

# PortStore5 MANUAL



## 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 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

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

# PortStore5

*RS-232 or RS-485 to Ethernet converter + 4 MB serial buffer + system messages by e-mail*

*PortStore5 is an Ethernet to serial converter ("terminal server"), featuring support for full 9-bit RS-232 and RS-485 serial ports and 4096 kB Flash buffer for storing data whenever the TCP connection is unavailable.*

*In the TCP Client/Server mode, PortStore5 automatically establishes a TCP connection and attempts to upload data to the server.*

*If the internal buffer is full, an e-mail alert is sent.*



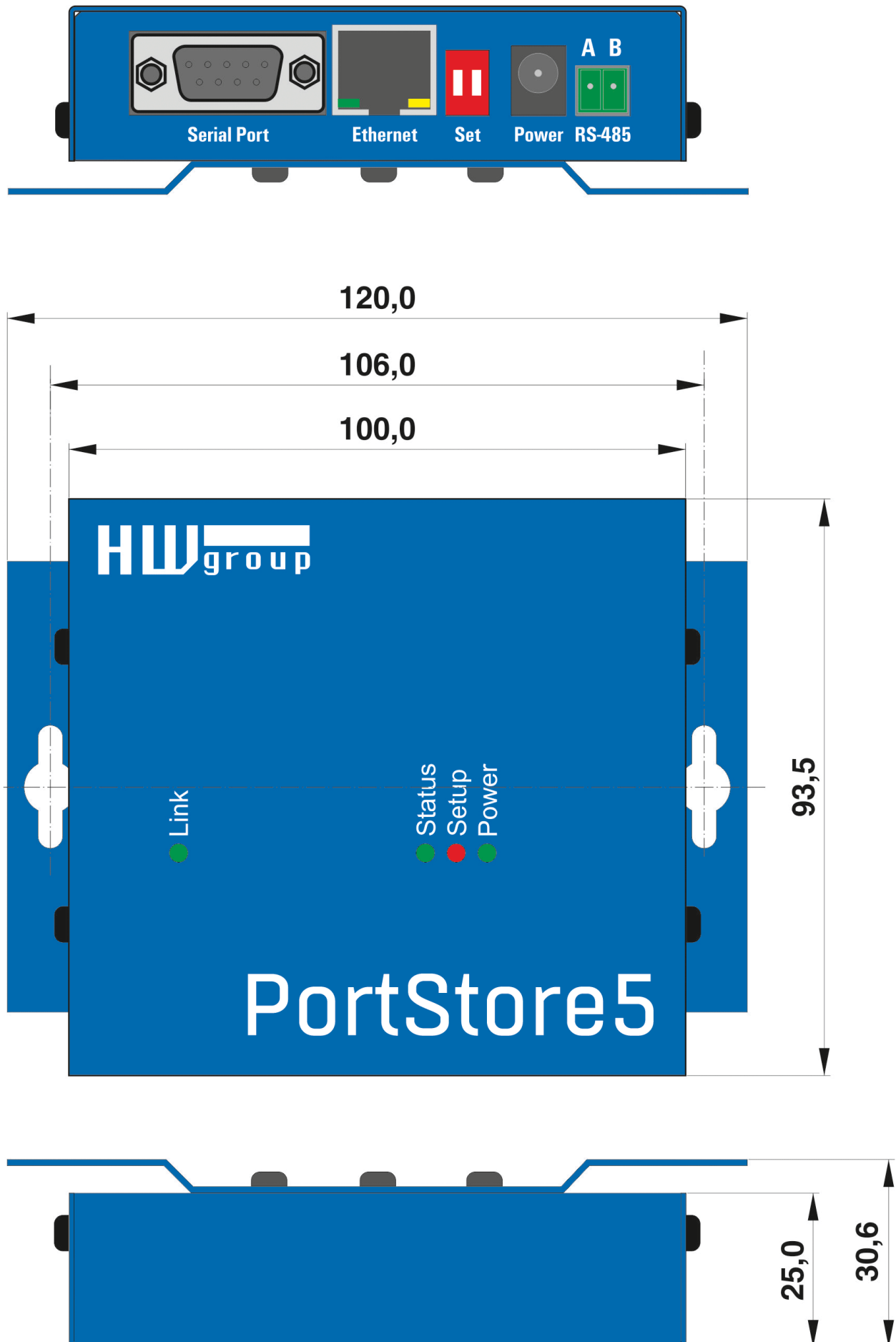
## Basic features

- **1x full RS-232 (9-pin) or RS-485** serial port accessible over the Ethernet.
- Serial port data are stored in a **4096 kB Flash buffer** and uploaded after establishing a TCP connection. When the connection is established, stored data are sent as a continuous "RAW TCP stream".
- The remote port can be controlled with a **virtual driver for Windows** just like, for example, **COM 5** (a Windows XP / Vista / Windows 7 / Windows 8 / Server 2003 / Server 2008 / x64 driver is available free of charge). Compatible with RFC2217.
- 100 Mbps Ethernet interface – 100BASE-Tx, RJ45.
- Support for TCP/IP terminal, TELNET – NVT type (*Network Virtual Terminal*).
- **Two devices** can „tunnel“ the serial port over the Ethernet.
- **Web-based configuration interface.**
- Wide range of supported **serial interface** parameters:
  - 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)
- 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 to 115.2 kBd – entire range
+ Data flow control	XON/XOFF, CTS/RTS, None
+ Interface	1x DB9M (RxD, TxD, CTS, RTS, DSR, DTR, RI, CD, GND)
+ Implemented RS-232 signals	RxD, TxD, CTS, RTS, DSR, DTR, RI, CD
RS-485 serial port	
+ Termination	None (We recommend external 120-470Ω termination)
+ Isolation	RS-485 line not optocoupled to the device's power supply - electrically isolated RS-232/485 to Ethernet (1000 V)
Ethernet port	
+ RS-232 Buffer memory	4096 kB Flash for incoming serial data
+ Buffer overflow signaling	<b>Network:</b> None <b>RS-232:</b> HW or SW handshake if buffer is full
+ Buffer type	Linear FIFO data space
+ Buffer data readout format	RAW binary TCP/IP stream
Ethernet port	
+ Interface	RJ45 100BASE-T / 100 Mbit/s
+ Compatibility	Ethernet: Version 2.0/IEEE 802.3
+ Supported protocols	IP: ARP, TCP + NVT (Network Virtual Terminal)
+ TCP connection closing	Data – 5s timeout (with NVT – ACK/NOP support)
Physical parameters & Environment	
+ Temperature & humidity	Operating: -10 to 75°C (14 to 167 °F) Storage: -10 to 85°C (14 to 185°F), humidity 5 to 95 %
+ Power supply requirements	8-30V / Max. device current consumption 200 mA DC
	- barrel (coaxial) power connector, GND on the shield
+ Dimensions / Weight	120 x 94 x 25 [mm] (H x W x D) / 260 g

