Operating instructions
ISOBUS gateway

CR3121

Contents

1   Preliminary note

You will find instructions, technical data, approvals and further information using the QR code on the unit / packaging or at www.ifm.com.

1.1   Symbols used

✓	Requirement
▶	Instructions
▷	Reaction, result
[...]	Designation of keys, buttons or indications
→	Cross-reference
⚠	Important note Non-compliance may result in malfunction or interference.
ℹ	Information Supplementary note

1.2   Warnings used

ATTENTION	Warning of damage to property

⚠ **CAUTION**	Warning of personal injury ▷ Slight reversible injuries may result.

⚠ **WARNING**	Warning of serious personal injury ▷ Death or serious irreversible injuries may result.

2   Safety instructions

- The unit described is a subcomponent for integration into a system.

  – The system architect is responsible for the safety of the system.

  – The system architect undertakes to perform a risk assessment and to create documentation in accordance with legal and normative requirements to be provided to the operator and user of the system. This documentation must contain all necessary information and safety instructions for the operator, the user and, if applicable, for any service personnel authorised by the architect of the system.

- Read this document before setting up the product and keep it during the entire service life.

- The product must be suitable for the corresponding applications and environmental conditions without any restrictions.

- Only use the product for its intended purpose ($\rightarrow$ Intended use).

- If the operating instructions or the technical data are not adhered to, personal injury and/or damage to property may occur.

- The manufacturer assumes no liability or warranty for any consequences caused by tampering with the product or incorrect use by the operator.

- Installation, electrical connection, set-up, programming, configuration, operation and maintenance of the product must be carried out by personnel qualified and authorised for the respective activity.

- Protect units and cables against damage.

- Replace damaged units, otherwise the technical data and safety will be impaired.

- Observe applicable documents.

3   Intended use

The device serves as an interface between an agricultural vehicle (tractor) and an add-on unit. The ISOBUS gateway exchanges data bidirectionally between the Virtual Terminal (VT) (ISOBUS) and the add-on unit or controller (CAN bus).

3.1   Properties

•   Closed plastic housing for mounting on the add-on unit inside or outside a control cabinet (IP67)

•   Configurable ISOBUS visualisation via IEC 61131 application

4   Function

The unit fulfils the following functions:

*   1 CAN interface with ISO11783 protocol (ISOBUS)

*   1 CAN interface with CAN layer 2 support (CAN bus → control)

*   Transmission of the designed data masks (IOP files) to the VT (Virtual Terminal) via ISOBUS

*   Transfer of the visualisation configuration from the controller to the VT

*   Bidirectional communication between the VT (ISOBUS) and the control on the add-on unit (CAN bus)

    –   Transmission of values and commands from the controller, display on the VT

    –   Transmission of inputs and actions from the user at the VT to the controller

4.1   Application example
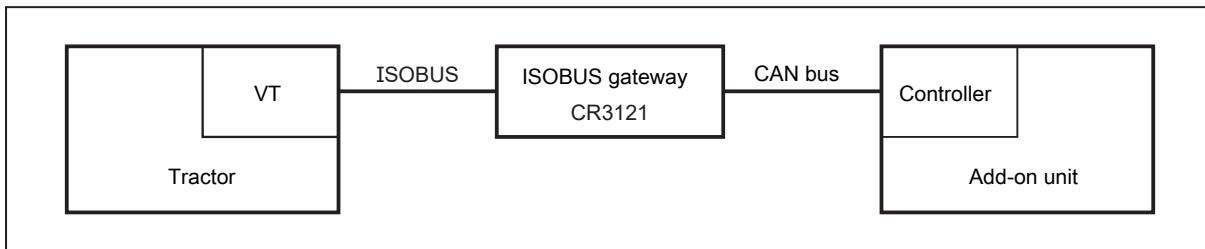
5 Installation

! ▶ Disconnect the power of the machine before installation.

! ▶ For installation choose a flat mounting surface.

! The following applies to all types of mounting:

The responsibility for the compliance with the requirements concerning mounting of the device in the application with regard to shock, vibration, acceleration and weight lies with the system architect.

▶ Fix the unit to the mounting surface using 4 M4 mounting screws and washers.
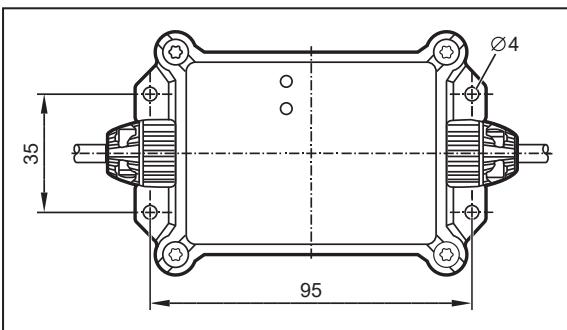
Fig. 1: Installation

6 Electrical connection

> [!] The unit must be connected by a qualified electrician.
>
> ▶ Observe the national and international regulations for the installation of electrical equipment.

▶ Disconnect power.

▶ Connect the unit as follows:
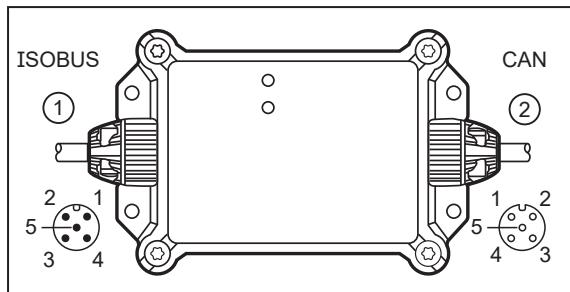
Fig. 2: Electrical connection

1: 5-pole M12 connector → ISOBUS
- Pin 1: not used
- Pin 2: VBB
- Pin 3: GND
- Pin 4: CAN_H
- Pin 5: CAN_L

2: 5-pole M12 socket → CAN bus
- Pin 1: not used
- Pin 2: not used
- Pin 3: not used
- Pin 4: CAN_H
- Pin 5: CAN_L

6.1 Connection accessories

More information about available accessories at www.ifm.com

6.2 Connection technology

> [!] Observe all notes on connection technology.

▶ Note the device label.

