

I/O Controller PL

*RS-232/485 - Ethernet converter
+ 8 digital inputs and 8 digital outputs*



Contents

| | |
|--|-----------|
| Basic features | 4 |
| TECHNICAL PARAMETERS..... | 5 |
| I/O CONTROLLER – CONNECTOR DESCRIPTION | 6 |
| Connectors wiring..... | 6 |
| | |
| Digital inputs and outputs..... | 7 |
| I/O Controller standard..... | 8 |
| Standard version (connecting block) – mechanical dimensions..... | 9 |
| | |
| QUICK SETUP | 10 |
| | |
| HW VSP – VIRTUAL SERIAL PORT | 14 |
| | |
| I/O CONTROLLER SETUP – COMMANDS DESCRIPTION | 16 |
| Factory defaults | 16 |
| | |
| Device network parameters | 17 |
| TCP and UDP parameters | 18 |
| Serial port parameters..... | 19 |
| Security and remote TCP Setup | 20 |
| Setting up the inputs and outputs control..... | 21 |
| UDP/IP mode settings..... | 25 |
| | |
| Practical examples of the configuration | 26 |
| 2x I/O Controller connected to each other (Box-2-Box)..... | 26 |
| I/O Controller on the Charon 1 module | 27 |
| The input's and output's control using the NVT..... | 28 |
| Commands for GPIO - 50, 51, 52 (32 .. 34 hex) control..... | 28 |
| | |
| NVT control examples | 29 |
| | |
| Ordering number..... | 32 |
| | |
| Free application software | 32 |

I/O Controller PL

Ethernet - RS-232/485 + inputs and outputs

I/O Controller is an ethernet enabled unit containing 8 binary inputs, 8 binary outputs and RS-232/485 serial interface.

All the interfaces are accessible over a network using the M2M protocol on TCP/IP.

Two devices can be connected to each other (the Box-2-Box mode), digital signals and RS-232 are then prolonged over the network.



Basic features

- **1 RS-232 or RS-485 serial port** accessible over Ethernet.
- The remote port can be controlled using a **virtual port Windows driver** as for example **COM 5** RFC2217 compatible (Windows 2000 / XP driver is free).
- Ethernet interface 10 Mbit/s - RJ 45 10BASE-T.
- TCP/IP TELNET terminal - NVT (*Network Virtual Terminal*) support.
- **Two devices** can connect (tunnel) the inputs, outputs and the serial port over the Ethernet network.
- **Possible ways of device configuration:**
 - RS232 local configuration (serial terminal).
 - TCP/IP terminal on port 99.
 - Using simple windows software (with UDP Broadcast).
- **8 optocoupled binary inputs** (common ground).
- **8 binary open-collector outputs** (max. 100 mA / 50V) – common ground.
- **The inputs and outputs are controlled with a M2M protocol** (Machine to Machine) and the **NVT** (Network Virtual Terminal).
- Serial interface versatility
 - Communication speed variable in range 300..115.200 Bd.
 - Handshake (CTS/RTS, Xon/Xoff, none).
 - Support of 7 to 9bit parity (9th bit transfer over Ethernet).
- UDP/IP wide support of question-answer protocols over RS-485.
- **Routine libraries and examples** for MS Visual Basic, Delphi, Borland C++, JAVA, PHP ..

Technical parameters

| Serial port RS-232 | |
|--|---|
| + 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, step = 50 Bd |
| + 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 |
| Serial port RS-485 | |
| + Termination | Selectable internal termination 120Ω / ext. termination |
| + Isolation | RS-485 line not optocoupled to the device's power supply - galvanic isolated RS-232/485 to Ethernet (1.000 V) |
| Binary digital Inputs & Outputs | |
| + Input type | Binary optocoupled |
| + Log. 0 voltage | 0 .. 3V |
| + Log. 1 voltage treshold / "on" current | from <u>5V / 5mA</u> to <u>20V / 25mA</u> |
| + Max. input Voltage and Current | up to 40V / 50 mA / 1 second |
| + Isolation Voltage | max. 50V to power supply |
| + Sampling period | 10 ms |
| + Inputs and Outputs SW control | Inputs and Outputs controlled over M2M protocol, based on the NVT (extended RFC2217) – short 6 bytes binary commands over TCP/IP |
| + Output type | Darlington transistors with common emitter, suppression diodes |
| + Output voltage | max. 50V |
| + Log. 1 output current | max. 500 mA / 1 output and max. 1.500 mA / all 8 outputs |
| Ethernet port | |
| + Interface | RJ45 (10BASE-T) – 10 Mbit or 10/100 Mbit network compatible only! |
| + Compatibility | Ethernet: Version 2.0/IEEE 802.3 |
| + Supported protocols | IP: ARP, TCP + NVT (Network Virtual Terminal) |
| + TCP connection closing | Data - timeout 50s (with NVT - ACK/NOP support) |
| Physical parameters & Environment | |
| + Temperature & Humidity | Operating: +5 .. 65 °C / Storage: -10 to 85 °C, Humidity 5 to 95 % |
| + Power supply requirements | 8-24V / Max. device current consumption 200 mA DC - barrel (coaxial) power connector, GND on the shield |
| + Dimensions / Weight | 28 x 105 x 135 [mm] (H x W x D) / 395 g |
| Functional parameters | |
| Device SETUP configuration options | RS-232 Setup - over any RS-232 terminal TCP Setup - over TCP/IP - any telnet terminal on the TCP/IP 99 port UDP Setup - network parameters via UDP using the Hercules SETUP |
| Diagnostic LEDs | Power (<i>green</i>), Link & Activity (<i>yellow</i>) |

I/O Controller – connector description

A sturdy metal box for wall mounting (“L” profiles can be ordered separately), surface placement, or mounting to a DIN molding (depending on the ordering number).



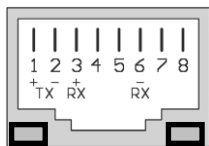
RS-232 and RS-485 interfaces are connected to the Cannon DB9M connector.

Signalization:

- Power** – green on RJ-45 connector..... External power source
- LINK** – yellow on RJ-45 connector LINK & Ethernet Activity
- RxD** – red Receiving data on RS-232
- TxD** – green Transmitting data on RS-232

Connectors wiring

Ethernet:



Power LINK

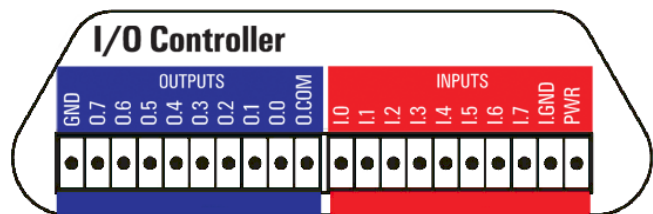
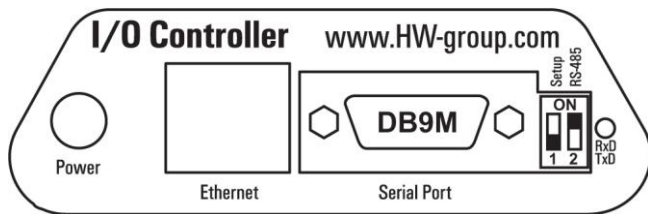
Power source:



DC 8...20V / max. 0.3 A

| I/O Controller Port | |
|---------------------|-----------|
| Pin | Signal |
| 1 <> | „A” RS485 |
| 2 <- | IN RxD |
| 3 -> | OUT TxD |
| 4 | |
| 5 -- | GND |
| 6 | |
| 7 -> | OUT RTS |
| 8 <- | IN CTS |
| 9 <> | „B” RS485 |
| Cannon 9 - Male | |

