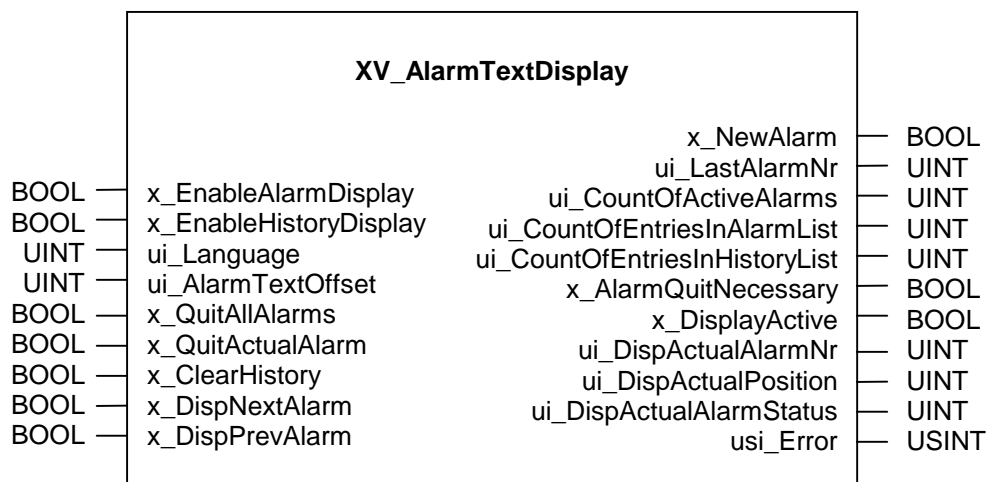
The individual parameters are assigned to the respective setpoint and actual values via the index. For example, the value i_ar_DisplayINTValue[1] is shown on the Page ui_ar_DisplayINTValues_Page[1] in line usi_ar_DisplayINTValues_Line[1] and column usi_ar_DisplayINTValues[1].

This value can be edited if x_ar_DisplayINTValues[1] is set to TRUE. On FALSE it is only a display value and is skipped in Edit mode.

When assigning parameters to the pages in ui_ar_DisplayINTValues_Page, remember that no further values are scanned and displayed after the first 0 entered. In this way it is possible to restrict the display of values during operation.

Integer values are normally displayed in 6 digits.

Alarm processing



VAR_INPUT	
x_EnableAlarmDisplay	Enable Alarm/history display on the device
x_EnableHistoryDisplay	Switch to history display (only if x_EnableAlarmDisplay = TRUE)
ui_Language	Language
ui_AlarmTextOffset	Offset for first alarm text
x_QuitAllAlarms	Acknowledge all alarms present
x_QuitActualAlarm	Acknowledge the alarm currently displayed on the device
x_ClearHistory	Clear all history entries
x_DispatchNextAlarm	Show next alarm
x_DispatchPrevAlarm	Show previous alarm
VAR_OUTPUT	
x_NewAlarm	New alarm
ui_LastAlarmNr	Last alarm generated
ui_CountOfEntriesInAlarmList	Number of alarms present
ui_CountOfEntriesInHistoryList	Number of history entries
x_AlarmQuitNecessary	Acknowledge current alarm necessary
x_DisplayActive	Enable alarm/history display
ui_DispatchActualAlarmNr	No. of the currently displayed alarm
ui_DispatchActualPosition	Position of the currently displayed alarm in the list
ui_DispatchActualAlarmStatus	Status of the currently displayed alarm: 0: No alarm, 1: activated, 2: deactivated, 3: acknowledged, 4: not acknowledged, 5: deactivated and not acknowledged
usi_Error	Error during display: 0: OK, 1: Parameter error, 2: Function block number for display control exceeded, 3: Text database not present, 4: Text not present



VAR_GLOBAL	In XV_AlarmTextDisplay under Global Variables
ui_AlarmTextDisplay_AlarmIndexMax (CONSTANT)	Dimensioning (array size) for alarms Default value: 50
ui_AlarmTextDisplay_HistoryIndexMax (CONSTANT)	Dimensioning (array size) of the history memory Default value: 50
x_ar_AlarmTextDisplay_Alarms	INPUT for FB: Alarms, alarm parameter assignment with subsequent array
x_ar_AlarmTextDisplay_NoQuitNecessary	INPUT for FB: Acknowledgement setting for the alarm assigned via the index -> FALSE: Alarm acknowledgement required TRUE: Alarm acknowledgement not required
x_ar_AlarmTextDisplay _DispAlarmsQuitNecessary	OUTPUT for FB: Indicates which alarms still need to be acknowledged

When setting the dimensioning for the alarm array (ui_AlarmTextDisplay_AlarmIndexMax), remember that the cycle time will increase the larger the size of array.