▶ Use M12 connectors with gold-plated contacts.

▶ The M12 connection parts in the device comply with the ingress resistance requirements of the standard EN 61076-2-101. To adhere to the protection rating, only cables certified to this standard must be used. The system architect undertakes to ensure the ingress resistance of cables they have cut to length.

▶ Carry out the fitting according to the indications of the cable manufacturer. The permitted maximum is 10 Nm.

▶ During installation, place the M12 connector vertically so that the coupling nut will not damage the thread.

7   Set-up

After power on, the unit is ready for operation.

The operating status is signalled by the two LEDs.

1: LED power
2: LED status
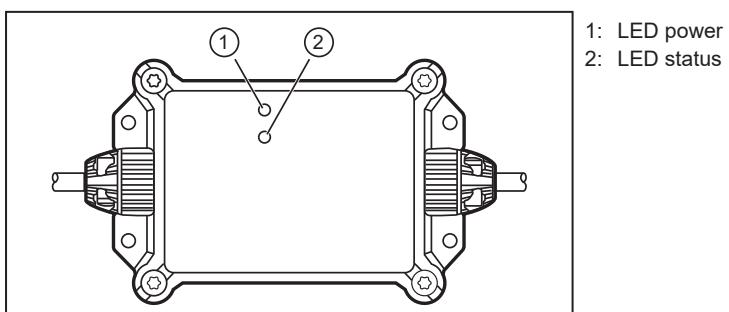
Fig. 3: LEDs

LED	Colour	Status	Designation
Power		off	Supply voltage too low.
	green	on	Waiting for connection to an ISOBUS-VT
		flashing	Communication with a ISOBUS-VT
	red	on	Incorrect / missing product licence
Status		off	Supply voltage too low.
	green	on	Unit in operating mode, connection interrupted
		flashing	Unit in operating mode, connection established
	red	on	Communication error

8   Programming

The basic procedure for programming is as follows:

▶ Program the application (CODESYS).

▶ Configure the visualisation (CODESYS).

▶ Load the application into the controller.

   ▷ Visualisation is automatically loaded onto the Virtual Terminal (VT) via the CR3121 (ISOBUS).

▷ In operation, bidirectional communication between VT (ISOBUS) and control (CAN bus) via the CR3121.

The CR3121 unit is supplied with IOP (Implement Object Pool) pre-installed. The configuration takes place exclusively via CODESYS with the help of the libraries.

8.1   Required documentation

The online help CODESYS is available for download on the Internet: www.ifm.com

A table with the icons is included in the help file.

In the download area, step-by-step instructions, icon overviews and sample projects are available for download.

8.2   Visualisation elements

In the following, the basic function blocks of the ifm_ISOBUS library are presented by means of examples. The input parameters must be adapted to the specific project. It is advisable to use variables instead of constant values in order to be able to change colours and numerical values at runtime.

Up to 12 data masks (pages) are available for configuration. Each data mask has 6 slots that can be filled with operating and display elements as required.

Basic functions can be configured for all data masks (pages) via operating elements in the footer as well as via softkeys on the left and right edges.

If a function module has a status output, the following status messages are possible:

• Initialisation

• Idle

• Operation

• Error

8.2.1   Defining the title of a data mask

The function module `FB_DataMask_Title` defines the title of a specific data mask (page).

Configuration example in CODESYS:
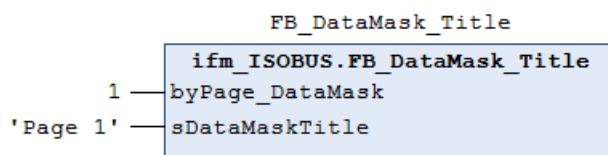
```
                              FB_DataMask_Title
                      ┌─────────────────────────────┐
                      │  ifm_ISOBUS.FB_DataMask_Title│
              1 ──────┤ byPage_DataMask             │
         'Page 1' ────┤ sDataMaskTitle              │
                      └─────────────────────────────┘
```

Visualisation example:

8.2.2 Issuing warnings

The function module `FB_Warnings` issues warning symbols as well as acoustic warnings.

The input parameter `wIconWarning<n>` defines the type of warning symbol.

The input parameter `byStateIcon<n>` defines the colour of the warning symbol:

- 0 = white (e.g. off)
- 1 = green (e.g. on)
- 2 = yellow (e.g. warning)
- 3 = red (e.g. alarm)

The input parameter `xAcusticWarning` switches the acoustic warning on/off. If the acoustic warning is switched on, the following signal tones are possible:

- If at least one input parameter `byStateIcon<n>` has the value 2, a warning tone will sound.
- If at least one input parameter `byStateIcon<n>` has the value 3, an alarm tone will sound.

**Configuration example
in CODESYS:**

```
                                                                      FB_Warnings
                                                                 ifm_ISOBUS.FB_Warnings
         ifm_ISOBUS.Warning.HydraulicFilter ── wIconWarning1      StateWarnings ──
                                          0 ── byStateIcon1
         ifm_ISOBUS.Warning.HydraulicLevel ── wIconWarning2
                                          1 ── byStateIcon2
      ifm_ISOBUS.Warning.HydraulicPressure ── wIconWarning3
                                          2 ── byStateIcon3
   ifm_ISOBUS.Warning.HydraulicTemperature ── wIconWarning4
                                          3 ── byStateIcon4
           ifm_ISOBUS.Warning.CoolantLevel ── wIconWarning5
                                          0 ── byStateIcon5
     ifm_ISOBUS.Warning.CoolantTemperature ── wIconWarning6
                                          1 ── byStateIcon6
                ifm_ISOBUS.Warning.Break ── wIconWarning7
                                          2 ── byStateIcon7
          ifm_ISOBUS.Warning.LoadingVolume ── wIconWarning8
                                          3 ── byStateIcon8
             ifm_ISOBUS.Warning.LowVoltage ── wIconWarning9
                                          0 ── byStateIcon9
            ifm_ISOBUS.Warning.Temperature ── wIconWarning10
                                          1 ── byStateIcon10
                  ifm_ISOBUS.Warning.Tire ── wIconWarning11
                                          2 ── byStateIcon11
         ifm_ISOBUS.Warning.UndefinedError ── wIconWarning12
                                          3 ── byStateIcon12
                                             ─ wIconWarning13
                                             ─ byStateIcon13
                                             ─ wIconWarning14
                                             ─ byStateIcon14
                                             ─ wIconWarning15
                                             ─ byStateIcon15
                                             ─ wIconWarning16
                                             ─ byStateIcon16
                                             ─ wIconWarning17
                                             ─ byStateIcon17
                                             ─ wIconWarning18
                                             ─ byStateIcon18
                           xAcousticWarning ── xAcousticWarning
                                             ─ xReset
```

