Manual

11/11 MN04902001Z-EN
replaces AWB2528-1423GB

Control Relay
easy800

Powering Business Worldwide

All brand and product names are trademarks or registered trademarks of the owner concerned.

Emergency On Call Service
Please call your local representative:
http://www.eaton.com/moeller/aftersales
or
Hotline of the After Sales Service:
+49 (0) 180 5 223822 (de, en)
AfterSalesEGBonn@eaton.com

Original Operating Instructions
The German-language edition of this document is the original operating manual.

Translation of the original operating manual
All editions of this document other than those in German language are translations of the original German manual.

1st published 2002, edition date 05/02
2nd edition 2002, edition date11/02
3rd edition 2003, edition date 06/03
4th edition 2004, edition date 08/04
5th edition 2010, edition date 05/10
6th edition 2011, edition date 11/11
See revision protocol in the "About this manual" chapter

© 2002 by Eaton Industries GmbH, 53105 Bonn

Production: René Wiegand
Translation: globaldocs GmbH

All rights reserved, including those of the translation.

No part of this manual may be reproduced in any form (printed, photocopy, microfilm or any other process) or processed, duplicated or distributed by means of electronic systems without written permission of Eaton Industries GmbH, Bonn.

Subject to alteration without notice.

Danger!
Dangerous electrical voltage!

Before commencing the installation

- Disconnect the power supply of the device.
- Ensure that devices cannot be accidentally restarted.
- Verify isolation from the supply.
- Earth and short circuit.
- Cover or enclose neighbouring units that are live.
- Follow the engineering instructions (AWA/IL) 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

0 About This Manual

This manual describes the installation, commissioning and programming of the control relay easy800.

Specialist electrical training is needed for commissioning and creating circuit diagrams.

The current edition of this manual in other languages can be obtained from the Internet:

www.eaton.com/moeller/support

by entering easy800 in the Quick Search box.

You can find further information on SmartWire-DT and the SmartWire-DT stations in the following manuals:

- SmartWire-DT system MN05006002Z-EN
- SmartWire-DT station MN05006001Z-EN

0.1 List of revisions

This edition was fully revised. There is a major difference to previous editions: the new EASY802-DC-SWD and EASY806-DC-SWD devices have been included in this description. The following table shows the essential changes.

Publication date	Page	Key word	New	Change	omitted
11/11	15	„easy800 basic devices“ easy802-DC-SWD easy806-DC-SWD	✓		
	67	„Install SmartWire-DT“	✓		
	76	„Defining easy800-SWD inputs/outputs“	✓		
	98	„Putting the SmartWire-DT into operation“	✓		
	124	„Program transfer from and to the memory card“	✓		
	129	„Function block list“		✓	
	282	„SmartWire-DT“	✓		
	308	„Retentive data storage“		✓	
	313	„Summer/winter time“		✓	
	328	„Loadable operating system“	✓		
	329	„Diagnostics possibilities“		✓	
	333	„What Happens If …?“		✓	
	336	„Functionality of the SWD line faulty“	✓		
	360	„SmartWire-DT“	✓		
	363	„Compatibility of SWD stations“	✓		

0.2 Target group

This operator manual is aimed at persons who wish to

- develop an application with easy800.
- test or commission an application developed with easy800.
- maintain an application with easy800.
- diagnose faults in an application with easy800.

A easy800 may only be fitted and connected by a qualified electrician or a person who is familiar with electrical installations.

0.3 Exclusion of liability

All information in this operator manual was provided by us to the best of our knowledge and belief and in accordance with the current state-of-the-art. However, this does not exclude the possibility of inaccuracies so that we cannot accept any liability for the accuracy and completeness of the information. This information in particular does not contain any guarantee for the provision of specific properties.

The devices described must only be installed and operated in accordance with the content of this operator manual and the installation instructions provided with the device. Installation, commissioning, operation, maintenance and refitting of the devices must only be carried out by qualified persons. The devices must only be used in the areas recommended and only in conjunction with third-party devices and components that have been approved by us. Their use is allowed fundamentally only in technically faultless condition. Fault-free and safe operation of the system requires proper transport, storage, mounting and commissioning as well as careful operation and maintenance.If the following safety instructions are not observed, particularly with regard to commissioning and maintenance of the devices by insufficiently qualified personnel and/or in the event of improper use of the devices, any hazards caused by the devices cannot be excluded. We cannot accept any liability for any resulting injury or damage.

The use of such sample programs and of the easySoft-Pro programming software are subject to the following safety instructions and operating guidelines:

1. The program examples provided were created to the best of our knowledge and belief and in accordance with the current state-of-the-art. The program examples provided were created to the best of our knowledge and belief and in accordance with the current state-of-the-art. However, errors cannot be totally excluded, and the example programs do not cover all function blocks and applications that are available for the easy800 devices.

2. A specialist knowledge of electrical engineering is needed for the programming and commissioning of easy800 devices. Plant sections and persons are at risk if a easy800 device is incorrectly connected or configured and active components such as motors or pressure cylinders are controlled.

3. When using the provided sample programs and generating a program with easySoft, the user has the sole responsibility to observe the following:

- All relevant regulations regarding the preparation of a circuit diagram for easy800 devices according to the updated user manuals of easy800 devices.
- b) All relevant regulations, directives, rules and standards of occupational safety and accident prevention regarding the intended preparation of circuit diagrams, initial start-up and the operation of easy800 devices, particularly those issued by Employers' Liability Insurance Associations (Berufsgenossenschaft).
- Acknowledged rule of technology and state of science.
- All other general due diligence regarding the prevention of damages to life and physical condition of persons as well as material damage.

4. The manufacturer cannot accept any liability for any damages that are caused by customers not using the example programs provided in accordance with the conditions of use specified under points 1 to 3.

0.4 Device designations and abbreviations

The following terms are used for the device types if the description applies to all these types:

- easy600 for
  - EASY618-.C-RE
  - EASY620-DC-TE
- easy800 for
  - easy800 standard
  - Easy800-SWD
- easy800 standard for
  - EASY819-…,
  - EASY820-…,
  - EASY821-…,
  - EASY822-…
- Easy800-SWD for
  - EASY802-DC-SWD
  - EASY806-DC-SWD
- MFD for
  - the programmable devices, consisting of power supply/CPU module (MFD-..-CP8…/CP10…), display/operator unit (MFD-80…) and optional I/O module.
- MFD-CP8… for
  - the MFD-CP8… and MFD-AC-CP8… power supply/CPU modules including the easyNet-enabled modules.
- MFD-CP10… for
  - the MFD-CP10-ME power supply/CPU modules and the easyNet-enabled MFD-CP10-NT.
- MFD-…-NT for
  - easyNet-enabled power supply/CPU modules.
- MFD-CP4… for
  - MFD-CP4… and MFD-AC-CP4… power supply unit/ communication modules
- NET for easyNet
- SWD for SmartWire-DT

0.5 Writing conventions

The symbols used in this manual have the following meanings:

▶ indicates actions to be taken.

⚠	**DANGER** Warns of hazardous situations that result in serious injury or death.

⚠	**WARNING** Warns of the possibility of hazardous situations that could result in serious injury or even death.

	CAUTION Warns about the possibility of material damage.

→ Draws your attention to interesting tips and supplementary information.

For greater clarity, the name of the current chapter and the name of the current section are shown at the top of each page.

1 easy800

1.1 Proper use

An easy800 device is a programmable switching and controller device that is used to replace relay and contactor controls.

Power supply and signal terminals must be protected against accidental contact and covered.

The easy800 device may only be operated if it has been correctly fitted and connected by qualified electrical specialists. The installation must comply with regulations for electromagnetic compatibility (EMC).

> ⚠ **DANGER**
>
> The power up of the easy800 device must not cause any hazards arising from activated devices, such as unexpected motor startups or power ups.

The easy800 device should not be used as a substitute for safety-related controls such as burner controls, emergency-stop or two-hand safety controls.

1.2 Overview of functions

An easy800 device is an electronic control relay with:

- Logic functions
- Timing relay and counter functions
- Time switch functions
- Arithmetic functions
- PID controllers
- Integrated display and operating elements.

An easy800 combines the functions of a control relay and an input device in one unit.

The integrated easyNet network makes it possible to connect up to eight NET stations to a PLC. All devices with a NET terminal connection can be used as NET stations. At present these devices consist of easy800, MFD, easySafety devices or type XC200/EC4-200 PLCs.

Each Net station can process its own circuit diagram.

This allows the design of systems using high-speed controllers with decentralized intelligence.

With easy you can create a circuit diagram in ladder diagram. The circuit diagram entry can be carried out directly on the device using the operator buttons or on your PC using the easySoft-Pro programming software.

For example, you can:

- Connect N/O and N/C contacts in series and in parallel
- Connect output relays and markers.
- Define outputs as coils, impulse relays, rising or falling edge-triggered relays or as latching relays.

The function blocks provide you with several functions such as arithmetic functions, value comparison functions or up/down counting. All function blocks provided are listed in alphabetical order in the Appendix on → Chapter 12 "Appendix", page 369. The cross-references provide links to detailed descriptions of these function blocks.

If you wish to wire an easy800 via your PC, i.e. create a circuit diagram, use the easySoft-Pro programming software.

easySoft-Pro also allows you to:

- test your circuit diagram by simulating the power flow (offline test).
- transfer the circuit diagram to a connected and operational easy800 device.
- after the transfer, follow the power flow and view the operand states after the transfer during operation (online test)
- print out and thus fully document your circuit diagram in different formats (e.g. DIN; ANSI or easy format).

You enter your password to protect your know-how.

1.3 Versions

1.3.1 easy800 basic devices

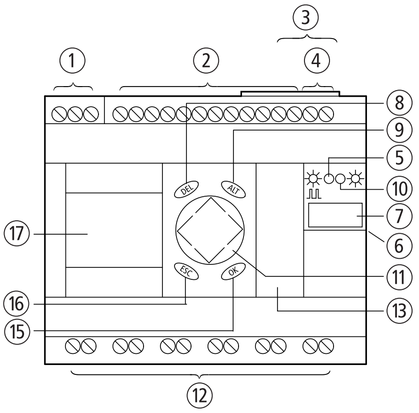


Figure 1: easy800 standard

① Power supply, → Page 26
② Inputs, → Page 30
③ NET-Connection, → Page 53
④ Analog output 0 to 10 V DC/10 Bit, → Page 50
⑤ Power/RUN LED, → Page 71
⑥ easyLink connection, → Page 22
⑦ Place for device identification sticker
⑧ DEL pushbutton, → Page 17
⑨ ALT pushbutton, → Page 17
⑩ NET LED, → Page 71
⑪ Cursor buttons,
⑫ Outputs, → Page 44
⑬ Serial interface (with cover) → Page 56
⑭ OK Button
⑮ ESC button
⑯ LCD display (if provided) → Page 17

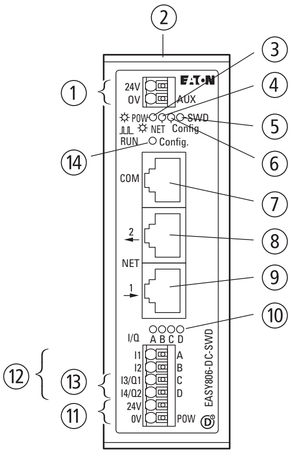
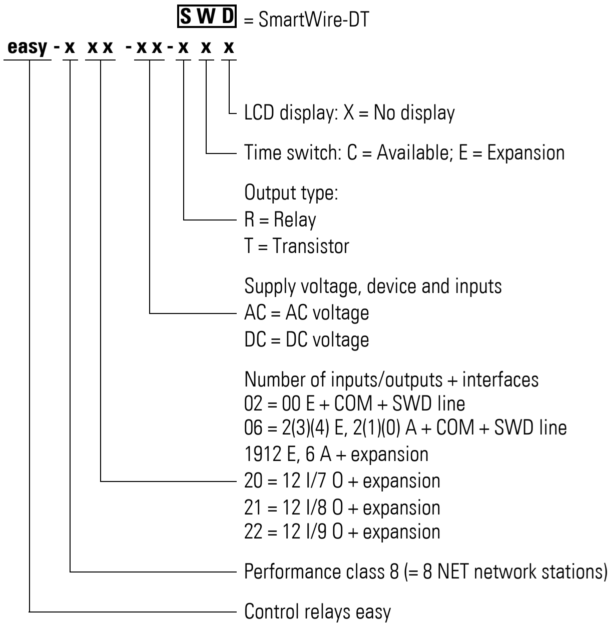


Figure 2: easy800-SWD

① Power supply, 24 V DC AUX
② SWD-OUT
③ POW-LED
④ NET LED
⑤ SWD-LED
⑥ Config-LED
⑦ Serial interface (COM)
⑧ NET OUT[1)]
⑨ NET IN[1)]
⑩ LED (I1 - I4)[1)]/ (Q1, Q2)[1)] → Page 71
⑪ Power supply, 24 V DC POW
⑫ Inputs[1)]
⑬ Outputs[1)]
⑭ Configuration button → Page 17

1) not available with EASY802-DC-SWD

1.3.2 Key to part numbers easy800



LCD display: X = No display

Time switch: C = Available; E = Expansion

Output type:
R = Relay
T = Transistor

Supply voltage, device and inputs
AC = AC voltage
DC = DC voltage

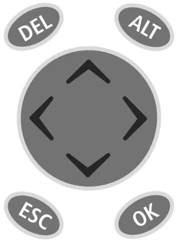
Number of inputs/outputs + interfaces
02 = 00 E + COM + SWD line
06 = 2(3)(4) E, 2(1)(0) A + COM + SWD line
1912 E, 6 A + expansion
20 = 12 I/7 O + expansion
21 = 12 I/8 O + expansion
22 = 12 I/9 O + expansion

Performance class 8 (= 8 NET network stations)

Control relays easy

1.4 Operation of the menu selection and value entry

1.4.1 Key Pad



DEL: Delete object in circuit diagram

ALT: Special functions in circuit diagram, status display

Cursor buttons < > ∧ ∨:
Move cursor
Select menu items
Change numbers, contacts and values

OK: Next menu level, Save your entry

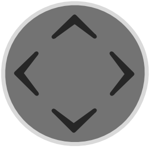
ESC: Back, Cancel

1.4.2 Selecting menus and entering values

DEL and ALT Calling the System menu

OK Move to next menu level Call menu item Activate, change, store entries

ESC Move to previous menu level
Cancel entries since last OK

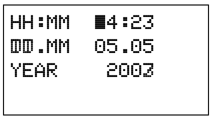


∧ Change menu item
∨ Change value
< > Change place

P spring-return:

<	Input P01	∧	Input P02
>	Input P03	∨	Input P04

1.4.3 Cursor display

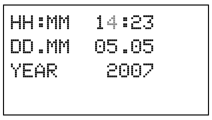


```
HH:MM  █4:23
DD.MM  05.05
YEAR    2007
```

The cursor flashes.

Full cursor █/:

- Move cursor with < >
- in circuit diagram also with ∧ ∨



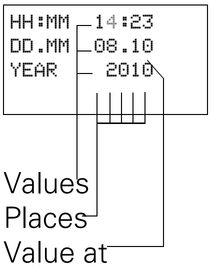
```
HH:MM  14:23
DD.MM  05.05
YEAR    2007
```

Value H/H

- Change position with < >
- Change values with ∧ ∨

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

1.4.4 Entering of values



```
HH:MM  14:23
DD.MM  08.10
YEAR    2010
```

Select value ∧ ∨

Select digit < >

Change value at digit ∧ ∨

OK Store settings

ESC Retain previous value

2 Installation

easy800 devices must only be installed and wired up by qualified electricians or other persons familiar with the mounting of electrical equipment.

DANGER OF ELECTRIC SHOCK!

All installation work must be carried out with the entire installation in a de-energized state.
Always follow the safety rules:

- Switch off and isolate.
- Verify isolation from the supply.
- Secure against retriggering.
- Short-circuit and ground.
- Cover adjacent live parts.

The easy800 devices are installed in the following order:

- "Mounting easy800", Page 20.
- "Mounting the expansion unit", Page 22 (optional).
- "NET easynetwork", Page 53 (optional).
- "Install SmartWire-DT"(optional), → Page 67.
- "Connecting the serial interface", Page 60 (optional).
- "Connecting the inputs", Page 30 (optional).
- "Connecting outputs", Page 44 (optional).
- "Connecting the power supply", Page 26.

2.1 Mounting easy800

Install a easy800 device in an enclosure, switch cabinet or distribution board so that the power supply and terminal connections cannot be touched accidentally during operation.

Snap the device onto an IEC/EN 60715 top-hat rail or fix it in place using fixing brackets. The easy800 device can be mounted either vertically or horizontally.

→ When using the easy800 device with expansion units, connect the expansion unit concerned before mounting (→ section 2.2, "Mounting the expansion unit", page 22).

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

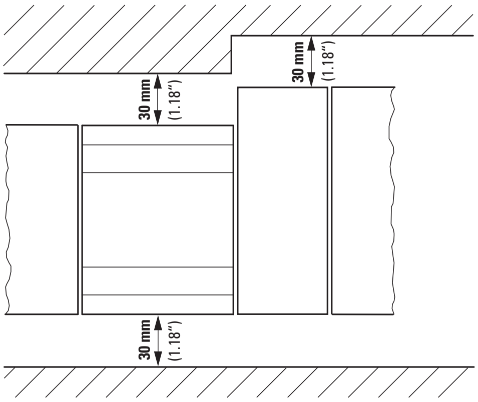


Figure 3: Vertical mounting easy800 standard and easy800-SWD

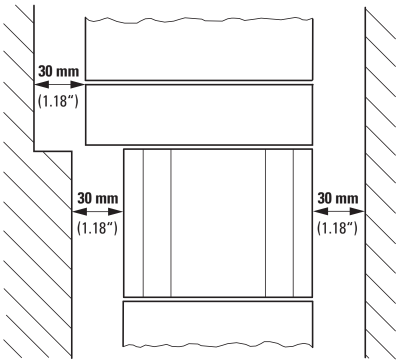
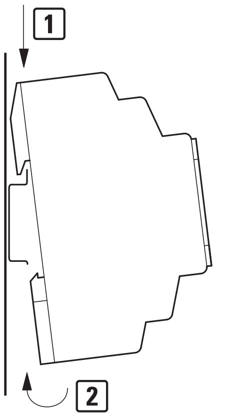


Figure 4: Horizontal mounting easy800 standard and easy800-SWD

easy800 mounting on a top-hat rail



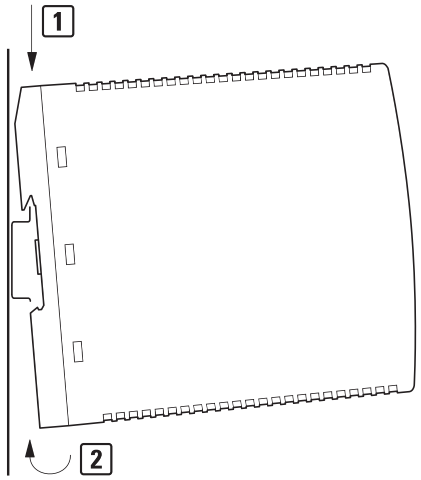


Figure 5: easy800 standard

Figure 6: easy800-SWD

▶ Place the easy800 device diagonally on the upper lip of the top-hat rail. 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 easy800 device 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.

easy800 screw fixing

Fixing brackets that can be inserted on the rear of the easy800 device are required for screw fixing. The fixing brackets are available as an accessory.

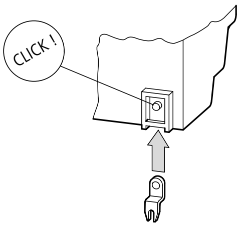


Figure 7: Inserting a fixing bracket

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

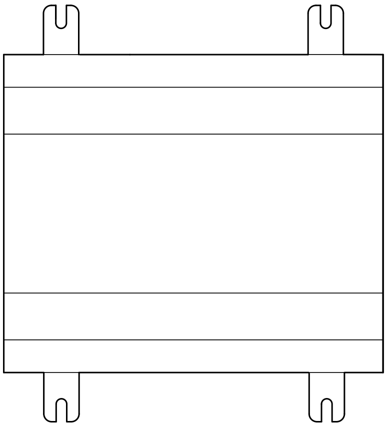


Figure 8: Screw fixingeasy800 standard

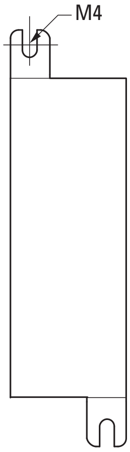


Figure 9: Screw fixing easy800-SWD

2.2 Mounting the expansion unit

To increase the number of inputs/outputs, you can connect an expansion unit via the easyLink connection of the easy800 standard. easy800-SWD devices are not provided with an easyLink terminal.

Fit the expansion units in exactly the same way as a basic device on the top-hat rail or by screwing on fixing brackets.

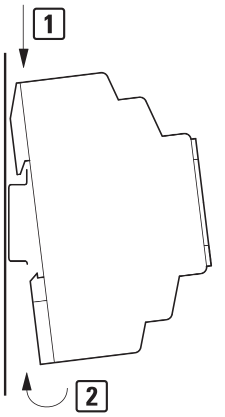


Figure 10:Top-hat rail mounting

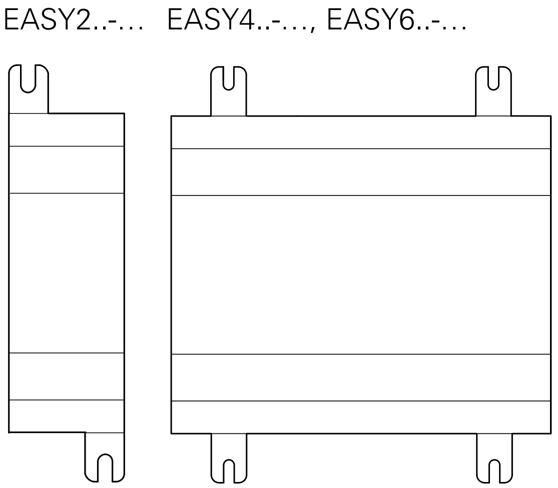


Figure 11:Screw fixing

Locally and remotely connected expansion units have a different connection.

Local expansion module

Local expansion modules are connected directly next to the easy800 standard with easyLink connection.

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

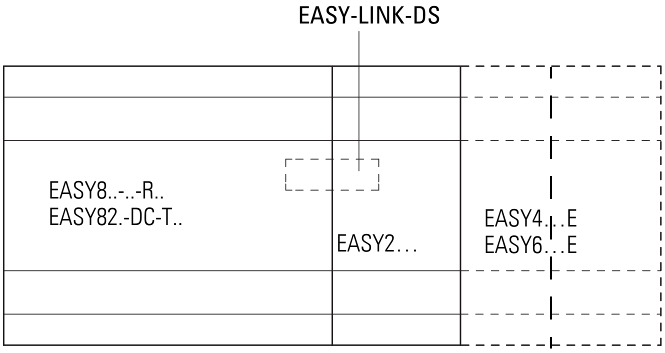


Figure 12:Connecting local expansion module with easy800 standard

The following electrical isolation is implemented between the easy800 standard and the expansion unit (isolation always in the local terminal of the expansion unit):

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

→ easy800 standard and expansion unit can be provided with different DC power supplies.

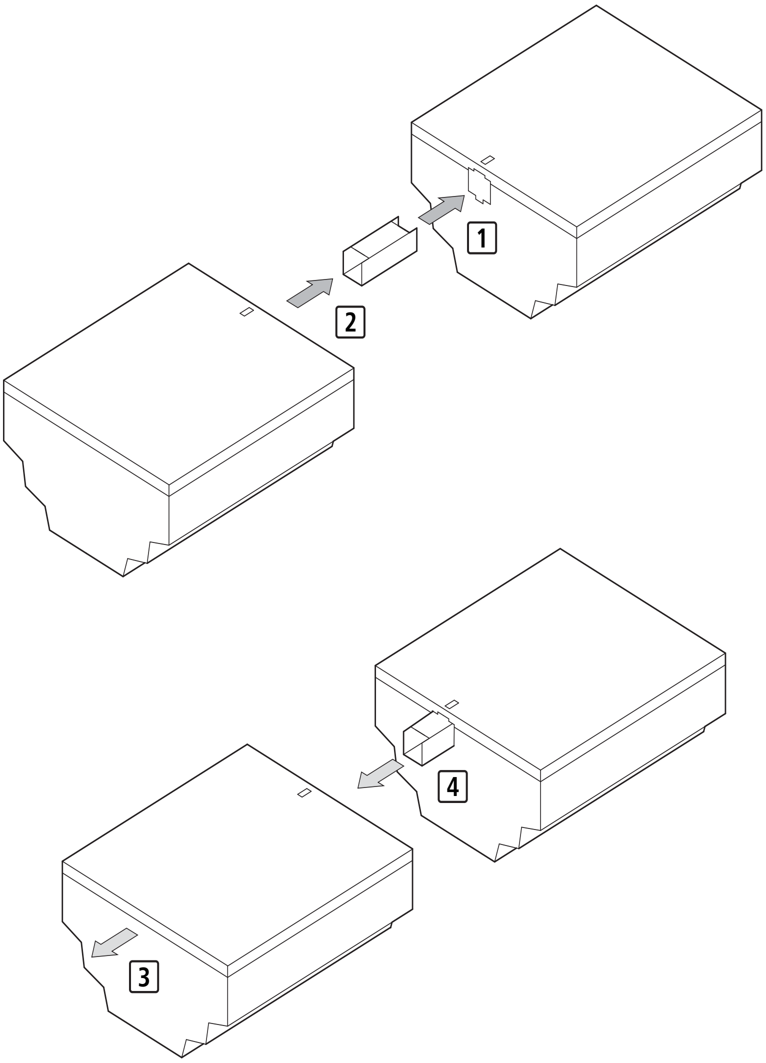


Figure 13:Mounting 1 + 2 and removing 3 + 4 the local expansion module on the easy800 device

Specific system information is shown in the status display of the easy800 standard (e.g. DC, AC or GW) depending on the expansion unit that is connected and ready for operation.

Remote expansion module

With remote expansion, you connect the expansion unit via a connecting cable to the EASY200-EASY coupling unit, which itself is fitted on the easy800 standard directly via the easyLink terminal. A two-wire or multi-core connecting cable up to 30 m in length can be used.

DANGER

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 U_e = 300/500 V AC, is normally sufficient.

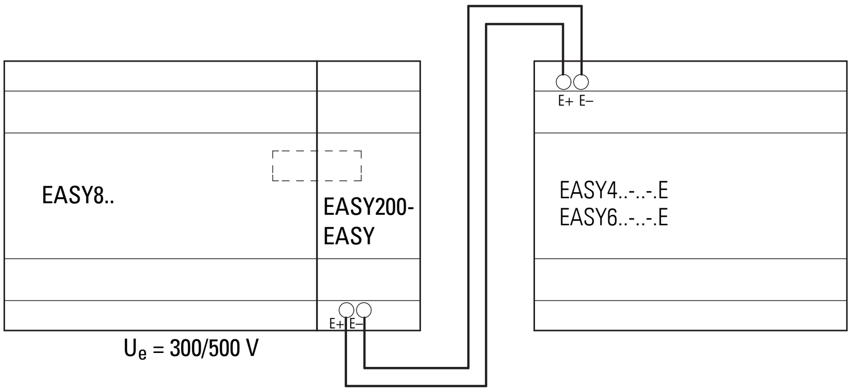


Figure 14:Connecting remote expansion module

→ 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–.

2.3 Terminations

Tools

easy800 standard: Slot-head screwdriver, width 3.5 x 0.6 mm, tightening torque for screw terminals 0.6 Nm.

easy800-SWD: the push-in terminals eliminate the need for a tool.

Cable cross-sections

	Terminal capacities at mm²	
	easy800 standard	**easy800-SWD**
Solid	0.2 to 4 (AWG 22 -12)	0.2 to 1.5 (AWG 24 -16)
flexible with ferrule	0.2 to 2.5	0.25 to 1.5

How to connect the easyNet network → section 2.7, “NET easynetwork”, page 53.

2.4 Connecting the power supply

Cable protection

Connect cable protection (F1) of at least 1 A (slow) with the AC and DC variant of the easy800 standard device and with the expansion unit. Connect on easy800-SWD a cable protection (F1) rated for at least 3 A (slow).

System test

The devices run a system test after the supply voltage has been switched on. On easy800 standard, this system test lasts 1 s. After this time RUN or STOP mode are activated - depending on the device and preset. On easy800-SWD the system test lasts at least 5.6 s. easy800-SWD then automatically switches to RUN mode.

The length of the system test varies:

Device model	Time in s
easy800 standard	1
easy800-SWD	5.6 + n 0.12

n = Number of stations on the SWD line (1 - 99)

CAUTION

When the basic devices and expansion units are switched on, they behave like a capacitor, so that an inrush current higher than the rated input current is present.
Take this inrush current into account when designing the electrical equipment by using slow fuses and suitable switches.
Never use reed relay contacts to switch the supply voltage as these may burn or stick.
When designing the DC power supply unit, bear the inrush current in mind.

→ The relevant connection data for the different device types → section 12.3.3, “Power supply”, page 346.

AC-easy800 standard

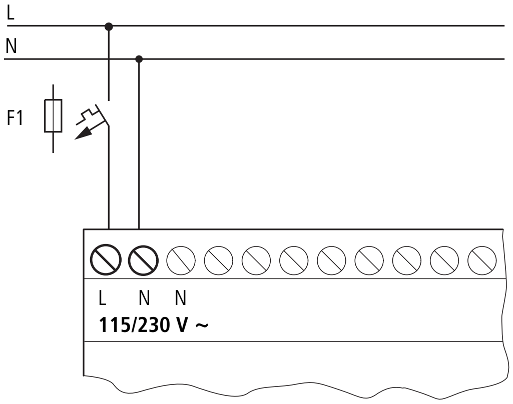


Figure 15:Power supply on easy800 standard AC version

DANGER

Ensure that the L and N conductor are not reversed.

AC expansion unit EASY618-AC-RE

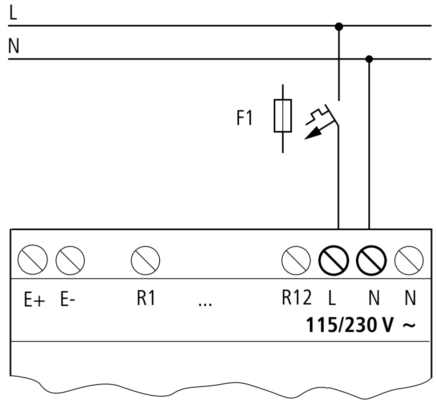


Figure 16:Power supply on the AC expansion unit

DC basic deviceeasy800 standard

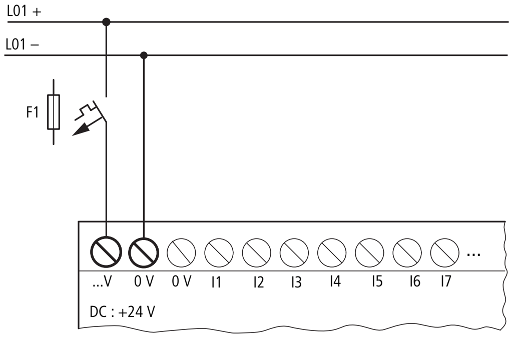


Figure 17:Power supply on the DC basic device easy800 standard

easy800 SWD

The device and SmartWire-DT power supply is implemented via the POW terminals. On easy800-SWD provide cable protection (F1) rated for at least 3 A (slow).

Loads such as contactors, circuit-breakers or motor-protective circuit-breakers can be fed with power via the AUX terminals. For AUX use a maximum cable protection (F2) of 3 A to meet IEC requirements or max. 2 A to meet UL requirements.

WARNING

In safety-relevant applications the power supply providing power to the POW must be a PELV (protective extra low voltage) power supply unit.

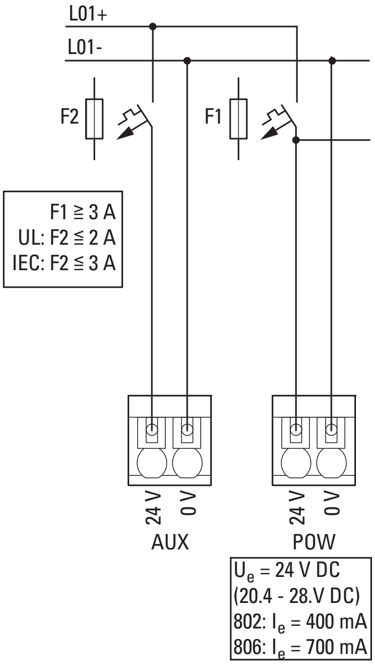


Figure 18:Power supply at easy800-SWD

→ easy800 standard DC version, the expansion units as well as easy800-SWD are protected against reverse polarity.

DC expansion unit EASY…-DC-.E

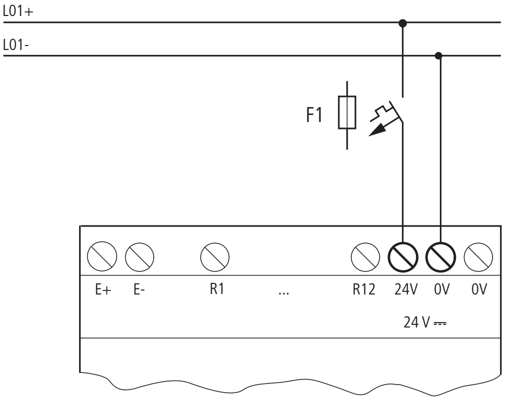


Figure 19:Power supply on the DC expansion unit

2.5 Connecting the inputs

The inputs of the easy800 devices switch electronically. Once you have connected a contact via an input terminal, you can reuse it as a contact in your circuit diagram as often as you like.

Connect the contacts, for example pushbuttons or switches, to the input terminals of the easy800 device. This does not apply to EASY802-DC-SWD as it does not have any input terminals. easy800-SWD devices primarily use SmartWire-DT for wiring.

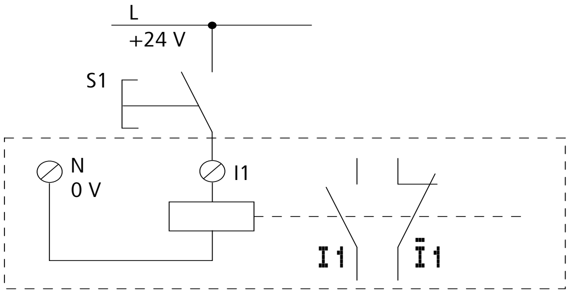


Figure 20:Connecting input, in this case I1.

See also:

- “Connecting the digital inputs“, Page 31.
- “Connecting the DC inputs of the easy800 standard“, Page 35.
- “Connecting analog inputs“, Page 40.

When using special function blocks, certain inputs are permanently assigned to these function blocks and cannot be used for any other purpose. The relevant function block description describes which inputs are affected.

Example: With the “high-speed counter” function block, the device inputs I1 - I4 are connected directly on one function block.

You can increase the number of inputs of the easy800 standard by using an expansion unit. The inputs of the EASY4..-DC-.E and EASY6..-DC-.E digital expansion units are detected by the easy800 standard automatically - without any configuration.

The universal EASY4..-DC-ME expansion units allow you to process both digital and analog values. The digital inputs of these expansion units are detected by the easy800 standard automatically - without any configuration. The analog inputs/outputs must normally be configured.

See also:

- → section , “Connecting AC inputs of the expansion unit EASY618-AC-RE”, page 31
- → Section “Connecting DC inputs of the expansion units”, Page 38
- → section , “Connecting analog inputs of the EASY4..-DC-ME expansion units”, page 42

2.5.1 Connecting the digital inputs

DANGER

Connect the inputs for AC basic and expansion units in accordance with the safety regulations of VDE, IEC, UL and CSA with the same phase conductor that provides the power supply voltage. The device will otherwise not detect the switching level or may be destroyed by overvoltage.

Connecting the AC inputs of the easy800 standard

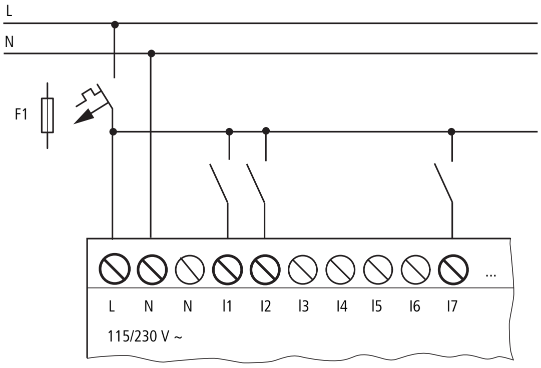


Figure 21:Connecting easy800 standard

- Input signal voltage range
  - Off signal: 0 to 40 V.
  - On signal: 79 to 264 V
- Input current
  - I1 to I6, I9 to I12: 0.5 mA/0.25 mA at 230 V/115 V.
  - I7, I8: 6 mA/4 mA at 230 V/115 V.

Connecting AC inputs of the expansion unit EASY618-AC-RE

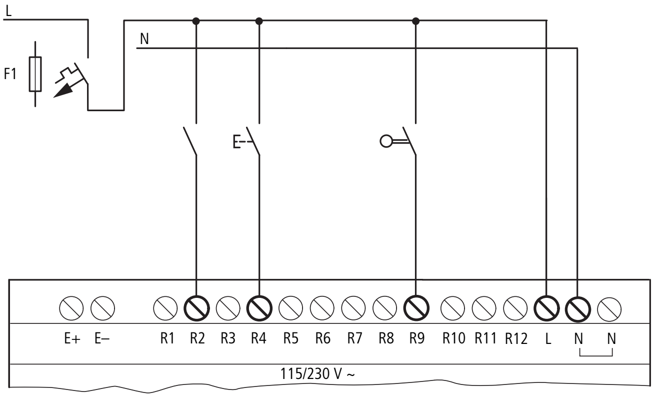


Figure 22:Connecting inputs on the easy618-AC-RE AC expansion unit

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

- Input signal voltage range
  - Off signal: 0 to 40 V.
  - On signal: 79 to 264 V
- Input current
  - R1…R12: 0.5 mA/0.25 mA at 230 V/115 V.

Length of input cables

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

- I1 to I6, I9 to I12: 100 m with input debounce switched on, 60 m without additional circuit with input debounce switched off.
- I7, I8: 100 m without additional circuit.
- R1 to R12: 40 m without additional circuit

The following also applies to the expansion units:
With longer cables connect a diode (e.g. 1N4007) with a minimum reverse voltage of 1 kV and a forward current of 1 A in series to the device input. Ensure that the diode is pointing to the input as shown in the circuit diagram; otherwise the device will not be able to detect the 1 state.

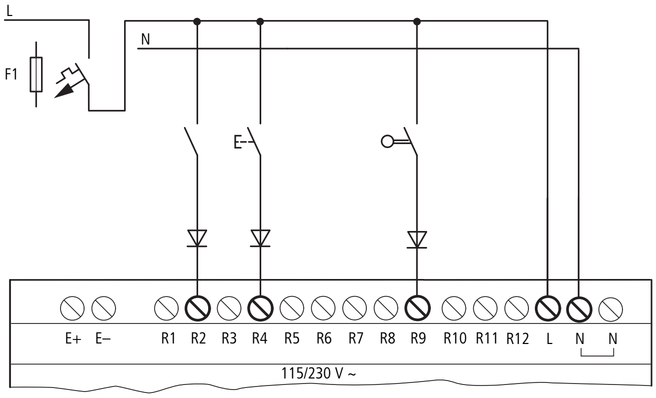


Figure 23:Connecting inputs on the EASY618-AC-RE AC expansion unit via diodes

Connecting inputs I7/I8 on the AC easy800 standard

Neon lamps with a maximum residual current of 2 mA/1 mA at 230 V/115 V can be connected to I7 and I8.

→ Always use neon lamps that are operated with a separate N connection.

WARNING

Do not use any Reed contacts at the I7 and I8 inputs. These may burn or melt due to the high inrush current of I7, I8.

Two-wire proximity switches have a residual current in the 0 state. If this residual current is too high, the device only detects the 1 state at the input.

With two-wire proximity switches or sensors with similar residual current consumption use therefore the inputs I7 and I8.

Use an additional input circuit if several inputs are required with a higher input current.

Increased input current

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

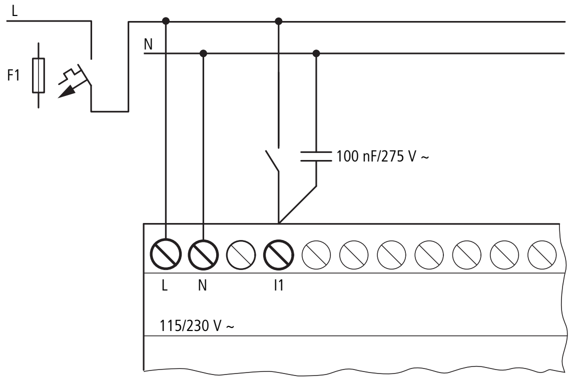


Figure 24:Increasing the input current on the easy800 standard AC version

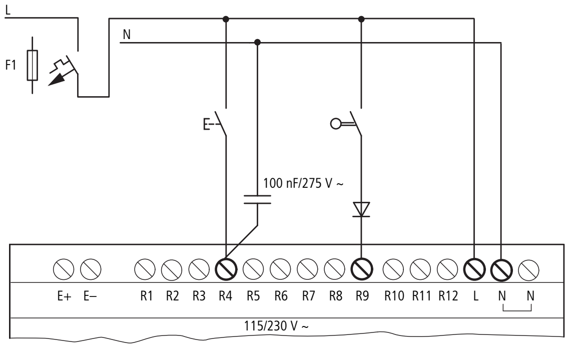


Figure 25:Increasing the input current at the AC expansion unit EASY618-AC-RE

→ When using a 100 nF capacitor, the drop out delay 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.

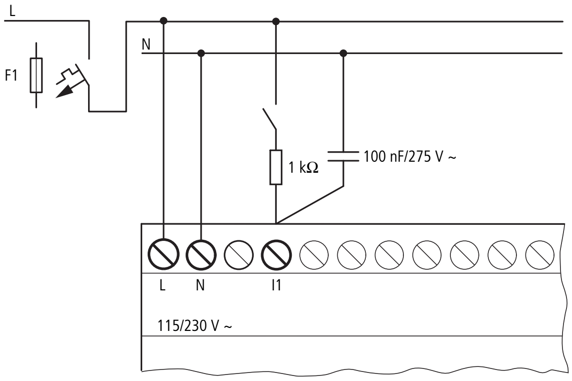


Figure 26:Limitation of the input current with a resistor at the AC easy800 standard

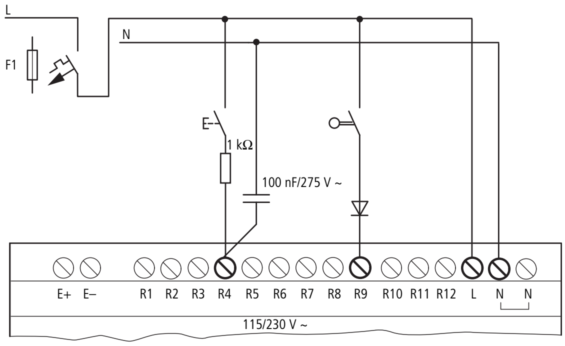


Figure 27:Limitation of the input current with a resistor at the AC expansion unit EASY618-AC-RE

Complete devices for increasing the input current are available under the part no. EASY256-HCI.

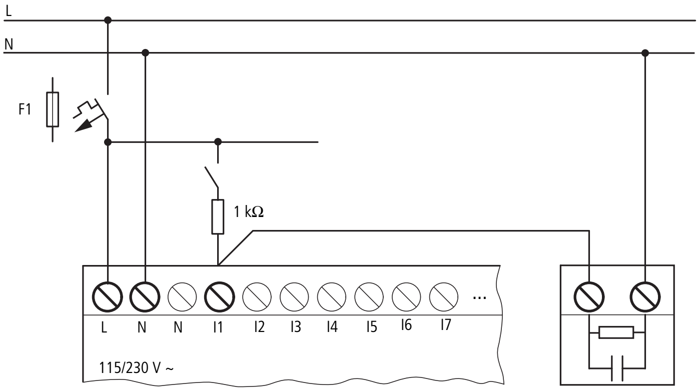


Figure 28:easy800 standard as AC version with EASY256-HCI to increase the input current

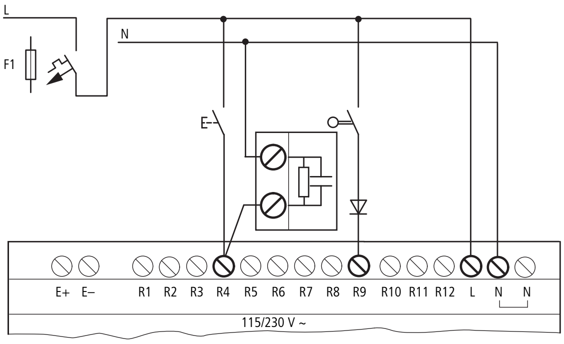


Figure 29:AC expansion unit with EASY618-AC-RE with EASY256-HCI to increase the input current

Connecting the DC inputs of the easy800 standard

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

Use input terminals I1 to I12 of the DC easy800 standard to connect pushbutton actuators, switches or 3- or 4-wire proximity switches. Given the high residual current, do not use 2-wire proximity switches.

These DC easy800 standard enable you to read both digital and analog input signals via the inputs I7 (IA01), I8 (IA02), I11 (IA03) and I12 (IA04).

- Digital input signal voltage range
  - I1 to I6, I9, I10.
    Signal OFF: 0 to 5 V.
    Signal ON: 15 to 28.8 V.
  - I7, I8, I11, I12.
    Signal OFF: < 8 V.
    Signal ON: > 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

→ Feed the digital inputs from the same 24 V DC power source as the device.

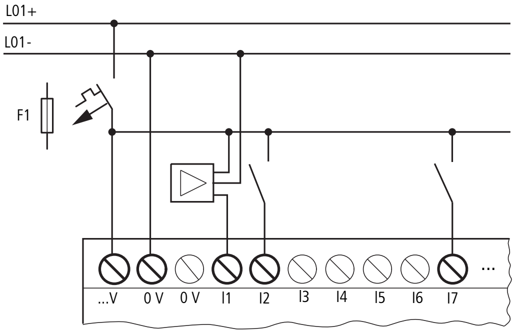


Figure 30:Application example with easy800 standard DC version

Connect easy-DC inputs EASY806-DC-SWD

Use input terminals I1 to I4 of EASY806-DC-SWD to connect pushbutton actuators, switches or 3- or 4-wire proximity switches. Given the high residual current, do not use 2-wire proximity switches.

- Digital input signal voltage range
  - I1 to I4
    Signal OFF: 0 to 5 V.
    Signal ON: 15 to 28.8 V.
- Input current:
  - I1 to I4: 3.9 mA at 24 V

→ Feed the digital inputs from the same 24 V DC power source as the device.

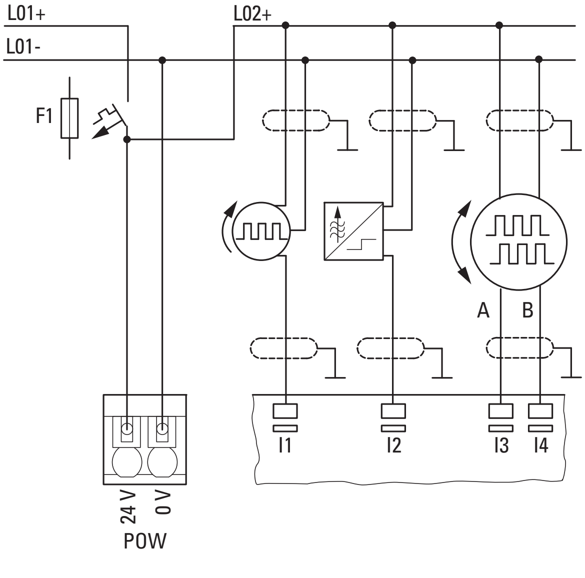


Figure 31:Application example with EASY806-DC-SWD

Connecting high-speed counters and frequency generators

With the easy800 standard DC version and with EASY806-DC-SWD, you can accurately measure high-speed counter signals or frequencies on digital inputs I1 - I4 independently of (i.e. bypassing) the cycle time.

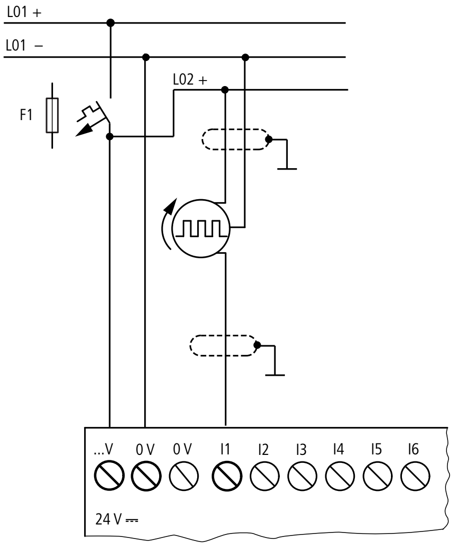


Figure 32: High-speed counters, frequency encoders on the easy800 standard DC version

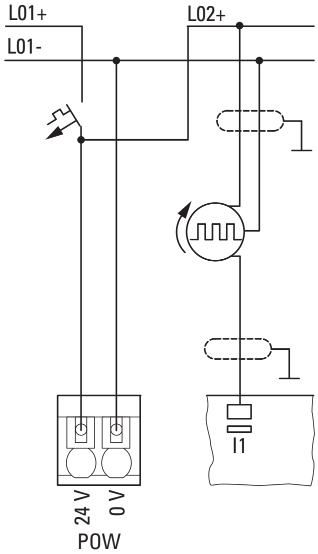


Figure 33: High-speed counters, frequency generators on EASY806-DC-SWD

Connecting incremental encoders

The easy800 standard DC version and EASY806-DC-SWD allow you to connect an incremental encoder to each of the digital inputs I1, I2, I3 and I4.

This enables you to count high-speed incremental values independently of the cycle time. The incremental encoder must output two 24 V DC square wave signals with a phase offset.

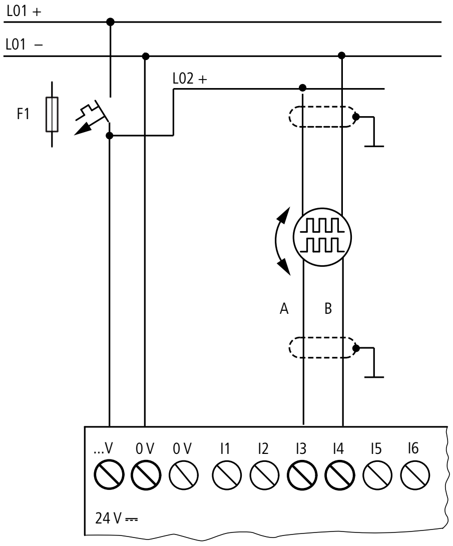


Figure 34:Connecting an incremental encoder on the easy800 standard DC version

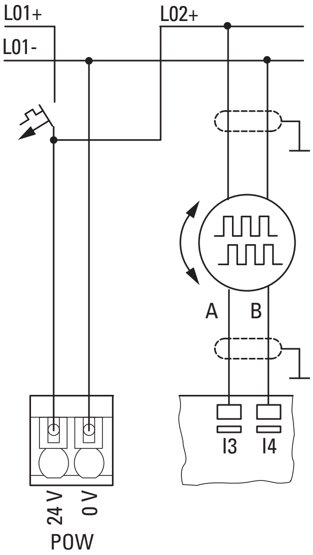


Figure 35:Incremental encoder on EASY806-DC-SWD

Connecting DC inputs of the expansion units EASY410... and EASY6...

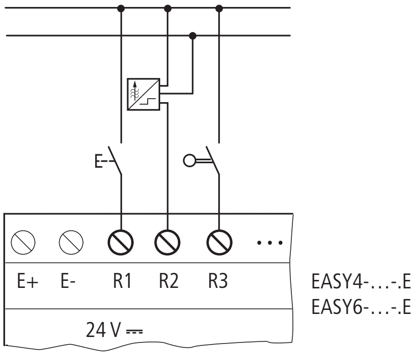


Figure 36:EASY410-.C-.E and EASY6..-DC-.E (digital expansion unit)

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

- Input signal voltage range
  - R1 to R6 (EASY410...), or R1 to R12 (EASY6...)
    Signal OFF: 0 to 5 V.
    Signal ON: 15 to 28.8 V.

- Input current
  - R1…R6 or R12: 3.3 mA at 24 V

Connecting DC inputs of the EASY4..-DC-ME expansion units

In addition to the analog inputs/outputs, the EASY4..-DC-ME DC expansion units are provided with the digital input R1. In addition to R1 you can also always use the analog 0 V - 10 V voltage inputs R2/R3 (terminals 2 and 7) as additional digital inputs and actuate them with up to 24 V DC.

Both expansion devices output a 0 signal on R2/R3 if the voltage goes below 8 V DC and outputs a 1 if this voltage is exceeded.

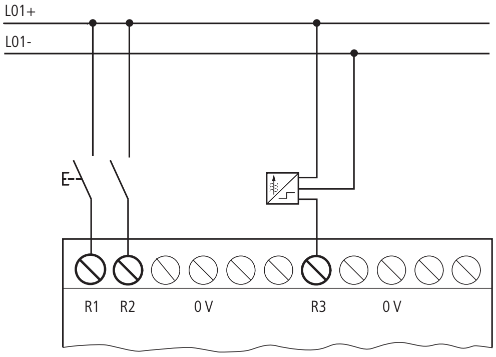


Figure 37:Digital inputs of the EASY4..-DC-ME

→ R2/R3 can always be used as additional digital inputs, even if they have not been expressly configured in the programming software as analog voltage inputs. Current and temperature inputs cannot be used as digital inputs.

Further information on the voltage range and input current of the input signals → section , “Connecting the DC inputs of the easy800 standard”, page 35.

2.5.2 Connecting analog inputs

All easy800 standard devices with a 24 V DC supply voltage are provided with analog inputs. On the easy800 standard from device version 02, it is also possible to connect the EASY406-DC-ME and EASY411-DC-ME DC expansion units (→ section , "Connecting DC inputs of the EASY4..-DC-ME expansion units", page 39).

DANGER

Analog signals are more susceptible to interference than digital signals, so that the signal cables must be laid and connected more carefully. The measures described below must be adhered to in order to prevent any deviations in analog values. An incorrect connection can lead to unwanted switching states.

- ▶ Use screened or twisted pair cables.
- ▶ Keep signal cables as short as possible and not longer than 10 m.
- ▶ With short cable lengths ground the shield of the signal cables at both ends with a large surface area connection. With longer signal cables, the shield should only be grounded at one end, on the side of the easy800 standard or expansion unit. Otherwise compensation currents between both grounding points may flow, leading to the interference of analog signals.
- ▶ Lay signal cables separately from heavy current cables.
- ▶ Connect inductive loads to be switched via the easy800 standard device outputs to a separate power feed, or use a suppressor circuit for motors and valves. If loads such as motors, solenoid valves or contactors and easy800 standard are operated with the same power feed, switching may give rise to interference on the analog input signals.
- ▶ Ensure a galvanic connection of the reference potential as described below.
- ▶ Avoid the operation of the easy800 standard or expansion unit on an ungrounded power supply.

Connecting the analog inputs of the easy800 standard

You can read analog voltages in the range 0 to 10 V via the inputs I7, I8, I11 and I12 of the easy800 standard DC version.

The signal has a 10-bit resolution, value range 0 - 1023.

The following therefore applies:

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

The analog voltage inputs can also be used as digital inputs.

The following circuits are examples of using the analog value measuring via inputs of the easy800 standard DC version.

→ Ensure that the reference potential is galvanically 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 easy800 standard power supply.

Setpoint potentiometer

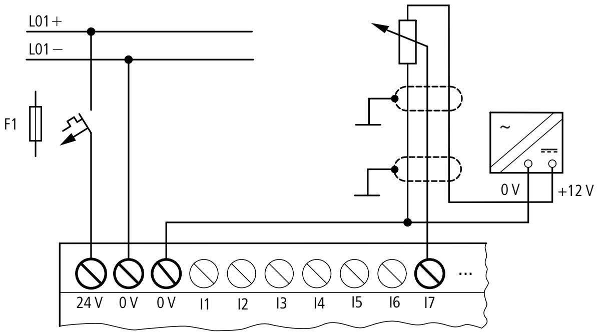


Figure 38:Setpoint potentiometer

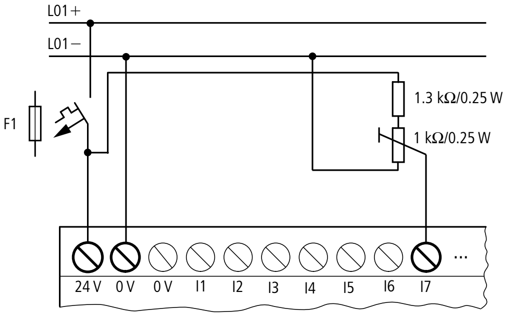


Figure 39:Setpoint potentiometer with upstream resistor

Use a potentiometer with the resistance ≤ 1 kΩ e.g. 1 kΩ, 0.25 W.

Brightness sensor

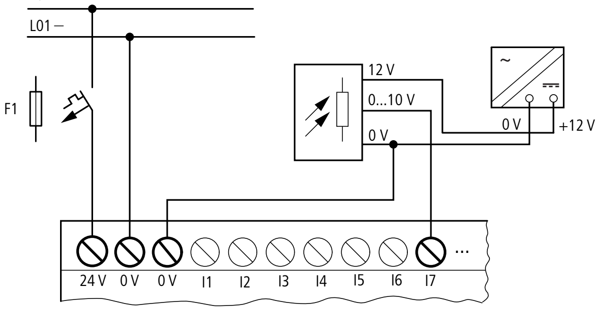


Figure 40:Brightness sensor

Temperature sensor

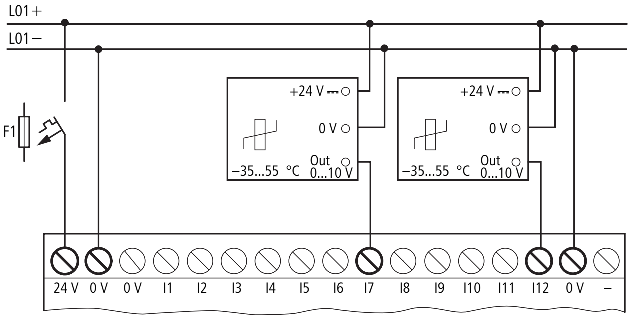


Figure 41:Temperature sensor connection

20 mA sensor

A 4 to 20 mA (0 to 20 mA) sensor can be connected on the voltage input using an external 500 Ω resistor (twice 1 kΩ in parallel).

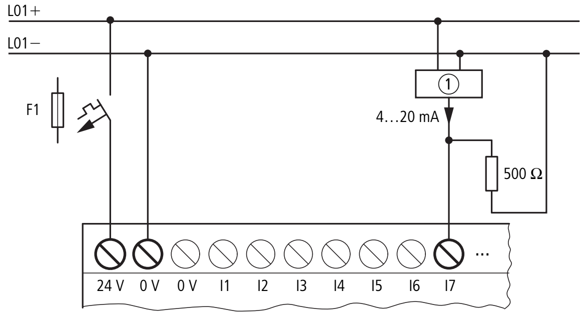


Figure 42:20 mA sensor

① Analog sensor

The following values apply:

- 4 mA = 2.0 V
- 10 mA = 5.0 V
- 20 mA = 10.0 V

(according to U = R · I = 500 Ω · 10 mA = 5.0 V)

Modules with ..TAP13.. in their designation are provided with an additional analog output.

Connecting analog inputs of the EASY4..-DC-ME expansion units

These DC expansion units allow you to process analog values of voltage, current and temperature. The EASY4..-DC-ME DC expansion units convert analog input and output values with a 10-bit resolution (0 - 1023).
When a Pt100 sensor is connected, the DC expansion units operate in the range between -50 °C and 200 °C.

→ You process the analog input and output values using permanently assigned marker words (MW) which you configure and program in easySoft-Pro.

The DC expansion units are provided with digital inputs/outputs as well as analog inputs/outputs (→ Section “Connecting DC inputs of the EASY4..-DC-ME expansion units“,Page 39).

Two devices are available:

- With the EASY406-DC-ME you can read one “voltage”, “current” or “temperature” analog value via the first analog input (terminals 2 - 6) and via the second analog input (terminals 7 - 11).
- The EASY411-DC-ME allows you to evaluate measured values at the same time via the six analog inputs (terminals 2 - 11), and read voltage, current or temperature analog values via two inputs each.

Table 1: Terminals of the analog inputs on the EASY4..-DC-ME expansion units

Terminal(s)	EASY406-DC-ME	Meaning	EASY411-DC-ME	Meaning
2	1. Analog value	0 - 10 V, 0(4) - 20 mA or PT100 resistance sensor	1. Analog value	0 - 10 V
3	1. Analog value		2. Analog value	0(4) - 20 mA
4, 5, 6	1. Analog value		3. Analog value	PT100 resistance sensor
7	2. Analog value	0 - 10 V, 0(4) - 0 mA or PT100 resistance sensor	4. Analog value	0 - 10 V
8	2. Analog value		5. Analog value	0(4) - 20 mA
9, 10, 11	2. Analog value		6. Analog value	PT100 resistance sensor

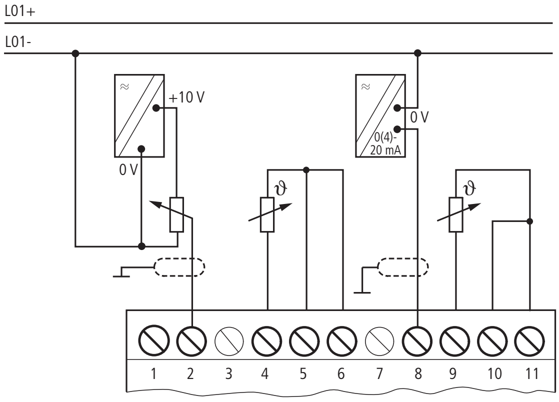


Figure 43:Connecting the analog inputs on the EASY4..-DC-ME expansion units

Voltages are measured in the 0 - 10 V signal range. A potentiometer with an upstream resistor that is connected between the supply voltage and reference potential can be used as a setpoint potentiometer. For voltage measuring, connect the output of the sensor to the voltage input (terminal 2 or 7), → section Figure 39:, “Setpoint potentiometer with upstream resistor”, page 41.

Currents are measured in the 4 - 20 mA (0 - 20 mA) signal range. For current measurement connect the output of the sensor to the current input (terminal 3 or 8), → section , “Temperature sensor”, page 42.

→ When current measuring is activated, we recommend the implementation of open-circuit monitoring via the diagnostics contact R13. The expansion unit switches the diagnostics contact to the 1 state if a current < 3 mA is present. Use current sensors for the 4 - 20 mA signal range if you wish to evaluate the diagnostics contact.

Temperatures are measured in the range -50 - +200 °C using PT100 temperature sensors. The sensor inputs 1 (terminal 4 - 6) and 2 (terminal 9 - 11) are used for temperature measuring.

On terminals 6 and 11, the analog expansion unit outputs a constant current to the temperature sensor.

Connect the ground connection of a sensor to the corresponding 0 V input (terminal 4 or 9) of the expansion unit.

You connect the temperature sensor directly to the relevant three terminals of these sensor inputs. A three-wire connection or a two-wire connection is possible.

With three-wire measuring, the line resistance is compensated by means of a measuring bridge in the expansion unit.

With two-wire connections, connect a short-circuit bridge between terminals 5 and 6 or 10 and 11 (→ Figure 43, page 43).

→ The inputs and outputs of these analog expansion units require the same reference potential (chassis ground) as the DC power supply of the sensors and actuators.
Connect the 0 V of the DC power supply of the different sensors with the 0 V of the DC power supply of the expansion unit.

2.6 Connecting outputs

Devices either have transistor outputs or relay outputs. easy800 standard outputs: The Q… outputs of both device versions are potentially isolated from the power supply and the inputs and therefore potential-free.

EASY806-DC-SWD outputs: There is no potential isolation between the outputs and

- the local inputs
- POW
- SWD

Galvanic isolation is provided between AUX and NET.

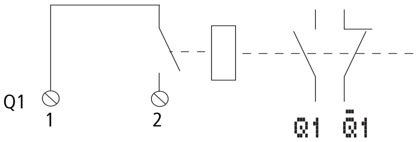


Figure 44:Connecting outputs, in this case Q1

The respective relay coils are actuated in the circuit diagram via the output relays Q1…Q6 (Q8).

You can use the signal states of the output relays as N/O or N/C contacts in the 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. Prior to installation observe the technical limit values and data for the outputs (→ Section "Transistor outputs“, Page 355).

See also:

- “Connect relay outputs“, Page 46.
- “Connecting transistor outputs“, Page 48.
- “Connect analog output“, Page 50.

The number of outputs of the easy800 standard can be increased by using an expansion unit. The outputs of the EASY202-RE, EASY4..-DC-.E and EASY6..-DC-. digital expansion units are detected by the easy800 standard automatically - without any configuration.

The universal EASY4..-DC-ME expansion units allow you to process both digital and analog values.

→ The factory activated analog outputs can also be processed immediately, but should always be adapted to your application during commissioning, → Section “Configuring the analog expansion unit“, Page 84.

All available input/output expansions → section 12.5, “Possible combinations of basic devices and expansion units”, page 362.

Whether an easy800 standard device has transistor or relay outputs can be easily determined by the product designation. The easy800 standard devices have an R in the key to part numbers for relay outputs and a T for transistor outputs, e.g. EASY819-DC-RC and EASY821-DC-TC.

easy806-DC-SWD is provided with two transistor outputs Q1 and Q2 in order to offer functions such as pulse width modulation and pulse output.

Parallel connection of outputs

The parallel connection of outputs Q1 and Q2 is not allowed on the easy806-DC-SWD.

Up to four outputs can be connected in parallel on easy800 standard and the expansion units in order to increase the power. This enables a maximum output current of 2 A.

Note that only outputs of the same group can be connected in parallel. Groups are

- Q1 to Q4
- Q5 to Q8
- S1 to S4
- S5 to S8

Outputs connected in parallel must be switched at the same time.

DANGER

Only outputs of the same group (S1 to S4 or S5 to S8) can be connected in parallel; e.g. S1 and S3 or S5, S7 and S8. Outputs connected in parallel must be switched at the same time.

2.6.1 Connect relay outputs

Connecting relay outputs of the easy800 standard

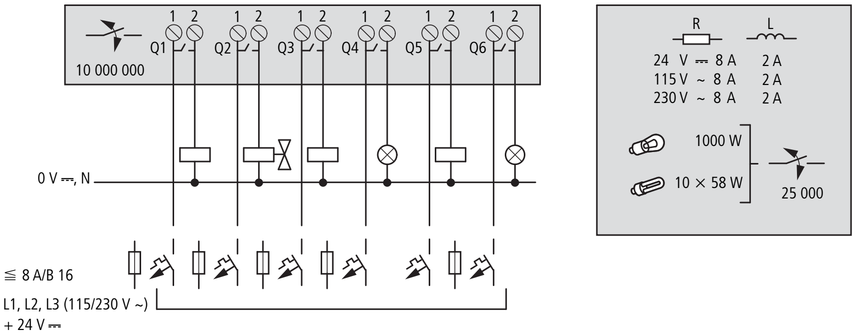


Figure 45:Relay outputs EASY8..-..RC.

DANGER

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 relay outputs of the expansion units

EASY4..-..-RE/EASY6..-..-RE

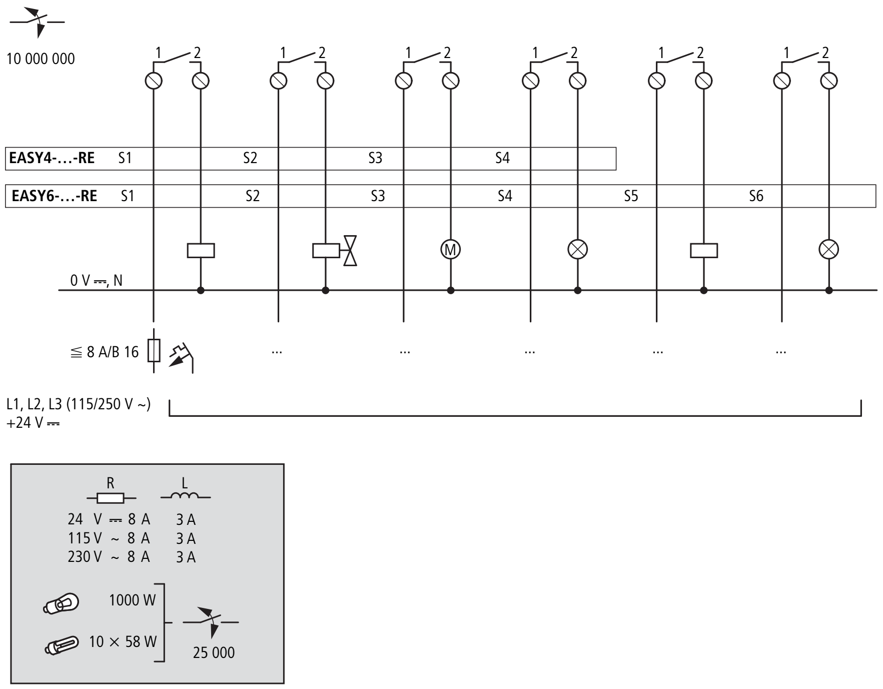


Figure 46:Relay outputs EASY4..-..-RE/EASY6..-..-RE.

EASY2..-RE

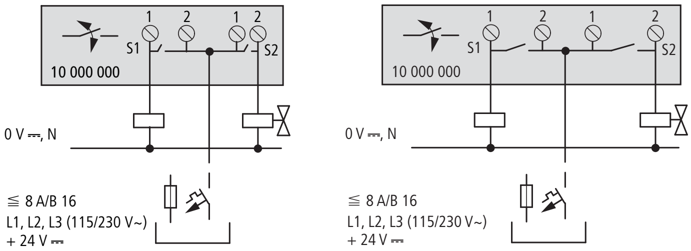


Figure 47:Relay outputs EASY2..-RE

Unlike the connection of the inputs, you can connect different phase conductors to the relay outputs of the EASY819-..-R.., EASY6..-..RE devices.

2.6.2 Connecting transistor outputs

The transistor outputs of easy800 standard must be provided with a suppressor circuit. The suppressor circuit is optional for easy800-SWD.

Suppressor circuit of the transistor outputs for easy800 standard

The switching off of inductive loads can produce high currents. Transistor outputs should be protected from this with a suppressor circuit. If you are not using a suppressor circuit, observe the following: Several inductive loads must not be switched off at the same time, so as not to adversely overheat the driver modules in the worst case. If the +24 V DC power supply is switched off via a contact in the event of an emergency, and if more than one actuated output with an inductive load can be switched off, you must provide these inductive loads with a suppressor circuit (→ Figure 48).

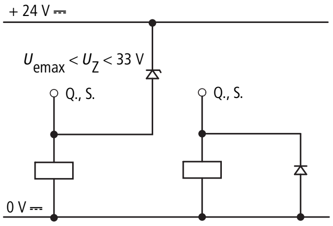


Figure 48:Inductive load with suppressor circuit

DANGER

Please note the following when switching off inductive load. Suppressed inductances cause less interference in the entire electrical system. It is recommended that the suppressor circuit is connected as close to the inductance as possible.

Behavior of transistor outputs in the event of a short-circuit/overload

The following function applies to easy800 standard:

In the event of a short-circuit or overload at a transistor output, this output will switch off and the group alarm I15 or I16 will be set to 1. The output will switch back on up to the maximum temperature after a cooling time that depends on the ambient air temperature and the current level. If the fault condition persists, the output will keep switching off and on until the fault is rectified or until the supply voltage is switched off (→ Chapter 10, “Diagnostics possibilities“, page 329).

easy800-SWD:

In the event of a short-circuit or overload at a transistor output, both outputs will switch off and the group alarm I16 will be set to 1. The output will switch back on after a wait time of 10 s and will measure the current. If the fault continues, the outputs will switch off and on until the fault is rectified or the supply voltage is switched off.

Connecting transistor outputs of the easy800 standard

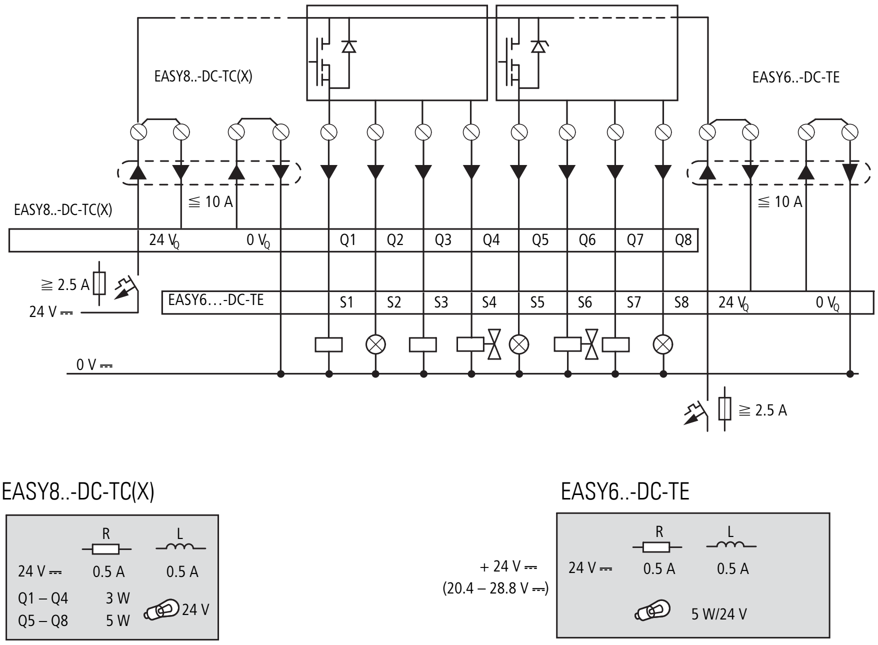


Figure 49:Digital transistor outputs EASY8..-DC-TC(X), EASY6..-DC-TE

→ Power the transistor outputs with voltage!

Connect transistor outputs EASY806-DC-SWD

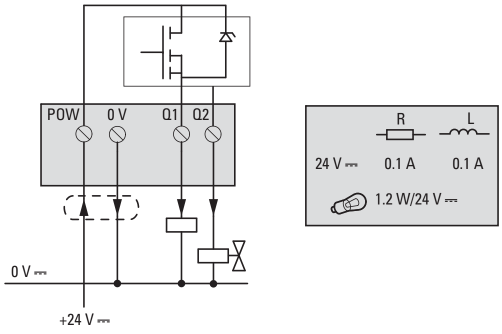


Figure 50:Transistor outputs EASY806-DC-SWD

Connecting transistor outputs of the expansion unit

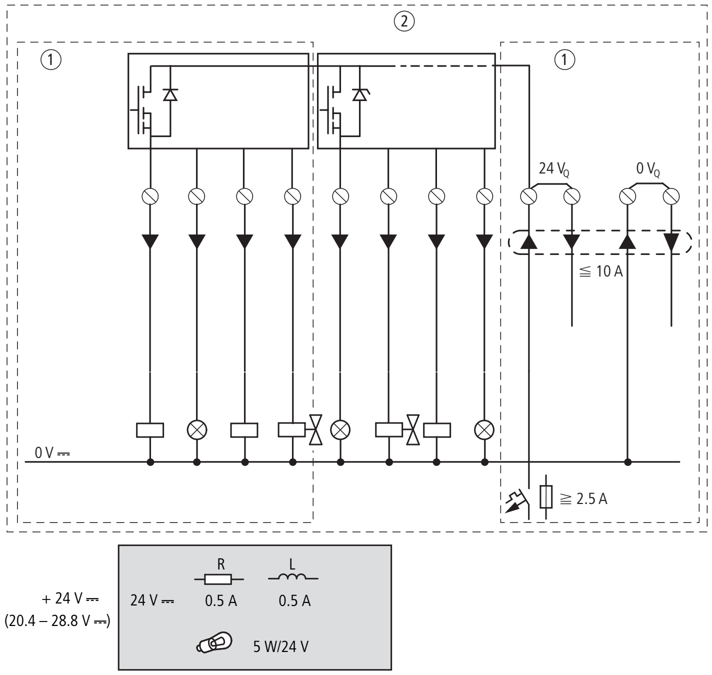


Figure 51:Digital transistor outputs EASY6..-DC-TE and EASY4...-DC-TE

① EASY4... -DC-TE
② EASY6... -DC-TE

2.6.3 Connect analog output

All easy800 standard devices with a 24 V DC supply voltage have an analog output QA 01. QA 01 can be used to output a DC voltage from 0 - 10 V, with a 10-bit resolution, value range 0 - 1023.

