I/O Controller Manual

-I/O Controller PL

-HWg-ER88a

-HWg-ER84a

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.

The device must not be used 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

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

I/O Controller

Ethernet – RS-232/485 + inputs and outputs

*I/O Controller is an Ethernet-enabled device with 8 digital inputs, 8 (*HWG-ER-84a: 4*) digital outputs and a RS-232/485 serial interface.*

All interfaces are accessible over TCP/IP using a M2M protocol. Two devices can be connected against each other (Box-2-Box mode) in order to transfer digital and RS232 signals over the computer network.

Basic features

- **1x RS-232 or RS-485** serial port accessible over the Ethernet
- The remote port can be controlled with a **virtual driver for Windows** (free driver for Windows XP / Vista / Windows 7 / Server 2003 / Server 2008 / x64) compatible with RFC2217. The virtual port appears, for instance, as **COM5**.
- 10 Mbps Ethernet interface – 10BASE-T, RJ45
- Support for TCP/IP terminal, TELNET - NVT type (*Network Virtual Terminal*)
- **Two devices** can „tunnel“ the serial port, 8 inputs and 8 outputs over the Ethernet
- Three ways to **configure the device**:
    - Locally over RS232 (serial terminal)
    - TCP/IP terminal access at port 99
    - UDP Broadcast using a simple Windows utility
- Wide range of supported **serial interface** parameters:
    - Communication speed configurable from 300...115200 Bd
    - Handshake (CTS/RTS, Xon/Xoff, none)
    - Support for 7th to 9th parity bit (9th parity bit transferred over the Ethernet)
- SDK (Software Development Kit) is available for the device with examples for MS Visual Basic, Delphi, Borland C++, JAVA, PHP and more

Technical specifications

RS-232 serial port		
+ Data bits		7 or 8 or 9
+ Stop bits, Parity		1 or 2, None / Odd / Even / Mark / Space parity
+ Baudrates		50..115.2 kBd – entire range
+ Data flow control		XON/XOFF, CTS/RTS, None
+ Interface		1x DB9M (RxD,TxD,RTS,CTS,GND)
+ Used RS-232 signals		RxD,TxD,RTS,CTS, (DTR output – defined voltage level only)
+ Used RS-232 internal buffer memory		740 B for incoming data, 740 B for outgoing data
RS-485 serial port		
+ Termination		None or internal 120Ω termination (HWg-ER8x only)
+ Isolation		RS-485 line not optocoupled to the device power supply - electrically isolated RS-232/485 to Ethernet (1.000 V)
Digital (binary) inputs & outputs		
+ Input type		Dry contact input
+ Logic LOW voltage		0 .. 3V
+ Logic HIGH voltage treshold / "on" current		from 5V / 5mA to 20V / 25mA
+ Max. input voltage and current		up to 40V / 50mA / 1 second
+ Isolation Voltage		max. 50V to power supply
+ Sampling period		10ms
+ SW control of inputs and outputs		Inputs and Outputs controlled over a M2M protocol based on the NVT (extended) – short 7-byte binary commands over TCP/IP
+ Output type	IO Controller PL, HWg-ER88a	Darlington transistors with common emitter, suppression diodes max. 50V max. 500mA / 1 output and max. 1.500mA / all 8 outputs
	IO Controller HWg-ER84a	Relay output 50V/1A with NC/NO contacts
Ethernet port		
+ Interface		RJ45 (10BASE-T) – 10 Mbps or 10/100 Mbps networks only!
+ Compatibility		Ethernet: Version 2.0/IEEE 802.3
+ Supported protocols		IP: ARP, TCP + NVT (Network Virtual Terminal)
+ TCP connection closing		Data – 50s timeout (with NVT – ACK/NOP support)
Physical parameters & Environment		
+ Temperature & humidity		Operating: 0 – 65°C (32 – 149°F) Storage: -10 to 85°C (14 to 185°F), humidity 5 to 95 %
+ Power supply requirements		8–24V / Max. device current consumption 200mA DC
		- barrel (coaxial) power connector, GND on the shield
+ Dimensions / Mass		I/O Controller PL 28 x 105 x 135 [mm] (H x W x D) / 395 g HWg-ER8x 145 x 90 x 45 [mm] (H x W x D) / 241 g

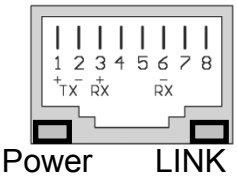
Note: This parameter table is only indicative.
For a valid table, see the specification for the particular device type.

Connectors and LEDs

Indication:

Power – green (RJ-45 connector)..... External power connected
LINK – yellow (RJ-45 connector).....Ethernet interface activity
RxD – red..... Incoming RS-232 data
TxD – green.....….. Outgoing RS-232 data

Connector pinouts

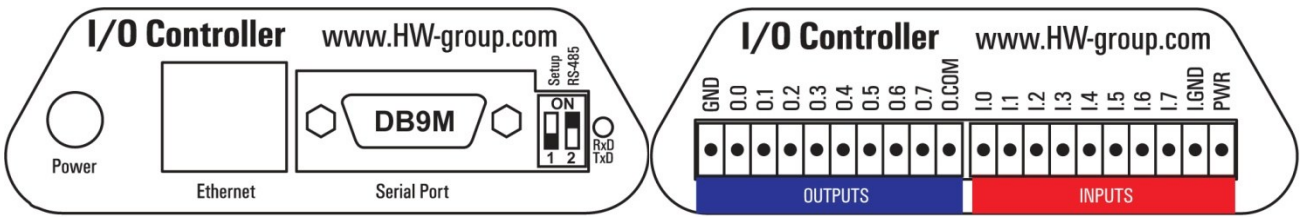


I/O Controller port		
Pin	Signal	
1		
2 <-	IN	RxD
3 ->	OUT	TxD
4		
5 --		GND
6		
7 ->	OUT	RTS
8 <-	IN	CTS
9		
D-sub 9 - Male		

Standard IBM PC RS-232 Port	
Pin	Signal
1 <-	CD
2 <-	RxD
3 ->	TxD
4 ->	DTR
5 --	GND
6 <-	DSR
7 ->	RTS
8 <-	CTS
9 <-	RI
D-sub 9 - Male	

