

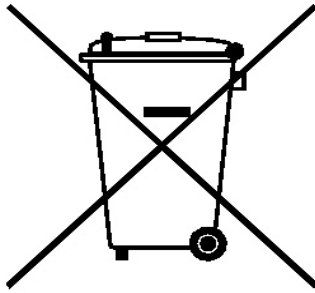


net io – Rel 4

Kurzanleitung

zu Version 1.0.0





**WEEE Reg Nr.
DE9314509**

Inhalt

Produktbeschreibung	3
Lieferumfang.....	3
Anschlüsse und Statusanzeigen.....	4
Konfiguration (Webinterface)	5
Installation des netIOMaster auf dem iSMaster.....	8
Betriebsarten	9
Verdrahtungsbeispiele.....	10
Werkseinstellungen laden	11
Kommunikation	11
Scriptbeispiel	12
Technische Daten.....	13

Produktbeschreibung

Das net io – Rel 4 ist ein Schaltmodul zur Hutschienenmontage. Es wird direkt über POE mit Spannung versorgt und benötigt somit keine zusätzliche Spannungsversorgung.

Das net io – Rel 4 ermöglicht das Schalten von bis zu 4 Lastkreisen mit 230V/8A und findet somit zumeist seinen Einsatz in Elektrounterverteilungen und Schaltkästen.

Das Produkt bietet eine komfortable Möglichkeit elektrische Verbraucher über ein TCP/IP-Netzwerk durch eine Mediensteuerung ein- bzw. auszuschalten.

Durch eine besondere Einschaltautomatik wird das „Klebenbleiben“ von Relais bei Schaltvorgängen reduziert.

Die Konfiguration erfolgt über den integrierten Web Server. Die eingelesenen Signale werden über die 10/100Mbit Ethernet-Schnittstelle in das Netzwerk übermittelt.

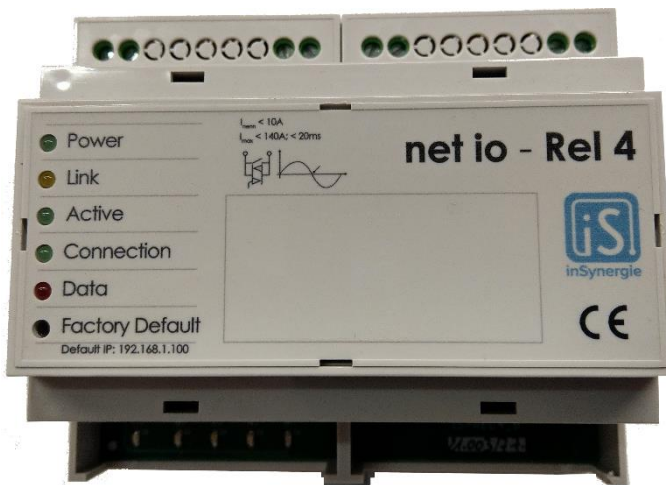
Als Übertragungsprotokoll wird MQTT (Message Queuing Telemetry Transport - <https://mqtt.org>) eingesetzt. Dafür wird im Netzwerk ein MQTT-Server ("Broker") benötigt. Dieser ist in die netIOMaster-Software integriert, es kann aber auch ein beliebiger Server verwendet werden.

Die Konfiguration erfolgt über den integrierten Web Server.

