

# IP Relay HWg-ER02b

## Manual



EN 600 552

# Package contents

A complete shipment contains the following items:

- IP Relay HWg-ER02b
- Printed manual + datasheet

# Safety information

The device complies with regulations and industrial standards in force in the Czech Republic and the European Union. The device has been tested and is supplied in working order. To keep the device in this condition, it is necessary to adhere to the following safety and maintenance instructions.

**Never remove the device cover if the relay terminals are connected to the electrical network!**

**Using the device in a manner other than prescribed by the manufacturer may cause its safeguards to fail!**

**The power supply outlet or disconnection point must be freely accessible.**

**The device must not be used in particular under any of the following conditions:**

- The device is noticeably damaged
- The device does not function properly
- Unfastened parts can move inside the device
- The device has been exposed to moisture or rain
- The device has been serviced by unauthorized personnel
- The power adapter or power supply cable are noticeably damaged
- If the device is used in a manner other than designed for, the protection provided by the device may fail.
- The local electrical system must include a power switch or a circuit breaker and overcurrent protection.

The manufacturer warrants the device only if it is powered by the supplied power adapter or an approved power supply.

If you have any problems with installing or operating the device, please contact technical support:

HW group s.r.o.  
<http://www.hw-group.com>  
Email: [support@HWg.cz](mailto:support@HWg.cz)

U Pily 3  
143 00 Praha 4  
Czech Republic  
Tel. +420 222 511 918

When contacting technical support, please keep at hand the exact type of your device (at the type plate) and, if possible, the firmware version (see later in this manual).

# IP Relay HWg-ER02b

## *Ethernet relay, DIN rail, Web interface.*

The Ethernet relay features 2x DI (dry contact input), a full RS-232/485 serial port, and 2x DO (relay output, 230V/16A). It is controlled over the web or Modbus/TCP.

Relay outputs can directly switch 230V/16A AC mains, dry contact inputs can be used e.g. for buttons. The device features a full RS-232/485 serial port for connecting e.g. a RFID or barcode reader. Industrial model, fits on a DIN rail.

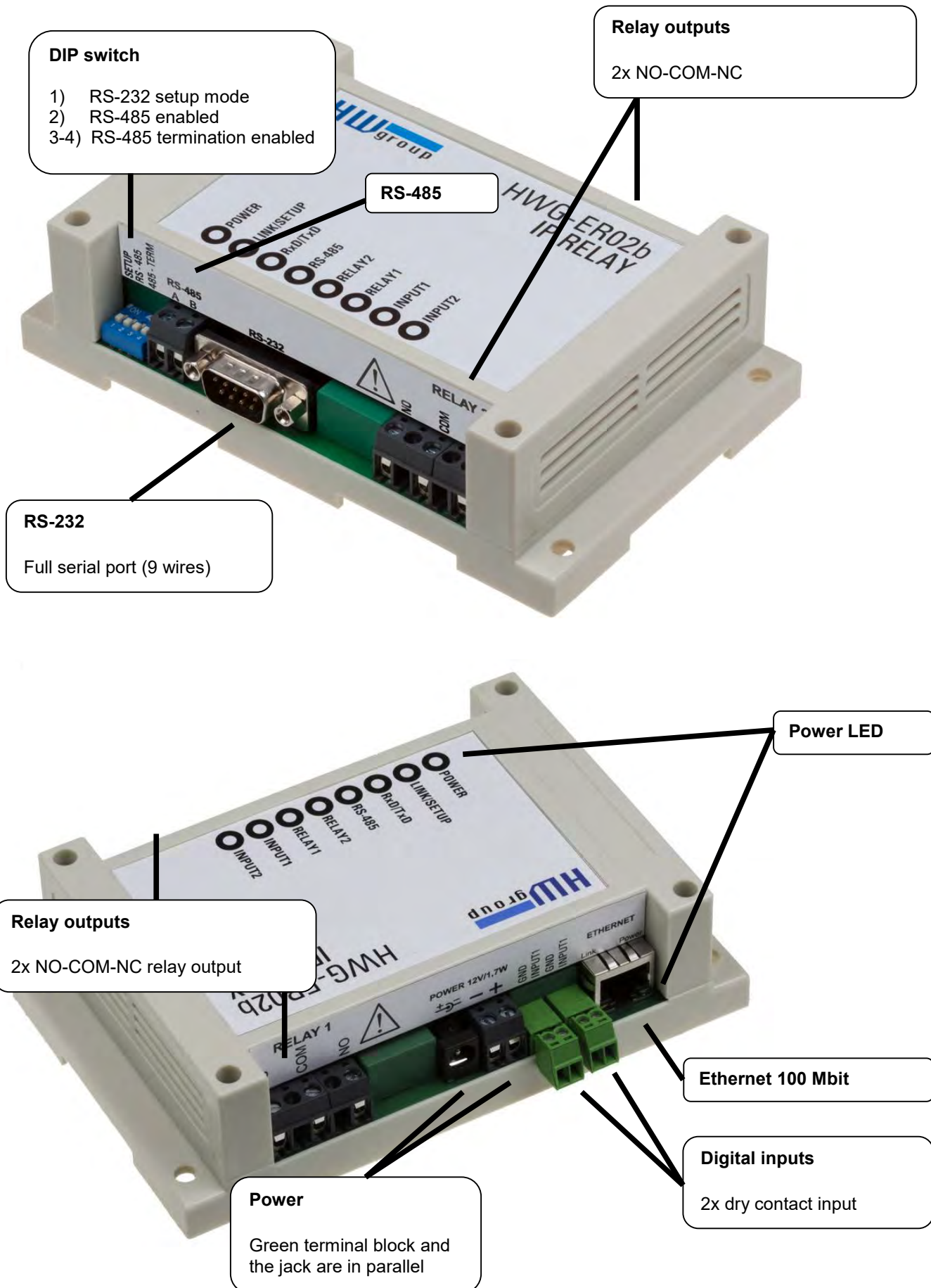
Two devices can be connected against each other (Box-2-Box mode) to extend the digital and RS232 signals over a computer network.



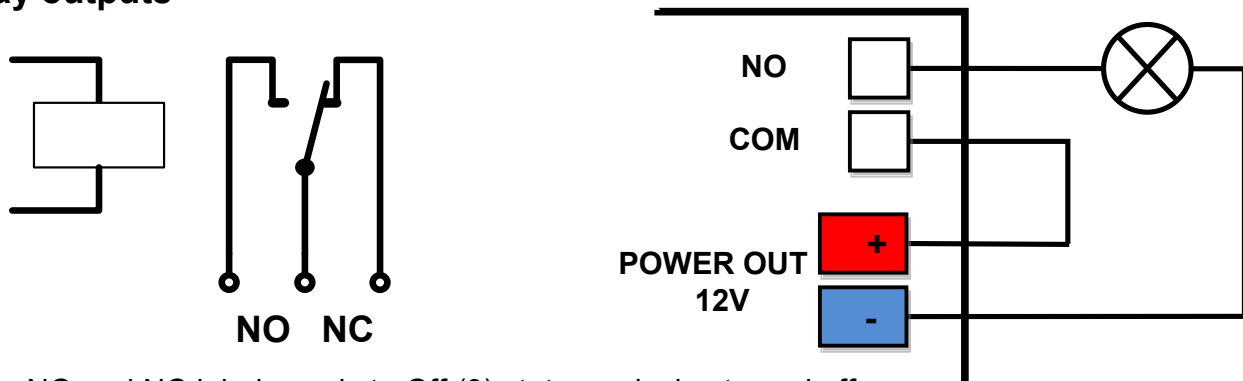
### Basic features

- **Ethernet:** RJ45 (100BASE-T / 100 Mbit/s)
- **Inputs:** 2 digital (dry contact) inputs
- **Outputs:** 2 NO/NC digital (relay) outputs, rated 230V/16A
- **Serial interface:** 1x full RS-232 or RS-485 serial port
- **Full RS-232 serial port:** Rx/D, Tx/D, GND, CTS, RTS, DSR, DTR, RI, CD / 115.200 Bd.
- **Configuration:** Web, Telnet
- **Box-2-Box mode:** Two units can be connected back-to-back over the network and tunnel the I/O and the serial port.
- **M2M protocols:** http, Modbus/TCP, Telnet, TCP Server, TCP Client/Server
- **Support for programmers:** HWg-SDK
- **Mechanical:** 145 x 90 x 45 mm, wall-mountable, fits on a DIN rail
- **Free HW VSP software:** Virtual serial port for Windows (NT service, x64 systems). Supports up to 100 remote serial ports. HW VSP is the world's most popular virtual serial port for Windows!
- **Serial interface features:**
  - Communication speed configurable from 300...115200 Bd
  - Handshake (CTS/RTS, Xon/Xoff, none)
  - Full serial port (Cannon DB9M - Rx/D, Tx/D, CTS, RTS, DSR, DTR, RI, CD, GND)
  - Support for 7th to 9th parity bit (9th parity bit transferred over the Ethernet)
  - Remote serial port is RFC2217 compatible
  - Support for TCP/IP terminal, TELNET - NVT type (Network Virtual Terminal)

# Product description



## Relay outputs

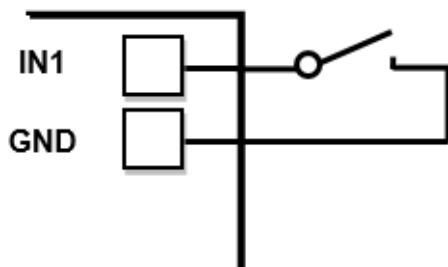


- NO and NC labels apply to Off (0) state, or device turned off
- When the output is On (1), a “**Normally Open**” (NO) relay contact is closed
- **LEDs:** Closed contact is indicated by a LED
- **Isolation:** The double-throw contact is electrically isolated from the rest of the device
- **ID range:** Outputs use ID addresses from 1 to 2

## Digital (dry contact) inputs

Digital input terminals may be connected to voltage-free contacts or the GND pin. The inputs are electrically connected to the 12V power supply. Never connect the inputs to 48V!

- Unconnected inputs read as “**0 (Off)**”.
- Active inputs read as “**1 (On)**”.
- **Supported sensors:** Any contact without external voltage (dry contact)
- **Polling period:** 200 ms
- **Range of sensor IDs:** Inputs use ID addresses from 1 to 2