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



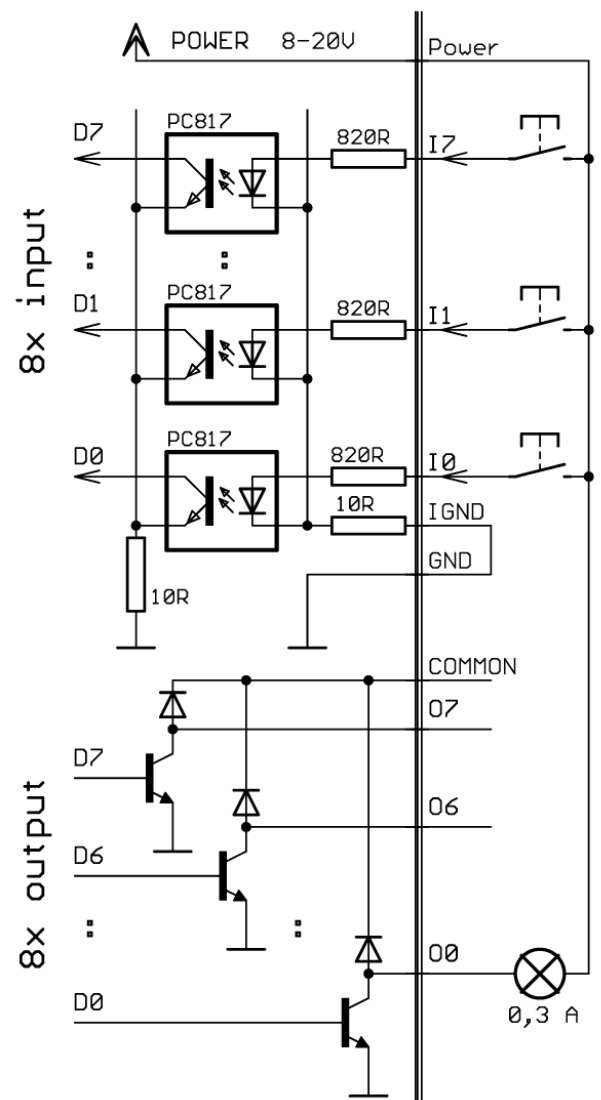
| DIP 1 | DIP 2 | Function |
|-------|-------|---|
| ON | OFF | RS-232 Setup mode (9600 8N1) <i>Ethernet interface is disabled!</i> |
| OFF | OFF | Serial port is in RS-232 mode |
| OFF | ON | Serial port is in RS-485 mode - Check the &R and &H parameters (we recommend &R3 &H1) |

Digital inputs and outputs

The inputs and outputs are connected according to the following table.

- **GND** – device ground.
- **PWR** – device power (max. 200 mA).
The device can be powered from a standard coaxial cable or from the I/O connector (200 mA internal fuse = suitable only for testing purposes).
- **I.GND** – common ground for optocouplers.
- **I.0 to I.7 - 8 optocoupled inputs 5-15V (Max. current on the Power pin is 200 mA!)**
All 8 inputs are realized using common ground optocouplers (IGND pin). The input voltage range is 5 -15 V.
The pins can then be controlled using the contacts together with the POWER pin, which has the power adaptor voltage (connect IGND with GND).
- **O.COM (Common)** – Overvoltage output connected to the plus pole of the power source. This can be useful when you want to control a relay from this output.
- **O.0 to O.7 - 8 open collector outputs, with common diode overvoltage protection.**
The outputs are realized using 8 open collector transistors. Two outputs are internally connected to a relay, which contacts are also led out to the device's connector. The protecting diodes are connected to the "common" pin, which should be connected to the plus pole of the next device (for example a relay). This way the pins can be protected against load peaks.

| | Name | Description | Can37F | standard |
|------------|--------------|------------------------|--------|----------|
| 8x INPUTS | Power | Ext. power supply | 28 | 1 |
| | GND | Ground | 29, 20 | 20 |
| | I0 | Input 0 (5-15V) | 16 | 10 |
| | I1 | Input 1 (5-15V) | 15 | 9 |
| | I2 | Input 2 (5-15V) | 14 | 8 |
| | I3 | Input 3 (5-15V) | 13 | 7 |
| | I4 | Input 4 (5-15V) | 12 | 6 |
| | I5 | Input 5 (5-15V) | 11 | 5 |
| | I6 | Input 6 (5-15V) | 10 | 4 |
| | I7 | Input 7 (5-15V) | 09 | 3 |
| | I.GND | Optocouplers ground | 08 | 2 |
| 8x OUTPUTS | O.COM | Common wheeling diodes | 33 | 11 |
| | O0 | Output transistor 0 | 05 | 12 |
| | O1 | Output transistor 1 | 24 | 13 |
| | O2 | Output transistor 2 | 04 | 14 |
| | O3 | Output transistor 3 | 23 | 15 |
| | O4 | Output transistor 4 | 03 | 16 |
| | O5 | Output transistor 5 | 22 | 17 |
| | O6 | Output transistor 6 | 02 | 18 |
| | O7 | Output transistor 7 | 21 | 19 |

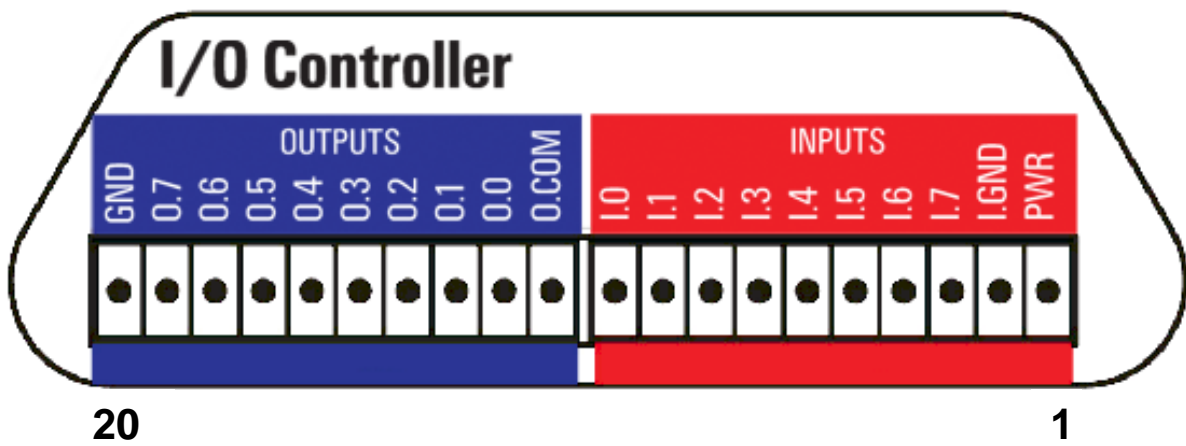


I/O Controller standard

The binary inputs and outputs are led out to the rear side of the device.