On the easy800 standard from device version 02, it is also possible to connect the EASY406-DC-ME and EASY411-DC-ME DC expansion units with an analog output (→ Section “Connecting DC inputs of the EASY4..-DC-ME expansion units“, Page 39).

The analog output can be used, for example, for controlling solenoid valves and other actuators.

Observe the hazard warnings concerning the processing of analog signals (→ section 2.5.2, “Connecting analog inputs”, page 40) and the note concerning the same reference potential (chassis ground).

Connect the analog output of easy800 standard

Connecting solenoid valve

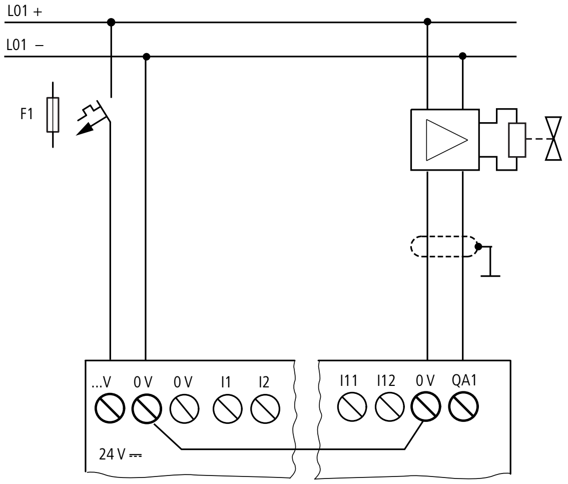


Figure 52:Connecting a solenoid valve; the 0V terminals are connected inside the device.

Setpoint entry for a drive

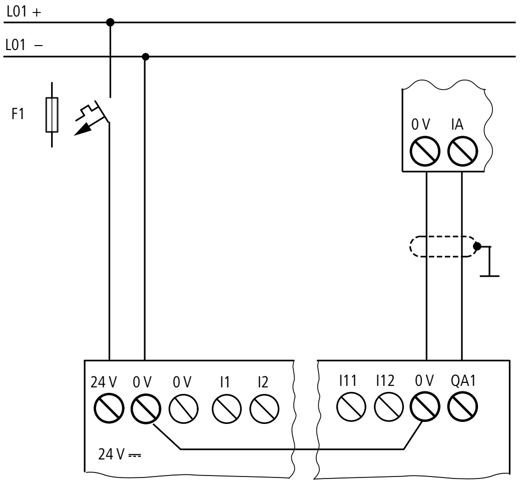


Figure 53:Setpoint entry for a drive; the 0V terminals are connected inside the device.

Connecting the analog output of the EASY4..-DC-ME expansion units

Analog expansion units are used to increase the number of outputs. The EASY406-DC-ME analog expansion unit offers you one and the EASY411-DC-ME two additional analog outputs. Both devices provide a DC voltage of 0 - 10 V with a 10-bit resolution and 0 - 1023 value range. Connect the actuator(s) to QA1(/QA2) and to the relevant 0 V terminal to its (their) left.

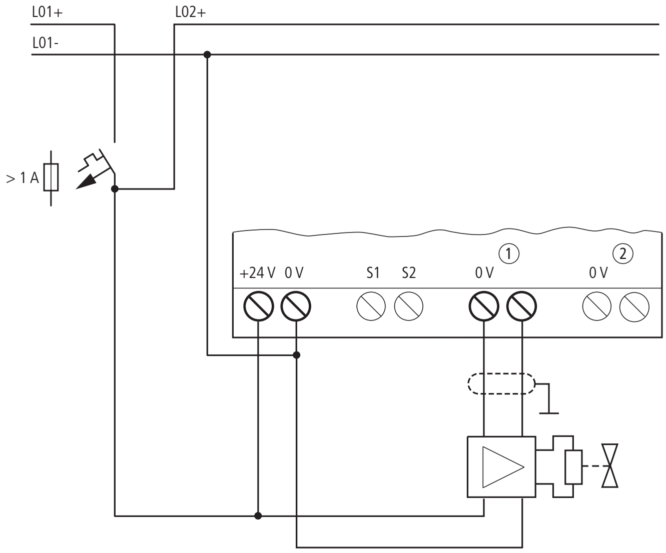


Figure 54:Connecting the analog output of an EASY4..-DC-ME expansion unit

① Analog output QA1 (EASY406-DC-ME/EASY411-DC-ME), 0 - 10 V,

② Analog output QA2 (only EASY411-DC-ME), 0 - 10 V.

→ Then configure the analog inputs/outputs you have connected, → Section “Configuring the analog expansion unit“, Page 84.

2.7 NET easynetwork

2.7.1 Connecting the easyNET network

easyNet is a network to which a maximum of 8 stations can be connected. Any device with a network connection can be an easyNet station. EASY802-DC-SWD is therefore excluded.

The easyNet has a linear topology. Two types of cable routing are possible:

- Passing through.
- T connector and spur line.

Passing through

With this wiring arrangement, easyNet is looped through the device. Advantage: The easyNet stations can be addressed and easyNet configured very simply via station 1 (→ section 3.10.1, "Parameterizing NET stations", page 91) or via easySoft-Pro.

Disadvantage: If the line is interrupted, the easyNet network is no longer operational due to the missing bus termination.

T connector and spur line.

Advantage: If an easyNet station fails, the spur line can be removed from the station for the exchange, whilst all other devices on the easyNet remain functional.

Each device must be addressed individually with this wiring method. This can be done as follows:

- By downloading the program including the NET-ID via easySoft-Pro (see Online Help). For this the programming cable must be plugged in locally on each individual device.
- Via the function key and display of the individual device (→ section 3.10.1, "Parameterizing NET stations", page 91).

→ The spur line between the T connector and the device must not exceed 0.3 m. Otherwise communication via easyNET may be impaired.

When using the EASY223-SWIRE gateway, you cannot use any spur lines as the device cannot be addressed individually due to the lack of a serial interface.

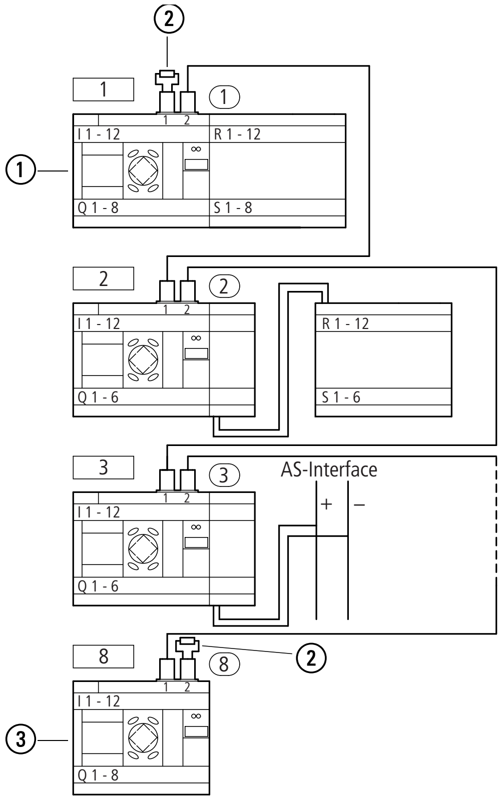


Figure 55:Example of loop-through wiring of easy800 standard

① First station on the easyNet
② Bus termination resistor
③ Last station on the EasyNET network
□ Physical location, place
⬭ Station number

Further information on configuring and commissioning the easyNet → Chapter 6, "The network easyNet", Page 273. Specifications for the easyNet are provided in the Technical Data, Page 359.

Pin assignment of the RJ45 socket on the device

The networkable devices are provided with two RJ45 sockets.

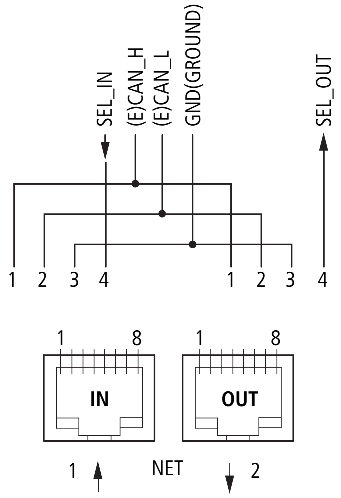


Figure 56:RJ45 socket

Socket 1 (IN) is used for connecting the incoming data cable from the physically preceding easyNet station. Socket 2 (OUT) is used for connecting the outgoing data cable to the physically next easyNet station.

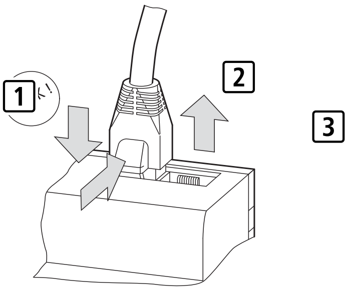
Bus termination resistor

The physically first and last stations in the easyNET network must each be terminated with a bus termination resistor. Because the physically first easyNet station has no predecessor, the bus termination resistor is connected to socket 1 here. Accordingly, the bus termination resistor is connected to socket 2 of the last easyNet station.

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

Fitting and removing the easyNet connection cable

- ▶ With easy800 standard you must first remove the cover plate. Both RJ45 interfaces become visible.
- ▶ Plug in the cable; it must snap into the socket with an audible click 1.
- ▶ Before a cable is removed, the mechanical interlock must be undone 2, 3.



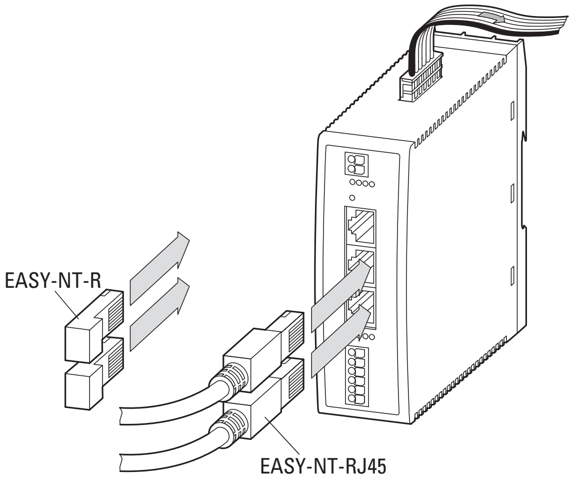


Figure 57: Plugging and unplugging a cable easy800 standard

Figure 58: Plugging and unplugging a cable easy806-DC-SWD

Prefabricated

The following easyNet connection cables are available for simple installation:

Table 2: Prefabricated cables, RJ45 plug on both ends

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

Prepared by user

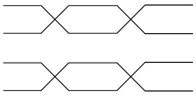
The following components are available for assembling different cable lengths:

- Connection cable
  - Type - EASY-NT-CAB, cable 100 m, 4 x 0.18 mm^2
- Bus connection plug
  - RJ45 plug, part no.: EASY-NT-RJ45 (8-pole)
- Tools
  - Crimping tool for RJ45 plug, part no.: EASY-RJ45-TOOL. AWG 24, 0.2 mm^2 is the largest crimpable cross section.

The table on Page 59 provides information for the required cable cross-section of longer cable lengths, taking the cable resistance into account.

Cable layout

4-core, twisted pair; → section 12.3.7, “NET easynetwork”, page 359.

	Cable pair	Pen	Line
	A	1	Data cable ECAN_H
	A	2	Data cable ECAN_L
	B	3	Ground cable GND (Ground)
	B	4	Selection cable SEL (IN/OUT)

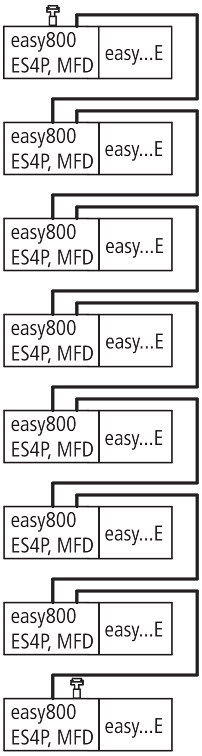
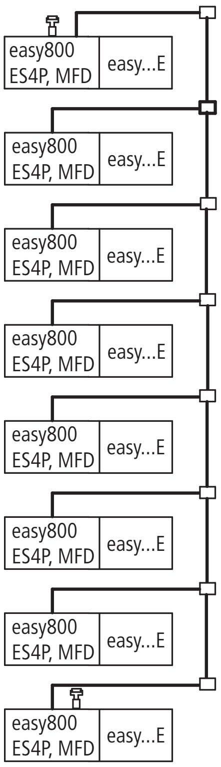
The most basic easyNet configuration operates with the cables ECAN_H, ECAN_L, GND. The selection cable is only used for automatic addressing and configuration.

easyConnecting Net stations

Proceed as follows in order to connect the easyNet stations:

- Plug in the easyNet connection cable into the easyNet socket 2 of the first physical easyNet station.
- Plug in the easyNet connection cable into the easyNet socket 1 of the last physical easyNet station.
- Fit the bus terminating resistor in socket 1 of the first station and in socket 2 of the last station.
- Connect the power supply to all stations.

Addressing examples

Physical location, place	Station number		"Loop through the unit"	T connector and spur line
	Example 1	Example 2		
1	1	1		
2	2	3		
3	3	4		
4	4	8		
5	5	7		
6	6	2		
7	7	6		
8	8	5		

- 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.

2.7.2 Cable length with cross-sections

For correct operation of the easyNet network the cable lengths, cross-sections and cable resistances must match those listed in the following table.

Cable length	Cable resistance	Cross section	
m	mΩ/m	mm²	AWG
- 40	≦ 140	0.13	26
- 175	≦ 70	0.25 - 0.34	23, 22
- 250	≦ 60	0.34 - 0.5	22, 21, 20
- 400	≦ 40	0.5 - 0.6	20, 19
- 600	≦ 26	0.75 - 0.8	18
- 1000	≦ 16	1.5	16

The impedance of the cables used must be 120 Ω.

→ The installation of an expensive fibre optic link is recommended with cable lengths > 500 m.

The maximum transmission speed depends on the total length of the network cables, (→ section 12.3.7, "NET easynetwork", page 359).

Calculating the cable length for a known cable resistance

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

Baud Rate	Cable resistance R_L
Kbit/s	Ω
10 - 125	≦ 30
250	≦ 25
500, 1000	≦ 12

l_{max} = Maximum cable length in m.
R_L = Total cable resistance in Ω.
R' = Cable resistance per unit length in Ω/m.

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

Calculating cross-section with known cable lengths

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

l = Cable length in m
S_{min} = Minimum cable cross-section in mm^2
ρ_{cu} = Resistivity of copper, if not otherwise stated 0.018 Ωmm^2/m

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

→ If the calculation result does not correspond to a standard cross section, take the next higher cross section.

Calculating length with known cable cross-section

The maximum cable length for a known cable cross-section is calculated as follows:

l_{max} = Cable length in m
S = Cable cross-section in mm^2
ρ_{cu} = Resistivity of copper, if not otherwise stated 0.018 Ωmm^2/m

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

2.8 Connecting the serial interface

Each easy800 device is provided on the front with a serial interface. The interface on easy800 standard is factory shipped with a protective cover fitted. Remove this cover carefully.

The serial interface is used for:

- connecting the programming cable to the COM serial interface on the PC and thus as a connection for the easySoft or easySoft-Pro programming software.
- the point-to-point communication with MFD-CP4, MFD-CP8 or MFD-CP10. MFD-CP… must be operated in Terminal mode and thus enable texts to be output and operator entries. This terminal connection is supported by easy800 standard from device version 04 and easy800-SWD from device version 01. Information on MFD-CP4 is provided in the manual → MN05013011Z-EN.
- the point-to-point communication between an MFD in COM-LINK mode and an easy800 device for exchanging data (MB, MW, MD). This serial connection is supported by easy800 standard from device version 04 and easy800-SWD from device version 01.
- the communication via the EASY209-SE Ethernet gateway with an Ethernet network, see manual "EASY209-SE Ethernet Gateway", MN05013012Z-EN.
- a memory card on easy800 standard ("memory interface" function).

The following chapters show the permissible cables for connecting your easy800 device via the serial interface.

2.8.1 Connection to a PC

Connect one of the following programming cables to the COM or USB interface of your PC and to the serial interface of the easy800 device.

Table 3: Programming cable for the connection to a PC

Device	Programming cable	Baud Rate
easy800 standard	EASY800-PC-CAB	19.2 kBaud
easy800 standard	EASY800-USB-CAB	57.6 kBaud
easy800 standard	EASY800-MO-CAB	57.6 KBaud, universal high-speed cable
easy800-SWD	EU4A-RJ45-USB-CAB1	57.6 kBaud

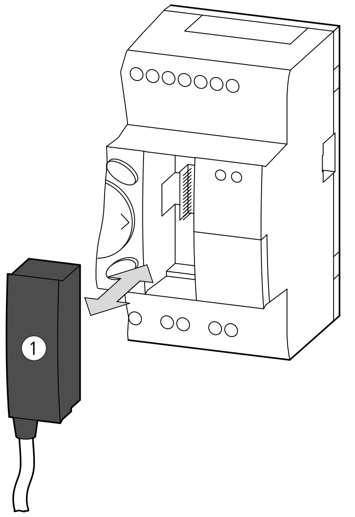


Figure 59: Fitting the programming cable ① to an easy800 standard

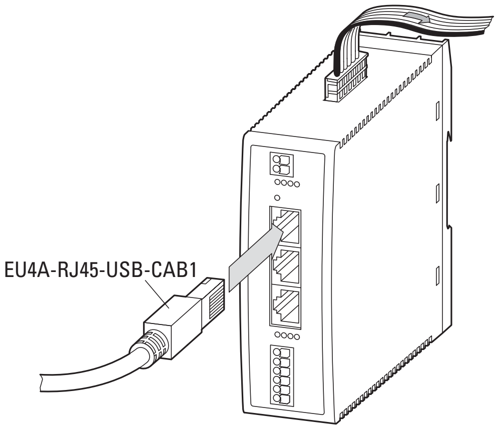


Figure 60: Fitting the programming cable to an easy800-SWD device

CAUTION

When connecting the programming cable, the easySoft-Pro programming software must be offline. On no account should the programming cable be moved from one easy800 device to another when the connection is online.

2.8.2 Connection to a modem

To set up a connection via modems connect a standard modem to the COM port of your PC and one to the serial interface of the easy800 standard device.

Modem parameter assignment

- Transfer speed 9.6 Kbit/s.
- Data format 8-N-1.

Connecting the modem

- On the PC side connect the modem via a null modem cable on which RxD and TxD are reversed.
- Connect the easy800 standard device to the modem via an EASY800-MO-CAB cable.

Table 4: Tab. SUB-D connector assignment on the EASY800-MO-CAB for modem operation

Modem SUB-D 9-pole, Pin	Wire color of the EASY800-MO-CAB
2	brown
3	white
4	gray
5	green
6	yellow

CAUTION

You can link up easy800 standard via a modem connection (9.6 Kbit/s) and locally via a universal high-speed cable (up to 57.6 Kbit/s). If slower access via the modem connection must be possible after local access, the transfer speed for easy800 standard must be reset to 9.6 Kbit/s.

The timing requirements for fault-free data exchange via the serial interface are described in → section 9.7, “Monitoring the serial interface”, page 327.

2.8.3 Connecting for point-to-point communication

Connect a connection cable from the MFD device to the serial interface of the easy800 device.

Table 5: Prefabricated connection cables for point-to-point communication

Device	Connection cable	Device	Baud Rate
easy800 standard	MFD-CP4-800-CAB5	EASY209-SE	up to 57.6 KBaud
		MFD-CP4	up to 19.2 KBaud
	MFD-800-CAB5	MFD-CP8 MFD-CP10	up to 19.2 KBaud
easy800-SWD	EU4A-RJ45-CAB2	MFD-CP4	up to 19.2 KBaud
		EASY209-SE	up to 57.6 KBaud

→ In order ensure compliance with EMC requirements, the MFD-800-CAB5 cable must not be lengthened.

Connecting example easy800 standard to MFD-...-CP8/CP10

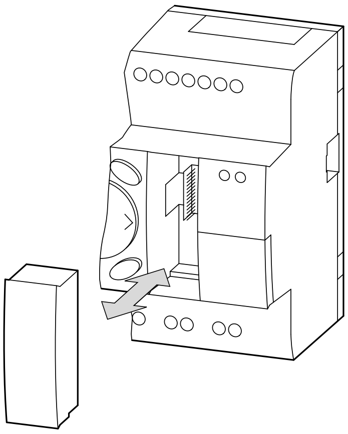


Figure 61:Fitting/removing the interface cover

▶ Remove the cover from the serial interface and connect the connector plug of the MFD-800-CAB5.

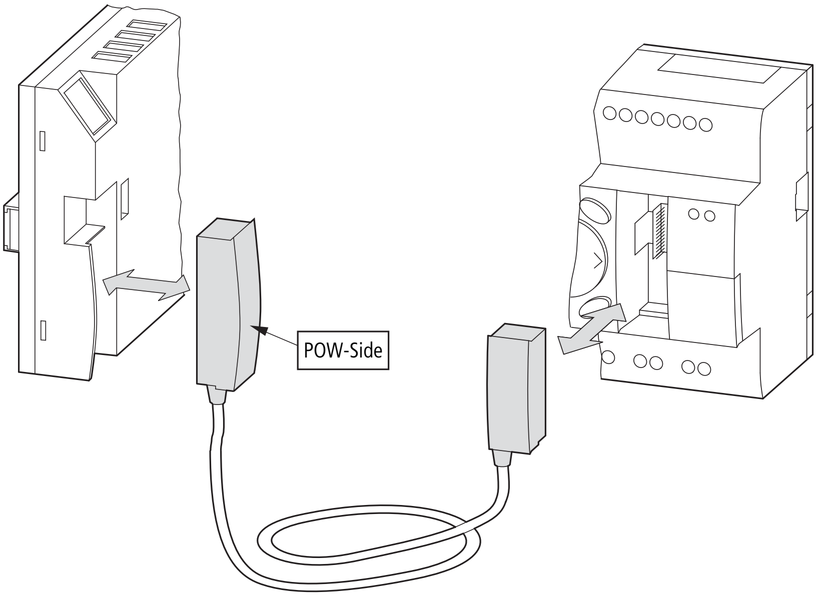


Figure 62:Fitting the bus connector plug: left MFD-…-CP8/CP10 with plug “POW-Side”

- ▶ The connector marked POW-Side must be plugged into the interface of an MFD.

→ The serial interface only functions if the MFD device is providing the power feed required for the interface cable.

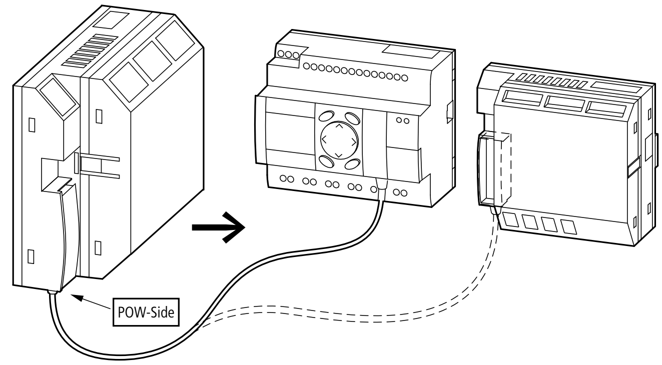


Figure 63:Point-to-point serial interface

Connecting example easy800-SWD to MFD-CP4

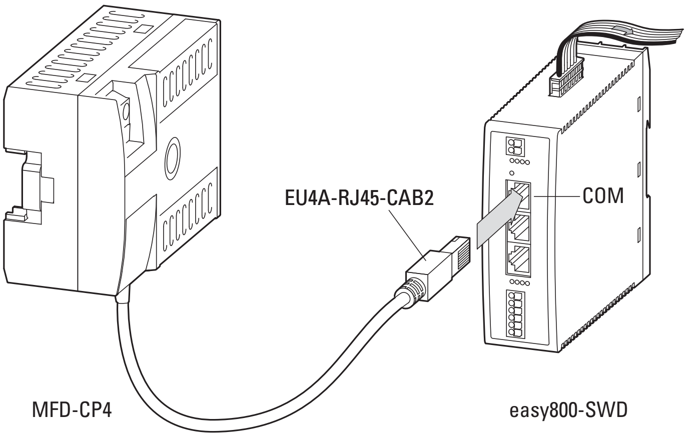
easy800-SWD devices are not provided with an operator keypad and display. In order to operate easy800-SWD directly, connect an MFD-CP4. For this use the EU4A-RJ45-CAB2 cable and observe the following terminal assignments:

Table 6: Terminal assignment for connecting MFD-CP4 to the serial interface of easy800-SWD

MFD-CP4	Wire color of the EU4A-RJ45-CAB2
1	-
2	brown
3	blue
4	white
5	-

Carry out the following work with the installation in a de-energized state:

- ▶ Use the EU4A-RJ45-CAB2 cable.
- ▶ Insert the cables into the push-in terminals of the MFD-CP4 according to the terminal assignment → Table 6.
- ▶ Insert the RJ45 plug into the serial interface of the easy800-SWD device.



2.8.4 Inserting the memory card

easy800 standard devices support the use of memory cards that are fitted into the serial interface. The memory card can be used to store a single program that can be read from any suitable easy800 standard device.

With easy800 standard only use the EASY-M-256K card.

▶ Remove the cover from the interface and connect the connector memory card → Figure 64, page 66.

→ With easy800 standard you can also remove or insert the memory card when the power supply is switched on without any risk of data loss.

Working with the memory card → section 4.6, “Program transfer from and to the memory card”, page 124.

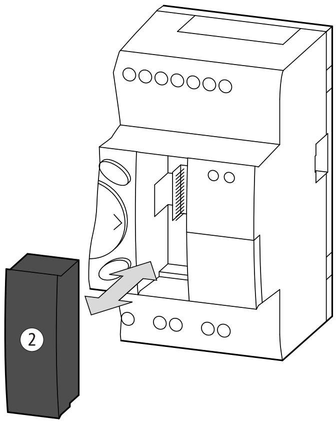


Figure 64:Connecting and removing the memory card ②

2.9 Install SmartWire-DT

SmartWire-DT is only supported by easy800-SWD devices.

Connection SmartWire-DT

> ⚠ Warning
> SWD stations may be destroyed if they are refitted in the SWD line when the power supply is on. Switch off the power supply beforehand.

The ribbon cable is provided with direction arrows. Proceed as follows to connect SmartWire-DT:

1. Prepare the ribbon cable or use a prefabricated cable (see manual MN05006002Z-EN).
2. Fit the bus termination to the blade terminal to which the arrow in the ribbon cable is pointing.
3. Finally check the correct polarity of the SWD stations:
   - All arrows provided on the SWD stations and ribbon cable must be pointing in the same direction.
   - The black conductor of the ribbon cable must be on the same height as the station‘s designation +15 V shown in black. (see manual MN05006002Z-EN).
4. Hold the blade terminal from which the arrow on the ribbon cable is pointing away.
5. Insert this blade terminal into the SWD-OUT socket.

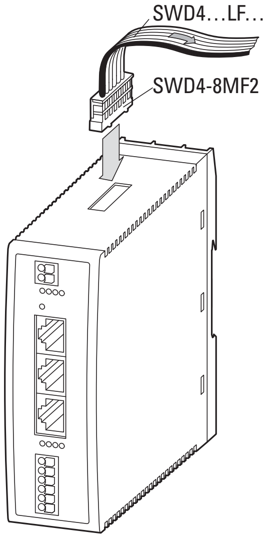


Figure 65:Connecting the SWD system

Potential isolation

SmartWire-DT is galvanically isolated from:

- AUX
- COM
- easyNet

SmartWire-DT is galvanically connected to:

- POW
- Inputs
- Outputs

3 Commissioning

The easy800 devices without display and operator functions can be commissioned. However, a display and operating facility is required in order to follow all explanations in this chapter. If you are operating an easy800, you can connect via a point-to-point connection or easyNet display and operating units. To find out the correct procedure, read in → section "Connecting example easy800 standard to MFD-…-CP8/CP10", page 63 for easy800 standard or in → section "Connecting example easy800-SWD to MFD-CP4", page 64 for easy800-SWD.

3.1 Switch on

Before startup, check whether the power supply, inputs and outputs, as well as the optional easyLink, easyNet and SmartWire-DT connection are properly connected.

DANGER

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

Startup behavior of easy800 devices without a program

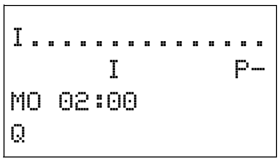
easy800 does not start without a program but switches to STOP mode. The meaning of the LEDs is provided in → section "LEDs of easy800-SWD", page 295.

Startup behavior of easy800 devices with a program

In the following considerations we assume that a valid program is provided in easy800 and no peripheral faults are present that may cause STOP.

The following cases are possible:

easy800 standard with integrated display



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

easy800 standard shows the Status display when you switch on. This status display shows the switching state of the inputs and outputs and also indicates whether the device is currently running a circuit diagram.

The device switches to the operating state that was set with the device parameters START MODE, CARD MODE and REMOTE RUN (see → section "Activating RUN MODE", page 304, → section "Activating CARD MODE", page 305, → section "Automatic change of the RUN and STOP mode (Station Menu REMOTE RUN)", page 280).

easy800 without integrated display

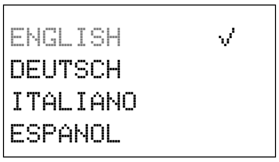
The device parameters RUN MODE and CARD MODE are activated automatically on an easy800 standard without an integrated display. The device therefore starts immediately after startup and processes the circuit diagram in the device if a memory card is not fitted.

If a memory card is fitted in the device, the program is transferred from the memory card to the easy800 standard after the supply voltage is switched on. The device will then process the circuit diagram in RUN mode.

With an easy800-SWD, the device determines the actual configuration of the SWD line after power up. If the actual configuration, target configuration and project configuration are identical, the device operates in RUN mode.

3.2 Setting the menu language

When you switch on the easy800 device for the first time, you will be asked to select the menu language.



```
ENGLISH      √
DEUTSCH
ITALIANO
ESPANOL
```

- ▶ Use the cursor buttons ⌃ or ⌄ to select the language required.
- ▶ Press OK to confirm your choice and press ESC to exit the menu.

The device will then switch to the status display.

The menu language setting is a easy800 device function. Even when the circuit diagram is deleted this does not result in the loss of the setting selected.

The languages available to you are shown in → Chapter 12, “Appendix“, page 379.

→ You can change the language setting at a later time, (→ Section “Changing the menu language“, Page 311).
If you do not set the language, the easy800-device will display this menu every time you switch on and wait for you to select a language.

3.3 LED display

The following LEDs are provided on the front of the easy800 depending on device type:

1. POW/RUN
2. NET
3. Config (only easy800-SWD)
4. SWD (only easy800-SWD).

LED POW/RUN

The POW/RUN LED indicates the state of the POW power supply as well as the RUN or STOP operating mode.

LED off	No supply voltage
LED green continuous light	STOP operating mode, supply voltage present
LED green flashing	RUN operating mode, supply voltage present.

LED NET (easyNet)

The NET LED provides information on the status of the Net (→ Section “Functionality of the NET faulty“, Page 337).

LED off	NET not operational, fault in configuration.
LED green continuous light	Fault! Net is initialized but at least one station has not been detected.
LED green flashing	NET operating fault-free

Config. LED

The Config LED indicates that the project configuration and target configuration of the device match.

Table 7: Config.-LED (only easy800-SWD)

LED off	no project configuration present, because e.g. • Initial Commissioning or • Program in easy806-SWD was deleted
LED green continuous light	Target configuration = Project Configuration
LED red continuous light	Target configuration ≠ Project Configuration No SWD line present

LED SWD (SmartWire-DT)

The SWD LED indicates whether SmartWire-DT is operating error-free.

Table 8: LED SWD (only easy800-SWD)

LED off	no supply voltage at POW no target configuration present
LED green continuous light	Actual configuration = Target configuration
LED green flashing	Actual configuration = Target configuration during addressing

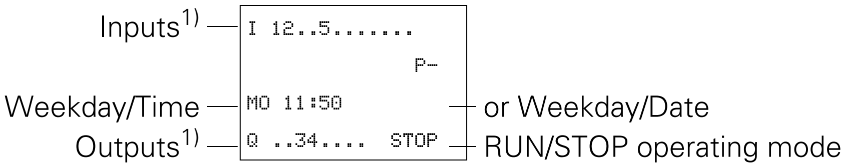
LED orange flashing	Target configuration currently being determined
LED red flashing	One station too many External device plug empty - station missing Target configuration ≠ Actual configuration
LED red continuous light	No connection between easy800-SWD and the SWD line Short-circuit of the 15 V DC device voltage

3.4 Status display

Status display easy800

After power on, the easy800 shows the status display of the basic device. The status display has four lines. If a local expansion is connected to easy800 standard, press OK to show the status display of the local expansion in the display (→ section “Status display additional information”, page 74).

Each dot of the status display represents the status of the corresponding input, output, ID etc. If the operand number is indicated, the status is 1; if a dot is indicated, the status is 0.

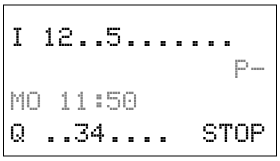


```
I 12..5.......
             P-
MO 11:50
Q ..34....  STOP
```

1) On: I1, 2, 5 and Q3, 4
Off:...

Figure 66:Status display when you switch on

Lines 1 and 4: Inputs/outputs



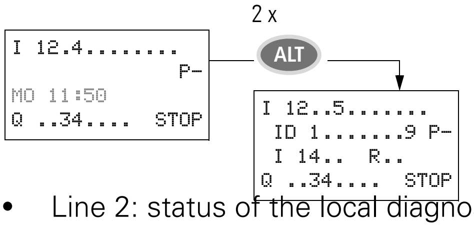
```
I 12..5.......
             P-
MO 11:50
Q ..34....  STOP
```

Line 1 shows the status of the basic device inputs. Line 4 shows the basic device outputs and its operating mode (RUN, STOP or BUSY).
→ Section “easy800 operating principle in the circuit diagram and function block editor“, Page 75.

Line 2: System information, diagnostics

You can change the content of line 2 (and 3) by pressing ALT. After the initial power up, the easy800 device indicates with P- that the cursor buttons (P buttons) are not active and therefore cannot be used as button inputs in the standard circuit diagram. Active P buttons are indicated with P+ (→ Section “P PUSHBUTTONS“, Page 303). The will show further system information easy800in line 2 (→ Section “Status display additional information“, Page 74) depending on the parameters and the peripheral devices connected.

Starting from the initial status display, you can press ALT twice to show the following diagnostics messages in line 2 and 3:



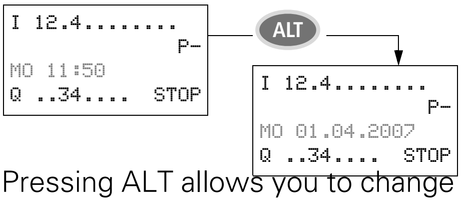
```
I 12.4........
            P-
MO 11:50
Q ..34....  STOP
```

```
I 12..5.......
 ID 1.......9 P-
 I 14..  R..
Q ..34....  STOP
```

- Line 2: status of the local diagnostics bits ID1...ID9 and P-/+.
- Line 3: Status of the local diagnostics bit I14...I16 as well as R15 and R16.

Line 3: Weekday, time/date, diagnostics

In the initial status display, line 3 shows the weekday and the time.



```
I 12.4........
            P-
MO 11:50
Q ..34....  STOP
```

```
I 12.4........
            P-
MO 01.04.2007
Q ..34....  STOP
```

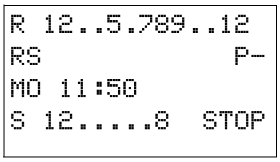
Pressing ALT allows you to change the content of line 2 and 3 and the status display of the basic device as well as the local expansion.

→ When operating the easy800 as a NET station you can press ESC to switch from the status display and select another NET station. You can then display the status of the inputs and outputs of the other NET station (→ Section “Showing the status display of other NET stations“, Page 97).

Status display of the local expansion

Press OK to move from the Status display for the easy800 standard to the Status display for the local expansion unit if this is being operated via the easyLink connection. Pressing ALT allows you to switch the content of line 2 and 3 in the Status display.

Lines 1 and 4: Inputs/outputs



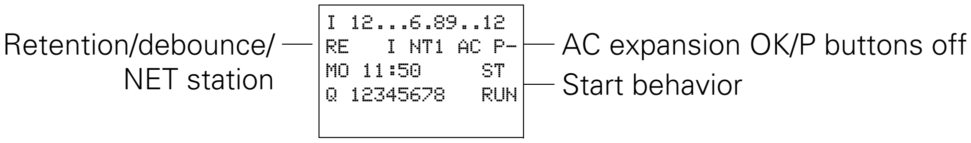
```
R 12..5.789..12
RS          P-
MO 11:50
S 12.....8  STOP
```

Line 1 shows the status of inputs R of the local expansion station and line 4 that of outputs S. In addition line 4 indicates the basic devices operating mode.

→ Apart from the ID RS in line 2, by which a connected functional expansion unit is indicated, lines 2 and 3 indicate the same contents for basic and expansion units.
Press ALT once therefore to show in line 2 the weekday and the date, and press ALT twice to show also in line 2 and 3 the same diagnostic alarms as in the status display for the easy800 standard.

Status display additional information

Depending on the additional parameters set (e.g. retentive operands) or the connected peripheral devices (e.g. when running as an Net station) the system information listed below is also shown in the status display.



```
I 12...6.89..12
RE   I NT1 AC P-
MO 11:50     ST
Q 12345678   RUN
```

System Information

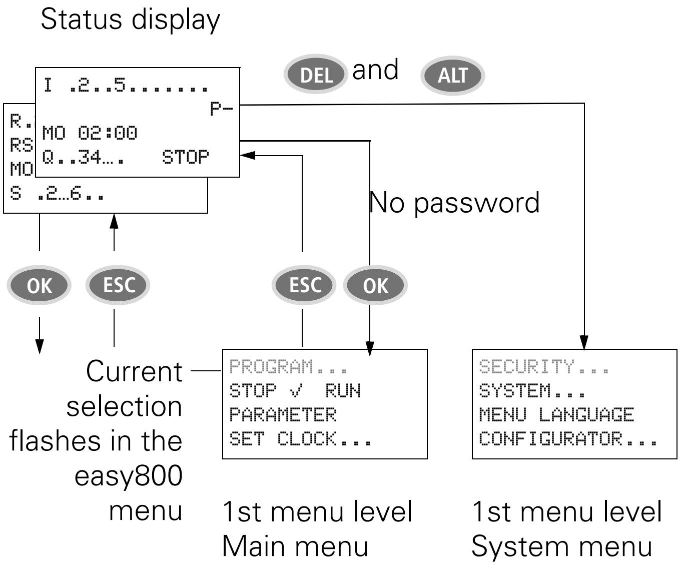
RE	:	Retention active or RS: Local expansion station operating correctly
I	:	Debounce switched on
NT1	:	NET stations with Net-ID (1 this case)
AC	:	AC expansion functioning correctly or
DC	:	DC expansion functioning correctly
GW	:	Expansion of part no. Gateway detected, that is used for communication between other bus systems or for processing analog values and temperatures. GW flashes: Only EASY200-EASY, but no connected I/O expansion.
ST	:	When the power supply is switched on, the easy800 switches to STOP operating mode

How to show the status display of other NET stations is described in → Section “Showing the status display of other NET stations“, Page 97)

3.4.1 Choosing the main and system menu

You can move to several submenus starting from the Status display. The complete menu structure is shown in

Only the menus of the first menu level are shown as an overview: the main menu and the system menu. The device menu is not important for operation



```
I .2..5.......
             P-
MO 02:00
Q..34….   STOP
```

```
R.
RS
MO
S .2…6..
```

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

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

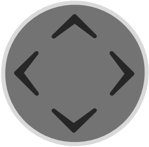
Figure 67:Entering the main or system menu

3.5 easy800 operating principle in the circuit diagram and function block editor

Pushbutton for editing circuit diagrams and function blocks

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

ALT Toggle between N/O and N/C contact
Connect contacts, relays and rungs
Add rungs



ʌ v Change value
Move cursor up/down
< > Change place
Cursor left/right

Cursor buttons set as P pushbuttons:

<	Input P1,	ʌ	Input P2
>	Input P3,	v	Input P4

ESC Undo setting from last OK
Leave current display, menu

OK Change, add new contact/relay,
Save setting

Operating principle in the easy800 circuit diagram

The cursor buttons in the easy800 circuit diagram have three functions:

- Moving.
- Enter.
- Connecting.

The current mode is indicated by the appearance of the flashing cursor.

▦ In “Move” mode use the ʌ v < > buttons to position the cursor on the circuit diagram in order to select a rung, a contact, a relay coil or the selection position of a coil function or a NET-ID.

I 01 Use OK to switch to Entry mode so that you can enter or change a value at the current cursor position. Press ESC in Entry mode to restore easy800 to the last changes of an entry.

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

Press ESC to exit the circuit diagram and parameter display.

→ The easy800 device performs a large part of this cursor change automatically. For example it switches the cursor to Move mode when it is no longer possible to make an entry or connection at the selected cursor position.

3.6 Defining easy800-SWD inputs/outputs

EASY806-DC-SWD provides four local inputs. This is the factory setting. Inputs I3 and I4 can be operated alternatively as outputs Q1 and Q2. A display and operating facility is required to reconfigure the inputs as outputs (→ section "Connecting example easy800-SWD to MFD-CP4", page 64). Proceed as follows to define the inputs and outputs:

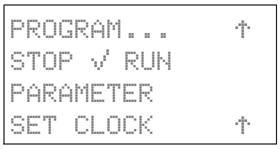
▶ Simultaneously press the DEL and ALT buttons with the Status display active.
▶ Use the cursor buttons to select the System menu item.
▶ Confirm with OK.
▶ Use the cursor buttons to select the ACTIVATE Q1 menu item.
▶ Confirm with OK.
▶ Use the cursor buttons to select the ACTIVATE Q2 menu item.
▶ Confirm with OK.

A tick appears behind the ACTIVATE Q1 menu item, indicating that I3 is configured as an output. The same applies to Q2. You can now use the outputs in the circuit diagram.

→ Be aware that certain function blocks may not be available with certain configurations. For example, output Q2 is permanently associated with PO01. If Q2 is configured as an input, PO01 is no longer available.

3.7 easyoperating modes

A easy800 device recognizes operating modes RUN, STOP and BUSY.



```
PROGRAM...     ↑
STOP √ RUN
PARAMETER
SET CLOCK      ↑
```

RUN

In RUN operating mode the stored circuit diagram is run continuously until STOP is selected or the power supply is switched off. The outputs are activated according to the switch logic of the circuit diagram.
The circuit diagram and the parameters 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. In RUN operating mode:

- the process image of the inputs are read.
- circuit diagram is processed.
- the process image of the outputs is transferred to the physical outputs.

DANGER

Configure your machine/plant so that the automatic starting of the easy800 device never causes unintentional starting of the machine/plant concerned.
Create your circuit diagram so that a defined and safe startup procedure is always provided after the power supply is switched on.

The easy800 standard device with an integrated display does not start with RUN mode if you deactivate the RUN MODE startup behavior.

easy800 devices without display and function keys have a different startup behavior. The RUN START and CARD START functions are activated automatically in this case, since no device operations for manual starting are possible. More information on the CARD START function is provided on → section “CARD MODE”, page 305 → section “Behaviour during upload/ download to card or PC”, page 305.

The switch from RUN to STOP and vice versa is carried out in the main menu.

STOP

In STOP operating mode, the circuit diagram is not executed. Only in this mode is circuit diagram entry, modification of the system parameters or configuration of the easyNet possible. It is also possible to store the circuit diagram on the external memory card or load them from the external memory card.

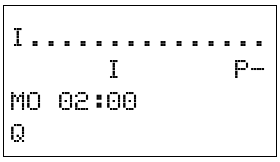
3.8 The first circuit diagram

The following provides a step by step example of how to wire your first circuit diagram. In this way you will learn all the rules, quickly enabling you to use an easy800 device for your own projects.

As with conventional wiring, you use contacts and relays in the diagram.

The easy800 device saves you having to use and wire different components. At the push of a few buttons, the easy800 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.

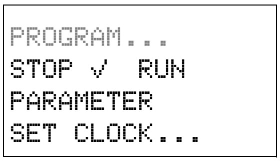
3.8.1 Switching to the circuit diagram



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

Requirements for circuit diagram entry:

The easy800 device is not in STOP mode. They are in the Status display.

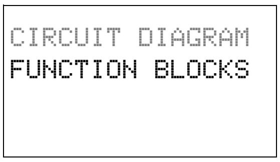


```
PROGRAM...
STOP √  RUN
PARAMETER
SET CLOCK...
```

▶ Press OK to move from the Status display to the main menu.

The PROGRAM menu option must then be selected on the easy800 device.

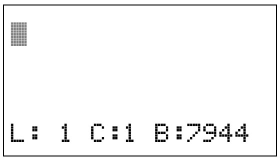
In general, press OK to switch to the next menu level, and press ESC to move one level back.



```
CIRCUIT DIAGRAM
FUNCTION BLOCKS
```

▶ Press OK twice to enter the circuit diagram display via menu options PROGRAM… -> CIRCUIT DIAGRAM. This is where you will create the circuit diagram.

Circuit diagram display



```
▒


L: 1 C:1 B:7944
```

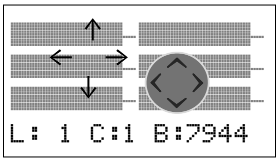
The content of the circuit diagram is displayed in the first three lines.
This window can be moved over the circuit diagram. At the moment the circuit diagram is empty. The cursor flashes at the top left, which is where you will start to wire your circuit diagram.

The last line indicates the position of the cursor:

- L: = Rung (**L**ine)
- C: = Contact or coil field (**C**olumn)
- B: = Amount of free memory in **B**ytes

3.8.2 Draw a wiring diagram

The circuit diagram supports four contacts and one coil in series. The display shows 6 fields of the circuit diagram.



```
L: 1 C:1 B:7944
```

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. easy800 automatically connects the contact to the power supply.

The following example is provided for a lighting control. The easy800 device takes on the wiring and the tasks of the circuit shown below.

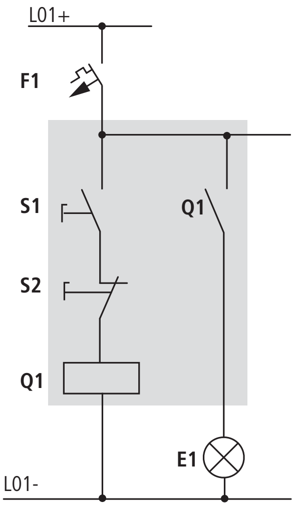


Figure 68:Lamp controller with relays

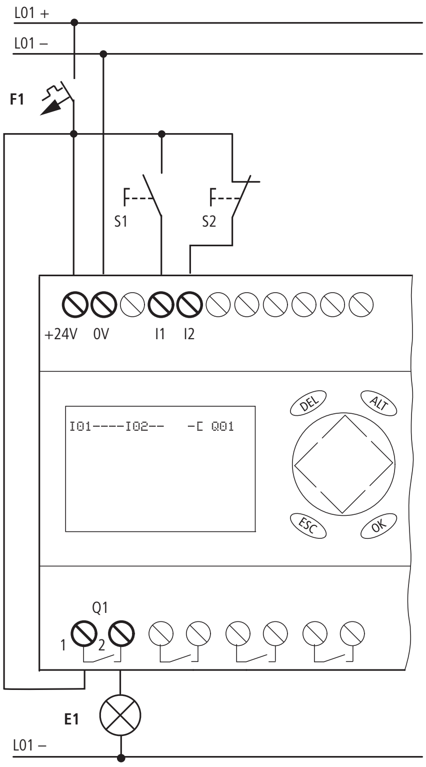
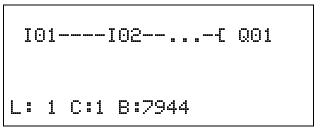


Figure 69:Lamp control with an easy800 standard

▶ Now wire the circuit diagram as described below.



```
I01----I02--...-[ Q01

L: 1 C:1 B:7944
```

Figure 70:Circuit diagram with inputs I01, I02 and output Q1

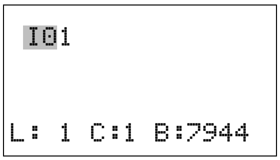
With this example the switches S1 and S2 are at the input. I01 and I02 are the switch contacts for the input terminals in the circuit diagram.

The relay Q1 is represented in the circuit diagram by the relay coil -[Q01.

The symbol -[identifies the coil's function, in this case a relay coil acting as a contactor. Q01 is one of the easy800device's outputs.

From the first contact to the output coil

With easy800 devices wire from the input to the output. The first input contact is I01.

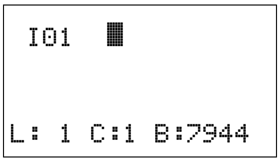


```
I01

L: 1 C:1 B:7944
```

- ▶ Press OK.

easy800 proposes the first contact I01 at the cursor position.

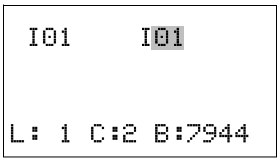
I flashes and can be changed, for example, to a P for a pushbutton input using the cursor buttons ⌃ or ⌄. However, nothing needs to be changed at this point.



```
I01  ▓

L: 1 C:1 B:7944
```

- ▶ Press OK twice, 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.



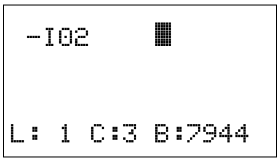
```
I01     I01

L: 1 C:2 B:7944
```

- ▶ Press OK.

Again, inserts easy800 a contact I01 at the cursor position.

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

→ Press DEL to delete a contact at the cursor position.



```
-I02    ▓

L: 1 C:3 B:7944
```

- ▶ 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.

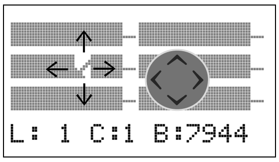
Wiring

easy800 device 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. Pressing ALT once more switches the cursor back to Move mode.

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

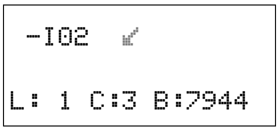
- In the left contact field, you can press ALT to insert a new empty circuit connection.
- The contact under the cursor can be changed between a N/O and N/C contact by pressing the ALT button.



```
L: 1 C:1 B:7944
```

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 if required.

→ The easy800 device automatically connects adjacent contacts up to the coil.



```
 -I02   ↙

L: 1 C:3 B:7944
```

- ▶ Press ALT to wire the cursor from I02 through to the coil field.

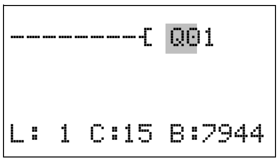
The cursor changes into a flashing wiring arrow and automatically jumps to the next logical wiring position.

- ▶ Press the cursor button >. Contact I02 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.

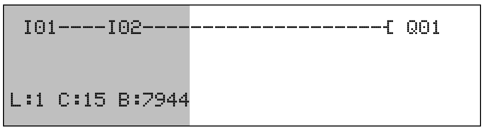


```
---------[ Q01


L: 1 C:15 B:7944
```

- ▶ Press OK.

The specified coil function [and the output relay Q01 are correct and do not have to be changed.

Your result will look as follows:



```
 I01----I02---------------------[ Q01


L:1 C:15 B:7944
```

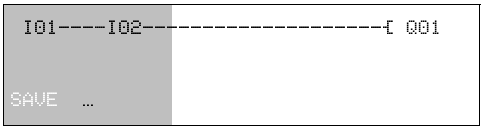
Figure 71:Your first wired and functional circuit diagram

▨ = Visible area

- ▶ Press ESC to leave the circuit diagram display.

The SAVE menu appears.

Saving



```
 I01----I02---------------------[ Q01


SAVE  …
```

Figure 72:SAVE menu

▨ = Visible area

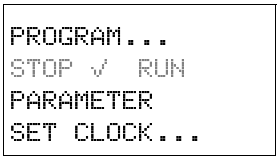
- ▶ Confirm with OK.

The circuit diagram is stored.

- ▶ Press the ESC pushbutton twice to return to the main menu.

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

3.8.3 Testing the circuit diagram



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

▶ Switch back to the main menu and select the STOP RUN menu option. The current operating mode is indicated in the device display by a tick at RUN or STOP stop. Pressing the OK pushbutton enables you to toggle between the modes.

▶ Press the OK pushbutton in order to change to RUN operating mode.

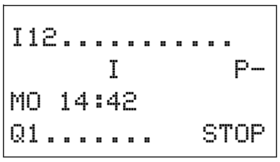
→ The Status display also shows the current mode set.

3.8.4 Control options in RUN mode

There are two control options in RUN mode. Control of:

1. Inputs and outputs with Status display
2. power flow with power flow display

Status display during RUN mode



```
I12...........
       I      P-
MO 14:42
Q1.......   STOP
```

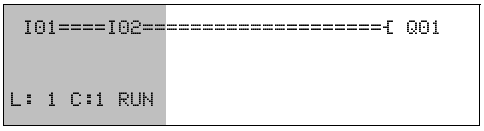
▶ Change to the status display and press pushbutton S1. Do not execute pushbutton S2.

The contacts for inputs I01 and I02 are activated and relay QS1 picks up, indicated by the highlighted numbers.

Test using the power flow display

▶ Change to the circuit diagram display and press pushbutton S1.

The relay picks up and the easy800 displays the power flow with a double line.



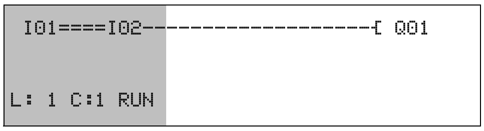
```
 I01====I02====================[ Q01


L: 1 C:1 RUN
```

Figure 73:Power flow display: Inputs I01 and I02 are closed, relay Q1 has picked up

▨ = Visible area

▶ Press pushbutton actuator S2, that has been connected as the N/C contact.

The power flow is interrupted and relay Q1 drops out.



```
 I01====I02--------------------[ Q01

L: 1 C:1 RUN
```

Figure 74:Power flow display: Input I01 is closed, input I02 is open, relay Q1 has dropped out

▨ = Visible area

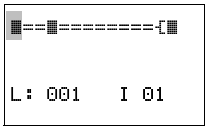
- ▶ Press ESC to return to the Status display.

→ The circuit diagram does not have to be completed in its entirety so that it is possible to test parts of it. The easy800 device simply ignores any incomplete wiring that is not yet working and only uses the finished wiring.

Power flow display with Zoom function

A reduced display of the circuit diagram is possible for a better overview. Proceed as follows:

- ▶ Change to the circuit diagram display and press the ALT button. The circuit diagram display is reduced.
- ▶ Press pushbutton S1.



```
■==■=========[■

L: 001   I 01
```

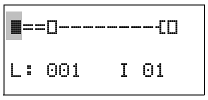
Figure 75:Power flow display in Zoom function: Input I01 and I02 are closed, relay Q1 picked up

■ Contact closed, coil is triggered:

□ Contact opened, coil not triggered.

- ▶ Press pushbutton S2, that has been connected as an N/C contact.

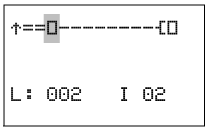
The power flow is interrupted and relay Q1 drops out.



```
■==□---------[□
L: 001   I 01
```

Use the cursor buttons ∧ ∨ < > to move between the contacts or coil.

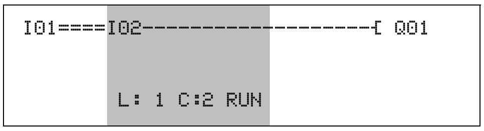
- ▶ Press the cursor button >.



```
↑==□---------[□

L: 002   I 02
```

The cursor moves to the second contact.

- ▶ Press the ALT button. The Zoom function is switched off. The display switches to the display state with contact and/or coil designation.



```
I01====I02-------------------[ Q01

       L: 1 C:2 RUN
```

Figure 76:Power flow display: Input I01 is closed, input I02 is open, relay Q1 has dropped out

▨ = Visible area

3.8.5 Delete Program

The DELETE PROGRAM function not only deletes the circuit diagram but all elements of a program. These are as follows:

- Circuit Diagram
- Function block list
- Function block plan
- Project configuration (applies only to easy800-SWD)
- Assignment list (applies only to easy800-SWD)
- SWD parameters (applies only to easy800-SWD)

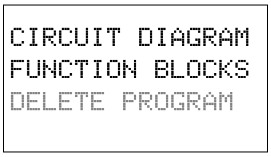
The system settings and operating parameters are reset to the default settings and also a possible NET parameterization.

Proceed as follows to delete the program in the easy800 device:

▶ easy800 must be in STOP mode in order to extend, delete or modify the circuit diagram. Switch the easy800 device to STOP mode.

▶ Choose STANDARD -> PROGRAM… to switch from the main menu to the appropriate menu level.

▶ Select DELETE PROGRAM



```
CIRCUIT DIAGRAM
FUNCTION BLOCKS
DELETE PROGRAM
```

The easy800 device will display the prompt DELETE?.

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

▶ Pressing ESC once more returns you to the previous menu level.

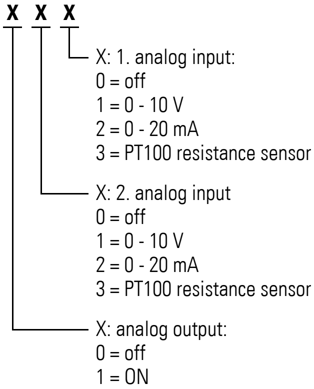
3.9 Configuring the analog expansion unit

The EASY4..-DC-ME expansion units for analog value and temperature measuring must be configured before commissioning. The digital inputs/ outputs of these expansion units are detected by the easy800 standard automatically - without any configuration.

The analog expansion units operate with connected Pt100 sensors in a range between -50 °C and 200 °C.

By entering a configuration code in easy800 standard you can activate or deactivate the analog inputs and outputs of the expansion unit separately. The configuration code consists of the following:

EASY406-DC-ME



x x x

X: 1. analog input:
0 = off
1 = 0 - 10 V
2 = 0 - 20 mA
3 = PT100 resistance sensor

X: 2. analog input
0 = off
1 = 0 - 10 V
2 = 0 - 20 mA
3 = PT100 resistance sensor

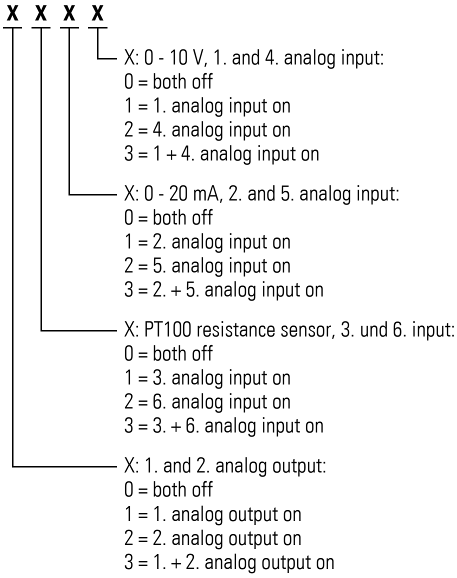
X: analog output:
0 = off
1 = ON

Figure 77:Configuration code EASY406-DC-ME (three-digit)

Factory setting = 111
(analog input 1 and 2 are activated, both for 0 - 10 V as well as the analog output). Marker words MW93 - MW95 are therefore assigned (→ Table 9, Page 87).

With EASY406-DC-ME you can activate and commission a voltage, current or temperature input for analog value 1 and 2.

EASY411-DC-ME



x x x x

X: 0 - 10 V, 1. and 4. analog input:
0 = both off
1 = 1. analog input on
2 = 4. analog input on
3 = 1 + 4. analog input on

X: 0 - 20 mA, 2. and 5. analog input:
0 = both off
1 = 2. analog input on
2 = 5. analog input on
3 = 2. + 5. analog input on

X: PT100 resistance sensor, 3. und 6. input:
0 = both off
1 = 3. analog input on
2 = 6. analog input on
3 = 3. + 6. analog input on

X: 1. and 2. analog output:
0 = both off
1 = 1. analog output on
2 = 2. analog output on
3 = 1. + 2. analog output on

Figure 78:Configuration code EASY411-DC-ME (four-digit)

Factory setting = 3333
(analog input 1 to 6 as well as analog output 1 and 2).
Marker words MW89 - MW96 are therefore assigned (→ Table 9, Page 87).

With EASY411-DC-ME you can activate and commission all six analog inputs (voltage, current or temperature) at the same time.

3.9.1 Setting device parameters for the expansion unit

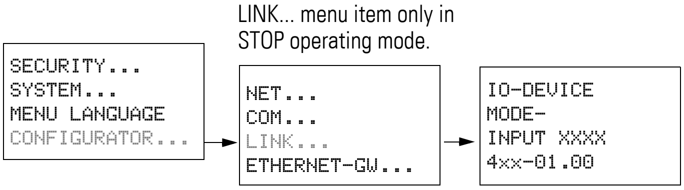
Carry out the configuration settings for an analog expansion unit in the IO-DEVICE MODE- INPUT XXXX menu:

Basic requirements:

- easy800 standard and the expansion unit are connected via the EASY-LINK-DS connector.
- Both devices must be connected to the power supply and mounted.
- easy800 standard must be in STOP mode and the Status display must be showing.

They are processed in the program using marker words, → Table 9 Page 87.

→ The EASY4..-DC-ME expansion units can be operated on easy800 standard from device version 02.



LINK... menu item only in STOP operating mode.

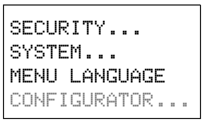
```
SECURITY...
SYSTEM...
MENU LANGUAGE
CONFIGURATOR...
```

```
NET...
COM...
LINK...
ETHERNET-GW...
```

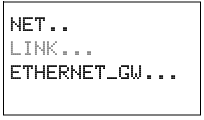
```
IO-DEVICE
MODE-
INPUT XXXX
4xx-01.00
```

▶ Simultaneously press the DEL and ALT buttons with the status display active.

▶ Select the CONFIGURATOR menu option and press the OK pushbutton.



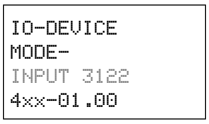
```
SECURITY...
SYSTEM...
MENU LANGUAGE
CONFIGURATOR...
```



```
NET..
LINK...
ETHERNET_GW...
```

▶ Now choose the LINK menu item and press the OK button.

The INPUT XXXX or XXX menu item is selected and the lowest parameter digit flashes.



```
IO-DEVICE
MODE-
INPUT 3122
4xx-01.00
```

▶ Use the cursor buttons ‹ › to select the digits to be modified, and then use the ∧ ∨ buttons to set the required value. Set the configuration code accordingly (→ Figure 77, page 85 and → Figure 78, page 85.

▶ After you have carried out the necessary parameter settings, confirm with the OK pushbutton.

The device switches back to the Status display.

Pressing the ESC pushbutton returns you to the Status display without changing the parameters.

→ If you select invalid device parameters, e.g. 3334 on EASY411-DC-ME, and leave the menu with OK, the device carries out an internal check and then re-opens INPUT XXXX. It proposes the last valid value as the device parameter, which you can now change or confirm with OK.

→ The parameters configured here are stored retentively and only in the expansion unit, not in the easy800 standard.
This means: You must complete this configuration again when exchanging a configurable expansion unit.

Processing analog input/output values

You access the now configured analog input/output values via marker words (MW). The process image is provided in the marker words.

Marker configuration with operation on easy800 standard (from device version 02)

Table 9: Marker configuration using analog input/output values

Analog value Input/Output	EASY406-DC-ME	EASY411-DC-ME
1. Analog input	MW93	MW89
2. Analog input	MW94	MW91
3. Analog input	–	MW93
4. Analog input	–	MW90
5. Analog input	–	MW92
6. Analog input	–	MW94
1. Analog output	MW95	MW95
2. Analog output	–	MW96

→ A new analog expansion unit is factory set for all inputs/outputs to be activated. All the assigned marker words are thus also taken.
For your expansion unit only configure the analog inputs/outputs that are actually required.
This has several advantages:

- More free marker words for use in the program or in the visualization.
- Only relevant error messages, as otherwise an activated but unconnected input will signal a fault such as a wire breakage.
- Shorter reaction times due to the faster data exchange between easy800 standard and expansion unit.

Example configuration with EASY411-DC-ME

This analog expansion unit is required to read a voltage 0 - 10 V via analog input 1 and output a voltage 0 - 10 V via analog output 1.

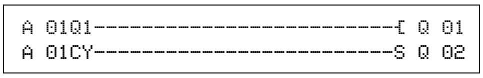
→ Resulting parameter setting for expansion device: 1001

The marker words MW89 (analog input 1) and MW95 (analog output 1) are used for this application.

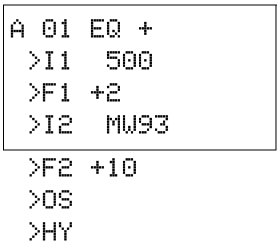
Example configuration with EASY406-DC-ME

Two analog values are to be monitored for equal voltage.

The voltage at analog input 1 of the expansion device (terminal R2) is required to be compared with a constant and the "greater than/equal to" result is to be monitored. The "greater than/equal to" result is to be indicated by the state 1 at output Q1. Analog input 2 and the analog output of the expansion unit are deactivated.



```
A 01Q1--------------------------[ Q 01
A 01CY--------------------------S Q 02
```



```
A 01 EQ +
 >I1  500
 >F1 +2
 >I2  MW93
 >F2 +10
 >OS
 >HY
```

Parameter setting of the "analog value comparison" function block

Wiring digital inputs/outputs

The digital inputs R1 - R3 and digital outputs S1/S2:

- do not have to be activated in the configuration.
- can always be wired in the circuit diagram.
- can be used parallel to every configured analog input/output

Digital inputs R1 - R3

The digital inputs R1 - R3 of the analog expansion inputs are used in exactly the same way as the digital expansion units.

The state of the digital input R1 is evaluated by the hardware. To use the digital inputs R2/R3 you have connected the two analog 0 - 10 V inputs (terminal 2 and 7). Their state is evaluated by the software. The expansion device outputs a 0 signal on R2/R3 if the voltage goes below 8 V DC or outputs a 1 if this voltage is exceeded.

Digital outputs S1, S2

The digital outputs S1 and S2 of the analog expansion units are used in exactly the same way as the digital expansion units.

Diagnostics of an analog expansion unit

Table 10: Group fault messages of the analog expansion units

Diagnostics contact **Fault on 1 state**	**Meaning of group message** **(with identical measuring processes, the same fault can be present at two analog inputs or outputs).**
R13	Wire breakage on at least one analog input for measuring 4 - 20 mA current. With current measuring activated the fault message can only appear if a current < 3 mA is present.
R14	Wire breakage on at least one PT100 input for temperature measuring. The fault message can only appear with PT100 measuring activated.
R15	Short-circuit at analog output 0 - 10 V.
R16	Short-circuit at one or more digital outputs.

3.10 Commissioning an easyNet network

The following description uses the abbreviated designation NET for easyNet. EASY802-DC-SWD cannot be connected to the NET.

Requirements for commissioning the NET:

- You have selected a type of cable routing and connected all stations properly to the NET (→ Section 2.7, “NET easynetwork“, Page 53)
- The terminating resistors at the physically first and last NET station have been fitted.
- The supply voltage is connected on all NET stations.
- The POW LED of all NET stations must be lit or flashing.
- As the first NET station, use a device with a display.
- The device to be parameterized must be in STOP operating mode. The POW LED of this NET station must be permanently lit.

When commissioning for the first time, the NET LED is “Off”.

An introduction to the easyNet network is provided in → section “Introduction network easyNet“, page 273.

→ Your NET can only be configured by the station with NET-ID 1 if it has been set up according to the “passing through” (→ Page 53) cable routing. In order to simplify the process, you should define all parameters through this NET station with NET-ID 1.

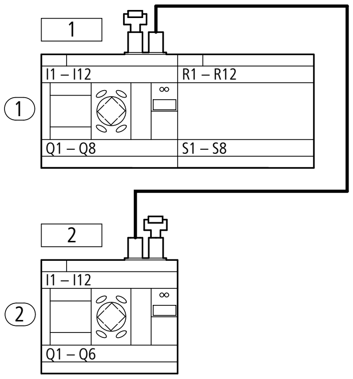


Figure 79:Example cabling with two NET stations

① Bus termination resistor

▭ physical location ⬭ station number

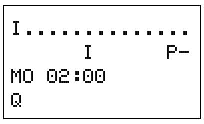
→ Start the parameterisation with the physically first device on the NET, the station with NET ID 1. This NET station can be used to configure the entire NET.

You should only configure the individual device locally in the following cases:

- When exchanging devices
- When connecting an EC4P to easy800 standard

3.10.1 Parameterizing NET stations

Requirement: The easy800 device must be in STOP operating mode.

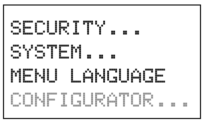


```
I.............
      I      P-
MO 02:00
Q
```

Figure 80:

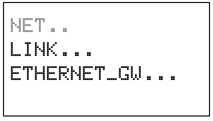
The Status display is active and you move as follows to the NET Parameters menu:

- ▶ Simultaneously press the DEL and ALT buttons.



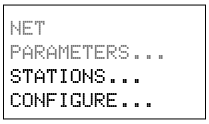
```
SECURITY...
SYSTEM...
MENU LANGUAGE
CONFIGURATOR...
```

- ▶ Select the CONFIGURATOR menu option and press the OK pushbutton.



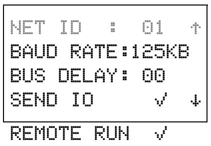
```
NET..
LINK...
ETHERNET_GW...
```

- ▶ Select the NET menu option and press the OK pushbutton.



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

- ▶ Select the NET PARAMETERS menu option and press the OK button.



```
NET ID  :  01  ↑
BAUD RATE:125KB
BUS DELAY: 00
SEND IO     √  ↓
REMOTE RUN  √
```

- ▶ Use the ⌃ and ⌄ buttons to assign the NET ID (in this case: NET ID 01) in the NET PARAMETERS menu and confirm the setting with OK.
- ▶ Switch to Baud rate and set here the highest possible value that your NET allows (→ section “NET easynetwork”, page 359).

The remaining parameters are not described further here. They are described in detail (→ section “Description of the NET-PARAMETERS”, page 277).

- ▶ If all settings are ok, press ESC to exit the menu.

→ The functions REMOTE RUN and REMOTE IO are not required for the station with NET ID 1 and are therefore not available here.

You have parameterized the station with NET-ID 1.

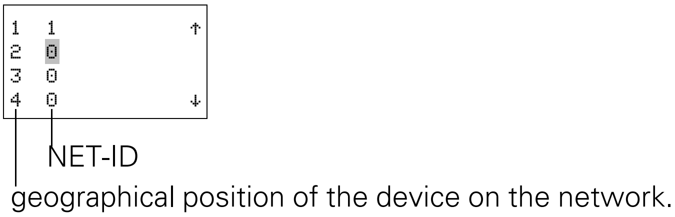
If you have chosen a loop-through cable routing, you can now create the NET station list in the STATIONS menu, and then assign the NET-IDs via the CONFIGURE menu item, and set a standard baud rate and bus delay time.

If you have chosen spur line cable routing, repeat the steps of this chapter for each further station of the NET.

3.10.1.1 Creating the station list

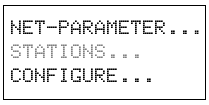
The station list is only important when selecting Loop-through cable routing. The left-hand column shows the geographical position of the device on the network. Use the number in the right-hand column to assign a NET-ID to the device at this position. Position 1 is always assigned to NET-ID 1. You can only assign each NET-ID once.
You create the station list in the STATION menu.

Only the physically first station with NET-ID 1 manages a station list.
You can only log in other NET stations via this NET station. The STATIONS menu remains inactive for all other NET-IDs.



```
1  1         ↑
2  0
3  0
4  0         ↓
```

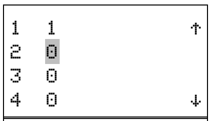
Figure 81:



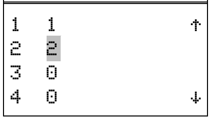
```
NET-PARAMETER...
STATIONS...
CONFIGURE...
```

Requirement: The easy800 device must be in STOP operating mode.
You must be in the NET-PARAMETER menu.

- ▶ Use the ⌃ and ⌄ cursor buttons to select the STATION... menu and press the OK pushbutton.
- ▶ Proceed to the station with physical position 2.



```
1  1         ↑
2  0
3  0
4  0         ↓
```



```
1  1         ↑
2  2
3  0
4  0         ↓
```

- ▶ Select the required physical location with the cursor buttons ⌃ and ⌄, in this case position 2, and press OK.
- ▶ Select the NET ID 2 with the ⌃ and ⌄ cursor buttons and confirm your input with the OK button.

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

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

3.10.1.2 Change station list

You can change the NET configuration at any time via the NET station with the geographical position 1. Existing NET-IDs can only be changed to free NET-IDs that have not yet been assigned.

▶ Move in the station list to the geographical position to which you wish to assign another NET-ID. (→ Section 3.10.1.1, "Creating the station list" on Page 92).
▶ Press the OK pushbutton.

▶ Select the required NET ID with the ⌃ and ⌄ cursor buttons and confirm your input with the OK pushbutton.
▶ Configure NET stations again using the CONFIGURATION menu, → section "Procedure", page 95.

> *CAUTION*
>
> Operating the NET with two masters (two stations with NET-ID 1) is not allowed, and leads to malfunctions on the NET.
> Possible effects of this type of incorrect configuration with two masters competing against each other, and in which one is operated in the STOP operating mode and the other in the RUN operating mode, are as follows:
> NET stations with NET-ID 2 - 8 that should follow the operating mode changes of the NET station with NET-ID 1 automatically during operation (REMOTE RUN parametric programming) keep toggling constantly between the STOP and RUN operating states.
> When integrating a master that is already configured into an existing NET, please make sure that there is no other station with NET-ID 1 on the NET.
> Tip: Carry out a NET configuration through NET station 1 after every change to the NET.

If your NET configuration does not function as required you will find a description of possible causes in → section "Functionality of the NET faulty", page 337.

Changing the station list although all NET-IDs are assigned

If all eight NET-IDs are assigned, the NET-IDs that are to be assigned to other geographical positions must be switched to zero. A trick can be used to simplify this:
If the NET-ID of a geographical position is set to 0 in the station list, the NET-IDs of all subsequent positions are likewise set to 0.

Example: NET-ID (= 2) of geographical position 3 is set to 0

physical position	NET-ID
1	1
2	7
3	6
4	4
5	5
6	3
7	2

physical Position	NET-ID
1	1
2	7
3	0
4	0
5	0
6	0
7	0

3.10.2 Configuring NET

Once you have completed the parameter assignment of the NET stations and created a station list, you can then configure the easyNet. For the configuration, all devices connected on the NET are assigned their NET ID in the order in which they were entered in the station list in the STATION menu. Furthermore, all devices are automatically set with the baud rate and bus delay time set on NET station 1.

Carry out this configuration in the CONFIGURE menu.

You can only configure the NET network by entering settings on the NET station 1.