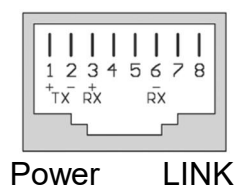
*Note: This parameter table is only indicative.  
For a valid table, see the specifications for the particular device model.*



## Connectors



### Connector pinouts



RS-485 Port	
Pin	Signal
A	+ (A>B)
B	-

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

DIP 1	DIP 2	Functions
ON	OFF	<b>RS-232 Setup</b> mode (9600 8N1) Ethernet is disabled
OFF	OFF	<b>RS-232</b> serial port mode
OFF	ON	<b>RS-485</b> serial port mode - check the &R and &H parameters (recommended: &R3 &H1)

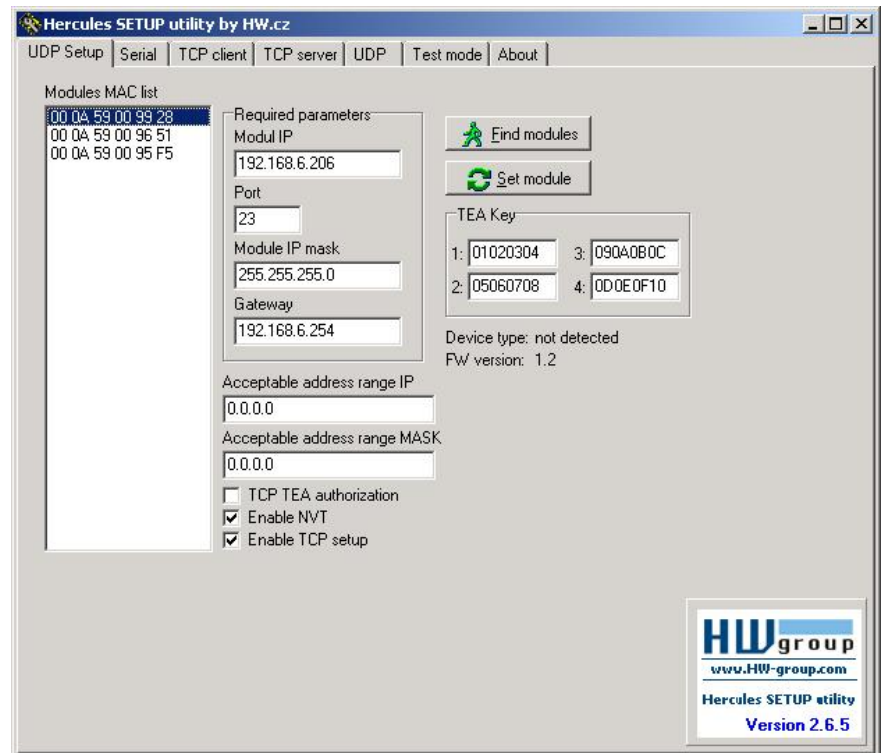
### LEDs:

- Power** – green..... External power connected
- LINK** – amber..... Ethernet activity
- Status** – green..... Device starting up, firmware upgrade
- Setup** – red..... Device in Serial Setup mode

## Quick SETUP

### Connections

- Connect the supplied power adapter to an electrical outlet.
- Set **DIP1** and **DIP2** to **OFF**.
- Connect the device to the Ethernet.
- 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

- Start the "**Hercules.exe**" utility. It is available on the DVD (included in the "set" package) or for free download at [www.HW-group.com](http://www.HW-group.com).
- In the "**UDP Setup**" tab, click "**Find modules**". The MAC address of the device appears in the left column. Single-click the MAC address and set the required parameters (at least the IP address, mask and gateway).
- Make sure that "Enable TCP Setup" is checked. 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.

