Xtra Combinations

Building Automation | Industrial Automation | Systems

User Manual

MFD-Titan
Multi-Function Display

06/04 AWB2528-1480GB

Moeller

Think future. Switch to green.

1st published 2003, edition date 06/03
2nd edition 06/2004
See revision protocol in the "About this manual" chapter

Author: Dieter Bauerfeind
Editor: Heidrun Riege
Translator: Terence Osborn

Warning!
Dangerous electrical voltage!

Before commencing the installation

- Disconnect the power supply of the device.
- Ensure that devices cannot be accidentally restarted.
- Verify isolation from the supply.
- Earth and short circuit.
- Cover or enclose neighbouring units that are live.
- Follow the engineering instructions (AWA) of the device concerned.
- Only suitably qualified personnel in accordance with EN 50110-1/-2 (VDE 0105 Part 100) may work on this device/system.
- Before installation and before touching the device ensure that you are free of electrostatic charge.
- The functional earth (FE) must be connected to the protective earth (PE) or to the potential equalisation. The system installer is responsible for implementing this connection.
- Connecting cables and signal lines should be installed so that inductive or capacitive interference does not impair the automation functions.
- Install automation devices and related operating elements in such a way that they are well protected against unintentional operation.
- Suitable safety hardware and software measures should be implemented for the I/O interface so that a line or wire breakage on the signal side does not result in undefined states in the automation devices.
- Ensure a reliable electrical isolation of the low voltage for the 24 volt supply. Only use power supply units complying with IEC 60364-4-41 (VDE 0100 Part 410) or HD 384.4.41 S2.
- Deviations of the mains voltage from the rated value must not exceed the tolerance limits given in the specifications, otherwise this may cause malfunction and dangerous operation.
- Emergency stop devices complying with IEC/EN 60204-1 must be effective in all operating modes of the automation devices. Unlatching the emergency-stop devices must not cause restart.
- Devices that are designed for mounting in housings or control cabinets must only be operated and controlled after they have been installed with the housing closed. Desktop or portable units must only be operated and controlled in enclosed housings.

- Measures should be taken to ensure the proper restart of programs interrupted after a voltage dip or failure. This should not cause dangerous operating states even for a short time. If necessary, emergency-stop devices should be implemented.
- Wherever faults in the automation system may cause damage to persons or property, external measures must be implemented to ensure a safe operating state in the event of a fault or malfunction (for example, by means of separate limit switches, mechanical interlocks etc.).

Contents

About This Manual

This manual describes the installation, commissioning and programming (circuit diagram generation) of the MFD-Titan control relay.

A specialist knowledge of electrical engineering is needed for commissioning and creating circuit diagrams. When active components such as motors or pressure cylinders are controlled, parts of the system can be damaged and persons put at risk if the MFD device is connected or programmed incorrectly.

Device designation

This manual uses the following abbreviated designations for different device models:

- MFD-Titan
- MFD
- easy-AC for EASY618-AC-RC
- easy-DC for EASY6..-DC-.E

Writing conventions

Symbols used in this manual have the following meanings:

▶ Indicates actions to be taken.

Attention!
Warns of the possibility of light damage.

Caution!
Warns of the possibility of serious damage and slight injury.

Warning!
Warns of the possibility of substantial damage, serious injury or death.

→ Indicates interesting tips and additional information

For greater clarity, the name of the current chapter is shown in the header of the left-hand page and the name of the current section in the header of the right-hand page. This does not apply to pages at the start of a chapter and empty pages at the end of a chapter.

List of revisions

Edition	Page	Topic	New	Changed	Removed
06/04	9	Device designation		✓	
	30	Section "Setting values"		✓	
	64	Section "Temperature sensor, brightness sensor, 20 mA sensor"		✓	
	73	Connection cable:		✓	
	74	Section "Cable length and cross-sections"		✓	
	117	Initialise display	✓		
	131	Table 8		✓	
	132	Section "Composition of the markers"		✓	
	194	High-speed counters		✓	
	203	Figure 105		✓	
	226	Section "Function of the GET function block"		✓	
	233	Section "Function of the year time switch function block"		✓	
	250	Section "Accuracy"		✓	
	254	Section "Function of the pulse width modulation function block"		✓	

Edition	Page	Topic	New	Changed	Removed
06/04	261	Section "Time range"		✓	
	262	Section "Variable setpoint values"		✓	
	265	Figure 101		✓	
	266, 267	Figure 135 and Figure 136		✓	
	279	Chapter 5	✓		
	365	Section "Password incorrect or no longer known"		✓	
	391	Section "Memory management of the MFD-Titan"		✓	
	401	Section "Memory card"		✓	
	406	Section "EASY-SOFT-PRO"		✓	
	412	Graphic Dimensions of the MFD-80.. display/operating unit		✓	
	413	Graphic Dimensions of the MFD-CP8.. power supply/CPU module.		✓	
	414	Climatic ambient conditions		✓	
	414	Ambient mechanical conditions		✓	
	415	Electromagnetic compatibility (EMC)		✓	
	415	Dielectric strength		✓	
	416	Operating buttons, input current		✓	

1 MFD-Titan

Target readership

MFD must only be installed and wired up by trained electricians or other persons familiar with the installation of electrical equipment.

A specialist knowledge of electrical engineering is needed for commissioning and creating circuit diagrams. When controlling active components such as motors or pressure cylinders, parts of the system can be damaged and persons put at risk if MFD is connected or programmed incorrectly.

Proper use

MFD is a programmable device that provides HMI, switching, closed-loop and open-loop control functions, and can be used to replace relay and contactor controls as well as being used as an MFD device. MFD must be properly installed before use.

The display and operating unit of the MFD is protected to IP65 and does not normally require any special housing protection. The rear MFD units are designed as mounting units and must be installed in an enclosure, control cabinet or a service distribution board. Both the power feed and the signal terminals must be laid and covered so as to prevent accidental contact.

The installation must comply with regulations for electromagnetic compatibility (EMC).

The power up of the MFD must not cause any hazards arising from activated devices, such as unexpected motor startups or power ups.

Improper use

MFD should not be used as a substitute for safety-related controls such as burner or crane controls, emergency-stop or two-hand safety controls.

Overview

MFD-Titan is an electronic HMI unit and control relay with the following features:

- Logic functions,
- Timing relay and counter functions,
- Time switch functions,
- Arithmetic functions,
- PID controllers,
- Operator and display functions.

MFD-Titan is a display, HMI, control and input device in one. With MFD-Titan you can create solutions for domestic applications as well as for tasks in machine and plant construction. MFD-Titan is a modular and flexible device.

The integral easy-NET network enables the connection of up to eight easy-NET stations to form a single control system. Each easy-NET station can contain a program. This allows the design of systems using high-speed controllers with decentralised intelligence.

In Terminal mode, the MFD can be used to control any device in the network and display data from it, regardless of whether it is an easy800 or an MFD. In this mode, the MFD makes its keypad and display available to the other device for use.

Two devices, MFD with easy800 or MFD, can be connected simply via the serial interfaces.

Circuit diagrams are connected up using ladder diagrams, and each element is entered directly via the MFD display. For example, you can:

- connect make and break contacts in series and in parallel
- switch output relays and auxiliary contacts,
- define outputs as coils, impulse relays, rising or falling edge-triggered relays or as latching relays,
- select timing relays with different functions:
  - on-delayed,
  - on-delayed with random switching,

  - off-delayed,
  - off-delayed with random switching,
  - on and off delayed,
  - on and off delayed with random switching,
  - single pulse,
  - synchronous flashing,
  - asynchronous flashing.
- use up and down counters,
- count high-speed signals:
  - up and down counters with upper and lower limit values,
  - preset,
  - frequency counters,
  - high-speed counters,
  - count incremental encoder values.
- compare values,
- display graphics, texts, variables, enter setpoints, display flashing values and graphics, change and replace graphics and texts by pushbutton,
- process additional inputs and outputs,
- use 7-day and year time switches,
- count operating hours (operating hours counter),
- communicate via the integrated easy-NET network,
- set up point-to-point communication via the serial interface,
- provide closed-loop control with P, PI and PID controllers,
- scale arithmetic values,
- output manipulated variables as pulse-width modulated signals,
- run arithmetic functions:
  - add,
  - subtract,
  - multiply,
  - divide.

- track the flow of current in the circuit diagram,
- load, save and password-protect circuit diagrams.

If you prefer to wire up MFD-Titan from a PC, then use EASY-SOFT-PRO. EASY-SOFT-PRO allows you to create and test your circuit diagram on the PC. All display and operator functions on the MFD device are created exclusively using the EASY-SOFT-PRO software. EASY-SOFT-PRO is also used to print out your circuit diagram in DIN, ANSI or easy format.

Device overview

MFD devices at a glance

Display and operating unit

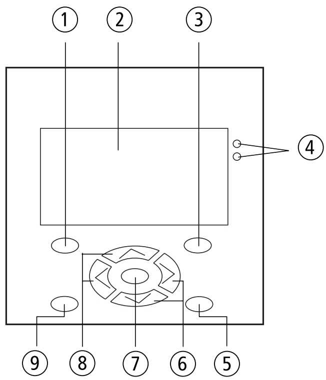


Figure 1: Overview of the display and operating unit

1. DEL button
2. Graphic display
3. ALT button
4. LEDs for signalling
5. Mode button
6. Right, down cursor buttons
7. OK button
8. Left, up cursor buttons
9. ESC button

Power supply and CPU

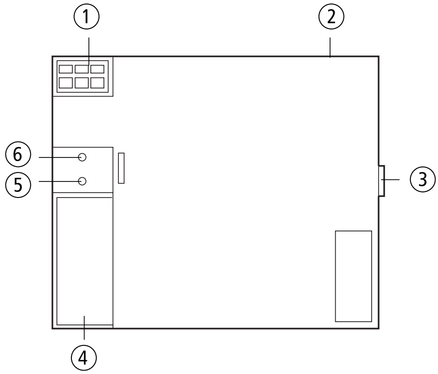


Figure 2: Device overview of power supply and CPU

1. Power supply
2. easy-NET terminals
3. EASY-LINK terminal
4. Interface for memory card, PC and point-to-point connection
5. Power supply / operating mode LED
6. easy-NET LED

Inputs/outputs

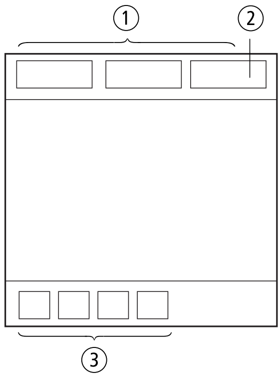


Figure 3: Device overview of inputs/outputs

1. Inputs
2. Analog output (optional)
3. Outputs

Type references for the MFD-Titan

MFD - xxx x - x x - xx

- xx: Additional functions
- x: Equipment
  B = Operator buttons
  ME = Open-loop control, closed-loop control; NT = Open-loop control, closed-loop control, easy-NET
- xxx x: Device function
  Number = display and enclosure in mm
  CP8 = Power supply and CPU rating 8
  R = relay outputs, T = Transistor outputs
  A = Analog output
  Number = Number of inputs and outputs
- MFD: Multi-function display

MFD operation

Buttons

DEL: Delete object in circuit diagram

ALT: Special functions in circuit-diagram, Status display

Cursor buttons ‹ › ˄ ˅:
Move cursor
Select menu items
Set contact numbers, contacts and values

OK: Next menu level, Save your entry

ESC: Previous menu level, Cancel

*: Toggle between visualization display and Status display close Terminal mode

→ In visualization applications, the operating unit can be used for other functions than the ones stated above. In this case the buttons are assigned the function selected in the application. The standard button functions are only restored when you leave the application.

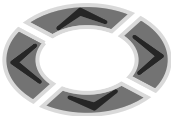
Moving through menus and choosing values

and


Show System menu

Go to next menu level
Select menu item
Activate, modify, save your entry

Return to last menu level
Cancel your entry since the last **OK**



∧ ∨ Change menu item
Change value
< > Change place

Function of P buttons:

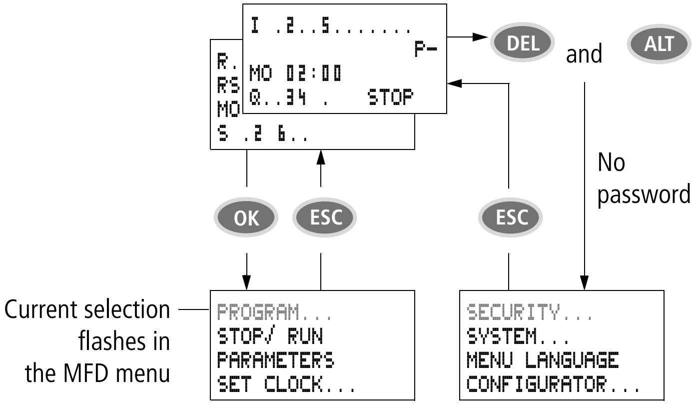
<	Input P1,	∧	Input P2
>	Input P3,	∨	Input P4

and


Reset the MFD display

Selecting main and system menu

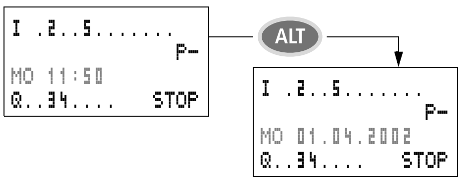
Status display



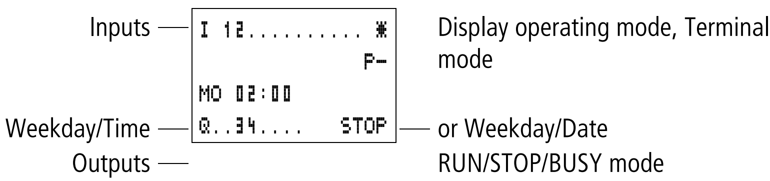
1st menu level
Main menu

1st menu level
System menu

Date display

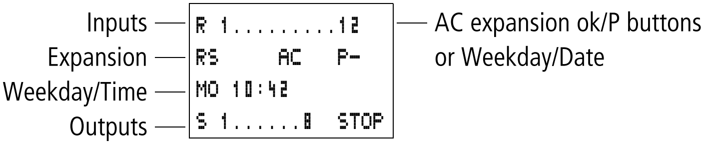


MFD-Titan Status display



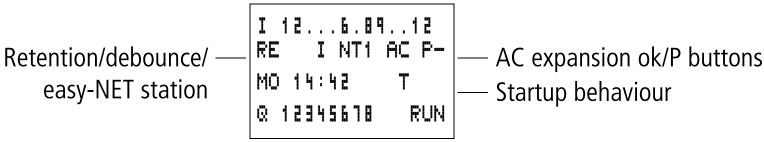
On: 1, 2, 3, 4/Off:…

Status display for local expansion



On: 1, 2, 3, 4/Off:…
RS = Expansion functioning correctly

MFD-Titan advanced Status display



RE : Retention switched on
I : Debounce switched on
NT1 : easy-NET station with station address
COM The COM connection is active
AC : AC expansion functioning correctly
DC : DC expansion functioning correctly
GW : Bus coupling module detected
GW flashes: Only easy200-easy detected. I/O expansion not detected.
ST : When the power supply is switched on, MFD switches to STOP mode

MFD-Titan LED display

MFD-Titan features two LEDs on the back of the MFD-CP... power supply /CPU device. These indicate the status of the power supply (POW) and the RUN or STOP operating mode (→ Fig. 1, Page 16).

Table 1: Power supply/RUN-STOP mode LED

LED OFF	No power supply
LED continuously lit	Power supply present, STOP mode
LED flashing	Power supply present, RUN mode

Table 2: easy-NET LED (easy-NET)

LED OFF	easy-NET not operational, fault, in configuration
LED continuously lit	easy-NET is initialised and no station has been detected.
LED flashing	easy-NET operating fault-free

The front of the MFD-80.. device has a green LED and a red LED. These can be used in your visualization application as light indicators.

The following applies to Terminal mode:

Green LED

Table 3: Power supply/RUN-STOP mode LED

LED OFF	No power supply
LED continuously lit	Power supply present, STOP mode
LED flashing	Power supply present, RUN mode

Red LED

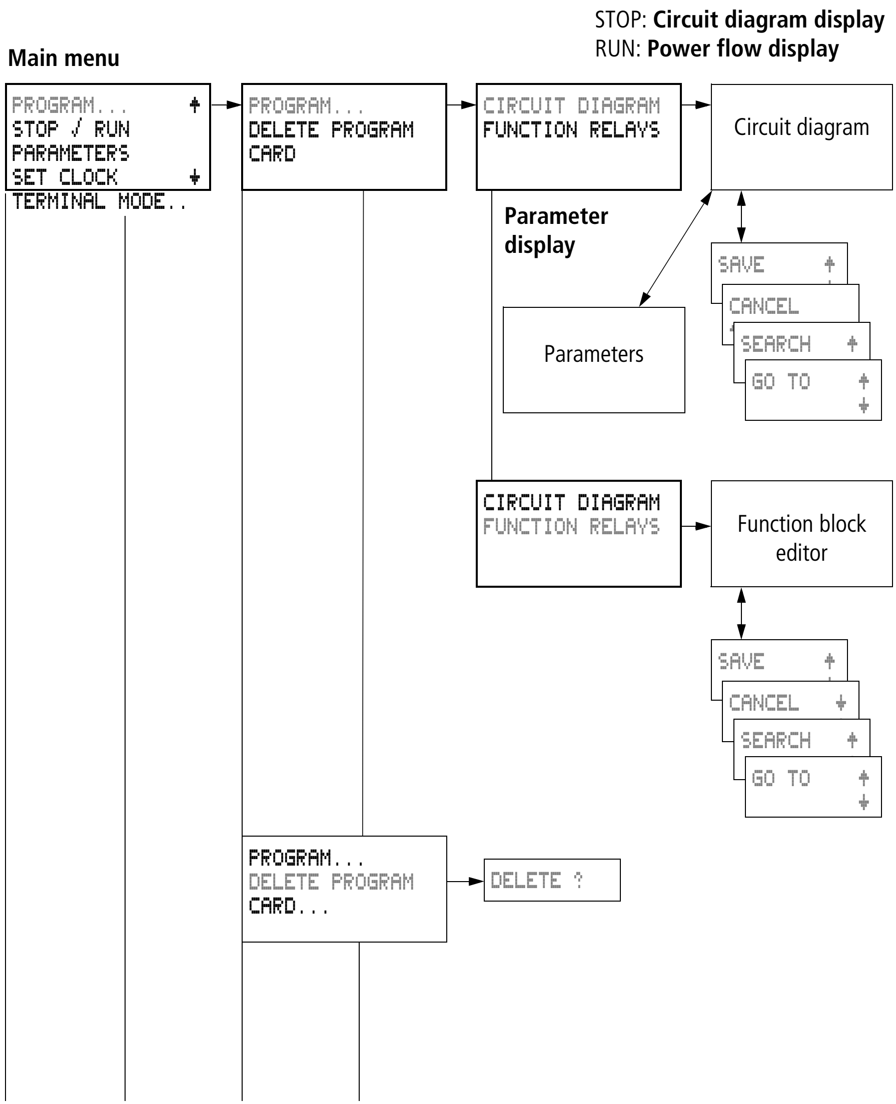
Table 4: Fault on the easy-NET

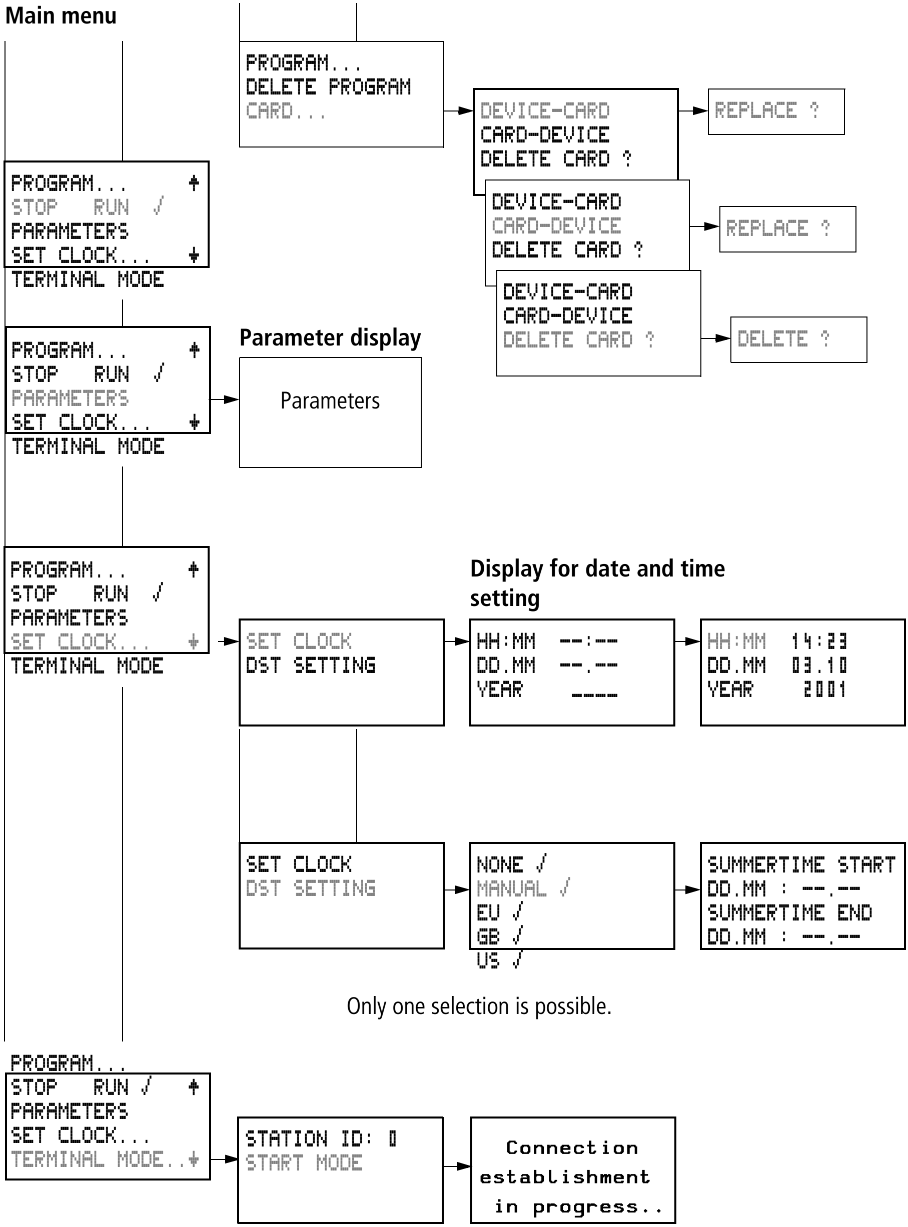
LED OFF	Operation correct
LED continuously lit	easy-NET remote device faulty

Menu structure

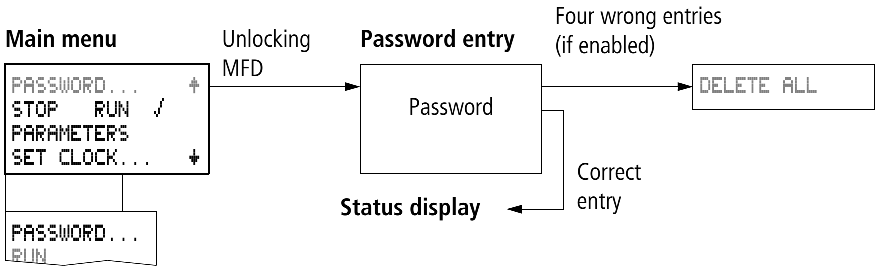
Main menu without password protection

▶ You access the main menu by pressing **OK**.



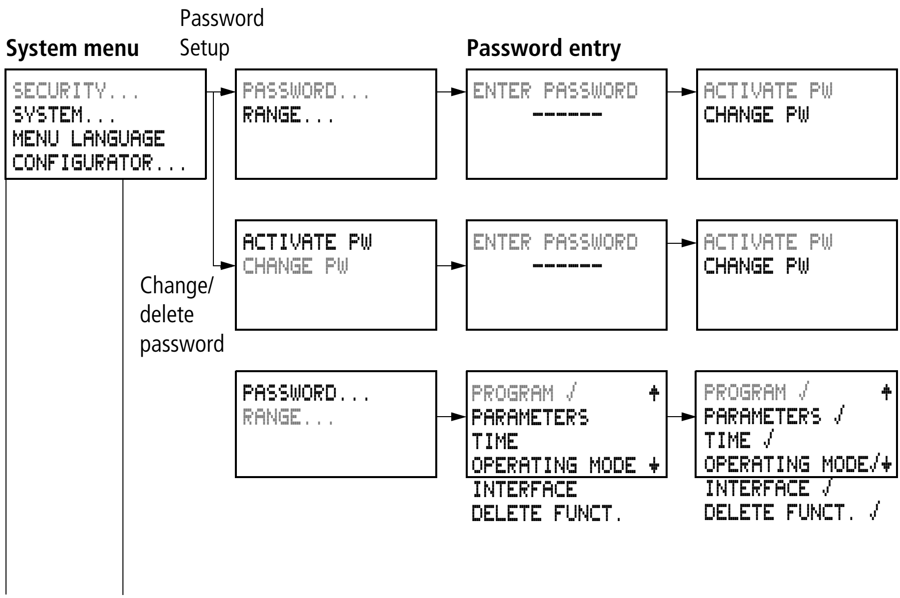

Main menu
PROGRAM...
DELETE PROGRAM
CARD...
DEVICE-CARD
CARD-DEVICE
DELETE CARD ?
REPLACE ?
DEVICE-CARD
CARD-DEVICE
DELETE CARD ?
REPLACE ?
DEVICE-CARD
CARD-DEVICE
DELETE CARD ?
DELETE ?
PROGRAM...
STOP RUN √
PARAMETERS
SET CLOCK...
TERMINAL MODE
Parameter display
Parameters
Display for date and time setting
SET CLOCK
DST SETTING
HH:MM --:--
DD.MM --.--
YEAR ____
HH:MM 14:23
DD.MM 03.10
YEAR 2001
NONE √
MANUAL √
EU √
GB √
US √
SUMMERTIME START
DD.MM : --.--
SUMMERTIME END
DD.MM : --.--
Only one selection is possible.
STATION ID: 0
START MODE
Connection
establishment
in progress..

Main menu with password protection

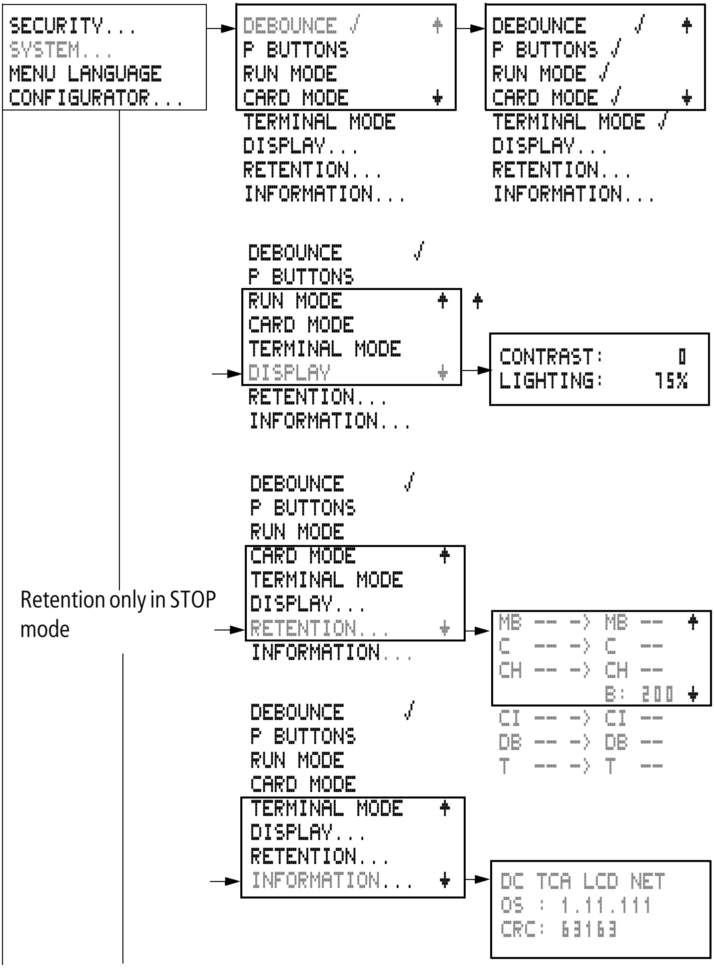


MFD-Titan system menu

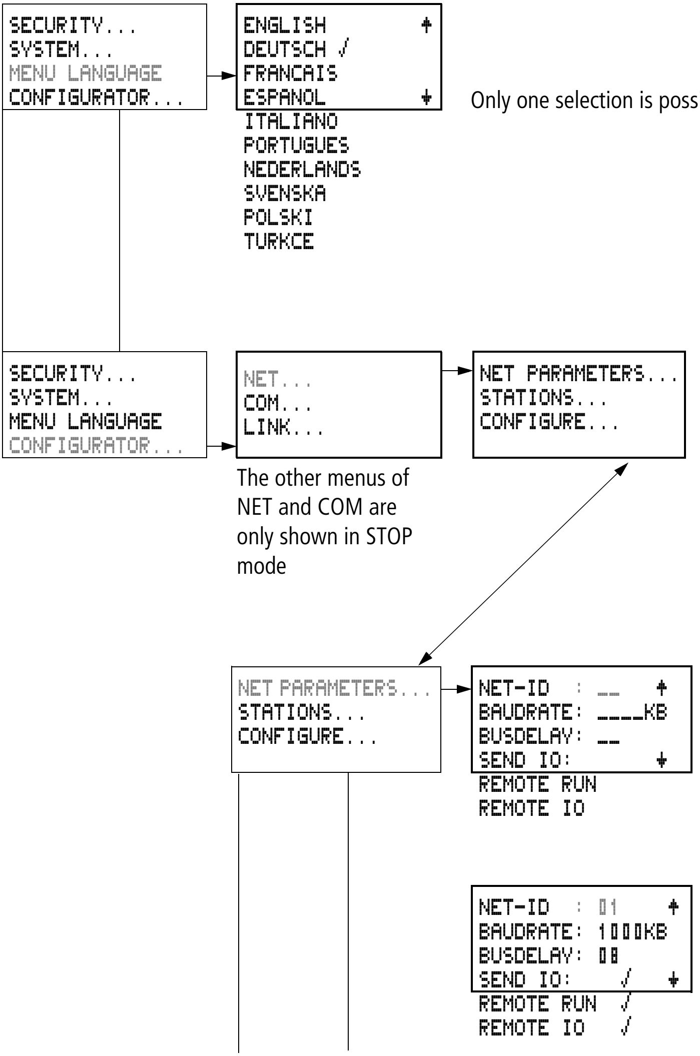
▶ The System menu is accessed by simultaneously pressing **DEL** and **ALT**.



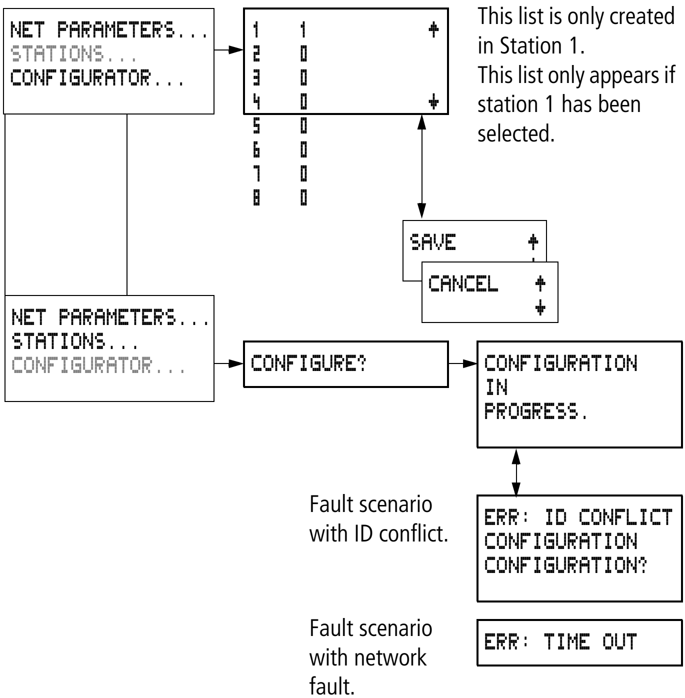
System menu


SECURITY...
SYSTEM...
MENU LANGUAGE
CONFIGURATOR...
DEBOUNCE √
P BUTTONS
RUN MODE
CARD MODE
TERMINAL MODE
DISPLAY...
RETENTION...
INFORMATION...
DEBOUNCE √
P BUTTONS √
RUN MODE √
CARD MODE √
TERMINAL MODE √
DISPLAY...
RETENTION...
INFORMATION...
DEBOUNCE √
P BUTTONS
RUN MODE
CARD MODE
TERMINAL MODE
DISPLAY
RETENTION...
INFORMATION...
CONTRAST: 0
LIGHTING: 75%
DEBOUNCE √
P BUTTONS
RUN MODE
CARD MODE
TERMINAL MODE
DISPLAY...
RETENTION...
INFORMATION...
Retention only in STOP mode
MB -- -> MB --
C -- -> C --
CH -- -> CH --
B: 200
CI -- -> CI --
DB -- -> DB --
T -- -> T --
DEBOUNCE √
P BUTTONS
RUN MODE
CARD MODE
TERMINAL MODE
DISPLAY...
RETENTION...
INFORMATION...
DC TCA LCD NET
OS : 1.11.111
CRC: 63163

System menu


SECURITY...
SYSTEM...
MENU LANGUAGE
CONFIGURATOR...
ENGLISH
DEUTSCH √
FRANCAIS
ESPANOL
ITALIANO
PORTUGUES
NEDERLANDS
SVENSKA
POLSKI
TURKCE
Only one selection is possible.
SECURITY...
SYSTEM...
MENU LANGUAGE
CONFIGURATOR...
NET...
COM...
LINK...
The other menus of NET and COM are only shown in STOP mode
NET PARAMETERS...
STATIONS...
CONFIGURE...
NET PARAMETERS...
STATIONS...
CONFIGURE...
NET-ID : __
BAUDRATE: ____KB
BUSDELAY: __
SEND IO:
REMOTE RUN
REMOTE IO
NET-ID : 01
BAUDRATE: 1000KB
BUSDELAY: 08
SEND IO: √
REMOTE RUN √
REMOTE IO √

System menu


NET PARAMETERS...
STATIONS...
CONFIGURATOR...
1 1
2 0
3 0
4 0
5 0
6 0
7 0
8 0
This list is only created in Station 1.
This list only appears if station 1 has been selected.
SAVE
CANCEL
NET PARAMETERS...
STATIONS...
CONFIGURATOR...
CONFIGURE?
CONFIGURATION
IN
PROGRESS.
Fault scenario with ID conflict.
ERR: ID CONFLICT
CONFIGURATION
CONFIGURATION?
Fault scenario with network fault.
ERR: TIME OUT

System menu

```
NET...
COM...
CONFIGURE...
```

→

```
BAUDRATE: 9600B
COM-LINK
REMOTE MARKER...
```

→

```
BAUDRATE:19200B
COM-LINK       √
REMOTE MARKER...
```

```
BAUDRATE:19200B
COM-LINK       √
REMOTE MARKER...
```

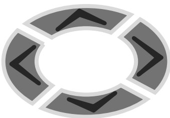
→

```
READ:
 1MD00    1MD00
WRITE:
 1MD00    1MD00
```

This menu only appears if COM-LINK was selected.

Selecting or toggling between menu items

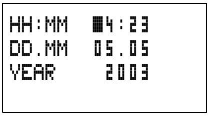
```
PROGRAM...
STOP
PARAMETERS
SET CLOCK...
```



Cursor ⋀ ⋁

Select or switch

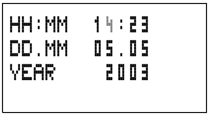
Cursor display



The cursor flashes.

Full cursor ▮/:

- Move cursor with ‹ ›,
- in circuit diagram also with ˄ ˅

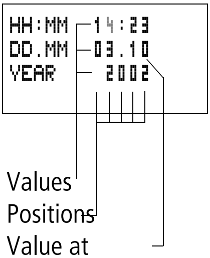


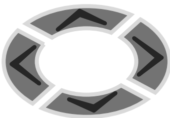
Value M/M

- Change position with ‹ ›
- Change values with ˄ ˅

Flashing values/menus are shown in grey in this manual.

Setting values





Change value ˄ ˅

Select cursor position in value ‹ ›

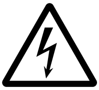
Change value at position ˄ ˅

Store entries

Retain previous value

2 Installation

The MFD must only be installed and wired up by qualified electricians or other persons familiar with the installation of electrical equipment.



Danger of electric shock!
Never carry out electrical work on the device while the power supply is switched on.

Always follow the safety rules:

- Switch off and isolate,
- Ensure that the device is no longer live,
- Secure against reclosing,
- Short-circuit and ground,
- Cover adjacent live parts.

The MFD is installed in the following order:

- Mounting,
- Wiring up the inputs,
- Wiring up the outputs,
- Wiring up the NET network (if required),
- Setting up the serial interface (if required),
- Connecting the power supply.

Mounting

Install the display/operating unit of the MFD in the front of a control cabinet, a service distribution board, operator panel or in an enclosure. Install the power supply/CPU module and the input/output module so that all the terminals are protected against direct contact, liquids and dust during operation.

When using the MFD without a display/operating unit, snap it onto a IEC/EN 60715 top-hat rail or fix it in place using fixing brackets. The MFD can be mounted either vertically or horizontally.

→ When using the MFD with expansion units, connect the expansion concerned before mounting (see → page 49).

For ease of wiring, leave a gap of at least 3 cm between the MFD terminals and the wall or adjacent devices.

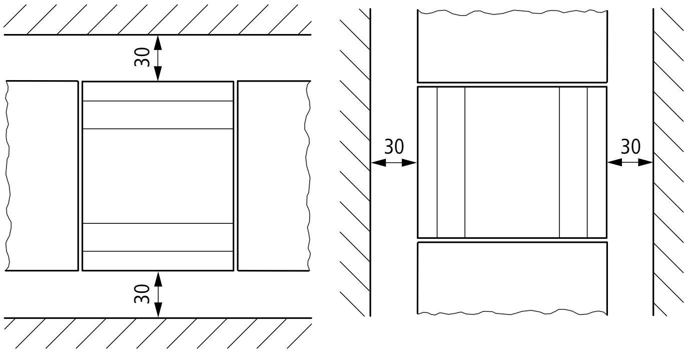


Figure 4: Clearances to the MFD

Fitting the protective membrane

For special applications such as in the food industry, the operating unit must be protected against the ingress of dust, liquids etc.

For this use the specially designed protective membrane.

Fit the protective membrane before mounting the display/ operating unit.

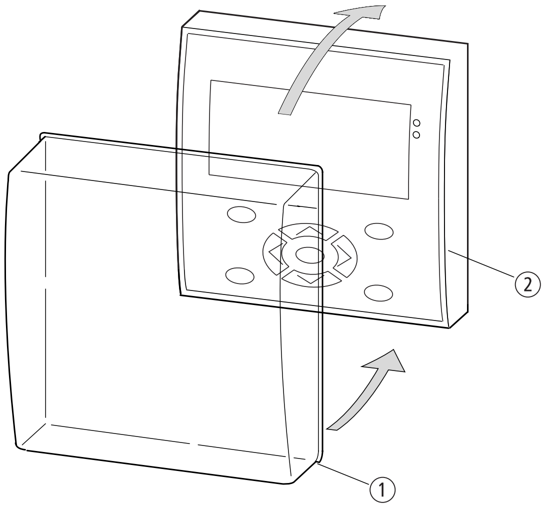


Figure 5: Fitting the protective membrane

① Protective membrane

② Display/operating unit

▶ Place the protective membrane over the display/operating unit.

Caution!
Ensure that the membrane fits snugly in the groove of the display/operating unit.

Otherwise a proper seal cannot be guaranteed and particles may enter underneath the membrane. This may cause malfunctions in the keypad.

In food industry applications, there is the risk of bacteria building up underneath the membrane.

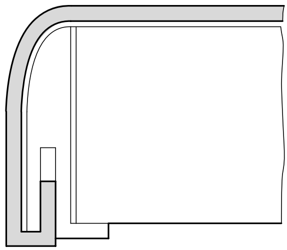


Figure 6: Correct position of the protective membrane

→ If the protective membrane has to be replaced, the display and the operating unit have to be removed. Replace the membrane and refit the device.

Mounting the protective cover

The protective cover is provided for using the device in aggressive environments. This protects the display and the operating unit against mechanical damage or destruction. Protection to IP65 is maintained.

The protective cover can be opened so that the operating unit can be used.

The protective cover can be closed with a sealing facility to provide protection against unauthorised operation.

Before mounting the display/operating unit, fit the protective cover.

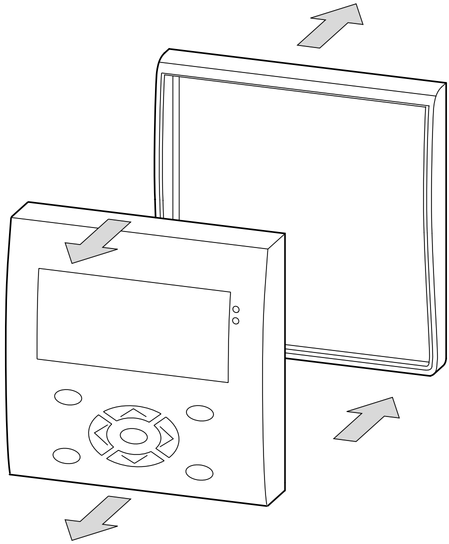


Figure 7: Removing the front frame

▶ Remove the front frame as shown in the figure.

The protective cover can be mounted in two different positions. Choose the position that is most suitable for the application at hand and your requirements.

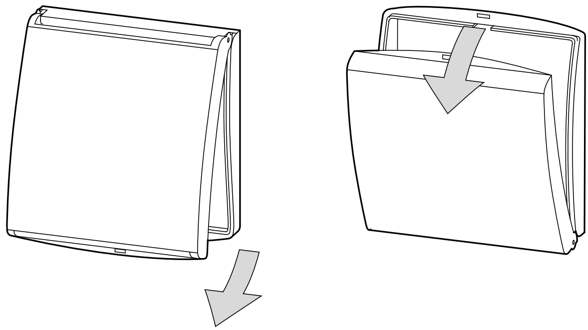


Figure 8: Position of the protective cover

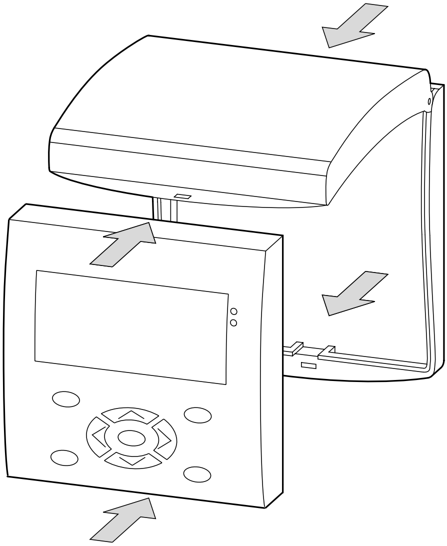


Figure 9: Mounting the protective cover

▶ Mount the protective cover as shown in the figure.

Sealing the protective cover

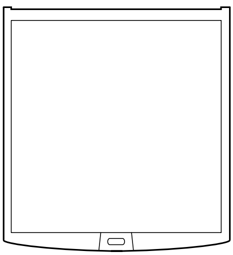


Figure 10: Sealing the protective cover

The grip handle of the protective cover is provided with holes that can be used in any mounting position. You can fit a wire or similar material through these holes in order to seal the cover. If the wire is provided with a lead seal, the cover is sealed. The cover can then only be opened by breaking the seal or the wire.

Mounting the display/operating unit, “front mounting”

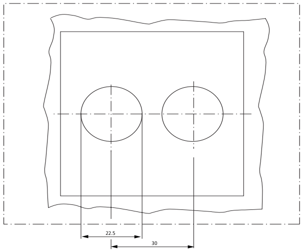


Figure 11: Drill holes for the MFD

▶ Drill and punch out two 22.5 mm diameter holes. The diameter is the same as is normally required for control circuit devices.

→ Observe the following technical requirements:

- The hole spacing is 30 mm.
- The maximum thickness of the front plate for mounting the power supply/CPU module must not be more than 6 mm.
- The maximum thickness of the front plate for mounting an expansion unit with a top-hat rail in addition to the power supply/CPU module must not be more than 4 mm.
- Leave enough space at the side for the power supply/ CPU module, and if necessary, the expansion unit.
- In order to ensure protection to IP65, the surface of the mounting front must be even and smooth.

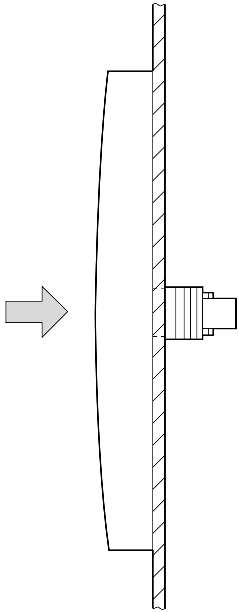


Figure 12: Mounting the display/operating unit

The protective membrane or the protective cover must be fitted.

▶ Fit the display/operating unit in the punched fixing holes.

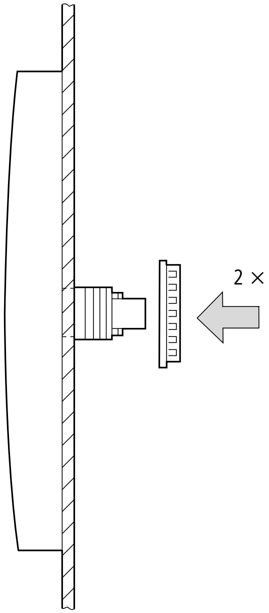


Figure 13: Screw fastening the display/operating unit

▶ Screw fasten the display/operating unit.

The tightening torque must be between 1.2 and 2 Nm

→ Ensure that the correct torque is used. If the tightening torque is too low or high, this may impair the seal.

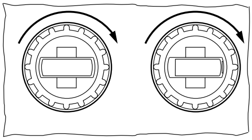


Figure 14: Mounting the display/operating unit, front mounting

Use the combination box spanner with the designation M22-MS.

Removing the display/operating unit, "front mounting"

▶ Unscrew the fixing element and remove the display/operating unit.

Mounting the power supply/CPU module

If you wish to add expansion units to the power supply/CPU module, the top-hat rail must be fitted beforehand.

Fitting the top-hat rail

Ensure that the cutout of the top-hat rail was prepared for the fixing shafts according to the specified dimensions.

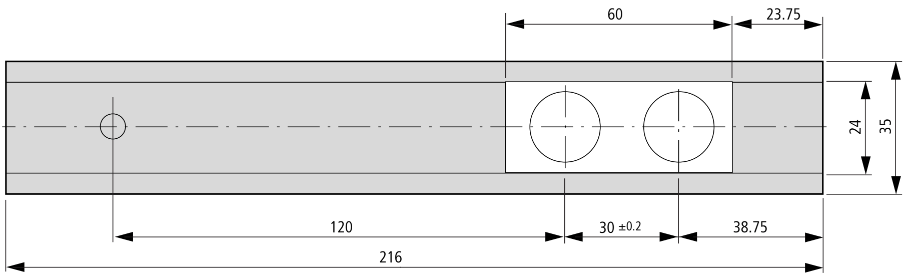


Figure 15: Top-hat rail with cutout

→ The two fixing shafts of the display/operating unit are designed for a 2 space unit expansion device.

If you wish to fit wider expansion units, the top-hat rail must be supported at a third support point.

This third support point should be located in the area 216 mm from the end of the device. It should not be possible to twist the top-hat rail.

Caution!
The fixing shafts of the display/operating unit are designed for mounting the expansion units. Other devices such as contactors must not be mounted on this top-hat rail.

Attach the expansion unit before fitting the top-hat rail.

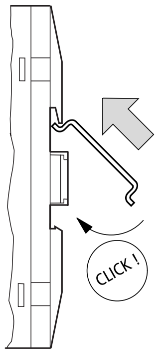


Figure 16: Fitting the top-hat rail

- ▶ Fit the top-hat rail in the groove using the slide catch of the power supply/CPU module and the expansion unit.
- ▶ Turn the top-hat rail towards the housing.
- ▶ Let the top-hat rail snap into position.
- ▶ Press the power supply/CPU module onto the fixing shaft.

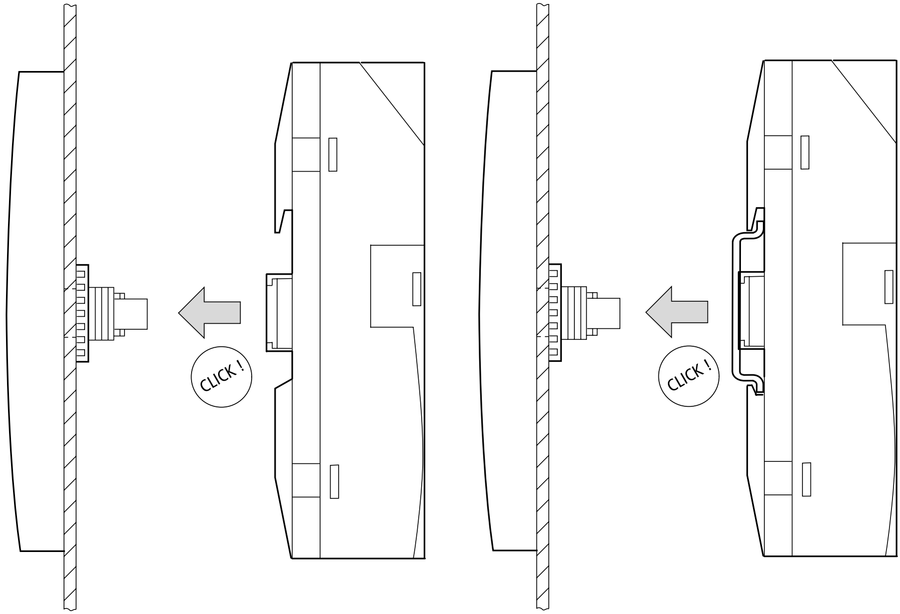


Figure 17: Fitting the CPU with and without top-hat rail

Mounting the inputs/outputs onto the power supply/ CPU module

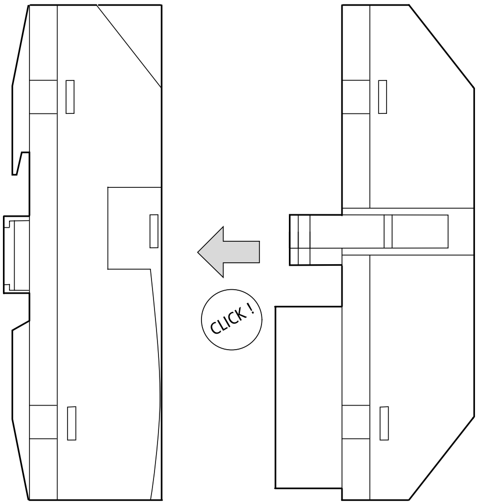


Figure 18: Plugging in the I/O module

→ The inputs/outputs can be mounted before or after mounting the power supply/CPU module onto the fixing shaft.

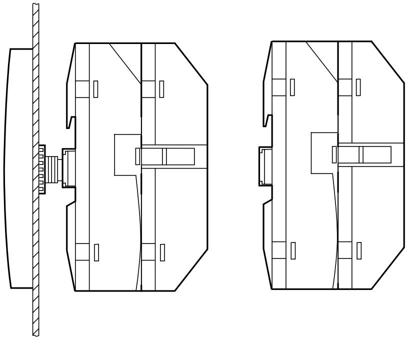


Figure 19: CPU with I/O module

Removing the inputs/outputs

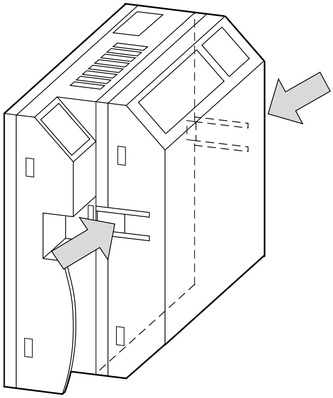


Figure 20: Releasing the I/O module

- ▶ Press the two catches together.
- ▶ Pull one side out of the catch.
- ▶ Pull the other side out of the second catch.

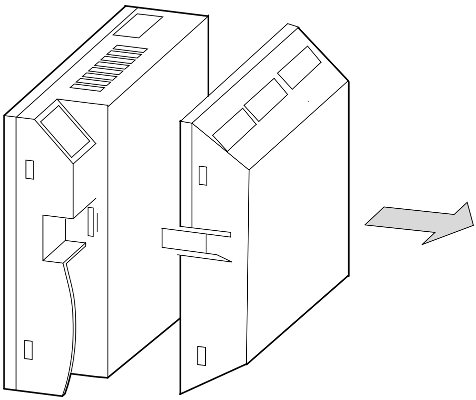


Figure 21: Removing the I/O module

- ▶ Remove the I/O module.

Removing the power supply/CPU module

The power supply/CPU module can be removed with or without the I/O module.

→ If there is another fixing point for the top-hat rail, apart from the one for the display/operating unit, undo it.

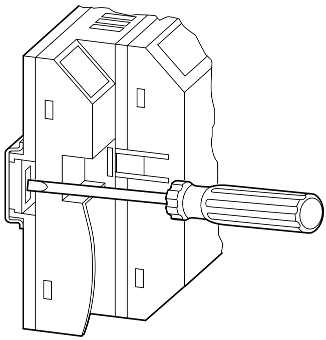


Figure 22: Releasing the fixing shaft

Use a screwdriver with a 100 x 3.5 mm slot width.

▶ Insert the screwdriver into the lug of the fixing shaft catch.

▶ Lever out the slide catch.

▶ Pull out the power supply/CPU module from the fixing shafts.

Mounting on top-hat rail

The power supply/CPU module can be mounted on a top-hat rail without the display/operating unit.

The fastening catches must be removed in order to mount the device on a fastened top-hat rail.

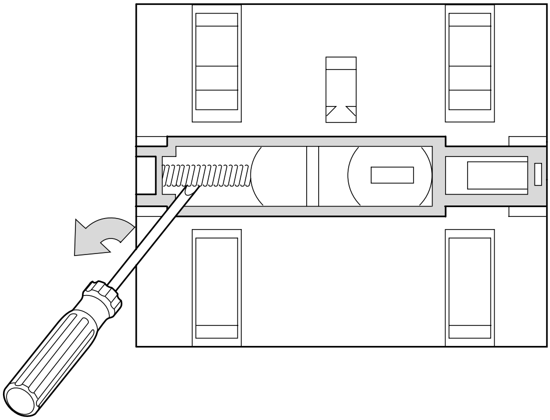


Figure 23: Removing the spring

▶ Remove the spring with a screwdriver.

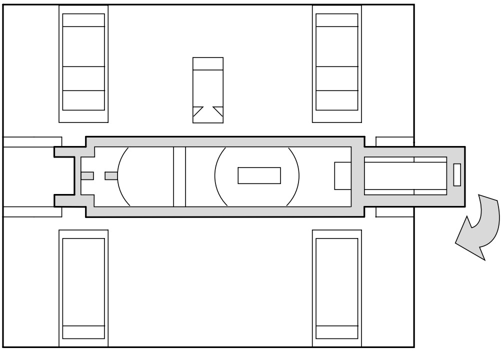
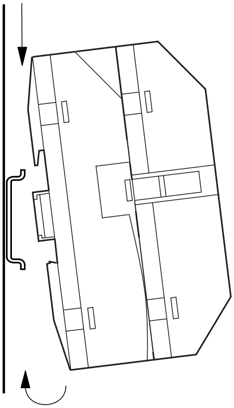


Figure 24: Pulling out and removing the slide catch

▶ Pull the slide catch out of the guide and remove it.



▶ Hook the MFD to the top edge of the top-hat rail and hinge into place while pressing down slightly. Press down lightly on both the device and the top-hat rail until the unit snaps over the lower edge of the top-hat rail.

The MFD will clip into place automatically.

▶ Check that the device is seated firmly.

The device is mounted vertically on a top-hat rail in the same way.

Screw mounting

For screw mounting on a mounting plate, fixing brackets must be used that can be fixed to the back of the MFD. The fixing brackets are available as an accessory. The power supply/CPU module can be screwed onto a mounting plate without the display/operating unit.

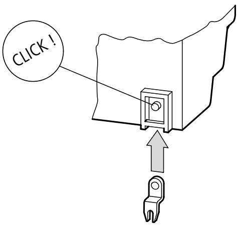


Figure 25: Inserting a fixing bracket

→ Three fixing brackets are sufficient for a device with four fixing points.

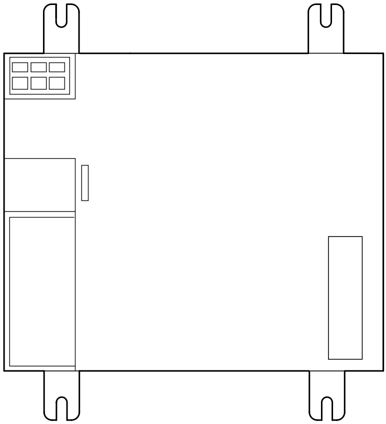


Figure 26: Screw mounting for the MFD

EASY2..-..: easy600:

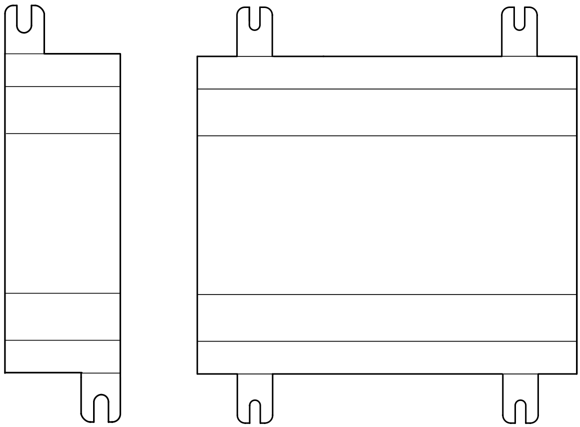


Figure 27: Screw mounting for easy

Connecting the expansion unit

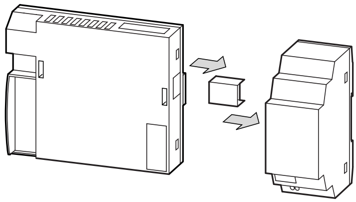


Figure 28: Connecting expansion units

Terminals

Tool for cage clamp terminals

Slot-head screwdriver, width 3.5 × 0.6 mm.

Connection cross-sections of the MFD cage clamp terminal cables

- Solid: 0.2 to 4 mm^2 (AWG 24 -12)
- Flexible with ferrule: 0.2 to 2.5 mm^2 (AWG 24-12)

Tool for slot-head screws for easy expansion unit

Slot-head screwdriver, width 3.5 × 0.6 mm, tightening torque 0.6 Nm.

Connection cross-sections of screw terminal cables

- Solid: 0.2 to 4 mm^2 (AWG 22 - 12)
- Flexible with ferrule: 0.2 to 2.5 mm^2 (AWG 22 -12)

Network cables and plugs

If possible use the prefabricated EASY-NT cables (e.g. 30 or 80 cm).

Other cable lengths can be made using the EASY-NT-CAB cable, the EASY-NT-RJ45 plug and the easy-RJ45-TOOL crimping tool.

AWG 24, 0.2 mm^2 are the largest crimpable cross-sections.

The first and last stations in the network must be provided with an EASY-NT-R bus termination resistor.

Connecting the power supply

→ The required connection data for both MFD-DC, **easy-DC** device types with 24 V DC and MFD-AC, **easy-AC** with standard voltages of 100 V to 240 V AC are given in the Chapter "Technical data", Page 409.

The MFD-Titan devices run a system test for one second after the power supply has been switched on. Either RUN or STOP mode will be activated after this time depending on the default setting.

