Rückenbreite 15,4 mm

Moeller GmbH
Industrieautomation
Hein-Moeller-Straße 7–11
D-53115 Bonn
E-Mail: info@moeller.net
Internet: www.moeller.net

© 2003 by Moeller GmbH
Subject to alteration
AWB2528-1480GB IM-DIM-Elux GG/GB
Printed in the Federal Republic of Germany
Article No.: 267188

For Immediate Delivery call KMParts.com at (866) 595-9616
Rückenbreite festlegen! (1 Blatt = 0,106 mm)
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
Device designation 7
Writing conventions 7

1 MFD-Titan 9
Target readership 9
Proper use 9
– Improper use 9
Overview 10
Device overview 12
– MFD devices at a glance 12
– Type references for the MFD-Titan 14
MFD operation 14
– Buttons 14
– Moving through menus and choosing values 15
– MFD-Titan Status display 16
– Status display for local expansion 17
– MFD-Titan advanced Status display 17
– MFD-Titan LED display 17
– Selecting or toggling between menu items 25
– Cursor display 26
– Setting values 26

2 Installation 27
Mounting 27
Connecting the expansion unit 45
Terminals 45
– Tool for cage clamp terminals 45
– Connection cross-sections of the MFD cage clamp terminal cables 45
– Tool for slot-head screws for easy expansion unit 45
– Connection cross-sections of screw terminal cables 45
Connecting the power supply 46
– Cable protection 48
Contents

Connecting the inputs  49
  – Connecting easy-AC inputs  49
  – Connecting MFD-DC inputs  53
Connecting the outputs  58
Connecting relay outputs  59
  – MFD-R..  59
  – EASY6..-RE..  59
  – EASY2..-RE  60
Connecting transistor outputs  60
  – MFD-T..  60
  – EASY6..-DC-TE  61
Connecting analog outputs  63
  – Connecting servo valves  63
  – Setpoint entry for a drive  64
Connecting the NET network  64
  – Accessories  64
  – Cable length and cross-sections  66
  – Plugging and unplugging network cables  68
Connecting the serial interface  70
Expanding inputs/outputs  73
  – Local expansion  73

3  Commissioning  77
Switching on  77
Setting the menu language  77
MFD operating modes  78
Creating your first circuit diagram  79
  – Testing the circuit diagram  86
  – Fast circuit diagram entry  89
Configuring an easy-NET network  90
  – Entering the network station number  91
  – Changing the easy-NET network configuration  94
  – Displaying the Status display of other stations  94
Configuring the interface for the COM-LINK mode  96
  – Setting up the COM-LINK  97
Terminal mode  101
  – Terminal mode  101

For Immediate Delivery call KMParts.com at (866) 595-9616
4 Wiring with MFD-Titan 111

MFD-Titan operation 111
– Buttons for drawing circuit diagrams and function block usage 111
– Usable relays and function blocks (coils) 119
– Markers, analog operands 122
– Circuit diagram display 125
– Saving and loading programs 127

Working with contacts and relays 128
– Creating and modifying connections 131
– Inserting and deleting a circuit connection 133
– Aborting circuit diagram entry 134
– Searching for contacts and coils 134
– “Go to” a circuit connection 135
– Deleting the circuit connection 135
– Switching via the cursor buttons 136
– Checking the circuit diagram 137
– Function block editor 138
– Checking function blocks 142
– Coil functions 143

Function blocks 148
– Analog value comparator/threshold value switch 150
– Arithmetic function block 153
– Data block transfer 164
– Boolean sequence 175
– Counters 178
– Frequency counters 185
– High-speed counters 189
– Comparators 200
– Text output function block 202
– Data function block 203
– PID controller 205
– Signal smoothing filter 211
– GET, fetch a value from the network 214
– Seven-day time switch 216
– Year time switch 221
– Value scaling 225
– Jumps 229
– Master reset 232
## Contents

- Numerical converters 233
- Operating hours counter 239
- PUT, send a value onto the network 240
- Pulse width modulation 242
- Setting date/time 245
- Set cycle time 247
- Timing relay 249
- Value limitation 262
- Example with timing relay and counter function block 264

### 5 Visualization with MFD-Titan 269

### 6 easy-NET Network, COM-LINK Serial Connection 271

Introduction to easy-NET 271
easy-NET network topologies, addressing and functions 272
  - Loop through the unit wiring method 272
  - T connector and spur line 272
  - Position and addressing of the operands via easy-NET 274
  - Possible write and read authorisation in the network 276
Configuration of the easy-NET network 277
  - Station number 277
  - Transmission speed 277
  - Pause time, changing the write repetition rate manually 278
  - Automatic change of the RUN and STOP mode 279
  - Input/output device (REMOTE IO) configuration 280
  - Station message types 281
  - Transfer behaviour 281
  - Signs of life of the individual stations and diagnostics 282
Introduction to COM-LINK 286
  - Topology 286
  - Configuration of the COM-LINK 290
7 MFD Settings 295

Password protection 295
– Password setup 296
– Selecting the scope of the password 297
– Activating the password 298
– Unlocking MFD 299
Changing the menu language 302
Changing parameters 303
– Adjustable parameters for function blocks 304
Setting date, time and daylight saving time 305
Changing between winter/summer time (DST) 306
Activating input delay (debounce) 308
– Deactivating debounce (input delay) 308
Activating and deactivating the P buttons 309
– Activating the P buttons 309
– Deactivating the P buttons 310
Startup behaviour 310
– Setting the startup behaviour 310
– Behaviour when the circuit diagram is deleted 311
– Behaviour during upload/download to card or PC 311
– Possible faults 312
– Card startup behaviour 312
– Terminal mode 313
Setting LCD contrast and backlight 314
Retention 316
– Requirements 317
– Setting retentive behaviour 317
– Deleting ranges 318
– Deleting retentive actual values of markers and function blocks 318
– Transferring retentive behaviour 319
Displaying device information 320
8 Inside MFD 323
MFD Program cycle 323
– How MFD evaluates the high-speed counters CF, CH and CI 327
– Memory management of the MFD-Titan 327
Delay times for inputs and outputs 328
Monitoring of short-circuit/overload with EASY..-D.-T.. 331
Expanding MFD-Titan 333
– How is an expansion unit recognised? 333
– Function monitoring of expansion units 334
QA analog output 336
Loading and saving programs 336
– MFD without display and keypad 336
– Interface 336
– Memory card 337
– EASY-SOFT-PRO 342
Device version 344

Appendix 345
Technical data 345
– General 345
– CPU, real-time clock/timing relay/memory 353
– Transistor outputs 360
– Analog output 363
– easy-NET network 364
List of the function blocks 366
– Function blocks 366
– Function block coils 367
– Function block output (operands) 370
– Other operands 370
Memory requirement 371

Index 373
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

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.
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 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,
  – 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](image)

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

- **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

Figure 2: Device overview of power supply and CPU

Inputs/outputs

- **1** Inputs
- **2** Analog output (optional)
- **3** Outputs

Figure 3: Device overview of inputs/outputs
**Type references for the MFD-Titan**

**MFD - xxx x - xx x - xx**

- **Additional functions**
- **Equipment**
  - B = Operator buttons
  - ME = Open-loop control, closed-loop control; NT = Open-loop control, closed-loop control, easy-NET
- **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
- **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.

For Immediate Delivery call KMParts.com at (866) 595-9616
Moving through menus and choosing values

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 P3,
Input P2
Input P4

Reset the MFD display
Selecting main and system menu

Status display

Current selection flashes in the MFD menu

1st menu level
Main menu

Date display

1st menu level
System menu

MFD-Titan Status display

Inputs — Display operating mode, Terminal mode
Weekday/Time — or Weekday/Date
Outputs — RUN/STOP/BUSY mode

On: 1, 2, 3, 4/Off:…
MFD operation

Status display for local expansion

Inputs
Expansion
Weekday/Time
Outputs

| R 1 ........ 12 |
| RS AC P- |
| MO 10:42 |
| $ 1 8 STOP |

AC expansion ok/P buttons or Weekday/Date

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

MFD-Titan advanced Status display

Retention/Debounce/ easy-NET station

| 1 12...6.89..12 |
| RE I NT1 AC P- |
| MO 14:42 T |
| Q 12345678 RUN |

AC expansion ok/P buttons
Startup behaviour

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 12).

For Immediate Delivery call KMParts.com at (866) 595-9616
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>
<thead>
<tr>
<th>LED State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LED OFF</td>
<td>No power supply</td>
</tr>
<tr>
<td>LED continuously lit</td>
<td>Power supply present, STOP mode</td>
</tr>
<tr>
<td>LED flashing</td>
<td>Power supply present, RUN mode</td>
</tr>
</tbody>
</table>

**Red LED**

<table>
<thead>
<tr>
<th>LED State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LED OFF</td>
<td>Operation correct</td>
</tr>
<tr>
<td>LED continuously lit</td>
<td>easy-NET remote device faulty</td>
</tr>
</tbody>
</table>
Menu structure

Main menu without password protection

- You access the main menu by pressing **OK**.
Main menu with password protection

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

MFD-Titan system menu

For Immediate Delivery call KMParts.com at (866) 595-9616
System menu

The other menus of NET and COM are only shown in STOP mode.

Only one selection is possible.

For Immediate Delivery call KMParts.com at (866) 595-9616
System menu

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

NET-ID : __
BAUDRATE: ____KB
BUSDELAY: __
SEND IO: __
REMOTE RUN
REMOTE IO

This list is only created in Station 1.
This list only appears if station 1 has been selected.

Fault scenario with ID conflict.

Fault scenario with network fault.

ERR: ID CONFLICT
OVERWRITE CONFIGURATION ?

ERR: TIME OUT

For Immediate Delivery call KMParts.com at (866) 595-9616
System menu

This menu only appears if the COM LINK was selected.

Selecting or toggling between menu items

Cursor ↑↓

Select or toggle
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 DIN EN 50022 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 45).

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:T Fitting the protective membrane

① Protective membrane
② display/operating unit module

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/operating unit has 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.
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 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.

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 14: Top-hat rail with cutout](image)

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.
Attach the expansion unit before 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.

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.

For Immediate Delivery call KMParts.com at (866) 595-9616
Mounting the inputs/outputs onto the power supply/CPU module

For Immediate Delivery call KMParts.com at (866) 595-9616
The inputs/outputs can be mounted before or after mounting the power supply/CPU module onto the fixing shaft.

Removing the inputs/outputs

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

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

► Insert the screwdriver into the lug of the fixing shaft catch.
Mounting

- 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.

- Remove the spring with a screwdriver.
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**
The power supply/CPU module can be screwed onto a mounting plate without the display/operating unit.

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

Figure 16: Screw mounting for the MFD

For Immediate Delivery call KMParts.com at (866) 595-9616
Figure 17: Screw mounting for the EASY2...

Figure 18: Screw mounting for the easy600
Connecting the expansion unit

Figure 19: Connecting expansion units

<table>
<thead>
<tr>
<th>Terminals</th>
<th>Tool for cage clamp terminals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slot-head screwdriver, width 3.5 mm x 0.6 mm.</td>
<td></td>
</tr>
</tbody>
</table>

**Connection cross-sections of the MFD cage clamp terminal cables**

- Solid: 0.2 to 4 mm² (AWG 24 - 12)
- Flexible with ferrule: 0.2 to 2.5 mm² (AWG 24-12)

**Tool for slot-head screws for easy expansion unit**

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

**Connection cross-sections of screw terminal cables**

- Solid: 0.2 to 4 mm² (AWG 22 - 12)
- Flexible with ferrule: 0.2 to 2.5 mm² (AWG 22 -12)
Connecting the power supply

The required connection data for both device types MFD-DC, easy-DC with 24 V DC and easy-AC with standard voltages of 100 V to 240 V AC are given in the Chapter “Appendix”, Page 345.

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.

EASY…-AC-.E expansion units

Figure 20: 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.
Connecting the power supply

DC power supply for 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

![Diagram of power supply on DC expansion units]

Figure 22: 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 23: Connecting the inputs

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

Connecting easy-AC inputs

Caution!

For easy-AC, connect the inputs to the same line as the power feed in accordance with the VDE, IEC, UL and CSA safety regulations. Otherwise easy will not detect the switching level and may be damaged or destroyed by overvoltage.
Figure 24: EASY…-AC-.E expansion unit

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

Input signal voltage range
- OFF signal: 0 V to 40 V
- ON signal: 79 V to 264 V

Input current
- 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:
- 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.
Connecting 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:

![Diagram of increasing the input current](image)

Figure 26: 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 27: Limitation of the inrush current with a resistor](image)

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

![Figure 28: easy600 with EASY256-HCI](image)

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

The digital inputs must have the same voltage as the power supply of the MFD.
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 1023.

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.

![Setpoint potentiometer](image)

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 31: Setpoint potentiometer, setpoint potentiometer with upstream resistor
Use a potentiometer with a resistance of \( \leq 1 \, \Omega \), e.g. 1 k\( \Omega \), 0.25 W.

**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 \( \Omega \) resistor.

The following values apply:

- 4 mA = 0.2 V
- 10 mA = 4.8 V
- 20 mA = 9.5 V

(according to \( U = R \times I = 478 \, \Omega \times 10 \, mA \sim 4.8 \, V \))

---

For Immediate Delivery call KMParts.com at (866) 595-9616
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 33: 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.
Connecting the outputs

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

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 “Appendix”, from Page 345).
Connecting relay outputs

**MFD-R..**

![Diagram of MFD-R.. relay outputs]

Figure 36: MFD-R.. relay outputs

**EASY6..-..RE..**

![Diagram of EASY6..-..RE.. relay outputs]

Figure 37: EASY6..-..RE.. relay outputs
EASY2...-RE

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 39: MFD-T.. transistor outputs
Connecting transistor outputs

**EASY6..-DC-TE**

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).

![Diagram of Inductivity with Suppressor Circuit]

Figure 41: 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 331).
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 42: Connecting servo valves
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 271.

Accessories

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

Connection assignment of the RJ45 socket on the device

Figure 43: Setpoint entry for a drive

Figure 44: RJ45 socket
**Connection cable:**
4-pair twisted cable; → Section “Technical data”, Page 345

<table>
<thead>
<tr>
<th>A</th>
<th>1</th>
<th>ECAN_H</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2</td>
<td>ECAN_L</td>
</tr>
<tr>
<td>B</td>
<td>3</td>
<td>GND (Ground)</td>
</tr>
<tr>
<td>B</td>
<td>4</td>
<td>SEL_IN</td>
</tr>
</tbody>
</table>

Figure 45: 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_OUT cable is only used for automatic addressing.

Table 5: Prefabricated cables, RJ45 plug on both ends

<table>
<thead>
<tr>
<th>Cable length cm</th>
<th>Type designation</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>EASY-NT-30</td>
</tr>
<tr>
<td>80</td>
<td>EASY-NT-80</td>
</tr>
<tr>
<td>150</td>
<td>EASY-NT-150</td>
</tr>
</tbody>
</table>

Material for self-manufactured cables
100 m 4 × 0.18 mm²: 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.

<table>
<thead>
<tr>
<th>Cable length (m)</th>
<th>Cable resistance (mΩ/m)</th>
<th>Cross-section (mm²)</th>
<th>AWG</th>
</tr>
</thead>
<tbody>
<tr>
<td>up to 40</td>
<td>≤ 140</td>
<td>0.13</td>
<td>26</td>
</tr>
<tr>
<td>up to 175</td>
<td>≤ 70</td>
<td>0.25 to 0.34</td>
<td>23, 22</td>
</tr>
<tr>
<td>up to 250</td>
<td>≤ 60</td>
<td>0.34 to 0.5</td>
<td>22, 21, 20</td>
</tr>
<tr>
<td>up to 400</td>
<td>≤ 40</td>
<td>0.5 to 0.6</td>
<td>20, 19</td>
</tr>
<tr>
<td>up to 600</td>
<td>≤ 26</td>
<td>0.75 to 0.8</td>
<td>18</td>
</tr>
<tr>
<td>up to 1000</td>
<td>≤ 16</td>
<td>1.5</td>
<td>16</td>
</tr>
</tbody>
</table>

The surge impedance of the cables used must be 120 Ω.
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 \( \Omega/m \)), the entire cable resistance \( R_L \) must not exceed the following values. \( R_L \) depends on the selected baud rates:

<table>
<thead>
<tr>
<th>Baud rate kBaud</th>
<th>Cable resistance ( R_L ) ( \Omega )</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 to 125</td>
<td>( \leq 30 )</td>
</tr>
<tr>
<td>250</td>
<td>( \leq 25 )</td>
</tr>
<tr>
<td>500</td>
<td>( \leq 25 )</td>
</tr>
<tr>
<td>1000</td>
<td>( \leq 12 )</td>
</tr>
</tbody>
</table>

\( l_{\text{max}} = \) maximum cable length in m

\( R_L = \) Total cable resistance in \( \Omega \)

\( R' = \) Cable resistance per unit length in \( \Omega/m \)

\( l_{\text{max}} = \frac{R_L}{R'} \)

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

\( S_{\text{min}} = \) minimum cable cross-section in mm²

\( \rho_{\text{cu}} = \) resistivity of copper, if not otherwise stated 0.018 \( \Omega \text{mm}^2/\text{m} \)

\( S_{\text{min}} = \frac{l \times \rho_{\text{cu}}}{12.4} \)

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

\[ l_{\text{max}} = \text{cable length in m} \]
\[ S = \text{minimum cable cross-section in mm}^2 \]
\[ \rho_{\text{cu}} = \text{resistivity of copper, if not otherwise stated } 0.018 \ \Omega \text{mm}^2/\text{m} \]

\[ l_{\text{max}} = \frac{S \times 12.4}{\rho_{\text{cu}}} \]

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.
Connecting the NET network

Figure 46: 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.
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 to ensure compliance with EMC requirements.
Remove the interface cover or other plugs from the interface.

Fit the connectors in the devices.
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.
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**

| MFD-CP8-.. | EASY618-..-RE | • 12 AC inputs,  
|           |               | • 6 relay outputs  
| EASY620-..-TE | • 12 DC inputs,  
| EASY202-RE | • 8 transistor outputs  

| 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.

![EASY-LINK-DS diagram]

**Figure 49: 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.

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−.
For Immediate Delivery call KMParts.com at (866) 595-9616
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

For Immediate Delivery call KMParts.com at (866) 595-9616
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 date, if you wish, see Section “Changing the menu language”, Page 302.

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 51: Lamp controller with relays
In the following example, MFD carries out all the wiring and performs the tasks of the circuit diagram shown below.

![Diagram of Lamp Controller with MFD](image)

**Figure 52: Lamp controller with MFD**
Creating your first circuit diagram

Starting point Status display

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 mask 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.

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.

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: = Circuit connection (line), C: = Contact or coil (contact), B: = Free memory available in bytes. Start value 7944, with the first three circuit connections 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 circuit connection. MFD automatically connects the contact to the power supply.

Figure 53: 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 Ω 01. The symbol Ω identifies the coil’s function, in this case a relay coil acting as a contactor. Ω 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 $\times$, 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 circuit connection.
- 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 circuit connection up to the coil.

Press **ALT** to wire the cursor from 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 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 Q01 are correct and do not have to be changed.

Your first working MFD circuit diagram now looks like this:

![Diagram](image)

Press ESC to leave the circuit diagram display.

The SAVE menu appears.

![Diagram](image)

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

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.

Change to the Status display and press pushbutton actuator S1.

The contacts (boxes) 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 circuit connections 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.

![Power flow display](image)

Figure 56: 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 circuit connection is interrupted and relay Q1 drops out.

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 circuit connections

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

Figure 57: Power flow display: Input I1 is closed, input I2 is open, relay Q1 has picked up

Figure 58: 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 circuit connection is interrupted and relay Q1 drops out.

Use the cursor buttons ñ í Œ ì í to move between the contacts or coil.

Press the cursor button Œ.

The cursor moves to the second contact.

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

Figure 59: 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

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 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 circuit connection 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.

![Example topology with two easy-NET stations](image)

Figure 60: Example topology with two easy-NET stations

1. Network termination resistor

   ![Physical location](image)

   ![Station number](image)

► 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 are supplied with voltage.
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 displays 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 \( \wedge \) and \( \vee \). 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.

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

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

Press the OK button.

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.

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 \(^{\wedge}\) and \(^{\vee}\) 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 271.

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 the number of the required station with the ñ 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 (see also 286).

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 61: 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.
Configuring the interface for the COM-LINK mode

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 with the Status display active.

The System menu appears

Select the CONFIGURATOR menu option.

Press the OK button.

Select the COM... menu option.

Press the button.

Press the OK button.
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.

- Press the **OK** button.
- Select 19200 baud as the baud rate.
- Press the **▼** or **▲** button.
- Confirm with 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.

- 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.
Configuring the interface for the COM-LINK mode

Select the REMOTE MARKER menu item.

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

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

For Immediate Delivery call KMParts.com at (866) 595-9616
Press the OK button.

Use the ▲ button to select the start of the READ marker range.

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.

Use the ñ button to enter the WRITE range.

Enter the WRITE range.

Press ESC to leave the entry menu.

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

Press ESC to return to the Status display.

The entry in the second COM line indicates that the COM connection is active.
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 62: Terminal mode using the point-to-point serial interface
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 64: 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 65: 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.
The first menu will appear.

➤ Press the button.

This will display the TERMINAL MODE menu item.

➤ Press the OK button.

The START MODE menu item will flash.

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

Station ID:
0 = Station at 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

- Press the OK button.

Select your station.
- Press the \( \swarrow \) or \( \searrow \) button.
- Press the OK button.

Select the START MODE menu option.
- Press the \( \swarrow \) button.
- Press the OK button.

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

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

This mask is not displayed in Terminal mode.

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

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

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.
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.