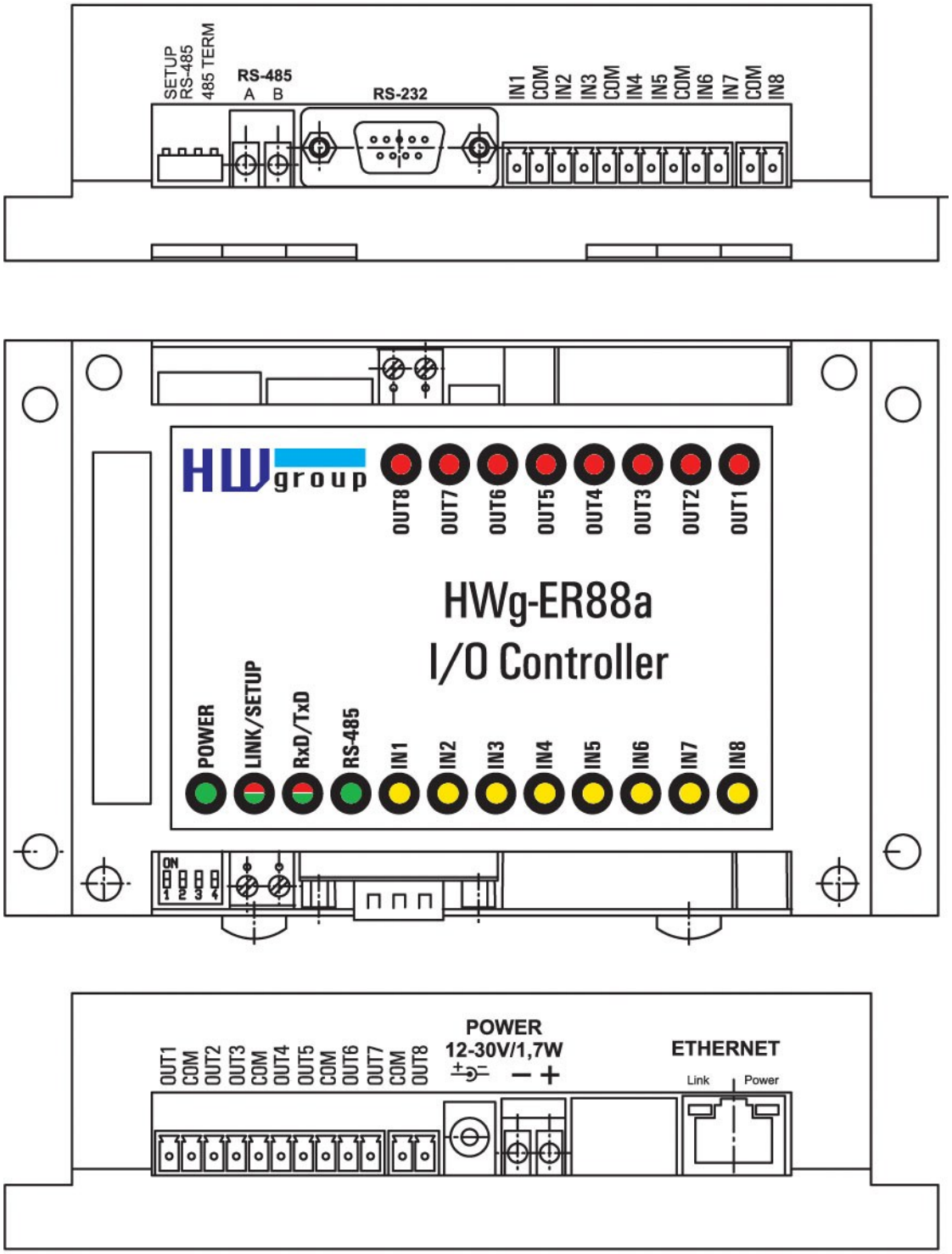
DIP 1	DIP 2	Function
ON	OFF	**RS-232 Setup** mode (9600 8N1) *Ethernet interface is disabled*
OFF	OFF	Serial port in **RS-232** mode
OFF	ON	Serial port in **RS-485** mode *- check the &R and &H parameter settings (we recommend &R3 &H1)*

I/O ControllerPL



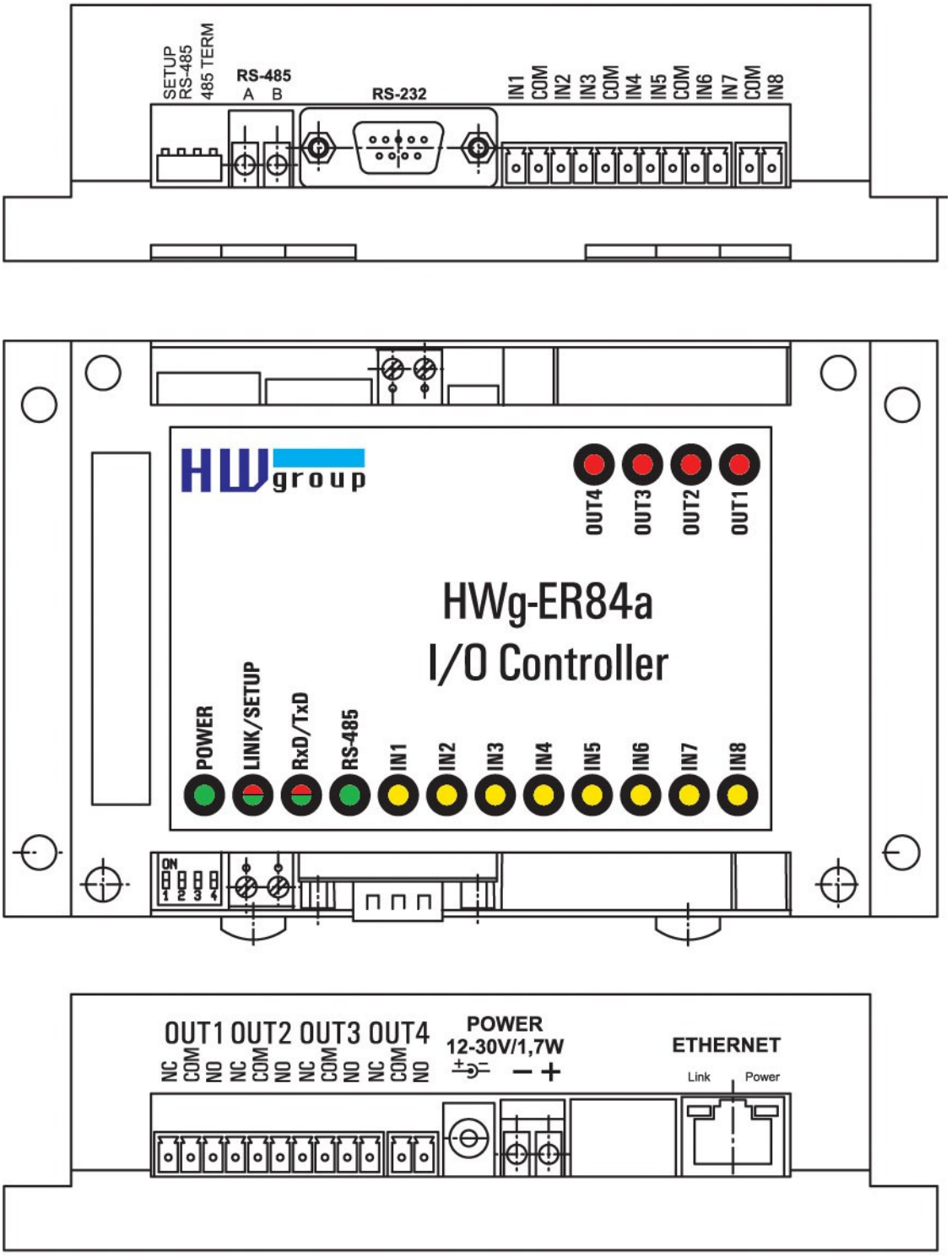
Note: For the input and output wiring diagrams, see the type specification.