MFD-AC power supply

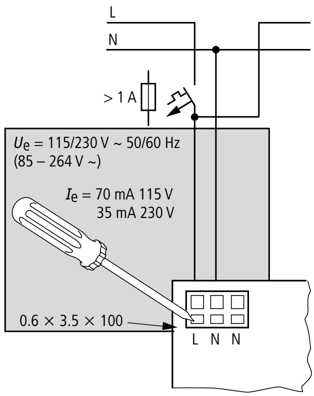
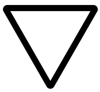


Figure 29: AC power supply on the MFD



Attention!
A short current surge will be produced when switching on for the first time. Do not switch on AC via Reed contacts since these may burn or melt.

EASY…-AC-.E expansion units

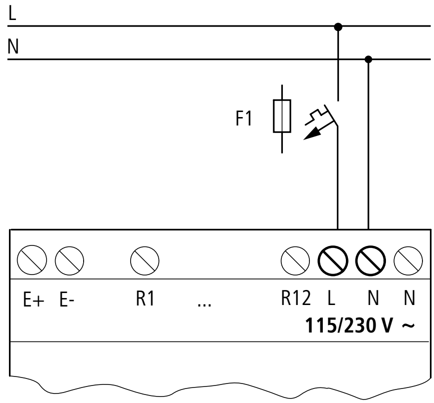


Figure 30: Power supply on the AC expansion units

Attention!
A short current surge will be produced when switching on for the first time. Do not switch on easy AC via Reed contacts since these may burn or melt.

MFD DC power supply

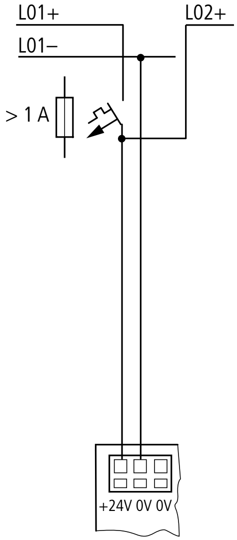


Figure 31: Power supply on the MFD

The MFD power supply/CPU module supplies the necessary power supply to itself, the display, the input/output electronics, the easy-LINK, and optionally the easy-NET.

→ The MFD device power supply/CPU module is protected against reverse polarity. Ensure the correct polarity of the terminals to ensure that the MFD functions correctly.

EASY…-DC-.E DC expansion units

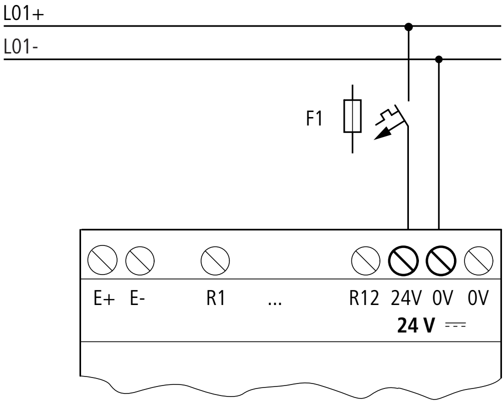


Figure 32: Power supply on the DC expansion units

→ easy DC is protected against polarity reversal. To ensure that easy works correctly, ensure that the polarity of each terminal is correct.

Cable protection

With easy-AC, easy-DC and the MFD power supply/CPU module, provide cable protection (F1) for at least 1 A (slow).

→ When easy or the MFD is switched on for the first time, its power supply circuit behaves like a capacitor. Use a suitable device for switching on the power supply and do not use any reed relay contacts or proximity switches.

Connecting the inputs

easy or MFD inputs switch electronically. Once you have connected a contact via an input terminal, you can reuse it as a contact in your MFD circuit diagram as often as you like.

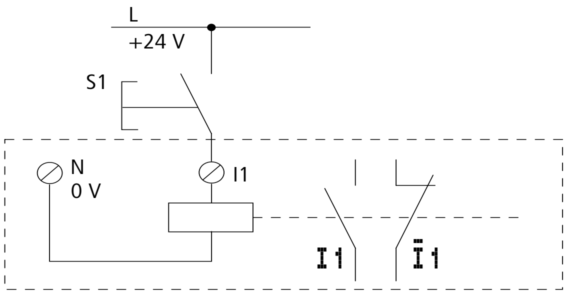


Figure 33: Connecting the inputs

Connect contacts such as pushbutton actuators or switches to easy or MFD input terminals.

Connecting the AC inputs

Caution!
Connect up the AC inputs in accordance with the safety requirements of the VDE, IEC, UL and CSA with the same phase conductor that provides the power supply. Otherwise easy will not detect the switching level and may be damaged or destroyed by overvoltage.

MFD-AC basic unit

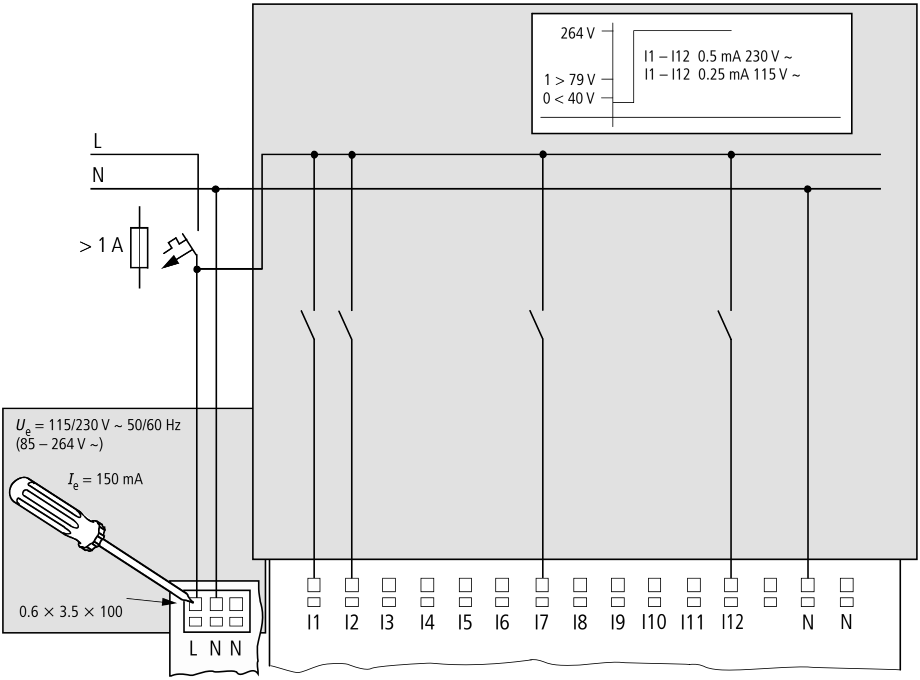


Figure 34: Connecting the inputs

Warning!
The AC inputs must only be used with MFD-AC-CP8... devices. Other devices may be destroyed.

AC expansion device

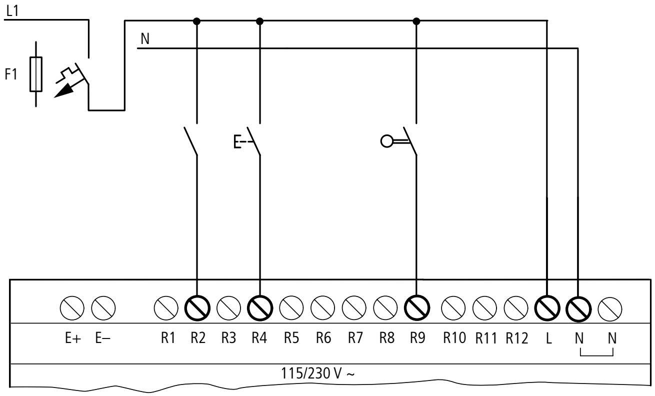


Figure 35: Inputs on the EASY…-AC-.E expansion device

Connect the inputs, for example, to pushbutton actuators, switches or relay/contactor contacts.

Input signal voltage range

- OFF signal: 0 to 40 V
- ON signal: 79 to 264 V

Input current

- I1 to I12 0.5 mA/0.2 mA at 230 V/115 V
- R1 to R12
  0.5 mA/0.25 mA at 230 V/115 V

Cable lengths

Severe interference can cause a "1" signal on the inputs without a proper signal being applied. Observe therefore the following maximum cable lengths:

- I1 to I12 and
- R1 to R12: 40 m without additional circuit

The following applies to expansion units:
With longer cables, connect a diode (e.g. 1N4007) for 1 A, minimum 1 000 V reverse voltage, in series to the easy input.

Ensure that the diode is pointing towards the input as shown in the circuit diagram, otherwise easy will not detect the 1 state.

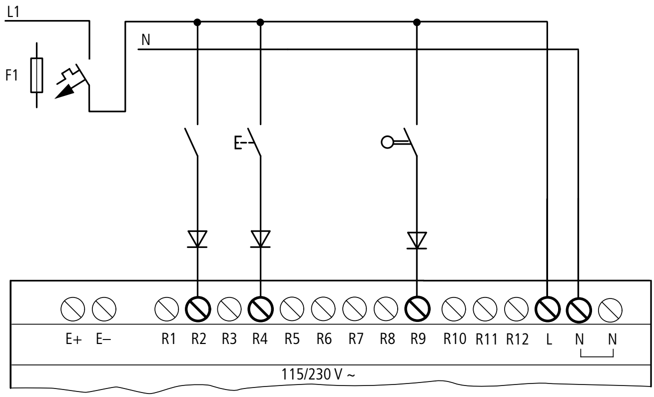


Figure 36: easy-AC with a diode on the inputs

Two-wire proximity switches have a residual current with the “0” state. If this residual current is too high, the easy input may detect a “1” signal.

If inputs with a higher input current are required, an additional input circuit must be used.

Increasing the input current

The following input circuit can be used in order to prevent interference and also when using two-wire proximity switches:

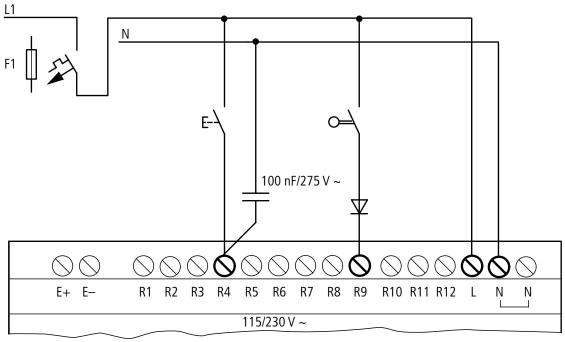


Figure 37: Increasing the input current

→ When using a 100 nF capacitor, the drop-out time of the input increases by 80 (66.6) ms at 50 (60) Hz.

A resistor can be connected in series with the circuit shown in order to restrict the inrush current.

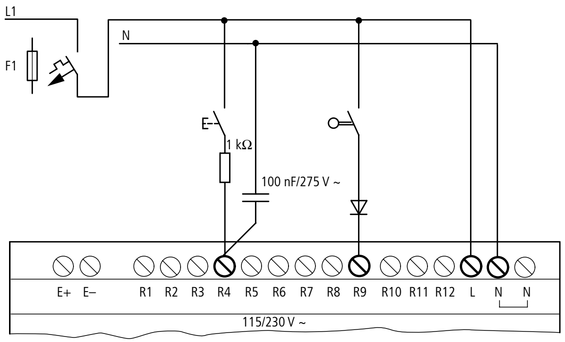


Figure 38: Limitation of the inrush current with a resistor

Complete devices for increasing the input current are available under the type reference EASY256-HCI.

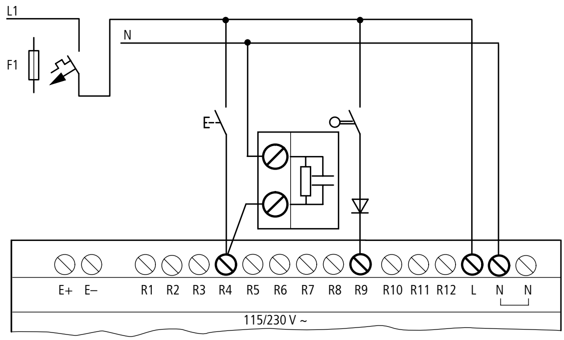


Figure 39: easy600 with EASY256-HCI

→ The increased capacitance increases the drop-out time by approx. 40 ms.

Connecting MFD-DC inputs

Use input terminals I1 to I12 to connect pushbutton actuators, switches or 3 or 4-wire proximity switches. Given the high residual current, do not use 2-wire proximity switches.

Input signal voltage range

- I1 to I6, I9, I10
  - OFF signal: 0 to 5 V
  - ON signal: 15 V to 28.8 V
- I7, I8, I11, I12
  - OFF signal: < 8 V
  - ON signal: > 8 V

Input current

- I1 to I6, I9, I10, R1 to R12: 3.3 mA at 24 V
- I7, I8, I11, I12: 2.2 mA at 24 V

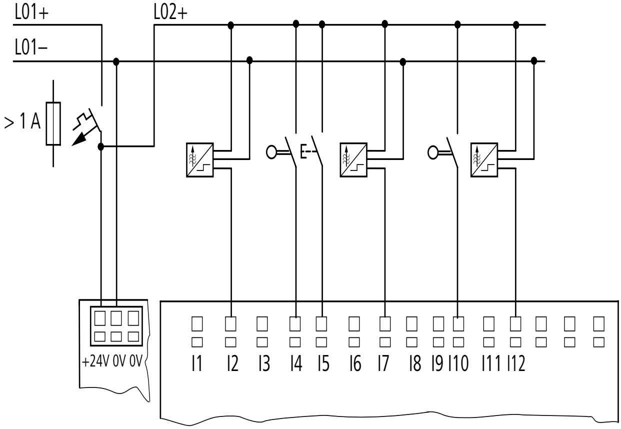


Figure 40: MFD-DC

→ The digital inputs must have the same voltage as the power supply of the MFD.

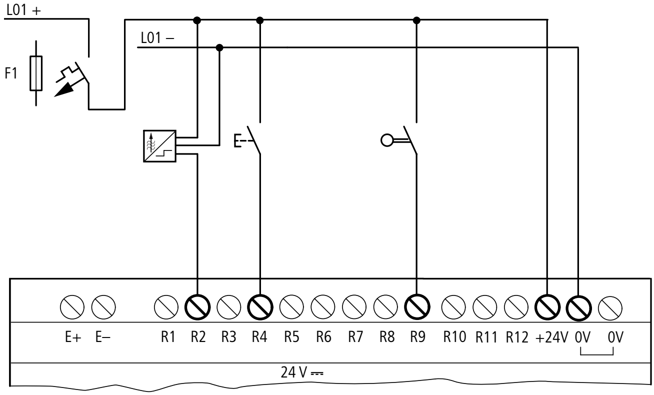


Figure 41: EASY…-DC-.E

Connecting analog inputs

Inputs I7, I8, I11 and I12 can also be used to connect analog voltages ranging from 0 V to 10 V.

The following applies:

- I7 = IA01
- I8 = IA02
- I11 = IA03
- I12 = IA04

The resolution is 10-bit = 0 to 1 023.

Caution!
Analog signals are more sensitive to interference than digital signals. Consequently, more care must be taken when laying and connecting the signal lines. Incorrect switching states may occur if they are not connected correctly.

▶ Use shielded twisted pair cables to prevent interference with the analog signals.

▶ For short cable lengths, ground the shield at both ends using a large contact area. If the cable length exceeds 30 m or so, grounding at both ends can result in equalisation currents between the two grounding points and thus in the interference of analog signals. In this case, only ground the cable at one end.

▶ Do not lay signal lines parallel to power cables.

▶ Connect inductive loads to be switched via the MFD outputs to a separate power feed, or use a suppressor circuit for motors and valves. If loads such as motors, solenoid valves or contactors are operated with MFD via the same power feed, switching may give rise to interference on the analog input signals.

The following circuits contain examples of applications for analog value processing.

→ Ensure that the reference potential is connected. Connect the 0 V of the power supply unit for the different setpoint potentiometers and sensors shown in the examples to the 0 V terminal of the MFD power feed.

Setpoint potentiometer

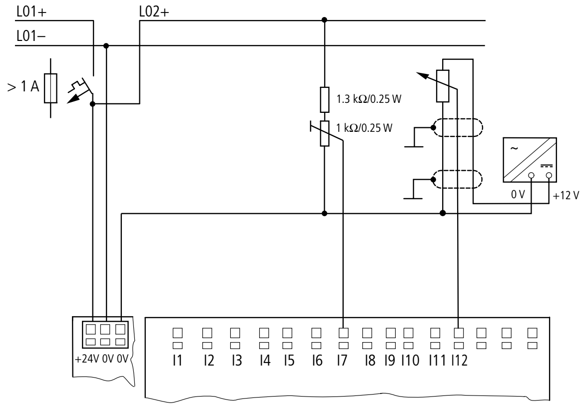


Figure 42: Setpoint potentiometer with upstream resistor

Use a potentiometer with a resistance of ≦ 1 kΩ, e.g. 1 kΩ, 0.25 W.

Temperature sensor, brightness sensor, 20 mA sensor

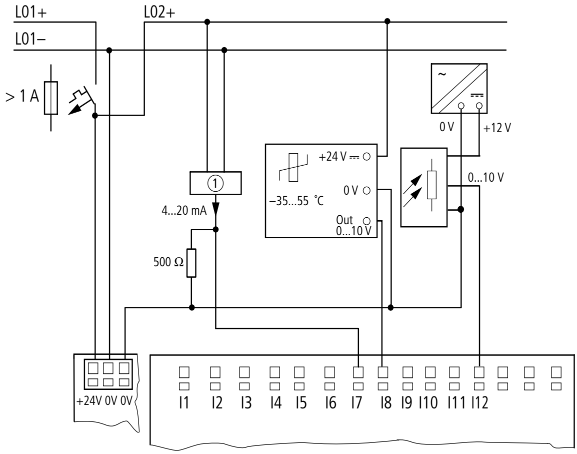


Figure 43: Temperature sensor, brightness sensor, 20 mA sensor

4 to 20 mA (0 to 20 mA) sensors can be connected easily without any problem using an external 500 Ω resistor.

The following values apply:

- 4 mA = 1.9 V
- 10 mA = 4.8 V
- 20 mA = 9.5 V

(according to $U = R \times I$ = 478 Ω × 10 mA ~ 4.8 V)

Connecting high-speed counters and frequency generators

High-speed counter signals on the MFD-Titan can be counted correctly on inputs I1 to I4 independently of the cycle time.

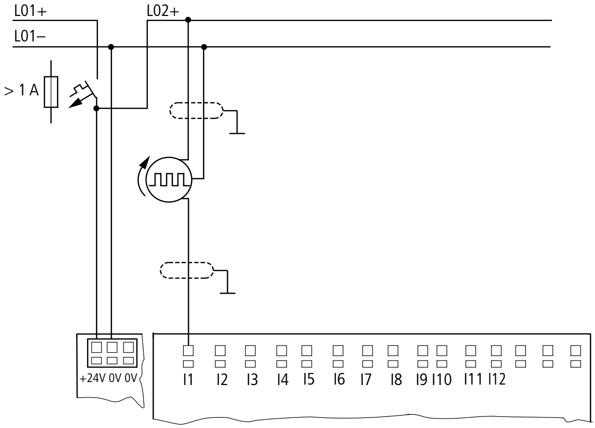


Figure 44: High-speed counter, frequency generator

Connecting incremental encoders

Inputs I1, I2 and I3, I4 on the MFD-Titan can each be used for the high-speed counting of an incremental encoder independently of the cycle time. The incremental encoder must generate two 24 V DC square wave signals with a 90° phase shift between them.

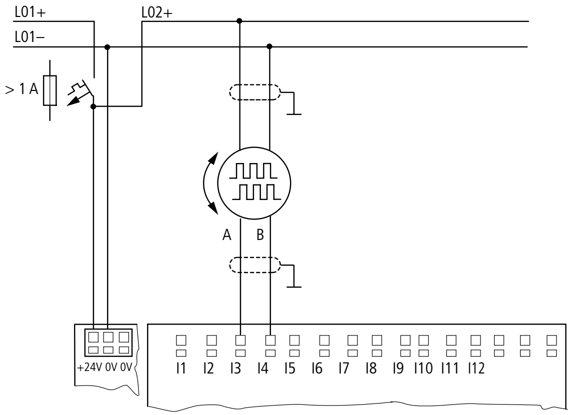


Figure 45: Connecting incremental encoders

Connecting the outputs

The Q... outputs function inside MFD as isolated contacts.

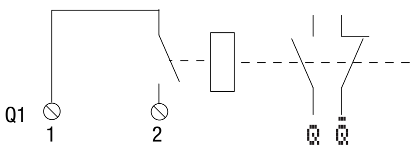


Figure 46: Output “Q”

The respective relay coils are actuated in the MFD circuit diagram via the output relays Q 01 to Q 04 or S 01 to S 06 (S 08). You can use the signal states of the output relays as make or break contacts in the MFD circuit diagram for additional switching conditions.

The relay or transistor outputs are used to switch loads such as fluorescent tubes, filament bulbs, contactors, relays or motors. Check the technical thresholds and output data before installing such devices (see → chapter "Technical data", from Page 409).

Connecting relay outputs MFD-R..

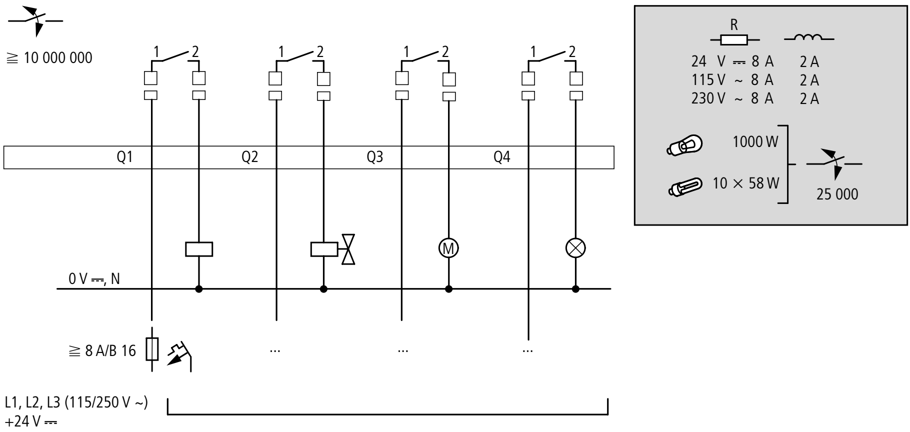


Figure 47: MFD-R.. relay outputs

EASY6..-..RE..

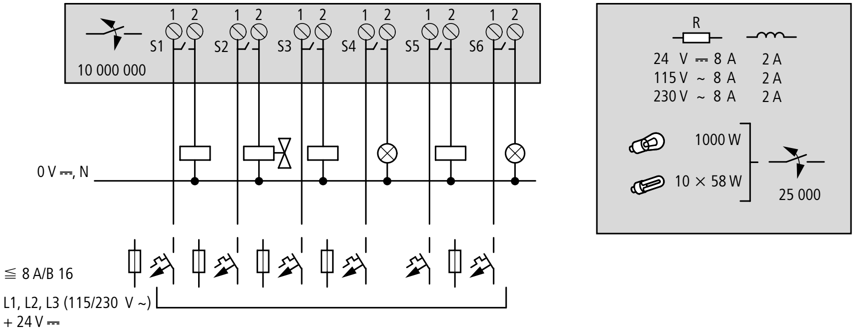


Figure 48: EASY6..-..-RE.. relay outputs

EASY2..-RE

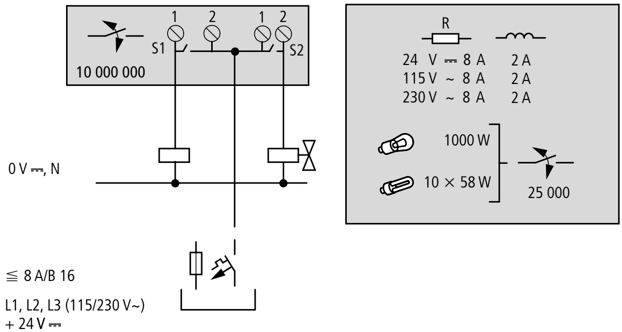


Figure 49: EASY2..-..-RE.. relay outputs

Unlike the inputs, the MFD-R.., EASY6..-..RE relay outputs can be connected to different lines.

Do not exceed the maximum voltage of 250 V AC on a relay contact. If the voltage exceeds this threshold, flashover may occur at the contact, resulting in damage to the device or a connected load.

Connecting transistor outputs

MFD-T..

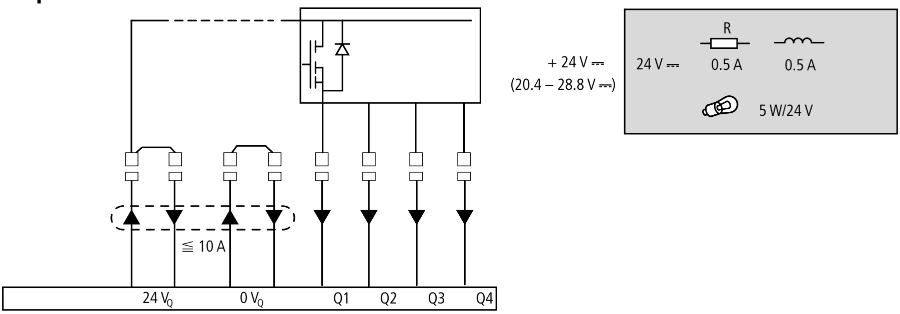


Figure 50: MFD-T.. transistor outputs

EASY6..-DC-TE

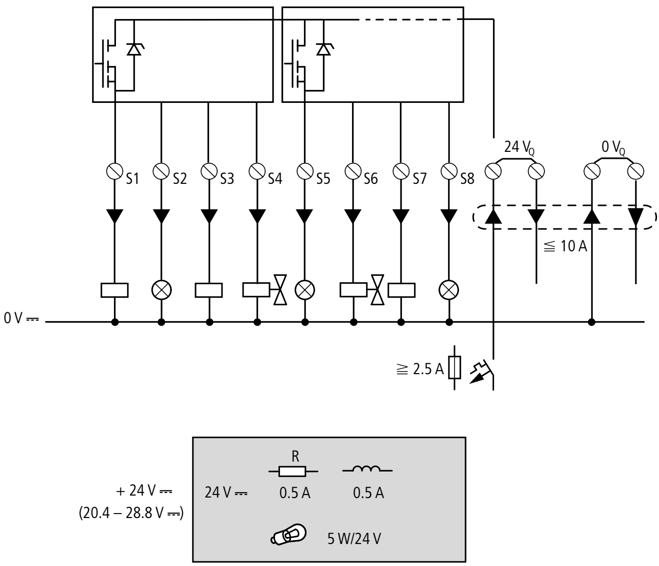


Figure 51: EASY6..-DC-TE transistor outputs

Parallel connection:
Up to four outputs can be connected in parallel in order to increase the power. The output current will increase in this case to a maximum of 2 A.

Caution!
Outputs may only be connected in parallel within a group (Q1 to Q4 or Q5 to Q8, S1 to S4 or S5 to S8), such as Q1 and Q3 or Q5, Q7 and Q8. Outputs connected in parallel must be switched at the same time.

Caution!
Please note the following when switching off inductive loads:
Suppressed inductive loads cause less interference in the entire electrical system. For optimum suppression the suppressor circuits are best connected directly in the proximity of the inductive load.

If inductive loads are not suppressed, the following applies: Several inductive loads should not be switched off simultaneously to avoid overheating the driver blocks in the worst possible case. If in the event of an emergency stop the +24 V DC power supply is to be switched off by means of a contact, and if this would mean switching off more than one controlled output with an inductive load, then you must provide suppressor circuits for these loads (→ following diagrams).

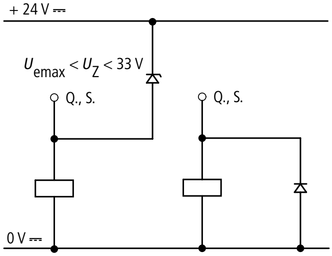


Figure 52: Inductivity with suppressor circuit

Behaviour with short-circuit/overload

Should a short circuit or overload occur on a transistor output, this output will switch off. The output will switch on up to maximum temperature after the cooling time has elapsed. This time depends on the ambient temperature and the current involved. If the fault condition persists, the output will keep switching off and on until the fault is corrected or until the power supply is switched off (→ Section "Monitoring of short-circuit/overload with EASY..-D.-T..", Page 395).

Connecting analog outputs

MFD-RA.. and MFD-TA.. each have an analog output QA 01, 0 V to 10 V DC, 10 bit resolution (0 to 1023). The analog output allows you to control servo valves and other final controlling elements.

⚠

Caution!
Analog signals are more sensitive to interference than digital signals. Consequently, more care must be taken when laying and connecting the signal lines. Incorrect switching states may occur if they are not connected correctly.

Connecting servo valves

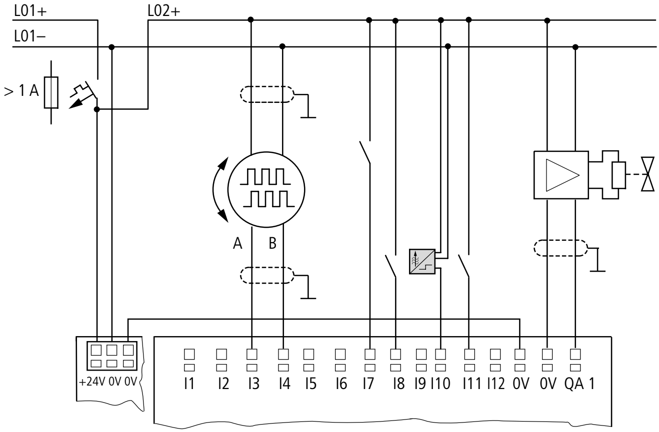


Figure 53: Connecting servo valves

Setpoint entry for a drive

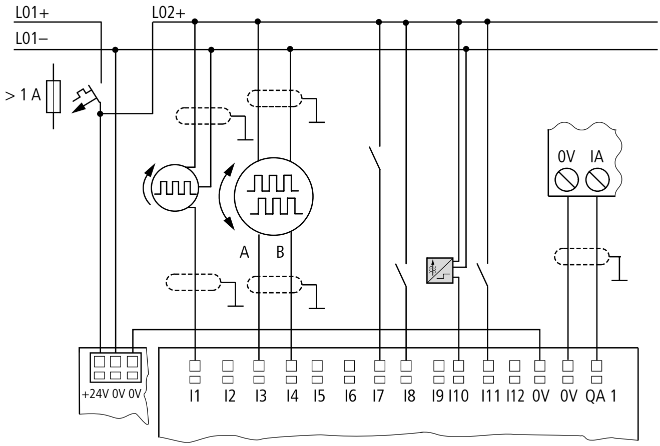


Figure 54: Setpoint entry for a drive

Connecting the NET network

MFD-Titan with network connection (MFD-CP.-NT) can be used for creating the NET network. Up to eight devices can be connected to this network. Further information can be found in the Chapter "easy-NET Network, COM-LINK Serial Connection", Page 335.

Accessories

Connection plug:
8-pole RJ45, EASY-NT-RJ45

Connection assignment of the RJ45 socket on the device

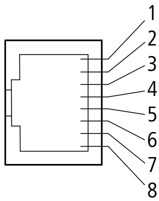


Figure 55: RJ45 socket

Connection cable:
4-wire, twisted pair; → chapter "Technical data", Page 428

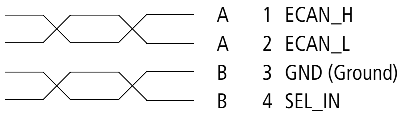


Figure 56: Connection assignment

ECAN_H data cable, pin 1, cable pair A

ECAN_L data cable, pin 2, cable pair A

Ground cable GND, pin 3, cable pair B

Select cable SEL_IN, pin 4, cable pair B

→ Minimum operation with easy-NET functions with the cables ECAN_H, ECAN_L and GND. The SEL_IN cable is only used for automatic addressing.

Table 5: Prefabricated cables, RJ45 plug on both ends

Cable length cm	Type designation
30	EASY-NT-30
80	EASY-NT-80
150	EASY-NT-150

Material for self-manufactured cables
100 m 4 × 0.18 mm^2: EASY-NT-CAB

Required crimping tool for RJ45 plug: EASY-RJ45-TOOL

Bus termination resistor
The first and last stations in the network must be provided with a bus termination resistor.

- Value: 124 Ω
- Termination connector: EASY-NT-R

Cable length and cross-sections

For correct operation of the network the cable lengths, cross-sections and cable resistances must correspond to the following table.

Cable length m	Cable resistance mΩ/m	Cross-section mm^2	 AWG
up to 40	≦ 140	0.13	26
up to 175	≦ 70	0.25 to 0.34	23, 22
up to 250	≦ 60	0.34 to 0.5	22, 21, 20
up to 400	≦ 40	0.5 to 0.6	20, 19
up to 600	≦ 26	0.75 to 0.8	18
up to 1 000	≦ 16	1.5	16

The surge impedance of the cables used must be 120 Ω.

→ With cables >500 m it may be feasible to install a fibre optic run.

Calculating the cable length with known cable resistance

If the resistance of the cable per unit of length is known (resistance per unit length R' in Ω/m), the entire cable resistance R_L must not exceed the following values. R_L depends on the selected baud rates:

Baud rate KBaud	Cable resistance R_L Ω
10 to 125	≦ 30
250	≦ 25
500 1 000	≦ 12

l_{max} = maximum cable length in m

R_L = Total cable resistance in Ω

R' = Cable resistance per unit length in Ω/m

$$l_{max} = \frac{R_L}{R'}$$

Calculating cross-section with known cable lengths

The minimum cross-section is determined for the known maximum extent of the network.

l = cable length in m

S_{min} = minimum cable cross-section in mm^2

ρ_{cu} = resistivity of copper, if not otherwise stated 0.018 $\Omega mm^2/m$

$$S_{min} = \frac{l \times \rho_{cu}}{12.4}$$

→ If the result of the calculation does not yield a standard cross-section, the next larger cross-section is used.

Calculating length with known cable cross-section

The maximum cable lengths are calculated for a known conductor cross-section

l_{max} = cable length in m

S = minimum cable cross-section in mm^2

ρ_{cu} = resistivity of copper, if not otherwise stated 0.018 $\Omega mm^2/m$

$$l_{max} = \frac{S \times 12.4}{\rho_{cu}}$$

Plugging and unplugging network cables

MFD-Titan is provided with two RJ45 network sockets.

The socket 1 in the first station is for the bus terminating resistor. For other stations, socket 1 is used for plugging in the incoming cable. Socket 2 is used for the outgoing cable or for the bus termination resistor on the last physical station in the network.

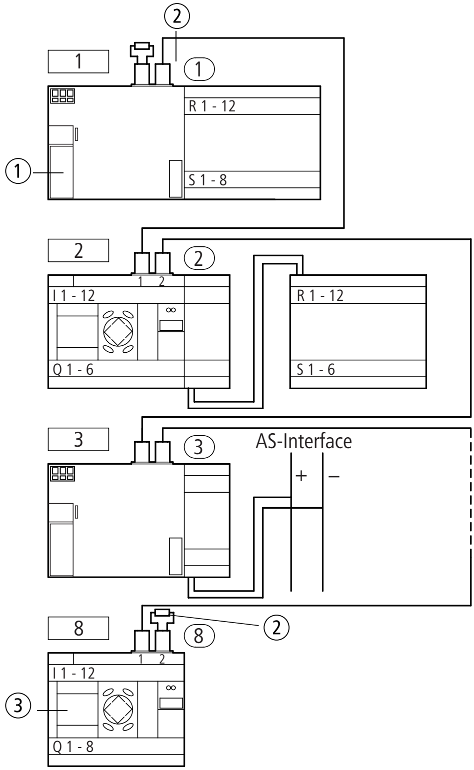


Figure 57: Bus termination resistors

① First station on the NET network

② Bus termination resistor

③ Last station on the NET network

▭ Physical location, place

⬭ Station number

Both RJ45 interfaces are visible after the cover plate has been removed.

When a cable is plugged in, the mechanical connection must be audible (click) and visible [1].

Before a plug or cable is removed, the mechanical locking feature must be undone [2], [3].

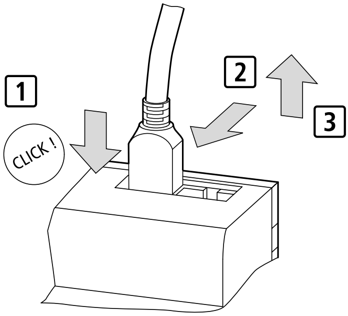


Figure 58: Plugging and unplugging cables

Connecting the serial interface

The MFD power supply/CPU module is provided with a multi-function interface. This can be used to set up point-to-point communication between different devices. The interface is also used for connecting EASY-SOFT-PRO.

The following device configurations are possible:

- MFD with MFD,
- MFD with easy800 (from device version 04).

The serial interface must be implemented using special cables.

The standard MFD-800-CAB cable is 2 m in length.

→ The MFD-800-CAB cable must not be lengthened in order ensure compliance with EMC requirements.

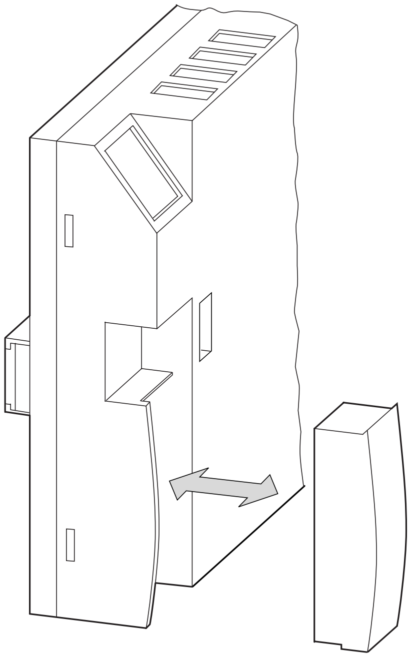


Figure 59: Fitting/removing the interface cover

▶ Remove the interface cover or other plugs from the interface.

▶ Fit the connectors in the devices.

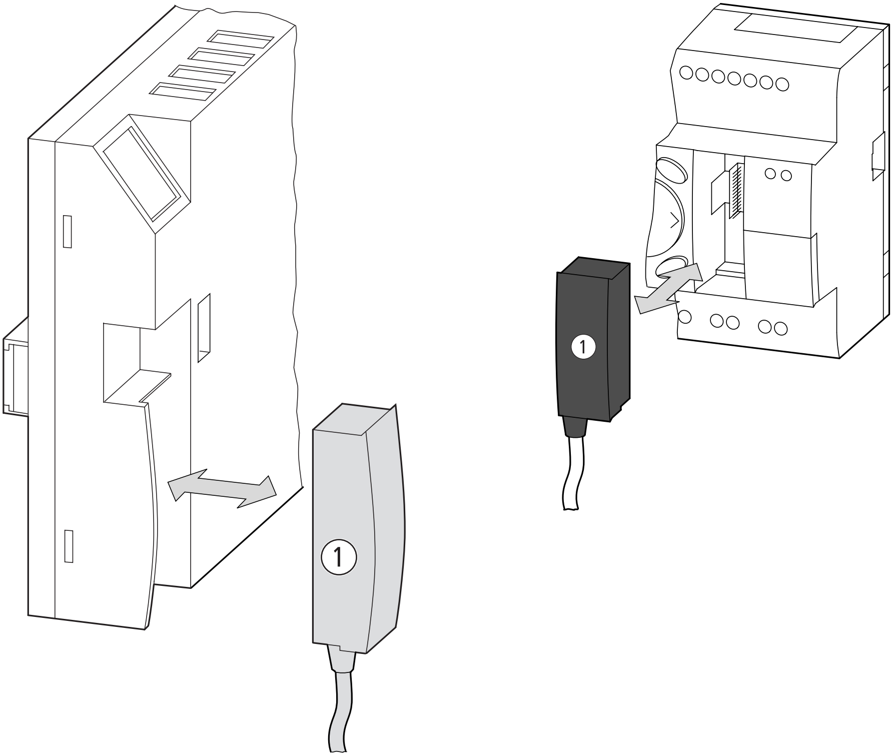


Figure 60: Fitting the connection plug

→ It must be ensured in all circumstances that the connector with the marking POW-Side is fitted in the interface of the MFD device. The serial interface only functions if the MFD device is providing the power feed required for the interface cable.

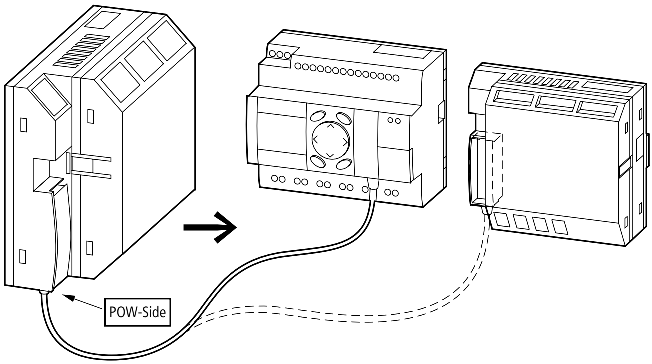


Figure 61: Point-to-point serial interface

Expanding inputs/outputs

You can add expansion units to all MFD types with an easy LINK connection in order to increase the number of inputs and outputs:

Expandable easy basic units	Expansion units	
MFD-CP8-..	EASY618-..-RE	• 12 AC inputs, • 6 relay outputs
	EASY620-..-TE	• 12 DC inputs, • 8 transistor outputs
	EASY202-RE	2 relay outputs, common[1)]
	Special expansion units for connecting to other bus systems are shown in the latest product catalogue.	

1) Common supply for multiple outputs

Local expansion

Local expansion units are fitted directly next to the power supply/CPU module with an easy-LINK connection.

▶ Connect the easy expansion unit via the easy-LINK-DS plug connector.

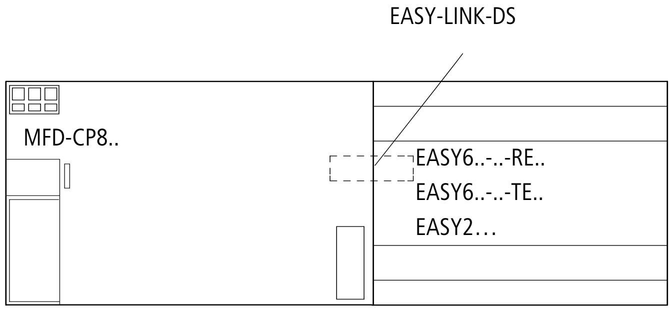


Figure 62: Connecting local expansion units with MFD-CP8..

The following electrical separation is implemented between the power supply/CPU module of the MFD device and the expansion unit (separation always in local connection of expansion unit)

- Simple isolation 400 V AC (+10 %)
- Safe isolation 240 V AC (+10 %)

Units may be destroyed if the value 400 V AC +10 % is exceeded, and may cause the malfunction of the entire system or machine!

→ MFD power supply/CPU modules and expansion units can be fed by different DC power supplies.

Remote expansion

Remote expansion units can be installed and run up to 30 m away from the basic unit.

Warning!
The two-wire or multiple-wire cable between the devices must adhere to the insulation voltage requirement which is stipulated for the installation environment. Otherwise, a fault (ground fault, short-circuit) may lead to the destruction of the units or injury to persons.

A cable such as NYM-0 with a rated operational voltage of U_e = 300/500 V AC is normally sufficient.

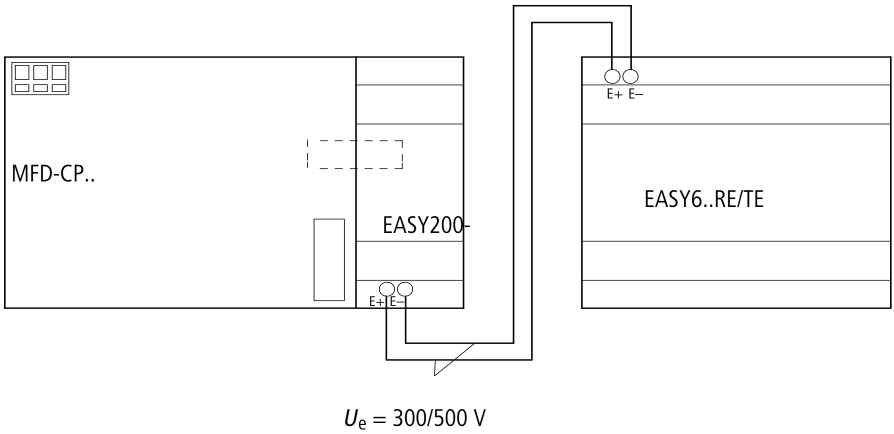


Figure 63: Connecting remote expansion units to the MFD-Titan

→ Terminals E+ and E– of the EASY200-EASY are protected against short-circuits and polarity reversal.
Functionality is only ensured if E+ is connected with E+ and E- with E-.

3 Commissioning

Switching on

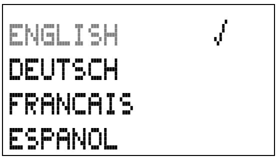
Before startup check whether the power supply, inputs, outputs, the serial interface and the easy-NET connection are properly connected:

- 24 V DC version:
  - Terminal +24 V: +24 V voltage
  - Terminal 0 V: 0 V voltage
  - Terminals I1 to I12, R1 to R12: Actuation via +24 V
- 230 V AC version
  - Terminal L: Phase conductor L
  - Terminal N: Neutral conductor N
  - Terminals R1 to R12: Actuation via phase conductor L

If you have already integrated devices into a system, secure any parts of the system connected to the working area to prevent access and ensure that no-one can be injured if, for example, motors start up unexpectedly.

Setting the menu language

When you switch on MFD for the first time, you will be asked to select the menu language.



▶ Use the cursor buttons ∧ or ∨ to select the language required.
- English
- German
- French
- Spanish
- Italian
- Portuguese
- Dutch
- Swedish
- Polish
- Turkish

▶ Press **OK** to confirm your choice and press **ESC** to exit the menu.

easy will then switch to the Status display.

→ You can change the language setting at a later time, if you wish, see → Section "Changing the menu language", Page 366.

If you do not set the language, MFD will display this menu every time you switch on and wait for you to select a language.

MFD operating modes

MFD operating modes - RUN, STOP and TERMINAL MODE.

In RUN mode the MFD continuously processes a stored program until you select STOP, disconnect the power supply or switch to TERMINAL MODE. The program, parameters and the MFD settings are retained in the event of a power failure. All you will have to do is reset the real-time clock after the back-up time has elapsed. Circuit diagram entry is only possible in STOP mode.

Caution!
In RUN mode, the MFD will immediately run the program saved in the unit when the power supply is switched on. This will not happen if STOP or TERMINAL mode was set as startup mode. In RUN mode outputs are activated according to the switch logic involved.

The following applies to devices without display/operating unit:

- Memory card containing a valid circuit diagram must be fitted.
- Device must be switched on.

If the device has no program, the program stored on the memory card is loaded automatically and the device immediately starts running the program in RUN mode.

Creating your first circuit diagram

The following single line diagram takes you step by step through wiring up your first circuit diagram. In this way you will learn all the rules, quickly enabling you to use MFD for your own projects.

As with conventional wiring, you use contacts and relays in the MFD diagram. With MFD, however, you no longer have to connect up components individually. At the push of a few buttons, the MFD circuit diagram produces all the wiring required. All you have to do is then connect any switches, sensors, lamps or contactors you wish to use.

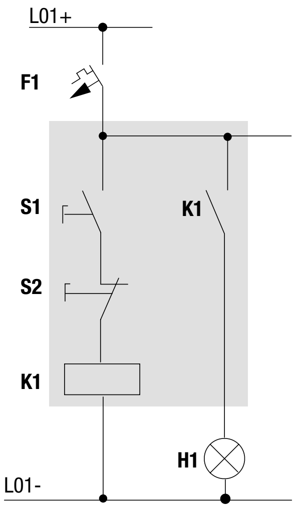


Figure 64: Lamp controller with relays

In the following example, MFD carries out all the wiring and performs the tasks of the circuit diagram shown below.

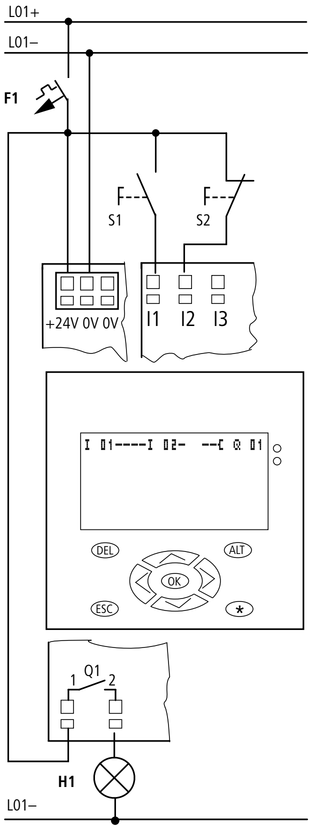


Figure 65: Lamp controller with MFD

Starting point Status display

```
I ............
      I     P-
MO 02:00
Q........  STOP
```

When you switch on MFD, it opens the Status display immediately to show the switching state of the inputs and outputs. It also indicates whether the MFD is already running a program. **Note:** If another display is visible, a visualization screen is shown.

→ The examples were written without the use of expansion units. If an expansion unit is connected, the Status display will first show the status of the basic unit and then the status of the expansion unit before showing the first selection menu.

```
PROGRAM...
STOP √ RUN
PARAMETERS
SET CLOCK...
```

▶ Press **OK** to switch to the main menu.

Press **OK** to switch to the next menu level, and press **ESC** to move one level back.

→ **OK** has two other functions:

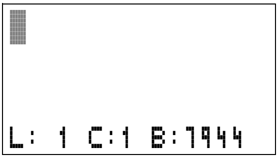
- Press **OK** to save modified settings.
- In the circuit diagram, you can also press **OK** to insert and modify contacts and relay coils.

In this case MFD must be in STOP mode.

```
CIRCUIT DIAGRAM
FUNCTION RELAYS
```

▶ Press **OK** 2 × to enter the circuit diagram display via menu items PROGRAM… → PROGRAM. This is where you will create the circuit diagram.

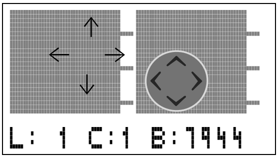
Circuit diagram display



The circuit diagram display is currently empty. The cursor flashes at the top left, which is where you will start to create your diagram.

The location of the cursor is indicated in the status line. L: = Rung (line), C: = Contact or coil (contact), B: = Free memory available in bytes. Start value 7944, with the first three rungs already generated.

The MFD-Titan circuit diagram supports 4 contacts and one coil in series. The MFD-Titan display can display 6 circuit diagram contact fields.



Use the ︿ ﹀ < > cursor buttons to move the cursor over the invisible circuit diagram grid.

The first four columns are contact fields, the fifth column is a coil field. Each line is a rung. MFD automatically connects the contact to the power supply.

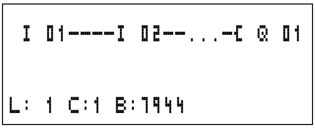


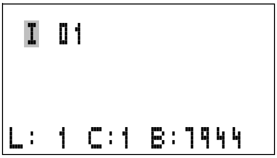
Figure 66: Circuit diagram with inputs I1, I2 and output Q1

▶ Now try to wire up the following MFD diagram.

Switches S1 and S2 are at the input. I 01 and I 02 are the contacts for the input terminals. Relay K1 is represented by the relay coil [Q 01. The symbol identifies the coil's function, in this case a relay coil acting as a contactor. Q 01 is one of the MFD output relays.

From the first contact to the output coil

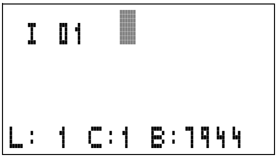
With MFD, you work from the input to the output. The first input contact is I 01.



▶ Press **OK**.

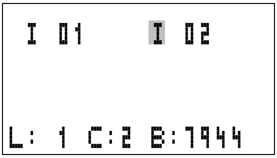
MFD proposes the first contact I 01 at the cursor position.

I flashes and can be changed, for example, to a P for a pushbutton input using the cursor buttons ⌃ or ⌄. However, nothing needs to be changed at this point.



▶ Press **OK** 2 ×, to move the cursor across the 01 to the second contact field.

You could also move the cursor to the next contact field using the cursor button.

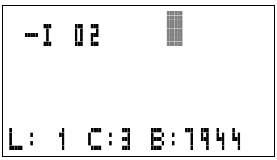


▶ Press **OK**.

Again, MFD inserts a contact I 01 at the cursor position. Change the contact number to I 02, so that break contact S2 can be connected to input terminal I2.

▶ Press **OK** so that the cursor jumps to the next position and press cursor button ⌃ or ⌄ to change the number to 02.

→ You can press **DEL** to delete a contact at the cursor position.



▶ Press **OK** to move the cursor to the third contact field.

You do not need a third relay contact, so you can now wire the contacts directly up to the coil field.

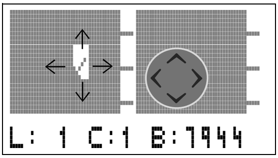
Wiring

MFD displays a small arrow ↙ in the circuit diagram when creating the wiring.

Press **ALT** to activate the wiring arrow cursor and use the cursor buttons ∧ ∨ < > to move it.

→ **ALT** also has two other functions depending on the cursor position:

- In the left contact field, you can press **ALT** to insert a new empty rung.
- The contact under the cursor can be changed between a make and break contact by pressing the **ALT** button.



The wiring arrow works between contacts and relays. When you move the arrow onto a contact or relay coil, it changes back to the cursor and can be reactivated with ALT if required.

→ MFD automatically wires adjacent contacts in a rung up to the coil.

▶ Press **ALT** to wire the cursor from I 02 through to the coil field.

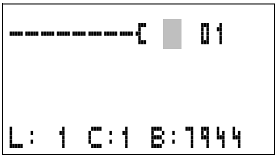
The cursor changes into a flashing wiring arrow and automatically jumps to the next possible wiring position.

▶ Press the cursor button >. Contact I 02 will be connected up to the coil field.

→ You can use **DEL** to erase a connection at the cursor or arrow position. Where connections intersect, the vertical connections are deleted first, then, if you press **DEL** again, the horizontal connections are deleted.

▶ Press the cursor button > again.

The cursor will move to the coil field.



▶ Press **OK**.

MFD inserts the relay coil Q 01. The specified coil function [and the output relay Q 01 are correct and do not have to be changed.

Your first working MFD circuit diagram now looks like this:

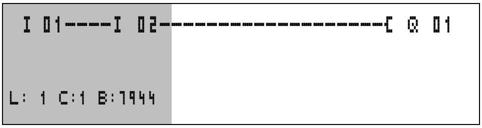


Figure 67: Your first circuit diagram

= visible area

▶ Press **ESC** to leave the circuit diagram display.

The SAVE menu appears.

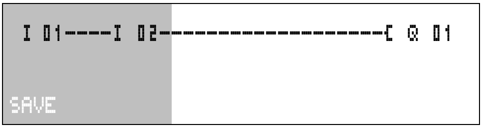


Figure 68: SAVE menu

= visible area

▶ Press the **OK** button.

The circuit diagram is stored.

Once you have connected pushbutton actuators S1 and S2, you can test your circuit diagram straight away.

Testing the circuit diagram

```
PROGRAM...
STOP √ RUN
PARAMETERS
SET CLOCK...
```

▶ Switch to the main menu and select the STOP RUN menu option.

With a tick at RUN or STOP you switch to the RUN or STOP operating modes.

MFD runs in the mode indicated by the tick.

▶ Press the **OK** button. MFD will change to RUN mode.

→ The mode assigned the tick is always active.

The Status display shows the current mode and the switching states of the inputs and outputs.

```
 I 12..........
      I       P-
MO 14:42
Q 1.......   RUN
```

▶ Change to the Status display and press pushbutton actuator S1.

The contacts for inputs I1 and I2 are activated and relay Q1 picks up. This is indicated on the numbers which are displayed.

Power flow display

MFD allows you to check rungs in RUN mode. This means that you can check your circuit diagram via the built-in power flow display while it is being processed by the MFD.

▶ Change to the Circuit diagram display and press pushbutton actuator S1.

The relay picks up. MFD shows the power flow.

```
 I 01====I 02===================[ Q 01


L: 1 C:1 RUN
```

Figure 69: Power flow display: Inputs I1 and I2 are closed, relay Q1 has picked up

= visible area

▶ Press pushbutton actuator S2, that has been connected as a break contact.

The rung is interrupted and relay Q1 drops out.

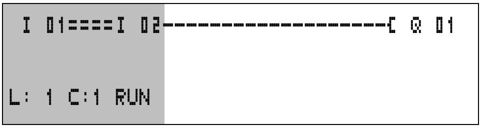


Figure 70: Power flow display: Input I1 is closed, input I2 is open, relay Q1 has dropped out

= visible area

▶ Press **ESC** to return to the Status display.

→ With MFD you can test parts of a circuit diagram before it is entirely completed.

MFD simply ignores any incomplete wiring that is not yet working and only runs the finished wiring.

Power flow display with Zoom function

MFD enables you to check the following at a glance:

- all four contacts plus one coil in series
- and 3 rungs

▶ Change to the Circuit diagram display and press the ALT button. Press pushbutton actuator S1.

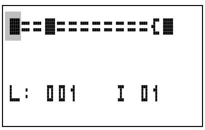


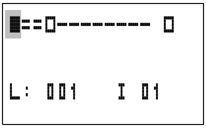
Figure 71: Power flow display in Zoom function: Input I1 and I2 are closed, relay Q1 picked up

▓ Contact closed, coil is triggered

☐ Contact opened, coil dropped out

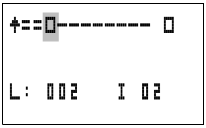
▶ Press pushbutton actuator S2, that has been connected as a break contact.

The rung is interrupted and relay Q1 drops out.



Use the cursor buttons ⋀ ⋁ < > to move between the contacts or coil.

▶ Press the cursor button >.



The cursor moves to the second contact.

▶ Press the ALT button. The display changes to display status with contact and/or coil designation.

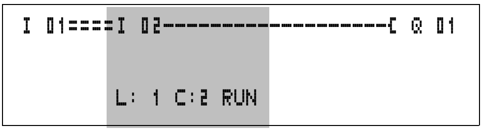


Figure 72: Power flow display: Input I1 is closed, input I2 is open, relay Q1 has dropped out

= visible area

Deleting the circuit diagram

▶ Switch the MFD to STOP mode.

→ MFD must be in STOP mode in order to extend, delete or modify the circuit diagram.

▶ Use PROGRAM… to switch from the main menu to the next menu level.

▶ Select DELETE PROGRAM

```
PROGRAM...
DELETE PROGRAM
```

The MFD will display the prompt DELETE?

▶ Press **OK** to delete the program or **ESC** to cancel.

▶ Press **ESC** to return to the Status display.

Fast circuit diagram entry

You can create a circuit diagram in several ways: The first option is to enter the elements in the circuit and then to wire all the elements together. The other option is to use the enhanced operator guidance of the MFD and create the circuit diagram in one go, from the first contact through to the last coil.

If you use the first option, you will have to select some of the elements in order to create and connect up your circuit diagram.

The second, faster option is what you learned in the example. In this case you create the entire rung from left to right.

Configuring an easy-NET network

If you want to work with the easy-NET network and communicate with several stations, the network must be configured first.

Proceed as follows:

▶ Connect all network stations. easy-NET socket 2↑ to easy-NET socket 1↓.
▶ The first station 1 (socket 1↓) and the last station (socket 2↑) must be provided with a network termination resistor ①.
▶ Connect all stations to the power supply.

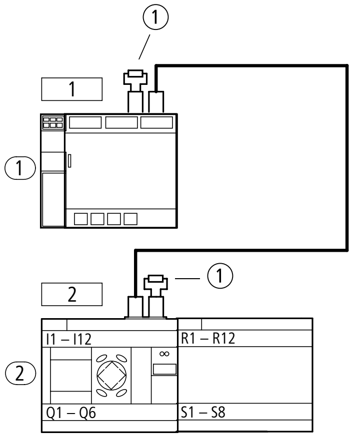


Figure 73: Example topology with two easy-NET stations

① Network termination resistor
▭ Physical location
⬭ Station number

▶ Switch on the power supply on all stations.
▶ Ensure that all stations have a power supply. The POW LED must light up or flash. It is only possible to configure the stations which have an active power supply.
▶ Proceed to the first physical station (Location 1). This station has the termination resistor inserted on socket 1.

