



Building Automation

Industrial Automation

Systems

User Manual

easy800 Control Relay

08/04 AWB2528-1423GB



Think future. Switch to green.

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

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

1st published 2002, edition date 05/02

2nd edition 11/2002

3rd edition 06/2003

4th edition 08/2004

See revision protocol in the "About this manual" chapter

© Moeller GmbH, 53105 Bonn

Author: Dieter Bauerfeind

Editor: Heidrun Riege

Translator: Terence Osborn, David Long

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 Moeller GmbH, Bonn.

Subject to alteration without notice.

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



Warning! Dangerous electrical voltage!

Before commencing the installation

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

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

Contents

<hr/>	
About This Manual	9
Device designation	9
Reading conventions	10
List of revisions	11
<hr/>	
1 easy800	15
Target readership	15
Proper use	15
– Improper use	15
Overview	16
Device overview	18
– easy basic units at a glance	18
– easy800 type references	19
easy operation	20
– Buttons	20
– Moving through menus and choosing values	20
– Selecting main and system menu	21
– easy800 status display	21
– Status display for local expansion	22
– easy800 advanced Status display	22
– easy800 LED display	23
– Menu structure	24
– Selecting or toggling between menu items	29
– Cursor display	30
– Setting values	30
<hr/>	
2 Installation	31
Mounting	31
Connecting the expansion device	34
Terminals	35
– Tools	35
– Cable cross-sections	35
Network cables and plug	35
Connecting the power supply	35
– DC basic units	38
– Cable protection	39

- Connecting the inputs 39
- Connecting the AC inputs 39
- Connecting the easy-DC 44
- Connecting the outputs 50
- Connecting relay outputs 51
- EASY8....-RC.. 51
- EASY6...-RE.. 52
- EASY2..-RE 52
- Connecting transistor outputs 53
- EASY8..-DC-TC, EASY6..-DC-TE 53
- Connecting analog outputs 55
- Connecting servo valves 55
- Setpoint entry for a drive 56
- Connecting the NET network 56
- Accessories 56
- Cable length and cross-sections 58
- Plugging and unplugging network cables 60
- Expanding inputs/outputs 62
- Local expansion 63
- Remote expansion 64

- 3 Commissioning 65**
- Switching on 65
- Setting the menu language 65
- easy operating modes 66
- Creating your first circuit diagram 67
- Starting point Status display 69
- Circuit diagram display 70
- From the first contact to the output coil 71
- Wiring 72
- Testing the circuit diagram 74
- Deleting the circuit diagram 77
- Fast circuit diagram entry 77
- Configuring an easy-NET network 78
- Entering the network station number 79
- Entering network stations 80
- Configuring an easy-NET network 81
- Changing the easy-NET network configuration 82
- Displaying the Status display of other stations 83

4	Wiring with easy800	85
	easy800 operation	85
	– Buttons for drawing circuit diagrams and function block usage	85
	– Operation	86
	– Usable relays and function blocks (coils)	94
	– Markers, analog operands	97
	– Number formats	99
	– Circuit diagram display	100
	– Saving and loading programs	101
	Working with contacts and relays	103
	– Creating and modifying connections	106
	– Inserting and deleting a circuit connection	108
	– Saving circuit diagrams	109
	– Aborting circuit diagram entry	109
	– Searching for contacts and coils	109
	– “Go to” a circuit connection	110
	– Deleting the rung	110
	– Switching via the cursor buttons	111
	– Checking the circuit diagram	112
	– Function block editor	113
	– Checking function blocks	117
	– Coil functions	118
	Function blocks	123
	– Analog value comparator/ threshold value switch	125
	– Arithmetic function block	128
	– Data block comparator	132
	– Data block transfer	139
	– Boolean operation	150
	– Counters	153
	– High-speed counters	159
	– Frequency counter	160
	– High-speed counters	164
	– High-speed incremental encoder counters	170
	– Comparators	175
	– Text output function block	177
	– Data function block	181
	– PID controller	183
	– Signal smoothing filter	189

- GET, fetch a value from the network 192
- Seven-day time switch 194
- Year time switch 200
- Value scaling 204
- Jumps 208
- Master reset 211
- Numerical converters 212
- Operating hours counter 218
- PUT, send a value onto the network 219
- Pulse width modulation 221
- Setting date/time 224
- Set cycle time 226
- Timing relay 228
- Value limitation 241
- Example with timing relay and counter function block 243

- 5 easy-NET network 247**
- Introduction to easy-NET 247
- easy-NET network topologies, addressing and functions 248
 - Loop through the unit wiring method 248
 - T connector and spur line 248
 - Topology and addressing examples 249
 - Position and addressing of the operands via easy-NET 250
 - Functions of the stations in the network 252
 - Possible write and read authorisation in the network 252
- Configuration of the easy-NET network 253
 - Station number 253
 - Transmission speed 253
 - Pause time, changing the write repetition rate manually 254
 - Send each change on the inputs/outputs (SEND IO) 255
 - Automatic change of the RUN and STOP mode 255
 - Input/output device (REMOTE IO) configuration 256
 - Displaying the Status display of other stations 257
 - Station message types 258

– Transfer behaviour	258
– Signs of life of the individual stations and diagnostics	259
– Network transmission security	262
<hr/>	
6 easy Settings	263
Password protection	263
– Password setup	264
– Selecting the scope of the password	265
– Activating the password	266
– Unlocking easy	267
– Changing or deleting the password range	268
Changing the menu language	270
Changing parameters	271
– Adjustable parameters for function blocks	272
Setting date, time and daylight saving time	273
Changing between winter/summer time (DST)	274
– Selecting DST	275
Activating input delay (debounce)	276
– Deactivating debounce (input delay)	276
Activating and deactivating the P buttons	277
– Activating the P buttons	277
– Deactivating the P buttons	278
Startup behaviour	278
– Setting the startup behaviour	278
– Behaviour when the circuit diagram is deleted	279
– Behaviour during upload/download to card or PC	279
– Possible faults	280
– Card startup behaviour	280
Setting LCD contrast and backlight	281
Retention	283
– Requirements	284
– Setting retentive behaviour	284
– Deleting ranges	285
– Deleting retentive actual values of markers and function blocks	285
– Transferring retentive behaviour	286
Displaying device information	287

7	Inside easy	289
	easy Program cycle	289
	– How easy evaluates the high-speed counters CF, CH and CI	292
	Delay times for inputs and outputs	292
	– Delay times with easy-DC basic units	293
	– Delay time with easy AC basic units	294
	– Behaviour with and without delay time	295
	Monitoring of short-circuit/overload with EASY...-D.-T..	297
	Expanding easy800	299
	– How is an expansion unit recognised?	299
	– Transfer behaviour	300
	– Function monitoring of expansion units	300
	QA analog output	302
	– Behaviour with assignment of values exceeding 1023	302
	Loading and saving programs	303
	– EASY...-...-X.MFD without display and keypad	303
	– Program compatibility of the hardware	303
	– Interface	304
	– COM connection	304
	– Terminal mode	304
	– Memory card	305
	– EASY-SOFT (-PRO)	308
	Compatibility of different easy800 versions.	310
	– Device compatibility	310
	Device version	311

Appendix	313
Technical data	313
– General	313
– Power supply	318
– Inputs	319
– Relay outputs	324
– Transistor outputs	326
– Analog output	329
– easy-NET Network	330
List of the function blocks	332
– Function blocks	332
– Function block coils	333
– Function block contacts	334
– Function block inputs (constants, operands)	335
– Function block output (operands)	336
– Other operands	336
Memory space requirement	337
– Optimization of the memory space requirement	338

Index	339
--------------	-----

About This Manual

This manual describes the installation, commissioning and programming (circuit-diagram generation) of the easy800 control relay.

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

Device designation

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

- easy800 for
 - EASY819-..,
 - EASY820-..,
 - EASY821-..,
 - EASY822-..
- easy412 for
 - EASY412-AC-....,
 - EASY412-D-....
- easy600 for
 - EASY6..-AC-RC(X)
 - EASY..-DC-.C(X)

- easy-AC for
 - EASY8..-AC-...
 - EASY412-AC..
 - EASY6..-AC-RC(X)
- easy-DC for
 - EASY8..-DC-...
 - EASY12-DC..
 - EASY620/621-DC-.C(X)
- easy-DA for
 - EASY412-DA-RC

Reading conventions

Symbols used in this manual have the following meanings:

► Indicates actions to be taken.



Attention!

Warns of the possibility of light damage.



Caution!

Warns of the possibility of serious damage and slight injury.



Warning!

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



Indicates interesting tips and additional information

For greater clarity, the name of the current chapter is shown in the header of the left-hand page and the name of the current section in the header of the right-hand page. Pages at the start of a chapter and empty pages at the end of a chapter are exceptions.

List of revisions

Edition	Page	Description	New	Modifi- cation	Omitted
08/04	48	Section "20 A sensor"		✓	
	97	Table 6		✓	
	97	Section "Composition of the markers"		✓	
	168	Figure 76, "Signal diagram of high-speed counter"		✓	
	194	Section "Function of the GET function block"		✓	
	202	Section "Function of the year time switch function block"		✓	
	219	Section "Accuracy"		✓	
	223	Section "Function of the pulse width modulation function block"		✓	
	230	Section "Time range", table		✓	
	231	Section "Variable setpoint values", example		✓	
	234	Figure 106		✓	
	235, 236	Figure 107 and Figure 108		✓	
	269	Section "Password incorrect or no longer known"		✓	
	305	Section "Memory card"		✓	
	308	Section "EASY-SOFT (-PRO)"		✓	
	315	Electromagnetic compatibility (EMC), radio interference suppression		✓	
	316	Dielectric strength			

Edition	Page	Description	New	Modification	Omitted
06/03	58	Cable length and cross-sections		✓	
	59	Section "Calculating the cable length with known cable resistance"	✓		
	197	Section "Memory requirement of the 7-day time switch"		✓	
	202	Section "Memory requirement for the year time switch"		✓	
	220	Coils		✓	
	242	Coil		✓	
	248	Note: length of spur line	✓		
	332	List of the function blocks		✓	
	337	HW, HY: space requirement at the function block output		✓	
05/03	132	Data block comparator	✓		
	139	Data block transfer	✓		
	153	NOT Boolean sequence		✓	
	180	Entering setpoint values	✓		
	183	PID controller	✓		
	189	Signal smoothing filter	✓		
	195	Warning switching behaviour	✓		
	204	Value scaling	✓		
	212	Numerical converters	✓		
	221	Pulse width modulation	✓		
	226	Set cycle time	✓		
	257	Note on Status display	✓		
	266	Activating the password		✓	
	274	Note on DST time change	✓		

Edition	Page	Description	New	Modifi- cation	Omitted
11/02	302	QA analog output, Value range	✓		
	336	Function block output (operands)	✓		
	15	Target readership	✓		
	15	Proper use	✓		
	21/21	Menu displays		✓	
	24	Operating buttons	✓		
	26	Operating buttons	✓		
	46	Number of circuits		✓	
	58	Surge impedance	✓		
	61	Fig. 36, Station number		✓	
	78	Fig. 49, Station number		✓	
	89	Short-circuit/overload with expansion			✓
	108	Number of circuit connections		✓	
	115	Notation for >I1 and QV>		✓	
	117	Notation for >I1 and QV>		✓	
	117	Notation for >I1 and QV>		✓	
	122	Useful coil functions			✓
	129	Notation for QV>		✓	
	151	Notation for QV>		✓	
	163	Notation for I1 and I4		✓	
	177	Memory requirement 160 Bytes		✓	
	197	Memory requirement 68 Bytes		✓	
	202	Memory requirement 68 Bytes		✓	
	218	Fig. 98, last line	✓		
	228	Notation for parameter display		✓	
	232	Memory requirement 48 Bytes		✓	
	240	Pulse and pause time		✓	
	250	Point to point connection	✓		

Edition	Page	Description	New	Modifi- cation	Omitted
11/02	253	125 KB factory default setting		✓	
	290	Processing function blocks		✓	
	294/ 295	Off-delay I7 and I8		✓	
	302	QA analog output	✓		
	303	Saving and loading circuit diagrams	✓		
	318	List of the function blocks	✓		
	318	Memory requirement	✓		

1 easy800

Target readership

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

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

Proper use

easy is a programmable switching and control device and is used as a replacement for relay and contactor control circuits. easy must be properly installed before use.

easy is designed to be installed in an enclosure, switch cabinet or distribution board. Both the power feed and the signal terminals must be laid and covered so as to prevent accidental contact.

The installation must conform to regulations for electromagnetic compatibility (EMC).

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

Improper use

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

Overview

easy800 is an electronic control relay with:

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

easy800 is a control and input device rolled into one. With easy800 you can create solutions for domestic applications as well as for tasks in machine and plant construction.

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

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

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

- use up and down counters,
- count high-speed signals:
 - up and down counters with upper and lower limit values,
 - preset,
 - frequency counters,
 - high-speed counters,
 - count incremental encoder values.
- compare values,
- display any texts with variables, enter setpoints
- process analog input and output values (DC units),
- process additional inputs and outputs,
- use 7-day and year time switches,
- count operating hours (operating hours counter),
- communicate via the integrated easy-NET network,
- provide closed-loop control with P, PI and PID controllers,
- scale arithmetic values,
- output manipulated variables as pulse-width modulated signals,
- run arithmetic functions:
 - add,
 - subtract,
 - multiply,
 - divide.
- track the flow of current in the circuit diagram
- load, save and password-protect circuit diagrams

If you prefer to wire up easy800 from a PC, then use EASY-SOFT or EASY-SOFT-PRO. EASY-SOFT or EASY-SOFT-PRO allow you to create and test your circuit diagram on the PC. EASY-SOFT (-PRO) enables you to print out your circuit diagram in DIN, ANSI or easy format.

Device overview

easy basic units at a glance

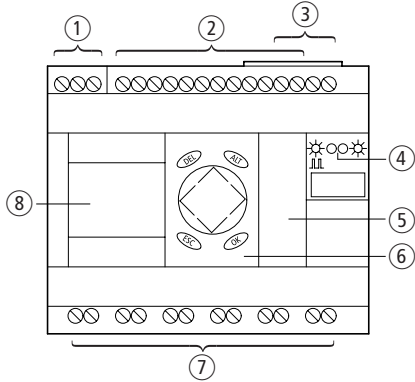
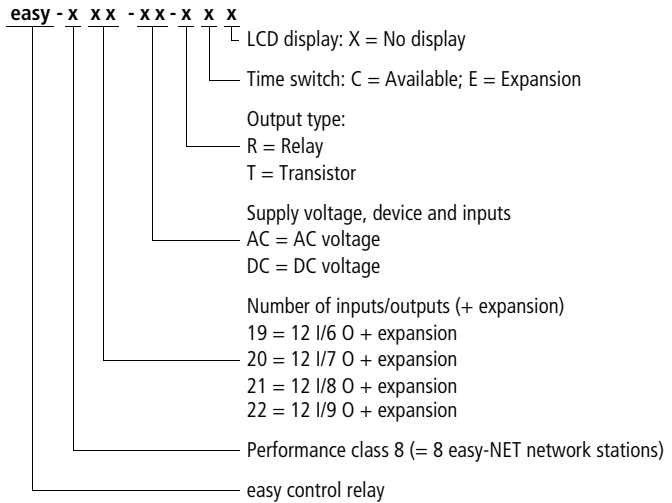


Figure 1: Device overview

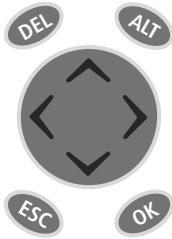
- ① Power supply
- ② Inputs
- ③ easy-NET connections
- ④ Operating status LEDs
- ⑤ Interface socket for memory card or PC connection
- ⑥ Buttons
- ⑦ Outputs
- ⑧ Display

easy800 type references



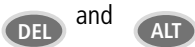
easy operation

Buttons



- DEL:** Delete object in circuit diagram
- ALT:** Special functions in circuit-diagram, Status display
- Cursor buttons** < > ^ ∨:
Move cursor
Select menu items
Set contact numbers, contacts and values
- OK:** Next menu level, Save your entry
- ESC:** Previous menu level, Cancel

Moving through menus and choosing values



and

Show System menu



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



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



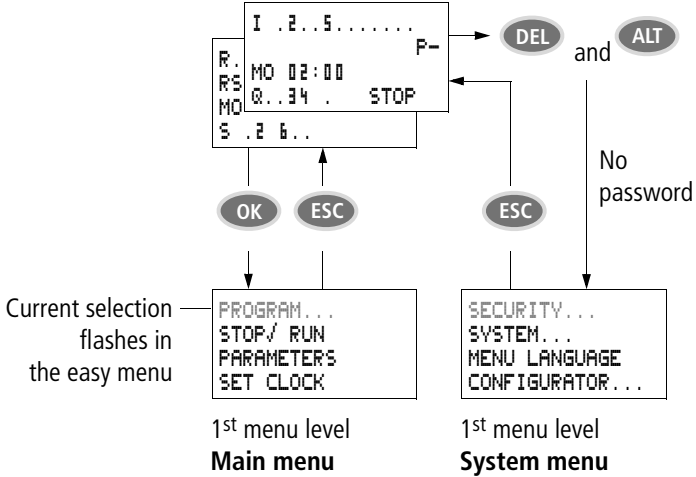
- ^ ∨ Change menu item
- Change value
- < > Change place

Function of P buttons:

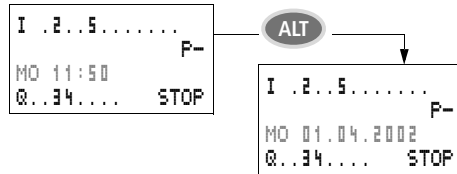
- | | | | |
|---|-----------|---|----------|
| < | Input P1, | ^ | Input P2 |
| > | Input P3, | ∨ | Input P4 |

Selecting main and system menu

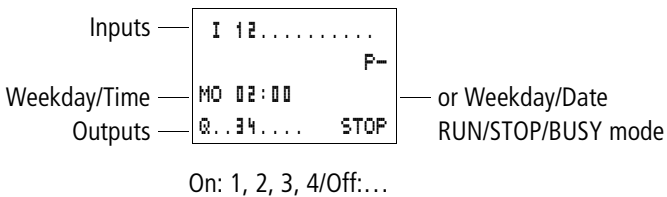
Status display



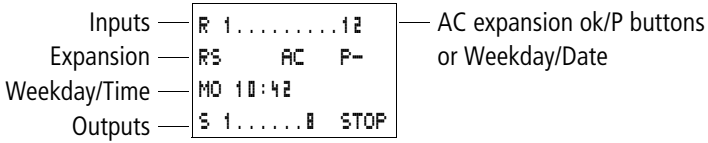
Date display



easy800 status display



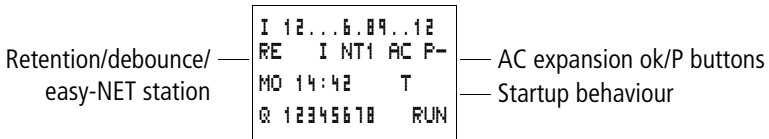
Status display for local expansion



On: 1, 2, 3, 4/Off:...

RS = Expansion functioning correctly

easy800 advanced Status display



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

easy800 LED display

easy800 has two LEDs located on the front which indicate the state of the power supply voltage (POW) as well as the RUN or STOP modes (→ Fig. 1, Page 18).

Table 1: LED power supply/RUN-STOP mode

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

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

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

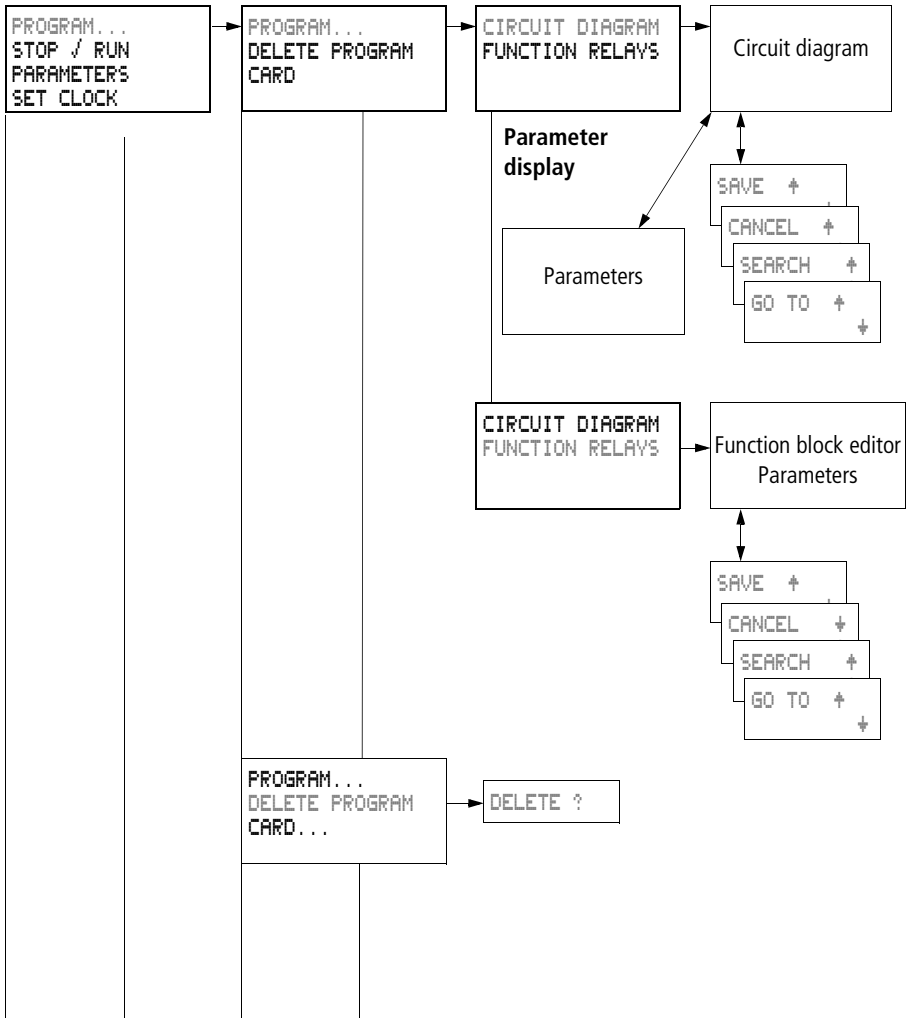
Menu structure

Main menu without password protection

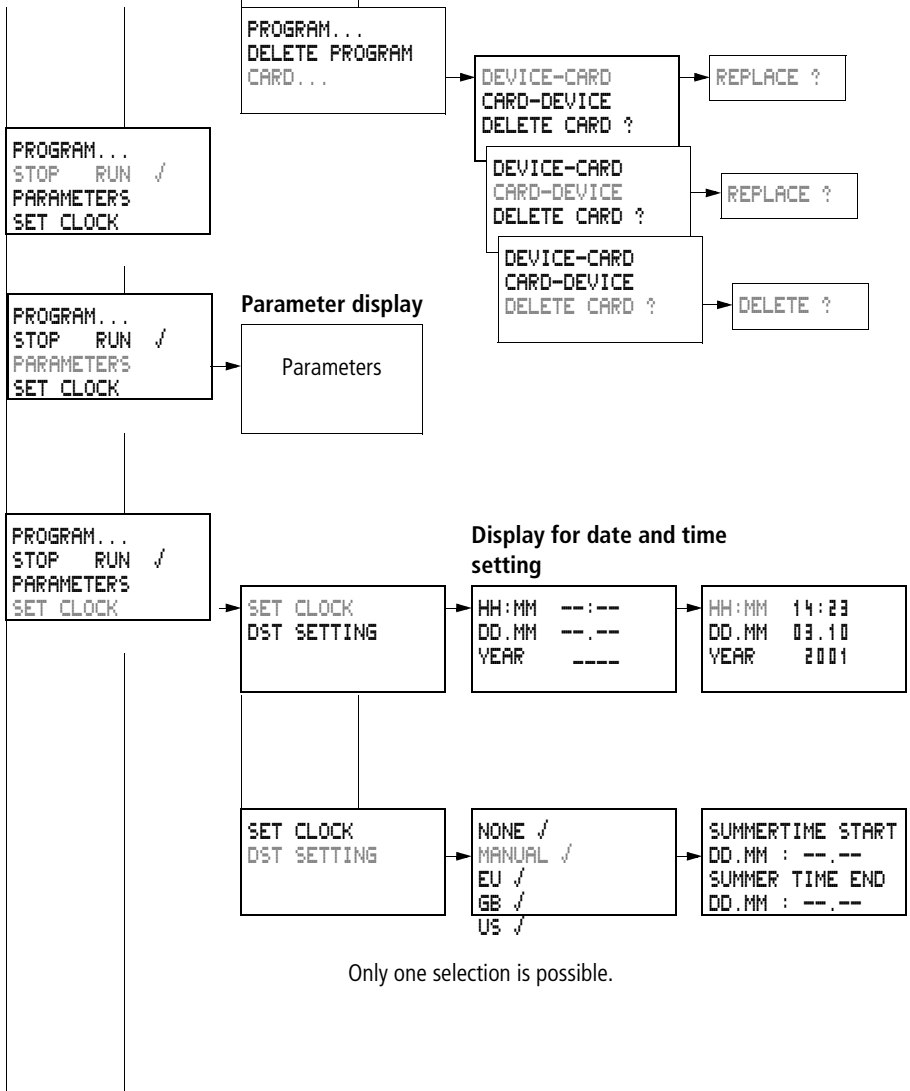
► You access the main menu by pressing OK.

STOP: **Circuit diagram display**
RUN: **Power flow display**

Main menu

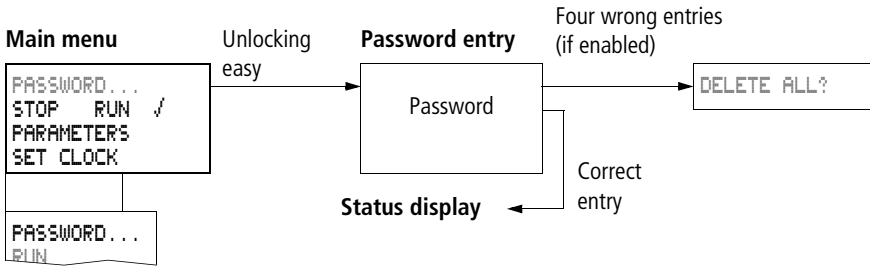


Main menu



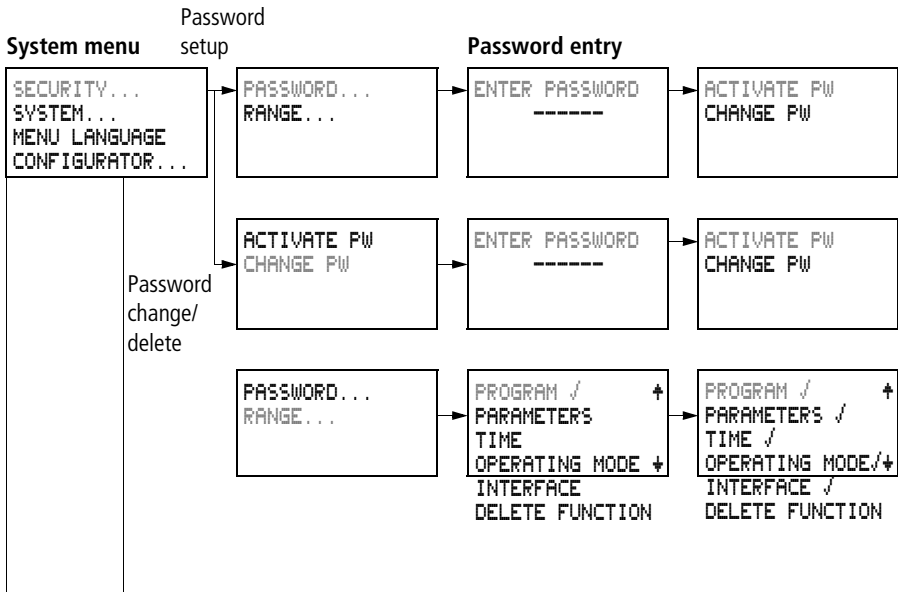
Only one selection is possible.

Main menu with password protection

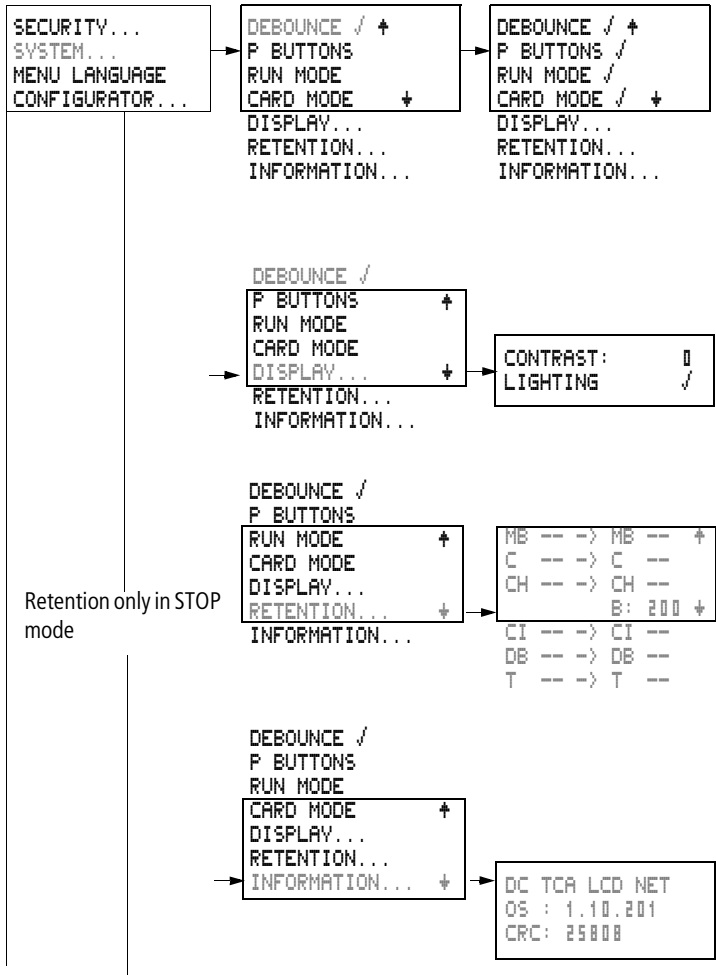


easy800 System menu

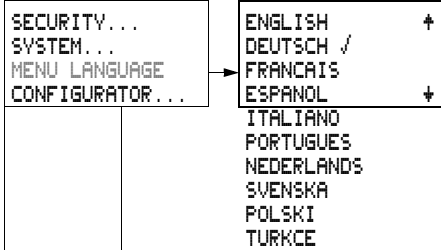
- ▶ The System menu is accessed by simultaneously pressing DEL and ALT.



System menu



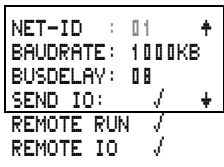
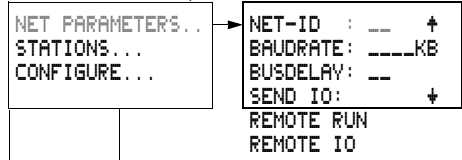
System menu



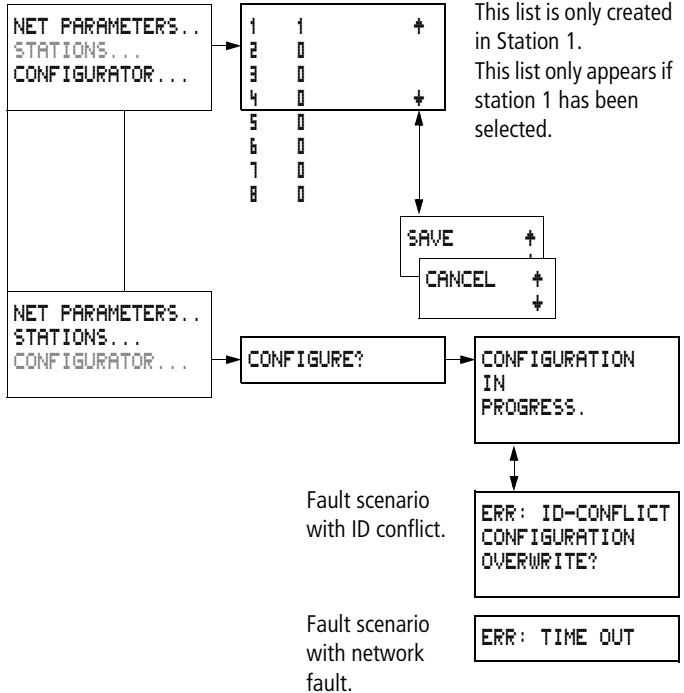
Only one selection is possible.



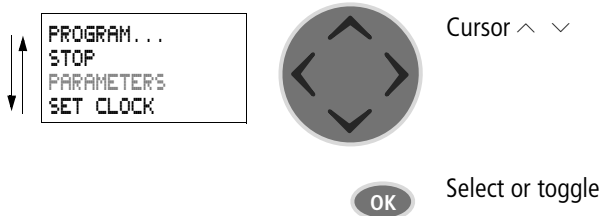
easy-NET only in STOP mode



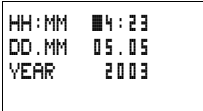
System menu



Selecting or toggling between menu items



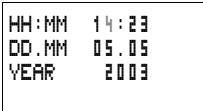
Cursor display



The cursor flashes:

Full cursor

- Move cursor with < >
- in circuit diagram also with ^ v

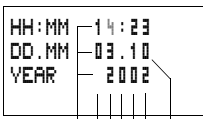


Value M/M

- Change position with < >
- Change values with ^ v

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

Setting values



Values
Positions
Value at position



- Change value ^ v
- Select cursor position in value < >
- Change value at position ^ v



Store entries



Retain previous value

2 Installation

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



Danger of electric shock!

Never carry out electrical work on the device while the power supply is switched on.

Always follow the safety rules:

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

The easy is installed in the following order:

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

Mounting

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

Snap easy onto a IEC/EN 60715 top-hat rail or fix easy in place using fixing brackets. easy can be mounted vertically or horizontally.



When using the easy with expansion units, connect the expansion concerned before mounting (see → page 34).

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

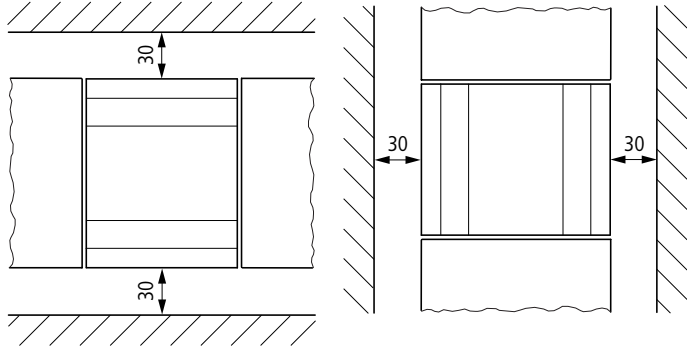
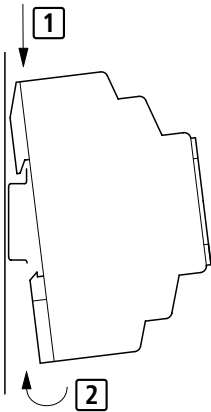


Figure 2: Clearances to the easy



Mounting on mounting rail

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

The easy will clip into place automatically.

- ▶ Check that easy is seated firmly.

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

Screw mounting

For screw mounting on a mounting plate, fixing brackets must be used that can be fixed to the back of the easy. The fixing brackets can be ordered as an accessory.

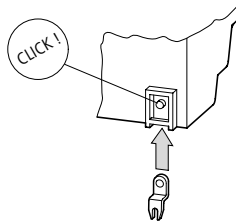


Figure 3: Inserting a fixing bracket



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

EASY2...-...:

easy600, easy800:

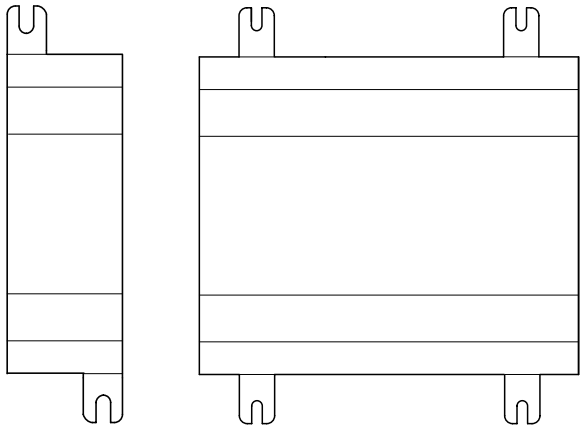


Figure 4: Screw mounting for easy

Connecting the expansion device

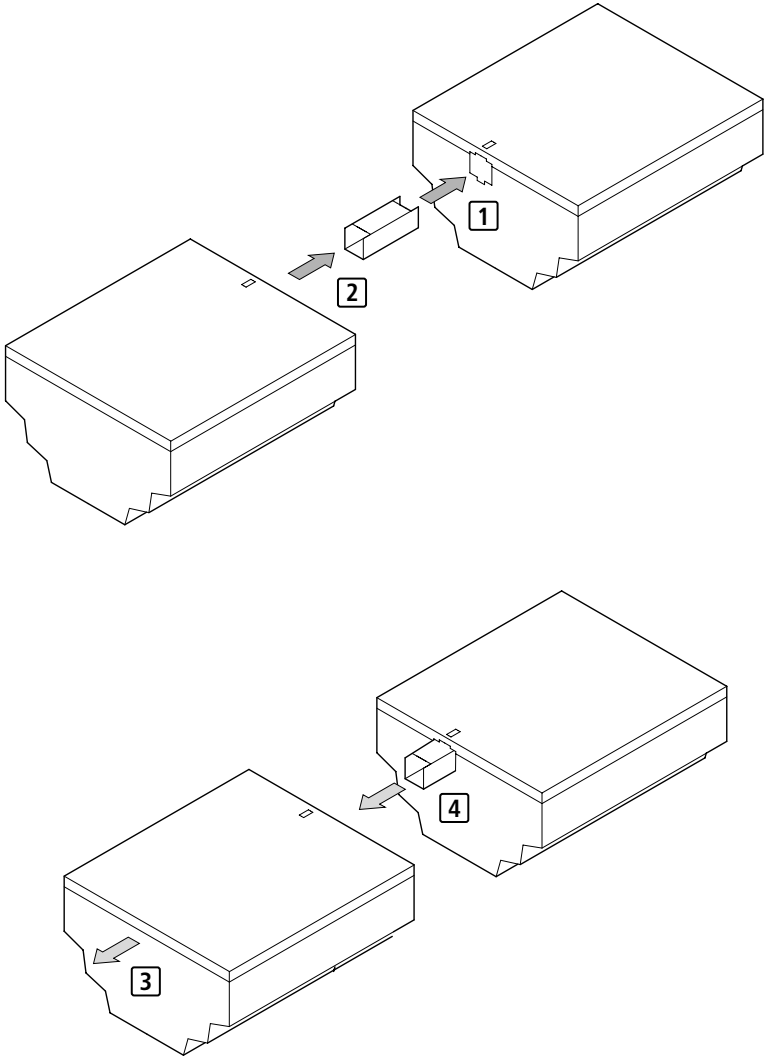


Figure 5: Expansion connection

Terminals**Tools**

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

Cable cross-sections

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

Network cables and plug

Use the prefabricated EASY-NT-“Length” cables when possible.

Other cable lengths can be manufactured using the EASY-NT-CAB cable, the EASY-NT-RJ45 plug as well as the EASY-RJ45-TOOL crimping tool.

AWG 24, 0.2 mm² are the largest crimpable cross-sections.

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

Connecting the power supply

For the connection data of both versions, **easy-DC** with 24 V DC and **easy-AC** with standard voltages of 100 V to 240 V AC, refer to Chapter “Technical data” from Page 318.

The easy800 devices run a system test for one second after the power supply has been switched on. Either RUN or STOP mode will be activated after this time depending on the default setting.

AC basic units

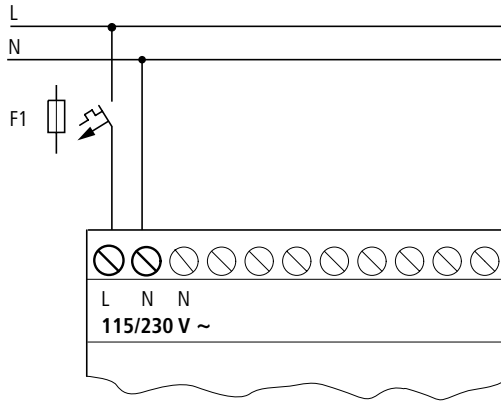


Figure 6: Power supply on the AC-basic units

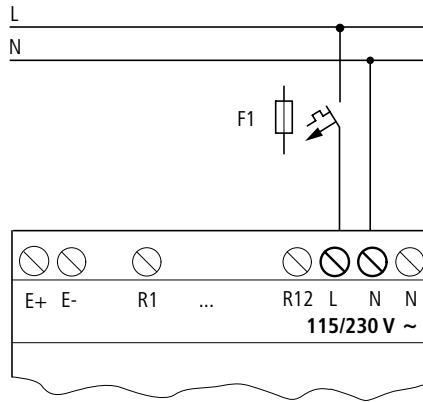
EASY...-AC-E expansion units

Figure 7: Power supply on the AC expansion units

**Attention!**

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

DC basic units

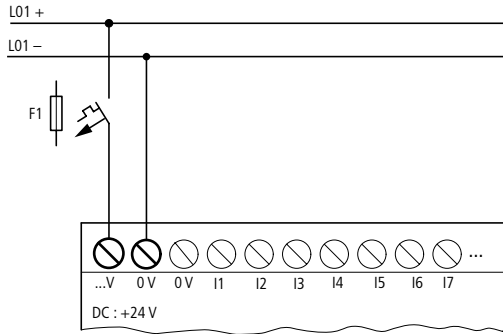


Figure 8: Power supply on the DC basic units

EASY...-DC-E DC expansion units

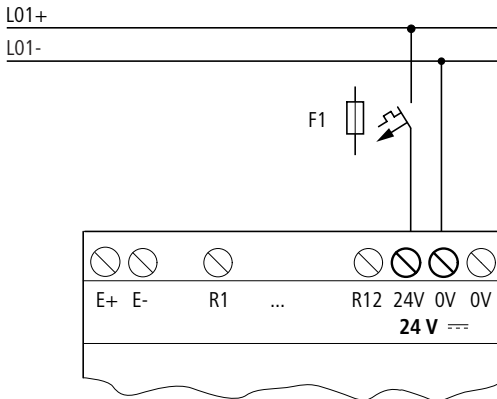


Figure 9: Power supply on the DC expansion units



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

Cable protection

Both easy AC and DC versions require cable protection (F1) rated for at least 1 A (slow).