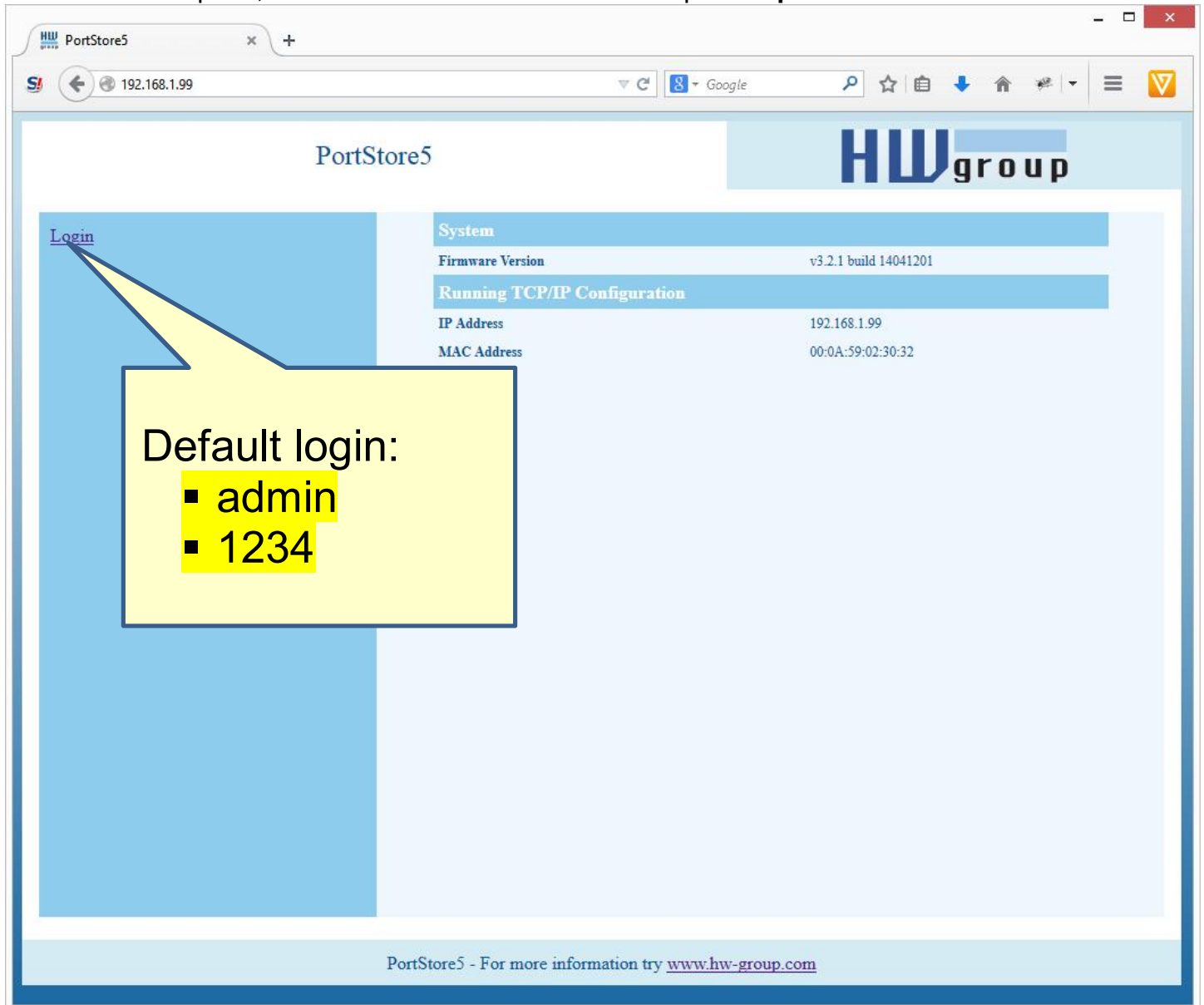
### MAC address not present in the list?

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

- Are you using a correct TP cable (straight-wired TP Patch when connecting to an Ethernet switch, crossover cable when connecting to a PC)?
- Check the DIP switch settings (all should be OFF).
- Check your power adapter, make sure the Power LED is on.
- Check the firewall settings at your PC.

## Configuring the device over WWW

In the UDP Setup tab, click the correct MAC address and press **Open in the WEB browser**.

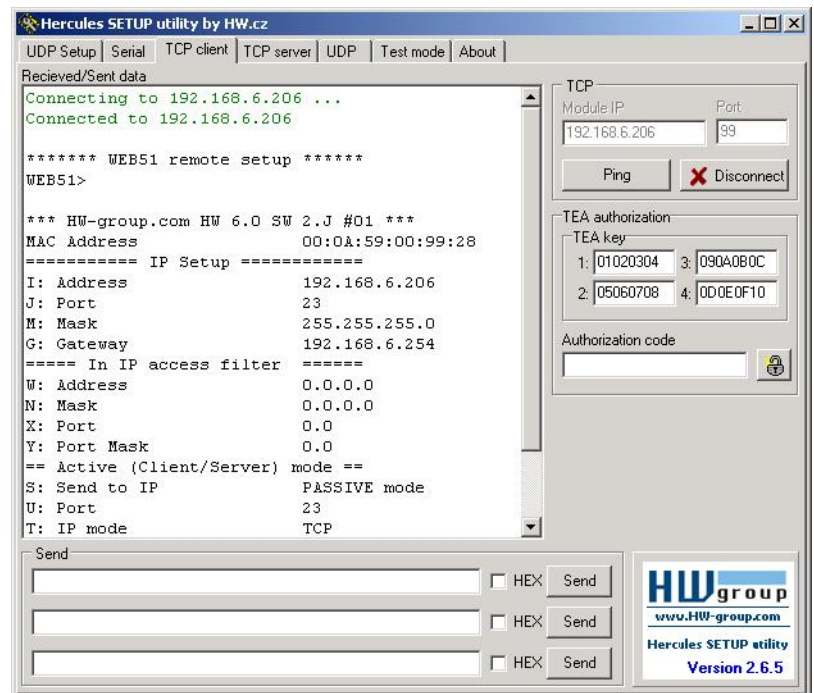


You need to log in to access the configuration.  
The default credentials are “**admin**” + “**1234**”.



## Setting up the device using TCP Setup

- 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, “**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.

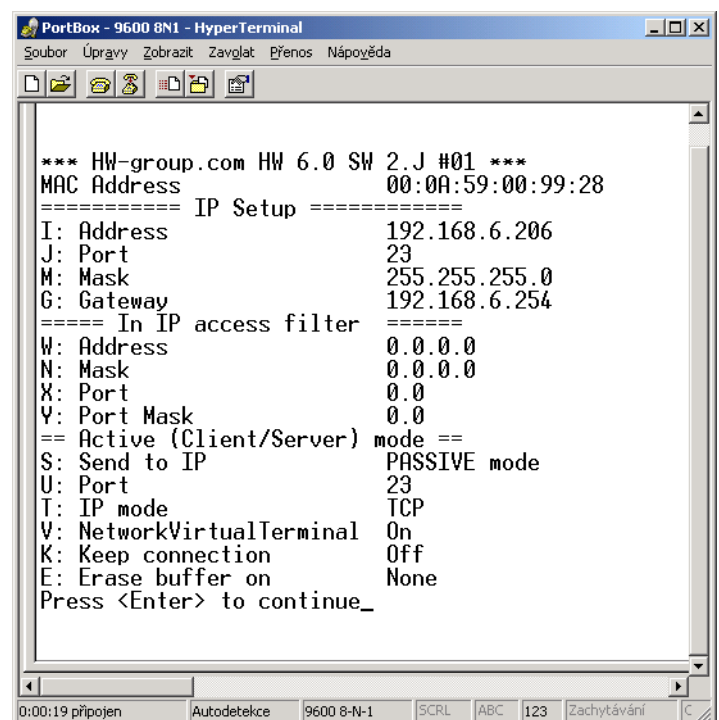


## Setting up the device using a RS-232 terminal

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 lights 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.