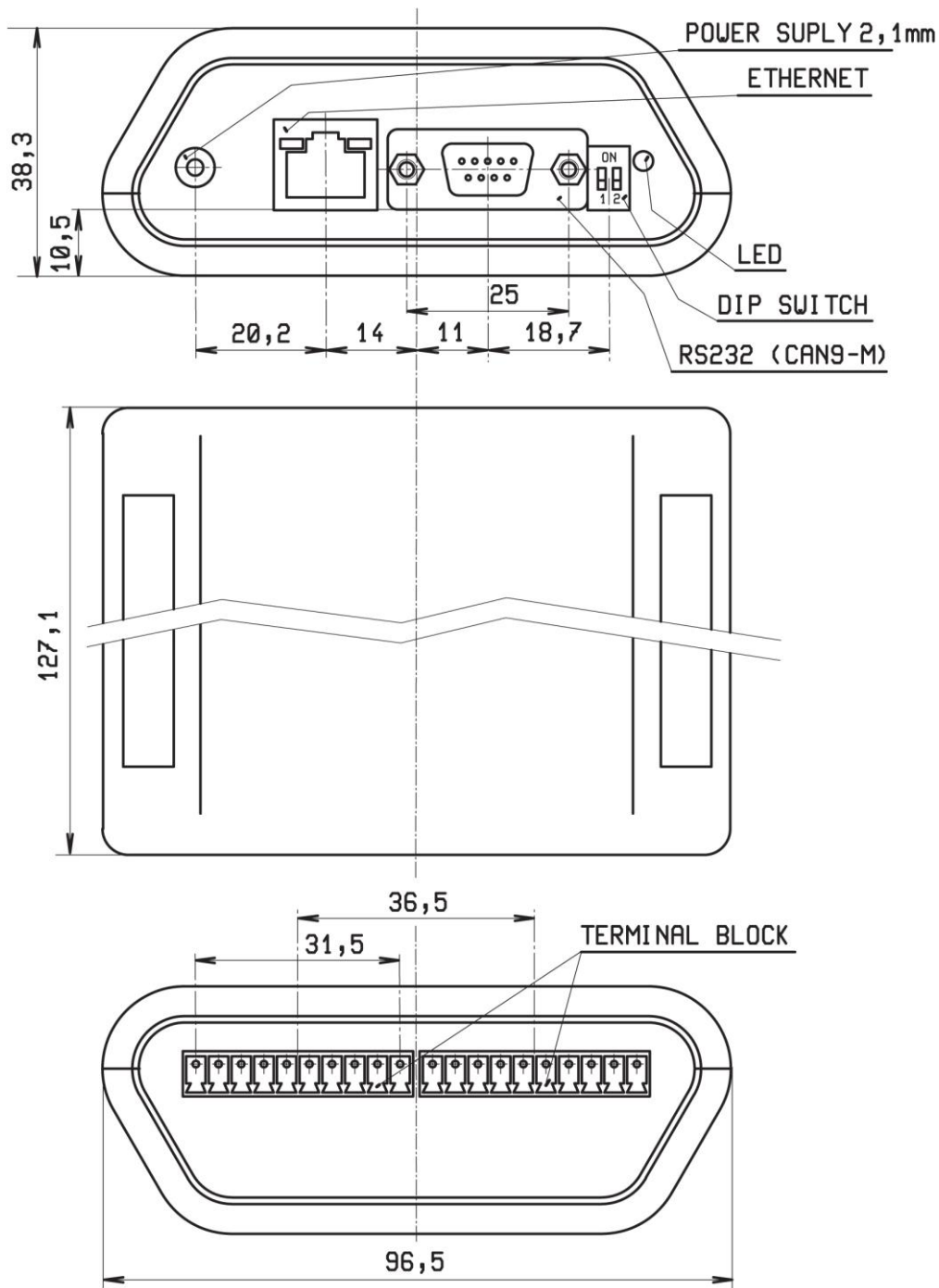
Two dismountable connecting blocks (10 pins each, with 3.5 mm pins gap) are used for connection.

One connector set is part of the shipment (removable connecting block 2x 10 pins).



| | | Name | Description | |
|-------------------|--------------|------------------------|---------------------|----|
| | | Power | Ext. power supply | 1 |
| | | GND | Ground | 20 |
| 8x INPUTS | I0 | Input 0 (5-15V) | 10 | |
| | I1 | Input 1 (5-15V) | 9 | |
| | I2 | Input 2 (5-15V) | 8 | |
| | I3 | Input 3 (5-15V) | 7 | |
| | I4 | Input 4 (5-15V) | 6 | |
| | I5 | Input 5 (5-15V) | 5 | |
| | I6 | Input 6 (5-15V) | 4 | |
| | I7 | Input 7 (5-15V) | 3 | |
| | | I.GND | Optocouplers ground | 2 |
| 8x OUTPUTS | O.COM | Common wheeling diodes | 11 | |
| | O0 | Output transistor 0 | 12 | |
| | O1 | Output transistor 1 | 13 | |
| | O2 | Output transistor 2 | 14 | |
| | O3 | Output transistor 3 | 15 | |
| | O4 | Output transistor 4 | 16 | |
| | O5 | Output transistor 5 | 17 | |
| | O6 | Output transistor 6 | 18 | |
| | O7 | Output transistor 7 | 19 | |

Standard version – mechanical dimensions



Optional accessories for mounting

- The box is equipped with rubber stands, but the package also includes a quality self-adhesive hook and loop fastener, which is suitable for easy and quick mounting to a flat and non-absorbent surface.
- The box is equipped with rubber stands, but the package also includes a metal holder for wall mounting, allowing...
- The box is equipped with rubber stands, but the package also includes a removable holder for a DIN molding, which is mounted from below into the center of the surface.

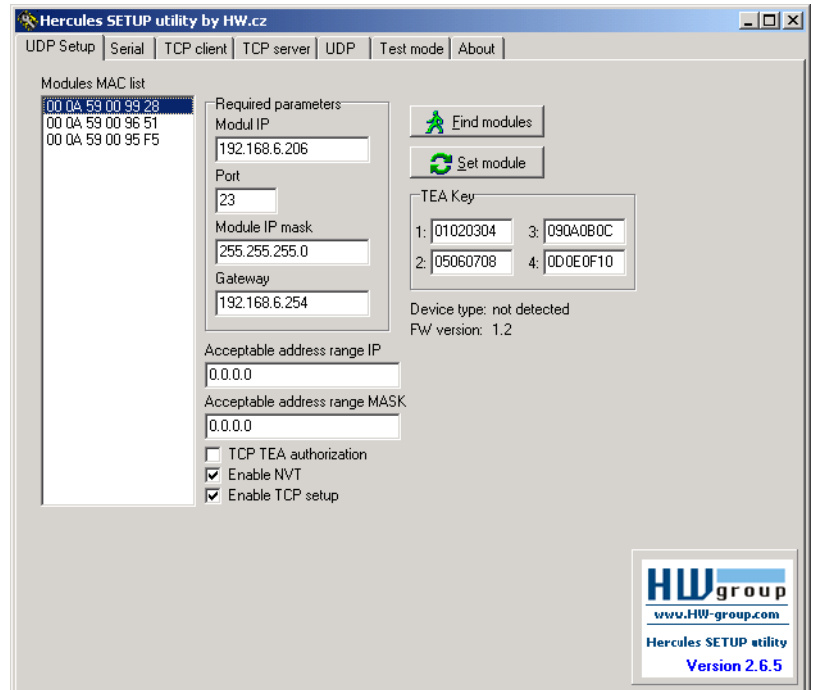
Note: These modifications can be ordered using a specific ordering number (see the last page of this manual).

Quick SETUP

In this chapter, you will learn, how to configure and ping your device as soon as possible. For more information or in case of any problems please read the next chapter.

Connecting cables

- Connect the supplied power adaptor to 230V power source.
- Set **DIP1** and **DIP2** to **OFF** position.
- Connect the device to a **10 Mbit** or 10/100 Mbit Ethernet network.
- Plug the power connector to the device.
- If the power supply is ok, the green **Power** LED is now lit.
- If the Ethernet connection is ok, the **LINK** LED should light up. This LED wil now blink commensurable to data transfered to/from the Ethernet network.



IP address setup

- Run the “HerculesSetup.exe” program, that can be found on the supplied CD, or you can download it from our website: www.HWgroup.cz
- Click the “Find modules” button in the “UDP Setup” tab, you can see the MAC address of the device in the left column. Click on the MAC address and set the desired parameters (you must at least set the IP address, Mask and GateWay).
- Check the “Enable TCP Setup” box and click on the “Set module”. This will save the network parameters.
- The device’s IP address and network parameters are now set and you can work with the device.