Press the ESC button twice.
The Status display of the MFD device is active.
The flashing star at the top right of the display is no longer present.
4  Wiring with MFD-Titan

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

<table>
<thead>
<tr>
<th>MFD-Titan operation</th>
<th>Buttons for drawing circuit diagrams and function block usage</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Delete circuit connection, contact, relay or empty line in the circuit diagram</td>
</tr>
<tr>
<td></td>
<td>Toggle between break and make contact</td>
</tr>
<tr>
<td></td>
<td>Connect contacts and relays</td>
</tr>
<tr>
<td></td>
<td>Add circuit connections</td>
</tr>
<tr>
<td>^ ^</td>
<td>Change values</td>
</tr>
<tr>
<td></td>
<td>Cursor up, down</td>
</tr>
<tr>
<td>&lt; &gt;</td>
<td>Change place</td>
</tr>
<tr>
<td></td>
<td>Cursor left, right</td>
</tr>
<tr>
<td></td>
<td>Cursor buttons set as P buttons:</td>
</tr>
<tr>
<td>&lt;</td>
<td>Input P1, ^ Input P2</td>
</tr>
<tr>
<td>&gt;</td>
<td>Input P3, \ Input P4</td>
</tr>
<tr>
<td></td>
<td>Undo setting from previous OK</td>
</tr>
<tr>
<td></td>
<td>Exit current display or menu</td>
</tr>
<tr>
<td></td>
<td>Change, add contact/relay</td>
</tr>
<tr>
<td></td>
<td>Save setting</td>
</tr>
<tr>
<td></td>
<td>Terminal mode on/off</td>
</tr>
</tbody>
</table>
**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 circuit connection, 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 masks.

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

Visualisation masks
Visualisation masks 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: a Usable contacts

<table>
<thead>
<tr>
<th>Contact</th>
<th>MFD-Titan display</th>
</tr>
</thead>
<tbody>
<tr>
<td>Make contact, open in release position</td>
<td>I, Q, M, A, .... Other contacts → Table</td>
</tr>
<tr>
<td>Break contact, closed in release position</td>
<td>I, Q, M, A, ... Other contacts → Table</td>
</tr>
</tbody>
</table>

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

Table 7: a Contacts

<table>
<thead>
<tr>
<th>Contact</th>
<th>Make contact</th>
<th>Break contact</th>
<th>Number</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inputs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inputs of a network station</td>
<td>*I</td>
<td>*I</td>
<td>01...12</td>
<td>274</td>
</tr>
<tr>
<td>* = Station address 1 to 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COM slave inputs</td>
<td>1I</td>
<td>1I</td>
<td>01...12</td>
<td></td>
</tr>
<tr>
<td>MFD-Titan input terminal</td>
<td>I</td>
<td>I</td>
<td>01...12</td>
<td></td>
</tr>
<tr>
<td>Cursor button</td>
<td>P</td>
<td>P</td>
<td>01...04</td>
<td></td>
</tr>
<tr>
<td>Network station expansion input terminal</td>
<td>*R</td>
<td>*R</td>
<td>01...12</td>
<td>274</td>
</tr>
<tr>
<td>* = Station address 1 to 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Input terminal for expansion unit COM slave</td>
<td>1R</td>
<td>1R</td>
<td>01...12</td>
<td></td>
</tr>
<tr>
<td>Input terminal for expansion unit</td>
<td>R</td>
<td>R</td>
<td>01...12</td>
<td></td>
</tr>
<tr>
<td>Bit inputs via the network</td>
<td>*RN</td>
<td>*RN</td>
<td>01...32</td>
<td>274</td>
</tr>
<tr>
<td>* = Station address 1 to 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Diagnostics inputs

<table>
<thead>
<tr>
<th>Contact</th>
<th>Make contact</th>
<th>Break contact</th>
<th>Number</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expansion network station status</td>
<td>*I</td>
<td>*I</td>
<td>14</td>
<td>333</td>
</tr>
<tr>
<td>Network station short-circuit/overload</td>
<td>*I</td>
<td>*I</td>
<td>15...16</td>
<td>331</td>
</tr>
<tr>
<td>COM slave expansion unit status</td>
<td>1I</td>
<td>1I</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>COM slave short-circuit/overload</td>
<td>1I</td>
<td>1I</td>
<td>15...16</td>
<td></td>
</tr>
<tr>
<td>Expansion status</td>
<td>I</td>
<td>I</td>
<td>14</td>
<td>333</td>
</tr>
<tr>
<td>Short-circuit/overload</td>
<td>I</td>
<td>I</td>
<td>15...16</td>
<td>331</td>
</tr>
<tr>
<td>Short-circuit/overload in expansion network station</td>
<td>*R</td>
<td>*R</td>
<td>15...16</td>
<td>331</td>
</tr>
<tr>
<td>Short-circuit/overload in COM slave expansion</td>
<td>1R</td>
<td>1R</td>
<td>15...16</td>
<td></td>
</tr>
<tr>
<td>Short circuit/overload with expansion</td>
<td>R</td>
<td>R</td>
<td>15...16</td>
<td>331</td>
</tr>
</tbody>
</table>

## Outputs

<table>
<thead>
<tr>
<th>Contact</th>
<th>Make contact</th>
<th>Break contact</th>
<th>Number</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deactivate backlight of the MFD display</td>
<td>LE</td>
<td>LE</td>
<td>01</td>
<td></td>
</tr>
<tr>
<td>Red LED of MFD display</td>
<td>LE</td>
<td>LE</td>
<td>02</td>
<td></td>
</tr>
<tr>
<td>Green LED of MFD display</td>
<td>LE</td>
<td>LE</td>
<td>03</td>
<td></td>
</tr>
<tr>
<td>MFD-Titan output MFD network station</td>
<td>*Q</td>
<td>*Q</td>
<td>01...08</td>
<td>274</td>
</tr>
<tr>
<td>COM slave output</td>
<td>1Q</td>
<td>1Q</td>
<td>01...08</td>
<td></td>
</tr>
<tr>
<td>MFD-Titan output</td>
<td>0</td>
<td>0</td>
<td>01...08</td>
<td>-</td>
</tr>
<tr>
<td>MFD-Titan output expansion with network station</td>
<td>*S</td>
<td>*S</td>
<td>01...08</td>
<td>274</td>
</tr>
<tr>
<td>Output of COM slave expansion</td>
<td>1S</td>
<td>1S</td>
<td>01...08</td>
<td></td>
</tr>
<tr>
<td>MFD-Titan output expansion</td>
<td>S</td>
<td>S</td>
<td>01...08</td>
<td>-</td>
</tr>
<tr>
<td>Bit outputs via the network</td>
<td>*SN</td>
<td>*SN</td>
<td>01...32</td>
<td>274</td>
</tr>
</tbody>
</table>

* = Station address 1 to 8
### Contact

<table>
<thead>
<tr>
<th>Other contacts</th>
<th>Make contact</th>
<th>Break contact</th>
<th>Number</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Markers</td>
<td>M</td>
<td>M</td>
<td>01...96</td>
<td>122</td>
</tr>
<tr>
<td>COM slave marker (REMOTE MARKER)</td>
<td>1M</td>
<td>1M</td>
<td>01...96</td>
<td>289</td>
</tr>
<tr>
<td>Jump label</td>
<td></td>
<td>01...32</td>
<td></td>
<td>229</td>
</tr>
<tr>
<td>Diagnostics messages</td>
<td>ID</td>
<td>ID</td>
<td>01...16</td>
<td>283</td>
</tr>
<tr>
<td>COM slave diagnostics messages</td>
<td>1ID</td>
<td>1ID</td>
<td>01...16</td>
<td>289</td>
</tr>
</tbody>
</table>

### Function blocks

<table>
<thead>
<tr>
<th>Function blocks</th>
<th>Make contact</th>
<th>Break contact</th>
<th>Number</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analog value comparator function block</td>
<td>A × 01</td>
<td>A × 01</td>
<td>x=01...32</td>
<td>150</td>
</tr>
<tr>
<td>Arithmetic value overflow (CARRY) function block</td>
<td>AR × CY</td>
<td>AR × CY</td>
<td>x=01...32</td>
<td>153</td>
</tr>
<tr>
<td>Zero arithmetic value (zero) function block</td>
<td>AR × ZE</td>
<td>AR × ZE</td>
<td>x=01...32</td>
<td>153</td>
</tr>
<tr>
<td>Data block comparator function block, error: number of elements exceeded</td>
<td>BC × E1</td>
<td>BC × E1</td>
<td>x=01...32</td>
<td>157</td>
</tr>
<tr>
<td>Data block comparator function block, error: range overlap</td>
<td>BC × E2</td>
<td>BC × E2</td>
<td>x=01...32</td>
<td>157</td>
</tr>
<tr>
<td>Data block comparator function block, error: invalid offset</td>
<td>BC × E3</td>
<td>BC × E3</td>
<td>x=01...32</td>
<td>157</td>
</tr>
<tr>
<td>Data block comparator function block, comparison result</td>
<td>BC × EQ</td>
<td>BC × EQ</td>
<td>x=01...32</td>
<td>164</td>
</tr>
<tr>
<td>Data block comparator function block, error: number of elements exceeded</td>
<td>BT × E1</td>
<td>BT × E1</td>
<td>x=01...32</td>
<td>164</td>
</tr>
<tr>
<td>Data block transfer function block, error: range overlap</td>
<td>BT × E2</td>
<td>BT × E2</td>
<td>x=01...32</td>
<td>164</td>
</tr>
<tr>
<td>Data block transfer function block, error: invalid offset</td>
<td>BT × E3</td>
<td>BT × E3</td>
<td>x=01...32</td>
<td>164</td>
</tr>
<tr>
<td>Boolean sequence function block, value zero</td>
<td>BV × ZE</td>
<td>BV × ZE</td>
<td>x=01...32</td>
<td>175</td>
</tr>
<tr>
<td>Contact</td>
<td>Make contact</td>
<td>Break contact</td>
<td>Number</td>
<td>Page</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>--------------</td>
<td>---------------</td>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>Counter function block, upper setpoint value exceeded (Overflow)</td>
<td>C X OF</td>
<td>C X OF</td>
<td>X=01...32</td>
<td>178</td>
</tr>
<tr>
<td>Counter function block, lower setpoint value undershot (Fall below)</td>
<td>C X FB</td>
<td>C X FB</td>
<td>X=01...32</td>
<td>178</td>
</tr>
<tr>
<td>Counter function block, actual value equal to zero</td>
<td>C X ZE</td>
<td>C X ZE</td>
<td>X=01...32</td>
<td>178</td>
</tr>
<tr>
<td>Counter function block, actual value has exceeded counter range (CARRY)</td>
<td>C X CY</td>
<td>C X CY</td>
<td>X=01...32</td>
<td>178</td>
</tr>
<tr>
<td>Frequency counter function block, upper setpoint value exceeded (Overflow)</td>
<td>CF X OF</td>
<td>CF X OF</td>
<td>X=01...04</td>
<td>185</td>
</tr>
<tr>
<td>Frequency counter function block, lower setpoint value undershot (Fall below)</td>
<td>CF X FB</td>
<td>CF X FB</td>
<td>X=01...04</td>
<td>185</td>
</tr>
<tr>
<td>Frequency counter function block, actual value equal to zero</td>
<td>CF X ZE</td>
<td>CF X ZE</td>
<td>X=01...04</td>
<td>185</td>
</tr>
<tr>
<td>High-speed counter function block, upper setpoint value exceeded (Overflow)</td>
<td>CH X OF</td>
<td>CH X OF</td>
<td>X=01...04</td>
<td>189</td>
</tr>
<tr>
<td>High-speed counter function block, lower setpoint value undershot (Fall below)</td>
<td>CH X FB</td>
<td>CH X FB</td>
<td>X=01...04</td>
<td>189</td>
</tr>
<tr>
<td>High-speed counter function block, actual value equal to zero</td>
<td>CH X ZE</td>
<td>CH X ZE</td>
<td>X=01...04</td>
<td>189</td>
</tr>
<tr>
<td>High-speed counter function block, actual value has exceeded counter range (CARRY)</td>
<td>CH X CY</td>
<td>CH X CY</td>
<td>X=01...04</td>
<td>189</td>
</tr>
<tr>
<td>Incremental encoder counter function block, upper setpoint value exceeded (Overflow)</td>
<td>CI X OF</td>
<td>CI X OF</td>
<td>X=01...02</td>
<td>195</td>
</tr>
<tr>
<td>Incremental encoder counter function block, lower setpoint value undershot (Fall below)</td>
<td>CI X FB</td>
<td>CI X FB</td>
<td>X=01...02</td>
<td>195</td>
</tr>
<tr>
<td>Incremental encoder counter function block, actual value equal to zero</td>
<td>CI X ZE</td>
<td>CI X ZE</td>
<td>X=01...02</td>
<td>195</td>
</tr>
<tr>
<td>Contact</td>
<td>Make contact</td>
<td>Break contact</td>
<td>Number</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>--------------</td>
<td>---------------</td>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>Incremental encoder counter function block, actual value has exceeded counter range (CARRY)</td>
<td>CI X CY</td>
<td>CT X CY</td>
<td>X=01...02</td>
<td>195</td>
</tr>
<tr>
<td>Comparator function block, less than</td>
<td>CP X LT</td>
<td>CF X LT</td>
<td>X=01...32</td>
<td>200</td>
</tr>
<tr>
<td>Comparator function block, equal to</td>
<td>CP X E0</td>
<td>CF X E0</td>
<td>X=01...32</td>
<td>200</td>
</tr>
<tr>
<td>Comparator function block, greater than</td>
<td>CP X GT</td>
<td>CF X GT</td>
<td>X=01...32</td>
<td>200</td>
</tr>
<tr>
<td>Text output function block</td>
<td>D X 01</td>
<td>D X 01</td>
<td>X=01...32</td>
<td>202</td>
</tr>
<tr>
<td>Data function block</td>
<td>DB X 01</td>
<td>DB X 01</td>
<td>X=01...32</td>
<td>203</td>
</tr>
<tr>
<td>PID controller, value range of manipulated variable exceeded</td>
<td>DC X LI</td>
<td>DC X LI</td>
<td>X=01...32</td>
<td>205</td>
</tr>
<tr>
<td>Receive a variable from a station (Get)</td>
<td>GT X 01</td>
<td>GT X 01</td>
<td>X=01...32</td>
<td>205</td>
</tr>
<tr>
<td>Seven-day time switch</td>
<td>HW X 01</td>
<td>HW X 01</td>
<td>X=01...32</td>
<td>216</td>
</tr>
<tr>
<td>Year time switch function block</td>
<td>HV X 01</td>
<td>HV X 01</td>
<td>X=01...32</td>
<td>221</td>
</tr>
<tr>
<td>Master reset, sets all outputs and markers to zero state</td>
<td>MR X 01</td>
<td>MR X 01</td>
<td>X=01...32</td>
<td>232</td>
</tr>
<tr>
<td>Operating hours counter function block, set time reached</td>
<td>OT X 01</td>
<td>OT X 01</td>
<td>X=01...04</td>
<td>239</td>
</tr>
<tr>
<td>Operating hours counter, value overflow (CARRY)</td>
<td>OT X CY</td>
<td>OT X CY</td>
<td>X=01...04</td>
<td>239</td>
</tr>
<tr>
<td>Send a variable to the network, enable active Put</td>
<td>PT X 01</td>
<td>PT X 01</td>
<td>X=01...32</td>
<td>240</td>
</tr>
<tr>
<td>Pulse width modulation, error minimum on or off time exceeded</td>
<td>PW X E1</td>
<td>PW X E1</td>
<td>X=01...02</td>
<td>242</td>
</tr>
<tr>
<td>Send date and time via the network (easy-NET) function block</td>
<td>SC X 01</td>
<td>SC X 01</td>
<td>X=01</td>
<td>245</td>
</tr>
<tr>
<td>Timing relay function block</td>
<td>T X 01</td>
<td>T X 01</td>
<td>X=01...32</td>
<td>249</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Relay/function block</th>
<th>MFD-Titan display</th>
<th>Number</th>
<th>Coil</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Outputs</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MFD-Titan output relays, network stations (only network master)</td>
<td>☀</td>
<td>01…08</td>
<td>✓</td>
<td>–</td>
</tr>
<tr>
<td>* = Station address 2 to 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MFD-Titan output relay</td>
<td>☀</td>
<td>01…08</td>
<td>✓</td>
<td>–</td>
</tr>
<tr>
<td>MFD-Titan output relay expansion, network stations (only network master)</td>
<td>☔</td>
<td>01…08</td>
<td>✓</td>
<td>–</td>
</tr>
<tr>
<td>* = Station address 2 to 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MFD-Titan expansion output relay</td>
<td>☔</td>
<td>01…08</td>
<td>✓</td>
<td>–</td>
</tr>
<tr>
<td>Bit outputs</td>
<td>☔</td>
<td>01…32</td>
<td>✓</td>
<td>–</td>
</tr>
<tr>
<td>* = Station address 1 to 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>General coils</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Markers</td>
<td>M</td>
<td>01…96</td>
<td>✓</td>
<td>–</td>
</tr>
<tr>
<td>COM slave marker (REMOTE MARKER)</td>
<td>1M</td>
<td>01…96</td>
<td>✓</td>
<td>–</td>
</tr>
<tr>
<td>Jump label</td>
<td>:</td>
<td>01…32</td>
<td>✓</td>
<td>–</td>
</tr>
<tr>
<td><strong>Function blocks</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analog value comparator function block</td>
<td>A</td>
<td>01…32</td>
<td>–</td>
<td>✓</td>
</tr>
<tr>
<td>Arithmetic function block</td>
<td>AR</td>
<td>01…32</td>
<td>–</td>
<td>✓</td>
</tr>
<tr>
<td>Data block comparator, activate</td>
<td>BC × EN</td>
<td>01 32</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Transfer data block, trigger coil</td>
<td>BT × T_</td>
<td>01 32</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Boolean sequences</td>
<td>BV</td>
<td>01…32</td>
<td>–</td>
<td>✓</td>
</tr>
<tr>
<td>Counter function block, counter input</td>
<td>C × C_</td>
<td>01…32</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Relay/function block</td>
<td>MFD-Titan display</td>
<td>Number</td>
<td>Coil</td>
<td>Parameter</td>
</tr>
<tr>
<td>-----------------------------------------------------------------</td>
<td>-------------------</td>
<td>----------</td>
<td>------</td>
<td>-----------</td>
</tr>
<tr>
<td>Counter function block, direction</td>
<td>C X D_</td>
<td>x=01…32</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Counter function block, set counter value (Preset)</td>
<td>C X SE</td>
<td>x=01…32</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Counter function block, reset counter value</td>
<td>C X RE</td>
<td>x=01…32</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Frequency counter function block, activate counter (enable)</td>
<td>CF X EN</td>
<td>x=01…04</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>High-speed counter function block, direction</td>
<td>CH X D_</td>
<td>x=01…04</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>High-speed counter function block, activate counter (enable)</td>
<td>CH X EN</td>
<td>x=01…04</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>High-speed counter function block, set counter value (Preset)</td>
<td>CH X SE</td>
<td>x=01…04</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>High-speed counter function block, reset counter value</td>
<td>CH X RE</td>
<td>x=01…04</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Incremental encoder counter function block, set counter value (Preset)</td>
<td>CI X SE</td>
<td>x=01…02</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Incremental encoder counter function block, activate counter (enable)</td>
<td>CI X EN</td>
<td>x=01…02</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Incremental encoder counter function block, reset counter value</td>
<td>CI X RE</td>
<td>x=01…02</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Comparator function block</td>
<td>CP</td>
<td>x=01…32</td>
<td>—</td>
<td>✓</td>
</tr>
<tr>
<td>Activate text output function block (enable)</td>
<td>D X EN</td>
<td>x=01…32</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Data function block, trigger coil</td>
<td>DB X T_</td>
<td>x=01…32</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>PID controller, activate</td>
<td>DC X EN</td>
<td>x=01…32</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>PID controller, activate P component</td>
<td>DC X EP</td>
<td>x=01…32</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>PID controller, activate I component</td>
<td>DC X EI</td>
<td>x=01…32</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>PID controller, activate D component</td>
<td>DC X ED</td>
<td>x=01…32</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>PID controller, accept manual manipulated variable</td>
<td>DC X SE</td>
<td>x=01…32</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Activate signal smoothing filter</td>
<td>FT X EN</td>
<td>x=01…32</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
## Relay/function block

<table>
<thead>
<tr>
<th>Relay/function block</th>
<th>MFD-Titan display</th>
<th>Number</th>
<th>Coil</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get from network station function block</td>
<td>GT</td>
<td>X=01...32</td>
<td>–</td>
<td>✓</td>
</tr>
<tr>
<td>Seven-day time switch</td>
<td>HV</td>
<td>X=01...32</td>
<td>–</td>
<td>✓</td>
</tr>
<tr>
<td>Year time switch function block</td>
<td>HY</td>
<td>X=01...32</td>
<td>–</td>
<td>✓</td>
</tr>
<tr>
<td>Activate value scaling function block</td>
<td>LS X EN</td>
<td>X=01...32</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Master reset function block</td>
<td>MR X T_</td>
<td>X=01...32</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Activate numerical converter function block</td>
<td>NC X EN</td>
<td>X=01...32</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Function block operating hours counter, enable</td>
<td>OT X EN</td>
<td>X=01...4</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Operating hours counter function block, reset</td>
<td>OT X RE</td>
<td>X=01...4</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Send to the network (easy-NET) function block, trigger</td>
<td>PT X T_</td>
<td>X=01...32</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Activate pulse width modulation function block</td>
<td>PW X EN</td>
<td>X=01...2</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Send time to the network (easy-NET) function block, trigger</td>
<td>SC X T_</td>
<td>X=01</td>
<td>✓</td>
<td>–</td>
</tr>
<tr>
<td>Activate set cycle time function block</td>
<td>ST X EN</td>
<td>X=01</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Timing relay function block, trigger control coil (enable)</td>
<td>T X EN</td>
<td>X=01...32</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Timing relay function block, stop</td>
<td>T X ST</td>
<td>X=01...32</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Timing relay function block, reset</td>
<td>T X RE</td>
<td>X=01...32</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Activate value limitation function block</td>
<td>VC X EN</td>
<td>X=01...32</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Markers</th>
<th>MFD-Titan display</th>
<th>Number</th>
<th>Value range</th>
<th>Access type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analog operand</td>
<td></td>
<td></td>
<td></td>
<td>r = Read</td>
</tr>
<tr>
<td>Marker 32 bit</td>
<td>MD</td>
<td>01...96</td>
<td>32 bit</td>
<td>w = Write</td>
</tr>
<tr>
<td>Marker 16 bit</td>
<td>MW</td>
<td>01...96</td>
<td>16 bit</td>
<td></td>
</tr>
<tr>
<td>Marker 8 bit</td>
<td>MB</td>
<td>01...96</td>
<td>8 bit</td>
<td></td>
</tr>
<tr>
<td>Marker 1 bit</td>
<td>M</td>
<td>0...96</td>
<td>1 bit</td>
<td></td>
</tr>
<tr>
<td>Analog inputs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>basic unit</td>
<td>IA X</td>
<td>X=01...04</td>
<td>10 bit</td>
<td></td>
</tr>
<tr>
<td>Analog output</td>
<td>QA X</td>
<td>X=01</td>
<td>10 bit</td>
<td></td>
</tr>
</tbody>
</table>

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.