Lieferumfang

1. net io – Rel 4
2. Kurzanleitung zur Bedienung

Anschlüsse und Statusanzeigen



Power LED

Betriebsanzeige der net io Platine
Power-over-Ethernet (PoE) liegt an

Link LED

entspricht Link an der RJ45 Buchse

Active LED

entspricht Active an der RJ45 Buchse

Connection LED

Verbindung zum MQTT Broker hergestellt

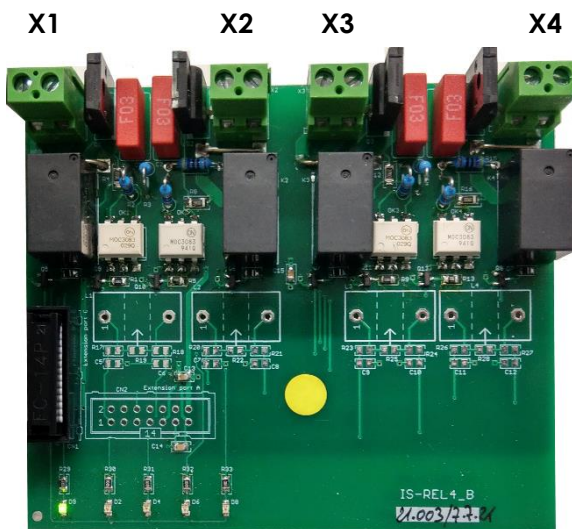
Data LED

Daten werden zum MQTT Broker gesendet
oder vom MQTT Broker empfangen

Taster Factory Default

Lädt Werkseinstellungen

(siehe Seite 11 „[Werkseinstellungen laden](#)“)



D9 D2 D4 D6 D8

Klemmen X1-X4

Relais 1-4

LED D9

Betriebsanzeige der Relais-Platine

LEDs D2, D4, D6, D8

Status-Anzeige der Relais 1-4

AN: Relais geschaltet (geschlossen)

AUS: Relais nicht geschaltet (offen)

LED D2 gehört zu Relais 1

LED D4 gehört zu Relais 2

LED D6 gehört zu Relais 3

LED D8 gehört zu Relais 4

ACHTUNG: Im Auslieferungszustand des net io – Rel 4 ist die Betriebsart für das Schalten von Wechselspannung aktiviert! Um Gleichspannung zu schalten, muss die Betriebsart umgestellt werden (siehe Seite 9 „[Betriebsarten](#)“).

Konfiguration (Webinterface)

ACHTUNG: Das net io – Rel 4 wird in der Werkseinstellung mit der IP-Adresse 192.168.1.100 ausgeliefert. Die IP-Adresse sollte nicht auf der Werksadresse bleiben. Bevor dieses oder auch mehrere dieser Geräte in das Netzwerk eingebunden werden, ist Folgendes sicher zu stellen:

- die werkseitig konfigurierte IP-Adresse ist nicht von einem anderen im Netzwerk vorhandenen Gerät belegt
- bei der Inbetriebnahme mehrerer net io – Rel 4 sind diese sequenziell mit unterschiedlichen Adressen in das Netzwerk einzubinden

Die Konfiguration des net io - Rel 4 erfolgt per HTTP über den Browser (z.B. Firefox, Chrome, Opera) und der entsprechenden IP-Adresse. Die Browser Microsoft Internet-Explorer und Microsoft Edge werden nicht unterstützt.

Der PC von dem aus die Konfiguration vorgenommen wird, muss sich im selben Subnetz wie das Gerät befinden. In der Werkseinstellung geben Sie folgende URL in die Adresszeile Ihres Browsers ein:

<http://192.168.1.100>

In der Werkseinstellung werden nun die folgenden Login-Daten eingegeben. Es wird dringend empfohlen, das Passwort nach erfolgreichem Login umgehend zu ändern!

Benutzer: **admin**
Passwort: **admin**

Nach dem Login wird die Eingabemaske für die Konfiguration des net io - Rel 4 angezeigt. Die Felder sind mit den aktuell konfigurierten Daten gefüllt. Ist dies nicht der Fall, verwenden Sie bitte einen alternativen Browser!

netIO-Rel4 2.3.2

Network

IP
192.168.1.100

Netmask
255.255.255.0

Gateway
192.168.1.254

DNS
192.168.1.254

Security

Password

Device

MQTT Broker
192.168.1.1

Device Name
devId_f950b4

Save

Command

Send

2022-10-04T12:09:13.845Z
> GetFirmwareVersion

2022-10-04T12:09:13.845Z
> GetConfig

2022-10-04T12:09:13.845Z
Connected

2022-10-04T12:09:13.858Z
< FirmwareVersion/netIO-Rel4 2.3.2

2022-10-04T12:09:13.862Z
<
Config/{"deviceName":"devId_f950b4","MAC":"04-91-62-f9-50-b4","ip":"192.168.1.100","netmask":"255.255.255.0","gateway":"192.168.1.254","dns":"192.168.1.254","mqttBroker":"192.168.1.1"}

IP* (default: 192.168.1.100)

IP-Adresse des Gerätes. Jedes Gerät benötigt eine einmalige Adresse, die sich im selben Subnetz wie der MQTT Brokers befindet und noch nicht in diesem Subnetz durch ein anderes Gerät belegt ist!

