SIEMENS

DIGSI CFC

V4.87

Manual

E50417-H1176-C098-B2

NOTE

For your own safety, please observe the warnings and safety instructions contained in this document.

Disclaimer of Liability

This document has been subjected to rigorous technical review before being published. It is revised at regular intervals, and any modifications and amendments are included in the subsequent issues. The content of this document has been compiled for information purposes only. Although Siemens AG has made best efforts to keep the document as precise and up-to-date as possible, Siemens AG shall not assume any liability for defects and damage which result through use of the information contained herein.

This content does not form part of a contract or of business relations; nor does it change these. All obligations of Siemens AG are stated in the relevant contractual agreements.

Siemens AG reserves the right to revise this document from time to time.

Document Release: E50417-H1176-C098-B2.00
Edition: 10.2012
Product version: V4.87

Copyright

Registered Trademarks

SIMATIC®, SIMATIC NET®, SIPROTEC®, DIGSI®, SICAM®, SIMEAS®, SINAUT® , OSCOP®, and DAKON® are registered trademarks of SIEMENS AG. An unauthorized use is illegal.

All other designations in this document can be trademarks whose use by third parties for their own purposes can infringe the rights of the owner.

Preface

Purpose of this manual

This manual provides:

- Information on using DIGSI CFC
- Implementation examples
- Details on the supplied DIGSI CFC blocks

Target group

This manual is intended for persons who commission, program and use DIGSI.
A basic knowledge of DIGSI is required.

Scope of validity of the manual

This manual is valid for DIGSI version 4.87 and higher.

Standards

DIGSI was developed in compliance with the ISO 9001:2008 standard.

Further support

If you have any further questions regarding DIGSI, please do not hesitate to contact your local Siemens representative.

Hotline

Our Customer Support Center provides around-the-clock support.

Phone: +49 (180) 524-8437
Fax: +49 (180) 524-2471
e-mail: support.ic@siemens.com

Training courses

If you are interested in our current training program, please contact our training center:

Siemens AG
Siemens Power Academy

Humboldtstr. 59
90459 Nuremberg
Germany

Phone: +49 (911) 433-7415
Fax: +49 (911) 433-5482
e-mail: poweracademy.ic-sg@siemens.com
Internet: www.siemens.com/energy/power-academy

Notes On Safety

This manual does not constitute a complete catalog of all safety measures required for operating the equipment (module, device) in question, because special operating conditions may require additional measures. However, it does contain notes that must be adhered to for your own personal safety and to avoid damage to property. These notes are highlighted with a warning triangle and different keywords indicating different degrees of danger.

DANGER

Danger means that death or severe injury **will** occur if the appropriate safety measures are not taken.

✧ Follow all advice instructions to prevent death or severe injury.

WARNING

Warning means that death or severe injury **can** occur if the appropriate safety measures are not taken.

✧ Follow all advice instructions to prevent death or severe injury.

CAUTION

Caution means that minor or moderate injury can occur if the appropriate safety measures are not taken.

✧ Follow all advice instructions to prevent minor injury.

NOTICE

Notice means that damage to property can occur if the appropriate safety measures are not taken.

✧ Follow all advice instructions to prevent damage to property.

NOTE

is important information about the product, the handling of the product, or the part of the documentation in question to which special attention must be paid.

Qualified Personnel

Commissioning and operation of the equipment (module, device) described in this manual must be performed by qualified personnel only. As used in the safety notes contained in this manual, qualified personnel are those persons who are authorized to commission, release, ground and tag devices, systems, and electrical circuits in accordance with safety standards.

Use as Prescribed

The equipment (device, module) must not be used for any other purposes than those described in the Catalog and the Technical Description. If it is used together with third-party devices and components, these must be recommended or approved by Siemens.

Correct and safe operation of the product requires adequate transportation, storage, installation, and mounting as well as appropriate use and maintenance.

During the operation of electrical equipment, it is unavoidable that certain parts of this equipment will carry dangerous voltages. Severe injury or damage to property can occur if the appropriate measures are not taken:

- Before making any connections at all, ground the equipment at the PE terminal.
- Hazardous voltages can be present on all switching components connected to the power supply.
- Even after the supply voltage has been disconnected, hazardous voltages can still be present in the equipment (capacitor storage).
- Equipment with current transformer circuits must not be operated while open.
- The limit values indicated in the manual or the operating instructions must not be exceeded; this also refers to testing and commissioning

Contents

1 Product Overview

DIGSI CFC

DIGSI CFC (Continuous Function Chart) is a component of DIGSI 4 for programming SIPROTEC devices.

The graphic user interface is used to connect items of information and program, for example, interlocks and switching sequences. In addition, you can edit measured values and generate messages.

Information

Types of information often used are:

- The position of the switching devices and other process elements,
- Measured values,
- Binary information on the state of the bay and the devices,
- Protective information,
- General messages and
- Interrupts.

You assign the required information to the DIGSI CFC in the configuration matrix of DIGSI 4.

Interconnecting items of information

Items of information are interconnected in DIGSI CFC by means of the supplied CFC blocks:

- Arithmetic,
- Basic logic,
- Information status,
- Memory,
- Control commands,
- Type converters,
- Comparison,
- Metered value and
- Time & Clock

Working with DIGSI CFC

In DIGSI CFC you draw the technological process of the program in the form of a CFC chart.

In the CFC chart you interconnect the information prepared in the DIGSI 4 configuration matrix by means of CFC blocks.

The completed CFC chart is compiled with DIGSI CFC into an executable program.

2 Getting Started

Overview

In this chapter, you are familiarised with DIGSI CFC and learn how to use it.

Note

The following instructions base on each other and each contain important information on working with DIGSI CFC. Work through the individual steps consecutively. Observe the general information on each example.

Note

The CFC blocks in this chapter are displayed with the option **Block width: Wide**:

- ✧ In the CFC Editor, click **Options > Customize > Block/Sheet Bar settings**.
- ✧ Activate the option **Wide** in the displayed window under **Block width** and confirm with **OK**.

Contents

2.1 Programming SIPROTEC Devices

You can program your own automation functions in SIPROTEC devices. Use DIGSI CFC for programming.

Selecting information items

In the configuration matrix of DIGSI select the items of information which you want to use to program the respective function.

DIGSI CFC

In DIGSI CFC you draw the technological sequence of the function in the form of a CFC chart.

Priority class

Set a priority class for each CFC chart depending on the necessary processing priority (e.g. protective function) and processing time (e.g. cyclical).

CFC chart

In the CFC chart you interconnect the items of information by means of CFC blocks.

Parameter set

After you have compiled the CFC charts and terminated DIGSI CFC, you can download the created program to the SIPROTEC device with the parameter set.

Note

The subsequent action instructions are usually described on the basis of the menu commands.

Many of the menu commands can be called up by using icons on the toolbar.
Use **View > Toolbar** to hide or show the toolbar.

In addition keyboard shortcuts are available for many functions. Take note of the information given behind the individual menu commands.

Useful information can also be found in the status bar which you can hide or display by using **View > Status Bar**:

- Information on the operating action currently being carried out,
- The state of the lockable keys,
- Number of the current sheet in the CFC chart and
- Currently active priority class (for example **PLC_BEARB [Fast PLC]**).

2.2 Priority Classes

Every function which you program with DIGSI CFC has to be assigned to a priority class. The individual priority classes differ in their processing priority and execution time:

- Fast PLC processing (**PLC_BEARB** / **PLC**)
- Slow PLC processing (**PLC1_BEARB** / **PLC1**)
- Measured value processing (**MW_BEARB**/ **MEASURE**)
- Interlocking (**SFS_BEARB** / **INTERLOCK**)

Processing priority

The processing priority also specifies the type of functions which you can assign to a specific priority class:

Table 2-1 Priority classes, processing priority and assigned functions in DIGSI CFC

Priority class	Processing priority	Assigned functions
Fast PLC processing (**PLC_BEARB**)	Functions in this priority class are processed event-controlled with the highest priority: Every change to a logical input signal is processed immediately. The processing of a function in this priority class can interrupt the processing of a function in the priority class Slow PLC processing (PLC1_BEARB).	Protective functions, blocking of protective functions **Note:** In this priority class, you can interconnect less blocks than in the priority class Slow PLC processing (PLC1_BEARB). Observe the technical data in the device manual of the SIPROTEC device which you want to use.
Slow PLC processing (**PLC1_BEARB**)	Functions in this priority class are event-controlled with a lower priority than the functions in the priority class Fast PLC processing (PLC_BEARB): Every change to a logical input signal is processed immediately. However, the processing can be interrupted by the processing of a function in the priority class Fast PLC processing (**PLC_BEARB**).	Event-controlled functions: e.g. applications with time and clock functions, operating of function keys **Note:** Use this priority class preferably for logic functions which are not protective functions.
Measured value processing (**MW_BEARB**)	Functions in this priority class are processed cyclically in the background. **Note:** The functions are **not** processed event-controlled.	Processing measured values: e.g. calculating power factor (ANSI 55), calculating apparent power (ANSI 32)
Interlocking (**SFS_BEARB**)	Functions in this priority class are activated by a control command and also processed cyclically in the background. In case of a response of a protective device slightly less often.	Interlocking: e.g. locking

Note

Some CFC blocks (e.g. TIMER) can only be used in the specified priority classes. Please refer to the corresponding information in chapter 4 of this manual.

Note

Several CFC charts can be assigned to a priority class.
The system firmware of the SIPROTEC devices always processes a chart completely before another chart of the same priority class is started.

Note

For the interconnection of **fast signals** with the left border of the CFC chart in the priority classes measured value processing (priority class **MW_BEARB**) and interlocking (priority class **SFS_BEARB**), the following applies:

Boolean inputs are **no** triggers for this level and remain **unnoticed** when the event or signal is shorter than the processing cycle of the processing level.

For the interconnection of **measured values** with the left border of the CFC in the task levels **fast PLC processing** (task level **PLC_BEARB**) and **slow PLC processing** (task level **PLC1_BEARB**), the following applies:

Measuring inputs are **no** triggers for this level and remain **unnoticed**.

Note

The order of processing for charts of a level during cyclical triggering is random and cannot be predicted.

2.3 Programming Guidelines

When programming functions with DIGSI CFC, you must observe a set of guidelines in order to avoid problems with program processing.

2.3.1 CFC Standard Formula

Work through the following eight steps consecutively when programming functions with DIGSI CFC:

1. Configure to CFC.
2. Save the configuration matrix.
3. Insert a CFC chart.
4. Specify the priority class.
5. Draw the chart.
6. Check the run sequence.
7. Compile the chart.
8. Save the parameter set.

2.3.2 Splitting and Joining Information Items

Description

Using DIGSI configuring, you can split information items into several indications. If you join these indications later on in a CFC chart or CFC block, malfunctions in the program processing may occur (e.g. a CFC block can be initiated even though it is not fully initialised).

Note

Ensure that you do **not** split an information item into several indications in DIGSI configuring **and** later join them in a CFC chart or CFC block.

Solution

If you split an information item into several indications for function programming and then later on need to join them in a CFC chart or CFC block, divert the required indications with suitable CFC blocks in the CFC chart.

2.3.3 Maximum Permissible Number of Blocks

Description

The maximum permissible number of blocks in the individual priority classes of the CFC charts of a SIPROTEC device depends on the computational capacity of the devices and is monitored via the CFC compiler. Observe the technical data in the device manual of the SIPROTEC device which you want to use.

Note

The maximum permissible number of **MEMORY**, **RS_FF_MEMO**, **SR_FF_MEMO**, **D_FF_MEMO** and **COUNTER** blocks depends on the available non-volatile memory and is monitored by the CFC compiler.

The maximum permissible number is checked during the compilation of the CFC chart. Consistency errors are indicated when a fault occurs. The exceeding of the resource is indicated in the displayed compilation log.

Note

The following applies for SIPROTEC devices with a device version **less than** V4.5:

The maximum permissible number of **CMD_CHAIN** blocks in the priority classes
Fast PLC processing (priority class **PLC_BEARB**) and
Slow PLC processing (priority class **PLC1_BEARB**)
is **20** blocks.

Note

The maximum permissible number of **TIMER** and **TIMER_SHORT** blocks is limited by the available system timers and is monitored by the CFC compiler.

The maximum permissible number is checked during the compilation of the CFC chart. Consistency errors are indicated when a fault occurs. The exceeding of the resource is indicated in the displayed compilation log.

2.4 Selecting Items of Information for CFC

Selecting items of information

Mark items of information which you want to use in CFC as follows:

– Open the configuration matrix in DIGSI:

✧ Select the **Settings** folder in the navigation pane and

✧ Double-click on the **Masking I/O (Configuration Matrix)** function in the data pane.

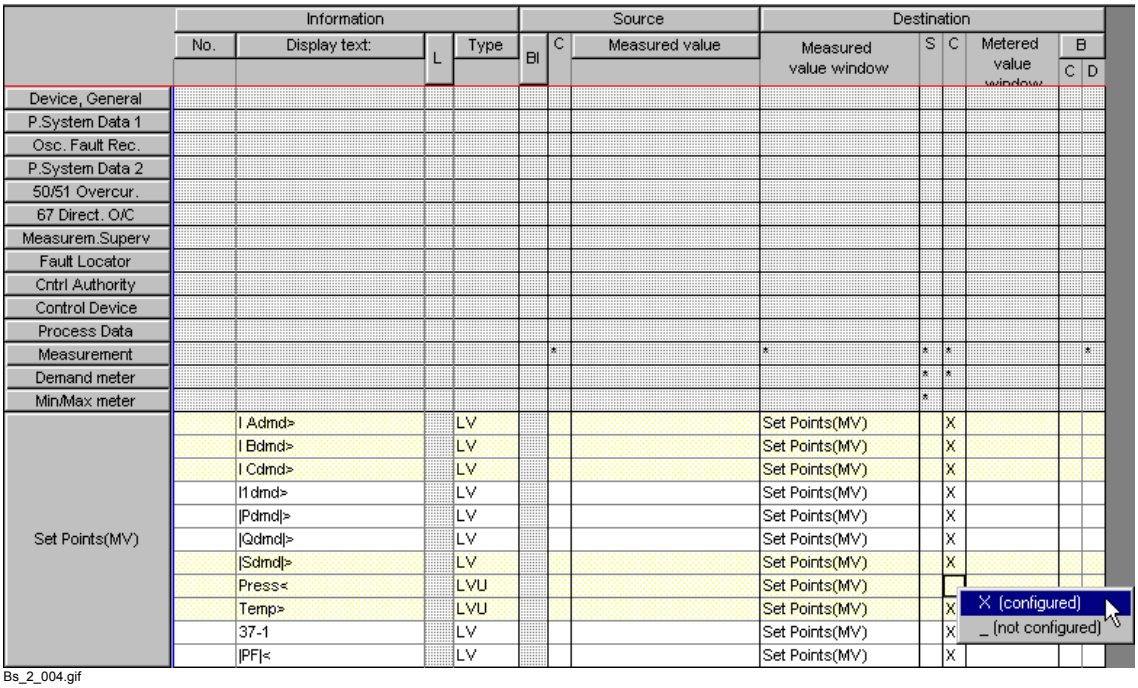
– Search for the items of information which you want to select in the opened configuration matrix or add required information here.
In the process, orientate yourself under **Information** to the columns **Display text**, **Long text** and **Type**.

Note

Note the settings of the filter in the toolbar of the configuration matrix
(e.g. **Measured and Metered Values Only** in conjunction with **No Filter**).

– In order to use an information as an input signal for the CFC on the left border, configure the information to CFC under **Destination**:

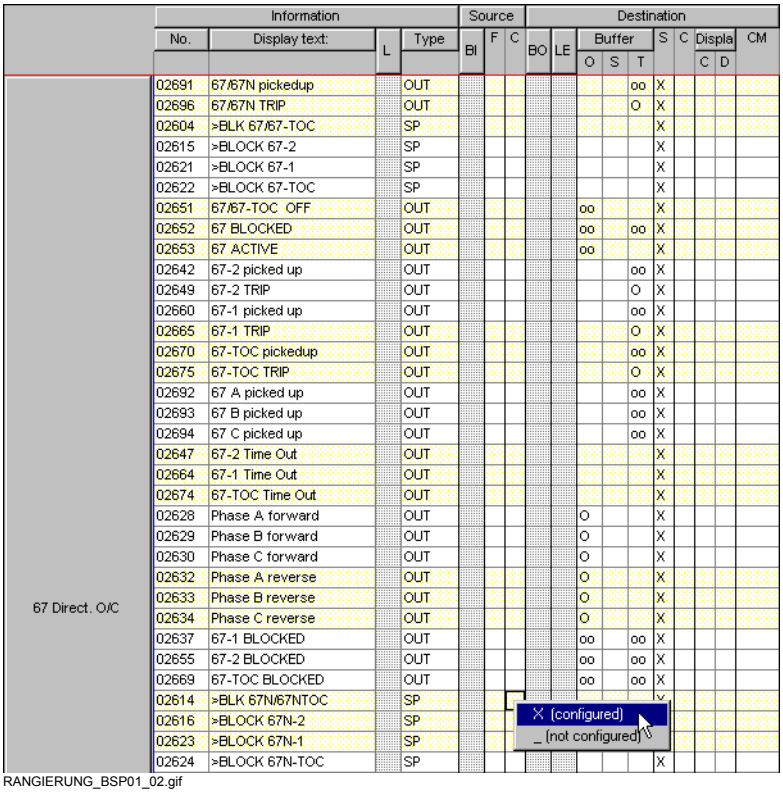
✧ Right-click in the corresponding cell of the column **C** of the configuration matrix and select **X (configured)** in the opened context menu.



Bs_2_004.gif

Fig. 2-1 Configuring an information as an input signal to the CFC

- ✧ The information configured to the CFC is indicated in the column **C** by an **X**.
- ✧ In order to use an information as an output signal for the CFC on the right border, configure the information to CFC under **Source**:
- ✧ Right-click in the corresponding cell of the column **C** of the configuration matrix and select **X (configured)** in the opened context menu.



RANGIERUNG_BSP01_02.gif

Fig. 2-2 Configuring an information as an output signal to the CFC

The information configured to the CFC is indicated in the column **C** by an **X**.

Note

If measured values are configured as input signals for the CFC on the left border and are interconnected in one of the priority classes PLC_BEARB [Fast PLC] and PLC1_BEARB [Slow PLC] in CFC charts, a value change in the measured values does not automatically cause these charts to be processed.

Measured values must be processed in the priority class MW_BEARB.

Note

Information configured as output signals for the CFC on the right border are only logged in the result logs of the respective device for a value change of the interconnected signal in the CFC chart.

Note

Transient information is not suitable as an input signal for logic operations in the CFC.

A transient information can be used to trigger the processing of charts in the event-oriented priority classes PLC_BEARB [Fast PLC] and PLC1_BEARB [Slow PLC]. The charts are then processed whenever the indication occurs.

2.5 Example of Fast PLC Processing

Fast PLC processing (priority class **PLC_BEARB**) is carried out with the highest priority. Processing is carried out event-controlled due to changes to the input signals.

Note

The functional scope of a SIPROTEC device with the MLFB **7SJ63655ER633HH3** is selected in the following example. If you want to understand the example, you must have inserted a comparable device from the device catalogue in DIGSI.

In the **functional scope** of the SIPROTEC device, the functions **67, 67-TOC** and the **67N, 67N-TOC** must be activated.

Task

Direction-measuring functions are to be blocked due to an error in the input voltage circuit.

Input signal

- Group: **Measurem. superv**
  Display text: **VT Fuse Failure**

Output signals

- Group **67 Direct. O/C**
  Display text **>BLK 67/67-TOC**
- Group **67 Direct. O/C**
  Display text **>BLK 67N/67NTOC**

CFC block

- **CONNECT** (connection)

How to proceed

- ✧ In the DIGSI configuration matrix, configure the input signals and the output signals to the CFC see chapter 2.4).
- ✧ Create a new CFC chart with the designation **BLK DIR FOR MCB TRIP**.
- ✧ Specify the priority class **PLC_BEARB [Fast PLC]**.
- ✧ Position the block **CONNECT**.
- ✧ Interconnect the input signal.
- ✧ Interconnect the output signals.
- ✧ Compile the CFC chart.

2.5.1 Creating a New CFC Chart

In order to create a new chart with the designation **BLK DIR FOR MCB TRIP**:

– Switch to the CFC folder:

✧ Select the **Settings** folder in the navigation pane of DIGSI and

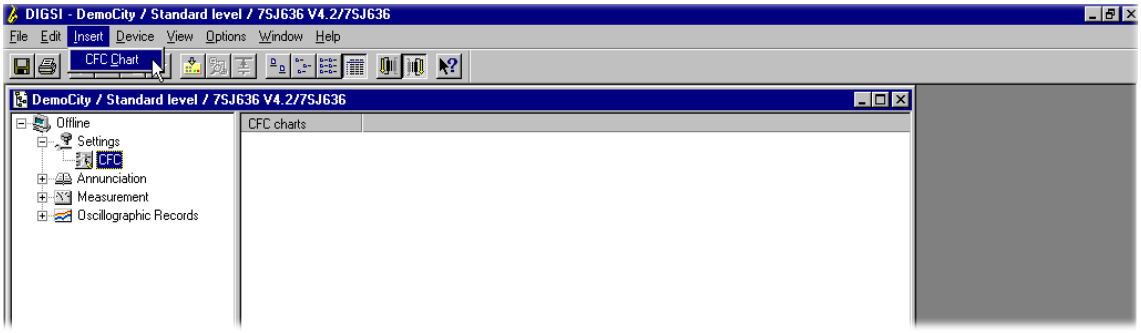
✧ Double-click on the **CFC** entry in the data pane.

All the existing charts are displayed in the data pane.

Note

Output signals displayed on the right border of DIGSI CFC can not be interconnected again on a right border, in order to ensure consistency (see chapter 2.5.5).

– Click on **Insert > CFC Chart**.



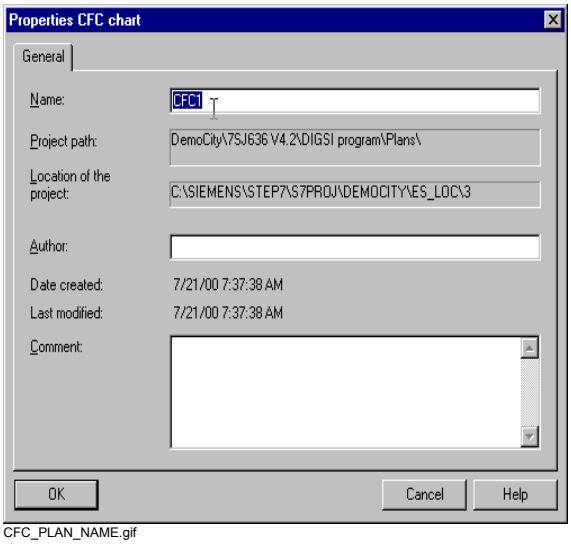
CFC_PLAN_EINFUEGENa.gif

Fig. 2-3 Inserting a CFC chart

– Rename the new CFC chart:

✧ Right-click on the automatically assigned name of the CFC chart and select **Object Properties...** in the displayed context menu.

✧ Enter the new designation **BLK DIR FOR MCB TRIP** under **Name** in the displayed **Properties CFC** dialog box.



CFC_PLAN_NAME.gif

Fig. 2-4 Renaming a CFC chart

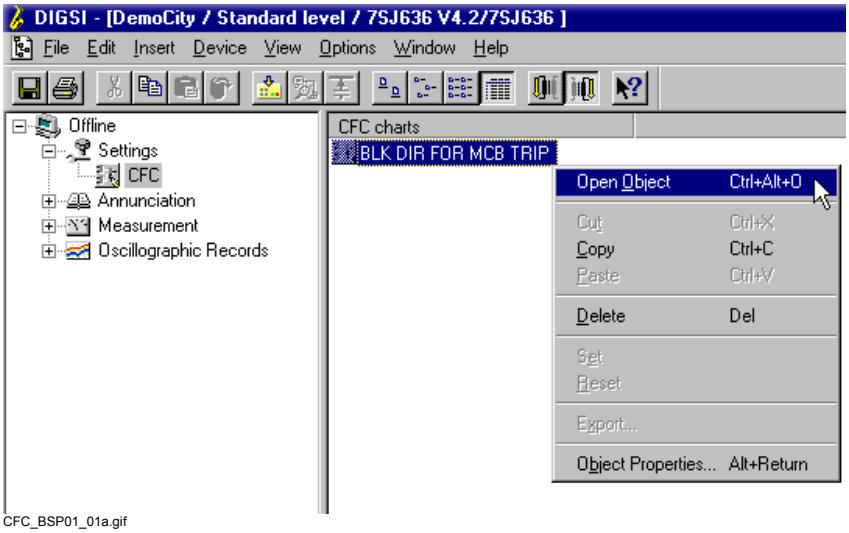
Note

The **Author** and **Comment** input fields can be used for documentation purposes: You can enter an electronic modification history under Comment, for example.

✧ Accept the new chart name by clicking on **OK**.

– Open the CFC chart:

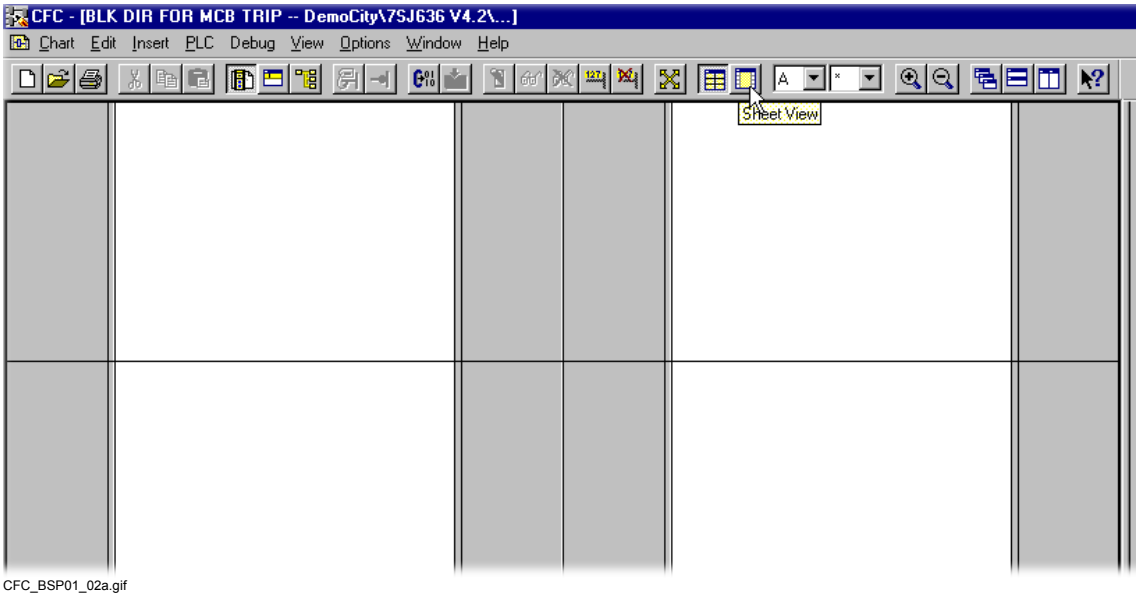
✧ Right-click on the name of the CFC chart and select **Open Object...** in the displayed context menu.



CFC_BSP01_01a.gif

Fig. 2-5 Opening a CFC chart

✧ If you see the chart in the overview (six sheets), switch the presentation to Sheet View. To do so, click on the corresponding icon in the toolbar.



CFC_BSP01_02a.gif

Fig. 2-6 Switching the display over to Sheet View

Note

You can divide the entire CFC chart into partial charts (with six sheets each) in order to make a large CFC chart clearer.

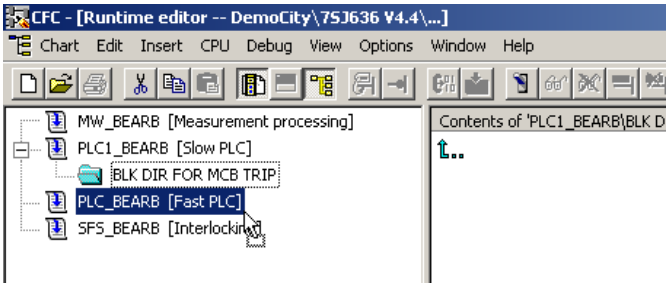
Links between partial charts are possible directly across the borders.

Further information on partial charts can be found in the on-line help on DIGSI CFC.

2.5.2 Defining a Priority Class

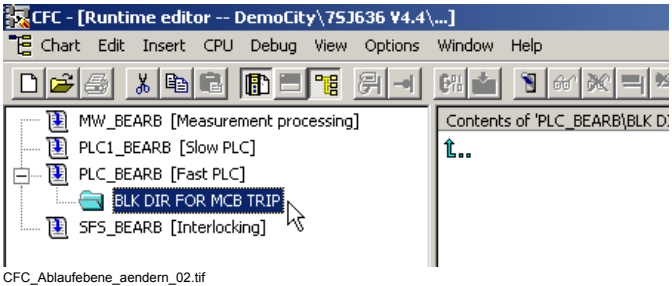
In order to define the priority class **PLC_BEARB** proceed as follows:

- Click on **Edit > Run Sequence...** in the CFC chart to open the **Run-time editor**.
- Navigate to the CFC chart named **BLK RICHT WG AUTOFALL** and copy the CFC chart into the priority class **PLC_BEARB** via drag&drop.



CFC_Ablaufebene_aendern_01.tif

Fig. 2-7 Editing the priority class via drag&drop



CFC_Ablaufebene_aendern_02.tif

Fig. 2-8 CFC chart in new priority class

- Click on **Edit > Run Sequence...** in the Run-time editor to open the CFC chart again.

Note

In order to automatically locate newly added CFC chart into the priority class **PLC_BEARB,** proceed as follows:

✧ Select the class **PLC_BEARB (Fast PLC)** on the navigation pane and click on **Edit > Predecessor for installation**.

✧ Click **OK** to confirm the message box that pops up informing you about the modified priority class.

2.5.3 Positioning a Block

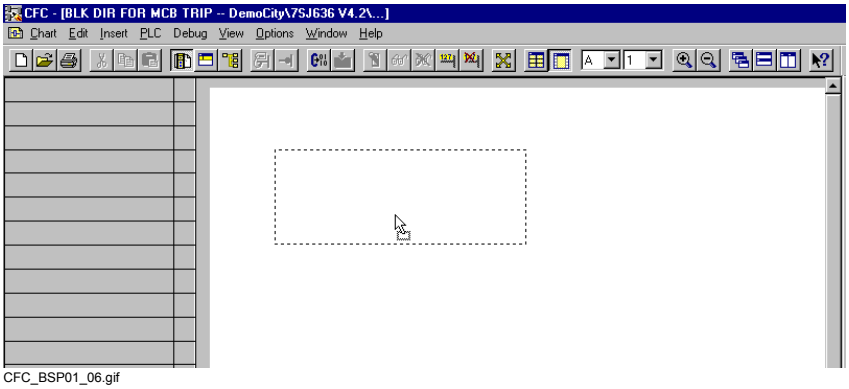
In order to position the **CONNECT** block in the CFC chart:

– Click in the catalogue on the **Blocks** tab.

Note

Further information on the individual tabs in the catalogue can be found in the on-line help of DIGSI CFC.

– Click the **CONNECT** block, hold down the mouse button and drag the block to the sheet of the CFC chart.

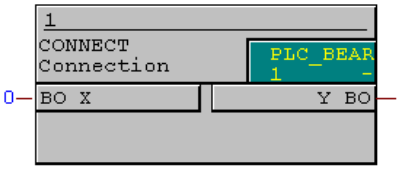


CFC_BSP01_06.gif

Fig. 2-9 Positioning a CFC block via Drag & Drop

– Release the mouse button. The block is displayed on the sheet.

Note



ConnectKopfInfo.tif

Fig. 2-10 Information in the block header

The header of a positioned block displays:

- The block name (for example **1**),
- The block type (for example **CONNECT**),
- The priority class (for example **PLC_BEARB**) and below the run sequence number (for example **1**).

Note

✧ All the blocks of a CFC chart have to lie in the same priority class.

Note

A CFC chart may be made easier to read if you adapt the name of CFC blocks based on their use:

To change a block name:

- ✧ Right-click on the block and select **Object Properties...** in the displayed context menu.
- ✧ Click on the **General** tab and enter the new designation in the **Name** input field.
- ✧ Click on **OK** to confirm the changes.

Note

Interconnected blocks in a CFC chart are processed in a specified sequence. The run sequence is shown by the numbering of the blocks. The numbering has to correspond to the logical sequence.

If you position several blocks in a CFC chart or insert new blocks later, you have to check the priority class or the run sequence and, if necessary, adapt it.

In order to change the priority class or the number in the run sequence of a block:

- ✧ Right-click on the block and select **Go To Installation Position** in the displayed context menu.
  The **Run-time editor** is displayed. The current block is marked in the navigation pane of the Run-time editor.
- ✧ In order to remove the block from the priority class or the run sequence, right-click on the block and select **Cut** in the displayed context menu.
  The block is displayed hidden.
- ✧ In order to insert the block at the new position in the priority class or the run sequence, right-click on the new position and select **Paste** in the displayed context menu.
  The block is inserted at the new position.
- ✧ Click in the Run-time editor on **Edit > Run Sequence...**. The CFC chart is displayed again.

Note

A block which has been positioned or interconnected can be moved within a CFC chart:

- ✧ Click on the block header, keep the mouse button pressed and drag the block to the new position.
- ✧ Release the mouse button. The block with its connections is displayed in the new position.

The run sequence is **not** automatically changed when a block is displaced.

2.5.4 Interconnecting an Input Signal

In order to interconnect the **Measurem. Superv Fuse Failure** input signal with the **CONNECT** block:

– Right-click on the **BO und X** connection as the input of the **CONNECT** block and select **Interconnection to Address...** in the displayed context menu.

CFC_BSP01_07.gif

Fig. 2-11 Interconnecting the input signal

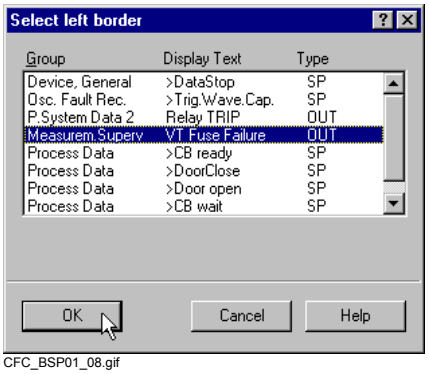
– In the displayed dialog box **Select Left Border** select the information **Measurem. Superv Fuse Failure**. Orientate yourself on the basis of the three columns **Group**, **Display Text** and **Type**, which you already know from the configuration matrix.

Note

The list box of the **Select Left Border** dialog box only displays the information,

- Which is identified in the **Configuration Matrix** of DIGSI as a **Destination** CFC and
- Which can be connected to the connection type of the block.

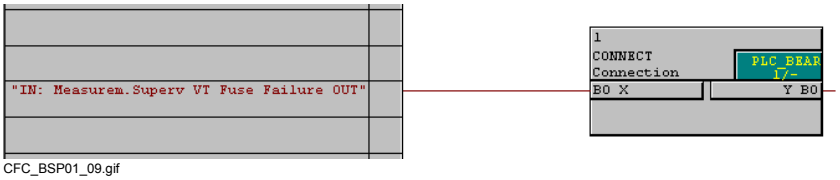
– Click on **OK** to confirm your selection.