## 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.
- **5-second timeout is too short to work with, is it possible to do something about it?**  
Yes, enable “Keep connection” in the Setup. PortStore5 then sends a command every 6 seconds that appears as “**NVT: NOF**” in the receiving pane. This prevents the connection from timing out.

## HW VSP – Virtual Serial Port

*Virtual serial port driver for Windows 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.*

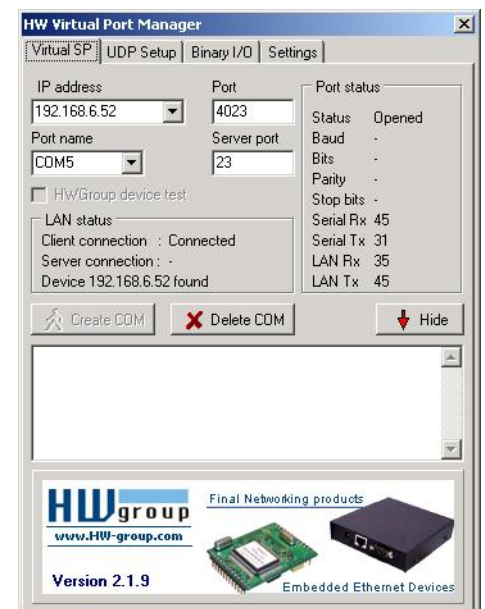


- The driver works in Windows 8, Windows 7, Windows XP, Vista, Windows 2003 Server, Windows 2008 Server, including 64-bit versions.
- If the device supports RFC 2217 (NVT), you can set the remote serial port parameters (speed, parity, stop bits).
- The communication can be recorded to a log file for easier debugging.
- 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 PortStore5

- Install HW VSP (“**HW\_VirtualSerialPort**” directory on our CD). When installing, make sure to check the option to install the second part. After installation, make sure to restart your PC.
- Make sure that NVT is enabled at the **UDP Setup** tab in the **Hercules** utility. If not, enable NVT and save the settings.
- Start HW VSP and find the device at the “**UDPsetup**” tab. Select the MAC address of the device and click “Use this IP”. 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.
- Select a serial port to create from the COM1 to COM20 range 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 PortStore5, configures the remote port (speed, parity, number of bits, handshake) to match 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 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.*



## HW VSP settings

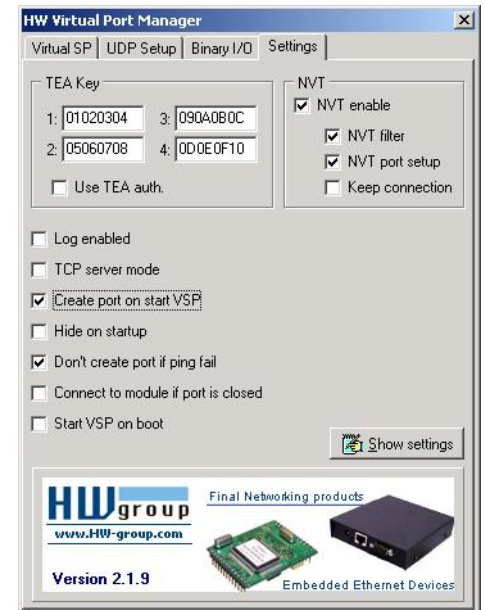
### TEA Key pane

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

### NVT pane

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

- **NVT filter** – Filters out NVT control characters from the data flow.
- **NVT port setup** – Configures the remote port parameters via control commands to match 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) to change the baudrate of the remote TCP/IP serial port.
- **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 – 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 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 module if port 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.
- **Start VSP on boot**  
Inserts the VSP path to the RUN key (HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\run) in the Windows registry. VSP is then started every time Windows starts.

# Configuring PortStore5 – Command description

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

## PortStore status page

Shows an overview of the basic device parameters.

PortStore5 Status

HW group

[Status](#)

[Network](#)

[TCP mode setup](#)

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

[Serial](#)

[SMTP setup](#)

[System](#)

[Save setup & restart PortStore5](#)

**System**

Firmware Version v3.2.1 build 14041201

**Running TCP/IP Configuration**

IP Address 192.168.1.99

Subnet Mask 255.255.252.0

Default Gateway 192.168.1.254

Send to IP (0.0.0.0 - PASSIVE mode) 0.0.0.0

MAC Address 00:0A:59:02:30:32

PortStore5 - For more information try [www.hw-group.com](http://www.hw-group.com)

## Network parameters

PortStore5 Network Setup

HW group

[Status](#)

[Network](#)

[TCP mode setup](#)

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

[Serial](#)

[SMTP setup](#)

[System](#)

[Save setup & restart PortStore5](#)

**IP Setup**

I: Address

M: Mask

G: Gateway

**In IP access filter**

W: Address

N: Mask

X: Port

Y: Port Mask

**Channel 0**

T: IP mode

J: Port

**Active (Client/Server) mode**

S: Send to IP (0.0.0.0 - PASIVE mode)

U: Port

PortStore5 - 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. 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 device.

**J: Port                    23**

Configures the device'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 device'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 device in “**In IP Setup**”. For debugging, we therefore recommend to keep this parameter set to 0.0.0.0.