For Immediate Delivery call KMParts.com at (866) 595-9616
The following rules apply if you want to use selective binary operands (contacts) from the markers MD, MW, MB:

Table 9: T Composition of the markers

<table>
<thead>
<tr>
<th>Applies to MD, MW, MB, M</th>
<th>Left = most significant bit, byte, word</th>
<th>Right = least significant bit, byte, word</th>
</tr>
</thead>
<tbody>
<tr>
<td>32 bit</td>
<td>MD1</td>
<td></td>
</tr>
<tr>
<td>16 bit</td>
<td>MW2</td>
<td>MW1</td>
</tr>
<tr>
<td>8 bit</td>
<td>MB4</td>
<td>MB3</td>
</tr>
<tr>
<td>1 bit</td>
<td>M32 to M25</td>
<td>M24 to M17</td>
</tr>
<tr>
<td>32 bit</td>
<td>MD2</td>
<td></td>
</tr>
<tr>
<td>16 bit</td>
<td>MW4</td>
<td>MW3</td>
</tr>
<tr>
<td>8 bit</td>
<td>MB8</td>
<td>MB7</td>
</tr>
<tr>
<td>1 bit</td>
<td>M64 to M57</td>
<td>M56 to M49</td>
</tr>
<tr>
<td>32 bit</td>
<td>MD3</td>
<td></td>
</tr>
<tr>
<td>16 bit</td>
<td>MW6</td>
<td>MW5</td>
</tr>
<tr>
<td>8 bit</td>
<td>MB12</td>
<td>MB11</td>
</tr>
<tr>
<td>1 bit</td>
<td>M96 to M89</td>
<td>M88 to M81</td>
</tr>
<tr>
<td>32 bit</td>
<td>MD4</td>
<td></td>
</tr>
</tbody>
</table>
### Wiring with MFD-Titan

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

Applies to MD, MW, MB, M

Left = most significant bit, byte, word

Right = least significant bit, byte, word

For Immediate Delivery call KMParts.com at (866) 595-9616
Number formats

MFD makes computations with a signed 31 bit value. The value range is:

\[-2,147,483,648 \text{ to } +2,147,483,647\]

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

Bit 32 = state “0” means a positive number.

Example:

\[0000000000000000000001000010010_\text{bin} = 412_{\text{hex}} = 1042_{\text{dec}}\]

Bit 32 = 1 means a negative number.

Example:

\[1111111111111111111110110010101110_\text{bin} = \text{FFFFDCAE}_{\text{hex}} = -9042_{\text{dec}}\]

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 circuit connections. 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 circuit connection. With the MFD-Titan up to 256 circuit connections 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 circuit connections. 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 circuit connection or one contact plus a coil in series. A total of 16 characters per circuit connection and three circuit connections plus the status line can be displayed simultaneously.

You can move between the contact fields with the `<` `>` cursor buttons. The number of the circuit connection 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 312.

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:**

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

Contact name
Contact number
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 114.
Values for contacts and coil fields are changed in Entry mode. The value to be changed flashes.

MFD-Titan proposes the contact I₀₁ or the coil C₀₁ 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.

<table>
<thead>
<tr>
<th>In the contact field to change I₀₁ to I₀₂</th>
<th>In the coil field change C₀₁ to S₀₀₈</th>
</tr>
</thead>
<tbody>
<tr>
<td>I₀₁</td>
<td>C₀₁</td>
</tr>
<tr>
<td>Q: &gt; or OK</td>
<td>M</td>
</tr>
<tr>
<td>M:</td>
<td>S</td>
</tr>
<tr>
<td>H:</td>
<td>R</td>
</tr>
<tr>
<td>C: 02</td>
<td>T</td>
</tr>
<tr>
<td>T:</td>
<td>C</td>
</tr>
<tr>
<td>P:</td>
<td>D</td>
</tr>
<tr>
<td>P:</td>
<td>S</td>
</tr>
<tr>
<td>T:</td>
<td>:</td>
</tr>
<tr>
<td>:</td>
<td>...</td>
</tr>
<tr>
<td>D:</td>
<td></td>
</tr>
<tr>
<td>S:</td>
<td></td>
</tr>
<tr>
<td>:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Deleting contacts and coils

- Move the cursor using the buttons \(\leftarrow \rightarrow \uparrow \downarrow\) 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 \(2 \times\) to confirm the change.

![Diagram showing the change of contact from make to break](image)

Figure 66: 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 \(\leftarrow \rightarrow \uparrow \downarrow\) 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 circuit connection).
Press **ALT** to switch to Connect mode.

Use < > to move the diagonal arrow between the contact fields and coil fields and ~~ to move between circuit connections.

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 circuit connection, 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 326.

Figure 67: Circuit diagram with five contacts, invalid

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

Figure 68: 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 circuit connections 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 circuit connection

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

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

1. Position the cursor on the **first** contact field of a circuit connection.
2. Press `ALT`.

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

![Figure 69: Inserting a new circuit connection](image)
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.

Abort 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 circuit connection

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

► Press ESC and use the \(<>/\) cursor buttons to select the GO TO menu.
► Press OK.
► Select the required circuit connection (L... ...) with the \(<>/\) cursor buttons.

The first contact on the circuit connection is always indicated.

► Press OK.

The cursor remains stationary at the required circuit connection contact L 1.

Deleting the circuit connection

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

► Delete all the contacts and coils from the circuit connection.
► Position the cursor on the first contact field of the empty circuit connection.
► Press DEL.
The subsequent circuit connection(s) will be “pulled up” and any existing links between circuit connections 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 1 to P 4. 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 and.

![Diagram](image)

Figure 70: 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 1 disconnects the circuit connection I 1.

![Diagram](image)

Figure 71: I5 switches over to the cursor buttons.

The P buttons are only detected as switches in the Status menu.
Working with contacts and relays

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.

![Parallel circuit diagram](image)

Figure 72: 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.

Figure 73: 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 circuit connections 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 87).

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

The following display appears if no function blocks are present.

The cursor flashes.

The following display appears if no function blocks are present.

Press the OK button.

For Immediate Delivery call KMParts.com at (866) 595-9616
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

<table>
<thead>
<tr>
<th>Function block:</th>
<th>Timing relay</th>
</tr>
</thead>
<tbody>
<tr>
<td>Switch function:</td>
<td>On-delayed with random switching</td>
</tr>
<tr>
<td>Time range:</td>
<td>M:S (Minute:Seconds)</td>
</tr>
<tr>
<td>Set time &gt;I1:</td>
<td>20 min 30 s</td>
</tr>
<tr>
<td>Actual time QV&gt;:</td>
<td>Copied to MD96</td>
</tr>
</tbody>
</table>
Assigning operands to an \( \text{> input of a function block} \)

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 \( \text{>QV}\)

Assigning operands to a \( \text{QV > output of a function block} \)

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 \text{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 \text{DEL} button.
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**.

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

- Press the **OK** button.

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.
The operand is displayed.

- \( I_1 \) = Actual value of counter C 01
- \( I_2 \) = Constant 1095
- \( QV \) = Marker double word MD56

Press the **ALT** button again.

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>
<thead>
<tr>
<th>MFD-Titan display</th>
<th>Coil function</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ç</td>
<td>Contactor function</td>
<td>C001, C002, C004, C005, C006, C007,..</td>
</tr>
<tr>
<td>ş</td>
<td>Impulse relay function</td>
<td>S001, S002, S004, S005, S007, S011,..</td>
</tr>
<tr>
<td>s</td>
<td>Set</td>
<td>S001, S002, S003, S004,..</td>
</tr>
<tr>
<td>R</td>
<td>Reset</td>
<td>R001, R002, R003, R004, R005, R006,..</td>
</tr>
<tr>
<td>ş</td>
<td>Contactor function with negated result</td>
<td>J006, J007,..</td>
</tr>
<tr>
<td>F</td>
<td>Cycle pulse with rising edge</td>
<td>F001,..</td>
</tr>
<tr>
<td>Ù</td>
<td>Cycle pulse with falling edge</td>
<td>U002,..</td>
</tr>
</tbody>
</table>

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 $S$, $R$, $J$ can be used several times.

The following applies to non-retentive coil functions such as $C$ (contactor), $F$ (negated contactor), $P$, $L$ (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 $C$

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

![Signal diagram of contactor function](image)

Figure 76: Signal diagram of contactor function

Impulse relay $J$

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

![Signal diagram of impulse relay](image)

Figure 77: 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 316).

“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.

![Signal diagram of “Set” and “Reset”](image)

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

![Simultaneous triggering of $Q_01$](image)

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.

![Signal diagram of inverse contactor function](image)

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.

![Signal diagram of cycle pulse with rising edge](image)

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.

![Signal diagram of cycle pulse with falling edge](image)
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 316).
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 sequences
- 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,
Function blocks

- 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 316).

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.
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 \( I_1 \) greater than or equal to, equal to, less than or equal to function block input \( I_2 \).
- Using the factors \( F_1 \) and \( F_2 \) 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 \( I_1 \) input.
- The \( HY \) function block input is used for the positive and negative switching hysteresis of the input \( I_2 \). The contact switches according to the selected comparison mode of the function block.

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.
Parameter display and parameter set for analog value comparators:

<table>
<thead>
<tr>
<th>&gt;A 02</th>
<th>Function block analog value comparator number 02</th>
</tr>
</thead>
<tbody>
<tr>
<td>GT</td>
<td>Greater than mode</td>
</tr>
<tr>
<td></td>
<td>Appears in the parameter display</td>
</tr>
<tr>
<td>&gt;I1</td>
<td>Comparison value 1</td>
</tr>
<tr>
<td>&gt;F1</td>
<td>Gain factor for &gt;I1 (I1 = F1 x value)</td>
</tr>
<tr>
<td>&gt;I2</td>
<td>Comparison value 2</td>
</tr>
<tr>
<td>&gt;F2</td>
<td>Gain factor for &gt;I2 (I2 = F2 x value)</td>
</tr>
<tr>
<td>&gt;OS</td>
<td>Offset for the value of &gt;I1</td>
</tr>
<tr>
<td>&gt;HV</td>
<td>Switching hysteresis for value &gt;I2 (Value HV applies to positive and negative hysteresis.)</td>
</tr>
</tbody>
</table>

**Inputs**

The function block inputs >I1, >F1, >I2, >F2, >OS and >HV 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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>GT</td>
<td>$&gt; I_1$ greater than $&gt; I_2$</td>
</tr>
<tr>
<td>EQ</td>
<td>$= I_1$ equal to $= I_2$</td>
</tr>
<tr>
<td>LT</td>
<td>$&lt; I_1$ less than $&gt; I_2$</td>
</tr>
</tbody>
</table>

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 84: Signal diagram of the analog value comparator
1: actual value on $> I_1$
2: setpoint value on $> I_2$
3: hysteresis on $> H$
4: switching contact (make contact)
5: offset for value $> I_1$

For Immediate Delivery call KMParts.com at (866) 595-9616
6: actual value plus offset

- **Range A: Compare \( I_1 > I_2 \)
  - The actual value \( I_1 \) 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 \( I_1 < I_2 \)
  - 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 \( I_1 > I_2 \) with offset
  - This example behaves as described in Range A. The offset value is added to the actual value.

- **Comparison \( I_1 = I_2 \)
  - 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 $I_1$ and $I_2$ 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.

Parameter display and parameter set for arithmetic function blocks:

<table>
<thead>
<tr>
<th>AR32</th>
<th>ADD</th>
<th>+</th>
<th>$I_1$</th>
<th>$I_2$</th>
<th>QV&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$I_1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$I_2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QV&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD</td>
<td>Addition of summand ( \times I_1 ) plus summand ( \times I_2 )</td>
</tr>
<tr>
<td>SUB</td>
<td>Subtraction of minuend ( \times I_1 ) minus subtrahend ( \times I_2 )</td>
</tr>
<tr>
<td>MUL</td>
<td>Multiplication of factor ( \times I_1 ) by factor ( \times I_2 )</td>
</tr>
<tr>
<td>DIV</td>
<td>Division of dividend ( \times I_1 ) by divisor ( \times I_2 )</td>
</tr>
</tbody>
</table>

**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**
\[ 42 + 1000 = 1042 \]
\[ 2,147,483,647 + 1 = \text{last valid value of this arithmetic operation, due to overflow (CARRY)} \]
\[ AR..CY = \text{Status 1} \]
\[-2048 + 1000 = -1048\]

**Subtraction**

\[1134 - 42 = 1092\]

\[-2147483648 - 3 = \text{last valid value of this arithmetic operation, due to overflow (CARRY)}\]

\[\text{AR..CY = Status 1}\]

\[-4096 - 1000 = -5096\]

\[-4096 - (-1000) = -3096\]

**Multiplication**

\[12 \times 12 = 144\]

\[1000042 \times 2401 = \text{last valid value of this arithmetic operation, due to overflow (CARRY)}\]

Correct value = 2401100842

\[\text{AR..CY = Status 1}\]

\[-1000 \times 10 = -10000\]

**Division**

\[1024: 256 = 4\]

\[1024: 35 = 29 \text{ (the places after the decimal point are omitted.)}\]

\[1024: 0 = \text{last valid value of this arithmetic operation, due to overflow (CARRY)}\]

(mathematically correct: “Infinite”)

\[\text{AR..CY = Status 1}\]

\[-1000: 10 = -100\]

\[1000: -10 = -100\]

\[-1000: (-10) = 100\]

\[10: 100 = 0\]
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.

![Circuit Diagram](image)

Figure 85: MFD-Titan circuit diagram with enabling of data block comparator function block

Parameter display and parameter set for a data block comparator:

<table>
<thead>
<tr>
<th>BC27</th>
<th>Data block comparator function block number 27</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Appears in the parameter display</td>
</tr>
<tr>
<td>&gt;I1</td>
<td>Start of comparison range 1</td>
</tr>
<tr>
<td>&gt;I2</td>
<td>Start of comparison range 2</td>
</tr>
<tr>
<td>&gt;NO</td>
<td>Number of elements to be compared in bytes per range. Value range 1 to +383</td>
</tr>
</tbody>
</table>

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 \( I_1 \) and \( I_2 \) 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 \( I_1 \) or \( I_2 \):

• 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
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: \( \text{I}1 \text{ MB15} \)
Comparison range 2: \( \text{I}2 \text{ 64} \)
Number of bytes: \( \text{NO} \text{ 4} \)
Marker MB01: 1

Comparison range 2: Constant 64:
MB01 plus Offset: \( 1 + 64 = 65 \rightarrow \) MB65.
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: \( \textit{I1} \text{ MB60} \)
Comparison range 2: \( \textit{I2} \text{ MD80} \)
Number of bytes: \( \textit{NO} \text{ 6} \)

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

---

<table>
<thead>
<tr>
<th>Comparison range 1</th>
<th>Value of marker range 1 (decimal/binary)</th>
<th>Comparison range 2</th>
<th>Value of marker range 2 (decimal/binary)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MB60</td>
<td>45/00101101</td>
<td>MD80 (Byte 1, LSB)</td>
<td>1097219629/010000010110011001111000101101</td>
</tr>
<tr>
<td>MB61</td>
<td>62/00111110</td>
<td>MD80 (Byte 2)</td>
<td>1097219629/01000001011001100011111000101101</td>
</tr>
<tr>
<td>MB62</td>
<td>102/01100110</td>
<td>MD80 (Byte 3)</td>
<td>1097219629/01000001011001100011111000101101</td>
</tr>
<tr>
<td>MB63</td>
<td>65/01000001</td>
<td>MD80 (Byte 4, MSB)</td>
<td>1097219629/01000001011001100011111000101101</td>
</tr>
<tr>
<td>MB64</td>
<td>173/10101101</td>
<td>MD81 (Byte 1, LSB)</td>
<td>15277/00111011010101110</td>
</tr>
<tr>
<td>MB65</td>
<td>59/00111011</td>
<td>MD81 (Byte 2)</td>
<td>15277/0000010010101101</td>
</tr>
</tbody>
</table>

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.

![Figure 86: MFD-Titan circuit diagram with enabling of transfer data block function block](image)

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

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BT07</td>
<td>Data block transfer function block number 07</td>
</tr>
<tr>
<td>INI</td>
<td>INI mode, initialise marker ranges</td>
</tr>
<tr>
<td>+</td>
<td>Appears in the parameter display</td>
</tr>
<tr>
<td>I1</td>
<td>Source range start</td>
</tr>
<tr>
<td>I2</td>
<td>Destination range start</td>
</tr>
<tr>
<td>NO</td>
<td>Number of elements to be written in bytes per range. Value range 1 to +383</td>
</tr>
</tbody>
</table>

Only constants can be modified in the parameter display of a function block.
Operating modes of the transfer data block function block

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>INI</td>
<td>Initialise marker ranges</td>
</tr>
<tr>
<td>CPY</td>
<td>Copy marker ranges</td>
</tr>
</tbody>
</table>

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 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.

Initialising 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: \( >I_1 \ MB_{10} \)
Destination range: \( >I_2 \ MB_{20} \)
Number of bytes: \( >N0 \ 10 \)

<table>
<thead>
<tr>
<th>Source range</th>
<th>Value of source marker range (decimal)</th>
<th>Destination range</th>
<th>Value of destination marker range (decimal)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MB10</td>
<td>123</td>
<td>MB20</td>
<td>123</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MB21</td>
<td>123</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MB22</td>
<td>123</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MB23</td>
<td>123</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MB24</td>
<td>123</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MB25</td>
<td>123</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MB26</td>
<td>123</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MB27</td>
<td>123</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MB28</td>
<td>123</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MB29</td>
<td>123</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Source range</th>
<th>Value of source marker range (decimal)</th>
<th>Destination range</th>
<th>Value of destination marker range (decimal)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MB15</td>
<td>45</td>
<td>MB65</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MB66</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MB67</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MB68</td>
<td>45</td>
</tr>
</tbody>
</table>

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.
### Comparisons

<table>
<thead>
<tr>
<th>Comparison range 1</th>
<th>Value of marker range 1 (decimal/binary)</th>
<th>Comparison range 2</th>
<th>Value of marker range 2 (decimal/binary)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MB60</td>
<td>45/00101101</td>
<td>MD80</td>
<td>757935405/00101101001011010010110100101101</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MD80 (Byte 2)</td>
<td>757935405/00101101001011010010110100101101</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MD80 (Byte 3)</td>
<td>757935405/00101101001011010010110100101101</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MD80 (Byte 4, MSB)</td>
<td>757935405/00101101001011010010110100101101</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MD81 (Byte 1, LSB)</td>
<td>757935405/00101101001011010010110100101101</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MD81 (Byte 2)</td>
<td>757935405/00101101001011010010110100101101</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MD81 (Byte 3)</td>
<td>757935405/00101101001011010010110100101101</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MD81 (Byte 4, MSB)</td>
<td>757935405/00101101001011010010110100101101</td>
</tr>
</tbody>
</table>

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:** \( \text{MB}60 \)
- **Destination range:** \( \text{MD}93 \)
- **Number of bytes:** 18
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: \(\text{\textcircled{I}}1 \text{MB40}\)
Comparison range 2: \(\text{\textcircled{I}}2 \text{MW54}\)
Number of bytes: \(\text{\textcircled{N}}0 \text{C 01QV}\)

The value of C 01QV is 788. This value is too big. The value at \(\text{\textcircled{N}}0\) 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 \(\text{\textcircled{I}}1\). The destination range is specified at \(\text{\textcircled{I}}2\). The length of the source and destination range is specified by the number of bytes at the \(\text{\textcircled{N}}0\) 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: \textgreater{} I1 MB10
Destination range: \textgreater{} I2 MB20
Number of bytes: \textgreater{} NO 10

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

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:  \( > \text{I1 MB15} \)
Destination range:  \( > \text{I2 64} \)
Number of bytes:  \( > \text{NO 4} \)
Marker MB01: 1

Destination range: Constant 64:
Marker MB01 plus Offset: 1 + 64 = 65 \( \rightarrow \text{MB65} \).

<table>
<thead>
<tr>
<th>Source range</th>
<th>Value of source marker range (decimal)</th>
<th>Destination range</th>
<th>Value of destination marker range (decimal)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MB15</td>
<td>68</td>
<td>MB65</td>
<td>68</td>
</tr>
<tr>
<td>MB16</td>
<td>189</td>
<td>MB66</td>
<td>189</td>
</tr>
<tr>
<td>MB17</td>
<td>203</td>
<td>MB67</td>
<td>203</td>
</tr>
<tr>
<td>MB18</td>
<td>3</td>
<td>MB68</td>
<td>3</td>
</tr>
</tbody>
</table>

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:  \( > \text{I1 MD60} \)
Destination range:  \( > \text{I2 MW40} \)
Number of bytes:  \( > \text{NO 12} \)

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.
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`

<table>
<thead>
<tr>
<th>Comparison range 1</th>
<th>Value of marker range 1 (decimal/binary)</th>
<th>Comparison range 2</th>
<th>Value of marker range 2 (decimal/binary)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MD60</td>
<td>866143319/00110011101000000100110001010111</td>
<td>MW40 (LSW)</td>
<td>19543/00110011101000000100110001010111</td>
</tr>
<tr>
<td>MD60</td>
<td>866143319/00110011101000000100110001010111</td>
<td>MW41 (MSW)</td>
<td>13216/00110011101000000100110001010111</td>
</tr>
<tr>
<td>MD61</td>
<td>173304101/000010100101010011010010010101</td>
<td>MW42 (LSW)</td>
<td>26917/000010100101010011010010010101</td>
</tr>
<tr>
<td>MD61</td>
<td>173304101/000010100101010011010010010101</td>
<td>MB43 (MSW)</td>
<td>2644/000010100101010011010010010101</td>
</tr>
<tr>
<td>MD62</td>
<td>982644150/00111010100100001111010110110</td>
<td>MB44 (LSW)</td>
<td>62902/00111010100100001111010110110</td>
</tr>
<tr>
<td>MD62</td>
<td>982644150/00111010100100001111010110110</td>
<td>MB45 (MSW)</td>
<td>14993/00111010100100001111010110110</td>
</tr>
</tbody>
</table>
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: \( \text{I1} \text{ MW60} \)
Comparison range 2: \( \text{I2} \text{ MW64} \)
Number of bytes: \( \text{NO} \ 12 \)

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: \( \text{I1} \text{ MW40} \)
Comparison range 2: \( \text{I2} \text{ MW54} \)
Number of bytes: \( \text{NO} \text{ C 01QV} \)

The message “The specified offset of the destination range is outside of the permissible range” is output.
BT01E3 is 1.