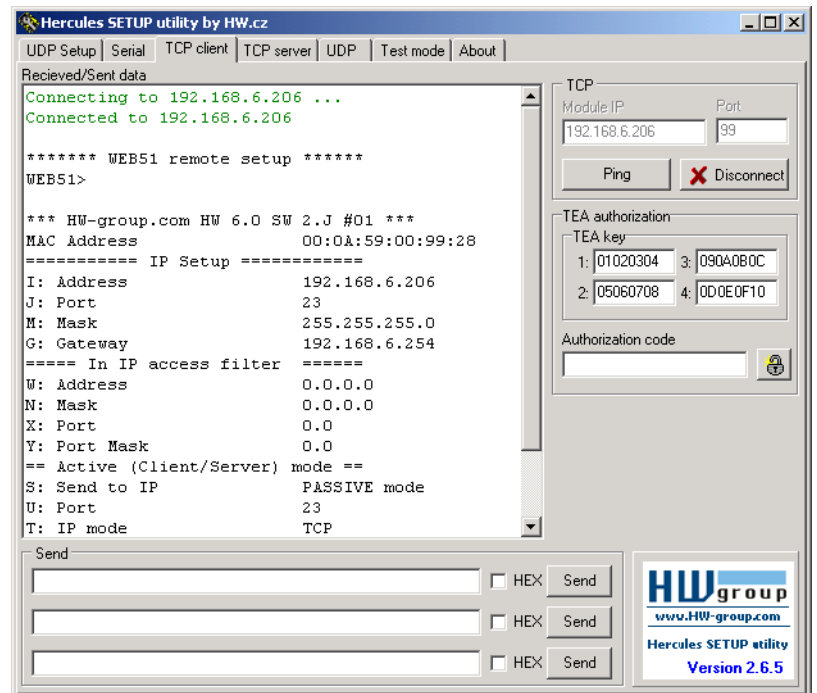
Are you unable to find the device’s MAC address in the list?

If the **LINK LED** is not lit, or the device does not respond, please check the following:

- Does you network support **10MBit** communication?
- Are you using the correct TP cable? (TP Patch cable for connecting to switch, TP Crossed cable for connecting directly to a PC)
- Check the DIP switches (they should all be in OFF position).
- Check your power adaptor and the power LED.

Setting up the device using TCP Setup

- Click on the „TCP Client” tab and enter the preset IP address. As a **TCP port number** choose port **99**.
- Click the „Connect” button, you can see the „**WEB51>**” prompt on the left side. Click in this window and press ENTER. You can now see the parameters listing.
- If you want to change any parameter, you must use the parameter name together with the desired value (for example „**192.168.6.8 <Enter>**” for setting up the device’s IP address). You can see help to any command when you send the commands name together with the question mark (for example „**I? <Enter>**”). You can also find a detailed commands list in another chapter of this manual.
- After you set all the desired parameters, call the “**R**” function (**Reboot**) and press the „Disconnect” button to disconnect from the restarting device.

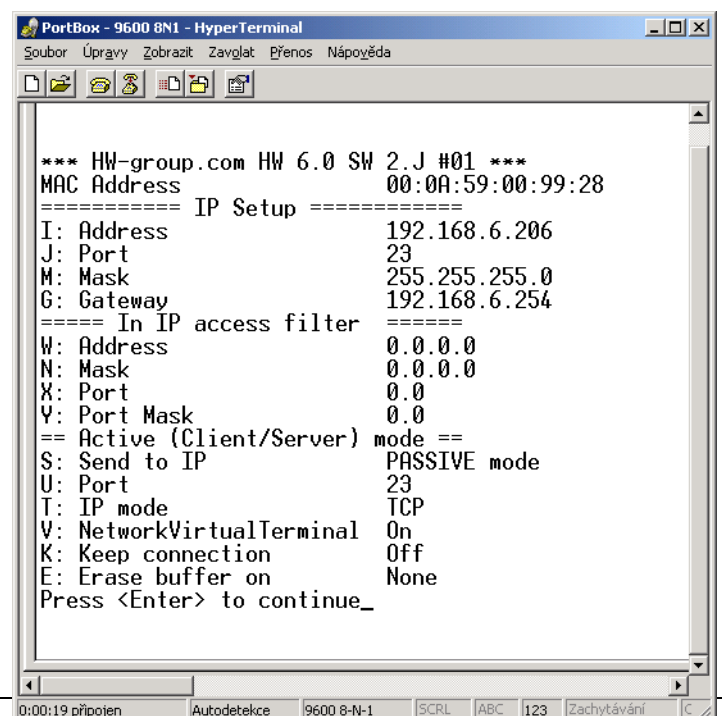


- Note:** the TCP Setup can be turned off using the „Enable TCP setup” checkbox. If this option is unchecked, the device will refuse connection on port 99.

Setting up the device using RS-232 terminal

If you are not using Windows, or if you don't have the Hercules utility, you can setup the device using any available RS-232 serial terminal.

- Set **DIP1 = ON**, **DIP2 = OFF**.
- Connect the RS-232 cable (Port 1) to the PC using the supplied **LapLink** cable.
- Open your favourite terminal software (for example Hyperterminal), choose the correct serial port and set communication parameters to **9600 8N1**.
- Connect the power supply cable to the device.
- If the power supply is ok, the green **Power** LED should light up
- If your terminal and connection are working correctly, you should now see the text menu listing.
- From now on, you can continue as in the previous example.

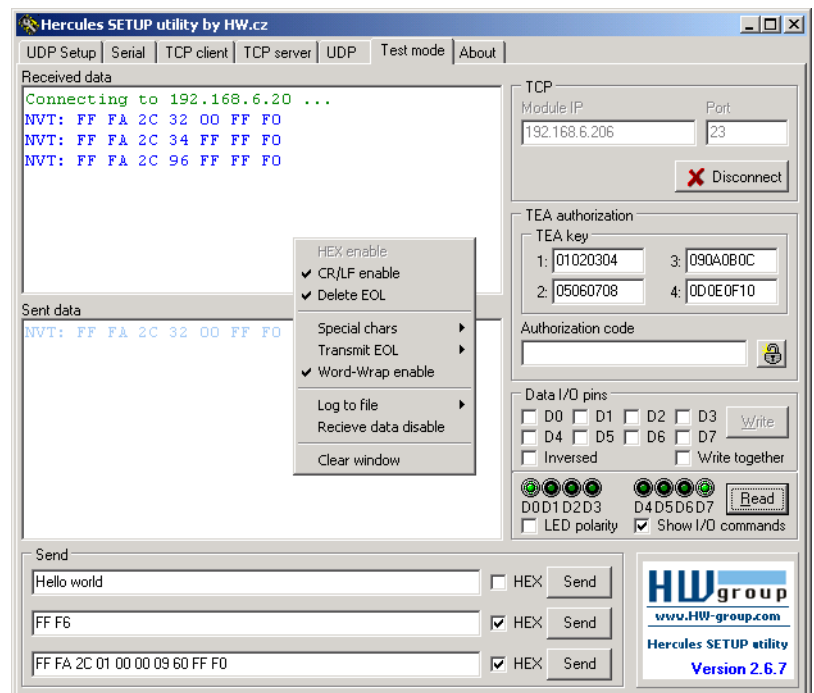


Binary inputs and outputs controlling

In this chapter, we will read the binary input and will set the output to the desired state. We assume using a testing board standardly supplied together with product's samples.

We also assume that you know how to enter the setup mode (RS-232 Setup or TCP Setup on port 99) as described in the previous chapter.