**(IP attempting to access ANDN ) = 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, PortStore5 works as a Client/Server in the TCP mode.

This means that if **no TCP/IP connection is established** and the device receives serial port data (even a single byte), the device establish a TCP socket to defined IP address as TCP client.

The device remains in the **TCP Server** mode in between attempts.

In the UDP mode, PortStore5 sends data from the serial port to this address/port.  
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 are sent only to the specified address.

## **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:

- $n < 16 \dots n * 0.2 \text{ s}$
- $n < 32 \dots (n - 15) * 0.8 \text{ s} + 3.0 \text{ s}$
- $n < 64 \dots (n - 31) * 3.2 \text{ s} + 15.8 \text{ s}$
- $n < 128 \dots (n - 63) * 12.8 \text{ s} + 118.2 \text{ s}$
- $n < 256 \dots (n - 127) * 51.2 \text{ s} + 937.4 \text{ s}$

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

### 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).***

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



## TCP mode settings

PortStore5 TCP Mode Setup

192.168.1.99/tcp.shtml?Q=ASXBxyEF

### PortStore5 Network Setup

HW group

[Status](#)

[Network](#)

[TCP mode setup](#)

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

[Serial](#)

[SMTP setup](#)

[System](#)

[Save setup & restart PortStore5](#)

#### TCP Mode Setup

P: IP Active Call Timer (65535 - Off)

B: IP Protocol Retry

V: NetworkVirtualTerminal

K: Keep connection

E: Erase buffer on

PortStore5 - For more information try [www.hw-group.com](http://www.hw-group.com)

**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.

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

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

Challenge-response protocols respond up to 40% faster.

**0: TCP**

**1: UDP with broadcast Rcv Off**

**2: UDP with broadcast Rcv On**

**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)

**K: Keep connection****Off**

Keeps TCP connections alive to avoid automatic termination upon timeout specified by the **B: IP protocol retry time** parameter. 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, PortStore5 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**

## 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.

PortStore5 UDP Mode Setup

HW group

[Status](#)

[Network](#)

[TCP mode setup](#)

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

[Serial](#)

[SMTP setup](#)

[System](#)

[Save setup & restart PortStore5](#)

**Triggering Setup**

\*L: Trigger Length(0..4)

\*P: Post Trigger Length

\*S: Start Trigger Pattern

\*M: Start Trigger Mask

\*X: Stop Trigger Pattern

\*Y: Stop Trigger Mask

\*E: Max. Start-Stop Length

**UDP/IP Mode Setup**

A: Max. Packet Length

PortStore5 - For more information try [www.hw-group.com](http://www.hw-group.com)

## ==== Trigerring 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.

### \*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.

#### 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

## ==== UDP/IP Mode Setup =====

### A: Max. Packet Length 250

Maximum UDP packet length in bytes.

## Serial port parameters

PortStore5 Serial Setup

Serial Setup

&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 continuously asserted [~+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

&F: Flash buffer mode Off

%S: TCP/IP setup On

Send

[Save setup & restart PortStore5](#)

PortStore5 - 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 “&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 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 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****&N: Multidrop network address** **0****&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 PortStore5 can no longer accept data.

- 1: none** – No control, see &R for RTS
- 2: RTS/CTS** – RTS/CTS control pins
- 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 &R3 – TxRTS 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.

**0: FULL duplex** (RS-232)

**1: HALF duplex** (RS-485)

**&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)

## SMTP setup

PortStore5 e-mail Setup

**Out IP Setup**

@S: SMTP server IP (0.0.0.0 - off SMTP)

@U: SMTP server Port

**e-mail Setup**

@G: From e-mail address

@N: System Name

**Power Fail e-mail**

@U: To e-mail address

@X: e-mail Subject

**Buffer Full e-mail**

@V: To e-mail address

@Y: e-mail Subject

**Buffer Overflow e-mail**

@W: To e-mail address

@Z: e-mail Subject

[Save setup & restart PortStore5](#)

PortStore5 - For more information try [www.hw-group.com](http://www.hw-group.com)

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

**@S: SMTP server IP**            **192.168.6.254**

**@U: SMTP Port**                **25**

If the address in the @S option is different from 0.0.0.0, PortStore works in the TCP Server + SMTP Email Client mode. In the 3 scenarios below, e-mails are sent according to defined conditions.

- If there was a **power failure**, PortStore sends a device restart notification to the specified e-mail address as soon as the power is restored. The **@U** and **@X** parameters are used.
- If the input buffer is more than **75% full**, an e-mail is sent to the specified address and with the specified subject (**@V** and **@Y**).
- If the input buffer is more than about **92% full**, an e-mail is sent to the specified address and with the specified subject (**@W** and **@Z**).



**===== e-mail Setup =====****@G: From e-mail address:** [PortStore@company.com](mailto:PortStore@company.com)**@N: System Name:** WEB51

Sender's e-mail, needs to be filled in because of mail programs that process the e-mail along the way. The device is not capable of receiving or processing of e-mails sent to this address.

**=== Power Fail e-mail****@U: To e-mail address** [ResponsiblePerson@company.com](mailto:ResponsiblePerson@company.com)**@X: e-mail Subject** PortStore Power-Up init

Defines e-mail address and subject to be sent when the device is turned on after a power failure.