**Boolean sequence**

MFD-Titan provides 32 function blocks from BV01 to BV32 for Boolean operations with values.

The following possibilities are provided by the Boolean sequence function block:

- Masking out of particular bits from values,
- Bit pattern recognition,
- Bit pattern modification.

A Boolean sequence function block is not wired in the circuit diagram.

Parameter display and parameter set for Boolean sequence function block:

<table>
<thead>
<tr>
<th>BV27</th>
<th>AND</th>
<th>( \triangleright I_1 )</th>
<th>( \triangleright I_2 )</th>
<th>QV</th>
<th>+</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean sequence function block number 27</td>
<td>AND operation mode</td>
<td>First value</td>
<td>Second value</td>
<td>Result of the sequence</td>
<td>Appears in the parameter display</td>
</tr>
</tbody>
</table>

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

**Operating modes of the Boolean sequence function block**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>AND</td>
<td>AND operation</td>
</tr>
<tr>
<td>OR</td>
<td>OR operation</td>
</tr>
<tr>
<td>XOR</td>
<td>Exclusive OR operation</td>
</tr>
<tr>
<td>NOT</td>
<td>Negation of the Boolean value of ( \triangleright I_1 )</td>
</tr>
</tbody>
</table>
Value range
32 bit signed value

Inputs
The function block inputs \( \text{I1} \) and \( \text{I2} \) 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 sequence function block does not have coils.

Memory requirement
Boolean sequence function block
The Boolean sequence function block requires 40 bytes of memory plus 4 bytes per constant on the function block inputs.
Function of Boolean sequence function block
The function block creates the sequence depending on the operating mode.

If you sequence a negative value, e.g.: $-10_{dec}$, the CPU will form the two's complement of the amount.

Example:
$-10_{dec} = 100000000000000000000000001010_{bin}$

Two's complement = $111111111111111111111110110_{bin} = FFFFFFF6_{hex}$

Bit 32 is the signed bit and remains as 1.

AND Boolean sequence
Value $I_1$: $13219_{dec} = 0011001110100011_{bin}$
Value $I_2$: $57193_{dec} = 1101111101101001_{bin}$
Result $QV$: $4897_{dec} = 0001001100100001_{bin}$

OR Boolean sequence
Value $I_1$: $13219_{dec} = 0011001110100011_{bin}$
Value $I_2$: $57193_{dec} = 1101111101101001_{bin}$
Result $QV$: $65515_{dec} = 1111111111101011_{bin}$

XOR Boolean sequence
Value $I_1$: $13219_{dec} = 0011001110100011_{bin}$
Value $I_2$: $57193_{dec} = 1101111101101001_{bin}$
Result $QV$: $60618_{dec} = 1110110011001010_{bin}$
**NOT Boolean sequence**

Value \( >I_1 \) = 13219\(_{\text{dec}}\) = 000000000000000001100110100011111010011\(_{\text{bin}}\)

Value \( >I_2 \) = Omitted

Result QV\( > \) = \(-13220\)\(_{\text{dec}}\) = 111111111111111100110001011100\(_{\text{bin}}\)

The NOT sequence operates according to the following rules:

\( >I_1 \), positive value  
Negate value of \( >I_1 \) and subtract 1:  
\[-|>I_1| - 1 = >I_2\]

\( >I_1 \), Negative value  
Value of \( >I_1 \) and subtract 1:  
\[|>I_1| - 1 = >I_2\]

**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 1200, 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.
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 $\text{SH}$, $\text{SL}$, and $\text{SV}$ 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 \( \rightarrow \) 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 88: Signal diagram of counter

1: counter coil C..C_
2: upper setpoint value \( \geq \text{SH} \)
3: preset actual value \( \geq \text{SV} \)
4: lower setpoint value \( \geq \text{SL} \)
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

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

Attention!

If an input is used several times, the last counter in the function block list is executed:
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.

![Frequency counter function block diagram](image)

- **CF01** - Frequency counter function block number 01
- **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

- $1 \text{ kHz} = 1000$

**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

1: counter input I1 to I4
2: upper setpoint value $>$ $>$H
3: lower setpoint value $>$ $>$L
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.
Parameter display and parameter set for high-speed counters:

<table>
<thead>
<tr>
<th>CH01</th>
<th>High-speed counter function block number 01</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Appears in the parameter display</td>
</tr>
<tr>
<td>&gt;SH</td>
<td>Upper setpoint</td>
</tr>
<tr>
<td>&gt;SL</td>
<td>Lower setpoint</td>
</tr>
<tr>
<td>&gt;SV</td>
<td>Defined actual value (Preset)</td>
</tr>
<tr>
<td>QV&gt;</td>
<td>Actual value in RUN mode</td>
</tr>
</tbody>
</table>

In the parameter display of a counter relay you change setpoint values and/or the preset value and 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 \( \geq \) Upper setpoint
- CH01FB to CH04FB: Actual value \( \leq \) Lower setpoint
- CH01ZE to CH04ZE: Actual value \( = \) Zero
- CH01CY to CH04CY: Value range exceeded

For Immediate Delivery call KMParts.com at (866) 595-9616
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 92: 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
You integrate a counter into your circuit in the form of a contact and coil. The counter relay has different coils and contacts.

Figure 93: MFD-Titan circuit diagram with high-speed incremental encoder counter

Parameter display and parameter set for high-speed incremental encoder counter:

<table>
<thead>
<tr>
<th><strong>CI01</strong></th>
<th>High-speed incremental encoder counter function block number 01</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>+</strong></td>
<td>Appears in the parameter display</td>
</tr>
<tr>
<td><strong>(\text{US})</strong></td>
<td>Upper setpoint</td>
</tr>
<tr>
<td><strong>(\text{UL})</strong></td>
<td>Lower setpoint</td>
</tr>
<tr>
<td><strong>(\text{SV})</strong></td>
<td>Defined actual value (Preset)</td>
</tr>
<tr>
<td><strong>(\text{QV})</strong></td>
<td>Actual value in RUN mode</td>
</tr>
</tbody>
</table>

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 \(-2147483648\) to \(2147483647\).

Each pulse is counted twice.

Example: value at CI.\(\text{QV}\) = 42000
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 94: Signal diagram of high-speed incremental encoder counter

1: counter input channel A
2: counter input channel B
3: upper setpoint value \( \geq \text{SH} \)
4: preset actual value \( \geq \text{SV} \)
5: lower setpoint value \( \geq \text{SL} \)
6: counter enable

For Immediate Delivery call KMParts.com at (866) 595-9616
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:

<table>
<thead>
<tr>
<th>Function block input</th>
<th>Greater than</th>
<th>Equal to</th>
<th>Less than</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;I1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&gt;I2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Parameter display and parameter set for the comparator function block:

<table>
<thead>
<tr>
<th>CP02</th>
<th>Function block analog value comparator number 02</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Appears in the parameter display</td>
</tr>
<tr>
<td>( \geq I_1 )</td>
<td>Comparison value 1</td>
</tr>
<tr>
<td>( \geq I_2 )</td>
<td>Comparison value 2</td>
</tr>
</tbody>
</table>

**Inputs**
The function block inputs \( \geq I_1 \) and \( \geq I_2 \) 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 \( \geq I_1 \) is less than the value on \( \geq I_2 \); \( \geq I_1 < \geq I_2 \).

- CP01EQ to CP32EQ, (equal to)
  Contact (make contact) switches to status 1, if the value on \( \geq I_1 \) is equal to the value on \( \geq I_2 \); \( \geq I_1 = \geq I_2 \).

- CP01GT to CP32GT, (greater than)
Contact (make contact) switches to status 1, if the value at \( I_1 \) is greater than the value at \( I_2 \); \( I_1 > I_2 \).

**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!

---

Figure 96: 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.

<table>
<thead>
<tr>
<th>GT01Q1 -----------------------------------</th>
<th>DB16T</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB16Q1 ------------------------------------</td>
<td>D02EN</td>
</tr>
</tbody>
</table>

Figure 97: MFD-Titan circuit diagram with data function block:

Parameter display and parameter set for the data function block:

<table>
<thead>
<tr>
<th>DB16</th>
<th>Data block function block number 16</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Appears in the parameter display</td>
</tr>
<tr>
<td>&gt;I1</td>
<td>Input value</td>
</tr>
<tr>
<td>QV</td>
<td>Actual value</td>
</tr>
</tbody>
</table>

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**

1: value at input >I1
2: trigger coil DB..T
3: value on DB..QV>

Figure 98: Signal diagram of data function block

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.

Parameter display and parameter set for PID controller:

<table>
<thead>
<tr>
<th>DC02</th>
<th>PID controller function block number 02</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNP</td>
<td>Unipolar mode</td>
</tr>
<tr>
<td>+</td>
<td>Appears in the parameter display</td>
</tr>
<tr>
<td>I1</td>
<td>Setpoint of PID controller</td>
</tr>
<tr>
<td>I2</td>
<td>Actual value of PID controller</td>
</tr>
<tr>
<td>KP</td>
<td>Proportional gain K_p</td>
</tr>
<tr>
<td>TN</td>
<td>Reset time T_n</td>
</tr>
<tr>
<td>TV</td>
<td>Rate time T_v</td>
</tr>
</tbody>
</table>
Function blocks

### Operating modes of the PID controller

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Manipulated variable is output as</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNP</td>
<td>Unipolar 12-bit value 0 to +4095</td>
</tr>
<tr>
<td>BIP</td>
<td>Bipolar 13-bit value (signed 12-bit value) –4096 to +4095</td>
</tr>
</tbody>
</table>

### Inputs

The function block inputs $\times I1$, $\times I2$, $\times KP$, $\times TN$, $\times TV$, $\times TC$ and $\times MV$ 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

<table>
<thead>
<tr>
<th>&gt;I1</th>
<th>Setpoint of PID controller</th>
<th>Value range</th>
<th>Resolution/unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;I2</td>
<td>Actual value of PID controller</td>
<td>−32768 to +32767</td>
<td></td>
</tr>
<tr>
<td>&gt;KP</td>
<td>Proportional gain $K_p$</td>
<td>0 to 65535</td>
<td>in -- /%</td>
</tr>
<tr>
<td>&gt;TN</td>
<td>Reset time $T_n$</td>
<td>0 to 65535</td>
<td>in 100/ms</td>
</tr>
<tr>
<td>&gt;TV</td>
<td>Rate time $T_v$</td>
<td>0 to 65535</td>
<td>in 100/ms</td>
</tr>
<tr>
<td>&gt;TC</td>
<td>Scan time</td>
<td>0 to 65535</td>
<td>in 100/ms</td>
</tr>
<tr>
<td>&gt;MV</td>
<td>Manual manipulated variable</td>
<td>−4096 to +4095</td>
<td></td>
</tr>
<tr>
<td>QV&gt;</td>
<td>Manipulated variable</td>
<td>0 to 4095 (unipolar)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>−4096 to +4095 (bipolar)</td>
<td></td>
</tr>
</tbody>
</table>

Example:

<table>
<thead>
<tr>
<th>&gt;KP</th>
<th>Proportional gain $K_p$</th>
<th>Value at input</th>
<th>Value processed in the function block.</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;KP</td>
<td>Proportional gain $K_p$</td>
<td>1500</td>
<td>15</td>
</tr>
<tr>
<td>&gt;TN</td>
<td>Reset time $T_n$</td>
<td>250</td>
<td>25 s</td>
</tr>
<tr>
<td>&gt;TV</td>
<td>Rate time $T_v$</td>
<td>200</td>
<td>20 s</td>
</tr>
<tr>
<td>&gt;TC</td>
<td>Scan time</td>
<td>500</td>
<td>50 s</td>
</tr>
<tr>
<td>&gt;MV</td>
<td>Manual manipulated variable</td>
<td>500</td>
<td>500</td>
</tr>
</tbody>
</table>

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.

\[
Y(t) = Y_P(t) + Y_I(t) + Y_D(t)
\]

- 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 timer
\( X_i(t) \) = Actual value with scan timer

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:

\[
Y_i(t) = K_p \times \frac{T_c}{T_n} \times [X_s(t) - X_i(t)] + Y_i(t-1)
\]

\( K_p \) = proportional gain
\( T_c \) = Scan time
\( T_n \) = Integration time (also known as reset time)
\( X_s(t) \) = Setpoint with scan timer
\( X_i(t) \) = Actual value with scan timer
\( Y_i(t-1) \) = Value of the integral component of the manipulated variable with scan timer \( t-1 \)

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:

\[
Y_d(t) = K_p \times \frac{T_v}{T_c} \times (X_i(t-1) - X_i(t))
\]

\( K_p \) = proportional gain
\( T_c \) = Scan time
\( T_v \) = Differential time of the control system (also called the rate time)
Function blocks

\[ X_i(t) = \text{Actual value with scan time } t \]
\[ X_i(t-1) = \text{Actual value with scan time } t - 1 \]

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 247) in order to keep the cycle time of the device constant.

Manual mode of the PID controller
A value must be present at the \( \gg MV \) input in order to set the manipulated variable directly. If the coil DC..SE is activated, the value at \( \gg MV \) 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 \( \gg MV \) 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 with MFD-Titan

Wiring a signal smoothing filter
You can integrate a signal smoothing filter into your circuit as a coil.

Figure 100: MFD-Titan circuit diagram with smoothing function block

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 $I_1$, $I_2$ 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

<table>
<thead>
<tr>
<th>I1</th>
<th>Input value of the function block</th>
<th>Value range</th>
<th>Resolution/unit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>(-32,768 \text{ to } +32,767)</td>
<td></td>
</tr>
<tr>
<td>TG</td>
<td>Recovery time ( T_g )</td>
<td>(0 \text{ to } 65,535)</td>
<td>in 100/ms</td>
</tr>
<tr>
<td>KP</td>
<td>Proportional gain ( K_p )</td>
<td>(0 \text{ to } 65,535)</td>
<td>in (\text{-- /}%)</td>
</tr>
<tr>
<td>V</td>
<td>Output value</td>
<td>(-32,768 \text{ to } +32,767)</td>
<td></td>
</tr>
</tbody>
</table>

#### Example:

<table>
<thead>
<tr>
<th>TG</th>
<th>Recovery time ( T_g )</th>
<th>Value at input</th>
<th>Value processed in the function block.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>250</td>
<td>25 s</td>
</tr>
<tr>
<td>KP</td>
<td>Proportional gain ( K_p )</td>
<td>1500</td>
<td>15</td>
</tr>
</tbody>
</table>

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:

$$ Y(t) = \left[ \frac{T_a}{T_g} \right] \times \left[ K_p \times x(t) - Y(t-1) \right] $$

- $Y(t)$ = 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.

<table>
<thead>
<tr>
<th>Recovery time $T_g$</th>
<th>Scan time $T_a$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1 s to 1 s</td>
<td>10 ms</td>
</tr>
<tr>
<td>1 s to 6553 s</td>
<td>$T_g \times 0.01$</td>
</tr>
</tbody>
</table>

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.
Function blocks

Parameter display and parameter set for the GET function block:

<table>
<thead>
<tr>
<th>GT01</th>
<th>GET function block (fetch a value from the network), number 01</th>
</tr>
</thead>
<tbody>
<tr>
<td>02</td>
<td>Station number from which the value is sent. Possible station number: 01 to 08</td>
</tr>
<tr>
<td>20</td>
<td>Send function block (PT 20) of the sending station. Possible function block number: 01 to 32</td>
</tr>
<tr>
<td>+</td>
<td>Appears in the parameter display</td>
</tr>
<tr>
<td>QV&gt;</td>
<td>Actual value from the network</td>
</tr>
</tbody>
</table>

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 282).
Function of the GET function block

![Signal diagram of GET function block](image)

1: GT..Q1
2: value on GT..QV>

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 305.

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 Section “Technical data”, Page 345 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.
Parameter display and parameter set for the 7-day time switch HW:

<table>
<thead>
<tr>
<th>HW14</th>
<th>7-day time switch function block number 14</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Time switch channel A</td>
</tr>
<tr>
<td>+</td>
<td>Appears in the parameter display</td>
</tr>
<tr>
<td>&gt;DY1</td>
<td>Day 1</td>
</tr>
<tr>
<td>&gt;DY2</td>
<td>Day 2</td>
</tr>
<tr>
<td>&gt;ON</td>
<td>On time</td>
</tr>
<tr>
<td>&gt;OFF</td>
<td>Off time</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>HW01 A</th>
<th>HW01 B</th>
</tr>
</thead>
<tbody>
<tr>
<td>DY1 MO</td>
<td>DY1 MO</td>
</tr>
<tr>
<td>DY2 FR</td>
<td>DY2 FR</td>
</tr>
<tr>
<td>ON 06:30</td>
<td>ON 17:00</td>
</tr>
<tr>
<td>OFF 09:30</td>
<td>OFF 22:30</td>
</tr>
</tbody>
</table>
**Switching at the weekend**
Time switch HW02 switches on at 16:00 on Friday and switches off at 6:00 on Monday.

<table>
<thead>
<tr>
<th>HW02 A</th>
<th>+</th>
</tr>
</thead>
<tbody>
<tr>
<td>DY1</td>
<td>FR</td>
</tr>
<tr>
<td>DY2</td>
<td></td>
</tr>
<tr>
<td>ON</td>
<td>16:00</td>
</tr>
<tr>
<td>OFF</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HW02 B</th>
<th>+</th>
</tr>
</thead>
<tbody>
<tr>
<td>DY1</td>
<td>MO</td>
</tr>
<tr>
<td>DY2</td>
<td></td>
</tr>
<tr>
<td>ON</td>
<td></td>
</tr>
<tr>
<td>OFF</td>
<td>06:00</td>
</tr>
</tbody>
</table>

Figure 104: Signal diagram of “weekend”

**Overnight switching**
Time switch HW03 switches on overnight at 22:00 Monday and switches off at 6:00 on Tuesday.

<table>
<thead>
<tr>
<th>HW03 D</th>
<th>+</th>
</tr>
</thead>
<tbody>
<tr>
<td>DY1</td>
<td>MO</td>
</tr>
<tr>
<td>DY2</td>
<td></td>
</tr>
<tr>
<td>ON</td>
<td>22:00</td>
</tr>
<tr>
<td>OFF</td>
<td>06:00</td>
</tr>
</tbody>
</table>

Figure 105: 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.

![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.

![Signal diagram of overlaps]

- **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.

<table>
<thead>
<tr>
<th>HW20 A +</th>
<th>HW20 B +</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;DY1 MO</td>
<td>&gt;DY1 TU</td>
</tr>
<tr>
<td>&gt;DY2</td>
<td>&gt;DY2</td>
</tr>
<tr>
<td>&gt;ON 00:00</td>
<td>&gt;ON</td>
</tr>
<tr>
<td>&gt;OFF</td>
<td>&gt;OFF 00:00</td>
</tr>
</tbody>
</table>

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 305.

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 Section “Technical data”, Page 345 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.

Parameter display and parameter set for the year time switch HY:

<table>
<thead>
<tr>
<th>HY30</th>
<th>Year time switch function block number 30</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>Time switch channel B</td>
</tr>
<tr>
<td>+</td>
<td>Appears in the parameter display</td>
</tr>
<tr>
<td>&gt;ON</td>
<td>Switch on time</td>
</tr>
<tr>
<td>&gt;OFF</td>
<td>Switch off time</td>
</tr>
</tbody>
</table>

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 and switch off at 00:00 01.01.2010.

Months
ON: 04 to OFF: 10 means:
Switch on at 00:00 on 1st April and switch off at 00:00 on 1st October

Days
ON: 02 to OFF: 25 means:
Switch on at 00:00 on the 2nd and switch off at 00:00 on the 25th

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.
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.

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.

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.

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”
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 the 17th of these months.

<table>
<thead>
<tr>
<th>HY01 A</th>
<th>HY01 B</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON 03.05.--</td>
<td>ON 02.06.--</td>
</tr>
<tr>
<td>OFF 25.10.--</td>
<td>OFF 17.12.--</td>
</tr>
</tbody>
</table>

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.

Figure 108: MFD-Titan circuit diagram with LS value scaling
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

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Value range</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1</td>
<td>Input value of the function block</td>
<td>-2147483648 to +2147483647</td>
</tr>
<tr>
<td>X1</td>
<td>Lower value of source range</td>
<td></td>
</tr>
<tr>
<td>X2</td>
<td>Lower value of target range</td>
<td></td>
</tr>
<tr>
<td>Y1</td>
<td>Upper value of source range</td>
<td></td>
</tr>
<tr>
<td>Y2</td>
<td>Upper value of target range</td>
<td></td>
</tr>
<tr>
<td>QV</td>
<td>Output value</td>
<td></td>
</tr>
</tbody>
</table>

#### 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

For Immediate Delivery call KMParts.com at (866) 595-9616
Figure 109: Value scaling function block – Reduce 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.
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.