I/O Controller HWg-HW88a



Note: For the input and output wiring diagrams, see the type specification.

HWg-HW84a



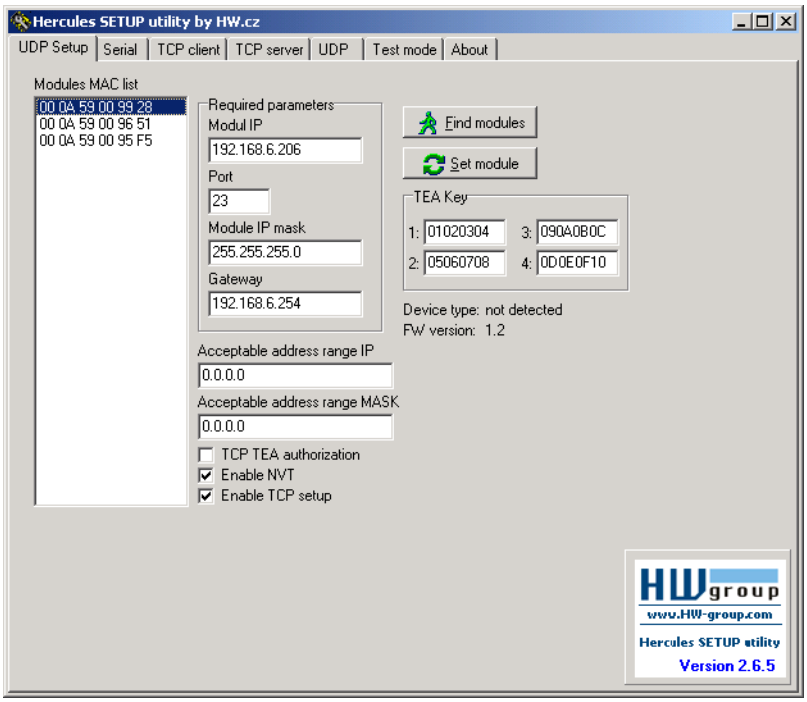
Note: For the input and output wiring diagrams, see the type specification.

Quick SETUP

Set up the device in 5 minutes. Details below.

Connecting the cables

- Connect the supplied power adapter to an electrical outlet.
- Set **DIP1 and DIP2** to **OFF.**
- Connect the device to a **10 Mbps** or 10/100 Mbps Ethernet network.
- Connect the power adapter to the power connector at the device.
- If the power is OK, the green **Power** LED lights up.
- If the Ethernet connection works properly, the **LINK** LED lights up, and then flashes whenever data transfer takes place (activity indication).



Configuring the IP address

- Run the "**HerculesSetup.exe**" utility available on the DVD (included in the "set" package) or for free download at www.HW-group.com.
- Click "**Find modules**" at the "**UDP Setup**" tab. The MAC address of the device displays in the left column. Single-click the MAC address and set the required parameters (at a minimum, the IP address, mask and gateway).
- Make sure that "Enable TCP Setup" is checked and click "Set module" to save the parameters to the device.
- You have now configured the IP address and other networking parameters and you can work with the device.

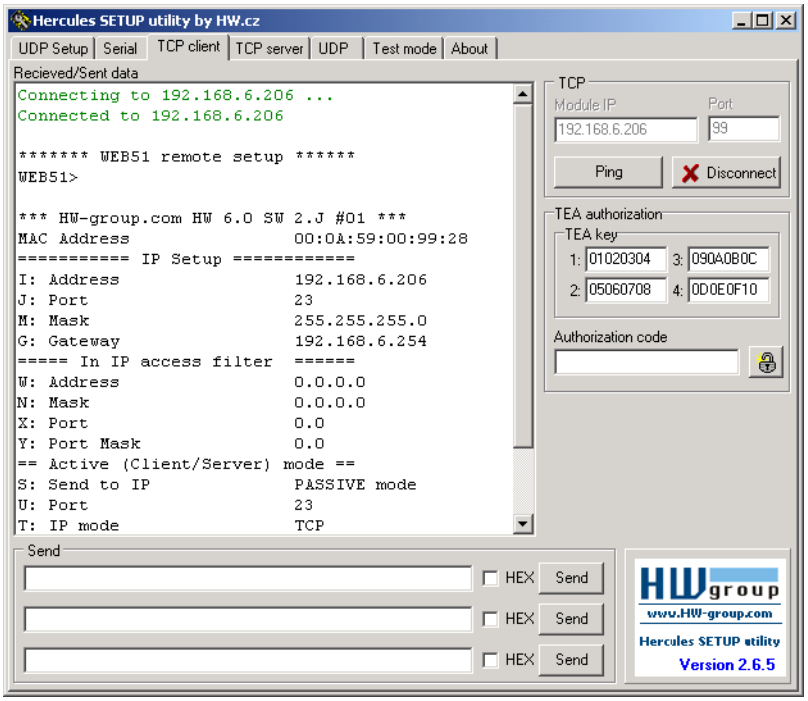
MAC address not visible in the list?

LINK led did not light up, or the device does not respond? Please double check the following.

- Does your Ethernet network support 10 Mbps devices?
- Are you using a correct TP cable (straight-wired TP Patch when connecting to an Ethernet switch, crossed cable when connecting to a PC)?
- Check the DIP switch settings (all should be OFF).
- Check your power adapter and that the Power LED is on.
- Check the firewall settings at your PC.