The following tasks are only possible in STOP mode.

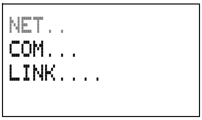
Entering the network station number

▶ Simultaneously press the **DEL** and **ALT** buttons while easy shows the Status display.

The System menu appears

Select the CONFIGURATOR menu option.

▶ Press the **OK** button.

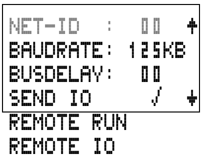


The NET menu appears.

▶ Press the **OK** button.

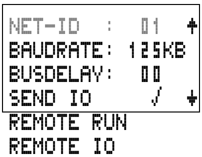
The NET PARAMETERS… menu appears.

▶ Press the **OK** button.



▶ Press the **OK** button and select the station number with ⌃ and ⌄. In this case the station number (NET-ID) “01”.

▶ Press the **OK** button.



▶ Exit the NET PARAMETERS menu with **ESC**.

The station designated station number 1 is the active station. For this reason the REMOTE RUN and REMOTE IO functions are not available.

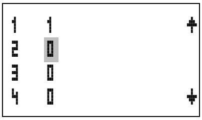
Entering network stations

Only the network station at physical location 1 with station number 1 has a station list.

→ The left-hand column is the physical location. You can only assign a physical location to unused station numbers. Physical location 1 is permanently assigned to station number 1.

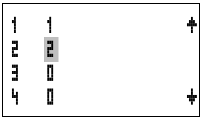
▶ Use the ∧ and ∨ cursor buttons to select the STATION menu and press the **OK** button.

▶ Proceed to the station with physical address 2.



▶ Select the required physical location with the ∧ and ∨ cursor buttons. Press the **OK** button.

▶ Use cursor buttons ∧ and ∨ to select station number 2.



▶ Press the **OK** button.

At physical location 2, the station has been assigned station address 2.

▶ Press **ESC** to return to the STATIONS menu item.

Configuring an easy-NET network

The easy-NET network can only be configured from station 1.

Requirement:
All stations are correctly connected to the network and the termination resistors have been connected.

All stations have a power supply and are in STOP mode. The POW LED is permanently lit. The NET LED is permanently lit.

→ If the connected stations are configured, all stations automatically switch to the STOP mode.

```
NET
PARAMETERS...
STATIONS...
CONFIGURE
```

▶ Proceed to the CONFIGURE menu item and press the **OK** button.

```
CONFIGURE?
```

You will be asked to acknowledge whether you want to configure the system.

▶ Press the **OK** button.

```
CONFIGURATION
IN
PROGRESS.
```

The message on the left appears:

All NET LEDs on the stations which are assigned station numbers higher than 1 (2 to 8) switch to the OFF state of easy-NET.

As soon as the configuration has been successfully completed, the NET LEDs on all stations flash. The easy-NET network is ready for operation.

→ An error message will appear if a station is assigned a station address which does not correspond to the physical location in the station list.

```
ERR: ID-CONFLICT

CONFIGURATION
CONFIGURATION ?
```

If you want to overwrite the station address press the **OK** button. The configuration can be aborted by pressing the **ESC** button.

Changing the easy-NET network configuration

The configuration of the easy-NET network can be modified at any time at station 1, physical location 1.

▶ The NET parameters are modified as described for inputting parameters for the first time.

Station addresses in the STATIONS menu are changed as follows:

▶ Go to the physical location which is to be modified.

▶ Press the **OK** button.

→ Existing station numbers can only be modified to free, non-assigned station numbers. If all eight numbers are assigned, all station numbers which are to be modified must be set to zero. Thereafter, all station numbers can be reassigned. (MFD-Titan sets all station numbers to zero which are assigned a physical location behind the leading zero.)

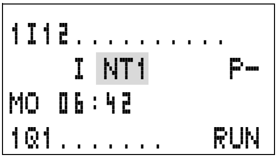
▶ Select the required station number with the ⌃ and ⌄ cursor buttons and confirm your input with the **OK** button.

▶ Configure all easy-NET stations again using the CONFIGURATION menu.

→ Further information concerning the easy-NET network topic can be found in Chapter “easy-NET Network, COM-LINK Serial Connection”, Page 335.

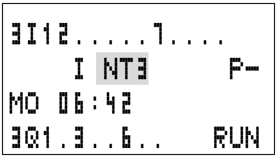
Displaying the Status display of other stations

On every device with a display, you can display the states of the inputs and outputs of each network station.



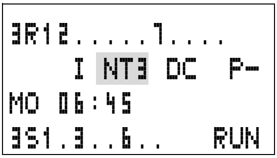
▶ Change to the Status display and press the **ESC** button.

The cursor changes to the display of the network station NT.. and flashes. The station number is displayed in front of the inputs and outputs.



▶ Change to the number of the required station with the ︿ and ﹀ cursor buttons.

▶ Press the **OK** button.



▶ If you want to view the state of the inputs and outputs of a local expansion, press the **OK** button.

If you press the **ESC** button again or the **OK** button, the display of the input and output states of the station is terminated.

→ The station showing the status on its display cannot read its own data from the network.

Example: NT3 flashes on station 3. The inputs and outputs 3I.., 3R.., 3Q.. and 3S.. cannot be displayed.

If the NT3 display is not flashing, the inputs and outputs are shown.

Configuring the interface for the COM-LINK mode

If you wish to set up point-to-point communication with another station, this can be done using either the serial interface or easy-NET. The MFD must be provided with a display and operating unit. The connection must be configured for this purpose (Page 350→ Section "Introduction to COM-LINK").

→ Ensure that the other station supports the COM-LINK mode.

Proceed as follows:

▶ Connect both stations together.

Only use original connection cables. The connector marked POW-Side must be plugged into an MFD. The MFD feeds the interface electronics of the connection line at both ends.

▶ Connect both stations to the power supply.

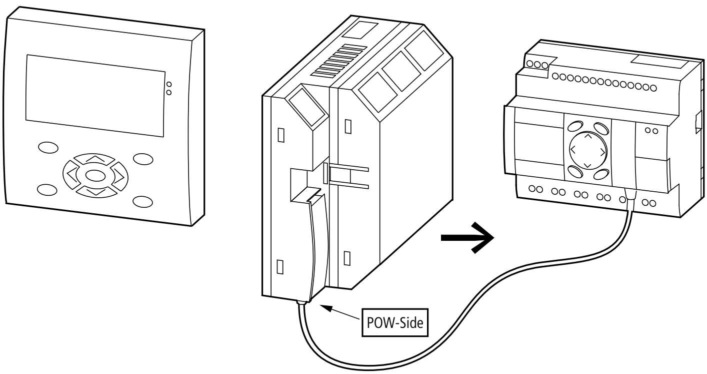


Figure 74: Example with both COM stations.

The MFD is the active station and the second station is the remote station.

▶ Switch on the power supply at both stations.

▶ Ensure that the power supply for both stations is switched on. The POW LED must light up or flash. It is only possible to configure the stations which have an active power supply.

▶ Go to the MFD device that is the active station running the serial interface.

→ The following tasks are only possible in STOP mode.

Setting up the COM-LINK

Caution!
The MFD device can either run as a station on the easy-NET or as a station in a COM-LINK connection. Do not switch an MFD device running as an easy-NET station to COM-LINK. If this is done on easy-NET in RUN mode, the entire easy-NET network will be deactivated. There will be no further data transfer.

Solution:

▶ Deactivate the COM-LINK.

▶ Re-enter the easy-NET address.

▶ Switch the power supply off and then on again.

▶ Reconfigure the easy-NET on station 1.

▶ Simultaneously press the **DEL** and **ALT** buttons while easy shows the Status display.

```
SECURITY...
SYSTEM...
MENU LANGUAGE
CONFIGURATOR...
```

The System menu appears

Select the CONFIGURATOR menu option.

▶ Press the **OK** button.

Select the COM... menu option.

```
NET..
COM...
LINK....
```

▶ Press the ∨ button.

▶ Press the **OK** button.

```
BAUDRATE: 9600B
COM-LINK
REMOTE MARKER...
```

The menu BAUDRATE: 9600B will appear. The two baud rates are for 9600 or 19200 baud. Select the baud rate that your connection will support. Baud rate selection.

→ Select 19200 baud as the baud rate. Badly laid cables may give rise to electromagnetic interference. Select 9600 baud as the baud rate. If this is not satisfactory, the connection cable must be laid in a different location.

```
BAUDRATE:19200B
COM-LINK
REMOTE MARKER...
```

▶ Press the **OK** button.

Select 19200 baud as the baud rate.

▶ Press the ︿ or ﹀ button.

▶ Press the **OK** button.

Switch on the COM connection.

→ The COM-LINK must only be switched at the active station. Two devices with COM-LINK ✓ switched on cannot communicate with each other.

```
BAUDRATE:19200B
COM-LINK      ✓
REMOTE MARKER...
```

▶ Press the ﹀ button.

▶ Press the **OK** button.

The tick on the COM-LINK menu item indicates that COM-LINK has been selected.

No tick means that COM-LINK has not been selected.

→ The following applies to the active station:

If you wish to run data transfers between the two devices in both directions, you must select the marker range on the active station.

```
BAUDRATE:19200B
COM-LINK      ✓
REMOTE MARKER...
```

▶ Select the REMOTE MARKER menu item.

The following selection will only be displayed if the COM-LINK menu item has been ticked.

```
READ:
 1MD00 + 1MD00
WRITE:
 1MD00 + 1MD00
```

▶ Press the **OK** button.

The data is physically located in the second station, i.e. the remote station.

→ The active station reads and writes data from and to the markers of the remote station. At the same time, the remote station has read and write access to the same marker range.

Ensure that both stations do not have write access to the same markers simultaneously. The last write operation will be the one that is retained.

Example:

READ 1MD2 → 1MD2

WRITE 1MD3 → 1MD3

The MFD device accesses the markers with station address 1xx... These markers correspond to the local markers MD2 and MD3 in the remote station.

These marker double words contain:

MD2, MW3, MW4, MB5; MB6, MB7, MB8, M33 to M64

MD3, MW5, MW6, MB9, MB19, MB11, MB12; M65 to M96

The following marker ranges can be selected:

1MD1 to 1MD20

This corresponds to the following range in the remote station:

MD1 to MD20

```
READ:
 1MD00 + 1MD00
WRITE:
 1MD00 + 1MD00
```

▶ Press the **OK** button.

▶ Use the ∧ button to select the start of the READ marker range.

```
READ:
 1MD11 → 1MD14
WRITE:
 1MD00 → 1MD00
```

▶ Use the > button to enter the upper limit of the READ range.

▶ Use the ^ button to select the value.

▶ Confirm the entry with the **OK** button.

```
READ:
 1MD11 → 1MD14
WRITE:
 1MD00 → 1MD00
```

```
READ:
 1MD11 → 1MD14
WRITE:
 1MD00 → 1MD00
```

▶ Use the ⌄ button to enter the WRITE range.

Enter the WRITE range.

```
READ:
 1MD11 → 1MD14
WRITE:
 1MD15 → 1MD17
```

▶ Press **ESC** to leave the entry menu.

```
BAUDRATE:19200B
COM-LINK      √
REMOTE MARKER...
```

The COM-LINK has now been set. No COM settings are required at the remote station.

▶ Press **ESC** to return to the Status display.

```
I 12.4.67.....
     I COM    P-
FR 02:02      ST
Q........   STOP
```

The entry in the second COM line indicates that the COM connection is active.

Terminal mode

Terminal mode

The MFD device also supports the TERMINAL MODE operating mode. This allows you to remotely control other devices. This is especially useful if the other device is located in an inaccessible place. Terminal mode can also be used to show the menus and displays of devices that do not have their own display or operating unit. Terminal mode can be used both with the serial interface and in the easy-NET. The serial interface enables you to access a remote device. If you use the easy-NET network, all other network stations can be addressed.

→ Terminal mode is a separate operating mode like RUN mode. It only functions when a program is not running. For this mode to be active, the MFD must be in STOP mode.

→ All connected devices must also support Terminal mode.

The following topologies are permissible.

Terminal mode using the point-to-point serial interface

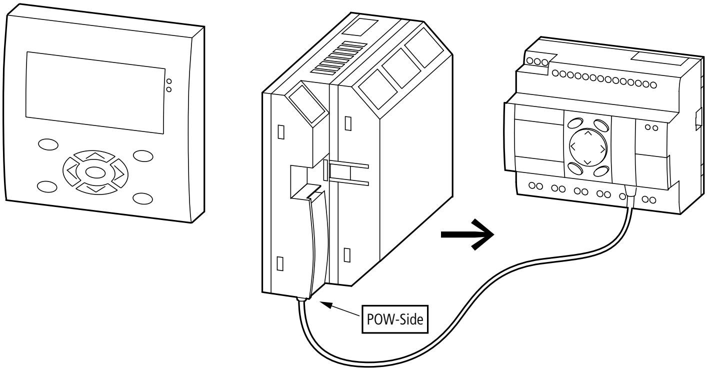


Figure 75: Terminal mode using the point-to-point serial interface

Terminal mode using the easy-NET topology

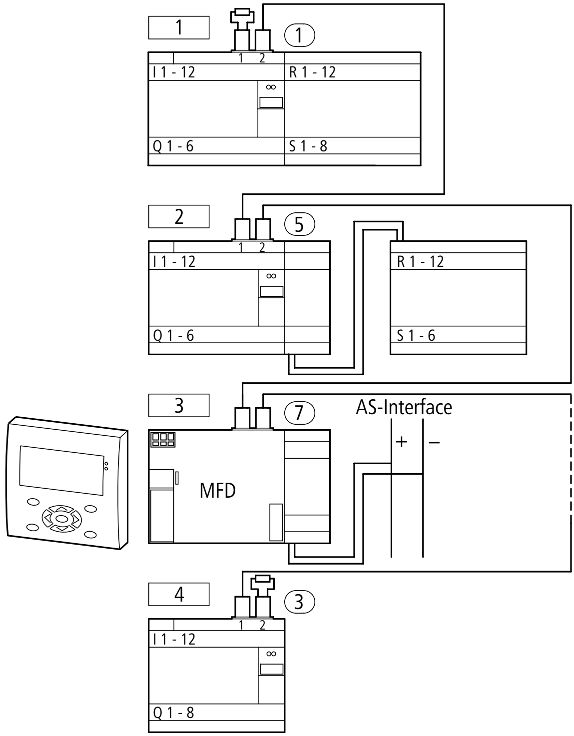


Figure 76: Terminal mode using the easy-NET topology

In the above topology, the physical location is not identical to the station number. The MFD device was connected in the middle of the network line. Terminal mode functions irrespective of the device location and station number.

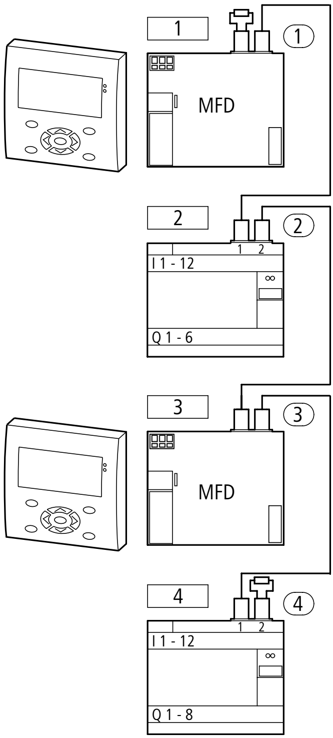


Figure 77: Terminal mode in easy-NET with two MFD devices

The above topology allows two MFD devices to be run in easy-NET Terminal mode. Each MFD device can run with the other devices in Terminal mode.

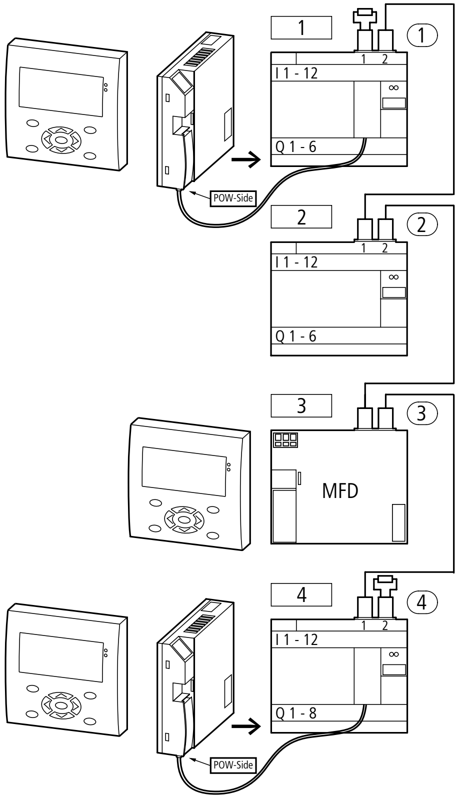


Figure 78: Terminal mode in easy-NET as well as via two serial interfaces

The above topology is a combination of easy-NET operation and serial interface operation. Bear in mind the access rights of the individual devices in easy-NET and in the corresponding serial interface.

Caution!

Data collision!

In order to ensure proper operation, the following conditions must be observed.

The following applies:

If there is more than one MFD device in Terminal mode, each MFD device must access a different easy-NET station.

A device running in Terminal mode must not access any two devices communicating with each other in Terminal mode.

If a PC with EASY-SOFT (-PRO) or an MFD device with a serial interface is in active communication with an easy-NET station, this station must not be accessed at the same time in Terminal mode via the easy-NET.

Proceed as follows:

Your easy-NET or your serial interface must be running correctly.

▶ Press the **OK** button from the Status Display.

```
PROGRAM...     ↕
STOP √ RUN
PARAMETERS
SET CLOCK...
```

The first menu will appear.

▶ Press the ⌃ button.

```
STOP √ RUN      ↕
PARAMETERS
SET CLOCK...
TERMINAL MODE..↕
```

This will display the TERMINAL MODE menu item.

▶ Press the **OK** button.

```
STATION ID: 0
START MODE
```

The START MODE menu item will flash.

▶ Press the ⌃ button.

```
STATION ID: 0
START MODE
```

Select the second station. This station will control the display and respond to the operating unit.

→ Station ID:

0 = Station on the serial interface
1 = Station 1 easy-NET
2 = Station 2 easy-NET
3 = Station 3 easy-NET
4 = Station 4 easy-NET
5 = Station 5 easy-NET
6 = Station 6 easy-NET
7 = Station 7 easy-NET
8 = Station 8 easy-NET

```
STATION ID: 0
START MODE
```

▶ Press the **OK** button.

Select your station.

▶ Press the ︿ or ﹀ button.

▶ Press the **OK** button.

```
STATION ID: 0
START MODE
```

Select the START MODE menu option.

▶ Press the ﹀ button.

▶ Press the **OK** button.

```
STATION ID: 2
START MODE
```

In this case the easy-NET station 2 is connected.

```
  Connection
establishment
in progress...
```

The MFD tries to establish connection to the selected device. The text flashes.

Once the connection is established, the menu appears or the Status display in which the selected device is active.

If the text “Connection establishment in progress...” is displayed for longer than 10 s, the connection to the selected device is faulty. Press **ESC** to cancel the selection. Rectify the fault. Try to re-establish the connection.

The following applies if the device to be operated is in RUN mode and is displaying a screen:

This screen is not displayed in Terminal mode.

```
  The remote
device is in
Graphic mode
```

MFD message: “The remote device is in Graphic mode.”

▶ Press the ALT and **ESC** button simultaneously.
This will call up the Status display.

```
I 1..4..789... *
     I NT2    P-
TU 06.05.2003
Q 1..45678  RUN
```

The easy-NET station 2 controls the display of the MFD device.

The activation of Terminal mode is indicated by the flashing star at the top right of the Status display.

Caution!
In Terminal mode, you can operate a device that may be positioned far from your actual location. All access rights that you would also require "locally" are granted to you. It is not always possible to obtain a view of the situation “locally”. Use of this operating mode and the execution of any changes to device settings should only be carried out with the utmost caution.

A device with a display and operating unit can also be operated locally. In this case, operation at the device concerned is always faster than operation via Terminal mode. Bear in mind that this may lead to conflicts that may trigger faults or unforeseen events.

→ In Terminal mode, the MFD device makes its display and operating unit available to the connected device. Only data for the display and the status of the buttons is sent via the connection. This ensures that the local data of the connected device is not destroyed in the event of a communication fault.

Close Terminal mode.

```
STATION ID: 2
START MODE
```

Press the * button to close the Terminal mode.

→ The * button cannot be assigned to other tasks if you wish to use Terminal mode in your application. Use the * button to change from the visualization to the Status display. Otherwise the Terminal mode menu cannot be reached.

▶ Press the * button.

This returns you back to your local device.

```
STATION ID: 2
START MODE
```

▶ Press the **ESC** button twice.

```
I ..345..89...
      I      P-
SA 06:47     ST
Q 2 4      STOP
```

The Status display of the MFD device is active.

The flashing star at the top right of the display is no longer present.

The display must be initialised again if the CPU was fitted to it under live conditions.

▶ Press the **DEL** and **ESC** button simultaneously.
This will reinitialise the display.

4 Wiring with MFD-Titan

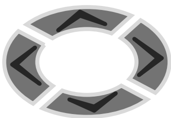
This chapter describes all the functions available with MFD-Titan.

MFD-Titan operation

Buttons for drawing circuit diagrams and function block usage

Delete rung, contact, relay or empty line in the circuit diagram

Toggle between break and make contact
Connect contacts and relays
Add rungs



^ ˅ Change values
Cursor up, down
< > Change position
Cursor left, right

Cursor buttons set as P buttons:

<	Input P1,	^	Input P2
>	Input P3,	˅	Input P4

Undo setting from previous **OK**
Exit current display or menu

Change, add contact/relay
Save setting

Terminal mode on/off

Operating principles

The cursor buttons in the MFD-Titan circuit diagram perform three functions. The current mode is indicated by the appearance of the flashing cursor.

- Move
- Entering
- Connect

In Move mode you can use ∧∨ < > to move the cursor around the circuit diagram in order to select a rung, contact or relay coil.

Use **OK** to switch to Entry mode so that you can enter or change a value at the current cursor position. If you press **ESC** in Entry mode, MFD-Titan will undo the most recent changes.

Press **ALT** to switch to Connect mode for wiring contacts and relays. Press **ALT** again to return to Move.

Press **ESC** to leave the circuit diagram and parameter display.

→ MFD-Titan performs many of these cursor movements automatically. For example, MFD-Titan switches the cursor to Move mode if no further entries or connections are possible at the selected cursor position.

Opening the parameter display for function blocks with contacts or coils

If you specify the contact or coil of a function relay type in Entry mode, MFD-Titan automatically switches from the contact number to the function block parameter display when you press **OK**.

Press > to switch to the next contact or coil field without entering any parameters.

Program
A program is a sequence of commands which the MFD-Titan executes cyclically in RUN mode. An MFD-Titan program consists of the necessary settings for the device, easy-NET, COM-LINK, password, system settings, a circuit diagram and/or function blocks and/or the visualization screens.

The circuit diagram is that part of the program where the contacts are connected together. In RUN mode a coil is switched on and off in accordance with the current flow and the coil function specified.

Function blocks
Function blocks are program elements with special functions. Example: timing relays, time switches, arithmetic function blocks. Function blocks are elements provided with or without contacts and coils as required. In RUN mode the function blocks are processed according to the circuit diagram and the results are updated accordingly.

Examples:
Timing relay = function block with contacts and coils
Time switch = function block with contacts

Visualization screens
Visualisation screens are the sections of programs containing the display and operator functions.

Relays
Relays are switching devices which are electronically simulated in MFD-Titan. They actuate their contacts according to their designated function. A relay consists of a coil and at least one contact.

Contacts
You modify the current flow with the contacts in the MFD-Titan circuit diagram. Contacts such as make contacts carry a 1 signal when closed and 0 when open. Every contact in the MFD-Titan circuit diagram can be defined as either a make contact or a break contact.

Coils

Coils are the actuating mechanisms of relays. In RUN mode, the results of the wiring are sent to the coils, which switch on or off accordingly. Coils can have seven different coil functions.

Table 6: Usable contacts

Contact	MFD-Titan display
Make contact, open in release position	I, Q, M, A, …. Other contacts → Table
Break contact, closed in release position	$\overline{I}$, $\overline{Q}$, $\overline{M}$, $\overline{A}$, … Other contacts → Table

MFD-Titan works with different contacts, which can be used in any order in the contact fields of the circuit diagram.

Table 7: Contacts

Contact	Make contact	Break contact	Number	Page
Inputs				
Inputs of a network station * = Station address 1 to 8	*I	*$\overline{I}$	01...12	338
COM slave inputs	1I	1$\overline{I}$	01..12	–
MFD-Titan input terminal	I	$\overline{I}$	01...12	–
Cursor button	P	$\overline{P}$	01...04	–
Network station expansion input terminal * = Station address 1 to 8	*R	*$\overline{R}$	01...12	338
Input terminal for expansion unit COM slave	1R	1$\overline{R}$	01....12	–
Input terminal for expansion unit	R	$\overline{R}$	01...12	–
Bit inputs via the network * = Station address 1 to 8	*RN	*$\overline{RN}$	01...32	338

Contact	Make contact	Break contact	Number	Page
Diagnostics inputs				
Expansion network station status * = Station address 1 to 8	*I	$*\overline{I}$	14	397
Network station short-circuit/overload * = Station address 1 to 8	*I	$*\overline{I}$	15...16	395
COM slave expansion unit status	1I	$1\overline{I}$	14	–
COM slave short-circuit/overload	1I	$1\overline{I}$	15...16	–
Expansion status	I	$\overline{I}$	14	397
Short-circuit/overload	I	$\overline{I}$	15...16	395
Short-circuit/overload in expansion network station * = Station address 1 to 8	*R	$*\overline{R}$	15...16	395
Short-circuit/overload in COM slave expansion	1R	$1\overline{R}$	15...16	–
Short circuit/overload with expansion	R	$\overline{R}$	15...16	395
Outputs				
Deactivate backlight of the MFD display	LE	$\overline{LE}$	01	–
Red LED of MFD display	LE	$\overline{LE}$	02	–
Green LED of MFD display	LE	$\overline{LE}$	03	–
MFD-Titan output MFD network station * = Station address 1 to 8	*Q	$*\overline{Q}$	01...08	338
COM slave output	1Q	$1\overline{Q}$	01...08	–
MFD-Titan output	Q	$\overline{Q}$	01...08	–
MFD-Titan output expansion with network station * = Station address 1 to 8	*S	$*\overline{S}$	01...08	338
Output of COM slave expansion	1S	$1\overline{S}$	01...08	–
MFD-Titan output expansion	S	$\overline{S}$	01...08	–
Bit outputs via the network * = Station address 1 to 8	*SN	$*\overline{SN}$	01...32	338

Contact	Make contact	Break contact	Number	Page
Other contacts				
Markers	M	$\overline{\text{M}}$	01...96	131
COM slave marker (REMOTE MARKER)	1M	1$\overline{\text{M}}$	01..96	353
Jump label	:		01...32	239
Diagnostics messages	ID	ID	01...16	347
COM slave diagnostics messages	1ID	1$\overline{\text{I}}$D	01...16	353
Function blocks				
Analog value comparator function block	A X Q1	A X Q1	X=01...32	160
Function blocks Analog value comparator Value overflow (CARRY)	A X CY	$\overline{\text{A}}$ X CY	X=01...32	160
Arithmetic value overflow (CARRY) function block	AR X CY	$\overline{\text{AR}}$ X CY	X=01...32	163
Zero arithmetic value (zero) function block	AR X ZE	$\overline{\text{AR}}$ X ZE	X=01...32	163
Data block comparator function block, error: number of elements exceeded	BC X E1	$\overline{\text{BC}}$ X E1	X=01...32	167
Data block comparator function block, error: range overlap	BC X E2	$\overline{\text{BC}}$ X E2	X=01...32	167
Data block comparator function block, error: invalid offset	BC X E3	$\overline{\text{BC}}$ X E3	X=01...32	167
Data block comparator function block, comparison result	BC X EQ	$\overline{\text{BC}}$ X EQ	X=01...32	174
Data block comparator function block, error: number of elements exceeded	BT X E1	$\overline{\text{BT}}$ X E1	X=01...32	174

Contact	Make contact	Break contact	Number	Page
Data block transfer function block, error: range overlap	BT X E2	$\overline{BT}$ X E2	X=01...32	174
Data block transfer function block, error: invalid offset	BT X E3	$\overline{BT}$ X E3	X=01...32	174
Boolean operation function block, value zero	BV X ZE	$\overline{BV}$ X ZE	X=01...32	185
Counter function block, upper setpoint value exceeded (Overflow)	C X OF	$\overline{C}$ X OF	X=01...32	188
Counter function block, lower setpoint value undershot (Fall below)	C X FB	$\overline{C}$ X FB	X=01...32	188
Counter function block, actual value equal to zero	C X ZE	$\overline{C}$ X ZE	X=01...32	188
Counter function block, actual value has exceeded counter range (CARRY)	C X CY	$\overline{C}$ X CY	X=01...32	188
Frequency counter function block, upper setpoint value exceeded (Overflow)	CF X OF	$\overline{CF}$ X OF	X=01...04	195
Frequency counter function block, lower setpoint value undershot (Fall below)	CF X FB	$\overline{CF}$ X FB	X=01...04	195
Frequency counter function block, actual value equal to zero	CF X ZE	$\overline{CF}$ X ZE	X=01...04	195
High-speed counter function block, upper setpoint value exceeded (Overflow)	CH X OF	$\overline{CH}$ X OF	X=01...04	199
High-speed counter function block, lower setpoint value undershot (Fall below)	CH X FB	$\overline{CH}$ X FB	X=01...04	199
High-speed counter function block, actual value equal to zero	CH X ZE	$\overline{CH}$ X ZE	X=01...04	199
High-speed counter function block, actual value has exceeded counter range (CARRY)	CH X CY	$\overline{CH}$ X CY	X=01...04	199

Contact	Make contact	Break contact	Number	Page
Incremental encoder counter function block, upper setpoint value exceeded (Overflow)	CI X OF	$\overline{\text{CI}}$ X OF	X=01...02	205
Incremental encoder counter function block, lower setpoint value undershot (Fall below)	CI X FB	$\overline{\text{CI}}$ X FB	X=01...02	205
Incremental encoder counter function block, actual value equal to zero	CI X ZE	$\overline{\text{CI}}$ X ZE	X=01...02	205
Incremental encoder counter function block, actual value has exceeded counter range (CARRY)	CI X CY	$\overline{\text{CI}}$ X CY	X=01...02	205
Comparator function block, less than	CP X LT	$\overline{\text{CP}}$ X LT	X=01...32	210
Comparator function block, equal to	CP X EQ	$\overline{\text{CP}}$ X EQ	X=01...32	210
Comparator function block, greater than	CP X GT	$\overline{\text{CP}}$ X GT	X=01...32	210
Text output function block	D X Q1	$\overline{\text{D}}$ X Q1	X=01...32	212
Data function block	DB X Q1	$\overline{\text{DB}}$ X Q1	X=01...32	213
PID controller, value range of manipulated variable exceeded	DC X LI	$\overline{\text{DC}}$ X LI	X=01...32	215
Receive a variable from a station (Get)	GT X Q1	$\overline{\text{GT}}$ X Q1	X=01...32	215
Seven-day time switch	HW X Q1	$\overline{\text{HW}}$ X Q1	X=01...32	226
Year time switch function block	HY X Q1	$\overline{\text{HY}}$ X Q1	X=01...32	231
Master reset, sets all outputs and markers to zero state	MR X Q1	$\overline{\text{MR}}$ X Q1	X=01...32	242
Operating hours counter function block, set time reached	OT X Q1	$\overline{\text{OT}}$ X Q1	X=01...04	249

Contact	Make contact	Break contact	Number	Page
Operating hours counter, value overflow (CARRY)	OT X CY	$\overline{\text{OT}}$ X CY	X=01...04	249
Send a variable to the network, enable active Put	PT X Q1	$\overline{\text{PT}}$ X Q1	X=01...32	250
Pulse width modulation, error minimum on or off time exceeded	PW X E1	$\overline{\text{PW}}$ X E1	X=01...02	252
Send date and time via the network (easy-NET) function block	SC X Q1	$\overline{\text{SC}}$ X Q1	X=01	255
Timing relay function block	T X Q1	T X Q1	X=01...32	259

Usable relays and function blocks (coils)

MFD-Titan provides various relay types as well as function blocks and their coils for wiring in a circuit diagram.

Relay/function block	MFD-Titan display	Number	Coil	Parameter
Outputs				
MFD-Titan output relays, network stations (only network master) * = Station address 2 to 8	*Q	01...08	✓	–
MFD-Titan output relay	Q	01...08	✓	–
MFD-Titan output relay expansion, network stations (only network master) * = Station address 2 to 8	*S	01...08	✓	–
MFD-Titan expansion output relay	S	01...08	✓	–
Bit outputs * = Station address 1 to 8	*SN	01...32	✓	–
General coils				
Markers	M	01...96	✓	–
COM slave marker (REMOTE MARKER)	1M	01...96	✓	–
Jump label	:	01...32	✓	–
Function blocks				
Analog value comparator function block	A	01...32	–	✓
Arithmetic function block	AR	01...32	–	✓
Data block comparator, activate	BC X EN	01 32	✓	✓
Transfer data block, trigger coil	BT X T_	01 32	✓	✓
Boolean operation	BV	01...32	–	✓
Counter function block, counter input	C X C_	X=01...32	✓	✓

Relay/function block	MFD-Titan display	Number	Coil	Parameter
Counter function block, direction	C X D_	X=01...32	✓	✓
Counter function block, set counter value (Preset)	C X SE	X=01...32	✓	✓
Counter function block, reset counter value	C X RE	X=01...32	✓	✓
Frequency counter function block, activate counter (enable)	CF X EN	X=01...04	✓	✓
High-speed counter function block, direction	CH X D_	X=01...04	✓	✓
High-speed counter function block, activate counter (enable)	CH X EN	X=01...04	✓	✓
High-speed counter function block, set counter value (Preset)	CH X SE	X=01...04	✓	✓
High-speed counter function block, reset counter value	CH X RE	X=01...04	✓	✓
Incremental encoder counter function block, set counter value (Preset)	CI X SE	X=01...02	✓	✓
Incremental encoder counter function block, activate counter (enable)	CI X EN	X=01...02	✓	✓
Incremental encoder counter function block, reset counter value	CI X RE	X=01...02	✓	✓
Comparator function block	CP	X=01...32	–	✓
Activate text output function block (enable)	D X EN	X=01...32	✓	✓
Data function block, trigger coil	DB X T_	X=01...32	✓	✓
PID controller, activate	DC X EN	X=01...32	✓	✓
PID controller, activate P component	DC X EP	X=01...32	✓	✓
PID controller, activate I component	DC X EI	X=01...32	✓	✓
PID controller, activate D component	DC X ED	X=01...32	✓	✓
PID controller, accept manual manipulated variable	DC X SE	X=01...32	✓	✓
Activate signal smoothing filter	FT X EN	X=01...32	✓	✓

Relay/function block	MFD-Titan display	Number	Coil	Parameter
Get from network station function block	GT	X=01...32	–	✓
Seven-day time switch	HW	X=01...32	–	✓
Year time switch function block	HY	X=01...32	–	✓
Activate value scaling function block	LS X EN	X=01...32	✓	✓
Master reset function block	MR X T_	X=01...32	✓	✓
Activate numerical converter function block	NC X EN	X=01 32	✓	✓
Function block operating hours counter, enable	OT X EN	X=01...04	✓	✓
Operating hours counter function block, reset	OT X RE	X=01...04	✓	✓
Send to the network (easy-NET) function block, trigger	PT X T_	X=01...32	✓	✓
Activate pulse width modulation function block	PW X EN	X=01...02	✓	✓
Send time to the network (easy-NET) function block, trigger	SC X T_	X=01	✓	–
Activate set cycle time function block	ST X EN	X=01		
Timing relay function block, trigger control coil (enable)	T X EN	X=01...32	✓	✓
Timing relay function block, stop	T X ST	X=01...32	✓	✓
Timing relay function block, reset	T X RE	X=01...32	✓	✓
Activate value limitation function block	VC X EN	X=01...32	✓	✓

The switching behaviour of these relays is set by the coil functions and parameters selected.

The options for setting output and marker relays are listed with the description of each coil function.

The function block coil functions and parameters are listed with the description of each function block type.

Markers, analog operands

Specific markers are available for actively addressing values or inputs/outputs.

Table 8: Markers

Markers Analog operand	MFD-Titan display	Number	Value range	Access type r = Read w = Write
Marker 32 bit	MD	01...96	32 bit	r, w
Marker 16 bit	MW	01...96	16 bit	r, w
Marker 8 bit	MB	01...96	8 bit	r, w
Marker 1 bit	M	1...96	1 bit	r, w
Analog inputs basic unit	IA X	X=01...04	10 bit	r
Analog output	QA X	X=01	10 bit	r, w

When using the COM communication mode, you can make the following data accesses on the slave. Note the REMOTE MARKER SETTING in the following table.

Markers Analog operand	MFD-Titan display	Number	Value range	Access type r = Read w = Write
Marker 32 bit	1MD	01...20	32 bit	r, w
Marker 16 bit	1MW	01...40	16 bit	r, w
Marker 8 bit	1MB	01...80	8 bit	r, w
Marker 1 bit	1M	1...96	1 bit	r, w
Analog inputs basic unit	1IA X	X=01...04	10 bit	r
Analog output	1QA X	X=01	10 bit	r

The following rules apply if you want to use selective binary operands (contacts) from the markers MD, MW, MB:

Table 9: Composition of the markers

Applies to MD, MW, MB, M	Left = most significant bit, byte, word			Right = least significant bit, byte, word
32 bit	MD1			
16 bit	MW2		MW1	
8 bit	MB4	MB3	MB2	MB1
1 bit	M32 to M25	M24 to M17	M16 to M9	M8 to M1
32 bit	MD2			
16 bit	MW4		MW3	
8 bit	MB8	MB7	MB6	MB5
1 bit	M64 to M57	M56 to M49	M48 to M41	M40 to M33
32 bit	MD3			
16 bit	MW6		MW5	
8 bit	MB12	MB11	MB10	MB9
1 bit	M96 to M89	M88 to M81	M80 to M73	M72 to M65
32 bit	MD4			

Applies to MD, MW, MB, M	Left = most significant bit, byte, word			Right = least significant bit, byte, word
16 bit	MW8		MW7	
8 bit	MB16	MB15	MB14	MB13
32 bit	MD5			
16 bit	MW10		MW9	
8 bit	MB20	MB19	MB18	MB17
…				
…				
…				
32 bit	MD23			
16 bit	MW46		MW45	
8 bit	MB92	MB91	MB90	MB89
32 bit	MD24			
16 bit	MW48		MW47	
8 bit	MB96	MB95	MB94	MB93
32 bit	MD25			
16 bit	MW50		MW49	
32 bit	MD26			
16 bit	MW52		MW51	
…				
…				
32 bit	MD48			
16 bit	MW96		MW95	
32 bit	MD49			
32 bit	MD50			
…				
32 bit	MD95			
32 bit	MD96			

→ You should only write the markers once.

Marker double words always contain all data formats. When several write accesses to MD, MW, MB or M (within an MD) are made, it is the last write operation that is retained. This also applies if you are writing markers from a visualization screen.

Number formats

MFD makes computations with a signed 31 bit value.

The value range is:
–2 147 483 648 to +2 147 483 647

With a 31 bit value, the 32nd bit is the sign bit.

Bit 32 = state "0" means a positive number.

Example:
$0000\,0000\,0000\,0000\,0000\,0100\,0001\,0010_{bin}$ = 412_{hex} = 1042_{dec}

Bit 32 = 1 means a negative number.

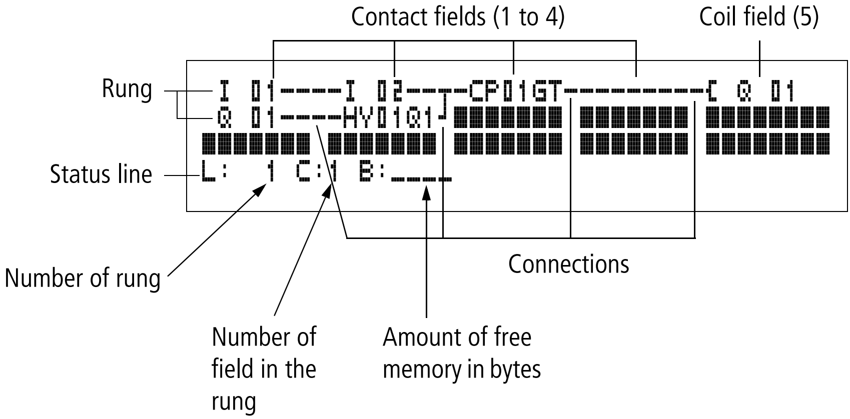
Example:
$1111\,1111\,1111\,1111\,1101\,1100\,1010\,1110_{bin}$ = $FFFFDCAE_{hex}$ = -9042_{dec}

→ The marker byte (MB) and marker word (MW) number formats are unsigned.

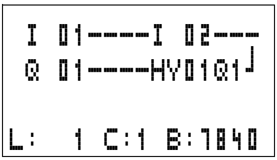
Circuit diagram display

In the MFD-Titan circuit diagram, contacts and coils of relays are connected up from left to right - from the contact to the coil. The circuit diagram is created on a hidden wiring grid containing contact fields, coil fields and rungs. It is then wired up with connections.

- Insert contacts in the four **contact fields**. The first contact field on the left is automatically connected to the voltage.
- Insert the relay coil to be controlled together with its function and designation in the **coil field**. The coil designation consists of the coil name, coil number and function block from the function designation. The coil function defines the method of operation of the coil.
- Every line in the circuit diagram forms a **rung**. With the MFD-Titan up to 256 rungs can be wired in a circuit diagram.



- **Connections** are used to produce the electrical contact between relay contacts and the coils. They can be created across several rungs. Each point of intersection is a connection.
- The number of free bytes is displayed so that you can recognise how much **memory** is available for the circuit diagram and function blocks.



MFD-Titan circuit diagram display

For greater legibility, the circuit diagram display of the MFD-Titan shows two contacts per rung or one contact plus a coil in series. A total of 16 characters per rung and three rungs plus the status line can be displayed simultaneously.

You can move between the contact fields with the ‹ › cursor buttons. The number of the rung and the contact are displayed in the lower status line.

→ The circuit diagram display performs two functions:

- In STOP mode it is used to edit the circuit diagram.
- In RUN mode it is used to check the circuit diagram using the Power flow display.

Saving and loading programs

MFD-Titan provides you with two ways of saving circuit diagrams externally:

- Saving to a memory card.
- Saving on a PC with EASY-SOFT-PRO.

Once they have been saved, programs can be reloaded into MFD-Titan, edited and run.

All program data is saved in MFD-Titan. In the event of a power failure the data will be retained until the next time it is overwritten or deleted.

Memory card
Each memory card contains a circuit diagram which is inserted into the MFD-Titan interface.

The MFD-Titan behaves in the following manner in accordance with the type and setting.

Requirement:
A valid circuit diagram must be stored on the card.

Variants with display:

▶ Go to the CARD menu and load the circuit diagram into the unit in STOP mode via CARD → DEVICE.

CARD MODE setting → page 376.

Variants without display:
If the circuit diagram on the card is different to the circuit diagram in the device, the program from the card is loaded as soon as the power supply is turned on.

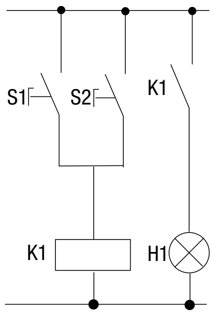
EASY-SOFT-PRO
EASY-SOFT-PRO is a PC program for creating, testing and managing MFD-Titan programs, visualization applications and circuit diagrams.

Completed programs are exchanged between the PC and MFD-Titan via the connection cable. Once you have transferred a circuit diagram, you can start MFD-Titan straight from your PC.

Working with contacts and relays

In MFD-Titan circuit diagrams, the switches, buttons and relays of conventional circuit diagrams are connected up using input contacts and relay coils.

Conventional circuit



MFD-Titan circuit diagram

MFD-Titan connection
Connect make contact S1 to input terminal I1
Connect make contact S2 to input terminal I2
Connect load H1 to output Q1

S1 or S2 switch on H1.

MFD-Titan circuit diagram:

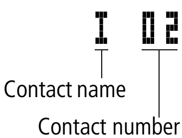
```
I 01--┬-----------------------[ Q 01
I 02--┘
```

First specify which input and output terminals you wish to use in your circuit.

The signal states on the input terminals are detected in the circuit diagram with the input contacts I, R* or RN. The outputs are switched in the circuit diagram with the output relays Q, S or SN.

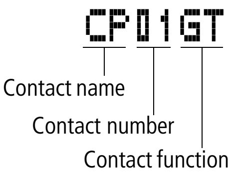
Entering and changing contacts and relay function coils

Contacts



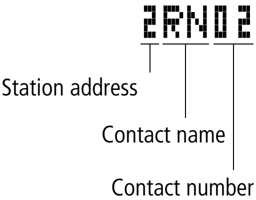
An input contact is selected in the MFD-Titan via the contact name and contact number.

Example: input contact



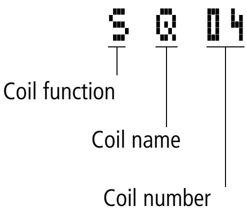
A contact of a function relay is assigned the name of the function block, the number and the contact function.

Example: contact of comparator function block



If the contact on a network station is used, the address of the station is placed before the contact name.

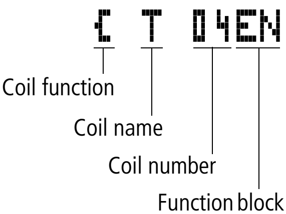
Example: contact of an easy-NET station



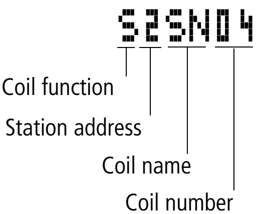
Coils

With a relay coil or function block select the coil function, coil or function block name, coil or function block number as well as coil of the function block. With coils of an easy-NET network station, the network address is selected before the coil name.

Example: relay coil output



Relay coil for timing relay function block with control coil



Relay coil of an easy-NET station

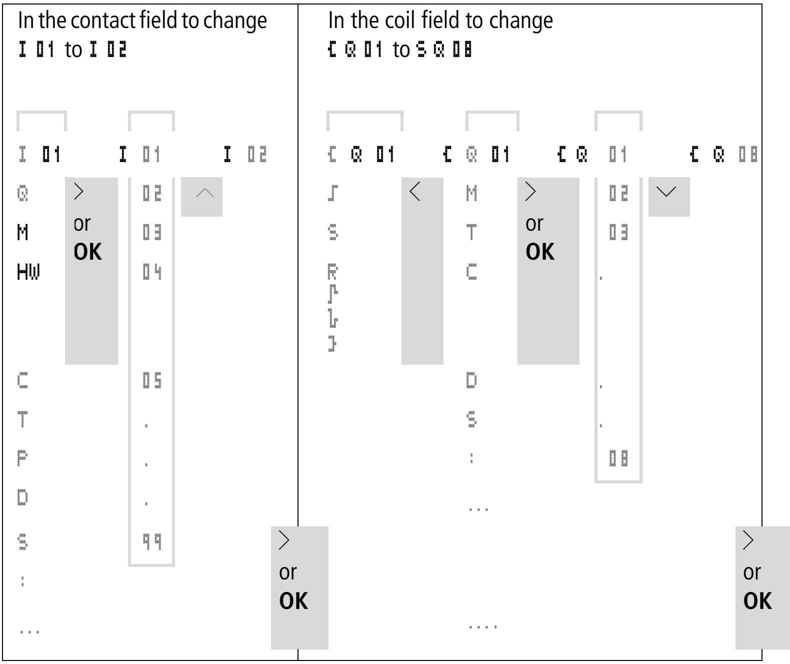
→ A full list of all the contacts and relays is given in the overview starting on Page 122.

I 01 Values for contacts and coil fields are changed in Entry mode. The value to be changed flashes.

→ MFD-Titan proposes the contact I 01 or the coil [Q 01 when starting entries in an empty field.

- ▶ Move the cursor using the buttons ‹ › ˄˅ to a contact or coil field.
- ▶ Press **OK** to switch to Entry mode.
- ▶ Use ‹ › to select the position you wish to change, or press **OK** to jump to the next position.
- ▶ Use ˄˅ to modify the value at the position.

MFD-Titan will leave Entry mode when you press ‹ › or **OK** to leave a contact field or coil field.



Deleting contacts and coils

▶ Move the cursor using the buttons ‹ › ˄˅ to a contact or coil field.

▶ Press **DEL**.

The contact or the coil will be deleted, together with any connections.

Changing make contacts to break contacts

Every contact in the MFD-Titan circuit diagram can be defined as either a make contact or a break contact.

▶ Switch to Entry mode and move the cursor over the contact name.

▶ Press **ALT.** The make contact will change to a break contact.

▶ Press **OK** 2 × to confirm the change.

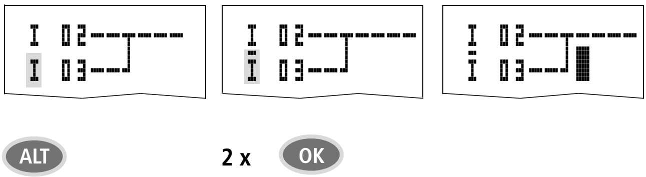


Figure 79: Change contact I 03 from make to break

Creating and modifying connections

Contacts and relay coils are connected with the arrow in the Connect mode. MFD-Titan displays the cursor in this mode as an arrow.

▶ Use ‹ › ˄˅ to move the cursor onto the contact field or coil field from which you wish to create a connection.

→ Do not position the cursor on the first contact field. At this position the **ALT** button has a different function (Insert rung).

- ▶ Press **ALT** to switch to Connect mode.
- ▶ Use ‹ › to move the diagonal arrow between the contact fields and coil fields and ⌃⌄ to move between rungs.
- ▶ Press **ALT** to leave Connect mode.

MFD-Titan will leave the mode automatically when you move the diagonal arrow onto a contact field or coil field which has already been assigned.

→ In a rung, MFD-Titan automatically connects contacts and the terminal to the relay coil if there are no empty fields in-between.

Never work backwards. You will learn why wiring backwards does not work in Section "Effects on the creation of the circuit diagram" on Page 390.

```
I 01----Q 04----Ī 03--┐
      ┌-----------------┘
      └-I 02----I 04------------S Q 02
```

Figure 80: Circuit diagram with five contacts, invalid

When wiring more than four contacts in series, use one of the 96 M marker relays.

```
I 01----Q 04----Ī 03------------S M 01
I 02----I 04----M 01------------S Q 02
```

Figure 81: Circuit diagram with M marker relay

Deleting connections

- ▶ Move the cursor onto the contact field or coil field to the right of the connection that you want to delete. Press **ALT** to switch to Connect mode.
- ▶ Press **DEL**.

MFD-Titan will delete a connection. Closed adjacent connections will be retained.

If several rungs are connected to one another, MFD-Titan first deletes the vertical connection. If you press **DEL** again, it will delete the horizontal connection as well.

→ You cannot delete connections that MFD-Titan has created automatically.

Close the delete operation with **ALT** or by moving the cursor to a contact or coil field.

Inserting and deleting a rung

The MFD-Titan circuit diagram display shows three of the 256 rungs on the display at the same time. MFD-Titan automatically scrolls up or down the display to show hidden rungs – even empty ones – if you move the cursor past the top or bottom of the display.

A new rung is added below the last connection or inserted above the cursor position:

▶ Position the cursor on the **first** contact field of a rung.

▶ Press **ALT**.

The existing rung with all its additional connections is “shifted” downwards. The cursor is then positioned directly in the new rung.

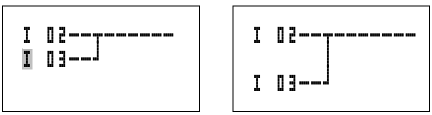
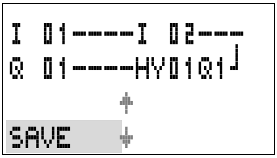


Figure 82: Inserting a new rung

Saving circuit diagrams

▶ Press the **ESC** button to save a circuit diagram.

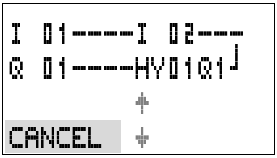


The menu on the left appears in the status line.

▶ Press **OK** to save the entire program, circuit diagram and function blocks.

After saving you will be in the CIRCUIT DIAGRAM menu.

Aborting circuit diagram entry



▶ If you want to exit without saving the circuit diagram, press **ESC**.

▶ Use the cursor buttons ∧∨ to select the CANCEL menu.

▶ Press **OK**.

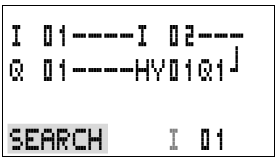
The circuit diagram is closed without saving.

Searching for contacts and coils

You can search for contacts and coils in the following way:

▶ Press **ESC**. Use the cursor buttons ∧∨ to select the SEARCH menu.

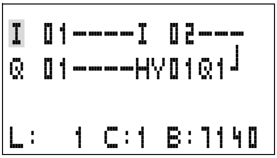
▶ Press **OK**.



▶ Select the desired contact, coil and number with the ∨ and ‹ › cursor buttons.

With function relays, select the function block, the number and the coil.

▶ Confirm the search with the **OK** button.



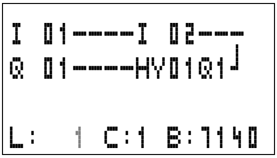
The device will search for the first occurrence of the contact or coil from the start of the search to the end of the circuit diagram. If no contact or coil is found, the MFD-Titan circuit diagram editor will continue the search from the start of the circuit diagram. If a contact or coil is found, the MFD-Titan editor automatically jumps to the respective field in the circuit diagram.

"Go to" a rung

The MFD-Titan circuit diagram editor provides a Go To function in order to enable fast access to a rung.

▶ Press **ESC** and use the ⌃⌄ cursor buttons to select the GO TO menu.

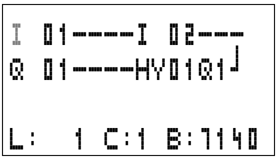
▶ Press **OK**.



▶ Select the required rung (L... …) with the ⌃⌄ cursor buttons.

The first contact on the rung is always indicated.

▶ Press **OK**.



The cursor remains stationary at the required rung contact L 1.

Deleting the rung

MFD-Titan only removes empty rungs (without contacts or coils).

▶ Delete all the contacts and coils from the rung.

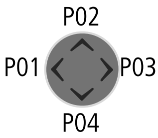
▶ Position the cursor on the first contact field of the empty rung.

▶ Press **DEL**.

The subsequent rung(s) will be "pulled up" and any existing links between rungs will be retained.

Switching via the cursor buttons

With MFD-Titan, you can also use the four cursor buttons as hard-wired inputs in the circuit diagram.



The buttons are wired in the circuit diagram as contacts P 01 to P 04. The P buttons can be activated and deactivated in the → System menu.

The P buttons can also be used for testing circuits or manual operation. These button functions are also useful for servicing and commissioning purposes.

Example 1
A lamp at output Q1 is switched on and off via inputs I1 and I2 or by using cursor buttons ⌃⌄.

```
I 01--┬-------------------------S Q 01
P 02--┘
I 02--┬-------------------------R Q 01
P 04--┘
```

Figure 83: Switch Q1 via I1, I2, ⌃, or ⌄

Example 2
Input I1 is used to control output Q1. I5 switches over to cursor operation and via M 01 disconnects the rung I 01.

```
I 05----------------------------S M 01
I 01----M̄ 01--┬-----------------S Q 01
P 01----M 01--┘
```

Figure 84: I5 switches over to the cursor buttons.

→ The P buttons are only detected as switches in the Status menu.

```
I123456789...
             P2
MO 14:55
Q.2...6.8    RUN
```

The Status menu display shows whether the P buttons are used in the circuit diagram.

Displayed on the Status display:

- P: button function wired and active,
- P2: button function wired, active and P2 button ^ pressed,
- P-: button function wired and not active,
- Empty field: P buttons not used.

Checking the circuit diagram

MFD-Titan contains a built-in measuring device enabling you to monitor the switching states of contacts, relays and function block coils during operation.

▶ Create the small parallel circuit below and save it.

```
I 02--┬-------------------------S Q 01
I 03--┘
```

Figure 85: Parallel circuit

▶ Switch MFD-Titan to RUN mode via the main menu.

▶ Return to the circuit diagram display.

You are now unable to edit the circuit diagram.

→ If you switch to the circuit diagram display and are unable to modify a circuit diagram, first check whether MFD-Titan is in STOP mode.

The circuit diagram display performs two functions depending on the mode:

- STOP: Creation of the circuit diagram,
- RUN: Power flow display.

▶ Switch on I3.

```
 I 02--┬------------------------S Q 04
 I 03--┘

L:001 C:1  RUN
```

Figure 86: Power flow display

In the power flow display, energized connections are thicker than non-energized connections.

You can follow a current-carrying connection across all rungs by scrolling the display up and down.

The bottom right of the power flow display indicates that the controller is in RUN mode. (→ Section "Power flow display with Zoom function" Page 95).

→ The power flow display will not show signal fluctuations in the millisecond range. This is due to the inherent delay factor of LCD displays.

Function block editor

The MFD-Titan has the FUNCTION RELAYS menu in order to edit the function blocks without circuit diagrams. The function blocks are an inherent component of the program.

Calling the function blocks via the FUNCTION RELAYS menu

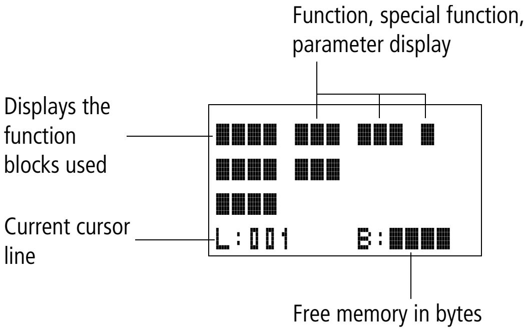


Figure 87: Explanation of the function block display

Display of the function blocks for editing

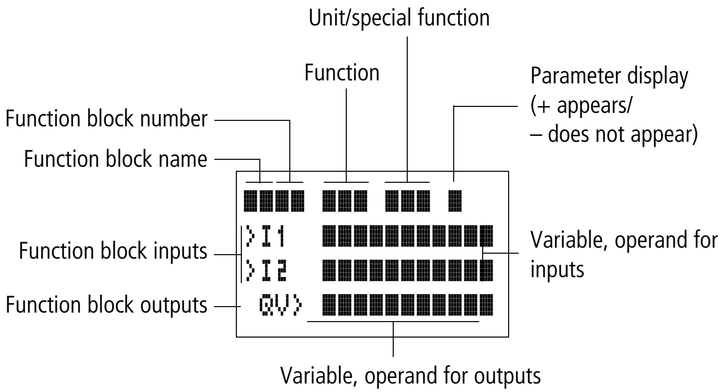
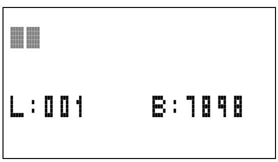


Figure 88: Function block display during editing

Editing function blocks

▶ Go to the FUNCTION RELAYS menu.

▶ Press the **OK** button.

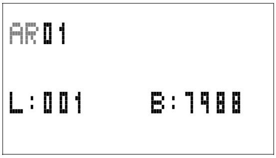


The following display appears if no function blocks are present.

The cursor flashes.

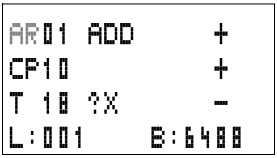
▶ Press the **OK** button.

The editor for inputting a function block is displayed.



Select the desired function block and number with the ∧∨< > cursor buttons.

The functions of the individual function blocks are explained in the individual function block descriptions on the following pages.

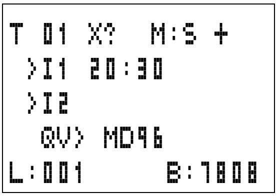


This display appears if there are function blocks present.

The function blocks are created in the sequence in which they were edited.