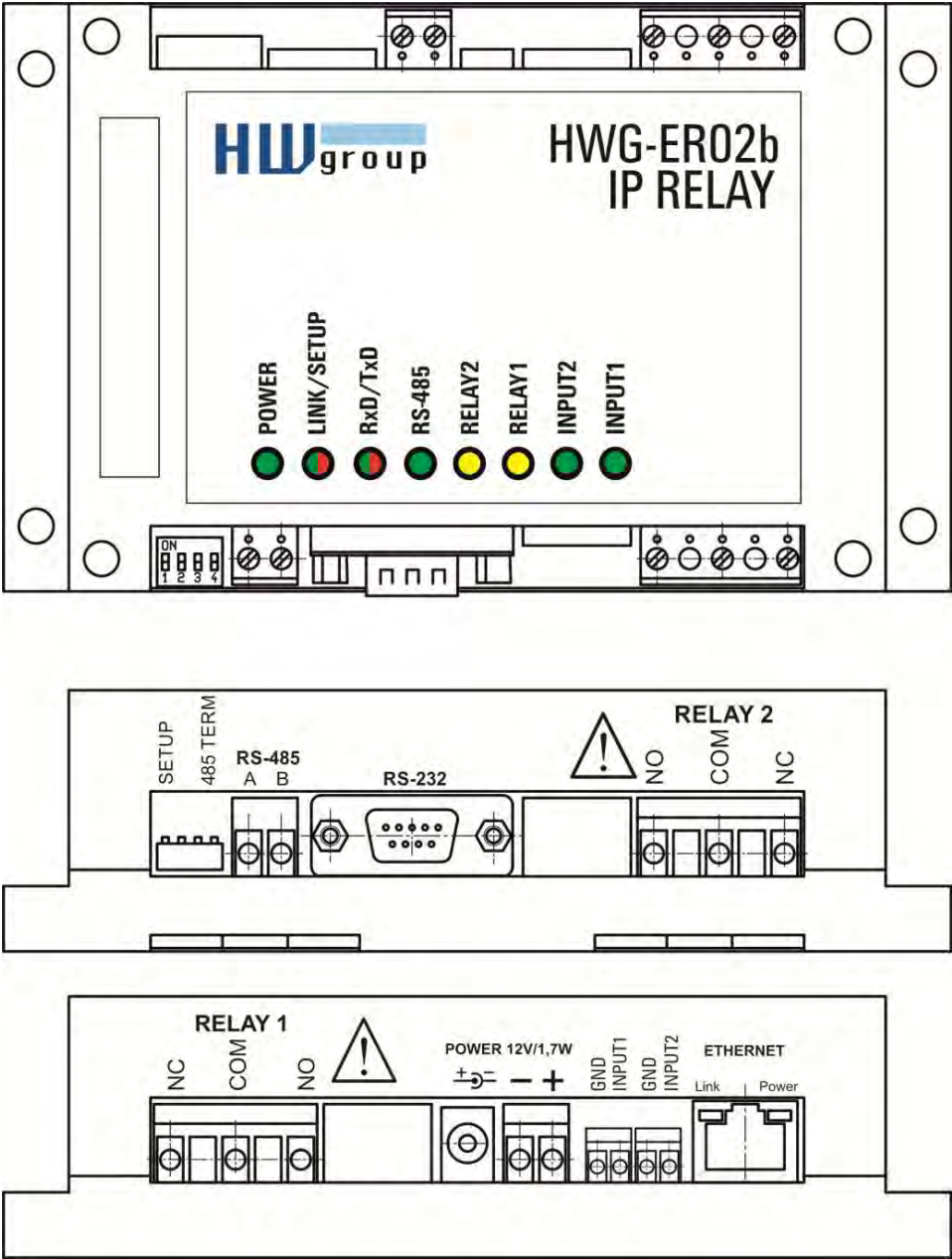


# Technical specifications

Ethernet port	
Interface	RJ45 (100BASE-Tx)
Supported protocols	IP: ARP, TCP/IP, (HTTP, Modbus over TCP, NVT, RFC2217), UDP/IP
Network modes	TCP/IP server, TCP/IP client/server, UDP/IP
Serial port	
RS-232 connector	Cannon 9 - DB9M
RS-232 interface	RxD, TxD, RTS, CTS, DTR, DSR, CD, RI, GND (full serial port)
Data bits / Stop bits	7, 8 or 9 / 1 or 2
Speed / Parity	50 – 115 200 Bd / Parity options: None / Odd / Even / Mark / Space
RS-485 connector	2-pin terminal block (A > B)
RS-485 isolation	RS-485 not optocoupled to the device's power supply Galvanic isolated RS-232/485 to Ethernet (1.000 V)
RS-485 termination	Jumper DIP3+4 (2x On = 120Ω for A-B, 2x 470Ω to 5V power)
Digital Inputs	
Type	2x Dry Contact
Sensitivity / Max. distance	1 (On) = 0 – 500 Ω / up to 50m
Digital Outputs	
Type	2x Relay contacts (NC-COM-NO) – Power up state (no state restart memory)
Max. load	max. 250V / 16A AC / max. 24V / 16A DC
Environment parameters	
Temperature	Operating: -5 to 75 °C (23 to 167 °F) / Storage: -25 to 85 °C (-13 to 185 °F)
Relative humidity	5 to 95 % (non-condensing)
LEDs	
+ POWER (green)	GREEN – power supply turned on
+ LINK & Activity (green)	GREEN - Ethernet connection
+ RxD/TxD (green/red)	Green – RS-232/RS485 receiving activity Red – RS-232/RS485 transmission activity
+ RS-485 (green)	RS-485 Active
+ Relay 1, 2 (yellow)	Relay closed (1)
+ Digital Inputs (green)	Digital Input closed (1)
DIP switches	
DIP1 - RS-232 Setup	ON = RS-232 Setup mode (9600 8N1) – Ethernet interface disabled! OFF = Ethernet mode
DIP2	Enable RS-485 (RS-232 disabled)
DIP3, DIP4	RS-485 termination
Other parameters	
Supply voltage	12-30 V/ 150 mA DC (coaxial connector (barrel) & terminal block in parallel)
Dimensions / Mass	145 x 90 x 45 [mm] / 225 g
Mount	Wall / DIN rail
EMC	FCC Part 15, Class B, CE - EN 55022, EN 55024, EN 61000

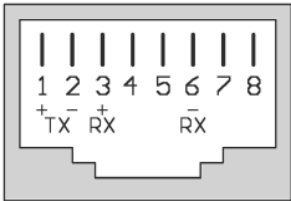


Connectors



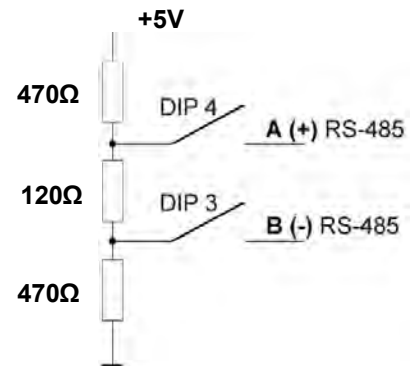
RS-232			
1	CD	<--	Carrier Detect
2	TxD	-->	Transmit Data
3	RxD	<--	Receive Data
4	DTR	-->	Data Terminal Ready
5	GND	---	System Ground
6	DSR	<--	Data Set Ready
7	CTS	<--	Clear to Send
8	RTS	-->	Request to Send
9	RI	<--	Ring Indicator

RS-485			
1	A	<->	A > B
2	B	<->	A > B



DIP 1 SETUP	DIP 2 RS-485	Function
ON	OFF	<b>RS-232 Setup</b> mode ( <u>9600 8N1</u> ) <i>Ethernet is disabled</i>
OFF	OFF	Serial port in the <b>RS-232</b> mode
OFF	ON	Serial port in the <b>RS-485</b> mode ( <i>RS-232 is inactive</i> ) - check the &R and &H parameters (recommended: &R3 &H1)

DIP 3	DIP 4	Function
OFF	OFF	RS-485 unterminated
ON	ON	RS-485 terminated



### LEDs:

**Power** – green ..... External power connected

**LINK** – amber ..... Ethernet interface activity

**Status** – green ..... Device starting up, firmware upgrade

**Setup** – red ..... Device in serial Setup mode

**INx** – green ..... Input x is closed

**OUTx** – red ..... Output x is closed



# How to use the device

## Configuring the IP address

- 1) Install the HWg-Config (**UDP Config**) utility [at our website or on the DVD].
- 2) Set **DIP1** and **DIP2** to **OFF**.
- 3) Connect the device to the Ethernet.
- 4) Connect the power to the device (plug in the adapter).
- 5) If the power is OK, the green **Power** LED lights up.
- 6) If Ethernet works, the **LINK** LED lights up, and blinks when data is transferred.
- 7) Run **HWg-Config** (or the older version, UDP config) and configure:

**UDP Config 4.11.3 for HW group products (www.hw-group.com)**

Version: 4.11.3 HW group  
www.hw-group.com  
Config utility for the HW group devices

Your PC network settings  
IP address: 192.168.2.39  
Netmask: 255.255.252.0  
Gateway: 192.168.1.254

**Your LAN settings**

Find Devices

Device list:

MAC	Name	* IP	Device type	Port	Parameters
00:0A:59:02:07:00		192.168.1.41	IP Relay2	23	TCP setup=Y, TEA=N, NVT=Y

**1** Double-click to configure the device

**2** This device does NOT support DHCP. Manually configure:

- IP address
- Subnet mask
- Gateway

Click "Apply Changes" to save the changes.

**Note:**

If the device is not visible on your PC, it may be connected to a different LAN segment.

Connect the LAN cable from the device to the same switch where your PC is connected.

**3** Open in the web browser

Click the underlined IP address to browse to the WEB interface.

**Details**

Name: IP address: 192.168.1.41 Port: 23

Open in WEB Browser

Mask: 255.255.255.0 MAC: 00:0A:59:02:07:00

Gateway: 192.168.1.254 FW version: 3.1.7

Device type: IP Relay2 (62)