CFC_BSP01_08.gif

Fig. 2-12 Select left border dialog box

The new connection is displayed in the CFC chart. The output signal is entered in the left border and connected by a line to the block.



CFC_BSP01_09.gif

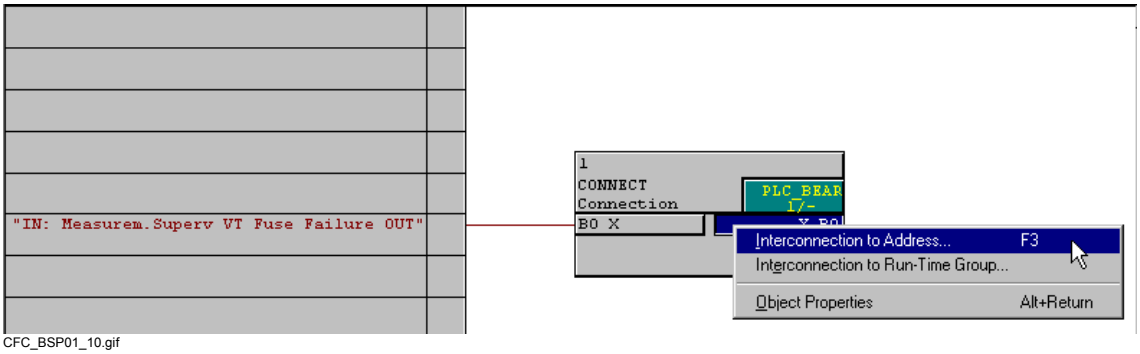
Fig. 2-13 Interconnected input signal fast PLC processing

2.5.5 Interconnecting Output Signals

In order to interconnect the **67 Direct. O/C >BLK 67/67-TOC** and **67 Direct.O/C >BLK 67N/67NTOC** output signals to the **CONNECT** block:

– Connect the information **67 Direct. O/C >BLK 67/67-TOC** to the block:

✧ Right-click on the **Y** connection as the output of the **CONNECT** block and select **Interconnection to Address...** in the displayed context menu.



CFC_BSP01_10.gif

Fig. 2-14 Interconnecting an output signal

Note

Do not under any circumstances use the **Interconnection to Run-Time Group...** function.

The interconnection across charts is not supported by DIGSI CFC.

Note

In order to ensure consistency, output signals may only be used once in all the CFC charts.

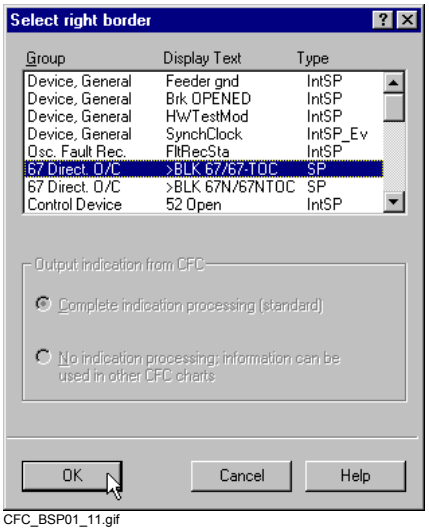
✧ In the displayed dialog box **Select Right Border** select the information **67 Direct. O/C >BLK 67/67-TOC**. Orientate yourself on the basis of the three columns **Group**, **Display Text** and **Type**, which you already know from the configuration matrix.

Note

The list box of the **Select Right Border** dialog box only displays the information,

- Which is identified in the **Configuration Matrix** of DIGSI as a **Source** CFC and
- Which can be connected to the connection type of the block.

✧ Click on **OK** to confirm your selection.



CFC_BSP01_11.gif

Fig. 2-15 Select right border dialog box

The new connection is displayed in the CFC chart. The input signal is entered in the right border and connected by a line to the block.

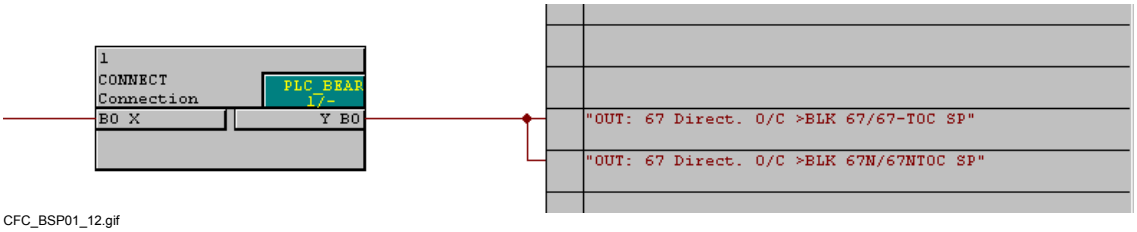
– Connect the information **67 Direct. O/C >BLK67N/67NTOC** to the block:

✧ Right-click on the **BO Y** output of the **CONNECT** block and select **Interconnection to Address...** in the displayed context menu.

✧ In the displayed dialog box **Select Right Border** select the information **67 Direct. O/C >BLK 67N/ 67NTOC**. Orientate yourself on the basis of the three columns **Group**, **Display Text** and **Type**.

✧ Click on **OK** to confirm your selection.

The new connection is displayed in the CFC chart. The input signal is entered in the right border and connected by a line to the block.



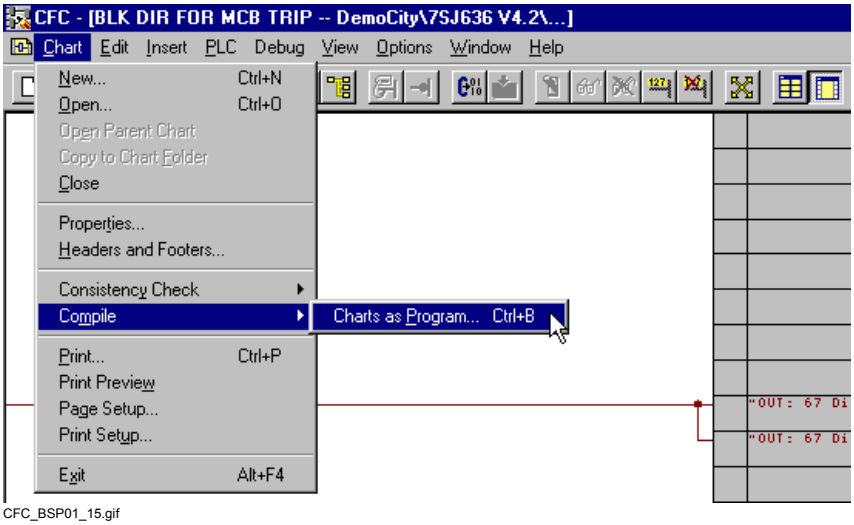
CFC_BSP01_12.gif

Fig. 2-16 Interconnected output signals: Fast PLC processing

2.5.6 Compiling a CFC Chart

In order to use the created CFC chart and thus the programmed functions in the SIPROTEC device, the chart has to be compiled:

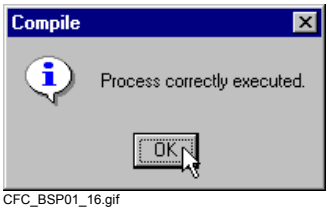
– Click on **Chart > Compile > Charts as Program...**. All the existing charts are compiled again.



CFC_BSP01_15.gif

Fig. 2-17 Compiling a CFC chart

– Confirm the displayed message box by clicking on **OK**.



CFC_BSP01_16.gif

Fig. 2-18 Compiling a message box

To use the programmed functions after compilation, you must save the parameter set in DIGSI and reload it in the SIPROTEC device.

2.6 Example of Slow PLC Processing

Slow PLC processing (priority class **PLC1_BEARB**) is carried out with a low priority. Processing is carried out event-controlled due to changes to the input signals.

Note

The functional scope of a SIPROTEC device with the MLFB **7SJ63655ER633HH3** is selected in the following example. If you want to understand the example, you must have inserted a comparable device from the device catalogue in DIGSI.

Task

The local control mode is to be visualized by an LED at the SIPROTEC device.

Input signal

- Group **Cntrl Authority**
  Display text **Cntrl Auth**

Output signal

- Group **Cntrl Authority**
  Display text **LocalCntrl**

This information does not exist in the DIGSI configuration matrix.

CFC block

- **DI_TO_BOOL** (double-point indication to Boolean)

How to proceed

- ✧ Insert the new information **Cntrl Authority LocalCntrl** into the DIGSI configuration matrix.
- ✧ Configure the **Cntrl Authority LocalCntrl** information to an LED in the DIGSI configuration matrix.
- ✧ In the DIGSI configuration matrix, configure the input signals and the output signals to the CFC see chapter 2.4).
- ✧ Create a new CFC chart with the designation **DEVICE PANEL CTRL MODE** (see chapter 2.5.1)
- ✧ Specify the priority class **PLC1_BEARB** (see chapter 2.5.2).
- ✧ Position the block **DI_TO_BOOL** (see chapter 2.5.3).
- ✧ Interconnect the input signal (see chapter 2.5.4).
- ✧ Configure the **DI_TO_BOOL** block with the thresholds **IS_OFF** and **IS_ON** for the **Cntrl Authority Cntrl Auth** double-point indication to be evaluated.
- ✧ Interconnect the output signal (see chapter 2.5.5).
- ✧ Compile the CFC chart (see chapter 2.5.6).

2.6.1 Inserting New Information

In order to insert the new information **Cntrl Authority LocalCntrl** in the DIGSI configuration matrix:

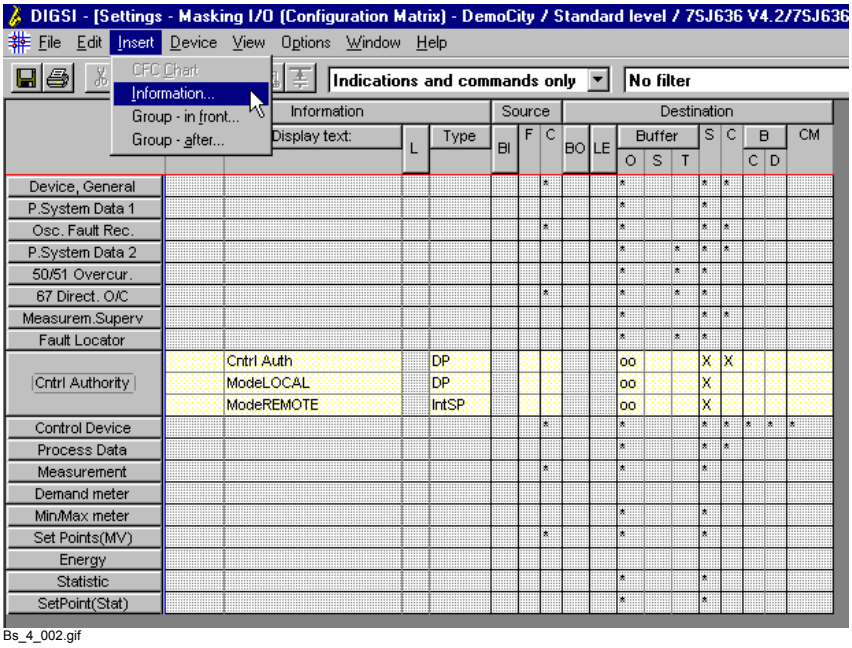
– Open the configuration matrix in DIGSI:

✧ Select the **Settings** folder in the navigation pane and

✧ Double-click on the **Masking I/O (Configuration Matrix)** function in the data pane.

– Display the group **Cntrl Authority**.

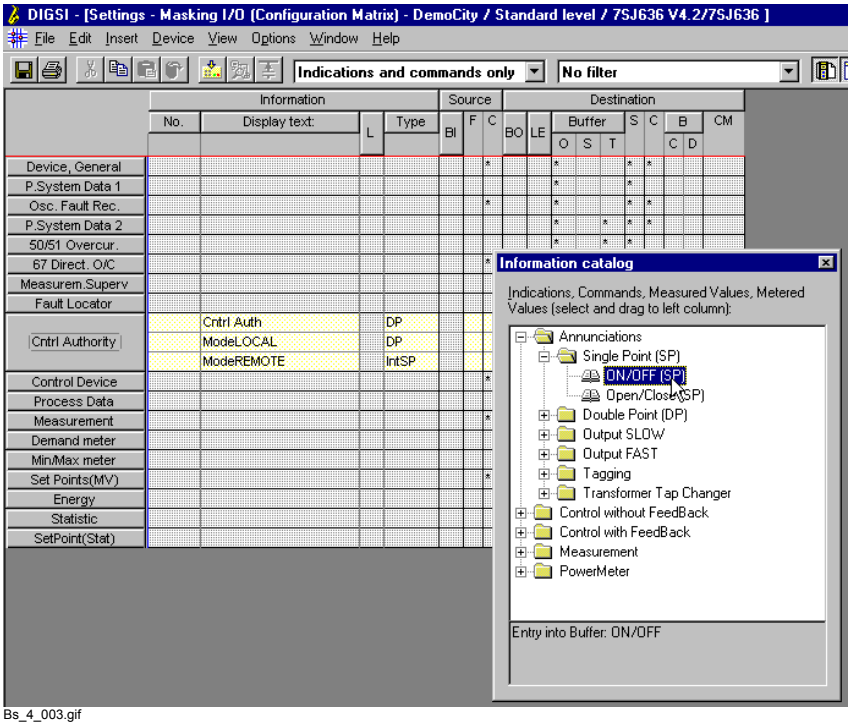
– Click on **Insert > Information...**. The **Information Catalog** dialog box is displayed.



Bs_4_002.gif

Fig. 2-19 Insert Information

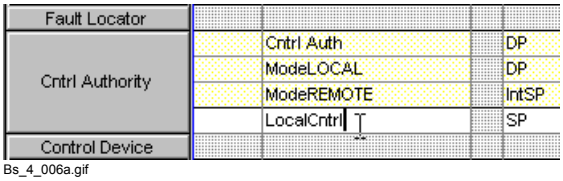
– Click on the **Annunciations\Single Point (SP)\ON/OFF (SP)** information type in the **Information Catalog** dialog box, keep the mouse button pressed and drag the information type into the group **Cntrl Authority**.



Bs_4_003.gif

Fig. 2-20 Selecting the information type

– In order to rename the inserted information, double-click on the default designation in the column **Display Text** and enter the designation **LocalCntrl**.



Bs_4_006a.gif

Fig. 2-21 Renaming information

2.6.2 Configuring Information to an LED

Configure the information **Cntrl Authority LocalCntrl** in the DIGSI configuration matrix onto an LED.

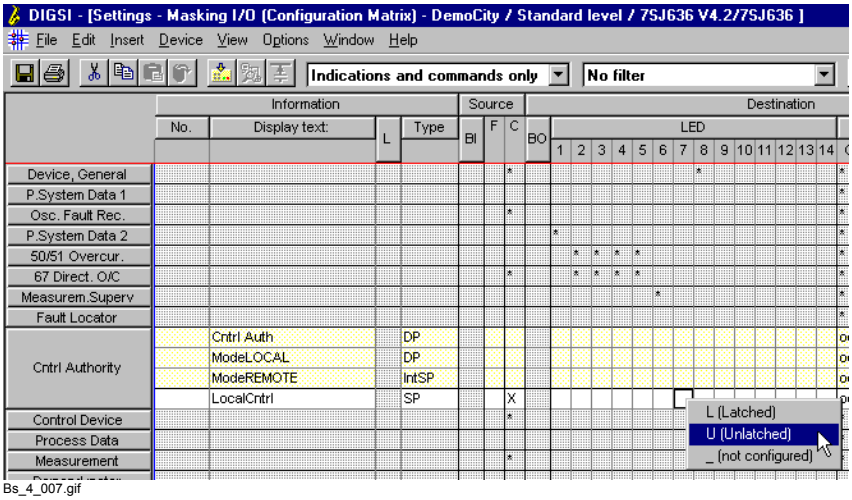
– Open the configuration matrix in DIGSI:

✧ Select the **Settings** folder in the navigation pane and

✧ Double-click on the **Masking I/O (Configuration Matrix)** function in the data pane.

– Display the group **Cntrl Authority** and select the information **LocalCntrl**.

– Right-click below **Destination** into cell 7 of the **LED** column of the configuration matrix and select **U (unlatched)** in the displayed context menu.

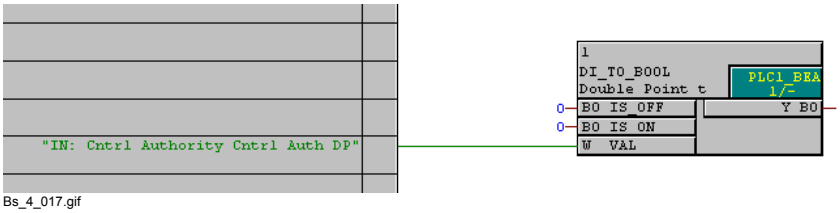


Bs_4_007.gif

Fig. 2-22 Configuring information to an LED

2.6.3 Interconnecting the Input Signal

In order to interconnect the **Cntrl Authority Cntrl Auth** input signal to the **DI_TO_BOOL** block, proceed as described in chapter 2.5.4.



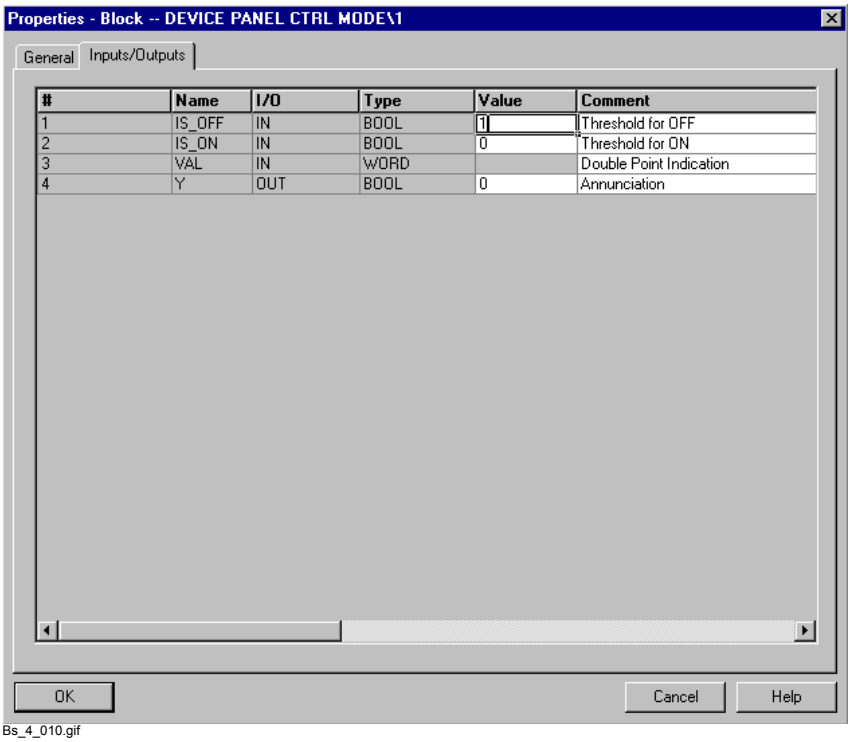
Bs_4_017.gif

Fig. 2-23 Interconnected input signal example Slow PLC processing

2.6.4 Parameterising a Block

In order to parameterise the **DI_TO_BOOL** block with the thresholds **IS_OFF** and **IS_ON** for the **Cntrl Authority Cntrl Auth DP** double-point indication to be evaluated.

- Right-click on the block and select **Object Properties...** in the displayed context menu.
- Click the **Inputs/Outputs** tab and enter 1 as threshold in the **IS_OFF** row of the **Value** column.



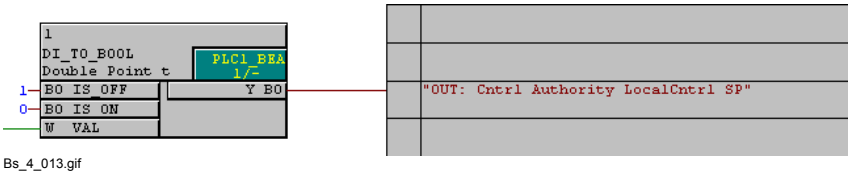
Bs_4_010.gif

Fig. 2-24 Parameterising a block

- Confirm your changes by clicking on **OK**.

2.6.5 Interconnecting the Output Signal

In order to interconnect the **Cntrl Authority LocalCntrl** output signal to the **DI_TO_BOOL** block, proceed as described in chapter 2.5.5.



Bs_4_013.gif

Fig. 2-25 Interconnected output signal example slow PLC processing

2.7 Example of Measured Value Processing

Measured value processing (priority class **MW_BEARB**) is carried out with medium priority. Processing is performed cyclically.

Note

The functional scope of a SIPROTEC device with the MLFB **7SJ63655ER633HH3** is selected in the following example. If you want to understand the example, you must have inserted a comparable device from the device catalogue in DIGSI.

Task

A sensor input is to supply the measured value pressure only in the linear range.
Messages are to be output if the range is left or if the value drops below a limit which can be set at the SIPROTEC device.

Input signals

- Group **Measurement**
  Display text **Td1=**
- Group **Set Points (MV)**
  Display text **Press<**

Output signals

- Group **Measurement**
  Display text **Superv. Pressure**
- Group **Set Points (MV)**
  Display text **SP. Pressure<**

CFC blocks

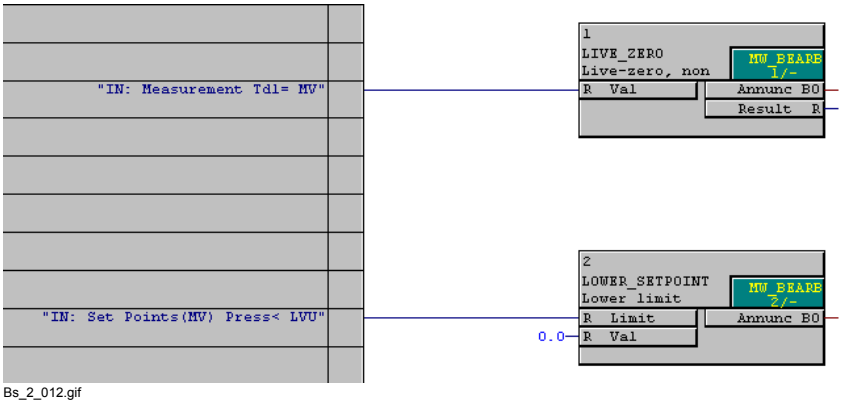
- **LIVE_ZERO** (Live Zero Monitoring)
- **LOWER_SETPOINT** (Lower Limit)

How to proceed

- ✧ In the DIGSI configuration matrix, configure the input signals and the output signals to the CFC (see chapter 2.4).
- ✧ Create a new CFC chart with the designation **MES VAL PROCESSING** (see chapter 2.5.1).
- ✧ Specify the priority class **MW_BEARB** (see chapter 2.5.2).
- ✧ Position the **LIVE_ZERO** and **LOWER_SETPOINT** blocks (see chapter 2.5.3).
- ✧ Interconnect the input signals (see chapter 2.5.4).
- ✧ Configure the **LIVE_ZERO** block with the thresholds **DetecKnee**, **DispKnee** and **LiveZero** for the measured value to be evaluated **measured values Td1=** (see chapter 2.6.4).
- ✧ Interconnect the **LIVE_ZERO** and **LOWER_SETPOINT** blocks.
- ✧ Interconnect the output signals (see chapter 2.5.5).
- ✧ Compile the CFC chart (see chapter 2.5.6).

2.7.1 Interconnecting Input Signals

To interconnect the **Measurement Td1=** and **Set Points Press<** input signals to the **LIVE_ZERO** and **LOWER_SETPOINT** blocks, proceed as described in chapter 2.5.4.



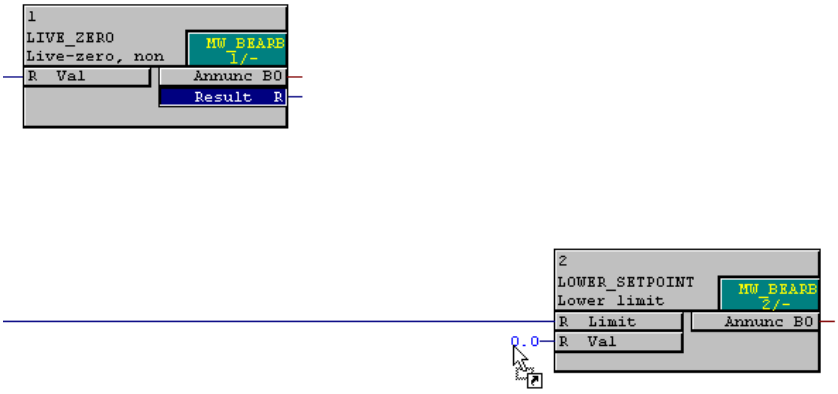
Bs_2_012.gif

Fig. 2-26 Interconnected input signals example measured value processing

2.7.2 Interconnecting Blocks

In order to interconnect the **LIVE_ZERO** and **LOWER_SETPOINT** blocks to each other:

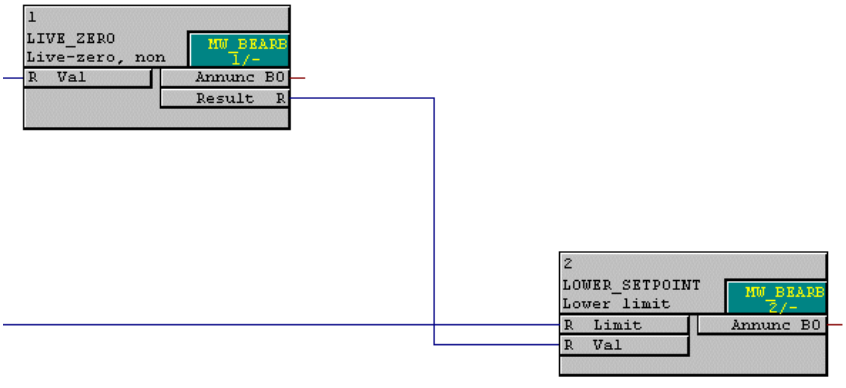
– Click on the **Result** connection of the **LIVE_ZERO** block, keep the mouse button pressed and drag the cursor to the **Val** connection of the **LOWER_SETPOINT** block.



Bs_2_013a.gif

Fig. 2-27 Interconnecting blocks example measured value processing

– Release the mouse button. A connection is displayed between the two I/Os.



Bs_2_014a.gif

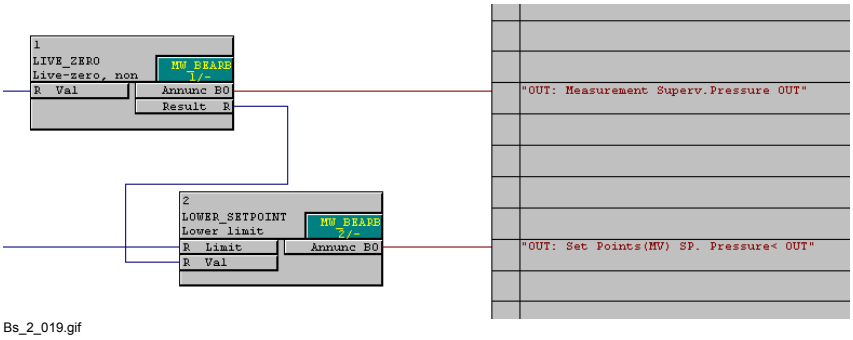
Fig. 2-28 Interconnected blocks example measured value processing

Note

Two I/Os can only be connected to each other if their data types agree.

2.7.3 Interconnecting Output Signals

In order to interconnect the **Measurement Superv. Pressure** and **Set Points (MV) SP<** to the **LIVE_ZERO** and **LOWER_SETPOINT**, proceed as follows under chapter 2.5.5.



Bs_2_019.gif

Fig. 2-29 Interconnected output signals example measured value processing

2.8 Example of Interlocking

Interlocking (priority class **SFS_BEARB**) is used when a command is to be output. In addition, cyclical processing occurs in the background.

Note

The functional scope of a SIPROTEC device with the MLFB **7SJ63655ER633HH3** is selected in the following example. If you want to understand the example, you must have inserted a comparable device from the device catalogue in DIGSI.

When a SIPROTEC device is inserted, standard CFC charts are also inserted. The CFC chart **Interlocking** already uses the output signals used in the following example. To understand the example, you must first delete the CFC chart **Interlocking**.

Task

The interlocking condition for switching an earth switch on and off is to be checked and the enable indications generated.

Input signals

- Group **Control Device**
  Display text **52Breaker**
  Long text **52 Breaker**
- Group **Control Device**
  Display text **Disc.Swit.**
  Long text **Disconnect Switch**
- Group **Control Device**
  Display text **GndSwit.**
  Long text **Ground Switch**
- Group **Process Data**
  Display text **>DoorClose**
  Long text **>Door closed**

Output signals

- Group **Control Device**
  Display text **GndSw. Cl.**
  Long text **Interlocking: Ground switch Close**
- Group **Control Device**
  Display text **GndSw. Open**
  Long text **Interlocking: Ground switch open**

CFC blocks

- **DM_DECODE** (Double Point decoding)
- **X_OR** (XOR Gate)
- **AND** (AND Gate)

How to proceed

- ✧ In the DIGSI configuration matrix, configure the input signals and the output signals to the CFC see chapter 2.4).
- ✧ Create a new CFC chart with the designation **INTERL GND SWITCH** (see chapter 2.5.1).
- ✧ Specify the priority class **SFS_BEARB** (see chapter 2.5.2).
- ✧ Position the **DM_DECODE**, **X_OR** and **AND** blocks (see chapter 2.5.3).
- ✧ Interconnect the input signals (see chapter 2.5.4).
- ✧ Increase the number of inputs of the **AND** block to four.
- ✧ Interconnect the **DM_DECODE**, **X_OR** and **AND** blocks (see chapter 2.7.2).
- ✧ Interconnect the output signals (see chapter 2.5.5).
- ✧ Compile the CFC chart (see chapter 2.5.6).

2.8.1 Interconnecting Input Signals

In order to interconnect the input signals **Control Device 52Breaker DP**, **Control Device Dis.Swit. DP**, **Control Device GndSwit.** and **Process indications >DoorClose SP** to the **DM_DECODE**, **X_OR** and **AND** blocks, proceed as described in chapter 2.5.4.

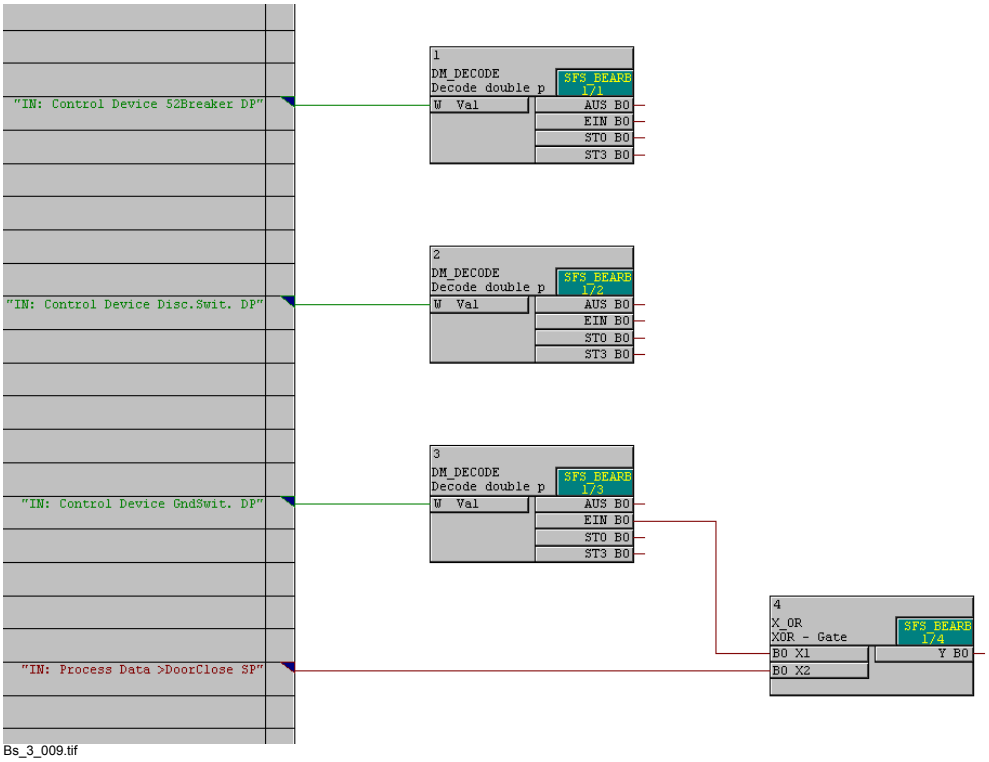
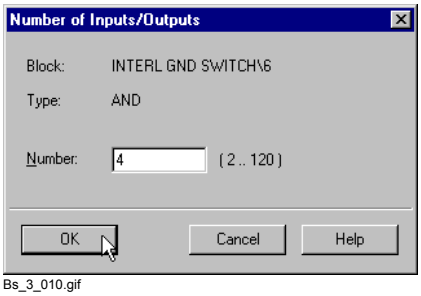


Fig. 2-30 Interconnected input signals example interlocking

2.8.2 Increasing the Number of Inputs of a Block

In order to increase the number of inputs of the **AND** block to four:

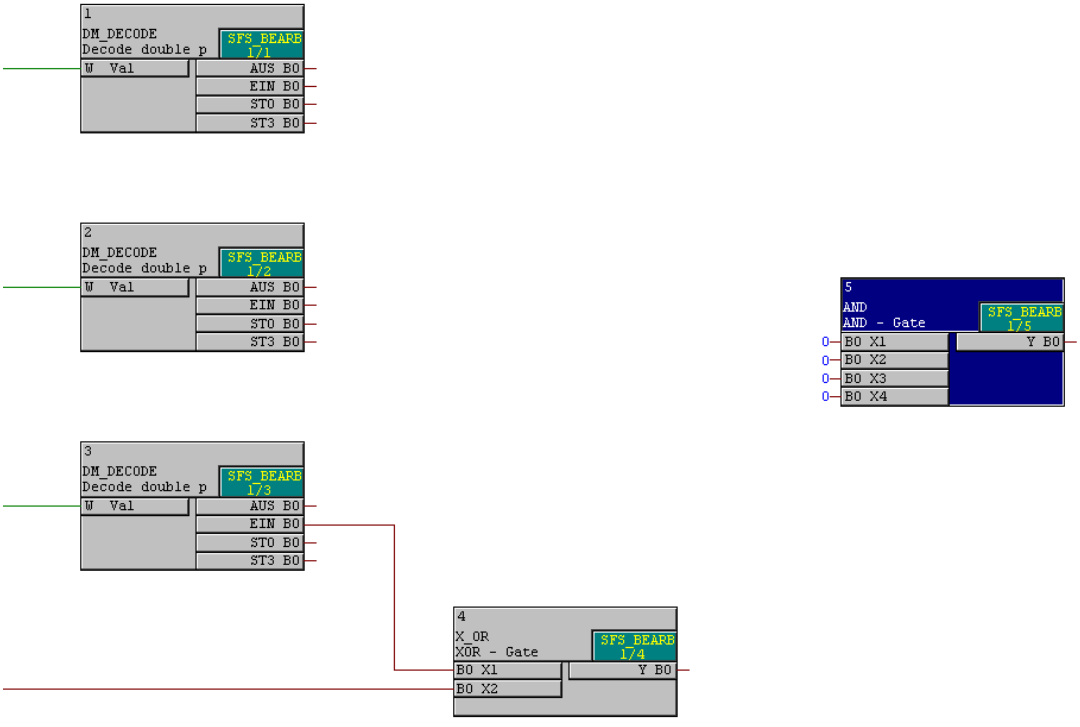
- Right-click on the block and select **Number of I/Os...** in the displayed context menu.
- Enter the value 4 in the **Length** input field of the displayed window **Number of I/Os** and confirm by clicking on **OK**.