**Visualisation
example:**
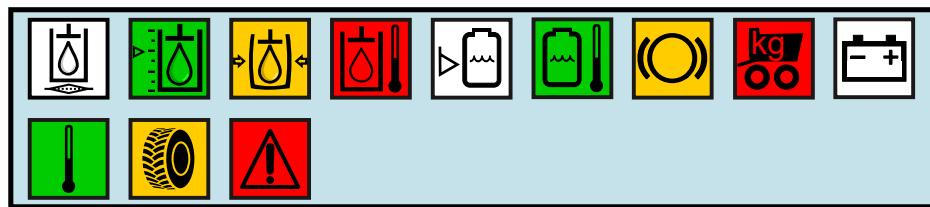
8.2.3 Adapting the softkey mask

The function block `FB_SoftkeyMask` defines the softkeys for all data masks (pages). A maximum of 12 softkeys can be configured. The actual appearance may differ from the example and depends on the manufacturer of the VT used.

The input parameter `xSoftkey<n>` hides/unhides the softkey.

The input parameter `wID_IconSoftkey<n>` defines the softkey symbol.

The input parameter `byColorIcon<n>` defines the colour of the softkey symbol:

- 0 = black
- 1 = green
- 2 = yellow

• 3 = red

**Configuration example
in CODESYS:**

```
                                              FB_SoftkeyMask
                                          ifm_ISOBUS.FB_SoftkeyMask
                            TRUE ── xSoftkey1                          State ──────────
          ifm_ISOBUS.w_Icons.Arrow_Up ── wID_IconSoftkey1    axSoftkeyStates ── axSoftKeyStates
                               0 ── byColorIcon1
                            TRUE ── xSoftkey2
        ifm_ISOBUS.w_Icons.Arrow_Down ── wID_IconSoftkey2
                               1 ── byColorIcon2
                            TRUE ── xSoftkey3
        ifm_ISOBUS.w_Icons.Arrow_Left ── wID_IconSoftkey3
                               2 ── byColorIcon3
                            TRUE ── xSoftkey4
       ifm_ISOBUS.w_Icons.Arrow_Right ── wID_IconSoftkey4
                               3 ── byColorIcon4
                           FALSE ── xSoftkey5
     ifm_ISOBUS.w_Icons.Arrow_TiltLeft ── wID_IconSoftkey5
                               0 ── byColorIcon5
                           FALSE ── xSoftkey6
    ifm_ISOBUS.w_Icons.Arrow_TiltRight ── wID_IconSoftkey6
                               1 ── byColorIcon6
                            TRUE ── xSoftkey7
              ifm_ISOBUS.w_Icons.Plus ── wID_IconSoftkey7
                               2 ── byColorIcon7
                           FALSE ── xSoftkey8
             ifm_ISOBUS.w_Icons.Minus ── wID_IconSoftkey8
                               3 ── byColorIcon8
                            TRUE ── xSoftkey9
          ifm_ISOBUS.w_Icons.Arrow_Up ── wID_IconSoftkey9
                               0 ── byColorIcon9
                           FALSE ── xSoftkey10
        ifm_ISOBUS.w_Icons.Arrow_Down ── wID_IconSoftkey10
                               1 ── byColorIcon10
                            TRUE ── xSoftkey11
        ifm_ISOBUS.w_Icons.Arrow_Left ── wID_IconSoftkey11
                               2 ── byColorIcon11
                           FALSE ── xSoftkey12
       ifm_ISOBUS.w_Icons.Arrow_Right ── wID_IconSoftkey12
                               3 ── byColorIcon12
                          xReset ── xReset
```

**Visualisation
example:**

8.2.4 Defining the footer

The function module `FB_Footer` defines the footer for all data masks (pages). A maximum of 4 controls can be configured in the footer.

The input parameter `xButton<n>` fades the control elements on/off (counting direction from left to right control elements 1...4).

The input parameter `wIconButton<n>` defines the symbol on the control element.

The input parameter `byColorIcon<n>` defines the colour of the symbol:

• 0 = black

• 1 = green

• 2 = yellow

- 3 = red

Configuration example in CODESYS:

```
                                          FB_Footer
                                   ifm_ISOBUS.FB_Footer
                          TRUE ── xButton1                        State ──────────
     ifm_ISOBUS.w_Icons.Power ── wIconButton1   axFooterButtonStates ── axFooterButtonState
                             1 ── byColorIcon1
                                 ─ xButton2
                                 ─ wIconButton2
                                 ─ byColorIcon2
                          TRUE ── xButton3
       ifm_ISOBUS.w_Icons.Day ── wIconButton3
                                 ─ byColorIcon3
                          TRUE ── xButton4
     ifm_ISOBUS.w_Icons.Night ── wIconButton4
                                 ─ byColorIcon4
                        xReset ── xReset
```
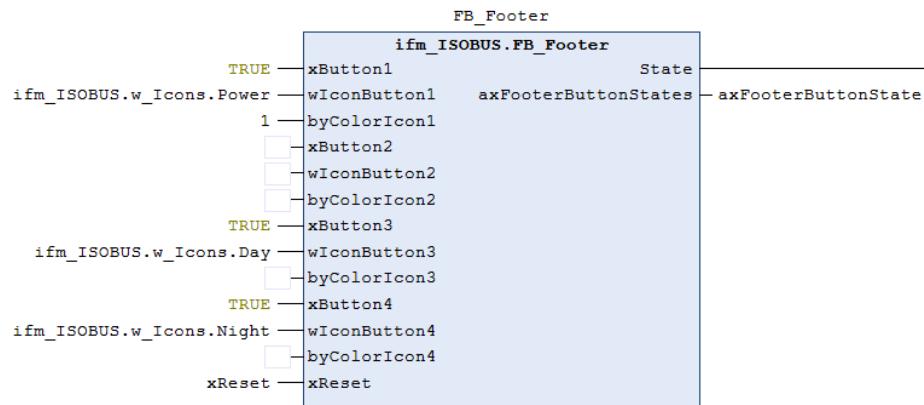
Visualisation example:
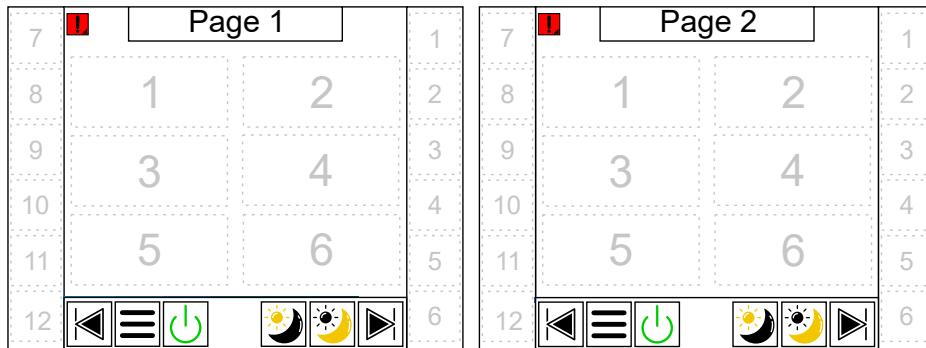
8.2.5 Defining operating elements

The function module `FB_FunctionCreator_Button` defines operating elements for a specific data mask (page). A maximum of 6 of these objects can be placed in a data mask.

The input parameter `wID_IconButton<n>` defines the symbol on the corresponding operating element.

The input parameter `byColorIcon<n>` defines the colour of the softkey symbol:

- 0 = black

- 1 = green

- 2 = yellow

- 3 = red

The output parameter `axButtonState` contains the current states of the two operating elements.

Configuration example in CODESYS:

```
                                          FB_FunctionCreator_Button
                              ifm_ISOBUS.FB_FunctionCreator_Button
                            1 ── byPage_DataMask              State ──────────
                            1 ── bySlotNumber          axButtonStates ── axButtonStates
              'Title Button' ── sFunctionalityTitle
                            2 ── byNumber_Of_Button
 ifm_ISOBUS.w_Icons.CylinderExtend ── wIcon_Button1
                            1 ── byColorIcon1
ifm_ISOBUS.w_Icons.CylinderRetract ── wIcon_Button2
                            3 ── byColorIcon2
                       xReset ── xReset
```
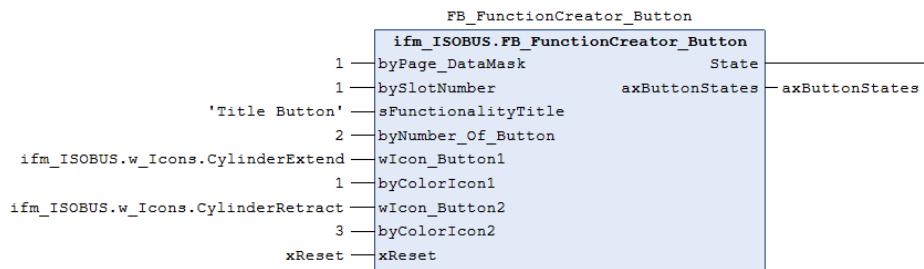
Visualisation example:
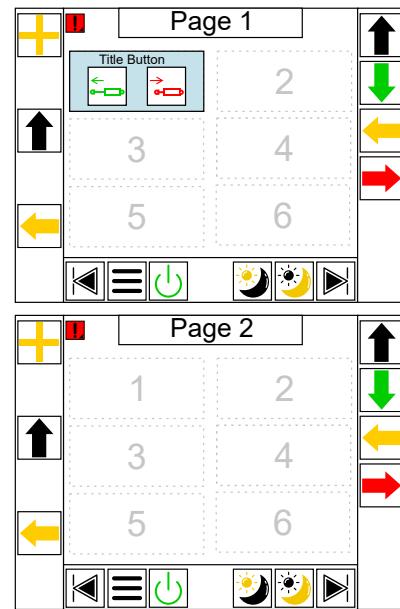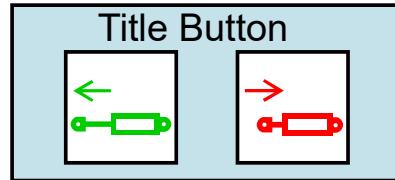

8.2.6   Defining numeric input fields

The function module `FB_FunctionCreator_NumericInput` defines simple numeric input fields for a specific data mask (page). A maximum of 6 of these objects can be placed in a data mask.

The following data can be configured using the input parameters:

*   Required data mask (page)
*   Required slot
*   Title
*   Number of input fields (max. 2)
*   Number of decimal places for the corresponding input field
*   Initial value for the corresponding input field
*   Name of the corresponding input field

The output parameter `adw_InputValues` contains the entered values.

Configuration example in CODESYS:
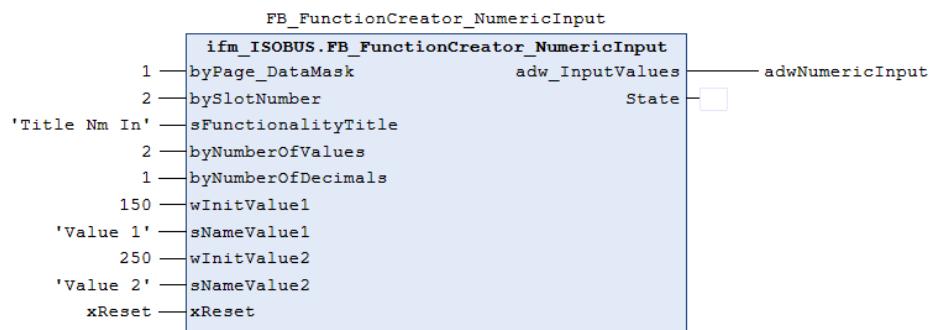
Visualisation example:

Title Nm In
Value 1   15.0
Value 2   25.0

Page 1
Title Button   Title Nm In
Value 1 15.0
Value 2 25.0
3   4
5   6

Page 2
1   2
3   4
5   6
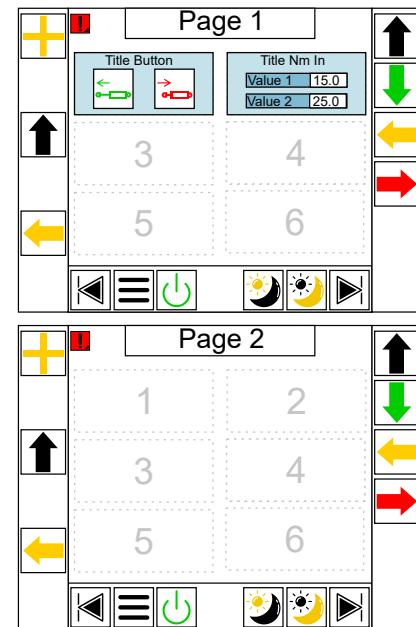
8.2.7 Defining numeric input fields with units

The function module `FB_FunctionCreator_NumericInput_Long` defines numeric input fields with unit specification for a specific data mask (page). A maximum of 3 of these objects can be placed in a data mask (slots 1, 3 and 5). The visualisation element also occupies the adjacent slot on the right.

The following data can be configured using the input parameters:

- Required data mask (page)
- Required slots
- Title
- Number of input fields (max. 2)
- Number of decimal places for the corresponding input field
- Initial value for the corresponding input field
- Name of the corresponding input field
- Unit for the corresponding input field

The output parameter `adw_InputValues` contains the entered values.

Configuration example in CODESYS:

```
                        FB_FunctionCreator_NumericInput_Long
              ifm_ISOBUS.FB_FunctionCreator_NumericInput_Long
        1 — byPage_DataMask                 adw_InputValues — adNumericInputLong
        3 — bySlotNumber                              State —
'Title Nm In Long' — sFunctionalityTitle
        2 — byNumberOfValues
        3 — byNumberOfDecimals
      310 — wInitValue1
'Value 1' — sNameValue1
 'Unit 1' — sUnit1
      320 — wInitValue2
'Value 2' — sNameValue2
 'Unit 2' — sUnit2
   xReset — xReset
```
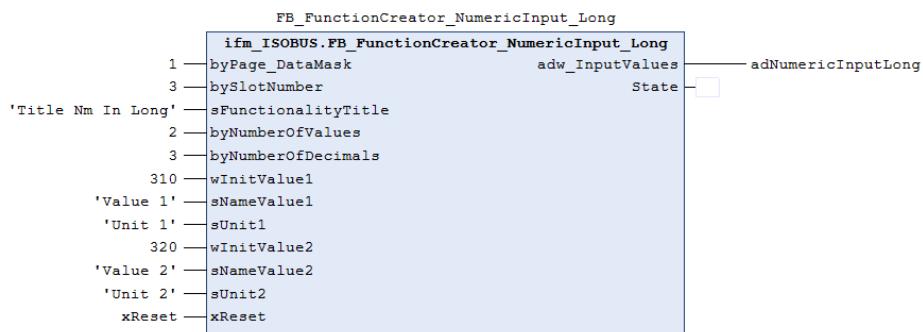
Visualisation example:
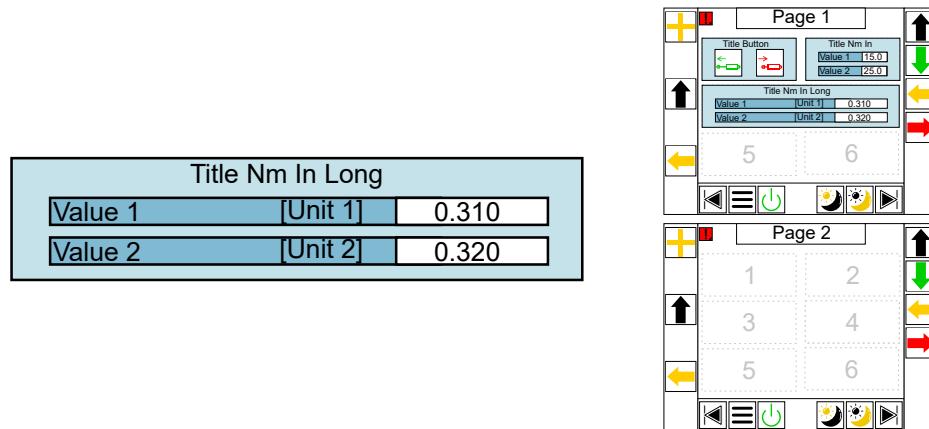
8.2.8 Defining the output of numerical values

The function module `FB_FunctionCreator_NumericOutput` defines the display of simple numerical values for a specific data mask (page). A maximum of 6 of these objects can be placed in a data mask.

The following data can be configured using the input parameters:

- Required data mask (page)
- required slot
- Title
- Number of output fields (max. 2)
- Number of decimal places for the corresponding output field
- Current value for the corresponding output field
- Name of the corresponding output field