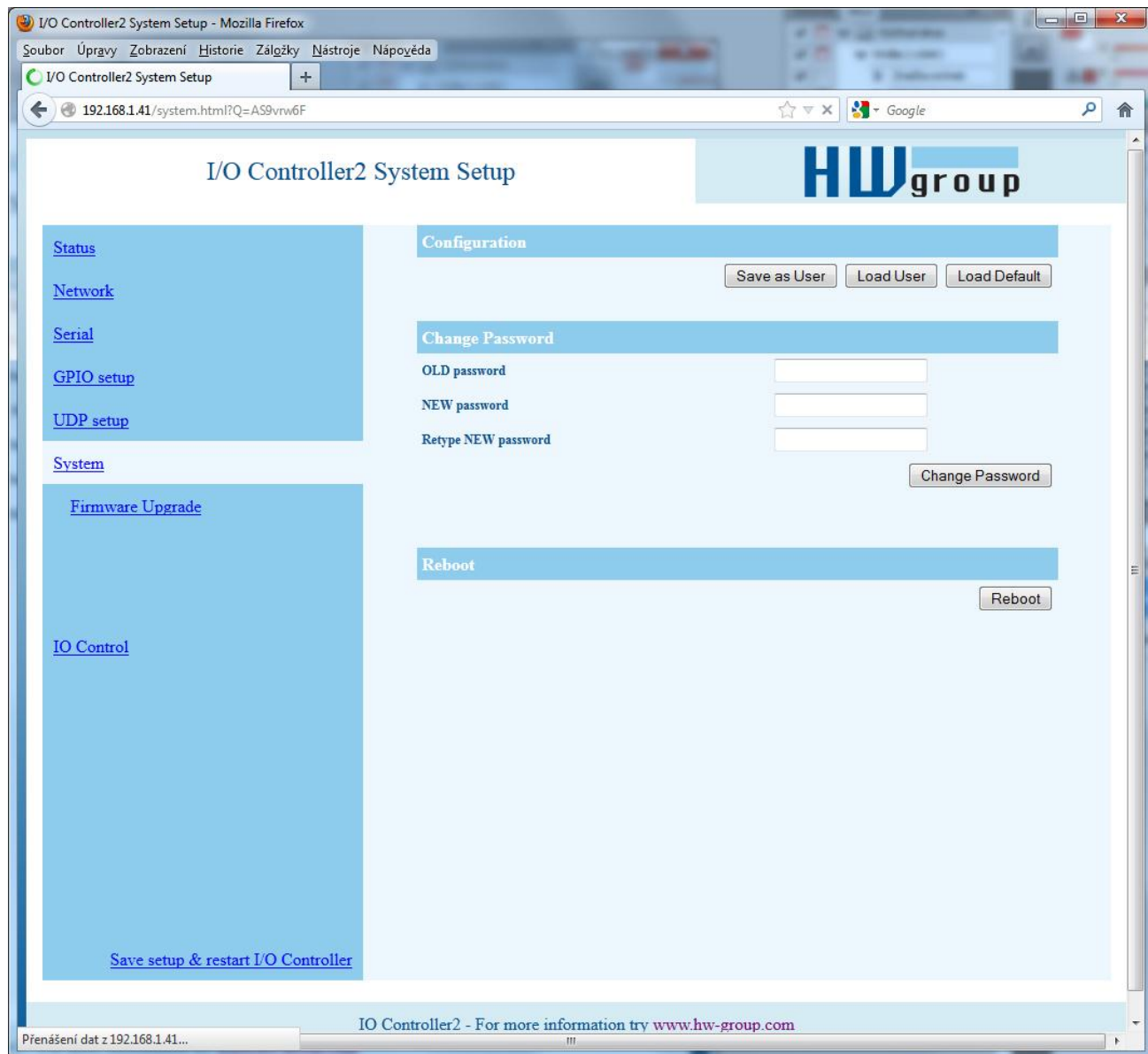
**=== Buffer Full e-mail****@V: To e-mail address** [ResponsiblePerson@company.com](mailto:ResponsiblePerson@company.com)**@Y: e-mail Subject** PortStore buffer almost FULL

Defines e-mail address and subject to be sent when the data buffer is about 75% full (75% out of 2048 kb = about 1536 kb).

**=== Buffer Overflow e-mail****&W: To e-mail address** [ResponsibleChief@company.com](mailto:ResponsibleChief@company.com)**&Z: e-mail Subject** PortStore buffer TOTAL FULL!

Defines e-mail address and subject to be sent when the data buffer is about 90% full (87.5% out of 4096 kb = about 3584 kb).

# System



## ===== 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 changing parameters in TCP Setup.

### Change Password

Applies only to WWW access. The default password is 1234.

## UDP/IP mode settings

### 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 value shown above, 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