Beispiel für ein Klasse C Subnetz:

MQTT Broker hat die IP-Adresse 192.168.1.1

Das net io – Rel 4 muss eine freie IP-Adresse zwischen 192.168.1.2 und 192.168.1.254 mit der Subnetzmaske 255.255.255.0 bekommen.

Netmask* (default: 255.255.255.0)

Subnetzmaske des Netzwerkes.

Gateway (default: 192.168.1.254)

Standardgateway des Netzwerkes. Kann leer bleiben, wenn nicht benötigt.

DNS (default: 192.168.1.254)

DNS-Server des Netzwerkes. Kann leer bleiben, wenn nicht benötigt.

Passwort (default: admin)

Passwort des Benutzers „admin“ für das Webinterface

MQTT Broker* (default: 192.168.1.1)

IP-Adresse des Servers, auf dem sich der Broker befindet. Zum Beispiel die iSMaster-Software inkl. dem Modul „netIOMaster“ (siehe Seite 8 „[Installation des netIOMaster auf dem iSMaster](#)“).

Device Name* (default: „devID_“ + die letzten 6 Stellen der MAC-Adresse)

Name des Gerätes, mit dem es sich am MQTT Broker anmeldet und in der Programmierung der iSMaster-Software angesprochen wird. Verwenden Sie daher für jedes Gerät einen eindeutigen und aussagekräftigen Namen!

ACHTUNG: Der Device Name darf nur Buchstaben, Zahlen und Unterstriche enthalten!

Save

Übernimmt die geänderten Einstellungen und startet das Gerät neu. Die Einstellungen sind nach wenigen Sekunden aktiv, ohne das Gerät spannungsfrei zu schalten.

*** Pflichtfelder**

ACHTUNG: Die Eingabefelder werden nicht auf Fehler überprüft! Bei Fehleingaben ist das Gerät unter Umständen nicht mehr erreichbar und muss auf Werkseinstellungen zurückgesetzt werden (siehe Seite 11 „[Werkseinstellungen laden](#)“).

Installation des netIOMaster auf dem iSMaster

Das net io - Rel 4 ist grundsätzlich mit allen MQTT Brokern ab v3.1.1 kompatibel.

Integrierter MQTT Broker

Das Modul „netIOMaster“ beinhaltet einen integrierten MQTT Broker. Das Modul startet automatisch den MQTT Broker (mosquitto.exe) und stellt die Schnittstelle zum iSMaster her.

Systemvoraussetzungen:

- Microsoft Visual C++ 2010 Redistributable Package (x86)
(vc_redist.x86.exe)
- .NET Framework 4.8
(dotNET4.8-x86-x64-allos-enu.exe)

Download

<https://media.insynergie.de/tools/insynergie-net-io-master.zip>

Installationsverzeichnis:

C:\Program Files (x86)\inSynergie\Module\netIOMaster

Nach der Installation der Systemvoraussetzungen wird das Modul „netIOMaster“ in das Installationsverzeichnis des iSMaster unter Module kopiert. Nach einem Neustart des SysClients auf dem System kann das Modul wie gewohnt über den Systemmanager des iSMaster gestartet werden.

Bei aktivierter Firewall muss der Zugriff für die „netIOMaster.exe“ und „mosquitto.exe“ ein- und ausgehend erlaubt werden!

Externer MQTT Broker

Das Modul „netIOMaster“ kann auch für einen externen MQTT Broker die Schnittstelle zum iSMaster herstellen. Dazu wird die IP-Adresse des externen Brokers in der Datei „netIOMaster.exe.config“ eingetragen (default value: 127.0.0.1).

```
<setting name="MQTTBroker" serializeAs="String">  
  <value>127.0.0.1</value>  
</setting>
```

Um die Änderung zu aktivieren, muss das Modul „netIOMaster“ neu gestartet werden.

Bei aktivierter Firewall muss der Zugriff für die „netIOMaster.exe“ ein- und ausgehend erlaubt werden!