- Set **DIP1 = OFF**, **DIP2 = OFF**, connect the testing board and power the device.
- Click on the „**UDP Setup**” tab, search for the device and make sure you have the „**Enable TCP setup**” and „**Enable NVT**” options checked. If not, check both options and save the settings to the module.
- Doubleclick the device's IP address in the „**UDP Setup**” tab and switch to the „**Test mode**” tab. You will have the IP address and port fields filled. If not, fill in the correct device's IP address and port.
- Click the „**Connect**” button, in the received data windows, you should now see „**Connecting...**” and then on eor three NVT commands. All the recognized commands are blue.
- You can see the listing on the picture. You can access the shown menu by rightclicking in the received data or the sent data window.
- If you now click the „**Read**” button, you will read the input's state (8 switches on the testing board). The input's state is signalized with the virtual D0 to D7 LEDs. You can invert the signalization using the „**LED polarity**” option.
- If you click the D0 to D8 checkbox, you will set the outputs according to the screen. The first command will set all the outputs to the defined state, and then you can setup bit by bit. The output states can also be inverted using the „**Inversed**” option. The „**Write together**” option will not set the output once you check the desired box, but only when you press the „**Write**” button.
- If you have the „**Show I/O commands**” function enabled, notice, that you can see the commands in the received data and the sent data windows. This way you can practice and understand the used commands.
- Write the “FF F6” string to one of the lines in the „**Send**” field, check the HEX option and press the corresponding send button. You will now see „<WEB51 HW 4.7 SW 2.J SN 00A608 #01>” (or similar) in the received data window. You just sent your first NVT command with identification request („Are You There”) and the I/O Controller responded with the HW and SW versions. The string after the „SN” are 3 bytes of the MAC address.
- Do not worry if you encounter the „**Connection refused by remote host**” message, just press the „**Connect**” button and connect again. This is a standard 50 seconds TCP timeout of the module.

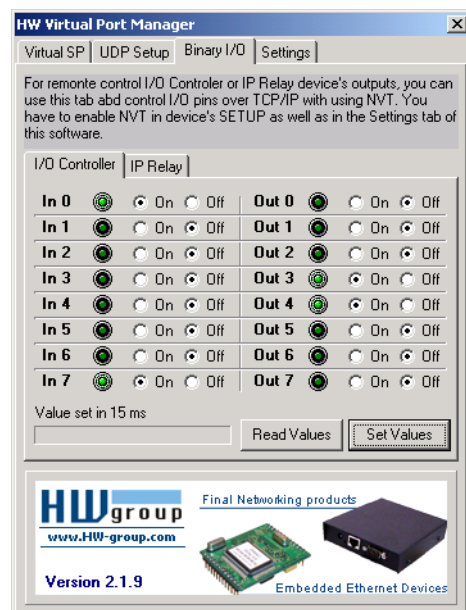


Device setup – Frequently asked questions

- **Ethernet is not working, but the LINK LED is lit.**
Didn't you forget to turn off the „RS-232 Setup” mode? (This mode is enabled when **DIP1** = ON) In this mode, the Ethernet part is not working. Set **DIP1** = OFF and restart the device with unplugging it's power source for at least 3 seconds.
- **The RS-485 communication is not working.**
Check if the line is equipped with **termination resistors** (120 – 470 ohmů). If you are using the RS-485 conversion, set the configuration to **&R2** or **&R3** and dont forget to enable HALF DUPLEX using **&H1**
- **I need power supply for my RS-232 application.**
If you do not need a flow control (HW handshake) but a power source on your serial port (up to 5 – 10 mA), you can power your application from the RTS output (pin 7 on the RS-232 connector). You can enable +8V to +12V power on this pin with setting **&R0** in the setup mode (&R: RS485/RS422 control).
- **I cannot control the binaty inputs and outputs, but I can send the RS-232 data.**
Probably the NVT is disabled. Please check the „UDP Setup” tab in the Hercules setup utility or in the TCP setup or RS-232 setup listing.
- **The 50 seconds timeout is too short, can it be prolonged?**
Yes, just enable the „Keep connection” function. I/O Controller will then send a „NVT: NOP” command every 6 seconds, so the connection will persist.

HW VSP – virtual serial port

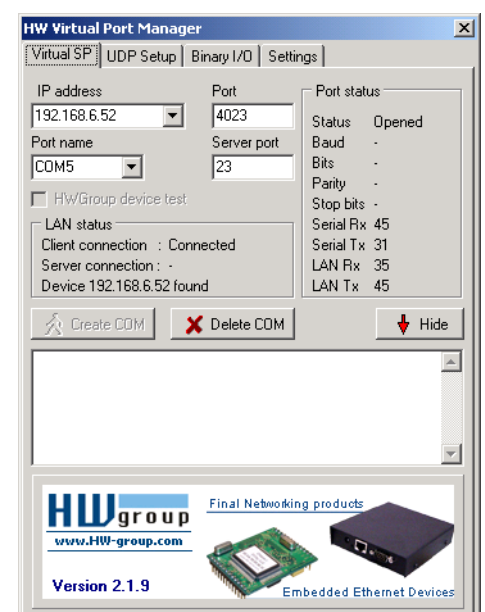
The virtual serial port driver for windows is a software utility, which adds a virtual serial port (for example COM 5) into your system and redirects all the data to another hardware device over the Ethernet network.



- Runs on Windows 2000, Win XP, Win NT 4.0, Win 98
- If the device supports RFC 2217 (NVT) you can set parameters of the remote serial port (baud rate, parity, stop bits).
- We added support of **binary inputs and outputs control** from the VSP software (see picture).
- Debugging is eased by the possibility of a recording the communication into a LOG file.
- It is possible to create more Virtual Serial Ports on a single computer (COM5, COM6, COM7...) by running VSP.EXE via command line.

Using HW VSP with the I/O Controller device

- Run the HW VSP software (in the „**HW_VirtualSerialPort**” directory on our CD). Do not forget to check the option to finish the second part of the installation. When the installation is finished, make sure to restart your PC.
- Check that NVT is enabled (on the **UDP Setup** tab from Hercules utility). If not, enable NVT and save the configuration.
- Run HW VSP and search for the PortStore2 in the „**UDP**” bookmark. Choose the MAC address of the device and click on the „**Use this IP**” button. Switch to the „**Virtual SP**” bookmark. The IP and Port should be already set. This searching via UDP Broadcast works only on a local network.
- Choose the serial port you want to create (COM1 – COM20) and click the “Create COM” button. The frame “LAN status” will show whether the device has been found and if correct, the virtual serial port is created.
- As you run any application which opens selected virtual port (here it's COM5) the HW VSP will link up with PortStore2, set the remote port according to the opened serial port (speed, parity, number of bits, handshake) and starts to transfer data. Some applications have problems with serial port numbers higher than COM4 but you can use the original Hyperterminal from Windows, Serial tab from the Hercules setup utility or Slovene program “Terminal.exe”. These programs are located in the “utils” directory of our CD.
- On the “**Binary I/O**” tab, choose the **I/O Controller** sub-tab and you can set and read the binary inputs and outputs without additional software.



HW VSP settings

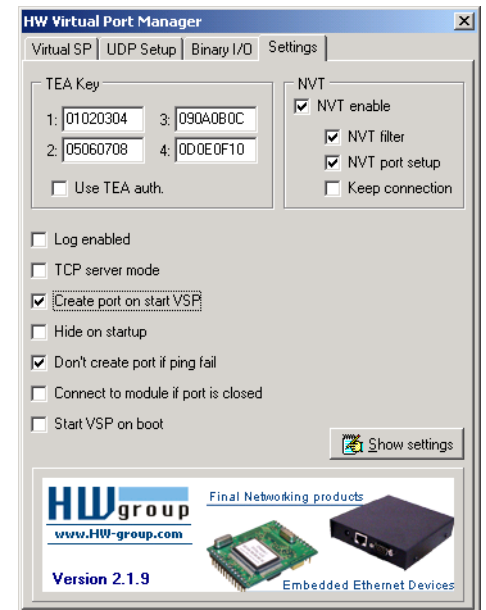
TEA Key frame

You can use the TEA algorithm to secure the TCP/IP access (you must set the same TEA key on both sides of the communication).

NVT frame

Enables the RFC2217 detection of our remote ports. Do not forget to turn on NVT support on the remote device.