DHCP: Not supported

☐ Enable IP access filter

IP filter value: 0.0.0.0

IP filter mask: 0.0.0.0

Default values: Load defaults

☒ Enable NVT

☒ Enable TCP setup

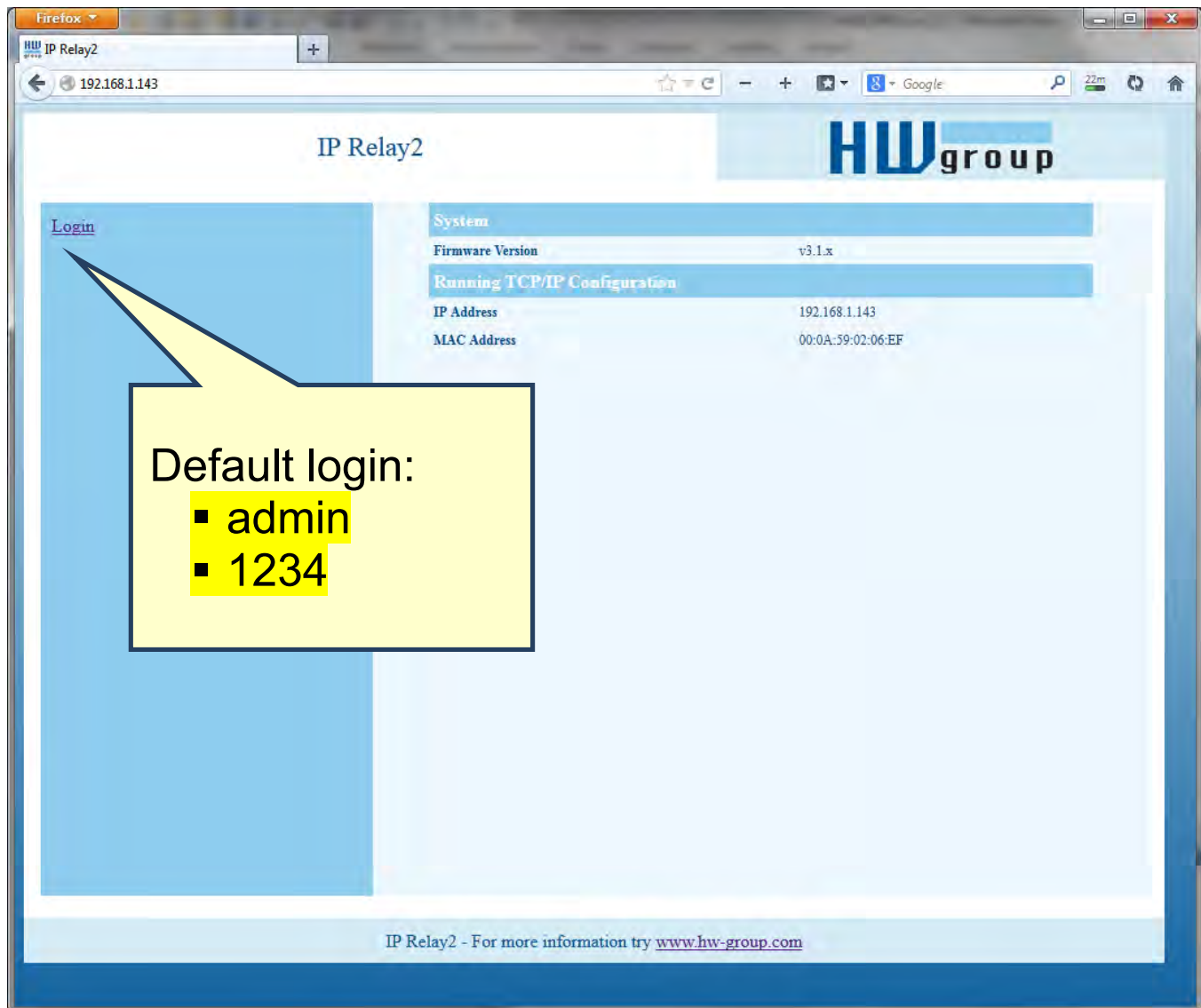
☐ Enable TEA authorisation

☒ Check if new IP address is empty

Cancel Apply changes

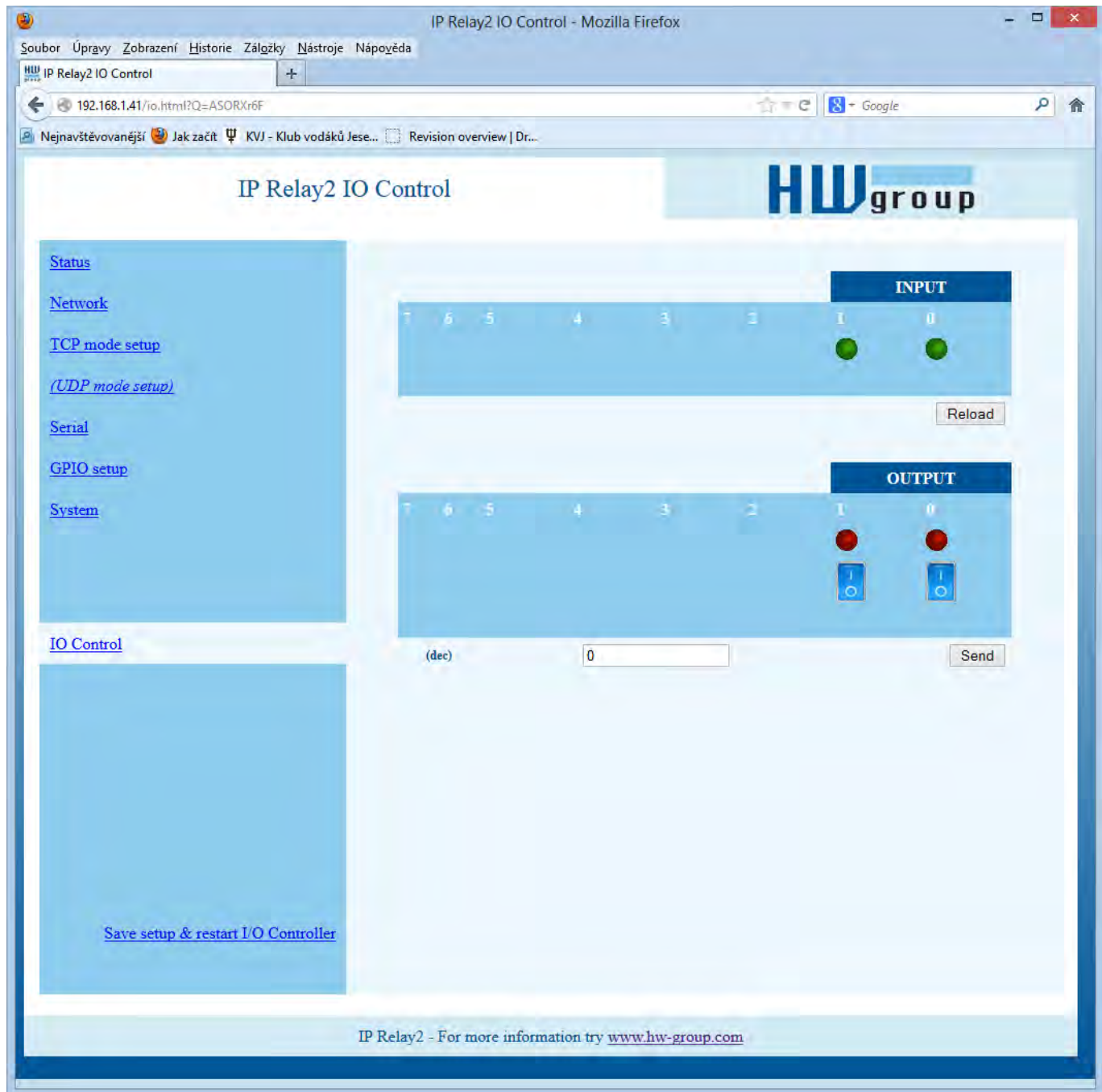
## Configuring the device over WWW

In the HWg-Config tab, click the correct MAC address and press **Open in the WEB browser**.



You need to log in to access the configuration.  
The default credentials are “**admin**” + “**1234**”.

## Controlling DI and DO over the WEB

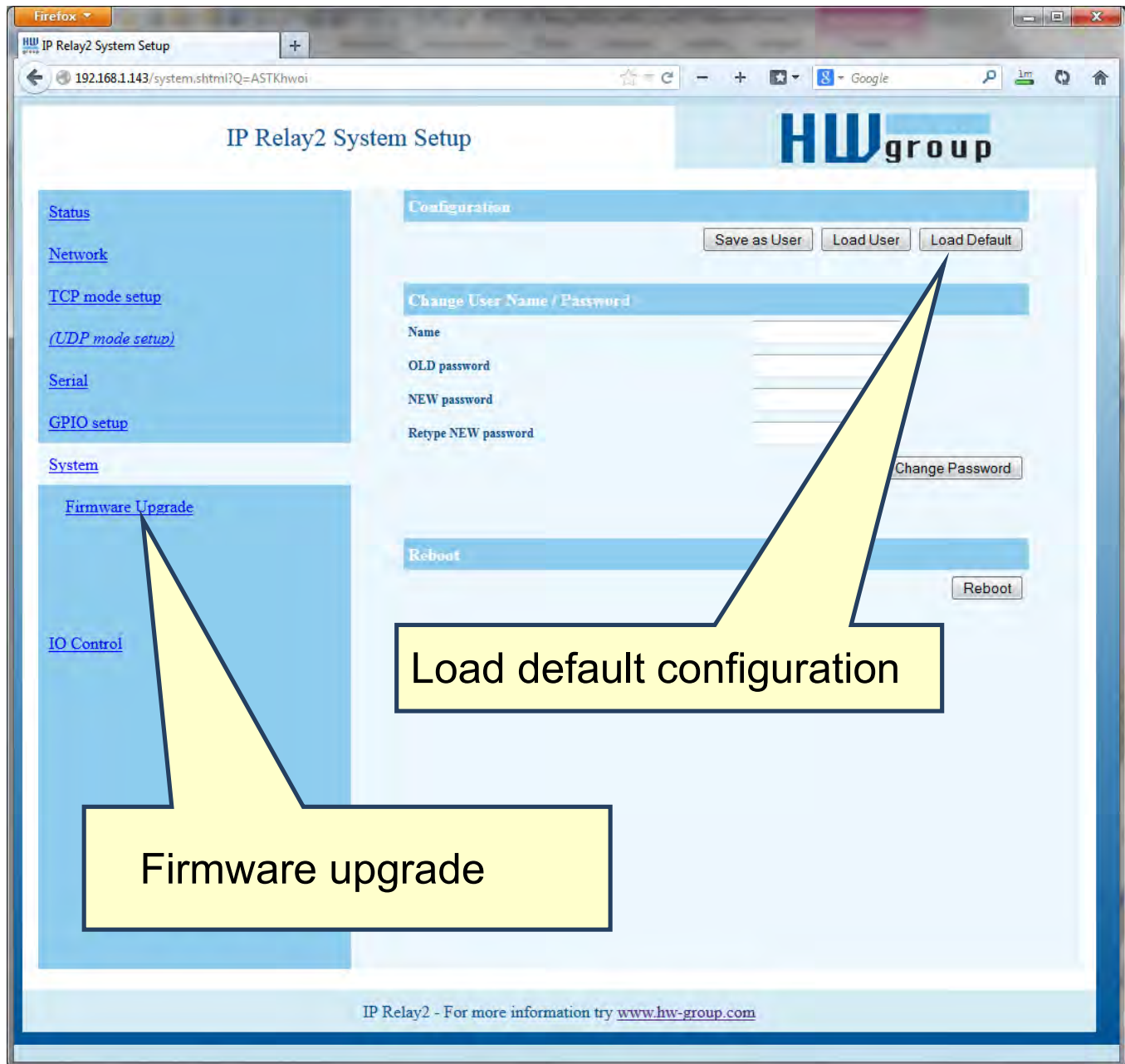


Relay outputs are controlled individually by clicking the switch icons, or by entering a decimal value and clicking Send.

## System + firmware upgrade

The System page lets you change the admin's password and back-up or restore configuration.

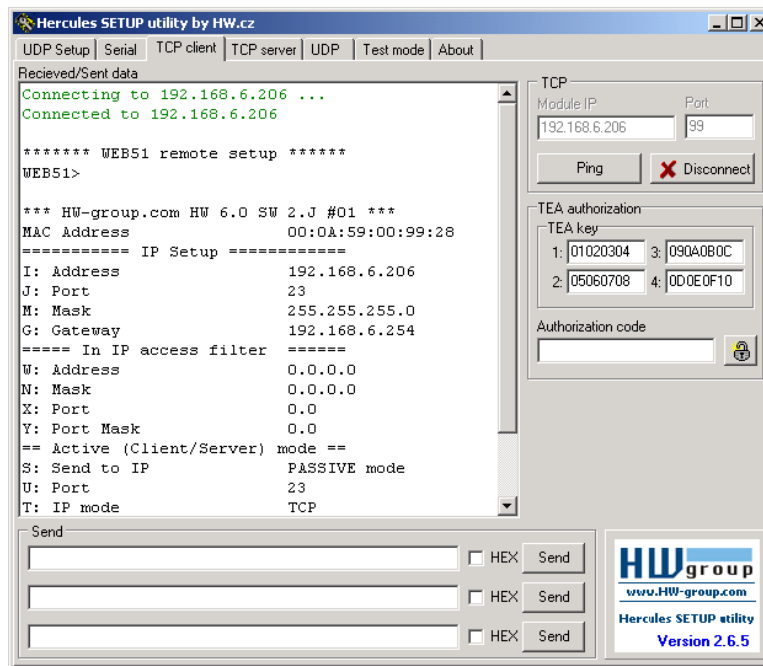
The Firmware Upgrade link in the left-hand menu allows you to upload firmware to the device over the web interface.



## Setting up without WWW (TCP Setup)

Use the Hercules utility:

- Change to the “**TCP Client**” tab and enter the configured IP address. Set the **TCP port** to **99**.
- Click “Connect”. The listing on the left displays the “**WEB51>**” prompt. Click inside this pane and press ENTER. The window displays the current configuration.
- To set a parameter, enter the corresponding letter and the desired value (for example “**I192.168.6.8**” to set the device IP address). To show the help for a command, enter the command and a question mark, and press ENTER – for example, “**I? <Enter>**”. All the settings are described in detail later in this manual.
- After setting the parameters, use R for **Reboot** and press “Disconnect” to disconnect from the device. The device restarts.
- **Note:** TCP Setup can be disabled with the “Enable TCP setup” option. When unchecked, the device refuses connections to port 99.

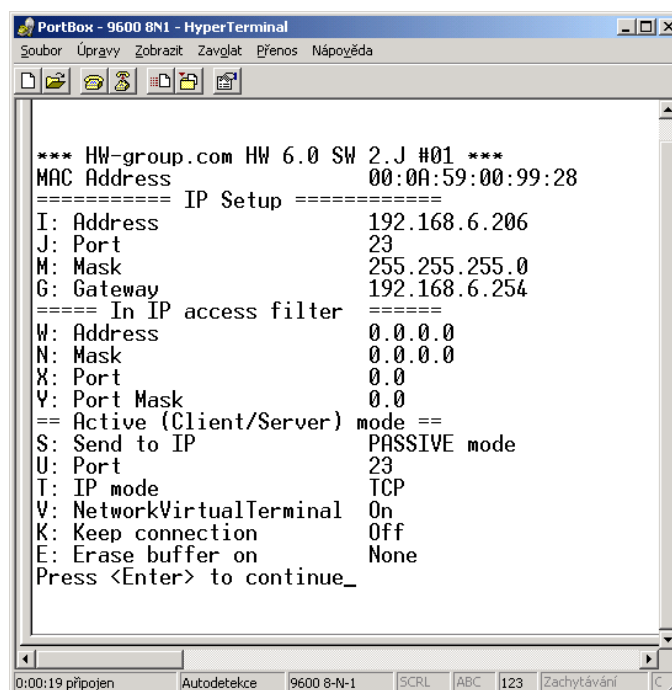


## Setting up the device using a RS-232 terminal

If you do not have Windows or our Hercules utility available, you can set up the device over RS-232 using any terminal.