Calling up function blocks from the circuit diagram

If you enter a function block parameter from the circuit diagram, you will jump from the circuit diagram editor to the function block editor automatically. Once you have assigned the parameters, you will return to the position where you left the circuit diagram with Save or Cancel. The operation is carried out in the same way as with circuit diagram operation.



Example: timing relay function block

Function block:	Timing relay
Switch function:	On-delayed with random switching
Time range:	M:S (Minute:Seconds)
Set time >I1:	20 min 30 s
Actual time QV>:	Copied to MD96

Assigning operands to an > input of a function block

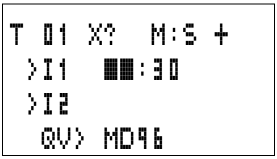
→ Only the following variables can be assigned to the input of a function block:

- Constants, e.g.: 42,
- Markers such as MD, MW, MB,
- Analog output QA,
- Analog inputs IA,
- All output variables of the function blocks …QV>

Assigning operands to a QV> output of a function block

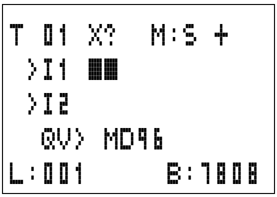
→ Only markers such as MD, MW, MB or the analog output QA can be assigned to a variable output of a function block.

Deleting operands on the function block inputs/ outputs



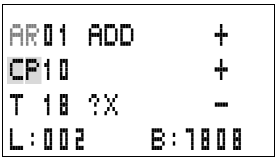
Position the cursor on the required operand.

▶ Press the **DEL** button.



The operand is deleted.

Deleting an entire function block



Ensure that all contacts and coils of the function block are deleted.

▶ Select the required function block from the list.

In this case CP10.

▶ Press the **DEL** button.

```
AR01 ADD     +
T 18 ?X      -

L:001
```

The function block is deleted.

Checking function blocks

You can check function blocks in the same way as circuit diagrams. The device is in RUN mode.

Checking from the circuit diagram:
 Position the cursor on a contact or a coil of the required function block. Press **OK**.

```
T 01 X?  M:S +
 >I1  20:30
 >I2
 QV> 14:42
.. EN..
```

The function block will be displayed, in this case a timing relay.

- >I1= set time of the timing relay,
- QV> = the actual value is 14 minutes 42 seconds,
- The enable coil is actuated, EN is visible.

If a coil of a function block is actuated in RUN mode, the coil name with the coil designation will appear on the display.

Checking the function block via the function block editor:
You access the function block list via the FUNCTION RELAYS menu.

Select the required function block:

```
AR01 ADD     +
CP10         +
T 18 ?X      -
L:001    RUN
```

In this case the arithmetic function block AR01 in the Adder mode.

▶ Press the **OK** button.

```
AR01 ADD     +
 >I1  20056
 >I2   1095
  QV> 21151
```

The function block is presented with the actual values and the result.

Displaying the operands when checking the function blocks:
If you want to know which operands are used on the function block inputs and outputs when checking the function block, press the **ALT** button on the displayed value.

```
AR01 ADD      +
 >I1   C 01QV>
 >I2    1095
  QV> MD 56
```

The operand is displayed.

- >I1 = Actual value of counter C 01
- >I2 = Constant 1095
- QV> = Marker double word MD56

▶ Press the **ALT** button again.

```
AR01 ADD      +
 >I1   20056
 >I2    1095
  QV> 21151
```

The display shows the values.

Coil functions

You can set the coil function to determine the switching behaviour of relay coils. The following coil functions are assigned to all coils:

Table 10: Coil function

MFD-Titan display	Coil function	Example
Ɛ	Contactor function	ƐQ01,ƐD02,ƐS04,Ɛ:01,ƐM07,...
ʃ	Impulse relay function	ʃQ03,ʃM04,ʃD08,ʃS07,ʃ:01,...
S	Set	SQ08,SM02,SD03,SS04..
R	Reset	RQ04,RM05,RD07,RS03..
Ǝ	Contactor function with negated result	ƎQ06, ƎM96..
Γ	Cycle pulse with rising edge	ΓM01..
L	Cycle pulse with falling edge	LM42..

→ The function block descriptions state which coil functions can be used with the function block concerned.

Rules for wiring relay coils

Relay with contactor function

→ A coil should only be used once in order to retain an overview of the relay states. However, retentive coil functions such as S, R, ſ can be used several times.

The following applies to non-retentive coil functions such as [(contactor),] (negated contactor), ⌈, ⌊ (rising and falling edge detection): Each coil must only be used once. The last coil in the circuit diagram determines the status of the relay.

Exception: When working with jumps, the same coil can be used twice.

Coil with contactor function [

The output signal follows immediately after the input signal and the relay acts as a contactor.

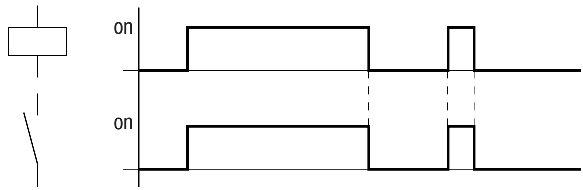


Figure 89: Signal diagram of contactor function

Impulse relay ſ

The relay coil switches whenever the input signal changes from 0 to 1. The relay behaves like a bistable flip-flop.

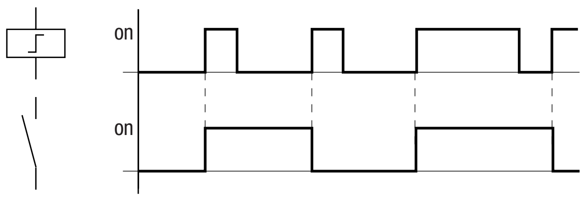


Figure 90: Signal diagram of impulse relay

A coil is automatically switched off if the power fails and if STOP mode is active. Exception: Retentive coils retain signal 1 (see → Section “Retention”, Page 380).

“Set” S and “Reset” R coil function

The “Set” S and “Reset” R coil functions are normally used in pairs.

The relay picks up when the coil is set (A) and remains in this state until it is reset (B) by the coil function.

The supply voltage is switched off (C), the coil does not have a retentive effect.

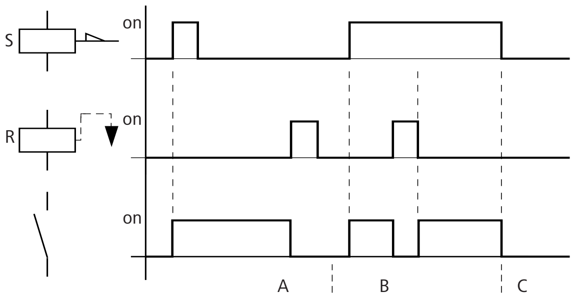


Figure 91: Signal diagram of “Set” and “Reset”

If both coils are triggered at the same time, priority is given to the coil in the circuit diagram with the higher rung number. This is shown in the above signal diagram in section B.

```
I 05----------------------------S Q 01
I 10----------------------------R Q 01
```

Figure 92: Simultaneous triggering of Q 01

In the example above, the reset coil has priority with simultaneous triggering of the set and reset coils.

Coil negation (inverse contactor function) ⌉

The output signal is simply an inversion of the input signal; the relay operates like a contactor with contacts that have been negated. If the coil is triggered with the 1 state, the coil switches its make contacts to the 0 state.

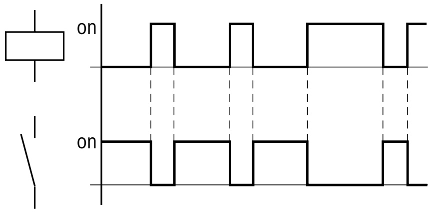


Figure 93: Signal diagram of inverse contactor function

Rising edge evaluation (cycle pulse)

If the coil is only meant to switch on a rising edge, this function will be applied. With a change in the coil state from 0 to 1, the coil switches its make contacts to the 1 state for one cycle.

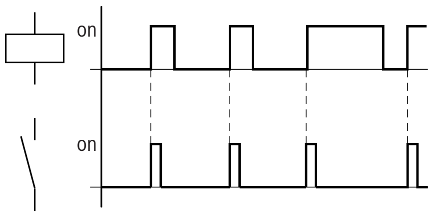


Figure 94: Signal diagram of cycle pulse with rising edge

Falling edge evaluation (cycle pulse)

If the coil is only meant to switch on a falling edge, this function will be applied. With a change in the coil state from 1 to 0, the coil switches its make contacts to the 1 state for one cycle.

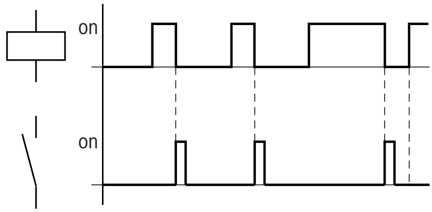


Figure 95: Signal diagram of cycle pulse with falling edge

→ A set coil is automatically switched off if the power fails and if the device is in STOP mode. Exception: Retentive coils retain signal 1 (see → Section "Retention", Page 380).

Function blocks

The function blocks are used to simulate some of the devices used in conventional open-loop and closed-loop control systems. MFD-Titan provides the following function blocks:

- Analog value comparator/threshold controller (only with MFD-Titan 24 V DC variants)
- Arithmetic,
  - addition, subtraction, multiplication, division
- Compare data blocks
- Transfer data blocks
- Boolean operation
- Counters,
  - up and down counters with upper and lower threshold values, preset
  - frequency counters,
  - high-speed counters,
  - incremental encoder counters
- Comparators
- Text, output freely editable texts, enter values
- Data function block
- PID controllers
- Smoothing filters
- Value scaling
- Pulse width modulator
- Read (GET) data from the easy-NET
- Time switches,
  - weekday/time
  - year, month, day (date),
- Numerical converters
- Master reset
- Operating hours counter
- Write (PUT) data to the easy-NET
- Synchronisation of date and time via the easy-NET
- Timing relays
  - on-delayed,
  - on-delayed with random switching,

– off-delayed, also retriggerable,
– off-delayed with random switching, also retriggerable,
– on and off delayed,
– on and off delayed with random switching,
– single pulse,
– synchronous flashing,
– asynchronous flashing,

- Set cycle time
- Value limitation

The following applies to function blocks:

→ The most recent actual values are cleared if the power supply is switched off or if MFD-Titan is switched to STOP mode. Exception: Retentive data keeps its state (→ Section "Retention", Page 380).

The most recent actual values are transferred to the operands every cycle. The data function block is an exception.

→ **Attention!**
The following applies to RUN mode: MFD-Titan processes the function block after a pass through the circuit diagram. The last state of the coils is used for this.

→ If you want to prevent other people from modifying the parameters, change the access enable symbol from "+" to "–" when creating the circuit diagram and setting parameters and protect the circuit diagram with a password.

→ **Attention!**
The function blocks are designed so that a function block output can be assigned directly to the input of another function block. This enables you always to have an overview of which value is transferred.

If different data formats are used, such as if the first function block uses 32 bits and an 8-bit or 16-bit format is used for further processing, sign value errors or value errors may occur when transferring from one function block to another one.

Analog value comparator/threshold value switch

MFD-Titan provides 32 analog value comparators from A 01 to A 32.

With an analog value comparator or threshold value switch you can, for example, compare analog input values with a setpoint value.

All MFD-Titan DC variants have analog inputs.

The following comparisons are possible:

- Function block input >I1 **greater than or equal to, equal to, less than or equal to** function block input >I2
- Using the factors >F1 and >F2 as inputs enables you to amplify and adjust the values of the function block inputs.
- The >OS function block input can be used as an offset for the >I1 input.
- The >HY function block input is used for the positive and negative switching hysteresis of the input >I2. The contact switches according to the selected comparison mode of the function block.

```
I 01----A 01Q1------------------Ä Q 01
I 02--┬-A 02Q1------------------S Q 02
      └-A 03Q1------------------R Q 03
```

Figure 96: MFD-Titan circuit diagram with analog value comparators

```
A 02 GT       +
 >I1
 >F1
 >I2
 >F2
 >OS
 >HY
```

Parameter display and parameter set for analog value comparators:

A 02	Function block analog value comparator number 02
GT	Greater than mode
+	Appears in the parameter display
>I1	Comparison value 1
>F1	Gain factor for >I1 (>I1 = >F1 × value)
>I2	Comparison value 2
>F2	Gain factor for >I2 (>I2 = >F2 × value)
>OS	Offset for the value of >I1
>HY	Switching hysteresis for value >I2 (Value HY applies to positive and negative hysteresis.)

Inputs

The function block inputs >I1, >F1, >I2, >F2, >OS and >HY can have the following operands:

- Constants
- Markers MD, MW, MB
- Analog inputs IA01 to IA04
  - IA01: terminal I7
  - IA02: terminal I8
  - IA03: terminal I11
  - IA04: terminal I12
- Analog output QA01
- Actual value … QV> of another function block

Analog value comparator operating modes

Parameter	Function
GT	>I1 greater than >I2
EQ	>I1 equal to >I2
LT	>I1 less than >I2

Contacts
A 01Q1 to A 32Q1

Memory requirement of the analog value comparator
The analog value comparator function block requires 68 bytes of memory plus 4 bytes per constant on the function block inputs.

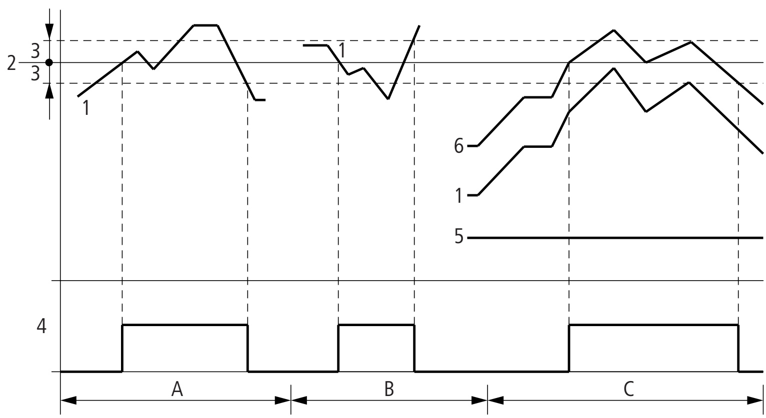


Figure 97: Signal diagram of the analog value comparator

1: actual value on >I1

2: setpoint value on >I2

3: hysteresis on >HY

4: switching contact (make contact)

5: offset for value >I1

6: actual value plus offset

- Range A: Compare >I1 > >I2
  - The actual value >I1 increases.
  - The contact switches when the actual reaches the setpoint value.
  - The actual value changes and falls below the value of the setpoint value minus the hysteresis.
  - The contact goes to the normal position.
- Range B: Compare >I1 < >I2
  - The actual value drops.
  - The contact switches if the actual reaches the setpoint value.
  - The actual value changes and rises above the value of the setpoint value plus hysteresis.
  - The contact goes to the normal position.
- Range C: Compare >I1 > >I2 with offset
  - This example behaves as described in Range A. The offset value is added to the actual value.
- Comparison >I1 = >I2
  The contact switches on:
  - If the setpoint is exceeded with the actual value rising.
  - If the setpoint is undershot with the actual value decreasing.

  The contact switches off:
  - If the hysteresis limit is exceeded with the actual value rising.
  - If the hysteresis limit is undershot with the actual value decreasing.

Arithmetic function block

MFD-Titan provides 32 arithmetic function blocks AR01 to AR32.

The arithmetic function block is used for arithmetic operations. All four basic arithmetic operations are supported:

- add,
- subtract,
- multiply,
- divide.

Inputs

The function block inputs >I1 and >I2 can have the following operands:

- Constants
- Markers MD, MW, MB
- Analog inputs IA01 to IA04
  - IA01: terminal I7
  - IA02: terminal I8
  - IA03: terminal I11
  - IA04: terminal I12
- Analog output QA01
- Actual value … QV> of another function block

Actual value …QV>

The actual value …QV> can be assigned the following operands:

- Markers MD, MW, MB
- Analog output QA01

An arithmetic function block is not wired in the circuit diagram.

```
AR32 ADD     +
 >I1
 >I2
  QV>
```

Parameter display and parameter set for arithmetic function blocks:

AR32	Arithmetic function block number 32
ADD	Addition mode
+	Appears in the parameter display
>I1	First value
>I2	Second value
QV>	Result of the addition

Only constants can be modified in the parameter display of a function block.

Arithmetic function block modes

Parameter	Function
ADD	Addition of summand value >I1 plus summand >I2
SUB	Subtraction of minuend >I1 minus subtrahend >I2
MUL	Multiplication of factor >I1 by factor >I2
DIV	Division of dividend >I1 by divisor >I2

Value range

The function block operates in the integer range from –2 147 483 648 to +2 147 483 647.

Behaviour when value range is exceeded

- The function block sets the switching contact AR..CY to status 1.
- The function block retains the value of the last valid operation. The value is zero when it is first called.

Displaying the parameter set in the PARAMETERS menu

- + Access enabled
- – Access disabled

Contacts

AR01CY to AR32CY: CARRY overflow bit, value on function block output greater than or less than the value range.

AR01ZE to AR32ZE: ZERO zero bit, value on function block output is equal to zero.

Coils

The arithmetic function block does not have any coils.

Memory requirement of the arithmetic function block

The arithmetic function block requires 40 bytes of memory plus 4 bytes per constant on the function block inputs.

Addition

42 + 1000 = 1042

2 147 483 647 + 1 = last valid value of this arithmetic operation, due to overflow (CARRY)
AR..CY = Status 1

–2 048 +1 000 = –1048

Subtraction
1 134 – 42 =1 092

–2 147 483 648 – 3 = last valid value of this arithmetic operation, due to overflow (CARRY)
AR..CY = Status 1

–4 096 – 1 000 = –5 096

–4 096 – (–1 000) = –3 096

Multiplication
12 × 12 = 144

1 000 042 × 2 401 = last valid value of this arithmetic operation, due to overflow (CARRY)
Correct value = 2 401 100 842
AR..CY = Status 1

–1 000 × 10 = –10 000

Division
1 024: 256 = 4

1 024: 35 = 29 (the places after the decimal point are omitted.)

1 024: 0 = last valid value of this arithmetic operation, due to overflow (CARRY)
(mathematically correct: "Infinite")
AR..CY = Status 1

–1 000: 10 = –100

1 000: –10 = –100

–1 000: (–10) = 100

10: 100 = 0

Data block comparator

MFD-Titan provides 32 function blocks BC01 to BC32 for comparing values of two consistent marker ranges. The comparison is in byte format. The following marker types can be compared:

- MB,
- MW,
- MD.

The function block is enabled in the circuit diagram.

```
I 05---------------------------[ BC27EN
BC27E1┐
BC27E2┤
BC27E3┴------------------------[ M 48
BC27EQ-BC27EN------------------[ M 49
```

Figure 98: MFD-Titan circuit diagram with enabling of data block comparator function block

```
BC27           +
 >I1
 >I2
 >NO
```

Parameter display and parameter set for a data block comparator:

BC27	Data block comparator function block number 27
+	Appears in the parameter display
>I1	Start of comparison range 1
>I2	Start of comparison range 2
>NO	Number of elements to be compared in bytes per range. Value range 1 to + 383

Only constants can be modified in the parameter display of a function block.

According to the operands at the inputs >I1 and >I2 the following operating modes are possible:

Inputs

The function block inputs >I1, >I2 and >NO can have the following operands:

- Constants
- Markers MD, MW, MB
- Analog inputs IA01 to IA04
  - IA01: terminal I7
  - IA02: terminal I8
  - IA03: terminal I11
  - IA04: terminal I12
- Analog output QA01
- Actual value … QV> of another function block

Specifying the marker range without offset

If MB, MW or MD markers are specified at both >I1 and >I2 the number of the markers is the start of comparison range 1 or 2.

Specifying the marker range with offset

If you wish to work with an offset, specify one of the following variables at function block input >I1 or >I2:

- Constant,
- Actual value ..QV of a function block,
- Analog input IA..,
- Analog output QA..

The value at the input is taken as the offset to marker MB01.

Displaying the parameter set in the PARAMETERS menu

- + Access enabled
- –: Access disabled

Contacts

BC01E1 to BC32E1: the number of comparison elements exceeds one of the comparison ranges.

BC01E2 to BC32E2: the two comparison ranges overlap.

BC01E3 to BC32E3: the specified offset of the comparison ranges is outside of the permissible range.

BC01EQ to BC32EQ: output of the comparison result. Only valid if the BC..EN enable has been triggered.
Status 0 = Comparison ranges not equal,
Status 1 = Comparison ranges equal

Coils
BC01EN to BC32EN: Enable coil of the data block comparator function block.

Memory requirement of the data block comparator function block
The data block comparator function block requires 48 bytes of memory plus 4 bytes per constant on the function block inputs.

Function of the data block comparator function block
The data block comparator function block compares two consistent data blocks.

The comparison is active if the BC..EN (enable) is triggered.

→ No data blocks are compared if an error is present.

The error outputs E1, E2 and E3 are evaluated regardless of the status of the enable.

Example:
Comparison of marker blocks, definition of marker ranges direct

Two marker blocks are to be compared. Block 1 starts at MB10, Block 2 at MB40. Each block is 10 bytes long.

Parameters of BC01 function block:
Comparison range 1:>I1 MB10
Comparison range 2:>I2 MB40
Number of bytes: >NO 10

Comparison range 1	Value of marker range 1 (decimal)	Comparison range 2	Value of marker range 2 (decimal)
MB10	39	MB40	39
MB11	56	MB41	56
MB12	88	MB42	88
MB13	57	MB43	57
MB14	123	MB44	123
MB15	55	MB45	55
MB16	134	MB46	134
MB17	49	MB47	49
MB18	194	MB48	194
MB19	213	MB49	213

The comparison result of the function block BC01 is:
BC01EQ = 1, the data block ranges have the same content.

Example:
Comparison of marker blocks, definition of a marker range with offset

Two marker blocks are to be compared. Block 1 starts at MB15, Block 2 at MB65. Each block is 4 bytes long.

Parameters of BC01 function block:
Comparison range 1: >I1 MB15
Comparison range 2: >I2 64
Number of bytes: >NO 4
Marker MB01: 1

→ Comparison range 2: Constant 64:
MB01 plus Offset: 1 + 64 = 65 → MB65.

Comparison range 1	Value of marker range 1 (decimal)	Comparison range 2	Value of marker range 2 (decimal)
MB15	45	MB65	45
MB16	62	MB66	62
MB17	102	MB67	102
MB18	65	MB68	57

The comparison result of the function block BC01 is: BC01EQ = 0, the data block ranges do not have the same content.

MB18 are MB68 not identical.

Example:
Comparison of marker blocks, definition of a marker range in a different format.

Two marker blocks are to be compared. Block 1 starts at MB60, Block 2 at MD80. Each block is 6 bytes long.

Parameters of BC01 function block:
Comparison range 1:>I1 MB60
Comparison range 2:>I2 MD80
Number of bytes: >NO 6

→ The comparison is in byte format. MD80 has 4 bytes. Therefore the first two bytes of MD81 are also compared.

Comparison range 1	Value of marker range 1 (decimal/ binary)	Comparison range 2	Value of marker range 2 (decimal/ binary)
MB60	45/ 00101101	MD80 (Byte 1, LSB)	1097219629/ 010000010110011000111110**0010 1101**
MB61	62/ 00111110	MD80 (Byte 2)	1097219629/ 0100000101100110**00111110**0010 1101
MB62	102/ 01100110	MD80 (Byte 3)	1097219629/ 01000001**01100110**001111100010 1101
MB63	65/ 01000001	MD80 (Byte 4, MSB)	1097219629/ **01000001**01100110001111100010 1101
MB64	173/ 10101101	MD81 (Byte 1, LSB)	15277/ 00111011**10101101**
MB65	59/ 00111011	MD81 (Byte 2)	15277/ **00001000**10101101

The comparison result of the function block BC01 is: BC01EQ = 0, the data block ranges do not have the same content.

MB65 and MD81 (Byte 2) are not identical.

Example:
Comparison of marker blocks, range violation error.

Two marker blocks are to be compared. Block 1 starts at MD60, Block 2 at MD90. Each block is 30 bytes long.

Parameters of BC01 function block:
Comparison range 1:>I1 MD60
Comparison range 2:>I2 MD90
Number of bytes: >NO 30

→ The comparison is in byte format. MD90 to MD96 is 28 bytes. The number of bytes is 30 bytes.

The error message “Number of comparison elements exceeds one of the comparison ranges” is output.

BC01E1 is 1.

Example
Comparison of marker blocks, range overlap error.

Two marker blocks are to be compared. Block 1 starts at MW60, Block 2 at MW64. Each block is 12 bytes long.

Parameters of BC01 function block:
Comparison range 1: >I1 MW60
Comparison range 2: >I2 MW64
Number of bytes: >NO 12

→ The comparison is in byte format. MW60 to MW64 is 8 bytes. The number of bytes is 12 bytes.

The error message “Comparison ranges overlap” is output.

BC01E2 is 1.

Example:
Comparison of marker blocks, invalid offset error.

Two marker blocks are to be compared. Block 1 starts at MW40, Block 2 at MW54. The block length is specified by the value of the counter C 01QV.

Parameters of BC01 function block:
Comparison range 1: >I1 MW40
Comparison range 2: >I2 MW54
Number of bytes: >NO C 01QV

→ The value of C 01QV is 1024. This value is too big. The value at >NO can be between 1 and +383.

The message "The specified offset of the comparison ranges is outside of the permissible range" is output.

BC01E3 is 1.

Data block transfer

MFD-Titan is provided with 32 function blocks BT01 to BT32 for transferring values from one marker range (Copy data). The marker ranges can be overwritten with a particular value (data initialisation). The following marker types can be transferred and overwritten:

- MB,
- MW,
- MD.

The function block is enabled in the circuit diagram.

```
I 05--------------------------[ BT07T_
BT07E1┐
BT07E2┤
BT07E3┴-----------------------[ M 42
```

Figure 99: MFD-Titan circuit diagram with enabling of transfer data block function block

```
BT07 INI      +
 >I1
 >I2
 >NO
```

Parameter display and parameter set for a data block transfer function block:

BT07	Data block transfer function block number 07
INI	INI mode, initialise marker ranges
+	Appears in the parameter display
>I1	Source range start
>I2	Destination range start
>NO	Number of elements to be written in bytes per range. Value range 1 to + 383

Only constants can be modified in the parameter display of a function block.

Operating modes of the transfer data block function block

Parameter	Function
INI	Initialise marker ranges
CPY	Copy marker ranges

Inputs

The function block inputs >I1, >I2 and >NO can have the following operands:

- Constants
- Markers MD, MW, MB
- Analog inputs IA01 to IA04
  - IA01: terminal I7
  - IA02: terminal I8
  - IA03: terminal I11
  - IA04: terminal I12
- Analog output QA01
- Actual value … QV> of another function block

Specifying the marker range without offset

If MB, MW or MD markers are specified both at >I1 and at >I2, the number of markers defines the source or destination range.

Specifying a marker range with offset

If you wish to work with an offset, specify one of the following variables at function block input >I1 or >I2

- Constant,
- Actual value ..QV of a function block,
- Analog input IA..,
- Analog output QA..

The value at the input is taken as the offset to marker MB01.

Displaying the parameter set in the PARAMETERS menu

- + Access enabled
- –: Access disabled

Contacts

BT01E1 to BT32E1: the number of marker bytes exceeds the source or destination range.

BT01E2 to BT32E2: source and destination range overlap. Only valid for CPY mode, copy marker ranges.

BT01E3 to BT32E3: the specified offset is invalid.

Coils

BT01T_ to BT32T_: trigger coil of the transfer data block function block.

Memory requirement of the transfer data block function block

The transfer data block function block requires 48 bytes of memory plus 4 bytes per constant at the function block inputs.

Function of the transfer data block function block

The transfer data block comparator function block has two operating modes.

→ No data blocks are initialised or copied if an error occurs.

Initalising INI marker ranges

There is one source range and one destination range. The source range is specified at >I1. The length of the source range is one byte. The destination range is specified at >I2. The length of the destination range is specified by the number of bytes at the >NO input.

The content of the source range is transferred to the marker bytes in the destination range.

The function block executes the transfer if there is a rising edge from 0 to 1 at the BT..T_ (Trigger) coil.

The error outputs E1, E2 and E3 are evaluated regardless of the status of the trigger.

Example:
Initialising marker blocks, specifying marker ranges directly

The value of marker byte 10 is to be transferred to marker bytes 20 to 29.

Parameters of BT01 function block:
Source range: >I1 MB10
Destination range: >I2 MB20
Number of bytes: >NO 10

Source range	Value of source marker range (decimal)	Destination range	Value of destination marker range (decimal)
MB10	123	MB20	123
		MB21	123
		MB22	123
		MB23	123
		MB24	123
		MB25	123
		MB26	123
		MB27	123
		MB28	123
		MB29	123

After a rising edge from 0 to 1 at coil BT01T_ the value 123 is present in the marker bytes MB20 to MB29.

Example:
Initialisation of marker blocks, definition of a range with offset

The content of marker byte MB15 is to be transferred to marker bytes MB65 to MB68.

Parameters of BT01 function block:
Source range: >I1 MB15
Destination range: >I2 64
Number of bytes: >NO 4
Marker MB01: 1

→ Destination range: Constant 64:
Marker MB01 plus Offset: 1 + 64 = 65 → MB65.

Source range	Value of source marker range (decimal)	Destination range	Value of destination marker range (decimal)
MB15	45	MB65	45
		MB66	45
		MB67	45
		MB68	45

After a rising edge from 0 to 1 at coil BT01T_ the value 45 is present in the marker bytes MB65 to MB68.

Example:
Initialisation of marker blocks, definition of a range in a different format.

The value of marker byte MB60 is to be transferred to MD80 and MD81.

Parameters of BT01 function block:
Source range: >I1 MB60
Destination range: >I2 MD80
Number of bytes: >NO 8

→ The transfer is in byte format. MD80 has 4 bytes and MD81 has 4 bytes, which means that :NO has the value 8.

Comparison range 1	Value of marker range 1 (decimal/ binary)	Comparison range 2	Value of marker range 2 (decimal/binary)
MB60	45/ 00101101	MD80 (Byte 1, LSB)	757935405/ 00101101 00101101 **00101101** 00101101
		MD80 (Byte 2)	757935405/ 00101101 00101101 **00101101** 00101101
		MD80 (Byte 3)	757935405/ 00101101 00101101 **00101101** 00101101
		MD80 (Byte 4, MSB)	757935405/ **00101101** 00101101 00101101 00101101
		MD81 (Byte 1, LSB)	757935405/ 00101101 00101101 **00101101** 00101101
		MD81 (Byte 2)	757935405/ 00101101 00101101 **00101101** 00101101
		MD81 (Byte 3)	757935405/ 00101100 **01011011** 00101101 00101101
		MD81 (Byte 4, MSB)	757935405/ **00101101** 00101101 00101101 00101101

After a rising edge from 0 to 1 at coil BT01T_ the value 757935405 is present in the marker double words MD80 and MD81.

Example:
Transfer of marker byte, range violation error.

The value of marker byte MB96 is to be transferred to MD93, MD94, MD95 and MD96. The length is 16 bytes.

Parameters of BT01 function block:
Source range: >I1 MD96
Destination range: >I2 MD93
Number of bytes: >NO 16

→ The transfer is in byte format. MD93 to MD96 is 16 bytes. 18 bytes were incorrectly defined as length.

The error message “Number of elements exceeds the destination range” is output.

BT01E1 is 1.

Example:
Transfer of marker bytes, invalid offset error.

The value of marker byte MB40 is to be transferred to MW54 and subsequent marker words. The block length is specified by the value of the counter C 01QV.

Parameters of BC01 function block:
Comparison range 1: >I1 MB40
Comparison range 2: >I2 MW54
Number of bytes: >NO C 01QV

→ The value of C 01QV is 788. This value is too big. The value at >NO can be between 1 and +383.

The message "The specified offset of the destination range is outside of the permissible range" is output.

BT01E3 is 1.

CPY mode, copy marker ranges

There is one source range and one destination range. The source range is specified at >I1. The destination range is specified at >I2. The length of the source and destination range is specified by the number of bytes at the >NO input.

The content of the source range is copied to the marker bytes in the destination range.

The function block executes the copy operation if there is a rising edge from 0 to 1 at the BT..T_ (Trigger) coil.

The error outputs E1, E2 and E3 are evaluated regardless of the status of the trigger.

Example:
Copy of marker blocks, definition of marker ranges direct

The content of marker bytes 10 to 19 is to be transferred to marker bytes 20 to 29.

Parameters of BT01 function block:
Source range: >I1 MB10
Destination range:>I2 MB20
Number of bytes: >NO 10

Source range	Value of source marker range (decimal)	Destination range	Value of destination marker range (decimal)
MB10	42	MB20	42
MB11	27	MB21	27
MB12	179	MB22	179
MB13	205	MB23	205
MB14	253	MB24	253
MB15	17	MB25	17
MB16	4	MB26	4
MB17	47	MB27	47
MB18	11	MB28	11
MB19	193	MB29	193

After a rising edge from 0 to 1 at coil BT01T_ the content of MB10 to MB19 is copied to the marker bytes MB20 to MB29.

Example:
Copying of marker blocks, definition of a marker range with offset

The content of marker bytes MB15 to MB18 is to be copied to marker bytes MB65 to MB68.

Parameters of BT01 function block:
Source range: >I1 MB15
Destination range: >I2 64
Number of bytes: >NO 4
Marker MB01: 1

→ Destination range: Constant 64:
Marker MB01 plus Offset: 1 + 64 = 65 → MB65.

Source range	Value of source marker range (decimal)	Destination range	Value of destination marker range (decimal)
MB15	68	MB65	68
MB16	189	MB66	189
MB17	203	MB67	203
MB18	3	MB68	3

After a rising edge from 0 to 1 at coil BT01T_ the content of MB15 to MB18 is copied to the marker bytes MB65 to MB68.

Example:
Copying of marker blocks, definition of a marker range in a different format.

The value of marker byte MD60 to MD62 is to be copied to MW40 to MW45.

Parameters of BT01 function block:
Source range: >I1 MD60
Destination range: >I2 MW40
Number of bytes: >NO 12

→ The transfer is in byte format. 12 bytes are to be copied. The range MD60 to MD62 is 12 bytes. This is copied to the range MW40 to MW45.

Comparison range 1	Value of marker range 1 (decimal/binary)	Comparison range 2	Value of marker range 2 (decimal/binary)
MD60	866143319/ 0011001110100000 **0100110001010111**	MW40 (LSW)	19543/ 0011001110100000 **0100110001010111**
MD60	866143319/ **0011001110100000** 0100110001010111	MW41 (MSW)	13216/ **0011001110100000** 0100110001010111
MD61	173304101/ 0000101001010100 **0110100100100101**	MW42 (LSW)	26917/ 0000101001010100 **0110100100100101**
MD61	173304101/ **0000101001010100** 0110100100100101	MB43 (MSW)	2644/ **0000101001010100** 0110100100100101
MD62	982644150/ 0011101010010001 **1111010110110110**	MB44 (LSW)	62902/ 0011101010010001 **1111010110110110**
MD62	982644150/ **0011101010010001** 1111010110110110	MB45 (MSW)	14993/ **0011101010010001** 1111010110110110

After a rising edge from 0 to 1 at coil BT01T_ the values are copied to the appropriate range.

Example:
Copying of marker bytes, destination range violation error.

The value of marker bytes MB81 to MB96 is to be transferred to MD93, MD94, MD95 and MD96. The length is 16 bytes.

Parameters of BT01 function block:
Source range: >I1 MB81
Destination range: >I2 MD93
Number of bytes: >NO 18

→ The transfer is in byte format. MD93 to MD96 is 16 bytes. 18 bytes were incorrectly defined as length.

The error message “Number of elements exceeds the destination range” is output.

BT01E1 is 1.

Example
Comparison of marker blocks, range overlap error.

12 bytes are to be copied starting from MW60. MW64 is specified as destination address.

Parameters of BT01 function block:
Comparison range 1:>I1 MW60
Comparison range 2:>I2 MW64
Number of bytes: >NO 12

→ The copy operation is in byte format. MW60 to MW64 is 8 bytes. The number of bytes is 12 bytes.

The error message “Both ranges overlap” is output.

BC01E2 is 1.

Example:
Copying of marker bytes, invalid offset error.

The value of marker word MW40 is to be copied to MW54 and subsequent marker words. The block length is specified by the value of the counter C 01QV.

Parameters of BT01 function block:
Comparison range 1:>I1 MW40
Comparison range 2:>I2 MW54
Number of bytes: >NO C 01QV

→ The value of C 01QV is 10042. This value is too big. The value at >NO can be between 1 and +383.

The message "The specified offset of the destination range is outside of the permissible range" is output.

BT01E3 is 1.

Boolean operation

MFD-Titan provides 32 function blocks from BV01 to BV32 for Boolean operations with values.

The following possibilities are provided by the Boolean operation function block:

- Screening out of particular bits from values,
- Bit pattern recognition,
- Bit pattern modification.

A Boolean operation function block is not wired in the circuit diagram.

```
BV27 AND     +
 >I1
 >I2
  QV>
```

Parameter display and parameter set for Boolean operation function block:

BV27	Boolean operation function block number 27
AND	AND operation mode
+	Appears in the parameter display
>I1	First value
>I2	Second value
QV>	Result of the operation

Only constants can be modified in the parameter display of a function block.

Operating modes of the Boolean operation function block

Parameter	Function
AND	AND operation
OR	OR operation
XOR	Exclusive OR operation
NOT	Negation of the Boolean value of >I1

Value range

32 bit signed value

Inputs

The function block inputs >I1 and >I2 can have the following operands:

- Constants
- Markers MD, MW, MB
- Analog inputs IA01 to IA04
  - IA01: terminal I7
  - IA02: terminal I8
  - IA03: terminal I11
  - IA04: terminal I12
- Analog output QA01
- Actual value … QV> of another function block

Actual value …QV>

The actual value …QV> can be assigned the following operands:

- Markers MD, MW, MB
- Analog output QA01

Displaying the parameter set in the PARAMETERS menu

- + Access enabled
- –: Access disabled

Contacts

BV01ZE to BV32ZE: ZERO zero bit, value on output function block is equal to zero

Coils

The Boolean operation function block does not have coils.

Memory requirement Boolean operation function block

The Boolean operation function block requires 40 bytes of memory plus 4 bytes per constant on the function block inputs.

Function of Boolean operation function block

The function block creates the operation depending on the operating mode.

> If you program a negative value, e.g.: -10_{dec}, the CPU will form the two's complement of the amount.
>
> Example:
> -10_{dec} = $1000\,0000\,0000\,0000\,0000\,0000\,0000\,1010_{bin}$
>
> Two's complement =
> $1111\,1111\,1111\,1111\,1111\,1111\,1111\,0110_{bin}$ =
> $FFFFFFF6_{hex}$
>
> Bit 32 is the signed bit and remains as 1.

AND Boolean operation

Value >I1: $13219_{dec} = 0011\,0011\,1010\,0011_{bin}$

Value >I2: $57193_{dec} = 1101\,1111\,0110\,1001_{bin}$

Result QV>: $4897_{dec} = 0001\,0011\,0010\,0001_{bin}$

OR Boolean operation

Value >I1: $13219_{dec} = 0011\,0011\,1010\,0011_{bin}$

Value >I2: $57193_{dec} = 1101\,1111\,0110\,1001_{bin}$

Result QV>: $65515_{dec} = 1111\,1111\,1110\,1011_{bin}$

XOR Boolean operation

Value >I1: $13219_{dec} = 0011\,0011\,1010\,0011_{bin}$

Value >I2: $57193_{dec} = 1101\,1111\,0110\,1001_{bin}$

Result QV>: $60618_{dec} = 1110\,1100\,1100\,1010_{bin}$

NOT Boolean operation

Value >I1: 13219$_{dec}$ = 0000 0000 0000 0000 0011 0011 1010 0011$_{bin}$

Value >I2: Omitted

Result QV>: –13220$_{dec}$ = 1111 1111 1111 1111 1100 1100 0101 1100$_{bin}$

The NOT operation operates according to the following rules:

>I1, positive value
Negate value of >I1 and subtract 1:
–|>I1| – 1 = >I2

>I1, Negative value
Value of >I1 and subtract 1:
|>I1| – 1 = >I2

Counters

MFD-Titan provides 32 up/down counters from C 01 to C 32. The counter relays allow you to count events. You can enter upper and lower threshold values as comparison values. The contacts will switch according to the actual value. To specify a start value, for example, counting from the value 1 200, this can be implemented using a “C ..” counter.

The “C..” counters are cycle time dependent.

Wiring of a counter

You integrate a counter into your circuit in the form of a contact and coil. The counter relay has different coils and contacts.

→ To prevent unpredictable switching states, use each coil of a relay once only in the circuit diagram.

```
I 05----------------------------[ C 20C_
I 06----------------------------[ C 20RE
I 07----------------------------[ C 20D_
I 08----------------------------[ C 20SE
C 20OF--------------------------[ Q 01
C 20FB--------------------------[ Q 02
C 20ZE--------------------------[ Q 03
C 20CY--------------------------S M 42
```

Figure 100: MFD-Titan circuit diagram with counter relay

```
C 20        +
 >SH
 >SL
 >SV
  QV>
```

Parameter display and parameter set for the counter relay:

C 20	Counter relay function block number 20
+	Appears in the parameter display
>SH	Upper setpoint
>SL	Lower setpoint
>SV	Defined actual value (Preset)
QV>	Actual value in RUN mode

In the parameter display of a counter relay you change setpoint values and/or the preset value and the enabling of the parameter display.

Value range

The function block operates in the integer range from –2147483648 to 2147483647.

Behaviour when value range is exceeded

The function block sets the switching contact C .. CY to the status 1 and retains the value of the last valid operation.

→ The counter C counts every rising edge on the counter input. If the value range is exceeded, the switching contact C ..CY switches to status 1 for one cycle per rising edge detected.

Inputs
The function block inputs >SH, >SL and >SV can have the following operands:

- Constants
- Markers MD, MW, MB
- Analog inputs IA01 to IA04
  - IA01: terminal I7
  - IA02: terminal I8
  - IA03: terminal I11
  - IA04: terminal I12
- Analog output QA01
- Actual value … QV> of another function block

Actual value …QV>
The actual value …QV> can be assigned the following operands:

- Markers MD, MW, MB
- Analog output QA01

Displaying the parameter set in the PARAMETERS menu
- + Access enabled
- – Access disabled

Contacts
- C 01OF to C 32OF: Actual value ≧ upper setpoint
- C 01FB to C 32FB: Actual value ≦ Lower setpoint value
- C 01ZE to C 32ZE: Actual value = zero
- C 01CY to C 32CY: Value range exceeded

Coils
- C 01C_ to C 32C_: counter coils, count with rising edge
- C 01D_ to C 32D_: count direction definition, status 0 = count upwards, status 1 = count downwards
- C 01RE to C 32RE: Reset actual value to zero
- C 01SE to C 32SE: accept defined actual value with rising edge.

Memory requirement of the counter relay

The counter relay function block requires 52 bytes of memory plus 4 bytes per constant on the function block inputs.

Retention

Counter relays can be operated with retentive actual values. The number of retentive counter relays can be selected in the SYSTEM → RETENTION menu.

The retentive actual value requires 4 bytes of memory.

If a counter relay is retentive, the actual value is retained when the operating mode changes from RUN to STOP as well as when the power supply is switched off.

If MFD is started in RUN mode, the counter relay operates with the retentively saved actual value.

Function of the counter function block

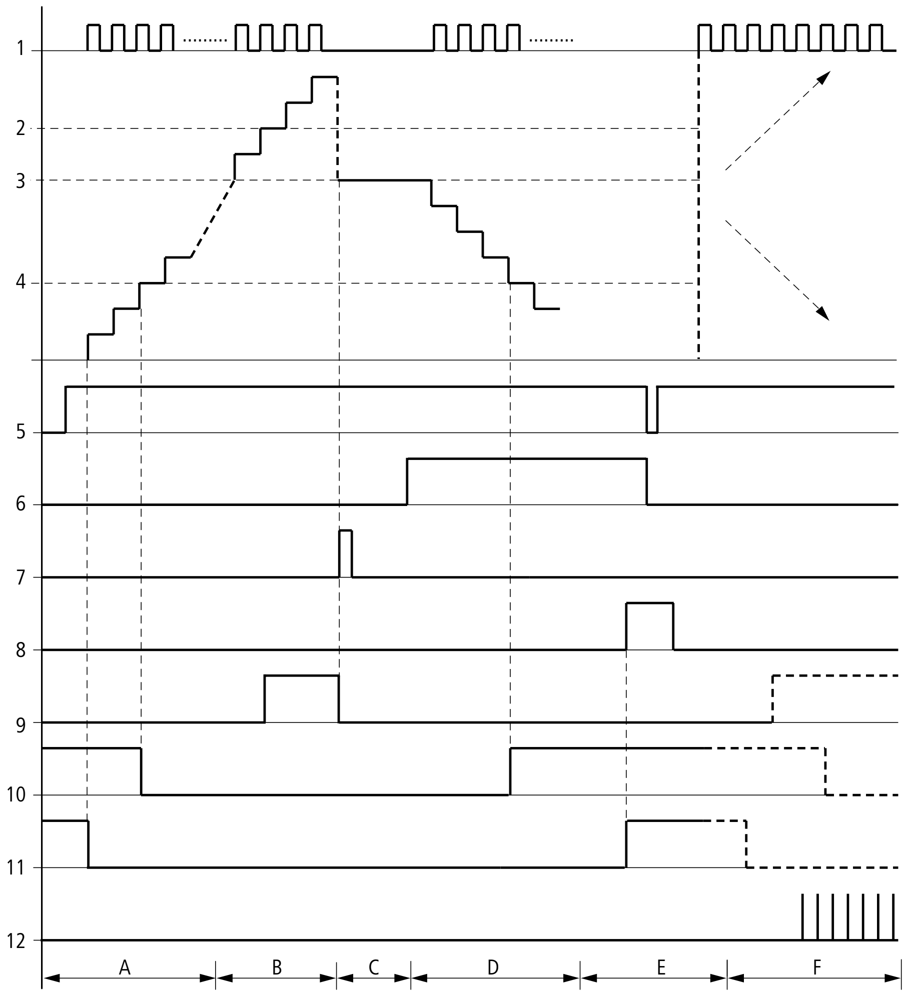


Figure 101: Signal diagram of counter

1: counter coil C..C_

2: upper setpoint value >SH

3: preset actual value >SV

4: lower setpoint value >SL

5: counting direction, coil C..D_

6: accept preset actual value, coil C..SE

7: reset coil C..RE

8: contact (make contact) C..OF upper setpoint value reached, exceeded

9: contact (make contact) C..FB lower setpoint value reached, undershot

10: actual value equal to zero

11: out of value range

- Range A:
  - The counter has the value zero.
  - The contacts C..ZE (actual value equal to zero) and C..FB (lower setpoint value undershot) are active.
  - The counter receives counter values and increases the actual value.
  - C..ZE drops out as well as C..FB and also when the lower setpoint value is reached.
- Range B:
  - The counter counts upwards and reaches the upper setpoint value. The “upper setpoint value reached” contact C..OF becomes active.
- Range C:
  - The coil C..SE is briefly actuated and the actual value is set to the preset actual value. The contacts go to the respective position.
- Range D:
  - The counting direction coil C..D_ is actuated. If counting pulses are present, downward count is initiated.
  - If the lower setpoint value is undershot, the contact C..FB becomes active.
- Range E:
  - The reset coil C..RE is activated. The actual value is set to zero.
  - The contact C..ZE is active.
- Range F:
  - The actual value goes outside the value range of the counter.
  - The contacts become active according to the direction of the values (positive or negative).

High-speed counters

MFD-Titan provides various high-speed counter functions. These counter function blocks are coupled directly to the digital inputs. The high-speed counter functions are only available with MFD-DC inputs.

The following functions are possible:

- Frequency counters, measure frequencies **CF..**
- High-speed counters, count high-speed signals **CH..**
- Incremental encoder counters, count two-channel incremental encoder signals **CI..**

The high-speed digital inputs are I1 to I4.

The following wiring rules apply:

- I1: CF01 or CH01 or CI01
- I2: CF02 or CH02 or CI01
- I3: CF03 or CH03 or CI02
- I4: CF04 or CH04 or CI02

Attention!
Every digital input I .. may only be used once by the CF, CH, CI function blocks.

The incremental encoder requires an input pair.

Example:

- I1: high-speed counter CH01
- I2: frequency counter CF02
- I3: incremental encoder channel A CI02
- I4: incremental encoder channel B CI02

Example: function block list in the FUNCTION RELAYS menu:

CI01
CF01
CH01

All function blocks access digital input I1.

Only CH01 supplies the correct value.

Frequency counters

MFD-Titan provides four frequency counters which are CF01 to CF04. The frequency counters can be used for measuring frequencies. You can enter upper and lower threshold values as comparison values. The high-speed frequency counters are hardwired to the digital inputs I1 to I4.

The CF.. frequency counters operate independently of the cycle time.

Counter frequency and pulse shape

The maximum counter frequency is 3 kHz.

The minimum counter frequency is 4 Hz.

The signals must be square waves. The mark-to-space ratio is 1:1.

Measurement method

The pulses on the input are counted for one second irrespective of the cycle time, and the frequency is determined. The result of the measurement is made available as a value to the function block output CF..QV.

Wiring of a counter

The following assignment of the digital inputs apply.

- I1 counter input for the counter CF01
- I2 counter input for the counter CF02
- I3 counter input for the counter CF03
- I4 counter input for the counter CF04

→ To prevent unpredictable switching states, use each coil of a relay once only in the circuit diagram. Use a counter input for the CF, CH, CI counters only once.

Wiring of a frequency counter

You integrate a frequency counter into your circuit in the form of a contact and coil. The counter relay has different coils and contacts.

```
I 05---------------------------[ CF01EN
CF01OF-------------------------[ Q 01
CF01FB-------------------------[ Q 02
CF01ZE-------------------------[ Q 03
```

Figure 102: MFD-Titan circuit diagram with frequency counter

```
CF01          -
 >SH
 >SL
  QV>
```

Parameter display and parameter set for frequency counter:

CF01	Frequency counter function block number 01
-	Does not appear in the parameter display
>SH	Upper setpoint
>SL	Lower setpoint
QV>	Actual value in RUN mode

In the parameter display of a counter relay you change setpoint values and/or the preset value and the enable of the parameter display.

Value range

The function block operates in the integer range from
0 to 5000
1 kHz 1 = 1000

Behaviour when value range is exceeded

The value range cannot be exceeded as the maximum measured value is less than the value range.

Inputs

The function block inputs >SH and >SL can have the following operands:

- Constants
- Markers MD, MW, MB
- Analog inputs IA01 to IA04
  - IA01: terminal I7
  - IA02: terminal I8
  - IA03: terminal I11
  - IA04: terminal I12
- Analog output QA01
- Actual value … QV> of another function block

Actual value …QV>

The actual value …QV> can be assigned the following operands:

- Markers MD, MW, MB
- Analog output QA01

Displaying the parameter set in the PARAMETERS menu

- + Access enabled
- – Access disabled

Contacts

- CF01OF to CF04OF: Actual value ≧ Upper setpoint
- CF01FB to CF04FB: Actual value ≦ Lower setpoint
- CF01ZE to CF04ZE: Actual value = Zero

Coils

CF01EN to CF04EN: enable of the counter with coil status = 1.

Memory requirement of the frequency counter

The frequency counter function block requires 40 bytes of memory plus 4 bytes per constant on the function block inputs.

Retention

The frequency counter does not retain actual values, as the frequency is remeasured continuously.

Function of the frequency counter function block

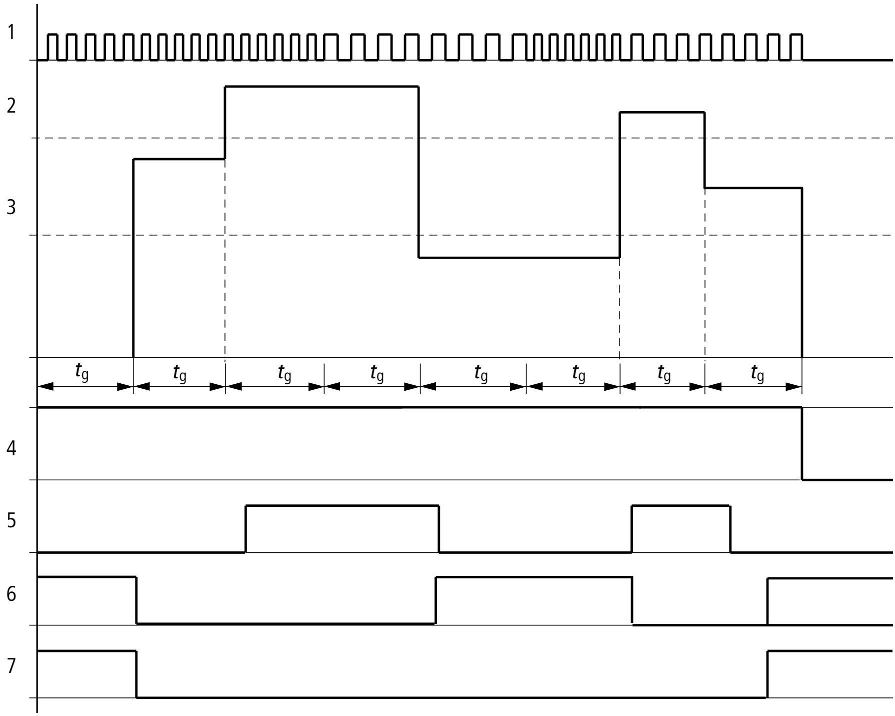


Figure 103: Signal diagram of frequency counter

1: counter input I1 to I4

2: upper setpoint value >SH

3: lower setpoint value >SL

4: enable CF..EN

5: contact (make contact) CF..OF upper setpoint value exceeded

6: contact (make contact) CF..FB lower setpoint value undershot

7: actual value equal to zero CF..ZE

t_g: gate time for the frequency measurement

- The first measurements are made after the CF..EN enable signal has been activated. The value is output after the gate time has timed out.
- The contacts are set in accordance with the measured frequency.
- If the CF..EN enable signal is removed, the output value is set to zero.

High-speed counters

MFD-Titan provides four high-speed up/down counters CH01 to CH04 for use. The high-speed frequency counters are hardwired to the digital inputs I1 to I4. These counter relays allow you to count events independently of the cycle time. You can enter upper and lower threshold values as comparison values. The contacts will switch according to the actual value. To specify a start value, for example, counting from the value 1989, this can be implemented using a CH .. counter.

The CH.. counters operate independently of the cycle time.

Counter frequency and pulse shape

The maximum counter frequency is 3 kHz.

The signals must be square waves. The mark-to-space ratio is 1:1.

Wiring of a counter

The following assignment of the digital inputs apply.

- I1 counter input for the counter CH01
- I2 counter input for the counter CH02
- I3 counter input for the counter CH03
- I4 counter input for the counter CH04

→ To prevent unpredictable switching states, use each coil of a relay once only in the circuit diagram. Use a counter input for the CF, CH, CI counters only once.

You integrate a counter into your circuit in the form of a contact and coil. The counter relay has different coils and contacts.

```
I 05---------------------------[ CH01EN
I 06---------------------------[ CH01RE
I 07---------------------------[ CH01D_
I 08---------------------------[ CH01SE
CH01OF-------------------------[ Q 01
CH01FB-------------------------[ Q 02
CH01ZE-------------------------[ Q 03
CH01CY-------------------------S M 94
```

Figure 104: MFD-Titan circuit diagram with high-speed counter

```
CH01          +
 >SH
 >SL
 >SV
  QV>
```

Parameter display and parameter set for high-speed counters:

CH01	High-speed counter function block number 01
+	Appears in the parameter display
>SH	Upper setpoint
>SL	Lower setpoint
>SV	Defined actual value (Preset)
QV>	Actual value in RUN mode

In the parameter display of a counter relay you change setpoint values and/or the preset value and the enable of the parameter display.

Value range

The function block operates in the integer range from –2 147 483 648 to 2 147 483 647.

Behaviour when value range is exceeded

- The function block sets the switching contact CH..CY to status 1.
- The function block retains the value of the last valid operation.

→ Counter CH counts every rising edge on the counter input. If the value range is exceeded, the switching contact CH ..CY switches to status 1 for one cycle per rising edge detected.

Inputs

The function block inputs >SH, >SL and >SV can have the following operands:

- Constants
- Markers MD, MW, MB
- Analog inputs IA01 to IA04
  - IA01: terminal I7
  - IA02: terminal I8
  - IA03: terminal I11
  - IA04: terminal I12
- Analog output QA01
- Actual value … QV> of another function block

Actual value ..QV>

The actual value …QV> can be assigned the following operands:

- Markers MD, MW, MB
- Analog output QA01

→ The actual value is only cleared in RUN mode with a specific reset signal.

Displaying the parameter set in the PARAMETERS menu

- + Access enabled
- – Access disabled

Contacts

- CH01OF to CH04OF: Actual value ≧ Upper setpoint
- CH01FB to CH04FB: Actual value ≦ Lower setpoint
- CH01ZE to CH04ZE: Actual value = Zero
- CH01CY to CH04CY: Value range exceeded

Coils

- CH01EN to CH04EN: enable of the counter
- CH01D to CH04D: count direction definition,
  Status 0 = count upwards,
  Status 1 = count downwards
- CH01RE to CH04RE: reset actual value to zero
- CH01SE to CH04SE: accept preset actual value with rising edge.

Memory requirement of the high-speed counter

The high-speed counter function block requires 52 bytes of memory plus 4 bytes per constant on the function block inputs.

Retention

High-speed counter relays can be operated with retentive actual values. The number of retentive counter relays can be selected in the SYSTEM → RETENTION menu.

If a counter relay is retentive, the actual value is retained when the operating mode changes from RUN to STOP as well as when the power supply is switched off.

If MFD is started in RUN mode, the counter relay operates with the retentively saved actual value.

Function of the high-speed counter function block

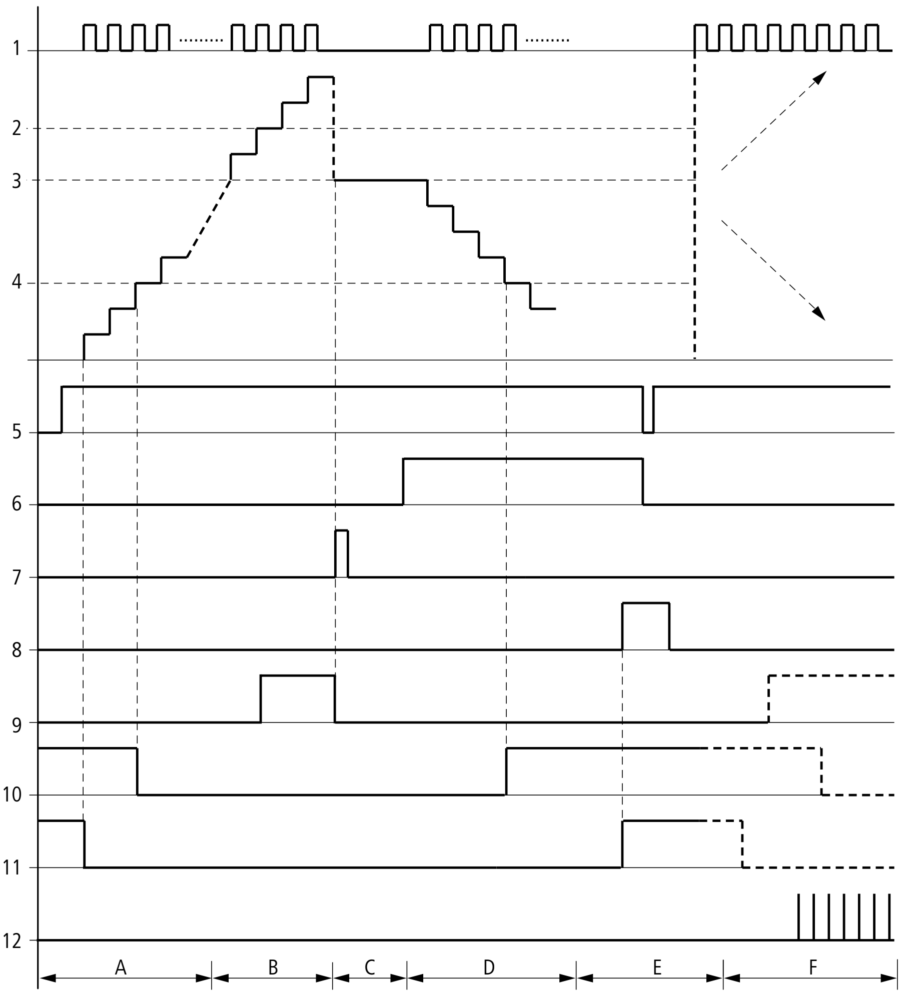


Figure 105: Signal diagram of high-speed counter

1: counter input I1 to I4

2: upper setpoint value >SH

3: preset actual value >SV

4: lower setpoint value >SL

5: enable of the counter CH..EN

6: counting direction, coil CH..D

7: accept preset actual value, coil CH..SE

8: reset coil CH..RE

9: contact (make contact) CH..OF upper setpoint value reached, exceeded

10: contact (make contact) CH..FB lower setpoint value reached, undershot

11: contact (make contact) CH..ZE actual value equal to zero

12:out of value range

- Range A:
  - The counter has the value zero.
  - The contacts CH..ZE (actual value equal to zero) and CH..FB (lower setpoint value undershot) are active.
  - The counter receives counter values and increases the actual value.
  - CH..ZE drops out as well as CH..FB after the lower setpoint value is reached.
- Range B:
  - The counter counts upwards and reaches the upper setpoint value. The contact "upper setpoint value" CH..OF becomes active.
- Range C:
  - The coil CH..SE is briefly actuated and the actual value is set to the preset actual value. The contacts go to the respective position.
- Range D:
  - The counting direction coil CH..D is actuated. If counting pulses are present, downward count is initiated.
  - If the lower setpoint value is undershot, the contact CH..FB becomes active.
- Range E:
  - The reset coil CH..RE is activated. The actual value is set to zero.
  - The contact CH..ZE is active.
- Range F:
  - The actual value goes outside the value range of the counter.
  - The contacts become active according to the direction of the values (positive or negative).