Bs_3_010.gif

Fig. 2-31 Number of I/Os dialog box

The block is displayed with four inputs.

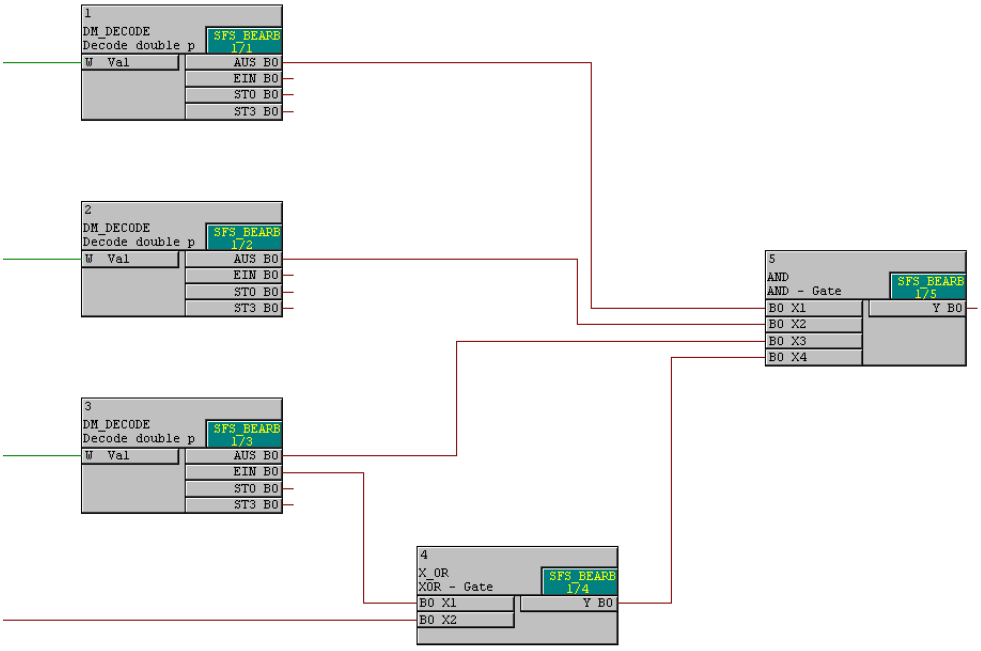


Bs_3_011.tif

Fig. 2-32 Number of inputs at the AND block increased

2.8.3 Interconnecting Blocks

To interconnect the **DM_DECODE**, **X_OR** and **AND** blocks to one another, proceed as described in chapter 2.7.2.

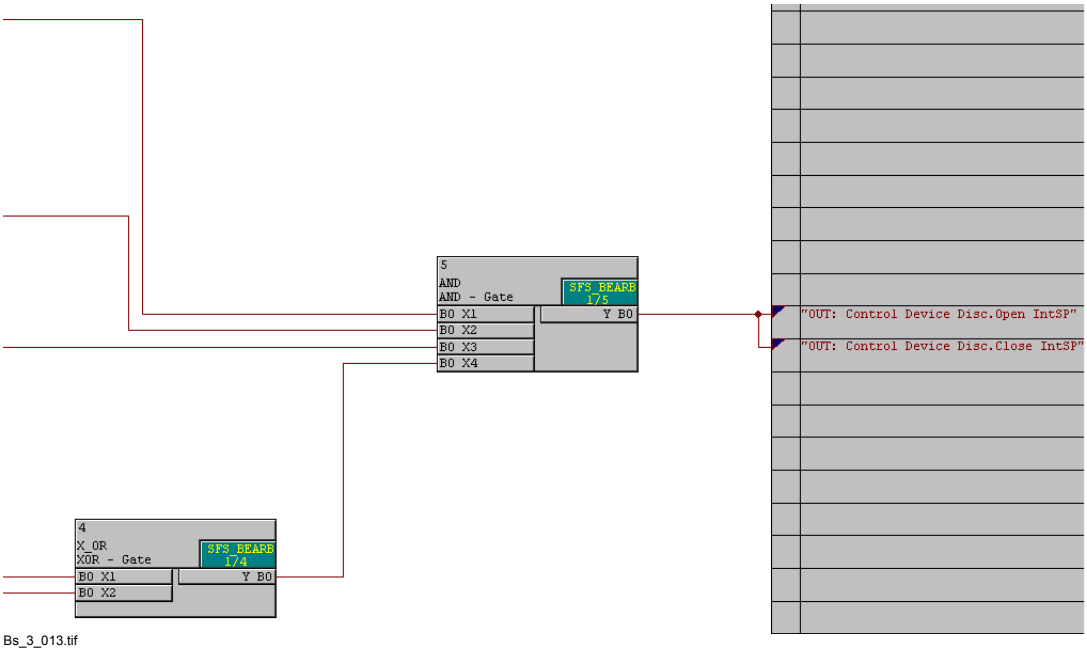


Bs_3_012.tif

Fig. 2-33 Interconnected blocks example interlocking

2.8.4 Interconnecting Output Signals

In order to interconnect the **Control Device GndSw Cl.** and **Control Device GndSw Open** to the **AND** block, proceed as described in chapter 2.5.5.



Bs_3_013.tif

Fig. 2-34 Interconnected output signals example interlocking

3 Implementation Examples

Overview

This chapter contains typical solutions from practical implementation.

Note

It is assumed for the description that you are familiar with the operation of DIGSI V4 or DIGSI CFC.

Information on the basic operation of DIGSI CFC can be found in chapter 2 of this manual.

Note

The CFC blocks in this chapter are displayed with the option **Block width: Wide**:

- ✧ In the CFC Editor, click **Options > Customize > Block/Sheet Bar settings**.
- ✧ Activate the option **Wide** in the displayed window under **Block width** and confirm with **OK**.

Contents

3.1 Setting Group Change Option

Task

The following task is to be implemented:

- A motor is connected to the binary output BA1. The motor is to be started directly via the function key F1.
- When all the phase currents lie under 5% of the nominal value
  and in the first 10 seconds after the motor has been started,
  setting group B is to be activated.
- Setting group A is to be activated in normal operation.

Starting a motor directly via F1

The motor is started directly via the function key F1 as follows:

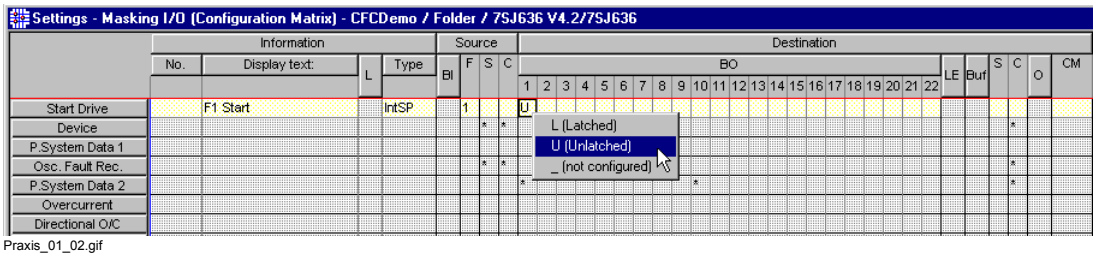
✧ Create a group called **Start Motor** in the DIGSI configuration matrix.

✧ Insert a new information of the type **Marking ON/OFF (IE)** into the **Start Motor** group. The information is inserted with a default designation.

✧ Rename the new information to **F1 Start**.

✧ Configure the information **F1 Start** to the function key **1**.

✧ Configure the information **F1 Start** to the binary output **BO1**. The output is to be realized **unlatched**.



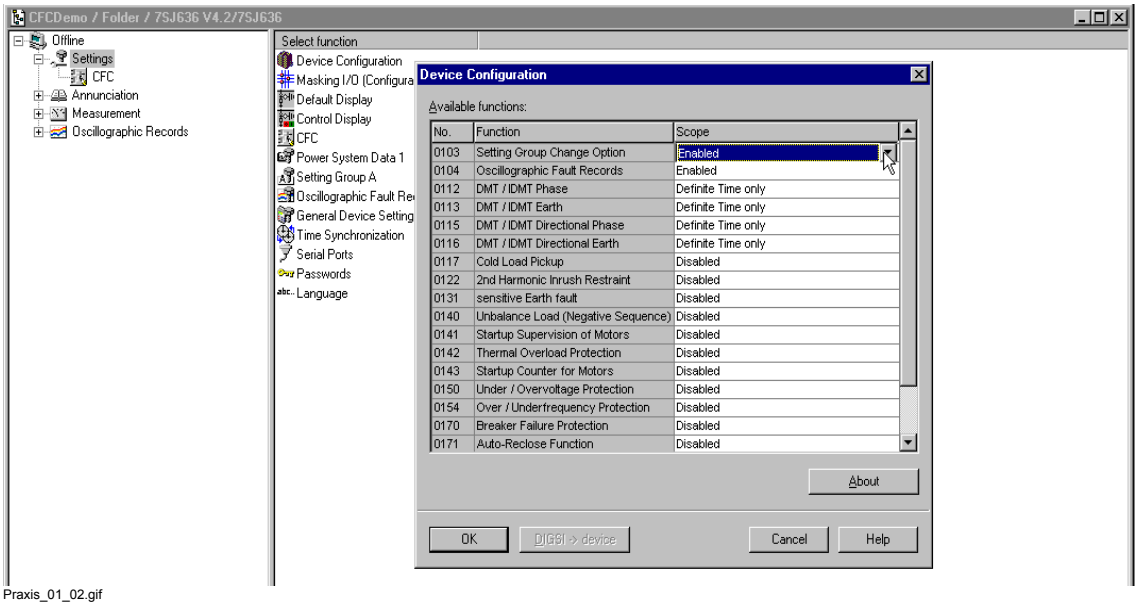
Praxis_01_02.gif

Fig. 3-1 starting motor directly via the function key F1

Activation the setting group change option

In order to activate the setting group change option:

✧ Define the **Setting Group Change Option** as **existing** in the **Functional scope** of the device.



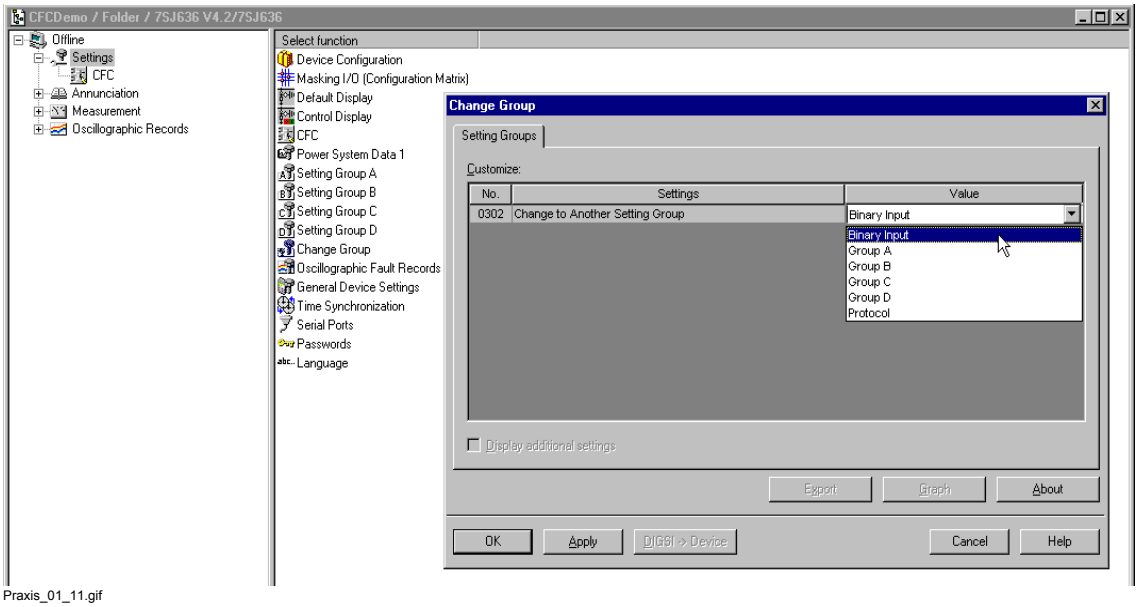
Praxis_01_02.gif

Fig. 3-2 Activating the setting group change option

Setting setting group change option dynamically

To affect setting group change option dynamically via CFC logic, activate setting group change option via binary input:

✧ Specify the value **Via binary input** or **Via Protocol** for the **Activation** in the **Setting Group Change Option**.

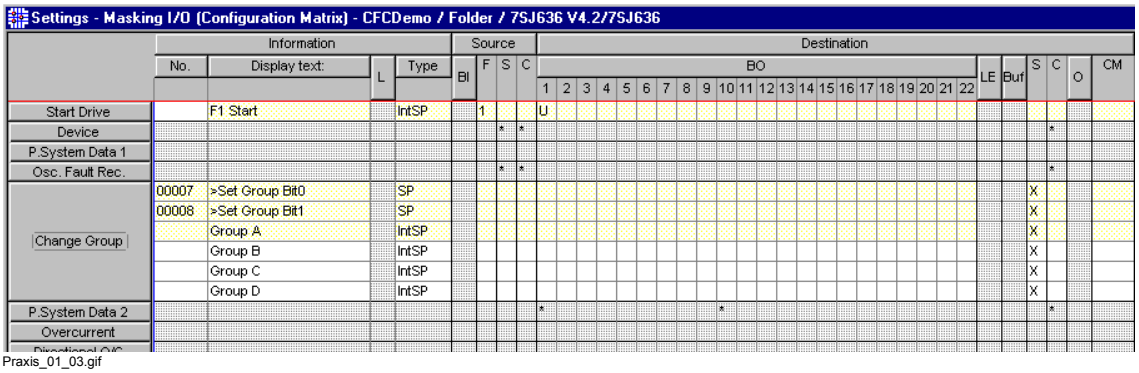


Praxis_01_11.gif

Fig. 3-3 Setting setting group change option dynamically

Setting group change option

When the setting group change option is activated, a new group **Grp Chge OPTION** is displayed in the DIGSI configuration.

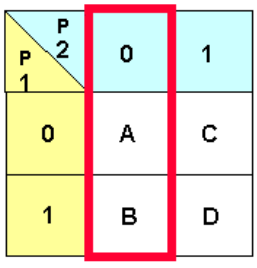


Praxis_01_03.gif

Fig. 3-4 Setting group change option in DIGSI configuration

Specifying the active setting group

The active setting group (A, B, C and D in Figure 3-5) is specified in the group **Grp Chge OPTION** via the two information items **>Set Group Bit1** (P1 in Figure 3-5) and **>Set Group Bit2** (P2 in Figure 3-5):
The setting group **B** is active when **>Set Group Bit1** has the binary value **1** and **>Set Group Bit2** has the binary value **0**.



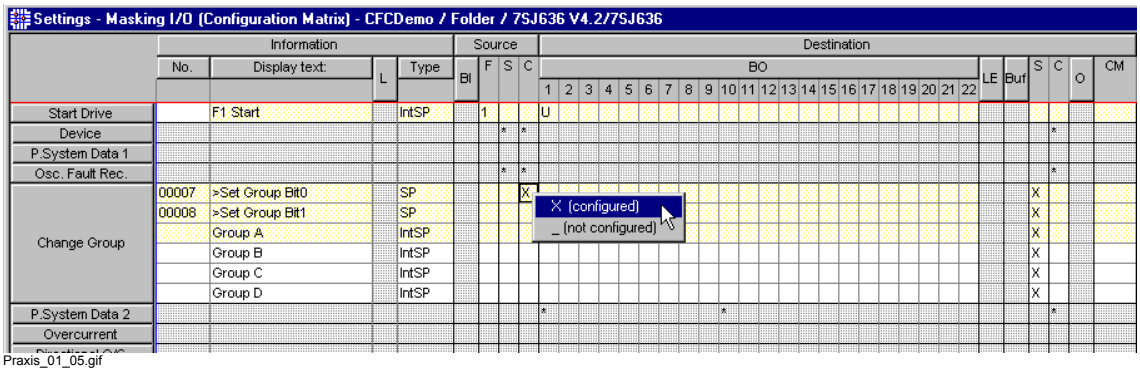
Praxis_01_04.gif

Fig. 3-5 Active setting group

Changing the active setting group

To switch between the setting groups A and B, it is sufficient to specify the binary value of the information **>Set Group Bit1** (P1 in Figure 3-5) via a CFC program:

✧ Configure the information **>Set Group Bit1** as a source to CFC.



Praxis_01_05.gif

Fig. 3-6 Configuring >Set Group Bit1 as a source to the CFC

Division of the change option into various priority classes

A CFC program changes to setting group **B**,

- When all the phase currents lie under 5% of the nominal value or
- In the first 10 seconds after the motor has been started.

The phase currents are monitored in the CFC priority class **MW_BEARB** (Measured value processing).

In order to change the setting group after 10 seconds have expired, use a timer. However, the provided **TIMER** (universal timer) block cannot be executed in the CFC priority class **MW_BEARB** (measured value processing). The CFC priority class **PLC1_BEARB** (Slow PLC processing) is required for this purpose.

Since different priority classes are required, divide the CFC program into two CFC charts with the following priority classes:

- CFC chart **ParaUmsch:**
  Priority class **MW_BEARB** (meter processing):
  Monitoring the phase currents
- CFC chart **ParaUmsch1**:
  Priority class **PLC1_BEARB** (Slow PLC processing):
  Monitoring the start-up time and changing the setting group

Communication between various priority classes

Various priority classes (for example MW_BEARB and PLC1_BEARB) can communicate with each other via an information if the type **Marking ON/OFF (IE)**:
In the one priority class (for example MW_BEARB in the CFC chart **ParaUmsch**) a value is assigned to the information while in another priority class (for example PLC1_BEARB in the CFC chart **ParaUmsch1**) die information is evaluated.

Note

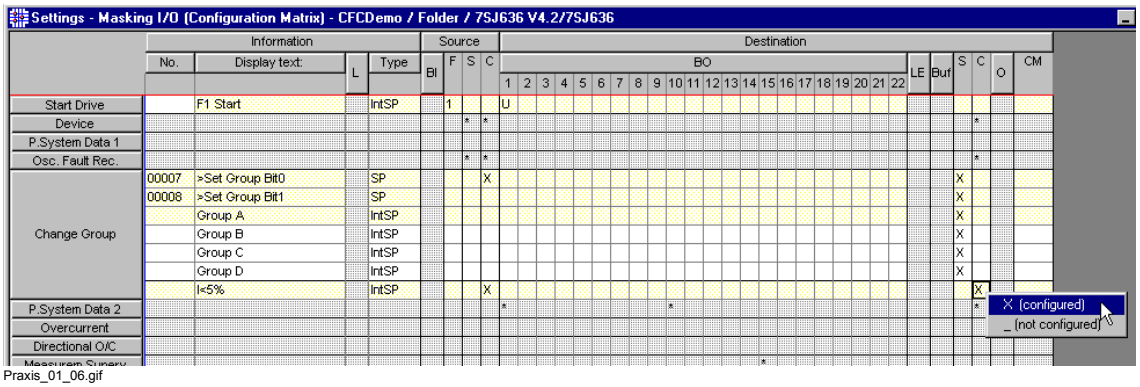
Blocks of different priority classes within a CFC chart are not permissible. Use a separate CFC chart for each priority class!

Within a CFC chart you can use an information of the type **Marking ON/OFF (IE)** either exclusively on the right border or exclusively on the left border.

Using information for communication

In order to use an information for communication between the priority classes MW_BEARB and PLC1_BEARB:

- Insert a new information of the type **Marking ON/OFF (IE)** into the **Grp Chge OPTION** group. The information is inserted with a default designation.
- Rename the new information to **I<5%**.
- Configure the information **I<5%** as a **Source** and as a **Desitnation** to the CFC.



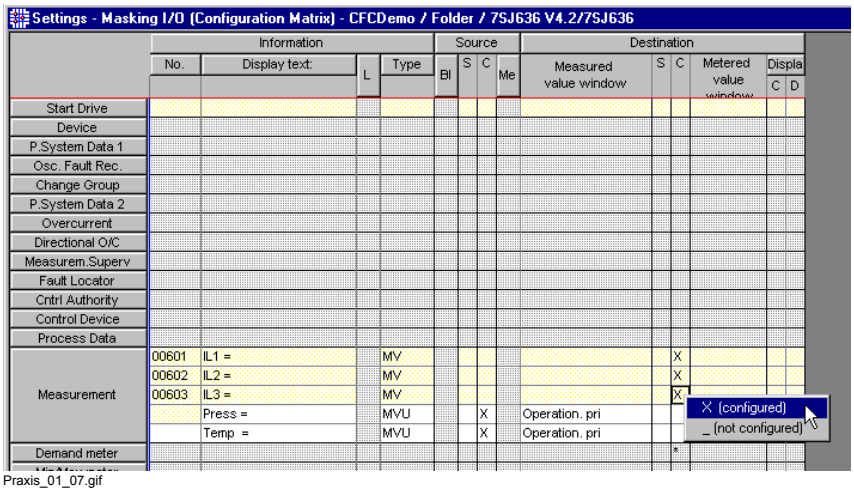
Praxis_01_06.gif

Fig. 3-7 Configuring information as a source and as a destination to the CFC

Monitoring phase currents as a CFC program

In order to create the monitoring of the phase currents as a CFC program:

✧ Configure the phase currents as a **Destination** to the CFC.



Praxis_01_07.gif

Fig. 3-8 Configuring phase currents as a destination to the CFC

✧ Create a new chart called **ParaUmsch** in the CFC and open the CFC chart.

✧ Use the **Edit > Run Sequence** menu to specify the priority class **MW_BEARB** as the standard priority class. Every new block is inserted into the priority class **MW_BEARB**.

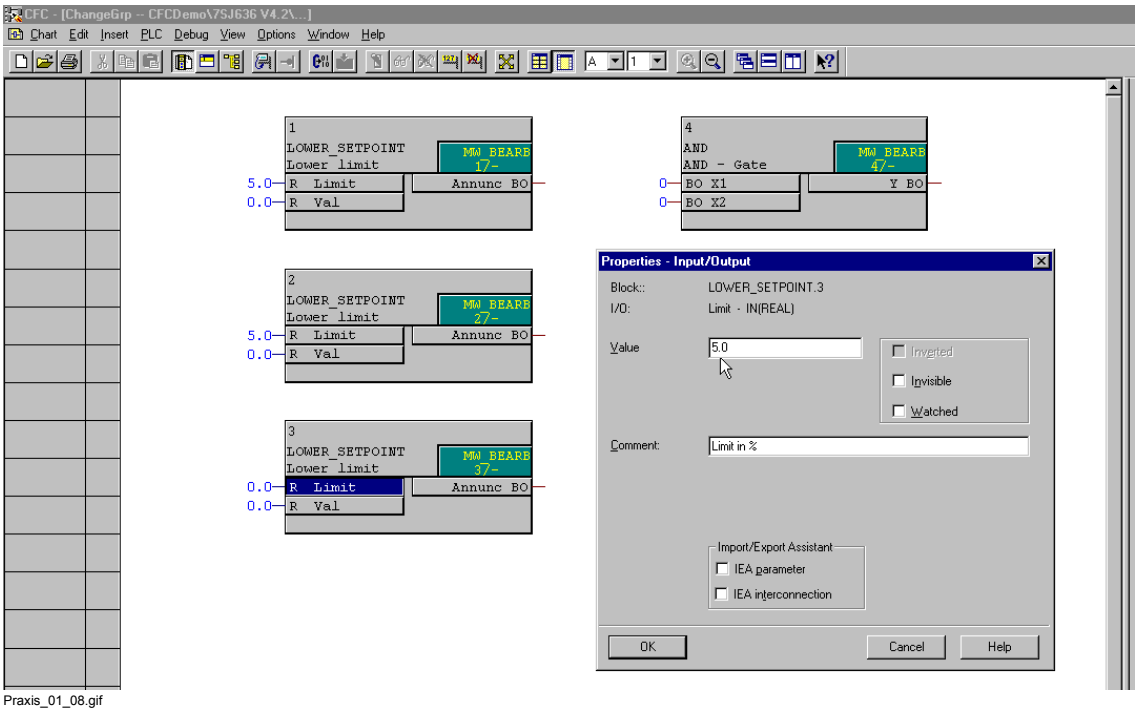
✧ Monitoring of the phase current is implemented with the **LOWER_SETPOINT** (lower limit) block. All three monitoring functions are gated via the **AND** (AND gate) block from which they form the information **I<5%** for communicating with the further priority class **PLC1_BEARB**.

✧ Position the CFC blocks.

Note

When inserting the individual CFC blocks observe the default run sequence. If necessary, correct the run sequence by using the menu **Edit > Run Sequence**.

✧ Parameterise the LOWER_SETPOINT blocks by using the **Object Properties** context menu to the limit **5.0%**.

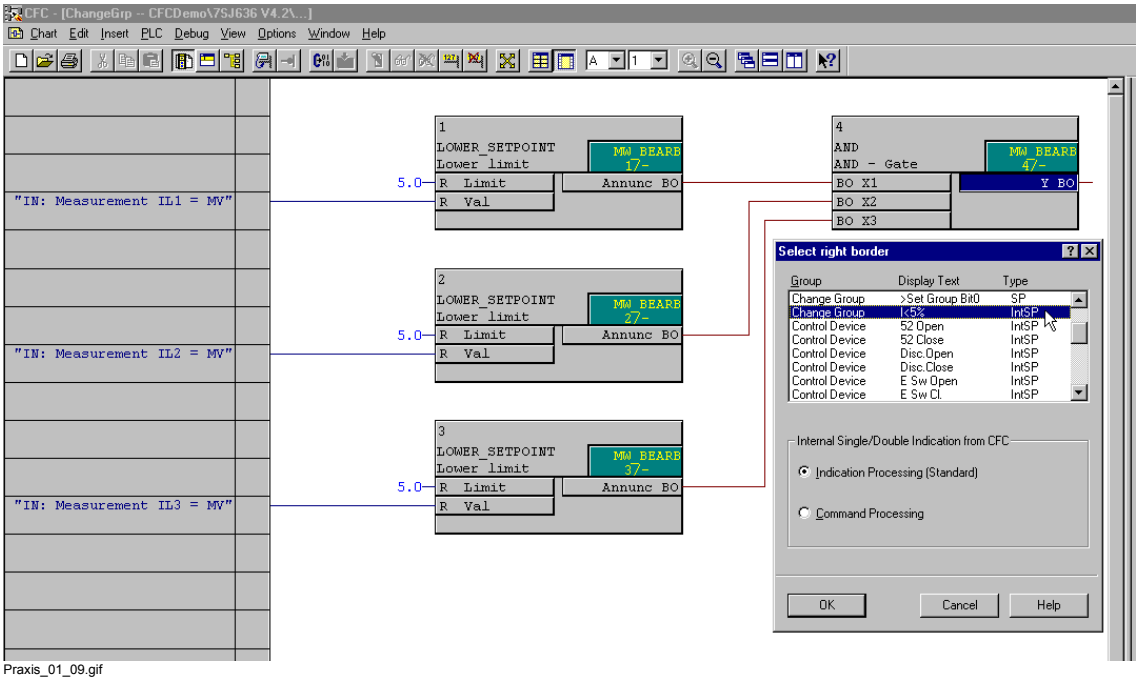


Praxis_01_08.gif

Fig. 3-9 Parameterising the limit 5.0%

✧ Use the **Number of I/Os** context menu to increase the number of I/Os to 3 for the **AND** block.

✧ Interconnect the blocks with each other and with the operands on the borders. In the process interconnect the output of the **AND** block to the information **I<5%** on the right border.



Praxis_01_09.gif

Fig. 3-10 Creating connections between blocks and border

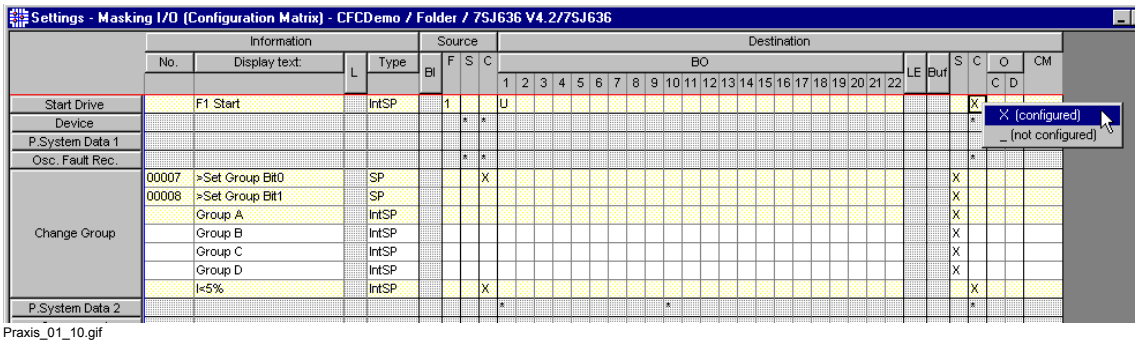
Compiling CFC chart

After you have created the CFC chart, you can compile the CFC chart via the **Chart > Compile > Charts as Program** menu.

Monitoring the start-up time and changing the setting group as a CFC program

In order to create the monitoring of the start-up time and the changing of the setting group as a CFC program:

- ✧ Configure the information **F1 Start** as a destination to the CFC.



Praxis_01_10.gif

Fig. 3-11 Configuring the information F1 Start as a destination to the CFC

- ✧ Create a new chart called **ParaUmsch1** in the CFC and open the CFC chart.
- ✧ Use the **Edit > Run Sequence** menu to specify the priority class **PLC1_BEARB** as the standard priority class. Every new block is inserted into the priority class **PLC1_BEARB**.
- ✧ Change to an empty sheet of the existing CFC chart.
- ✧ Monitoring of the start-up time is implemented with the **TIMER** (universal timer) block which is started via the information **F1 Start**. The binary signal **1** is applied at the **QT1** output while the timer is running. Link the output signal of the timer with the information **I<5%** from the priority class **MW BEARB** via the **OR** (OR gate) block. The signal at the output of the **OR** block controls the information **>Set Group Bit1** directly and thus the setting group change option.
- ✧ Position the CFC blocks.

Note

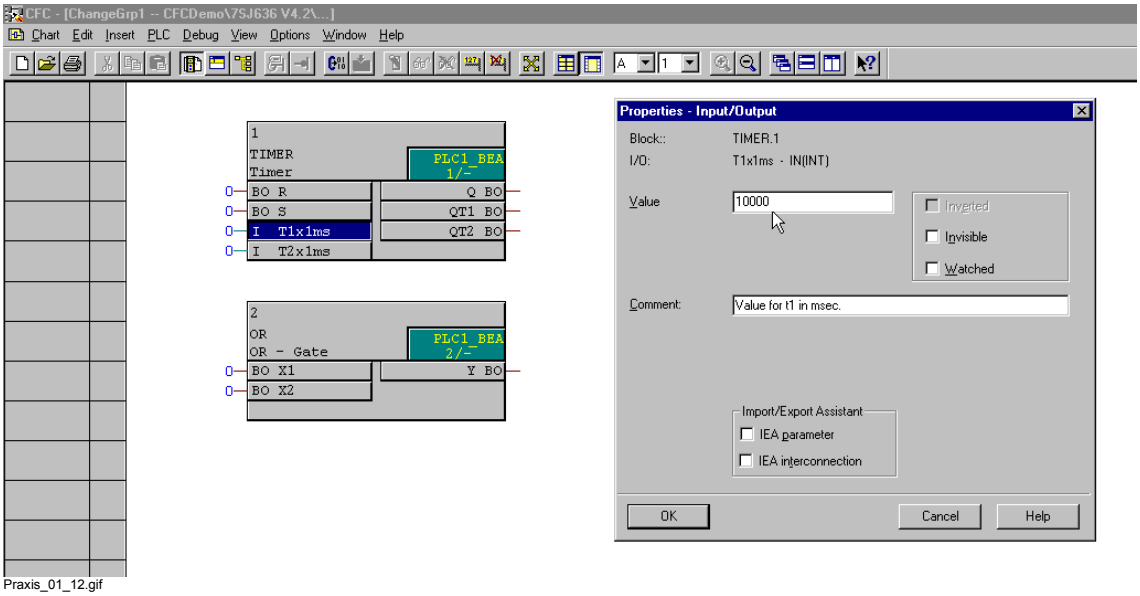
When inserting the individual CFC blocks observe the default run sequence. If necessary, correct the run sequence by using the menu **Edit > Run Sequence**.

Note

In the example above, the **TIMER** (universal timer) block is used for the monitoring of the start-up time of the motor. Depending on the version of the SIPROTEC device used, you can also use the **TIMER_SHORT** (simple timer) or **LONG_TIMER** (timer (max. 1,193 h)) blocks for monitoring.

The **LONG_TIMER** (timer (max. 1,193 h)) block can also be executed in the CFC priority class **MW_BEARB** (measured value processing). When using this block, you can implement the entire setting group switching in the CFC priority class **MW_BEARB** (measured value processing). A division of switching to different priority classes is not necessary.

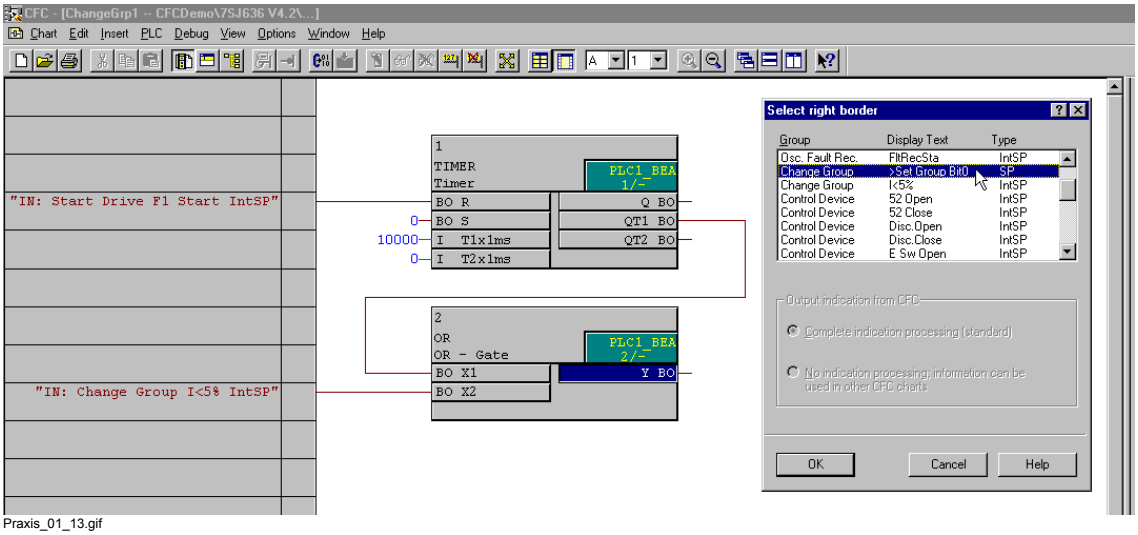
✧ Use the **Object Properties** context menu to parameterise the value for **T1x1ms** to **10000 ms** in the **TIMER** block.