- Set **DIP1 = ON**, **DIP2 = OFF**.
- Connect RS-232 (port 1) to a PC using the supplied **LapLink** cable.
- Open your favorite terminal program (e.g. Hyperterminal), choose the correct serial port and configure it to **9600 8N1**.
- Connect the power adapter to the electrical outlet and to the device.
- If the power is OK, the green **Power** LED should light up.
- If the serial cable and terminal program are working, the text menu for configuring the device appears.

Continue by following the steps as in the previous case.



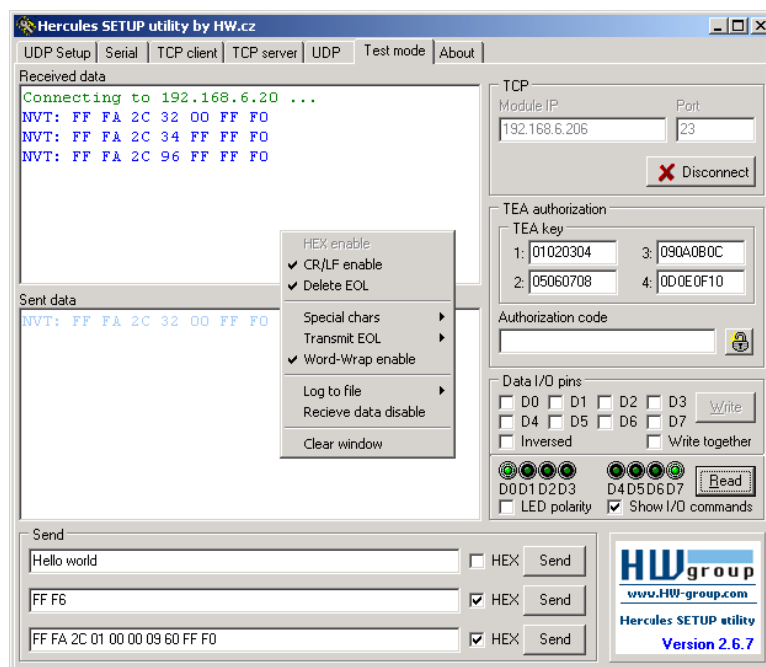


## Controlling I/O using NVT

Use the Hercules utility:

- Set **DIP1 = OFF**, **DIP2 = OFF**, connect the test board to the connector and turn on the device.
- Change to the “**UDP Setup**” tab, find the devices, and make sure that the “**Enable TCP setup**” and “**Enable NVT**” options in the lower left-hand portion next to the MAC list are checked for the respective device. If not, check both options and save your settings to the device.
- Double-click the IP address in the “UDP Setup” tab and change to the “**Test mode**” tab. The double-click action should transfer the IP address and port to the Test tab. If this does not happen, enter the values manually.
- Click “**Connect**”. The listing at the left-hand side displays Connecting, followed by one or three NVT commands in the received data. All recognized commands are displayed in blue.

The image shows the listing and the configuration options of the Hercules utility. To invoke the menu, right-click the receiving or transmitting pane.



- At this point, click “**Read**” in the lower right-hand corner to read the inputs (8 switches at the test board). Input states are indicated by the virtual LEDs D0 to D7, respectively. States can be inverted with the “LED polarity” option.
- The D0 – D7 checkboxes set the corresponding outputs. The first command sets all outputs to defined states, subsequent commands set individual bits. Outputs can be again inverted using the “Inversed” option. The “Write together” option does not send commands immediately when an output is selected; instead, outputs are set after clicking “**Write**”.
- When “Show I/O commands” is enabled, notice the IP Relay control sequences in the incoming and outgoing panels. This makes it easy to test the commands.
- Enter “FF F6” to the Send lines at the bottom, check HEX and click the corresponding Send button to send this command. The receiving pane should show “<WEB51 HW 4.7 SW 2.J SN 00A608 #01>” or something similar. You have just sent your first NVT command requesting “Are You There” identification, and the IP Relay replied with its HW and SW version. The SN number consists of the last three bytes of the MAC address.
- If the TCP connection is closed while working (red message “**Connection refused by remote host**”), click “Connect” to reconnect. The module uses a rather short timeout (50 s) for manual control.



## Setting up the device – Frequently Asked Questions

- **Ethernet no longer works but LINK is lit.**  
Perhaps the device was left in the “RS-232 Setup” mode that is activated by setting **DIP1** = ON? In this mode, the Ethernet does not respond. Set **DIP1** = OFF and restart the device by disconnecting the power supply for at least 3 seconds.
- **RS-485 communication does not work.**  
Make sure that **termination resistors** (120 – 470 Ohms) are present at the line or at the connector.
- When using a RS485 converter, configure **&I1** or **&I2** and remember to turn on HALF DUPLEX using **&H1**.
- **I need to supply power to a RS-232 application.**  
If you don't need to control data flow (HW handshake) but need to power a device connected to the serial port (max. 5 – 10 mA), power your application from the RTS output (pin 7 at the RS-232 connector). Use the **&R0** parameter (&R:RS485/RS422 control) in the setup mode to connect +8 V to +12 V to this pin. I cannot control digital inputs and outputs.
- **Digital inputs and outputs cannot be controlled. RS-232 data work fine.**  
Most likely, “NVT” is disabled. Check this at the “UDP Setup” tab in the Hercules utility, in the TCP Setup, or in the RS-232 Setup.
- **5-second timeout is too short to work with, is it possible to do something about it?**  
Yes, enable “Keep connection” in the Setup. IP Relay then sends a command every 6 seconds that appears as “**NVT: NOP**” in the receiving pane. The connection does not timeout.

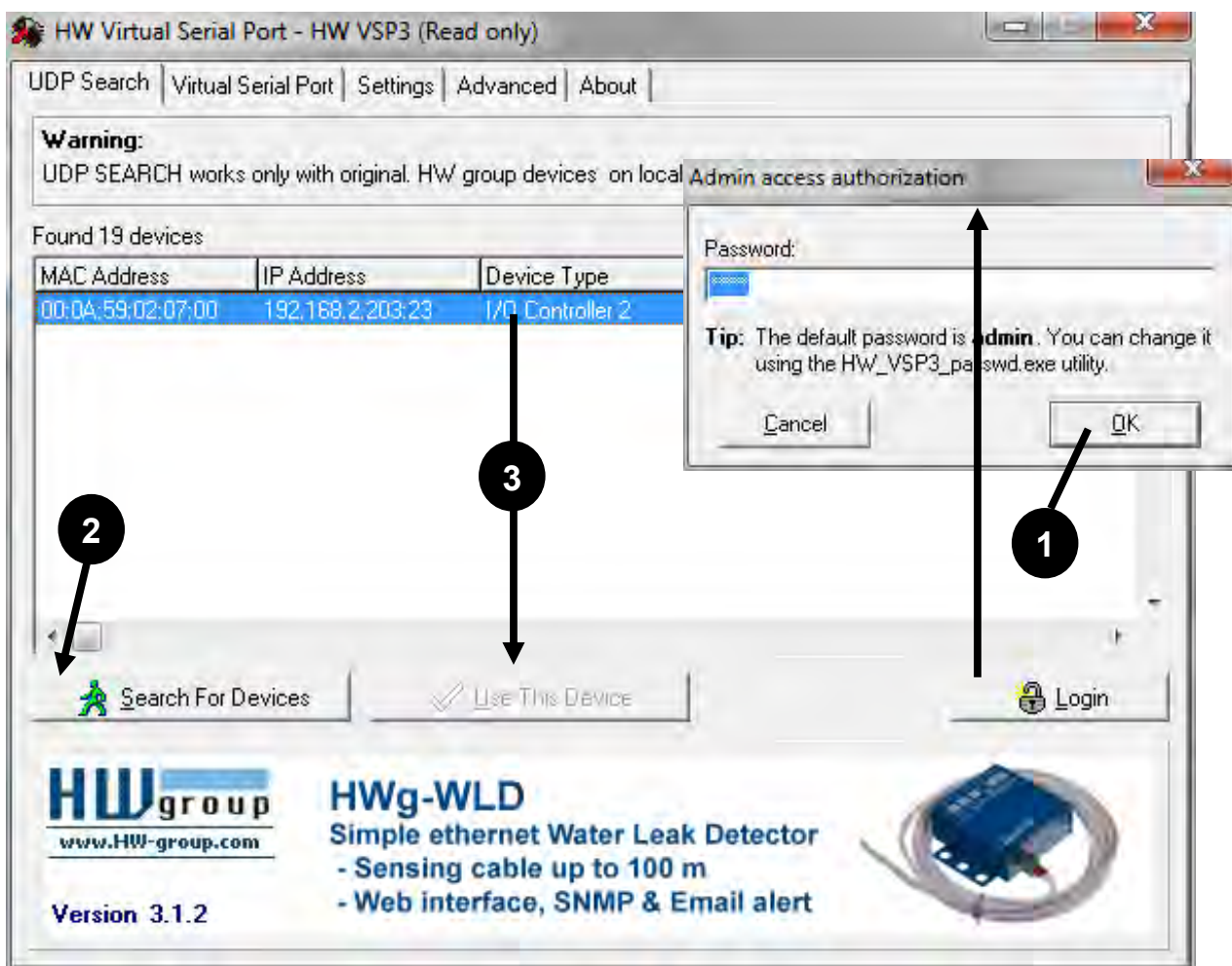
## HW VSP – Virtual Serial Port

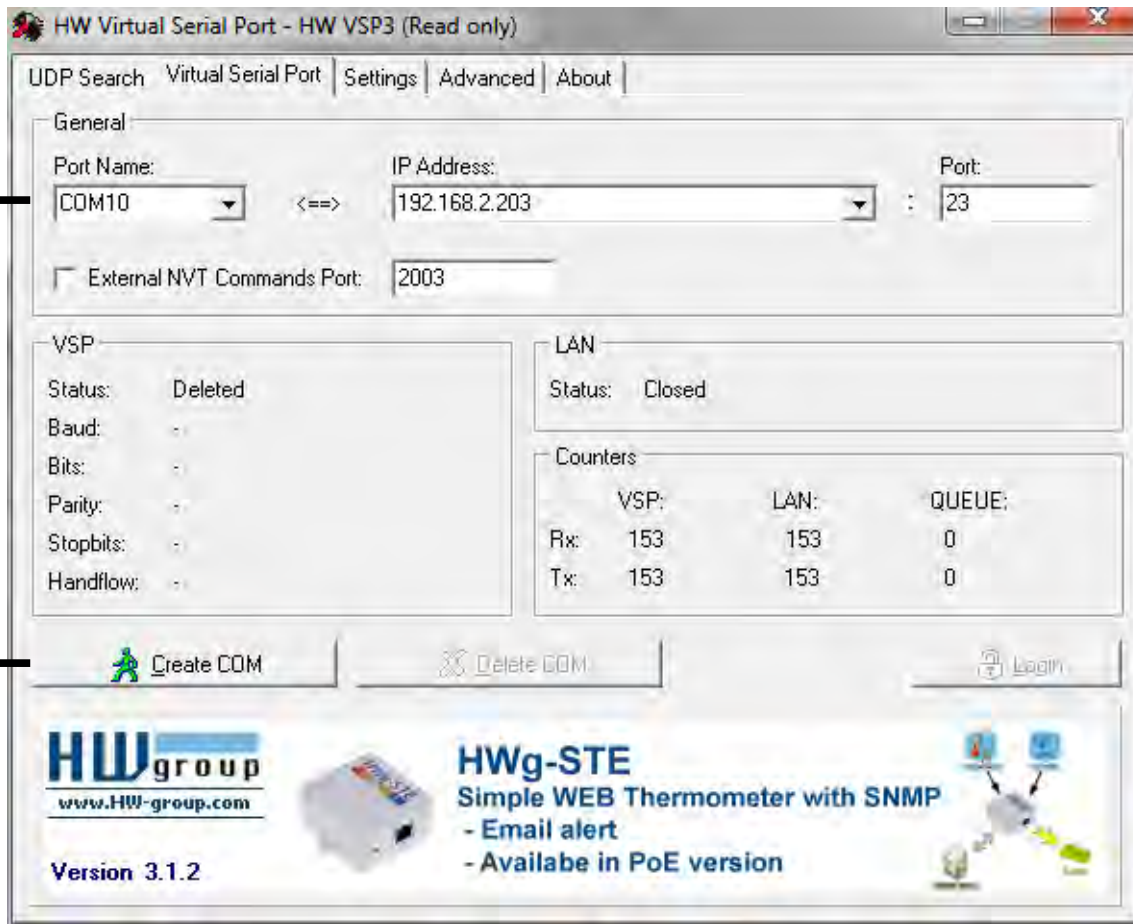
*Virtual Serial Port driver is a software tool that adds a virtual serial port (e.g. COM5) to the operating system and redirects data from this port via the Ethernet network to another hardware interface.*

- **HW VSP Singleport** – Virtual Serial Port (Windows) for one device only
- **HW VSP Multiport** – Virtual Serial Port (Windows) for up to 100 remote serial ports
- The driver works in Windows XP, Vista, Windows 7, Windows 2003 Server, Windows 2008 Server, including 64-bit versions.
- If the device supports RFC 2217 (NVT), you can set the remote serial port parameters (speed, parity, stop bits).
- The communication can be recorded to a LOG file for easier debugging.