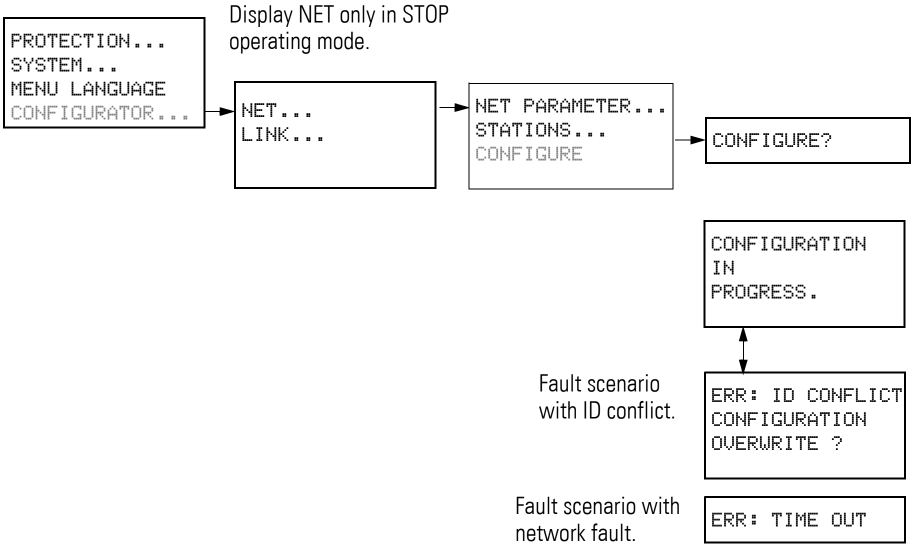


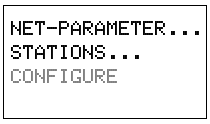
Figure 82:CONFIGURE menu

Requirement for the configuration using NET station 1:

- You must have wired your network as a “passing through” cable routing.
- All stations must be correctly connected to the NET and the termination resistors must have been connected.
- All stations have power, show the basic menu in the device display (the status display) and are in operating mode STOP. The POW and NET LEDs must be permanently lit.

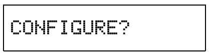
> *CAUTION*
>
> The REMOTE IO, SEND IO and REMOTE RUN parameters required for operation are not transferred to the connected device with the NET configuration.
> If you require a parameter setting that is different to the factory setting, you can do this locally via the NET PARAMETERS menu of the NET station concerned. An easier way is to carry out the parameter setting in easySoft-Pro in conjunction with a program transfer to the device concerned.

Procedure

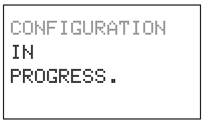


▶ Proceed to the CONFIGURE menu item and press the OK pushbutton.

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



▶ Press OK.



```
CONFIGURATION
IN
PROGRESS.
```

The configuration is successfully completed if the message on the left appears. The NET menu is displayed again and the NET LEDs of all stations flash. The NET is ready for operation.

▶ Press the ESC key to exit the menu.

During the NET configuration, all connected stations switch automatically to STOP. The NET LED for the stations with NET ID 2-8 switches to OFF.

→ Any errors during the configuration are indicated by messages or via the NET LEDs.
Further information can be found in following section and → section "Functionality of the NET faulty", page 337.

3.10.2.1 Possible error messages

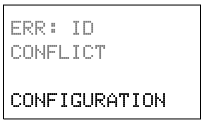
The following messages may be output during the configuration:

Table 11: Possible messages during the configuration

Message	Meaning
ERR: ID CONFLICT CONFIGURATION CONFIGURATION ?	At least two NET stations have the same NET ID → Section , "Error message: ID conflict".
ERR: TIME OUT	Network fault, for example if the cable is not correctly fitted or interrupted. Or you are trying to assign NET ID 1 to a NET station, although a master with NET ID 1 is already present on the NET. Solution: Check the plug connection or the assigned NET IDs and restart the configuration.

Error message: ID conflict

An error message will appear if a station has a NET ID that does not correspond to the physical location in the station list:



```
ERR: ID
CONFLICT

CONFIGURATION
```

If you wish to correct the NET ID (→ section "Change station list", page 93), confirm the prompt with OK. Otherwise you can cancel the configuration with ESC.

3.10.3 Deleting the program of a NET station

Selecting DELETE PROGRAM on a configured NET station will also delete its NET ID.

The device concerned is no longer a NET station, indicated by its switched off LED. The NET LED of all other configured NET stations in operation indicates the NET fault with a "continuous light".

CAUTION
If you delete the program in NET station 1, the NET configuration managed by this station is also deleted.

The project configuration is also deleted with easy800-SWD. The target configuration is retained.

Solution for the accidental deletion of the NET configuration

- With a "loop-through" cable routing.

You must create a new NET configuration via the NET station 1 (→ section "Configuring NET", page 94).

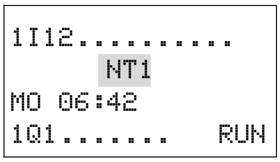
- With a "T-piece and spur line" cable routing.

Locally reassign a NET ID to the device concerned.

3.10.4 Showing the status display of other NET stations

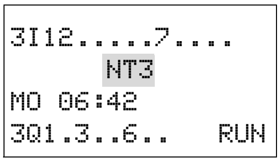
On each device with a program and a display with operating buttons, you display the status of the inputs and outputs of every other NET station.

▶ For this change to the status display and press the ESC button.



```
1I12..........
      NT1
MO 06:42
1Q1.......   RUN
```

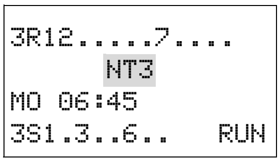
The cursor switches to the display of NET station NT1 and flashes.
The NET ID precedes the display of the input/output ID (e.g. 1I../1Q..).



```
3I12.....7....
      NT3
MO 06:42
3Q1.3..6..   RUN
```

▶ Change the number of the required NET station with the ∧ and ∨ cursor buttons.

The input/output states of this NET station are shown in the display (e.g. 3I../3Q..).



```
3R12.....7....
      NT3
MO 06:45
3S1.3..6..   RUN
```

▶ Press the ALT button several times if you wish to see the locally set time, the date or the group fault alarms for transistor outputs on this NET station.

Press ESC to return to the local display of the NET station - in this case NT1.

3.11 Putting the SmartWire-DT into operation

For a better understanding read chapter „SmartWire-DT“, Page 282, before commissioning SmartWire-DT.

Before switching on, check whether the power supply for the easy800-SWD is connected correctly. The installation of the SWD line must also have been carried out correctly with all stations connected.

> **DANGER**
>
> Unforeseen motor startup or unexpected connection of voltages may cause death. When switching on the device, ensure that the device switches to a controlled and safe operational state.

You must have completed the following steps before commissioning:

1. POW supply voltage is connected
   (→ section "easy800 SWD", page 28).
2. If necessary, connect the AUX supply voltage if contactors are used
   (→ section "easy800 SWD", page 28).
3. SmartWire-DT connected to easy800-SWD
   (→ section "Install SmartWire-DT", page 67).
4. All configured SWD stations are connected to the SWD line.
5. The bus termination must be connected.

Commissioning

▶ Switch on the POW supply voltage on the device.
The status of the LEDs on the device is as follows:
- POW-LED:green continuous light
- SWD-LED:red flashing
- Config. LED: off
- The SWD status LEDs on the SWD stations also flash green.

▶ Press the Config configuration button on easy800-SWD for at least 2 s.
The status of the LEDs on the device is as follows:
- POW-LED:green continuous light
- SWD-LED: orange flashing (during addressing);
  then continuously lit green (after addressing)
- Config. LED: green

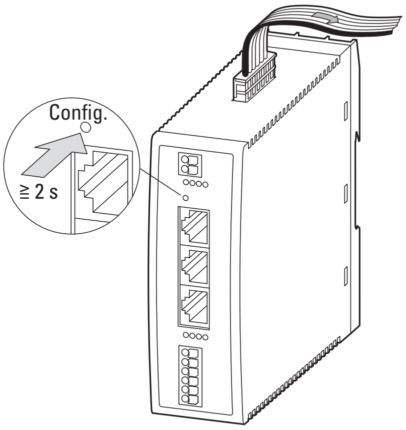


Figure 83:Determining the target configuration of the SWD system

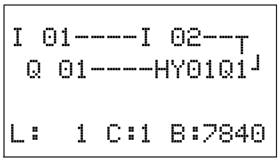
All stations are addressed consecutively in ascending order. The physically present actual configuration is determined on the SWD line. This is stored in easy800-SWD retentively as the target configuration. The SWD-LED indicates with a green continuous light that the target configuration matches the actual configuration.

4 Wiring with easy800

This chapter describes how to wire easy800 contacts and coils with the display and keypad. If you are operating an easy800 without these features, you can connect devices for this task via a point-to-point connection or easyNet display. Refer to → section , "Connecting example easy800 standard to MFD-…-CP8/CP10", page 63 or → section , "Connecting example easy800-SWD to MFD-CP4", page 64 on how to proceed.

Use the easySoft-Pro software to wire up easy800-SWD devices. Make changes to the wiring on the device as described in this chapter.

4.1 Circuit diagram display



```
I 01----I 02--┬
 Q 01----HY01Q1┘

L:  1 C:1 B:7840
```

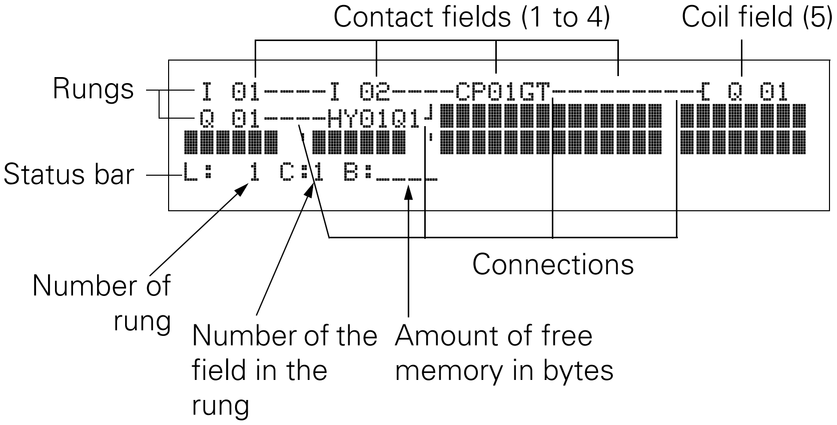
You wire contacts and coils of relays in the easy800 circuit diagram from left to right, from the contact to the coil. The circuit diagram is created on a hidden wiring grid containing contact fields, coil fields and rungs. It is then wired up with connections.

- Insert contacts in the four contact fields. The first contact field on the left is automatically connected to the voltage.
- In the coil field the relay coil to be controlled is entered with the coil identifier and coil function. The coil identifier consists of the coil name, coil number and, in the case of function blocks, with the function identifier. The coil function defines the method of operation of the coil.

The cursor buttons ‹ › can be used to change the contact fields. The number of the circuit connection and the contact are displayed in the lower status line.

There are 256 rungs available in the circuit diagram, for wiring contacts and coils.

For greater legibility the circuit diagram of the easy800 device shows two contacts or one contact plus coil in a row on each rung. A total of 16 characters per circuit connection and three circuit connections plus the status line can be displayed simultaneously.



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

→ The circuit diagram display performs two functions:

- It is used to edit the circuit diagram in STOP mode.
- It is used in RUN mode to check the circuit diagram using the power flow display.

4.2 Program

An easy800 program consists of the necessary settings for the device, easyNet, password, system settings and operating parameters and the following components:

- Circuit Diagram
- Function block list
- Function block plan
- Project configuration (applies only to easy800-SWD)
- Assignment list (applies only to easy800-SWD)
- SWD parameters (applies only to easy800-SWD)

4.3 Loading a program on the easy800-SWD device

When commissioning easy800-SWD for the first time, the program must be transferred from the PC and stored on easy800-SWD. The easySoft-Pro software is required to complete the following steps.

▶ Connect the COM serial interface on easy800-SWD to the serial interface of the PC (→ Chapter 2, "Installation", page 19)
▶ Load the program and thus also the project configuration onto easy800-SWD (see easySoft-Pro Online Help).
▶ Start easy800-SWD by selecting RUN mode.
Select RUN in the main menu on the device.
In easySoft-Pro click the buttons Communication -> Program/Configurator->RUN.

- POW-LED: green flashing
- Config-LED: green continuous light
- SWD-LED: green flashing
- The SWD status LEDs on the two SWD stations are continuously lit green.

If the project configuration matches the target configuration, the device switches to operation and processes its circuit diagram. The actual configuration was already stored as the target configuration when SmartWire-DT was installed and is therefore identical.

4.4 Circuit diagram elements

A circuit diagram is a series of commands that the easy800 device processes cyclically in RUN operating mode.

Coils and contacts are interconnected in the circuit diagram. In RUN mode, a coil is switched on or off depending on the power flow and coil function set.

4.4.1 Function blocks

Function blocks are program blocks with special functions. Examples: Timing relays, time switch, data block comparators. Function blocks are provided with or without contacts and coils. How a safety or standard function block is added to the safety or standard circuit diagram and parameterized is described in → section 4.7, "Working with function blocks", page 127 on.

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

4.4.2 Relays

Relays are switching devices that are emulated in easy800 device electronically, which activate their contacts according to the assigned function. A relay consists of at least one coil and contact.

4.4.3 Contacts

Contacts are used to change the power flow inside the easy800 circuit diagram. Contacts such as N/O contacts are set to 1 when they are closed and 0 when they are opened. In the easy800 circuit diagram you can wire contacts as make or break contacts. N/C contacts are indicated with a horizontal line above the operand concerned.

An easy800 device operates with different contacts that you arrange in any order in the contact fields of the circuit diagram.

Table 12: Usable contacts

Contact	easy800 representation
N/O contact, open in release position	I, Q, M, A, …
N/C contact, closed in release position	$\overline{\text{I}}$, $\overline{\text{S}}$, $\overline{\text{S}}$, $\overline{\text{A}}$, …

A detailed list of all contacts used in the circuit diagram is provided on Page 378.

4.4.4 Coils

Coils are the actuating mechanisms of relays. The results of the wiring are transferred to the coils when the device is in RUN mode. These switch to the On (1) or Off (0) state according to these results. The options for setting output and marker relays are listed with the description of each coil function.

A easy800 device is provided with different types of relays and function blocks which can be wired in a circuit diagram via their coils (inputs).

Coil Functions

The switching of the relays is set by means of coil functions and parameters.

→ If you wish to map coils from your circuit diagram in the easy800 device, use the coils with the Contactor coil function in your device!

The following coil functions are available for all coils:

Table 13: Coil function

easy800 display	Coil function	Example	→ Page
-[	Contactor function	-[Q01,-[D02,-[S04,-[:01,-[M07,..	105
⌠	Impulse relay function	⌠Q03,⌠M04,⌠D08,⌠S07,⌠:01,..	105
S	Set	SQ08,SM02,SD03,SS04..	106
R	Reset	RQ04,RM05,RD07,RS03..	106
]-	Contactor function with negated result	]-Q04,]-M96..	106
⎍	Cycle pulse with rising edge	⎍M01..	107
⎎	Cycle pulse on falling edge	⎎M42..	107

→ With non-retentive coil functions such as -[(contactor),]- (negated contactor), ⎍, ⎎ (rising and falling edge evaluation): Each coil must only be used once. The last coil in the circuit diagram determines the state of the relay. Exception: When working with jumps, the same coil can be used twice. Retentive coil functions such as S, R, ⌠ can be used several times.

The usable coil functions of the function blocks are described in the relevant chapter (→ Chapter 5, “Function blocks“, page 135).

Coil with contactor function -[

The output signal follows the input signal directly, the relay operates as a contactor.

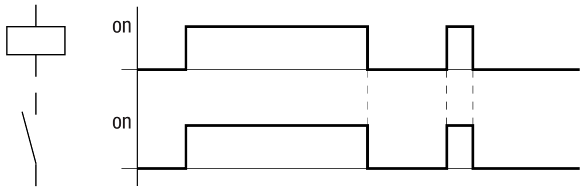


Figure 84:Contactor function signal diagram

Coil with the impulse relay function ⌠

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

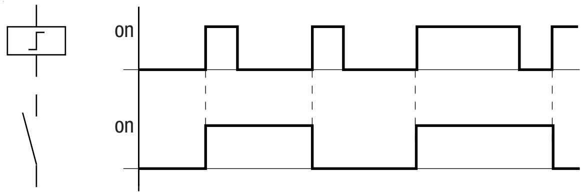


Figure 85:Impulse relay signal diagram

A coil is automatically switched off in the event of a power failure and in STOP mode. Exception: retentive coils retain the 1 state (→ section 8.2.6, "Retentive data storage", page 308).

"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 is not retentive.

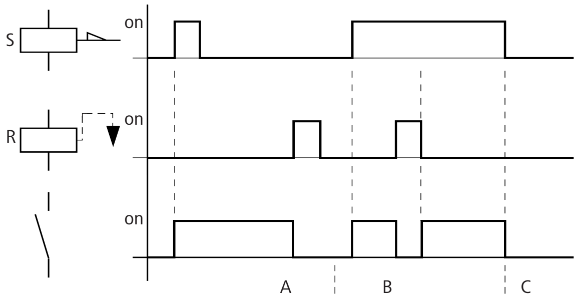
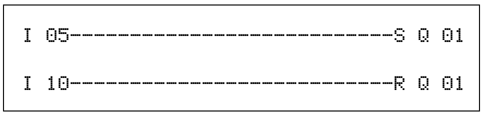


Figure 86:Set and Reset signal diagram

If both coils are actuated simultaneously, the coil with the higher rung number has priority, in this case the reset coil, as shown in the signal diagram at (B).



```
I 05---------------------------S Q 01

I 10---------------------------R Q 01
```

Figure 87:Simultaneous triggering of Q 01

The above example shows the Reset coil with priority when the Set and Reset coil are triggered at the same time.

Coil negation (inverse contactor function) ∤

The output signal follows the inverted input signal, the relay works like a contactor with negated contacts. If the coil is set to 1, the coil will switch its make contacts to 0.

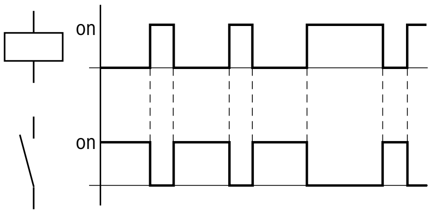


Figure 88:Inverse contactor function signal diagram

Rising edge evaluation (cycle pulse)

This function is used if the coil is only meant to switch on a rising edge. When the coil status changes from 0 to 1, the coil switches its make contacts to 1 for one cycle.

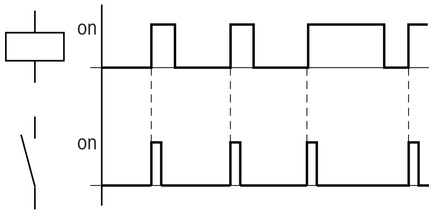


Figure 89:Signal diagram of cycle pulse with rising edge

Falling edge evaluation (cycle pulse)

This function is used if the coil is only meant to switch on a falling edge. When the coil status drops out from 1 to 0, the coil switches its make contacts to 1 for one cycle.

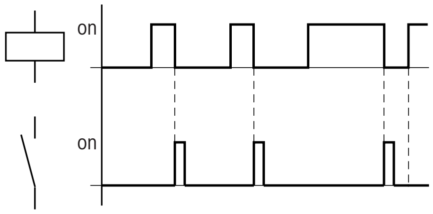


Figure 90:Signal diagram of cycle pulse with negative edge

→ A set coil is automatically switched off in the event of a power failure and in STOP mode. Notes: Retentive coils keep their logic state (→ section 8.2.6, "Retentive data storage", page 308).

4.4.5 Markers and analog operands

The term "marker" is used to represent marker bits (M). Marker bits (M) are used to store the Boolean states 0 or 1. A marker bit is also called an auxiliary relay.

easy800 devices also manage the marker bits in marker bytes (MB), marker words (MW) and in marker double words (MD). A marker byte consists of 8 marker bits, a marker word of 16 marker bits and a marker double word of 32 marker bits.

To store the state of a contact, you can use a specific bit and thus also a specific byte. For example, marker bit 9 is contained in marker byte 2, in marker word 1 and in marker double word 1. The following conversion tables will help you to determine the word in which a bit is contained or the bits of a specific double word.

Bear in mind that after the division, a rounding up to the next higher integer is necessary, even if the decimal number is below 0.5.

Table 14: Conversion table M, MB, MW, DW

	given	Bit	Byte	Word	Double word
		M n	MB n	MW n	DW n
Searched for					
Bit (M)			8n-7 to 8n	16n-15 to 16n	32n-31 up to 32n
Byte (MB)		$\frac{n}{8}$		2n-1 to 2n	4n-3 to 4n
Word (MW)		$\frac{n}{16}$	$\frac{n}{2}$		2n-1 to 2n
Double word (DW)		$\frac{n}{32}$	$\frac{n}{4}$	$\frac{n}{2}$	

easy800 standard : n = 1 to 96
easy800-SWD: n = 1 to 128

Example 1: given Bit81 = M81; n = 81;

- Byte(MB) = n/8 = 10.125 ≈ 11
- Word(MW) =n/16 = 5.0625 ≈ 6
- Double word(DW) = n/32=2.5313 ≈ 3

Bit81 is contained in MB11, MW6 and DW3.

Example 2: given Byte21=MB21; n = 21;

- Bit(M) = 8n-7 to 8n = Bit161 to 168
- Word(MW) = n/2 = 10.5 ≈ 11
- Double word(DW) = n/4=5.25 ≈ 6

Byte21 contains the bits Bit161 to Bit168. Byte21 is contained in MW11 and DW6.

Marker bit (M), marker bytes (MB), marker words (MW) and marker double words (MD) are used in the circuit and function block diagram.

Expansion units for analog value measuring can be used on easy800 standard devices from device version number 04. The analog inputs and outputs of the expansion units are read or written via permanently assigned marker words:

Table 15: Markers and analog operands

Markers and analog operand	Display	easy800 standard	easy800-SWD	Value range	Access type[1)]
Marker 32 bit	MD	01-96	01-128	32 Bit	r, w
Marker 16 bit	MW	01-96	01-128	16 bits	r, w
Marker 8 bit	MB	01-96	01-128	8 bit	r, w
Marker 1 bit	M	1-96	01-128	1-bit	r, w
Analog inputs easy800 standard	IA X	X=01-04	None	10 Bit	r
Analog output	QA X	X=01	None	10 Bit	r, w
Analog inputs, expansion unit with 2 analog inputs	MW X	X=93, 94	None	10 Bit	r
Analog inputs, expansion unit with 2 analog inputs	MW X	X=89...94	None	10 Bit	r
Analog output, expansion unit with 1 analog output	MW X	X=95	None	10 Bit	r, w
Analog output, expansion unit with 2 analog outputs	MW X	X=95, 96	None	10 Bit	r, w

[1)] r = Read; w = Write

Avoid any accidental double assignment of markers. In this way, you can assign the available 96 or 128 marker bit contacts and also change the state of these marker bits via the first 12 marker bytes (MB), 6 marker words or 3 marker double words. This will generate undefined states. When write accesses are made successively within an MD, such as to MD1, MW2, MB4 or M32, the last write operation is retained.

→ Tip: Observing the following wiring rules will prevent the double assignment of marker bits.

- For easy800 standard with the marker bits M01-M96 use
  - Marker bytes, starting at MB13
  - Marker words, starting at MW07
  - Marker double words, starting at MD04
- For easy800-SWD with the marker bits M01-M128 use
  - Marker bytes, starting at MB13
  - Marker words, starting at MW07
  - Marker double words starting at MD04

Retentive markers

You can declare a freely definable contiguous range between marker bytes as retentive.

Device	Marker range that can be declared as retentive
easy800 standard	MB01 - MB96
easy800-SWD	max. 400 bytes from MB01 - MB512

The setting of retentive markers is explained in → section 8.2.6, “Retentive data storage”, page 308.

Number formats

The values of the marker byte (MB) and marker word (MW) data types are processed as unsigned. In order to store negative values, you must use a marker double word.

This fact must be particularly taken into account if the output of a function block can take on a negative value. You must temporarily store this value in a marker double word in order to transfer it to the input of a function block, otherwise the sign information will be lost.
The easy800 device processes calculations with a signed 31-bit value.

The value range is: –2147483648 to +2147483647

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

Bit 32 = status 0 -> positive number.

Bit 32 = status 1 -> negative number.

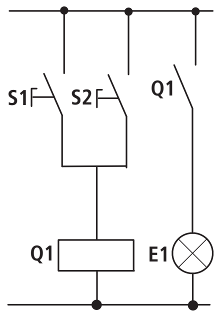
Example:
$00000000000000000000010000010010_{bin} = 412_{hex} = 1042_{dec}$

Example:
$11111110110111001111010001000111_{bin} = \text{FEDCF447}_{hex} = -19073977_{dez}$

4.5 Working with contacts and coils

Switches, pushbuttons and relays from a conventional hardwired circuit diagram are wired in the easy800 circuit diagram via input contacts and relay coils.

Hardwired

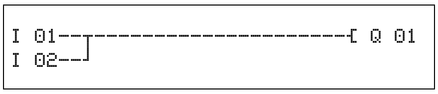


Wired with an easy800 device

easy800 connection
N/O contact S1 to input terminal I1
N/O contact S2 to input terminal I2
Load E1 to output terminal Q1

S1 or S2 switch on E1.

easy800 circuit diagram:



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

Figure 91:Circuit diagram with inputs I 01, I 02 and output Q 01

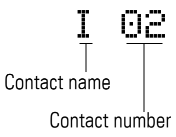
First specify which input and output terminals you wish to use in your circuit.

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.

The jump destination has a special position for the input contacts and the jump location for the output relays. These are used for structuring a circuit diagram (→ Section 4.5.15, “Jumps“, Page 119).

The following describes how you wire the different contacts and coils of different relay types or function blocks (inputs) in the circuit diagram.

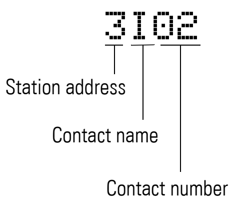
4.5.1 Entering and changing contacts



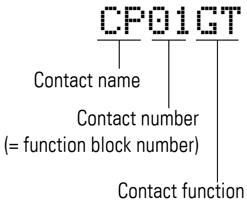
You choose an input contact in the easy800 device by means of the contact name and the contact number.

Example: Input contact basic device

or



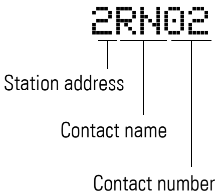
Example: Input contact via NET (here from NET station 3)



The contact of a function block consists of the abbreviated function block name, the number and the contact function.

Example: Contact of “comparator” function block

How a safety or standard function block is added to the circuit diagram and parameterized is described in → Section 4.7, “Working with function blocks“ on Page 127.



If the contact of an Net station is used in the circuit diagram, the NET ID (address) of the station is placed in front of the contact name, → section , “Wiring a contact or coil of another NET station in the circuit diagram”, page 121.

Example: Contact of a NET station

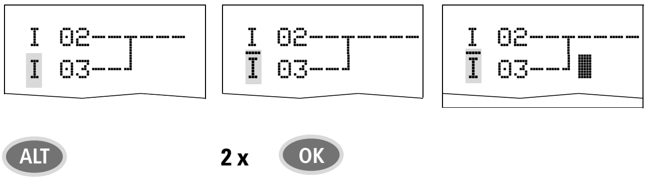
4.5.2 Changing N/O contacts to N/C contacts

DANGER

Persons, systems and machine may be put at risk if an N/C contact is misinterpreted. When using N/C contacts in the program of easy800-SWD always evaluate the PRSNT and DIAG diagnostic bits of this station.

You can define each contact in the circuit diagram as a N/O or N/C.

- ▶ Switch to Entry mode and move the cursor over the contact name.
- ▶ Press ALT. The N/O contact will change to a N/C contact.
- ▶ Press OK twice to confirm the change.



```
I 02--┬----    I 02--┬----    I 02--┬----
I 03--┘        Ī 03--┘        Ī 03--┘█
```

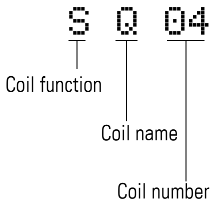
Change contact I 03 from N/O to N/C

Bear in mind that the active state of an N/C contact is 0. The 0 state of a contact may, however, be present if the station is missing or is operating incorrectly. The use of an N/C contact in the circuit diagram without evaluating the diagnostics bit may cause incorrect interpretations. Further information on the diagnostics bits is provided in → Chapter 7, “SmartWire-DT“, page 288.

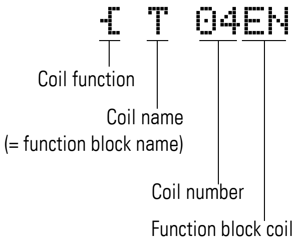
4.5.3 Entering and changing coils

With a relay coil or a function block, you choose the coil function, coil name, coil number as well as the function block coil. With coils of an easyNet station, choose the address (NET ID) in front of the coil name.

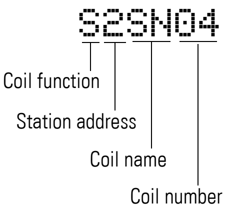
→ The coil number in the figures on the left must be the same as the function block number!



Example: Relay coil "Output Q"



Example: Relay coil for timing relay function block with control coil



Example: Relay coil of a NET station

→ A complete list of all contacts and coils,
→ section 12.8, "List of function blocks", page 369.

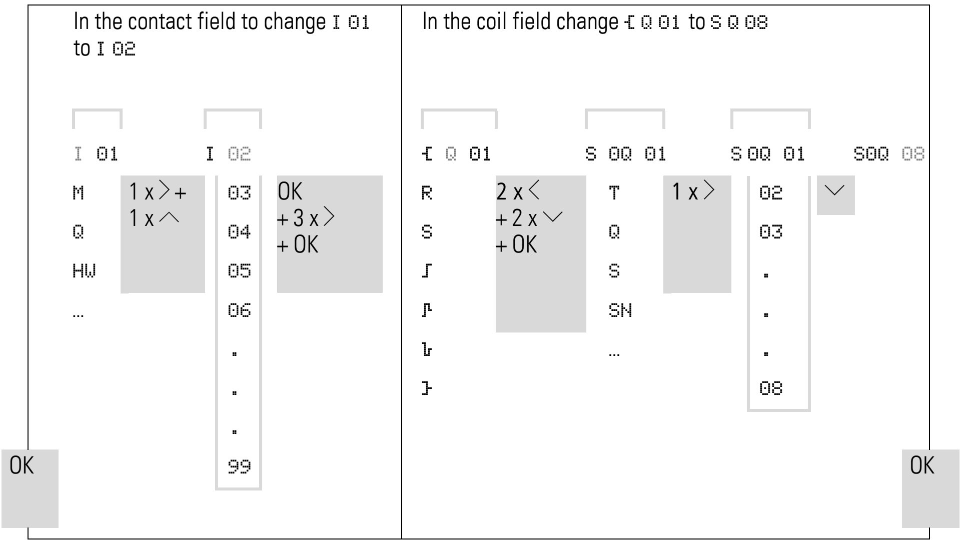
I 01

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

→ The easy800 device proposes contact I 01 or the coil -[Q 01 for entry in an empty field.

- ▶ Use < > ∧∨ to move the cursor to a contact or coil field.
- ▶ Press OK to switch to Entry mode.
- ▶ Use < > to select the position you wish to change and press OK to move to the next position (a selected position is shown in light grey in the following figure).
- ▶ Use ∧∨ to modify the value of the position.

The easy800 device closes entry mode as soon as you leave a contact or coil field with ‹ › or OK.



4.5.4 Deleting contacts and coils

- Use ‹ › ∧∨ to move the cursor to a contact or coil field.
- Press DEL.

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

4.5.5 Creating or changing connections

Contacts and relay coils are connected with the wiring arrow in Connect mode. The easy800 device displays the cursor in this mode as a wiring arrow.

- Use ‹ › ∧∨ to move the cursor to the contact or coil field from which you wish to create a connection.

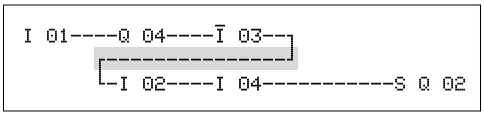
→ Do not position the cursor on the first contact field. Here the ALT button has a different function (add rung).

- Press ALT to switch to “Connect mode”.
- Use ‹ › to move the arrow between the contact and coil fields and use ∧∨ to move between the rungs.
- Press ALT to leave Connect mode.

The easy800 device closes the mode automatically as soon as you have moved the arrow to an occupied contact or coil field.

→ In a rung, the easy800 device connects contacts and the connection to the relay coil automatically if no empty fields are between them.

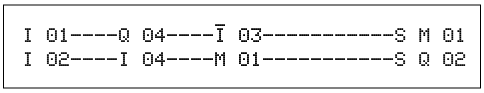
Never wire backwards, → Section , “Example: Do not wire backwards“, Page 318.



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

Figure 92:Circuit diagram with five contacts, invalid

When using more than four contacts in series, use one of the 96 M or 128 M markers.



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

Figure 93:Circuit diagram with M marker relay

4.5.6 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.

The easy800 device deletes a connection branch. Closed adjacent connections will be retained.

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

4.5.7 Insert Rung

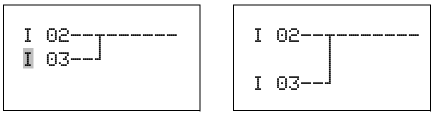
The circuit diagram display shows three of the 256 rungs at the same time. Rungs outside of the display, including empty rungs, are scrolled by easy800 automatically in the circuit diagram display if you move the cursor beyond the top or bottom of the display.

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

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

▶ Press ALT.

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



```
I 02--┬-------
I 03--┘
```

```
I 02--┬--------
      │
I 03--┘
```

Figure 94:Inserting a new rung

4.5.8 Delete Rung

The easy800 device only removes empty rungs (without contacts or coils).

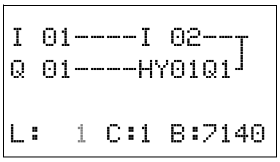
- Delete all contacts and coils from the rung.
- Position the cursor on the first contact field of the empty rung.
- Press DEL.

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

4.5.9 Got to a rung

The GO TO function can be used to go to a different rung.

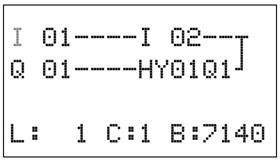
- Press ESC and use cursor buttons ∧∨ to select the GO TO menu.
- Press OK.
- Select the required rung (L...) with the ∧∨ cursor buttons.



```
I 01----I 02--┐
Q 01----HY01Q1┘

L:  1 C:1 B:7140
```

Always the first contact of the rung is displayed.

- Press OK.



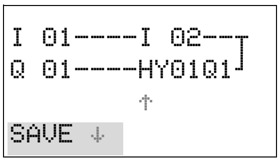
```
I 01----I 02--┐
Q 01----HY01Q1┘

L:  1 C:1 B:7140
```

The cursor stops at the first contact L 1 of the required rung.

→ You can use the "GO TO" function to jump to any rung, up to the last wired rung.

4.5.10 Saving circuit diagrams

- Press ESC to save a circuit diagram.



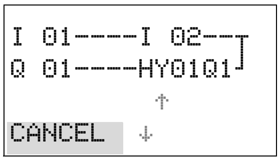
```
I 01----I 02--┐
Q 01----HY01Q1┘
         ↑
SAVE ↓
```

The menu on the left appears in the status line.

- Pressing OK will save the entire program, circuit diagram and function blocks.

After saving, you will be returned to the previous menu from which you have opened the circuit diagram.

4.5.11 Aborting circuit diagram entry

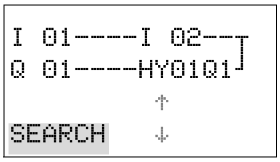


```
I 01----I 02--┐
Q 01----HY01Q1┘
          ↑
CANCEL    ↓
```

- ▶ Press ESC to leave the circuit diagram entry without saving.
- ▶ Use the cursor buttons ∧∨ to select the CANCEL menu.
- ▶ Press OK.

The circuit diagram is closed without saving.

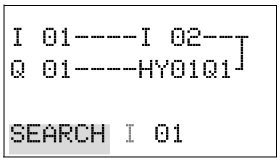
4.5.12 Searching for contacts and coils



```
I 01----I 02--┐
Q 01----HY01Q1┘
          ↑
SEARCH    ↓
```

Boolean operands or function blocks that are wired as contacts or coils can be found in the following way:

- ▶ Press ESC. Use the cursor buttons ∧∨ to enter the SEARCH menu.
- ▶ Press OK.

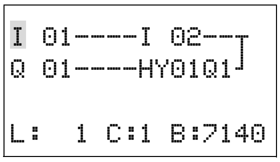


```
I 01----I 02--┐
Q 01----HY01Q1┘

SEARCH I 01
```

- ▶ Use the cursor buttons ∨ and ‹ › to select a contact or coil as well as the required number.

For a function block you select the function block name and the number.

- ▶ Confirm the search with the OK pushbutton.

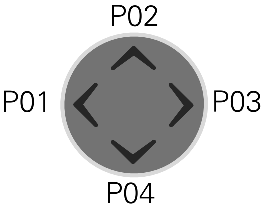


```
I 01----I 02--┐
Q 01----HY01Q1┘

L:  1 C:1 B:7140
```

The search starts from where the search was called and is continued to the end of the circuit diagram. It applies only to this area. If the required contact or coil is located above the point of calling, start the search at the beginning of the circuit diagram. If the search is successful, you will automatically reach the required contact or coil field in the circuit diagram.

4.5.13 Switching with the Cursor Buttons

With an easy800 device, you can also use the four cursor buttons as hardwired inputs in the circuit diagram.



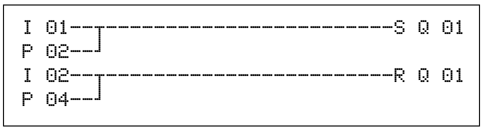
The P buttons can also be used for testing circuits or manual mode. These button functions are also useful for servicing and commissioning purposes.

The cursor buttons are wired in the circuit diagram as contacts `P 01` to `P 04`.

Requirement:

The P buttons have been activated in the System menu → section 12.7.3, "System menu", page 367.

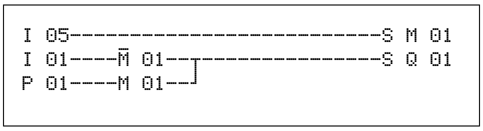
Example 1
This standard circuit diagram example enables a lamp at output Q1 to either be switched on or off via the inputs I1 and I2 or via the cursor buttons ∧∨.



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

Figure 95:Switch Q1 via I1, I2, ∧, or ∨

Example 2
This circuit diagram example causes output Q1 to be actuated by input I1. I5 switches to cursor operation and disconnects the rung `I 01` via M `01`.

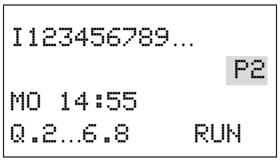


```
I 05--------------------------S M 01
I 01----M̄ 01--┬-----------------S Q 01
P 01----M 01--┘
```

Figure 96:I5 switches to cursor buttons.

→ The easy800 device only evaluates P button entries if the status display is shown.

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



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

Display in Status display:

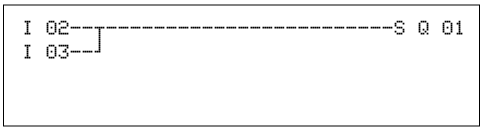
- P: Button function wired and active,
- P2: Pushbutton function wired, active and P2 pushbutton ∧ executed,
- P-: Button function wired, not active,
- Empty field: P pushbuttons not used.

4.5.14 Checking the circuit diagram

The easy800 device features an integrated power flow display with which you can follow the switching states of contacts, relay and function block coils during operation. The circuit diagram display performs two functions depending on the mode:

- STOP: Create the circuit diagram.
- RUN: Power flow display.

▶ Create the small parallel circuit below and save it.



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

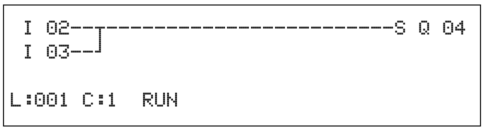
Figure 97:Parallel connection

▶ Switch the easy800 to the operating mode RUN via the main menu.
▶ Return to the circuit diagram display.

You are now unable to edit the circuit diagram.

→ If you change to the circuit diagram display but cannot change a circuit diagram, first check whether the easy800 device is in STOP operating mode.

▶ Switch on I3.



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

L:001 C:1  RUN
```

Figure 98:Power flow display

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

You can follow energized connections across all rungs by scrolling the display up and down.

The power flow display indicates at the bottom right the controller is in RUN mode. (→ section , “Power flow display with Zoom function”, page 83).

→ The power flow display does not show signal changes in the millisecond range due to the inherent technical delay of LCD displays.

4.5.15 Jumps

Jumps can be used for structuring a circuit diagram. They can replace the function of a selector switch, for example, for Manual/Automatic mode or for different machine programs.

Jumps consist of a jump location and a jump destination (label).
Jumps exist in the

- circuit diagram, for skipping rungs:
  Jump location and jump destination are located in the same circuit diagram
- function block editor, for skipping function blocks:
  Jump location is located in the circuit diagram and jump destination in the function block editor
  The use of jumps in the function block diagram is explained in Chapter “Function blocks”→ section 5.19, “LB, jump label”, page 213 and → section 5.18, “JC, conditional jump”, page 211.

The easy800 device allows the use of up to 32 jumps.

Circuit diagram elements for jumps in the circuit diagram

Contact (N/O1))	
Numbers	01 up to 32

Coils	-[
Numbers	01 up to 32
Coil function	-[,]-, ⌋, ⎍, ⎿

1) Can only be used as first contact on the left

Function of jumps

If the jump coil is triggered, the rungs after the jump coil are no longer processed. 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 leftmost contact = Jump label

The jump destination is always an N/O contact with the status 1.

→ Backward jumps cannot be executed due to the way in which easy800 works. If the jump label does not come after the jump coil, the jump will be made to the end of the circuit diagram. The last rung is also skipped.
Multiple usage of the same jump coil and the same contact is possible as long as this is done in pairs, this means:

- Coil -[:1/jumped area/contact :1,
- Coil -[:1/jumped area/contact :1,

etc.

CAUTION
If rungs are skipped, the states of the coils are retained.
The time of started timing relays continues to run.

Power flow display of skipped area

Jumped sections 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 for jumps

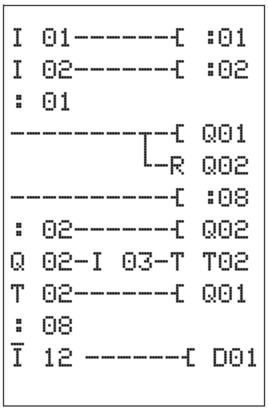
A selector switch is used to select two different sequences.

- Sequence 1: Switch on motor 1 immediately.
- Sequence 2: Activate barrier 2, wait time, then switch on motor 1.

Contacts and relays used:

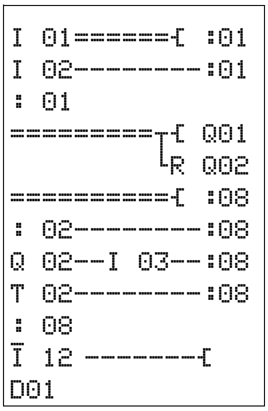
- I1 sequence 1
- I2 sequence 2
- I3 guard 2 moved out
- I12 motor-protective circuit-breaker switched on
- Q1 motor 1
- Q2 guard 2
- T 01 wait time 30.00 s, on-time
- D 01 Text “Motor protective circuit breaker tripped”

Circuit diagram:



```
I 01-------[ :01
I 02-------[ :02
: 01
---------┬-[ Q01
         └-R Q02
-----------[ :08
: 02-------[ Q02
Q 02-I 03-T T02
T 02-------[ Q01
: 08
Ī 12 -------[ D01
```

Power flow display: I 01 selected:



```
I 01=======[ :01
I 02---------:01
: 01
=========┬[ Q01
         └R Q02
==========[ :08
: 02---------:08
Q 02--I 03--:08
T 02---------:08
: 08
Ī 12 --------[
D01
```

Range from jump label 1 processed.

Jump to label 8.
Range to jump label 8 skipped.

Jump label 8, circuit diagram processed from this point on.

4.5.16 Wiring NET operands in the circuit diagram

A NET with several stations always allows the reading of all inputs and outputs. This is possible regardless of whether a circuit diagram is being processed or not on the NET station to be read. The inputs and outputs are addressed in the NET by using the preceding NET-ID of the station. The inputs and outputs of a NET station are identified with nI.. and nQ, and its local expansion unit with nR.. and nS...

The permissible access by stations to the inputs and outputs of other stations depends on the operating mode of the devices on the NET, in which the following applications are possible:

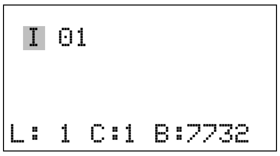
Operating devices on the NET	**Usable NET operands of data type…**	
	Bit	**32 Bit (DWord)**
REMOTE IO mode; only NET station 1 processes a circuit diagram.	nI.., nR.., nQ.., nS...	–
All NET stations each process a circuit diagram.	nI.., nR.., nQ.., nS.., nRN.., nSN...	nPT.. and nGT...
Mixed operation of NET stations with and without a circuit diagram.	nI.., nR.., nQ.., nS.., nRN.., nSN...	nPT.. and nGT...

n = NET-ID

For EASY806-DC-SWD, the device inputs and outputs I1-I4 and Q1-Q2 cannot be read nor written by other NET stations. REMOTE IO is always deactivated. Other NET stations cannot access the stations of the SWD line.

Wiring a contact or coil of another NET station in the circuit diagram

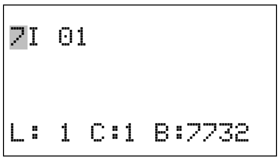
Requirement: You must have selected an I.., R.., Q.., R.., RN.. S.. or SN.. operand in the circuit diagram and Entry mode must have been selected.



```
 I 01


L: 1 C:1 B:7732
```

This mode is displayed by a flashing operand.

▶ Use the cursor button ‹ to move the cursor to the position to the left of the operand. A flashing zero appears as the initial value.



```
7I 01


L: 1 C:1 B:7732
```

Enter the cursor buttons ^ or ˅ to define the required NET ID; in this case NET ID 7.

Confirm the entry with the OK button.

The local I.., R.., Q.. or S.. operand has been changed to a NET operand nI.., nR.., nQ.. and nS...

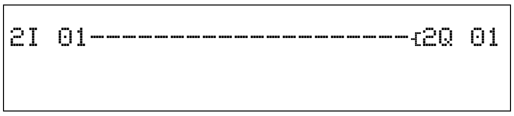
REMOTE IO mode

Station 1 has read and write access to all inputs and outputs of the NET stations with REMOTE IO mode.

Only NET station 1 has write access via the nQ.. and nS.. operands. It writes via coils nQ.. and nS.. to the output terminals of another NET station which is running in REMOTE IO mode without a circuit diagram.

Example 1:

NET station 1 is required to read data from input I1 of NET station 2 and write data to output Q1 of NET station 2. NET station 2 does not have a circuit diagram.



```
2I 01--------------------Ɛ2Q 01
```

Figure 99:Circuit diagram in NET station 1

Several NET stations with their own circuit diagram

The relevant NET stations each process a circuit diagram.

- Every station has read access to all inputs and outputs of the other stations.
- The station only has write access to its local outputs and outputs of its local expansion unit.
  Example: Station 1 uses the state of Q1 of station 2 in its circuit diagram. Station 1, however, cannot set Q1 of station 2 to 1.
- Send Net (SN) and Receive NET(RN) is used for exchanging bits. These operands are always used in pairs.
- Put (PT) and Get (GT) are used in order to exchange double word operands via the NET. More information about the function blocks → section 5, “Function blocks”, page 135.

SN-RN combination for bit exchange on the NET

- Writing via SN

The NET operand SN (Send NET) is used for sending bit data from one NET station to another. To do this you select the SN operand in a coil field.

- Reading using RN

The RN (Receive NET) NET operand is used to receive the bit data that another NET station has sent. To do this you select the RN operand in a contact field.

As the RN and SN must always be used in pairs, the following rule applies:

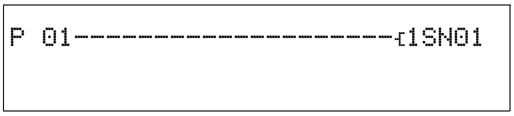
- on the sending station and receiving station use the same operand number for each SN/RN pair to be formed.
- in the circuit diagram of the sending station you set for the SN operand (coil) the station number (NET ID) of the receiving station.
- in the circuit diagram of the receiving station you set for the RN operand (contact) the station number (NET ID) of the sending station.

RN-SN Example

NET station 2 sends the status of the P button P01 via SN1 to NET station 1.

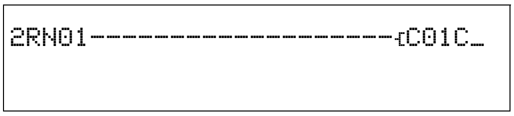
Station 2 Destination-ID			Station 1 Source-ID	
1	SN	◄────────►	2	RN

The relevant circuit diagram is as follows:



```
P 01--------------------[1SN01
```

On NET station 1 the status of P1 is associated via RN01 as a count pulse for the counter relay C01.



```
2RN01-------------------[C01C_
```

NET operand GT.. (receive), PT.. (send) and SC.. (set date and time)

The function blocks are of data type 32-bit. They only function if the NET is operating correctly (→ section 10, “Diagnostics possibilities”, page 329). More information about the function blocks → section 5, “Function blocks”, page 135.

4.6 Program transfer from and to the memory card

A program with all the necessary parameters can be loaded from the memory card into the easy800 standard device or written from the device to the card.

Information stored on the memory card is "non-volatile" and thus you can use the card to archive, transfer and copy circuit diagrams.

On an easy800 standard without display or keypad you can automatically load the program from the fitted memory card every time the supply voltage is switched on (see also → Section 8.2.3.1, "CARD MODE", Page 305).

→ Observe the restrictions described in → section 4.6.2, "Program exchange between devices with memory card", page 126!

Programs are transferred via the serial interface. How to remove the cover of the serial interface, connect a programming cable and fit a memory card is described in → section 2.8, "Connecting the serial interface", page 60.

On the memory card you can save:

- easy800 circuit diagram
- All parameter settings of the circuit diagram.
- P pushbuttons.
- INPUT DELAY (debounce).
- Password.
- Retention on/off and retentive range.
- easyNet configuration.
- Daylight Saving Time settings.

The loading and storing of a program using the memory card are carried out in the CARD menu. The following overview shows how you can access this menu without a password.

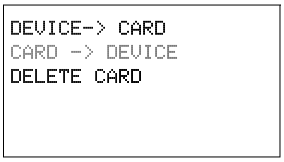
For easy800 standard only use the EASY-M-256K card.

Requirement:

- The device is in STOP operating mode.
- The supply voltage is switched on.
- A memory card must be fitted in the device.
- You must be in the CARD menu.
- The password protection must be removed (→ Section 8.1, "Password protection").

→ If the supply voltage fails when the card is accessed, you must repeat the transfer as the data is possibly not fully transferred.

Loading the program from the card (CARD -> DEVICE)



```
DEVICE-> CARD
CARD -> DEVICE
DELETE CARD
```

easy800 standard always loads the program of the card, even if this is different to the program in the device. All passwords that are present when saving to the card are also reactivated after the card is loaded.

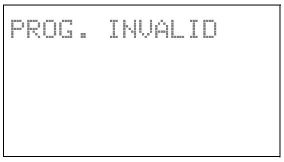
Basic requirements:

The memory card must contain a valid program.

▶ Select DEVICE->CARD.
▶ Confirm your selection with OK.

The program is transferred from the card to the device.

The load operation can be cancelled by pressing ESC beforehand.



```
PROG. INVALID
```

Transfer problems

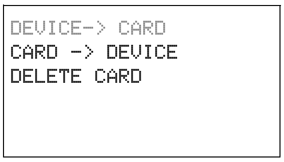
If there is a transfer problem or a checksum error, the easy800 standard device will output the message PROG INVALID.

The storing device saved its type and version on the card. A program to be loaded from the card must be suitable for the destination device, otherwise this will refuse the transfer and a program in the easy800 device will be retained (→ section 4.6.2, “Program exchange between devices with memory card”, page 126)

Possible causes:

- The memory card is empty.
- A checksum error was detected when comparing the stored checksum and the checksum calculated for the transfer.
- The circuit diagram on the card uses a function block that the easy800 standard device does not recognize.

Loading the program on the card (DEVICE -> CARD)



```
DEVICE-> CARD
CARD -> DEVICE
DELETE CARD
```

Requirement:

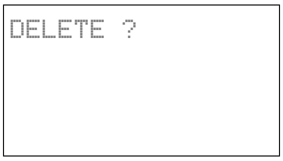
easy800 standard has a display and keypad.

▶ Select DEVICE->CARD.
▶ Confirm your selection with OK.

The program is transferred from the device to the card.

4.6.1 Deleting a program on the card

Requirement: You must be in the CARD menu.



```
DELETE ?
```

- ▶ Select the DELETE CARD menu.
- ▶ Confirm the safety prompt DELETE ? with OK if you wish to delete the card content.

Press ESC to cancel.

- ▶ After transferring or deleting a program, remove the memory card and close the cover.

4.6.2 Program exchange between devices with memory card

Upward compatibility on easy800 standard devices

The easy800 standard device family allows programs from an older version to be read in newer device versions without any restrictions in function.

A memory card with a program can always be read by an easy800 standard device that has a newer operating system (with a higher version number) than the device from which the memory card was written. The program read back is executable.

Downward compatibility with easy800 standard devices

If a program is written to the memory card by an easy800 standard device with a newer operating system, this program can also be read and run by an easy800 standard device with an older operating system. This requires that the program does not contain any functions that are not supported by the older operating system.

If the program contains functions that are not supported by the older operating system, the PROG INVALID! fault message appears with the CARD -> DEVICE transfer.

The following special point applies to older easy800 standard devices:

→ Memory cards that are created with an easy800 standard device version > 03 cannot be read by devices with version ≦ 03 (no upward compatibility).

Table 16: Compatibility matrix for program exchange between devices

Write card from device …→ ↓ Card read with device…	easy800 standard, V01	easy800 standard, V03/04	easy800 standard, V07	easy800 standard, V10
easy800 standard, V01	OK	–	–	–
easy800 standard, V03/04	OK	OK	–	–
easy800 standard, V07	OK	OK	OK	–
easy800 standard, V10	OK	OK	OK	OK

Program exchange between devices with memory card

Programs can be loaded and run on devices with the same device version within the easy800 standard device family or devices with the same device version within the MFD-Titan device family, even if the target device does not have the necessary hardware requirements, see example.

Example: A program with NET operands (e.g. 3RN01) that was created for a network-enabled MFD-Titan is loaded on an MFD-Titan without an easyNet connection. On the MFD without the easyNet connection, all NET operands supply the value zero.

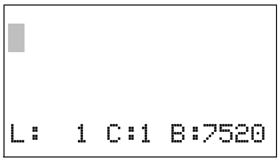
4.7 Working with function blocks

This chapter explains the basic operating principles involved in working with function blocks. A detailed description of all available function blocks can be referred to in → Section 5, "Function blocks" from Page 135. The function blocks are used to simulate some of the devices used in conventional open-loop and closed-loop control systems. You can first use the function block in the circuit diagram and then define the actual and setpoint parameters for the function block inputs and outputs in the function block editor. Or vice versa: you create the function block in the function block editor, define the parameters and use it then in the circuit diagram. With easy800 standard you can insert up to 255 function blocks in the function block list and with easy800-SWD 256 function blocks.

→ There is no entry limit on easy800 standard. You must check the maximum number of function blocks yourself otherwise a function block error may occur.

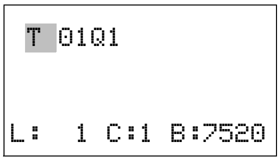
4.7.1 Adding function blocks to the circuit diagram for the first time

Add a function block to the circuit diagram for the first time in the following way:



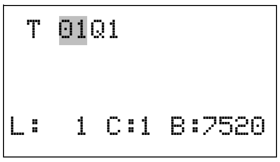
```
L:  1 C:1 B:7520
```

- Press OK -> PROGRAM->PROGRAM>CIRCUIT DIAGRAM to enter the circuit diagram display from the Status display.
- Use the ∧ ∨ < > buttons to move the cursor to a free contact or coil field.
- Press OK so that the easy800 device switches to entry mode.



```
 T 01Q1


L:  1 C:1 B:7520
```

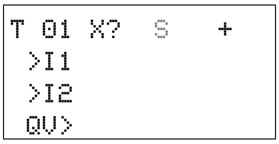
- Then use cursor buttons ∧ ∨ to select the required function block, e.g. timing relay using the T identifier.



```
 T 01Q1



L:  1 C:1 B:7520
```

- ▶ When the function block identifier flashes, press OK or the cursor button > to move to the function block number and press OK.

The display switches to the function block editor. Here you normally define all function block parameters. As you have reached the function block editor via the circuit diagram in this case, you can only set the basic parameters.



```
T 01 X?  S   +
 >I1
 >I2
 QV>
```

The figure on the left shows the function block editor of the Timing relay function block.

The basic parameters are the parameters in the first line. The basic parameters may vary according to the function block. All function blocks have the basic parameter +/–. Through the +/– symbol you can switch the parameter display on and off in RUN mode to allow (+) or prevent (–) changes to be made to setpoints (constants). The +/– symbol must at least be confirmed with OK.

→ 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.

- ▶ Use the cursor buttons < > to select the parameter to be changed, for example the time range "S".
- ▶ Use the cursor buttons ∧∨ to change the parameter value to a different time range such as M:S.
- ▶ Press OK to exit the parameter dialog if you wish to save the parameters or press ESC, if you do not wish to parameterize the function block and add it to the circuit diagram.

After saving or canceling, the cursor returns to the position in the circuit diagram where you last left it.

To complete the parameter setting of the function block, such as the setting of setpoints, call up the function block editor in the following way:

- ▶ Press ESC in order to save the circuit diagram with the newly added function block.
- ▶ Answer the subsequent SAVE prompt with OK.

The circuit diagram is saved and the easy800 device changes to the next higher menu level.

How to carry out the complete parameterization is described in → section 4.7.3, "Parameter setting in the function block editor", page 129.

4.7.2 Function block list

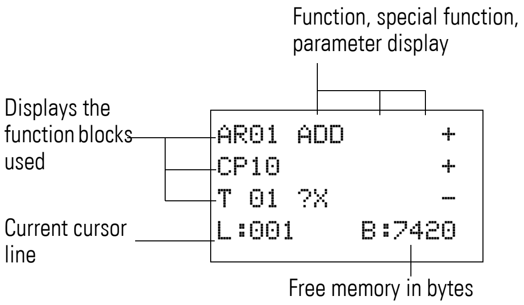
The function block list is shown on the way to the function block editor.

To access the function block list from the status list, click

- ▶ OK -> PROGRAM -> PROGRAM -> FUNCTION BLOCKS

This lists all function blocks that were ever used in the circuit diagram, and also those that were already deleted in the circuit diagram. The function block list is empty if no function blocks are present.

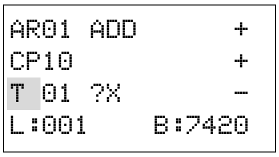
In our example, the function block list contains the function blocks AR, CP and T. The function blocks are created in the order in which they were edited.



```
AR01 ADD      +
CP10          +
T 01 ?X       -
L:001    B:7420
```

Function, special function, parameter display
Displays the function blocks used
Current cursor line
Free memory in bytes

Figure 100:Explanation of the function block list



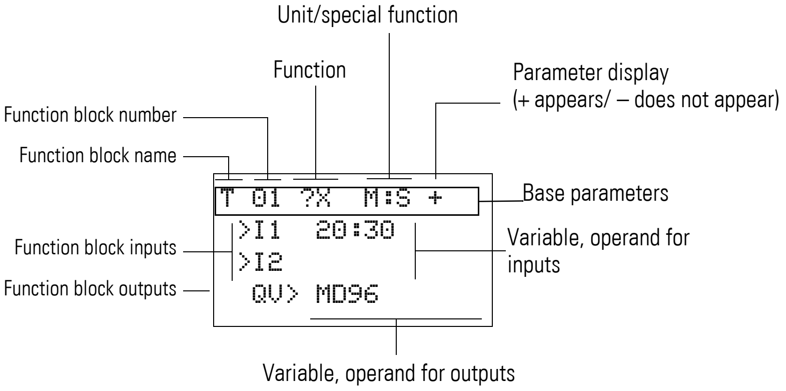
```
AR01 ADD      +
CP10          +
T 01 ?X       -
L:001    B:7420
```

- ▶ Use the cursor buttons ∧∨ ‹ › to select the required function block from the function block list, in this case timing relay T01.
- ▶ Confirm your selection with OK.

The timing relay is displayed in the function block editor.

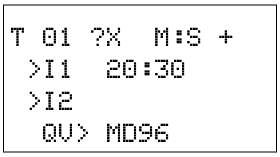
4.7.3 Parameter setting in the function block editor

The function block can be fully parameterized in the function block editor. This is accessed via the function block list. Access is blocked if the program is password protected.



```
T 01 ?X  M:S +
 >I1  20:30
 >I2
  QV> MD96
```

Figure 101:Display of the function blocks in the function block editor



```
T 01 ?X   M:S +
 >I1  20:30
 >I2
  QV> MD96
```

Example: Timing relay function block

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

Assigning operands at the input of a function block

The following operands 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,
- the QV outputs of all function blocks.

You can set the parameters of the function block as follows:

▶ Scroll with the ∧ or ∨ cursor buttons through the constants of the function block inputs.

▶ Change the values for a parameter set:
  - OK: Switch to Entry mode.
  - ‹ › Change decimal place.
  - ∧∨ Change value at a decimal place.

▶ OK: Save constant immediately and exit the parameter display by pressing ESC.

ESC: Keep the previous setting and exit the parameter display.

Ensure that the input of a function block is not assigned impermissible values during operation. There is a danger of this if you assign negative values to the input although the function block can only accept positive values.

If, for example, the T timing relay function block is triggered with a negative time setpoint, it will no longer switch in the expected way.

You should therefore take care to exclude such situations, as the easy800 device cannot foresee these when the parameters are assigned.

→ Tip: If, for example, you have set the output QV of the AR arithmetic function block at input I1 of the T function block, you should connect a CP comparator in between in order to signal the occurrence of a negative value.

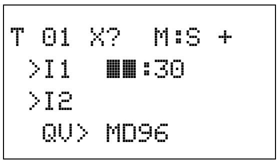
In most applications, a thorough simulation is enough to prevent any impermissible values at the function block input.

Assigning operands at the output of a function block

The following operands can be assigned to the output of a QV function block:

- Markers such as MD, MW, MB,
- or the analog output QA.

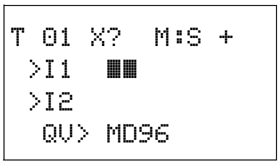
Deleting operands on the function block inputs/outputs



```
T 01 X?  M:S +
 >I1  ■■:30
 >I2
  QV> MD96
```

Position the cursor on the required operand.

▶ Press the DEL pushbutton.



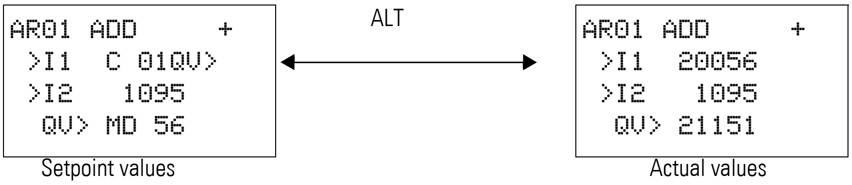
```
T 01 X?  M:S +
 >I1  ■■
 >I2
  QV> MD96
```

The operand is deleted.

Behavior of the function block editor with different operating modes

When working with the function block editor, the mode of the device is important.

1. STOP: Access is possible to all parameters of the function blocks.
2. RUN:
   - Only access to the basic parameters is possible.
   - It is only possible to change input values at function blocks if they are constants. The modified constants are used directly for further processing in the circuit diagram.
   - Pressing ALT enables you to toggle between setpoints and actual values
     Example:
     - >I1 = Actual value, here from the output of the counter C 01
     - >I2 = Constant 1095.
     - QV> = Marker double word MD56



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

Setpoint values

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

Actual values

4.7.4 PARAMETERS menu

This menu item can only be activated in RUN mode.

Function blocks with basic parameters that you have set to + via the +/- character in the function block editor are displayed in the PARAMETERS menu and can be changed. It is only possible to change constants. Other operands cannot be changed.

It is also possible to change parameters via the PARAMETERS menu if you have saved the program and therefore password protected the function block editor. This is the point of this menu. When the password is activated and the +/- basic parameter for each function block is set, you can allow or deny the operator of the system the possibility to change the values.

▶ Move from the Status display to the Parameters display by pressing OK -> PARAMETERS.
▶ Follow the operating steps described in → Section "Assigning operands at the input of a function block".

4.7.5 Deleting function blocks

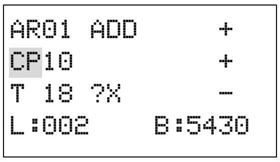
To remove a function block, you must remove it from the circuit diagram and from the function block list.

Requirement: The easy800 device must be in STOP operating mode.

▶ Press OK -> PROGRAM->PROGRAM>CIRCUIT DIAGRAM to enter the circuit diagram display from the Status display.
▶ Move the cursor in the circuit diagram to all the contact fields and the coil field in which the function block to be deleted is used and press DEL each time.

Deleting function blocks from the function block list

In order to prevent accidental deletion, a function block continues to be managed in the function block list, even if it was already removed in the circuit diagram. To delete the function block permanently and therefore to free up more memory, you must remove it from the function block list.



```
AR01 ADD     +
CP10         +
T 18 ?X      -
L:002    B:5430
```

▶ Open the function block list via PROGRAM -> FUNCTION BLOCKS.
▶ Select the function block to be deleted in the function block list, in this case CP10.
▶ Press the DEL pushbutton.

The function block is removed from the function block list.

▶ Press ESC in order to save the function block list with the deleted function block.
▶ Confirm with OK.

▶ In the function block list select the required function block.

In this example, select the data block comparator AR01 in Adder mode.

▶ Press OK.

Depending on the display selected the function block is shown with the actual values and the result, or with the set operands and constants.

Press the ALT pushbutton if you wish to switch from operand to actual value display or vice versa during the check of the standard function block.

▶ Press the ALT pushbutton again (see also the following note).

Tips for working with function blocks

- Current actual values are deleted when you switch off the power supply or switch the easy800 device to STOP mode. Exception: Retentive data keeps its state (→ section 8.2.6, "Retentive data storage", page 308,). The most recent actual values are transferred to the operands every cycle. The data function block is an exception.
- If you want to prevent other people from modifying the parameters of the function blocks, change the access enable symbol from "+" to "–" when creating the circuit diagram and setting parameters. You can then protect the circuit diagram with a password.
- As every function block in the function block list takes up memory, even if it is no longer being used and has been deleted in the circuit diagram, you should clear it occasionally. Check the function block diagram for any function blocks that are no longer required and delete them, → Section 4.7.5, "Deleting function blocks", Page 132.
- The function blocks are designed so that an output value of a function block can be assigned directly with an input of another function block. The 32-bit data format is used automatically. This also enables the transfer of negative values.

CAUTION

The following applies to RUN mode: A easy800 device processes the function block after a pass through the circuit diagram. This takes the last status of the coils into account.

5 Function blocks

The following description of the function blocks is in alphabetical order.

Each function block is first described in brief. All contacts and coils of the function block are shown in tabular form. The meaning of the 1 state for the contact is described, and the function of the coil in the 1 state are described.

Another table uses an example to explain and then describe in detail the parameters of the function block.

All function blocks have the adjustable basic parameter +/-. It is therefore described here for all function blocks and only mentioned in brief in the particular function block descriptions.

The basic parameter +/- is displayed in the first line of the function block editor. It defines whether the parameter set of the function block can be called via the PARAMETERS menu during RUN mode. Additional information on this → section 4.7.4, "PARAMETERS menu", page 132.

Parameters	Function
+	Parameter set can be called via the PARAMETERS menu
–	Parameter set is accessible via PARAMETERS men u

The default setting of this parameter is +.

Not all easy800 have all function blocks. Information on this is provided in the relevant function block description.

5.1 A, analog value comparator/threshold value switch

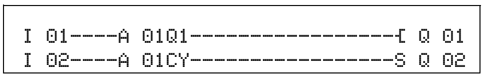
An analog value comparator or threshold value switch is used, for example, to compare analog values or marker contents and switch when defined threshold values are reached.

easy800 allows the use of up to 32 analog value comparators/threshold switches.

Contacts and Coils

Contacts	Meaning
A ..Q1	Reaching of the comparison result, according to the selected operating mode GT, EQ or LT
A ..CY	Value range exceeded

Example for function block analog value comparator

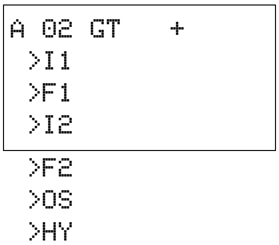


```
I 01----A 01Q1------------------[ Q 01
I 02----A 01CY------------------S Q 02
```

Figure 102:Wiring the contacts

Parametric programming of the function block inputs/outputs

When using the function block in the circuit diagram for the first time, use OK to automatically enter the general display of function block parameters, as shown in the figure on the left. Enter the function block settings here. The display contains the following elements:



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

A 02	Function block analog value comparator number 02
GT	Operating mode: Greater than
+	Parameter set can be called via the PARAMETERS menu
>I1	Comparison value 1, is compared with the comparison value 2 at >I2 Value range: -2147483648 - 2147483647
>F1	Gain factor for >I1 (>I1 = >F1· Value) Value range: -2147483648 - 2147483647
>I2	Comparison value 2 I1 Value range: -2147483648 - 2147483647
>F2	Gain factor for >I2 (>I2 = >F2· Value) Value range: -2147483648 - 2147483647
>OS	Offset (zero point offset) for the value of >I1 Value range: -2147483648 - 2147483647
>HY	Positive and negative switching hysteresis for comparison value I2 Value range: -2147483648 - 2147483647

Operating mode

This parameter defines which comparison result is to be used to close function block contact Q1.

Parameters	Function, parameterization in the function block
GT	>I1 greater than >I2
EQ	I1 equal to >I2
LT	>I1 less than >I2

The default settings of this parameter is LT.

Inputs

I1 and I2: The signals at function block inputs I1 and I2 are compared.

F1 and F2: If required adjust the comparison values and increase them by entering gain factors F1 and F2.

- F1 is permanently assigned to I1.
- F2 is permanently assigned to I2.

OS: You can use the offset input OS to move the zero point of the function block input I1.

HY: The HY function block input is used for the positive and negative switching hysteresis of the input I2.

All function block inputs can be assigned the following operands:

- Constant.
- Markers MD, MW, MB.
- Analog inputs IA01 to IA04:
  - IA01: Terminal I7.
  - IA02: Terminal I8.
  - IA03: Terminal I11.
  - IA04: Terminal I12.
- Analog output QA01.
- Output ...QV of a different function block

If they are not associated, the inputs have a default value of:

- F1 and F2 = 1
- OS and HY = 0

Function of the function block

The function block compares values present at the inputs I1 and I2. You control the comparison in the function block by setting other input parameters:

- Multiplication factors F1 and F2 for the comparison values 1 and 2,
- Offset of comparison value 1 via function block input OS,
- Application of a switch hysteresis to comparison value 2 on HY.

Contact Q1 indicates the comparison result “greater”, “equal to” or “less” depending on the set operating mode.

Contact CY closes if the value range is exceeded. If an out-of-range value is signalled with CY = 1, Q1 remains in state 0. With the following formulae you can calculate the permissible value range.

$-2^{31} \leqq I1\ \ F1 + OS \leqq (2^{31} - 1) \rightarrow CY = 0$

$-2^{31} \leqq I2\ \ F2 + HY \leqq (2^{31} - 1) \rightarrow CY = 0$

$-2^{31} \leqq I2\ \ F2 - HY \leqq (2^{31} - 1) \rightarrow CY = 0$

The following signal diagram illustrates the function:

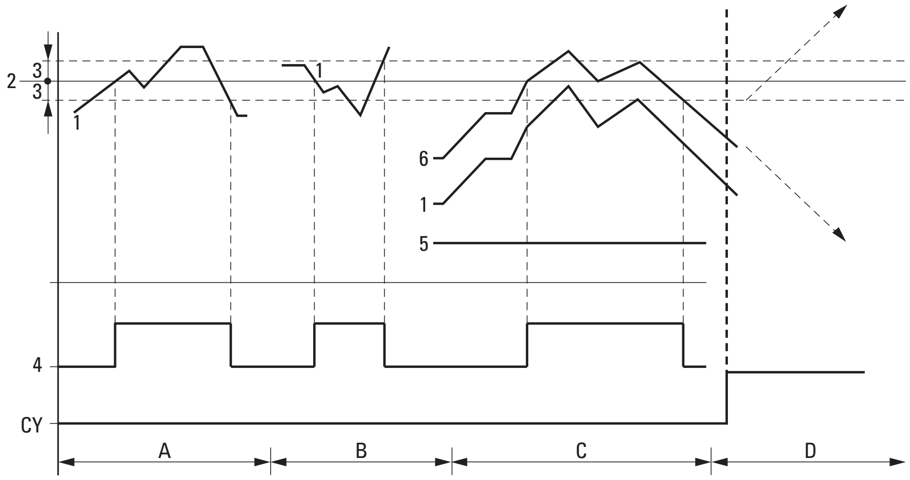


Figure 103:Signal diagram for analog value comparator

1: actual value at I1

2: Setpoint value on I2

3: Hysteresis on HY

4: Switching contact Q1 (N/O contact)

5: Offset for value I1

6: Actual value plus offset

- Range A: Compare I1 greater than I2
  - The actual value I1 increases.
  - The contact switches when the actual reaches the setpoint value.
  - The actual value changes and falls below the value of the setpoint value minus the hysteresis.
  - The contact goes to the normal position.
- Range B: Compare I1 less than I2
  - The actual value drops.
  - The contact switches if the actual value reaches the setpoint value.
  - The actual value changes and rises above the value of the setpoint value plus hysteresis.
  - The contact goes to the normal position.

- Range C: Compare I1 with Offset greater than I2
  - This example behaves as described in Range A. The offset value is added to the actual value.
  - Comparison I1 equals I2
    The contact switches on:
  - If I1 is equal to I2, i.e. the actual value is equal to the setpoint value.
    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.
- Range D: I1 with offset leaves the permissible value range. The contact CY closes.
  CY opens as soon as I1 with offset is once more within the value range.

Memory	68 + per 4 Byte/Constant at function block input	
Cycle time [µs] Mode = INI/CPY	easy800 standard	easy800-SWD
GT	96	22
EQ	96	22
LT	96	22

5.2 AR, Arithmetic

The Arithmetic function block performs all four basic calculations:

- add
- subtract
- multiply
- Divide.

easy800 allows the use of up to 32 arithmetic function blocks.

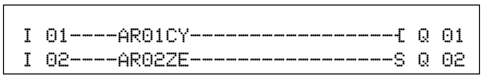
An arithmetic function block does not require an enable or start signal and is therefore not wired in a coil field of the standard circuit diagram. The function block is provided with the two Boolean outputs mentioned above for controlling the calculation result that you wire as contacts in the circuit diagram.

Contacts and Coils

A closed contact indicates:

Contacts	Meaning
AR..CY	Outside of the permissible value range
AR..ZE	Result at output QV> is zero

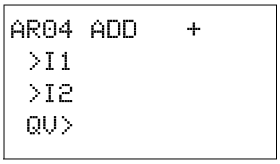
Example of an arithmetic function block



```
I 01----AR01CY-----------------[ Q 01
I 02----AR02ZE-----------------S Q 02
```

Figure 104:Wiring the function block contacts

Parametric programming of the function block inputs/outputs