The actual value at the analog input DC01QV is 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

<table>
<thead>
<tr>
<th>Contact Make contact(^1)</th>
<th>(\cdot)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numbers</td>
<td>01 to 32</td>
</tr>
<tr>
<td>Coils</td>
<td>(\xi)</td>
</tr>
<tr>
<td>Numbers</td>
<td>01 to 32</td>
</tr>
<tr>
<td>Coil function</td>
<td>(\xi, \jmath, \phi, \theta)</td>
</tr>
</tbody>
</table>

1) can only be used as first leftmost contact

**Function**

If the jump coil is triggered, the circuit connections coming directly after it will not be processed. The states of the coils before the jump will be retained, unless they are overwritten in circuit connections 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 circuit connection 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 \(\xi\):1/jumped range/Contact:1,
- Coil \(\xi\):1/jumped range/Contact:1 etc.
Attention!
If circuit connections 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 sequences 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
- T0 1 Wait time 30.00 s, on-delayed
- D0 1 Text “motor-protective circuit-breaker tripped”
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.

Parameter display and parameter set for the master reset function block:

<table>
<thead>
<tr>
<th>MR16</th>
<th>Master reset function block number 16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ø</td>
<td>Reset outputs mode</td>
</tr>
<tr>
<td>+</td>
<td>Appears in the parameter display</td>
</tr>
</tbody>
</table>
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.

![MFD-Titan circuit diagram with numerical converter](image)

Parameter display and parameter set for the numerical converter:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCD</td>
<td>Convert BCD coded values to decimal values</td>
</tr>
<tr>
<td>BIN</td>
<td>Convert decimal value to BCD coded values</td>
</tr>
</tbody>
</table>

In the parameter display of a numerical converter you can change the mode and the enable of the parameter display.

### Numerical converter modes

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCD</td>
<td>Convert BCD coded values to decimal values</td>
</tr>
<tr>
<td>BIN</td>
<td>Convert decimal value to BCD coded values</td>
</tr>
</tbody>
</table>

### Number range

<table>
<thead>
<tr>
<th>Value</th>
<th>Number system</th>
</tr>
</thead>
<tbody>
<tr>
<td>$-161061273$ to $+161061273$</td>
<td>BCD</td>
</tr>
<tr>
<td>$-9999999$ to $+9999999$</td>
<td>Decimal</td>
</tr>
</tbody>
</table>
## Inputs

The function block input \( I_1 \) 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

---

### BCD code

<table>
<thead>
<tr>
<th>BCD code</th>
<th>Decimal value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001</td>
<td>1</td>
</tr>
<tr>
<td>0010</td>
<td>2</td>
</tr>
<tr>
<td>0011</td>
<td>3</td>
</tr>
<tr>
<td>0100</td>
<td>4</td>
</tr>
<tr>
<td>0101</td>
<td>5</td>
</tr>
<tr>
<td>0110</td>
<td>6</td>
</tr>
<tr>
<td>0111</td>
<td>7</td>
</tr>
<tr>
<td>1000</td>
<td>8</td>
</tr>
<tr>
<td>1001</td>
<td>9</td>
</tr>
<tr>
<td>1010 to 1111</td>
<td>Not permissible</td>
</tr>
<tr>
<td>10000</td>
<td>10</td>
</tr>
<tr>
<td>10001</td>
<td>11</td>
</tr>
</tbody>
</table>

The BCD code only allows the number range \( 0_{\text{hex}} \) to \( 9_{\text{hex}} \). The number range \( A_{\text{hex}} \) to \( F_{\text{hex}} \) cannot be represented. The NC function block converts the impermissible range to 9.
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
Example 3:  
Input value \( I_1: 19_{\text{dec}} \)  
Binary value: 00010011  
Decimal value \( QV>: 13 \)

Example 4:  
Input value \( I_1: 161061273_{\text{dec}} \)  
Binary value: 1001100110011001100110011001  
Decimal value \( QV>: 9999999 \)

Example 5:  
Input value \( I_1: -61673_{\text{dec}} \)  
Binary value: 10000000000000001111000011101001  
Decimal value \( QV>: -9099 \)

Bit 32 is the sign bit. Bit 32 = 1 \( \rightarrow \) Sign = Minus.

Example 6:  
Input value \( I_1: 2147483647_{\text{dec}} \)  
Binary value: 01111111111111111111111111111111  
Decimal value \( QV>: 9999999 \)

Values greater than 161061273 are output as 9999999. Values less than \( -161061273 \) are output as \( -9999999 \). The working range of the function block has been exceeded.

**BIN mode**  
The decimal value is assigned to input \( I_1 \). 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_{\text{dec}}
BCD binary value: 0111
Hexadecimal value: 0111
Decimal value QV>: + 7

Example 2:
Input value >I1: +11_{\text{dec}}
BCD binary value: 00010001
Hexadecimal value: 00010001
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_{\text{dec}}
BCD binary value: 00011001
Hexadecimal value: 00011001
Decimal value QV>: 25 (1 + 8 + 16)

Example 4:
Input value >I1: 9999999_{\text{dec}}
BCD binary value: 1001100110011001100110011001
Hexadecimal value: 1001100110011001100110011001
Decimal value QV>: 161061273

Example 5:
Input value >I1: –61673_{\text{dec}}
BCD binary value:
10000000000001100001011001110011
Hexadecimal value:
10000000000001100001011001110011
Decimal value QV>: –398963

Example 6:
Input value >I1: 2147483647_{\text{dec}}
BCD binary value:
01111111111111111111111111111111

Bit 32 is the sign bit. Bit 32 = 1 \rightarrow \text{Sign} = \text{Minus}.
Hexadecimal value:
01111111111111111111111111111111
Decimal value QV>: 161061273

Values greater than 9999999 are output as 161061273. Values less than –9 999999 are output as –161061273. 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.

![Figure 113:MFD-Titan circuit diagram with operating hours counter.](image)

Parameter display and parameter set for the operating hours counter function block:

<table>
<thead>
<tr>
<th>OT04</th>
<th>Operating hours counter number 04</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Appears in the parameter display</td>
</tr>
<tr>
<td>&gt;I1</td>
<td>Upper threshold value in hours</td>
</tr>
<tr>
<td>QV&gt;</td>
<td>Actual value of the operating hours counter</td>
</tr>
</tbody>
</table>

**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

For Immediate Delivery call KMParts.com at (866) 595-9616
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.

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.
Parameter display and parameter set for the PUT function block:

<table>
<thead>
<tr>
<th></th>
<th>PUT function block (places a value onto the network), number 11</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Does not appear in the parameter display</td>
</tr>
<tr>
<td>&gt;I1</td>
<td>Setpoint value which is put onto the easy-NET network</td>
</tr>
</tbody>
</table>

**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 282).
Function of the 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.
Parameter display and parameter set for pulse width modulation:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value and time range</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>SV</td>
<td>0 to 4095</td>
<td>1 digit</td>
</tr>
<tr>
<td>PD</td>
<td>0 to 65535</td>
<td>m</td>
</tr>
<tr>
<td>ME</td>
<td>0 to 65535</td>
<td>ms</td>
</tr>
</tbody>
</table>

The minimum time setting for the period duration is: 0.005 s (5 ms)

Inputs

The function block inputs $\gg SV$, $\gg PD$ and $\gg ME$ 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 smallest permissible minimum on duration must not be less than 100 µs.
- 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).

Figure 117: 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 282).

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.

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value and time range</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1</td>
<td>0 to 1000 ms</td>
<td>ms</td>
</tr>
</tbody>
</table>

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.
Function blocks

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--------------------------Ä T 02EN
I 02--------------------------Ä T 02RE
T 02Q1------------------------Ä Q 01
I 03--------------------------Ä T 02ST

Figure 119:MFD-Titan circuit diagram with timing relay.

For Immediate Delivery call KMParts.com at (866) 595-9616
Parameter display and parameter set for timing relay:

<table>
<thead>
<tr>
<th>T 02</th>
<th>Timing relay function block number 02</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>On-delayed mode</td>
</tr>
<tr>
<td>M:S</td>
<td>Time range Minute: Seconds</td>
</tr>
<tr>
<td>+</td>
<td>Appears in the parameter display</td>
</tr>
<tr>
<td>&gt;I1</td>
<td>Time setpoint value 1</td>
</tr>
<tr>
<td>&gt;I2</td>
<td>Time setpoint value I2 (on a timing relay with 2 setpoint values)</td>
</tr>
<tr>
<td>QV&gt;</td>
<td>Timed-out actual time in RUN mode</td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Switch function</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>On-delayed switching</td>
</tr>
<tr>
<td>?X</td>
<td>On-delayed with random time range</td>
</tr>
<tr>
<td>■</td>
<td>Off-delayed switching</td>
</tr>
<tr>
<td>?■</td>
<td>Off-delayed with random time range</td>
</tr>
<tr>
<td>×■</td>
<td>On and off delayed</td>
</tr>
<tr>
<td>□</td>
<td>Off-delayed with random time range, setpoint retriggerable</td>
</tr>
<tr>
<td>?□</td>
<td>Off-delayed with random time range, retriggerable</td>
</tr>
<tr>
<td>?×□</td>
<td>On and off delayed switching with random time, 2 time setpoints</td>
</tr>
<tr>
<td>▼</td>
<td>Single-pulse switching</td>
</tr>
<tr>
<td>▼</td>
<td>Switch with flashing, synchronous, 2 time setpoint values</td>
</tr>
<tr>
<td>▼</td>
<td>Switch with flashing, asynchronous, 2 time setpoint values</td>
</tr>
</tbody>
</table>
## Function blocks

### Time range

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Time range and setpoint time</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ \text{Seconds, 0.005 to 2147483.645 s (596 h) for constants and variable values} $</td>
<td>5 ms</td>
<td></td>
</tr>
<tr>
<td>M:$ \text{Minutes: Seconds 00:00 to 99:59 only for constants and variable values} $</td>
<td>1 s</td>
<td></td>
</tr>
<tr>
<td>H:M $ \text{Hours: Minutes, 00:00 to 99:59 only for constants and variable values} $</td>
<td>1 min.</td>
<td></td>
</tr>
</tbody>
</table>

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 $I_1$ and $I_2$ 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)”.

Example:
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 9,500 s.
Operand value 45507
The time value is 45,510 s.

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 120: 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 121: 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 sequence. 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 122: Signal diagram of timing relay, on-delayed
(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

\[ t_s = t_1 + t_2 \]

\[ t_1 \quad t_2 \quad t_s \]

- 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.
Function blocks

Figure 123: Signal diagram of timing relay, on-delayed (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 124: 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 125: 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 126: 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 127: Signal diagram of timing relay, single pulse

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.
Function blocks

Figure 128: 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 129: 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.

![Figure 130: MFD-Titan circuit diagram with VC value limitation](image-url)

Parameter display and parameter set for the VC function block:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VC27</td>
<td>VC value limitation function block number 27</td>
</tr>
<tr>
<td>+</td>
<td>Appears in the parameter display</td>
</tr>
<tr>
<td>&gt;I1</td>
<td>Input value</td>
</tr>
<tr>
<td>&gt;SH</td>
<td>Upper limit value</td>
</tr>
<tr>
<td>&gt;SL</td>
<td>Lower limit value</td>
</tr>
<tr>
<td>&gt;QV&gt;</td>
<td>Output value limited</td>
</tr>
</tbody>
</table>
Inputs
The function block inputs \( \text{I1, SH} \) and \( \text{SL} \) 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

<table>
<thead>
<tr>
<th></th>
<th>Value range</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{I1} )</td>
<td>Input value ( -2147483648 ) to ( +2147483647 )</td>
</tr>
<tr>
<td>( \text{SH} )</td>
<td>Upper limit value</td>
</tr>
<tr>
<td>( \text{SL} )</td>
<td>Lower limit value</td>
</tr>
<tr>
<td>QV&gt;</td>
<td>Output value</td>
</tr>
</tbody>
</table>

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 131:Hardwiring with relays
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.

For Immediate Delivery call KMParts.com at (866) 595-9616
**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 01 | C 01 |

*Figure 134: 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**.

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.
The timing relay works like a flashing relay. The MFD-Titan symbol for the flashing relay is \( \text{\textvisiblespace} \). 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 \( \text{s} \) for seconds.

- Move the cursor to the right over the + character in order to input the time setpoint value \( \text{I}_1 \).

If the same setpoint value is input for \( \text{I}_1 \) and \( \text{I}_2 \), 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 \text{OK}.
- Press \text{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 \text{C 01} and press \text{OK}.

The parameter set for the counter is displayed with actual and setpoint values.

- Move the cursor \( \text{v} \) downwards until you see the value \( \text{QV} \).

- Switch the input \text{I}_5. The actual value changes.

On the display \text{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.
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

MFD-Titan is a display, HMI, closed-loop and open-loop control and input device in one. In the following description, the term “visualization” is used for the display and operator function.

This chapter is still in preparation. The complete chapter will be available in the next edition of this manual.

All visualization functions can only be programmed in EASY-SOFT-PRO.

The Help system in EASY-SOFT-PRO provides descriptions for all “visualization” requirements.
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.
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

<table>
<thead>
<tr>
<th>Physical location, place</th>
<th>Station number</th>
<th>Loop through the unit</th>
<th>T connector and spur line</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Example 1</td>
<td>Example 2</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>easy800 MFD easy...E</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>4</td>
<td>easy800 MFD easy...E</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>8</td>
<td>easy800 MFD easy...E</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>7</td>
<td>easy800 MFD easy...E</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>2</td>
<td>easy800 MFD easy...E</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>6</td>
<td>easy800 MFD easy...E</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>5</td>
<td>easy800 MFD easy...E</td>
</tr>
</tbody>
</table>

- **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.

For Immediate Delivery call KMParts.com at (866) 595-9616
### Position and addressing of the operands via easy-NET

<table>
<thead>
<tr>
<th>Stations</th>
<th>Basic unit</th>
<th>Local expansion</th>
<th>Network bit data</th>
<th>Network word data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Input I</td>
<td>Output Q</td>
<td>Input R</td>
<td>Output S</td>
</tr>
<tr>
<td>1</td>
<td>1 I 1 to 16</td>
<td>1 Q 1 to 8</td>
<td>1 R 1 to 16</td>
<td>1 S 1 to 8</td>
</tr>
<tr>
<td>2</td>
<td>2 I 1 to 16</td>
<td>2 Q 1 to 8</td>
<td>2 R 1 to 16</td>
<td>2 S 1 to 8</td>
</tr>
<tr>
<td>3</td>
<td>3 I 1 to 16</td>
<td>3 Q 1 to 8</td>
<td>3 R 1 to 16</td>
<td>3 S 1 to 8</td>
</tr>
<tr>
<td>4</td>
<td>4 I 1 to 16</td>
<td>4 Q 1 to 8</td>
<td>4 R 1 to 16</td>
<td>4 S 1 to 8</td>
</tr>
<tr>
<td>5</td>
<td>5 I 1 to 16</td>
<td>5 Q 1 to 8</td>
<td>5 R 1 to 16</td>
<td>5 S 1 to 8</td>
</tr>
<tr>
<td>6</td>
<td>6 I 1 to 16</td>
<td>6 Q 1 to 8</td>
<td>6 R 1 to 16</td>
<td>6 S 1 to 8</td>
</tr>
<tr>
<td>7</td>
<td>7 I 1 to 16</td>
<td>7 Q 1 to 8</td>
<td>7 R 1 to 16</td>
<td>7 S 1 to 8</td>
</tr>
<tr>
<td>8</td>
<td>8 I 1 to 16</td>
<td>8 Q 1 to 8</td>
<td>8 R 1 to 16</td>
<td>8 S 1 to 8</td>
</tr>
</tbody>
</table>

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.

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.

For Immediate Delivery call KMParts.com at (866) 595-9616
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 279).

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 279).

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 (Section “Technical data”, Page 345).

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.

\[ t_{\text{new}} = t_p \times (1 + n) \]

- \( t_{\text{new}} \) = 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 138: 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:

<table>
<thead>
<tr>
<th>Baud rate [KB]</th>
<th>Stations must send a “sign of life” every ... [ms]</th>
<th>Stations recognise the absence of a “sign of life” signal after [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>60</td>
<td>180</td>
</tr>
<tr>
<td>500</td>
<td>60</td>
<td>180</td>
</tr>
<tr>
<td>250</td>
<td>120</td>
<td>360</td>
</tr>
<tr>
<td>125</td>
<td>240</td>
<td>720</td>
</tr>
<tr>
<td>50</td>
<td>600</td>
<td>1800</td>
</tr>
<tr>
<td>20</td>
<td>1500</td>
<td>4500</td>
</tr>
<tr>
<td>10</td>
<td>3000</td>
<td>9000</td>
</tr>
</tbody>
</table>

If the absence of a “sign of life” is detected, the respective diagnostics contact is set to 1.

<table>
<thead>
<tr>
<th>Diagnostics contact</th>
<th>Station number</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID 01</td>
<td>1</td>
</tr>
<tr>
<td>ID 02</td>
<td>2</td>
</tr>
<tr>
<td>ID 03</td>
<td>3</td>
</tr>
<tr>
<td>ID 04</td>
<td>4</td>
</tr>
<tr>
<td>ID 05</td>
<td>5</td>
</tr>
<tr>
<td>ID 06</td>
<td>6</td>
</tr>
<tr>
<td>ID 07</td>
<td>7</td>
</tr>
<tr>
<td>ID 08</td>
<td>8</td>
</tr>
</tbody>
</table>
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 KBaup
- 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 \(1.12 \times 10^{10}\) messages per year of which \(1.12 \times 10^{7}\) 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 139: COM-LINK connection to an easy800 or another MFD
Establishing a COM-LINK connection to an easy-NET station.

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:

Figure 140: easy-NET operation and COM-LINK connections.
### Introduction to COM-LINK

#### Write/read accesses in the marker range

<table>
<thead>
<tr>
<th>Active station</th>
<th>Remote station</th>
</tr>
</thead>
<tbody>
<tr>
<td>MD01</td>
<td>MD01</td>
</tr>
<tr>
<td>MW01</td>
<td>MW01</td>
</tr>
<tr>
<td>MB01 1MB02</td>
<td>MB01 MB02</td>
</tr>
<tr>
<td>MB01 1MB02 1MB03 1MB04</td>
<td>MB01 MB02 MB03 MB04</td>
</tr>
<tr>
<td>M01 to M32</td>
<td>M01 to M32</td>
</tr>
<tr>
<td>MD02</td>
<td>MD02</td>
</tr>
<tr>
<td>MW03</td>
<td>MW03</td>
</tr>
<tr>
<td>MB05 1MB06 1MB07 1MB08</td>
<td>MB05 MB06 MB07 MB08</td>
</tr>
<tr>
<td>M33 to M64</td>
<td>M33 to M64</td>
</tr>
<tr>
<td>MD03</td>
<td>MD03</td>
</tr>
<tr>
<td>MW05</td>
<td>MW05</td>
</tr>
<tr>
<td>MB09 1MB10 1MB11 1MB12</td>
<td>MB09 MB10 MB11 MB12</td>
</tr>
<tr>
<td>M65 to M96</td>
<td>M65 to M96</td>
</tr>
<tr>
<td>MD04</td>
<td>MD04</td>
</tr>
<tr>
<td>MW07 1MW08</td>
<td>MW07 MW08</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For Immediate Delivery call KMParts.com at (866) 595-9616
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.

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

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.

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.
The time required to detect that the COM-LINK is not working properly depends on the baud rate selected and the event concerned.

<table>
<thead>
<tr>
<th>Baud rate</th>
<th>Time required for detection of faulty COM-LINK connection.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CRC error (data content incorrect)</td>
</tr>
<tr>
<td></td>
<td>No response, no hardware connection, remote station not in operation</td>
</tr>
<tr>
<td>9600 Baud</td>
<td>250 ms</td>
</tr>
<tr>
<td>19200 Baud</td>
<td>120 ms</td>
</tr>
</tbody>
</table>

**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.

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,
  – ^ v set a value between 0 to 9.
Password protection

► Save the new password by pressing OK.

Use OK to exit the password display and proceed with ESC and ñ 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

► 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 FUNCTION: 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…

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.
- Press **OK** to enter the password entry menu.

   ![Password Entry Menu]

MFD will display the password entry field.

- Set the password using the cursor buttons.
- Confirm with **OK**.

   ![Password Entry Field]

If the password is correct, MFD will return automatically to the Status display.

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 protection

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.

You have entered an incorrect password?
► Re-enter the password.

After the fourth entry attempt MFD will ask whether you wish to delete the circuit diagram and data.
► Press
  – ESC: No entry 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.

<table>
<thead>
<tr>
<th>Language</th>
<th>Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>ENGLISH</td>
</tr>
<tr>
<td>German</td>
<td>DEUTSCH</td>
</tr>
<tr>
<td>French</td>
<td>FRANCAIS</td>
</tr>
<tr>
<td>Spanish</td>
<td>ESPANOL</td>
</tr>
<tr>
<td>Italian</td>
<td>ITALIANO</td>
</tr>
<tr>
<td>Portuguese</td>
<td>PORTUGUES</td>
</tr>
<tr>
<td>Dutch</td>
<td>NEDERLANDS</td>
</tr>
<tr>
<td>Swedish</td>
<td>SVENSKA</td>
</tr>
<tr>
<td>Polish</td>
<td>POLSKI</td>
</tr>
<tr>
<td>Turkish</td>
<td>TURKCE</td>
</tr>
</tbody>
</table>

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.

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.

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.

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.
- Select SET CLOCK.
- 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.

- 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.

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).

- Select the required variant and press the OK button.

Manual selection

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.

The input delay (debounce) is set with the DEBOUNCE menu item.

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.
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.

The P buttons are activated and deactivated via 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. 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 models 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 299).

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.
**Startup behaviour**

- Otherwise select RUN MODE and press **OK**. RUN mode is activated.
- Press **ESC** to return to the Status display.

**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.

<table>
<thead>
<tr>
<th>Startup behaviour</th>
<th>Menu display</th>
<th>Status of MFD after startup</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFD starts in STOP mode</td>
<td><strong>RUN MODE</strong></td>
<td>MFD is in STOP mode</td>
</tr>
<tr>
<td>MFD starts in RUN mode</td>
<td><strong>RUN MODE</strong></td>
<td>MFD is in RUN mode</td>
</tr>
</tbody>
</table>

**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 models 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 299).

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.

▶ 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.
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 299).

#### Activating an automatic startup in TERMINAL MODE

Requirement: The MFD is in RUN or STOP mode without visualization (the System menu must be reachable).
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 77)

**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 299).
Select the **SYSTEM** menu.

Press the **OK** button.

Use the **\(\downarrow\)** button to select the **DISPLAY** menu and press **OK**.