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

Connecting the inputs

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

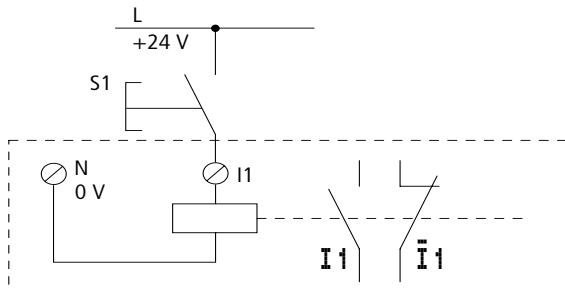


Figure 10: Connecting the inputs

Connect contacts such as push-button actuators or switches to easy input terminals.

Connecting the AC inputs



Caution!

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

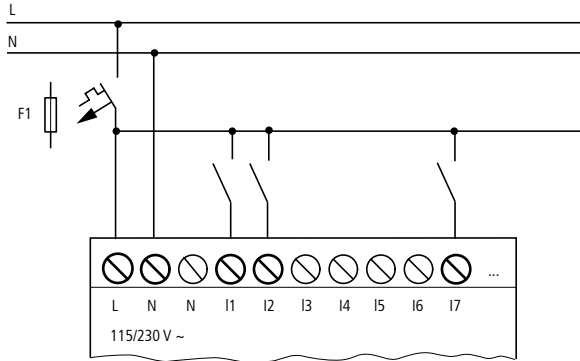


Figure 11: easy-AC basic unit

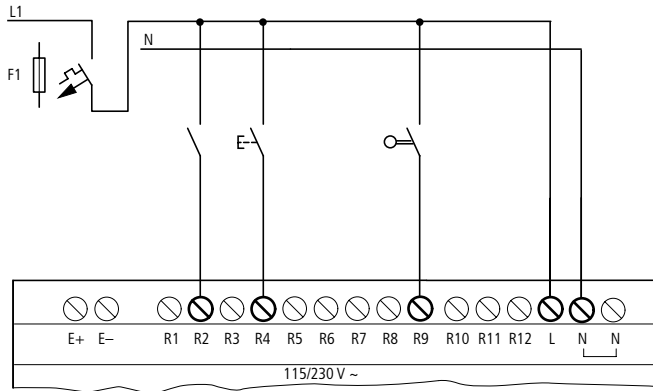


Figure 12: Inputs on the EASY...-AC-.E expansion device

Connect the inputs, for example, to push-button actuators, switches or relay/contactors contacts.

Input signal voltage range

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

Input current

- R1 to R12, 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

Cable length

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

- R1 to R12: 40 m without additional circuit
- 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

The following applies to expansion units:

With longer cables, connect a diode (e.g. 1N4007) for 1 A, minimum 1 000 V reverse voltage, in series to the easy input. Ensure that the diode is pointing towards the input as shown in the circuit diagram, otherwise easy will not detect the 1 state.

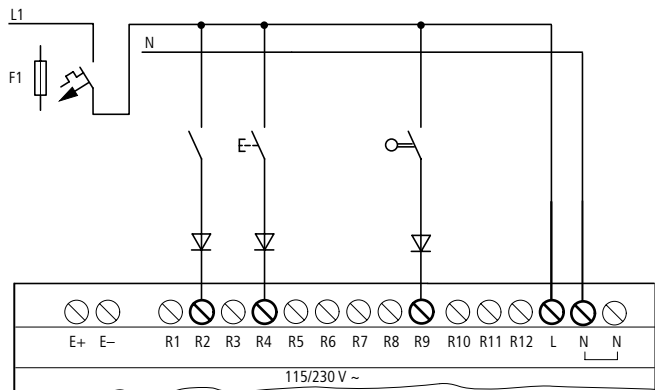


Figure 13: easy-AC with a diode on the inputs

Neon bulbs 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 bulbs that are operated with a separate N connection.



Caution!

Do not use reed relay contacts on I7, I8. These may burn or melt due to the high inrush current of I7, I8.

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

Therefore, use inputs I7 and I8. An additional input circuit is required if more inputs are used.

Increasing the input current

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

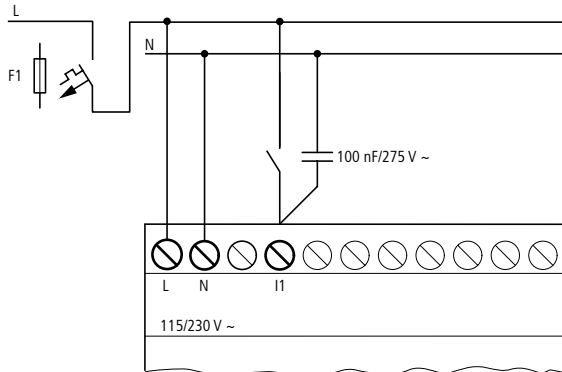


Figure 14: Increasing the input current



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

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

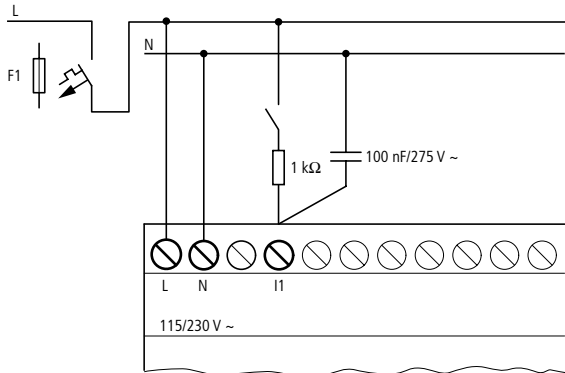


Figure 15: Limitation of the inrush current with a resistor

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

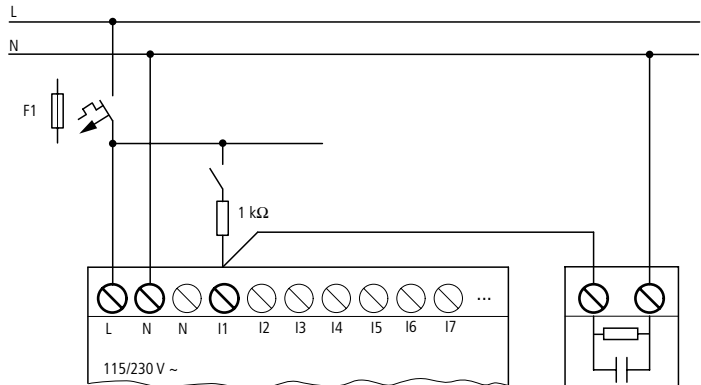


Figure 16: easy800 with easy256-HCI



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

Connecting the easy-DC

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

Input signal voltage range

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

Input current

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

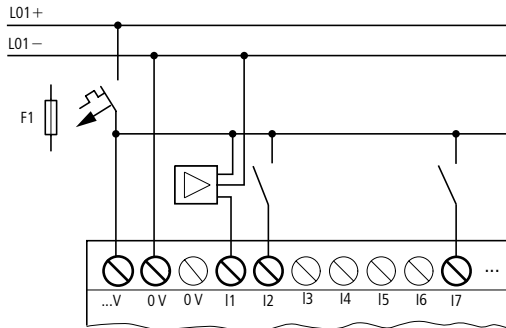


Figure 17: easy-DC

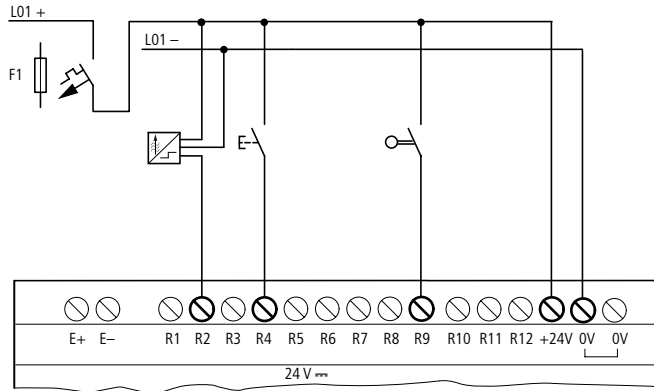


Figure 18: EASY...-DC-.E

Connecting analog inputs

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

The following applies:

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

The resolution is 10-bit = 0 to 1023.



Caution!

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

- ▶ Use shielded twisted pair cables to prevent interference with the analog signals.
- ▶ For short cable lengths, ground the shield at both ends using a large contact area. If the cable length exceeds 30 m or so, grounding at both ends can result in equalisation currents between the two grounding points

and thus in the interference of analog signals. In this case, only ground the cable at one end.

- ▶ Do not lay signal lines parallel to power cables.
- ▶ Connect inductive loads to be switched via the easy outputs to a separate power feed, or use a suppressor circuit for motors and valves. If loads such as motors, solenoid valves or contactors are operated with easy via the same power feed, switching may give rise to interference on the analog input signals.

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



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

Setpoint potentiometer

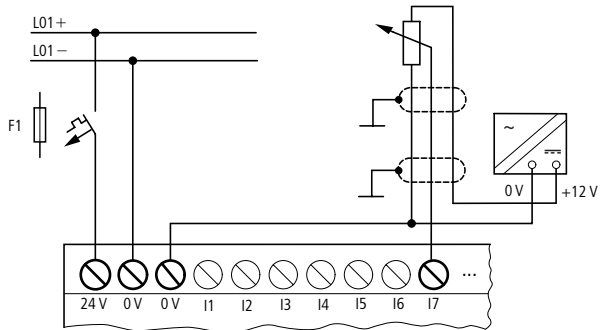


Figure 19: Setpoint potentiometer

Use a potentiometer with a resistance of $\leq 1 \text{ k}\Omega$, e.g. $1 \text{ k}\Omega$, 0.25 W .

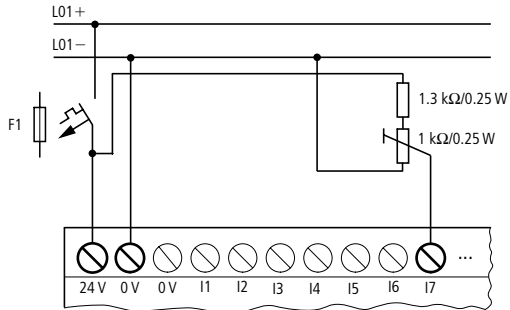


Figure 20: Setpoint potentiometer with upstream resistor

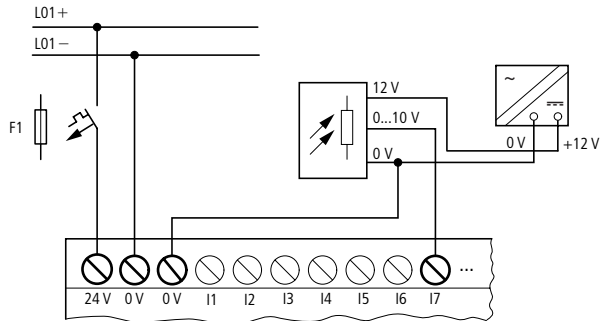


Figure 21: Brightness sensor

Temperature sensor

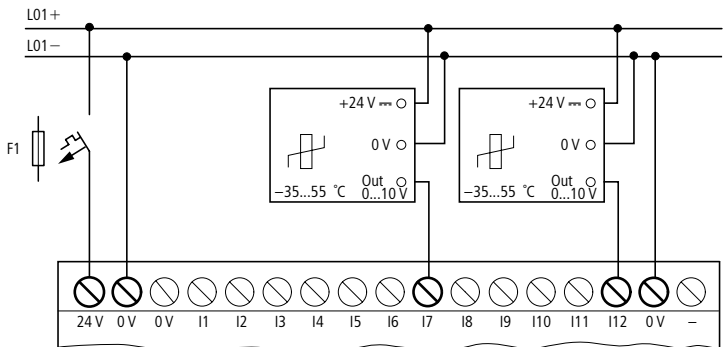


Figure 22: Temperature sensor

20 A sensor

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

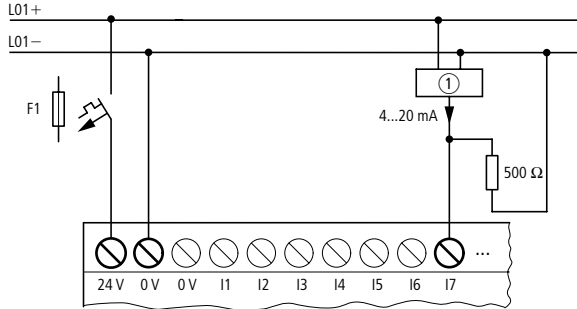


Figure 23: 20 A sensor

① Analog sensor

The following values apply:

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

(according to $U = R \times I = 478 \Omega \times 10 \text{ mA} \sim 4.8 \text{ V}$)

Connecting high-speed counters and frequency generators

High-speed counter signals on the easy800 can be counted correctly on inputs I1 to I4 independently of the cycle time.

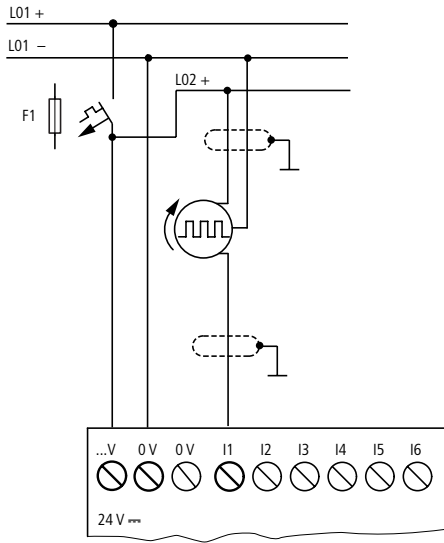


Figure 24: High-speed counter, frequency generator

Connecting incremental encoders

Inputs I1, I2 and I3, I4 on the easy800 can each be used for the high-speed counting of an incremental encoder independently of the cycle time. The incremental encoder must generate two 24 V DC square wave signals with a 90° phase shift between them.

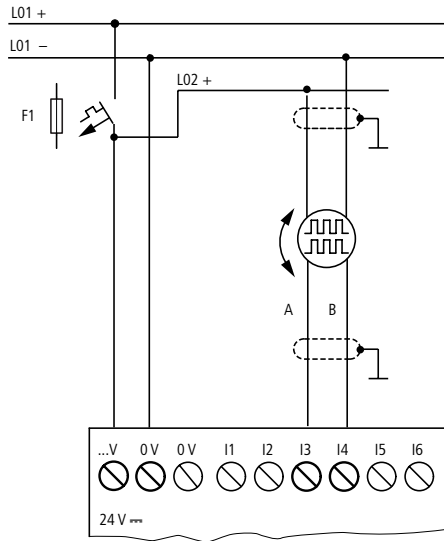


Figure 25: Connecting incremental encoders

Connecting the outputs

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

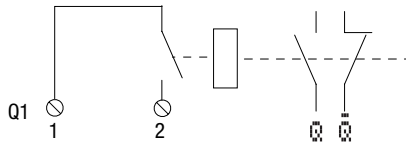


Figure 26: Output "Q"

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

The relay or transistor outputs are used to switch loads such as fluorescent tubes, filament bulbs, contactors, relays or motors. Check the technical thresholds and output data before installing such devices (see → chapter "Technical data", from Page 313).

Connecting relay outputs EASY8...-RC..

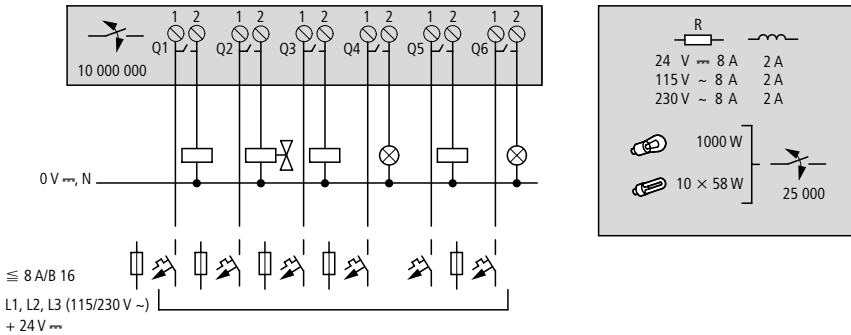


Figure 27: EASY8...-RC.. relay outputs

EASY6...-RE..

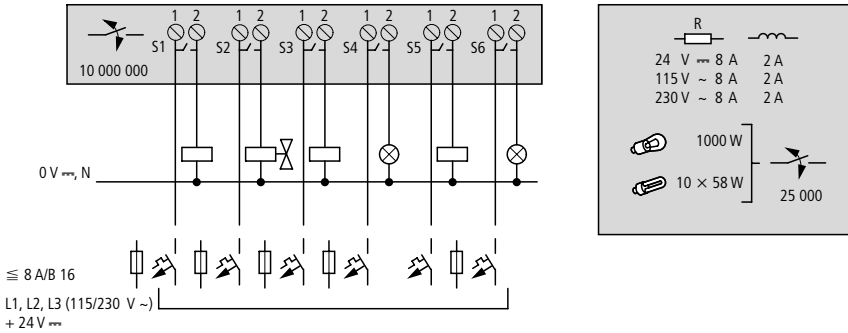


Figure 28: EASY6...-RE.. relay outputs

EASY2...-RE

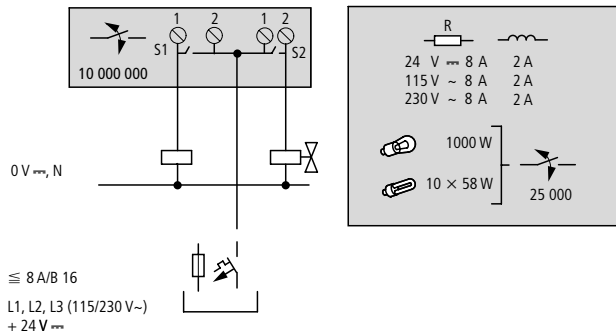


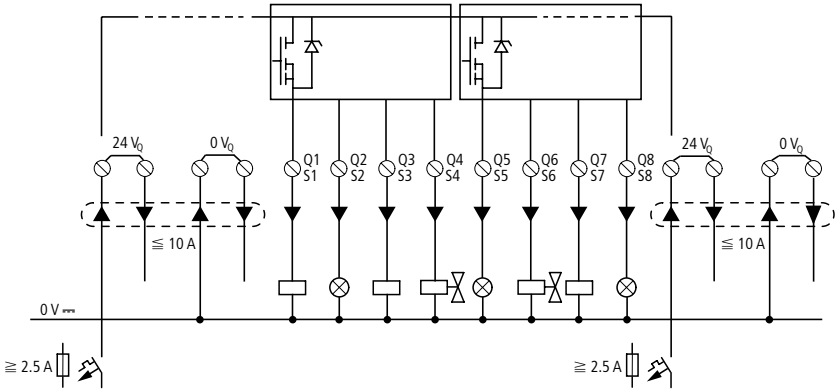
Figure 29: EASY2...-RE.. relay outputs

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



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

Connecting transistor outputs **EASY8..-DC-TC, EASY6..-DC-TE**

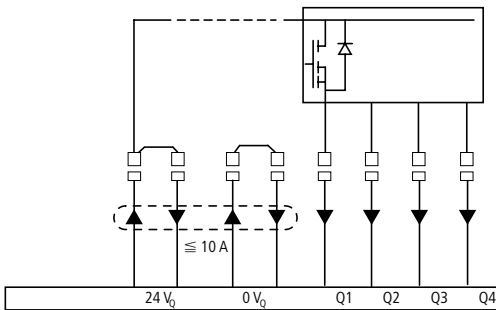


EASY8..-DC-...

24 V ~	0.5 A	0.5 A
Q1 - Q4	3 W	24 V
Q5 - Q8	5 W	

EASY6..-DC-..

+ 24 V ~	24 V ~	0.5 A
(20.4 - 28.8 V ~)	0.5 A	0.5 A
		5 W/24 V



+ 24 V ~	24 V ~	0.5 A
(20.4 - 28.8 V ~)	0.5 A	0.5 A
		5 W/24 V

Figure 30: Transistor outputs EASY8..-DC-TC, EASY6..-DC-TE

Parallel connection:

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



Caution!

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



Caution!

Please note the following when switching off inductive loads:

Suppressed inductive loads cause less interference in the entire electrical system. For optimum suppression the suppressor circuits are best connected directly in the proximity of the inductive load.

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

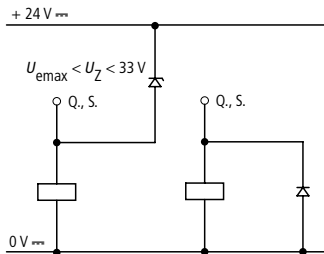


Figure 31: Inductivity with suppressor circuit

Behaviour with short-circuit/overload

Should a short circuit or overload occur on a transistor output, this output will switch off. The output will switch on up to maximum temperature after the cooling time has elapsed. This time depends on the ambient temperature and

the current involved. If the fault condition persists, the output will keep switching off and on until the fault is corrected or until the power supply is switched off (→ Section "Monitoring of short-circuit/overload with EASY...-D.-T..", Page 297).

Connecting analog outputs

EASY820-DC-RC and EASY822-DC-TC each have an analog output QA 01, 0 V to 10 V DC, 10 bit resolution (0 to 1023). The analog output allows you to control servo valves and other final controlling elements.



Caution!

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

Connecting servo valves

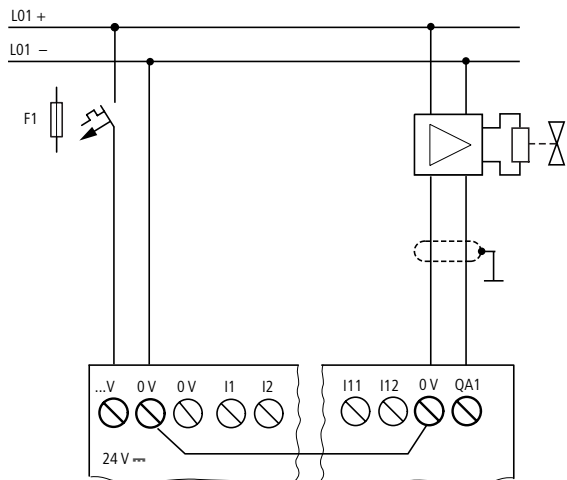


Figure 32: Connecting servo valves

Setpoint entry for a drive

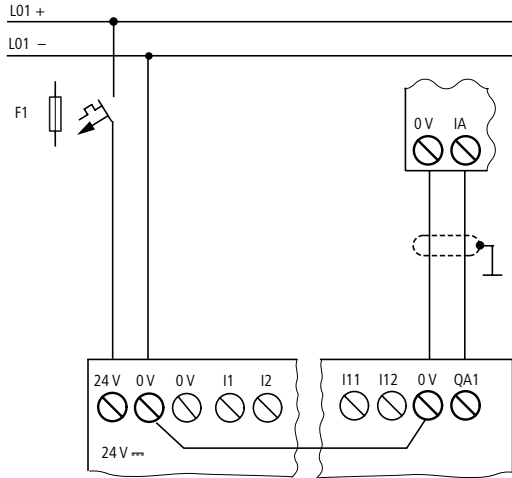


Figure 33: Setpoint definition for a drive

Connecting the NET network

easy800 enables the installation and configuration of the NET network. Up to eight devices can be connected to this network. Further information can be found in the Chapter “easy-NET network”, Page 247.

Accessories

Connection plug:

8-pole RJ45, EASY-NT-RJ45

Connection assignment of the RJ45 socket on the device

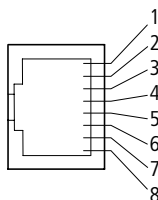


Figure 34: RJ45 socket

Connection cable:

4-pair twisted cable; → chapter "Technical data",
Page 330

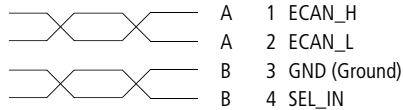


Figure 35: Connection assignment

ECAN_H data cable, pin 1, cable pair A

ECAN_L data cable, pin 2, cable pair A

Ground cable GND, pin 3, cable pair B

Select cable SEL_IN, pin 4, cable pair B



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

Table 3: Prefabricated cables, RJ45 plug on both ends

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

Material for self-manufactured cables

100 m 4 × 0.18 mm²: EASY-NT-CAB

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

Bus termination resistor

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

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

Cable length and cross-sections

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

Cable length m	Cable resistance mΩ/m	Cross-section	
		mm ²	AWG
up to 40	≦ 140	0,13	26
up to 175	≦ 70	0.25 to 0.34	23, 22
up to 250	≦ 60	0.34 to 0.5	22, 21, 20
up to 400	≦ 40	0.5 to 0.6	20, 19
up to 600	≦ 26	0.75 to 0.8	18
up to 1000	≦ 16	1,5	16

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

Calculating the cable length with known cable resistance

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

Baud rate kBaud	Cable resistance R_L Ω
10 to 125	≤ 30
250	≤ 25
500 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 network.

l = cable length in m

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

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

$$S_{min} = \frac{l \times \rho_{cu}}{12,4}$$



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

Calculating length with known cable cross-section

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

l_{max} = cable length in m

S = minimum cable cross-section in mm^2

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

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

Plugging and unplugging network cables

easy800 is provided with two RJ45 network sockets.

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

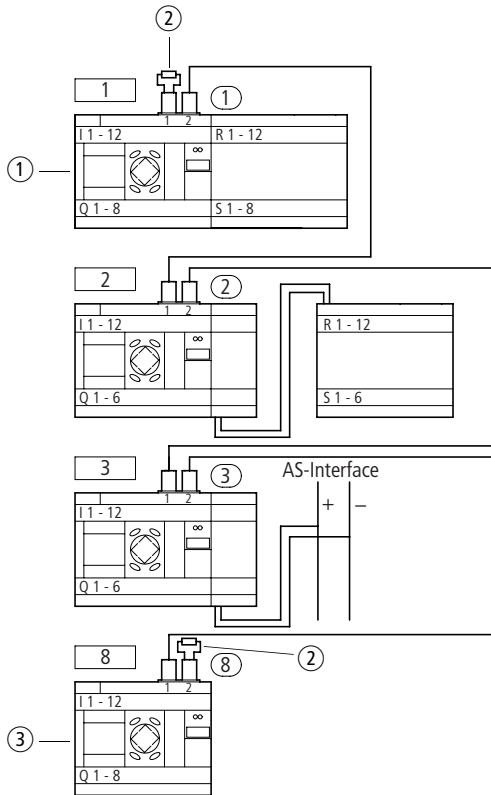


Figure 36: Bus termination resistors

- ① First easy800 in the NET
- ② Bus termination resistor
- ③ Last easy800 in the NET
- ▭ Physical location, place
- Station number

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

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

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

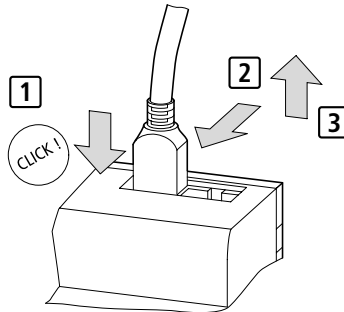


Figure 37: Plugging and unplugging cables

Expanding inputs/outputs You can add expansion units to all easy800 in order to increase the number of inputs and outputs:

Expandable easy basic units	Expansion units	
EASY8...-R.. EASY8...-T..	EASY618...-RE	<ul style="list-style-type: none"> • 12 AC inputs, • 6 relay outputs
	EASY620...-TE	<ul style="list-style-type: none"> • 12 DC inputs, • 8 transistor outputs
	EASY202-RE	2 relay outputs, common ¹⁾
Special expansion units for connecting to other bus systems are shown in the latest product catalogue.		

1) Common supply for multiple outputs

Local expansion

Local expansion units are connected directly next to the basic unit.

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

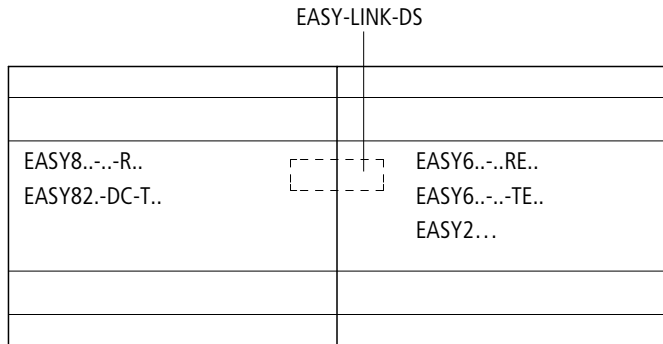


Figure 38: Connecting local expansions with easy800



The following electrical separation is implemented between the EASY8...-C. basic unit and the expansion device (separation always in local connection of expansion unit)

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

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



Basic unit and expansion unit can be provided with different DC power supplies.

Remote expansion

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



Warning!

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

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

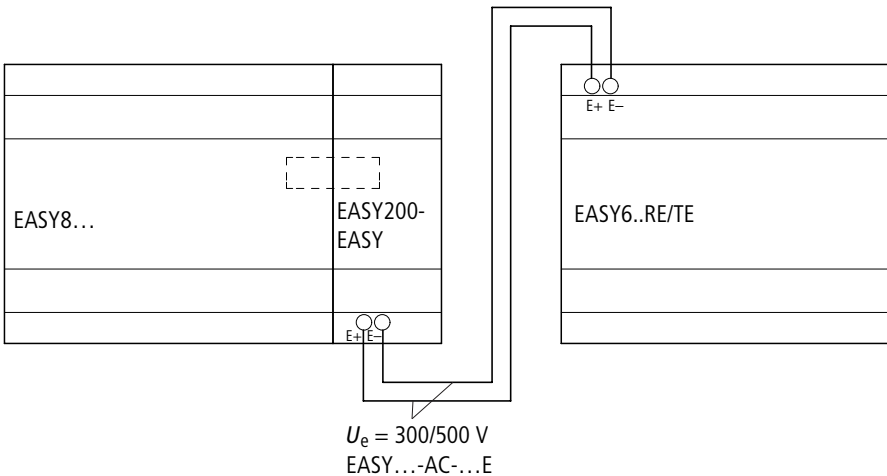


Figure 39: Connecting remote expansions to the easy800



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

3 Commissioning

Switching on

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

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

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

Setting the menu language

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



- ▶ Use the cursor buttons \wedge or \vee to select the language required.
 - English
 - Deutsch
 - French
 - Spanish
 - Italian
 - Portuguese
 - Dutch
 - Swedish
 - Polish
 - Turkish

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

easy will then switch to the Status display.



You can change the language setting at a later date, if you wish, see → Section “Changing the menu language”, Page 270.

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

easy operating modes

easyeasy has two operating modes - RUN and STOP.

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



Caution!

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

The following applies to devices without display/operating unit:

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

If the device does not contain a circuit diagram, the circuit diagram on the memory card is loaded automatically and the device processes the circuit diagram immediately in RUN mode.

Creating your first circuit diagram

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

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

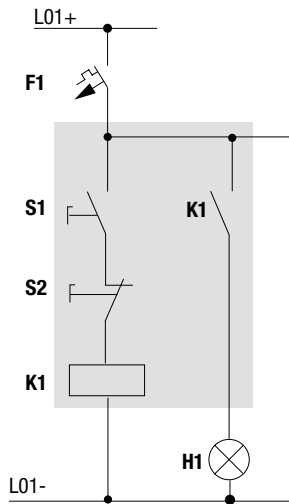


Figure 40: Lamp controller with relays

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

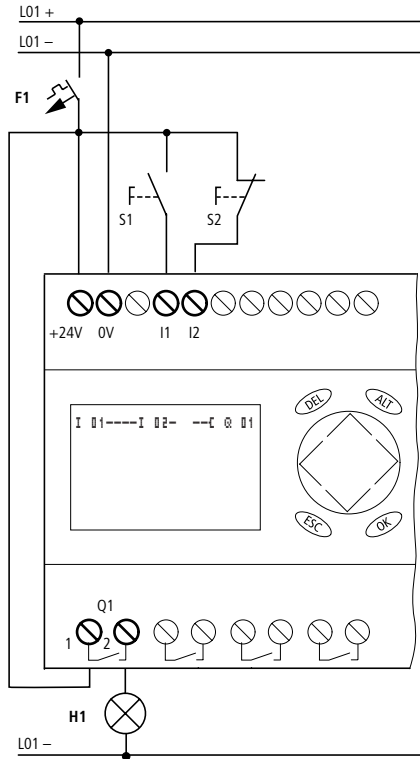


Figure 41: Lamp controller with easy

Starting point Status display

```

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

When you switch on easy, it opens the Status display immediately to show the switching state of the inputs and outputs. It also indicates whether easy is already running a circuit diagram.



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

```

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

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

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



OK has two other functions:

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

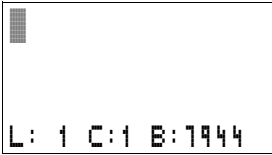
In this case easy must be in STOP mode.

```

CIRCUIT DIAGRAM
FUNCTION RELAYS
  
```

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

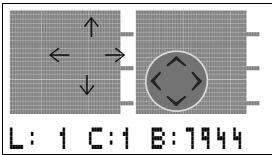
Circuit diagram display



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

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

The easy800 circuit diagram supports 4 contacts and one coil in series. The easy800 display can display 6 circuit diagram contact fields.



Use the ^ v < > cursor buttons to move the cursor over the invisible circuit diagram grid.

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

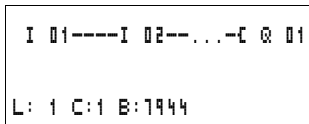


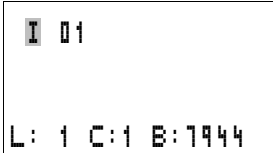
Figure 42: Circuit diagram with inputs I1, I2 and output Q1

► Now try to wire up the following easy diagram.

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

From the first contact to the output coil

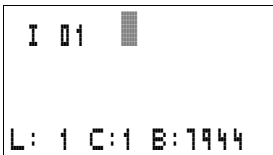
With easy, you work from the input to the output. The first input contact is **I 01**.



► Press **OK**.

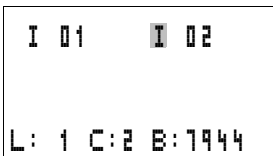
easy proposes the first contact **I 01** at the cursor position.

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



► Press **OK 2** \times , to move the cursor across the **01** to the second contact field.

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



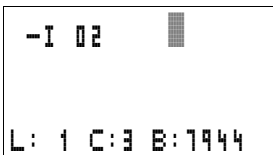
► Press **OK**.

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

► Press **OK** so that the cursor jumps to the next position and press cursor button \wedge or \vee to change the number to **02**.




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



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

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

Wiring

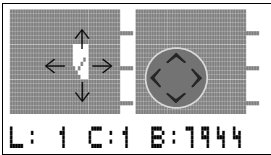
easy displays a small arrow  in the circuit diagram when creating the wiring.

Press **ALT** to activate the wiring arrow cursor and use the cursor buttons $\wedge \vee \lt \gt$ to move it.



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

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

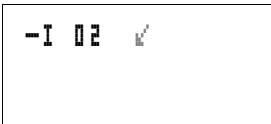


The wiring arrow works between contacts and relays. When you move the arrow onto a contact or relay coil, it changes back to the cursor and can be reactivated with **ALT** if required.



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

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



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

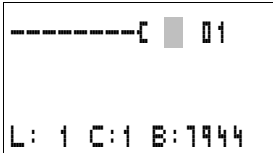
- ▶ Press the cursor button \gt . Contact **I 02** will be connected up to the coil field.



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

- ▶ Press the cursor button \triangleright again.

The cursor will move to the coil field.



- ▶ Press **OK**.

easy inserts the relay coil $\text{C } 01$. The specified coil function C and the output relay $\text{C } 01$ are correct and do not have to be changed.

Your first working easy circuit diagram now looks like this:

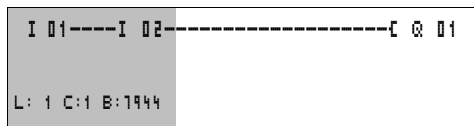



Figure 43: Your first circuit diagram

 = visible area

- ▶ Press **ESC** to leave the circuit diagram display. The SAVE menu appears.

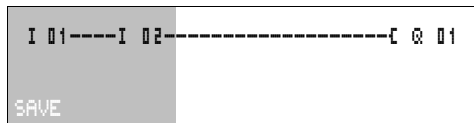



Figure 44: SAVE menu

 = visible area

- ▶ Press the **OK** button.

The circuit diagram is stored.

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

Testing the circuit diagram

```

PROGRAM...
STOP / RUN
PARAMETERS
SET CLOCK

```

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

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

easy runs in the mode indicated by the tick.

► Press the **OK** button. easy will change to RUN mode.



The mode assigned with the tick is always active.

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

```

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

```

► Change to the Status display and press push-button actuator S1.

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

Power flow display

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

► Change to the Circuit diagram display and press push-button actuator S1.


The relay picks up. easy shows the power flow.

```

I 01====I 02=====Q 01
L: 1 C:1 RUN

```

Figure 45: Power flow display: Inputs I1 and I2 are closed, relay Q1 has picked up

 = visible area

- ▶ Press push-button actuator S2, that has been connected as a break contact.
- The circuit connection is interrupted and relay Q1 drops out.

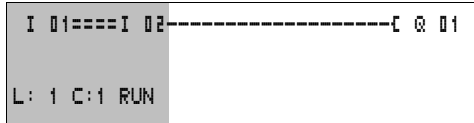


Figure 46: Power flow display: Input I1 is closed, input I2 is open, relay Q1 has picked up

■ = visible area

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



With easy you can test parts of a circuit diagram before it is entirely completed.
 easy simply ignores any incomplete wiring that is not yet working and only runs the finished wiring.

Power flow display with Zoom function

easy enables you to check the following at a glance:

- all four contacts plus one coil in series
- and 3 circuit connections

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

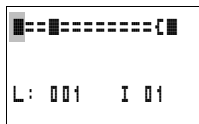
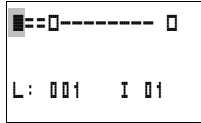


Figure 47: Power flow display in Zoom function: Input I1 and I2 are closed, relay Q1 picked up

- Contact closed, coil is triggered:
- Contact opened, coil dropped out

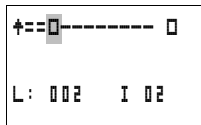
- ▶ Press push-button actuator S2, that has been connected as a break contact.

The rung is interrupted and relay Q1 drops out.



Use the cursor buttons ^ v < > to move between the contacts or coil.

- ▶ Press the cursor button >.



The cursor moves to the second contact.

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

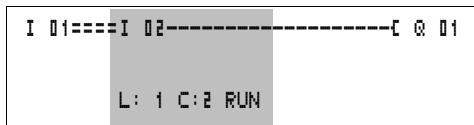


Figure 48: Power flow display: Input I1 is closed, input I2 is open, relay Q1 has picked up

■ = visible area

Deleting the circuit diagram

- ▶ Switch the easy to STOP mode.



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

- ▶ Use PROGRAM... to switch from the main menu to the next menu level.
- ▶ Select DELETE PROGRAM

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

The easy will display the prompt DELETE?

- ▶ Press **OK** to delete the program or **ESC** to cancel.
- ▶ Press **ESC** to return to the Status display.

Fast circuit diagram entry

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

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

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

Configuring an easy-NET network

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

Proceed as follows:

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

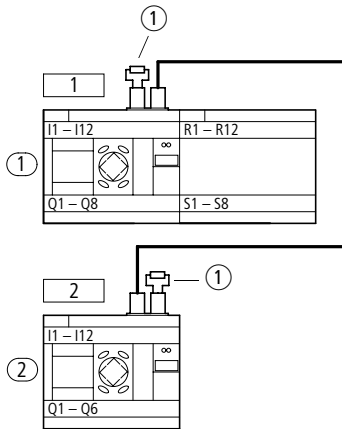


Figure 49: Example topology with two easy-NET stations

① Network termination resistor

□ Physical location

○ Station number

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



The following tasks are only possible in STOP mode.

Entering the network station number

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

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

The System menu appears

Select the CONFIGURATOR menu option.

- ▶ Press the **OK** button.

```
NET..
.
```

The NET menu appears.

- ▶ Press the **OK** button.

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

The NET PARAMETERS... menu appears.

- ▶ Press the **OK** button.

```
NET-ID : 00 +
BAUDRATE: 125KB
BUSDELAY: 00
SEND IO / +
REMOTE RUN
REMOTE IO
```

- ▶ Press the **OK** button and select the station number with \wedge and \vee . In this case the station number (NET-ID) "01".

- ▶ Press the **OK** button.

```
NET-ID : 01 +
BAUDRATE: 125KB
BUSDELAY: 00
SEND IO / +
REMOTE RUN
REMOTE IO
```

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



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

Entering network stations

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



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

- ▶ Use the ^ and v cursor buttons to select the STATION menu and press the **OK** button.
- ▶ Proceed to the station with physical address 2.
- ▶ Select the required physical location with the ^ and v cursor buttons. Press the **OK** button.
- ▶ Use cursor buttons ^ and v to select station number 2.
- ▶ Press the **OK** button.

1	1	+
2	0	
3	0	
4	0	+

1	1	+
2	2	
3	0	
4	0	+

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

- ▶ Press **ESC** to return to the STATION menu item.