Configuration example in CODESYS:
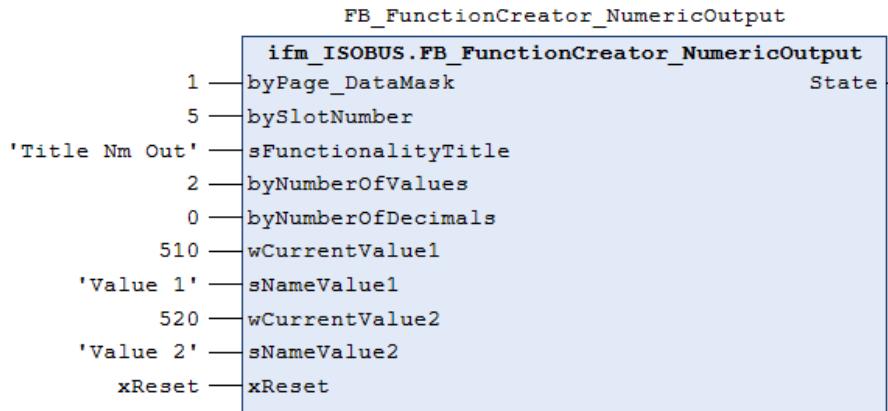
Visualisation example:
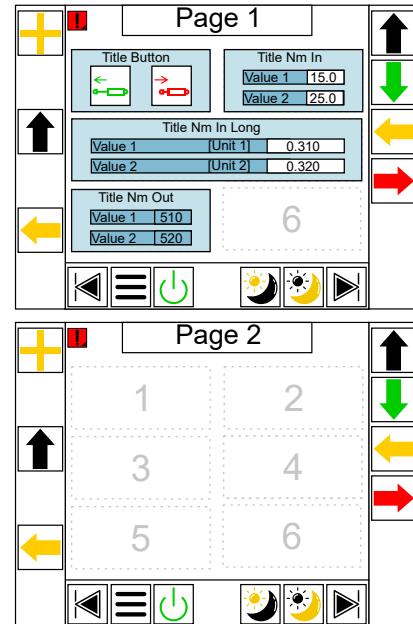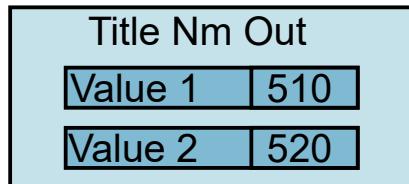
8.2.9 Defining the output of numerical values with units

The function module `FB_FunctionCreator_NumericOutput` defines the display of simple numeric values with indication of the unit for a specific data mask (page). A maximum of 3 of these objects can be placed in a data mask (slots 1, 3 and 5). The visualisation element also occupies the adjacent slot on the right.

The following data can be configured using the input parameters:

• Required data mask (page)

• required slot

• Title

• Number of output fields (max. 2)

• Number of decimal places for the corresponding output field

• Current value for the corresponding output field

• Name of the corresponding output field

• Unit for the corresponding output field

**Configuration example in CODESYS:**

Visualisation example:

Title Nm Out Long		
Value 1	[Unit 1]	21.10
Value 2	[Unit 2]	21.20
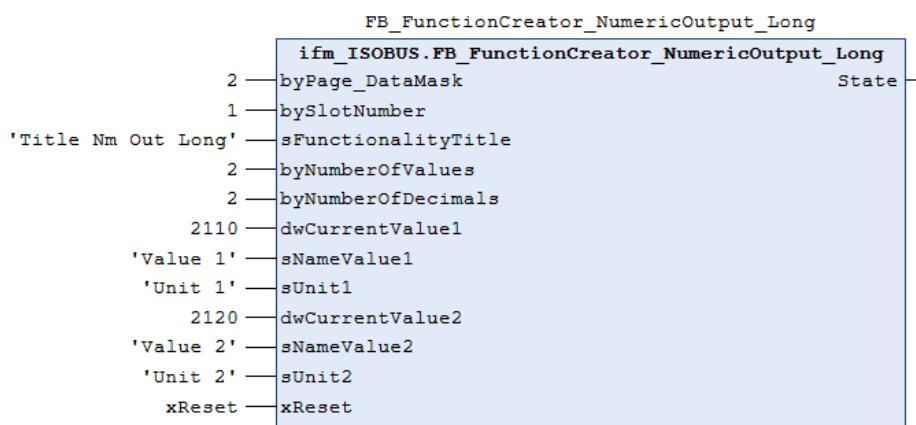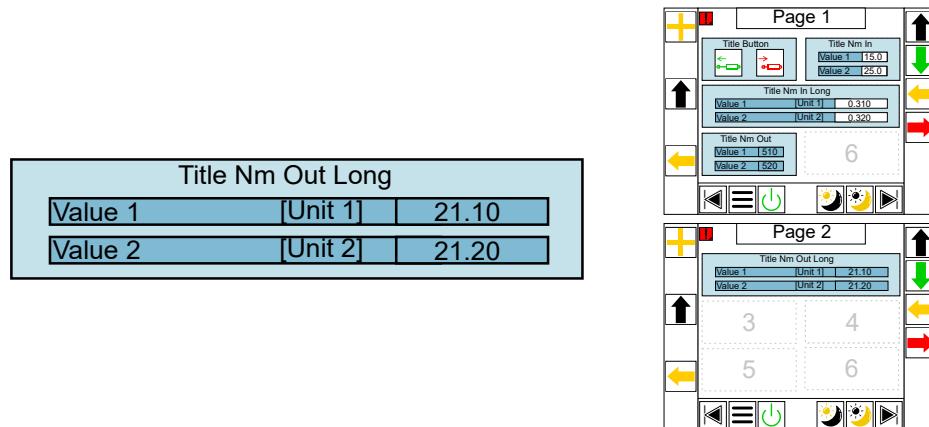
8.2.10 Defining the curved bar chart

The function module `FB_FunctionCreator_ArchedBargraph` defines the display of values in a curved bar chart for a specific data mask (page). A maximum of 6 of these objects can be placed in a data mask.

The following data can be configured using the input parameters:

* Required data mask (page)

* required slot

* Title

* Unit for the displayed value

* Current value

* Number of decimal places of the current value

* Target value

* Minimum value of the display

* Maximum value of the display

* Colour combination of the bar

* Line colour for displaying the target value

Configuration example in CODESYS:

```
                            FB_FunctionCreator_ArchedBargraph
                    ┌──────────────────────────────────────────────┐
                    │   ifm_ISOBUS.FB_FunctionCreator_ArchedBargraph │
            2 ──────┤ byPage_DataMask                          State ├──
            3 ──────┤ bySlotNumber                                   │
 'Title Arched' ────┤ sFunctionalityTitle                            │
        'Unit' ─────┤ sUnit                                          │
           63 ──────┤ wCurrentValue                                  │
            0 ──────┤ byNumberOfDecimals                             │
           63 ──────┤ dwTargetValue                                  │
            0 ──────┤ dwMinValue                                     │
          100 ──────┤ dwMaxValue                                     │
          153 ──────┤ dwGraphColor                                   │
           12 ──────┤ dwColorTargetLine                              │
        xReset ─────┤ xReset                                         │
                    └──────────────────────────────────────────────┘
```