High-speed incremental encoder counters

MFD-Titan provides two high-speed incremental encoder counters CI01 and CI02. The high-speed counter inputs are hardwired to the digital inputs I1, I2, I3 and I4. These counter relays allow you to count events independently of the cycle time. You can enter upper and lower threshold values as comparison values. The contacts will switch according to the actual value. You can use a CI.. counter if you wish to define a start value.

The CI.. counters operate independently of the cycle time.

Counter frequency and pulse shape
The maximum counter frequency is 3 kHz.

The signals must be square waves. The mark-to-space ratio is 1:1. The signals on channels A and B must lead or lag by 90°. Otherwise the counting direction cannot be determined.

→ Double the number of pulses are counted as a result of the internal method of operation of the incremental encoder. The incremental encoder evaluates the rising and falling edges. This ensures that the pulse count is not affected by oscillation of a signal edge. If the number of pulses are required, divide the value by two.

Wiring of a counter
The following assignment of the digital inputs apply:

- I1 counter input for the counter CI01 channel A
- I2 counter input for the counter CI01 channel B
- I3 counter input for the counter CI02 channel A
- I4 counter input for the counter CI02 channel B

→ To prevent unpredictable switching states, use each coil of a relay once only in the circuit diagram.

Use a counter input for the CF, CH, CI counters only once.

You integrate a counter into your circuit in the form of a contact and coil. The counter relay has different coils and contacts.

```
I 05---------------------------Ä CI01EN
I 06---------------------------Ä CI01RE
I 08---------------------------Ä CI01SE
CI01OF-------------------------Ä Q 01
CI01FB-------------------------Ä Q 02
CI01ZE-------------------------Ä Q 03
CI01CY-------------------------S M 94
```

Figure 106: MFD-Titan circuit diagram with high-speed incremental encoder counter

```
CI01          +
 >SH
 >SL
 >SV
  QV>
```

Parameter display and parameter set for high-speed incremental encoder counter:

CI01	High-speed incremental encoder counter function block number 01
+	Appears in the parameter display
>SH	Upper setpoint
>SL	Lower setpoint
>SV	Defined actual value (Preset)
QV>	Actual value in RUN mode

In the parameter display of a counter relay you change setpoint values and/or the preset value and the enable of the parameter display.

Value range
The function block operates in the integer range from –2 147 483 648 to 2 147 483 647.

Each pulse is counted twice.

Example: value at CI..QV>= 42 000

The counter has counted 21000 pulses.

Behaviour when value range is exceeded

- The function block sets the switching contact CI..CY to status 1.
- The function block retains the value of the last valid operation.

→ Counter CI counts every rising edge on the counter input. If the value range is exceeded, the switching contact CI ..CY switches to status 1 for one cycle per rising edge detected.

Inputs

The function block inputs >SH, >SL and >SV can have the following operands:

- Constants
- Markers MD, MW, MB
- Analog inputs IA01 to IA04
  - IA01: terminal I7
  - IA02: terminal I8
  - IA03: terminal I11
  - IA04: terminal I12
- Analog output QA01
- Actual value … QV> of another function block

Actual value ..QV>

The actual value …QV> can be assigned the following operands:

- Markers MD, MW, MB
- Analog output QA01

→ The actual value is only erased in RUN mode with a selective reset signal.

Displaying the parameter set in the PARAMETERS menu

- + Access enabled
- – Access disabled

Contacts

- CI01OF to CI02OF: Actual value≧ Upper setpoint
- CI01FB to CI02FB: Actual value ≦ Lower setpoint
- CI01ZE to CI02ZE: Actual value = Zero
- CI01CY to CI02CY: Value range exceeded

Coils

- CI01EN to CI02EN: Counter enable
- CI01RE to CI02RE: Reset actual value to zero
- CI01SE to CI02SE: Accept preset actual value with rising edge.

Memory requirement of the counter relay

The high-speed counter function block requires 52 bytes of memory plus 4 bytes per constant on the function block inputs.

Retention

High-speed counter relays can be operated with retentive actual values. The number of retentive counter relays can be selected in the SYSTEM → RETENTION menu.

If a counter relay is retentive, the actual value is retained when the operating mode changes from RUN to STOP as well as when the power supply is switched off.

If MFD is started in RUN mode, the counter relay operates with the retentively saved actual value.

Function of the high-speed incremental encoder counter function block

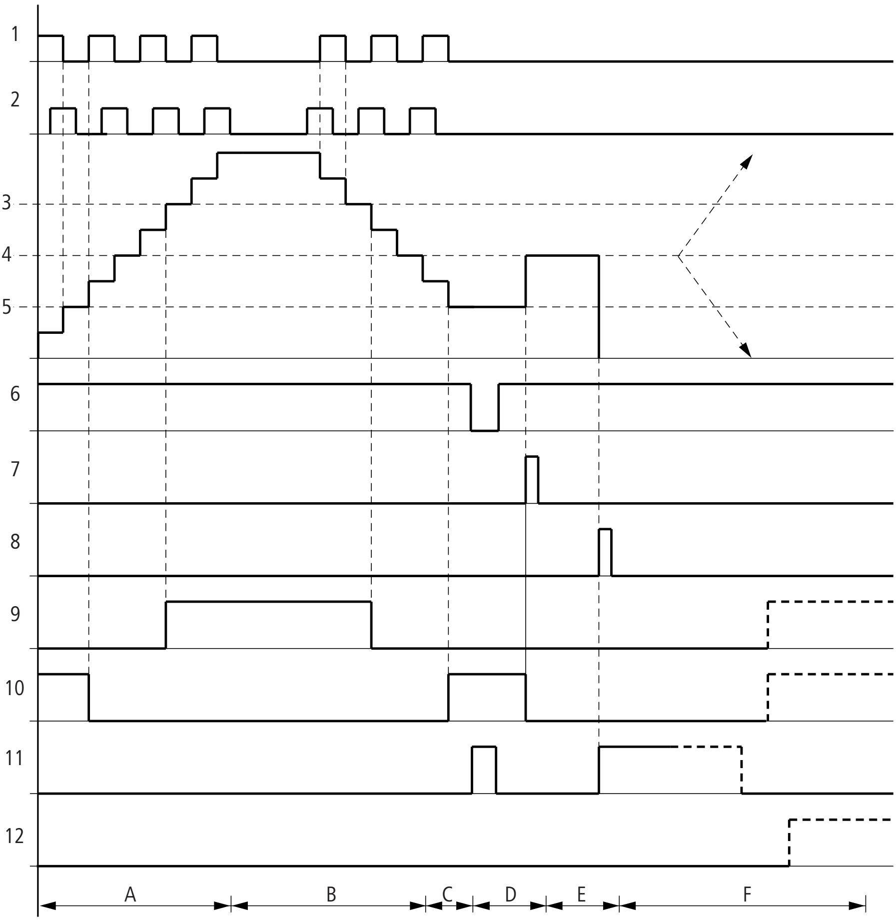


Figure 107: Signal diagram of high-speed incremental encoder counter

1: counter input channel A

2: counter input channel B

3: upper setpoint value >SH

4: preset actual value >SV

5: lower setpoint value >SL

6: counter enable

7: accept preset actual value, coil CI..EN

8: reset coil CI..RE

9: contact (make contact) CI..OF upper setpoint value reached, exceeded

10: contact (make contact) CI..FB lower setpoint value reached, undershot

11: contact (make contact) CI..ZE actual value equal to zero

12: contact (make contact) CI..CY value range exceeded or undershot

- Range A:
  - The counter counts upwards.
  - The value leaves the lower threshold value and reaches the upper value.
- Range B:
  - The count direction changes to a downward count.
  - The contacts switch in accordance with the actual value.
- Range C:
  - The enable signal is set to 0. The actual value becomes 0.
- Range D:
  - The rising edge on the accept preset value coil sets the actual value to the preset value.
- Range E:
  - The reset pulse sets the actual value to zero.
- Range F:
  - The actual value goes outside the value range of the counter.
  - The contacts become active according to the direction of the values (positive or negative).

Comparators

Comparator function blocks allow you to compare constants and variables with one another.

The following comparisons are possible:

Function block input		Function block input
>I1	Greater than	>I2
	Equal to	
	Less than	

```
CP32LT--------------------------S Q 01
CP32EQ--------------------------S Q 02
CP32GT-------------------------┬R Q 01
                               └R Q 02
```

Figure 108: MFD-Titan circuit diagram with comparator

```
CP02         +
 >I1
 >I2
```

Parameter display and parameter set for the comparator function block:

CP02	Function block analog value comparator number 02
+	Appears in the parameter display
>I1	Comparison value 1
>I2	Comparison value 2

Inputs

The function block inputs >I1 and >I2 can have the following operands:

- Constants
- Markers MD, MW, MB
- Analog inputs IA01 to IA04
  - IA01: terminal I7
  - IA02: terminal I8
  - IA03: terminal I11
  - IA04: terminal I12
- Analog output QA01
- Actual value … QV> of another function block

Contacts

- CP01LT to CP32LT, (less than)

Contact (make contact) switches to status 1, if the value on >I1 is less than the value on >I2; >I1 < >I2.

- CP01EQ to CP32EQ, (equal to)

Contact (make contact) switches to status 1, if the value on >I1 is equal to the value on >I2; >I1 = I2.

- CP01GT to CP32GT, (greater than)

Contact (make contact) switches to status 1, if the value at >I1 is greater than the value at >I2; >I1 > I2.

Memory requirement of the counter relay
The comparator function block requires 32 bytes of memory plus 4 bytes per constant on the function block inputs.

Text output function block

The MFD device provides 32 function blocks that operate in an easy800 as text output function blocks. These function blocks work in the MFD device as contacts and coils in the same way as in an easy800. In the MFD device, texts, actual value output and setpoint entry are implemented using the appropriate visualization elements.

→ The text function block was adopted from the easy800 so that the circuit diagram is compatible. This simplifies the processing of easy800 programs. easy800 programs with text function blocks can be loaded directly onto the MFD-Titan, either from the memory card or from EASY-SOFT-PRO. The contacts and coils are processed in the same way as in an easy800.

However, texts are not output.

The function block requires the entire memory, even if no texts have been transferred.

→ Do not use the text function block on the MFD device. This will use up memory unnecessarily with unused functions!

```
M 42----------------------------S D 01EN
D 01Q1--------------------------S Q 02
```

Figure 109: MFD-Titan circuit diagram with a text output function block

Contacts

A contact has been assigned to the text output function block.
D01Q1 to D32Q1, text function block is active.

Coils

D01EN to D32EN, enable of the text function block

Memory requirement of the text output function block

The text output function block function block requires 160 bytes of memory. This is irrespective of the text size.

Data function block

The data function block allows you to selectively save a value. Setpoint values for the function block can be saved in this manner.

```
GT01Q1---------------------------[ DB16T
DB16Q1---------------------------S D 02EN
```

Figure 110: MFD-Titan circuit diagram with data function block:

```
DB16          +
>I1
 QV>
```

Parameter display and parameter set for the data function block:

DB16	Data block function block number 16
+	Appears in the parameter display
>I1	Input value
QV>	Actual value

Inputs

The function block input >I1 can be assigned the following operands:

- Constants
- Markers MD, MW, MB
- Analog inputs IA01 to IA04
  - IA01: terminal I7
  - IA02: terminal I8

– IA03: terminal I11
– IA04: terminal I12
- Analog output QA01
- Actual value … QV> of another function block

Output
The function block output QV> can be assigned the following operands:

- Markers MD, MW, MB
- Analog output QA01

Contacts
DB01Q1 to DB32Q1

Contact (make contact) DB..Q1 switches to status 1 if the trigger signal is set to 1.

Coils
DB01T_ to DB32T_, acceptance of the value at >I1 with a rising edge.

Memory requirement of the data function block
The data block function block requires 36 bytes of memory plus 4 bytes per constant at the function block input.

Retention
Data function blocks can be operated with retentive actual values. The quantity can be selected in the SYSTEM → RETENTION menu.

Function of the data function block

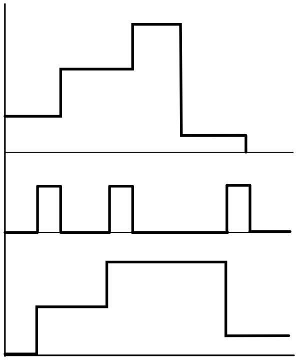


Figure 111: Signal diagram of data function block

1: value at input >I1

2: trigger coil DB..T_

3: value on DB..QV>

→ The value at input >I1 is only transferred with a rising trigger edge to an operand (e.g.: MD42, QA01) on output QV>. Output QV retains its value until it is overwritten.

PID controller

MFD-Titan provides 32 PID controllers DC01 to DC32. The PID controllers allow you to implement closed-loop control functions.

Caution!
A knowledge of closed-loop control is required in order to use the PID controllers.

The control system must be familiar so that the PID controller can function correctly.

→ Three separate manipulated variables can be output. One manipulated variable can be output via an analog output. Two manipulated variables can be processed via two pulse-width modulated outputs. It is therefore useful to run up to three closed-loop controllers per program simultaneously. Projects can be structured by selecting the controller number.

Example: Project with 3 devices

Program 1: Controller DC 10, 11

Program 2: Controller DC20, 21 and 22

Program 3: Controller DC30

Wiring a PID controller

You integrate a PID controller in your circuit as a contact and coil.

```
M 51--------------------------┬Ɛ DC02EN
                              ├Ɛ DC02EP
                              ├Ɛ DC02EI
                              └Ɛ DC02ED
M 52---------------------------Ɛ DC02SE
DC02LI-------------------------S M 96
```

Figure 112: MFD-Titan circuit diagram with PID controller

```
DC02 UNP      +
 >I1
 >I2
 >KP
 >TN
 >TV
 >TC
 >MV
  QV>
```

Parameter display and parameter set for PID controller:

DC02	PID controller function block number 02
UNP	Unipolar mode
+	Appears in the parameter display
>I1	Setpoint of PID controller
>I2	Actual value of PID controller
>KP	Proportional gain K_p
>TN	Reset time T_n
>TV	Rate time T_v

>TC	Scan time
>MV	Manual manipulated variable
QV>	Manipulated variable

In the parameter display of a PID controller you set the operating mode, the setpoints and enable the parameter display.

Operating modes of the PID controller

Parameter	Manipulated variable is output as
UNP	Unipolar 12-bit value 0 to +4095
BIP	Bipolar 13-bit value (signed 12-bit value) –4096 to +4095

Inputs

The function block inputs >I1, >I2, >KP, >TN, >TV, >TC and >MV can have the following operands:

- Constants
- Markers MD, MW, MB
- Analog inputs IA01 to IA04
  - IA01: terminal I7
  - IA02: terminal I8
  - IA03: terminal I11
  - IA04: terminal I12
- Analog output QA01
- Actual value … QV> of another function block

Actual value …QV>

The actual value …QV> can be assigned the following operands:

- Markers MD, MW, MB
- Analog output QA01

Value range for inputs and outputs

		Value range	Resolution/unit
>I1	Setpoint of PID controller	–32 768 to +32767	
>I2	Actual value of PID controller,	–32 768 to +32767	
>KP	Proportional gain K_p	0 to 65535	in -- /%
>TN	Reset time T_n	0 to 65535	in 100/ms
>TV	Rate time T_v	0 to 65535	in 100/ms
>TC	Scan time	0 to 65535	in 100/ms
>MV	Manual manipulated variable	–4096 to +4095	
QV>	Manipulated variable	0 to 4095 (unipolar) –4096 to +4095 (bipolar)	

Example:

		Value at input	Value processed in the function block.
>KP	Proportional gain K_p	1500	15
>TN	Reset time T_n	250	25 s
>TV	Rate time T_v	200	20 s
>TC	Scan time	500	50 s
>MV	Manual manipulated variable	500	500

Displaying the parameter set in the PARAMETERS menu

- + Access enabled
- – Access disabled

Contacts

DC01LI to DC32LI, value range of the manipulated variable exceeded.

Coils

- DC01EN to DC32EN: Enable PID controller;
- DC01EP to DC32EP: Activate proportional component;
- DC01EI to DC32EI: Activate integral component;

- DC01ED to DC32ED: Activate the differential component;
- DC01SE to DC32SE: Activate the manual manipulated variable

Memory requirement of the PID controller

The PID controller function block requires 96 bytes of memory plus 4 bytes per constant on the function block input.

Function of the PID controller function block

The PID controller works on the basis of the PID algorithm. According to this, the manipulated variable Y(t) is the result of the calculation of the proportional component, an integral component and a differential component.

→ The PID controller must be enabled so that it can work. Coil DC..EN is active. If coil DC..EN is not active, the entire PID controller is deactivated and reset. The manipulated variable is set to zero.

The corresponding coils for the P, I and D components must be active.

Example: If only coils DC..EP and DC..EI are activated, the controller operates as a PI controller.

→ The device calculates the manipulated variable every time the scan time T_C has elapsed. If the scan time is zero, the manipulated variable is calculated every cycle.

Equation of PID controller:

$$Y(t) = Y_P(t) + Y_I(t) + Y_D(t)$$

$Y(t)$ = calculated manipulated variable with scan time *t*

$Y_P(t)$ = Value of the proportional component of the manipulated variable with scan time *t*

$Y_I(t)$ = Value of the integral component of the manipulated variable with scan time *t*

$Y_D(t)$ = Value of the differential component of the manipulated variable with scan time *t*

The proportional component in the PID controller

The proportional component Y_P is the product of the gain (K_p) and the control difference (e). The control difference is the difference between the setpoint (X_s) and the actual value (X_i) at a specified scan time. The equation used by the device for the proportional component is as follows:

$$Y_P(t) = K_p \times [X_s(t) - X_i(t)]$$

K_p = Proportional gain

$X_s(t)$ = Setpoint with scan time t

$X_i(t)$ = Actual value with scan time t

The integral component in the PID controller

The integral component Y_I is proportional to the sum of the control difference over time. The equation used by the device for the integral component is as follows:

$$Y_I(t) = K_p \times T_c/T_n \times [X_s(t) - X_i(t)] + Y_I(t-1)$$

K_p = Proportional gain

T_c = Scan time

T_n = Integration time (also known as reset time)

$X_s(t)$ = Setpoint with scan time t

$X_i(t)$ = Actual value with scan time t

$Y_I(t-1)$ = Value of the integral component of the manipulated variable with scan time $t-1$

The differential component in the PID controller

The differential component Y_D is proportional to the change in the control difference. So as to avoid step changes or jumps in the manipulated variable caused by the differential behaviour when the setpoint is changed, the change of the actual value (the process variable) is calculated and not the change in the control difference. This is shown in the following equation:

$$Y_D(t) = K_p \times T_v/T_c \times (X_i(t-1) - X_i(t))$$

K_p = Proportional gain

T_c = Scan time

T_v = Differential time of the control system (also called the rate time)

$X_i(t)$ = Actual value with scan time *t*

$X_i(t-1)$ = Actual value with scan time *t* – 1

Scan time T_C

Scan time T_C determines the duration of the interval in which the function block is called by the operating system for processing. The value range is between 0 and 6553.5 s.

If the value 0 is set, the cycle time of the device is the pause time between the function block calls.

→ The device cycle time varies according to the length of the program. With a scan time of 0 s, this may cause an irregular control response.

→ Use the Set cycle time function block (→ page 256) in order to keep the cycle time of the device constant.

Manual mode of the PID controller

A value must be present at the >MV input in order to set the manipulated variable directly. If the coil DC..SE is activated, the value at >MV is transferred as manipulated variable QV>. This value is present for as long as the DC..SE coil is activated or the value at the >MV input is changed. If coil DC..SE is deactivated, the control algorithm is reactivated.

→ Extreme changes in the manipulated variable can occur when the manual manipulated variable is transferred or deactivated.

→ If the function block is running in UNI (unipolar) mode, a negative signed manipulated variable value will be output.

Signal smoothing filter

MFD provides 32 signal smoothing filters FT01 to FT32. The function block allows you to smooth noisy input signals.

Wiring a signal smoothing filter

You can integrate a signal smoothing filter into your circuit as a coil.

```
M 48----------------------------Ę FT17EN
```

Figure 113: MFD-Titan circuit diagram with smoothing function block

```
FT17           +
 >I1
 >TG
 >KP
 QV>
```

Parameter display and parameter set for the FT function block:

FT17	FT PT1 signal smoothing filter function block, number 17
+	Appears in the parameter display
>I1	Input value
>TG	Recovery time
>KP	Proportional gain
QV>	Output value, smoothed

→ The recovery time T_g is the time in which the output value is calculated.

The recovery time T_g must be set so that it is an integer multiple of the cycle time or controller scan time T_c.

Inputs

The function block inputs >I1, >I2 and >KP can have the following operands:

- Constants
- Markers MD, MW, MB
- Analog inputs IA01 to IA04
  - IA01: terminal I7
  - IA02: terminal I8
  - IA03: terminal I11
  - IA04: terminal I12
- Analog output QA01
- Actual value … QV> of another function block

Output

The function block output QV> can be assigned the following operands:

- Markers MD, MW, MB
- Analog output QA01

Value range for inputs and outputs

		Value range	Resolution/unit
>I1	Input value of the function block	–32 768 to +32767	
>TG	Recovery time T_g	0 to 65535	in 100/ms
>KP	Proportional gain K_p	0 to 65535	in -- /%
QV>	Output value	–32 768 to +32767	

Example:

		Value at input	Value processed in the function block.
>TG	Recovery time T_g	250	25 s
>KP	Proportional gain K_p	1500	15

Displaying the parameter set in the PARAMETERS menu

- + Access enabled
- – Access disabled

Coil

FT01EN to FT32EN, function block enable

Memory requirement of the FT function block

The FT function block requires 56 bytes of memory plus 4 bytes per constant on the function block input.

Function of the signal smoothing filter function block

→ The signal smoothing filter must be enabled so that it can work. Coil FT..EN is active. If coil FT..EN is not active, the entire function block is deactivated and reset. The output value is set to zero.

If the function block is called for the first time, the output value is initialised with the input value when the device is started or after a reset. This speeds up the startup behaviour of the function block.

→ The function block updates the output value every time recovery time T_g expires.

The function block operates according to the following equation:

$$Y(t) = [T_a/T_g] \times [K_p \times x(t) - Y(t{-}1)]$$

$Y(t)$	= Calculated output value for scan time t
T_a	= Scan time
T_g	= Recovery time
K_p	= Proportional gain
$x(t)$	= Actual value with scan time t
$Y(t{-}1)$	= Output value with scan time $t - 1$

Scan time:
Scan time T_a depends on the set recovery time value.

Recovery time T_g	Scan time T_a
0.1 s to 1 s	10 ms
1 s to 6553 s	$T_g \times 0.01$

GET, fetch a value from the network

The function block allows you to selectively read (get) a 32 bit value from the network. The GET function block fetches data which another station has made available on the easy-NET network with a PUT function block.

```
GT01Q1----------------------------[ DB16T
```

Figure 114: MFD-Titan circuit diagram with GET function block

```
GT01 02 20     +
 QV>
```

Parameter display and parameter set for the GET function block:

GT01	GET function block (fetch a value from the network), number 01
02	Station number from which the value is sent. Possible station number: 01 to 08
20	Send function block (PT 20) of the sending station. Possible function block number: 01 to 32
+	Appears in the parameter display
QV>	Actual value from the network

Output

The function block output QV> can be assigned the following operands:

- Markers MD, MW, MB
- Analog output QA01

Contacts

GT01Q1 to GT32Q1

Contact (make contact) GT..Q1 switches to status 1 if a new value transferred on the easy-NET network is present.

Memory requirement of the GET function block

The GET function block requires 28 bytes of memory.

GET diagnostics

The GET function block only functions when the easy-NET network is functioning correctly (→ Section "Signs of life of the individual stations and diagnostics", Page 346).

Function of the GET function block

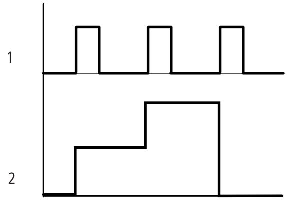


Figure 115: Signal diagram of GET function block

1: GT..Q1

2: value on GT..QV>

→ The GET function blocks are assigned the value 0 when the power supply is switched on.

Seven-day time switch

MFD-Titan is equipped with a real-time clock which you can use in the circuit diagram as a 7-day time switch and a year time switch.

→ The procedure for setting the time is described under Section "Setting date, time and daylight saving time" on Page 369.

MFD provides 32 seven-day time switches HW01 to HW32 for a total of 128 switching times.

Each time switch has four channels which you can use to set four on and off times. The channels are set via the parameter display.

The time is backed up in the event of a power failure and continues to run, although the time switch relays will no longer switch. The contacts are kept open when de-energized. Refer to Chapter "Technical data", Page 409 for information on the buffer time.

Wiring of a 7-day time switch

A 7-day time switch is integrated into the circuit diagram as a contact.

```
HW14Q1-----------------------------[ Q 01
```

Figure 116: MFD-Titan circuit diagram with 7-day time switch

```
HW14  A       +
 >DY1
 >DY2
 >ON
 >OFF
```

Parameter display and parameter set for the 7-day time switch HW:

HW14	7-day time switch function block number 14
A	Time switch channel A
+	Appears in the parameter display
>DY1	Day 1
>DY2	Day 2
>ON	On time
>OFF	Off time

Channels

4 channels are available per time switch, channels A, B, C and D. These channels all act on the contact of the 7-day time switch.

Day 1 and day 2

Either the time period acts from day 1 to day 2, e.g. Monday to Friday, or for one day only.

Monday = MO, Tuesday = TU, Wednesday = WE, Thursday = TH, Friday = FR, Saturday = SA, Sunday = SU,

Time

00:00 to 23:59

Displaying the parameter set in the PARAMETERS menu

- + Access enabled
- – Access disabled

Contacts

HW01Q1 to HW32Q1

Memory requirement of the 7-day time switch

The 7-day time switch function block requires 68 bytes of memory plus 4 bytes per channel used.

Function of the 7-day time switch

The switching points are defined according to the parameters entered.

MO to FR: on the weekdays Mo, Tu, We, Th, Fr

ON 10:00, OFF 18:00: on and off switching times for the individual days of the week.

MO: every Monday
ON 10:00: switch on time

SA: every Saturday
OFF 18:00: switch off time

Switching on working days

Time switch HW01 switches on Monday to Friday between 6:30 and 9:30 and between 17:00 and 22:30.

HW01 A +	HW01 B +
>DY1 MO	>DY1 MO
>DY2 FR	>DY2 FR
>ON 06:30	>ON 17:00
>OFF 09:30	>OFF 22:30

Switching at the weekend

Time switch HW02 switches on at 16:00 on Friday and switches off at 6:00 on Monday.

```
HW02 A       +
>DY1 FR
>DY2
>ON   16:00
>OFF
```

```
HW02 B       +
>DY1 MO
>DY2
>ON
>OFF 06:00
```

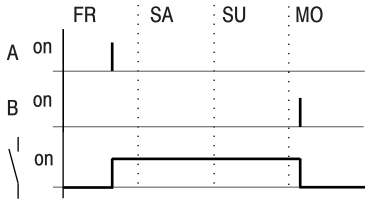


Figure 117: Signal diagram of "weekend"

Overnight switching

Time switch HW03 switches on overnight at 22:00 Monday and switches off at 6:00 on Tuesday.

```
HW03 D       +
>DY1 MO
>DY2
>ON   22:00
>OFF 06:00
```

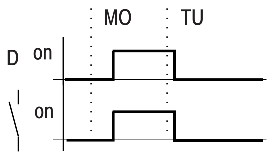


Figure 118: Signal diagram for night switching

→ If the Off time is before the On time, MFD will switch off on the following day.

Time overlaps

The time settings of a time switch overlap. The clock switches on at 16:00 on Monday, whereas on Tuesday and Wednesday it switches on at 10:00. On Monday to Wednesday the switching-off time is 22:00.

HW04 A +	HW04 B +
>DY1 MO	>DY1 TU
>DY2 WE	>DY2 WE
>ON 16:00	>ON 10:00
>OFF 22:00	>OFF 00:00

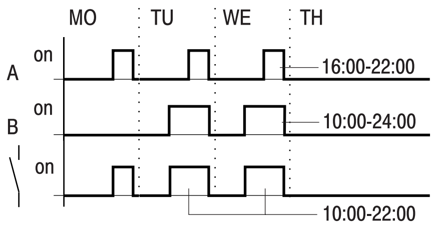


Figure 119: Signal diagram of overlaps

→ Switch-on and switch-off times are always based on the channel which switches first.

Response in the event of a power failure

The power is removed between 15:00 and 17:00. The relay drops out and remains off, even after the power returns, since the first switching-off time was at 16:00.

HW05 A +	HW05 B +
>DY1 MO	>DY1 MO
>DY2 SU	>DY2 SU
>OFF 16:00	>ON 12:00
	>OFF 18:00

→ When switched on, MFD always updates the switching state on the basis of all the available switching time settings.

24 hour switching

The time switch is to switch for 24 hours. Switch-on time at 0:00 on Monday and switch-off time at 0:00 on Tuesday.

```
HW20 A      +
>DY1 MO
>DY2
>ON   00:00
>OFF
```

```
HW20 B      +
>DY1 TU
>DY2
>ON
>OFF  00:00
```

Year time switch

MFD-Titan is equipped with a real-time clock which you can use in the circuit diagram as a 7-day time switch and a year time switch.

→ The procedure for setting the time is described under Section "Setting date, time and daylight saving time" on Page 369.

MFD provides 32 year time switches HY01 to HY32 for a total of 128 switching times.

Each time switch has four channels which you can use to set four on and off times. The channels are set via the parameter display.

The time and date are backed up in the event of a power failure and continue to run. However, the time switch relays will no longer continue to switch. The contacts are kept open when de-energized. Refer to Chapter "Technical data", Page 409 for information on the buffer time.

Wiring of a year time switch

A year time switch is integrated into the circuit diagram as a contact.

```
HY30Q1----------------------------[ S 08
```

Figure 120: MFD-Titan circuit diagram with year time switch

```
HY30   B        +
 >ON
 >OFF
```

Parameter display and parameter set for the year time switch HY:

HY30	Year time switch function block number 30
B	Time switch channel B
+	Appears in the parameter display
>ON	Switch on time
>OFF	Switch off time

Channels

4 channels are available per time switch, channels A, B, C and D. These channels all act on the contact of the year time switch.

Date

Day.Month.Year: DD.MM. YY

Example: 11.11.02

On/off switch points

ON: switch on time
OFF: switch off time

→ The switch on year must not be later than the switch off year. Otherwise the year time switch will not function.

Displaying the parameter set in the PARAMETERS menu

- + Access enabled
- – Access disabled

Contacts
HY01Q1 to HY32Q1

Memory requirement for the year time switch
The year time switch function block requires 68 bytes of memory plus 4 bytes per channel used.

Function of the year time switch function block
The year time switch can operate with ranges, individual days, months, years or combinations.

Years
ON: 2002 to OFF: 2010 means:
Switch on at 00:00 on 01.01.2002 00 and switch off at 00:00 on the 01.01.2011.

Months
ON: 04 to OFF: 10 means:
Switch on at 00:00 on 1st April and switch off at 00:00 on 1st November

Days
ON: 02 to OFF: 25 means:
Switch on at 00:00 on the 2nd and switch off at 00:00 on the 26th

Rules for the year time switch
The contact switches in the defined years (ON to OFF), the defined months (ON to OFF) and in the days entered (ON to OFF).

Time ranges must be input with two channels, one for ON and one for OFF.

Overlapping channels:
The first ON date switches on and the first OFF date switches off.

→ Avoid incomplete entries. It hinders transparency and leads to unwanted functions.

```
HY01  A       +
 >ON  --.--.02
 >OFF --.--.05
```

Example 1
Year range selection
The year time switch HY01 should switch on at 00:00 on January 1 2002 and remain on until 23:59 on 31 December 2005.

```
HY01  A       +
 >ON  __.03.--
 >OFF --.09.--
```

Example 2
Month range selection
The year time switch HY01 should switch on at 00:00 on 01st March and remain on until 23:59 on 30th September.

```
HY01  A       +
 >ON  01.--.--
 >OFF 28.--.--
```

Example 3
Day range selection
The year time switch HY01 should switch on at 00:00 on the 1st of each month and remain on until 23:59 on the 28th of each month.

```
HY01  A       +
 >ON  25.12.--
 >OFF 26.12.--
```

Example 4
Holiday selection
The year time switch HY01 should switch on at 00:00 on the 25.12 each year and remain on until 23:59 on 26.12. "Christmas program"

Example 5
Time range selection
The year time switch HY01 should switch on at 00:00 on 01.05 each year and remain on until 23:59 on the 31.10. "Open air season"

```
HY01  A       +
 >ON  01.05.--
 >OFF --.--.--
```

```
HY01  B       +
 >ON  --.--.--
 >OFF 31.10.--
```

Example 6
Overlapping ranges

The year time switch HY01 channel A switches on at 00:00 on the 3rd of the months 5, 6, 7, 8, 9, 10 and remains on until 23:59 on the 25th of these months.

The year time switch HY01 channel B switches on at 00:00 on the 2nd in the months 6, 7, 8, 9, 10, 11, 12 and remains on until 23:59 on 17th of these months.

```
HY01  A      +
 >ON  03.05.--
 >OFF 25.10.--
```

```
HY01  B      +
 >ON  02.06.--
 >OFF 17.12.--
```

Total number of channels and behaviour of the contact HY01Q1:
The time switch will switch on at 00:00 from the 3rd May and off at 23:59 on the 25th May.
In June, July, August, September, October, the time switch will switch on at 00:00 on the 2nd of the month and switch off at 23:59 on the 17th.
In November and December, the time switch will switch on at 00:00 on the 2nd of the month and switch off at 23:59 on the 17th.

Value scaling

MFD provides 32 value scaling function blocks LS01 to LS32. The function block enables you to convert values from one value range to another one. In this way it is possible to reduce or increase values.

Wiring of a scaling function block

You can integrate a value scaling function block into your circuit as a coil.

```
M 48----------------------------Ä LS27EN
```

Figure 121: MFD-Titan circuit diagram with LS value scaling

```
LS27        +
 >I1
 >X1
 >Y1
 >X2
 >Y2
 QV>
```

Parameter display and parameter set for the LS function block:

LS27	LS value scaling function block number 27
+	Appears in the parameter display
>I1	Input value, actual value source range
>X1	Lower value of source range
>Y1	Lower value of target range
>X2	Upper value of source range
>Y2	Upper value of target range
QV>	Output value, scaled

Inputs

The function block inputs >I1, >X1, >X2, >Y1 and >Y2 can have the following operands:

- Constants
- Markers MD, MW, MB
- Analog inputs IA01 to IA04
  - IA01: terminal I7
  - IA02: terminal I8
  - IA03: terminal I11
  - IA04: terminal I12
- Analog output QA01
- Actual value … QV> of another function block

Output

The function block output QV> can be assigned the following operands:

- Markers MD, MW, MB
- Analog output QA01

Value range for inputs and outputs

		Value range
>I1	Input value of the function block	–2 147 483 648 to +2 147 483 647
>X1	Lower value of source range	
>X2	Lower value of target range	
>Y1	Upper value of source range	
>Y2	Upper value of target range	
QV>	Output value	

Displaying the parameter set in the PARAMETERS menu

- + Access enabled
- – Access disabled

Coil

VC01EN to VC32EN, function block enable

Memory requirement of the LS function block

The LS function block requires 64 bytes of memory plus 4 bytes per constant on the function block input.

Function of the function block

→ The scaling function block must be enabled so that it can work. Coil LS..EN is active. If coil LS..EN is not active, the entire function block is deactivated and reset. The output value is set to zero.

The function block operates according to the following equation:

$$Y(x) = X \times \frac{Y_2 - Y_1}{X_2 - X_1} + \frac{X_2 \times Y_1 - X_1 \times Y_2}{X_2 - X_1}$$

Y(x) = Actual output value of target range

X = Actual input value of source range

X_1 = Lower value of source range

X_2 = Upper value of source range
Y_1 = Lower value of target range
Y_2 = Upper value of target range

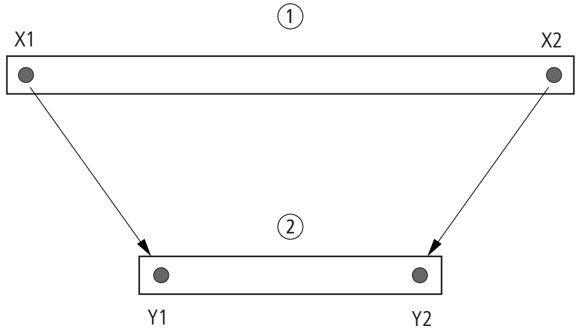


Figure 122: Value scaling function block – Reduce value range

① Source range
② Target range

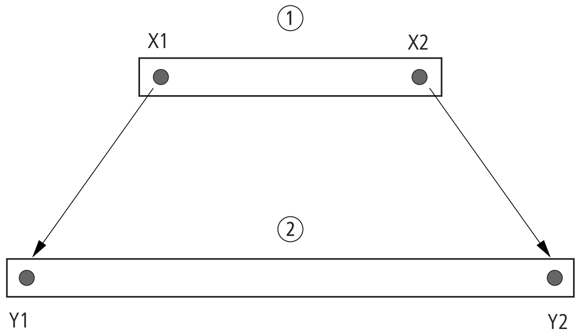


Figure 123: Value scaling function block – Increase value range

① Source range
② Target range

Example 1:
The source range is a 10-bit value, source is the analog input IA01.

The target range has 12 bits.

```
LS01         +
 >I1  IA01
 >X1  0
 >Y1  0
 >X2  1023
 >Y2  4095
 QV>
```

Parameter display and parameter set for the LS01 function block:

The actual value at the analog input IA01 is 511.
The scaled output value is 2045.

Example 2:
The source range has 12 bits.

The target range has 16 signed bits.

>I1 = DC01QV

>X1 = 0

>X2 = 4095

>Y1 = –32768

>Y2 = +32767

The actual value at the analog input DC01QV 1789.
The scaled output value is –4137.

Jumps

Jumps can be used to optimise the structure of a circuit diagram or to implement the function of a selector switch. For example, jumps can be used to implement whether manual/automatic mode, or different machine programs are to be selected.

Jumps consist of a jump location and a jump destination (label).

Circuit diagram symbols for jumps

Contact	
Make contact[1)]	:
Numbers	01 to 32
Coils	{
Numbers	01 to 32
Coil function	{, }, ⌐, ⎾, ⎿

1) can only be used as first leftmost contact

Function

If the jump coil is triggered, the rungs coming directly after it will not be processed. The states of the coils before the jump will be retained, unless they are overwritten in rungs that were not missed by the jump. Jumps are always made forwards, i.e. the jump ends on the first contact with the same number as that of the coil.

- Coil = jump when 1
- Contact only at the first left-hand contact position = Jump destination

The Jump contact point will **always be set to 1**

→ Backward jumps are not possible with MFD due to the way it operates. If the jump label does not come after the jump coil, the jump will be made to the end of the circuit diagram. The last rung will also be skipped.

If a jump destination is not present, the jump is made to the end of the circuit diagram.

Multiple use of the same jump coil and jump contact is possible as long as this is implemented in pairs, i.e.:
Coil {:1/jumped range/Contact:1,
Coil {:1/jumped range/Contact:1 etc.

Attention!
If rungs are skipped, the states of the coils are retained. The time value of timing relays that have been started will continue to run.

Power flow display

Jumped ranges are indicated by the coils in the power flow display.

All coils after the jump coil are shown with the symbol of the jump coil.

Example

A selector switch allows two different operations to be set.

- Sequence 1: Switch on Motor 1 immediately.
- Sequence 2: Switch on Guard 2, Wait time, then switch on Motor 1.

Contacts and relays used:

- I1 Sequence 1
- I2 Sequence 2
- I3 Guard 2 moved out
- I12 Motor-protective circuit-breaker switched on
- Q1 Motor 1
- Q2 Guard 2
- T 01 Wait time 30.00 s, on-delayed
- D 01 Text “motor-protective circuit-breaker tripped”

Circuit diagram:

```
I 01-------[ : 01
I 02-------[ : 02
: 01
---------┬-[ Q 01
         └-R Q 02
-----------[ : 08
: 02-------[ Q 02
Q 02-I 03-T T 02
T 02-------[ Q 01
: 08
Ī12 -------[ D 01
```

Power flow display: I 01 selected:

```
I 01-------[ : 01
I 02---------: 01
: 01
-----------┬[ Q 01
           └R Q 02
-----------[ : 08
: 02---------: 08
Q 02--I 03--: 08
T 02---------: 08
: 08
Ī12 -------[ D 01
```

Range from jump label 1 processed.

Jump to label 8.
Range to jump label 8 skipped.

Jump label 8, circuit diagram processed from this point on.

Master reset

The master reset function block allows you to reset the state of the markers and all outputs to the 0 state with a single command. Depending on the operating mode of this function block, it is possible to reset the outputs only, or the markers only, or both. 32 function blocks are available.

```
M 96----------------------------[ MR16T
```

Figure 124: MFD-Titan circuit diagram with master reset function block

```
MR16  Q       +
```

Parameter display and parameter set for the master reset function block:

MR16	Master reset function block number 16
Q	Reset outputs mode
+	Appears in the parameter display

Operating modes

- Q: Acts on the outputs Q.., *Q.., S.., *S.., *SN.., QA01; *: network station address
- M: acts on the marker range MD01 to MD48.
- ALL: acts on Q and M.

Contacts

MR01Q1 to MR32Q1

The contact switches on the marker if the trigger coil MR..T has the 1 state.

Coils

MR01T to MR32T: trigger coils

Memory requirement of the data function block

The master reset function block requires 20 bytes of memory.

Function of the data master reset

The outputs or the markers are set to the 0 state in accordance with the operating mode when a rising edge is detected on the trigger coil.

→ The master reset function block should be used as the last function block in a program so that all data ranges are cleared reliably. Otherwise subsequent function blocks may overwrite the data ranges.

The contacts MR01Q1 to MR32Q1 assume the state of their own trigger coil.

Numerical converters

MFD-Titan provides 32 numerical converters NC01 to NC32.

A numerical converter function block enables you to convert BCD coded values to decimal values or decimal coded values to BCD coded values.

Wiring of a numerical converter

A numerical converter in the circuit diagram only has the enable coil.

```
I 05----------------------------Ĉ NC02EN
```

Figure 125: MFD-Titan circuit diagram with numerical converter

```
NC02 BCD    +
 >I1
  QV>
```

Parameter display and parameter set for the numerical converter:

NC02	Numerical converter function block number 02
BCD	Convert BCD code to decimal value mode
+	Appears in the parameter display
>I1	Input value
QV>	Output value

In the parameter display of a numerical converter you can change the mode and the enable of the parameter display.

Numerical converter modes

Parameter	Mode
BCD	Convert BCD coded values to decimal values
BIN	Convert decimal value to BCD coded values

Number range

Value	Number system
–161 061 273 to +161 061 273	BCD
–9 999 999 to +9 999 999	Decimal

BCD code	Decimal value
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010 to 1111	Not permissible
10000	10
10001	11

→ The BCD code only allows the number range 0_{hex} to 9_{hex}. The number range A_{hex} to F_{hex} cannot be represented. The NC function block converts the impermissible range to 9.

Inputs

The function block input >I1 can be assigned the following operands:

- Constants
- Markers MD, MW, MB
- Analog inputs IA01 to IA04
  - IA01: terminal I7
  - IA02: terminal I8
  - IA03: terminal I11
  - IA04: terminal I12
- Analog output QA01
- Actual value … QV> of another function block

Actual value …QV>

The actual value …QV> can be assigned the following operands:

- Markers MD, MW, MB
- Analog output QA01

Displaying the parameter set in the PARAMETERS menu

- + Access enabled
- – Access disabled

Coil

NC01EN to NC32EN: enable coil.

Memory requirement of the numerical converter

The numerical converter function block requires 32 bytes of memory plus 4 bytes per constant on the function block input.

Function of the numerical converter function block

→ The numerical converter function block must be enabled so that it can work. Coil NC..EN is active. If coil NC..EN is not active, the entire function block is deactivated and reset. The output value is set to zero.

BCD mode

The BCD value at >I1 is provided in decimal format at the input. The binary value is formed from this. The binary value is interpreted as a BCD value. Values greater than 9 (1001) are set to the value 9. The BCD value is output as a decimal value at the output QV>.

Example 1:
Input value >I1: $+9_{dec}$
Binary value: 1001
Decimal value QV>: + 9

Example 2:
Input value>I1: $+14_{dec}$
Binary value: 1110
Decimal value QV>: + 9

→ The highest binary value represented in BCD is 1001 = 9. All other higher binary values from 1010 to 1111 are output as 9. This behaviour is correct as BCD encoders normally don't generate these values.

Example 3:
Input value>I1: 19_{dec}
Binary value: 0001 0011
Decimal value QV>: 13

Example 4:
Input value>I1: $161\,061\,273_{dec}$
Binary value: 1001 1001 1001 1001 1001 1001 1001
Decimal value QV>: 9 999 999

Example 5:
Input value>I1: $–61\,673_{dec}$
Binary value: 1000 0000 0000 0000 1111 0000 1110 1001
Decimal value QV>: –9 099

→ Bit 32 is the sign bit. Bit 32 = 1 → Sign = Minus.

Example 6:
Input value>I1: $2\,147\,483\,647_{dec}$
Binary value: 0111 1111 1111 1111 1111 1111 1111 1111
Decimal value QV>: 9 999 999

→ Values greater than 161 061 273 are output as 9 999 999. Values less than –161 061 273 are output as –9 999 999. The working range of the function block has been exceeded.

BIN mode

The decimal value is assigned to input >I1. The decimal value is represented as a BCD coded value. The BCD coded value is interpreted as a hexadecimal value and output as a decimal value at output QV>.

Example 1:
Input value >I1: $+7_{dec}$
BCD binary value: 0111
Hexadecimal value: 0111
Decimal value QV>: + 7

Example 2:
Input value >I1: $+11_{dec}$
BCD binary value: 0001 0001
Hexadecimal value: 0001 0001
Decimal value QV>: +17 (1 + 16)

Hexadecimal value:
Bit 0 has the value 1.
Bit 4 has the value 16
Total: Bit 0 plus Bit 4 = 17

Example 3:
Input value >I1: 19_{dec}
BCD binary value: 0001 1001
Hexadecimal value: 0001 1001
Decimal value QV>: 25 (1 + 8 + 16)

Example 4:
Input value >I1: $9\,999\,999_{dec}$
BCD binary value: 1001 1001 1001 1001 1001 1001 1001
Hexadecimal value: 1001 1001 1001 1001 1001 1001 1001
Decimal value QV>: 161 061 273

Example 5:
Input value >I1: $-61\,673_{dec}$
BCD binary value:
1000 0000 0000 0110 0001 0110 0111 0011
Hexadecimal value:
1000 0000 0000 0110 0001 0110 0111 0011
Decimal value QV>: –398 963

→ Bit 32 is the sign bit. Bit 32 = 1 → Sign = Minus.

Example 6:
Input value >I1: $2\,147\,483\,647_{dec}$
BCD binary value:
0111 1111 1111 1111 1111 1111 1111 1111

Hexadecimal value:
01111111111111111111111111111111111
Decimal value QV>: 161 061 273

→ Values greater than 9 999 999 are output as 161 061 273. Values less than –9 999 999 are output as –161 061 273. The working range of the function block has been exceeded.

Operating hours counter

MFD-Titan provides 4 independent operating hours counters. The counter states are retained even when the device is switched off. As long as the enable coil of the operating hours counter is active, MFD-Titan counts the hours in minute cycles.

```
I 10-----------------------------[ OT01EN
M 42-----------------------------[ OT01RE
OT01Q1---------------------------[ Q 01
```

Figure 126: MFD-Titan circuit diagram with operating hours counter.

```
OT04          +
 >I1
 QV>
```

Parameter display and parameter set for the operating hours counter function block:

OT04	Operating hours counter number 04
+	Appears in the parameter display
>I1	Upper threshold value in hours
QV>	Actual value of the operating hours counter

Contacts
OT01Q1 to OT04Q1

The contact switches when the upper threshold value has been reached (greater than or equal to).

Coils
- OT01EN to OT04EN: enable coil
- OT01RE to OT04RE: reset coil

Memory requirement of the operating hours counter

The operating hours counter function block requires 36 bytes of memory plus 4 bytes per constant on the function block input.

Function of the operating hours counter function block

If the enable coil OT..EN is triggered to the 1 state, the counter adds the value 1 to its actual value every minute (basic clock rate: 1 minute).

If the actual value on QV> reaches the setpoint value of >I1, the contact OT..Q1 switches for as long as the actual value is greater than or equal to the setpoint value.

The actual value is retained in the unit until the Reset coil OT..RE is actuated. The actual value is then set to zero.

→ Operating mode change RUN, STOP, Voltage On, Off, Delete program, Change program, Load new program. All these actions do not delete the actual value of the operating hours counter.

Accuracy

The operating hours counter is accurate to the nearest minute. If the enable coil signal is terminated within a minute, the value for seconds is lost.

The value range of the operating hours counter is between 0 hours and 100 years.

PUT, send a value onto the network

The function block allows you to selectively send a 32 bit value onto the network. The PUT function block provides data on the easy-NET that another station indicates it requires via the GET function block.

```
T 01Q1-----------------------------Ä PT16T
PT16Q1-----------------------------Ä C 01C
```

Figure 127: MFD-Titan circuit diagram with PUT function block

```
PT01 11          -
 >I1
```

Parameter display and parameter set for the PUT function block:

PT01	PUT function block (places a value onto the network), number 11
-	Does not appear in the parameter display
>I1	Setpoint value which is put onto the easy-NET network

Input

The function block input >I1 can be assigned the following operands:

- Constants
- Markers MD, MW, MB
- Analog inputs IA01 to IA04
  - IA01: terminal I7
  - IA02: terminal I8
  - IA03: terminal I11
  - IA04: terminal I12
- Analog output QA01
- Actual value … QV> of another function block

Contacts

PT01Q1 to PT32Q1: state of the trigger coil

Coils

PT01T to PT32T: trigger coils

Memory requirement of the PUT function block

The PUT function block requires 36 bytes of memory plus 4 bytes per constant on the function block input.

PUT diagnostics

The PUT function block only functions when the easy-NET network is functioning correctly (→ Section “Signs of life of the individual stations and diagnostics”, Page 346).

Function of the PUT function block

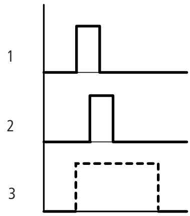


Figure 128: Signal diagram of PUT function block

1: trigger coil

2: trigger coil contact feedback

3: send

Pulse width modulation

MFD-Titan provides 2 pulse width modulation function blocks PW01 and PW02. The function blocks are connected directly to the outputs.

They are assigned as follows:

PW01 → Q1

PW02 → Q2

→ When using the pulse width modulation function block with a minimum on time of less than 1 s only use devices with transistor outputs.

The pulse width modulation function block is primarily used for outputting the manipulated variable of a PID controller. The maximum frequency is 200 kHz. This corresponds to a period duration of 5 ms. The maximum period duration is 65.5 s.

Wiring a pulse width modulation function block

A pulse width modulation function block is integrated in the circuit diagram as a contact or coil.

→ To prevent unpredictable switching states, use each coil of a relay once only in the circuit diagram.

```
M 95---------------------------[ PW01EN
PW01E1-------------------------[ M 96
```

Figure 129: MFD-Titan circuit diagram with pulse width modulation.

```
PW02          +
 >SV
 >PD
 >ME
```

Parameter display and parameter set for pulse width modulation:

PW02	Pulse width modulation function block number 02
+	Appears in the parameter display
>SV	Manipulated variable input
>PD	Period duration in ms
>ME	Minimum on duration, minimum off duration in ms

The parameter display for a timing relay is used to modify the period duration, the minimum on time and the enabling of the parameter display.

Value and time ranges

Parameter	Value and time range	Resolution
SV	0 to 4095	1 digit
PD	0 to 65535	ms
ME	0 to 65535	ms

→ The minimum time setting for the period duration is: 0.005 s (5 ms)

Inputs
The function block inputs >SV, >PD and >ME can be assigned the following operands:

- Constants
- Markers MD, MW, MB
- Analog inputs IA01 to IA04
  - IA01: terminal I7
  - IA02: terminal I8

– IA03: terminal I11
– IA04: terminal I12
- Analog output QA01
- Actual value … QV> of another function block

Displaying the parameter set in the PARAMETERS menu

- + Access enabled
- – Access disabled

Contacts

PW01E1 to PW02E1, the minimum on duration or the minimum off duration was undershot.

Coils

PW01EN to PW02EN, enable coil.

Memory requirement of the function block

The pulse width modulation function block requires 48 bytes of memory plus 4 bytes per constant on the function block input.

Function of the pulse width modulation function block

→ The pulse width modulation function block must be enabled so that it can work. Coil PW..EN is active. If coil PW..EN is not active, the entire function block is deactivated and reset. The output value is set to zero.

The manipulated variable at function block input >SV is converted to a pulse string with a constant period duration. The pulse width is proportional to the manipulated variable >SV. The period duration and the minimum on duration can be selected as required within the specified limits.

The function block causes a direct output of the pulse to the corresponding output. The output image of the circuit diagram is always updated.

→ The following applies if the output of a pulse width modulator is used as a coil in the circuit diagram:

The state of the output is not refreshed from the circuit diagram.

→ The following applies to the minimum on duration:

- The minimum on duration is the same as the minimum off duration.
- The minimum on duration must not exceed 10 % of the period duration. The ratio of period duration/minimum on duration (P/M) determines which percentage of the manipulated variable has no effect. The minimum on duration must be set as low as possible so that the P/M ratio is as high as possible. If the minimum on duration must not be too low, due to the output relay, the period duration must be increased accordingly.
- The minimum on duration is 1 ms.
- If the actual value of the pulse length is less than the minimum on duration, the minimum on duration has the effect of the pulse time. Note the state of the contact PW..E1.
- If the off duration of the pulse is less than the minimum off duration, outputs Q1 and Q2 are continuously in operation. Note the state of the contact PW..E1.

Setting date/time

This function block allows you to selectively place the date and time onto the network. All other stations accept the date and time of the sending station. The function block name is SC01 (send clock).

```
HW01QW1--------------------------Ɛ SC01T
```

Figure 130: MFD-Titan circuit diagram with SC function block

Parameter display and parameter set for the SC function block:

The SC01 function block has no parameters as it is a triggered system service.

Coil

SC01T: trigger coil

Memory requirement of the SC function block

The SC function block requires 20 bytes of memory.

SC diagnostics

The SC function block only functions when the easy-NET network is functioning correctly (→ Section "Signs of life of the individual stations and diagnostics", Page 346).

Function of the date/time function block

If the trigger coil of the function block is activated, the current date, the day of the week and time from the sending station is automatically put onto the easy-NET network. All other network stations must accept these values.

→ The station that sends its date and time does this when the seconds value is zero.

Example: The trigger pulse is actuated at 03:32:21 (hh:mm:ss). The other stations are synchronised at 03:33:00. This time is accessed by all other stations.

This process can be repeated as often as desired. The trigger coil must be triggered again from the 0 to the 1 state.

Accuracy of time synchronisation

The maximum time deviation between the functional stations is 5 s.

Set cycle time

MFD-Titan provides one set cycle time function block ST01. The set cycle time function block is a supplementary function block for the PID controller.

The set cycle time function block provides a fixed cycle time for processing the circuit diagram and the function blocks.

Wiring a set cycle time function block
The ST set cycle time function block is integrated in the circuit diagram as a coil.

→ To prevent unpredictable switching states, use each coil of a relay once only in the circuit diagram.

```
--------------------------------[ ST01EN
```

Figure 131: MFD-Titan circuit diagram with enabling of set cycle time function block.

```
ST01          +
 >I1
```

Parameter display for set cycle time:

ST01	Set cycle time function block number 01
+	Appears in the parameter display
>I1	Set cycle time

The parameter display is used to modify the set cycle time, the minimum on time and the enabling of the parameter display.

Time range

Parameter	Value and time range	Resolution
I1	0 to 1000	ms

Inputs
The function block input >I1 can be assigned the following operands:

- Constants
- Markers MD, MW, MB
- Analog inputs IA01 to IA04
  - IA01: terminal I7
  - IA02: terminal I8

– IA03: terminal I11
– IA04: terminal I12
- Analog output QA01
- Actual value … QV> of another function block

Displaying the parameter set in the PARAMETERS menu

- + Access enabled
- – Access disabled

Coils

ST01EN, enable coil.

Memory requirement of the function block

The set cycle time function block requires 24 bytes of memory plus 4 bytes per constant on the function block input.

Function of the set cycle time function block

The function block is used to define a fixed cycle time.

→ The function block must be enabled so that it can work. Coil ST01EN is active. If coil ST01EN is not active, the entire function block is deactivated and reset.

Actual cycle time is less than the set cycle time:

If the maximum cycle time present is less than the set cycle time, the set cycle time is constant.

Actual cycle time is greater than the set cycle time:

If the maximum cycle time present is greater than the set cycle time, the set cycle time has no effect.

l

Attention!
The shorter the cycle time, the faster the control and regulation process.

Set as small a value for the set cycle time as possible. The processing of the function blocks, reading of the inputs and writing of outputs is only carried out once every cycle. Exception: All function blocks that are processed irrespective of the controller.

Timing relay

MFD-Titan provides 32 timing relays from T 01 to T 32.

A timing relay is used to change the switching duration and the make and break times of a switching contact. The delay times can be configured between 5 ms and 99 h 59 min.

Wiring a timing relay

You integrate a timing relay into your circuit in the form of a contact and coil. The function of the relay is defined via the parameter display. The relay is started via the trigger coil T..EN and can be selectively reset via the reset coil T..RE. The actual timeout running can be stopped via the third coil T..ST.

→ To prevent unpredictable switching states, use each coil of a relay once only in the circuit diagram.

```
I 01----------------------------[ T 02EN
I 02----------------------------[ T 02RE
T 02Q1--------------------------[ Q 01
I 03----------------------------[ T 02ST
```

Figure 132: MFD-Titan circuit diagram with timing relay.

```
T 02 X  M:S +
 >I1
 >I2
  QV>
```

Parameter display and parameter set for timing relay:

T 02	Timing relay function block number 02
X	On-delayed mode
M:S	Time range Minute: Seconds
+	Appears in the parameter display
>I1	Time setpoint value 1
>I2	Time setpoint value I2 (on a timing relay with 2 setpoint values)
QV>	Timed-out actual time in RUN mode

The parameter display for a timing relay is used to modify the switching function, time base or setpoint times and enable the parameter display.