```
AR04 ADD   +
 >I1
 >I2
 QV>
```

When using the function block in the circuit diagram for the first time, use OK to automatically enter the general display of function block parameters, as shown in the figure on the left. Enter the function block settings here. The display contains the following elements:

AR04	Function block: Arithmetic, number 04
ADD	Mode: Add
+	Parameter set can be called via the PARAMETERS menu
>I1	First value is associated with the value at I2 via the arithmetic operation, value range: -2147483648 - 2147483647
>I2	Second value, -2147483648 - 2147483647
QV>	Outputs the calculation result, value range: -2147483648 - 2147483647

Operating mode

This parameter is used to set the arithmetic operation required.

Parameter s	Function, parameterization in the function block
ADD	Addition of summand value >I1 plus Summand >I2
SUB	Subtraction of minuend >I1 minus Subtrahend >I2
MUL	Multiplication of factor >I1 by factor >I2
DIV	Division of dividend >I1 by divisor >I2

The factory setting of this parameter is ADD.

Inputs

I1 and I2: The signals present at the inputs I1 and I2 are associated with the operation set as the operating mode.

You can assign the following operands to function block inputs I1 and I2:

- Constant.
- Markers MD, MW, MB.
- Analog inputs IA01 to IA04:
  - IA01: Terminal I7.
  - IA02: Terminal I8.
  - IA03: Terminal I11.
  - IA04: Terminal I12.
- Analog output QA01.
- Output ...QV of a different function block.

Output

The output provides the calculation result as an integer. You can assign the following operands to output ...QV:

- Analog output QA01
- Markers MD, MW, MB and double word inputs of other function blocks.

Function of the function block

The values present at the inputs I1 and I2 are associated with the arithmetic operation set as the mode.

If the calculation result exceeds the value range that can be represented, the overflow signal contact CY closes and output QV contains the value of the last valid operation.

When the function block is called for the first time, the value at output QV equals zero.

Memory	40 + per 4 Byte/Constant at function block input	
Cycle time [µs]	easy800 standard	easy800-SWD
ADD	32	12
SUB	33	13
MUL	38	16
DIV	41	18

The following examples illustrate the functions of the function block for every arithmetic operation:

Addition examples

- 42 + 1000 = 1042
- 2147483647 + 1 = Last valid value before this arithmetic operation, due to overflow (CARRY) AR..CY = Status 1
- -2048 + 1000 = -1048

Subtraction examples

- 1134 - 42 = 1092
- -2147483648 - 3 = Last valid value before this arithmetic operation, due to overflow (CARRY) AR..CY = Status 1
- -4096 - 1000 = -5096
- -4096 - (-1000) = -3096

Multiplication examples

- 12 x 12 = 144
- 1000042 x 2401 = Last valid value before this arithmetic operation, due to overflow (CARRY) Correct value = 2401100842 AR..CY = Status 1
- -1000 x 10 = -10000

Division examples

- 1024 : 256 = 4
- 1024 : 35 = 29 (The places after the decimal point are omitted.)
- 1024 : 0 = Last valid value before this arithmetic operation, due to overflow (CARRY) (mathematically correct “Infinity”) AR..CY = Status 1
- -1000 : 10 = -100
- 1000 : (-10) = -100
- -1000 : (-10) = 100
- 10 : 100 = 0

5.3 BC, data block comparator

This function block is available for easy800-SWD from device version 01 and for easy800 standard devices from device version 04.

The data block comparator (BC = Block Compare) function block compares two contiguous marker ranges. For this you define the number of bytes to be compared. The comparison is carried out in byte format for marker types MB, MW and MD.

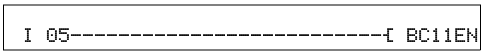
easy800 device allows the use of up to 32 data block comparators.

Contacts and Coils

Contacts	Meaning
BC..EQ	Compared ranges are identical.
BC..E1	The number of compared elements for a marker range are exceeded.
BC..E2	Compared ranges overlap.
BC..E3	Offset of compared blocks outside of the permitted range

Coils	Function
BC..EN	Selective activation of the function block

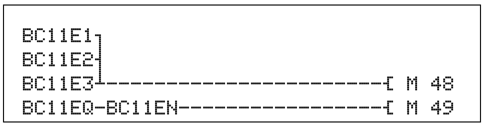
Example of a function block for comparing data blocks:



```
I 05--------------------------[ BC11EN
```

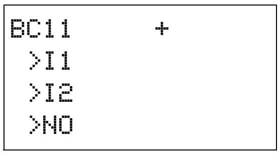
Figure 105:Wiring the enable coil

Figure 106:Wiring the contacts



```
BC11E1┐
BC11E2┤
BC11E3┴------------------------[ M 48
BC11EQ-BC11EN------------------[ M 49
```

Parametric programming of the function block inputs/outputs



```
BC11      +
 >I1
 >I2
 >NO
```

When using the function block in the circuit diagram for the first time, use OK to automatically enter the general display of function block parameters, as shown in the figure on the left. Enter the function block settings here. The display contains the following elements:

BC11	Function block: Data block comparator, number 11
+	Parameter set can be called via the PARAMETERS menu
>I1	Start of comparison range 1; the data block with the start address present at input I1 is compared with the data block with the start address present at input I2.
>I2	Start of comparison range 2
>NO	Number of elements to be compared in bytes per range, number: 1 - 383

Inputs

The marker ranges corresponding to the start addresses specified on inputs I1 and I2 are compared with each other.

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

- Constant.
- Markers MD, MW, MB.
- Analog inputs IA01 to IA04:
  - IA01: Terminal I7.
  - IA02: Terminal I8.
  - IA03: Terminal I11.
  - IA04: Terminal I12.
- Analog output QA01.
- Output ...QV of a different function block

Specifying the marker range

Marker ranges can be specified with or without an offset.

Marker range without offset

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

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:

- Analog input IA..,
- Analog output QA..,
- Constant.
- Output ..QV> of a function block.

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

Example:
A value 0 at input I1 means that the reference data block for the comparison starts at MB01. A value 100 at I2 means that the destination data block for the comparison starts at MB101.

→ The offset definition enables you to address the marker ranges (e.g. MB380) that you cannot address when using marker operands (direct addressing).

Offset calculation for addressing marker words:
Offset = MW (x - 1) x 2

Offset calculation for addressing marker double words:
Offset = MD (x - 1) x 4

Function of the function block

The data block comparator function block compares two contiguous data blocks.

The comparator is active if coil 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.

Memory	48 + per 4 Byte/Constant at function block input	
Cycle time [µs]	easy800 standard	easy800-SWD
1 Byte	52	18
10 Bytes	60	26
100 Bytes	138	48

Example 1: Comparison of marker blocks, direct definition of marker ranges

Two marker blocks are to be compared. Block 1 starts at MB10, Block 2 starts at MB40. Each block is 5 bytes long.

- Parameters of function block BC01:

Comparison range 1:	I1	MB10
Comparison range 2:	I2	MB40
Number of bytes:	NO	5

Comparison range 1	Value of marker range 1 (decimal)	Comparison range 2	Value of marker range 2 (decimal)
MB10	39	MB40	39
MB11	56	MB41	56
MB12	88	MB42	88
MB13	57	MB43	57
MB14	123	MB44	123

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

Example 2: 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 starts at MB65. Each block is 4 bytes long.

- Parameters of function block BC01:

Comparison range 1:	I1	MB15
Comparison range 2:	I2	64
Number of bytes:	NO	4
Marker MB01:		1

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

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

The comparison result of the function block BC01 is: BC01EQ = 0, the data block ranges do not have the same content. MB18 and MB68 are not identical.

Example 3: 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 starts at MD80. Each block is 6 bytes long.

- Parameters of function block BC01:

Comparison range 1:	I1	MB60
Comparison range 2:	I2	MD80
Number of bytes:	NO	6

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

Comparison range 1	Value of marker range 1 (decimal/binary)	Comparison range 2	Value of marker range 2 (decimal/ binary)
MB60	45/ 00101101	MD80 (Byte 1, LSB)	1097219629/ 01000001011001100011111000101101
MB61	62/ 00111110	MD80 (Byte 2)	1097219629/ 01000001011001100011111000101101
MB62	102/ 01100110	MD80 (Byte 3)	1097219629/ 01000001011001100011111000101101
MB63	65/ 01000001	MD80 (Byte 4, MSB)	1097219629/ 01000001011001100011111000101101
MB64	173/ 10101101	MD81 (Byte 1, LSB)	15277/ 0011101110101101
MB65	59/ 00111011	MD81 (Byte 2)	15277/ 0000100010101101

The data block ranges do not have the same content. BC01EQ = 0, MB65 and MD81 (Byte 2) are not identical.

Example 4:
Comparison of marker blocks, out-of-range value error

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

- Parameters of function block BC01:

• Comparison range 1:	I1	MD60
• Comparison range 2:	I2	MD90
• Number of bytes:	NO	30

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

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

BC01E1 is 1.

Example 5: Comparison of marker blocks, range overlap error

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

- Parameters of function block BC01:

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 6: 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 determined by the value of counter C 01QV.

- Parameters of function block BC01:

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 error message “The specified offset of the comparison ranges is outside of the permissible range” is output.

BC01E3 is 1.

5.4 BT, Data block Transformer

This function block is available for easy800-SWD from device version 01 and for easy800 standard devices from device version 04.

The block transfer function block (BT) is used for transferring values from one marker range to another (copy data). The marker ranges can be overwritten with a particular value (data initialization). The following marker ranges can be transferred and written:

- MB.
- MW.
- MD.

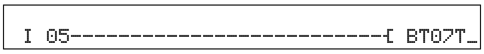
easy800 device allows the use of up to 32 data block transfer function blocks.

Contacts and Coils

Contacts	Meaning
BT..E1	Outside of source or destination range
BT..E2	Overlapping of source and destination range
BT..E3	Invalid offset

Coils	Function
BT..T_	Triggering the transfer on a rising edge

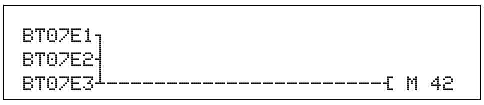
Example of a block transfer function block



```
I 05---------------------------[ BT07T_
```

Figure 107:Wiring the trigger coil

The trigger coil is connected to a device input.

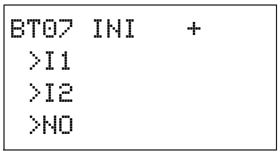


```
BT07E1┐
BT07E2┤
BT07E3┴------------------------[ M 42
```

Figure 108:Wiring the contacts

The messages of the function block are sent as a group message to a marker.

Parametric programming of the function block inputs/outputs



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

When using the function block in the circuit diagram for the first time, use OK to automatically enter the general display of function block parameters, as shown in the figure on the left. Enter the function block settings here. The display contains the following elements:

BT07	Function block: Data block transfer, number 07
INI	Operating mode: INI
+	Parameter set can be called via the PARAMETERS menu
>I1	Start of source range or initialization markers (MB,MW,MD)
>I2	Destination range start
>NO	Number of elements to be written in bytes per range, number: 1 - 383

Operating mode

The operating mode setting defines whether marker ranges are to be copied or initialized.

Parameters	Function
INI	Initialize marker ranges
CPY	Copy marker ranges

The factory setting of this parameter is INI.

Inputs

I1 and I2: Function block inputs I1 and I2 are used to define the beginning of the source and destination range.

The entry at function block input NO determines the number of elements to be written.

You can assign the following operands to function block inputs I1 and I2 and NO:

- Constant.
- Markers MD, MW, MB.
- Analog inputs IA01 to IA04:
  - IA01: Terminal I7.
  - IA02: Terminal I8.
  - IA03: Terminal I11.
  - IA04: Terminal I12.
- Analog output QA01.
- Output ...QV of a different function block.

Specifying the marker range

Marker ranges can be specified with or without an offset.

Marker range without offset

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

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.

- Analog input IA..
- Analog output QA..

- Constant.
- Output ..QV of a function block.

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

→ The offset definition enables you to address the marker ranges (e.g. MB380) that you cannot address when using marker operands (direct addressing).

Offset calculation for addressing marker words:
Offset = MW (x - 1) x 2

Offset calculation for addressing marker double words:
Offset = MD (x - 1) x 4

Function of the function block

The transfer data block function block has two operating modes.

→ No data blocks are initialized or copied in the event of an error.

Initializing 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 carries out a transfer on a rising edge at coil T_ (Trigger).

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

Memory	48 + per 4 Byte/Constant at function block input	
Cycle time [µs] Mode=INI/CPY	easy800 standard	easy800-SWD
1 Byte	64	18
10 Bytes	50	26
100 Bytes	71	42

Example 1:
Initializing marker blocks, direct definition marker ranges

The value of marker byte 10 is to be transferred to marker bytes 20 to 24.

- Parameters of function block BT01:
  - Source range: I1 MB10
  - Target range: I2 MB20
  - Number of bytes: NO 5

Source range	Value of source marker range (decimal)	Destination range	Value of destination marker range (decimal)
MB10	123	MB20	123
		MB21	123
		MB22	123
		MB23	123
		MB24	123

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

Example 2: Initialization 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 function block BT01:
  - Source range: I1 MB15
  - Target range: I2 64
  - Number of bytes: NO 4
  - Marker MB01: 1

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

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

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

Example 3:
Initialization of marker blocks, transfer of a range to a different format

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

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

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

Comparison range 1	Value marker range 1 (dec./binary)	Comparison range 2	Value of marker range 2 (decimal/binary)
MB60	45/ 00101101	MD80 (Byte 1, LSB)	757935405/ 00101101001011010010110100101101
		MD80 (Byte 2)	757935405/ 00101101001011010010110100101101
		MD80 (Byte 3)	757935405/ 00101101001011010010110100101101
		MD80 (Byte 4, MSB)	757935405/ 00101101001011010010110100101101
		MD81 (Byte 1, LSB)	757935405/ 00101101001011010010110100101101
		MD81 (Byte 2)	757935405/ 00101101001011010010110100101101
		MD81 (Byte 3)	757935405/ 00101100010110110010110100101101
		MD81 (Byte 4, MSB)	757935405/ 00101101001011010010110100101101

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 4: Transfer of marker byte, destination 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 function block BT01:

Source range:	I1	MD96
Target range:	I2	MD93
Number of bytes:	NO	18

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

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

BT01E1 is 1.

Example 5: 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.

Source range:	I1	MB40
Target range:	I2	MW54
Number of bytes:	NO	C 01QV

→ The value of C 01QV is 788. This value is too big.
The value at >NO can be between 1 and +192.

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

BT01E3 is 1.

CPY mode, copy marker ranges

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

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

The function block executes the copy operation if there is a rising edge from 1 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 1:
Copy of marker blocks, direct definition of marker ranges

The value of marker byte 10 to 14 is to be transferred to marker bytes 20 to 24.

- Parameters of function block BT01:

Source range:	I1	MB10
Target range:	I2	MB20
Number of bytes:	NO	5

Source range	Value of source marker range (decimal)	Destination range	Value of destination marker range (decimal)
MB10	42	MB20	42
MB11	27	MB21	27
MB12	179	MB22	179
MB13	205	MB23	205
MB14	253	MB24	253

After a rising edge from 0 to 1 at coil BT01T_ the content of MB10 to MB14 is copied to the marker bytes MB20 to MB24.

Example 2:
Copying of marker blocks, definition of a range with offset

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

- Parameters of function block BT01:

Source range:	I1	MB15
Target range:	I2	64
Number of bytes:	NO	4
Marker MB01:		1

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

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

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

Example 3:
Copying of marker blocks, definition of a range in a different format

Marker range MD60 to MD62 is to be copied to MW40 to MW45.

- Parameters of function block BT01:

Source range:	I1	MD60
Target range:	I2	MW40
Number of bytes:	NO	12

→ The transfer is in byte format. 12 bytes are to be copied.
The range MD60 to MD62 is 12 bytes. This is copied to the range MW40 to MW45.

Comparison range 1	Value of marker range 1 (decimal/binary)	Comparison range 2	Value of marker range 2 (decimal/binary)
MD60	866143319/ 0011001110100000 0100110001010111	MW40 (LSW)	19543/0011001110100000 0100110001010111
MD60	866143319/ 0011001110100000 0100110001010111	MW41 (MSW)	13216/0011001110100000 0100110001010111
MD61	173304101/0000101001010100 0110100100100101	MW42 (LSW)	26917/0000101001010100 0110100100100101
MD61	173304101/0000101001010100 0110100100100101	MB43 (MSW)	2644/0000101001010100 0110100100100101
MD62	982644150/ 0011101010010001 1111010110110110	MB44 (LSW)	62902/0011101010010001 1111010110110110
MD62	982644150/0011101010010001 1111010110110110	MB45 (MSW)	14993/0011101010010001 1111010110110110

After a rising edge from 0 to 1 at coil BT01T_ the values are copied to the appropriate range.

Example 4: Copying of marker bytes, destination range violation error

Marker range MB81 to MB96 is to be transferred to MD93 to MD96. The length is 16 bytes. The length is 16 bytes.

- Parameters of function block BT01:
  - Source range: I1 MB81
  - Target range: I2 MD93
  - Number of bytes: NO 18

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

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

BT01E1 is 1.

Example 5:
Comparison of marker blocks, range overlap error

12 bytes are to be copied starting from MW60. MW64 is specified as destination address.

- Parameters of function block BT01:

Source range:	I1	MW60
Target range:	I2	MW64
Number of bytes:	NO	12

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

The error message “Both ranges overlap” is output.

BT01E2 is 1.

Example 6:
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 function block BT01:

Source range:	I1	MM40
Target range:	I2	MW54
Number of bytes:	NO	C 01QV

→ The value of C 01QV is 10042. This value is too big. The value at >NO can be between 1 and +383.

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

BT01E3 is 1.

5.5 BV, Boolean operation

The function block enables the values to be combined in Boolean (logic) operations. It has the following functions:

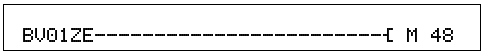
- Masking out of particular bits from values,
- Bit pattern recognition,
- Bit pattern modification.

easy800 device allows the use of up to 32 Boolean operations.

Contacts and Coils

Contacts	Meaning
BV..ZE	Result at output QV> is zero

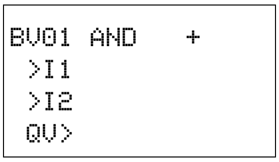
Example of a function block for Boolean operation



```
BV01ZE------------------------[ M 48
```

Figure 109:Wiring of the contact to marker

Parametric programming of the function block inputs/outputs



```
BV01 AND    +
 >I1
 >I2
 QV>
```

When using the function block in the circuit diagram for the first time, use OK to automatically enter the general display of function block parameters, as shown in the figure on the left. Enter the function block settings here. The display contains the following elements:

BV01	Function block: Boolean operation, number 01
AND	Operating mode: AND
+	Parameter set can be called via the PARAMETERS menu
>I1	First value (32-bit), the data block at this input is associated with the value at I2
>I2	Second value: (32-bit)
QV>	Result of the operation (32 Bit)

Operating mode

This parameter is used to set the type of operation required.

Parameter s	Function
AND	AND operation
OR	OR operation
XOR	Exclusive OR operation
NO T	Negation of the Boolean value of >I1

The factory setting of this parameter is AND.

Inputs

I1 and I2: At function block inputs I1 and I2 you write the values to be associated. You can assign the following operands to function block inputs I1 and I2:

- Constant.
- Markers MD, MW, MB.
- Analog inputs IA01 to IA04:
  - IA01: Terminal I7.
  - IA02: Terminal I8.
  - IA03: Terminal I11.
  - IA04: Terminal I12.
- Analog output QA01.
- Output ...QV of a different function block.

Output

The output ...QV> supplies the operation result. You can assign the following operands:

- Analog output QA01.
- Markers MD, MW, MB.

Function of the function block

The function block creates the operation depending on the operating mode.

Memory	48 + per 4 Byte/Constant at function block input	
Cycle time [µs]	easy800 standard	easy800-SWD
AND	27	10
OR	27	10
XOR	27	10
NO T	27	10

→ If you associate a negative value (e.g. -10_{dec}) the central processing unit will form the two's complement of the amount.

Example:

Magnitude(-10_{dez})	$00000000000000000000000000001010_{bin}$
Ones' complement formation	$11111111111111111111111111110101_{bin}$
adding 1_{bin}	1_{bin}
Binary result	$11111111111111111111111111110110_{bin}$
Hexadecimal result	$FFFFFFF6_{hex}$

Boolean associations

	Decimal	**Binary values**
AND		
Value >I1:	13219	0011001110100011
Value >I2:	57193	1101111101101001
Result QV>:	4897	0001001100100001
OR		
Value I1:	13219	0011001110100011
Value I2:	57193	1101111101101001
Result QV>:	65515	1111111111101011
XOR		
Value I1:	13219	0011001110100011
Value I2:	57193	1101111101101001
Result QV>:	60618	1110110011001010
NO T		
Value I1:	13219	00000000000000000011001110100011
Value I2:		Deleted
Result QV:	-13220	11111111111111111100110001011100

The NOT sequence operates according to the following rules:

I1, positive value
Subtract value of I1 and 1: –|I1| – 1 = I2

I1, negative value
Subtract value of I1 and 1: |I1| – 1 = I2

5.6 C, counter relay

The counter relay (function relay) counts pulses that are received at counter input C_. The count direction can be set.

Counting is carried out cyclically. The time of a pulse must therefore be greater than twice the cycle time. The CH high-speed counter can be used for shorter pulses.

You can set a lower and higher limit setpoint as comparison values for the counter relay function block, as well as a start value.

easy800 device allows the use of up to 32 counter relays.

Square wave count pulses are required to ensure a 1:1 mark to space ratio.
The relay counts only in RUN operating mode.

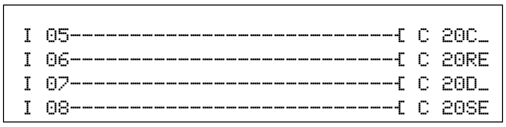
Page 269 on Section "Example with timing relay and counter function block" provides information on the timing relay and counter function block.

Contacts and Coils

Contacts	Meaning
C ..OF	Reaching or exceeding the upper limit setpoint
C ..FB	Reaching or falling below the lower limit setpoint
C ..ZE	Actual value at the function block output QV equals zero
C ..CY	Exceeding of the value range

Coils	Function
C ..C_	counter input, counts with every rising edge
C ..RE	Reset of the counter relay to zero
C ..D_	Count direction: 0: up counting, Status 1: down counting.
C ..SE	SET: Transfer of the start value (SV) on a rising edge.

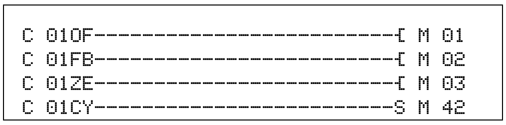
Example of a counter relay:



```
I 05---------------------------[ C 20C_
I 06---------------------------[ C 20RE
I 07---------------------------[ C 20D_
I 08---------------------------[ C 20SE
```

Figure 110:Wiring the function block coils

The function block coils are connected to the device inputs.

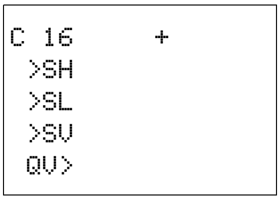


```
C 01OF--------------------------[ M 01
C 01FB--------------------------[ M 02
C 01ZE--------------------------[ M 03
C 01CY--------------------------S M 42
```

Figure 111:Wiring the contacts

The messages of the function block are sent to markers.

Parametric programming of the function block inputs/outputs



```
C 16      +
 >SH
 >SL
 >SV
 QV>
```

When using the function block in the circuit diagram for the first time, use OK to automatically enter the general display of function block parameters, as shown in the figure on the left. Enter the function block settings here. The display contains the following elements:

C 16	Function block: Counter relay, number 16
+	Parameter set can be called via the PARAMETERS menu
>SH	Upper setpoint,-2147483648 - 2147483647
>SL	Lower setpoint, -2147483648 - 2147483647
>SV	Start value (Defined actual value, Preset), -2147483648 - 2147483647
QV>	Current actual value, -2147483648 - 2147483647

Inputs

SH and SL: The upper and lower limit setpoints are set at function block inputs SH and SL respectively.

SV: The start value is used to cause the function block to count from a specified value (Offset).

Parameters	Function
>SH	Upper setpoint: -2147483648 - 2147483647
>SL	Lower setpoint: -2147483648 - 2147483647
>SV	Start value: -2147483648 - 2147483647

You can assign the following operands to function block inputs SH, SL and SV:

- Constant.
- Markers MD, MW, MB.
- Analog inputs IA01 to IA04:
  - IA01: Terminal I7.
  - IA02: Terminal I8.
  - IA03: Terminal I11.
  - IA04: Terminal I12.

- Analog output QA01.
- Output ...QV of a different function block.

Output

Actual value output C..QV supplies the current actual value as an integer.

Parameters	Function
QV>	Actual value: -2147483648 - 2147483647

You can assign the following operands to output ...QV:

- Analog output QA01.
- Markers MD, MW, MB.

Retention

Counter relays can be operated with retentive actual values. The number of retentive counter relays can be selected in the SYSTEM → RETENTION menu. The retentive actual value requires 4 bytes of memory.

If a counter relay is retentive, the actual value is retained when the operating mode changes from RUN to STOP as well as when the power supply is switched off. If easy800 is started in RUN operating mode, the counter relay operates with the retentively saved actual value.

Function of the function block

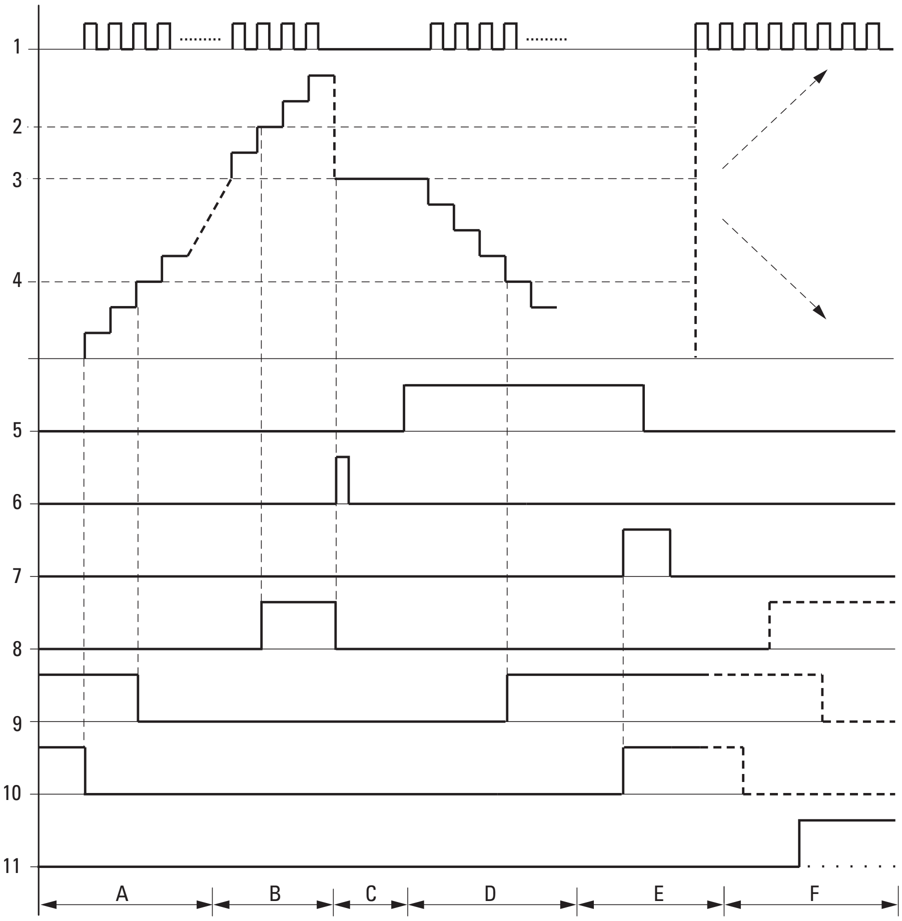


Figure 112:Signal diagram of counter relay

Legend for Figure 112:

1: Counter coil C..C_
2: Upper setpoint value >SH
3: Start value >SV.
4: Lower setpoint value >SL.
5: Counting direction, coil C..D_
6: Transfer start value, coil C..SE.
7: Reset coil C..RE
8: Contact (N/O) C..OF: Upper limit setpoint reached, exceeded
9: Contact (N/O) C..FB lower limit setpoint reached, undershot
10: Actual value equal to zero.
11: Out of value range

- Range A:
  - The counter range 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 pulses and increases the actual value.
  - C..ZE drops out as well as C..FB after the lower setpoint value is reached.
- Range B:
  - The counter relay 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 start value. The contacts go to the respective position.
- Range D:
  - The counting direction coil C..D_ is actuated. If counting pulses are present, downward counting 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 relay.
  - The contacts OF, FB, ZE become active according to the direction of the values (positive or negative).

Memory	52 + per 4 Byte/Constant at function block input	
Cycle time [µs]	easy800 standard	easy800-SWD
	46	12

5.7 CF, frequency counter

All devices with DC inputs except EASY802-DC-SWD contain CF function blocks that enable frequency measurement independent of the cycle time.

4 frequency counters are provided. The digital inputs I1 to I4 are permanently assigned to the frequency counter function blocks provided:

- I1: Counter input for counter CF01.
- I2: Counter input for counter CF02.
- I3: Counter input for counter CF03.
- I4: Counter input for counter CF04.

You can set a lower and higher limit setpoint comparison values for the frequency counter function block.

→ Square wave count pulses are required to ensure a 1:1 mark to space ratio.

The function block operates in the integer range from 4 to 5000 (1 kHz = 1000).

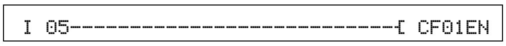
The value range cannot be exceeded as the maximum measured value is less than the value range.

Contacts and Coils

Contacts	Meaning
CF..OF	Reaching or exceeding the upper limit setpoint
CF..FB	Reaching or falling below the lower limit setpoint
CF..ZE	Actual value equals zero

Coils	Function
CF..EN	Selective activation of the function block

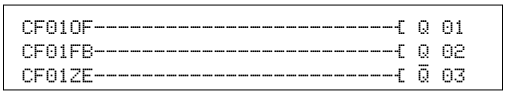
Example of a frequency counter



```
I 05---------------------------[ CF01EN
```

Figure 113:Wiring the function block coils

The counter input of the function block CF01 is already permanently connected to device input I1, the function of the function block is activated by the device input I5.



```
CF01OF-------------------------[ Q 01
CF01FB-------------------------[ Q 02
CF01ZE-------------------------[ Q̄ 03
```

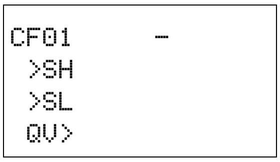
Figure 114:Wiring the function block contacts

The messages of the function block are sent directly to the device outputs.

Avoid unforeseeable switch states:

- Only use each function block coil once in the circuit diagram.
- If you use one of the inputs I1, I2, I3 or I4 as a counter input for the counter function block CF, CH or CI, they cannot be used for other purposes.
  If an input I1 … I4 is used several times, only the high-speed counter is run that is positioned last in the order of processing in the function block diagram.

Parametric programming of the function block inputs/outputs



```
CF01     -
 >SH
 >SL
 QV>
```

When using the function block in the circuit diagram for the first time, use OK to automatically enter the general display of function block parameters, as shown in the figure on the left. Enter the function block settings here. The display contains the following elements:

CF01	Function block: Frequency counter, number 01
–	Parameter display: Call not enabled
>SH	Upper setpoint: 0 - 5000 Hz
>SL	Lower setpoint: 0 - 5000 Hz
QV>	Current frequency value in RUN mode: 0…5000 Hz

Inputs

SH und SL: The upper and lower limit setpoints are set at SH and SL respectively. The function block inputs >SH and >SL can have the following operands:

- Constant.
- Markers MD, MW, MB.
- Analog inputs IA01 to IA04:
  - IA01: Terminal I7.
  - IA02: Terminal I8.
  - IA03: Terminal I11.
  - IA04: Terminal I12.
- Analog output QA01.
- Output ...QV of a different function block.

Output

QV: The output CF..QV outputs the actual frequency at the counter input as an integer. You can assign the following operands to output ...QV:

- Analog output QA01
- Markers MD, MW, MB

Retention

The frequency counter does not have any retentive actual values since the frequency is continuously remeasured.

Function of the function block

The first measurement is carried out when the function block is enabled via the signal at CF..EN. The value is output after the gate time (t_g) 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.

The pulses on the input are counted for one second irrespective of the cycle time, and the frequency is determined. The measurement result is provided at the function block output CF..QV.

Memory	40 + per 4 Byte/Constant at function block input	
Cycle time [µs]	easy800 standard	easy800-SWD
	30	8

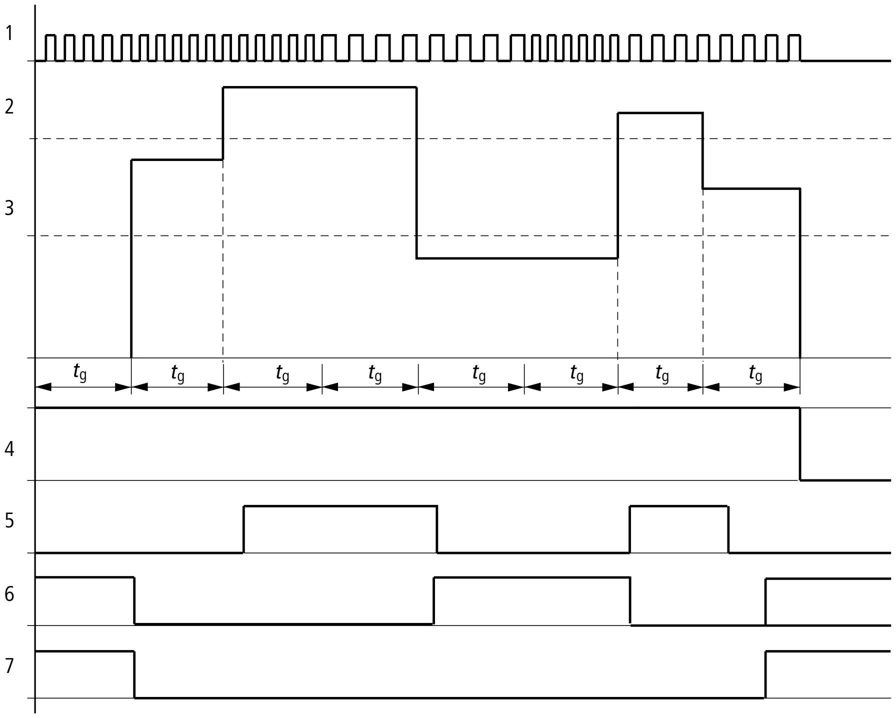


Figure 115:Signal diagram of frequency counter

1: Counter input I1 to I4.
2: Upper setpoint value >SH
3: Lower setpoint value >SL
4: Enable CF..EN
5: Contact (N/O) CF..OF: Upper setpoint value exceeded
6: Contact (N/O) CF..FB: Lower setpoint value undershot
7: Actual value equal to zero CF..ZE

5.8 CH, high-speed counter

All devices with DC inputs apart from EASY802-DC-SWD contain CH function blocks that enable up and down counting of rising edges independently of the cycle time.

You can set a lower and higher limit as comparison values for the high-speed counter function block, as well as a start value.

Four high-speed function blocks are provided.

→ Square wave count pulses are required to ensure a 1:1 mark to space ratio.

The maximum count frequency is 5000 Hz.

→ Note that the digital inputs I1 to I4 are permanently assigned to the frequency counter function blocks provided:

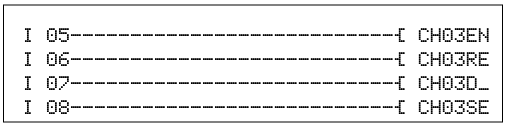
- I1: Counter input for counter CH01.
- I2: Counter input for counter CH02.
- I3: Counter input for counter CH03.
- I4: Counter input for counter CH04.

Contacts and Coils

Contacts	Meaning
CH..OF	Reaching or exceeding the upper limit setpoint
CH..FB	Reaching or falling below the lower limit setpoint
CH..ZE	Actual value equals zero
CH..CY	Value range exceeded

Coils	Function
CH..EN	Selective activation of the function block
CH..RE	Reset the actual value to zero
CH..D_	Count direction Picked up: up counting, Not picked up: down counting.
CH..SE	Transfer of the start value on a rising edge.

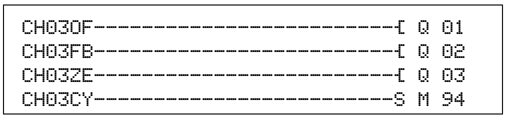
High-speed counter example



```
I 05----------------------------[ CH03EN
I 06----------------------------[ CH03RE
I 07----------------------------[ CH03D_
I 08----------------------------[ CH03SE
```

Figure 116:Wiring the function block coils

The counter input of the function block CH03 is already permanently connected to device input I3. The coils of the function block are connected directly to the device inputs. The coils of the function block are connected directly here to the device inputs.



```
CH03OF--------------------------Ɛ Q 01
CH03FB--------------------------Ɛ Q 02
CH03ZE--------------------------Ɛ Q 03
CH03CY--------------------------S M 94
```

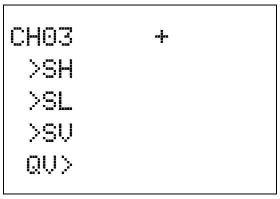
Figure 117:Wiring the function block contacts

The contacts of the function block are connected to the device outputs and a marker.

→ Avoid unforeseeable switch states:

- Only use each function block coil once in the circuit diagram.
- If you use one of the inputs I1, I2, I3 or I4 as a counter input for the counter function block CF, CH or CI, they cannot be used for other purposes.
  If an input I1…I4 is used several times, only the high-speed counter is run that is positioned last in the order of processing in the function block diagram

Parametric programming of the function block inputs/outputs



```
CH03     +
 >SH
 >SL
 >SV
 QV>
```

When using the function block in the circuit diagram for the first time, use OK to automatically enter the general display of function block parameters, as shown in the figure on the left. Enter the function block settings here.

The display contains the following elements:

CH03	Function block: High-speed counter, number 03
+	Parameter set can be called via the PARAMETERS menu
>SH	Upper setpoint: -2147483648 - 2147483647
>SL	Lower setpoint: -2147483648 - 2147483647
>SV	Start value (Preset): -2147483648 - 2147483647
QV>	Actual value: -2147483648 - 2147483647

Inputs

SH and SL: The upper and lower limit setpoints are set at SH and SL respectively.

SV: The start value is used to cause the function block to count from a specified value (Offset).

You can assign the following operands to function block inputs SH, SL and SV:

- Constant.
- Markers MD, MW, MB.
- Analog inputs IA01 to IA04:
  - IA01: Terminal I7.
  - IA02: Terminal I8.
  - IA03: Terminal I11.
  - IA04: Terminal I12.
- Analog output QA01.
- Output ...QV of a different function block.

Output

QV>: Actual value output CH..QV outputs the current actual value as an integer. You can assign the following operands to output ...QV:

- Analog output QA01.
- Markers MD, MW, MB.

→ The actual value can only be cleared in RUN mode with a specific reset signal.

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 meter is retentive, the actual value is retained when the operating mode changes from RUN to STOP as well as when the supply voltage is switched off.

If a easy800 is started in RUN mode, the counter operates with the retentively saved actual value.

Function of the function block

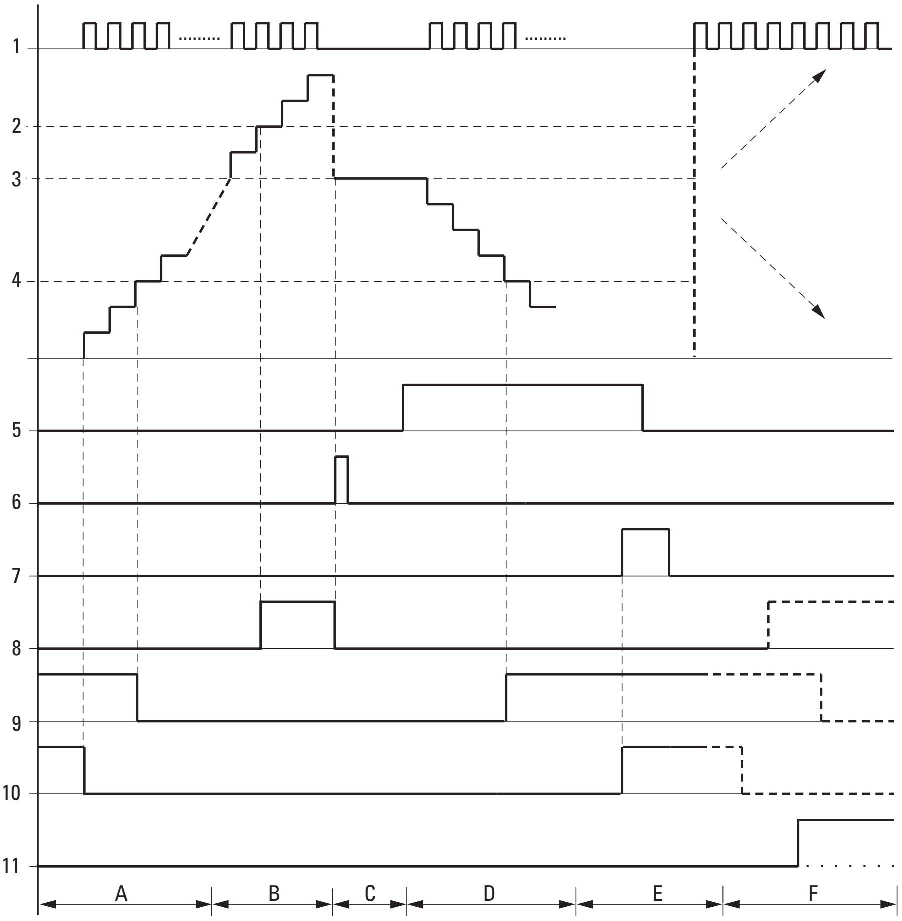


Figure 118:Signal diagram of high-speed counter

1: Counter input I1 to I4.

2: Upper setpoint value >SH

3: Start value >SV.

4: Lower setpoint value >SL.

5: Count direction, coil CH…D

6: Transfer start value, coil CH..SE.

7: Reset coil CH…RE

8: Contact (N/O) CH…OF: Upper setpoint value reached or exceeded.

9: Contact (N/O) CH…FB: Lower setpoint value reached or undershot.

10: Contact (N/O) CH…ZE: Actual value equal to zero

11:Out of value range.

- Range A:
  - The counter has the value zero.
  - The contacts CH..ZE (actual value = zero) and CH..FB (lower setpoint value undershot) are active.
  - The counter receives pulses and increases the actual value.
  - CH..ZE as well as CH..FB drop out when 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 start 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 OF, FB, ZE become active according to the direction of the values (positive or negative).

Memory	52 + per 4 Byte/Constant at function block input	
Cycle time [µs]	easy800 standard	easy800-SWD
	52	13

5.9 CI, High-speed incremental value counter

All devices with DC inputs apart from EASY802-DC-SWD enable high-speed up and down counting on a rising or falling edge. The counting operation is independent of the cycle time.

You can set a lower and higher setpoint value as comparison values for the incremental counter function block, as well as a start value.

There are 2 incremental counter available.

Square wave count pulses are required to ensure a 1:1 mark to space ratio.
The signals of channels A and B must be offset by 90° otherwise the count direction will not be detected.

The maximum counter frequency for easy800 standard is 3000 Hz and 5000 Hz for EASY806-DC-SWD.

Note that the digital inputs I1 to I4 are permanently assigned to the incremental counter function blocks provided:

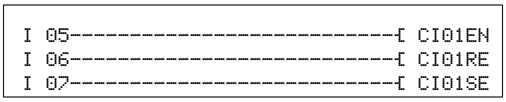
- I1: Counter input for counter CI01, channel A.
- I2: Counter input for counter CI01, channel B.
- I3: Counter input for counter CI02, channel A.
- I4: Counter input for counter CI02, channel B.

Contacts and Coils

Contacts	Meaning
CI..OF	Reaching or exceeding the upper limit setpoint
CI..FB	Reaching or falling below the lower limit setpoint
CI..ZE	Actual value equals zero
CI..CY	Value range exceeded

Coils	Function
CI..EN	Selective activation of the function block
CI..RE	Reset the actual value to zero
CI..SE	Transfer of the start value on a rising edge.

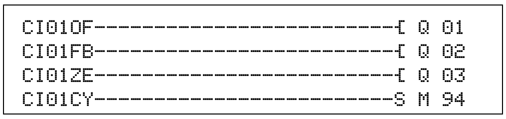
Example of a high-speed incremental counter



```
I 05----------------------------[ CI01EN
I 06----------------------------[ CI01RE
I 07----------------------------[ CI01SE
```

Figure 119:Wiring the function block coils

The count inputs of function block CI01 are already permanently assigned to device inputs I1 and I2. The coils of the function block are connected directly here to the device inputs.



```
CI01OF-------------------------[ Q 01
CI01FB-------------------------[ Q 02
CI01ZE-------------------------[ Q 03
CI01CY-------------------------S M 94
```

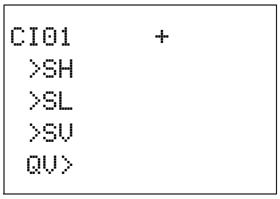
Figure 120:Wiring the function block contacts

The contacts of the function block are connected to the device outputs and a marker.

→ Avoid unforeseeable switch states:

- Only use each function block coil once in the circuit diagram.
- If you use one of the inputs I1, I2, I3 or I4 as a counter input for the counter function block CF, CH or CI, they cannot be used for other purposes.
  If an input I1…I4 is used several times, only the high-speed counter is run that is positioned last in the order of processing in the function block diagram

Parametric programming of the function block inputs/outputs



```
CI01      +
 >SH
 >SL
 >SV
 QV>
```

When using the function block in the circuit diagram for the first time, use OK to automatically enter the general display of function block parameters, as shown in the figure on the left. Enter the function block settings here. The display contains the following elements:

CI01	Function block: High-speed Incremental value counter, number 01
+	Parameter set can be called via the PARAMETERS menu
>SH	Upper setpoint: -2147483648 - 2147483647
>SL	Lower setpoint: -2147483648 - 2147483647
>SV	Start value (Preset): -2147483648 - 2147483647
QV>	Current actual value: -2147483648 - 2147483647

Inputs

SH and SL: The upper and lower limit setpoints are set at function block inputs SH and SL respectively.

SV: The start value is used to cause the function block to count from a specified value (Offset).

You can assign the following operands to function block inputs SH, SL and SV:

- Constant.
- Markers MD, MW, MB.
- Analog inputs IA01 to IA04:
  - IA01: Terminal I7.
  - IA02: Terminal I8.
  - IA03: Terminal I11.
  - IA04: Terminal I12.
- Analog output QA01.
- Output ...QV of a different function block.

Output

QV: Actual value output CI..QV outputs the current actual value as an integer. You can assign the following operands to output ...QV:

- Analog output QA01.
- Markers MD, MW, MB.

→ The actual value can only be cleared in RUN mode with a Reset signal.

Retention

The counter can be operated with retentive actual values. The number of retentive counters is selected in SYSTEM → RETENTION.

If a meter is retentive, the actual value is retained when the operating mode changes from RUN to STOP as well as when the supply voltage is switched off.

If a easy800 is started in RUN mode, the counter operates with the retentively saved actual value.

Function of the function block

Memory	52 + per 4 Byte/Constant at function block input	
Cycle time [µs]	easy800 standard	easy800-SWD
	47	12

→ The incremental counter evaluates the rising and falling edges and counts twice the number of pulses. This prevents too many or too few pulses from being counted if there is oscillation on a signal edge. If the number of pulses are required, divide the value by two.

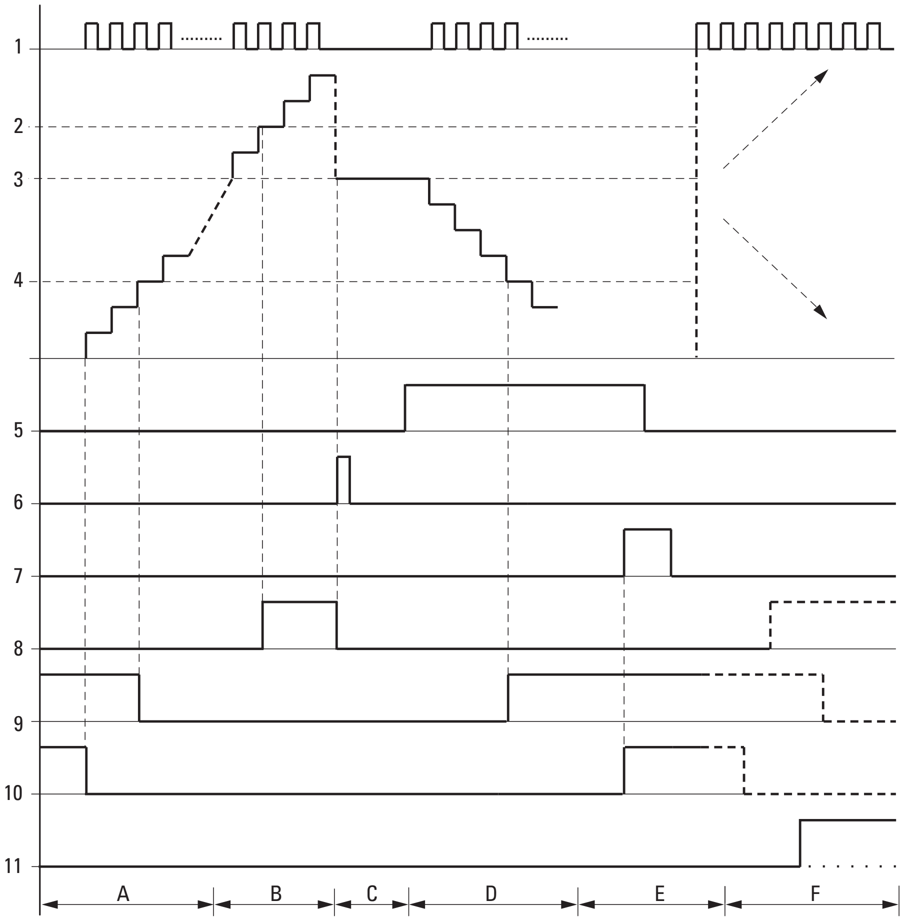


Figure 121:Signal diagram High-speed incremental value counter

1: Counter input I1 to I4.

2: Upper setpoint value SH.

3: Start value SV.

4: Lower setpoint value SL.

5: Counting direction, coil CI…D

6: Transfer start value, coil CI..SE.

7: Reset coil CI…RE

8: Contact (N/O) CI…OF: Upper setpoint value reached or exceeded.

9: Contact (N/O) CI…FB: Lower setpoint value reached or undershot.

10: Contact (N/O) CI…ZE: Actual value equal to zero

11:Out of value range.

- Range A:
  - The counter has the value zero.
  - The contacts CI..ZE (actual value = zero) and CI..FB (lower setpoint value undershot) are active.
  - The counter receives pulses and increases the actual value.
  - CI..ZE drops out as well as CI..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 CI..OF becomes active.
- Range C:
  - The coil CI..SE is briefly actuated and the actual value is set to the start value.
    The contacts go to the respective position.

- Range D:
  - The counting direction coil CI…D is actuated.
    If counting pulses are present, downward counting is initiated.
  - If the lower setpoint value is undershot, the contact CI..FB becomes active.
- Range E:
  - The reset coil CI..RE is activated. The actual value is set to zero.
  - The contact CI..ZE is active.
- Range F:
  - The actual value goes outside the value range of the counter.
  - The contacts OF, FB, ZE become active according to the direction of the values (positive or negative).

5.10 CP, comparator

This function block is used to compare variables and/or constants with each other. The following scans are possible:

- I1 greater than I2.
- I1 equal to I2.
- I1 less than I2.

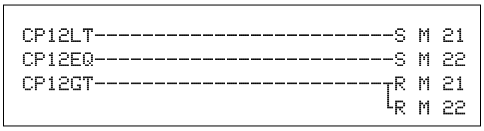
easy800 device allows the use of up to 32 comparators.

Contacts and Coils

A closed contact indicates:

Contacts	Meaning
CP..LT	>I1 less than >I2
CP..EQ	>I1 equal to >I2
CP..GT	>I1 greater than >I2

Example:

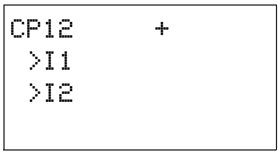


```
CP12LT-------------------------S M 21
CP12EQ-------------------------S M 22
CP12GT------------------------┬R M 21
                              └R M 22
```

Figure 122:Wiring the function block contacts

The contact of the function block is sent to markers.

Parametric programming of the function block inputs/outputs



```
CP12     +
 >I1
 >I2
```

When using the function block in the circuit diagram for the first time, use OK to automatically enter the general display of function block parameters, as shown in the figure on the left. Enter the function block settings here. The display contains the following elements:

CP12	Function block: Data block comparator, number 12
+	Parameter set can be called via the PARAMETERS menu
>I1	First value to be compared with the value at I2: -2147483648 - 2147483647
>I2	Second value: -2147483648 - 2147483647

Inputs

I1 and I2: The signals present at inputs I1 and I2 are compared with each other. You can assign the following operands to function block inputs:

- Constant.
- Markers MD, MW, MB.
- Analog inputs IA01 to IA04:
  - IA01: Terminal I7.
  - IA02: Terminal I8.
  - IA03: Terminal I11.
  - IA04: Terminal I12.

- Analog output QA01.
- Output ...QV of a different function block.

Function of the function block

The function block compares values present at the inputs I1 and I2.

The following contacts close depending on the comparison result:

- I1 greater than I2, GT contact closes.
- I1 equal to I2, EQ contact closes.
- I1 less than I2, LT contact closes.

Memory	32 + per 4 Byte/Constant at function block input	
Cycle time [µs]	easy800 standard	easy800-SWD
	24	8

5.11 D, text display

This function block is used to output text with/without variables on the device display. These variables can be actual values and setpoints of function blocks, marker values, analog inputs/outputs as well as date and time. The display/keypad allows you to enter a setpoint for a function block if it is defined as a constant. The SP function block can be used to output the text and the variables via the multi-function interface to a peripheral device such as a terminal.

A easy800 device allows the use of up to 32 function blocks for text display.

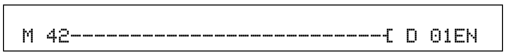
The programming software easySoft-Pro is used for editing the individual text lines in the Parameters tab. Carry this out for every D function block used. This is where you also set the variable to be shown. A detailed description of this topic is provided in the easySoft-Pro online help.

5.11.0.1 Contacts and Coils

Contacts	Meaning
D ..Q1	Display of text active

Coils	Function
D ..EN	Selective activation of the function block

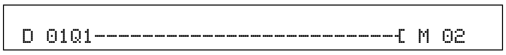
Example of a text function block



```
M 42---------------------------[ D 01EN
```

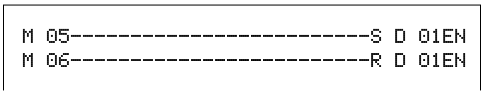
Figure 123:Wiring the enable coil

The enable coil of the function block is triggered by a marker.



```
D 01Q1--------------------------[ M 02
```

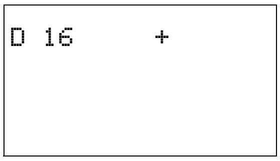
Figure 124:Wiring of the function block contact



```
M 05-------------------------S D 01EN
M 06-------------------------R D 01EN
```

Figure 125:Wiring the enable coil with Set/Reset

Parametric programming of the function block inputs/outputs

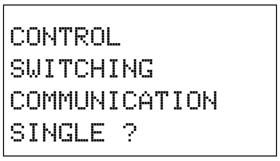


```
D 16      +
```

When using the function block in the circuit diagram for the first time, use OK to automatically enter the general display of function block parameters, as shown in the figure on the left. Enter the function block settings here. The display contains the following elements:

D16	Function block: Text output, number 16
+	Parameter set can be called via the PARAMETERS menu

Display



```
CONTROL
SWITCHING
COMMUNICATION
SINGLE ?
```

Up to 16 characters per line and up to 4 lines can be displayed.

Displaying a variable

The function block enables the display of:

- Actual values/setpoints of all function blocks except for HW and HY.
- Markers (MB, MW and MD).
- Analog inputs (scaleable)
- Analog outputs.
- Date and time.

The variables and setpoints can be inserted at any desired position in the single-line text. The variables can be 4, 5, 7 and 11 places long. Incorrect entries can cause characters to be overwritten or hidden.

The following formats for timing relay, time and date are supported.

Variable	Format
Timing relay	s s s . s s s
	min min : s s
	h h : min min
Time	h h : min min
Date	DD.MM : YYYY

Entering setpoints via keypad/display

The function block setpoints shown on the display and which can be defined as constants can be edited. The possibility of editing must be activated in the programming software. The values can be 4, 5, 7 and 11 places long. Incorrect entries can cause characters to be overwritten or hidden.

Scaling

The values of the analog inputs an outputs can be scaled.

Area	Selectable display range	Example
0 - 10 V	0 - 9999	0000 - 0100
0 - 10 V	± 999	-025 - 050
0 - 10 V	± 9.9	-5.0 - 5.0

Operation of the function block

The text is output if you activate the EN coil. For this the status display must be active before the text display is called.

If the coil (EN) is returned to 0, the control relay display reactivates the status display after 4 s.

The additional Central Europe and Cyrillic fonts can also be used for the text display and combined with the display attributes Flashing and Inverse. These options can be set for individual lines, and the attributes can be used in any combination.

Different behaviour of the function blocks D01 and D02 to D32

D01 outputs an alarm text i. e., If D01 is activated, the text assigned to it will be displayed until:

- The coil D01 is reset to 0.
- STOP operating mode is selected.
- The easy800 device is switched off.
- A menu was selected via OK or DEL + ALT.

D 02 to D 32: If several texts are present and are triggered, each text is automatically displayed in turn every 4 s. This process will be repeated until:

- No other text display function block is set to 1.
- STOP operating mode is selected.
- The easy800 device is switched off.
- The OK or DEL + ALT pushbuttons are used to switch to a menu.
- The text stored for D01 is displayed.

The function of the text display function is fully simulated by the Simulation tool.

Memory	48 + per 4 Byte/Constant at function block input

Permissible Character Set

You can enter characters from the character maps “Western European”, “Central European” and “Cyrillic”.

Example characters from the character map “Western European”:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

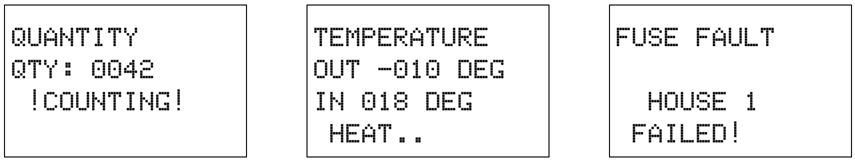
a b c d e f g h i j k l m n o p q r s t u v w x y z

0 1 2 3 4 5 6 7 8 9

Examples of special characters:

° ! " # $ % & ' () * + , - . / : ; < = > ? [\] ^ _ ` { | } ~. and space.

Display examples



Counter with actual value

```
QUANTITY
QTY: 0042
 !COUNTING!
```

Temperature values of an easyNet station

```
TEMPERATURE
OUT -010 DEG
IN 018 DEG
 HEAT..
```

D1 as error message on fuse failure

```
FUSE FAULT

  HOUSE 1
 FAILED!
```

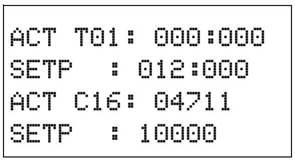
Figure 126:Text output examples

Setpoint entry in a display

This function is available for easy800-SWD from device version 01 and for easy800 standard from device version 04.

→ When values are being entered, the text is retained statically on the display. The actual values are updated.

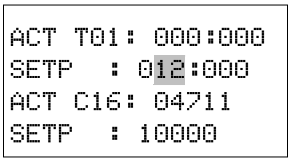
Example:



```
ACT T01: 000:000
SETP   : 012:000
ACT C16: 04711
SETP   : 10000
```

The setpoint of timing relay T01 is to be changed from 12 s to 15 s.

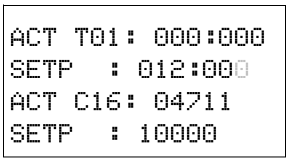
- Line 1: Actual value of timing relay T 01
- Line 2: Setpoint of timing relay T 01, editable



```
ACT T01: 000:000
SETP   : 012:000
ACT C16: 04711
SETP   : 10000
```

▶ Pressing the ALT pushbutton will cause the cursor to jump to the first editable value.

In this operating mode, you can use the cursor buttons ⌃⌄ to move between editable constants.

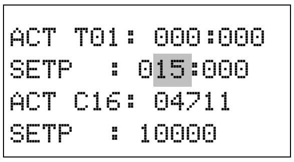


```
ACT T01: 000:000
SETP   : 012:000
ACT C16: 04711
SETP   : 10000
```

▶ Press the OK pushbutton to move the cursor to the least significant value of the constant to be modified.

In this operating mode you use the ⌃⌄ cursor buttons to change the value. Use the cursor buttons ‹› to move between places.

Use the OK pushbutton to accept the modified value. Use the ESC pushbutton to abort the entry and leave the previous value.



```
ACT T01: 000:000
SETP   : 015:000
ACT C16: 04711
SETP   : 10000
```

▶ Press the OK pushbutton, the cursor will move from constant to constant.

The modified value is accepted.

Press the ALT pushbutton in order to leave Entry mode.
(The ESC pushbutton has the same function here.)

5.12 DB, data block

The function block stores values such as setpoints for function blocks. The value applied to input I1 is saved in the operand linked to output QV.

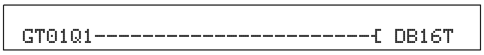
easy800 device allows the use of up to 32 data function blocks.

Contacts and Coils

Contacts	Meaning
DB..Q1	Transfer confirmation with 1, if T_ is 1.

Coils	Function
DB..T_	Transfer the value to >I1 with a rising edge.

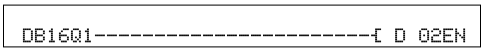
Example of a data function block



```
GT01Q1-----------------------Ɛ DB16T
```

Figure 127:Wiring the trigger coil

The trigger coil is addressed via the network.

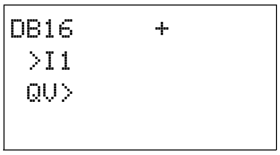


```
DB16Q1-----------------------Ɛ D 02EN
```

Figure 128:Wiring of the function block contact

The output of the data function block DB16Q1 is assigned to the input D02 EN of the text display function block.

Parametric programming of the function block inputs/outputs



```
DB16     +
 >I1
 QV>
```

When using the function block in the circuit diagram for the first time, use OK to automatically enter the general display of function block parameters, as shown in the figure on the left. Enter the function block settings here.

The display contains the following elements:

DB16	Function block: Data function block, number 16
+	Parameter set can be called via the PARAMETERS menu
>I1	Input value: -2147483648 - 2147483647
QV>	Outputs the value of DB..I1 when triggered: -2147483648 - 2147483647

Input

I1: You apply the value that is transferred on triggering to the function block input I1. The function block input I1 can be assigned the following operands:

- Constant.
- Markers MD, MW, MB.
- Analog inputs IA01 to IA04:
  - IA01: Terminal I7.
  - IA02: Terminal I8.
  - IA03: Terminal I11.
  - IA04: Terminal I12.

- Analog output QA01.
- Output ...QV of a different function block.

Output

QV: The output provides the input value when triggered as an integer. You can assign the following operands to output ...QV:

- Analog output QA01.
- Markers MD, MW, MB.

Retention

Data function blocks can be operated with retentive actual values. The quantity can be selected in the SYSTEM → RETENTION menu.

Function of the function block

The value at input >I1 is only transferred with a rising trigger edge to an operand (e.g.: MD42) at output QV. It contains the value up to the next time it is overwritten.

Memory	36 + per 4 Byte/Constant at function block input	
Cycle time [µs]	easy800 standard	easy800-SWD
	28	9

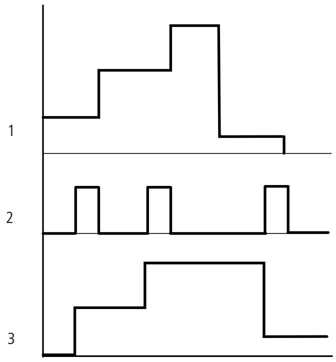


Figure 129:Signal diagram of data function block

1: Value at input DB..>I1
2: Trigger coil DB…T_
3: Value on DB…QV>

5.13 DC, PID controller

This function block is available for easy800-SWD from device version 01 and for easy800 standard devices from device version 04.

32 PID controllers are provided.

DANGER

A knowledge of closed-loop control is required in order to use the PID controllers.
The control system must be familiar to the user so that the PID controller can function correctly.

→ Three manipulated variables can be output independently of each other:

- One manipulated variable can be output via an analog output.
- Two other manipulated variables can be processed via two pulse-width modulated outputs.

However, it is in most cases useful to run no more than three closed-loop controllers per program simultaneously. If your application requires more than three closed-loop controllers, distribute them over several programs. You can structure your projects 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.

→ If you are using the function block in time consuming applications such as in conjunction with an FT signal smoothing filter, you should not program any other time consuming applications on this device, such as visualizations on the MFD. The refreshing of screen(s) can increase the cycle time and thus restrict the controller function. Distribute the applications among two or more devices that you can interconnect via easyNet.

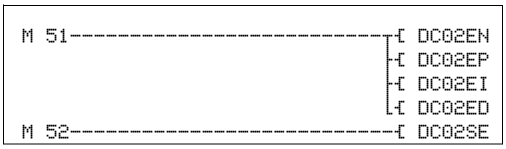
Contacts and Coils

Contacts	Meaning
DC..LI	Closes when exceeding the manipulated value range.

You use the coils of the function block to activate it, the individual elements of the PID controller and manual mode.

Coils	Function
DC..EN	Selective activation of the function block
DC..EP	Activation the proportional function
DC..EI	Activating the integral function
DC..ED	Activating the differential function
DC..SE	Activating the manual manipulated variable, manual mode

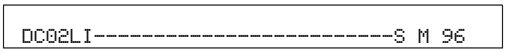
Example of a PID controller



```
M 51--------------------------┬-[ DC02EN
                              ├-[ DC02EP
                              ├-[ DC02EI
                              └-[ DC02ED
M 52----------------------------[ DC02SE
```

Figure 130:Wiring the function block coils

The function block coils are activated by markers.

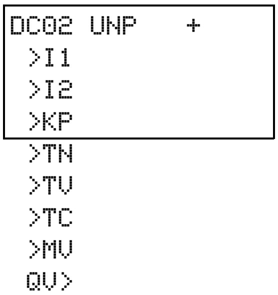


```
DC02LI-------------------------S M 96
```

Figure 131:Wiring the contacts

The message of the function block is sent to a marker.

Parametric programming of the function block inputs/outputs



```
DC02 UNP    +
 >I1
 >I2
 >KP
 >TN
 >TV
 >TC
 >MV
 QV>
```

When using the function block in the circuit diagram for the first time, use OK to automatically enter the general display of function block parameters, as shown in the figure on the left. Enter the function block settings here. The display contains the following elements:

DC02	Function block: PID controller, number 02
UNP	Operating mode: Unipolar
+	Parameter set can be called via the PARAMETERS menu
>I1	Setpoint of PID controller: -32768 - +32767
>I2	Actual value of PID controller: -32768 - +32767
>KP	Proportional Gain K_p; 0 - 65535, in %; Example: The value 1500 is processed in the function block as 15.
>TN	Reset Time T_n: 0 - 65535, in 100 ms; Example: The value 250 is processed in the function block as 25 s.
>TV	Derivate Action Time T_v: 0 - 65535, in 100 ms; Example: The value 20 is processed in the function block as 2 s.
>TC	Scan Time T_c: 0 - 65535, in 100 ms
>MV	Manual manipulated variable: -4096 - +4095
QV>	Manipulated variable: • Unipolar: 0 - 4095 • Bipolar: -4096 - +4095

Operating mode

The operating mode defines the value range of the manipulated variable.

Parameters	Function
UNP	Unipolar 12-bit value: 0 to +4095
BIP	Bipolar 13-bit value: (signed 12-bit value) -4096 - +4095

Inputs

I1 and I2: The PID controller setpoint and actual values are set at function block inputs I1 and I2.

KP: Input KP is used to adjust the control difference to the required proportional gain.

TN: The value for the reset time T_n at input TN is used for forming the integral component.

TV: The value for the rate time T_v at input TV is used for forming the differential component.

TC: The value for the scan time T_c at input TC determines the time intervals in which the operating system calls the function block for processing. 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 (→ section 5.31, “ST, Set cycle time”, page 253) in order to keep the cycle time of the device constant.

MV: The set value of the manual manipulated variable M_V is output on the output DC..QV as a manipulated variable during activation of coil DC..SE.

You can assign the following operands to function block inputs I1, I2, KP, TN, TV, TC and MV:

- Constant.
- Markers MD, MW, MB.
- Analog inputs IA01 to IA04:
  - IA01: Terminal I7.
  - IA02: Terminal I8.
  - IA03: Terminal I11.
  - IA04: Terminal I12.
- Analog output QA01.
- Output ...QV of a different function block.

Output

The output DC..QV supplies the current manipulated variable as an integer.

You can assign the following operands to output ...QV:

- Analog output QA01.
- Markers MD, MW, MB.

Function of the function block

The controller operates on the basis of the equation of the PID algorithm. According to this, the manipulated variable Y(t) is the result of the calculation of the proportional component, an integral component and a differential component.

→ The PID controller must be enabled so that it can work, i.e. the coil DC..EN must be active. If coil DC…EN is not active, the entire PID controller is deactivated and reset. The manipulated variable is set to zero.
The coils for calculating the P, I and/or 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.

PID controller equation:

$$Y(t) = Y_P(t) + Y_I(t) + Y_D(t)$$

$Y(t)$ = 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

Proportional component

The proportional component Y_P is the product of the gain (K_p) and the control difference (e). The control difference is the difference between the setpoint (X_S) and the actual value (X_i) at a specified scan time. The equation used by the device for the proportional component is as follows:

$$Y_P(t) = K_p \times [X_S(t) - X_i(t)]$$

K_p = Proportional gain
$X_S(t)$ = Setpoint with scan time t
$X_i(t)$ = Actual value with scan time t

Integral component

The integral component Y_I is proportional to the sum of the control difference over time. The equation used by the device for the integral component is as follows:

$$Y_I(t) = K_p \times T_c / T_n \times [X_s(t) - X_i(t)] + Y_I\ (t-1)$$

K_p = Proportional gain
T_c = Scan time
T_n = Reset time (also known as integration time)
$X_s(t)$ = Setpoint with scan time t
$X_i(t)$ = Actual value with scan time t
$Y_I(t-1)$ = Value of the integral component with scan time t - 1

Differential component

The differential component Y_D is proportional to the change in the control difference. So as to avoid step changes or jumps in the manipulated variable caused by the differential behaviour when the setpoint is changed, the change of the actual value (the process variable) is calculated and not the change in the control difference. This is shown in the following equation:

$$Y_D(t) = K_p \times T_v / T_c \times (X_i\ (t-1) - X_i(t))$$

K_p = Proportional gain
T_c = Scan time
T_v = Rate time of the control system (also called the differential time)
$X_i(t)$ = Actual value with scan time t
$X_i(t-1)$ = Actual value with scan time t - 1

Memory	96 + per 4 Byte/Constant at function block input	
Cycle time [µs]	easy800 standard	easy800-SWD
P	84	30
I	96	30
PI	99	31
PD	96	31
PID	111	34

Manual mode of the PID controller

A value must be present at the MV input in order to set the manipulated variable directly. If the coil DC…SE is activated, the value at MV is transferred directly as manipulated variable QV. This value is present for as long as the DC…SE coil is activated or the value at the MV input is changed. If the coil DC..SE is no longer triggered, the control algorithm is active again.

→ 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 as the value zero.

5.14 FT, signal smoothing filter

This function block is available for easy800-SWD from device version 01 and for easy800 standard devices from device version 04.

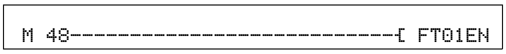
The function block smoothes noisy signals such as analog input signals. It operates as a low pass filter. 32 signal smoothing filter function blocks are provided.

→ If you are using the function block in time consuming applications such as in conjunction with a DC PID controller, you should not program any other time consuming applications on this device, which may excessively lengthen the cycle time.
A trouble-free function block operation is only ensured if the ratio of cycle time to scan time stated below is observed.
Divide the applications over two or several devices that you interconnect via easyNet.

Contacts and Coils

Coils	Function
FT..EN	EN=1 -> Activation of the function block function; EN=0 -> Reset of the function block, the output QV is set to 0.

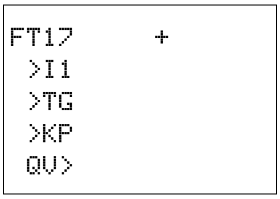
Example of a smoothing filter



```
M 48---------------------------[ FT01EN
```

Figure 132:Wiring the function block coil

Parametric programming of the function block inputs/outputs



```
FT17     +
 >I1
 >TG
 >KP
 QV>
```

When using the function block in the circuit diagram for the first time, use OK to automatically enter the general display of function block parameters, as shown in the figure on the left. Enter the function block settings here.

The display contains the following elements:

FT17	Function block: Signal smoothing, number 17
+	Parameter set can be called via the PARAMETERS menu
>I1	Input value: -32768 - +32767
>TG	Recovery time: 0 - 65535 resolved in 100 ms Example: The value 250 is processed in the function block as 25 s.
>KP	Proportional gain: 0 - 65535 in %; Example: With KP=1500 the function block performs the calculation with K_p =15
QV>	Output value: -32768 - +32767, smoothed

Inputs

I1: Assign the unfiltered input value to function block input I1.

TG: TG is used to define the recovery time.

KP: The proportional gain at input KP is used to adjust the input value.

You can assign the following operands to function block inputs I1, TG and KP:

- Constant.
- Markers MD, MW, MB.
- Analog inputs IA01 to IA04:
  - IA01: Terminal I7.
  - IA02: Terminal I8.
  - IA03: Terminal I11.
  - IA04: Terminal I12.

- Analog output QA01.
- Output ...QV of a different function block.

Output

Output FT..QV: Outputs the filtered value as an integer. You can assign the following operands to output ...QV>:

- Analog output QA01.
- Markers MD, MW, MB.

Function of the function block

The signal to be smoothed is added via input I1. The delayed and smoothed output value is transferred to QV.

The "recovery time" T_g is used for setting the time in which the smoothing is to take place. The recovery time should not be greater than necessary otherwise the signals will be delayed more than is required by the smoothing (unavoidable side effect of signal smoothing).

The (proportional gain) input K_P is used to define a proportional gain factor. The input signal is multiplied with this factor. The value 100 corresponds to a K_P (factor) of 1.

→ The signal filter must be enabled so that it can work, i.e. the coil FT..EN must be 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 initialized with the input value when the device is started or after a reset. This speeds up the startup behavior of the function block.

Memory	56 + per 4 Byte/Constant at function block input	
Cycle time [µs]	easy800 standard	easy800-SWD
	60	15

Response of the function block

The step response of the FT function is an e function. After a time t = T_g the normalized output value is 0.63 QV/QV_{max}.

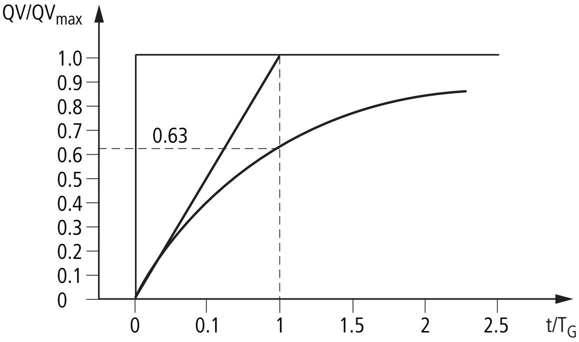


Figure 133:Figure: Step response of the FT function block

→ The function block updates the output value every time recovery time T_g expires.

The function block operates according to the following equation:

$Y(t) = [T_a/T_g] \times [K_p \times (x(t) - Y(t-1))]$

Y(t)	= Calculated output value for scan time t
T_a	= Scan time
T_g	= Recovery time
K_p	= Proportional gain
x(t)	= Output value with scan time t
Y(t–-1)	= Output value with scan time t - 1

Scan time:
Scan time T_a depends on the set recovery time value.

Recovery time T_g	Scan time T_a
0.1 s - 1 s	10 ms
1 s - 6553 s	0.01 T_g

Cycle time to scan time: The scan time should be very large, approx. factor 10 compared to the cycle time, in order to ensure the correct ratio of cycle time t_{cyc} to scan time T_a. The scan time T_a is set indirectly via the value for the recovery time T_g (see above table).

The following applies: $t_{cyc} << T_a$

Some applications exist in which this condition cannot be reached. In this case you should set the cycle time using the ST function block (set cycle time) so that the scan time is an integer multiple of the cycle time.

t_{cyc} n = T_a with n = 1, 2, 3, …

The function block always operates with a scan time that corresponds to an integer multiple of the cycle time. This can cause the set recovery time to be lengthened.

5.15 GT, get value from the NET

With function block GT (from GET) you can poll a 32-bit value from the easyNet network . The easyNet network is referred to as NET below. GET function block GT automatically fetches the data designated for itself as soon as it is provided on the NET by another NET station using PUT function block PT.

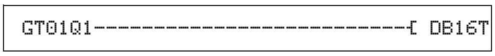
easy800 device allows the use of up to 32 GT function blocks. EASY802-DC-SWD is not provided for the function block.

Contacts and Coils

A closed contact indicates for the duration of a processing cycle that new data was received via the NET and is present at the function block output QV:

Contacts	Meaning
GT..Q1	Transferred value present at QV> (for one processing cycle)

Example of a GT function block

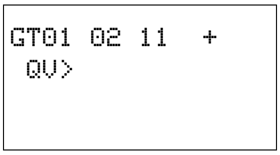


```
GT01Q1--------------------------[ DB16T
```

Figure 134:Wiring of the function block contact

The contact triggers a data function block.

Parametric programming of the function block inputs/outputs



```
GT01 02 11  +
 QV>
```

When using the function block in the circuit diagram for the first time, use OK to automatically enter the general display of function block parameters, as shown in the figure on the left. Enter the function block settings here. The display contains the following elements:

GT01	Function block: GT, number 01
02	The number (NET-ID) of the transmitting station, possible NET IDs: 01 - 08
11	Send function block of the sending station (here PT 11), possible function block numbers: 01 - 32
+	Parameter set can be called via the PARAMETERS menu
QV>	32 Bit value received from the NET network, Value range: -2147483648 - 2147483647

Number of the sending station

The NET can connect up to 8 stations that can exchange data with each other. The stations are identified by their station number. Further information on the easyNET network is provided in → Chapter 6, “The network easyNet“ from Page 273.

Number of the send function block

The sending station has 32 function blocks for transmitting 32-bit values to the network.

Output

The output provides the value transferred via the NET for one cycle. You can assign the following operands to output ...QV:

- Analog output QA01.
- Markers MD, MW, MB.

GET diagnostics

The GT function block only functions if the NET is operating correctly (→ section 9.6, “easy800 standard expanding”, page 325).

Function of the function block

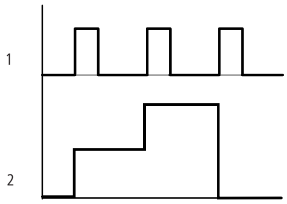


Figure 135:Signal diagram function block GT
1: GT..Q1
2: Value on GT..QV

→ The GT function block has the value 0 when the power supply is switched on.

Memory	28 Byte	
Cycle time [µs]	easy800 standard	easy800-SWD
	21	8

5.16 HW, weekly timer

An easy800 device is provided with a real-time clock, which you can use in the circuit diagram as a weekly and year time switch.

→ The procedure for setting the time,
→ Section “ Setting time and date“, Page 313.

easy800 device allows for the use of up to 32 function blocks of type weekly timer.

For each weekly timer four switch channels A, B, C and D are available. You can choose an on and off switching time for every channel. These channels of the weekly timer all act jointly on the contact Q1 that you include in the circuit diagram. The channels are set via the parameter display or easySoft.

⚠ **DANGER**

Requirement:

- The function block must have been entered directly via the operating buttons of the easy800 device.
- At least one channel was not assigned parameters.
- The on time is to be set from 23:59 on Saturday to Sunday after 00:00.

Behavior:

- The time switch switches off on Sunday at 00:00.
- This cannot be the required off time!

Remedy:

- Assign all time switch channels with the required closing delay and break times.
- To create the program you must use the easySoft or easySoft-Pro programming software.

In this case, the behavior does not occur.

Behaviour in the event of a power failure

The time and date are backed up in the event of a power supply failure and continue to run. This means that it will continue to run in the event of a power failure, although the time switch relays will not switch. The contacts are kept open when de-energized. Information on the battery back-up time are provided on → section 12.3.4, “Central processing unit, real-time clock/ timing relay/memory”, page 348.

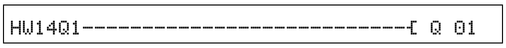
→ The electronic clock module integrated into the easy800 device works with a date range of 2001-01-01 to 2099-12-31.

Contacts and Coils

The contact of the function block is evaluated in the circuit diagram. It closes for the on time and opens for the off time on the defined days.

Contacts	Meaning
HW..Q1	Switching contact; closes in the defined period

Example for a weekly timer:



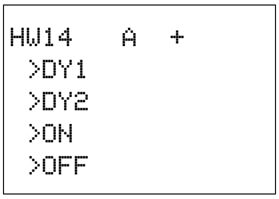
```
HW14Q1----------------------------[ Q 01
```

Figure 136:Wiring of the function block contact

The weekly timer controls a device output directly.

Parametric programming of the function block inputs/outputs

When using the function block in the circuit diagram for the first time, use OK to automatically enter the general display of function block parameters, as shown in the figure on the left. The function block settings are carried out here. The display contains the following elements:



```
HW14   A  +
 >DY1
 >DY2
 >ON
 >OFF
```

HW14	Function block: Weekly timer, number 14
A	Time switch channel: A
+	Parameter set can be called via the PARAMETERS menu
>DY1	Day 1: Day on which the switch-on is to occur for the first time (start of time period).
>DY2	Day 2: Day on which the switch-off is to occur for the last time (end of time period).
>ON	Closing delay: 00:00 - 23:59, No on switching: --:--
>OFF	Break time: 00:00-23:59, No off switching: --:--

Channel

For each weekly timer channels A, B, C and D are available. These channels all act jointly on the contact Q1 of the 7-day time switch.

Parameters	Function
A	Display of the parameters: Channel A
B	Display of the parameters: Channel B
C	Display of the parameters: Channel C
D	Display of the parameters: Channel D

Day 1 and day 2

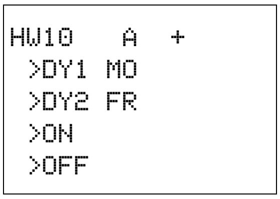
These parameters DY1/DY2 are used to limit the period in which the output HW..Q1 is switched.

Enter either a period of several days or only of one day:

Entry	On and off time
MO	Monday
TU	Tuesday
WE	Wednesday
DO	Thursday
FR	Friday
SA	Saturday
SO	Sunday
--	No day

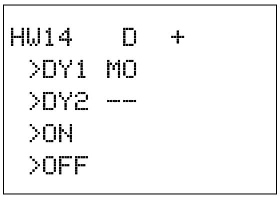
Examples:
Timing relay = function relay with contacts and coils
Time switch = function relay with contacts

- Monday to Friday



```
HW10   A  +
 >DY1 MO
 >DY2 FR
 >ON
 >OFF
```

- Only on Monday



```
HW14   D  +
 >DY1 MO
 >DY2 --
 >ON
 >OFF
```

On and off time

The ON/OFF parameters are used for setting the on and off time.

Operation of the function block

The switching points are defined according to the parameters entered.

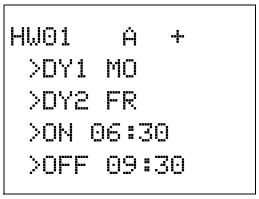
MO to FR: on weekdays Mo, Tu, We, Th, Fr
ON 10:00, OFF 18:00: On and off time on each weekday.

MO: Every Monday
ON 10:00: On time
SA: Every Saturday
OFF 18:00: Off time

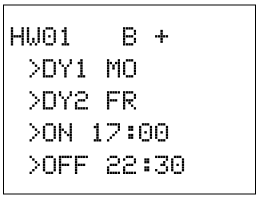
Switching on working days

The weekly timer HW01 switches on from Mondays to Fridays with channel:

- A between 6:30 and 9:30
- B between 17:00 and 22:30



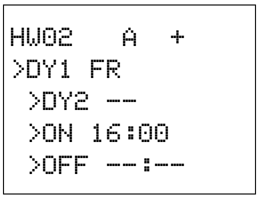
```
HW01   A  +
 >DY1 MO
 >DY2 FR
 >ON 06:30
 >OFF 09:30
```



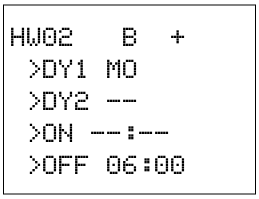
```
HW01   B +
 >DY1 MO
 >DY2 FR
 >ON 17:00
 >OFF 22:30
```

Switching at the weekend

The weekly timer HW02 switches on at 16:00 on Friday and switches off at 6:00 on Monday.