Configuring an easy-NET network

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

Requirement:

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

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



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

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

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

```
CONFIGURE ?
```

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

► Press the **OK** button.

```
CONFIGURATION
IN
PROGRESS.
```

The message on the left appears:

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

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



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

```
ERR: ID-KONFLIKT
CONFIGURATION
OVERWRITE?
```

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

Changing the easy-NET network configuration

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

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

Station addresses in the STATIONS menu are changed as follows:

- ▶ Go to the physical location which is to be modified.
- ▶ Press the **OK** button.



Existing station numbers can only be modified to free, non-assigned station numbers. If all eight numbers are assigned, all station numbers which are to be modified must be set to zero. Thereafter, all station numbers can be reassigned. (easy800 sets all station numbers to zero which are assigned with a physical location behind the leading zero.)

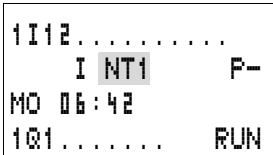
- ▶ Select the required station number with the \wedge and \vee cursor buttons and confirm your input with the **OK** button.
- ▶ Configure all easy-NET stations again using the CONFIGURATION menu.



Further information concerning the easy-NET network topic can be found in Chapter "easy-NET network", Page 247.

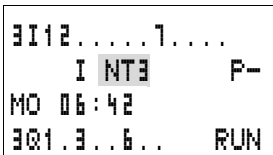
Displaying the Status display of other stations

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



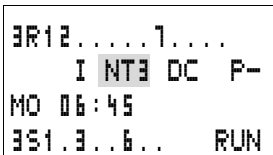
► Change to the Status display and press the **ESC** button.

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



► Change to the number of the required station with the ^ and v cursor buttons.

► Press the **OK** button.



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

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



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

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

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

4 Wiring with easy800

This chapter describes all the functions available with easy800.

easy800 operation

Buttons for drawing circuit diagrams and function block usage



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



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



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

Cursor buttons set as P buttons:

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



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



Change, add contact/relay
Save setting




Terminal mode on/off

Operation

The cursor buttons in the easy800 circuit diagram perform three functions. The current mode is indicated by the appearance of the flashing cursor.

- Move
- Entering
- Connect

 In Move mode you can use $\wedge \vee \langle \rangle$ to move the cursor around the circuit diagram in order to select a rung, contact or relay coil.



Use **OK** to switch to Entry mode so that you can enter or change a value at the current cursor position. If you press **ESC** in Entry mode, easy800 will undo the most recent changes.



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

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



easy800 performs many of these cursor movements automatically. For example, easy800 switches the cursor to Move mode if no further entries or connections are possible at the selected cursor position.

Opening the parameter display for function blocks with contacts or relays

If you specify the contact or coil of a function relay type in Entry mode, easy800 automatically switches from the contact number to the function block parameter display when you press **OK**.

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

Program

A program is a sequence of commands which the easy800 executes cyclically in RUN mode. An easy800 program consists of the necessary settings for the device, easy-NET, password, system settings, a circuit diagram and/or function blocks.

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

Function blocks

Function blocks are program elements with special functions. Example: timing relays, time switches, arithmetic function blocks. Function blocks are provided with or without contacts and coils accordingly. 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

Relays

Relays are switching devices which are electronically simulated in easy800. They actuate their contacts according to their designated function. A relay consists of a coil and at least one contact.



Contacts

You modify the current flow with the contacts in the easy800 circuit diagram. Contacts such as make contacts carry a 1 signal when closed and 0 when open. Every contact in the easy800 circuit diagram can be defined as either a make contact or a break contact.

Coils

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

Table 4: Usable contacts

Contact	easy800 display
 Make contact, open in release position	I, Q, M, A, ... Other contacts → Table
 Break contact, closed in released position	$\bar{I}, \bar{Q}, \bar{M}, \bar{A}, \dots$ Other contacts → Table

easy800 works with different contacts, which can be used in any order in the contact fields of the circuit diagram.

Table 5: Contacts

Contact	Make contact	Break contact	Number	Page
Inputs				
Inputs of a network station * = Station address 1 to 8	$*I$	$*\bar{I}$	01...12	250
easy800 input terminal	I	\bar{I}	01...12	—
Cursor button	P	\bar{P}	01...04	—
Network station expansion input terminal * = Station address 1 to 8	$*R$	$*\bar{R}$	01...12	250
Expansion input terminal	R	\bar{R}	01...12	—
Bit inputs via the network * = Station address 1 to 8	$*RN$	$*\bar{RN}$	01...32	250

Contact	Make contact	Break contact	Number	Page
Diagnostics inputs				
Network station expansion status * = Station address 1 to 8	*I	* \bar{I}	14	299
Network station short-circuit/ overload * = Station address 1 to 8	*I	* \bar{I}	15...16	297
Expansion status	I	\bar{I}	14	299
Short-circuit/overload	I	\bar{I}	15...16	297
Short-circuit/overload in expansion network station * = Station address 1 to 8	*R	* \bar{R}	15...16	297
Short circuit/overload with expansion	R	\bar{R}	15...16	297
Outputs				
easy800 output easy network station * = Station address 1 to 8	*Q	* \bar{Q}	01...08	250
easy800 output	Q	\bar{Q}	01...08	—
easy800 output expansion with network station * = Station address 1 to 8	*S	* \bar{S}	01...08	250
easy800 output expansion	S	\bar{S}	01...08	—
Bit outputs via the network * = Station address 1 to 8	*SN	* $\bar{S}\bar{N}$	01...32	250
Other contacts				
Marker relay contact	M	\bar{M}	01...96	97
Jump label	:		01...32	208
Diagnostics messages	ID	ID	01...16	260
Function blocks				
Analog value comparator function block	A X Q1	A X Q1	X=01...32	125
Arithmetic value overflow (CARRY) function block	AR X CV	$\bar{A}\bar{R}$ X CV	X=01...32	128

Contact	Make contact	Break contact	Number	Page
Zero arithmetic value (zero) function block	AR X ZE	$\bar{A}R$ X ZE	X=01...32	128
Data block comparator function block, error: number of elements exceeded	BC X E1	$\bar{B}C$ X E1	X=01...32	132
Data block comparator function block, error: range overlap	BC X E2	$\bar{B}C$ X E2	X=01...32	132
Data block comparator function block, error: invalid offset	BC X E3	$\bar{B}C$ X E3	X=01...32	132
Data block comparator function block, comparison result	BC X E0	$\bar{B}C$ X E0	X=01...32	139
Data block comparator function block, error: number of elements exceeded	BT X E1	$\bar{B}T$ X E1	X=01...32	139
Data block transfer function block, error: range overlap	BT X E2	$\bar{B}T$ X E2	X=01...32	139
Data block transfer function block, error: invalid offset	BT X E3	$\bar{B}T$ X E3	X=01...32	139
Boolean sequence function block, value zero	BV X ZE	$\bar{B}V$ X ZE	X=01...32	150
Counter function block, upper setpoint value exceeded (Overflow)	C X OF	\bar{C} X OF	X=01...32	153
Counter function block, lower setpoint value undershot (Fall below)	C X FB	\bar{C} X FB	X=01...32	153
Counter function block, actual value equal to zero	C X ZE	\bar{C} X ZE	X=01...32	153
Counter function block, actual value has exceeded counter range (CARRY)	C X CV	\bar{C} X CV	X=01...32	153
Frequency counter function block, upper setpoint value exceeded (Overflow)	CF X OF	$\bar{C}F$ X OF	X=01...04	160

Contact	Make contact	Break contact	Number	Page
Frequency counter function block, lower setpoint value undershot (Fall below)	CF X FB	\overline{CF} X FB	X=01...04	160
Frequency counter function block, actual value equal to zero	CF X ZE	\overline{CF} X ZE	X=01...04	160
High-speed counter function block, upper setpoint value exceeded (Overflow)	CH X OF	\overline{CH} X OF	X=01...04	164
High-speed counter function block, lower setpoint value undershot (Fall below)	CH X FB	\overline{CH} X FB	X=01...04	164
High-speed counter function block, actual value equal to zero	CH X ZE	\overline{CH} X ZE	X=01...04	164
High-speed counter function block, actual value has exceeded counter range (CARRY)	CH X CV	\overline{CH} X CV	X=01...04	164
Incremental encoder counter function block, upper setpoint value exceeded (Overflow)	CI X OF	\overline{CI} X OF	X=01...02	170
Incremental encoder counter function block, lower setpoint value undershot (Fall below)	CI X FB	\overline{CI} X FB	X=01...02	170
Incremental encoder counter function block, actual value equal to zero	CI X ZE	\overline{CI} X ZE	X=01...02	170
Incremental encoder counter function block, actual value has exceeded counter range (CARRY)	CI X CV	\overline{CI} X CV	X=01...02	170
Comparator function block, less than	CP X LT	\overline{CP} X LT	X=01...32	175

Contact	Make contact	Break contact	Number	Page
Comparator function block, equal to	CP X EQ	CP̄ X EQ	X=01...32	175
Comparator function block, greater than	CP X GT	CP̄ X GT	X=01...32	175
Text output function block	D X Q1	D̄ X Q1	X=01...32	177
Data function block	DB X Q1	DB̄ X Q1	X=01...32	181
PID controller, value range of manipulated variable exceeded	DC X LI	DC̄ X LI	X=01...32	183
Receive a variable from a station (Get)	GT X Q1	GT̄ X Q1	X=01...32	183
Seven-day time switch	HW X Q1	HW̄ X Q1	X=01...32	194
Year time switch function block	HV X Q1	HV̄ X Q1	X=01...32	200
Master reset, sets all outputs and markers to zero state	MR X Q1	MR̄ X Q1	X=01...32	211
Operating hours counter function block, set time reached	OT X Q1	OT̄ X Q1	X=01...04	218

Contact	Make contact	Break contact	Number	Page
Operating hours counter, value overflow (CARRY)	OT X CV	\overline{OT} X CV	X=01...04	218
Send a variable to the network, enable active Put	PT X 01	\overline{PT} X 01	X=01...32	219
Pulse width modulation, error minimum on or off time exceeded	PW X E1	\overline{PW} X E1	X=01...02	221
Send date and time via the network (easy-NET) function block	SC X 01	\overline{SC} X 01	X=01	224
Timing relay function block	T X 01	\overline{T} X 01	X=01...32	228

Usable relays and function blocks (coils)

easy800 various relay types as well as function blocks and their coils for wiring in a circuit diagram.

Relay/function block	easy800 display	Number	Coil	Parameter
Outputs				
easy800 output relays, network stations (only network master) * = Station address 2 to 8	*Q	01...08	✓	–
easy800 output relay	Q	01...08	✓	–
easy800 output relay expansion, network stations (only network master) * = Station address 2 to 8	*S	01...08	✓	–
easy800 expansion output relay	S	01...08	✓	–
Bit outputs * = Station address 1 to 8	*SM	01...32	✓	–
General coils				
Marker, auxiliary relay	M	01...96	✓	–
Jump label	:	01...32	✓	–
Function blocks				
Analog value comparator function block	A	01...32	–	✓
Arithmetic function block	AR	01...32	–	✓
Data block comparator, activate	BC X EN	01 32	✓	✓
Transfer data block, trigger coil	BT X T_	01 32	✓	✓
Boolean sequence	BV	01...32	–	✓
Counter function block, counter input	C X C_	X=01...32	✓	✓
Counter function block, direction	C X D_	X=01...32	✓	✓
Counter function block, set counter value (Preset)	C X SE	X=01...32	✓	✓

Relay/function block	easy800 display	Number	Coil	Parameter
Counter function block, reset counter value	C X RE	X=01...32	✓	✓
Frequency counter function block, activate counter (enable)	CF X EN	X=01...04	✓	✓
High-speed counter function block, direction	CH X D_	X=01...04	✓	✓
High-speed counter function block, activate counter (enable)	CH X EN	X=01...04	✓	✓
High-speed counter function block, set counter value (Preset)	CH X SE	X=01...04	✓	✓
High-speed counter function block, reset counter value	CH X RE	X=01...04	✓	✓
Incremental encoder counter function block, set counter value (Preset)	CI X SE	X=01...02	✓	✓
Incremental encoder counter function block, activate counter (enable)	CI X EN	X=01...02	✓	✓
Incremental encoder counter function block, reset counter value	CI X RE	X=01...02	✓	✓
Comparator function block	CP	X=01...32	–	✓
Activate text output function block (enable)	D X EN	X=01...32	✓	✓
Data function block, trigger coil	DB X T_	X=01...32	✓	✓
PID controller, activate	DC X EN	X=01...32	✓	✓
PID controller, activate P component	DC X EP	X=01...32	✓	✓
PID controller, activate I component	DC X EI	X=01...32	✓	✓
PID controller, activate D component	DC X ED	X=01...32	✓	✓
PID controller, accept manual manipulated variable	DC X SE	X=01...32	✓	✓
Activate signal smoothing filter	FT X EN	X=01...32	✓	✓
Get from network station function block	GT	X=01...32	–	✓

Relay/function block	easy800 display	Number	Coil	Parameter
Seven-day time switch	HW	X=01...32	–	✓
Year time switch function block	HY	X=01...32	–	✓
Activate value scaling function block	LS X EN	X=01...32	✓	✓
Master reset function block	MR X T_	X=01...32	✓	✓
Activate numerical converter function block	NC X EN	X=01 32	✓	✓
Function block operating hours counter, enable	OT X EN	X=01...04	✓	✓
Operating hours counter function block, reset	OT X RE	X=01...04	✓	✓
Send to the network (easy-NET) function block, trigger	PT X T_	X=01...32	✓	✓
Activate pulse width modulation function block	PW X EN	X=01...02	✓	✓
Send time to the network (easy-NET) function block, trigger	SC X T_	X=01	✓	–
Activate set cycle time function block	ST X EN	X=01		
Timing relay function block, trigger control coil (enable)	T X EN	X=01...32	✓	✓
Timing relay function block, stop	T X ST	X=01...32	✓	✓
Timing relay function block, reset	T X RE	X=01...32	✓	✓
Activate value limitation function block	VC X EN	X=01...32	✓	✓

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

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

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

Markers, analog operands

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

Table 6: Markers

Markers	easy800 display	Number	Value range	Access type
Analog operand				r = Read w = Write
Marker 32 bit	MD	01...96	32 bit	r, w
Marker 16 bit	MW	01...96	16 bit	r, w
Marker 8 bit	MB	01...96	8 bit	r, w
Marker 1 bit	M	1...96	1 bit	r, w
Analog inputs basic unit	IA X	X=01...04	10 bit	r
Analog output	OA X	X=01	10 bit	r, w

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

Table 7: Composition of the markers

Applies to MD, MW, MB, M	Left = most significant bit, byte, word			Right = least significant bit, byte, word
32 bit	MD1			
16 bit	MW2		MW1	
8 bit	MB4	MB3	MB2	MB1
1 bit	M32 to M25	M24 to M17	M16 to M9	M8 to M1
32 bit	MD2			
16 bit	MW4		MW3	
8 bit	MB8	MB7	MB6	MB5
1 bit	M64 to M57	M56 to M49	M48 to M41	M40 to M33

Applies to MD, MW, MB, M	Left = most significant bit, byte, word			Right = least significant bit, byte, word
32 bit	MD3			
16 bit	MW6		MW5	
8 bit	MB12	MB11	MB10	MB9
1 bit	M96 to M89	M88 to M81	M80 to M73	M72 to M65
32 bit	MD4			
16 bit	MW8		MW7	
8 bit	MB16	MB15	MB14	MB13
32 bit	MD5			
16 bit	MW10		MW9	
8 bit	MB20	MB19	MB18	MB17
...				
...				
...				
32 bit	MD23			
16 bit	MW46		MW45	
8 bit	MB92	MB91	MB90	MB89
32 bit	MD24			
16 bit	MW48		MW47	
8 bit	MB96	MB95	MB94	MB93
32 bit	MD25			
16 bit	MW50		MW49	
32 bit	MD26			
16 bit	MW52		MW51	
...				
...				
32 bit	MD48			
16 bit	MW96		MW95	

Applies to MD, MW, MB, M	Left = most significant bit, byte, word				Right = least significant bit, byte, word
32 bit	MD49				
32 bit	MD50				
...					
32 bit	MD95				
32 bit	MD96				



You should only write the markers once.

Marker double words always contain all data formats. When several write accesses to MD, MW, MB or M (within an MD) are made, it is the last write operation that is retained.

Number formats

easy makes computations with a signed 31 bit value.

The value range is:

-2147483648 to $+2147483647$

The 32nd bit is the sign bit with a 31 bit value.

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

Example:

$0000000000000000000010000010010_{bin} =$

$412_{hex} = 1042_{dec}$

Bit 32 = state 1 means a negative number.

Example:

$11111111111111111101110010101110_{bin} =$

$FFFDCAE_{hex} = -9042_{dec}$

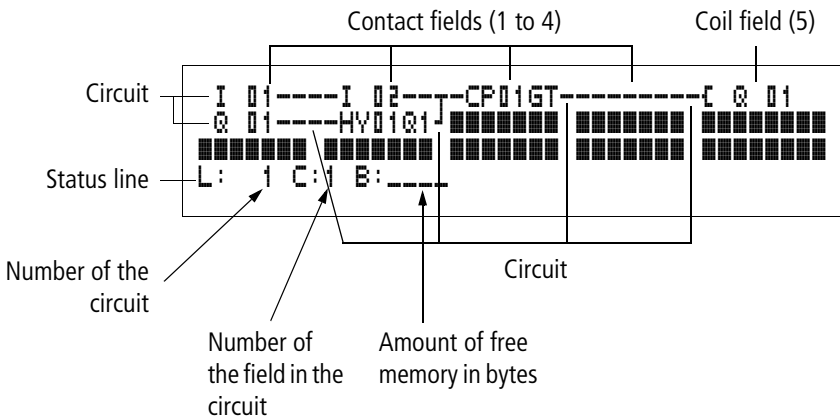


The marker byte (MB) and marker word (MW) number formats are unsigned.

Circuit diagram display

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

- Insert contacts in the four **contact fields**. The first contact field on the left is automatically connected to the voltage.
- Insert the relay coil to be controlled together with its function and designation in the **coil field**. The coil designation consists of the coil name, coil number and function block from the function designation. The coil function defines the method of operation of the coil.
- Every line in the circuit diagram forms a **rung**. With the easy800 up to 256 circuit connections can be wired in a circuit diagram



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

```

I 01----I 02---
Q 01----HY01Q1J
L:  1 C:1 B:7840

```

easy800 circuit diagram display

For improved readability, the circuit diagram display of the easy800 shows two contacts per circuit connection or one contact plus a series coil. A total of 16 characters per circuit connection and three circuit connections plus a status line can be displayed simultaneously.

You can move between the contact fields with the < > cursor buttons. The number of the circuit connection and the contact are displayed in the lower status line.



The circuit diagram display performs two functions:

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

Saving and loading programs

easy800 provides you with two ways of saving circuit diagrams externally:

- By saving to a memory card
- By saving to a PC running EASY-SOFT (-PRO) with EASY .

Once they have been saved, programs can be reloaded into easy800, edited and run.

All program data is saved in easy800. In the event of a power failure the data will be retained until the next time it is overwritten or deleted.

Memory card

Each memory card contains a circuit diagram which is inserted into the easy800 interface.

The easy800 behaves in the following manner in accordance with the type and setting.

Requirement:

A valid circuit diagram must be stored on the card.

Variants with display:

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

CARD MODE setting → page 280.

Variants without display:

If the circuit diagram on the card is different to the circuit diagram in the device, the program from the card is loaded as soon as the power supply is turned on.

EASY-SOFT (-PRO)

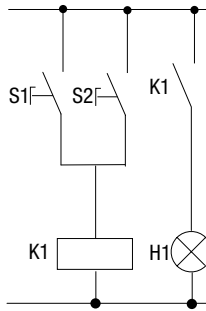
EASY-SOFT (-PRO) is a PC program for creating, testing and managing easy800 and circuit diagrams.

Completed programs are exchanged between the PC and easy800 via the connection cable. Once you have transferred a circuit diagram, simply run easy800 straight from your PC.

Working with contacts and relays

In easy800 circuit diagrams, the switches, buttons and relays of conventional circuit diagrams are connected up using input contacts and relay coils.

Conventional circuit

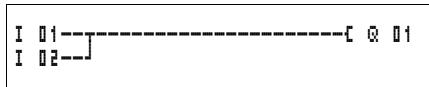


easy800 circuit diagram

easy800 connection

- Connect make contact S1 to input terminal I1
 - Connect make contact S2 to input terminal I2
 - Connect load H1 to output Q1
- S1 or S2 switches on H1.

easy800 circuit diagram:



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

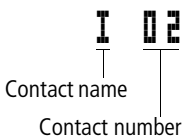
The signal states on the input terminals are detected in the circuit diagram with the input contacts I, R* or RN. The outputs are switched in the circuit diagram with the output relays Q, S or SN.

Entering and changing contacts and relay function coils

Contacts

An input contact is selected in the easy800 via the contact name and contact number.

Example: input contact



CP01GT

Contact name
Contact number
Contact function

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

Example: contact comparator function block

2RND2

Station address
Contact name
Contact number

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

Example: contact of an easy-NET station

S Q 04

Coil function
Coil name
Coil number

Coils

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

Example: relay coil output

C T 04EN

Coil function
Coil name
Coil number
Function block

Relay coil function block - timing relay with control coil

S2SN04

Coil function
Station address
Coil name
Coil number

Relay coil of an easy-NET station



A full list of all the contacts and relays is given in the overview starting on Page 88.

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



easy800 proposes the contact **I 01** or the coil **C 01** when starting entries in an empty field.

- ▶ Move the cursor using the buttons < > ^ v to a contact or coil field.
- ▶ Press **OK** to switch to Entry mode.
- ▶ Use < **I** to select the position you wish to change, or press **OK** to jump to the next position.
- ▶ Use ^ v to modify the value at the position.

easy800 will leave Entry mode when you press < > or **OK** to leave a contact field or coil field.

In the contact field to change I 01 to I 02	In the coil field to change C 01 to S 08																																																																					
<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%; border-bottom: 1px solid black;">I 01</td> <td style="width: 33%; border-bottom: 1px solid black;">I 01</td> <td style="width: 33%; border-bottom: 1px solid black;">I 02</td> </tr> <tr> <td style="border-bottom: 1px solid black;">Q</td> <td style="border-bottom: 1px solid black;">02</td> <td style="border-bottom: 1px solid black;">^</td> </tr> <tr> <td style="border-bottom: 1px solid black;">M</td> <td style="border-bottom: 1px solid black;">03</td> <td style="border-bottom: 1px solid black;">></td> </tr> <tr> <td style="border-bottom: 1px solid black;">HW</td> <td style="border-bottom: 1px solid black;">04</td> <td style="border-bottom: 1px solid black;">or OK</td> </tr> <tr> <td style="border-bottom: 1px solid black;">C</td> <td style="border-bottom: 1px solid black;">05</td> <td style="border-bottom: 1px solid black;">></td> </tr> <tr> <td style="border-bottom: 1px solid black;">T</td> <td style="border-bottom: 1px solid black;">.</td> <td style="border-bottom: 1px solid black;">></td> </tr> <tr> <td style="border-bottom: 1px solid black;">P</td> <td style="border-bottom: 1px solid black;">.</td> <td style="border-bottom: 1px solid black;">></td> </tr> <tr> <td style="border-bottom: 1px solid black;">D</td> <td style="border-bottom: 1px solid black;">.</td> <td style="border-bottom: 1px solid black;">></td> </tr> <tr> <td style="border-bottom: 1px solid black;">S</td> <td style="border-bottom: 1px solid black;">99</td> <td style="border-bottom: 1px solid black;">></td> </tr> <tr> <td style="border-bottom: 1px solid black;">:</td> <td style="border-bottom: 1px solid black;"></td> <td style="border-bottom: 1px solid black;">or OK</td> </tr> <tr> <td style="border-bottom: 1px solid black;">...</td> <td style="border-bottom: 1px solid black;"></td> <td style="border-bottom: 1px solid black;"></td> </tr> </table>	I 01	I 01	I 02	Q	02	^	M	03	>	HW	04	or OK	C	05	>	T	.	>	P	.	>	D	.	>	S	99	>	:		or OK	...			<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%; border-bottom: 1px solid black;">C 01</td> <td style="width: 25%; border-bottom: 1px solid black;">C 01</td> <td style="width: 25%; border-bottom: 1px solid black;">C 01</td> <td style="width: 25%; border-bottom: 1px solid black;">C 08</td> </tr> <tr> <td style="border-bottom: 1px solid black;">J</td> <td style="border-bottom: 1px solid black;"><</td> <td style="border-bottom: 1px solid black;">M</td> <td style="border-bottom: 1px solid black;">></td> </tr> <tr> <td style="border-bottom: 1px solid black;">S</td> <td style="border-bottom: 1px solid black;"><</td> <td style="border-bottom: 1px solid black;">T</td> <td style="border-bottom: 1px solid black;">or OK</td> </tr> <tr> <td style="border-bottom: 1px solid black;">R</td> <td style="border-bottom: 1px solid black;"><</td> <td style="border-bottom: 1px solid black;">C</td> <td style="border-bottom: 1px solid black;">></td> </tr> <tr> <td style="border-bottom: 1px solid black;">P</td> <td style="border-bottom: 1px solid black;"><</td> <td style="border-bottom: 1px solid black;">D</td> <td style="border-bottom: 1px solid black;">02</td> </tr> <tr> <td style="border-bottom: 1px solid black;">L</td> <td style="border-bottom: 1px solid black;"><</td> <td style="border-bottom: 1px solid black;">S</td> <td style="border-bottom: 1px solid black;">03</td> </tr> <tr> <td style="border-bottom: 1px solid black;">J</td> <td style="border-bottom: 1px solid black;"><</td> <td style="border-bottom: 1px solid black;">:</td> <td style="border-bottom: 1px solid black;">08</td> </tr> <tr> <td style="border-bottom: 1px solid black;">:</td> <td style="border-bottom: 1px solid black;"></td> <td style="border-bottom: 1px solid black;">...</td> <td style="border-bottom: 1px solid black;">></td> </tr> <tr> <td style="border-bottom: 1px solid black;">...</td> <td style="border-bottom: 1px solid black;"></td> <td style="border-bottom: 1px solid black;">....</td> <td style="border-bottom: 1px solid black;">or OK</td> </tr> </table>	C 01	C 01	C 01	C 08	J	<	M	>	S	<	T	or OK	R	<	C	>	P	<	D	02	L	<	S	03	J	<	:	08	:		...	>	or OK
I 01	I 01	I 02																																																																				
Q	02	^																																																																				
M	03	>																																																																				
HW	04	or OK																																																																				
C	05	>																																																																				
T	.	>																																																																				
P	.	>																																																																				
D	.	>																																																																				
S	99	>																																																																				
:		or OK																																																																				
...																																																																						
C 01	C 01	C 01	C 08																																																																			
J	<	M	>																																																																			
S	<	T	or OK																																																																			
R	<	C	>																																																																			
P	<	D	02																																																																			
L	<	S	03																																																																			
J	<	:	08																																																																			
:		...	>																																																																			
...		or OK																																																																			

Deleting contacts and coils

- ▶ Move the cursor using the buttons < > ^ v to a contact or coil field.
- ▶ Press **DEL**.

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

Changing make contacts to break contacts

Every contact in the easy800 circuit diagram can be defined as either a make contact or a break contact.

- ▶ Switch to Entry mode and move the cursor over the contact name.
- ▶ Press **ALT**. The make contact will change to a break contact.
- ▶ Press **OK** 2 x to confirm the change.

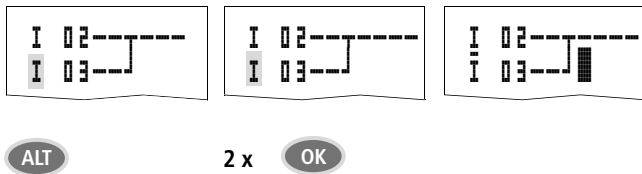


Figure 50: Change contact I 03 from make to break

Creating and modifying connections

Contacts and relay coils are connected with the arrow in the Connect mode. easy800 displays the cursor in this mode as an arrow.

- ▶ Use < > ^ v to move the cursor onto the contact field or coil field from which you wish to create a connection.



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

- ▶ Press **ALT** to switch to Connect mode.
- ▶ Use < > to move the diagonal arrow between the contact fields and coil fields and ^ v to move between rungs.
- ▶ Press **ALT** to leave Connect mode.

easy800 will leave the mode automatically when you move the diagonal arrow onto a contact field or coil field which has already been assigned.



In a circuit connection, easy800 automatically connects contacts and the terminal to the relay coil if there are no empty fields in-between.

Never work backwards. You will learn why wiring backwards does not work in Section "Effects on the creation of the circuit diagram" on Page 290.

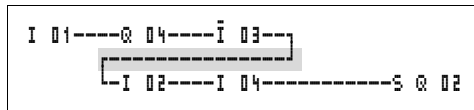


Figure 51: Circuit diagram with five contacts, invalid

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

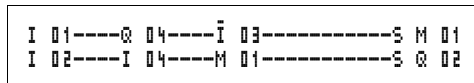


Figure 52: Circuit diagram with M marker relay

Deleting connections

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

easy800 will delete a connection. Closed adjacent connections will be retained.

If several circuit connections are connected to one another, easy800 first deletes the vertical connection. If you press **DEL** again, it will delete the horizontal connection as well.



You cannot delete connections that easy800 has created automatically.

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

Inserting and deleting a circuit connection

The easy800 circuit diagram display shows three of the 256 circuit connections on the display at the same time. easy800 automatically scrolls up or down the display to show hidden circuit connections – even empty ones – if you move the cursor past the top or bottom of the display.

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

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

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

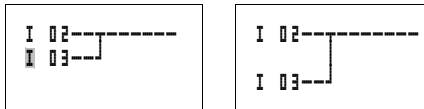
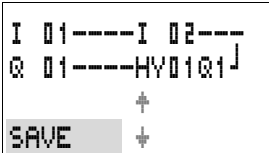


Figure 53: Inserting a new circuit connection

Saving circuit diagrams

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

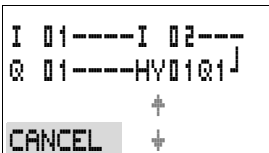


The menu on the left appears in the status line.

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

After saving you will be in the CIRCUIT DIAGRAM menu.

Aborting circuit diagram entry

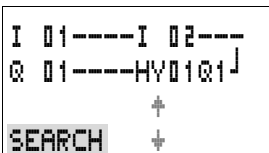


- ▶ If you want to exit without saving the circuit diagram, press **ESC**.
- ▶ Use the cursor buttons $\wedge \vee$ to select the CANCEL menu.
- ▶ Press **OK**.

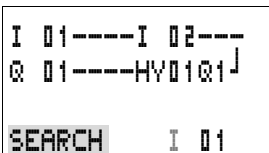
The circuit diagram is closed without saving.

Searching for contacts and coils

You search for contacts and coils in the following manner:



- ▶ Press **ESC**. Use the cursor buttons $\wedge \vee$ to select the SEARCH menu.
- ▶ Press **OK**.



- ▶ Select the desired contact, coil and number with the \vee and $\langle \rangle$ cursor buttons.

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

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

```

I 01----I 02---
Q 01----HYD1Q1J
L: 1 C:1 B:1140

```

The first contact or coil from the call position to the end of the circuit diagram is sought. If no contact or coil is found, the easy800 circuit diagram editor will continue the search from the start of the circuit diagram. If a contact or coil is found, the easy800 editor automatically jumps to the respective field in the circuit diagram.

“Go to” a circuit connection

The easy800 circuit diagram editor provides a Go To function in order to enable fast access to a circuit connection.

- ▶ Press **ESC** and use the ^v cursor buttons to select the GO TO menu.
- ▶ Press **OK**.
- ▶ Select the required rung (L... ...) with the ^v cursor buttons.

```

I 01----I 02---
Q 01----HYD1Q1J
L: 1 C:1 B:1140

```

The first contact on the circuit connection is always indicated.

- ▶ Press **OK**.

```

I 01----I 02---
Q 01----HYD1Q1J
L: 1 C:1 B:1140

```

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

Deleting the rung

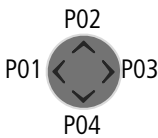
easy800 only removes empty circuit connections (without contacts or coils).

- ▶ Delete all the contacts and coils from the circuit connection.
- ▶ Position the cursor on the first contact field of the empty circuit connection.
- ▶ Press **DEL**.

The subsequent circuit connection(s) will be “pulled up” and any existing links between circuit connections will be retained.

Switching via the cursor buttons

With easy800, you can also use the four cursor buttons as hard-wired inputs in the circuit diagram.



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

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

Example 1

A lamp at output Q1 is switched on and off via inputs I1 and I2 or by using cursor buttons ^ or v.

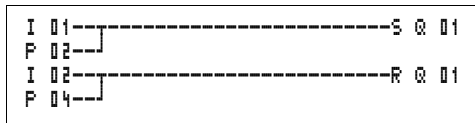


Figure 54: Switch Q1 via I1, I2, ^, or v

Example 2

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

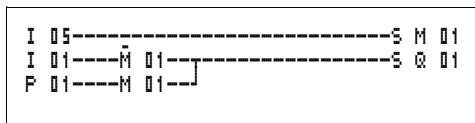
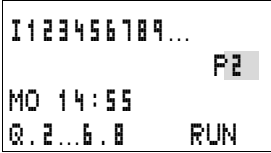


Figure 55: I5 switches over to the cursor buttons.



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

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



Displayed on the Status display:

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

Checking the circuit diagram

easy800 contains a built-in measuring device enabling you to monitor the switching states of contacts, relays and function block coils during operation.

► Create the small parallel circuit below and save it.

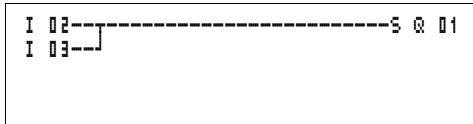


Figure 56: Parallel circuit

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

You are now unable to edit the circuit diagram.



If you switch to the circuit diagram display and are unable to modify a circuit diagram, first check whether easy800 is in STOP mode.

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

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

► Switch on I3.

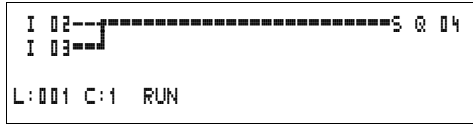


Figure 57: Power flow display

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

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

The bottom right of the power flow display indicates that the controller is in RUN mode. (→ Section "Power flow display with Zoom function" Page 75).



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

Function block editor

The easy800 has the FUNCTION RELAYS menu in order to edit the function blocks without circuit diagrams. The function blocks are an inherent component of the program.

Calling the function blocks via the FUNCTION RELAYS menu

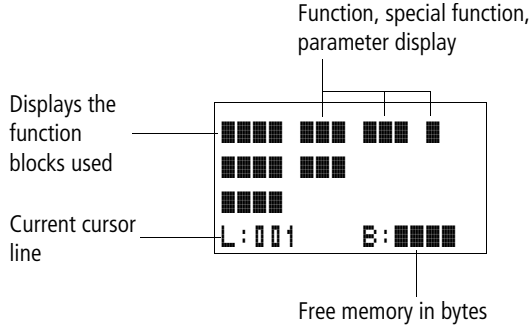


Figure 58: Explanation of the function block display

Display of the function blocks for editing

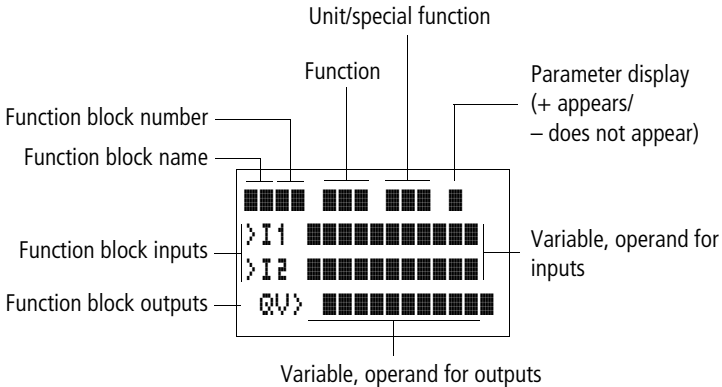
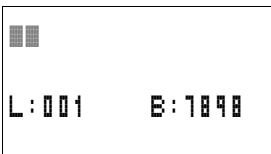


Figure 59: Function block display during editing

Editing function blocks

- ▶ Go to the FUNCTION RELAYS menu.
- ▶ Press the **OK** button.



The following display appears if no function blocks are present.

The cursor flashes.

- ▶ Press the **OK** button.

```
AR01
L:001      B:7900
```

The editor for inputting a function block is displayed.

Select the desired function block and number with the ^\< > cursor buttons.

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

```
AR01 ADD      +
CP10          +
T 10 ?X      -
L:001      B:6400
```

The following display appears if function blocks are available.

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

Calling up function blocks from the circuit diagram

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

Example: timing relay function block

```
T 01 X? M:S +
>I1 20:30
>I2
QV>MD96
L:001      B:7800
```

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

Assigning operands to an > input of a function block



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

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

Assigning operands to a QV> output of a function block



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

Deleting operands on the function block inputs/ outputs

Position the cursor on the required operand.

▶ Press the **DEL** button.

```
T 01 X? M:S +
>I1 ■■■:30
>I2
QV>MD96
```

The operand is deleted.

```
T 01 X? M:S +
>I1 ■■
>I2
QV>MD96
L:001 B:1808
```

Deleting an entire function block

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

▶ Select the required function block from the list.

```
AR01 ADD +
CP10 +
T 18 ?X -
L:002 B:1808
```

In this case CP10.

▶ Press the **DEL** button.


```
AR01 ADD      +
T 10 ?X      -
L:001
```

The function block is erased.

Checking function blocks

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

Checking from the circuit diagram:

Position the cursor on a contact or a coil of the required function block. Press **OK**.

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

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

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

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

Checking the function block via the function block editor:

You access the function block list via the FUNCTION RELAYS menu.

Select the required function block:

```
AR01 ADD      +
CP10          +
T 10 ?X      -
L:001      RUN
```

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

► Press the **OK** button.

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

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

Displaying the operands when checking the function blocks:

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

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

The operand is displayed.

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

▶ Press the **ALT** button again.

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

The display shows the values.

Coil functions

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

Table 8: Coil function

easy800 display	Coil function	Example
C	Contactor function	C001, C002, C504, C:01, CM01, ..
J	Impulse relay function	J003, JM04, J008, JS01, J:01, ..
S	Set	S008, SM02, SD03, SS04, ..
R	Reset	R004, RM05, RD01, RS03, ..
J	Contactor function with negated result	J006, JM96, ..
F	Cycle pulse with rising edge	FM01, ..
f	Cycle pulse with falling edge	fm42, ..



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

Rules for wiring relay coils

Relay with contactor function



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

The following applies to non-retentive coil functions such as \overline{C} (contactor), \overline{N} (negated contactor), \overline{R} , \overline{F} (rising and falling edge detection): Each coil must only be used once. The last coil in the circuit diagram determines the status of the relay.

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

Coil with contactor function \overline{C}

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

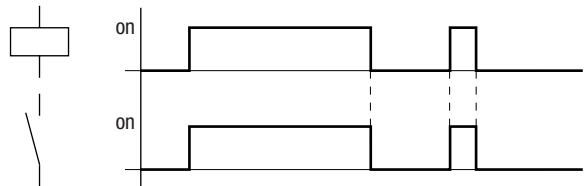


Figure 60: Signal diagram of contactor function

Impulse relay \overline{J}

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

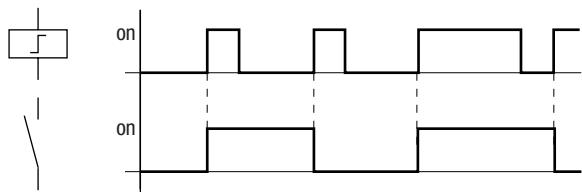


Figure 61: Signal diagram of impulse relay

A coil is automatically switched off if the power fails and if easy is in STOP mode. Exception: Retentive coils retain signal 1 (see → Section "Retention", Page 283).

"Set" S and "Reset" R coil function

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

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

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

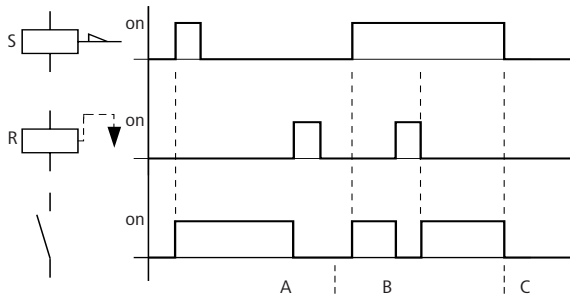


Figure 62: Signal diagram of "Set" and "reset"

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

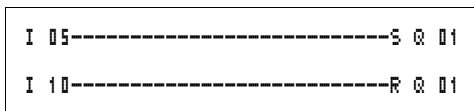


Figure 63: Simultaneous triggering of Q 01

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

Coil negation (inverse contactor function) \bar{I}

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

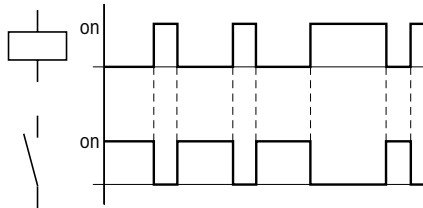


Figure 64: Signal diagram of inverse contactor function

Rising edge evaluation (cycle pulse) I^{\uparrow}

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

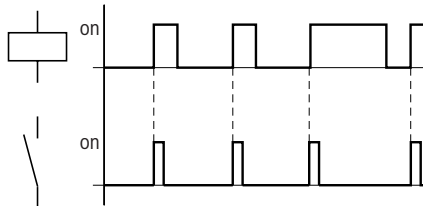


Figure 65: Signal diagram of cycle pulse with rising edge

Falling edge evaluation (cycle pulse) I^{\downarrow}

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

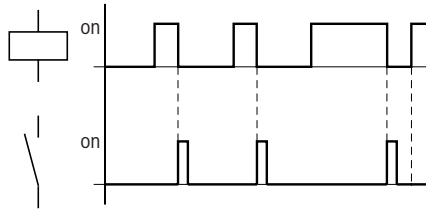


Figure 66: Signal diagram of cycle pulse with falling edge



A set coil is automatically switched off if the power fails and if easy is in STOP mode. Exception: Retentive coils retain signal 1 (see → Section "Retention", Page 283).