**Visualisation
example:**
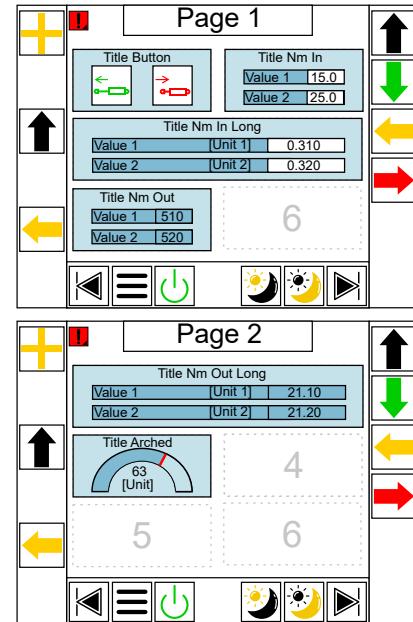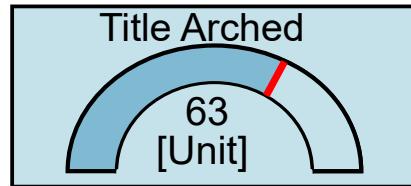
8.2.11   Defining the linear bar chart

The function module `FB_FunctionCreator_LinearBargraph` defines the display of values in a linear bar chart for a specific data mask (page). A maximum of 6 of these objects can be placed in a data mask.

The following data can be configured using the input parameters:

• Required data mask (page)

• required slot

• Title

• Unit for the displayed value

• Current value

• Number of decimal places of the current value

• Target value

• Minimum value of the display

• Maximum value of the display

• Colour combination of the bar

• Line colour for displaying the target value

**Configuration example
in CODESYS:**

```
                                          FB_FunctionCreator_LinearBargraph
                                  ┌─────────────────────────────────────────────┐
                                  │  ifm_ISOBUS.FB_FunctionCreator_LinearBargraph │
                            2 ────┤ byPage_DataMask                        State ├──
                            4 ────┤ bySlotNumber                                 │
               'Title Linear' ────┤ sFunctionalityTitle                          │
                       'Unit' ────┤ sUnit                                        │
                          300 ────┤ wCurrentValue                                │
                            0 ────┤ byNumberOfDecimals                           │
                          400 ────┤ dwTargetValue                                │
                            0 ────┤ dwMinValue                                   │
                          500 ────┤ dwMaxValue                                   │
                          153 ────┤ dwGraphColor                                 │
                           12 ────┤ dwColorTargetLine                            │
                       xReset ────┤ xReset                                       │
                                  └─────────────────────────────────────────────┘
```
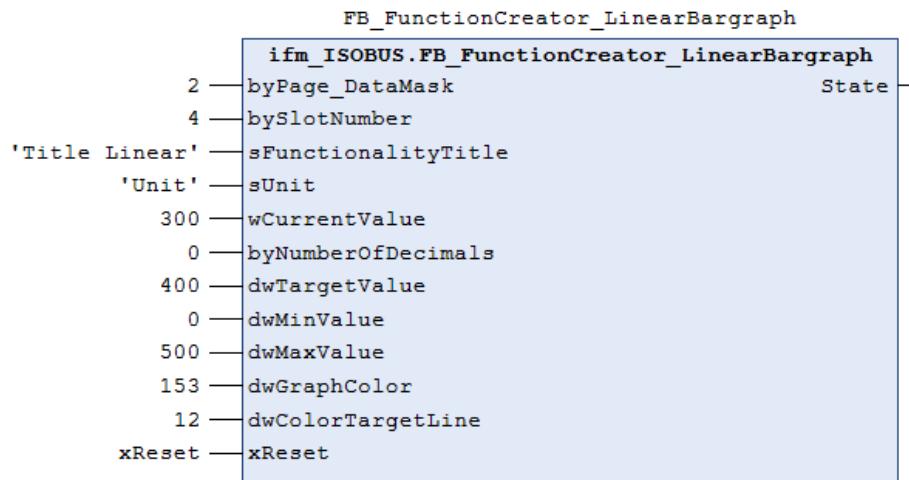
**Visualisation
example:**
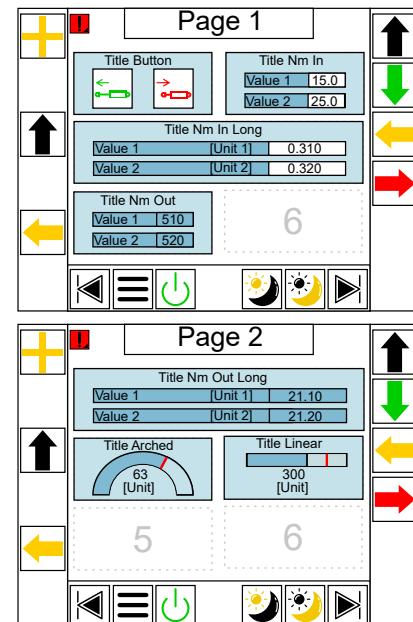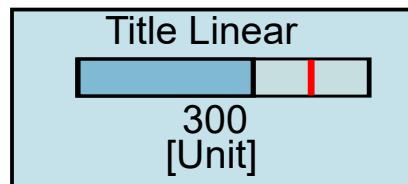
8.2.12   Defining the pointer object

The function module `FB_FunctionCreator_Meter` defines the representation of values with a pointer for a specific data mask (page). A maximum of 6 of these objects can be placed in a data mask.