### Getting started

- Set up the IP address of the remote serial port device using “UDP setup” or “Hercules”.
- Install HW VSP (from the DVD or from our website) and run it.





## Detailed VSP Settings

- **Log enabled**  
The VSP driver logs the virtual serial port activity to "C:\serialport.log".
- **Create port on start VSP**  
Automatically creates virtual ports when the driver is started. To create virtual ports at Windows startup, the "Start VSP on boot" box must be checked, too.
- **TCP Server Mode**  
Activates VSP as a TCP/IP server. The driver then behaves as a TCP Client/Server device – the first side to receive any data switches to Client mode and establishes the connection.

The incoming TCP server port is configured at the main "**Virtual SP**" tab. We recommend using port numbers higher than 1025.

- **Hide on startup**  
Hides VSP to the system tray. The VSP icon is accessible near the clock.
- **Don't create port if ping fail**  
Before creating the virtual serial port, tests if the device IP address responds.
- **Connect to Device even if VirtualCOM is closed**  
If the virtual serial port is not in use by any application, checking this box can lead to losing some received data from the remote device.
- **Strict Baudrate Emulation**  
Limits the application-to-VSP and TCP-to-VSP transfer rate according to the baudrate of the open serial port.

# Configuring the IP Relay – Command description

The description of the settings combines WWW-based configuration and TCP/Serial setup. Some functions are only available in Serial/TCP setup (TEA security settings, QUIT mode).

## Status

Overview of the basic device parameters.

**IP Relay2 Status**

**System**

Firmware Version	v3.1.x
------------------	--------

**Running TCP/IP Configuration**

IP Address	192.168.1.143
Subnet Mask	255.255.255.0
Default Gateway	192.168.1.254
Send to IP (0.0.0.0 - PASSIVE mode)	0.0.0.0
MAC Address	00:0A:59:02:06:EF

[Save setup & restart I/O Controller](#)

IP Relay2 - For more information try [www.hw-group.com](http://www.hw-group.com)

## Network parameters

Firefox

IP Relay2 Network Setup

192.168.1.143/network.shtml?Q=ASChAXa

Google

### IP Relay2 Network Setup

**HWgroup**

[Status](#)

[Network](#)

[TCP mode setup](#)

[\(UDP mode setup\)](#)

[Serial](#)

[GPIO setup](#)

[System](#)

[IO Control](#)

[Save setup & restart I/O Controller](#)

#### IP Setup

I: Address

J: Port

M: Mask

G: Gateway

#### In IP access filter

W: Address

N: Mask

X: Port

Y: Port Mask

#### Active (Client/Server) mode

S: Send to IP (0.0.0.0 - PASSIVE mode)

U: Port

#### UDP/TCP mode

T: IP mode

IP Relay2 - For more information try [www.hw-group.com](http://www.hw-group.com)



**MAC Address            00:0A:59:00:95:6C**

MAC address is a unique device address in the Ethernet network and it is always factory-preset. You can find it on the label inside the device. Using this address, individual devices can be distinguished for example in the UDP section of the setup program.

The address respects restoring of the default configuration with the “D0” command.

**I: Address            192.168.6.15**

Configures the IP address of the converter.

**J: Port                23**

Configures the converter's communication port – from 1 to 65535.

Port 99 is reserved for TCP Setup, if supported by the particular model and enabled in the setup.

**M: Mask              255.255.255.0**

IP network mask applicable in the local network. Communication with all devices outside of the local network (as determined by the IP address and network mask) is directed through the gateway.

**G: Gateway          192.168.6.254**

Address of the Gateway that enables access to external networks (outside of the IP range defined by the converter's IP address and the mask).

**===== In IP Setup =====****W: Address          0.0.0.0**

IP address of a network or a computer that is allowed to communicate with the device. This value must be the result of a bitwise AND of the remote IP address and the restriction mask (option N), otherwise the device does not react.

**N: Mask              0.0.0.0**

This mask restricts addresses that are allowed to communicate with the device. Security can be enhanced by setting a fixed address or a suitable restrictive mask that disallows communication with unauthorized parties.

**X: Port               0.0****Y: Port Mask        0.0**

Restricts the range of TCP ports that can communicate with the device.

**MASK, IP address and Gateway functions:**

An Ethernet device communicates:

**Within a local Ethernet network:** No Gateway is used or needed. However, IP addresses of both sides must be within the configured mask. For example, a mask of 255.255.255.0 means that the IP addresses can only differ in the last byte.

**Outside of the local network – through the Gateway,** which itself must have an IP address within the range delimited by the mask.

In addition to this basic configuration, it is possible to restrict the range of IP addresses allowed to communicate with the converter in “**In IP Setup**”. For debugging, we therefore recommend to keep this parameter set to 0.0.0.0.

$$(\text{IP attempting to access AND } N) = W$$

Access is granted if the above condition is true. AND represents bitwise multiplication.



## ===== Out IP Setup =====

**S: Send to IP** 192.168.0.252  
**U: Port** 23

If the address in the S option is different from 0.0.0.0, the IP Relay works as a Client/Server in the TCP mode. This means that if **no TCP/IP connection is established** and the device either receives serial port data (even a single byte) or the state of its digital inputs changes (in the range allowed by the #T option), the device periodically tries to initiate a TCP connection as a **TCP Client**. The device remains in the **TCP Server** mode in between attempts.

In the UDP mode, the IP Relay sends data from the serial port to this address/port. Control of digital inputs and outputs in the UDP mode is governed by #E and subsequent parameters.

Set **S = 0.0.0.0** to switch the converter to the **TCP Server** mode.

## Port configuration in UDP mode

When UDP communication is used, the remote address must be specified here.

Otherwise, the communication will be unidirectional. Data from the IP Relay are sent only to the specified address.

**T: IP mode** **TCP**

Switches between TCP and UDP protocols. UDP communication is faster but prone to lost packets or out-of-order delivery; therefore, it is only suitable for communication on a local network segment in a request-response mode, usually to convert RS485 communication.

The “**broadcast Rcv**” parameter allows receiving broadcasts.

**0: TCP**

### 1: UDP with broadcast Rcv Off

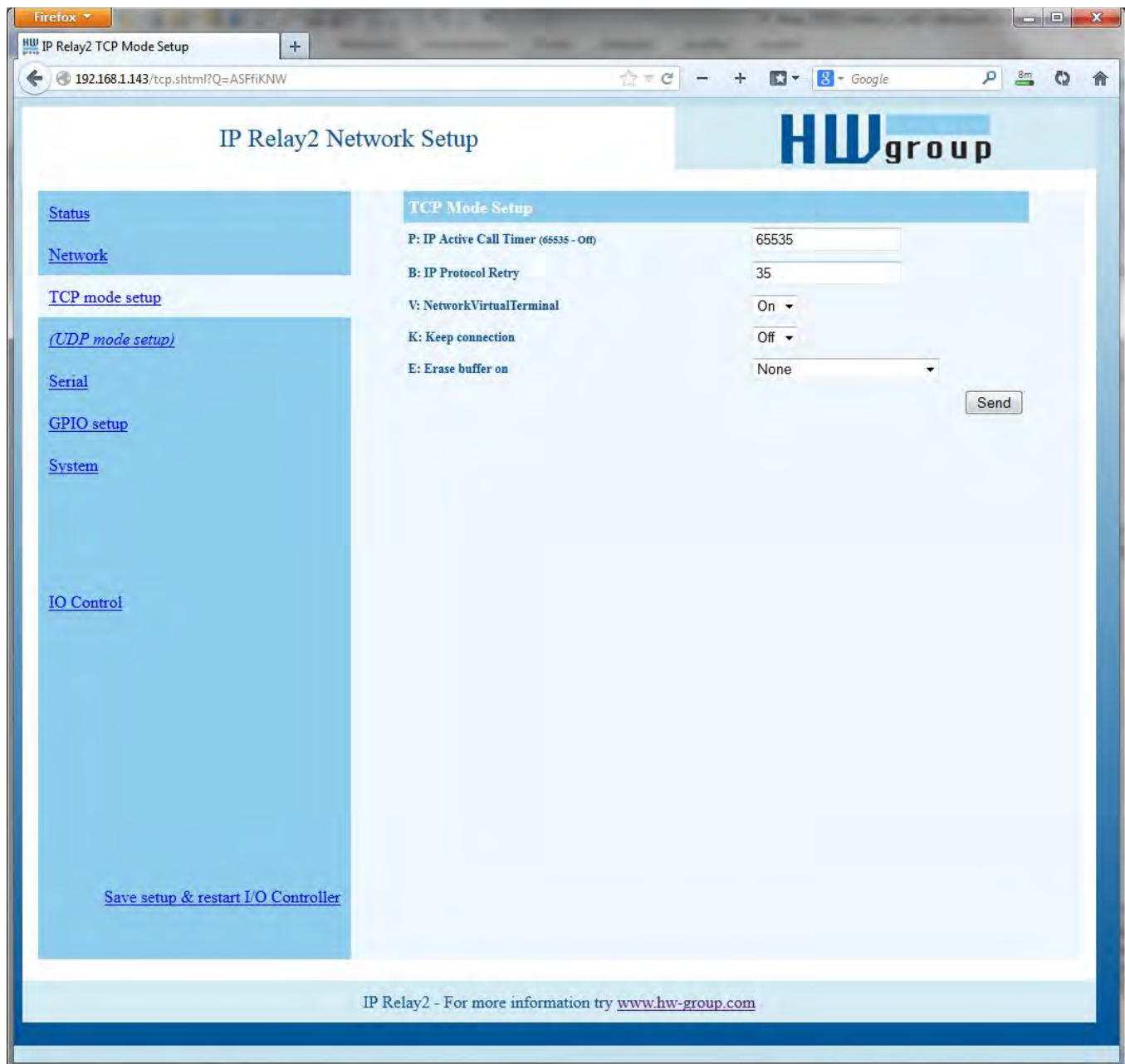
## 2: UDP with broadcast Rcv On

The **UDP/IP** mode does **NOT** support:

- **NVT** commands
- **TEA** authentication

Challenge-response protocols  
respond up to 40% faster.

## Serial port parameters



### P: IP Active Call Timer (65535 – Off)

Periodically establishes connection with the remote side at the specified time, even if there are no new data to send.