Function blocks

The function blocks are used to simulate some of the devices used in conventional open-loop and closed-loop control systems. easy800 provides the following function blocks:

- Analog value comparator/threshold controller (only with easy800 V DC variants)
- Arithmetic,
 - addition, subtraction, multiplication, division
- Compare data blocks
- Transfer data blocks
- Boolean sequence
- Counters,
 - up and down counters with upper and lower threshold values, preset
 - frequency counters,
 - high-speed counters,
 - Incremental encoder counter
- Comparators
- Text, output freely editable texts, enter values
- Data function block
- PID controllers
- Smoothing filters
- Value scaling
- Pulse width modulator
- Read (GET) data from the easy-NET
- Time switches,
 - weekday/time
 - year, month, day (date),
- Numerical converters
- Master reset
- Operating hours counter
- Write (PUT) data to the easy-NET
- Synchronisation of date and time via the easy-NET

- Timing relays
 - on-delayed,
 - on-delayed with random switching,
 - Off-delayed, also retriggerable
 - Off-delayed with random switching, also retriggerable
 - on and off delayed,
 - on and off delayed with random switching,
 - single pulse,
 - synchronous flashing,
 - asynchronous flashing,
- Set cycle time
- Value limitation

The following applies to function blocks:



The most recent actual values are cleared if the power supply is switched off or if easy800 is switched to STOP mode. Exception: Retentive data keeps its state (→ Section "Retention", Page 283).

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



Attention!

The following applies to RUN mode: easy800 processes the function block after a pass through the circuit diagram. The last state of the coils is used for this.



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



Attention!

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

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

Analog value comparator/threshold value switch

easy800 provides 32 analog value comparators from A 01 to A 32.

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

All easy800 DC variants have analog inputs.

The following comparisons are possible:

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

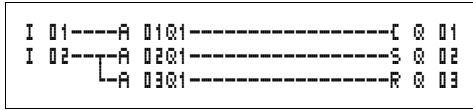


Figure 67: easy800 circuit diagram with analog value comparators

```

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

Parameter display and parameter set for analog value comparators:

A 02	Analog value comparator function block number 02
GT	Greater than mode
+	Appears in the parameter display
>I1	Comparison value 1
>F1	Amplification factor for >I1 ($>I1 = >F1 \times \text{value}$)
>I2	Comparison value 2
>F2	Amplification factor for >I2 ($>I2 = >F2 \times \text{value}$)
>OS	Offset for the value of >I1
>HY	Switching hysteresis for value >I2 (Value HY applies to positive and negative hysteresis.)

Inputs

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

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

Analog value comparator operating modes

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

Contacts

A 01Q1 to A 32Q1

Memory requirement of the analog value comparator

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

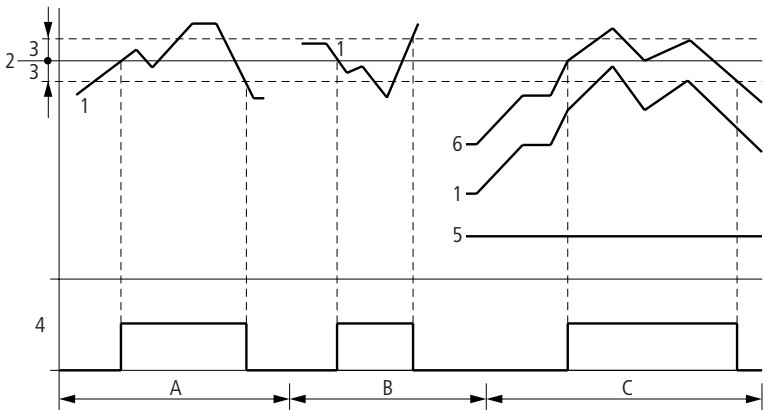


Figure 68: Signal diagram of analog value comparator

- 1: actual value on >I1
- 2: setpoint value on >I2
- 3: hysteresis on >HV
- 4: switching contact (make contact)
- 5: offset for value >I1

6: actual value plus offset

- Range A: Compare $\text{I1} > \text{I2}$
 - 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 $\text{I1} < \text{I2}$
 - The actual value drops.
 - The contact switches if the actual reaches the setpoint value.
 - The actual value changes and rises above the value of the setpoint value plus hysteresis.
 - The contact goes to the normal position.
- Range C: Compare $\text{I1} > \text{I2}$ with offset
 - This example behaves as described in Range A. The offset value is added to the actual value.
- Comparison $\text{I1} = \text{I2}$
 - The contact switches on:
 - If the setpoint is exceeded with the actual value rising.
 - If the setpoint is undershot with the actual value decreasing.
 - The contact switches off:
 - If the hysteresis limit is exceeded with the actual value rising.
 - If the hysteresis limit is undershot with the actual value decreasing.

Arithmetic function block

easy800 provides 32 arithmetic function blocks AR01 to AR32.

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

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

Inputs

The function block inputs $\rangle I1$ and $\rangle I2$ can have the following operands:

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

Actual value ...QV>

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

- Markers MD, MW, MB
- Analog output QA01

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

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

Parameter display and parameter set for arithmetic function blocks:

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

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

Arithmetic function block modes

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

Value range

The function block operates in the integer range from $-2\,147\,483\,648$ to $+2\,147\,483\,647$.

Behaviour when value range is exceeded

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

Displaying the parameter set in the PARAMETERS menu

- + Access enabled
- – Access disabled

Contacts

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

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

Coils

The arithmetic function block does not have any coils.

Memory requirement of the arithmetic function block

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

Addition

$$42 + 1000 = 1042$$

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

$$\text{AR..CY} = \text{Status 1}$$

$$-2048 + 1000 = -1048$$

Subtraction

$$1134 - 42 = 1092$$

-2 147 483 648 - 3 = last valid value of this arithmetic operation, due to overflow (CARRY)

$$\text{AR..CY} = \text{Status 1}$$

$$-4096 - 1000 = -5096$$

$$-4096 - (-1000) = -3096$$

Multiplication

$$12 \times 12 = 144$$

1 000 042 \times 2 401 = last valid value of this arithmetic operation, due to overflow (CARRY)

$$\text{Correct value} = 2\,401\,100\,842$$

$$\text{AR..CY} = \text{Status 1}$$

$$-1000 \times 10 = -10000$$

Division

$$1024 : 256 = 4$$

1024 : 35 = 29 (The digits after the decimal point are omitted.)

1024 : 0 = last valid value of this arithmetic operation, due to overflow (CARRY)

(mathematically correct: "Infinite")

$$\text{AR..CY} = \text{Status 1}$$

$$-1000 : 10 = -100$$

$$1000 : -10 = -100$$

$$-1000 : (-10) = 100$$

$$10 : 100 = 0$$

Data block comparator

This function block is available from device version 04.

easy800 provides 32 function blocks BC01 to BC32 for comparing values of two consistent marker ranges. The comparison is in byte format. The following marker types can be compared:

- MB,
- MW,
- MD.

The function block is enabled in the circuit diagram.

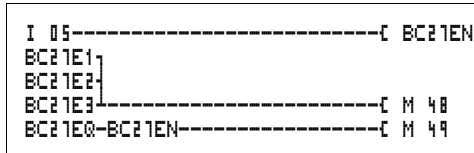
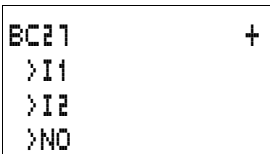


Figure 69: easy800 circuit diagram with enabling of data block comparator function block



Parameter display and parameter set for a data block comparator:

BC27	Data block comparator function block number 27
+	Appears in the parameter display
>I1	Start of comparison range 1
>I2	Start of comparison range 2
>N0	Number of elements to be compared in bytes per range. Value range 1 to + 383

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

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

Inputs

The function block inputs $\gt I1$, $\gt I2$ and $\gt NO$ can have the following operands:

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

Specifying the marker range without offset

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

Specifying the marker range with offset

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

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

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

Displaying the parameter set in the PARAMETERS menu

- + Access enabled
- -: Access disabled

Contacts

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

BC01E2 to BC32E2: The number of comparison elements exceeds one of the comparison ranges.

BC01E3 to BC32E3: the number of comparison elements exceeds one of the comparison ranges.

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

Coils

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

Memory requirement of the data block comparator function block

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

Function of the data block comparator function block

The data block comparator function block compares two consistent data blocks.

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



No data blocks are compared if an error is present.

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

Example:

Comparison of marker blocks, definition of marker ranges direct

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

Parameters of BC01 function block:

Comparison range 1: >I1 ME10

Comparison range 2: >I2 ME40

Number of bytes: >NO 10

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

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

Example:

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

Two marker blocks are to be compared. Block 1 starts at MB105, Block 2 at MB65. Each block is 4 bytes long.

Parameters of BC01 function block:

Comparison range 1: >I1 MB15

Comparison range 2: >I2 64

Number of bytes: >NO 4

Marker MB01: 1



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

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

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

MB18 are MB68 not identical.

Example:

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

Two marker blocks are to be compared. Block 1 starts at MB60, Block 2 at MB80. Each block is 6 bytes long.

Parameters of BC01 function block:

Comparison range 1: >I1 MB60

Comparison range 2: >I2 MD80

Number of bytes: >NO 6



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

Comparison range 1	Value of marker range 1 (decimal/binary)	Comparison range 2	Value of marker range 2 (decimal/binary)
MB60	45/ 00101101	MD80 (Byte 1, LSB)	1097219629/ 010000010110011000111110 00101101
MB61	62/ 00111110	MD80 (Byte 2)	1097219629/ 0100000101100110 00111110 00101101
MB62	102/ 01100110	MD80 (Byte 3)	1097219629/ 01000001 01100110 0011111000101101
MB63	65/ 01000001	MD80 (Byte 4, MSB)	1097219629/ 01000001 011001100011111000101101
MB64	173/ 10101101	MD81 (Byte 1, LSB)	15277/ 00111011 10101101
MB65	59/ 00111011	MD81 (Byte 2)	15277/ 0001000 10101101

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

MB65 and MD81 (Byte 2) are not identical.

Example:

Comparison of marker blocks, range violation error.

Two marker blocks are to be compared. Block 1 starts at MB60, Block 2 at MB90. Each block is 30 bytes long.

Parameters of BC01 function block:

Comparison range 1: >I1 MB60

Comparison range 2: >I2 MD90

Number of bytes: >NO 30



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

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

BC01E1 is 1.

Example

Comparison of marker blocks, range overlap error.

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

Parameters of BC01 function block:

Comparison range 1: >I1 MW60

Comparison range 2: >I2 MW64

Number of bytes: >NO 12



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

The error message "Comparison ranges overlap" is output.

BC01E2 is 1.

Example:

Comparison of marker blocks, invalid offset error.

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

Parameters of BC01 function block:

Comparison range 1: >I1 MW40

Comparison range 2: >I2 MW54

Number of bytes: >NO C 01QV



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

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

BC01E3 is 1.

Data block transfer

This function block is available from device version 04.

easy800 is provided with 32 function blocks BT01 to BT32 for transferring values from one marker range (Copy data). The marker ranges can be overwritten with a particular value (data initialisation). The following marker types can be transferred and overwritten:

- MB,
- MW,
- MD.

The function block is enabled in the circuit diagram.

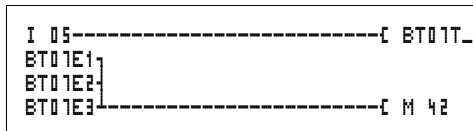


Figure 70: easy800 circuit diagram with enabling of transfer data block function block

```
BT01 INI      +
  >I1
  >I2
  >NO
```

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

BT01	Data block transfer function block number 07
INI	INI mode, initialise marker ranges
+	Appears in the parameter display
>I1	Source range start
>I2	Destination range start
>NO	Number of elements to be written in bytes per range. Value range 1 to + 383

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

Operating modes of the transfer data block function block

Parameter	Function
INI	Initialise marker ranges
CPV	Copy marker ranges

Inputs

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

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

Specifying the marker range without offset

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

Specifying a marker range with offset

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

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

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

Displaying the parameter set in the PARAMETERS menu

- + Access enabled
- -: Access disabled

Contacts

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

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

BT01E3 to BT32E3: the specified offset is invalid.

Coils

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

Memory requirement of the transfer data block function block

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

Function of the transfer data block function block

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



No data blocks are initialised if an error occurs.

Initialising INI marker ranges

There is one source range and one destination range. The source range is specified at >I1 . The length of the source range is one byte. The destination range is specified at >I2 . The length of the destination range is specified by the number of bytes at the >N0 input.

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

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

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

Example:

Initialising marker blocks, specifying marker ranges directly

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

Parameters of BT01 function block:

Source range: >I1 MB10

Destination range: >I2 MB20

Number of bytes: >NO 10

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

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

Example:

Initialisation of marker blocks, definition of a range with offset

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

Parameters of BT01 function block:

Source range: `>I1 MB15`

Destination range: `>I2 64`

Number of bytes: `>NO 4`

Marker MB01: `1`



Destination range: Constant 64:
Marker MB01 plus Offset: $1 + 64 = 65 \rightarrow$ MB65.

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

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

Example:

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

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

Parameters of BT01 function block:

Source range: `>I1 MB60`

Destination range: `>I2 MD80`

Number of bytes: `>NO 8`



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

Comparison range 1	Value of marker range 1 (decimal/binary)	Comparison range 2	Value of marker range 2 (decimal/binary)
MB60	45/ 00101101	MD80 (Byte 1, LSB)	757935405/ 0010110100101101 00101101 00101101
		MD80 (Byte 2)	757935405/ 0010110100101101 00101101 00101101
		MD80 (Byte 3)	757935405/ 0010110100101101 00101101 00101101
		MD80 (Byte 4, MSB)	757935405/ 00101101 001011010010110100101101
		MD81 (Byte 1, LSB)	757935405/ 0010110100101101 00101101 00101101
		MD81 (Byte 2)	757935405/ 0010110100101101 00101101 00101101
		MD81 (Byte 3)	757935405/ 00101100 01011011 0010110100101101
		MD81 (Byte 4, MSB)	757935405/ 00101101 001011010010110100101101

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

Example:

Transfer of marker byte, range violation error.

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

Parameters of BT01 function block:

Source range: >I1 MD96

Destination range: >I2 MD93

Number of bytes: >NO 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:

Transfer of marker bytes, invalid offset error.

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

Parameters of BC01 function block:

Comparison range 1: >I1 MB40

Comparison range 2: >I2 MW54

Number of bytes: >NO C 01QV



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

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

BT01E3 is 1.

CPY mode, copy marker ranges

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

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

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

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

Example:

Copy of marker blocks, definition of marker ranges direct

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

Parameters of BT01 function block:

Source range: >I1 MB10

Destination range:>I2 MB20

Number of bytes: >NO 10

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

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

Example:

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

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

Parameters of BT01 function block:

Source range: >I1 MB15

Destination range: >I2 64

Number of bytes: >NO 4

Marker MB01: 1



Destination range: Constant 64:
Marker MB01 plus Offset: $1 + 64 = 65 \rightarrow$ MB65.

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

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

Example:

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

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

Parameters of BT01 function block:

Source range: >I1 MD60

Destination range: >I2 MW40

Number of bytes: >NO 12



The transfer is in byte format. 12 bytes are to be copied. The range MD60 to MD62 is 12 bytes. This is copied to the range MW40 to MW45.

Comparison range 1	Value of marker range 1 (decimal/binary)	Comparison range 2	Value of marker range 2 (decimal/binary)
MD60	866143319/ 0011001110100000 0100110001010111	MW40 (LSW)	19543/ 0011001110100000 0100110001010111
MD60	866143319/ 0011001110100000 0100110001010111	MW41 (MSW)	13216/ 0011001110100000 0100110001010111
MD61	173304101/ 0000101001010100 0110100100100101	MW42 (LSW)	26917/ 0000101001010100 0110100100100101
MD61	173304101/ 0000101001010100 0110100100100101	MB43 (MSW)	2644/ 0000101001010100 0110100100100101
MD62	982644150/ 0011101010010001 1111010110110110	MB44 (LSW)	62902/ 0011101010010001 1111010110110110
MD62	982644150/ 0011101010010001 1111010110110110	MB45 (MSW)	14993/ 0011101010010001 1111010110110110

After a rising edge from 0 to 1 at coil BT01T_ the values are copied to the appropriate range.

Example:

Copying of marker bytes, destination range violation error.

The value of marker bytes MB81 to MB96 is to be transferred to MD93, MD94, MD95 and MD96. The length is 16 bytes.

Parameters of BT01 function block:

Source range: >I1 MB81
 Destination range: >I2 MD93
 Number of bytes: >NO 16



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

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

BT01E1 is 1.

Example

Comparison of marker blocks, range overlap error.

12 bytes are to be copied starting from MW60. MW64 is specified as destination address.

Parameters of BT01 function block:

Comparison range 1: >I1 MW60

Comparison range 2: >I2 MW64

Number of bytes: >NO 12



The copy operation is in byte format. MW60 to MW64 is 8 bytes. The number of bytes is 12 bytes.

The error message "Both ranges overlap" is output.

BC01E2 is 1.

Example:

Copying of marker bytes, invalid offset error.

The value of marker word MW40 is to be copied to MW54 and subsequent marker words. The block length is specified by the value of the counter C 01QV.

Parameters of BT01 function block:

Comparison range 1: >I1 MW40

Comparison range 2: >I2 MW54

Number of bytes: >NO C 01QV



The value of C 01QV is 10042. This value is too big. The value at >NO can be between 1 and +383.

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

BT01E3 is 1.

Boolean operation

easy800 provides 32 function blocks from BV01 to BV32 for Boolean operations with values.

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

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

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

```

BV27 AND      +
  >I1
  >I2
  QV>
  
```

Parameter display and parameter set for Boolean sequence function block:

BV27	Boolean sequence function block number 27
AND	AND operation mode
+	Appears in the parameter display
>I1	First value
>I2	Second value
QV>	Result of the sequence

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

Operating modes of the Boolean sequence function block

Parameter	Function
AND	AND operation
OR	OR operation
XOR	Exclusive OR operation
NOT	Negation of the Boolean value of >I1

Value range

32 bit signed value

Inputs

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

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

Actual value ...QV>

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

- Markers MD, MW, MB
- Analog output QA01

Displaying the parameter set in the PARAMETERS menu

- + Access enabled
- -: Access disabled

Contacts

BV01ZE to BV32ZE: ZERO zero bit, value on output function block is equal to zero

Coils

The Boolean sequence function block does not have coils.

Memory requirement Boolean sequence function block

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

NOT Boolean sequence

Value $\>I1$: $13219_{dec} =$
 $00000000000000000011001110100011_{bin}$

Value $\>I2$: Omitted

Result QV $\>$: $-13220_{dec} =$
 $11111111111111111100110001011100_{bin}$

The NOT sequence operates according to the following rules:

$\>I1$, positive value

Negate value of $\>I1$ and subtract 1:

$$-\>I1| - 1 = \>I2$$

$\>I1$, Negative value

Value of $\>I1$ and subtract 1:

$$|\>I1| - 1 = \>I2$$

Counters

easy800 provides 32 up/down counters from C 01 to C 32. The counter relays allow you to count events. You can enter upper and lower threshold values as comparison values. The contacts will switch according to the actual value. To specify a start value, for example, counting from the value 1200, this can be implemented using a "C.." counter.

The "C.." counters are cycle time dependent.

Wiring of a counter

You integrate a counter into your circuit in the form of a contact and coil. The counter relay has different coils and contacts.



To prevent unpredictable switching states, use each coil of a relay once only in the circuit diagram.



Figure 71: easy800 circuit diagram with counter relay

```

C 20      +
>SH
>SL
>SV
QV>
    
```

Parameter display and parameter set for counter relay:

C 20	Counter relay function block number 20
+	Appears in the parameter display
>SH	Upper setpoint
>SL	Lower setpoint
>SV	Defined actual value (Preset)
QV>	Actual value in RUN mode

In the parameter display of a counter relay you change setpoint values and/or the preset value and enabling of the parameter display.

Value range

The function block operates in the integer range from -2147483648 to 2147483647.

Behaviour when value range is exceeded

The function block sets the switching contact C .. CY to the status 1 and retains the value of the last valid operation.



Counter C counts every rising edge on the counter input. If the value range is exceeded, the switching contact C ..CY switches to status 1 for one cycle per rising edge detected.

Inputs

The function block inputs $\>SH$, $\>SL$ and $\>SV$ can have the following operands:

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

Actual value ...QV>

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

- Markers MD, MW, MB
- Analog output QA01

Displaying the parameter set in the PARAMETERS menu

- + Access enabled
- – Access disabled

Contacts

- C 010F to C 320F: Actual value \geq upper setpoint
- C 01FB to C 32FB: Actual value \leq Lower setpoint value
- C 01ZE to C 32ZE: Actual value = zero
- C 01CY to C 32CY: Value range exceeded

Coils

- C 01C_ to C 32C_: counter coils, count with rising edge
- C 01D_ to C 32D_: count direction definition, status 0 = count upwards, status 1 = count downwards
- C 01RE to C 32RE: Reset actual value to zero
- C 01SE to C 32SE: accept defined actual value with rising edge.

Memory requirement of the counter relay

The counter relay function block requires 52 bytes of memory plus 4 bytes per constant on the function block inputs.

Retention

Counter relays can be operated with retentive actual values. The number of retentive counter relays can be selected in the SYSTEM → RETENTION menu.

The retentive actual value requires 4 bytes of memory.

If a counter relay is retentive, the actual value is retained when the operating mode changes from RUN to STOP as well as when the power supply is switched off.

If easy is started in RUN mode, the counter relay operates with the retentively saved actual value.

Function of the counter function block

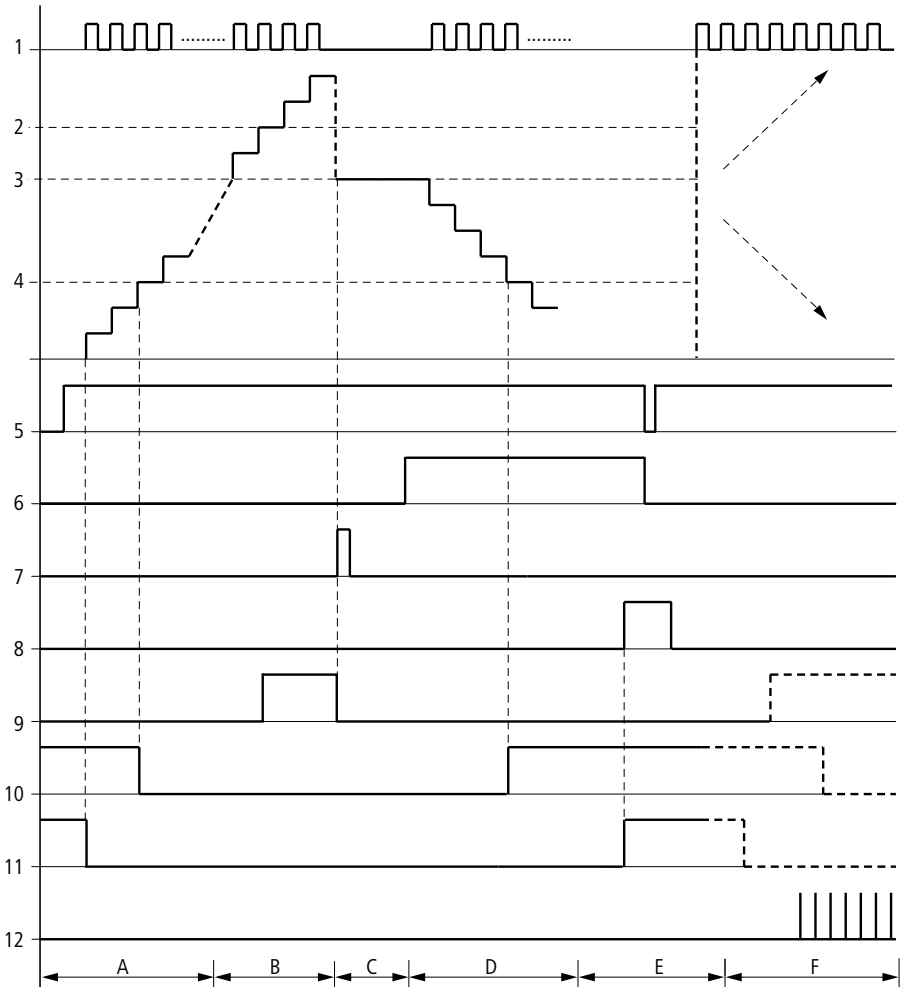


Figure 72: Signal diagram of counter

- 1: counter coil C..C_
- 2: upper setpoint value >SH
- 3: preset actual value >SV
- 4: lower setpoint value >SL
- 5: counting direction, coil C..D_
- 6: accept preset actual value, coil C..SE

7: reset coil C..RE

8: contact (make contact) C..OF upper setpoint value reached, exceeded

9: contact (make contact) C..FB lower setpoint value reached, undershot

10: actual value equal to zero

11: out of value range

- Range A:

- The counter has the value zero.
- The contacts C..ZE (actual value equal to zero) and C..FB (lower setpoint value undershot) are active.
- The counter receives counter values and increases the actual value.
- C..ZE drops out as well as C..FB and also when the lower setpoint value is reached.

- Range B:

- The counter counts upwards and reaches the upper setpoint value. The contact "upper setpoint value reached" C..OF becomes active.

- Range C:

- The coil C..SE is briefly actuated and the actual value is set to the preset actual value. The contacts go to the respective position.

- Range D:

- The counting direction coil C..D_ is actuated. If counting impulses are present, downward count is initiated.
- If the lower setpoint value is undershot, the contact C..FB becomes active.

- Range E:

- The reset coil C..RE is activated. The actual value is set to zero.
- The contact C..ZE is active.

- Range F:

- The actual value goes outside the value range of the counter.
- The contacts become active according to the direction of the values (positive or negative).

High-speed counters

easy800 provides various high-speed counter functions. These counter function blocks are coupled directly to the digital inputs. The high-speed counter functions are only available with EASY8..-DC...

The following functions are possible:

- Frequency counters, measure frequencies **CF..**
- High-speed counters, count high-speed signals **CH..**
- Incremental encoder counters, count two-channel incremental encoder signals **CI..**

The high-speed digital inputs are I1 to I4.

The following wiring rules apply:

- I1: CF01 or CH01 or CI01
- I2: CF02 or CH02 or CI01
- I3: CF03 or CH03 or CI02
- I4: CF04 or CH04 or CI02



Attention!

Every digital input I .. may only be used once by the CF, CH, CI function blocks.

The incremental encoder occupies an input pair.

Example:

- I1: high-speed counter CH01
- I2: frequency counter CF02
- I3: incremental encoder channel A CI02
- I4: incremental encoder channel B CI02

Example: function block list in the FUNCTION RELAYS menu:

```
CI01
CF01
CH01
```

All function blocks access digital input I1.

Only CH01 supplies the correct value.

Frequency counter

easy800 provides four frequency counters which are CF01 to CF04. The frequency counters enable the measurement of frequencies. You can enter upper and lower threshold values as comparison values. The high-speed frequency counters are hardwired to the digital inputs I1 to I4.

The CF.. frequency counters operate independently of the cycle time.

Counter frequency and pulse shape

The maximum counter frequency is 5 kHz.

The minimum counter frequency is 4 Hz.

The signals must be square waves. The mark-to-space ratio is 1:1.

Measurement method

The pulses on the input are counted for one second irrespective of the cycle time, and the frequency is determined. The result of the measurement is made available as a value to the function block output CF..QV.

Wiring of a counter

The following assignment of the digital inputs apply.

- I1 counter input for the counter CF01
- I2 counter input for the counter CF02
- I3 counter input for the counter CF03
- I4 counter input for the counter CF04



To prevent unpredictable switching states, use each coil of a relay once only in the circuit diagram. Use a counter input for the CF, CH, CI counters only once.

Wiring of a frequency counter

You integrate a frequency counter into your circuit in the form of a contact and coil. The counter relay has different coils and contacts.

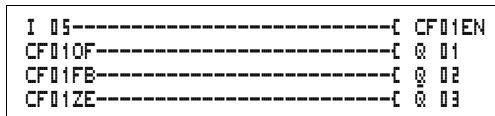


Figure 73: easy800 circuit diagram with frequency counter

```

CF01      -
>SH
>SL
@V>
  
```

Parameter display and parameter set for frequency counter:

CF01	Frequency counter function block number 01
-	Does not appear in the parameter display
>SH	Upper setpoint
>SL	Lower setpoint
@V>	Actual value in RUN mode

In the parameter display of a counter relay you change setpoint values and/or the preset value and the enable of the parameter display.

Value range

The function block operates in the integer range from 0 to 5000

1 kHz = 1000

Behaviour when value range is exceeded

The value range cannot be exceeded as the maximum measured value is less than the value range.

Inputs

The function block inputs >SH and >SL can have the following operands:

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

Actual value ...QV>

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

- Markers MD, MW, MB
- Analog output QA01

Displaying the parameter set in the PARAMETERS menu

- + Access enabled
- – Access disabled

Contacts

- CF01OF to CF04OF: Actual value \geq Upper setpoint
- CF01FB to CF04FB: Actual value \leq Lower setpoint
- CF01ZE to CF04ZE: Actual value = Zero

Coils

CF01EN to CF04EN: enable of the counter with coil status = 1.

Memory requirement of the frequency counter

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

Retention

The frequency counter does not retain actual values, as the frequency is remeasured continuously.

Function of the frequency counter function block

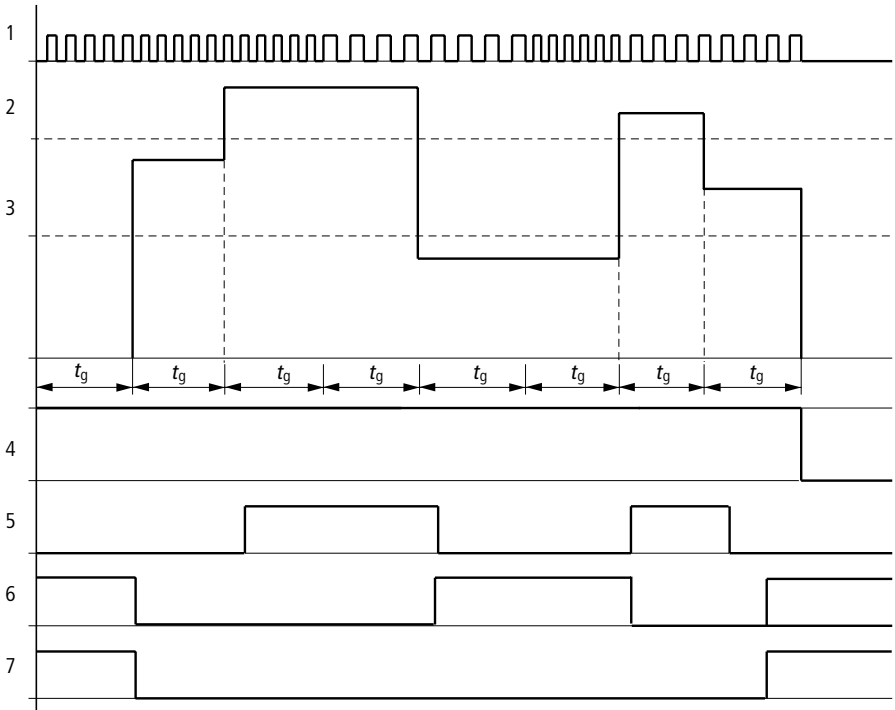


Figure 74: Signal diagram of frequency counter

- 1: counter input I1 to I4
- 2: upper setpoint value $\gg H$
- 3: lower setpoint value $\gg L$
- 4: enable CF..EN
- 5: contact (make contact) CF..OF upper setpoint value exceeded
- 6: contact (make contact) CF..FB lower setpoint value undershot
- 7: actual value equal to zero CF..ZE
- t_g : gate time for the frequency measurement

- The first measurements are made after the CF..EN enable signal has been activated. The value is output after the gate time has timed out.
- The contacts are set in accordance with the measured frequency.
- If the CF..EN enable signal is removed, the output value is set to zero.

High-speed counters

easy800 provides four high-speed up/down counters CH01 to CH04 for use. The high-speed frequency counters are hardwired to the digital inputs I1 to I4. These counter relays allow you to count events independently of the cycle time. You can enter upper and lower threshold values as comparison values. The contacts will switch according to the actual value. To specify a start value, for example, counting from the value 1989, this can be implemented using the "CH .." value.

The CH.. counters operate independently of the cycle time.

Counter frequency and pulse shape

The maximum counter frequency is 5 kHz.

The signals must be square waves. The mark-to-space ratio is 1:1.

Wiring of a counter

The following assignment of the digital inputs apply.

- I1 counter input for the counter CH01
- I2 counter input for the counter CH02
- I3 counter input for the counter CH03
- I4 counter input for the counter CH04



To prevent unpredictable switching states, use each coil of a relay once only in the circuit diagram. Use a counter input for the CF, CH, CI counters only once.

You integrate a counter into your circuit in the form of a contact and coil. The counter relay has different coils and contacts.



Figure 75: easy800 circuit diagram with high-speed counter

```

CH01      +
>SH
>SL
>SV
@V>
    
```

Parameter display and parameter set for high-speed counters:

CH01	High-speed counter function block number 01
+	Appears in the parameter display
>SH	Upper setpoint
>SL	Lower setpoint
>SV	Defined actual value (Preset)
@V>	Actual value in RUN mode

In the parameter display of a counter relay you change setpoint values and/or the preset value and enable of the parameter display.

Value range

The function block operates in the integer range from -2147483648 to 2147483647.

Behaviour when value range is exceeded

- The function block sets the switching contact CH..CY to status 1.
- The function block retains the value of the last valid operation.



Counter CH counts every rising edge on the counter input. If the value range is exceeded, the switching contact CH ..CY switches to status 1 for one cycle per rising edge detected.

Inputs

The function block inputs >SH , >SL and >SV can have the following operands:

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

Actual value ..QV>

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

- Markers MD, MW, MB
- Analog output QA01



The actual value is only cleared in RUN mode with a selective reset signal.

Displaying the parameter set in the PARAMETERS menu

- + Access enabled
- – Access disabled

Contacts

- CH01OF to CH04OF: Actual value \geq Upper setpoint
- CH01FB to CH04FB: Actual value \leq Lower setpoint
- CH01ZE to CH04ZE: Actual value = Zero
- CH01CY to CH04CY: Value range exceeded

Coils

- CH01EN to CH04EN: enable of the counter
- CH01D to CH04D: count direction definition, state "0" = count upwards, state "1" = count downwards
- CH01RE to CH04RE: reset actual value to zero
- CH01SE to CH04SE: accept preset actual value with rising edge.

Memory requirement of the high-speed counter

The high-speed counter function block requires 52 bytes of memory plus 4 bytes per constant on the function block inputs.

Retention

High-speed counter relays can be operated with retentive actual values. The number of retentive counter relays can be selected in the SYSTEM → RETENTION menu.

If a counter relay is retentive, the actual value is retained when the operating mode changes from RUN to STOP as well as when the power supply is switched off.

If easy is started in RUN mode, the counter relay operates with the retentively saved actual value.

Function of the high-speed counter function block

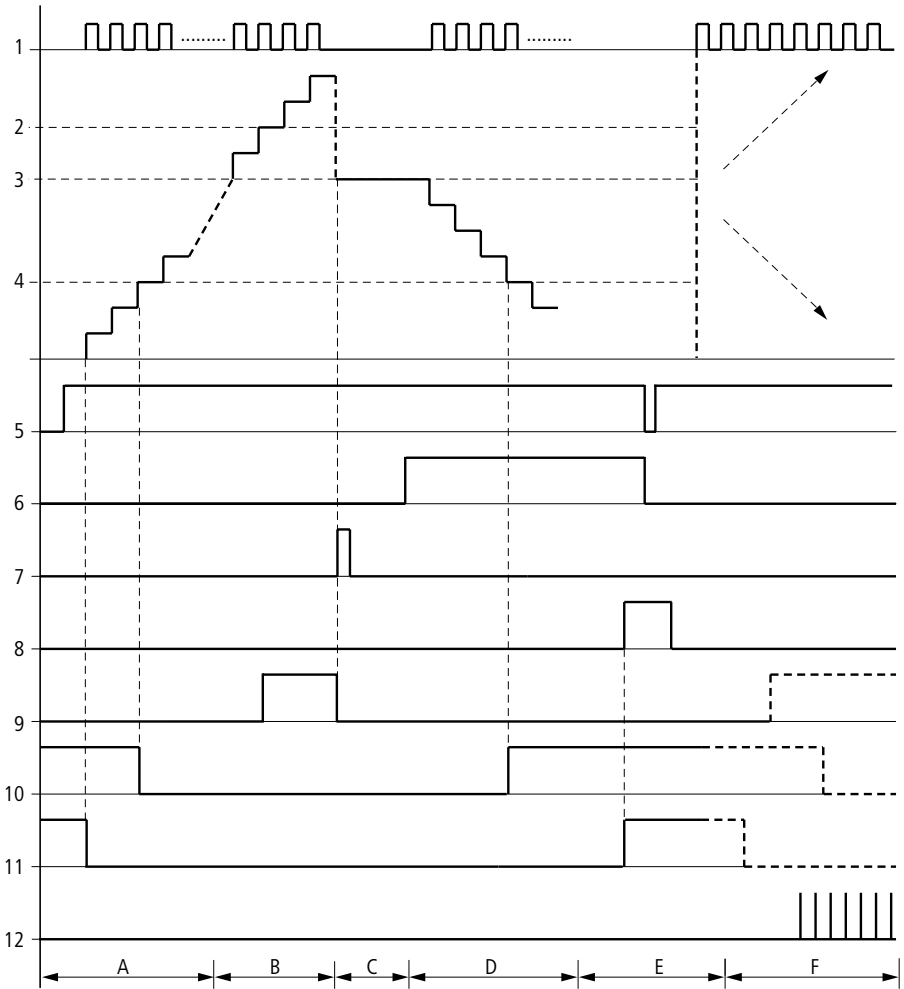


Figure 76: Signal diagram of high-speed counter

- 1: counter input I1 to I4
- 2: upper setpoint value $\setminus S\setminus H$
- 3: preset actual value $\setminus S\setminus V$
- 4: lower setpoint value $\setminus S\setminus L$
- 5: enable of the counter CH..EN
- 6: counting direction, coil CH..D

7: accept preset actual value, coil CH..SE

8: reset coil CH..RE

9: contact (make contact) CH..OF upper setpoint value reached, exceeded

10: contact (make contact) CH..FB lower setpoint value reached, undershot

11: contact (make contact) CH..ZE actual value equal to zero

12: out of value range

- Range A:
 - The counter has the value zero.
 - The contacts CH..ZE (actual value equal to zero) and CH..FB (lower setpoint value undershot) are active.
 - The counter receives counter values and increases the actual value.
 - CH..ZE drops out as well as CH..FB after the lower setpoint value is reached.
- Range B:
 - The counter counts upwards and reaches the upper setpoint value. The contact "upper setpoint value" CH..OF becomes active.
- Range C:
 - The coil CH..SE is briefly actuated and the actual value is set to the preset actual value. The contacts go to the respective position.
- Range D:
 - The counting direction coil CH..D is actuated. If counting impulses are present, downward count is initiated.
 - If the lower setpoint value is undershot, the contact CH..FB becomes active.
- Range E:
 - The reset coil CH..RE is activated. The actual value is set to zero.
 - The contact CH..ZE is active.
- Range F:
 - The actual value goes outside the value range of the counter.
 - The contacts become active according to the direction of the values (positive or negative).

High-speed incremental encoder counters

easy800 provides two high-speed incremental encoder counters CI01 and CI02. The high-speed counter inputs are hardwired to the digital inputs I1, I2, I3 and I4. These counter relays allow you to count events independently of the cycle time. You can enter upper and lower threshold values as comparison values. The contacts will switch according to the actual value. You can use a CI.. counter if you wish to define a start value.

The CI.. counters operate independently of the cycle time.

Counter frequency and pulse shape

The maximum counter frequency is 3 kHz.

The signals must be square waves. The mark-to-space ratio is 1:1. The signals on channels A and B must lead or lag by 90°. Otherwise the counting direction cannot be determined.



Double the number of pulses are counted as a result of the internal method of operation of the incremental encoder. The incremental encoder evaluates the rising and falling edges. This ensures that the pulse count is not affected by oscillation of a signal edge. If the number of pulses are required, divide the value by two.

Wiring of a counter

The following assignment of the digital inputs apply:

- I1 counter input for the counter CI01 channel A
- I2 counter input for the counter CI01 channel B
- I3 counter input for the counter CI02 channel A
- I4 counter input for the counter CI02 channel B



To prevent unpredictable switching states, use each coil of a relay once only in the circuit diagram.

Use a counter input for the CF, CH, CI counters only once.

You integrate a counter into your circuit in the form of a contact and coil. The counter relay has different coils and contacts.



Figure 77: easy800 circuit diagram with high-speed incremental encoder counter