Setting up the device using TCP Setup

- Switch to the "**TCP Client**" tab and enter the configured IP address. Set **TCP port number** to **99**.
- Click "Connect". The listing on the left shows a "**WEB51>**" prompt. Click this field and press ENTER. The window lists the current parameter settings.
- To set the individual parameters, enter the corresponding letter and the desired value (for example "**I192.168.6.8**" to set the device IP address). To print the help for a command, enter the command, 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 and restart it.
- ***Note:*** *TCP Setup can be disabled using the "Enable TCP setup" option. If the option is unchecked, the device refuses connections to port 99.*

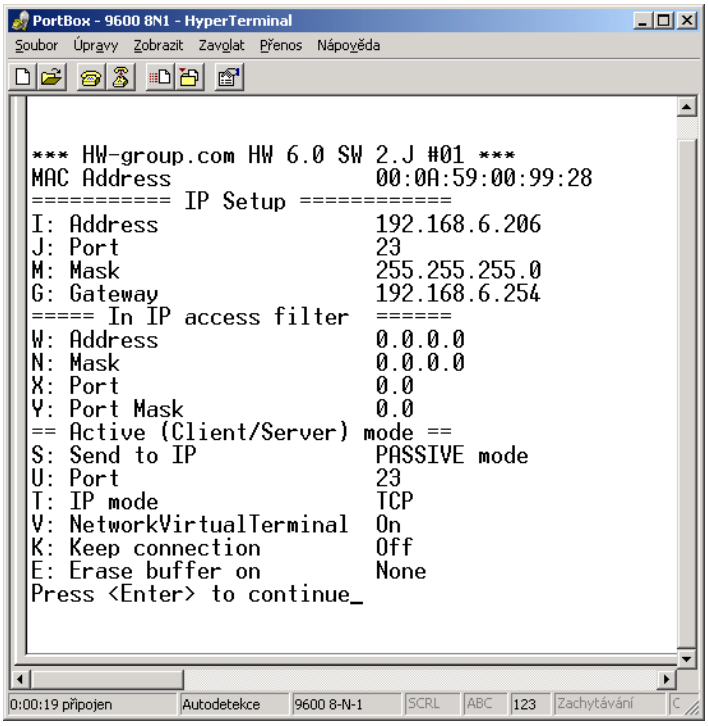


Setting up the device using a RS-232 terminal

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 the 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 connect the other end to the power connector at the device.
- If the power is OK, the green **Power** LED lights up.
- If the serial cable and terminal program are working, the text menu for configuring the device appears.

Follow the steps as in the previous case.

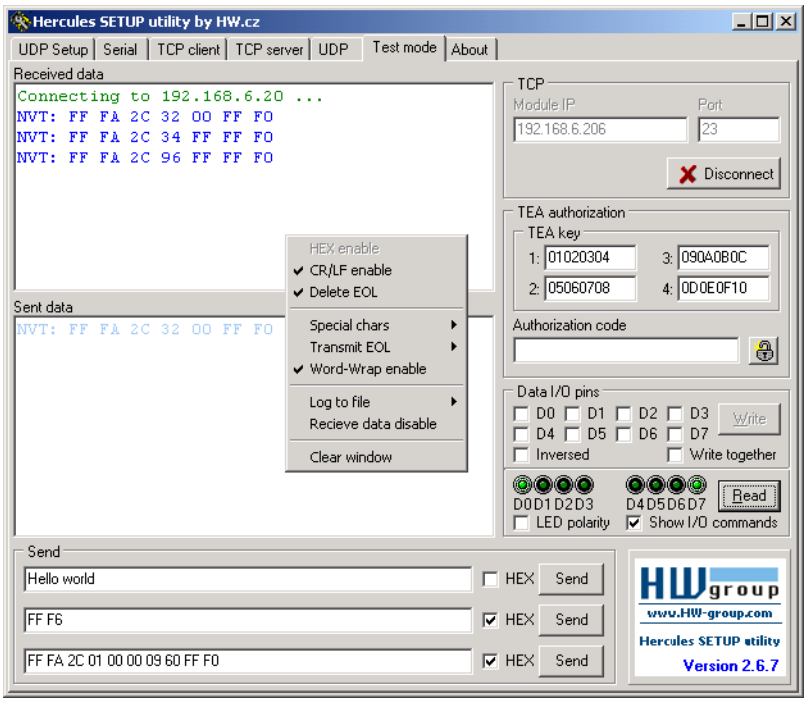


Quick control of inputs and outputs

In this chapter, we will read a digital input for the first time and set a digital output to a desired value. We assume that a test board supplied with device samples is used.
The following assumes that you know how to enter the setup mode (RS-232 Setup or TCP Setup at port 99). The steps were described above.

- Set **DIP1 = OFF**, **DIP2 = OFF,** connect the test board to the connector and turn on the device.
- Change to the "**UDP Setup**" tab, search for the device 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. If they are 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 the port to the Test mode 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 received data. All recognized commands are displayed in blue.

  The image shows the listing and the configuration 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 input states (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.
- Clicking the D0 through D7 boxes sets the outputs as shown on the screen. The first command sets all outputs to defined states, individual bits are controlled afterwards. Output states can again be inverted with the "Inversed" option.
  The "Write together" option avoids sending commands when changing individual outputs. The outputs are set to the defined values by clicking "**Write**".
- When "Show I/O commands" is enabled, the incoming and outgoing message panes display the I/O Controller control sequences. This makes it easy to test the needed commands.
- Enter "FF F6" to the Send lines at the bottom, check HEX and click the corresponding Send 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 I/O Controller replied with its HW and SW version. The SN number consists of the last three bytes of the MAC address.
- If the TCP connection closes while working (red message "`Connection refused by remote host`"), click "Connect" to reconnect. The module uses a rather short timeout for manual control, 50 seconds.

Setting up the device – Frequently Asked Questions

- **Ethernet stopped working but LINK is still on.**
  The device might be in "RS-232 Setup" mode, that is, **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 conversion to RS485, configure **&I1** or **&I2** and make sure to enable HALF DUPLEX with **&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. RS-232 data work fine.**
  Most likely, "NVT" is disabled. Check the setting at the "UDP Setup" tab in the Hercules utility or in the TCP Setup or RS-232 Setup.
- **50 second timeout is too short to work with, what can I do?**
  Enable "Keep connection" in the Setup. I/O Controller then sends a command every 6 seconds that appears as "`NVT: NOP`" in the receiving pane. The connection does not timeout.

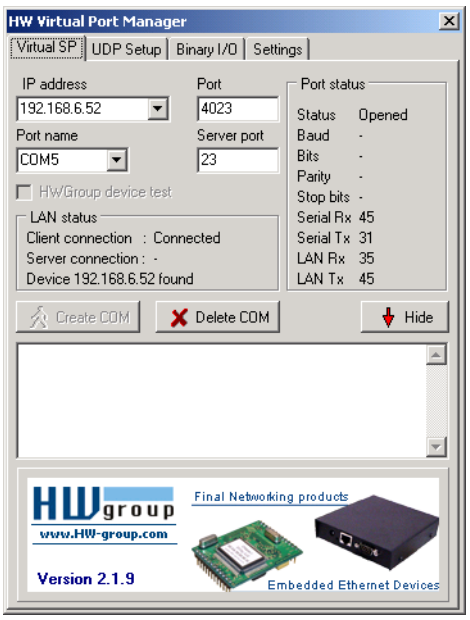
HW VSP – virtual serial port

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.

- The tool 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 parameters of the remote serial port (speed, parity, stop bits).
- The communication can be recorded to a LOG file for easier debugging.
- It is possible to create multiple virtual serial ports on a single computer (COM5, COM6, COM7) by starting VSP.EXE from the command line with appropriate parameters.