```
HW02    A   +
>DY1 FR
 >DY2 --
 >ON 16:00
 >OFF --:--
```



```
HW02    B   +
 >DY1 MO
 >DY2 --
 >ON --:--
 >OFF 06:00
```

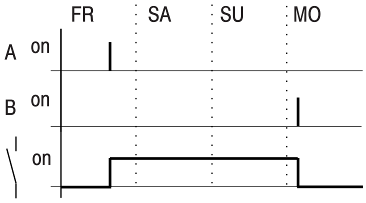
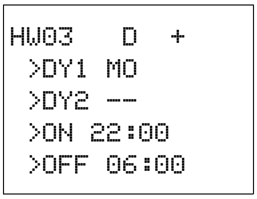


Figure 137:Signal diagram of “weekend”

Overnight switching

The weekly timer HW03 switches on overnight at 22:00 Monday and switches off at 6:00 on Tuesday.



```
HW03    D   +
 >DY1 MO
 >DY2 --
 >ON 22:00
 >OFF 06:00
```

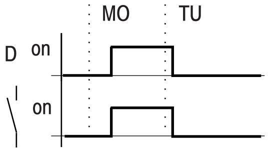
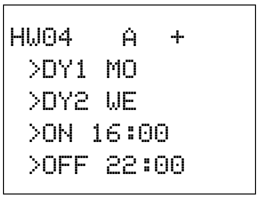


Figure 138:Signal diagram for night switching

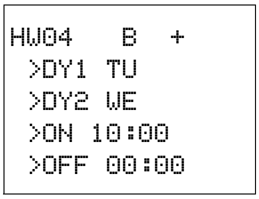
If the break time is before the closing delay, easy800 device will switch off on the following day.

Time overlaps

The 7-day time switch 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. The off time for Mondays to Wednesdays is 22:00.



```
HW04    A   +
 >DY1 MO
 >DY2 WE
 >ON 16:00
 >OFF 22:00
```



```
HW04    B   +
 >DY1 TU
 >DY2 WE
 >ON 10:00
 >OFF 00:00
```

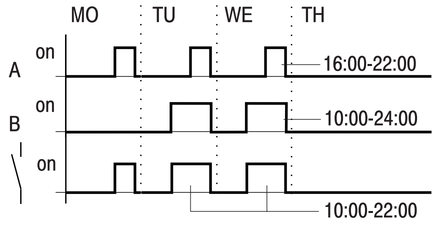


Figure 139:Signal diagram of overlaps

→ On and off times of output Q1 always follow the channel which switches first.
In other words, the first On time at one of the four channels switches the state at output Q1 to 1.
The first Off time of a channel switches the state at output Q1 to 0.
If the on time and off time are the same, the output Q1 is switched off.

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 off time was at 16:00.

```
HW05   A  +
 >DY1 MO
 >DY2 SU
 >OFF 16:00
```

```
HW05   B  +
 >DY1 MO
 >DY2 SU
 >ON 12:00
 >OFF 18:00
```

→ When it is switched on, easy800-device 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 at 00:00 on Mondays and switch off at 00:00 on Tuesdays.

```
HW20   A  +
 >DY1 MO
 >DY2
 >ON 00:00
 >OFF
```

```
HW20   B  +
 >DY1 TU
 >DY2
 >ON
 >OFF 00:00
```

Memory	68 + per 4 Byte/Constant at function block input	
Cycle time [µs]	easy800 standard	easy800-SWD
	45	14 - 93[1]

[1] depending on the parameter setting

5.17 HY - Year Time Switch

An easy800 device is fitted with a real-time clock that you can use in the circuit diagram as a year and weekly timer.

→ The procedure for setting the time
→ section , "Setting time and date", page 313.

Year time switch HY1…HY32 (Hora Year)							
Boolean inputs/outputs				Parameters			
Contact		Coil		Input value		Output value	
Q1	"1" if the on condition is fulfilled ①						
					On/off date ② + ③		

Read the description of the inputs/outputs in the following tables

If you have to implement special on and off switching functions on public holidays, vacations, company holidays, school holidays and special events, these can be implemented easily with the year time switch.

easy800 device allows for the use of up to 32 function blocks of type year time switch.

For each year time switch four channels A, B, C and D are available.
You can choose an on and off switching time for every channel. These channels of a year time switch all act jointly on the contact Q1 that you include in the circuit diagram. The channels are set via the parameter display or easySoft.

The year time switch can:

- Switch at recurring intervals by switching on and off for individual days, months, or years.
- Switch for continuous periods of time by remaining continuously switched on from the start of any user-defined day until the end of any user-defined day, month, or year.

→ The parameters for the switch-on and switch-off times for recurring intervals are configured in one single channel for each. The parameters for the switch-on and switch-off times for a continuous period of time are configured in two neighbouring channels. If you enter the ON information on channel A, the OFF information must be entered on channel B; likewise, if the ON information is on channel B, the OFF information must be on channel C.

Behaviour in the event of a power failure

The time and date are backed up in the event of a power supply failure and continue to run. This means that it will continue to run in the event of a power failure, although the time switch relays will not switch.

The contacts are kept open when de-energized. Information on the battery back-up time are provided on → section 12.3.4, “Central processing unit, real-time clock/timing relay/memory”, page 348.

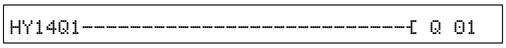
→ The electronic clock module integrated into the easy800 device works with a date range of 2001-01-01 to 2099-12-31

Contacts and Coils

The contact closes each day for the on time and opens for the off time.

Contacts	Meaning
HY..Q1	Switching contact; closes in the defined period

Example for a year time switch:

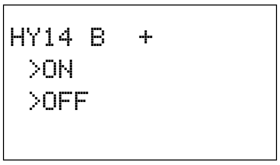


```
HY14Q1----------------------------[ Q 01
```

Figure 140:Wiring of the function block contact

The year time switch controls a device output directly.

Parametric programming of the function block inputs/outputs



```
HY14 B  +
 >ON
 >OFF
```

When using the function block in the circuit diagram for the first time, use OK to automatically enter the general display of function block parameters, as shown in the figure on the left. Enter the function block settings here. The display contains the following elements:

HY14	Function block: Year time switch, number 14
B	Time switch channel: B
+	Parameter set can be called via the PARAMETERS menu
>ON	On date
>OFF	Off date

Channel

For each year time switch function block, you can configure the parameters for the A, B, C, and D channels that act jointly on contact Q1 of this function block.

Parameters	Function
A	Display of the parameters: Channel A
B	Display of the parameters: Channel B
C	Display of the parameters: Channel C
D	Display of the parameters: Channel D

On and off date

Use these parameters to set the data for on and off switching in the format DD.MM.YY.

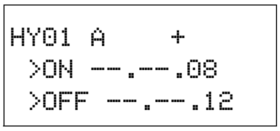
Example

Month/Day/Year: 11/11/07

Parameters	Function
>ON	On date
>OFF	Off date

Function of the function block

The year time switch can switch ranges, individual days, months, years or combinations of all three.

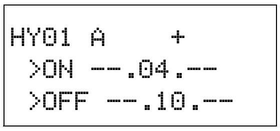


```
HY01 A    +
 >ON --.--.08
 >OFF --.--.12
```

Years

Switching at 00:00:

- On on 01.01.2008 and
- Off on 01.01.2013.

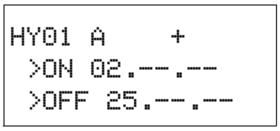


```
HY01 A    +
 >ON --.04.--
 >OFF --.10.--
```

Months

Switching at 00:00:

- On on 01 April every year.
- Off on 01 November every year.



```
HY01 A    +
 >ON 02.--.--
 >OFF 25.--.--
```

Days

Switching at 00:00:

- On on the 2nd day of every month.
- On on the 26th day of every month.

Entry rules

The contact switches (ON to OFF)

- in the years.
- in the defined months.
- on the entered days.

CAUTION

The year time switch only operates correctly if you observe the following rules:

- The On year must be before the Off year,
- ON and OFF times must be specified the same in pairs, for example:
  - ON = Day/–/–, OFF = Day/--/–
  - ON = –/–/Year, OFF = –/--/Year,
  - ON = –/Month/Year, OFF = –/Month/Year
  - ON = Day/Month/Year, OFF = Day/Month/Year
- Normally, the ON day should be before the OFF day; otherwise, the exception described on Example 8 (→ section , “Example 8: Selecting a time range of two days spanning the turn of the year (2-channel)”, page 209) on applies.

The following nine entry rules are possible (XX = position used)

```
HY01 A    +
 >ON XX.--.--
 >OFF XX.--.--
```

Rule 1

ON: Day

OFF: Day

```
HY01 A    +
 >ON --.XX.--
 >OFF --.XX.--
```

Rule 2

ON: Month

OFF: Month

```
HY01 A    +
 >ON --.--.XX
>ON --.--.XX
```

Rule 3

ON: Year

OFF: Year

```
HY01 A    +
 >ON XX.XX.--
 >OFF XX.XX.--
```

Rule 4

ON: Day/month

OFF: Day/month

```
HY01 A    +
 >ON --.XX.XX
 >OFF --.XX.XX
```

Rule 5

ON: Month/year

OFF: Month/year

```
HY01 A    +
 >ON XX.XX.XX
 >OFF XX.XX.XX
```

Rule 6

ON: Day/month/year

OFF: Day/month/year

```
HY01 A    +
 >ON XX.XX.--
 >OFF --.--.--
```

Rule 7

Two-channel

Channel A ON: Day/month

```
HY01 B    +
 >ON --.--.--
 >OFF XX.XX.--
```

Channel B OFF: Day/month

```
HY01 C +
 >ON XX.XX.XX
 >OFF --.--.XX
```

Rule 8

Two-channel

Channel C ON: Day/Month/Year

```
HY01 D +
 >ON --.--.XX
>ON --.--.XX
```

Channel D OFF: Day/month/year

With this rule, the same year number must be entered in each channel in the ON and OFF entry area.

Rule 9

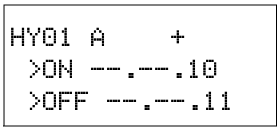
Overlapping channels: The first ON date switches on and the first OFF date switches off.

Memory	68 + per 4 Byte/Constant at function block input	
Cycle time [µs]	easy800 standard	easy800-SWD
	61	14 - 129[1]

[1] depending on the parameter setting

Sample parameter configurations

Configuring the parameters for recurring intervals



```
HY01 A    +
 >ON --.--.10
 >OFF --.--.11
```

Example 1: Selecting year range (1-channel)

The year time switch HY1 is required to switch on at 00:00 on January 1 2010 and stay on until 00:00 January 1 2012.

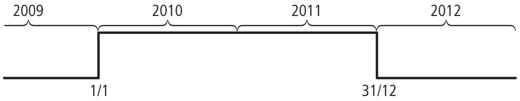
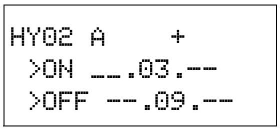


Figure 141:Select year range



```
HY02 A    +
 >ON __.03.--
 >OFF --.09.--
```

Example 2: Selecting month ranges (1-channel)

The year time switch HY02 is required to switch on at 00:00 on March 1 and stay on until 00:00 October 1.

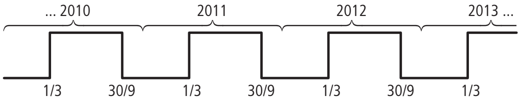
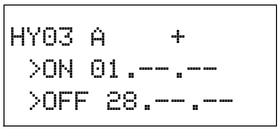


Figure 142:Select month range



```
HY03 A    +
 >ON 01.--.--
 >OFF 28.--.--
```

Example 3:Selecting day ranges (1-channel)

The year time switch HY03 is required to switch on at 00:00 on day 1 of each month and switch off at 00:00 on day 29 of each month.

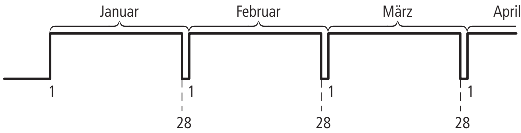
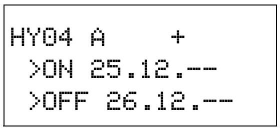


Figure 143:Select day range



```
HY04 A    +
 >ON 25.12.--
 >OFF 26.12.--
```

Example 4: Selecting public holidays (1-channel)

The year time switch HY04 is required to switch on at 00:00 on day 25.12 of each year and switch off at 00:00 on day 27.12 of each year, “Christmas program”.

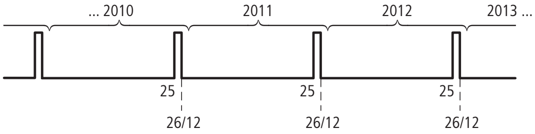
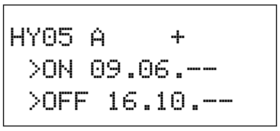


Figure 144:Select public holiday



```
HY05 A    +
 >ON 09.06.--
 >OFF 16.10.--
```

Example 5: Specific days of specific months (1-channel)

The year time switch HY05 is required to switch on at 0:00 on day 9 of months 6, 7, 8, 9 and 10 of each year and switch off at 00:00 on day 17 of the month.

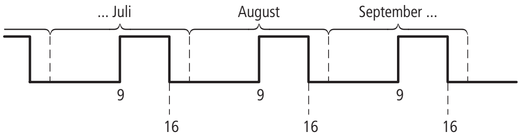


Figure 145:Chose specific days of specific months

Configuring the parameters for continuous periods of time

The following examples show how to configure the parameters for continuous periods of time in which the time switch remains continuously switched on. The parameter definition is carried out in pairs on two adjacent channels.

Example 6: Time range for each year (2-channel)

The year time switch HY01 is required to switch on at 00:00 on day 02.05 of each year and stay on continuously until 00:00 on 1.11 of each year, “Open air season”.

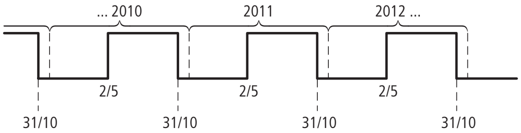
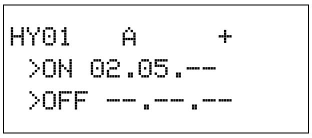


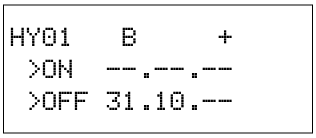
Figure 146:Select time range for each year

Parameter settings of the year time switch HY01:



```
HY01   A     +
 >ON 02.05.--
 >OFF --.--.--
```

ON/OFF: First/last switch-on years are not restricted in this case.



```
HY01   B     +
 >ON  --.--.--
 >OFF 31.10.--
```

ON/OFF: First/last switch-off years are not restricted in this case.

Example 7: Selecting a time range for a limited number of years (2-channel)

The HY01 year time switch should switch on at 00:00, on 05/02, in the years 2011 and 2012, and remain switched on until 00:00 on 11/1 in each case.

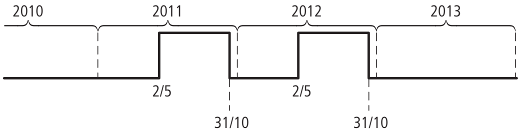


Figure 147:Select a time range for a limited number of years

Parameter settings of the year time switch HY01:

```
HY01   A     +
 >ON  05/02/11
 >OFF --.--.12
```

ON/OFF: First/last switch-on year.

```
HY01   B     +
 >ON  --.--.11
 >OFF 10/31/12
```

ON/OFF: First/last switch-off year.

Example 8: Selecting a time range of two days spanning the turn of the year (2-channel)

The HY01 year time switch should switch on at 00:00 on 12/31 of every year and remain continuously switched on until 00:00 on 02/01 of every year.

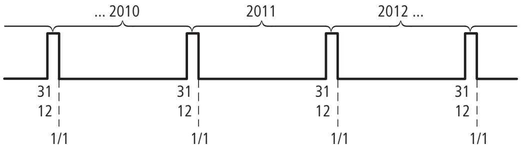


Figure 148:Select a time range of two days spanning the turn of the year

Parameter settings of the year time switch HY01:

```
HY01   A     +
 >ON  31.12.--
 >OFF --.--.--
```

ON/OFF: First/last switch-on years are not restricted in this case.

```
HY01   B     +
 >ON  --.--.--
 >OFF 01.01.--
```

ON/OFF: First/last switch-off years are not restricted in this case.

Example 9: Selecting a time range of two days spanning the turn of the year on 2010/2011, 2011/2012 (2-channel)

The HY01 year time switch should switch on at 00:00 on 12/31 of 2010 and remain switched on until 00:00 on 01/02 of 2011, and switch on at 00:00 on 12/31 of 2011 and remain switched on until 00:00 on 01/02 of 2012.

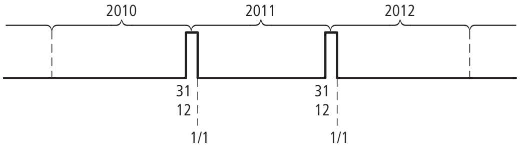
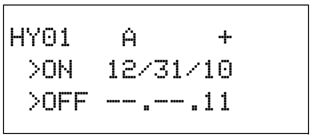


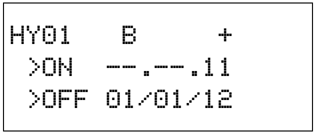
Figure 149:Select a time range of two days spanning the turn of the year

Parameter settings of the year time switch HY01:



```
HY01   A     +
 >ON  12/31/10
 >OFF --.--.11
```

ON/OFF: First/last switch-on year.



```
HY01   B     +
 >ON  --.--.11
 >OFF 01/01/12
```

ON/OFF: First/last switch-off year.

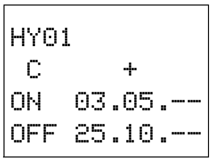
Example 10: Overlapping time ranges (2-channel)

This example shows how to configure the parameters for continuous overlapping time ranges on adjacent channels.

The HY01 year time switch, channel C, switches on at 00:00 on the third of months 5, 6, 7, 8, 9, 10 and remains switched on until 00:00 on the 26th of these months.

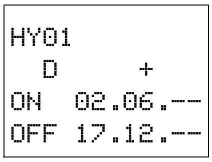
The HY01 year time switch, channel D, switches on at 00:00 on the second of months 6, 7, 8, 9, 10, 11, 12 and remains switched on until 00:00 on the 18th of these months.

Parameter settings of the year time switch Y01



```
HY01
 C     +
ON  03.05.--
OFF 25.10.--
```

①



```
HY01
  D     +
ON  02.06.--
OFF 17.12.--
```

②

Total result corresponding to the channel configurations and behavior of contact Q1 in this example:

The time switch switches on at 00:00 on 3 May and stays on until 00:00 on 26 May. In June, July, August, September, October, the time switch will switch on at 00:00 on day 2 of the month and switch off at 00:00 on day 18. In November and December, the time switch will switch on at 00:00 on the second of the month and switch off until 00:00 on the 18th. The following note explains this switching behavior.

→ Switching behaviour with overlapping channel settings:
If parameterized ranges overlap, the year time switch switches the contact on with the first detected ON, irrespective of which channel supplies this ON signal. In the same way, the year time switch switches the contact off with the first detected OFF, irrespective of whether another channel still supplies the ON signal!

5.18 JC, conditional jump

With function block JC.. you can branch off forwards to a jump label (Label, LB) within the function block list to skip several function blocks.

You use LB.. (jump label) as the jump destination. The JC and LB function blocks are used in the circuit diagram.

easy800 device allows up to 32 conditional jumps.

For each JC.. function block, an LB.. function block must be provided as the jump label. The JC.. and LB.. function blocks must always be used in pairs. JC01 is always assigned the jump label LB01.

The description of the LB.. function block is provided on → section 5.19, “LB, jump label”, page 213.

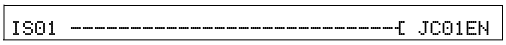
Contacts and Coils

The contact indicates a fault if there is not an associated jump present or is in front of the jump location (backward jump).

Contacts	Meaning
JC..E1	Error message

Coils	Function
JC..EN	Activation of the function block

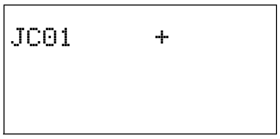
Example of a conditional jump



```
IS01 ---------------------------[ JC01EN
```

Figure 150:Wiring the enable coil

Parametric programming of the function block inputs/outputs



```
JC01      +
```

When using the function block in the circuit diagram for the first time, use OK to automatically enter the general display of function block parameters, as shown in the figure on the left. Enter the function block settings here. The display contains the following elements:

JC01	Function block: Conditional jump, number 01
+	Without function

Function of the function block

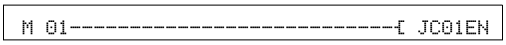
If the JC..EN enable coil is active, the program jumps forwards over one or several function blocks. The next function block to be processed is then the first to follow the LB jump label in the function block diagram. If the JC..EN enable coil is not active, the program is processing the next function block that is added in the function block diagram behind JC…

In the function block list a jump label must be positioned downstream, i.e. towards the end of the list when viewed from associated function block "conditional jump". If it is placed in front of the associated jump location (backward jump), the program branches to the end of the function block list and fault contact JC..E1 of function block "conditional jump" closes. If no associated jump label is set in the function block list, the program jumps to the end of the list and fault contact JC..E1 also closes.

Memory	28 Byte	
Cycle time [µs]	easy800 standard	easy800-SWD
	20	8

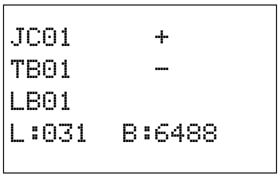
Example

A jump is to be made downstream of a table function block when marker M01 has the state 1.



```
M 01---------------------------[ JC01EN
```

Figure 151:Example of conditional jump



```
JC01     +
TB01     -
LB01
L:031  B:6488
```

Enter the function block editor via <PROGRAM…, PROGRAM -> FUNCTION RELAYS>.

- ▶ In this menu use the cursor buttons ⌃⌄ to move to the location from where the jump is to be made and press OK.
- ▶ Then use the cursor keys ⌃⌄ to select function block JC01 and close the input dialog with two clicks on OK.
- ▶ Insert table function block TB01 into then next position in the function block list, configure it and complete your input with ESC.
- ▶ Use the cursor keys ⌃⌄ to navigate to the next position in the function block list and insert the jump label LB01 described below there.

During processing, the program branches to jump label LB01 for as long as JC01EN is 1.

→ For subsequently inserting function block JC.. in the function block list to skip other, existing function blocks, the use of the easySoft-Pro programming software is recommended.
This software allows function blocks to be moved into any position.

→ For subsequently inserting function block JC.. in the function block list to skip other, existing function blocks, the use of the easySoft programming software is recommended .
This software allows function blocks to be moved into any position.

5.19 LB, jump label

Within the function block list jump label LB.. is the jump target for a conditional jump with function block JC.. . The function blocks JC and LB are always used in pairs in the circuit diagram.

easy800 device allows the use of up to 32 jump labels.

Function Block

Because function block LB.. acts only as a jump target, it does not have to be linked in the circuit diagram and does not have to be parameterized. It has neither coils nor contacts and is simply placed at the required position in the function block list.

→ A corresponding JC.. function block (conditional jump) must exist for every LB.. function block. Conditional jump JC01, for example, is always associated with jump label LB01.

In the function block list a jump label must be positioned downstream, i.e. towards the end of the list when viewed from associated function block “conditional jump”. If it is placed in front of the associated jump location (backward jump), the program branches to the end of the function block list and fault contact JC..E1 of function block “conditional jump” closes. If no associated jump label is set in the function block list, the program jumps to the end of the list and fault contact JC..E1 also closes.

Enter the function block editor from the main menu via <PROGRAM… -> PROGRAM -> FUNCTION RELAYS>.

▶ In this menu use the cursor buttons ⌃⌄ to move to the location to which the jump is to be made and press OK.
▶ Then use the cursor keys ⌃⌄ to select function block LB.. and close the input dialog with two clicks on OK.

The description of the JC.. conditional jump function block is provided in → Chapter 5, “Function blocks“ from Page 211. This also contains an example.

5.20 LS, value scaling

This function block is available for easy800-SWD from device version 01 and for easy800 standard devices from device version 04.

The function block enables you to convert values from one value range to another one.

The function block scales values at input LS..I1 according to the mathematical relationship you have set, and then outputs the result as a larger or smaller value at the output at LS..QV. The mathematical relationship is based on a straight line defined by the coordinate pairs X1, Y1 and X2, Y2 (see under "The mathematical relationship is:").

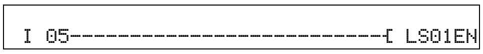
A typical application is the conversion of values, such as 0…20 mA to 4…20 mA. A easy800 device is provided with 32 value scaling function blocks.

Contacts and Coils

The enable coil LS..EN activates the function block in the standard circuit diagram; when the coil is deactivated, the output QV> is set to 0.

Coils	Function
LS..EN	Enables the function block

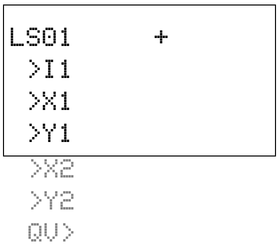
Example of a value scaling function block



```
I 05--------------------------[ LS01EN
```

Figure 152:easy800 circuit diagram with enabling of value scaling function block

Parametric programming of the function block inputs/outputs



```
LS01      +
 >I1
 >X1
 >Y1
 >X2
 >Y2
 QV>
```

When using the function block in the circuit diagram for the first time, use OK to automatically enter the general display of function block parameters, as shown in the figure on the left. Enter the function block settings here. The display contains the following elements:

LS01	Function block: Value scaling, number 01 Value range: -2147483648 - 2147483647, valid for all values.
+	Parameter set can be called via the PARAMETERS menu
>I1	Input value
>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

I1: Assign the unscaled value to function block input I1.

X1 and Y1: You define the first support point of the straight line at function block inputs X1 and Y1.

X2 and Y21: The second pair of values at function block inputs X2 and Y2 determines the second support point of the straight line.

You can assign the following operands to function block inputs I1, X1, Y1, X2 and Y2:

- Constant
- Markers MD, MW, MB
- Analog inputs IA01 to IA04:
  - IA01: Terminal I7
  - IA02: Terminal I8
  - IA03: Terminal I11
  - IA04: Terminal I12
- Analog output QA01
- Output ...QV> of a different function block

Output

The output QV supplies the scaled value as an integer.

You can assign the following operands:

- Markers MD, MW, MB
- Analog output QA01

Function of the function block

→ The scaling function block must be enabled via coil LS..EN = 1 in order for it to work. If the coil LS..EN is not active, the entire function block is deactivated and reset. The output value is set to zero.

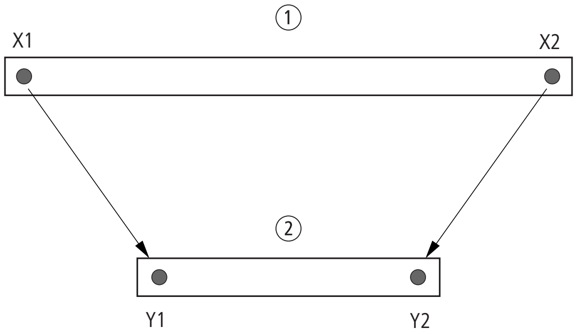


Figure 153:Value scaling function block – reduce value range

① Source range

② Destination range

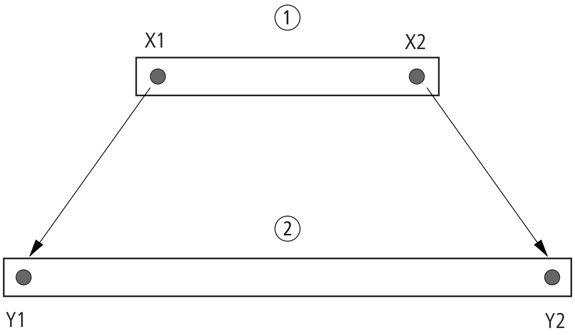


Figure 154:Function block value scaling - increasing value range

① Source range

② Destination range

The mathematical relationship is follows:

- Y = m x X + Y0
- m = (Y2 - Y1) / (X2 - X1)
- Y0 = ((X2 x Y1) - (X1 x Y2)) / (X2 - X1)

m = gradient
Y0 = Y offset when X = 0
X1, Y1 = First value pair
X2, Y2 = Second value pair
g = Straight line with positive gradient

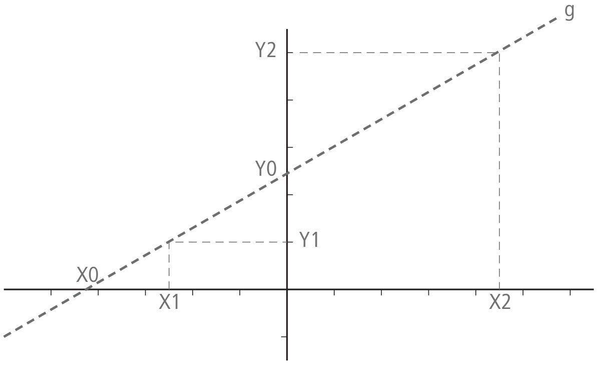


Figure 155:Mathematical interrelationship

Memory	64 + per 4 Byte/Constant at function block input	
Cycle time [µs]	easy800 standard	easy800-SWD
	224	98

Application example

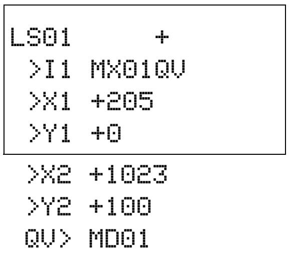
The signal of measuring value converters with a current output reaches the input LS..I1 of the scaling function block via the output of a multiplexer function block. The measuring value is between:

- 4 mA, corresponds to measured value 0 and
- 20 mA, corresponds to maximum measured value.

The task of the function block is to adjust the value in order to show it at output LS..QV as a value between 0 and 100 %.

The transferred measured value has a resolution between 0…1023 bits. This means:

- The minimum measured value is 1023 bits x 4 mA / 20 mA = 205 bit.
- The maximum measured value is at 1023 bits.



```
LS01     +
 >I1 MX01QV
 >X1 +205
 >Y1 +0
 >X2 +1023
 >Y2 +100
 QV> MD01
```

The parameter display and the parameter set for function block LS01 is shown on the left. After the function block number 01 is defined, you enter here the following values:

- the operand MX01QV with the value to be scaled.
- the first support point of the straight line with its zero point coordinates:
  - X1 = 205
  - Y1 = 0 and
- the second support point of the straight line with its maximum point coordinates:
  - X1 = 1023
  - Y1 = 100
- the transfer marker MD01 with the scaled value.

5.21 MR, master reset

The master reset function block allows you to reset the states of the markers and all outputs to 0 with a single command. The operating mode of the function block allows you to set whether only the outputs, only the markers or outputs and markers are to be reset.

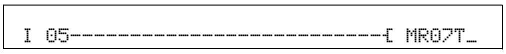
easy800 device allows the use of up to 32 master reset function blocks.

5.21.0.1 Contacts and Coils

Contacts	Meaning
MR..Q1	1 on activated trigger signal

Coils	Function
MR..T_	Function block reset on rising edge

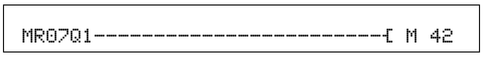
Example of a master reset function block



```
I 05---------------------------[ MR07T_
```

Figure 156:Wiring the trigger coil

The trigger coil is connected to a device input.

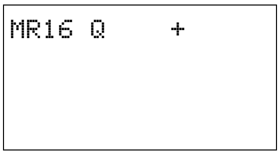


```
MR07Q1-------------------------[ M 42
```

Figure 157:Wiring the contacts

The message of the function block is sent to a marker.

Parametric programming of the function block inputs/outputs



```
MR16 Q     +
```

When using the function block in the circuit diagram for the first time, use OK to automatically enter the general display of function block parameters, as shown in the figure on the left. Enter the function block settings here. The display contains the following elements:

MR16	Function block: Master reset, number 16
Q	Operating mode: Reset outputs
+	Parameter set can be called via the PARAMETERS menu

Operating mode

The operating mode setting allows you to set whether markers, outputs or both are to be reset.

Parameters	Function
Q	Resets the outputs used in the circuit diagram Q.., *Q.., QA., S.., *S.., *SN..; * = NET-ID
M	Acts on the marker range MD01 to MD96.
ALL	Acts on the operands specified above under Q and M that are used in the circuit diagram.

The factory setting of this parameter is Q.

Function of the function block

All outputs or the markers are set to 0 in accordance with the operating mode when a rising edge is detected on the trigger coil.

→ Carry out the master reset function block so that all data ranges are cleared reliably. Otherwise subsequent function blocks may overwrite the data ranges.

The contact MR1..Q1 assumes the state of its own trigger coil and confirms the reset operation.

Memory	20 Byte	
Cycle time [µs]	easy800 standard	easy800-SWD
Q	47	24
M	185	116
ALL	217	134

5.22 MX, data multiplexer

This function block is available for easy800-SWD from device version 01 and for easy800 standard devices from device version 07.

The data multiplexer function block allows you to select a value from eight input values MX..I1 to MX..I8 as if using a mode selector switch.
The function block provides this value after transferring it to its output MX..QV.

A easy800 device allows the use of up to 32 data multiplexers.

The function block is enabled via the coil in the circuit diagram and its contact is then queried as an error output.

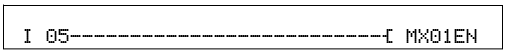
Contacts and Coils

Contacts	Meaning
MX..E1	Fault, e.g. invalid channel setting

When the enable coil MX..EN is set to 1, it switches the selected input Ix to output QV. This switch operation is retained for as long as MX..EN is 1. When the MX..EN state changes from 1 to 0, output QV is reset to 0.

Coils	Function
MX..EN	Enables the function block

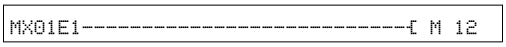
Example



```
 I 05--------------------------[ MX01EN
```

Figure 158:Wiring the enable coil

Its coil MX01EN is directly connected to the device terminal I05.

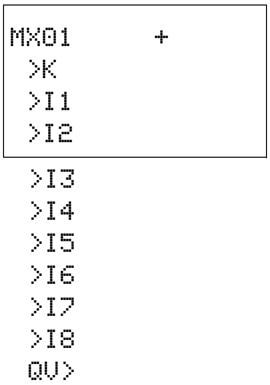


```
MX01E1---------------------------[ M 12
```

Figure 159:Wiring of the function block contact MX01E1

The function block contact MX01E1 (fault message) is associated with the marker M12.

Parametric programming of the function block inputs/outputs



```
MX01      +
 >K
 >I1
 >I2
 >I3
 >I4
 >I5
 >I6
 >I7
 >I8
 QV>
```

When using the function block in the circuit diagram for the first time, press OK to enter the general display of the function block parameters automatically, as shown in the figure on the left for an on-delayed timing relay. Enter the function block settings here. The display contains the following elements:

MX01	Function block; Data multiplexer, number 01
+	Parameter set can be called via the PARAMETERS menu
>K	Channel number (0 - 7)
>I1 >I8	Input channels I1 to I8, value range: - 2147483648 - 2147483647
QV>	Supplies the selected input value

Inputs

K: Function block input K is used to define the input channel to be supplied at function block output QV.

I1 to I8: Assign the values to function block inputs I1 to I8 that you wish to output on function block output QV.

You can assign the following operands to the function block inputs K and I1 to I8:

- Constant.
- Markers MD, MW, MB.
- Analog inputs IA01 to IA04:
  - IA01: Terminal I7.
  - IA02: Terminal I8.
  - IA03: Terminal I11.
  - IA04: Terminal I12.
- Analog output QA01.
- Output …QV of a different function block.

Output

The output QV supplies the value present at the selected input channel.

You can assign the following operands to output MX..QV:

- Analog output QA01.
- Markers MD, MW, MB.

Function of the function block

The multiplexer is active if the Enable coil MX..EN is set to 1.

You use the channel number input MX..K to set which input is to be switched through to the output. The channel number input switches the input values in their order, i.e.:

- Channel number 0 supplies the input MX..I1 to MX..QV.
- Channel number 7 supplies the input MX..I8 to MX..QV.

You can also change the channel number when the EN input is set.

When EN = 1, the value at the selected input Ix is continuously output and refreshed with every processing cycle.

The fault contact MX..E1 closes if an invalid parameter is present at the channel number input MX..K.

Memory	96 + per 4 Byte/Constant at function block input	
Cycle time [µs]	easy800 standard	easy800-SWD
	64	16

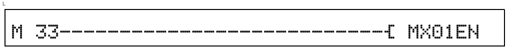
Example

A drive is required to run at a fixed speed and at three dependent speeds.

It receives the setting from a multiplexer function block.

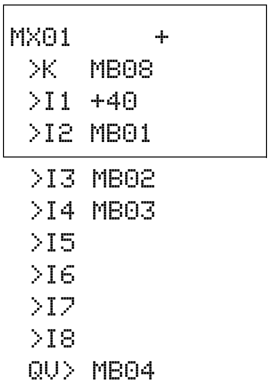
MB08 defines the number of the channel that is to reach the output.
The fixed speed is set as a constant at input MX01I1, the subsequent inputs are assigned the variable values from markers MB01 to MB03. The setpoints at output MX01QV of the function block are output at MB04.

The function block is first activated in the circuit diagram by marker M33.
The following figure shows the association.



```
M 33----------------------------[ MX01EN
```

Figure 160:Circuit diagram for enabling the data multiplexer function block



```
MX01      +
 >K  MB08
 >I1 +40
 >I2 MB01
 >I3 MB02
 >I4 MB03
 >I5
 >I6
 >I7
 >I8
 QV> MB04
```

Enter the parameters of the function block as shown on the left.

5.23 NC, numerical converter

This function block is available for easy800-SWD from device version 01 and for easy800 standard devices from device version 04.

The numerical converter function block is used to convert a value of:

- BCD in binary value.
- Binary value in BCD.

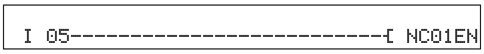
The binary value is further processed in the device. For example it can be added to a value. The binary value is displayed in the simulator as a decimal value. easy800 device allows the use of up to 32 numerical converters.

Contacts and Coils

Coil NC..EN is used to activate the function block. If the coil NC..EN is not active, the entire function block is deactivated and reset. The output value is set to zero.

Coils	Function
NC..EN	Enables the function block

Example:

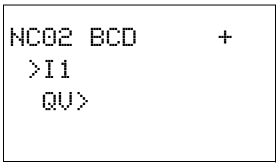


```
 I 05--------------------------[ NC01EN
```

Figure 161:Wiring the enable coil

Its coil NC..EN is directly connected to the device terminal IS5.

Parametric programming of the function block inputs/outputs



```
NC02 BCD      +
 >I1
  QV>
```

NC02	Function block; Numerical converter, number 02
BCD	Operating mode: Converting BCD to binary
+	Parameter set can be called via the PARAMETERS menu
>I1	Input value: Decimal: -161061273 - +161061273 BCD: -9999999 - +9999999
QV>	Output value: Decimal: -161061273 - +161061273 BCD: -9999999 - +9999999

Operating mode

The operating mode determines whether a binary value is to be converted to BCD or vice versa.

Parameters	Function, parameterization in the function block
BCD	Converting BCD to binary
BIN	Converting binary to BCD

The factory setting of this parameter is BCD.

Number range of the BCD code

The BCD code only allows the number range 0 to 9. Each digit of a decimal number is represented in BCD code by four bits. The number 134_{dec} thus corresponds to the BCD code 0001 0011 0100. The decimal representation of the BCD code is 308.

BCD code	Decimal value
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010 - 1111	Not permissible

Input

Assign the input value to be converted to function block input I1.

The function block input I1 can be assigned the following operands:

- Constant.
- Markers MD, MW, MB.
- Analog inputs IA01 to IA04:
  - IA01: Terminal I7.
  - IA02: Terminal I8.
  - IA03: Terminal I11.
  - IA04: Terminal I12.
- Analog output QA01.
- Output …QV> of a different function block.

Output

Output QV> supplies the converted value.

You can assign the following operands to the function block output NC..QV:

- Analog output QA01.
- Markers MD, MW, MB.

Function of the function block

BCD mode

The BCD value present at input I1 is converted to a binary value and supplied on the output QV. The binary value is displayed as a decimal value.

Examples:

I1 (dec)	BCD →	NC	BIN →	QV (dec)
9	0000 1001		1001	9
17	0001 0001		1011	11
25	0001 1001		0001 0011	19
19	0001 0011		1101	13
161061273	0000 1001 1001 1001 1001 1001 1001 1001		0001 0001 0001 0110 0111 1111	9999999
-18585	1000 0000 0000 0000 1001 0000 1001 1001		10000 0000 0000 0000 0010 0011 1000 1011	-9099

Bit 32 is the sign bit.
Bit 32 = 1 → Sign = Minus.

As each value is represented with 4 nibbles and each nibble can take on the value 9 in BCD code, the highest number that can be represented is 9999999. The smallest representable number is -9999999.

BIN mode

The binary value present at input >I1 is converted to a BCD value and supplied at the output QV>. The binary value is displayed as a decimal value.

Examples:

I1 (dec)	BIN	BCD	QV (dec)
7	0000 0111	1001	7
11	0000 1011	0001 0001	17
19	0001 0011	0001 1001	25
13	0000 1101	0001 0011	19
9999999	0001 0001 0001 0110 0111 1111	0000 1001 1001 1001 1001 1001 1001 1001	161061273
-9099	10000 0000 0000 0000 0010 0011 1000 1011	1000 0000 0000 0000 1001 0000 1001 1001	-18585

→ Bit 32 is the sign bit. Bit 32 = 1 → Sign = Minus.

→ Values greater than 9999999 are output as 161061273.
Values less than –9999999 are output as –161061273.
The working range of the function block has been exceeded.

Memory	32 + per 4 Byte/Constant at function block input	
Cycle time [µs]	easy800 standard	easy800-SWD
BCD	34	14
BIN	127	28

5.24 OT, operating hours counter

The function block is used to record the operating hours of independent events or states. The status of the operating hours counter is also retained when in a de-energized state. As soon as the enable coil of the function block is active, the easy800 counts the hours in minutes, although the display only shows hours.

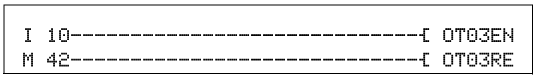
easy800 device allows the use of up to 4 hours-run counters.

Contacts and Coils

Contacts	Meaning
JC..Q1	Reaching the upper limit setpoint

Coils	Function
OT..EN	Selective activation of the function block
OT..RE	Reset the counter to zero

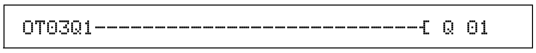
Example of an operating hours counter:



```
 I 10-----------------------------[ OT03EN
 M 42-----------------------------[ OT03RE
```

Figure 162:Wiring the function block coils

The enable coil of the function block is connected here directly to the device inputs. A marker activates the reset coil.

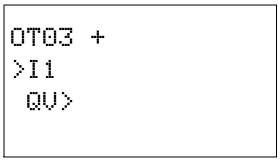


```
 OT03Q1---------------------------[ Q 01
```

Figure 163:Wiring of the function block contact

The message of the function block was wired to a device output.

Parametric programming of the function block inputs/outputs



```
OT03 +
>I1
 QV>
```

When using the function block in the circuit diagram for the first time, use OK to automatically enter the general display of function block parameters, as shown in the figure on the left. Enter the function block settings here. The display contains the following elements:

OT03	Function block: Operating hours, number 03
+	Parameter set can be called via the PARAMETERS menu
>I1	Upper setpoint in hours: 0 hours - > 100 years
QV>	Actual value of the operating hours counter

Input

The >I1 block input is used to specify the upper setpoint value in hours. When it is reached, the block contact sends a signal.

The function block input I1 can be assigned the following operands:

- Constant.
- Markers MD, MW, MB.
- Analog inputs IA01 to IA04:
  - IA01: Terminal I7.
  - IA02: Terminal I8.
  - IA03: Terminal I11.
  - IA04: Terminal I12.
- Analog output QA01.
- Output ...QV> of a different function block.

Output

The output QV supplies the counted operating hours as an integer value in hours.

You can assign the following operands to output ...QV:

- Analog output QA01
- Markers MD, MW, MB

Function of the 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.

→ The following operations do not clear the actual value of the hours-run counter: Operating mode change RUN, STOP, voltage ON, OFF, Delete program, Change program, Load new program.

Memory	36 + per 4 Byte/Constant at function block input	
Cycle time [µs]	easy800 standard	easy800-SWD
	25	9

Accuracy

The operating hours counters are accurate to the nearest minute. If the enable coil signal is terminated within a minute, the value for seconds is lost. The value range of the operating hours counter is between 0 hours and 100 years.

5.25 PO, pulse output

This function block is available for easy806-SWD from device version 01 and for easy800 standard devices from device version 07. This function block is not available for EASY802-DC-SWD it has no local outputs.

The Pulse output function block enables you to generate a defined number of pulses or pulse sequence at output Q01 or Q2. You can change the frequency within the pulse sequence. You can generate several pulse sequences in defined intervals. The function block also has Jog mode as well as normal mode.

Two function blocks are available: PO01 and PO02. Both function blocks are permanently assigned to device outputs Q01 and Q02: If you use function block PO01, the pulse sequence is output at device output Q01, device output Q02 is connected at function block PO02. The outputs Q01 and Q02 used must not be used again in the circuit diagram. The reason for this is that the PO function blocks overwrite all other status changes at the device outputs Q01 and Q02. This can cause unpredictable switching states.

→ Avoid unforeseeable switch states. Only use each coil of a relay once in the circuit diagram.

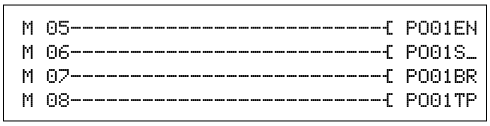
The function block can be used in order to control a stepper motor via a power amplifier.

Contacts and Coils

Contacts	Meaning
PO..AC	as long as pulses are output at device output Q01 or Q02.
PO..E1	Error: FO < FS (Operating frequency < Start frequency) PF < FS (Jog frequency < Start frequency)

Coils	Function
PO..EN	Activation of the function block.
PO..S_	Start of normal operation with a rising edge. Active normal operation is acknowledged with AC = 1.
PO..BR	Aborting of normal mode with a rising edge.
PO..TP	Activates the Jog mode .

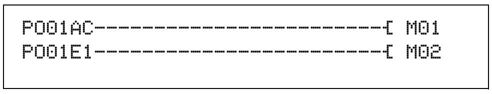
Examples



```
M 05---------------------------[ PO01EN
M 06---------------------------[ PO01S_
M 07---------------------------[ PO01BR
M 08---------------------------[ PO01TP
```

Figure 164:Wiring the function block coils

The function block coils are switched by markers.

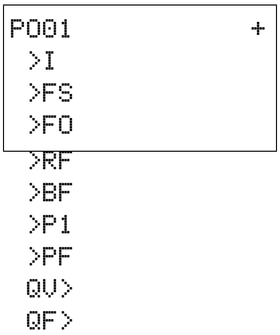


```
PO01AC------------------------[ M01
PO01E1------------------------[ M02
```

Figure 165:Wiring the function block contacts

The contacts of the function block are connected with markers.

Parametric programming of the function block inputs/outputs



```
PO01            +
 >I
 >FS
 >FO
 >RF
 >BF
 >P1
 >PF
 QV>
 QF>
```

When using the function block in the circuit diagram for the first time, use OK to automatically enter the general display of function block parameters, as shown in the figure. Enter the function block settings here.

PO01	Function block: PO pulse output, number 01
+	Parameter set can be called via the PARAMETERS menu
>I1	Number of pulses for the entire distance, value range 0 - 2147483647
>FS	Start frequency, value range 0 - 5000 Hz
>FO	Operating frequency, value range 0 - 5000 Hz
>RF	Frequency change in the acceleration phase [0.001 Hz/step], value range 0 - 65535
>BF	Frequency change in the braking phase [0.001 mHz/step], value range 0 - 65535
>P1	Number of steps in jog mode (0 - 65535)
>PF	Frequency in jog mode (0 - 5000 Hz)
QV>	Current number of completed steps, value range: 0 - 2 147 483 647
QF>	Actual output frequency, value range 0 - 5000 Hz

Function of the function block

The function block is active if the PO..EN coil is actuated. After you have parameterized the function block, you can actuate the coil PO..S_. This will start normal operation. Alternatively, you can also actuate the coil PO..TP and start the function block in Jog mode.

If the voltage at the coil PO..EN drops out, the pulse sequence at the output is immediately switched off and the function block is reset.

→ Bear in mind that a motor with sufficient inertia may not come to a standstill immediately after the pulse sequence is switched off.

The function block sends 24 V pulse sequences at output Q01 (function block PO01) or Q02 (function block PO02). The frequency of the pulse sequence is determined by the parameters that are described in the sections Normal mode and Jog mode.

Procedure with possible faults for normal and jog mode

- If the function block detects parameter errors directly at the start, no pulse sequences are output to the device output.
- If the function block detects a faulty parameter change during a pulse sequence, the pulse sequence is reduced in accordance with the brake ramp.

- If one of the two device outputs Q01 or Q02 is used several times due to a user error in the circuit diagram, the pulse sequence of the PO function block is not output at the device output.

The function block can be used for normal operation and jog mode.

Normal operation

The following parameters are required for normal mode:

I1	Number of pulses
FS	Start frequency
FO	Operating frequency
RF	Frequency change in acceleration phase
BF	Frequency change in braking phase

A rising edge at the PO..S_ coil starts normal operation.

A pulse sequence with the start frequency is initiated at the device output. The frequency starts continuously up to the operating frequency, depending on parameter RF. The function block determines the start of the braking phase from the parameters I1, BF, FO and FS. From this point, the operating frequency is continuously reduced down to the starting frequency as a function of parameter BF.

After the function block is started properly, contact AC is closed until the braking phase is completed. Contact E1 is closed if the function block detects incorrect parameters at the start.

In order to stop Normal mode, actuate the BR coil. In this case, the pulse frequency will be reduced according to the braking ramp.

A further rising edge at S has no effect if normal mode is activated.

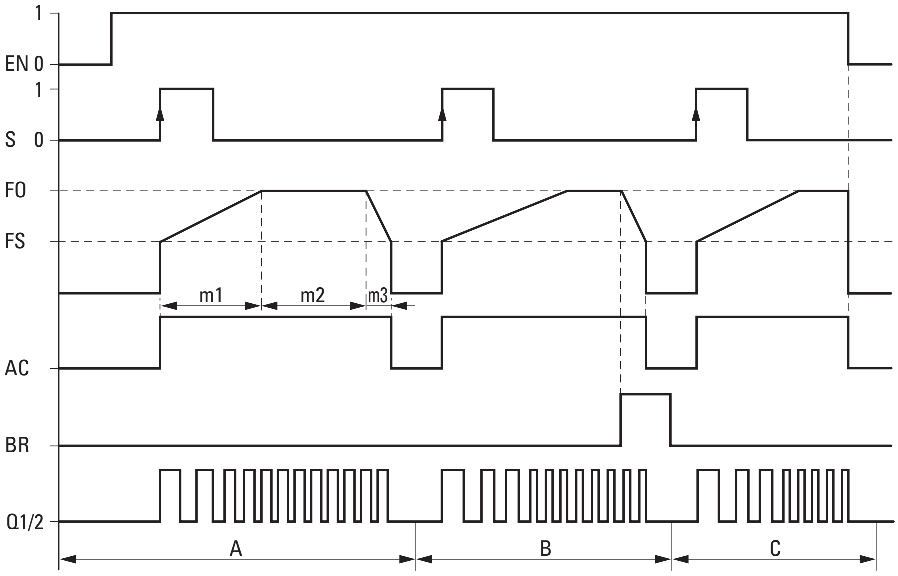


Figure 166:PO pulse output - possible phases of Normal mode

EN: Enable coil
S: Start coil for pulse sequence
FO: Start frequency, FS = Operating frequency,
m1 = Acceleration phase, m2 = Operating phase, m3 = Braking phase
AC: Pulse sequence applied at device output, contact
BR: Stop coil for pulse sequence
Q1/2: Pulse sequence at the device output Q1 and/or Q2

- Range A: The pulse sequence is present at the device output until the number of pulses defined at I1 has been reached.
- Range B: Activating the coil PO..BR initiates the braking phase and reduces the frequency of the pulse sequence.
- Range C: A voltage drop at coil PO..EN immediately switches off the pulse sequence from the device output.

The function block calculates the required number of pulses for the acceleration and braking phase with the following formula:

Acceleration phase

$$n_1 = \frac{(FO - FS)}{RF} \cdot 1000$$

Braking phase

$$n_3 = \frac{(FO - FS)}{BF} \cdot 1000$$

FS = Start frequency [Hz], FO = Operating frequency [Hz]
n = Total number of pulses for the entire distance
n_1 = Number of pulses in acceleration phase
n_3 = Number of pulses in operating phase

RF = Frequency change in the acceleration phase [mHz/step]
BF = Frequency change in the braking phase [mHz/step]

Example with Normal mode:

Set parameter:FO = 3000 Hz, FS = 200 Hz, RF = BF = 1000

Result: $n_1 = n_3 = 2800$

Example with Normal mode:

1000 = Frequency change by 1 Hz per step

P1: Number of pulses in Jog mode,

Jog mode

Jog mode (TP = 1) can be activated for diagnostics and testing purposes. For this coil PO..TP is actuated. The following parameters are required for Jog mode:

FS	Start frequency
P1	Number of pulses in Jog mode
PF	Frequency in jog mode

The behavior of the function block in Jog mode varies according to the length of the signal at coil PO..TP:

1. TP on duration < 0.5 seconds:
The function block generates at the device output the number of pulses defined at P1 with the starting frequency defined at FS.

see range “A“ in → Figure 167, page 234.

2. TP on duration ≥ 0.5 seconds:

The function block generates at the device output the start frequency set at FS for 0.5 s and then accelerates up to the Jog frequency PF.

The pulse operation can be interrupted at any time by a 0 signal at the TP input.

If Jog mode is deactivated (TP = 0) while a pulse sequence is output at the device output with a Jog frequency (time point E), the pulse sequence is continuously reduced to the start frequency (braking phase) see Range B in → Figure 167, page 234.

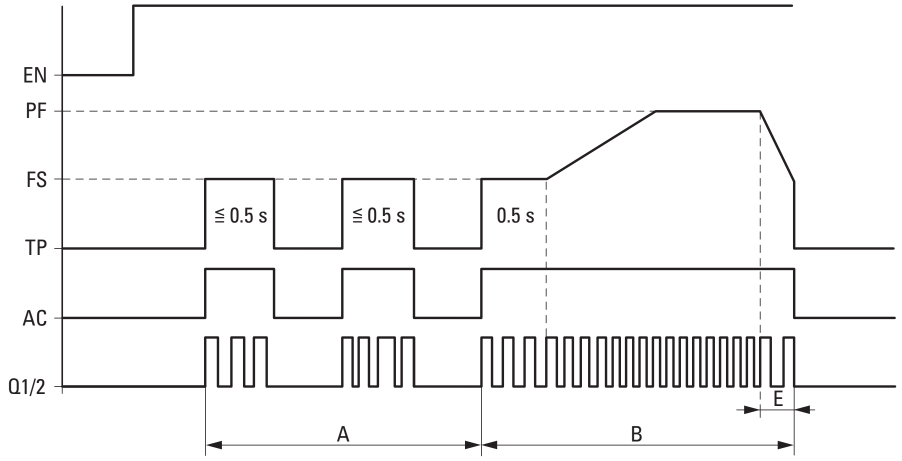


Figure 167:PO pulse output - jog mode

FS = Start frequency
PF = Jog frequency
A = Number of pulses at TP on time < 0.5 s
B = Number of pulses at TP on time > 0.5 s
E = Braking phase

If a very small number of pulses are set, the start pulse at the TP input also has to be very short. Otherwise the function block will output several pulse sequences (see range A Figure 166).

Application

The function block is often used in order to control a stepper motor via a power amplifier. The motor converts pulse sequences into steps. The number of pulses therefore determines the travel distance. The frequency of the pulses determines the speed of travel. The travel behavior of a stepper motor in Normal mode can be divided up into three phases, which the PO function block supports with variable frequencies:

- Acceleration: Frequency increase that the motor converts into acceleration; starting from a starting frequency up to operating frequency
- Operation: constant operating frequency
- Braking: Frequency reduction that the motor converts into braking; starting from the operating frequency down to the starting frequency

If you require a direction signal for bidirectional stepper motors, use one of the other device outputs.

Jog mode is often used to set up an installation.

Memory	96 + per 4 Byte/Constant at function block input

5.26 PT - Put value to the NET

The PUT function block PT.. enables you to transfer a 32-bit operand to the NET. The operand value is transferred and automatically read by the corresponding GET function block GT of another NET station.

easy800 device allows the use of up to 32 PUT function blocks. EASY802-DC-SWD is not provided for the function block.

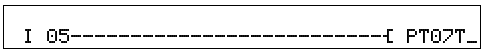
Contacts and Coils

Contacts	Meaning
PT..Q1	The trigger coil is active
PT..AC	The temporary storage and subsequent transfer of the input value to the NET. 1).
PT..E1	A transfer error.[1]

[1] These outputs are available for EASY806-DC-SWD from device version 01 and for easy800 standard from device version 07.

Example of a PUT function block:

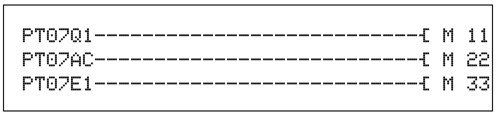
Coils	Function
PT..T_	Trigger coil, on a rising edge the function block temporarily stores the input value and transfers it to the NET



```
 I 05---------------------------[ PT07T_
```

Figure 168:Wiring the trigger coil

The trigger coil is connected to a device input.

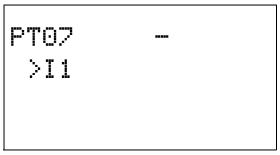


```
PT07Q1----------------------------[ M 11
PT07AC----------------------------[ M 22
PT07E1----------------------------[ M 33
```

Figure 169:Wiring the function block contacts

The contacts transfer the function block information to markers.

Parametric programming of the function block inputs/outputs



```
PT07      -
 >I1
```

When using the function block in the circuit diagram for the first time, use OK to automatically enter the general display of function block parameters, as shown in the figure on the left. Enter the function block settings here. The display contains the following elements:

PT07	Function block: PUT, number 07
-	Parameter display: Call not enabled
>I1	Input value that is to be put onto the NET: -2147483648 - 2147483647

Input

I1: When the function block is triggered, the value present at function block input >I1 is temporarily stored and then transferred to the NET. The function block input can be assigned the following operands:

- Constant.
- Markers MD, MW, MB.
- Analog inputs IA01 to IA04:
  - IA01: Terminal I7.
  - IA02: Terminal I8.
  - IA03: Terminal I11.
  - IA04: Terminal I12.
- Analog output QA01.
- Output ...QV of a different function block.

PUT diagnostics

The PUT function block only functions if the NET is running properly (→ section 9.6, “easy800 standard expanding”, page 325).

Function of the function block

When the trigger coil PT..T_ picks up, the contact PT..Q1 and stays closed for as long as the trigger coil remains activated.

The pickup triggers the buffering of the input value present at input I1. The function block then transfers this value to the NET. During this time, the function block contact PT..Q1 closes.

If the temporarily stored value is not sent, the contact PT..E1 closes and therefore indicates a transfer error. It stays closed until the trigger coil PT..T_ picks up to issue a new send job.

The signal diagram shows the relationship of the send job and its triggering by the trigger coil:

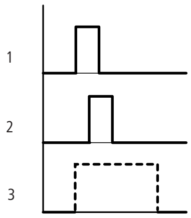


Figure 170:Signal diagram of PUT function block

1: Trigger coil

2: trigger coil contact feedback

3: Send

Memory	36 + per 4 Byte/Constant at function block input	
Cycle time [µs]	easy800 standard	easy800-SWD
	25	9

5.27 PW, pulse width modulation

This function block is available for easy806-SWD from device version 01 and for easy800 standard devices from device version 04. EASY802-DC-SWD is not available as this does not have any local outputs.

The PW.. function block modulates the mark to space ratio of a square wave signal. It therefore changes the ratio of on and off time and the pulse duration. The period duration of the signal stays constant.

The function block can be used to convert the manipulated variable of a PID controller into pulses. The maximum frequency is 200 Hz. This corresponds to a minimum period duration of 5 ms. The maximum period duration is 65.5 s.

→ When using the pulse width modulation function block with a minimum on time of less than 1 s only use devices with transistor outputs.

Two function blocks are available: PW01 and PW02. Both function blocks are permanently assigned the device outputs Q01 and Q02: If you use the PW01, the pulse sequence is output at device output Q01; with the PW02 function block, the device output Q02 is activated. When the outputs Q01 and Q02 are used with the PW function block, they must not be used again in the circuit diagram. The reason for this is that the PW function blocks overwrite all other status changes at the device outputs Q01 and Q02. This can cause unpredictable switching states.

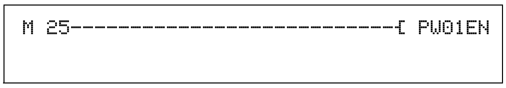
→ Avoid unforeseeable switch states. Only use each coil of a relay once in the circuit diagram.

Contacts and Coils

Contacts	Meaning
PW..E1	Error: Time below minimum on or off duration

Coils	Function
PW..EN	Selective activation of the function block

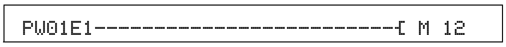
Example



```
M 25---------------------------[ PW01EN
```

Figure 171:Wiring the enable coil

The enable coil PW..EN is directly assigned to marker M25.

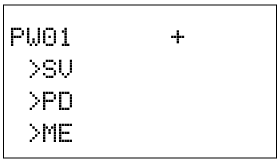


```
PW01E1-------------------------[ M 12
```

Figure 172:Wiring of the signal contact

The function block contact PW..E1 signals to marker M12.

Parametric programming of the function block inputs/outputs



```
PW01      +
 >SV
 >PD
 >ME
```

When using the function block in the circuit diagram for the first time, use OK to automatically enter the general display of function block parameters, as shown in the figure on the left. Enter the function block settings here. The display contains the following elements:

PW01	Function block; Data multiplexer, number 01
+	Parameter set can be called via the PARAMETERS menu
>SV	Manipulated variable: 0 - 4095
>PD	Period duration: 5 - 65535 ms
>ME	Minimum on time = Minimum off time: 1 - 65535 ms

You can assign the following operands to function block inputs SV and PD to ME:

- Constant
- Markers MD, MW, MB
- Analog inputs IA01 to IA04:
  - IA01: Terminal I7
  - IA02: Terminal I8
  - IA03: Terminal I11
  - IA04: Terminal I12
- Analog output QA01
- Output …QV> of a different function block

Function of the function block

If the function block is enabled by 1 at the PW..EN coil, the set value at input SV is transformed into a pulse sequence with a constant period duration (PD). The pulse width (PW) is therefore proportional to the manipulated variable (SV). The period duration and the minimum on time can be selected as required within the set limits.

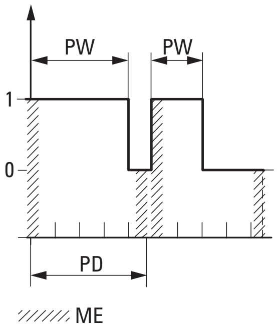


Figure 173:Pulses at the function block output

PW: Pulse width

PD: Pulse duration

ME: Minimum on duration, Minimum off duration

The function block causes a direct output of the pulse to the corresponding device output (Q1 and Q2).

The following applies to the minimum on duration:

- Minimum on time = minimum off time = ME
- ME ≧ 1 ms
- ME ≦ 10 % PD
- If the actual value of the pulse width is less than ME, the output stays at 0 and the contact E1 closes.
- If the off time of the pulse at the output is less than ME, the output stays at 1, Q1 or Q2 are in continuous operation. The contact E1 closes.
- If ME stays at 0 or is not connected, ME is preset to 0.1 ms.

→ 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.

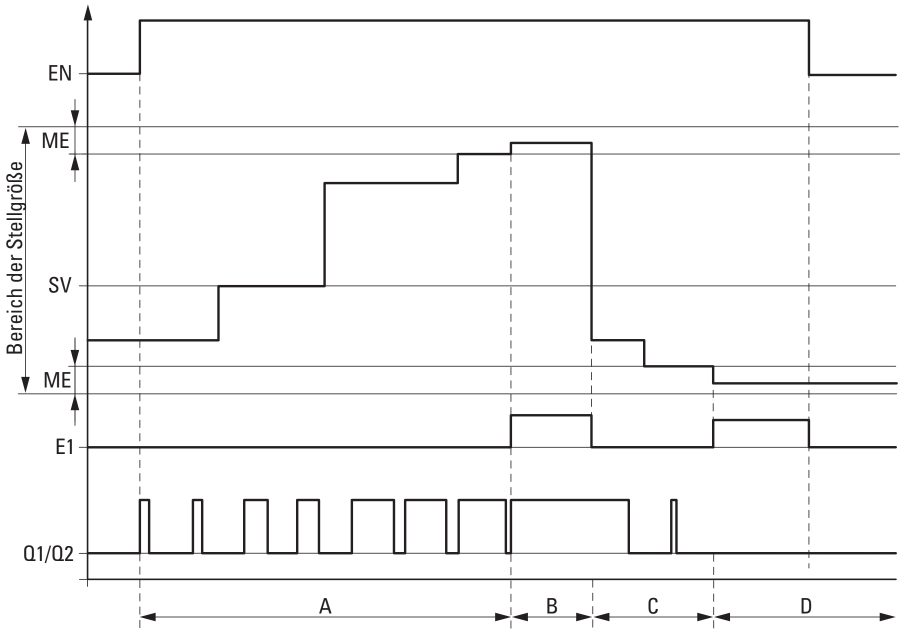


Figure 174:Pulse width modulation signal diagram

EN: Enable coil
SV: Manipulated variable
ME: Minimum on time or minimum off time
E1: Error in indication contact
Q1/2: Pulse sequence at the device output Q1 and/or Q2

Memory	48 + per 4 Byte/Constant at function block input	
Cycle time [µs]	easy800 standard	easy800-SWD
	243	20

5.28 SC, set 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.

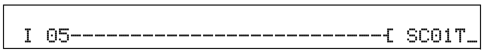
easy800 device allows the use of one “set date/time function block”. EASY802-DC-SWD is not provided for the function block.

Contacts and Coils

Contacts	Meaning
SC01Q1	The trigger coil is active

Coils	Function
SC01T_	Trigger coil, on rising edge the function block transfers date and time of the sending station to the NET

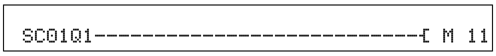
Example of a set date/time function block:



```
I 05--------------------------[ SC01T_
```

Figure 175:Wiring the trigger coil

The trigger coil is connected to a device input.



```
SC01Q1---------------------------[ M 11
```

Figure 176:Wiring of the function block contact

The contact is assigned to a marker.

Parametric programming of the function block inputs/outputs

This function block is a triggerable system service. It therefore does not have any parameters.

SC diagnostics

The SC function block only functions if the easyNet network is running properly (→ section 9.6, “easy800 standard expanding”, page 325).

Function of the 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 easyNet network. The sending stations executes this operation as soon as a new minute starts on the device real-time clock. The other stations accept this value.

Memory	20 Byte	
Cycle time [µs]	easy800 standard	easy800-SWD
	23	8

Example

The trigger pulse is set at 03:32:21 (hh:mm:ss). The other stations are synchronized at 03:33:00. The time is accepted by all stations. The procedure can be repeated as often as required.

Accuracy of time synchronization

The maximum time deviation between the functional stations is 5 s.

5.29 SP, serial protocol

This function block is available for easy800-SWD from device version 01 and for easy800 standard devices from device version 07.

The function block parameters can only be set via easySoft or easySoft-Pro. The SP function block enables you to send texts or data to a printer or terminal via the serial interface. With a connected modem, it is also possible to send data/texts to data terminals via a leased line or a dial-up connection. The serial interface is the same one to which you also connect the PC. You enter the parameters of the SP function blocks, as well as the text for the D function block, on the PC when creating the circuit diagram in the easySoft-Pro programming software. 32 Serial Protocol function blocks are available. In this way you can send up to 32 texts including variables.
64 characters from the displayable range of the ASCII table (32d to 126d) are available for creating text.

Contacts and Coils

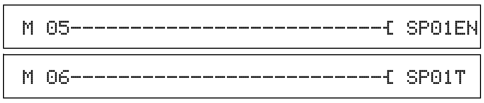
Contacts	Meaning
SP..AC	Function block is active
SP..E1	Send error, e.g. due to incorrect telephone number

After an error is detected there must be an edge change on SP..EN from 1 -> 0 in order for a new send job to be started. For this you must activate SP..EN once more.

Coils	Function
SP..EN	Activate the function block
SP..T_	On a rising edge the function block sends the text of the corresponding text module via the serial interface.

When the modem is activated, the connection to the connected device is established for dial-up connections when SP..EN = 1 is detected for the first time. The connection is not closed until SP..EN = 0 on all SP function blocks used or the device is set in the STOP state.

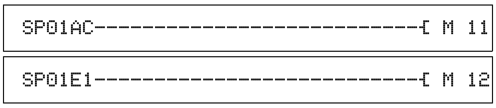
Example of the SP function block:



```
M 05---------------------------[ SP01EN
M 06---------------------------[ SP01T
```

Figure 177:Wiring of the coils

In this example, the SP function block is number 01. It takes the texts from the D text function block with the number 01, → Section “5.11 D, text display“, Page 181.