Timing relay modes

Parameter	Switch function
X	On-delayed switching
?X	On-delayed with random time range
■	Off-delayed switching
?■	Off-delayed with random time range
X■	On and off delayed
□	Off-delayed with random time range, setpoint retriggerable
?□	Off-delayed with random time range, retriggerable
?X■	On and off delayed switching with random time, 2 time setpoints
⎍	Single-pulse switching
⎍⎍	Switch with flashing, synchronous, 2 time setpoint values
⎍⎍	Switch with flashing, asynchronous, 2 time setpoint values

Time range

Parameter	Time range and setpoint time	Resolution
S 000.000	Seconds, 0.005 to 999.995 s for constants and variable values	5 ms
M:S 00:00	Minutes: Seconds 00:00 to 99:59 only for constants and variable values	1 s
H:M 00:00	Hours: Minutes, 00:00 to 99:59 only for constants and variable values	1 min.

→ Minimum time setting:
0.005 s (5 ms).

If a time value is less than the MFD cycle time, the elapsed time will only be recognised in the next cycle.

Inputs

The function block inputs >I1 and >I2 can have the following operands:

- Constants
- Markers MD, MW, MB
- Analog inputs IA01 to IA04
  - IA01: terminal I7
  - IA02: terminal I8
  - IA03: terminal I11
  - IA04: terminal I12
- Analog output QA01
- Actual value … QV> of another function block

Outputs

Actual value …QV>

The actual value …QV> can be assigned the following operands:

- Markers MD, MW, MB
- Analog output QA01

Variable setpoint values

Behaviour of the setpoint value when variable values are used.

- Variable values can be used.
- Variable values must be transferred using operands.
- With the time base “s” the value is accepted as a “value in ms”.
- The last position is rounded up to a zero or five.
- With the time base “M:S” the value is accepted as a “value in s”.
- With the time base “H:M:” the value is accepted as a “value in M (minutes)”.

→ The delay times are the same as described for the constants.

Example:
Time base “s”
The operand has the value 9504.
The time value is 9.50 s.
Operand value 45507
The time value is 45.510 s.

Time base “M:S”
The operand has the value 5999.
The time value is 99 min, 59 s: This is the maximum value.

Time base “H:M”
The operand has the value 5999.
The time value is 99 h, 59 min.

Displaying the parameter set in the PARAMETERS menu

- + Access enabled
- – Access disabled

Contacts

T 01Q1 to T 32Q1

Coils

- T 01EN to T 32EN: trigger coil;
- T 01RE to T 32RE: reset coil;
- T 01ST to T 32ST: stop coil.

Memory requirement of the timing relay

The time relay function block requires 48 bytes of memory plus 4 bytes per constant on the function block input.

Retention

Timing relays can be operated with retentive actual values. The number of retentive timing relays can be selected in the SYSTEM → RETENTION menu.

If a timing relay is retentive, the actual value is retained when the operating mode changes from RUN to STOP as well as when the power supply is switched off.

If MFD is started in RUN mode, the timing relay operates with the retentively saved actual value. The state of the trigger pulse must correspond to the function of the timing relay.

1 signal when:

- on-delayed,
- single pulse,
- flashing.

0 status for offdelayed.

Function of the timing relay function block

Timing relay, on-delayed with and without random switching

Random switching

The contact of the timing relay switches randomly within the setpoint value range.

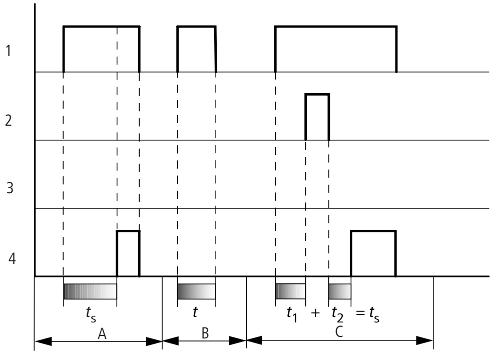


Figure 133: Signal diagram of timing relay, on-delayed (with and without random switching)

1: trigger coil T..EN

2: stop coil T..ST

3: reset coil T..RE

4: switching contact (make contact) T..Q1

t_S: setpoint time

- Range A:
  The set time elapses normally.
- Range B:
  The entered setpoint does not elapse normally because the trigger coil drops out prematurely.
- Range C:
  The Stop coil stops the time from elapsing.

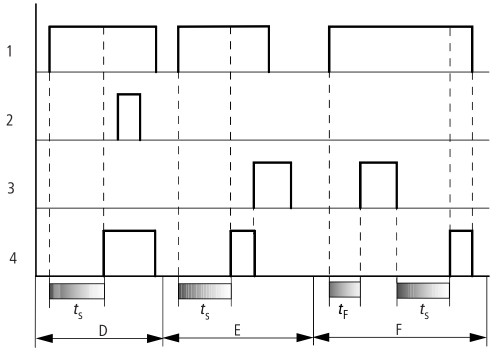


Figure 134: Signal diagram of timing relay, on-delayed (with and without random switching)

- Range D:
  The Stop coil is inoperative after the time has elapsed.
- Range E:
  The Reset coil resets the relay and the contact.
- Range F:
  The Reset coil resets the time during the timeout. After the Reset coil drops out, the time elapses normally.

Timing relay, off-delayed with and without random switching

Random switching, with and without retriggering
The contact of the timing relay randomly switches within the set value range.

Retriggering
When the time is running and the trigger coil is reactivated or deactivated, the actual value is reset to zero. The set time of the timing relay is timed out once more.

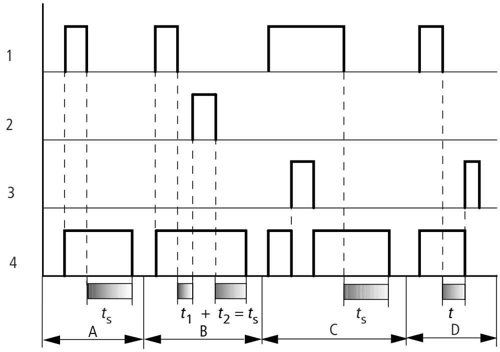


Figure 135: Signal diagram of offdelayed timing relay (with/without random switching, with/without retriggering)

1: trigger coil T..EN

2: stop coil T..ST

3: reset coil T..RE

4: switching contact (make contact) T..Q1

t_S: setpoint time

- Range A:
  The time elapses after the trigger coil is deactivated.
- Range B:
  The Stop coil stops the time from elapsing.
- Range C:
  The Reset coil resets the relay and the contact. After the Reset coil drops out, the relay continues to work normally.
- Range D:
  The Reset coil resets the relay and the contact when the function block is timing out.

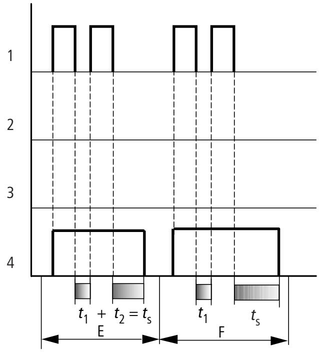


Figure 136: Signal diagram of offdelayed timing relay (with/without random switching, with/without retriggering)

- Range E:
  The Trigger coil drops out twice. The set time t_s consists of t_1 plus t_2 (switch function not retriggerable).
- Range F:
  The Trigger coil drops out twice. The actual time t_1 is cleared and the set time t_s elapses completely (retriggerable switch function).

Timing relay, on-delayed and off-delayed with and without random switching

Time value >I1: On-delayed time
Time value >I2: Off-delayed time

Random switching
The contact of the timing relay switches randomly within the setpoint value range.

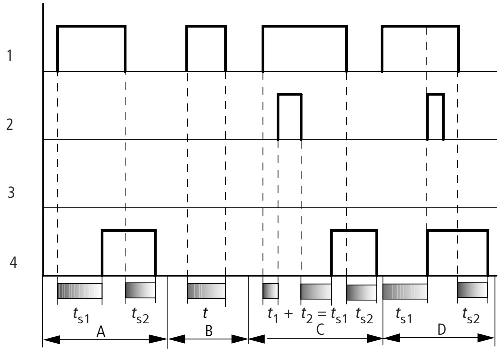


Figure 137: Signal diagram of timing relay, on and off-delayed 1

1: trigger coil T..EN

2: stop coil T..ST

3: reset coil T..RE

4: switching contact (make contact) T..Q1

t_{s1}: pick-up time

t_{s2}: drop-out time

- Range A:
  The relay processes the two times without any interruption.
- Range B:
  The trigger coil drops out before the on-delay is reached.
- Range C:
  The Stop coil stops the timeout of the on-delay.
- Range D:
  The stop coil has no effect in this range.

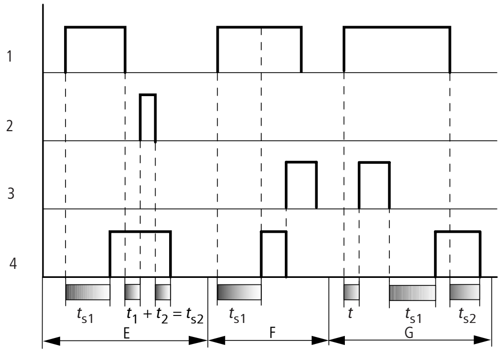


Figure 138: Signal diagram of timing relay, on and off-delayed 2

- Range E:
  The Stop coil stops the timeout of the off-delay.
- Range F:
  The Reset coil resets the relay after the on delay has elapsed
- Range G:
  The Reset coil resets the relay and the contact whilst the on delay is timing out. After the Reset coil drops out, the time elapses normally.

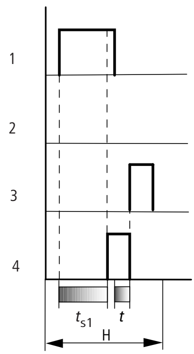


Figure 139: Signal diagram of timing relay, on and off-delayed 3

- Range H:
  The Reset signal interrupts the timing out of the set time.

Timing relay, single pulse

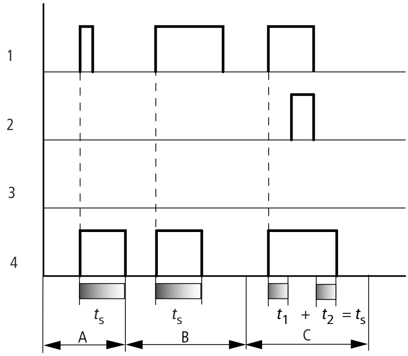


Figure 140: Signal diagram of timing relay, single pulse 1

1: trigger coil T..EN

2: stop coil T..ST

3: reset coil T..RE

4: switching contact (make contact) T..Q1

- Range A:
  The trigger signal is short and is lengthened
- Range B:
  The Trigger signal is longer than the set time.
- Range C:
  The Stop coil interrupts the timing out of the set time.

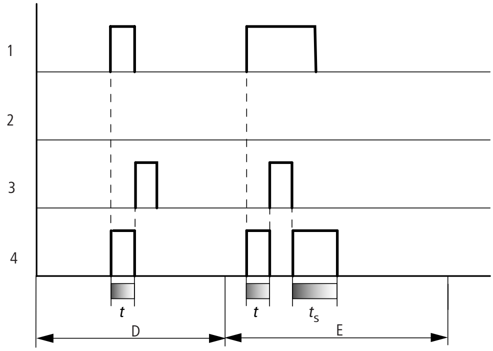


Figure 141: Signal diagram of timing relay, single pulse 2

- Range D:
  The Reset coil resets the timing relay.
- Range E:
  The Reset coil resets the timing relay. The Trigger coil is still activated after the Reset coil has been deactivated and the time is still running.

Timing relay, synchronous and asynchronous flashing

Time value >I1: Pulse time
Time value >I2: Pause time

Synchronous (symmetrical) flashing: >I1 equal >I2
Asynchronous flashing: >I1 not equal >I2

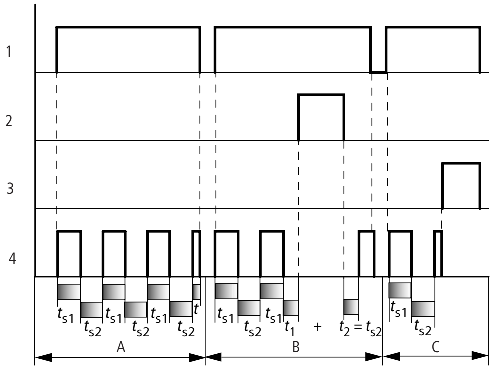


Figure 142: Signal diagram of timing relay, synchronous and asynchronous flashing

1: trigger coil T..EN

2: stop coil T..ST

3: reset coil T..RE

4: switching contact (make contact) T..Q1

- Range A:
  The relay flashes for as long as the Trigger coil is activated.
- Range B:
  The Stop coil interrupts the timing out of the set time.
- Range C:
  The Reset coil resets the relay.

Value limitation

MFD-Titan provides 32 value limitation function blocks VC01 to VC32. The value limitation function block allows you to limit values. You can define an upper and lower limit value. The function block will then only output values within these limits.

Wiring of a value limitation function block

You can integrate a value limitation function block into your circuit as a coil.

```
M 48----------------------------Ĕ VC27EN
```

Figure 143: MFD-Titan circuit diagram with VC value limitation

```
VC27        +
 >I1
 >SH
 >SL
 QV>
```

Parameter display and parameter set for the VC function block:

VC27	VC value limitation function block number 27
+	Appears in the parameter display
>I1	Input value
>SH	Upper limit value
>SL	Lower limit value
QV>	Output value limited

Inputs

The function block inputs >I1, >SH and >SL can be assigned the following operands:

- Constants
- Markers MD, MW, MB
- Analog inputs IA01 to IA04
  - IA01: terminal I7
  - IA02: terminal I8
  - IA03: terminal I11
  - IA04: terminal I12
- Analog output QA01
- Actual value … QV> of another function block

Output

The function block output QV> can be assigned the following operands:

- Markers MD, MW, MB
- Analog output QA01

Value range for inputs and outputs

		Value range
>I1	Input value	–2 147 483 648 to +2 147 483 647
>SH	Upper limit value	
>SL	Lower limit value	
QV>	Output value	

Displaying the parameter set in the PARAMETERS menu

- + Access enabled
- – Access disabled

Coil

VC01EN to VC32EN, function block enable

Memory requirement of the value limitation function block

The value limitation function block requires 40 bytes of memory plus 4 bytes per constant on the function block input.

Function of the value limitation function block

→ The function block must be enabled so that it can work. Coil VC..EN is active. If coil VC..EN is not active, the entire function block is deactivated and reset. The output value is set to zero.

The value is accepted at input VC...I1 if the enable coil is active. If the value is greater than the upper limit value or less than the lower limit value, the respective limit value is output at VC..QV.

Example with timing relay and counter function block

A warning light flashes when the counter reaches 10. In the example, both function blocks C 01 and T 01 are wired.

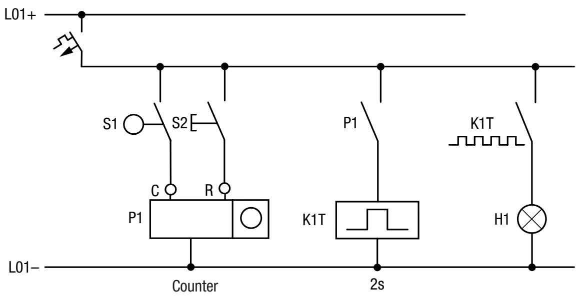


Figure 144: Hardwiring with relays

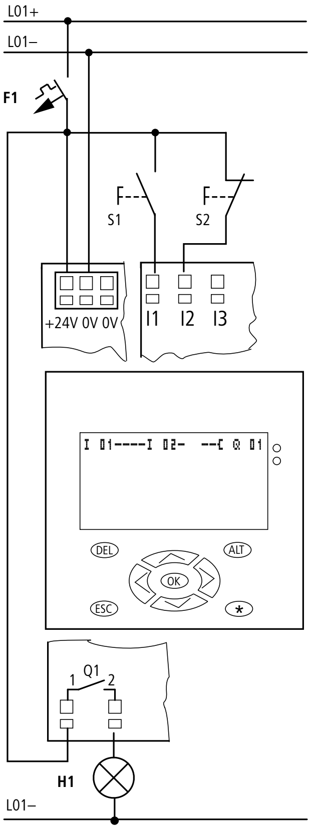


Figure 145: Wiring with the MFD-Titan

```
I 05------------------------------[ C 01C
I 06------------------------------[ C 01RE
C 01------------------------------[ T 01EN
T 01Q1----------------------------[ Q 01
```

Figure 146: MFD-Titan wiring and circuit diagram

Entering function block parameters from the circuit diagram.

You can access the parameter entry from the contact as well as from a coil.

▶ Enter the circuit diagram up to C 01 as a coil.

C 01C is the counter coil of the counter 01 function block.

```
I 05----------------------------[ C 01C
I 06----------------------------[ C 01RE
C 01----------------------------[ T 01EN
T 01Q1--------------------------[ Q 01
```

Figure 147: MFD-Titan wiring and circuit diagram

▶ Keep the cursor on the number.

▶ Press the **OK** button.

→ If the cursor is on the contact number, MFD-Titan will call up the parameter display when you press **OK**.

```
C 01          +
 >SH +10
 >SL
 >SV
```

The first part of the parameter set of a counter is displayed.

▶ Move the cursor > over the + character to the value input behind >SH:

– >SH means: function block input upper counter setpoint value

– The + character means that the parameters of this timing relay can be modified using the PARAMETERS menu.

▶ Change the upper counter setpoint to 10:

– Use < > to move the cursor onto the tens digit.

– Use ∧∨ to modify the value of the digit.

▶ Press **OK** to save the value and **ESC** to return to the circuit diagram.

→ MFD-Titan has specific parameter displays for the function blocks. The meaning of these parameters is explained under each function block type.

▶ Enter the circuit diagram up to contact T 01 of the timing relay. Set the parameters for T 01.

```
T 01 ⊥⊥  S   +
 >I1 002.000
 >I2 002.000
  QV>
```

The timing relay works like a flashing relay. The MFD-Titan symbol for the flashing relay is ⊥⊥. The function is set on the top right beside the number in the parameter display.

The time base is set to the right of the "flashing" function. Leave the time base set to S for seconds.

- ▶ Move the cursor to the right over the + character in order to input the time setpoint value >I1.

If the same setpoint value is input for >I1 and >I2, the timing relay functions as a synchronous flasher.

The + character means that the parameters of this timing relay can be modified using the PARAMETERS menu.

- ▶ Confirm the value input with **OK**.
- ▶ Press **ESC** to leave circuit diagram entry.
- ▶ Complete the circuit diagram.
- ▶ Test the circuit diagram using the power flow display.
- ▶ Switch MFD-Titan to RUN mode and return to the circuit diagram.

Each parameter set can be displayed using the power flow display for the circuit diagram.

- ▶ Move the cursor onto C 01 and press **OK**.

```
C 01         +
 >SL
 >SV
  QV>+0
```

The parameter set for the counter is displayed with actual and setpoint values.

- ▶ Move the cursor ∨ downwards until you see the value QV>.

```
C 01         +
 >SL
 >SV
  QV>+1
.. C_  ..  ..
```

- ▶ Switch the input I5. The actual value changes.

On the display C_ indicates that the counting coil is actuated.

If the actual and upper setpoint values of the counter are the same, the timing relay switches the warning light on and off every 2 seconds.

```
T 01 ⫫  S   +
 >I1 001.000
 >I2
  QV> 0.550
.. EN..
```

Doubling the flashing frequency:

▶ Select the power flow display T 01 and change the constant of the setpoint time to 001.000.

When you press **OK**, the warning light will flash at twice the frequency.

On the display EN indicates that the enable coil is actuated.

Setpoint value settings with constants can be modified via the PARAMETERS menu.

→ The actual value is only displayed in RUN mode. Call up the parameter display for this via the power flow display or PARAMETERS menu.

5 Visualization with MFD-Titan

In the following description, the term “visualization” is used for the display and operator function.

Whilst you can enter the circuit diagram via EASY-SOFT as well as directly on the device, all the visualization functions can only be programmed using EASY-SOFT-PRO. The visualization functions can then be loaded from there to the MFD-Titan or onto a memory card (download function).

This chapter uses the examples supplied (from version 6.0) to describe the basic methods of designing visualization systems using EASY-SOFT-PRO. The EASY-SOFT Help system provides a more detailed description of the software.

Screens

EASY-SOFT-PRO manages the visualization elements in screens. These elements can be inserted in the screens and are called screen elements in the following description. You can use up to 255 screen elements inside one screen. Due to the memory allocation of the system, it is more advisable to use several screens (→ Section “Memory division”, Page 280).

The following screen element are available:

- Static text,
- Bit display,
- Date and time,
- Bitmap,
- Numerical value,
- Value entry,
- Message text.

The individual screen elements are explained in the examples.

Memory division

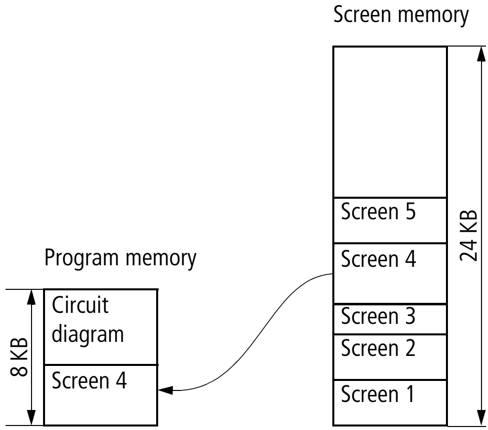


Figure 148: Memory division MFD

The maximum size of the program memory is 8 KByte. This memory area is used to store the circuit diagram and also reserves enough space for displaying the largest screen.

The screen memory has a total memory capacity of 24 KByte, which is used in this memory area to store all the screens created.

It should therefore be ensured that screens have the lowest possible memory requirement so that enough space is available in the program memory for the largest screen and the circuit diagram. If a circuit diagram is not required, the memory requirement of the largest screen must not exceed the 8 KByte limit.

→ The EASY-SOFT-PRO status bar indicates the available memory, the available screen memory and the required memory for the active screen element.
If the available memory is exceeded the indicator will turn red..

Western European character table

Code	Meaning	Code	Meaning	Code	Meaning	Code	Meaning
0		28		56	8	84	T
1		29		57	9	85	U
2		30		58	:	86	V
3		31		59	;	87	W
4		32	Blank	60	<	88	X
5		33	!	61	=	89	Y
6		34	"	62	>	90	Z
7		35	#	63	?	91	[
8		36	$	64	@	92	\
9		37	%	65	A	93	]
10		38	&	66	B	94	^
11		39	'	67	C	95	_
12		40	(	68	D	96	`
13		41	)	69	E	97	a
14		42	*	70	F	98	b
15		43	+	71	G	99	c
16		44	,	72	H	100	d
17		45	-	73	I	101	e
18		46	.	74	J	102	f
19		47	/	75	K	103	g
20		48	0	76	L	104	h
21		49	1	77	M	105	i
22		50	2	78	N	106	j
23		51	3	79	O	107	k
24		52	4	80	P	108	l
25		53	5	81	Q	109	m
26		54	6	82	R	110	n
27		55	7	83	S	111	o

Code	Meaning	Code	Meaning	Code	Meaning	Code	Meaning
112	p	141	ì	170	¬	199	Ã
113	q	142	Ä	171	½	200	+
114	r	143	Å	172	¼	201	+
115	s	144	É	173	¡	202	-
116	t	145	æ	174	«	203	-
117	u	146	Æ	175	»	204	¦
118	v	147	ô	176	¦	205	-
119	w	148	ö	177	¦	206	+
120	x	149	ò	178	¦	207	€
121	y	150	û	179	¦	208	ð
122	z	151	ù	180	¦	209	Đ
123	{	152	ÿ	181	Á	210	Ê
124	|	153	Ö	182	Â	211	Ë
125	}	154	Ü	183	À	212	È
126	~	155	ø	184	©	213	i
127	¦	156	£	185	¦	214	Í
128	Ç	157	Ø	186	¦	215	Î
129	ü	158	×	187	+	216	Ï
130	é	159	ƒ	188	+	217	+
131	â	160	á	189	¢	218	+
132	ä	161	í	190	¥	219	¦
133	à	162	ó	191	+	220	_
134	å	163	ú	192	+	221	¦
135	ç	164	ñ	193	-	222	Ì
136	ê	165	Ñ	194	-	223	¯
137	ë	166	ª	195	+	224	Ó
138	è	167	º	196	-	225	ß
139	ï	168	¿	197	+	226	Ô
140	î	169	®	198	ã	227	Ò

Code	Meaning	Code	Meaning	Code	Meaning	Code	Meaning
228	õ	235	Ù	242	=	249	¨
229	Õ	236	ý	243	¾	250	·
230	µ	237	Ý	244	¶	251	^{1}
231	þ	238	¯	245	§	252	^{3}
232	Þ	239	´	246	÷	253	^{2}
233	Ú	240	-	247	¸	254	¦
234	Û	241	±	248	°	255	

Screen overview

The first time the visualization function is called the screen overview will appear with the following tabs:

- Screens,
- Passwords,
- Languages,
- Screen activation.

Screens tab

The Screens tab is used for entering the screen name, the start screen and password protection (if required) for the screens concerned.

Passwords tab

The Passwords tab enables you to define up to three passwords and assign a logout time with each one for closing the screen.

Languages tab

The Languages tab is used for entering different languages that are to be used. You can then define texts in these languages for all the text elements of the visualization program.However, only one download language can be defined for downloading to the device. This is also defined here and can be modified quickly when the devices are commissioned.

You can export/import the languages to/from a Microsoft Excel spreadsheet, enabling texts to be sent off in this way for external translation.

Screen activation tab

The Screen activation tab allows you to define associated variables for activating the screen. The variables available depend on the variable type concerned. You can use markers (Byte, Word, DWord), analog inputs and outputs (word) or function block inputs/outputs (DWord) as associated variables. In the List of activation values you define whether the screen is to be activated by the associated variable and the value at which this is done.

If the Force screen change check box is activated, the MFD device will immediately abort every job started and activate this mask when the value of the associated variable triggers it. This could mean, for example, that value entries, macros for processing functions after pressing an operator button, and also a password entries are aborted. This can be useful for outputting appropriate alarm messages.

→ Unsaved entries will therefore be lost if a screen is activated that is associated with a particular event and the Force screen change option is active!
In this case, while the associated variable is set by the program, it is also not possible to carry out a screen change via the keypad.

In the List of activation values you can select whether the screen is to be activated by the set variable and at which value.

Screen editor

You process the individual screens in the Screen editor. This provides the following different screen elements for selection:

- Static text,
- Bit display,
- Date and time,
- Bitmap,
- Numerical value,
- Value entry,
- Message text.

These screen elements are fully described in the following sections.

The Project info, Program info and Screen info tabs are displayed if there are no screen elements activated. The relevant tabs for a particular screen element are shown if it is inserted in the screen.

Static text

Static text is a simple text display that is shown as soon as the screen is active. Up to 16 normal font size characters can fit in one text field and up to 4 text lines can be arranged in rows. The number of characters per text field and the number of displayable lines is halved if double font characters are used. The Static text is assigned to a language in the Language selection field. The languages available are defined in the Screen overview.

Programming in EASY-SOFT-PRO

▶ Hold down the left mouse button to drag the Static text screen element onto the screen.

▶ Place the mouse over the Static text screen element, hold down the left mouse button and drag the screen element to the required position.

▶ Enter the required text in the Static text tab and select the language to which the text is to be assigned.

You define the available languages in the Languages tab of the Screen overview. These languages apply to the text elements of all screens.

The height of the element frame adjusts itself to the font size – either normal or double font size. You can change the font size by activating the context menu (right mouse button) or by dragging the selection handle at the bottom edge of the element frame concerned. When changing to double font size, ensure that there is sufficient space available underneath the element.

The width of the element can be adjusted as required. You can set the size required by holding down the left mouse button and dragging a selection handle of the element frame. When applicable, ensure that the elements provide sufficient space in all languages. This can be checked easily in the screen overview.

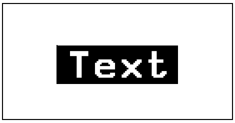
The following examples explain the function of Static text:

Example program 1– different display formats
The program consists of four screens in which the static text is displayed in different ways. The individual screens can be selected with the Cursor buttons ⌃ and ⌄. The screen change was defined in the Button editor (→ Section "Button editor", Page 333).

Standard path: C:\Programme\Moeller Software\EASY-SOFT 5 [Pro]\Samples\AWB2528-1480\

Program: Statischer_Text_BspProg_01.e40

Screen 1
Screen 1 displays a static text in normal font size.



Screen 2

Screen 2 displays a static text enlarged.

Screen 3

Screen 3 shows three static texts in different font sizes.

Screen 4

Screen 4 shows three static texts arranged in rows.

Example program 2 – different display formats with password request for one screen

The program consists of four screens in which the static text is displayed in different ways. The fourth screen contains a password request. The individual screens can be selected with Cursor buttons ∧ and ∨. The screen change was defined in the Button editor (→ Section "Button editor", Page 333).

The defined password is requested when exiting screen 4. In this case, it is the number 2 which has to be confirmed with **OK**. Further scrolling through the screens cannot be done without entering the correct password. You can leave the password request by pressing **ESC**, which, however, will only move you back to the previous screen.

Standard path: C:\Programme\Moeller Software\EASY-SOFT 5 [Pro]\Samples\AWB2528-1480\

Program: Statischer_Text_BspProg_02.e40

Screen 1

Screen 1 displays a static text in normal font size.

Screen 2

Screen 2 displays a static text enlarged.

Screen 3

Screen 3 shows three static texts in different font sizes.

```
Four
static texts
arranged
in rows!!
```

Screen 4

Screen 4 shows three static texts arranged in rows.

Example program 3 – different display formats with screen activation via a counter

The program consists of four screens in which the static text is displayed in different ways. The individual screens are activated via counter C01. The counter's SH upper switch value is 5. The activation pulse for activating the screens is supplied by timing relay T01. This generates the counter values 0 to 5. Screen 1 is activated with counter value 1, screen 2 with counter value 2 etc. Once the counter's upper switch value has been reached, timing relay T02 is started which resets the counter. This therefore produces a continuous loop. Timing relay T02 provides a reset delay for the counter.

Standard path: C:\Programme\Moeller Software\EASY-SOFT 5 [Pro]\Samples\AWB2528-1480\

Program: Statischer_Text_BspProg_03.e40

Screen 1

Screen 1 displays a static text in normal font size.

Screen 2

Screen 2 displays a static text enlarged.

Screen 3

Screen 3 shows three static texts in different font sizes.

Screen 4

Screen 4 shows four static texts arranged in rows.

Bit display

The bit display screen element has an input that can be associated with a Boolean variable. The signal status of this variable changes the bit display screen element from a full screen to a frame in the MFD display.

Programming in EASY-SOFT-PRO

▶ Hold down the left mouse button and drag the Bit display screen element onto the screen.

▶ Place the mouse over the element, hold down the left mouse button and position it as required.

The height and width of the element frame is variable and can be adjusted accordingly by enlarging or reducing the element frame vertically, horizontally or diagonally. This is done by dragging a selection handle of the element frame with the left mouse button held down.

Associated variable tab: defines the Boolean variable for activating the bit display.

Visibility tab: option for making the element invisible by means of an associated variable.

→ The bit display is always output as a solid image. The invisible function is the only option available. If there is an overlap of elements the bit display that was inserted first is positioned at the back and the last one at the front. This can be modified for the activated screen element in the toolbar (Move to foreground/background buttons).

→ Large bit displays require a large amount of processor capacity and should be avoided in time-critical applications.

Example program 1 – associated variable and visibility

The program consists of five screens that can be selected with the Cursor buttons ⌃ and ⌄. The screen change was defined in the Button Editor (→ Section "Button editor", Page 333).

The screens show examples of the use of both associated variable and visibility elements.

The circuit diagram uses six on-delayed timing relays that activate outputs Q1 to Q4 and LED 3 in succession after an elapsed time. This operation is run in a continuous loop since timing relay T06 resets all timing relays after a set time.

Standard path: C:\Programme\Moeller Software\EASY-SOFT 5 [Pro]\Samples\AWB2528-1480\

Program: Bitanzeige_BspProg_01.e40

Screen 1

Screen 1 illustrates the activation of the bit display exclusively via the associated variable. The first screen contains four bit display elements. These are activated in succession via Boolean operands Q1 to Q4. A make contact bit logic is selected for setting the bit display elements to the state of the corresponding outputs (on/off).

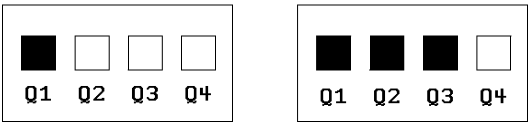


Figure 149: Bit display and static text

Screen 2

Screen 2 illustrates the activation of the bit display with overlaid static text, exclusively via the associated variable. Screen 2 contains four bit display elements. These are activated in succession via Boolean operands Q1 to Q4. A make contact bit logic is selected for setting the bit display elements to the state of the corresponding outputs (on/off). The bit display is overlaid partly with static texts.

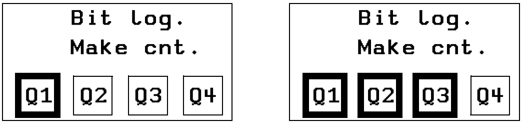


Figure 150: Bit display with overlaid static text

Screen 3

Screen 3 illustrates the possible uses of the Visibility tab for the bit display. In this screen a bit display element has been enlarged to the full size of the screen. It is activated via the Boolean operand Q1, with a make contact bit logic so that the bit display element takes on the state of the output Q1 (on/off).

The bit display is made invisible via the Boolean operand Q3. A make contact bit logic is also used here so that invisibility is activated for as long as Q3 in the On state. When the Reset pulse is present, only the frame of the bitmap is visible (state of the bitmap is "off", bitmap is visible). The display is overlaid partly with static text.

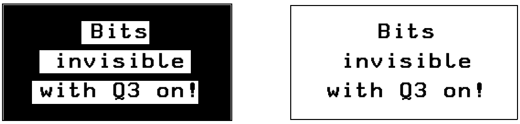


Figure 151: Bit display visible/invisible

Screen 4

Screen 4 illustrates the activation of the bit display (negated) with overlaid static text, exclusively via the associated variable. This screen contains four bit display elements. These are activated in succession via Boolean **operands** Q1 to Q4. A break contact bit logic is selected here. The bit display is overlaid partly with static texts.

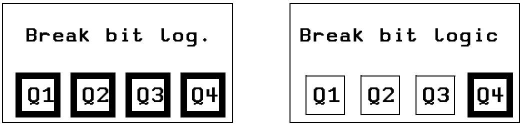


Figure 152: Break contact logic bit display with overlaid text

Screen 5

Screen 5 illustrates the possible uses of the Visibility tab for the bit display (negated) with overlaid static text. In this screen a bit display element has been enlarged to the full size of the screen. This is activated via the Boolean operand Q1 with the break contact bit logic. In this way, the bit display takes on the opposite status of output Q1. In other words, when Q1 is On, only the frame of the bit display is visible. However, the visibility of the bit display is only activated with the Boolean **operand** Q3 as the break contact bit logic is selected. The bit display is overlaid partly with static text.

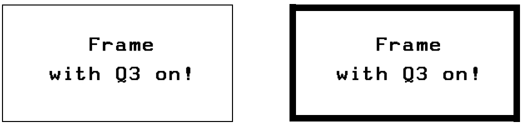


Figure 153: Bit display with frame

Example program 2 – bit display with automatic screen change

This program is a copy of the program Bitanzeige_BspProg_01.e40. The only difference is that the screens are activated in succession automatically. (screen change in the Screen overview → Screen activation tab → Activate Yes).
The program consists of five screens containing bit display elements.

Standard path: C:\Programme\Moeller Software\EASY-SOFT 5 [Pro]\Samples\AWB2528-1480\

Program: Bitanzeige_BspProg_02.e40

Date and time

This screen element shows the date and time of the MFD-real-time clock on the display. You can also select for this international display formats according to the ten easy system languages plus the USA display format.

Programming in EASY-SOFT-PRO:

▶ Hold down the left mouse button and drag the Date and time screen element onto the screen.

▶ Place the mouse over the screen element, hold down the left mouse button and position it as required.

The height of the element frame depends on the font size used. Three sizes are possible:

- Normal font,
- Double font and
- Quadruple font.

You can change the font size by activating the context menu (right mouse button) or by dragging the selection handle at the bottom edge of the screen element frame concerned. When increasing the font size, ensure that there is sufficient space available underneath the element.

The maximum width of the element is limited and depends on the font size. This also determines the display a format.

Date formats tab

Select the language and one of the four formats available in the Date formats tab. These are independent of the language set on the MFD device.

Visibility tab

The visibility tab provides the option of making the screen element invisible by means of an associated variable.

Example program 1 – different display formats and invisibility option for a screen

The program consists of eight screens that can be selected via the Cursor buttons ⌃ and ⌄. The screen change is defined in the Button editor (→ Section "Button editor", Page 333).
The language setting for all eight screens is "Deutsch". The first six screens show various data and time display formats. The seventh screen shows all four formats at the same time and the eighth screen illustrates the invisibility function.

Standard path: C:\Programme\Moeller Software\EASY-SOFT 5 [Pro]\Samples\AWB2528-1480\

Program: Datum_und_Zeit_BspProg_01.e40

Screen 1:

Format: DD.MM:YYYY
Display in double font size, therefore only day and month visible in display!

Screen 2:

Format: DD.MM.YYYY
Display in normal font size, therefore year display also visible!

Screen 3:

Format: DD.MM.YY HH:MM
Display in normal font size, display of time in hours and minutes.

Screen 4:

Format: HH:MM
Display of hour and minute in double font size.

```
WED 01.10.03
```

Screen 5:
Format: WD DD.MM.YY
Display of weekday and date in normal font size.

Screen 6:
Format: HH:MM
Display of hour and minute in single font size.

Screen 7:
Display of different formats possible in one screen. Display in normal font size.

```
Invisible
via I1!
01.10.2003
```

Screen 8:
Display invisible if I1 actuated. (make contact bit logic!)

Example program 2 – different display formats with automatic screen change
This program is a copy of the program Datum_und_Zeit_BspProg_01.e40 except that the screen change here is automatic. The program consists of eight screens that can be activated in succession via the counter C01. The pulse signals are generated with timing relay T01. This produces the counter values 0 to 9. Counter value 1 activates screen 1, counter value 2 activates screen 2 etc. If upper switch value SH = 9, the counter resets itself. This therefore produces a continuous loop. The language setting for all eight screens is “Deutsch”.

Standard path: C:\Programme\Moeller Software\EASY-SOFT 5 [Pro]\Samples\AWB2528-1480\

Program: Datum_und_Zeit_BspProg_02.e40

Example program 3 – different country settings
The program consists of 11 screens that can be activated automatically via the counter C01. The pulse signals are generated with timing relay T01. This produces the counter values 0 to 12. Counter value 1 activates screen 1, counter value 2 activates screen 2 etc. If upper switch value SH = 12, the counter resets itself. This therefore produces a continuous loop.

Each screen contains a different country setting. However, this same WD DD MM YY format is shown.

Standard path: C:\Programme\Moeller Software\EASY-SOFT 5 [Pro]\Samples\AWB2528-1480\

Program: Datum_und_Zeit_BspProg_03.e40

Screen 4	Screen 7
Country setting Spanish MI 01/10/03	Country setting Portuguese QU 01-10-03

Figure 154: Date display with different country settings

Bitmap

The bitmap mask element makes it possible to display graphics in the MFD display that you have made yourself or have purchased. Display and visibility can change during operation. To do this, you need to associate the bitmap graphics with Boolean variables.

EASY-SOFT-PRO supports the following bitmap graphic formats:

- Windows Bitmap format (bmp),
- JPEG format (jpg),
- Tiff format (tif) and
- Icons (ico).

The formats are converted to monochrome format using suitable conversion procedures and then saved accordingly in the program. The size and position of the bitmap graphic can be modified later in the Mask Editor.

→ Bitmap graphics require a large amount of memory. Overlaid bitmaps are also stored fully in the memory. The more bitmaps are used use, the more the cycle time will be increased considerably due to the additional memory requirement. This may possibly lead to program malfunctions (e.g. loss of count pulses).

Black and white graphics should be used ideally. These should be between 16 × 16 and 32 × 32 pixels in size.

Programming in EASY-SOFT-PRO:

▶ Hold down the left mouse button and drag the Bitmap screen element into the mask.

This will open the Picture File Selection dialog.

For the optimum display of the graphic file select one of the three conversion processes provided in the Conversion Type area. The Preview shows the selected bitmap and the conversion result. EASY-SOFT-PRO always generates the optimum black and white bitmap graphic for display on the monochrome MFD display. You can also support EASY-SOFT-PRO by optimizing the graphic file.

▶ Position the mouse above the screen element and move it to the required position with the left mouse button depressed.

The height and width of the element frame is variable and can be adjusted accordingly by enlarging or reducing the screen element frame vertically, horizontally or diagonally. This is done holding down the left mouse button and dragging a selection handle of the element frame. The side ratios of the original graphic are retained when you use the diagonal zoom function.

Display tab

The Display tab provides the following display formats for the bitmap:

- Flashing (via associated variable),
- Background transparent or covered,
- Inverted

Visibility tab

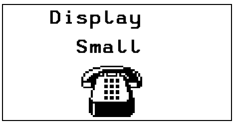
The Visibility tab enables you to make the screen element invisible via an associated variable.

Example program 1 – Bitmap display

The program consists of eight screens that can be selected by the Cursor buttons ∧ and ∨. The screen change was defined in the Button editor (→ Section “Button editor”, Page 333).
This program uses simple examples to explain the bitmap display options available.

Standard path: C:\Programme\Moeller Software\EASY-SOFT 5 [Pro]\Samples\AWB2528-1480\

Program: Bitmap_BspProg_01.e40



Screen 1:

Screen 1 shows a bitmap without any particular features: small display.

Screen 2:

Screen 2 contains a bitmap in large display format. The Bitmap flashing field of the Display tab shows an association via the Boolean operand I1. If I1 is actuated, the Bitmap flashes.

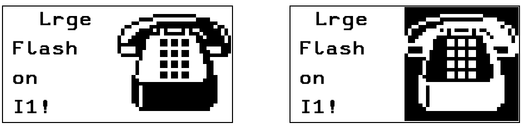
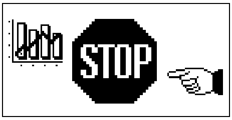
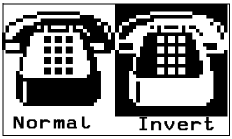


Figure 155: Flashing bitmap



Screen 3:

Screen 3 contains three bitmaps that are arranged in different sizes next to each other.



Screen 4:

Screen 4 contains two bitmaps that are arranged in different sizes next to each other. The Invert Bitmap Display check box activates the inverted display.

Screen 5:

Screen 5 contains one bitmap. The Visibility tab controls the bitmap. I1 is used to make the bitmap invisible.

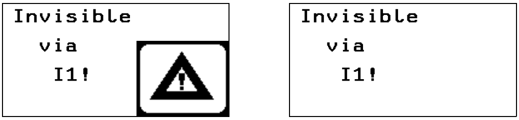
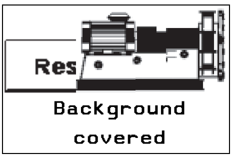


Figure 156: Switch invisible function

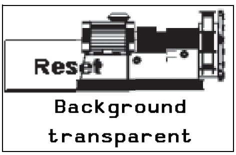
Screen 6:

Screen 6 contains only one bitmap. The Invert Bitmap Display check box is activated in the Display tab. In the Visibility tab the associated variable I2 is defined for switching visibility (break contact bit logic). When I2 is actuated, the bitmap is inverted as shown below.



Screen 7:

Screen 7 contains two bitmaps. Background covered is selected in the Display tab for both bitmaps. However, as the right bitmap covers the left one, the right bitmap is displayed completely.



Screen 8:

Screen 8 contains two bitmaps. The right one covers the left one. Both bitmaps are displayed fully since the background of the right bitmap is set for transparent in the Display tab.

Example program 2 – bitmap display with automatic screen change

The program is a copy of the program Bitmap_BspProg_01.e40 with the difference that the individual screens are displayed automatically in succession. The circuit diagram contains a pulse generator T01 that triggers the output Q3 (display clock signal) and the counter C01. This operation is run in a continuous loop as the counter activates timing relay T02 when its upper limit value SH is reached. This resets counter C01 after a set time. The counter values activate the corresponding individual screens.

Standard path: C:\Programme\Moeller Software\EASY-SOFT 5 [Pro]\Samples\AWB2528-1480\

Program: Bitmap_BspProg_02.e40

Example program 3 – overlaying bitmaps

The program consists of three screens that you can select via the Cursor buttons ⌃ and ⌄. The screen change is defined in the Button editor (→ Section "Button editor", Page 333). This program illustrates the overlaying of bitmaps in a screen. The circuit diagram uses six on-delayed timing relays (T01 to T06), that activate outputs Q1 to Q4 and LED 3 in succession after an elapsed time. This operation is run in a continuous loop since timing relay T06 resets all timing relays after a set time.

Standard path: C:\Programme\Moeller Software\EASY-SOFT 5 [Pro]\Samples\AWB2528-1480\

Program: Bitmap_BspProg_03.e40

Screen 1:

The first screen contains five half-overlaid bitmaps. These are activated via the outputs Q1 to Q4 and marker M01, and appear in the display. The half overlaid bitmaps are activated in succession. The Covered option is selected in the Background area of the Display tab, therefore causing the overlaid part of the bitmap to be covered by the bitmap in front of it. In this way it is possible to create the impression of a moving arrow.

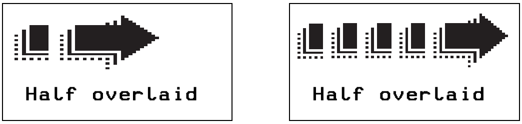


Figure 157: Overlaid bitmaps

Screen 2:
The second screen contains five overlaid bitmaps with each subsequent bitmap being larger than the previous one displayed. The individual bitmaps are made visible in succession via the outputs Q1 to Q4 and the marker M01 (break contact bit logic). The Covered option is selected in the Background area of the Display tab, therefore causing the overlaid part of the bitmap to be covered by the bitmap in front of it. In this way it is possible to create the impression of a moving and expanding arrow.

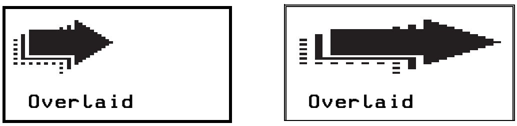


Figure 158: “Expanding” arrow

Screen 3:
The third screen contains five fully overlaid bitmaps. These are made invisible in succession from the top to the bottom via the outputs Q1 to Q4 and the LED 3 (make contact bit logic). The Covered option is selected in the Background area of the Display tab. The overlaid section of the bitmap is therefore not visible.

The first bitmap at the lowest level is permanently activated, making it visible as long as all other bitmaps are invisible. This produces a small moving picture.

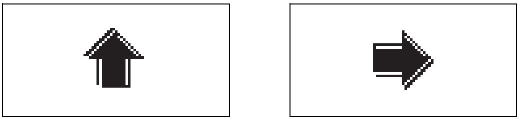


Figure 159: Rotating arrows

Numerical value

This mask element allows you to display untreated or scaled signal states in decimal format.

Value and scaling range

The value range defines the range that is to be displayed. If the values are below or above this range, the MFD activates an underflow or overflow signal.

The scaling range is used for scaling the value range. The lower and upper values of the scaling range are assigned to the lower and upper values of the value range respectively. The MFD displays the value range if a scaling range has not been defined.

Examples:

In order, for example, to display the value range (0 to 255) of a counter as a percentage (0 to 100 %), enter “0” as the minimum value and “255” as the maximum value in the Value range field. Enter “%” as the unit of measure. Activate the Scaling range field and enter “0” as the minimum value and “100” as the maximum value.

In order, for example, to display the value range (0 to 120) of a timing relay in minutes (0 to 2 min), enter “0” as the minimum value and “120” as the maximum value in the Value range field. Enter “min” as the unit of measure. Enter “0” in the activated Scaling range field as the minimum value and “120” as the maximum value.

Programming in EASY-SOFT-PRO:

▶ Hold down the left mouse button and drag the Numerical value screen element into the screen.

▶ Position the mouse above the screen element and move it to the required position with the left mouse button depressed.

The height of the screen element frame depends on the font size selected. Three sizes are available:

- Normal font,
- Double font and
- Quadruple font

You can change the font size by activating the context menu (right mouse button) or by dragging the selection handle at the bottom edge of the element frame concerned.
When increasing the font size, ensure that there is sufficient space available underneath the element.

The width of the screen element frame can be scaled as required by dragging the selection handle on the sides.

Associated variable tab

On the Associated variable tab you define the Boolean variable for activating the numerical value.

Number range/format tab

The Number range/format tab is used for defining the following:

- Value range,
- Unit of measure,
- Scaling range,
- Display change and
- Showing a signed value.

Visibility tab

The Visibility tab enables you to make the screen element invisible by means of an associated variable.

Example – numerical value:

The program consists of nine screens that can be selected via the Cursor buttons ⌃ and ⌄. The screen change is defined in the Button editor (→ Section “Button editor”, Page 333).

Standard path: C:\Programme\Moeller Software\EASY-SOFT 5 [Pro]\Samples\AWB2528-1480\

Program: Zahlenwert_BspProg_01.e40

Screen 1:

This screen shows six simple output formats of the numerical values and provides an overview of the Number range/ format properties. The first screen shows six numerical values. These are associated with the on-delayed timing relay T04 (Associated variable tab) for which a time range of 4 seconds is set. The actual value of the timing relay is provided at its QV output (FB parameter) which is then displayed in the appropriate number format. The variable type of the associated variable is DWord.

Table 11: Numerical value and output formats

Left column	Right column
Numerical value 1 • Value range: 0 to 50000 • Unit: None • Scaling range: None • Always show sign: No	Numerical value 1 • Value range: 0 to 4000 • Unit: None • Scaling range: 0 to 4 • Decimal places: 0 • Always show sign: No
Numerical value 2 • Value range: 0 to 50000 • Unit: ms • Scaling range: None • Always show sign: No	Numerical value 2 • Value range: 0 to 4000 • Unit: s • Scaling range: 0 to 4 • Decimal place: 1 • Always show sign: No
Numerical value 3 • Value range: 0 to 50000 • Unit: ms • Scaling range: None • Always show sign: Yes	Numerical value 3 • Value range: 0 to 4000 • Unit: s • Scaling range: 0 to 4 • Decimal places: 2 • Always show sign: Yes

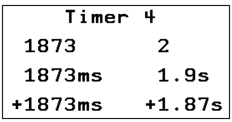


Figure 160: Output formats

Screen 2:

Screen 2 shows a simple example of the visualization of timing relay times using the function block parameter (FB parameter) from the Associated variable tab. The screen contains three numerical values. These are associated with the QV output of timing relay T04.

Table 12: Numerical value – visualization of timing relay times

Numerical value 1:	Start value of the timing relay T04 FB parameter: I2 (zero as input I2 is not assigned with on-delayed relays!)
Numerical value 2:	Setpoint of the timing relay T04 FB parameter: I1 (4000 as input I1 is assigned with the constant 4 s (= 4000 ms) in the function block editor)
Numerical value 3:	Actual value of the timing relay T04 FB parameter: QV (display of value present at the function block output)

In order for the values to be displayed meaningfully, "ms" (milliseconds) is entered in the Unit of measure field in the Number range/format tab.

```
Start Val     0ms
Setpoint  4000ms
Act Val   1452ms
```

Figure 161: Timing relay times

Screen 3:

Screen 3 shows an example of outputting analog values (here IA3) on the display. Note the Scaling range field on the Number range/format tab.

The screen contains three numerical value that output the analog value in different formats. The values are associated with the analog input IA3 (Associated variable tab) which has a value provided as a DWord variable at the QV output (FB parameter).
All numerical values are assigned a value range from 0 to 1 019 as this scan range is defined by the connected potentiometer.

Table 13: Numerical value - analog value output

Numerical value 1:	Display of the analog value (0 to 1 019)
Numerical value 2:	Display of the analog value with the scaling range 0 to 10, two decimal places, unit of measure “V” (Volt)
Numerical value 3:	Display of the analog value with the scaling range –5 to +5, two decimal places, unit of measure “V” (Volt), signed

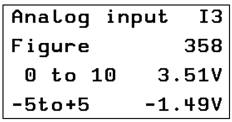


Figure 162: Analog value output

Screen 4:

Screen 4 shows an example of the output of analog values (here IA3) on the display which is an extension of screen 3. For greater simplicity, all the basic settings of the scaling range were taken from screen 3.

Note the Numerical display setting in the Display change area of the Number range/format tab.

Table 14: Numerical value - extended analog value output

Numerical value 1:	Display of the analog value (0 to 1019)
Numerical value 2:	Display of the analog value with the scaling range 0 to 10, two decimal places; unit of measure "V" (Volt); Always show sign: yes; Display change - Detection via: internal limit value comparison; Numerical display: flashing
Numerical value 3:	Display of the analog value with the scaling range –5 to +5, two decimal places, unit of measure "V" (Volt); Always show sign: no; Display change - detection via: internal limit value comparison; Numerical value display: inverted

```
Analog input   I3
Figure          0
Flashing   +0.00V
Invert     -5.00V
```

Figure 163: Extended analog value output

Screen 5:

Screen 5 shows an example of the output of analog values (here IA3) on the display with the visibility option switched via I1 and I2. The settings for this were defined on the Visibility tab. Two numerical values are shown on the screen that output the analog value in different formats. These values are associated with analog input IA3 which provides a Word type variable for further processing.

Table 15: Numerical value – analog value output with invisibility activated

Numerical value 1:	Display of the analog value with the scaling range 0 to 10; two decimal places; Unit of measure "V" (Volt); make number invisible via Boolean operand I1; make contact bit logic (Visibility tab)
Numerical value 2:	Display of the analog value with the scaling range –5 to +5; two decimal places; Unit of measure "V" (Volt); make number invisible via Boolean operand I2; make contact bit logic (Visibility tab)

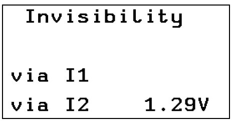


Figure 164: Make invisible function

Screen 6:

Screen 6 shows a simple example of the display change using an external trigger. In this example input I1 is the external trigger. The screen contains two numerical values. These are associated with the output QV of timing relay T08 which is a DWord type variable. The timing relay is run in a loop from 0 to 10 seconds.

Table 16: Numerical value – analog value output with display change via external trigger input

General settings:	Value range from 0 to 11 000; unit of measure: s (seconds); scaling range from 0 to 11, two decimal places; display change active; detection via external trigger input
Left numerical value:	Display of timing relay value T08, I1 actuated: the numerical value flashes as the external trigger is set for flashing display via Boolean operand I1
Right numerical value:	Display of timing relay value T08; I1 actuated: the numerical value is inverted as the external trigger is set for inverted display via Boolean operand I1

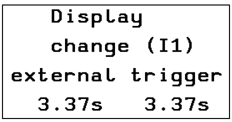


Figure 165: External trigger

Screen 7:

Screen 7 shows a simple example of the display change using an internal limit value comparison. The screen shows two numerical values that show the output values (QV output) of the timing relay T08. The variable type is DWord. The timing relay is run in a loop from 0 to 10 seconds.

A value range from 0 to 11000 is defined in the Number range/format tab. The unit of measure is set to “s” (seconds). The scaling range is defined from 0 to 11 with 2 decimal places. The display change is active and is detected via the internal limit value comparison. The numerical value display flashes for the left numerical value and is inverted for the right numerical value.

Table 17: Numerical value – analog value output with display change via internal limit value comparison

Left numerical value:	The numerical value flashes in the range from 0 to 3 and from 7 to 10. The upper limit value is set to 7 (the display change occurs from numerical value 7 to 10) in the Display change tab and the lower limit value is set to 3 (the display change occurs from numerical value 0 to 3). There is therefore no display change between 3 and 7.
Right numerical value:	The numerical value is inverted in the range from 0 to 3 and from 7 to 10. The upper limit value is set to 7 (the display change occurs from numerical value 7 to 10) in the Display change tab and the lower limit value is set to 3 (the display change occurs from numerical value 0 to 3). There is therefore no display change between 3 and 7.

No display change | Display change

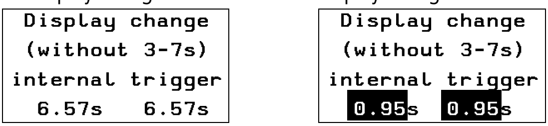
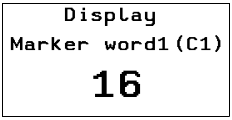


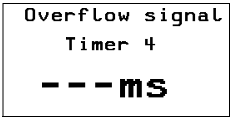
Figure 166: Display change via internal limit value comparison



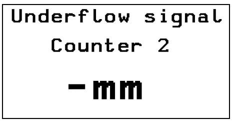
Screen 8:
Screen 8 shows an example of outputting a marker word. The numerical value in the screen is associated with marker word MW01. The variable type is Word.

A value range from 0 to 999 is defined in the Number range/format tab. The timing relay T09 switches the counter C01 (→ circuit diagram in EASY-SOFT-PRO) every two seconds. The output QV of the counter function block writes the data to marker word MW01 (defined in the Parameters tab in the Function block output area). If the upper switch value SH (38) is reached, the counter resets itself.

→ In EASY-SOFT-PRO the contents of the marker can be viewed both in decimal and hexadecimal format. Only decimal format, however, is shown in the display!



Screen 9:
Screen 9 shows an example of how to display a value overflow. The associated variable is the QV output of timing relay T04. The variable type is DWord. The timing relay runs from 0 to 4000 ms. A value range from 0 to 3000 and “ms” as unit of measure are defined in the Number range/format tab. This causes a value overflow as soon as 3000 ms is exceeded. This is indicated in the display by an overflow signal.



Screen 10:
Screen 10 shows an example of how to display a value underflow. The associated variable is the QV output of counter C02. The variable type is DWord. The counter C02 is triggered by a flashing pulse from timing relay T07. The counter counts down from 12 to 0. A value range from 6 to 12 and “mm” as unit of measure are defined in the Number range/format tab. This causes a value underflow as soon as 6 mm is undershot. This is indicated in the display by an underflow signal.

Value entry

This screen element enables you to enter numerical setpoint values on the device during operation and thus make interventions in a process. The process value is thus entered via the set variable linked in the program. The MFD saves the entered value internally. The value is processed by the program and displayed until it is modified again by the operator or the program. Without an operator entry this mask element functions like the numerical value mask element and therefore displays the value of the associated set variable.

When you start to enter a value on the MFD device, the entry field shows the last variable value. You start the entry by pressing **OK**. This activates Selection mode in which you can use the **cursor buttons** to move between the value entry elements of a screen. The order of the selected elements is from the back to the front. This order is defined by their positioning and can be altered in the toolbar. Pressing **OK** once more activates Entry mode.

Value and scaling range
The value range defines the range that is to be displayed. If the values are below or above this range, the MFD activates an underflow or overflow signal.

The scaling range is used for scaling the value range. The lower and upper values of the scaling range are assigned to the lower and upper values of the value range respectively. The MFD displays the value range if a scaling range has not been defined.

Examples:

In order, for example, to display the value range (0 to 255) of a counter as a percentage (0 to 100 %), enter "0" as the minimum value and "255" as the maximum value in the Value range field. Enter "%" as the unit of measure. Activate the Scaling range field and enter "0" as the minimum value and "100" as the maximum value.

In order, for example, to display the value range (0 to 120) of a timing relay in minutes (0 to 2 min), enter "0" as the minimum value and "120" as the maximum value in the Value range field. Enter "min" as the unit of measure. Enter "0" in the activated Scaling range field as the minimum value and "120" as the maximum value.

Programming in EASY-SOFT-PRO:

▶ Hold down the left mouse button and drag the Value entry screen element into the screen.

▶ Position the mouse above the screen element and move it to the required position with the left mouse button depressed.

The height of the screen element frame depends on the font size selected. Three sizes are available:

- Normal font,
- Double font and
- Quadruple font.

You can change the font size by activating the context menu (right mouse button) or by dragging the selection handle at the bottom edge of the element frame concerned. When increasing the font size, ensure that there is sufficient space available underneath the element.