The following data can be configured using the input parameters:

*   Required data mask (page)

*   required slot

*   Title

*   Unit for the displayed value

*   Current value

*   Offset (for oscillating around a zero position, for example to visualise a slope)

*   Number of decimal places of the current value

*   Minimum value of the display

*   Maximum value of the display

*   Number of graduation marks in the scale

*   Scale colour

- Pointer colour

**Configuration example
in CODESYS:**

```
                              FB_FunctionCreator_Meter
                        ┌─────────────────────────────────────────┐
                        │  ifm_ISOBUS.FB_FunctionCreator_Meter      │
              2 ────────┤ byPage_DataMask                    State ├──
              5 ────────┤ bySlotNumber                              │
  'Title Meter' ────────┤ sFunctionalityTitle                       │
       'Unit' ──────────┤ sUnit                                     │
            237 ────────┤ wCurrentValue                             │
              0 ────────┤ wOffset                                   │
              1 ────────┤ byNumberOfDecimals                        │
              0 ────────┤ dwMinValue                                │
            600 ────────┤ dwMaxValue                                │
              5 ────────┤ dwNumberOfTicks                           │
              0 ────────┤ dwArcAndTickColor                         │
             12 ────────┤ dwNeedleColor                             │
         xReset ────────┤ xReset                                    │
                        └─────────────────────────────────────────┘
```
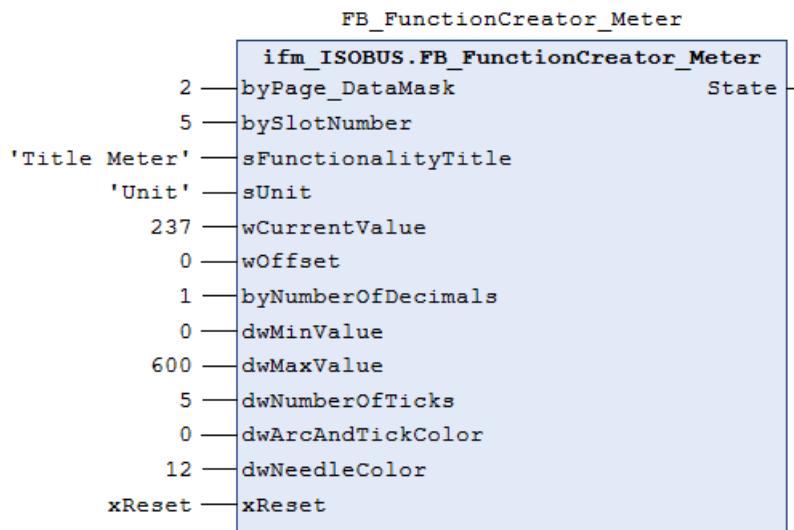
**Visualisation
example:**
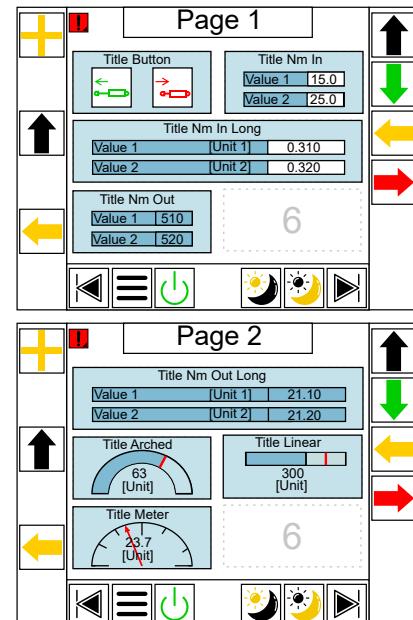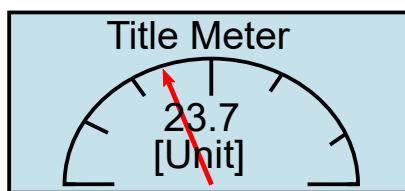
8.2.13 Managing colour settings

The function module `FB_SetColors` defines the colour representation of all data masks (pages).

For example, background, line and text colours can be configured via the input parameters.

**Configuration example
in CODESYS:**

```
                              FB_SetColors
                        ┌─────────────────────────────────────────┐
                        │      ifm_ISOBUS.FB_SetColors              │
      White ────────────┤ dwBackgroundcolor                         │
     Orange ────────────┤ dwContainerColor                          │
     Orange ────────────┤ dwNumericBackgroundColor                  │
      White ────────────┤ dwNumericInputBackgroundColor             │
      Black ────────────┤ dwLineColor                               │
      Black ────────────┤ dwTextColor                               │
                        └─────────────────────────────────────────┘
```
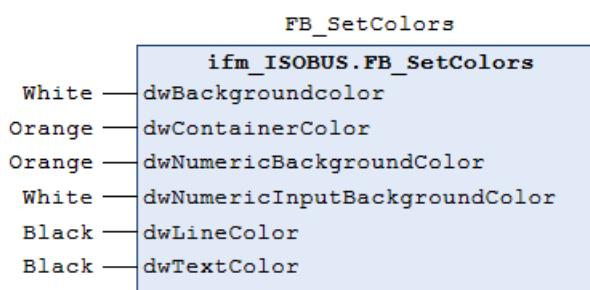
Visualisation example:
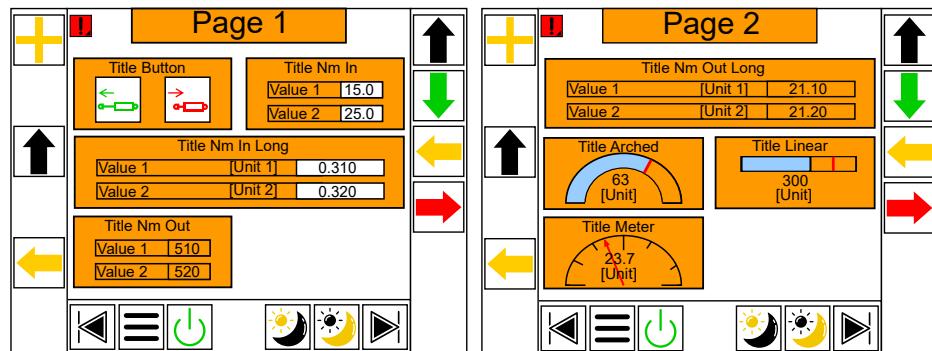
9   Maintenance, repair and disposal

Cleaning the unit:

▶ Disconnect the unit from the voltage supply.

▶ Clean the unit from dirt using a soft, chemically untreated and dry micro-fibre cloth.

The operation of the unit is maintenance-free.

Only the manufacturer is allowed to repair the unit.

▶ After use dispose of the device in an environmentally friendly way in accordance with the applicable national regulations.