```
SP01AC----------------------------[ M 11
SP01E1----------------------------[ M 12
```

Figure 178:Wiring the function block contacts

Parametric programming of the function block inputs/outputs

If you enter a coil of the function block in the circuit diagram, e.g. SP..EN or SP..T, or drag the SP function block onto the function block diagram, the Circuit diagram element and Global communication parameters tab are displayed. You can change the function block number and add a comment in the Global communication parameters tab. This shows 3 fields:

- Communication parameters.
- Start sequence.
- End sequence.

Communication parameters

The Communication parameters field contains the following parameters:

Parameters	Function, parameterization in the function block
Telephone number with dial-up parameters	Only with modem connected. The SP function block places <ATD> string in front of the dial-up sequence you define. The dial-up parameters are shown in → Figure 17.
Baud	9600, 19200
Data Bit	normally 8 bits
Stop Bit	1, 2
Parity	none, even, uneven
Time Stamp	Date and time are added to the text
Leased Line	This option can be selected after the modem function is activated. When the leased line is activated, the text is sent immediately when SP..T is switched on, since it assumes that a connection is present.
Modem	Connecting devices via modem (only if the leased line is deactivated): 1. Activate the modem function. 2. Switch on coil SP..EN (activate SP function block). 3. Activate coil SP..T (send data): • The SP function block establishes a dial-up connection based on the telephone number stored and sends the data.
Carriage return, automatic	If the Automatic line break option is activated, the SP function block sends a control sequence that causes a connected printer or terminal to carry out an automatic line break after each text line output. This option applies to all SP function blocks and only therefore has to be activated by one function block. It can help you, for example to create highly legible label prints.

Table 17: Dial-in parameter

Parameters	Function
P	Pulse dialing
T	Tone Dialing Procedure
W	Wait for dial tone from the outside line (important for PBX systems). Example of the entry of a tone dial string with wait time: T 00 W 0228 123456
,	(comma) Pause before dialing the subsequent numbers (the pause duration in seconds is determined by the value in the modem register S8).

Start Sequence

Up to 32 characters from the range (0...31d) of the ASCII table (→ section 12.11, "ASCII Table", page 380), each with a preceding backslash "\", can be entered for the start sequence. Backslash and subsequent ASCII character form together one character of the start and end sequence. For example, you use the string \LF for the line feed and \CR for carriage return as a printer control character in order to represent a text block in the first column of a new line.

End Sequence

The same procedure must be observed for the end sequence as for the start sequence (→ section 12.11, "ASCII Table", page 380). For example, you use the \NUL (end of string) as the normal end sequence.

Structure of a send telegram

total length up to 158 characters

Start Sequence	Time Stamp	ASCII characters of a text function block D01 - D23	End Sequence
0 - 63 characters optional	20 characters optional	64 characters	0 - 11 characters optional

Memory requirement

228 bytes are required for the first SP function block that is parameterized and a further 24 bytes are required for any other SP function block instantiated. Allow also for the additional 160 bytes required for a D function block.

Function of the function block

This function block outputs texts/data via the serial interface. Enter the interface parameters in the function block diagram of easySoft-Pro. After you have dragged a coil (SP..EN/SP..T) onto the circuit diagram or the function block onto the function block diagram you can enter the interface parameters in the Global communication parameters tab. The interface accepts these parameters as soon as the SP..EN coil is set on a called SP function block in RUN mode.

You create the text with the D text display function block. After you have dragged the coil (D..EN) onto the circuit diagram or the function block onto the function block diagram you can enter the text and variable in the Parameters tab. Entry via display/keypad is not possible. Four lines of 16 characters each are available for characters.

A D function block and an SP function block must be interlinked for text output by assigning both function blocks with the same function block number. To send a new text, a new D and SP function block combination must be created.

The text is output with every activation of the trigger coil SP..T, which also contains the function block number in its identifier, such as SP01T.
The SP function block accesses the D function block with the same number (01). The individual characters of the D function block including blanks are sent and shown in succession, for example, on a terminal: Starting with the leftmost character in the first row of the entry field up to the rightmost character of the fourth row. If a variable that is continually updated is to be displayed in the text, the enable coil of the D function block (D..EN) must have picked up.

If you are using several SP function blocks that are all activated with EN = 1 in your program, and which receive a send job at the same time with T = 1, the control relay then processes all jobs in succession before it can start with a new job.

→ As a large amount of memory is required for text output, the following solution should be used with networked control relays: one control relay (text output relay) is used for the entire text output. With 32 SP and D function blocks, this can output up to 32 texts. The text output is initiated by other control relays in the network. The network bit data SN is used by the control relays to send a signal to the text output relay, which in turn outputs the assigned text. The assignment of the received RN network bit data and the SP function block must be programmed in the text output relay.

5.30 SR, shift register

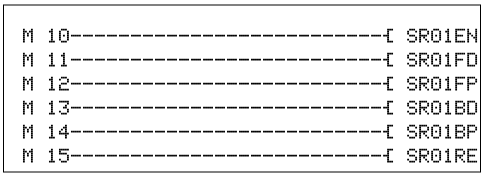
The shift register function block allows you to shift bits or marker double words one position up or down with every clock pulse applied. The Bit or DW mode can be defined via a parameter. You set the shift direction by activating the FP (Forward Pulse) coil or the BP (Backward Pulse) coil.
The values to be accepted in the shift register are located at different inputs depending on the shift direction and operating mode. A easy800 device allows the use of up to 32 shift registers.

Contacts and Coils

Contacts	Meaning
SR..Q1 - Q8	Output of the bit register fields 1 - 8

Coils	Function
SR..EN	Activation of the function block
SR..RE	Reset of the register
SR..FP	Shift direction forward
SR..BP	Shift direction backward
SR..FD	Bit data input, shift register forward
SR..BD	Bit data input, shift register backward

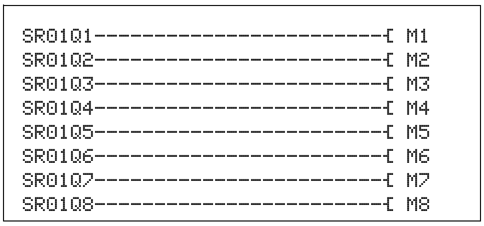
Examples



```
M 10---------------------------[ SR01EN
M 11---------------------------[ SR01FD
M 12---------------------------[ SR01FP
M 13---------------------------[ SR01BD
M 14---------------------------[ SR01BP
M 15---------------------------[ SR01RE
```

Figure 179:BIT operating mode, wiring the function block coils

The function block coils are switched by markers.

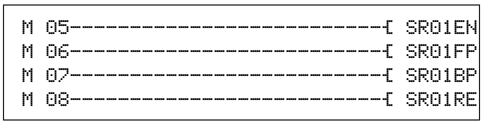


```
SR01Q1-------------------------[ M1
SR01Q2-------------------------[ M2
SR01Q3-------------------------[ M3
SR01Q4-------------------------[ M4
SR01Q5-------------------------[ M5
SR01Q6-------------------------[ M6
SR01Q7-------------------------[ M7
SR01Q8-------------------------[ M8
```

Figure 180:BIT operating mode, wiring the function block contacts

The contacts of the function block are connected with markers.

In DW operating mode you only integrate the coils in the circuit diagram:



```
M 05--------------------------[ SR01EN
M 06--------------------------[ SR01FP
M 07--------------------------[ SR01BP
M 08--------------------------[ SR01RE
```

Figure 181:DW operating mode, wiring the function block contacts

You assign parameters to inputs and outputs in the function block diagram.

Parametric programming of the function block inputs/outputs

When using the function block in the circuit diagram for the first time, use OK to automatically enter the general display of function block parameters (→ Page 249 figure left). Enter the function block settings here.

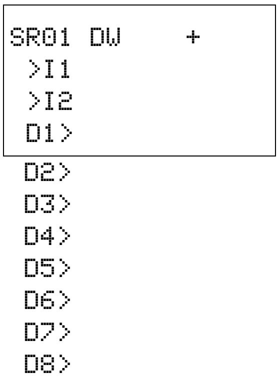
Operating mode

With this parameter you define whether you wish to run the shift register as a bit register or in double word format.

Parameters	Function, parameterization in the function block
BIT	Bit operation
DW	Double word operation

The factory setting of this parameter is BIT.

If the BIT operating mode is selected, the inputs I1, I2 and outputs D1-D8 are displayed. They have no function in BIT mode! If they are assigned operands, these will have no effect. The wiring of the SR function block (BIT) is carried out in the circuit diagram.



```
SR01 DW    +
 >I1
 >I2
 D1>
 D2>
 D3>
 D4>
 D5>
 D6>
 D7>
 D8>
```

SR01	Function block: SR shift register, number 01
DW	Operating mode: Double word
+	Appears in the parameter display
>I1	Input value DW forwards: -2147483648 - 2147483647
>I2	Input value DW backwards: -2147483648 - 2147483647
D1>	Register value 1 of the shift register, Value range of all registers: -2147483648 - 2147483647
D2>	Register value 2
D3>	Register value 3
D4>	Register value 4
D5>	Register value 5
D6>	Register value 6
D7>	Register value 7
D8>	Register value 8

Inputs

In DW operating mode you assign function block inputs I1 and I2 with the values that are to be read into the shift register. You can assign the following operands to function block inputs:

- Constant.
- Markers MD, MW, MB.
- Analog inputs IA01 to IA04:
  - IA01: Terminal I7.
  - IA02: Terminal I8.
  - IA03: Terminal I11.
  - IA04: Terminal I12.
- Analog output QA01.
- Output ...QV of a different function block.

Outputs

Likewise, only in DW operating mode do the function block outputs D1 to D8 output register values 1 to 8. The outputs can have the following operands:

- Analog output QA01.
- Markers MD, MW, MB.

Function of the function block

The SR..EN coil is actuated.

You have set STOP mode

A rising edge at FP (ForwardPulse) causes the bit value to be accepted at data input FD (ForwardData) into the first register field Q1. The original contents of the register fields are then moved by one field in the direction of the next higher field numbers.

A rising edge at the BP (BackwardPulse) transfers the bit value at the BD (BackwardData) data input to the last register field Q8. The original contents of the register fields are then moved by one field in the direction of the next lower field number.

Example: Shift register BIT mode, forward

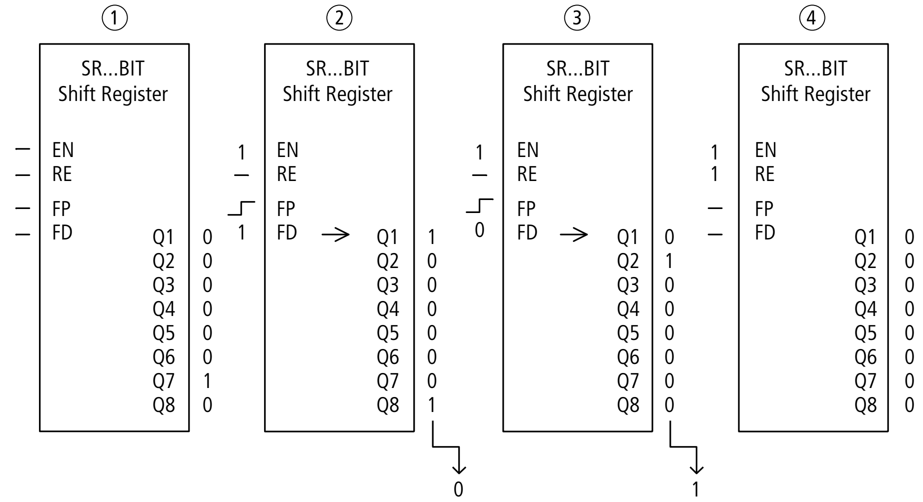


Figure 182:Shift register SR..: Forwards operation in BIT operating mode

① Initial situation
- SR..EN is not activated, the function block is not active.
- SR..Q7 contains a data bit 1, a 0 is contained in the other register fields.

② Transfer of a data bit:
- SR..EN is activated, the function block is active.
- SR..FD has the value 1.
- With the forwards pulse from SR..FP the register field SR..Q1 shifts the content of all register fields one place forwards and accepts the 1 from SR..FD.

③ Transfer of a data bit:
- SR..EN is activated, the function block is active.
- SR..FD has the value 0.
- with the forwards pulse from SR..FP the register field SR..Q1 shifts the content of all register fields one place forwards and accepts the 0 from SR..FD.

④ Reset of the register:
- SR..EN is activated, the function block is active.
- Activating SR..RE clears the content of the register.

You have set DW mode

A rising edge at FP (ForwardPulse) causes the double word value at data input I1 to be transferred to the first register field D1. The original contents of the register fields are then moved by one field in the direction of the next higher field numbers.

A rising edge at the BP (BackwardPulse) transfers the double word value at data input I2 to the last register field D8. The original contents of the register fields are then moved by one field in the direction of the next lower field numbers.

Shift direction	Coil	Control mode		
		BIT	DW	
Forwards	FP	FD	I1	Relevant data inputs
Reverse	BP	BD	I2	

Memory	96 + per 4 Byte/Constant at function block input	
Cycle time [µs]	easy800 standard	easy800-SWD
BIT	No entry	16
DW	No entry	28

Example: Shift register DW mode, backward

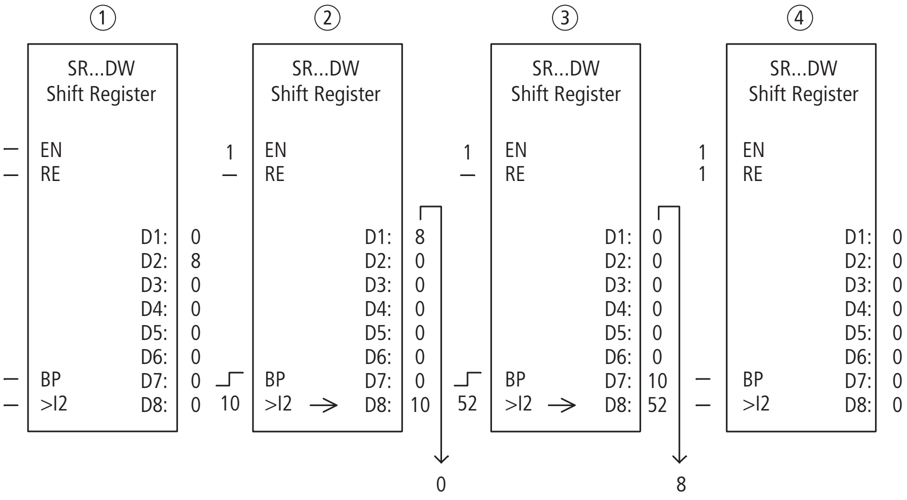


Figure 183:Shift register SR..: Backwards operation in DW operating mode

① Initial situation:
- SR..EN is not activated, the function block is not active.
- SR..D2 contains the value 8, a 0 is contained in the other register fields.

② Transfer of value:
- SR..EN is activated, the function block is active.
- SR..I2 has the value 10.
- With the backwards pulse from SR..BP the register field SR..D8 shifts the content of all register fields one place back and accepts the 10 from SR..I2.

③ Transfer of value:
- SR..EN is activated, the function block is active.
- SR..I2 has the value 52.
- With the backwards pulse from SR..BP the register field SR..D8 shifts the content of all register fields one place back and accepts the 52 from SR..I2.

④ Reset of the register:
- SR..EN is activated, the function block is active.
- Activating SR..RE clears the content of the register.

Application example

Different workpieces are moved along a production line with several machining stations. An operator determines the work required for the individual workpieces, creates a production code for it and writes it into a shift register.

The workpieces reach the machining stations in this order. When the workpiece is changed, the stations read the required production steps from their permanently assigned register field.

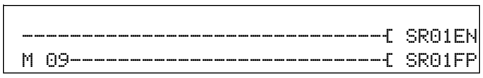
When workpiece 1 moves to the first station, the forwards pulse input coil SR01FP and the shift register SR01 receive production code 1 at input SR01I1 from the double word marker MD11.

Production code 1 is now at the register field SR01D1 for the first machining station which reads it from double word marker MD01.

The finished workpiece is then transferred to station 2.
The shift register accepts the production code 2 for the next workpiece. Production code 1 moves one place forwards, as does the remaining register content. It now stands at register output SR01D2. It reaches machining station 2 via double word marker MD02.

The process is repeated for each further workpiece and for each further machining station until the finished workpieces leave the line.

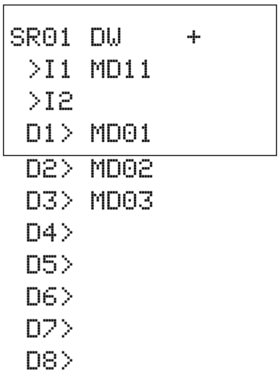
The coils are programmed in the circuit diagram of easy800:



```
 ------------------------------[ SR01EN
 M 09---------------------------[ SR01FP
```

Figure 184:easy800 circuit diagram for example 2

- The enable coil SR01EN is permanently active, the function block is not switched off.
- Marker M09 switches the forwards pulse input coil SR01FP.



```
SR01 DW    +
 >I1 MD11
 >I2
 D1> MD01
 D2> MD02
 D3> MD03
 D4>
 D5>
 D6>
 D7>
 D8>
```

The figure on the left shows the parameter display and the parameter set. After defining shift register number 01, you set here the following parameters:

- the DW operating mode for the double word marker format.
- the double word markers for receiving the production code.

5.31 ST, Set cycle time

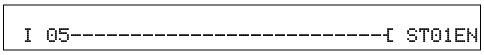
This function block is available for easy800-SWD from device version 01 and for easy800 standard devices from device version 04.

The Set cycle time function block is used as an additional function block for the PID controller and the counter function blocks. It provides a fixed cycle time for processing the circuit diagram and the function blocks. When used with a PID controller it improves the control response. It can also be used to suppress the sporadic lengthening of the cycle time when changing screens in the MFD. One function block is provided.

Contacts and Coils

Coils	Function
ST01EN	Selective activation of the function block

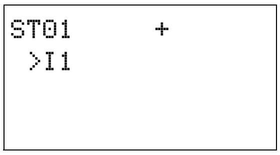
Example of a function block for setting the set cycle time



```
I 05--------------------------[ ST01EN
```

Figure 185:Wiring the enable coil

Parametric programming of the function block inputs/outputs



```
ST01     +
 >I1
```

When using the function block in the circuit diagram for the first time, use OK to automatically enter the general display of function block parameters, as shown in the figure on the left. Enter the function block settings here. The display contains the following elements:

ST01	Function block: Set cycle time, number 11
+	Parameter set can be called via the PARAMETERS menu
>I1	Set cycle time: 0 - 1000 ms

Input

I1: The set cycle time is entered at function block input I1.

The function block input I1 can be assigned the following operands:

- Constant.
- Markers MD, MW, MB.
- Analog inputs IA01 to IA04:
  - IA01: Terminal I7.
  - IA02: Terminal I8.
  - IA03: Terminal I11.
  - IA04: Terminal I12.
- Analog output QA01.
- Output ...QV of a different function block.

Function of the 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.

Memory		24 + per 4 Byte/Constant at function block input
Cycle time		

Memory	24 + per 4 Byte/Constant at function block input	
Cycle time [µs]	easy800 standard	easy800-SWD
	Remaining time up to set time	Remaining time up to set time

Cycle time less than 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 greater than set cycle time:

If the cycle time present is greater than the set cycle time, the set cycle time has no effect.

> *CAUTION*
>
> 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 independently of the cycle time.

5.32 T, timing relay

The T function block on easy800-SWD from device version 01 and on the easy800 standard from device version 04 has the additional modes Off-delayed with retriggering and Random switching.

With this timing relay you change the switch time and the on and off time of a switch contact. The adjustable delay times are between 50 ms and 99 h 59 min.

An easy800 device allows the use of up to 32 timing relays.

→ Page 269 on „Example with timing relay and counter function block“ provides information on the timing relay and counter function block.

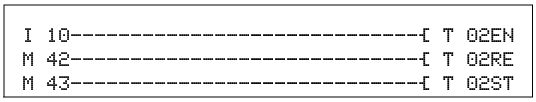
Contacts and Coils

A closed contact indicates:

Contacts	Meaning
T ..Q1	Time function active

Coils	Function
T ..EN	Selective activation of the function block
T ..RE	Reset of the timing relay to zero
T ..ST	Stop of the timing relay

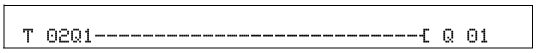
Timing relay example:



```
I 10------------------------------[ T 02EN
M 42------------------------------[ T 02RE
M 43------------------------------[ T 02ST
```

Figure 186:Wiring the function block coils

The trigger coil of the function block is connected here directly to the device inputs. A marker activates the reset coil, another marker activates the stop coil.

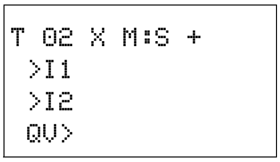


```
T 02Q1----------------------------[ Q 01
```

Figure 187:Wiring of the function block contact

The function block indication signal is assigned directly to a device output.

Parametric programming of the function block inputs/outputs



```
T 02 X M:S +
 >I1
 >I2
 QV>
```

When using the function block in the circuit diagram for the first time, use OK to automatically enter the general display of function block parameters, as shown in the figure on the left. Enter the function block settings here. The display contains the following elements:

T 02	Function block: Timing relay, number 02
X	Operating mode: On-delayed
m:s	Time range: Minute: Seconds
+	Parameter set can be called via the PARAMETERS menu
>I1	Setpoint 1
>I2	Setpoint 2 (for relay functions with 2 times)
QV>	Display: Elapsed time

Operating mode

This parameter defines the switch function of the timing relay:

Parameters	Switch function
X	On-delayed switching
?X	On-delayed switching with random time range
■	Off-delayed switching
?■	Off-delayed switching with random time range
X■	On- and off-delayed
□	Off-delayed with random time range, setpoint retriggerable 1)
?□	Off-delayed with random time range, retriggerable 1)
?X■	On- and off-delayed switching with random time, 2 time setpoints
⎍	Pulse shaping switching
⊔⊔	Flashing, synchronous, 2 setpoints
⊔⊔	Flashing, asynchronous, 2 setpoints

1) Only when using an easy800 standard control relay from device version 04 or easy800-SWD from device version 01.

Time Range

The function block parameter determines in which time range and at which resolution the relay operates.

Parameters	Time range and setpoint	Resolution
S 000.00	Seconds; 0.05 to 999.995 s for constants and variable values	5 ms
M:S 00:00	Minutes:Seconds; 00:00 to 99:59 only for constants and variable values	1 s
H:M 00:00	Hours:Minutes; 00:00 to 99:59 only for constants and variable values	1 min

→ Minimum time setting: 0.005 s (5 ms)
If a setpoint value is less than the easy800 cycle time, the elapsed time will only be recognized in the next cycle.

Inputs

I1 and I2: You assign a time value at I1 and I2, depending on the switch function of the relay.

You can assign the following operands to function block inputs I1 and I2:

- Constant.
- Markers MD, MW, MB.
- Analog inputs IA01 to IA04:
  - IA01: Terminal I7.
  - IA02: Terminal I8.
  - IA03: Terminal I11.
  - IA04: Terminal I12.
- Analog output QA01.
- Output ...QV of a different function block.

Output

QV: Output QV supplies the currently elapsed time. You can assign the following operands:

- Analog output QA01.
- Markers MD, MW, MB.

Retention

Timing relays can be run 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 is changed from RUN to STOP and when the power supply is switched off.

If easy800 is started in RUN operating 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.

- Status 1 with:
  - On-delayed.
  - Single pulse.
  - Flashing.
- Status 0 with:
  - Off-delayed.

Function of the function block

The relay is started via the trigger coil T…EN and can be selectively reset via the reset coil T…RE. The third coil T..ST terminates the run down of the actual time.

Memory	52 + per 4 Byte/Constant at function block input	
Cycle time [µs]	easy800 standard	easy800-SWD
	60	18

Variable time values

If you associate the function block inputs T ..I1 and T ..I2 with operands, you can use variable setpoints. The setpoints are transferred as follows, depending on the time range selected:

- S, value in milliseconds. The last digit is rounded up or down to 0 or 5, maximum value = 999995 ms.
- M:S, value in seconds, maximum value = 5999 s.
- H:M, value in minutes, maximum value = 5999 min.

→ The delay times are the same as described for the constants.

- Examples of time range S:
  - Operand value 9504 -> Time value is 9.50 s.
  - Operand value 45507 -> is 45.51 s.
- Examples of time range M:S:
  - Operand value 5999 -> Time value is 99 h, 59 min.
- Example of time range H:M:
  - Operand value 5999 -> Time value is 99 h, 59 min.

Different methods are used depending on the operating modes of the 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.

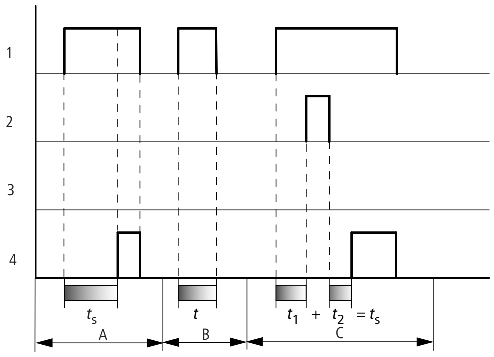


Figure 188:Signal diagram of timing relay, on-delayed (with and without random switching)

1: Trigger coil T...EN

2: Stop coil T...ST

3: Reset coil T...RE

4: Switching contact (N/O contact) T...Q1

t_S: setpoint time

- Range A: The time runs down from the set time value.
- Range B: The time does not elapse because the trigger coil drops out prematurely.
- Range C: The Stop coil stops the time from elapsing.

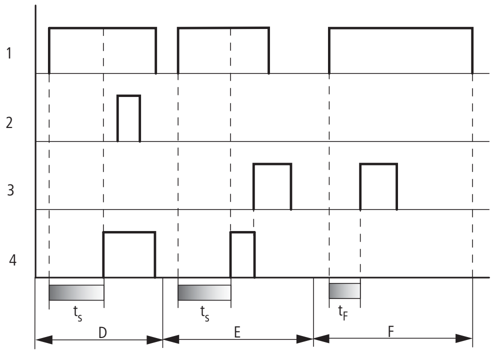


Figure 189:Signal diagram of timing relay, on-delayed (with and without random switching)

- 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: After the reset coil is activated, the switching contact is switched off and the internal time counter is reset. The function relay waits for a new trigger pulse.

Timing relay, off-delayed with and without random switching

Random switching, with and without retriggering
The contact of the timing relay switches randomly within the setpoint 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.

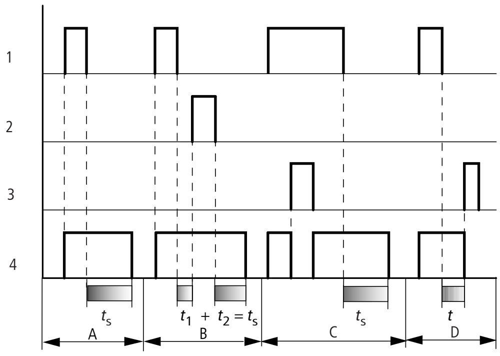


Figure 190:Signal diagram of timing relay, off delayed (with/without random switching, with/without retriggering)

1: Trigger coil T…EN

2: Stop coil T…ST

3: Reset coil T…RE

4: Switching contact (N/O contact) T…Q1

t_S: Setpoint time.

- Range A: The time elapses after the trigger coil is deactivated.
- Range B: The Stop coil stops the time from elapsing.
- Range C: The reset coil resets the relay and the contact.
  After the reset coil drops out, the relay continues to work normally.
- Range D: The Reset coil resets the relay and the contact when the function block is timing out.

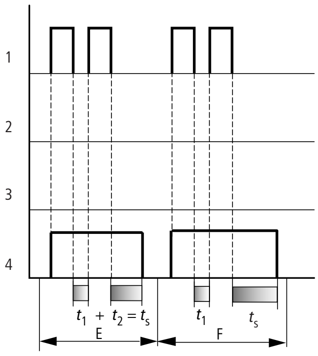


Figure 191:Signal diagram of timing relay, off delayed (with/without random switching, with/without retriggering)

- 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- and off-delayed with and without random switching

Time value I1: on-delay time
Time value I2: off-delay time

Random switching
The contact of the timing relay switches randomly within the setpoint value ranges.

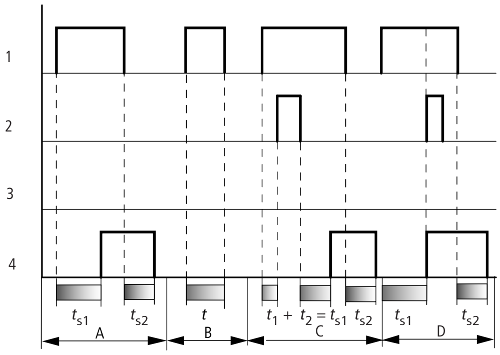


Figure 192:Operational diagrams timing relay, on and off delayed 1

1: Trigger coil T…EN

2: Stop coil T…ST

3: Reset coil T…RE

4: Switching contact (N/O 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.

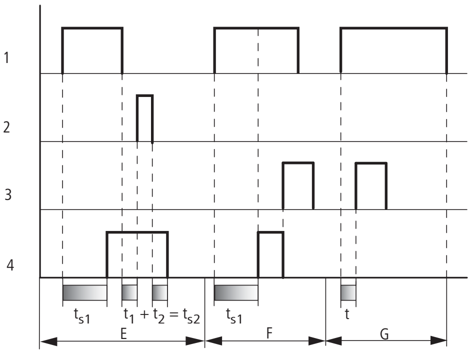


Figure 193:Operational diagrams timing relay, on and off delayed 2

- 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: After the reset coil is activated, the internal time counter is reset. The switching contact remains switched off. The function relay waits for a new trigger pulse.

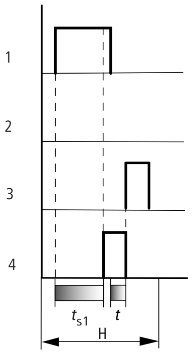


Figure 194:Operational diagrams timing relay, on and off delayed 3

- Range H: The Reset signal interrupts the timing out of the set time.

Timing relay, single pulse

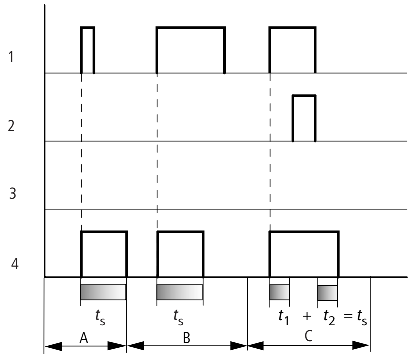


Figure 195:Signal diagram timing relay, single pulse 1

1: Trigger coil T…EN

2: Stop coil T…ST

3: Reset coil T…RE

4: Switching contact (N/O contact) T…Q1

- Range A: The trigger signal is short and is lengthened.
- Range B: The trigger signal is longer than the set time.
- Range C: The stop coil interrupts the timing out of the set time.

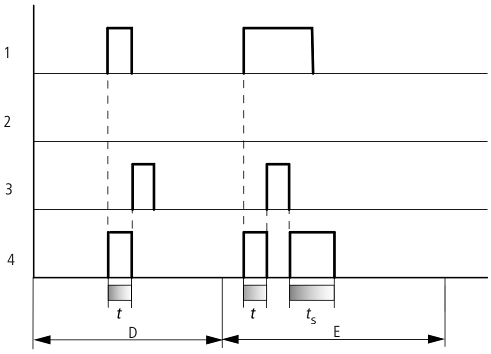


Figure 196:Operational diagram timing relay, pulse shaping 2

- Range D: The reset coil resets the timing relay.
- Range E: The reset coil resets the timing relay.
  The trigger coil is still energized after the reset coil is disconnected, whilst the delay time runs down.

Timing relay, synchronous and asynchronous flashing

Time value I1: mark time
Time value I2: space time

Synchronous (symmetrical) flashing: I1 equal I2
Asynchronous flashing: I1 not equal I2

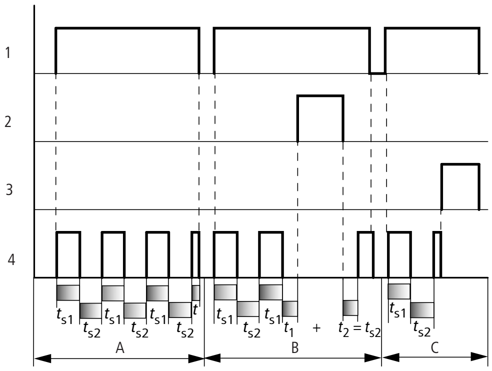


Figure 197:Signal diagram: timing relay, synchronous and asynchronous flashing

1: Trigger coil T...EN

2: Stop coil T...ST

3: Reset coil T...RE

4: Switching contact (N/O 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.

5.33 TB, table function

This function block is available for easy800-SWD from device version 01 and for easy800 standard devices from device version 07.

The table function block enables you to create and read table entries in the form of double words (32-bit). The function block works on the memory stack principle, whereby you can choose to apply the FIFO (first-in-first-out) or the LIFO (last-in-first-out) principle for read operations. A table can contain up to 16 double words.

easy800 device allows the use of up to 32 table function blocks.

Contacts and Coils

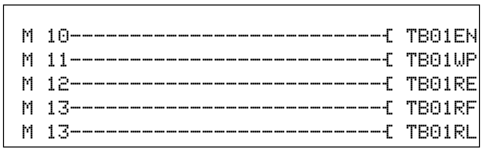
Contacts	Meaning
TB..TE	Table is empty
TB..TF	Table is full

Coils	Function
TB..EN	Selective activation of the function block
TB..WP	A rising edge at the TB..WP coil transfers the value at input TB..I1 into the table and outputs it at output QV.
TB..RF	Clears the value entered first from the table and outputs it at output QV.
TB..RL	Clears the most recently entered value from the table and outputs it at output QV.
TB..RE	Clears the table content.

The data at function block output QV depends on which coil was activated

- TB..WP: the currently read input value.
- TB..RF: the first table entry.
- TB..RL: the last table entry.

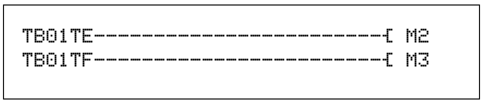
Example of a table function block:



```
M 10---------------------------[ TB01EN
M 11---------------------------[ TB01WP
M 12---------------------------[ TB01RE
M 13---------------------------[ TB01RF
M 13---------------------------[ TB01RL
```

Figure 198:Wiring the function block coils

Markers switch the coils of the function block.



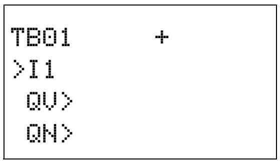
```
TB01TE-------------------------[ M2
TB01TF-------------------------[ M3
```

Figure 199:Wiring the function block contacts

Function block contacts TE (empty) and TF (full) display an empty or full table respectively.

Operands are assigned to inputs and outputs in the parameter display.

Parametric programming of the function block inputs/outputs



```
TB01     +
>I1
 QV>
 QN>
```

When using the function block in the circuit diagram for the first time, use OK to automatically enter the general display of function block parameters, as shown in the figure on the left. This is where you make the function block settings, in this case, the allocation of the inputs and outputs to the operands. The display contains the following elements:

TB01	Function block: Table function, number 01
+	Parameter set can be called via the PARAMETERS menu
>I1	Table input value: -2147483648 - 2147483647
QV>	Actual, first or last table entry as output value: -2147483648 - 2147483647
>QN	Current number of table entries: 0 - 16

Input

I1: The value at input I1 is read into the table when function block coil TB..WP is activated. The function block input I1 can be assigned the following operands:

- Constant.
- Markers MD, MW, MB.
- Analog inputs IA01 to IA04:
  - IA01: Terminal I7.
  - IA02: Terminal I8.
  - IA03: Terminal I11.
  - IA04: Terminal I12.
- Analog output QA01.
- Output ...QV of a different function block.

Outputs

QV: The first and last table entry is output at output QV, depending on the function block coils TB..RF and TB..RL.

QN: The output QN indicates the current number of table entries.

You can assign the following operands to the function block outputs TB..QV and TB..QN:

- Analog output QA01.
- Markers MD, MW, MB.

Function of the function block

→ To allow function block table function to work, it must be enabled through an active coil TB..EN. If the coil is not active, the entire function block will be disabled.

Up to 16 table fields can be overwritten and read.

Memory	112 + per 4 Byte/Constant at function block input	
Cycle time [µs]	easy800 standard	easy800-SWD
	No entry	12

Entering a value in the table

When the function block is active, the current value at the double word input at TB..I1 is entered in the table every time there is a rising edge at the WP coil. Coils TB..EN and TB..WP can be actuated simultaneously with a rising edge.

Each new table entry is entered behind the previous one until the sixteenth entry is reached. Immediately after the value of TB..I1 is transferred, this value is output TB..QV. Output TB..QN is incremented by 1 and then supplies the current number of table entries.

If the maximum number of 16 table entries is reached, no more data is transferred to the table and signal contact TB..TF closes. If you wish to make more table entries, read values from the table beforehand or clear the entire table with a rising edge at the RE coil.

A rising edge at the RE coil causes:

- The value at output TB..QN to be set to 0.
- The signal contact TB..TE to close.
- The signal contact TB..TF to open.

Reading out the table

The TB..RF and TB..RL coils are used to control whether you are reading out from the start or end of the table.

A rising edge at the RF coil (Read First) removes the first table entry (FIFO function) and outputs it at the output TB..QV. A rising edge at the RL coil (Read Last) removes the last table entry (LIFO function) and outputs it at TB..QV.

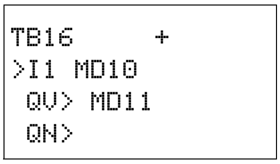
With each output of a table entry the signal at output TB..QN is reduced by 1.

Example

An operator determines the work required for a particular workpiece, creates a production code for it and writes it in a table.

The workpieces reach the machining stations in this order. When the workpiece is changed, the stations read the required production steps from the table.

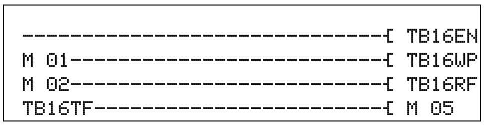
As soon as the table has reached 16 entries, the function block disables entry by the operator.



```
TB16      +
>I1 MD10
 QV> MD11
 QN>
```

The figure on the left shows the parameter display and the parameter set for the function block TB16. Once function block number 16 has been defined, you set here the operands for the production code at input TB16I1 and the output TB16QV.

The coils and contacts are programmed in the circuit diagram of easy800:



```
------------------------------[ TB16EN
M 01--------------------------[ TB16WP
M 02--------------------------[ TB16RF
TB16TF------------------------[ M 05
```

Figure 200:Table function example

- The enable coil TB16EN is permanently active, the function block is not switched off.
- With a rising edge of marker M01 write clock pulse coil TB16WP for reading the production code from MD10 and transfers it to the table.
- The contact TB16TF is used to show the sixteenth entry via the marker M 05 to the operator.
- With a rising edge from marker M02 to TB16RF the processing station reads out the first production code from the table and supplies it at output TB16QV.
- After the sixteenth read process or less, all production codes have been read out and the table is empty. Contact TB16TE closes.

5.34 VC, value limitation

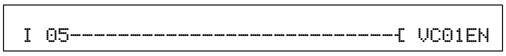
This function block is available for easy800-SWD from device version 01 and for easy800 standard devices from device version 04.

The Value limitation function block enables you to output values within defined limit values. A easy800 device allows the use of up to 32 value limiters.

Contacts and Coils

Coils	Function
VC..EN	Selective activation of the function block

Example of the value limitation function block:

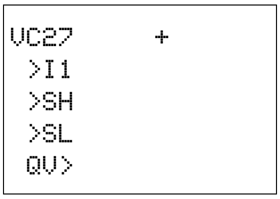


```
I 05---------------------------[ VC01EN
```

Figure 201:Wiring the function block coil

The enable coil of the function block is connected directly to a device terminal.

Parametric programming of the function block inputs/outputs



```
VC27     +
 >I1
 >SH
 >SL
 QV>
```

When using the function block in the circuit diagram for the first time, use OK to automatically enter the general display of function block parameters, as shown in the figure on the left. Enter the function block settings here. The display contains the following elements:

VC27	Function block: Value limiter, number 27
+	Parameter set can be called via the PARAMETERS menu
>I1	Input value: -2147483648 - 2147483647
>SH	Upper setpoint value: -2147483648 - 2147483647
>SL	Lower setpoint value: -2147483648 - 2147483647
QV>	Output value within the set limits: -2147483648 - 2147483647

Inputs

I1: You assign the value to be output within the limit values at the function block input I1.

SH and SL: The upper and lower limit setpoint values are set at function block inputs SH and SL respectively.

You can assign the following operands to function block inputs I1, SH and SL:

- Constant
- Markers MD, MW, MB
- Analog inputs IA01 to IA04:
  - IA01: Terminal I7
  - IA02: Terminal I8
  - IA03: Terminal I11
  - IA04: Terminal I12

- Analog output QA01
- Output ...QV> of a different function block

Output

The output VC..QV provides the input value within the defined limits as an integer. You can assign the following operands to output ...QV:

- Analog output QA01
- Markers MD, MW, MB

Function of the 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 appropriate limit value is output at VC…QV.

Memory	40 + per 4 Byte/Constant at function block input	
Cycle time [µs]	easy800 standard	easy800-SWD
	34	9

5.35 Example with timing relay and counter function block

A warning light flashes when the counter reaches 10. In this example the function blocks C01 and T01 are wired in the standard circuit diagram and their inputs/outputs are defined.

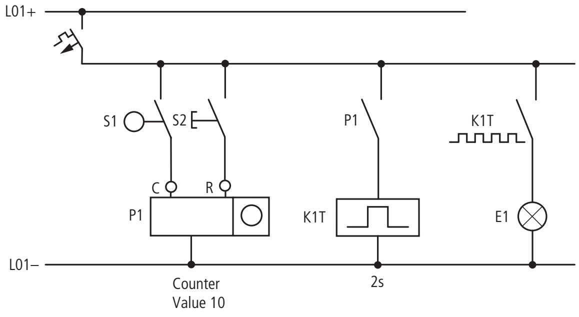


Figure 202:Hardwiring with relays

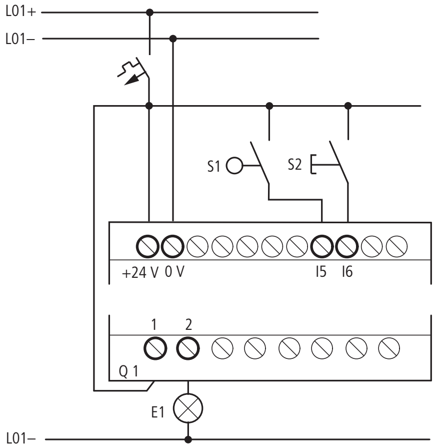
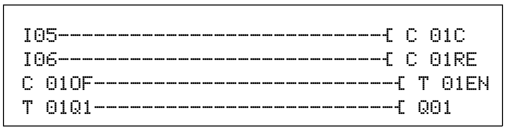


Figure 203:Wiring with easy800

Enter circuit diagram

▶ Enter the following circuit diagram.



```
I05----------------------------[ C 01C
I06----------------------------[ C 01RE
C 01OF-------------------------[ T 01EN
T 01Q1-------------------------[ Q01
```

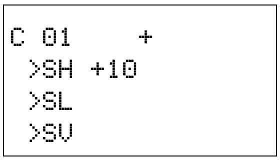
Figure 204:Wiring of counter and timing relay

Entering function block parameters

If you enter the coils or contacts of a function block, the inputs/outputs of the function block are displayed that you can parameterize. The parameters can be entered via the function blocks menu.

The meaning of these parameters is explained under each function block type.

Entry:



```
C 01    +
 >SH +10
 >SL
 >SV
```

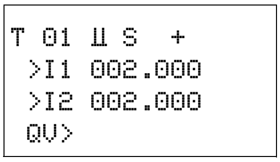
The first part of the parameter set of a counter is displayed.

▶ Move the cursor > over the "+" character to the value entry 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 ^v to modify the value of the digit.

▶ Press OK to save the value and ESC to return to the circuit diagram.

▶ Set the parameters for T 01.



```
T 01 ll S  +
 >I1 002.000
 >I2 002.000
 QV>
```

The timing relay works like a flashing relay. The function is set on the top right beside the number in the parameter display.

The time base is set to the right of the “flashing” function. Leave the time base set to S for seconds.

▶ Move the cursor to the right over the + character in order to input the time setpoint value I1.

If you enter the same setpoint value at I1 and I2, the timing relay operates as a synchronous flasher.

The + character means that the parameters of this timing relay can be modified using the PARAMETERS menu.

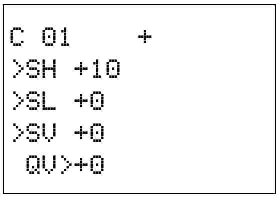
▶ Confirm the value input with OK.
▶ Press ESC to leave entry.

Testing the circuit diagram:

▶ Switch easy800 to RUN operating mode and return to the program.

You can display every parameter set via the function relays menu.

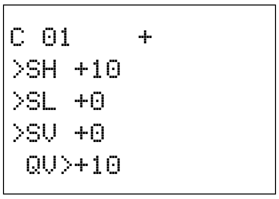
▶ Move the cursor onto C 01 and press OK.



```
C 01    +
>SH +10
>SL +0
>SV +0
 QV>+0
```

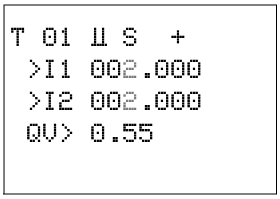
The parameter set for the counter is displayed with actual and setpoint values.

▶ Move the cursor ⌵ downwards until you see the value QV.
▶ Switch input I05. The actual value changes.



```
C 01    +
>SH +10
>SL +0
>SV +0
 QV>+10
```

If the actual and upper setpoint values of the counter are the same, the timing relay switches the warning light on and off every 2 seconds.



```
T 01 ll S  +
 >I1 002.000
 >I2 002.000
 QV> 0.55
```

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.

If the setpoint value is a constant, it can also be modified via the PARAMETERS menu.

→ The actual value is only shown in RUN mode.

6 The network easyNet

6.1 Introduction network easyNet

The easyNet network, termed the NET in the following document, is based on the CAN (Controller Area Network) serial bus system. CAN complies with the ISO 11898 standard.

The NET is designed for up to 8 stations.

Using the 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.

All easy800 devices apart from EASY802-DC-SWD can be NET stations.

→ Distribute the tasks over several devices that exchange the data between each other via NET: easy800 or MFD devices for open-loop and closed-loop control and MFD devices for display and HMI tasks.

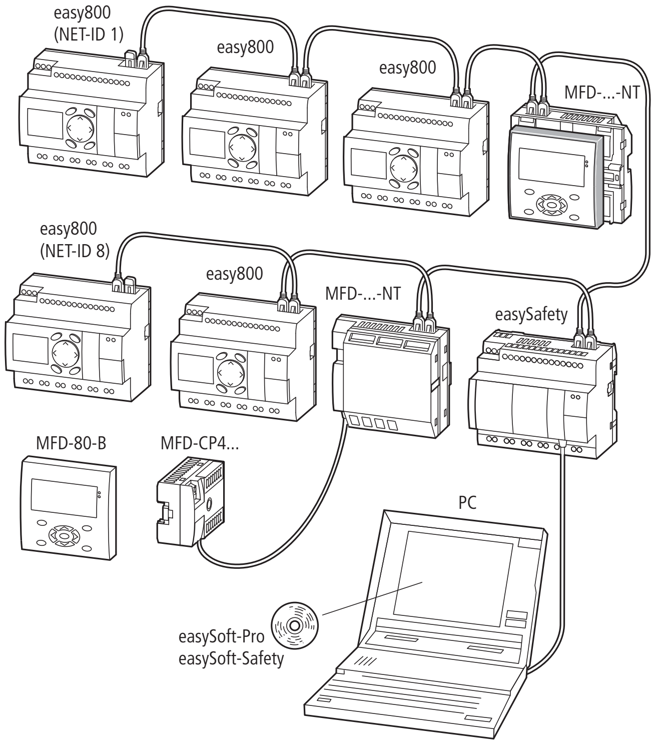


Figure 205:Mixed operation of NET stations

Only the physically first NET station that is located at the beginning of the network manages the configuration and commissioning of the NET. In this way, the commissioning of NET stations is very simple. Apart from the installation with prefabricated cables, this only requires the following for the standard "passing through" cable routing:

- Definition of the first NET station by assigning NET ID 1 and the possible necessary modification of the baud rate in the NET PARAMETERS menu.
- Creation of the station list with automatic assignment of the NET IDs for the connected devices in the STATION menu.
- Configuration via the CONFIGURATION menu.

Further information on the configuration of the NET can be found on Page 94.

Commissioning is even simpler when using easySoft-Pro. Information on this is provided in the Online Help of easySoft-Pro.

⚠ **DANGER**

Ensure that the automatic starting of easy800 will not cause the unwanted starting of machines and plants.

Transfer behavior of the NET stations

In addition to the process that runs the program, a device with a NET connection also runs a parallel communication process that controls the exchange of network data.

Before every program cycle the network data is written from the NET receive buffer to the process image of the program. The program runs through the next cycle with this data. At the end of the program cycle, the updated network data is written from the process image to the NET send buffer and then transferred.

Reading and sending the data via the NET

The communication process of a NET station reads every message on the NET and checks whether any messages are meant for its own address (NET ID). If yes, they are transferred to a receive memory.

If the content of the send memory changes, a message is sent provided that no other message is being transferred on the NET at this time. The NET ensures that each station can send its messages. This means that each station must maintain a delay time between the sending of messages.

The pause time increases the higher the number of stations and the lower baud rate setting.

The number of NET stations present is recognized by every station by means of a "sign of life" signal.

→ The following applies to a 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 transferring (download/upload) a circuit diagram while the NET stations are in RUN mode.

Functions of the NET stations

The different types of NET stations vary according to their technical overview:

- easy800 control relay,
- MFD-...-NT operating and visualization devices
- Safety control relay (ES4-…)
- SmartWire gateways (without program, with configuration)
- Control systems EC4-200 and
- XC200 controllers

6.2 Net stations with or without program

NET stations can be run with or without a program. Whether a program is present is indicated by the REMOTE-IO NET parameter. With EASY806-DC-SWD: REMOTE IO is always deactivated.

The NET station with NET-ID 1 must always be run with a program.
The NET parameter REMOTE-IO is automatically deactivated.

- NET stations with their own program.
  - The NET parameter REMOTE-IO of this NET station is automatically deactivated.
  - Possible NET IDs: 01-08
  - Possible devices: All
- NET stations without their own program
  - Used as input/output device in REMOTE IO mode
  - The NET parameter REMOTE-IO of this NET station is automatically activated.
  - Possible NET IDs: 02-08
  - Possible devices: easy800 and MFD

Possible combinations of NET stations

- The NET station with NET-ID 1 has its own program and all other NET stations are run in REMOTE IO mode without their own program.
- All NET stations have their own program. REMOTE IO is deactivated on all.
- The NET station with NET-ID 1 has its own program and all other NET stations are run mixed: some with their own program and some without.

Access options for NET stations

All NET stations with a program can read all inputs and outputs of the other NET stations.

Only the station with NET-ID 1 can write to all the outputs of the NET stations without a program.

A NET station with a NET-ID ≠ 1 cannot write to the outputs of other NET stations. This is irrespective of whether the other NET stations are run in REMOTE IO mode or not.

6.3 Description of the NET-PARAMETERS

The NET parameters described below can be found in the NET PARAMETERS menu:

▶ Simultaneously press the DEL and ALT buttons with the status display active. Then choose CONFIGURATOR -> NET -> NET PARAMETERS and press the OK button.

Station number (Station Menu NET-ID)

Default settings NET ID = 00.

Selectable, valid NET-IDs 01 - 08.

The NET ID of a device is used to give it a unique identification on the NET. A NET ID can therefore only be assigned once within a network.

The default setting NET ID 00 prevents the possibility of double addressing when an existing NET station (NET ID 1 - 8) has to be replaced.

Data transfer speed (Station Menu BAUD RATE)

Factory set baud rate = 125 Kbaud.

The devices allow a NET transfer speed of 10 - 1000 Kbaud, selectable in defined steps. Possible baud rates are: 10, 20, 50, 125, 250, 500 and 1000 KB.

▶ Use ^ and ⌄ to select a baud rate of 10 - 1000 Kbit/s and confirm your selection with OK.

→ All NET stations must have an identical transfer speed set. This is achieved automatically if you carry out the NET configuration via NET station 1 (→ section “Configuring NET”, page 94).

The maximum transmission speed depends on the total length of the network cables (a Technical data, → Section 12.3.7, “NET easynetwork“ on Page 359).

→ The reaction speed of devices connected via the NET depends essentially on the baud rate, the delay time and the volume of transferred data.
Therefore always set the maximum transfer speed. The factory set baud rate depends on the total length of the network cables of 125 m.
A maximum baud rate is therefore very important if an MFD is required to work on the NET since additional NET transfer time is needed.

Pause time, changing the write repetition rate manually (Station Menu BUSDELAY)

Default setting: Bus delay time = 0 (recommended setting that you should only change for the reasons stated below).

In normal operation, the NET stations organize their data exchange independently. The NET stations also determine the minimum delay time required so that all messages can be sent.

Use BUS DELAY to change a multiplier that increases the bus delay time [tp] between the transmission cycles of user data in the NET.

- A value 1 means that the bus delay time is doubled.
- A value 15 means that the bus delay time is increased tenfold.

→ All NET stations must have an identical bus delay time set. This is achieved automatically if you carry out the NET configuration via NET station 1 (→ section "Configuring NET", page 94).

▶ If necessary, use ⌃ and ⌄ to set a multiplier of 1 - 15 and confirm your selection with OK.

$t_{pnew} = t_p \times (1 + n)$
t_{pnew} = new bus delay time
t_p = bus delay time determined by the network
n = Value at BUS DELAY

Even if a maximum baud rate is set, it may be necessary to increase the bus delay time for the following reasons:

- In addition to the user data, additional data is transferred to the PC, i.e.to easySoft-Pro in Online status or to an MFD in terminal mode.
- The transfer rate is low.

By increasing the bus delay time you increase the time window for online services such as the power flow display, status display of actual values of function block operands and the forcing of markers.

An increase of the bus delay time is only really necessary in the following conditions:

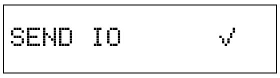
- You use an MFD in terminal mode via the NET and see the message "Connection establishment in progress...". In this case the bus delay time must be increased permanently.
- You wish to use the online services of the easySoft-Pro programming software via the NET. In Communication View, however, you will receive the error message "Device not responding". This fault message appears if you wish to establish an online connection or are already online. In this case, a temporary increase of the bus delay time is useful. It should be reduced after commissioning.

Remember also that the easySoft-Pro programming software displays status changes more quickly on the device if you select the maximum PC baud rate.

An increase in the bus pause time means that fewer messages (inputs, outputs, bit data, word data) are transferred per time unit.
The reaction speed of all devices connected via the NET depends essentially on the baud rate, the delay time and the volume of transferred data.
You should therefore retain the proposed bus delay of the multiplier 0!

Send each change on the inputs/outputs (Station Menu SEND IO)

Default setting SEND IO = On



```
SEND IO       ✓
```

Activate the SEND IO if you require short response times to each status change on the input or output of a NET station (e.g. at 2I 02, 8Q 01, etc.).

Activate the SEND IO function if all NET stations are to be notified immediately of a status change at the input or output of a NET station (e.g. at 2I 02, 8Q 01, etc.).

In addition to the cyclical data transfer, the status change will be sent to all other NET stations in a separate telegram.

NET stations without a circuit diagram send each status change of an input or output, without taking the SEND IO setting into account.

An activated SEND IO function will considerably increase the number of telegrams on the NET and can slow down the transfer of other user data.

If you are using a high-speed counter (e.g. CH01) with constantly changing input data in the circuit diagram of a NET station, SEND IO should be deactivated for this station.
Otherwise the station changes at the counter input will unnecessarily increase the number of telegrams on the NET.
Transfer these status changes with SN/RN operands.

Automatic change of the RUN and STOP mode (Station Menu REMOTE RUN)

Default settings REMOTE RUN = On

REMOTE RUN must be activated on NET stations with NET ID 2 - 8 which are required to automatically follow the operating mode change of the NET station with NET ID 1.

As the station with NET-ID 1 cannot be run in REMOTE RUN, the menu item is displayed but cannot be selected.

NET stations with a circuit diagram and REMOTE RUN mode only follow the operating mode change if the device does not expect any operator entry. This is the case if the display shows the Status display or a display text. If it has an active menu, the device expects an operator entry. This condition obviously only applies to devices with a keypad.
If you make an entry on a device in REMOTE RUN mode and then return to the Status display, you must carry out the change to RUN mode manually. This also applies if the NET station with NET-ID 1 is already in RUN mode. The operating mode change does not therefore automatically occur at a later time.

DANGER

Ensure that the automatic starting of easy800 will not cause the unwanted starting of machines and plants.

→ Note: If EASY806-DC-SWD is the NET station with NET-ID 1, the NET stations are also possibly switched to STOP if there is a fault in the SWD line. This is the case if the “STOP on SWD fault” device parameter is activated on EASY806-DC-SWD and REMOTE RUN on the NET stations.

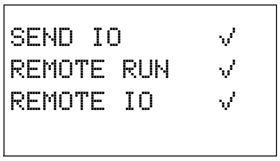
Configuring an input/output device without its own circuit diagram

REMOTE IO = OnREMOTE IO is only indicated in the NET-PARAMETERS menu. As the devices are factory set without a program, the REMOTE IO display is checked with a tick. The conventional NET stations with NET-ID 2 - 8 without their own program function as input/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 have to assign the NET-ID, baud rate and bus delay to this type of device.

As soon as you enter a circuit diagram on this kind of NET station with NET-ID 2 - 8, the REMOTE IO function is automatically deactivated. This prevents write access to the outputs of this kind of NET station, both locally from its own circuit diagram as well as from REMOTE station 1.

With EASY806-DC-SWD: REMOTE IO is always deactivated.

The default settings for the input and output devices are:



```
SEND IO      √
REMOTE RUN   √
REMOTE IO    √
```

6.4 Network transmission security

easyNet is a CAN-based network with the same error detection capability as 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 using a network with:

- 500 Kbaud
- average bus load 25 %
- average operating time 2000 h/year
- average error rate of 10^{-3}, i.e.: 1 message is faulty every 1000
- transmission of $1.12 - 10^{10}$ messages per year of which $1.12 - 10^{7}$ messages are faulty
- Residual error probability: $r < 10^{-10} - 10^{-3} = 10^{-13}$

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.

7 SmartWire-DT

SmartWire-DT can be run with easy800-SWD. This is not possible with easy800 standard.

This chapter gives you an overview of SmartWire-DT. Detailed descriptions are provided in the following manuals:

- SmartWire-DT system (MN05006002Z-EN)
- SmartWire-DT station (MN05006001Z-EN)

With SmartWire-DT you can interconnect up to 99 stations very quickly in simple steps. Stations can consist of switching devices as well as I/O components.

The ribbon cable with the SWD stations and all components required for connecting them is called the SWD line. The SWD line starts at the blade terminal as a possible connection to the control relay and then ends at the bus termination.

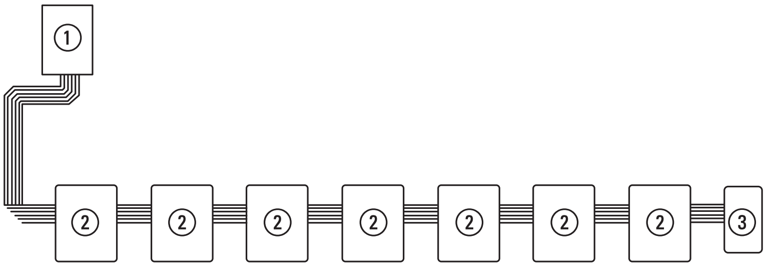


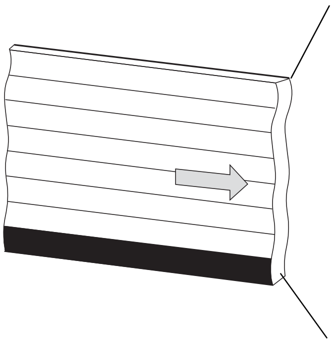
Figure 206:easy800-SWD with SWD string

① easy800-SWD
② SWD station
③ Bus termination

Cable layout

An 8-core connection cable is used by easy800-SWD to feed the SWD stations with power and exchange data. SmartWire-DT is self-configuring and addresses its stations at the push of a button.

Table 18: Configuration of the SmartWire-DT flat band conductor

	Meaning	
	+24 V DC	Contactor control voltage
	Common	Contactor control voltage
	GND	for device supply voltage and data cable
	Data B	Data cable B
	Data A	Data cable A
	GND	for device supply voltage and data (data A, data B)
	SEL	Select cable for automatic addressing of the SmartWire-DT stations
	+15 V DC	Device supply voltage

The power supply for easy800-SWD and the SWD stations is connected via the POW terminals. The 15 V power supply of the SWD stations is generated from POW.

If you wish to use contactors, circuit-breakers or motor starters in the installation with SmartWire-DT, a 24 V DC voltage must be fed in at AUX.

Definition of terms: components - elements - stations

The SWD components consist of SWD stations and SWD elements.

SWD stations are all those SWD components that respond to a request from the easy800-SWD, i.e. can exchange data. They are assigned an SWD station address.

SWD elements are passive SWD components that do not exchange data but are required for operating the SmartWire-DT network. SWD elements are, for example, the power feeder modules, SWD cables, adapters etc.
SWD elements are not given an SWD station address.

SWD Universal module

If functions have to be retrofitted on an installation, modifications in the hardware configuration and program extension are normally very labor-intensive. The use of a universal station enables you to initially replace the SWD stations that are required at a later time. If the installation is then extended later, the planned stations are then installed instead of the universal modules. In this way, you will be able to retain all station addresses. A detailed description on connecting the universal modules is provided in the “SmartWire-DT Station” manual (MN05006001Z-EN).

Power feeder module

The number and the types of SWD stations determine the volume of data to be transferred and the electrical load.

You can use power feed modules if your application has a higher power consumption than the device provides, or if the cable length of the SWD line causes an increased voltage drop. Calculation examples are provided in the "SmartWire-DT - System" manual (MN05006002Z-EN).

Power feeder modules are power supply units that are connected additionally to the SmartWire-DT network when there is an increased current consumption or increased voltage drop. Being a passive SWD element, a power feeder module does not have a station address.

The EU5C-SWD-PF1 power feed module feeds the 24 V DC supply voltage for contactors on the SWD line.

The EU5C-SWD-PF2 power feed module feeds both the 24 V DC supply voltage for contactors as well as the 15 V DC supply voltage for other SWD stations on the SWD line.

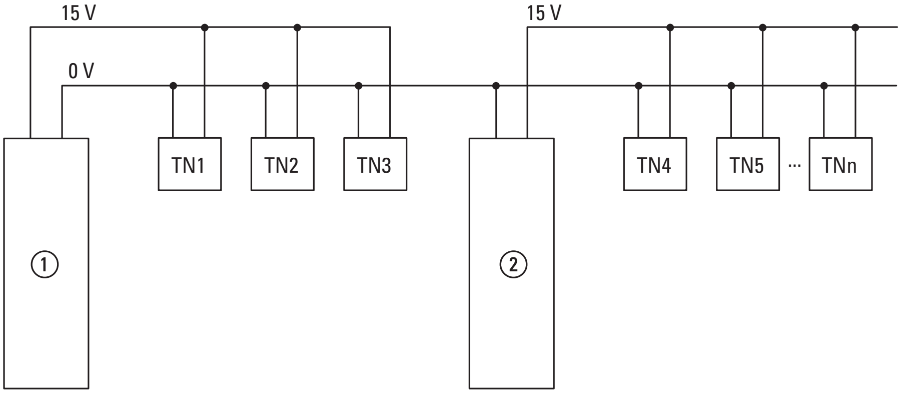


Figure 207:Schematic layout of an SWD line with the EU5C-SWD-PF2 power feed module

① EASY806-DC-SWD

② EU5C-SWD-PF2

7.1 Configurations

The following different SWD configurations are possible:

1. Project configuration
   Before easy800-SWD can be put into operation, the device must be notified which SWD stations are present and how their parameters are set. It must be clear which inputs and outputs of the SWD stations are assigned to which operands in the circuit diagram. This information is collected in the project configuration and stored with the program in easy800-SWD. You create the project configuration using the easySoft-Pro software.
2. Target configuration
   Pressing the Config configuration button on the device determines the actual configuration and stores it as the target configuration in easy800-SWD.
3. Actual configuration
   By this is meant the actual physical configuration present on the SWD line. Every time easy800-SWD is started, the actual configuration is determined and compared with the stored target configuration.

To operate the device, the target configuration and the project configuration must be stored. Both must be identical. The device will only startup up error-free if the actual configuration at the start matches the target and project configuration.

Creating the project configuration

The easySoft-Pro software is required for the project configuration. Information on this is provided in the Online Help of easySoft-Pro. Changes to the program can then be made directly on the device without any need of the software. This requires that display and operating elements are connected to the easy800-SWD.

Change to the configuration

To change the configuration of the SWD line, such as by adding a new station, you must change the project configuration and determine the target configuration again. Proceed as described in → section “Putting the SmartWire-DT into operation”, page 98.

7.2 Cyclical input and output data of the SWD stations

All input and output data of the SWD stations present are stored in a data storage area of the easy800-SWD, the so-called process image. The maximum input data is 800 bytes and the maximum output data 642 bytes. In all the data area must not exceed 1000 bytes. The cyclical data exchange between the easy800-SWD and the SWD stations is implemented with a common data telegram. The input data can consist of several bytes, depending on the station. The input data contains information such as the switch position or diagnostic information.

The input and output data of the stations is assigned to the bit operands or the value operands by means of an assignment list (see easySoft-Pro Online Help). The operands can be wired like inputs and outputs in the circuit diagram (see → section "Wiring", page 80).

Overview of easy800-SWD bit operands

Bit operand	Function
I1 - I4[1]	Local device inputs
I5 – I12	functionless
I13	SWD line group alarm
I14	always set to 1
I15	functionless
I16[1]	group fault alarm for short-circuit on Q1 or Q2.
I17 – I99	can be assigned to input data in the assignment table
Q1/Q2[1]	local device outputs
Q3 – Q16	functionless
Q17 – Q99	can be assigned to output data in the assignment table

[1] only EASY806-DC-SWD

→ Bit operand I14 on easy800-SWD is always set to 1. easy800 standard shows the status of the expansion device via I14. easy800-SWD does not know any expansion devices and therefore has no function.

Overview of value operands

- Marker byte MB1 to MB128
- Marker word MW1 to MW128
- Marker double word MD1 to MD128.

Diagnostics possibilities

easy800-SWD has the following diagnostic alarms:

- I13 - group alarm;
  is 1 if a mandatory station has failed on the SWD line or if easy800-SWD finds an "incorrect" station on the SWD line.
- Cyclic diagnostic information
- Extended diagnostics information

Cyclic diagnostic information

SmartWire-DT provides you with cyclical diagnostic information. This information can be evaluated in the circuit diagram and used for further control tasks.

This basic diagnostic information is coded in the normal process image of each SWD station. The information about whether the SWD station is exchanging data normally or whether diagnostic alarms are currently present is provided on all SWD stations in the first input byte (byte 0) at bit positions 4 and 6.

DANGER

Persons, systems and machines may be put at risk if an N/C contact is misinterpreted. When using N/C contacts always evaluate the PRSNT and DIAG diagnostic bits of this station.

Example of input and diagnostic data

The M22-SWD-K11-LED-R function element has a red LED for illuminated pushbuttons and the two switch states 0 and 1. With cyclical input data of one byte, it contains information on switch position and diagnostic information.

Byte 0:

7	6	5	4	3	2	1	0
Diagnostics data				**Input data**			
SUBST	PRSNT	–	DIAG	–	–	N01	NC1

Bit	Designation	Meaning
0	NC1 = Normally Closed	0: contact actuated 1: Contact not actuated
1	NO1 = Normally Open	0: Contact not actuated 1: contact actuated
2	not used	–
3	not used	–
4	DIAG	0: No diagnostic alarm 1: Diagnostic alarm present
5	not used	–
6	PRSNT	0: station not present 1: Station present
7	SUBST	0: planned station is present 1: M22-SWD-NOP(C) universal module present

The diagnostic bits of the individual SWD stations can be looked up in the “SmartWire-DT Stations” manual.

Extended diagnostic alarms

If there is a diagnostic alarm present for an SWD station, this means that there is a fault on this station. Using the easySoft-Pro software you can display the extended diagnostic alarms to identify the faults more closely. The extended diagnostic alarms cannot be evaluated and processed in the circuit diagram.

Proceed as follows for easySoft-Pro to display the extended diagnostic alarms:

▶ Click the Communication button in easySoft-Pro.
▶ Establish a connection to easy800-SWD by choosing Connection -> Online.
▶ In the Properties field choose the Diagnostics buffer tab.

The table in the Properties field shows you the extended diagnostic alarms. The alarms are logged and displayed with a time stamp. The time stamp is based on the system time of your PC. The code is displayed in decimal notation. Diagnostic alarms can be categorized as follows:

- Code < 1000:
  Alarm comes directly from the station; the code is the same as an error on fieldbus systems. These error messages are described in the "SmartWire-DT Station" manual. The error codes stated there are in hexadecimal notation.
- Code ≧ 1000:
  Alarm was generated by easy800-SWD as the SWD station can no longer send.

The extended diagnostic alarms are not retentive. If easy800-SWD is switched off, the diagnostics buffer is cleared.

Further information on extended diagnostics is provided in the Online Help of easySoft-Pro.

7.3 SWD device parameters

All device parameters can only be set via easySoft-Pro.

The device parameter “STOP on SWD error” enables you to define whether easy800-SWD continues to process its circuit diagram in the event of an error or switches to STOP mode. The POW LED also indicates the current mode of easy800-SWD. A green flashing POW-LED indicates RUN mode, a continuous green light indicates STOP mode. You can set the “STOP on error” device parameter for easy800-SWD using the easySoft-Pro software.

A mandatory station is one with the “Mandatory station” device parameter activated.

An “incorrect” station is one that easy800-SWD finds which, when determining the actual configuration, is not expected in the target configuration. Bear in mind that the serial number of the station is also examined for matching as well as the station type. This means that if you replace a pushbutton actuator with one of the same type when the SWD line is switched off, it will be identified as an “incorrect” station by easy800-SWD.

The “All devices are optional” and “Mandatory station” device parameters determine whether an SWD error is present if the actual configuration is different to the target configuration.

The “Compatible devices permissible” and “Replaceable with universal module” device parameters determine whether an SWD error is present if the target configuration is different to the project configuration.

The following table provides you with an overview:

Table 19: Summary of SWD device parameters

Device parameter definable on		SmartWire-DT reaction with option activated
easy800-SWD		
Stop on SWD error	√	easy800-SWD behaves as follows in the event of SWD error and • with option activated: easy800-SWD switches to STOP mode. • with option deactivated: easy800-SWD stays in RUN mode, the circuit diagram continues running.
All devices are optional	√	The activated option causes the “Mandatory station” option adjustable on the station to be no longer taken into account. The SWD line can be run with any number of failed SWD stations. A missing SWD station causes • the PRSNT diagnostic bit to be set to 0 • the DIAG diagnostic bit to be set to 1 • with the option activated, that no SWD error is reported, • with the option deactivated, that easy800-SWD checks the set “Mandatory station” device parameter.
SWD station		
Compatible devices permissible	√	If easy800-SWD checks this option, it has already determined that an “incorrect” SWD station is present and that the station is not a universal module. easy800-SWD checks the option whether compatible devices are permissible. • With the activated option, easy800-SWD checks whether the station found is compatible with the station expected by the target configuration. ◦ If this is the case, no SWD error is reported. ◦ If the station found is not compatible, an SWD error is reported. • with the option deactivated, the SWD error is reported Compatible stations are shown in → Table 12.6Page 363.
Required station	√	This device parameter is only evaluated if the “All devices are optional” parameter is not clicked on easy800-SWD. A missing SWD station causes • with the option activated, an SWD error to be reported. • with the option deactivated, no SWD error to be reported
Replaceable with universalmodule	√	An “incorrect” SWD station causes easy800-SWD to check whether the SWD station present is a universal module. If this is the case, the SUBST diagnostic bit is set to 1. Furthermore • with the option activated, no SWD error is reported. • with the option deactivated, an SWD error is reported, easy800-SWD switches to STOP mode and the data exchange is interrupted. If the SWD station present is not a universal module, easy800-SWD checks whether the “Compatible devices permissible” option is activated.

Application cases for the effective use of the device parameters

Stop on SWD error, deactivated:

- You leave this option deactivated if you wish to run an easy800-SWD only with the local inputs I1 to I4 and without an SWD line.
- You leave this option deactivated if you wish to address a local output such as an alarm, despite an SWD error.

Compatible devices permissible, activated:

- The installation was already in operation and the target and project configuration are on the easy800-SWD. easy800 is switched off in order to replace a faulty station. As you don't have a replacement of the same type, you use a compatible type according to the table.
  If you switch on easy800-SWD, the actual configuration is determined. The device compares the actual configuration with the target configuration and determines that they are different. You then have to press the Config configuration button. In response, easy800-SWD reads the new target configuration. easy800-SWD compares the target configuration with the project configuration and determines they are different. The device checks whether the "Compatible devices permissible" option was selected. If this is the case, there is no SWD error and easy800-SWD starts to run.

Can be replaced with universal module, activated:

- You are configuring an installation and wish to first use a universal module for a particular station. The actual station is not to be installed until a later expansion stage. You activate for this the "Replaceable with universal module" device parameter. This enables you to replace the station of the project configuration with a universal module in the target configuration. After you have installed the universal module on the de-energized SWD line, you switch on easy800-SWD and press the Config configuration button. The target configuration is read in. The comparison with the project configuration determines that they do not match. easy800-SWD checks whether the device parameter was activated for this station. If this is the case, easy800-SWD starts up, otherwise an SWD error is reported.

7.4 Errors on the SWD line

Different events can occur in conjunction with the SWD line, which require rectification.

No SWD line present

If there is no connection between the SWD line and easy800-SWD, or if the actual configuration has not found any station at all, the "STOP on SWD error" device parameter determines how easy800-SWD is to respond:

- "STOP on SWD error" is activated: easy800-SWD switches to STOP. The POW LED shows continuous green, Config LED shows continuous green; SWD LED shows continuous red.
- "STOP on SWD error" is deactivated: easy800-SWD stays in RUN mode. The POW LED flashes green, Config LED shows continuous green; SWD LED shows continuous red.

Missing station: When is this an SWD error?

If easy800-SWD determines that an SWD station of the target configuration is missing in the actual configuration, the PRSNT diagnostic bit is set to 0 and the group alarm I13 is set to 1.

If the "All devices optional" device parameter has been clicked on easy800-SWD, missing stations do not trigger an SWD error. The "Mandatory station" device parameter, which can be set individually for each SWD station, is no longer checked in this case. easy800-SWD stays in RUN mode. The POW LED flashes green, Config LED shows continuous green; SWD LED shows continuous green.

If the "All devices optional" device parameter has not been clicked on easy800-SWD, the "Mandatory station" device parameter is checked for each SWD station. If the missing SWD station is not a mandatory SWD station, easy800-SWD stays in RUN mode and continues working with the remaining SWD stations. The POW LED flashes green, Config LED shows continuous green; SWD LED shows continuous green.

If the SWD station is mandatory, the SWD LED flashes red and the "STOP on SWD error" device parameter decides how easy800-SWD is to respond further:

- "STOP on SWD error" is activated: easy800-SWD switches to STOP. The POW LED shows continuous green, Config LED shows continuous green; SWD LED flashes red.
  If this error occurs during operation, e.g. because a station responds too late or is faulty, easy800-SWD switches directly to RUN mode as soon as the error is rectified. A restart is not necessary.
- "STOP on SWD error" is deactivated: easy800-SWD stays in RUN mode. The POW LED flashes green, Config LED shows continuous green; SWD LED flashes red.

“Incorrect” station: When is this an SWD error?

An “incorrect” SWD station on the SWD line can only be found on startup since the replacement of an SWD station is not permitted when the device is switched on. The following therefore distinguishes which cases an “incorrect” station can occur on an SWD line after startup.

If easy800-SWD determines that an SWD station of the actual configuration does not match the target configuration, it defines it as an “incorrect” station. easy800-SWD sets the group alarm I13 to 1. The DIAG diagnostic bit stays at 0.

The same response also takes place if easy800-SWD finds more stations on the SWD line than is stored in the target configuration.

With an “incorrect” station easy800-SWD will not start in RUN mode. The POW LED shows continuous green, Config LED shows continuous green; SWD LED flashes red.

You may wish to permit the “incorrect” station, because, for example, you wish to replace a faulty SWD station with a new one. Perhaps you wish to replace an SWD station with a compatible SWD station or with a universal module.

In every case, you must press the Config configuration button. This will store the “incorrect” station in the target configuration. If you then restart the device, easy800-SWD will not find any difference between the target configuration and the actual configuration, i.e. no more “incorrect” stations. However, it will find a difference between the project configuration and target configuration. The “Replaceable with universal module” and “Compatible devices permissible” device parameters determine whether operation can continue despite this difference.

You set the “Replaceable with universal module” device parameter on the SWD station. You set the “Compatible devices permissible” on the easy800-SWD.

Universal module is “incorrect” station

If the station is a universal module and the “Replaceable with universal module” device parameter has been clicked, the SUBST diagnostic bit is set to 1 and easy800-SWD switches to RUN mode.

If the station is a universal module and the “Replaceable with universal module” device parameter has not been clicked, the “STOP on SWD error” device parameter is checked:

- “STOP on SWD error“ is activated: easy800-SWD stays in STOP mode. The POW LED shows continuous green, Config LED shows continuous red; SWD LED shows continuous green.
- “STOP on SWD error“ is deactivated: easy800-SWD switches to RUN mode. The POW LED flashes green, Config LED shows continuous red; SWD LED shows continuous green.

Compatible SWD station is “incorrect” station

Stations on the SWD line can be replaced with compatible stations. Normally, compatible stations can be provided with a greater range of functions than the stations to be replaced. For example, the M22-SWD-LED-W station can be replaced with M22-SWD-K11LED-W. Compatible SWD stations are shown in the list → section “Compatibility of SWD stations”, page 363.

If the station is a compatible station and the “Compatible devices permissible” device parameter has been clicked on easy800-SWD, the “STOP on SWD error” device parameter is checked. The Config LED flashes green to indicate the replacement of the SWD node from the project configuration with a compatible SWD station in the target configuration.

If the station is a compatible station and the “Compatible devices permissible” device parameter has not been clicked, the “STOP on SWD error” device parameter is checked. This device parameter is also checked if the SWD station is not compatible. The Config LED shows a continuous red light to indicate the impermissible SWD station in both cases.

- “STOP on SWD error“ is activated: easy800-SWD stays in STOP mode. The POW LED shows continuous green, Config LED is continuously lit red, as described above, or flashes green; SWD LED shows continuous green.
- “STOP on SWD error” is deactivated: easy800-SWD switches to RUN mode. The POW LED flashes green, Config LED is continuously lit red, as described above, or flashes green; SWD LED shows continuous green.

The meaning of the LEDs is described in → Table 20.

LEDs on the device

The SWD-LED indicates whether the actual configuration of the SWD line matches the target configuration.

The Config LED shows the result of the configuration comparison between the saved target configuration and the project configuration below.
The devices and station parameters are taken into account.

Table 20: LEDs of easy800-SWD

LED		Status
POW	Off	no supply voltage at POW
	green continuous light	easy800-SWD in STOP mode
	Green flashing	easy800-SWD in RUN mode
SWD	Off	no supply voltage at POW no target configuration present
	green continuous light	Actual configuration = Target configuration
	green flashing	Actual configuration is determined
	Orange flashing	Target configuration is determined
	Red flashing	a required station missing One station too many Target configuration ≠ Actual configuration
	red continuous light	No SWD line present No 15 V DC device voltage present
Config	Off	no project configuration present, because e.g. • Initial Commissioning or • Program in easy806-SWD was deleted
	green continuous light	Target configuration = Project Configuration
	Green flashing	The station of the project configuration was replaced by a compatible station in the target and actual configuration.
	red continuous light	Target configuration ≠ Project Configuration

LEDs on the station

Table 21: LEDs on the relevant station

LED		Status
LED device	Off	no 15 V supply voltage
	green flashing	Waiting for addressing
	green fast flashing	Error on station and diagnostic alarm is 1
	green continuous light	Station matches the target configuration.

Further information is provided in the manual “SmartWire-DT - The System” (MN05006002Z-EN).

8 easy800 Settings

The following easy800 settings are only possible with devices provided with function keys and a display. These settings must be carried out in the main or System menu.

Alternatively use the easySoft-Pro programming software to complete all the settings via the software.

8.1 Password protection

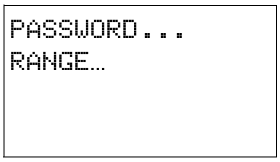
Password protection prevents access to different areas such as the CIRCUIT DIAGR., CLOCK or the serial INTERFACE. Further information on the areas concerned → section "Protecting special areas", page 298. The System menu is always protected when a password is activated.

→ A password entered in the easy800 device is transferred to the memory card with the circuit diagram, irrespective of whether it was activated or not.
If this easy800 circuit diagram is reloaded from the card, the password is also transferred to the easy800 device and is activated immediately.

8.1.1 Inputting a password

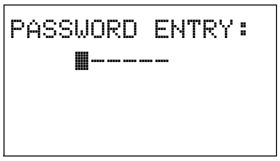
You can set up a password in the System menu, irrespective of whether operating mode RUN or STOP is selected. You cannot enter the System menu if a password is already activated.

Numbers between 0 and 9 can be used for the password. In this case the password consists of a value between 000001 and 999999.