ACHTUNG: Im Interact-Editor des iSMaster wird das Modul erst sichtbar, wenn sich mindestens ein physikalisches net io - Rel 4 mit dem Broker verbunden hat!

Betriebsarten

Das net io - Rel 4 unterstützt ab Firmware-Version 1.0.0 zwei Betriebsarten, die sich durch den Befehl „SetDCMode“ für jedes Relais unabhängig einstellen lässt. Im Werkzustand ist für alle Relais der AC Modus aktiviert.

AC Modus

Die Relais können zum Schalten von vier getrennten Lastkreisen mit je 230V/8A genutzt werden. Um ein „Kleben“ des Lastrelais zu verhindern wird der Einschaltstrom zunächst durch einen vorgeschalteten Triac abgefangen.

DC Modus

Die Relais werden in diesem Modus ohne den vorgeschalteten Triac direkt geschaltet und können somit zum Schalten von Gleichspannung betrieben werden.

Betriebsmodus ändern

DCMode/[INTEGER]/[BOOL]

Die INTEGER Zahl steht für die Relais (1-4)
Beispiel DCMoDe/[1]/[FALSE]
Hier wird das erste Relais auf AC gestellt
Der BOOL Wert steht für den Zustand.
FALSE steht bei diesem Befehl für AC und
TRUE für DC.

Um den Betriebsmodus auf DC zu stellen muss man in dem iSMaster-Skript folgenden Befehl ausführen:

Beispiel: Relais1 auf DC setzen

```
DCMode/1/TRUE
```

Beispiel: Relais1 wieder auf AC setzen

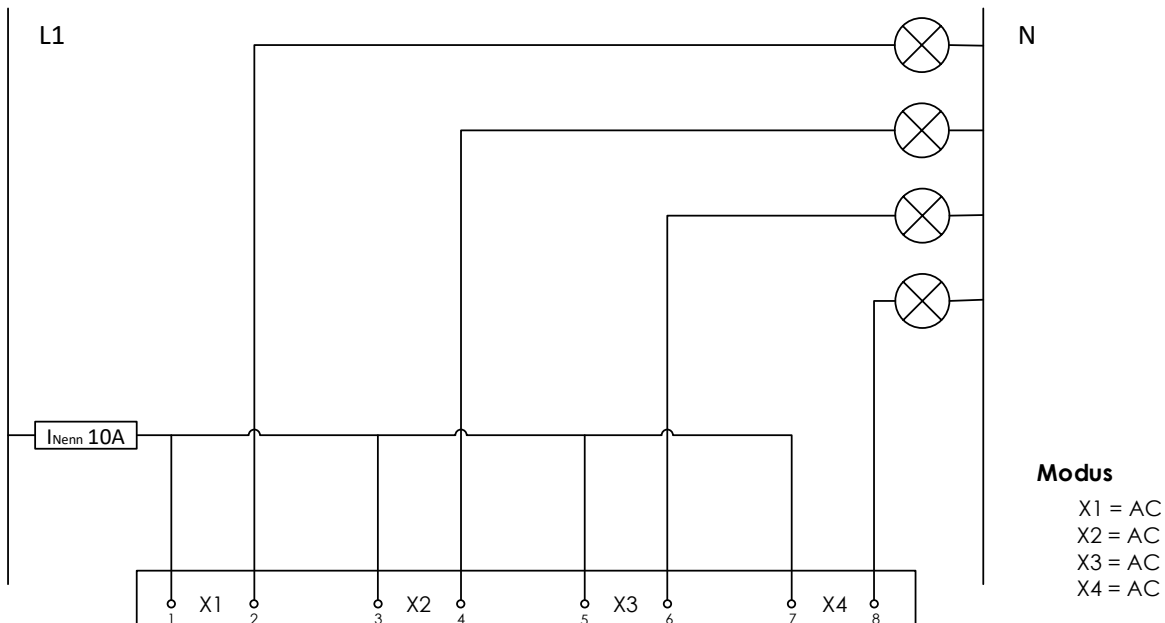
```
DCMode/1/FALSE
```

Verdrahtungsbeispiele

Beispiel 1: 4 Verbraucher an 230V

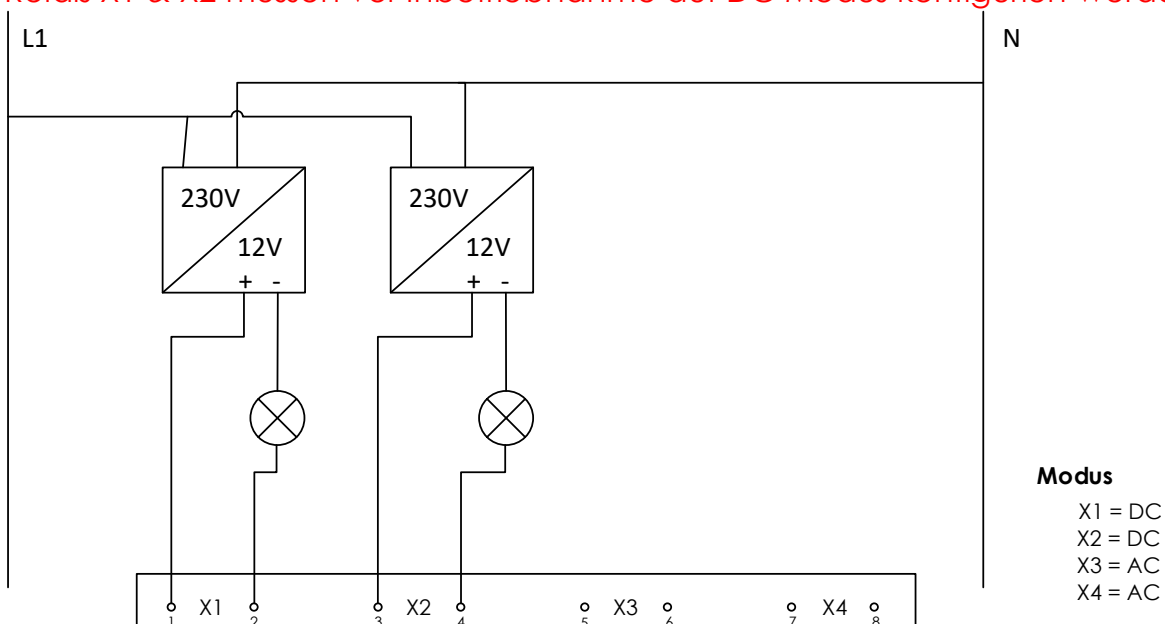
$I_{\text{nenn}} < 10\text{A}$

$I_{\text{max}} < 140\text{A}; < 20\text{ms}$



Beispiel 2: Zwei Verbraucher mit 12 Netzteil

Relais X1 & X2 müssen vor Inbetriebnahme auf DC Modus konfiguriert werden!



Werkseinstellungen laden

Sollten Sie die IP-Adresse oder die Login-Daten für das Webinterface vergessen haben, können Sie das net io - Rel 4 auf die Werkseinstellungen zurücksetzen.

Dazu wird das Gerät vom Netzwerk getrennt, damit es spannungsfrei ist (PoE). Während das net io - Rel 4 wieder mit dem Netzwerk verbunden wird, muss der Taster „Factory Default“ gedrückt gehalten werden, bis die rote LED „Data“ aufhört zu leuchten.

Anschließend ist das Gerät wieder unter der IP-Adresse 192.168.1.100 erreichbar und kann neu konfiguriert werden (siehe Seite 5 „[Konfiguration](#)“).