The width of the screen element frame can be scaled as required by dragging the selection handle on the sides.

Set variable tab

On the Set variable tab you define the Boolean variable you wish to set.

Number range/format tab

The Number range/format tab is used for defining the following:

- Value range,
- Unit of measure,
- Scaling range and
- Input format.

The display of the value sign can be forced.

Visibility tab

The Visibility tab enables you to make the screen element invisible via an associated variable.

Operability tab

In the Operability tab, select the associated variable for disabling the entry element.

Example program – value entry

The program consists of seven screens that can be selected using the Cursor buttons ∧ and ∨. The screen change was defined in the Button editor (→ Section "Button editor", Page 333). When the MFD is in RUN mode, you can change values that are processed in the program by using the Cursor buttons ‹ › ∧ ∨. The MFD shows the actual values in the display.

In order to change values, the MFD must first switch to Selection mode. In Selection mode you can select the value entry element containing the values you wish to change.

▶ Press **OK** to switch the MFD to Selection mode. Use **ESC** to exit Selection mode.

The selected value entry element will flash. If several value entry elements are present, use the Cursor buttons ‹ › ∧ ∨ to select the required element (→ Screen 7).

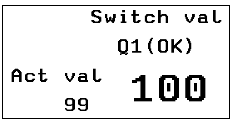
▶ Press the **OK** button to move from Selection mode to Entry mode.

- Move to the required position using the Cursor buttons < and >, change the value with the Cursor buttons ^ and ∨.
- Accept the modified value by pressing **OK**. Press **ESC** if you wish to retain the previous value.
  In both cases you return to Selection mode and can exit by pressing **ESC**.

The circuit diagram contains timing relay T01 which triggers counter C02. When the counter reaches the upper switch threshold SH, it switches output Q1 to 1. The value of the counter's function block output QV is transferred to marker word MW06. The upper setpoint SH is associated with marker word MW07 and the preset actual value SV with marker word MW05. In RUN mode, the marker words are where the actual data is stored and where new data is written via the value entry elements. The counter can be reset at any time via input I2. I1 (rising edge) is used to accept the value in marker word MW05 as a new preset actual value SV. As there is no setpoint value in marker word MW07 (switch value for Q1) when the program is started, this is interpreted as switch value "zero", and output Q1 is switched immediately to 1.

Standard path: C:\Programme\Moeller Software\EASY-SOFT 5 [Pro]\Samples\AWB2528-1480\

Program: Werteingabe_BspProg_01.e40



Screen 1:

Screen 1 illustrates the value entry option using the set variable marker word MW07 (switch value for Q1). The value entry element is enlarged on the display. A value range from 0 to 9999 and Allow digit selection as input format are defined in the Number range/format tab. Input I01 is assigned make contact bit logic in the Operability tab. The value is written to marker word 7 and accepted by counter input SH. Q1 is set to 1 if the counter actual value QV is greater than or equal to the upper setpoint SH. If the actual value already has a higher value than the switch value you can reset the counter via I2. Q1 is immediately reset to 0 and is not active again until the switch value is reached.

Screen 2:

Screen 2 illustrates the activation of the value entry via the set variable marker word MW05 (preset actual value SV). After you have entered a new value, this is written to marker word MW05 by actuating I1. This is associated with the preset actual value SV of counter C02. When transferred, the actual value QV switches to the SV value entered and continues counting from this value.

Set entry: 80

```
         Set
QV    Entry(OK)
304       80
      Transfer I1
```

Transfer of value

```
         Set
QV    Entry(OK)
 80       80
      Transfer I1
```

Figure 167: Transfer of value when I1 closes

Screen 3:

Screen 3 shows an example from the Number range/format tab, particularly the Scaling range area and Input format area → Allow digit selection. You can enter any value from 0.00 to 10.00 (Allow digit selection). After I1 closes, the preset actual value SV is transferred. When transferred, the actual value QV switches to the SV value entered and continues counting from this value. The maximum scaling range (10.00) is assigned to the maximum value range (9999). The value of QV will therefore jump to 9999 when the entered value is 10.00 and I1 is actuated.

Value entry = 10.00

```
Allowdigit selec
       Set entry
  QV      (OK)
   15     10.00
```

QV value = 9999

```
Allowdigit selec
       Set entry
  QV      (OK)
 9999     10.00
```

Figure 168: Transfer of value with Allow digit selection set

Screen 4:

Screen 4 shows an example of the Number range/format tab, particularly the Scaling range area and Input format area → Fixed step width. You can enter any value from 0.00 to 10.00 in step widths of 0.50 (fixed step width). After I1 closes, the preset actual value SV is transferred. When transferred, the actual value QV switches to the SV value entered and continues counting from this value. The maximum scaling range (10.00) is assigned to the maximum value range (9999). The value of QV will therefore jump to 9999 when the entered value is 10.00 and I1 is actuated.

Value entry

```
  Step width
      Set entry
  QV      (OK)
   16      8.50
```

Transfer of QV value

```
Step width
      Set entry
  QV      (OK)
 8500      8.50
```

Figure 169: Transfer of value with fixed step width set

Screen 5:

Screen 5 shows an example of the Visibility tab. If I4 is closed, the value entry element is made invisible. The make contact bit logic is set. When using break contact bit logic, this is the same as a "visible circuit". Even when invisible the element is operable and value entry is still possible.

Set value visible

```
Set value entry
     Invisible
   QV  via I4
  299    1037
```

Set value invisible

```
Set value entry
     Invisible
   QV  via I4
  299
```

Figure 170: Making the value entry element invisible

Screen 6:

Screen 6 shows an example of the Operability tab. (visibility set as in screen 5.) When I3 closes, the entry function is disabled. Make contact bit logic is selected. If the value entry element is disabled, you can still select it (flashing) but Entry mode (changing values) is disabled. If the value entry element is in Entry mode when I3 closes, the MFD automatically switches to Selection mode. When using break contact bit logic, entry is only possible if I3 is actuated. The value entry element is made invisible via I4. However, it is still operable and values can still be modified.

I3 not actuated I3 actuated

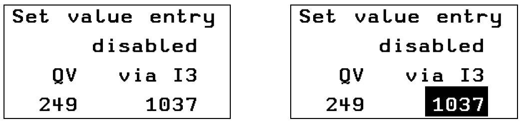


Figure 171: Value entry element disabled

Message text

This mask element can be used to display texts that are stored beforehand in a text table inside the program. A text can have a maximum length of 16 characters. Additional blanks are added to the displayed text if it is shorter than the element. A message text is used for indicating status changes in the process. In order to visually indicate changes of this kind, you can link message texts with a variable (associated variable). When the variable concerned assumes a specified value, the MFD outputs the appropriate message text. The default text is output if the variable assumes a value that is not assigned to a text.

Programming in EASY-SOFT-PRO:

▶ Hold down the left mouse button and drag the Message text screen element into the screen.

▶ Position the mouse above the screen element and move it to the required position with the left mouse button depressed.

The height of the screen element frame depends on the font size selected. Two sizes are available:

- Normal font size and
- Double font size.

You can change the font size by activating the context menu (right mouse button) or by dragging the selection handle at the bottom edge of the element frame concerned. When increasing the font size, ensure that there is sufficient space available underneath the element.

The width of the screen element frame can be scaled as required by dragging the selection handle on the sides.

→ Ensure that the screen elements are sufficiently large for containing the corresponding texts of all languages selected. This can be checked easily in the screen overview.

Associated variable tab

On the Associated variable tab you define the variable with the value for activating the output text.

Message texts tab

Assign in the Message texts tab the value of the associated variable for its corresponding message text, and select the language and the default text.

Visibility tab

The Visibility tab enables you to make the screen element invisible via an associated variable.

Display change tab

The Display change tab offers the following display forms of the message, which can be controlled with an associated variable:

- flashing,
- Inverted.

Example program 1 – activating message texts with a Boolean variable

The program consists of seven screens that can be selected using the Cursor buttons ⌃ and ⌄. The screen change was defined in the Button editor (→ Section "Button editor", Page 333). The circuit diagram uses six on-delayed timing relays that activate outputs Q1 to Q4 and LED03 in succession after an elapsed time. This operation is run in a continuous loop since timing relay T06 resets all timing relays after a set time. The message texts are controlled in all screens by means of Boolean variables. As these only have two states (0/1), only two different messages can be output via each message text element. If other variable types are used, the number of possible message texts can be increased accordingly.

Standard path: C:\Programme\Moeller Software\EASY-SOFT 5 [Pro]\Samples\AWB2528-1480\

Program: Meldungstext_BspProg_01.e40

Screen 1:

Screen 1 illustrates the activation of message texts by means of the associated variable. The output of two different texts in one message text element is also illustrated. The first screen contains four message text elements. These are activated via the outputs Q1 to Q4 and appear in the display. The Message text tab is used to assign the message to be output with a particular state of the associated variable. In this example, if Q1 = 0, the message “no data” is output, and “Information” is output if Q1 = 1. All other message text elements have only one message text which is displayed when the corresponding output Q.. = 1.

Q1 = 0 Q1, Q2, Q3 = 1

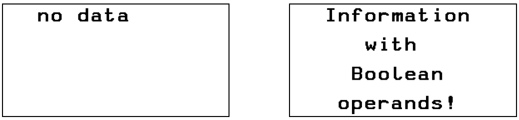
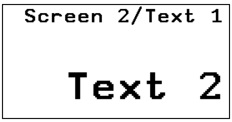


Figure 172: Text display using Boolean operands



Screen 2:

Screen 2 illustrates the activation of message texts by means of the associated variable. A message text is also enlarged in the display. The mask contains two message text elements. These are activated via the outputs Q1 and Q3 and appear in the display. The texts are only displayed if the corresponding outputs are 1. No message is assigned to output status 0.

Screen 3:

Screen 3 illustrates the control of an enlarged message text element containing two message texts that are assigned to the two states of output Q2.

Q2 = 0 → Message text: Q2 OFF
Q2 = 1 → Message text: Q2 ON

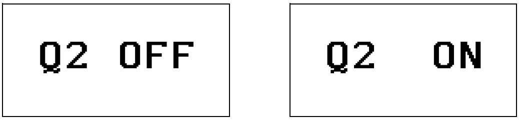


Figure 173: Message texts using an associated variable

Screen 4:

Screen 4 is basically the same as screen 3 except that it also includes the Visibility element. The message text can be made invisible via input I1. The setting for this was made in the Visibility tab. Make contact bit logic is selected. With break contact bit logic, the message text element is made visible when actuating I1.

I1 not actuated | I1 actuated

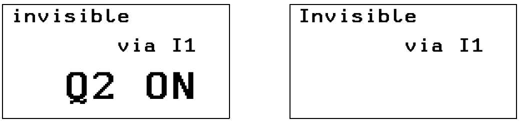


Figure 174: Hiding a message text

Screen 5:

Screen 5 is basically the same as screen 4 except that the Invisible function has been replaced with Flash as the Display change function. When input I2 = 1 (make contact bit logic), the message text can be made to flash.

I2 not actuated — I2 actuated

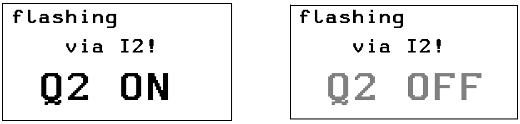


Figure 175: Message text flashing

Screen 6:

Screen 6 is basically the same as screen 5 except that Inverted was selected for the Display change function. When input I2 = 1 (make contact bit logic), the message text in this screen can thus be inverted.

I2 not actuated — I2 actuated

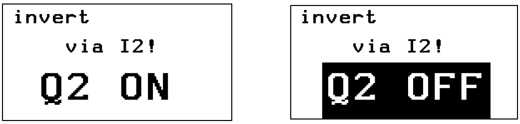


Figure 176: Message text inverted

Screen 7:

Screen 7 is basically the same as screen 6 except that the inverted display is controlled by timing relay T07 (asynchronous clock pulse). The text flashes in the display at different pulse and pause times.

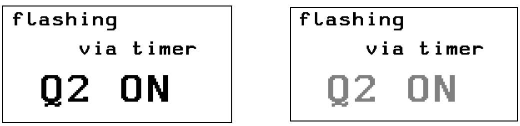


Figure 177: Message text inverted flashing

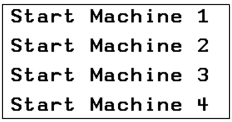
Example program 2 – activating message texts with a timing relay

The program consists of two screens containing message texts. The screens are activated automatically in succession in the display. The circuit diagram uses six on-delayed timing relays T01 to T06 that activate outputs Q1 to Q4 and LE03 in succession after an elapsed time. This operation is run in a continuous loop since timing relay T06 resets all timing relays after a set time.

The screen activation is executed via counter C01. This has the value 1 as the upper setpoint SH. The counter values 0 and 1 are therefore used as activation values for the two screens. The resetting of timing relays T01 to T06 by T06 activates counter C01 which then reaches the value 1 and therefore has already reached its upper setpoint. This then causes screen 2 to be displayed. The on-delayed timing relay T08 resets counter C01 back to 0 after the set time (0.8 s) has elapsed. Screen 1 is then displayed again and timing relay T08 is no longer activated. This loop is repeated continuously.

Standard path: C:\Programme\Moeller Software\EASY-SOFT 5 [Pro]\Samples\AWB2528-1480\

Program: Meldungstext_BspProg_02.e40



Screen 1:

The first screen contains four message text elements. These are activated in succession via the outputs Q1 to Q4 and appear in the display.

Screen 2:

The second screen contains one message text element with one message text.

Example program 3 – message text with a display change

The program consists of two screens containing message texts. The screens are activated automatically in succession in the display. This program is an extension of the program Meldungstext_BspProg_02.e40. The extension consists of two messages being contained in one message text element in screen 2, which are also toggled automatically. A display change is also executed in one message text. The circuit diagram uses six on-delayed timing relays T01 to T06, that activate outputs Q1 to Q4 and LE03 in succession after an elapsed time. This operation is run in a continuous loop since timing relay T06 resets all timing relays after a set time. The make contacts (C01ZE) of the counter in the circuit diagram are switched directly before the outputs Q1 to Q4 as well as before LE 3. The outputs are only meant to be active with screen 1, i.e. when the counter has the value 0.

The screens are activated via counter C01. This has the value 2 as the upper setpoint SH, thus providing the counter values 0, 1 and 2. Counter value 0 activates mask 1, counter value 1 activates the first message text in screen 2 and counter value 2 the second message text in screen 2. When the timing relays T01 to T06 are reset, the counter receives a pulse and is incremented by one value. If the counter C01 reaches its upper setpoint with the second pass, this starts the on-delayed timing relay T08. This then resets the counter C01 to 0 after the set time has elapsed (0.8 s). This loop is repeated continuously.

Standard path: C:\Programme\Moeller Software\EASY-SOFT 5 [Pro]\Samples\AWB2528-1480\

Program: Meldungstext_BspProg_03.e40

```
Start Machine 1
Start Machine 2
Start Machine 3
Start Machine 4
```

Screen 1:

The first screen contains four message text elements. These are activated in succession via the outputs Q1 to Q4 and appear in the display.

Screen 2:

The second screen contains a message text element with two message texts. The message texts are activated via counter values 1 and 2. The first resetting of timing relays T01 to T06 by T06 also activates counter C01 which then reaches the value 1. The Error message text is then displayed. This flashes since Flashing was set on the Display change tab and with function block parameter FB (counter value <= 1). RESTART is displayed if the counter value of C01 is 2.

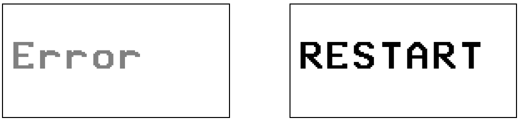


Figure 178: Message text as status display

Example program 4 – activating message texts with a default text

The program consists of two screens containing message texts. The screens are activated automatically in succession in the display. This program is an extension of the program Meldungstext_BspProg_03.e40. The extension consists of the display of a default text in screen 2 when the counter takes on values that are not assigned to any messages. The circuit diagram uses six on-delayed timing relays T01 to T06 that activate outputs Q1 to Q4 and LE03 in succession after an elapsed time. This operation is run in a continuous loop since timing relay T06 resets all timing relays after a set time. Counter make contacts (C01ZE) in the circuit diagram are switched directly before outputs Q1 to Q4 and LE03 so that the outputs are only active if screen 1 is active when counter C01 has the value 0.

The screen activation is executed via counter C01. This has the value 4 as the upper setpoint SH. This therefore generates the counter values 0, 1, 2, 3 and 4. The counter is activated with every reset of timing relays T01 to T06. The counter value 0 activates screen 1 and counter value 1 screen 2. Screen 2 remains activated when the counter value is 2, 3 or 4, and counter value 2 and 3 cause the default text to be displayed as there are no message text assigned to these values. The counter value 4 causes the message text RESTART to be displayed in screen 2. If the counter C01 reaches its upper setpoint with this value, this starts the on-delayed timing relay T08. This then resets the counter C01 to 0 after the set time of 0.8 s has elapsed. This loop is repeated continuously.

Standard path: C:\Programme\Moeller Software\EASY-SOFT 5 [Pro]\Samples\AWB2528-1480\

Program: Meldungstext_BspProg_04.e40

```
Start Machine 1
Start Machine 2
Start Machine 3
Start Machine 4
```

Screen 1:

The first screen contains four message text elements. These are activated in succession via the outputs Q1 to Q4 and appear in the display.

Screen 2:

The second screen contains a message text element with two message texts. The message texts are activated via counter values 1 and 4. A default text "default" has also been defined.

- Status value 1. "Error" message text
- Status value 2 and 3: "default" as default text
- Status value 4. "RESTART" message text

If the status value of C01 is 1, the message "error" flashes as the function block parameter FB has been set in the Display change area in the Display change tab. The function block parameter switches if the SL value (lower switch threshold) of the counter is reached (→ EASY-SOFT-PRO function block editor, counter C01).

```
Error
```

```
default:
```

```
RESTART
```

Figure 179: Message text with default text

Example program 5 – several message texts in one text element

The program consists of three screens that you can select via the Cursor buttons ⌃ and ⌄. The screen change was defined in the Button editor (→ Section "Button editor", Page 333). The purpose of this program is to display several messages in one message text element. The Visibility and Display change tabs are also covered. The clock pulse T01 activates counter C01. This counts up to 7 and then resets itself automatically. This therefore produces a continuous loop that activates the individual messages.

Standard path: C:\Programme\Moeller Software\EASY-SOFT 5 [Pro]\Samples\AWB2528-1480\

Program: Meldungstext_BspProg_05.e40

Screen 1:

The first screen contains a message text element with six message texts.

- Status value 1: Message text "These"
- Status value 2: Message text "are six"
- Status value 3: Message text "message texts"
- Status value 4: Message text "in one"
- Status value 5: Message text "message text"
- Status value 6: Message text "element!!!".

It can be seen that the messages can be moved within the entry field.

Screen 2:

Screen 2 is almost the same as screen 1 apart from one addition which enables the message text element to be made invisible via I1 (make contact bit logic). The use of the break contact bit logic would make the element visible.

I1 not actuated I1 actuated

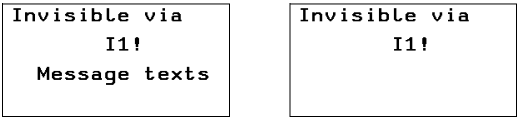


Figure 180: Make message text invisible

Screen 3:

Screen 3 is almost the same as screen 1 apart from one addition, by which the actuation of I1 causes the first message text element to flash and the second message text element to be inverted.

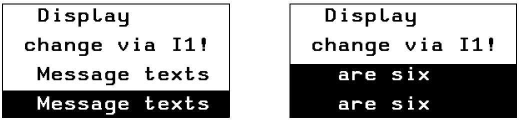


Figure 181: Making message text flash or inverted

Button editor

All buttons of the MFD can be assigned in the button editor of the EASY-SOFT-PRO with appropriate functions which overwrite the basic button functions. Carry out the following steps to associate the buttons of the MFD:

- Select an operable screen element,
- Display backlight,
- Screen change,
- Password logout,
- Set variable to fixed value,
- Increment variable,
- Decrement variable,
- Changeover relay.

Select an operable screen element

If a screen with a value entry is present, you can jump to these value entry elements directly. The value entry element is in Entry mode so that you can change the value directly.

Display backlight

The brightness of the display backlight can be adjusted in stages.

Screen change

This function enables the operator to change to other saved screens during operation. If the operator is to be able to move between several screens using a button function you must assign a screen change button element to an operator button in each of these masks.

Password logout

The Screen overview contains the Passwords tab for defining a logout time. This logout time is skipped with the Password logout function.

Set variable to fixed value

This function assigns a fixed value to the selected variable, such as for resetting to a defined value.

Increment variable

The variable value is increased by the set step width.

Decrement variable

The variable value is decreased by the set step width.

Changeover relay

The state of a variable or a function block input is negated.

6 easy-NET Network, COM-LINK Serial Connection

Introduction to easy-NET

All MFD-Titan units have an easy-NET network interface connection. This network is designed for eight stations.

Using the easy-NET you can:

- Process additional inputs and outputs.
- Implement faster and improved control using decentralised programs.
- Synchronise date and time
- Read and write inputs and outputs.
- Send values to other stations.
- Receive values from other stations.
- Load programs from or to any station.

The easy-NET network is based on the CAN network (Controller Area Network). CAN is specified by the ISO 11898 standard. CAN has the following in-built features:

- Message oriented transmission protocol.
- Multimaster bus access capabilities with non-destructive bitwise bus arbitration via priority messaging (Arbitration: An instance which defines which hardware can use the bus next).
- Multicast broadcast messaging with receiver side message filtering.
- High level of real-time capability (short reaction time for high priority messages, short fault message get times).
- Functionality in environments with severe interference (short block lengths).
- High level of error security.

→ CAN has been used as the basis for the design of the easy-NET network. The messages have been adapted and optimised to suit the requirements of the MFD-Titan environment.

easy-NET network topologies, addressing and functions

The easy-NET allows the configuration of a line topology. There are two wiring methods which can be used for the required addressing options:

- "Loop through the unit" wiring arrangement,
- Wiring arrangement using a T connector and a spur line.

Loop through the unit wiring method

With this wiring method it is possible to implement the addressing of the stations via station 1 or the EASY-SOFT (-PRO). If the line is interrupted, the network is no longer operational from this point in the network.

T connector and spur line

Each device must be addressed individually with this wiring method by:

- Downloading the program,
- Downloading the address with EASY-SOFT (-PRO),
- Using the display or
- The device is already assigned an address.

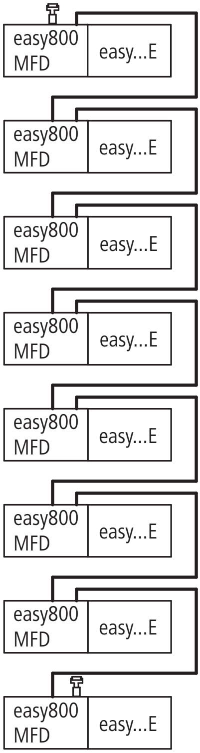
If a spur line is removed on a station, all other devices in the network remain functional.

→ The spur line between the T connector and the device must not exceed 0.3 m. Otherwise communication via easy-NET may be impaired.

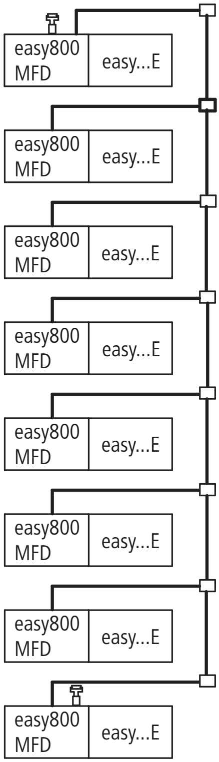
Topology and addressing examples

Physical location, place	Station number Example 1	Station number Example 2
1	1	1
2	2	3
3	3	4
4	4	8
5	5	7
6	6	2
7	7	6
8	8	5

Loop through the unit



T connector and spur line



- Example 1: physical location is the same as the station number
- Example 2: physical location is not the same as the station number (apart from location 1 being the same as station 1).

→ Physical location 1 is always assigned as station 1. Station 1 is the only station which must be present.

Position and addressing of the operands via easy-NET

Stations	Basic unit		Local expansion		Network bit data		Network word data	
	Input I	Output Q	Input R	Output S	Input RN	Output SN	Receive	Send
1	1 I 1 to 16	1 Q 1 to 8	1 R 1 to 16	1 S 1 to 8	2 to 8 RN 1 to 32	2 to 8 SN 1 to 32	GT 1 to 32	PT 1 to 32
2	2 I 1 to 16	2 Q 1 to 8	2 R 1 to 16	2 S 1 to 8	1, 3 to 8 RN 1 to 32	1, 3 to 8 SN 1 to 32	GT 1 to 32	PT 1 to 32
3	3 I 1 to 16	3 Q 1 to 8	3 R 1 to 16	3 S 1 to 8	1, 2, 4 to 8 RN 1 to 32	1, 2, 4 to 8 SN 1 to 32	GT 1 to 32	PT 1 to 32
4	4 I 1 to 16	4 Q 1 to 8	4 R 1 to 16	4 S 1 to 8	1 to 3, 5 to 8 RN 1 to 32	1 to 3, 5 to 8 SN 1 to 32	GT 1 to 32	PT 1 to 32
5	5 I 1 to 16	5 Q 1 to 8	5 R 1 to 16	5 S 1 to 8	1 to 4, 6 to 8 RN 1 to 32	1 to 4, 6 to 8 SN 1 to 32	GT 1 to 32	PT 1 to 32
6	6 I 1 to 16	6 Q 1 to 8	6 R 1 to 16	6 S 1 to 8	1 to 5, 7, 8 RN 1 to 32	1 to 5, 7, 8 SN 1 to 32	GT 1 to 32	PT 1 to 32
7	7 I 1 to 16	7 Q 1 to 8	7 R 1 to 16	7 S 1 to 8	1 to 6, 8 RN 1 to 32	1 to 6, 8 SN 1 to 32	GT 1 to 32	PT 1 to 32
8	8 I 1 to 16	8 Q 1 to 8	8 R 1 to 16	8 S 1 to 8	1 to 7 RN 1 to 32	1 to 7 SN 1 to 32	GT 1 to 32	PT 1 to 32

→ The RN-SN connection is a point-to-point connection between the stations indicated. With RN and SN the number of the contact must have the same number as the coil. Example: 2SN30 from station 8 is sent to 8RN30 of station 2.

→ Every station with a circuit diagram has read access to the physical station inputs and outputs of other stations and can process them locally.

Example 1

Station 1 is to read the input I1 of station 2 and write to output Q1 of station 2. Station 2 does not have a circuit diagram.

```
2I 01------------------------------[2Q 01
```

Figure 182: Circuit diagram in station 1

Example 2:

Marker M 01 of station 4 is to switch the output Q1 of station 3 via the network. Both stations have a circuit diagram.

```
 M 01------------------------------[3SN 01
```

Figure 183: Circuit diagram in station 4: Set coil 01 in station 3

```
4RN 01------------------------------[ Q 01
```

Figure 184: Circuit diagram in station 3: Get value from coil 01 in station 4

Functions of the stations in the network

The stations on the easy-NET can have two different functions:

- Intelligent stations with their own programs (stations 1 to 8),
- Input/output devices (REMOTE IO) without their own program (stations 2 to 8).

→ Station 1 must always have a circuit diagram.

Possible write and read authorisation in the network

The stations have differing read and write authorisation in the easy-NET network according to their functions and configuration.

Station 1
Authorised read access to all inputs and outputs of all stations regardless of the function. Observe the setting of SEND IO (→ Section “Send each change on the inputs/outputs (SEND IO)”, Page 343).

Authorised write access to the station's own local outputs.

Authorised write access to the physical digital outputs of the stations which are functioning as input/output devices.

Write authorisation to the network bit data 2 to 8 SN 1 to 32.

Stations 2 to 8

Input/output device function
No read and write authorisation.

Intelligent station function
Authorised read access to all inputs and outputs of all stations regardless of the function. Observe the setting of SEND IO (→ Section “Send each change on the inputs/outputs (SEND IO)”, Page 343).

Write authorisation to its own local outputs.

Write authorisation to the network bit data SN 1 to 32.

Configuration of the easy-NET network

easy-NET can be configured so that it can be optimised for your application.

Station number

The station number is identified as the easy-NET-ID: in the device. The station number can be set on devices with a display using the buttons on the MFD-Titan.

→ All the easy-NET settings are best carried out on station 1. The entire network can be configured via station 1. The configuration should only be carried out locally when a device is replaced.

Valid station numbers for operation are 01 to 08.

Station number 00 = factory default setting

With station number 00, double address assignment cannot occur when an existing device is being exchanged.

Transmission speed

The MFD-Titan device hardware allows you set transmission speeds between 10 and 1000 kBaud in specific stages. The length of all cables is determines the maximum permissible data transfer rate (→ chapter “Technical data”, Page 428).

The data transfer rate is set under the BAUDRATE: menu item.

Possible baud rates are: 10, 20, 50, 125, 250, 500 and 1000 kB

125 kB = factory default setting

Pause time, changing the write repetition rate manually

Every easy-NET network connection automatically determines the number of stations which are active on the network, the baud rate which is used and the total number of bytes which are transmitted. The minimum pause time which a device requires is automatically determined using this data in order to ensure that all devices can send their messages. If a pause time is to be increased, the value of the BUSDELAY: must be set greater than zero.

Value "1" doubles the pause time, value "15" will increase it by a factor of 16.

$t_{pnew} = t_p \times (1 + n)$
t_{pnew} = new pause time
t_p = pause time determined by the network
n = value on BUSDELAY

→ An increase in the pause time means that fewer messages (inputs, outputs, bit data, word data) are transferred per time unit.

The reaction speed of the entire controller depends on the baud rate, the pause time and the quantity of transferred data.

The smaller the amount of data transferred, the faster the reaction times of the system.

→ An increase in the pause time is only useful during commissioning. To ensure that the data for the power flow display is updated faster in the PC, a longer range for this data is created on the network within this pause time.

Send each change on the inputs/outputs (SEND IO)

The SEND IO function should be used if you wish to send any change in input or output status immediately to all other network stations. SEND IO should be activated if intelligent stations have read access to the inputs and outputs of other stations (2I 02, 8Q 01, etc.).

```
SEND IO      √
```

This means that the quantity of messages on the network can increase significantly.

→ If high-speed counters are used, the SEND IO function should be deactivated. Otherwise the input data is written very rapidly onto the network as they change continuously, leading to unnecessary loading of the network.

If intelligent devices are required to exchange bit information, it should be implemented via RN and SN.

SEND IO √ = factory default setting

Automatic change of the RUN and STOP mode

REMOTE RUN should be activated if stations 2 to 8 are to automatically follow the mode change of station 1 during operation.

→ Input and output devices must always have SEND IO activated, to ensure that station 1 always receives up-to-date input and output data.

→ Intelligent stations with display only follow the operating mode change when the display is showing the Status display or a text.

The following is of utmost importance during commissioning!

Attention!
If several engineers are commissioning a machine or system involving several spatially separated elements via the easy-NET network, it must be ensured that REMOTE RUN is not activated.

Otherwise unwanted machine or system starts may occur during commissioning. The associated events depend on the machines or systems involved.

REMOTE RUN √ = factory default setting

Input/output device (REMOTE IO) configuration

All devices are factory set for operation as input and output devices. This has the advantage that devices can be used immediately as I/O devices, regardless of whether they have a display or not. You only need to assign the station number. This can be implemented via EASY-SOFT (-PRO) or on a Station 1 with a display.

If you want to assign a device as an intelligent station on the network, the REMOTE IO should be deactivated.

```
REMOTE IO
```

Figure 185: Remote IO deactivated

The standard settings for the input and output devices are:

```
SEND IO       √
REMOTE RUN    √
REMOTE IO     √
```

Station number (easy-NET-ID) and baud rate can be specified via station 1.

Station message types

The easy-NET network recognises various message types. They are:

- Output data of station 1 (Q., S.) which is sent to stations without programs.
- Network outputs and inputs sent and received between stations with programs (*SN, *RN).
- Data sent and received via the network between stations with programs (PT and GT function blocks).
- Inputs, outputs, station status (I, R, Q, S) transfers.
- Loading programs to and from every station.

The easy-NET network is based on a CAN (Controller Area Network) system. Each message type has its own ID. The message priority is determined via the respective ID. This is important in transmission borderline cases to ensure that all messages reach their destination.

Transfer behaviour

Network CPU data transfer to program image

The MFD-Titan network connection is equipped with its own CPU. Network data can therefore be processed whilst the program is running. After each program cycle, the status of the network data is written to the operand image of the program and the send data is read from the image. The program runs through the next cycle with this data.

Reading and sending the network data from the CPU
The network CPU of the station reads every message on the network. If the message is relevant to the station, it is accepted into a message memory.

If the content of a send message changes, it is sent. Transmission only occurs when there is no message on the network.

easy-NET is configured so that every station can send its messages. This means that the station must observe a pause time between sending messages. The pause time increases the higher the number of stations and the lower baud rate setting.

The number of stations is recognised by every station via a "sign of life" signal.

→ The following applies to fast message transfer:

- Set the fastest possible baud rate to suit the network length and cable cross-section.
- Fewer messages means faster messages.
- Avoid program downloads during the RUN mode.

Signs of life of the individual stations and diagnostics

The inputs and outputs message type is used as a "sign of life" recognition to ensure that the state of a station can be recognised by other stations. The states of the inputs and outputs are sent cyclically and at the set baud rate, irrespective of the SEND IO setting. If the inputs and outputs of a station are not recognised by other stations after a time determined by the baud rate, the station is deemed to be disconnected until the next "sign of life" is recognised.

Evaluation occurs at the following intervals:

Baud rate [KB]	Stations must send a “sign of life” every ... [ms]	Stations recognise the absence of a “sign of life” signal after [ms]
1000	60	180
500	60	180
250	120	360
125	240	720
50	600	1800
20	1500	4500
10	3000	9000

If the absence of a “sign of life” is detected, the respective diagnostics contact is set to 1.

Diagnostics contact	Station number
ID 01	1
ID 02	2
ID 03	3
ID 04	4
ID 05	5
ID 06	6
ID 07	7
ID 08	8

→ If a station does not send a "sign of life" signal (station not available, easy-NET interrupted), the respective diagnostics contact ID .. is activated.

Attention!
If the states of the inputs, outputs or data are required by a station without fail, the respective diagnostics contact should be evaluated and the information applied in accordance with its respective application.

If the respective diagnostics contacts are not evaluated, it may cause faults in your application.

→ The data to be read from a faulty station is set to 0 after the fault is detected.

Network transmission security

easy-NET is a CAN-based network. CAN is used in cars and commercial vehicles in all areas. The same fault recognition capability with data transfer applies as with CAN. A BOSCH study relating to undiscovered and corrupt messages determined the following:

The probability of non-discovery of a corrupted message (residual error probability) is: $< 10^{-10}$ message error rate.

The message error rate depends on:

- Bus loading,
- Telegram length,
- Malfunction frequency,
- Number of stations.

Example:

Network with:

- 500 kBaud,
- average bus load 25 %,
- average operating time 2000 h/year,
- average error rate of 10^{-3},
  i.e.: 1 message is faulty every 1000,
- transmission of 1.12×10^{10} messages per year of which 1.12×10^{7} messages are faulty,
- residual error probability: $r < 10^{-10} \times 10^{-3} = 10^{-13}$.

This means: one of 10^{13} messages is so corrupt that the fault cannot be recognised as such. For a network, this corresponds to a working time of approx. 1000 years.

Introduction to COM-LINK

The COM-LINK is a point-to-point connection using the serial interface. This interface connection allows the reading of input/output states as well as the reading and writing of marker ranges. This data can be used for setpoint entry or for display functions. The stations of the COM-LINK have different functions. The active station controls the entire interface connection. The remote station responds to the requests of the active station. The remote station cannot distinguish whether the COM-LINK is active or whether a PC with EASY-SOFT-PRO is using the interface.

→ The two devices must support the COM-LINK, e.g.: MFD and easy800 devices from device version 04 are COM-LINK enabled.

→ Only the MFD can be the active station in a COM-LINK connection.

Remote stations can be MFD or easy800.

Topology

The following topologies are possible:

Two devices, MFD as active station and easy800 or MFD as remote station

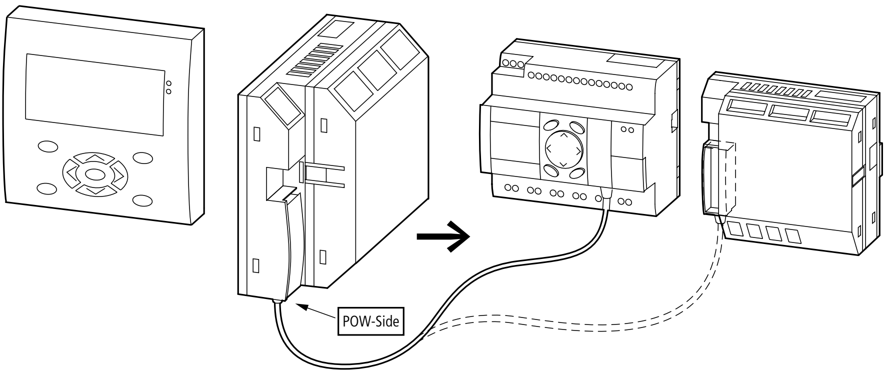


Figure 186: COM-LINK connection to an easy800 or another MFD

Establishing a COM-LINK connection to an easy-NET station.

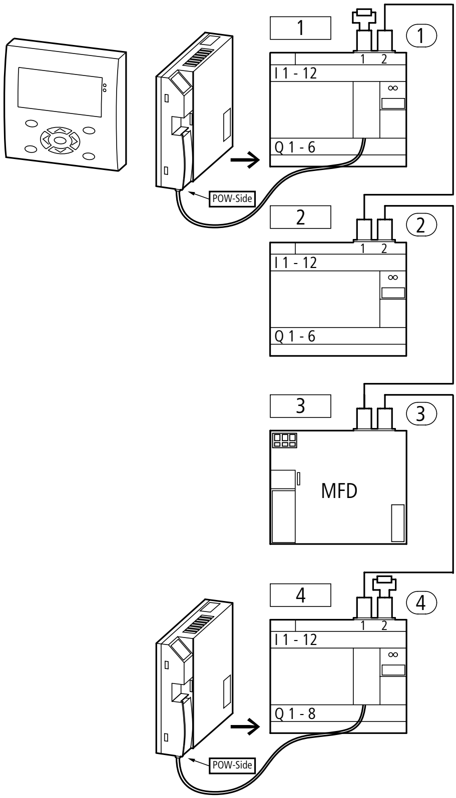


Figure 187: easy-NET operation and COM-LINK connections.

A COM-LINK connection can be established with an easy-NET station. The same conditions apply here as with operation without easy-NET.

Data accesses via COM-LINK

The following data access operations are possible from the active station to the remote station:

Active station, read		Remote station
Inputs	1I01 to 1I16	I01 to I16
Inputs of local expansion unit	1R1 to 1R16	R01 to R16
Outputs	1Q01 to 1Q08	Q01 to Q08
Outputs of local expansion unit	1S01 to 1S08	S01 to S08
Diagnostics bits of easy-NET	1ID01 to 1ID08	ID01 to ID08
Analog inputs	1IA01 to 1IA04	IA01 to IA04
Analog output	1QA01	QA01

Write/read accesses in the marker range

Active station				Remote station			
1 MD01				MD01			
1 MW01		1MW02		MW01		MW02	
1 MB01	1MB02	1MB03	1MB04	MB01	MB02	MB03	MB04
1 M01 to 1 M32				M01 to M32			
1 MD02				MD02			
1 MW03		1MW04		MW03		MW04	
1 MB05	1 MB06	1 MB07	1 MB08	MB05	MB06	MB07	MB08
1 M33 to 1 M64				M33 to M64			
1 MD03				MD03			
1 MW05		1 MW06		MW05		MW06	
1 MB09	1 MB10	1 MB11	1 MB12	MB09	MB10	MB11	MB12
1 M65 to 1 M96				M65 to M96			
1 MD04				MD04			
1 MW07		1 MW08		MW07		MW08	
.....				....			
....							

1 MD20				MD20			
1 MW39		1 MW40		MW39		MW40	
1 MB77	1 MB78	1 MB79	1 MB80	MB77	MB78	MB79	MB80

The normal rules for addressing the markers apply.

→ Ensure a clear separation of the write range of the two stations. The active station should write different markers to the remote station. Otherwise the markers in the last write operation will be overwritten.

Configuration of the COM-LINK

The active station must have the following settings in order for the COM-LINK connection to be functional:

- Baud rate,
- COM-LINK (active),
- Remote marker range (data exchange range).

Baud rate COM-LINK

The baud rate can be 9600 Baud or 19200 Baud.

```
BAUDRATE:19200B
COM-LINK
REMOTE MARKER...
```

→ In normal applications, select the higher baud rate of 19200 Baud. The baud rate of 9600 baud should only be selected if the connection is frequently faulty.

Factory setting: 9600 Baud

Activating COM-LINK

COM-LINK must be activated in order for it to function.

Factory setting: not active

```
BAUDRATE:19200B
COM-LINK       √
REMOTE MARKER...
```

The tick on the COM-LINK menu item indicates that COM-LINK is active.

Remote markers, COM-LINK data exchange range

→ The REMOTE MARKER.. menu only opens if a tick is displayed next to COM-LINK.

Select the REMOTE MARKER... menu option. Here you can determine, select and modify the data exchange range.

```
READ:
 1MD11 → 1MD015
WRITE:
 1MD16 → 1MD18
```

In the example the READ range was selected with the marker double words MD11 to MD15.

The WRITE range consists of the marker double words MD16 to MD18.

The entire data exchange range available is the marker range MD01 to MD20 of the remote station. The active station accesses these markers with 1MD*. The * indicates the number of the marker concerned.

The smallest possible unit for defining a marker range is an MD marker double word.

Example:

The read range of the active station is 1MD02.

The write range of the active station is 1MD03.

The read range of the remote station is therefore MD03.

The write range of the remote station is MD02.

Operating principle of the COM-LINK connection

The active station at the COM-LINK must be in RUN mode.

Data can only be exchanged with the active station in RUN mode.

The remote station must be in RUN or STOP mode.

The active station scans the remote station. The entire READ marker range is transferred as a string. The entire WRITE marker range is transferred as a string.

→ Data consistency

The data is located in the image range (data range storing the marker states) of the active station (1MD..) as well as in the image range of the remote station (MD..)

Each station writes data to its own image range asynchronously for data communication. As the serial interface transfers large data volumes slower than the devices overwrite the image ranges, the following applies: one marker double word 1MD.., MD.. is consistent.

Within a program cycle, a marker double word that is overwritten via COM-LINK is not constant. The data via the COM-LINK is written to the status image over the course of the program cycle. This means that a different marker value can be present at the start of the program cycle to after the write operation via COM-LINK.

Sign of life detection COM-LINK, diagnostics contact ID09

In order to determine the proper functioning of the COM-LINK connection, the diagnostics contact ID09 is provided on the active station of the COM-LINK connection.

Status of diagnostics contact ID09	Status of the connection
“0“	COM-LINK connection operating correctly or no COM-LINK connection selected.
“1“	COM-LINK connection not functioning, faulty

The time required to detect that the COM-LINK is not working properly depends on the baud rate selected and the event concerned.

Baud rate	Time required for detection of faulty COM-LINK connection.	
	CRC error (data content incorrect)	No response, no hardware connection, remote station not in operation
9600 Baud	250 ms	1.5 s
19200 Baud	120 ms	0.8 s

Attention!
If the states of the inputs, outputs or data are required by a station without fail, the respective diagnostics contact should be evaluated and the information applied in accordance with its respective application.

If the respective diagnostics contacts are not evaluated, it may cause faults in your application.

7 MFD Settings

Settings can only be carried out on MFD models provided with buttons and LCD display.

EASY-SOFT -PRO can be used to set all models via the software.

Password protection

The MFD can be protected by a password against unauthorised access.

In this case the password consists of a value between 000001 and 999999. The number combination 000000 is used to delete a password.

Password protection inhibits access to selected areas. The System menu is always protected by an activated password.

The password can protect the following inputs and areas:

- Start or modification of the program.
- Transfer of a circuit diagram to or from a memory card (display/operating unit variants).
- Change of the RUN or STOP mode.
- Calling and modification of function block parameters
- All settings of the real-time clock.
- Modifications of all system parameters.
- Communication with the individual device (looping to other devices possible).
- Switching off the password delete function.

→ A password that has been entered in MFD is transferred to the memory card together with the circuit diagram, irrespective of whether it was activated or not.

If this MFD circuit diagram is loaded back from the memory card, the password will also be transferred to MFD and is activated immediately.

Password setup

A password can be set via the System menu regardless of the RUN or STOP modes. You cannot change to the System menu if a password is already activated.

▶ Press **DEL** and **ALT** to call up the System menu.

▶ Select the menu option SECURITY… to enter the password.

▶ Press the **OK** button and change over to the PASSWORD… menu.

▶ If you press the **OK** button again, you will access the password input..

```
ENTER PASSWORD
   ------
```

If no password has been entered, MFD changes directly to the password display and displays six dashes: No password available.

▶ Press **OK**, six zeros will appear

▶ Set the password using the cursor buttons:

– 〈 〉 select position in password,

– ︿﹀ set a value between 0 to 9.

```
ENTER PASSWORD
     000042
```

▶ Save the new password by pressing **OK**.

Use **OK** to exit the password display and proceed with **ESC** and ⌄ to the RANGE… menu.

The scope of the password has not yet been considered. The password is now valid but not yet activated.

Selecting the scope of the password

```
CIRCUIT DIAGRAM√↕
PARAMETERS
TIME
OPERATING MODE  ↕
INTERFACE
DELETE FUNCT.
```

▶ Press the **OK** button.

▶ Select the function to be protected or the menu.

▶ Press the **OK** button in order to protect the function or menu (tick = protected).

→ Standard protection encompasses the programs and circuit diagram.

At least one function or menu must be protected.

- CIRCUIT DIAGRAM: The password is effective on the program with circuit diagram and non-enabled function blocks.
- PARAMETERS: The PARAMETERS menu is protected.
- TIME: Date and time are protected with the password.
- OPERATING MODE: The toggling of the RUN or STOP operating mode is protected.
- INTERFACE: The interface is inhibited for access to a connected device. Programs or commands to other devices connected via the NET network are routed further.
- DELETE FUNCT.: After four failed attempts to enter a password, the “DELETE FUNCTION?” prompt appears. This prompt is not displayed if selected. However, it is no longer possible to make changes in protected areas if you forget the password.

Activating the password

You can activate an existing password in three different ways:

- automatically when MFD is switched on again,
- automatically after a protected circuit diagram is loaded
- automatically if a telegram has not been sent on the PC interface 30 minutes after unlocking the device (using EASY-SOFT (-PRO),
- via the password menu.

▶ Press **DEL** and **ALT** to call up the System menu.

▶ Open the password menu via the menu option SECURITY…

```
CHANGE PW
ACTIVATE
```

MFD will only show this password menu if a password is present.

→ Make a note of the password before you activate it. If the password is no longer known, MFD can be unlocked (DELETE FUNCTION is not active), however, the circuit diagram and data settings will be lost.

Attention!
If the password is unknown or lost, and the delete password function is deactivated: The unit can only be reset to the factory setting by the manufacturer. The program and all data are lost.

▶ Select ACTIVATE PW and confirm with **OK**.

The password is now active. MFD will automatically return to the Status display.

You must unlock MFD with the password before you implement a protected function, enter a protected menu or the System menu.

Unlocking MFD

Unlocking MFD will deactivate the password. You can reactivate password protection later via the password menu or by switching the power supply off and on again.

▶ Press **OK** to switch to the main menu.

The PASSWORD… entry will flash.

```
PASSWORD...
STOP   RUN  √
PASSWORD...
SET CLOCK...
```

▶ Press **OK** to enter the password entry menu.

→ If MFD shows PROGRAM… in the main menu instead of PASSWORD…, this means that there is no password protection active.

```
ENTER PASSWORD
     XXXXXX
```

MFD will display the password entry field.

▶ Set the password using the cursor buttons.

▶ Confirm with **OK**.

If the password is correct, MFD will return automatically to the Status display.

```
PROGRAM...
STOP
PARAMETERS
SET CLOCK...
```

The PROGRAM… menu option is now accessible so that you can edit your circuit diagram.

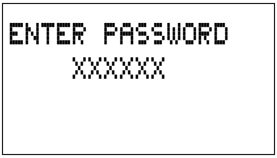
The System menu is also accessible.

Changing or deleting the password range

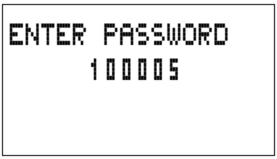
- ▶ Unlock MFD.
- ▶ Press **DEL** and **ALT** to call up the System menu.
- ▶ Open the password menu via the menu option SECURITY… and PASSWORD….

The CHANGE PW entry will flash.

MFD will only show this password menu if a password is present.



- ▶ Press **OK** to enter the password entry menu.
- ▶ Use **OK** to move to the 6-digit entry field.
- ▶ The current password will be displayed.



- ▶ Modify the six password digits using the cursor buttons.
- ▶ Confirm with **OK**.

Use **ESC** to exit the security area.

Deleting

Use number combination 000000 to delete a password.

If a password has not been entered already, MFD will show six dashes:

Password incorrect or no longer known

If you no longer know the exact password, you can try to re-enter the password several times.

→ The DELETE FUNCTION has not been deactivated.

```
ENTER PASSWORD
     XXXXXX
```

You have entered an incorrect password?

▶ Re-enter the password.

```
DELETE ALL?
```

After the fourth entry attempt MFD will ask whether you wish to delete the circuit diagram and data.

▶ Press
- **ESC**: No data will be deleted.
- **OK**: Circuit diagram, data and password are deleted.

MFD will return to the Status display.

→ If you no longer know the exact password, you can press **OK** to unlock the protected MFD. The saved circuit diagram and all function relay parameters will be lost.

Pressing **ESC** will retain the circuit diagram and data. You can then make another four attempts to enter the password.

Changing the menu language

MFD-Titan provides ten menu languages which are set as required via the System menu.

Language	Display
English	ENGLISH
German	DEUTSCH
French	FRANCAIS
Spanish	ESPANOL
Italian	ITALIANO
Portuguese	PORTUGUES
Dutch	NEDERLANDS
Swedish	SVENSKA
Polish	POLSKI
Turkish	TURKCE

→ Language selection is only possible if MFD is not password-protected.

▶ Press **DEL** and **ALT** to call up the System menu.

▶ Select MENU LANGUAGE… to change the menu language.

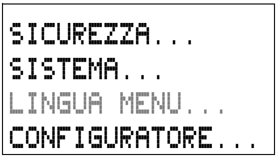
```
ENGLISH      ↑
DEUTSCH     √
FRANCAIS
ESPANOL      ↓
ITALIANO
PORTUGUES
NEDERLAND
SVENSKA
POLSKI
TURKCE
```

The language selection for the first entry ENGLISH is displayed.

▶ Use ︿ or ﹀ to select the new menu language, e.g. Italian (ITALIANO).

▶ Confirm with **OK**. ITALIANO is assigned a tick.

▶ Exit the menu with **ESC**.



MFD will now show the new menu language.

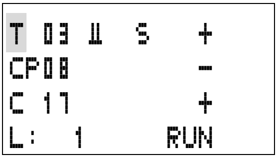
Press **ESC** to return to the Status display.

Changing parameters

MFD allows you to change function relay parameters such as timing relay setpoint values and counter setpoints without having to call up the circuit diagram. This is possible regardless of whether MFD is running a program or is in STOP mode.

▶ Press **OK** to switch to the main menu.

▶ Start the parameter display by selecting PARAMETERS.

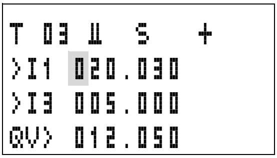


All function blocks are displayed as a list.

The following preconditions must be fulfilled in order for a parameter set to be displayed:

- A function relay must have been included in the circuit diagram.
- The PARAMETERS menu must be available.
- The parameter set must have been enabled for access, indicated by the + character at the bottom right of the display.

→ Parameter sets can only be enabled or protected via the FUNCTION RELAYS menu, or via the circuit diagram with the "+" enable and with "–" inhibit parameter set characters.



- ▶ Select the required function block with ∧ or ∨.
- ▶ Press the **OK** button.
- ▶ Scroll with the ∧ or ∨ cursor buttons through the constants of the function block inputs.
- ▶ Change the values for a parameter set:
  - – With **OK** in the Entry mode,
  - – 〈 〉 change decimal place,
  - – ∧∨ change the value of a decimal place,
  - – **OK** save constants or
  - – **ESC** Retain previous setting.

Press **ESC** to leave the parameter display.

→ Only constants on the function block inputs can be changed.

Adjustable parameters for function blocks

You can modify the function blocks used in the circuit diagram in three different ways:

- All circuit diagram parameters can be adjusted in STOP mode via the function block editor.
- Setpoints (constants) can be modified in RUN mode via the function block editor.
- Setpoints (constants) can be modified via the PARAMETERS menu.

Adjustable setpoint values are:

- The inputs with all function blocks if constants have been used.
- Switch on and off times with time switches.

In RUN mode MFD operates with a new setpoint as soon as it has been modified in the parameter display and saved with **OK**.

Setting date, time and daylight saving time

The easy800 devices are equipped with a real-time clock with date and time functions. The “time switch” function block can be used to implement time switch applications.

If the clock has not yet been set or if MFD is switched on after the buffer time has elapsed, the clock starts with the setting “WE 1:00 01.05.2002”. The MFD clock operates with date and time so the hour, minute, day, month and year must all be set.

→ The time, such as: 1:00, indicates the version of the device operating system.

▶ Select SET CLOCK… from the main menu.

This will open the menu for setting the time.

```
SET CLOCK
DST SETTING
```

▶ Select SET CLOCK.

```
HH:MM    00:27
DD.MM   05.05
YEAR :    2002
```

▶ Set the values for day, time, month and year.

▶ Press the **OK** button to access the Entry mode

– ‹ › Move between the parameters,

– ⌃⌄ Change the value,

– **OK** Save day and time,

– **ESC** Retain previous setting.

Press **ESC** to leave the time setting display.

Changing between winter/ summer time (DST)

The easy800 models are fitted with a real-time clock. The clock has various possibilities for changing the DST setting. These are subject to different legal requirements in the EU, GB and USA.

→ The time change algorithm only applies to the northern hemisphere.

- NONE: no daylight saving time setting.
- MANUAL: a user-defined date for the DST change.
- EU: date defined by the European Union; Commences: last Sunday in March; Ends: last Sunday in October.
- GB: date defined in Great Britain; Commences: last Sunday in March; Ends: fourth Sunday in October.
- US: date defined in the United States of America: Commences: first Sunday in April; Ends: last Sunday in October.

The following applies to all DST variants:

Winter time → Summer time: On the day of conversion, the clock moves forward one hour at 2:00 to 3:00

Summer time → Winter time: On the day of conversion, the clock moves back one hour at 3:00 to 2:00.

Select SET CLOCK… from the main menu.

```
SET CLOCK
DST SETTING
```

This will open the menu for setting the time.

▶ Select the DST SETTING menu option.

Selecting DST

MFD shows you the options for the DST change.

The standard setting is NONE for automatic DST changeover (Tick at NONE).

```
NONE          √  ↑
MANUAL
EU
GB               ↓
US
```

▶ Select the required variant and press the **OK** button.

Manual selection

```
SUMMERTIME START
 DD.MM 00.00
SUMMERTIME END
 DD.MM:  00:00
```

You want to enter your own date.

→ The following applies to MFD-Titan devices:

The time change algorithm always calculates the date from the year 2000. Enter the time change for the year 2000.

▶ Proceed to the MANUAL menu and press 2 × **OK.**

- ‹ › Move between the parameters,
- ⌃⌄ Change the value,
- **OK** Save day and time,
- **ESC** Retain previous setting.

▶ Press **ESC** to leave the display.

▶ Select the day and time at which summer time is to commence.

▶ Select the day and time at which summer time is to end.

→ The same time for conversion applies as for the legally determined variants (EU, GB, US).

Activating input delay (debounce)

Input signals can be evaluated by MFD with a debounce delay. This enables, for example, the trouble-free evaluation of switches and pushbutton actuators subject to contact bounce.

In many applications, however, very short input signals have to be monitored. In this case, the debounce function can be switched off.

▶ Press **DEL** and **ALT** to call up the System menu.

▶ Select the SYSTEM menu.

→ If MFD is password-protected you cannot open the System menu until you have “unlocked” it.

```
DEBOUNCE √     ↑
P BUTTONS
RUN MODE
CARD MODE      ↓
```

The input delay (debounce) is set with the DEBOUNCE menu item.

```
DEBOUNCE √     ↑
P BUTTONS
RUN MODE
CARD MODE      ↓
```

Activating debounce

If a tick √ is set beside DEBOUNCE, the input delay is set.

If this is not so, proceed as follows:

▶ Select DEBOUNCE and press **OK.**

Debounce mode will be activated and the display will show DEBOUNCE √.

Press **ESC** to return to the Status display.

Deactivating debounce (input delay)

If MFD is showing DEBOUNCE in the display, this means that Debounce mode has already been deactivated.

▶ Otherwise select DEBOUNCE √ and press **OK.**

Debounce mode will be deactivated and the display will show DEBOUNCE.

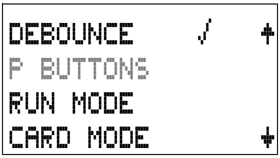
→ How MFD input and output signals are processed internally is explained in Section "Delay times for inputs and outputs", from Page 392.

Activating and deactivating the P buttons

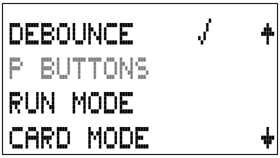
Even though the cursor buttons (P buttons) have been set as pushbutton actuator inputs in the circuit diagram, this function is not activated automatically. This prevents any unauthorised use of the cursor buttons. The P buttons can be activated in the System menu.

→ If MFD is password-protected you cannot open the System menu until you have "unlocked" it.

The P buttons are activated and deactivated via the P BUTTONS menu.



- ▶ Press **DEL** and **ALT** to call up the System menu.
- ▶ Select the SYSTEM menu.
- ▶ Move to the cursor to the P BUTTONS menu.



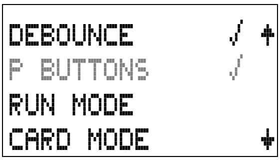
Activating the P buttons

If MFD is displaying P BUTTONS √, this means that the P buttons are active.

- ▶ Otherwise select P BUTTONS and press **OK.**

MFD changes the display to P BUTTONS √ and the P buttons are activated.

- ▶ Press **ESC** to return to the Status display.



The P buttons are only active in the Status display and the text display. In this display you can use the P buttons to activate inputs in your circuit diagram.

Deactivating the P buttons

▶ Select P BUTTONS √ and press **OK**.

MFD changes the display to P BUTTONS and the P buttons are deactivated.

→ The P buttons are automatically deactivated when loading a circuit diagram from the memory card or via EASY-SOFT (-PRO) to MFD, or when deleting a circuit diagram in MFD.

Startup behaviour

The startup behaviour is an important aid during the commissioning phase. The circuit diagram which MFD contains is not as yet fully wired up or the system or machine is in a state which MFD is not permitted to control. The outputs should not be controlled when MFD is switched on.

Setting the startup behaviour

→ The MFD devices without a display can only be started in RUN mode.

Requirement: MFD must contain a valid circuit diagram.

▶ Switch to the System menu.

→ If MFD is protected by a password, the System menu will not be available until MFD is "unlocked" (see → Section "Unlocking MFD" from Page 363).

Specify the operating mode which MFD must use when the supply voltage is applied.

Activating RUN mode

If MFD displays RUN MODE √, this means that MFD will start in Run mode when the supply voltage is applied.

```
DEBOUNCE      √ ↑
P BUTTONS
RUN MODE      √
CARD MODE       ↓
```

▶ Otherwise select RUN MODE and press **OK.**

RUN mode is activated.

▶ Press **ESC** to return to the Status display.

```
DEBOUNCE    √   ↑
P BUTTONS
RUN MODE
CARD MODE       ↓
```

Deactivating RUN mode

▶ Select RUN MODE √ and press **OK**.