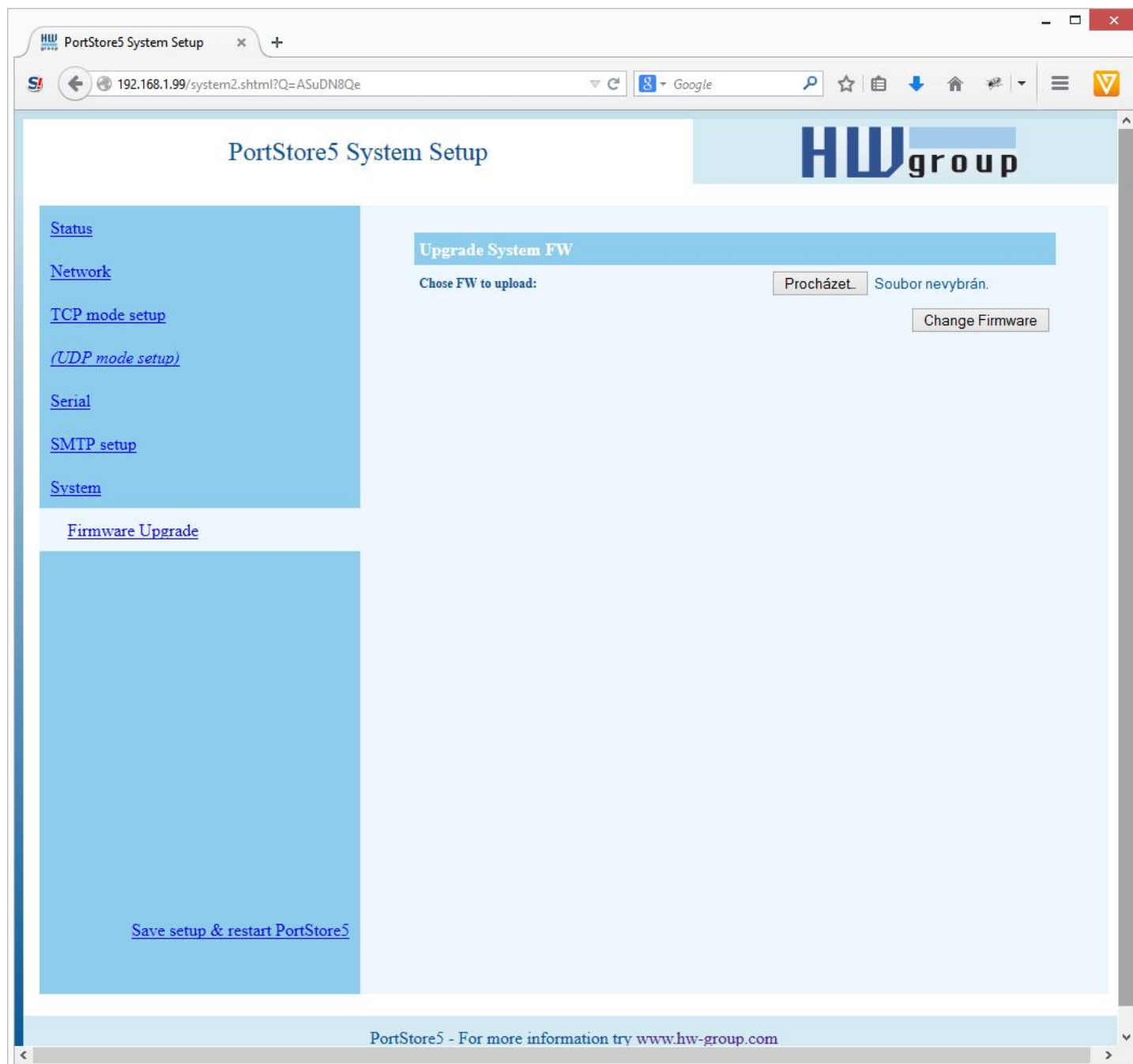
Quiet mode is useful when the configuration parameters need to be handled automatically, with a script. Quiet mode is enabled with “**Q1**”. After pressing Enter, the device responds with a list of parameters in the following format:

```
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;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;#Y0;#Z3;#W0;
*L1;*P0;*S58.0.0.0;*M255.0.0.0;*X10.0.0.0;*Y255.0.0.0;*E999
```

## What's new in the WEB interface

### Firmware upgrade

Upgrades the current device firmware. This operation may change some settings.



## Default settings

```

*** HW-group.com HW 6.0 SW 3.1.w #01 ***
***      PortStore I/O      ***

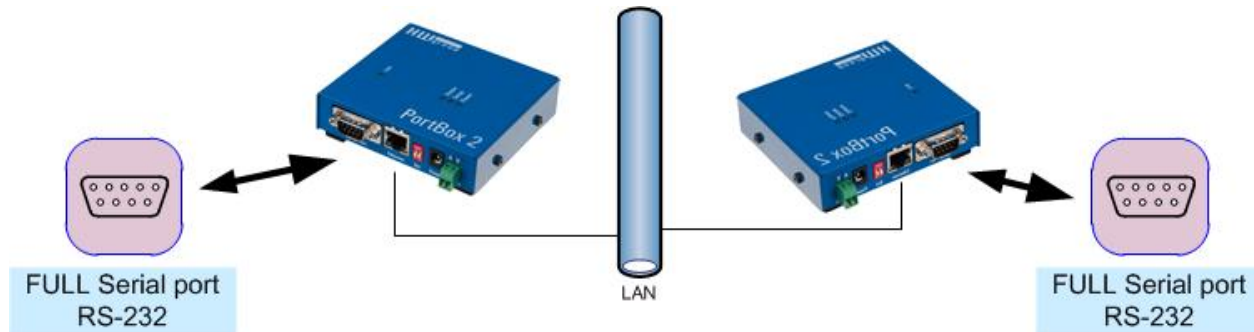
MAC Address          00:0A:59:01:F5:F5
===== IP Setup =====
I: Address           192.168.1.41
J: Port              23
M: Mask              255.255.252.0
G: Gateway           192.168.1.254
===== In IP access filter =====
W: Address           0.0.0.0
N: Mask              0.0.0.0
X: Port              0.0
Y: Port Mask         0.0
== Active (Client/Server) mode ==
S: Send to IP        PASSIVE mode
U: Port              23
B: IP Protocol Retry 35
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: RTS Output        continuously asserted [~ +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
Press <Enter> to continue
===== Other =====
D: Load/Save Settings from/to Flash
R: Reboot

```

## Practical configuration examples

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

### 2x PortStore5 back-to-back (Box-2-Box)



Serial port extension. The serial port is configured for 19000 Bd, 8N1, SW handshake.

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.

#### PortStore5 – unit 1

```
*** HW-group.com HW 6.0 SW 3.1.w #01 ***
*** PortStore I/O ***

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
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 continuously asserted [~
+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
===== Other =====
D: Load/Save Settings from/to Flash
R: Reboot

WEB51>
```

#### PortStore5 – unit 2

```
*** HW-group.com HW 6.0 SW 3.1.w #01 ***
*** PortStore I/O ***

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
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 continuously asserted [~
+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
===== Other =====
D: Load/Save Settings from/to Flash
R: Reboot

WEB51>
```

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

If security is an essential requirement and the device should only communicate within the local network, use the following settings. In this case, only communication within the LAN 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

## Typical applications

PortStore5 is typically used to enable remote access and data capture for technology that is controlled over RS-232. When the TCP/IP connection is unavailable, the device stores incoming serial port data to its internal **4096 kB Flash** memory.

If the PortStore is configured as a TCP Client, it attempts to establish a TCP/IP connection and upload the data from the buffer. Data are sent in a “RAW” format in the order in which it was received. After that, PortStore behaves as a standard terminal server with remote serial port access.