Using HW VSP with I/O Controller

- Install HW VSP ("**HW_VirtualSerialPort**" directory on our CD). When installing, make sure to check the option to install the second part. After installation, restart your PC.
- Make sure that NVT is enabled on the **UDP Setup** tab in the **Hercules** utility. If not, enable NVT and save the settings.
- Start HW VSP and search for the device on the "**UDPsetup**" tab. Select the MAC address of the device and click the "Use this IP" button. Change to the "**Virtual SP**" tab. The IP and Port should be already set. This search via UDP Broadcast works only on a local network.
- From the COM1 to COM20 range, select a serial port to create and click "Create COM". The "LAN status" pane shows whether the device has been found. If so, the virtual serial port is created.



- As soon as you start any application that opens the selected virtual serial port (COM5 in this example), the HW VSP driver establishes a connection with the I/O Controller, configures the remote port (speed, parity, number of bits, handshake) according to the virtual serial port being opened, and starts transferring data.

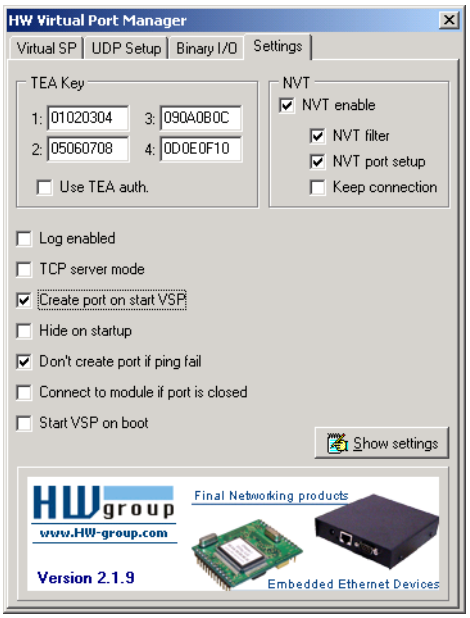
  Some applications have problems with serial port numbers higher than COM4. You can always use the Hyperterminal in Windows, the Serial tab in the Hercules setup utility, or the "Terminal.exe" utility originating from Slovenia and available on the CD in the utils directory.

- Select the I/O Controller sub-tab in the "Binary I/O" tab. Here you can set and read digital inputs and outputs without additional software.

HW VSP settings

TEA Key pane

You can use TEA authentication to secure TCP/IP access. The same TEA key must be set at both sides of the communication.



NVT pane

Enables RFC2217 and detection of our remote ports. Remember to activate NVT support on the corresponding device as well.

- **NVT filter** – Filters NVT control characters from the data flow.
- **NVT port setup** – Sends control commands to the remote port according to the VSP in your PC. For example, if your terminal program (e.g. Hyperterminal) changes the baudrate to 19200 Bd and this box is checked, the VSP driver sends a NVT command (according to the RFC 2217 standard) for the remote TCP/IP serial port to change its baudrate.
- **Keep Connection** – Keeps the TCP/IP connection open even after 50 seconds of inactivity.

Main HW VSP parameters

- **Log enabled**
  The VSP driver logs the virtual serial port activity to "**C:\serialport.log**".
- **TCP server mode**
  Activates VSP as a TCP/IP server. The driver then behaves as a TCP Client/Server device - this means that 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.
- **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.
- **Hide on startup**
  Hides VSP to to the system tray. The VSP icon is accessible next to the clock.
- **Don't create port if ping fail**
  Before creating the virtual serial port, tests if the device IP address responds.
- **Connect to module if port is closed**
  If the virtual serial port is not in use by an application, checking this box can lead to losing received data from the remote device.
- **Start VSP on boot**
  Inserts the VSP path to RUN key (HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\run) in the Windows registry. VSP is then started every time Windows starts.

HW VSP – command line parameters

HW VSP can be also started with specific parameters on the command line. In this way, it is possible to create several virtual serial ports simultaneously on one computer. Detailed description of parameters can be found at our website: www.HWgroup.com.

Example: `CharonVirtualCom.exe -R -i192.168.6.21:23 -c5 -S0 -N1 -Nf -Np -H1`

I/O Controller configuration – Command description

Default settings

```
MAC Address                 00:0A:59:00:AD:8E
=========== IP Setup ============
I: Address                  192.168.1.41
J: Port                     23
M: Mask                     255.255.255.0
G: Gateway                  192.168.1.253
===== 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        124
T: IP mode                  TCP
V: NetworkVirtualTerminal   On
K: Keep connection          Off
E: Erase buffer on          None
Press <Enter> to continue
========= 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: RS485/RS422 control     RTS = On [+8V]
&T: Serial Line Timeout     0 - Off
&G: Char. Transmit Delay    0 - Off
&H: Tx Control              Tx FULL duplex
&Q: EOT Trigger character   122
======== 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        255
#A: Power Up INIT           0
#B: Power Up AND mask       0
#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 edge mask  -------
#R: Rise edge mask          0
#F: Fall edge mask          0
-------     I/O control   -------
#E: GPIO control from UDP   On
#J: Port                    24
-------- Active UDP mode --------
#S: Send to IP              192.168.0.252
#U: Port                    4024
#V: GPIO control from COM   Off
============  Other  ============
D: Load/Save Settings from/to Flash
R: Reboot
```

Network parameters

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, the 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

Configuration of the device's IP address.

J: Port 23

Configuration of the device's communication port – range: 1 .. 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 of the local network. Communication with all devices outside of the local network (delimited by the IP address and network mask) is directed through the gateway.

G: Gateway 192.168.6.254

Address of the gateway that provides access to external networks (IP addresses outside of the range defined by the IP address and network mask).

IP address, Mask and Gateway description:

An Ethernet device communicates:

- **Within the local Ethernet segment**, no Gateway is needed or used; IP addresses of both devices must be in the same range as limited by the Mask. If the Mask is set to 255.255.255.0, the IP addresses may differ only in the last byte.
- **Outside of the local network – using Gateway** (which itself must be within the range of IP addresses delimited by the Mask).

Besides these basic settings, you can also restrict the range of IP addresses that the converter communicates with using the "**In IP Setup**" settings. For debugging, we recommend leaving this parameter set to 0.0.0.0.

====== In IP Setup ======

W: Address 0.0.0.0

IP address of a network or computer that is allowed to communicate with the device. This value must result from 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.

(**IP requesting access** AND **N**) = **W**

Access is granted if this condition is true. AND represents bitwise Boolean multiplication.

TCP and UDP connection parameters

====== 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, I/O Controller 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 is still in the **TCP Server** mode in the meantime between attempts.
In the UDP mode, the I/O controller sends data from the serial port to this address/port. Control of digital inputs and outputs in UDP mode is governed by #E and subsequent settings.

UDP mode and port configuration

If you use the UDP communication, you need to enter the remote address here. Otherwise, communication will be unidirectional only. Data from I/O Controller are sent only to the configured address.