The menus for setting the contrast and backlight are displayed.

Press the **OK** button and move to the contrast entry field.

Use the **\(\uparrow\)** and **\(\downarrow\)** cursor buttons to set the contrast to a value between \(-2\) and \(+2\).

Select your setting.

Complete your setting by pressing **OK**.

The contrast setting will be retained until it is modified.

Use the cursor buttons **\(\uparrow\)** and **\(\downarrow\)** to move to the **LIGHTING** menu.

Press the **OK** button.

Use cursor buttons **\(\uparrow\)** and **\(\downarrow\)** 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.
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 299).

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.
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.

Example:
MB 01 to MB 04, C 12 to C 16, CH 00 to CH 00, CI 00 to CI 00, 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 the MFD 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.

For Immediate Delivery call KMParts.com at (866) 595-9616
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 299).

Select the SYSTEM menu.

Press the OK button.

Use the button to select the INFORMATION menu and press OK.

This will display all device information.
Example: MFD-80-B, MFD-CP8-NT, MFD-TA17

Display in STOP mode.

Example: MFD-80-B, MFD-CP8-ME, MFD-R16

Display in RUN mode.

The CRC checksum is not displayed.

Press **ESC** to leave the display.
8 Inside MFD

MFD Program cycle

In conventional control systems, a relay or contactor control processes all the circuit connections 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 MFD circuit diagram is processed cyclically every 0.1 to 40 ms.

During this time, MFD passes through six segments in succession.

How MFDevaluates 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 circuit connection 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 masks to be displayed have to be loaded. This operation is repeated every 200 ms. With large masks this can take up to one millisecond of cycle time.

The time required for loading in the event of a mask change depends on the size of the masks to be loaded. During a mask change, the new mask is loaded from the mask memory into the RAM.

The loading time can be calculated as follows:
mask size in byte multiplied by 80 µs.

Example:

Mask size 250 bytes:
The loading time for the mask is: 250 × 80 µs = 20 ms

If you require the MFD to have a small cycle time:
use several small masks so that the loading time is not too long during a mask change. Only display necessary information in the masks concerned.

The loading of mask data and mask 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 eas--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 Q01 is “latched” via contact Q01.

- **1st cycle**: Inputs I1 and I2 are switched on. Coil Q1 picks up.
- Contact Q01 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 Q01.

**Example: Do not wire backwards**

This example is shown in Section “Creating and modifying connections” Page 131. It was used here to illustrate how NOT to do it.