```
PASSWORD...
RANGE...
```

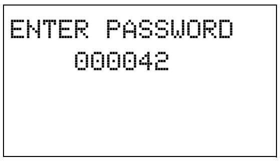
- ▶ Press DEL and ALT to call up the System menu.
- ▶ Select the menu option SECURITY… to enter the password.
- ▶ Press the OK pushbutton and enter the STD PASSWORD.
- ▶ Press OK once more to enter password entry mode.



```
PASSWORD ENTRY:
    ▮-----
```

If still no password has been entered, easy800 device changes directly to the password display and displays six dashes: No password available.

- ▶ Press OK to cause six zeros to appear.
- ▶ Set the password using the cursor buttons:
  - ‹ › Select position in the password.
  - ∧∨ set a value between 0 to 9.
  - use ‹ › to select other positions in the password as required and select a value between 0 to 9.

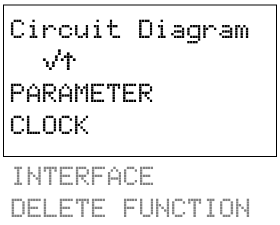


```
ENTER PASSWORD
     000042
```

- ▶ Press OK to save the new password.

Use OK to exit the password display and proceed with ESC and ∨ to the RANGE… menu. You can define here which area you wish to protect with the password.

Protecting special areas



```
Circuit Diagram
  √↑
PARAMETER
CLOCK
INTERFACE
DELETE FUNCTION
```

- ▶ Press OK.
- ▶ Select the function or the menu to be protected.
- ▶ Press the OK in order to protect the function or the menu (√ = protected).

→ At least one area must be protected. In the default setting the circuit diagram is selected.

- CIRCUIT DIAGRAM: The password is applied to the circuit diagram as well as function blocks that are not enabled. This area also prevents the transfer of a circuit diagram from and to the memory card.
- PARAMETER: The PARAMETER menu is protected.
- CLOCK: Date and time are protected with the password.
- MODE: It is not possible to change the operating mode from RUN to STOP operating mode and vice versa using the function keys of the device concerned. The same applies for the Run/Stop buttons of easySoft-Safety.

> ⚠ **DANGER**
>
> This password protection does not prevent operating mode changes in the following cases:
>
> - With REMOTE RUN activated the NET station 2-8 follows the operating mode change of the NET station with NET-ID 1, → Section “Description of the NET-PARAMETERS“ ->„Automatic change of the RUN and STOP mode (Station Menu REMOTE RUN)“ on Page 280.
> - Operating mode changes can also be triggered through local expansion units (communication modules).

- INTERFACE: Protects against access to the multi-function interface of this device. Data exchange via the easyNet is not affected. Take into account the restricting effect of a protected interface if you have to reset the easy800 device to factory settings (→ Section “Password no longer known or incorrect“, Page 301).
- DELETE PROGRAM: When this function is not activated, the question “DELETE PROG?” will appear if the password entry is entered incorrectly four times. This question does not appear if you activate this function and protect the DELETE PROGRAM area. However, it is no longer possible to make changes in protected areas if you forget the password.

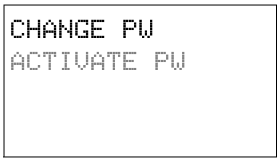
The password is now valid but not yet activated.

8.1.2 Activating passwords

An already entered password can be activated in four ways in order to activate password protection:

- via the password menu (see below)
- automatically when the device is switched on again.
- automatically after a protected circuit diagram is loaded.
- automatically if 30 minutes after unlocking
  - a telegram is not sent from easySoft-Safety via the multi-function interface.
  - a pushbutton is not actuated.

Activating password via the password menu



```
CHANGE PW
ACTIVATE PW
```

▶ In the system menu enter the ACTIVATE PW menu via SAFETY -> PASSWORD.

The device shows this password menu only if a password is present.

▶ Then press the OK button.

→ Make a notes of the password before activating it. If the password is no longer known, the device can be unlocked if the DELETE PROG area was not activated. However, the circuit diagram and data settings will be lost.

> *CAUTION*
>
> The following applies if the password is not known or is lost and the Delete password function is disabled: The device can only be reset to the factory setting at the manufacturers.
> The program and all data will be lost.

▶ If you have deactivated a password beforehand, select ACTIVATE PW in the system menu and press OK.

The password is now active, i.e. password protection is activated.

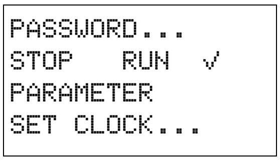
You must unlock the device with the password before you implement a protected function, enter a protected menu or the System menu.

8.1.3 Unlocking easy800

Unlocking easy800 deactivates password protection. You can reactivate password protection later via the Password menu or by switching the supply voltage off and on again.

→ The function of an area is accessible if a password is entered to unlock an area. All other functions protected by this password are now also unlocked.

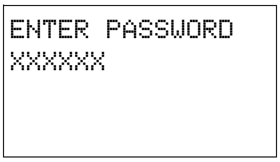
The procedure to unlock the configuration or parts of it is always the same.



```
PASSWORD...
STOP   RUN  ✓
PARAMETER
SET CLOCK...
```

- Press OK to enter the main menu. The PASSWORD… menu item will flash.
- Press OK to enter the password entry menu.

→ If easy800 shows the PROGRAM main menu instead of PASSWORD, this means that no password protection is active.

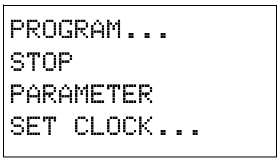


```
ENTER PASSWORD
XXXXXX
```

The easy800 device shows the password entry field.

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

The easy800 device will automatically return to the status display if the password is entered correctly.



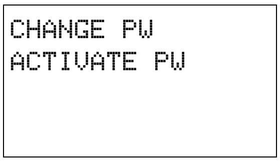
```
PROGRAM...
STOP
PARAMETER
SET CLOCK...
```

The PROGRAM menu option is now accessible so that you can edit your circuit diagram.

The System menu is also accessible.

8.1.4 Changing passwords

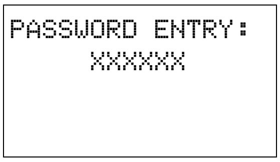
- Unlock the easy800 device as previously described.
- Press DEL and ALT to call up the System menu.
- Open the password menu via SECURITY... -> PASSWORD…-> CHANGE PW.



```
CHANGE PW
ACTIVATE PW
```

The CHANGE PW entry flashes.

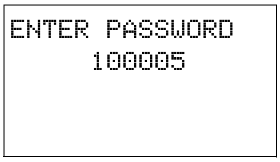
The easy800 device only shows this menu if a password is already assigned.



```
PASSWORD ENTRY:
     XXXXXX
```

- Press OK to enter the password entry menu.
- Press OK to move to the 6-digit entry field.

The current password will be displayed.

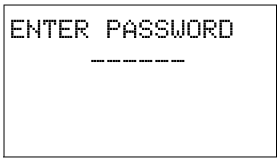


```
ENTER PASSWORD
    100005
```

- ▶ Modify the six password digits using the cursor buttons.
- ▶ Confirm with OK.

Use ESC to close the entry screen.

8.1.5 Delete password



```
ENTER PASSWORD
    ------
```

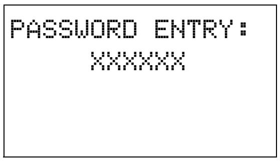
- ▶ Press DEL and ALT to call up the System menu.
- ▶ Open the password menu via SECURITY... -> PASSWORD…-> CHANGE PW.

If a password is not entered, the easy800 device shows six dashes. An existing password is displayed with XXXXXX.

8.1.6 Password no longer known or incorrect

If you no longer know the exact password, you can try to re-enter the password several times. If the DELETE PROG function is activated in the AREAS menu, the password can be entered as often as required.

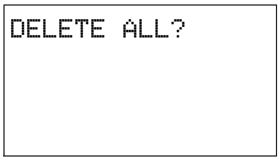
> Assumption: The DELETE PROG area was not selected, → Section “Protecting special areas“, Page 298.



```
PASSWORD ENTRY:
    XXXXXX
```

Have you entered an incorrect password?

- ▶ Re-enter the password.



```
DELETE ALL?
```

After the fifth entry attempt easy800 device will ask whether you wish to delete the circuit diagram and data.

- ▶ Press
  - **ESC**: Circuit diagram, data or password are not deleted.
  - **OK**: Circuit diagram, data and password are deleted.

easy800 will return to the Status display.

> If you no longer know the exact password, you can press **OK** to unlock the protected easy800 device. 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 five attempts to enter the password.

8.2 System

8.2.1 Switch on input delay

The easy800 device is factory set to evaluate input signals with an input delay, the so-called DEBOUNCE function. This ensures that any contact bouncing of switches and pushbuttons is masked out.

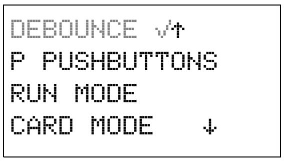
However, for many applications, it is necessary to evaluate very short input signals. In this case, the debounce function can be switched off.

The input delay is set in the System menu.

▶ Press DEL and ALT to call up the System menu.
▶ Move to the menu SYSTEM and then to the DEBOUNCE menu.

→ If the easy800 device is protected with an STD PASSWORD, you must remove password protection before you can call the system menu.

Activating DEBOUNCE



```
DEBOUNCE √↑
P PUSHBUTTONS
RUN MODE
CARD MODE   ↓
```

▶ To activate DEBOUNCE, choose DEBOUNCE and press OK.
The debounce is activated and the display changes to DEBOUNCE √.

▶ Press ESC to return to the previous menu.

Deactivating DEBOUNCE

If the menu displays DEBOUNCE √ (with a tick), this means that the debounce function is already activated.

▶ To deactivate DEBOUNCE, choose DEBOUNCE √ and press OK.
Debounce mode will be deactivated and the display will show DEBOUNCE without a tick.

→ How easy800 input signals are processed internally,
→ Section “Input delay (debounce)“, Page 321.

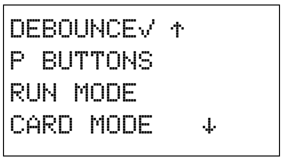
8.2.2 P PUSHBUTTONS

Even though the cursor buttons (P buttons) have been set as contacts in the circuit diagram, these are not activated automatically. In order to prevent accidental actuation, the cursor buttons cannot switch until they have been activated.

Activate the P pushbuttons in the system menu.

→ If easy800 is protected with a password, you must remove password protection before you can call the system menu.

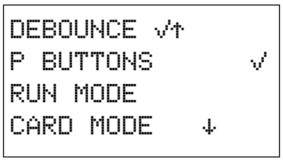
The P buttons are activated and deactivated via the P BUTTONS menu.



```
DEBOUNCE✓ ↑
P BUTTONS
RUN MODE
CARD MODE   ↓
```

- ▶ Press DEL and ALT to call up the System menu.
- ▶ Select the SYSTEM menu.
- ▶ Move the cursor to the P BUTTONS menu.

Activating P buttons



```
DEBOUNCE ✓↑
P BUTTONS     ✓
RUN MODE
CARD MODE   ↓
```

If is displaying easy800 P PUSHBUTTONS ✓, this means that the P pushbuttons are active.

- ▶ Otherwise select P PUSHBUTTONS and press OK.

The easy800 device chances the display to P BUTTONS ✓ indicating that the P buttons are activated.

- ▶ Press ESC to return to the status display.

→ The P pushbuttons are only active in the status display and the text display. In this display you can use the P buttons to activate inputs in your circuit diagram.

Deactivating the P pushbuttons

- ▶ Select P PUSHBUTTONS ✓ and press OK.

The tick behind P PUSHBUTTONS disappears and the P pushbuttons are deactivated.

→ The P buttons are automatically deactivated if you load a circuit diagram from the memory card or via easySoft-Pro onto the easy800 device or if you are deleting a circuit diagram in the easy800 device.

8.2.3 Setting the startup behavior

easy800 devices without a display can only be started in RUN mode. After power up, the device switches directly to RUN mode if it contains a valid circuit diagram.

→ If the device is in the NET, observe the possible startup behavior settings available in the NET → Section “Automatic change of the RUN and STOP mode (Station Menu REMOTE RUN)“Page 280.

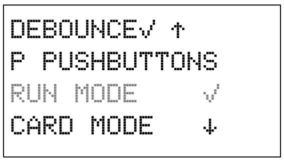
The startup behavior of easy800 devices with a display can be set. The startup behavior is an important aid during the commissioning phase. The circuit diagram which easy800 contains is not yet fully wired up or the system or machine is in a state which easy800 is not permitted to control. If the easy800 device is then switched on, it should not be possible to activate the outputs.

Set the Startup behavior in the system menu.

→ If the easy800 device is password protected, the System menu is only accessible after the device has been unlocked (→ Section “Unlocking easy800“, from Page 300).

Activating RUN MODE

▶ Switch to the system menu.
If the easy800 device shows RUN MODE √, the device will start in RUN operating mode when the supply voltage is switched on. However, if there is no circuit diagram in the device, it will stay in STOP operating mode until a circuit diagram is entered or downloaded.

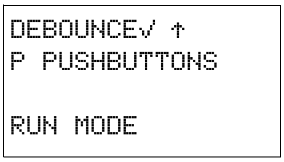


```
DEBOUNCE√ ↑
P PUSHBUTTONS
RUN MODE    √
CARD MODE   ↓
```

▶ Otherwise select RUN MODE and press OK.
RUN mode is activated.

▶ Press ESC to return to the status display.

Deactivating RUN MODE



```
DEBOUNCE√ ↑
P PUSHBUTTONS

RUN MODE
```

▶ Switch to the system menu.
▶ Select RUN MODE √ and press OK.

The RUN mode function is deactivated.

The easy800 device is factory set with RUN MODE å displayed, i.e. the device starts in RUN operating mode when the voltage supply is switched on.

Menu displayed	Status of easy800 after startup
RUN MODE	easy800 is in STOP mode
RUN MODE√	easy800 switches to RUN mode

Behavior on deleting the circuit diagram

The setting for the startup behavior is a device function of easy800 standard. 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

The setting is retained if a valid circuit diagram is transferred from easy800 standard to a memory card or PC. This also applies in the opposite direction, i.e. from the memory card or PC to the easy800.

8.2.3.1 CARD MODE

The startup behavior with a memory card is designed for applications in which a program change can even be carried out by unskilled personnel by simply exchanging the memory card. easy800-SWD does not have this function.

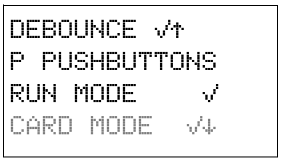
If the program on the memory card is different to the program in the easy800 standard device, the program from the card will be loaded on power up first of all and then started in RUN mode. If the programs only contain different function block setpoint values (constants), the program on the memory card is not loaded. The program is retained in the device and is started.

Default settings: CARD START inactive

▶ Switch to the system menu.

→ If the easy800 standard device is password protected, the System menu is only accessible after the easy800 standard device has been unlocked (→ Section “Unlocking easy800”, from Page 300).

Activating CARD MODE



```
DEBOUNCE √↑
P PUSHBUTTONS
RUN MODE    √
CARD MODE  √↓
```

▶ Otherwise choose CARD MODE and press OK.

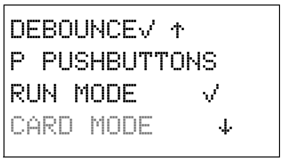
The easy800 standard device now displays CARD MODE √.

▶ Return to the status display using ESC.

→ With CARD MODE activated:

- A easy800 device only switches to RUN mode on power up if a memory card with a valid program is fitted.
- An easy800 standard device cannot be switched to RUN mode if the program in the easy800 standard device is different to the program on the memory card when the supply voltage is switched on. In this case, the modified program must be transferred to the memory card, or CARD MODE must be deactivated.

Deactivating CARD MODE



```
DEBOUNCE√ ↑
P PUSHBUTTONS
RUN MODE    √
CARD MODE    ↓
```

▶ Select CARD MODE √ and press OK.
The CARD MODE function is deactivated.

8.2.4 Activating external pushbuttons

The I buttons and R buttons options enable the operator to operate an easy800 standard in a harsh environment by means of external function keys such as RMQ control circuit devices. easy800-SWD does not have this function.

→ You can only activate one of the two I or R button options. Inputs that are activated as operating buttons are no longer available for control tasks in the circuit diagram.

External pushbuttons that are connected to the local I input terminals or the R input terminals of the expansion unit can thus take on the same functions as the operator buttons of the device. The I or R input terminals are permanently assigned to the function keys.

Table 22: Assigning Operator Buttons and I or R Input Terminals

Function key	I-Input terminal	R-Input terminal
Cursor left	I1	R1
Cursor up	I2	R2
Cursor right	I3	R3
Cursor down	I4	R4
OK	I5	R5
ESC	I6	R6
DEL	I7	R7
ALT	I8	R8

The operator can produce the same effect as pressing the function keys (e.g. DEL+ALT) by pressing two buttons (e.g. I7+I8) simultaneously.

External pushbuttons and operator buttons of the device are evaluated at the same time thus enabling mixed operation.

In exceptional cases when the external pushbutton and the assigned function key of the device are pressed simultaneously, the actuation of the external pushbutton is ignored. This excludes the possibility of contradictory operating states.

If an external pushbutton and an function key with different functions are pressed simultaneously, this is interpreted by the device as a button combination.

8.2.5 Setting contrast and backlight

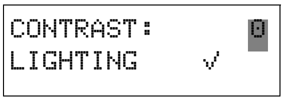
easy800-SWD does not have this function. The backlight of the LCD display can be switched off. The contrast of the display can be set to one of 5 stages. The display is not required during operation. The backlight only needs to be active during maintenance operations or when texts have to be displayed.

The backlight switches on automatically when a button is pressed and switches off 60 s after the last button was pressed. The contrast and backlight settings are device settings.

▶ Switch to the system menu.

→ If the easy800 device is password protected, the System menu is only accessible after the easy800 device has been unlocked (→ Section "Unlocking easy800", from Page 300).

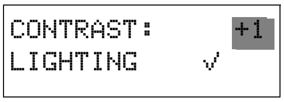
▶ In the system menu choose SYSTEM -> System-> DISPLAY and press OK.



```
CONTRAST:      0
LIGHTING    √
```

The menus for setting the contrast and backlight are displayed.

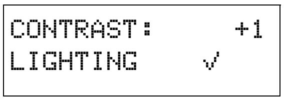
▶ Press the OK pushbutton and change the contrast setting.



```
CONTRAST:     +1
LIGHTING    √
```

▶ Use the cursor buttons ∧ and ∨ to set a contrast between –2 and +2 and press OK pushbutton to confirm the setting.

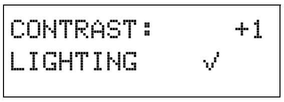
The contrast setting is valid until it is changed again.



```
CONTRAST:     +1
LIGHTING    √
```

▶ Use the cursor buttons ∧ and ∨ to move to the LIGHTING menu.

▶ Press the OK button to define whether the backlight is to be activated or deactivated.



```
CONTRAST:     +1
LIGHTING    √
```

The tick √ indicates that the backlight has been switched on.

→ easy800 standard device comes with the following default setting:

- The contrast is set to 0.
- The backlight is permanently activated. LIGHTING √.

8.2.6 Retentive data storage

Plant and machine controls require the possibility to retentively set operating states or actual values, i.e. the values should be retained safely even after the supply voltage of a machine or plant has been switched off and should be left unchanged until the actual value is overwritten.

The following operands and function blocks can be set to operate retentively:

- Markers.
- Counter function blocks.
- Data function block.
- Timing relays.

A easy800 device features 4 retentive hours-run counters. They are always retentive and can only be specifically deleted with a reset command. Hours-run counters do not reduce the memory available for retentive data.

Retentive data memory

The following table shows the maximum available memory for retentive data inside the RAM.

Device	The maximum retentive memory [Byte]
easy800 standard	200
easy800-SWD	400

Markers

The following table shows the maximum number of marker bytes that can be declared retentive and that can be limited by the above memory.

Device	Marker bytes declarable as retentive
easy800 standard	MB 01 - MB96
easy800-SWD	max. 400 bytes in the range MB01 - MB512

Meters

All C.., CH.. and CI.. function blocks can be operated with retentive actual values.

Data Function Blocks

A user-definable consistent function block DB.. range can be operated with retentive actual values.

Timing relay

A user-definable and contiguous range for timing relays T.. 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.

CAUTION

The retentive data is stored every time the supply voltage is switched off and read when the supply voltage is switched on. The data integrity of the memory is guaranteed for 10^{12} write-read cycles with easy800 standard and 10^{14} write-read cycles with easy800-SWD.

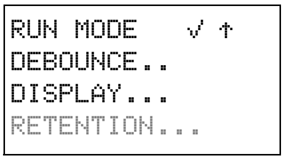
8.2.6.1 Setting retentive behaviour

Requirement:
The easy800 device must be in STOP mode.

▶ Press DEL and ALT simultaneously to switch from the status display to the system menu.

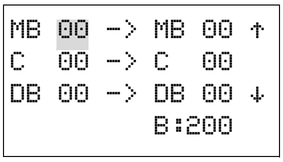
→ If the easy800 device is password protected, the System menu is only accessible after the easy800 device has been unlocked (→ Section “Unlocking easy800“, Page 300).

The devices are factory set with no retentive actual value data selected. All actual values are therefore deleted if the easy800 device is switched to STOP mode or de-energized.



```
RUN MODE    √ ↑
DEBOUNCE..
DISPLAY...
RETENTION...
```

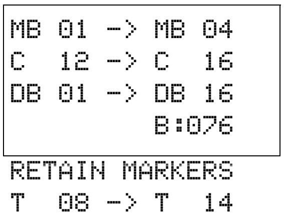
▶ Switch to STOP mode.
▶ Switch to the system menu.
▶ Enter the system menu via STD PARAMETERS and then on to the RETENTION… menu.
▶ Press the OK pushbutton.



```
MB 00 -> MB 00 ↑
C  00 -> C  00
DB 00 -> DB 00 ↓
         B:200
```

The first screen that appears is a selection dialog for operand marker bytes (MB). The display on the lower right B:200 indicates the number of free bytes.

▶ Use ⌃⌄ to select the operand.
▶ Press OK to access the Entry modes.
- ‹ › Select position from to.
- ⌃⌄ Set a value.



```
MB 01 -> MB 04
C  12 -> C  16
DB 01 -> DB 16
         B:076
RETAIN MARKERS
T  08 -> T  14
```

▶ If necessary select the next operand.

In all, you can select six different operand ranges.

Example:
MB 01 to MB 04, C 12 to C 16, DB 01 to DB 16, T 08 to T 14 should be retentive.

124 bytes have been assigned to the retentive data range. 76 bytes are still available.

▶ Select the RETAIN MARKERS option if you wish to retain the retentive actual values when transferring the circuit diagram from easySoft-Pro or the memory card to the easy800 device.

Alternatively you can select the Retain markers during transfer option in easySoft-Pro.

▶ Save the entry from .. to .. with OK.

Press ESC to exit the input for the retentive ranges.

8.2.6.2 Deleting areas

Set the ranges to be erased to the values from 00 to 00.

Example: 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 easySoft-Pro (PC) or from the memory card to the easy800 device, the retentive actual values are reset to 0. This does not occur if you have selected the Retain markers during transfer option in easySoft-Pro or activated the RETAIN MARKERS option on the device. easy800-SWD does not have this option..
- When changing the respective retentive range.
- If the set retentive ranges are reduced, only the actual values that are outside of the new range are deleted.
- When the circuit diagram is deleted via the DELETE PROG menu.

The retentive actual values are retained in easy800 on:

- Changing the operating mode
- Changing the circuit diagram
- Relays no longer used
- Changing from RUN to STOP
- Changing the startup behaviour in the SYSTEM menu
- Expanding the retentive range

8.2.6.3 Transferring retentive behavior

The retention setting is a circuit diagram setting and is carried with the program when it is transferred from the memory card or from easySoft-Pro.

8.3 Menu languages

The easy800 device provides 13 different menu languages:

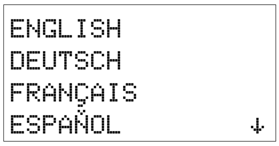
Changing the menu language

You can change the menu language in the System menu.

Language	Display
English	ENGLISH
German	DEUTSCH
French	FRANÇAIS
Spanish	ESPAÑOL
Italian	ITALIANO
Portuguese	PORTUGUÊS
Dutch	NEDERLANDS
Swedish	SVENSKA
Polish	POLSKI
Turkish	TÜRKÇE
Hungarian	MAGYAR
Czech	ČESKY
Russian	PУCCKИЙ

→ Language selection is only possible if the easy800 device is not password-protected.

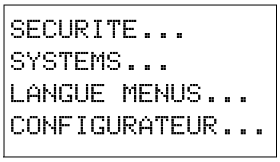
- ▶ Press DEL and ALT to call up the System menu.
- ▶ Select here the MENU LANGUAGE… for changing the language.



```
ENGLISH
DEUTSCH
FRANÇAIS
ESPAÑOL      ↓
```

The language selection for the first entry ENGLISH is displayed.

- ▶ Use ⌃ or ⌄ to select the new menu language, e.g. French FRANÇAIS.
- ▶ Confirm with OK. FRANÇAIS is provided with a tick.
- ▶ Exit the menu with ESC.



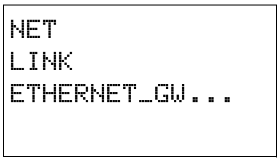
```
SECURITE...
SYSTEMS...
LANGUE MENUS...
CONFIGURATEUR...
```

easy800 device will now show the new menu language.

Press ESC to return to the Status display.

The menu language setting is a device function. This means that when the circuit diagram is deleted this does not result in the loss of the setting selected.

8.4 Configurator



```
NET
LINK
ETHERNET_GW...
```

Call the configurator in the System menu. The following menu items vary according to the device or the connected periphery.

NET

A description of how to parameterize and configure the easyNet network is provided in, → section "Commissioning an easyNet network", page 90.

LINK

In this menu you assign parameters to a configurable local expansion unit that is connected to the easyLink terminal of the easy800 standard. The menu is only accessible if this type of local expansion unit is connected. easy800-SWD does not have this feature..

The configurable expansion units include the bus modules such as EASY 204-DP and the EASY4..-DC-ME analog expansion units.

Configuring the bus module

With a bus module you only configure the address via which it is to be accessed on the fieldbus. The relevant operator manual for the bus module provides the relevant information.

Configuring analog expansion units

How to configure analog expansion devices,
→ Section "Configuring the analog expansion unit", Page 84.

ETHERNET-GW

This menu is used to assign parameters for an EASY209-SE Ethernet gateway that is connected to the serial interface.

→ An EASY209-SE Ethernet gateway can be run on easy800 standard from device version 10 and on an easy800-SWD from device version 01.

Basic requirements:

- The easy800 device is in STOP operating mode.
- easy800 standard and the Ethernet gateway are interconnected via the EASY800-MO-CAB connection cable. To connect easy800-SWD use the EU4A-RJ45-CAB2 connection cable.

The following table shows the available parameters and their factory settings for an Ethernet coupling unit.

The description of the parameters is provided in the operating manual for the Ethernet gateway AWB2528-1596., chapter Configuration via the RS232 interface and in the Help system of easySoft-Pro.

Table 23: Factory settings for an Ethernet coupling unit on easy800 or MFD

IP ADDRESS	0000.000.000.000
SUBNET-MASK	0000.000.000.000
GATEWAY-ADDRESS	0000.000.000.000 (standard gateway)
REMOTE-ADDRESS	0000.000.000.000
PORT-NUMBER	10001
REMOTE-PORT	00000
BAUD RATE	9600

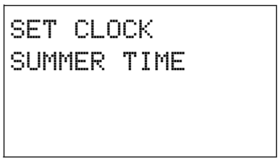
8.5 Setting time

easy800 devices feature a real-time clock with date and time functionality. The function blocks HW and HY can be used to implement time switch applications.

Setting time and date

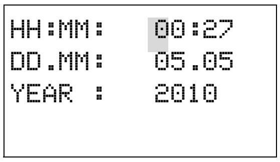
If the clock is not set or easy800 device is switched on after the battery backup time has elapsed, the clock starts with the setting
WE 1:00 2002-05-01. The easy800 clock operates with date and time so that hour, minute, day, month and year have to be set.

▶ Select SET CLOCK… from the main menu.
This will open the menu for setting the time.



```
SET CLOCK
SUMMER TIME
```

▶ Choose SET CLOCK here.



```
HH:MM:   00:27
DD.MM:   05.05
YEAR :   2010
```

▶ Set the values for time, day, month and year.
▶ Press OK pushbutton to activate Entry mode:

- < > Select the position.
- ∧∨ Change the value of a parameter
- OK: Save day and time
- ESC: Retain previous setting.

Press ESC to leave the time setting display.

Summer/winter time

The easy800 devices are fitted with a real-time clock. These devices can be automatically switched between summer and winter time. These are subject to different legal requirements in the EU, GB and USA.

→ The DST setting you have defined is always linked with the program. When commissioning systems controlled by easy800 for worldwide use it may be necessary to set and store this program also with the regional DST setting.
For these kinds of easy800 devices for worldwide use, use the "Rule" option for a DST setting.

The following options are available for setting DST parameters:

- NONE: No DST setting.
- MANUAL: (only available with easy800 standard)
  User-defined daylight saving time setting.
- EU: Date for European Union;
  Start: Last Sunday in March; End: Last Sunday in October.
  Winter time → summer time change: from 2:00 AM to 3:00 AM
  Summer time → winter time change: from 3:00 AM to 2:00 AM.
- GB: (only available with easy800 standard)
  Start: Last Sunday in March; End: Last Sunday in October.
  Winter time → summer time change: from 1:00 AM to 2:00 AM
  Summer time → winter time change: from 2:00 AM to 1:00 AM.

→ When using an easy800-SWD, you set the start and end of summer time via the RULE menu or select the GB rule supplied in easySoft-Pro for the currently applicable DST setting.

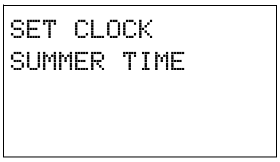
- US: Start: Second Sunday in March; End: First Sunday in November.
  Winter time → summer time change: from 2:00 AM to 3:00 AM
  Summer time → winter time change: from 2:00 AM to 1:00 AM.

→ When using an easy800 standard, set the beginning and the end of the summer time via the RULE menu or in easySoft-Pro choose the supplied rule New North America DST for the currently applicable DST setting.

- RULE: User-defined dates and changeover times as well as time difference selectable between 0 and 3 hrs; can be set in stages of 0.5 hrs.

Setting DST parameters

Select SET CLOCK… from the main menu and press OK.



```
SET CLOCK
SUMMER TIME
```

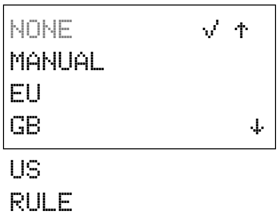
This will open the menu for setting the time.

▶ Select the DST menu and press OK.

The time change option is shown in the next menu.

Selecting the DST setting option

The standard setting is NONE for automatic DST changeover (✓ at NONE).

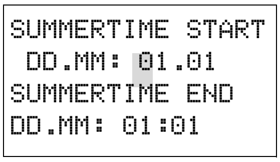


```
NONE        √ ↑
MANUAL
EU
GB            ↓
US
RULE
```

▶ Select the required option for the DST setting and press OK.

Manual option

To set your own date.



```
SUMMERTIME START
 DD.MM: 01.01
SUMMERTIME END
DD.MM: 01:01
```

- ▶ Change to the MANUAL menu and press OK 2 x.
  - Select the position with ‹ ›.
  - Change the value with ∧∨.
  - Press OK to save day and time or press ESC to restore the previous setting.
- ▶ Press ESC to leave the display.
- ▶ Select the day and the month on which the summer time is to start.
- ▶ Select the day and the month on which the summer time is to end.

On the easy800 device, the relevant time change in the subsequent years always takes place on the selected days, for example each Sunday.

→ On the changeover day, the following times apply:
Winter time → summer time change: from 2:00 AM to 3:00 AM
Summer time → winter time change: from 3:00 AM to 2:00 AM.

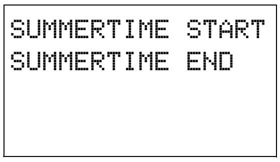
"EU", "GB" or "US" options

- ▶ Move to one of the EU, GB or US menus and press OK.
- ▶ Exit the menu with ESC.

The time setting is active and the easy800 device switches to the following times according to these regional presets:

Rule option

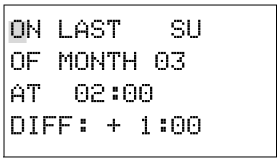
→ If you create a rule in the rule editor of the easySoft-Pro, you can access the national time scenarios provided on a row. These include possible worldwide time scenarios both for the northern and for the southern hemisphere. In special cases you can create your own special rules based on these.



```
SUMMERTIME START
SUMMERTIME END
```

- ▶ Change to the RULE menu and press OK.
- ▶ Choose SUMMERTIME START and press the OK button.

The flashing cursor is located in line 1, column 1 on the word ON.



```
ON LAST   SU
OF MONTH 03
AT  02:00
DIFF: + 1:00
```

- ▶ Press OK gain to move to Entry mode in order to set the changeover day parameters for summer time.
- ▶ Use cursor buttons ∧ or ∨ to set day 1 to 4 or the last day and press OK.
- ▶ Position the cursor in succession with the ‹ › or ∧∨ buttons at the required entry locations and change the Weekday parameter (SO), Month (OF xx. Month), Time (AT xx:xx) or the time difference (DIFF:) as previously described for the changeover day.
- ▶ Press ESC to exit the SUMMER TIME START and choose SUMMER TIME END.
- ▶ Set the changeover data to winter time as previously described for the changeover day to summer time and exit the menu with ESC.

The time difference (DIFF:) parameter corresponds to the entry for the changeover day to summer time and cannot be changed in this menu. The easy800 device now alternates between summer and winter time according to your rule.

9 Inside easy800

In conventional control systems, a relay or contactor control processes all the rungs in parallel. The speed with which a contactor switches is thus dependent on the components used, and ranges from 15 to 40 ms for relay pick-up and drop-out.

With easy800 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 circuit diagram is processed cyclically every 0.1 to 40 ms.

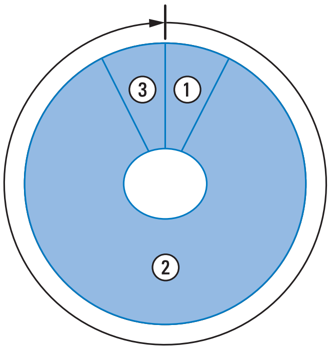


Figure 208:Cycle timeeasy800

① Read process image
- Local inputs
- Function block inputs
- Markers
- Diagnostics information
- NET stations inputs
- SWD input operands

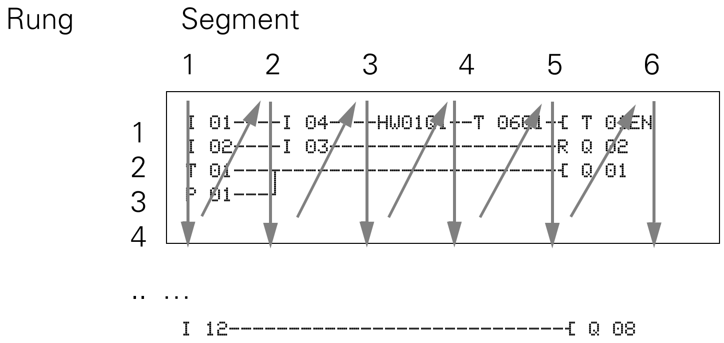
② Process circuit diagram and function block diagram

③ Write process image
- Local outputs
- Function block outputs
- Markers
- NET stations outputs
- SWD output operands

During this time, the easy800 device passes through six segments in succession.

If the operands of the inputs and outputs are addressed in the easy800, the signal states of the digital inputs/outputs are not scanned but a memory range in the device system memory is accessed. This memory range is called the process image. The process image is divided into two sections: the process image for the inputs and the process image for the outputs.

9.1 How the easy800 device evaluates circuit diagrams and function blocks


Rung
Segment
1
2
3
4
5
6
1
2
3
4
..
...
I 01----I 04----HW01Q1--T 06Q1--[T 01EN
I 02----I 03-----------------------R Q 02
T 01---┬---------------------------[Q 01
P 01---┘
I 12-----------------------------[Q 08

Segment 1 - 4

The easy800 device evaluates the contact fields within the first four segments. The evaluation starts in the first segment in circuit diagram line 1 and continues from top to bottom until circuit diagram line n is reached. The easy800 device then moves to the next contact segment and continues to evaluate from top to bottom until it has reached the last contact in the fourth segment. During this process it checks whether contacts are switched in parallel or in series and saves the switching states of all contact fields.

Segment 5

In the fifth segment the easy800 device assigns all coils in one pass, from the circuit diagram line 1 - n, with the new switch states from the process image of the outputs.

Segment 6

In the sixth segment which is outside of the circuit diagram, the function blocks present in the function block list are evaluated.

The easy800 device uses this sixth segment in order to:

- process the existing function blocks.
  The output data of a function block is immediately up-to-date after it has been processed.
  The function blocks are processed by the easy800 device in the order of the function block list (→ FUNCTION BLOCKS menu).

The following requirements must be fulfilled when using particular function blocks (→ section 9.4, "Allowing for the special features of function blocks", page 320).

- Contact the "outside world": Output relays Q 01 to Q… are switched and inputs I 1 to I… are re-read. For the expansion modules the S... output relays are switched and the R... inputs are read accordingly.
- to exchange easyNet data if this easy800 device receives new read data or provides new send data (→ Section 6, "The network easyNet" on Page 317).
- to copy all new switching states to the process image.

The easy800 device uses a fixed process image that lasts for the duration of a cycle. This ensures that each rung is evaluated with the same switching states for one cycle, even if the input signals at I1 to I12, for example, change their status several times within a cycle.

9.2 Cycle-time monitoring

The device monitors the time required for a cycle to be completed.

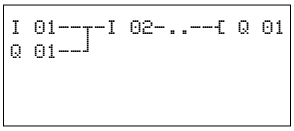
If a fault causes the cycle time to exceed 1.05 s, the device switches the Q .. output relays to the safe state 0.

9.3 What you must consider when creating the circuit diagram

The easy800 device evaluates the circuit diagram in segments, and from top to bottom within a segment, from circuit diagram line 1 - n. You should therefore remember two points when you create your circuit diagrams:

- 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



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

Start condition:
Inputs I 01 and I 02 are switched on.
Q 01 is switched off.

This is the circuit diagram of a self-latching circuit. If I 01 and I 02 are closed, the switching state of relay coil `[ Q 01` is “latched” via contact `Q 01`.

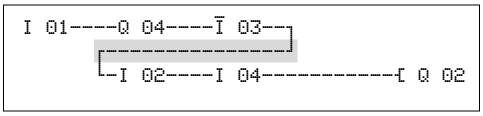
1st cycle: Inputs I 01 and I 02 are switched on. Coil Q 01 picks up.

Switching contact `Q 01` remains off because the easy800 device processes the circuit diagram segment for segment from top to bottom and from left to right from the first segment. By the time the easy800 device refreshes the process image of the outputs and therefore the coil field with `Q 01` in the fifth segment, contact field `Q 01` in the second line of the circuit diagram has already been processed.

2nd cycle: The self-latching only now becomes active. The easy800 device has transferred the coil states at the end of the first cycle to contact `Q 01`.

Example: Do not wire backwards

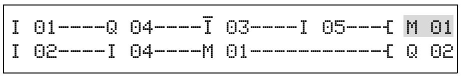
The following circuit diagram shows an example of why you should not wire backwards. In the third rung, the easy800 device finds a connection to the second rung in which the first contact field is empty. The output relay is therefore not switched.



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

Figure 209:Circuit diagram with five contacts, invalid

→ Use one of the 96 auxiliary relays M.., if you have to connect more than four contacts in series.



```
I 01----Q 04----Ī 03----I 05---[ M 01
I 02----I 04----M 01-----------[ Q 02
```

Figure 210:Circuit diagram with M 01 marker relay

9.4 Allowing for the special features of function blocks

How the easy800 device evaluates the high-speed counters CF, CH and CI

In order to evaluate the count pulses of up to 5 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.

Operating a DC or FT function block

→ The following has to be taken into account when operating a PID controller (DC) or a signal smoothing filter (FT):
The cycle time of the program must be less than the scan time of these function blocks. If the cycle time is greater than the scan time, these function blocks cannot supply any constant results.

9.5 Time behavior of the inputs and outputs

The response time, which is measured from the reading in of a digital input signal to the setting of the associated output, is determined by the timing characteristics of the device inputs and outputs as well as the size and design of the circuit diagram.

Input delay (debounce)

The time required to switch the inputs and outputs in the circuit diagram can be increased on the easy800 device using an input delay, the so-called DEBOUNCE, → Section 8.2.1, "Switch on input delay", Page 302.

This function is useful, for example, in order to ensure a clean switching signal despite contact bounce.

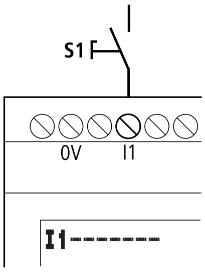


Figure 211:easy800input assigned a switch

easy800 DC and easy800 AC units function with different input voltages and therefore also have different evaluation methods and delay times.

9.5.1 Delay time on easy800 DC devices

Delay time with DEBOUNCE activated

When DEBOUNCE is activated, the delay time for DC signals is 20 ms.

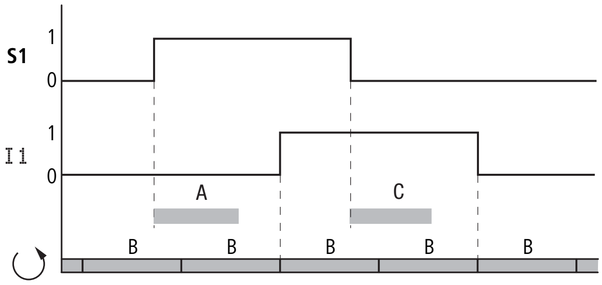


Figure 212:Delay times for evaluating an DC input signal and an activated DEBOUNCE

An input signal S1 must therefore be > 15 V on the input terminal for at least 20 ms before the switch contact will change from 0 to 1 (A). If applicable, this time must also include the cycle time (B) since a easy800 device does not detect the signal until the start of a cycle.

When the DC voltage signal drops out from 1 to 0 and DEBOUNCE is active, the same delay time (C) of 20 ms is applied before the signal is transferred to the next circuit diagram cycle. For this the input signal S1 must be < 5 V at the input terminal.

→ When using high-speed counter function blocks, the delay time for the input signals must not exceed 0.025 ms.
DEBOUNCE must therefore be deactivated.

Delay time with DEBOUNCE deactivated

When DEBOUNCE is deactivated, the typical delay time for DC signals on the input for easy800 standard caused by the hardware lies between 0.025 ms and 0.25 ms (A). The delay time for EASY806-DC-SWD is 0.02 ms. The cycle time (B) here must be added as well before the signal is safely transferred to the circuit diagram.

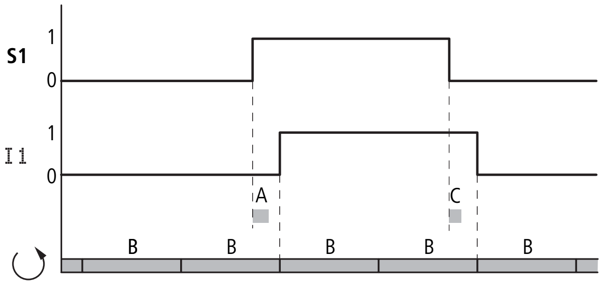


Figure 213:Switching behaviour with DEBOUNCE deactivated

→ When DEBOUNCE is deactivated ensure that input signals are free of noise. The easy800 device responds to very short signals.

→ To allow reliable recognition and processing of the input signal, it must be applied stably for a certain duration, the length of which depends on the circuit diagram processing cycle time.

Typical delay times with DEBOUNCE deactivated are:

- Switch-on delay for:
  - I1 to I4 easy800 standard: 0.025 ms
  - I1 to I4 EASY806-DC-SWD: 0.02 ms
  - I5 to I12: 0.25 ms
- Off-delay for:
  - I1 to I4 easy800 standard: 0.025 ms
  - I1 to I4 EASY806-DC-SWD: 0.02 ms
  - I5, I6 and I9 to I10: 0.4 ms
  - I7, I8, I11 and I12: 0.2 ms

9.5.2 Delay time on easy800 AC devices

With AC inputs the easy800 standard device scans the input signal in long or short scan cycles t_{SC} depending on whether DEBOUNCE is activated/ deactivated. The scan cycle also depends on the frequency of the supply network.

Delay time with DEBOUNCE activated

If the DEBOUNCE function is activated, the easy800 device checks whether a positive half-wave is present at an input terminal during two successive scan cycles t_{SC1} (1st and 2nd scan pulse at A). If the easy800 device registers two positive half waves in succession, it switches the appropriate input (contact) internally from 0 to 1.

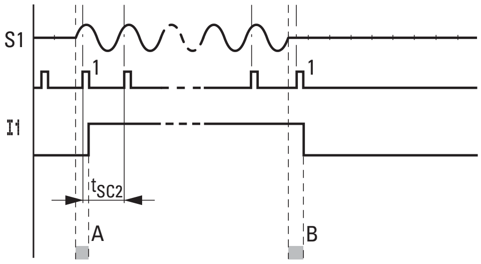


Figure 214:Delay times for evaluating an AC input signal and an activated DEBOUNCE

The typical input delay caused by the DEBOUNCE is therefore at least 80 ms. The cycle time must be added to this since an easy800 device does not transfer the signal into the circuit diagram until the start of a cycle. Conversely, the input is switched off if the easy800 device does not detect any half-cycles twice in succession (1st and 2nd pulse at B).

t_{SC1} (50 Hz) = 40 ms, t_{SC} (60 Hz) = 33.3 ms

- Switch-on delay (normally):
  - I1…I12: 80 ms (66.7)
- Off-delay (normally):
  - I1…I12: 80 ms (66.7)

The corresponding values for 60 Hz are given in brackets.

Additional delay time due to bouncing pushbuttons

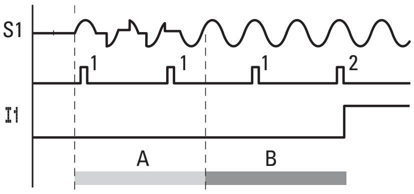


Figure 215:Pushbutton with bounce

If a pushbutton or switch bounces (A), the delay time may be extended by 20 ms (16.7 ms) (A).

Delay time with DEBOUNCE deactivated

When DEBOUNCE is deactivated, the delay time is reduced so that the easy800 device switches in shorter scan cycles t_{SC2} the appropriate input (contact) internally from 0 to 1 (A) when a positive half wave is detected.

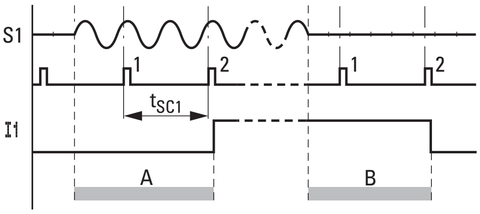


Figure 216:Switching behaviour with DEBOUNCE deactivated

If a positive half-cycle is not detected the easy800 switches off the contact (B).

t_{SC2} (50 Hz) = 20 ms, t_{SC} (60 Hz) = 16.7 ms

- Switch-on delay (normally):
  - I1…I12: 20 ms (16.7)
- Off-delay (normally):
  - I1…I12: 20 ms (16.7)

The corresponding values for 60 Hz are given in brackets.

→ A description of how to modify the delay times is provided in the section (→ Section 9.5, “Time behavior of the inputs and outputs“, Page 321).

9.6 easy800 standard expanding

You can expand an easy800 standard using the easyLink network:

- Locally with expansion units EASY618-..-RE, EASY620-DC-TE, EASY410-DC-.E, EASY406-DC-ME, EASY411-DC-ME, EASY202-RE.
- Remotely via the non-safety coupling module EASY200-EASY and one of the above mentioned easy600 expansions connected to it.

You can also use communication expansion units such as:

- EASY204-DP for PROFIBUS-DP.
- EASY221-CO for CANopen-Bus.
- EASY205-ASI for the Actuator-Sensor-Interface bus.
- EASY209-SE for Ethernet.
- EASY222-DN for DeviceNet.

Install the units and connect the inputs and outputs as described (→ Section 2.2, "Mounting the expansion unit", Page 22).

The inputs of the expansion units are processed in the circuit diagram of an easy800 device as contacts like the inputs of the basic device. The input contacts are assigned the operand identifiers R1 to R12.

R15 and R16 are the group fault alarms of the expansion devices with transistor outputs (→ section 9.8, "Displaying device information", page 327).

The outputs of the expansion units are processed as relay coils or contacts like the outputs in the basic device. The output relays are S1 to S8.

→ The EASY618-..-RE provides the outputs S1 to S6. The other outputs S7, S8 can be used internally.

How is an expansion unit detected?

If at least one R contact or contact/coil S is used in the circuit diagram, the basic device assumes that an expansion unit is connected.

easyLink transfer behavior

The input and output data of the expansion units is transferred via easyLink serially in both directions.

Take into account the longer response times of the inputs and outputs of expansion units compared to those of the basic device.

Response times of the inputs/outputs of the expansion units

The debounce setting has no effect on the expansion unit.

Transfer times for input and output signals:

- Central expansion
  - Time for inputs R1 to R12: 30 ms + 1 cycle.
  - Time for outputs S1 to S6 (S8): 15 ms + 1 cycle.

- Remote expansion module
  - Time for inputs R1 to R12: 80 ms + 1 cycle.
  - Time for outputs S1 to S6 (S8): 40 ms + 1 cycle.

Monitoring the functionality of the expansion unit

If the power supply of the expansion unit is not present, no connection can be established between it and basic unit. The expansion inputs R1 to R12, R15, R16 are incorrectly processed in the basic device and show status 0. The outputs S1 to S8 in the expansion device cannot be set.

DANGER

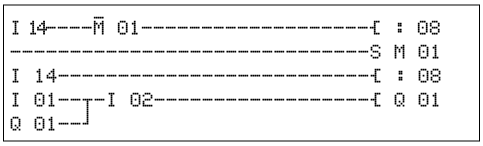
Ensure the continuous monitoring of easy800 expansion units in order to prevent switching faults in machines or systems.

The status of the internal input I14 of the basic device signals the status of the expansion unit:

- I14 = 0: expansion unit is functional.
- I14 = 1: expansion unit is not functional.

Example:

The expansion unit may be powered up later than the basic unit. This means that the basic device is switched to RUN when an expansion unit is missing. The following easy800 circuit diagram detects if the expansion unit is ready for operation or not functional.



```
I 14----M̄ 01--------------------[ : 08
-------------------------------S M 01
I 14---------------------------[ : 08
I 01--┬-I 02-------------------[ Q 01
Q 01--┘
```

Figure 217: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 voltage supply is switched on. If the circuit diagram is skipped, all the outputs retain their previous state.

9.7 Monitoring the serial interface

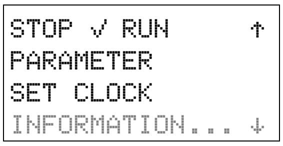
The device runs a baud rate dependent timeout monitoring of the communication via the serial multi-function interface. If a data exchange in progress is interrupted for the set timeout time, the easy800 device discards the data received up to this time and waits for a new start telegram.

Baud rate [Kbaud]	Timeout time [ms]
9.6	4000
19.2	3200
38.4	2400
57.6	2000

9.8 Displaying device information

The device information is provided for service tasks or in order to determine the performance level of the device.

This function is only available with devices featuring a display.

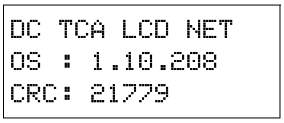


```
STOP √ RUN     ↑
PARAMETER
SET CLOCK
INFORMATION... ↓
```

- ▶ Switch to the main menu and use the cursor button ⌃ or ⌄ to move to the INFORMATION menu.
- ▶ Press OK.

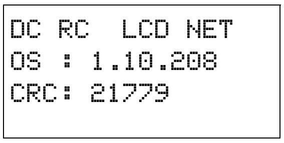
This will display all device information:

Example: EASY822-DC-TC



```
DC TCA LCD NET
OS : 1.10.208
CRC: 21779
```

Example: EASY819-DC-RC



```
DC RC  LCD NET
OS : 1.10.208
CRC: 21779
```

Line 1:

- DC/AC
  - AC (AC voltage)
  - or DC (DC voltage).
- TCA-/RC
  - T: Transistor output,
  - or R: Relay output,
  - C: Real-time clock provided,
  - A: Analog output provided.
- LCD/—
  - Display present/not present. This information can only be displayed by a locally connected MFD-CP4/-CP8 device in Terminal mode. If the remote device is run without a display, the MFD-CP4/-CP8 shows the dashes — at this position.
- NET/—
  - easyNET connection present/not present.

Line 2:

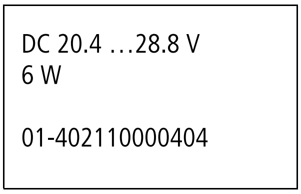
- OS: e. g. 1.12.204, operating system version.

Line 3:

- CRC: e.g. 25825 (Checksum of the operating system is only displayed in STOP mode).

▶ Press ESC to exit the display.

Device Version Number



DC 20.4 …28.8 V
6 W

01-402110000404

The device version number is shown on the back of the housing of each easy800 device. These are the first two digits of the device number.

Example: 01-402110000404

This device is of device version 01.

The device version number provides useful service information about the hardware version and the version of the operating system. The device version is important for selecting the correct control relay for easySoft or easySoft-Pro.

9.9 Loadable operating system

easy800-SWD enables you to install a new version of the operating system. For this you need a connection between the serial interface of easy800-SWD and your PC. The new operating system version provides you with a tool with an Online Help to guide you in individual steps and provide user-friendly support during the installation. For later support inquiries, the version number of the operating system file must be noted on a plate on the device. The location where the device version number is also shown is recommended for this purpose.

10 Diagnostics possibilities

10.1 easyNet

The easy800 devices output information on their own operating state via ID diagnostics contacts. This information can be evaluated in the circuit diagram and shown on the display. To activate the display press the ALT button on the device twice from the Status display.

EASY802-DC-SWD has no diagnostic contacts. EASY806-DC-SWD has the diagnostic contacts ID1 to ID8. The following table applies to easy800 standard and EASY806-DC-SWD.

Diagnostics contact	Fault on 1 state
ID 01	Fault in easyNet station 1.
ID 02	Fault in easyNet station 2.
…	…
ID 07	Fault in easyNet station 7.
ID 08	Fault in easyNet station 8.

10.2 easyLink

easy800 standard also has a group alarm for expansion units:

- I14 = 0: expansion unit is functional.
- I14 = 1: expansion unit is not functional.

10.3 SmartWire-DT

I13: Group alarm for SmartWire-DT

If an error occurs on the SWD line and the data exchange is therefore interrupted, the group alarm I13 is set to 1.

Possible causes:

1. No SWD station present
2. Incorrect SWD station present
3. More SWD stations than stored in the target configuration

Diagnostic bits of the station

If an SWD station of the target configuration is missing in the actual configuration, the PRSNT bit is set to 0.

If an SWD station of the target configuration is replaced by a universal module, the SUBST bit is set to 1.

The PRSNT and SUBST bits are located in byte 0 of the cyclical input data of the SWD stations (→ section 7.2, “Cyclical input and output data of the SWD stations”, page 286).

10.4 Transistor output

To scan whether there is a short-circuit or an overload on an output of a device with transistor outputs, use the internal inputs I15, I16, and use R15, R16 if an expansion unit is connected.

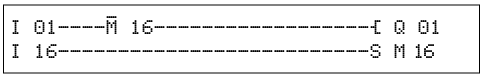
- EASY82 .-D.-T..:
  - I16: Group fault alarm for outputs Q1 - Q4.
  - I15: Group fault alarm for outputs Q5 - Q8.
- EASY620-D.-TE (expansion unit):
  - R16: Group fault alarm for outputs S1 - S4.
  - R15: Group fault alarm for outputs S5 - S8.
- easy800-SWD
  - I14: as group alarm out of function; permanently set to 1.
  - I16: Group alarm for the outputs Q1/Q2
    If both terminals are configured as inputs I3/I4, I16 is always 0.

State Fault alarms	I15 or I16, R15 or R16
No fault present	0 = switched off (N/O)
Fault present on at least one output	1 = switched on (N/O)

→ I15 and I16 can only be used on easy800 variants with transistor outputs.

The following examples show the use of the group fault alarm I16 for checking the function of the transistor outputs Q1…Q4 on easy800 standard.

Example 1: Output with fault indication



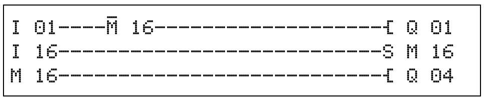
```
I 01----M̄ 16-------------------[ Q 01
I 16--------------------------S M 16
```

Figure 218:Circuit diagram for fault output via I16

The above circuit diagram functions as follows:

If a transistor output indicates a fault, I16 sets marker M16. The N/C contact of M16 switches off output Q1. M16 can be reset by switching the easy800 supply voltage off and on so that the N/C contact closes again.

Example 2: Output of operating state

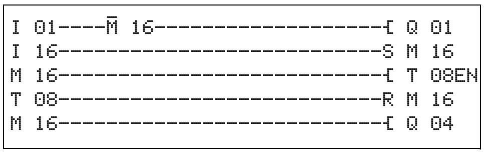


```
I 01----M̄ 16--------------------[ Q 01
I 16----------------------------S M 16
M 16----------------------------[ Q 04
```

Figure 219:Output of operating state

The above circuit diagram operates as described in example 1. If an overload is detected, the indicator light at Q4 is also actuated. If Q4 had an overload, the light would pulse.

Example 3: Automatic reset of error signal



```
I 01----M̄ 16--------------------[ Q 01
I 16----------------------------S M 16
M 16----------------------------[ T 08EN
T 08----------------------------R M 16
M 16----------------------------[ Q 04
```

Figure 220:Automatic reset of error signal

The above circuit diagram has the same function as described in example 2. Timing relay T08 (on-delayed, 60 s) is also used to reset marker M16 every 60 s. If I16 is still at 1, M16 stays set. Output Q1 is set to 1 for a short time before I16 resets it again.

11 What Happens If …?

If a easy800 does not behave as expected, the following tips can help you in rectifying any possible problems. If a program does not function as expected, in spite of a thorough simulation in easySoft-Pro, the power flow display in the easy800 display enables you to test the logic operations of the circuit diagram.

Only qualified persons should test electrical voltages while the easy800 is in operation.

11.1 Messages from the easy800 operating system

Messages on the LCD display	Explanation	Remedy
No display	Power supply interrupted	Restore power supply
	LCD is faulty	easy800 replace
Temporary display		
TEST: EEPROM	Only when switched on for the first time	-
TEST: CLOCK		
Continuous display		
ERROR: EEPROM	The memory for storing the retentive values or the easy800 circuit diagram memory is faulty.	easy800 replace
ERROR: CLOCK	Clock error	easy800 replace

11.2 Possible situations when creating circuit diagrams

Possible situations when creating circuit diagrams	Explanation	Remedy
Cannot enter contact or relay in circuit diagram	easy800 is in RUN mode	Select STOP mode
Time switch switches at wrong times	Time or time switch parameters not correct	Check time and parameters
Message when using a memory card PROG INVALID	easy800 memory card contains no circuit diagram	Change easy800 type or the circuit diagram on the memory card
	Circuit diagram on the memory card uses contacts/ relays that easy800 does not recognise	
Power flow display does not show changes to the rungs	easy800 is in STOP mode	Select RUN mode
	Association/ connection not fulfilled	Check and modify circuit diagram and parameter sets
	Relay does not activate coil	
	Incorrect parameter values/time	
	Analog value comparison is incorrect Time value of timing relay is incorrect Function of timing relay is incorrect	
Relay Q or M does not pick up	Relay coil has been wired up several times	Check coil field entries
Input not detected	Loose terminal contact	Check installation instructions, check external wiring
	No voltage to switch/button	
	Wire breakage	
	easy800Input is faulty	easy800 replace
Relay output Q does not switch and activate the load	easy800 is in operating mode STOP	Select RUN mode
	No voltage at relay contact	Check installation instructions, check external wiring
	easy800 power supply interrupted	
	easy800circuit diagram does not activate relay output	
	Wire breakage	
	easy800relay is faulty	easy800 replace

11.3 Event

Event	Explanation	Remedy
The actual values are not being stored retentively.	Retention has not been switched on.	Switch on retention in the SYSTEM menu.
The RETENTION… menu is not displayed in the SYSTEM menu.	easy800 is in RUN mode	Select STOP mode
The retentive data is cleared when there is a mode change from RUN to STOP.	This behavior only occurs when using the PW02 (pulse width modulation) function block in easy800 standard.	Avoid using the PW02 function block.
The SYSTEM menu is not displayed.	This easy800 Type model does not have this menu.	easy800 exchange if you need retention
When the device is switched on, the easy800 switches to STOP mode	No circuit diagram in the easy800	Load, input circuit diagram
	RUN START is deactivated at easy800 standard.	Activating RUN mode in Menu SYSTEM.
The contacts of the BC (data block comparison) and block transfer (BT) function blocks flash in the power flow display	The display of the easy800 standard is being updated to intermediate states too frequently although the contacts are operating properly	Ignore this section of the power flow display.
LCD display showing nothing.	No supply voltage	Switch on the power supply
	easy800 defect	Press the **OK** button. If no menu appears, replace the easy800 device.
	Text displayed with too many spaces	Enter text or do not activate text output
GW flashes on the Status display	EASY200-EASY bus coupler detected without I/O expansion	Connect I/O expansion to external easyLink
The switching devices are not working	No supply voltage at AUX	Switch on the supply voltage at AUX

11.4 Functionality of the SWD line faulty

easy800-SWD during operation or after switching on

If an error occurs on the SWD line, the SWD-LED indicates this by flashing red or with a continuous light. easy800-SWD can nevertheless process the circuit diagram further if you have not clicked the “STOP on SWD error” device parameter. However, you must decide whether this would be useful.

SWD-LED	Meaning
Red flashing	A mandatory station is missing on the SWD line. • Fit the mandatory station on the external device plug provided. • If easy800-SWD is to continue processing the circuit diagram without the SWD line, you must deactivate the “STOP on SWD error” device parameter. • You can define the station as non-mandatory by deactivating the “Mandatory station” device parameter.
red continuous light	Possible causes: • The connection to the SWD line is broken. • The 15 V DC device voltage is missing.

easy800-SWD after switching on

SWD-LED	Meaning
Red flashing	There is an “incorrect” station on the SWD line. Possible causes: • For example, you have replaced a faulty station. The serial number of the new station must first be stored in the target configuration. To do this press the Config configuration button.

The following behavior may occur:

The device stays in STOP mode after being switched on. The SWD LED indicates missing stations by flashing red. After a set time, the SWD LED shows a continuous green light and easy800-SWD switches to RUN mode.

This may be due to the fact that stations on the SWD line could not initially be found because the 15 V power supply of these stations is switched on too late by the power feed modules and the stations are not responding. As soon as the power feed module supplies the SWD stations with 15 V DC, easy800-SWD switches to operation.

easy800-SWD after pressing the configuration button

In this case, the actual configuration matches the target configuration.

Config-LED	Meaning
Green flashing	A station of the project configuration was replaced by a compatible station in the target configuration. The replacement is permitted by the "Compatible devices permissible" device parameter being clicked.
red continuous light	Configuration error: the target configuration does not match the project configuration. Possible causes: • The SWD station of the project configuration was replaced by a incompatible station. • SWD station was replaced by a compatible station, but the "Compatible devices permissible" device parameter has not been clicked. • The station was replaced by an universal module, but the "Replaceable with universal module" device parameter has not been clicked

11.5 Functionality of the NET faulty

CAUTION

You can make a visual check the functionality of the NET by means of the NET LED and in the circuit diagram via diagnostics bit ID01-ID08.

Table 24: Checking the functionality of the NET using the NET LED

Status of the NET LED	Meaning
Off	NET not operational, fault in configuration
Continuous light	NET station fault, possible causes: • Net is initialized and at least one station has not been detected. Check the plug-in connections. • You have modified the NET ID or baud rate for at least one station after the configuration has been completed. Change the configuration. • You have deleted the program on a NET station and thus also its NET configuration, → Page 94. Reconfigure the NET via station 1. • You have expanded an existing NET station and replaced it with a new device which cannot be assigned parameters.
Flashing	NET operating fault-free

Event	Explanation	Remedy
easy800 standard cannot configure EC4P easyControl devices reliably (NET parameters).	Synchronization between the devices	Set the NET parameters directly on the EC4P.

11.5.1 Monitoring via diagnostics bits

Diagnostic bits ID01 - ID08 (→ section 9.6, “easy800 standard expanding”, page 325) provide information on the presence or absence of NET stations. The state of the diagnostic bits is defined by the reception of a sign of life signal that every NET station sends.

If the sign of life signal of a configured NET station is no longer received, the diagnostics bit ID01 -ID08 for the missing NET station is immediately set to 1 for all remaining stations.

The sign of life signal is sent cyclically depending on the baud rate.

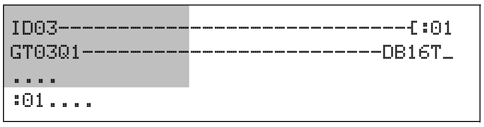
The presence of the configured NET stations is evaluated at the following time intervalls:

Baud Rate [Kbit/s]	Station must send a “sign of life” every … [ms]	Station detects a missing sign of life signal from … [ms]
1000	60	180
500	60	180
250	120	360
125	240	720
50	600	1800
20	1500	4500
10	3000	9000

If, for example, the station with NET ID 7 is missing, the diagnostics bit ID7 changes to 1 on every remaining NET station. The missing station is also indicated by means of a permanent light on the LED NET for all remaining NET stations.

> *CAUTION*
> In order to ensure that a circuit diagram with NET operands (e.g. 3GT01) is always working with the latest data, you should query the diagnostics bit ID.. of the sending NET station (see following example) before evaluating the operands.
> If you do not evaluate the appropriate diagnostics bit, this may cause malfunctions in your application.

Example of scanning the diagnostics bit ID..