- **NVT filter** – filters NVT commands from the data flow.
- **NVT port setup** – sends commands to the remote port according to the VSP settings in your PC. If your terminal program (e.g. HyperTerminal) changes data transfer speed to 19.200 Bd and this function is enabled, the VSP driver sends NVT command (according to RFC2217) to change the remote TCP/IP port's speed as well.
- **Keep Connection** – keeps the TCP/IP connection opened even after 50 seconds of inactivity.



HW VSP parameters

- **log. enabled**
VSP driver creates file “C:\serialport.log” for logging all activity of the virtual serial port.
- **TCP server mode**
Activates VSP as a TCP/IP server. Driver then behaves as a TCP Client/Server = side which first receives any data switches to the role of a client and opens the connection.
TCP server's port, which is used as input can be configured in main „Virtual SP” tab and we recommend using numbers higher than 1025.
- **Create port on start VSP**
If checked all virtual port will be created automatically on startup of VSP. If you need to create virtual ports on Windows startup it's necessary to check the box “Start VSP on boot”.
- **Hide on startup**
Hides VSP to system tray. Icon can be then found next to the clock.
- **Don't create port if ping fail**
IP address of the device is tested before creating virtual port. If testing fails the port is not created.
- **Connect to module if port is closed**
When checked, you can loose data incoming from remote device, if the virtual serial port isn't used by some application.
- **Start VSP on boot**
Adds a path to VSP into register RUN (HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\run) in Windows. VSP is then run on Windows startup.

HW VSP – command line parametry

HW VSP can be also started through specific parameters from command line. This way more virtual serial ports can be created at once. Detailed description of parameters can be found at our website: www.HW-group.com.

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

I/O Controller setup – commands description

Here you can find the detailed description of all the possible setup parameters.

Factory defaults

This setup can be restored with „D0” or „D1” commands in the setup mode. There are two ways to access this mode:

- **TCP Setup** on TCP port 99 (if enabled), you need to restart after changing any parameter.
- **RS-232 Setup**, which is enabled using the DIP1. You can connect to the device using the RS-232 interface (9600 8N1).

```

MAC Address          00:0A:59:00:00:00
===== IP Setup =====
I: Address          192.168.6.41
J: Port             23
M: Mask             255.255.255.0
G: Gateway          192.168.6.254
===== In IP access filter =====
W: Address          0.0.0.0
N: Mask             0.0.0.0
X: Port             0.0
Y: Port Mask        0.0
== Active (Client/Server) mode ==
S: Send to IP       PASSIVE mode
U: Port             4023
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: 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>

```

Device network parameters

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

MAC address is a unique network device address in the Ethernet 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 mode of the configuration 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 .. 19.999.

Port 99 is used for TCP configuration, if supported by the version and enabled in the setup.

M: Mask 255.255.255.0

Configuration of the IP mask for the local network. All IP addresses outside the area delimited by the device's own IP address and this mask will be accessed via the Gateway.

G: Gateway 192.168.6.254

Address of the gateway that provides access to outside networks, as defined by the IP address and the mask.

==== 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 multiplying 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 can communicate with the device. Security can be greatly enhanced by setting a fixed address or a suitable restrictive mask that disallow communication with unauthorized parties.

X: Port 0.0

Y: Port Mask 0.0

This restricts TCP ports that can communicate with the Device.

IP address, Mask and Gateway description:

Ethernet device communicates:

- **On local network segment**
You don't need a Gateway, but IP addresses of both devices must be in the range allowed by the Mask setting. If you set the Mask to 255.255.255.0 the IP addresses may differ only in the last byte.
- **Outside local network – using Gateway**
(which must be in the range allowed by Mask parameter).

You can also restrict the IP address range using the “In IP Setup” commands. We recommend keeping this parameter on 0.0.0.0. for debugging.

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

If this condition is valid, you can access the device (AND is binary multiplication).

- 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**

Configuration of the communication speed for the serial line, range 50..115.200 Bd, step 50 Bd. To set 9600 Bd enter: „&B9600”.

&D: Data bits **8**

Number of data bits for the serial transfer.

7: 7 bits / 8: 8 bits – call „&D8”.

&P: Parity **NONE**

Parity of the serial asynchronous communication:

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

&V: Variable Parity Parity **Off**

Function for the serial 9-bit data transfer. It uses the 0xFE prefix, recommended for box-2-box data transfer. Only a difference from the parity bit's default value is transferred.

Off: incorrect parity bit ignored

On: incorrect parity bit tranfered to other device

&S: Stop bits **2**

Number of stop bits for the RS-232 serial line. This respects the rule of 9 bits (max. 10 bits). If you set 7N1, setup will correct this to 7N2, as well as 8E2 will be 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 signalize that the device can no longer accept data.

1: none – no control, RTS - see &R.

2: RTS/CTS – RTS/CTS control

3: Xon/Xoff – SW data flow control.

&R: RS485/RS422 control On [+8V]

Defines idle level of the output RTS pin. Important for devices powered from RTS or for add-on RS485 Devices that use RTS to switch direction. Especially for the internal RS485 module, the “**HW echo**” option should be on. This means that the receiver reads the data back from RS485 and generates hardware echo from the actual RS485 bus.

- 0: RTS = On [+8V]** (recomended for RS-232 or RS-485 mode)
- 1: RTS = Off [-8V]**
- 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 **&R=3** TxRTS HW echo OFF.

&T: Serial Line Timeout 0 – Off

Specifies how long the Device waits **after receiving the last character** before wrapping up the data in a packet and transmitting them. The timeout is specified as the **number of characters** and displayed as the number of chars as well as the time according to the given serial communication speed. If the speed changes, the time is also changed but the number of characters defining the timeout is not (10 characters at 9600 Bd = about 11 ms, or 5.7ms at 19,200 Bd).

&G: Char. Transmit Delay 0 – Off

For controlling units with small RS232 buffer, it is sometimes advantageous to keep relatively high baud rate but insert delays between individual characters. The delay is defined in **milliseconds** and it is defined as the time between the starts of individual characters; so, for 2400 Bd a 2ms delay has no effect since the character starts are 2.4 ms apart.

&H: Tx Control Tx FULL duplex

When HALF duplex is activated, the Device assumes unidirectional medium connected to the serial line (e.g. RS485) and won't start transmitting data while receiving.

- | |
|--|
| <ul style="list-style-type: none"> 0: FULL duplex (RS-232) 1: HALF duplex (RS-485) |
|--|

Security and remote TCP Setup**==== Security Setup =====****%A: TCP autorisation Off**

Activates TEA authorization (password for establishing a TCP connection) - requested from the remote side after the connection is established.

- | |
|---|
| <ul style="list-style-type: none"> 0: TEA autorisation Off 1: TEA autorisation On |
|---|

%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” option. Set 16 bytes in quadruples using four hexadecimal values separated with colons. The first string defines 0-3rd quadruple of bytes. So, to set the last 4 bytes to the displayed value, use “%K 3:0D:0E:0F:10”.

%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 the port 99)

Inputs and Outputs control

You need to set the TCP/IP mode and NVT (Network virtual terminal) enabled in order to work with the binary inputs and outputs, otherwise you will not even see the commands.

The binary inputs and outputs are controlled **only over the Network Virtual Terminal**. This means, that the I/O control command are integrated in the TCP/IP data stream. These command are only part of the Ethernet communication and cannot be seen on the serial port, as well as the I/O part cannot be controlled from the serial port!

The NVT commands have a predefined binary format and they always begin with a control character. The detailed description of the NVT protocol can be found on our website in the „Support & Download” section.

Note: Do not forget to set the desired value of inputs after reset!

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

#A: Power Up INIT 102

Value in range **0 .. 255 decimal**, which will be written in the **output register** after **device reset** (power failure), before the I/O Controller tries to connect to the opposite side and synchronize the inputs and outputs.

#T: Trigger AND mask 240

Defines inputs, of which changes are **automatically transmitted to the opposite side** (IP address S=X.X.X.X and port U) and synchronized with its outputs, if their value changes. Only transmits inputs, of which bit is set to log. 1 in the #T option.