In the third circuit connection, MFD finds a connection to the second circuit connection 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-----I 03-----I 05----C M 01
I 02-----I 04----M 01----------C Q 02
```

Figure 141: 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 three different memories.

- The working memory or RAM, size 8 KByte
  The RAM only stores the data when the device power supply is active.
- The mask memory, size 24 KByte
  The mask 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 masks to be displayed. This has a direct effect on the size of the program and the masks. The number of retentive data bytes reduces the memory available for program and masks. The largest mask to be displayed likewise reduces the memory available for the program.
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.

Only use as much retentive data as is actually required.

The mask with the largest memory requirement reduces the memory available for the program. Several smaller masks 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.

Figure 142: MFD input assigned a switch
Delay times for the MFD inputs

The delay time for DC signals is 20 ms.

An input signal S1 must therefore be 15 V or 8 V (DA) 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.
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.

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.

<table>
<thead>
<tr>
<th>State</th>
<th>Outputs</th>
<th>I16, R15 or R16</th>
</tr>
</thead>
<tbody>
<tr>
<td>No fault found</td>
<td></td>
<td>“0” = switched off (make contact)</td>
</tr>
<tr>
<td>At least one output has a fault</td>
<td></td>
<td>“1” = switched on (make contact)</td>
</tr>
</tbody>
</table>

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**

![Figure 145:Circuit diagram for fault output via I16](image)

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**

![Figure 146:Output of operating state](image)
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**

![Circuit Diagram]

Figure 147: 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 45).

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 331).

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.

Figure 148: 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.
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 masks,
- 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
  - Card mode.

► Insert the memory card in the open interface slot.
Loading and saving programs

Figure 150: 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.
Program and card operations

1. Switch to STOP mode.
2. Select PROGRAM… from the main menu.
3. Select the CARD… menu option.

The CARD… menu option will only appear if you have inserted a functional memory card.

You can transfer a circuit diagram from MFD to 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 MFD may 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**

1. Select DEVICE-CARD.
2. 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 and saving programs

Loading a circuit diagram from the card

► Select the CARD-> DEVICE menu option.
► Press OK to confirm the prompt if you want to delete the MFD memory and replace it with the card content.

Press ESC to cancel.

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.

Deleting a circuit diagram on the card

► 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 if the circuit diagram uses function relays which the MFDunit does not recognise:
The analog value comparator function relay is only available on the 24 V DC devices easy-DC and easy-DA.

→ 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 151: Fitting and removing EASY800-PC-CAB

- After transmission, remove the cable and close the cover.
Every MFD-Titan 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

<table>
<thead>
<tr>
<th>MFD-80.. display/operating unit</th>
<th>MFD-80..</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Front dimensions</strong></td>
<td><strong>With keys [mm]</strong></td>
</tr>
<tr>
<td></td>
<td><strong>[inches]</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Without keys [mm]</strong></td>
</tr>
<tr>
<td><strong>Total dimensions with fixing shaft</strong></td>
<td><strong>With keys [mm]</strong></td>
</tr>
<tr>
<td></td>
<td><strong>[inches]</strong></td>
</tr>
<tr>
<td><strong>Thickness of fixing wall (without intermediate top-hat rail)</strong></td>
<td><strong>Minimum; maximum</strong></td>
</tr>
<tr>
<td></td>
<td><strong>[mm]</strong></td>
</tr>
<tr>
<td></td>
<td><strong>[inches]</strong></td>
</tr>
<tr>
<td><strong>Thickness of fixing wall (with intermediate top-hat rail)</strong></td>
<td><strong>Minimum; maximum</strong></td>
</tr>
<tr>
<td></td>
<td><strong>[mm]</strong></td>
</tr>
<tr>
<td></td>
<td><strong>[inches]</strong></td>
</tr>
<tr>
<td><strong>Weight</strong></td>
<td><strong>[g]</strong></td>
</tr>
<tr>
<td></td>
<td><strong>[lb]</strong></td>
</tr>
<tr>
<td><strong>Mounting</strong></td>
<td><strong>Maximum tightening torque of the fixing rings [Nm]</strong></td>
</tr>
</tbody>
</table>

For Immediate Delivery call KMParts.com at (866) 595-9616
### Protective membrane

**MFD-XM-80**

<table>
<thead>
<tr>
<th>Dimensions</th>
<th>W × H × D</th>
</tr>
</thead>
<tbody>
<tr>
<td>[mm]</td>
<td>88 × 88 × 25</td>
</tr>
<tr>
<td>[inches]</td>
<td>3.46 × 3.46 × 0.98</td>
</tr>
<tr>
<td>Weight</td>
<td>[g]</td>
</tr>
<tr>
<td></td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>[lb]</td>
</tr>
<tr>
<td></td>
<td>0.055</td>
</tr>
<tr>
<td>Mounting</td>
<td>Is fitted over the display/keypad (with Titan front ring)</td>
</tr>
</tbody>
</table>

### Protective cover

**MFD-XS-80**

<table>
<thead>
<tr>
<th>Dimensions</th>
<th>W × H × D</th>
</tr>
</thead>
<tbody>
<tr>
<td>[mm]</td>
<td>86.5 × 94 × 25</td>
</tr>
<tr>
<td>[inches]</td>
<td>3.41 × 3.41 × 0.98</td>
</tr>
<tr>
<td>Weight</td>
<td>[g]</td>
</tr>
<tr>
<td></td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>[lb]</td>
</tr>
<tr>
<td></td>
<td>0.079</td>
</tr>
<tr>
<td>Mounting</td>
<td>Is fitted over the display/keypad (without Titan front ring)</td>
</tr>
</tbody>
</table>

### Power supply/CPU module

**MFD-CP8..**

<table>
<thead>
<tr>
<th>Dimensions</th>
<th>W × H × D</th>
</tr>
</thead>
<tbody>
<tr>
<td>[mm]</td>
<td>107.5 × 90 × 30</td>
</tr>
<tr>
<td>[inches]</td>
<td>4.23 × 3.54 × 1.18</td>
</tr>
</tbody>
</table>
## Technical data

### Power supply/CPU module

<table>
<thead>
<tr>
<th></th>
<th>MFD-CP8..</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Weight</strong></td>
<td></td>
</tr>
<tr>
<td>[g]</td>
<td>145</td>
</tr>
<tr>
<td>[lb]</td>
<td>0.32</td>
</tr>
<tr>
<td><strong>Mounting</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>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)</td>
</tr>
</tbody>
</table>

### Inputs/outputs

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dimensions when fitted</strong></td>
<td></td>
</tr>
<tr>
<td>W × H × D [mm]</td>
<td>89 × 90 × 25</td>
</tr>
<tr>
<td>[inches]</td>
<td>3.5 × 3.54 × 0.98</td>
</tr>
<tr>
<td><strong>Dimensions when removed</strong></td>
<td></td>
</tr>
<tr>
<td>W × H × D [mm]</td>
<td>89 × 90 × 44</td>
</tr>
<tr>
<td>[inches]</td>
<td>3.5 × 3.54 × 1.73</td>
</tr>
<tr>
<td><strong>Weight</strong></td>
<td></td>
</tr>
<tr>
<td>MFD-R..; MFD-T.. [g]</td>
<td>150; 140</td>
</tr>
<tr>
<td>MFD-R..; MFD-T.. [lb]</td>
<td>0.33; 0.31</td>
</tr>
<tr>
<td><strong>Mounting</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Snap fitted into the power supply module</td>
</tr>
</tbody>
</table>
Dimensions of the MFD-80.. display/operating unit

Dimensions of the MFD-80-XM protective membrane

Dimensions of the MFD-80-XS protective cover

For Immediate Delivery call KMParts.com at (866) 595-9616
Dimensions of the MFD-CP8.. power supply/CPU module

### General ambient conditions

<table>
<thead>
<tr>
<th>Climatic conditions</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(damp heat constant to IEC 60068-2-78; cyclical to IEC 600618-2-30)</td>
<td></td>
</tr>
<tr>
<td>(cold to IEC 60068-2-1, heat to IEC 60068-2-2)</td>
<td></td>
</tr>
<tr>
<td><strong>Ambient temperature</strong></td>
<td>°C, (°F)</td>
</tr>
<tr>
<td>Installed horizontally/vertically</td>
<td>–25 to 55, (–13 to 131)</td>
</tr>
<tr>
<td><strong>Condensation (power supply unit/CPU; inputs/outputs)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Display legibility</strong></td>
<td>°C, (°F)</td>
</tr>
<tr>
<td></td>
<td>–5 to 50, (–23 to 122)</td>
</tr>
<tr>
<td><strong>Storage/transport temperature</strong></td>
<td>°C, (°F)</td>
</tr>
<tr>
<td></td>
<td>–40 to 70, (–40 to 158)</td>
</tr>
<tr>
<td><strong>Relative humidity (IEC 60068-2-30), non-condensing</strong></td>
<td>%</td>
</tr>
<tr>
<td></td>
<td>5 to 95</td>
</tr>
<tr>
<td><strong>Air pressure (operation)</strong></td>
<td>hPa</td>
</tr>
<tr>
<td></td>
<td>795 to 1080</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ambient mechanical conditions</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pollution degree</strong></td>
<td></td>
</tr>
<tr>
<td>Power supply unit/CPU; inputs/outputs</td>
<td>2</td>
</tr>
<tr>
<td>Display/operating unit</td>
<td>3</td>
</tr>
<tr>
<td><strong>Degree of protection (EN 50178, IEC 60529, VBG4)</strong></td>
<td></td>
</tr>
<tr>
<td>Power supply unit/CPU; inputs/outputs</td>
<td>IP20</td>
</tr>
<tr>
<td>Display/operating unit</td>
<td>IP65</td>
</tr>
<tr>
<td>Display/operating unit with protective cover</td>
<td>IP65</td>
</tr>
<tr>
<td>Display/operating unit with protective membrane</td>
<td>IP65</td>
</tr>
<tr>
<td><strong>Oscillations (IEC 60068-2-6)</strong></td>
<td></td>
</tr>
<tr>
<td>Constant amplitude 0.15 mm</td>
<td>Hz</td>
</tr>
<tr>
<td></td>
<td>10 to 57</td>
</tr>
<tr>
<td>Constant acceleration 2 g</td>
<td>Hz</td>
</tr>
<tr>
<td></td>
<td>57 to 150</td>
</tr>
<tr>
<td><strong>Shocks (IEC 60068-2-27)</strong></td>
<td></td>
</tr>
<tr>
<td>semi-sinusoidal 15 g/11 ms</td>
<td>Shocks</td>
</tr>
<tr>
<td></td>
<td>18</td>
</tr>
<tr>
<td><strong>Drop (IEC 60068-2-31)</strong></td>
<td></td>
</tr>
<tr>
<td>Drop height</td>
<td>mm</td>
</tr>
<tr>
<td></td>
<td>50</td>
</tr>
<tr>
<td><strong>Free fall, when packed (IEC 60068-2-32)</strong></td>
<td>m</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

For Immediate Delivery call KMParts.com at (866) 595-9616
**Technical data**

### Electromagnetic compatibility (EMC)

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrostatic discharge (ESD), (IEC/EN 61000-4-2, severity level 3)</td>
<td></td>
</tr>
<tr>
<td>Air discharge</td>
<td>8 kV</td>
</tr>
<tr>
<td>Contact discharge</td>
<td>6 kV</td>
</tr>
<tr>
<td>Electromagnetic fields (RFI), (IEC/EN 61000-4-3)</td>
<td>10 V/m</td>
</tr>
<tr>
<td>Radio interference suppression (EN 55011, EN 55022), limit class</td>
<td>B</td>
</tr>
<tr>
<td>Burst (IEC/EN 61000-4-4, severity level 3)</td>
<td></td>
</tr>
<tr>
<td>Power cables</td>
<td>2 kV</td>
</tr>
<tr>
<td>Signal cables</td>
<td>2 kV</td>
</tr>
<tr>
<td>High energy pulses (Surge) MFD (IEC/EN 61000-4-5, severity level 2, power cable symmetrical)</td>
<td>0.5 kV</td>
</tr>
<tr>
<td>Line-conducted interference (IEC/EN 61000-4-6)</td>
<td>10 V</td>
</tr>
</tbody>
</table>

### Dielectric strength

Measurement of the air clearance and creepage distance: EN 50178, UL 508, CSA C22.2, No 142

### Tools and cable cross-sections

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solid, minimum to maximum</td>
<td>0.2 to 4 mm²</td>
</tr>
<tr>
<td></td>
<td>24 to 12 AWG</td>
</tr>
<tr>
<td>Flexible with ferrule, minimum to maximum</td>
<td>0.2 to 2.5 mm²</td>
</tr>
<tr>
<td></td>
<td>24 to 12 AWG</td>
</tr>
<tr>
<td>Slot-head screwdriver, width</td>
<td>3.5 × 0.5 mm</td>
</tr>
<tr>
<td></td>
<td>0.14 × 0.02 inch</td>
</tr>
</tbody>
</table>
### Display/operating unit

<table>
<thead>
<tr>
<th>Feature</th>
<th>MFD-80, MFD80-B</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Power supply</strong></td>
<td></td>
</tr>
<tr>
<td>Power supply</td>
<td></td>
</tr>
<tr>
<td>Power supply using power supply unit/CPU MFD-CP8..</td>
<td></td>
</tr>
<tr>
<td><strong>LCD display</strong></td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>Graphic/monochrome</td>
</tr>
<tr>
<td>Visible area W x H</td>
<td>mm 62 x 33</td>
</tr>
<tr>
<td>Size of pixels</td>
<td>mm 0.4 x 0.4</td>
</tr>
<tr>
<td>Number of pixels (W x H)</td>
<td>mm 132 x 64</td>
</tr>
<tr>
<td>Spacing (pixel centre to pixel centre)</td>
<td>mm 0.42</td>
</tr>
<tr>
<td><strong>LCD backlight</strong></td>
<td></td>
</tr>
<tr>
<td>Backlight colour</td>
<td>Yellow/green</td>
</tr>
<tr>
<td>The backlight can be used and programmed in visualization applications</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>LEDs</strong></td>
<td></td>
</tr>
<tr>
<td>The backlight can be used and programmed in visualization applications</td>
<td>2</td>
</tr>
<tr>
<td><strong>Operating buttons</strong></td>
<td></td>
</tr>
<tr>
<td>Number</td>
<td>9</td>
</tr>
<tr>
<td>Can be used and programmed in visualization applications</td>
<td>9</td>
</tr>
<tr>
<td><strong>Pushbutton illumination (LED)</strong></td>
<td></td>
</tr>
<tr>
<td>Number</td>
<td>5</td>
</tr>
<tr>
<td>Colour</td>
<td>Green</td>
</tr>
</tbody>
</table>
### Power supply

<table>
<thead>
<tr>
<th></th>
<th>MFD-CP8...</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Rated voltage</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Rated value</strong></td>
<td>V DC, (%)</td>
</tr>
<tr>
<td><strong>Permissible range</strong></td>
<td>V DC</td>
</tr>
<tr>
<td><strong>Residual ripple</strong></td>
<td>%</td>
</tr>
<tr>
<td><strong>Input current</strong></td>
<td>mA</td>
</tr>
<tr>
<td>For 24 V DC, MFD-CP8.., normally</td>
<td>125</td>
</tr>
<tr>
<td>For 24 V DC, MFD-CP8.., MFD-80.., normally</td>
<td>250</td>
</tr>
<tr>
<td>For 24 V DC, MFD-CP8.., MFD-80.., MFD-R.., MFD-T.., normally</td>
<td>20</td>
</tr>
<tr>
<td><strong>Voltage dips, IEC/EN 61131-2</strong></td>
<td>ms</td>
</tr>
<tr>
<td><strong>Heat dissipation</strong></td>
<td></td>
</tr>
<tr>
<td>For 24 V DC, MFD-CP8.., normally</td>
<td>W</td>
</tr>
<tr>
<td>For 24 V DC, MFD-CP8.., MFD-80.., normally</td>
<td>W</td>
</tr>
<tr>
<td>For 24 V DC, MFD-CP8.., MFD-80.., MFD-R.., MFD-T.., normally</td>
<td>W</td>
</tr>
</tbody>
</table>

### CPU, real-time clock/timing relay/memory

#### Backup/accuracy of real-time clock (see graph)

![Graph showing the accuracy of the real-time clock over time at different temperatures.](image)

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Accuracy of the real-time clock</strong></td>
<td></td>
</tr>
<tr>
<td>Per day</td>
<td>s/day</td>
</tr>
<tr>
<td>Per year</td>
<td>h/year</td>
</tr>
<tr>
<td></td>
<td>± 5</td>
</tr>
<tr>
<td></td>
<td>± 0.5</td>
</tr>
</tbody>
</table>

For Immediate Delivery call KMParts.com at (866) 595-9616
<table>
<thead>
<tr>
<th>Repetition accuracy of timing relays</th>
<th>%</th>
<th>± 0.02</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolution</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Range “s”</td>
<td>ms</td>
<td>5</td>
</tr>
<tr>
<td>Range “M:S”</td>
<td>s</td>
<td>1</td>
</tr>
<tr>
<td>Range “H:M”</td>
<td>min</td>
<td>1</td>
</tr>
<tr>
<td>Circuit connections</td>
<td></td>
<td>256</td>
</tr>
<tr>
<td>Contacts in series</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>Coil per circuit connection</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Program memory for program/circuit diagram</td>
<td>KByte</td>
<td>8</td>
</tr>
<tr>
<td>Program memory for display objects (visualization)</td>
<td>KByte</td>
<td>24</td>
</tr>
<tr>
<td>RAM working memory</td>
<td>KByte</td>
<td>8</td>
</tr>
<tr>
<td>Storage of programs (retentive)</td>
<td>FRAM</td>
<td></td>
</tr>
<tr>
<td>Retentive memory (retentive data, retentive)</td>
<td>FRAM</td>
<td></td>
</tr>
<tr>
<td>Size</td>
<td>Byte</td>
<td>200</td>
</tr>
<tr>
<td>Operating hours counter</td>
<td>Byte</td>
<td>16</td>
</tr>
<tr>
<td>Write-read cycles FRAM (minimum)</td>
<td></td>
<td>$10^{10}$</td>
</tr>
</tbody>
</table>
### Technical data

**Inputs**

<table>
<thead>
<tr>
<th><strong>Digital inputs</strong></th>
<th>MFD-R.., MFD-T..</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Number</strong></td>
<td>12</td>
</tr>
<tr>
<td><strong>Inputs usable as analog inputs, (I7, I8, I11, I12)</strong></td>
<td>4</td>
</tr>
<tr>
<td><strong>Status display</strong></td>
<td>LCD status display, if available</td>
</tr>
<tr>
<td><strong>Electrical isolation</strong></td>
<td></td>
</tr>
<tr>
<td>To supply voltage</td>
<td>No</td>
</tr>
<tr>
<td>Between each other</td>
<td>No</td>
</tr>
<tr>
<td>To the outputs</td>
<td>Yes</td>
</tr>
<tr>
<td>To PC interface, memory card, easy-NET network, EASY-LINK</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Rated voltage</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Rated value</strong> V DC</td>
<td>24</td>
</tr>
<tr>
<td><strong>0 signal</strong></td>
<td></td>
</tr>
<tr>
<td>I1 to I6 and I9 to I10 V DC</td>
<td>&lt; 5</td>
</tr>
<tr>
<td>I7, I8, I11, I12 V DC</td>
<td>&lt; 8</td>
</tr>
<tr>
<td><strong>On 1 signal</strong></td>
<td></td>
</tr>
<tr>
<td>I1 to I6 and I9 to I10 V DC</td>
<td>&gt; 15</td>
</tr>
<tr>
<td>I7, I8, I11, I12 V DC</td>
<td>&gt; 8</td>
</tr>
<tr>
<td><strong>Input current on 1 signal</strong></td>
<td></td>
</tr>
<tr>
<td>I1 to I6, I9 to I10 at 24 V DC</td>
<td>3.3</td>
</tr>
<tr>
<td>I7, I8, I11, I12 at 24 V DC</td>
<td>2.2</td>
</tr>
<tr>
<td><strong>Delay time for 0 to 1</strong></td>
<td></td>
</tr>
<tr>
<td>Debounce ON ms</td>
<td>20</td>
</tr>
<tr>
<td>Debounce off, typical</td>
<td></td>
</tr>
<tr>
<td>I1 to I4 ms</td>
<td>0.025</td>
</tr>
<tr>
<td>I5, I6, I9, I10 ms</td>
<td>0.25</td>
</tr>
<tr>
<td>I7, I8, I11, I12 ms</td>
<td>0.15</td>
</tr>
</tbody>
</table>

For Immediate Delivery call KMParts.com at (866) 595-9616
### Delay time for “1” to “0”

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Debounce ON</td>
<td>20 ms</td>
</tr>
<tr>
<td>Debounce off, typical</td>
<td></td>
</tr>
<tr>
<td>I1 to I4</td>
<td>0.025 ms</td>
</tr>
<tr>
<td>I5, I6, I9, I10</td>
<td>0.25 ms</td>
</tr>
<tr>
<td>I7, I8, I11, I12</td>
<td>0.15 ms</td>
</tr>
</tbody>
</table>

### Cable length (unshielded)

- **Value**: 100 m

### High-speed counter inputs, I1 to I4

- **Number**: 4
- **Cable length (shielded)**: 20 m

### High-speed up and down counters

- **Counting frequency**: < 3 kHz
- **Pulse shape**: Square wave
- **Mark-to-space ratio**: 1:1

### Frequency counters

- **Counting frequency**: < 3 kHz
- **Pulse shape**: Square wave
- **Mark-to-space ratio**: 1:1

### Incremental encoder counters

- **Counting frequency**: < 3 kHz
- **Pulse shape**: Square wave
- **Counter inputs I1 and I2, I3 and I4**: 2
- **Signal offset**: 90°
- **Mark to space ratio**: 1:1
### Analog inputs

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>4</td>
</tr>
<tr>
<td>Electrical isolation</td>
<td></td>
</tr>
<tr>
<td>To supply voltage</td>
<td>No</td>
</tr>
<tr>
<td>To the digital inputs</td>
<td>No</td>
</tr>
<tr>
<td>To the outputs</td>
<td>Yes</td>
</tr>
<tr>
<td>To the easy-NET network</td>
<td>Yes</td>
</tr>
<tr>
<td>Input type</td>
<td>DC voltage</td>
</tr>
<tr>
<td>Signal range</td>
<td>V DC</td>
</tr>
<tr>
<td>Resolution analog</td>
<td>V</td>
</tr>
<tr>
<td>Resolution digital</td>
<td>Bit</td>
</tr>
<tr>
<td>Value</td>
<td>0 to 1023</td>
</tr>
<tr>
<td>Input impedance</td>
<td>kΩ</td>
</tr>
<tr>
<td>Accuracy</td>
<td>± 3</td>
</tr>
<tr>
<td>Two MFD devices, from actual value</td>
<td></td>
</tr>
<tr>
<td>Within a unit, from actual value, (I7, I8, I11, I12)</td>
<td>± 2</td>
</tr>
<tr>
<td>Conversion time, analog/digital</td>
<td>ms</td>
</tr>
<tr>
<td>Debounce ON:</td>
<td>20</td>
</tr>
<tr>
<td>Debounce OFF:</td>
<td>Each cycle time</td>
</tr>
<tr>
<td>Input current</td>
<td>mA</td>
</tr>
<tr>
<td>Cable length (shielded)</td>
<td>m</td>
</tr>
<tr>
<td></td>
<td>30</td>
</tr>
</tbody>
</table>

For Immediate Delivery call KMParts.com at (866) 595-9616
## Relay outputs

<table>
<thead>
<tr>
<th></th>
<th>MFD-R..</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>4</td>
</tr>
<tr>
<td>Type of outputs</td>
<td>Relays</td>
</tr>
<tr>
<td>In groups of</td>
<td>1</td>
</tr>
<tr>
<td>Connection of outputs in parallel to increase the output</td>
<td>Not permissible</td>
</tr>
<tr>
<td>Protection for an output relay</td>
<td></td>
</tr>
<tr>
<td>Miniature circuit-breaker B16</td>
<td>A 16</td>
</tr>
<tr>
<td>or fuse (slow-blow)</td>
<td>A 8</td>
</tr>
<tr>
<td>Potential isolation to mains power supply, input, PC interface, memory card, NET network, EASY-LINK</td>
<td>Yes</td>
</tr>
<tr>
<td>Safe isolation</td>
<td>V AC 300</td>
</tr>
<tr>
<td>Basic insulation</td>
<td>V AC 600</td>
</tr>
<tr>
<td>Mechanical lifespan</td>
<td>Switch Operations $10 \times 10^6$</td>
</tr>
<tr>
<td>Contacts relays</td>
<td></td>
</tr>
<tr>
<td>Conventional thermal current, (UL)</td>
<td>A 8, (10)</td>
</tr>
<tr>
<td>Recommended for load at 12 V AC/DC</td>
<td>mA &gt; 500</td>
</tr>
<tr>
<td>Protected against short-circuit cos $\varphi$ = 1 16 A characteristic B (B16) at</td>
<td>A 600</td>
</tr>
<tr>
<td>Protected against short-circuit cos $\varphi$ = 0.5 to 0.7 16 A characteristic B (B16) at</td>
<td>A 900</td>
</tr>
<tr>
<td>Rated impulse withstand voltage $U_{\text{imp}}$ contact coil</td>
<td>kV 6</td>
</tr>
<tr>
<td>Rated insulation voltage $U_i$</td>
<td></td>
</tr>
<tr>
<td>Rated operational voltage $U_e$</td>
<td>V AC 250</td>
</tr>
<tr>
<td>Safe isolation to EN 50178 between coil and contact</td>
<td>V AC 300</td>
</tr>
<tr>
<td>Safe isolation to EN 50178 between two contacts</td>
<td>V AC 300</td>
</tr>
</tbody>
</table>
## Technical data

<table>
<thead>
<tr>
<th>Making capacity, IEC 60947</th>
<th>MFD-R..</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC-15 250 V AC, 3 A (600 Ops/h)</td>
<td>Switch operations</td>
</tr>
<tr>
<td>DC-13 L/R ≤ 150 ms 24 V DC, 1 A (500 Ops/h)</td>
<td>Switch operations</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AC-15 250 V AC, 3 A (600 Ops/h)</td>
<td>Switch operations</td>
</tr>
<tr>
<td>DC-13 L/R ≤ 150 ms 24 V DC, 1 A (500 Ops/h)</td>
<td>Switch operations</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Filament lamp load</th>
<th>MFD-R..</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000 W at 230/240 V AC</td>
<td>Switch operations</td>
</tr>
<tr>
<td>500 W at 115/120 V AC</td>
<td>Switch operations</td>
</tr>
<tr>
<td>Fluorescent tube with ballast, 10 × 58 W at 230/240 V AC</td>
<td>Switch operations</td>
</tr>
<tr>
<td>Conventional fluorescent tube, compensated, 1 × 58 W at 230/240 V AC</td>
<td>Switch operations</td>
</tr>
<tr>
<td>Conventional fluorescent tube, uncompensated, 10 × 58 W at 230/240 V AC</td>
<td>Switch operations</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Operating frequency, relays</th>
<th>MFD-R..</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mechanical switch operations</td>
<td>Switch operations</td>
</tr>
<tr>
<td>Mechanical switching frequency</td>
<td>Hz</td>
</tr>
<tr>
<td>Resistive lamp load</td>
<td>Hz</td>
</tr>
<tr>
<td>Inductive load</td>
<td>Hz</td>
</tr>
</tbody>
</table>
## UL/CSA

<table>
<thead>
<tr>
<th>MFD-T..</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of outputs</td>
<td>4</td>
</tr>
<tr>
<td>Contacts</td>
<td>Semiconductors</td>
</tr>
<tr>
<td>Rated voltage $U_e$ V DC</td>
<td>24</td>
</tr>
<tr>
<td>Permissible range V DC</td>
<td>20.4 to 28.8</td>
</tr>
<tr>
<td>Residual ripple %</td>
<td>$\leq 5$</td>
</tr>
<tr>
<td>Supply current mA</td>
<td></td>
</tr>
<tr>
<td>On 0 state, typical/maximum</td>
<td>18/32</td>
</tr>
<tr>
<td>On 1 state, typical/maximum</td>
<td>24/44</td>
</tr>
<tr>
<td>Reverse polarity protection</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### Transistor outputs

<table>
<thead>
<tr>
<th>MFD-T..</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of outputs</td>
<td>4</td>
</tr>
<tr>
<td>Contacts</td>
<td>Semiconductors</td>
</tr>
<tr>
<td>Rated voltage $U_e$ V DC</td>
<td>24</td>
</tr>
<tr>
<td>Permissible range V DC</td>
<td>20.4 to 28.8</td>
</tr>
<tr>
<td>Residual ripple %</td>
<td>$\leq 5$</td>
</tr>
<tr>
<td>Supply current mA</td>
<td></td>
</tr>
<tr>
<td>On 0 state, typical/maximum</td>
<td>18/32</td>
</tr>
<tr>
<td>On 1 state, typical/maximum</td>
<td>24/44</td>
</tr>
<tr>
<td>Reverse polarity protection</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### Attention!
If voltage is applied to the outputs when the polarity of the power supply is reversed, this will result in a short circuit.

<table>
<thead>
<tr>
<th>MFD-T..</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Potential isolation to the inputs, supply voltage, PC interface, memory card, NET network, EASY-Link</td>
<td>Yes</td>
</tr>
<tr>
<td>Rated current $I_e$ at state 1, maximum A</td>
<td>0.5</td>
</tr>
</tbody>
</table>
## Technical Data

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Lamp load</strong></td>
<td></td>
</tr>
<tr>
<td>Q1 to Q4 without $R_V$</td>
<td>W 5</td>
</tr>
<tr>
<td>Residual current at state “0” per channel</td>
<td>mA &lt; 0.1</td>
</tr>
<tr>
<td><strong>Max. output voltage</strong></td>
<td></td>
</tr>
<tr>
<td>On 0 state with external load, 10 MΩ</td>
<td>V 2.5</td>
</tr>
<tr>
<td>On 1 state, $I_e = 0.5$ A</td>
<td>$U = U_e - 1$ V</td>
</tr>
<tr>
<td><strong>Short-circuit protection</strong></td>
<td></td>
</tr>
<tr>
<td>Thermal (Q1 to Q4) (evaluation with diagnostics inputs I16, I15)</td>
<td>Yes</td>
</tr>
<tr>
<td>Short-circuit tripping current for $R_a \leq 10$ mΩ (depending on number of active channels and their load)</td>
<td>A $0.7 \leq I_e \leq 2$</td>
</tr>
<tr>
<td>Maximum total short-circuit current</td>
<td>A 8</td>
</tr>
<tr>
<td>Peak short-circuit current</td>
<td>A 16</td>
</tr>
<tr>
<td><strong>Thermal cutout</strong></td>
<td></td>
</tr>
<tr>
<td>Maximum switching frequency with constant resistive load $R_L = 100$ kΩ (depends on program and load)</td>
<td>Switch operations/h 40000</td>
</tr>
<tr>
<td>Parallel connection of outputs with resistive load; inductive load with external suppression circuit (Section “Connecting transistor outputs”, Page 60); combination within a group</td>
<td>Yes</td>
</tr>
<tr>
<td>Group 1: Q1 to Q4</td>
<td></td>
</tr>
<tr>
<td>Maximum number of outputs</td>
<td>4</td>
</tr>
<tr>
<td>Total maximum current</td>
<td>A 2</td>
</tr>
</tbody>
</table>

**Attention!**

Outputs must be actuated simultaneously and for the same time duration.

Status display of the outputs

LCD Status display (if provided)
Inductive load without external suppressor circuit

General explanations:
\( T_{0.95} = \text{time in milliseconds until 95 % of the stationary current is reached} \)

\[ T_{0.95} \approx 3 \times T_{0.65} = 3 \times \frac{L}{R} \]

Utilisation category in groups for
- Q1 to Q4, Q5 to Q8

<table>
<thead>
<tr>
<th>( T_{0.95} ) = 1 ms</th>
<th>Utilisation factor per group ( g = )</th>
<th>0.25</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R = 48 \ \Omega )</td>
<td>Relative duty factor</td>
<td>%</td>
</tr>
<tr>
<td>( L = 16 \ \text{mH} )</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Max. switching frequency</td>
<td>Switch operations/h</td>
</tr>
<tr>
<td></td>
<td>( f = 0.5 \ \text{Hz} )</td>
<td>1500</td>
</tr>
<tr>
<td></td>
<td>Max. duty factor</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DF = 50 %</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DC13</th>
<th>Utilisation factor ( g = )</th>
<th>0.25</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T_{0.95} = 72 \ \text{ms} )</td>
<td>Relative duty factor</td>
<td>%</td>
</tr>
<tr>
<td>( R = 48 \ \Omega )</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>( L = 1.15 \ \text{H} )</td>
<td>Switch operations/h</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Max. switching frequency</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( f = 0.5 \ \text{Hz} )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Max. duty factor</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DF = 50 %</td>
<td></td>
</tr>
</tbody>
</table>
### Technical data

**Other inductive loads:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{0.95}$ = 15 ms</td>
<td>0.25</td>
</tr>
<tr>
<td>$R = 48 , \Omega$</td>
<td>100</td>
</tr>
<tr>
<td>$L = 0.24 , H$</td>
<td>1500</td>
</tr>
<tr>
<td>Relative duty factor %</td>
<td></td>
</tr>
<tr>
<td>Max. switching frequency $f = 0.5 , Hz$</td>
<td></td>
</tr>
<tr>
<td>Max. duty factor $DF = 50 %$</td>
<td></td>
</tr>
<tr>
<td>Inductive loading with external suppressor circuit for each load</td>
<td>Depends on the suppressor circuit</td>
</tr>
</tbody>
</table>

**Utilisation factor $g$ =**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relative duty factor %</td>
<td>1</td>
</tr>
<tr>
<td>Max. switching frequency $f = 0.5 , Hz$</td>
<td>100</td>
</tr>
<tr>
<td>Max. duty factor $DF = 50 %$</td>
<td></td>
</tr>
</tbody>
</table>

**Analog output**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>1</td>
</tr>
<tr>
<td>Electrical isolation</td>
<td></td>
</tr>
<tr>
<td>To power supply</td>
<td>No</td>
</tr>
<tr>
<td>To the digital inputs</td>
<td>No</td>
</tr>
<tr>
<td>To the digital outputs</td>
<td>Yes</td>
</tr>
<tr>
<td>To the easy-NET network</td>
<td>Yes</td>
</tr>
<tr>
<td>Output type</td>
<td>DC voltage</td>
</tr>
<tr>
<td>Signal range $V$ DC</td>
<td>0 to 10</td>
</tr>
<tr>
<td>Output current max. $mA$</td>
<td>10</td>
</tr>
<tr>
<td>Load resistor $k , \Omega$</td>
<td>1</td>
</tr>
<tr>
<td>Short-circuit and overload proof</td>
<td>Yes</td>
</tr>
</tbody>
</table>

For Immediate Delivery call KMParts.com at (866) 595-9616
### easy-NET network

<table>
<thead>
<tr>
<th>Parameter</th>
<th>MFD-RA17, MFD-TA17</th>
<th>MFD-CP8-NT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolution analog V DC</td>
<td>0.01</td>
<td>6/1000</td>
</tr>
<tr>
<td>Resolution digital Bit</td>
<td>10</td>
<td>25/500</td>
</tr>
<tr>
<td>Resolution digital Value</td>
<td>0 to 1023</td>
<td>40/250</td>
</tr>
<tr>
<td>Transient recovery time µs</td>
<td>100</td>
<td>125/125</td>
</tr>
<tr>
<td>Accuracy (–25 to 55 °C), related to the range %</td>
<td>2</td>
<td>300/50</td>
</tr>
<tr>
<td>Accuracy (25 °C), related to the range %</td>
<td>1</td>
<td>700/20</td>
</tr>
<tr>
<td>Conversion time</td>
<td>Each CPU cycle</td>
<td>1000/10</td>
</tr>
</tbody>
</table>

#### Number of stations

- **MFD-RA17, MFD-TA17**: 8
- **MFD-CP8-NT**: 8

#### Bus length/transmission speed

- **MFD-RA17, MFD-TA17**: 6/1000, 25/500, 40/250, 125/125, 300/50, 700/20, 1000/10
- **MFD-CP8-NT**: 6/1000, 25/500, 40/250, 125/125, 300/50, 700/20, 1000/10

#### Electrical isolation

- To power supply, inputs, outputs, EASY-LINK, PC interface, memory module: Yes

#### Bus termination (→ accessories)

- **MFD-RA17, MFD-TA17**: Yes
- **MFD-CP8-NT**: Yes

#### First and last station

- **MFD-RA17, MFD-TA17**: Yes
- **MFD-CP8-NT**: Yes

#### Plug connector (→ accessories)

- **MFD-RA17, MFD-TA17**: 8 poles
- **MFD-CP8-NT**: RJ45
### Technical data

<table>