Predefined values are available through the asterisk notation:

The timeout can be fine-tuned by specifying a value from 20 to 255: 20..65534 [\* 13.2 sec]

**Usage example** 273 =>  $273 \cdot 13.2 = 3603 \text{ s} = 1 \text{ h}$

#### Predefined values:

*0:	Off	
*1:	10 m	(n = 46)
*2:	15 m	(n = 68)
*3:	30 m	(n = 136)
*4:	1 h	(n = 273)
*5:	2 h	(n = 536)
*6:	3 h	(n = 819)
*7:	6 h	(n = 1637)
*8:	12 h	(n = 3274)
*9:	24 h	(n = 6549)

## B: IP Protocol Retry 124

Determines the TCP connection timeout if no communication takes place. Predefined values are available through the asterisk notation:

```
*1: 30 s (n = 35)
*2: 1 min (n = 45)
*3: 2 min (n = 63)
*4: 5 min (n = 77)
*5: 10 min (n = 101)
*6: 15 min (n = 124)
*7: 30 min (n = 144)
*8: 1 h (n = 179)
*9: 2 h (n = 249)
```

The timeout can be fine-tuned by specifying a value from 10 to 255 according to this formula:

```
n < 16 .. n * 0.2 s
n < 32 .. (n- 15)* 0.8 s + 3.0 s
n < 64 .. (n- 31)* 3.2 s + 15.8 s
n < 128 .. (n- 63)*12.8 s + 118.2 s
n < 256 .. (n-127)*51.2 s + 937.4 s
```

**Example:** B42 =>  $(42-31)*3.2+15.8 = 51s$

### Setting the connection length

Keeping a TCP socket open is useful when small volumes of data need to be transferred as soon as they come in but HW VSP is not used or a permanent connection needs to be avoided. For the duration specified by the B option, the connection is kept open and data are sent immediately, without the delay for establishing a TCP connection.

***On the other hand, a long timeout means a long period of inaccessibility through other sockets (disconnected cable, TCP setup).***

## V: NetworkVirtualTerminal Off

Network Virtual Terminal enables interpretation of Telnet sequences, including certain RFC2217 extensions that enable on-the-fly changes of serial port parameters (speed, parity, etc.). NVT description can be found at our website: >> [“Support & download”](#) >> [“NVT \(Network Virtual Terminal\) protocol description”](#).

When communicating with the serial port using telnet, e.g. with the TeraTerm program or the Hercules utility, NVT should be enabled. If you don't want to use this option, set your client to “RAW” communication mode.

- 0: Off** (do not use telnet control codes, pass through to serial port)
- 1: On** (accept telnet control codes)

**Note:** With NVT off, it is not possible to control the digital inputs and outputs, and the SETUP does not even show the corresponding settings.

## K: Keep connection Off

Keeps TCP connections alive to avoid automatic termination upon timeout specified by the **B:** parameter. **IP protocol retry time.** When enabled, the device sends a NVT NOP command approximately every 5 seconds to check the connection state. If port sampling is enabled with the #T: Trigger AND mask parameter, the device sends an I/O keep command (FF FA 2C 37 ..) instead of a NOP (see #K, #L, #M, #N). NVT must be enabled for this parameter to take effect – when NVT is off, IP Relay repeats the last packet sent and the Keep function is dependent on the remote TCP implementation.

- 0: no keep connection** (preferred)
- 1: keep connection**

**E: Erase buffer on****Open connection**

Clears the internal device buffer whenever a connection is established or closed. This option is useful for instance when your remote peripheral periodically sends some kind of "I'm alive" characters, you only need to access it once in a while, and receiving all these characters from the buffer wastes time.

- 0: none
- 1: Close TCP/IP connection
- 2: Open TCP/IP connection
- 3: Open & Close TCP/IP connection

The screenshot shows the 'IP Relay2 Serial Setup' web interface. The browser address bar displays '192.168.1.143/serial.shtml?Q=ASBmLux'. The page has a blue header with the 'HWgroup' logo. On the left, there is a sidebar with links: Status, Network, TCP mode setup, (UDP mode setup), Serial, GPIO setup, System, and IO Control. The main content area is titled 'Serial Setup' and contains the following configuration options:

- &B: Speed: 9600
- &D: Data bits: 8
- &P: Parity: None
- &V: Variable Parity: Off
- |- &M: Multidrop mask (0 = DISABLE Multidrop): 0
- |- &N: Multidrop network address: 0
- &S: Stop bits: 1
- &C: Flow Control: None
- &R: RTS Output: unasserted [~-8V]
- &A: DTR Output: unasserted [~-8V]
- &T: Serial Line Timeout (0 - off): 0
- &G: Char. Transmit Delay (0 - off): 0
- &H: Tx Control: Tx FULL duplex
- &I: RS232/RS485/RS422 mode: RS232
- %S: TCP/IP setup: On

A 'Send' button is located at the bottom right of the configuration area. At the bottom of the page, there is a footer that reads 'IP Relay2 - For more information try [www.hw-group.com](http://www.hw-group.com)'.

**===== Serial Setup =====****&B: Speed 9600**

Sets the communication speed for the serial line. Any speed from 50 to 115,200 Bd can be set. To set 9600 Bd, enter at the command prompt: "&B9600". The resolution depends on the speed. Up to about 1000 Bd, 1 Bd steps can be used. At 10 kBd, only 100 Bd steps are still usable.

**&D: Data bits 8**

Number of data bits in the serial communication.

**7: 7 bits / 8: 8 bits** – for instance, to set 8 data bits, enter "&D8".

**&P: Parity NONE**

Parity of the serial asynchronous communication.

For example, to configure communication without parity, enter "&PN".

**N: none / O: odd / E: even / M: mark / S: space**

**&V: Variable Parity Parity Off**

Supplemental function for 9-bit protocols. Only the difference from the pre-set parity bit value is transmitted. For correct operation, a parity needs to be set (usually Mark/Space). A double 0xFE character followed by "P" is used to transfer the difference from the pre-set parity. In this mode, the 0xFE character is a prefix; if it appears within the data stream, it needs to be doubled. This option is recommended for the box-2-box mode (two converters connected back-to-back) and 9-bit protocols.

**Off:** incorrect parity bit ignored

**On:** incorrect parity bit transmitted to the other device

For instance, to set up 9-bit communication with a majority of data having "space parity", enter: "&PS;&V1" (space parity + variable parity on).

**&M: Multidrop mask (0 = DISABLE Multidrop)****&N: Multidrop network address**

Evaluates parity errors at the serial line. &V1 Variable parity must be enabled and a parity must be set. Usual settings are 8 bits, SPACE, 1 stop bit.

At the beginning, nothing is transferred from the serial line and the device waits for a byte with "incorrect" parity; usually 8 MARK 1.

When such a byte is received, it is masked with &M and compared with &N (<received byte> AND &M == &N). If true, data transfer is enabled. 0xFE 'P' + received byte is inserted before the data. Then, data is transferred in a transparent manner, until another byte with "incorrect" parity is received.

When such a byte arrives, it is tested. If the test fails, the COM->TCP transfer is stopped and the device waits for another byte with "incorrect" parity. This invalid "address" byte nor other marks are sent over TCP.

In other cases, the configured address is sent to the TCP channel as defined by the variable parity protocol – <0xFE><P><address>. The transfer is stopped as soon as another byte with the 9th bit set (modified parity) is received.

**&S: Stop bits** **2**

Number of stop bits for the serial communication. As a rule, there should be at least 9 bits and at most 10 bits in total, excluding the start bit. For instance, the 7N1 setting (7+0+1 bits) is corrected by the setup to 7N2. Similarly, 8E2 (8+1+2 bits) is corrected to 8E1.

**&C: Flow Control** **NONE**

Serial data flow control. If you use data flow control and the input buffer is full, handshake will signal over the serial port that the IP Relay can no longer accept data.

- 1: none** – No control, see &R for RTS
- 2: RTS/CTS** – RTS/CTS control signals
- 3: Xon/Xoff** – Software flow control

- 4: Xon/Xoff HeartBeat** – SW flow control with periodic Xon transmission (heartbeat)

**&R: RTS Output Continuously asserted [~ +8V]**

Defines the idle level of the RTS output pin. Important for devices powered from the RTS pin.

- 0: RTS = continuously asserted [~ +8V]**
- 1: RTS = unasserted [~ -8V]**
- 2: RTS = asserted while connected**

**&A: DTR Output**

Defines the idle level of the DTR output pin. Important for devices powered from the DTR pin.

- 0: DTR = continuously asserted [~ +8V]**
- 1: DTR = unasserted [~ -8V]**
- 2: DTR = asserted while connected**

**&I: RS485/RS422 control Off**

For RTS or downstream RS485 converters that use RTS to toggle transmission/reception. For the built-in RS-485 driver, the “**HW echo**” option applies, meaning that the device reads back the data sent to the RS-485 line and generates an echo from the actual RS-485 bus.

- 0: Off**
- 2: TxRTS HW echo ON** (recommended for **RS-485 debug only!**)
- 3: TxRTS HW echo OFF** (**RS-485**)

**Note:** For most RS-485 applications, set &R3TxRTS HW echo OFF.

**&T: Serial Line Timeout** **0 – Off**

If no data are received from the serial line for the specified time, characters received so far are packed into an Ethernet packet and sent off.

The timeout is specified as the **number of characters**, and displayed as the number of chars as well as the time based on the current serial communication speed. If the speed changes, the time is recalculated but the number of characters defining the timeout stays the same (10 characters means about 11 ms at 9600 Bd, or 5.7 ms at 19200 Bd).



**&G: Char. Transmit Delay                      0 – Off**

When controlling devices with a small RS232 buffer, it is sometimes advantageous to keep a relatively high baud rate but insert delays between individual characters. The delay is specified in **milliseconds** and defines the time between the starts of individual characters. Therefore, delays under 2 ms have no effect at 2400 Bd because individual characters are 2.4 ms apart.

**&H: Tx Control                                      Tx FULL duplex**

When HALF duplex is activated, the converter expects unidirectional communication over the serial line (RS485) and never starts to transmit data while receiving.

<b>0: FULL duplex (RS-232)</b> <b>1: HALF duplex (RS-485)</b>
--

**&Q: EOT Trigger character                      26**

Packet termination character. The default is 26 (ctrl-Z in ASCII). In common operating systems, ctrlZ is the EOF (End Of File) character. When this character is received from the serial line, the device does not wait for the "&T" timeout and immediately sends everything as a packet over LAN.

**%S: TCP/IP setup                                      On**

Enables or disables remote configuration through the TCP Setup at port 99. This command only works in the **RS-232 Setup** mode.

- 0: TCP Setup disabled**
- 1: TCP Setup enabled** (TCP server at port 99)

## Configuring the control of inputs and outputs

*TCP/IP mode needs to be configured and NVT (Network virtual terminal) enabled in order to work with the digital inputs and outputs. Otherwise, the commands are not even displayed.*

Digital inputs and outputs are controlled over the **Network Virtual Terminal**. This means that the I/O control commands are embedded into the TCP/IP data stream, along the serial port data. However, these commands only appear in the Ethernet communication, they never pass through to the serial port.

In recent firmware versions, digital inputs and outputs can be controlled from the serial port or over an UDP connection (see #E and #V).

NVT commands are always prefixed with a control character and have a predefined binary format. A detailed description of the NVT protocol with examples of commands is available at our website (in the “Support & Download” section).

**Note:** Remember to set the desired initial value of outputs after reset.

The screenshot shows the 'IP Relay2 GPIO Setup' web interface. The left sidebar contains navigation links: Status, Network, TCP mode setup, (UDP mode setup), Serial, GPIO setup, System, and IO Control. The main content area is titled 'IP Relay2 GPIO Setup' and features the HWgroup logo. It contains several configuration sections with input fields and dropdown menus:

- I/O Power Up**: #A: Power Up INIT (0), #B: Power Up AND mask (255), #C: Power Up OR mask (0), #D: Power Up XOR mask (0).
- GPIO Mask**: #X/#K: KEEP mask (0), #Y/#L: AND mask (255), #Z/#M: OR mask (0), #W/#N: XOR mask (0).
- I/O TCP/IP Connection**: #H: I/O HeartBeat (Off), #I/#4: AND mask (255), #2/#5: OR mask (0), #3/#6: XOR mask (0).
- I/O trigger and edge mask**: #T: Trigger AND mask (0), #R/#F: Rise/Fall edge mask (0).
- UDP I/O control**: #E: GPIO control from UDP (Off), #J: Port (24).
- Active UDP mode**: #S: Send to IP (0.0.0.0), #U: Port (24).
- COM I/O control**: #V: GPIO control from COM (Off).

