

```

FUNCTION Disp_RegisterVariable : BOOL
(* Register one IEC-Variable for using as display variable *)
VAR_INPUT (* Symbolic name for display variable *)
  sName :STRING(16); (* Address of corresponding IEC-Variable *)
  dwAddress :DWORD; (* Datatyp of corresponding IEC-Variable *)
  eVarTyp :DISP_VARTYP; (* Returnvalue: *)
END_VAR

```

Figure 98: Function Disp\_RegisterVariable

In this program section you can also define the position of the variables on the display by specifying the Line and Column. Call the function block (FB) "Disp\_DisplayElement" and assign parameters for the inputs sName, byLine, and byColumn, e.g.:

```

fbDisplayElement1.sName := 'S1';
fbDisplayElement1.byLine := 1;
fbDisplayElement1.byColumn := 1;

```

The element S1 with the text "motor1" would be displayed in the first line starting from the first column.

In order to display/enter several elements call the function block "Disp\_DisplayElement" in the following program section that is continuously processed and assign external inputs to the "xEnable" inputs, e.g. I3.

```

VAR
  xIsDisplayEnabled: BOOL;
  fbGetDisplayInfo:          Disp_GetDisplayInfo;
  fbDisplayElement1: Disp_DisplayElement;
  fbDisplayElement2: Disp_DisplayElement;
  fbDisplayElement3: Disp_DisplayElement;
  fbDisplayElement4: Disp_DisplayElement;
  byError:                  BYTE;
  byValue:                   BYTE;
  wValue:                    WORD;
END_VAR

```

(\* Initialisation in the first cycle after program start \*)

```
IF g_xFirstCycleAfterStartProgram = TRUE THEN
```

```
Disp_ClearScreen(xEnable:=TRUE);
```

```
Disp_RegisterVariable('S1', ADR(g_sDisp_String1),
DISP_TYP_STRING);
```

```
Disp_RegisterVariable('S2', ADR(g_sDisp_String2),
DISP_TYP_STRING);
```

```
Disp_RegisterVariable('V1', ADR(PLC_PRG.byValue),
DISP_TYP_BYTE);
```

```
Disp_RegisterVariable('V2', ADR(PLC_PRG.wValue),
DISP_TYP_WORD);
```

```

fbDisplayElement1.sName      := 'S1';
fbDisplayElement1.byLine    := 1;
fbDisplayElement1.byColumn  := 1;

```

```

fbDisplayElement2.sName     := 'S2';
fbDisplayElement2.byLine    := 3;

```

```

fbDisplayElement2.byColumn  := 1;

fbDisplayElement3.sName     := 'V1';
fbDisplayElement3.byLine    := 1;
fbDisplayElement3.byColumn  := 8;
fbDisplayElement3.byDigits  := 4;
fbDisplayElement3.byPrecision := 1;

```

```

fbDisplayElement4.sName     := 'V2';
fbDisplayElement4.byLine    := 3;
fbDisplayElement4.byColumn  := 8;
fbDisplayElement4.byDigits  := 6;
fbDisplayElement4.byPrecision := 1;

```

(\* The first cycle is completed, reset flag \*)

```
g_xFirstCycleAfterStartProgram := FALSE;
```

```
END_IF
```

```
xIsDisplayEnabled := Disp_EnableDisplay(I1, I2);
```

```
fbDisplayElement1( xEnable:= I3 );
```

```
fbDisplayElement2( xEnable:= I5 );
```

```
fbDisplayElement3( xEnable:= I3 );
```

```
fbDisplayElement4( xEnable:= I5 );
```

► Start the programs.

### Example of a screen output with texts and value entries

With the Disp\_DisplayPage function block

The following display has to be implemented

The contents of the variables MO11 and TEMP8 are changed continuously by the user program.

MO11	3.5
TIM14	0
MOZ14	0
TEMP8	183

Figure 99: Example of a page for entries and outputs

### Operations via the PLC inputs

- I1 = FALSE: Status display
- I1 = TRUE: Entry/output mode
- I2 = FALSE: ESC button active
- I2 = TRUE: ESC button disabled
- I3 = TRUE: The values are refreshed by the program.
- I4 = TRUE: Entry active.

### Execution

The example program consists of programs:

- "Startprogram": (called on system event Start)
  - Auxiliary variable "g\_xFirstCycleAfterStartProgram" is set.
- PLC\_PRG:
  - 2 values are incremented.
  - The program "Visualisation" is called.
- VISUALIZATION
  - Registering and positioning of variables on the display in the first cycle.
  - The auxiliary variable g\_xFirstCycleAfterStartProgram is reset.
  - Activation of Entry/output mode (I1).
  - Enable ESC button (I2).
  - Start display (I3).
  - Start entry (I4).