Praxis_01_12.gif

Fig. 3-12 Parameterising the limit 5.0%

✧ Interconnect the blocks with each other and with the operands on the borders. In the process interconnect an input of the **OR** block to the information **I<5%** on the left border. In the process interconnect the output of the **OR** block to the information **>Set Group Bit1** on the right border.



Praxis_01_13.gif

Fig. 3-13 Creating connections between blocks and border

Compiling CFC chart

After you have created the CFC chart, you can compile the CFC chart via the **Chart > Compile > Charts as Program** menu.

3.2 Flashing LED

Flashing LEDs are often used to draw attention to a particular state (for example, safety-critical state) of a plant. In SIPROTEC 4 devices you can use a CFC program to simulate the flashing mechanism.

Note

The **BLINK** block is available for SIPROTEC devices of version V4.5.
This block can be used in all priority classes.

Task

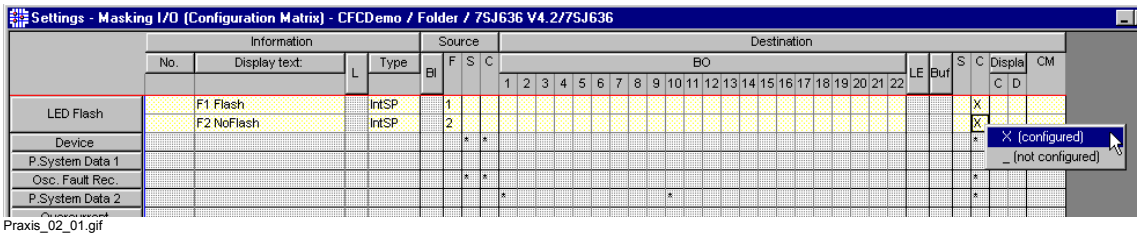
The following task is to be implemented:

- In many cases a short pulse (for example pressing the function key F1) is used to change to a safety-critical state. The state is signalled by a flashing LED.
- A further short pulse (for example the function key F2) is used to terminate the safety-critical state and thus the flashing of the LED.

Preparing evaluation of function keys via CFC program

In order to prepare the evaluation of the function keys F1 and F2 via a CFC program:

- ✧ Create a group called **LED Flash** in the DIGSI configuration function.
- ✧ Insert a new information of the type **Marking ON/OFF (IE)** into the **LED Flash** group for each function key. The items of information are inserted with a default designation.
- ✧ Rename the new items of information to **F1 Flash** and **F2 NoFlash**.
- ✧ Configure the function key **1** to the information **F1 Flash**.
- ✧ Configure the function key **2** to the information **F2 NoFlash**.
- ✧ Configure the items of information **F1 Flash** and **F2 NoFlash** as a **Destination** to the CFC.



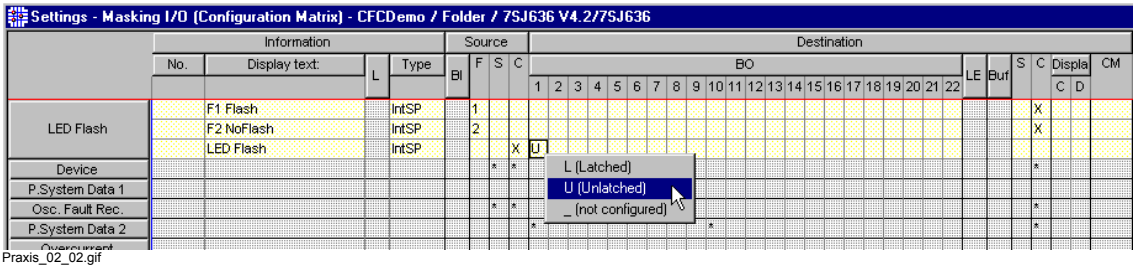
Praxis_02_01.gif

Fig. 3-14 Preparing the function keys for evaluation via the CFC program

Preparing control of LED via CFC Program

In order to prepare the control of an LED via the CFC program:

- ✧ Insert a new information of the type **Marking ON/OFF (IE)** into the **LED Flash** group for the LED. The information is inserted with a default designation.
- ✧ Rename the new information to **LED Flash**.
- ✧ Configure the information **LED Flash** as a **source** to CFC.
- ✧ Configure the LED **1** to the information **LED Flash**. The output is to be realized **unlatched**.



Praxis_02_02.gif

Fig. 3-15 Preparing the control of an LED via the CFC program

Flashing rhythm

The chronological sequence of the flashing rhythm can be described by the interval changes for **LED is off** (t_{Off}=250 ms) and **LED is on** (t_{On}=50 ms).

Simulating flashing in CFC program

In order to simulate the flashing of an LED in a CFC program:

- ✧ Create a new chart called **LEDFlash** in the CFC and open the CFC chart.
- ✧ Use the **Edit > Run Sequence** menu to specify the priority class **PLC1_BEARB** as the standard priority class. Every new block is inserted into the priority class **PLC1_BEARB**.

Note

The **TIMER** (universal timer) block functions exclusively in the priority classes Fast PLC processing (priority class **PLC_BEARB**) and Slow PLC processing (priority class **PLC1_BEARB**).

- ✧ Flashing is implemented with two **TIMER** (universal timer) blocks:

  The first timer is started via the signal **F1 Flash**. Immediately after the start the internal timer **T1** runs for 250ms (t_{Off}) and sets the output **QT1** to the binary signal **1** s for this period.

  The output signal **QT1** of the first timer serves as the starting signal for the second timer. Here, however, the internal timer **T2** is used which does not start until the starting signal at the start input drops again, meaning exactly after the time t_{Off}=250ms has expired. It runs for 50ms (t_{On}) and sets the corresponding output **QT2** to the binary signal **1** for this period. This signal is gated to the LED (**LED Flash**) which is illuminated correspondingly. After the timer expires the LED extinguishes.

  The sequence has to be repeated in order for the LED to flash. Therefore the output **QT2** of the second timer is gated via the **OR** (OR gate) block to the start input of the first timer block. This is a permitted feedback. The warning which occurs when the CFC chart is compiled is meaningless.

  In order to switch off the flashing LED connect the reset inputs of the two timers to the signal **F2 NoFlash**.

- ✧ Position the CFC blocks.

Note

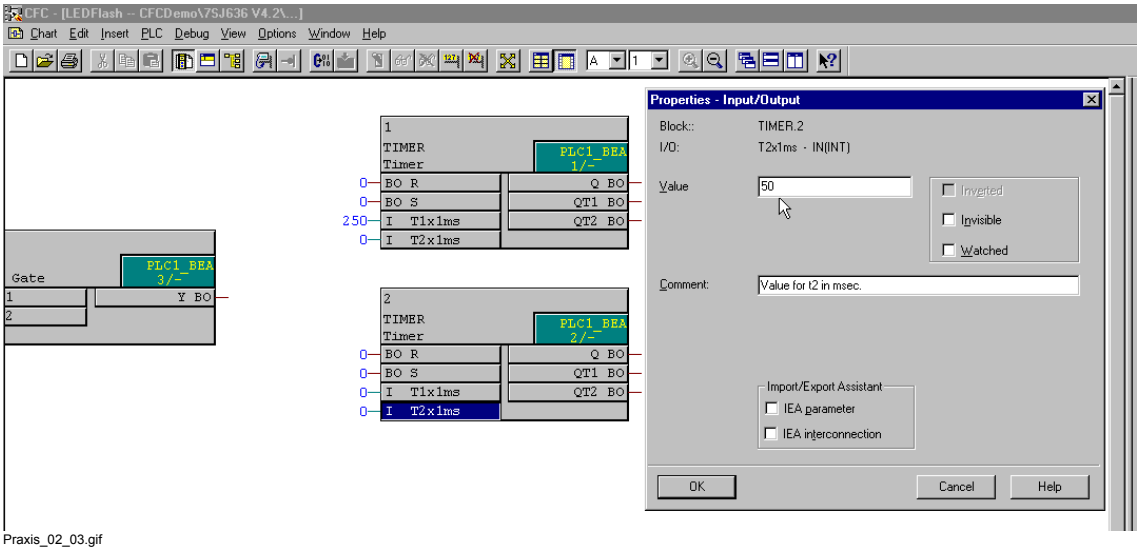
When inserting the individual CFC blocks observe the default run sequence. The **OR** block has to have a sequence number which is larger than the sequence number of the second **TIMER** block. If necessary, correct the run sequence by using the menu **Edit > Run Sequence**.

Note

In the example above, the **TIMER** (universal timer) block is used for the simulation of the flash function. Depending on the version of the SIPROTEC device used, you can also use the **TIMER_SHORT** (simple timer) or **LONG_TIMER** (timer (max. 1,193h)) blocks for monitoring.

The **LONG_TIMER** (timer (max. 1,193 h)) block can be executed in all CFC priority classes.

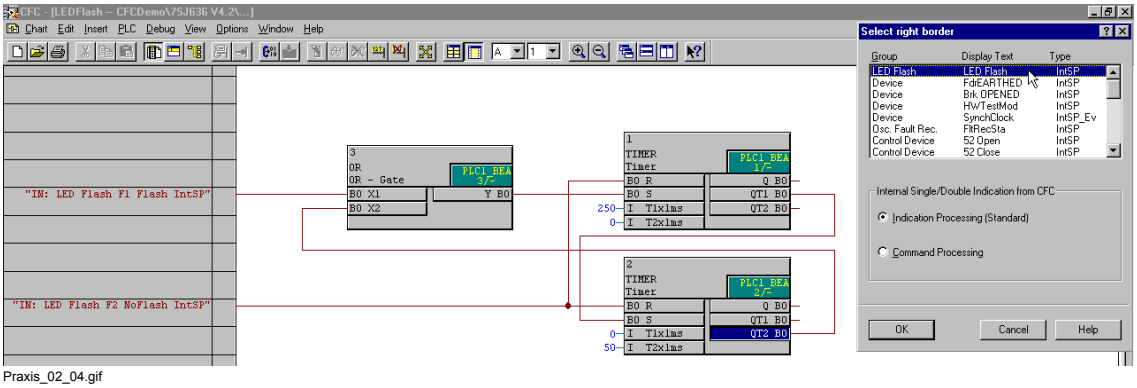
✧ Parameterise the **TIMER** blocks to the values for the flashing rhythm by means of the **Object Properties** context menu:
Value for **T1x1ms** to **250 ms** (Timer 1) and
Value for **T2x1ms** to **50 ms** (Timer 2).



Praxis_02_03.gif

Fig. 3-16 Parameterising the timer

✧ Interconnect the blocks with one another and with the operands at the borders (see figure 3-17).
Connect the set input **S** of the first **TIMER** block to the information **F1 Flash** at the left border.
Connect the reset inputs **R** of the two **TIMER** blocks to the information **F2 NoFlash** at the left border.
Connect the output **QT2** of the second **TIMER** block to the information **LED Flash** at the right border.



Praxis_02_04.gif

Fig. 3-17 Creating connections between blocks and border

Compiling CFC chart

After you have created the CFC partial chart, you can compile the CFC overall chart via the **Chart > Compile > Charts as Program** menu.

Note

When compiling the CFC chart a warning is output due to a feedback within the CFC chart. The feedback is permitted in this case and the warning is meaningless.

Continuous signal as a starting signal for flashing

If a continuous signal (for example keyswitch) is to be used as the starting signal for flashing instead of a short pulse (for example function key F1), the continuous signal has to be converted.

The continuous signal can be converted via an additional upstream **TIMER** block into two short signals corresponding to the signals **F1 Flash** and **F2 NoFlash**:

The signal to be converted is gated to the start input of this additional timer. **T1x1ms** and **T2x1ms** each have a value of 10 ms assigned to them:
Irrespective of the signal duration the coming start signal results in a short (10 ms) pulse at the output **QT1**. The going start signal results in a short (10 ms) pulse at the output **QT2**. The output **QT1** (corresponding to the signal **F1 Flash**) is therefore connected to the set input **S** of the first timer. The output **QT2** (corresponding to the signal **F2 NoFlash**) is connected to the reset input **R** of both timers.

3.3 Reverse Interlocking

Economical bus-bar protection with overcurrent protection devices can be set up using **reverse interlocking**.

It is assumed that the bus-bar is fed via a feeder and the other feeders go to the loads. A typical use is for bus-bars in electricity distribution networks of high and average voltage. In the high and highest voltage, this principle is seldom used, and separate bus-bar protection is used.

Principle

The principle is quite simple. The normal pickup stage **I>** of overcurrent protection devices in the feeders, which lead to the loads, blocks the **high-current pickup stage I>>** of the feeding feeder via a binary input.
The delay time of the high-current stage of the feeding feeder is set so that a secure blocking by the **I>** pickup of the feeder is ensured (50-100 ms).
The **I>>** pickup of the feeding feeder must be set clearly higher than the **I>** pickup of the feeder to the loads to ensure secure addressing of the **I>** pickup.
A malfunction during reverse interlocking causes the bus-bar to switch off.
For this reason, this function has a high security relevance, from a protection point of view.

3.3.1 Intended Behaviour of Interlocking During a Short-Circuit

External short-circuit on a feeder

An external short-circuit causes pickup of the **I>** stage of the overcurrent protection at a feeder. This pickup is configured to a contact and blocks the **I>>** stage of the feeding feeder via a binary input assigned to block.
Thus the **I>>** stage cannot trip, even if its delay time has expired. The short-circuit is switched off by overcurrent protection devices of the short-circuiting feeder. With dropout of the pickup of the **I>** stage, blocking is reset, as the short-circuit is no longer in effect.

Short-circuit on the bus-bar

The **I>>** stage of the feeding feeder is set so that it securely picks up during a bus-bar short-circuit. A bus-bar short-circuit does not lead to pickup of one of the **I>** stages of the other feeder.
Only the **I>>** stage picks up with this. Once the set delay time has expired, a TRIP command is output and the feeding circuit breaker trips. Thus the bus-bar short-circuit is remedied.

3.3.2 Reverse Interlocking via Discrete Wiring

In principle, there are two options for implementing blocking with discrete wiring.

Solution 1

Blocking of the binary input occurs via the connection of a voltage to the binary input. The contacts of the **I>** pickups of the feeders are wired in parallel to the binary input block **I>>**; in logical terms, they are OR-combined. Addressing one or more contacts causes blocking of the binary input and thus the **I>>** stage.

This method has a disadvantage: If the overcurrent protection device of the feeder fails temporarily and a short-circuit then occurs on this feeder, blocking cannot occur. The result would be the incorrect switching of the bus-bar. This disadvantage prevents the second method of blocking.

Solution 2

The contacts of the **I>** pickups are connected in series and form a quiescent current loop. NC (normally-closed) contacts are used here. Under normal conditions, a voltage is connected to the binary input. The binary input is assigned the **Pickup I>>** indication.

If the device only has the indication **Blocking I>>**, which is often the case, it is set to 'active without voltage'; this corresponds to an **Inverted Pickup** in logical terms.

As long as no **I>** stage is picked up, the **I>>** stage is enabled.
Tripping occurs for a bus-bar fault.
If, on the other hand, an **I>** stage picks up in the feeders, the contact opens and the loop is broken. No voltage is connected to the binary input at this point. This condition leads to blocking of the **I>>** stage, which is assigned **active without voltage**.

The advantage of this method is that a device can be allowed to fail. In this case, the loop is interrupted, and automatic blocking of the **I>>** stage is achieved. Thus unwanted operation during a short-circuit is prevented.

3.3.3 Reverse Interlocking via IEC 61850

Due to the security relevance of the reverse interlocking, the security philosophy corresponding to **solution 2** should be implemented in the same way as with the discrete wiring via IEC 61850.

Requirements

The following requirements must be fulfilled:

- Reverse interlocking is implemented via the **IEC 61850 GOOSE service**.
- The overcurrent protection (O/C) of feeding offers the function of blocking for the **I>>** stage.
- The pickup of the **I>** stage of the other O/C is transferred with IEC 61850.
- The following GOOSE transfer characteristics are configured:
  First repetition time = 4 ms
  Last repetition time = 1 s.

Implementation

The following implementation is suggested:

The pickup indications **I>** from the individual feeder O/C devices, including the accompanying quality attribute and time stamp, are transferred as a GOOSE message with the transfer characteristics listed above.

These GOOSE messages are all subscribed by the feeding device. The pickup indications and the accompanying status information **valid** is transferred to the internal CFC logic.

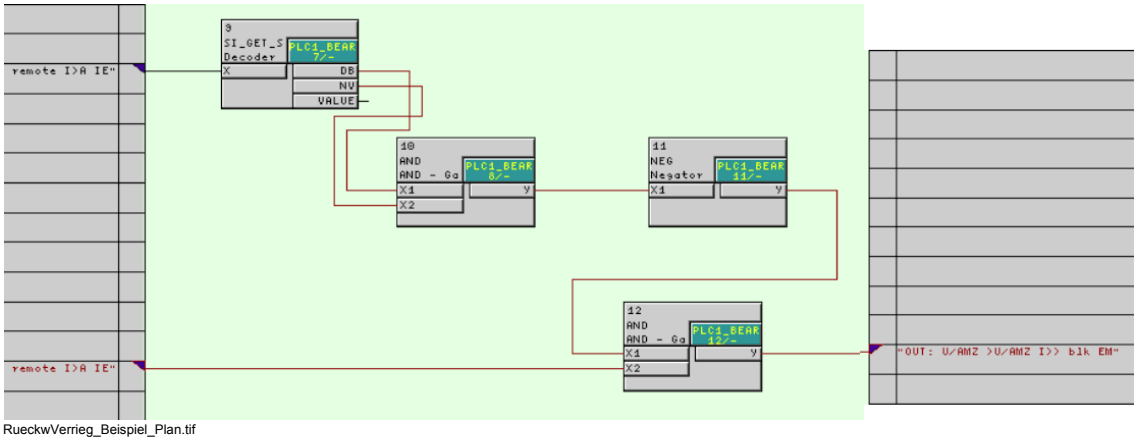
Using a CFC chart, the conventional procedure listed above
(**solution 2**: quiescent current loop) is simulated:

- Each pickup indication **I>** leads to blocking of the **I>>** stage.
- If a feeder O/C fails, the status of the accompanying pickup indication is set to **invalid** after the expiration of the Time Out time (2 x last repetition time) in the feeding device. This status change then also leads to a blocking of the **I>>** pickup stage.

Using this logic, the behaviour of the discrete wiring with **solution 2** is simulated: A device pickup or a failure of a device causes a blocking of the **I>>** stage and prevents unwanted operation of the feeding overcurrent protection.

3.3.4 Diagram: Reverse Interlocking as CFC Chart

Using the CFC chart below, you can simulate the logic of a reverse interlocking in a SIPROTEC device.

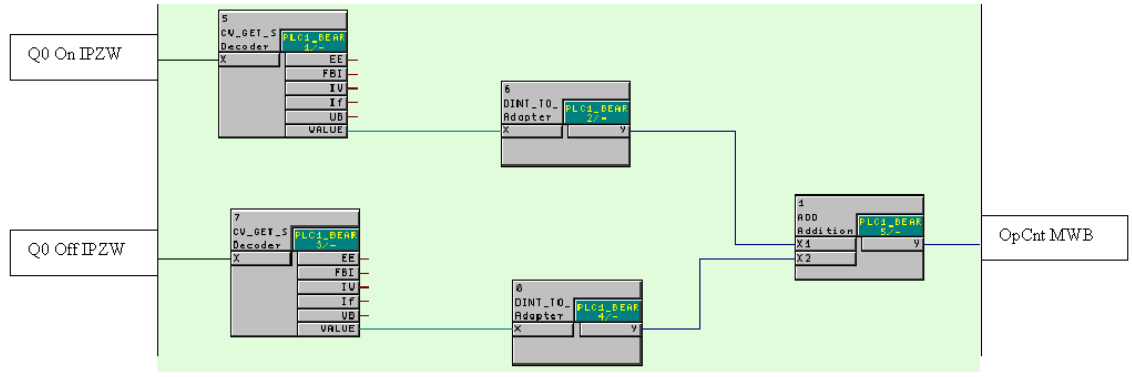


RueckwVerrieg_Beispiel_Plan.tif

Fig. 3-18 Application example of reverse interlocking, CFC chart section

3.4 Counting Operations

Using the CFC chart shown below, you can count the operations in a SIPROTEC device for all operable control devices.



OpCnt_Beispiel_Plan.tif

Fig. 3-19 Application example of counting operations, CFC chart section

Principle of function

Q0 On PMV and **Q0_Off_PMV** are pulse metered values whose sources (via CFC configuration) are the high/low byte of the double-point indication **Q0 ON/OFF**.

The metered values are each separated from the status information with the **CV_GET_STATUS** block.

Via the type converter **DINT_TO_REAL**, the metered values are prepared for addition by the arithmetic block **ADD**.

ADD totals up the number of ON/OFF switches and outputs the sum to the operation counter **OpCnt MVU**.

4 CFC Blocks

Overview

This chapter contains a detailed description of the blocks in DIGSI CFC.

Note

The device version of a SIPROTEC device determines which blocks are available in the device and therefore also in DIGSI CFC.

Note

The CFC blocks in this chapter are displayed with the option **Block width: Wide**:

- ✧ In the CFC Editor, click **Options > Customize > Block/Sheet Bar settings**.
- ✧ Activate the option **Wide** in the displayed window under **Block width** and confirm with **OK**.

Contents

4.1 Data Types

The following data types are available in DIGSI CFC:

Table 4-1 Data types in DIGSI CFC

Type	Presentation in DIGSI	Meaning	Range of values
BOOL	BO	Logical value (binary value)	0 (false), 1 (true)
DINT	DI	Signed integer (32-bit)	-2147483632 to 2147483632 (in DIGSI -2147483648 to 2147483647) **Note:** The limited value range in DIGSI results from the definition of additional values as status information (see table 4-2).
INT	I	Address (e.g. command address)	(Value range specified and entered via DIGSI.)
REAL	R	Floating-point number	$-3.402823466e^{+38}$ to $3.402823466e^{+38}$
STRUCT	ST	Data structure	Various (The contained information can be processed via information status blocks.)
UINT	UI	Unsigned integer (16-bit)	0 to 65535
WORD	W	Bit field (e.g. for double-point indications)	16#0000 to 16#FFFF

Status information for data type DINT

For the DINT data type, additional values are defined as status information in DIGSI CFC as a supplement to the valid value range:

Table 4-2 Status information for the data type DINT in DIGSI CFC

DINT value	Status	Meaning
-2147483647	OVERFLOW_NEG	Value range undershot
-2147483646	NOT_DEFINED	Undefined value
2147483644	LIVE_ZERO	Invalid, e.g. due to a wire break
2147483645	NOT_CALCULATED	No calculation performed
2147483646	INVALID	Value is invalid
2147483647	OVERFLOW_NEG	Value range undershot

Note

In DIGSI, the value range for the DINT data type is **not** limited. When using information in DIGSI CFC, ensure that you do not **inadvertently** set status information.

Status information for data type REAL

The same status information is available for the REAL data type as for the DINT data type. The status values used lie outside the valid value range, however, and therefore cannot be represented as numbers.

Data structures of data type STRUCT

The data structures of the data type STRUCT each consist of two elements which are value (VAL) and status (STAT):

- VAL [BOOL] and STAT [WORD]
  e. g. Output Y of block SI_SET_STATUS
- VAL [WORD] and STAT [WORD]
  e.g. Output Y of block DI_SET_STATUS
- VAL [DINT] and STAT [WORD]
  e.g. Output Y of block MV_SET_STATUS

Note

You can display the data structures via the **Object properties** context menu of the block connection in question.

4.2 Arithmetic

You can process measured values of type REAL and perform calculations with arithmetic blocks.

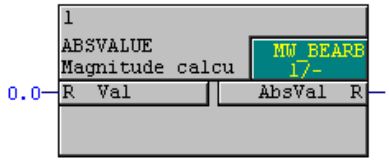
The following arithmetic blocks are available:

- **ABSVALUE** (absolute value)
- **ADD** (addition)
- **DIV** (division)
- **MUL** (multiplication)
- **SQUARE_ROOT**
- **SUB** (subtraction)

4.2.1 ABSVALUE

Function

At the output **AbsVal**, the **Absolute Value** block forms the absolute value of a measured value at the input **Val**.



ABSVALUE.tif

Fig. 4-1 ABSVALUE block

I/O assignment

The **ABSVALUE** block has the following I/O assignment:

Table 4-3 I/O assignment of ABSVALUE block

	Name	Data type	Comment	Default setting
Inputs:	Val	REAL	Value in %	0.0
Outputs:	AbsVal	REAL	Absolute value in %	0.0

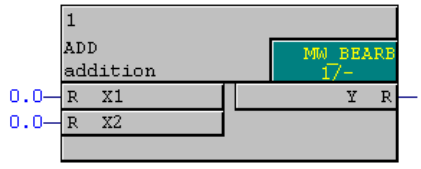
4.2.2 ADD

Function

With the **Addition** block, you can add two values (e.g. measured values) **X1** and **X2**. The result of the addition is output at **Y**.

You can increase the number of inputs to a maximum of 120 via the context menu of the block:

- ✧ Right-click on the block and select **Number of I/Os** in the displayed context menu.
- ✧ Enter the **number** and confirm with **OK**.



ADD.gif

Fig. 4-2 ADD block

I/O assignment

The **ADD** block has the following I/O assignment:

Table 4-4 I/O assignment of ADD block

	Name	Data type	Comment	Default setting
Inputs:	X1	REAL	Addend 1	0.0
	X2	REAL	Addend 2	0.0
Outputs:	Y	REAL	Result of the addition (Y = X1 + X2)	0.0

4.2.3 DIV

Function

With the **Division** block, you can divide the value **X1** by the value **X2** (e.g. measured values). The result of the division is output at **Y**.

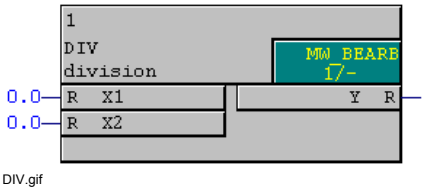


Fig. 4-3 DIV block

I/O assignment

The **DIV** block has the following I/O assignment:

Table 4-5 I/O assignment of DIV block

	Name	Data type	Comment	Default setting
Inputs:	X1	REAL	Dividend	0.0
	X2	REAL	Divisor	0.0
Outputs:	Y	REAL	Result of the division (Y = X1 / X2)	0.0

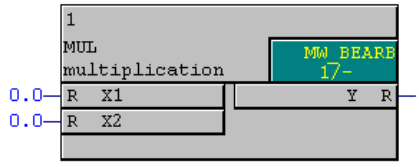
4.2.4 MUL

Function

With the **Multiplication** block, you can multiply two values **X1** and **X2**. The result of the multiplication is output at **Y**.

You can increase the number of inputs to a maximum of 120 via the context menu of the block:

- ✧ Right-click on the block and select **Number of I/Os** in the displayed context menu.
- ✧ Enter the **number** and confirm with **OK**.



MUL.gif

Fig. 4-4 MUL block

I/O assignment

The **MUL** block has the following I/O assignment:

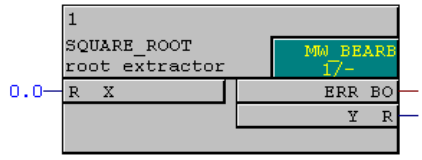
Table 4-6 I/O assignment of MUL block

	Name	Data type	Comment	Default setting
Inputs:	X1	REAL	Factor 1	0.0
	X2	REAL	Factor 2	0.0
Outputs:	Y	REAL	Result of the multiplication (Y = X1 x X2)	0.0

4.2.5 SQUARE_ROOT

Function

The **Square Root Extractor** block can be used to calculate the square root of the radicant **X**. If the value of the radicand **X** is less than 0, the error output **ERR** is set to 1 and the value 0.0 is output at the **Y** output.



SQUARE_ROOT.gif

Fig. 4-5 SQUARE_ROOT block

I/O assignment

The **SQUARE_ROOT** block has the following I/O assignment:

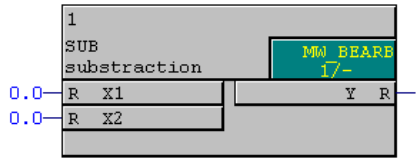
Table 4-7 I/O assignment of the SQUARE_ROOT block

	Name	Data type	Comment	Default setting
Inputs:	X	REAL	Radicant	0.0
Outputs:	ERR	BOOL	Error output (is set to 1, if X < 0)	0
	Y	REAL	Result of the root extraction (Y = SQR(X))	0.0

4.2.6 SUB

Function

With the **Subtraction** block, you can subtract the value **X2** from the value **X1** (e.g. measured values). The result of the subtraction is output at **Y**.



SUB.gif

Fig. 4-6 SUB block

I/O assignment

The **SUB** block has the following I/O assignment:

Table 4-8 I/O assignment of SUB block

	Name	Data type	Comment	Default setting
Inputs:	X1	REAL	Minuend	0.0
	X2	REAL	Subtrahend	0.0
Outputs:	Y	REAL	Result of the subtraction (Y = X1 - X2)	0.0

4.3 Basic Logic

With basic logic blocks, you can link and process logical signals (Boolean).

The following basic logic blocks are available:

- **AND** (AND gate)
- **CONNECT** (connection)
- **DYN_OR** (dynamic OR gate)
- **NAND** (NAND gate)
- **NEG** (negator)
- **NOR** (NOR gate)
- **OR** (OR gate)
- **RISE_DETECT** (rise detector)
- **X_OR** (XOR gate)

4.3.1 AND

Function

The **AND Gate** block combines all input signals by the logical operation AND and generates the output signal from it.

You can increase the number of inputs to a maximum of 120 via the context menu of the block:

✧ Right-click on the block and select **Number of I/Os** in the displayed context menu.

✧ Enter the **number** and confirm with **OK**.

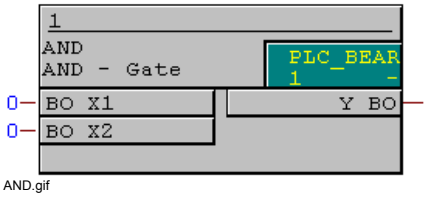


Fig. 4-7 AND block

Note

For the interconnection of **fast signals** with the left border of the CFC chart in the priority classes measured value processing (priority class **MW_BEARB**) and interlocking (priority class **SFS_BEARB**), the following applies:

Boolean inputs are **no** triggers for this level and remain **unnoticed** when the event or signal is shorter than the processing cycle of the processing level.

For the interconnection of **measured values** with the left border of the CFC in the task levels **fast PLC processing** (task level **PLC_BEARB**) and **slow PLC processing** (task level **PLC1_BEARB**), the following applies:

Measuring inputs are **no** triggers for this level and remain **unnoticed**.

I/O assignment

The **AND** block has the following I/O assignment:

Table 4-9 I/O assignment of AND block

	Name	Data type	Comment	Default setting
Inputs:	X1	BOOL	Input	0
	X2	BOOL	Input	0
Outputs:	Y	BOOL	Output (Y = X1 AND X2)	0

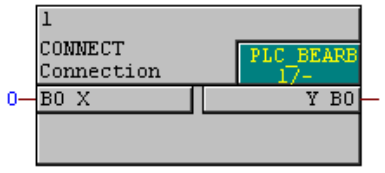
Table 4-10 Truth table of AND block

X1 input	X2 input	Y output
0	0	0
0	1	0
1	0	0
1	1	1

4.3.2 CONNECT

Function

The **Connection** block allows you to interconnect a signal directly between the two borders (that is from a CFC input to a CFC output).
Without this block, a direct interconnection is not possible.



CONNECT.tif

Fig. 4-8 CONNECT block

Note

The **Connection** block works exclusively in the priority classes Fast PLC processing (priority class **PLC_BEARB**), Slow PLC processing (priority class **PLC1_BEARB**) and Interlocking (priority class **SFS_BEARB**).

Note

For the interconnection of **fast signals** with the left border of the CFC chart in the priority classes measured value processing (priority class **MW_BEARB**) and interlocking (priority class **SFS_BEARB**), the following applies:

Boolean inputs are **no** triggers for this level and remain **unnoticed** when the event or signal is shorter than the processing cycle of the processing level.

For the interconnection of **measured values** with the left border of the CFC in the task levels **fast PLC processing** (task level **PLC_BEARB**) and **slow PLC processing** (task level **PLC1_BEARB**), the following applies:

Measuring inputs are **no** triggers for this level and remain **unnoticed**.

I/O assignment

The **CONNECT** block has the following I/O assignment:

Table 4-11 I/O assignment of CONNECT block

	Name	Data type	Comment	Default setting
Inputs:	X	BOOL	Input (to left border)	0
Outputs:	Y	BOOL	Output (to right border)	0

4.3.3 DYN_OR

Function

The **Dynamic OR Gate** block can be used to combine the messages.

In contrast to the logical OR gate, the dynamic OR gate signals **each** incoming input signal change at the output.

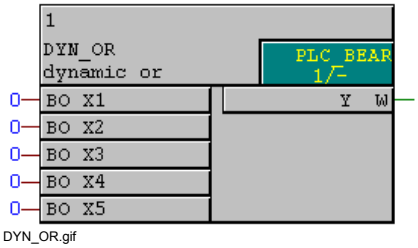
While the device starts up, the double-point indication **intermediate position (00)** is output at the output.

If no signals are connected to the inputs during the chart run, the double-point indication **OFF (01)** is output.

If a signal is active at an output during the chart run, the double-point indication **ON (10)** is output.

If an additional incoming signal is detected at one input after a first signal already active at another input is detected, the output is set for a chart run at **intermediate position (11)**. The signal change is only output as a double-point indication **ON (10)** with the following chart run.

The dynamic OR gate has 5 inputs. If you require more than 5 inputs, you can interconnect several dynamic OR gates in series. To do so, the respective double-point indication must be decoded at the output of the preceding dynamic OR gate using the **DM_DECODE** block and the output signal **ON (10)** must be connected to the input of the next dynamic OR gate.



DYN_OR.gif

Fig. 4-9 DYN_OR block

Note

The **Dynamic OR Gate** block works exclusively in the priority classes Fast PLC processing (priority class **PLC_BEARB**), Slow PLC processing (priority class **PLC1_BEARB**) and Interlocking (priority class **SFS_BEARB**).

Note

For the interconnection of **fast signals** with the left border of the CFC chart in the priority classes measured value processing (priority class **MW_BEARB**) and interlocking (priority class **SFS_BEARB**), the following applies:

Boolean inputs are **no** triggers for this level and remain **unnoticed** when the event or signal is shorter than the processing cycle of the processing level.

For the interconnection of **measured values** with the left border of the CFC in the task levels **fast PLC processing** (task level **PLC_BEARB**) and **slow PLC processing** (task level **PLC1_BEARB**), the following applies:

Measuring inputs are **no** triggers for this level and remain **unnoticed**.

I/O assignment

The **DYN_OR** block has the following I/O assignment:

Table 4-12 I/O assignment of the DYN_OR block

	Name	Data type	Comment	Default setting
Inputs:	X1	BOOL	Input signal	0
	X2	BOOL	Input signal	0
	X3	BOOL	Input signal	0
	X4	BOOL	Input signal	0
	X5	BOOL	Input signal	0
Outputs:	Y	WORD	Output signal as double-point indication	16#0000

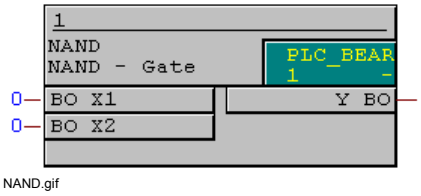
4.3.4 NAND

Function

The **NAND gate** block combines all input signals by the logical operation NAND and generates the output signal from it.