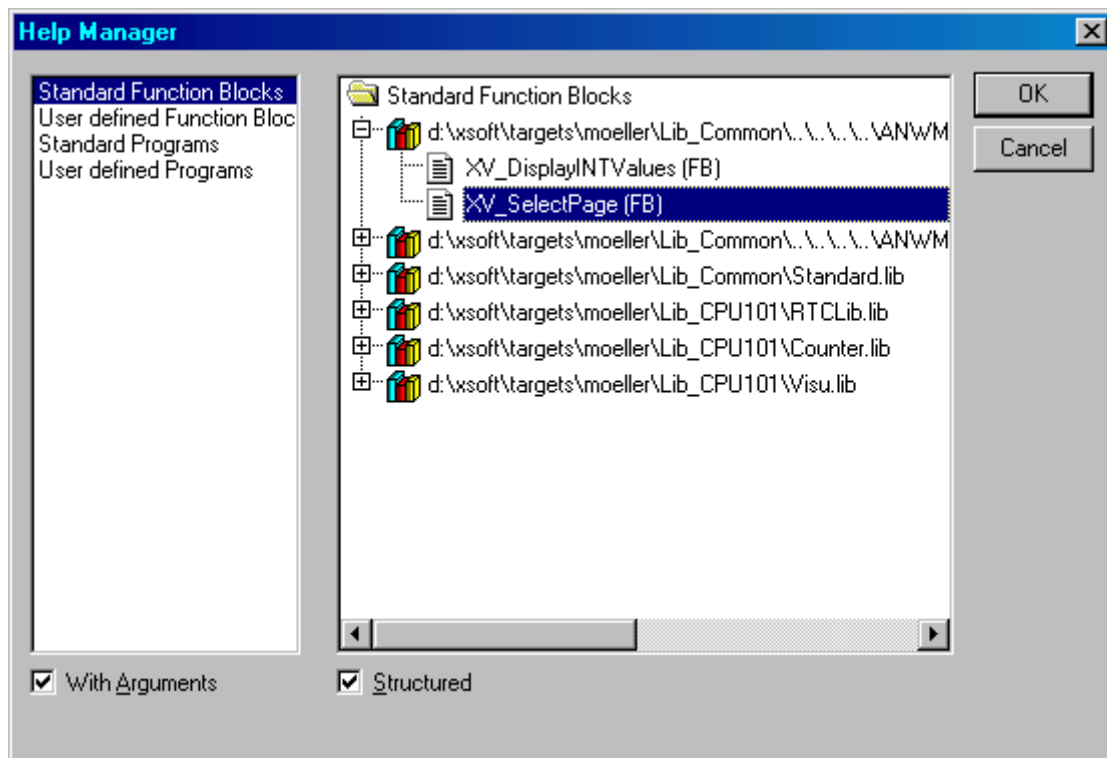
If the page selection and alarm processing functions are to be used simultaneously, the alarm texts should be defined in the text editor of XSoft in a separate column (language), in order to prevent overlaps with texts from the page selection -> Set Parameter ui_Language accordingly on the function block.

Note: The correct time must be set in order to ensure the correct alarm status and correct entry in the history memory!

Incorporating the function blocks

The function block is incorporated by writing the appropriate call, such as `CAL XV_...`, in the IL program.

The function blocks are located in the libraries under the standard function blocks:



The function block should be called cyclically in order to ensure continuous and fast job processing.

Only the function block `XV_AlarmTextDisplay` should be used from the XSoft-APPEXP-XV_AlarmTextDisplay library. All other objects in this library are required internally.



Page selection

Page selection is implemented using the XV_SelectPage function block.

As already described in chapter 3, the function block is used to divide up the texts contained in the text database into virtual pages (starting with Text 1; one text in each line). If an XV-101-K42 is connected, each page contains 4 lines, with the XV-101-K84 8 lines per page with optional display features. For example, with the 8-line display it is possible to display texts 1 to 8 on page 1; page 2 shows texts 9 to 16 etc. The texts of a page always start in column 0 on line 0 of the display.

After the application has started, i.e. after the first cycle of the function block, the page set at the ui_PgNr input will be selected automatically; provided that this input has been assigned a valid setting. If a value has not been assigned or has been assigned 0, page 1 will be selected.