Examples:

- **#T = 0 (0x00)** – the I/O Controller does not react to changes of inputs from I0 to I8
- **#T = 240 (0xF0)** - I/O Controller only responds to changes on inputs I7, I6, I5, I4. Changes on inputs I3 .. I0 are ignored, but their changed value is transferred when any of the monitored inputs changes.
- **#T = 255 (0xFF)** - I/O Controller responds to any change on inputs I0 to I8

Transmission means, that when the I/O controller is in the Client/Server mode („Active mode”) it reacts to any input’s change as if there is data coming from the serial line. This means that if the connection is closed, it will initialize connection to the predefined opposite side and send the corresponding NVT command to set the output’s values.

If you are using the TCP Server („Passive mode”) and the connection is closed, nothing will happen. If the connection is open, the data will be send.

Note: The inputs are not transmitted to the opposite side, but can of course be read using the NVT commands.

#B: Power Up AND mask 255

Binary range of values from the opposite side, which affect the output's during the first initialization (after reset or power-up).

After RESET data synchronization (Power Up init):

OUTPUT = ((data read from the opposite side AND #B) OR #C) XOR #D

Please note, that I/O Controller recognizes two types of synchronization when two devices are connected to each other. The parameters settings are analogical:

- **Power Up init - (commands #B, #C, #D)** – When the device is reset, I/O Controller set the outputs to the values defined in #A variable and tries to connect to the opposite side's IP address (defined as S=x.x.x.x). When the connection is established, it reads the input's values and sets its outputs according to the described pattern..
- **Data change - (commands #X, #Y, #W, #Z)** – In case of any input's change (delimited by the #T value) the I/O Controller sends this information to the opposite side, which will use the pattern to change its outputs.

The unit tries to connect during the first 50 seconds after power-up. If the connection is unsuccessful, the **#A: Power Up INIT** value will remain set on the outputs.

#C: Power Up OR mask 0

The range of input's values, which will be affected by the opposite side's values read after device reset.

OR – Bitwise sum

| | | |
|------------|-------------|--------------|
| 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) |

Note: *If you are using the embedded Charon 1 module, please note, that its inputs and outputs are joined on one 8-bit port.*

You can use the #C and #Z on the joined inputs and outputs to define what is an input and what is an output. (The inputs must be set as log. 1). For example #Z=0x0F, the lower four bits (D0, D1, D2, D3) will always be log. 1, so they can be used as inputs.

#D: Power Up XOR mask 0

The binary XOR function is suitable for output's bits inversion. This is used for example when you connect a button against GND, but you need to connect a relay against the + PWR signal (the log. 0 on the input must be inverted to log. 1 on the output).

Examples:

- **#D = 0 (0x00)** – The binary values are transmitted 1:1 = Log. 1 on the input will be Log. 1 on each output.
- **#D = 1 (0x01)** – the D0 output is inverted against the D0 input. Binary values D1 to D7 are transmitted 1:1 = Log. 1 on the input will be Log. 1 on each output.
- **#D = 255 (0xFF)** – All output values are inverted against the input values.

#X: KEEP mask 0

Defines range of outputs, where the previous value will persist using the bitwise product.

Common data synchronization:

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

- **PrevOut** = previous output's state

- **RxData** = the received data

Note: *The previous pattern is used only when two I/O controllers are connected to each other. Inputs from one side are automatically transmitted to the other side's outputs. This is called a **Box-2-Box mode**.*

You can access all the 8 output bits using the standard NVT commands, no matter what the #A to #Z values are.

You cannot access the output's settings from your PC in the common Box-2-Box mode, because the I/O Controller supports only one TCP connection at a time.

#Y: AND mask 255

Defines the bit range transmitted from the opposite side's inputs to this device outputs.

Note: *Please be careful when setting the outputs over NVT and using Box-2-Box mode at the same time. The #X and #Y can define, which bits can be set only over NVT and which bits will be synchronized.*

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

AND – Bitwise product

| | | |
|------------|---------------|------------|
| 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) |

#W: XOR mask 0

The binary XOR function is used for output bits inversion (against opposite input bits) – see the #D parameter description.

XOR – Bitwise negation

| | | |
|------------|----------------|--------------|
| 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) |

#Z: OR mask 0

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

Setup examples

Using the described parameters, you can set each bit its independent value, synchronize with inputs changes, or invert its logical value. Read the following examples to get you started:

Output = (X.n =0 Y.n =0 Z.n =0) - persistent 0 on the output, can be set only using the NVT commands

Output = (Z.n =1) - persistent 1 on the output, can be set only using the NVT commands

Output = (X.n=0 Y.n=1 W.n=0 Z.n=0) – bit copy (tunnel) of the opposite side

Output = (X.n=0 Y.n=1 W.n=0 Z.n=1) - bit copy (tunnel) of the opposite side + inversion

Note: *Commonly, the inputs are scanned each one ms and a change is considered when the input's value differs from the last value in two following scans (= the change last at least 1,2 to 2,0 ms).*

Other device parameters

Q: Quiet (Batch) mode

If you need to manage device's configuration from your SW application, you have to work with the list of the device's settings. It can be helpful the quiet mode, you can enable it with „**Q1**“. In the Quiet mode, device reply to pressing Enter not in 2 pages parameters listing but under example:

```
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
```

===== Other =====

D: Load/Save Settings from/to Flash

Saves the current settings to memory or restores saved settings.

0: Reloads settings from **position 1**

1: Reloads settings from **position 2**

2: Stores settings to **position 1**

3: Stores settings to **position 2**

R: Reboot

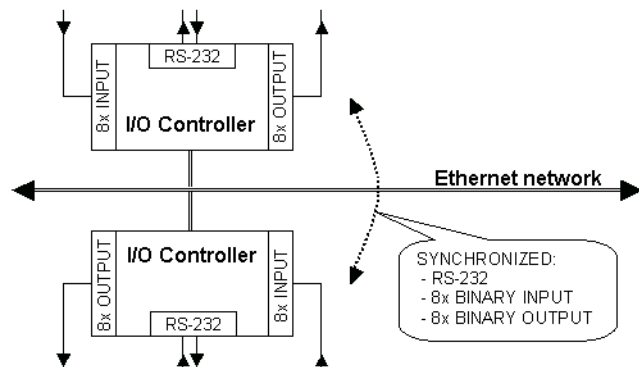
Software reset. Recommended when changing TCP Setup parameters.

Practical examples of the configuration

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

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

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



The serial port runs on 19.000 Bd, 8N1, SW handshake, the input's values are transferred to the corresponding outputs.

TCP will automatically close after 50 seconds, and will open again when there is an input change, or when any data is received on 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 only one address which the device would communicate with, set N:255.255.255.255 and W:IP address of the other device.
- If you want a 9-bit communication, enable &V1 = &V: Variable Parity On on both devices.
- If you want to rise the security level, disable on both devices TCP Setup on port 99 %S0 = %S: TCP/IP setup Off.

If you need to improve the security level and allow I/O Controller A to communicate only within local network segment ((0 to 255 at the end of IP address)) use following setting.

```
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
```

I/O Controller on the Charon 1 module

the I/O Controller application also works on the Charon 1 modules. Only the inputs and outputs are joined on one 8-bit port. The following example only differs in transferring 4+4 bits, instead of 8+8 bits over the Ethernet network.