```

CID1      +
  >SH
  >SL
  >SV
QV>
    
```

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

CID1	High-speed incremental encoder counter function block number 01
+	Appears in the parameter display
>SH	Upper setpoint
>SL	Lower setpoint
>SV	Defined actual value (Preset)
QV>	Actual value in RUN mode

In the parameter display of a counter relay you change setpoint values and/or the preset value and enable of the parameter display.

Value range

The function block operates in the integer range from -2147483648 to 2147483647.

Each pulse is counted twice.

Example: value at CI..QV>= 42000

The counter has counted 21 000 pulses.

Behaviour when value range is exceeded

- The function block sets the switching contact CI..CY to status 1.
- The function block retains the value of the last valid operation.



Counter CI counts every rising edge on the counter input. If the value range is exceeded, the switching contact CI ..CY switches to status 1 for one cycle per rising edge detected.

Inputs

The function block inputs >SH, >SL and >SV can have the following operands:

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

Actual value ..QV>

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

- Markers MD, MW, MB
- Analog output QA01



The actual value is only erased in RUN mode with a selective reset signal.

Displaying the parameter set in the PARAMETERS menu

- + Access enabled
- – Access disabled

Contacts

- CI01OF to CI02OF: Actual value \geq Upper setpoint
- CI01FB to CI02FB: Actual value \leq Lower setpoint
- CI01ZE to CI02ZE: Actual value = Zero
- CI01CY to CI02CY: Value range exceeded

Coils

- CI01EN to CI02EN: Counter enable
- CI01RE to CI02RE: Reset actual value to zero
- CI01SE to CI02SE: Accept preset actual value with rising edge.

Memory requirement of the counter relay

The high-speed counter function block requires 52 bytes of memory plus 4 bytes per constant on the function block inputs.

Retention

High-speed counter relays can be operated with retentive actual values. The number of retentive counter relays can be selected in the SYSTEM → RETENTION menu.

If a counter relay is retentive, the actual value is retained when the operating mode changes from RUN to STOP as well as when the power supply is switched off.

If easy is started in RUN mode, the counter relay operates with the retentively saved actual value.

Function of the high-speed incremental encoder counter function block

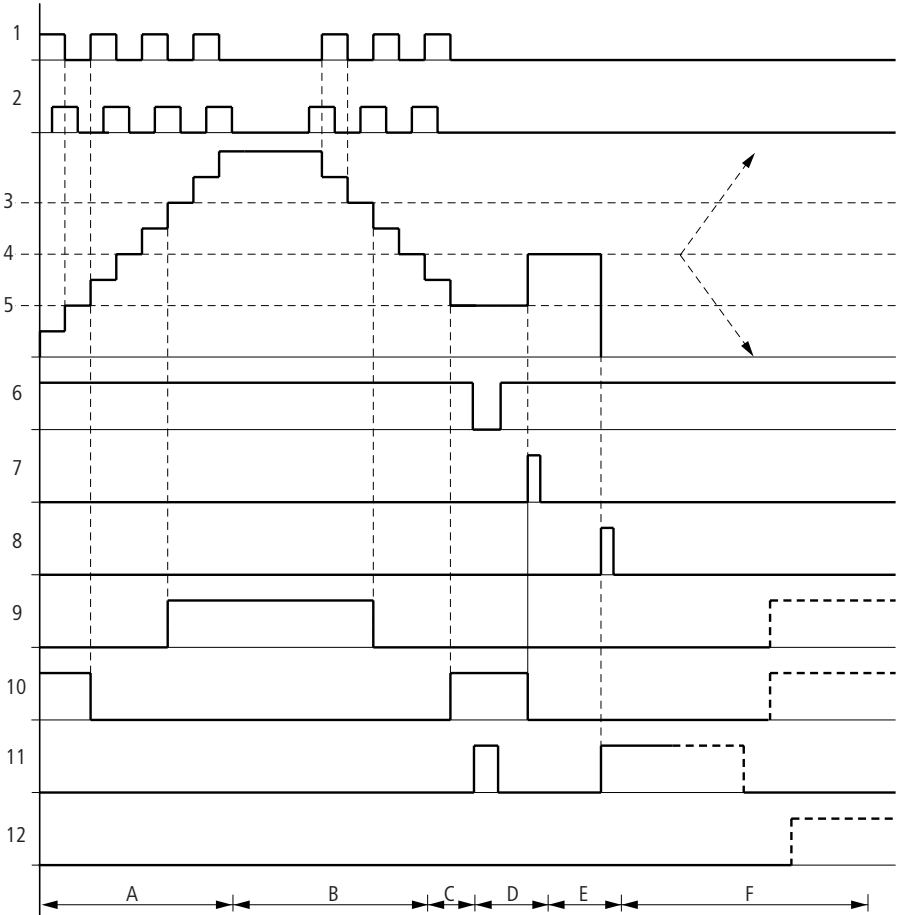


Figure 78: Signal diagram of high-speed incremental encoder counter

- 1: counter input channel A
- 2: counter input channel B
- 3: upper setpoint value $\gg H$
- 4: preset actual value $\gg V$
- 5: lower setpoint value $\gg L$
- 6: counter enable

- 7: accept preset actual value, coil CI..EN
- 8: reset coil CI..RE
- 9: contact (make contact) CI..OF upper setpoint value reached, exceeded
- 10: contact (make contact) CI..FB lower setpoint value reached, undershot
- 11: contact (make contact) CI..ZE actual value equal to zero
- 12: contact (make contact) CI..CY value range exceeded or undershot
- Range A:
 - The counter counts upwards.
 - The value leaves the lower threshold value and reaches the upper value.
- Range B:
 - The count direction changes to a downward count.
 - The contacts switch in accordance with the actual value.
- Range C:
 - The enable signal is set to 0. The actual value becomes 0.
- Range D:
 - The rising edge on the accept preset value coil sets the actual value to the preset value.
- Range E:
 - The reset pulse sets the actual value to zero.
- Range F:
 - The actual value goes outside the value range of the counter.
 - The contacts become active according to the direction of the values (positive or negative).

Comparators

Comparator function blocks allow you to compare constants and variables with one another.

The following comparisons are possible:

Function block input		Function block input
› I1	Greater than	› I2
	Equal to	
	Less than	

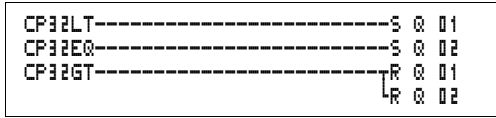
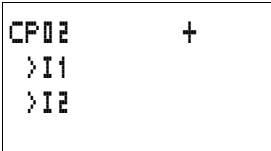


Figure 79: easy800 circuit diagram with comparator



Parameter display and parameter set for the comparator function block:

CP02	Analog value comparator function block number 02
+	Appears in the parameter display
>I1	Comparison value 1
>I2	Comparison value 2

Inputs

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

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

Contacts

- CP01LT to CP32LT, (less than)
 Contact (make contact) switches to status 1, if the value on >I1 is less than the value on >I2; >I1 < >I2.
- CP01EQ to CP32EQ, (equal to)
 Contact (make contact) switches to status 1, if the value on >I1 is equal to the value on >I2; >I1 = >I2.
- CP01GT to CP32GT, (greater than)

Contact (make contact) switches to status 1, if the value at $\text{I}1$ is greater than the value at $\text{I}2$; $\text{I}1 > \text{I}2$.

Memory requirement of the counter relay

The comparator function block requires 32 bytes of memory plus 4 bytes per constant on the function block inputs.

Text output function block

easy800 allows you to display 32 user-defined texts. These texts can be used to display function block actual values, marker values (MB, MW, MD). Function block setpoints, and marker values (MB, MW, MD) that are constants can be entered. The texts can be only edited with EASY-SOFT (-PRO).

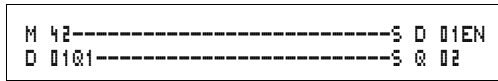


Figure 80: easy800 circuit diagram with a text output function block

Contacts

A contact has been assigned to the text output function block.

D01Q1 to D32Q1, text function block is active.

Coils

D01EN to D32EN, enable of the text function block

Memory requirement of the text output function block

The text output function block function block requires 160 bytes of memory. This is irrespective of the text size.

Display

Up to 16 characters per line and up to 4 lines can be displayed.



Displaying a variable

Actual values of function blocks, markers (MB, MW and MD), as well as analog inputs (scalable) can be displayed. The display of date and time is also possible.

Entering setpoint values

This function is available from device version 04.

Function block setpoints, and marker values (MB, MW, MD) that are constants can be modified via the text function block.

The variables and setpoints can be inserted at any desired position in the text. The variables and setpoints can be 4, 7 and 11 places long.

Pay attention to the maximum character length of the variables setpoints. Otherwise, characters will be overwritten or not displayed.

Scaling

The values of the analog inputs and analog output can be scaled.

Range	Selectable display range	Example
0 to 10 V	0 to 9999	0000 to 0100
0 to 10 V	± 999	-025 to 050
0 to 10 V	± 9.9	-5.0 to 5.0

Function

Text display function block, D = Display: the text display in the circuit diagram functions as a normal M marker. If a text is stored in a marker, this is displayed in the easy display when the coil is set to 1. To use this function, the easy must be in RUN mode and the Status display must be activated before the text is displayed.

D 02 to D 32:

If several texts are present and are triggered, each text is automatically displayed in succession every 4 s. This procedure is repeated until

- none of the text output function blocks are set to 1.
- STOP mode is selected.
- the easy power supply is switched off.
- the **OK** or **DEL + ALT** buttons are used to switch to a menu.
- the text for D01 is displayed.

The following applies to **D 01**

: D1 is assigned as an alarm text. If D01 is activated, the text assigned to it will be displayed until

- the coil D 01EN is reset to 0.
- STOP mode is selected.
- the easy power supply is switched off.
- the **OK** or **DEL + ALT** buttons are used to switch to a menu.

Text entry

Text entry is only possible using EASY–SOFT, from version V 4.0 upwards.

Character set

All ASCII characters in upper and lower case are permissible.

- 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

The following special characters are permissible:

! „" # \$ % & ` () * + , - . / 0 1 2 3 4 5 6 7 8 9

Counter with actual value

```

QUANTITY
QTY: 0042
!COUNTING!
    
```

Analog values scaled as temperature values

```

TEMPERATURE
OUT -010 DEG
IN +010 DEG
HEATING
    
```

D1 as error message on fuse failure

```

FUSE FAULT
HOUSE 1
FAILED!
    
```

Figure 81: Text output examples

Entering a setpoint in a display

This function is available from device version 04.

The setpoints of function blocks can be incorporated within a text. This is useful if the PARAMETERS menu is not available for making entries.



The appropriate text function block must be available in order to modify a setpoint. The setpoint must be a constant.



The text remains static on the display while the values are entered. The actual values are updated.

```

ACT T01: 000:000
SETP  : 012:000
ACT C16: 04711
SET   : 10000
    
```

The example shows the following: 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
SET   : 10000
    
```

▶ Press the **ALT** button to move the cursor to the first editable value.

In this operating mode, you can use the cursor buttons \leftarrow \rightarrow to move between editable constants.

```

ACT T01: 000:000
SETP  :
012:000
ACT C16: 04711
    
```

▶ Press the **OK** button to move the cursor to the least significant value of the constant to be modified.

In this operating mode you use the ^v cursor buttons to change the value. Use the cursor buttons < > to move between places.

Use the **OK** button to accept the modified value. Use the **ESC** button to abort the entry and retain the previous value.

```
ACT T01: 000:000
SETP  : 015:000
ACT C16: 04711
SET   : 10000
```

► Press the **OK** button, the cursor switches to the operating mode: Move from constant to constant.

The modified value is transferred.

```
ACT T01: 000:000
SETP  : 012:000
ACT C16: 04711
SET   : 10000
```

Press the **ALT** button in order to leave Entry mode. (the **ESC** button has the same function here.)

Data function block

The data function block allows you to selectively save a value. Setpoint values for the function block can be saved in this manner.

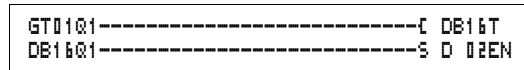


Figure 82: easy800 circuit diagram with data function block:

```
DB16      +
>I1
@V>
```

Parameter display and parameter set for the data function block:

DB16	Data block function block number 16
+	Appears in the parameter display
>I1	Input value
@V>	Actual value

Inputs

The function block input >I1 can be assigned the following operands:

- Constants
- Markers MD, MW, MB

- Analog inputs IA01 to IA04
 - IA01: terminal I7
 - IA02: terminal I8
 - IA03: terminal I11
 - IA04: terminal I12
- Analog output QA01
- Actual value ... QV> of another function block

Output

The function block output QV> can be assigned with the following operands:

- Markers MD, MW, MB
- Analog output QA01

Contacts

DB01Q1 to DB32Q1

Contact (make contact) DB..Q1 switches to status 1 if the trigger signal is set to 1.

Coils

DB01T_ to DB32T_, acceptance of the value at >I1 with a rising edge.

Memory requirement of the data function block

The data block function block requires 36 bytes of memory plus 4 bytes per constant at the function block input.

Retention

Data function blocks can be operated with retentive actual values. The quantity can be selected in the SYSTEM → RETENTION menu.

Function of the data function block

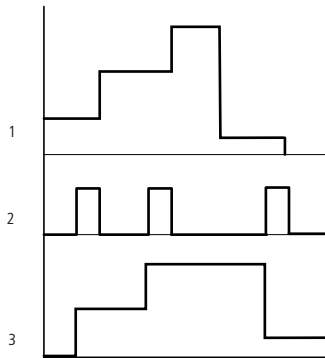


Figure 83: Signal diagram of data function block

- 1: value on input >I1
- 2: trigger coil DB..T_
- 3: value on DB..QV>



The value at input >I1 is only transferred with a rising trigger edge to an operand (e.g.: MD42, QA01) on output >QV>. Output QV retains its value until it is overwritten.

PID controller

This function block is available from device version 04.

easy800 provides 32 PID controllers DC01 to DC32. The PID controllers allow you to implement closed-loop control functions.



Caution!

A knowledge of closed-loop control is required in order to use the PID controllers.

The control system must be familiar so that the PID controller can function correctly.



Three separate manipulated variables can be output. One manipulated variable can be output via an analog output. Two manipulated variables can be processed via two pulse-width modulated outputs. It is therefore useful to run up to three closed-loop controllers per program simultaneously. Projects can be structured by selecting the controller number.

Example: Project with 3 devices

Program 1: Controller DC 10, 11

Program 2: Controller DC20, 21 and 22

Program 3: Controller DC30

Wiring a PID controller

You integrate a PID controller in your circuit as a contact and coil.

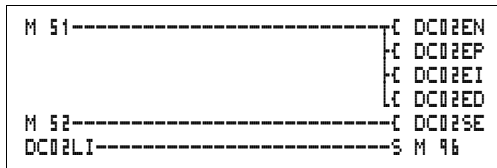


Figure 84: easy800-Circuit diagram with PID controller

```
DC02 UNP      +
>I1
>I2
>KP
>TN
>TV
>TC
>MV
@V>
```

Parameter display and parameter set for PID controller

DC02	PID controller function block number 02
UNP	Unipolar mode
+	Appears in the parameter display
>I1	Setpoint of PID controller
>I2	Actual value of PID controller
>KP	Proportional gain K_p
>TN	Reset time T_n
>TV	Rate time T_v

>TC	Scan time
>MV	Manual manipulated variable
QV>	Manipulated variable

In the parameter display of a PID controller you set the operating mode, the setpoints and enable the parameter display.

Operating modes of the PID controller

Parameter	Manipulated variable is output as
UNP	Unipolar 12-bit value 0 to +4095
EIP	Bipolar 13-bit value (signed 12-bit value) -4096 to +4095

Inputs

The function block inputs >I1, >I2, >KP, >TN, >TV, >TC and >MV can have the following operands:

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

Actual value ...QV>

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

- Markers MD, MW, MB
- Analog output QA01

Value range for inputs and outputs

		Value range	Resolution/Unit
>I1	Setpoint of PID controller	-32 768 to +32767	
>I2	Actual value of PID controller,	-32 768 to +32767	
>KP	Proportional gain K_p	0 to 65535	in -- /%
>TN	Reset time T_n	0 to 65535	in 100/ms
>TV	Rate time T_v	0 to 65535	in 100/ms
>TC	Scan time	0 to 65535	in 100/ms
>MV	Manual manipulated variable	-4096 to +4095	
QV>	Manipulated variable	0 to 4095 (unipolar) -4096 to +4095 (bipolar)	

Example:

		Value at input	Value processed in the function block.
>KP	Proportional gain K_p	1500	15
>TN	Reset time T_n	250	25 s
>TV	Rate time T_v	200	20 s
>TC	Scan time	500	50 s
>MV	Manual manipulated variable	500	500

Displaying the parameter set in the PARAMETERS menu

- + Access enabled
- – Access disabled

Contacts

DC01LI to DC32LI, value range of the manipulated variable exceeded.

Coils

- DC01EN to DC32EN: Enable PID controller;
- DC01EP to DC32EP: Activate proportional component;
- DC01EI to DC32EI: Activate integral component;

- DC01ED to DC32ED: Activate the differential component;
- DC01SE to DC32SE: Activate the manual manipulated variable

Memory requirement of the PID controller

The PID controller function block requires 96 bytes of memory plus 4 bytes per constant on the function block input.

Function of the PID controller function block

The PID controller works on the basis of the PID algorithm. According to this, the manipulated variable $Y(t)$ is the result of the calculation of the proportional component, an integral component and a differential component.



The PID controller must be enabled so that it can work. Coil DC..EN is active. If coil DC..EN is not active, the entire PID controller is deactivated and reset. The manipulated variable is set to zero.

The corresponding coils for the P, I and D components must be active.

Example: If only coils DC..EP and DC..EI are activated, the controller operates as a PI controller.



The device calculates the manipulated variable every time the scan time T_c has elapsed. If the scan time is zero, the manipulated variable is calculated every cycle.

Equation of PID controller:

$$Y(t) = Y_P(t) + Y_I(t) + Y_D(t)$$

$Y(t)$ = calculated manipulated variable with scan time t

$Y_P(t)$ = Value of the proportional component of the manipulated variable with scan time t

$Y_I(t)$ = Value of the integral component of the manipulated variable with scan time t

$Y_D(t)$ = Value of the differential component of the manipulated variable with scan time t

The proportional component in the PID controller

The proportional component Y_P is the product of the gain (K_p) and the control difference (e). The control difference is the difference between the setpoint (X_s) and the actual value (X_i) at a specified scan time. The equation used by the device for the proportional component is as follows:

$$Y_P(t) = K_p \times [X_s(t) - X_i(t)]$$

K_p = Proportional gain

$X_s(t)$ = Setpoint with scan time t

$X_i(t)$ = Actual value with scan time t

The integral component in the PID controller

The integral component Y_I is proportional to the sum of the control difference over time. The equation used by the device for the integral component is as follows:

$$Y_I(t) = K_p \times T_c/T_n \times [X_s(t) - X_i(t)] + Y_I(t-1)$$

K_p = Proportional gain

T_c = Scan time

T_n = Integration time (also known as reset time)

$X_s(t)$ = Setpoint with scan time t

$X_i(t)$ = Actual value with scan time t

$Y_I(t-1)$ = Value of the integral component of the manipulated variable with scan time $t-1$

The differential component in the PID controller

The differential component Y_D is proportional to the change in the control difference. So as to avoid step changes or jumps in the manipulated variable caused by the differential behaviour when the setpoint is changed, the change of the actual value (the process variable) is calculated and not the change in the control difference. This is shown in the following equation:

$$Y_D(t) = K_p \times T_v/T_c \times (X_i(t-1) - X_i(t))$$

K_p = Proportional gain

T_c = Scan time

T_v = Differential time of the control system (also called the rate time)

$X_i(t)$ = Actual value with scan time t

$X_i(t-1)$ = Actual value with scan time $t-1$

Scan time T_c

Scan time T_c determines the duration of the interval in which the function block is called by the operating system for processing. The value range is between 0 and 6553.5 s.

If the value 0 is set, the cycle time of the device is the pause time between the function block calls.

→ The device cycle time varies according to the length of the program. With a scan time of 0 s, this may cause an irregular control response.

→ Use the Set cycle time function block (→ page 226) in order to keep the cycle time of the device constant.

Manual mode of the PID controller

A value must be present at the $\setminus MV$ input in order to set the manipulated variable directly. If the coil DC..SE is activated, the value at $\setminus MV$ is transferred as manipulated variable QV . This value is present for as long as the DC..SE coil is activated or the value at the $\setminus MV$ input is changed. If coil DC..SE is deactivated, the control algorithm is reactivated.

→ Extreme changes in the manipulated variable can occur when the manual manipulated variable is transferred or deactivated.

→ If the function block is running in UNI (unipolar) mode, a negative signed manipulated variable value will be output.

Signal smoothing filter

This function block is available from device version 04.

easy provides 32 signal smoothing filters FT01 to FT32. The function block allows you to smooth noisy input signals.

Wiring a signal smoothing filter

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

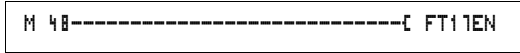


Figure 85: easy800 circuit diagram with smoothing function block

```

FT11      +
>I1
>TG
>KP
QV>
    
```

Parameter display and parameter set for the FT function block:

FT11	FT PT1 signal smoothing filter function block, number 17
+	Appears in the parameter display
>I1	Input value
>TG	Recovery time
>KP	Proportional gain
QV>	Output value, smoothed



The recovery time T_g is the time in which the output value is calculated.

The recovery time T_g must be set so that it is an integer multiple of the cycle time or controller scan time T_c .

Inputs

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

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

Output

The function block output QV can be assigned the following operands:

- Markers MD, MW, MB
- Analog output QA01

Value range for inputs and outputs

		Value range	Resolution/Unit
$I1$	Input value of the function block	-32 768 to +32 767	
TG	Recovery time T_g	0 to 65 535	in 100/ms
KP	Proportional gain K_p	0 to 65 535	in -- /%
QV	Output value	-32 768 to +32 767	

Example:

		Value at input	Value processed in the function block.
TG	Recovery time T_g	250	25 s
KP	Proportional gain K_p	1500	15

Displaying the parameter set in the PARAMETERS menu

- + Access enabled
- – Access disabled

Coil

FT01EN to FT32EN, function block enable

Memory requirement of the FT function block

The FT function block requires 56 bytes of memory plus 4 bytes per constant on the function block input.

Function of the signal smoothing filter function block



The signal smoothing filter must be enabled so that it can work. Coil FT..EN is active. If coil FT..EN is not active, the entire function block is deactivated and reset. The output value is set to zero.

If the function block is called for the first time, the output value is initialised with the input value when the device is started or after a reset. This speeds up the startup behaviour of the function block.



The function block updates the output value every time recovery time T_g expires.

The function block operates according to the following equation:

$$Y(t) = [T_a/T_g] \times [K_p \times x(t) - Y(t-1)]$$

$Y(t)$ = Calculated output value for scan time t

T_a = Scan time

T_g = Recovery time

K_p = Proportional gain

$x(t)$ = Actual value with scan time t

$Y(t-1)$ = Output value with scan time $t - 1$

Scan time:

Scan time T_a depends on the set recovery time value.

Recovery time T_g	Scan time T_a
0.1 s to 1 s	10 ms
1 s to 6553 s	$T_g \times 0.01$

GET, fetch a value from the network

The function block allows you to selectively read (get) a 32 bit value from the network. The GET function block fetches data which another station has made available on the easy-NET network with a PUT function block.

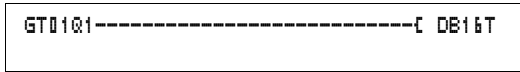
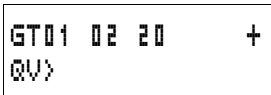


Figure 86: easy800 circuit diagram with GET function block



Parameter display and parameter set for the GET function block:

GT01	GET function block (fetch a value from the network), number 01
02	Station number from which the value is sent. Possible station number: 01 to 08
20	Send function block (PT 20) of the sending station. Possible function block number: 01 to 32
+	Appears in the parameter display
QV	Actual value from the network

Output

The function block output QV can be assigned the following operands:

- Markers MD, MW, MB
- Analog output QA01

Contacts

GT01Q1 to GT32Q1

Contact (make contact) GT..Q1 switches to status 1, if a new value transferred on the easy-NET network is present.

Memory requirement of the GET function block

The GET function block requires 28 bytes of memory.

GET diagnostics

The GET function block only functions when the easy-NET network is functioning correctly (→ Section "Signs of life of the individual stations and diagnostics", Page 259).

Function of the GET function block

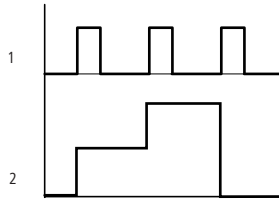


Figure 87: Signal diagram of GET function block

1: GT..Q1

2: value on GT..QV>



The GET function blocks are assigned the value 0 when the power supply is switched on.

Seven-day time switch

easy800 is equipped with a real-time clock which you can use in the circuit diagram as a 7-day time switch and a year time switch.



The procedure for setting the time is described under Section "Setting date, time and daylight saving time" on Page 273.

**Caution!**

The 7-day time switch of easy800 devices of device version 03 or lower have the following characteristics.

Requirement:

- The function block was entered directly on the easy800.
- 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.

Behaviour:

- The time switch switches off on Sunday at 00:00.
- This cannot be the required off time!

Solution:

- Assign all time switch channels with the required on and off times.
- Use the program EASY-SOFT (-PRO) or EASY-SOFT (-PRO) for making the entry.

In this case, this behaviour does not occur.

easy provides 32 seven-day time switches HW01 to HW32 for a total of 128 switching times.

Each time switch has four channels which you can use to set four on and off times. The channels are set via the parameter display.

The time is backed up in the event of a power failure and continues to run, although the time switch relays will no longer switch. The contacts are kept open when de-energized. Refer to Chapter "Technical data", Page 317 for information on the buffer time.

Wiring of a 7-day time switch

A 7-day time switch is integrated into the circuit diagram as a contact.

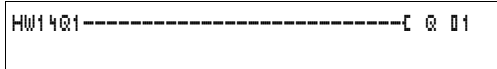


Figure 88: easy800 circuit diagram with 7-day time switch

```

HW14 A      +
  >DY1
  >DY2
  >ON
  >OFF

```

Parameter display and parameter set for the 7-day time switch HW:

HW14	7-day time switch function block number 14
A	Time switch channel A
+	Appears in the parameter display
>DY1	Day 1
>DY2	Day 2
>ON	On time
>OFF	Off time
F	

Channels

4 channels are available per time switch, channels A, B, C and D. These channels all act on the contact of the 7-day time switch.

Day 1 and day 2

Either the time period acts from day 1 to day 2, e.g. Monday to Friday, or for one day only.

Monday = MO, Tuesday = TU, Wednesday = WE, Thursday = TH, Friday = FR, Saturday = SA, Sunday = SU,

Time

00:00 to 23:59

Displaying the parameter set in the PARAMETERS menu

- + Access enabled
- – Access disabled

Contacts

HW01Q1 to HW32Q1

Memory requirement of the 7-day time switch

The 7-day time switch function block requires 68 bytes of memory plus 4 bytes per channel used.

Function of the 7-day time switch

The switching points are defined according to the parameters entered.

MO to FR: on the weekdays Mo, Tu, We, Th, Fr

ON 10:00, OFF 18:00: on and off switching times for the individual days of the week.

MO: every Monday

ON 10:00: switch on time

SA: every Saturday

OFF 18:00: switch off time

Switching on working days

Time switch HW01 switches on Monday to Friday between 6:30 and 9:30 and between 17:00 and 22:30.

HW01 A	+	HW01 B	+
>DY1 MO		>DY1 MO	
>DY2 FR		>DY2 FR	
>ON 06:30		>ON 17:00	
>OFF 09:30		>OFF 22:30	

Switching at the weekend

Time switch HW02 switches on at 16:00 on Friday and switches off at 6:00 on Monday.

HW02 A +	HW02 B +
>DY1 FR	>DY1 MO
>DY2	>DY2
>ON 16:00	>ON
>OFF	>OFF 06:00

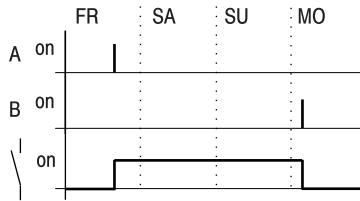


Figure 89: Signal diagram of weekend

Overnight switching

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

HW03 D +
>DY1 MO
>DY2
>ON 22:00
>OFF 06:00

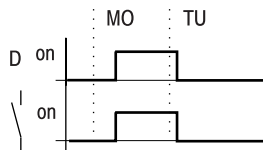


Figure 90: Signal diagram for night switching



If the Off time is before the On time, easy will switch off on the following day.

Time overlaps

The time settings of a time switch overlap. The clock switches on at 16:00 on Monday, whereas on Tuesday and Wednesday it switches on at 10:00. On Monday to Wednesday the switching-off time is 22:00.

HWD4 A	+	HWD4 B	+
>DY1 MO		>DY1 TU	
>DY2 WE		>DY2 WE	
>ON 16:00		>ON 10:00	
>OFF 22:00		>OFF 00:00	

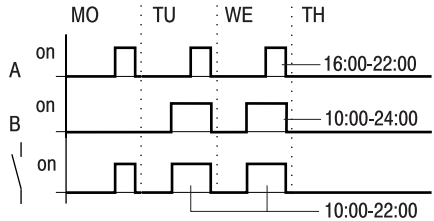


Figure 91: Signal diagram of overlaps



Switch-on and switch-off times are always based on the channel which switches first.

Response in the event of a power failure

The power is removed between 15:00 and 17:00. The relay drops out and remains off, even after the power returns, since the first switching-off time was at 16:00.

HWD5 A	+	HWD5 B	+
>DY1 MO		>DY1 MO	
>DY2 SU		>DY2 SU	
>OFF 16:00		>ON 12:00	
		>OFF 18:00	



When switched on, easy always updates the switching state on the basis of all the available switching time settings.

24 hour switching

The time switch is to switch for 24 hours. Switch-on time at 0:00 on Monday and switch-off time at 0:00 on Tuesday.

HW20 A	+	HW20 B	+
>DY1 MO		>DY1 TU	
>DY2		>DY2	
>ON 00:00		>ON	
>OFF		>OFF 00:00	

Year time switch

easy800 is equipped with a real-time clock which you can use in the circuit diagram as a 7-day time switch and a year time switch.



The procedure for setting the time is described under Section "Setting date, time and daylight saving time" on Page 273.

easy provides 32 year time switches HY01 to HY32 for a total of 128 switching times.

Each time switch has four channels which you can use to set four on and off times. The channels are set via the parameter display.

The time and date are backed up in the event of a power failure and continue to run. However, the time switch relays will no longer continue to switch. The contacts are kept open when de-energized. Refer to Chapter "Technical data", Page 317 for information on the buffer time.

Wiring of a year time switch

A year time switch is integrated into the circuit diagram as a contact.

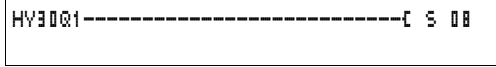


Figure 92: easy800 circuit diagram with year time switch

```

HY30 B +
>ON
>OFF
  
```

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

HY30	Year time switch function block number 30
B	Time switch channel B
+	Appears in the parameter display
>ON	Switch on time
>OFF	Switch off time
F	

Channels

4 channels are available per time switch, channels A, B, C and D. These channels all act on the contact of the year time switch.

Date

Day.Month.Year: DD.MM.YY

Example 11.11.02

On/off switch points

ON: switch on time

OFF: switch off time



The switch on year must not be later than the switch off year. Otherwise the year time switch will not function.

Displaying the parameter set in the PARAMETERS menu

- + Access enabled
- – Access disabled

Contacts

HY01Q1 to HY32Q1

Memory requirement for the year time switch

The year time switch function block requires 68 bytes of memory plus 4 bytes per channel used.

Function of the year time switch function block

The year time switch can operate with ranges, individual days, months, years or combinations.

Years

ON: 2002 to OFF: 2010 means:

Switch on at 00:00 on 01.01.2002 00 and switch off at 00:00 on the 01.01.2011.

Months

ON: 04 to OFF: 10 means:

Switch on at 00:00 on 1st April and switch off at 00:00 on 1st November

Days

ON: 02 to OFF: 25 means:

Switch on at 00:00 on the 2nd and switch off at 00:00 on the 26th

Rules for the year time switch

The contact switches in the defined years (ON to OFF), the defined months (ON to OFF) and in the days entered (ON to OFF).

Time ranges must be input with two channels, one for ON and one for OFF.

Overlapping channels:

The first ON date switches on and the first OFF date switches off.



Avoid incomplete entries. It hinders transparency and leads to unwanted functions.

```

HY01 A      +
>ON  --.---.02
>OFF --.---.05

```

Example 1

Year range selection

The year time switch HY01 should switch on at 00:00 on January 1 2002 and remain on until 23:59 on 31 December 2005.

```

HY01 A      +
>ON  --.03.--
>OFF --.09.--

```

Example 2

Month range selection

The year time switch HY01 should switch on at 00:00 on 01st March and remain on until 23:59 on 30th September.

```

HY01 A      +
>ON  01.---.--
>OFF 28.---.--

```

Example 3

Day range selection

The year time switch HY01 should switch on at 00:00 on the 1st of each month and remain on until 23:59 on the 28th of each month.

```

HY01 A      +
>ON  25.12.--
>OFF 26.12.--

```

Example 4

Holiday selection

The year time switch HY01 should switch on at 00:00 on the 25.12 each year and remain on until 23:59 on 26.12. "Christmas program"

Example 5

Time range selection

The year time switch HY01 should switch on at 00:00 on 01.05 each year and remain on until 23:59 on the 31.10. "Open air season"

```

HY01 A      +
>ON  01.05.--
>OFF --.---.--

```

```

HY01 B      +
>ON  --.---.--
>OFF 31.10.--

```

Example 6
Overlapping ranges

The year time switch HY01 channel A switches on at 00:00 on the 3rd of the months 5, 6, 7, 8, 9, 10 and remains on until 23:59 on the 25th of these months.

The year time switch HY01 channel B switches on at 00:00 on the 2nd in the months 6, 7, 8, 9, 10, 11, 12 and remains on until 23:59 on 17th of these months.

```
HY01 A      +
>ON  03.05.--
>OFF 25.10.--
```

```
HY01 B      +
>ON  02.06.--
>OFF 17.12.--
```

Total number of channels and behaviour of the contact HY01Q1:

The time switch will switch on at 00:00 from the 3rd May and off at 23:59 on the 25th May.

In June, July, August, September, October, the time switch will switch on at 00:00 on the 2nd of the month and switch off at 23:59 on the 17th.

In November and December, the time switch will switch on at 00:00 on the 2nd of the month and switch off at 23:59 on the 17th.

Value scaling

This function block is available from device version 04.

easy provides 32 value scaling function blocks LS01 to LS32. The function block enables you to convert values from one value range to another one. In this way it is possible to reduce or increase values.

Wiring of a scaling function block

You can integrate a value scaling function block into your circuit as a coil.



Figure 93: easy800 circuit diagram with LS value scaling


```

LS27      +
>I1
>X1
>Y1
>X2
>Y2
QV>

```

Parameter display and parameter set for the LS function block:

LS27	LS value scaling function block number 27
+	Appears in the parameter display
>I1	Input value, actual value source range
>X1	Lower value of source range
>Y1	Lower value of target range
>X2	Upper value of source range
>Y2	Upper value of target range
QV>	Output value, scaled

Inputs

The function block inputs >I1, >X1, >X2, >Y1 and >Y2 can have the following operands:

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

Output

The function block output QV> can be assigned with the following operands:

- Markers MD, MW, MB
- Analog output QA01

Value range for inputs and outputs

		Value range
>I1	Input value of the function block	-2147483648 to +2147483647
>X1	Lower value of source range	
>X2	Lower value of target range	
>Y1	Upper value of source range	
>Y2	Upper value of target range	
QV>	Output value	

Displaying the parameter set in the PARAMETERS menu

- + Access enabled
- - Access disabled

Coil

VC01EN to VC32EN, function block enable

Memory requirement of the LS function block

The LS function block requires 64 bytes of memory plus 4 bytes per constant on the function block input.

Function of the function block



The scaling function block must be enabled so that it can work. Coil LS..EN is active. If coil LS..EN is not active, the entire function block is deactivated and reset. The output value is set to zero.

The function block operates according to the following equation:

$$Y(x) = X \times \frac{Y_2 - Y_1}{X_2 - X_1} + \frac{X_2 \times Y_1 - X_1 \times Y_2}{X_2 - X_1}$$

- Y(x) = Actual output value of target range
- X = Actual input value of source range
- X₁ = Lower value of source range
- X₂ = Upper value of source range
- Y₁ = Lower value of target range
- Y₂ = Upper value of target range

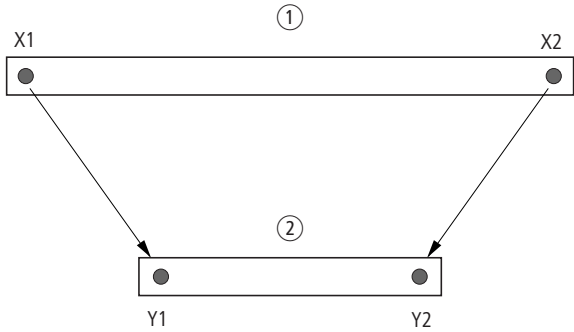


Figure 94: Value scaling function block – Reduce value range

- ① Source range
- ② Target range

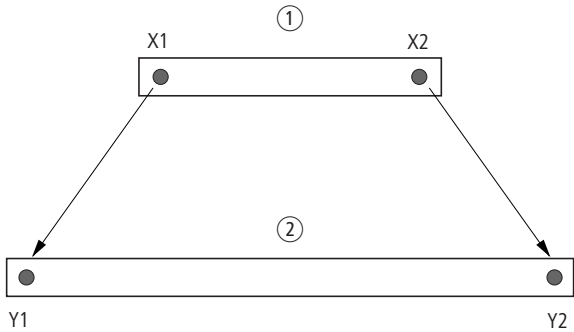


Figure 95: Value scaling function block – Increase value range

- ① Source range
- ② Target range

Example 1:

The source range is a 10-bit value, source is the analog input IA01.

The target range has 12 bits.

```

LS01      +
>I1 IA01
>X1 0
>Y1 0
>X2 1023
>Y2 4095
QV>

```

Parameter display and parameter set for the LS01 function block:

The actual value at the analog input IA01 is 511.
The scaled output value is 2045.

Example 2:

The source range has 12 bits.

The target range has 16 signed bits.

>I1 = DC01QV

>X1 = 0

>X2 = 4095

>Y1 = -32768

>Y2 = +32767

The actual value at the analog input DC01QV 1789.
The scaled output value is -4137.

Jumps

Jumps can be used to optimise the structure of a circuit diagram or to implement the function of a selector switch. For example, jumps can be used to implement whether manual/automatic mode, or different machine programs are to be selected.

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

Circuit diagram symbols for jumps

Contact	
Make contact ¹⁾	:
Numbers	01 to 32
Coils	⌈
Numbers	01 to 32
Coil function	⌈, ⌋, ⌌, ⌍, ⌎

1) can only be used as first leftmost contact

Function

If the jump coil is triggered, the circuit connections coming directly after it will not be processed. The states of the coils before the jump will be retained, unless they are overwritten in circuit connections that were not missed by the jump. Jumps are always made forwards, i.e. the jump ends on the first contact with the same number as that of the coil.

- Coil = jump when 1
- Contact only at the first left-hand contact position = Jump destination

The Jump contact point will **always be set to 1**



Backward jumps are not possible with easy due to the way it operates. If the jump label does not come after the jump coil, the jump will be made to the end of the circuit diagram. The last circuit connection will also be skipped.

If a jump destination is not present, the jump is made to the end of the circuit diagram.

Multiple use of the same jump coil and jump contact is possible as long as this is implemented in pairs, i.e.:

Coil ⌈:1/jumped range/Contact:1,

Coil ⌋:1/jumped range/Contact:1 etc.



Attention!

If rungs are skipped, the states of the coils are retained. The time value of timing relays that have been started will continue to run.

Power flow display

Jumped ranges are indicated by the coils in the power flow display.

All coils after the jump coil are shown with the symbol of the jump coil.

Example

A selector switch allows two different sequences to be set.

- Sequence 1: Switch on Motor 1 immediately.
- Sequence 2: Switch on Guard 2, Wait time, then switch on Motor 1.

Contacts and relays used:

- I1 Sequence 1
- I2 Sequence 2
- I3 Guard 2 moved out
- I12 Motor-protective circuit-breaker switched on
- Q1 Motor 1
- Q2 Guard 2
- T 1 Wait time 30.00 s, on-delayed
- D 1 Text "motor-protective circuit-breaker tripped"

Operating modes

- Q: Acts on the outputs Q., *Q., S., *S., *SN., QA01; *: network station address
- M: acts on the marker range MD01 to MD48.
- ALL: acts on Q and M.

Contacts

MR01Q1 to MR32Q1

The contact switches on the marker if the trigger coil MR..T has the 1 state.

Coils

MR01T to MR32T: trigger coils

Memory requirement of the data function block

The master reset function block requires 20 bytes of memory.

Function of the data master reset

The outputs or the markers are set to the 0 state in accordance with the operating mode when a rising edge is detected on the trigger coil.



The master reset function block should be used as the last function block in a program so that all data ranges are deleted safely. Otherwise subsequent function blocks may overwrite the data ranges.

The contacts MR01Q1 to MR32Q1 assume the state of their own trigger coil.

Numerical converters

This function block is available from device version 04.

easy800 provides 32 numerical converters NC01 to NC32.

A numerical converter function block enables you to convert BCD coded values to decimal values or decimal coded values to BCD coded values.

Wiring of a numerical converter

A numerical converter in the circuit diagram only has the enable coil.

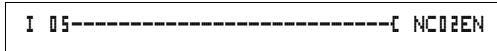
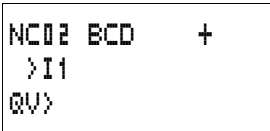


Figure 97: easy800 circuit diagram with numerical converter



Parameter display and parameter set for the numerical converter:

NCO2	Numerical converter function block number 02
BCD	Convert BCD code to decimal value mode
+	Appears in the parameter display
>I1	Input value
@V>	Output value

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

Numerical converter modes

Parameter	Mode
BCD	Convert BCD coded values to decimal values
BIN	Convert decimal value to BCD coded values

Number range

Value	Number system
-161061273 to +161061273	BCD
-9999999 to +9999999	Decimal

BCD code	Decimal value
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010 to 1111	Not permissible
10000	10
10001	11



The BCD code only allows the number range 0_{hex} to 9_{hex}. The number range A_{hex} to F_{hex} cannot be represented. The NC function block converts the impermissible range as 9.