You can increase the number of inputs to a maximum of 120 via the context menu of the block:

- ✧ Right-click on the block and select **Number of I/Os** in the displayed context menu.
- ✧ Enter the **number** and confirm with **OK**.



NAND.gif

Fig. 4-10 NAND block

Note

For the interconnection of **fast signals** with the left border of the CFC chart in the priority classes measured value processing (priority class **MW_BEARB**) and interlocking (priority class **SFS_BEARB**), the following applies:

Boolean inputs are **no** triggers for this level and remain **unnoticed** when the event or signal is shorter than the processing cycle of the processing level.

For the interconnection of **measured values** with the left border of the CFC in the task levels **fast PLC processing** (task level **PLC_BEARB**) and **slow PLC processing** (task level **PLC1_BEARB**), the following applies:

Measuring inputs are **no** triggers for this level and remain **unnoticed**.

I/O assignment

The **NAND** block has the following I/O assignment:

Table 4-13 I/O assignment of NAND block

	Name	Data type	Comment	Default setting
Inputs:	X1	BOOL	Input	0
	X2	BOOL	Input	0
Outputs:	Y	BOOL	Output (Y = NEG(X1 AND X2))	0

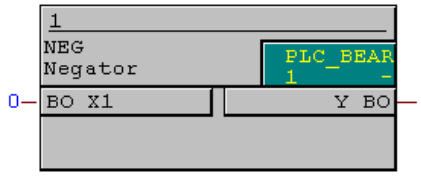
Table 4-14 Truth table of NAND block

X1 input	X2 input	Y output
0	0	1
0	1	1
1	0	1
1	1	0

4.3.5 NEG

Function

The **Negator** block inverts the input signal and generates the output signal from it.



NEG.gif

Fig. 4-11 NEG block

Note

For the interconnection of **fast signals** with the left border of the CFC chart in the priority classes measured value processing (priority class **MW_BEARB**) and interlocking (priority class **SFS_BEARB**), the following applies:

Boolean inputs are **no** triggers for this level and remain **unnoticed** when the event or signal is shorter than the processing cycle of the processing level.

For the interconnection of **measured values** with the left border of the CFC in the task levels **fast PLC processing** (task level **PLC_BEARB**) and **slow PLC processing** (task level **PLC1_BEARB**), the following applies:

Measuring inputs are **no** triggers for this level and remain **unnoticed**.

I/O assignment

The **NEG** block has the following I/O assignment:

Table 4-15 I/O assignment of NEG block

	Name	Data type	Comment	Default setting
Inputs:	X1	BOOL	Input	0
Outputs:	Y	BOOL	Output (Y = NEG(X1))	1

Table 4-16 Truth table of NEG block

X1 input	Y output
0	1
1	0

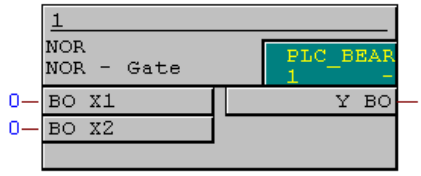
4.3.6 NOR

Function

The **NOR Gate** block combines all input signals by the logical operation NOR and generates the output signal from it.

You can increase the number of inputs to a maximum of 120 via the context menu of the block:

- ✧ Right-click on the block and select **Number of I/Os** in the displayed context menu.
- ✧ Enter the **number** and confirm with **OK**.



NOR.gif

Fig. 4-12 NOR block

Note

For the interconnection of **fast signals** with the left border of the CFC chart in the priority classes measured value processing (priority class **MW_BEARB**) and interlocking (priority class **SFS_BEARB**), the following applies:

Boolean inputs are **no** triggers for this level and remain **unnoticed** when the event or signal is shorter than the processing cycle of the processing level.

For the interconnection of **measured values** with the left border of the CFC in the task levels **fast PLC processing** (task level **PLC_BEARB**) and **slow PLC processing** (task level **PLC1_BEARB**), the following applies:

Measuring inputs are **no** triggers for this level and remain **unnoticed**.

I/O assignment

The **NOR** block has the following I/O assignment:

Table 4-17 I/O assignment of NOR block

	Name	Data type	Comment	Default setting
Inputs:	X1	BOOL	Input	0
	X2	BOOL	Input	0
Outputs:	Y	BOOL	Output (Y = NEG(X1 OR X2))	0

Table 4-18 Truth table of NOR block

X1 input	X2 input	Y output
0	0	1
0	1	0
1	0	0
1	1	0

4.3.7 OR

Function

The **OR Gate** block combines all input signals by the logical operation OR and generates the output signal from it.

You can increase the number of inputs to a maximum of 120 via the context menu of the block:

- ✧ Right-click on the block and select **Number of I/Os** in the displayed context menu.
- ✧ Enter the **number** and confirm with **OK**.

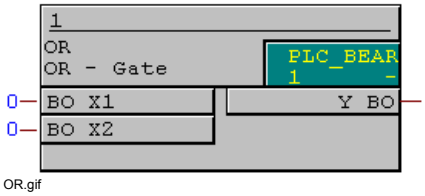


Fig. 4-13 OR block

Note

For the interconnection of **fast signals** with the left border of the CFC chart in the priority classes measured value processing (priority class **MW_BEARB**) and interlocking (priority class **SFS_BEARB**), the following applies:

Boolean inputs are **no** triggers for this level and remain **unnoticed** when the event or signal is shorter than the processing cycle of the processing level.

For the interconnection of **measured values** with the left border of the CFC in the task levels **fast PLC processing** (task level **PLC_BEARB**) and **slow PLC processing** (task level **PLC1_BEARB**), the following applies:

Measuring inputs are **no** triggers for this level and remain **unnoticed**.

I/O assignment

The **OR** block has the following I/O assignment:

Table 4-19 I/O assignment of OR block

	Name	Data type	Comment	Default setting
Inputs:	X1	BOOL	Input	0
	X2	BOOL	Input	0
Outputs:	Y	BOOL	Output (Y = X1 OR X2)	0

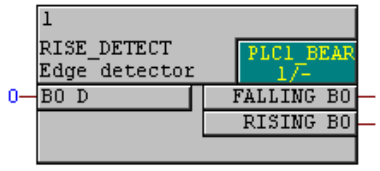
Table 4-20 Truth table of OR block

X1 input	X2 input	Y output
0	0	0
0	1	1
1	0	1
1	1	1

4.3.8 RISE_DETECT

Function

The **Rise Detector** block indicates that the signal at input **D** has a positive or negative rise change at the **RISING** and **FALLING** outputs for the duration of the chart run.



RISE_DETECT.tif

Fig. 4-14 RISE_DETECT block

Note

For the interconnection of **fast signals** with the left border of the CFC chart in the priority classes measured value processing (priority class **MW_BEARB**) and interlocking (priority class **SFS_BEARB**), the following applies:

Boolean inputs are **no** triggers for this level and remain **unnoticed** when the event or signal is shorter than the processing cycle of the processing level.

For the interconnection of **measured values** with the left border of the CFC in the task levels **fast PLC processing** (task level **PLC_BEARB**) and **slow PLC processing** (task level **PLC1_BEARB**), the following applies:

Measuring inputs are **no** triggers for this level and remain **unnoticed**.

I/O assignment

The **RISE_DETECT** block has the following I/O assignment:

Table 4-21 I/O assignment of RISE_DETECT block

	Name	Data type	Comment	Default setting
Inputs:	D	BOOL	Rise detector input	0
Outputs:	FALLING	BOOL	Negative rise change detected	0
	RISING	BOOL	Positive rise change detected	0

4.3.9 X_OR

Function

The **XOR Gate** block combines all input signals by the logical operation XOR (exclusive OR) and generates the output signal from it.

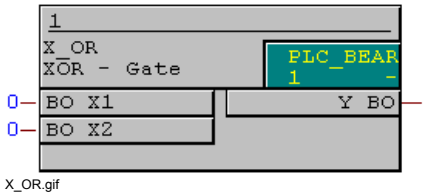


Fig. 4-15 X_OR block

Note

For the interconnection of **fast signals** with the left border of the CFC chart in the priority classes measured value processing (priority class **MW_BEARB**) and interlocking (priority class **SFS_BEARB**), the following applies:

Boolean inputs are **no** triggers for this level and remain **unnoticed** when the event or signal is shorter than the processing cycle of the processing level.

For the interconnection of **measured values** with the left border of the CFC in the task levels **fast PLC processing** (task level **PLC_BEARB**) and **slow PLC processing** (task level **PLC1_BEARB**), the following applies:

Measuring inputs are **no** triggers for this level and remain **unnoticed**.

I/O assignment

The **X_OR** block has the following I/O assignment:

Table 4-22 I/O assignment of the X_OR block

	Name	Data type	Comment	Default setting
Inputs:	X1	BOOL	Input	0
	X2	BOOL	Input	0
Outputs:	Y	BOOL	Output (Y = X1 XOR X2)	0

Table 4-23 Truth table of the X_OR block

X1 input	X2 input	Y output
0	0	0
0	1	1
1	0	1
1	1	0

4.4 Information Status

With information status blocks, you can read, set and process the status of an indication, a measured value or a metered value in the SIPROTEC device.

Possible statuses are:

Table 4-24 Information status overview

CFC input/ output	CFC long text	DIGSI indication logs	IEC 61850	Value in device	Meaning
Indications					
NV	Not updated	NA	oldData	0x0040h	The value is not current. E.g. after start-up, communication failure, not configured (or configured to a non-existing or defective module).
DB	Chatter ON	FS	oscillatory	0x0002h	Chatter block is active (binary input is blocked).
Measured values					
NV	Not updated	NA	oldData	0x0040h	The value is not current. E.g. after start-up, communication failure, not configured (or configured to a non-existing or defective module).
OF	Overflow	UL	overflow	0x0001h	The value is incorrect due to overflow.
UG	Invalid	UG		0x0080h	The value is invalid. E.g. failure of the measured value processing (overload).

Table 4-24 Information status overview

CFC input/ output	CFC long text	DIGSI indication logs	IEC 61850	Value in device	Meaning
Measured values					
NV	Not updated	NA	oldData	0x0040h	The value is not current. E.g. after start-up, communication failure, not configured (or configured to a non-existing or defective module).
IF	Incorrect due to reset		substituted	0x0020h	An initial start-up was executed. The metered value was set to 0.
FBI	Overflow		overflow	0x0100h	The last reading lead to an overflow of the metered value.
IV	Invalid		invalid	0x0200h	The value is invalid. E.g. during start-up. A reset was executed. Metered values may have been lost.
EE	Ext. fault	EE	failure	0x0400h	Image of the external fault input.
UB	Toggle restore bit			0x0800h	The status is changed for each restore.

The following information status blocks are available:

- **CV_GET_STATUS** (decoder)
- **DI_GET_STATUS** (decoder)
- **DI_SET_STATUS** (encoder)
- **MV_GET_STATUS** (decoder)
- **MV_SET_STATUS** (encoder)
- **SI_GET_STATUS** (decoder)
- **SI_SET_STATUS** (encoder)
- **ST_AND** (AND gate) with status
- **ST_NOT** (negator) with status
- **ST_OR** (OR gate) with status

4.4.1 CV_GET_STATUS

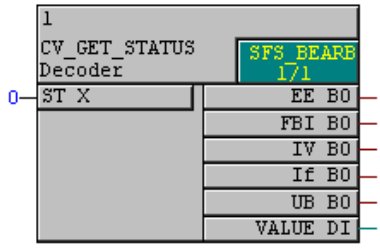
Function

The block **CV_GET_STATUS** decodes a metered value. The outputs provide two pieces of information, the value and the associated status.

This block is intended for interconnection to the **left** border for reading out configured metered values with their status.

Here, the structure of the metered value at input **X** is split into the value of the metered value **VALUE** and the following status information:

- **EE** (external error)
- **FBI** (overflow)
- **IV** (invalid)
- **IF** (incorrect due to reset)
- **UB** (toggle restore bit)



CV_GET_STATUS.tif

Fig. 4-16 CV_GET_STATUS block

I/O assignment

The **CV_GET_STATUS** block has the following I/O assignment:

Table 4-25 I/O assignment of CV_GET_STATUS block

	Name	Data type	Comment	Default setting
Inputs:	X	ST	Metered value with status	(0)
Outputs:	EE	BOOL	External error	0
	FBI	BOOL	Overflow	0
	IV	BOOL	Invalid	0
	IF	BOOL	Incorrect due to reset	0
	UB	BOOL	Toggle restore bit	0
	VALUE	DINT	Metered value	0

4.4.2 DI_GET_STATUS

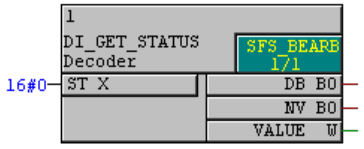
Function

The block **DI_GET_STATUS** decodes the status of a double-point indication. The outputs provide two pieces of information, the value and the associated status.

This block is intended for interconnection to the **left** border for processing double-point indications and their status.

Here, the structure of the double-point indication at input **X** is split into the value of the double-point indication **VALUE** and the following status information:

- **DB** (chatter block)
- **NV** (not updated)



DI_GET_STATUS.tif

Fig. 4-17 DI_GET_STATUS block

I/O assignment

The **DI_GET_STATUS** block has the following I/O assignment:

Table 4-26 I/O assignment of DI_GET_STATUS block

	Name	Data type	Comment	Default setting
Inputs:	X	ST	double-point indication with status	(0)
Outputs:	DB	BOOL	Chatter ON	0
	NV	BOOL	Not updated	0
	VALUE	WORD	Double point indication	16#0000

4.4.3 DI_SET_STATUS

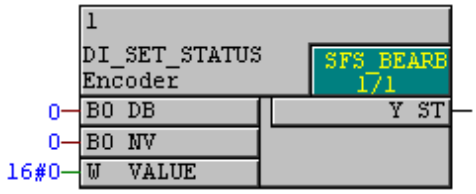
Function

The **DI_SET_STATUS** block generates a double-point indication with status.

This block is intended for interconnection with the **right** border to set double-point indications with status.

Here, the structure of the double-point indication at output **Y** is generated depending on the value of the double-point indication **VALUE** and the following status information:

- **DB** (chatter block)
- **NV** (not updated)



DI_SET_STATUS.tif

Fig. 4-18 DI_SET_STATUS block

I/O assignment

The **DI_SET_STATUS** block has the following I/O assignment:

Table 4-27 I/O assignment of DI_SET_STATUS block

	Name	Data type	Comment	Default setting
Inputs:	DB	BOOL	Chatter ON	0
	NV	BOOL	Not updated	0
	VALUE	WORD	double-point indication	16#0000
Outputs:	Y	ST	double-point indication with status	(0)

Note

VALUE input: Only the value of the information is transferred. The status of this information is not taken into account.

Note

Y output: If the output is linked to the right border, i.e. this information is processed further in the device, the following special points apply for processing in the device:

Status NV = TRUE
Causes a block. I.e. if the output is set to **not current**, all other changes of the value and status are ignored.

Status NV
A status change of **not updated** which comes or goes is only logged in the spontaneous indications and not in the operating indications.

Status DB
A status change **chatter block** which comes or goes is only used for checking the following processes. The **chatter block** function is not activated with this.

All changes are processed within a chart (e.g. forwarded to decoder **DI_SET_STATUS**).

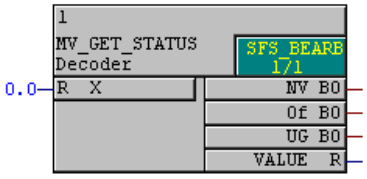
4.4.4 MV_GET_STATUS

Function

The block **MV_GET_STATUS** decodes a value (e. g. measured value). The outputs provide two pieces of information, the value and the associated status.

Here, the measured value at the **X** input is checked for the following status information and then output unchanged at the **VALUE** output.

- **NV** (not updated)
- **OF** (overflow)
- **UG** (invalid)



MV_GET_STATUS.tif

Fig. 4-19 MV_GET_STATUS block

I/O assignment

The **MV_GET_STATUS** block has the following I/O assignment:

Table 4-28 I/O assignment of MV_GET_STATUS block

	Name	Data type	Comment	Default setting
Inputs:	X	REAL	Measured value with status	0.0
Outputs:	NV	BOOL	Not updated	0
	OF	BOOL	Overflow	0
	UG	BOOL	Invalid	0
	VALUE	REAL	Measured value	0.0

4.4.5 MV_SET_STATUS

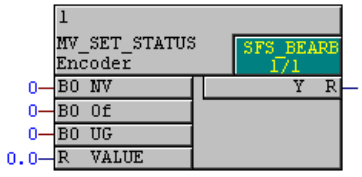
Function

The **MV_SET_STATUS** block sets the status in a measured value.

Here, the measured value **VALUE** is replaced by the respective status information **NV**, **OF**, **UG** and output at the **Y** output:

- **NV** (not updated)
- **OF** (overflow)
- **UG** (invalid)

The replacement occurs with the following priority: **UG**, **NV**, **OF**.



MV_SET_STATUS.tif

Fig. 4-20 MV_SET_STATUS block

I/O assignment

The **MV_SET_STATUS** block has the following I/O assignment:

Table 4-29 I/O assignment of MV_SET_STATUS block

	Name	Data type	Comment	Default setting
Inputs:	NV	BOOL	Not updated	0
	OF	BOOL	Overflow	0
	UG	BOOL	Invalid	0
	VALUE	REAL	Measured value	0.0
Outputs:	Y	REAL	Measured value with status	(0)

Note

VALUE input: Only the value of the information is transferred. The status of this information is not taken into account.

Note

Y output: If the output is linked to the right border, i.e. this information is processed further in the device, the following special points apply for processing in the device:

Status NV = TRUE
Causes a block. I.e. if the output is set to **not current**, all other changes of the value and status are ignored.

Status NV
A status change of **not updated** which comes or goes is only logged in the spontaneous indications and not in the operating indications.

Status DB
A status change **chatter block** which comes or goes is only used for checking the following processes. The **chatter block** function is not activated with this.

All changes are processed within a chart (e.g. forwarded to decoder **DI_SET_STATUS**).

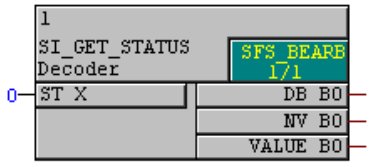
4.4.6 SI_GET_STATUS

Function

The block **SI_GET_STATUS** decodes a single-point indication (e. g. external single-point indication). The outputs provide two pieces of information, the value and the associated status.

Here, the structure of the single point indication at input **X** is split into the value of the single point indication **VALUE** and the following status information:

- **DB** (chatter block)
- **NV** (not updated)



SI_GET_STATUS.tif

Fig. 4-21 SI_GET_STATUS block

Note

For the interconnection of **fast signals** with the left border of the CFC chart in the priority classes measured value processing (priority class **MW_BEARB**) and interlocking (priority class **SFS_BEARB**), the following applies:

Boolean inputs are **no** triggers for this level and remain **unnoticed** when the event or signal is shorter than the processing cycle of the processing level.

For the interconnection of **measured values** with the left border of the CFC in the task levels **fast PLC processing** (task level **PLC_BEARB**) and **slow PLC processing** (task level **PLC1_BEARB**), the following applies:

Measuring inputs are **no** triggers for this level and remain **unnoticed**.

I/O assignment

The **SI_GET_STATUS** block has the following I/O assignment:

Table 4-30 I/O assignment of SI_GET_STATUS block

	Name	Data type	Comment	Default setting
Inputs:	X	ST	Single point indication with status	(0)
Outputs:	DB	BOOL	Chatter ON	0
	NV	BOOL	Not updated	0
	VALUE	BOOL	Single point indication	16#0000

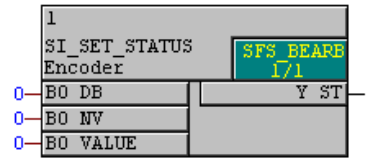
4.4.7 SI_SET_STATUS

Function

The **SI_SET_STATUS** block generates a single point indication with status.

Here, the structure of the single point indication at output **Y** is generated depending on the value of the single point indication **VALUE** and the following status information:

- **DB** (chatter block)
- **NV** (not updated)



SI_SET_STATUS.tif

Fig. 4-22 SI_SET_STATUS block

I/O assignment

The **SI_SET_STATUS** block has the following I/O assignment:

Table 4-31 I/O assignment of SI_SET_STATUS block

	Name	Data type	Comment	Default setting
Inputs:	DB	BOOL	Chatter ON	0
	NV	BOOL	Not updated	0
	VALUE	BOOL	Single point indication	0
Outputs:	Y	ST	Single point indication with status	(0)

Note

Note the following for the **VALUE** input:

Only the value of the information is transferred. The status of this information is not taken into account.

Note

Note the following for the **Y** output:

If the output is linked to the right border, i.e. this information is processed further in the device, the following special points apply for processing in the device:

Status NV = TRUE
Causes a block. I.e. if the output is set to **not current**, all other changes of the value and status are ignored.

Status NV
A status change of **not updated** which comes or goes is only logged in the spontaneous indications and not in the operating indications.

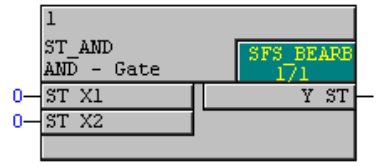
Status DB
A status change **chatter block** which comes or goes is only used for checking the following processes. The **chatter block** function is not activated with this.

All changes are processed within a chart (e.g. forwarded to decoder **DI_SET_STATUS**).

4.4.8 ST_AND

Function

The **ST_AND** block combines two single point indications and their status (NV bit) by the logical operation AND and generates the output signal from it.



ST_AND.tif

Fig. 4-23 ST_AND block

Note

For the interconnection of **fast signals** with the left border of the CFC chart in the priority classes measured value processing (priority class **MW_BEARB**) and interlocking (priority class **SFS_BEARB**), the following applies:

Boolean inputs are **no** triggers for this level and remain **unnoticed** when the event or signal is shorter than the processing cycle of the processing level.

For the interconnection of **measured values** with the left border of the CFC in the task levels **fast PLC processing** (task level **PLC_BEARB**) and **slow PLC processing** (task level **PLC1_BEARB**), the following applies:

Measuring inputs are **no** triggers for this level and remain **unnoticed**.

I/O assignment

The **ST_AND** block has the following I/O assignment:

Table 4-32 I/O assignment of ST_AND block

	Name	Data type	Comment	Default setting
Inputs:	X1	ST	Input with status	(0)
	X2	ST	Input with status	(0)
Outputs:	Y	ST	Output with status	(0)

Messages and status

If all signals are connected to the inputs the output will be set to **INCOMING**. The status of the output message is the OR-combined status of all inputs.

Note

If at least one of the inputs is **OUTGOING** and the status of the signal is **Current** (= 0), the status of the output is set to **Current** (= 0).

Status values

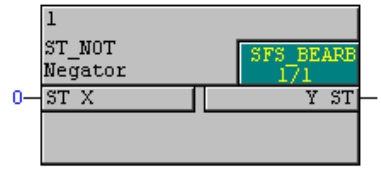
The **ST_AND** block considers the following status values only:

- **NV** (Not updated)
- **DB** (Chatter ON)

4.4.9 ST_NOT

Function

The **ST_NOT** block inverts the single point indication with status and forms the output signal with status from it.



ST_NOT.tif

Fig. 4-24 ST_NOT block

Note

For the interconnection of **fast signals** with the left border of the CFC chart in the priority classes measured value processing (priority class **MW_BEARB**) and interlocking (priority class **SFS_BEARB**), the following applies:

Boolean inputs are **no** triggers for this level and remain **unnoticed** when the event or signal is shorter than the processing cycle of the processing level.

For the interconnection of **measured values** with the left border of the CFC in the task levels **fast PLC processing** (task level **PLC_BEARB**) and **slow PLC processing** (task level **PLC1_BEARB**), the following applies:

Measuring inputs are **no** triggers for this level and remain **unnoticed**.

I/O assignment

The **ST_NOT** block has the following I/O assignment:

Table 4-33 I/O assignment of ST_NOT block

	Name	Data type	Comment	Default setting
Inputs:	X	ST	Input with status	(0)
Outputs:	Y	ST	Output with status	(0)

Messages and status

The block **ST_NOT** negates the value of a message at the input and forwards it at the block output. The status information is preserved.

Status values

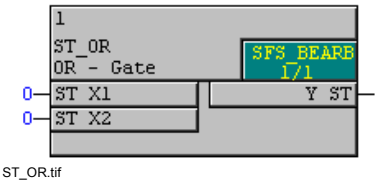
The **ST_NOT** block considers the following status values only:

- **NV** (Not updated)
- **DB** (Chatter ON)

4.4.10 ST_OR

Function

The **ST_OR gate** block combines two single point indications and their status (NV bit) by the logical operation OR and generates the output signal from it.



ST_OR.tif

Fig. 4-25 ST_OR block

Note

For the interconnection of **fast signals** with the left border of the CFC chart in the priority classes measured value processing (priority class **MW_BEARB**) and interlocking (priority class **SFS_BEARB**), the following applies:

Boolean inputs are **no** triggers for this level and remain **unnoticed** when the event or signal is shorter than the processing cycle of the processing level.

For the interconnection of **measured values** with the left border of the CFC in the task levels **fast PLC processing** (task level **PLC_BEARB**) and **slow PLC processing** (task level **PLC1_BEARB**), the following applies:

Measuring inputs are **no** triggers for this level and remain **unnoticed**.

I/O assignment

The **ST_OR** block has the following I/O assignment:

Table 4-34 I/O assignment of ST_OR block

	Name	Data type	Comment	Default setting
Inputs:	X1	ST	Input with status	(0)
	X2	ST	Input with status	(0)
Outputs:	Y	ST	Output with status	(0)

Messages and status

When the first signal is connected to the block, the output is set to **INCOMING**. If the last signal connected to the block is **OUTGOING**, the output will also be set to **OUTGOING**.

The status of the output message is the OR-combined status of all inputs.

Note

If at least one of the inputs is **OUTGOING** and the status of the signal is **Current** (= 0), the status of the output is set to **Current** (= 0).

Status values

The **ST_AND** block considers the following status values only:

- **NV** (Not updated)
- **DB** (Chatter ON)

4.5 Memory

With the memory blocks, you can save logical states of the device beyond the chart run.

The following memory blocks are available:

- **D_FF** (D flipflop)
- **D_FF_MEMO** (status memory for restart)
- **RS_FF** (RS flipflop)
- **RS_FF_MEMO** (status memory for restart)
- **SR_FF** (SR flipflop)
- **SR_FF_MEMO** (status memory for restart)
- **MEMORY** (data memory)

4.5.1 D_FF

Function

With the **D Flipflop** block, the signal at the **D** input is transferred to the **Q** output with the rising pulse edge of the **Clk** input. The value at the **Q** output remains intact until the next rising pulse flank is detected at the **Clk** input.

Note

The signal at the **Q** output can be set to standard during initial start-up and saved prior to each rerun and then restored. To do so, you must use the **D_FF_MEMO** block.

D_FF.tif

Fig. 4-26 D_FF block

Note

The **D Flipflop** block works exclusively in the priority classes Fast PLC processing (priority class **PLC_BEARB**), Slow PLC processing (priority class **PLC1_BEARB**) and Interlocking (priority class **SFS_BEARB**).

Note

For the interconnection of **fast signals** with the left border of the CFC chart in the priority classes measured value processing (priority class **MW_BEARB**) and interlocking (priority class **SFS_BEARB**), the following applies:

Boolean inputs are **no** triggers for this level and remain **unnoticed** when the event or signal is shorter than the processing cycle of the processing level.

For the interconnection of **measured values** with the left border of the CFC in the task levels **fast PLC processing** (task level **PLC_BEARB**) and **slow PLC processing** (task level **PLC1_BEARB**), the following applies:

Measuring inputs are **no** triggers for this level and remain **unnoticed**.

I/O assignment

The **D_FF** block has the following I/O assignment:

Table 4-35 I/O assignment of the D_FF block

	Name	Data type	Comment	Default setting
Inputs:	Clk	BOOL	Clock	0
	D	BOOL	Data	0
Outputs:	Q	BOOL	Output	0

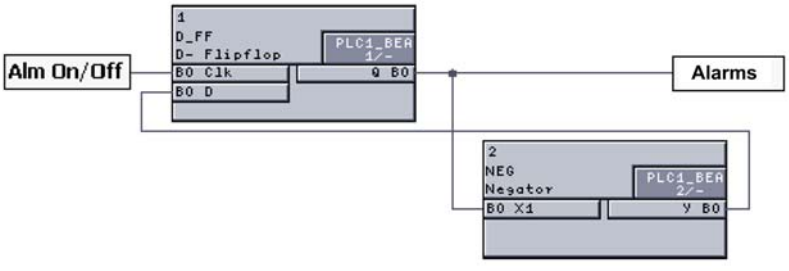
Table 4-36 Truth table of D_FF block

Clk input	D input	Q output
0 or 1	0	Q_{n-1}
Change from 0 to 1	0	0
0 or 1	1	Q_{n-1}
Change from 0 to 1	1	1

Application example

Using the CFC chart shown below, you can switch a horn on and off alternately with a single function key:

- ✧ Insert an internal single point indication **horn on/off**, which you configure to a function key (source) and the CFC (destination), in the configuration.
- ✧ A new output indication called **horn**, formed in the CFC (source), is linked to the horn via a binary output.



D_FF_Beispiel.tif

Fig. 4-27 Application example of D_FF block, CFC chart section

4.5.2 D_FF_MEMO

Function

The **D Flipflop with State Memory** operates like the **D_FF** block:

The signal at the **D** input is transmitted to the **Q** output with the rising pulse edge at the **Clk** input. The value at the **Q** output remains intact until the next rising pulse flank is detected at the **Clk** input.

In addition, the signal at the **Q** output is assigned by default with the value of the **INIT** input during initial start-up, salvaged before each restart and then restored.

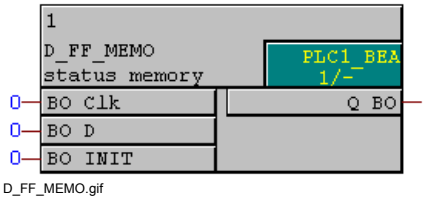


Fig. 4-28 D_FF_MEMO block

Note

The maximum permissible number of **MEMORY**, **RS_FF_MEMO**, **SR_FF_MEMO**, **D_FF_MEMO** and **COUNTER** blocks depends on the available non-volatile memory and is monitored by the CFC compiler. Observe the technical data in the device manual of the SIPROTEC device which you want to use.

The maximum permissible number is checked during the compilation of the CFC chart. Consistency errors are indicated when a fault occurs. The exceeding of the resource is indicated in the displayed compilation log.

Note

For the interconnection of **fast signals** with the left border of the CFC chart in the priority classes measured value processing (priority class **MW_BEARB**) and interlocking (priority class **SFS_BEARB**), the following applies:

Boolean inputs are **no** triggers for this level and remain **unnoticed** when the event or signal is shorter than the processing cycle of the processing level.

For the interconnection of **measured values** with the left border of the CFC in the task levels **fast PLC processing** (task level **PLC_BEARB**) and **slow PLC processing** (task level **PLC1_BEARB**), the following applies:

Measuring inputs are **no** triggers for this level and remain **unnoticed**.

I/O assignment

The **D_FF_MEMO** block has the following I/O assignment:

Table 4-37 I/O assignment of the D_FF_MEMO block

	Name	Data type	Comment	Default setting
Inputs:	Clk	BOOL	Clock	0
	D	BOOL	Data	0
	INIT	BOOL	Data default setting **Note:** Default settings only apply for the initial start-up.	0
Outputs:	Q	BOOL	Output	Initial start-up: Q = INIT Resume: Q = Q when the device is switched off

Table 4-38 Truth table for D_FF_MEMO block

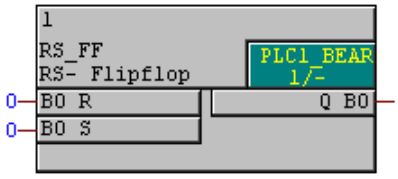
Clk input	D input	Q output
0 or 1	0	Q_{n-1}
Change from 0 to 1	0	0
0 or 1	1	Q_{n-1}
Change from 0 to 1	1	1

4.5.3 RS_FF

Function

With the **RS Flipflop** block, a 1 at the **S** input sets the **Q** output to the value 1. This value remains intact until **R** accepts the value 1.

With this block, the **R** input is dominant, i.e. the **Q** output is also reset when a 1 is active at the **S** and **R** inputs.



RS_FF.tif

Fig. 4-29 RS_FF block

Note

The **RS Flipflop** block works exclusively in the priority classes Fast PLC processing (priority class **PLC_BEARB**), Slow PLC processing (priority class **PLC1_BEARB**) and Interlocking (priority class **SFS_BEARB**).

Note

For the interconnection of **fast signals** with the left border of the CFC chart in the priority classes measured value processing (priority class **MW_BEARB**) and interlocking (priority class **SFS_BEARB**), the following applies:

Boolean inputs are **no** triggers for this level and remain **unnoticed** when the event or signal is shorter than the processing cycle of the processing level.

For the interconnection of **measured values** with the left border of the CFC in the task levels **fast PLC processing** (task level **PLC_BEARB**) and **slow PLC processing** (task level **PLC1_BEARB**), the following applies:

Measuring inputs are **no** triggers for this level and remain **unnoticed**.

I/O assignment

The **RS_FF** block has the following I/O assignment:

Table 4-39 I/O assignment of the RS_FF block

	Name	Data type	Comment	Default setting
Inputs:	R	BOOL	Reset	0
	S	BOOL	Set	0
Outputs:	Q	BOOL	Output	0

Table 4-40 Truth table of RS_FF block

R input	**S input**	**Q_n output**
0	0	Q_{n-1}
0	1	1
1	0	0
1	1	0

4.5.4 RS_FF_MEMO

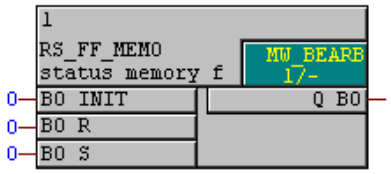
Function

The **RS Flipflop with State Memory** operates like the **RS_FF** block:

A 1 at the **S** input sets the **Q** output to the value 1. This value remains intact until **R** accepts the value 1.

With this block, the **R** input is dominant, i.e. the **Q** output is also reset when a 1 is active at the **S** and **R** inputs.

In addition, the signal at the **Q** output is assigned by default with the value of the **INIT** input during initial start-up, salvaged before each restart and then restored.



RS_FF_MEMO.tif

Fig. 4-30 RS_FF_MEMO block

Note

The maximum permissible number of **MEMORY**, **RS_FF_MEMO**, **SR_FF_MEMO**, **D_FF_MEMO** and **COUNTER** blocks depends on the available non-volatile memory and is monitored by the CFC compiler. Observe the technical data in the device manual of the SIPROTEC device which you want to use.

The maximum permissible number is checked during the compilation of the CFC chart. Consistency errors are indicated when a fault occurs. The exceeding of the resource is indicated in the displayed compilation log.

Note

For the interconnection of **fast signals** with the left border of the CFC chart in the priority classes measured value processing (priority class **MW_BEARB**) and interlocking (priority class **SFS_BEARB**), the following applies:

Boolean inputs are **no** triggers for this level and remain **unnoticed** when the event or signal is shorter than the processing cycle of the processing level.

For the interconnection of **measured values** with the left border of the CFC in the task levels **fast PLC processing** (task level **PLC_BEARB**) and **slow PLC processing** (task level **PLC1_BEARB**), the following applies:

Measuring inputs are **no** triggers for this level and remain **unnoticed**.