Setting S = 0.0.0.0 switches the device to the **TCP Server** mode.

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 m (n = 45)
*3: 2 m (n = 63)
*4: 5 m (n = 77)
*5: 10 m (n = 101)
*6: 15 m (n = 124)
*7: 30 m (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*

Connection timeout setting

Keeping a TCP socket open is useful when sending small volumes of data that need to be transferred quickly; however, HW VSP is not used, or it is not desirable to keep the connection permanently alive. The connection is kept open for the duration in the B setting and data are sent immediately without delays due to establishing a TCP connection.

A disadvantage of a long timeout is that the device is unavailable through another socket (cable disconnected, TCP Setup entered) before this

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

Request-response protocols can

V: NetworkVirtualTerminal Off

Network Virtual Terminal allows the interpretation of Telnet sequences, including certain RFC2217 extensions that enable on-the-fly changes of serial port parameters (speed, parity, ...). NVT description can be found on 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 (don't 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 show the corresponding settings.*

K: Keep connection Off

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

0: no keep connection (prefered)
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

Serial port parameters

====== Serial Setup ======

&B: Speed 9600

Communication speed for the serial line. Any speed from 50 to 115.200 Bd. To set 9600 Bd, enter: "&B9600". The smallest step 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** – 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 transferred. For correct function, a parity needs to be set (usually Mark/Space). A double character 0xFE 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 mode is recommended for box-2-box mode (two converters connected against each other) and 9-bit protocols.

Off: incorrect parity bit ignored
On: incorrect parity bit transferred to the other device

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

&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, excluding the start bit. If you set, for example, 7N1 (7+0+1 bits), setup corrects this 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 I/O Controller can no longer accept data.

1: none – no control, see &R for the behavior of RTS.
2: RTS/CTS – RTS/CTS control signals
3: Xon/Xoff – software-based flow control

4: Xon/Xoff HeartBeat – SW flow control with periodic transmission of Xon (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

&I: RS485/RS422 control Off

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

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

Note: *For most applications on RS-485, 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 units 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 it is defined as the time between the starts of individual characters. Therefore, a 2 ms delay has 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 received over the Ethernet to the serial line if it is currently receiving data.

0: FULL duplex (RS-232)
1: HALF duplex (RS-485)

&Q: EOT Trigger character 26

A character that triggers an end of a packet. The default of 26 is ctrlZ 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.

Security and remote TCP Setup

===== Security Setup ======

%A: TCP autorisation Off

0: TEA authorisation **Off**
1: TEA authorisation **On**

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.

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

To set the TEA key, use the "**%K**" command. 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.

%S: TCP/IP setup On

Allows or denies the remote configuration using TCP setup on port 99. This command works only in the **RS-232 Setup** mode.

0: TCP Setup disabled
1: TCP Setup enabled (TCP server on port 99)

Input and output control settings

You need to set the TCP/IP mode and enable NVT (Network virtual terminal) 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 command are inserted into the TCP/IP data stream, among the serial port data. However, these command are only present in the Ethernet communication, they never pass through to the serial port.
In new firmware versions, digital inputs and outputs can be controlled from the serial port or over an UDP connection (see #E and #V).

NVT commands have a predefined binary format. They are always prefixed with a control character. 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!*

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

#A: Power Up INIT 0

Value from **0 to 255** (decimal) that is written to the **output register** when the device is **reset** (or power cycled), before the I/O Controller attempts to establish connection and synchronize digital inputs and outputs.

#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.
Transmitted are only those inputs for which the corresponding bits of #T are set to 1.

Examples:

- **#T = 0 (0x00)** - I/O Controller does not react to any changes at digital inputs I0 through I8
- **#T = 240 (0xF0)** - I/O Controller 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)** - I/O Controller reacts to any change at digital inputs I0 through I8

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

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

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

I/O Controller distinguishes three **synchronization types** when using two devices against each other, the parameters are usually set in a similar way:

- **Power Up init - (#B, #C, #D commands)** – After reset, I/O Controller outputs a value specified in #A and attempts to connect to the remote IP defined by S=x.x.x.x. Upon success, it requests the value at the remote input and sets its own outputs to the value at remote inputs according to the formula below.

  OUTPUT = ((**read remote data** AND **#B**)OR **#C**)XOR **#D**

- **Data change - (#X, #Y, #Z, #W commands)** – Upon every change at the inputs that is restricted with #T, I/O Controller informs the remote side, which accepts the value and uses the second formula to set its outputs.
- **Keep I/O - (#K, #L, #M, #N commands)** – I/O Controller periodically sends the states of its inputs to the remote side.

Data synchronization after RESET (Power Up init):
The device attempts to establish the connection approximately for 120 seconds after power up. If this does not succeed, the **#A: Power Up INIT** value remains at the outputs.

#B: Power Up AND mask 255

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

#C: Power Up OR mask 0

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

OR - Bitwise addition

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 is switched against +PWR (logic LOW at the input needs to be inverted to logic HIGH at the output).

Examples:

- **#D = 0 (0x00)** – Binary 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 is inverted with respect to the D0 input. Binary values at D1 through D7 are are transferred without changes. Logic HIGH at an input results in a logic HIGH at the respective output.
- **#D = 255 (0xFF)** – All output values are inverted with respect to the input states.

Common data synchronization:

OUTPUT = ((**PrevOut** AND **#X**) OR (**RxData** AND **#Y**) OR **#Z**) XOR **#W**

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

Note: *The previous formula is only used when two I/O Controllers are connected to each other. Inputs from one side are automatically transmitted to the outputs at 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 I/O Controller supports only one TCP connection at a time. As long as a TCP connection to the other I/O Controller is established, the I/O Controller cannot be accessed from a PC.

#X: KEEP mask 0

Defines the range of outputs that will be kept at the previous value (using bitwise AND).

#Y: AND mask 255

Using bitwise AND, defines the inputs whose states are transferred from the remote side to the outputs of the I/O Controller.

AND – Bitwise Boolean multiplication

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 at the same using the Box-2-Box mode. #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 will not be affected by the remote side, all output bits can be controlled over NVT.

#Z: OR mask 0

Range of output bits which will be affected by automatic synchronization – see the #C parameter description.

#W: XOR mask 0

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

XOR – Bitwise inversion

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

Cyclic synchronization in common operation (since version 3.1.9):

OUTPUT = ((**PrevOut** AND **#K**) OR (**RxData** AND **#L**) OR **#M**) XOR **#N**

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

Keep I/O - (#K, #L, #M, #N commands) – I/O Controller periodically sends the states 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 I/O Controllers are configured to periodically exchange port states. #X#Y#Z#W commands define the behavior of the two I/O Controllers whenever the port states change (more precisely, whenever the bits determined by #T change).

#H: I/O HeartBeat Off
Configures Controller 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 every about 5 seconds regardless of the serial port communication.