Inputs

The function block input $\rightarrow I 1$ can be assigned the following operands:

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

Actual value ...QV>

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

- Markers MD, MW, MB
- Analog output QA01

Displaying the parameter set in the PARAMETERS menu

- + Access enabled
- – Access disabled

Coil

NC01EN to NC32EN: enable coil.

Memory requirement of the numerical converter

The numerical converter function block requires 32 bytes of memory plus 4 bytes per constant on the function block input.

Function of the numerical converter function block

The numerical converter function block must be enabled so that it can work. Coil NC..EN is active. If coil NC..EN is not active, the entire function block is deactivated and reset. The output value is set to zero.

BCD mode

The BCD value at $\text{I}1$ is provided in decimal format at the input. The binary value is formed from this. The binary value is interpreted as a BCD value. Values greater than 9 (1001) are set to the value 9. The BCD value is output as a decimal value at the output QV>.

Example 1:

Input value $\text{I}1$: +9_{dec}

Binary value: 1001

Decimal value QV>: + 9

Example 2:

Input value $\text{I}1$: +14_{dec}

Binary value: 1110

Decimal value QV>: + 9



The highest binary value represented in BCD is $1001 = 9$. All higher binary values from 1010 to 1111 are output as 9. This behaviour is correct as BCD encoders normally don't generate these values.

Example 3:

Input value>I1: 19_{dec}

Binary value: 00010011

Decimal value QV>: 13

Example 4:

Input value>I1: 161061273_{dec}

Binary value: 1001100110011001100110011001

Decimal value QV>: 9999999

Example 5:

Input value>I1: -61673_{dec}

Binary value: 10000000000000001111000011101001

Decimal value QV>: -9099



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

Example 6:

Input value>I1: 2147483647_{dec}

Binary value: 01111111111111111111111111111111

Decimal value QV>: 9999999



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

BIN mode

The decimal value is assigned to input I1 . The decimal value is represented as a BCD coded value. The BCD coded value is interpreted as a hexadecimal value and output as a decimal value at output QV>.

Example 1:

Input value >I1: $+7_{\text{dec}}$

BCD binary value: 0111

Hexadecimal value: 0111

Decimal value QV>: + 7

Example 2:

Input value >I1: $+11_{\text{dec}}$

BCD binary value: 00010001

Hexadecimal value: 00010001

Decimal value QV>: +17 (1 + 16)

Hexadecimal value:

Bit 0 has the value 1.

Bit 4 has the value 16

Total: Bit 0 plus Bit 4 = 17

Example 3:

Input value >I1: 19_{dec}

BCD binary value: 00011001

Hexadecimal value: 00011001

Decimal value QV>: 25 (1 + 8 + 16)

Example 4:

Input value >I1: 9999999_{dec}

BCD binary value: 1001100110011001100110011001

Hexadecimal value: 1001100110011001100110011001

Decimal value QV>: 161061273

Example 5:

Input value >I1: -61673_{dec}

BCD binary value:

10000000000001100001011001110011

Hexadecimal value:

10000000000001100001011001110011

Decimal value QV>: -398963



Bit 32 is the sign bit. Bit 32 = 1 → Sign = Minus.

Example 6:

Input value >I1: 2147483647_{dec}

BCD binary value:

01111111111111111111111111111111

Hexadecimal value:
01111111111111111111111111111111
Decimal value QV>: 161061273



Values greater than 9999999 are output as 161061273.
Values less than -9 999999 are output as
-161061273. The working range of the function block
has been exceeded.

Operating hours counter

easy800 provides 4 independent operating hours counters. The counter states are retained even when the device is switched off. As long as the enable coil of the operating hours counter is active, easy800 counts the hours in minute cycles.

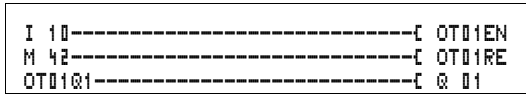
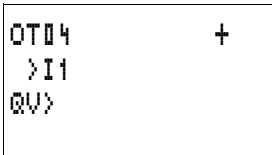


Figure 98: easy800 circuit diagram with operating hours counter.



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

OT04	Operating hours counter number 04
+	Appears in the parameter display
>I1	Upper threshold value in hours
QV>	Actual value of the operating hours counter

Contacts

OT01Q1 to OT04Q1

The contact switches when the upper threshold value has been reached (greater than or equal to).

Coils

- OT01EN to OT04EN: enable coil
- OT01RE to OT04RE: reset coil

Memory requirement of the operating hours counter

The operating hours counter function block requires 36 bytes of memory plus 4 bytes per constant on the function block input.

Function of the operating hours counter function block

If the enable coil OT..EN is triggered to the 1 state, the counter adds the value 1 to its actual value every minute (basic clock rate: 1 minute).

If the actual value on QV> reaches the setpoint value of >I1, the contact OT..Q1 switches for as long as the actual value is greater than or equal to the setpoint value.

The actual value is retained in the unit until the Reset coil OT..RE is actuated. The actual value is then set to zero.



Operating mode change RUN, STOP, Voltage On, Off, Delete program, Change program, Load new program. All these actions do not delete the actual value of the operating hours counter.

Accuracy

The operating hours counter is accurate to the nearest minute. If the enable coil signal is terminated within a minute, the value for seconds is lost.

The value range of the operating hours counter is between 0 hours and 100 years.

PUT, send a value onto the network

The function block allows you to selectively send a 32 bit value onto the network. The PUT function block provides data on the easy-NET that another station indicates it requires via the GET function block.

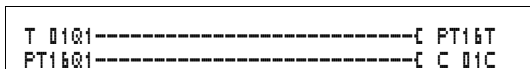


Figure 99: easy800 circuit diagram with PUT function block

```

PT01 11      -
>I1
    
```

Parameter display and parameter set for the PUT function block:

PT01	PUT function block (places a value onto the network), number 11
-	Does not appear in the parameter display
>I1	Setpoint value which is put onto the easy-NET network

Input

The function block input >I1 can be assigned the following operands:

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

Contacts

PT01Q1 to PT32Q1: state of the trigger coil

Coils

PT01T to PT32T: trigger coils

Memory requirement of the PUT function block

The PUT function block requires 36 bytes of memory plus 4 bytes per constant on the function block input.

PUT diagnostics

The PUT function block only functions when the easy-NET network is functioning correctly (→ Section "Signs of life of the individual stations and diagnostics", Page 259).

Function of the PUT function block

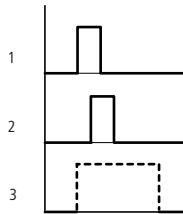


Figure 100: Signal diagram of PUT function block

- 1: trigger coil
- 2: trigger coil contact feedback
- 3: send

Pulse width modulation

This function block is available from device version 04.

easy800 provides 2 pulse width modulation function blocks PW01 and PW02. The function blocks are connected directly to the outputs.

They are assigned as follows:

PW01 → Q1

PW02 → Q2



When using the pulse width modulation function block with a minimum on time of less than 1 s only use devices with transistor outputs.

The pulse width modulation function block is primarily used for outputting the manipulated variable of a PID controller. The maximum frequency is 200 kHz. This corresponds to a period duration of 5 ms. The maximum period duration is 65.5 s.

Wiring a pulse width modulation function block

A pulse width modulation function block is integrated in the circuit diagram as a contact or coil.



To prevent unpredictable switching states, use each coil of a relay once only in the circuit diagram.

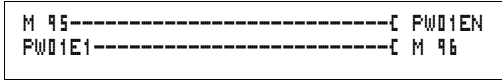


Figure 101: easy800 circuit diagram with pulse width modulation.

FWD2	+
>SV	
>PD	
>ME	

Parameter display and parameter set for pulse width modulation:

FWD2	Pulse width modulation function block number 02
+	Appears in the parameter display
>SV	Manipulated variable input
>PD	Period duration in ms
>ME	Minimum on duration, minimum off duration in ms

The parameter display for a timing relay is used to modify the period duration, the minimum on time and the enabling of the parameter display.

Value and time ranges

Parameter	Value and time range	Resolution
SV	0 to 4095	1 digit
PD	0 to 65535	ms
ME	0 to 65535	ms



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

Inputs

The function block inputs >SV, >PD and >ME can be assigned the following operands:

- Constants
- Markers MD, MW, MB

- Analog inputs IA01 to IA04
 - IA01: terminal I7
 - IA02: terminal I8
 - IA03: terminal I11
 - IA04: terminal I12
- Analog output QA01
- Actual value ... QV> of another function block

Displaying the parameter set in the PARAMETERS menu

- + Access enabled
- – Access disabled

Contacts

PW01E1 to PW02E1, the minimum on duration or the minimum off duration was undershot.

Coils

PW01EN to PW02EN, enable coil.

Memory requirement of the function block

The pulse width modulation function block requires 48 bytes of memory plus 4 bytes per constant on the function block input.

Function of the pulse width modulation function block



The pulse width modulation function block must be enabled so that it can work. Coil PW..EN is active. If coil PW..EN is not active, the entire function block is deactivated and reset. The output value is set to zero.

The manipulated variable at function block input >SV is converted to a pulse string with a constant period duration. The pulse width is proportional to the manipulated variable SV. The period duration and the minimum on duration can be selected as required within the specified limits.

The function block causes a direct output of the pulse to the corresponding output. The output image of the circuit diagram is always updated.



The following applies if the output of a pulse width modulator is used as a coil in the circuit diagram:

The state of the output is not refreshed from the circuit diagram.



The following applies to the minimum on duration:

- The minimum on duration is the same as the minimum off duration.
- The minimum on duration must not exceed 10 % of the period duration. The ratio of period duration/minimum on duration (P/M) determines which percentage of the manipulated variable has no effect. The minimum on duration must be set as low as possible so that P/M ratio is as high as possible. If the minimum on duration must not be too low, due to the output relay, the period duration must be increased accordingly.
- The minimum on duration is 1 ms.
- If the actual value of the pulse length is less than the minimum on duration, the minimum on duration has the effect of the pulse time. Note the state of the contact PW..E1.
- If the off duration of the pulse is less than the minimum on duration, outputs Q1 and Q2 are continuously in operation. Note the state of the contact PW..E1.

Setting date/time

This function block allows you to selectively place the date and time onto the network. All other stations accept the date and time of the sending station. The function block name is SC01 (send clock).

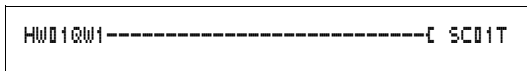


Figure 102: easy800 circuit diagram with SC function block

Parameter display and parameter set for the SC function block:

The SC01 function block has no parameters as it is a triggered system service.

Coil

SC01T: trigger coil

Memory requirement of the SC function block

The SC function block requires 20 bytes of memory.

SC diagnostics

The SC function block only functions when the easy-NET network is functioning correctly (→ Section "Signs of life of the individual stations and diagnostics", Page 259).

Function of the date/time function block

If the trigger coil of the function block is activated, the current date, the day of the week and time from the sending station is automatically put onto the easy-NET network. All other network stations must accept these values.



The station that sends its date and time does this when the seconds value is zero.

Example: The trigger pulse is actuated at 03:32:21 (hh:mm:ss). The other stations are synchronised at 03:33:00. This time is accessed by all other stations.

This process can be repeated as often as desired. The trigger coil must be triggered again from the 0 to the 1 state.

Accuracy of time synchronisation

The maximum time deviation between the functional stations is 5 s.

Set cycle time

This function block is available from device version 04.

easy800 provides one set cycle time function block ST01. The set cycle time function block is an additional function block for the PID controller.

The set cycle time function block provides a fixed cycle time for processing the circuit diagram and the function blocks.

Wiring a set cycle time function block

The ST set cycle time function block is integrated in the circuit diagram as a coil.



To prevent unpredictable switching states, use each coil of a relay once only in the circuit diagram.

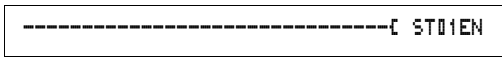
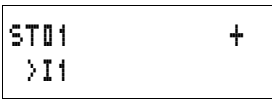


Figure 103: easy800 circuit diagram with enabling of set cycle time function block.



Parameter display for set cycle time:

ST01	Set cycle time function block number 01
+	Appears in the parameter display
>I1	Set cycle time

The parameter display for a timing relay is used to modify the set cycle time, the minimum on time and the enabling of the parameter display.

Time range

Parameter	Value and time range	Resolution
I1	0 to 1000	ms

Inputs

The function block input $\rightarrow I1$ can be assigned the following operands:

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

Displaying the parameter set in the PARAMETERS menu

- + Access enabled
- – Access disabled

Coils

ST01EN, enable coil.

Memory requirement of the function block

The set cycle time function block requires 24 bytes of memory plus 4 bytes per constant on the function block input.

Function of the set cycle time function block

The function block is used to define a fixed cycle time.



The function block must be enabled so that it can work. Coil ST01EN is active. If coil ST01EN is not active, the entire function block is deactivated and reset.

Actual cycle time is less than the set cycle time:

If the maximum cycle time present is less than the set cycle time, the set cycle time is constant.

Actual cycle time is greater than the set cycle time:

If the maximum cycle time present is greater than the set cycle time, the set cycle time has no effect.



Attention!

The shorter the cycle time, the faster the control and regulation process.

Set the set cycle time as short as possible. The processing of the function blocks, reading of the inputs and writing of outputs is only carried out once every cycle. Exception: All function blocks that are processed irrespective of the controller.

Timing relay

easy800 provides 32 timing relays from T 01 to T 32.

A timing relay is used to change the switching duration and the make and break times of a relay contact. The delay times can be configured between 5 ms and 99 h 59 min.

Wiring a timing relay

You integrate a timing relay into your circuit in the form of a contact and coil. The function of the relay is defined via the parameter display. The relay is started via the trigger coil T..EN and can be selectively reset via the reset coil T..RE. The actual timeout running can be stopped via the third coil T..ST.



To prevent unpredictable switching states, use each coil of a relay once only in the circuit diagram.

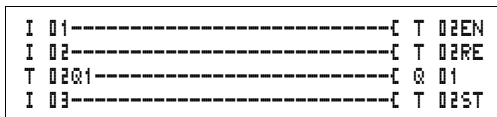


Figure 104: easy800 circuit diagram with timing relay


```
T 02 X M:S +
>I1
>I2
@V>
```

Parameter display and parameter set for timing relay:

T 02	Timing relay function block number 02
X	On-delayed mode
M:S	Time range Minute: Seconds
+	Appears in the parameter display
>I1	Time setpoint value 1
>I2	Time setpoint value I2 (on a timing relay with 2 setpoint values)
@V>	Timed-out actual time in RUN mode

The parameter display for a timing relay is used to modify the switching function, time base or setpoint times and enable the parameter display.

Timing relay modes

Parameter	Switch function
X	On-delayed switching
?X	On-delayed with random time range
■	Off-delayed switching
?■	Off-delayed with random time range
X■	On and off delayed
□	Off-delayed with random time range, retriggerable
?□	Off-delayed with random time range, retriggerable
?X■	On and off delayed switching with random time, 2 time setpoints
⌌	Switch with single-pulse
⌌	Switch with flashing, synchronous, 2 time setpoint values
⌌	Switch with flashing, asynchronous, 2 time setpoint values

Time range

Parameter	Time range and setpoint time	Resolution
S 000.000	Seconds, 0.005 to 999.995 s for constants and variable values	5 ms
M:S 00:00	Minutes: Seconds 00:00 to 99:59 only for constants and variable values	1 s
H:M 00:00	Hours: Minutes, 00:00 to 99:59 only for constants and variable values	1 min.



Minimum time setting:
0.005 s (5 ms).

If a time value is less than the easy cycle time, the elapsed time will only be recognised in the next cycle.

Inputs

The function block inputs $\rightarrow I1$ and $\rightarrow I2$ can have the following operands:

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

Outputs

Actual value ...QV>

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

- Markers MD, MW, MB
- Analog output QA01

Variable setpoint values

Behaviour of the setpoint value when variable values are used.

- Variable values can be used.
- Variable values must be transferred using operands.
- With the time base "s" the value is accepted as a "value in ms".
- The last position is rounded up to a zero or five.
- With the time base "M:S" the value is accepted as a "value in s".
- With the time base "H:M:" the value is accepted as a "value in M (minutes)".



The delay times are the same as described for the constants.

Example:

Time base "s"

The operand has the value 9504.

The time value is 9.50 s.

Operand value 45 507

The time value is 45.510 s.

Time base "M:S"

The operand has the value 5999.

The time value is 99 min, 59 s: This is the maximum value.

Time base "H:M"

The operand has the value 5999.

The time value is 99 h, 59 min.

Displaying the parameter set in the PARAMETERS menu

- + Access enabled
- – Access disabled

Contacts

T 01Q1 to T 32Q1

Coils

- T 01EN to T 32EN: trigger coil;
- T 01RE to T 32RE: reset coil;
- T 01ST to T 32ST: stop coil.

Memory requirement of the timing relay

The time relay function block requires 48 bytes of memory plus 4 bytes per constant on the function block input.

Retention

Timing relays can be operated with retentive actual values. The number of retentive timing relays can be selected in the SYSTEM → RETENTION menu.

If a timing relay is retentive, the actual value is retained when the operating mode changes from RUN to STOP as well as when the power supply is switched off.

If easy is started in RUN mode, the timing relay operates with the retentively saved actual value. The state of the trigger impulse must correspond to the function of the timing relay.

1 signal when:

- on-delayed,
- single pulse,
- flashing.

0 status for offdelayed.

Function of the timing relay function block**Timing relay, on-delayed with and without random switching**

Random switching

The contact of the timing relay switches randomly within the setpoint value range.

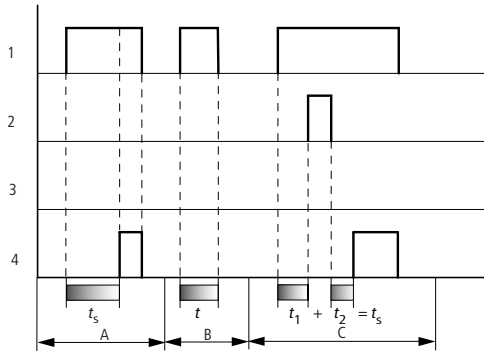


Figure 105: Signal diagram of timing relay, on-delayed
(with and without random switching)

1: trigger coil T..EN

2: stop coil T..ST

3: reset coil T..RE

4: switching contact (make contact) T..Q1

t_s : setpoint time

- Range A:
The set time elapses normally.
- Range B:
The entered setpoint does not elapse normally because the trigger coil drops out prematurely.
- Range C:
The Stop coil stops the time from elapsing.

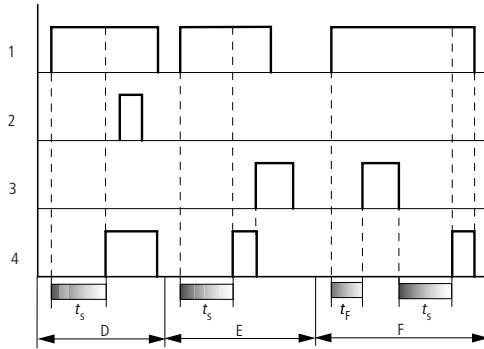


Figure 106: Signal diagram of timing relay, on-delayed (with and without random switching)

- Range D:
The Stop coil is inoperative after the time has elapsed.
- Range E:
The Reset coil resets the relay and the contact.
- Range F:
The Reset coil resets the time during the timeout. After the Reset coil drops out, the time elapses normally.

Timing relay, off-delayed with and without random switching

Random switching, with and without retriggering
The contact of the timing relay randomly switches within the set value range.

Retriggering

When the time is running and the trigger coil is reactivated or deactivated, the actual value is reset to zero. The set time of the timing relay is timed out once more.

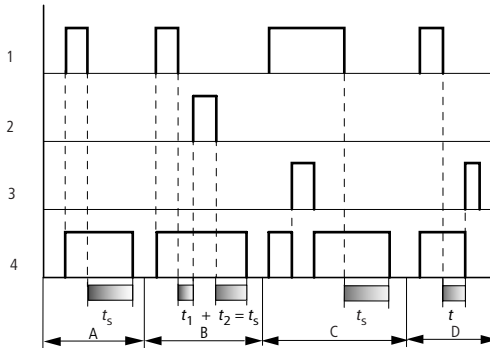


Figure 107: Signal diagram of offdelayed timing relay
(with/without random switching, with/without retriggering)

1: trigger coil T..EN

2: stop coil T..ST

3: reset coil T..RE

4: switching contact (make contact) T..Q1

t_s : setpoint time

- Range A:
The time elapses after the trigger coil is deactivated.
- Range B:
The Stop coil stops the time from elapsing.
- Range C:
The Reset coil resets the relay and the contact. After the Reset coil drops out, the relay continues to work normally.
- Range D:
The Reset coil resets the relay and the contact when the function block is timing out.

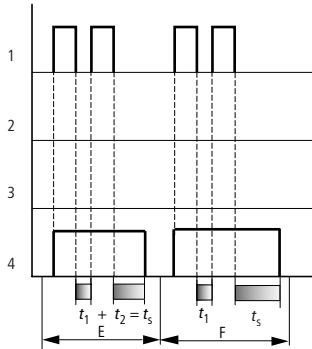


Figure 108: Signal diagram of offdelayed timing relay (with/without random switching, with/without retriggering)

- Range E:
The Trigger coil drops out twice. The set time t_s consists of t_1 plus t_2 (switch function not retriggerable).
- Range F:
The Trigger coil drops out twice. The actual time t_1 is cleared and the set time t_s elapses completely (retriggerable switch function).

Timing relay, on-delayed and off-delayed with and without random switching

Time value >I1: On-delayed time

Time value >I2: Off-delayed time

Random switching

The contact of the timing relay switches randomly within the setpoint value range.

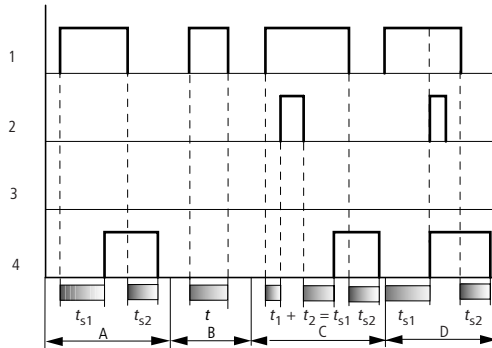


Figure 109: Signal diagram of timing relay, on and off-delayed 1

1: trigger coil T..EN

2: stop coil T..ST

3: reset coil T..RE

4: switching contact (make contact) T..Q1

t_{s1} : pick-up time

t_{s2} : drop-out time

- Range A:
The relay processes the two times without any interruption.
- Range B:
The trigger coil drops out before the on-delay is reached.
- Range C:
The Stop coil stops the timeout of the on-delay.
- Range D:
The stop coil has no effect in this range.

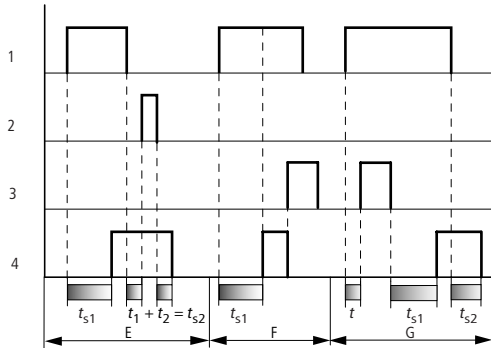


Figure 110: Signal diagram of timing relay, on and off-delayed 2

- Range E:
The Stop coil stops the timeout of the off-delay.
- Range F:
The Reset coil resets the relay after the on delay has elapsed
- Range G:
The Reset coil resets the relay and the contact whilst the on delay is timing out. After the Reset coil drops out, the time elapses normally.

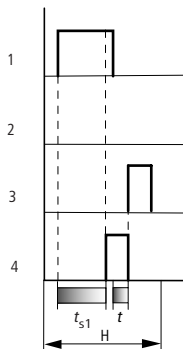


Figure 111: Signal diagram of timing relay, on and off-delayed 3

- Range H:
The Reset signal interrupts the timing out of the set time.

Timing relay, single pulse

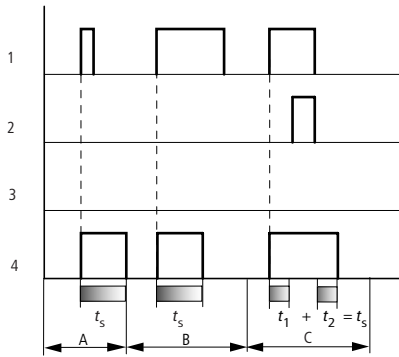


Figure 112: Signal diagram of timing relay, single pulse 1

1: trigger coil T..EN

2: stop coil T..ST

3: reset coil T..RE

4: switching contact (make contact) T..Q1

- Range A:
The trigger signal is short and is lengthened
- Range B:
The Trigger signal is longer than the set time.
- Range C:
The Stop coil interrupts the timing out of the set time.

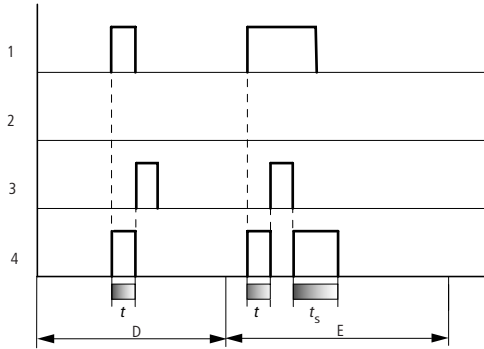


Figure 113: Signal diagram of timing relay, single pulse 2

- Range D:
The Reset coil resets the timing relay.
- Range E:
The Reset coil resets the timing relay. The Trigger coil is still activated after the Reset coil has been deactivated and the time is still running.

Timing relay, synchronous and asynchronous flashing

Time value >I1: Pulse time

Time value >I2: Pause time

Synchronous (symmetrical) flashing: >I1 equal >I2

Asynchronous flashing: >I1 not equal >I2

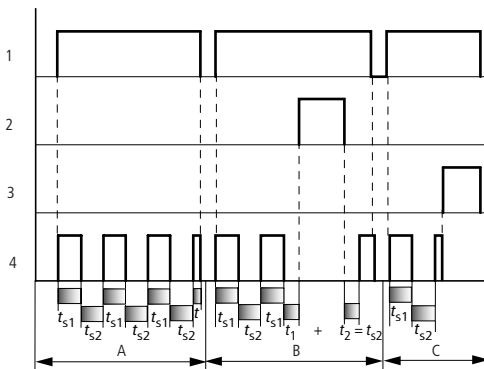


Figure 114: Signal diagram of timing relay, synchronous and asynchronous flashing

1: trigger coil T..EN

- 2: stop coil T..ST
- 3: reset coil T..RE
- 4: switching contact (make contact) T..Q1
- Range A:
The relay flashes for as long as the Trigger coil is activated.
- Range B:
The Stop coil interrupts the timing out of the set time.
- Range C:
The Reset coil resets the relay.

Value limitation

This function block is available from device version 04.

easy800 provides 32 value limitation function blocks VC01 to VC32. The value limitation function block allows you to limit values. You can define an upper and lower limit value. The function block will then only output values within these limits.

Wiring of a value limitation function block

You can integrate a value limitation function block into your circuit as a coil.

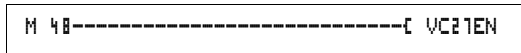
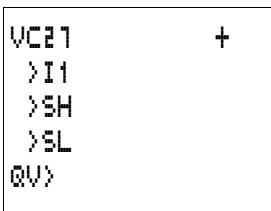


Figure 115: easy800 circuit diagram with VC value limitation



Parameter display and parameter set for the VC function block:

VC27	VC value limitation function block number 27
+	Appears in the parameter display
>I1	Input value
>SH	Upper limit value
>SL	Lower limit value
@V>	Output value limited

Inputs

The function block inputs >I1, >SH and >SL can be assigned the following operands:

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

Output

The function block output QV> can be assigned with the following operands:

- Markers MD, MW, MB
- Analog output QA01

Value range for inputs and outputs

		Value range
>I1	Input value	-2 147 483 648 to +2 147 483 647
>SH	Upper limit value	
>SL	Lower limit value	
QV>	Output value	

Displaying the parameter set in the PARAMETERS menu

- + Access enabled
- – Access disabled

Coil

VC01EN to VC32EN, function block enable

Memory requirement of the value limitation function block

The value limitation function block requires 40 bytes of memory plus 4 bytes per constant on the function block input.

Function of the value limitation function block



The function block must be enabled so that it can work. Coil VC..EN is active. If coil VC..EN is not active, the entire function block is deactivated and reset. The output value is set to zero.

The value is accepted at input VC...I1 if the enable coil is active. If the value is greater than the upper limit value or less than the lower limit value, the respective limit value is output at VC..QV.

Example with timing relay and counter function block

A warning light flashes when the counter reaches 10. In the example, both function blocks C 01 and T 01 are wired.

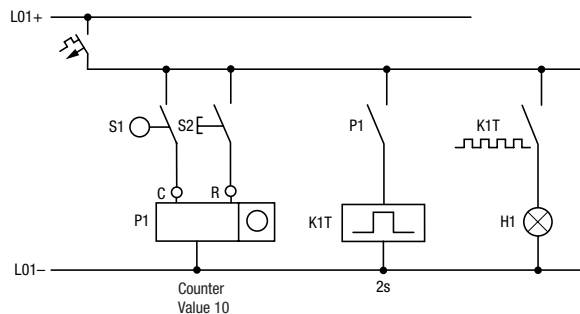


Figure 116: Hardwiring with relays

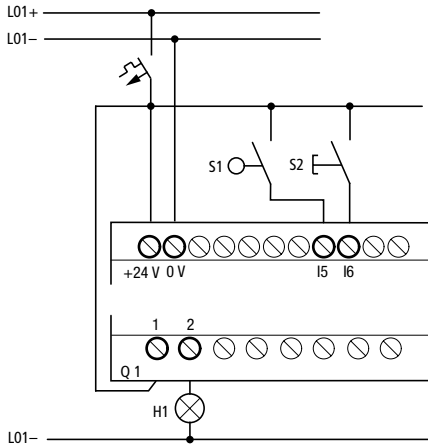


Figure 117: Wiring with easy...-DC-R...

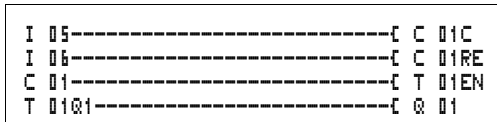


Figure 118: easy800 wiring and circuit diagram

Entering function block parameters from the circuit diagram.

You can access the parameter input from the contact as well as from a coil.

- ▶ Enter the circuit diagram up to **C 01** as a coil.

C 01C is the counter coil of the counter 01 function block.

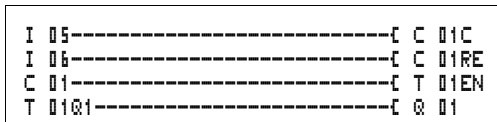


Figure 119: easy800 wiring and circuit diagram

- ▶ Keep the cursor on the number.
- ▶ Press the **OK** button.



If the cursor is on the contact number, easy800 will call up the parameter display when you press **OK**.

```
C 01      +
  >SH +10
  >SL
  >SV
```

The first part of the parameter set of a counter is displayed.

- ▶ Proceed with the cursor > over the + character to the value input behind >SH:
 - >SH means: function block input upper counter setpoint value
 - The + character means that the parameters of this timing relay can be modified using the PARAMETERS menu.
- ▶ Change the upper counter setpoint to 10:
 - Use < > to move the cursor onto the tens digit.
 - Use ^ v to modify the value of the digit.
- ▶ Press **OK** to save the value and **ESC** to return to the circuit diagram.



easy800 has specific parameter displays for the function blocks. The meaning of these parameters is explained under each function block type.

- ▶ Enter the circuit diagram up to contact **T 01** of the timing relay. Set the parameters for **T 01**.

```
T 01  ⚡  S  +
  >I1 002.000
  >I2 002.000
  QV>
```

The timing relay works like a flashing relay. The easy800 symbol for the flashing relay is ⚡. The function is set on the top right beside the number in the parameter display.

The time base is set to the right of the “flashing” function. Leave the time base set to **S** for seconds.

- ▶ Proceed with the cursor to the right over the + character in order to input the time setpoint value >I1.

If the same setpoint value is input for >I1 and >I2, the timing relay functions as a synchronous flasher.

The + character means that the parameters of this timing relay can be modified using the PARAMETERS menu.

- ▶ Confirm the value input with **OK**.
- ▶ Press **ESC** to leave circuit diagram entry.
- ▶ Complete the circuit diagram.
- ▶ Test the circuit diagram using the power flow display.
- ▶ Switch easy800 to RUN mode and return to the circuit diagram.

Each parameter set can be displayed using the power flow display for the circuit diagram.

- ▶ Move the cursor onto **C 01** and press **OK**.

```
C 01      +
>SL
>SV
QV>+0
```

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

- ▶ Move the cursor \vee downwards until you see the value QV>.

```
C 01      +
>SL
>SV
QV>+1
.. C_ . . . .
```

- ▶ Switch the input I5. The actual value changes.

On the display **C_** indicates that the counting coil is actuated.

If the actual and upper setpoint values of the counter are the same, the timing relay switches the warning light on and off every 2 seconds.

```
T 01 u S  +
>I1 001.000
>I2
QV> 0.550
.. EN..
```

Doubling the flashing frequency:

- ▶ Select the power flow display **T 01** and change the constant of the setpoint time to **001 . 000**.

When you press **OK**, the warning light will flash at twice the frequency.

On the display **EN** indicates that the enable coil is actuated.

Setpoint value settings with constants can be modified via the PARAMETERS menu.



The actual value is only displayed in RUN mode. Call up the parameter display for this via the power flow display or PARAMETERS menu.

5 easy-NET network

Introduction to easy-NET

All easy800 units have an easy-NET network interface connection. This network is designed for eight stations.

Using the easy-NET you can:

- Process additional inputs and outputs.
- Implement faster and improved control using decentralised programs.
- Synchronise date and time
- Read and write inputs and outputs.
- Send values to other stations.
- Receive values from other stations.
- Load programs from or to other stations.

The easy-NET network is based on the CAN network (Controller Area Network). CAN is specified by the ISO 11898 standard. CAN has the following in-built features:

- Message oriented transmission protocol.
- Multimaster bus access capabilities with non-destructive bitwise bus arbitration via priority messaging (Arbitration: An instance which defines which hardware can use the bus next).
- Multicast broadcast messaging with receiver side message filtering.
- High level of real-time capability (short reaction time for high priority messages, short fault message get times).
- Functionality in environments with severe interference (short block lengths).
- High level of error security.



CAN has been used as the basis for the design of the easy-NET network. The messages have been adapted and optimised to suit the requirements of the easy800 environment.

easy-NET network topologies, addressing and functions

The easy-NET allows the configuration of a line topology. There are two wiring methods which can be used for the required addressing options:

- “Loop through the unit” wiring arrangement,
- Wiring arrangement using a T connector and a spur line.

Loop through the unit wiring method

With this wiring method it is possible to implement the addressing of the stations via station 1 or the EASY-SOFT(-PRO). If the line is interrupted, the network is no longer operational from this point in the network.

T connector and spur line

Each device must be addressed individually with this wiring method by:

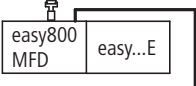
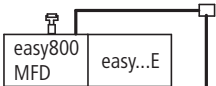
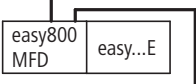
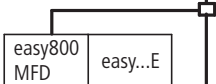
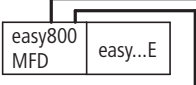
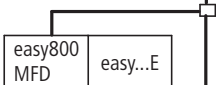

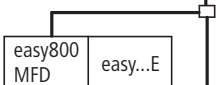

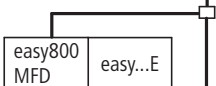

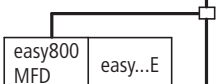

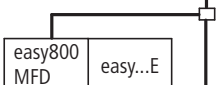
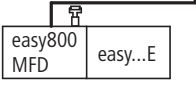
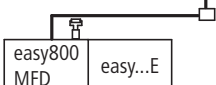
- Downloading the program,
- Downloading the address with EASY-SOFT (-PRO),
- Using the display or
- The device is already assigned with an address.

If a spur line is removed on a station, all other devices in the network remain functional.



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

Topology and addressing examples

Physical location, place	Station number		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 (with the exception that location 1 is the same as station 1).



Physical location 1 is always assigned as station 1. Station 1 is the only station which must be present.

Position and addressing of the operands via easy-NET

Stations	Basic unit		Local expansion		Network bit data		Network word data	
	Input I	Output Q	Input R	Output S	Input RN	Output SN	Receive	Send
1	1 I 1 to 16	1 Q 1 to 8	1 R 1 to 16	1 S 1 to 8	2 to 8 RN 1 to 32	2 to 8 SN 1 to 32	GT 1 to 32	PT 1 to 32
2	2 I 1 to 16	2 Q 1 to 8	2 R 1 to 16	2 S 1 to 8	1, 3 to 8 RN 1 to 32	1, 3 to 8 SN 1 to 32	GT 1 to 32	PT 1 to 32
3	3 I 1 to 16	3 Q 1 to 8	3 R 1 to 16	3 S 1 to 8	1, 2, 4 to 8 RN 1 to 32	1, 2, 4 to 8 SN 1 to 32	GT 1 to 32	PT 1 to 32
4	4 I 1 to 16	4 Q 1 to 8	4 R 1 to 16	4 S 1 to 8	1 to 3, 5 to 8 RN 1 to 32	1 to 3, 5 to 8 SN 1 to 32	GT 1 to 32	PT 1 to 32
5	5 I 1 to 16	5 Q 1 to 8	5 R 1 to 16	5 S 1 to 8	1 to 4, 6 to 8 RN 1 to 32	1 to 4, 6 to 8 SN 1 to 32	GT 1 to 32	PT 1 to 32
6	6 I 1 to 16	6 Q 1 to 8	6 R 1 to 16	6 S 1 to 8	1 to 5, 7, 8 RN 1 to 32	1 to 5, 7, 8 SN 1 to 32	GT 1 to 32	PT 1 to 32
7	7 I 1 to 16	7 Q 1 to 8	7 R 1 to 16	7 S 1 to 8	1 to 6, 8 RN 1 to 32	1 to 6, 8 SN 1 to 32	GT 1 to 32	PT 1 to 32
8	8 I 1 to 16	8 Q 1 to 8	8 R 1 to 16	8 S 1 to 8	1 to 7 RN 1 to 32	1 to 7 SN 1 to 32	GT 1 to 32	PT 1 to 32



The RN-SN connection is a point to point connection between the stations indicated. With RN and SN the number of the contact must have the same number as the coil. Example: 2SN30 from station 8 is sent to 8RN30 of station 2.



Every station with a circuit diagram has read access to the physical station inputs and outputs of other stations and can process them locally.

Example 1

Station 1 is to read the input I1 of station 2 and write to output Q1 of station 2. Station 2 does not have a circuit diagram.

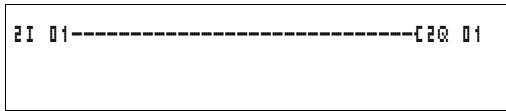


Figure 120: Circuit diagram in station 1

Example 2:

Marker M 01 of station 4 is to switch the output Q1 of station 3 via the network. Both stations have a circuit diagram.

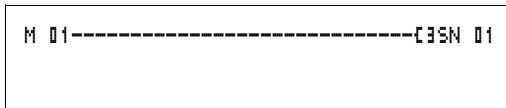


Figure 121: Circuit diagram in station 4: Set coil 01 in station 3

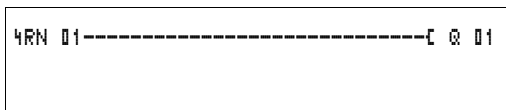


Figure 122: Circuit diagram in station 3: Get value from coil 01 in station 4

Functions of the stations in the network

The stations on the easy-NET can have two different functions:

- Intelligent stations with their own programs (stations 1 to 8)
- Input/output devices (REMOTE IO) without their own program (stations 2 to 8)



Station 1 must always have a circuit diagram.

Possible write and read authorisation in the network

The stations have differing read and write authorisation in the easy-NET network according to their functions and configuration.

Station 1

Authorised read access to all inputs and outputs of all stations regardless of the function. Observe the setting of SEND IO (→ Section "Send each change on the inputs/ outputs (SEND IO)", Page 255).

Authorised write access to the station's own local outputs.

Authorised write access to the physical digital outputs of the stations which are functioning as input/output devices.

Write authorisation to the network bit data 2 to 8 – SN 1 to 32.

Stations 2 to 8

Input/output device function

No read and write authorisation.

Intelligent station function

Authorised read access to all inputs and outputs of all stations regardless of the function. Observe the setting of SEND IO (→ Section "Send each change on the inputs/ outputs (SEND IO)", Page 255).

Write authorisation to its own local outputs.

Write authorisation to the network bit data SN 1 to 32.

Configuration of the easy-NET network

easy-NET can be configured so that it can be optimised for your application.

Station number

The station number is identified as the NET-ID: in the device. The station number can be set on devices with a display using the buttons on the easy800.



All the easy-NET settings are best carried out on station 1. The entire network can be configured via station 1. The configuration should only be carried out locally when a device is replaced.

Valid station numbers for operation are 01 to 08.

Station number 00 = factory default setting

With station number 00, double address assignment cannot occur when an existing device is being exchanged.

Transmission speed

The easy800 device hardware allows you set transmission speeds between 10 and 1000 KBaud in specific stages. In doing so, the length of all cables is linked to the maximum data transfer rate (→ chapter "Technical data", Page 330).

The data transfer rate is set under the BAUDRATE: menu item.

Possible baud rates are: 10, 20, 50, 125, 250, 500 and 1000 KB

125 KB = factory default setting

Pause time, changing the write repetition rate manually

Every easy-NET network connection automatically determines the number of stations which are active on the network, the baud rate which is used and the total number of bytes which are transmitted. The minimum pause time which a device requires is automatically determined using this data in order to ensure that all devices can send their messages. If a pause time is to be increased, the value of the BUSDELAY: must be set greater than zero.