```
ID03--------------------------------[:01
GT03Q1--------------------------DB16T_
....
:01....
```

Figure 221:Scanning the diagnostics bit in the circuit diagram

▨ = Visible area

Table 25: Reaction to missing sign of life signal on a NET station with or without a program.

Station with circuit diagram	NET station 1 detects the failure of a NET station 2 to 8 without a circuit diagram: Input values received via nI.., nR.., nQ.., nS.., nRN.. and GT.. are set to 0. The output values for this NET station are not changed.
Station without circuit diagram NET station with a NET ID 2…8 in REMOTE IO mode	The output values Q01 - Q08 and S01 - S08 received from NET station 1 are set to 0.

→ After the supply voltage is switched on, the other NET stations and NET stations of type easy800 start up NET communication at different speeds. When these different types are used on a NET in mixed operation and are started simultaneously, a faster starting NET station will set its diagnostics contact ID.. = 1 to indicate a NET station of type easy800 as missing.
Solution: On “all other NET stations” delay the evaluation of the diagnostics bit ID.. for the NET stations of type easy800 by approx. 5 s by using the T function block (timing relay).

11.5.2 Replacing a NET station

If you wish to replace a NET station NT2 - NT8 in a running plant, you should take preparatory measures in order to ensure fault-free commissioning of this NET station.

Example

In a network run with a baud rate of 250 KB, the NET station NT7 is to be replaced.

If you are using a easy800 device as replacement that is still in its default state and does not have a NET-ID, this device is not automatically integrated in the network. Even connected to the power supply and the NET, the LED NET remains off. It is then also not possible to load a program via the easySoft-Pro programming software (Communication View -> Device list box, NT7) into this replacement device.

The following procedure ensures fault-free commissioning of the replacement device.

- ▶ Connect the easy800 replacement device to the supply voltage and connect it at the same position to the NET as the device to be replaced.
- ▶ Switch the device display of all NET stations to Status display.
- ▶ Start the automatic configuration of the NET, → section 3.10.2, “Configuring NET”, page 94 . However, you can also do this via the programming software. As an alternative to the automatic configuration, you can also switch the supply voltage of the configured NET stations off and on.

All NET stations in the NET configuration will temporarily switch to STOP.

11.5.3 Station with NET-ID 1 fails

If the supply voltage of station NET-ID 1 fails during operation, the other NET stations can no longer be notified of the associated operating mode change.

NET stations with NET-ID 2 - 8 on which REMOTE RUN is activated therefore stay in RUN mode until the missing sign of life is detected. The time until the missing sign of life is detected depends on the baud rate and is from 180 - 9000 ms (→ Section 11.5.1, “Monitoring via diagnostics bits“ , Page 338). Up to this time, a NET station 2 - 8 operates with the data previously received via the NET.

The further behavior of the NET stations 2 - 8 depends on whether REMOTE IO is also activated as well as REMOTE RUN:

- NET station 2 - 8 with activated REMOTE RUN

This type of NET station sets the data received via the NET to zero. It retains its operating state and sets the diagnostics bit ID1 to 1 (see following safety measure).

- NET stations 2 - 8 with activated REMOTE RUN and REMOTE IO that function as input/output devices, i.e. without their own circuit diagram

This type of NET station sets the data received via the NET to zero. It switches to the STOP mode if the device display is showing the Status menu. The device follows NET station 1 when it switches to RUN mode.

Safety measure: On every NET station with a program, monitor the diagnostics bits ID.. in order to detect a missing NET station as quickly as possible (→ Section 11.5.1, “Monitoring via diagnostics bits“ , Page 338).

12 Appendix

12.1 Approval and Certification

easy800 devices are approved for use in several countries and regions as well as for operation on ships.

The same applies to expansion units.

12.1.1 Approvals and national approvals for easy800 devices

Product standards	- IEC/EN → section "12.3.1 Standards", page 344; - UL 508 (INDUSTRIAL CONTROL EQUIPMENT); - CSA C22.2 No. 142-M1987 (Process Control Equipment); - CSA C22.2 No. 213-M1987 (Non-Incendive Electrical Equipment for Use in Class I, Division 2 Hazardous Locations) - CE-mark
UL File No.	E135462
UL CCN	NRAQ
CSA File No.	012528
CSA Class No.	2252-01 + 2258-02 (for Hazardous Locations)
NA Certification	UL Listed, CSA certified
Protection type	IEC: IP20; UL/CSA Type : -

easy800 standard devices are approved for Russia, in accordance with GOST-R and for the Ukraine in accordance with Ukrain-GOST.

12.1.2 Shipping approvals for easy800 standard devices

easy800 standard	**BV** **(Bureau Veritas)**	**DNV** **(DET NORSKE VERITAS)**	**GL** **(Germanischer Lloyd)**	**LR** **(Lloyds Register of Shipping)**
Certificate number:	21607/A0 BV	A-11005	47923- 03 HH	04/20035

12.1.3 Approvals and national approvals for expansion units

Product standards	• IEC/EN → section "12.3.1 Standards", page 344; • UL 508 (INDUSTRIAL CONTROL EQUIPMENT); • CSA C22.2 No. 142-M1987 (Process Control Equipment); • CSA C22.2 No. 213-M1987 (Non-Incendive Electrical Equipment for Use in Class I, Division 2 Hazardous Locations) • CE-mark
UL File No.	E135462
UL CCN	NRAQ, NRAQ7
CSA File No.	012528
CSA Class No.	2252-01 + 2258-02 (for Hazardous Locations)
NA Certification	UL Listed, CSA certified
Protection type I/O modules	IEC: IP20; UL/CSA Type : -

Expansion units are also approved for Russia, in accordance with GOST-R and for the Ukraine in accordance with Ukrain-GOST.

12.1.4 Shipping approvals for easy6.. expansion units

easy6..	**BV** **(Bureau Veritas)**	**DNV** **(DET NORSKE VERITAS)**	**GL** **(Germanischer Lloyd)**	**LR** **(Lloyds Register of Shipping)**
Certificate number:	-	-	42991- 02 HH easy6..	02/20029 EASY618-..-RE

12.2 Dimensions

		easy800 standard	easy800-SWD
Dimensions (W x H x D)	mm	107.5 (6 TE) x 90 x 72	35 (2 space units) x 110 x 125.5
	inch	4.23 x 3.54 x 2.84	1.38 x 4.33 x 4.92
Space units (SU) width		6	2
Weight	kg	0.3	0.16
	lb	0.66	0.35
Mounting		Top-hat rail IEC/EN 60715, 35 mm or screw fixing usingfixing brackets ZB4-101-GF1 (accessories).	
Mounting position		Horizontal or vertical	

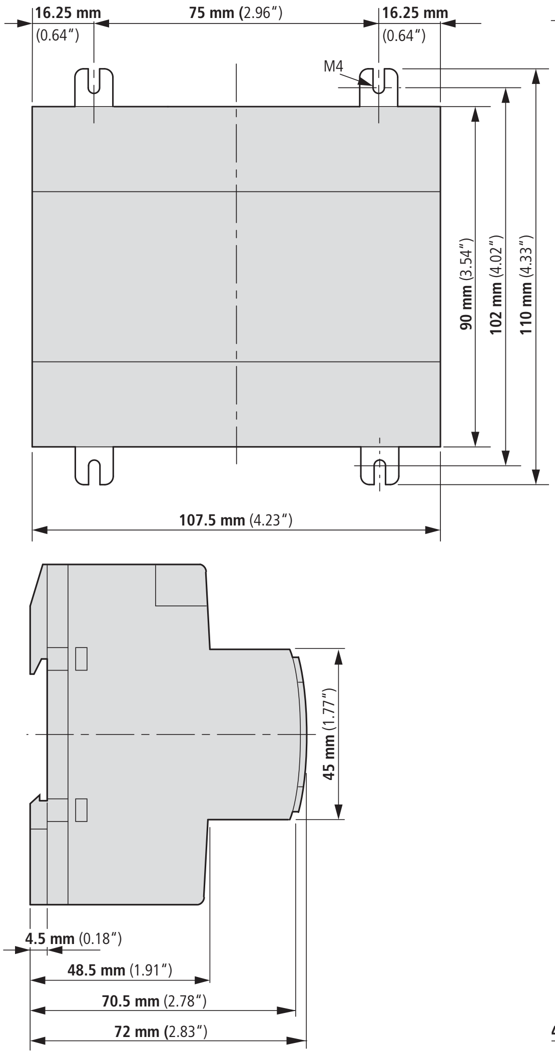


Figure 222:Dimensions easy800 standard

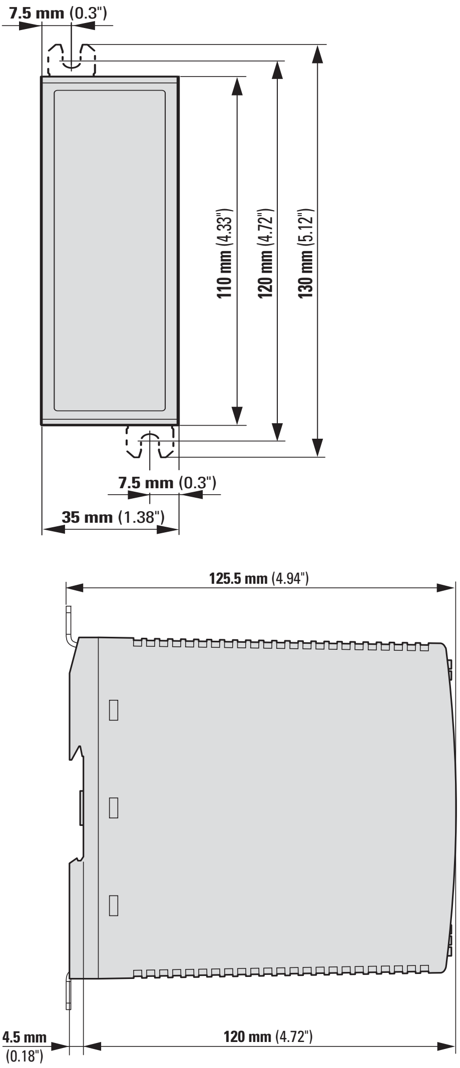


Figure 223:easy800-SWD dimensions

12.3 Technical data

Any data that is different for the easy800 standard and easy800-SWD is presented separately. Otherwise the information is valid for all easy800 devices.

12.3.1 Standards

			easy800
Standards			EN 55011, EN 50178, EN 61131-2, IEC EN 61000-4, IEC60068-2-6, IEC60068-2-27

12.3.2 General ambient conditions

Ambient climatic conditions			
Cold to IEC 60068-2-1, heat to IEC 60068-2-2, damp heat, constant, to IEC 60068-2-78; cyclical to IEC 60068-2-30, temperature change to IEC68000-2-14			
Operating ambient temperature		°C, (°F)	–25 to +55 (–13 to +131)
Condensation			Prevent condensation by means of suitable measures
LCD display (reliably legible)		°C, (°F)	0 to +55, (–32 to +131)
Storage		°C, (°F)	–40 to +70, (–40 to +158)
Relative humidity, non-condensing (IEC/EN 60068-2-30)		%	5 to 95
Air pressure (in operation)		hPa	795 up to 1080
Ambient mechanical conditions			
Protection type EN 50178, IEC 60529, VBG4			IP20
Vibrations (IEC EN 60068-2-6)			
constant amplitude			
easy800 standard:0.15 mm easy800-SWD: 3.5 mm		Hz	easy800 standard: 10 - 57 easy800-SWD: 5 - 8.4
constant acceleration			
easy800 standard: 2 g easy800-SWD: 1 g		Hz	easy800 standard: 57 - 150 easy800-SWD: 8.4 - 150
Mechanical shock resistance (IEC EN 60068-2-27) semi-sinusoidal 15 g/11 ms	Shocks		18
Drop (IEC EN 60068-2-31)	Drop height	mm	50
Free fall, packaged (IEC/EN 60068-2-32)		m	easy800 standard: 1 easy800-SWD: 0.3
Electromagnetic compatibility (EMC)			
Electrostatic discharge (ESD), to IEC EN 61000-4-2			

Air discharge		kV	8
Contact discharge		kV	6
Electromagnetic fields (RFI), to IEC EN 61000-4-3			
0.8 - 1.0 GHz		V/m	10
1.4 - 2 GHz		V/m	3
2.0 - 2.7 GHz		V/m	1
Radio interference suppression			EN 55011 Class B
Burst, to IEC EN 61000-4-4			
Supply cables		kV	2
Signal cables		kV	2
easyNet		kV	2
SWD-line		kV	2
Power pulses (surge), to IEC EN 61000-4-5 (supply cables, symmetrical)		kV	easy8..-AC: 2 easy8..-DC: 1 easySWD: 1
radiated RFI, to IEC EN 61000-4-6		V	10
Insulation resistance			
Overvoltage category			III
Pollution degree			2
Clearance in air and creepage distances			EN 50178, UL 508, CSA C22.2, No. 142
Insulation resistance			EN 50178
Tools and terminal capacity			
Solid, minimum to maximum		mm^2	easy800 standard: 0.2 – 4 (AWG 22 – 12) easy800-SWD: 0.2 to 1.5 (AWG 24 -16)
Flexible with ferrule, minimum to maximum		mm^2	easy800 standard: 0.2 – 2.5 (AWG 22 – 12) easy800-SWD: 0.2 to 1.5 (AWG 24 -16)
Tools			
easy800 standard			
standard screwdriver		mm	3.5 x 0.8
standard screwdriver		inch	0.14 x 0.03
max. tightening torque		Nm	0.6
easy800-SWD			No tool required, due to push-in terminals

12.3.3 Power supply

12.3.3.1 AC Power supply

		EASY8..-AC-..
Rated operational voltage		
Rated value (sinusoidal) U_e	V AC, (%)	100/110/115/120/230/240 AC (+10/-15)
permissible range	V AC	85 - 264
Frequency, tolerance	Hz (%)	50/60 (± 5)
Input current consumption		
At 115/120 V AC 60 Hz	m A	Normally 70
at 230/240 V AC 50 Hz	m A	Normally 35
Voltage dips, IEC/EN 61131-2	ms	20
Heat dissipation		
At 115/120 V AC	VA	Normally 10
at 230/240 V AC	VA	Normally 10

12.3.3.2 Voltage supply: DC

		EASY8..-DC-.., easy800-SWD
Rated operational voltage		
Rated value U_e	V DC, (%)	24 DC (-15/+20)
permissible range[1]	V DC	20.4 - 28.8
Residual ripple	%	≦ 5
Protection against polarity reversal		yes
Input current		
At rated operating voltage	m A	easy800 standard: 140 EASY802-DC-SWD: 500 EASY806-DC-SWD: 900
Inrush current and duty factor		easy800-SWD: 12.5 A for 6 ms
Voltage dips (IEC/EN 61131-2)	ms	10
Heat dissipation		
At 24 V DC	W	easy800 standard: normally 3.4 EASY802-DC-SWD: max 5 EASY806-DC-SWD: max. 6
Fuse	A	1 to 4 easy800-SWD: ≧ 3

[1] Use power feed modules if the cable length of the SWD line causes excessive voltage drop.

	EASY8..-DC-.., easy800-SWD
Potential isolation	
to AUX[1]	yes
to easyNet[1]	yes
to serial interface	easy800 standard: no (potential isolation in the cable) easy800-SWD: yes
to easyLink[1]	no
to inputs[1]	no
to outputs[1]	easy800 standard: yes easy800-SWD: no
to SWD[1]	no

[1] if present

12.3.3.3 Power supply DC AUX

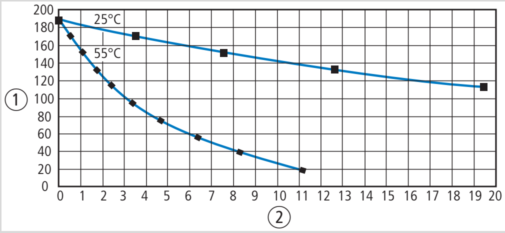
			easy800-SWD
Rated operational voltage			
Rated value U_e		V DC, (%)	24 (-15/+20)
permissible range U_{AUX}		V DC	20.4 - 28.8
Residual ripple		%	≦ 5
Input current[1]	max	A	3 (ICE) 2 (UL)
Rated operational voltage of the 24 V DC stations		V	U_{AUX} - 0.3 V
Protection against polarity reversal			yes
Short-circuit strength			no
Fuse		A	≦ 3 (IEC) external fuse with FAZ Z3 ≦ 2 (UL) external fuse with FAZ B2
Heat dissipation			
At 24 V DC		W	Normally 1
Potential isolation			
to POW power supply, inputs and outputs			yes
to PC interface (COM), easyNet,			yes
to SWD			yes

[1] If contactors with a higher total power consumption are connected, an EU5C-SWD-PF1 or EU5C-SWD-PF2 power feed module must be used.

12.3.4 Central processing unit, real-time clock/timing relay/memory

A double-layer capacitor (super capacitor) is used to back up the supply voltage for the real-time clock.

The full charge of the double layer capacitor is reached if the easy800 device is connected to the power supply for 24 hours.

Backup/accuracy of the real-time clock (see figure) only EASY8..-..-.C.“(Clock) and easy800-SWD		
Back-up time		 ① Backup time (hours) with fully charged double layer capacitor ② Service life (years)
Accuracy of the real-time clock	s/day	Normally ± 2 (± 0.2 h/Year) The accuracy of the real-time clock can vary, depending on the average ambient air temperature, up to ± 5 s/Day (± 0.5 h/Year).
Repetition accuracy of timing relays		
Accuracy of timing relay (from value)	%	± 0.02
Resolution		
Range “S”	ms	5
Range “M:S”	s	1
Range “H:M”	min	1
Memory		
Possible Current paths		256
Possible contacts in series		4
Coil per circuit connection		1
Program memory[1]	KByte	easy800 standard: 8 easy800-SWD: 16
Storage of programs (Zero-voltage safe, i.e. retentive)		FRAM
Write/read cycles memory (minimum)		easy800 standard: 10^{12} easy800-SWD: 10^{14}
Retentive memory		
Size	Byte:	easy800 standard: 200 easy800-SWD: 400
Hours-run counter	Byte:	16

[1] for circuit diagram, function blocks and system parameters

12.3.5 Inputs

12.3.5.1 Digital inputs 115/230 V AC

		EASY819-AC-RC.
Digital inputs 115/230 V AC		
Number		12
Status indicator		LCD display (if provided)
Potential isolation		
from power supply		no
between digital inputs		no
from the outputs		yes
to PC interface (COM), memory card, network NET, easyLink		yes
Rated operating voltage L (sinusoidal)		
at signal ”0”	V AC	0 – 40
at signal ”1”	V AC	79 – 264
Rated frequency	Hz	50 – 60
Input current at signal "1"		
I1 to I6, I9 to I12	m A	10 x 0.25 (at 115 V AC, 60 Hz) 10 x 0.5 (at 230 V AC, 50 Hz)
I7, I8	m A	2 x 4 (at 115 V AC, 60 Hz) 2 x 6 (at 230 V AC, 50 Hz)
Delay time		
Delay time (0 – 1/1 – 0) I1 - I6, I9 - I12		
Debounce ON, 50/60 Hz	ms	80⁄66⅔
Debounce OFF, 50/60 Hz	ms	20⁄16⅔
Delay time I7, I8 (1 – 0)		
Debounce ON, 50/60 Hz	ms	120⁄100
Debounce OFF, 50/60 Hz	ms	40⁄33⅓
Delay time I7, I8 (0 –> 1)		
Debounce ON, 50/60 Hz	ms	80⁄66⅔
Debounce OFF, 50/60 Hz	ms	20⁄16⅔
Max. admissible cable length (per input)		
I1 to I6, I9 to I12		
Debounce ON	m	Normally 100
Debounce OFF	m	Normally 60
I7, I8	m	Normally 100

12.3.5.2 Digital inputs 24 V DC

The transistor inputs I3 and I4 on EASY806-DC-SWD can alternatively also be configured as outputs Q1 and Q2.

			easy800 standard DC version	**EASY806-DC-SWD**
Number of device inputs			12 (I1 - I12)	4 (I1 - I4)
Inputs of the SWD stations			None	83 (I17-I99)
Inputs can be used as analog inputs			4 (I7, I8, I11, I12)	None
Status indicator			LCD display (if provided)	LED
Potential isolation				
from power supply			no	no
between digital inputs			no	no
from the outputs			yes	no
to PC interface (COM), memory card, easyLink			no	yes (if provided)
to easyNet			yes	yes
to AUX			not available	yes
to SWD			not available	no
Rated operational voltage	U_e	V DC	24	24
at signal ”0“	U_e	V DC	< 5 (I1 – I6, I9, I10) < 8 (I7, I8, I11, I12)	< 5 (I1 – I4)
at signal ”1“	U_e	V DC	> 15.0 (I1 – I6, I9, I10), > 8.0 (I7, I8, I11, I12)	> 15.0 (I1 – I4)
Input current at signal ”1“				
		m A	3.3 (at 24 V DC) (I1 to I6, I9, I10)	3.9 (to 24 V DC) (I1-I4)
		m A	2.2 (at 24 V DC) (I7, I8, I11, I12)	
Hardware delay time ”0“ -> „1“				
Debounce ON		ms	20	20
Debounce OFF		ms	Normally 0.025 (I1 – I4) normally 0.25 (I5, I6, I9, I10) normally 0.15 (I7, I8, I11, I12)	max. 0.02
Hardware delay time ”1“ -> ”0“				
Debounce ON		ms	20	20
Debounce OFF		ms	Normally 0.025 (I1 - I4) normally 0.25 (I5, I6, I9, I10), normally 0.15 (I7, I8, I11, I12)	max. 0.02 (I1-I4)
Cable length (unscreened)		m	100	100

		easy800 standard DC version	EASY806-DC-SWD
Rapid counter inputs			
Number		4 (I1, I2, I3, I4)	4 (I1, I2, I3, I4)
Cable length, shielded	m	20	20
High-speed up and down counter			
Counter frequency	kHz	≦ 5	≦ 5
Pulse shape		Square	Square
Pulse Pause Ratio		1:1	1:1
Frequency counter			
Quantity		4 (I1, I2, I3, I4)	4 (I1, I2, I3, I4)
Counter frequency	kHz	≦ 5	≦ 5
Pulse shape		Square	Square
Pulse Pause Ratio		1:1	1:1
Incremental counter			
Number		2 (I1 + I2, I3 + I4)	2 (I1 + I2, I3 + I4)
Counter frequency	kHz	≦ 3	≦ 5
Pulse shape		Square	Square
Signal offset		90°	90°
Pulse Pause Ratio		1:1	1:1

12.3.5.3 Analog inputs

			easy800 standard DC version
Number			4 (I7, I8, I11, I12)
Potential isolation			
from power supply			no
to the digital inputs			no
from the outputs			yes
to PC interface, memory card, network NET, easyLink			yes
Input type			DC voltage
Signal range		V DC	0 – 10
Resolution, analog		V	0.01
Resolution, digital		V	0.01
Resolution, digital		Bit	10 (value 0 – 1023)
Input impedance		kΩ	11.2
Accuracy of actual value			
two easy800 standard devices			± 3% ± 0.12 V
Within a single device			± 2% ± 0.12 V
Conversion time, analog/digital			
Debounce ON:		ms	20
Debounce OFF		ms	each CPU cycle
Input current		m A	< 1
Cable length screened		m	< 30

12.3.6 Outputs

12.3.6.1 Relay outputs

			EASY8..-R...
Number			6
Outputs in groups of			1
Parallel switching of outputs to increase performance			Not permissible
Protection of an output relay			Miniature circuit-breaker B16 or fuse 8 A (slow).
Potential isolation			
from power supply			yes
from the inputs			yes
to PC interface, memory card, network NET, easyLink			yes
Safe isolation		V AC	300
Basic isolation		V AC	600
Lifespan, mechanical		Operations	10 x 10^6
Contacts			
conventional thermal current (UL/CSA)		A	8 (10)
Recommended for load: 12 V AC/DC		m A	> 500
Short-circuit proof p.f. = 1, characteristic B16 at 600 A		A	16
Short-circuit proof cos φ = 0.5 - 0.7; characteristic B16 at 900 A		A	16
Rated impulse withstand voltage U_{imp} contact to coil		kV	6
Rated operational voltage U_e		V AC	250
Rated insulation voltage U_i		V AC	250
Safe isolation to EN 50178 between coil and contact		V AC	300
Safe isolation to EN 50178 between two contacts		V AC	300
Making capacity, IEC 60947			
AC—-15, 250 V AC, 3 A (600 ops./h)		Operations	300000
DC-13, L/R ≦ 150 ms, 24 V DC, 1 A (500 S/h)		Operations	200000
Breaking capacity, IEC 60947			
AC-15, 250 V AC, 3 A (600 ops./h)		Switch operations	300000
DC-13, L/R ≦ 150 ms, 24 V DC, 1 A (500 S/h)		Switch operations	200000

			EASY8..-R...
Filament bulb load			
1000 W at 230/240 V AC		Switch operations	25000
500 W at 115/120 V AC		Switch operations	25000
Fluorescent lamp load			
Fluorescent lamp load 10 x 58 W at 230/240 V AC			
With upstream electrical device		Switch operations	25000
Uncompensated		Switch operations	25000
Fluorescent lamp load 1 x 58 W at 230/240 V AC, conventionally compensated		Switch operations	25000
Switching Frequency			
Mechanical operations		Switch operations	10x 10^6
Switching Frequency		Hz	10
Resistive load/lamp load		Hz	2
Inductive load		Hz	0.5
UL/CSA			
Continuous current at 240 V AC		A	10
AC			
Control circuit rating codes (utilization category)			B 300 Light Pilot Duty
Max. rated operational voltage		V AC	300
max. thermal continuous current cos $\varphi = 1$ at B 300		A	5
max. make/break cos $\varphi \neq$ capacity 1 at B 300		VA	3600/360
DC			
Control circuit rating codes (utilization category)			R 300 Light Pilot Duty
Max. rated operational voltage		V DC	300
Max. thermal continuous current at R 300		A	1
Max. make/break capacity at R 300		VA	28/28

12.3.6.2 Transistor outputs

The transistor outputs on easy800-SWD can alternatively also be configured as inputs.

			EASY8..-DC-T..., easy800-SWD
Number of device outputs			easy800 standard: 8 (Q1-Q8) easy800-SWD: 2 (Q1/Q2)
Outputs of SWD stations			easy800 standard: none easy800-SWD: 83 (Q17-Q99)
Power supply of the transistor outputs on easy800 standard:			
Rated operational voltage	Ue	V DC	24
permissible range	Ue	V DC	20.4 – 28.8
Residual ripple		%	≦ 5
Supply current			
at signal "0"	typ./max.	m A	18/32
at signal "1"	typ./max.	m A	24/44
Protection against polarity reversal			yes
Caution! If voltage is applied to the outputs of easy800 standard when the polarity of the supply voltage is reversed, this will result in a short-circuit.			
Power supply of the transistor outputs on easy800-SWD:			Corresponds to the DC power supply of the device (→ section "12.3.3.2 Voltage supply: DC", page 346)
Potential isolation			
from power supply			EASY8...-DC-T...: yes easy800-SWD: no
from the inputs			EASY8..-DC-T...: yes easy800-SWD: no
to the PC interface (COM), memory card, easyNet, easyLink, AUX			yes
to SWD			easy800-SWD: no
Rated operational current at signal "1", maximum	Ie	A	easy800 standard: 0.5 easy800-SWD: 0.1
Lamp load without R_V		W	easy800 standard: 3 (Q1 – Q4), 5 (Q5 – Q8) easy800-SWD: 1.2
Residual current at signal "0" per channel		m A	< 0.1
Max. output voltage			
With condition "0" at external load < 10 MΩ		V	2.5
on 1 signal at easy800 standard: I_e = 0.5 A easy800-SWD: I_e = 0.1 A		V	easy800 standard: U = U_e-1 V easy800-SWD: U = U_e-2 V

		EASY8..-DC-T..., easy800-SWD
Short-circuit protective device		
Electronic (evaluation possible via diagnostic input I16)		easy800 standard: yes (Q1 – Q4) easy800-SWD: yes (Q1/Q2)
thermal (evaluation possible via diagnostic input I15)		easy800 standard: yes (Q5 – Q8) easy800-SWD: no
Short-circuit tripping current for $R_a \leqq 10$ mΩ (depending on number of active channels and their load)	A	easy800 standard: $0.7 \leqq I_e \leqq 2$ per output easy800-SWD: 0.15 to 0.35 per output
Maximum, total short-circuit current	A	easy800 standard: 8
Peak short-circuit current	A	easy800 standard:16
in the event of short-circuit on attempted restart of device after 10 s	A	easy800-SWD: 10A/80 ms easy800-SWD: 10A/20ms
Thermal cutout		easy800 standard: yes easy800-SWD: no
Max. switching frequency of resistive load		depending on cycle time
Parallel wiring capability of outputs		easy800 standard: yes easy800-SWD: no
with resistive load, with inductive load with external suppressor circuit (see section Connecting transistor outputs), with combination within a group		Group 1: Q1 to Q4 Group 2: Q5 - Q8
Number of outputs (max.)		4
Max. total current Caution! Outputs must be actuated simultaneously and for the same time duration.	A	2
Status display of the outputs		easy800 standard: LCD display (if provided) easy800-SWD: LED

Inductive load without external suppressor circuit

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

$$T_{0.95} \approx 3\ T_{0.65} = 3\ \frac{L}{R}$$

easy800 standard:

Utilization categories in groups Q1 to Q4, Q5 to Q8

$T_{0.95}$ = 1 ms
R = 48 Ω
L = 16 mH
	Utilization factor per group g =		0.25

	Relative duty factor	%	100

	Max. switching frequency
	f = 0.5 Hz
	Max. duty factor
	DF = 50 %
		Switching operations/h	1 500

DC13
$T_{0.95}$ = 72 ms
R = 48 Ω
L = 1.15 H
	Utilization factor g =		0.25

	Relative duty factor	%	100

	Max. switching frequency
	f = 0.5 Hz
	Max. duty factor
	DF = 50 %
		Switching operations/h	1500

Other inductive loads:

$T_{0.95}$ = 15 ms
R = 48 Ω
L = 0.24 H
	Utilization factor g =		0.25

	Relative duty factor	%	100

	Max. switching frequency
	f = 0.5 Hz
	Max. duty factor
	DF = 50 %
		Switch operations/h	1500

Inductive loading with external suppressor circuit for each load (→ section “2.6.2 Connecting transistor outputs”, page 48)			
	Utilization factor g =		1
	Relative duty factor	%	100
	Max. switching frequency Max. duty factor	Switch operations/h	Depending on the suppressor circuit

easy800-SWD:

(according to EN 60947-5-1) DC13, DC14 $T_{0.95}$ = 15 ms R = 240 Ω L = 1.15 H	Simultaneity factor g		1
	Relative duty factor	%	100
	Max. switching frequency at maximum duty factor DF = 50 %	Hz	10

12.3.6.3 Analog output

			EASY821-DC-…, EASY822-DC-…
Quantity			1
Potential isolation			
from power supply			no
to the digital inputs			no
to the digital inputs			yes
to PC interface, memory card, network NET, easyLink			yes
Output type			DC voltage
Signal range		V DC	0 – 10
Max. output current		A	0.01
Shunt resistor			1 kΩ
Overload and short-circuit protection			yes
Resolution, analog		V DC	0.01
Resolution, digital		Bit	10, (value: 0 – 1023)
Recovery time		µs	100
Accuracy			
-25 °C – +55 °C		%	2
25C		%	1
Conversion time, analog/digital		ms	each CPU cycle

12.3.7 NET easynetwork

Number of stations		max. 8
max. bus lengths/at baud rate[1]	m/Kbaud	6/1000 25/500 40/250 125/125 300/50 700/20 1000/10
Potential isolation		
To power supply (POW), inputs, outputs, easyLink, PC interface, memory module, SWD, AUX		Yes
Bus termination (→ accessories)		
First and last station		Yes
Plug connector (→ accessories)	poles	8
Type		RJ45
Cable cross-sections, with cable lengths and cable resistance/m		
Cross-section up to 1000, < 16 mΩ/m	mm^2 (AWG)	1.5 (16)
Cross-section up to 600, < 26 mΩ/m	mm^2 (AWG)	0.75 to 0.8 (18)
Cross-section up to 600, < 26 mΩ/m	mm^2 (AWG)	0.5 to 0.6 (20, 19)
Cross-section up to 400 m, < 40 mΩ/m	mm^2 (AWG)	0.34 to 0.5 (22, 21, 20)
Cross-section up to 250 m, < 60 mΩ/m	mm^2 (AWG)	0.25 to 0.34 (23, 22)
Cross-section up to 175 m, < 70 mΩ/m	mm^2 (AWG)	0.13 (26)
Cross-section up to 40 m, < 140 mΩ/m	mm^2 (AWG)	1.5 (16)

[1] Bus lengths above 40 m can only be achieved with cables with reinforced cross-section and connection adapter.

1 baud = 1 bit/s

12.3.8 SmartWire-DT

SWD power supply (generated from the POW supply voltage)			**easy800-SWD**
Rated operational voltage	U_e	V	14.5 ± 3 %
Current[1)]	I_{max}	A	EASY802-DC-SWD: 0.4 EASY806-DC-SWD: 0.7
Short-circuit strength			yes
Potential isolation			
from power supply, POW, inputs, outputs			no
to easyNet			yes
to the PC interface (COM)			yes
SWD-line			
Number of SWD stations			max. 99
Maximum length of SWD line		m	max. 600
Baud Rate		kBd	125/250
Physics of the data cable			RS485
Address setting			Automatically via Configuration button
Status display			Multi-color SWD-LED: orange/green/red Dual colored Config LED: green/red
Connections			Plug, 8-pole
Plug connectors			Blade terminal SWD4-8MF2
Bus termination			Integrated in the device The end of the SWD line must be terminated with: SWD-RC8-10

1) If SWD components with a higher total power consumption are connected, an EU5C-SWD-PF2 power feed module must be used.

12.4 Inputs/outputs of the expansion units

Part no.	Inputs							Outputs			Note
	Digital		Analog					Relays	Transistor	Analog	
				number of which can be used as …							
	24 V DC	100-240 V AC	Total	0 - 10 V, 10 Bit	4 - 20 mA, 10 Bit	Pt100, -50 … 200 °C, 10 Bit	Digital				
EASY202-RE								2			common[1]
EASY410-DC-RE	6							4			
EASY410-DC-TE	6								4		
EASY406-DC-ME[2]	1		2	2[3]	2	2	2[4]		2	1	up to 2 analog inputs simultaneously, 4 diagnostic bits (R13…R16)
EASY411-DC-ME[2]	1		6	2[3]	2	2	2[4]		2	2	up to 6 analog inputs simultaneously, 4 diagnostic bits (R13…R16)
EASY618-AC-RE		12						6			
EASY618-DC-RE	12							6			
EASY620-DC-TE	12								8		

[1] Common supply of both relays

[2] can be used for basic device from device version 02

[3] Can also be used as digital inputs R2/R3

[4] Digital inputs R2/R3 for evaluating the 0 - 10 V voltage inputs

Special expansion units for connecting to other bus systems are shown in the latest main catalog.

12.5 Possible combinations of basic devices and expansion units

The following table shows the possible combinations of basic devices and expansion units. Possible combinations are marked with a reference to the description page.

easy800-SWD devices do not have an easyLink interface and cannot therefore connect any expansion units. The SWD line can be expanded by connecting additional I/O module stations.

	easy800											
	EASY8..-DC-...						**EASY819-AC-R..**					
	Digital				**Analog**		**Digital**				**Analog**	
	Inputs		**Outputs**		**Inputs**	**Outputs**	**Inputs**		**Outputs**		**Inputs**	**Outputs**
	100 V - 240 V AC	**24 V DC**	**Transistor**	**Relays**			**100 V - 240 V AC**	**24 V DC**	**Transistor**	**Relays**		
Basic device												
easy800	-	✓ (p. 35)	✓ (p. 49)	✓ (p. 46)	✓ (p. 40)	✓ (p.p. 51)	✓ (p. 31)	-	-	✓ (p. 46)	-	-
Expansion Units												
EASY618-AC-RE	✓ (p. 31)	-	-	✓ (p. 47)	-	-	✓ (p. 31)	-	-	✓ (p. 47)	-	-
EASY618-DC-RE	-	✓ (p. 38)	-	✓ (p. 47)	-	-	-	✓ (p. 39)	-	✓ (p. 47)	-	-
EASY620-DC-TE	-	✓ (p. 38)	✓ (p. 50)	-	-	-	-	✓ (p. 39)	✓ (p. 50)	-	-	-
EASY410-DC-TE	-	✓ (p. 38)	✓ (p. 50)	-	-	-	-	✓ (p. 39)	✓ (p. 50)	-	-	-
EASY410-DC-RE	-	✓ (p. 38)	-	✓ (p. 47)	-	-	-	✓ (p.39)	-	✓ (p. 47)	-	-
EASY406-DC-ME	-	✓ (p. 39)	✓ (p.50)	-	✓ (p. 42)	✓ (p. 51)	-	✓ (p. 39)	✓ (p. 50)	-	✓ (p. 42)	✓ (p. 51)
EASY411-DC-ME	-	✓ (p. 39)	✓ (p. 50)	-	✓ (p. 42)	✓ (p. 51)	-	✓ (p. 39)	✓ (p. 50)	-	✓ (p. 42)	✓ (p. 51)
EASY202-RE	-	-	-	✓ (p. 47)	-	-	-	-	-	✓ (p. 47)	-	-

12.6 Compatibility of SWD stations

Table 26: List of types that are compatible with one another

Part no.	Description	Allowed replacement type 1	Allowed replacement type 2
RMQ function elements (front mount)			
M22-SWD-K11	Function element, 2 pos. front	M22-SWD-K22	
M22-SWD-K22	Function element, 3 pos. front	-	
M22-SWD-LED-W	Function element, LED W front	M22-SWD-K11LED-W	M22-SWD-K22LED-W
M22-SWD-K11LED-W	Function element, 2 pos., LED-W, front	M22-SWD-K22LED-W	
M22-SWD-K22LED-W	Function element, 3 pos., LED-W, front	-	
M22-SWD-LED-R	Function element, LED-R front	M22-SWD-K11LED-R	M22-SWD-K22LED-R
M22-SWD-K11LED-R	Function element, 2 pos., LED-R, front	M22-SWD-K22LED-R	
M22-SWD-K22LED-R	Function element, 3 pos., LED-R, front	-	
M22-SWD-LED-G	Function element, LED-G front	M22-SWD-K11LED-G	M22-SWD-K22LED-G
M22-SWD-K11LED-G	Function element, 2 pos., LED-G, front	M22-SWD-K22LED-G	
M22-SWD-K22LED-G	Function element, 3 pos., LED-G, front	-	
M22-SWD-LED-B	Function element, LED-B front	M22-SWD-K11LED-B	M22-SWD-K22LED-B
M22-SWD-K11LED-B	Function element, 2 pos., LED-B, front	M22-SWD-K22LED-B	
M22-SWD-K22LED-B	Function element, 3 pos., LED-B, front	-	

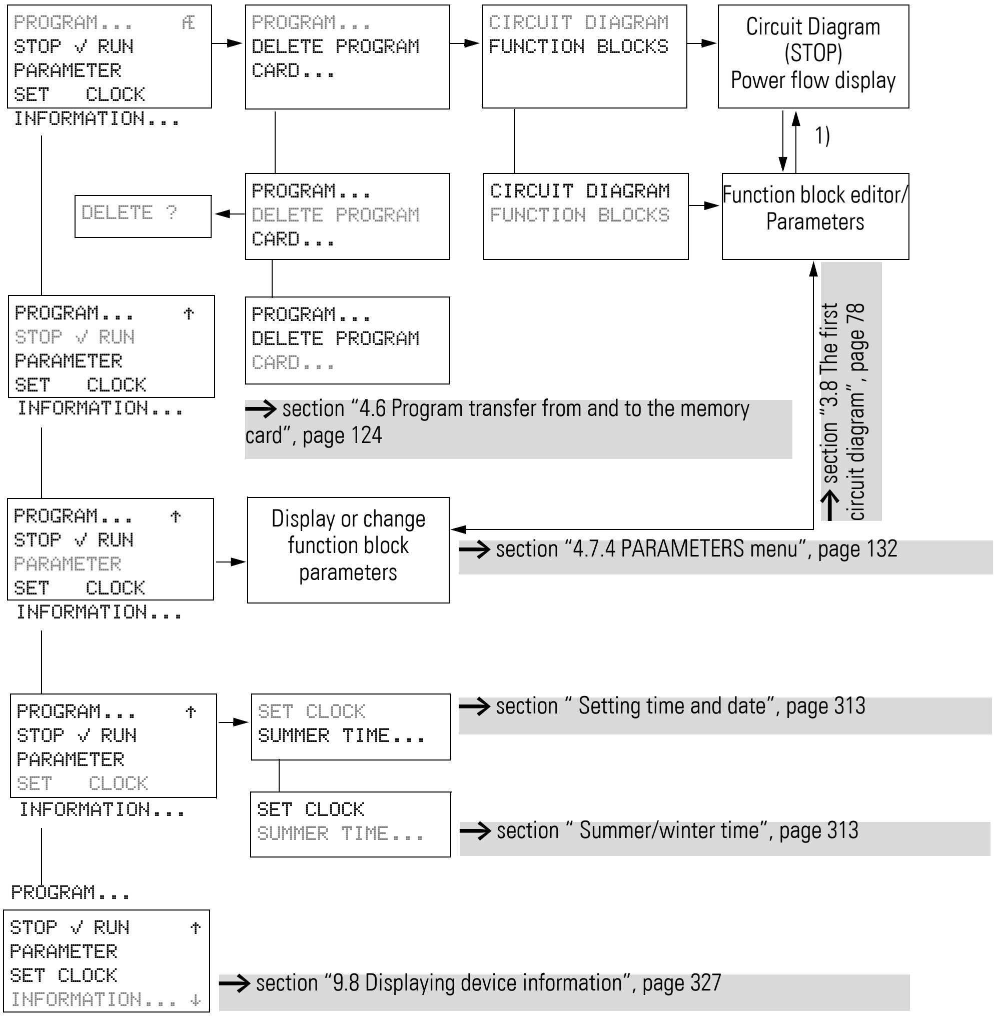
Part no.	Description	Allowed replacement type 1	Allowed replacement type 2
RMQ function elements (base fixing)			
M22-SWD-KC11	Function element, 2 pos., base	M22-SWD-KC22	
M22-SWD-KC22	Function element, 3 pos., base	-	
M22-SWD-LEDC-W	Function element, LED-W, base	M22-SWD-K11LEDC-W	M22-SWD-K22LEDC-W
M22-SWD-K11LEDC-W	Function element, 2 pos., LED-W, base	M22-SWD-K22LEDC-W	
M22-SWD-K22LEDC-W	Function element, 3 pos., LED-W, base	-	
M22-SWD-LEDC-R	Function element, LED-R, base	M22-SWD-K11LEDC-R	M22-SWD-K22LEDC-R
M22-SWD-K11LEDC-R	Function element, 2 pos., LED-R, base	M22-SWD-K22LEDC-R	
M22-SWD-K22LEDC-R	Function element, 3 pos., LED-R, base	-	
M22-SWD-LEDC-G	Function element, LED-G, base	M22-SWD-K11LEDC-G	M22-SWD-K22LEDC-G
M22-SWD-K11LEDC-G	Function element, 2 pos., LED-G, base	M22-SWD-K22LEDC-G	
M22-SWD-K22LEDC-G	Function element, 3 pos., LED-G, base	-	
M22-SWD-LEDC-B	Function element, LED-B, base	M22-SWD-K11LEDC-B	M22-SWD-K22LEDC-B
M22-SWD-K11LEDC-B	Function element, 2 pos., LED-B, base	M22-SWD-K22LEDC-B	
M22-SWD-K22LEDC-B	Function element, 3 pos., LED-B, base	-	

Part no.	Description	Allowed replacement type 1	Allowed replacement type 2
DILM/MSC function elements			
DIL-SWD-32-001	DIL/MSC	DIL-SWD-32-002	
DIL-SWD-32-002	DIL/MSC, Manual/Auto	-	
PKE-SWD-32			
I/O function elements			
EU5E-SWD-4DX	Digital module, 4 inputs		
EU5E-SWD-8DX	Digital module, 8 inputs	-	
EU5E-SWD-4D4D	Digital module, 4 inputs, 4 outputs	-	
EU5E-SWD-4D2R	Digital module, 4 inputs, 2 outputs	-	
EU5E-SWD-X8D	Digital module, 8 outputs		
EU5E-SWD-4AX	Analog module 4 inputs		
EU5E-SWD-2A2A	Analog module 2 inputs, 2 outputs		
EU5E-SWD-4PT	Analog module, 4 temperature inputs		

12.7 Menu structures

12.7.1 Main menu without activated password protection

▶ You access the main menu by pressing

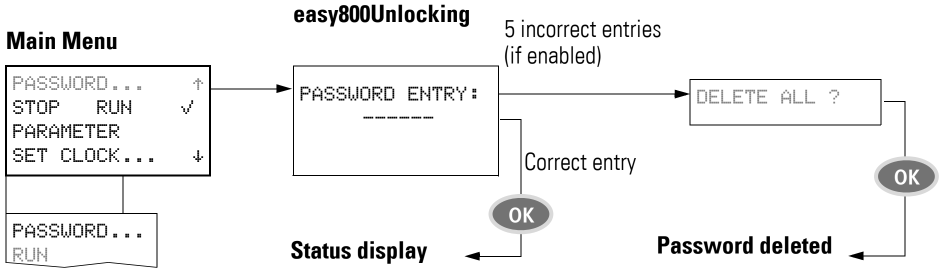

PROGRAM... Æ
STOP √ RUN
PARAMETER
SET CLOCK
INFORMATION...
PROGRAM...
DELETE PROGRAM
CARD...
CIRCUIT DIAGRAM
FUNCTION BLOCKS
Circuit Diagram (STOP) Power flow display
1)
DELETE ?
Function block editor/ Parameters
section "3.8 The first circuit diagram", page 78
section "4.6 Program transfer from and to the memory card", page 124
Display or change function block parameters
section "4.7.4 PARAMETERS menu", page 132
SET CLOCK
SUMMER TIME...
section " Setting time and date", page 313
section " Summer/winter time", page 313
section "9.8 Displaying device information", page 327

1) Change only on initial call of the function block.

12.7.2 Main menu with activated password protection

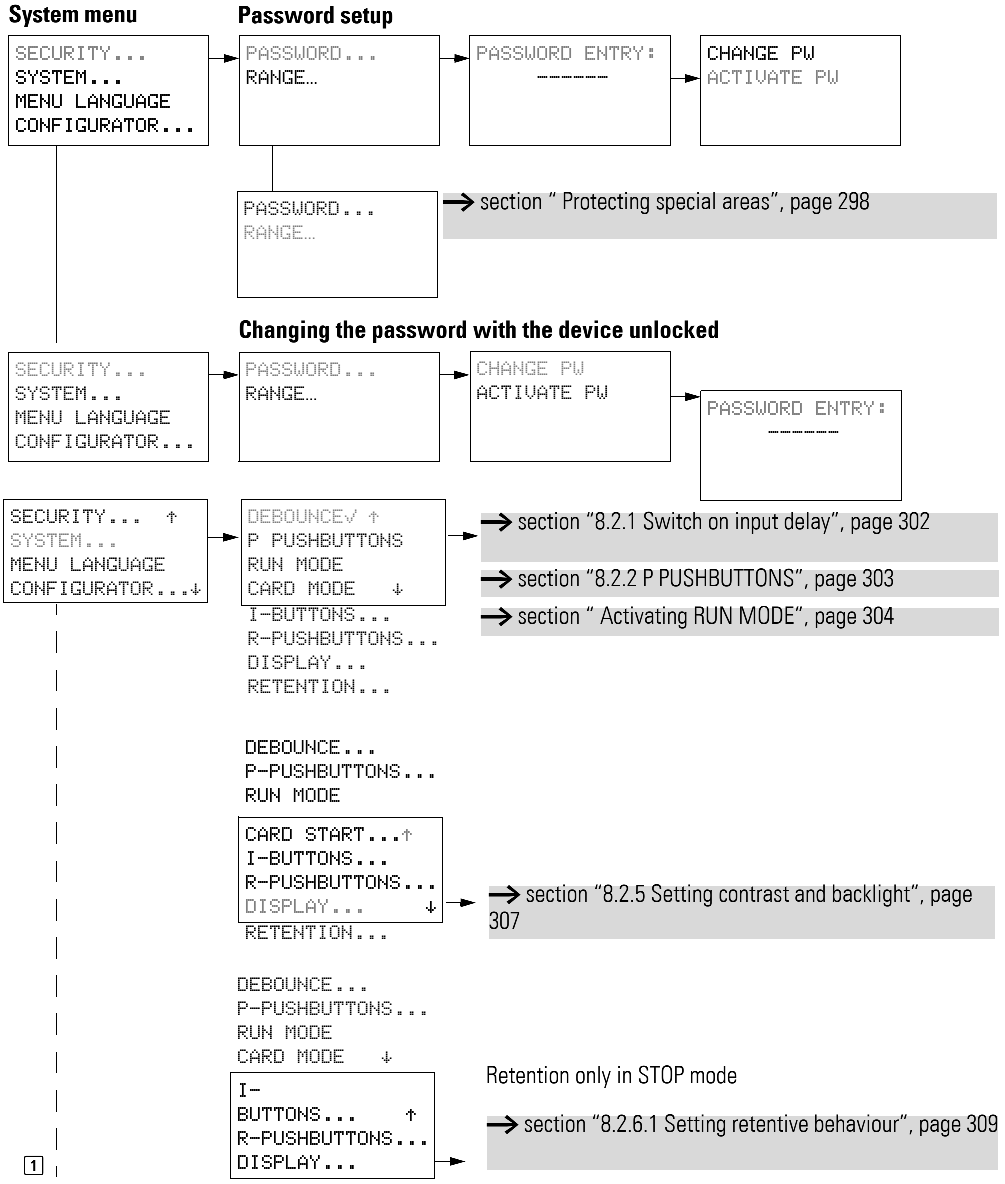
When the password protection is activated and the easy800 is locked, you are guided to the PASSWORD menu to unlock the device.

12.7.2.1 easy800Unlocking



12.7.3 System menu

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



→ section " Protecting special areas", page 298

→ section "8.2.1 Switch on input delay", page 302

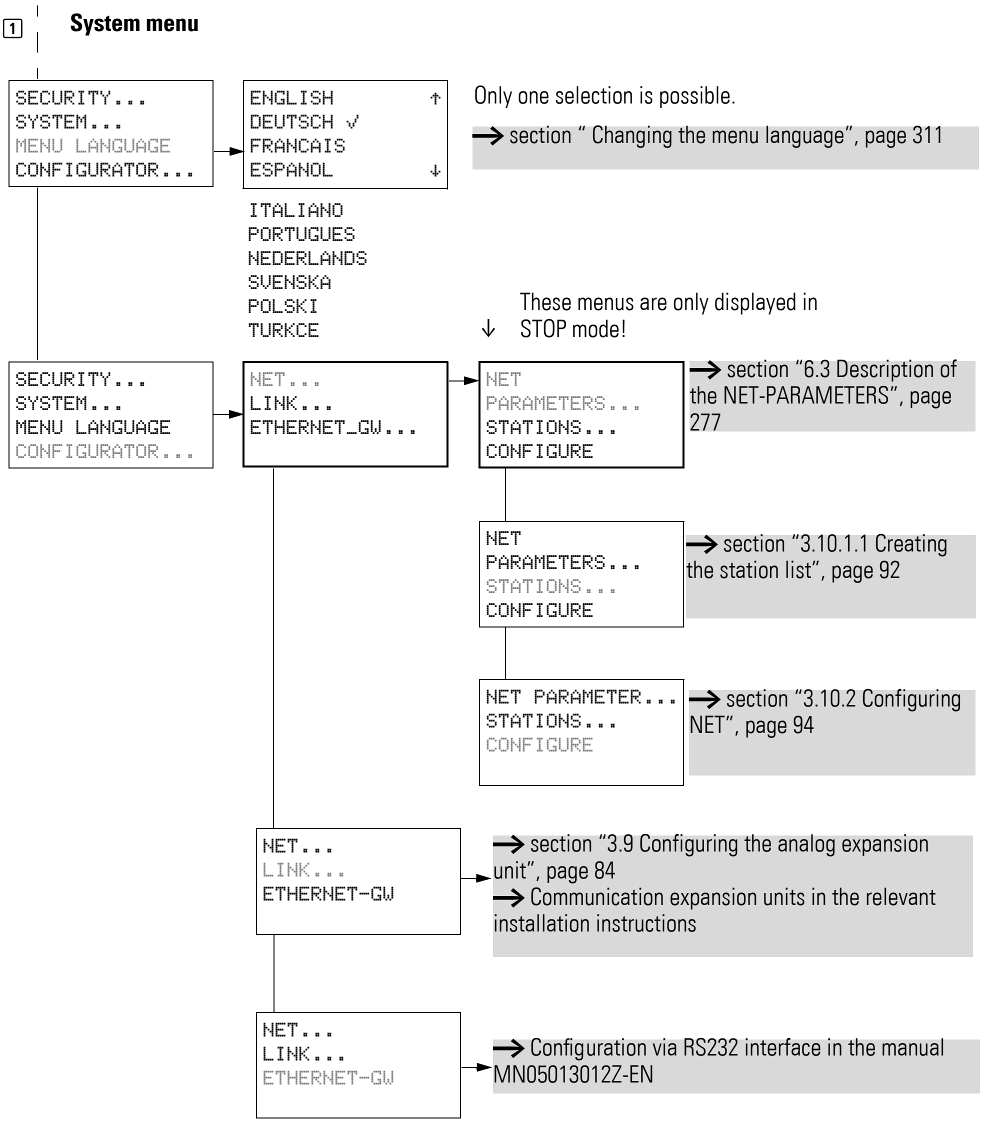
→ section "8.2.2 P PUSHBUTTONS", page 303

→ section " Activating RUN MODE", page 304

→ section "8.2.5 Setting contrast and backlight", page 307

Retention only in STOP mode

→ section "8.2.6.1 Setting retentive behaviour", page 309



Only one selection is possible.

→ section " Changing the menu language", page 311

↓ These menus are only displayed in STOP mode!

→ section "6.3 Description of the NET-PARAMETERS", page 277

→ section "3.10.1.1 Creating the station list", page 92

→ section "3.10.2 Configuring NET", page 94

→ section "3.9 Configuring the analog expansion unit", page 84

→ Communication expansion units in the relevant installation instructions

→ Configuration via RS232 interface in the manual MN05013012Z-EN

12.8 List of function blocks

Function block	Meaning of abbreviation	Function block designation	Page	From dev. vers. no.
A	Analog value comparator	Analog value comparator	136	01
AR	Arithmetic functions	Arithmetic functions	140	01
BC	Block Compare	Data block comparator	143	04
BT	Block Transfer	Data block transfer	149	04
BV	Boolean operation	BV - Boolean operation	159	01
C	Counter	Counters	162	01
CF	Counter Frequency	Frequency counter	166	01
CH	Counter High Speed	High-speed counters	169	01
CI	Counter fast Incremental Value Encoder	High-speed incremental value counter	174	01
CP	Comparators	Comparator	179	01
D	Display	Text function block	181	01
DB	Data Block	Data function block	185	01
DC	DDC controller (direct Digital Control)	PID controller	187	04
FT	Filter	PT1 signal smoothing filter	192	04
GT	GET	GET (Value from NET)	196	01
HW	Hora$_{(lat)}$ Week	7-day time switch	198	01
HY	Hora$_{(lat)}$ Year	Year time switch	203	01
JC	Jump Conditional	Conditional jump	211	07
LB	Label	Jump label	213	07
LS	Linear Scaling	Value scaling	214	04
MR	Master Reset	Master reset	218	01
MX	Multiplexer	Data multiplexer	220	07
NC	Numeric Coding	Numerical converter	223	04
OT	Operating Time	Hours-run counter	227	01
PO	Pulse Output	Pulse output	228	07
PT	PUT	PUT (Value on NET)	235	01
PW	Pulse Width Modulation	Pulse width modulation	237	04
SC	Synchronize Clocks	Synchronize clock via NET	241	01
SP	Serial Protocol	Serial protocol	243	07
SR	Shift Register	Shift register	247	07
ST	Set Time	Set cycle time	253	04
T	Timing relays	Timing relay	255	01
TB	Table function	Table function	264	07
VC	Value Capsuling	Value limitation	268	04

12.8.1 Function block coils

The following table is a general overview of the function block contacts. A detailed overview of all contacts used in the circuit diagram is provided in → Section "12.10 Memory requirement" on Page 378.

Table 27: General overview of the function block coils

Coil	Meaning of abbreviation	Description
BD	Backward Direction	Bit data input, shift direction backward
BP	Backward Pulse	Trigger coil Clock input, shift direction backward
C_	Count input	Counter input
D_	Direction input	Count direction
ED	Enable Differential Ratio	Activate differential component
EI	Enable Integral Ratio	Activate integral component
EN	Enable	Enable function block
EP	Enable Proportional component	Activate proportional component
FD	Forward Direction	Bit data input, forwards shift direction
FP	Forward Pulse	Trigger coil Clock input, forward shift direction
RE	Reset	Reset actual value to zero
RF	Read First	Trigger coil On detecting a rising edge (trigger) and EN = 1 the oldest value entered in the table is read out and supplied at output QV (FIFO function)
RL	Read Last	Trigger coil On detecting a rising edge (trigger) and EN = 1 the newest value entered in the table is read out and output at output QV (LIFO function)
SE	Set Enable	Set to a setpoint value
ST	Stop	Stop block processing
T_	Trigger	Trigger coil for starting the timing relay
WP	Word Pulse	Trigger coil On detecting a rising edge (trigger) and EN = 1 the value is entered in the table

12.8.2 Function block contacts

The following table is a general overview of the function block contacts. A detailed overview of all contacts used in the circuit diagram is provided in → Section “12.10 Memory requirement“ on Page 378.

Table 28: General overview of the function block contacts

Contact	Meaning of abbreviation	Description
AC	Active	Status 1, as long as the input value has been buffered but not yet transferred to the easyNet with the PUT function block.
CY	Carry	Status 1, if the value range is exceeded
E1	Error 1	Error 1, dependent on function block
E2	Error 2	Error 2, dependent on function block
E3	Error 3	Error 3, dependent on function block
EQ	Equal	Comparison result, status 1 if values equal.
FB	Fall Below	Status 1, if the actual value is less than or equal to the lower setpoint value
GT	Greater Than	Status 1 if the value at I1 > I2
LI	Limit Indicator	Value range manipulated variable exceeded
LT	Less Than	Status 1 if the value at I1 < I2;
OF	Overflow	Status 1, if the actual value is greater than or equal to the upper setpoint value
Q	Output (Q...)	Switch output
QV	Output Value	Current actual value of the function block (e.g. counter value)
TE	Table Empty	Status 1 if the table is empty
TF	Table Full	Status 1 if the table is full
ZE	Zero	Status 1, if the value of the element input QV is equal to zero

12.9 Contacts and Coils

The following table shows a list of all contacts and coils useable in the standard circuit diagram.

Contact	Coil (* = NET-ID 1-8)	Contact N/O - N/C (* = NET-ID 1-8)	Number	Page
Inputs				
Inputs				
easy800 standard	–	I/Ī	01-12	111
EASY802-DC-SWD	–	I/Ī	17-99	
EASY806-DC-SWD	–	I/Ī	01-04, 17-99	
Inputs I.. (basic device) of other stations via NET				
easy800 standard	–	*I/*Ī	01-12	121
EASY806-DC-SWD	–	*I/*Ī	01-04	

Contact	Coil (* = NET-ID 1-8)	Contact N/O - N/C (* = NET-ID 1-8)	Number	Page
Cursor buttons	–	P/P̅	01-04	117
Inputs easy800 standard (expansion)	–	R/R̅	01-12	111
Inputs easy800 standard (standard expansion) of other stations via NET	–	*R/*R̅	01-12	121
Bit inputs via NET for easy800 standard and EASY806-DC-SWD	–	*RN/*R̅N̅	01-32	121
Diagnostics inputs				
Status of expansion of other station via NET ateasy800 standard	–	*I/*I̅	14	325
Short-circuit/overload on other station via NET				
easy800 standard	–	*I/*I̅	15-16	325
EASY806-DC-SWD	–	*I/*I̅	16	325
Expansion status at easy800 standard	–	I/I̅	14	325
Short-circuit/overload on basic device with transistor output				
easy800 standard	–	I/I̅	15-16	325
EASY806-DC-SWD	–	I/I̅	16	325
Short-circuit/overload on a expansion unit with transistor output at easy800 standard	–	R/R̅	15-16	325
Short-circuit/overload on a expansion unit other NET stations at easy800 standard	–	*R/*R̅	15-16	325
Diagnostic alarms				
easy800 standard	–	ID/I̅D̅	01-32	329
easy800-SWD	–	ID/I̅D̅	01-08	
Outputs				
Q outputs..				
easy800 standard	Q	–	01-08	111
	–	Q/Q̅		
EASY806-DC-SWD	Q	–	01-02, 17-99	49
	–	Q/Q̅		
EASY802-DC-SWD	–	–	17-99	49
	–	–		
Outputs (Expansion) easy800 standard	S	–	01-08	111
	–	S/S̅		
Q outputs.. (basic device) of other NET stations				
easy800 standard	*Q	–	8	121
	–	*Q/*Q̅	01-08	
EASY806-DC-SWD	–	–	2	121
	–	*Q/*Q̅	01-02	
Outputs (expansion) of other NET stations easy800 standard	*S	–	01-08	121
	–	*S/*S̅		
Bit outputs via NET for easy800 standard and EASY806-DC-SWD	*SN	–	01-32	121
	–	*SN/*S̅N̅		

Contact	Coil (* = NET-ID 1-8)	Contact N/O - N/C (* = NET-ID 1-8)	Number	Page
Other coils and contacts				
Markers				
easy800 standard	M	–	01–96	109
	–	M/M̄		
easy800-SWD	M	–	01–128	109
	–	M/M̄		
Jump label	:	–	01–32	247
	–	:		
Function blocks				
Analog value comparator				136
Value overflow (carry)	–	A X CY/Ā X CY	X = 01–32	
Condition fulfilled		A X Q1/Ā X Q1		
Arithmetic functions				140
Value overflow (carry)	–	AR X CY/ĀR̄ X CY	X = 01–32	
Value zero		AR X ZE/ĀR̄ X ZE		
Data block comparator				143
Enable	BC X EN	–	X = 01–32	
Failure: number of elements exceeded	–	BC X E1/B̄C̄ X E1		
Failure: range overlap		BC X E2/B̄C̄ X E2		
Failure: invalid offset		BC X E3/B̄C̄ X E3		
Comparison result		BC X EQ/B̄C̄ X EQ		
Data block transfer				149
Activate (trigger)	BT X T_	–	X = 01–32	
Failure: number of elements exceeded	–	BT X E1/B̄T̄ X E1		
Failure: range overlap		BT X E2/B̄T̄ X E2		
Failure: invalid offset		BT X E3/B̄T̄ X E3		
BV - Boolean operation				159
Value zero	–	BV X ZE/B̄V̄ X ZE	X = 01–32	
Counters				162
Count input	C X C_	–	X = 01–32	
Direction	C X D_			
Reset count value	C X RE			
Set counter value (preset)	C X SE			
Actual value has exceeded counter range (carry)	–	C X CY/C̄ X CY		
Lower setpoint value undershot (Fall below)		C X FB/C̄ X FB		
Upper setpoint value exceeded (overflow)		C X OF/C̄ X OF		
Actual value equal to zero		C X ZE/C̄ X ZE		
Frequency counter				166

Contact	Coil (* = NET-ID 1-8)	Contact N/O - N/C (* = NET-ID 1-8)	Number	Page
Enable counter	CF X EN	–	X = 01-04	
Lower setpoint value undershot (Fall below)	–	CF X FB/$\overline{\text{CF}}$ X FB		
Upper setpoint value exceeded (overflow)		CF X OF/$\overline{\text{CF}}$ X OF		
Actual value equal to zero		CF X ZE/$\overline{\text{CF}}$ X ZE		
High-speed counters				169
Direction	CH X D_	–	X = 01-04	
Enable counter	CH X EN			
Reset count value	CH X RE			
Set counter value (Preset)	CH X SE			
Actual value has exceeded counter range (carry)	–	CH X CY/$\overline{\text{CH}}$ X CY		
Lower setpoint value undershot (Fall below)		CH X FB/$\overline{\text{CH}}$ X FB		
Upper setpoint value exceeded (overflow)		CH X OF/$\overline{\text{CH}}$ X OF		
Actual value equal to zero		CH X ZE/$\overline{\text{CH}}$ X ZE		
Incremental counter				174
Enable counter	CI X EN	–	X = 01-02	
Reset count value	CI X RE			
Set counter value (Preset)	CI X SE			
Actual value has exceeded counter range (carry)	–	CI X CY/$\overline{\text{CI}}$ X CY		
Lower setpoint value undershot (Fall below)		CI X FB/$\overline{\text{CI}}$ X FB		
Upper setpoint value exceeded (overflow)		CI X OF/$\overline{\text{CI}}$ X OF		
Actual value equal to zero		CI X ZE/$\overline{\text{CI}}$ X ZE		
Comparator, for variables / constants				179
Equal to	–	CP X EQ/$\overline{\text{CP}}$ X EQ	X = 01-32	
Greater than		CP X GT/$\overline{\text{CP}}$ X GT		
Less than		CP X LT/$\overline{\text{CP}}$ X LT		
Text output				181
Enable	D X EN	–	X = 01-32	
Active	–	D X Q1/$\overline{\text{D}}$ X Q1		
Data function block				185
Transfer 32-bit value (trigger)	DB X T_	–	X = 01-32	
Block is active	–	DB X Q1/$\overline{\text{DB}}$ X Q1		
PID controller				187
Activate D component	DC X ED	–	X = 01-32	
Activate I component	DC X EI			
Enable	DC X EN			
Activate P component	DC X EP			
Accept manual manipulated variable	DC X SE			
Value range manipulated variable exceeded	–	DC X LI/$\overline{\text{DC}}$ X LI		

Contact	Coil (* = NET-ID 1-8)	Contact N/O - N/C (* = NET-ID 1-8)	Number	Page
Signal smoothing filter				192
Enable	FT X EN	–	X = 01-32	
Get value from NET (GET)				196
New transferred value present	–	GT X Q1/$\overline{\text{GT}}$ X Q1	X = 01-32	
7-day time switch				198
Switch on condition fulfilled	–	HW X Q1/$\overline{\text{HW}}$ X Q1	X = 01-32	
Year time switch				203
Switch on condition fulfilled	–	HY X Q1/$\overline{\text{HY}}$ X Q1	X = 01-32	
Conditional jump				211
Enable	JC X EN	–	X = 01-32	
Fault: No corresponding jump label present or backward jump	–	JC X E1/$\overline{\text{JC}}$ X E1		
Jump Label (LaBel) Passive jump destination for JC				213
Value scaling				214
Enable	LS X EN	LS X EN	X = 01-32	
Master reset				218
Activate (trigger)	MR X T_	–	X = 01-32	
Set all outputs and markers to zero state	–	MR X Q1/$\overline{\text{MR}}$ X Q1		
Data multiplexer (easy800 and MFD-CP10...)				220
Enable	MX X EN	–	X = 01-32	
Fault: Incorrect parameter setting, 0 > K > 7	–	MX X E1/$\overline{\text{MX}}$ X E1		
Numerical converter				223
Enable	NC X EN	–	X = 01-32	
Operating hours counter				227
Enable	OT X EN	–	X = 01-04	
Reset	OT X RE			
Value overflow (carry)	–	OT X CY/$\overline{\text{OT}}$ X CY		
Set time reached		OT X Q1/$\overline{\text{OT}}$ X Q1		
Pulse output				228
Enable	PO X EN	–	X = 01, 02	
Start positioning job	PO X S_			
Activate jog mode	PO X TP			
Positioning job active	PO X BR			
Positioning job active	–	PO X AC/$\overline{\text{PO}}$ X AC		
Fault: Incorrect parameter setting, FO < FS or PF < FS.		PO X E1/$\overline{\text{PO}}$ X E1		
Put value on NET (PUT)				235
Activate (trigger)	PT X T_	–	X = 01-32	
Enable active	–	PT X Q1/$\overline{\text{PT}}$ X Q1		
Pulse width modulation				237

Contact	Coil (* = NET-ID 1-8)	Contact N/O - N/C (* = NET-ID 1-8)	Number	Page
Enable	PW X EN	–	X = 01, 02	
Fault: Minimum on/off time exceeded	–	PW X E1/$\overline{PW}$ X E1		
Synchronize clock via NET (Set clock)				241
Trigger	SC X T_	SC X T_	X = 01	
Trigger acknowledgement	SC X Q1 $\overline{SC}$ X Q1	SC X Q1/$\overline{SC}$ X Q1		
Serial protocol, send data via serial interface				243
Enable	SP X EN	SP X EN	X = 01-32	
Trigger	SP X T_	SP X T_		
Send job active	SP X AC $\overline{SP}$ X AC	SP X AC/$\overline{SP}$ X AC		
Group fault output	SP X E1 $\overline{SP}$ X E1	SP X E1/$\overline{SP}$ X E1		
Shift register				247
Enable	SR X EN	–	X = 01-32	
Shift register	–	SR X Q1-8/$\overline{SR}$ X Q1-8		
Set cycle time				253
Enable	ST X EN	–	X = 01	
Timing relay				255
Enable	T X EN	–	X = 01-32	
Timing relay reset	T X RE			
Timing relay Stop	T X ST			
Timing Relay-Switching contact	–	T X Q1/$\overline{T}$ X Q1		
Table function				264
Enable	TB X EN	–	X = 01-32	
Table empty	–	TB X TE/$\overline{TB}$ X TE		
Table full		TB X TF/$\overline{TB}$ X TF		
Value limitation				268
Enable	VC X EN	–	X = 01-32	

12.9.1 Function block inputs (constants, operands)

Input	Meaning of abbreviation	Description
BF	Standard	Frequency change in braking phase
F1	Factor 1	Gain factor for I1 (I1 = F1 value)
F2	Factor 2	Gain factor for I2 (I2 = F2 value)
FO	Frequency Operation	Operating frequency
FS	Frequency Start	Start frequency
HY	HYsteresis	Switching hysteresis for value I2 (Value HY applies to positive and negative hysteresis.)
I1…I8	Input 1 to Input 8	1. to 8. input word
KP	Channel	Channel number Value range: 0...7
KP	Standard	Proportional gain
ME	Minimum duty factor	Minimum duty factor
MV	Manual Value	Manual manipulated variable
NO	Numbers Of elements	Number of elements
OS	OffSet	Offset for the value I1
PI	Standard	Number of steps in Jog mode
PF	Standard	Frequency in jog mode
PD	Period of Duration	Period duration
RF	Standard	Frequency change in acceleration phase
SH	Setpoint High	Upper limit value
SL	Setpoint Low	Lower limit value
SV	Set Value	Defined actual value (Preset)
TC	Standard	Scan Time
TG	Standard	Recovery Time
TN	Standard	Derivate action time
Tv	Standard	Reset time
X1	X1, interpolation point 1 abscissa	Lower value of source range
X2	Interpolation point 2 abscissa	Upper value of source range
Y1	Interpolation point 1 ordinate	Lower value of target range
Y2	Interpolation point 2 ordinate	Upper value of target range

12.9.2 Function block output (operands)

Output	Meaning of abbreviation	Description
D1…D8	Data 1…8	Output values
QN	Output Number	Number of table entries present, 0…16
QV	Output Value	Output value

12.9.3 Other operands

Other operands	Meaning of abbreviation	Description
MB	Marker byte	8-Bit value
IA	Input Analog	Analog input (if available on device!)
MW	Merker Word	16-Bit value
QA	Output Analog	Analog output (if present on device!)
MD	Marker Double Word	32-Bit value
NU	NUmber	Constant, value range from –2147483648 to +2147483647

12.10 Memory requirement

The following table serves as an overview of the memory requirement of the for rungs, function blocks and their associated constants.

→ Avoid empty rungs dispersed between written rungs after operands have been deleted. These kinds of empty rungs take up just as much space as a written rung.

	Space requirement per rung/ function block	Space requirement per constant at function block input
	Byte	Byte
Rung	20	–
Function blocks		
A	68	4
AR	40	4
BC	48	4
BT	48	4
BV	40	4
C	52	4
CF	40	4
CH	52	4
CI	52	4
CP	32	4

	Space requirement per rung/ function block Byte	Space requirement per constant at function block input Byte
D	160	-
DB	36	4
DC	96	4
FT	56	4
GT	28	-
HW	68	4 (per channel)
HY	68	4 (per channel)
JC	20	-
LB	-	-
LS	64	4
MR	20	-
MX easy800	96	4
NC	32	4
OT	36	4
PO	96	4
PT	36	4
PW	48	4
SC	20	-
SR	96	4
ST	24	4
T	48	4
TB	112	4
VC	40	4
:	-	-

Optimization of the memory space requirement

If, for example, you are implementing a project with several easy800 devices in which station “1” uses 32 text elements and you intend to introduce further functions or elements, it is recommended that the expansion of the circuit diagram is distributed on the individual intelligent easy800 devices on the NET, and that the relevant information is exchanged between the individual stations via the NET.

Advantages:

- faster programs,
- commissioning becomes easier to handle (by distributed intelligence),
- straightforward configuration of the NET using the programming software easySoft-Pro.

12.11 ASCII Table

The ASCII characters for the start and end sequence are highlighted in grey

Table 29: Values in the range 0 to 31d

Dec	Hex	Character	Code	Note
0	0x00	Ctrl-@	ZERO	Zero terminator
1	0x01	Ctrl-A	SOH	Start of heading
2	0x02	Ctrl-B	STX	Start of text
3	0x03	Ctrl-C	ETX	End of text
4	0x04	Ctrl-D	EOT	End of transmission
5	0x05	Ctrl-E	ENQ	Enquiry
6	0x06	Ctrl-F	ACK	Acknowledge
7	0x07	Ctrl-G	BEL	Bell
8	0x08	Ctrl-H	BS	Backspace
9	0x09	Ctrl-I	HT	Horizontal tab
10	0x0A	Ctrl-J	LF	Line feed
11	0x0B	Ctrl-K	VT	Vertical tab
12	0x0C	Ctrl-L	FF	Form feed
13	0x0D	Ctrl-M	CR	Carriage return
14	0x0E	Ctrl-N	SO	Shift out
15	0x0F	Ctrl-O	SI	Shift in
16	0x10	Ctrl-P	DLE	Data link escape
17	0x11	Ctrl-Q	DC1	X-ON
18	0x12	Ctrl-R	DC2	
19	0x13	Ctrl-S	DC 3	X-Off
20	0x14	Ctrl-T	DC4	
21	0x15	Ctrl-U	NAK	No acknowledge
22	0x16	Ctrl-V	SYN	Synchronous idle
23	0x17	Ctrl-W	ETB	End transmission blocks
24	0x18	Ctrl-X	CAN	Cancel
25	0x19	Ctrl-Y	EM	End of medium
26	0x1A	Ctrl-Z	SUB	Substitute
27	0x1B	Ctrl-[	ESC	Escape
28	0x1C	Ctrl-\	FS	File separator
29	0x1D	Ctrl-]	GS	Group separator
30	0x1E	Ctrl-^	RS	Record separator
31	0x1F	Ctrl-_	US	Unit separator

12.12 Basic and extended Western European character set

Table 30: Western European basic character set

Unicode (hex)	Character	Unicode (hex)	Character	Unicode (hex)	Character	Unicode (hex)	Character
U+0020		U+0040	@	U+0060	`		
U+0021	!	U+0041	A	U+0061	a	U+00A1	¡
U+0022	"	U+0042	B	U+0062	b	U+00A2	¢
U+0023	#	U+0043	C	U+0063	c	U+00A3	£
U+0024	$	U+0044	D	U+0064	d		
U+0025	%	U+0045	n E T	U+0065	e	U+00A6	¦
U+0026	&	U+0046	F	U+0066	f	U+00A9	©
U+0027	'	U+0047	G	U+0067	g		
U+0028	(	U+0048	H	U+0068	h	U+00B0	°
U+0029	)	U+0049	I	U+0069	i	U+00B1	±
U+002A	*	U+004A	J	U+006A	j	U+00B2	²
U+002B	+	U+004B	KP	U+006B	k	U+00B3	³
U+002C	,	U+004C	L	U+006C	l	U+00B4	´
U+002D	-	U+004D	M	U+006D	m	U+00B5	µ
U+002E	.	U+004E	n E T	U+006E	n		
U+002F	/	U+004F	O	U+006F	o	U+00BF	¿
U+0030	0	U+0050	P	U+0070	p		
U+0031	1	U+0051	Q	U+0071	q	U+201E	„
U+0032	2	U+0052	R	U+0072	r	U+2030	‰
U+0033	3	U+0053	S	U+0073	s	U+20AC	€
U+0034	4	U+0054	T	U+0074	t	U+2190	←
U+0035	5	U+0055	U	U+0075	u	U+2191	↑
U+0036	6	U+0056	V	U+0076	v	U+2192	→
U+0037	7	U+0057	W	U+0077	w	U+2193	↓
U+0038	8	U+0058	X	U+0078	x	U+2194	↔
U+0039	9	U+0059	Y	U+0079	y	U+2195	↕
U+003A	:	U+005A	Z	U+007A	z	U+2264	≤
U+003B	;	U+005B	[	U+007B	{	U+2265	≥
U+003C	<	U+005C	\	U+007C	|	U+2588	█
U+003D	=	U+005D	]	U+007D	]	U+25A0	■
U+003E	>	U+005E	^	U+007E	~	U+25A1	□
U+003F	?	U+005F	_	U+007F		U+25AA	▪

Table 31: Extension of the Western European basic character set

Unicode (hex)	Character	Unicode (hex)	Character	Unicode (hex)	Character	Unicode (hex)	Character
U+00C0	À	U+00E0	à				
U+00C1	Á	U+00E1	á				
U+00C2	Â	U+00E2	â	U+0152	Œ		
U+00C3	Ã	U+00E3	ã	U+0153	œ		
U+00C4	Ä	U+00E4	ä				
U+00C5	Å	U+00E5	å	U+0160	Š		
U+00C6	Æ	U+00E6	æ	U+0161	š		
U+00C7	Ç	U+00E7	ç				
U+00C8	È	U+00E8	è	U+0178	Ÿ		
U+00C9	É	U+00E9	é	U+017D	Ž		
U+00CA	Ê	U+00EA	ê	U+017E	ž		
U+00CB	Ë	U+00EB	ë				
U+00CC	Ì	U+00EC	ì				
U+00CD	Í	U+00ED	í				
U+00CE	Î	U+00EE	î				
U+00CF	Ï	U+00EF	ï				
U+00D0	Đ	U+00F0	ð				
U+00D1	Ñ	U+00F1	ñ				
U+00D2	Ò	U+00F2	ò				
U+00D3	Ó	U+00F3	ó				
U+00D4	Ô	U+00F4	ô				
U+00D5	Õ	U+00F5	õ				
U+00D6	Ö	U+00F6	ö				
U+00D7		U+00F7					
U+00D8	Ø	U+00F8	ø				
U+00D9	Ù	U+00F9	ù				
U+00DA	Ú	U+00FA	ú				
U+00DB	Û	U+00FB	û				
U+00DC	Ü	U+00FC	ü				
U+00DD	Ý	U+00FD	ý				
U+00DE		U+00FE	þ				
U+00DF	ß	U+00FF	ÿ				

The basic and extended Western European character sets shown above can be output via the D text display function block of an easy800 device.

Index

Eaton's Electrical Sector is a global leader in power distribution, power quality, control and automation, and monitoring products. When combined with Eaton's full-scale engineering services, these products provide customerdriven PowerChain™ solutions to serve the power system needs of the data center, industrial, institutional, public sector, utility, commercial, residential, IT, mission critical, alternative energy and OEM markets worldwide. PowerChain solutions help enterprises achieve sustainable and competitive advantages through proactive management of the power system as a strategic, integrated asset throughout its lifecycle, resulting in enhanced safety, greater reliability and energy efficiency.

For more information, visit
www.eaton.com/electrical.

Eaton addresses worldwide:
www.moeller.net/address

E-Mail: info-bonn@eaton.com
Internet: www.eaton.com/moellerproducts

Eaton Industries GmbH
Hein-Moeller-Str. 7–11
D-53115 Bonn

© 2003 by Eaton Corporation
All rights reserved

MN04902001Z-EN Doku/Doku/Eb 11/11
Printed in Germany (07/11)
Article No.: 262671

4 015082 626716

Powering Business Worldwide