Kommunikation

Befehle

```
// System
EnableIdentificationMode      = "EnableIdentificationMode\x0d\x0a";
DisableIdentificationMode    = "DisableIdentificationMode\x0d\x0a";
GetFirmwareVersion           = "GetFirmwareVersion\x0d\x0a";
GetConfig                    = "GetConfig\x0d\x0a";
GetMAC                       = "GetMAC\x0d\x0a";
ChangeName/[STRING]         = "ChangeName/\ 1a00\x0d\x0a"; p1=new name;

// Relay
GetStateOfAllRelais          = "GetStateOfAllRelais\x0d\x0a";
InvertAllRelais              = "InvertAllRelais\x0d\x0a";
GetStateOfRelais/[INTEGER]   = "GetStateOfRelais/\ 1a00\x0d\x0a"; p1=relais nummer (1-4);
SetRelais/[INTEGER]          = "SetRelais/\ 1a00\x0d\x0a"; p1=relais nummer (1-4);
ResetRelais/[INTEGER]        = "ResetRelais/\ 1a00\x0d\x0a"; p1=relais nummer (1-4);
SetRelaisState/[INTEGER]/[BOOL] = "SetRelaisState/\ 1a00/\ 2a00\x0d\x0a"; p1=relais nummer (1-4), p2=relais state;
SetAllRelaisState/[BOOL]     = "SetAllRelaisState/\ 1a00\x0d\x0a"; p1=relais state;
SetDCMode/[INTEGER]/[BOOL]   = "SetDCMode/\ 1a00/\ 2a00\x0d\x0a"; p1=relais nummer (1-4), p2=DC mode;
GetDCMode/[INTEGER]         = "GetDCMode/\ 1a00\x0d\x0a"; p1=relais nummer (1-4);
TestAllRelais                = "TestAllRelais\x0d\x0a";
TestRelais/[INTEGER]        = "TestRelais/\ 1a00\x0d\x0a"; p1=relais nummer (1-4);
```

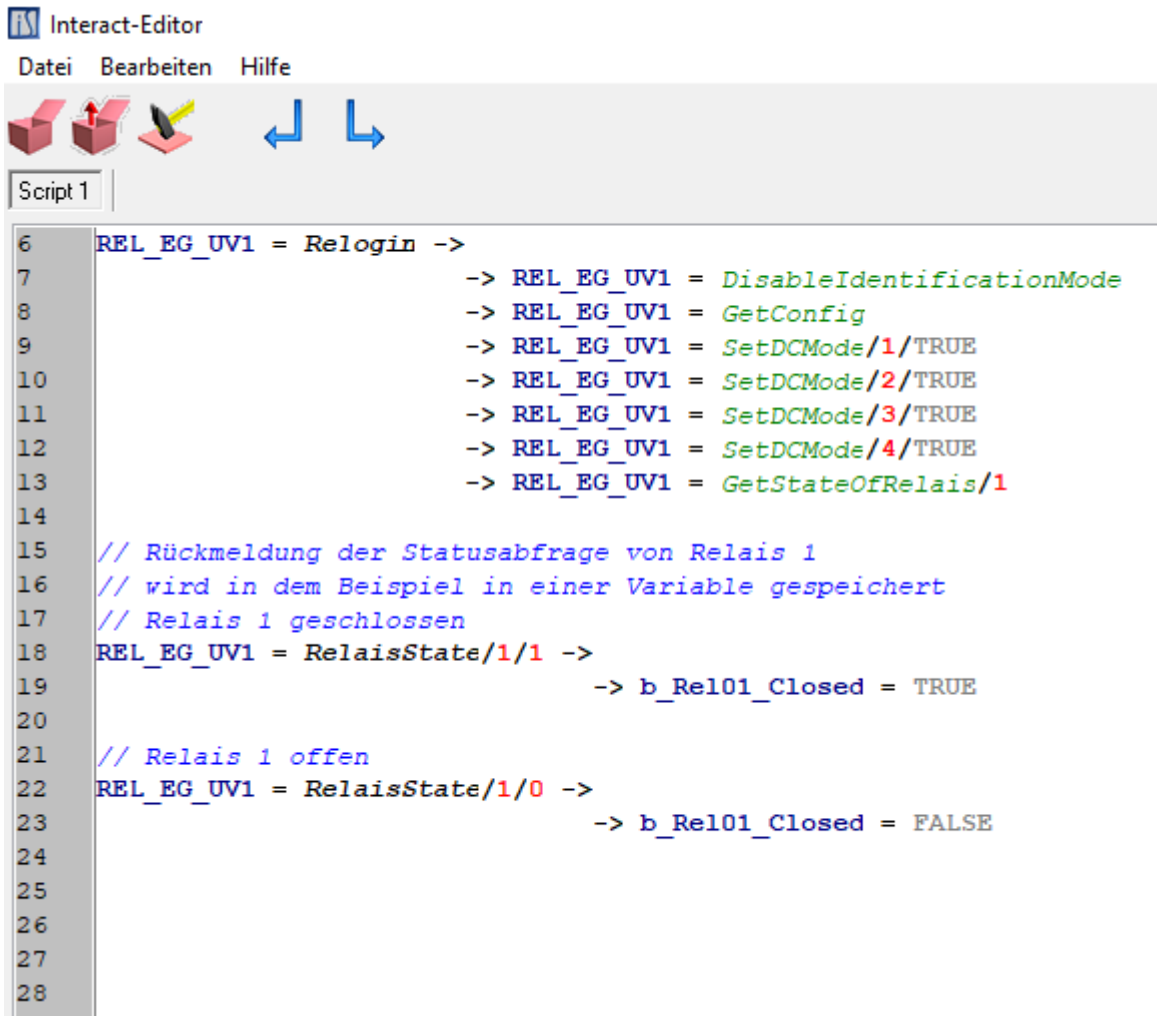
Antworten

```
// System
FirmwareVersion/[STRING]    = "FirmwareVersion=\ 1a00\x0d\x0a"; p1=firmware version;
Config/[STRING]             = "Config=\ 1a00\x0d\x0a"; p1=config (as JSON);
MAC/[STRING]                = "MAC=\ 1a00\x0d\x0a"; p1=MAC;
Error/[STRING]              = "Error=\ 1a00\x0d\x0a"; p1=error message;

// Relay
AllRelaisState/[STRING]     = "AllRelaisState=\ 1a00\x0d\x0a"; p1=relais states;
RelaisState/[INTEGER]/[BOOL] = "RelaisState\ 1a00=\ 2a00\x0d\x0a"; p1=relais nummer (1-4), p2=relais state;
DCMode/[INTEGER]/[BOOL]     = "DCMode\ 1a00=\ 2a00\x0d\x0a"; p1=relais nummer (1-4), p2=DC mode;
```