A 'Send' button is located at the bottom right of the configuration area. At the bottom of the page, there is a link: 'Save setup & restart I/O Controller'.

## ===== I/O Control Setup =====

### #T: Trigger AND mask 240

Defines inputs whose **changes are automatically transmitted to the remote side** (IP address S=X.X.X.X and port U) and synchronized with its outputs, if they change.

Only those inputs for which the corresponding bits of #T are set to 1 are transmitted.

#### Examples:

- **#T = 0 (0x00)** – IP Relay does not react to any changes at digital inputs I0 through I8.
- **#T = 240 (0xF0)** – IP Relay only reacts to changes at digital inputs I7, I6, I5, I4. Changes at inputs I3 to I0 are ignored. However, their changed values are always transmitted together with any reaction to a change at I7..I4.
- **#T = 255 (0xFF)** – IP Relay reacts to any change at any digital input I0 through I8.

Transmission means that whenever the IP Relay is in the Client/Server mode ("Active mode"), it reacts to changes at its inputs just as it reacts to incoming data from the serial port = if the connection is closed, the IP Relay establishes a connection with the specified remote side and sends the appropriate NVT command to set the remote outputs.

If the TCP Server mode ("Passive mode") is used and the connection is closed, nothing happens. If the connection is open, data are sent through the open connection.

**Note:** *Even if the input states are not transmitted to the remote side, they can still be read using the standard NVT commands for reading inputs.*

IP Relay distinguishes three **types of synchronization** when using two devices back-to-back. Their parameters are usually configured in a similar way:

- **Power Up init – (#B, #C, #D parameters)** – After reset, IP Relay sets the output to the value specified in #A and attempts to contact the remote IP defined with S=x.x.x.x. When successful, it requests the state of the remote device's inputs and sets its own inputs according to the following formula.

$$\text{OUTPUT} = ((\text{retrieved remote data AND \#B}) \text{OR } \#C) \text{XOR } \#D$$

- **Data change – (#X, #Y, #Z, #W parameters)** – Upon every change at the inputs (as restricted with #T), IP Relay informs the remote device. The remote device receives the value and uses the second formula to set its outputs.
- **Keep I/O – (#K, #L, #M, #N parameters)** – IP Relay periodically sends the state of its inputs to the remote side.

#### Data synchronization after RESET (Power Up init):

The device keeps trying to establish the connection for approximately 120 seconds after powering up. If this does not succeed, the **#A: Power Up INIT** value remains at the outputs.

**#A: Power Up INIT                      0**

A decimal value from **0 to 255** that is written to the **output register** after **device reset** (or power failure), before the IP Relay attempts to establish a connection with the remote unit and synchronize their digital inputs and outputs.

**#B: Power Up AND mask                255**

Binary mask of values from the remote side that affect the outputs when initializing for the first time (after reset or power up).

**#C: Power Up OR mask                 0**

Range of digital output values that can be influenced by the states of inputs retrieved from the remote side after RESET.

**Bitwise OR**

0 (0x00)	OR 0 (0x00)	= 0 (0x00)
255 (0xFF)	OR 0 (0x00)	= 255 (0xFF)
255 (0x0F)	OR 3 (0x03)	= 255 (0xFF)
240 (0xF0)	OR 8 (0x08)	= 248 (0xF8)

**#D: Power Up XOR mask               0**

The binary XOR function is suitable inverting individual output bits. This is used, for example, when a button closes against GND but the corresponding relay needs to switch against +PWR (logic LOW at the input needs to be inverted to logic HIGH at the output).

**Examples:**

- **#D = 0 (0x00)** – Values are transferred without changes. Logic HIGH at an input results in a logic HIGH at the respective output.
- **#D = 1 (0x01)** – The D0 output bit is inverted with respect to the D0 input bit. Digital values at D1 to D7 are transferred without changes (logic HIGH at input = logic HIGH at output).
- **#D = 255 (0xFF)** – All output bits are inverted with respect to the respective input bits.

**Data synchronization in normal operation:**

$$\text{OUTPUT} = ((\text{PrevOut} \text{ AND } \#X) \text{ OR } (\text{RxData} \text{ AND } \#Y) \text{ OR } \#Z) \text{ XOR } \#W$$

- **PrevOut** = previous output state
- **RxData** = received remote data

**Note:** The previous formula is only used when two IP Relays are connected to each other back-to-back. Inputs at one device are automatically mirrored at the outputs of the other device. This is called the **Box-2-Box** mode.

Standard NVT commands can be used to access all the 8 output bits either directly, without respect to the #B to #Z parameters, or using the defined masks.

However, in the common Box-2-Box mode, it is not possible to access the output states from a PC because the IP Relay supports only one TCP connection at a time. As long as a TCP connection to the other IP Relay is established, the IP Relay cannot be accessed from a PC.

**#X: KEEP mask 0**

Defines the outputs that will keep their previous values (using bitwise AND).

**#Y: AND mask 255**

Using bitwise AND, defines the inputs whose states are transmitted from the remote side to the outputs of the IP Relay.

**Bitwise AND**

0 (0x00)	AND 0 (0x00)	= 0 (0x00)
255 (0xFF)	AND 0 (0x00)	= 0 (0x00)
255 (0x0F)	AND 3 (0x03)	= 3 (0x03)
240 (0xF0)	AND 16 (0x0F)	= 0 (0x00)

**Note:** Be careful when controlling the outputs over NVT and using the Box-2-Box mode at the same time. #X and #Y can define which bits can be set only over NVT and which bits will be synchronized with the remote inputs.

*Example: #Y= 0x00, #X=0xFF – The output is not affected by the remote side at all, all output bits can be controlled over NVT.*

**#Z: OR mask 0**

Defines the output bits that will be affected by automatic synchronization – see #C.

**#W: XOR mask 0**

The bitwise XOR function is useful for inverting individual output bits with respect to the respective remote inputs – see #D.

**Bitwise XOR**

0 (0x00)	XOR 0 (0x00)	= 0 (0x00)
255 (0xFF)	XOR 0 (0x00)	= 255 (0xFF)
255 (0x0F)	XOR 3 (0x03)	= 252 (0xFC)
0 (0x00)	XOR 255 (0xFF)	= 255 (0xFF)

**Periodic data synchronization in normal operation:**

$$\text{Output} = ((\text{PrevOut} \text{ AND } \#K) \text{ OR } (\text{RxData} \text{ AND } \#L) \text{ OR } \#M) \text{ XOR } \#N$$

- **PrevOut** = previous output state
- **RxData** = received remote data

**Keep I/O – (#K, #L, #M, #N) – IP Relay periodically sends the state of its inputs to the remote side.**

#K: KEEP mask 255  
 #L: AND mask 0  
 #M: OR mask 0  
 #N: XOR mask 0

#K#L#M#N commands correspond to #X#Y#Z#W and define the behavior when two IP Relays are connected back-to-back and configured to periodically transmit port states. #X#Y#Z#W commands define the behavior of the two IP Relays whenever the port states change (more precisely, whenever the bits determined by #T change).

**#H: I/O HeartBeat Off**

Configures IP Relay behavior when periodically transmitting input states. It is tied with the **K: Keep connection** command.

- **K0** disables periodic state transmission, #K#L#M#N#H have no effect and are inaccessible in the menu.
- **K1;H0** enables periodic state transmission only when nothing else is sent (e.g. data from the serial port).
- **K1;H1** enables periodic state transmission approximately every 5 seconds regardless of the serial port communication.

**Setup examples**

Using the described parameters, each bit can be independently configured either to keep a fixed value, to synchronize itself with changes at an input, or to invert its state with respect to the state of an input. Study the following examples:

**Output = (X.n =0 Y.n =0 Z.n =0)** – output at logic LOW, configurable only with NVT commands

**Output = (Z.n =1)** – output at logic HIGH, configurable only with NVT commands

**Output = (X.n=0 Y.n=1 Z.n=0 W.n=0)** – output is an exact copy of the remote input

**Output = (X.n=0 Y.n=1 Z.n=0 W.n=1)** – output is an inverted copy of the remote input

**Note:** *In normal operation, the inputs are sampled every millisecond and a change is registered whenever the input value differs from the last transmitted value in two consecutive samples (= the change lasts at least 1.2 to 2.0 ms).*



## Transmitting input states upon an edge

IP Relays can transmit current input states to the corresponding outputs at a specified remote device. The inputs to mirror are defined by the edge mask.

### ----- I/O edge mask -----

#### #R: Rise edge mask      255

Defines the inputs that will be monitored for **rising edges**. A rising edge triggers the transmission of information about an input state change ("closed" state is transferred):

255=all inputs, 00=no inputs.

#### #F: Fall edge mask      255

Defines the inputs that will be monitored for **falling edges**. A falling edge triggers the transmission of information about an input state change ("open" state is transferred):

255=all inputs, 00=no inputs.

## Configuration of I/O control

I/O lines can be controlled over:

- **WEB**
- **TCP Telnet** (NVT commands)
- **RS-232** (NVT commands sent to the serial port – #V)
- **UDP** (NVT commands via UDP to the specified port – #U)
- **Modbus/TCP**

### ----- I/O control -----

#### #E: GPIO control from UDP      Off

#### #J: Port      24

When enabled, commands to change output states can be sent to the IP Relay over UDP as well as over TCP.

#J defines the UDP port where the I/O commands need to be sent.

#### #S: Send to IP      192.168.0.252

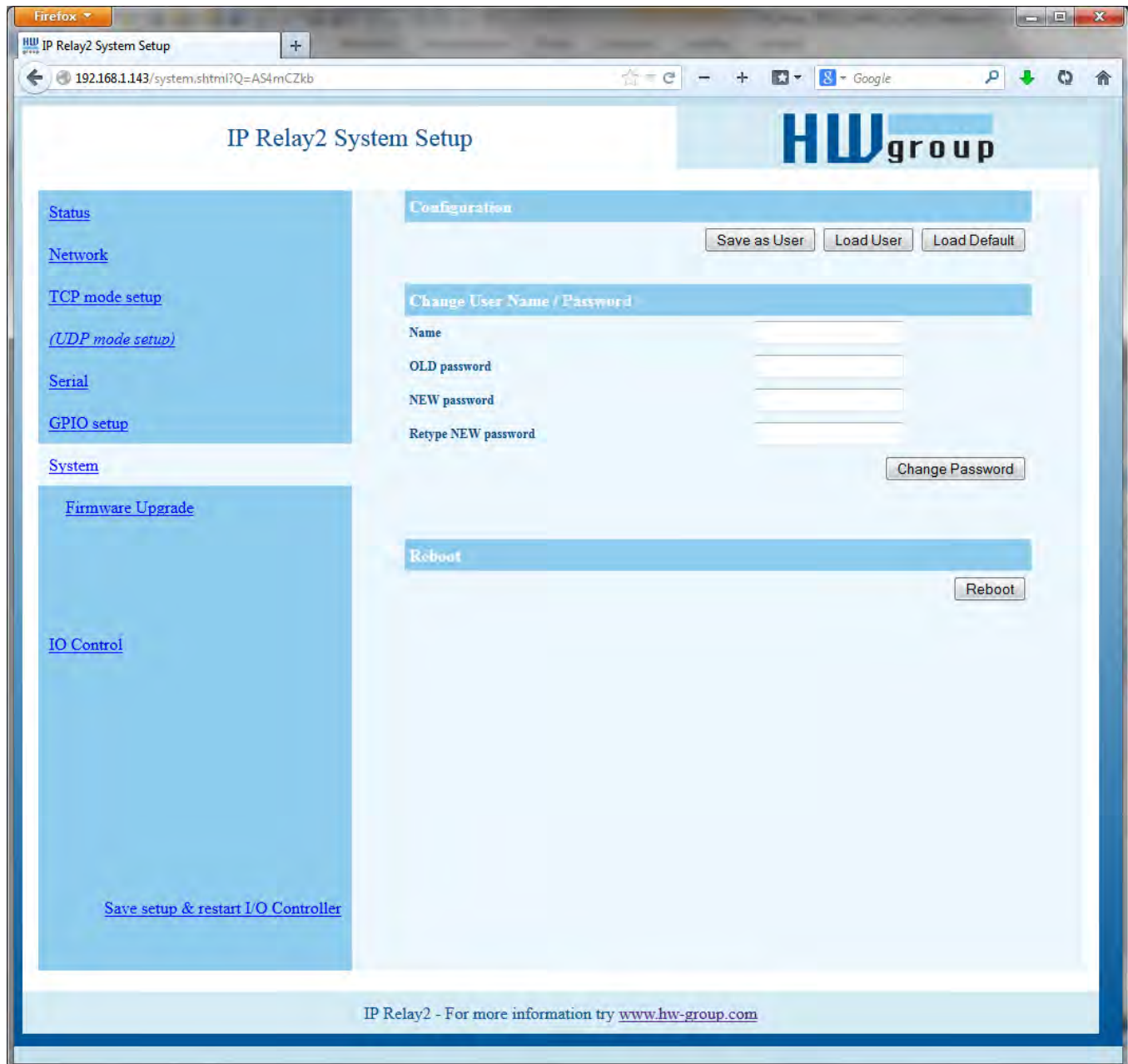
#### #U: Port      4024

When the UDP GPIO Control mode is enabled, IP Relay sends the state changes to the specified address and port.