Value "1" doubles the pause time, value "15" will increase it by a factor of 16.

$$t_{pnew} = t_p \times (1 + n)$$

t_{pnew} = new pause time

t_p = pause time determined by the network

n = value on BUSDELAY



An extension of the pause time means that fewer messages (inputs, outputs, bit data, word data) are transferred per time unit.

The reaction speed of the entire controller depends on the baud rate, the pause time and the quantity of transferred data.

The smaller the amount of data transferred, the faster the reaction times of the system.



An increase in the pause time is only useful during commissioning. To ensure that the data for the power flow display is updated faster in the PC, a longer range for this data is created on the network within this pause time.

Send each change on the inputs/outputs (SEND IO)

The SEND IO function should be used if you wish to send any change in input or output status immediately to all other network stations. SEND IO should be activated if intelligent stations have read access to the inputs and outputs of other stations (2I 02, 8Q 01, etc.).

SEND IO ✓

This means that the quantity of messages on the network can increase significantly.



If high-speed counters are used, the SEND IO function should be deactivated. Otherwise the input data is written very rapidly onto the network as they change continuously leading to unnecessary loading of the network.

If intelligent devices are required to exchange bit information, it should be implemented via RN and SN.

SEND IO ✓ = factory default setting

Automatic change of the RUN and STOP mode

REMOTE RUN should be activated if stations 2 to 8 are to automatically follow the mode change of station 1 during operation.



Input and output devices must always have SEND IO activated, to ensure that station 1 always receives up-to-date input and output data.



Intelligent stations with display only follow the operating mode change when the display is showing the Status display or a text.

The following is of utmost importance during commissioning!



Attention!

If several engineers are commissioning a machine or system involving several spatially separated elements via the easy-NET network, it must be ensured that REMOTE RUN is not activated.

Otherwise unwanted machine or system starts may occur during commissioning. The associated events depend on the machines or systems.

REMOTE RUN ✓ = factory default setting

Input/output device (REMOTE IO) configuration

All devices are factory set for operation as input and output devices. This has the advantage that devices can be used immediately as I/O devices, regardless of whether they have a display or not. You only need to assign the station number. This can be implemented via EASY-SOFT(-PRO) or on a Station 1 with a display.

If you want to assign a device as an intelligent station on the network, the REMOTE IO should be deactivated.

REMOTE IO

Figure 123: Remote IO deactivated

The standard settings for the input and output devices are:

SEND IO	✓
REMOTE RUN	✓
REMOTE IO	✓

Station number (easy-NET-ID) and baud rate can be specified via station 1.

Displaying the Status display of other stations

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

1I12.....7....	
I NT1	P-
MO 06:42	
1Q1.....	RUN

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

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

3I12.....7....	
I NT3	P-
MO 06:42	
3Q1.3..6..	RUN

► Change to the number of the required station with the \wedge and \vee cursor buttons.

► Press the **OK** button.

3R12.....7....	
I NT3 DC	P-
MO 06:45	
3S1.3..6..	RUN

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

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



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

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

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

Station message types

The easy-NET network recognises various message types. They are:

- Output data of station 1 (Q., S.) which is sent to stations without programs.
- Network outputs and inputs sent and received between stations with programs (*SN, *RN).
- Data sent and received via the network between stations with programs (PT and GT function blocks).
- Inputs, outputs, station status (I, R, Q, S) transfers.
- Loading programs to and from every station.

The easy-NET network is based on a CAN (Controller Area Network) system. Each message type has its own ID. The message priority is determined via the respective ID. This is important in transmission borderline cases to ensure that all messages reach their destination.

Transfer behaviour

Network CPU data transfer to program image

The easy800 network connection is equipped with its own CPU. Network data can therefore be processed whilst the program is running. After each program cycle, the status of the network data is written to the operand image of the program and the send data is read from the image. The program runs through the next cycle with this data.

Reading and sending the network data from the CPU

The network CPU of the station reads every message on the network. If the message is relevant to the station, it is accepted into a message memory.

If the content of the sent message changes, this is sent. Transmission only occurs when there is no message on the network.

easy-NET is configured so that every station can send its messages. This means that the station must observe a pause time between sending messages. The pause time increases the higher the number of stations and the lower baud rate setting.

The number of stations is recognised by every station via a "sign of life" signal.



The following applies to fast message transfer:

- Set the fastest possible baud rate to suit the network length and cable cross-section.
- Fewer messages means faster messages.
- Avoid program downloads during the RUN mode.

Signs of life of the individual stations and diagnostics

The inputs and outputs message type is used as a "sign of life" recognition to ensure that the state of the station can be recognised by other stations. The states of the inputs and outputs are sent cyclically and at the set baud rate, irrespective of the SEND IO setting. If the inputs and outputs of a station are not recognised by other stations after a time determined by the baud rate, the station is deemed to be disconnected until the next "sign of life" is recognised.

Evaluation occurs at the following intervals:

Baud rate	Stations must send a "sign of life" every ...	Stations recognise the absence of a "sign of life" signal after
[KB]	[ms]	[ms]
1000	60	180
500	60	180
250	120	360
125	240	720
50	600	1800
20	1500	4500
10	3000	9000

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

Diagnostics contact	Station number
ID 01	1
ID 02	2
ID 03	3
ID 04	4
ID 05	5
ID 06	6
ID 07	7
ID 08	8



If a station does not send a "sign of life" signal (station not available, NET interrupted), the respective diagnostics contact ID .. is activated.



Attention!

If the states of the inputs, outputs or data are required by a station without fail, the respective diagnostics contact should be evaluated and the information applied in accordance with its respective application.

If the respective diagnostics contacts are not evaluated, it may cause faults in your application.



The data to be read from a faulty station is set to the 0 state after the fault is detected.

Network transmission security

easy-NET is a CAN-based network. CAN is used in cars and commercial vehicles in all areas. The same fault recognition capability with data transfer applies as with CAN. A BOSCH study relating to undiscovered and corrupt messages determined the following:

The probability of non-discovery of a corrupted message (residual error probability) is: $< 10^{-10}$ message error rate.

The message error rate depends on:

- Bus loading
- Telegram length
- Malfunction frequency
- Number of stations

Example:

Network with:

- 500 KBaud
- average bus load 25 %
- average operating time 2000 h/year
- average error rate of 10^{-3} ,
i.e.: 1 message is faulty every 1000
- transmission of 1.12×10^{10} messages per year of which
 1.12×10^7 messages are faulty
- residual error probability: $r < 10^{-10} \times 10^{-3} = 10^{-13}$

This means: one of 10^{13} messages is so corrupt that the fault cannot be recognised as such. For a network, this corresponds to a working time of approx. 1000 years.

6 easy Settings

Settings can only be carried out on easy models provided with buttons and LCD display.

EASY-SOFT -PRO can be used to set all models via the software.

Password protection

The easy can be protected by a password against unauthorised access.

In this case the password consists of a value between 000001 and 999999. The number combination 000000 is used to delete a password.

Password protection inhibits access to selected areas. The System menu is always protected by an activated password.

The password can protect the following inputs and areas:

- Start or modification of the program
- Transfer of a circuit diagram to or from a memory card (Display variants).
- Change of the RUN or STOP mode.
- Calling and modification of function block parameters
- All settings of the real-time clock.
- Modifications of all system parameters.
- Communication with the individual device. (looping to other devices possible.)
- Switching off the password delete function.



A password that has been entered in easy is transferred to the memory card together with the circuit diagram, irrespective of whether it was activated or not.

If this easy circuit diagram is loaded back from the memory card, the password will also be transferred to easy and is activated immediately.

Password setup

A password can be set via the System menu regardless of the RUN or STOP modes. You cannot change to the System menu if a password is already activated.

- ▶ Press **DEL** and **ALT** to call up the System menu.
- ▶ Select the menu option SECURITY... to enter the password.
- ▶ Press the **OK** button and change over to the PASSWORD... menu.
- ▶ If you press the **OK** button again, you will access the password input.



If no password has been entered, easy changes directly to the password display and displays six dashes: No password available.

- ▶ Press **OK**, six zeros will appear
- ▶ Set the password using the cursor buttons:
 - < > select position in password,
 - ^ v set a value between 0 to 9.



- ▶ Save the new password by pressing **OK**.

Use **OK** to exit the password display and proceed with **ESC** and v to the RANGE... menu.

The scope of the password has not yet been considered. The password is now valid but not yet activated.

Selecting the scope of the password

```

CIRCUIT
DIAGRAM  /+
PARAMETERS
TIME
OPERATING
MODE    +

```

- ▶ Press the **OK** button.
- ▶ Select the function to be protected or the menu.
- ▶ Press the **OK** button in order to protect the function or menu (tick = protected).



Standard protection encompasses the programs and circuit diagram.

At least one function or menu must be protected.

- **CIRCUIT DIAGRAM:** The password is effective on the program with circuit diagram and non-enabled function blocks.
- **PARAMETERS:** The PARAMETERS menu is protected.
- **TIME:** Date and time are protected with the password.
- **OPERATING MODE:** The toggling of the RUN or STOP operating mode is protected.
- **INTERFACE:** The interface is inhibited for access to a connected device. Programs or commands to other devices connected via the NET network are routed further.
- **DELETE FUNCTION:** After four failed attempts to enter a password, the "DELETE FUNCTION?" prompt appears. This prompt is not displayed if selected. However, it is no longer possible to make changes in protected areas if you forget the password.

Activating the password

You can activate an existing password in three different ways:

- automatically when easy is switched on again,
 - automatically after a protected circuit diagram is loaded
 - automatically if a telegram has not been sent on the PC interface 30 minutes after unlocking the device (using EASY-SOFT (-PRO),
 - via the password menu
- ▶ Press **DEL** and **ALT** to call up the System menu.
- ▶ Open the password menu via the menu option SECURITY...



easy will only show this password menu if a password is present.



Make a note of the password before you activate it. If the password is no longer known, easy can be unlocked (DELETE FUNCTION is not active), however, the circuit diagram and data settings will be lost.



Attention!

If the password is unknown or lost, and the delete password function is deactivated: The unit can only be reset to the factory setting by the manufacturer. The program and all data are lost.

- ▶ Select **ACTIVATE PW** and confirm with **OK**.

The password is now active. easy will automatically return to the Status display.

You must unlock easy with the password before you implement a protected function, enter a protected menu or the System menu.

Unlocking easy

Unlocking easy will deactivate the password. You can reactivate password protection later via the password menu or by switching the power supply off and on again.

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

The PASSWORD... entry will flash.

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

```
PASSWORD...
STOP RUN ✓
PASSWORD...
SET CLOCK
```



If easy shows PROGRAM... in the main menu instead of PASSWORD..., this means that there is no password protection active.

```
ENTER PASSWORD
XXXXXX
```

easy will display the password entry field.

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

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

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

The PROGRAM... menu option is now accessible so that you can edit your circuit diagram.

The System menu is also accessible.

Changing or deleting the password range

- ▶ Unlock easy.
- ▶ Press **DEL** and **ALT** to call up the System menu.
- ▶ Open the password menu via the menu option SECURITY... and PASSWORD....



The CHANGE PW entry will flash.

easy will only show this password menu if a password is present.



- ▶ Press **OK** to enter the password entry menu.
- ▶ Use **OK** to move to the 6-digit entry field.
- ▶ The current password will be displayed.



- ▶ Modify the six password digits using the cursor buttons.
- ▶ Confirm with **OK**.

Use **ESC** to exit the security area.



Deleting

Use number combination 000000 to delete a password.

If a password has not been entered already, easy will show six dashes:

Password incorrect or no longer known

If you no longer know the exact password, you can try to re-enter the password several times.



DELETE FUNCTION has not been deactivated.

ENTER PASSWORD
XXXXXX

You have entered an incorrect password?

► Re-enter the password.

DELETE ALL?

After the fourth entry attempt easy will ask whether you wish to delete the circuit diagram and data.

► Press

- **ESC**: No data will be deleted.
- **OK**: Circuit diagram, data and password are deleted.

easy will return to the Status display.



If you no longer know the exact password, you can press **OK** to unlock the protected easy. The saved circuit diagram and all function relay parameters will be lost.

Pressing **ESC** will retain the circuit diagram and data. You can then make another four attempts to enter the password.

Changing the menu language

easy800 provides ten menu languages which are set as required via the System menu.

Language	Display
English	ENGLISH
Deutsch	DEUTSCH
French	FRANCAIS
Spanish	ESPANOL
Italian	ITALIANO
Portuguese	PORTUGUES
Dutch	NEDERLANDS
Swedish	SVENSKA
Polish	POLSKI
Turkish	TURKCE



Language selection is only possible if easy is not password-protected.

- ▶ Press **DEL** and **ALT** to call up the System menu.
- ▶ Select MENU LANGUAGE... to change the menu language.

ENGLISH	⊕
DEUTSCH	✓
FRANCAIS	
ESPANOL	⊕
ITALIANO	
PORTUGUES	
NEDERLAND	
SVENSKA	
POLSKI	
TURKCE	

The language selection for the first entry ENGLISH is displayed.

- ▶ Use **^** or **v** to select the new menu language, e.g. Italian (ITALIANO).
- ▶ Confirm with **OK**. ITALIANO is assigned with a tick.
- ▶ Exit the menu with **ESC**.

```
SICUREZZA...
SISTEMA...
LINGUA MENU...
CONFIGURATORE...
```

easy will now show the new menu language.

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

Changing parameters

easy allows you to change function relay parameters such as timing relay setpoint values and counter setpoints without having to call up the circuit diagram. This is possible regardless of whether easy is running a program or is in STOP mode.

- ▶ Press **OK** to switch to the main menu.
- ▶ Start the parameter display by selecting **PARAMETERS**.

```
T 03 W S +
CP08      -
C 11      +
L: 1      RUN
```

All function blocks are displayed as a list.

The following preconditions must be fulfilled in order for a parameter set to be displayed:

- A function relay must have been included in the circuit diagram.
- The **PARAMETERS** menu must be available.
- The parameter set must have been enabled for access, indicated by the **+** character at the bottom right of the display.



Parameter sets can only be enabled or protected via the **FUNCTION RELAYS** menu, or via the circuit diagram with the **“+”** enable and with **“-”** inhibit parameter set characters.

```

T 03 U S +
>I1 020.030
>I3 005.000
QV> 012.050

```

- ▶ Select the required function block with ^ or v .
- ▶ Press the **OK** button.
- ▶ Scroll with the ^ or v cursor buttons through the constants of the function block inputs.
- ▶ Change the values for a parameter set:
 - With **OK** in the Entry mode.
 - < > change decimal place
 - ^ v change the value of a decimal place
 - **OK** save constants or
 - **ESC** Retain previous setting.

Press **ESC** to leave the parameter display.



Only constants on the function block inputs can be changed.

Adjustable parameters for function blocks

You can modify the function blocks used in the circuit diagram in three different ways:

- All circuit diagram parameters can be adjusted in STOP mode via the function block editor.
- Setpoints (constants) can be modified in RUN mode via the function block editor.
- Setpoints (constants) can be modified via the PARAMETERS menu.

Adjustable setpoint values are:

- The inputs with all function blocks if constants have been used.
- Switch on and off times with time switches.

In RUN mode easy operates with a new setpoint as soon as it has been modified in the parameter display and saved with **OK**.

Setting date, time and daylight saving time

The easy800 devices are equipped with a real-time clock with date and time functions. The “time switch” function block can be used to implement time switch applications.

If the clock has not yet been set or if easy is switched on after the buffer time has elapsed, the clock starts with the setting “WE 1:00 01.05.2002”. The easy clock operates with date and time so the hour, minute, day, month and year must all be set.



The time, such as: 1:00, indicates the version of the device operating system.

```
SET CLOCK
DST SETTING
```

```
HH:MM  00:27
DD.MM  05.05
YEAR :  2002
```

- ▶ Select SET CLOCK... from the main menu.

This will open the menu for setting the time.

- ▶ Select SET CLOCK.

- ▶ Set the values for day, time, month and year.
- ▶ Press the **OK** button to access the Entry mode.
 - < > Move between the parameters
 - ^ v Change the value.
 - **OK** Save day and time
 - **ESC** Retain previous setting.

Press **ESC** to leave the time setting display.

Changing between winter/summer time (DST)

The easy800 models are fitted with a real-time clock. The clock has various possibilities for changing the DST setting. These are subject to different legal requirements in the EU, GB and USA.



The time change algorithm only applies to the northern hemisphere.

- NONE: no daylight saving time setting
- MANUAL: a user defined date for the DST change
- EU: date defined by the European Union; Commences: last Sunday in March; Ends: last Sunday in October
- GB: date defined in Great Britain; Commences: last Sunday in March; Ends: fourth Sunday in October
- US: date defined in the United States of America: Commences: first Sunday in April; Ends: last Sunday in October

The following applies to all DST variants:

Winter time → Summer time: On the day of conversion, the clock moves forward one hour at 2:00 to 3:00

Summer time → Winter time: On the day of conversion, the clock moves back one hour at 3:00 to 2:00.

Select SET CLOCK... from the main menu.

This will open the menu for setting the time.

▶ Select the DST SETTING menu option.



Selecting DST

easy shows you the options for the DST change.

The standard setting is NONE for automatic DST changeover (Tick at NONE).

```

NONE      /  +
MANUAL
EU
GB        +
US
  
```

▶ Select the required variant and press the **OK** button.

```

SUMMERTIME START
DD.MM  00.00
SUMMER TIME END
DD.MM: 00:00
  
```

Manual selection

You want to enter your own date.



The following applies to easy800 devices:

The time change algorithm always calculates the date from the year 2000. Enter the time change for the year 2000.

- ▶ Proceed to the MANUAL menu and press 2 × **OK**.
 - < > Move between the parameters
 - ^ v Change the value.
 - **OK** Save day and time
 - **ESC** Retain previous setting.
- ▶ Press **ESC** to leave the display.
- ▶ Select the day and time at which Summer time is to commence.
- ▶ Select the day and time at which Summer time is to end.



The same time for conversion applies as for the legally determined variants (EU, GB, US).

Activating input delay (debounce)

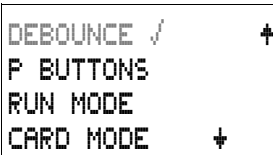
Input signals can be evaluated by easy with a debounce delay. This enables, for example, the trouble-free evaluation of switches and push-button actuators subject to contact bounce.

In many applications, however, very short input signals have to be monitored. In this case, the debounce function can be switched off.

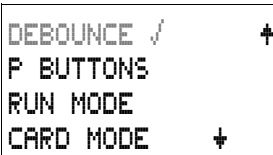
- ▶ Press **DEL** and **ALT** to call up the System menu.
- ▶ Select the **SYSTEM** menu.



If easy is password-protected you cannot open the System menu until you have "unlocked" it.



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



Activating debounce

If a tick ✓ is set beside **DEBOUNCE**, the input delay is set.

If this is not so, proceed as follows:

- ▶ Select **DEBOUNCE** and press **OK**.

Debounce mode will be activated and the display will show **DEBOUNCE /**.

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

Deactivating debounce (input delay)

If easy is showing **DEBOUNCE** in the display, this means that Debounce mode has already been deactivated.

- ▶ Otherwise select **DEBOUNCE /** and press **OK**.

Debounce mode will be deactivated and the display will show **DEBOUNCE**.



How easy input and output signals are processed internally is explained in Section "Delay times for inputs and outputs", from Page 292.

Activating and deactivating the P buttons

Even though the cursor buttons (P buttons) have been set as push-button actuator inputs in the circuit diagram, this function is not activated automatically. This prevents any unauthorised use of the cursor buttons. The P buttons can be activated in the System menu.



If easy is password-protected you cannot open the System menu until you have "unlocked" it.

The P buttons are activated and deactivated via the P BUTTONS menu.

```
DEBOUNCE ✓ +
P BUTTONS
RUN MODE
CARD MODE +
```

- ▶ Press **DEL** and **ALT** to call up the System menu.
- ▶ Select the SYSTEM menu.
- ▶ Move to the cursor to the P BUTTONS menu.

```
DEBOUNCE ✓ +
P BUTTONS
RUN MODE
CARD MODE +
```

Activating the P buttons

If easy is displaying **P BUTTONS** ✓, this means that the P buttons are active.


- ▶ Otherwise select P BUTTONS and press **OK**.
easy changes the display to **P BUTTONS** ✓ and the P buttons are activated.

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

```
DEBOUNCE ✓ +
P BUTTONS ✓
RUN MODE
CARD MODE +
```

The P buttons are only active in the Status display and the text display. In this display you can use the P buttons to activate inputs in your circuit diagram.

Deactivating the P buttons

► Select **P BUTTONS**  and press **OK**.

easy changes the display to **P BUTTONS** and the P buttons are deactivated.



The P buttons are automatically deactivated when loading a circuit diagram from the memory card or via EASY-SOFT (-PRO) to easy, or when deleting a circuit diagram in easy.

Startup behaviour

The startup behaviour is an important aid during the commissioning phase. The circuit diagram which easy contains is not as yet fully wired up or the system or machine is in a state which easy is not permitted to control. The outputs should not be controlled when easy is switched on.

Setting the startup behaviour



The easy devices without a display can only be started in RUN mode.

Requirement: easy must contain a valid circuit diagram.


► Switch to the System menu.



If easy is protected by a password, the System menu will not be available until easy is "unlocked" (see → Section "Unlocking easy" from Page 267).

Specify the operating mode which easy must use when the supply voltage is applied.

Activating RUN mode

If easy displays **RUN MODE** , this means that easy will start in Run mode when the supply voltage is applied.

```

DEBOUNCE / +
P BUTTONS
RUN MODE /
CARD MODE +

```

► Otherwise select RUN MODE and press **OK**.
RUN mode is activated.

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

```

DEBOUNCE / +
P BUTTONS
RUN MODE
CARD MODE +

```

Deactivating RUN mode

► Select RUN MODE / and press **OK**.

The RUN mode function is deactivated.

The default setting for easy is for RUN MODE to be displayed. In other words, easy starts in RUN MODE / when the power is switched on.

Startup behaviour	Menu display	Status of easy after startup
easy starts in STOP mode	RUN MODE	easy is in STOP mode
easy starts in RUN mode	RUN MODE /	easy is in RUN mode

Behaviour when the circuit diagram is deleted

The setting for the startup behaviour is an easy device function. When the circuit diagram is deleted, this does not result in the loss of the setting selected.

Behaviour during upload/download to card or PC

When a valid circuit diagram is transferred from easy to a memory card or the PC or vice versa, the setting is still retained.



The easy devices without a display can only be started in RUN mode.

Possible faults

easy will not start in RUN mode:

- a program is not available in easy.
- you have selected easy startup in STOP MODE (RUN MODE menu).

Card startup behaviour

The startup behaviour with memory card is for applications where unskilled personnel change the memory card under no-voltage conditions.

easy only starts in the Run mode if a memory card with a valid program is inserted.

If the program on the memory card is different to the program in easy, the program from the card is loaded into easy and easy starts in RUN mode.

► Switch to the System menu.



If easy is protected by a password, the System menu will not be available until easy is "unlocked" (see → Section "Unlocking easy" from Page 267).

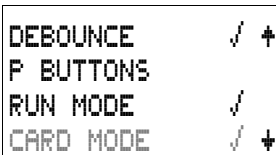
Activation of card mode

Requirement: RUN MODE is active.

If easy displays **CARD MODE** ✓, this means that when the power supply is switched on, easy will only start in RUN mode if a memory card with a valid program has been inserted.

► Otherwise select **CARD MODE** and press **OK**. easy will start up with the program on the card.

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



```

DEBOUNCE / +
P BUTTONS
RUN MODE /
CARD MODE +

```

Deactivation of card mode

► Select **CARD MODE** / and press **OK**.

The RUN mode function is deactivated.

The easy default setting is for display of the CARD MODE menu, i.e. easy starts in RUN mode without the memory card when the power is switched on.

Setting LCD contrast and backlight

The background illumination of the LCD display can be switched off. The display contrast can be set to one of five 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.

If the backlight is switched off, it will switch back on when a button is pressed. The backlight switches off automatically 60 s after the last button is pressed.

The contrast and backlight settings are implemented as device settings.

► Switch to the System menu.



If easy is protected by a password, the System menu will not be available until easy is "unlocked" (see → Section "Unlocking easy" from Page 267).

```

SECURITY +
SYSTEM...
MENU LANGUAGE
CONFIGURATOR... +

```

► Select the **SYSTEM** menu.

► Press the **OK** button.

```

P BUTTONS +
RUN MODE
CARD MODE
DISPLAY... +

```

► Use the ∨ button to select the **DISPLAY** menu and press **OK**.



The menus for setting the contrast and backlight are displayed.

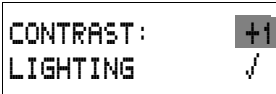
▶ Press the **OK** button and move to the contrast entry field



Use the ^ and v cursor buttons to set the contrast to a value between -2 and +2.

▶ Enter your setting

▶ Complete your setting by pressing **OK**.



The contrast setting will be retained until it is modified.



▶ Use the cursor buttons ^ and v to move to the LIGHTING menu.

▶ Press the **OK** button.



▶ The backlight is deactivated.



▶ Press **OK** if you wish to reactivate the backlight

▶ The tick ✓ indicates that the backlight has been activated.



easy comes with the following factory setting:

The contrast is set to 0.

The backlight is permanently activated. Menu setting:

LIGHTING ✓

Retention

It is a requirement of system and machine controllers for operating states or actual values to have retentive settings. What this means is that the values will be retained safely even after the supply voltage to a machine or system has been switched off and are also retained until the next time the actual value is overwritten.

The following operands and function blocks can be set to operate retentively:

- Markers
- Counter function blocks,
- Data function blocks and
- Timing relays.

Operating hours counter

easy800 provides four retentive operating hours counters. They are always retentive and can only be selectively deleted with a reset command.

Retentive data volume

200 bytes is the maximum memory range for retentive data (operating hours counters are not included).

Markers

A user-definable and consistent marker range can be declared as retentive.

Counters

All C., CH.. and CI.. function blocks can be operated with retentive actual values.

Data function blocks

A user-definable consistent data function block range can be operated with retentive actual values.

Timing relays

A user-definable and consistent range for timing relays can be run with retentive actual values.

Requirements

In order to make data retentive, the relevant markers and function blocks must have been declared as retentive.



Attention!

The retentive data is saved every time the power supply is switched off, and read every time the device is switched on. The data integrity of the memory is guaranteed for 10¹⁰ read/write cycles.

Setting retentive behaviour

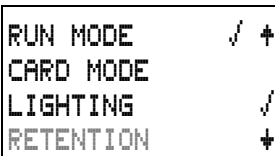
Requirement:
easy must be in STOP mode.

- ▶ Switch to the System menu.

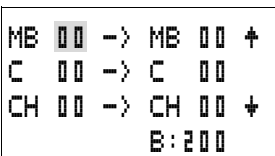


If easy is protected by a password, the System menu will not be available until easy is "unlocked" (see → Section "Unlocking easy" from Page 267).

The default setting of easy is for no retentive actual value data to be selected. When easy is in STOP mode or has been switched to a de-energized state, all actual values are cleared.



- ▶ Switch to STOP mode.
- ▶ Switch to the System menu.
- ▶ Proceed to the SYSTEM menu and continue to the RETENTION... menu.
- ▶ Press the **OK** button.



The first screen display is the selection of the marker range.

- ▶ ^v Select a range.
- ▶ Press **OK** to access the Entry modes.
 - < > Select a position from/to,
 - ^v Set a value.
- ▶ Save the input from .. to .. with **OK**.

Press **ESC** to exit the input of the retentive ranges.

Up to six differing ranges can be selected.

```

CI 00 -> CI 00 +
DB 00 -> DB 00
T  00 -> T  00 +
      B: 200
  
```



The display on the lower right **B: 200** indicates the number of free bytes.

```

MB 01 -> MB 04
C  12 -> C  16
CH 00 -> CH 00
CI 00 -> CI 00
DB 01 -> DB 16
T  26 -> T  32
      B: 076
  
```

Example:

MB 01 to MB 04, C 12 to C 16, DB 01 to DB 16, T 26 to T 32 should be retentive.

124 bytes have been assigned to the retentive data range. 76 bytes are still available.

Deleting ranges

Set the ranges to be erased to the values from 00 to 00.

e.g.: **MB 00 -> MB 00**. The markers are no longer retentive.

Deleting retentive actual values of markers and function blocks

The retentive actual values are cleared if the following is fulfilled (applies only in STOP mode):

- When the circuit diagram is transferred from EASY-SOFT(-PRO) (PC) or from the memory card to easy, the retentive actual values are reset to 0. This also applies when there is no program on the memory card. In this case the old circuit diagram is retained in easy.
- When changing the respective retentive range.
- When the circuit diagram is deleted via the DELETE PROGRAM menu.

Transferring retentive behaviour

The setting for retentive behaviour is a circuit diagram setting. In other words, the setting of the retentive menu may also under certain circumstances be transferred to the memory card or by uploading or downloading from the PC.

Changing the operating mode or the circuit diagram

When the operating mode is changed or the easy circuit diagram is modified, the retentive data is normally saved together with their actual values. The actual values of relays no longer being used are also retained.

Changing the operating mode

If you change from RUN to STOP and then back to RUN, the actual values of the retentive data will be retained.

Modifying the easy circuit diagram

If a modification is made to the easy circuit diagram, the actual values will be retained.

Changing the startup behaviour in the SYSTEM menu

The retentive actual values are retained in easy regardless of the setting.

Modification of the retentive range

If the set retentive ranges are reduced, only the actual values saved in the range will remain.

If the retentive range is extended, the older data is retained. The new data is written with the current actual values in RUN mode.

Displaying device information

Device information is provided for service tasks and for determining the capability of the device concerned.

This function is only available with devices featuring a display.

Exception: Terminal mode MFD-Titan.

easy800 enables the display of the following device information:

- AC or DC power supply
- T (transistor output) or R (relay output)
- C (clock provided)
- A (analog output provided)
- LCD (display provided)
- easy-NET (easy-NET provided)
- OS: 1.10.204 (operating system version)
- CRC: 25825 (checksum of the operating system).

► Switch to the System menu.



If easy is protected by a password, the System menu will not be available until easy is "unlocked" (see → Section "Unlocking easy" from Page 267).

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

- Select the SYSTEM menu.
- Press the **OK** button.

```
CARD MODE †
DISPLAY...
RETENTION...
INFORMATION... †
```

- Use the \vee button to select the INFORMATION menu and press **OK**.

This will display all device information.

```
DC TCA LCD NET
OS : 1.10.208
CRC: 21119
```

Example: EASY822-DC-TC

```
DC RC LCD NET
OS : 1.10.208
CRC: 21119
```

Example: EASY819-DC-RC

► Press **ESC** to leave the display.

```
CARD MODE ↑
DISPLAY...
RETENTION...
INFORMATION... ↓
```

7 Inside easy

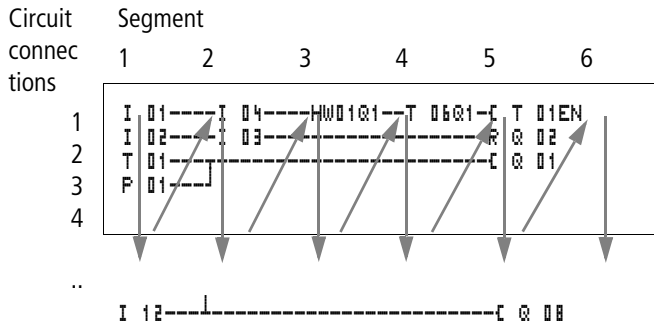
easy Program cycle

In conventional control systems, a relay or contactor control processes all the circuit connections in parallel. The speed with which a contactor switches is thus dependent on the components used, and ranges from 15 to 40 ms for relay pick-up and drop-out.

With easy 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 easy circuit diagram is processed cyclically every 0.1 to 40 ms.

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

How easy evaluates the circuit diagram:



In the first four segments easy evaluates the contact fields in succession. easy checks whether contacts are switched in parallel or in series and saves the switching states of all contact fields.

In the fifth segment, easy assigns the new switching states to all the coils in one pass.

The sixth segment is located outside of the circuit diagram. easy uses this to:

Evaluating function blocks

- process the function blocks which are used: the output data of a function block is updated immediately after processing. easy processes the function blocks according to the function block list (→ FUNCTION RELAYS menu) from top to bottom. You can sort the function block list with EASY-SOFT (-PRO) from Version 4.04. You can then for example, use the results consecutively.
- to establish contact to the "outside world": The output relays Q 01 to Q (S).. are switched and the inputs I1 to I (R).. are read once more.
- easy also copies all the new switching states to the status image register.
- exchange all data on the easy-NET network (read and write).

easy only uses this status image for one cycle. This ensures that each circuit connection is evaluated with the same switching states for one cycle, even if the input signals I1 to I12 change their status several times within a cycle.



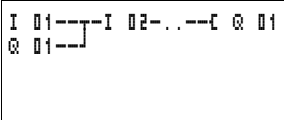
The following must be observed when operating a PID controller function block!

The cycle time of the program must be less than the scan time of the controller. If the cycle time is greater than the controller scan time, the controller will not be able to achieve constant results.

Effects on the creation of the circuit diagram

easy evaluates the circuit diagram in these six segments in succession. You should take into account two factors when creating your circuit diagram.

- The changeover of a relay coil does not change the switching state of an associated contact until the next cycle starts.
- Always wire forward or from top to bottom. Never work backwards.

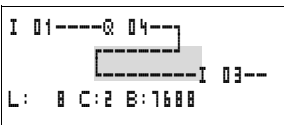


Example: self-latching with own contact

Start condition:
Inputs I1 and I2 are switched on.
Q1 is switched off.

This is the circuit diagram of a self-latching circuit. If I1 and I2 are closed, the switching state of relay coil Q 01 is “latched” via contact Q 01.

- **1st cycle:** Inputs I1 and I2 are switched on. Coil Q1 picks up.
- Contact Q 01 remains switched off since easy evaluates from left to right. The first coil field was already passed when easy refreshes the output image in the 6th segment.
- **2nd cycle:** The self-latching now becomes active. easy has transferred the coil states at the end of the first cycle to contact Q 01.



Example: Do not wire backwards

This example is shown in Section “Creating and modifying connections” Page 106. It was used here to illustrate how NOT to do it.

In the third rung, easy finds a connection to the second rung in which the first contact field is empty. The output relay is not switched.

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

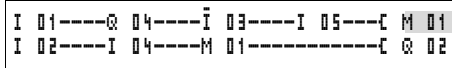


Figure 124: Circuit diagram with M 01 marker relay

How easy evaluates the high-speed counters CF, CH and CI

In order to evaluate the count pulses of 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.

Delay times for inputs and outputs

The time from reading the inputs and outputs to switching contacts in the circuit diagram can be set in easy via the delay time.

This function is useful, for example, in order to ensure a clean switching signal despite contact bounce.

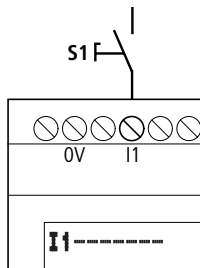


Figure 125: easy input assigned a switch

easy-DC and easy-AC units function with different input voltages and therefore also have different evaluation methods and delay times.

Delay times with easy-DC basic units

The delay time for DC signals is 20 ms.

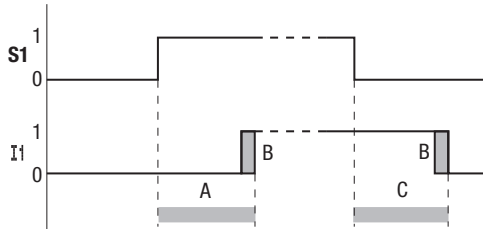


Figure 126: Delay times for easy-DC

An input signal S1 must therefore be 15 V or 8 V for at least 20 ms on the input terminal before the switch contact will change from 0 to 1 (A). If applicable, this time must also include the cycle time (B) since easy does not detect the signal until the start of a cycle.

The same debounce delay (C) applies when the signal drops out from 1 to 0.



If you use high-speed counter function blocks, the debounce delay time for the inputs is 0.025 ms. Otherwise it is not possible to count high-speed signals.

If the debounce is switched off, easy responds to an input signal after just 0.25 ms.

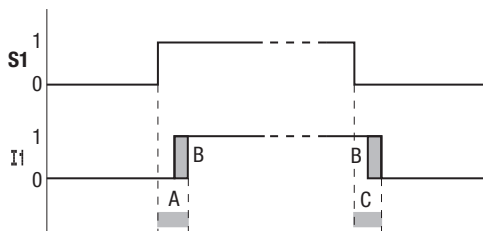


Figure 127: Switching behaviour with input debounce disabled

Typical delay times with the debounce delay disabled are:

- On-delay for
 - I1 to I4: 0.025 ms
 - I5 to I12: 0,25 ms (DC)
- Off-delay for
 - I1 to I4: 0.025 ms
 - I5, I6 and I9 to I10: 0.4 ms
 - I7, I8, I11 and I12: 0.2 ms (DC),



Ensure that input signals are noise-free if the input debounce is disabled. easy will even react to very short signals.

Delay time with easy AC basic units

The input debounce with AC voltage signals depends on the frequency:

- On-delay
 - 80 ms at 50 Hz, 66 ms at 60 Hz
- Off-delay for
 - I1 to I6 and I9 to I12: 80 ms (66 ms)
 - I7 and I8: 120 ms (100 ms) with EASY412-AC

The corresponding values for 60 Hz are given in brackets.

Behaviour with and without delay time

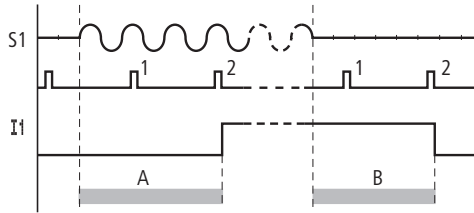


Figure 128: On-delay for easy-AC

If the debounce delay is switched on, easy checks at 40 ms (33 ms) intervals whether there is a half-wave present at an input terminal (1st and 2nd pulses in A). If easy detects two pulses in succession, the device switches on the corresponding input internally.

If this is not the case, the input is switched off again as soon as easy does not detect two successive half-waves (1st and 2nd pulses in B).

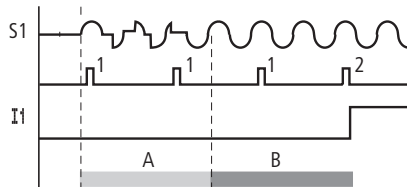


Figure 129: Push-button with bounce

If a button or switch bounces (A), the delay time may be extended by 40 ms (33 ms) (A).

If the debounce delay is switched off, the delay time is reduced.

- On-delay
20 ms (16,6 ms)
- Off-delay for
I1 to I6 and I9 to I12: 20 ms (16.6 ms)
- Off-delay I7 and I8: 20 ms (16.6 ms) with EASY412-AC..

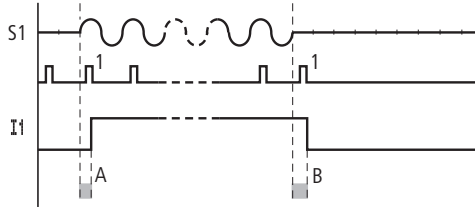


Figure 130: On and off delays

easy switches the contact as soon as it detects a pulse (A). If no pulse is detected, easy switches off the contact (B).



The procedure for changing the delay times is described in Section "Delay times for inputs and outputs" on Page 292.

Monitoring of short-circuit/overload with EASY...D.-T..

Depending on the type of easy in use, it is possible to use the internal inputs I15 and I16, R15, R16 to monitor for short-circuits or overloads on an output.

- EASY82 .-D.-T...:
 - I16: Group fault signal for outputs Q1 to Q4.
 - I15: Group fault signal for outputs Q5 to Q8.
- EASY620-D.-TE:
 - R16: Group fault signal for outputs S1 to S4.
 - R16: Group fault signal for outputs S5 to S8.

State	
Outputs	I15 or I16, R15 or R16
No fault found	0 = switched off (make contact)
At least one output has a fault	1 = switched on (make contact)



I15 and I16 can only be edited with easy versions which have transistor outputs.

The following examples are for I16 = Q1 to Q4. I15 indicates in the same way short-circuits and overloads on Q5 to Q8.

Example 1: Output with fault indication

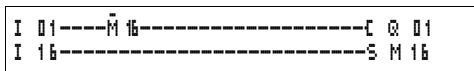


Figure 131: Circuit diagram for fault output via I16

The above circuit diagram functions as follows:

If a transistor output reports a fault, M16 is set by I16. The break contact of M16 switches off output Q1. M16 can be cleared by resetting the easy power supply.

Example 2: Output of operating state

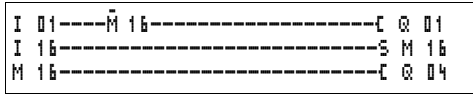


Figure 132: Output of operating state

The above circuit operates as described in example 1. The signal light is triggered at Q4 for additional overload monitoring. If Q4 has an overload, it would "pulse".

Example 3: Automatic reset of error signal

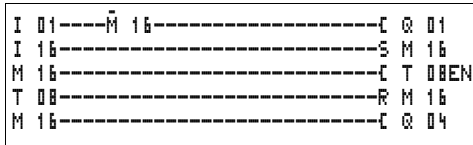


Figure 133: Automatic reset of error signal

The above circuit diagram functions in the same way as Example 2. In addition the marker M16 is reset every 60 seconds by timing relay T08 (on-delayed, 60 s). Should I16 remain at 1, M16 will continue to be set. Q1 is set briefly to 1 until I16 switches off again.

Expanding easy800

You can expand easy800 with easy models EASY618-..-RE , EASY620-D.-TE, EASY202-RE locally or use the EASY200-EASY coupling module for remote expansion with easy600 expansion. All available bus interface devices such as EASY204-DP, EASY221-CN, EASY205-ASI or EASY222-DN can be used if present.

Install the units and connect the inputs and outputs as described (see → Section "Connecting the expansion device" Page 34).