I/O assignment

The **RS_FF_MEMO** block has the following I/O assignment:

Table 4-41 I/O assignment of the RS_FF_MEMO block

	Name	Data type	Comment	Default setting
Inputs:	INIT	BOOL	Data default setting **Note:** Default settings only apply for the initial start-up.	0
	R	BOOL	Reset	0
	S	BOOL	Set	0
Outputs:	Q	BOOL	Output	Initial start-up: Q = INIT Resume: Q = Q when the device is switched off

Table 4-42 Truth table of RS_FF_MEMO block

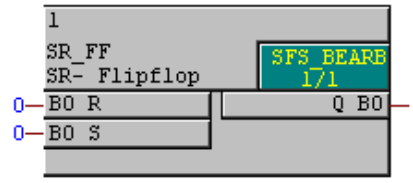
R input	S input	Q_n output
0	0	Q_{n-1}
0	1	1
1	0	0
1	1	0

4.5.5 SR_FF

Function

With the **SR Flipflop** block, a 1 at the **S** input sets the **Q** output to the value 1. This value remains intact until **R** accepts the value 1.

With this block, the **S** input is dominant, i.e. the **Q** output is also set when a 1 is active at the **S** and **R** inputs.



SR_FF.tif

Fig. 4-31 SR_FF block

Note

The **SR Flipflop** block works exclusively in the priority classes Fast PLC processing (priority class **PLC_BEARB**), Slow PLC processing (priority class **PLC1_BEARB**) and Interlocking (priority class **SFS_BEARB**).

Note

For the interconnection of **fast signals** with the left border of the CFC chart in the priority classes measured value processing (priority class **MW_BEARB**) and interlocking (priority class **SFS_BEARB**), the following applies:

Boolean inputs are **no** triggers for this level and remain **unnoticed** when the event or signal is shorter than the processing cycle of the processing level.

For the interconnection of **measured values** with the left border of the CFC in the task levels **fast PLC processing** (task level **PLC_BEARB**) and **slow PLC processing** (task level **PLC1_BEARB**), the following applies:

Measuring inputs are **no** triggers for this level and remain **unnoticed**.

I/O assignment

The **SR_FF** block has the following I/O assignment:

Table 4-43 I/O assignment of the SR_FF block

	Name	Data type	Comment	Default setting
Inputs:	S	BOOL	Set	0
	R	BOOL	Reset	0
Outputs:	Q	BOOL	Output	0

Table 4-44 Truth table of SR_FF block

R input	**S input**	**Q_n output**
0	0	Q_{n-1}
0	1	1
1	0	0
1	1	1

4.5.6 SR_FF_MEMO

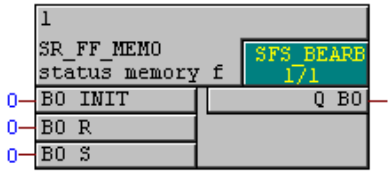
Function

The **SR Flipflop with State Memory** operates like the **SR_FF** block:

A 1 at the **S** input sets the **Q** output to the value 1. This value remains intact until **R** accepts the value 1.

With this block, the **S** input is dominant, i.e. the **Q** output is also set when a 1 is active at the **S** and **R** inputs.

In addition, the signal at the **Q** output is assigned by default with the value of the **INIT** input during initial start-up, salvaged before each restart and then restored.



SR_FF_MEMO.tif

Fig. 4-32 SR_FF_MEMO block

Note

The maximum permissible number of **MEMORY**, **RS_FF_MEMO**, **SR_FF_MEMO**, **D_FF_MEMO** and **COUNTER** blocks depends on the available non-volatile memory and is monitored by the CFC compiler. Observe the technical data in the device manual of the SIPROTEC device which you want to use.

The maximum permissible number is checked during the compilation of the CFC chart. Consistency errors are indicated when a fault occurs. The exceeding of the resource is indicated in the displayed compilation log.

Note

For the interconnection of **fast signals** with the left border of the CFC chart in the priority classes measured value processing (priority class **MW_BEARB**) and interlocking (priority class **SFS_BEARB**), the following applies:

Boolean inputs are **no** triggers for this level and remain **unnoticed** when the event or signal is shorter than the processing cycle of the processing level.

For the interconnection of **measured values** with the left border of the CFC in the task levels **fast PLC processing** (task level **PLC_BEARB**) and **slow PLC processing** (task level **PLC1_BEARB**), the following applies:

Measuring inputs are **no** triggers for this level and remain **unnoticed**.

I/O assignment

The **SR_FF_MEMO** block has the following I/O assignment:

Table 4-45 I/O assignment of the RS_FF_MEMO block

	Name	Data type	Comment	Default setting
Inputs:	INIT	BOOL	Data default setting **Note:** Default settings only apply for the initial start-up.	0
	S	BOOL	Set	0
	R	BOOL	Reset	0
Outputs:	Q	BOOL	Output	Initial start-up: Q = INIT Resume: Q = Q when the device is switched off

Table 4-46 Truth table of SR_FF_MEMO block

R input	**S input**	**Q_n output**
0	0	Q_{n-1}
0	1	1
1	0	0
1	1	1

4.5.7 MEMORY

Function

The signal at the **D** input of the **MEMORY** block is transmitted to the **Q** output with the rising pulse edge at the **Clk** input. The value at the **Q** output remains intact until the next rising pulse flank is detected at the **Clk** input.

In addition, the signal at the **Q** output is assigned by default with the value of the **INIT** input during initial start-up, salvaged before each restart and then restored.

Note

The maximum permissible number of **MEMORY**, **RS_FF_MEMO**, **SR_FF_MEMO**, **D_FF_MEMO** and **COUNTER** blocks depends on the available non-volatile memory and is monitored by the CFC compiler. Observe the technical data in the device manual of the SIPROTEC device which you want to use.

The maximum permissible number is checked during the compilation of the CFC chart. Consistency errors are indicated when a fault occurs. The exceeding of the resource is indicated in the displayed compilation log.

Note

For the interconnection of **fast signals** with the left border of the CFC chart in the priority classes measured value processing (priority class **MW_BEARB**) and interlocking (priority class **SFS_BEARB**), the following applies:

Boolean inputs are **no** triggers for this level and remain **unnoticed** when the event or signal is shorter than the processing cycle of the processing level.

For the interconnection of **measured values** with the left border of the CFC in the task levels **fast PLC processing** (task level **PLC_BEARB**) and **slow PLC processing** (task level **PLC1_BEARB**), the following applies:

Measuring inputs are **no** triggers for this level and remain **unnoticed**.

I/O assignment

The **MEMORY** block has the following I/O assignment:

Table 4-47 I/O assignment of the MEMORY block

	Name	Data type	Comment	Default setting
Inputs:	Clk	BOOL	Clock	0
	D	REAL	Data	0.0
	INIT	REAL	Data default setting **Note:** Default settings only apply for the initial start-up.	0.0
Outputs:	Q	REAL	Output	Initial start-up: Q = INIT Resume: Q = Q when the device is switched off

4.6 Control Commands

With control command blocks, you can trigger or cancel control commands or receive information on switching commands.

The following control command blocks are available:

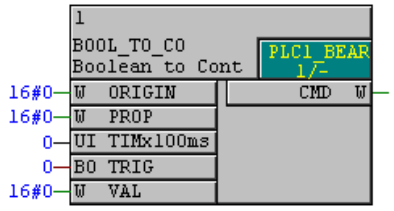
- **BOOL_TO_CO** (Boolean to command conversion)
- **BOOL_TO_IC** (Boolean to internal single point indication conversion)
- **CMD_CANCEL** (cancel command)
- **CMD_CHAIN** (switching sequence)
- **CMD_INF** and **CMD_INF_EXE** (command information)
- **LOOP** (signal feedback)

4.6.1 BOOL_TO_CO

Function

The **Boolean to Command** block generates a switching command.

The switching command is defined via the parameters **ORIGIN**, **PROP**, **VAL** and **TIME** and trigered via a signal at the **TRIG** input.



BOOL_TO_CO.tif

Fig. 4-33 BOOL_TO_CO block

Note

The **Boolean to Command** block functions exclusively in the priority classes Fast PLC processing (priority class **PLC_BEARB**) and Slow PLC processing (priority class **PLC1_BEARB**) and for immediate switching with the right border.

I/O assignment

The **BOOL_TO_CO** block has the following I/O assignment:

Table 4-48 I/O assignment of the BOOL_TO_CO block

	Name	Data type	Comment	Default setting
Inputs:	PROP	WORD	Command properties	16#0000
	TRIG	BOOL	Trigger	0
	ORIGIN	WORD	Command source	16#0000
	VAL	WORD	Switching direction	16#0000
	TIMx100ms	UINT	Output time (resolution 100 ms; valid value range 0 to 32767)	0
Outputs:	CMD	WORD	Command	16#0000

PROP settings

The following values can be set in the **PROP** (command properties) settings:

Table 4-49 Values for the BOOL_TO_CO block, PROP settings

Value (hexadecimal)	Value (decimal)	Command properties
00	0	No deblocking
01	1	Deblocking setpoint=actual
02	2	Deblocking of station blocking
04	4	Deblocking of bay blocking
08	8	(ignored)
10	16	(ignored)
20	32	Deblocking of double-command blocking
40	64	Deblocking of protection device blocking
80	128	Deblocking when station or bay blocking is set

If **PROP** is set to 00 (hexadecimal), the original command properties configured in the DIGSI configuration matrix apply.

Note

If you would like to define several command properties at the same time, you must add up the hexadecimal values individually.

ORIGIN settings

The **ORIGIN** settings is used to assign a certain command source to the switching command you have generated. If you assign for example the value 01 to **ORIGIN**, the command will be treated during execution like a local command (at device itself).

The following values can be set in the **ORIGIN** (command source) parameter:

Table 4-50 Values for the BOOL_TO_CO block, ORIGIN settings

Value	Meaning
00	Automatically generated command
01	Integrated operation, device panel control (e.g. at device itself)
02	DIGSI, SICAM, local control (e.g. DIGSI Remote, DIGSI on bus)
03	Telecontrol centers, remote control (e.g. WinCC, telecontrol station)

VAL settings

The following values can be set in the **VAL** (actuating direction) settings:

Table 4-51 Values for the BOOL_TO_CO block, VAL settings

Value	Meaning
01	Off (01)
02	On (10)

TIMx100ms settings

In the **TIMx100ms** (output time) parameter, you set the output time in increments of 100 milliseconds.

If **TIMx100ms** is set to 0, the original output time of the command configured in the DIGSI configuration matrix applies.

The **TIMx100ms** parameter (output time) is applied during the start-up of the SIPROTEC device. It **cannot** be changed during operation.

Only times less than 3,276.8 seconds are accepted. Larger values are limited to this value.

Output IE

The **IE** output must be interconnected with the command directly at the right border.

At the **IE** output the block shows the following behaviour:
If the signal changes from 0 to 1 at the **TRIG** input, the command generated by the **PROP**, **ORIGIN**, **VAL** and **TIME** settings are switched through to **CMD**.

CMD output

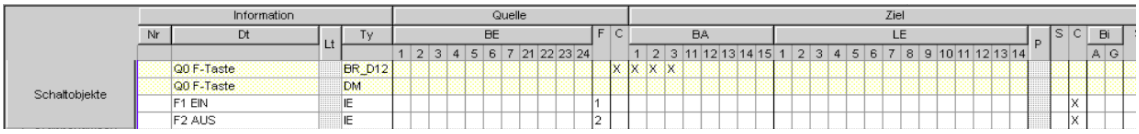
The **CMD** output must be interconnected with the command directly at the right border.

At the **CMD** output the block shows the following behaviour:
If the signal changes from 0 to 1 at the **TRIG** input, the command generated by the **PROP**, **ORIGIN**, **VAL** and **TIME** settings are switched through to **CMD**.

Application example

Using the CFC chart shown below, you can control a circuit breaker via two function keys, for example:
F1 switches the circuit breaker on,
F2 switches the circuit breaker off.

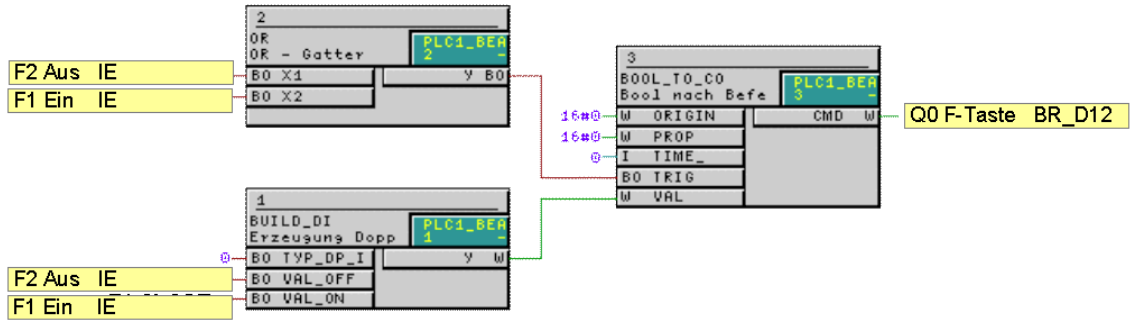
✧ Insert the required information into the configuration based on the following section:



BOOL_TO_CO_Beispiel_Matrix.tif

Fig. 4-34 Application example BOOL_TO_CO block, configuration section

✧ Implement the following CFC chart in the priority class **PLC1_BEARB** (Slow PLC processing):



BOOL_TO_CO_Beispiel_Plan.tif

Fig. 4-35 Application example of BOOL_TO_CO block, CFC chart section

4.6.2 BOOL_TO_IC

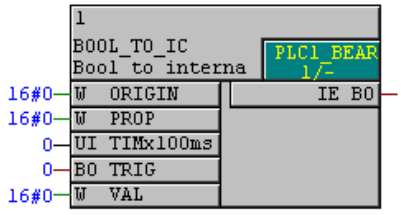
Note

The structure and function of the blocks **BOOL_TO_IC** and **BOOL_TO_IE** are identical. In SIPROTEC 4 devices with older firmware states the block can still be called **BOOL_TO_IE**.

Function

The **Boolean to Internal IC** block generates a command as an internal single point indication.

The internal single-point indication is defined as a command via the parameters **ORIGIN** and **VAL** and triggered via a signal at the **TRIG** input. The inputs **PROP** and **TIM** are not relevant.



BOOL_TO_IC.tif

Fig. 4-36 BOOL_TO_IC block

Note

The **Boolean to Internal IC** block works exclusively in the priority classes Fast PLC processing (priority class **PLC_BEARB**), Slow PLC processing (priority class **PLC1_BEARB**) and Interlocking (priority class **SFS_BEARB**).

I/O assignment

The block **BOOL_TO_IC** has the following I/O assignment:

Table 4-52 Anschlussbelegung Baustein BOOL_TO_IC

	Name	Data type	Explanation	Presetting
Input:	PROP	WORD	No function	16#0000
	TRIG	BOOL	Trigger	0
	ORIGIN	WORD	Origin	16#0000
	VAL	WORD	Switching direction	16#0000
	TIMx100ms	UINT	No function	0
Outputs:	IE	BOOL	Command (internal single-point indication)	0

ORIGIN settings

The **ORIGIN** settings is used to assign a certain command source to the switching command you have generated. If you assign for example the value 01 to **ORIGIN**, the command will be treated during execution like a local command (at device itself).

The following values can be set in the **ORIGIN** (command source) parameter:

Table 4-53 Values for the BOOL_TO_CO block, ORIGIN settings

Value	Meaning
00	Automatically generated command
01	Integrated operation, device panel control (e.g. at device itself)
02	DIGSI, SICAM, local control (e.g. DIGSI Remote, DIGSI on bus)
03	Telecontrol centers, remote control (e.g. WinCC, telecontrol station)

VAL settings

The following values can be set in the **VAL** (actuating direction) settings:

Table 4-54 Values for the BOOL_TO_IC block VAL settings

Value	Meaning
01	Off (01)
02	On (10)

IE output

At the **IE** output the block shows the following behaviour:
If the signal changes from 0 to 1 at the **TRIG** input, the command generated as an internal single point indication by the **PROP**, **ORIGIN**, **VAL** and **TIME** parameters is switched through to **CMD**.

4.6.3 BOOL_TO_IE

Note

Structure and function of the blocks **BOOL_TO_IE** and **BOOL_TO_IC** are identical. In SIPROTEC 4 relays using more recent firmware versions, the block is named **BOOL_TO_IC** (for a description of the block see chapter 4.6.2).

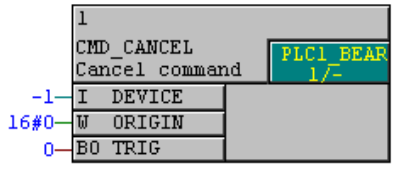
4.6.4 CMD_CANCEL

Function

The **Cancel Command** block enables you to cancel a running command.

If the value -1 (default setting) is assigned to the object address **DEVICE**, all commands are cancelled. Cancelling is triggered by a rising pulse edge at the **TRIG** input without taking the secondary conditions into account (switching authority, interlocking conditions etc.).

The specification for the **ORIGIN** source does not affect the function of the block. The source is just additional information for the indication log.



CMD_CANCEL.tif

Fig. 4-37 CMD_CANCEL block

Note

For the interconnection of **fast signals** with the left border of the CFC chart in the priority classes measured value processing (priority class **MW_BEARB**) and interlocking (priority class **SFS_BEARB**), the following applies:

Boolean inputs are **no** triggers for this level and remain **unnoticed** when the event or signal is shorter than the processing cycle of the processing level.

For the interconnection of **measured values** with the left border of the CFC in the task levels **fast PLC processing** (task level **PLC_BEARB**) and **slow PLC processing** (task level **PLC1_BEARB**), the following applies:

Measuring inputs are **no** triggers for this level and remain **unnoticed**.

I/O assignment

The **CMD_CANCEL** block has the following I/O assignment:

Table 4-55 I/O assignment of CMD_CANCEL block

	Name	Data type	Comment	Default setting
Inputs:	DEVICE	INT	Command to be cancelled The input is connected to the border. The name of the command is displayed there.	-1
	ORIGIN	WORD	Command source	16#0000
	TRIG	BOOL	Start input	0

ORIGIN settings

The **ORIGIN** parameter is used to assign a certain command source to the command cancellation. If you assign for example the value 01 to **ORIGIN**, the command will be treated during execution as if it were a local command.

The following values can be set in the **ORIGIN** (source) settings:

Table 4-56 Values for CMD_CANCEL block, ORIGIN settings

Value	Meaning
00	Automatically generated command
01	Integrated operation, device panel control (e.g. at device itself)
02	DIGSI, SICAM, local control (e.g. DIGSI Remote, DIGSI on Bus)
03	Telecontrol centers, remote control (e.g. WinCC, telecontrol station)

4.6.5 CMD_CHAIN

Function

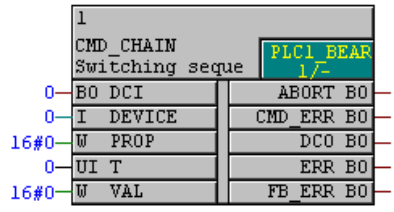
The **Switching Sequence** block generates chains which consist of several individual commands.

Basic functions of this block are:

- Render command.
- Wait for reply of successfully executed command and
- Signal success at the output of the block for further commands.

Switching sequences are parameterized as a command chain. Several **CMD_CHAIN** blocks are switched in series. The Daisy Chain mechanism is used for block chaining. The sequence of the switching operations in a command chain is determined by the block's position in the command chain.

The **LOOP** block can be used to reset the complete switching sequence after the execution of the last successful command.



CMD_CHAIN.tif

Fig. 4-38 CMD_CHAIN

Note

The **CMD_CHAIN** block functions exclusively in the priority classes Fast PLC processing (priority class **PLC_BEARB**) and Slow PLC processing (priority class **PLC1_BEARB**).

Note

If the **CMD_CHAIN** block is used in the priority class Slow PLC processing (priority class **PLC1_BEARB**), the block **LOOP** block has to be used in order to reset the complete switching sequence after the last command has been completed successfully.

Note

The following applies for SIPROTEC devices with a device version **less than** V4.5:

The maximum permissible number of **CMD_CHAIN** blocks in the priority classes
– Fast PLC processing (priority class **PLC_BEARB**) and
– Slow PLC processing (priority class **PLC1_BEARB**)
is **20** blocks.

I/O assignment

The **CMD_CHAIN** block has the following I/O assignment:

Table 4-57 I/O assignment of CMD_CHAIN block

	Name	Data type	Comment	Default setting
Inputs:	DCI	BOOL	Daisy Chain input of the block In a command chain, the connection must be interconnected with the **DCO** output of the preceding **CMD_CHAIN** block.	0
	DEVICE	INT	Command to be addressed The input must be interconnected with the left border. The name of the command is entered there.	0
	PROP	WORD	Command properties	16#0000
	T	UINT	Output time (resolution 100 ms; valid value range 0 to 32767)	0
	VAL	WORD	Switching direction	16#0000
Outputs:	ABORT	BOOL	Cancel The output is active if a running command has been cancelled (AB+).	0
	CMD_ERR	BOOL	Command error The output is active if a command has not been executed successfully (BF-).	0
	DCO	BOOL	Daisy Chain output of the block In a command chain, the connection must be interconnected with the **DCI** input of the subsequent **CMD_CHAIN** block. The output is active, if the switching command has been successfully processed by the block. **DCO** is switched inactive, as soon as the **DCI** input has been switched inactive.	0
	ERR	BOOL	Group error The output is active, if **ABORT**, **CMD_ERR** or **FB_ERR** is active, or if an internal error has occurred.	0
	FB_ERR	BOOL	Feed back error The output is active, if command execution does not receive a feedback (RM-).	0

PROP settings

The following values can be set in the **PROP** (command properties) settings:

Table 4-58 Values for CMD_CHAIN block, PROP settings

Value (hexadecimal)	Value (decimal)	Command properties
00	0	No deblocking
01	1	Deblocking setpoint=actual
02	2	Deblocking of station blocking
04	4	Deblocking of bay blocking
08	8	Synchrocheck deblocking (SY configuration is ignored)
10	16	With synchrocheck (synchronisation conditions are force-checked)
20	32	Deblocking of double-command blocking
40	64	Deblocking of protection device blocking
80	128	Deblocking when station or bay blocking is set

If **PROP** is set to 00 (hexadecimal), the original command properties configured in the DIGSI configuration matrix apply.

Note

If you would like to define several command properties at the same time, you must add up the hexadecimal values individually.

VAL settings

The following values can be set in the **VAL** (actuating direction) settings:

Table 4-59 Values for CMD_CHAIN block VAL settings

Value	Meaning
01	Off (01)
02	On (10)

T settings

In the **T** (output time) parameter, you set the output time in increments of 100 milliseconds.

If **T** is set to 0, the original output time of the command configured in the DIGSI configuration matrix applies.

The **T** parameter (output time) is applied during the start-up of the SIPROTEC device. It **cannot** be changed during operation.

Only times less than 3,276.8 seconds are accepted. Larger values are limited to this value.

Note

A current switching sequence can be cancelled by resetting the signal at the **DCI** input of the first block in the command chain. This block switches the signal through to all the following blocks in the chain via the **DCO** output. The active block in the command chain is marked by **DCI** = 1 and **DCO** = 0.

If the signal at the **DCI** input drops out during a current switching command, an abortion task is generated for the current command and the switching sequence is terminated. The result of the abortion task is awaited:

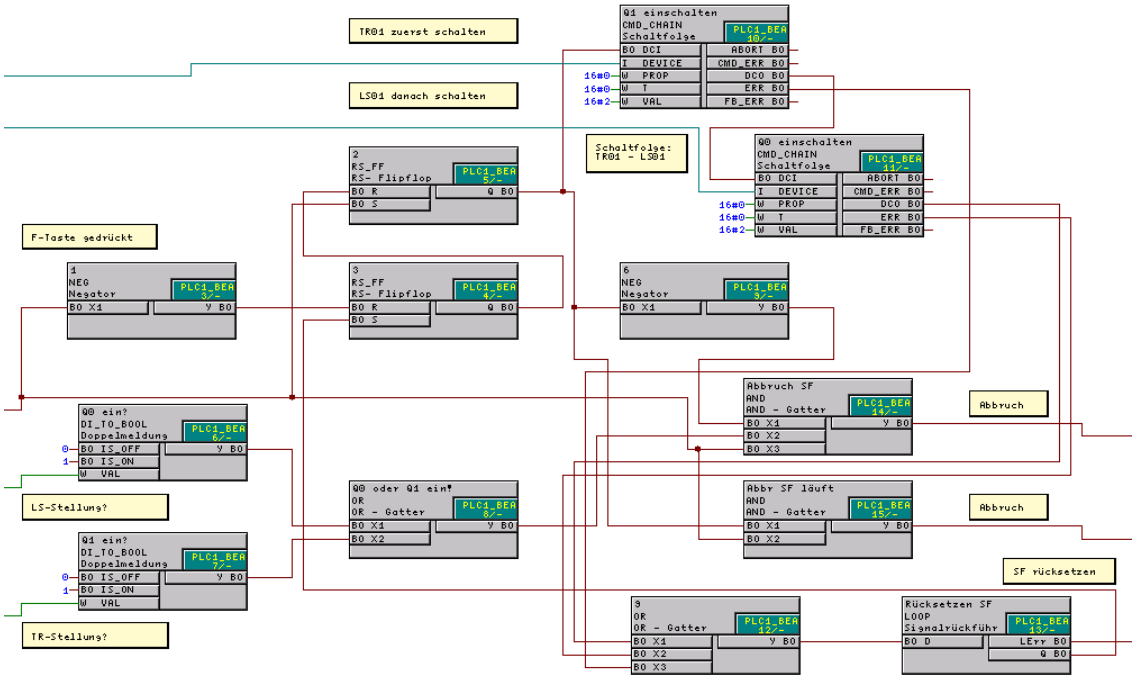
- Command can be aborted
  The switching operation which can be aborted currently is terminated immediately. Switching operations following the command chain are not executed.
- Command cannot be aborted
  The switching operation which cannot be aborted currently is terminated immediately. Switching operations following the command chain are not executed.

Whether or not a command can be aborted depends on the operating mode (object properties) of the command:

- Persistent output cannot be aborted.
- Pulse output can be aborted.

Application example

The following example shows two neighbouring switching sequence blocks which first switch TR01 and then LS01:



Schaltfolge.gif

Fig. 4-39 Example of a switching sequence

4.6.6 CMD_INF

Function

The block **CMD_INF** (Command Information) provides the information whether a command is available in the command management and is currently being processed. This applies to all switching commands, from command selection (Select before operate) to command output, processing a feedback and end of command (COE).
In **switching devices requiring synchronization**, the synchronization process is a part of the close command.

Note

The **CMD_INF** block is intended for use with Interlocking (priority class **SFS_BEARB**), to check the switching state of a switching object. The block only functions conditionally in the other priority classes.

Interlocking conditions will not be taken into account. It is only shown whether a switching command is attached (regardless of whether the switching command is also carried out).

You must interconnect a **CMD_INF** block for each control device from which you require information for an initialized switching command.

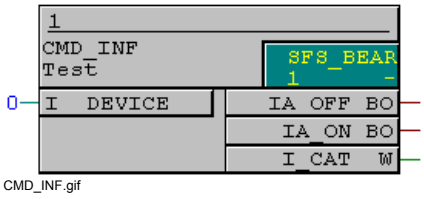


Fig. 4-40 CMD_INF block

I/O assignment

The **CMD_INF** block has the following I/O assignment:

Table 4-60 I/O assignment of the CMD_INF block

	Name	Data type	Comment	Default setting
Inputs:	DEVICE	INT	The input is connected to the left border. The name of the command object is displayed there.	0
Outputs:	IA_ON	BOOL	Information active if a close command was selected or released (including synchronization process).	0.
	IA_OFF	BOOL	Information active if a trip command was selected or released.	0
	I_CAT	WORD	Information on the initiator of the command	16#0000

I_CAT settings

The following values can be put out via the **I_CAT** settings:

Table 4-61 Values for CMD_INF block I_CAT settings

Value	Meaning
00	Automatically generated command
01	Integrated operation, device panel control (e.g. at device itself)
02	DIGSI, SICAM, local control (e.g. DIGSI Remote, DIGSI on Bus)
03	Telecontrol centers, remote control (e.g. WinCC, telecontrol station)

4.6.7 CMD_INF_EXE

Function

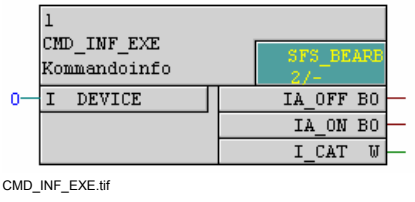
The block **CMD_INF_EXE** (Command Information) provides the information whether a command is available in the command management and is currently being processed. This applies to all switching commands, from command output to processing a feedback and end of command (COE).
In **switching devices requiring synchronization**, the synchronization process is a part of the close command.

Note

The **CMD_INF_EXE** block is intended for use with Interlocking (priority class **SFS_BEARB**), to check the switching state of a switching object. The block only functions conditionally in the other priority classes.

Interlocking conditions will not be taken into account. It is only shown whether a switching command is attached (regardless of whether the switching command is also carried out).

You must interconnect a **CMD_INF_EXE** block for each control device from which you require information for an initialized switching command.



CMD_INF_EXE.tif

Fig. 4-41 CMD_INF_EXE block

I/O assignment

The **CMD_INF_EXE** block has the following I/O assignment:

Table 4-62 I/O assignment of the CMD_INF_EXE block

	Name	Data type	Comment	Default setting
Inputs:	DEVICE	INT	The input is connected to the left border. The name of the command object is displayed there.	0
Outputs:	IA_ON	BOOL	Information active if a close command was released (including synchronization process). A select command is ignored.	0.
	IA_OFF	BOOL	Information active if a trip command was released. A select command is ignored.	0
	I_CAT	WORD	Information on the initiator of the command	16#0000

I_CAT settings

The following values can be put out via the **I_CAT** settings:

Table 4-63 Values for CMD_INF_EXE block I_CAT settings

Value	Meaning
00	Automatically generated command
01	Integrated operation, device panel control (e.g. at device itself)
02	DIGSI, SICAM, local control (e.g. DIGSI Remote, DIGSI on Bus)
03	Telecontrol centers, remote control (e.g. WinCC, telecontrol station)

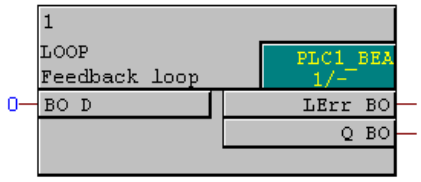
4.6.8 LOOP

Function

The **signal feedback** block enables a signal to be fed back.
Here, feedback means the interconnection of a block output to the input of a block with a smaller sequence number within a CFC chart.

The CFC chart that includes the fed back signal is processed again. New input signals that might have arrived at the blocks are ignored. During the worksheet's reprocessing, input signals do not change except for the signal that is being fed back by the **LOOP** block.

The **LOOP** block is used, for example, at switching sequences (see **CMD_CHAIN** block): After termination of the last successful command, the signal is fed back thus resetting the complete switching sequence.



LOOP.gif

Fig. 4-42 LOOP block

Note

The **Loop** block works exclusively in the priority classes Slow PLC processing (priority class **PLC1_BEARB**), measured value processing (priority class **MW_BEARB**) and interlocking (priority class **SFS_BEARB**).

Note

If the **CMD_CHAIN** block is used in the priority class Slow PLC processing (priority class **PLC1_BEARB**), the block **LOOP** block has to be used in order to reset the complete switching sequence after the last command has been completed successfully.

Note

Optimizing the run sequence (**Options > Optimize run sequence**) may change the sequence numbers of the blocks: As a consequence, it is possible that the LOOP block does not work properly.
You must not optimize the run sequence for charts containing LOOP blocks.

I/O assignment

The **LOOP** block has the following I/O assignment:

Table 4-64 I/O assignment of LOOP block

	Name	Data type	Comment	Default setting
Inputs:	D	BOOL	The signal at the **D** input is switched through to the **Q** output.	0
Outputs:	Q	BOOL	The signal at the **Q** output is switched through to the **D** input.	0
	LErr	BOOL	Error Output Output is active, if a signal has been fed back by **LOOP** for more than 5 times.	0

Note

In order to avoid endless feedback, the signal cannot be fed back by the **LOOP** block for more than 5 times (input **D** = output **Q**). If this number is exceeded, the error output is set to **LErr** and feedbacking is interrupted.

4.7 Type Converters

With type converter blocks, you can convert the data type of information items.

The following type converter blocks are available:

- **BOOL_TO_DI** (Boolean to double-point indication)
- **BUILD_DI** (double-point indication generation)
- **DI_TO_BOOL** (double-point indication to Boolean)
- **DINT_TO_REAL** (adapter)
- **DIST_DECODE** (decode double-point indication) with status
- **DM_DECODE** (double-point indication decoding)
- **REAL_TO_DINT** (adapter)
- **REAL_TO_UINT** (adapter)
- **REAL_TO_INT** (Adapter)
- **INT_TO_REAL** (Adapter)
- **UINT_TO_REAL** (adapter)

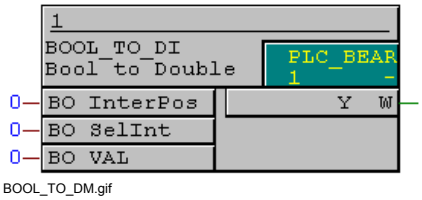
4.7.1 BOOL_TO_DI

Note

The **BOOL_TO_DI** block is part of the original scope of supply of SIPROTEC 4 devices. To generate new CFC charts, we recommend using the easier to use **BUILD_DI** block.

Function

The **Boolean to Double-Point Indication** block generates a double-point indication.



BOOL_TO_DM.gif

Fig. 4-43 BOOL_TO_DI block

Note

For the interconnection of **fast signals** with the left border of the CFC chart in the priority classes measured value processing (priority class **MW_BEARB**) and interlocking (priority class **SFS_BEARB**), the following applies:

Boolean inputs are **no** triggers for this level and remain **unnoticed** when the event or signal is shorter than the processing cycle of the processing level.

For the interconnection of **measured values** with the left border of the CFC in the task levels **fast PLC processing** (task level **PLC_BEARB**) and **slow PLC processing** (task level **PLC1_BEARB**), the following applies:

Measuring inputs are **no** triggers for this level and remain **unnoticed**.

Note

The **Boolean to Double-Point Indication** block works exclusively in the priority classes Fast PLC processing (priority class **PLC_BEARB**), Slow PLC processing (priority class **PLC1_BEARB**) and Interlocking (priority class **SFS_BEARB**).

I/O assignment

The **BOOL_TO_DI** block has the following I/O assignment:

Table 4-65 I/O assignment of the BOOL_TO_DI block

	Name	Data type	Comment	Default setting
Inputs:	SelInt	BOOL	Toggles between the VAL and InterPos inputs (Toggles between switch end position and switch intermediate position)	0
	VAL	BOOL	Value (switch end position)	0
	InterPos	BOOL	Intermediate position	0
Outputs:	Y	WORD	Created double-point indication	16#0000

Created double-point indications

The signals at the **VAL** and **InterPos** inputs can generate the following **Y** double-point indications as a function of the **SelInt** input (toggles between switch end position and switch intermediate position):

The abbreviated used have the following meaning:
DP double-point indication and
DP_I double-point indication with distinction of the intermediate position.