<thead>
<tr>
<th>Cable cross-sections, with cable lengths and cable resistance/m</th>
<th>MFD-CP8-NT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross-section up to 1000, &lt; 16 mΩ/m</td>
<td>mm² (AWG)</td>
</tr>
<tr>
<td>Cross-section up to 600, &lt; 26 mΩ/m</td>
<td></td>
</tr>
<tr>
<td>Cross-section up to 400, &lt; 40 mΩ/m</td>
<td></td>
</tr>
<tr>
<td>Cross-section up to 250, &lt; 60 mΩ/m</td>
<td>mm² (AWG)</td>
</tr>
<tr>
<td>Cross-section up to 175, &lt; 70 mΩ/m</td>
<td>mm² (AWG)</td>
</tr>
<tr>
<td>Cross-section up to 40, &lt; 140 mΩ/m</td>
<td>mm² (AWG)</td>
</tr>
</tbody>
</table>

1) Bus lengths above 40 m can only be achieved with cables with reinforced cross-section and connection adapter.
## List of the function blocks

<table>
<thead>
<tr>
<th>Element</th>
<th>Meaning of abbreviation</th>
<th>Function block designation</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Analog value comparator</td>
<td>Analog value comparator</td>
<td>150</td>
</tr>
<tr>
<td>AR</td>
<td>Arithmetic</td>
<td>Arithmetic</td>
<td>153</td>
</tr>
<tr>
<td>BC</td>
<td>block compare</td>
<td>Data block</td>
<td>157</td>
</tr>
<tr>
<td>BT</td>
<td>block transfer</td>
<td>Data block transfer</td>
<td>164</td>
</tr>
<tr>
<td>BV</td>
<td>Boolean sequence</td>
<td>Boolean sequence</td>
<td>175</td>
</tr>
<tr>
<td>C</td>
<td>counter</td>
<td>Counters</td>
<td>178</td>
</tr>
<tr>
<td>CF</td>
<td>counter frequency</td>
<td>Frequency counters</td>
<td>185</td>
</tr>
<tr>
<td>CH</td>
<td>counter high-speed</td>
<td>High-speed counters</td>
<td>189</td>
</tr>
<tr>
<td>CI</td>
<td>counter fast incremental value encoder</td>
<td>High-speed incremental encoder</td>
<td>195</td>
</tr>
<tr>
<td>CP</td>
<td>comparators</td>
<td>Comparators</td>
<td>200</td>
</tr>
<tr>
<td>D</td>
<td>display</td>
<td>Text function block</td>
<td>202</td>
</tr>
<tr>
<td>DB</td>
<td>data block</td>
<td>Data function block</td>
<td>203</td>
</tr>
<tr>
<td>DC</td>
<td>DDC controller (direct digital control)</td>
<td>PID controllers</td>
<td>205</td>
</tr>
<tr>
<td>FT</td>
<td>filter</td>
<td>PT1 signal smoothing filter</td>
<td>211</td>
</tr>
<tr>
<td>GT</td>
<td>GET</td>
<td>GET network</td>
<td>205</td>
</tr>
<tr>
<td>HW</td>
<td>hora(lat) week</td>
<td>Seven day timer</td>
<td>216</td>
</tr>
<tr>
<td>HY</td>
<td>hora(lat) year</td>
<td>Twelve month timer</td>
<td>221</td>
</tr>
<tr>
<td>LS</td>
<td>linear scaling</td>
<td>Value scaling</td>
<td>225</td>
</tr>
<tr>
<td>MR</td>
<td>master reset</td>
<td>Master reset</td>
<td>232</td>
</tr>
<tr>
<td>NC</td>
<td>numeric coding</td>
<td>Numerical converters</td>
<td>233</td>
</tr>
<tr>
<td>OT</td>
<td>operating time</td>
<td>Operating hours counter</td>
<td>239</td>
</tr>
<tr>
<td>PT</td>
<td>PUT</td>
<td>PUT network</td>
<td>240</td>
</tr>
<tr>
<td>PW</td>
<td>pulse width modulation</td>
<td>Pulse width modulation</td>
<td>216</td>
</tr>
<tr>
<td>SC</td>
<td>synchronize clocks</td>
<td>Synchronise clock via network</td>
<td>245</td>
</tr>
</tbody>
</table>
### List of the function blocks

<table>
<thead>
<tr>
<th>Element</th>
<th>Meaning of abbreviation</th>
<th>Function block designation</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST</td>
<td>set time</td>
<td>Set cycle time</td>
<td>221</td>
</tr>
<tr>
<td>T</td>
<td>timing relays</td>
<td>Timing relay</td>
<td>249</td>
</tr>
<tr>
<td>VC</td>
<td>value capsuling</td>
<td>Value limitation</td>
<td>262</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Jumps</td>
<td>203</td>
</tr>
</tbody>
</table>

### Function block coils

<table>
<thead>
<tr>
<th>Coil</th>
<th>Meaning of abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C_</td>
<td>count input</td>
<td>Counter input</td>
</tr>
<tr>
<td>D_</td>
<td>direction input</td>
<td>Count up/down indicator</td>
</tr>
<tr>
<td>ED</td>
<td>enable Differential component</td>
<td>Activate differential component</td>
</tr>
<tr>
<td>EI</td>
<td>enable integral component</td>
<td>Activate integral component</td>
</tr>
<tr>
<td>EN</td>
<td>enable</td>
<td>Enable module</td>
</tr>
<tr>
<td>EP</td>
<td>enable proportional component</td>
<td>Activate proportional component</td>
</tr>
<tr>
<td>RE</td>
<td>reset</td>
<td>Reset actual value to zero</td>
</tr>
<tr>
<td>SE</td>
<td>set enable</td>
<td>Set to a predefined value</td>
</tr>
<tr>
<td>ST</td>
<td>stop</td>
<td>STOP block processing</td>
</tr>
<tr>
<td>T_</td>
<td>trigger</td>
<td>Trigger coil</td>
</tr>
</tbody>
</table>
## Function block contacts

<table>
<thead>
<tr>
<th>Contact</th>
<th>Meaning of abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CY</td>
<td>carry</td>
<td>Status “1”, if the value range is exceeded; (carry)</td>
</tr>
<tr>
<td>E1</td>
<td>error 1</td>
<td>Error 1, dependent on function block</td>
</tr>
<tr>
<td>E2</td>
<td>error 2</td>
<td>Error 2, dependent on function block</td>
</tr>
<tr>
<td>E3</td>
<td>error 3</td>
<td>Error 3, dependent on function block</td>
</tr>
<tr>
<td>EQ</td>
<td>equal</td>
<td>Comparison result, status 1 if values equal.</td>
</tr>
<tr>
<td>FB</td>
<td>fall below</td>
<td>Status “1”, if the actual value is less than or equal to the lower setpoint value;</td>
</tr>
<tr>
<td>GT</td>
<td>greater than</td>
<td>Status 1 if the value at I1 &gt; I2;</td>
</tr>
<tr>
<td>LI</td>
<td>limit indicator</td>
<td>Value range manipulated variable exceeded</td>
</tr>
<tr>
<td>LT</td>
<td>less than</td>
<td>Status 1 if the value at I1 &lt; I2;</td>
</tr>
<tr>
<td>OF</td>
<td>overflow</td>
<td>Status “1”, if the actual value is greater than or equal to the upper setpoint value;</td>
</tr>
<tr>
<td>Q1</td>
<td>output (Q1)</td>
<td>Switch output</td>
</tr>
<tr>
<td>QV</td>
<td>output value</td>
<td>Current actual value of the function block (e.g. counter value)</td>
</tr>
<tr>
<td>ZE</td>
<td>zero</td>
<td>Status “1”, if the value of the element input QV is equal to zero;</td>
</tr>
</tbody>
</table>
## List of the function blocks

### Function block inputs (constants, operands)

<table>
<thead>
<tr>
<th>Input</th>
<th>Meaning of abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>Factor 1</td>
<td>Gain factor for I1 (I1 = F1 × Value)</td>
</tr>
<tr>
<td>F2</td>
<td>Factor 2</td>
<td>Gain factor for I2 (I2 = F2 × Value)</td>
</tr>
<tr>
<td>HY</td>
<td>Hysteresis</td>
<td>Switching hysteresis for value I2 (Value HY applies to positive and negative hysteresis.)</td>
</tr>
<tr>
<td>I1</td>
<td>Input 1</td>
<td>1st input word</td>
</tr>
<tr>
<td>I2</td>
<td>Input 2</td>
<td>2nd input word</td>
</tr>
<tr>
<td>KP</td>
<td>Standard</td>
<td>Proportional gain</td>
</tr>
<tr>
<td>ME</td>
<td>Minimum make time</td>
<td>Minimum make time</td>
</tr>
<tr>
<td>MV</td>
<td>Manual value</td>
<td>Manual manipulated variable</td>
</tr>
<tr>
<td>NO</td>
<td>numbers of elements</td>
<td>Number of elements</td>
</tr>
<tr>
<td>OS</td>
<td>Offset</td>
<td>Offset for the value I1</td>
</tr>
<tr>
<td>PD</td>
<td>Period duration</td>
<td>Period duration</td>
</tr>
<tr>
<td>SH</td>
<td>Setpoint high</td>
<td>Upper limit value</td>
</tr>
<tr>
<td>SL</td>
<td>Setpoint low</td>
<td>Lower limit value</td>
</tr>
<tr>
<td>SV</td>
<td>Set value</td>
<td>Defined actual value (Preset)</td>
</tr>
<tr>
<td>TC</td>
<td></td>
<td>Scan time</td>
</tr>
<tr>
<td>TG</td>
<td></td>
<td>Recovery time</td>
</tr>
<tr>
<td>TN</td>
<td>Standard</td>
<td>Rate time</td>
</tr>
<tr>
<td>TV</td>
<td>Standard</td>
<td>Reset time</td>
</tr>
<tr>
<td>X1</td>
<td>X1, interpolation point 1 abscissa</td>
<td>Lower value of source range</td>
</tr>
<tr>
<td>X2</td>
<td>Interpolation point 2 abscissa</td>
<td>Upper value of source range</td>
</tr>
<tr>
<td>Y1</td>
<td>Interpolation point 1 ordinate</td>
<td>Lower value of target range</td>
</tr>
<tr>
<td>Y2</td>
<td>Interpolation point 2 ordinate</td>
<td>Upper value of target range</td>
</tr>
</tbody>
</table>
Function block output (operands)

<table>
<thead>
<tr>
<th>Input</th>
<th>Meaning of abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QV</td>
<td>Output value</td>
<td>Output value</td>
</tr>
</tbody>
</table>

Other operands

<table>
<thead>
<tr>
<th>Other operands</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MB</td>
<td>Marker byte (8-bit value)</td>
</tr>
<tr>
<td>IA</td>
<td>Analog input (if available on device!)</td>
</tr>
<tr>
<td>MW</td>
<td>Marker word (16-bit value)</td>
</tr>
<tr>
<td>QA</td>
<td>Analog output (if available on device!)</td>
</tr>
<tr>
<td>MD</td>
<td>Marker double word (32-bit value)</td>
</tr>
<tr>
<td>NU</td>
<td>Constant (number), value range from –2147483648 to +2147483647</td>
</tr>
</tbody>
</table>
The following table provides an overview of the memory requirement of the easy800 circuit connections, function blocks and their respective constants:

<table>
<thead>
<tr>
<th>Circuit connection</th>
<th>Space requirement per circuit conn./function block</th>
<th>Space requirement per constant on the function block input</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Byte</td>
<td>Byte</td>
</tr>
<tr>
<td>A</td>
<td>68</td>
<td>4</td>
</tr>
<tr>
<td>AR</td>
<td>40</td>
<td>4</td>
</tr>
<tr>
<td>BC</td>
<td>48</td>
<td>4</td>
</tr>
<tr>
<td>BT</td>
<td>48</td>
<td>4</td>
</tr>
<tr>
<td>BV</td>
<td>40</td>
<td>4</td>
</tr>
<tr>
<td>C</td>
<td>52</td>
<td>4</td>
</tr>
<tr>
<td>CF</td>
<td>40</td>
<td>4</td>
</tr>
<tr>
<td>CH</td>
<td>52</td>
<td>4</td>
</tr>
<tr>
<td>CI</td>
<td>52</td>
<td>4</td>
</tr>
<tr>
<td>CP</td>
<td>32</td>
<td>4</td>
</tr>
<tr>
<td>D</td>
<td>160</td>
<td>4</td>
</tr>
<tr>
<td>DC</td>
<td>96</td>
<td>4</td>
</tr>
<tr>
<td>DB</td>
<td>36</td>
<td>4</td>
</tr>
<tr>
<td>FT</td>
<td>56</td>
<td>4</td>
</tr>
<tr>
<td>GT</td>
<td>28</td>
<td>4</td>
</tr>
<tr>
<td>HW</td>
<td>68</td>
<td>4 (per channel)</td>
</tr>
<tr>
<td>HY</td>
<td>68</td>
<td>4 (per channel)</td>
</tr>
<tr>
<td>LS</td>
<td>64</td>
<td>4</td>
</tr>
<tr>
<td>LS</td>
<td>64</td>
<td>4</td>
</tr>
<tr>
<td>MR</td>
<td>20</td>
<td>4</td>
</tr>
<tr>
<td>NC</td>
<td>32</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Space requirement per circuit conn./function block</td>
<td>Space requirement per constant on the function block input</td>
</tr>
<tr>
<td>------</td>
<td>---------------------------------------------------</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>Byte</td>
<td>Byte</td>
</tr>
<tr>
<td>OT</td>
<td>36</td>
<td>4</td>
</tr>
<tr>
<td>PT</td>
<td>36</td>
<td>4</td>
</tr>
<tr>
<td>PW</td>
<td>48</td>
<td>4</td>
</tr>
<tr>
<td>SC</td>
<td>20</td>
<td>--</td>
</tr>
<tr>
<td>ST</td>
<td>24</td>
<td>4</td>
</tr>
<tr>
<td>T</td>
<td>48</td>
<td>4</td>
</tr>
<tr>
<td>VC</td>
<td>40</td>
<td>4</td>
</tr>
<tr>
<td>:</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>
# Index

## A
- Actual values .....................................................149
- Add
  - Circuit connections .........................................84
  - Switching contact ..........................................83

## B
- Break contact ....................................................114
- Inverting .....................................................132
- Bus termination resistor .......................................65
- Button
  - ALT ..............................................................84
  - DEL ..............................................................84
  - OK ........................................................81, 112
- Buttons ..................................................................14

## C
- Cable cross-sections ............................................66
- Cable length ........................................................66
- Cable lengths ......................................................50
- Cable protection ..................................................48
- Circuit connection
  - Number ............................................................126
- Circuit connections
  - Add new ..........................................................84
  - Deleting .......................................................133, 136
  - Entering .........................................................132
  - Insert ............................................................134
  - Position in circuit diagram ..................................126
- Circuit diagram
  - Checking ..........................................................138
  - Circuit connections ..........................................126
  - Coil field .........................................................126
  - Contact fields ..................................................126
  - Deleting ..........................................................89
  - Display ..........................................................82, 126
  - Fast entry ........................................................89
  - Grid ..............................................................82, 126
Index

Internal processing ........................................... 323
Load .......................................................... 341
Operating buttons ........................................... 112
Overview .................................................... 126
Saving ................................................ 339, 341
Testing ................................................. 86, 138
Wiring .................................................. 84, 132
Coil field ........................................................... 126
Coil function
  Contactor ................................................... 144
  Impulse relay .............................................. 145
  Latching relay ............................................ 145
  Overview .................................................. 144
COM-LINK ......................................................... 286
  Activation .................................................. 289
  Baud rate ................................................... 289
  Configuration ............................................. 289
  Data access ........................................ 287, 325
  Operating principle .................................... 290
  Sign of life detection ................................ 291
Connecting the brightness sensor ...................... 55
Connecting transistor outputs ........................... 60
Connection
  20 mA sensor ............................................... 55
  Analog inputs .............................................. 54
  Analog output .............................................. 62
  Brightness sensor ......................................... 55
  Contactors, relays ........................................ 58
  Frequency generator ..................................... 57
  High-speed counters .................................... 57
  Incremental encoder ..................................... 57
  MFD-DC inputs ............................................ 53
  NET network ................................................ 64
  Outputs ........................................................ 58
  Proximity switch ........................................... 53
  Pushbutton actuators, switches .............. 50, 53
  Relay outputs ............................................... 59
  Serial interface .......................................... 69
  Setpoint potentiometer .................................. 55
  Temperature sensor ....................................... 55
  Transistor outputs ........................................ 60
Connection cross-sections
  MFD cables ..................................................45
  Screw terminals ...............................................45
Contact fields ..................................................126
Counter relay ..................................................178, 195
  Parameters ..................................................190, 196
Counter relays
  Parameters ..................................................186
Counters
  High-speed ..................................................57, 189
  High-speed incremental encoder .......................195
Cursor display ..................................................26, 112
Cycle .............................................................323

D
  Data consistency .............................................291
  DC power supply for the MFD .........................47
  Debounce
    Setting ......................................................308
  Delay time
    Input and output ........................................328
  Delay times
    For easy-DA ...............................................329
    For easy-DC ...............................................329
  Deleting retentive actual values ....................318
  Device overview ..........................................12
  Device version ............................................343
  Dimensions, easy ...........................................345

E
  EASY…-AC-.E AC expansion units .......................46
  Expanding .....................................................72
  Expanding inputs .........................................72
  Expanding outputs ........................................72
  Expansion
    Local ..........................................................72
    Remote .........................................................74
<table>
<thead>
<tr>
<th>Index</th>
<th>06/03 AWB2528-1480GB</th>
</tr>
</thead>
</table>

| F     | Fixing brackets ....................................................42  
|       | Frequency counters ................................................185  
|       | Function blocks  
|       | Evaluating ..................................................324  
|       | List .............................................................367  
|       | Function relay  
|       | Counter relay .............................................195  
|       | Counter, high-speed ...................................189  
|       | Counter, high-speed incremental encoder ..195  
|       | Counters ....................................................178  
|       | Example .....................................................263  
|       | Frequency counters ....................................185  
|       | Overview ....................................................148  
|       | Time switch ...............................................216, 221  
|       | Timing relay ...............................................249  
| I     | Impulse relay .....................................................145  
|       | Increasing the input current .................................51  
|       | Incremental encoder ....................................57, 195  
|       | Input contacts ...................................................129  
|       | Inrush current limitation ......................................52  
|       | Interface ............................................................336  
|       | Interference .........................................................50  
|       | Inverting ...........................................................132  
| J     | Jumps ...............................................................229  
| L     | Latching ............................................................144  
|       | Latching relay ..................................................145  
|       | LED display ............................................................17  

For Immediate Delivery call KMParts.com at (866) 595-9616
# Index

<table>
<thead>
<tr>
<th>M</th>
<th>Main menu</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Overview ......................................................19</td>
</tr>
<tr>
<td></td>
<td>Selection ......................................................16</td>
</tr>
<tr>
<td>Make contact .....................................................114</td>
<td></td>
</tr>
<tr>
<td>Inverting .....................................................132</td>
<td></td>
</tr>
<tr>
<td>Marker relay .....................................................326</td>
<td></td>
</tr>
<tr>
<td>Memory card</td>
<td></td>
</tr>
<tr>
<td>Deleting .....................................................340</td>
<td></td>
</tr>
<tr>
<td>Insertion .....................................................338</td>
<td></td>
</tr>
<tr>
<td>Memory management</td>
<td></td>
</tr>
<tr>
<td>Mask memory ............................................327</td>
<td></td>
</tr>
<tr>
<td>Program memory ........................................327</td>
<td></td>
</tr>
<tr>
<td>RAM ...........................................................327</td>
<td></td>
</tr>
<tr>
<td>Menu</td>
<td></td>
</tr>
<tr>
<td>Changing level .............................................81</td>
<td></td>
</tr>
<tr>
<td>Changing the language ..................302</td>
<td></td>
</tr>
<tr>
<td>Language setting .................................77, 272</td>
<td></td>
</tr>
<tr>
<td>Operator guidance ........................................15</td>
<td></td>
</tr>
<tr>
<td>Message</td>
<td></td>
</tr>
<tr>
<td>INVALID PROG ...................................340, 341</td>
<td></td>
</tr>
<tr>
<td>MFD</td>
<td></td>
</tr>
<tr>
<td>Evaluating the circuit diagram ........323</td>
<td></td>
</tr>
<tr>
<td>Operating modes .................................78</td>
<td></td>
</tr>
<tr>
<td>Overview ......................................................12</td>
<td></td>
</tr>
<tr>
<td>Mode</td>
<td></td>
</tr>
<tr>
<td>Changing .....................................................86</td>
<td></td>
</tr>
<tr>
<td>Mounting</td>
<td></td>
</tr>
<tr>
<td>Display/operating unit ......................33</td>
<td></td>
</tr>
<tr>
<td>Inputs/outputs onto the power supply/CPU module ......................38</td>
<td></td>
</tr>
<tr>
<td>On top-hat rail ......................40</td>
<td></td>
</tr>
<tr>
<td>Power supply/CPU module ......................36</td>
<td></td>
</tr>
<tr>
<td>Protective cover .......................................30</td>
<td></td>
</tr>
<tr>
<td>Protective membrane ......................28</td>
<td></td>
</tr>
<tr>
<td>Screw fixing ..............................................42</td>
<td></td>
</tr>
<tr>
<td>Top-hat rail ..................................................36</td>
<td></td>
</tr>
</tbody>
</table>
Index

N  Network
   Addressing ............................................... 273
   Automatic change of the RUN and
   STOP mode ............................................... 279
   Cables ..................................................... 67
   Changing the write repetition rate ............ 278
   Configuration of the COM-LINK ............... 289
   Configuring input/output devices ............. 280
   Connecting the NET network .................... 64
   Data accesses via COM-LINK .................... 287
   Introduction to COM-LINK ....................... 286
   Station message types ......................... 281
   Station signs of life ............................... 282
   Topology .................................................. 273, 286
   Transfer behaviour .................................. 281
   Transmission security ............................. 285
   Network cables ............................................ 67

O  Operating principles ........................................ 14
   Output relay ............................................. 129
   Overload .................................................. 62, 331
   Monitoring with EASY...D.-T. .................... 331

P  P buttons ...................................................... 137
   Activating and deactivating ....................... 309
   Parameter display
      Counter relay ........................................ 190, 196
      Timing relay ......................................... 186
   Parameters
      Change .................................................. 303
      Display .................................................. 303
      Inhibit access ....................................... 303
   Password
      Activation ............................................. 298
      Change .................................................. 300
      Deleting ............................................... 300
      Protection removal .................................. 301
      Setup .................................................. 296
Index

Pause time ........................................................278
Physical location ..................................................68
Power failure .......................................................78
Power flow display ...............................................86, 87, 138
Program ............................................................323
  Cycle ..................................................................323
  Load ..................................................................336
  Saving ..................................................................336

R
Relay coil
  Change .............................................................129
  Coil function .....................................................130, 144
  Deleting ..........................................................132
  Entering ..........................................................85, 129
Relays
  Connecting the outputs ......................................59
  Overview ........................................................119, 122, 126
Removing
  Inputs/outputs ..................................................39
  Power supply/CPU module ..................................39
Reset .................................................................145
Retention
  Setting behaviour ...............................................317
  Transferring behaviour .......................................319
Retention requirements
  Permitted easy models .......................................317
  Retentive behaviour ..........................................319
  On circuit diagram transfer .................................319
  Setting ............................................................317
RUN, start behaviour ...........................................78
RUN/STOP changeover ...........................................86
<table>
<thead>
<tr>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>Screw mounting .................................................. 42</td>
</tr>
<tr>
<td>Sealing the protective cover ........................................ 32</td>
</tr>
<tr>
<td>Sensor (20 mA) connection ........................................ 55</td>
</tr>
<tr>
<td>Serial interface .................................................... 69</td>
</tr>
<tr>
<td>Set ................................................................. 145</td>
</tr>
<tr>
<td>Setpoint potentiometer ........................................... 55</td>
</tr>
<tr>
<td>Setpoints ........................................................... 64, 149, 304</td>
</tr>
<tr>
<td>Setting LCD backlight ............................................. 314</td>
</tr>
<tr>
<td>Setting LCD contrast ................................................ 314</td>
</tr>
<tr>
<td>Setting the LCD .................................................... 314</td>
</tr>
<tr>
<td>Setting the time ................................................... 305</td>
</tr>
<tr>
<td>Short-circuit ........................................................ 62, 331</td>
</tr>
<tr>
<td>Monitoring with EASY..-D.-T. .................................... 331</td>
</tr>
<tr>
<td>Startup behaviour .................................................. 310, 312</td>
</tr>
<tr>
<td>After the circuit diagram is deleted ........................... 311</td>
</tr>
<tr>
<td>Basic setting ....................................................... 311, 313, 314, 315</td>
</tr>
<tr>
<td>Possible faults ..................................................... 312</td>
</tr>
<tr>
<td>Setting ............................................................... 310</td>
</tr>
<tr>
<td>With upload/download to card or PC ............................ 311</td>
</tr>
<tr>
<td>Station</td>
</tr>
<tr>
<td>Active ............................................................... 286</td>
</tr>
<tr>
<td>Remote ............................................................... 286</td>
</tr>
<tr>
<td>Status display .......................................................... 16</td>
</tr>
<tr>
<td>Status image register .............................................. 324</td>
</tr>
<tr>
<td>Suppressing contact bounce ....................................... 328</td>
</tr>
<tr>
<td>Switching contact ................................................... 132</td>
</tr>
<tr>
<td>Change ................................................................. 129</td>
</tr>
<tr>
<td>Contact name ........................................................ 129</td>
</tr>
<tr>
<td>Contact number ..................................................... 129</td>
</tr>
<tr>
<td>Cursor buttons ...................................................... 137</td>
</tr>
<tr>
<td>Deleting ............................................................. 132</td>
</tr>
<tr>
<td>Entering ............................................................. 83, 129</td>
</tr>
<tr>
<td>Invert ................................................................. 84</td>
</tr>
<tr>
<td>Overview ............................................................. 113</td>
</tr>
<tr>
<td>System menu</td>
</tr>
<tr>
<td>Selection .............................................................. 16</td>
</tr>
</tbody>
</table>
Index

T
Temperature sensor connection ........................... 55
Terminal mode ................................................. 78
Tightening torque ............................................ 45
Time switch
  24 hour switching ........................................... 221
  Overnight switching ....................................... 219
  Power failure ............................................... 220
  Switching at the weekend .................................. 219
  Switching on working days ................................. 218
  Time overlaps ............................................... 220
Timing relay
  On-delayed .................................................. 244, 253
  Operating modes ........................................... 250
  Wiring ........................................................ 249
Tool for cage clamp terminals ............................... 45
Tool for easy expansion unit ................................ 45
Two-wire proximity switches ................................ 51

V
Visualization data
  Load ................................................................ 325

W
Weekday setting .................................................. 305
Wiring
  Backwards ...................................................... 326
  Deleting ......................................................... 84
  Entering ........................................................ 84
  Relay coils ..................................................... 144
  Rules .......................................................... 144