Scriptbeispiel

Die vom MQTT Broker gelieferten Signale über die Scriptsprache in der iSMaster-Software verarbeiten:



The screenshot shows the 'Interact-Editor' window with a menu bar (Datei, Bearbeiten, Hilfe) and a toolbar. The main area displays a script for 'Script 1' with the following content:

```

6 REL_EG_UV1 = Relogin ->
7     -> REL_EG_UV1 = DisableIdentificationMode
8     -> REL_EG_UV1 = GetConfig
9     -> REL_EG_UV1 = SetDCMode/1/TRUE
10    -> REL_EG_UV1 = SetDCMode/2/TRUE
11    -> REL_EG_UV1 = SetDCMode/3/TRUE
12    -> REL_EG_UV1 = SetDCMode/4/TRUE
13    -> REL_EG_UV1 = GetStateOfRelais/1
14
15 // Rückmeldung der Statusabfrage von Relais 1
16 // wird in dem Beispiel in einer Variable gespeichert
17 // Relais 1 geschlossen
18 REL_EG_UV1 = RelaisState/1/1 ->
19     -> b_Rel01_Closed = TRUE
20
21 // Relais 1 offen
22 REL_EG_UV1 = RelaisState/1/0 ->
23     -> b_Rel01_Closed = FALSE
24
25
26
27
28

```

In diesem Beispiel ist der Name des net io - Rel 4 "REL_EG_UV1". Den Namen legt man über das Webinterface unter „Device Name“ fest hat (siehe Seite 5 „[Konfiguration](#)“).

Technische Daten

Abmessungen:

- 106 x 90 x 57 mm
- 6 TE Hutschienengehäuse zur Montage auf DIN Schiene

Mikrocontroller

- 32 MHz CPU
- 32 kb SRAM
- 256 kb Flash

Protokolle

- TCP, UDP, MQTT, Web Sockets
- HTTP (integrierter Webserver zur Konfiguration)

Relais:

- 4 Relais
- AC oder DC Modus für jedes Relais separat konfigurierbar
- Jedes Relais mit separater Statusanzeige (LED)

Netzwerk

- 10/100BaseT autosensing
- IEEE 802.3af kompatibel

Stromversorgung

- Power-over-Ethernet (PoE)

Anschlüsse

- 4 x Schraubklemme (AKZ 350)
- 1 x RJ45 für Netzwerk