Table 4-66 Created double-point indications BOOL_TO_DI block

SelInt input	InterPos input	VAL input	Y output	Meaning for DP	Meaning for DP_I
0	X	0	0001	OFF	OFF
0	X	1	0002	ON	ON
1	0	X	0000	Undefined	INTERM (intermediate position 00)
1	1	X	0003	INTERM	INTERM (intermediate position 11)

4.7.2 BUILD_DI

Function

The **Create Double-Point Indication** block generates a double-point indication from two input signals.

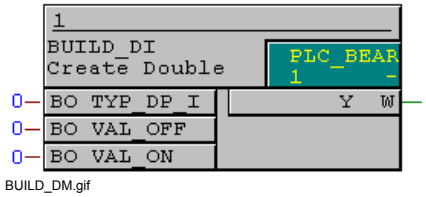


Fig. 4-44 BUILD_DI block

Note

The **Create Double-Point Indication** block works exclusively in the priority classes Fast PLC processing (priority class **PLC_BEARB**), Slow PLC processing (priority class **PLC1_BEARB**) and Interlocking (priority class **SFS_BEARB**).

Note

For the interconnection of **fast signals** with the left border of the CFC chart in the priority classes measured value processing (priority class **MW_BEARB**) and interlocking (priority class **SFS_BEARB**), the following applies:

Boolean inputs are **no** triggers for this level and remain **unnoticed** when the event or signal is shorter than the processing cycle of the processing level.

For the interconnection of **measured values** with the left border of the CFC in the task levels **fast PLC processing** (task level **PLC_BEARB**) and **slow PLC processing** (task level **PLC1_BEARB**), the following applies:

Measuring inputs are **no** triggers for this level and remain **unnoticed**.

I/O assignment

The **BUILD_DI** block has the following I/O assignment:

Table 4-67 I/O assignment of the BUILD_DI block

	Name	Data type	Comment	Default setting
Inputs:	VAL_OFF	BOOL	Indicates the OFF state	0
	VAL_ON	BOOL	Indicates the ON state	0
	TYP_DP_I	BOOL	Type of displayed double-point indication	0
Outputs:	Y	WORD	Created double-point indication	16#0000

TYP_DP_I settings

The following values can be set in the **TYP_DP_I** (type of created double-point indication) settings:

Table 4-68 Values for BUILD_DI block TYP_DP_I settings

Value	Meaning
0	DP (double-point indication)
1	DP_I (double-point indication with distinction of the intermediate position)

Created double-point indications

The signals at the **VAL_OFF** and **VAL_ON** inputs can generate the following **Y** double-point indications as a function of the **TYP_DP_I** input:

Table 4-69 Created double-point indications of the BUILD_DI block

TYP_DP_I input	VAL_ON input	VAL_OFF input	Y output	Meaning
0	0	0	0003	INTERM
0	0	1	0001	OFF
0	1	0	0002	ON
0	1	1	0003	INTERM

Table 4-69 Created double-point indications of the BUILD_DI block

TYP_DP_I input	VAL_ON input	VAL_OFF input	Y output	Meaning
1	0	0	0000	INTERM (intermediate position 00)
1	0	1	0001	OFF
1	1	0	0002	ON
1	1	1	0003	INTERM (intermediate position 11)

4.7.3 DI_TO_BOOL

Note

The **DI_TO_BOOL** block is part of the original scope of supply of SIPROTEC 4 devices. To generate new CFC charts, we recommend using the easier to use **DM_DECODE** block.

Function

The **Double-Point Indication to Boolean** block checks a double-point indication for one of the four possible states and generates a signal as a result.

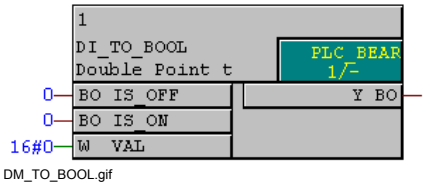


Fig. 4-45 DI_TO_BOOL block

Note

The **Double-Point Indication to Boolean** block works exclusively in the priority classes Fast PLC processing (priority class **PLC_BEARB**), Slow PLC processing (priority class **PLC1_BEARB**) and Interlocking (priority class **SFS_BEARB**).

I/O assignment

The **DI_TO_BOOL** block has the following I/O assignment:

Table 4-70 I/O assignment of the DI_TO_BOOL block

	Name	Data type	Comment	Default setting
Inputs:	IS_OFF	BOOL	Query for OFF	0
	IS_ON	BOOL	Query for ON	0
	VAL	WORD	double-point indication	16#0000
Outputs:	Y	BOOL	Result	0

VAL settings

The following values can be present in the **VAL** (double-point indication) settings:

Table 4-71 Values for the DI_TO_BOOL block VAL settings

Value	Meaning for DP_I (double-point indication with distinction of the intermediate positions)	Meaning for DP (double-point indication)
0000	INTERM (intermediate position **00**), Undefined	Undefined (corresponds to **00**)
0001	OFF (corresponds to **01**)	OFF (corresponds to **01**)
0002	ON (corresponds to **10**)	ON (corresponds to **10**)
0003	INTERM (intermediate position **11**)	INTERM (corresponds to **00**)

Generated output signal

The signals at the **IS_ON** and **IS_OFF** inputs can generate the following **Y** output signal as a function of the **VAL** double-point indication:

Table 4-72 Generated output signal of the DI_TO_BOOL block

IS_ON input	IS_OFF input	VAL input for DP	VAL input for DP_I	Y output
0	0	Undefined	INTERM (intermediate position 00), Undefined	1
		OFF, ON, INTERM	OFF, ON, INTERM (intermediate position 11)	0
0	1	OFF	OFF	1
		ON, INTERM, Undefined	ON, INTERM (intermediate position 00), INTERM (intermediate position 11) Undefined	0
1	0	ON	ON	1
		OFF, INTERM, Undefined	OFF, INTERM (intermediate position 00), INTERM (intermediate position 11) Undefined	0
1	1	INTERM, Undefined	INTERM (intermediate position 11)	1
		OFF, ON	OFF, ON, INTERM (intermediate position 00), Undefined	0

Application example

Using the CFC chart shown below, you can create the following interlocking conditions in the processing class **SFS_BEARB** (Interlocking):
Closing the **circuit breaker**: Disconnector = **ON** and earth switch = **OFF**,
Closing the **earth switch**: Circuit breaker = **OFF** and earth switch = **OFF**,
Closing the **disconnector**: Circuit breaker = **OFF** and earth switch = **OFF**.

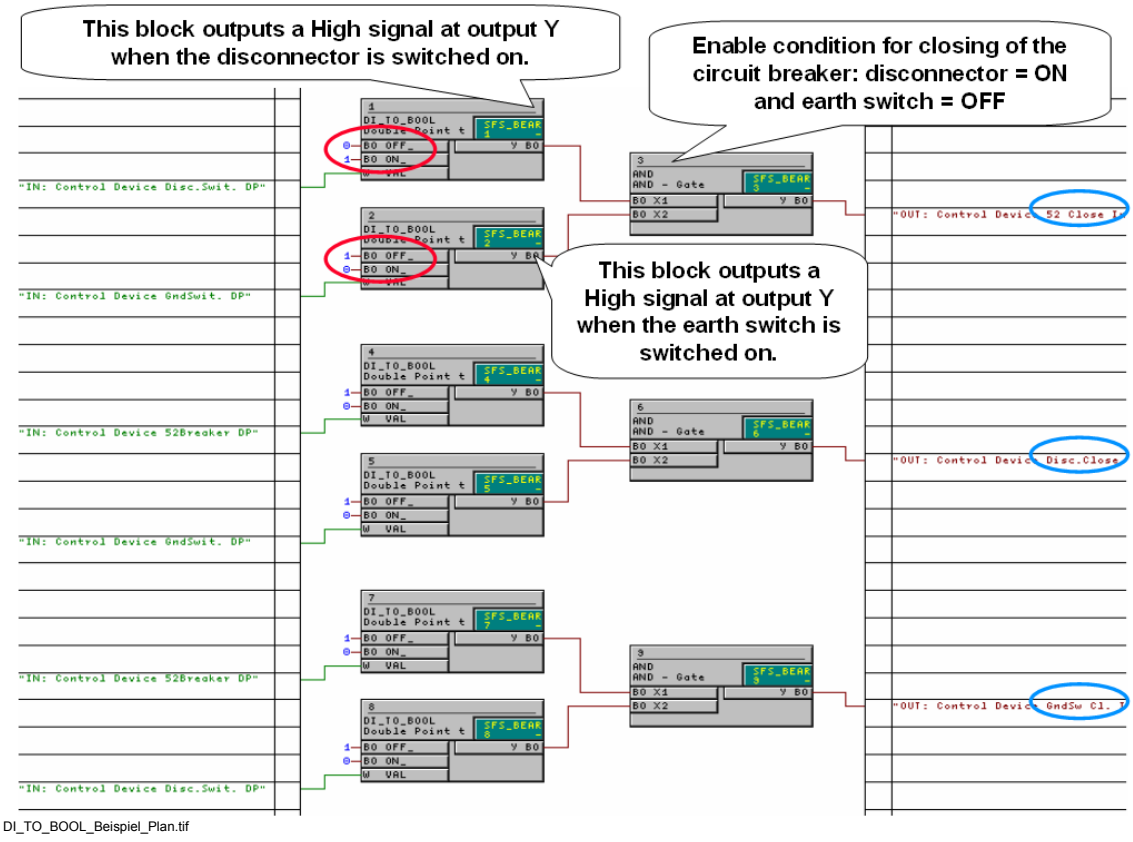


Fig. 4-46 Application example of DI_TO_BOOL block, chart section

4.7.4 DINT_TO_REAL

Function

The **DINT_TO_REAL** block converts double integer values to real values and is thereby the opposite of the **REAL_TO_DINT** block.

DINT_TO_REAL.tif

Fig. 4-47 DINT_TO_REAL block

I/O assignment

The **DINT_TO_REAL** block has the following I/O assignment:

Table 4-73 I/O assignment of DINT_TO_REAL block

	Name	Data type	Comment	Default setting
Inputs:	X	DINT	Double integer input size	0
Outputs:	Y	REAL	Real output size	0.0

Note

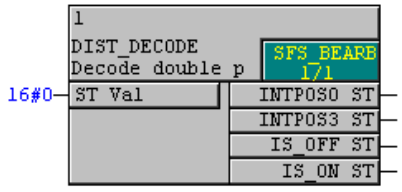
The status information contained in the REAL data type
(see table 4-2) remains intact during the conversion to the DINT data type and is output on the **Y** output.

4.7.5 DIST_DECODE

Function

The **Decode Double-Point Indication** block converts a double-point indication with status to four single point indications with status. As opposed to the **DM_DECODE** block, the status is passed on to the outputs unchanged.

IS_ON, IS_OFF, INTPOS0 and INTPOS3 are possible. These values are available as outputs with status.



DIST_DECODE.tif

Fig. 4-48 DIST_DECODE block

I/O assignment

The **DIST_DECODE** block has the following I/O assignment:

Table 4-74 I/O assignment of the DIST_DECODE block

	Name	Data type	Comment	Default setting
Inputs:	VAL	STRUCT	Value of the double-point indication	(0)
Outputs:	INTPOS0	SIST	INTERM0 (decoded double-point indication) with status	0
	INTPOS3	SIST	OFF (decoded double-point indication) with status	0
	IS_OFF	SIST	ON (decoded double-point indication) with status	0
	IS_ON	SIST	INTERM3 (decoded double-point indication) with status	0

VAL settings

The **VAL** parameter can contain the following values, depending on the interconnection:

Table 4-75 Values for DIST_DECODE block, VAL settings

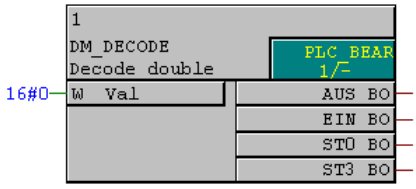
Input value ST Val	Output active for DP	Output active for DP_I
0000 (INTERM 00)	INTPOS3	INTPOS0
0001 (OFF)	IS_OFF	IS_OFF
0002 (ON)	IS_ON	IS_ON
0003 (INTERM 11)	INTPOS3	INTPOS3

4.7.6 DM_DECODE

Function

The **Decode Double-Point Indication** block converts a double-point indication into four Boolean values. As opposed to the **DIST_DECODE** block, the status is not passed on.

ON, OFF, INTERM0 and INTERM3 are possible. These values are available as outputs.



DM_DECODE.gif

Fig. 4-49 DM_DECODE block

I/O assignment

The **DM_DECODE** block has the following I/O assignment:

Table 4-76 I/O assignment of the DM_DECODE block

	Name	Data type	Comment	Default setting
Inputs:	VAL	WORD	Value of the double-point indication	16#0000
Outputs:	ON	BOOL	ON (decoded double-point indication)	0
	OFF	BOOL	OFF (decoded double-point indication)	0
	ST0	BOOL	INTERM0 (decoded double-point indication)	0
	ST3	BOOL	INTERM3 (decoded double-point indication)	0

VAL settings

The **VAL** parameter can contain the following values, depending on the interconnection:

Table 4-77 Values for DM_DECODE block VAL settings

Value	Meaning for DP (double-point indication)	Meaning for DP_I (double-point indication with distinction of the intermediate positions)	Output of DM_DECODE block
0000	Undefined	INTERM (intermediate position 00), Undefined	ST0
0001	OFF	OFF	OFF
0002	ON	ON	ON
0003	INTERM	INTERM (intermediate position 11)	ST3

Note

To process the status of a double-point indication, you can use the **DIST_DECODE** block instead of the **DM_DECODE** block.

4.7.7 REAL_TO_DINT

Function

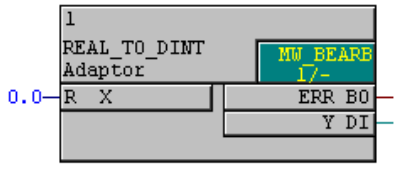
The **REAL_TO_DINT** block converts real values into double integer values.

The value is rounded: for a value of .5, it is rounded to the next highest value.

If overshooting or undershooting of the output value range occurs, the maximum/minimum value is output on the output. The **ERR** output is also set.

The **ERR** output remains set until the input value corresponds to the value range of the **Y** output again.

If there is no valid real value at the **X** input, the value 0 is output on the **Y** output and the **ERR** output is set.



REAL_TO_DINT.tif

Fig. 4-50 REAL_TO_DINT block

I/O assignment

The **REAL_TO_DINT** block has the following I/O assignment:

Table 4-78 I/O assignment of the REAL_TO_DINT block

	Name	Data type	Comment	Default setting
Inputs:	X	REAL	Real input size	0.0
Outputs:	Y	DINT	Double integer output size	0
	ERR	BOOL	Overshooting/under-shooting of the output value range	0

Note

The status information contained in the REAL data type (see table 4-2) remain intact during the conversion to the DINT data type and are output on the **Y** output.

4.7.8 REAL_TO_INT

Function

The **REAL_TO_INT** block converts real values into integer values. It allows, for example, a setpoint value (limit value) to be linked to a time input of a timer so that the timer can be set during device operation via the on-site controls.

The value is rounded: for a value of .5, it is rounded to the next highest value.

If no valid real value is active on the X input, the output ERR is set and the Y output behaves according to the following table:

Table 4-79 Verhalten der Ausgänge beim Baustein REAL_TO_INT

Status at input X	Output Y	Output ERR
OVERFLOW	65535	1
OVERFLOW_NEG	0	1
LIVE_ZERO	Y (n-1) unchanged	1
NOT_DEFINED	Y (n-1) unchanged	1
NOT_CALCULATED	Y (n-1) unchanged	1
INVALID	Y (n-1) unchanged	1

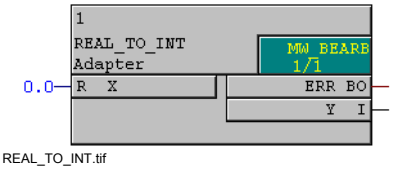


Fig. 4-51 REAL_TO_INT block

Note

The **REAL_TO_INT** block is part of the original scope of supply of SIPROTEC 4 devices. Existing CFC charts with this block continue to be supported. To generate new CFC charts, we recommend using the universally usable **REAL_TO_UINT** block.

4.7.9 REAL_TO_UINT

Function

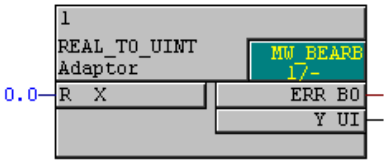
The **REAL_TO_UINT** block converts real values into unsigned integer values. It allows, for example, a setpoint value (limit value) to be linked to a time input of a timer so that the timer can be set during device operation via the on-site controls.

The value is rounded: for a value of .5, it is rounded to the next highest value.

If no valid real value is active on the **X** input, the output **ERR** is set and the **Y** output behaves according to the following table:

Table 4-80 Behaviour of the output with REAL_TO_UINT block

Status at input X	Output Y	Output ERR
OVERFLOW	65535	1
OVERFLOW_NEG	0	1
LIVE_ZERO	Y (n-1) unchanged	1
NOT_DEFINED	Y (n-1) unchanged	1
NOT_CALCULATED	Y (n-1) unchanged	1
INVALID	Y (n-1) unchanged	1



REAL_TO_UINT.tif

Fig. 4-52 REAL_TO_UINT block

I/O assignment

The **REAL_TO_UINT** block has the following I/O assignment:

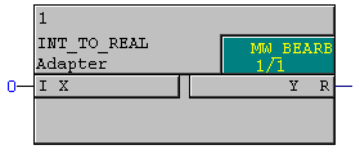
Table 4-81 I/O assignment of the REAL_TO_UINT block

	Name	Data type	Comment	Default setting
Inputs:	X	REAL	Real input size	0.0
Outputs:	Y	UINT	Unsigned integer output size	0
	ERR	BOOL	Overshooting/under-shooting of the output value range	0

4.7.10 INT_TO_REAL

Function

The **INT_TO_REAL** block converts double integer values to real values and is thereby the opposite of the **REAL_TO_INT** block.



INT_TO_REAL.tif

Fig. 4-53 Baustein INT_TO_REAL

I/O assignment

The **INT_TO_REAL** block has the following I/O assignment:

Table 4-82 I/O assignment of INT_TO_REAL block

	Name	Data type	Comment	Default setting
Inputs:	X	UINT	Unsigned integer input size	0
Outputs:	Y	REAL	Real output size	0.0

Note

The **INT_TO_REAL** block is part of the original scope of supply of SIPROTEC 4 devices. Existing CFC charts with this block continue to be supported. To generate new CFC charts, we recommend using the universally usable **UINT_TO_REAL** block.

4.7.11 UINT_TO_REAL

Function

The **UINT_TO_REAL** block converts unsigned integer values to real values and is thereby the opposite of the **REAL_TO_UINT** block.

UINT_TO_REAL.tif

Fig. 4-54 UINT_TO_REAL block

I/O assignment

The **UINT_TO_REAL** block has the following I/O assignment:

Table 4-83 I/O assignment of UINT_TO_REAL block

	Name	Data type	Comment	Default setting
Inputs:	X	UINT	Unsigned integer input size	0
Outputs:	Y	REAL	Real output size	0.0

4.8 Comparison

With comparison blocks, you can compare or, within certain limits, influence (e.g. correct) measured values.

The following comparison blocks are available:

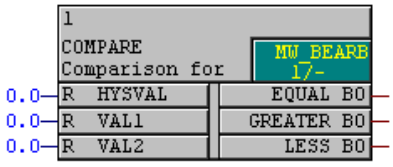
- **COMPARE** (measured value comparison)
- **LIVE_ZERO** (live zero monitoring)
- **LOWER_SETPOINT** (lower limit)
- **UPPER_SETPOINT** (upper limit)
- **ZERO_POINT** (zero suppression)

4.8.1 COMPARE

Function

The **measured value comparison** block enables the comparison of two real values **VAL1** and **VAL2** as **greater than**, **less than** and **equal to** with configurable hysteresis **HYSVAL**.

By using type converters (e.g. **DINT_TO_REAL**), universal comparisons can be made (e.g. metered value comparison). The results are output to the outputs **EQUAL**, **GREATER** and **LESS** as values of type BOOL.



COMPARE.tif

Fig. 4-55 COMPARE block

Note

For the interconnection of **fast signals** with the left border of the CFC chart in the priority classes measured value processing (priority class **MW_BEARB**) and interlocking (priority class **SFS_BEARB**), the following applies:

Boolean inputs are **no** triggers for this level and remain **unnoticed** when the event or signal is shorter than the processing cycle of the processing level.

For the interconnection of **measured values** with the left border of the CFC in the task levels **fast PLC processing** (task level **PLC_BEARB**) and **slow PLC processing** (task level **PLC1_BEARB**), the following applies:

Measuring inputs are **no** triggers for this level and remain **unnoticed**.

I/O assignment

The **COMPARE** block has the following I/O assignment:

Table 4-84 I/O assignment of COMPARE block

	Name	Data type	Comment	Default setting
Inputs:	HYSVAL	REAL	Hysteresis value	0.0
	VAL1	REAL	Operand 1	0.0
	VAL2	REAL	Operand 2	0.0
Outputs:	EQUAL	BOOL	Values are equal (VAL1 = VAL2)	0
	GREATER	BOOL	Value 1 is greater than value 2 (VAL1 > VAL2)	0
	LESS	BOOL	Value 1 is less than value 2 (VAL1 < VAL2)	0

Behaviour of the outputs when HYSVAL = 0

With a hysteresis value HYSVAL = 0 (simple comparison of two values), the outputs exhibit the following behaviour:

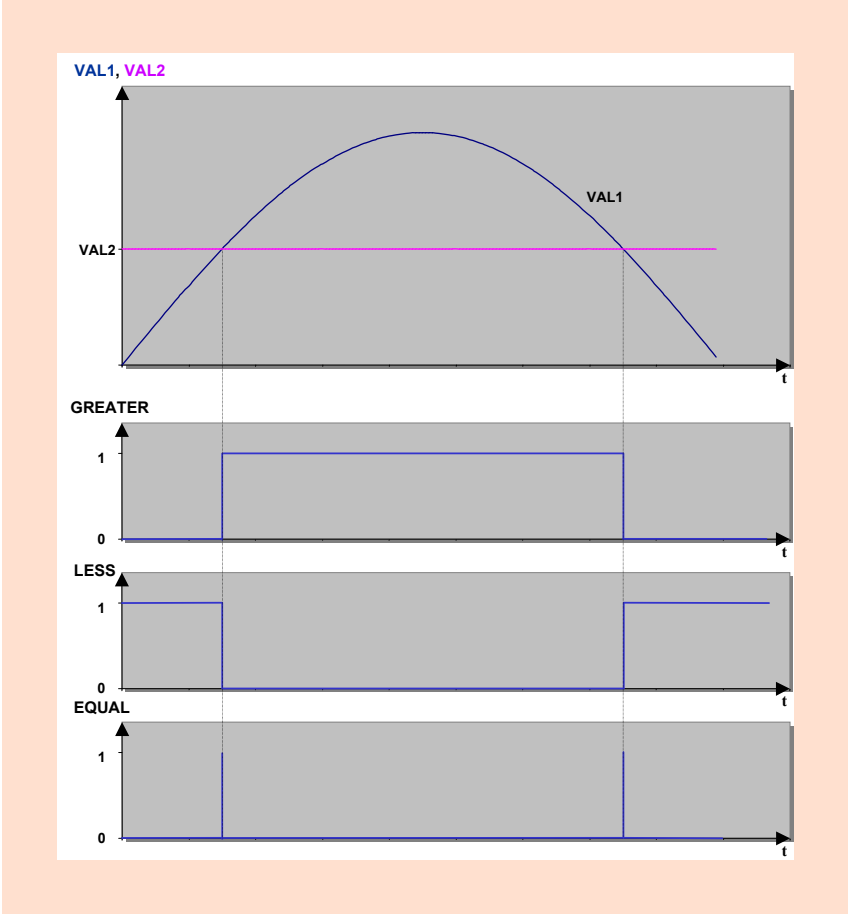


Fig. 4-56 Diagram for HYSVAL = 0 (simple comparison between two values)

Behaviour of the outputs when HYSVAL > 0

With a hysteresis value HYSVAL > 0 (hysteresis around the zero point), the outputs exhibit the following behaviour:

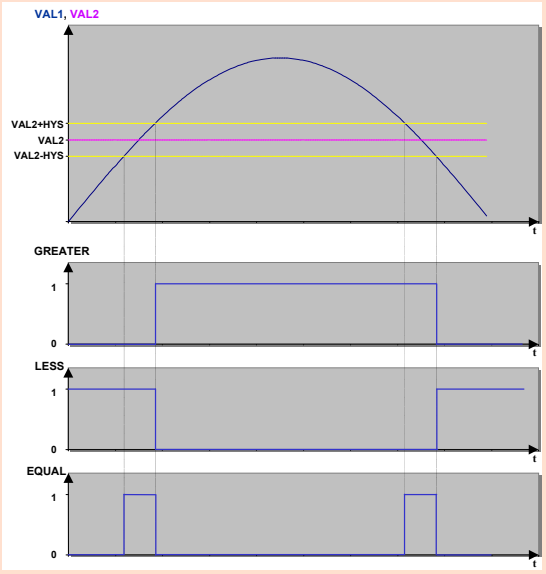


Fig. 4-57 Diagram for HYSVAL > 0 (hysteresis around the zero point)

Behaviour of the outputs when HYSVAL < 0

With a hysteresis value HYSVAL < 0 (comparison with delayed dropout point), the outputs exhibit the following behaviour:

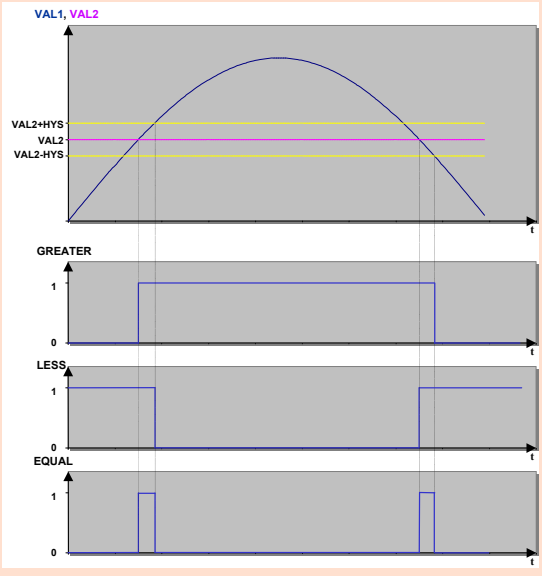


Fig. 4-58 HYSVAL < 0 (comparison with delayed dropout point)

4.8.2 LIVE_ZERO

Note

The blocks **LIVE_ZERO**, **LOWER_SETPOINT**, **UPPER_SETPOINT** and **ZERO_POINT** are designed to process measured values only.

Function

In order to be able to detect sensor faults or wiring faults between the transducer and the sensor, the measured values to be captured are not allowed to vary between 0% and 100%, but only between the live zero value (usually 20%) and 100%.
Values below the live zero value are interpreted as a fault and will cause the **Live Zero Monitoring** block to generate a message.

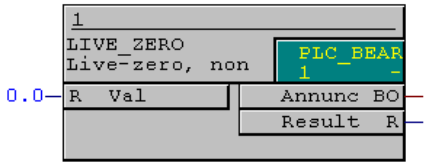


Fig. 4-59 LIVE_ZERO block

Note

For the interconnection of **fast signals** with the left border of the CFC chart in the priority classes measured value processing (priority class **MW_BEARB**) and interlocking (priority class **SFS_BEARB**), the following applies:

Boolean inputs are **no** triggers for this level and remain **unnoticed** when the event or signal is shorter than the processing cycle of the processing level.

For the interconnection of **measured values** with the left border of the CFC in the task levels **fast PLC processing** (task level **PLC_BEARB**) and **slow PLC processing** (task level **PLC1_BEARB**), the following applies:

Measuring inputs are **no** triggers for this level and remain **unnoticed**.

I/O assignment

The **LIVE_ZERO** block has the following I/O assignment:The **LiveZero**, **DetecKnee** and **DispKnee** inputs can

Table 4-85 I/O assignment of the LIVE_ZERO block

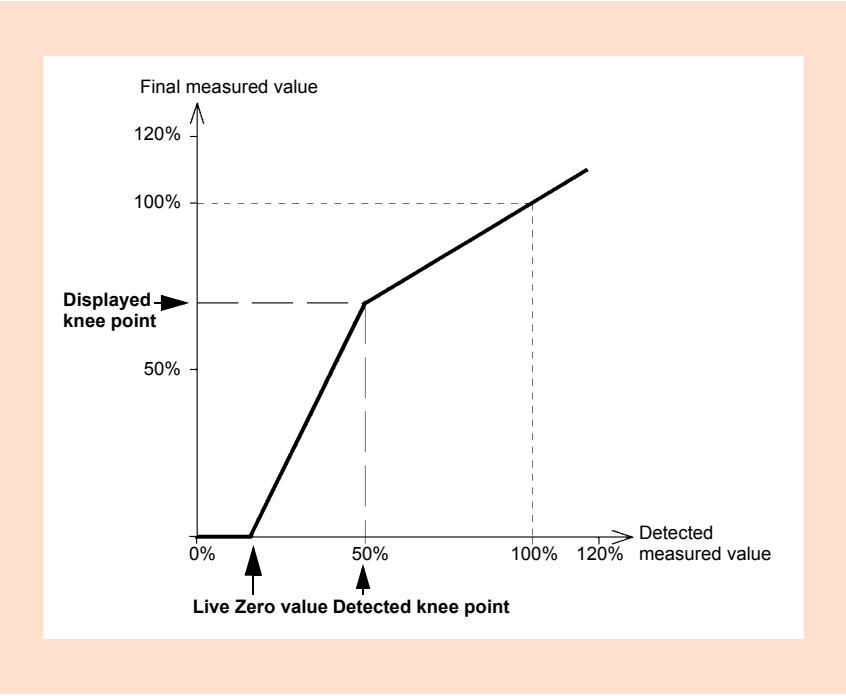
	Name	Data type	Comment	Default setting
Inputs:	Val	REAL	Measured value in %	0.0
	LiveZero	REAL	Live zero value in % (≥ 0.0 %)	20.0
	DetecKnee	REAL	Detection of knee point in % (≥ 0.0 % and < 100.0 %)	50,0
	DispKnee	REAL	Displayed knee point in % (0.0...200.0%)	70.0
Outputs:	Result	REAL	Live zero mesured value in %	0.0
	Annunc	BOOL	Indication: Live zero monitoring	0

Note

The **LiveZero**, **DetecKnee** and **DispKnee** inputs can be configured via the **Object Properties** context menu command of the block.

Note

In order to prevent the **Live Zero Monitoring** message from chattering, a hysteresis with a resetting ratio of 0.95, at least 0.5%, is applied when generating the indication.



LIVE.gif

Fig. 4-60 Diagram of live zero monitoring

Note

If you configure the **LIVE ZERO** block incorrectly and leave the permissible value ranges of the **LiveZero**, **DetecKnee** or **DispKnee** inputs, the **Result** output is set to **NOT_CALCULATED**. This value is displayed with three points **...** in the device display of a SIPROTEC device.

4.8.3 LOWER_SETPOINT

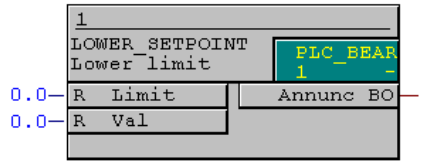
Note

The blocks **LIVE_ZERO**, **LOWER_SETPOINT**, **UPPER_SETPOINT** and **ZERO_POINT** are designed to process measured values only.

Function

The **Lower Limit** block generates a signal at the output **Annunc,** if the measured value at the **Val** input undershoots the configured **Limit** value.

You can connect the lower limit to the output signal of another block or specify a fixed value for it.



LOWER_SETPOINT.gif

Fig. 4-61 LOWER_SETPOINT block

Note

For the interconnection of **fast signals** with the left border of the CFC chart in the priority classes measured value processing (priority class **MW_BEARB**) and interlocking (priority class **SFS_BEARB**), the following applies:

Boolean inputs are **no** triggers for this level and remain **unnoticed** when the event or signal is shorter than the processing cycle of the processing level.

For the interconnection of **measured values** with the left border of the CFC in the task levels **fast PLC processing** (task level **PLC_BEARB**) and **slow PLC processing** (task level **PLC1_BEARB**), the following applies:

Measuring inputs are **no** triggers for this level and remain **unnoticed**.

I/O assignment

The **LOWER_SETPOINT** block has the following I/O assignment:

Table 4-86 I/O assignment of the LOWER_SETPOINT block

	Name	Data type	Comment	Default setting
Inputs:	Limit	REAL	Limit in % (± 1.0 e^{+38})	0.0
	Val	REAL	Measured value in % (± 1.0 e^{+38})	0.0
Outputs:	Annunc	BOOL	Message: Lower Limit	0

Note

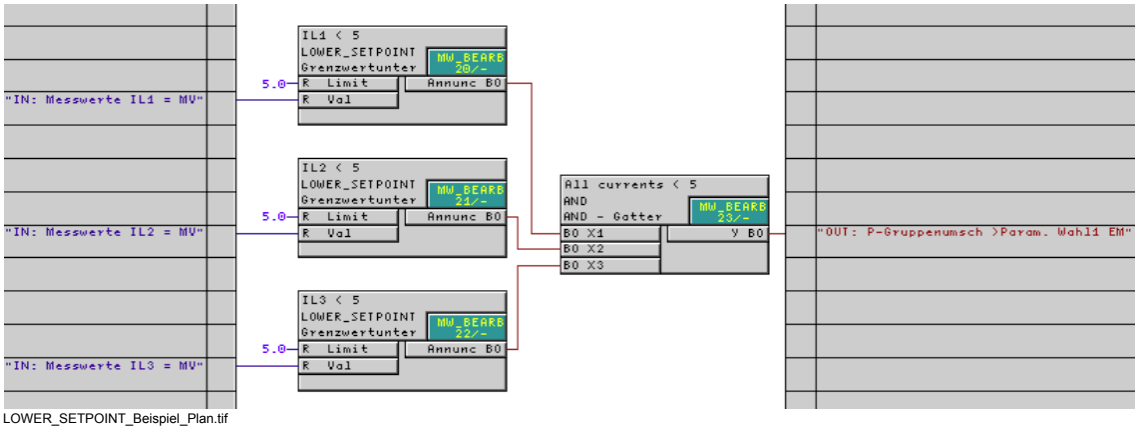
In order to prevent the **Lower Limit** message from chattering, a hysteresis with a resetting ratio of 0.95, at least 0.5%, is applied when generating the indication.

Note

A value at the **Val** input with the flag **Overflow** supplies the value 0 at the **Annunc** output.

Application example

Three phase currents are monitored using the CFC chart below:
If all three phase currents are under 5% of the nominal current, the setting group is changed.



LOWER_SETPOINT_Beispiel_Plan.tif

Fig. 4-62 Application example of LOWER_SETPOINT, chart section

4.8.4 UPPER_SETPOINT

Note

The blocks **LIVE_ZERO**, **LOWER_SETPOINT**, **UPPER_SETPOINT** and **ZERO_POINT** are designed to process measured values only.

Function

The **Upper Limit** block generates a signal at the output **Annunc** if the measured value at the **Val** input overshoots the configured **Limit** limit value.

You can connect the upper limit to the output signal of another block or specify a fixed value for it.

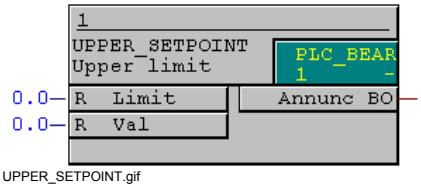


Fig. 4-63 UPPER_SETPOINT block

Note

For the interconnection of **fast signals** with the left border of the CFC chart in the priority classes measured value processing (priority class **MW_BEARB**) and interlocking (priority class **SFS_BEARB**), the following applies:

Boolean inputs are **no** triggers for this level and remain **unnoticed** when the event or signal is shorter than the processing cycle of the processing level.