Setup examples

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

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 general, the inputs are scanned every millisecond and a change is registered whenever the input value differs from the last transmitted value in two consecutive scans (= the change lasts at least 1.2 to 2.0 ms).*

Transfer of input states upon edge

I/O Controllers can transfer current input states to the respective outputs of the specified 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 transmission of information about input state change (“closed” state transferred):

255=all inputs, 00=no inputs.

#F: Fall edge mask 255

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

255=all inputs, 00=no inputs.

Configuration of I/O control

I/O lines can be controlled with:

- **TCP interface** (NVT commands)
- **RS-232** (NVT commands sent to the serial port – #V)
- **UDP interface** (NVT commands over UDP to the specified port – #U)

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

#E: GPIO control from UDP Off

#J: Port 24

When enabled, commands to change output states can be sent to the I/O Controller over UDP as well as over TCP.
#J defines the port where the I/O commands should be sent.

#S: Send to IP 192.168.0.252

#U: Port 4024

When UDP GPIO Control mode is enabled, I/O Controller sends state changes to the specified address and port.

#V: GPIO control from COM Off

When enabled, I/O Controller 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 same prefix as in &V is used.

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**". When you hit 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;Y0.0;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;#Y
0;#Z3;#W0;*L1;*P0;*S58.0.0.0;*M255.0.0.0;*X10.0.0.0;*Y255.0.0.0;*E999
```

========= Other =========

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 parameters in TCP Setup.

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.*

==== Trigerring Setup =====

*L: Trigger Length 1

Numbers of bytes of the packet start and packet end triggering condition. Allowed values are from 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 lengths – even though they contain actual frame data.

*P: Post Trigger Length 0

In some protocols, the packet end condition is followed by e.g. 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 of data at the input** AND

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 both the character and the mask to 0, the trigger is activated for any character.

*X: Stop Trigger Pattern 10.0.0.0

(**4B of data at the input** AND ***Y**

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.

*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.

Setup example

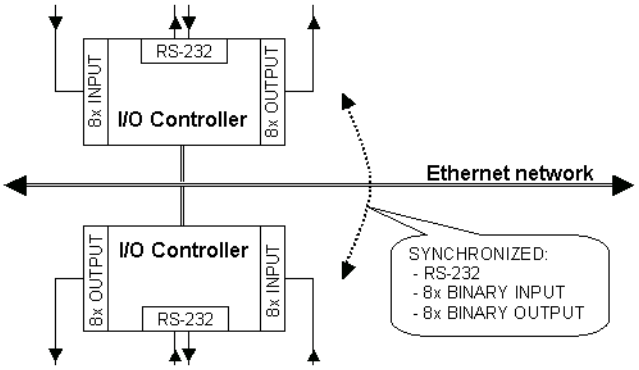
If you need to send all data from the serial line to the remote side, use:

*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

Practical configuration examples

The following examples show the configuration of essential parameters for typical applications.

2x I/O Controller connected to each other (Box-2-Box)



Virtual extension of the serial port + 8 inputs and 8 outputs over TCP/IP.

The serial port is set to 19000 Bd, 8N1, SW handshake, states of digital inputs are transferred to the corresponding outputs.

The TCP connection automatically closes after 50 seconds, and opens again upon a change at an input or upon receiving any data from the serial port.

I/O Controller 1

```
*** HW-group.com HW 4.7 SW 2.L #01 ***
***        I/O Controller          ***

MAC Address                 00:0A:59:00:00:00
=========== IP Setup ============
I: Address                  192.168.1.1
J: Port                     4023
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               192.168.1.2
U: Port                     4023
T: IP mode                  TCP
V: NetworkVirtualTerminal   On
K: Keep connection          Off
E: Erase buffer on          None
========= Serial Setup ==========
&B: Speed                   19200
&D: Data bits               8
&P: Parity                  NONE
&V: Variable Parity         Off
&S: Stop bits               1
&C: Flow Control            Xon/Xoff
&R: RS485/RS422 control     RTS = On [+8V]
&T: Serial Line Timeout     0 - Off
&G: Char. Transmit Delay    0 - Off
&H: Tx Control              Tx FULL duplex
======== 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        255
#A: Power Up INIT           189
#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
============  Other  ============
D: Load/Save Settings from/to Flash
R: Reboot

WEB51>
```

I/O Controller 2

```
*** HW-group.com HW 4.7 SW 2.L #01 ***
***        I/O Controller          ***

MAC Address                 00:0A:59:00:00:00
=========== IP Setup ============
I: Address                  192.168.1.2
J: Port                     4023
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               192.168.1.1
U: Port                     4023
T: IP mode                  TCP
V: NetworkVirtualTerminal   On
K: Keep connection          Off
E: Erase buffer on          None
========= Serial Setup ==========
&B: Speed                   19200
&D: Data bits               8
&P: Parity                  NONE
&V: Variable Parity         Off
&S: Stop bits               1
&C: Flow Control            Xon/Xoff
&R: RS485/RS422 control     RTS = On [+8V]
&T: Serial Line Timeout     0 - Off
&G: Char. Transmit Delay    0 - Off
&H: Tx Control              Tx FULL duplex
======== 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        255
#A: Power Up INIT           189
#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
============  Other  ============
D: Load/Save Settings from/to Flash
R: Reboot

WEB51>
```

- If you want to permit communication with only one address, set N:255.255.255.255 a W:IP address of the remote side.
- If you want 9-bit communication, enable &V1 = &V: Variable Parity On at both devices.
- To enhance security, disable TCP Setup on port 99 %S0 = %S: TCP/IP setup Off at both devices.

If security is an essential requirement and I/O Controller 1 should only communicate within the local network, use the following settings. In this case, only communication within a segment is allowed (0 to 255 at the end of the IP address).

```
I: Address                  192.168.1.1
M: Mask                     255.255.255.0
===== In IP access filter   ======
W: Address                  192.168.1.0
N: Mask                     255.255.255.0
S: Send to IP               192.168.1.2
```