The RUN mode function is deactivated.

The default setting for MFD is for RUN MODE to be displayed. In other words, MFD starts in RUN MODE √ when the power is switched on.

Startup behaviour	Menu display	Status of MFD after startup
MFD starts in STOP mode	RUN MODE	MFD is in STOP mode
MFD starts in RUN mode	RUN MODE √	MFD is in RUN mode

Behaviour when the circuit diagram is deleted

The setting for the startup behaviour is an MFD device function. When the circuit diagram is deleted, this does not result in the loss of the setting selected.

Behaviour during upload/download to card or PC

When a valid circuit diagram is transferred from MFD to a memory card or the PC or vice versa, the setting is still retained.

→ The MFD devices without a display can only be started in RUN mode.

Possible faults

MFD will not start in RUN mode:

- a program is not available in MFD.
- you have selected MFD startup in STOP MODE (RUN MODE menu).

Card startup behaviour

The startup behaviour with memory card is for applications where unskilled personnel change the memory card under no-voltage conditions.

MFD only starts in the Run mode if a memory card with a valid program is inserted.

If the program on the memory card is different to the program in MFD, the program from the card is loaded into easy and easy starts in RUN mode.

▶ Switch to the System menu.

→ If MFD is protected by a password, the System menu will not be available until MFD is "unlocked" (see → Section "Unlocking MFD" from Page 363).

Activation of card mode

Requirement: RUN MODE is active.

If MFD displays CARD MODE √, this means that when the power supply is switched on, MFD will only start in RUN mode if a memory card with a valid program has been inserted.

```
DEBOUNCE      √ ↕
P BUTTONS
RUN MODE      √
CARD MODE     √ ↕
```

▶ Otherwise select CARD MODE and press **OK**.
MFD will start up with the program on the card.

▶ Press **ESC** to return to the Status display.

```
DEBOUNCE    √ ↑
P BUTTONS
RUN MODE    √
CARD MODE     ↓
```

Deactivation of card mode

▶ Select CARD MODE √ and press **OK**.

The RUN mode function is deactivated.

The MFD default setting is for display of the CARD MODE menu, i.e. MFD starts in RUN mode without the memory card when the power is switched on.

Terminal mode

The MFD-Titan also supports the TERMINAL MODE. Terminal mode enables the display and the keypad of the MFD to be used as a terminal for operating another device. In this operating mode you are thus able to remotely control all devices supporting Terminal mode operation. The interface to the other device can be implemented using the serial interface or easy-NET.

→ TERMINAL MODE operation is only possible if the MFD is in STOP mode.

Permanent TERMINAL MODE setting

In the SYSTEM menu you set the MFD to start in TERMINAL MODE when the power supply is switched on.

▶ Switch to the System menu.

→ If MFD is protected by a password, the System menu will not be available until MFD is "unlocked" (see → Section "Unlocking MFD" from Page 363).

Activating an automatic startup in TERMINAL MODE

Requirement: The MFD is in RUN or STOP mode without visualization (the System menu must be reachable).

```
P BUTTONS      ↑
RUN MODE
CARD MODE
TERMINAL MODE √ ↓
```

- ▶ Select TERMINAL MODE in the System menu and press **OK**.
- ▶ The next time that the MFD is started, it will establish the connection to the selected device.
- ▶ Press **ESC** to return to the Status display.

→ The correct station number must be selected in order for the MFD to start TERMINAL MODE with the correct station. (→ chapter "Commissioning", Page 85)

```
P BUTTONS      ↑
RUN MODE
CARD MODE
TERMINAL MODE   ↓
```

Deactivating an automatic startup in TERMINAL MODE

▶ Select TERMINAL MODE √ and press **OK**.

The automatic starting in TERMINAL MODE has been deactivated.

The default setting of the MFD is for the display of the TERMINAL MODE menu, i.e. MFD starts in RUN or STOP mode when the power is switched on.

Setting LCD contrast and backlight

The backlight of the LCD display can be set to one of five stages in order to adapt it to local conditions. The display contrast can be set to one of five stages.

The contrast and backlight settings are implemented as device settings.

▶ Switch to the System menu.

→ If MFD is protected by a password, the System menu will not be available until MFD is "unlocked" (see → Section "Unlocking MFD" from Page 363).

```
SECURITY           ↑
SYSTEM...
MENU LANGUAGE
CONFIGURATOR..     ↓
```

▶ Select the SYSTEM menu.

▶ Press the **OK** button.

```
RUN MODE           ↑
CARD MODE
TERMINAL MODE
DISPLAY...         ↓
```

▶ Use the ﹀ button to select the DISPLAY menu and press **OK**.

```
CONTRAST:         0
LIGHTING:       75%
```

The menus for setting the contrast and backlight are displayed.

▶ Press the **OK** button and move to the contrast entry field

Use the ︿ and ﹀ cursor buttons to set the contrast to a value between –2 and +2.

```
CONTRAST:        +1
LIGHTING:       75%
```

▶ Select your setting.

▶ Complete your setting by pressing **OK**.

```
CONTRAST:        +1
LIGHTING:       75%
```

The contrast setting will be retained until it is modified.

```
CONTRAST:        +1
LIGHTING:       75%
```

▶ Use the cursor buttons ︿ and ﹀ to move to the LIGHTING menu.

▶ Press the **OK** button.

```
CONTRAST:        +1
LIGHTING        75%
```

▶ Use cursor buttons ︿ and ﹀ to change the value in 25 % steps.

▶ Set the required backlighting.

→ The backlight will immediately change to the set value. 0, 25, 50, 75 and 100 % are possible values.

```
CONTRAST:        +1
LIGHTING       100%
```

→ MFD comes with the following factory setting:

The contrast is set to 0.

The backlight is set to 75 %.

Retention

It is a requirement of system and machine controllers for operating states or actual values to have retentive settings. What this means is that the values will be retained safely even after the supply voltage to a machine or system has been switched off and are also retained until the next time the actual value is overwritten.

The following operands and function blocks can be set to operate retentively:

- Markers
- Counter function blocks,
- Data function blocks and
- Timing relays.

Operating hours counter

easy800 provides four retentive operating hours counters. They are always retentive and can only be selectively deleted with a reset command.

Retentive data volume

200 bytes is the maximum memory range for retentive data (operating hours counters are not included).

Markers

A user-definable and consistent marker range can be declared as retentive.

Counters

All C.., CH.. and CI.. function blocks can be operated with retentive actual values.

Data function blocks

A user-definable consistent data function block range can be operated with retentive actual values.

Timing relays

A user-definable and consistent range for timing relays can be run with retentive actual values.

Requirements

In order to make data retentive, the relevant markers and function blocks must have been declared as retentive.

Attention!
The retentive data is saved every time the power supply is switched off, and read every time the device is switched on. The data integrity of the memory is guaranteed for 10^{10} read/write cycles.

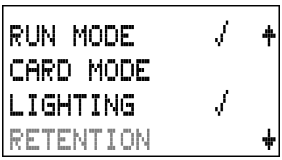
Setting retentive behaviour

Requirement:
MFD must be in STOP mode.

▶ Switch to the System menu.

If MFD is protected by a password, the System menu will not be available until MFD is "unlocked" (see → Section "Unlocking MFD" from Page 363).

The default setting of MFD is for no retentive actual value data to be selected. When MFD is in STOP mode or has been switched to a de-energized state, all actual values are cleared.



▶ Switch to STOP mode.

▶ Switch to the System menu.

▶ Proceed to the SYSTEM menu and continue to the RETENTION… menu.

▶ Press the **OK** button.

```
MB 00 -> MB 00 ↑
C  00 -> C  00
CH 00 -> CH 00 ↓
          B:200
```

The first screen display is the selection of the marker range.

▶ ∧∨ Select a range.

▶ Press **OK** to access the Entry modes.

– ‹ › Select a position from/to,

– ∧∨ Set a value.

▶ Save the input from .. to .. with **OK**.

Press **ESC** to exit the input for the retentive ranges.

Up to six different ranges can be selected.

```
CI 00 -> CI 00 ↑
DB 00 -> DB 00
T  00 -> T  00 ↓
          B:200
```

→ The display on the lower right B:200 indicates the number of free bytes.

```
MB 01 -> MB 04
C  12 -> C  16
CH 00 -> CH 00
CI 00 -> CI 00
DB 01 -> DB 16
T  26 -> T  32
          B:076
```

Example:
MB 01 to MB 04, C 12 to C 16, DB 01 to DB 16, T 26 to T 32 should be retentive.

124 bytes have been assigned to the retentive data range. 76 bytes are still available.

Deleting ranges

Set the ranges to be erased to the values from 00 to 00.

e.g.: MB 00 -> MB 00. The markers are no longer retentive.

Deleting retentive actual values of markers and function blocks

The retentive actual values are cleared if the following is fulfilled (applies only in STOP mode):

- When the circuit diagram is transferred from EASY-SOFT(-PRO) (PC) or from the memory card to MFD, the retentive actual values are reset to 0. This also applies when there is no program on the memory card. In this case the old circuit diagram is retained in MFD.
- When changing the respective retentive range.
- When the circuit diagram is deleted via the DELETE PROGRAM menu.

Transferring retentive behaviour

The setting for retentive behaviour is a circuit diagram setting. In other words, the setting of the retentive menu may also under certain circumstances be transferred to the memory card or by uploading or downloading from the PC.

Changing the operating mode or the circuit diagram
When the operating mode is changed or the MFD circuit diagram is modified, the retentive data is normally saved together with their actual values. The actual values of relays no longer being used are also retained.

Changing the operating mode
If you change from RUN to STOP and then back to RUN, the actual values of the retentive data will be retained.

Modifying theMFD circuit diagram
If a modification is made to the MFD circuit diagram, the actual values will be retained.

Changing the startup behaviour in the SYSTEM menu
The retentive actual values are retained in MFD regardless of the setting.

Modification of the retentive range
If the set retentive ranges are reduced, only the actual values saved in the range will remain.

If the retentive range is extended, the older data is retained. The new data is written with the current actual values in RUN mode.

Displaying device information

Device information is provided for service tasks and for determining the capability of the device concerned.

This function is only available with devices featuring a display.

Exception: Terminal mode with MFD-Titan.

easy800 enables the display of the following device information:

- AC or DC power supply,
- T (transistor output) or R (relay output),
- C (clock provided),
- A (analog output provided),
- LCD (display provided),
- easy-NET (easy-NET provided),
- OS: 1.10.204 (operating system version),
- CRC: 25825 (checksum of the operating system).

▶ Switch to the System menu.

→ If MFD is protected by a password, the System menu will not be available until MFD is “unlocked” (see → Section “Unlocking MFD” from Page 363).

```
SECURITY
SYSTEM...
MENU LANGUAGE
CONFIGURATOR...
```

▶ Select the SYSTEM menu.

▶ Press the **OK** button.

```
CARD MODE
DISPLAY...
RETENTION...
INFORMATION...
```

▶ Use the ﹀ button to select the INFORMATION menu and press **OK**.

This will display all device information.

```
DC TCA LCD NET
OS : 1.11.111
CRC: 63163
```

Example: MFD-80-B, MFD-CP8-NT, MFD-TA17

```
DC RC  LCD
OS : 1.11.111
CRC: 63163
```

Example: MFD-80-B, MFD-CP8-ME, MFD-R16

Display in STOP mode.

```
DC RC  LCD
OS : 1.11.111
CRC: - - -
```

Display in RUN mode.

The CRC checksum is not displayed.

▶ Press **ESC** to leave the display.

```
CARD MODE        ↑
DISPLAY...
RETENTION...
INFORMATION...   ↓
```

8 Inside MFD

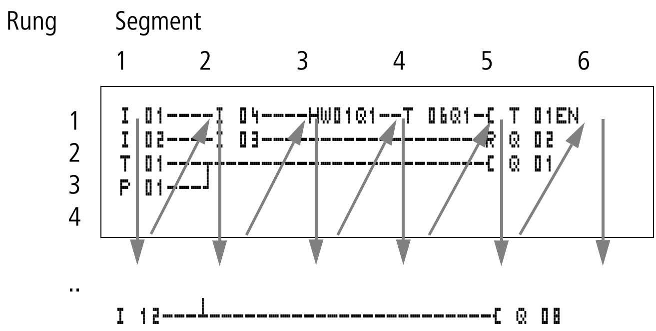
MFD Program cycle

In conventional control systems, a relay or contactor control processes all the rungs in parallel. The speed with which a contactor switches is thus dependent on the components used, and ranges from 15 to 40 ms for relay pick-up and drop-out.

With MFD the circuit diagram is processed with a microprocessor that simulates the contacts and relays of the circuit concerned and thus processes all switching operations considerably faster. Depending on its size, the MFDcircuit diagram is processed cyclically every 0.1 to 40 ms.

During this time, MFD passes through six segments in succession.

How MFD evaluates the circuit diagram:



In the first four segments MFD evaluates the contact fields in succession. MFD checks whether contacts are switched in parallel or in series and saves the switching states of all contact fields.

In the fifth segment, MFD assigns the new switching states to all the coils in one pass.

The sixth segment is located outside of the circuit diagram. MFD uses this to:

Evaluating function blocks

- process the function blocks which are used: the output data of a function block is updated immediately after processing. MFD processes the function blocks according to the function block list (→ FUNCTION RELAYS menu) from top to bottom. You can sort the function block list with EASY-SOFT (-PRO) from Version 4.04. You can then, for example, use the results consecutively.
- establish contact to the "outside world": The output relays Q 01 to Q (S).. are switched and the inputs I1 to I (R).. are read once more.
- MFD also copies all the new switching states to the status image register.
- exchange all data on the easy-NET network (read and write).

MFD only uses this status image for one cycle. This ensures that each rung is evaluated with the same switching states for one cycle, even if the input signals I1 to I12 change their status several times within a cycle.

→ The following must be observed when operating a PID controller function block!

The cycle time of the program must be less than the scan time of the controller. If the cycle time is greater than the controller scan time, the controller will not be able to achieve constant results.

COM-LINK data access during the program cycle

The data exchange with the point-to-point connection can be carried out in any segment of the program cycle. This data exchange increases the cycle time with both active and remote stations. Only use data that is absolutely necessary.

Loading visualization data

When setting a program to RUN that contains visualization data, the contents of the screens to be displayed have to be loaded.

The time required for loading in the event of a screen change depends on the size of the screens to be loaded. During a screen change, the new screen is loaded from the screen memory into the RAM. Every 200 ms MFD-Titan checks whether a new screen has to be loaded.

The loading time can be calculated as follows:
screen size in byte multiplied by 80 µs.

Example:

Screen size 250 bytes:

The loading time for the screen is: 250 × 80 µs = 20 ms

→ If you require the MFD to have a small cycle time:

use several small screens so that the loading time is not too long during a screen change. Only display necessary information in the screens concerned (→ Section "Memory division"Page 280).

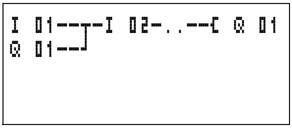
The loading of screen data and screen changes can be implemented in any segment of the program cycle. Take this behaviour into account when considering the reaction time of your entire control system.

→ Distribute tasks amongst several devices in the easy-NET. easy800 for open and closed-loop control functions, MFD-Titan for display and operator functions.

Effects on the creation of the circuit diagram

MFD evaluates the circuit diagram in these six segments in succession. You should take into account two factors when creating your circuit diagram.

- The changeover of a relay coil does not change the switching state of an associated contact until the next cycle starts.
- Always wire forward or from top to bottom. Never work backwards.

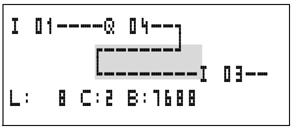


Example: self-latching with own contact

Start condition:
Inputs I1 and I2 are switched on.
Q1 is switched off.

This is the circuit diagram of a self-latching circuit. If I1 and I2 are closed, the switching state of relay coil Ĉ Q 01 is "latched" via contact Q 01.

- **1st cycle:** Inputs I1 and I2 are switched on. Coil Q1 picks up.
- Contact Q 01 remains switched off since MFD evaluates from left to right. The first coil field was already passed when MFD refreshes the output image in the 6th segment.
- **2nd cycle:** The self-latching now becomes active. MFD has transferred the coil states at the end of the first cycle to contact Q 01.



Example: Do not wire backwards

This example is shown in Section "Creating and modifying connections"Page 141. It was used here to illustrate how NOT to do it.

In the third rung, MFD finds a connection to the second rung in which the first contact field is empty. The output relay is not switched.

When wiring more than four contacts in series, use one of the marker relays.

```
I 01----Q 04----Ī 03----I 05---[ M 01
I 02----I 04----M 01-----------[ Q 02
```

Figure 188: Circuit diagram with M 01 marker relay

How MFD evaluates the high-speed counters CF, CH and CI

In order to evaluate the count pulses of 3 kHz, the high-speed counter function blocks operate with an interrupt routine. The length of the circuit diagram and the associated cycle time has no effect on the counter result.

Memory management of the MFD-Titan

The MFD is provided with different memories.

- The working memory or RAM, size 8 KByte
  The RAM only stores the data when the device power supply is active.
- The screen memory, size 24 KByte
  The screen memory stores the visualization data created with EASY-SOFT-PRO retentively.
- The program memory, size 8 KByte
  The program memory stores the program retentively.

Distribution of data in the RAM

When the power supply is switched on, the RAM stores the program, the retentive data and the screens to be displayed. This has a direct effect on the size of the program and the screens. The number of retentive data bytes reduces the memory available for program and screens. The largest screen to be displayed likewise reduces the memory available for the program.

→ Only use as much retentive data as is actually required.

The screen with the largest memory requirement reduces the memory available for the program. Several smaller screens allow more space for the program.

Use as small pictures as possible with 1 bit greyscale. The pictures should normally be 32 x 32 pixels in size in order to fully utilise the optimum brilliance of the display.

Delay times for inputs and outputs

The time from reading the inputs and outputs to switching contacts in the circuit diagram can be set in MFD via the delay time.

This function is useful, for example, in order to ensure a clean switching signal despite contact bounce.

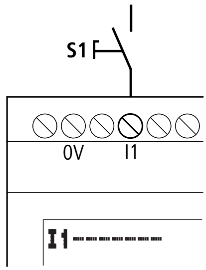


Figure 189: MFD input assigned a switch

Delay times for the MFD inputs

The delay time for DC signals is 20 ms.

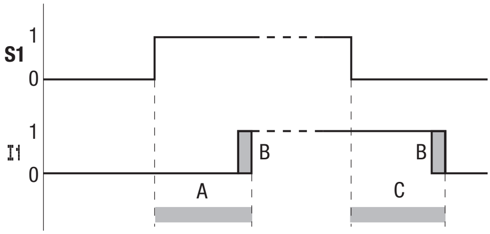


Figure 190: Delay times for MFD-DC

An input signal S1 must therefore be 15 V or 8 V for at least 20 ms on the input terminal before the switch contact will change from 0 to 1 (A). If applicable, this time must also include the cycle time (B) since MFD does not detect the signal until the start of a cycle.

The same debounce delay (C) applies when the signal drops out from 1 to 0.

→ If you use high-speed counter function blocks, the debounce delay time for the inputs is 0.025 ms. Otherwise it is not possible to count high-speed signals.

If the debounce is switched off, MFD responds to an input signal after just 0.25 ms.

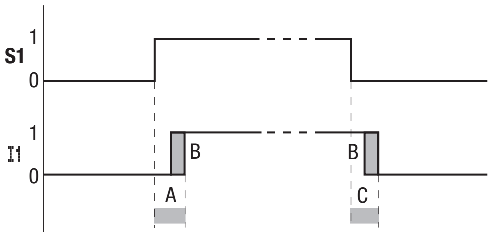


Figure 191: Switching behaviour with input debounce disabled

Typical delay times with the debounce delay disabled are:

- On-delay for
  - I1 to I4: 0.025 ms
  - I5 to I12: 0.25 ms
- Off-delay for
  - I1 to I4: 0.025 ms
  - I5, I6 and I9 to I10: 0.4 ms
  - I7, I8, I11 and I12: 0.2 ms

→ Ensure that input signals are noise-free if the input debounce is disabled. MFD will even react to very short signals.

Monitoring of short-circuit/overload with EASY..-D.-T..

Depending on the type of easy in use, it is possible to use the internal inputs I16, R15 and R16 to monitor for short-circuits or overloads on an output.

- MFD-Titan:
  - I16: Group fault signal for outputs Q1 to Q4.
- EASY620-D.-TE:
  - R16: Group fault signal for outputs S1 to S4.
  - R16: Group fault signal for outputs S5 to S8.

State Outputs	I16, R15 or R16
No fault found	0 = switched off (make contact)
At least one output has a fault	1 = switched on (make contact)

→ I16 can only be edited with MFD versions which have transistor outputs.

The following examples are for I16 = Q1 to Q4.

Example 1: Selecting an output with fault indication

```
I 01----M̄ 16--------------------[ Q 01
I 16----------------------------S M 16
```

Figure 192: Circuit diagram for fault output via I16

The above circuit diagram functions as follows:

If a transistor output reports a fault, M16 is set by I16. The break contact of M16 switches off output Q1. M16 can be cleared by resetting the MFD power supply.

Example 2: Output of operating state

```
I 01----M̄ 16--------------------[ Q 01
I 16----------------------------S M 16
M 16----------------------------[ Q 04
```

Figure 193: Output of operating state

The above circuit operates as described in example 1. The signal light is triggered at Q4 for additional overload monitoring. If Q4 has an overload, it would “pulse”.

Example 3: Automatic reset of error signal

```
I 01----M̄ 16--------------------[ Q 01
I 16----------------------------S M 16
M 16----------------------------[ T 08EN
T 08----------------------------R M 16
M 16----------------------------[ Q 04
```

Figure 194: Automatic reset of error signal

The above circuit diagram functions in the same way as Example 2. In addition the marker M16 is reset every 60 seconds by timing relay T08 (on-delayed, 60 s). Should I16 remain at 1, M16 will continue to be set. Q1 is set briefly to 1 until I16 switches off again.

Expanding MFD-Titan

You can expand MFD-Titan with easy models EASY618-..-RE, EASY620-D.-TE, EASY202-RE locally or use the EASY200-EASY coupling module for remote expansion with easy600 expansion. All available bus interface devices such as EASY204-DP, EASY221-CN, EASY205-ASI or EASY222-DN can be used if present.

Install the units and connect the inputs and outputs as described (see → Section "Connecting the expansion unit"Page 49).

You process the inputs of the expansion devices as contacts in the MFD circuit diagram in the same way as you process the inputs of the basic unit. The input contacts are assigned the operand designations R1 to R12.

R15 and R16 are the group fault inputs of the transistor expansion unit (→ Section "Monitoring of short-circuit/ overload with EASY..-D.-T..", Page 395).

The outputs are processed as relay coils or contacts like the outputs in the basic unit. The output relays are S1 to S8.

→ EASY618-..-RE provides the outputs S1 to S6. The other outputs S7, S8 can be used internally.

How is an expansion unit recognised?

If at least one R .. contact or S .. coil/contact is used in the circuit diagram, the basic unit assumes that an expansion unit is connected.

Transfer behaviour

The input and output data of the expansion units is transferred serially in both directions. Take into account the modified reaction times of the inputs and outputs of the expansion units.

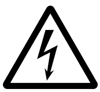
Input and output reaction times of expansion units
The debounce setting has no effect on the expansion unit.

Transfer times for input and output signals:

- Local expansion
  - Time for inputs R1 to R12: 30 ms + 1 cycle
  - Time for outputs S1 to S6 (S8): 15 ms + 1 cycle
- Decentralised expansion
  - Time for inputs R1 to R12: 80 ms + 1 cycle
  - Time for outputs S1 to S6 (S8): 40 ms + 1 cycle

Function monitoring of expansion units

If the power supply of the expansion unit is not present, no connection can be established between it and the basic unit. The expansion inputs R1 to R12, R15, R16 are incorrectly processed in the basic unit and show status 0. It cannot be assured that the outputs S1 to S8 are transferred to the expansion unit.



Warning!
Continuously monitor the functionality of the MFD expansion in order to prevent switching errors in the machine or plant.

The status of the internal input I14 of the basic unit indicates the status of the expansion unit:

- I14 = “0”: expansion unit is functional,
- I14 = “1”: expansion unit is not functional.

Example

Power can be applied to the expansion unit later than the basic unit. This means that the basic unit is switched to RUN when the expansion unit is absent. The following MFD circuit diagram detects if the expansion unit is functional or not functional.

```
I 14----M̄ 01--------------------Ĺ : 08
------------------------------S M 01
I 14----------------------------Ĺ : 08
I 01---┬-I 02-------------------Ĺ Q 01
Q 01---┘
```

Figure 195: Circuit diagram for expansion testing

As long as I 14 is 1, the remaining circuit diagram is skipped. If I 14 is 0, the circuit diagram is processed. If the expansion unit drops out for any reason, the circuit diagram is skipped. M 01 detects whether the circuit diagram was processed for at least one cycle after the power supply is switched on. If the circuit diagram is skipped, all the outputs retain their previous state.

QA analog output

The analog output operates with decimal values between 0 and 1023. This corresponds to a 10-bit resolution. At the output this corresponds to a physical voltage between 0 V and 10 V DC.

Negative values such as: –512 are evaluated as zero and output as 0 V DC.

Positive values greater than 1023, such as: 2047, are evaluated as 1023 and output as 10 V DC.

Loading and saving programs

You can either use the MFD interface to save programs to a memory card or use EASY-SOFT-PRO and a transmission cable to transfer them to a PC.

MFD without display and keypad

MFD models without buttons and a display can be loaded with the MFD program via EASY-SOFT-PRO or automatically from the fitted memory card every time the power supply is switched on.

Interface

The MFD interface is covered. Remove the cover carefully.

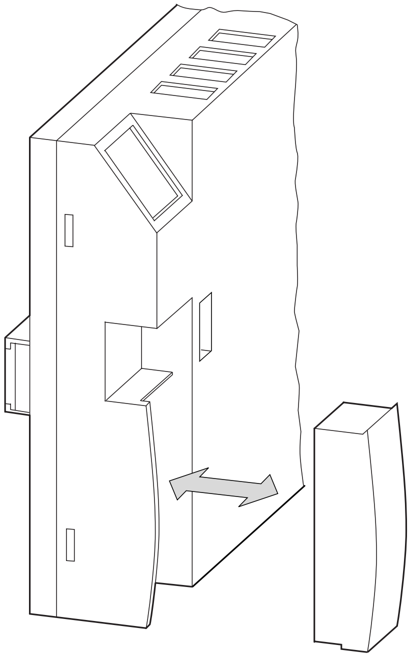


Figure 196: Remove cover and plug-in

▶ To close the slot again, push the cover back onto the slot.

Memory card

The card is available as an accessory EASY-M-256K for MFD-Titan.

Circuit diagrams containing all the relevant data can be transferred from the EASY-M-256K memory card to MFD-Titan.

Each memory card can hold one MFD program.

Information stored on the memory card is “non-volatile” and thus you can use the card to archive, transfer and copy your circuit diagram.

On the memory card you can save:

- the program,
- all the visualization data of the screens,
- all parameter settings of the circuit diagram,
  - the system settings,
  - debounce setting,
  - P buttons,
  - password,
  - retention on/off and range,
  - easy-NET configuration,
  - setting for automatic startup in Terminal mode,
  - COM-LINK settings,
  - DST settings,
  - Card mode.

▶ Insert the memory card in the open interface slot.

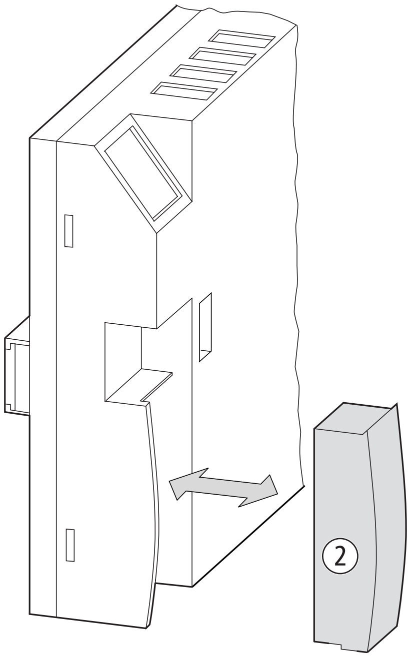


Figure 197: Fitting and removing the memory card

→ With MFD you can insert and remove the memory card even if the power supply is switched on, without the risk of losing data.

Loading or saving circuit diagrams

You can only transfer circuit diagrams in STOP mode.

The MFD versions without a keypad and display automatically transfer the circuit diagram from the inserted memory card to the MFD-CP8... when the power supply is switched on. If the memory card contains an invalid circuit diagram, MFD will keep the circuit diagram still present on the device.

→ If you are using a display without a keypad, load the programs with the EASY-SOFT-PRO software. The function for automatically loading from the memory card on power up is only supported on MFD-CP8.. without display and display operating unit.

```
PROGRAM
DELETE PROG
CARD
```

▶ Switch to STOP mode.

▶ Select PROGRAM… from the main menu.

▶ Select the CARD… menu option.

The CARD… menu option will only appear if you have inserted a functional memory card.

```
DEVICE-CARD
CARD-DEVICE
DELETE CARD
```

You can transfer a circuit diagram from MFDto the card and from the card to the MFD memory or delete the content of the card.

→ If the operating voltage fails during communication with the card, repeat the last procedure since MFDmay not have transferred or deleted all the data.

▶ After transmission, remove the memory card and close the cover.

Saving a circuit diagram on the card

```
REPLACE ?
```

▶ Select DEVICE-CARD.

▶ Confirm the prompt with **OK** to delete the contents of the memory card and replace it with the MFD circuit diagram.

Press **ESC** to cancel.

Loading a circuit diagram from the card

```
DEVICE-CARD
CARD-DEVICE
DELETE CARD
```

▶ Select the CARD-> DEVICE menu option.

▶ Press **OK** to confirm the prompt if you want to delete the MFDmemory and replace it with the card content.

Press **ESC** to cancel.

```
INVALID PROG
```

If there are transmission problems, MFD will display the INVALID PROG message.

This either means that the memory card is empty or that the circuit diagram on the card contains function relays that MFD does not recognise.

The analog value comparator function block is only available on 24 V DC versions of MFD. Programs with visualization components are only supported by the MFD.

→ If the memory card is password-protected, the password will also be transferred to the MFD memory and will be active immediately.

Deleting a circuit diagram on the card

```
DELETE ?
```

▶ Select the DELETE CARD menu option.

▶ Press **OK** to confirm the prompt and to delete the card content.

Press **ESC** to cancel.

Memory card compatibility of the programs

→ Memory cards with programs are always read by MFD-Titan devices with the newer (higher) operating system version. The program is executable. If programs are written with a newer operating system (higher number) on the memory card, this program can only be read and executed by the same version or a higher one.

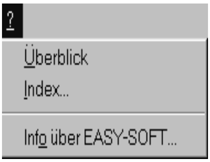
EASY-SOFT-PRO

EASY-SOFT-PRO is a PC program for creating, testing and managing circuit diagrams for MFD.

→ You should only transfer data between the PC and MFD using the MFDPC interface cable, which is available as accessory EASY800-PC-CAB.

→ MFD cannot exchange data with the PC while the circuit diagram display is on screen.

Use EASY-SOFT-PRO to transfer circuit diagrams from your PC to MFD and vice versa. Switch MFD to RUN mode from the PC to test the program using the current wiring.



EASY-SOFT-PRO provides extensive help on how to use the software.

▶ Start EASY-SOFT-PRO and click on Help.

The on-line help provides all additional information about EASY-SOFT-PRO that you will need.

If there are transmission problems, MFD will display the INVALID PROG message.

▶ Check whether you are using functions that the MFD device does not know:

→ If the operating voltage fails during communication with the PC, repeat the last procedure. It is possible that not all the data was transferred between the PC and MFD.

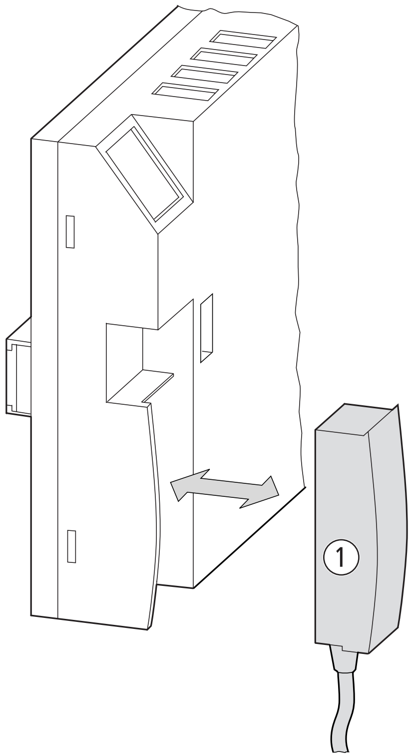


Figure 198: Fitting and removing EASY800-PC-CAB

▶ After transmission, remove the cable and close the cover.

Device version

Every easy800 has the device version number printed on the rear of the device housing. The device version is indicated by the first two digits of the device number.

Example:

```
01-10000003886

DC 20.4 …28.8 V
3 W
```

This device is of device version 01.

The device version provides useful service information about the hardware version and the version of the operating system.

Appendix

Technical data

General

MFD-80.. Display/operating unit	MFD-80..
Front dimensions W × H × D	
With keys [mm]	86.5 × 86.5 × 21.5
[inches]	3.41 × 3.41 × 0.85
Without keys [mm]	86.5 × 86.5 × 20
[inches]	3.41 × 3.41 × 0.79
Total dimensions with fixing shaft W × H × D	
With keys [mm]	86.5 × 86.5 × 43
[inches]	3.41 × 3.41 × 1.69
Thickness of fixing wall (without intermediate top-hat rail) minimum; maximum	
[mm]	1; 6
[inches]	0.04; 0.24
Thickness of fixing wall (with intermediate top-hat rail) minimum; maximum	
[mm]	1; 4
[inches]	0.04; 0.16
Weight	
[g]	130
[lb]	0.287
Mounting	2 22.5 mm (0.886 in) holes Display fastened with two fixing rings
Maximum tightening torque of the fixing rings [Nm]	1.2 to 2

Protective membrane	MFD-XM-80
Dimensions W × H × D	
[mm]	88 × 88 × 25
[inches]	3.46 × 3.46 × 0.98
Weight	
[g]	25
[lb]	0.055
Mounting	Is fitted over the display/keypad (with Titan front ring)

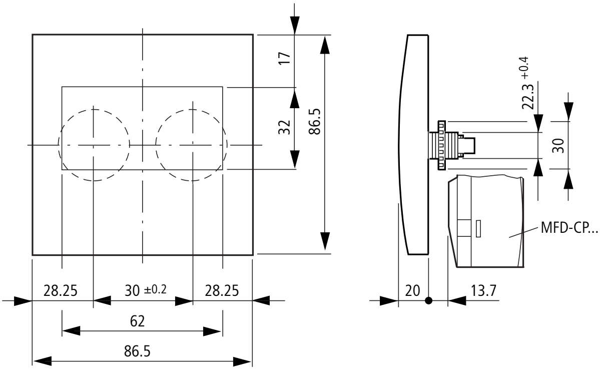
Protective cover	MFD-XS-80
Dimensions W × H × D	
[mm]	86.5 × 94 × 25
[inches]	3.41 × 3.41 × 0.98
Weight	
[g]	36
[lb]	0.079
Mounting	Is fitted over the display/keypad (without Titan front ring)

Power supply/CPU module	MFD-CP8..
Dimensions W × H × D	
[mm]	107.5 × 90 × 30
[inches]	4.23 × 3.54 × 1.18

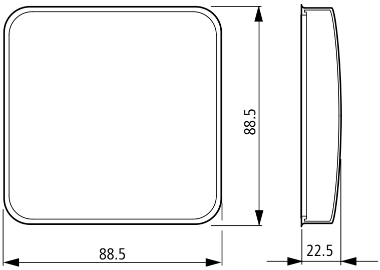
Power supply/CPU module	MFD-CP8..
Weight	
[g]	145
[lb]	0.32
Mounting	Fitted on the fixing shaft of the display or on the top-hat rail to DIN 50022, 35mm (without display) or by means of fixing feet (without display)

Inputs/outputs	MFD-R.., MFD-T..
Dimensions when fitted W × H × D	
[mm]	89 × 90 × 25
[inches]	3.5 × 3.54 × 0.98
Dimensions when removed W × H × D	
[mm]	89 × 90 × 44
[inches]	3.5 × 3.54 × 1.73
Weight	
MFD-R..; MFD-T..[g]	150; 140
MFD-R..; MFD-T..[lb]	0.33; 0.31
Mounting	Snap fitted into the power supply module

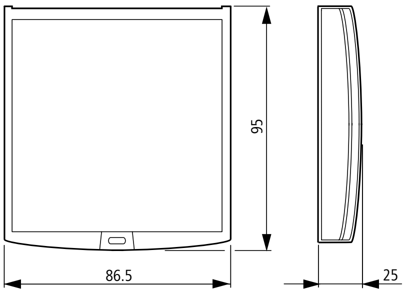
Dimensions of the MFD-80.. display/operating unit



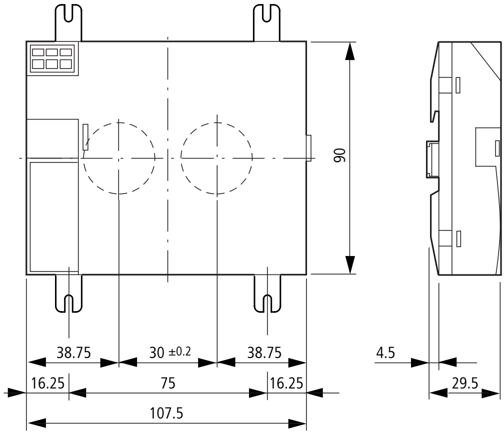
Dimensions of the MFD-80-XM protective membrane



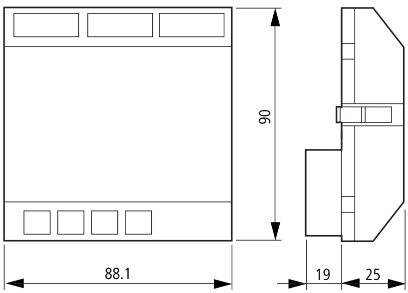
Dimensions of the MFD-80-XS protective cover



Dimensions of the MFD-CP8.. power supply/CPU module.



Dimensions of the MFD-R.. I/O module , MFD-T..



General ambient conditions

Climatic conditions (damp heat constant to IEC 60068-2-78; cyclical to IEC 600618-2-30) (cold to IEC 60068-2-1, heat to IEC 60068-2-2)			
Ambient temperature Installed horizontally/vertically		°C, (°F)	–25 to 55, (–13 to 131)
Condensation (power supply unit/CPU; inputs/outputs)			Prevent condensation with suitable measures
Legibility of the display (–10 to 0 °C with activated backlight, uninterrupted duty)		°C, (°F)	–5 to 50, (–23 to 122)
Storage/transport temperature		°C, (°F)	–40 to 70, (–40 to 158)
Relative humidity (IEC 60068-2-30), non-condensing		%	5 to 95
Air pressure (operation)		hPa	795 to 1080
Ambient mechanical conditions			
Pollution degree			
Power supply unit/CPU; inputs/outputs			2
Display/operating unit			3
Degree of protection (EN 50178, IEC 60529, VBG4)			
Power supply unit/CPU; inputs/outputs			IP20
Display/operating unit			IP65, Type 3R, Type 12R
Display/operating unit with protective cover			IP65, Type 3R, Type 12R
Display/operating unit with protective membrane			IP65, NEMA Type 4X (Type 3R rain-tight and Type 12 dust-tight)
Oscillations (IEC 60068-2-6)			
Constant amplitude 0.15 mm		Hz	10 to 57
Constant acceleration 2 g		Hz	57 to 150
Shocks (IEC 60068-2-27) semi-sinusoidal 15 g/11 ms		Shocks	18
Drop (IEC 60068-2-31)	Drop height	mm	50
Free fall, when packed (IEC 60068-2-32)		m	1

Electromagnetic compatibility (EMC)		
Electrostatic discharge (ESD), (IEC/EN 61000-4-2, severity level 3)		
Air discharge	kV	8
Contact discharge	kV	6
Electromagnetic fields (RFI), (IEC/EN 61000-4-3)	V/m	10
Radio interference suppression (EN 55011, IEC 61000-6-1, 2, 3, 4), limit class		B
Burst (IEC/EN 61000-4-4, severity level 3)		
Power cables	kV	2
Signal cables	kV	2
High energy pulses (Surge) MFD (IEC/EN 61000-4-5, severity level 2), power cable symmetrical	kV	0.5
Line-conducted interference (IEC/EN 61000-4-6)	V	10
Dielectric strength		
Measurement of the air clearance and creepage distance		EN 50178, UL 508, CSA C22.2, No 142
Dielectric strength		EN 50178
Overvoltage category/degree of pollution		II/2
Tools and cable cross-sections		
Solid, minimum to maximum	mm^2	0.2 to 4
	AWG	24 to 12
Flexible with ferrule, minimum to maximum	mm^2	0.2 to 2.5
	AWG	24 to 12
Slot-head screwdriver, width	mm	3.5 × 0.5
	inch	0.14 × 0.02

Display/operating unit

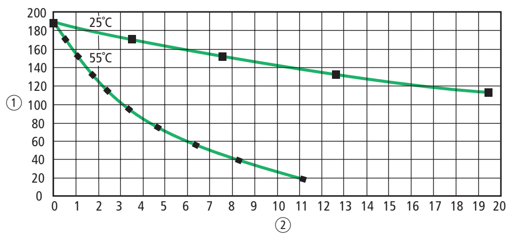
		MFD-80, MFD80-B
Power supply		
Power supply using power supply unit/CPU MFD-CP8..		
LCD display		
Type		Graphic/monochrome
Visible area W x H	mm	62 x 33
Size of pixels	mm	0.4 x 0.4
Number of pixels (W x H)		132 x 64
Spacing (pixel centre to pixel centre)	mm	0.42
LCD backlight		Yes
Backlight colour		Yellow/green
The backlight can be used and programmed in visualization applications		Yes
LEDs		
The backlight can be used and programmed in visualization applications		2
Operating buttons		
Number		9
Can be used and programmed in visualization applications		9
Mechanical lifespan de	Actuations	typ. 1×10^6
Pushbutton illumination (LED)		
Number		5
Colour		Green

Power supply

		MFD-CP8...
Rated voltage		
Rated value	V DC, (%)	24, (+20, –15)
Permissible range	V DC	20.4 to 28.8
Residual ripple	%	≦ 5
Input current		
For 24 V DC, MFD-CP8.., normally	mA	125
For 24 V DC, MFD-CP8.., MFD-80.., normally	mA	250
For 24 V DC, MFD-CP8.. ,MFD-80.., MFD-R.., MFD-T.., normally	mA	270
Voltage dips, IEC/EN 61131-2	ms	10
Heat dissipation		
For 24 V DC, MFD-CP8.., normally	W	3
For 24 V DC, MFD-CP8.., MFD-80.., normally	W	6
For 24 V DC, MFD-CP8.., MFD-80.., MFD-R.., MFD-T.., normally	W	6.5

CPU, real-time clock/timing relay/memory

Backup/accuracy of real-time clock (see graph)



Accuracy of the real-time clock		
Per day	s/day	± 5
Per year	h/year	± 0.5

Repetition accuracy of timing relays		
Accuracy of timing relay (from value)	%	± 0.02
Resolution		
Range "s"	ms	5
Range "M:S"	s	1
Range "H:M"	min	1
Rung		256
Contacts in series		4
Coil per rung		1
Program memory for program/circuit diagram	kByte	8
Program memory for display objects (visualization)	kByte	24
RAM working memory	kByte	8
Storage of programs (retentive)		FRAM
Retentive memory (retentive data, non-volatile)		FRAM
Size	Byte	200
Operating hours counter	Byte	16
Write-read cycles FRAM (minimum)		10^{10}

Inputs

		MFD-R.., MFD-T..
Digital inputs		
Number		12
Inputs usable as analog inputs, (I7, I8, I11, I12)		4
Status display		LCD status display, if available
Electrical isolation		
To supply voltage		No
Between each other		No
To the outputs		Yes
To PC interface, memory card, easy-NET network, EASY-LINK		Yes
Rated voltage		
Rated value	V DC	24
0 signal		
I1 to I6 and I9 to I10	V DC	< 5
I7, I8, I11, I12	V DC	< 8
On 1 signal		
I1 to I6 and I9 to I10	V DC	> 15
I7, I8, I11, I12	V DC	> 8
Input current on 1 signal		
I1 to I6, I9 to I10 at 24 V DC	mA	3.3
I7, I8, I11, I12 at 24 V DC	mA	2.2
Delay time for 0 to 1		
Debounce ON	ms	20
Debounce off, typical		
I1 to I4	ms	0.025
I5, I6, I9, I10	ms	0.25
I7, I8, I11, I12	ms	0.15

		MFD-R.., MFD-T..
Delay time for “1” to “0”		
Debounce ON	ms	20
Debounce OFF, typical		
I1 to I4	ms	0.025
I5, I6, I9, I10	ms	0.25
I7, I8, I11, I12	ms	0.15
Cable length (unshielded)	m	100
High-speed counter inputs, I1 to I4		
Number		4
Cable length (shielded)	m	20
High-speed up and down counters		
Counting frequency	kHz	< 3
Pulse shape		Square wave
Mark-to-space ratio		1:1
Frequency counters		
Counting frequency	kHz	< 3
Pulse shape		Square wave
Mark-to-space ratio		1:1
Incremental encoder counters		
Counting frequency	kHz	< 3
Pulse shape		Square wave
Counter inputs I1 and I2, I3 and I4		2
Signal offset		90°
Mark to space ratio		1:1

		MFD-R.., MFD-T..
Analog inputs		
Number		4
Electrical isolation		
To supply voltage		No
To the digital inputs		No
To the outputs		Yes
To the easy-NET network		Yes
Input type		DC voltage
Signal range	V DC	0 to 10
Resolution analog	V	0.01
Resolution digital	Bit	10
	Value	0 to 1023
Input impedance	kΩ	11.2
Accuracy		
Two MFD devices, from actual value	%	± 3
Within a unit, from actual value, (I7, I8, I11, I12)	%	± 2
Conversion time, analog/digital		
Debounce ON:	ms	20
Debounce OFF:		Each cycle time
Input current	mA	< 1
Cable length (shielded)	m	30

Relay outputs

		MFD-R..
Number		4
Type of outputs		Relays
In groups of		1
Connection of outputs in parallel to increase the output		Not permissible
Protection for an output relay		
Miniature circuit-breaker B16	A	16
or fuse (slow-blow)	A	8
Potential isolation to mains power supply, input, PC interface, memory card, NET network, EASY-LINK		Yes
Safe isolation	V AC	300
Basic insulation	V AC	600
Mechanical lifespan	Switch operations	10×10^6
Contacts relays		
Conventional thermal current, (UL)	A	8, (10)
Recommended for load at 12 V AC/DC	mA	> 500
Protected against short-circuit cos $\varphi = 1$ 16 A characteristic B (B16) at	A	600
Protected against short-circuit cos $\varphi = 0.5$ to 0.7 16 A characteristic B (B16) at	A	900
Rated impulse withstand voltage U_{imp} contact coil	kV	6
Rated insulation voltage U_i		
Rated operational voltage U_e	V AC	250
Safe isolation to EN 50178 between coil and contact	V AC	300
Safe isolation to EN 50178 between two contacts	V AC	300

		MFD-R..
Making capacity, IEC 60947		
AC-15 250 V AC, 3 A (600 Ops/h)	Switch operations	300000
DC-13 L/R ≦ 150 ms 24 V DC, 1 A (500 Ops/h)	Switch operations	200000
Breaking capacity, IEC 60947		
AC-15 250 V AC, 3 A (600 Ops/h)	Switch operations	300000
DC-13 L/R ≦ 150 ms 24 V DC, 1 A (500 Ops/h)	Switch operations	200000
Filament lamp load		
1000 W at 230/240 V AC	Switch operations	25000
500 W at 115/120 V AC	Switch operations	25000
Fluorescent tube with ballast, 10 × 58 W at 230/240 V AC	Switch operations	25000
Conventional fluorescent tube, compensated, 1 × 58 W at 230/240 V AC	Switch operations	25000
Conventional fluorescent tube, uncompensated, 10 × 58 W at 230/240 V AC	Switch operations	25000
Operating frequency, relays		
Mechanical switch operations	Switch operations	10 mill. (10^7)
Mechanical switching frequency	Hz	10
Resistive lamp load	Hz	2
Inductive load	Hz	0.5

UL/CSA

Uninterrupted current at 240 V AC/24 V DC		A	10/8
AC	Control Circuit Rating Codes (utilisation category)		B300 Light Pilot Duty
	Max. rated operational voltage	V AC	300
	Max. thermal uninterrupted current cos φ = 1 at B300	A	5
	Maximum make/break capacity cos φ k 1 (Make/break) at B300	VA	3600/360
DC	Control Circuit Rating Codes (utilisation category)		R300 Light Pilot Duty
	Max. rated operational voltage	V DC	300
	Max. thermal uninterrupted current at R300	A	1
	Maximum make/break capacity at R300	VA	28/28

Transistor outputs

		MFD-T..
Number of outputs		4
Contacts		Semiconductors
Rated voltage U_e	V DC	24
Permissible range	V DC	20.4 to 28.8
Residual ripple	%	≦ 5
Supply current		
On 0 state, typical/maximum	mA	18/32
On 1 state, typical/maximum	mA	24/44
Reverse polarity protection **Attention!** If voltage is applied to the outputs when the polarity of the power supply is reversed, this will result in a short circuit.		Yes
Potential isolation to the inputs, supply voltage, PC interface, memory card, NET network, EASY-Link		Yes
Rated current I_e at state 1, maximum	A	0.5

		MFD-T..
Lamp load		
Q1 to Q4 without R_V	W	5
Residual current at state “0” per channel	mA	< 0.1
Max. output voltage		
On 0 state with external load, 10 MΩ	V	2.5
On 1 state, $I_e = 0.5$ A		$U = U_e - 1$ V
Short-circuit protection Thermal (Q1 to Q4) (evaluation with diagnostics inputs I16, I15)		Yes
Short-circuit tripping current for $R_a \leqq 10$ mΩ (depending on number of active channels and their load)	A	$0.7 \leqq I_e \leqq 2$
Maximum total short-circuit current	A	8
Peak short-circuit current	A	16
Thermal cutout		Yes
Maximum switching frequency with constant resistive load $R_L = 100$ kΩ (depends on program and load)	Switch operations/h	40000
Parallel connection of outputs with resistive load; inductive load with external suppression circuit (→ Section “Connecting transistor outputs”, Page 69); combination within a group		Yes
Group 1: Q1 to Q4		
Maximum number of outputs		4
Total maximum current **Attention!** Outputs must be actuated simultaneously and for the same time duration.	A	2
Status display of the outputs		LCD Status display (if provided)

Inductive load **without external suppressor circuit**

General explanations:
$T_{0.95}$ = time in milliseconds until 95 % of the stationary current is reached

$$T_{0.95} \approx 3 \times T_{0.65} = 3 \times \frac{L}{R}$$

Utilisation category in groups for

- Q1 to Q4, Q5 to Q8

$T_{0.95}$ = 1 ms R = 48 Ω L = 16 mH	Utilisation factor per group g =		0.25
	Relative duty factor	%	100
	Max. switching frequency f = 0.5 Hz Max. duty factor DF = 50 %	Switch operations/h	1 500
DC13 $T_{0.95}$ = 72 ms R = 48 Ω L = 1.15 H	Utilisation factor g =		0.25
	Relative duty factor	%	100
	Max. switching frequency f = 0.5 Hz Max. duty factor DF = 50 %	Switch operations/h	1 500

Other inductive loads:

$T_{0.95}$ = 15 ms *R* = 48 Ω *L* = 0.24 H	Utilisation factor g =		0.25
	Relative duty factor	%	100
	Max. switching frequency *f* = 0.5 Hz Max. duty factor DF = 50 %	Switch operations/h	1 500
Inductive loading with external suppressor circuit for each load (→ Section "Connecting transistor outputs", Page 69)			
	Utilisation factor g =		1
	Relative duty factor	%	100
	Max. switching frequency Max. duty factor	Switch operations/h	Depends on the suppressor circuit

Analog output

		MFD-RA17, MFD-TA17
Number		1
Electrical isolation		
To power supply		No
To the digital inputs		No
To the digital outputs		Yes
To the easy-NET network		Yes
Output type:		DC voltage
Signal range	V DC	0 to 10
Output current max.	mA	10
Load resistor	k Ω	1
Short-circuit and overload proof		Yes

		MFD-RA17, MFD-TA17
Resolution analog	V DC	0.01
Resolution digital	Bit	10
	Value	0 to 1023
Transient recovery time	µs	100
Accuracy (–25 to 55°C), related to the range	%	2
Accuracy (25°C), related to the range	%	1
Conversion time		Each CPU cycle

easy-NET network

		MFD-CP8-NT
Number of stations		8
Bus length/transmission speed[1)]	m/kBaud	6/1 000 25/500 40/250 125/125 300/50 700/20 1 000/10
Electrical isolation		Yes
To power supply, inputs, outputs, EASY-LINK, PC interface, memory module		
Bus termination (→ accessories)		Yes
First and last station		
Plug connector (→ accessories)	poles	8
Type		RJ45

		MFD-CP8-NT
Cable cross-sections, with cable lengths and cable resistance/m		
Cross-section up to 1000, < 16 mΩ/m	mm^2 (AWG)	1.5 (16)
Cross-section up to 600, < 26 mΩ/m	mm^2 (AWG)	0.75 to 0.8 (18)
Cross-section up to 400, < 40 mΩ/m	mm^2 (AWG)	0.5 to 0.6 (20, 19)
Cross-section up to 250, < 60 mΩ/m	mm^2 (AWG)	0.34 to 0.5 (22, 21, 20)
Cross-section up to 175, < 70 mΩ/m	mm^2 (AWG)	0.25 to 0.34 (23, 22)
Cross-section up to 40, < 140 mΩ/m	mm^2 (AWG)	0.13 (26)

1) Bus lengths above 40 m can only be achieved with cables with reinforced cross-section and connection adapter.

List of the function blocks

Function blocks

Element	Meaning of abbreviation	Function block designation	Page
A	**a**nalog value comparator	Analog value comparator	160
AR	**ar**ithmetic	Arithmetic	163
BC	**b**lock **c**ompare	Data block compare	167
BT	**b**lock **t**ransfer	Data block transfer	174
BV	**B**oolean **o**peration	Boolean operation	185
C	**c**ounter	Counters	188
CF	**c**ounter **f**requency	Frequency counters	195
CH	**c**ounter **h**igh-speed	High-speed counters	199
CI	**c**ounter fast **i**ncremental value encoder	High-speed incremental encoder	205
CP	**c**om**p**arators	Comparators	210
D	**d**isplay	Text function block	212
DB	**d**ata **b**lock	Data function block	213
DC	DDC controller (direct **d**igital **c**ontrol)	PID controllers	215
FT	**f**il**t**er	PT1 signal smoothing filter	221
GT	**GET**	GET network	215
HW	**h**ora$_{(lat)}$ **w**eek	Seven day timer	226
HY	**h**ora$_{(lat)}$ **y**ear	Twelve month timer	231
LS	**l**inear **s**caling	Value scaling	235
MR	**m**aster **r**eset	Master reset	242
NC	**n**umeric **c**oding	Numerical converters	243
OT	**o**perating **t**ime	Operating hours counter	249
PT	**PUT**	PUT network	250
PW	**p**ulse **w**idth modulation	Pulse width modulation	216
SC	**s**ynchronize **c**locks	Synchronise clock via network	255

Element	Meaning of abbreviation	Function block designation	Page
ST	**s**et **t**ime	Set cycle time	221
T	**t**iming relays	Timing relay	259
VC	**v**alue **c**apsuling	Value limitation	272
:		Jumps	203

Function block coils

Coil	Meaning of abbreviation	Description
C_	**c**ount input	Counter input
D_	**d**irection input	Count up/down indicator
ED	**e**nable **D**ifferential component	Activate differential component
EI	**e**nable **i**ntegral component	Activate integral component
EN	**en**able	Enable module
EP	**e**nable **p**roportional component	Activate proportional component
RE	**re**set	Reset actual value to zero
SE	**s**et **e**nable	Set to a predefined value
ST	**st**op	STOP block processing
T_	**t**rigger	Trigger coil

Function block contacts

Contact	Meaning of abbreviation	Description
CY	**c**arr**y**	Status "1", if the value range is exceeded; (**c**arr**y**)
E1	**e**rror **1**	Error 1, dependent on function block
E2	**e**rror **2**	Error 2, dependent on function block
E3	**e**rror **3**	Error 3, dependent on function block
EQ	**equ**al	Comparison result, status 1 if values equal.
FB	**f**all **b**elow	Status "1", if the actual value is less than or equal to the lower setpoint value;
GT	**g**reater **t**han	Status 1 if the value at I1 > I2;
LI	limit indicator	Value range manipulated variable exceeded
LT	**l**ess **t**han	Status 1 if the value at I1 < I2;
OF	**o**ver**f**low	Status "1", if the actual value is greater than or equal to the upper setpoint value;
Q1	output (**Q1)**	Switch output
QV	**o**utput **v**alue	Current actual value of the function block (e.g. counter value);
ZE	**ze**ro	Status "1", if the value of the element input QV is equal to zero;

Function block inputs (constants, operands)

Input	Meaning of abbreviation	Description
F1	**F**actor **1**	Gain factor for I1 (I1 = F1 × Value)
F2	**F**actor **2**	Gain factor for I2 (I2 = F2 × Value)
HY	**Hy**steresis	Switching hysteresis for value I2 (Value HY applies to positive and negative hysteresis.)
I1	**I**nput **1**	1st input word
I2	**I**nput **2**	2nd input word
KP	Standard	Proportional gain
ME	**M**inimum mak**e** time	Minimum make time
MV	**m**anual **v**alue	Manual manipulated variable
NO	**n**umbers **o**f elements	Number of elements
OS	**O**ff**s**et	Offset for the value I1
PD	**P**eriod **d**uration	Period duration
SH	**S**etpoint **h**igh	Upper limit value
SL	**S**etpoint **l**ow	Lower limit value
SV	**S**et **v**alue	Defined actual value (Preset)
TC		Scan time
TG		Recovery time
TN	Standard	Rate time
TV	Standard	Reset time
X1	X1, interpolation point 1 abscissa	Lower value of source range
X2	Interpolation point 2 abscissa	Upper value of source range
Y1	Interpolation point 1 ordinate	Lower value of target range
Y2	Interpolation point 2 ordinate	Upper value of target range

Function block output (operands)

Input	Meaning of abbreviation	Description
QV	Output value	Output value

Other operands

Other operands	Description
MB	Marker byte (8-bit value)
IA	Analog input (if available on device!)
MW	Marker word (16-bit value)
QA	Analog output (if available on device!)
MD	Marker double word (32-bit value)
NU	Constant (**number**), value range from –2 147 483 648 to +2 147 483 647

Memory requirement

The following table provides an overview of the memory requirement of the easy800 rungs, function blocks and their respective constants:

	Space requirement per circuit conn./function block Byte	Space requirement per constant on the function block input Byte
Rung	20	–
Function blocks		
A	68	4
AR	40	4
BC	48	4
BT	48	4
BV	40	4
C	52	4
CF	40	4
CH	52	4
CI	52	4
CP	32	4
D	160	
DC	96	4
DB	36	4
FT	56	4
GT	28	
HW	68	4 (per channel)
HY	68	4 (per channel)
LS	64	4
MR	20	
NC	32	4

	Space requirement per circuit conn./function block Byte	Space requirement per constant on the function block input Byte
OT	36	4
PT	36	4
PW	48	4
SC	20	
ST	24	4
T	48	4
VC	40	4
:	–	–

Index

Moeller GmbH
Industrieautomation
Hein-Moeller-Straße 7–11
D-53115 Bonn

E-Mail: info@moeller.net
Internet: www.moeller.net

AWB2528-1480GB Doku/Doku/Eb 06/04
Printed in the Federal Republic of Germany (xx/xx)
Article No.: 267188

4 015082 671884

MOELLER

Think future. Switch to green.