A strobe signal on x_PgUp will select the next page. A strobe on x_PgDown will select the previous page. On x_PgBack, this will select the previously displayed page, and on x_SwitchToPgNr this will select the page with the number defined at ui_PgNr. If this page number is invalid or 0, the page is cleared and error code 4 is generated. The value 0 will be output as the active page number on output ui_PgActive.

Output ui_PgActive will be refreshed accordingly as long as the pages selected are valid. Output x_Active indicates that a page selection is currently active.

The language (ui_Language), Font (usi_DisFont) and text display (b_TextReverse, b_TextBlink) can be reset for each page selection. It should be taken into account that half the number of lines are available when large fonts are used.

The text display is configured with binary parameters, so that the display of text can be set individually -> Bit 0 for text in Line 0, Bit 1 for text in Line 1 etc.

When the x_AlarmMode input is activated, the clearing and display of the first three display lines is suppressed when a page is selected. This should be used when the XV_AlarmTextDisplay alarm function block is being used at the same time. Otherwise the displayed alarm will be overwritten when a page is selected.

If the display is in Entry mode, the page selection is disabled (usi_Error = 9).

Value display / value entry

The XV_DisplayINTValues function block works in conjunction with the Page selection function. Up to 8 integer variables from the global array i_ar_DisplayINTValues can be shown and edited on each page. Each value of the array is thus shown exactly once.

The parameter assignment for the setpoints and actual values to be shown is described in detail in chapter 4.

Page number ui_PgActive determines which variables are to be displayed. The display is refreshed automatically within the scan time t_ScanTimeAutoScan with every page selection. This requires that the automatic scan function x_ScanAutoDisable = FALSE is not deactivated by means of x_ScanAutoDisable = TRUE.

Automatic scanning must be disabled when page selection is active. This means that the input can be connected directly with the x_Active output of the XV_SelectPage function block.

If automatic scanning is permanently deactivated (x_ScanAutoDisable = TRUE), manual scanning is possible via the x_ScanManualStrobe input.

The x_ScanActive output indicates whether a scan operation is taking place with display refresh.

usi_CountOfValOnActPg indicates the number of the values shown on the current page.

Output usi_CountOfValToEditOnActPg indicates how many of the values shown can be edited.

If an error is indicated during display (usi_ErrorScan), all values are shown up to the point where the error occurred.

The first strobe on the x_EditValue input activates Variable entry mode. The cursor is positioned on the first value that can be edited. Further switching of x_EditValue will move the cursor to the next value that can be edited. The order of this is determined by the declaration. Confirming a value entry with ENTER will exit Edit mode. Pressing ESC, or a renewed actuation of x_EditValue when the cursor is already on the last value that can be entered will abort the entry. The entry can, however, be restarted at any time by actuating x_EditValue.

The x_EditActive output indicates Edit mode and the usi_EditValNr output indicates the setpoint just being



edited or the last edited one if Edit mode has been terminated.

All of the values displayed will continue to be refreshed in Edit mode with the defined conditions apart from the currently edited values. The displayed values are also refreshed when changing from one setpoint to the next.

The activation of the `x_AlarmMode` input suppresses with every refresh the display of all variables that are defined in the first three display lines. This should be used when the `XV_AlarmTextDisplay` alarm function block is activated at the same time. Otherwise the variables will possibly be written over the alarm information. Values can also be displayed inside the alarm text if the variables are suitably positioned.

Alarm processing

Alarm management is active as soon as the `XV_AlarmTextDisplay` function block is called; irrespective of the status of `x_EnableAlarmDisplay`.

The `x_NewAlarm` output indicates that at least one alarm is active in the global alarm array `x_ar_AlarmTextDisplay_Alarms`. The `ui_LastAlarmNr` output shows the number of the last alarm activated and the `ui_CountOfActiveAlarms` output indicates the number of currently active alarms. The `ui_CountOfEntriesInAlarmList` indicates the number of entries in the alarm list. All active or not yet acknowledged alarms are added to this.

The `ui_CountOfEntriesInHistoryList` output contains the number of entries in the history memory. The history memory is designed as a ring buffer. It contains information on activated, deactivated and acknowledged alarms. The `x_AlarmQuitNecessary` output is only set when the alarm display is activated, and shows that alarms still need to be acknowledged. The global array `x_ar_AlarmTextDisplay_DisAlarmsQuitNecessary` indicates which alarms still need to be acknowledged.