For the interconnection of **measured values** with the left border of the CFC in the task levels **fast PLC processing** (task level **PLC_BEARB**) and **slow PLC processing** (task level **PLC1_BEARB**), the following applies:

Measuring inputs are **no** triggers for this level and remain **unnoticed**.

I/O assignment

The **UPPER_SETPOINT** block has the following I/O assignment:

Table 4-87 I/O assignment of the UPPER_SETPOINT block

	Name	Data type	Comment	Default setting
Inputs:	Limit	REAL	Limit in % (± 1.0 e^{+38})	0.0
	Val	REAL	Measured value in % (± 1.0 e^{+38})	0.0
Outputs:	Annunc	BOOL	Message: Upper Limit Exceeded	0

Note

In order to prevent the **Upper Limit Exeeded** message from chattering, a hysteresis with a resetting ratio of 0.95, at least 0.5%, is applied when generating the indication.

Note

A value at the **Val** input with the flag **Overflow** supplies the value 1 at the **Annunc** output.

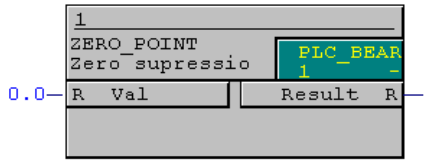
4.8.5 ZERO_POINT

Note

The blocks **LIVE_ZERO**, **LOWER_SETPOINT**, **UPPER_SETPOINT** and **ZERO_POINT** are designed to process measured values only.

Function

As a result of measuring inaccuracies, the measured zero point may slightly deviate from the exact zero point. For compensating this effect, a measured value which lies below a settable value is set to zero. This task is carried out by the **Zero Suppression** block.



ZERO_POINT.gif

Fig. 4-64 ZERO_POINT block

Note

For the interconnection of **fast signals** with the left border of the CFC chart in the priority classes measured value processing (priority class **MW_BEARB**) and interlocking (priority class **SFS_BEARB**), the following applies:

Boolean inputs are **no** triggers for this level and remain **unnoticed** when the event or signal is shorter than the processing cycle of the processing level.

For the interconnection of **measured values** with the left border of the CFC in the task levels **fast PLC processing** (task level **PLC_BEARB**) and **slow PLC processing** (task level **PLC1_BEARB**), the following applies:

Measuring inputs are **no** triggers for this level and remain **unnoticed**.

I/O assignment

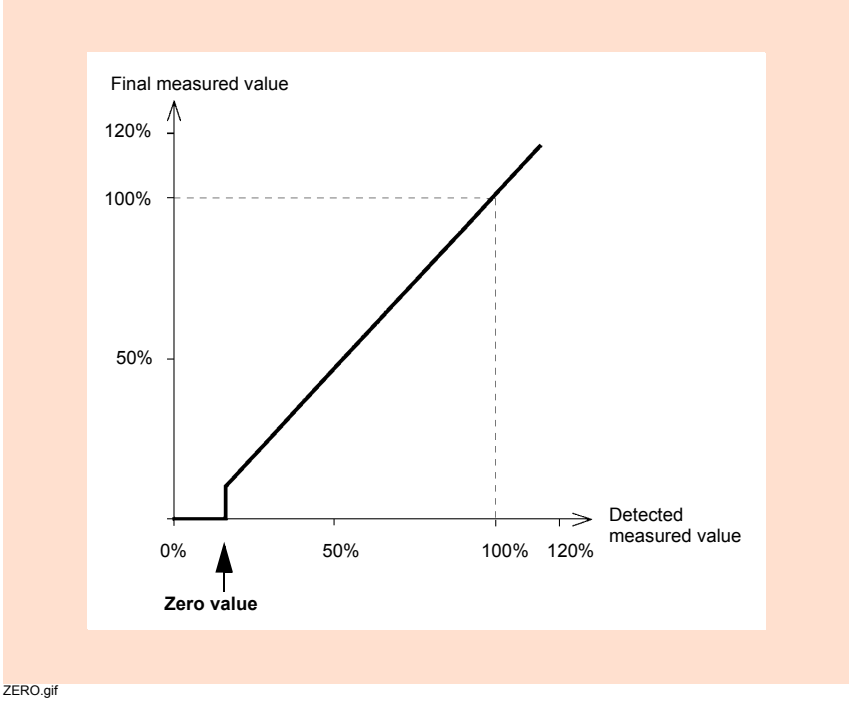
The **ZERO_POINT** block has the following I/O assignment:

Table 4-88 I/O assignment of the ZERO_POINT block

	Name	Data type	Comment	Default setting
Inputs:	Val	REAL	Measured value in %	0.0
	ZeroPoint	REAL	Zero point value in % (0.0...30.0%)	5.0
Outputs:	Result	REAL	Zero-point suppressed measured value in %	0.0

Note

The **ZeroPoint** input can be configured via the **Object Properties** context menu command of the block.



ZERO.gif

Fig. 4-65 Diagram of zero-point suppression

Note

If you configure the **ZERO_POINT** block incorrectly and leave the permissible value ranges of the **Val** or **ZeroPoint** inputs, the **Result** output is set to **NOT_CALCULATED**. This value is displayed with three points **...** in the device display of a SIPROTEC device.

4.9 Metered Value

You can implement counters using the metered value blocks.

The following metered value blocks are available:

- **COUNTER**

4.9.1 COUNTER

Function

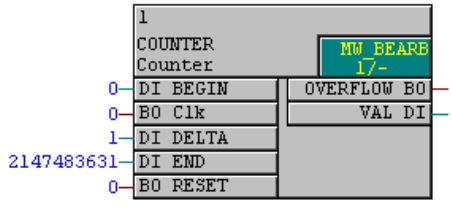
The **counter** block counts the positive edges of the **Clk** input and adds the value **DELTA** to the saved value with each positive edge.

During initial start-up, the counter status **VAL** is initialised with the value active at the **BEGIN** input.

During a restart, the non-volatile saved value of the counter status **VAL** is used.

If the counter status exceeds the value **END**, the counter status is initialised with the value active at the **BEGIN** input, and the output **OVERFLOW** switches from the value 0 to 1. This indicates an overflow. The signal for the overflow remains set until the value 0 is active at the **Clk** input again or the device is restarted.

The counter value can be explicitly set to the value active at the **BEGIN** input by a rising edge at the **RESET** input.



COUNTER.tif

Fig. 4-66 COUNTER block

Note

The maximum permissible number of **RS_FF_MEMO**, **SR_FF_MEMO**, **D_FF_MEMO** and **COUNTER** blocks depends on the available non-volatile memory and is monitored by the CFC compiler. Observe the technical data in the device manual of the SIPROTEC device which you want to use.

The maximum permissible number is checked during the compilation of the CFC chart. Consistency errors are indicated when a fault occurs. The exceeding of the resource is indicated in the displayed compilation log.

Note

For the interconnection of **fast signals** with the left border of the CFC chart in the priority classes measured value processing (priority class **MW_BEARB**) and interlocking (priority class **SFS_BEARB**), the following applies:

Boolean inputs are **no** triggers for this level and remain **unnoticed** when the event or signal is shorter than the processing cycle of the processing level.

For the interconnection of **measured values** with the left border of the CFC in the task levels **fast PLC processing** (task level **PLC_BEARB**) and **slow PLC processing** (task level **PLC1_BEARB**), the following applies:

Measuring inputs are **no** triggers for this level and remain **unnoticed**.

I/O assignment

The **COUNTER** block has the following I/O assignment:

Table 4-89 I/O assignment of COUNTER block

	Name	Data type	Comment	Default setting
Inputs:	BEGIN	DINT	Initial value of the counter	0
	Clk	BOOL	Counts the positive edges	0
	DELTA	DINT	Change of the metered value with positive edge at input Clk	0
	END	DINT	Final value of the counter	2147483631
	RESET	BOOL	Resets the counter to the initial value	0
Outputs:	VAL	DINT	Current status of the counter	Saved value
	OVERFLOW	BOOL	Counter overflow Indicates that the final value has been reached	0

Note

If a value outside the value range is active at the **BEGIN**, **DELTA** and **END** inputs, (data type DINT: **-2147483631** to **2147483631**), the output **VAL** is set to **NOT_CALCULATED** (output **OVERFLOW** = **0**). The inputs **Clk** and **RESET** are not evaluated. The internal metered value is not changed.
When new valid values are available at **BEGIN**, **DELTA**, and **END**, the internal metered value at output **VAL** is output.

4.10 Time & Clock

You can control functions based on time with the time & clock blocks.

The following time & clock blocks are available:

- **ALARM**
- **BLINK** (blink block)
- **LONG_TIMER** (timer (max. 1,193h))
- **TIMER** (universal timer)
- **TIMER_SHORT** (simple timer)

4.10.1 ALARM

Function

The **Alarm** block indicates that the alarm time specified with the input sizes has been reached by changing from the value 0 to 1 at the **Q** output.

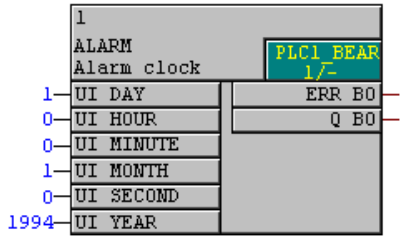
Q remains set for the duration of the matching (1 second).

By using placeholders, time specifications enabling cyclical triggering (annual, monthly, daily, hourly, once per minute) can be implemented.

If no valid time period is specified (e.g. 31 February), this is indicated via the **ERR** output. The **ERR** output remains set until a valid time period is specified again.

Note

Inconsistencies in the time (change from winter to summer) are not taken into account. If the time period is within this time frame, either no double triggering occurs or only one does.



ALARM.tif

Fig. 4-67 ALARM block

I/O assignment

The **ALARM** block has the following I/O assignment:

Table 4-90 I/O assignment of ALARM block

	Name	Data type	Comment	Placeholder	Default setting
Inputs:	DAY	UINT	Day	0	1
	HOUR	UINT	Hour	24	0
	MINUTE	UINT	Minute	60	0
	MONTH	UINT	Month	0	1
	SECOND	UINT	Second	none	0
	YEAR	UINT	Year	0	1994
Outputs:	ERR	BOOL	Invalid date		0
	Q	BOOL	Alarm time period is reached (1 second set)		0

4.10.2 BLINK

Function

The Blink module was designed especially for signal encoding to light sources (e.g. LEDs). For this purpose, it has one input each for the setting of the time for the light and dark phases.

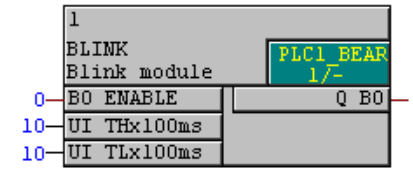
With a rise change from the value 0 to 1 at the **ENABLE** input, the flashing block is started. At the **Q** output, the signal switches from the value 0 to 1 and vice versa, depending on the set times. The flashing block is stopped again with a rise change from the value 1 to 0.

The flashing block always starts with the light phase **THx100ms** and can be interrupted at any time. After a cancellation, the value 0 is always output at the **Q** output of the flasher.

Note

The resolution of the flashing timer is 100 ms. In this way, the first light phase can be up to 99 ms over the set value **THx100ms**, depending on the initial time of the flasher.

The smallest cycle time is 100 ms. This cycle time cannot be undershot for values less than 100 ms for the light phase **THx100ms** or the dark phase **TLx100ms** either. In case of undershooting, the value 100 ms is used automatically.



BLINK.tif

Fig. 4-68 Blink module

Note

For the interconnection of **fast signals** with the left border of the CFC chart in the priority classes measured value processing (priority class **MW_BEARB**) and interlocking (priority class **SFS_BEARB**), the following applies:

Boolean inputs are **no** triggers for this level and remain **unnoticed** when the event or signal is shorter than the processing cycle of the processing level.

For the interconnection of **measured values** with the left border of the CFC in the task levels **fast PLC processing** (task level **PLC_BEARB**) and **slow PLC processing** (task level **PLC1_BEARB**), the following applies:

Measuring inputs are **no** triggers for this level and remain **unnoticed**.

I/O assignment

The **BLINK** module has the following I/O assignment:

Table 4-91 I/O assignment of BLINK module

	Name	Data type	Comment	Default setting
Inputs:	ENABLE	BOOL	Release Starts and stops the flashing block	0
	THx100ms	UINT	Time for the light phase (Resolution 100 ms)	10 (= 1 s)
	TLx100ms	UINT	Time for the dark phase (Resolution 100 ms)	10 (= 1 s)
Outputs:	Q	BOOL	When operation is started, switches between the value 1 and 0 based on the light/dark phase	0

4.10.3 LONG_TIMER

Function

You can implement delays in hours in the **Long Timer** block.

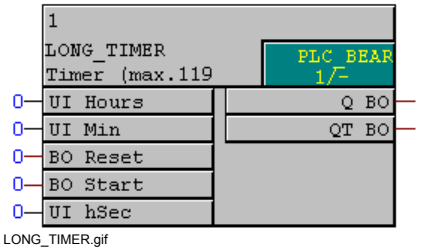


Fig. 4-69 LONG_TIMER block

Note

For the interconnection of **fast signals** with the left border of the CFC chart in the priority classes measured value processing (priority class **MW_BEARB**) and interlocking (priority class **SFS_BEARB**), the following applies:

Boolean inputs are **no** triggers for this level and remain **unnoticed** when the event or signal is shorter than the processing cycle of the processing level.

For the interconnection of **measured values** with the left border of the CFC in the task levels **fast PLC processing** (task level **PLC_BEARB**) and **slow PLC processing** (task level **PLC1_BEARB**), the following applies:

Measuring inputs are **no** triggers for this level and remain **unnoticed**.

I/O assignment

The **LONG_TIMER** block has the following I/O assignment:

Table 4-92 I/O assignment of the LONG_TIMER block

	Name	Data type	Comment	Default setting
Inputs:	Hours	UINT	Value for full hours	0
	Min	UINT	Value for full minutes	0
	Reset	BOOL	Resets timer	0
	hSec	UINT	Value for seconds (Resolution 100 ms)	0
	Start	BOOL	Starts or retriggers timer (at 0-1 transition)	0
Outputs:	Q	BOOL	Timer has elapsed	0
	QT	BOOL	Timer still running	0

Retrigger long timer

The **Long Timer** block can be retriggered:
The running timer is cancelled and restarted with the configured time values via a new rise change from the value 0 to 1 at the **Start** input.

Reset input

The current timer is cancelled with a signal at the reset input **Reset**. The **Q** and **QT** outputs are set to the value 0.

Note

The following applies for SIPROTEC devices with a device version **less than** V4.5:

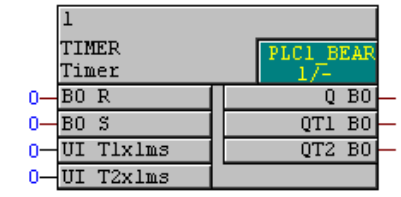
The maximum delay is 1,193 hours.
If you enter a time that is greater than 1,193 hours in the parameters **Hours**, **Min** and **Sec**, an internal fault indication is output in the SIPROTEC device. The runtime is set to 0.

4.10.4 TIMER

Function

The **Universal Timer** block allows you to realize different timer functions:

- Normal Timer Function
- Retriggerable Monoflop Timer Function
- Delay Timer Function
- Pulse-Stretching Timer Function



TIMER.tif

Fig. 4-70 TIMER block

Note

The **Universal Timer** block only functions in the priority classes:

- Fast PLC processing (priority class **PLC_BEARB**) and
- Slow PLC processing (priority class **PLC1_BEARB**).

Note

A simple alternative to the **Universal Timer** block (**TIMER**) is the **simple timer** (**TIMER_SHORT**) block.

The maximum permissible number of **TIMER** and **TIMER_SHORT** blocks is limited by the available system timers and is monitored by the CFC compiler. Observe the technical data in the device manual of the SIPROTEC device which you want to use.

The maximum permissible number is checked during the compilation of the CFC chart. Consistency errors are indicated when a fault occurs. The exceeding of the resource is indicated in the displayed compilation log.

I/O assignment

The **TIMER** block has the following I/O assignment:

Table 4-93 I/O assignment of TIMER block

	Name	Data type	Comment	Default setting
Inputs:	S	BOOL	Set input	0
	R	BOOL	Reset input	0
	T1x1ms	UINT	Value for T1 (Resolution 1 ms)	0
	T2x1ms	UINT	Value for T2 (Resolution 1 ms)	0
Outputs:	Q	BOOL	Q output	0
	QT1	BOOL	T1 output	0
	QT2	BOOL	T2 output	0

Note

The minimum permissible time values for **T1x1ms** and **T2x1ms** are dependent on the time resolution of the SIPROTEC device used. If time values are used that are smaller than the time resolution, the timers do not being running when a start pulse is received. Observe the technical data in the device manual of the SIPROTEC device which you want to use.

The function and circuiting of the individual timer types are described below:

Reset input

The reset input acts in the same way with each timer function (not illustrated in the diagrams):

All the running times are aborted when there is a signal at the Reset input **R**.

The **QT1** and **QT2** outputs are set to the value 0.

The signal at the **S** input is mapped directly to the **Q** output.

Normal timer function

The following applies for the normal timer function:

T1 is started if the signal at the **S** input has a positive edge.

If time **T1** has expired and a signal is present at input **S**, the signal is output at **Q**.

When the signal is output at **Q**, time **T2** can be started with the negative edge of the signal at input **S**.

The times are retriggerable.

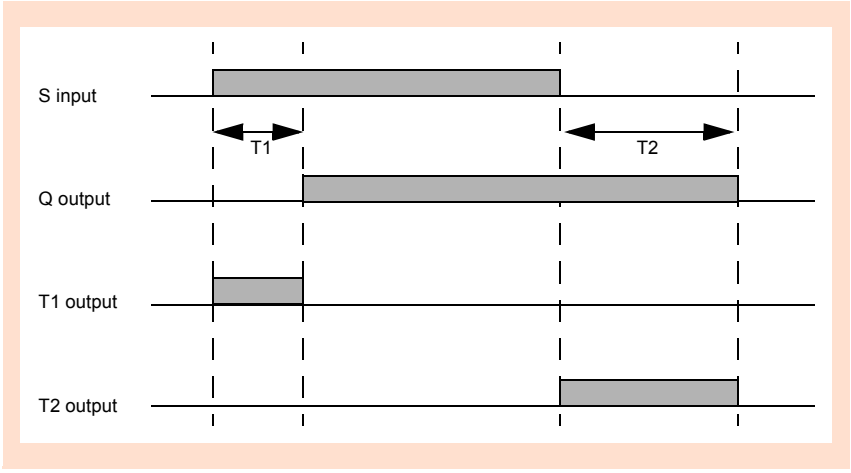


Fig. 4-71 Diagram of the timer function

Retriggerable monoflop timer function

The following applies for the retriggerable monoflop timer function:

T2 is configured to 0.

Only the **T1** output is used.

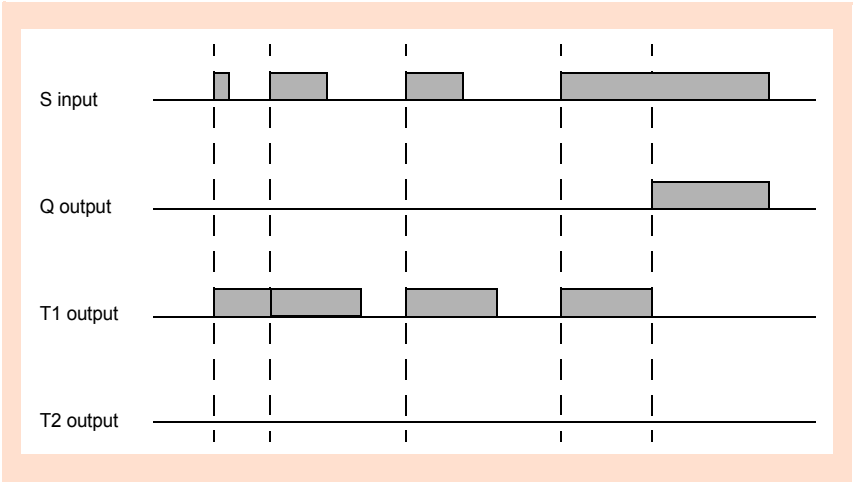


Fig. 4-72 Diagram of the retriggerable monoflop timer function

Delay timer function

The following applies for the delay timer function:

T2 is configured to 0.

Only the **Q** output is used.

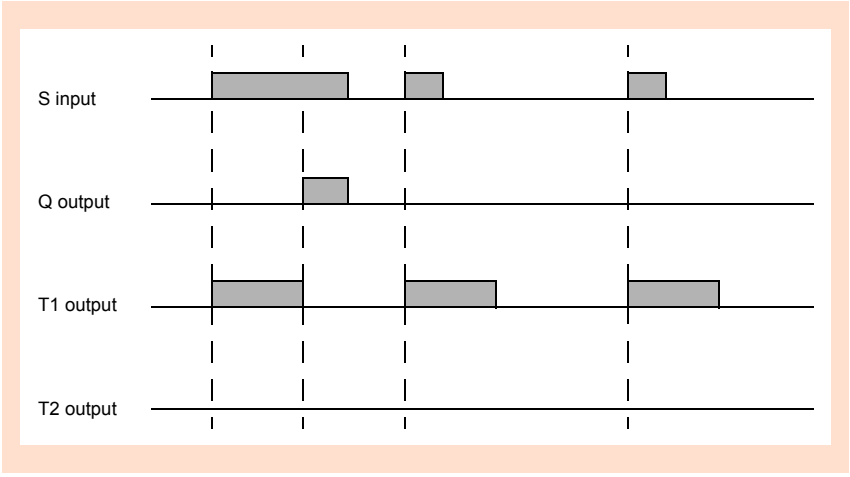


Fig. 4-73 Diagram of the delay timer function

Pulse-stretching timer function

The following applies for the pulse-stretching timer function:

T1 is configured to 0.

Only the **Q** output is used.

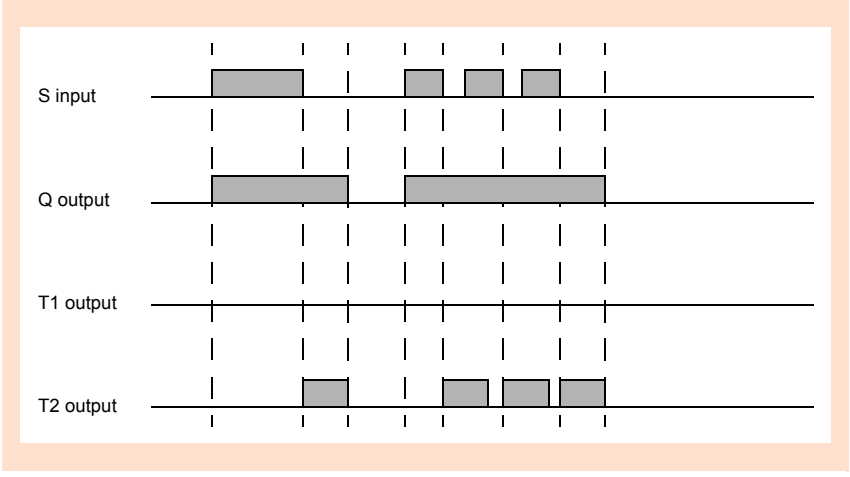


Fig. 4-74 Diagram of the pulse-stretching timer function

Application example

Using the CFC chart shown below, you can implement a simple switching sequence, for example:
If **F1** is pressed, **LED1** illuminates for 5 seconds.
Then **LED2** illuminates for 15 seconds.

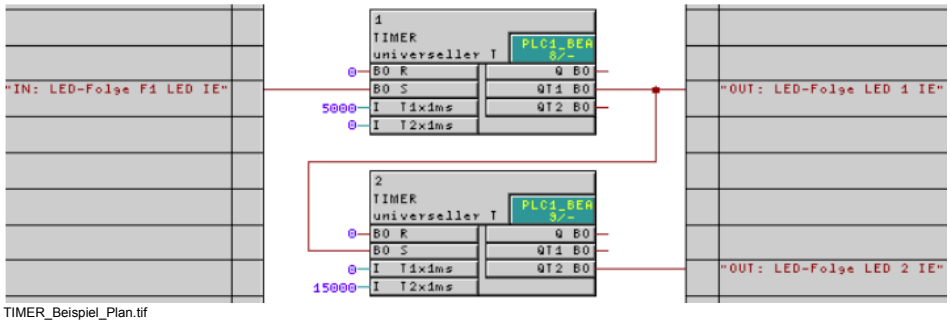
✧ Insert the required information into the configuration based on the following section:

	Information				Quelle					Ziel																							
	Nummer	Displaytext	L	Typ	BE	F	S	C	BA	LED																P	S	C	B	ST		
										1	2	3	4	5	6	7	8	9	10	11	12	13	14									
Fehlerorter																										*						
LED-Folge		F1 LED		IE		1																					X					
		LED 1		IE				X		U																						
		LED 2		IE				X			U																						

TIMER_Beispiel_Matrix.tif

Fig. 4-75 Application example of timer block, configuration section

✧ Implement the following CFC chart in the priority class **PLC1_BEARB** (Slow PLC processing):



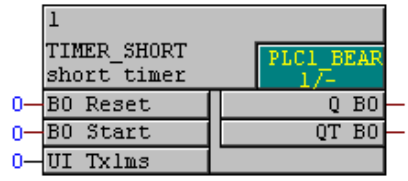
TIMER_Beispiel_Plan.tif

Fig. 4-76 Application example of timer block, CFC chart section

4.10.5 TIMER_SHORT

Function

You can implement simple time tasks (e.g. delays) with the **simple timer** block. Here, a time of up to 65,535 seconds with a resolution of 1 ms can be set.



TIMER_SHORT.tif

Fig. 4-77 TIMER_SHORT block

Note

The **simple timer** block only functions in the priority classes:

- Fast PLC processing (priority class **PLC_BEARB**) and
- Slow PLC processing (priority class **PLC1_BEARB**).

Note

The **simple timer** (**TIMER_SHORT**) block is available as an alternative to the **universal timer** block (**TIMER**).

The maximum permissible number of **TIMER** and **TIMER_SHORT** blocks is limited by the available system timers and is monitored by the CFC compiler. Observe the technical data in the device manual of the SIPROTEC device which you want to use.

The maximum permissible number is checked during the compilation of the CFC chart. Consistency errors are indicated when a fault occurs. The exceeding of the resource is indicated in the displayed compilation log.

I/O assignment

The **TIMER_SHORT** block has the following I/O assignment:

Table 4-94 I/O assignment of TIMER_SHORT block

	Name	Data type	Comment	Default setting
Inputs:	Reset	BOOL	Reset input	0
	Start	BOOL	Set input	0
	Tx1ms	UINT	Time value (Resolution 1 ms)	0
Outputs:	Q	BOOL	Time has run out	0
	QT	BOOL	Time still running	0

Note

The minimum permissible time values for **T1x1ms** are dependent on the time resolution of the SIPROTEC device used. If time values are used that are smaller than the time resolution, the timers do not being running when a start pulse is received. Observe the technical data in the device manual of the SIPROTEC device which you want to use.

Retrigger simple timer

The **simple timer** block can be retriggered:
The running timer is cancelled and restarted with the configured time value via a new rise change from the value 0 to 1 at the **Start** input.

Reset input

The current timer is cancelled with a signal at the reset input **Reset**. The **Q** and **QT** outputs are set to the value 0.

Timer function

The following figure shows a functional diagram of the TIMER_SHORT:

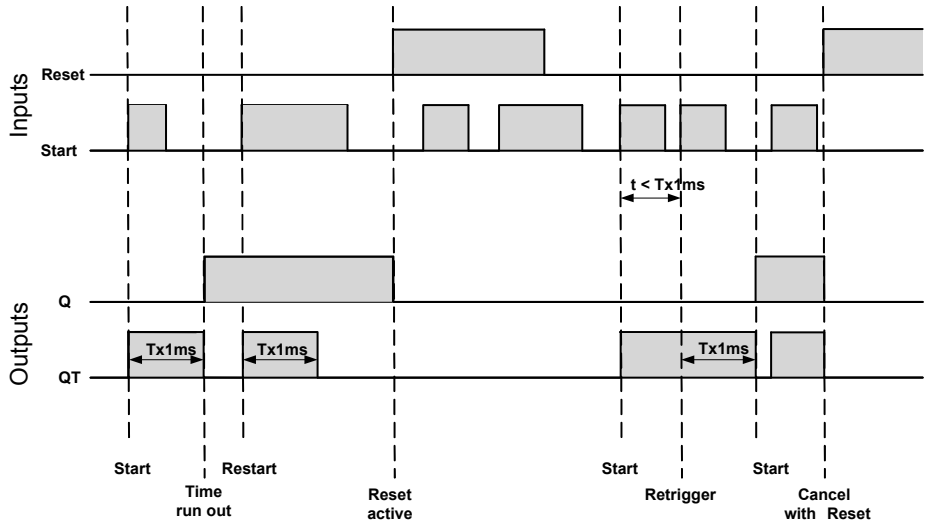


Fig. 4-78 Diagram TIMER_SHORT

Literature

/1/ SIPROTEC, System Description
E50417-H1100-C151

/2/ SIPROTEC DIGSI 4, Start Up
E50417-G1100-C152

Glossary

Bay controllers

Bay controllers are devices with control and monitoring functions without protective functions.

Control display

The display which is displayed on devices with a large (graphic) display after you have pressed the control key is called the control display. It contains the switchgear that can be controlled in the feeder with status display. It is used to perform switching operations. Defining this diagram is part of the configuration.

Cleaning up

Frequent addition and deletion of objects gives rise to memory areas that can no longer be used. By cleaning up projects, you can release these memory areas again. However, a clean up also reassigns the VD addresses. The consequence of that is that all SIPROTEC 4 devices have to be reinitialised.

Combination devices

Combination devices are bay devices with protection functions and a control display.

Combination matrix

Up to 16 compatible SIPROTEC 4 devices can communicate with one another in an Inter Relay Communication combination, (IRC combination). Which device exchanges which information is defined with the help of the combination matrix.

Communication reference CR

The communication reference describes the type and version of a station in communication by PROFIBUS.

Component view

In addition to a topological view, SIMATIC Manager offers you a component view. The component view does not offer any overview of the hierarchy of a project. It does, however, provide an overview of all the SIPROTEC 4 devices within a project.

Container

General term summarizing objects comprising further objects. The object **Folder** is an example of such a container.

Device container

In the Component View, all SIPROTEC 4 devices are assigned to an object of the type **Device container**. This object is also a special object of DIGSI 4 Manager. However, since there is no component view in DIGSI 4 Manager, this object only becomes visible in conjunction with STEP 7.

Field devices

Generic term for all devices assigned to the field level: Protection devices, combination devices, bay controllers.

Folder

This object type is used to create the hierarchical structure of a project.

HV field description

The HV project description file contains details of fields which exist in a ModPara-project. The actual field information of each field is memorized in a HV field description file. Within the HV project description file, each field is allocated such a HV field description file by a reference to the file name.

IEC address

Within an IEC bus a unique IEC address has to be assigned to each SIPROTEC 4 device. A total of 254 IEC addresses are available for each IEC bus.

Initialization string

An initialization string comprises a range of modem-specific commands. These are transmitted to the modem within the framework of modem initialization. The commands can, for example, force specific settings for the modem.

IRC combination

Inter Relay Communication, IRC, is used for directly exchanging process information between SIPROTEC 4 devices. You require an object of type **IRC combination** to configure an Inter Relay Communication. Each user of the combination and all the necessary communication parameters are defined in this object. The type and scope of the information exchanged among the users is also stored in this object.

Link address

The link address indicates the address of a V3/V2-device. You can change this address by means of DIGSI 3.x.

List view

The right pane of the project window displays the names and icons of objects which represent the contents of a container selected in the tree view. Because they are displayed in the form of a list, this area is called the list view.

MLFB Number

MLFB is the abbreviation of the term **Maschinenlesbare Fabrikatebezeichnung** (machine-readable product designation).
This is the equivalent of an order number. The type and version of a SIPROTEC 4 device are coded in the order number.

Modem connection

This object type contains information on both partners of a modem connection, the local modem and the remote modem.

Modem profile

A modem profile consists of the name of the profile, a modem driver and may also comprise several initialization commands and a user address. You can create several modem profiles for one physical modem. To do so you need to link various initialization commands or user addresses to a modem driver and its properties and save them under different names.

Modems

Modem profiles for a modem connection are saved in this object type.

Object

General term specifying each element in a CFC project structure.

Object properties

Each object has properties. These might be general properties that are common to several objects. An object can also have specific properties.

Parameter set

The parameter set is the set of all parameters that can be set for a SIPROTEC 4 device.

Phone Book

User addresses for a modem connection are saved in this object type.

PROFIBUS

PROcess **FI**eld **BUS**, German process and field bus standard that is defined in standard EN 50170, volume 2, PROFIBUS. It defines the functional, electrical, and mechanical properties for a bit-serial field bus.

PROFIBUS address

Within a PROFIBUS network a unique PROFIBUS address has to be assigned to each SIPROTEC 4 device. A total of 254 PROFIBUS addresses are available for each PROFIBUS network.

Project

Content-wise, a project is the image of a real power supply system. *Graphically*, a project is represented by a number of objects which are integrated in a hierarchical structure. *Physically*, a project consists of a series of folders and files containing project data.

Protection devices

All devices with a protective function.

Service Interface

Rear serial interface on the devices for connecting DIGSI 4 (for example, via modem).

Setting Parameters

General term for all adjustments made to the device. Configuration is executed by means of DIGSI 4 or, in some cases, directly on the device.

SIPROTEC

The registered trademark SIPROTEC 4 is used for devices implemented on system base V4.

SIPROTEC 4 device

This object type represents a real SIPROTEC 4 device with all the setting values and process data it contains.

SIPROTEC 4 Variants

This object type represents a variant of an object of type **SIPROTEC 4 device**. The device data of this variant may well differ from the device data of the source object. However, all variants derived from the source object have the same VD address as the source object. For this reason they always correspond to the same real SIPROTEC 4 device as the source object. Objects of type **SIPROTEC 4 variant** have a variety of uses, such as documenting different operating states when entering parameter settings of a SIPROTEC 4 device.

System interface

Rear serial interface on the devices for connecting to a control system via IEC or PROFIBUS.

Topological view

DIGSI 4 Manager always displays a project in the topological view. This shows the hierarchical structure of a project with all available objects.

Tree view

The left pane of the project window displays the names and symbols of all containers of a project in the form of a folder tree. This area is called the tree view.

User address

A user address comprises the name of the station, the national code, the area code and the user-specific phone number.

Users

Up to 16 compatible SIPROTEC 4 devices can communicate with one another in an Inter Relay Communication combination. The individual participating devices are called users.

V3/V2 device

This object type represents a reference to existing data of a device of version V3 or V2.

VD address

The VD address is assigned automatically by DIGSI 4 Manager. It exists only once in the entire project and thus serves to identify unambiguously a real SIPROTEC 4 device. The VD address assigned by DIGSI 4 Manager must be transferred to the SIPROTEC 4 device in order to allow communication with DIGSI 4 Device Editor.

VFD

A VFD (**V**irtual **F**ield **D**evice) includes all communication objects as well as their properties and states, used via services by a communication partner.

Index

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W

X

Z