### Declaring display variables

- ▶ First declare for each text element that you wish to display, such as "MO11", a variable of type "String" in the "Global\_Variables\_Display" folder as in the following example::

```
VAR_GLOBAL
g_sDisp_String1  :STRING := 'MO11  ':';
g_sDisp_String2  :STRING := 'TIM14 ':';
g_sDisp_String3  :STRING := 'MOZ14 ':';
g_sDisp_String4  :STRING := 'TEMP8 ':' ;
END_VAR
```

- ▶ Create an auxiliary variable and write the program "Startprogram" as in the "Example of text and values output".
- ▶ Write the PLC\_PRG and Visualisation programs according to the following example:

```
PROGRAM PLC_PRG (*****)
VAR
    fbTimer1      :TON;
    (* Display values of the application *)
    byValue       :BYTE;
    wValue        :WORD;
    dwValue       :DWORD;
    usiValue      :USINT;
    siValue       :SINT;
END_VAR

-----
fbTimer1(IN:=NOT fbTimer1.Q , PT:=t#50ms );
IF fbTimer1.Q = TRUE THEN
    usiValue := usiValue + 1;
    byValue:=byValue+1;
END_IF

Visualisation(); (* Call visualisation *)
```

```

PROGRAM Visualization (*****)
VAR
    xIsDisplayEnabled      :BOOL;
    fbDisplayPage1        :Disp_DisplayPage;
    byError                :BYTE;
    siValue                :SINT;
END_VAR

-----
(* Initialisation in the first cycle after program start *)
IF g_xFirstCycleAfterStartProgram = TRUE THEN

Disp_RegisterVariable('S1', ADR(g_sDisp_String1), DISP_TYP_STRING);
Disp_RegisterVariable('S2', ADR(g_sDisp_String2), DISP_TYP_STRING);
Disp_RegisterVariable('S3', ADR(g_sDisp_String3), DISP_TYP_STRING);
Disp_RegisterVariable('S4', ADR(g_sDisp_String4), DISP_TYP_STRING);
Disp_RegisterVariable('V1', ADR(PLC_PRG.byValue), DISP_TYP_BYTE);
Disp_RegisterVariable('V2', ADR(PLC_PRG.wValue), DISP_TYP_WORD);
Disp_RegisterVariable('V3',ADR(PLC_PRG.dwValue),
DISP_TYP_DWORD);
Disp_RegisterVariable('V4', ADR(PLC_PRG.usiValue), DISP_TYP_USINT);

fbDisplayPage1.aElementDescription[1].sName      := 'S1';
fbDisplayPage1.aElementDescription[1].byLine    := 1;
fbDisplayPage1.aElementDescription[1].byColumn  := 1;
fbDisplayPage1.aElementDescription[2].sName      := 'S2';
fbDisplayPage1.aElementDescription[2].byLine    := 2;
fbDisplayPage1.aElementDescription[2].byColumn  := 1;
fbDisplayPage1.aElementDescription[3].sName      := 'S3';
fbDisplayPage1.aElementDescription[3].byLine    := 3;
fbDisplayPage1.aElementDescription[3].byColumn  := 1;
fbDisplayPage1.aElementDescription[4].sName      := 'S4';
fbDisplayPage1.aElementDescription[4].byLine    := 4;
fbDisplayPage1.aElementDescription[4].byColumn  := 1;

fbDisplayPage1.aElementDescription[5].sName      := 'V1';
fbDisplayPage1.aElementDescription[5].byLine    := 1;
fbDisplayPage1.aElementDescription[5].byColumn  := 13;
fbDisplayPage1.aElementDescription[5].byDigits  := 4;
fbDisplayPage1.aElementDescription[5].byPrecision := 1;
fbDisplayPage1.aElementDescription[5].xInputEnable := FALSE;
fbDisplayPage1.aElementDescription[5].diMinInputValue := 1;
fbDisplayPage1.aElementDescription[5].diMaxInputValue := 100;
fbDisplayPage1.aElementDescription[6].sName      := 'V2';
fbDisplayPage1.aElementDescription[6].byLine    := 2;
fbDisplayPage1.aElementDescription[6].byColumn  := 12;
fbDisplayPage1.aElementDescription[6].byDigits  := 5;
fbDisplayPage1.aElementDescription[6].byPrecision := 0;
fbDisplayPage1.aElementDescription[6].xInputEnable := TRUE;
fbDisplayPage1.aElementDescription[6].diMinInputValue := 0;
fbDisplayPage1.aElementDescription[6].diMaxInputValue := 33333;
fbDisplayPage1.aElementDescription[7].sName      := 'V3';

```