`x_EnableAlarmDisplay = TRUE` activates the alarm display function and sets the `x_DisplayActive` output. If `x_EnableHistoryDisplay` is also set, the display switches to the history display. This is where the texts of the alarms are displayed from the history memory with date / time / status.

The display uses two lines from line 0/column 0: 1st line -> Date/Time/Status, 2nd line -> Alarm text.

```
2002-12-01-15:34:22+
This is alarm 1
```

The status is always shown by a character directly after the time.

The following symbols are used in the History display:

```
+      activated
-      acknowledged
x      deactivated
```

The status line is limited to 20 characters so that it can be displayed in the same way on every display. Actuating `x_DispNextAlarm` and `x_DispPrevAlarm` enables a switch to the next or previous alarm here and in the alarm display. `x_ClearHistory` clears the history memory.

In addition to date, time, status and alarm text, Alarm display mode (last-up message display) shows in the third line which alarm (chronologically from the last one) of the number of active alarms is currently displayed.

```
2002-12-01-15:34:22+
This is alarm 1
```

Alarm 1/5

The appropriate status is also indicated here as a symbol directly behind the time.

The following symbols are used in the Alarm display:

- + activated
- acknowledged
- * deactivated and not acknowledged

Input `x_QuitActualAlarm` acknowledges the currently displayed alarm (only if `AlarmDisplay` is active).
`x_QuitAllAlarm` enables all alarms in the Alarm list to be acknowledged (even if `AlarmDisplay` is not active).

Output `ui_ActualAlarmNr` indicates which alarm text is currently displayed. Output `ui_DisplActualPosition` indicates the current position of the pointer within the list of active alarms. The position is determined chronologically on a last-up message basis.

Output `ui_DisplActualAlarmStatus` shows the status of the currently displayed alarm:

- 0 No alarm
- 1 Alarm activated
- 2 Alarm deactivated
- 3 Alarm active and acknowledged
- 4 Alarm active and not acknowledged
- 5 Alarm deactivated and not acknowledged

General information on the display PLC

The three function blocks described access the `Visu.Lib` library which contains all the control functions required by the display.

The function blocks from the `Visu.Lib` can, of course, be used together with the function block described here to execute other display functions. It must, however, be ensured that no time overlaps occur that may cause display control functions not to be executed or only in part. It must also be taken into account that a maximum of 10 display jobs can be actively processed simultaneously in the stack.

Each additional job started is rejected if 10 are already active. The error code (Error Code = 2, when stack is full) after a job is completed must always be taken into account.

Jobs are best processed in succession.