### Application examples

- **PBX systems** – collecting billing data from a branch exchange

The PBX usually provides billing data – which extension called what number at what time, and how long did the call last. It is often desirable to store this data for a later analysis in special software. The PBX sends the data to a RS232 serial port. However, it usually does not have enough buffer memory if the receiving device shuts down.

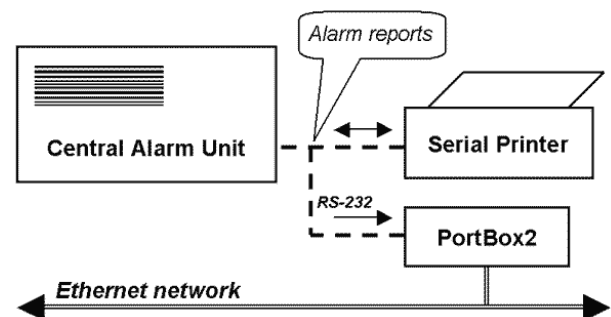


**PortStore5** is used to receive this data over RS-232 and store it into the internal Flash buffer. Once in a while, the special software downloads the data over the Ethernet and processes it. It is often the case that the server downloading the data is at the headquarters, and the branches are all over the country and interconnected only with a computer network. The buffer capacity is sufficient for 35,000 typical lines.

- **Storing data from a serial printer output**

A central security alarm unit in a building prints basic information about the premises on a serial printer. It is desirable to archive this output. However, a printer at a branch office needs to be attended by personnel with at least some training, and the data is not available electronically at the headquarters.

**PortStore5** stores alarm data into the internal Flash buffer. A server at the headquarters regularly downloads and archives the data. The buffer capacity is sufficient for 35,000 typical lines.



- Card readers, RFID
- Remote power control
- Displays in exchanges
- Security alarms
- Access control systems
- UPS control and logging devices
- Data loggers
- CNC machines
- Barcode scanners



## NVT control

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

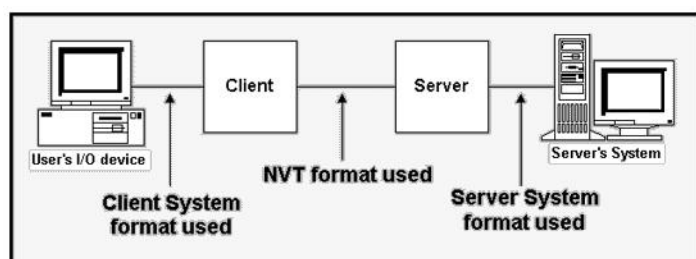
[http://www.hw-group.com/support/nvt/index\\_en.html](http://www.hw-group.com/support/nvt/index_en.html)

### 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 Pin Input/Output) functions listed below. These functions can control the digital input and output pins of PortStore5. 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**).

## 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
```

### Changing the RS-232 / RS-485 serial port baudrate

*Request to change the serial port communication speed.*

The current port speed can be retrieved by sending the value of **00 00 00 00** (the response contains 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 value directly indicates the communication speed in Bauds.

```
Outgoing sequence:    FF FA 2C 01 00 00 00 00 FF F0
PortStore5 returns:   FF FA 2C 65 00 00 25 80 FF F0
```

Converting **00 00 25 80** to decimal, the current speed of **9600 Bd** is obtained.

### 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 a NOP command (**FF F1**) from the PortStore5 approximately every 5s.

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

### Are You There?

The Telnet standard supports an “Are you there” request to determine if there is a live device at the other side. Standard Unix devices usually reply with “Yes”.

```
Outgoing sequence:    FF F6
PortStore5 returns:   <WEB51 HW XXX SW XXX SN XXX #0F *OvErr *ParErr *FIErr>
```

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>**

# Basic types of communication

*Very brief overview of basic concepts. For a detailed version with a list of abbreviations etc., see our website (under "Support, Download").*

## TCP Server (passive mode)

When the device is turned on, it listens at the specified port and waits for a client to connect. When the client connects, data from the Ethernet are sent over the serial line, and vice versa, data from the serial line are sent over the Ethernet. When no client is connected and there are incoming data from the serial line, the data are stored to the buffer (configurable size) and sent immediately after reconnection (unless the option to clear the buffer when a client connects is configured in SETUP).

## TCP Client (active mode)

The device acts as a server (described above). In addition, when there are incoming data from the serial line, it actively tries to establish a connection as a client with a specified remote IP address and transfer the data.

If that fails, data are stored into the buffer and transferred whenever a connection is established (regardless of whether the converter is a client or a server).

## UDP

Data from the Ethernet are sent to the serial line. Incoming data from the RS232 serial line are assembled into packets according to configured rules and sent to the configured IP address. In UDP mode, there are no acknowledgment checks; the application itself must tolerate data loss.

The advantage is a somewhat faster response time, which is particularly useful for RS-485 lines.

## NVT (Network Virtual Terminal)

When using TCP/IP communication, the device capabilities include a NVT (RFC2217) compliant control stream that can be used e.g. to change the serial port baudrate, control inputs and outputs or clear the buffer. These control commands are included in the data stream and prefixed with "FF" (data bytes of this value must then be doubled). For a detailed NVT description, see the "Programming Ethernet applications" guide (which also describes the freely supplied communication subroutines) or a separate article at our website.

## VSP (virtual serial port)

The device is in the TCP Server mode and the PC behaves as a TCP client. The [HW VSP](#) software is installed at the PC in order to create a virtual serial port (e.g. COM6). The serial port at the device then acts as a physical port at your PC, even though it may be far away.

When buffering is enabled, data are again stored to the internal buffer whenever the TCP connection is disconnected, and transferred after the connection is established.

If, in your scenario, it is more advantageous to initiate the connection by the device and per-packet data charges apply (e.g. GPRS), it is possible to configure the device as a TCP Client/Server and switch the VSP to the TCP Server mode. Connection (bidirectional of course) is then established whenever data are received at the serial port.

## 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>