You process the inputs of the expansion devices as contacts in the easy circuit diagram in the same way as you process the inputs of the basic unit. The input contacts are assigned the operand designations R1 to R12.



R15 and R16 are the group fault inputs of the transistor expansion unit (→ Section "Monitoring of short-circuit/overload with EASY..-D.-T..", Page 297).

The outputs are processed as relay coils or contacts like the outputs in the basic unit. The output relays are S1 to S8.



EASY618-..-RE provides the outputs S1 to S6. The other outputs S7, S8 can be used internally.

How is an expansion unit recognised?

If at least one  . . contact or  . . coil/contact is used in the circuit diagram, the basic unit assumes that an expansion device is connected.

Transfer behaviour

The input and output data of the expansion units is transferred serially in both directions. Take into account the modified reaction times of the inputs and outputs of the expansion units:

Input and output reaction times of expansion units

The debounce setting has no effect on the expansion unit.

Transfer times for input and output signals:

- Local expansion
 - Time for inputs R1 to R12: 30 ms + 1 cycle
 - Time for outputs S1 to S6 (S8): 15 ms + 1 cycle
- Decentralised (remote) expansion
 - Time for inputs R1 to R12: 80 ms + 1 cycle
 - Time for outputs S1 to S6 (S8): 40 ms + 1 cycle

Function monitoring of expansion units

If the power supply of the expansion unit is not present, no connection can be established between it and the basic unit. The expansion inputs R1 to R12, R15, R16 are incorrectly processed in the basic unit and show status 0. It cannot be assured that the outputs S1 to S8 are transferred to the expansion unit.



Warning!

Continuously monitor the functionality of the easy expansion in order to prevent switching errors in the machine or plant.

The status of the internal input I14 of the basic unit indicates the status of the expansion unit:

- I14 = "0": expansion unit is functional
- I14 = "1": expansion unit is not functional

Example

Power can be applied to the expansion unit later than the basic unit. This means that the basic unit is switched to RUN when the expansion unit is absent. The following easy circuit diagram detects if the expansion unit is functional or not functional.

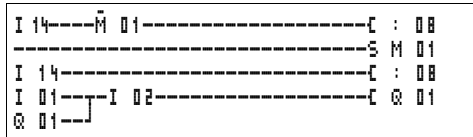


Figure 134: Circuit diagram for expansion testing

As long as I 14 is 1, the remaining circuit diagram is skipped. If I 14 is 0, the circuit diagram is processed. If the expansion unit drops out for any reason, the circuit diagram is skipped. M 01 detects whether the circuit diagram was processed for at least one cycle after the power supply is switched on. If the circuit diagram is skipped, all the outputs retain their previous state.

QA analog output

The analog output operates with decimal values between 0 and 1023. This corresponds to a 10-bit resolution. At the output this corresponds to a physical voltage between 0 V and 10 V DC.

Negative values such as: -512 are evaluated as zero and output as 0 V DC.

Positive values greater than 1023, such as: 2047, are evaluated as 1023 and output as 10 V DC.

Behaviour with assignment of values exceeding 1023



Warning!

Valid for operating system versions up to 1.02.154: If an analog output is assigned with a value greater than 1023, a physical value is output which is calculated using the following formula. The result can cause sudden steps on the analog output.

Formula for calculation of the physical value:

$$Y = X - 1023 \times n$$

Y = physical value

X = value which is assigned to the QA operands

n = factor which defines the frequency of occurrence of the 1023 as a whole number in "X".

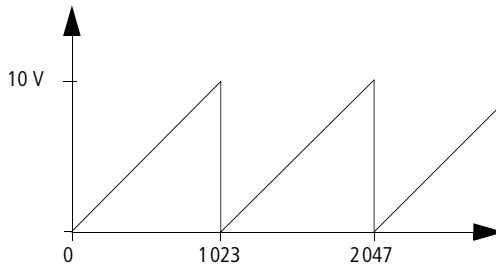


Figure 135: Step behaviour on analog output QA

Example

$$X = 1539; \rightarrow n = 1$$

$$Y = 1539 - 1023 \times 1$$

$$Y = 515$$

$$QA = \frac{10 \text{ V}}{1023} \times 515$$

$$QA = 5.03 \text{ V}$$

Loading and saving programs

You can either use the easy interface to save programs to a memory card or use EASY-SOFT(-PRO) and a transmission cable to transfer them to a PC.

EASY...-...XMGD without display and keypad

easy models without buttons and a display can be loaded with the easy program via EASY-SOFT-PRO or automatically from the fitted memory card every time the power supply is switched on.

Program compatibility of the hardware

All programs, even if functions are not supported by the hardware can be loaded on an easy800 device.

Example: a program with analog value comparators is loaded on an AC version. The analog value comparator works with the value zero.

Interface

The easy interface is covered. Remove the cover carefully.

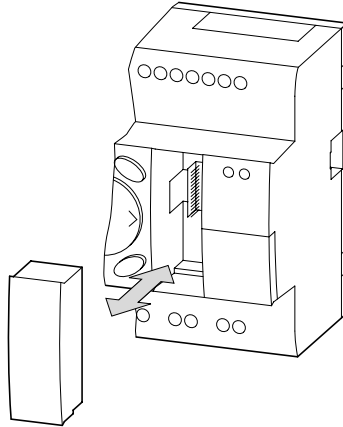


Figure 136: Remove cover and plug-in

► To close the slot again, push the cover back onto the slot.

COM connection

This function is available from device version 04.

easy800 supports the COM connection via the serial interface.

Terminal mode

This function is available from device version 04.

easy800 supports terminal mode via the serial interface or via easy-NET.

Memory card

The card is available as an accessory EASY-M-256K for easy800.

Circuit diagrams containing all the relevant data can be transferred from the EASY-M-256K memory to easy800.

Each memory card can hold one easy circuit diagram.

Information stored on the memory card is “non-volatile” and thus you can use the card to archive, transfer and copy your circuit diagram.

On the memory card you can save:

- the program
- all parameter settings of the circuit diagram,
- all display texts with functions
 - the system settings,
 - Debounce,
 - P Buttons,
 - Password,
 - retention on/off and range,
 - easy-NET configuration,
 - Card mode.



Memory cards that are created with an easy device version ≤ 03 can be read with device versions > 03 .
Memory cards that are created with a device version > 03 can also only be read with device versions > 03 .

- ▶ Insert the memory card in the open interface slot.

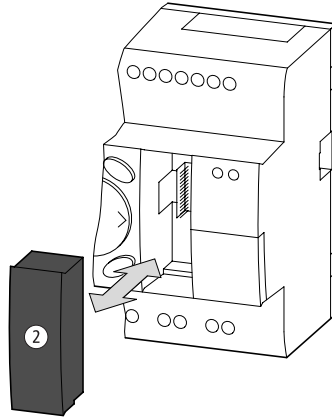


Figure 137: Fitting and removing the memory card



With easy you can insert and remove the memory card even if the power supply is switched on, without the risk of losing data.

Loading or saving circuit diagram

You can only transfer circuit diagrams in STOP mode.

The easy variants without keyboard and LCD automatically transfer the circuit diagram from the memory card to the EASY...-...X when the supply voltage is applied, if a memory card is inserted. If the memory card contains an invalid circuit diagram, the circuit-diagram installed in the easy is retained.

```
PROGRAM
DELETE PROG
CARD
```

- ▶ Switch to STOP mode.
- ▶ Select PROGRAM... from the main menu.
- ▶ Select the CARD... menu option.

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

```
DEVICE-CARD
CARD-DEVICE
DELETE CARD
```

You can transfer a circuit diagram from easy to the card and from the card to the easy memory or delete the content of the card.



If the operating voltage fails during communication with the card, repeat the last procedure since easy may not have transferred or deleted all the data.

- ▶ After transmission, remove the memory card and close the cover.

Saving a circuit diagram on the card

- ▶ Select DEVICE-> CARD.
- ▶ Confirm the prompt with **OK** to delete the contents of the memory card and replace it with the easy circuit diagram.

```
REPLACE ?
```

Press **ESC** to cancel.

Loading a circuit diagram from the card

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

```
DEVICE-CARD
CARD-DEVICE
DELETE CARD
```

Press **ESC** to cancel.

```
INVALID PROG
```

If there are transmission problems, easy will display the INVALID PROG message.

This either means that the memory card is empty or that the circuit diagram on the card contains function relays that easy does not recognise.

The "analog comparator" function relay is only used with easy DC 24 V DC versions.



If the memory card is password-protected, the password will also be transferred to the easy memory and will be active immediately.



Deleting a circuit diagram on the card

- ▶ Select the DELETE CARD menu option.
- ▶ Press **OK** to confirm the prompt and to delete the card content.

Press **ESC** to cancel.

Memory card compatibility of the programs



Memory cards with programs are always read by easy800 devices with the newer (higher) operating system version. The program is executable. If programs are written with a newer operating system (higher number) on the memory card, this program can only be read and executed by the same version or a higher one.

EASY-SOFT (-PRO)

EASY-SOFT (-PRO) is a PC program for creating, testing and managing programs and easy circuit diagrams.

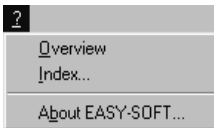


You should only transfer data between the PC and easy using the easyPC interface cable, which is available as accessory EASY800-PC-CAB.



easy cannot exchange data with the PC while the circuit diagram display is on screen.

Use EASY-SOFT(-PRO) to transfer circuit diagrams from your PC to easy and vice versa. Switch easy to RUN mode from the PC to test the program using the current wiring.



EASY-SOFT (-PRO) provides extensive help on how to use the software.

- ▶ Start EASY-SOFT (-PRO) and click on Help.

The on-line help provides all additional information about EASY-SOFT (-PRO) that you will need.

INVALID PROG

If there are transmission problems, easy will display the INVALID PROG message.

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



If the operating voltage fails during communication with the PC, repeat the last procedure. It is possible that not all the data was transferred between the PC and easy.

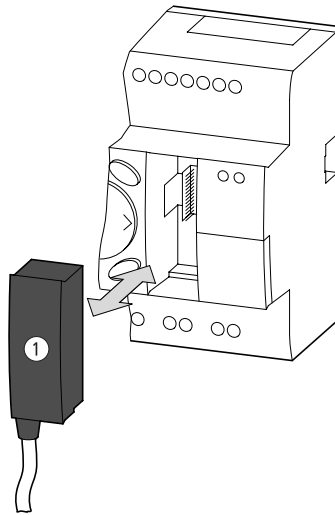


Figure 138: Fitting and removing EASY800-PC-CAB

- After transmission, remove the cable and close the cover.

Compatibility of different easy800 versions.

Device compatibility

easy800 is being further developed for you. The new easy800 devices have more functions than previous devices. Please note the device version number on the left of the device.

e.g.: 039000000607 device version 03

To version 03: easy800 1st edition; up to operating system 1.0x

From version 04: easy800 2nd edition; from operating system version 1.1x



The new easy800 devices have all the functions of the previous devices. All programs of the previous devices can be run on the new devices of the same type. Old programs can be transferred directly using the memory card.

The following functions have been added from device version 04 upwards.

- COM connection with an MFD-Titan,
- Terminal mode in conjunction with an MFD-Titan,
- BUSY indication in the Status display, PC interface active,
- DISPLAY menu, contrast and backlight setting,
- INFORMATION menu, information about the device
- Zoom in and out of the power flow display,
- D text display function block was developed for displaying all actual values and for inputting constants,
- BC block comparison function block,
- BT block transfer function block,
- DC PID controller function block,
- FT PT1 signal smoothing filter function block,
- LS value scaling function block,
- NC numerical converter function block,
- PW pulse width modulation function block,
- ST set cycle time function block,
- VC value limitation function block.

Device version

Every easy800 has the device version number printed on the left of the device housing. The device version is indicated by the first two digits of the device number.

Example:

DC 20.4 ...28.8 V 4 W 03-9000000607

This device is of device version 03.

The device version 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 EASY-SOFT or EASY-SOFT -PRO.

Appendix

Technical data

General

easy800

	easy800
Dimensions W × H × D	
[mm]	107.5 × 90 × 72
[inches]	4.23 × 3.54 × 2.84
Space units (SU) width	6
Weight	
[g]	320
[lb]	0,705
Mounting	Top-hat rail to DIN 50022, 35 mm or screw mounting with 3 ZB4-101-GF1 mounting feet

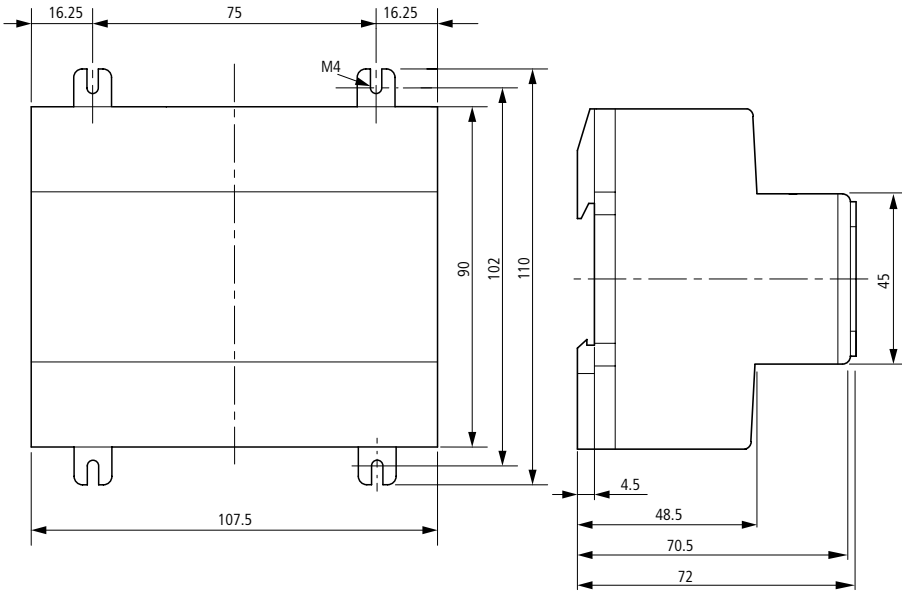


Figure 139: Dimensions easy800 in mm
 (Stated in inches → Table 9)

Table 9: Dimensions in inches

mm	inches	mm	inches
4.5	0.177	75	2.95
16.25	0.64	90	3.54
48.5	1.91	102	4.01
70.5	2.78	107.5	4.23
72	2.83	110	4.33

Climatic environmental conditions (Cold to IEC 60068-2-1, Heat to IEC 60068-2-2)		
Ambient temperature Installed horizontally/vertically	°C, (°F)	-25 to 55, (-13 to 131)
Condensation		Prevent condensation with suitable measures
LCD display (reliably legible)	°C, (°F)	0 to 55, (32 to 131)
Storage/transport temperature	°C, (°F)	-40 to 70, (-40 to 158)
Relative humidity (IEC 60068-2-30), no moisture condensation	%	5 to 95
Air pressure (operation)	hPa	795 to 1080
Ambient mechanical conditions		
Pollution degree		2
Degree of protection (EN 50178, IEC 60529, VBG4)		IP20
Oscillations (IEC 60068-2-6)		
constant amplitude 0.15 mm	Hz	10 to 57
constant acceleration 2 g	Hz	57 to 150
Shocks (IEC 60068-2-27) semi-sinusoidal 15 g/11 ms	Shocks	18
Drop (IEC 60068-2-31)	Drop height	mm
		50
Free fall, when packed (IEC 60068-2-32)	m	1
Electromagnetic compatibility (EMC)		
Electrostatic discharge (ESD), (IEC/EN 61000-4-2, severity level 3)		
Air discharge	kV	8
Contact discharge	kV	6
Electromagnetic fields (RFI), (IEC/EN 61000-4-3)	V/m	10
Radio interference suppression (EN 55011, IEC 61000-6-1, 2, 3, 4), limit class		B

Burst (IEC/EN 61000-4-4, severity level 3)		
Power cables	kV	2
Signal cables	kV	2
High energy pulses (Surge) easy-AC (IEC/EN 61000-4-5), power cable symmetrical	kV	2
High energy pulses (Surge) easy-DC (IEC/EN 61000-4-5, severity level 2), power cable symmetrical	kV	0,5
Line-conducted interference (IEC/EN 61000-4-6)	V	10
Dielectric strength		
Measurement of the air clearance and creepage distance		EN 50 178, UL 508, CSA C22.2, No 142
Dielectric strength		EN 50 178
Overvoltage category/degree of pollution		II/2
Tools and cable cross-sections		
Solid, minimum to maximum	mm ²	0.2 to 4
	AWG	22 to 12
Flexible with ferrule, minimum to maximum	mm ²	0.2 to 2.5
	AWG	22 to 12
Factory wiring:	AWG	30
Slot-head screwdriver, width	mm	3.5 × 0.8
	inch	0.14 × 0.03
Tightening torque	Nm	0,6

Backup/accuracy of real-time clock (only with easy-C)																																																																	
Clock battery back-up																																																																	
<table border="1"> <caption>Graph Data: Backup time (hours) vs service life (years)</caption> <thead> <tr> <th>Service life (years)</th> <th>Backup time (hours) at 25°C</th> <th>Backup time (hours) at 55°C</th> </tr> </thead> <tbody> <tr><td>0</td><td>180</td><td>180</td></tr> <tr><td>1</td><td>175</td><td>140</td></tr> <tr><td>2</td><td>170</td><td>110</td></tr> <tr><td>3</td><td>165</td><td>90</td></tr> <tr><td>4</td><td>160</td><td>75</td></tr> <tr><td>5</td><td>155</td><td>65</td></tr> <tr><td>6</td><td>150</td><td>55</td></tr> <tr><td>7</td><td>145</td><td>45</td></tr> <tr><td>8</td><td>140</td><td>35</td></tr> <tr><td>9</td><td>135</td><td>25</td></tr> <tr><td>10</td><td>130</td><td>20</td></tr> <tr><td>11</td><td>125</td><td>15</td></tr> <tr><td>12</td><td>120</td><td>-</td></tr> <tr><td>13</td><td>115</td><td>-</td></tr> <tr><td>14</td><td>110</td><td>-</td></tr> <tr><td>15</td><td>105</td><td>-</td></tr> <tr><td>16</td><td>100</td><td>-</td></tr> <tr><td>17</td><td>95</td><td>-</td></tr> <tr><td>18</td><td>90</td><td>-</td></tr> <tr><td>19</td><td>85</td><td>-</td></tr> </tbody> </table>			Service life (years)	Backup time (hours) at 25°C	Backup time (hours) at 55°C	0	180	180	1	175	140	2	170	110	3	165	90	4	160	75	5	155	65	6	150	55	7	145	45	8	140	35	9	135	25	10	130	20	11	125	15	12	120	-	13	115	-	14	110	-	15	105	-	16	100	-	17	95	-	18	90	-	19	85	-
Service life (years)	Backup time (hours) at 25°C	Backup time (hours) at 55°C																																																															
0	180	180																																																															
1	175	140																																																															
2	170	110																																																															
3	165	90																																																															
4	160	75																																																															
5	155	65																																																															
6	150	55																																																															
7	145	45																																																															
8	140	35																																																															
9	135	25																																																															
10	130	20																																																															
11	125	15																																																															
12	120	-																																																															
13	115	-																																																															
14	110	-																																																															
15	105	-																																																															
16	100	-																																																															
17	95	-																																																															
18	90	-																																																															
19	85	-																																																															
1 backup time in hours																																																																	
2 service life in years																																																																	
Accuracy of the real-time clock																																																																	
Per day	s/day	± 5																																																															
Per year	h/year	± 0.5																																																															
Repetition accuracy of timing relays																																																																	
Accuracy of timing relay (from value)	%	± 0.02																																																															
Resolution																																																																	
Range "s"	ms	5																																																															
Range "M:S"	S	1																																																															
Range "H:M"	min.	1																																																															
Retentive memory																																																																	
Read/write cycles of retentive memory		10 ¹⁰																																																															
Circuit connections (basic units)																																																																	
easy800		256																																																															

Power supply

EASY819-AC-RC..

		EASY819-AC-RC..
Rated value (sinusoidal) U_e	V AC, (%)	100/110/115/120/230/240, (+10/-15)
Operating range	V AC	85 to 264
Frequency, rated value, tolerance	Hz, (%)	50/60, (± 5)
Input current consumption		
at 115/120 V AC 60 Hz, typical	mA	70
at 230/240 V AC 50 Hz, typical	mA	35
Voltage dips, IEC/EN 61131-2	ms	20
Power loss		
at 115/120 V AC, typical	VA	10
at 230/240 V AC, typical	VA	10

EASY8..-DC-...

		EASY8..-DC-...
Rated voltage		
Rated value	V DC, (%)	24, (+20, -15)
Permissible range	V DC	20.4 to 28.8
Residual ripple	%	≤ 5
Input current at 24 V DC, typical	mA	140
Voltage dips, IEC/EN 61131-2	ms	10
Power loss at 24 V DC, typical	W	3,4

Inputs**EASY8..-AC-...**

		EASY8..-AC-R..
Digital inputs 115/230 V AC		
Number		12
Status display		LCD (if provided)
Electrical isolation		
To supply voltage		No
Between each other		No
To the outputs		Yes
to the PC interface, memory card, easy-NET, EASY-LINK		Yes
Rated voltage L (sinusoidal)		
0 signal	V AC	0 to 40
1 signal	V AC	79 to 264
Rated frequency	Hz	50/60
Input current for state "1" I1 to I6, I9 to I12,		
at 230 V 50 Hz	mA	10 × 0.5
at 115 V, 60 Hz	mA	10 × 0.25

		EASY8..-AC-R..
Input current at state "1" I7, I8,		
at 230 V 50 Hz	mA	2 × 6
at 115 V, 60 Hz	mA	2 × 4
Delay time for 0 to 1 and 1 to 0 for I1 to I6, I9 to I12		
Debounce ON	ms, (Hz)	80, (50) 66 ^{2/3} , (60)
Debounce OFF	ms, (Hz)	20, (50) 16 ^{2/3} , (60)
Delay time I7, I8 for 1 to 0		
Debounce ON	ms, (Hz)	120, (50) 100, (60)
Debounce OFF	ms, (Hz)	40, (50) 33, (60)
Delay time I7, I8 for 0 to 1		
Debounce ON	ms, (Hz)	80, (50) 66 ^{2/3} , (60)
Debounce OFF	ms, (Hz)	20, (50) 16 ^{2/3} , (60)
Max. permissible cable length (per input)		
I1 to I6, I9 to I12), typical, debounce on	m	100
I1 to I6, I9 to I12), typical, debounce off	m	60
I7, I8, typical	m	100

EASY8..-DC-...

		EASY8..-DC-...
Digital inputs		
Number		12
Inputs usable as analog inputs, (I7, I8, I11, I12)		4
Status display		LCD (if provided)
Electrical isolation		
To supply voltage		No
Between each other		No
To the outputs		Yes
to the PC interface, memory card, easy-NET, EASY-LINK		Yes
Rated voltage		
Rated value	V DC	24
0 signal		
I1 to I6 and I9 to I10	V DC	< 5
I7, I8, I11, I12	V DC	< 8
1 signal		
I1 to I6 and I9 to I10	V DC	> 15
I7, I8, I11, I12	V DC	> 8
Input current on 1 signal		
I1 to I6, I9 to I10 at 24 V DC	mA	3,3
I7, I8, I11, I12 at 24 V DC	mA	2,2
Delay time for 0 to 1		
Debounce ON	ms	20
Debounce off, typical		
I1 to I4	ms	0,025
I5, I6, I9, I10	ms	0,25
I7, I8, I11, I12	ms	0,15

		EASY8..-DC-...
Delay time for 1 to 0		
Debounce ON	ms	20
Debounce off, typical		
I1 to I4	ms	0,025
I5, I6, I9, I10	ms	0,25
I7, I8, I11, I12	ms	0,15
Cable length (unshielded)	m	100
High-speed counter inputs, I1 to I4		
Number		4
Cable length (shielded)	m	20
High-speed up and down counters		
Counting frequency	kHz	< 5
Pulse shape		Square wave
Mark-to-space ratio		1:1
Frequency counters		
Counting frequency	kHz	< 5
Pulse shape		Square wave
Mark-to-space ratio		1:1
Incremental encoder counters		
Counting frequency	kHz	< 3
Pulse shape		Square wave
Counter inputs I1 and I2, I3 and I4		2
Signal offset		90°
Mark to space ratio		1:1

		EASY8..-DC-...
Analog inputs		
Number		4
Electrical isolation		
To supply voltage		No
To the digital inputs		No
To the outputs		Yes
To the easy-NET network		Yes
Input type		DC voltage
Signal range	V DC	0 to 10
Resolution analog	V	0,01
Resolution digital	Bit	10
	Value	0 to 1023
Input impedance	k Ω	11,2
Accuracy		
2 easy-devices, from actual value	%	± 3
Within a unit, from actual value, (I7, I8, I11, I12)	%	± 2
Conversion time, analog/digital		
Debounce ON:	ms	20
Debounce OFF:		Each cycle time
Input current	mA	< 1
Cable length (shielded)	m	30

Relay outputs

EASY8...-R..

		EASY8...-R..
Number		6
Type of outputs		Relays
In groups of		1
Connection of outputs in parallel to increase the output		Not permissible
Protection for an output relay		
Miniature circuit-breaker B16	A	16
or fuse (slow-blow)	A	8
Potential isolation to mains power supply, input, PC interface, memory card, NET network, EASY-LINK		Yes
Safe isolation	V AC	300
Basic insulation	V AC	600
Mechanical lifespan	Switch operations	10×10^6
Contacts relays		
Conventional thermal current, (UL)	A	8, (10)
Recommended for load at 12 V AC/DC	mA	> 500
Protected against short-circuit $\cos \varphi = 1$ 16 A characteristic B (B16) at	A	600
Protected against short-circuit $\cos \varphi = 0.5$ to 0.7 16 A characteristic B (B16) at	A	900
Rated impulse withstand voltage U_{imp} contact coil	kV	6
Rated insulation voltage U_i		
Rated operational voltage U_e	V AC	250
Safe isolation to EN 50178 between coil and contact	V AC	300
Safe isolation to EN 50178 between two contacts	V AC	300

		EASY8...-R..
Making capacity, IEC 60947		
AC-15 250 V AC, 3 A (600 Ops/h)	Switch operations	300 000
DC-13 L/R \leq 150 ms 24 V DC, 1 A (500 Ops/h)	Switch operations	200 000
Breaking capacity, IEC 60947		
AC-15 250 V AC, 3 A (600 Ops/h)	Switch operations	300 000
DC-13 L/R \leq 150 ms 24 V DC, 1 A (500 Ops/h)	Switch operations	200 000
Filament lamp load		
1 000 W at 230/240 V AC	Switch operations	25 000
500 W at 115/120 V AC	Switch operations	25 000
Fluorescent tube with ballast, 10 \times 58 W at 230/240 V AC	Switch operations	25 000
Conventional fluorescent tube, compensated, 1 \times 58 W at 230/240 V AC	Switch operations	25 000
Conventional fluorescent tube, uncompensated, 10 \times 58 W at 230/240 V AC	Switch operations	25 000
Operating frequency, relays		
Mechanical switch operations	Switch operations	10 mill. (10^7)
Mechanical switching frequency	Hz	10
Resistive lamp load	Hz	2
Inductive load	Hz	0,5

UL/CSA

Uninterrupted current at 240 V AC/24 V DC		A	10/8
AC	Control Circuit Rating Codes (utilisation category)		B300 Light Pilot Duty
	Max. rated operational voltage	V AC	300
	Max. thermal uninterrupted current $\cos \varphi = 1$ at B300	A	5
	Maximum make/break capacity $\cos \varphi k 1$ (Make/break) at B300	VA	3600/360
DC	Control Circuit Rating Codes (utilisation category)		R300 Light Pilot Duty
	Max. rated operational voltage	V DC	300
	Max. thermal uninterrupted current at R300	A	1
	Maximum make/break capacity at R300	VA	28/28

Transistor outputs**EASY8..-D.-T..**

		EASY8..-DC-T..
Number of outputs		8
Contacts		Semiconductors
	Rated voltage U_e	V DC
	Permissible range	24
	Residual ripple	V DC
		20.4 to 28.8
		%
		≤ 5
Supply current		
	at 0 state, typical/maximum	mA
	at 1 state, typical/maximum	mA
		18/32
		24/44
Reverse polarity protection		Yes
Attention! If voltage is applied to the outputs when the polarity of the power supply is reversed, this will result in a short circuit.		
Potential isolation to the inputs, supply voltage, PC interface, memory card, NET network, EASY-Link		Yes
Rated current I_e at state 1, maximum		A
		0,5

		EASY8..-DC-T..
Lamp load		
Q1 to Q4 without R_V	W	3
Q5 to Q8 without R_V	W	5
Residual current at state "0" per channel	mA	< 0.1
Max. output voltage		
On 0 state with external load, 10 M Ω	V	2,5
On 1 state, $I_e = 0.5$ A		$U = U_e - 1$ V
Short-circuit protection		Yes
Thermal (Q1 to Q4) (evaluation with diagnostics inputs I16, I15)		
Short-circuit tripping current for $R_a \leq 10$ m Ω (depending on number of active channels and their load)	A	$0.7 \leq I_e \leq 2$
Maximum total short-circuit current	A	8
Peak short-circuit current	A	16
Thermal cutout		Yes
Maximum switching frequency with constant resistive load $R_L = 100$ k Ω (depends on program and load)	Switch operations/h	40000
Parallel connection of outputs with resistive load; inductive load with external suppression circuit (\leftrightarrow Section "Connecting transistor outputs", Page 53); combination within a group		Yes
Group 1: Q1 to Q4		
Group 2: Q5 to Q8		
Maximum number of outputs		4
Total maximum current	A	2
Attention! Outputs must be actuated simultaneously and for the same time duration.		
Status display of the outputs		LCD display (if provided)

Inductive load **without external suppressor circuit**

General explanations:

$T_{0.95}$ = time in milliseconds until 95 % of the stationary current is reached

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

Utilisation category in groups for

- Q1 to Q4, Q5 to Q8

$T_{0.95} = 1 \text{ ms}$ $R = 48 \Omega$ $L = 16 \text{ mH}$	Utilisation factor per group g =		0,25
	Relative duty factor	%	100
	Max. switching frequency $f = 0.5 \text{ Hz}$ Max. duty factor DF = 50 %	Switch operations/h	1 500
DC13 $T_{0.95} = 72 \text{ ms}$ $R = 48 \Omega$ $L = 1.15 \text{ H}$	Utilisation factor g =		0,25
	Relative duty factor	%	100
	Max. switching frequency $f = 0.5 \text{ Hz}$ Max. duty factor DF = 50 %	Switch operations/h	1 500

Other inductive loads:

$T_{0.95} = 15 \text{ ms}$ $R = 48 \Omega$ $L = 0.24 \text{ H}$	Utilisation factor $g =$		0,25
	Relative duty factor	%	100
	Max. switching frequency $f = 0.5 \text{ Hz}$	Switch operations/h	1500
	Max. duty factor DF = 50 %		
Inductive loading with external suppressor circuit for each load (→ Section "Connecting transistor outputs", Page 53)			
	Utilisation factor $g =$		1
	Relative duty factor	%	100
	Max. switching frequency Max. duty factor	Switch operations/h	Depends on the suppressor circuit

Analog output

EASY8..-D.-T..

		EASY8..-DC-T..
Number		1
Electrical isolation		
To power supply		No
To the digital inputs		No
To the digital outputs		Yes
To the easy-NET network		Yes
Output type:		DC voltage
Signal range	V DC	0 to 10
Output current max.	mA	10
Load resistor	k Ω	1
Short-circuit and overload proof		Yes

		EASY8..-DC-T..
Resolution analog	V DC	0,01
Resolution digital	Bit	10
	Value	0 to 1023
Transient recovery time	µs	100
Accuracy (-25 to 55 °C), related to the range	%	2
Accuracy (25 °C), related to the range	%	1
Conversion time		each CPU cycle

easy-NET Network

EASY8..-...-...

		EASY8..-...-...
Number of stations		8
Bus length/transmission speed ¹⁾	m/kBaud	6/1 000 25/500 40/250 125/125 300/50 700/20 1 000/10
Electrical isolation		Yes
To power supply, inputs, outputs, EASY-LINK, PC interface, memory module		
Bus termination (→ accessories)		Yes
First and last station		
Plug connector (→ accessories)	poles	8
Type		RJ45

		EASY8...-...-...
Cable cross-sections, with cable lengths and cable resistance/m		
Cross-section up to 1000 m, < 16 mΩ/m	mm ² (AWG)	1.5 (16)
Cross-section up to 600 m, < 26 mΩ/m	mm ² (AWG)	0.75 to 0.8 (18)
Cross-section up to 400, < 40 mΩ/m	mm ² (AWG)	0.5 to 0.6 (20, 19)
Cross-section up to 250, < 60 mΩ/m	mm ² (AWG)	0.34 to 0.5 (22, 21, 20)
Cross-section up to 175, < 70 mΩ/m	mm ² (AWG)	0.25 to 0.34 (23, 22)
Cross-section up to 40, < 140 mΩ/m	mm ² (AWG)	0.13 (26)

Bus lengths above 40 m can only be achieved with cables with reinforced cross-section and connection adapter.

Surge impedance: 120 Ω.

List of the function blocks Function blocks

Element	Meaning of abbreviation	Function block designation	Page	from V.04
A	Analog value comparator	Analog value comparator	125	
AR	Arithmetic	Arithmetic	128	
BC	block compare	Data block compare	132	x
BT	block transfer	Data block transfer	139	x
BV	Boolean operation	Boolean sequence	150	
C	counter	Counter	153	
CF	counter frequency	Frequency counter	160	
CH	counter high-speed	High-speed counters	164	
CI	counter fast incremental value encoder	High-speed incremental encoder	170	
CP	comparators	Comparators	175	
D	display	Text function block	177	
DB	data block	Data function block	181	
DC	DDC controller (direct digital control)	PID controllers	183	x
FT	filter	PT1 signal smoothing filter	189	x
GT	GET	GET network	183	
HW	hora_(lat) week	Seven day timer	194	
HY	hora_(lat) year	Twelve month timer	200	
LS	linear scaling	Value scaling	204	x
MR	master reset	Master reset	211	
NC	numeric coding	Numerical converters	212	x
OT	operating time	Operating hours counter	218	
PT	PUT	PUT network	219	
PW	pulse width modulation	Pulse width modulation	216	x
SC	synchronize clocks	Synchronise clock via network	224	

Element	Meaning of abbreviation	Function block designation	Page	from V.04
ST	set time	Set cycle time	221	x
T	timing relays	Timing relays	228	
VC	value capsuling	Value limitation	241	x
:		Jumps	203	

Function block coils

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

Function block contacts

Contact	Meaning of abbreviation	Description
CY	carry	Status "1", if the value range is exceeded; (carry)
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;
Q1	output (Q1)	Switch output
QV	output value	Current actual value of the function block (e.g. counter value);
ZE	zero	Status "1", if the value of the element input QV is equal to zero;

Function block inputs (constants, operands)

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

Function block output (operands)

Input	Meaning of abbreviation	Description
QV	Output value	Output value

Other operands

Other operands	Description
MB	Marker byte (8-bit value)
IA	Analog input (if available on device!)
MW	Marker word (16-bit value)
QA	Analog output (if available on device!)
MD	Marker double word (32-bit value)
NU	Constant (number), value range from -2 147 483 648 to +2 147 483 647

Memory space requirement

The following table provides an overview of the memory requirement of the easy800 circuit connections, function blocks and their respective constants:

	Space requirement per circuit conn./function block	Space requirement per constant on the function block input
	Byte	Byte
Circuit connection	20	–
Function blocks		
A	68	4
AR	40	4
BC	48	4
BT	48	4
BV	40	4
C	52	4
CF	40	4
CH	52	4
CI	52	4
CP	32	4
D	160	
DC	96	4
DB	36	4
FT	56	4
GT	28	
HW	68	4 (per channel)
HY	68	4 (per channel)
LS	64	4
MR	20	
NC	32	4

	Space requirement per circuit conn./function block	Space requirement per constant on the function block input
	Byte	Byte
OT	36	4
PT	36	4
PW	48	4
SC	20	
ST	24	4
T	48	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 easy-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 easy-NET using EASY-SOFT (-PRO).

Index

A	Actual values	124
	Add	
	Circuit connections	72
	Contact type	71
<hr/>		
B	Break contact	88
	Inverting	106
	Bus termination resistor	57
	Button	
	ALT	72
	DEL	72
	OK	69, 86
	Buttons	20
<hr/>		
C	Cable cross-sections	58
	Cable length	41, 58
	Cable protection	39
	Circuit connections	
	Add	108
	Add new	72
	Deleting	107, 110
	Entering	106
	Number	100
	Position in circuit diagram	100

Circuit diagram	
Checking	112
Circuit connections	100
Coil field	100
Contact fields	100
Deleting	77
Display	70, 100
Fast entry	77
Grid	70, 100
Internal processing	289
Load	308
Operating buttons	86
Overview	100
Saving	306, 308
Testing	74, 112
Wiring	72, 106
Coil field	100
Coil function	
Contactor	119
Impulse relay	119
Latching relay	120
Overview	118
COM connection	304
Connecting the setpoint potentiometer	46
Connecting transistor outputs	53

Connection	
20 A sensor	48
Analog inputs	45
Analog output	55
Contactors, relays	51
Frequency generator	49
High-speed counters	49
Incremental encoders	50
Neon bulbs	42
NET network	56
Outputs	50
Proximity switch	44
Push-button actuators, switches	40, 44
Relay outputs	51
Setpoint potentiometer	46
Temperature sensor	47
Transistor outputs	53
Connection, COM	304
Contact fields	100
Contact type	106
Change	103
Contact name	103
Contact number	103
Cursor buttons	111
Deleting	106
Entering	71, 103
Invert	72
Overview	87
Counter	
High-speed	49, 164
High-speed incremental encoder	170
Counter relay	153, 170
Parameters	161, 165, 171
Cursor display	30, 86
Cycle	289

D	Debounce	
	Setting	276
	Delay time	
	Input and output	292
	Delay times	
	For easy-AC	294
	For easy-DC	293
	Deleting retentive actual values	285
	Device overview	18
	Dimensions, easy	313

E	easy	289
	easy overview	18
	EASY...-AC-E AC expansion units	37
	Expanding	62
	Expanding inputs	62
	Expanding outputs	62
	Expansion	
	Local	63
	Remote	64

F	Fixing brackets	33
	for	292
	Frequency counter	160
	Function blocks	
	Evaluating	290
	List	332
	Function relay	
	Counter	153
	Counter relay	170
	Counter, high-speed	164
	Counter, high-speed incremental encoder ..	170
	Example	243
	Frequency counter	160
	Overview	123
	Time switch	194, 200
	Timing relays	228

<hr/>	
I	Impulse relay 119
	Increasing the input current 42
	Incremental encoders 50, 170
	Input contacts 103
	Inrush current limitation 43
	Interface 304
	Interference 41
	Invert 106
<hr/>	
J	Jumps 208
<hr/>	
L	Latching 119
	Latching relay 120
	LED display 23
<hr/>	
M	Main menu
	Overview 24
	Selection 21
	Make contact 88
	Inverting 106
	Marker relay 291
	Memory card
	Deleting 308
	Insertion 305
	Menu
	Changing level 69
	Changing the language 270
	Language setting 65, 248
	Operator guidance 20
	Message
	INVALID PROG 307, 309
	MFD
	Evaluating the circuit diagram 289
	Operating modes 66
	Mode
	Changing 74

	Mounting	
	Screw fixing	33
	Top-hat rail	32
	Mounting on a mounting plate	33
<hr/>		
N	Neon bulbs	42
	Network	
	Addressing	249
	Automatic change of the RUN and STOP mode	
	255	
	Cable	35
	Cables	60
	Changing the write repetition rate	254
	Configuring input/output devices	256
	Connecting the NET network	56
	Station message types	258
	Station signs of life	259
	Topology	249
	Transfer behaviour	258
	Transmission security	262
	Network cables	60
	Number formats	99
<hr/>		
O	Operation	20
	Output relay	103
	Overload	54, 297
	Monitoring with EASY...D.-T..	297
<hr/>		
P	P buttons	111
	Activating and deactivating	277
	Parameter display	
	Counter relay	165, 171
	Timing relays	161
	Parameters	
	Change	271
	Display	271
	Inhibit access	271

Password	
Activation	266
Change	268
Deleting	268
Protection removal	269
setup	264
Pause time	254
PC connection	308
Plug (network cable)	35
Power failure	66
Power flow display	74, 75, 112
Program	289
Cycle	289
Load	303
Saving	303
<hr/>	
R Reed relay contacts	42
Relay coil	
Change	103
Coil function	104, 118
Deleting	106
Entering	73, 103
Relays	
Connecting the outputs	51
Overview	94, 97
Reset	120
Retention	
Setting behaviour	284
Transferring behaviour	286
Retention requirements	
Permitted easy models	284
Retentive behaviour	286
On circuit diagram transfer	286
Setting	284
RUN, start behaviour	66
RUN/STOP changeover	74

S	Sensor (20 mA) connection	48
	Set	120
	Setpoints	56, 124, 272
	Setting LCD backlight	281
	Setting LCD contrast	281
	Setting the time	273
	Short-circuit	54, 297
	Monitoring with EASY.-D.-T.	297
	Startup behaviour	278, 280
	After the circuit diagram is deleted	279
	Basic setting	279, 281, 282
	Possible faults	280
	Setting	278
	With upload/download to card or PC	279
	Status display	21
	Status image register	290
	Suppressing contact bounce	292
	System menu	
	Selection	21

T	Temperature sensor connection	47
	Terminal mode	304
	Tightening torque	35
	Time switch	
	24 hour switching	200
	Overnight switching	198
	Power failure	199
	Switching at the weekend	198
	Switching on working days	197
	Time overlaps	199
	Timing relays	
	On-delayed	223, 232
	Operating modes	229
	Wiring	228
	Two-wire proximity switches	42

W	Weekday setting	273
	Wiring	
	Backwards	291
	Deleting	72
	Entering	72
	Relay coils	119
	Rules	119
	with	297