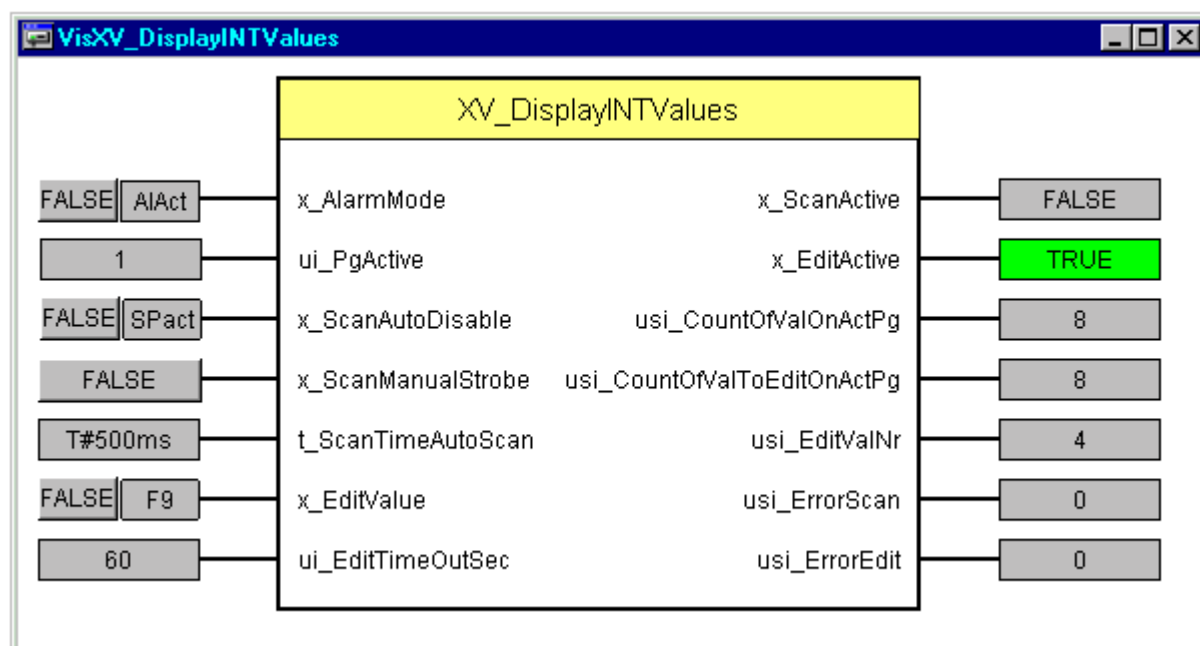
Program examples / visualization

Subfolder ..\XS-XV_TextDisplay\EXAMPLE\.. contains program examples in the form of export files.

Here are two export files VisXV_... which contain a program example and a visualization example of the same name for each function block:

VisXV_TextDisplay.Exp

Example programs VisXV_SelectPage, VisXV_DisplayINTValues with visualizations of the same name



Example: Visualization VisXV_DisplayINTValues

The example programs are set so that the corresponding function blocks can be operated fully from the same location. Some parameters are also interlinked in the program examples. Some inputs can also be accessed via the keypad (Next page >/Previous Page <, as well as the activation to Setpoint entry via F9 and the selection of the previously displayed page with Shift).

The visualization VisXV_DisplayINTValuesGlobalData provided can also be used with the PLC in HALT or RUN status for assigning the global data parameters of the XV_DisplayINTValues function block.

VisXV_AlarmTextDisplay.Exp

Example program VisXV_AlarmTextDisplay with the visualization of the same name.

The CursorUp, CursorDown and ENTER keys of the text display are also used here to scroll the alarm display and acknowledge alarms.

To effectively link the alarm display with the example programs for page selection and value display (VisXV_SelectPage and VisXV_DisplayINTValues), the blocks in these programs that have been masked out as comments between the two lines (*-----*) should be activated. The x_AlarmMode inputs of XV_SelectPage and XV_DisplayINTValues are linked with the alarm display. The program headers also provide relevant information.

Some default settings of the variables have been set for an 8-line display.

When using 4-line displays, the global declarations should be adjusted accordingly.

The VisXV_TextDisplay.Xtd text file can be downloaded for the visualizations on the display.



Proceed as follows in order to commission the example:

1. Open a new project
2. Load the text database VisXV_TextDisplay into the PLC configuration via Other Parameters
3. Add the required library XSoft-APPEXP-XV_TextDisplay and XSoft-APPEXP XV_AlarmTextDisplay in the Library Manager, as previously described
4. Import the required global definitions XS-OFT-APPEXP-XV_TEXTDISPLAYGLOBALDATA.EXP and XS-OFT-APPEXP-XV_ALARMTEXTDISPLAYGLOBALDATA.EXP
5. Import the required example programs XS-VISXV_TEXTDISPLAY.EXP and XS-VISXV_ALARMTEXTDISPLAY.EXP
6. In order to run the alarm management function together with the page selection and the value display functions, activate, as previously described, the selected blocks that have been commented out in the programs VisXV_SelectPage and VisXV_DisplayINTValues
7. Then incorporate the calls of the required example programs in your main program PLC_PRG
8. Debug everything - Compile everything – Download and start

Example application

Subfolder ..\XS-XV_TextDisplay\EXAMPLE\.. contains an example application XS-EXPXV_TEXTDISPLAY.EXP. The relevant text file for the display types AppXV_TextDisplayK42 and AppXV_TextDisplayK84 can be downloaded to the display using the XVision Tool. The associated global definitions are already loaded during the import.

The AppXV_TextDisplay example program controls the display of 6 pages partly containing variables. The alarm manager is also incorporated. 8 alarm texts are defined. The alarms can also be activated or deactivated via the visualization AppXV_TextDisplay.

The application is controlled via the keypad:

F1:	Go to start page
F2:	Alarm display On/Off
F3:	Production
F4:	Motor speed
F5:	Motor temperature
F6:	Language selection (German/English)
F7:	Set time (to 11.12.2002, 10:00:00)
F9:	Setpoint entry
>:	Next page
<:	Previous page
CursorUp:	Show previous alarm
CursorDown:	Show next alarm
ENTER:	Acknowledge current alarm
CLEAR:	Acknowledge all alarms