Charon 1 – module 1

```
===== I/O Control Setup =====
#T: Trigger AND mask      15
#A: Power Up INIT        15
#B: Power Up AND mask    240
#C: Power Up OR mask     240
#D: Power Up XOR mask    0
#X: KEEP mask            0
#Y: AND mask              240
#Z: OR mask               15
#W: XOR mask              0
===== Other =====
D: Load/Save Settings from/to Flash
R: Reboot
WEB51>
```

Charon 1 – module 2

```
===== I/O Control Setup =====
#T: Trigger AND mask      240
#A: Power Up INIT        240
#B: Power Up AND mask    15
#C: Power Up OR mask     240
#D: Power Up XOR mask    0
#X: KEEP mask            0
#Y: AND mask              15
#Z: OR mask               240
#W: XOR mask              0
===== Other =====
D: Load/Save Settings from/to Flash
R: Reboot
WEB51>
```

Inputs = D0, D1, D2, D3 (→)
Outputs = D4, D5, D6, D7 (←)

Outputs = D0, D1, D2, D3 (→)
Inputs = D4, D5, D6, D7 (←)

- If you need to invert for example the D6 output on module 1 against the D6 input on module 2, set **#D = #W = 64** in the module 1 configuration.
- If you need to keep the communication running, set „**K: Keep connection On**” on one of the devices. After this, the modules only communicate with each other and you cannot connect to them from the PC.

The input's and output's control using the NVT

The following brief descriptions of the M2M protocol (NVT) device control is only part of the detailed description that can be found on our website:

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

What is NVT and RFC2217

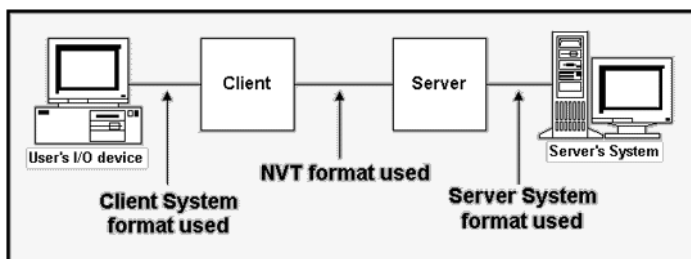
NVT is a way of integrating the commands into the data stream. The NVT (Network Virtual Terminal) is also used when transferring commands over the Telnet protocol. Telnet uses this to transfer special commands like CTRL-Pause, for setting the cursor position on the screen, terminal type switching, etc...

The RFC2217 was defined by Cisco, for changing the baud rate of the serial port, reading binary signals etc... Most of these commands are implemented in our devices. You can find the device list in the NVT detailed description on our website.

We extended the RFC2217 standard to incorporate some GPIO (General Purpose Input Output) functions listed below. These functions can control the I/O Controllers digital pins. It's not standardised, but we didn't find any standards for GPIO functions during the year 2001.

How does it work?

The commands are part of the TCP/IP data stream; where the „FF” character starts the control sequence in a defined format. If the data contains the „FF” (255 decimal) character, it must be doubled by the transmitting side,



while the receiving side will convert it back to one „FF” character. If there is NVT support on both sides, the „FF” character is always a beginning of the control sequence.

Commands for GPIO - 50, 51, 52 (32 .. 34 hex) control

For direct I/O pin control we use the double byte command GPIO-50 or 51 (which is behind the COM-PORT-OPTION 44 command) followed by the 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's state reading - the answer contains the value of the input port.
- **16 .. 23 (10 .. 17 hex)** – set the output bit 0..7 as log. 1
- **32 .. 39 (20 .. 27 hex)** – set the output bit 0..7 as log. 0
- **48 (30 hex)** - Request for output's state reading – the answer contains the value of the output port

| 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 from 100 Dec = **Server >> Client**

CAS_ requests the device to execute a command
ASC_ device's response, command executed + values confirmation

The commands start with a <IAC><SB> (**FF FA**) sequence and end with a <IAC><SE> (**FF F0**) sequence.

Sub option 51 (33 hex)

Sets a value to the output port. Returns the same value (read from the inner register) as an answer.

Sub option 52 (34 hex)

This command is used when the I/O Controller **transmits the input values automatically, if there has been an input change**, or after device power-up. This command does not request answer, therefore there is no value for 152 (98 HEX) in the table. This is because an automatic report is basically an unrequested answer!

Before the 52 (34 hex) command there is commonly a „**FF FA 2C 32 00 FF F0**” sequence, which is a request for output values. When you answer this question with I/O Controller’s outputs settings (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 values for the other side’s outputs.

Note: *to activate this function, you need to enable the transmitted inputs range using the „#T:Trigger AND mask” command.*

- Set #T=255 to transmit all inputs
- Set #T=0 to turn this function off

NVT control examples

Most of the NVT commands have a fixed length. If the values has a 4 byte length, and you want to read the current value with „setting” the 0 values, you need to send this 0 value as 00 00 00 00 hex.

Output byte setting

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
```

The answer confirms the port’s setting:

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

Input reading

Request for the XX value on GPIO.

```
sequence: FF FA 2C 32 00 FF F0
I/O Controller answers with: FF FA 2C 96 XX FF F0
```

Output reading

Request for the output’s register value (XX). This register is used to set the output pins.

```
sequence: FF FA 2C 32 30 FF F0
I/O Controller answers with: FF FA 2C 97 XX FF F0
```

D5 output reset

Set on eoutput to log. 0, without affecting the other outputs

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

Where DF is the actual output port's values (depend also on the previous port's state). I/O Controller changes only one bit, but the answer contains the actual value.

Output change reporting

Unrequested report of any output change. It can start with the FF FA 2C 32 00 FF F0 sequence – see the detailed description.

sequence: none
 I/O Controller answers with: **FF FA 2C 34 XX FF F0**

Where XX is the actual output port's value.

Note: *The function works similar to the serial port data. This means, that if you work in the TCP Client/Server mode with NVT enabled, the device tries to initialize a connection when any serial data is received or any input bit has changed. Only the current input values will be 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 actual port speed can be determined with sending the **00 00 00 00** sequence. If you send any other values, the server will set the serial port speed to this values.

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

If you convert the 00 00 25 80 to decimal => the actual speed is **9600 Bd**.

Keep Connection

You can use the “**K: Keep connection**” command to override the standard TCP connection timeout (the connection will close after 50 seconds of inactivity). NVT must be enabled for this to work. The connection is held open with sending the NOP (**FF F1**) commands each 5 seconds.

Are You There?

There is a way to determine, whether there is device connected. It's the „Are you there” sequence, which a unix device will answer with some kind of “Yes”.

sequence: **FF F6**
 I/O Controller answers with: <WEB51 HW XXX SW XXX SN XXX #0F *OvErr *ParErr *FIErr>

The XXX values have proprietary meaning, or can be bypassed if they are error messages. Example of answer: <**WEB51 HW 4.5 SW 2.3 SN 01A03B #01**>

Ordering number

| OID | Device types |
|---------|---|
| 600 291 | I/O Controller PL - Terminal Block version Standard version with 20-pin connection block, standalone I/O Controller unit. The connectors "20 pin connector" [600 034] included. |
| 600 292 | I/O Controller PL set Recommended starting set, it contains: - I/O Controller PL [600291] - 20 pin connector [600 034] - EU Power adaptor [600080] - I/O Controller Test board [600032] - DB9 LapLink cable 2m [600063] - Manual, CD |
| 600 025 | DIN Rail box set for mounting to a DIN molding |
| 600 295 | Metal holder for wall mounting |
| 600 301 | Self-adhesive hook and loop fastener 3M™ Dual Lock™ |

Free application software

You can use the supplied software for device setup or as an inspiration when programming your own applications. A detailed description of the routines, examples and NVT can be found on our website.

- **Borland C++**
A simple TCP Client terminal and basic NVT commands (Are You There and I/O functions).
- **Delphi**
[Delphi Charon 1](#) – Communication example and NVT - (TCP Client).
[Delphi TCP/IP logger/server example](#) - (TCP Server).
- **JAVA**
A TCP/IP – NVT (TCP Client) communication example.
- **PHP**
A TCP/IP – NVT (TCP Client) communication example.
- **Visual Basic** – A TCP/IP a NVT (TCP Client) Communication example.
- **SDK** – Request full Software Development Kit to I/O Controller