```
I: Address                  192.168.1.2
M: Mask                     255.255.255.0
===== In IP access filter   ======
W: Address                  192.168.1.0
N: Mask                     255.255.255.0
S: Send to IP               192.168.1.1
```

Controlling inputs and outputs using NVT

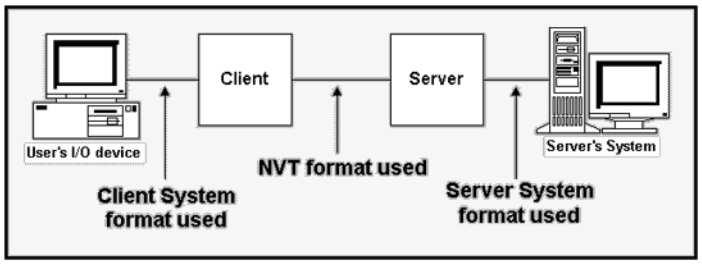
The following is a short overview of controlling the I/O Controller using the M2M NVT protocol. It is an extract from the detailed description of NVT that is available at our website:

http://www.hw-group.com/support/nvt/index_en.html

What is NVT and RFC2217

NVT is a method of adding control commands into a binary data stream. NVT (Network Virtual Terminal) also uses 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, etc.
For remote control of serial ports, Cisco has defined RFC2217 that defines commands for changing the serial port speed, querying the states of binary signals, etc. Our devices implement most of these commands. For a list, see the online description of NVT at the address given in the introduction.

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 I/O Controller. 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 data contains the "FF" (255 decimal) character, it must be doubled by the transmitting side. The receiving side automatically converts double "FF" to single "FF". If there is NVT support on both sides, 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_LINESTATE
7	07	CAS_NOTIFY_MODEMSTATE
8	08	CAS_FLOWCONTROL_SUSPEND
9	09	CAS_FLOWCONTROL_RESUME
10	0A	CAS_SET_LINESTATE_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 under 100 Dec = **Client >> Server**
Values over 100 Dec = **Server >> Client**

CAS_ request for the device to perform a command
ASC_ response from the device, command performed + values acknowledged

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

Description of commands for GPIO control - 50, 51, 52 (32 .. 34 hex)

Control of I/O pins uses the double byte GPIO-50 or 51 command (stands after the COM-PORT-OPTION 44 command) followed by the following sub option sequence.

Sub option 50 (32 hex)

Reading inputs, setting outputs bit by bit, reading the output register.

- **0 (00 hex)** – Request for input states, response includes input port state.
- **16 .. 23 (10 .. 17 hex)** – Set output bit 0..7 to logic HIGH.
- **32 .. 39 (20 .. 27 hex)** – Set output bit 0..7 to logic LOW.
- **48 (30 hex)** – Request for output states, response includes output port state.

Sub option 51 (33 hex)

Sets a value to the output port. In the response, returns the same value (read from the internal register).

Sub option 52 (34 hex)

This command is used by the I/O Controller for automatic transmission of **input states, if there is a state change** or after after device power-up. This command does not expect a response, therefore there is no value for 152 (98 HEX) in the table. The notification itself is actually an unsolicited response without a matching request. (Theoretically, this command can be used in NVT communication, too. It differs from sub option 51 in that the value sent to the port is modified with #X#Y#Z#W.) I/O Controller responds to this command with the same reply as for sub option 51, that is, the response number includes sub option 151.

The 52 (34 hex) command is often preceded by the "**FF FA 2C 32 00 FF F0**" sequence which represents a request for a value for I/O Controller outputs. When a response to this request with I/O Controller's output states is received (for example "**FF FA 2C 33 FF FF F0**"), the "**FF FA 2C 34 ZZ FF F0**" sequence is sent, where ZZ is the new value for the remote outputs.

Note: *To enable the input change notification function, it is necessary to set the range of transmitted inputs with "#T: Trigger AND mask" at our devices.*

- To transfer all inputs, set #T=255
- To disable the input change transmission, set #T=0

Sub option 55 (37 hex)

This command is used by the I/O Controller for periodic transmission of input states. This command does not expect a response, therefore there is no value for 155 in the table. The notification itself is actually an unsolicited response without a matching request. It is an equivalent of Sub option 52 command, intended for periodic transmission.

NVT control examples

Most NVT commands have a fixed length. So, for example, if a value has a 4-byte format, and the current setting can be read by “setting” a value of 0, this zero needs to be sent as 00 00 00 00 hex.

Setting the output byte

This command sets the output port to AA (10101010 bin).

<IAC>	<SB>	<COM_PORT_OPTION>	<CAS_SET_GPIO>	<byte to output)>	<IAC>	<SE>
FF	FA	2C	33	AA	FF	F0

As a response, the following sequence confirms the port settings:

<IAC>	<SB>	<COM_PORT_OPTION>	<ACS_SET_GPIO>	<byte to output)>	<IAC>	<SE>
FF	FA	2C	97	AA	FF	F0

Reading the inputs

Request for the XX value on GPIO.

Outgoing sequence **FF FA 2C 32 00 FF F0**
I/O Controller returns: **FF FA 2C 96 XX FF F0**

Reading the output

Request for the current value XX of the output register that controls the output pins.

Outgoing sequence **FF FA 2C 32 30 FF F0**
I/O Controller returns: **FF FA 2C 97 XX FF F0**

Resetting the D5 output

Request to set one output bit to logic LOW, without changing other output bits.

Outgoing sequence **FF FA 2C 32 25 FF F0**
I/O Controller returns: **FF FA 2C 97 DF FF F0**

where DF is the actual output port value (depends also on the previous port state). I/O Controller changes only one bit, but the response contains the actual port value.

Input state change notification

Unsolicited notification that reports a change at the inputs of the I/O Controller with which there is an established TCP connection. It can start with the FF FA 2C 32 00 FF F0 sequence – see the more detailed description.

outgoing sequence none
I/O Controller sends: **FF FA 2C 34 XX FF F0**

where **XX** is the current output port value.

Note: *The function works in a similar way as the incoming serial port data. Therefore, in the TCP Client/Server mode (IP address of the remote side is specified) with NVT enabled, the device attempts to establish a connection and send the input state change information whenever a single byte of serial data is received or any digital inputs change. When the TCP connection is established, only the current input state is sent, not the whole history of changes (when the connection was closed)!*

How to change the RS-232 / RS-485 serial port baudrate

Request to change the serial port baudrate.

The current port speed can be retrieved by sending the value of **00 00 00 00** (the response includes the current speed). If any other value is sent, the server uses it to set the serial port speed. When converted to a decimal value, the sequence directly indicates speed in Bauds.

Outgoing sequence **FF FA 2C 01 00 00 00 00 FF F0**
I/O Controller returns: **FF FA 2C 65 00 00 25 80 FF F0**

Converting **00 00 25 80** to decimal => current port speed is **9600 Bd**.

Keep Connection

To keep the connection alive and avoid the 50 second inactivity timeout, use the "**`K: Keep connection`**" option in the device setup. NVT must be enabled for this to work. The connection is kept alive by sending the NOP (**FF F1**) command **from the I/O Controller approximately every 5 s**.

If port sampling is enabled with #T: Trigger AND mask, the device sends the I/O keep command ("FF FA 2C 37 ZZ FF F0") instead of NOP (see #K, #L, #M, #N).

Are You There?

The Telnet standard includes the "Are you there" request to determine if there is a device at the other side. Standard Unix devices usually reply with "Yes".

Outgoing sequence **FF F6**
I/O Controller returns: <WEB51 HW XXX SW XXX SN XXX #0F *OvErr *ParErr *FlErr>

Individual XXX values are often proprietary or can be left out in case of error messages. Sample reply: **<WEB51 HW 4.5 SW 2.3 SN 01A03B #01>**