#### #V: GPIO control from COM      Off

When enabled, IP Relay inputs and outputs can be controlled over the serial line. Commands are similar to NVT commands – 0xFE prefix followed by a NVT command. For example, 0xFE 0x33 xx sets the output to xx. The prefix is the same as for the &V command.

## System



### D: Load/Save Settings from/to Flash

Saves the current settings to memory, or restores saved settings.

- 0: Restores settings from **slot 1**
- 1: Restores settings from **slot 2**
- 2: Stores current settings to **slot 1**
- 3: Stores current settings to **slot 2**

### R: Reboot

Software restart. Necessary e.g. when the IP address is changed. Recommended after changing parameters in TCP Setup.

### Change Password

Applies only to WWW access. The default credentials are **admin** and **1234**.

## UDP/IP mode settings

If you select “**T: IP mode UDP**”, the device will communicate with the remote side using UDP packets. Also, the following menu appears in the Setup.

### ==== Triggerring Setup =====

#### \*L: Trigger Length 1

Number of bytes for the packet start and packet end triggering condition. Allowed values are 0 to 4. If the lengths of your start and end conditions differ, use the trigger condition mask and remember to include the masked characters in the length of the triggering condition, even though they contain actual frame data.

**\*P: Post Trigger Length                      0**

In some protocols, the packet end condition is followed e.g. by a checksum. This condition defines the number of characters that make up a packet after the packet termination condition. If the start and stop trigger conditions are equal, this value specifies the packet length without the leading 0 to 4 bytes of the start trigger.

**\*S: Start Trigger Pattern                      58.0.0.0**

(4B input data AND \*M) = \*S

Start trigger condition for packet transmission. Four bytes are set; however, only the number of bytes specified in "L: Trigger Length" is considered.

**\*M: Start Trigger Mask                      255.0.0.0**

Mask of the start trigger condition. Masking works similarly to the Ethernet netmasks using a bitwise AND. Value of 255 means that the tested character must be equal to the character specified in "V: Start Trigger Pattern". For example, to start the transfer with any ASCII control character (0..31d), use 0.0.0.0 for the trigger pattern, 224.0.0.0 for the mask and 1 for the length. If you set the character to 0 and the mask to 0, the trigger matches any character.

**\*X: Stop Trigger Pattern                      10.0.0.0**

(4B input data AND \*Y)= \*X

Sets the stop trigger condition for sending data to the Ethernet.

**\*Y: Stop Trigger Mask                      255.0.0.0**

Mask of the stop packet trigger condition for serial line data. For example, the settings displayed here are intended for transferring data in the IntelHEX format over RS485. The start trigger is a colon and the transfer is terminated after receiving the <LF> (0Ah = 10d) control character.

**Configuration example**

*To send all data from the serial line to the remote device, use this configuration:*

*L: Trigger Length	0
*P: Post Trigger Length	0
*S: Start Trigger Pattern	0.0.0.0
*M: Start Trigger Mask	0.0.0.0
*X: Stop Trigger Pattern	255.0.0.0
*Y: Stop Trigger Mask	255.0.0.0
*E: Max. Start-Stop Length	200

**\*E: Max. Start-Stop Length                      999**

Maximum number of characters that the device sends after the START trigger, unless the STOP trigger is encountered sooner. After transmission, another START trigger is expected. Essentially, this is a "timeout" specified as the number of characters.

**\*A: Max. Packet Length                      250**

Only active in the UDP mode.

Maximum data size in an UDP packet; if more data received, sends the data immediately.

## Parameters unavailable in the WWW interface

### Security

#### ===== Security Setup =====

#### %A: TCP autorisation Off

Activates TEA authentication (one-time exchange and password verification to allow the TCP connection to be established) – requested from the remote side after the connection is established.

<b>0: TEA authentication Off</b> <b>1: TEA authentication On</b>
---

#### %K: TEA key 0:01:02:03:04 1:05:06:07:08 2:09:0A:0B:0C 3:0D:0E:0F:10

Use “%K” to set the TEA key. Set 16 bytes as four quadruples of colon-separated hex values. The first character identifies the quadruple (0th to 3rd). So, to set the last 4 bytes to the displayed value, use “%K3:0D:0E:0F:10”. The key is used to verify one-time passwords (OTP) when authenticating the remote side.

### Other device parameters

#### Q: Quiet (Batch) mode

Quiet mode is useful when the configuration parameters need to be handled automatically, with a script. Quiet mode is enabled with “Q1”. After pressing Enter, the device responds with a list of parameters in the following format:

```
WEB51=2.L=00:0A:59:00:A6:08;I192.168.1.24;J23;M255.255.255.0;G192.168.1.1;W0.0.0.0;N0.0.0.0;X0.0.0.0;Y0.0.0.0;S192.168.6.51;U4023;T0;V0;K1;A250;B9600;D8;P1;S1;C1;R0;T0;G0;H0;A0;%K01:02:03:04:05:06:07:08:09:0A:0B:0C:0D:0E:0F:10;%S1;#T3;#A3;#B192;#C3;#D0;#X0;#Y0;#Z3;#W0;*L1;*P0;*S58.0.0.0;*M255.0.0.0;*X10.0.0.0;*Y255.0.0.0;*E999
```



## Default settings

```

*** HW-group.com HW 6.0 SW 3.1.x #01 ***
***          PortBox I/O          ***

MAC Address          00:0A:59:02:06:EF
===== IP Setup =====
I: Address           192.168.1.143
J: Port              23
M: Mask              255.255.255.0
G: Gateway           192.168.1.254
===== In IP access filter =====
W: Address           0.0.0.0
N: Mask              0.0.0.0
X: Port              0.0
Y: Port Mask         0.0
== Active (Client/Server) mode ==
S: Send to IP        PASSIVE mode
U: Port              4023
B: IP Protocol Retry 35
T: IP mode            TCP
V: NetworkVirtualTerminal On
K: Keep connection   Off
E: Erase buffer on    None
===== Serial Setup =====
&B: Speed            9600
&D: Data bits         8
&P: Parity            None
&V: Variable Parity   Off
&S: Stop bits         1
&C: Flow Control      None
&R: RTS Output        unasserted [~ -8V]
&A: DTR Output        unasserted [~ -8V]
&T: Serial Line Timeout 0 - Off
&G: Char. Transmit Delay 0 - Off
&H: Tx Control        Tx FULL duplex
&I: RS485/RS422 control Off
===== Security Setup =====
%A: TCP autorisation  Off
%K: TEA key 0:01:02:03:04 1:05:06:07:08 2:09:0A:0B:0C 3:0D:0E:0F:10
%S: TCP/IP setup      On
===== I/O Control Setup =====
#T: Trigger AND mask  0
#A: Power Up INIT     0
#B: Power Up AND mask 255
#C: Power Up OR mask  0
#D: Power Up XOR mask 0
#X: KEEP mask         0
#Y: AND mask          255
#Z: OR mask           0
#W: XOR mask          0
--- I/O tcp/ip OPEN connection ---
#1: AND mask          255
#2: OR mask           0
#3: XOR mask          0
-- I/O tcp/ip CLOSE connection --
#4: AND mask          255
#5: OR mask           0
#6: XOR mask          0
----- I/O edge mask -----
#R: Rise edge mask    0
#F: Fall edge mask    0
----- I/O control -----
#E: GPIO control from UDP Off
----- Active COM mode -----
#V: GPIO control from COM Off
===== Other =====
D: Load/Save Settings from/to Flash
R: Reboot

```

## Controlling inputs and outputs via NVT

The following is a short overview of controlling the IP Relay via the M2M NVT protocol. It is an extract from the detailed description of NVT that is available at our website:

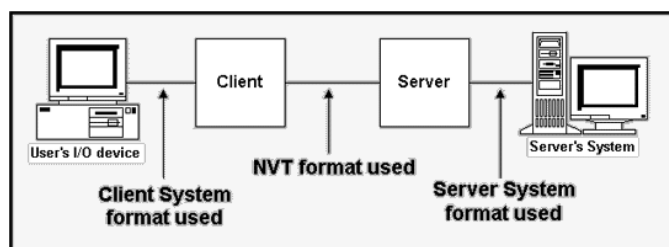
[http://www.hw-group.com/support/nvt/index\\_en.html](http://www.hw-group.com/support/nvt/index_en.html)

### What is NVT and RFC2217

NVT is a method of embedding control commands into a binary data stream. NVT (Network Virtual Terminal) is also used by the Telnet protocol to transfer commands. In this way, Telnet transfers control sequences such as CTRL-Pause, cursor positioning on the screen, changing terminal type, and so on.

For remote control of serial ports, Cisco has defined RFC2217 that defines commands for changing the serial port speed, querying the states of digital I/O signals, etc. Our devices implement most of these commands. For a list, see the online description of NVT at the above-mentioned website.

We extended the standard RFC2217 commands to include several GPIO (General Purpose Input Output) functions listed below. These functions can control the digital input and output pins of the IP Relay. Our extension is not standardized; however, when we implemented this extension in 2001, no such standard was known to us.



### How does it work

The commands are control sequences in the TCP/IP data stream; the "FF" character starts the control sequence with a defined format. If the "FF" (255 decimal) character appears in the normal data, it must be doubled at the transmitting side. The receiving side automatically converts a double "FF" to a single "FF". When both sides support NVT, the "FF" character always prefixes a control sequence.

Supported NVT commands COM-PORT-OPTION - 44 (2C)		
Dec	HEX	Function
0	00	CAS_SIGNATURE
1	01	CAS_SET_BAUDRATE
2	02	CAS_SET_DATASIZE
3	03	CAS_SET_PARITY
4	04	CAS_SET_STOPSIZE
5	05	CAS_SET_CONTROL
6	06	CAS_NOTIFY_LINSTATE
7	07	CAS_NOTIFY_MODEMSTATE
8	08	CAS_FLOWCONTROL_SUSPEND
9	09	CAS_FLOWCONTROL_RESUME
10	0A	CAS_SET_LINSTATE_MASK
11	0B	CAS_SET_MODEMSTATE_MASK
12	0C	CAS_PURGE_DATA
50	32	CAS_OPT_GPIO
51	33	CAS_SET_GPIO
52	34	CAS_SET_GPIOM
:	:	
+100	+64	ASC_
150	96	ASC_OPT_GPIO
151	97	ASC_SET_GPIO
152	98	Not implemented, one way "answer" only

Values up to 100 Dec = **Client >> Server**  
 Values over 100 Dec = **Server >> Client**

**CAS\_** request for the device to perform a command

**ASC\_** device response, command performed + confirmation

Commands start with <IAC><SB> (**FF FA**) and end with <IAC><SE> (**FF F0**).

### Note:

For examples of NVT commands, see the manual to the I/O Controller2 product. The commands are compatible.

## Contact

---

HW group s.r.o

Rumunská 26 / 122  
Praha 2, 120 00, Czech republic

Tel. +420 222 511 918  
Fax. +420 222 513 833



<http://www.HW-group.com>